

- Las bases de datos OLAP se suelen alimentar de información procedente de los sistemas operacionales existentes, mediante un proceso de extracción, transformación y carga (ETL).

En la base de cualquier sistema OLAP se encuentra el concepto de cubo OLAP (también llamado cubo multidimensional o hipercubo). Se compone de hechos numéricos llamados medidas que se clasifican por dimensiones. El cubo de metadatos es típicamente creado a partir de un esquema en estrella o copo de nieve, esquema de las tablas en una base de datos relacional. Las medidas se obtienen de los registros de una tabla de hechos y las dimensiones se derivan de la dimensión de los cuadros.

Los sistemas OLAP se clasifican en:

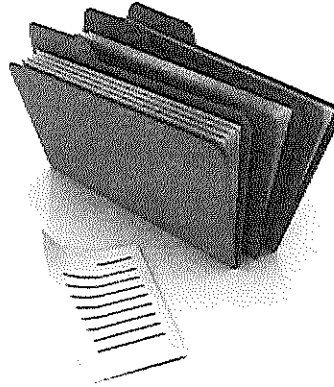
- **ROLAP:** es una implementación OLAP que almacena los datos en un motor relacional. Los datos son detallados para evitar las agregaciones y las tablas se encuentran desnormalizadas. Suelen trabajar sobre esquemas estrella o copo de nieve, aunque es posible trabajar sobre cualquier base de datos relacional.
- **MOLAP:** almacena datos en una base de datos multidimensional. Para de este optimizar los tiempos de respuesta, el resumen de la información es usualmente calculado por adelantado.
- **HOLAP:** almacena datos en un motor relacional y en otros en una base de datos multidimensional.

Estas herramientas permiten presentar al usuario una visión multidimensional de los datos para cada actividad que es objeto de análisis. El usuario formula consultas a la herramienta OLAP seleccionado los atributos de este esquema multidimensional sin conocer la estructura interna del almacén de datos, es decir sin conocer el esquema físico del almacén de datos.

La herramienta OLAP genera la correspondiente consulta y la envía al gestor de consultas del sistema.

Una consulta a un almacén de datos consiste en la obtención de medidas sobre los hechos parametrizados por atributos de las dimensiones y restringidas por condiciones impuestas sobre las dimensiones.

### 1.1.2. Estructuración de la información para adecuarse a las necesidades de la empresa



La **información** es un conjunto de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje.

Cualquier tipo de material que posea las características de un documento, cuenta tanto con un contexto externo como interno.

En el ambiente de la recuperación de información todo tipo de documentación tiene unos parámetros que permiten ubicarlos de un modo fácil y rápido y esto es lo que se conoce como **palabras claves**, las cuales son identificadas en la búsqueda por los buscadores y toman como referencia para llegar a la información.

Los **descriptores** son los encargados de sintetizar la información para así optimizar la localización. Los **listados de encabezamiento de materia** son una frase que contiene el tema que se quiere referenciar; los **términos** o las palabras que expresan una idea de una disciplina determinada, son los que pueden ser términos compuestos conformados por una unión de letras y además cuentan con un significado.

Los **descriptores libres** o **no descriptores**, están formados por palabras no relevantes sobre el tema y que no sintetizan la información para su posterior ubicación. Finalmente las **etiquetas**, son un conjunto de caracteres útiles en la identificación de los datos o instrucciones.

En el proceso de extracción de datos es importante tener en cuenta la tipología genérica de los documentos. La extracción de datos se suele utilizar para

el relleno automático de metadatos o como entrada para otro tipo de software empresarial.

A continuación se describen los tipos de documentos generales.

### **Documentos estructurados**

Son documentos en los que se sabe que información se va a encontrar y la posición que esta información ocupa dentro de las dimensiones físicas del documento.

En este tipo de documentos es relativamente sencillo encontrar y extraer datos, pues podemos saber dónde buscarlos.

### **Documentos semiestructurados**

La dificultad de gestión va creciendo conforme a la información se vuelve estructurada. Los documentos semiestructurados, son aquellos en los que se sabe que información se va a encontrar, pero no se sabe exactamente en dónde se encuentra.

La forma en que se extraen datos de documentos semiestructurados no puede ser la misma que en el caso anterior, aquí es necesario enseñarle al software cómo es lo que se está buscando.

### **Documentos desestructurados**

En este tipo de documentos ni se sabe qué vamos a encontrar, ni donde se encuentra.

La dificultad de extracción de datos de este tipo de documentos es máxima. En este grupo se incluye los informes y las cartas. Según algunos autores este grupo de información no existe.

Según algunos autores estos documentos cumplen generalmente tres características:

- La estructura del documento no ha sido diseñada por la empresa que ahora quiere gestionarlos.

- La estructura de este documento puede variar dependiendo de quien la envía.
- No puede ser procesados ciñéndose a un template o plantilla.

### **Estructura lógica y estructura física**

En el intento de estructuración de la información y los datos es imprescindible tener claras las ideas básicas sobre los mecanismos que se encuentran involucrados.

- **Estructura lógica:** se relaciona con la idea inicial de los programados acerca de cómo están organizados los diferentes datos, y coincide aproximadamente con la forma en que son manipulados los datos por el programador de alto nivel.
- **Estructura física:** se relaciona con la forma en que están contenidos los datos en la máquina, de la que existen dos versiones:
  - Una corresponde a la que adoptan los datos en memoria.
  - Otra correspondiente a su almacenamiento externo.

Es evidente que ambas estructuras no se corresponden, la **estructura física** de datos en los almacenamientos externos no se corresponden de forma exacta con la estructura lógica. En primer lugar, el documento de clientes antes mencionado, puede estar representado físicamente por varios ficheros que pueden ser multi-volumen. Es decir, que ocupan más de un volumen lógico en la máquina que los contiene. Si son aplicaciones de red, pueden estar incluso en máquinas remotas, distintas de la que ejecuta la aplicación. Además, aunque nos figuremos la estructura lógica es un todo continuo, sabemos que la estructura física correspondiente, incluso si se tratase de un solo fichero, está compuesta por trozos que pueden estar dispersos en el disco.

La **estructura lógica** está ordenada por números o bien por nombres. En cambio la estructura física puede estar construida simplemente por el orden natural, es decir, por el orden de creación de los propios registros. Normalmente, la apariencia de ordenación es el resultado de un proceso complejo que utiliza índices, tablas y punteros, para proporcionar un acceso ordenado a una estructura mucho más caótica.

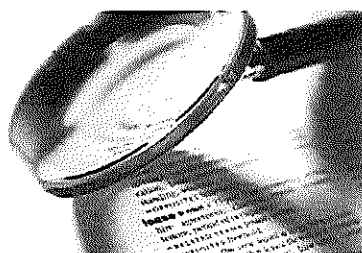
El programador con ayuda del programa se encarga de manejar los datos en términos de estructura lógica. Con respecto a ello, las herramientas que proporcionen el lenguaje o entorno de programación, deberán ser de mayor nivel cuanto mayor sea la distancia con que pueda ser manejada la estructura lógica de datos respecto de su verdadera estructura física. Precisamente el manejo de estas estructura a dado lugar a la distancia con que pueda ser manejada la estructura lógica de datos respecto de su verdadera estructura física.

De este modo el manejo de dichas estructuras ha dado lugar a toda una rama de la industria del software que ha logrado alcanzar un elevado nivel de sofisticación y especialización.

## 1.2. Creación de extractores de datos

Un **extractor de datos** es un conjunto de motores de extracción que permite obtener mediante la extracción los datos de las bases de datos y utilizar programas de visualización.

Es decir permite la extracción automática de entidades o conceptos del contenido de texto visible de un elemento y asignarlos a una propiedad administrada. Al mismo tiempo, estas propiedades se pueden usar para restringir las consultas mediante filtros de propiedades o como opciones de refinamiento de consultas.



Existen dos tipos de extractores:

- **Extractores de propiedades de coincidencia exactas:** son adecuados para establecer coincidencias de cadenas en todos los idiomas. Las entradas en el diccionario personalizado pueden ser palabras individuales o una cadena de palabras. La búsqueda de cadenas coincidentes se realiza después de una tokenización básica, que reemplaza los caracteres separados presentes en el texto por

espacios en blanco. Tras la tokenización básica, los extractores deben buscar una coincidencia exacta para la cadena.

- **Extractores de propiedades de coincidencia parcial:** son adecuados para establecer coincidencias de cadenas en todos los documentos del este asiático ya que las palabras de estos idiomas no están separados por espacios. También pueden ser usado en aquellos casos específicos en los que se necesitan coincidencias de subcadenas.

En ambos casos es importante tener en cuenta que el establecimiento de coincidencias se distingue mayúsculas y minúsculas.

A continuación se explicará los aspectos claves en el proceso de creación de un extractor de datos:

- Obtención de información de fuentes internas o externas.
- Agrupación, transformación y homogeneización de la información para su posterior estudio.

### 1.2.1. Obtención de información de fuentes internas o externas

El proceso de obtención de la información comienza por la selección de las fuentes utilizables, es decir, los lugares en los que se obtiene la información útil para poder realizar las diferentes tareas con ella.

Se puede hablar de dos tipos de fuentes:

- **Internas:** propias de la empresa.
- **Externas:** ajenas a la empresa.

Al mismo tiempo, dentro de ambos tipos de información se distingue:

- **Fuentes primarias:** son aquellas que se adquieren mediante los procesos de investigación de forma directa, ya sea de la propia empresa o por encargo a empresas dedicadas a la elaboración de este tipo de información.
- **Fuentes secundarias:** son aquellos datos que proceden de estadísticas y documentos ya publicados, proporcionan información

de tipo general ya elaborado. Es recomendable empezar la investigación acudiendo a ellas. Son fuentes secundarias externas las publicaciones de organismos oficiales, bancos, etc.

Dentro de ambos tipos de información existen amplios grupos de datos, que dan lugar a la clasificación del tipo de información dentro de las fuentes de información primaria y secundaria.

### **Información cuantitativa**

Es la que se extrae mediante métodos diversos a partir de una muestra representativa de la población para proyectar los resultados y conclusiones a toda la población.

- **La encuesta:** ofrece información abundante si se cuenta con un buen cuestionario. Un cuestionario es un documento que recoge una serie de preguntas formuladas con claridad, de fácil comprensión, sin implicación de respuestas y colocadas con un orden lógico. Además se debe determinar la población que se quiere estudiar y la muestra correcta. Hay diferentes tipos de encuestas: personal, postal, telefónica, en Internet.
  - **La encuesta Ómnibus:** consiste en una entrevista personal con varios apartados dentro del cuestionario sobre diferentes temas o productos.
  - **Método Delphi:** es una variante específica de la encuesta Ómnibu, que está basada en entrevistas a expertos y consultores que elaboran informes independientes entre sí acerca de las cuestiones investigadas.
- **El panel:** es una encuesta periódica que se realiza a las mismas personas, es decir, la muestra es permanente. Permite identificar cambios y la evolución en el tiempo de las variables investigadas. Se utiliza para medir audiencias de televisión, en otros usos.
- **La observación:** consiste en observar la conducta de los diferentes consumidores y extraer conclusiones a partir de ellas. Las personas son totalmente libres para comportarse y reaccionar de forma espontánea. La observación puede ser directa o indirecta, a través de cámaras.

- **La experimentación o prueba de mercado simulada:** la observación se hace en un escenario preparado, donde se busca la participación interactiva de las personas que componen la muestra. Se pretende reproducir a escala reducida situaciones reales para poder prever los resultados, problemas, ventajas, inconvenientes, etc.

### Información cualitativa

Emplea métodos adecuados para investigar o buscar necesidades, hábitos de consumo, etc.

- **Encuesta en profundidad:** es una entrevista abierta realizada por un profesional. Supone establecer un diálogo con un individuo para conocer sus motivaciones, gustos, personalidades, actitudes, etc.
- **Reunión de grupo:** se trata de una reunión activa de un grupo de entre seis a diez consumidores con un moderador para hablar sobre un producto.
- **La pseudocompra:** el investigador se pone en el lugar de posibles compradores o clientes.
- **Técnicas proyectivas:** intenta conocer los impulsos psicológicos que subyacen detrás del comportamiento del consumidor estudiando su reacción ante determinados estímulos externos, por ejemplo ante frases, imágenes, marcas, colores, etc. es necesaria la observación directa del cliente, o consumidor, en el lugar de compra o el uso de medios como las tarjetas de fidelización con las que se puede analizar cómo reaccionan los consumidores o sus políticas comerciales.





La obtención de la información, de los datos, requiere del cumplimiento de diferentes aspectos que configuran el proceso completo.

