

**Ministerio de Planificación y Cooperación
División de Planificación, Estudios e Inversión
Departamento de Inversiones**

**Metodología de Proyectos
Informáticos**

<u>1</u>	<u>Introducción</u>	4
<u>2</u>	<u>Aspectos Generales</u>	6
2.1	<u>TEORÍA EN LA QUE SE BASA LA METODOLOGÍA</u>	6
2.1.1	<u>Evolución de los proyectos de Informática</u>	6
2.1.2	<u>Fundamentos para la adopción de un criterio de Costo/ Eficiencia.</u>	6
2.1.3	<u>Criterios de aprobación o rechazo de proyectos informáticos</u>	8
2.2	<u>ESTRUCTURA DE LA METODOLOGÍA</u>	8
2.3	<u>CICLO DE PROYECTOS INFORMÁTICOS</u>	9
2.4	<u>LOS PROYECTOS EN EL SNI.</u>	10
2.5	<u>TIPOLOGÍA DE PROYECTOS</u>	13
2.6	<u>ALCANCE DE LA METODOLOGÍA EN CADA CASO</u>	15
2.7	<u>HORIZONTE DE EVALUACIÓN</u>	16
<u>3</u>	<u>Preparación de Proyectos</u>	17
3.1	<u>RESUMEN EJECUTIVO</u>	18
3.2	<u>PLAN O POLÍTICA INFORMÁTICA DE LA INSTITUCIÓN</u>	18
3.3	<u>IDENTIFICACIÓN Y DEFINICIÓN DEL PROBLEMA</u>	19
3.4	<u>DIAGNÓSTICO DE LA SITUACIÓN ACTUAL (SIN PROYECTO)</u>	19
3.4.1	<u>Descripción de la Organización y/o entorno Afectado por el Proyecto</u>	19
3.4.2	<u>Descripción de la Unidad o Departamento</u>	20
3.4.3	<u>Presentación de la solución informática actual</u>	20
3.4.4	<u>Descripción de los procesos</u>	20
3.4.5	<u>Diagrama de Flujo de Datos (DFD) presentando la situación actual</u>	20
3.5	<u>DESCRIPCIÓN GENERAL DE REQUERIMIENTOS</u>	21
3.6	<u>PROGRAMACIÓN DE ACTIVIDADES PARA LA ETAPA DE DISEÑO</u>	21
3.7	<u>REQUERIMIENTOS DE PERSONAL PARA POSTULAR A LA ETAPA DE DISEÑO.</u>	21
3.8	<u>ESTIMACIÓN DE BENEFICIOS</u>	22
3.9	<u>ESTIMACIÓN DE COSTOS DE INVERSIÓN, OPERACIÓN Y MANTENCIÓN PARA LA ETAPA DE EJECUCIÓN</u>	22
3.10	<u>CRONOGRAMA Y CARTA GANTT</u>	22
3.11	<u>TÉRMINOS DE REFERENCIA PARA CONTRATAR ETAPA DE DISEÑO</u>	23
3.12	<u>OPTIMIZACIÓN DE LA SITUACIÓN ACTUAL</u>	24
3.12.1	<u>Medidas administrativas o de rediseño organizacional</u>	24
3.12.2	<u>Inversión marginal a la solución existente</u>	24
3.13	<u>ANÁLISIS DE REQUERIMIENTOS (DISEÑO LÓGICO)</u>	25
3.13.1	<u>Diagrama de flujo de datos (lógico)</u>	25
3.13.2	<u>Modelo de datos</u>	25
3.13.3	<u>Otra documentación</u>	25
3.14	<u>ALTERNATIVAS DE SOLUCIÓN</u>	26
3.14.1	<u>Restricciones asociadas a cada alternativa</u>	26
3.14.2	<u>Producto o servicio esperado en cada alternativa</u>	26
<u>4</u>	<u>Evaluación Costo - Eficiencia</u>	27
4.1	<u>EVALUACIÓN Y SELECCIÓN DE ALTERNATIVAS</u>	27
4.2	<u>ATRIBUTOS RELEVANTES</u>	28
4.3	<u>TÉCNICA DE EVALUACIÓN DE ALTERNATIVAS</u>	29
4.3.1	<u>Evaluación de los atributos</u>	30

	<u>4.3.2</u>	<u>Detalle de la Inversión y cálculo del indicador costo eficiencia</u>	36
5		<u>Términos de referencia para la etapa de ejecución</u>	38
6		<u>Sugerencias para el proceso de licitación</u>	38
7		<u>Bibliografía</u>	40
8		<u>Anexos</u> ¡Error!Marcador no definido.	
		<u>ANEXO 1: TÉCNICA PARA PRIORIZACIÓN Y ASIGNACIÓN DE PONDERADORES</u>	41
		<u>ANEXO 2: ELEMENTOS A CONSIDERAR EN LA EVALUACIÓN TÉCNICA DE PROYECTOS INFORMÁTICOS</u>	44
		<u>ANEXO 3: BENEFICIOS Y COSTOS</u>	48
		<u>ANEXO 4: MODELAMIENTO DE DATOS</u>	52
		<u>ANEXO 5: DIAGRAMA DE FLUJO DE DATOS (DFD)</u>	71
		<u>ANEXO 6 CALIDAD FUNCIONAL</u>	76
		<u>ANEXO 7: ETAPAS DE UN PROYECTO DE DESARROLLO EN INFORMÁTICA</u>	77
		<u>ANEXO 8: EJEMPLOS DE MEDIDAS DE EFECTIVIDAD</u>	85
		<u>ANEXO 9 ESFUERZO REQUERIDO PARA CADA FASE</u>	86
		<u>ANEXO 10 DEFINICIONES LEGALES</u>	88
		<u>ANEXO 11 GLOSARIO</u>	90

1 Introducción

El presente documento tiene como propósito entregar una guía metodológica para la preparación y evaluación de proyectos de informática y su presentación al Sistema Nacional de Inversiones.

Como herramienta metodológica, guiará a las instituciones que desarrollan proyectos de informática en los siguientes aspectos:

- Presentación de un proyecto de informática
- Presentación de alternativas de solución (cuando exista más de una)
- Evaluación de la o las alternativas de solución mediante el criterio costo - eficiencia

El resultado de la aplicación de esta metodología será un documento estándar, el que se presentará a Mideplan, con el propósito de que esta institución revise la factibilidad técnica y económica de desarrollar dicho proyecto.

La metodología se orienta a mejorar la presentación de proyectos, en respuesta a la problemática que se ha detectado en este tipo de inversiones.

A continuación se presentan algunos problemas que se han presentado en las áreas estratégica, táctica y operacional, en relación a los proyectos de informática¹.

A nivel estratégico: se aprecia falta de definiciones estratégicas con respecto a la información dentro de la organización. Uno de los activos más importantes de la organización actual es la información y la tecnología que la soporta, por lo tanto no hay que perder de vista que la tecnología implementada debe apoyar la definición estratégica de la organización con respecto a la información. La tecnología ayuda a administrar la información y ésta debe existir en función de generar conocimiento, entendido como la capacidad de realizar tareas o actividades en forma efectiva.

A nivel táctico: en el área informática hay un descuido del tema calidad, principalmente porque al momento de licitar la principal variable de elección es el precio.

Adicionalmente, es destacable que el cliente chileno es poco exigente con el proveedor. Es así que se ha descuidado, en los desarrollos de software, velar por la reusabilidad del código. La ventaja de que el código sea reusable, es que permite fácilmente agregar o modificar funciones del software, pero esto exige que el proveedor se preocupe de programar de tal manera que esta condición se cumpla. Esto requiere más recursos por parte del proveedor.

En este caso habría que invertir en un mayor grado, pero el costo de mantención disminuye.

¹ Estas deficiencias se vieron confirmadas con el problema informático del año 2000.

Además, cabe destacar que usualmente no se hacen los estudios que se debieran, para determinar la capacidad del hardware a adquirir, o se define a priori la solución informática a implementar sin haber hecho un levantamiento de requerimientos adecuado.

También es usual que no hayan respaldos de la información que es estratégica para la organización.

Otro problema muy importante es que se le da más importancia a la tecnología que a la gestión del proyecto.

A nivel operacional: el principal problema es la inexistencia de documentación de los sistemas. Esta situación es comparable a construir un edificio sin haber hecho previamente los planos y, como ya se mencionó, esto influye en no tener independencia del proveedor del sistema.

Otra falencia a nivel operacional es la inexistencia de programas fuentes, es decir, inexistencia del código humanamente entendible. En este caso sólo existe el programa en idioma máquina, lo cual se traduce en un conjunto de ceros y unos.

2 Aspectos Generales

2.1 Teoría en la que se basa la metodología

2.1.1 Evolución de los proyectos de Informática

Los proyectos de informática han ido evolucionando junto con las organizaciones y a medida que se han ido produciendo cambios tecnológicos. Las organizaciones han evolucionado desde estructuras mecanicistas a flexibles para poder hacer frente a un medio ambiente externo muy cambiante y orientado al cliente. La informática ha ido cambiando tanto tecnológicamente, como también para apoyar la transformación, ya mencionada, en las organizaciones. Es así que la informática ha evolucionado desde sistemas fuertemente centralizados basados en mainframe, cambiando posteriormente a sistemas interactivos (terminales de usuarios), luego vino la computación personal que hacía hincapié en las redes de PC's (surgen los sistemas gerenciales, estratégicos). Finalmente se ha llegado al punto actual con Internet, aplicaciones multimedia, video-conferencia, realidad virtual, etc.

Con respecto a los criterios de inversión, éstos han ido cambiando paralelamente a la evolución ya mencionada de la siguiente forma:

- Primeramente, sólo tenía importancia la evaluación costo – beneficio.
- Luego se evaluaba además si se facilitaba la obtención de los objetivos de la organización y si es que las tecnologías de información (TI) mejoraban la calidad de las inversiones.
- Posteriormente cobró importancia cómo las TI podían mejorar las tomas decisiones y aumentar la participación de mercado.
- Finalmente, ahora se da mayor importancia a cómo las TI pueden aumentar la capacidad de la información(Datawarehouse, Internet, etc.) e innovación.

Se aprecia que la gestión de la organización es muy relevante para considerar los sistemas necesarios a implementar, y se aprecia cómo el énfasis en el criterio de inversión ha ido variando.

2.1.2 Fundamentos para la adopción de un criterio de Costo/ Eficiencia.

Como se señaló en el punto anterior, el criterio de inversión a lo largo de los años ha ido evolucionando haciendo hincapié en distintos aspectos. En la década de los 90, se consideraba como criterio de inversión relevante la posibilidad de aumento de capacidad de la información. Este aumento se entiende orientado a la generación de conocimiento y a una mayor satisfacción del cliente. Es decir, la idea es conocer la tecnología y explotarla de tal manera que permita ofrecer mejores servicios con la información disponible.

La anterior metodología para proyectos de informática evaluaba según el criterio de costo – beneficio. Sin embargo, existen beneficios que son muy difíciles de cuantificar, medir y valorar. Junto con ello, se presentan beneficios intangibles tales como mejoras en la calidad de la información, efecto modernizador, redes sociales que se pueden establecer por Internet, aprendizaje debido al contacto con la tecnología, etc. Además, como ya se expuso,

se espera que conociendo las TI se hagan innovaciones haciendo uso de ella, lo que agrega mayor dificultad, ya que los beneficios se obtendrían después de tener contacto con las TI.

Además, el análisis se centraba sólo en los aspectos tecnológicos, lo que producía que se perdiera de vista la administración de la **información**, con la consiguiente pérdida de conciencia de la administración del riesgo de la misma. Es decir, no se apreciaba una definición de cuál información era la relevante para la organización y qué medidas se tomaban para asegurarla o respaldarla. Hay que considerar que esto es muy importante, porque se ha constatado que empresas que han perdido sus bases de datos han quebrado como promedio en cuatro meses. Las instituciones públicas no quiebran, pero puede afectarse seriamente su funcionamiento.

Considerando que la evaluación encuentra su sentido en ser un apoyo veraz para la toma de decisiones, esta metodología propone el criterio de costo-eficiencia sin perjuicio de que en algunos casos en que se puedan medir y valorar los beneficios, se aplicará costo-beneficio. En particular este esfuerzo es más justificable, en aquellos casos en que hay cambios de procesos y ahorros de costos de transacción para usuarios. Ejemplos: Teletrámites, infocentros.

Además, en el caso de outsourcing, es necesario hacer una evaluación costo-beneficio para determinar la mejor alternativa entre realizar outsourcing o realizar una inversión tradicional.

La idea es conceptualizar factores estratégicos, que tengan que ver con la decisión de cuál información almacenar, la determinación de la información crítica para la organización o cómo ésta se puede aprovechar mejor, lo que se expresa normalmente en temas cualitativos.

El enfoque costo - eficiencia plantea que la conveniencia de la ejecución de un proyecto se determina por la observación conjunta de dos factores:

- El costo: involucra la implementación de la solución informática, adquisición y puesta en marcha del sistema hardware / software y los costos de operación asociados.
- La eficiencia: se entiende como la relación entre bienes y servicios finales (resultados) y los insumos requeridos para ello (esfuerzo). Así se trata de medir en qué grado el gasto de recursos se justifica por los resultados, minimizando costos u optimizando insumos.

Estos dos elementos evaluados en forma conjunta configuran el análisis costo - eficiencia; éste, en definitiva, establece el costo por unidad de cumplimiento del objetivo.

Ahora, al considerar los criterios a utilizar en el método costo-eficiencia es necesario que éstos sean un reflejo de la estrategia que está tomando la organización con respecto a la información.

2.1.3 Criterios de aprobación o rechazo de proyectos informáticos

Como ya se expuso, un importante elemento para aprobar o rechazar un proyecto, es la evaluación costo-eficiencia. Sin embargo, esto no basta, para que un proyecto sea aprobado, tiene que estar bien justificado, ya sea con beneficios cualitativos o cuantitativos. En este sentido, será muy importante la coherencia del proyecto con un plan informático o con líneas estratégicas que se haya planteado la institución, es decir se revisará que los sistemas a implementar cumplan con los objetivos que se ha planteado la institución. También se verificará que el cálculo de ponderadores responda a la estrategia, es así que si un departamento de la organización ha planteado que privilegiará la seguridad, no podría presentar el ponderador correspondiente un porcentaje bajo.

En general, se apreciará la coherencia que presenta el proyecto en sí.

2.2 Estructura de la Metodología

La metodología consta de dos partes: la preparación y la evaluación del proyecto. En la preparación se define la información requerida.

La evaluación propiamente tal consiste en :

- Definición de atributos
- Calificación de los atributos de cada alternativa
- Asignación de un puntaje a cada alternativa, en base a sus atributos
- Cálculo del índice de eficiencia

Además consta de varios anexos, para ayudar a una mejor evaluación del proyecto, como también un mejor control del mismo cuando se este ejecutando.

Los anexos son:

- Técnica para priorización y asignación de ponderadores
- Elementos a considerar en la evaluación técnica de proyectos
- Beneficios y costos
- Modelamiento de Datos
- Diagrama de Flujo de Datos

- Calidad funcional
- Etapas de un proyecto de Desarrollo en informática
- Ejemplos de Medidas de Efectividad
- Esfuerzo requerido para cada fase
- Definciones legales
- Glosario

2.3 Ciclo de proyectos informáticos

En un proyecto informático se pueden distinguir las siguientes etapas:

- **Diseño Lógico.** Los resultados típicos esperados son las respuestas a las preguntas: qué sistemas administrativos se van a apoyar, qué sistemas computacionales se desarrollarán, qué flujos de información son relevantes, qué procesamiento se requiere, qué tipo de datos se manejarán.
- **Diseño Físico.** Se definen los aspectos computacionales del sistema: qué tipo de archivos se necesitan, qué tipo de acceso a archivos, qué programas, qué lenguaje o aplicaciones, qué configuración de hardware/software.
- **Construcción.** Es la elaboración de los programas computacionales anteriormente diseñados
- **Implementación.** Se realizan pruebas, poblamiento de datos, marcha blanca y puesta en marcha definitiva
- **Operación y mantención.** En esta etapa, se deben considerar los costos de operación y mantención. Los costos de operación se refieren a aquellos que permiten el funcionamiento diario del sistema y los de mantención los que permiten la actualización, así como la reparación del mismo.

Es importante destacar, que el término mantención en los proyectos informáticos, se refiere a adecuaciones que requieran los sistemas de propiedad institucional para mantener su vigencia y utilidad. Esta diferencia se debe a que el software tiene características distintas como producto de ingeniería, ya que el software está sujeto a un mayor cambio en los requerimientos, así como en el ambiente con el cual interactúa el sistema.

Existen distintas alternativas de desarrollo de estas etapas: secuencial en cascada, desarrollo incremental, desarrollo en espiral y otras. Estas alternativas se presentan en el Anexo 7 “Etapas de un Proyecto de Desarrollo en Informática”.

2.4 Los proyectos en el SNI

Ciclo de vida de los proyectos

Entenderemos como proyecto al diseño y ejecución de cambios en la asignación actual de recursos que sigue un objetivo y que genera beneficios y costos, cualitativos y cuantitativos, tanto al realizador del proyecto como a terceros. El proceso de transformación de las ideas de inversión, pasando por el diseño y llegando hasta su puesta en marcha, se puede dividir en los siguientes estados:

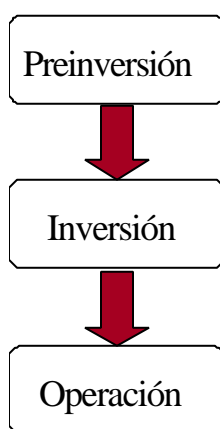


Grafico N° 1 Ciclo de vida de un proyecto

En el estado de preinversión, se estima la factibilidad técnica y económica,. En el estado de Inversión, se diseña y se materializa físicamente la inversión requerida por el proyecto de acuerdo a lo especificado en la etapa anterior. En el estado de operación, se pone en marcha el proyecto y se concretan los beneficios netos que fueron estimados previamente.

El estado de preinversión se detalla a continuación. En este estado se estudia la factibilidad técnico-económica mediante aproximaciones sucesivas por etapas, siendo estas las de idea, perfil, prefactibilidad y factibilidad. La evaluación ex ante de un proyecto, en cualquiera de sus distintas etapas (idea, perfil, prefactibilidad y factibilidad) tiene como objetivos:

- Abordar en forma explícita el problema de la asignación de recursos escasos en forma óptima
- Recomendar al tomador de decisiones, a través de distintas metodologías, sobre la conveniencia relativa de que una acción o un proyecto determinado se realice por sobre otras iniciativas. (Estado de Preinversión)
- Identificar, medir y valorizar, cuantitativa y cualitativamente, los beneficios y costos para la persona(s) o instituciones relevantes.

Los resultados esperados de cada etapa de preinversión, pasando desde idea a factibilidad, se especifican a continuación:

Etapas de la Evaluación Ex–Ante de Proyectos (etapas del estado de preinversión)

Como se ha dicho, la selección de los mejores proyectos de inversión, es decir, los de mayor conveniencia relativa (evaluación) y hacia los cuales deben destinarse preferentemente los recursos disponibles, constituye un proceso que sigue las siguientes etapas secuenciales:

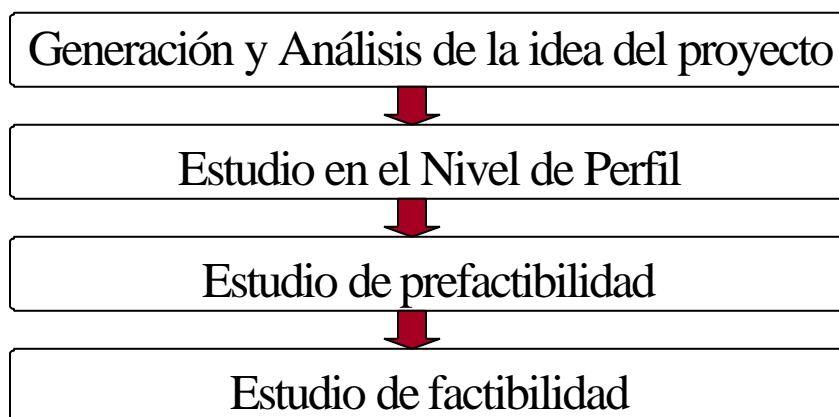


Gráfico N° 2: Etapas de Preinversión

Cada una de ellas busca reproducir el ciclo de vida del proyecto, de manera que a medida que se avanza en las etapas, los estudios van tomando mayor profundidad y se va reduciendo la incertidumbre, respecto a los costos y beneficios netos esperados del mismo.

La secuencia por etapas tiene por justificación evitar los elevados costos de los estudios y poder desechar en las primeras etapas los proyectos que no son adecuados.

Cada etapa se presenta en la forma de un informe, cuyo objetivo fundamental es presentar los elementos que intervienen orientados claramente a la toma de decisiones de abandonar o proseguir la idea. En mayor detalle:

Etapas de Generación y análisis de la idea de proyecto

Es crucial contar con un buen diagnóstico, de modo que la generación de una idea de proyecto de inversión surja como consecuencia clara de necesidades insatisfechas, de objetivos y/o políticas generales de la organización, de un plan de desarrollo, etc.

Se debe establecer su magnitud, a quienes afecta y la confiabilidad de la información utilizada. Así como también las alternativas disponibles.

Del análisis surgirá la especificación precisa del bien que se desea construir o el servicio que se pretende dar. Y servirá para adoptar la decisión de abandonar, postergar o profundizar la idea de proyecto.

Etapa de Estudio en el Nivel de Perfil

Se estudian los antecedentes que permitan formar un juicio respecto de la conveniencia y factibilidad técnico-económica de llevar a cabo la idea de proyecto.

El énfasis está en identificar los beneficios y costos pertinentes respecto de la situación base (situación actual optimizada), sin incurrir en mayores costos en recursos financieros y humanos para medirlos y valorarlos, debe incluir un análisis preliminar de los aspectos técnicos, estudios de mercado y los de evaluación. Se utilizan estimaciones gruesas de los beneficios y costos. Generalmente basadas en información existente, es decir, sin incurrir en costos significativos por concepto de levantamiento de información.

Como conclusión de esta etapa, están las decisiones alternativas de abandonar, postergar o profundizar el proyecto pasando a la etapa de prefactibilidad.

Etapa de Estudio de Prefactibilidad

Se examinan con mayor detalle las alternativas viables desde el punto de vista técnico y económico que fueron determinadas en la etapa anterior, y se descartan las menos atractivas.

El énfasis de esta etapa es medir los beneficios y costos identificados en la etapa de perfil. Existe un esfuerzo de inversión en información para disminuir la incertidumbre.

Es necesario estudiar con especial atención los aspectos de mercado, la tecnología, el tamaño y la localización del proyecto, las condiciones institucionales y legales relevantes para el proyecto.

El estudio de mercado es la base para estimar los ingresos, incluirá un estudio de la oferta y demanda, así como de los precios de comercialización. El análisis tecnológico incluye equipos, materias primas y procesos, que permiten determinar los costos del proyecto. Sobre el tamaño y localización del proyecto se debe considerar la identificación y localización de los centros de consumo, de abastecimiento de insumos, canales de distribución, competencia, proyecciones de crecimiento, así como el impacto en el medio ambiente.

El análisis de los aspectos administrativos permite determinar algunas componentes de costo fijo y la organización de los recursos humanos, físicos y financieros. El análisis de los aspectos legales permite conocer las restricciones de ese tipo que limitan al proyecto. Ejemplo: tributación (pago de impuestos), permisos requeridos, contaminación ambiental, eliminación de desechos.

Todo lo anterior permite tener una estimación de los montos de inversión, costos de operación y de los ingresos que generaría el proyecto durante su vida útil. Lo que se utiliza para la evaluación económica y para determinar las alternativas más rentables. Conviene sensibilizar los resultados de la evaluación a cambios en las variables más importantes.

Como resultado de la etapa se debe decidir realizar el proyecto o postergar, abandonar o profundizar pasando a la etapa de factibilidad.

Etapa de Estudio de Factibilidad

La factibilidad se enfoca a un análisis detallado y preciso de la alternativa que se ha considerado más viable en la etapa anterior. El énfasis está en medir y valorar en la forma más precisa posible sus beneficios y costos.

Dada la cantidad de recursos destinados a esta etapa, sólo llegarán a ella los proyectos para los que no hay duda de su rentabilidad positiva, es decir, que se van a llevar a cabo. Por ello, toma más importancia los flujos financieros y la programación de obras.

