

## **Conceptos básicos.**

Empresa: entidad integrada por capital, trabajo y organización como factores de producción y se dedica a actividades industriales, mercantiles o de prestación de servicios, con fines lucrativos y con la consiguiente responsabilidad.

La función de la empresa es la producción de bienes y servicios para el mercado y la sociedad.

Los sectores productivos:

- Sector primario: explotación de las materias primas.
- Sector secundario: transformación de las materias primas.
- Sector terciario: servicios.

Empresario: sujeto de la unidad productora que ejerce la actividad económica organizando de forma combinada los factores de producción, en función de un fin empresarial preestablecido.

El empresario puede ser una persona física (empresario individual) o una persona jurídica (empresario social).

## **Características generales de la empresa**

No toda actividad organizada tiene el carácter de empresarial, para serlo se requiere que sea económica y profesional.

El empresario individual es la persona física que ejerce en nombre propio por sí o por medio de un representante una actividad constitutiva de empresa.

El empresario social surge cuando 2 o más personas se agrupan construyendo un patrimonio y juntando sus esfuerzos en la consecución de un fin común que es la obtención de beneficios.

Sociedad anónima: sociedad de tipo capitalista de carácter mercantil con capital propio dividido en acciones y bajo el principio de responsabilidad limitada y no personal de los socios por las deudas sociales a la explotación de su objetivo social.

## **Conceptos:**

- Cliente: persona física o jurídica que nos compra de forma habitual los productos objeto de nuestro negocio. (Se considera cliente a aquel al que se le hace una factura). Los consumidores finales son los únicos que son clientes sin ser proveedores.
- Proveedor: persona física o jurídica que nos vende de forma habitual los productos que nuestra empresa vende o sirve de materias primas para la elaboración de nuestras mercancías.
- Representantes: persona física o jurídica que vende mercancía en nombre de una empresa. La relación entre el representante y el cliente se reduce a la venta del producto y a recibir las posibles reclamaciones o a prestar apoyo técnico en su caso.

La empresa vendedora suele pagar un porcentaje a los representantes.

- Mercancía: todos aquellos productos, artículos, objetos de transacción entre diferentes empresas y entre éstas y los consumidores.
- Almacén: lugar físico donde se almacena o se guarda la mercancía, se entiende que esa mercancía está

disponible para servir en cualquier momento.

- Stok: la mercancía que está parada en el almacén, pendiente de venta.
- S. Inicial: la cantidad de mercancía que había cuando comenzó la actividad.
- S. Real: la cantidad de mercancía que hay en cada momento.
- Stok mínimo: la cantidad mínima de mercancía que se debe tener en almacén antes de realizar otro pedido.

Hay que ajustar los stok mínimos para que no haya dinero parado. Esto se ajusta mediante datos estadísticos.

- Pedidos: son órdenes para que la mercancía pase de unos sitios a otros. La forma de realizar los pedidos puede ser por carta, teléfono, mediante impresos establecidos.
- Pedidos a proveedores: son lo que realiza mi empresa a los proveedores, generalmente se utiliza un impresos diseñado por el propio proveedor. Como mínimo en dicho impresos debe figurar:
  - Un detalle completo de quién los pide.
  - La mercancía que pide.
  - El precio estipulado.
  - Los descuentos acordados.
  - Los plazos de entrega establecidos.
- También pueden figurar datos como el tipo de embalaje, destino de la mercancía, dónde se va a enviar la factura.
- Pedidos de clientes: los pedidos que hacen a mi empresa sobre los productos que vendo.
- Confirmación de pedido.
- Recibo de proveedores: es la mercancía que nos envían los proveedores para que la transformemos o vendamos directamente a los clientes.
- Envío a clientes: el cliente nos hace un pedido y cuando le enviamos la mercancía, eso es el envío a clientes.
- Pedidos – envíos a clientes: cuando se hace el pedido y el envío en un solo documento.
- Portes: lo que cobra una empresa o persona física por llevar la mercancía de un sitio a otro.
- Portes debido: cuando paga el porte quien recibe la mercancía.
- Portes pagado: cuando paga el porte quien envía la mercancía.
- Albarán: es un documento que sirve para justificar la salida de géneros del almacén. Generalmente se extiende por duplicado, una copia para la oficina y otra para el almacén.
- Descuento: la cantidad que se rebaja sobre el precio marcado de un producto. Generalmente se expresa en % del valor total.
- Factura: es un documento expresivo de los géneros vendidos o de los servicios prestados y que acredita la compra realizada.

Los requisitos de una factura son:

- Número de serie. La numeración tiene que ser siempre correlativa.
- Los objetos pasivos podrán establecer series diferentes, especialmente si disponen de diversos centros de facturación.
- El nombre, apellidos y razón social, NIF o CIF, domicilio del que expide la factura y de destinatario, la localización del establecimiento permanente.
- Tiene que reflejar todo lo que se factura.
- Contraprestación total de la operación.

