

C.F.G.S. DESARROLLO DE APLICACIONES INFORMÁTICAS

MÓDULO: Análisis y Diseño de Aplicaciones
Informáticas de Gestión

Unidad 6

**Análisis de necesidades
Y
Estudio de viabilidad.**



ÍNDICE DE CONTENIDOS

OBJETIVOS	1
1.- INTRODUCCIÓN	2
2.- DEFINICIÓN DE ANÁLISIS DE SISTEMAS (ANÁLISIS PREVIO)	2
3.- ANÁLISIS DE REQUISITOS DE SOFTWARE.....	5
4.- BÚSQUEDA Y DESCRIPCIÓN DE REQUISITOS FUNCIONALES.....	8
4.1.- Entrevistas	10
4.2.- Desarrollo conjunto de aplicaciones (JAD)	14
4.3.- El prototipazo.....	16
5.- ANÁLISIS DE VIABILIDAD TÉCNICA Y ECONÓMICA.....	17
5.1.- Análisis de coste y beneficios	20
6.- GUIÓN PARA UN ANÁLISIS PREVIO.....	23
SOLUCIONES AUTOEVALUACIÓN.....	24
GLOSARIO.....	25



OBJETIVOS

En esta unidad el alumno comienza a realizar los primeros análisis y documentos en los que se refleja la primera etapa del análisis de requerimientos de un proyecto software.

El alumno se va a introducir en el concepto del análisis de las necesidades de un proyecto software. Este concepto incluye qué es lo que el cliente te pide, y lo que el analista sabrá que deberá incluir, además de enseñar a realizar un estudio de viabilidad del proyecto software.





1.- Introducción.

Seguramente al **instalar** cualquiera de las aplicaciones que existen actualmente en el mercado, te habrás parado a pensar acerca de la complejidad o dificultad que habrá tenido el desarrollo de esa aplicación, incluso te habrás planteado de qué manera los desarrolladores habrán tenido en cuenta las **necesidades reales** de los potenciales usuarios de cada una de esas aplicaciones, ya que no es lo mismo realizar un proyecto para una tienda de mascotas, que para una entidad bancaria. Además también tendrás en cuenta que no es lo mismo un solo desarrollador realizando una aplicación que un grupo de ellos que han de trabajar coordinados.

Pues bien en esta **unidad**:

- Analizaremos las primeras actividades que llevan al desarrollo del software de calidad.
- Veremos cuáles son los puntos de partida de un proyecto software, es decir, abordar el estudio de las necesidades del cliente/sistema que queremos construir.
- Para ello utilizaremos técnicas de **búsqueda de información** tales como las entrevistas, el desarrollo conjunto de aplicaciones o el prototipado.
- Evaluaremos la **viabilidad** del proyecto tras este análisis, una vez obtenida una idea clara de los objetivos que se desean conseguir desde el punto de vista técnico, económico, etc.
- Obtendremos con ello un primer documento de análisis del sistema de información, que se centra en el estudio de todos los elementos del sistema, pero sin profundizar demasiado como lo hará el análisis de requisitos, con el objetivo de poder decidir si finalmente se construye el sistema o lo abandonamos por no ser viable su construcción.
- Una vez decidida la construcción del sistema y de desarrollar el software, es esencial comprender totalmente los requisitos del software. Poca importancia tiene lo bien diseñado o codificado que esté el programa si no se ha analizado correctamente, por lo tanto definiremos los principios fundamentales del análisis y qué debe contener el documento de especificación funcional del software o análisis de requisitos.

2.- Definición de análisis de sistemas (Análisis previo).

Podemos considerar el análisis del sistema como el primer paso dentro del análisis de requisitos del software. Es una primera aproximación a la Especificación de Requisitos del Software (ERS) que se concretará y se ampliará una vez decidida la construcción del software. **Se trata únicamente de un reconocimiento del problema:**





- Estudiaremos la especificación del sistema, llegando a comprender el software dentro del concepto de sistema.
- Luego se debe establecer una comunicación adecuada para el análisis, para facilitar el reconocimiento del problema.
- El analista debe establecer contacto con el equipo desarrollador y con el cliente. **El objetivo del analista es reconocer los elementos básicos del problema tal como lo ve el usuario.**
- En definitiva, se intentan conocer los requisitos de funcionamiento del sistema.

Nos vamos a meter en la piel del analista, para ello lo primero que tenemos que hacer es reconocer los objetivos del análisis del sistema, que serán:

- **Identificar las necesidades del cliente (informe de necesidades).**
Éste es el punto de partida del análisis previo. El analista se entrevista con el cliente definiendo los objetivos del sistema:
 - ✓ La información que se va a obtener (salidas).
 - ✓ La información que se va a suministrar (entradas).
 - ✓ Las funciones y el rendimiento requerido, utilizando la técnica del diagrama de caja negra o **modelo sináptico**.
 - ✓ El analista se asegura en distinguir lo que el cliente *necesita* (elementos críticos para la realización) y lo que el cliente *quiere* (elementos deseables pero no esenciales).
- **Definir diferentes alternativas de solución (modelos alternativos).**
 - ✓ Evaluación de cada una de las alternativas, indicando la **viabilidad** económica, técnica, legal y operativa (estudio de viabilidad). Se elegirá aquélla que sea más viable (que obtenga el máximo beneficio).
 - ✓ Especificación detallada de la alternativa elegida.
 - ✓ Definición de un plan de desarrollo del proyecto.
- **Evaluar las dificultades de análisis del sistema.**
 - ✓ Transformar un concepto vago en un conjunto concreto de elementos tangibles.
 - ✓ Necesidad de alta comunicación: posibilidad de errores, malos entendimientos, omisiones, etc.
 - ✓ El esfuerzo de análisis del sistema, que puede variar del 10% al 15% del total de desarrollo del software.

