



TESINA DE LICENCIATURA

Programa de apoyo para alumnos con experiencia profesional

TÍTULO: Migración de base de datos heredadas a otras base de datos relacionales
Caso Sistemas de Gestión–Sec..de Administración - Poder Judicial de la Pcia de Bs.As

AUTORES: Ana María de Jesús Fantinelli

DIRECTOR ACADÉMICO: Mg Rodolfo Bertone

DIRECTOR PROFESIONAL: Lic. Gabriela V. Demo

CARRERA: Licenciatura en Sistemas

Resumen

Este trabajo, además de presentar una reseña histórica del avance tecnológico de Bases de Datos que sugieren la necesidad de migrar datos y aplicaciones para lograr mejoras competitivas, pretende demostrar que, utilizando las características del modelo relacional, una fuente de datos migrada de versiones anteriores mejora el rendimiento, refuerza la seguridad y genera sistemas de mayor competitividad en la industria informática. Caso Sistemas de Gestión de la Secretaría de Administración del Poder Judicial de la Provincia de Buenos Aires.

Palabras Clave

Avance tecnológico – Base de datos – Migración de datos – Modernización de aplicaciones – Agilización de tareas – SQL – Usuarios

Conclusiones

Si bien, en términos de tecnología, el cambio sigue estando lejos del avance informático de la actualidad, para el trabajo diario de la Secretaría de Administración resulta muy beneficioso. Dados los resultados de una encuesta realizada a 46 usuarios se puede asegurar que la mayoría está muy conforme con el cambio de aplicaciones

Trabajos Realizados

Se implementaron aplicaciones desarrolladas en Borland Developer Studio 2006 para reemplazar los sistemas en Delphi 5 que se ejecutaban en DOS. Además se realizó la migración de los datos de estos sistemas de tablas DBF a una base de datos relacional SQL A partir de la migración, se hicieron nuevos procesos que interconectan las bases de datos de distintos servidores y constantemente se agregan funcionalidades a las aplicaciones basadas en los beneficios de las bases de datos relacionales

Trabajos Futuros

Almacenar los documentos que actualmente se graban en las bases de datos, en la nube y asimismo recuperarlos, dado que el tamaño de las bases crece enormemente e incide en la performance de los procesos.

Desarrollar interfaces con un sistema web recientemente implementado por la Subsecretaría de Tecnología Informática, SGE (Sistema de Gestión de Expedientes) que los usuarios deben utilizar. De este modo, al intercambiar ambos sistemas las novedades, se evita la doble carga por parte de los usuarios.

Dedicatoria

Para Julia, Ignacio y Justina.

Agradecimientos

- a mi director académico Mg Rodolfo Bertone por guiarme para cerrar un ciclo.
- a mi directora profesional Lic. Gabriela Verónica Demo por todos sus aportes a este trabajo.
- a Mg. Claudia Banchoff Tzancoff, por convencerme que todavía estaba a tiempo de hacerlo.
- a la Lic. María Belén Goyhenespe que me orientó en cómo empezar este trabajo.
- a todos los que están siempre: mis hijos, mi familia y mis amigos.
- a la Facultad de Informática de la UNLP porque me enseñó todas las herramientas necesarias para que pueda desarrollar mi profesión por más de 30 años. Además porque me permitió cerrar esta etapa.

ÍNDICE DE CONTENIDOS

CAPÍTULO 1 - INTRODUCCIÓN, OBJETIVO, MOTIVACIÓN y DESARROLLO PROPUESTO	5
1.1- INTRODUCCIÓN	5
1.2- OBJETIVOS	5
1.3- MOTIVACIÓN DEL PROBLEMA	6
1.4- DESARROLLO PROPUESTO	6
CAPÍTULO 2 - FUNDAMENTOS TEÓRICOS	8
2.1 - ESTADO DEL ARTE	8
2.2 - MIGRACIÓN DE BASE DE DATOS	9
2.2.1- DEFINICIÓN	9
2.2.2- RAZONES PARA UNA MIGRACIÓN DE DATOS	9
2.2.3- CARACTERÍSTICAS Y CONSIDERACIONES DE UNA MIGRACIÓN DE DATOS	11
2.2.4- ELEMENTOS BÁSICOS EN UNA MIGRACIÓN DE DATOS	12
2.3- ETAPAS DE LA MIGRACIÓN DE DATOS	13
2.3.1- Relevamiento de Información.	13
2.3.2- Transformación/Mapeo de datos	13
2.3.3- Depuración de Datos	13
2.3.4- Construcción ETC (ETL)	13
2.3.5- Pruebas	14
2.3.6- Simulación de Cargas	14
2.3.7- Cargas Reales	14
2.3.8- Revisión de Resultados	14
CAPÍTULO 3 - EVOLUCIÓN DE LAS BASES DE DATOS	15
3.1- UN POCO DE HISTORIA...	15
3.1.1- La Evolución de los Sistemas de Bases de Datos	16
3.1.2- Primeros sistemas de base de datos	17
CAPÍTULO 4 - MIGRACIÓN DE BASES DE DATOS HEREDADAS A UNA BASE DE DATOS RELACIONAL - DESARROLLO PROPUESTO.	39
4.1- INCONVENIENTES DETECTADOS EN LAS APLICACIONES EN MS-DOS	41
4.2- SOLUCIÓN PROPUESTA Y DESARROLLADA	44
4.3- ETAPAS EN LA MIGRACIÓN	45
4.3.1- Relevamiento de la información	45
4.3.2- Transformación/mapeo de datos	46
4.3.3- Depuración de datos	46
4.3.4- Procesos de Migración.	47
4.3.5- Pruebas	52
4.3.6- Simulación de carga.	53
4.3.7- Cargas reales	53
4.3.8- Revisión de resultados	53
4.4- INCONVENIENTES DURANTE LA MIGRACIÓN DE DATOS E IMPLEMENTACIÓN DE LOS APLICATIVOS	54
CAPÍTULO 5 - RESULTADOS OBTENIDOS	56

5.1- MANEJO DE INFORMACIÓN	56
5.2- NUEVAS APLICACIONES	60
5.2.1- Inmuebles	60
5.2.2- Compras y Contrataciones	62
5.2.3- Contaduría.	63
5.2.4- Tesorería	66
CAPÍTULO 6 - CONCLUSIONES Y TRABAJOS FUTUROS	68
6.1- Conclusiones	68
6.2- Trabajos futuros	70
BIBLIOGRAFÍA	71
ANEXO I - RESULTADO DE LA ENCUESTA	73

ÍNDICE DE FIGURAS

Figura 1 - Razones para una migración de datos	11
Figura 2 - Elementos básicos de la migración de datos	12
Figura 3 - Etapas de la migración de datos	13
Figura 4 - Migración de datos	15
Figura 5 - Modelo de Base de datos Jerárquico	19
Figura 6 - Modelo de Base de datos de Red	20
Figura 7 - Relaciones Base	23
Figura 8 - Relaciones derivadas	24
Figura 9 - Primary key (clave primaria)	26
Figura 10 - Foreign key (clave foránea)	27
Figura 11 - Indexes (Clave índice)	28
Figura 12 - Stored Procedure (Procedimientos almacenados)	28
Figura 13 - Modelo de Base de datos orientador a objetos	30
Figura 14 - Modelo de Base de datos relacional orientada a objetos	32
Figura 15 - Modelo de Base de datos paralela y distribuida	33
Figura 16 - Marcas usuarias de bases NoSql	36
Figura 17 - Modelos de base de datos entre 1950 y 2010	38
Figura 18 - Organización de las aplicaciones por área/sector	40
Figura 19 - Mapeo de la unidad G	40
Figura 20 - Ambiente DBU	41
Figura 21 -Captura de impresora	43
Figura 22 - Organización de las aplicaciones bajo Windows	45
Figura 23 - Representación gráfica de las relaciones de un trámite	48
Figura 24 - Proceso para migrar tablas con n relaciones	49
Figura 25 - Proceso para migrar PROVEEDORES	49
Figura 26 - Representación gráfica de las relaciones de un trámite	52
Figura 27 - Ejemplo de conexión entre base de datos.	57
Figura 28 - Ejemplo de generación de consulta en SQL	57
Figura 29 - Ejemplo de Vistas	58
Figura 30 - Ejemplo de Stored Procedure	58
Figura 31 - Ejemplo de Trigger	59
Figura 32 - Ejemplo de Function	59
Figura 33 - Inmuebles - conexiones entre bases de datos	61
Figura 34 - Inmuebles - consulta de contratos	61
Figura 35 - Compras y contrataciones - conexiones entre bases de datos	62
Figura 36 - Compras y contrataciones - Consulta de órdenes de compra	63
Figura 37 - Aplicación área Contaduría	64
Figura 38 - Consulta de Alquileres - Aplicación de Contaduría	65
Figura 39 - Consulta de pagos - Aplicación de Contaduría	66
Figura 40 - Tesorería - Aplicaciones	67