Una vez definido y caracterizado el proyecto, debe ser optimizado en tamaño, localización, momento óptimo de la inversión, etc.

Se debe coordinar la organización, puesta en marcha y operación del proyecto. Determinar el calendario de desembolsos para la inversión, disponibilidad de equipos y sus plazos, anteproyecto de ingeniería, selección y entrenamiento del personal de administración, operación y mantenimiento. También las fuentes, condiciones y plazos de financiamiento.

Esta etapa es la conclusión del proceso de aproximaciones sucesivas en la formulación y preparación de un proyecto y constituye la base de la decisión respecto a su ejecución.

2.5 Tipología de proyectos

Las tipologías existentes en el SNI responden a la necesidad de clasificación de los distintos proyectos que se presentan al Sistema. Dada la gran variedad de proyectos presentados, estas tipologías son generales.

Las tipologías relevantes para proyectos de informática son:

- Proyectos de Desarrollo de aplicaciones: elaboración y puesta en marcha de programas o sistemas computacionales.
- Proyectos de Equipamiento: adquisición por primera vez de equipos, incluyendo tanto hardware como software básico utilitario².

² Sistemas operativos, procesadores de texto, planillas de cálculo, navegadores y antivirus.

- Proyectos de Mejoramiento, ampliación o reposición: aumento de capacidad y calidad de servicios de hardware y/o mejoramiento de software.

El método de evaluación de cada tipología, variará según el monto involucrado. Es así que para montos inferiores a US\$ 50.000³ el análisis será cualitativo, para el caso de montos mayores se aplicará costo-eficiencia o si se pueden medir y valorar los beneficios costo-beneficio. En síntesis:

Costo del Proyecto	Criterio de evaluación
Menos de U\$ 50.000	Costo eficiencia a nivel cualitativo
Más de U\$ 50.000	Costo eficiencia a nivel cuantitativo (según metodología)
Máde U\$ 50.000 y hay cambios de procesos que involucran ahorro de costos para usuarios.	Costo Beneficio

Tabla N°1: Criterios de Evaluación según tipo de proyecto

En los criterios de evaluación, "costo eficiencia a nivel cualitativo" corresponde al nivel de evaluación que se establece en el Instructivo SEBI para los proyectos de Informática, mientras que "costo eficiencia a nivel cuantitativo" corresponde a la presente metodología⁴.

³ En lo sucesivo, todas las cifras en dólares son al valor del dólar observado a la fecha de la evaluación del proyecto.

⁴ En particular el capítulo 4.

2.6 Alcance de la Metodología en cada caso

En esta metodología se usa una técnica de evaluación de alternativas aplicable a cada una de las tipologías. La técnica de evaluación está basada en tablas con ponderadores. Cada tipología tiene sus tablas pertinentes.

Además, según cada tipología y la etapa en curso se determinan distintos requisitos de información que se debe presentar, esto queda reflejado en el Gráfico N° 3:

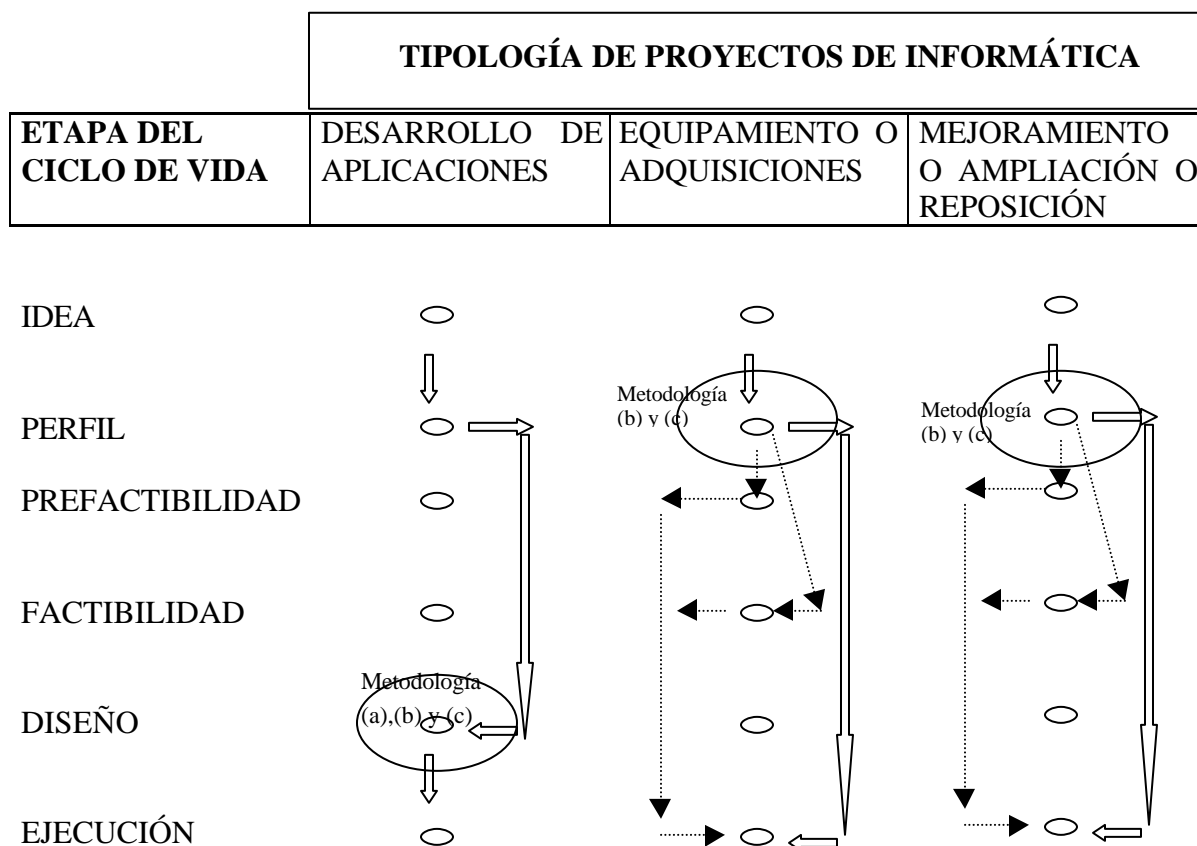


Gráfico N° 3: Ciclo de vida y Tipología de Proyectos

La metodología se aplica a distintas etapas del ciclo de vida del proyecto dependiendo de la tipología. En efecto, para proyectos de desarrollo se requiere tener el diseño lógico, luego la metodología se aplica para pasar de la etapa de diseño a la de ejecución (para pasar de perfil a ejecución se requiere la documentación obtenida en el diseño lógico, así como la evaluación de las distintas alternativas de solución).

Para proyectos de equipamiento, adquisiciones, mejoramiento, ampliación o reposición, la metodología se aplica para pasar de la etapa de perfil a ejecución, sin perjuicio de que en algunos casos se puede pasar de perfil a prefactibilidad o factibilidad (y posteriormente a ejecución) dependiendo de la posibilidad de cuantificar los beneficios del proyecto, esta posibilidad se indica con las flechas de líneas punteadas en el gráfico anterior.

Adicionalmente, se debe considerar que la metodología considera tres resultados esperados de distinta naturaleza y que no aplican a todas las tipologías:

- a) Diseño Lógico.
- b) Evaluación de alternativas de solución.
- c) Elaboración de Términos de Referencia para la licitación de la etapa siguiente.

El diseño lógico (a) , se requiere como resultado sólo para proyectos de desarrollo de aplicaciones, por lo tanto para las otras tipologías sólo se requiere (b) y (c). Estas diferencias están representadas en el Gráfico N° 3, en el cual para cada tipología se indica con un óvalo la etapa en la cual se aplica la metodología y en cada caso se señala el alcance de la aplicación de la metodología, es decir, si se aplica (a), (b) y (c) o sólo (b) y (c).

Es importante notar que para el caso de informática, el significado de la etapa de Diseño varía, con respecto a un proyecto tradicional, ya que como producto de esta etapa se obtendrá el Diseño Lógico, así como la evaluación de las distintas alternativas de solución, evaluación que normalmente se presenta como resultado final de la etapa de factibilidad. Cabe mencionar que la tipología para el caso de compra de paquetes de software⁵, corresponde a la tipología de Desarrollo para la etapa de Ejecución. Sin embargo en este caso sólo se pedirá el diagrama de procesos o el diagrama de flujo de datos (DFD), porque es importante considerar, si es que la organización es capaz de adaptarse al software comprado, situación que se puede apreciar de mejor manera, si es que existe un levantamiento de requerimientos.

Cada uno de estos requisitos quedan explicados en forma más amplia en el resto de la metodología.

Además en el Anexo 9 se describe el esfuerzo necesario para cada fase de un proyecto informático.

2.7 Horizonte de Evaluación

El horizonte de evaluación, debe considerar como máximo cuatro años, debido a los cambios tecnológicos que en el área informática ocurren con gran velocidad.

⁵ No utilitarios, es decir, distintos a los mencionados en el pié de página anterior.

3 Preparación de Proyectos

La información necesaria para postular a cada etapa de acuerdo a cada tipología de proyectos se enumera a continuación en la Tabla N° 2. La marca en cada celda indica los requisitos mínimos necesarios para esa tipología, la celda en blanco indica que ese requisito no es imprescindible para esa tipología, no obstante, en proyectos específicos se pueden exigir más requisitos que los que aquí se señalan. Los números entre paréntesis en la primera columna de requisitos de información, hacen referencia al punto dentro de este capítulo en que se describe dicho requisito

Etapa desde la que postula	Proyectos de desarrollo		Equipamiento, adquisición, mejoramiento, ampliación o reposición
	Perfil	Diseño	Perfil
Etapa a la que postula	Diseño	Ejecución	Ejecución
REQUISITOS DE INFORMACION			
Resumen Ejecutivo (3.1)		✓	✓
Plan, o Política Informática de la Institución (3.2)	✓	✓	✓
Identificación y Definición del Problema (3.3)	✓	✓	✓
Diagnóstico de la situación actual (3.4)	✓	✓	✓
Optimización de la situación actual (3.12)	✓	✓	✓
Descripción general de requerimientos (3.5)	✓		✓
Análisis de requerimientos (diseño lógico) (3.13): <ul style="list-style-type: none"> • Descripción de la solución. • Diagrama de flujo de datos. • Modelo conceptual de datos 		✓	Ver Nota al pie ⁶
Programación de actividades para etapa de diseño (3.6)	✓		
Requerimiento de personal estimadas del Proyecto en su etapa de Diseño (3.7)	✓		
Análisis de alternativas de solución (3.15)		✓	✓
Evaluación de costo – eficiencia (4)		✓	✓
Estimación de beneficios esperados (3.8)	✓	✓	✓
Estimación de costos de inversión, operación y mantención para la etapa de ejecución (3.9)	✓	✓	✓
Cronograma o carta gantt (3.10)	✓	✓	✓
TDR para la etapa de Diseño (3.11)	✓		
TDR para la etapa de Ejecución (5)		✓	✓

Tabla N° 2: Requisitos de información según tipologías y etapa del ciclo de vida

⁶ En el caso de proyectos de mejoramiento, ampliación o reposición, si involucran aumento de capacidad o calidad de software (no básico ni utilitario), se sugiere exigir el diagrama de flujo de datos

En el caso de proyectos de desarrollo, con esa información a nivel de perfil, se podrá analizar la idea y trabajar en la propuesta de diseño, obteniéndose la información relevante para la próxima etapa de inversión

La información entregada para llevar a cabo la etapa de diseño en el caso de proyectos de desarrollo es un elemento preliminar de desarrollo. Ésta sirve de base para el análisis que se efectuará en el levantamiento, por ello algunos requerimientos de información para pasar de perfil a diseño se repiten cuando se pasa de diseño a ejecución en este caso cada requisito debe ser tratado en mayor profundidad.

Para mayores detalles sobre el Diagrama de Flujo de datos consultar el Anexo 5, para el modelo conceptual de datos ver el Anexo 4.

A continuación se detallan cada uno de los requisitos de información mencionados en la Tabla N° 2.

3.1 Resumen ejecutivo

El resumen ejecutivo debe contener:

- Identificación del problema a resolver
- Objetivo del proyecto
- Requerimientos
- Breve justificación de la solución escogida
- Costos de inversión y operación de la solución

3.2 Plan o Política Informática de la Institución

La política informática debe contener estrategias encaminadas a una buena gestión, tanto de la información como de la tecnología que la soporta.

En particular, se debe definir, en los casos que corresponda:

- El papel de la información dentro de las distintas áreas de la institución. Dependiendo del área, la información debiera cumplir con los siguientes atributos en distintos grados de importancia:
 - Confidencialidad: nivel de protección de la información que se necesita.
 - Integridad: precisión y suficiencia de la información
 - Disponibilidad: para cuáles usuarios estará la información disponible
 - Confiabilidad: la información obtenida debe ser apropiada para la gestión y operación de la institución.
 - Información Externa: se debe poder acceder a requerimientos de información formulados por otras instituciones.

- Clasificación de la información contenida en las bases de datos que posee la institución en cuanto a su relevancia. La relevancia se define en función de lo que significa la pérdida de información, para la misión de la institución, de manera que se debe considerar estratégica aquella información cuya pérdida afecta a la misión de la institución y no estratégica a aquella cuya pérdida no afecta a la misión.
- Procesos claves dentro de la institución, ya que las mejoras a estos procesos es lo que se debe desprender del plan informático
- Estrategia de capacitación
- Estrategia de software básico: sistemas operativos, sistemas de bases de datos, para servidores y estaciones de trabajo.
- Estrategia de hardware: procesadores, capacidades de almacenamiento, capacidades de crecimientos, estándares, etc.
- Estrategia de arquitectura: centralizada, distribuida o mixta. Sistemas propietarios o sistemas abiertos.
- Estrategia de comunicaciones y redes: protocolos, arquitectura de redes, enlaces de comunicaciones, etc.
- Estrategia de desarrollo de software de aplicaciones: desarrollo local, desarrollo externo, desarrollo mixto.
- Estrategia de recursos humanos: Unidad de Informática y sus alcances.

3.3 Identificación y Definición del Problema

Se debe determinar qué problema se intenta solucionar o qué objetivo se pretende alcanzar mediante el proyecto (en términos generales, ya que el análisis en detalle se abordará en la etapa de diagnóstico). Es importante aclarar este punto, por cuanto constituirá el motivo por el que se origina el proyecto. Se sugiere utilizar la metodología del árbol causa – efecto⁷.

3.4 Diagnóstico de la Situación Actual (Sin Proyecto)

3.4.1 Descripción de la Organización y/o entorno Afectado por el Proyecto

Esta etapa tiene como objetivo dar una descripción completa del área o departamento involucrado en el estudio sirviendo como base para el análisis.

⁷ Si se desea emplear esta metodología, se pueden consultar mayores detalle en la “Guía metodológica general para la preparación y evaluación de proyectos de inversión social”, Sanín, Héctor, ILPES, 1995

3.4.2 Descripción de la Unidad o Departamento

Este hito consta en la generación de un documento en el cual se describe la situación actual de la unidad o departamento en estudio. El documento debe contener los siguientes puntos insertos:

- Organigrama de la unidad o departamento: un organigrama simple de cómo está hoy la unidad que se debe estudiar
- Funciones y responsabilidades de la unidad o departamento: descripción de funcionalidades permiten acotar el proyecto a realizar.
- Objetivos actuales: se debe exponer cuáles son los objetivos tanto de corto como de largo plazo que se ha planteado la unidad o departamento. Para esto, se debe realizar una enumeración y una breve descripción.
- Interacción con su entorno: definir con un esquema simple la interrelación del Departamento con su entorno. Mencionar las relaciones que guardan nexos con el tema que se desea estudiar y que puede ser importante acotarlas.

3.4.3 Presentación de la solución informática actual

La idea es describir los sistemas, software y hardware del área problema. Para lo anterior hay que basarse en el anexo 2 “Elementos a considerar en la Evaluación Técnica de Proyectos Informáticos”. Además, es importante presentar en este caso un diagrama de la arquitectura de la solución actual.

Para las tipologías de desarrollo o mejoramiento (las que incluyen desarrollo de software), se requiere:

3.4.4 Descripción de los procesos

Se deberá definir cuáles son los procesos que tienen relación con el tema en estudio, dando un nombre simple al proceso y una breve descripción de cómo opera.

3.4.5 Diagrama de Flujo de Datos (DFD) presentando la situación actual

Los DFD que se deberán describir, son los indicados en el punto anterior. El objetivo es visualizar en un esquema simple cómo fluye la información. Como ejemplo de Diagrama de Flujo de Datos ver el Anexo Diagrama de Flujo de Datos.

Estos diagramas pueden situarse en varios niveles. Para esta etapa, sólo se necesita una presentación a nivel intermedio (siguiente al nivel de contexto, con los hijos más relevantes).

3.5 Descripción general de requerimientos

La idea es describir los requerimientos principales a los cuales debe responder la solución. Estos requerimientos deben ligar el rendimiento de la solución a implementar con procesos estratégicos de la solución. Por ejemplo, para un servicio determinado para el cual es muy importante el número de reportes para beneficiarios y se ha determinado como decisión estratégica disminuir las colas, el requerimiento debiera fijarse en el número de cotizaciones por unidad de tiempo que se necesitan para cumplir ese objetivo.

3.6 Programación de actividades para la etapa de diseño

En este punto es deseable un cronograma o carta gantt mostrando las actividades necesarias para realizar el levantamiento y cuanto tiempo requerirá. A diferencia de la programación de la planificación del desarrollo del software propiamente tal, el tiempo planificado para esta actividad debiera ser bastante exacto.

3.7 Requerimientos de personal para postular a la etapa de diseño.

Se debe presentar un presupuesto detallado por fases o total del estudio. La información pertinente debe desagregarse en, al menos, los ítems que se muestran en el siguiente cuadro, identificando la cantidad y el precio unitario de cada uno. Se deben valorar sólo los ítems que signifiquen desembolso adicional para el servicio; en consecuencia, no debe incluir personal propio. Se entiende por personal propio los funcionarios de la institución que financia o que está postulando el estudio y que se estima se dedicarán en jornada parcial o completa a ser contraparte del mismo o a participar en su ejecución. Por otra parte, el personal externo son las personas que asignará la empresa o institución que desarrolle el estudio (empresa consultora u otra institución). También debe incluirse en esta categoría el personal que se contrate específicamente para la ejecución de éste o para hacer de contraparte, y cuyo contrato finalice junto con su término.

ITEM	UNIDADES*	PRECIO UNITARIO	CANTIDAD	VALOR TOTAL (M\$)
Profesionales ¹ Técnicos Secretarias Viáticos y pasajes Materiales y equipos				
Total				
Gastos Generales				

* La unidad de medida del recurso humano es el número de horas.

1\ - Los profesionales deben ser desagregados por tipo y nivel.

Tabla N° 3: Costos Directos de Personal

Además, se debe informar de los requerimientos totales de personal del estudio de acuerdo al siguiente esquema:

PERSONAL	PROPIO (Número de horas)	EXTERNO (Número de horas)
Profesionales ¹ Técnicos Secretarías, Asistentes y otros Otros		
TOTAL		

1\ - Los profesionales deben ser desagregados por tipo y nivel.

Tabla N° 4: Costos Totales de personal

Este cuadro debe completarse en base a la información de estudios similares ya efectuados (si existen), en base a información de los posibles proveedores (cotizaciones) y en base a las actividades del estudio que se presenta.

3.8 Estimación de beneficios

Se deben describir los beneficios en forma cualitativa. De ser posible identificar, medir y valorar los beneficios, se sugiere considerar los ítemes incluidos en el Anexo3.

3.9 Estimación de costos de inversión, operación y mantención para la etapa de ejecución

La estimación de los costos de inversión, operación y mantención debe estar fundamentada en la experiencia anterior de la institución, si esta existe. La idea es presentar claramente como se obtuvieron los valores correspondientes, explicando las cifras usadas y describiendo el software, hardware y servicios profesionales que se usaría, después de realizar el levantamiento de requerimientos.

La estimación debe afinarse después de realizar el levantamiento de requerimientos, ya que se tendrá más claro que funcionalidades se implementarán y cuales no, lo que permite predecir con mayor información cuánto será el costo de inversión, operación y mantención futuro.

Es importante notar que en informática se entiende por costos de mantención los destinados a las adecuaciones que requieren los sistemas para mantener su vigencia y utilidad.

3.10 Cronograma y Carta Gantt

Un cronograma o carta gantt, establece el orden de las actividades a abordar, detallando cuales tareas pueden ser elaboradas en forma paralela y cuales tareas son necesarias para

realizar otras. Esta descripción se hace simbolizando cada tarea por una barra, cuyo largo dependerá del tiempo que toma realizar cada tarea.

A continuación se muestra un ejemplo, elaborado en Microsoft project 98®⁸:

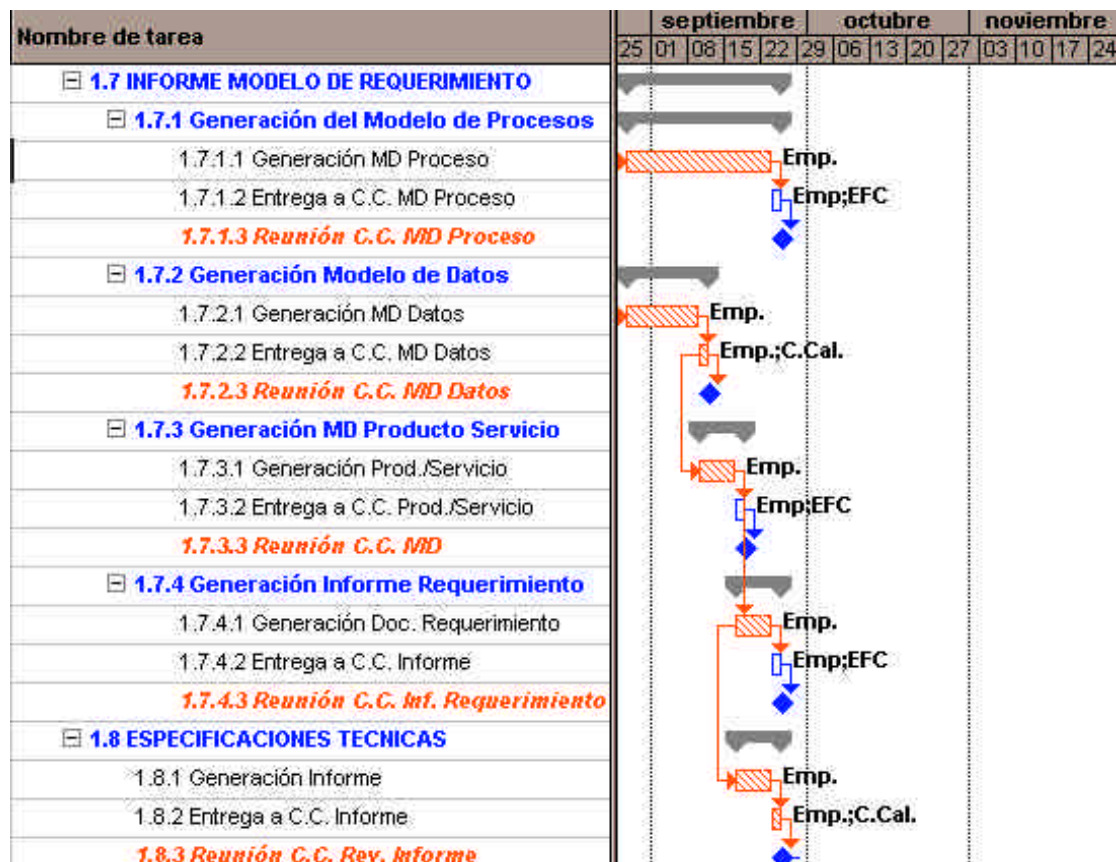


Gráfico N°4: Carta Gantt

Como se observa en el ejemplo, las flechas indican la dependencia entre las distintas tareas. Las indicaciones que aparecen al lado de cada barra (Emp,EFC, etc.) indican quién es el responsable de la tarea. Los diamantes indican hitos de control (reuniones, entrega de resultados)

3.11 Términos de Referencia para contratar etapa de diseño

Los términos de referencia deben incluir toda la información necesaria, para poder licitar el diseño, así como la evaluación de las distintas alternativas de solución (en el caso que el formulador estime que no hay capacidad técnica al interior de la institución para evaluar el proyecto).