Por tanto **el análisis de sistemas concluye con un documento de “análisis previo” que contiene la descripción detallada de la solución propuesta y su estudio de viabilidad.** Un posible esquema para el documento de análisis previo es el siguiente:



- **Introducción.**
 - ✓ Objetivos del sistema.
 - ✓ Entorno de desarrollo.
 - ✓ Resumen del estudio de viabilidad.
 - ✓ Recursos requeridos. Coste y tiempo.
- **Descripción funcional.**
 - ✓ Entradas/funciones/salidas.
- **Solución propuesta.**
 - ✓ Descripción separada del hardware y del software.
 - ✓ Asignación de las funciones a los distintos elementos hardware y software.
- **Restricciones.**
 - ✓ De gestión, técnicas, interfaces, diseño, implementación, coste y tiempo.
- **Planificación temporal.**

En la **introducción** definimos el objetivo del sistema analizado, la influencia del entorno donde estará implantado el sistema, ver cómo influye sobre los usuarios del mismo, además se realiza un estudio de viabilidad, estimaremos los recursos necesarios tanto de coste (económico o de contratación humana) y el tiempo que tardaremos en desarrollar el proyecto.

En la **descripción funcional** haremos un estudio de la caja negra tal y como aprendimos en las primeras unidades del módulo. Después mostraremos una serie de **alternativas de solución**

Por ultimo se presentan los problemas o **restricciones** que el sistema puede tener, así como la **planificación** de desarrollo de la solución o alternativas de solución. Al final de la unidad tienes unos ejemplos de estos documentos.

PARA SABER MÁS.

Si quieres profundizar más sobre el análisis previo:

IMPROVEM Consultores: Análisis Previo a la implantación

<http://www.improven.com/Servicios/BIAnalisisEIS.aspx?ind=220&sec=28>

AUTOEVALUACIÓN

1.- Indica la opción que **NO** es un objetivo del análisis previo.

- a) Identificar las necesidades del cliente.
- b) Definir diferentes alternativas de solución (modelos alternativos).
- c) Realizar repartos de tareas.
- d) Definición de un plan de desarrollo del proyecto.

3.- Análisis de requisitos de software.

Según el estándar IEEE, el análisis de requisitos es:

- El proceso del estudio de las necesidades de los usuarios para llegar a una definición de los requisitos del sistema, de hardware y de software.
- El proceso de estudio y refinamiento de dichos requisitos.



Esto se plasma en un documento de Especificación de Requisitos del Software (ERS) o documento de Análisis de Requisitos del Software.

La definición de los requisitos debe ser el fruto del trabajo conjunto de las partes involucradas en un desarrollo:

- Los desarrolladores del software (a través de los analistas),
- Los clientes y
- Los usuarios.

El **análisis de requisitos** facilita la especificación de los requisitos esenciales (funciones, rendimiento, diseño, restricciones y atributos) del software y de sus interfaces con otros elementos del sistema. Se indica qué hay que desarrollar, no el cómo, ni el cuándo se desarrolla el software. No se indica ningún detalle del diseño del software.



El análisis de requisitos consiste en realizar las siguientes actividades:

- **Definir los requisitos del software.** Una tarea **iterativa** de definición y especificación de los requisitos del software a partir de la información obtenida durante el análisis previo.
- **Definir los requisitos de las interfaces** con el resto de elementos del sistema (usuarios, hardware y otro software) y con el exterior.
- **Integrar los requisitos en un documento de especificación de requisitos del software (ERS) o especificación funcional del software (EFS) o análisis de requerimientos.**

Por lo tanto el análisis debe:

- Representar y comprender el ámbito de información del sistema.
- Realizar modelos que representen:
 - ✓ La información (**análisis de datos**).
 - ✓ Función (**análisis funcional**).
 - ✓ El comportamiento del sistema (**análisis de control**).
- Subdividir el problema.
- Definir los requisitos del sistema con independencia de los detalles de implementación.
- Ayudar al cliente a especificar lo que desea.
- Ayudar a los desarrolladores a entender qué quiere exactamente el cliente.
- Se debe especificar lo que el usuario desea sin considerar cómo se va a desarrollar.
- Debe incluir la definición correcta de los requisitos, pero no más:
 - ✓ Funciones.
 - ✓ Rendimiento.
 - ✓ Prestaciones.
 - ✓ Restricciones de diseño.
 - ✓ Atributos.
 - ✓ Interfaces externos.
- No debe incluir detalles de diseño y gestión de proyectos.



El análisis de requisitos proporciona al analista una representación de la información, y de las funciones que se puede traducir en un diseño de datos, diseño arquitectónico y funcional.



Por último, la especificación de requisitos suministra al cliente y al analista un medio para valorar la calidad del software una vez que se haya construido.

El análisis de requerimientos supone el 15-20% del esfuerzo total del desarrollo del software. El análisis previo supone el 10-15% del esfuerzo total de desarrollo. La fase completa de análisis conlleva aproximadamente el 30% del esfuerzo.

La Estructura de una buena ERS (según IEEE) podría ser ésta:

<ul style="list-style-type: none">1. Introducción<ul style="list-style-type: none">1.1. Objetivo1.2. Ámbito1.3. Definiciones, Siglas y Abreviaturas1.4. Referencias1.5. Visión Global2. Descripción general<ul style="list-style-type: none">2.1. Perspectiva del producto2.2. Funciones del producto2.3. Características del usuario2.4. Limitaciones generales2.5. Supuestos y dependencias3. Requisitos específicosApéndicesÍndice	<ul style="list-style-type: none">3. Requisitos específicos<ul style="list-style-type: none">3.1. Requisitos funcionales<ul style="list-style-type: none">3.1.1. Requisito funcional 1<ul style="list-style-type: none">3.1.1.1. Introducción3.1.1.2. Entradas3.1.1.3. Procedimiento3.1.1.4. Salidas3.1.2. Requisito funcional 2.....3.1.n. Requisito funcional n3.2. Requisito de Interfaz externa<ul style="list-style-type: none">3.2.1. Interfaces de usuario3.2.2. Interfaces hardware3.2.3. Interfaces software3.2.4. Interfaces de comunicaciones3.3. Requisitos de ejecución3.4. Restricciones de diseño<ul style="list-style-type: none">3.4.1. Acatamiento de estándares3.4.2. Limitaciones hardware3.5. Atributos de calidad<ul style="list-style-type: none">3.5.1. Seguridad3.5.2. Mantenimiento3.6. Otros requisitos<ul style="list-style-type: none">3.6.1. Base de datos3.6.2. Operaciones3.6.3. Adaptación de situación
---	--