CAPÍTULO 1 - INTRODUCCIÓN, OBJETIVO, MOTIVACIÓN y DESARROLLO PROPUESTO

1.1- INTRODUCCIÓN

En todas las organizaciones, tanto públicas como privadas, es imprescindible que utilicen aplicaciones que son de vital importancia para su funcionamiento, debido a que manejan volúmenes de información acorde a la envergadura de la empresa y requieren obtener resultados confiables, consistentes, de manera rápida y sencilla.

Ante el avance de la tecnología de forma exponencial, es inevitable que las aplicaciones muchas veces pasen a ser obsoletas y el manejo de las mismas se convierta en una tarea sumamente complicada y a veces imposible.

Para contrarrestar este avance, se plantea una modernización de los sistemas (tanto en su desarrollo, manejo de datos y plataforma), que garantice seguir obteniendo mínimamente los mismos resultados, como así también, nueva información, producto de las mejoras realizadas.

El caso que se presenta en este trabajo (Sistemas de la Secretaría de Administración del Poder Judicial de la Provincia de Buenos Aires) describe cómo se renovaron las aplicaciones y el almacenamiento de datos de cada área, dado que la tecnología que se estaba utilizando era tan obsoleta que prácticamente era imposible utilizar dichas aplicaciones. Lo más importante de este proyecto es implementar nuevos sistemas y que los datos existentes sigan disponibles para el usuario, con lo cual se planificó una migración desde archivos DBF a bases de datos relacionales SQL.

1.2- OBJETIVOS

- Demostrar que, desarrollando una reseña histórica de los últimos 30 años en el tema de Base de Datos, efectivamente el avance tecnológico creció de manera exponencial; y además, reconocer la necesidad de migrar datos y aplicaciones, para lograr mejoras competitivas. Si bien la cronología arranca en 1960, el crecimiento tecnológico se acelera a partir de los '90.

- Demostrar que, utilizando las características del modelo relacional, una fuente de datos migrada de versiones anteriores mejora el rendimiento, refuerza la seguridad y genera sistemas de mayor competitividad en la industria informática.

1.3- MOTIVACIÓN DEL PROBLEMA

Considerando que lo más valioso en una organización es la información que posee, es clave para la misma, mantener sus aplicaciones de gestión y control actualizadas, de manera de evitar inconvenientes producidos por el avance tecnológico en el manejo de sus datos.

Esta actualización, en la mayoría de los casos, implica cambios en los desarrollos, plataformas y/o almacenamiento de datos. Es imprescindible definir una estrategia correcta para migrar la información, en función de las necesidades, tiempos, presupuesto y objetivos.

Se define migración de datos “al proceso que necesitamos hacer para transferir los datos de un sistema a otro mientras cambiamos el sistema de almacenamiento donde se encuentran los datos, o bien mientras se practican las modificaciones necesarias en la base de datos o la aplicación que los gestiona” (<https://www.powerdata.es/migracion-de-datos> - 2022)

1.4- DESARROLLO PROPUESTO

Este trabajo reseña el crecimiento, desde un punto de vista cronológico, del manejo de datos entre los años 1960 y 2020, BD relacionales, tecnologías Orientadas a Objetos, etc. Como ejemplo, se presenta la transformación de los sistemas de gestión de la Secretaría de Administración del Poder Judicial de la Provincia de Buenos Aires.

Además, demuestra que utilizando las características del modelo relacional, una fuente de datos migrada de versiones anteriores, mejora el rendimiento, refuerza la seguridad y genera sistemas de mayor competitividad.

SISTEMAS DE ADMINISTRACIÓN DE ADMINISTRACIÓN DEL PODER JUDICIAL DE LA PROVINCIA DE BUENOS AIRES

En estos sistemas están afectadas las áreas de Compras y Contrataciones, Inmuebles, Contaduría, Bienes Patrimoniales y Tesorería.

Los sistemas originales datan del año 1993 y están desarrollados en Clipper 5.0 con tablas DBF y, se ejecutan sobre DOS bajo una red Novell.

Los sistemas actuales son aplicaciones desarrolladas en Borland Developer Studio 2006 con bases de datos relacionales (Microsoft SQL) bajo Windows, implementadas progresivamente desde 2016.

A continuación se enumeran los principales inconvenientes que se presentaron con los avances tecnológicos y que, de alguna manera, la organización se vió obligada a hacer la transformación:

- Manejo de teclado
- Manejo de los datos
- Manejo de índices de las tablas y campos memo
- Crisis Y2K
- Manejo de impresoras
- Aparición de los procesadores de 64 bits

Por otro lado, este trabajo expone otros tipos de inconvenientes no técnicos, que afectaron, entre otras cosas, los tiempos del proyecto, a saber:

- Grupo de trabajo
 - Poco compromiso de las partes
 - Implementación del Sistema Integrado de Gestión Administrativa Financiera (SIGAF)
 - Módulo de Registro Patrimonial (RP)
-

CAPÍTULO 2 - FUNDAMENTOS TEÓRICOS

2.1 - ESTADO DEL ARTE

Actualmente se puede afirmar que casi la totalidad de las actividades que se desarrollan en comercios, hospitales, fábricas, entidades financieras, educativas, sociales, gubernamentales, etc., utilizan sistemas informáticos. Dichos sistemas almacenan la información vital para todo tipo de operatoria.

Además, el avance de la tecnología de manera exponencial trajo consigo nuevos recursos tecnológicos: el uso de internet con nuevos fines, dispositivos móviles, comunicaciones más veloces y económicas, etc. Estas situaciones llevan a los diferentes organismos a enfrentar frecuentes cambios en sus sistemas informáticos. La mayoría de las veces se determina reemplazar las aplicaciones por otras más modernas. El desafío más grande que se presenta en estos casos es que la información existente en las aplicaciones originales siga disponible en las nuevas.

Existen en el mercado muchas herramientas para la migración de datos de una tecnología a otra. Muchos de los fabricantes de motores de base de datos ofrecen soluciones para, por ejemplo, alimentar una base de datos desde archivos de texto, o una base de datos de objetos desde una base relacional, o de una base de datos relacional de un fabricante a otra de distinto fabricante.

Los verdaderos problemas surgen porque todas estas herramientas no resuelven las diferencias funcionales ni organizan el proceso en sí.

