

UNIDAD II

PLANEACION DE UN PROYECTO DE PROGRAMACION

Contenido:

- 2.1 Introducción
- 2.2 Definición del Problema
- 2.3 Recolección de Requisitos
- 2.4 Desarrollo de Estrategias de Solución
- 2.5 Analisis de Factibilidad
- 2.6 Especificación de la Definición del Sistema

2.1. INTRODUCCION

Inquietudes del Cliente
en la primera entrevista:

- 1.- Dar a conocer su problema
- 2.- Cómo cuesta una solución en software
- 3.- Cuánto tiempo se lleva.

Para responder a los puntos 2 y 3 se requiere un perfecto entendimiento del problema (punto 1), de definir los objetivos de la solución y hacer una buena *PLANEACION*.

Durante todo el ciclo de vida de un proyecto de software se debe aplicar una tarea llamada Administración o Gestión del Proyecto de Software. Dentro de las actividades que se llevan a cabo en esta tarea de Administración del Proyecto se encuentra justamente la PLANEACION.

Administración
del Proyecto de
Software

- ***Planeación del proyecto***

Definición

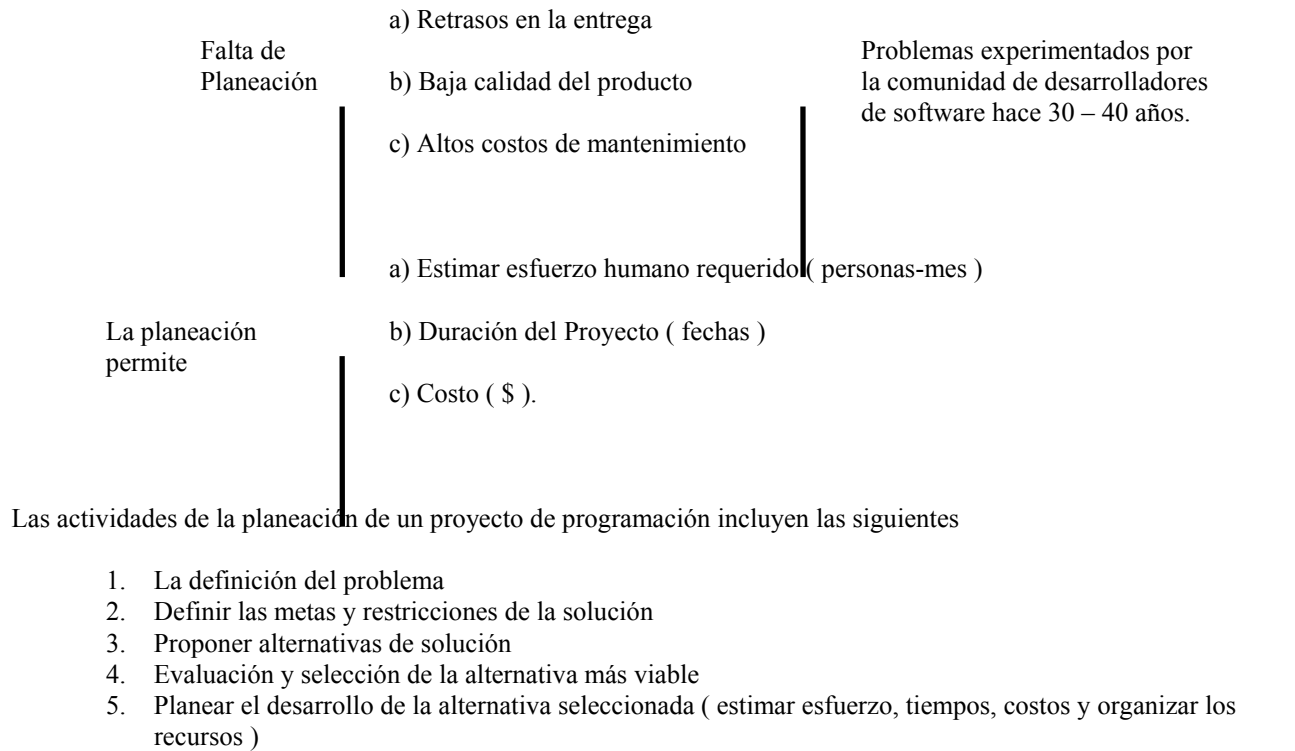
Estimación

- Asignación de recursos
- Seguimiento de avances
- Seguimiento y control de riesgos
- Control del presupuesto
- etc.

