

Temas Generales para la preparación de la Oposición al Cuerpo Superior de Estadísticos del Estado.

**Cuerpo Superior de Estadísticos del Estado
Especialidad de Estadística-Ciencia de Datos.**

Almacenamiento y modelos de datos

Tema 5. La planificación y diseño de base de datos.

AUTOR: Antonio J. Sánchez-Padial

**Asociación Profesional de Cuerpos Superiores de Sistemas y
Tecnologías de la Información de las Administraciones Públicas**

Creación: Junio 2021

ÍNDICE

1	INTRODUCCIÓN	4
2	EL CICLO VITAL DE LOS SISTEMAS DE INFORMACIÓN...	5
3	EL CICLO VITAL DE DESARROLLO DE UN SISTEMA DE BASES DE DATOS	6
4	PLANIFICACIÓN DE BASES DE DATOS.....	7
5	ALCANCE DEL SISTEMA.....	8
o	VISTAS DE USUARIOS.....	8
6	ANÁLISIS Y RECOLECCIÓN DE REQUISITOS	9
▪	LA SOLUCIÓN CENTRALIZADA	9
▪	LA SOLUCIÓN DE INTEGRACIÓN DE VISTAS	9
7	DISEÑO DE BASES DE DATOS.....	10
o	APROXIMACIONES AL DISEÑO DE BASES DE DATOS.....	10
o	MODELADO DE DATOS	10
o	FASES DEL DISEÑO DE LA BASE DE DATOS	11
▪	DISEÑO CONCEPTUAL DE LA BASE DE DATOS.....	11
▪	DISEÑO LÓGICO DE LA BASE DE DATOS	11
▪	DISEÑO FÍSICO DE LA BASE DE DATOS	11
8	SELECCIÓN DEL SISTEMA DE GESTIÓN DE BASES DE DATOS	13
-	DEFINIR EL ALCANCE DEL ANÁLISIS	13
-	PRESELECCIÓN DE DOS O TRES PRODUCTOS	13
-	EVALUAR PRODUCTOS.....	13
-	INFORME FINAL Y RECOMENDACIONES	13
9	DISEÑO DE LA APLICACIÓN.....	15
o	DISEÑO DE TRANSACCIONES	15
o	DISEÑO DE LA INTERFAZ DE USUARIO.....	15
10	PROTOTIPOS.....	17
11	IMPLEMENTACIÓN.....	18
12	CONVERSIÓN Y CARGA DE DATOS.....	19
13	TESTEO.....	20
14	RESUMEN ESQUEMÁTICO.....	21

15	GLOSARIO	22
	Z BIBLIOGRAFÍA BÁSICA	23

1 Introducción

La **crisis del software** es un término que engloba la complejidad intrínseca y la dificultad del desarrollo de software que conlleva a retrasos en los proyectos, presupuestos que exceden con creces lo previsto, bajo rendimiento de los sistemas, errores y bugs, sistemas difíciles de mantener, en definitiva, merman la confianza en el software desarrollado. Aunque el término fue acuñado en los años 60, sigue muy vigente.

Algunos de los motivos que provocan el fracaso de los proyectos software son:

- falta de unos requisitos correctamente especificados,
- falta de una metodología de desarrollo adecuada,
- diseño pobremente descompuesto en componentes mantenibles

Como respuesta a este desafío se propuso el Ciclo de Vida de los Sistemas de Información (ISLC, Information Systems Lyfecycle). Cuando el sistema de información consiste principalmente en una base de datos se usa el término Ciclo de Vida del Desarrollo de Bases de Datos.

2 El ciclo vital de los sistemas de información

Desde la aparición de las bases de datos relacionales en los primeros años 70, los SGBD han sustituido a los sistemas de ficheros en el papel de infraestructura principal de los Sistemas de Información de las organizaciones. Al mismo tiempo, la importancia de los datos ha crecido en todos los sectores, siendo en la actualidad un activo de primer orden para cualquier organización.

Hoy día, un sistema de información incluirá una base de datos, software de base de datos, software de aplicación, el soporte físico o hardware, así como el personal que usa y desarrolla este sistema de información.

La base de datos juega un papel central en el sistema de información, y tanto su desarrollo como el uso que se hace de ella, debe estar alineado con los requisitos y centrales más básicos de la organización. Por este motivo, además, el ciclo de vida del sistema de información corre paralelo al ciclo de vida de la base de datos. Las etapas típicas en este ciclo de vida son: planificación, recogida de requisitos, diseño, prototipo, implementación, testeo, migración y mantenimiento. Aunque en este tema se aborda este ciclo de vida desde el punto de vista del desarrollo de una base de datos, es necesario tener en cuenta que siempre será necesario poner este desarrollo en perspectiva del ciclo de vida del sistema de información al que está ligada la base de datos.