- **Planteamiento de una necesidad:** la falta de información puede ser un problema, y por tanto es necesario dar solución mediante la obtención de los datos requeridos.
- **Determinación de los objetivos a alcanzar:** determinar qué se quiere conseguir, que resultados son los esperados y cómo se van a emplear.
- **Fijar el contenido del estudio:** dependiendo de los objetivos que se quieran conseguir, se definen en detalle el contenido de la información que se necesita.
- **Búsqueda de información:** se obtiene la información a partir de las fuentes primarias y secundarias.
- **Planificación de la intervención:**
  - Determinación de las fuentes secundarias de obtención de la información.
  - Determinar las fuentes primarias, técnicas a emplear, selección de la muestra y elaboración del cuestionario.
  - Estimación y planificación del tiempo necesario.
  - Cuantificación del coste de la investigación.
  - Señalar las personas encargadas de la investigación y asignación de responsabilidad.
- **Obtención de la información:** la información es adquirida en base a todos los aspectos anteriormente señalados.
- **Interpretación de resultados:** se le da sentido a la información obtenida, para poder dar respuesta a la necesidad inicial que puso en marcha todo este proceso.

La obtención de información útil, óptima y adecuada, en un momento determinado, es uno de los recursos más importantes para una empresa. El proceso comienza con la captación de datos de la realidad en la que desenvuelven, continúa con su procesamiento y culmina con la emisión de informes que les permiten tomar decisiones con una menor dosis de incertidumbre.

La obtención de los datos comprende varios procesos:

- **Procesamiento de los datos:** consiste en la actividad de clasificación, registro, cálculo y almacenamiento de datos, estableciendo relaciones entre ellos de forma tal que brinden información.
- **Obtención de la información:** es decir los informes de salida del sistema de información. Los informes presentan la información de manera ordenada para que los datos procesados puedan ser interpretados para poder sacar conclusiones válidas.
- **Sistemas de información:** se entiende por sistema al conjunto de elementos que se relacionan entre sí para poder aportar y contribuir a la consecución de un objetivo concreto. Uno de los sistemas más importantes, es el de organización de la información. Se compone de los siguientes elementos:
  - Datos que ingresan.
  - Un proceso de elaboración.
  - Combinación.
  - Modificación.
  - Información emitida de diferentes formatos.

### 1.2.2. Agrupación, transformación y homogeneización de la información para su posterior estudio

Según la definición de la Real Academia Española, los siguientes términos deben entenderse del siguiente modo:

- **Agrupar:** es reunir en grupo o apiñar, constituyendo una agrupación.
- **Transformar:** hacer mutar algo en otra cosa.
- **Homogeneizar:** hacer homogéneo, por medios físicos o químicos, un compuesto o mezcla de elementos diversos.

En base a dichas definiciones nos adentraremos en la utilización de dichos términos para la comprensión del proceso de tratamiento de información para su posterior estudio.



Los contenidos se pueden **agrupar** de diferentes modos:

- Arriba a bajo.
- Abajo arriba.
- Orden cronológico.
- Geográfico.
- Jerárquico.
- Por tareas.

Pero siempre con el objetivo de responder a las necesidades que ha dado lugar la necesidad de agrupar la información.

La información frecuentemente se compone de diferentes contenidos que pueden ser desarrollados en diversos formatos:

- Texto.
- Video.
- Audio.
- Fotografías.

Cuando estos elementos forman parte de un listado que no tiene ningún orden, el usuario que va a trabajar con ellos se verá obligado a recorrer toda la lista, en la que todos los elementos pueden aparecer mezclados.

Si sólo tenemos cuatro contenidos quizás pueda funcionar dicha forma de buscar los datos, pero sin duda puede que el usuario los encuentre por casualidad, y por tanto esta opción no parece ser la más operativa, sobre todo si se trata de decenas, cientos o millones de contenidos.

Surge de este modo la necesidad de organizar la información de algún modo. Lo más adecuado sería diseñar una forma de agrupación de contenidos con la que el mayor porcentaje de usuarios pueda encontrar lo que necesita lo antes posible y con el menor esfuerzo posible, dependiendo sus necesidades concretas en cuanto a la búsqueda de información.

Hay bastantes formas de llevar a cabo la agrupación de la información:

- **Agrupaciones exactas:** un contenido sólo podrá pertenecer a una categoría y no a otra.
  - Por orden cronológico.
  - Por ubicación geográfica.
- **Agrupaciones ambiguas:** un contenido podría encontrarse en más de una categoría. Qué esté en una u otra va a depender de diversos factores.
  - Por jerarquías.
  - Por facetas.
  - Por tareas.
  - Por audiencias.
  - Por prominencia. Contenidos que son destacados.
  - Híbridos. Ocurren cuando se mezclan varios tipos que no se deberían usar en un mismo espacio ya que pueden desorientar al usuario.

La creación de la agrupación o clasificación de los diferentes contenidos, se puede llevar a cabo mediante dos estrategias fundamentalmente:

- **De arriba abajo:** también conocida como tip-down. Primero se deciden las categorías principales, y después, cada una se va desglosando en subcategorías.
- **De abajo arriba:** también se conoce como bottom-up. Primero se listan las subcategorías o temáticas que se van a tratar o para las cuales se tiene información y después se van agrupando en función de los objetivos establecidos, y de cómo se considere que será el modelo mental más probable que utilice el usuario, es decir como creamos que el usuario realizará la búsqueda.

La **transformación**, es también conocida como conversión, y es el proceso de transformación de datos informáticos de una representación concreta a otra, cambiando los bits de un formato a otro, normalmente para lograr la interoperabilidad de aplicaciones o sistemas diferentes. A nivel más simple, la conversión de datos puede ejemplificarse por la conversión de un fichero de texto

desde una codificación de caracteres a otra. Son conversiones más complejas las de formatos de ficheros ofimáticas y multimedia, a veces fuera de las capacidades de ordenadores domésticos.

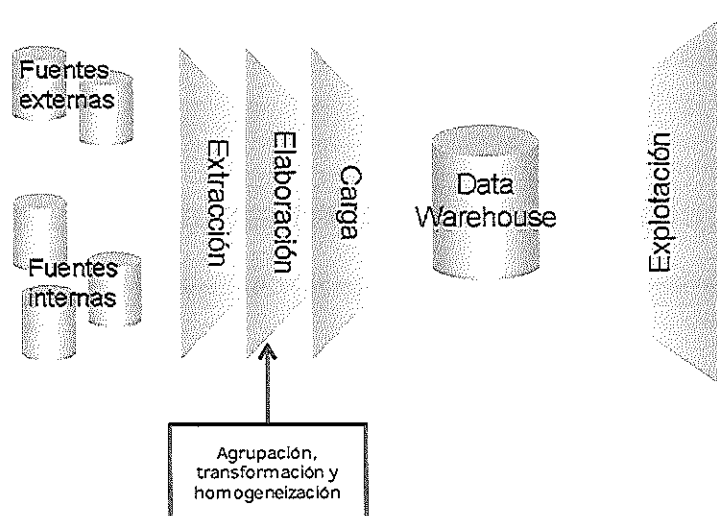
Antes de que pueda efectuarse cualquier tipo de transformación de datos el programador o usuario que sea el encargado de ello debe tener en cuenta unos cuantos conceptos básicos:

- Es fácil descartar información usando un ordenador, pero añadirla requiere esfuerzo.
- El ordenador puede usarse para añadir información sólo con base en reglas; la mayoría de adiciones que interesa a los usuarios sólo puede lograrse con la ayuda de humanos.
- Sobremuestrear los datos o convertirlos en un formato con más posibilidades no añade información: sólo hace hueco para dicha adición, que suele tener que hacer un humano.

El proceso de **homogeneización** es aplicable a diferentes campos, en este contexto hace referencia al proceso mediante el que se produce la mezcla de información, mediante una regla general, para obtener una mejor calidad en los resultados obtenidos del trabajo con la información.

En el proceso de homogeneización se realizan los diferentes ajustes necesarios para poder realizar la comparación y agregación de los diferentes documentos, información, o datos concretos, de un modo adecuado.

Para trabajar con la información es muy importante garantizar que los datos de que se disponen poseen un control adecuado y homogéneo.



Estos tres tipos de procesos tienen lugar en la etapa de elaboración de la información.

En la etapa de extracción se produce la obtención de la información de las fuentes internas y externas como se ha explicado en temas anteriores.

En la fase de elaboración es donde se produce los procesos de agrupación, transformación y homogeneización de la información.

En la fase de carga se organiza y actualizan los datos.

Y en la última fase se produce la extracción y el análisis de la información en los diferentes niveles de agrupación.

Es importante conocer, y diferenciar los siguientes términos:

- **Cubos OLAP (On-Line Analytical Processing):** Estructura cúbica de ordenamiento de los datos con fin de hacer un análisis n dimensional sobre la base de datos.
- **ETL (Extraction, Transformation Load):** Proceso de extracción transformación y carga de datos requerido para alimentar un Data Warehouse.
- **Metadata (datos sobre los datos):** Los metadatos permiten mantener información de la procedencia de la información, la periodicidad de refresco, su fiabilidad, forma de cálculo, etc., relativa a los datos de nuestro almacén.
- **Data Mart:** Base de datos construida para un fin específico. Por lo general, podemos decir que un Data Mart es un subconjunto del Data Warehouse.
- **ODS (operacional data store):** bases de datos operacionales.

## 2. Extracciones de informaciones contenidas en sistemas ERP-CRM, procesamiento de datos

La **recogida de datos** realizada por la solución suele ocurrir en nivel interno, aprovechando buena parte de los datos que son dispuestos en formato

legible por el ordenador generados por medio de las actividades desarrolladas en distintas áreas y unidades de negocio de la empresa. Pero muchos datos externos son convertidos al formato legible por el ordenador por medio de empleados que registran la entrada de datos, como por ejemplo las informaciones registradas por un operador de telemarketing. Los datos también son ingresados en el sistema por las interacciones iniciadas por los clientes, como es el caso de una visita al sitio Web de la empresa por medio del n° de usuario y contra seña, y el software ayuda en esa recogida en los varios canales de comunicación manejados por las empresas.

El sistema central de almacenamiento de datos del CRM está compuesto por los datos almacenados, un mecanismo que los mantiene actualizados y otro mecanismo que gestiona y controla los datos. Ese sistema tiene por objetivo garantizar que sean generadas bases de datos de todos los sectores de la empresa, y que esos datos estén ordenados para que sean analizables y aprovechados según el nivel de acceso designado a cada persona de la empresa, desde cualquier punto de acceso.

El sistema de **análisis y entrega de información** esta basado en las conexiones lógicas, análisis descriptivo y cruces de los datos solicitados por los usuarios, según su puesto en la empresa y nivel de acceso a las bases de datos en su estación de trabajo. En ese proceso, la solución dispone de mecanismos que auxilian en la extracción de información de los datos, como por ejemplo, generación de informes, consultas a datos, gráficos con algunos datos estadísticos descriptivos, y también tablas interrelacionados con los respectivos cruces de datos que permite al usuario hacer su búsqueda en cadena, en creciente grado de complejidad y hacer un análisis en nivel multidimensional.

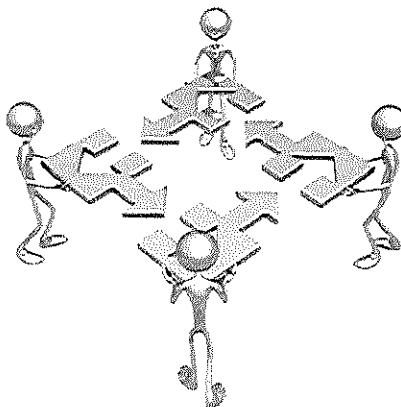
El CRM puede poseer una herramienta de "data mining" para realizar la minería de datos, en esa minería el software auxilia al usuario con informaciones no buscadas por el usuario y ayúdale a verificar eventuales hipótesis. También puede llevar una herramienta de análisis geográfico de marketing, el "geomarketing" que auxilia la empresa principalmente en la segmentación geográfica de sus clientes.

De forma más práctica y con enfoque más aplicado a la estrategia el CRM ayuda a detectar los mejores clientes actuales y los potenciales, posibilita la captación y evaluación del comportamiento y desempeño de los servicios y productos de la empresa, permite a la empresa crear y efectivamente usar nuevos canales de comunicación, ayuda a desarrollar nuevos modelos para el establecimiento de precios y también captar el nivel de retención, satisfacción de

los clientes de la empresa (Rigby, Reichheld y Schefter, 2002). Esta parte del CRM permite explorar los datos del cliente para responder cuestiones del tipo "¿que voy a ofrecer para mi cliente en seguida?" "¿cual es la probabilidad de mi cliente dar de baja de mis servicios como puedo segmentar mis clientes?" (Ang y Buttle, 2006) y también "¿como puede mi empresa comunicar con mis clientes?" o "¿cuales son las preferencias de mis clientes en color y tamaño?" (Chen y Popovich, 2003).