El objetivo de la planeación del proyecto de software es proporcionar un marco de trabajo que permita al director del proyecto hacer estimaciones razonables de:

- Recursos
- Costos
- Planificación temporal

Consecuencias



Muchos denominan colectivamente a las primeras 4 actividades como **Ingeniería de Sistemas de Computadora** y la consideran como una etapa independiente previa a la tarea de planeación.

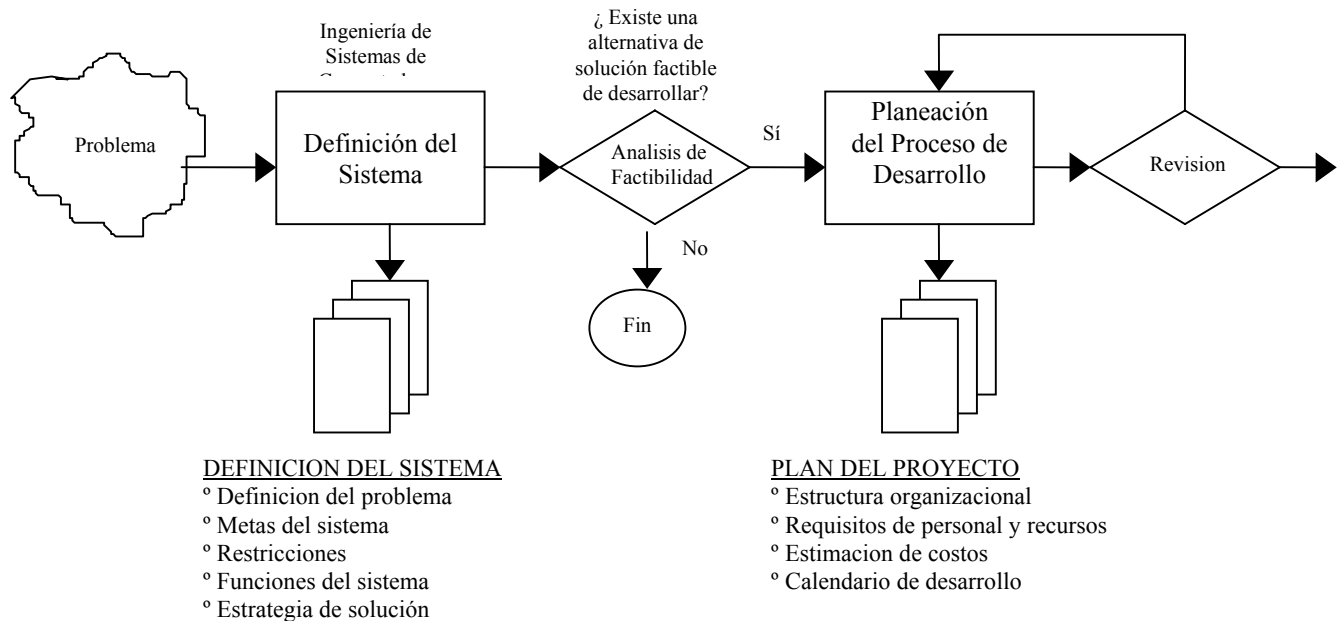


Fig.- Planeación de un proyecto de software

2.2. DEFINICION DEL PROBLEMA

Antes de poder planificar un proyecto se deberían establecer sus objetivos y su ámbito, se deberían considerar soluciones alternativas e identificar las dificultades técnicas y de gestión. Sin esta información es imposible definir estimaciones razonables de costo, riesgo, esfuerzo, etc.