⁸ ®: Microsoft Corporation

En este sentido, se debe especificar claramente el producto final, el cual se traduce en:

- Análisis de procesos o flujo de datos, en relación a la problemática detectada, lo cual se puede traducir en una optimización de procesos.
- Documentación asociada al Diseño lógico
- Propuesta y evaluación de las alternativas de solución

Estos puntos quedarán más claros en los restantes puntos de la metodología.

3.12 Optimización de la situación actual

En base a la información recopilada, vale decir, antecedentes preliminares, presentación del problema y el diagnóstico, se determinará si es posible mejorar la situación actual, ya sea con medidas administrativas, de rediseño organizacional, o con inversiones marginales.

La optimización de la situación actual puede convertirse en una importante fuente de ahorro de recursos, por ello se recomienda explorar esta alternativa.

A modo de ejemplo, se puede optimizar la situación actual por medio de alguna de las siguientes medidas:

3.12.1 Medidas administrativas o de rediseño organizacional

- Rediseño de procesos al interior de la institución
- Eliminar trámites innecesarios
- Redistribuir físicamente al personal de manera de optimizar los procesos
- Elaborar manuales de procedimientos administrativos

3.12.2 Inversión marginal a la solución existente

-Rediseñar y/o normalizar las bases de datos, eliminando duplicidades. Aparte de proporcionar una mayor eficiencia, esta medida permitirá una mayor seguridad, menor duplicidad y por lo tanto, una mejor eficiencia de la información mantenida en bases de datos.

- Capacitar, tanto a usuarios, como a especialistas del área informática de la institución. Muchas veces se dispone de las herramientas de hardware, software o comunicaciones, pero no se hace un adecuado uso de ellas.

- Redistribuir de forma más racional los recursos computacionales entre los distintos usuarios. En este caso, se recomienda considerar aspectos tales como: nivel de uso y capacidades de los recursos.

- Segmentación de redes mediante el balance y reasignación de la carga en la red de informática.

- Balances de carga en dispositivos como discos, cpu, memoria.
- Otras inversiones

3.13 Análisis de requerimientos (Diseño lógico)

3.13.1 Diagrama de flujo de datos (lógico)

El objetivo es visualizar en un esquema simple la información requerida y cómo fluye dicha información entre las distintas entidades y procesos. Como ejemplo, ver el **Anexo 5** correspondiente al Diagrama de flujo de datos (lógico). En lo que se refiere al flujo de datos, estos Diagramas pueden ir aumentando en complejidad, en la medida que cada flujo se vaya describiendo en mayor profundidad.

3.13.2 Modelo de datos

Para aquellos proyectos de desarrollo que incluyen la etapa de ejecución (y por ende el diseño lógico) un hito importante del análisis de requerimientos es la formulación del modelo de datos. Para efectos de presentación de antecedentes a MIDEPLAN está descrito cómo hacer un modelamiento en el **Anexo 4** Modelamiento de Datos.

3.13.3 Otra documentación

En el caso de que se desarrollen aplicaciones específicas:

- Orientadas al objeto: Presentar casos de uso, diagramas de clase, diagramas de objetos y si es posible de estado.
- No orientadas al objeto: Presentar lista de funciones, con sus parámetros, entradas y salidas.

En el caso de sistemas de información geográficos, debe pedirse como parte del análisis requerimientos, las capas y cruces mínimos necesarios. En el caso de que el diagnóstico determine la necesidad de contar con capas adicionales, se debe identificar que instituciones (distintas a la que presenta el proyecto), pueden disponer de dichas capas, y se debe considerar la alternativa de adquirirlas o acceder a ellas vía convenio, *versus* la alternativa de desarrollarlas nuevamente para la institución.

Para proyectos de desarrollo de páginas web y otros desarrollos de Internet, Intranet o Extranet, debe solicitarse un mapa de navegación que de cuenta de la información que se requiere en el sitio Web. Además es deseable un análisis de los procesos involucrados mediante DFD's u otra herramienta, que permitan un uso cooperativo real de las herramientas de Internet. Por otra parte, es importante presentar procedimientos administrativos así como adquirir software y hardware en lo que se refiere a seguridad.

Aquellos proyectos, que involucren la compra de software de clase mundial, o paquetes desarrollados, se debe presentar un informe que especifique que requerimientos de la organización son satisfechos por la organización y cuales no, para poder determinar la factibilidad de ser implementada con éxito.

Para proyectos de las tipologías: equipamiento, mejoramiento, ampliación reposición, los requerimientos de información, se especifican en el anexo2 (todos los puntos del anexo, excepto el 1)

3.14 Alternativas de Solución

La presentación de alternativas de solución está relacionada en forma directa con las capacidades técnicas para generar alternativas y el nivel de problemas que se desean solucionar.

Una adecuada presentación de alternativas será el paso inicial en una correcta presentación y preparación del proyecto de informática o alternativa final de solución. Además, será la base para el documento de especificaciones técnicas en el proceso de formalización de compra o licitación.

3.14.1 Restricciones asociadas a cada alternativa

La idea es mencionar las restricciones de precio, mantención, operación y tecnología que presenta cada alternativa.

3.14.2 Producto o servicio esperado en cada alternativa

Debe establecerse si se espera el mismo servicio o producto por cada alternativa de solución y en qué consiste en términos generales. Por ejemplo, se podría mencionar que el producto de la alternativa seleccionada cumplirá con un requerimiento específico y que en cambio no solucionará otro requerimiento menos importante.

4 Evaluación Costo - Eficiencia

Los fundamentos para la adopción de este criterio fueron presentados en la sección 2.12. Antes de iniciar la presentación de la metodología de evaluación y selección de alternativas bajo este criterio, cabe definir la diferencia entre efectividad y eficiencia.

La efectividad se asocia al logro de los objetivos y metas establecidas, la eficiencia es alcanzar la efectividad haciendo un uso racional de los recursos disponibles⁹. De acuerdo a lo anterior, la eficiencia incluye necesariamente la efectividad, y por tanto un criterio de costo – eficiencia es más exigente que uno de costo - efectividad

En la metodología que se presenta en los siguientes capítulos, se incluyen algunos atributos de efectividad pero también algunos de eficiencia, por lo que cabe clasificar a esta metodología como de costo – eficiencia.

4.1 Evaluación y Selección de Alternativas

Es importante tener en cuenta que, en la generalidad, los proyectos de informática se comportan siguiendo la siguiente curva de efectividad v/s costos:

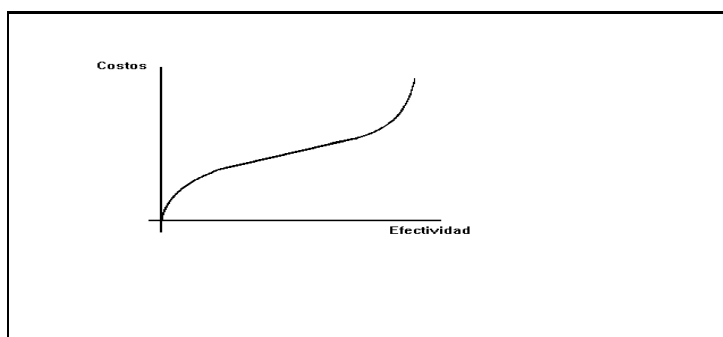


Gráfico N° 5: Relación costo - efectividad

En el gráfico se aprecia que:

- Los primeros índices de efectividad pueden tener un alto costo.
- En una etapa intermedia, los costos y la efectividad se valoran en unidades fijas.
- Finalmente, el logro total o a medida que nos acercamos a la efectividad total, los costos se pueden elevar en forma exponencial.

Si bien, la idea es satisfacer la totalidad de los requerimientos planteados en el diagnóstico, es recomendable ser muy cuidadoso en un dimensionamiento adecuado de las alternativas de solución, evaluando dichas alternativas a la luz de los costos involucrados e intentando lograr la mayor efectividad al menor costo.

⁹ A modo de ejemplo, si el objetivo es acertar en el blanco de una diana, quien logre esa meta después de lanzar 100 dardos, ha sido efectivo, pero sumamente ineficiente, el que ni siquiera logra acertar después de 100 intentos es ineficaz, y el que acierta a la primera ocasión es eficaz y eficiente.

El cálculo de los indicadores (cómo VAN y TIR), puede mostrar una rentabilidad alta (que en general es difícil de medir) y no guarda relación con la calidad de la solución tecnológica seleccionada, cabe centrarse por lo tanto en la optimización del proyecto.

Obviamente, la selección final se hace en el proceso de licitación, sin embargo, el análisis previo de generación y selección de alternativas debería ayudar a una mejor especificación de las bases técnicas para la formalización de la compra o licitación, evitando conflictos por vacíos en las bases y acotando el espacio de alternativas. Esto permitirá un análisis más minucioso de las propuestas.

Para la evaluación de alternativas se medirán ciertos atributos de la solución propuesta y se definirán ponderadores para dichos atributos. Esto se explica en la siguiente sección.

El uso de ponderadores de atributos permitirá evaluar las distintas alternativas planteadas, intentando seleccionar las alternativas que ofrezcan el mejor nivel técnico y que resuelvan de la mejor manera posible el problema planteado.

La dificultad del modelo radica justamente en la definición de atributos y la estimación de los ponderadores. En efecto, el proceso de generación de atributos, asignación de puntajes y ponderadores, presupone claridad respecto de los requerimientos, de los problemas del actual sistema, de los objetivos del nuevo sistema y de las funciones y sistemas administrativos a ser apoyados por la configuración.

Si se dan las condiciones anteriores y se realiza un adecuado análisis, debería esperarse que los atributos y el valor asignado a los ponderadores refleje las reales necesidades de la institución con respecto al sistema computacional.

Al no darse esas condiciones, queda abierta la posibilidad de que el evaluador "maneje" los ponderadores para "seleccionar" alguna alternativa preconcebida, lo que hace que la herramienta resulte inservible para los objetivos de acercarse a la selección de una buena configuración computacional.

4.2 Atributos Relevantes

Se pueden plantear dos tipos de atributos:

Atributos imprescindibles

Los atributos imprescindibles son aquellos que obligatoriamente deben cumplirse en su totalidad, en la alternativa a evaluar. De lo contrario, dicha alternativa no deberá ser considerada.

A lo menos, se deben considerar los siguientes atributos como imprescindibles:

- La alternativa de solución está de acuerdo con la política informática (si es que existe) de la institución.
- La institución dispone de las capacidades técnicas y administrativas para soportar la solución. (por ejemplo para administrar la red)

Atributos evaluables (deseables y muy deseables)

Los atributos evaluables son aquellos medibles y por tanto que permiten una evaluación y discriminación de cada alternativa, lo que es importante pues las alternativas de la solución pueden ser variadas y complejas, para la decisión de implementar una solución.

La clasificación de atributos en muy deseables o deseables debe formularse en base al plan informático de la institución.

Como sugerencia, creemos importante que la evaluación de alternativas considere atributos que hagan hincapié en la información, la cual es soportada por la tecnología. Muchas veces, por ejemplo, se eligen soluciones eficientes, pero en áreas que la institución no necesita tal eficiencia. Por otra parte, alguna institución puede que tenga información disponible, sin embargo, la seguridad de la información es mala. En este caso, aparentemente la solución es eficiente, pero está descuidando un punto importante que es el riesgo de pérdida de la información.

Es así que más adelante, en la exposición de técnicas de evaluación de alternativas, se toman en cuenta atributos de la información, junto con otros de carácter técnico.

En cuanto a los atributos es importante para la evaluación de soluciones, construir tablas en las cuales se diga cuáles fueron los conceptos considerados y los pesos relativos que para ellos se asignaron. La técnica se expone a continuación.

4.3 Técnica de Evaluación de Alternativas

La técnica descrita a continuación busca obtener un puntaje para cada una de las soluciones a evaluar, considerando los criterios señalados anteriormente y los antecedentes recogidos en las etapas anteriores.

Si existiera sólo una alternativa, el puntaje deberá ser calculado de todas maneras para ella, ya que permite apreciar cómo se tomó la decisión de optar por la solución.

Además, se sugiere que las matrices expuestas a continuación sean completadas también en el proceso de licitación para la evaluación de las propuestas en concurso.

4.3.1 Evaluación de los atributos

Los factores de evaluación a considerar son los siguientes:

- A) Efectividad
- B) Plataforma Tecnológica
- C) Calidad Técnica de la Solución
- D) Ahorro de costos operacionales

Además es importante considerar la **Calidad Funcional de la Solución**, pero que no puede ser evaluada con el avance que tiene el proyecto a esta altura (postulando a etapas que nunca van más allá del diseño), porque la información necesaria para evaluar los atributos se obtiene del Diseño Físico de la solución, por este motivo solamente se deja como referencia la información correspondiente a los atributos de Calidad Funcional en el anexo 6, sin exigir su evaluación.

Cada uno de estos factores será calificado con un puntaje de 1 a 100, de acuerdo a los siguientes procedimientos:

A) Efectividad

El objetivo de esta evaluación es calificar el nivel de satisfacción de las necesidades a ser cubiertas por el sistema en cuestión. Para ello, se deberán considerar todas aquellas funciones que debieran satisfacerse, tanto las de carácter operativo como las estratégicas y tácticas. Esta evaluación debe seguir los siguientes pasos:

- i. Definir funciones que debe satisfacer el sistema
- ii. Clasificar dichas funciones en tres grupos:
 - Imprescindibles
 - Muy deseables
 - Deseables
- iii. Verificar que las alternativas satisfagan todas las funciones imprescindibles, descartando las que no lo hagan
- iv. Si existe más de una alternativa que cumpla el criterio anterior, generar la siguiente matriz (las funciones indicadas lo son a modo de ejemplo):

Funcionalidades del Sistema	Altern. 1	Altern. 2	...	Altern. N
MUY DESEABLES (%Cumplimiento)	100%	50%		100%
Información en Línea	1	1		1
Interfaces Gráficas	1	0		1
DESEABLES (%Cumplimiento)	33%	100%		33%
Emisión de cartas	0	1		0
Control de cambios	1	1		1
Otros atributos menores	0	1		0
Total	79.9 (EF1)	65 (EF2)		79.9 (EFN)

Tabla 5: Matriz de Efectividad

Para otros ejemplo de atributos que miden efectividad en distintas dimensiones, ver Anexo 8

En cada celda, colocar un 1 (uno) si la alternativa cubre la función y un 0 (cero) en caso contrario. Posteriormente se calcula el % de cumplimiento para c/alternativa, como la suma de 1 (unos) divididos por el total de atributos dentro de c/categoría (deseable o muy deseable) . Luego, obtener el puntaje de cada alternativa por grupo de función y calcular el promedio ponderado de ambas. Se utilizará un factor de 0.7 para las funciones “muy deseables” y de un 0.3 para las funciones “deseables”. En el ejemplo, utilizando dichos ponderadores, los puntajes (multiplicados por 100) son 79.9 ($0.7*100+0.3*33$), 65 ($0.7*50+0.3*100$) y 79.9 respectivamente.

Los valores sugeridos para ponderar las funciones “muy deseables” y las “deseables” pueden ser modificados, incluyendo la justificación por hacerlo.

B) Plataforma Tecnológica

En este factor se busca capturar que la solución esté basada en un conjunto de herramientas que permitan, con una alta probabilidad de éxito, la construcción de un sistema que satisfaga los siguientes criterios:

- **Confidencialidad:** debe evaluarse el nivel de protección que cada alternativa ofrece contra la divulgación no autorizada de la información. En ésta, deberán considerarse aspectos como:
 1. Sistema operativo
 2. Base de datos
 3. Conexión con otros sistemas de información (a través de Internet o localmente)
 4. Acceso a medios de respaldo
- **Integridad:** está relacionado con la precisión y suficiencia de la información. También con la validez de la información.

- **Disponibilidad:** las alternativas de solución deben proveer:
 1. Acceso a la información por parte de todos los usuarios autorizados, en el momento en que lo requieran
 2. Tiempos de respuesta acordes con las necesidades de los procesos
- **Confiabilidad de la información (Gestión):** Esto tiene que ver con que la información obtenida debe ser apropiada para la gestión con el fin de operar la institución y para ejercer las responsabilidades de cumplimiento de las tareas institucionales.
- **Información Externa:** Esto tiene que ver con que la información obtenida debe ser apropiada para satisfacer los requerimientos de otras instituciones con respecto a la organización.

En lo posible, cada uno de estos criterios deberá ser evaluado objetivamente. En todo caso, la existencia de opiniones de expertos podrá ser incorporada, así como estadísticas que exhiban una validación de la industria informática respecto al cumplimiento de cada uno de ellos. De todos modos, cada criterio será calificado con una nota de 1 a 100, en base al siguiente criterio:

Si cumple totalmente:	100 puntos
Si cumple adecuadamente:	80 puntos
Si cumple con restricciones:	60 puntos
Cumple con muchas restricciones:	40 puntos
Si no cumple:	0 puntos

Con el fin de obtener todos los antecedentes necesarios para la evaluación, el formulador se deberá apoyar en la información del anexo2 que sea relevante para la tipología del proyecto. Una vez hecho esto, se deberá elaborar la siguiente matriz.

ASPECTOS PLATAFORMA TECNOLÓGICA	Ponderador	Altern. 1	Altern. 2	...	Altern. N
Confidencialidad	x%	100	100		40
Integridad	X%	100	20		100
Disponibilidad	Y%	100	100		30
Confiabilidad	Z%	80	100		100
Información Externa	W%	80	80		100
TOTAL	100%	PT1	PT2		PTN

Tabla 6: Matriz de Evaluación Plataforma Tecnológica

En base a estos resultados, se debe calcular un promedio ponderado. Para calcular los ponderadores, se debe aplicar la técnica descrita en el anexo 1.

C) Calidad Técnica

Este punto tiene que ver con aspectos técnicos de la solución propiamente tal, más allá de la plataforma en la cual se basa. El objetivo es asegurar que la implementación de las herramientas disponibles en la plataforma tecnológica seleccionada cumpla con los criterios deseados. Para estos efectos, se deberá crear una matriz con todos los aspectos técnicos evaluables de las alternativas, clasificándolos en los siguiente grupos:

Seguridad : Da cuenta de la seguridad de la solución tanto en los ámbitos de hardware como de software

Disponibilidad: Se refiere a la capacidad de la plataforma de no sufrir caídas dentro de un rango de tiempo determinado

Portabilidad: Compatibilidad con otras plataformas, en cuanto a hardware y software

Accesibilidad: Se refiere a la disposición de la plataforma, para ser accesada desde otra.

Escalabilidad: Factibilidad de hacer crecer el sistema por etapas

En cada celda se debe colocar un 1 si la alternativa cumple con el aspecto técnico y un 0 si no. Luego se debe obtener el porcentaje de cumplimiento de cada uno de los cuatro grupos de aspectos técnicos. En base a estos porcentajes se calcula un promedio ponderado para cada alternativa. Los ponderadores deben calcularse de acuerdo a lo indicado en el anexo 1.

A continuación se presenta un ejemplo. Para un listado más completo de aspectos técnicos, se puede ver el punto 3 del **Anexo 2:** Elementos a considerar en la Evaluación Técnica de Proyectos Informáticos.

ASPECTOS TÉCNICOS SISTEMA	Ponderador	Altern. 1	Altern. 2	...	Altern. N
SEGURIDAD (% cumplimiento)	X%	%100	%75		%50
Sistemas de Respaldos		1	1		1
Sistema de recuperación		1	1		1
Control de acceso		1	1		0
Encriptación de datos		1	0		0
PORTABILIDAD (%Cumplimiento)	Y%	%100	%100		%100
Herramientas para importación y exportación de datos.		1	1		1
DISPONIBILIDAD	Z%	%0	%100		%100
Up time garantizado de más de 98%		0	1		1
ESCALABILIDAD	W%	%100	%100		%100
ACCESIBILIDAD (%Cumplimiento)	U%	%0	%100		%0
Canales de comunicación en línea con otras aplicaciones		0	1		0
TOTAL		CT1	CT2		CTN

Tabla 7: Matriz de Evaluación Calidad Técnica

D) Ahorro de costos operacionales

Una variable importante al tomar la decisión de cuál alternativa elegir, o si conviene implementar la solución, es el ahorro que producirá con respecto a los costos operacionales que tenía la institución antes de llevar a cabo el proyecto.

El ahorro de costos se puede obtener recurriendo al Anexo 3 y verificando en qué ítems se produce ahorro. Lo importante es que en este punto se incluyan los ahorros de costos operacionales que **objetivamente son cuantificables**.

Se deberá usar la siguiente formula:

$$\frac{AC_j}{CO_j}$$

Donde:

AC_j: Ahorro de costos operacionales con proyecto en la alternativa j

Co_j: costos operacionales para alternativa j

Se considera que el máximo ahorro en costos operacionales puede llegar a ser del 10%. Para llevar esto a puntaje, se amplificará por 1000 el porcentaje obtenido. Así si el ahorro fuera del 10% el puntaje sería 100. Si el ahorro fuera del 4%, el puntaje sería de 40.

	Altern. 1	Altern. 2	...	Altern. N
Ahorro de papel	100	100		40
Ahorro en promoción(marketing)	100	100		100
Ahorro en distribución de la información	100	100		100
Ahorro en reparaciones	100	100		100
:	:	:	:	:
:	:	:	:	:
etc	:	:	:	:
SUMA				
Promedio	ACO1	ACO2		ACON

Tabla 8: Matriz de Ahorro en Costos Operacionales

Para calcular el Total, se suma el puntaje de la columna correspondiente y posteriormente se divide por el número de atributos contenidos en la tabla.

Ejemplo:

	Altern. 1	Altern. 2	Altern. 3
Ahorro de papel	20	30	10
Ahorro en promoción(marketing)	30	80	50
Ahorro en distribución de la información	100	40	80
Ahorro en reparaciones	10	70	90
SUMA	160	220	230
Promedio	40	55	57,5

Tabla N° 9: Ejemplo de ahorro de costos

En este caso los atributos son 4, por lo que se divide la suma de cada columna por ese número, para obtener el total.

Evaluación de alternativas

Una vez evaluados todos estos factores, se deberá generar la siguiente matriz:

ATRIBUTOS EVALUABLES	Ponderador	Altern. 1	Altern. 2	...	Altern. n
Efectividad	x%	EF1	EF2		EFN
Plataforma Tecnológica	y%	PT1	PT2		PTN
Calidad Técnica	z%	CT1	CT2		CTN
Ahorro de Costos Op.	w%	ACO1	ACO2		ACON
TOTAL	100%	P1	P2		PN

Tabla N°10: Matriz de Evaluación de alternativas

Los ponderadores se obtienen aplicando el criterio indicado en el anexo 1. Finalmente, se aplica la siguiente fórmula:

$$P_i = \sum_j \frac{PA_{ji} * Ponderador_j}{100}$$

Donde:

Pi	:	Puntaje Alternativa i
PA _{ji}	:	Puntaje del atributo j de la alternativa i
Ponderador _j	:	Ponderador del atributo j (corresponde a los x%, y%, z% y w%)

Los puntajes se pueden interpretar de la siguiente forma:

Cumple totalmente:	100 puntos
Cumple adecuadamente:	80-99 puntos
Cumple con restricciones:	60-79 puntos
Cumple con muchas restricciones:	40-59 puntos
No cumple:	0-39 puntos

Una vez obtenida una calificación para cada una de las alternativas, es posible el cálculo de una razón costo / eficiencia que incorpora criterios estratégicos y de calidad.

4.3.2 Detalle de la Inversión y cálculo del indicador costo eficiencia

Los ítems de costos de inversión y operación, están identificados en el Anexo 3 de Beneficios y costos¹⁰.

Si todas las alternativas tuvieran costos similares, podría bastar con los puntajes para decidir una selección. En caso contrario, se deberá hacer un análisis en base al indicador costo – eficiencia, el que está definido como:

$$RC_j = \frac{C_j}{P_j}$$

RC_j : Razón de costo - eficiencia de la alternativa j, (costo por unidad de cumplimiento de los objetivos)

C_j : Costo Anual Equivalente de la Alternativa j

P_j: : Puntaje de la alternativa j

Para calcular C_j, se calcula el Costo anual equivalente (CAE) del proyecto dentro de su vida útil considerando los costos de inversión, mantención y operación. El CAE se calcula como el producto del Factor de Recuperación del Capital (FR) por el Valor Actual de Costos de la alternativa j (VAC_j), donde;

¹⁰ No se debe olvidar en el caso de adquisición de software, incluir los costos asociados al número de licencias de uso que se desea habilitar.

$$FR = \frac{r (1+r)^n}{(1+r)^n - 1}$$

$$VAC_j = I_j + \sum_{t=1}^n (CO_{tj} + CM_{tj}) / (1+r)^t$$

Con:

r: tasa de descuento

n: vida útil del sistema

CO_{tj}: costo de operación de la alternativa j en el período t

CM_t: costo de mantenimiento de la alternativa j en el período t

I_j: costo de inversión de la alternativa j

De forma que

$$C_j = CAE_j = VAC_j * FR$$

Para los sistemas se considera generalmente una vida útil de cuatro años y una tasa social de descuento del 10 %, con lo cual se obtiene: Factor de Recuperación = 0,3154. En el caso que se estime que la vida útil de alguna alternativa tecnológica difiere significativamente de 4 años se deberá recalcular su FR de acuerdo a la fórmula anterior.