AUTOEVALUACIÓN

2.- Una ERS (Especificación de Requisitos del Software) debe:

- a) Representar y comprender el ámbito de información del sistema.
- b) Definir los requisitos del sistema dependiendo de los detalles de implementación.
- c) Especificar lo que el usuario desea considerando cómo se va a desarrollar.
- d) Incluir detalles de diseño y gestión de proyectos.

PARA SABER MÁS.

Para conocer más sobre el análisis de requerimientos:

Manual de desarrollo de software de la wikilearning

http://www.wikilearning.com/analisis_de_requerimientos-wkccp-3471-3.htm

4.- Búsqueda y descripción de requisitos funcionales.

¿Cómo comenzamos a identificar y describir los **requisitos funcionales** de un determinado problema?

El análisis de requisitos siempre comienza con una **comunicación** entre dos o más partes.

- Existe un **cliente** que tiene un problema al que se le puede dar una solución basada en ordenador.
- El **analista** responde a la petición del cliente y comienza una comunicación.



Las técnicas de recogida de información surgen como un medio para mejorar la comunicación entre usuarios/clientes y los desarrolladores de software.

El hecho es que **los técnicos de software** normalmente no conocen todos los detalles del trabajo de la empresa para la cual se va a desarrollar la aplicación y **los usuarios** no saben qué información es necesaria o relevante para el desarrollo de una aplicación. Para facilitar la colaboración de ambos mundos (usuarios y desarrolladores) en el proceso de análisis de los requisitos, se recurre a las técnicas de comunicación y recopilación de información.

El **proceso de análisis** debería seguir los siguientes pasos:

- Identificar las fuentes de información relevantes para el proyecto (usuarios).
- Realizar las **preguntas** apropiadas para comprender sus necesidades.
- **Analizar la información** recogida para detectar los aspectos que quedan poco claros.
- Confirmar con los usuarios lo que parece haberse comprendido de los requisitos.
- **Sintetizar** los requisitos en un documento de especificación apropiado.

El resultado de este proceso es un **documento** que especifique lo más claramente posible los requisitos que debe cumplir el software. Para obtener la información necesaria, para generar dicha especificación o bien para realizar un análisis de viabilidad, usaremos una serie de técnicas de recogida de información siendo las siguientes las más utilizadas:



- **Entrevistas.** Es la técnica más empleada y la que requiere una mayor preparación y experiencia por parte del analista. Es similar a una entrevista periodística en la que el desarrollador entrevista uno a uno a los futuros usuarios del software.
- **Desarrollo conjunto de aplicaciones (JAD).** Se crean equipos de usuarios y analistas que se reúnen para trabajar conjuntamente en la determinación de las características que debe tener el software para satisfacer las necesidades de los usuarios.
- **Prototipado.** Consiste en la construcción de un modelo o *maqueta* del sistema que permite a los usuarios evaluar mejor sus necesidades.
- **Observación.** Consiste en analizar in situ cómo funciona la unidad o el departamento que se quiere informatizar.
- **Estudio de documentación.** En casi todas las organizaciones existen documentos que describen el **funcionamiento** del negocio, desde planes estratégicos hasta manuales de operación. El analista debe estudiar esta documentación para hacerse una idea de la normativa que rige la empresa. También es conveniente que recopile muestras de los impresos que se utilizan, ya que nos permiten conocer datos que se manejan.
- **Cuestionarios.** Resultan útiles para recoger información de un gran número de personas en poco tiempo, especialmente en situaciones en las que se da gran dispersión geográfica.
- **Tormenta de ideas.** Consiste en reuniones de cuatro a diez personas (usuarios) en las cuales, se sugieren toda clase de ideas sin juzgarse su validez, por muy disparatadas que parezcan. En una segunda fase, se realiza un análisis detallado de cada propuesta.



En la realidad no se usa solamente una de estas técnicas por separado, sino una combinación de algunas de ellas.

AUTOEVALUACIÓN

3.- Los pasos a seguir en el proceso de Análisis son:

- Identificar las fuentes de información relevantes para el proyecto (usuarios).
- Realizar las preguntas apropiadas para comprender sus necesidades.
- Analizar la información recogida para detectar los aspectos que quedan poco claros.
- Confirmar con los usuarios lo que parece haberse comprendido de los requisitos.
- Sintetizar los requisitos en un documento de especificación apropiado.
- Todas las respuestas anteriores son correctas.
- Ninguna de las respuestas anteriores es correcta.

PARA SABER MÁS.

Para conocer más sobre técnicas de recogida de información:

Técnicas de recogida de información

<http://www.marketing-xxi.com/principales-tecnicas-de-recogida-de-informacion-27.htm>

4.1.- Entrevistas.

¿Qué pretendemos conseguir mediante la realización de una **entrevista**?

La entrevista se puede definir como un **intento sistemático de recoger información de otra persona a través de una comunicación interpersonal que se lleva a cabo por medio de una conversación estructurada.**

El aspecto más destacable de la entrevista es que se propone un fin ajeno al simple placer de la conversación, por lo que su preparación resulta esencial para cumplir sus objetivos.

Debe quedar claro que **no basta con hacer preguntas** para obtener toda la información necesaria. Es muy importante la forma en la que se plantea la conversación y la relación que se establece en la entrevista. Por ello, el entrevistador tiene que poseer una serie de cualidades, tales como ser:

- Imparcial.
- Ponderado.
- Buen oyente, capaz de **escuchar activamente**, adaptándose y manteniendo el ritmo de la conversación.

