



1. Marco Teórico

1.1 Gestión del Servicio

Para empezar, antes de entender qué es la gestión del servicio, se tiene que entender qué son para que de esa forma entendamos cómo la gestión de los mismos puede ayudar a los proveedores de servicio a administrar y gestionarlos de una mejor manera. Un servicio es una forma de entregar valor a los clientes facilitándoles los resultados que quieren alcanzar sin que conozcan los costos y riesgos específicos de la labor brindada [6].

Los resultados que los clientes buscan son la razón por la que usan un servicio, entonces el valor de un servicio esta directamente relacionado en cuan fácil el resultado buscado pueda ser conseguido. Entonces, la Gestión del Servicio es lo que permite al proveedor de un servicio entender los servicios que esta brindando, y asegurarse que los servicios que esta brindando ayuden a conseguir con facilidad los resultados que los clientes buscan alcanzar, además de entender y administrar todos los costos y riesgos asociados con la labor brindada por dichos servicios. La gestión del servicio va mas allá que solo entregar un servicio, cada servicio, proceso o infraestructura tiene un ciclo de vida, por lo que la gestión del servicio considera desde el diseño, la transición, la operación y la mejora continua. Las entradas de la gestión del servicio son los recursos y capacidades que representan los activos del proveedor del servicio, mientras que las salidas son el servicio que el proveedor entrega a los clientes. La adopción de buenas prácticas ayudan a un proveedor de servicios a crear un sistema de gestión de servicios efectivo. Una buena práctica es simplemente hacer cosas que se han visto que funcionan y son efectivas. Las buenas prácticas vienen de diferentes fuentes, estándares (como ITIL, COBIT o CMMI), o conocimiento propietario de personas u organizaciones.

1.1.1 ¿Qué es ITIL?

ITIL es un marco de desarrollo que describe las Mejores Prácticas en la Gestión de Servicios de TI, basada en la filosofía de administración que tiene un *Gobierno Corporativo*, que llevándose acabo en el área de TI, se le conoce como *Gobierno de TI* [4]; estas prácticas, han venido a convertirse en uno de los enfoques más ampliamente aceptados en lo que se refiera a la Administración de Servicios de TI en el mundo [3], por lo que su adopción está encaminada a ayudar en el largo plazo.



1.1.2 Beneficios de ITIL

ITIL se enfoca en la continua medición y mejoramiento de la calidad del servicio entregado; desde el punto de vista tanto del negocio, cómo de la perspectiva del cliente. Este enfoque ha ayudado al éxito y a la proliferación de su uso, de tal forma que dentro de sus beneficios podemos mencionar:

- Incrementa la satisfacción del cliente con los servicios de TI.
- Mejora la disponibilidad del servicio, conduciendo directamente al aumento de beneficios empresariales y de ingresos.
- Ahorros financieros de la reducción de retrabajos, pérdida de tiempo y la mejora del uso y gestión de los recursos.
- Mejoramiento del tiempo de salida al mercado de nuevos productos y servicios.
- Mejor toma de decisiones y el riesgo optimizado.

1.1.3 Antecedentes de ITIL

ITIL fue publicado entre 1989 y 1995 por *La Oficina de Papelería de su Majestad* (HMSO por sus siglas en inglés) en el Reino Unido a favor de la *Agencia Central de Comunicaciones y Telecomunicaciones* (CCTA por sus siglas en inglés), que en nuestros días quedó incluida en la *Oficina de Gobierno de Comercio* (OGC por sus siglas en inglés). La versión inicial consistía de 31 libros que cubrían todos los aspectos provistos por TI, fue principalmente usada en el Reino Unido y los Países Bajos, caso contrario a la segunda versión que fue publicada como un conjunto de 7 libros (entre el 2000 y el 2004) más estrechamente conectados.

Esta segunda versión vino a ser universalmente usada en muchos países y organizaciones como base para la provisión eficaz de servicios de TI. En el 2007 la versión 2 de ITIL fue sustituida por una consolidada tercera versión consistiendo de 5 libros principales, que cubren efectivamente todo el ciclo de vida del servicio [3]; desde la definición inicial y el análisis de los requerimientos del negocio en la Estrategia y el Diseño del Servicio a través de la migración en el ambiente real con la Transición del Servicio, al servicio real y su mejoramiento en la Operación del Servicio y la Mejora Continua del Servicio.



Figura 1.1 El ciclo de vida del Servicio.

1.1.4 Ciclo de vida del Servicio

ITIL se basa en la idea de que tanto los servicios como todas las actividades del área de TI deben estar iniciadas por las necesidades y requerimientos del negocio, y que estos requerimientos junto con un conjunto de conceptos del negocio (políticas, estrategias, recursos y limitaciones) deben alimentar el primer estado del ciclo que es, la *Estrategia de Servicio*.