- Impuesto tributario que se pague.
- La base imponible sobre la que se aplica el impuesto.
- El neto.
- Lugar y fecha de emisión.
- Factura – albarán: cuando no existe más que un documento que sirve para controlar la salida de la mercancía del almacén y a su vez sirve como documento justificativo del pago.
- Factura proforma: es una factura simulada, un compromiso.
- Notas de abono: es una factura negativa. El proveedor devuelve el dinero al cliente. Los datos de este documento son los mismos que los de la factura.
- Domiciliación del pago: cuando el dinero físico no pasa del cliente al proveedor, sino que interviene un tercero (banco).
- Formas de pago: puede haber pago diferido, a plazos.
- Al contado o en efectivo.
- Las letras o pagarés.
- Las letras de cambio.
- Recibo: es el documento que justifica el pago, éste no es negociable.
- Letra de cambio: es negociable, el banco adelanta el dinero que se va a pagar.
- Impuestos:
  - IVA: impuesto sobre el valor añadido. Lo paga siempre el consumidor final y se aplica en todas las transacciones económicas de la empresa. Es un impuesto directo. Depende de la mercancía objeto de transacción, y es variable.
  - Recargo de equivalencia: es un impuesto que dependiendo del régimen fiscal al que esté sujeto, se paga o no.
  - Base imponible: el neto menos los descuentos.

## TEMA 1: INGENIERÍA DEL SOFTWARE

Todo lo que va desde que surge una idea para hacer un programa hasta que esta idea se deja de utilizar.

### Definiciones:

- Sistema: reunión de objetos, denominados partes, que se relacionan de algún modo.
- Información: todo aquello que reduce la incertidumbre para la toma de decisiones. Para que algo sea información deben cumplirse 3 condiciones:
  - Debe existir la oportunidad de decidir.
  - Debe existir la incertidumbre.
  - El dato debe reducir la incertidumbre.
- Sistema de información: conjunto de mensajes que proveen a una persona de conocimientos a cerca del mundo real. Tienen 2 aspectos:
  - Aspecto interno: el sistema interno es el que registra hechos y acontecimientos del mundo real. También estructuras de datos, reglas y limitaciones del mundo real.
  - Aspecto externo: posibilidad de actuar o consultar los datos memorizados y la posibilidad de cambiar las estructuras, reglas y limitaciones del modelo de información.

- Sistema informático: conjunto lógico de subsistemas y datos necesarios como el soporte de las necesidades de información de 1 más procesos.
- Ingeniería: conjunto de estudios que permiten determinar, para la realización de una obra, las condiciones de rentabilidad óptimas, los materiales y recursos a utilizar y los procedimientos más adecuados.
- Software:
  - Instrucciones (programas de computadora) que cuando se ejecutan suministran la función y comportamiento deseado.
  - Estructuras de datos que facilitan a los programas manipular adecuadamente la información
  - Documentos que describen la operación y uso de los programas.

El software no se estropea.

El software generalmente se construye a medida (específico para cada empresa).

El software tiene una parte ejecutable y otras no, éstas son las que ayudan a la programación, ejecución, mantenimiento.

- Software de sistemas: programa que nos sirve para ayudar a otros programas (sistemas operativos).
- Software de gestión: el que hace contabilidad, nóminas, gestión de ventas, estadísticas. Todo lo relacionado con el sistema productivo.
- Software de ingeniería y científico: programas que resuelven integrales, programas que gestionan redes neuronales.
- Software empotrado: los que vienen en circuitos integrados (lavadoras, coches, autómatas).
- Software de utilidades: es aquel software que facilita la gestión, pero no entra dentro de sistema productivo.
- Software didáctico: aquel que se hace para aprender (enseñanza asistida por ordenador).

### **Elementos claves en la ingeniería de software:**

- Métodos: suministran el cómo construir técnicamente el software. Abarcan un amplio aspecto de tareas, incluyen:
  - Planificación y estimación de proyectos,
  - Análisis de requerimientos del sistema y del software.
  - Diseño de estructura de datos.
  - Arquitectura de escritura y procedimiento.
  - Codificación.
  - Prueba.
  - Mantenimiento.

Con los métodos suele ir un conjunto de criterios para medir la calidad.

### **Las herramientas CASE:**

Las herramientas suministran un soporte automático o semiautomático para los métodos.

Tecnología C A S E Engineering

Software

AIDED

Computer

Tecnología case:

Disciplina automatizada para el desarrollo, mantenimiento y gestión de software.

Incluye:

- Metodologías estructuradas.
- Herramientas automatizadas.

**Procedimientos:**

Los procedimientos de la ingeniería de software son los que unen a los métodos y a las herramientas y facilita un desarrollo racional y oportuno del software.

Los procedimientos definen la secuencia en la que se aplican los métodos, las entregas de documentos que se requieren, los controles que ayudan a asegurar la calidad y las guías que facilitan a los gestores del software a establecer su desarrollo.

**Crisis del software:**

Causas:

- Comunicación imprecisa: a la hora de hacer un programa, hay que saber concretamente lo que el empresario quiere, para que luego el programa no tenga errores.

Hay herramientas y métodos para aminorar los problemas de comunicación.

- Falta de formación y entrenamiento: se hace sin los conocimientos oportunos.
- Resistencia al cambio: se está acostumbrado a programar con un cierto programa, cuesta cambiar de programa aunque en que estemos acostumbrados a programar se haya quedado obsoleto.

**TIPOS DE CICLOS DE VIDA**

**Ciclo de vida:** todos los procesos tanto de construcción como de ejecución que se llevan a cabo en una aplicación, desde que se concibe la idea de realizarla hasta que se deja de utilizar.

Royce (1970).

Clásico Bohem (1981).

Sommerville (1985).

**Convencional** Gigwart (1990).

Prototipo Bohem (1984).

Lehman.