En resumen, según (Serra, 2018), la situación actual puede expresarse de esta manera:

- Es muy frecuente que la implementación de un nuevo sistema informático se lleve adelante para reemplazar a otro.
- En estos casos, la información almacenada tiene gran valor y no puede perderse. El nuevo sistema y su antecesor no difieren solamente en aspectos técnicos, sino que lo hacen en cuestiones funcionales lo que presenta entonces un desafío técnico y funcional al momento de migrar datos.
- Si bien existen herramientas para resolver aspectos técnicos, sus posibilidades de utilización dependen del contexto y con ellas no se llega a abarcar todo el

problema. La migración de datos continúa siendo un proceso artesanal, poco estandarizado y que puede presentar una gran dificultad tanto en el desarrollo como en la implementación de sistemas.

2.2 - MIGRACIÓN DE BASE DE DATOS

2.2.1- DEFINICIÓN

Rodríguez y Rosso, (2013) definen la migración de datos como: “Es la transferencia segura y organizada de datos entre dos sistemas de bases de datos diferentes considerando la compatibilidad de tipos de datos, hardware, requerimientos, etc “.

Además Serra (2018) afirma que: “De esta definición queda claro que los procesos de migración de datos son aquellos que persiguen como objetivo transformar y transportar datos de una plataforma a otra, pero esto puede ocurrir en contextos diferentes y con motivaciones y objetivos distintos”

De las definiciones anteriores se puede concluir que una migración de datos incluye:

- La definición de un nuevo modelo de datos que contemple las características del existente.
- La utilización de las cualidades de los nuevos modelos para hacer más versátil el manejo de la información.
- La transformación de datos del modelo pre existente al nuevo modelo.
- La carga de dicha información. Esto significa que los datos extraídos tendrán que pasar por una serie de funciones de preparación, tras las cuales podrán cargarse en la ubicación que les está destinada.

2.2.2- RAZONES PARA UNA MIGRACIÓN DE DATOS

Serra, (2018) describe diversas razones por las que las organizaciones necesitan migrar datos entre distintas plataformas:

- Migrar transformando los datos para hacer posible un análisis con un enfoque diferente al previsto en el sistema que los genera. Por ejemplo:
-

entornos de data mining y data warehousing. En ellos los datos se migran a una base especialmente diseñada para poder hacer un análisis particular de los mismos. Esta base puede contener información de un solo origen o de varios.

- Migrar por razones de rendimiento. Algunos sistemas, sobre todo los que trabajan con grandes volúmenes de datos, pueden requerir en algunos procesos tiempos de respuestas difíciles de lograr. Una opción suele ser transportar datos desde una base transaccional a otra que esté organizada de manera tal que agilice las consultas y las acciones de búsqueda.
- Migrar por razones de distribución geográfica. Algunos sistemas mantienen bases de datos replicadas en lugares geográficamente separados con el objeto de garantizar el acceso eficiente a los mismos. En estos entornos pueden hallarse operaciones de traspaso de una fuente a otra, entendiendo por fuente, un servidor, un conjunto de servidores o la infraestructura que los contenga.
- Migrar por razones de seguridad o de disponibilidad o de facilidad de acceso, etc.

En fin, pueden encontrarse muchas razones y muchos contextos en los que se implementan pasajes de datos de una estructura a otra con mayor o menor grado de transformación y con diferentes niveles de complejidad.

El siguiente gráfico, basado en datos recolectados por IBM (Serra, 2018), muestra las situaciones frecuentes por las que se realizan migraciones de datos, siendo la principal la actualización de tecnología que llega a un 48%, considerando los ítems reemplazo de almacenamiento, reemplazo de servidores y fin del arrendamiento de almacenamiento. El menor porcentaje (10%), corresponde a la relocación de servidores. Esto se refiere a reubicar los servidores tanto en el mismo edificio como en lugares alrededor del mundo.

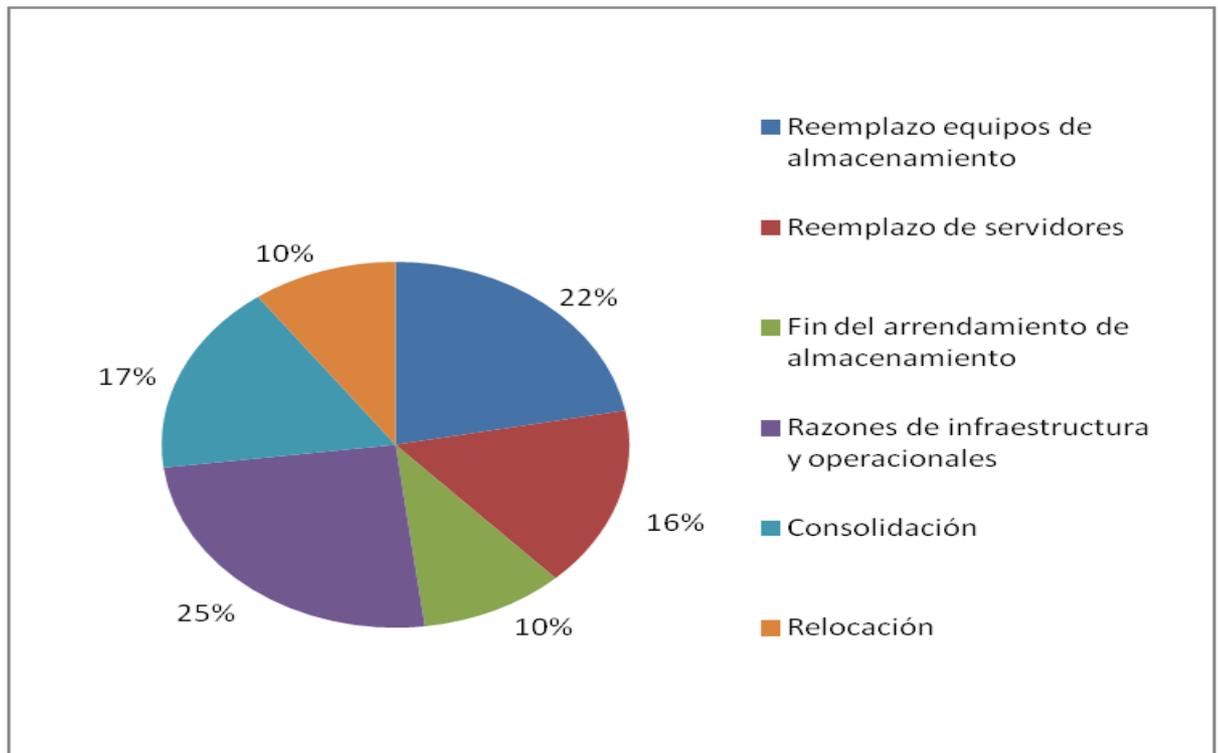


Figura 1 - Razones para una migración de datos

2.2.3- CARACTERÍSTICAS Y CONSIDERACIONES DE UNA MIGRACIÓN DE DATOS

La migración es un proceso y como tal debe obtener resultados medibles. Suele ser una tarea complicada ya que son muchos los factores a tener en cuenta para llegar a resultados exitosos.

Es importante tener en claro lo que se quiere migrar, a donde se quiere migrar y que se espera de la migración de datos

Siempre existe el caso de tener que realizar transformaciones de algunos tipos de datos, debido a que los mismos se comportan diferentes en la base de datos origen que en la destino. Por ejemplo: datos booleanos, formato de fechas, campos memo etc.

Además de los programas para ejecutar en una migración de datos, se deben considerar otros factores. Por ejemplo: Procesos de negocio, limpieza de datos, fuentes de información, equipos de trabajo, herramientas a utilizar, planes de pruebas, etc.

Para obtener una migración exitosa dentro de la organización, es fundamental efectuar una planificación y un análisis exhaustivo de lo que se quiere migrar, a donde se quiere migrar y que se espera de la migración de datos.

2.2.4- ELEMENTOS BÁSICOS EN UNA MIGRACIÓN DE DATOS

Como se ha mencionado anteriormente, una migración de datos tiene como “objetivo transformar y transportar datos de una plataforma a otra, pero esto puede ocurrir en contextos diferentes y con motivaciones y objetivos distintos” (Serra, 2018). La figura X representa los elementos básicos de una migración para satisfacer dicho objetivo.