El primer paso en la planeación del proyecto es DEFINIR EL PROBLEMA.

Para tal fin el cliente o usuario y el desarrollador se reúnen para definir los objetivos del proyecto y su ámbito. En la entrevista el desarrollador puede iniciar preguntando:

- ¿ Qué problema resolverá la solución ¿
- ¿ De quién ha surgido la petición de este trabajo ¿
- ¿ Quién va a utilizar la solución ¿
- ¿ Puede mostrarme o describirme el entorno en el que se utilizará la solución ¿
- ¿ Hay alguna limitación especial de rendimiento que vaya a afectar la forma en que se enfoque la solución?
- ¿ La información que me proporciona es “oficial” ¿
- ¿ Hay alguien más que pueda proporcionar información adicional ¿

Esta serie de preguntas puede ayudar para que el cliente defina su problema, exponga sus ideas y la percepción que tiene de la solución en software.

Las técnicas usuales que pueden aplicarse para Definir el Problema son:

1. Las entrevistas con el cliente y las personas involucradas en el proceso.
2. La observación del proceso actual.
3. La identificación de tareas problemáticas.

Para documentar la Definición el Problema primero se debe preparar brevemente, en la terminología del cliente

1. Un enunciado del problema que se solucionará

Posteriormente se debe preparar un enunciado definitivo donde se incluya:

1. Una descripción de la situación actual
Puede hacerse mediante una descripción en lenguaje natural del proceso básico que se estudiará y la forma en que actualmente se lleva a cabo.
2. Las metas que debe lograr el sistema nuevo
Los objetivos definen las metas generales del proyecto sin considerar cómo se conseguirán.
3. Las restricciones impuestas en la solución, las cuales pueden ser de tipo:
 - Técnicas (ej. sw para un sistema operativo determinado, o que deba usarse un lenguaje específico)
 - Tiempo (cuando el cliente impone una fecha para la entrega del software)
 - Económicas (limitantes de presupuesto)
 - Equipo (limitantes en las que el sw debe interactuar con un equipo de características específicas)
 - Operativas (limitantes acerca del ambiente en que va a operar, tal como el tipo de usuario, etc)

Un ejemplo simplificado de Definición del Problema es el siguiente:

- 1) Problema en terminología del cliente:

Hay un problema en el área de producto terminado, concretamente con la forma en que se acomodan las cajas de producto para su almacenamiento. Hay tres personas trabajando todo el día acomodando cajas en su correspondiente compartimiento. Las cajas se las van aventando hasta que le llega al acomodador, maltratando el producto. A veces cuando se termina su turno no han acomodado ni la mitad de las cajas. Se necesita dar un mejor manejo a las cajas, por que si se empiezan a maltratar desde antes de salir de la fábrica, hay que imaginar como llegarán al consumidor final después de todo el manejo que deben pasar.

Se piensa que instalando una cinta transportadora las cajas pueden viajar hasta una estación de clasificación donde sean identificadas y ordenadas en uno de los compartimentos al final de la cinta. Las cajas pueden llevar una etiqueta con el código de barras del producto para que puedan ser identificadas rápidamente.

2) Descripción de la situación actual:

En el área de producto terminado de la empresa X las cajas de producto salen de la línea de producción y llegan aquí para almacenarlas. Las cajas se deben acomodar en un compartimento específico que depende de su código de producto. Como es una bodega donde la distancia del punto donde llegan las cajas a los compartimentos es de unos 15 metros están tres personas, una en el punto donde llegan las cajas, otro a la mitad del camino y el último en los compartimentos. El primero toma una caja lee el código de producto, se lo comunica al segundo al mismo tiempo en que se la avienta; el segundo recibe la caja, busca en una lista el código y le comunica al tercero en cuál compartimento debe ir al tiempo que le avienta la caja. El tercero recibe la caja y la acomoda en el compartimento que le indicaron.

