

CAPITULO I

1.- METODOLOGÍA

1.1. QUE ES METODOLOGÍA

Según de la Torre (1991), debemos considerar el método como "un proceso lógico, surgido del raciocinio de la inducción".

Según de Lasty Balseiro (1991), el método general de la ciencia "es un procedimiento que se aplica al cielo entero de la investigación en el marco de cada problema de conocimiento".

En <http://es.wikipedia.org/wiki/Metodolog%C3%ADa> establece que: "Metodología, del griego (metá "más allá" odós "camino" logos "estudio"). Se refiere a los métodos de investigación que se siguen para alcanzar una gama de objetivos en una ciencia. Aun cuando el término puede ser aplicado a las artes cuando es necesario efectuar una observación o análisis más riguroso o explicar una forma de interpretar la obra de arte. En resumen son el conjunto de métodos que se rigen en una investigación científica o en una exposición doctrinal.

A diferencia de lo que sucede con el epistemólogo, el metodólogo no pone en juicio el conocimiento ya obtenido y aceptado por la Comunidad científica. Su problema se centra en la búsqueda de estrategias válidas para incrementar el conocimiento.

Por ello, la metodología se entenderá aquí como la parte del proceso de investigación (Método Científico), que sigue a la propedéutica, y permite sistematizar los métodos y las técnicas necesarias para llevarla a cabo.

Según Martínez Migueles (1999), los Métodos "Son vías que facilitan el descubrimiento de conocimientos seguros y confiables para solucionar los problemas que la vida nos plantea".

La metodología, como etapa específica que dimana de una posición teórica y epistemológica, da pie a la selección de técnicas concretas de investigación. La postura filosófica acerca de la ciencia de la que parte el investigador, orientará su elección metodológica, es decir, lo guiará a la hora de resolver: cómo investigar el problema de investigación, con bases racionalistas, empiristas, pragmáticas, constructivistas, con un sentido crítico, escéptico o dogmático, con un enfoque positivista o dialéctico hermenéutico.

La metodología dependerá, de esta forma, de los postulados que el investigador considere como válidos de aquello que considere objeto de la ciencia y conocimiento científico pues será a través de la acción metodológica como recolecte, ordene y analice la realidad estudiada.

No existe una metodología que sea la medicina absoluta, así que aparecen muchas veces ellas mezcladas unas con otras en relación simbiótica.

La validez otorgada al uso de uno u otro método estará dada en el marco de los paradigmas de la ciencia.

1.1.1. LA NECESIDAD DE UNA METODOLOGÍA

En <http://www.geocities.com/autogestion/metodologia/metodologia.html> establece que: “Los profesionales de distintas disciplinas trabajan con cierto método, lo mismo hace el especialista o facilitador de la participación. Facilitar la participación en el desarrollo es una especialización legítima y requiere habilidades especializadas. Ningún buen especialista en participación puede proceder sin un “mapa de carreteras” o metodología de trabajo.

Por ejemplo, si un Líder del Proyecto encarga una tarea o actividad a un grupo de interesados locales sin darles guías, objetivos, lineamientos los resultados pueden ser desastrosos. Los directores o líderes no dejan de ser responsables por los resultados simplemente porque han involucrado a un grupo de personas. El enfoque participativo funciona mejor cuando hay un liderazgo fuerte e innovador diferenciado del modelo burocrático tradicional.

La esencia de una metodología participativa se manifiesta en la transición de un deseo en un primer intento y de este intento a la práctica”

1.1.2. SELECCIÓN DE UNA METODOLOGÍA

En <http://www.geocities.com/autogestion/metodologia/metodologia.html> establece que: “La metodología que se escoja debe tener en cuenta lo siguiente:

- ❖ La naturaleza del proyecto;
- ❖ Los resultados que desean obtener los que proponen el proyecto;
- ❖ Las características de los diferentes interesados;
- ❖ El nivel de los distintos interesados en la jerarquía socio-política.
- ❖ Las relaciones sociales existentes - incluyendo los conflictos latentes y manifiestos;
- ❖ La experiencia participativa que los interesados locales hayan tenido.

Todos estos factores son claves para la selección de la metodología más apropiada para el proyecto. Requieren pensar cuidadosamente y evitar tomar decisiones apresuradas. Es conveniente tener en cuenta las siguientes consideraciones antes de decidir:

- ❖ Explicitar claramente el propósito del proyecto o actividad propuestos. El método deriva directamente de este propósito.
- ❖ Preguntarse "¿qué trata de lograr la participación en este proyecto particular?". La respuesta a esta pregunta permitirá hacer sugerencias en cuanto a la metodología y las técnicas "más adecuadas".
- ❖ Es conveniente usar una variedad de técnicas en su metodología. Las mismas deben complementarse mutuamente y permitir la inclusión de una amplia gama más amplia de interesados y de ideas.
- ❖ La metodología escogida y los responsables de su aplicación deben tener experiencia y ser sensibles al medio cultural y social en que se aplicará”.

1.1.3. CONCEPTO DE METODOLOGIA PARA EL DESARROLLO DE SOFTWARE

En <http://www.geocities.com/trescapas/msf.htm> establece que: “El modelo del equipo de MICROSOFT SOLUTION FRAMEWORK (MSF por sus siglas en inglés) describe el enfoque de Microsoft a fin de organizar tanto a las personas como a sus actividades a fin de permitir el éxito del proyecto. El modelo define los grupos de roles, áreas funcionales, responsabilidades e instrucciones para que los miembros del equipo los dirijan de tal manera que puedan alcanzar sus metas únicas durante el ciclo de vida del proyecto.

A fin de maximizar el éxito de los proyectos y operaciones durante todo el ciclo de vida. MICROSOFT SOLUTION FRAMEWORK y la Estructura de Operaciones de Microsoft (MOF) proporcionan orientación y prácticas probadas para planificar, construir, poner en marcha y dirigir soluciones en forma efectiva. Esta información se deriva de la experiencia ganada durante el desarrollo del software a gran escala de Microsoft y los proyectos de operación de servicios, la experiencia de los consultores de Microsoft, y las prácticas más comunes de las industrias a nivel mundial. Se entrega en forma de reportes, guías, herramientas, plantillas, estudios de casos prácticos y cursos educativos. La orientación y prácticas están organizadas en dos conjuntos de conocimientos complementarios y bien integrados”.

1.2. METODOLOGIA CUALITATIVA

En <http://pdf.rincondelvago.com/introduccion-a-la-metodologia-cualitativa.html> establece que: “La Metodología Cualitativa describe estrategias con objetividades claras y precisas, es así como está dividida de la siguiente manera:

➤ **Definición**

- ❖ Estrategia de Investigación
- ❖ Descripción del contexto con la máxima objetividad
- ❖ Conocimiento válido con potencia explicativa.

➤ **Axiomas (supuestos)**

- ❖ Concepción múltiple de la realidad (como fenómeno complicado y no evidente: conocer qué elementos la componen, cómo intervienen)
- ❖ Comprensión de los fenómenos (La investigación experimental pretende explicarlos)
- ❖ Interacción entre investigador y objeto investigado.
- ❖ Conocimiento ideográfico. Descripción de casos.
- ❖ Dificultad para establecer relaciones de causa–efecto (no es su objetivo)
- ❖ Valores implícitos en la investigación (evitando presupuestos, intenta situarse en posiciones objetivas.

➤ **Características de la metodología cualitativa.**

La fuente de datos son las situaciones naturales (algunas estrategias provocan también fenómenos, intentan romper la rutina diaria de un grupo determinado para ver qué sucede).

El investigador recoge los datos (a través de las distintas técnicas, como observación, análisis de contenidos, encuestas, grupos de discusión.)

Utilización del conocimiento tácito (a través de la experiencia y la interacción, es un conocimiento experto; aparece tanto a la hora de recoger la información como a la hora de analizar los datos; en la investigación experimental sólo se daría en el momento de definir los problemas e hipótesis).

Recogida de datos abierta (en la continua interacción se aprecian nuevos campos para obtener información e informaciones no previstas).

Muestreo intencional (no probabilístico, se buscan datos que nos den información de claves para entender el contexto en que nos movemos).

➤ **Relación entre cuestiones, estrategias y técnicas de investigación.**

¿Por qué tantas estrategias? ¿Por qué tanto enfrentamiento entre métodos cuantitativos y cualitativos?

La elección del método dependerá de la realidad social que queramos investigar. Será el problema a estudiar el que determine el método a emplear.

No existe enfrentamiento entre el uso de uno u otro método, hay que tomar lo importante en cada uno.

El objetivo es conocer la teoría que nos llevan a investigar de una determinada manera”.

1.3. METODOLOGIA CUANTITATIVA

En <http://www.uacm.edu.mx/bestudiante/imagenes/cuantitativa.pdf> establece que: “El curso de Metodología Cuantitativa es el primero del Eje de la Metodología. Aquí se pretende que el estudiante comprenda la naturaleza de la investigación, entendida ésta, como bien plantea Bunge, como una forma de conocimiento racional, sistemático, exacto, verificable y por consiguiente falible. En este curso, el estudiante se adentrará en la aventura que comporta todo proceso de investigación. El reto será descubrir cómo podemos plantear interrogantes susceptibles de ser resueltas desde una perspectiva científica. Para ello habrá que hurgar en las áreas de interés que tenga cada uno.

Eso le permitirá al estudiante darse cuenta que la investigación surge siempre del interés general que cada uno tiene sobre ciertas cosas. Hay muchos temas sobre los cuales conocemos poco y la idea será justamente encontrarle el sentido a la investigación ahí: partir de la curiosidad por descubrir cosas que no conocíamos antes.

Se aprenderá a construir problemas de investigación. Se cuestionará sobre los temas de interés, se formulará preguntas, se pondrá objetivos, justificará el tema elegido y explorará la viabilidad de la investigación que plantea.

Para llevar a cabo esta labor, trabajará colectivamente con el emisor y receptor. Así descubrirá que una característica fundamental de la investigación es la discusión. Poner en común con el otro las ideas ayuda siempre a esclarecer y fortalecer nuestros argumentos, nos ordena la mente.

Por otro lado, se hará particular hincapié en que los problemas planteados puedan ser resueltos a partir de la metodología cuantitativa. Para eso, será necesario en primer lugar que el identificar las diferencias básicas entre la investigación cuantitativa y la investigación cualitativa. A partir de esta primera diferenciación, se conocerá los principios constructivos de la investigación cuantitativa y sabrá identificarse correctamente”.

1.4. METODOLOGIAS PARTICIPATIVAS

En

http://www.regioncajamarca.gob.pe/descargas/desecho/coremype_07_adex_material_metodolog_a_20participativas.pdf establece que: “La participación se ha convertido en un término de moda, de uso obligado por quienes hacen desarrollo. Como la señalan Farrington y Bebbington, el concepto de la participación ha sido motivo por la ideología como por el pragmatismo, vale decir tanto por el compromiso honesto de devolver el protagonismo a los sin poder como para hacer eficaz y sostenible un propuesta.

Pero también puede ser un maquillaje de las metodologías convencionales, que utilizan el concepto para mantener su hegemonía. Sus prácticas y sus principios absolutamente intocados.

Hay bastante tinta que sostiene que es un nuevo mecanismo de cooptación que usa el poder para evitar que se desborde la sociedad civil, mientras otros sostienen que son técnicas neutras que pueden servir tanto por la cooptación como para el “empoderamiento” (reforzamiento de la capacidad y del poder) de los grupos de base, que su distinto uso depende más bien del contexto, los fines y de quien las usa.

Las metodologías participativas adquirieron fuerza en el mundo en las dos últimas décadas. No hay duda alguna que ellas tomaron cuerpo en un contexto mundial que había revalorizado la democracia, pero que al mismo tiempo resultaba esquiva para las grandes mayorías”

1.5. MSF (MICROSOFT SOLUTION FRAMEWORK)

En <http://www.geocities.com/trescapas/msf.htm> establece que: “La manera más fácil de llegar al éxito es seguir los pasos exitosos de otros. Esto no significa que se debería esperar a ver qué es lo que la competencia está haciendo, y luego colocar esto en un producto duplicado o copiado. La idea es aprender a hacer las cosas de la misma manera que lo hacen “los grandes” siguiendo sus planes de acción. Esto es lo que hace MSF; toma las mejores prácticas de los expertos y las integra en distintos modelos, principios y guías”.

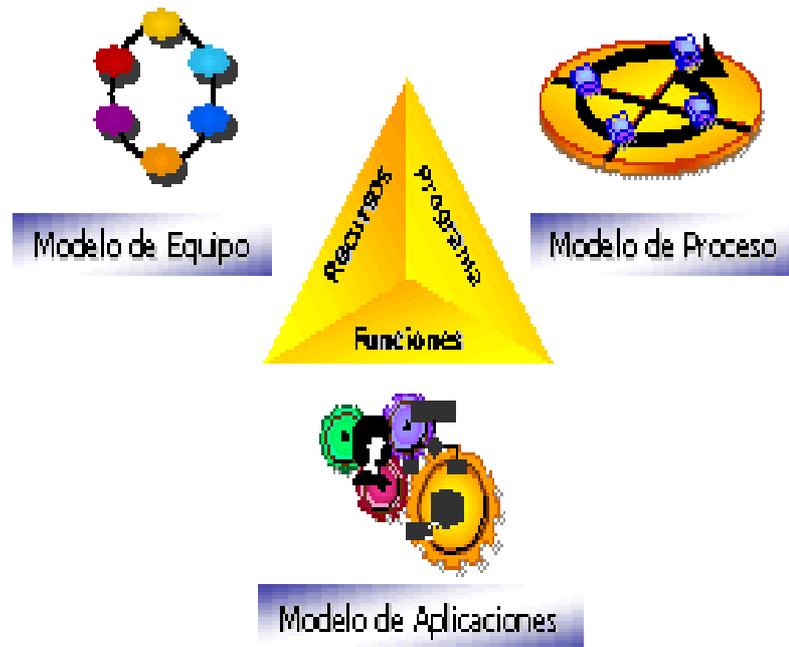


Gráfico N°- 1

<http://www.geocities.com/trescapas/msf.htm>

En http://www.innovaempresarial.com/docs/Metodologia_Desarrollo.pdf establece que: “MSF sirve como guía para administrar la gente y los procesos en el desarrollo de software y contempla:

- ❖ El modelo de equipo enfocado a la administración de recursos
- ❖ El modelo de la aplicación enfocado a la funcionalidad del desarrollo
- ❖ El modelo de proceso enfocado a la programación del desarrollo.

1.5.1. MODELO DE EQUIPOS

Los equipos organizados bajo el Modelo de Equipos del MSF son pequeños y multidisciplinarios. Los miembros comparten responsabilidades y complementan sus habilidades para enfocarse al proyecto.

Comparten una visión de proyecto, un objetivo claro de desarrollo, altos estándares de calidad y una disposición para el aprendizaje. El equipo trabaja en conjunto, con cada miembro con redefinidos, en los que cada rol adquiere relevancia en las distintas etapas del proceso de desarrollo.

Existen 6 roles en el Modelo de Equipos:



Gráfico N°- 2

http://www.innovaempresarial.com/docs/Metodologia_Desarrollo.pdf

ROL

OBJETIVO

Gerente de Producto	Cliente satisfecho
Gerente de Programa	Entrega dentro de las restricciones del proyecto
Desarrollador	Entrega en función de especificaciones
Pruebas	Aseguramiento de funcionalidad
Educación a usuarios	Máximo aprovechamiento del producto por el usuario
Gerente de Logística	Asegurar el despliegado de la solución Roles.

1.5.2. MODELO DE LA APLICACIÓN

Este modelo contempla un diseño lógico en tres capas para el diseño de aplicaciones (soluciones) distribuidas multicapas. Define una aplicación como una red lógica de servicios distribuibles y reutilizables que cooperan en tareas comunes.

Contempla tres categorías de servicios:

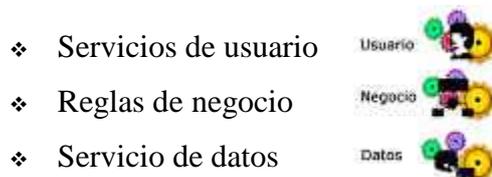


Gráfico N°- 3

http://www.innovaempresarial.com/docs/Metodologia_Desarrollo.pdf

Gracias al seguimiento de este modelo, los desarrollos pueden ser reutilizables y diseñados de manera modular. Ello permite que una solución crezca y sea escalable al permitir fácilmente que algún módulo existente se modifique sin afectar los demás componentes, ó que se agregue un nuevo módulo encargado de nueva funcionalidad.

1.5.3. MODELO DE PROCESOS

El Modelo de Procesos de MSF provee una estructura para el desarrollo de aplicaciones que consiste en 4 etapas distintas, cada una de las cuales culmina con una meta definida. Las 4 etapas son:

1. Visión
2. Planeación
3. Desarrollo
4. Estabilización

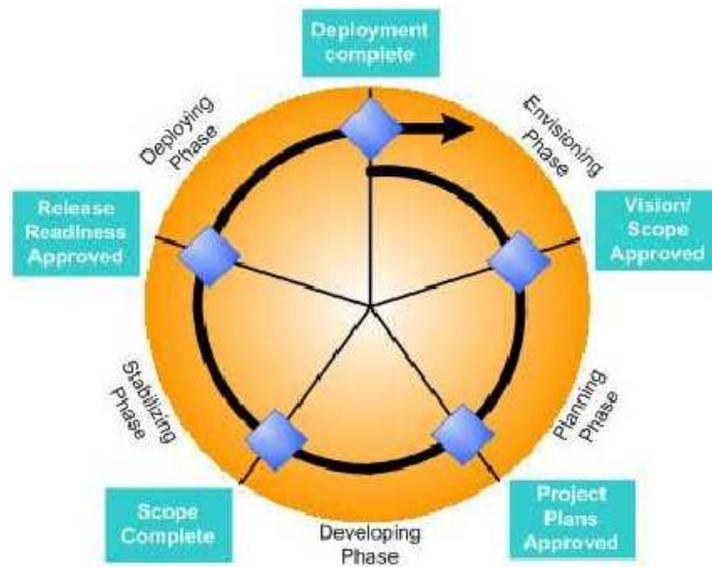


Gráfico N°- 4

http://www.innovaempresarial.com/docs/Metodologia_Desarrollo.pdf

Etapa 1.- Visión

El propósito de esta etapa es que el equipo y el cliente definan el objetivo del proyecto y las restricciones del mismo. El principal entregable de esta etapa es el Documento de Visión, que contiene un análisis del problema del cliente, la descripción de los objetivos de la solución, un esquema de la conceptualización de la solución, perfil de los usuarios de la solución y objetivos del diseño. El alcance del proyecto también se determina. Para cada proyecto se solicita la participación de un miembro del cliente, con el fin de que adquiera el rol de gerente de producto y coadyuve a la conceptualización de la solución.

Esta etapa culmina con la meta de Visión Aprobada.

Etapa 2. Planeación

En esta etapa el equipo hace las especificaciones funcionales, un plan maestro para el proyecto y un calendario maestro. Las especificaciones funcionales describen lo que se va a desarrollar, incluyendo contenido como metas de diseño, requerimientos, características y dependencias.

Comprende el diseño conceptual, lógico y físico. El plan maestro describe cómo se va a desarrollar la solución con qué recursos y costos. El calendario maestro describe cuándo y en qué orden se va a desarrollar la solución.

Así mismo se documentan los posibles riesgos que pudieran modificar el plan de desarrollo.

Esta etapa culmina con la Aprobación del Plan, lo que representa que el desarrollo puede iniciar.

Etapa 3. Desarrollo

En esta etapa el equipo se enfoca en desarrollar y probar la solución. Esta etapa involucra una serie de entregas internas de componentes del desarrollo, hechas en paralelo y en segmentos, para medir el avance del desarrollo y asegurar que los componentes interactúen entre sí. Las pruebas de calidad, son parte integral de esta etapa, pues el rol de pruebas no implica únicamente encontrar errores, sino asegurar la calidad y que el producto resuelva las necesidades del cliente.

En este sentido, se harán pruebas de cobertura, enfocadas a probar las características y código de la solución y pruebas de uso, enfocadas a probar la solución en el ambiente esperado de trabajo. Otra parte importante de esta etapa es la administración de errores.

Gracias al seguimiento de este modelo, los desarrollos pueden ser reutilizables y diseñados de manera modular. Ello permite que una solución crezca y sea escalable al permitir fácilmente que algún módulo existente se modifique sin afectar los demás componentes, ó que se agregue un nuevo módulo encargado de nueva funcionalidad.

Etapa 4. Estabilización

Esta etapa comienza con pruebas beta de la solución y termina cuando se libera el producto con el cliente. Las pruebas en esta etapa se enfocan en uso en condiciones reales. El equipo se concentra en corregir posibles fallos de la solución y preparar el producto para su liberación. Cuando la solución se libera, ésta se transfiere el administrador de la misma por parte del cliente”.

1.5.4. ORIGEN

En <http://diegumzone.spaces.live.com/blog/cns!1AD5096D63670065!119.entry> establece que: “Que las empresas opten en algún momento por imponer a sus desarrolladores la aplicación de una metodología rigurosa es algo que ocurre frecuentemente y por la necesidad de poner orden a las versiones que se distribuyen a los usuarios, frente a cambios caóticos y frecuentes donde es casi imposible determinar cada nuevo bug en base a qué resolución de requerimiento anterior (o bug anterior) fue introducido.