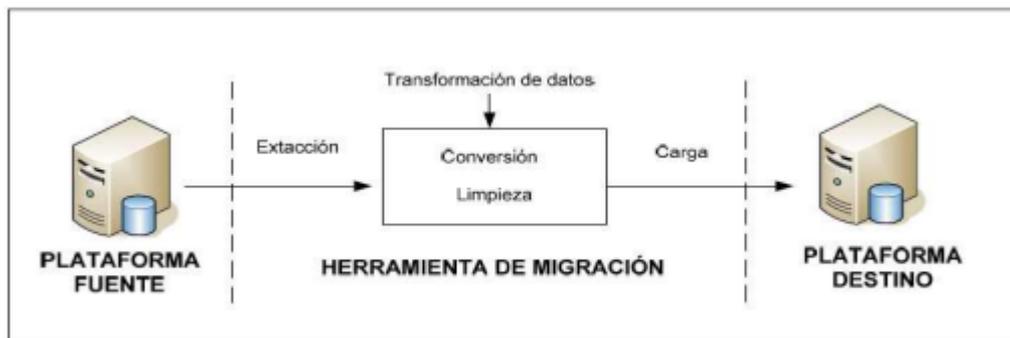


Figura 2 - Elementos básicos de la migración de datos. (Rodriguez y Rosso (2013)).

Según lo detallado gráficamente, los elementos básicos de una migración son::

- Plataforma Fuente: lugar de almacenamiento de la información principal a analizar para su migración hacia otra base de datos.
- Plataforma Destino: lugar de destino de la información de la Plataforma Fuente después de su transformación y limpieza respectiva.
- Herramienta de Migración: Permitirá llevar a cabo la extracción, la transformación y la carga de los datos desde la base de datos fuente hacia la base de datos destino de acuerdo a los requerimientos establecidos en una organización.

2.3- ETAPAS DE LA MIGRACIÓN DE DATOS

En una migración de datos no sólo hay que definir los elementos básicos sino también, las etapas que se deben realizar.

Estas etapas están representadas gráficamente en la figura 3.

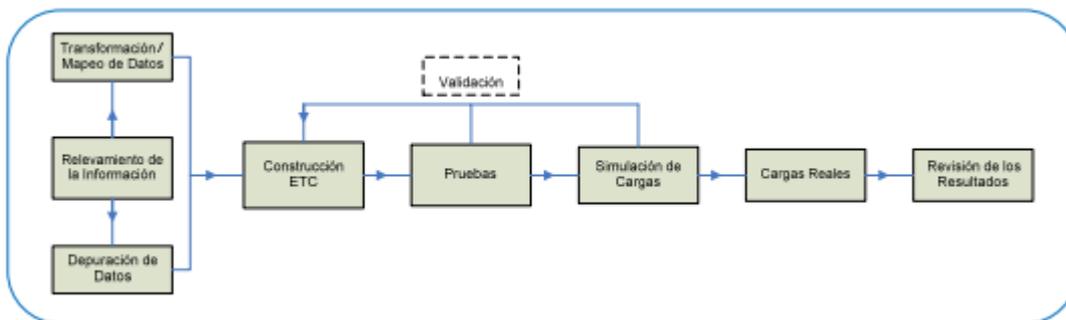


Figura 3 - Etapas de la migración de datos.(Rodríguez y Rosso (2013)).

Según lo detallado en la figura 3 se definen las siguientes etapas:

2.3.1- Relevamiento de Información: Esta etapa contempla la definición de los datos que requiere el nuevo sistema y la identificación de la fuente de origen.

2.3.2- Transformación/Mapeo de datos: Esta etapa, contempla la definición de los diccionarios con las reglas de transformación y el mapeo de campos entre las fuentes de origen y destino.

2.3.3- Depuración de Datos: Esta etapa contempla la depuración de los datos en las fuentes de origen.

2.3.4- Construcción ETC (ETL): En esta etapa, se realiza la construcción de herramientas necesarias para realizar la migración, basándose en la extracción , transformación y carga de datos desde la base fuente hacia la base destino de acuerdo a los requerimientos establecidos en una organización.

- Extracción de datos: La extracción es el proceso de análisis y recopilación de los datos desde la base de datos fuente que se quieren migrar hacia la base de datos destino.
- Transformación de datos: Los tipos de datos, esquemas y modelos conceptuales de la base de datos fuente son transformados, convertidos y

limpiados, mediante procesos, de tal forma que serán transferidos hacia la base de datos destino de acuerdo a los requerimientos de una organización.

- Carga o integración de datos. Este proceso permite cargar e integrar los datos debidamente transformados y limpiados de la base de datos origen en la base de datos destino de acuerdo a los requerimientos establecidos en la organización.

2.3.5- Pruebas: En esta etapa, se realizan las pruebas funcionales de la migración de datos, determinando que la información sea traspasada con éxito al nuevo sistema.

2.3.6- Simulación de Cargas: En esta etapa, se pretende identificar los inconvenientes que se podrían tener en las cargas reales, a través de la simulación del proceso real.

2.3.7- Cargas Reales: En esta etapa, se realiza la migración real de los datos desde el antiguo sistema al nuevo sistema.

2.3.8- Revisión de Resultados: En esta etapa, se realiza la revisión final de la información cargada al nuevo sistema.

CAPÍTULO 3 - EVOLUCIÓN DE LAS BASES DE DATOS

3.1- Un poco de historia...

Como se definió anteriormente, la migración de datos consiste en transferir de manera segura y organizada los datos desde una fuente de datos origen a otra diferente, teniendo en cuenta la compatibilidad de tipos de datos, hardware, requerimientos, etc . La siguiente figura X representa gráficamente el proceso de migración.

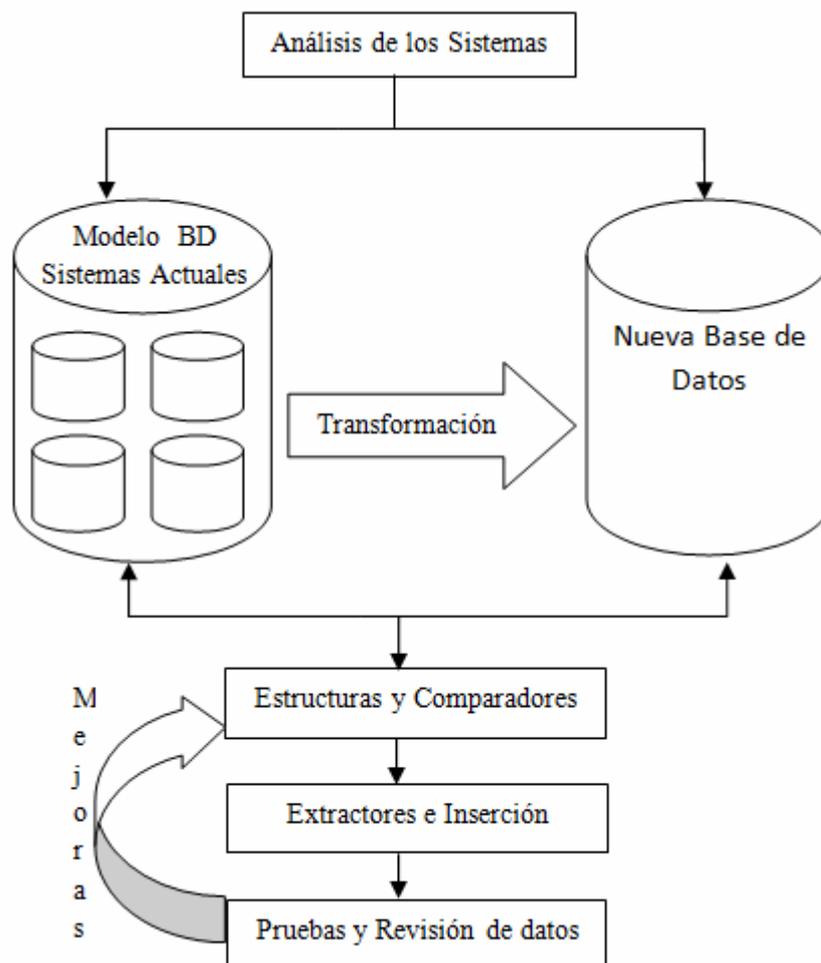


Figura 4 - Migración de Datos (Cerón, Aviña y Covarrubias, pág 117)