3) Metas del sistema

- *Prescindir del elemento humano en este proceso.*
- *Las cajas no deberán recibir maltrato al acomodarlas.*
- *Las cajas deben estar acomodadas el mismo día en que llegaron.*

4) Restricciones

- Operativa: Llevar a cabo el proceso sin emplear personal.
- Tiempo: Se requiere el nuevo sistema a lo más en 10 meses, cuando la época pre-navideña empiece.

2.3 RECOLECCION DE REQUISITOS

Al principio de un proyecto de software las cosas siempre están un poco borrosas. Se han definido una necesidad y se han enunciado las metas y objetivos generales, pero todavía no se ha definido adecuadamente las funciones y rendimiento que debe proporcionar el nuevo sistema basado en software.

Es necesario pues recolectar una serie de requisitos (o requerimientos) de alto nivel que permitan posteriormente proponer alternativas de solución.

Un requisito o requerimiento es una característica con la que debe contar el sistema nuevo y puede consistir en una forma de capturar o procesar los datos, de producir la información , de controlar una actividad o de apoyar una decisión.

Para lograr recolectar los requisitos es necesario identificar:

- los datos primarios
- las funciones y
- el comportamiento

que caracterizan al problema, y más importante es tratar de definirlos cuantitativamente.

Mediante las técnicas de entrevistas y la observación del proceso actual llevado a cabo en la Definición del Problema tratamos de responder a las preguntas:

- ¿ Qué se está haciendo ?
- ¿ Cómo se está haciendo ?

Tales respuestas permiten identificar las FUNCIONES del sistema.

Ahora es necesario delimitar cuantitativamente el COMPORTAMIENTO y RENDIMIENTO del sistema, apoyandose en preguntas como:

- ¿ Qué tan frecuente ocurre ?
- ¿ Qué tan grande es la cantidad de transacciones ?
- ¿ En qué tiempo límite se debe proporcionar la respuesta ¿
- ¿ Qué margen de errores / fallas es aceptable ¿
- ¿ Cuánto tiempo se consume actualmente ¿

Por ejemplo, no es suficiente con decir que el software para el control de un robot en un sistema automático de fabricación “responderá rápidamente si una bandeja de piezas está vacía”. El ingeniero de sistemas debe definir: 1) qué es lo que le indica al robot que la bandeja está vacía, 2) los límites precisos de tiempo (segundos quizá) en los que se espera que responda el software, 3) qué forma debe adquirir la respuesta.

Por último para conocer la naturaleza de los DATOS que actualmente se mueven en el sistema se debe:

- Recabar documentos fuente.
- Recabar documentos de la información que se produce actualmente.

El ingeniero de sistemas no pregunta al cliente cómo se debe hacer la tarea; más bien pregunta qué necesita.

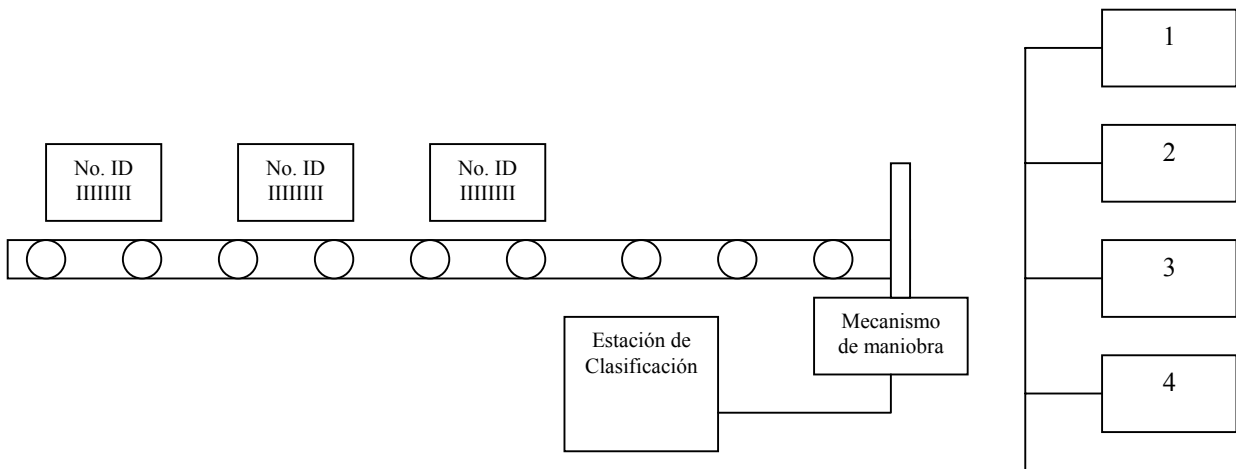
La documentación de los requisitos puede ir desde un simple listado de las funciones, comportamiento y datos primarios requeridos, hasta la elaboración de algunos diagramas desde esquemas sencillos hasta más formales tales como: Diagramas de arquitectura del sistema o Diagramas de flujo de datos.