Y además debe tener:

- Cierta grado de habilidad en el trato.
- Cordialidad y accesibilidad.
- Paciencia.



En general, es muy importante que muestre interés y entusiasmo por el trabajo que está desarrollando y que sepa transmitirlo a los entrevistados.

Mario Gil Piattini, que es un experto a nivel mundial en Ingeniería del Software, define las diferentes fases que se pueden distinguir en una entrevista, que son las siguientes:

a) Preparación.

- Durante esta fase **el entrevistador debe documentarse e investigar la situación de organización, analizando los documentos de la empresa disponibles.** También se debe recurrir, a información de fuentes externas: folletos, informes sobre el sector, publicaciones, etc. Esta preparación es esencial para una entrevista eficaz.
- El **entrevistador debe conocer los conceptos básicos del negocio o actividad.** La entrevista ha de centrarse en aquellos aspectos del trabajo no accesibles por otros medios (como la observación y el análisis de documentos) que son, en general, aquellos que están sólo en la mente del entrevistado.
- Otra actividad importante es la **identificación de las personas a las que se debe entrevistar.** La mayoría de los analistas adoptan un enfoque top-down, comenzando a entrevistar a directivos (que pueden ofrecer una visión global) y terminando por hablar con los empleados que puedan aportar detalles importantes del funcionamiento de la empresa. El objetivo principal es minimizar el número necesario de personas a entrevistar para obtener una visión lo más completa posible sobre el sistema que se debe desarrollar.
- Hay que **planificar el lugar y la hora** en la que se va a desarrollar la entrevista, tratando de adaptarse a la agenda del entrevistado.
- Algunas veces se realiza **un cuestionario previo** que debe rellenar el entrevistado y un pequeño documento de introducción al proyecto de desarrollo. El cuestionario permite que el entrevistado conozca los temas que se van a tratar y el analista recoja información valiosa para preparar la conversación.

b) Realización.

Es la parte esencial de la entrevista. Como en la mayoría de las interacciones humanas, conviene seguir el protocolo o serie de actos habituales para las conversaciones y reuniones que marcan las costumbres sociales.

Se distinguen tres etapas en el acto de la entrevista: **apertura, desarrollo y terminación.**

- **Apertura.** Se comienza presentándose e informando al entrevistado sobre:
 - ✓ El motivo de la entrevista.
 - ✓ Qué esperamos de él.
 - ✓ Cómo utilizaremos la información obtenida.
 - ✓ La mecánica de las preguntas.
 - ✓ Etc.



Esta parte es fundamental para que se cree un ambiente confortable que ayude a que la siguiente fase sea productiva.



- **Desarrollo.** Es la entrevista propiamente dicha. No debería prolongarse durante más de **dos horas**. Durante el desarrollo de la entrevista, debe procurarse lo siguiente:
 - ✓ Que el **entrevistador** hable un **20%** del tiempo, mientras que el **entrevistado** sea el que ocupe el **80%** restante.
 - ✓ **Evitar los monólogos** y conseguir que constituya un verdadero diálogo.
 - ✓ **Es muy importante en todo momento cumplir las reglas del protocolo, no interrumpir al entrevistado y hacer ver que nos está ayudando.**
 - ✓ Es importante mantener en todo momento el **control de la entrevista**, afrontando los problemas que puedan surgir y que suelen tener principalmente dos causas:
 - Discrepancia de objetivos entre el emisor y el receptor del mensaje
 - Barreras de comunicación, como problemas en el significado de las palabras, distinta percepción de la realidad, etc.

Es posible que el entrevistado sea una persona difícil, que está a la defensiva, distraído, etc. Habrá que determinar entonces la causa de este comportamiento y tratar de corregirla.

El formato de respuestas para las preguntas puede ser abierto o cerrado. Las preguntas para respuesta abierta permiten a los entrevistados dar cualquier respuesta que parezca apropiada. Pueden contestar por completo con sus propias palabras. Con las preguntas para respuesta cerrada se proporcionan al usuario un conjunto de respuestas de entre las que se pueda seleccionar la que considere más adecuada. Todas las personas que responden se basan en un mismo conjunto de posibles respuestas.

Los analistas también deben dividir el **tiempo** entre desarrollar preguntas para entrevistas y analizar respuestas. La **entrevista no estructurada** requiere menos tiempo de preparación, porque no necesita tener por anticipado las palabras precisas de las preguntas. Analizar las respuestas después de la entrevista lleva más tiempo que con las **entrevistas estructuradas**. El mayor costo radica en la preparación, administración y análisis de las entrevistas estructuradas para preguntas cerradas.

Ejemplo de pregunta con respuesta cerrada en la entrevista estructurada

Ejemplo: obtener la información sobre las características diseño, críticas para los empleados.

"...algunos empleados han sugerido que la mejor forma para hacer eficiente el procesamiento de pedidos es instalar un sistema informatizado que maneje todos los cálculos..."

bajo estas circunstancias ¿apoyaría usted el desarrollo de un sistema de este tipo?.



Ejemplo de pregunta con respuesta abierta en la entrevista estructurada

Ejemplo: obtener la información sobre las características de diseño, críticas para los empleados.

"La experiencia le ha proporcionado una amplia visión en cuanto a la forma en la que la empresa maneja los pedidos..." Me gustaría que usted contestara algunas preguntas específicas en relación con lo anterior:

- ¿Qué etapas trabajas bien? ¿Cuáles no?
- ¿En dónde se presenta la mayor parte del problema?
- ¿Cuándo ocurre un atraso, cómo se maneja?

- **Terminación.** Se termina **recapitulando** la entrevista, **agradeciendo** su esfuerzo al entrevistado y **citándole** para una nueva entrevista si fuera necesaria.
 - ✓ Es importante dejar siempre abierta la posibilidad de volver a contactar para aclarar temas que surjan al estudiar la información o al contrastarla con otros entrevistados.
 - ✓ También hay que convencer al entrevistado de que se le ha entendido.

c) Análisis.