Para escoger la o las alternativas finales, se seleccionan aquellas con menor razón de costo-eficiencia y cuyo precio esté dentro de los rangos presupuestados.

En el caso de que las alternativas tuvieran la misma razón costo-eficiencia, hay que escoger la que cumpla al menos con restricciones.

Ejemplo:

A continuación, se presenta un ejemplo de la aplicación de la tabla de atributos evaluables.

ATRIBUTOS EVALUABLES	Ponderad or	Altern. 1	Altern. 2	Altern. 3
Efectividad	30%	100	80	60
Plataforma Técnica	25%	60	100	100
Calidad Técnica	35%	40	80	60
Costos operacionales	10%	50	40	60
TOTAL	100%	64	81	70

Tabla N° 11: Ejemplo de resultado de evaluación

En este caso, el resultado del análisis de alternativas es el siguiente:

- Alternativa 1: Cumple con restricciones
- Alternativa 2: Cumple adecuadamente
- Alternativa 3: Cumple con restricciones

5 Términos de referencia para la etapa de ejecución

Los términos de referencia deben incluir toda la información necesaria, para poder licitar la etapa de ejecución. Deben especificar que la solución se debe adaptar al diseño lógico ya desarrollado, así como a los distintos atributos desarrollados en esta metodología. Por otro lado deben establecerse claramente los productos entregables (códigos fuentes, documentación del diseño físico, otros), también debe especificarse el tiempo de entrega de la solución y que hacer en caso de atraso. Puntos importantes a considerar son la marcha blanca, capacitación y la operación, mantención del sistema.

6 Sugerencias para el proceso de licitación

Además de los resultados arrojados por el análisis técnico - económico, es fundamental tomar en cuenta una serie de características que debe poseer un proveedor de equipos computacionales y que se recomienda tener en cuenta, en conjunto con los resultados de la preparación y evaluación del proyecto obtenidos hasta las etapas anteriores, para la elaboración de las bases de licitación.

Capacitación

Se debe considerar si el proveedor entrega:

- Capacitación en todos los niveles requeridos, incluyendo los ámbito de operación, ingeniería de sistemas, etc.
- Existencia de un centro de Capacitación.
- Plan de cursos regulares.

Mantenimiento

Se debe considerar si la propuesta del proveedor contempla:

- Responsabilidades como mantenimiento y reparación de unidades del sistema.
- Inspecciones, pruebas, limpieza de partes internas, lubricaciones, ajustes y reemplazo de partes.
- Período de garantía.

- Mantenimiento preventivo.
- Ingeniería de Sistemas con Soporte Técnico, Desarrollo de Aplicaciones y Consejería Informática.

Respaldo y prestigio

Se debe considerar si el proveedor es capaz de ofrecer garantías como cumplimiento de plazo de entrega, soporte ante fallas, ampliaciones, compatibilidades con otros equipos, lo que constituye el respaldo del proveedor.

Los indicadores más significativos del prestigio son: presencia en el mercado, nómina de equipos instalados y montos que estos representan, rentabilidad, liquidez, razón deuda patrimonio, etc.

Como es sabido, las bases de una licitación se conforman a grandes rasgos por las bases técnicas y las bases administrativas.

7 Bibliografía

- Eduardo Contreras y otros. Metodología de informática de MIDEPLAN 1992.
- Information Systems Audit and Control Foundation (ISACF). <http://www.isaca.org>
Resumen Ejecutivo de COBIT (Objetivos de control para la información y tecnologías)
2da Edición Abril de 1998. Copyright 1996, 1998
- Information Systems Audit and Control Foundation (ISACF).
Marco Referencial de COBIT(Objetivos de control para la información y tecnologías) “2da Edición Abril de 1998. Copyright 1996, 1998
- Information Systems Audit and Control Foundation (ISACF).
Objetivos de control de COBIT(Objetivos de control para la información y tecnologías)
“2da Edición Abril de 1998. Copyright 1996, 1998
- Tecnologías de la Información y su uso en Gestión de Oscar Barros V. Primera edición
Copyright 1998 McGraw-Hill.
- Página <http://www.ji.si.ehu.es/users/tap/ADSI/19992000/Tema1/>

Anexo 1: Técnica para priorización y asignación de ponderadores

Dada una lista de ítems a los cuales hay que clasificar para poder determinar el nivel de importancia entre ellos, se debe proceder del siguiente modo:

1. Determinar los ítems relevantes a clasificar y asignarles una identificación.
2. Colocar la identificación asignada en las filas y columnas de la matriz de la página siguiente (la matriz tiene la misma cantidad de filas y columnas).
3. Colocar alguna marca en la diagonal de la matriz (sobre la diagonal no habrá ninguna clase de información).
4. Completar cada una de las celdas por sobre de la diagonal respondiendo a la siguiente pregunta: El ítem de la fila, ¿es más importante que el ítem de la columna? Si la respuesta es afirmativa, se debe colocar un 1 en la celda, en caso contrario, un 0. En el ejemplo, el ítem 1 (en la fila) es más importante que el ítem 2 (en la columna) y por este motivo se coloca un 1 en la celda.
5. Cuando todas las celdas de una fila (por encima de la diagonal) están completas, las celdas de la columna correspondiente al mismo ítem se deben llenar con el inverso del número (donde hay un 1 se coloca 0 y viceversa).
6. Cuando todas las celdas están llenas, se las suma obteniéndose el total de la fila.
7. Luego de calcular los totales por fila, se asigna un número de orden a aquella fila cuyo total es el mayor (un 1) y así sucesivamente siguiendo en forma decreciente de importancia.
8. Si dos de los totales son iguales, se asigna mayor prioridad al ítem que la tiene con respecto al otro. En el ejemplo, puesto que a los ítems 3 y 7 les corresponde el mismo total (en este caso 3), debido a que el ítem 3 es más importante que el ítem 7, se le asigna al primero el número de orden 3 y al ítem 7, el número de orden 4.
9. En la columna "Orden" se obtiene la secuencia de ítems con su prioridad, uno respecto del otro.

		... que el ítem que está en la columna?							Total fila	Orden
		Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7		
¿El ítem que está en esta fila es más importante...	Item 1		1	0	0	1	1	1	4	2
	Item 2	0		1	0	1	0	0	2	6
	Item 3	1	0		0	0	1	1	3	3
	Item 4	1	1	1		1	1	1	6	1
	Item 5	0	0	1	0		0	0	1	7
	Item 6	0	1	0	0	1		0	2	5
	Item 7	0	1	0	0	1	1		3	4

10. Ordenar la lista de ítems de acuerdo al resultado obtenido, asignando los ponderadores en forma tal que lo satisfagan y que su suma sea 100%. Como referencia, puede utilizarse el porcentaje que representa el puntaje obtenido por un ítem con respecto a la suma de la columna "Total fila". En el ejemplo, el ítem 4 obtendría un ponderador de 28,6% (6 dividido por 21). En todo caso, debe tenerse presente que la importancia relativa de un ítem respecto a otro incorpora elementos subjetivos, por lo cual los ponderadores definitivos deben ser corregidos considerando dichos elementos, pero siempre respetando el orden obtenido.

Un ejemplo de esta metodología se presenta a continuación, para el caso del cálculo de la matriz de plataforma tecnológica.

Cálculo de ponderadores para evaluar la plataforma tecnológica:

	Confidencialidad	Integridad	Disponibilidad	Confiabilidad (Gestión)	Inf. externa	Total Fila	Orden	Ponderador %
Confidencialidad		0	0	1	0	1	4	10
Integridad	1		1	1	1	4	1	40
Disponibilidad	1	0		0	1	2	2	20
Confiabilidad (Gestión)	0	0	1		0	1	5	10
Inf. externa	1	0	0	1		2	3	20

Matriz de Evaluación de la plataforma tecnológica

ASPECTOS PLATAFORMA TECNOLÓGICA	Ponderador	Alternativa 1	Alternativa 2	Alternativa 3
Confidencialidad	10%	60	100	80
Integridad	40%	70	100	90
Disponibilidad	20%	60	100	70
Confiabilidad (Gestión)	10%	100	80	80
Inf. externa	20%	100	100	60
TOTAL	100%	76	98	78

Anexo 2: Elementos a considerar en la Evaluación Técnica de Proyectos Informáticos

Los siguientes son los antecedentes a entregar para la evaluación técnica de un proyecto informático:

1. Estimación de volúmenes de información
2. Arquitectura lógica de la solución
3. Arquitectura de hardware
4. Herramientas a utilizar para la construcción de la solución
5. Costos de operación
6. Costos de mantención
7. Necesidades de capacitación técnica
8. Personal nuevo necesario

A continuación, se detalla cada uno de ellos:

Estimación de volúmenes de Información

Se deben entregar estimaciones para los siguientes indicadores:

- a) Información de la base de datos:
 - Número de tablas
 - Tamaño de registro de cada tabla
 - Número de índices por tabla (permanentes y temporales)
 - Número de registros por tabla
 - Crecimiento (anual, mensual) esperado
- b) Número de clientes
- c) Número de transacciones (total o por cliente) por tipo (actualizaciones, consultas)
- d) Crecimiento esperado de clientes
- e) Crecimiento esperado de transacciones (total o por cliente)
- f) Número de procesos masivos

Arquitectura Lógica de la Solución

Se debe indicar si la solución se basará en una arquitectura cliente - servidor, tecnología Internet, aplicaciones “*stand-alone*”, etc. Deberá justificarse la arquitectura seleccionada en términos funcionales y otras consideraciones que se estimen relevantes. Particular relevancia tienen consideraciones de carácter estratégico de la institución.

Arquitectura de Hardware y Software Básico

Detallar, con diagramas y cuadros descriptivos, la solución en términos de sus componentes de hardware y software básico:

a) Servidores

- CPU
- Unidades de almacenamiento (tipo, capacidad)
- Memoria RAM
- Unidades de respaldo
- Periféricos
- Otros (hardware redundante, etc.)

b) Sistemas operativos

- Fabricante
- Versión (número y fecha liberación)
- Tipo de licenciamiento

c) Software de aplicación de terceros

- Fabricante
- Versión (número y fecha liberación)
- Tipo de licenciamiento

d) Componentes de red y comunicaciones (hardware y software)

- Servidores de comunicaciones
- Routers, modems, DTU, etc.
- Protocolos de comunicación

e) Estaciones de trabajo

- CPU
- Unidades de almacenamiento (tipo, capacidad)
- Memoria RAM
- Unidades de respaldo

- Periféricos
- f) Impresoras
- Tipo de impresoras (láser, inyección de tinta, etc.)
 - Breve descripción características técnicas (calidad de impresión, velocidad, etc.)
- g) Otros dispositivos (scanners, capturadores ópticos, dispositivos de video, etc.)

Herramientas a utilizar para la construcción de la solución

Detallar el software a utilizar para la construcción de la solución:

- a) Base de datos
- b) Herramientas de productividad
- c) Aplicaciones clientes
- d) Otros

Costos de Operación

Debe estimarse el costo de operación de la solución y la curva de evolución de éste, con el fin de predecir la vida útil de la solución. Aquí deben considerarse:

- a) Insumos y materiales físicos
- b) Recursos humanos
- c) Otros

Costos de Mantenimiento de la Solución

Esta información complementa la anterior, debiendo incluirse:

- a) Upgrade o mantención de licencias
- b) Actualizaciones de hardware
- c) Proyección de requerimientos de nuevos desarrollos
- d) Otros

Capacitación Técnica

- a) Personal involucrado
- b) Costo de capacitación (inicial y mantención)

Personal nuevo necesario

Indicar eventual necesidad de contratación de personal para la operación y mantención de la solución, describiendo perfil y costos.

Anexo 3: Beneficios y costos

Los costos de los proyectos de informática son relativamente simples de cuantificar, no así los beneficios, que se presentan como ahorro de costos con respecto a la situación base, siendo particularmente compleja la estimación de las horas - hombre liberadas.

Por otra parte, este tipos de proyectos tienen costos y beneficios intangibles, los cuáles se deberán describir en forma cualitativa.

Beneficios privados

Dependiendo de la naturaleza del proyecto, se pueden presentar algunos de los siguientes beneficios:

Ahorro de Horas-Hombre (H-H)

Por no tener que contratar personal adicional con respecto a la situación optimizada.

Se considera como situación base optimizada (sin proyecto) la contratación de personal adicional que permitiría alcanzar los mismos objetivos que la configuración computacional; es decir, la alternativa de sustitución de recursos de capital por trabajo. Este beneficio lo es en la medida que exista dicha alternativa.

Del personal que actualmente labora en el sistema.

Este beneficio lo es bajo el supuesto de que las H-H liberadas tengan un uso alternativo productivo. Si la alternativa es el ocio, en el caso de que con el proyecto disminuyan los requerimientos diarios de H-H, tendríamos sólo un beneficio individual difícil de valorar. Este ahorro de H-H corresponde a un aumento de la productividad.

Tipos de aumento de productividad

El aumento de la productividad puede provenir de tres tipos:

Ahorro del tiempo de desplazamiento

Con el nuevo sistema, se pretende reducir o eliminar el tiempo que las personas gastan en desplazarse para intercambiar información o para realizar alguna acción que pudiera ser llevada a cabo desde su escritorio.

Como ejemplos típicos se tiene:

Entrega de información vía diskette.

Ir a colocar papel a una impresora compartida.

Levantarse a buscar información escrita.

Mejora del actual sistema

Con el nuevo sistema, se pretende mejorar las características básicas del sistema actual.
Por ejemplo:

Aumentar la robustez del sistema.
Hacer más rápido el acceso y procesamiento de la información.
Reducción de tiempo de espera en colas de impresión.

Automatización

En este caso, mediante la implantación de un sistema computacional se pretende lograr que una tarea deje de ser manual para ser automatizada.

Como por ejemplos típicos se tiene:

Ordenamiento de archivos.
Generación automática de cheques.
Búsqueda de información.

El primero de estos tres tipos de aumento de la productividad se produce principalmente en proyectos que involucran mejoras en las comunicaciones.

El segundo tipo se produce con mayor frecuencia en proyectos que involucran un aumento de la capacidad de procesamiento y un mejoramiento del diseño del sistema.

El tercer tipo de aumento de la productividad está relacionado con proyectos que formulan el equipamiento de un sistema computacional por primera vez en alguna área determinada.
Venta de información

Al implementar el proyecto se deben cuantificar los servicios que la institución puede ofrecer hacia el exterior. Como ejemplo se tiene: venta de software, venta de información, etc.

Ahorro en arriendo de oficinas

En el caso en que se esté arrendando una oficina que ya no se va a necesitar una vez adquirido el equipo computacional, se cuenta como ahorro el monto de dicho arriendo. También ocurre cuando se traspasa a medios magnéticos la información antes contenida en archivos y carpetas.

En el caso en que la oficina sea de propiedad de la institución que adquiere el equipo, el ahorro proviene del uso alternativo que se le puede dar a esta oficina.

Ahorro en costos de operación

Se refiere a ahorros en costos de operación, con respecto a situación base. A modo de ejemplo, una disminución de los costos de mantención; o bien, dejar de pagar por servicios a empresas, pues con la realización del proyecto estos servicios podrán desarrollarse internamente. Se debe mencionar el detalle de cada uno de los costos de operación que van a disminuir o bien desaparecer, acompañado por el monto anual del ahorro que se produce al adquirir el equipo.

Mejoras en la gestión y en la toma de decisiones

Este tipo de beneficios son frecuentes, pero generalmente de muy difícil cuantificación, lo que puede en ocasiones llevar a que se consideren sólo como intangibles, o bien, como el primer tipo de ahorro de H-H antes expuesto, es decir, del personal adicional que se requeriría para obtener el mismo efecto de mejora en la gestión y la toma de decisiones.

Es importante tener presente no cometer el error de contabilizar más de una vez algún beneficio. Para ello debe ponerse atención al clasificarlo. Por ejemplo, si se usa el método de estimar el ahorro de H-H adicionales equivalentes para alcanzar la misma mejora en la gestión que logra el equipamiento computacional, no debe considerarse como un beneficio adicional del proyecto dicha mejora de la gestión.

Costos privados

En general tendremos los siguientes ítems:

- Compra de hardware
- Compra de software
- Conversión/Adaptación de software existente
- Desarrollo de software
- Estudios y capacitación
- Instalación y puesta en marcha
- Habilitación de locales y muebles

Costos de operación

- Remuneraciones (cuando se requiera personal adicional)
- Servicios externos
- Comunicaciones (arriendo de líneas)
- Arriendo de programas
- Materiales de uso y consumo corriente (diskettes, hojas perforadas, cintas de impresoras, etc.)
- Mantención y reparaciones
- Consumo de energía

Beneficios y costos sociales

Para el caso social, la estimación de beneficios y costos es similar al caso privado. Sólo deben hacerse ciertos ajustes a los costos y beneficios privados de modo que representen en forma adecuada los beneficios y costos sociales.

Ahorro de tiempo de usuarios que realizan trámites en la Institución.

En este caso es necesaria la realización de una encuesta o un estudio durante un tiempo determinado, para medir la frecuencia media de público que llega. Para poder cuantificar este beneficio se debe hacer una estimación de la calificación de las personas que realizan los trámites, para luego calcular un sueldo promedio por unidad de tiempo.

Los supuestos de mantención de frecuencias y composición de la clientela deben quedar explícitamente indicados.

Beneficios y costos intangibles (no valorables)

Se debe entregar un listado que incluya aquellos costos y beneficios que no se pudieron valorar. Típicamente, se tratan de los siguientes:

Costos

Como ejemplo: resistencia al cambio, problemas organizacionales por la introducción de computadores, cambios en las políticas de la organización, retrasos en la entrega por parte de los proveedores.

Beneficios

Los beneficios intangibles, corresponden a aquellos, cuya valoración económica es difícil de obtener. Estos pueden corresponder a mayor comodidad de los usuarios, mejor imagen de la institución, mejoramiento de las condiciones de trabajo para los funcionarios, etc.

Arriendo y leasing

En el caso que se adopte por un arriendo o un leasing, es necesario que se justifique esta opción, versus la inversión. La justificación tiene que ser económica o cualitativa.

Anexo 4: Modelamiento de datos

TÉCNICA DE MODELAMIENTO DE DATOS

I DEFINICIÓN

Conjunto de entidades y relaciones que representan los datos de una organización o sistema bajo análisis.

Es construido en forma "bottom-up" a partir de los requerimientos establecidos para el producto o servicio en desarrollo. Esto implica que se hace siempre necesario un estudio previo de la organización o sistema que permita identificar y definir los requerimientos, y a partir de ellos generar un MODELO que describa qué datos participan en el sistema y cómo estos interactúan entre sí.

II. ESTRUCTURA DE LOS DATOS

2.1 ESQUEMA

En la figura 1 se representa una serie de secuencia de pasos a seguir, desde el punto de vista metodológico, para hacer modelamiento de datos:

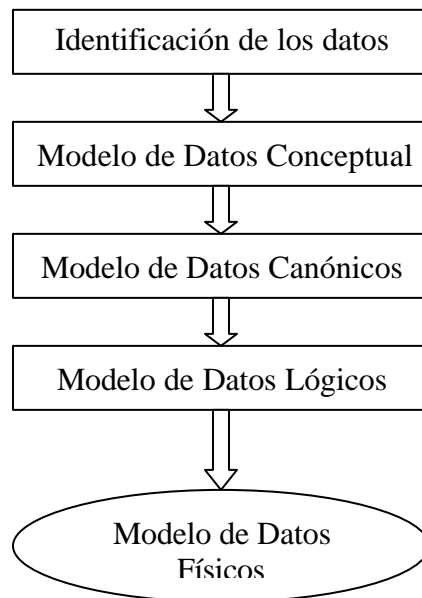


Figura 1: Esquema metodológico

2.2 IDENTIFICACIÓN DE REQUERIMIENTOS

Consiste en buscar las fuentes de información, es decir, elementos que entreguen antecedentes sobre el sistema o "problema" en estudio, con el fin de obtener información para la definición de entidades relacionadas.

Se consideran los siguientes puntos:

- i) Definir qué hace el sistema: funciones - actividades - tareas
- ii) Definir o recolectar los documentos: cheques, letras, pagarés, escrituras, reglamentos, etc. Éstos pueden formar una entidad por sí mismos o formarán parte de alguna.
- iii) Definir o recolectar formularios: todos los formularios que se utilizan en el Sistema en estudio.
- iv) Considerar los manuales de procedimientos administrativos.
- v) Considerar los requerimientos al interior del sistema de Información administrativo (SIA):
 - SALIDAS (informes, listados, etc)
 - ENTRADAS (registros de datos, etc)
 - TRANSACCIONES
- vi) Considerar entrevistas, información narrativa producto de definiciones, encuestas, libros, etc.

Una vez "terminada" (el proceso es esencialmente interactivo) la identificación de los requerimientos, se procede a lo que se conoce como ANÁLISIS DE DATOS, actividad que contempla las etapas de modelamiento y refinación de los datos. Para ello se parte definiendo un modelo inicial (MODELO CONCEPTUAL DE DATOS), luego un modelo refinado (MODELO CONCEPTUAL CANÓNICO) y, finalmente, se sobreponen al último modelo las especificaciones de requerimientos para verificar que puedan ser satisfechos por el modelo (se obtiene el MODELO LÓGICO DE DATOS).

2.3 MODELO DE DATOS CONCEPTUAL

Está formado por entidades y las relaciones entre ellas. A continuación se describen cada uno de estos elementos.

2.3.1 ENTIDADES

Corresponde al objeto que se quiere representar. Son elementos, reales o abstractos, acerca de las cuales se requiere registrar información.

Ejemplo: CLIENTE, EMPLEADO, CUENTA, PROVEEDOR, SERVICIO, CANAL, SEGMENTO, PRODUCTO, etc.

Una **entidad** es una clase de "cosas" con un nombre, que tiene el mismo conjunto de atributos: por ejemplo EMPLEADO es una entidad. Una instancia de entidad es una ocurrencia específica de la entidad: por ejemplo, PEDRO PÉREZ Z. es una instancia de la entidad.

La representación de una entidad será, en principio, un rectángulo. A futuro, el ícono particular que sea definido como estándar dependerá de la elección y capacidades de representación de una herramienta CLASE específica.

2.3.2 RELACIONES

Una relación es una asociación entre, exactamente, dos entidades. Por ejemplo, una PERSONA (entidad) trabaja para un DEPARTAMENTO (entidad).

2.3.3 ENFOQUE DE CONSTRUCCIÓN MODELO DE DATOS CONCEPTUAL

a. Identificar y Evaluar Entidades Candidatas

Las entidades son descubiertas a través de situaciones como las siguientes:

- Considerando sustantivos (o nombres de cosas) encontrados durante entrevistas.
- Considerando datos en documentos existentes. Esto a menudo indica la necesidad de información que debe ser registrada.
- Mientras se describe una entidad ya descubierta. Aparecen aspectos no considerados al inicio, que permiten la identificación de nuevas clases de datos que son relevantes para el negocio.
- Mientras se normaliza el modelo de datos conceptual (ver punto 2.4 de este documento).

Los siguientes factores se aplican a todas las entidades:

- Es posible distinguir entre diferentes instancias (ocurrencias) de una entidad. Por ejemplo, muchas empresas no dan importancia a diferenciar entre diferentes lápices, por lo que LÁPIZ no es una entidad. Sin embargo, es crítico distinguir entre diferentes clientes, por lo que CLIENTE sí es una entidad.

- Es posible registrar hechos que describen las instancias de una entidad. Por ejemplo, los siguientes hechos se aplican a las ocurrencias de la entidad PROVEEDOR: Nombre_proveedor, dirección_proveedor, fecha_ingreso_proveedor, calidad_proveedor, vigencia_proveedor.
- Cada hecho puede tomar sólo un valor para cada ocurrencia de la entidad. Por ejemplo si un proveedor tiene más de un vendedor para contactar la empresa, luego el nombre del vendedor es un hecho multivaluado acerca del proveedor. De esto se sigue que más de una entidad es necesaria para descubrir completamente al proveedor.