Ejemplo simplificado de la Recolección de Requisitos:

1) Funciones a proporcionar:

- Leer el código del producto que viene impreso en la caja.
- Buscar el código en el listado

- Determinar en que compartimento debe acomodarse.
 - Acomodar la caja en el compartimento.
 - Registrar el destino de cada caja (a qué compartimento se envió).
- 2) Comportamiento del sistema:
- Si un compartimento se llena se debe hacer lo siguiente....
 - Si una caja no trae código de identificación debe hacerse lo siguiente....
- 3) Limites de Rendimiento del sistema:
- Cantidad de compartimentos: 6
 - Cantidad de código de producto en el listado: 40
 - Cantidad de cajas a acomodar al día/hora/minuto: ...
 - Velocidad de la cinta transportadora: ...
 - Distancia de la estación de clasificación a los compartimentos:
- 4) Datos primarios
- Tabla de Ubicación de Cajas, es el listado que indica el compartimento que corresponde a cada código de producto.
 - Reporte de Destino de Cajas, es un listado de las cajas clasificadas durante el día.
- 5) Se incluyen como anexos:
- Copia de una hoja del Listado de Ubicación de Cajas.
 - Croquis del sistema propuesto por el cliente.



2.4 DESARROLLO DE UNA ESTRATEGIA DE SOLUCION.

El siguiente paso en la planeación de un proyecto de programación es determinar lo apropiado de una solución computacional.

Una solución computacional
es apropiada si. . .

- 1.- Se acepta en costo.
- 2.- Se acepta política y socialmente.

Para aceptarse en costo
el nuevo sistema debe
proporcionar . . .

- 1.- Misma información que el sistema antiguo con menos tiempo y personal.
- 2.- Proporcionar información antes inaccesible.

Un sistema que se acepta en costo pero desplaza a muchos trabajadores puede no ser aceptada política y socialmente por el usuario (por ejemplo: por problemas sindicales, rechazo al cambio, etc.).

Algunas veces los sistemas computacionales se construyen para aliviar un sintoma y no la causa primaria del problema. Esto ocurre cuando:

1. El problema se entiende, pero no puede resolverse debido a circunstancias económicas, políticas o sociales.
2. Cuando el cliente no es capaz de comunicar el problema real.
3. Cuando el desarrollador no entiende la explicación del cliente sobre el problema.

Las estrategias de solución se deben crear sin considerar aspectos de factibilidad.

Las estrategias de solución no se formulan describiendo detalladamente en que consiste la solución, sino en describir las características generales que deberán tener la posible solución.

Para formular una estrategia de solución debemos partir del hecho de que todos los sistemas basados en computadora hacen uso de varios elementos del sistema:

- Software.- Programas de computadora, estructuras de datos y su documentación que sirven para hacer efectivo el método, procedimiento o control que se requiera.
- Hardware.- Dispositivos electrónicos que proporcionan capacidad de cálculo y dispositivos electromecánicos (ej. Sensores, motores, bombas) que proporcionan una función externa.
- Personas.- Usuarios y operadores del hardware y el software.
- Base de datos.- Información organizada a la que se accede mediante el software.
- Documentación.- Manuales, formularios y otra información descriptiva que indica el uso y/o operación del sistema.