Es quizás la fase más descuidada, pero no por ello la menos importante. Se trata de:

- ✓ Leer las notas y pasarlas a limpio,
- ✓ Reorganizar la información,
- ✓ Contrastarla con otras entrevistas o fuentes de información, etc.
- ✓ También es importante evaluar cómo ha ido la entrevista y qué aspectos se pueden mejorar para las siguientes.

AUTOEVALUACIÓN:

4.- En el desarrollo de una entrevista, el reparto de intervención ideal es:

- a) Analista 80%, entrevistado 20%.
- b) Analista 70%, entrevistado 30%
- c) Analista 50%, entrevistado 50%
- d) Analista 20%, entrevistado 80%.



PARA SABER MÁS.

Más sobre técnicas de recogida de información:

Técnicas de recogida de información

<http://www.oadi.dip-caceres.org/vprofe/virtualprofe/cursos/c103/evaluacion5.htm>

4.2.- Desarrollo conjunto de aplicaciones (JAD).

Posiblemente hayas pensado que hacer las **entrevistas** que se han descrito en el apartado anterior tiene serios inconvenientes. Pues estás en lo cierto, y por eso se intentan evitar mediante el uso de otras técnicas, como es el caso de JAD (Joint Application Development)

JAD es una técnica que se utiliza para promover la cooperación y el trabajo en equipo entre los usuarios y los analistas, consiste en una serie de **reuniones**, en total entre dos y cuatro días, en las que participan analistas de software junto a varios usuarios **cualificados**. La idea es aprovechar la dinámica de grupos, toda clase de ayudas visuales de comunicación y comprensión de soluciones (proyectors, ordenadores, pizarras, etc.), un proceso de trabajo sistemático y organizado y una filosofía de documentación.



Las entrevistas que hemos estudiado en el apartado anterior presentan los siguientes inconvenientes, a los que el uso de JAD pretende dar solución:

- Se requiere **mucho tiempo** para las entrevistas, tanto en prepararlas y hacerlas como también en redactar el conjunto de requisitos, a partir de opiniones diferentes de los distintos entrevistados. Las sesiones JAD no requieren tanta preparación.
- Como sólo el analista revisa la especificación de requisitos tras las entrevistas, es más difícil apreciar posibles errores en él. Por el contrario, en el JAD todo el grupo puede actuar como revisor y detectar defectos.
- El JAD propugna una **participación** más profunda de los usuarios en el proyecto, lo que produce una especificación de requisitos más adaptada a las necesidades reales.

El proceso de un JAD se basa en lo que se denominan *sesiones JAD*.

Una sesión JAD consiste en:

- Una serie de **reuniones estructuradas** y organizadas
- Donde se parte de un documento de trabajo que hay que analizar para completar el conjunto de requisitos del sistema.
- Durante la propia sesión se creará la especificación de los requisitos, en ocasiones se utilizarán **herramientas CASE**.
- Al **final** de la sesión se tendrá concluido un **documento** de especificación aprobado por los presentes. El jefe del JAD es el responsable de conseguir que la reunión sea productiva y de mantener el orden.



Las empresas que han implantado este método han informado de importantes **ahorros de tiempo** en el desarrollo de software, así como de una mayor satisfacción de los usuarios con los sistemas construidos, aunque en la realidad no son muchos los que lo utilicen por la dificultad de las empresas para contar con el tiempo de dedicación que requiere por parte de los usuarios.

AUTOEVALUACIÓN:

5.- Refiriéndonos a la técnica JAD, podemos afirmar que ...

- Todo el grupo puede actuar como revisor pero no detectar defectos.
- No propugna una participación más profunda de los usuarios en el proyecto.
- Consiste en una serie de reuniones estructuradas y desorganizadas.
- Promueve la cooperación y el trabajo en equipo entre los usuarios y los analistas.

PARA SABER MÁS.

Artículo sobre el desarrollo conjunto de aplicaciones:

Desarrollo conjunto de aplicaciones (JAD)

<http://es.geocities.com/addrianortega/JADN.htm>

4.3.- El prototipado.

¿Qué ocurre si a pesar de haber hecho entrevistas o haber usado la técnica JAD, obtenemos una especificación de requisitos no demasiado ajustada a la realidad?

Puede suceder que hayamos dedicado un tiempo considerable a construir una aplicación en base a unos **requisitos incompletos o erróneos**, habiendo desperdiciado posiblemente bastante tiempo y trabajo. Es más, es posible que el propio usuario no tuviera muy claro lo que quería o lo que necesitaba, y por eso no nos lo supo explicar.



¿Piensas que sería útil presentarle una especie de **esqueleto de la aplicación**, en base a los requerimientos que hemos capturado, para que pueda tener una idea más clara de cual va a ser el resultado? Seguramente de esta forma conseguiremos **detectar fallos o carencias** antes de embarcarnos en el desarrollo completo de la aplicación.

El **prototipado** consiste en la construcción de un **modelo** del sistema o prototipo que se construye para evaluar mejor los requisitos que se desean cumplir. **Se usa** especialmente cuando:

- El área de la aplicación no está bien definida.
- El coste del rechazo de la aplicación por los usuarios, por no cumplir sus expectativas, es muy alto.



Estos modelos o prototipos son **versiones reducidas, demos o simulaciones** de pantallas de la aplicación a desarrollar.

Las dos razones principales para emplear prototipado, ordenadas por frecuencia de uso son:

1. **Prototipado de la interfaz de usuario para asegurarse de que está bien diseñada, que satisface las necesidades de quienes deben usar la aplicación.** Este tipo de prototipado es bastante frecuente, no es costoso y está formado por simples modelos de pantallas en papel, simuladas con programas de dibujo, o de presentación de la interfaz.
2. **Prototipado funcional. El prototipo supone una primera versión del sistema con funcionalidad limitada.** A medida que se comprueba si las funciones implementadas son las apropiadas, se corrigen, refinan o se añaden otras nuevas hasta llegar al sistema final.

