

## 1. Caso práctico

### Caso práctico

Cada vez que en **SI Andalucía** afrontan el desarrollo de una nueva aplicación, **María** se encarga de hacer el análisis de la misma, y entre las tareas que incluye ese análisis están la elaboración del **análisis de necesidades** o requisitos y un estudio de **viabilidad**.

En concreto **María** ha hecho recientemente el análisis y diseño de una aplicación para una **empresa constructora** en Roquetas de Mar, llamada **RoKemar**.

Como resultado ha obtenido una serie de documentos, derivados del análisis previo (análisis de Sistemas) para ver:

- qué sistema usan actualmente,
- qué necesidades tiene el cliente, y
- qué hace el sistema.

Esto se plasma en un **informe de necesidades**. Básicamente se trata de actualizar y extender con nuevas funcionalidades un sistema informático que ya estaba implantado, y que gestionaba los presupuestos de las obras, las ventas de promociones terminadas, la información de los clientes y proveedores, y la contabilidad. Los cambios que solicita el cliente se refieren a introducir la posibilidad de que los trabajadores puedan acceder a las **funcionalidades** que ya tiene el sistema vía web, desde cualquier ordenador con conexión a Internet, y que los clientes de la constructora puedan consultar alguna información también mediante conexión web.

**María** también ha elaborado un documento con modelos alternativos, que poder **evaluar** para que el cliente elija la solución que más se adapte a sus necesidades y su presupuesto. Esto se debe a que las cosas podrían hacerse de formas distintas a como se hacen actualmente. Cada una de estas soluciones alternativas ha sido objeto de un **estudio de viabilidad técnica**, económica, legal, operativa y social, de forma que el cliente tendrá todos los elementos posibles para decidir con el mejor criterio la solución a adoptar, en base al análisis de costes y beneficios.

Naturalmente, todos estos informes deben incluir una recomendación de la solución que a juicio de **María** es la más adecuada para este caso. Por eso **Víctor** se queda sorprendido y no entiende que, tras tanto trabajo, **el consejo de María haya sido que no es recomendable cambiar el sistema para incluir las nuevas funcionalidades que pide el cliente**, ya que el coste sería alto, y el tiempo de amortización excesivamente largo, no resultando imprescindible modificar la forma de trabajo del sistema debido a que el actual volumen de negocio haría que los beneficios no compensaran a los gastos. Por otra parte, se plantean algunos problemas de seguridad para permitir el acceso de algunos empleados a datos sensibles a través de Internet, cuya solución encarecería más aún el proyecto.

Pero más se sorprende **Víctor** cuando **el gerente de la empresa RoKemar decide que a pesar de todo quiere introducir esos cambios**. De hecho elige una de las soluciones sugeridas por **María**, y no precisamente la más económica.

**María** le explica a **Víctor** que el análisis previo y los estudios de viabilidad se hacen justamente para evitar que tanto la empresa como ellos se embarquen en unos cambios costosos que sin embargo no aporten ningún beneficio a la empresa, y que por eso su obligación como profesionales es aconsejar no continuar adelante con el proyecto si este no es claramente viable.

- "¿Y como se entiende entonces que a pesar de todo el gerente decida continuar con él, habiendo sido negativo nuestro consejo?" - pregunta **Víctor**.

- "El cliente ha valorado otra serie de beneficios intangibles, como la satisfacción de los trabajadores y de los clientes, la imagen de empresa actualizada que la nueva aplicación proporcionará, al publicidad que podrá hacer en la web, etc, muy por encima de lo que nosotros lo habíamos hecho, y en última instancia, él es el que toma la decisión al respecto, en

*base al conocimiento que tiene de su negocio, especialmente en esos puntos intangibles que él conoce bien, y que a nosotros como informáticos nos resultan más ajenos"- es la respuesta lógica de **María**.*

## 2. Introducción

### 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.

## 3. Definición de análisis de sistemas (Análisis previo)

### Definición de análisis de sistemas (Análisis previo)

¿Qué hace el sistema?.