El ingeniero de sistemas puede obtener diversas estrategias de solución al hacer asignaciones diferentes de las funciones del sistema a cada uno de los elementos de un sistema basado en computadora.

Es decir, se organizan las funciones del sistema de tal manera que unas se asignan al software, otras al hardware, otras a las personas, etc. De forma que se pueda obtener la función y comportamiento esperados.

Cada asignación diferente representa una estrategia de solución.

Al documentar cada estrategia de solución se debe incluir:

- Una descripción general de las características del producto
- Las prioridades de cada una de las características.
- Qué nuevas capacidades proporcionará el sistema.
- Qué procesos se van a preservar y cuales se van a mejorar.
- Qué aspectos de seguridad se manejarán.
- Cuánto personal se requerirá para operar el sistema.
- Ampliaciones posteriormente factibles.

Un ejemplo simplificado de las estrategias de solución formuladas para el problema del almacén de producto terminado sería:

Estrategía 1: *Un operador estará situado en la estación de clasificación. El lee las cajas y las coloca en el compartimento apropiado.*

Observe que esta es una solución puramente manual pero efectiva. Aquí la asignación de las funciones del sistema se han hecho a las *Personas*. Puede ser necesaria alguna *Documentación* que relacione el número de identificación con el compartimento.

Estrategía 2: *Se coloca un lector de código de barras y un controlador automático en la estación de clasificación. El código de barras se pasa al controlador programable que controla un mecanismo de separación. El separador desliza la caja al compartimento apropiado.*

Observe que esta es una solución totalmente automática, en la que se han asignado las funciones del sistema al *Hardware* (lector de código de barras, control programable, mecanismo de separación), al *Software* (para decodificar el código de barras y el control programable) y a una *Base de Datos* (una tabla que relaciona los códigos de identificación con los compartimentos). Es probable que cada uno de ellos deba llevar sus propios instructivos como *Documentación*.

Estrategía 3: *Se coloca un lector de código de barras y un controlador programable en la estación de clasificación.. La salida del código de barras se pasa a un brazo de robot que toma la caja y la lleva al compartimento correspondiente.*

Observe que esta también es una solución totalmente automática, en la que se han asignado las funciones del sistema al *Hardware*, al *Software*, a *Base de Datos* y a la *Documentación*.

2.5 ANALISIS DE FACTIBILIDAD

El siguiente paso en la planeación del proyecto de software es determinar la factibilidad de las estrategias de solución.

La factibilidad de cada estrategia de solución propuesta se puede establecer por el análisis de restricciones de la solución. Estas establecen las fronteras del espacio de soluciones. Una estrategia de solución es factible si las metas y restricciones del proyecto se pueden satisfacer dentro de las restricciones de tiempo disponible, recursos y tecnología por medio de esa estrategia.

No será necesario llevar a cabo un estudio de factibilidad para sistemas en los que la justificación económica es obvia, el riesgo técnico es bajo, se esperan pocos problemas legales y no existe una alternativa razonable.

Los aspectos de factibilidad a evaluar de cada estrategia son:

1) **FACTIBILIDAD TECNICA**

Es frecuentemente el el área más difícil de evaluar en esta etapa del proceso de desarrollo del sistema. Debido a que los objetivos, las funciones y el rendimiento son de alguna manera confusos.

- ¿ Se puede desarrollar el proyecto con el equipo, personal y tecnología disponible ?
- ¿ Se requiere nueva tecnología ?
- ¿ Se requiere equipo nuevo o instalaciones especiales ?
- ¿ Puede el elemento del sistema ser diseñado de tal forma que las funciones y el rendimiento necesarios se consigan dentro de las restricciones determinadas en el análisis ¿
- ¿ Hay personal calificado para desarrollar el elemento del sistema en cuestión ¿

2) **FACTIBILIDAD ECONOMICA**

Entre la información más relevante que contiene el estudio de viabilidad se encuentra el análisis de costo – beneficio. Este análisis señala los costos de desarrollo del proyecto y los contrasta con los beneficios tangibles (los que se miden directamente en dinero) e intangibles del sistema.