Este trabajo presenta una revisión de las “nociones y evolución básicas de los modelos pre-relacionales, relacional, objetual y objeto-relacional, las bases de datos

paralelas y distribuidas, multimedia, los almacenes de datos, la relación entre las bases de datos y la web”.(Orallo, 2002)

3.1.1. La Evolución de los Sistemas de Bases de Datos

El concepto de sistemas de información se refiere a recopilar datos, estructurarlos, centralizarlos y almacenarlos convenientemente, de manera de poder recuperarlos en cualquier momento, como así también, proporcionar a los usuarios información fidedigna para la toma de decisiones y acciones.

Resumiendo, se llama base de datos a la “colección de datos recopilados y estructurados que existe durante un período de tiempo... Generalmente, un sistema de información consta de una o más bases de datos, junto con los medios para almacenarlas y gestionarlas, sus usuarios y sus administradores” (Orallo, J. H. (2002))

Desde hace un tiempo la tecnología deriva la gestión de una base de datos a determinadas aplicaciones específicas llamadas **Sistemas de Gestión de Bases de Datos (SGBD)**

Según Ullman y Widom.(1997), las funciones básicas de un sistema de gestión de Base de Datos, son:

1. Permitir a los usuarios crear nuevas bases de datos y especificar su estructura, utilizando un lenguaje o interfaz especializado, llamado lenguaje o interfaz de definición de datos.
 2. Dar a los usuarios la posibilidad de consultar los datos (es decir, recuperarlos parcial o totalmente) y modificarlos, utilizando un lenguaje o interfaz apropiado, generalmente llamado lenguaje de consulta o lenguaje de manipulación de datos.
 3. Permitir el almacenamiento de grandes cantidades de datos durante un largo periodo de tiempo, manteniéndolos seguros de accidentes o uso no autorizado y permitiendo un acceso eficiente a los datos para consultas y modificaciones.
 4. Controlar el acceso a los datos de muchos usuarios a la vez, impidiendo que las acciones de un usuario puedan afectar a las acciones de otro sobre datos diferentes y que el acceso simultáneo no corrompa los datos.
-

3.1.2- Primeros sistemas de base de datos

Los primeros sistemas de base de datos aparecieron a finales de 1950, evolucionaron de los sistemas de archivos ya que ofrecían almacenamiento de grandes cantidades de datos por largos períodos de tiempo, pero no garantizaban la pérdida de datos ante fallos. La recuperación de los mismos era solamente por copia de seguridad. Además, los sistemas de archivo eran limitados tanto el lenguaje de consulta como la interfase para crear nuevas bases y tampoco ofrecían acceso concurrente a los archivos con lo cual no aseguraban la correcta actualización de un archivo ante el ingreso de más de un usuario al mismo tiempo a ese archivo.

Las primeras aplicaciones importantes de los sistemas de archivo, se caracterizaban por datos definidos por partes bien diferenciadas y tenían poca interrelación entre ellas. Como por ejemplo, sistemas de reservas e información de vuelos, sistemas bancarios, y sistemas de ventas, facturación, etc.

“Los primeros verdaderos SGBDs, evolucionados de los sistemas de archivos, obligaban a que el usuario visualizara los datos de manera similar a como se almacenaban. Los primeros sistemas de archivos habían evolucionado del código máquina a un lenguaje ensamblador con ciertas instrucciones de acceso a disco, nociones que se pueden ver en sistemas todavía en funcionamiento hoy en día, tales como la línea AS de IBM. No es de extrañar que con este nivel de abstracción la manera de recuperar los datos estuviera estrechamente ligada al lenguaje de programación utilizado. Un avance importante lo constituyó el comité formado en la **CON**ference on **DA**ta **SY**stems and **L**anguages, **CODASYL**, en 1960, estableciendo el **CO**mmon **B**usiness-**O**riented **L**anguage (**COBOL**) como un lenguaje estándar para interrelacionar con datos almacenados en archivos. En aquella época representó lo que se vinieron a llamar los lenguajes de programación de tercera generación. Las instrucciones específicas de un programa Cobol para tratamiento de archivos eran las de abrir un archivo, leer un archivo y añadir un registro a un archivo. Lo típico en gestión de datos en esta época era un archivo ‘batch’ de transacciones que se aplicaba a un maestro en cinta Pero pronto los discos magnéticos empezaron a sustituir a las cintas magnéticas, lo que supuso una reconcepción del almacenamiento, al pasarse del acceso secuencial al acceso aleatorio (este paso es el que se conoce como el paso de los sistemas de bases de datos de primera a segunda generación)”. (Orallo, 2002)

Durante los sesenta empezaron a aparecer distintos modelos de datos con el objetivo de conseguir mayor independencia entre las aplicaciones y la organización física de los datos.

Los modelos más popularizados en esta época, fueron el modelo jerárquico o basado en árboles, y el modelo en red o basado en grafos.

Modelo de base de datos jerárquico

Una base de datos jerárquica consiste en un conjunto de segmentos (registro) que se conectan entre sí por medio de enlaces. Cada segmento es un grupo de campos (atributos), que contienen un solo valor cada uno de ellos. Un enlace es una asociación o unión entre dos segmentos exclusivamente. Es bastante rígida. Una vez diseñada, es compleja cambiarla y además, es necesario un conocimiento amplio de la forma en la que se han almacenado los datos para poder manipularlos efectivamente. Gráficamente tiene una estructura de árbol.

Los SGBD jerárquicos son modelos lógicos basados en registros de formato fijo de distintos tipos, con una cantidad fija de campos o atributos, donde cada uno tiene longitud fija que describen datos en los niveles lógicos y de vista.

Las características principales de implementar este modelo son:

- Globalización de la información: permite a los diferentes usuarios considerar la información como un recurso corporativo que carece de dueños específicos.
- Eliminación de información inconsistente: si existen dos o más archivos con la misma información, los cambios que se hagan a éstos deberán hacerse a todas las copias del archivo de facturas.
- Permite compartir información.
- Permite mantener la integridad de la información: la integridad de la información es una de sus cualidades altamente deseable y tiene por objetivo que sólo se almacena la información correcta.
- Independencia de datos: el concepto de independencia de datos es quizás el que más ha ayudado a la rápida proliferación del desarrollo de Sistemas de Bases de Datos.

En este tipo de modelos se establece en forma de árbol donde la raíz es un nodo ficticio. Es decir, una base de datos jerárquica es una colección de árboles.

Modelo Jerárquico

Un modelo de datos jerárquico es un modelo en el cual los datos son organizados en una estructura parecida a un árbol. La estructura usa relaciones padre/hijo: cada padre puede tener muchos hijos pero cada hijo sólo tiene un

padre. Todos los atributos de un registro específico son catalogados bajo un tipo de entidad.

Una base de datos jerárquica es un tipo de SGBD que almacena la información en una estructura jerárquica que alcanza los registros en forma de estructura de árbol en donde un nodo padre de información puede tener varios nodos hijo. De la misma manera se puede establecer relación entre los nodos hermanos, en este caso la estructura en forma de árbol se convierte en una estructura en forma de grafo dirigido.