3 El ciclo vital de desarrollo de un sistema de bases de datos

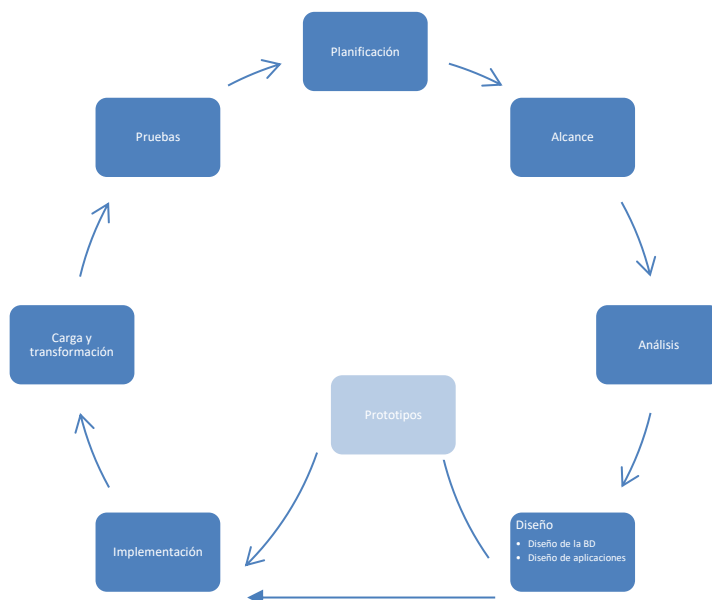
Ya hemos visto como el ciclo de vida del desarrollo de una base de datos va siempre ligado a la perspectiva más amplia del sistema de información central de la organización.

Se describen a continuación las siguientes etapas:

- Planificación de la base de datos
- Definición del sistema
- Recogida de requisitos y análisis
- Diseño conceptual de la base de datos
- Selección del SGBD
- Diseño lógico de la base de datos
- Diseño de la/s aplicación/es
- Diseño físico de la base de datos
- Prototipado (opcional)
- Implementación
- Migración y carga de datos
- Pruebas
- Mantenimiento operativo

Nótese que estas etapas no siguen un orden cronológico estricto, si no que el desarrollo de las etapas posteriores provoca descubrimientos y decisiones sobre el sistema, que puede hacer replantear las etapas previas en retroalimentación. El esquema adjunto muestra solo algunas de estas retroalimentaciones, las más frecuentes.

Para las bases de datos pequeñas, con un pequeño número de usuarios, el ciclo de vida no tiene porque ser demasiado complejo. Sin embargo, para las bases de datos de tamaño mediano o grande con cientos o miles de usuarios, que usan cientos de consultas y aplicaciones distintas, el ciclo de vida puede resultar bastante complejo. Se presentan a continuación los detalles de cada una de las etapas , en el caso de uno de esos sistemas de información de tamaño mediano o grande.



4 Planificación de bases de datos

La planificación es el conjunto de actividades de base que facilita el desarrollo del resto de etapas del ciclo de vida. Es una actividad clave, que debe realizarse en el contexto de la estrategia general de sistemas de información de la organización. Esta estrategia debe abordar los siguientes puntos:

- Determinar las necesidades de información asociadas a los objetivos y actividades estratégicas de la organización,
- Análisis de las debilidades y fortalezas de los sistemas de información actuales,
- Prospectiva de los avances en tecnologías de la información que pueden proporcionar ventajas a la organización

El primer paso en la planificación de una base de datos es determinar claramente su **misión**. En la declaración de esta misión se establecen los objetivos principales de la base de datos. Esta misión no puede ser definida por el área de desarrollo o de sistemas, si no que debe ser articulada por los responsables del sistema de información en particular. La declaración de la misión facilita comprender el fin de la base de datos y ayuda a determinar una hoja de ruta para la creación de la base de datos. Una vez definida esta misión, el siguiente paso será identificar los **objetivos de la misión**. Cada uno de estos objetivos identifica una actividad que hará uso de la base de datos. De este modo, si la base de datos satisface los objetivos de la misión, podremos asegurar que cumple con su misión. Junto a la misión de la base de datos y sus objetivos pueden acompañarse con información adicional describiendo en términos generales el proyecto que se desea realizar, los recursos que se utilizarán para ello, y el presupuesto disponible o necesario.

Además, la planificación de la base de datos debería especificar los estándares que deberán cumplir los datos incorporados. Establecer estándares puede ser una tarea muy demandante tanto al definirlos como en su mantenimiento. Sin embargo, proporcionan la documentación necesario para la formación inicial del personal y para definir controles de calidad, de modo que las aportaciones a la base de datos son homogéneas. Por ejemplo, se podrían definir reglas para indicar cómo deben ser nombrados los elementos en el diccionario de datos, lo que servirá para prevenir redundancias e inconsistencias.

Por último, cualquier consideración legal o corporativa que afecta a los datos debería ser incluida en la planificación de la base de datos, por ejemplo consideraciones sobre el tratamiento de datos personales, o sobre el nivel de confidencialidad de determinada información.