- ¿Existen beneficios en la creación del sistema ?
- ¿El posible costo del sistema es aceptable ?
- ¿El costo es tan alto que el proyecto no es factible ?

Posibles beneficios de un sistema de información:

- Reducción o eliminación de costos
- Reducción de errores
- Aumento de la confiabilidad
- Aumento de la velocidad de la actividad
- Mejoras en el control de la actividad

Posibles costos de un sistema de información:

Costos de Preparación

- Costo de la compra o alquiler de equipo
- Costo de la instalación de equipo
- Costo de acondicionamiento del lugar destinado al equipo (aire acondicionado, seguridad)
- Costo de los gestores y personal encargados.

Costos de Puesta a Punto

- Costo del Software
- Costo del personal dedicado a la puesta a punto
- Costo de la contratación de personal
- Costo de la gestión requerida para la puesta a punto

Costos del Proyecto

- Costo de la compra de software de aplicación
- Costo de modificaciones del software para ajustarse a sistemas locales
- Costo del personal del desarrollo interno de aplicaciones
- Costo de la formación de personal en el uso de aplicaciones
- Costo de los procedimientos de recolección de datos
- Costo de la preparación de la documentación

- Costo de la gestión del desarrollo

Costos Continuos

- Costos de mantenimiento del sistema
- Costo de los alquileres
- Costo de la depreciación del hardware

3) FACTIBILIDAD OPERATIVA

- ¿ Se utilizará el sistema si se desarrolla y se instala ?
- ¿ Habrá resistencia del usuario ?
- ¿ Requiere conocimientos especiales del usuario ? ¿ los tiene ?

Cuando se recomienda una estrategia de solución, es importante documentar las razones por las que se rechazan las otras estrategias. Esto da justificación a la estrategia recomendada, y puede prevenir revisiones equivocadas en fechas posteriores.

El estudio debe producir una decisión de “seguir/no seguir” .

El estudio de factibilidad puede documentarse en un informe separado de los otros documentos importantes de gestión e incluirse como apéndice en la Especificación del Sistema.

FORMATO DEL ESTUDIO DE FACTIBILIDAD

- I. Introducción
 - A. Declaración del problema
 - B. Entorno de implementación
 - C. Restricciones
- II. Resumen y recomendaciones de gestión
 - A. Hallazgos importantes
 - B. Comentarios
 - C. Recomendaciones
- III. Alternativas
 - A. Configuraciones del sistema alternativas
 - B. Criterio utilizado para la selección del enfoque definitivo
- IV. Descripción del sistema
 - A. Declaración resumida del ámbito
 - B. Viabilidad de los elementos asignados
- V. Análisis de Costo Beneficio
- VI. Evaluación del Riesgo Técnico
- VII. Consideraciones Legales

2.6 ESPECIFICACION DE LA DEFINICION DEL SISTEMA

Cada paso del proceso de ingeniería del software debe producir algo que se pueda entregar, que se pueda revisar y que pueda servir de base para los pasos posteriores.

La Definición del Sistema es un documento que, por lo regular, se expresa en lenguaje natural, y que puede contener cuadros, gráficas o diagramas de distintos estilos; la notación exacta empleada depende mucho del área del problema.

FORMATO DE LA DEFINICION DEL SISTEMA

- Sección 1: Definición del problema
 Sección 2: Justificación del sistema

Sección 3:	Metas del sistema y del proyecto
Sección 4:	Restricciones del sistema y del proyecto
Sección 5:	Funciones que se proporcionarán.
Sección 6:	Características del usuario.
Sección 7:	Ambientes de desarrollo/ operación / mantenimiento.
Sección 8:	Estrategia de solución.
Sección 9:	Prioridades para las características del sistema.
Sección 10:	Criterios de aceptación del sistema
Sección 11:	Fuentes de información
Sección 12:	Glosario de Terminos.