## Los gestores de datos

Los **gestores de datos** son un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información de una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos.



Hay diferentes tipos de gestores de datos en función del manejo que se hacen de ellos, es decir según funcionen en ordenadores personales con poca memoria o grandes sistemas que funcionan con sistemas de almacenamiento especiales.

Es importante tener claro los diferentes tipos de gestores, por lo que a continuación se detallarán estas diferencias, así como:

- Modelos lógicos.
- Modelos conceptuales.

Cada sistema gestor de datos puede utilizar un modelo diferente para los datos. Por lo que hay modelos conceptuales diferentes según el sistema gestor de bases de datos que se utilicen.



La diferencia entre los distintos sistemas de gestión de datos está en que proporcionan diferentes modelos lógicos.

Las principales diferencias entre los modelos lógicos y conceptuales son las siguientes:

- El **modelo conceptual** es independiente del sistema gestor de base de datos que se vaya a utilizar. El lógico depende de un tipo de sistema de gestión de datos en particular.
- El **modelo lógico** está más cerca del modelo físico, el que utiliza internamente el ordenador.
- El **modelo conceptual** es el más cercano al usuario, el lógico es el encargado de establecer el paso entre el modelo conceptual y el modelo físico del sistema.

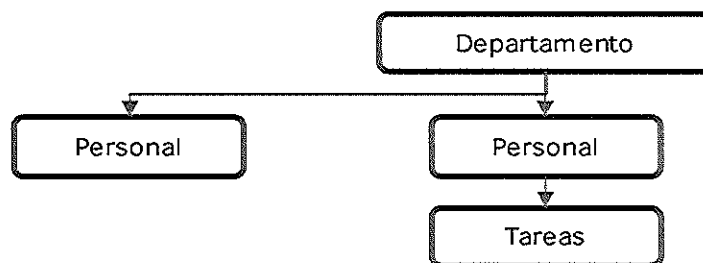
## Modelos lógicos

### *Modelo jerárquico*

Es utilizado por los primeros gestores de datos. Es también conocido como modelo de árbol ya que se utiliza una estructura en árbol para organizar los datos.

La información es organizada en orden jerárquico, en el que la relación entre las entidades siempre es del tipo padre/hijo. De este modo hay una serie de nodos que contienen atributos y que se relacionarán con nodos hijos de tal forma que puede haber más de un hijo para el mismo padre, pero solo un padre para un hijo.

En este modelo los datos se almacenan en estructuras lógicas conocidas como segmentos. Los segmentos se relacionan entre sí utilizando arcos.

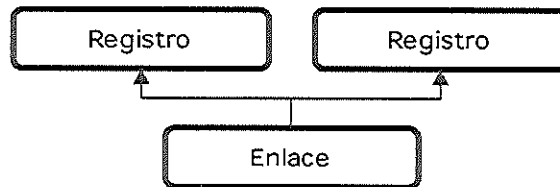


### *Modelo en red*

Es un modelo ampliamente aceptado. Organiza la información en registros, denominados nodos, y enlaces. En los registros se almacenan los datos, mientras

que los enlaces permiten relacionarlos. Las bases de datos en red son parecidas a las jerárquicas, pero en este caso puede haber más de un padre.

En este modelo se pueden representar perfectamente cualquier tipo de relación entre los datos pero esto dificulta su manejo.



#### *Modelo relacional*

En este modelo los datos se organizan en tablas donde los datos se relacionan. Es el modelo más popular de todos.

#### *Modelo de bases de datos orientados a objetos*

Tras la aparición de la programación orientada a objetos, surge la necesidad de crear bases de datos adaptadas a este lenguaje. La programación orientada a objetos permite la cohesión de datos y procedimientos, haciendo que se diseñen estructuras que poseen datos en las que se definen los procedimientos que pueden realizar con los datos. En las bases de datos orientadas a objetos se utiliza esta misma idea.

A través de esta idea se pretenden que estas bases de datos consigan arreglar las limitaciones de las relacionales.

Se supone por tanto que estas son las bases de datos de tercera generación, pero siguen sin reemplazar a las relacionales, aunque son el tipo de base de datos que más está creciendo en los últimos años.

#### *Bases de datos objeto-relacionales*

Pretenden ser un híbrido entre el modelo relacional y el orientado a objetos. El problema de la base de datos orientados a objetos es que requieren de una reinversión de capital y esfuerzo para convertir las bases de datos relacionales en bases de datos orientadas a objetos.

En las bases de datos objeto relacional se intenta conseguir una compatibilidad relacional dando la posibilidad de integrar mejoras de la orientación a objetos.

Estas bases de datos se basan en el estándar SQL 99. En este estándar se añade a las bases relacionales la posibilidad de almacenar procedimientos de usuarios, tipos definidos por el usuario, recursos para consulta, etc.

### **Modelos conceptuales**

Estos modelos representan la información de forma absolutamente independiente al sistema gestor de base de datos. Los esquemas internos de las diferentes bases de datos no captan suficientemente bien la semántica del mundo real, de ahí que primero haya que pasar por uno o dos esquemas previos más cercanos al mundo real.

Algunos de los ejemplos de modelos conceptuales son:

- Modelo entidad-relación.
- Modelo RM/T.
- Modelos semánticos.

### **Diferentes tipos de parámetros**

Algunos de los parámetros más comunes en los sistemas de gestión de bases de datos son:

- Redireccionamiento.
- Ruta de programas.
- Ruta de menús.
- Ruta de mensajes.
- Ruta de archivo de trabajo.
- Ruta de bases de datos.
- Idiomas implícitos.
- Separador de campos repetibles.
- Lista inicial de autodigitado.
- Modo de edición inicial.
- Caracteres gráficos para recuadros.
- Manejo de memoria expandida.

- Manejo de redes.
- Parámetros para meno de teclas de función.

## RECUERDA

- El proceso a través del cual se guardan los datos en un almacén de datos, se basa en:
  - El proceso de extracción.
  - El proceso de transformación.
  - El proceso de carga.
- Los mecanismos utilizados para la extracción de datos son:
  - Los sistemas OLAP On-Line Analytical Processing.
- La información es un conjunto de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje.
- Los descriptores son los encargados de sintetizar la información para así optimizar la localización.
- Un extractor de datos es un conjunto de motores de extracción que permite obtener mediante la extracción los datos de las bases de datos y utilizar programas de visualización.
- El proceso de obtención de la información comienza por la selección de las fuentes utilizables, es decir, los lugares en los que se obtiene la información útil para poder realizar las diferentes tareas con ella.
- En el proceso de tramitación de la información para su posterior estudio, usamos:
  - La agrupación.
  - La transformación.
  - La homogeneización
- El sistema central de almacenamiento de datos del CRM está compuesto por los datos almacenados, un mecanismo que los mantiene actualizados y otro mecanismo que gestiona y controla los datos.
- Cada sistema gestor de datos puede utilizar un modelo diferente para los datos. Por lo que hay modelos conceptuales diferentes según el sistema gestor de bases de datos que se utilicen. Que son:

**5. Indica si es verdadero o falso el siguiente enunciado:**

*"Los gestores de datos son un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información de una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos".*

- a) Verdadero.
- b) Falso.

# UD7 Biblioteca de funciones básicas



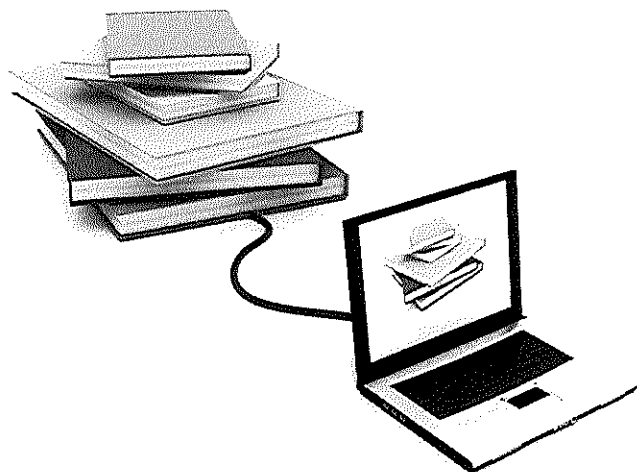
UF1889 Desarrollo de  
Componente Software  
en Sistemas ERP-CRM





# 1. Definición de funciones

En informática, una **biblioteca** (del inglés *library*) es un conjunto de implementaciones de comportamiento, escritas para un lenguaje de programación, que tienen una interfaz bien definida para el comportamiento que se invoca.



A diferencia de un programa ejecutable, el comportamiento que implementa una biblioteca no espera ser utilizada de forma autónoma (un programa sí: tiene un punto de entrada principal), sino que su fin es ser utilizada por otros programas, independientes y de forma simultánea. Por otra parte, el comportamiento de una biblioteca no tiene porqué diferenciarse en demasía del que pudiera especificarse en un programa. Es más, unas bibliotecas pueden requerir de otras para funcionar, pues el comportamiento que definen refina, o altera, el comportamiento de la biblioteca original; o bien la hace disponible para otra tecnología o lenguaje de programación.

Las bibliotecas pueden vincularse a un programa (o a otra biblioteca) en distintos puntos del desarrollo o la ejecución, según el tipo de vínculo que se quiera establecer, tal y como se detalla en el apartado de #Tipos.

La mayoría de los sistemas operativos modernos proporcionan bibliotecas que implementan los servicios del sistema. De esta manera, estos servicios se han convertido en una "materia prima" que cualquier aplicación moderna espera que el sistema operativo ofrezca. Como tal, la mayor parte del código utilizado por las aplicaciones modernas se ofrece en estas bibliotecas.

## **Tipos de bibliotecas**

Hay dos tipos de bibliotecas: estáticas y dinámicas. Veamos a continuación cada una de ellas.

### **Bibliotecas estáticas**

Históricamente, las bibliotecas sólo podían ser estáticas. Una biblioteca estática, también conocido como archivo es un fichero contenedor con varios archivos de código objeto empaquetados, que en el proceso de enlazado durante la compilación serán copiados y relocados (si es necesario) en el fichero ejecutable final, junto con el resto de ficheros de código objeto. Este proceso, y el archivo ejecutable, se conoce como una construcción estática de la aplicación objetivo. En este caso, la biblioteca actúa simplemente como un recipiente para ficheros de código objeto que no se diferencian (más que semánticamente) de los ficheros objeto intermedios producidos durante la etapa previa de compilación del programa. En la construcción estática de ficheros compilados se resuelven las direcciones de las subrutinas ensambladas en tiempo de compilación (más específicamente, en la etapa de enlazado), de modo que las referencias a subrutinas de la biblioteca se resuelven estáticamente, del mismo modo que las referencias a cualquier otra función del programa. Así, la dirección real, las referencias para saltos y otras llamadas a rutinas se almacenan en una dirección relativa o simbólica.

El enlazador resuelve todas las direcciones no resueltas convirtiéndolas en direcciones fijas o relocizables (desde una base común) cargando todo el código (incluyendo las bibliotecas) en posiciones de memoria en tiempo de ejecución. Este proceso de enlazado puede durar incluso más tiempo que el proceso de compilación, y debe ser realizado cada vez que alguno de los módulos es recompilado.

Un enlazador pueden trabajar sobre tipos específicos de ficheros objeto, y por lo tanto requiere tipos específicos (compatibles) de bibliotecas. Los ficheros objeto recompilados en una biblioteca pueden distribuirse y utilizarse fácilmente. Un cliente, ya sea un programa u otra biblioteca, accede a una biblioteca objeto referenciando sólo por su nombre. El proceso de enlazado resuelve las referencias buscando en las bibliotecas del orden dado. Por lo general, no se considera un error si un nombre puede encontrarse varias veces en un determinado conjunto de las bibliotecas.

## Bibliotecas dinámicas

Las bibliotecas dinámicas, vinculadas dinámicamente, o de vínculos dinámicos son ficheros que contienen código objeto construido de forma independiente a su ubicación de tal modo que están preparadas para poder ser requeridas y cargadas en tiempo de ejecución por cualquier programa, en lugar de tener que ser enlazadas, previamente, en tiempo de compilación. Por tanto, han de estar disponibles como ficheros independientes al programa ejecutable (generalmente en directorios del sistema). En el proceso de enlazado (en tiempo de compilación) se genera un fichero ejecutable con anotaciones de qué bibliotecas dinámicas requiere (pero no de dónde encontrarlas), y funciones de "esbozo" que se encargan de delegar la llamada a la función al cargador dinámico (o dynamic-loader) (en Linux, ld.so). En el resto del programa, las llamadas a las funciones de la biblioteca se cambian por una llamada a la función de esbozo generada por el enlazador.