Que las empresas opten en algún momento por permitir que sus desarrolladores dejen de aplicar rigurosamente la metodología es algo que ocurre frecuentemente y por la necesidad de no entorpecer la productividad de las versiones que se distribuyen a los usuarios, frente a necesidades de cambios donde es casi imposible determinar su ocurrencia dado los vaivenes de las necesidades de negocio.

El primer problema con el proceso de desarrollo de aplicaciones es lograr que sea conocido (comprendido) por el equipo de trabajo. Luego viene el problema siguiente: conseguir que sea aceptado.

A principios de los '90 Microsoft comenzó a recopilar a nivel interno las mejores prácticas en términos de procesos de desarrollo de software, no con la intención de establecer una metodología, sino con la idea de tener una colección de prácticas individuales de lo que funciona, aplicables dentro de determinados contextos.

Aquí no se buscó reinventar ninguna rueda sino que a esta colección iban a parar tanto prácticas hecho en Microsoft como prácticas populares de la industria, por ejemplo, la de formalizar como parte del proceso las pruebas unitarias.

Así es como surge Microsoft Solution Framework (MSF o, pronunciado, em-es-De manera tal que para cada proyecto se podrán seleccionar aquellas prácticas que realmente agreguen valor al proceso (de ahí el concepto de framework, por si algún despistado pensó que podría tratarse de un framework de software como lo es Spring, Struts o la Enterprise Library)

La próxima versión de Visual Studio Team System permite definir procesos MSF y crear proyectos de equipo que se enmarquen dentro de los mismos. Por defecto ya vienen dos procesos: uno ágil y el otro formal. Pero además pueden conseguirse adaptaciones de terceros o incluso adaptaciones hechas por uno mismo, ya que el nuevo Visual Studio tiene un manager de procesos.

Los roles intervinientes dentro del proceso MSF Ágil son mayormente 6: Analista de Negocio, Jefe de Proyecto, Arquitecto, Desarrollador, Tester y Administrador de Releases. No necesariamente esos roles deban ser mapeados a personas diferentes. Si bien ciertos roles pueden ser realizados por una misma persona, hay otras combinaciones en las que eso no es lo deseable.

De ese modo, es sencillísimo lograr adaptar un proceso existente a uno más acorde a nuestro contexto, y que la herramienta nos garantice su adecuación (como debe ser) para evitarnos el tener que cumplir con un procedimiento en forma voluntaria”

1.5.5. VERSIONES DE MSF

1.5.5.1. MSF 3.0

En <http://eproano334.blogspot.es/> establece que: “Para maximizar el éxito de los proyectos de una empresa. Microsoft ha puesto a disposición una guía para un efectivo diseño, desarrollo y funcionamiento de soluciones, incluyendo las tecnologías de Microsoft.

Este conocimiento se deriva de la experiencia ganada dentro de Microsoft con sus clientes y vendedores en proyectos de grande desarrollo de software y de prestación de servicios, la experiencia de los grandes consultores de Microsoft y el mejor conocimiento de la industria mundial de Tecnologías de Información.

MSF (Microsoft Solution Framework) provee un conjunto de modelos, principios y lineamientos para diseñar y desarrollar soluciones empresariales de manera que todos los elementos de un proyecto (como: la gente, procesos, y herramientas) puedan ser administrados apropiadamente. MSF también provee prácticas probadas para planear, diseñar, desarrollar e implementar soluciones empresariales exitosamente.

Este proceso es flexible y se puede adaptar al diseño y desarrollo de una amplia gama de proyectos de una empresa.

El modelo de proceso MSF combina los mejores principios del modelo en cascada y del modelo en espiral. Combina la claridad que planea el modelo en cascada y las ventajas de los puntos de transición del modelo en espiral.

Mediante un cronograma se puede especifica las fechas para puntos de control, revisiones por parte del cliente (hay un acta de finalización en donde el cliente firma después de cada fase para dar su aprobación y seguir con la siguiente fase), actividades y tiempo de desarrollo.

Sin embargo los precios de licencias, capacitación y soporte por parte de Microsoft son caros”.



Gráfico N°- 5

<http://eproano334.blogspot.es/>

En <http://scrucz334.blogspot.es/i2007-11/> establece que: “Ahora se va hablar de una metodología más sencilla, tanto de comprender como de seguir, esta metodología está enfocada a la administración a nivel proyecto, no se enfoca a la organización completa como tal, todos recordamos las clases de ingeniería de software y las fases del ciclo de vida clásico, Análisis, planeación, diseño, desarrollo, pruebas e implementación, esta metodología comprende 5 fases, Previsión, Planeación, Desarrollo, Estabilización e Implementación, es un proceso prácticamente con las mismas fases pero mucho más robusto, MSF te proporciona además una guía con los documentos que te ayudarán a dar un correcto seguimiento al proyecto.

Es común ver en proyectos que no se tienen roles definidos, ver personas haciendo de todo, y haciendo tareas que se supone que no se deberían de mezclar porque sería como ser juez y parte, esta metodología cuenta con roles que a su vez tienen actividades y responsabilidades definidas para cada uno.

- ❖ **Program Manager**, en esta versión el Program Manager tiene responsabilidades tanto administrativas como de Arquitecto de la solución, ya que debe de tener conocimiento tanto del negocio como del proyecto.
- ❖ **Product Manager**, es el que tiene contacto directo con el cliente y se centraliza en la satisfacción del mismo.
- ❖ **Desarrollo**, se refiere como tal a los desarrolladores.

- ❖ **Test**, personal que se designa para realizar las pruebas.
- ❖ **Release Manager**, es la persona que se encarga de estabilizar la aplicación y hacer la implementación.
- ❖ **User Experience**, se encarga de la funcionalidad.

MSF es escalable dependiendo del tamaño del proyecto, es por esto que los roles pueden ser combinados pero nada más hasta cierto punto, hay roles que es mejor no combinarlos:

	Product Management	Program Management	Development	Test	User Experience	Release Management
Product Management		N	N	P	P	U
Program Management			N	U	U	P
Development				N	N	N
Test					P	P
User Experience						U
Release Management						

- ❖ **P** Probable
- ❖ **U** Poco Probable
- ❖ **N** No Recomendado

La combinación de roles que aparecen como no recomendados es porque interfieren entre sí como un ejemplo está el desarrollo con el test, no es una buena práctica para el desarrollo de aplicaciones, que la persona que lo crea lo pruebe a esto me refiero en ser juez y parte”.

1.5.5.1.1. ESTRUCTURA DE MSF VERSIONES MSF 3.0 Y MSF 4.0



Gráfico N°- 6

<http://www.microsoft.com/MSF>

1.5.5.1.2. PRINCIPIOS BÁSICOS DE MSF 3.0

- ❖ Promover comunicaciones abiertas.
- ❖ Trabajar para una visión compartida.
- ❖ Fortalecer los miembros del equipo.
- ❖ Establecer responsabilidades claras y compartidas.
- ❖ Focalizarse en agregar valor al negocio.
- ❖ Permanecer ágil, y esperar los cambios.
- ❖ Invertir en calidad
- ❖ Aprender de todas las experiencias.

Son aspectos que se debe tener muy en cuenta para tener calidad en el modelo y en el producto.

- ❖ Comunicaciones abiertas.
- ❖ Visión compartida (un documento de visión en donde todos los miembros tengan un fin común).
- ❖ Fortalecer el equipo (Capacitación a los miembros, un aspecto que los otros modelos no lo hacen).
- ❖ Responsabilidades (establecer claramente las personales y las compartidas).
- ❖ Agregar valor. (Dar valor al cliente, al darle productos con funcionalidad, esto quiere decir que MSF es un proceso versionado).
- ❖ Ágil. (Algo muy raro en un proceso prescriptivo, pero de gran ayuda ya que es posible hacer cambios).
- ❖ Calidad. (Como esto cuesta trabajo, se lo ve como una inversión para llegar a la calidad).
- ❖ Aprender experiencias.

Lo que hace interesante a este modelo, son sus 3 disciplinas:

- ❖ Disciplina de administración de proyectos.
- ❖ Disciplina de administración de riesgos.
- ❖ Disciplina de administración de la preparación.

Las disciplinas son conceptos que Microsoft adaptó y adopto a MSF.

Disciplina de proyectos:

Un equipo para administrar los principios fundamentales del modelo.

Disciplina de riesgos:

Equipo para identificar prioridades, tomar decisiones y controlar emergencias.

Disciplina de la preparación:

Se diagnostica el nivel de conocimiento de los participantes.

1.5.5.2. MSF 4.0

En <http://eproano334.blogspot.es/> establece que: “Incorpora algunas modificaciones en relación con su anterior versión, sin embargo puede ser instanciado por una o más metodologías prescriptivas que reflejen las necesidades específicas de una organización.

Es un desarrollo full tecnología Microsoft, por lo tanto está orientado a la utilización de la propia tecnología de Microsoft.

Incorpora a su equipo de trabajo un rol más en relación a su anterior versión, que es, el Rol de Arquitectura. Hace un análisis completo sobre que roles pueden fusionarse, combinarse, es decir que roles pueden ser desarrollados por una sola persona.

Visual Studio 2005 Team System es la herramienta para administrar el proceso de desarrollo de software (MSF proporciona la guía de proceso y Team System un conjunto de herramientas integradas y extensibles).

Entre sus ventajas existe mayor vinculación con el cliente como también orientado al trabajo en equipo.

Siendo una metodología adaptable al proyecto, le permite ser más flexible que el RUP. Garantizando la probabilidad de éxito del proyecto mediante el análisis de riesgo.

Elegir una metodología no es una cuestión simple, en mi opinión es algo que depende principalmente de dos factores, el tipo de proyectos, las herramientas con las que se puede contar (RUP y MSF se implanta bien si se tienen las herramientas de Rational o Team System respectivamente) y la cultura que exista en la empresa”.

En <http://geekswithblogs.net/cyoung/articles/63354.aspx> establece que: “MSF ha sido objeto de continuo desarrollo y mejora de más de una década. La visión original de MSF es reunir y organizar un conjunto de orientaciones y buenas prácticas para el desarrollo de software de ciclo de vida extraídas de la experiencia de Microsoft, mejores prácticas de la industria y de los insumos.

Históricamente, Microsoft siempre ha sido muy cuidadoso para evitar MSF de inflexión en una metodología. Por ello, quiero decir que la adopción de MSF nunca ha encomendado la compra de costosas herramientas, plantillas, etc, o la adopción de altamente prescriptivo y detallada los procesos. En lugar de ello, MSF conscientemente aspira a proporcionar un mayor nivel de marco, o meta-modelo, de la orientación y los principios que puede ser asignada a una variedad de métodos discretos.

Esto incluye tanto los procesos, la industria y metodologías reconocidas como DSDM y RUP. A mediados del decenio de 1990 hubo varias razones para la adopción de este enfoque. Uno de ellos era el medio ambiente autónomo de los grupos de productos dentro de Microsoft.

MSF ha recorrido un largo camino desde sus primeras encarnaciones, y ha sido objeto de dos revisiones importantes. Equipo de Visual Studio coincide con sistema de una tercera revisión (versión 4.0), y esta versión representa una importante evolución del marco. Dos características generales son inmediatamente evidentes a este respecto.

Hay un mucho mayor énfasis en el uso de un conjunto de herramientas de apoyo (Visual Studio Team Services). MSF mantiene su papel como una meta-modelo, pero la llegada de equipo de Servicio de metodología para las plantillas de MSF significa que el marco probablemente ahora se percibe, erróneamente, como una metodología (o, más bien, dos metodologías asociadas) en su propio derecho.

1.5.5.2.1. MSF 4.0 para Agile Software Development

Métodos ágiles de desarrollo han tomado cada vez más central en los últimos años. MSF Cuando se publicó por primera vez, hay pocos verdaderamente metodologías ágiles, y "el desarrollo rápido de aplicaciones" fue considerada por muchos como un vacío término que se utiliza para justificar malas prácticas. MSF ha desempeñado un papel en la promoción de muchas de las centrales y más valioso de los principios y prácticas de desarrollo ágil, a pesar de que históricamente no ha utilizado ese término.

Las características centrales del ágil enfoque se definen por las declaraciones de los énfasis. Según el "Manifiesto de Agile Software Development", ágiles métodos de valor:

- ❖ Los individuos e interacciones sobre procesos y herramientas
- ❖ Trabajo de software más completa de la documentación
- ❖ Cliente colaboración en la negociación de los contratos
- ❖ En respuesta a cambios a lo largo siguiendo un plan

El manifiesto está directamente en Microsoft se hizo eco de la descripción de MSF Ágil figura dentro de la plantilla. Este texto es el siguiente:

“Elija el MSF de Agile Software Development proceso de los proyectos con ciclos de vida cortos y orientada a los resultados de equipos que pueden trabajar sin los lotes intermedios de la documentación. MSF Este proceso es un marco flexible de orientación que ayuda a crear un sistema adaptable para el desarrollo de software. Ágil metodología prevé la necesidad de adaptarse a los cambios, y se centra en las personas como el componente más importante para el éxito de un proyecto. Ágil metodología también se hace hincapié en la prestación de trabajo y promueve el software de validación del cliente como clave del éxito medidas”.

Metodologías ágiles defensor prácticas que incluyen la participación directa de los clientes y usuarios empresariales en el equipo de desarrollo, haciendo hincapié en la rápida y altamente iterativo código de prensa y de la prueba-impulsada por el desarrollo, la colaboración en las prácticas de trabajo y el fomento de la responsabilidad compartida y la propiedad. Ágiles de desarrollo de software espera y acoge con satisfacción el cambio y perfeccionamiento a la lista de requisitos, y gestiona a través de esta estrecha alianza con la empresa y por avanzar a través de pequeños y sencillos pasos.

MSF 4.0 describe cinco pistas para el proyecto de gobierno, cada uno de los cuales se asocia con un puesto de control. Estas corresponden en general a las cinco fases y etapas en el modelo antiguo. La lista de los cinco ciclos es independiente. Metodologías son libres de sacar sus propias asociaciones entre el gobierno y la secuencia.

Un beneficio inmediato de este nuevo enfoque es que es mucho más fácil de comunicar el concepto (siempre presente en MSF) que los grupos de actividades y flujos de trabajo pueden coincidir en mayor o menor grado. Históricamente, las RUP han hecho una mejor tarea de comunicación de este principio en el contexto de un proyecto a nivel de modelo de proceso.

1.5.5.2.2. El impacto de MSF 4.0

Toda organización pensando en la adopción de una o ambas de las nuevas metodologías de 4,0 MSF en primer lugar preguntarse algunas preguntas. Para muchas organizaciones, estas se incluyen la cuestión central de si una empresa puede ser el caso para la evolución de su proceso de desarrollo de software.

Una organización sólo es verdaderamente dispuesta a buscar CMMI TickIT certificación o cuando se compromete a la adopción de dicho sistema, independientemente de la consecución de la certificación.

Puede considerar que su proceso de madurez suficiente para satisfacer adecuadamente las exigencias impuestas a usted, y que la evolución no va a provocar en todo un valor añadido significativo. Por otra parte, usted puede creer que un proceso más maduro se traduciría en una mayor calidad, mayor eficiencia y reducción de los gastos generales de revisión, etc., o ayudar a diferenciar aún más que de su competencia.

En la adopción de Team Foundation Server, las organizaciones necesitan para decidir qué metodologías, en su caso, que desee adoptar, y la medida en que se desea personalizar una metodología existente. Como se ha descrito anteriormente, así como las dos plantillas de MSF de Microsoft, hay plantillas más en construcción para otras metodologías.

La adopción de la MSF enfoque tiene dos ventajas evidentes: a) hay que plantillas libre puede personalizar si lo desea y b) la naturaleza de los equipos de MSF permite elegir entre las dos plantillas, según proceda, para proyectos específicos”.

1.5.5.2.3. PRINCIPIOS BÁSICOS DE MSF 4.0

Una meta-modelo, es decir, un modelo de entre modelos solo para evitar confusiones.

Esta versión del modelo se caracteriza porque tiene fundamentos de modelo ágil.

Además agrega 2 principios más:

- Vinculación con los clientes. (Necesario para el feedback).
- Siempre crear productos entregables.

Según el equipo de trabajo que maneja MSF, no parece ser tan rígido, además sus roles a desempeñar no son rígidos, ya que pueden ser escalados dependiendo la magnitud del proyecto.

VENTAJAS MSF:

- ❖ Aplica mucho e incentiva al trabajo en equipo y a la colaboración.
- ❖ Es útil para proyectos de pequeña y gran escala.
- ❖ Crea una disciplina de análisis de riesgos que ayuda y evoluciona con el proyecto.
- ❖ Gracias a Microsoft, este modelo cuenta con plantillas que nos ayuda para el proceso de documentación.

DESVENTAJAS MSF:

- ❖ Por ser un modelo prescriptivo, solicita demasiada documentación en sus fases.
- ❖ El análisis de riesgos es necesario, pero si se lo hace muy exhaustivo puede demorar o hasta frenar el avance del proyecto.
- ❖ Al estar basado en tecnología Microsoft, trata de obligar a usar herramientas de ellos mismo, pero si es posible no usar esa tecnología pero lo que esto produce es más complejidad en el proyecto”

1.6. MODELOS DE CICLO DE VIDA

1.6.1. DEFINICIÓN

En http://rguerrero334.blogspot.es/img/Def.Modelo_de_Ciclo_de_Vida.pdf establece que: “Un modelo de ciclo de vida de software es una vista de las actividades que ocurren durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociadas entre estas etapas.

Un modelo de ciclo de vida del software:

- 1.- Describe las fases principales de desarrollo de software.
- 2.- Define las fases primarias esperadas de ser ejecutadas durante esas fases.
- 3.- Ayuda a administrar el progreso del desarrollo, y
- 4.- Provee un espacio de trabajo para la definición de un detallado proceso de desarrollo de software.

Así, los modelos por una parte suministran una guía para los ingenieros de software con el fin de ordenar las diversas actividades técnicas en el proyecto, por otra parte suministran un marco para la administración del desarrollo y el mantenimiento, en el sentido en que permiten estimar recursos, definir puntos de control intermedios, monitorear el avance, etc.”

En <http://es.kioskea.net/contents/genie-logiciel/cycle-de-vie.php3> establece que: “El ciclo de vida básico de un software consta de los siguientes procedimientos:

- ❖ **Definición de objetivos:** definir el resultado del proyecto y su papel en la estrategia global.
- ❖ **Análisis de los requisitos y su viabilidad:** recopilar, examinar y formular los requisitos del cliente y examinar cualquier restricción que se pueda aplicar.
- ❖ **Diseño general:** requisitos generales de la arquitectura de la aplicación.
- ❖ **Diseño en detalle:** definición precisa de cada subconjunto de la aplicación.
- ❖ **Programación (programación e implementación):** es la implementación de un lenguaje de programación para crear las funciones definidas durante la etapa de diseño.

- ❖ **Prueba de unidad:** prueba individual de cada subconjunto de la aplicación para garantizar que se implementaron de acuerdo con las especificaciones.
- ❖ **Integración:** para garantizar que los diferentes módulos se integren con la aplicación. Éste es el propósito de la prueba de integración que está cuidadosamente documentada.
- ❖ **Prueba beta (o validación):** para garantizar que el software cumple con las especificaciones originales.
- ❖ **Documentación:** sirve para documentar información necesaria para los usuarios del software y para desarrollos futuros.
- ❖ **Mantenimiento:** para todos los procedimientos correctivos (mantenimiento correctivo) y las actualizaciones secundarias del software (mantenimiento continuo).

El orden y la presencia de cada uno de estos procedimientos en el ciclo de vida de una aplicación dependen del tipo de modelo de ciclo de vida acordado entre el cliente y el equipo de desarrolladores.”

En Fiarley, Richard, Ingeniería de Software establece: “El modelo de las fases del ciclo de vida del producto de programación en una serie de actividades sucesivas; cada fase requiere información de entrada, procesos y resultados, todos ellos bien definidos”.

1.6.2. ELEMENTOS DEL CICLO DE VIDA

En <http://licethtorres.blogspot.com/2008/03/el-ciclo-de-vida-del-software.html> establece que: “Un ciclo de vida para un proyecto se compone de fases sucesivas compuestas por tareas panificables. Según el modelo de ciclo de vida, la sucesión de fases puede ampliarse con bucles de realimentación, de manera que lo que conceptualmente se considera una misma fase se pueda ejecutar más de una vez a lo largo de un proyecto, recibiendo en cada pasada de ejecución aportaciones de los resultados intermedios que se van produciendo (realimentación).

Para un adecuado control de la progresión de las fases de un proyecto se hace necesario especificar con suficiente precisión los resultados evaluables, o sea, productos intermedios que deben resultar de las tareas incluidas en cada fase. Normalmente estos productos marcan los hitos entre fases.

A continuación presentamos los distintos elementos que integran un ciclo de vida: Fases. Una fase es un conjunto de actividades relacionadas con un objetivo en el desarrollo del proyecto. Se construye agrupando tareas (actividades elementales) que pueden compartir un tramo determinado del tiempo de vida de un proyecto. La agrupación temporal de tareas impone requisitos temporales correspondientes a la asignación de recursos (humanos, financieros o materiales). Cuanto más grande y complejo sea un proyecto, mayor detalle se necesitará en la definición de las fases para que el contenido de cada una siga siendo manejable. De esta forma, cada fase de un proyecto puede considerarse un “micro-proyecto” en sí mismo, compuesto por un conjunto de micro-fases.

Otro motivo para descomponer una fase en sub-fases menores puede ser el interés de separar partes temporales del proyecto que se subcontraten a otras organizaciones, requiriendo distintos procesos de gestión.