5 Alcance del sistema

En esta fase se fija el alcance y los límites de la base de datos. Antes de comenzar el diseño de la base de datos, es necesario identificar los límites del sistema que se va a desarrollar y cuáles serán sus interfaces con otros sistemas de la organización. No solo se deben tener en cuenta los usuarios y sus aplicaciones actuales, si no también aquellos que se puedan prever para el futuro. En esta definición del alcance y de los límites también hay que incluir las vistas de usuarios que ofrecerá la base de datos.

○ Vistas de usuarios

Una base de datos puede tener una o varias vistas de usuarios. Identificarlas es clave en las etapas iniciales porque de este modo podremos garantizar que estamos teniendo en cuenta a todos los usuarios principales del sistema. Las vistas de usuarios son especialmente útiles en el desarrollo de bases de datos de relativa complejidad, al permitir descomponer los requisitos en bloques más fáciles de manejar.

Cada vista de usuario define las acciones que ese usuario será capaz de realizar en la base de datos, tanto datos que debe poder consultar, como transacciones que ejecutará. Los requisitos de la vista de cada usuario pueden ser distintos o compartidos con los de otra vista de usuario.

6 Análisis y recolección de requisitos

En esta etapa se recoge y analiza la información sobre la actividad de la organización que usará la base de datos. Se recopila información para cada una de las vistas de usuario, incluyendo:

- la descripción de los datos que se usarán o que se generarán,
- detalles sobre cómo se usarán o generarán estos datos,
- cualquier otro requisito que deba cumplir la base de datos

A partir de esta información se determinarán los requisitos que deberán incorporarse a la nueva base de datos. Estos requisitos se recogen en un documento llamado la **especificación de requisitos**.

La recopilación y análisis de requisitos debe ser una fase temprana del diseño de una base de datos. Aunque esta recopilación debe ser tan detallada como sea posible, es frecuente que en el comienzo del proyecto haya información poco clara o ambigua. Intentar cubrir esa información demasiado temprano puede llevar a un bloqueo o a una solución errónea, ya que es demasiado pronto para discernir la solución. Por ello en la recopilación de requisitos hay que buscar un equilibrio en el nivel de detalle, que permita poner en marcha el proyecto en la dirección adecuada, pero manteniendo la especificación como un documento vivo, que se verá enriquecido conforme avanza el proyecto.

La información recopilada suele carecer de una estructura clara e incluso ser algo informal, por lo que es necesario sintetizarla utilizando técnicas como los Diagramas de Flujo de Datos (DFD).

También es importante realizar una selección adecuada de que ideas, requisitos o características serán incorporadas al sistema. Un sistema que no incluye las necesidades fundamentales de los usuarios puede ser difícil de manejar, o incluso inoperante. Por otro lado, incorporar demasiadas funciones, puede hacer que la base de datos sea innecesariamente compleja, dificultando el desarrollo y el mantenimiento, o aumentando la posibilidad de incorporar errores o bugs.

Otra actividad importante que debe llevarse a cabo en esta fase es decidir la arquitectura que dará soporte a una base de datos con múltiples vistas de usuario. Las tres alternativas más frecuentes son las siguientes:

- la solución centralizada
- la solución de integración de vistas
- combinaciones de ambas soluciones

▪ La solución centralizada

En esta solución se unen todos los requisitos de las distintas vistas de usuarios en una única lista de requisitos. Cada una de las listas de requisitos de las vistas de usuarios es etiquetada con un nombre acorde a las funcionalidades que cubre. Más adelante, en la fase de diseño, se creará un modelo único para la base de datos. Esta alternativa es preferible cuando los requisitos de las diferentes vistas de usuario se solapan de manera significativa y la base de datos no es demasiado compleja

▪ La solución de integración de vistas

En esta aproximación se mantienen separadas cada una de las listas de requisitos generadas a partir de cada una de las vistas de usuarios. A partir de cada una de estas listas de requisitos, o de unas pocas, se construirá un **modelo de datos local** diferenciado. En una fase de diseño posterior estos modelos se combinan en un **modelo de datos global**, que representa todos los requisitos del sistema. Este método es preferible cuando hay grandes diferencias entre las vistas de usuario y la complejidad del sistema de base de datos hace preferible trabajar con partes más manejables.

En algunos casos será incluso recomendable un enfoque mixto, donde algunas los requisitos de algunas vistas de usuario se combinan en una lista única, allí donde las funcionalidades y datos asociados a las vistas se solapan, mientras que para otras vistas de usuario del sistema se mantienen listas de requisitos particulares que den lugar a su propio modelo local.

7 Diseño de Bases de Datos

○ Aproximaciones al Diseño de Bases de Datos

Para afrontar el diseño de una base de datos tenemos dos caminos a nuestra disposición: el diseño desde abajo (o *bottom-up*), o el diseño desde arriba (o *top-down*). En el **diseño desde abajo** se comienza analizando las piezas más básicas del modelo, los atributos, es decir, las propiedades de las entidades y las relaciones. El análisis de las asociaciones entre estos atributos hará que vayan agrupándose en tablas que representan a las entidades del modelo, y en las relaciones entre estas entidades. Esta estrategia de diseño está fuertemente relacionada con el procedimiento de normalización de bases de datos relacionales.