Espiral Bohem (1988).

Warner Jackson (1974).

**Estructurado** Estructurado Gane Sarson (1977).

Ward y Mellor (1985).

Yourdon (1989).

**Orientado a objetos Cluster Mayer (1990).**

Fuente Henderson, Sellers y Edwars (1990).

Remolino Edwars (1990).

Pinball Rumbaugh (1992).

- **CICLO DE VIDA CLÁSICO.**

Solo se da en aquellos sistemas en los que existe.

Es un estudio de condiciones inicial.

Qué es lo que hay que hacer

Cómo hay que hacerlo.

Se hace y se implanta.

Poner lo a funcionar y modificarlo según

las circunstancias.

**Cada fase tiene:** ENTRADA

SALIDA

- **CICLO DE VIDA EN CASACADA:**

- **CICLO DE VIDA POR PROTOTIPOS:**

- MODELO EN CASCADA UTILIZADO EN EL DESARROLLO INCREMENTAL

- MODELO EN ESPIRAL: (fotocopia).

- CICLO DE VIDA ESTRUCTURADO Yuordon (1989). (fotocopia).

Introduce un cierto desorden, a cambio de poder utilizar un mayor número de recursos.

- **MODELO DEL CICLO DE VIDA ORIENTADO A OBEJTOS**

## Agrupamiento n

- MODELO FUENTE: (fotocopias).
- PINBALL.

Eta metodología se asemeja a las máquinas de pinball. Dependiendo de por dónde vaya la bola va cogiendo unos recursos.

Se puede dirigir la bola hacia ciertos sitios, pero es muy difícil prever cuántos puntos se van a conseguir. No es previsible el resultado final.

El factor suerte es importante en este tipo de metodologías.

## METODOLOGÍAS

- Funciones:

### Características deseables de una buena metodología:

- Existencia de reglas predefinidas.
- Cobertura total del ciclo de desarrollo.
- Verificaciones inmediatas.
- Planificación y control.
- Comunicación efectiva.
- Utilización en un abanico amplio de proyectos.
- Fácil formación.
- Soporte de herramientas CASE.
- Disponer de métricas de calidad.
- Soporte al mantenimiento.
- Soporte a la reutilización del software.

Año	Metodología
	<b>Clásica</b>
	<b>Metodologías estructuradas.</b>
1968	Conceptos sobre programación estructurada. Dijkstra.
1974	Técnicas de programación estructurada. Warnier Jackson.
1975	Primeros conceptos sobre diseño estructurado. Myers – Yourdon.
1977	Primeros conceptos sobre análisis estructurado. Gane – Sarson.
1978	Ánálisis estructurado. De Marco y Weinberg.
1978	Nace la metodología oficial francesa. MERISE
1979	Nuevas versiones de la metodología estructurada. Gane – Sarson. Marco
1981	Nace la metodología oficial inglesa. SSADM.
1985	Ánálisis y diseño estructurado para sistemas de tiempo real. Ward Mellor.
1986	Versión 3 de SSADM.
1987	Ánálisis y diseño estructurado para sistemas de tiempo real. Hatley y Pirabay.
1989	Metodología estructurada de YOURDON.
1989	Nace la metodología oficial española. METRICA.
1989	Nace la metodología oficial europea. EUROMÉTODO.

1990	Versión 4 de SSADM.
1993	Nueva versión de Métrica V.2.
1994	Versión operativa de Eurométodo V.0.
1995	Métrica V2.1.
1996	Última versión del Eurométodo.
	<b>Metodologías orientadas a objetos</b>
1991	OMT de James Rumbaugh (Técnicas de modelado de objetos).
1991–1994	OOD de Boom (Ignora el ciclo estructurado).
1992	Synthesis Martin y Odel.
1993	Rood Yamazi Sistemas en tiempo real.
1994	Fusión de Coleman.
1994	MOSES Henderson, Sellers y Edwards
1994	MEDEA. Piattini.

### CICLOS EN LA METODOLOGÍA MERISE.

Esta metodología sale de la universalidad. Siempre que se va a desarrollar una aplicación es como si se estuviese moviendo entre 3 ejes.

Abstracción

Decisión

Vida

### Ciclo de abstracción.

Niveles de abstracción	Tratamientos	Datos
<b>Conceptual</b>	Modelo conceptual: Indica lo que el sistema tiene que resolver.	Modelo conceptual: Trabaja con los objetos, entidades, describe propiedades.
<b>Organizativo</b>	Modelo organizativo: Indica quién lo hace, cuándo y dónde.	Modelo lógico: Transforma el modelo conceptual en estructural, lógicas de datos.
<b>Técnico</b>	Modelo operativo: Indica cómo se hace.	Modelo físico: Se concreta la estructura final en función del gestor a utilizar.

### Ciclo de decisión:

Estaría formado por los analistas y los órganos directivos de la empresa. Son los que plantean el sistema, los que dicen lo qué hay que hacer y toman la decisión de hacer o no hacer el proyecto.

Están todos los recursos humanos con lo que cuenta la empresa. Se recoge una serie de información, para tomar unas decisiones.

Los programadores con un jefe de equipo que coordina los trabajos. Tb. Tiene que haber un representante la empresa a la que le está desarrollando la aplicación.

## METODOLOGÍA SSADM.

Administración y control

Son una serie de reglas que hay que cumplir, no se puede tener iniciativa propia.