Una vez que están alineados los requerimientos conforme a el negocio, pasamos al siguiente estado, que se llama *Diseño del Servicio*, donde se tiene que producir una solución que tome en cuenta las ideas producidas en el estado anterior, además de también generar un Paquete de Diseño de Servicio (SDP por sus siglas en inglés) que contiene todo lo necesario para llevar el servicio a través de los restantes estados. El siguiente paso es recibir el SDP por el estado de *Transición del Servicio*, que se encarga de probar, validar y evaluar el servicio; en paralelo a todas estas operaciones que se han descrito, se va alimentando en cada estado el *Sistema de Gestión de Conocimiento del Servicio* (SKMS por sus siglas en inglés).

Finalmente, el servicio pasa al siguiente estado que se llama *Servicio en Operación*, en el cual el servicio es puesto en un ambiente real, siendo este el punto culminante u operativamente más importante. Ahora bien, en cualquier lugar posible de todo el proceso, el estado de Mejoramiento Continuo del



1. Marco Teórico



Servicio debe identificar las oportunidades que puede tener el ciclo, para mejorar alguna debilidad, o alguna falla dentro de todos los estados del ciclo de vida (ver la figura 1.2).

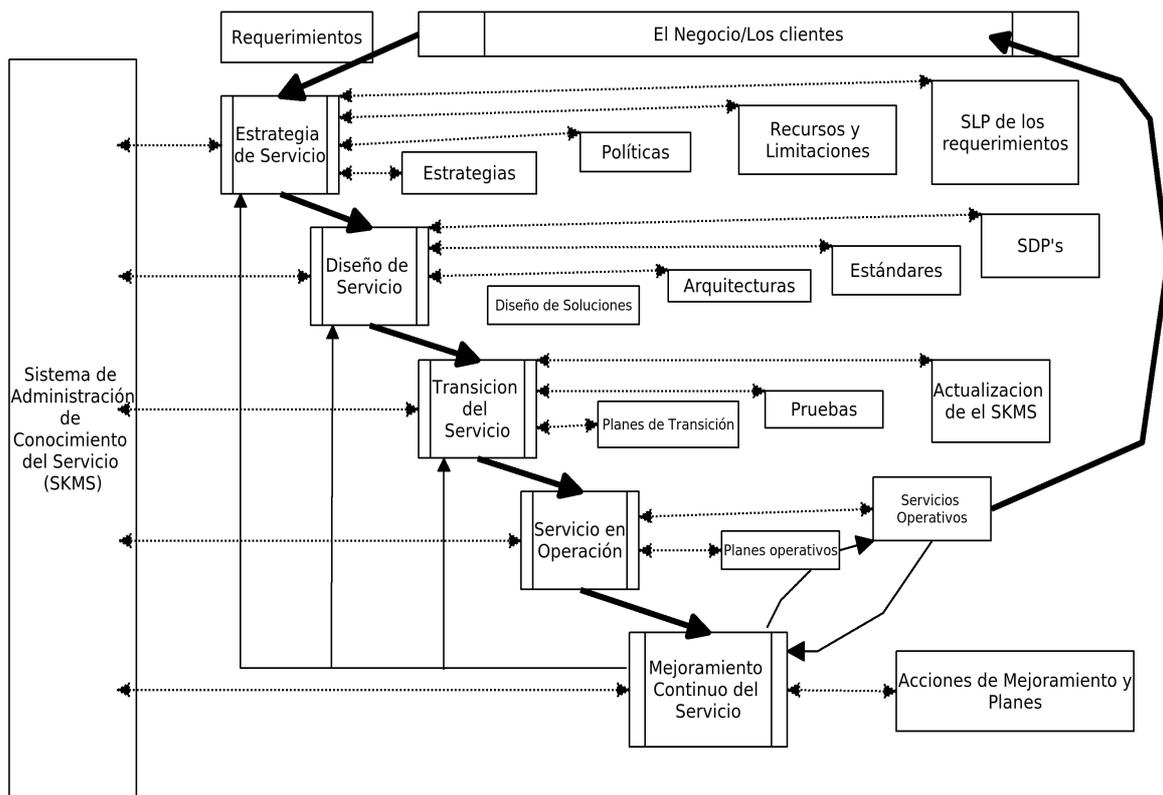


Figura 1.2 Diagrama de Estados de Ciclo de Vida de un Servicio, en ITIL

Si bien, lo anterior es sólo un esbozo del ciclo de vida de un servicio, no se trata de hablar extensamente sobre cada uno de los estados descritos en estas prácticas, sino más bien tomar como referencia el estado de *Servicio en Operación*, debido a que en el se encuentra la *Mesa de Ayuda*, que es precisamente el elemento que queremos desarrollar mediante una herramienta de software.



1.2 El Servicio en Operación

El objetivo del Servicio en Operación es el de ofrecer los niveles de servicio acordados, tanto para los usuarios como para los clientes, así como administrar las aplicaciones, la tecnología, y la infraestructura que soporta la entrega de estos servicios. Es únicamente durante este estado, en el que el servicio entrega valor real al negocio, y es responsabilidad del grupo encargado de este estado, de asegurarse que se cumpla esta expectativa en el servicio. Es importante tener un balance en las metas del Servicio en Operación, como lo son:

- Visión interna de TI contra la visión externa del negocio.
- Estabilidad contra capacidad de respuesta.
- Calidad del servicio contra costo del servicio.
- Actividades reactivas contra actividades pro activas.