La cualidad esencial de un prototipo debe ser la posibilidad de ser construido más rápidamente que la aplicación correspondiente. Por esta razón, se suele hablar de este proceso de desarrollo como **prototipado rápido**.

Las herramientas empleadas en el prototipado pueden ser muy variadas:

- En el **nivel inferior** se encuentran herramientas comunes y fácilmente disponibles como programas de dibujo o de presentación, hojas de cálculo o generadores de informes. El usuario puede ver cómo quedará la entrada/salida de la aplicación cuando esté acabada.
- En un nivel un poco más **evolucionado** se situarían algunos gestores de bases de datos y sistemas de cuarta generación que permiten prototipos más sofisticados que incluyen no sólo interfaces, sino también prototipado del manejo de datos.
- Otra opción son las posibilidades de prototipado que proporcionan ciertas herramientas CASE o determinados generadores de aplicaciones.

PARA SABER MÁS.

Ejemplo de empresa que realiza prototipado rápido:

Roboticspot

<http://www.roboticspot.com/robotica/serv.shtml>

5.- Análisis de viabilidad técnica y económica.

Todos los proyectos serían realizables si se contara con recursos ilimitados y un tiempo infinito.

Pero claro, sabemos que en el mundo real ni los recursos son ilimitados ni el tiempo disponible infinito.

¿Qué limitaciones tendrá todo proyecto que emprendamos? Como imaginas, deberá ajustarse a un presupuesto y ejecutarse en unos plazos.



Desafortunadamente, el desarrollo de un sistema basado en soluciones informáticas se caracteriza por la **escasez de recursos** y la dificultad de cumplir los **plazos de entrega**. Es necesario y prudente evaluar la viabilidad de un proyecto lo antes posible. Se pueden evitar meses o años de esfuerzo, millones de euros y una inversión profesional incalculable, si un sistema mal concebido es reconocido al principio de la etapa de definición.

Nuestro **objetivo** será comprobar la posibilidad de hallar una solución informática a los problemas observados, antes de gastar demasiado dinero.

Durante el análisis del sistema centramos nuestra atención en cinco áreas de interés básico:

- **Viabilidad económica.** Una evaluación del coste de desarrollo frente al beneficio final producido por el sistema desarrollado.
- **Viabilidad técnica.** Un estudio de la funcionalidad, el rendimiento y las restricciones que pueden afectar a la posibilidad de realización de un sistema aceptable.
- **Viabilidad legal.** Una determinación de cualquier infracción, violación o ilegalidad que pudiera resultar del desarrollo del sistema (por ejemplo, debe comprobarse que se cumple la Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal (LOPD) o la Ley 34/2002, de Servicios de la Sociedad de la Información y de Comercio Electrónico (LSSI o LSSICE) y otras asociadas, y en definitiva cualquier reglamentación en vigor)
- **Viabilidad Operativa.** Determinar si se puede implantar de manera efectiva en la empresa.
- **Viabilidad Social.** Estudio del grado de aceptación de los usuarios.



No será necesario llevar a cabo un estudio de viabilidad para sistemas en los que la justificación económica es obvia, el riesgo técnico es bajo, se esperan pocos problemas legales y no existe una alternativa razonable. Sin embargo, cuando no se da alguna de las anteriores condiciones, debe realizarse el estudio.

La **justificación económica** es normalmente la principal consideración para la mayoría de los sistemas (excepciones notables son los sistemas impuestos por la ley o de empresas públicas). La justificación económica comprende un amplio rango de aspectos, entre los que se encuentran:

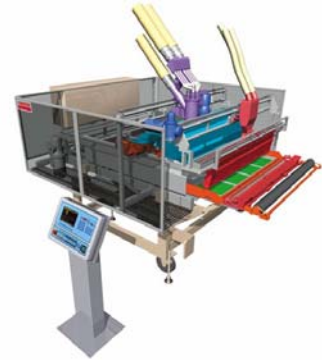
- El análisis de coste-beneficio (tratado en el punto siguiente).
- Las estrategias de ingresos a largo plazo.
- El impacto en otros productos o en centros de explotación.
- El coste de los recursos que se necesitan para el desarrollo.
- El crecimiento potencial del mercado.



La **viabilidad técnica** es frecuentemente el área más **difícil** de evaluar en esta etapa del proceso de desarrollo del sistema. Debido a que los objetivos, las funciones y el rendimiento son de alguna manera confusos, cualquier cosa puede parecer posible si se hacen las consideraciones adecuadas. En sistemas de gestión no existe tal problema, la viabilidad técnica va a ser siempre positiva.

Las consideraciones que van asociadas normalmente a la **viabilidad técnica** son:

- **Riesgo del desarrollo.** ¿Puede el elemento del sistema ser diseñado de tal forma que las funciones y el rendimiento necesarios se consigan dentro de las restricciones determinadas en el análisis?
- **Disponibilidad de recursos.** ¿Hay personal cualificado para desarrollar el elemento del sistema en cuestión? ¿Están disponibles para el sistema otros recursos necesarios (de hardware y de software)?
- **Tecnología.** ¿Ha progresado la tecnología relevante lo suficiente como para poder soportar el sistema?



La **viabilidad legal** comprende un amplio rango de aspectos que incluyen los contratos, la responsabilidad, las infracciones y muchos más detalles frecuentemente desconocidos para el personal técnico.

Como producto de los estudios de viabilidad, se define **un informe o estudio de viabilidad**, con el objetivo de que el cliente pueda decidir seguir/no seguir (se incluye este estudio en el documento de análisis de sistemas).