Cada fase viene definida por un conjunto de elementos observables externamente, como son las actividades con las que se relaciona, los datos de entrada (resultados de la fase anterior, documentos o productos requeridos para la fase, experiencias de proyectos anteriores), los datos de salida (resultados a utilizar por la fase posterior, experiencia acumulada, pruebas o resultados efectuados) y la estructura interna de la fase”.

1.6.3. ALTERNATIVAS DE MODELOS DE CICLO DE VIDA

1.6.3.1. MODELOS DE CICLO DE VIDA LINEAL

En <http://licethtorres.blogspot.com/2008/03/el-ciclo-de-vida-del-software.html> se establece que “Es el más utilizado, siempre que es posible, precisamente por ser el más sencillo. Consiste en descomponer la actividad global del proyecto en fases que se suceden de manera lineal, es decir, cada una se realiza una sola vez, cada una se realiza tras la anterior y antes que la siguiente. Con un ciclo lineal es fácil dividir las tareas entre equipos sucesivos, y prever los tiempos (sumando los de cada fase).

Requiere que la actividad del proyecto pueda descomponerse de manera que una fase no necesite resultados de las siguientes (realimentación), aunque pueden admitirse ciertos supuestos de realimentación correctiva. Desde el punto de vista de la gestión (para decisiones de planificación), requiere también que se sepa bien de antemano lo que va a ocurrir en cada fase antes de empezarla.

Ejemplo de ciclo lineal para un proyecto de construcción”.

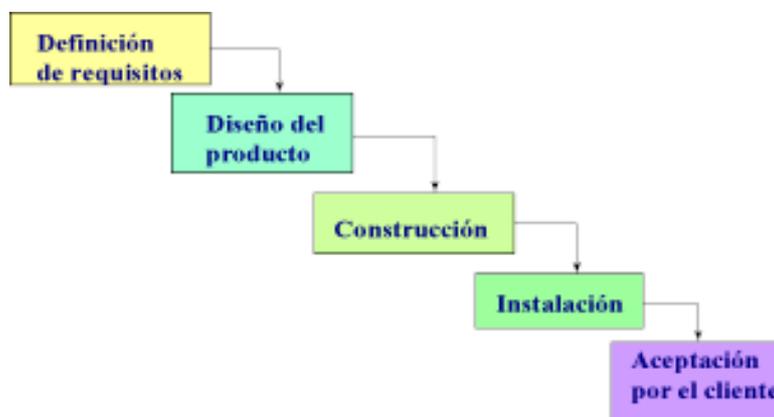


Gráfico N°- 7

<http://licethtorres.blogspot.com/2008/03/el-ciclo-de-vida-del-software.html>

1.6.3.2. MODELOS DE CICLO DE VIDA EN PROTOTIPO

Los prototipos son una visión preliminar del sistema futuro que se implantará.

La elaboración de prototipos de un sistema de información es una técnica valiosa para la recopilación rápida de información específica a cerca de los requerimientos de información de los usuarios.

Los prototipos efectivos deben hacerse tempranamente en el ciclo de vida del desarrollo de sistemas, durante la fase de determinación de requerimientos.

En esta forma el analista está buscando las reacciones iniciales de los usuarios y de la administración hacia el prototipo, sugerencias de los usuarios sobre cambios o limpieza del sistema para el que construye un prototipo, posibles innovaciones y planes de revisión que detallan que parte del sistema necesita realizarse primero.

Tipos de Información que busca el Analista durante la Elaboración de Prototipos.

- ❖ Reacciones del usuario.
- ❖ Innovaciones.
- ❖ Sugerencias del usuario.
- ❖ Plan de revisión.

Reacciones: Son recopiladas por medio de observaciones, entrevista y formas de retroalimentación, diseñadas para recoger la opinión de cada persona acerca del prototipo cuando interactúa con él.

Por medio de estas reacciones el analista descubre muchas perspectivas en el prototipo incluyendo el agrado que tenga el usuario al sistema.

Sugerencias: El analista también está interesado en la sugerencia de los usuarios y la administración acerca como refinar o cambiar el prototipo presentado. Las sugerencias son recolectadas de aquellos que experimenta con el prototipo, mediante un periodo de tiempo específico.

El tiempo que pasan los usuarios con el prototipo depende por lo general de su dedicación e interés en el proyecto de sistemas. Las sugerencias son el producto de la interacción de los usuarios con el prototipo. Estas sugerencias deben apuntar a la analista hacia formas de refinación, cambio o limpieza del prototipo para que se ajuste mejor a las necesidades de los usuarios.

Innovaciones: Son parte de las informaciones buscada por el equipo de análisis de sistema. Son capacidades nuevas del sistema que no habían sido pensadas antes de la interacción con el prototipo.

Van más allá de las características prototípicas actuales añadiendo algo nuevo e innovador.

Plan de Revisión: Ayuda a identificar prioridades para lo que se debe construir un prototipo a continuación. En situaciones donde están involucradas muchas ramas de la organización, los planes de revisión ayuda a determinar para cuáles hay que construir un prototipo a continuación.

La información recolectada en la fase de echadura del prototipo permite al analista asignar prioridades y redirigir los planes sin realizar gastos con un mínimo de ruptura. La elaboración de prototipo y la planeación van mano a mano.

TIPOS DE PROTOTIPO

En <http://rguerrero334.blogspot.es/1192897080/--modelo-de-prototipos--/> establece que: “

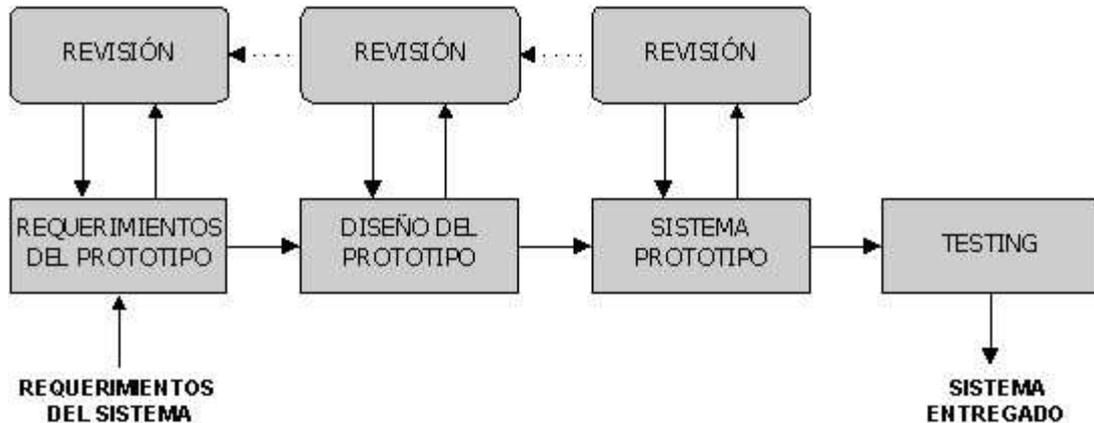


Gráfico N°- 8

<http://rguerrero334.blogspot.es/1192897080/--modelo-de-prototipos--/>

Este modelo consiste en un procedimiento que permite al equipo de desarrollo diseñar y analizar una aplicación que representa el sistema que sería implementado (McCracken y Jackson, 1982). Dicha aplicación, llamada prototipo, está compuesta por los componentes que se desean evaluar (Las funciones principales). Las etapas del modelo son:

- ❖ Investigación preliminar.
- ❖ Colecta y refinamiento de los requerimientos y proyecto rápido:
- ❖ Análisis y especificación del prototipo.
- ❖ Diseño y construcción del prototipo.
- ❖ Evaluación del prototipo por el cliente.
- ❖ Renacimiento del prototipo.

Para construir un prototipo del software se aplican los siguientes pasos:

PASO 1.- Evaluar la petición del software y determinar si el programa a desarrollar es un buen candidato para construir un prototipo.

Debido a que el cliente debe interactuar con el prototipo en los últimos pasos, es esencial que:

- ❖ El cliente participe en la evaluación y refinamiento del prototipo, y
- ❖ El cliente sea capaz de tomar decisiones de requerimientos de una forma oportuna. Finalmente, la naturaleza del proyecto de desarrollo tendrá una fuerte influencia en la eficacia del prototipo.

PASO 2.- Dado un proyecto candidato aceptable, el analista desarrolla una representación abreviada de los requerimientos.

Antes de que pueda comenzar la construcción de un prototipo, el analista debe representar los dominios funcionales y de información del programa y desarrollar un método razonable de partición. La aplicación de estos principios de análisis fundamentales, pueden realizarse mediante los métodos de análisis de requerimientos.

PASO 3.- Después de que se haya revisado la representación de los requerimientos, se crea un conjunto de especificaciones de diseño abreviadas para el prototipo.

El diseño debe ocurrir antes de que comience la construcción del prototipo. Sin embargo, el diseño de un prototipo se enfoca normalmente hacia la arquitectura a nivel superior y a los aspectos de diseño de datos, en vez de hacia el diseño procedimental detallado.

PASO 4.- El software del prototipo se crea, prueba y refina idealmente, los bloques de construcción de software preexistentes se utilizan para crear el prototipo de una forma rápida. Desafortunadamente, tales bloques construidos raramente existen.

Incluso si la implementación de un prototipo que funcione es impracticable, es escenario de construcción de prototipos puede aun aplicarse. Para las aplicaciones interactivas con el hombre, es posible frecuentemente crear un prototipo en papel que describa la interacción hombre-máquina usando una serie de hojas de historia.

PASO 5.- Una vez que el prototipo ha sido probado, se presenta al cliente, el cual "conduce la prueba" de la aplicación y sugiere modificaciones.

Este paso es el núcleo del método de construcción de prototipo. Es aquí donde el cliente puede examinar una representación implementada de los requerimientos del programa, sugerir modificaciones que harán al programa cumplir mejor las necesidades reales.

PASO 6.- Los pasos 4 y 5 se repiten iterativamente hasta que todos los requerimientos estén formalizados o hasta que el prototipo haya evolucionado hacia un sistema de producción.

El paradigma de construcción del prototipo puede ser conducido con uno o dos objetivos en mente:

- ❖ El propósito del prototipado es establecer un conjunto de requerimientos formales que pueden luego ser traducidos en la producción de programas mediante el uso de métodos y técnicas de ingeniería de programación
- ❖ El propósito de la construcción del prototipo es suministrar un continuo que pueda conducir al desarrollo evolutivo de la producción del software. Ambos métodos tienen sus meritos y ambos crean problemas”.

DESARROLLO DE UN PROTOTIPO

Cuando haya que decidir si hay que incluir la elaboración de prototipos como parte del ciclo de vida de desarrollo de sistemas, el analista necesita considerar cuál tipo de problema está siendo resuelto y en qué forma el sistema presenta la solución.

Lineamientos para el Desarrollo de un Prototipo.

1. Trabajar en módulos manejables.
2. Construir el prototipo rápidamente.
3. Modificar el prototipo en interacción sucesiva.
4. Enfatizar la interfaz del usuario.

Trabajar en Módulos Manejables: Es bueno que el analista en modelos manejables cuando se realiza el prototipo de algunas de las características de un sistema para obtener un modelo funcional.

Un modelo manejable es aquel que permite la interacción con sus características principales, pero todavía puede ser construido por separado de otros módulos del sistema. Las características del módulo que se consideran menos importantes son intencionalmente dejadas fuera del prototipo inicial.

Construcción Rápido del Prototipo: La velocidad es esencial para la elaboración satisfactoria de un prototipo en un sistema. El prototipo ayuda a acortar el tiempo de la interacción del sistema con el usuario para que pueda empezar a experimentar con él.

Se usan técnicas de recolección de información tradicional tales como: entrevistas, las observaciones e investigaciones de datos de archivo.

Modificaciones del Prototipo: Un tercer lineamiento para el desarrollo del prototipo es que debe ser flexible para futura modificaciones. Esto significa crearlo en módulos que no sean muy interdependientes.

Por lo general el prototipo es modificado varias veces pasando a través de varias interacciones. Los cambios al prototipo deben mover al sistema más cerca a lo que los usuarios dicen que es importante.

Cada modificación necesitan otras evaluaciones de los usuarios, estas modificaciones se deben realizar velozmente en uno o dos días, esto depende también del usuario y que tan rápido sea su evaluación.

Enfatizar la Interfaz de Usuarios: La interfaz del usuario con el prototipo (y eventualmente con el sistema) es muy importante debido que lo que se está tratando realmente de lograr con el prototipo es hacer que los usuarios muestren cada vez más sus requerimientos de información, debe ser capaz de interactuar fácilmente con el prototipo del sistema.

El objetivo del analista es diseñar una interfaz que permita al usuario interactuar con el sistema con un mínimo de entrenamiento y que permita el máximo de control del usuario sobre las funciones representadas.

1.6.3.3. MODELOS DE CICLO DE VIDA EN ESPIRAL

En http://es.wikipedia.org/wiki/Desarrollo_en_espiral establece que “El Desarrollo en Espiral es un modelo de ciclo de vida desarrollado por Barry Boehm en 1985, utilizado generalmente en la Ingeniería de software. Las actividades de este modelo se conforman en una espiral, cada bucle representa un conjunto de actividades. Las actividades no están fijadas a priori, sino que las siguientes se eligen en función del análisis de riesgo, comenzando por el bucle interior.

La Ingeniería del Software, se vale y establece de una serie de modelos que establecen y muestran las distintas etapas y estados por lo que pasa un producto software, desde su concepción inicial, pasando por su desarrollo, puesta en marcha y posterior mantenimiento, hasta la retirada del producto. A estos modelos se les denomina "Modelos de Ciclo de Vida del Software". El primer modelo concebido fue el de Royce, más comúnmente conocido como Cascada o "Lineal Secuencial". Este modelo establece que las diversas actividades que se van realizando al desarrollar un producto software se suceden de forma lineal.

Boehm, autor de diversos artículos de ingeniería del software; modelos de estimación de esfuerzo y tiempo que se consume en hacer productos software; y Modelos de Ciclo de Vida; ideó y promulgó un modelo desde un enfoque distinto al tradicional en Cascada: El Modelo Evolutivo Espiral. Su Modelo de Ciclo de Vida en Espiral tiene en cuenta fuertemente el riesgo que aparece a la hora de desarrollar software. Para ello, se comienza mirando las posibles alternativas de desarrollo, se opta por la de riesgo más asumible y se hace un ciclo de la espiral. Si el cliente quiere seguir haciendo mejoras en el software, se vuelve a evaluar las distintas nuevas alternativas y riesgos y se realiza otra vuelta de la espiral, así hasta que llegue un momento en el que el producto software desarrollado sea aceptado y no necesite seguir mejorándose con otro nuevo ciclo.

Este modelo fue propuesto por Boehm en 1988. Básicamente consiste en una serie de ciclos que se repiten en forma de espiral, comenzando desde el centro. Se suele interpretar como que dentro de cada ciclo de la espiral se sigue un Modelo Cascada, pero no necesariamente debe ser así. Aunque el Espiral puede verse como un modelo evolutivo que conjuga la naturaleza iterativa del modelo MCP con los aspectos controlados y sistemáticos del Modelo Cascada, con el agregado de gestión de riesgos.

En cada vuelta o iteración hay que tener en cuenta.

- ❖ **Los Objetivos:** Que necesidad debe cubrir el producto.
- ❖ **Las Alternativas:** Las diferentes formas de conseguir los objetivos de forma exitosa, desde diferentes puntos de vista como pueden ser:
 1. Características: experiencia del personal, requisitos a cumplir, etc.
 2. Formas de gestión del sistema.
 3. Riesgo asumido con cada alternativa.
- ❖ **Desarrollar y Verificar:** Programar y probar el software.

Si el resultado no es el adecuado o se necesita implementar mejoras o funcionalidades

Se planificarán los siguientes pasos y se comienza un nuevo ciclo del espiral. El espiral tiene una forma de caracola y se dice que mantiene dos dimensiones, la radial y la angular:

- ❖ **Angular:** Indica el avance del proyecto software dentro de un ciclo.
- ❖ **Radial:** Indica el aumento del coste del proyecto, ya que con cada nueva iteración se pasa más tiempo desarrollando.

Este sistema es muy utilizado en proyectos grandes y complejos como puede ser, por ejemplo, la creación de un Sistema Operativo.

Al ser un modelo de Ciclo de Vida orientado a la gestión de riesgo se dice que uno de los aspectos fundamentales de su éxito radica en que el equipo que lo aplique tenga la necesaria experiencia y habilidad para detectar y catalogar correctamente los riesgos.

Para cada ciclo habrá cuatro actividades:

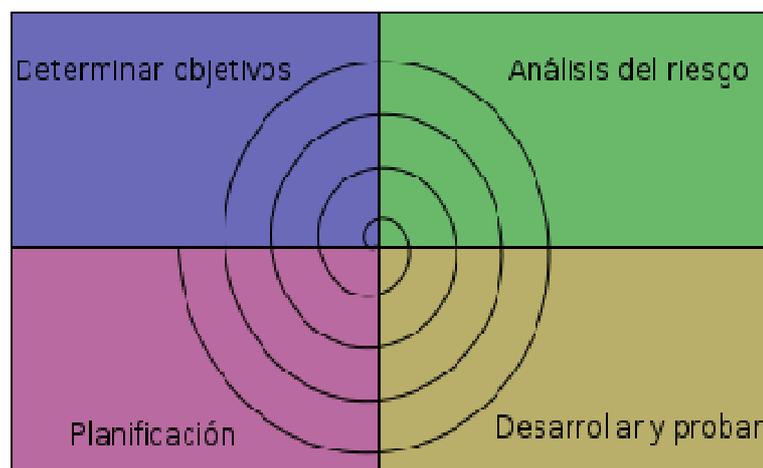


Gráfico N°- 9

http://es.wikipedia.org/wiki/Desarrollo_en_espiral

Determinar o fijar objetivos

- ❖ Fijar también los productos definidos a obtener: requerimientos, especificación, manual de usuario.
- ❖ Fijar las restricciones.
- ❖ Identificación de riesgos del proyecto y estrategias alternativas para evitarlos.
- ❖ Hay una cosa que solo se hace una vez: planificación inicial o previa.

Análisis del riesgo

- ❖ Se estudian todos los riesgos potenciales y se seleccionan una o varias alternativas propuestas para reducir o eliminar los riesgos.

Desarrollar, verificar y validar (probar)

- ❖ Tareas de la actividad propia y de prueba.
- ❖ Análisis de alternativas e identificación resolución de riesgos.
- ❖ Dependiendo del resultado de la evaluación de los riesgos.

Así si por ejemplo si los riesgos en la interfaz de usuario son dominantes, un modelo de desarrollo apropiado podría ser la construcción de prototipos evolutivos. Si lo riesgos de protección son la principal consideración, un desarrollo basado en transformaciones formales podría ser el más apropiado.

Planificar

- ❖ Revisamos todo lo hecho, evaluándolo, y con ello decidimos si continuamos con las fases siguientes y planificamos la próxima actividad”.

1.6.3.4. MODELOS DE CICLO DE VIDA EN CASCADA

En <http://www.google.com.ec/search?hl=es&q=ciclo+de+vida+en+cascada&start=20&sa=N> establece que “El más conocido, está basado en el ciclo convencional de una ingeniería, el paradigma del ciclo de vida abarca las siguientes actividades:

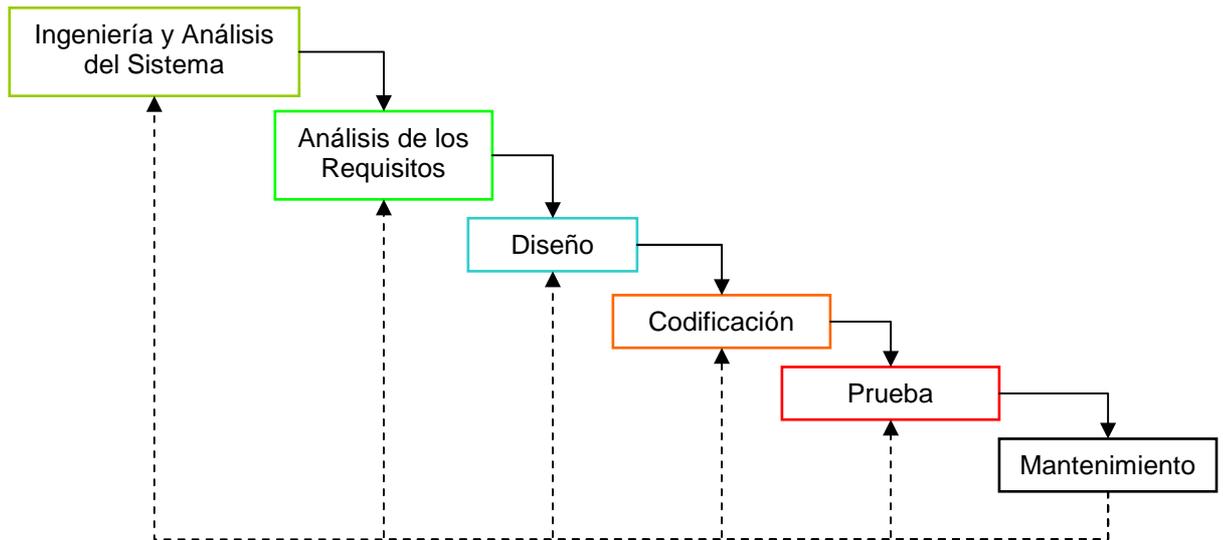


Gráfico N°- 10

<http://www.google.com.ec/search?hl=es&q=ciclo+de+vida+en+cascada&start=20&sa=N>

Ingeniería y Análisis del Sistema: Debido a que el software es siempre parte de un sistema mayor el trabajo comienza estableciendo los requisitos de todos los elementos del sistema y luego asignando algún subconjunto de estos requisitos al software.