El grupo encargado de este estado, debe mantener un balance, ya que enfocarse demasiado en uno de estos puntos puede resultar en un servicio pobre. Muchas organizaciones, encuentran de ayuda considerar la “operación saludable” del servicio, la cual se encarga de identificar los “signos vitales” que son críticos para la ejecución de las funciones vitales del negocio. Si éstos se encuentran dentro de los parámetros normales, el sistema o servicio está saludable. Esto produce una reducción en el costo de monitoreo, y permite al staff enfocarse en las áreas que pudieran llevar al éxito del servicio.

Existen dos procesos clave en el desarrollo de este estado y que a su vez establecen los criterios del desarrollo de nuestra herramienta:

- Proceso de Administración de Eventos.
- Proceso de Administración de Incidentes.

1.2.1 Proceso de Administración de Eventos

Un evento es un cambio de estado que tiene significancia para la administración o la configuración de un elemento perteneciente a TI o uno de sus servicios. Un evento puede indicar que algo no esta funcionando de manera correcta y en ese caso, llevarnos a una bitácora de incidentes. Los eventos también indican una actividad normal o de rutina, aunque en general detectan y generan notificaciones de que algo no está funcionando correctamente. La administración de eventos depende del monitoreo, aunque



en el fondo son distintos. Por ejemplo, la administración de eventos genera y detecta notificaciones, mientras que el monitoreo checa el estatus de los componentes aún cuando ningún evento este ocurriendo. Una vez que un evento ha sido detectado, éste puede llevarnos a un incidente o a un problema (es decir, escalar), o simplemente al registro del mismo en caso de que la información pueda tener utilidad. La respuesta a un evento puede ser automatizada o puede requerir una intervención manual, en el caso de que necesite una acción, entonces se debe disparar una alerta que avise al equipo de soporte.

1.2.2 Proceso de Administración de Incidentes

Un incidente es la interrupción no planeada de un servicio de TI o la reducción en la calidad de un servicio de TI. La falla en la configuración de un elemento de TI que todavía no ha impactado en el servicio, también es un incidente.

El objetivo de la Administración de incidentes es normalizar el servicio tan pronto sea posible, para así minimizar impactos adversos en la operación del negocio. Los incidentes son detectados usualmente por la *Administración de Eventos* o por los usuarios que contactan la *Mesa de Ayuda* y ahí deben ser categorizados para saber quién debe trabajar en ellos o para visualizar tendencias, así como para priorizar el incidente de acuerdo a la urgencia y al impacto que pueda dejar en el negocio. Si un incidente no puede ser resuelto rápidamente, éste deber ser escalado, lo cual significa que el incidente pasa al soporte técnico que tenga las habilidades apropiadas para resolver el problema; el escalar jerárquicamente el problema involucra niveles adecuados de administración. Una vez que el incidente ha sido investigado y diagnosticado y la resolución ha sido probada, la *Mesa de Ayuda* debe asegurarse que el usuario esté satisfecho antes de que el incidente sea cerrado. Una herramienta de Mesa de Ayuda, es esencial para el registro y administración de la información arrojada por los incidentes.

1.2.3 Proceso de Solicitud de Cumplimento

Una petición de servicio es una solicitud de un usuario por información, por asesoramiento de un cambio de estándar o para acceso a un servicio de TI. El propósito de una solicitud de cumplimiento es habilitar a los usuarios para solicitar y recibir servicios, para proveer a los usuarios y clientes acerca de servicios y procedimientos para obtenerlos, y para asistir con información



general, así como para poder recibir quejas y comentarios. Todas las peticiones deben ser registradas y rastreadas, el proceso debe incluir la apropiada aprobación antes de cumplir la solicitud.

1.2.4 Proceso de Administración de Problemas

Un problema es derivado de uno o más incidentes. La causa usualmente es desconocida al momento que el problema es registrado o creado y el proceso de administración del problema es responsable de una investigación más profunda. El objetivo principal de la Administración de Problemas, es evitar problemas y resultantes de incidentes que ocurran, impidiendo casos recurrentes y minimizando el impacto de los incidentes que no se pudieran prevenir. La Administración de los Problemas también incluye el diagnóstico causante de incidentes, determinando la resolución, y asegurándose que la resolución sea implementada. También registra la información sobre los problemas y su apropiada solución. Los problemas son categorizados de una forma similar a los incidentes, pero la meta es entender las causas, documentar las soluciones y hacer los cambios necesarios para resolver permanentemente los problemas. Las soluciones deben documentarse en una Base de Errores, la cual mejorará la eficiencia y efectividad de la Administración de Incidentes.

1.3 La Mesa de Servicio

Aclarando el nombre de *Mesa de Ayuda* tenemos que decir que como tal, el término no existe dentro de ITIL, se trata de un término que se desarrolló básicamente debido a la similitud con los servicios de asistencia. La disciplina a la que en realidad se refiere ITIL, es a la Mesa de Servicio, la cual es una función en la organización, y no un proceso.