Nombres para las entidades:

El nombre de una entidad deberá ser un nombre comúnmente usado dentro de la organización. La única excepción será cuando diferentes partes de la organización usan ya sea diferentes términos para el mismo objeto, o cuando se usa el mismo nombre para más de un concepto. Estos casos corresponden, respectivamente a sinónimos, y homónimos.

Sinónimos- un ejemplo de sinónimo es: el resultado de una operación con el cliente, en respuesta a una orden, puede ser llamado, por diferentes departamentos, "venta", "orden completa", o "entrega". Después de confirmar que existe sólo una entidad para los tres casos, es mejor elegir un nombre neutro, como por ejemplo "DESPACHO".

Homónimos- un ejemplo de homónimo es: una orden de trabajo (las instrucciones de fabricación) y una orden de compra (el pedido de un cliente) pueden ser referenciados en diferentes partes de la organización simplemente como "orden". Tales homónimos indican la inmediata necesidad de más de una entidad. Ninguna de las entidades deberá ser referenciada sólo con el término simple común. En el ejemplo anterior, se debiera usar nombres como ORDEN COMPRA Y ORDEN TRABAJO para separar los casos.

Los nombres de las entidades deberán ser escritos en modo singular y con letras mayúsculas, por ejemplo "CLIENTE", y no usar "CLIENTES" o "cliente".

Muchas veces será necesario utilizar un calificador u otra palabra agregada al sustantivo para aclarar el alcance que tiene el nombre de la clase de datos identificada. Por ejemplo, usar "CONTRATO DE LEASING" en vez de sólo "CONTRATO". Por supuesto, esto dependerá del contexto global y de las características particulares del sistema bajo estudio.

b. Identificar y evaluar relaciones entre entidades

Las relaciones a menudo son encontradas en circunstancias como las siguientes:

- Considerando los verbos anotados durante las entrevistas.
- Examinando documentos y formularios.

Describiendo entidades y relaciones ya existentes.

Una relación debe ser tanto **relevante** como **significativa** para la organización. Por relevante se entenderá el hecho de que es necesario registrar la asociación, y por significativa se entenderá el que aporte el entendimiento y uso del modelo. Estos dos factores aparecerán más claros al observar el uso del modelo en los procesos del sistema.

Nombres para las relaciones

Las relaciones deben tener nombre con frases que sean **significativas para la asociación entre las dos entidades**. Por ejemplo, PROVEEDOR provee PRODUCTO. La frase de asociación casi siempre tendrá un verbo presente. La forma activa del verbo denotará el primer sentido de asociación (primero en términos de que es el primer sentido estudiado), y la forma pasiva del verbo denotará la asociación inversa, por ejemplo, PRODUCTO provisto por el PROVEEDOR.

Es muy importante evitar nombres generales tales como "es", "tiene", "con ", que aportan poco a aclarar el significado de una relación. Por ejemplo, ORDEN contiene (no "con") ITEMS ORDEN, o CIUDAD ubicación de (no "tiene) HOTEL.

c. Construir Diagrama Entidad - Relación

Las entidades y relaciones se van dibujando a medida que se van descubriendo, por lo que el diagrama va siendo modificado continuamente a medida que se van encontrando nuevas entidades y relaciones.

Una entidad es dibujada como un rectángulo, que contiene el nombre de la entidad, escrito en mayúscula, tal como aparece en la figura 2:



figura 2: Diagrama de una entidad

Una relación se dibuja como una línea entre dos entidades, con el nombre de la relación escrito en minúsculas sobre esta línea, tal como en la figura 3:

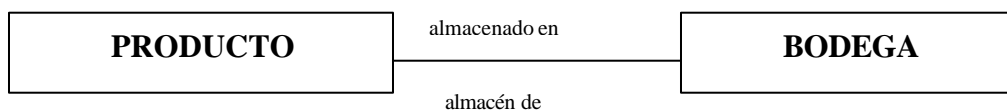


Figura 3: Diagrama de una relación

Las relaciones tienen una naturaleza que describe el tipo de asociación, tales como:

- Pertenencia
- Jerarquía
- Temporal
- Complementariedad
- Parentesco
- Autoridad

Adicionalmente, interesa la participación de las entidades involucradas en la relación, también conocida como **atributos de existencia de la relación**, que se representa como los símbolos "0" (cero) ó "1" (uno), ubicados en los extremos de la línea de la relación.

Un "1" representa que una ocurrencia de una entidad **debe** estar asociada con cada ocurrencia de la otra entidad, mientras que un "0" representa que una ocurrencia de una entidad puede **no estar asociada** con cada ocurrencia de la otra entidad.

La combinación de estos símbolos describe tres tipos de relaciones: **obligatorias**, **opcionales** y de **contingencia**, que se interpretan como:

Obligatorias: toda ocurrencia de una entidad debe participar en la relación con una ocurrencia de la otra entidad. La figura 4 muestra un ejemplo:

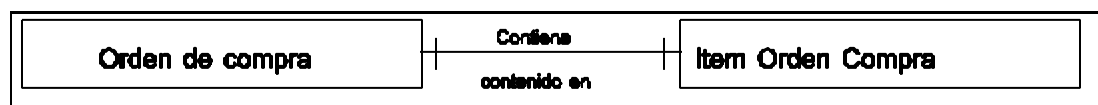


Figura 4: Relación Obligatoria

Opcionales: no es necesario que una ocurrencia de una entidad este asociada con una ocurrencia de la otra entidad. La figura 5 muestra un ejemplo:

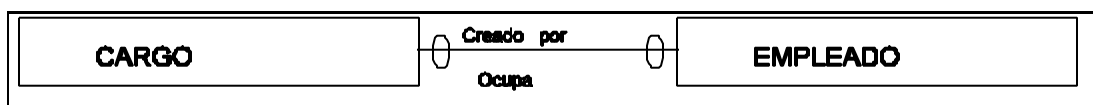


Figura 5: Relación Opcional

De Contingencia: una ocurrencia de una de las entidades puede existir sólo si está relacionada con una ocurrencia de la otra entidad, pero una ocurrencia de la otra entidad puede existir independientemente y no necesita participar en la relación. La figura 6 muestra un ejemplo:

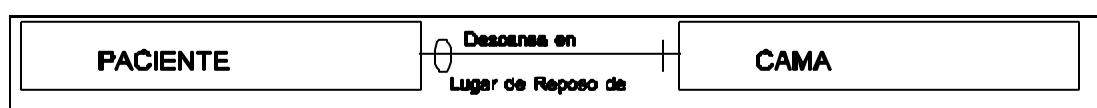


Figura 6: Relación de Contingencia

Otra característica de las relaciones es la **cardinalidad**, que corresponde a una medida del límite máximo de la relación. Las medidas de cardinalidad posibles son:

- **1:1 (uno a uno):** una ocurrencia de una entidad está relacionada con sólo una ocurrencia de la otra entidad, y vice versa.

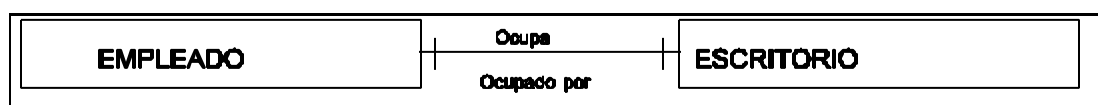


Figura 7: Relación de Uno a Uno

- **1:M (uno a muchos):** una ocurrencia de la primera entidad está relacionada con muchas ocurrencias de la otra entidad, mientras que una ocurrencia de la otra entidad está relacionada con sólo una ocurrencia de la primera entidad.



Figura 8: Relación de Uno a Muchos

- M:N (muchos a muchos): una ocurrencia de la primera entidad está relacionada con muchas ocurrencias de la otra entidad, y viceversa.

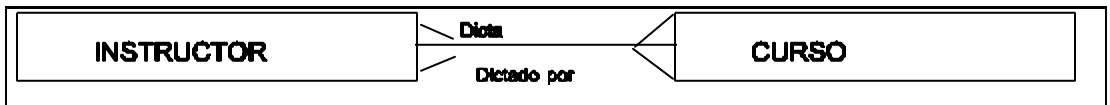


Figura 9: Relación de Muchos a Muchos

La cardinalidad puede combinarse con los atributos de existencia, entregando información más detallada en el diagrama. Las convenciones de notación e interpretación que se definen para estas combinaciones quedan del siguiente modo, tal como aparece en la tabla 1:

Uno de A está asociado con cero uno de B	
Uno de A está asociado con uno y sólo uno de B	
Uno de A está asociado con cero, uno, muchos de B	
Uno de A está asociado con uno muchos de B	

TABLA 1: Convenciones de notación para Cardinalidad de Relaciones.

Existe un tipo especial de relaciones, las relaciones recursivas, que indican asociaciones entre las ocurrencias de la misma entidad. La figura 10 muestra un ejemplo.

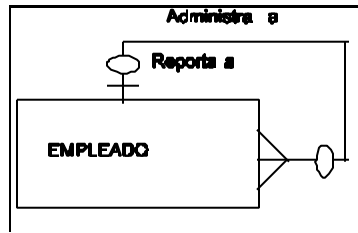


Figura 10: Relación Recursiva

Puede existir más de un tipo de relación entre dos entidades, las que deben distinguirse con nombres diferentes. La figura 11 muestra un ejemplo.

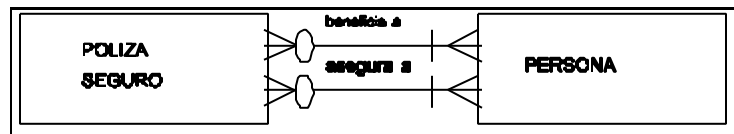


Figura 11 : Relaciones múltiples entre entidades

d. Identificación de Atributos

Un atributo es un hecho o propiedad significativo y único acerca de una entidad, y tiene sentido sólo dentro de ésta. La figura 12 muestra esta situación.

Entidad	Atributos
Empleado	Nº, nombre, fecha de nacimiento, sueldo, fecha ingreso, fecha salida...
Cliente	Código, nombre, nivel crédito, límite crédito

Figura 12: Entidades y Atributos

Los atributos toman valor para cada ocurrencia de la entidad. Las ocurrencias de una entidad son descritas por los valores de sus atributos, tal como lo muestra la figura 13.

¡Error! Marcador no definido.Ocurrencia Entidad	Valores Atributos
José Pérez	2012, José Pérez, 0/04/61, \$250.000. 01/07/91, vacío.
Marambio, Consultores	1348A, Marambio Consultores, Normal US\$10.000

Figura 13: Ocurrencias de entidades y valores de atributos

El rango de valores posibles que puede tomar un atributo se denomina **dominio del atributo**.

La identificación de atributos sigue un esquema como el que se presenta a continuación:

- Considere cada entidad y establezca los hechos requeridos para sus ocurrencias; cada uno de estos hechos puede ser un atributo.
- Considere los requerimientos de procesamiento de los procesos de su modelo de procesos. Las transformaciones normalmente serán a nivel de atributos de entidades.
- Considere los contenidos de los flujos de datos.
- Considere los sistemas existentes, automatizados o no. Examine los datos registrados para ellos, pues cada uno puede ser un potencial atributo.
- Considere la definición de las entidades y su objetivo.
- Examine toda documentación relevante relacionada con el área de estudio. Incluya la revisión de manuales, formularios, especificación de sistemas computacionales relacionados, especificación de archivos, pantallas, listados, etc.

Cada una de las fuentes anteriores indicará los hechos que la organización requiere para ser registrados, siendo ellos potenciales atributos de entidades. Si no existe una entidad que pueda abarcar al atributo descubierto, cree una nueva, según las especificaciones anteriores.

La descripción de atributos será en extremo relevante para la especificación final de los datos y su modelamiento físico. Las siguientes propiedades serán exigidas en el modelo de datos:

Nombre: nacen fundamentalmente de un acuerdo con usuarios y/o miembros del equipo de trabajo. El nombre completo del atributo debe consistir del nombre del atributo seguido del nombre de la entidad, por ejemplo: **nombre.CLIENTE, ubicación.EMPLEADO.**

Definición: Debe indicar el significado y propósito del atributo, dejando absolutamente claro qué propiedad de la entidad describe cada atributo. Por ejemplo, **precio.PRODUCTO:** "Precio inicial actual ofrecido a un cliente estándar". Algunos casos pueden ser obvios, por lo que no siempre deberá ir una definición del atributo, como por ejemplo **nombre.CLIENTE.**

Sinónimos: cuando aparezcan sinónimos para algún atributo, deben ser registrados.

Cardinalidad: todo atributo tendrá una cardinalidad mínima y máxima, reflejando si es dato simple, o repetitivo, obligatorio u opcional. Se describirán de acuerdo a las convenciones siguientes:

- **<0,1> dato simple, opcional.** Indica cardinalidad mínima de cero (puede no tener un valor en alguna ocurrencia), y un valor máximo de 1 (si tiene valor, el máximo será uno). Por ejemplo, fecha.RESERVA: Tendrá valor sólo cuando exista una reserva, y en ese caso tendrá sólo un valor.

- **<0,m> dato repetitivo, opcional.** Indica cardinalidad mínima de cero (puede no tener un valor en alguna ocurrencia), y un valor máximo de m (si tiene valor, podrá haber muchos valores dentro de la misma ocurrencia de la entidad). Por ejemplo, crédito. CLIENTE: El cliente puede tener más de un crédito asignado, pero podría no tenerlo.

- **<1,1> dato simple, obligatorio.** Este caso corresponderá a aquellos atributos que son candidatos a llave. Corresponden a datos únicos. Por ejemplo, número.CUENTA CORRIENTE: Una cuenta corriente tendrá siempre un número que la identifique, y a lo más un número.

- **<1,m> dato repetitivo, obligatorio.** Similar al anterior, pero dentro de la misma ocurrencia puede tener varios valores asociados. Habrá al menos un valor para el atributo. Por ejemplo, teléfono.PROVEEDOR: El proveedor puede tener más de un teléfono registrado, pero al menos se le exigirá uno.

Los casos **<x,m>** corresponden a entidades que serán tratadas con la Primera Forma Normal, que justamente elimina los atributos repetitivos de las entidades, creando otras entidades como resultado.

Condiciones de Exigencia: Cuando los atributos sean declarados como opcionales, deben establecerse las reglas y restricciones, referidas a otros atributos y/o relaciones que determinan su existencia. Los atributos obligatorios, por su parte, deben tener asociado un valor por defecto, que regirá como valor del atributo hasta que sea ingresado el valor final.

Valores Permitidos y su significado: Cuando un atributo tenga un dominio distinto de un string de caracteres, una cantidad o una fecha, será necesario determinar el rango de valores permitidos y el significado de cada valor, si es necesario. Por ejemplo, tipo CLIENTE, puede tener los valores permitidos "vigente", "potencial", "convenio" y

otros. Estos valores no pueden ser conocidos a menos que quien define el atributo los declare. Todos los atributos que indican algún tipo de "código" o "tipo" deben ser establecidos.

El origen de un atributo puede traer ciertas complicaciones a la hora de determinar su inclusión en una entidad. En este sentido, será una regla el que todos los atributos de entrada, es decir aquellos que no pueden ser derivados o calculados de otros atributos, no serán incluidos en las entidades. Los **atributos derivables**, es decir aquellos que resultan de combinar otros atributos, **no serán incluidos ni en el modelo conceptual ni en el modelo canónico**. Será una decisión de diseño el incluir atributos de este tipo para propósitos como mejorar respuestas, agilizar navegaciones, etc. En todo caso, si llegase a existir alguna duda respecto de las características de algún atributo, en cuanto a si es de entrada o derivable, será mejor incluirlo.

Similar a lo anterior, cada atributo incluido en el modelo deberá corresponder a un hecho particular que se desea registrar acerca de la entidad. Los códigos compuestos o "inteligentes", en donde cada dígito que compone tal código en las distintas ocurrencias describe información distinta, no deben ser incluidos, y se descompondrá en los datos atómicos que aporta cada componente, convirtiéndose cada uno de ellos en un atributo por sí mismo. Será una decisión posterior el agruparlos para componer una clave o código especial.

e. Descripción de Entidades

Es una tarea que se realiza continuamente. No es prudente describir una entidad apenas ésta es descubierta. El modelo de datos debe estar estable antes de hacerlo. Sin embargo, la definición de una entidad debe delinearse apenas se descubre la entidad. Las propiedades que describirán a una entidad, y que son exigidas en los documentos de especificación, son las siguientes:

- **Nombre:** según las características ya discutidas antes.
- **Definición:** establece el significado de la entidad, describiendo su rol y propósito para la organización. Debe dejar absolutamente claro el alcance de la entidad, en términos de acotar el tipo de ocurrencias que incorpora.
- **Atributos:** lista de los atributos de la entidad.
- **Identificadores:** atributos atómicos o combinación de atributos y relaciones que permiten la identificación de las distintas ocurrencias (ver más adelante).

f. Descripción de Relaciones

De manera similar a las entidades, la descripción final debe ser hecha cuando el modelo se vea estabilizado, por lo tanto se sigue una descripción evolutiva de las mismas.

La descripción de relaciones especifica las reglas que gobiernan cuándo las instancias individuales de cada entidad en la relación están de hecho asociadas con una instancia.

Se requieren las siguientes propiedades para describir una relación:

Nombre: Según las normas establecidas antes.

Definición: Establece el significado de la relación, en términos de su rol y propósito.

Reglas de Existencia: establece las condiciones bajo las cuales la relación es creada y borrada.

g . Identificador de Entidades

- Debe haber al menos un atributo (o combinación de éstos y/o relaciones) cuyos valores identifiquen en forma única cada entidad. Este será el identificador de la entidad.
- No es necesario denotar al identificador inmediatamente descubierta la entidad, pero si es indispensable que la entidad tenga, al final, un identificador que distinga a las ocurrencias de la entidad.
- Una entidad podrá tener más de un identificador, pero no es importante distinguir uno de ellos como "primario" en el modelamiento conceptual. Incluso puede ser postergado hasta la fase de Diseño.
- Si una entidad usa una relación como parte de un identificador, la entidad deberá tener participación obligatoria en la relación, y debe tener una cardinalidad máxima de uno.

2.4 MODELO DE DATOS CONCEPTUAL CANÓNICO

Es la versión del modelo conceptual:

Canonizado, es decir, hay redundancia de entidades, relaciones, y atributos de ambas.

Normalizado, es decir, los datos están en su forma más simple e independiente.

2.4.1 CANONIZACIÓN DEL MODELO CONCEPTUAL

Canonización significa eliminar cualquier tipo de redundancia.

Entidades redundantes: se producen cuando los atributos son los mismos en diferentes entidades. Se puede evitar esta redundancia si es posible dejar clara la diferencia por la vía de las relaciones en que participan.

Ejemplo:



Figura 14: redundancia de estructura

Lo que se puede hacer es dejar una sola entidad, la que servirá para ambos propósitos. Se podrá reconocer su naturaleza por:

- Un atributo, que es lo menos aconsejable (e.g. TIPO)
- Mediante su participación en las relaciones, que es lo más ventajoso

Atributos redundantes, se produce cuando un atributo puede ser obtenido a partir de otros. Ejemplo: Total costo horas/ hombre en la entidad EMPLEADO.

Relaciones redundantes: No sólo los atributos pueden ser redundantes, también las relaciones. Este tipo de relaciones es preferible omitirlas del diagrama Entidad - Relación. Ocurren cuando hay trayectorias cerradas, pues entre cualquier par de entidades hay dos caminos alternativos de navegación. Pero, ¿ellas dicen lo mismo? ¿Entregan la misma información? La figura 15 muestra un ejemplo de este tipo de relaciones:

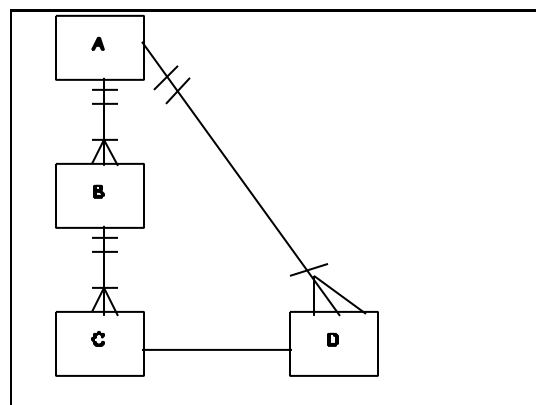


Figura 15: Relación Redundante

Una relación es redundante si es posible derivarla bajo cualquier condición todo el tiempo, es decir, si todas las asociaciones entre las concurrencias de las entidades fueran idénticas con o sin la posibilidad de la relación redundante.

No hay relaciones redundantes si en el ciclo cerrado sucede que:

- Las relaciones son de distinta naturaleza, por lo tanto, no hay transitividad.
- Hay dos o más relaciones M:N (normalizadas o no)

Si hay una sola relación M:N y todas las relaciones directas, será redundante pues dice lo mismo que la relación indirecta que une las dos entidades. Es decir:

Relación Directa: A-D

Relación Indirecta: A-B-C-D

Se podría eliminar la relación A-D ya que es redundante, sin embargo se debe considerar el largo de la trayectoria que significa recorrer las entidades versus el número de requerimientos asociados a esta trayectoria. La ventaja de la redundancia es que da más claridad al usuario, ya que representa con mayor exactitud la situación que se quiere modelar. Las desventajas de la redundancia son un mayor uso de procesador para actualizar varias ocurrencias y mayor espacio de almacenamiento.

2.4.2 NORMALIZACIÓN DEL MODELO CONCEPTUAL

Su propósito es analizar los atributos de las entidades para confirmar que ellos están ubicados en la entidad correcta y que no son necesarias entidades adicionales, proveyendo una base para el diseño de la base de datos lógica que minimice los accesos a la base de datos.

Para que un modelo de datos no tenga redundancia, y para prevenir anomalías de actualización, debe ser normalizado, esto es, que cada atributo de una entidad sea completamente dependiente de su identificador primario.

Es el proceso de búsqueda de la forma en la cual los datos pueden ser trabajados independientemente, pero manteniendo sus relaciones. Permite optimizar el modelo conceptual. En otras palabras, es una representación de los datos en una forma más clara, sencilla, visual, y fácil de implementar. Una representación que contiene nada más que la información necesaria.

Enfoque:

a) Obtener la lista de atributos de cada entidad

Obtenga una lista para la entidad, que incluya todos los atributos individuales (no compuestos) y las relaciones. Verifique que se dan las siguientes reglas.

- Cada entrada en la lista tiene un nombre único. Si aparece el mismo nombre dos veces, por ejemplo "fecha", agregue, un calificativo que los distinga, como "inicio" y "fin".
- El significado oculto de códigos estructurados debe abrirse en atributos separados, y los atributos originales deben ser excluidos. Por ejemplo, el ISBN, Internacional Standard Book Number, identifica al editor, el sujeto de la materia, la secuencia de publicación y un dígito verificador. Todos estos deben separarse en atributos distintos, excepto el último.

Un ejemplo de una lista correcta es el siguiente:

EMPLEADO (número empleado, nombre empleado, número departamento, nombre departamento, cargo empleado, sexo empleado, código carga empleado, nombre carga empleado, edad carga empleado).

Con este ejemplo se ilustrarán los pasos de normalización.

b) Seleccionar el Identificador Primario:

Si no hay identificador definido, defina uno en este momento. Si existe más de un identificador, seleccione uno que sea el identificador primario, usando alguno de estos criterios.

- Seleccione el identificador más simple, es decir, aquel que involucre el menor número de atributos.
- Seleccione aquel cuyo significado sea asociado más intuitivamente con la entidad.
- Seleccione aquel cuya tasa de uso se perciba mayor que los otros.