El diseño desde abajo es adecuado para bases de datos sencillas con no demasiados atributos. Su uso se vuelve cada vez más complicado cuando se aplica a bases de datos más complejas con gran número de atributos. En estos casos, es difícil establecer todas las dependencias funcionales entre los atributos que se usan para guiar el proceso de normalización. Los modelos conceptual y lógico de una base de datos compleja puede contener de cientos a miles de atributos, por lo que es necesario establecer un método que simplifique el proceso de diseño. Además, en las fases iniciales del diseño de una base de datos compleja, puede ser difícil tener en cuenta todos los atributos que formarán parte del modelo.

En estos casos es más conveniente seguir los pasos del **diseño desde arriba**. En este método se empieza creando un modelo que incluye unas pocas entidades de alto nivel (muy genéricas) y sus relaciones. En iteraciones sucesivas el modelo se va refinando y se van añadiendo nuevas entidades con sus relaciones, así como dando mayor nivel de detalle a las ya definidas. En general, este proceso de diseño está más relacionado con la creación de modelos Entidad-Relación.

Existen otros métodos de diseño de bases de datos, como el método excéntrico o el uso de estrategias mixta. En el **método excéntrico** (*inside out*) sigue los pasos del modelo desde abajo, con la diferencia de que empieza identificando un conjunto de entidades principales y a partir de ellas va descubriendo otras entidades, relaciones y atributos relacionadas con ellas. La **estrategia mixta** combina los métodos desde arriba y desde abajo en diferentes partes del modelo para finalmente ensamblar todos los submodelos obtenidos.

○ Modelado de datos

Construyendo el modelo de datos se consigue un doble propósito, por un lado, comprender el significado (la semántica) de los datos, y por otro, facilitar la comunicación sobre el sistema que se está desarrollando. Para construir el modelo de datos, el diseñador deberá realizar (y realizarse) preguntas sobre las entidades, sus relaciones y los atributos. En las respuestas, encontrará la semántica de los datos de la organización, que existía independientemente que fuera o no explicitada en un modelo de datos. Las entidades, sus relaciones y los atributos son clave en cualquier dominio de conocimiento u organización. Sin embargo, su significado real puede ser difícil de comprender hasta que no se plasman en un modelo de datos y son documentados. Así, el modelo de datos facilita comprender:

- El punto de vista sobre los datos de cada usuario,
- La naturaleza propia de los datos, sin que importe su representación física,
- El uso que se hace de los datos en las distintas vistas de usuario

Los modelos de datos pueden ser utilizados para transmitir los requisitos que se han recopilado sobre la organización. Para un usuario acostumbrado a la notación empleada, el modelo es una herramienta de gran utilidad para la comunicación con los diseñadores del sistema de información.

A continuación, se muestran los requisitos deseables que un diseño óptimo de una base de datos debe buscar, teniendo en cuenta que algunos de los mismos pueden ser incompatibles, caso en que el modelo deberá encontrar el equilibrio más adecuado entre ellos.

- Validez estructural: consistencia con el modo de definir y organizar la información en la organización

- Simplicidad: facilidad de ser comprendido tanto por personal técnico como no técnico
- Expresividad: describir cada entidad, relación o atributo con el suficiente nivel de detalle
- Unicidad: cada unidad de información debería aparecer una sola vez en el modelo
- Genericidad: el modelo no debería estar ligado a ninguna aplicación ni tecnología en particular
- Extensibilidad: el modelo debería poder evolucionar a partir de nuevos requisitos sin perjuicio a los usuarios
- Integridad: consistencia con el modo de usar y gestionar la información en la organización
- Representación gráfica: el modelo debería poder definirse mediante una notación gráfica que facilite su comprensión.

○ Fases del diseño de la base de datos

Se distinguen tres fases en el diseño de una base de datos: el diseño conceptual, el diseño lógico y el diseño físico

▪ Diseño conceptual de la base de datos

El diseño conceptual es absolutamente independiente de cualquier aspecto físico de los sistemas de información de la organización, y se centra en los datos de la misma y su uso de la manera más abstracta posible. Este modelo es construido a partir de los requisitos recopilados en las fases anteriores. El diseño conceptual es totalmente independiente de cualquier característica del software SGBD que se pretenda utilizar, así como de los programas de aplicación, lenguajes de programación, sistemas operativos, infraestructuras hardware, etc.

En este proceso de diseño el modelo es validado frente a los requisitos de usuario. Una vez se ha obtenido el modelo conceptual servirá de base para el desarrollo del siguiente paso, el diseño lógico.

▪ Diseño lógico de la base de datos

A partir del modelo de datos conceptual se continua el análisis para obtener el diseño lógico de la base de datos. Si el modelo conceptual era absolutamente independiente de cualquier consideración física, en el modelo lógico se tendrán en cuenta el tipo de modelo de datos utilizado por el SGBD de la organización: relacional, orientado a objetos, etc. Sin embargo, se seguirán ignorando otros aspectos físicos, como dispositivos de almacenamiento, etc.