Este **Informe del Estudio de Viabilidad**, podría tener la siguiente estructura:

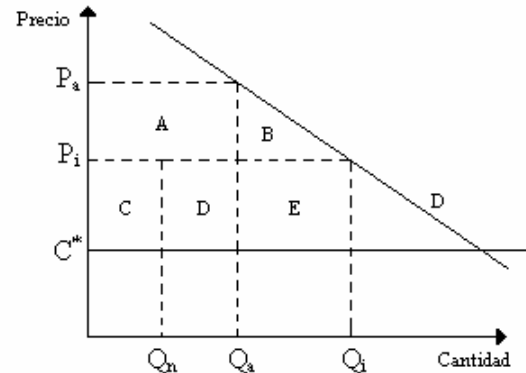
1. Entorno y restricciones del problema.
2. Alternativas y sus configuraciones.
3. Análisis económico (coste/beneficio).
4. Viabilidad técnica.
5. Viabilidad legal.

5.1.- Análisis de coste y beneficios.

Entre la **información** más relevante que contiene el estudio de viabilidad se encuentra el **análisis de coste-beneficio**, que es una **evaluación de la justificación económica** para un proyecto de sistema basado en ordenador.

El análisis de coste-beneficio señala los costes del desarrollo del proyecto y los contrasta con los beneficios tangibles (esto es, medibles directamente en euros) e intangibles del sistema.

El **análisis de coste-beneficio** es complicado porque los criterios varían según las características del sistema a desarrollar, el tamaño relativo del proyecto y recuperación esperada de la inversión como parte del plan estratégico de la compañía. Además, muchos beneficios obtenidos de los sistemas basados en soluciones informáticas son **intangibles** (p. ej.: una mejor calidad del diseño mediante una optimización iterativa, una mayor satisfacción del cliente debida a un control programable y unas mejores decisiones comerciales a partir de datos de ventas con formatos previamente analizados). Puede ser difícil lograr comparaciones directas cuantitativas.

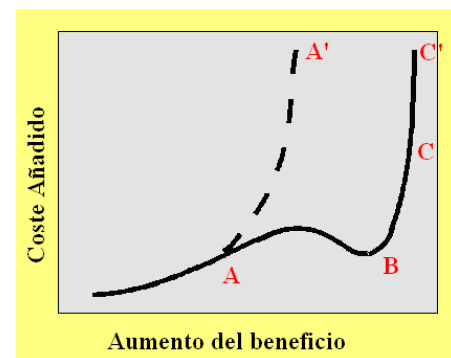


El analista debe **estimar cada coste** y luego utiliza los costes del desarrollo y los que surjan sobre la marcha para determinar la recuperación de la inversión, el punto de igualdad y el período de amortización. Contabilizando los beneficios intangibles, el director administrativo decide si los resultados económicos justifican el sistema.

Otro aspecto del análisis de coste-beneficio es la consideración de los **costes incrementales asociados con los beneficios** añadidos (mayor o mejor funcionalidad y rendimiento). Para los sistemas basados en soluciones informáticas, la relación incremental de coste-beneficio se puede representar como en la siguiente gráfica:

En algunos casos (**curva AA'**) los costes se incrementan proporcionalmente a los beneficios hasta un determinado punto. Después de ese punto, cada beneficio adicional es demasiado caro.

En otros casos (**curva ABCC'**), los costes aumentan proporcionalmente hasta A y después se nivelan a favor de los beneficios añadidos (hasta B), antes de aumentar drásticamente (en C) para los posteriores beneficios.



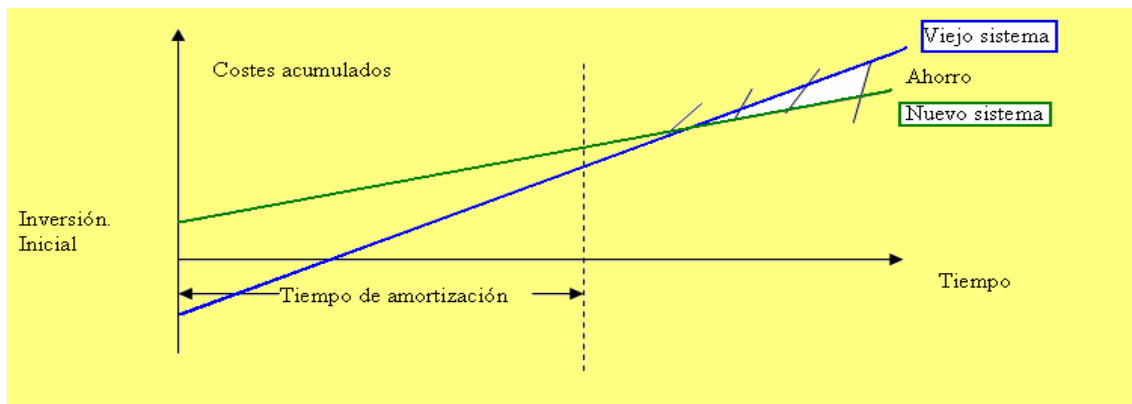
Sólo gastando el tiempo necesario para evaluar la viabilidad, reducimos las oportunidades de situaciones extremadamente embarazosas en etapas posteriores de un proyecto de un sistema. El esfuerzo gastado en el análisis de viabilidad, aunque termine aconsejando la cancelación de un proyecto propuesto, no es un esfuerzo desaprovechado.

En general, el **análisis coste-beneficio** se suele representar en forma de tabla:

- En las columnas aparecen los años de vida del proyecto, y
- En las filas los distintos conceptos de gasto y beneficio del proyecto.

Consideraciones sobre los costes:

Hay que evaluar el tiempo de recuperación de la inversión, el punto de igualdad y el periodo de amortización.



Posibles costes de un sistema de información:

- **Costes iniciales.**
 - ✓ Costes de compra e instalación de equipos.
 - ✓ Costes de acondicionamiento del lugar (mobiliario, obras,...)
 - ✓ Costes del personal informático.
 - ✓ Costes de compra de software general (Sistema Operativo, Sistema Gestor de Bases de Datos, ofimática,...).
 - ✓ Costes de desarrollo del software concreto.
 - ✓ Costes de formación del personal de la organización, etc.