Si el identificador seleccionado incluye una o más relaciones, incluya también en la lista el identificador primario de las entidades relacionadas relevantes. Si este identificador, a su vez, incluye relaciones, nuevamente incluya el identificador de las entidades relacionadas. Repita este proceso hasta que todas las relaciones hayan sido reemplazadas por identificadores primarios foráneos. Por supuesto, esto puede hacer que el identificador elegido sea mucho más complejo que lo que aparentaba originalmente, necesitándose un reemplazo de la elección original. Los atributos que forman el identificador deben ponerse en el principio de esta lista, subrayados, como en el ejemplo:

EMPLEADO (número empleado, nombre empleado, número departamento, nombre departamento, cargo empleado, sexo empleado, código carga empleado, nombre carga empleado, edad carga empleado).

c) PRIMERA FORMA NORMAL (1FN) Remove Grupos Repetitivos

Poner el modelo en 1FN significa sacar de las entidades los atributos repetitivos (o grupos repetitivos) como entidades separadas. Por cada atributo que no es identificador, pregunte: **¿Hay más de un valor para este atributo en cualquier ocurrencia de la entidad ?** Identifique todos los atributos en la lista para los cuales la respuesta es "sí". Busque el conjunto de tales atributos que siempre tienen el mismo número de valores dentro de cualquier ocurrencia individual de la entidad. En el ejemplo: código carga empleado, nombre carga empleado, edad carga empleado siempre ocurrirán una vez por cada carga que tenga **EMPLEADO**.

Por cada conjunto de estos atributos, cree una nueva lista de atributos, y copie en esta lista el identificador primario. Para cada lista de este tipo que surja de este trabajo, seleccione su identificador primario. Este identificador primario incluirá al menos uno de los atributos de la lista, además del identificador primario de la entidad original (entidad "padre"). Repita este proceso hasta que no queden grupos de atributos respectivos en la entidad padre. Las listas resultantes estarán en Primera Forma Normal (1FN). En el ejemplo:

EMPLEADO (número empleado, nombre empleado, número departamento, nombre departamento, cargo empleado, sexo empleado)

CARGA EMPLEADO (número empleado, código carga empleado, nombre carga empleado, edad carga empleado).

d) SEGUNDA FORMA NORMAL (2FN) - Remove las dependencias parciales

Es necesario normalizar en 2FN cuando se tiene una relación de M:N, la cual debe transformarse en dos relaciones de 1:N. Considere la lista de cada entidad que contiene identificadores primarios compuestos, y para cada atributo que no es identificador en la lista, pregunte:

¿ El valor de este atributo depende completamente de los identificadores parciales?

Si la respuesta es "no", saque el atributo de lista y determine los identificadores parciales de los cuales éste depende. Haga una lista separada para todos los atributos que dependen del (los) mismo (s) identificador (res) parcial (es). Cada lista que se

crea forma la base de una nueva entidad, en la que debe copiarse el identificador parcial relevante de la entidad original.

De un nombre a cada nueva entidad creada, documéntelas en el estándar definido, y selecciones un identificador primario. Elimine los calificativos heredados de la entidad original. Por ejemplo, "empleado" ya no forma parte del atributo "código carga" en el ejemplo que sigue. Las entidades resultantes estarán en segunda forma normal.

EMPLEADO (número empleado, nombre empleado, número departamento, nombre departamento, cargo empleado, sexo empleado).

CARGA EMPLEADO (número empleado, código carga).

CARGA (código carga, nombre carga, edad carga).

Las entidades que se agregan para romper la relación M:N, como en el caso de CARGA EMPLEADO, reciben el nombre de "entidades asociativas" o "NUB".

e) **TERCERA FORMA NORMAL - Eliminar las Dependencias Transitivas**

Para tener el modelo en 3FN deben resolverse las transitividades en los atributos (datos) repetidos, sacándolos como entidades independientes.

Para todos los atributos que no son identificador, pregunte:

¿ Los valores de este atributo dependen del valor de otro atributo cualquiera distinto del identificador primario?

Si la respuesta es "sí", el atributo dependiente debe ser trasladado a otra entidad, con el atributo determinante como identificador primario. Separe los atributos afectados y llévelos a una nueva entidad, la que deberá tener otro nombre, una por cada grupo de atributos con el mismo nuevo identificador primario. Las entidades resultantes estarán en tercera forma normal. En el ejemplo, quedaría:

EMPLEADO (número empleado, nombre empleado, número departamento, cargo empleado, sexo empleado).

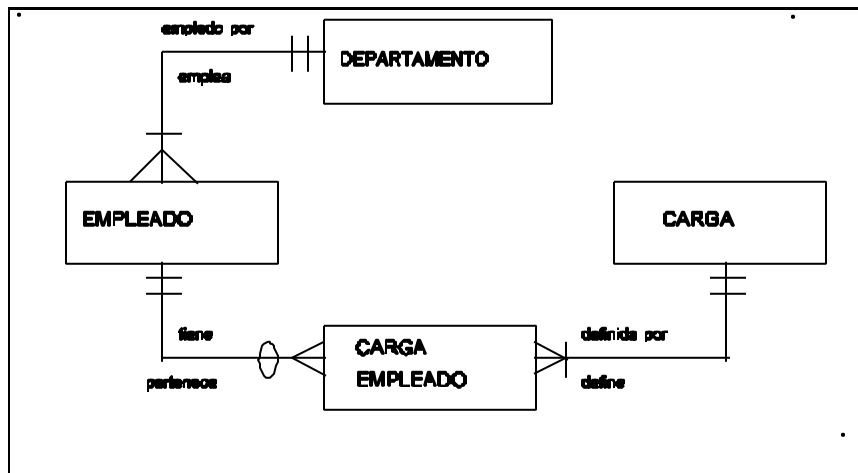
CARGA (código de carga, nombre carga, edad carga).

DEPARTAMENTO (número departamento, nombre departamento).

Sólo aquellos atributos que serán importantes (objeto de consultas) y aquellos más importantes por el usuario deben sacarse.

- Esto debería hacerse sólo si los requerimientos lo solicitan. Por ejemplo, si se quiere saber todos los empleados que pertenecen a un determinado departamento.

- La diferencia con respecto a la IFN es que en este caso (3FN) los datos repetidos salen como "padre", mientras que en la LFN los datos repetitivos salen como "hijos" de la entidad original.



Gráficamente, el ejemplo tratado quedaría de la siguiente manera:

Figura 16: Sub modelo normalizado en 3 FN

Anexo 5: Diagrama de Flujo de Datos (DFD)

La idea de analizar el flujo de datos es estudiar el empleo de datos en cada proceso al interior de una institución.

Este análisis permite:

- Tener una visión de lo general a lo particular
- Comprender las relaciones entre sistemas y subsistemas
- Especificar, en forma rigurosa, procesos o transformaciones que ocurren en cada sistema
- Mejor comunicación entre los desarrolladores y los usuarios
- Separación lógica y física del flujo de información

Para desarrollar un diagrama de flujo de datos hay que seguir los siguientes pasos:

1) Hacer lista de actividades para determinar:

- Entidades externas
- Flujos de datos
- Procesos
- Almacenes de datos

2) Diagrama de contexto

3) Diagrama de nivel 0

4) Crear diagrama hijo por cada proceso de nivel 0

5) Revisar el modelo

6) Desarrollar DFD físico a partir del DFD lógico

Representación gráfica de los procesos que componen el sistema y las interfaces entre ellos

Componentes

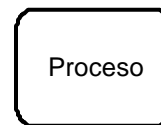
1) Entidad externa (rectángulo):

- Indica las fuentes o destinos de los datos

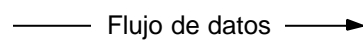


2) Proceso (rectángulo con puntas redondeadas):

- Transforman los datos
- Los flujos de entrada son diferentes a los de salida



3) Flujo de datos (flechas):



4) Almacén (rectángulo abierto)

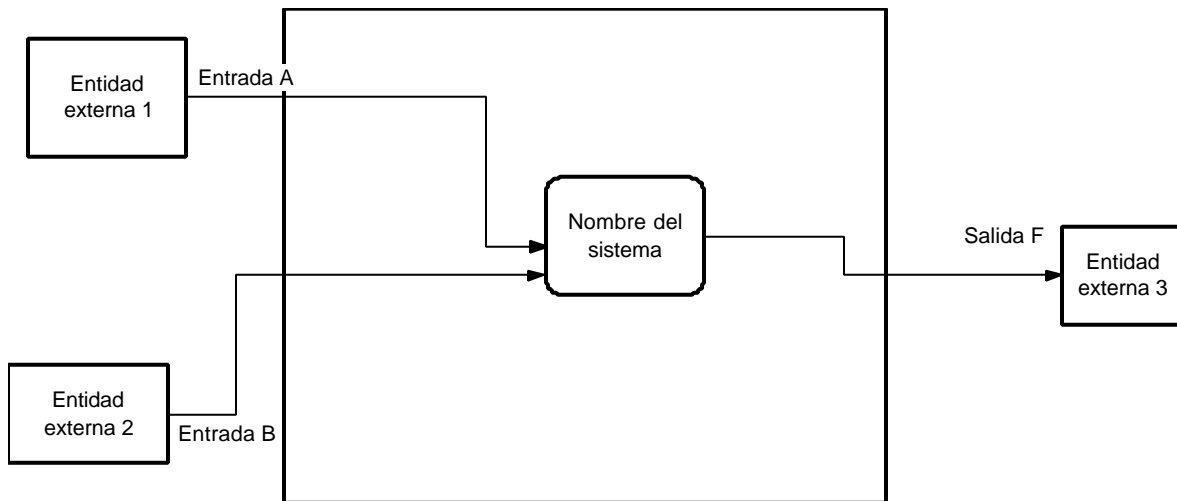
- El medio físico no es especificado; incluye medios manuales de almacenamiento
- No incluye almacenamientos temporales



Diferencia entre DFD lógico y físico

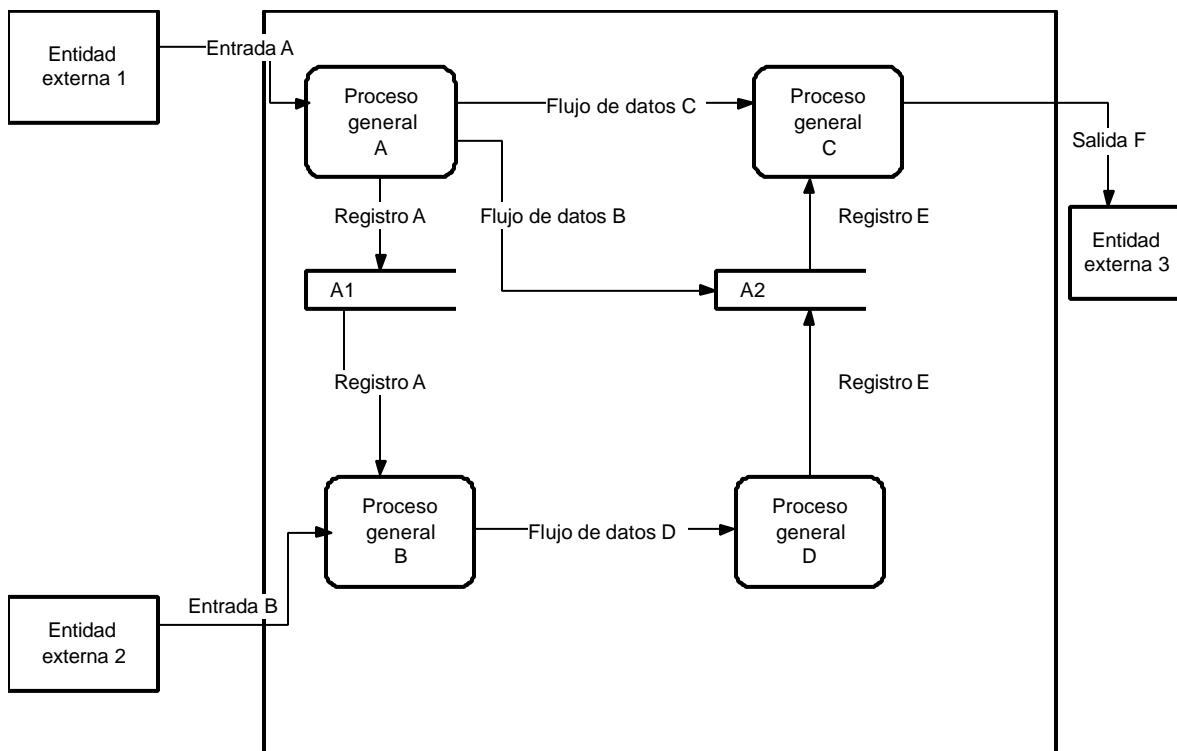
DFD lógico	DFD físico
Orientado al qué hacer	Apunta al cómo hacerlo
Enfocado a la actividad propia de la organización y como opera	Implementación del sistema. incluye hardware, software, archivos y personas involucradas
Describe los eventos de la actividad propia de la organización	Depende de la tecnología disponible
Funciones de la actividad propia son más estables que la tecnología	

Ejemplo general de diagrama DFD de contexto



- Este diagrama muestra todo el sistema como un proceso sencillo con entidades externas y sus principales entradas y salidas
- No muestra los almacenes de datos

Ejemplo genérico de diagrama DFD de nivel 0

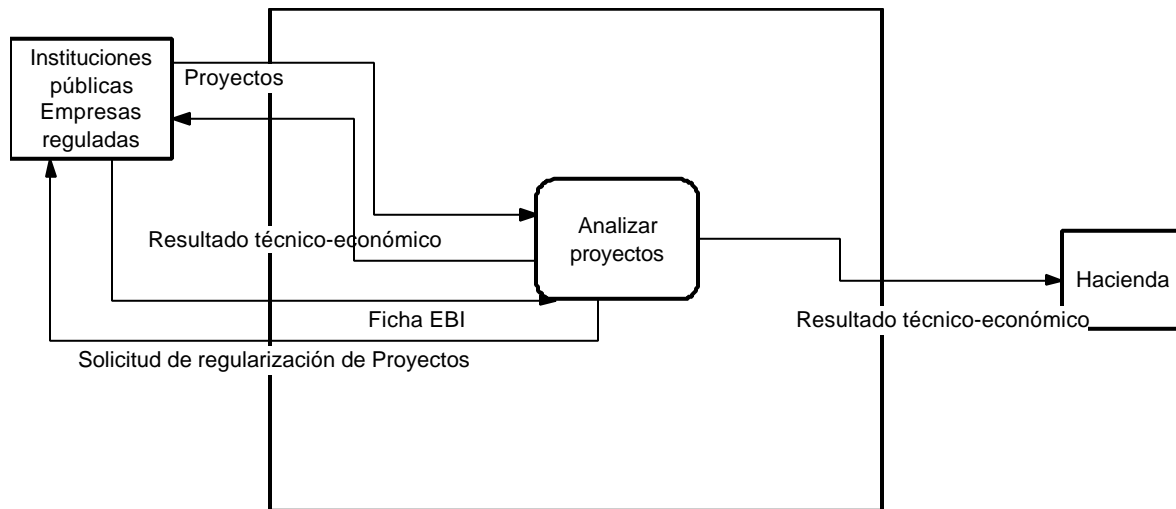


Este diagrama describe sólo los procesos más generales e incluye los almacenes de datos.

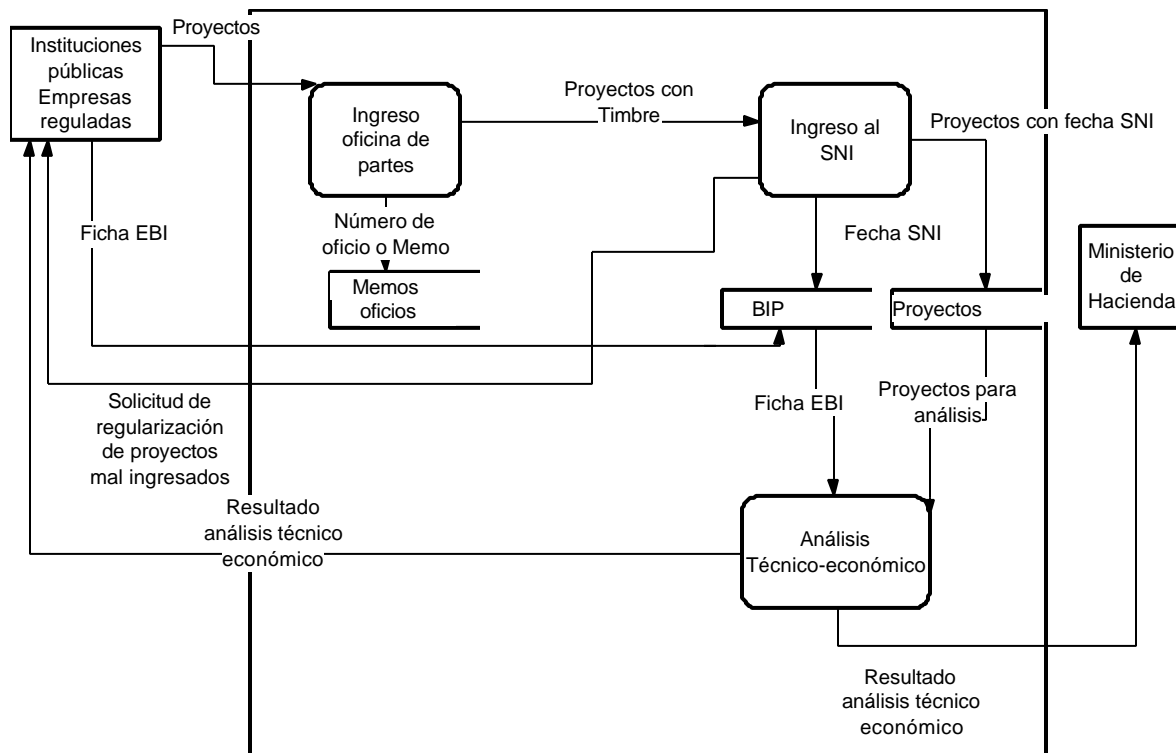
Posteriormente se crea diagrama hijo por cada proceso de nivel 0, usando la misma metodología, o simplemente se agrega al modelo procesos que no habían sido considerados anteriormente.

Como ejemplo se presenta a continuación, un esquema simple (no incluye todos los procesos) del análisis de proyectos en el SNI.

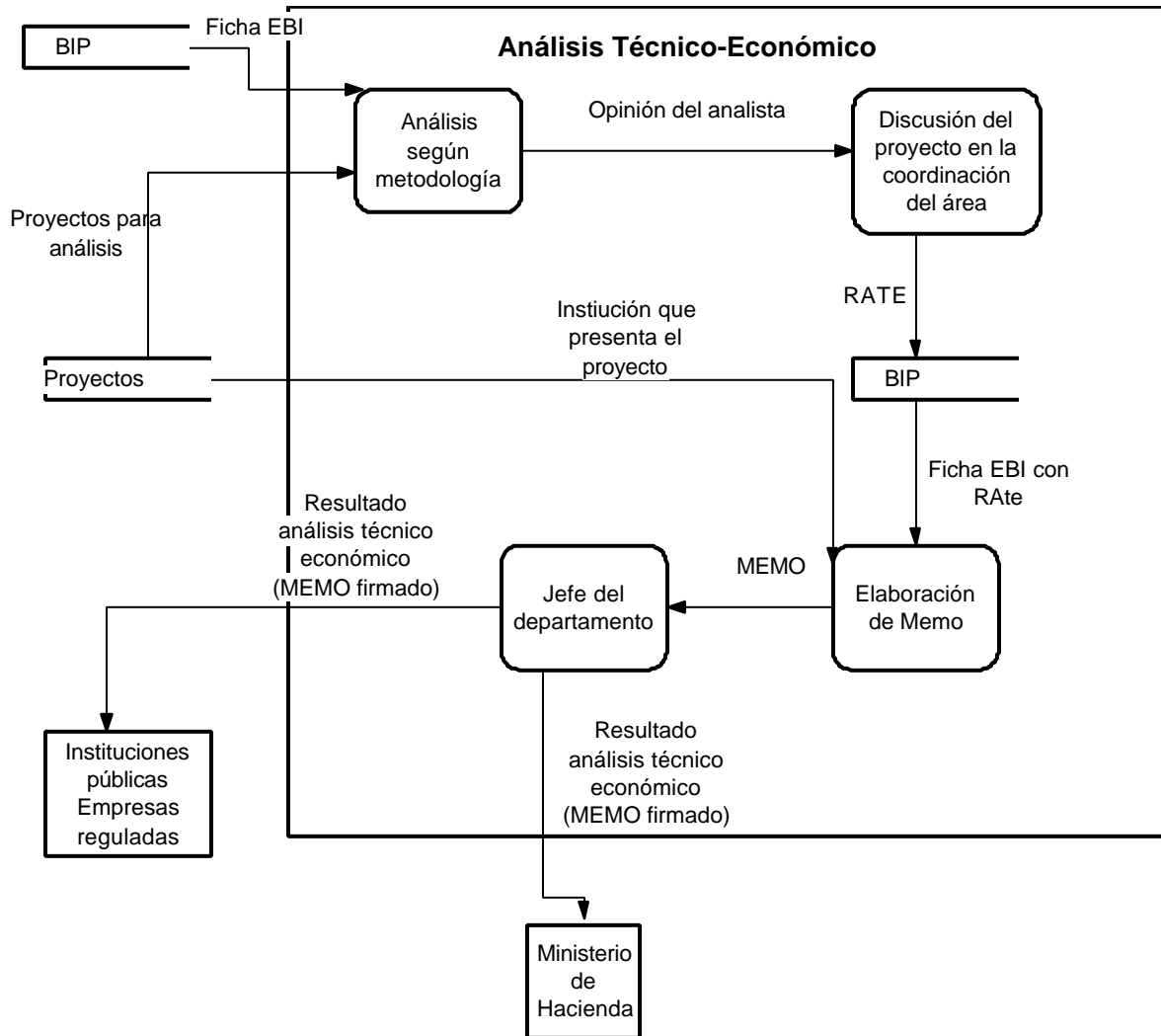
Ejemplo: Diagrama de Contexto de Análisis de proyectos en el SNI



Ejemplo: Diagrama Nivel 0 de Análisis de proyectos en el SNI



Ejemplo: Diagrama Hijo del proceso de Análisis Técnico - económico en el SNI



Anexo 6 Calidad Funcional

En este ítem se pretende evaluar las prestaciones que ofrecen las distintas alternativas de Diseño Físico, para implementar de la solución. Deberá considerarse la existencia de controles automatizados, redundancia de datos mínima y justificada, cobertura sobre los procesos, reglas para eliminación y modificación de datos y, en general, todas aquellas características que hagan que la solución sea eficiente y robusta.

En particular, deberán evaluarse los siguientes atributos:

Controles automatizados

Validaciones al ingreso y modificación de datos

Cuadraturas de datos

Reportes de inconsistencias

Consistencia de Datos

Ingreso único de datos en el origen

Registro de modificaciones históricas

Redundancia de datos mínima y justificada

Definición de reglas específicas para la mantención de datos históricos

Cobertura de procesos

Validación de consistencia del Modelo de Datos con respecto a los procesos diseñados

Cumplimiento de Reglas exógenas

Velar por el Cumplimiento de leyes, regulaciones, normas, acuerdos contractuales, etc.

De igual forma que en la evaluación de los factores anteriores, deberán calcularse ponderadores para cada grupo de atributos y completar una matriz como la siguiente, calificando cada alternativa con una nota de 0 a 100, en base a los criterios ya señalados.

Cuadro 4: Matriz de Evaluación Calidad Funcional

Aspectos de la calidad funcional	Ponderador	Altern. 1	Altern. 2	...	Altern. N
Controles Automatizados	x%	100	100		100
Consistencia de Datos	x%	100	100		100
Cobertura de Procesos	x%	100	100		100
Cumplimiento de reglas exógenas	x%	100	100		100
TOTAL	100%	CF1	CF2		CFN

Los ponderadores (x%), se determinan con la metodología del anexo1.