Por otra parte, cuando la aplicación que se ejecute requiera acceder a las rutinas almacenadas en una biblioteca dinámica, y ejecute la función de esbozo, el cargador de enlaces dinámicos podrá sustituir esta llamada por la función real de la biblioteca dinámica, cargándola en memoria si no lo estuviera ya, y mapeando las páginas de ésta en el espacio de memoria del proceso del programa.

En algunos sistemas operativos puede decidirse si una biblioteca ha de estar disponible inmediatamente o solamente cuando se haga referencia a una función de ella. Si se decide esto último, aparecerá un fenómeno denominado retraso de carga, derivado de tener que cargar de memoria secundaria la biblioteca, si no estuviera ya en memoria, y de ajustarla al espacio de direcciones del programa contra el que se vincula.

Ventajas del enlace dinámico respecto al estático son que se permite la reutilización no solo de código, sino de espacio físico: un mismo fichero de biblioteca compartida puede ser utilizada por varios programas sin que estos copien su contenido dentro de ellos. Esto puede llegar a ser bastante espacio, según el número de bibliotecas que requiera un programa. Además, puede reutilizarse memoria principal (RAM) para programas que utilicen la misma biblioteca (por ejemplo, puede ser necesario cargar las bibliotecas de Qt únicamente una vez para todos los programas que las utilicen).

Por otra parte, el mayor inconveniente es el aumento del tiempo de carga (debido a tener que buscar el fichero de la biblioteca, cargarlo y relocalizar las

llamadas en el programa) y el aumento de una indirección a la hora de llamar a las funciones de la biblioteca.

El enlace dinámico, por su naturaleza, tiene tan sólo las limitaciones establecidas por las licencias de software.

La tecnología que permite enlazar bibliotecas de forma dinámica es muy útil para la construcción de plugins, sobre todo cuando unas bibliotecas pueden ser sustituidas por otras con una interfaz similar, pero diferente funcionalidad. Se puede decir que un software tiene una "arquitectura de plugin" si utiliza bibliotecas con una funcionalidad básica con la intención de que puedan ser sustituidas. Sin embargo, el uso de las bibliotecas enlazadas dinámicamente en la arquitectura de una aplicación no significa necesariamente que puedan ser sustituidas.

El enlace dinámico se desarrolló originalmente en los sistemas operativos Multics a partir de 1964. Se trataba de una característica del MTS (Michigan Terminal System), construido a finales de los 60.

En distintos sistemas operativos toman distintos nombres, por ejemplo:

- En Microsoft Windows: DLL (dynamic-link library).
- En Linux: shared-objects.
- En Mac OS: "bibliotecas dinámicas" (dylibs).

## **Definición de funciones**

Cada lenguaje de programación será el responsable de definir las funciones de la Biblioteca.

Por ejemplo, el lenguaje de programación C ofrece un conjunto de funciones estándar que dan soporte a las operaciones que se utilizan con más frecuencia. Como hemos visto, estas funciones están agrupadas en bibliotecas. Para utilizar cualquiera de las funciones que forman parte de las bibliotecas estándar de C, sólo hace falta realizar una llamada a dicha función.

Las funciones que forman parte de la biblioteca estándar de C, funciones estándar o predefinidas, están divididas en grupos. Todas las funciones que pertenecen a un mismo grupo se definen en el mismo fichero de cabecera.

Los nombres de los ficheros cabeceras de C se muestran en la siguiente tabla:

assert	ctype	errno	float
limits	locale	math	setjmp
signal	stdarg	stddef	stdio
stdlib	string	time	

Cuando deseamos utilizar cualquiera de las funciones estándar de C, primero debemos utilizar la directiva de precompilación *#include* para incluir los ficheros cabecera en nuestro programa. Por otra parte, antes de utilizar una función, primero debemos conocer las características de dicha función, es decir, el número y tipo de datos de sus argumentos y el tipo de valor que devuelve. Esta información es proporcionada por los prototipos de función.

Los grupos de funciones estándar más comunes son:

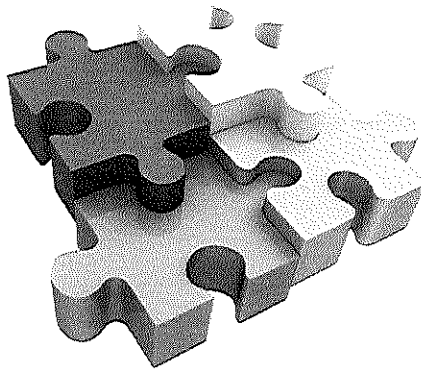
- Entrada/salida estándar.
- Matemáticas.
- De conversión.
- Diagnóstico.
- De manipulación de memoria.
- Control de proceso.
- Ordenación.
- Directorios.
- Fecha y hora.
- Búsqueda.
- Manipulación de cadenas.
- Gráficos.
- Etcétera.

Podemos incluir tantos ficheros de cabecera como nos haga falta, incluyendo los ficheros de cabecera que hemos creado y donde hemos definido nuestras funciones.

## 2. Definición de librerías de funciones (API)

Las siglas API significan Application Programming Interface que traducido literalmente al castellano sería interfaz de programación de aplicaciones.

Una API es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas.



Una interfaz de programación representa la capacidad de comunicación entre componentes de software. Se trata del conjunto de llamadas a ciertas bibliotecas que ofrecen acceso a ciertos servicios desde los procesos y representa un método para conseguir abstracción en la programación, generalmente (aunque no necesariamente) entre los niveles o capas inferiores y los superiores del software.

Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla. De esta forma, los programadores se benefician de las ventajas de la API haciendo uso de su funcionalidad, evitándose el trabajo de programar todo desde el principio. Las APIs asimismo son abstractas: el software que proporciona una cierta API generalmente es llamado la implementación de esa API.

Por ejemplo, se puede ver la tarea de escribir "*Hola Mundo*" sobre la pantalla en diferentes niveles de abstracción:

- Haciendo todo el trabajo desde el principio:
  - Traza, sobre papel milimetrado, la forma de las letras (y espacio) "H, o, l, a, M, u, n, d, o".
  - Crea una matriz de cuadrados negros y blancos que se asemeje a la sucesión de letras.
  - Mediante instrucciones en ensamblador, escribe la información de la matriz en la memoria intermedia ("buffer") de pantalla.
  - Mediante la instrucción adecuada, haz que la tarjeta gráfica realice el volcado de esa información sobre la pantalla.
- Por medio de un sistema operativo para hacer parte del trabajo:
  - Carga una fuente tipográfica proporcionada por el sistema operativo.
  - Haz que el sistema operativo borre la pantalla.
  - Haz que el sistema operativo dibuje el texto "Hola Mundo" usando la fuente cargada.
- Usando una aplicación (que a su vez usa el sistema operativo) para realizar la mayor parte del trabajo:
  - Escribe un documento HTML con las palabras "Hola Mundo" para que un navegador Web como Firefox, Chrome, Opera, Safari, Midori, Iceweasel, Web o Internet Explorer pueda representarlo en el monitor.

Como se puede ver, la primera opción requiere más pasos, cada uno de los cuales es mucho más complicado que los pasos de las opciones siguientes. Además, no resulta nada práctico usar el primer planteamiento para representar una gran cantidad de información, como un artículo enciclopédico sobre la pantalla, mientras que el segundo enfoque simplifica la tarea eliminando un paso y haciendo el resto más sencillos y la tercera forma simplemente requiere escribir "Hola Mundo". Sin embargo, las APIs de alto nivel generalmente pierden flexibilidad; por ejemplo, resulta mucho más difícil en un navegador web hacer girar texto alrededor de un punto con un contorno parpadeante que programarlo a bajo nivel. Al elegir usar

una API se debe llegar a un cierto equilibrio entre su potencia, simplicidad y pérdida de flexibilidad.



## RECUERDA

- En informática, una biblioteca (del inglés library) es un conjunto de implementaciones de comportamiento, escritas para un lenguaje de programación, que tienen una interfaz bien definida para el comportamiento que se invoca.
- Hay dos tipos de bibliotecas: estáticas y dinámicas:
  - Bibliotecas estáticas: la biblioteca actúa simplemente como un recipiente para ficheros de código objeto que no se diferencian (más que semánticamente) de los ficheros objeto intermedios producidos durante la etapa previa de compilación del programa.
  - Bibliotecas dinámicas: son ficheros que contienen código objeto construido de forma independiente a su ubicación de tal modo que están preparadas para poder ser requeridas y cargadas en tiempo de ejecución por cualquier programa, en lugar de tener que ser enlazadas, previamente, en tiempo de compilación.
- Cada lenguaje de programación será el responsable de definir las funciones de la Biblioteca.
- Una API es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.
- Uno de los principales propósitos de una API consiste en proporcionar un conjunto de funciones de uso general, por ejemplo, para dibujar ventanas o iconos en la pantalla.



## Preguntas de Autoevaluación

### 1. Completa el espacio en blanco del siguiente enunciado:

"En informática, una \_\_\_\_\_ es un conjunto de implementaciones de comportamiento, escritas para un lenguaje de programación, que tienen una interfaz bien definida para el comportamiento que se invoca".

- a) Biblioteca.
- b) Base de datos.
- c) Sentencia.

### 2. La biblioteca puede ser:

- a) Estática o dinámica.
- b) Estática o semidinámica.
- c) Flexible o inmutable.

### 3. El enlace dinámico se desarrolló originalmente en los sistemas operativos:

- a) Linux.
- b) Multics.
- c) Microsoft Windows.

**4. El conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción se denomina:**

- a) Interfaz.
- b) Stddef.
- c) API.

**5. Indica si es verdadero o falso el siguiente enunciado:**

*"Un enlazador pueden trabajar sobre tipos específicos de ficheros objeto, y por lo tanto no requiere tipos específicos (compatibles) de bibliotecas".*

- a) Verdadero.
- b) Falso.

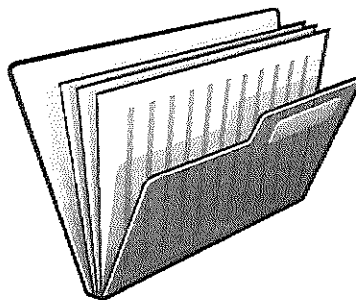
# UD8 Documentación



UF1889 Desarrollo de  
Componente Software  
en Sistemas ERP-CRM



# 1. Documentación del análisis funcional



El **Análisis Funcional** consiste en analizar la información obtenida en reuniones con el personal implicado, teniendo en cuenta los objetivos del proyecto y los recursos disponibles, y redactar toda esta información para que esté al alcance de todos. Además, servirá como referencia durante la propia ejecución del proyecto y a posteriori para conocer en todo momento el alcance, y determinar los niveles de éxito en la consecución del proyecto.

En cualquier proyecto de software basado en el desarrollo a medida o la implantación de estándares, es preciso elaborar documentación con toda la información que utilizarán como referencia las personas que ejecutarán los trabajos del proyecto a diferentes niveles.

El cometido de esta documentación es determinar los objetivos del proyecto en colaboración con el cliente y plasmar los resultados por escrito, incluyendo (en función de la complejidad) diagramas de flujo, modelos entidad-relación, navegabilidad, definición de servicios en forma narrativa, formatos de intercambio de información, etc.

El Análisis Funcional se realiza siempre como primera fase del proyecto, ya que la documentación que resulta es la entrada de información principal para las siguientes fases. De todo esto depende al menos el buen comienzo de un proyecto y del que va a depender todo lo demás.

A continuación, se muestra una definición que habla sobre el Analista Funcional como la figura entre los usuarios y el Departamento de Informática de una empresa: "El Analista Funcional es el vínculo de unión entre el usuario y el área informática de la empresa. Su misión consiste en elaborar el análisis funcional de nuevas aplicaciones para la organización, así como actualizar y mejorar las ya

existentes; es decir, debe controlar, analizar y supervisar el desarrollo funcional de las aplicaciones informáticas, asegurando su correcta explotación y su óptimo rendimiento”.

### **El analista funcional y el analista orgánico**

Por regla general, se considera **analista funcional** a quien se encarga de la recopilación del catálogo de requisitos y de la definición de los casos de uso (o historias de usuario). Su objetivo es describir las funcionalidades del sistema y su comportamiento mediante el estudio de las necesidades del usuario. Su trabajo es muy importante y sin embargo no se le da el valor que se merece, ya he comentado en más de una ocasión que lo técnico suele resultar más glamuroso.