La Mesa de Servicio provee el punto central de contacto para todos los usuarios de TI, usualmente registra y administra todos los incidentes, solicitudes de servicio, solicitudes de acceso y provee una interfaz para los procesos de Servicio en Operación y las actividades. Dentro de las responsabilidades que incluye tenemos:



1. Marco Teórico



- Registro de todas las solicitudes e incidentes, categorizándolos y priorizándolos.
- Primera línea de investigación y diagnóstico.
- Administración del ciclo de vida de incidentes y solicitudes, escalando cuando sea apropiado y cerrándolos cuando el usuario esté satisfecho.
- Mantener a los usuarios informados sobre el estado de los servicios, incidentes y solicitudes.
- Facilita la restauración del servicio operacional normal, con un mínimo de impacto para el cliente, de acuerdo al *Nivel de Acuerdo del Servicio* acordado.

Hay varias maneras de estructurar u organizar los tipos de mesa de ayuda que existen (ver tabla 1.1).

Tabla 1.1 Tipos de Mesa de Ayuda

Nombre	Descripción
Mesa de Ayuda local:	Físicamente cercana a los usuarios.
Mesa de Ayuda centralizada:	Permite que unos cuantos elementos del personal puedan hacerle frente a un mayor volumen de llamadas.
Servicio Virtual:	El staff se encuentra en locaciones distintas, sin embargo para el usuario parece un solo equipo.
Siguiendo al Sol:	Es la conexión de diversas Mesas de Ayuda que se encuentran en diferentes zonas horarias y que transfieren las llamadas a las locaciones donde el staff está trabajando, contribuyendo a tener una cobertura de 24 horas.

Ahora bien, el ámbito en que se da servicio en el INMEGEN es directamente con los usuarios, por lo que la implementación esta enfocada al tipo de Mesa de Ayuda Local.

1.3.1 Beneficios de la Mesa de Ayuda

Se puede decir que las ventajas de la Mesa de Ayuda pudieran abarcar bastantes ámbitos y ser bastante numerosas. Por mencionar algunos de los beneficios que ofrece una Mesa de Ayuda, tenemos lo siguiente:



- Crea una imagen positiva en la mente del cliente.
- La mejora del servicio al cliente, tiempos de respuestas mas rápidos y menores costos de tecnologías de la información.
- El servicio se lleva de una manera estructurada, permitiendo a mas de uno de los miembros del equipo participar activamente en la resolución.
- Reducción de tiempo y del esfuerzo usados por los usuarios para resolver sus problemas relacionados con TI, debido a la centralización del canal.
- Base de conocimientos para otras áreas, proveyendo información y estadísticas sobre los incidentes que tienen los usuarios.
- Aseguramiento de la entrega correcta de los servicios a los usuarios.

1.3.2 Software Libre y Desarrollo propio de Software

En la automatización de procedimientos una de las herramientas más útiles que se pueden implementar en cualquier industria pública o privada, son las *soluciones por software*; por lo que no es de extrañarse que es el primer recurso al que hemos de asistir y que ha sido considerado en el planteamiento del problema. En términos concisos, algunas ventajas [8] consideradas por el departamento de Servicios Computacionales del INMEGEN para el uso preferente de software libre para el desarrollo de sus aplicaciones, son:

- Libertad de uso y redistribución.
- Independencia Tecnológica.
- Fomento a la contratación de servicios de atención al cliente.
- Formatos estándar.
- Corrección rápida y eficiente de fallos.
- Sistemas sin puertas traseras y más seguros.
- Métodos simples y unificados de gestión de software.
- Económico.

Las anteriores ventajas ayudan a garantizar la seguridad e integridad de la información, cumpliendo con el propósito y con los criterios de austeridad, modernización tecnológica, transparencia, eficiencia y racionalidad que plantean los objetivos de las *políticas de uso de software, equipo y servicios de cómputo en el INMEGEN* [7].

En suma, la Dirección de Desarrollo Tecnológico es la área encargada de dimensionar las necesidades de software del INMEGEN y tomando en cuenta



los objetivos que tiene como dirección, así como también la utilidad del software libre, se asienta la elección preferente de éste tipo, no siendo la excepción para la implementación del Sistema de Mesa de Ayuda del INMEGEN.

La preferencia en el uso de software libre se justifica con los motivos mencionados; ventajas de la cual destaca su bajo costo, así como la independencia tecnológica que se obtiene. Si a ello se añade el desarrollo “en casa” de la herramienta, se obtienen aplicaciones a la medida, que son más fácilmente escalables, más fáciles de mantener, con un soporte técnico casi inmediato, y la mejor parte, a un costo muy por debajo del que tendría comprarlo hecho. Es por ello que ante cada problema que requiere la automatización de un proceso, Servicios Computacionales considera la posibilidad de desarrollar la aplicación desde cero, primeramente investigando los programas que existen en el mercado, con el fin de localizar las áreas de oportunidad y sobre todo, con el fin de poder desarrollar una aplicación amoldada a las propias necesidades que ofrezca ventajas competitivas sobre las demás aplicaciones.