## FASES DE MÉTRICA (fotocopia).

## EUROMÉTODO.

No es una metodología desde el punto de vista de la metodología de sistemas. Nace como una necesidad política, que las empresas tengan las mismas oportunidades. Reglas para que los proyectos sean entendibles en cualquier país.

Proceso de Contratación Proceso de Producción Proceso de finalización

### Visión general del Eurométodo.

Guías de eurométodo.

Manuales de conceptos de eurométodo.

### BIBLIOGRAFÍA:

- Ingeniería del software un enfoque práctico. Rogers Pressman. Edit: Mc Graw Hill.
- Ingeniería del software práctico y conciso. Hans Mahuke. Edit: Prentice Hall.
- Análisis estructurado moderno. Edward Yuordon. Edit: Prentice Hall.
- Análisis y diseño detallado de aplicaciones informáticas y de gestión. Mario G. Piattini. Edit: RAMA.
- Revista NovatICA (asociación técnicos de informática). N° 107, 118, 119.
- MERISE: metodología de desarrollo de sistemas (teoría aplicada). Jean Patrick Matheron. Edit: Paraninfo.
- MÉTRICA V2. Ministerio de administraciones públicas (MAP).
- Análisis y diseño detallado de aplicaciones informáticas y de gestión. Gregorio Cabrera, Guillermo Montoya. Edit: Mc Graw Hill.

## ENTREVISTA CON EL USUARIO

### Objetivos:

- Conseguir información sobre los requisitos del usuario.
- Conseguir información sobre el funcionamiento del sistema actual.
- Conseguir información sobre la organización de esa empresa.
- Responsabilidades y funciones de cada uno.

### Recomendaciones para el desarrollo de la entrevista:

- Preparación de los guiones: es aconsejable antes de realizar una entrevista, mandar al usuario un guión con lo punto que se van a tratar.

Guión para un:

Responsable de área:

- Funciones que realiza su departamento o grupo.
- Ver los sistemas de información actuales ya sean manuales o mecanizados.
- Relaciones o salidas o otros sistemas, definiendo destino de datos, medios empleados para la transmisión de datos, frecuencia con la que se mueven los datos.
- Nivel de satisfacción técnica con el sistema actual.

Se valorarán de forma global los siguientes aspectos:

PREGUNTA	VALORACIÓN.
Disponibilidad de los sistemas de información.	1-2-3-4-5
Tiempos de respuesta.	1-2-3-4-5
Facilidad de uso	1-2-3-4-5
Etc.	1-2-3-4-5

Conocimiento del resto de los usuarios:

Se suele utilizar una entrevista, hay que buscar los aspectos más funcionales del funcionamiento de la empresa.

Guión:

- Situación actual.
- Situación de los sistemas de información utilizados, incluyendo:
  - Entorno físico.
  - Tipos de entradas:
  - Origen (quién genera el dato).
  - Datos involucrados.
  - Soporte utilizado.
  - Frecuencia.
  - Procesos y funciones realizadas por dichos sistemas.
  - Tipos de salidas:
  - Destino.
  - Datos involucrados.
  - Soporte.
  - Frecuencia.
- Volumen de información manipulada.
- Conocer el almacenamiento de datos involucrados.
- Conocer la importancia de cada dato.
- Ventajas e inconvenientes de que existe que tiene la existencia de este departamento.
- Nivel de satisfacción técnica con los sistemas actuales (tablas).
- Conocer que requisitos esperarían de un sistema nuevo (Cómo proteger los datos).
- Conocer los factores críticos, es decir, conocer los momentos en los que pueden haber fallos.
- Consolidación de la entrevista: decirle al usuario lo que yo he entendido. El prototipo es un buen método para consolidar una entrevista.

## **Paso 1 Paso 2**

## **Paso 3 Paso 4**

**Diagramas de alto nivel:** explicación gráfica del funcionamiento de una empresa. Sirve para comunicarnos con el resto de los diseñadores y con el cliente.

**Diagrama de GANTT:** distribución de tareas en tiempos (cuánto tiempo se tarda en hacer cada tarea).

Tiempos

Tareas

**Diagrama de PERT:** el orden en el que deben realizarse esas tareas.

**A B**

En el punto en el que estamos, sabemos a grandes rasgos, cuánto me va costar hacer el programa.

## **Estudio de viabilidad técnica y económica del programa.**

Técnica:

- Parámetro: importancia que ese concepto tiene para esa empresa.
- Nota: valoración que se le da al programa respecto al parámetro.
- Ponderación total, es la suma de (Parámetro \* Nota).

Económica:

- Costes de implantación.
  - C. Centro de desarrollo.
  - C. Puesta en marcha.
  - C. Formación.
  - Total costes de implantación.
- Costes tecnología.
  - C. Hardware.
  - C. Software.
  - C. Comunicaciones.
- Costes operacionales.
  - C. Centro de proceso.
  - C. Mantenimiento.
  - Total costes operacionales.

## **Fronteras de mecanización:**

Hay que tratar de cerrar el programa de algún modo, diciendo que el programa no contemplará lo que no quede especificado. Todo aquello que no quede especificado en el programa, el programa no lo hace.

## TEMA 5: LOS DATOS.

### Fase de análisis:

- Crear datos.
- Crear procesos.
- Crear diccionario de datos.

Los datos y los procesos no se estudian simultáneamente. Sin embargo el diccionario de datos se va creando junto con los datos.