Para validar la corrección de este modelo lógico de datos, se utiliza la técnica de **normalización**. Con ella es posible asegurar que el modelo evita redundancias, que pueden provocar inconsistencias de información una vez que la base de datos esté operativa.

El modelo de datos lógico sirve de base para generar el modelo de datos físico, pero también es útil como documentación durante la fase de mantenimiento, una vez que la base de datos está en funcionamiento. Un modelo de datos lógico correctamente actualizado facilita los cambios en los programas de aplicación o en los datos del sistema de información.

▪ Diseño físico de la base de datos

El **diseño físico** es la tercera de las etapas en el diseño de la base de datos. En este nivel se consideran las características del SGBD escogido. Aunque en un principio el diseño físico deriva del diseño lógico, también hay influencia en la dirección contraria, ya que decisiones tomadas sobre el diseño físico (por ejemplo, por razones de rendimiento) pueden afectar y verse reflejadas también en el diseño lógico.

En resumen, el objetivo del diseño físico es describir la implementación que se realizará del modelo lógico en un SGBD concreta. Para SGBDs que utilizan el modelo relacional, se necesitará:

- Crear un conjunto de tablas relacionales y restricciones sobre dichas tablas a partir del modelo lógico
- Seleccionar las estructuras específicas de almacenamiento y los métodos de acceso a los datos de cara a optimizar el rendimiento en el trabajo con la base de datos,
- Diseñar la seguridad necesaria para el sistema

La separación de los modelos conceptual y lógico del modelo físico es clave en los grandes sistemas de información, ya que facilita separar el qué hay que construir del cómo se construirá; requieren ser realizados en distintas etapas del proyecto, ya que decidir cómo construir requiere entender bien qué hay que construir; son diseños que requieren habilidades distintas por lo que a menudo son realizados por distintos profesionales.

El diseño de base de datos es un proceso iterativo, que requiere un proceso casi constante de refinamientos. Conforme los diseñadores comprenden el funcionamiento de la organización y el significado de los distintos datos, y van incorporándolo al modelo, descubrirán que tienen que hacer ajustes a otras partes del diseño. Es crítico que los modelos conceptual y lógico sean un buen reflejo de la realidad, ya que de la implementación que se realice a partir del modelo físico depende en gran medida de estos.

8 Selección del sistema de gestión de bases de datos

Naturalmente no siempre que se diseña una nueva base de datos o sistema de información es necesario seleccionar un nuevo sistema de gestión de bases de datos, ya que será recomendable utilizar los sistemas que ya estén en uso en la organización. En caso de que fuera necesario, el momento adecuado para este análisis es después del diseño del modelo conceptual y antes del modelo lógico. De este modo, ya se habrán recogido los requisitos del sistema de información, que pueden ser determinantes a la hora de seleccionar uno u otro sistema; mientras que para completar la fase de diseño lógico es necesario conocer el SGBD que se utilizará.

A la hora de seleccionar un SGBD, el objetivo es encontrar una plataforma que satisfaga los requisitos presentes y futuros de la organización, teniendo en cuenta los costes, no solo de adquisición si no de la posible adaptación de los sistemas de información existentes al nuevo SGBD, así como los costes asociados a la migración de datos y a la formación de los usuarios.

- Definir el alcance del análisis

Como paso previo a la selección de un SGBD es conveniente acotar el alcance y los objetivos del análisis, así como las tareas que deben realizarse. Además de estos puntos, se pueden recopilar en el documento los criterios en base a los que será evaluado el SGBD (que derivarán de los requisitos de los sistemas de información), una lista preliminar de productos a tener en cuenta, y todas las restricciones y plazos necesarios para el análisis.

- Preselección de dos o tres productos

En base a los criterios considerados críticos se puede determinar una preselección de dos o tres productos para su evaluación. Por ejemplo, la selección puede estar condicionada a un presupuesto determinado, a la facilidad de encontrar empresas con capacidad de soporte en el mercado, a la compatibilidad con otro software, o la compatibilidad para funcionar sobre determinada infraestructura de la organización. En esta etapa es útil contactar con otros usuarios del SGBD, que pueden transmitir su experiencia con el apoyo recibido por el vendedor, o la capacidad del producto para funcionar correctamente bajo determinadas circunstancias. También es posible acceder a pruebas de rendimiento (*benchmarks*) que comparan diversos productos. Con esta información, y un estudio preliminar de funcionalidades y características de algunos SGBD, será posible establecer esta preselección de dos o tres sistemas.

- Evaluar productos

Los distintos SGBDs pueden ser evaluados en base a distintos parámetros. Estos parámetros pueden ser evaluados de manera conjunta (por ejemplo, capacidad de definición de datos) o de manera específica (por ejemplo, tipos de datos disponibles).

Marcar únicamente el cumplimiento o no de determinada característica no suele ser muy útil, salvo en el caso de aquellas características consideradas críticas, y que se habrán utilizado para construir la preselección del apartado anterior. Por ello, es más útil otorgar una ponderación distinta a las características en la evaluación, de forma que se obtenga finalmente una puntuación ponderada que permita comparar los distintos productos.