Análisis de los requisitos del software: el proceso de recopilación de los requisitos se centra e intensifica especialmente en el software. El ingeniero de software (Analistas) debe comprender el ámbito de la información del software, así como la función, el rendimiento y las interfaces requeridas.

Diseño: el diseño del software se enfoca en cuatro atributos distintos del programa: la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación.

Codificación: el diseño debe traducirse en una forma legible para la maquina. El paso de codificación realiza esta tarea. Si el diseño se realiza de una manera detallada la codificación puede realizarse mecánicamente.

Prueba: una vez que se ha generado el código comienza la prueba del programa. La prueba se centra en la lógica interna del software, y en las funciones externas, realizando pruebas que aseguren que la entrada definida produce los resultados que realmente se requieren.

Mantenimiento: el software sufrirá cambios después de que se entrega al cliente. Los cambios ocurrirán debidos a que hayan encontrado errores, a que el software deba adaptarse a cambios del entorno externo (sistema operativo o dispositivos periféricos), o debido a que el cliente requiera ampliaciones funcionales o del rendimiento.

El Modelo en cascada tiende a ser muy relacionado con el Modelo de Cascada puesto que es una evolución del mismo.

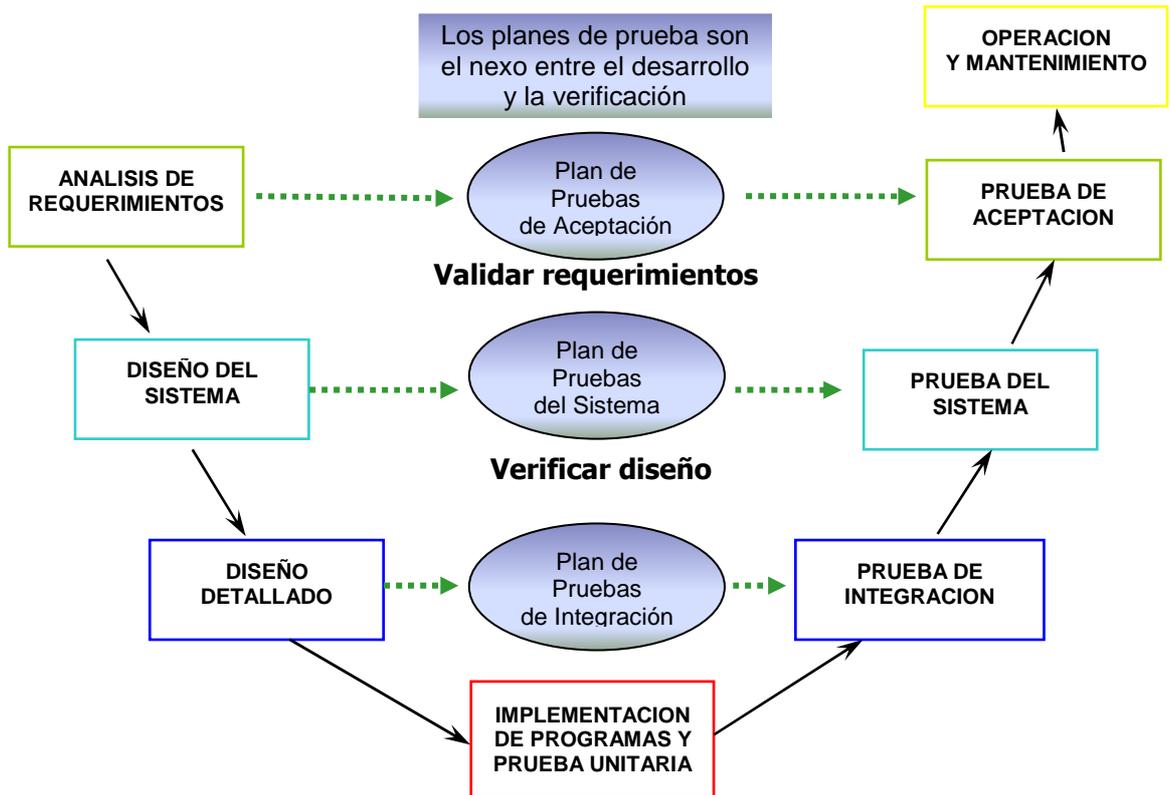


Gráfico N°- 11

<http://www.google.com.ec/search?hl=es&q=ciclo+de+vida+en+cascada&start=20&sa=N>

Puede notarse que su primera mitad es similar al Modelo en Cascada, y la otra mitad tiene como finalidad hacer pruebas e integración asociado a cada una de las etapas de la mitad anterior.

Se puede identificar una ventaja principal con respecto al Modelo Cascada más simple, y se refiere a que este modelo involucra chequeos de cada una de las etapas del modelo de cascada”.

1.6.3.5. MODELOS DE CICLO DE VIDA DE DESARROLLO INCREMENTAL

Los riesgos asociados con el desarrollo de sistemas largos y complejos son enormes. Una forma de reducir los riesgos es construir sólo una parte del sistema, reservando otros aspectos para niveles posteriores. El desarrollo incremental es el proceso de construcción siempre incrementando subconjuntos de requerimientos del sistema. Típicamente, un documento de requerimientos es escrito al capturar todos los requerimientos para el sistema completo.

El desarrollo incremental es 100% compatible con el modelo cascada. El desarrollo incremental no demanda una forma específica de observar el desarrollo de algún otro incremento.

Así, el modelo cascada puede ser usado para administrar cada esfuerzo de desarrollo.

El modelo de desarrollo incremental provee algunos beneficios significativos para los proyectos:

- ❖ Construir un sistema pequeño es siempre menos riesgoso que construir un sistema grande.
- ❖ Al ir desarrollando parte de las funcionalidades, es más fácil determinar si los requerimientos planeados para los niveles subsiguientes son correctos.
- ❖ Si un error importante es realizado, sólo la última iteración necesita ser descartada.
- ❖ Reduciendo el tiempo de desarrollo de un sistema (en este caso en incremento del sistema) decrecen las probabilidades que esos requerimientos de usuarios puedan cambiar durante el desarrollo.
- ❖ Si un error importante es realizado, el incremento previo puede ser usado.
- ❖ Los errores de desarrollo realizados en un incremento, pueden ser arreglados antes del comienzo del próximo incremento.

1.6.3.6. MODELOS DE CICLO DE VIDA CONCURRENTES

Como el modelo espiral, el modelo concurrente provee una meta-descripción del proceso software. Mientras que la contribución primaria del modelo espiral es en realidad que esas actividades del software ocurran repetidamente, la contribución del modelo concurrente es su capacidad de describir las múltiples actividades del software ocurriendo simultáneamente.

Esto no sorprende a nadie que ha estado involucrado con las diversas actividades que ocurren en algún tiempo del proceso de desarrollo de software. Discutamos un poco tales casos:

1.- Los requerimientos son usualmente "líneas de base", cuando una mayoría de los requerimientos comienzan a ser bien entendidos, en este tiempo se dedica un esfuerzo considerable al diseño. Sin embargo, una vez que comienza el diseño, cambios a los requerimientos son comunes y frecuentes (después de todo, los problemas reales cambian, y nuestro entendimiento de los problemas desarrollados también). Es desaconsejado detener el diseño en este camino cuando los requerimientos cambian; en su lugar, existe una necesidad de modificar y rehacer líneas de base de los requerimientos mientras progresa el diseño. Por supuesto, dependiendo del impacto de los cambios de los requerimientos el diseño puede no ser afectado, medianamente afectado o se requerirá comenzar todo de nuevo.

2.- Durante el diseño de arquitectura, es posible que algunos componentes comiencen a ser bien definidos antes que la arquitectura completa sea estabilizada. En tales casos, puede ser posible comenzar el diseño detallado en esos componentes estables. Similarmente, durante el diseño detallado, puede ser posible proceder con la codificación y quizás regular testeando en forma unitaria o realizando testeos de integración previo a llevar a cabo el diseño detallado de todos los componentes.

3.- En algunos proyectos, múltiples etapas de un producto se han desarrollado concurrentemente. Por ejemplo, no es inusual estar haciendo mantenimiento de la etapa 1 de un producto, y al mismo tiempo estar haciendo mantenimiento sobre un componente 2, mientras que se está haciendo codificación sobre un componente 3, mientras se realiza diseño sobre una etapa 4, y especificación de requisitos sobre un componente.

4.- En todos estos casos, diversas actividades están ocurriendo simultáneamente.

Eligiendo seguir un proyecto usando técnicas de modelación concurrente, se posibilita el conocimiento del estado verdadero en el que se encuentra el proyecto.

1.6.3.7. MODELOS DE CICLO DE VIDA DE DESARROLLO EVOLUTIVO

Como el modelo de desarrollo incremental, el modelo de desarrollo evolutivo (algunas veces denominado como prototipado evolutivo) construye una serie de grandes versiones sucesivas de un producto. Sin embargo, mientras que la aproximación incremental presupone que el conjunto completo de requerimientos es conocido al comenzar, el modelo evolutivo asume que los requerimientos no son completamente conocidos al inicio del proyecto.

En el modelo evolutivo, los requerimientos son cuidadosamente examinados, y sólo esos que son bien comprendidos son seleccionados para el primer incremento. Los desarrolladores construyen una implementación parcial del sistema que recibe sólo estos requerimientos.

El sistema es entonces desarrollado, los usuarios lo usan, y proveen retroalimentación a los desarrolladores. Basada en esta retroalimentación, la especificación de requerimientos es actualizada, y una segunda versión del producto es desarrollada y desplegada. El proceso se repite indefinidamente.

El desarrollo evolutivo es 100% compatible con el modelo cascada. El desarrollo evolutivo no demanda una forma específica de observar el desarrollo de algún incremento.

Así, el modelo cascada puede ser usado para administrar cada esfuerzo de desarrollo. Obviamente, el desarrollo incremental y evolutivo puede ser combinado también. Todo lo que uno tiene que hacer es construir un subconjunto de requerimientos conocidos (incremental), y comprender al principio que muchos nuevos requerimientos es probable que aparezcan cuando el sistema sea desplegado o desarrollado.

El desarrollo de software en forma evolutiva requiere un especial cuidado en la manipulación de documentos, programas, datos de test, etc. desarrollados para distintas versiones del software.

1.7. MÉTODO DE DESARROLLO DE SOFTWARE

1.7.1. DEFINICIÓN DEL MÉTODO DE DESARROLLO DE SOFTWARE

En http://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software establece que: “Es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.”

1.7.2. DEFINICIÓN DEL PROBLEMA

En <http://www.angelfire.com/scifi/jzavalar/apuntes/metodods.html> establece que: “En esta etapa, también conocida como Especificación de Requerimientos, se establece el problema, aclarándolo lo más posible. Es la parte más crítica de la solución. Amerita un estudio cuidadoso. Se deben identificar las teorías, fundamentos y/o principios matemáticos, físicos o de cualquier índole que permitan fundamentar satisfactoriamente el problema.

- ❖ Se deben eliminar los aspectos poco importantes para el planteamiento del problema
- ❖ Si el problema no está completamente definido se deben allegar la información adicional”.

1.7.3. ANÁLISIS

En <http://www.angelfire.com/scifi/jzavalar/apuntes/metodods.html> establece que: “En esta etapa se deben identificar las entradas del problema, los resultados deseados o salidas y cualquier requerimiento o restricción adicional en la solución

- ❖ Identificar qué información se proporciona (datos del problema)
- ❖ Identificar qué resultados deben calcularse y/o desplegarse
- ❖ Determinar la forma y las unidades en qué se deben desplegar los resultados
- ❖ Acotar las teorías, fundamentos y/o principios necesarios haciendo los supuestos y simplificaciones necesarias
- ❖ Identificar los tipos y estructuras de datos necesarios para los datos del problema y para los resultados
- ❖ Identificar las funciones u operaciones necesarias para cubrir los requerimientos del problema”.

1.7.4. DISEÑO

En <http://www.angelfire.com/scifi/jzavalar/apuntes/metodods.html> establece que: “El diseño consiste básicamente en desarrollar una lista de pasos llamados algoritmo o receta de la solución, verificando que el problema se resuelve como se desea.

- ❖ Es la parte más difícil del proceso de solución del problema
- ❖ Debe verificarse que es correcto el algoritmo antes de continuar
- ❖ Se auxilia de técnicas de diseño como pseudocódigo y diagramas de flujo.”

1.7.5. IMPLEMENTACIÓN

En <http://www.angelfire.com/scifi/jzavalar/apuntes/metodods.html> establece que: “Esta etapa consiste en implementar o escribir el algoritmo como un programa de computadora en un lenguaje de programación, convirtiendo cada paso del algoritmo en instrucciones en el lenguaje de programación.

- ❖ Se requiere el conocimiento de un lenguaje de programación particular en lo referente a su gramática, sintaxis y semántica, para ello se recomienda leer el manual del programador o su equivalente y utilizarlo como consulta siempre que sea necesario.

Una manera de iniciar el conocimiento del lenguaje de programación es interpretando programas ejemplo, ejecutarlos, observar los resultados y analizar las entradas, las salidas y los procesos de cálculo o flujo de información mediante instrucciones de salida que finalmente”.

- ❖ Se requiere mínimo de las siguientes herramientas:
 - Un editor de texto para escribir el código fuente como un archivo de tipo texto plano (por ejemplo notepad para guardar los archivos como html)
 - Un intérprete que procese el código fuente y lo ejecute (por ejemplo el browser que ejecuta scripts en javaScript al cargar la página web)
 - Un debugger que nos ayude a depurar los errores y a corregir el código fuente hasta lograr un programa ejecutable sin errores (por ejemplo el mismo browser que envía mensajes a encontrar errores al ejecutar nuestro programa)
- ❖ Se deben utilizar los tipos y estructuras de datos más adecuados que permita el lenguaje de programación, teniendo especial cuidado en el uso de tipos de datos reales y los errores de redondeo que introducen y pueden alterar los resultados.”

1.7.6. DOCUMENTACIÓN

En <http://www.deambulando.com/2007/04/24/pasos-para-el-desarrollo-de-software/> establece que: “Realización del manual de usuario, y posiblemente un manual técnico con el propósito de mantenimiento futuro y ampliaciones al sistema.”

1.7.7. VERIFICACIÓN Y PRUEBA

En <http://www.angelfire.com/scifi/jzavalar/apuntes/metodods.html> establece que: “Esta etapa consiste en probar el programa completo y verificar que trabaja como se esperaba

- ❖ Se deben probar cada una de las funciones primero por separado y luego en conjunto
- ❖ Se debe probar el programa completo con distintos conjuntos de datos de prueba
- ❖ En caso de que haya errores repetir el paso 4 y 5 hasta la satisfacción de los requerimientos.”

CAPITULO II

2. PRESENTACIÓN, ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

2.1. CARACTERIZACIÓN DE LA COMPAÑÍA DE TRANSPORTE PÚBLICO URBANO POPULAR DE PASAJEROS “LATACUNGA S.A. CITULASA”

2.1.1. ANÁLISIS DE DATOS ORGANIZACIONALES DE LA COMPAÑÍA DE TRANSPORTE PÚBLICO URBANO POPULAR DE PASAJEROS “LATACUNGA S.A. CITULASA”.

La compañía de Transporte Público Urbano Popular de Pasajeros “LATACUNGA S.A. CITULASA” es una organización jurídica domiciliada en el Cantón Latacunga Provincia de Cotopaxi, mediante Resolución N° 94-1-1-0056, emitida por la Superintendencia de Compañías, e inscrita en el Registro Mercantil con el número 03, de fecha 14 de Enero de 1.994.

Otorgada el permiso de Operación por el Consejo Nacional de Tránsito a la Institución para que preste el servicio de Transporte Urbano a la Ciudad de Latacunga a personas naturales y/o jurídicas, públicas y/o privadas, con vehículos acorde al confort y comodidad del usuario.

Por tal motivo en uso de las atribuciones legales, y de lo expuesto en los informes que anteceden resuelve el Consejo Nacional de Tránsito y Transporte Terrestres lo siguiente:

1.- Autorizar a la Compañía de Transportes CITULASA domiciliada en la Ciudad de Latacunga, Provincia del Cotopaxi, el cambio de su modalidad de servicio, esto es de EJECUTIVO a POPULAR.

2.- La vigencia del permiso de Operación concedido a esta compañía, es la fijada en la Resolución N° 001-cpo-005-CNT-95.: esto es de cinco años. Contados a partir del 11 de Julio de 1.995.

3.- Comunicar con esta resolución a la Compañía de Transporte Urbano de Pasajeros, “CITULASA”, y a los Organismos de Tránsito y Transporte Terrestres competentes, para sus registro, cumplimiento y control.

2.1.2. ORGANIGRAMA ESTRUCTURAL DEL ÁREA ADMINISTRATIVA DE LA COMPAÑÍA DE TRANSPORTE PÚBLICO URBANO POPULAR DE PASAJEROS “LATACUNGA S.A CITULASA”.

En el ANEXO N° 2 se puede observar el Organigrama Estructural y conjuntamente los diferentes nombres de los administrativos quienes conforman dichas dignidades. De esta manera se podrá establecer el factor importante de los roles que cumplen las diferentes designaciones en el campo administrativo.

De manera general se puede decir que, desempeñan los factores más importantes dentro de la Compañía de Transporte Urbano “CITULASA”. Por tal motivo:

En uso de las atribuciones que le otorga el Art. 14, incisos primero y segundo del Reglamento de Procedimientos Administrativos de los Consejos Nacional y Provinciales de Tránsito autoriza los siguientes aspectos:

1.- Renovar el permiso de operación de la Compañía de Transporte Público Urbano Popular de Pasajeros en Buses “LATACUNGA S.A. CITULASA”, domiciliada en el Cantón Latacunga, Provincia de Cotopaxi.

2.- Establecer a la Compañía Rutas Urbanas, con los diferentes recorridos para el desenvolvimiento óptimo y eficaz en el trabajo diario que la Institución presta a la colectividad de la Provincia de Cotopaxi; es así que a continuación se detalla las diferentes líneas de servicio con las que cuenta esta Organización:

- ❖ Línea N° 1 : Santa Bárbara – El Triángulo Sur (Gasolinera)
- ❖ Línea N° 3 : La Calera – Bethlemitas
- ❖ Línea N° 4 : Rumipamba – La Cocha – Isimbo
- ❖ Línea N° 5 : Zumbalica – Sigsicalle Sur – Cruz Loma

3.- El Permiso de Operación concedido a la Compañía de Transporte Público Urbano Popular de Pasajeros “LATACUNGA S.A. CITULASA”, beneficia a los socios debidamente calificados por el Organismo competente, y con sus diferentes vehículos.

4.- El tiempo de Duración del Permiso de Operación, concedido a favor de la Compañía de Transporte Urbano CITULASA, tendrá una vigencia de cinco años.

5.- Los vehículos que se utilizarán para el servicio autorizado, deberán reunir las condiciones de seguridad y confort.

6.- El aumento o disminución de los socios, el cambio de unidades, no podrá realizar la Compañía sin Resolución del Consejo Provincial de Tránsito de Cotopaxi.

7.- Las tarifas que regirán para la operación de este servicio serán fijadas por el Consejo Nacional de Tránsito y Transporte Terrestres; y,

8.- La Compañía se someterá a las normas establecidas en la Ley y Reglamentos de Tránsito y a las resoluciones que dictaren el Consejo Nacional de Tránsito y la Respectiva Jefatura Provincial de Tránsito.

2.1.3. RECOPIACIÓN, IDENTIFICACIÓN Y ANÁLISIS DE PROCESOS MANUALES QUE REALIZA LA COMPAÑÍA DE TRANSPORTE PÚBLICO URBANO POPULAR DE PASAJEROS “LATACUNGA S.A CITULASA”.

2.1.3.1. ANTECEDENTES

La Compañía de Transportes Urbano “Latacunga S.A.” CITULASA, se constituyó con su domicilio en la ciudad de Latacunga, con un capital suscrito de seis millones cuatrocientos mil sucres y pagado a la fecha, la suma de cinco millones novecientos mil sucres, dividido en seis mil cuatrocientos acciones de un mil sucres cada una, mediante escritura pública otorgada ante el Notario Décimo Octavo de la Ciudad de Quito, el cuatro de Enero de mil novecientos noventa y cuatro, aprobado por la Superintendencia de Compañías, mediante Resolución N° 94-1-1-10056, e inscrita en el Registro mercantil del Cantón Latacunga.

La compañía se caracteriza por brindar servicios de Transporte Urbano a las personas en la ciudad de Latacunga, Provincia de Cotopaxi de acuerdo a las autorizaciones debidas de las competentes autoridades de tránsito. Para el cumplimiento de su actividad social la compañía celebrará todo tipo de contrato permitido por la ley que sean afines a su actividad social.

2.1.3.2. JUNTA GENERAL

La junta General de accionistas es el órgano supremo de la Compañía y se halla integrada por los accionistas legalmente convocados y reunidos. Será ordinaria si se reúnen dentro de los tres primeros meses de cada ejercicio económico para conocer y resolver el informe del Gerente General, balance general y la cuenta de resultados, así como los demás asuntos puntualizados en la ley y en la convocatoria.