Los sistemas se desarrollan para el usuario y para que sean de utilidad hay que saber captar qué es lo que quiere el usuario y cómo desea interactuar con el sistema.

La técnica es muy importante, pero no lo es menos que la vertiente funcional, todos suman y todos restan si no hacen adecuadamente su trabajo.

El **analista orgánico** se encarga del diseño que no es otra cosa que la particularización de las necesidades del usuario a una implementación concreta. Para un proyecto concreto vendría a ser el arquitecto de la solución, ya que entraría incluso a definir el framework con el que se va a trabajar.

Lo habitual en los proyectos de desarrollo de software es encontrarnos con analistas que realizan las dos funciones, aunando en un solo perfil lo funcional y lo técnico. Tiene como principal ventaja que no es necesario transferir el conocimiento entre diferentes fases o procesos del desarrollo y que cada tarea es realizada por un experto en la misma. El principal inconveniente es que no existe una especialización, no obstante, hay que ser bastante bueno en uno de los dos perfiles para marcar de manera clara la diferencia respecto a analistas que realicen ambos trabajos (ahora, quien las marca es que es muy bueno y por tanto, vale mucho dinero).

En desarrollos iterativos e incrementales con una orientación ágil lo deseable es que la evolución del desarrollo sea rápida (realizándose los incrementos con cierta frecuencia) y resulta fundamental la participación del usuario, a ser posible como una pieza más del equipo de proyecto, lo normal es que no exista distinción entre analista funcional y orgánico, si bien, tendrá una vertiente más técnica, por



las características particulares de este tipo de desarrollos. Existirán excepciones, en proyectos que por su naturaleza requiera de desarrolladores que den apoyo funcional, por la experiencia que tengan en un tipo de negocio concreto.

### **La importancia de un buen análisis funcional**

De todos es sabido que cuanto antes se solucione un problema en un proyecto de desarrollo de software, menos coste tiene para el mismo y de ello salen beneficiados tanto proveedor como cliente.

Por ese motivo resulta esencial que un proyecto sea sólido desde la base, siendo la misma el análisis funcional, lo que hace que sea muy importante la figura del analista que es la persona o grupo de personas (si el proyecto es grande) que se tienen que encargar de entender, interpretar y traducir lo que el usuario demanda, sentando las bases de los posteriores procesos de diseño y construcción del sistema de información.

Hacer un buen análisis es una tarea bastante compleja, ya que resulta muy complicado obtener todos los requerimientos del usuario desde etapas muy tempranas, ya que por regla general el usuario empieza a descubrir el detalle de todo lo que quiere cuando empieza a utilizar el producto ya construido con ejemplos reales de su día a día de trabajo (también suelen comentar nuevos requisitos o enmendar requisitos previos, en otras etapas conforme se le vaya presentando la evolución del proyecto, de hecho no es malo que se corrijan, ya que cuanto más avanzado esté el proyecto, el esfuerzo de hacer los cambios es mucho mayor). De hecho es prácticamente inevitable no hacer evolutivos que solventen esos flecos que no se detectaron en análisis para dejar el producto lo más próximo posible a lo que los usuarios necesitan y demandan. Como consecuencia de lo anterior, y como es lógico, se puede considerar que un análisis funcional es más bueno conforme sea menor el número de ajustes que haya que hacer en etapas posteriores del proyecto.

Es importante matizar que un proyecto de desarrollo de software no es una barra libre y que es importante que el usuario conozca sus responsabilidades en el proceso de definición del sistema y que no se pueden estar cambiando de requisitos continuamente, como tampoco podría estar cambiando frecuentemente de opinión si le están construyendo una casa. Todo lo anterior además hay que compatibilizarlo con que todas las partes están interesadas en el que el proyecto vaya a buen término, por lo que tampoco es una buena política ser inflexibles en la

modificación del catálogo de requisitos, porque si el resultado final no es el que quiere el usuario, el sistema de información tendrá muchas papeletas para no ser utilizado. Equilibrio complicado: evitar que los usuarios modifiquen continuamente requisitos y tener un poco de mano izquierda cuando se planteen esos cambios. Como ese equilibrio es complicado de mantener y es fuente frecuente de conflictos, hay que intentar que el análisis tenga la mayor calidad posible.

Hacer un análisis funcional por tanto es una tarea compleja, a lo que hay que sumar que en muchos casos hay que aprender mucho sobre el proceso de negocio que se pretende informatizar, para entender de mejor manera lo que el usuario demanda, ya que resulta todo más fácil si el lenguaje que se utiliza es el mismo. En muchas ocasiones esos procesos de negocio son tremendamente complejos y además se dispone también de poco tiempo para entenderlos, teniendo en cuenta que por regla general y como he comentado muchas veces en mi blog, los proyectos informáticos suelen estar infravalorados (por el que contrata y/o por el que es contratado (para conseguir el contrato)).

Dado que el análisis funcional consiste en abstraer un conjunto de necesidades de los usuarios es muy importante la implicación de los mismos y eso no siempre se consigue. Si los usuarios no están implicados, por muy buen analista funcional que tenga el proyecto, las probabilidades de que este salga mal crecen exponencialmente. Evidentemente un buen analista puede paliar esos huecos que deja el usuario e incluso conseguir una mayor participación de los usuarios, pero más tarde o más temprano los problemas aparecerán y al final siempre termina pagando el proyecto (en primera instancia) y el que lo desarrolla (en segunda). Por todo lo anterior, se puede pedir que un analista funcional aprenda un proceso de negocio complejo, que consiga extraer de los usuarios lo que buscan y necesitan y que además lo haga en un tiempo record, pero lo que no se le puede pedir es que haga magia y resuelva problemáticas que le trascienden, como el caso que he comentado de la inacción de los usuarios en determinados proyectos, siendo esa falta la implicación la primera causa de que un análisis no salga bien y por tanto una de las causas más importantes del fracaso de un proyecto y no se trata en este caso de tirar pelotas fuera y ponerme del lado de mis colegas de profesión, se trata de algo que he podido vivir en diferentes proyectos de manera muy directa.

Nadie es infalible y un analista funcional tampoco lo es. Habrá errores (independientemente de que la causa de los mismos sea provocada por circunstancias adversas en el proyecto o no), puede que en este proyecto sean muy pocos y que en otros sean mayores, por eso es importante que el analista lo tenga

asumido desde un principio, como también lo es que de esos errores se debe aprender y que resultarán fundamentales en la formación del mismo. Al final esos errores terminan curtiendo y permiten que cada vez los análisis que se realicen sean mejores. Por tanto, la experiencia resulta importante. En cualquier caso, suponer un análisis perfecto es suponer que en un proyecto se dan circunstancias ideales y que todas las variables que pueden influir en que las cosas vayan mejor o peor, están todas a favor.

De todo lo comentado en los párrafos anteriores se extrae que es importante que un analista funcional sea un buen comunicador, primero con el usuario ya que resulta fundamental que el usuario conozca todo lo que hemos entendido y cómo se pretende llevar a cabo (no conseguir eso es ir a ciegas) y segundo con el equipo de proyecto ya que tiene que trasladar a documentación y hacerles entender la interpretación de lo que el usuario quiere y cómo lo quiere.

Un buen análisis funcional no asegura el éxito del proyecto, ya que la ejecución técnica del mismo también tiene un peso importante, pero lo que sí es seguro es que si el análisis funcional no es bueno, la ejecución técnica difícilmente puede salvar las deficiencias del mismo y tocará corregir el producto una vez construido y además, por regla general, en diferentes evoluciones, lo cual es muy costoso y tampoco asegura que ese árbol que empezó torcido, termine por enderezarse.

## 2. Documentación de las librerías y funciones



La **documentación** es el conjunto de elementos registrados sobre cualquier soporte que ayuda a instruir o informar acerca de algo, en función de las necesidades específicas de aquellos que lo utilizan.

Esta documentación es muy importante ya que es el elemento integrador que permite la apreciación unitaria y conjunta del sistema. Facilita el conocimiento, interpretación y comprensión del sistema y además permite el control interno en general y del sistema en particular. Los principales documentos son:

- Hoja de diseño de archivos o registros.
- Índice de archivos.
- Hoja de diagramación.
- Hoja de diseño de salidas impresas y/o formularios.
- Hoja de diseño de formatos de pantalla.
- Hoja de programación.
- Índice de programas.
- Tabla de decisiones y/alternativas.
- Hoja de especificaciones.

#### **Carpeta de papeles de trabajo**

- Síntesis del documento de generación.
- Presupuesto o plan de fijación de tareas.
- Documentación del relevamiento detallado.
- Formularios o comprobantes analizados.
- Papeles de trabajo de análisis.
- Estudio de factibilidad y diagnóstico.

#### **Carpeta de sistemas**

- Fijación de los objetivos del sistema.
- Descripción global del sistema.
- Modelo lógico del sistema.
- Diseño de entradas y salidas.
- Normas y procedimientos para los usuarios.
- Recursos materiales y humanos necesarios.

- Estudio técnico-económico acerca de la posibilidad de procesar el sistema mediante el uso de un computador.

### **Carpetas de programas**

- Descripción detallada del programa.
- Diagrama de lógica.
- Descripción de entradas.
- Descripción de salidas.
- Descripción de archivos.
- Tablas, cuadros de control de consistencia y parámetros utilizados.
- Controles de programas sobre archivos y datos.

### **Carpeta de operaciones**

- Normas de control de entradas, salidas y de procesamientos.
- Normas de operación, de recuperado, de back-up, de seguridad de archivos.
- Cronograma de procesos.
- Descripción de usuarios.



## RECUERDA

- El **Análisis Funcional** consiste en analizar la información obtenida en reuniones con el personal implicado, teniendo en cuenta los objetivos del proyecto y los recursos disponibles, y redactar toda esta información para que esté al alcance de todos.
- Hay dos **tipos de analista**:
  - Orgánico.
  - Funcional.
- **La importancia de un buen análisis funcional.** De todos es sabido que cuanto antes se solucione un problema en un proyecto de desarrollo de software, menos coste tiene para el mismo y de ello salen beneficiados tanto proveedor como cliente.
- La **documentación** es el conjunto de elementos registrados sobre cualquier soporte que ayuda a instruir o informar acerca de algo, en función de las necesidades específicas de aquellos que lo utilizan.
- Existen varios **tipos de carpetas**:
  - De sistemas
  - De programas
  - De operaciones
  - De papeles de trabajo





## Preguntas de Autoevaluación

### 1. Completa el espacio en blanco del siguiente enunciado:

"El Análisis Funcional se realiza siempre como \_\_\_\_\_ fase del proyecto, ya que la documentación que resulta es la entrada de información principal para las siguientes fases".

- a) Primera.
- b) Segunda.
- c) Tercera.

### 2. Los principales documentos son: (Respuesta múltiple)

- a) Hoja de diseño de archivos e índice de archivos.
- b) Hoja de diagramación y hoja de diseño de formularios.
- c) Tabla de programas y hoja de decisiones.

### 3. La descripción detallada del programa forma parte de la carpeta de:

- a) Papeles de trabajo.
- b) Sistemas.
- c) Programas.

### 4. ¿Cuáles son los contenidos mas importantes de la carpeta de operaciones? (Respuesta múltiple)

- a) Controles de programas sobre archivos y datos.
- b) Normas de control de entradas, salidas y de procesamientos.
- c) Cronograma de procesos.

**5. Indica si es verdadero o falso el siguiente enunciado:**

*"Un buen análisis funcional asegura el éxito del proyecto".*

- a) Verdadero.
- b) Falso.

# UD9 Pruebas y depuración de un programa



UF1889 Desarrollo de  
Componente Software  
en Sistemas ERP-CRM



# 1. Validación de programas

La **depuración de programas** es el proceso de identificar y corregir errores de programación. En inglés se le conoce como debugging, que se asemeja a la eliminación de bichos (bugs), manera en que se conoce informalmente a los errores de programación. Se dice que el término bug proviene de la época de los ordenadores de válvula termoiónica, en los cuales los problemas se generaban por los insectos que eran atraídos por las luces y estropeaban el equipo.

Si bien existen técnicas para la revisión sistemática del código fuente y se cuenta con medios computacionales para la detección de errores (depuradores) y facilidades integradas en los sistemas lower CASE y en los ambientes de desarrollo integrado, sigue siendo en buena medida una actividad manual, que desafía la paciencia, la imaginación y la intuición del programador.

Muchas veces se requiere incluir en el código fuente instrucciones auxiliares que permitan el seguimiento de la ejecución del programa, presentando los valores de variables y direcciones de memoria y ralentizando la salida de datos (modo de depuración). Dentro de un proceso formal de aseguramiento de la calidad, puede ser asimilado al concepto de prueba unitaria.