En este punto, eso es básicamente lo que queremos saber y lo que debemos responder.

**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 aquella 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** es otra tarea a realizar:

- 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:

1. **Introducción.**

- Objetivos del sistema.
- Entorno de desarrollo.
- Resumen del estudio de viabilidad.
- Recursos requeridos. Coste y tiempo.

2. **Descripción funcional.**

- Entradas/funciones/salidas.

3. **Solución propuesta.**

- Descripción separada del hardware y del software.
- Asignación de las funciones a los distintos elementos hardware y software.

4. **Restricciones.**

- De gestión, técnicas, interfaces, diseño, implementación, coste y tiempo.

5. **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 (que veremos en el punto 7 de esta unidad), 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 último 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:*

## 4. Análisis de requisitos de software

### Análisis de requisitos de software

CASO.

*Víctor y Carmen le han resultado de gran ayuda a María en la elaboración del análisis de requisitos software de la aplicación para la constructora RoKemar, y en la elaboración de los documentos resultado de este análisis. En algunos momentos le han ayudado a **distinguir lo que el cliente quería de lo que realmente necesitaba**, que no es siempre lo mismo. Y sobre todo le han ayudado a idear distintas **soluciones alternativas** que permitan satisfacer de la manera más eficiente los requisitos que se han establecido.*

*Por ejemplo, la información que generan los comerciales que van enseñando las viviendas a los clientes, se almacena en la base de datos cuando vuelven a la oficina, pero hay días que eso no sucede hasta el final de su jornada de trabajo, porque van de visita en visita con los clientes, sin tiempo para pasarse por la oficina. Eso provoca que los demás compañeros no tengan constancia de las ventas y alquileres que han realizado hasta el final de la jornada, generándose malentendidos y confusiones desagradables para los clientes. Carmen propone como solución el acceso a la base de datos vía web, mediante una conexión móvil, de forma que en el momento de acordar una venta o alquiler, la información esté disponible para todos los empleados en tiempo real.*

*Víctor propone sencillamente un sistema de bloqueo, de forma que antes de salir, el empleado bloquee los pisos que va a enseñar, para que ningún otro compañero los ofrezca a otros clientes hasta que él vuelva a la oficina y los desbloquee. Siempre además, se podrá contactar por teléfono para ponerse de acuerdo entre distintos comerciales en caso de necesidad. Es más simple, y puede resultar prácticamente igual de efectivo con un coste realmente bajo, casi nulo.*

¿Qué hace el software?

En este punto vamos a profundizar en el análisis necesario para contestar a esa pregunta.

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**) y
  - 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 e
  - 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 el 30% del esfuerzo.

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

1. Introducción
  1. Objetivo
  2. Ámbito
  3. Definiciones, siglas y abreviaturas
  4. Referencias
  5. Visión global
2. Descripción general
  1. Perspectiva del producto
  2. Funciones del producto
  3. Características del usuario
  4. Limitaciones generales
  5. Supuestos y dependencias