Además de esta evaluación cuantitativa, también puede ser conveniente solicitar una demostración a los proveedores del producto, incluso la posibilidad de realizar un testeo del mismo en la infraestructura de la organización. Para esta evaluación será necesario construir un prototipo que represente los requisitos esperados en la base de datos.

- Informe final y recomendaciones

Como resultado del análisis se debería generar un informe que recoja todo el proceso seguido y que presente como resultado la recomendación de un SGBD determinado.

DEFINICIÓN DE DATOS	DEFINICIÓN FÍSICA DEL SISTEMA
Restricciones de clave primaria	Estructuras de fichero disponibles
Especificación de claves externas	Mantenimiento de la estructura de ficheros
Tipos de datos disponibles	Facilidad de reorganización
Extensibilidad de los tipos de datos	Indexación
Especificaciones de dominio	Campos/registros de longitud variable
Facilidad de reestructuración	Compresión de datos
Controles de integridad	Rutinas de encriptación de datos
Mecanismo de vistas	Requisitos de memoria
Diccionario de datos	Requisitos de almacenamiento
Independencia de datos	
Modelo de datos	
Evolución de esquemas	
ACCESIBILIDAD	MANEJO DE TRANSACCIONES
Lenguaje de consultas: SQL2/ SQL 2011 ...	Rutinas de <i>backup</i> y recuperación. Capacidad de <i>checkpoints</i>
Interfaz con otros lenguajes	Logs
Multi usuario	Granularidad de la concurrencia
Seguridad	Estrategia de resolución de bloqueos
- controles de acceso	Modelos de transacciones avanzadas
- mecanismos de autorización	Proceso de consultas en paralelo
APLICACIONES	DESARROLLO
Métricas de rendimiento	Herramientas 4GL/5GL
Ajustes de rendimiento	Herramientas CASE
Aplicaciones de importación y exportación de datos	Interfaces gráficas
Monitorización de usuarios	Procedimientos almacenados, disparadores y reglas
Apoyo a la administración de la base de datos	Herramientas de desarrollo web
OTRAS CARACTERÍSTICAS	
Actualizaciones	Interoperabilidad con otros SGBDs y sistemas
Estabilidad del proveedor	Integración web
Base de usuarios	Herramientas de replicación
Formación y soporte a usuarios	Capacidad de trabajo distribuido
Documentación	Portabilidad
Requisitos de sistema operativo	Requisitos <i>hardware</i>
Coste	Capacidad de trabajo en red
Ayuda online	Arquitectura (2 o 3 capas)
Control de versiones	Rendimiento
Optimización de consultas	Rendimiento de transacciones
Escalabilidad	Número máximo de usuarios concurrentes
Herramientas analíticas y de informes	XML y servicios web

9 Diseño de la aplicación

En el ciclo de vida de las bases de datos, el diseño de las aplicaciones que harán uso o explotarán la base de datos se representa como una fase en paralelo a la del diseño de la base de datos. El diseño de las aplicaciones necesitará de muchos detalles sobre los modelos de datos que solo se averiguan cuando avanza la fase de diseño de la base de datos. Pero, a la vez, el diseño de las aplicaciones revelará detalles y requisitos que tendrán que ser incorporados al diseño de la base de datos, por lo que se trata de actividades fuertemente interconectadas.

En el diseño de la aplicación o aplicaciones deben incorporarse todos los requisitos de las vistas de usuarios. Además de diseñar cómo se conseguirán estas funcionalidades mediante el diseño de transacciones, será necesario diseñar la interfaz de usuario para la base de datos. Una interfaz adecuada, fácil de usar y de aprender, que muestre la información de una manera lógica y predecible hace más fácil que se cumplan los objetivos del sistema de información. Sin embargo, las interfaces de usuario son a menudo minusvaloradas, o se postergan hasta fases posteriores del diseño.

○ Diseño de transacciones

Una transacción es un concepto que representa situaciones del mundo real que deben ser recogidas en una base de datos como la compra de un bien, el registro de un usuario, o la modificación de los datos personales de un cliente.

Aunque en el mundo real se pueden percibir como eventos simples, su representación en la base de datos puede requerir de múltiples pasos, afectando a múltiples registros de una o varias entidades. El SGBD debe garantizar la consistencia de la base de datos antes y después de cada transacción, incluso en el caso que se produzca un error durante la misma. El ejemplo más clásico es el de una transferencia bancaria. Esta transacción debe modificar las cantidades registradas en las cuentas bancarias participantes, de forma que la cantidad transferida sea restada de la cuenta origen para ser sumada en la cuenta destino. Si se produjera un error en medio de la transacción, el proceso no debería quedar a medias, con una cuenta modificada y la otra no. Por ello, el SGBD debe poder asegurarse de poder retornar al estado inicial en caso de no ser capaz de ejecutar la transacción al completo.