Las juntas universales se sujetarán a lo dispuesto en el artículo doscientos ochenta de la ley de Compañías. Y son, extraordinarias cuando se las convoque conforme a la ley, para conocer los puntos citados en la respectiva convocatoria.

Las juntas generales se reunirán en el domicilio principal de la compañía sus atribuciones a más de las señaladas, son designar cada dos años al Presidente Ejecutivo, Gerente General, siete miembros principales y sus respectivos suplentes del directorio, así como removerlos por causas legales y fijar sus remuneraciones; nombrar un Comisario Principal y un suplente, un liquidador principal y su respectivo suplente, y las demás fijadas en la ley de las Compañías.

2.1.3.3. DEL DIRECTORIO

Estará integrado por siete vocales, tienen que ser accionistas de la Compañía, pueden ser reelectos en su cargo, además existirán sus respectivos suplentes.

➤ DEBERES Y ATRIBUCIONES

- ❖ Velar por la buena marcha de la compañía y el cumplimiento de las disposiciones legales, así como de los estatutos sociales, y las resoluciones de la Junta General.
- ❖ Resolver los asuntos que se someten a su consideración el Presidente y/o el gerente general.
- ❖ Autorizar al Presidente del Directorio el obligar a la compañía en la celebración de contratos que estén entre los cincuenta y ciento cincuenta salarios mínimos vitales generales, para cuantías superiores se necesitará autorización de la Junta General.
- ❖ Supervigilar la Administración de la Compañía.

2.1.3.4. SESIONES Y RESOLUCIONES DEL DIRECTORIO

Las sesiones de directorio estarán presididas por el Presidente que será elegido entre sus miembros, actuará como secretario el Gerente General de la Compañía, quien tendrá solo voz afirmativa; en caso de ausencia de los dos funcionarios citados se procederá a nombrar de entre los presentes los reemplazos para el efecto.

2.1.3.5. UTILIDADES

Sólo se pagarán utilidades, una vez declarados los dividendos, por beneficios realmente obtenidos y percibidos o de reservas expresas, efectivas de libre disposición. Estas se pagarán a los accionistas en proporción al valor pagado de las acciones.

2.1.3.6. DISOLUCIONES, SANCIONES, REMISIÓN

La junta general tendrá las más amplias facultades para resolver la disolución de la compañía y proceder a la designación de liquidadores. En este punto, así como en lo referente a las sanciones diversas, y a lo no previsto expresamente en este Estatuto, los accionistas se remiten a lo señalado en la Ley de Compañías y sus eventuales reformas.

Los socios de la Organización han tomado interés en las actividades que han venido desarrollando con el fin de aportar al engrandecimiento de la colectividad de la Ciudad de Latacunga y en especial a la Compañía, en este sentido el objetivo principal de un **SISTEMA PROTOTIPO DE ADMINISTRACIÓN** es apoyar al control de información de cada socio activo y pasivo y en la utilización de los diferentes turnos que son asignados a las unidades que prestan el servicio a la colectividad de una manera clara, precisa y fácil.

Se han aplicado diferentes Técnicas para la recopilación de resultados como entrevistas las mismas que han sido aplicadas directamente a los miembros del Área Administrativa quienes llevan los procesos principales de la Institución; esto con el fin de hacerles conocer la realidad de nuestra sociedad.

El Consejo Administrativo cumple diferentes obligaciones, funciones y atribuciones en sus cargos asignados. Sus funciones son las siguientes:

2.1.3.7. PRESIDENTE

El presidente de la Compañía obligatoriamente deberá ser socio Activo de la Institución, y además deberá tener una unidad en servicio, y será elegido del seno de directorio y durará dos años en sus funciones pudiendo ser reelegido en indefinida.

➤ SON FUNCIONES Y ATRIBUCIONES DEL PRESIDENTE

- ❖ Representar legal, judicial y extrajudicialmente a la Compañía conjuntamente con el gerente general.
- ❖ Mantener con cautela y energía el orden y disciplina, durante el desarrollo de las asambleas y de las sesiones.
- ❖ Controlar periódicamente los lugares de trabajo y las oficinas que tuviere la Compañía.
- ❖ Especial función del Presidente será el mantener relaciones óptimas, con los diferentes organismos que formen parte de la Compañía.
- ❖ Adoptar las medidas permanentes y necesarias para mantener el control de las unidades, que presten los servicios en la Compañía, a fin de garantizar un eficiente servicio a la colectividad.
- ❖ Proceder a sancionar a los conductores y ayudantes de las unidades que se encuentran en mal estado de funcionamiento interno como externo.
- ❖ Suspender a las unidades cuando los socios o conductores en forma desleal obstruyan el trabajo planificado en las rutas.

- ❖ Suspender a los socios y ayudantes por encontrarse en estado étlico, conduciendo las unidades.
- ❖ Presidir las sesiones de Junta General de accionistas y del Directorio de la Institución.

2.1.3.8. GERENTE

El gerente será elegido por la Junta General; deberá reunir los mismos requisitos de los vocales, durará dos años en sus funciones.

➤ SON FUNCIONES Y ATRIBUCIONES DEL GERENTE

- ❖ Representar, judicial y extrajudicialmente a la Compañía, con el Presidente de la misma.
- ❖ Suscribir los contratos conjuntamente con el Presidente y por los montos señalados y autorizados al Presidente.
- ❖ Organizar el personal de la Compañía de acuerdo al organigrama aprobado por el directorio.
- ❖ Llevar por sí mismo o por intermedio de otra persona, pero bajo su estricta responsabilidad, los libros de acciones y accionistas, el libro de actas a la junta General.
- ❖ Supervisar los libros de contabilidad y responder por la conservación de todos los documentos, libros, enseres, y activos en general de la Compañía.
- ❖ Presentar la proforma presupuestaria anual al Directorio y hasta el 10 de Noviembre de cada año, para su estudio y aprobación.
- ❖ Informar trimestralmente al Directorio en forma escrita sobre la marcha de los balances financieros, el estado de pérdidas y ganancias.
- ❖ Concurrir obligatoriamente a las sesiones de la Junta General y del Directorio.
- ❖ Velar porque los dineros de la Compañía sean depositados en forma inmediata en las cuentas bancarias que mantenga la Institución.

- ❖ Legalizar conjuntamente con el Presidente todos los documentos económicos de la Institución.

2.1.3.9. SECRETARIA Y CONTADORA

En la dependencia de Secretaría se realiza trámites de documentos sean estos externos o internos de la Institución con el fin de dar respuesta y soluciones a las problemáticas de los socios o de los usuarios.

En la dependencia de Contabilidad se lleva a cabo toda la contabilidad de la Compañía con sus diferentes comprobantes de ingresos y egresos de la Institución, los incrementos que ha tenido hasta la actualidad desde sus orígenes.

➤ DOCUMENTOS EXTERNOS

Los documentos externos son aquellos documentos o archivos con los cuales cuenta la institución como pueden ser:

- ❖ Solicitudes de personas que no tienen vinculación con la compañía.
- ❖ Invitaciones.
- ❖ Pedidos.
- ❖ Contrato de Unidades.
- ❖ Documentos Contables.

➤ DOCUMENTOS INTERNOS

Los documentos internos son archivos que posan desde la creación de la compañía hasta la actualidad como pueden ser:

- ❖ Actas.
- ❖ Solicitudes de socios.
- ❖ Documentos contables.
- ❖ Reglamento Interno.

2.1.3.10. COMISARIO

El comisario cumple un rol importante dentro de la institución ya que es la persona encargada de llevar todos los gastos, cuentas y en especial la administración de recursos del Organismo para así poder cumplir con los objetivos planteados por la compañía, ajustándose a lo dictado por la Ley.

2.1.4. SITUACIÓN ACTUAL DE LA ESTRUCTURA INFORMÁTICA CON LA QUE CUENTA LA COMPAÑÍA DE TRANSPORTE PÚBLICO URBANO POPULAR DE PASAJEROS “LATACUNGA S.A CITULASA”

La Compañía de Transporte Público Urbano Popular de Pasajeros “LATACUNGA S.A CITULASA” siendo el pionero principal es una entidad con calidad que brinda diferentes alternativas con sus unidades a los usuarios de la colectividad de la Provincia de Cotopaxi, a la vez impulsa profundos cambios que existe diariamente en el caminar de la carretera de esta manera enfrentar los diferentes cambios existentes que se producen por los retos que a menudo sucede entre chofer y usuario.

En la actualidad la Compañía de Transporte Público Urbano Popular de Pasajeros “LATACUNGA S.A CITULASA” cuenta con cinco oficinas las mismas que son distribuidas para los diferentes administrativos como son:

- ❖ **PRESIDENTE** (Dr. Roosevelt Icaza Endara)
- ❖ **GERENTE** (Sr. Fausto Salazar Corrales)
- ❖ **SECRETARIA Y CONTADORA** (Prisilla Manotoa Albán)
- ❖ **COMISARIO** (Ing. Wladimiro Herrera Toaquiza)

El principal problema de la infraestructura informática de la Compañía se debe a la falta de una red, la misma que es de mucha utilidad para la conexión entre las diferentes dependencias como son desde la oficina del **PRESIDENTE** a la secretaria, al gerente, y al comisario quienes cumplen funciones importantes.

Dentro de la organización el principal objetivo es salvaguardar la información confidencial y evitar problemas que perjudicarían la integridad de la Compañía.

El Presidente de la Compañía ha manifestado la mala infraestructura informática con la que detallan ya que actualmente cuentan con dos computadores y son las siguientes:

OFICINA (PRESIDENTE)

- ❖ 1 Monitor
- ❖ 1 CPU
- ❖ 1 teclado
- ❖ 1 mouse
- ❖ Parlantes.

Un computador con tecnología CELEROM PENTIUM 4.

OFICINA (SECRETARIA y CONTADORA)

- ❖ 1 Monitor
- ❖ 1 CPU
- ❖ 1 Teclado
- ❖ 1 Mouse
- ❖ Parlantes.

Un computador con tecnología MILENIUM.

2.2. ANÁLISIS DE RESULTADOS DE LA ENCUESTA APLICADA A LOS SOCIOS DE LA COMPAÑÍA DE TRANSPORTE PÚBLICO URBANO POPULAR DE PASAJEROS “LATACUNGA S.A CITULASA”

2.2.1. MÉTODOS DE INVESTIGACIÓN

La investigación del presente proyecto de Tesis se inicia desde el método científico por su procedimiento para llevar a cabalidad la investigación.

De esta manera descubrir la verdad hasta enriquecer la ciencia, la misma que nos llevará a llenar vacíos del conocimiento y descartar errores.

Este método ha ido evolucionando con el pasar de los días de acuerdo a las necesidades que se requiere dependiendo de la época; de esta manera se asume que de aquí parte la cúspide de todos los métodos teóricos, pues son un pilar fundamental para llegar a la verdad con hechos reales y precisos.

De igual manera se aplicado el Método Descriptivo donde se encarga de describir una realidad concreta en su totalidad y adquirir un dominio cognoscitivo, en tal sentido se desarrollará un proceso completo, partiendo de la definición del problema hasta llegar a las Conclusiones y soluciones prácticas de valor y trascendencia científica y social.

Los datos Estadísticos se obtuvieron aplicando la Técnica de la ENCUESTA realizada a socios activos de la Compañía de Transporte Público Urbano Popular de Pasajeros “LATACUNGA S.A. CITULASA”, la misma que ayudará a elaborar los diferentes cuadros, gráficos estadísticos y sus diferentes análisis.

Esta información se recopiló con formularios que fueron llenados por los encuestados (socios activos), de esta manera se podrá llegar a una conclusión con una visión clara y precisa. **VER ANEXO N° 3.**

2.2.1.1. ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS DE LAS ENCUESTAS APLICADAS A LOS SOCIOS DE LA COMPAÑÍA DE TRANSPORTE PÚBLICO URBANO POPULAR DE PASAJEROS “LATACUNGA S.A CITULASA”

PREGUNTA N° 1

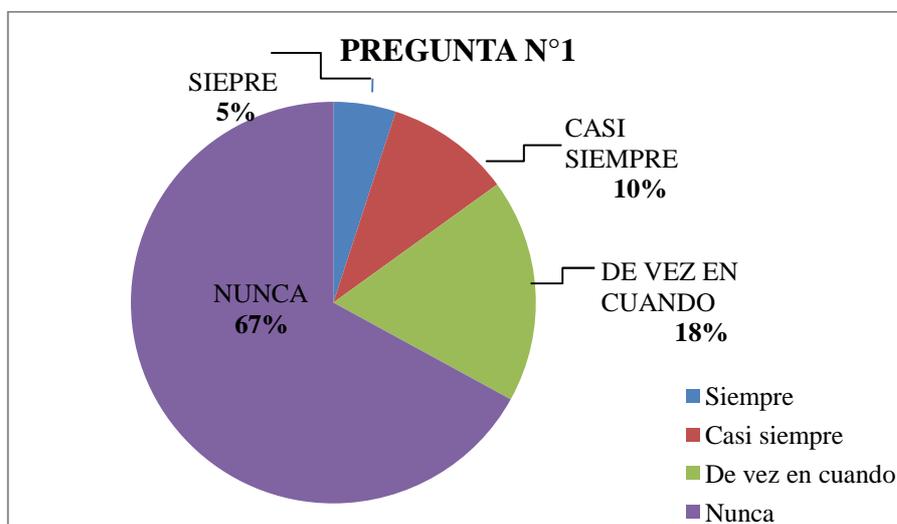
¿Usted como socio recurre a la Información de Archivos de la Compañía de Transporte Público Urbano Popular de Pasajeros “Latacunga S.A. CITULASA”?

TABLA N° 1

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SIEMPRE	2	5
CASI SIEMPRE	4	10
DE VEZ EN CUANDO	7	18
NUNCA	26	67
TOTAL	39	100

**FUENTE: SOCIOS ACTIVOS “CITULASA”
ELABORADO POR: TESISISTAS**

GRAFICO N° 12



Las 39 personas encuestadas respondieron que SIEMPRE 5%, CASI SIEMPRE el 10%, DE VEZ EN CUANDO 18% y NUNCA 67%, respectivamente por lo que se recurre a la información de archivos de la compañía en una minoría de porcentaje siendo un factor principal que los datos de cada socio deben ser actualizados veraz y ordenadamente.

PREGUNTA N° 2

¿Usted como socio hace uso de la información del Área Administrativa dentro de la Compañía de Transporte Público Urbano Popular de Pasajeros “Latacunga S.A. CITULASA”?

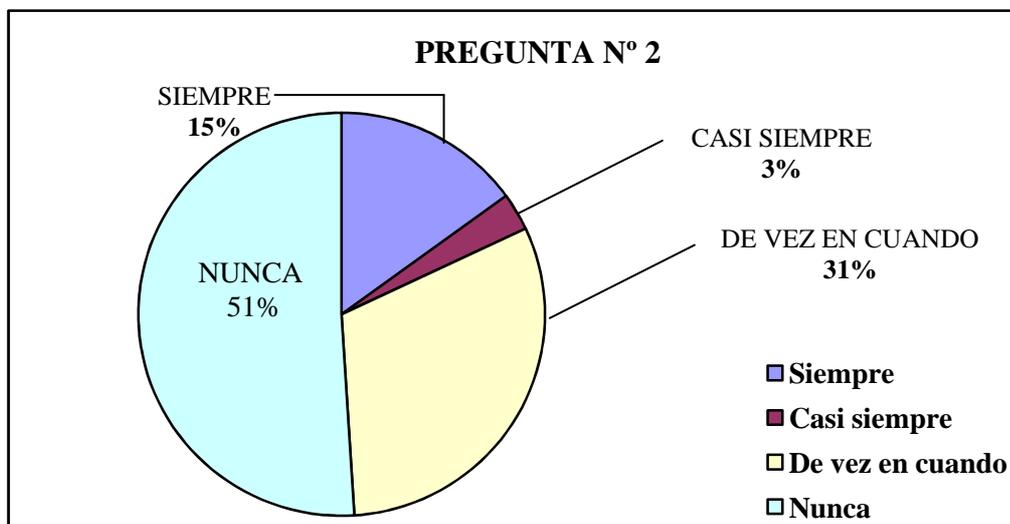
TABLA N° 2

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SIEMPRE	6	15
CASI SIEMPRE	1	3
DE VEZ EN CUANDO	12	31
NUNCA	20	51
TOTAL	39	100

FUENTE: SOCIOS ACTIVOS “CITULASA”

ELABORADO POR: TESISISTAS

GRAFICO N° 13



Las 39 personas encuestadas respondieron que SIEMPRE 15%, CASI SIEMPRE 3%, DE VEZ EN CUANDO 31% y NUNCA 51% respectivamente la mayor parte de los socios no recurre a la información de los Archivos de la Compañía, y es una mínima parte en el porcentaje que si hacen uso de la información administrativa.

PREGUNTA N° 3

¿Usted como socio conoce si la Compañía de Transporte Público Urbano Popular de Pasajeros “Latacunga S.A. CITULASA”, cuenta con un Sistema Computacional?

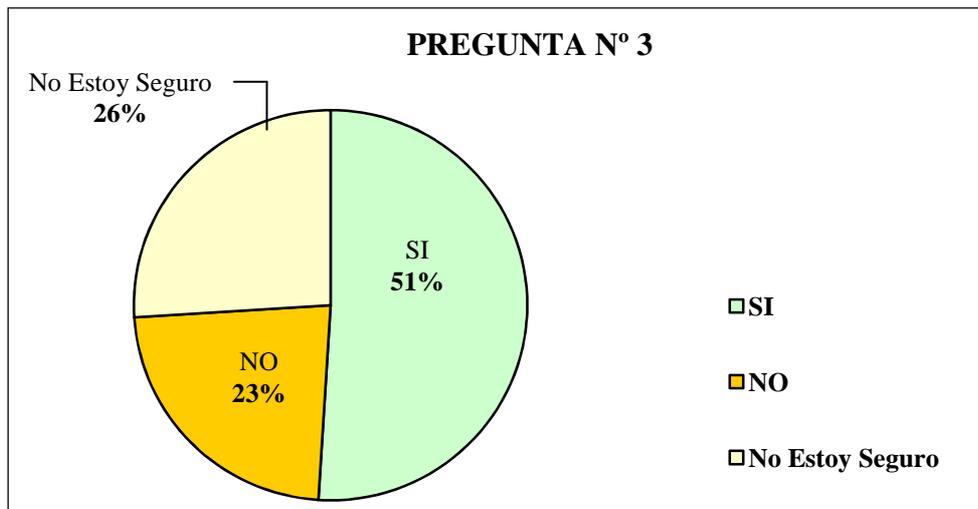
TABLA N° 3

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SI	20	51
NO	9	23
NO ESTOY SEGURO	10	26
TOTAL	39	100

FUENTE: SOCIOS ACTIVOS “CITULASA”

ELABORADO POR: TESISTAS

GRAFICO N° 14



Las 39 personas encuestadas respondieron que SI 51%, NO 23% y NO ESTOY SEGURO 26% respectivamente la mayor parte de los socios está seguro que la compañía cuenta con un sistema computacional pero la minoría de encuestados no conoce de un sistema en la institución.

PREGUNTA N° 4

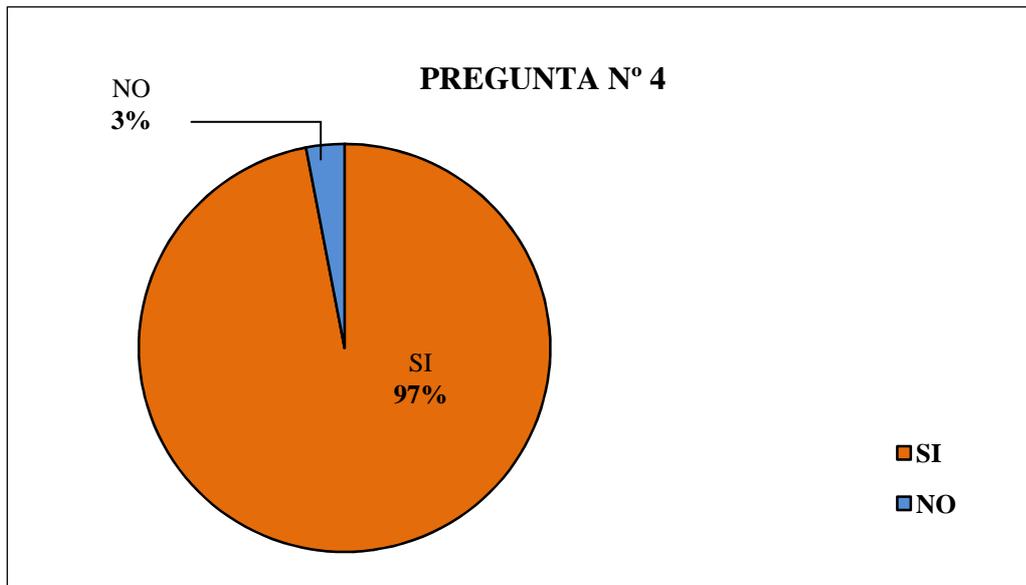
¿Cree usted que la Compañía de Transporte Público Urbano Popular de Pasajeros “Latacunga S.A. CITULASA” debe contar con un Sistema Computacional para el Área Administrativa?

TABLA N° 4

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SI	38	97
NO	1	3
TOTAL	39	100

FUENTE: SOCIOS ACTIVOS “CITULASA”
ELABORADO POR: TESISTAS

GRAFICO N° 15



Las 39 personas encuestadas respondieron que SI 97%, NO 3% respectivamente la mayor parte de los socios están seguro que la compañía cuenta con un sistema computacional pero la minoría de encuestados no conoce de un sistema en la Institución.