### 3. Requisitos específicos

#### 1. Requisitos funcionales

##### 1. Requisito funcional 1

1. Introducción
2. Entradas
3. Procedimiento
4. Salidas

##### 2. Requisito funcional 2

##### 3. ....

#### 2. Requisito de interfaz externa

1. Interfaces de usuario
2. Interfaces hardware
3. Interfaces software
4. Interfaces de comunicaciones

#### 3. Requisitos de ejecución

#### 4. Restricciones de diseño

1. Acatamiento de estándares
2. Limitaciones hardware

#### 5. Atributos de calidad

1. Seguridad
2. Mantenimiento

#### 6. Otros requisitos

1. Base de datos
2. Operaciones
3. Adaptación de situación

#### 4. Apéndices

#### 5. Índice

***Para saber más.***

***Para conocer herramientas para el análisis de requerimientos:***

***[Herramientas Microsoft para el análisis de requisitos](#) [Versión en caché]***

***Para conocer más sobre el análisis de requerimientos:***

***[Manual de desarrollo de software de la wikilearning](#) [Versión en caché]***

## 5. Búsqueda y descripción de requisitos funcionales

### Búsqueda y descripción de requisitos funcionales

***CASO.***

*Durante el análisis de la aplicación para la constructora RoKemar, para la fase de descripción de requisitos funcionales, **María** se plantea qué técnica será la más adecuada. Ella normalmente suele usar la técnica de las **entrevistas**, como la mayoría de los analistas que conoce, pero sabe de los buenos resultados que ha dado la técnica de **Desarrollo Conjunto de Aplicaciones (JAD)**, y que algunas de las grandes empresas que la han empleado hablan de importantes ahorros de tiempo en el desarrollo de software y de una mayor satisfacción de los usuarios con los sistemas construidos. Puesto que la **satisfacción de los usuarios** es uno de los motivos que ha decidido al gerente de **RoKemar** a*

*continuar con el proyecto de implantación de una nueva aplicación informática, **María** piensa que puede ser una estupenda oportunidad de poner en práctica también por su parte esta técnica, y comprobar los resultados.*

***Carmen** le comenta a **María** que no tiene del todo claro cómo se desarrolla la técnica JAD. Sabe que es con sesiones de trabajo en las que intervienen tanto los analistas como los usuarios del sistema, pero poco más. **María** le explica en qué consisten, y **Víctor** presta atención mientras tanto. Al finalizar la exposición de **María**, **Víctor** le pregunta:*

*"¿Piensas que los empleados de **RoKemar** van a tener tiempo de asistir a varias reuniones de este tipo, que requieren bastante tiempo, sin desatender su trabajo? "*

*"Ése es justamente uno de los inconvenientes que hacen difícil de aplicar esta técnica en algunos casos. Tendremos que preguntarles y valorar si es posible." - le contesta **María**.*

***Carmen** interviene: "También podemos plantearnos hacerles un **prototipo** rápido de las interfaces de usuario, y así tendrían más claro lo que va a permitir hacer el sistema, ayudándonos a detectar necesidades y a afinar más en los requisitos que debe cumplir la aplicación".*

*A **María** la idea de **Carmen** le parece bien, y decide que si no fuera adecuado el uso de la técnica JAD, tendrán en cuenta la posibilidad de realizar un prototipo de la aplicación, tal y como propone **Carmen**.*

¿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.

[Aquí tienes un ejemplo de cuestionario de una aplicación real, para que te hagas una idea de su formato.](#)

- **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. En los subapartados siguientes vamos a verlas más detalladamente.

*Para saber más.*

*Para conocer más sobre técnicas de recogida de información:*

[Técnicas de recogida de información](#) [\[Versión en caché\]](#)

## 6. Entrevistas

---

### 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 topdown, 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:
  - El entrevistador hable un 20% del tiempo, mientras que el entrevistado sea el que ocupe el 80% restante.
  - Se debe **evitar los monólogos** y conseguir que constituyan 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 y
    - 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 posible 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 pregunta cerradas.

### Ejemplos de las preguntas con respuestas abiertas y cerradas en la entrevista estructurada.

#### Forma de pregunta abierta

Ejemplo: obtener la información sobre las características de diseños críticos 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?.

#### Forma de pregunta cerrada

Ejemplo: obtener la información sobre las características de diseño críticos 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 en 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.

*Para saber más.*

*Más sobre técnicas de recogida de información:*

[Técnicas de recogida de información](#) [Versión en caché]

## 7. Desarrollo conjunto de aplicaciones (JAD)

---

### 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** con duración 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 (proyectores, 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. .