En el diseño de transacciones se definen y documentan las características generales de las transacciones necesarias en la base de datos, indicando para cada una:

- Los datos que se utilizan en la transacción,
- Características funcionales de la transacción ;
- Resultado de la transacción;
- Importancia o criticidad para los usuarios;
- Frecuencia esperada de ocurrencia.

Esta actividad no debe retrasarse demasiado, de forma que se asegure que la base de datos implementada puede satisfacer todas las transacciones necesarias. Hay tres tipos principales de transacciones que deben ser tenidas en cuenta:

- **Transacciones de recuperación de información.** Se usan a la hora de mostrar información por pantalla, o de generar informes .
- **Transacciones de actualización de información.** Se utilizan para añadir nuevos registros, borrarlos, o modificarlos los existentes.
- **Transacciones mixtas.** Requieren tanto la recuperación de información como su modificación.

○ Diseño de la interfaz de usuario

Los siguientes son algunos aspectos a tener en cuenta antes de implementar los formularios o informes de la base de datos.

- Título significativo de los formularios y pantallas
- Instrucciones comprensibles

- Agrupación lógica o secuencial de los campos
- Disposición atractiva del formulario/informe
- Etiquetas familiares para los campos
- Terminología y abreviaturas consistentes
- Uso consistente del color
- Identificación visual del tamaño de los campos
- Fácil desplazamiento por los campos
- Mensajes de error adecuados para los valores no válidos
- Campos opcionales marcados claramente
- Mensajes explicativos para los campos
- Signo de finalización

10 Prototipos

Un prototipo es un modelo del sistema de información que no reúne todos los requisitos del sistema final. Su propósito es mostrar algunos aspectos del sistema, o ayudar a valorar o seleccionar entre alternativas de diseño o construcción. Mediante su uso, es posible aclarar los requisitos del sistema tanto a los usuarios finales como a los desarrolladores del sistema. Su principal ventaja es que su producción es relativamente rápida y económica.

En el desarrollo de prototipos hay dos estrategias principales: por un lado es posible construir un prototipo para descartarlo una vez que se han probado en él los requisitos o alternativas que se deseen, y tal vez más adelante, construir otros prototipos para otros aspectos del sistema. Por otro lado, existen prototipos evolutivos a los que se van incorporando los nuevos requisitos, no son descartados, y se van transformando paulatinamente en la base de datos final.

11 Implementación

Una vez se ha completado el diseño de la base de datos (se haya o no hecho uso de prototipos) es el momento de desarrollar propiamente la base de datos y sus programas de aplicación. La base de datos se construirá utilizando el Lenguaje de Definición de Datos (LDD) de nuestro SGBD. Con él se construirán las estructuras de la base de datos, así como sus ficheros de soporte. En esta fase también se construyen las vistas de usuario.

Los programas de aplicación se desarrollan normalmente usando lenguajes de programación de tercera o cuarta generación. Parte de estos programas de aplicación serán las transacciones que se hayan definido en el diseño, que se desarrollan utilizando el Lenguaje de Manipulación de Datos (LMD) del SGBD, tal vez formando parte de un programa escrito en otro lenguaje de programación como Java, C#, Python, JavaScript, etc.

También es necesario implementar las restricciones de integridad y seguridad del sistema. Algunas se implementan mediante el LDD, pero otras serán desarrolladas fuera de este lenguaje.

12 Conversión y carga de datos

En esta etapa se transfiere a la base de datos cualquier información existente en un sistema previo. También puede ser necesario la modificación de los programas de aplicación que utilizaban el sistema previo para que utilicen el nuevo sistema de información. Muchos SGBD proporcionan utilidades para importar datos desde diversas fuentes, y realizar en ellos las transformaciones necesarias para incorporarlos a la base de datos. En esta etapa, es muy importante realizar una planificación de los procesos de adaptación, migración y transformación de forma que se asegure la continuidad de acceso a la información a lo largo de todo el proceso.

13 Testeo

Antes de pasar a producción, la nueva base de datos debe pasar una fase rigurosa de testeo. Para ello se diseñará cuidadosamente un plan de pruebas junto a un conjunto verosímil de datos. En general, el testeo no será capaz de demostrar que no existen ningún fallo o error en la base de datos, pero cuando se realiza diligentemente revelará fallos y errores en los programas de aplicación, y tal vez en el modelo de la base de datos que podremos corregir. Además, aunque las medidas de rendimiento que se realizan en la base de datos de prueba no son directamente extrapolables a los resultados en producción, sí que servirán de aproximación para estos.

Como en la fase de diseño, es importante contar con los usuarios en el proceso de testeo. En un caso ideal, se usarán datos ficticios sobre un sistema paralelo, pero si esto no es posible y es necesario hacer las pruebas sobre la base de datos final habrá que tomar las consideraciones de seguridad necesarias (copias de respaldo, etc.) para evitar incidentes; y será necesario tener en cuenta las normativas de protección de datos personales para el uso de estos durante las pruebas.