PREGUNTA N° 5

¿Cree usted que debe existir capacitación en el Área Informática dentro de la Compañía de Transporte Público Urbano Popular de Pasajeros “Latacunga S.A. CITULASA”?

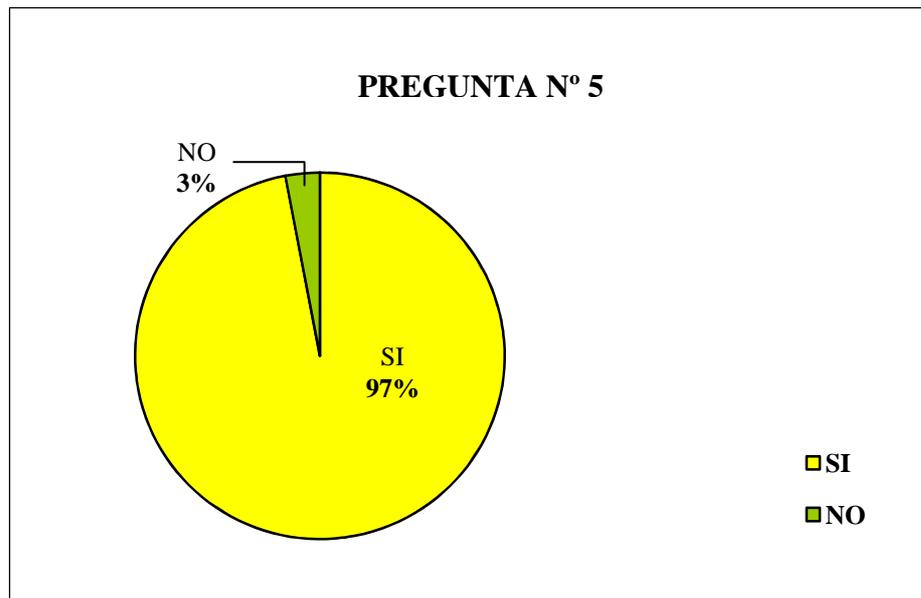
TABLA N° 5

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SI	38	97
NO	1	3
TOTAL	39	100

FUENTE: SOCIOS ACTIVOS “CITULASA”

ELABORADO POR: TESISISTAS

GRAFICO N° 16



Las 39 personas encuestadas respondieron que SI 97% y NO 3% respectivamente la mayor parte de los socios están seguros que la compañía debe contar con una capacitación constante dentro del Área Informática para que brinden un excelente servicio a todas las personas que integran la institución.

PREGUNTA N° 6

¿Usted como socio tiene el conocimiento de cómo se genera los turnos para las unidades dentro la Compañía de Transporte Público Urbano Popular de Pasajeros “Latacunga S.A. CITULASA”?

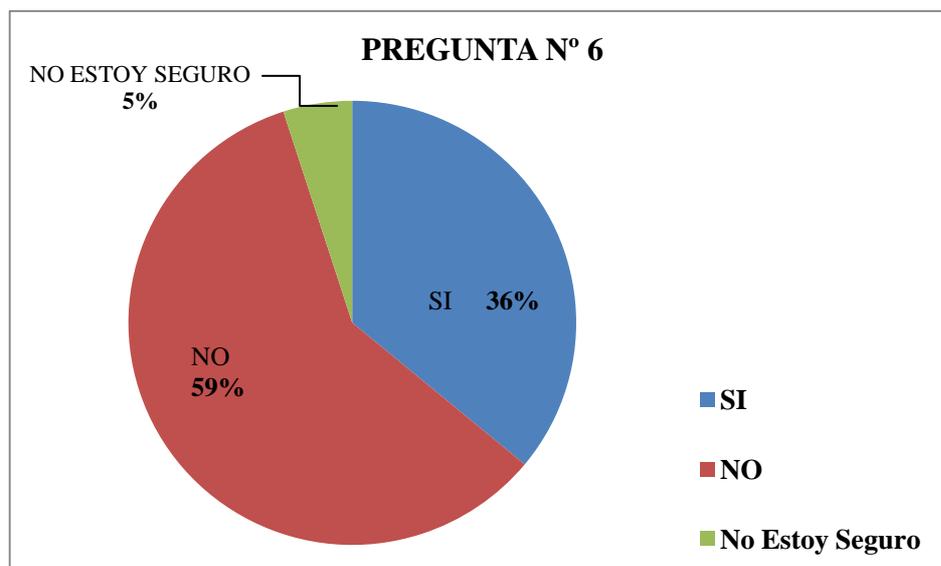
TABLA N° 6

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SI	14	36
NO	23	59
No Estoy Seguro	2	5
TOTAL	39	100

FUENTE: SOCIOS ACTIVOS “CITULASA”

ELABORADO POR: TESISISTAS

GRÁFICO N° 17



Las 39 personas encuestadas respondieron a la 6 pregunta con el SI 36%, NO 59% y NO ESTOY SEGURO 5 % respectivamente, por lo que en un gran porcentaje manifiestan que no tienen conocimiento cómo se generan los turnos, siendo fundamental para la orden de trabajo que diariamente rotan para las diferentes unidades de Transporte Urbano, de esta manera establecer un orden y ejemplo para la colectividad de la Provincia de Cotopaxi.

PREGUNTA N° 7

¿Usted como socio cuántas veces recurre a los Documentos de la Compañía de Transporte Público Urbano Popular de Pasajeros “Latacunga S.A. CITULASA”?

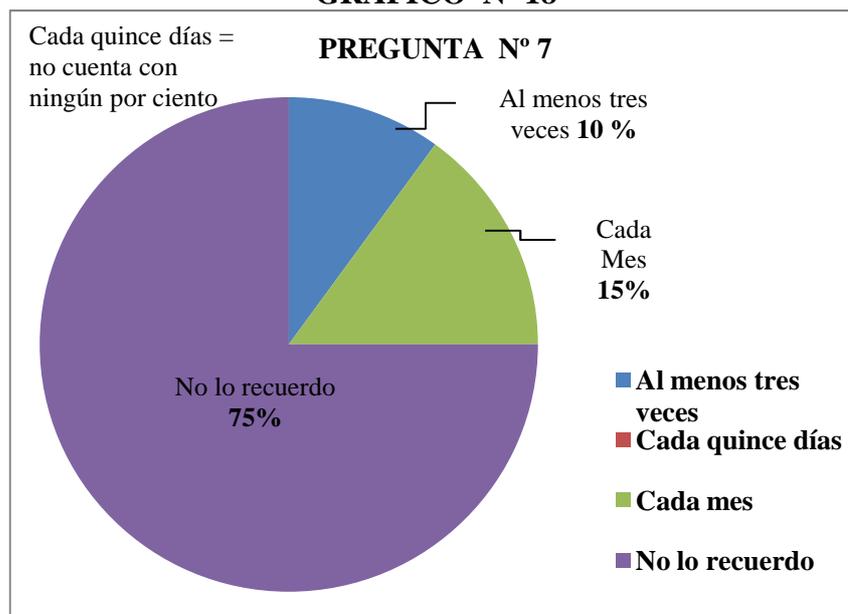
TABLA N° 7

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
Al menos tres veces	4	10
Cada quince días	0	0
Cada mes	6	15
No lo recuerdo	29	75
TOTAL	39	100

FUENTE: SOCIOS ACTIVOS “CITULASA”

ELABORADO POR: TESISTAS

GRÁFICO N° 18



Las 39 personas encuestadas respondieron a la 7 pregunta CON AL MENOS TRES VECES 10%, CADA QUINCE DÍAS 0%, CADA MES 15% y NO LO RECUERDO 75% respectivamente, de esta manera se puede demostrar que en su mayoría de socios no acceden habitualmente a los datos Informativos y Documentos de la Institución por falta de ayuda Tecnológica ya que el servicio que prestan actualmente dificulta una ayuda constante al socio activo.

PREGUNTA N° 8

¿Usted como socio sabe para qué sirve un Sistema Administrativo?

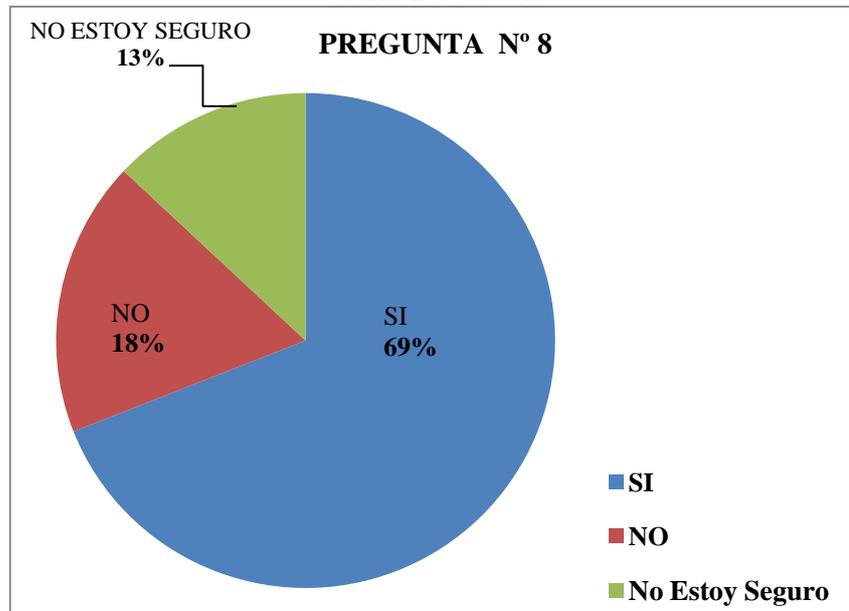
TABLA N° 8

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SI	27	69
NO	7	18
No Estoy Seguro	5	13
TOTAL	39	100

FUENTE: SOCIOS ACTIVOS "CITULASA"

ELABORADO POR: TESISTAS

GRÁFICO N° 19



Las 39 personas encuestadas respondieron a la 8 pregunta con SI el 69%, NO el 18% y NO ESTOY SEGURO el 13% respectivamente, la mayoría de socios activos de la Compañía manifiestan el manejo de un Sistema Administrativo, el mismo que ayudará al desenvolvimiento de la Documentación de la Institución y al ahorro de tiempo en los diferentes archivos.

PREGUNTA N° 9

¿Usted como socio cuántas veces a la semana consulta el Horario de Entrada/Salida de su Unidad de Transporte?

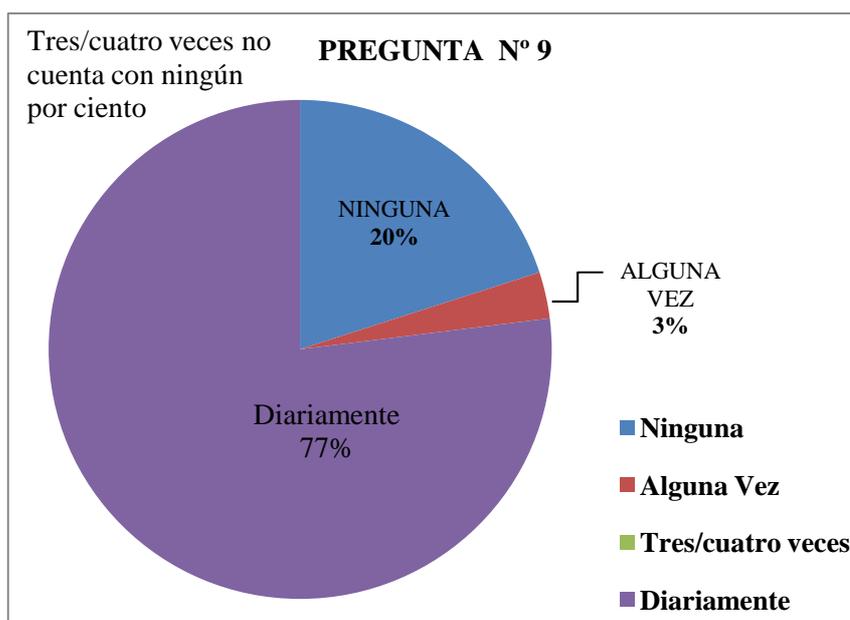
TABLA N° 9

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
Ninguna	8	20
Alguna Vez	1	3
Tres/Cuatro Veces	0	0
Diariamente	30	77
TOTAL	39	100

FUENTE: SOCIOS ACTIVOS "CITULASA"

ELABORADO POR: TESISISTAS

GRÁFICO N° 20



Las 39 personas encuestadas respondieron a la 9 pregunta con NINGUNA el 20%, ALGUNA VEZ el 3%, TRES/CUATRO VECES el 0% y DIARIAMENTE el 77% respectivamente, recurren diariamente a los Horarios de Entrada/Salida de su Unidad en un gran porcentaje, de esta manera controlan en la brevedad posible y permiten que exista un orden establecido.

PREGUNTA N° 10

¿Usted como socio sabe si alguna vez la compañía de transporte público urbano popular de pasajeros “Latacunga S.A. CITULASA” ha intentado realizar un Sistema Administrativo?

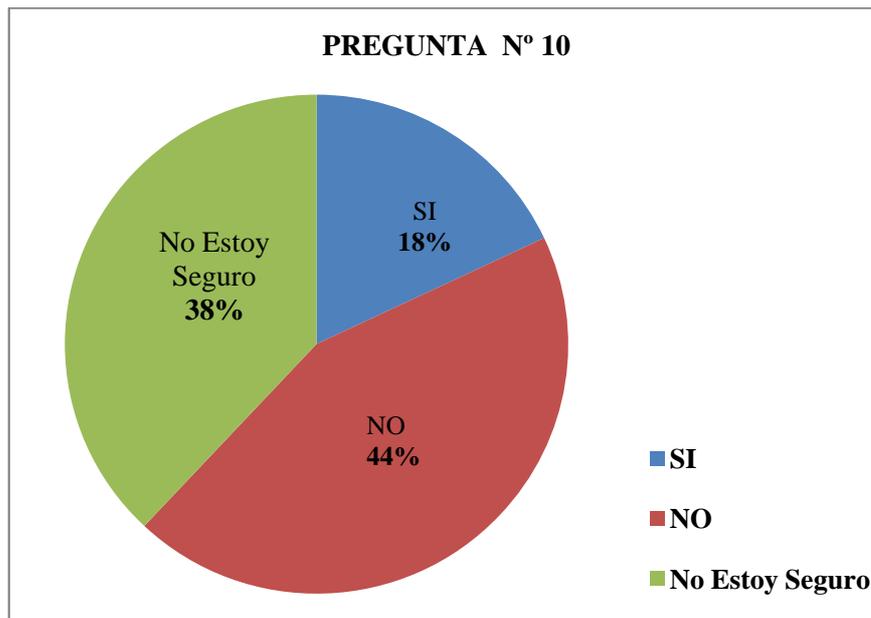
TABLA N° 10

ALTERNATIVAS	FRECUENCIA	PORCENTAJE
SI	7	18
NO	17	44
No Estoy Seguro	15	38
TOTAL	39	100

FUENTE: SOCIOS ACTIVOS “CITULASA”

ELABORADO POR: TESISISTAS

GRÁFICO N° 21



Las 39 personas encuestadas respondieron a la 10 pregunta con SI el 18%, NO el 44% y NO ESTOY SEGURO el 38% respectivamente, en su gran mayoría de porcentaje no tienen conocimiento si la Institución ha intentado realizar un Sistema Administrativo, la misma que sería de gran ayuda para llevar de una manera automatizada toda la Información de la Compañía.

2.3. ANÁLISIS DE RESULTADOS DE LA ENTREVISTA APLICADA AL ÁREA ADMINISTRATIVA DE LA COMPAÑÍA DE TRANSPORTE PÚBLICO URBANO POPULAR DE PASAJEROS “LATACUNGA S.A CITULASA” VER ANEXO N° 4.

2.3.1. INTERPRETACIÓN DE LOS RESULTADOS DE LA ENTREVISTA APLICADA AL ÁREA ADMINISTRATIVA DE LA COMPAÑÍA DE TRANSPORTE PÚBLICO URBANO POPULAR DE PASAJEROS “LATACUNGA S.A CITULASA”.

PREGUNTA N° 1

¿Qué tipo de información ofrece la Institución como Compañía de Transporte Público Urbano Popular de Pasajeros “Latacunga S.A. CITULASA”?

La compañía ofrece la información de documentos tanto internos como externos para que así el socio este informado de las actividades que cumple la institución estos archivos pueden ser actas de ingresos y egreso de capital y a la vez puede dar a conocer las fichas de cada socio que pertenece a la compañía, y dan a conocer a cada socio los diferentes turnos para el buen trabajo en las calles de la ciudad de Latacunga; de esta manera se puede evitar los diferentes roces y enfrentamientos con otras entidades del transporte urbano.

PREGUNTA N°2

¿En la Compañía de Transporte Público Urbano Popular de Pasajeros “Latacunga S.A. CITULASA”, a qué tipo de archivos o documentos puede acceder el Socio Activo y/o Pasivo?

Los socios pueden acceder con la respectiva autorización del presidente a la diferente documentación con la que cuenta la institución como:

- ❖ Constitución de la compañía
- ❖ Balance de activos
- ❖ Actas de juntas generales (Oficio al consejo administrativo)
- ❖ Ingreso y egreso de capital
- ❖ Turnos de cada unidad
- ❖ Fichas de los diferentes socios activos de la Institución.

PREGUNTA N° 3

¿Cómo ha ido evolucionando al pasar de los años la Compañía de Transporte Público Urbano Popular de Pasajeros “Latacunga S.A. CITULASA”?

La compañía de transporte ha ido evolucionando con eficacia ya que con el apoyo de cada socio apoyado desinteresadamente al crecimiento de la institución; de esta manera el Transporte Urbano CITULASA ofrece un servicio tanto a personas naturales y jurídicas como a públicas y privadas, con vehículos debidamente acondicionados para el efecto que reúnan las condiciones de confort y comodidad.

Con esto la organización ha ido ganando popularidad en la ciudad de Latacunga y en la provincia de Cotopaxi que ha llevado al engrandecimiento de dicha entidad y así crear en el transporte urbano esperanza y ganas de trabajar en beneficio de la colectividad.

PREGUNTA N° 4

¿Cuál es la relación que mantiene el Área Administrativa con los diferentes estamentos dentro de la Compañía de Transporte Público Urbano Popular de Pasajeros “Latacunga S.A. CITULASA”?

La relación que tiene la administración de la compañía es de suma importancia ya que de estas personas que conforman el Área Administrativa parten las diferentes actividades que se llevara a efecto como cabeza principal de la institución.

De esta manera se comparten ideas, opiniones, críticas y sugerencias para el bienestar y engrandecimiento de la institución.

A la vez los administrativos proporcionan centros de comercialización de automotores, repuestos, accesorios, grasas, lubricadoras y aceites para el buen desempeño del vehículo, en general la relación que tiene el Área Administrativa con los diferentes estamentos es factible ya que proporcionan informaciones vitales para una organización con objetivos claros y precisos.

PREGUNTA N° 5

¿Usted está de acuerdo con la creación de un Sistema de Administración para la Compañía de Transporte Público Urbano Popular de Pasajeros “Latacunga S.A. CITULASA”?

Como socio administrativo de la compañía será muy necesario un sistema administrativo ya que la tecnología ha ido evolucionando con el pasar de los años y se ha hecho inevitable en la actualidad, con esto ayudara a fomentar servicios automatizados en la compañía y evitar un conjunto de problemas en las diferentes actividades que realiza la institución, al realizar procesos manuales, esto hace que la documentación se conviertan lentas la misma que ocasiona pérdida de tiempo, con el prototipo del sistema se integraran los procesos manuales a procesos automatizados, que proporcionaran grandes cambios dentro de la compañía y de esta manera fortalecer los lazos de hermandad entre la institución y la Provincia.

PREGUNTA N° 6

¿Qué papel cumple el Gerente dentro de la Compañía de Transporte Público Urbano Popular de Pasajeros “Latacunga S.A. CITULASA”?

El gerente dentro de la compañía cumple un papel muy importante ya que es la persona primordial que ayuda a la aprobación de objetivos claros y precisos.

En beneficio de la organización y a la vez la persona que representa, judicial y extrajudicialmente a la Compañía dentro o fuera de la misma.

La persona encargada de la gerencia organiza todo el personal que prestará servicio dentro de la Compañía de acuerdo al organigrama aprobado por el directorio y tiene la obligación de supervisar los libros de contabilidad y responder por la conservación de todos los documentos, libros, enseres, y activos en general de la institución; y es un integrante que debe asistir obligatoriamente a las sesiones de la Junta General y del Directorio.

2.4. VERIFICACIÓN DE LA HIPÓTESIS

La hipótesis que se planteo al iniciar el proyecto de investigación es la siguiente:

Al realizar en la Compañía de Transporte Público Urbano Popular de Pasajeros “LATACUNGA S.A. CITULASA” un SISTEMA PROTOTIPO DE ADMINISTRACIÓN mejorará el ingreso de datos de cada uno de los socios y el servicio de turnos para las unidades y así brindar un mejor servicio a la colectividad.

2.4.1. COMPROBACIÓN DE LA HIPÓTESIS PLANTEADA AL INICIO DEL PROYECTO

Al comparar la metodología XP con la metodología MSF (Microsoft Solution frameworks) con la ayuda de la investigación científica hemos podido establecer que la metodología MSF es factible para diseñar proyectos a pequeños y grandes escalas ya que su análisis se da paso a paso para obtener resultados satisfactorios, pero al utilizar esta metodología se tiene que comprar software con su respectiva licencia para poder trabajar con nuevas metodologías que se presentan planteadas por Microsoft.