Hay que optimizar los datos, es decir, utilizarlos únicamente cuando sean necesarios.

**Fichero:** en los sistemas informáticos tradicionales los datos se almacenan en ficheros diseñados especialmente para cada aplicación. Aunque varias aplicaciones utilizaran los mismos datos, no los compartían, cada aplicación tenía un fichero diferente. Esto trae problemas a la hora de modificar los datos de los ficheros. La solución a este problema son las bases de datos.

En las bases de datos se almacenan los datos con el objetivo de que sirvan en distintas aplicaciones. No es preciso que existan redundancias. Hay ocasiones que los datos se tienen duplicados (solamente cuando sea necesario).

Problemas de utilizar ficheros:

- Redundancia de información.
- Más memoria ocupada.
- El tiempo de ejecución es mayor.

**Bases de datos:** sistemas orientados a datos, porque lo importante, son los datos, en función de éstos se crean los procesos.

En una base de datos se guarda la descripción y las relaciones que pueden existir entre los distintos datos.

Los datos se almacenan interrelacionados y estructurados de acuerdo con un determinado modelo de datos que permita recoger el máximo significado asociado a ellos.

En un sistema de base de datos los programas no acceden directamente a los datos sino que es un software el encargado de proporcionar a las aplicaciones, los datos que éstas demandan. Este software se conoce como Sistema Gestor de Base de Datos (SGBD).

También mantiene la integridad, confidencialidad y seguridad de los datos.

- Ventajas de las bases de datos frente a los ficheros:

- Independencia de los datos y procesos que los utilizan.
- Coherencia de los resultados obtenidos.
- Mejor disponibilidad de los datos.
- Mayor valor informativo.
- Mejor y más normalizada documentación de la información.

**Sistema Gestor de Bases de Datos:** conjunto de programas que permiten que los usuarios describan, recuperen y manipulen eficazmente los datos almacenados en la base de datos, protegiendo dichos datos

contra todas aquellas acciones intencionadas o no que los puedan corromper.

Se necesita un software de gestión que facilite las operaciones y las interfaces con los usuarios. Este software es el que conocemos como SGBD.

Pero también se necesita un modelo de datos de acuerdo con el cual se estructurará y manipulará los datos.

El SGBD implementa un determinado modelo de datos.

Se distinguen 3 tipos de usuarios de una base de datos:

- El administrador: la persona que define los datos.
- Los programadores: las personas que crean la forma de presentar los datos.
- Los usuarios finales: las personas que utilizan los datos.

Los datos se pueden ver desde dos puntos de vista:

- Ver una estructura lógica: que es lo que yo veo hacia fuera.
  - Nivel conceptual.
  - Nivel externo.
- Una estructura física: cómo están los datos grabados en el ordenador.

### **Funciones del sistema gestor de bases de datos.**

- **Descripción de datos:** un SGDB debe permitir la descripción de los distintos esquemas de las bases de datos. El administrador de la base de datos es el encargado de especificar los elementos de datos que la forman; su estructura, interrelaciones, restricciones, etc.

El administrador es el responsable de definir tanto el esquema conceptual, como interno y los distintos esquemas externos.

- **Manipulación de datos:** debe permitir que los usuarios de la base de datos recuperen, inserten, modifiquen o eliminen datos de ella. Las operaciones a realizar sobre la información contenida en la base de datos se especifican mediante un lenguaje de manipulación de datos (LMD). Lenguaje de manipulación que está formado por un conjunto de instrucciones que son admitidas dentro de un programa escrito en otro lenguaje.

Los usuarios finales interactúan con la base de datos a través de aplicaciones diseñadas para atender sus necesidades concretas de información, o por medio de utilidades pensadas específicamente para usuarios no informáticos.

Cuando se trata de satisfacer las necesidades de información que no fueron formalizadas en una aplicación en el SGBD suele disponer de un lenguaje autocontenido (que está dentro del sistema) que se utiliza de forma convencional que incluye algunas facilidades de descripción. El SGBD llevará a cabo las operaciones de manipulación de datos, cumpliendo siempre las restricciones de seguridad especificadas por el administrador.

- **Transformación de los datos: (MAPPING).**

Es sistema general de bases de datos es el responsable de realizar la correspondencia o transformación entre los formatos de datos, correspondientes a los distintos esquemas de la base de datos.

- **Protección de los datos.**

Debe tener mecanismos que permitan proteger los datos contra fallos que los altere y corrompa. También debe tener mecanismos que impida los accesos no autorizados.

Seguridad: mecanismos que nos protegen de cometer errores lógicos.

- **Integridad.**

El objetivo de la integridad es lograr que todos los datos contenidos en la base de datos sean correctos, para ello es sistema general de bases de datos debe de detectar y corregir las operaciones incorrectas que introduzcan inconsistencia a la base de datos.

Hay 2 tipos de operaciones que pueden afectar a la integridad:

- Las operaciones que atentan contra las restricciones de integridad
- Las interferencias por accesos concurrentes.

Una restricción de integridad debe de ir siempre acompañada de la especificación de la acción que va a realizar, en caso de intento de violación de la misma.