1.4 Desarrollo de Software

El esfuerzo de planificación y organización requerido en el desarrollo de software no es diferente al esfuerzo que se usa en muchas otras actividades. Como un ejemplo muy simple se podría comparar el desarrollo de un producto de software contra el inicio de un viaje. La primera decisión que se toma para iniciar el viaje es decidir el destino; después, antes que el viaje comience se hacen planes con el fin de llegar al destino en un plazo razonable con un costo aceptable, con el conocimiento de la duración y las limitaciones que implicaría. Al hacer lo anterior, se podrían evaluar rutas alternativas para el destino, así como identificar y evaluar posibles riesgos o peligros. De forma similar, el desarrollo de una solución de software requiere el establecimiento de un destino o meta del producto; así como también se evalúan cuidadosamente alternativas de diseño para el sistema, identificando posibles riesgos, así como actividades que tienen que ser hechas en secuencia (así como otras tantas en paralelo). Entre más complejo sea el sistema, mayor planeación se necesitará.

En cuanto a la Ingeniería de Software, una variedad de modelos para organizar el desarrollo han sido implementados a lo largo del tiempo. Estos modelos comparten ciertas características, como por ejemplo: identifican las metas de *los interesados*, especifican las actividades clave a ser seguidas de acuerdo a una cierta secuencia, trabajan contra limitantes de tiempo, y son basadas en experiencia que ha sido adquirida de proyectos pasados.



Se puede decir que una amplia gama de estrategias para organizar los procesos de desarrollo de software han sido producidas las últimas cuatro décadas, estas estrategias representan los patrones que han resultado exitosos bajo diferentes condiciones, dichas estrategias son conocidas como *el ciclo de vida del desarrollo de software o modelos del proceso de desarrollo de software*. El término de modelo de proceso intenta hacer notar la idea de que no se refiere a una herramienta o una técnica, sino mas bien a la idea general del desarrollo completo de software. Las características que comparten los modelos para el desarrollo de software, reflejan características humanas universales, como son las limitantes de tiempo y económicas en las que el desarrollo opera [10].

1.4.1 Características de las Estrategias en el Desarrollo de Software

Los modelos del proceso de desarrollo de software comparten un número de características como son: un énfasis en el rol de Ingeniería de requerimientos, los requisitos de documentación, el involucramiento de los interesados, la administración del proyecto, la relación de los objetivos a las restricciones económicas, y la implícita o explícita adopción de mejores prácticas [10]. Para el mejor entendimiento de cada una de estas características, se describirá cada una de ellas:

- **Requerimientos claros:** La necesidad de una definición correcta sobre el problema, como una entrada en el proceso de desarrollo de software obliga a usar la Ingeniería de requerimientos como un proceso. De hecho, la necesidad de un problema bien definido, es lo que distingue la era en que las aproximaciones de *codifica-corrige* prevalecían. Otra cosa que permite la clara definición de un problema es el involucramiento de los usuarios; los problemas son reconocidos como problemas de los usuarios, independientemente de si son internos o externos a una organización.
- **Paradigma de una tarea:** Aunque la nomenclatura y detalles de la descomposición de una tarea difieren, el paradigma *análisis-diseño-codificación-pruebas- mantenimiento* aparece en la mayoría de los modelos. Sin embargo la relación entre las tareas varía sustancialmente, con tareas secuenciales en algunos modelos, iterativas en otros o funcionalmente independientes.
- **El rol de los interesados:** Este rol varía en un rango de distintos usuarios, desde los que deciden sobre los requerimientos del sistema hasta los usuarios del producto de software en fase de desarrollo. Este factor representa la dimensión humana del proceso y afecta a todas la fases del



proceso, independientemente del grado de automatización, los involucrados nunca están ausentes del proceso.

- Documentación de entregables: Son una característica esencial para todos los procesos de modelos de software. Hay mecanismos que ayudan a automatizar la documentación, como las herramientas de un software denominado CASE, lo cual puede ayudar en una reducción importante, aunque no la elimina por completo.
- Valor económico: Esta es una característica esencial del proceso de desarrollo software, debido a que no existe un mercado para productos de software económicamente inviables. Como consecuencia de ello, beneficios en la reducción de costos, es una de las mediciones más comunes de la producción de software.

1.4.2 Ciclo de Vida del Desarrollo de Software

Un modelo de ciclo de vida se encarga de definir las fases que ocurren durante el desarrollo de software, intenta determinar el orden de las etapas involucradas y los criterios de transición asociadas entre las etapas [11], en general el término ciclo de vida del software describe el desarrollo de software desde la fase inicial hasta la fase final. La ISO (International Organization for Standardization) en su norma 12207 define al ciclo de vida del software como un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando desde la definición hasta la finalización de su uso. El ciclo de vida básico de un software contiene los siguientes procedimientos:

Definición de objetivos: Definir el resultado del proyecto.

Análisis de requisitos: Recopilar, examinar y formular los requisitos del cliente examinando cualquier restricción que se pueda aplicar.

Diseño: Requisitos generales de la arquitectura de la aplicación.