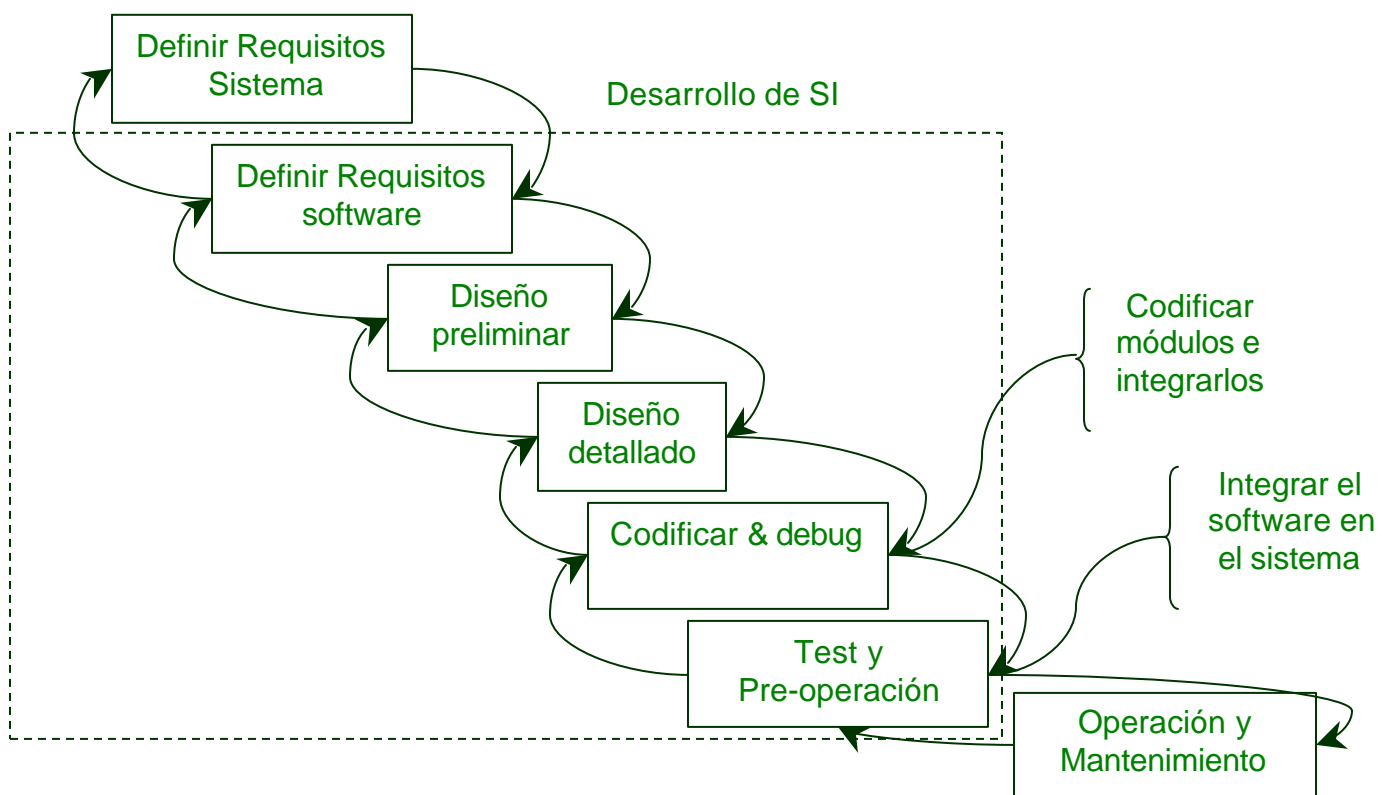
Anexo 7: Etapas de un Proyecto de Desarrollo en Informática.

En informática, la lista de etapas que se realizan **desde que** se decide desarrollar un proyecto de software **hasta que** se entrega dicho producto, dependen del ciclo de vida adoptado el cual puede ser :

- Sin prototipos
 - En cascada (Waterfall)
- Con prototipos
 - Desechable
 - Parte del sistema definitivo
 - Incremental
 - Evolutivo
- Ciclo de vida en espiral

Ciclo de Vida en Cascada

Es el más común y antiguo de los ciclos de vida. La idea es completar un paso antes de comenzar el siguiente.

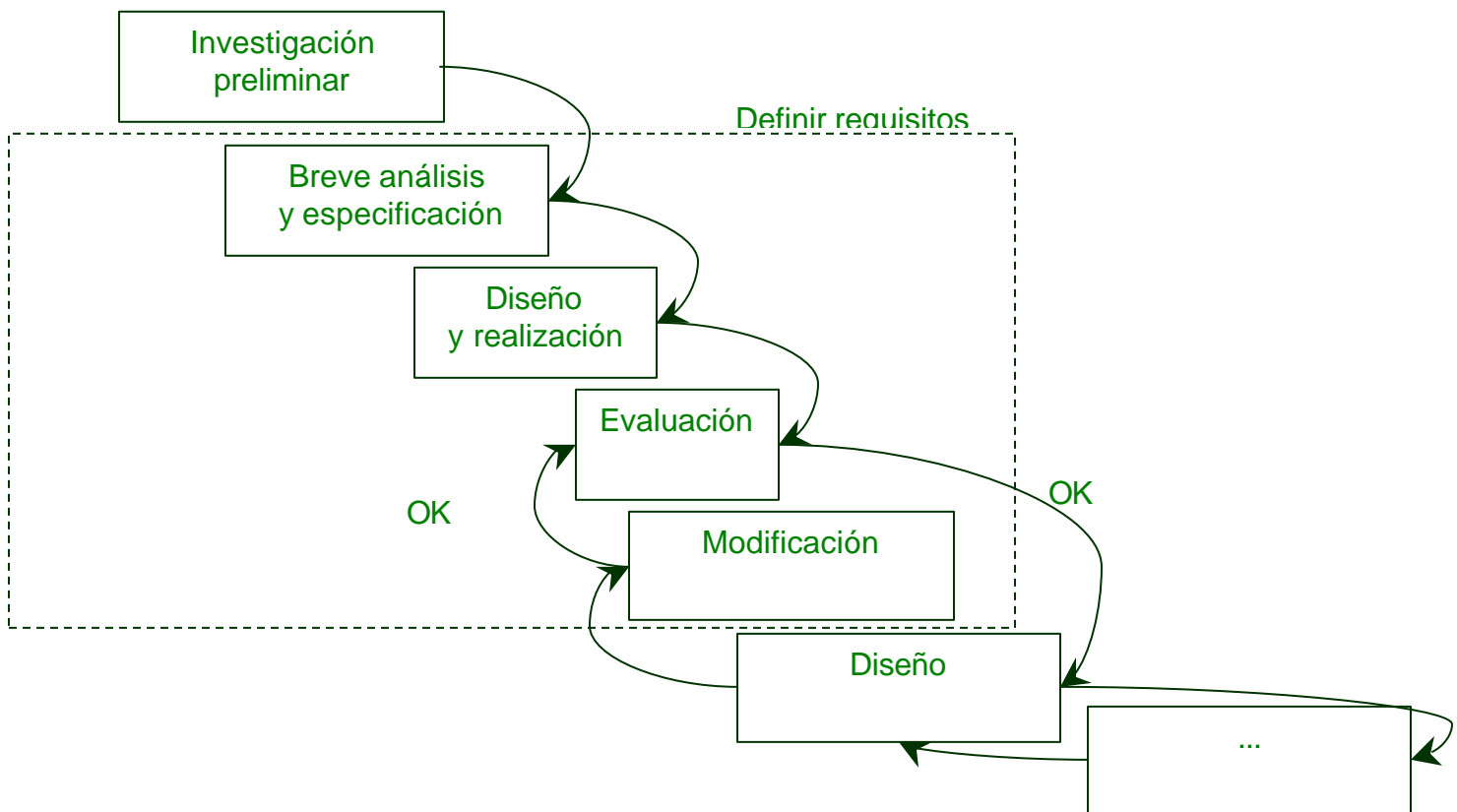


Problemas del Ciclo de Vida en Cascada

- Hay ocasiones en que no se puede pretender disponer e especificaciones correctas desde el primer momento (usuario poco preciso)
- Usuario (cliente) poco preciso o cambia de parecer en las necesidades
- Resultados al final. Los resultados no se ven hasta muy avanzado el proyecto y los cambios entonces pueden tener mucha incidencia en los plazos de entrega.

Ciclo de Vida con Prototipos

El prototipo es un modelo del sistema propuesto, que se construye para ilustrar la viabilidad del nuevo sistema, esto porque los prototipos constituyen un mejor medio de comunicación que los modelos en papel.

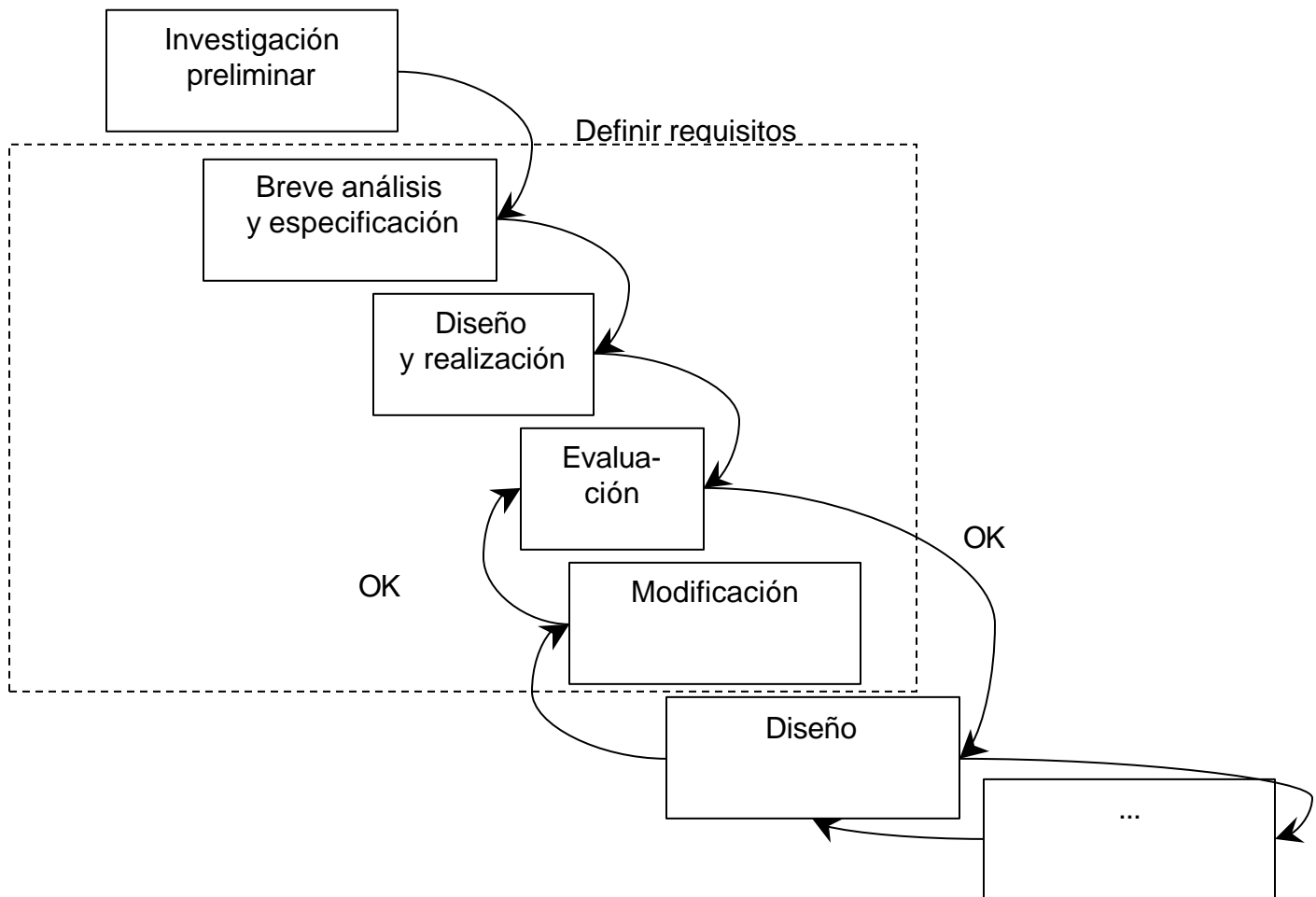


Características del Ciclo de Vida con prototipos

Alto grado de participación del usuario el cual evalúa los prototipos, propone mejoras y detalla requisitos.

Alto grado de participación del analista de sistemas, ya que en muchos casos los usuarios no pueden indicar los requisitos sin tener experiencia con el sistema. El prototipo da mayor conocimiento al usuario y analistas ayudando a que el usuario aprenda a utilizar el sistema. El prototipo se modifica hasta que los requisitos del usuario queden claros.

Ciclo de Vida en cascada con prototipos



Ciclo de vida con prototipos desechables

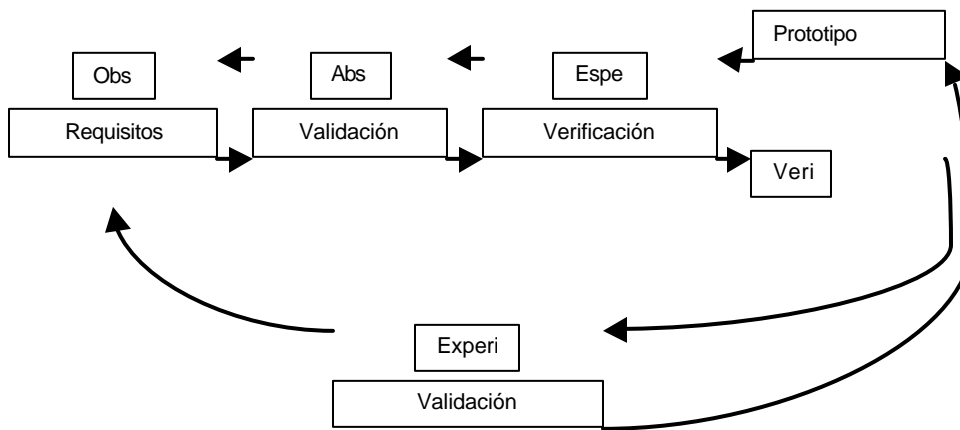
El prototipo no se utilizar para construir el sistema final. Se programan sin fijarse en usar buenas practicas de programación y se hacen muy rápido, usualmente se hacen en Perl, awk, csh, etc.). Lo importante es entender que en este caso no se debe tomar el prototipo como un producto final, y menos programar encima de este.

Ciclo de vida con prototipos no desechables

El prototipo es parte del sistema definitivo. Hay dos clases: Incremental el cual se divide a su vez en dos:

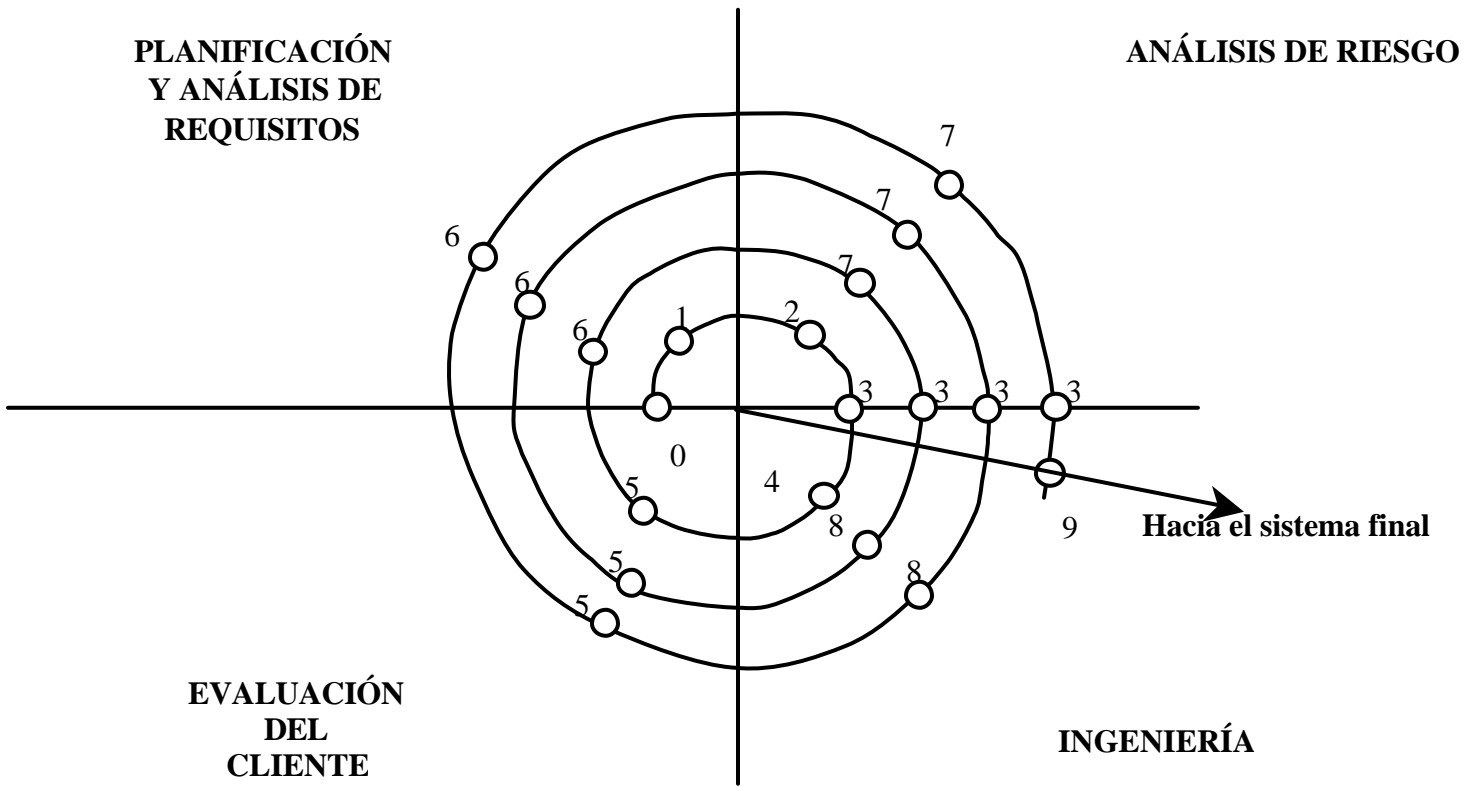
- Con requerimientos al principio
- Requerimientos+Diseño+Implmentación+Testing
- Requerimientos+Diseño+Implmentación+Testing
- Requerimientos antes de cada fase
- Requerimientos+Diseño+Implementación+Test
- Diseño+Implementación+Test
- Diseño+Implementación+Test

Evolutivo: se diferencia del incremental ya que en este caso los requerimientos no se pueden conocer totalmente al principio y se deben hacer poco a poco.



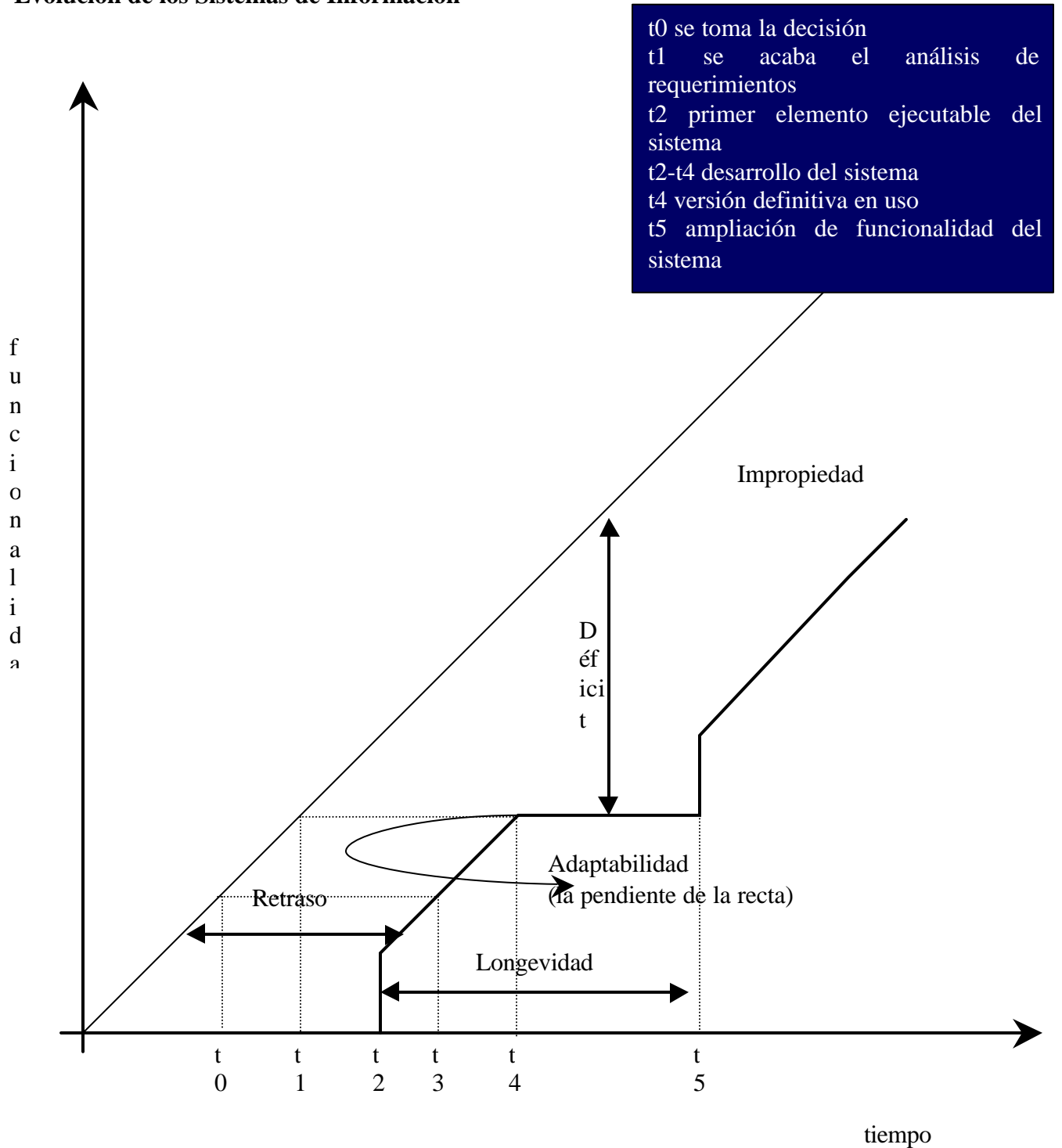
Ciclo de vida en espiral

El ciclo de vida en espiral, añade el análisis de riesgo en el proyecto de software y considera 4 fases: Planificación, análisis de riesgo, ingeniería y evaluación de proyectos:



- 0.- Decisión de desarrollar un determinado software
- 1.- Recolección de requisitos iniciales + planificación proyecto
- 2.- Análisis de riesgo según los requisitos iniciales
- 3.- ¿Continúa el proyecto?
- 4.- Prototipo inicial
- 5.- Evaluación del cliente
- 6.- Replanificación según comentarios del cliente
- 7.- Análisis del riesgo según la reacción del cliente
- 8.- Prototipo del siguiente nivel
- 9.- Sistema final

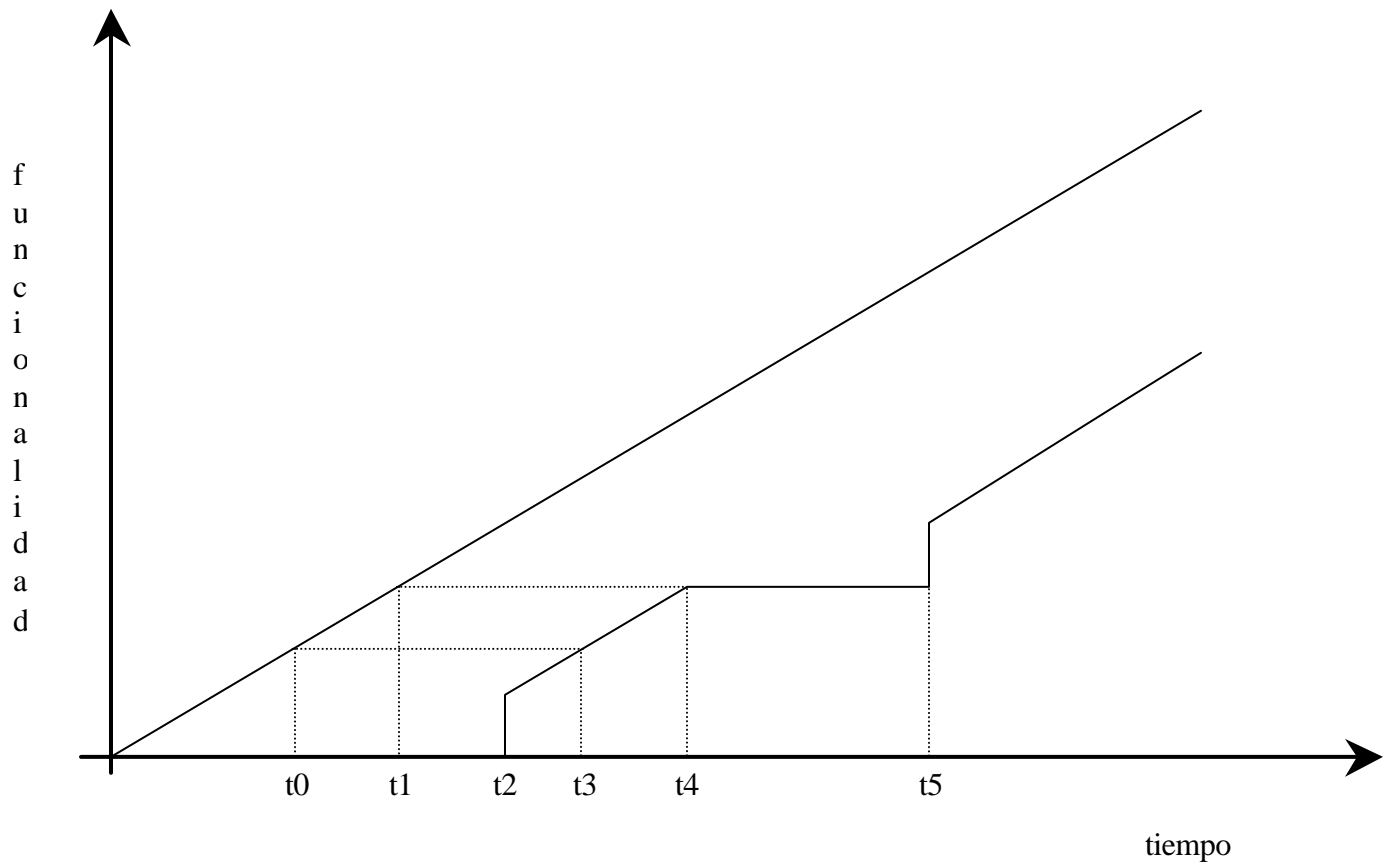
Evolución de los Sistemas de Información