El modelo jerárquico se clasifica en estructuras lineales y arborescentes. En las estructuras lineales, cada tipo de registros padres sólo puede tener un tipo de registro hijo. En las arborescentes, un tipo de registro padre puede tener varios tipos de registro hijo. El producto comercial de tipo jerárquico más extendido y el único que ha llegado hasta nuestros días es el IMS de IBM (nacido en 1968 sobre equipos System/360).

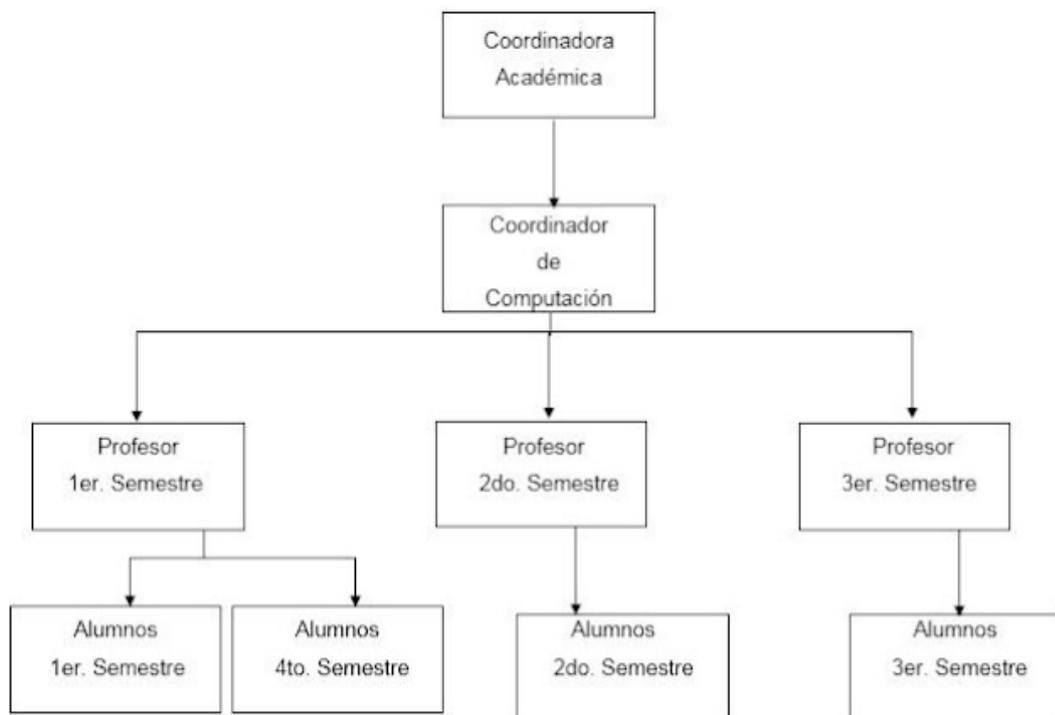


Figura 5 - Modelo de base de datos jerárquico
(<http://victorgh84.blogspot.com/2015/09/base-de-datos-jerarquica.html>)

Modelo de base de datos de red

Orallo, (2002) expuso que el modelo en red se estandarizó a finales de los sesenta mediante un informe de CODASYL (por esta razón se lo conoce como el modelo CODASYL). El primer informe de CODASYL ya incluía la sintaxis y semántica de un lenguaje de definición de datos (**DDL, Data Definition Language**) y de un lenguaje de manipulación de datos (**DML, Data Manipulation Language**). Siguiendo muchos comentarios y revisiones de expertos y usuarios se realizó un nuevo informe [CODASYL 1971], en el que ya se incluía la posibilidad de definir vistas para los distintos grupos de usuarios.

El término base de datos en red hace referencia a la manera en la que los datos se relacionan con otros datos. Es decir, un modelo en red porque representa los datos que contiene como una red de registros y conjuntos (en realidad listas circulares llamadas **sets**) que se relacionan entre sí, formando una red de relaciones. Para hacer esto utiliza registros, tipos de registro y tipos de conjunto. El modelo en red tampoco se utiliza casi hoy en día y si subsiste es como consecuencia del mantenimiento de un sistema todavía no reconvertido o no portado a modelos y SGBD más modernos. Aunque este modelo permite más flexibilidad que el modelo jerárquico, y en algunos casos se adapta muy bien a algunos tipos de transacciones, se considera superado por otros modelos, como el relacional o de objetos.

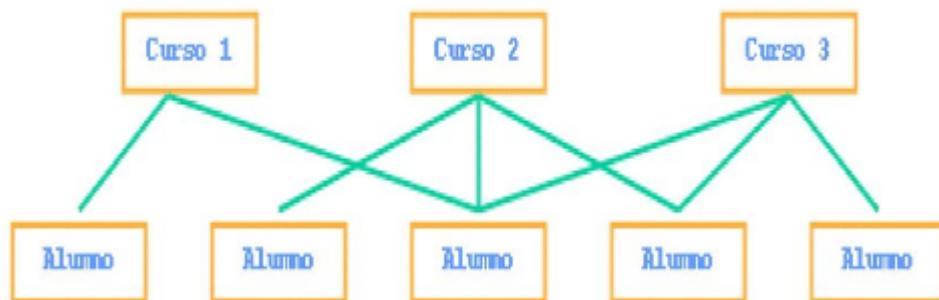


Figura 6 - Modelo de Base de Datos de Red

Conclusiones sobre modelo jerárquico y de red (Orallo, 2002)

Ambos modelos, con el paso de los años, se pueden considerar como modelos puente hacia el modelo relacional, ya que se incorporan en los primeros sistemas de gestión de bases de datos que introducen un mayor nivel de

independencia, respecto a la estructura interna, pero siguen teniendo una estructura de cierto bajo nivel y de compleja manipulación.

Otro problema con estos modelos y sistemas iniciales es que no van acompañados de lenguajes de consulta de alto nivel. Por ejemplo, el lenguaje de consulta CODASYL tenía sentencias que permitían al usuario saltar de un elemento de datos a otro a través de grafos de punteros entre estos elementos. Se requería un gran esfuerzo para escribir estos programas, incluso para consultas muy sencillas.

Hay que destacar el nacimiento y definición del concepto de transacción y sus propiedades asociadas, lo que se conoce como el “ACID test”. Aunque el concepto de transacción evoluciona en las primeras décadas del desarrollo de las bases de datos, se considera el trabajo de James Gray (Gray, 1981) como el que le da su forma actual. Se dice que un SGBD cumple el “**ACID test**” si observa las propiedades de **A**tomicidad, **C**onsistencia, **A**islamiento y **D**urabilidad.

- **Atomicidad:** los resultados de una transacción o bien pasan a ser completados todos (commit) o bien pasan a ser todos deshechos (rollback).
- **Consistencia:** las bases de datos se transforman de estados íntegros a estados íntegros, es decir, entre estados válidos. Una transacción sólo se puede completar si el estado final es íntegro.
- **Aislamiento:** los resultados de una transacción son invisibles para el resto de transacciones de otros procesos hasta que la transacción se ha completado.
- **Durabilidad o permanencia:** una vez completada una transacción, los resultados (cambios) de la misma se hacen permanentes, incluso frente a fallos del sistema y de medios de almacenamiento.

Modelo de base de datos relacional.

En 1970, el matemático Edgar F. Codd propuso que los sistemas de base de datos organizaran los datos en estructuras llamadas **relaciones**.

Se define una relación como un conjunto de tuplas en las que el orden no es importante, y en la que se pueda definir cualquier estructura de datos compleja que permita una rápida respuesta a las diferentes consultas. Además, el usuario de un sistema relacional, sólo debe preocuparse por el **qué** consultar y no el **cómo**. En otras palabras, el usuario no tiene que conocer la estructura de almacenamiento.