*Para saber más.*

*Artículo sobre el desarrollo conjunto de aplicaciones:*

[Desarrollo conjunto de aplicaciones \(JAD\)](#) [Versión en caché]

## 8. El prototipado

---

### 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 desea 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](#) [Versión en caché]

## 9. Especificación de soluciones técnicas

### Especificación de soluciones técnicas

*CASO.*

*Cuando **María** ha comenzado a realizar la especificación de las distintas soluciones técnicas que proponen, a **Víctor** le*

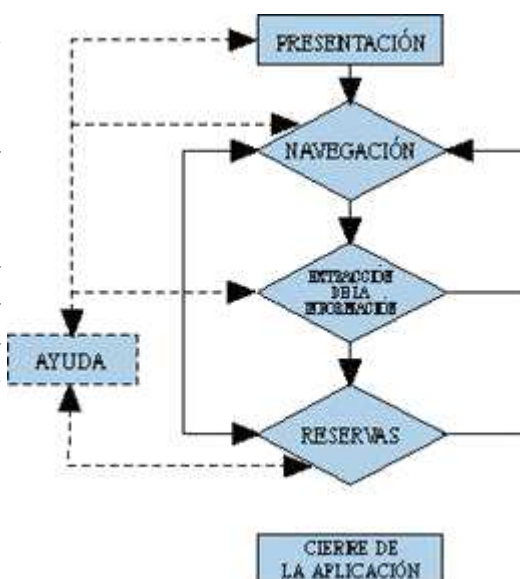
ha llamado la atención que emplee los **diagramas de flujo u organigramas**, que es algo que recuerda haber visto en alguna otra ocasión. **Carmen** le comenta que son una herramienta de Análisis, que se emplean para describir los sistemas y los algoritmos que se diseñan como solución del sistema de una forma gráfica, bastante fácil de entender.

### ¿Pero cómo representamos las soluciones que hemos diseñado?

Para la especificación de las distintas configuraciones alternativas del hardware es necesaria la realización de un **modelo**.





Los **organigramas** en sistemas de información están orientados a la máquina para representar unidades físicas. Se basa en la utilización de unos símbolos gráficos que denominamos cajas, en las que escribimos las acciones que tiene que realizar el software. Deben ser leídos de arriba abajo, y de izquierda a derecha. Tienen la ventaja de proporcionar una **representación muy visual**, que permite entender de un solo golpe de vista la estructura, el tipo de acciones que se realizan y el orden en que se realizan.

A continuación te presentamos un cuadro de la clasificación de los símbolos utilizados en los organigramas.







Símbolos de soporte de información			
Teclado	Pantalla	Impresora	Tarjeta perforada
Cinta de papel	Disco magnético	Cinta magnética	

Símbolos de proceso			
Manipulación de uno o varios ficheros (Intercalación)	Clasificación u ordenación de datos en un fichero	Fusión o mezcla de dos o más ficheros en uno solo	Partición o extracción de datos de un fichero
Proceso	Terminador	Operación E/S	Proceso predefinido

Símbolos de decisión		Líneas de flujo	
Decisión	Bucle	Flechas	Línea conectora
			

Símbolos de conexión			Símbolos infor.
Conector	Conector misma página	Conector distintas páginas	Comentarios
			

**DEMO:** Aquí tienes una aplicación que de forma interactiva va a permitirte consultar rápidamente cada uno de los símbolos utilizados en los diagramas de flujo.

*Para saber más.*

*Enlaces en el que podamos descargar alguna aplicación gratuita para hacer organigramas. Puedes encontrar varias opciones para Windows.*

*Editor e intérprete de diagramas de flujo con una interfaz gráfica que facilita en gran medida la creación de diagramas de flujo para la representación de algoritmos.*

[DFD 1.0](#)

*Versátil sistema de dibujo para crear diagramas de flujo, diagramas organizativos, y cualquier tipo de diagrama por bloques similar.*

[EDGE Diagrammer](#)

*Diagram Designer es un sencillísimo editor de gráficos vectoriales con el que puedes crear diagramas, esquemas, planos y croquis de todo tipo.*

[Diagram designer](#)