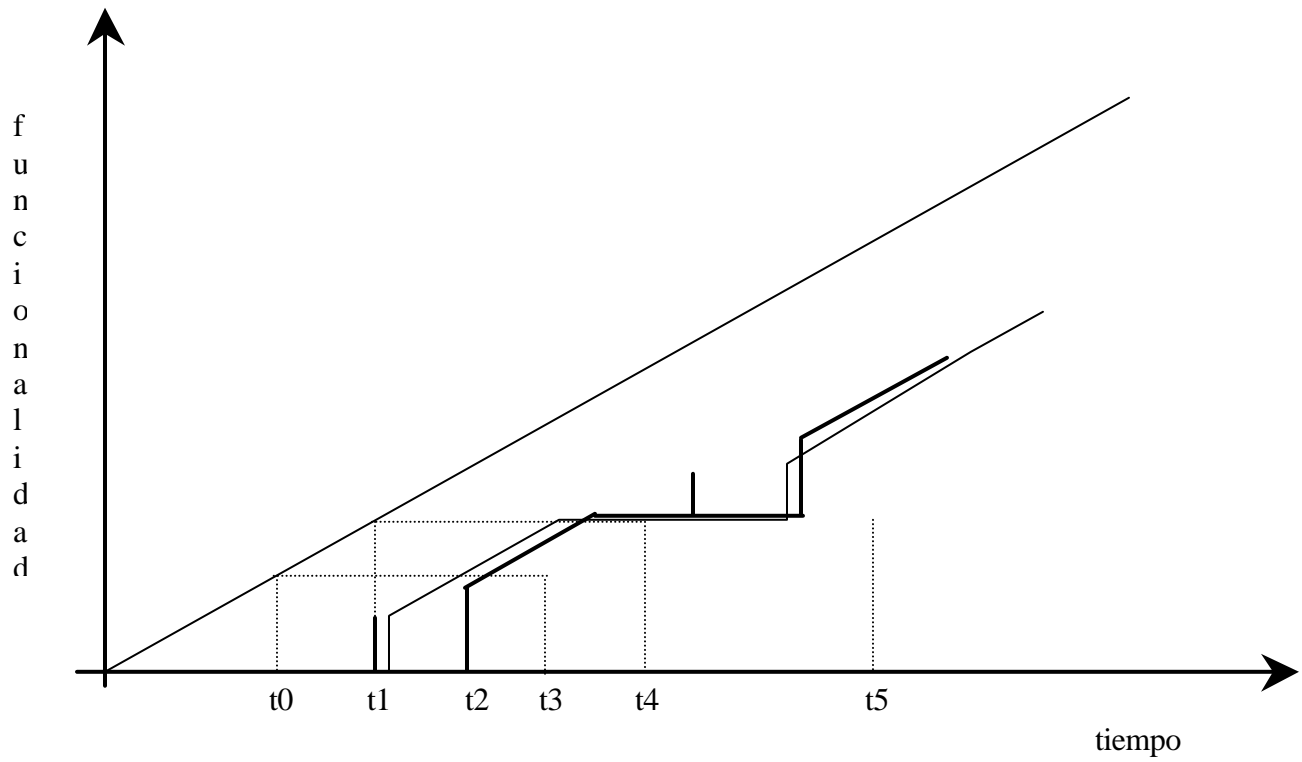
La línea más clara, representa el desarrollo ideal de un sistema de información, en el cual al tomarse la decisión de elaborar un SI, ya se sabría claramente parte de las funcionalidades necesarias del mismo, es así que al terminar el análisis de requerimientos en t_1 , se habría avanzado en la comprensión de las funcionalidades requeridas. En un proyecto real, lo cual

se aprecia en la línea negra, se aprecia un retraso debido a que cuando se obtiene el primer elemento ejecutable del sistema, todavía no se comprende en su totalidad las funcionalidades requeridas o simplemente no se han podido desarrollar en el tiempo deseado. Es así como existe un déficit entre lo que se lograría en condiciones ideales y lo que realmente se logra en cuanto a funcionalidad. La adaptabilidad, se refiere a la capacidad de agregar funcionalidades en el tiempo. La impropiedad se refiere a que al agregar funcionalidades a un sistema ya terminado, estas son más difíciles de obtener que en un sistema recién comenzado. No se logran todas las propiedades deseadas. A continuación se muestra la evolución de los SI, según los distintos ciclos de vida ya mencionados. Es importante notar, que dependiendo del ciclo de vida elegido el proyecto se acercará o alejará del ideal anteriormente expuesto.

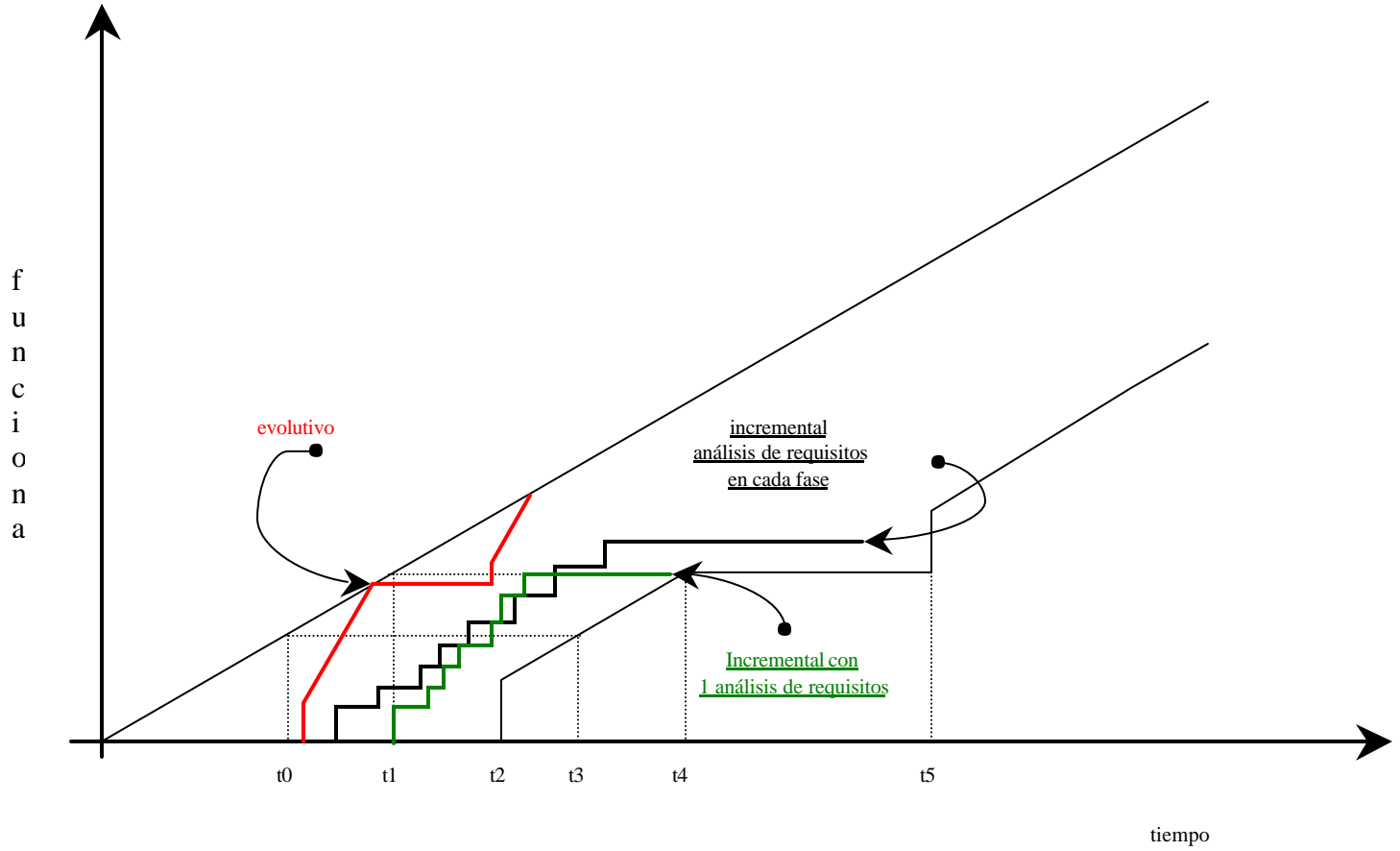
Evolución de SI en cascada



Evolución SI con prototipos desechables



Evolución SI con prototipos no desechables



Anexo 8: Ejemplos de medidas de efectividad.

Además de las señaladas en la Tabla 5 del punto 4.3 (Información en línea, Interfaces gráficas, Emisión de cartas, control de cambios), se pueden considerar las medidas, que a continuación se exponen para comparar alternativas, la idea en esta etapa es considerar la solución como un todo considerando hardware, software, comunicaciones:

- Capacidad de acceso y actualizaciones remotas
- Capacidad para disminuir los tiempos de procesos en la generación de información
- Capacidad de la solución para disminuir la duplicidad y errores en los datos
- Velocidad de procesamiento
- Capacidad de almacenamiento
- Capacidad de comunicación
- Ergonomía
- Compatibilidad con sistemas ya existentes en la organización
- Capacidades multimediales incorporadas
- Procesamiento distribuido
- Procesamiento centralizado
- Rendimiento esperado
- Facilidad de configurar
- Cumple con el número de transacciones requeridas
- Las funciones proveídas mejoran la eficiencia del usuario
- La solución actualiza sus archivos internos en línea
- La solución actualiza sus archivos internos en modo batch
- Necesidad de cálculos complejos
- Reusabilidad del código
- Facilidad de instalar
- Facilidad de operar
- La solución fue diseñada para ayudar al cambio organizacional

Este listado no pretende ser exhaustivo, y lo que es más importante, las medidas a considerar para medir la efectividad, dependerán de los objetivos del proyecto. Por ejemplo, si el proyecto no genera grandes demandas de almacenamiento, no tiene sentido considerar esta variable como deseable o muy deseable.

Anexo 9 Esfuerzo requerido para cada fase

El cálculo del esfuerzo requerido para cada fase es complicado, ya que depende de varias variables. A pesar de esto hay modelos que tratan de dar respuesta a esta dificultad, algunos de ellos son:

COCOMO: Constructive Cost Model que tiene como objetivo estimar el esfuerzo, costo y planificación del desarrollo de software.

Puntos de Función.: En este método la idea es obtener una medida cuantitativa del tamaño de las aplicaciones basándose en sus requisitos funcionales e ir midiendo la productividad del proyecto en la medida que este avanza.

Número de líneas de código: Otra manera bastante popular, es estimar la cantidad de líneas de código que generará el software a desarrollar. El problema con esto es que para una misma funcionalidad el número de líneas de código que se necesitarán dependerá del lenguaje utilizado. Para solucionar esto existen tablas que convierten líneas de código de diversos lenguajes a Assembler o a número de puntos de función.

Cómo estos modelos o metodologías para medir el esfuerzo requerido requieren cierta preparación e información histórica, que a menudo no es dominio de la institución es que facilitamos la siguiente información obtenida en conversaciones con diversos jefes de proyectos. Por lo mismo es solamente una referencia, ya que no responde a un estudio riguroso del problema, pero que consideramos un dato valioso y orientador, dada la gran variabilidad entre los tiempos y presupuestos estimados para un desarrollo y los que finalmente resultan. En opinión de las personas entrevistadas, el esfuerzo requerido en cada una de las fases del desarrollo de software es el siguiente:

Análisis: 15%

Diseño y programación: 75%

Testing: 10%

El Diseño aquí mencionado corresponde al Diseño lógico y físico.

Por otra parte hay cuadros elaborados en Estados Unidos en base a miles de proyectos de software que se pueden usar, pero que difieren entre sí debido a :

Diferencias en el significado de las fases

Métodos de desarrollo distintos

Productos distintos, en algunos casos se necesita más análisis que en otros

Habilidad de los programadores, lo cual impacta en el tiempo de desarrollo

Diferencia en los plazos requeridos

Ejemplos de cuadros:

Distribución de Recursos por Fase (%)				
Fase	Tamaño			
	Pequeño	Intermedio	Mediano	Grande
Diseño	16	16	16	16
Diseño programación	26	25	24	23
Programación y Testing	42	40	38	36
Integración	16	19	22	25

Distribución de Recursos por Fase (%)				
Fase	Tamaño			
	Pequeño	Intermedio	Mediano	Grande
Diseño	19	19	19	19
Programación	63	59	55	54
Integración	18	22	26	27

Anexo 10 Definiciones legales

Programa de computación: Conjunto de instrucciones para ser usadas directa o indirectamente en un computador a fin de efectuar u obtener un determinado proceso o resultado, contenidas en un cassette, diskette, cinta magnética u otro soporte material.

Copia de un programa de computación: Es un soporte material que contiene instrucciones tomadas directa o indirectamente de un programa computacional y que incorpora la totalidad o parte sustancial de las instrucciones fijadas en él.

Normas legales que protegen el programa: Ley N° 17.336 de 2 de Octubre de 1970, sus modificaciones y su reglamento (Decreto de Educación N° 1.122 del 17 de Mayo de 1971).

Modificaciones de la ley N° 17.336: Las siguientes leyes la modifican

N° 17.773 de 18 de Octubre 1972

N° 18.443 de 17 de Octubre 1985

N° 18.957 de 5 de Marzo 1990

N° 19.072 de 19 de Agosto 1991

N° 19.166 de 17 de Septiembre de 1992

La tenencia de los derechos sobre un programa corresponde a las siguientes personas:

El autor

Personas a las cuales el autor le haya transferido todos o algunos derechos, mediante contrato inscrito en el Registro de la Propiedad Intelectual

Los herederos o legatarios del autor

Personas naturales o jurídicas, cuyos dependientes, en el desempeño de sus funciones laborales, los hubiesen producido, salvo estipulación escrita en contrario.

Aquella que encarga la elaboración del programa para comercialización por su cuenta y riesgo, salvo estipulación de lo contrario.

Licencia de uso de un programa: Constituye la autorización para usar el programa en las condiciones descritas en ella. No hay transferencia de derechos de autor, salvo que se estipule explícitamente alguna cesión.

Duración de los derechos de autor sobre un programa, para:

- El autor: Toda la vida
- Conyuge sobreviviente: Toda la vida
- Hijas solteras, viudas o hijas casadas, del autor, cuyo cónyuge se encontrare afectado por una imposibilidad definitiva para todo género de trabajo: Toda la vida
- Otros herederos: 30 años post mortem del autor
- Cesionarios: 50 años post mortem del autor
- Legatarios: 50 años post mortem del autor

Adaptación de un programa: Es lícito cuando es efectuada por el tenedor del programa o autorizada por el legítimo dueño del mismo. Esta adaptación debe ser esencial para el uso del programa y no es destinada a un uso diverso. Estas adaptaciones no pueden ser transferidas bajo ningún título, sin que medie autorización previa del derecho de autor respectivo.

Copia de un programa: La copia es lícita cuando es efectuada por el tenedor del programa o autorizada por el legítimo dueño del mismo y es esencial para su uso en un computador determinado o para fines de archivo de respaldo. Estas copias no pueden ser transferidas bajo ningún título, salvo que lo sea con el programa computacional que le sirvió de matriz.

Inscripción de un programa en el registro de propiedad intelectual:

Se debe depositar un ejemplar del programa en el medio en que se encuentre incorporado. Además debe acompañarse con el manual de operación del programa (Manual de usuario, de instalación, otros)

Escrito explicativo que indique el lenguaje utilizado y la versión del programa si es pertinente, a menos que esta información ya esté contenida en el manual de operación.

Se debe informar al Registro los siguientes datos del autor:

Nombre completo o nombre de la persona jurídica

Domicilio

Cédula de identidad o RUT

Normas internacionales que protegen al autor de un programa: El autor nacional y extranjero domiciliado en el país están protegidos fuera del país en los países miembros de los siguientes convenios internacionales:

- Convención Universal sobre derecho de autor.
- Convenio de Berna
- Los autores extranjeros no domiciliados en el país, están protegidos en Chile por estas mismas convenciones y por la legislación nacional cuando proceda.

Anexo 11 Glosario

Arquitectura: En las tecnologías de la información (TI), especialmente en lo que refiere a computadores y más recientemente en lo que se refiere a redes, arquitectura es un término que se aplica al proceso y resultado de pensar y especificar la estructura, componentes lógicos, e interrelaciones lógicas de un computador, sistema operativo, red u otro concepto.

Bases de datos: Es una colección de datos, organizada de tal forma que sus contenidos pueden ser fácilmente obtenidos, gestionados y actualizados. El tipo de base de datos dominante actualmente es el modelo relacional (aunque en Chile aún hay un gran número de bases de datos que usa archivos indexados). En este tipo de bases de datos, los datos están definidos de tal manera, que es posible reorganizarlos y obtenerlos de diferentes maneras. Una base de datos distribuida es aquella que está dispersa o replicada en diferentes puntos de la red. Una base de datos orientada al objeto es aquella que es congruente con los datos definidos en clases de objetos y subclases.

Código Fuente: Consiste en declaraciones de programación que son creadas por un programador, mediante un editor de texto o una herramienta visual de programación y que posteriormente es grabada en un archivo con un determinado nombre. Después de este proceso el código fuente está listo para ser compilado. El resultado de esta compilación es el código objeto (esto no tiene que ver con orientación al objeto).

Código Objeto: Contiene una secuencia de instrucciones que el procesador puede entender, pero que es difícil de ser leído o modificado por seres humanos.

Cliente/Servidor: Describe la relación entre dos programas computacionales en el cual un programa, el cliente, pide un servicio a otro programa, el servidor, el cual satisface el requerimiento. La idea de cliente/servidor puede ser usada por programas localizados en un único computador, pero este concepto es más importante en redes. En una red, el modelo cliente/servidor provee una manera conveniente para interconectar programas que están distribuidos en diferentes lugares. Estas transacciones son muy comunes en las redes.

Datamart: Es un repositorio de datos obtenido desde datos operacionales y otras fuentes, que está diseñado para satisfacer requerimientos de una comunidad de trabajadores con ciertos conocimientos específicos. Es más pequeño que un Datawarehouse, de hecho el conjunto de Datamarts, pueden construir un Datawarehouse. A diferencia de los Datawarehouse, los Datamart se construyen a partir de los requerimientos del usuario final.

Datamining: Es el análisis de datos para encontrar relaciones que no habían sido descubiertas anteriormente. Puede revelar asociaciones por ejemplo entre productos, en este caso es conocida la relación encontrada entre pañales y cerveza. Esto porque muchos padres de familia al comprar pañales también compraban cervezas.

Datawarehouse: Es un repositorio central para toda la información importante que las diversas áreas de negocio de una organización acumula. El datawarehouse obtiene información de diversas fuentes, para análisis y accesos útiles a la información, pero generalmente no se obtiene esta información del usuario final el cual necesita una base de datos local específica sobre algún tema.

Este concepto ha ganado aceptación, en parte, por la posibilidad de realizar Datamining. El datawarehouse es usado también por otra aplicación, este es el Sistema de Soporte a la Decisión (DSS).

Decision Support System (DSS): Es una aplicación que analiza datos de negocio y lo presenta de tal manera, que los usuarios pueden tomar decisiones en forma más fácil. Esta aplicación puede presentar información gráfica y puede incluir un sistema experto o inteligencia artificial.

Evolución de los lenguajes de programación: La evolución de los programas de computación que descrita por generaciones:

Primera generación (1GL): Es un lenguaje de máquina, es decir los datos e instrucciones son entregados en el lenguaje que el procesador entiende, esto es una cadena de 0's o 1's.

Segunda generación (2GL): Es assembler, el cual consiste en declaraciones muy básicas que son convertidas a 0's o 1's. Una típica declaración se ve: ADD 3,8

Tercera generación (3GL): Es un lenguaje de alto nivel, tal como PL/I, Pascal, C, Java. Un programa Java se ve así:

```
class Hola {  
    public static void main(String[] args) {  
        System.out.println("Hola");  
    }  
}
```

Un compilador convierte las declaraciones, de un lenguaje específico de programación de alto nivel a lenguaje de máquina (en el caso de Java, la salida del compilador se llama bytecode el cual es transformado a idioma máquina por una máquina virtual.

Cuarta generación (4GL): Lenguaje diseñado para ser cercano al lenguaje natural. Los lenguajes para acceder bases de datos, son a menudo clasificados así. Un lenguaje 4GL, se vería así:

OBTIENE TODOS LAS PERSONAS DONDE "SUELDO" ES MAYOR QUE 100.

Claro que lo más probable es que esto esté expresado en Inglés.

Quinta generación (5GL): Estos lenguajes, usan una interface gráfica o visual para crear código fuente (declaraciones) que es compilado, con un compilador 3GL o 4 GL.

Mainframe : Término de la industria para un computador de gran capacidad, típicamente construido por una compañía grande. Históricamente asociado con computación centralizada, más que computación distribuida.

Plataforma: Se entiende que la plataforma es el sistema computacional base, dónde se ejecutan las aplicaciones. La plataforma está compuesta por el Sistema Operativo y el hardware sobre el cual este se ejecuta. También se entiende por plataforma cualquier base tecnológica, que sirve para que otras tecnologías o procesos sean construidos.

Programa: En computación un programa es un conjunto de instrucciones ordenadas, expresadas en algún lenguaje de programación.

Programa de aplicación: Llamado comúnmente como aplicación, es cualquier programa diseñado para una función específica, para el, usuario o en algunos casos para otros programas de aplicación.

Programación orientada al objeto: Es un revolucionario concepto que ha cambiado las reglas en el desarrollo de programas. Este concepto está organizado entorno a objetos más que a actividades, a datos más que a la lógica. Históricamente un programa ha sido visto como un procedimiento lógico en que se toman datos de entrada, que se procesan y produce datos de salida. El desafío, entonces, era cómo escribir la lógica, no como definir los datos. La orientación al objeto se preocupa más de los objetos a manipular, más que de la lógica para manipularlos. Los objetos varían desde seres humanos a construcciones. Una vez que se define un objeto, se generaliza este en una clase.

Servidor: En general un servidor es un programa computacional que provee servicios a otros programas computacionales en la misma computadora o en otras. También se le llama servidor al computador en que se ejecutan los programas computacionales servidores.

Sistema Operativo: Es el programa que después de ser inicializado en el computador mediante, otro programa llamado Boot, gestiona los demás programas en un computador.

TCP/IP (Transmission Control Protocol/ Internet Protocol): Es el lenguaje de comunicación básico o protocolo en Internet. Puede ser usado también como protocolo de comunicación en una red privada (intranet o extranet). TCP/IP es un programa de dos capas. La capa más alta (TCP), administra la división del mensaje o archivo enviado en pequeños paquetes, los cuales son transmitidos y recibidos por otra capa TCP que ensambla los paquetes para recuperar la información original. La capa más baja (IP) maneja la dirección de cada paquete de manera que estos lleguen al destino correcto. TCP/IP usa el modelo cliente/servidor de comunicación, en la cual un computador usuario (cliente) pide servicios (tal como pedir una página Web), los cuales son proveídos por otro computador (servidor).

Web Server: Un servidor Web es un software, que sirve peticiones de páginas HTML o archivos.