El sistema general de base de datos debe disponer de utilidades o programas de servicio, diseñados para facilitar al administrador de la base de datos las tareas de mantenimiento y seguridad de dicha base. Estos programas realizan una serie de funciones:

- Modificación del tamaño de los ficheros.
  - Obtención de estadísticas de utilización.
  - Copias de seguridad y recuperaciones tras la caída del sistema.
- 
- **Afinación:** es la posibilidad de crear nuevos objetos en la base de datos, como informes, formularios, sin la necesidad de volver a escribir los datos. Debe permitir, así mismo, la posibilidad de importar y exportar datos.
  - **Flexibilidad:** posibilidad que debe tener un sistema general de base de datos para crear informes, consultas, etc., mediante lenguajes de alto nivel, o herramientas propias de la base de datos, que interroguen a cualquier estructura de datos, de forma que sea sencillo y rápido de obtener la información deseada.
  - **Migración y compatibilidad de datos:** la migración de los datos, es la posibilidad que tiene las bases de datos de ser capaces de obtener datos que han sido almacenados en otro tipo de aplicaciones. La compatibilidad se refiere a que los datos pueden ser copiados a otras aplicaciones sin pérdida de información, ni de sus propiedades.
  - **Redundancia mínima:** posibilidad de que los datos no estén repetidos, ni en esa base ni en otras con las que esté relacionadas.
  - **Velocidad de respuesta:** las consultas han de tener un tiempo de respuesta lo suficientemente rápido como para atender todas las solicitudes de información, que le hagan los usuarios.
  - **Independencia física de los datos:** propiedad del sistema general de base de datos que hace que el sistema siga funcionando aun cambiando la ubicación física de los datos e incluso cambiando el sistema operativo que los soporta, sin la necesidad de cambiar ningún programa de acceso.
  - **Independencia lógica de los datos:** se refiere a que las modificaciones que se realicen en algunos elementos de la base de datos no obliguen a cambiar su estructura general.

Debe disponer de una interfaz de alto nivel con los programadores, se refiere a que internamente la base de datos debe disponer del software necesario para desarrollar consultas (visitas), formularios e informes, sin

necesidad de conocer la organización interna, ni la unidad en la que se archivan los datos.

La base de datos tiene que estar dotada de un diccionario, debe disponer de mecanismos que permitan el almacenamiento de las definiciones de los distintos esquemas de la base de datos y de las correspondencias entre ellas.

## **MODELO DE DATOS.**

Son un conjunto de conceptos, reglas y convenciones que se aplican a una parte del mundo real con el objetivo de que dicha representación pueda ser utilizada por distintas aplicaciones actuales o futuras.

- Nivel conceptual (modelo basado en objetos): es independiente de las máquinas y del usuario; y lo que pretende es obtener toda la información necesaria para el sistema, representar la información sin pérdida de significado, independizar los datos de los recursos físicos y de los requerimientos de usuario; proporcionar a los usuarios la información tal y como la necesitan.

Los modelos más comunes que describen los datos de forma conceptual son:

- Modelo entidad – relación.
- Modelo binario.
- Modelo semántico de datos.
- Modelo infológico.
- Nivel externo (modelo basado en registros): sus objetivos son especificar la estructura lógica, general de la base de datos.
  - Modelo relacional.
  - Modelo en red.
  - Modelo jerárquico.

### **Modelo entidad – relación.**

Se basa en la existencia de objetos, a los que se les da el nombre de **entidades** y las asociaciones entre esas entidades se les llama **relaciones**.

- Las características principales del modelo entidad – relación son:
  - Muestran todos los datos independientemente de su utilidad posterior y del sistema sobre el que se va a implantar.
  - Permite crecer y modificarse a medida que se necesita.
  - No tiene en cuenta las limitaciones de almacenamiento y la velocidad de proceso.
- Términos básicos del modelo entidad – relación.

**Entidad:** es cualquier objeto (persona, animal o cosa) que tiene interés para la empresa. Normalmente se nombra con un sustantivo y se representa mediante un rectángulo.

- Entidad fuerte: aquella que no depende de otra entidad para su existencia.
- Entidad débil: aquella que depende de otra entidad para su existencia. Se suele representar con un rectángulo de doble borde.

**Registro:** es el conjunto de atributos que se refieren a uno de los elementos que forman la entidad.

**Atributo:** es la información mínima que se puede tener de cualquier objeto. Se representan mediante una elipse y la clase mediante un cuadrado.

**Superclave (identificador):** es el conjunto de atributos que identifican de forma única a cada registro.

**Clave candidata:** es una superclave que sea mínima.

**Clave primaria o principal:** es una de las claves candidatas que selecciona el diseñador y que al menos debe tener 3 características:

- No contener valores nulos.
- Debe ser sencilla de crear y conocida.
- No ha de variar con el tiempo.

**Clave alternativa:** es una clave candidata no seleccionada.

**Clave ajena o foránea:** son los atributos de una entidad, que son clave primaria en otra entidad.

**Clave artificial:** es una clave primaria que ha sido creada independientemente del contenido de los atributos, que pudieran formar la clave primaria. (Pueden ser la unión de 2 atributos).

**Relación:** es una asociación entre entidades. Si no existen entidades no puede haber relación. Las relaciones no tienen atributos. Se utiliza un verbo en infinitivo para darle un nombre y se representa mediante un rombo.

**Grado de una relación:** es la participación de cada entidad en la relación. Está basada en la teoría de los conjuntos.

- Relación 1 a N: si un elemento del conjunto A está relacionado con varios elementos del conjunto B.
- Relación 1 a 1.
- Relación N a N.