Por su parte, la validación es la acción y el efecto de validar (convertir algo en válido, darle fuerza o firmeza). En el ámbito del software, se conoce como **pruebas de validación** al proceso de revisión al que se somete un programa informático para comprobar que cumple con sus especificaciones. Dicho proceso, que suele tener lugar al final de la etapa de desarrollo, se realiza principalmente con la intención de confirmar que el software esté en condiciones de desarrollar las tareas que el usuario que lo adquiere planea llevar a cabo.

Los principales objetivos de la validación de programas son:

- Detectar y corregir cuanto antes los defectos y las desviaciones del programa respecto al objetivo fijado.
- Disminuir los riesgos, las desviaciones sobre los presupuestos y sobre el calendario y programa de tiempos del proyecto.
- Mejorar la calidad y fiabilidad del software.
- Valorar rápidamente los cambios propuestos y sus consecuencias.

### **Establecimiento de métodos para trazar el software utilizado para reproducir errores**

Los nuevos componentes y elementos software del sistema de ERP y CRM se transportan entre los distintos entornos de desarrollo, prueba y explotación, ejecutando procedimientos específicos y realizando su seguimiento, para mantener el flujo de modificaciones y su verificación antes de ser liberadas para su utilización por el usuario final, siguiendo especificaciones técnicas y necesidades de la organización.

Un ejemplo de este tipo de software es KMKey Help Desk, que es un software de gestión de incidencias indicado para servicios de mantenimiento, ayuda al usuario y resolución de problemas en cualquier sector. Permite definir flujos de trabajo para abordar problemáticas derivadas de anomalías en servicios y maquinaria.



La incidencia puede recibirse de forma automática (e-mail, entrada a través de una web, desde un dispositivo móvil,...) o bien ser abierta por el servicio de atención. Una vez en marcha seguirá el flujo diseñado por el cliente para su resolución. Permite: resolución inmediata, escalado, consulta de información anterior, reparto de recursos, etc.

Si la incidencia da origen a una intervención de mayor orden se puede enlazar con la Gestión de Proyectos o la Gestión de Calidad.

Mediante KMKey Help Desk podrá realizar los trabajos necesarios para atender el mantenimiento de sus instalaciones y disponer, desde cualquier acceso Internet, de toda la información relevante.

#### Gestión de incidencias:

- **Recepción:** vía mail, entradas por el propio usuario, desde un formulario web etc. Abre de forma automática un expediente y avisa mediante notificación (interna, correo electrónico o SMS) al responsable.
- **Resolución:** atención y cerrado inmediato o planificación en el tiempo según tipo o urgencia. Asignación de recursos. Re-asignación. Escalado. Listas con el estado y los responsables. Retrasos. Avisos automáticos a quien sea necesario. Posibilidad de añadir notas, documentos, correos, fotos, manuales. Búsquedas en base de datos de conocimiento. Búsquedas indexadas en cualquier ítem, incluso dentro de los documentos. Introducción de horas y material utilizado. Facturación.
- **Control:** generación y envío automático del resumen de la incidencia. Generación de informes y estadísticos por varios criterios: urgencia, tipo, contrato, elemento, técnico.. Informes de horas invertidas. Informe de elementos problemáticos. Filtros e informes por periodos, personas, clientes etc.

USUARIO: Ramirez, Raúl ( 3 perfiles ) MENSAJE(S) 6 nuevos IMPUTADO 00.00 SALIR

### KMKey HelpDesk

DEFINICIÓN EQUIPO PLANIFICAR GESTIÓN CONTROL

Ver

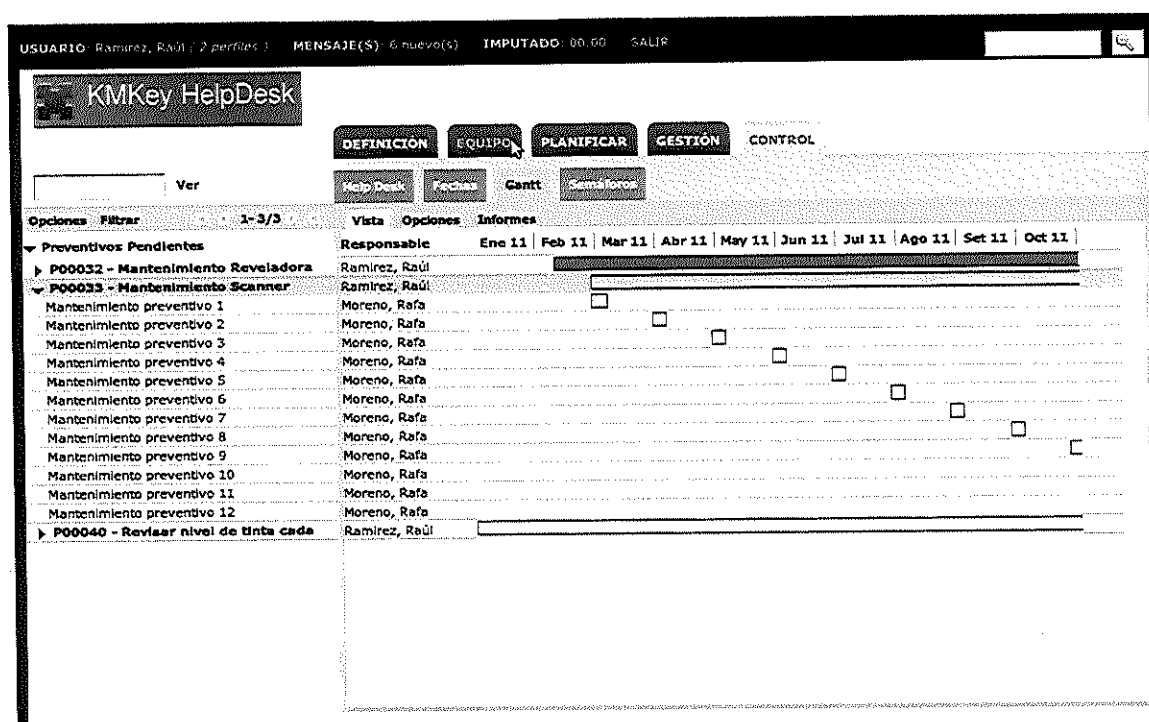
Opciones Filtrar 1-16/16

Vista Informes

	Cliente	Creator	Revisor	Asignado a	Inicio Prev.	Inicio Real	Fin Prev.	Fin Real
100029 - Rebasamiento de tinta	Semir, S.A.	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	2/02/11	2/02/11	2/02/11	3/03/11
100032 - La reveladora hace mucho	Estrada, S.L.	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	4/02/11	4/02/11	6/02/11	5/02/11
100033 - El software falla al enviar un	Cliente ejemplo	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	4/02/11	7/02/11	6/02/11	8/02/11
100035 - La filmadora no da registro	PROSA	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	7/02/11	7/02/11	8/02/11	8/02/11
100038 - El router se desconfigura	Cliente ejemplo	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	16/02/11	24/02/11	20/02/11	25/02/11
100039 - El scanner se para a media	Techno Print	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	24/02/11	24/02/11	25/02/11	25/02/11
100040 - La impresora se encalla	Brickford, S.L.	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	25/02/11	25/02/11	25/02/11	26/02/11
100043 - Se atasca el papel	Estrada, S.L.	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	7/03/11		8/03/11	
100044 - Al scanner le falta tinta	Techno Print	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	7/03/11	7/03/11	11/03/11	7/03/11
100045 - El plotter se atasca con	Samir, S.A.	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	2/11/11		3/11/11	
100046 - Impresora falla	Samir, S.A.	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	14/11/11	14/11/11	18/11/11	14/11/11
100047 - Cash machine se atasca	Samir, S.A.	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	7/12/11	7/12/11	8/12/11	7/12/11
100048 - Nueva en omnibab	Estrada, S.L.	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	14/12/11	14/12/11	18/12/11	14/12/11
100049 - No se scanner	Techno Print	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	19/12/11	9/01/12	20/12/11	
100050 - Sale error al entrar con	Samir, S.A.	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	9/01/12	9/01/12	13/01/12	9/01/12
100051 - No emite bien la señal	Brickford, S.L.	Gavidia, Eva	Ramirez, Raúl	Moreno, Rafa	11/01/12	11/01/12	15/01/12	11/01/12

### Mantenimiento preventivo:

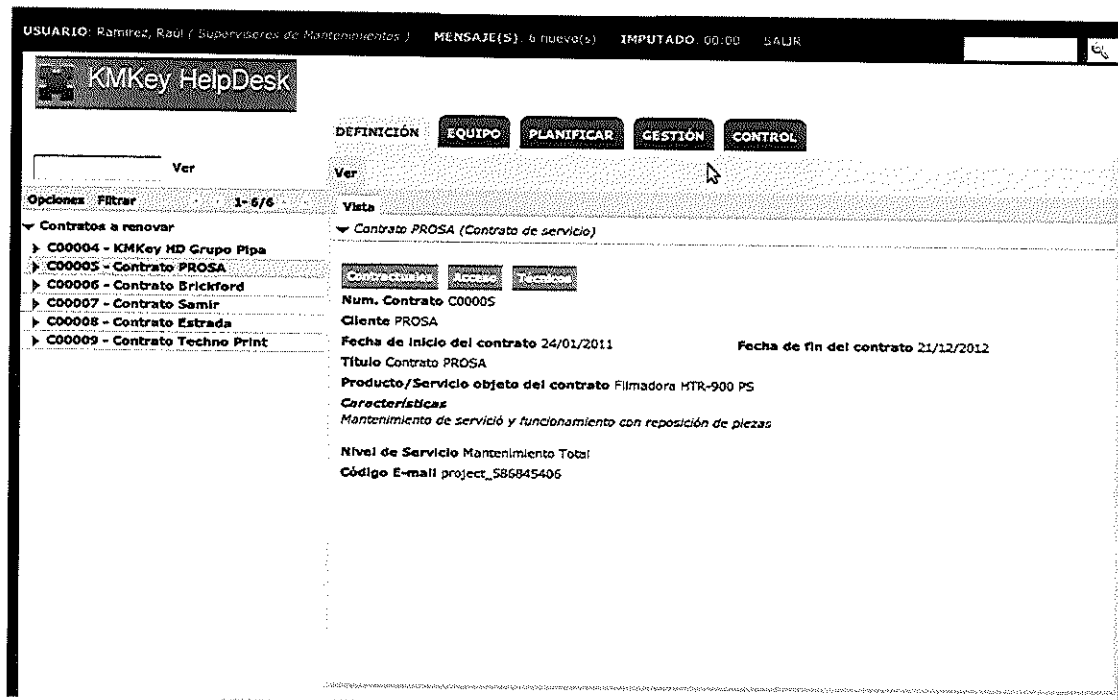
- **Planificación:** posibilidad de lanzar mantenimientos preventivos de elementos o contratos con diversas periodicidades. semanal, mensual, trimestral anual etc. Una vez lanzado, disponemos de tareas a realizar en determinados periodos. Podemos asignar técnicos, por defecto o mensualmente. Informes y listados de tareas a realizar en un periodo determinado.
- **Realización:** avisos y listados de tareas a realizar. Gráficos GANTT. Reflejar tiempos utilizados y material. facturación y seguimiento administrativo. Adjuntar documentación, correos electrónicos etc.
- **Control:** informes de cada actuación realizada. envío automático. listados y estadísticos de trabajos realizados.



### Gestión de contratos y elementos:

- **Inventario:** elementos organizados por diferentes criterios: tipo, departamento, contratos etc.
- **SLA:** posibilidad de guardar y controlar documentación relativa a los contratos y condiciones de servicio. Realización de mantenimientos e intervenciones según las condiciones reflejado en las SLA (tiempos, urgencias, visibilidades).





Otros:

- **Agenda:** base de datos de empresas y contactos. Calendario actividades. Mailing.
- **Integración e-mail y SMS:** notificaciones a terceros vía mail de acciones y tareas. Recepción automática de mails. Envío SMS.
- **Enlaces:** con la Gestión de Calidad o con la Gestión de proyectos. Enlaces con otros sistemas propios mediante desarrollos específicos: ERPs, BI, Blance Score Card etc.