Además, siguiendo con el texto de (Orallo, 2002), las consultas pueden expresarse en un lenguaje de muy alto nivel, lo que incrementa en gran medida la eficiencia de los programadores sobre bases de datos. En resumen, Codd concibió un sistema donde el usuario sería capaz de acceder a la información con comandos parecidos al lenguaje natural y donde la información estuviera almacenada en 'tablas'.

A pesar de las virtudes de este modelo, no tuvo una aceptación inmediata. Sobre todo, IBM (en ese momento líder del mercado) fue un fuerte opositor, debido a que había invertido mucho tiempo y dinero en su producto IMS basado en el modelo jerárquico.

Para contrarrestar la aparición en el mercado del primer sistema relacional Ingres (desarrollado por la Universidad de California), IBM, en 1974, puso en marcha el desarrollo de otro sistema relacional System R; un sistema multiusuario con un lenguaje de consulta estructurado SEQUEL, antecesor de **SQL (Structure Query Language)**.

Entre finales de los 70 y principio de los 80, varias empresas comenzaron a comercializar productos relacionados a sistemas de base de datos relacionales, los cuales fueron llamados SGBD de cuarta generación. A saber:

PRODUCTO	EMPRESA
Oracle	Oracle Inc.
Sysbase	Sysbase Inc.
RDB	Digital Equipment Corp, actualmente Compact
Informix	Informix Inc.
Unify	Unify Inc.

También, en los 80', aparecieron implementaciones del modelo relacional para computadoras personales, tales como: RIM, RBASE 5000, Paradox, OS/2 Database Manager, DBase IV, XDB, Watcom SQL, SQL Server de Sybase Inc., SQL Server de Microsoft, Access de Microsoft, PostgreSQL y MYSQL.

Características de las bases de datos relacionales (Suárez, 2008)

- Se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de registros (filas y columnas).
- La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y ajenas (o foráneas).
- Las claves primarias son la clave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de los datos.
- Las claves ajenas se colocan en la tabla hija, contienen el mismo valor que la clave primaria del registro padre; por medio de éstas se hacen las relaciones.

Elementos:

Relaciones base y derivadas

Suárez (2008), afirma que en una base de datos relacional, todos los datos se almacenan y se accede a ellos por medio de relaciones.

Las relaciones que almacenan datos son llamadas "relaciones base" y su implementación es llamada "tabla".

	Column Name	Data Type	Allow Nulls
	cod_departamento	smallint	<input type="checkbox"/>
	descripcion	varchar(40)	<input type="checkbox"/>
	nem	varchar(2)	<input type="checkbox"/>
	aop	char(3)	<input checked="" type="checkbox"/>
	aop_nro	char(3)	<input checked="" type="checkbox"/>
	vigente	char(1)	<input checked="" type="checkbox"/>
	instalacion	smallint	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Figura 7 - Relaciones base

Otras relaciones no almacenan datos, pero son calculadas al aplicar operaciones relacionales. Estas relaciones son llamadas "relaciones derivadas" y su implementación es llamada "vista" o "consulta".

Las relaciones derivadas son convenientes ya que expresan información de varias relaciones actuando como si fuera una sola.

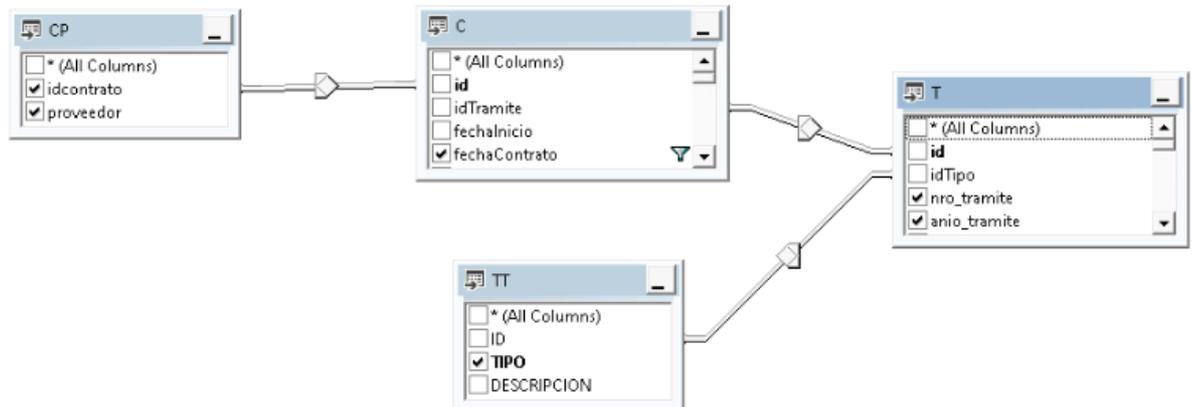


Figura 8 - Relaciones derivadas

Restricciones

Suarez (2008) define: “Una restricción es una condición que obliga el cumplimiento de ciertas condiciones en la base de datos ...Restringen los datos que pueden ser almacenados en las tablas. ” La mayoría no están determinadas por los usuarios, sino que están definidas para cumplir con las características de una base de datos relacional. Un

ejemplo de restricciones definidas por el usuario podría ser usar un campo con valores enteros entre 1 y 10.

El resultado de estas restricciones, usualmente, es un valor booleano, indicando si los datos cumplen o no la restricción. Ayudan en la organización de los datos.

Dominios

Del mismo modo que en el punto anterior, Suarez (2008), define: “Un dominio describe un conjunto de posibles valores para cierto atributo. Como un dominio restringe los valores del atributo, puede ser considerado como una restricción.”

Asignar un dominio a un atributo significa que los valores de este atributo deben estar incluidos en el conjunto especificado. Algunos de los tipos de dominios son: enteros, cadenas de texto, fechas, etc

Clave única

Se denomina clave única de una tabla a uno o más campos de la tabla cuyos valores identifican de manera única cada registro. Dicho de otra manera, no van a existir dos registros cuyos valores para los campos definidos como clave única sean iguales.

Clave primaria

La clave primaria o primary key, es una clave única que no puede tener valor NULL que identifica unívocamente un registro de la tabla. Esta clave permite relacionar los datos con otras tablas a través de claves foráneas (foreign key).

Una tabla sólo puede tener una clave primaria. A continuación se presenta un ejemplo de primary key.

Los campos TIPODOC y NRODOC conforman la clave primaria de la tabla de AGENTES. Esto asegura que en dicha tabla no van a existir dos registros con los mismos tipo y número de documento.

	Column Name	Data Type	Allow nulls
	TIPODOC	char(1)	<input type="checkbox"/>
	NRODOC	int	<input type="checkbox"/>
	APYNOM	varchar(50)	<input checked="" type="checkbox"/>
	NUMCAR	int	<input checked="" type="checkbox"/>
	LEGCON	varchar(6)	<input checked="" type="checkbox"/>
	LEGINT	varchar(7)	<input checked="" type="checkbox"/>
	LEGSUE	varchar(6)	<input checked="" type="checkbox"/>
	NUMERO	int	<input checked="" type="checkbox"/>
	DOMICILIO	varchar(45)	<input checked="" type="checkbox"/>
	CIUDAD	varchar(25)	<input checked="" type="checkbox"/>
	FECHANAC	datetime	<input checked="" type="checkbox"/>
	APORTA	datetime	<input checked="" type="checkbox"/>

Figura 9 - Primary key (clave primaria)

Clave foránea

Para determinar la existencia de una relación entre dos tablas, se utiliza una clave foránea o foreign key. En otras palabras, en una tabla se define un campo como clave foránea (no es necesario que sea clave única en la tabla). Este campo va a determinar la relación con otra tabla.