Programación: Es la implementación en un lenguaje de programación para crear las funciones definidas en la etapa de diseño.



- Pruebas: Prueba 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.
- Documentación: Sirve para recopilar información necesaria para los usuarios del software y para desarrollos futuros.
- Mantenimiento: Todos los procedimientos correctivos y las actualizaciones secundarias del software.

1.4.3 Codifica-corrige

Cuando surgió la necesidad de adaptar sistemas informáticos a las exigencias del mercado, el programador se encargaba de realizar el levantamiento de requerimientos de quién necesitaba el programa o producto de software, con esos requerimientos comenzaba la tarea de codificar. Esta tarea no estaba administrada, supervisada o gestionada de ningún modo, por lo que se iba corrigiendo a medida que surgían errores, tanto lógicos provenientes de la misma codificación, como los de requerimientos solicitados por el cliente o usuario final.

La técnica de codifica-corrige se basaba en requerimientos ambiguos y sin especificaciones puntuales. Al no seguir normas para el proyecto, el cliente o el usuario solo impartían especificaciones muy generales del producto final. Se programa y se corrige, de esta forma se vuelve a programar sobre la misma marcha del proyecto. El ciclo de vida de estos proyectos finaliza cuando se satisfacen las especificaciones, no solo las primeras por las cuales nació la necesidad del programa, sino también todas aquellas que fueron surgiendo bajo la marcha.

Esta técnica tiene las ventajas de no gastar en análisis, planificación, gestión de recursos, documentación, etc., bien sabemos que es muy cómoda y muchas veces recomendable cuando el proyecto es muy pequeño, y es llevado por uno o dos programadores. Para sistemas grandes trae desventajas en lo que se refiere a costo de recursos, que siempre sería mayor del previsto, aumentando el tiempo de desarrollo y la disminuyendo la calidad del software [11].

1.4.4 Modelo en cascada

Es un proceso secuencial, en el que el progreso se considera que fluye constantemente hacia abajo a través de fases. El modelo en cascada fue uno de los primeros y es de los más ampliamente seguidos por las organizaciones. Este enfoque metodológico ordena rigurosamente las etapas del ciclo de vida del software, de tal forma que el inicio de cada etapa debe esperar a la finalización de la inmediatamente anterior. En la figura 1.3 podemos ver la estructura que compone esta metodología.

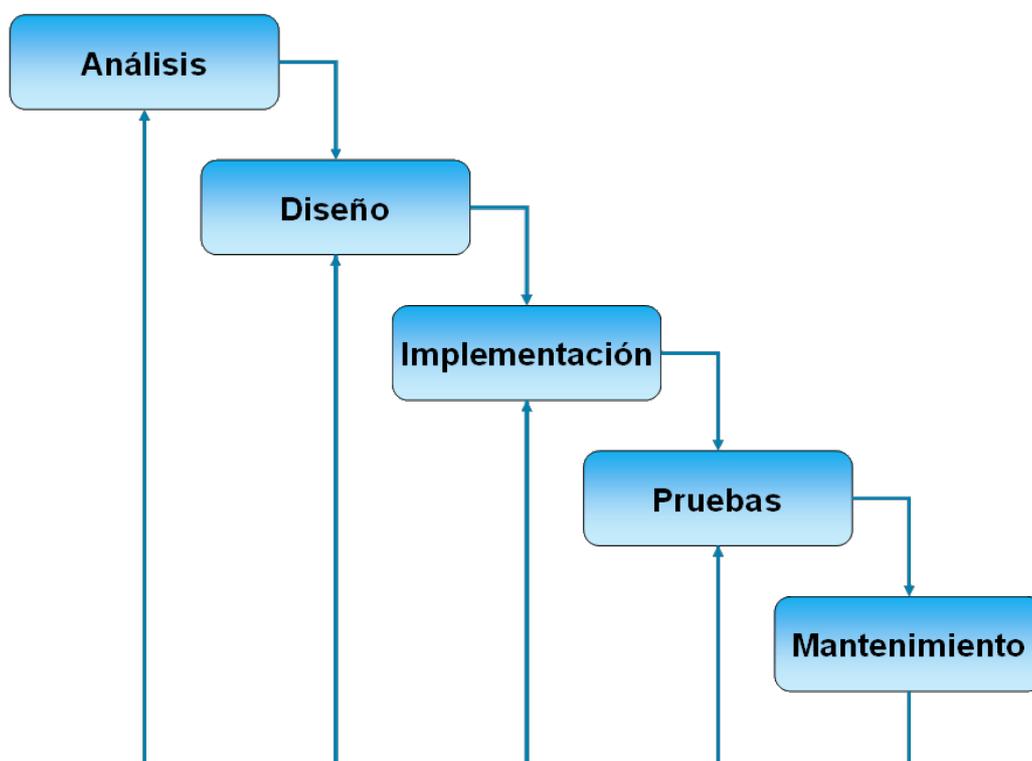


Figura 1.3 Ciclo de vida en cascada.