*Dia es una aplicación totalmente gratuita, distribuida bajo licencia GNU General Public License, que permite crear cómoda y rápidamente interesantes diagramas de flujo.*

[Dia Win32](#)

*Para entornos Linux tienes otras opciones, entre las que destacamos el enlace a DIA, un generador de diagramas de flujo muy útil. En esta página puedes encontrar una breve descripción y algunos enlaces interesantes:*

[Diagramas de Flujo en Linux](#)

## 10. Análisis de viabilidad técnica y económica

### Análisis de viabilidad técnica y económica

### CASO.

*Víctor comenta que si una empresa contacta con SI Andalucía, como empresa de servicios de informática, para hacerles una aplicación, **no entiende la necesidad de hacer un estudio de viabilidad**. Ellos son informáticos, y viven de hacer aplicaciones, por lo que piensa que lo que tienen que hacer es convencer al cliente siempre de que esa aplicación va a ser beneficiosa.*

*José le explica que nada más lejos de la realidad. Si la aplicación que hacen incumple la ley, o le hace perder dinero al cliente, o difícilmente le permite amortizar la inversión, o tiene dificultades técnicas que no van a poder ser capaces de solucionar en un tiempo y a un coste razonables, lo más sensato será decírselo así al cliente. De lo contrario, posiblemente perderían un cliente, o incluso futuros clientes potenciales que se enteraran de su forma de actuar, y perderían el prestigio de empresa seria y responsable que tanto esfuerzo les ha costado hacerse.*

*- "No olvides que nuestros clientes nos pagan para solucionarles los problemas, no para crearles problemas nuevos".- Le dice José a Víctor.*

**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 y
- 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** para varias alternativas suficientemente distintas, 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.

## 11. Análisis de coste y beneficios

---

### Análisis de coste y beneficios

---

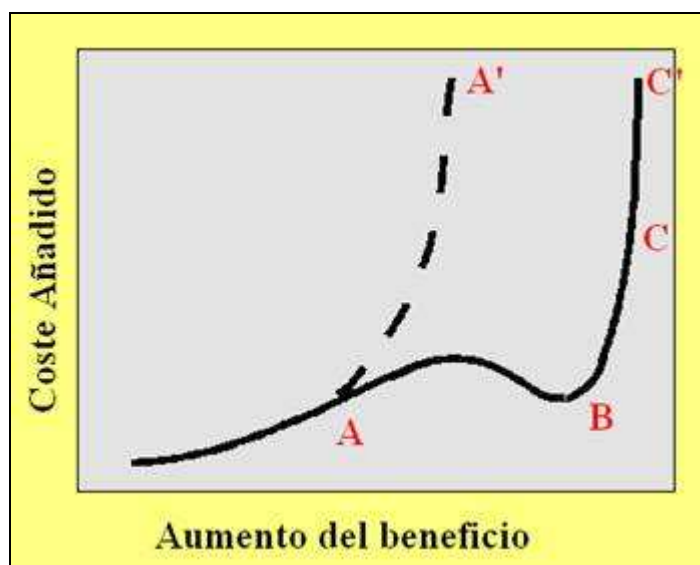
Entre la **información** más relevante que contiene el estudio de viabilidad se encuentra el **análisis de coste-beneficio**, que es una **evaluación de la justificación económica** para un proyecto de sistema basado en ordenador.

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.