- **Costes continuos:**

- ✓ Mantenimiento del sistema (hardware y software).
- ✓ Coste de uso de luz, teléfono, etc.

Por su parte, los beneficios pueden aparecer de muy diferentes maneras:

- Reducción o eliminación de costes (RC)
- Reducción o eliminación de errores. (RE)
- Aumento de velocidad en la actividad. (AV)
- Aumento de fiabilidad (resultado final más creíble). (AF)
- Mejora en la gestión del control y de la planificación. (MG)

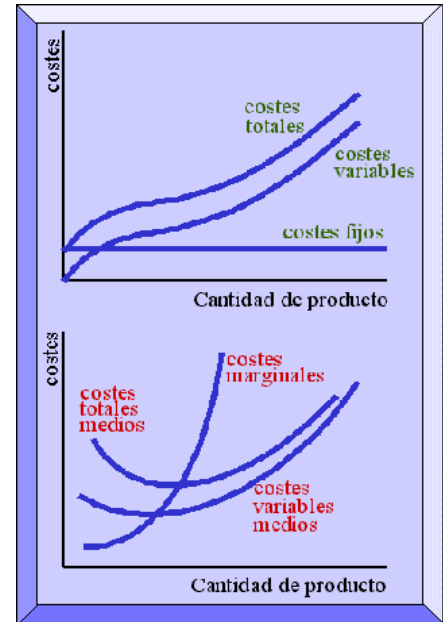
Estos beneficios se pueden dar en uno o varios apartados en la siguiente clasificación:

- **Beneficios tangibles:**

- ✓ Coste de reposición: coste de mantenimiento y uso del sistema antiguo que ahora se ahorra. Ej. Más trabajo con mismo personal, igual trabajo con menos personal.
- ✓ Mayor eficacia en el control de la empresa. Ej. Reducción del nivel de existencias en almacén.

- **Beneficios intangibles:**

- ✓ Mejor servicio al cliente.
- ✓ Mayor satisfacción de los empleados.



Uno de los mayores **problemas** para que el análisis económico sea realista es que la **valoración de los beneficios** es, en muchos casos, necesariamente **subjetiva**. Esto constituye una gran desventaja para convencer a la dirección de las ventajas del proyecto. Uno de los parámetros más utilizados en la valoración económica de los proyectos es el **plazo de amortización** necesario para recuperar el dinero invertido que en términos de economía se denomina con las siglas ROI (retorno sobre inversión, del inglés Return Over Inversion)

Para terminar este punto conviene resaltar ciertos aspectos importantes para un análisis de coste-beneficio eficaz:



1. **La mayoría de las estimaciones** de costes y de beneficios suelen consistir en rangos de valores probables. La dirección, que decidirá la continuidad o no del proyecto, no puede pretender estimaciones precisas en esta etapa. A medida que el proyecto avance, se puede refinar el análisis económico.
2. Es recomendable hacer estimaciones más bien pesimistas o conservadoras, ya que suele cumplirse la ley de Murphy: “si algo puede fallar, fallará”.

AUTOEVALUACIÓN

6.- Indica qué opción de entre las que te proponemos es un coste.

- a) Mantenimiento del sistema (hardware y software).
- b) Reducción o eliminación de errores.
- c) Aumento de velocidad en la actividad.
- d) Aumento de fiabilidad (resultado final más creíble).
- e) Todos los anteriores.

6.- Guión para un análisis previo.

Índice de contenidos para un análisis previo:

- 1.- Objetivos de sistema.
- 2.- Entorno de desarrollo.
- 3.- Resumen del estudio de viabilidad.
 - Económica.
 - Técnica.
 - Legal.
 - Operativa.
 - Social.
- 4.- Recursos requeridos. Coste. Tiempo.
- 5.- Descripción Funcional (Caja negra).
 - Entradas.
 - Funciones.
 - Salidas.





6.- Solución Propuesta.

- Hardware.
- Software.
- Asignación de las funciones a los distintos elementos hardware y software.

7.- Restricciones del Proyecto.

8.- Planificación temporal.

- Diagrama PERT del proyecto.
- Diagrama Gantt.

SOLUCIONES AUTOEVALUACIÓN

- 1.- C
- 2.- A
- 3.- F
- 4.- D
- 5.- D
- 6.- A





GLOSARIO

Cualificado.

Conjunto de conocimientos y capacidades, incluidos los modelos de comportamientos y las habilidades que se adquieren durante los procesos de socialización y de educación de los individuos. En concreto en este tema, el término “usuario cualificado”, se refiere a la persona que tiene habilidades para manejar algún tipo de programa de ordenador, o que no le presenta ningún tipo de problema su aprendizaje.

Estándar IEEE.

IEEE corresponde a las siglas del Institute of Electrical and Electronics Engineers, Instituto de Ingenieros Eléctricos y Electrónicos, una asociación estadounidense dedicada a la estandarización. Es una asociación internacional sin fines de lucro formada por profesionales de las nuevas tecnologías, como Ingenieros de telecomunicaciones, Ingenieros electrónicos, Ingenieros en informática.

Herramienta CASE.

Las Herramientas CASE (Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, documentación o detección de errores entre otras.

Interfaces.

En informática esta palabra tiene multitud de significados, por ejemplo una interfaz de usuario es una pantalla creada para mostrar los datos de una forma amigable a los usuarios. Éste es el sentido que se da al término en esta unidad.

Iterativa.

Un proceso iterativo es aquel que se repite una serie de veces, normalmente hasta que se cumpla alguna condición para su terminación.

Modelo sináptico.

Se refiere a un modelo en el cual todos sus componentes se encuentran interconectados siguiendo algún tipo de razón lógica.

Viabilidad.

Investigación encaminada a establecer las posibilidades de éxito de una determinada actividad dados unos recursos disponibles y unas limitaciones existentes, como por ejemplo, unos plazos de ejecución.