El modelo admite la posibilidad de hacer iteraciones; por ejemplo, al estar haciendo el mantenimiento se encuentra la necesidad de cambiar algo en el diseño, lo cual significaría que se tendrán que hacer los cambios necesarios en la codificación y se tendrán que realizar de nuevo las pruebas; es decir, que si se tiene que volver a una de las etapas anteriores al mantenimiento hay que recorrer de nuevo el resto de las etapas. Dentro de las ventajas que se tiene,



es que la planificación es sencilla y la calidad del producto resultante es bastante alta. Entre los inconvenientes se puede mencionar: el producto sólo se puede obtener hasta el final; si se han cometido errores en una fase se tiene que volver atrás; es necesario que se tengan bien definidos los requisitos, mientras que lo normal es que el cliente no tenga perfectamente definidas las especificaciones desde el principio e inclusive puede que surjan necesidades imprevistas.

Es adecuado para el tipo de proyectos en los que se dispone de todas las especificaciones desde el principio, particularmente en los que un tipo de producto que no es novedoso. Así también se recomienda para proyectos complejos que se entienden bien desde el principio.

1.5 Metodología para la Administración de Proyectos

Debido a la creciente aceptación y aplicación que está teniendo la Dirección de Proyectos, es de considerarse relevante la aplicación de conocimientos, procesos, habilidades, herramientas y técnicas adecuadas para alcanzar el éxito de un proyecto [9]. De esta forma, el Instituto de Administración de Proyectos (PMI por sus siglas en inglés) ha generado una norma, es decir ha generado un documento formal que describe las normas, métodos, procesos y prácticas establecidas, evolucionadas a partir de las prácticas reconocidas por profesionales dedicados a la dirección de proyectos, con miras a dirigir un proyecto con resultados éxitos; estas buenas prácticas, se encuentran contenidas en el PMBOK (Project Management Body of Knowledge) y aunque el conocimiento descrito debe aplicarse siempre de la misma manera en todos los proyectos, en el desarrollo del proyecto del *Sistema de Mesa de Ayuda*, gran parte de las delimitaciones las determinó el departamento de Servicios Computacionales, en base a la envergadura del proyecto.

En el PMBOK, se describe la naturaleza de los procesos de dirección de proyectos en cinco categorías conocidas como *Grupos de Procesos de la Dirección de Proyectos*:

- Grupo del Proceso de Iniciación.
- Grupo del Proceso de Planificación.
- Grupo del Proceso de Ejecución.
- Grupo del Proceso de Seguimiento y Control.
- Grupo del Proceso de Cierre.

Estos procesos de la dirección de proyectos llevados a la práctica son elementos que se superponen e interactúan en formas diversas a lo largo del proyecto. La naturaleza integradora de la dirección de proyectos, requiere que el Grupo de Proceso de Seguimiento y Control interactúe con los otros grupos de procesos; además dado que el esfuerzo requerido en un proyecto es finito, el Grupo de Proceso de Iniciación y el Grupo de Proceso de Cierre, se encargan de comenzar y finalizar el proyecto respectivamente. Esta forma de representar la interacción de los Grupos de Procesos, los podemos apreciar en la figura 1.4:

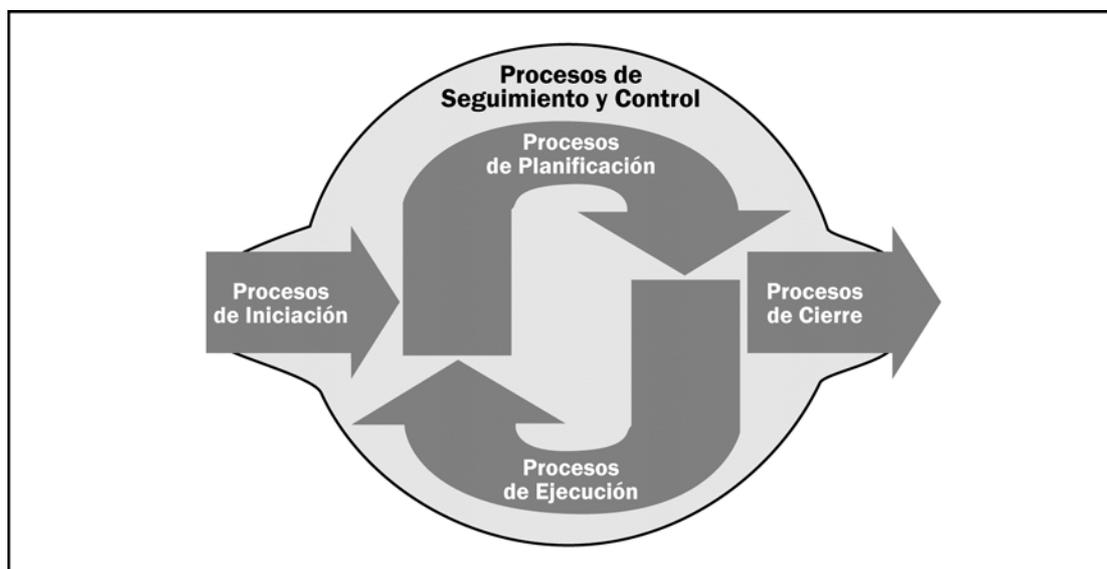


Figura 1.4 Grupos de Procesos de la Dirección de Proyectos.