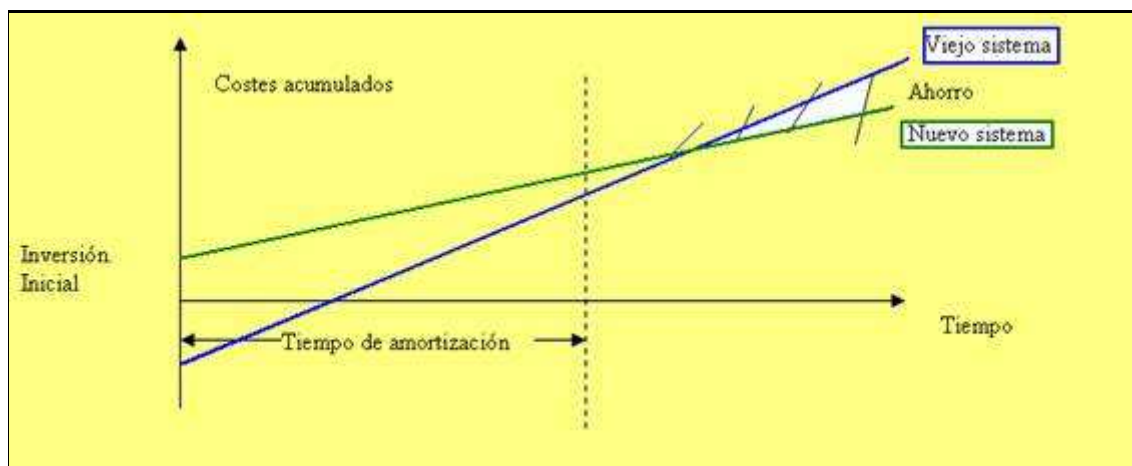
**Sólo gastando el tiempo necesario para evaluar la viabilidad, reducimos las oportunidades de situaciones extremadamente embarazosas en etapas posteriores de un proyecto de un sistema. El esfuerzo gastado en el análisis de viabilidad, aunque termine aconsejando la cancelación de un proyecto propuesto, 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.



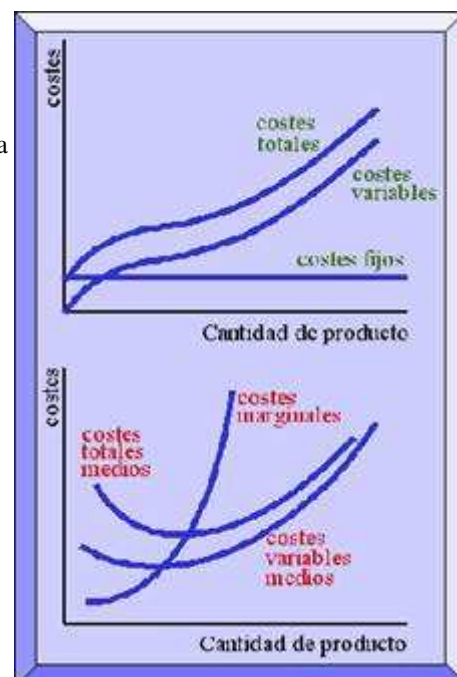
Posibles costes de un sistema de información:

### ■ 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)

Vamos a verlo con un ejemplo.

### [Aquí tienes un ejemplo realizado sobre una empresa en Diciembre de 1999](#)

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".

*Para saber más.*

*Aquí tienes una de tantas Metodología para la estimación de costes-beneficios en un proyecto informático:*

*Metodología de proyectos informáticos*

[Metodología de proyectos informáticos](#) [\[Versión en caché\]](#)

## 12. Casos de uso

### Casos de uso

Llegados a este punto de la unidad seguro que estarás pensando que hemos estado viendo todas las partes que componen un análisis previo, pero no hemos visto ninguno totalmente realizado. Pues bien ha llegado el momento de verlo, y como muestra qué mejor ejemplo que el realizado por alumnos que cursaron este mismo módulo del ciclo.

Te proponemos el siguiente índice de contenidos para un análisis previo:

1. Objetivos del sistema
2. Entorno de desarrollo
3. Resumen de estado de viabilidad
  - Económica
  - Técnica
  - Legal
  - Operativa
  - Social
4. Recursos requeridos. Coste. Tiempo (Aplicar algún método de los que viste en la unidad anterior Cocomo, Slim, ptos. de funcion, etc.)
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

[Ejemplo 1: Inmobiliaria.](#)

[Ejemplo 2: Tienda de bicicletas.](#)