**Cardinalidad de una relación:** sirve para conocer el grado máximo de participación de la entidad en la relación. La notación utilizada es:

1 Todos los elementos de mi entidad están relacionados 1 a 1 con todos los elementos de la otra entidad.

C Cuando los elementos de mi entidad están relacionados con uno de los de la siguiente, pero puede haber elementos que no estén relacionados.

M Cuando todos los elementos de mi entidad están relacionados con uno o varios de la entidad relacionada.

N en mi entidad pueden quedar elementos sin relación.

### **Modelo relacional.**

Se utiliza para especificar la estructura lógica general de la base de datos. En este modelo, las entidades se representan con tablas.

Una base de datos relacional está formada por un conjunto de tablas que se relacionan entre ellas y que se corresponden con las entidades y relaciones del modelo entidad – relación.

**Tabla:** es una estructura formada por filas y columnas que sirve para organizar los datos de una entidad. A las

filas se les llama TUPLAS, a cada fila le corresponde un registro. Las columnas son los atributos de la entidad y se corresponden con los campos.

Una tabla relacional debe cumplir con las siguientes condiciones:

- Tener un solo tipo de fila. Que las filas tengan todas las mismas columnas.
- Las columnas deben ser únicas, no pudiendo existir dos columnas con el mismo nombre.

Nombre

Atributos

Atributos

Atributos

Atributos

Las relaciones entre entidades se expresan mediante una línea.

**Dominio:** el conjunto de todos los valores posibles para una columna. Hay dominios continuos, discretos y de tipo general:

- D. Continuos: contienen todos los valores posibles entre 2 valores determinados.
- D. Discreto: sólo puede tomar unos valores prediseñados.
- D. Tipo general: puede tomar cualquier valor.

### Diagramas de flujo de datos

Diagrama de flujo de datos son una herramienta de modelización que nos permite representar gráficamente un sistema como una red de minisistemas conectados entre sí por rutas de datos.

Los dfd son fundamentalmente gráficos particionados, multidimensionales, centrados en los flujos de datos, almacenes y entidades externas. No muestran composición de los flujos de datos. Tampoco aparecen relaciones de acceso, ni representan decisiones. No representa cálculos, ni expresa cantidades.

#### Elementos de un dfd.

- **Procesos:** también se conoce con el nombre de burbuja, función o transformación. El proceso muestra una parte del sistema que transforma entradas en salidas.

Se representa gráficamente mediante un círculo (también se puede representar mediante un óvalo o rectángulo con las esquinas redondeadas).

Siempre que haya que representar un dato que se transforma, tiene que haber un proceso.

Para identificar un proceso dentro del círculo tiene que llevar un número y un nombre.

- **Flujos de datos:** el flujo de datos se representa mediante una flecha. La punta indica la dirección de los datos.

Los flujos de datos tiene que llevar un nombre significativo, no podrán existir 2 flujos de datos con el mismo nombre, a no ser que represente el mismo.

Si el flujo de datos entre o sale de un almacenamiento y la estructura es la misma que la del almacenamiento éste no se nombra.

- **Almacenamiento:** se usan para describir donde se almacenan los datos, utilizados por varios procesos. Se representa mediante dos rectas paralelas, con un nombre único, y tiene que aparecer descrito en el diccionario de datos.

### Nombre

- **Entidad externa:** es una persona, un organismo que reside fuera del entorno del sistema y que recibe o envía datos al mismo.

Tiene 3 características:

- Son exteriores al sistema que estamos diseñando. Sólo se pueden unir la entidad externa y mi sistema a través de un flujo de datos.
- Ni el analista, ni el programador están en disposición de cambiar el contenido de una entidad externa, ni la forma en la que ésta trabaja.
- Las relaciones existentes entre entidades externas no aparecen nunca en un dfd.

Hay 2 métodos para hacer un dfd:

- Conectividad.
- Rompecabezas.

**Conectividad:** consiste en representar la aplicación como un solo proceso, nº 0 y nombre: proyecto o empresa.

Se dibujan todas las entidades externas que estén relacionadas con el sistema.

Y los flujos de datos que entran se pone a la izquierda y los que salen a la derecha.

### Rompecabezas:

- Se representan los diferentes subsistemas.
- Definir los almacenes de datos que comparten estos procesos.
- Flujos de datos que relacionan. También se dibujan los flujos de datos que relacionan unos procesos con otros.
- Definir las entradas y salidas de cada uno de los subsistemas.

### Reglas para construir un dfd:

- No existen conexiones entre entidades externas (diagrama de contexto).
- Una entidad externa no puede mandar, no recibir datos de una almacén.
- No se pueden relacionar almacenes entre sí. Siempre tiene que ser a través de un proceso.
- Un proceso siempre tiene que tener alguna entrada y alguna salida.
- Todo proceso nuevo que se cree tiene que tener un nombre significativo, un número que indique el nivel en el que se encuentra. Tanto el nombre como los números deben de ser únicos.
- Los flujos de datos deben de tener nombres significativos a excepción de los que entran o salgan de un

almacén y su estructura coincide con la de estos.

- Los almacenes deberán estar nombrados de forma única y sólo aparecerá si hay más de un proceso representado que los utilice.
- Se debe mantener siempre la consistencia entre los distintos niveles. Esto es, cada burbuja debe tener exactamente el mismo flujo de datos que el diagrama que esté por debajo de él.
- Para establecer las equivalencias entre los padres y los hijos se utiliza el diccionario de datos.