La elaboración y análisis de esta metodología con un Sistema Prototipo nos ayudo a que tengamos claro cómo trabaja la metodología y sus fases, y al emplearlas ya

en un caso práctico nos hemos dado cuenta que cumplen un papel importante ya que estas fases van enlazadas entre sí, y no se puede continuar el proyecto cuando un ciclo no esté terminado.

El Sistema Prototipo ayudó a tomar pautas y que se siga procesos o pasos para poder llegar a una adecuada finalización de un proyecto sean estos grandes o pequeños.

En la actualidad todas las empresas deben utilizar metodologías nuevas y en especial MSF (Microsoft Solution Frameworks), ya que es uno de los métodos viables y satisfactorios para desarrollar proyectos a gran escala.

Con el respectivo análisis hemos llegado a comprobar que la metodología MSF (Microsoft Solution Frameworks) es factible para el diseño de software pero tomando en cuenta cómo funcionan sus fases, para así tener éxito cuando un programador tenga que realizar proyectos de diseño o desarrollo de software.

Al analizar la Metodología MSF (MICROSOFT SOLUTIONS FRAMEWORKS), y sus diferentes Fases, utilizando un Ciclo de Vida acorde al análisis se ha concluido que una **“METODOLOGÍA VA DENTRO DE UN CICLO DE VIDA”** porque para el desarrollo eficiente de proyectos grandes y/o pequeños se debe iniciar con un Análisis previo, de cada una de las Metodologías que se destacan en el campo Informático y a la vez conocer cada una de sus fases y así seleccionar el método eficaz para diseñar proyectos a gran escala, tomando en cuenta que esta debe trabajar con un ciclo de vida, ya que ayudará al mejor desenvolvimiento de la Metodología, para que sea aplicada en el Diseño de Software para obtener resultados óptimos en la ejecución de Fases y Procesos en cada una de sus aplicaciones.

CAPITULO III

PROPUESTA ALTERNATIVA

3.1. TEMA

“INVESTIGAR Y ANALIZAR LA METODOLOGÍA MSF (MICROSOFT SOLUTIONS FRAMEWORKS), A TRAVÉS DE UN SISTEMA PROTOTIPO; DE ESTA MANERA COMPROBAR SI SUS CONDICIONES SON FACTIBLES Y VIABLES PARA EL DESARROLLO Y LA APLICACIÓN DE SOFTWARE EN LOS DIFERENTES CAMPOS DE LA INFORMÁTICA”.

3.2. PRESENTACIÓN

En la actualidad la tecnología computacional ha ido evolucionando cada día más y a la vez abierto nuevos campos en el área informática, es el motivo por el cual se ha creído conveniente realizar un Sistema Prototipo de Administración; de esta manera se impulsará al desarrollo y el engrandecimiento de la Compañía de Transporte Público Urbano Popular de Pasajeros “LATACUNGA S.A. CITULASA”, brindando una información documental clara, precisa y eficiente para que esté acorde a cada una de las diferentes instituciones de Transporte Urbano.

Al investigar y analizar cada una de las metodologías que se encuentran inversas en el campo informático, se ha encontrado que la metodología MSF (Microsoft Solution Frameworks), se puede aplicar en proyectos grandes y/o pequeños de tal forma como investigadores se ha trabajado con la metodología antes mencionada ya que sus fases son entendibles y fáciles de desarrollarlas en el diseño y aplicación de Software.

Esta metodología ayudará a que se pueda cumplir con todos los objetivos planteados, y a la vez incrementar los conocimientos en el análisis de proyectos a gran escala utilizando un método en el cual estemos identificados constantemente como diseñadores, ya que las metodologías van cambiando frecuentemente en el proceso tecnológico.

3.3. JUSTIFICACIÓN

Al establecer el proyecto de Investigación se ayudará a incrementar los conocimientos Tecnológicos y Éticos, y a la vez se ofrecerá beneficios múltiples a la Institución para brindar mayor Fluidez de datos documentales, y al control de turnos de las diferentes unidades tanto entradas / salidas diariamente de cada vehículo, la cual ayudará a que los documentos que se llevan manualmente en la Compañía no se pierdan y no sean clonados; así se podrá disminuir los procesos manuales y su respectivo almacenamiento en archiveros, reduciendo en su totalidad la información que se transcribe diariamente en un escrito y así evitar la pérdida de tiempo y poder crear trabajos mancomunados que vayan en bien de la Compañía y de la Colectividad.

Al analizar el sistema prototipo servirá como soporte la Metodología MSF (Microsoft Solution Frameworks), ya que es uno de los métodos más factibles por la facilidad de comprensión en cada una de sus fases y servirá de apoyo para poder finalizar con el trabajo investigativo, y así poder incrementar nuestro conocimiento en el diseño y planificación de un proyecto de diseño de software a gran escala en la vida profesional.

Al estar inversos en el análisis de proyectos grandes y/o pequeños se ha contemplado que una metodología siempre va conjuntamente con cada una de sus fases para obtener un mejor diseño del proyecto que se plantea, para así tener resultados satisfactorios al culminar el propósito deseado.

Al analizar el sistema prototipo se está en la capacidad de fomentar y dar a conocer que cada una de las metodologías necesita de un ciclo de vida, para así cumplir con todos los requerimientos con la cual cuenta una metodología a utilizar.

3.4. OBJETIVOS

3.4.1. Objetivo General:

Analizar la Metodología MSF (MICROSOFT SOLUTIONS FRAMEWORKS), para determinar su fiabilidad en un Sistema Prototipo y que puedan ser ejecutados en grandes proyectos de Software.

3.4.2. Objetivos Específicos:

- ❖ Automatizar procesos manuales para brindar un mejor servicio a la colectividad.
- ❖ Facilitar a los socios activos de la Compañía un Sistema Prototipo de Administración, la misma que ayudará al desenvolvimiento de las actividades diarias en el Transporte Urbano.
- ❖ Mejorar los archivos existentes en la Institución para prevenir en su totalidad la pérdida de documentos que se lleva manualmente.

3.5. IMPACTO

El presente proyecto causará gran impacto dentro del Área Administrativa de la Compañía de Transporte Público Urbano Popular de Pasajeros “LATACUNGA S.A. CITULASA” y de manera principal plantear un Sistema Prototipo Administrativo, utilizando la nueva Metodología MSF, ya que sus beneficios y características son confiables, al ser aplicados a grandes y pequeños proyectos dentro del desarrollo de Software.

Por otro lado dentro del campo Institucional del Transporte Urbano en la actualidad no se ha aplicado la nueva Metodología MSF, está a la vez ofrece seguridad y confiabilidad al ingresar cada uno de los datos, y restringir datos importantes para la Compañía, dando así permiso de acceso a personas autorizadas de la Compañía.

Al conocer la nueva metodología se tomará en cuenta que para llegar a establecer un proyecto tenemos que seguir pasos y fases, para cumplir con los objetivos que se plantean al dar inicio al análisis de proyectos a gran escala.

La metodología estudiada ayudó a llenar vacíos que, como diseñadores no se conocía, dando así una mejor comprensión para el diseño de un proyecto y poderlo ejecutar en nuestra vida profesional, y poder compartir conocimientos con otros diseñadores que están inversos en diseño de software.

3.6. DESARROLLO TÉCNICO Y/O TECNOLÓGICO

3.6.1. QUE ES VISUAL STUDIO.NET

Visual Studio.Net es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y dispositivos móviles.

3.6.1.1. REQUERIMIENTOS DE VISUAL STUDIO.NET

Según Microsoft, los requerimientos mínimos de software y hardware para Visual Studio.Net son los siguientes:

- ❖ **PROCESADOR:** 600 Mhz, recomendado 1Ghz
- ❖ **RAM:** 128MB, recomendado 256MB
- ❖ **DISCO DURO:** 67.2GB espacio libre en disco
- ❖ Windows XP profesional SP2

3.6.1.2. HERRAMIENTAS QUE AYUDARÁ AL MODELAMIENTO Y AL DESARROLLO DE SOFTWARE.

3.6.1.2.1. POWER DESIGNER

Power Designer es una de las herramientas de modelamiento más utilizadas por las empresas desarrolladoras de software ya que es muy fácil visualizar y manipular metadatos para lograr una efectiva arquitectura dentro la información tecnológica.

Esta herramienta brinda técnicas de análisis, diseño y gestión de metadatos para el desarrollo de proyectos informáticos.

3.6.1.2.2. RATIONAL ROSE

Rational Rose es una herramienta software para el modelamiento Visual mediante UML de sistemas software, esta herramienta permite especificar, analizar, diseñar el sistema antes de codificarlo.

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

Características de Microsoft SQL Server

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Además permite administrar información de otros servidores de datos.

3.6.1.2.5. MODELO DE DESARROLLO DE SOFTWARE SELECCIONADO PARA TRABAJAR EN EL SISTEMA PROTOTIPO.

Al analizar cada uno de los modelos del ciclo de vida se tomó como primordial **EL MODELO EN CASCADA**, ya que es uno de los modelos que tiene mejor comprensión en su análisis y cada uno de sus pasos o fases están acorde con la metodología que se utilizó en el desarrollo del Sistema Prototipo y a la vez poder finalizar con éxito el desarrollo de la aplicación.

3.6.2. COMPARACIÓN DE LAS METODOLOGÍAS XP (EXTREME PROGRAMING) Y MSF (MICROSOFT SOLUTIONS FRAMEWORKS)

3.6.2.1. METODOLOGÍA XP (EXTREME PROGRAMING)

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

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.

Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. Kent Beck, el padre de XP, describe la filosofía de XP sin cubrir los detalles técnicos y de implantación de las prácticas. Posteriormente, otras publicaciones de experiencias se han encargado de dicha tarea. A continuación presentaremos las características esenciales de XP organizadas en los tres apartados siguientes: historias de usuario, roles, proceso y prácticas.

Las principales características de esta metodología son las siguientes:

- ❖ **Comunicación:** Los programadores están en constante comunicación con los clientes para satisfacer sus requisitos y responder rápidamente a los cambios de los mismos. Muchos problemas que surgen en los proyectos se deben a que después de concretar los requisitos que debe cumplir el programa no hay una revisión de los mismos, pudiendo dejar olvidados puntos importantes.
- ❖ **Simplicidad:** Codificación y diseños simples y claros. Muchos diseños son tan complicados que cuando se requiere mantenimiento o ampliación resulta imposible hacerlo y se tienen que desechar y partir de cero.
- ❖ **Realimentación (Feedback):** Mediante la realimentación se ofrece al cliente la posibilidad de conseguir un sistema adecuado a sus necesidades. Se le va mostrando el proyecto a tiempo para sugerir cambios y poder retroceder a una fase anterior para rediseñarlo a su gusto.
- ❖ **Tenacidad:** Se debe ser tenaz para cumplir los tres puntos anteriores. Hay que tener valor para comunicarse con el cliente y enfatizar algunos puntos a pesar de que esto pueda dar sensación de ignorancia por parte del programador; hay que ser decidido para mantener un diseño simple y no optar por lo que pudiera parecer mejor o un camino más fácil y por último hay que enfatizar que la realimentación será efectiva.

➤ **VENTAJAS**

- ❖ Detectar más fácilmente los errores de programación (el programador libre controla al que tipea)
- ❖ El programador poco experimentado aprende del que más lo está.
- ❖ Si una pareja consigue desarrollar algún trozo de código reutilizable, se comunica más fácilmente a los otros programadores.

➤ **DESVENTAJAS**

- ❖ Es recomendable emplearlo solo en proyectos a corto plazo.
- ❖ Altas comisiones en caso de fallar.
- ❖ No está en la capacidad de ofrecer un enfoque para determinados proyectos como el desarrollo de software.

3.6.2.2. METODOLOGÍA MSF (MICROSOFT SOLUTIONS FRAMEWORKS)

El MSF es un modelo diseñado específicamente para crear productos de muy buena calidad, donde para cumplir este objetivo prima la comunicación tanto entre el equipo de desarrollo como entre ellos y los clientes.

En el escenario de modelos de mejora para la producción de software, se ha marcado dos tendencias cada una de ellas con un contraste muy definido.

Por un lado se encuentran aquellos procesos que tienen gran cantidad de información ya que así han sido impuestas por organismos internacionales y son muy apreciados por grandes corporaciones o industrias de software.

Por otro lado se encuentran los modelos ágiles con procesos más rápidos y sencillos de desarrollo, lo que ahora más llama la atención es la aparición de Microsoft en el ámbito de los modelos de desarrollo de software, parecía mantenerse al margen de esto ya que ellos han permanecido defendiendo sus propios métodos y condenando errores ajenos.

Esto ha ido generando una masa crítica de seguidores, e incluso adoctrinados en uno y otro bando, que han comenzado a reclamar de la propia industria del software herramientas y plataformas adecuadas a las distintas creencias de cada uno.

Y este es el momento en el que aparece Microsoft, porque la filosofía doctrinal está dando paso al negocio y es un negocio que se sitúa en el centro de uno de los principales focos empresariales de herramientas y plataformas para desarrollo software.

Microsoft Solution Framework no es un método ágil ni un método ortodoxo. Es un marco de trabajo flexible, válido para ambas teorías que permite trabajar con este marco, es una herramienta apropiada tanto para los unos y como para los otros.

En este modelo sus 6 fases apuntan a objetivos claros, fomentan la comunicación para alcanzar metas claras, establecen alcances y limitaciones, previenen riesgos que posiblemente causen problemas a futuro.

➤ **VENTAJAS**

- ❖ Al ser un modelo desarrollado por Microsoft se puede tener mayor soporte y mantenimiento.
- ❖ Sirve para grandes y pequeños proyectos.

- ❖ Aplica mucho e incentiva al trabajo en equipo y a la colaboración.
- ❖ Es útil para proyectos de pequeña y gran escala.
- ❖ Crea una disciplina de análisis de riesgos que ayuda y evoluciona con el proyecto.

➤ **DESVENTAJAS**

- ❖ Se torna un trabajo bastante largo, ya que para cada fase se debe documentar profundamente todo lo que se haga, pero no deja de ser un modelo que tiene buenos resultados.
- ❖ Por ser un modelo prescriptivo, solicita demasiada documentación en sus fases.
- ❖ El análisis de riesgos es necesario, pero si se lo hace muy exhaustivo puede demorar o hasta frenar el avance del proyecto.
- ❖ Al estar basado en tecnología Microsoft, trata de obligar a usar herramientas de ellos mismos pero si es posible no usar esa tecnología pero lo que esto produce es más complejidad en el proyecto.

3.6.2.2.1. MOTIVOS POR LOS CUALES SE HA SELECCIONADO LA METODOLOGÍA MSF (MICROSOFT SOLUTIONS FRAMEWORKS)

Los principales motivos para ver escogido la Metodología MSF se debe a los siguientes:

- ❖ Promover comunicaciones abiertas
- ❖ La disciplina de trabajo que propone Microsoft
- ❖ Fortalecer a los miembros del Equipo
- ❖ Establecer responsabilidades claras y compartidas
- ❖ Aprender de las experiencias
- ❖ Ayuda a la administración de proyectos grandes y pequeños.

3.6.2.2.2. CARACTERÍSTICAS GENERALES DE LA METODOLOGÍA MSF (MICROSOFT SOLUTIONS FRAMEWORKS)

Fases y definiciones del proyecto presentadas por MSF:

- ❖ Identifica el objetivo del proyecto y crea documentos con ámbito del proyecto y declaración de objetivos.
- ❖ Desarrolla especificaciones funcionales
- ❖ Crea un laboratorio de pruebas para examinar cómo funcionan las soluciones en el mundo real.
- ❖ Alcanza objetivos indicados en las fases de desarrollo
- ❖ Una forma de llevar paso por paso esta Metodología es a través de las herramientas de Microsoft como es el caso de Visual Studio.
- ❖ MSF es flexible ya que permite agregar y extender nuevas características.
- ❖ MSF no es rígido ya que sabe que no existe una sola estructura que se pueda acoplar a todos los tipos de proyectos.
- ❖ Es una Metodología integrada, ya que cambia muchos elementos y características y además, es una metodología productiva, ya que incrementa la productividad de todo el equipo de trabajo.
- ❖ MSF es una metodología de mejores prácticas para el desarrollo de software.
- ❖ Se enfoca más en las habilidades y cualidades de las personas que en la eficacia de los modelos de procesos.
- ❖ MSF está basado en mejores prácticas del mundo real, basado en las experiencias de Microsoft.

3.6.4. FASES A UTILIZAR PARA EL DESARROLLO DE SOFTWARE

3.6.4.1. FASES DE MSF

3.6.4.1.1. FASE 1.- INICIO / VISIONAMIENTO

El equipo debe tener una visión clara de lo que quiere lograr y esto debe ser expresado de manera que motive a los miembros del grupo y al cliente.

Una de las primeras actividades que debe lograrse durante el visionamiento es la formación de un equipo base y la preparación y entrega del documento acerca de la visión y el alcance del proyecto. La determinación del alcance de la visión del proyecto son actividades diferentes, pero ambas necesarias para lograr un proyecto exitoso.

Establece las estrategias de prueba y analizar la documentación de proyecto (visión, alcance, cronograma inicial, posibles riesgos). **VER ANEXO N° 5.**

3.6.4.1.1.1. ANÁLISIS DE LA FASE INICIO / VISIONAMIENTO

- ❖ Es la primera fase de modelo MSF
- ❖ Se pretende dar una visión general del proyecto
- ❖ Se identificó las tareas y los entregables que permiten al equipo cumplir con los requerimientos y objetivos del proyecto.

3.6.4.1.1.2. RESPONSABILIDAD DE LOS INVESTIGADORES

➤ ANÁLISIS DEL SOFTWARE PROTOTIPO

- ❖ Se asegura que el software respectivo cumpla con las necesidades del socio y administradoes.
- ❖ Colaboración completa con los diseñadores del programa.

- ❖ Asegura que establezca una visión clara del Proyecto.

➤ **MISIÓN DEL PROYECTO**

- ❖ Establece metas precisas para el diseño del proyecto
- ❖ Definición de factores para el éxito durante la elaboración del proyecto
- ❖ Se Organizó la infraestructura del proyecto

➤ **DESARROLLO**

- ❖ Investigación por parte de los investigadores para obtener técnicas de desarrollo

➤ **PRUEBAS**

- ❖ Alimentación a los investigadores sobre las meta de calidad de la solución
- ❖ Se especificó las acciones que se necesitarán para alcanzar el nivel de calidad

➤ **EXPERIENCIA DEL ADMINISTRADOR Y/O SOCIO**

- ❖ Conceptualización de las necesidades de rendimiento de la Institución y los temas de soporte relativos a los usuarios.
- ❖ Considerar las implicaciones del sistema prototipo para que cumpla con esas necesidades.

3.6.4.1.1.3. DOCUMENTO DE LA ESTRUCTURA DE INVESTIGACIÓN

El presente documento, describe la estructura administrativa, los estándares, procesos, recursos y limitaciones del propósito.

Este documento sirve como referencia a los integrantes para saber cómo trabajar juntos de un modo eficiente y satisfactorio.

Los componentes que se utilizarán en el respectivo proyecto de Investigación son los siguientes:

- ❖ Investigadores
- ❖ Procesos del Proyecto
- ❖ Cronograma del proyecto

3.6.4.1.1.4. ENTREGABLES

Una vez definido y analizado la fase de Inicio/Visionamiento se puede obtener resultados entregables la misma que servirá para el desenvolvimiento eficaz de esta primera Fase. Y a continuación presentamos las siguientes:

➤ DOCUMENTO DE VISIÓN/ALCANCE.

- ❖ Problemas y objetivos de la Institución
- ❖ Definiciones claras de los requerimientos de los administrativos
- ❖ Perfiles de socios, identificando quién se beneficia de la aplicación

➤ ESTRUCTURA DEL PROYECTO

- ❖ Recopilación de los datos informativos existentes en la Institución
- ❖ Estructura del proyecto y esquemas a seguir por los investigadores.

➤ ELABORACION DE UN PLAN DE PROYECTO

- ❖ Antecedentes
- ❖ Objetivos
- ❖ Alcance
- ❖ Estructura de Desglose de Trabajo
- ❖ Definición de roles
- ❖ Asignación de Responsabilidades

- ❖ Cronograma
- ❖ Presupuesto
- ❖ Supuestos, Restricciones y Factores Críticos de Éxito
- ❖ Administración de riesgos
- ❖ Manejo de cambios
- ❖ Administración y Comunicación. **VER ANEXO N° 6.**

➤ **ENTREGABLES: FASE INICIO / VISIONAMIENTO**

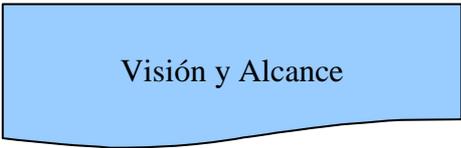
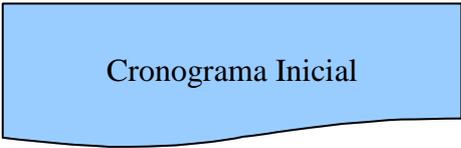
ENTREGABLES	DESCRIPCIÓN
 <p style="text-align: center;">Visión y Alcance</p>	<p>Este entregable nos detalla una estrategia que nos va a llevar al éxito del presente proyecto.</p>
 <p style="text-align: center;">Plan de Proyecto</p>	<p>Este entregable enumera y proporciona documentos rectificandos de la Institución para analizarlo durante todo el proceso.</p>
 <p style="text-align: center;">Cronograma Inicial</p>	<p>Este entregable contiene todas las ideas para generar fases iniciales.</p>

GRÁFICO N° 23

ENTREGABLES: FASE INICIO / VISIONAMIENTO

3.6.4.1.2. FASE 2.- ELABORACIÓN / PLANEACIÓN

Durante esta fase los investigadores desarrollan y preparan diversos puntos como: el proceso de diseño, planes de trabajo, valoraciones de costos, y fechas de entrega para los entregables, mientras los investigadores elaboran el plan de pruebas y establece su cronograma de trabajo.