USUARIO: Ramirez, Raúl ( 2 perfiles ) MENSAJE(S): 6 nuevo(s) IMPUTADO: 00:00 SALIR

## KMKey HelpDesk

DEFINICIÓN EQUIPO PLANIFICAR GESTIÓN CONTROL

Historia Agenda Documentos Info Mensaje

Opciones: Filtrar 1-3/3 Vista: Añadir Nota Imputar Trabajo Añadir Email Añadir Documento Cerrar Añadir Póliza

▼ Incidencias / Tickets Pendientes

- 100043 - Se atasca el papel
- ▼ 100045 - El plotter se atasca con Seguimiento
- 100049 - Ha pe scanner

▼ Cerrando Expediente: El plotter se atasca con frecuencia

**Título\***  
Plotter desatascado

**Descripción**  

 Hemos fijado la bandeja de alimentación para garantizar que no vuelva a suceder.  
 Carga de papel y pruebas correctas.

**Fecha de cierre**  
18/01/2012 17:03

**Horas**  
04 h 00 m

**Archivo adjunto:**  
 Tamaño máximo: 4.0 Mb.  
☒ Ninguno  
☐ Generarlo usando una Plantilla

**Email**  
 Gavidia Eva <jferren@earcon.com>  
 Moreno Rafa <jferren@earcon.com>  
 Ramirez Raúl <jferren@earcon.com>  
 Samir S.A. <group\_211869984>  
 Sánchez Laura <jferren@earcon.com>

**SMS**  
 Gavidia Eva - +34666999888  
 Moreno Rafa - +666999777  
 Ramirez Raúl - +66888111

USUARIO: Moreno, Rafa ( 2 perfiles ) MENSAJE(S): 17 nuevo(s) IMPUTADO: 00:00 SALIR

## KMKey HelpDesk

DEFINICIÓN EQUIPO PLANIFICAR GESTIÓN CONTROL

Ver Nuevo Help Desk Facturas Gestión

Opciones: Filtrar 1-17/17 Ver: Informes

▼ Incidencias / Tickets Todos

▼ Seleccione un listado

• Incidencias.xls (solo lectura) - OpenOffice.org Calc

Archivo Editar Ver Insertar Formato Herramientas Datos Ventana Ayuda

	A	B	C	D	E	F	G	H
1	KMKey HelpDesk							
2	Categoría	Incidencias						
3	Mantenimiento Correctivo	14.00						
4	Mantenimiento Preventivo	1.00						
5	Soporte Remoto	2.00						
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								

**Incidencias por categoría**

☒ Mantenimiento Correctivo  
☐ Mantenimiento Preventivo  
☐ Soporte Remoto

Hojas: 3 / 3 PageStyle: Categorías Suma=0 100%

## **Trazas y ficheros de confirmación de los procesos realizados (logs)**

Los procesos realizados, así como las incidencias producidas en la administración, se reflejan en la documentación para mantener registro de seguimiento de los trabajos realizados, siguiendo el procedimiento establecido por la organización.

La documentación técnica específica asociada se interpreta, en su caso, en la lengua extranjera de uso más frecuente en el sector.

La palabra **log** es un término anglosajón, equivalente a la palabra bitácora en español. Sin embargo, se utiliza en los países de habla hispana como un anglicismo derivado de las traducciones del inglés en la jerga informática. Del mismo término también proviene la palabra blog, que es la contracción de "web log".

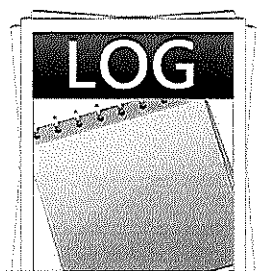
Un log es un registro oficial de eventos durante un rango de tiempo en particular. Para los profesionales en seguridad informática es usado para registrar datos o información sobre quién, qué, cuándo, dónde y por qué (who, what, when, where y why) un evento ocurre para un dispositivo en particular o aplicación.

La mayoría de los logs son almacenados o desplegados en el formato estándar, el cual es un conjunto de caracteres para dispositivos comunes y aplicaciones. De esta forma cada log generado por un dispositivo en particular puede ser leído y desplegado en otro diferente.

También se le considera como aquel mensaje que genera el programador de un sistema operativo, alguna aplicación o algún proceso, en virtud del cual se muestra un evento del sistema.

A su vez la palabra log se relaciona con el término evidencia digital. Un tipo de evidencia física construida de campos magnéticos y pulsos electrónicos que pueden ser recolectados y analizados con herramientas y técnicas especiales, lo que implica la lectura del log y deja al descubierto la actividad registrada en el mismo.

Para poder visualizar el contenido de estos ficheros, tan sólo necesitamos un editor de texto, como puede ser vi, nano, gedit, kwrite....



### **Documentación de procesos realizados**

Los LOG's, permiten detectar incidentes y comportamientos no habituales. Esta información es muy útil para localizar fallos en las configuraciones de los programas o cambios que se hayan hecho sobre dicha configuración. También detectan si se ha desconectado al dispositivo del sistema y controlar el uso de recursos por parte del usuario. Además de proporcionar información sobre el rendimiento del sistema, pueden resultar útiles para detectar cuándo un equipo está empezando a fallar.

La administración de usuarios, grupos y sus privilegios y también la de ficheros y directorios, se ha hecho correctamente. Hay que tener en cuenta las ACL (access control list) o listas de control de acceso. Mediante las ACL es posible la asignación de permisos a usuarios o grupos concretos. Esto puede ser útil en caso de que dos usuarios que pertenecen a grupos diferentes necesiten los mismos permisos a la hora de acceder a unos determinados directorios. Por ejemplo, un proyecto interdisciplinario entre profesores del departamento de Informática y el de Filosofía (los dos grupos de usuarios con perfiles perfectamente definidos dentro de cada departamento), podría requerir que los componentes del proyecto tuvieran que acceder a los mismos directorios, necesidad que se podría satisfacer con la creación de la ACL correspondiente.

En informática, el concepto de historial o de logging designa la grabación secuencial en un archivo o en una base de datos de todos los acontecimientos que afectan un proceso particular (aplicación, actividad de una red informática...). El término (en inglés log file o simplemente log) designa al archivo que contiene estas grabaciones. Generalmente fechadas y clasificadas por orden cronológico, estos últimos permiten analizar paso a paso la actividad interna del proceso y sus interacciones con su medio.

## Verificación de que los procesos se han realizado

Los Sistemas Operativos incorporan LOG's donde registran información sobre que usuarios y en qué momento abren y cierran sesión, que procesos están en ejecución dentro del sistema, las aplicaciones que son ejecutadas por los usuarios, y así como los posibles problemas de seguridad.

La grabación de un historial permite poner en marcha funciones tales como los «últimos archivos abiertos», los «últimos comandos tipeados» o las «últimas páginas web consultadas».

La **periodización aplicativa** designa la grabación cronológica de operaciones durante el funcionamiento de la aplicación. Una periodización aplicativa es en sí misma, una exigencia del trabajo. Está definida como una función que forma parte de la lógica aplicativa. Por lo tanto, no debería estar detenida durante el funcionamiento de la aplicación.

La **periodización sistema** designa la grabación cronológica de acontecimientos que sobrevienen a nivel de componentes del sistema. El nivel de esta periodización puede ser medido, con el fin de filtrar los diferentes acontecimientos, según su categoría de gravedad. Las categorías generalmente utilizadas son: información, depuración, advertencia, error. Por ejemplo los sistemas Unix ponen en marcha esta periodización sistema con la ayuda del protocolo Syslog.

## Características y tipos

No existe sólo un fichero .log, porque sino sería interminable e inmanejable, para ello están divididos en distintos ficheros y cada uno con su función específica. Por ejemplo:

- **syslog**: se encarga de registrar los mensajes de seguridad del sistema.
- **kern**: se encarga de los mensajes del núcleo (kernel).
- **messages**: archiva los distintos mensajes generales que nos manda el sistema.
- **debug**: mensajes de depuración de los programas.
- **user.log**: información sobre el usuario.

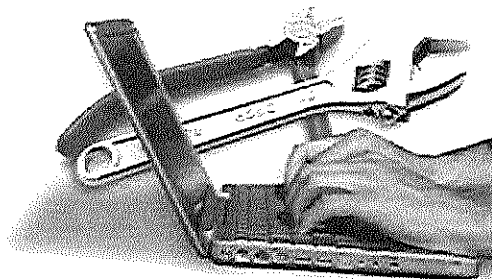
- **Xorg.0.log:** guarda información sobre el entorno gráfico.
- **auth.log:** contiene los accesos al sistema, incluidos los intentos fallidos.

## Herramientas de software utilizadas para el control de alarmas y la identificación de procesos

El administrador de un sitio web o de una red se basa software de monitorización red/servidor que continuamente se puede mantener un ojo en el sistema y pueden detectar componentes lentos o defectuosos. Estas herramientas de supervisión pueden enviar automáticamente las actualizaciones o activar copias de seguridad en caso de interrupciones causadas por accidentes o sobrecargas del servidor, conexiones de red y otros factores. Por ejemplo, para averiguar el estado de un servidor web, el software de monitoreo de vez en cuando puede enviar una petición HTTP a buscar una página.

## 2. Manejo de errores

Para ver el significado de un número de error en un programa, se utiliza el programa **perro**, que viene con la distribución de MySQL.



La siguiente tabla proporciona una lista con algunos códigos de error de sistema comunes en Linux.

<b>(EPERM)</b>	Operación no permitida
<b>(ENOENT)</b>	No existe el fichero o directorio
<b>(ESRCH)</b>	No existe el proceso

<b>(EINTR)</b>	Llamada de sistema interrumpida
<b>(EIO)</b>	Error de E/S
<b>(ENXIO)</b>	No existe el dispositivo o dirección
<b>(E2BIG)</b>	Lista de argumentos demasiado extensa
<b>(ENOEXEC)</b>	Error de formato ejecutable
<b>(EBADF)</b>	Número de fichero erróneo
<b>(ECHILD)</b>	No hay procesos hijos
<b>(EAGAIN)</b>	Intente nuevamente
<b>(ENOMEM)</b>	Memoria agotada
<b>(EACCES)</b>	Permiso denegado
<b>(EFAULT)</b>	Dirección errónea
<b>(ENOTBLK)</b>	Se necesita un bloque de dispositivo
<b>(EBUSY)</b>	El dispositivo o recurso está ocupado
<b>(EEXIST)</b>	El fichero ya existe
<b>(EXDEV)</b>	Vínculo de dispositivos cruzado (Cross-device link)
<b>(ENODEV)</b>	No existe el dispositivo
<b>(ENOTDIR)</b>	No es un directorio
<b>(EISDIR)</b>	Es un directorio
<b>(EINVAL)</b>	Argumento inválido
<b>(ENFILE)</b>	Desbordamiento de tabla de fichero
<b>(EMFILE)</b>	Demasiados ficheros abiertos
<b>(ENOTTY)</b>	Ioctl no apropiada para el dispositivo

<b>(ETXTBSY)</b>	Fichero de texto ocupado
<b>(EFBIG)</b>	El fichero es demasiado grande
<b>(ENOSPC)</b>	Espacio agotado en el dispositivo
<b>(ESPIPE)</b>	Búsqueda ilegal
<b>(EROFS)</b>	Fichero de sistema de sólo lectura
<b>(EMLINK)</b>	Demasiados vínculos

La siguiente tabla proporciona una lista con algunos códigos de error de sistema comunes en Windows.