## **Diccionario de datos**

Es una lista organizada de todos los elementos significativos para el sistema; con definiciones precisas y rigurosas para usuarios y analistas, que permiten una total comprensión de todas las entradas, salidas, composición de los almacenamientos y cálculos intermedios.

En el diccionario de datos se hacen definiciones rigurosas de:

- Datos elementales.
- Estructuras de datos.
- Almacenamientos.
- Flujos de datos.
- Procesos.
- Entidades externas.

**Datos elementales:** tiene que figurar:

- Nombre.
- Descripción.
- Alias.
- Datos con los que está relacionado.
- El dominio (posibles valores que puede tomar).
- Cualquier otra información sobre su validación.

**Estructuras de Datos:** está formada por datos elementales o por otras estructuras de datos y/o por combinación de ambas. Al menos deberá contener 2 datos elementales. Estos datos pueden agruparse o repetirse un número indeterminado de veces.

Representación gráfica:

Estructura = estructuras de datos que la componen, separadas por comas, signo +. Todo lo que va entre paréntesis significa que es opcional. Los corchetes se ponen cuando dentro de las opciones hay que elegir una y en vez del + se ponen :

\*\* ..... comentario a la estructura de datos.

@ delante de un dato elemental significa que ese dato es clave.

Ejemplo: Estructura = @ pepe +[Luis:Alfonso] \*\* .....

## **Almacenes:**

- Nombre.
- Descripción
- Los flujos de entrada y salida que tenga cualquiera de los diagramas.

- Estructura de datos que lo integra.

### **Flujos de datos:**

- Nombre.
- Descripción.
- Estructuras de datos que lo componen.
- Origen y destino del flujo.

### **Procesos:**

- Nombre.
- Nº.
- Descripción.
- Flujos de datos que entran.
- Flujos de datos que salen.

### **Entidades externas:**

- Nombre.
- Descripción.
- Los flujos de entrada y salida que tenga en cualquiera de los diagramas.

**Miniespecificaciones:** es una técnica que define las acciones que realiza un proceso primitivo. Debe describir de una manera más o menos formal cómo se obtienen los flujos de datos de salida, a partir de los flujos de datos de entrada, más una información local del proceso.

### **Técnicas para describir un proceso:**

- Lenguaje estructurado: lenguaje formado por un subconjunto de palabras del idioma elegido, de las que unas se utilizan para formar construcciones propias de la programación estructurada, y otras incluyen un conjunto de verbos que indican acción.

Análisis y diseño de aplicaciones informáticas de gestión.

1

37

## **ANÁLISIS PREVIO**

## **ANÁLISIS FUNCIONAL**

## **ANÁLISIS ORGÁNICO**

## **PROGRAMACIÓN**

**Y**

## **PRUEBAS**

## **IMPLANTACIÓN**

Y

**MANTENIMIENTO**

ACTORES

PROCESOS

HERRAMIENTAS

DOCUMENTACIÓN

**Análisis y requisitos del sistema**

**Análisis y requisitos del software**

**Diseño preliminar**

**Diseño detallado**

**Codificación y pruebas**

**Explotación y mantenimiento**

**Explotación y mantenimiento**

**Codificación y pruebas**

**Diseño detallado**

**Explotación y mantenimiento**

**Codificación y pruebas**

**Diseño detallado**

**Diseño preliminar**

**Análisis y requisitos del software**

**Análisis y requisitos del sistema**

3.2

Modelo de entregable.

3.1

Modelo de transacción.

2.5

Caso–ejemplo

2.4.

Guía de conexión con los métodos

2.3.

Guía de planificación de entregas

2.2.

Guía proveedor

2.1.

Guía cliente

Adaptación del sistema informático

- Aprobación de entregables finales.
- Finalización del contrato.
- Aprobación de los entregables.
- Aprob. del estado y el plan del proyecto.
- Control de cambios en el contrato.
- Petición de ofertas.
- Ofertas.
- Selección del proveedor.
- Adjudicación del contrato.

Producción

Desarrollo

Especificación

Diseño Realización

Validación Generación

Validación Generación

Diseño Realización

Especificación

Validación Generación

Diseño Realización

Especificación

Construcción y pruebas.

Diseño físico

Estudio completo

Especificación lógica del sistema.

Especificación de requisitos

Análisis de requisitos

Estudio de viabilidad

### **Planificación estratégica**

#### **Sistema operativo**

#### **Sistema de información**

#### **Sistema de pilotaje**

### COMO DIVIDIR EL PROYECTO EN ETAPAS

ETAPA 1

ETAPA 2

ETAPA 3

...

Salidas que hay y cuándo se producen

Restricciones

Documentación

Herramientas

Gestión control del proyecto

y

Documentaciones globales

3.1

Modelo de estrategia.

4.

Diccionario de eurométodo.

**3**

**2**

**1**

**5**

**4**

**3**

**2**

**1**

Definir la arquitectura de información.

Definir las clases de datos.

Definir los procesos de la empresa.

Definir los objetivos de la empresa.

Cliente

Recibir

Factura

La clave principal se subraya o se pone un \*

**Nº**

**Nombre**

Nivel en el que se encuentra el proceso.

Identifica el proceso en el diccionario de datos. El nombre debe ser único y significativo

Graba

Lee

**0**

**Nombre**

DIAGRAMA DE CONTEXTO

**Nombre**