El objetivo de esta fase es crear el plan de pruebas y el cronograma. Así mismo es la continuación de la fase de visionamiento. En esta fase los miembros de la Investigación toman el trabajo, hecho en la fase de visionamiento y se continúa elaborando algo más evidente para el buen desarrollo de esta fase. **VER ANEXO N° 7.**

3.6.4.1.2.1. RESPONSABILIDAD DE LOS INVESTIGADORES

➤ ANÁLISIS DEL SOFTWARE PROTOTIPO

- ❖ Asegura de que el plan de pruebas cumpla con el Sistema Prototipo según la manipulación del socio.
- ❖ Los investigadores son responsable de analizar los requerimientos que tiene la Institución.
- ❖ Analizar el estado actual de la organización.
- ❖ Crea el diseño conceptual.

➤ MISIÓN DEL PROYECTO

- ❖ Asegurar de que los Investigadores tengan todos los recursos que necesitan para completar el plan del proyecto.
- ❖ Los investigadores son los responsable del diseño general, con énfasis en el diseño lógico.

➤ **DESARROLLO**

- ❖ Asegurar de que el plan sea técnicamente factible.
- ❖ El Investigador es el responsable de crear el diseño lógico y físico de la solución y acoplarlo a un detalle eficaz.
- ❖ También determina el tiempo y esfuerzo para desarrollar y asegurar la solución a cada uno de los problemas.

➤ **PRUEBAS**

- ❖ Asegura que el plan cumpla con los requerimientos.
- ❖ Es responsable el investigador de evaluar el diseño para determinar qué características pueden ser probadas.

➤ **EXPERIENCIA DEL ADMINISTRADOR Y/O SOCIO**

- ❖ Asegurar que los socios podrán utilizar el sistema prototipo.
- ❖ Es responsable el investigador de analizar las necesidades del socio activo y de crear estrategias para evaluar el uso del diseño.
- ❖ También el tiempo y el esfuerzo requerido para desarrollar los sistemas de soporte de los socios.

3.6.4.1.2.2. DETALLES UTILIZABLES

- ❖ Un detalle utilizable es un campo virtual de artefactos relacionados con el diseño y con el proyecto. Estos artefactos son creados durante la fase de planificación.
- ❖ Son resultado de actividades de diseño de los procesos conceptual, lógico y físico de la fase de planificación. Pueden incluir modelos UML como los diagramas de casos de uso.

3.6.4.1.2.3. DISEÑOS EN LA FASE DE PLANEACIÓN

Los diseños que se utilizarán en la Fase de Planificación son los siguientes:

- ❖ Diseño Conceptual
- ❖ Diseño lógico
- ❖ Diseño Físico

3.6.4.1.2.3.1. DISEÑO CONCEPTUAL

Este es el proceso de recoger, analizar y priorizar las perspectivas del negocio y del usuario acerca del problema y de la solución, y de luego crear una representación a alto nivel de la solución. **VER ANEXO N° 7.1**

Para crear diseños conceptuales usables y precisos, se necesita preparar un método efectivo de comprensión de la solución y de comunicar la solución a todos los tipos de usuarios. Para modelar las tareas cubiertas por el alcance de la solución, se generan casos de uso y escenarios de uso. **VER ANEXO N° 7.1.1**

3.6.4.1.2.3.1.1. METAS QUE PERSIGUE EL DISEÑO CONCEPTUAL

- ❖ Comprender el problema de la Institución
- ❖ Comprender los requerimientos de la Institución
- ❖ Describir el estado actual y a futuro de la Institución que se desea alcanzar

3.6.4.1.2.3.1.2. PASOS A SEGUIR EN EL DISEÑO CONCEPTUAL

- Búsqueda
- Análisis
- Optimización

3.6.4.1.2.3.1.3. MODELAMIENTO DEL DISEÑO CONCEPTUAL

Dentro de este modelamiento se conoce la forma como se estructura un modelo de Base de datos en el Diseño Conceptual.

3.6.4.1.2.3.2. DISEÑO LÓGICO

El diseño lógico es el proceso de describir la solución en términos de su organización, su estructura y la interpretación de sus partes desde la perspectiva de los Investigadores.

3.6.4.1.2.3.2.1. IDENTIFICAR LOS ATRIBUTOS EN EL DISEÑO LÓGICO

Atributo de un objeto se le llama a las definiciones de los valores de datos que contiene el objeto. También son conocidos como propiedades. Al grupo de valores de los atributos de un objeto se le conoce como estado del objeto.

3.6.4.1.2.3.2.2. IDENTIFICAR LOS SERVICIOS EN EL DISEÑO LÓGICO

Servicio es llamado al comportamiento específico que puede realizar un componente de la Institución. Se refiere a una operación, función o transformación que puede aplicarse a un objeto o ser implementada por él.

Los servicios se utilizan para implementar las reglas de negocio, manipular datos y acceder a la información. Puede realizar cualquier actividad que pueda ser descrita por reglas.

3.6.4.1.2.3.2.3. IDENTIFICAR LAS RELACIONES EN EL DISEÑO LÓGICO

Las relaciones ilustran el modo en el que los objetos se relacionan entre ellos.

UML define cuatro tipos de relaciones:

- Dependencia
- Asociación
- Generalización

3.6.4.1.2.3.2.4. DOCUMENTACIÓN DE SALIDAS

➤ DIAGRAMAS DE SECUENCIAS

Los diagramas de secuencia muestran los actores y los objetos que participan en una interacción junto a la lista cronológica de eventos que los mismos generan. Para mayor visualización **VER ANEXO N° 7.2**

➤ MODELO LÓGICO DE DATOS

Se puede utilizar tanto el modelo lógico de objetos como el de datos para representar el diseño lógico.

Cuando se realiza el modelo lógico de datos, se convierten las necesidades conceptuales de datos identificadas en el diseño conceptual en entidades y relaciones que definirán cómo interactúan los datos. Esta información ayuda a modelar el diseño físico.

Al concluir con el Análisis del Diseño Lógico se puede obtener resultados entregables la misma que servirá para el desenvolvimiento eficaz de esta Segunda Fase. Y a continuación presentamos las siguientes:

3.6.4.1.2.3.2.5. ENTREGABLES

- ❖ Un modelo lógico de objetos
- ❖ Un modelo de datos lógico

3.6.4.1.2.3.3. DISEÑO FÍSICO

El diseño físico tiene como tarea principal describir los componentes, servicios y tecnologías de la solución desde la perspectiva de los requerimientos de desarrollo. Define, además, las partes de la solución que van a desarrollarse, cómo serán desarrolladas y cómo interactúan entre ellas.

Al finalizar el modelamiento del diseño físico los Investigadores entregarán un Documento donde especificará un grupo de componentes; y detalles de los reportes que usaremos para la solución. **VER ANEXO N° 7.3**

3.6.4.1.2.3.3.1. METAS DEL DISEÑO FÍSICO

- ❖ Identificar las tecnologías para el desarrollo
- ❖ Transformar el diseño lógico en modelos de diseño físico
- ❖ Brindar proceso eficientes para el respectivo desarrollo

3.6.4.1.2.3.3.2. ENTREGABLES

Una vez definido y analizado el Diseño Físico se puede obtener resultados entregables la misma que servirá para el desenvolvimiento eficaz de esta Segunda Fase. Y a continuación presentamos las siguientes:

- ❖ Diagramas de clase de la solución **VER ANEXO N° 7.3.1**
- ❖ Esquema de bases de datos de la solución.

➤ **ENTREGABLES: FASE ELABORACIÓN/PLANEACIÓN**

Entregables	Descripción
 <p>Plan de Proyectos</p>	<p>El plan de pruebas se basa en el plan estratégico, es un plan más detallado. Este documento identifica: Antecedentes, objetivos, metodologías y herramientas, responsabilidades y recursos requeridos. Este es el documento principal para los Investigadores.</p>
 <p>Modelo BBDD y cases</p>	<p>Enumera y proporciona atributos como eventos para el buen desarrollo de la Base de Datos, la cual se genera a lo largo del proyecto.</p>

GRÁFICO N° 24

ENTREGABLES: FASE ELABORACIÓN/PLANEACIÓN

3.6.4.1.3. FASE 3.- CONSTRUCCIÓN / DESARROLLO

La fase de construcción involucra el desarrollo del código de la aplicación por parte del grupo de desarrolladores. Todas las responsabilidades adquiridas involucran un papel activo en la construcción y prueba de los entregables.

La fase de desarrollo termina cuando se han alcanzado todos los propósitos planteados de esta fase y el Software está listo para ser probado y establecido. Y esta es la oportunidad para todos los administradores y los socios; evalúen la solución e identifiquen algún inconveniente que deba ser mejorado.

El objetivo principal de esta fase es preparar las respectivas pruebas basándose en los casos de uso. Para mejor visualización **VER ANEXO N° 8.**

3.6.4.1.3.1. RESPONSABILIDAD DE LOS INVESTIGADORES

➤ ANÁLISIS DEL SOFTWARE PROTOTIPO

- ❖ Expectativas de los socios y administrativos

➤ MISIÓN DEL PROYECTO

- ❖ Gestión de los detalles utilizables
- ❖ Ejecución del proyecto investigativo
- ❖ Plan de pruebas en constante actualización

➤ PRUEBAS

- ❖ Prueba práctica
- ❖ Caracterización de resultados

➤ DESARROLLO

- ❖ Infraestructura del desarrollo
- ❖ Configuración de documentación.

➤ EXPERIENCIA DEL ADMINISTRADOR Y/O SOCIO

- ❖ Formación
- ❖ Actualización del plan de formación
- ❖ Prueba de uso del sistema prototipo

3.6.4.1.3.2. ENTREGABLES

Una vez definido y analizado la fase de Construcción/Desarrollo se puede obtener resultados entregables la misma que servirá para el desenvolvimiento eficaz de esta Tercera Fase. Y a continuación presentamos las siguientes:

- ❖ Scripts de instalación
- ❖ Detalles utilizables terminada
- ❖ Detalles para pruebas.

➤ **ENTREGABLES: CONSTRUCCIÓN/DESARROLLO**

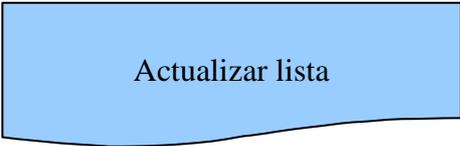
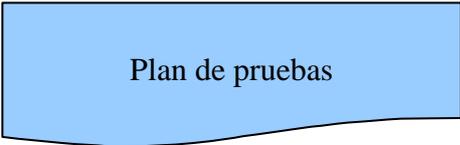
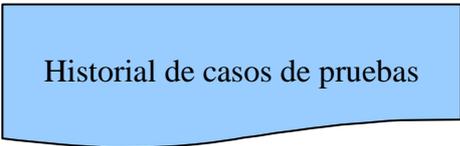
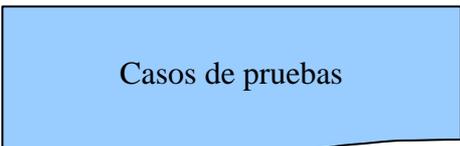
Entregables	Descripción
 <p>Actualizar lista</p>	<p>Esta es una actualización que se realiza durante todas las fases del proyecto</p>
 <p>Plan de pruebas</p>	<p>Este es una guía técnica para el manejo de proceso de pruebas. Aquí se define entradas, rendimientos, ambientes y procedimiento</p>
 <p>Historial de casos de pruebas</p>	<p>Este entregable enumera y proporciona un documento de procesos ya rectificadas y que se generan a lo largo del proyecto de Investigación.</p>
 <p>Casos de pruebas</p>	<p>Proporciona una apreciación importante, donde se da a conocer el estado del caso de prueba de acuerdo a la fecha de ejecución.</p>

GRÁFICO Nº 25

ENTREGABLES: CONSTRUCCIÓN/DESARROLLO

3.6.4.1.4. FASE 4.- ESTABILIZACIÓN

En esta fase de estabilización se realizan pruebas con el fin de determinar si las características de la solución están completas. El equipo enfoca sus esfuerzos en corregir los posibles errores que existan y en preparar la solución para luego lanzarla.

La fase de estabilización tiene como objetivo mejorar la calidad de la solución encontrando criterios de aceptación para la producción. Para mayor visualización **VER ANEXO N° 9.**

3.6.4.1.4.1. RESPONSABILIDAD DE LOS INVESTIGADORES

➤ ANÁLISIS DEL SOFTWARE PROTOTIPO

- ❖ Verificación del estado del proyecto investigativo
- ❖ Planificación de la ejecución del Sistema Prototipo.

➤ MISIÓN DEL PROYECTO

- ❖ Alcance del proyecto investigativo.
- ❖ Prioridad de errores.

➤ DESARROLLO

- ❖ Verificación de errores
- ❖ Optimización del script

➤ PRUEBAS

- ❖ Pruebas, detección y verificación de errores.

➤ **GESTIÓN DE ENTREGAS DE LA FASE ESTABILIZACIÓN**

- ❖ Soporte y proyección del proyecto
- ❖ Formación de soporte técnico del proyecto.

➤ **EXPERIENCIA DEL ADMINISTRADOR Y/O SOCIO**

- ❖ Estabilización en la formación de cada uno quienes conforman la Compañía “CITULASA”.

3.6.4.1.4.2. CIERRE DE LA FASE DE ESTABILIZACIÓN

El cierre de la fase de estabilización finaliza con la aprobación de cada una de las fases analizadas para ponerlas en práctica y dar solución a los problemas de la Institución y comprobar si se cumplen las expectativas en el desarrollo de software.

3.6.4.1.4.3. ENTREGABLES

Una vez definido y analizado la Fase de Estabilización se puede obtener resultados entregables la misma que servirá para el desenvolvimiento eficaz de esta Cuarta Fase. Y a continuación presentamos las siguientes:

- ❖ Verificación final
- ❖ Elementos de soporte del funcionamiento del software
- ❖ Resultado de las pruebas.

➤ **ENTREGABLES: ESTABILIZACIÓN**

Entregables	Descripción
<div data-bbox="352 427 812 568" style="border: 1px solid black; background-color: #ADD8E6; padding: 5px; text-align: center;"> Historial casos de pruebas </div>	Proporciona una apreciación importante, donde se da a conocer el estado del caso de prueba de acuerdo a la fecha de ejecución.
<div data-bbox="352 703 812 844" style="border: 1px solid black; background-color: #ADD8E6; padding: 5px; text-align: center;"> Actualizar lista de verificación de documentos </div>	Este entregable enumera y proporciona todos los documentos rectificadas y que generan a lo largo del proyecto investigativo.

GRÁFICO N° 26

ENTREGABLES: ESTABILIZACIÓN

3.6.4.1.5. FASE 5.- IMPLEMENTACIÓN / TRANSICIÓN

En esta fase los investigadores realizan una revisión del proyecto y la respectiva opinión sobre el nivel de satisfacción del socio con respecto a la solución del problema que tiene la Institución.

El objetivo principal de esta fase e dar solución a cada uno de los problemas con los que cuenta la Compañía, en el caso de estar completamente estable procedemos a su instalación. Para mayor visualización **VER ANEXO N° 10.**

3.6.4.1.5.1. RESPONSABILIDAD DE LOS INVESTIGADORES

➤ **ANÁLISIS DEL SOFTWARE PROTOTIPO**

- ❖ Expectativas del socio y administrativo
- ❖ Cierre del software prototipo.

➤ **GESTIÓN DEL PROYECTO**

- ❖ Alcance / Solución del software.
- ❖ Misión y estabilización del proyecto.

➤ **DESARROLLO**

- ❖ Solución a los problemas del prototipo
- ❖ Soporte de crecimiento y mejora.

➤ **PRUEBAS**

- ❖ Pruebas de excelencia.
- ❖ Solución de problemas.

➤ **GESTIÓN DE ENTREGAS DE LA FASE
IMPLEMENTACIÓN / TRANSICIÓN**

- ❖ Despliegue de datos del sistema
- ❖ Aprobación de cambios

➤ **EXPERIENCIA DEL ADMINISTRADOR Y/O SOCIO**

- ❖ Formación.
- ❖ Cumplimiento del cronograma.

**3.6.4.1.5.2. FINALIZACIÓN DE LA FASE DE IMPLEMENTACIÓN /
TRANSICIÓN**

- ❖ Cuando se realiza la aprobación por parte del cliente se realiza las tareas de cierres del proyecto.

3.6.4.1.5.3. ENTREGABLES

Una vez definido y analizado la última Fase de Implementación / Transición se puede obtener resultados entregables la misma que servirá para el desenvolvimiento eficaz de esta Quinta Fase. Y a continuación presentamos las siguientes:

➤ SOPORTE DEL SISTEMA PROTOTIPO

- ❖ Procedimientos y procesos
- ❖ Base de conocimientos e informes.

➤ INFORME DE FINALIZACIÓN DEL PROYECTO INVESTIGATIVO.

- ❖ Al culminar la última fase contaremos con la certificación de aprobación del sistema prototipo por parte del Director de Tesis.

➤ ENTREGABLES: IMPLEMENTACIÓN/TRANCISIÓN

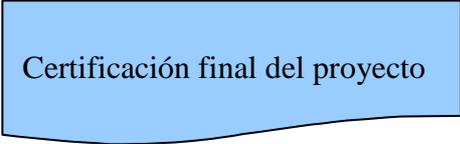
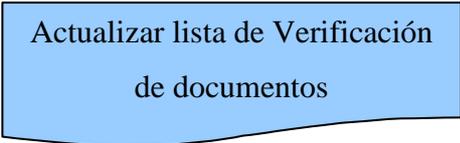
Entregables	Descripción
	Una vez que en el proyecto investigativo se ha concluido se obtendrá un certificado del director de Tesis.
	Enumera y proporciona una documentación detallada de documentos rectificadas y que se generan a lo largo de la depuración del Sistema prototipo.

GRÁFICO N° 27

ENTREGABLES: IMPLEMENTACIÓN/TRANCISIÓN

CONCLUSIONES Y RECOMENDACIONES GENERALES

Al terminar con este Análisis de Investigación podemos concluir y recomendar lo siguiente:

CONCLUSIONES

- ❖ Al estudiar minuciosamente las fases y parámetros de la Metodología MSF, se ha demostrado que es factible su manejo ya que puede ser aplicado a grandes y pequeños proyectos.
- ❖ Al analizar la Metodología MSF (MICROSOFT SOLUTIONS FRAMEWORKS), se demostró que una Metodología va dentro de un ciclo de vida, debido a que es el pilar fundamental en la elaboración de un proyecto previo a un análisis Metodológico, para luego proceder al respectivo estudio de las fases del ciclo de vida y de esta manera obtener resultados eficientes.
- ❖ El manejo adecuado del Sistema Prototipo de Administración ha permitido que la Compañía de Transporte Público Urbano Popular de Pasajeros “LATACUNGA S.A. CITULASA”, desarrolle de manera eficiente y rápida las diferentes actividades que realizan en la entidad antes mencionada.
- ❖ Al analizar la Metodología MSF (MICROSOFT SOLUTIONS FRAMEWORKS), con su respectivo Sistema Prototipo de Administración permitirá brindar mejor servicio a los socios y a la colectividad; de esta manera la Institución estará unánime al avance tecnológico.

- ❖ La organización de los procesos manuales a procesos automatizados son evidentes pero sin ninguna unión a otra Metodología que no sea MSF (MICROSOFT SOLUTIONS FRAMEWORKS).
- ❖ La realización de un Sistema Prototipo de Administración ha servido como conocimientos útiles y necesarios para la formación profesional de los investigadores y a la vez una ayuda para los administrativos y socios de la Institución.

RECOMENDACIONES

- ❖ En la Compañía de Transporte Público Urbano Popular de Pasajeros “LATACUNGA S.A. CITULASA”, debe existir capacitación constante a todos quienes conforman la Institución para que así tengan conocimiento del avance tecnológico que día a día va evolucionando.
- ❖ La idea propuesta como tema de tesis ha sido debidamente analizada con el fin de aprovechar la nueva metodología para brindar un mejor servicio a la colectividad; de tal manera es recomendable manipular correctamente y aprovechar sus beneficios.
- ❖ Cuando se utilice el Sistema Prototipo se recomienda que, las aplicaciones como juegos, videos, reproductor multimedia no estén siendo ejecutados para garantizar una mayor rapidez en la utilización del Sistema.
- ❖ Se recomienda a las Instituciones de Transporte Urbano y en general, opten por el desarrollo de Sistemas bajo la Metodología MSF (MICROSOFT SOLUTIONS FRAMEWORKS), ya que es confiable y se puede aplicar a grandes y pequeños proyectos.
- ❖ Consideramos factible establecer un espacio físico para poder trabajar satisfactoriamente con el Sistema Prototipo y dar un mantenimiento periódico y sistemático al software establecido.