Las pruebas no deben limitarse a los requisitos funcionales de la base de datos y sus aplicaciones, si no que deben incluir la usabilidad de estas. Algunas medidas que deberían realizarse incluyen:

- Facilidad de aprendizaje. ¿Cuánto tiempo necesita un usuario para usar el sistema de manera eficiente?
- Rendimiento. ¿Con qué velocidad responde el sistema a las peticiones del usuario?
- Robustez. ¿Es el sistema tolerante a los fallos del usuario?
- Recuperabilidad. ¿Cómo de bien se recupera el sistema de los fallos de los usuarios?
- Adaptabilidad: ¿Está el sistema demasiado ajustado a una única forma de trabajo?

Una vez que las pruebas han terminado, será posible entregar el sistema y ponerlo a disposición de los usuarios.

14 RESUMEN ESQUEMÁTICO

Un sistema de información incluye los siguientes elementos: base de datos; software de base de datos, software de aplicación, hardware (incluyendo medios de almacenamiento), usuarios y desarrolladores del sistema.

La base de datos es una parte fundamental del sistema de información, por lo que su ciclo de vida está estrechamente vinculado al del ciclo de vida del sistema de información.

Las principales etapas del ciclo de vida de una base de datos son: planificación, definición del alcance, análisis y toma de requisitos, diseño de la base de datos, selección del SGBD (opcional), diseño de las aplicaciones, prototipado (opcional), implementación, transformación y carga de datos, y pruebas.

En la fase de planificación se realizan las actividades de gestión necesarias para asegurar el éxito del resto de fases.

En la fase de definición del alcance, se establecen los límites y objetivos principales de la base de datos, así como las vistas de usuario. Cada vista de usuario fija las necesidades de un tipo de usuario en la base de datos.

Durante la fase de Análisis y recogida de requisitos se recopila información sobre las áreas de la organización que estarán representadas en la base de datos. Hay tres aproximaciones para la gestión de los requisitos de una base de datos con múltiples vistas de usuario: la gestión centralizada, la integración de vistas, y la combinación de ambos métodos.

El método de gestión centralizada recopila todos los requisitos en una única colección, a partir de la que se crea un modelo global en la fase de diseño. En el método de integración de vistas, se recopilan colecciones diferenciadas de requisitos para cada una de las vistas de usuario. En la fase de diseño se crean modelos para cada una de las vistas sobre los que a continuación se realiza un proceso de integración.

Se distinguen tres etapas en la fase de Diseño:

En el Diseño conceptual de la base de datos se construye un modelo de la base de datos independiente de los detalles relacionados con la implementación.

En la fase de Diseño lógico se construye un modelo ajustado al paradigma de modelado que ofrece el SGBD utilizado, aunque se mantiene independiente de características específicas del SGBD o de otros detalles físicos.

Por último, en la fase de Diseño físico de la base de datos se describen los detalles de más bajo nivel de la base de datos, los ajustes en la organización de ficheros e índices, mejoras del rendimiento o de la capacidad de trabajo concurrente, o medidas relacionadas con la seguridad e integridad de los datos.

En algunos casos será necesario un proceso de selección del SGBD.

Durante el Diseño de aplicaciones se realizan tanto el Diseño de Transacciones como el Diseño de la Interfaz de Usuario. Una transacción es un conjunto de acciones que lleva a cabo un único usuario o aplicación para modificar o recuperar la información de la base de datos.

En ocasiones, se utilizan prototipos durante la fase de diseño para visualizar las soluciones propuestas y encontrar alternativas a problemas que se encuentren en esta fase.

Durante la fase de implementación, se construyen tanto la base de datos como las aplicaciones diseñadas, utilizando el SGBD y los lenguajes de programación escogidos. También se realiza la conversión de aplicaciones existentes a la nueva base de datos.

En el caso de que existe información previamente existente en el sistema será necesaria la fase de Transformación y carga de datos.

Antes de la puesta en producción del sistema y la entrega a los usuarios finales, será necesario pasar una fase rigurosa de pruebas o testeo.

14 GLOSARIO

Especificación de requisitos

Actividad en el ciclo de vida de los sistemas de información en la que se recogen y definen detalladamente las características y funcionalidades que deberá ofrecer el sistema una vez construido.

Modelo de datos

global

Modelo de datos que recoge todos los aspectos de la base de datos

local

Modelo de datos que recoge únicamente los elementos involucrados con una vista de usuario

Normalización

En bases de datos se dice que un atributo depende de otro cuando sus valores están asociados de algún modo, por ejemplo, el atributo Comunidad Autónoma depende del de Provincia. Una base de datos está en forma normal cuando se reducen o limitan las dependencias entre los atributos de cada tabla, de forma que solo existen dependencias con respecto a los atributos identificados como claves candidatas de la tabla. La normalización es un procedimiento de transformación del esquema de una base de datos para que alcance una forma normal.

16 BIBLIOGRAFÍA BÁSICA

T. Connolly y C.Begg. Database systems: a practical approach to design, implementation, and management (6th ed.). Capítulo 10, secciones 10.1, 10.2, 10.3, 10.4, 10.5, 10.6, 10.7, 10.8, 10.9, 10.10, 10.11 y 10.12.