Continuando con la metodología a la que hace referencia el PMBOK, así como se ha dividido en 5 grupos de procesos, el número total de procesos que constituyen la metodología corresponden a 42 procesos, que a su vez, se agrupan en 9 áreas de conocimiento. De esta forma la correspondencia entre los procesos y grupos de procesos la podemos ver en la figura 1.5.

Hay que aclarar que los grupos de procesos individuales que los constituyen a menudo se repiten antes de concluir el proyecto, los procesos constitutivos pueden representar interacciones dentro de un grupo de procesos y entre grupos de procesos, estas interacciones, cuya naturaleza varía de un proyecto a otro pueden realizarse o no en un orden determinado. Por otro lado, también cabe aclarar que los grupos de procesos no son fases del proyecto, cuando los proyectos son complejos o de gran tamaño, se dividen en subproyectos o fases



1. Marco Teórico

diferenciadas; como por ejemplo: estudio de viabilidad, diseño, desarrollo conceptual, prototipo, etc., por lo general, todos los grupos de procesos se repetirán en cada fase o subproyecto.

Knowledge Areas	Project Management Process Groups				
	Initiating Process Group	Planning Process Group	Executing Process Group	Monitoring & Controlling Process Group	Closing Process Group
4. Project Integration Management	4.1 Develop Project Charter	4.2 Develop Project Management Plan	4.3 Direct and Manage Project Execution	4.4 Monitor and Control Project Work 4.5 Perform Integrated Change Control	4.6 Close Project or Phase
5. Project Scope Management		5.1 Collect Requirements 5.2 Define Scope 5.3 Create WBS		5.4 Verify Scope 5.5 Control Scope	
6. Project Time Management		6.1 Define Activities 6.2 Sequence Activities 6.3 Estimate Activity Resources 6.4 Estimate Activity Durations 6.5 Develop Schedule		6.6 Control Schedule	
7. Project Cost Management		7.1 Estimate Costs 7.2 Determine Budget		7.3 Control Costs	
8. Project Quality Management		8.1 Plan Quality	8.2 Perform Quality Assurance	8.3 Perform Quality Control	
9. Project Human Resource Management		9.1 Develop Human Resource Plan	9.2 Acquire Project Team 9.3 Develop Project Team 9.4 Manage Project Team		
10. Project Communications Management	10.1 Identify Stakeholders	10.2 Plan Communications	10.3 Distribute Information 10.4 Manage Stakeholder Expectations	10.5 Report Performance	
11. Project Risk Management		11.1 Plan Risk Management 11.2 Identify Risks 11.3 Perform Qualitative Risk Analysis 11.4 Perform Quantitative Risk Analysis 11.5 Plan Risk Responses		11.6 Monitor and Control Risks	
12. Project Procurement Management		12.1 Plan Procurements	12.2 Conduct Procurements	12.3 Administer Procurements	12.4 Close Procurements

Figura 1.5 Relación entre Grupos de Procesos, y Áreas de Conocimiento de la Dirección de Proyectos.



1.5.1 Definición de proyecto

Empezando por mencionar la definición de un proyecto se dice que es un esfuerzo temporal que se lleva a cabo para crear un producto, servicio o resultado único. La naturaleza temporal de un proyecto indica un principio y un final definidos, alcanzando el final cuando los objetivos se cumplen o no pueden ser cumplidos, o cuando ya no existe la necesidad que dio origen al proyecto [9]. Como ejemplos de proyectos tenemos:

- Desarrollar un nuevo procedimiento de negocio.
- Desarrollar o adquirir un sistema de información nuevo o modificado.
- Desarrollar un nuevo producto o servicio.
- Desarrollar un cambio en el estilo de una organización.

1.5.2 La Dirección de Proyectos

Por otro lado, la Dirección de Proyectos se refiere a la aplicación de conocimientos, habilidades, herramientas y técnicas a las actividades que se deben llevar a cabo para cumplir con los requisitos del mismo [9]. Estos conocimientos, habilidades y técnicas se encuentran contenidos en el PMBOK, mediante la aplicación adecuada de los 42 procesos de la Dirección de Proyectos, que agrupados conforman 5 grupos de procesos que son:



1. Marco Teórico



Grupo del Proceso	Descripción
Iniciación	Contiene los procesos relacionados con obtención de la autorización para comenzar dicho proyecto o fase.
Planificación	Los procesos relacionados con establecer el alcance del proyecto, así como la definición del curso de acción para alcanzar los objetivos.
Ejecución	Aquellos procesos realizados para completar el trabajo definido en el plan para la dirección del proyecto.
Seguimiento y Control	Procesos requeridos para dar seguimiento, analizar y regular el progreso y desempeño del proyecto. Así como la identificación de áreas en las que el plan requiera cambios.
Cierre	Procesos realizados para finalizar todas las actividades, con el fin de cerrar el proyecto.

Tabla 1.2 Grupos de Procesos.