<b>1 (ERROR_INVALID_FUNCTION)</b>	Función incorrecta
<b>2 (ERROR_FILE_NOT_FOUND)</b>	El sistema no puede hallar el fichero especificado
<b>3 (ERROR_PATH_NOT_FOUND)</b>	El sistema no puede hallar la ruta especificada
<b>4 (ERROR_TOO_MANY_OPEN_FILES)</b>	El sistema no puede abrir el fichero
<b>5 (ERROR_ACCESS_DENIED)</b>	Acceso denegado
<b>6 (ERROR_INVALID_HANDLE)</b>	El manejador es inválido
<b>7 (ERROR_ARENA_TRASHED)</b>	Los bloques de control del almacenamiento fueron destruidos
<b>8 (ERROR_NOT_ENOUGH_MEMORY)</b>	No hay suficiente almacenamiento disponible para procesar este comando
<b>9 (ERROR_INVALID_BLOCK)</b>	La dirección del bloque de control de almacenamiento es inválida
<b>10 (ERROR_BAD_ENVIRONMENT)</b>	El entorno es incorrecto
<b>11 (ERROR_BAD_FORMAT)</b>	Se intentó cargar un programa con un



	formato incorrecto
<b>12 (ERROR_INVALID_ACCESS)</b>	El código de acceso es inválido
<b>13 (ERROR_INVALID_DATA)</b>	El dato es inválido
<b>14 (ERROR_OUTOFMEMORY)</b>	No hay suficiente espacio de almacenamiento para completar esta operación
<b>15 (ERROR_INVALID_DRIVE)</b>	El sistema no puede hallar la unidad especificada
<b>16 (ERROR_CURRENT_DIRECTORY)</b>	El directorio no puede eliminarse
<b>17 (ERROR_NOT_SAME_DEVICE)</b>	El sistema no puede mover el fichero a una unidad de disco diferente
<b>18 (ERROR_NO_MORE_FILES)</b>	No hay más ficheros
<b>19 (ERROR_WRITE_PROTECT)</b>	El medio de almacenamiento está protegido contra escritura
<b>20 (ERROR_ERROR_BAD_UNIT)</b>	El sistema no puede hallar el dispositivo especificado
<b>21 (ERROR_NOT_READY)</b>	El dispositivo no está listo
<b>22 (ERROR_BAD_COMMAND)</b>	El dispositivo no reconoce el comando
<b>23 (ERROR_CRC)</b>	Error de datos (verificación de redundancia cíclica)
<b>24 (ERROR_BAD_LENGTH)</b>	El programa emitió un comando pero la longitud del comando es incorrecta
<b>25 (ERROR_SEEK)</b>	La unidad no puede hallar un área o pista específica en el disco
<b>26 (ERROR_NOT_DOS_DISK)</b>	No puede accederse al disco especificado
<b>27 (ERROR_SECTOR_NOT_FOUND)</b>	La unidad no puede hallar el sector solicitado
<b>28 (ERROR_OUT_OF_PAPER)</b>	La impresora no tiene papel

<b>29 (ERROR_WRITE_FAULT)</b>	El sistema no puede escribir desde la unidad especificada
<b>30 (ERROR_READ_FAULT)</b>	El sistema no puede leer desde la unidad especificada
<b>31 (ERROR_GEN_FAILURE)</b>	Un dispositivo conectado al sistema está fuera de funcionamiento
<b>32 (ERROR_SHARING_VIOLATION)</b>	El proceso no puede acceder al fichero porque está en uso por otro proceso
<b>33 (ERROR_LOCK_VIOLATION)</b>	El proceso no puede acceder al fichero porque una parte fue desbloqueada por otro proceso
<b>34 (ERROR_WRONG_DISK)</b>	La unidad contiene el diskette incorrecto. Inserte %2 (Número de Serie de Volumen: %3) en la unidad %1
<b>36 (ERROR_SHARING_BUFFER_EXCEEDED)</b>	Demasiados ficheros abiertos en modo compartido
<b>38 (ERROR_HANDLE_EOF)</b>	Se alcanzó el fin del fichero
<b>39 (ERROR_HANDLE_DISK_FULL)</b>	El disco está lleno
<b>87 (ERROR_INVALID_PARAMETER)</b>	El parámetro es incorrecto (si se obtiene un error en Windows y se ha establecido <b>innodb_file_per_table</b> en my.cnf o my.ini, entonces de agregarse la línea <b>innodb_flush_method=unbuffered</b> en el fichero my.cnf o my.ini
<b>112 (ERROR_DISK_FULL)</b>	El disco está lleno
<b>123 (ERROR_INVALID_NAME)</b>	El nombre de fichero, de directorio, o la etiqueta de volumen, tienen la sintaxis incorrecta
<b>1450 (ERROR_NO_SYSTEM_RESOURCES)</b>	No hay suficientes recursos de sistema para completar el servicio requerido

## **Determinar causas de los errores producidos y posibles soluciones**

Cuando el sistema operativo no carga, los problemas pueden deberse a dos causas principales: a un conflicto de configuración del hardware o a la infección del sistema por parte de un virus.

Los inconvenientes en la configuración del hardware pueden darse tanto en el mismo componente, por mala posición del jumper, como en la declaración del Setup. Ambos casos se detallarán al extremo en los apartados posteriores.

Por otro lado, una vez iniciada la carga del sistema operativo, contaremos con algunas herramientas nativas que podremos utilizar para restaurar y corregir errores del sistema operativo. Asimismo, podemos encontrar escenarios en los cuales la computadora haya sido afectada por virus que corrompieron archivos de arranque del sistema operativo. Estos casos deberemos aplicar las vacunas correspondientes. Finalmente, veremos los pasos por realizar para arreglar el sistema operativo, si esto fuese necesario.

Algunos de los errores producidos más habituales son los que se detallan a continuación.

### **PC se congela**

Si detallamos los problemas que encontramos cuando el equipo está funcionando, seguramente el más habitual será que la PC se cuelga o se congela. Esto puede deberse, principalmente, a dos causas: un conflicto en el sistema operativo o problemas de hardware (disco duro, fuente, alta temperatura en el gabinete, etc.). Para solucionar estos inconvenientes, podremos recurrir a las herramientas del sistema, como iniciarlo en modo a prueba de fallos, restaurarlo mediante las herramientas de asistencia del sistema operativo y, en el último de los casos, formatear nuevamente el disco duro y reinstalar Windows (si es este el sistema operativo que estamos utilizando).

### **PC anda lentamente**

La lentitud en el sistema es una consulta recurrente por parte de los usuarios de computadoras en general. Por un lado, la velocidad con la que avanza la tecnología deja obsoletos los componentes y, por el otro, el software y sus

aplicaciones necesitan cada vez más recursos para su ejecución. Éste es uno de los motivos por los cuales el PC puede funcionar lentamente. Sin embargo, suele suceder que, por fallas en algunos dispositivos o carencia de recursos, la computadora no responda de forma correcta. Además, la lentitud podría ser consecuencia de la incompatibilidad entre los programas instalados y el sistema operativo, o por aplicaciones "malignas" (malware), entre otras causas.

### **Problemas con dispositivos**

En este aspecto, los elementos con inconvenientes pueden ser muchos, desde el micrófono y los altavoces, hasta la unidad de CD, la disquetera, el monitor, etc.

En el caso del monitor, la falla puede darse por la influencia de campos magnéticos que modifican los colores, mientras que la suciedad puede arruinar las unidades de CD/DVD, disqueteras, teclados y mouses. Otras causas pueden ser problemas con los drivers, códecs y configuración.

### **Acceso a redes y a Internet**

En general, los ordenadores poseen una conexión a Internet (o a una red), ya sea para trabajar o, simplemente, para divertirse. Como bien sabemos, para navegar en Internet necesitamos una empresa que nos provea de una conexión (ISP), un módem, que puede ser interno o externo, y un navegador o browser. En todos estos componentes pueden estar la fuente de los problemas que nos impiden conectarnos a Internet.

Además de los elementos físicos comprometidos en la conexión (módem, placa de red), debemos agregar el sistema operativo y muchas de las aplicaciones instaladas en la computadora. Como podemos observar, son numerosos los elementos que intervienen en esta tarea, y en cada uno de ellos puede producirse una falla que nos impida conectarnos adecuadamente a la Red.

### **Fallas en aplicaciones**

Las fallas en las aplicaciones corresponden siempre al nivel del software y, generalmente, tienen que ver con errores ocasionados durante la instalación de la aplicación o modificaciones producidas en el Registro de Windows (seguimos poniendo como ejemplo este sistema operativo). Toda aplicación estará

comprometida con el sistema y, como tal, deberá guardar cierta compatibilidad con él. Además, cada vez que instalemos un programa de cualquier índole, deberemos asegurarnos de que la PC posea los requerimientos mínimos para ejecutarlo.



## RECUERDA

- La depuración de programas es el proceso de identificar y corregir errores de programación. En inglés se le conoce como debugging, que se asemeja a la eliminación de bichos (bugs), manera en que se conoce informalmente a los errores de programación.
- Los nuevos componentes y elementos software del sistema de ERP y CRM se transportan entre los distintos entornos de desarrollo, prueba y explotación, ejecutando procedimientos específicos y realizando su seguimiento, para mantener el flujo de modificaciones y su verificación antes de ser liberadas para su utilización por el usuario final, siguiendo especificaciones técnicas y necesidades de la organización.
- La gestión de incidencias se basa en:
  - Recepción.
  - Resolución.
  - Control.
- El mantenimiento preventivo se basa en:
  - Planificación.
  - Realización.
  - Control.
- La gestión de contratos y elementos se basa en:
  - Inventario.
  - SLA.
- Los procesos realizados, así como las incidencias producidas en la administración, se reflejan en la documentación para mantener registro de seguimiento de los trabajos realizados, siguiendo el procedimiento establecido por la organización.
- Los LOG's, permiten detectar incidentes y comportamientos no habituales. Esta información es muy útil para localizar fallos en las configuraciones de los programas o cambios que se hayan hecho sobre dicha configuración.
- La periodización aplicativa designa la grabación cronológica de

operaciones durante el funcionamiento de la aplicación.

- La periodización sistema designa la grabación cronológica de acontecimientos que sobrevienen a nivel de componentes del sistema.
- Para ver el significado de un número de error en un programa, se utiliza el programa perror, que viene con la distribución de MySQL.
- Algunos de los errores producidos más habituales son:
  - PC se congela.
  - PC anda lentamente.
  - Problemas con dispositivos.
  - Acceso a redes y a Internet.
  - Fallas en aplicaciones.



## Preguntas de Autoevaluación

### 1. Completa el espacio en blanco del siguiente enunciado:

“La \_\_\_\_\_ es el proceso de identificar y corregir errores de programación. En inglés se le conoce como debugging, que se asemeja a la eliminación de bichos (bugs), manera en que se conoce informalmente a los errores de programación”.

- a) Depuración de programas.
- b) Pruebas de validación.
- c) Recepción.

### 2. La Planificación es: (Respuesta múltiple)

- a) La posibilidad de lanzar mantenimientos preventivos de elementos.
- b) La posibilidad de lanzar contratos con diversas periodicidades.
- c) Ninguna de las anteriores es correcta.

### 3. ¿Qué es un LOG'S? (Respuesta múltiple)

- a) Es un registro no oficial de eventos.
- b) Es un término anglosajón, equivalente a la palabra bitácora en español.
- c) Permiten detectar incidentes y comportamientos no habituales.

### 4. ¿Qué es el ERROR\_WRITE\_FAULT?

- a) La unidad no puede hallar el sector solicitado.
- b) La impresora no tiene papel.
- c) El sistema no puede escribir desde la unidad.

**5. Indica si es verdadero o falso el siguiente enunciado:**

*"Cuando el sistema operativo no carga, los problemas pueden deberse a dos causas principales: a un conflicto de configuración del hardware o a la infección del sistema por parte de un virus".*

- a) Verdadero.
- b) Falso.

# Actividades Prácticas



UF1889 Desarrollo de  
Componente Software  
en Sistemas ERP-CRM



## Actividad Práctica RP2

Para la elaboración de esta actividad práctica se atenderá a lo indicado en el Real Decreto 1531/2011, de 31 de octubre, en lo correspondiente a la UC1215\_3 (Realizar y mantener componentes software en un sistema de planificación de recursos empresariales y de gestión de relaciones con clientes).

El ejercicio que se expone a continuación, se corresponde con la realización profesional RP2 (Crear y mantener componentes software para manipular y cargar informaciones en sistemas de planificación de recursos empresariales y de gestión de relaciones con clientes (sistemas ERP-CRM), utilizando los lenguajes y herramientas proporcionadas, siguiendo especificaciones de diseño establecidas).

### Ejercicio

Para la elaboración de este ejercicio práctico debes hacer una breve descripción de los lenguajes que se utilizan en la creación y el mantenimiento de componentes software para la manipulación y la carga de información en los sistemas de planificación de recursos empresariales y de gestión de relaciones con los clientes.

En la descripción, puedes utilizar tanto la información proporcionada en el temario del curso (los lenguajes de programación) como cualquier tipo de información que puedas buscar en Internet relacionada con este tema.

En la corrección del ejercicio se valorará la descripción realizada en función del nivel de comprensión adquirido sobre este aspecto del desarrollo de componente software en sistemas ERP-CRM.



# Respuestas a las Preguntas de Autoevaluación

UNIDAD DIDÁCTICA 1		UNIDAD DIDÁCTICA 2		UNIDAD DIDÁCTICA 3	
1	A	1	A	1	A
2	B	2	B, C	2	A, B
2	B, C	3	B, C	3	B, C
4	B, C	4	A	4	C
5	B	5	A, B	5	B

UNIDAD DIDÁCTICA 4		UNIDAD DIDÁCTICA 5		UNIDAD DIDÁCTICA 6	
1	A	1	A	1	A
2	A, B	2	A	2	A, B
3	B, C	3	B	3	B, C
4	B, C	4	C	4	B, C
5	B	5	A	5	A

UNIDAD DIDÁCTICA 7		UNIDAD DIDÁCTICA 8		UNIDAD DIDÁCTICA 9	
1	A	1	A	1	A
2	A	2	A, B	2	A, B
3	B	3	C	3	B, C
4	C	4	B, C	4	C
5	B	5	B	5	A