La figura [x] es un ejemplo de clave foránea. El campo IDOP de la tabla OP_ALQUILER se relaciona con el campo ID de la tabla OP. Esto significa que todos los valores del campo IDOP de la tabla OP_ALQUILER deben existir como valores del campo ID en la tabla OP.

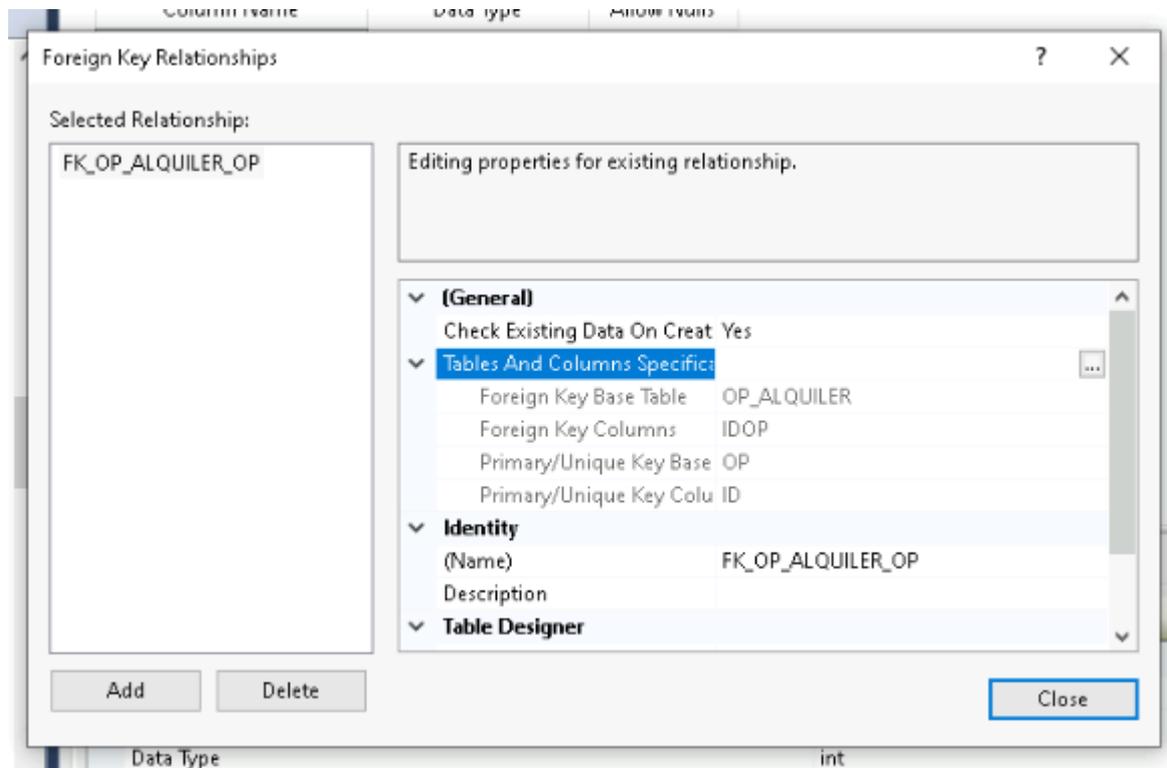


Figura 10 - Foreign key (clave foránea)

Clave índice

Ante la necesidad de acceder, de manera más rápida a los datos, surgen las claves índice (Indexes). Estas claves pueden crearse con cualquier combinación de campos de una tabla, permitiendo localizar los registros de manera no secuencial

La figura 11 muestra la definición de la clave índice para la tabla de FACTURAS. Está formada por los campos FECHA y ANIOFRO de forma ascendente.

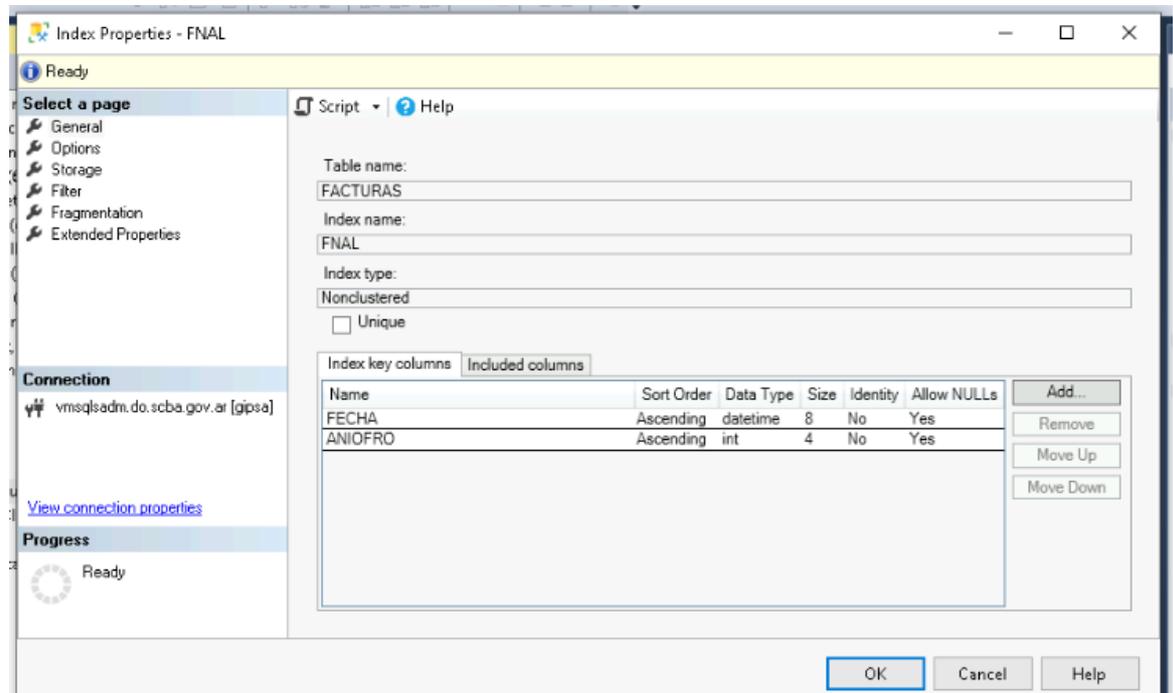


Figura 11 - Indexes (Clave índices)

Procedimientos almacenados

Un procedimiento almacenado o Stored Procedure, es un código ejecutable, tales como procesos y funciones, que se almacena en la base de datos. Mediante estos procesos se realizan operaciones comunes como insertar, modificar, eliminar registros de una tabla, o también para operaciones que recopilan datos estadísticos , encapsulan cálculos complejos, etc.

La Figura 12 muestra un segmento de código de una stored procedure.

```
ALTER PROCEDURE [dbo].[ACTUALIZA_MONTO]
    @IDCONTRATO INT,
    @MONTO MONEY
AS
BEGIN
    UPDATE ALQUILERES
    SET MONTOANTERIOR=MONTO
    ,MONTO=@MONTO
    WHERE IDCONTRATO=@IDCONTRATO
END
```

Figura 12 - Stored Procedure (Procedimientos almacenados)

Protocolo para la conectividad entre base de datos - ODBC

Con respecto a lo expuesto por Orallo (2002), la aparición de diversos sistemas relacionales planteaba problemas de compatibilidad. Existían muchas corporaciones que utilizaban distintos SGBDs para diferentes partes de su sistema de información.

Era muy usual que la organización tuviera un sistema central con su SGBD además de otros SGBD sobre computadoras personales. Era fundamental que estos SGBD estuvieran relacionados y pudieran compartir sus datos. Ya no sólo el poder importarlos o exportarlos entre los distintos SGBD, sino poder acceder a ellos, es decir utilizar un SGBD para acceder a los datos que gestionara otro SGBD.

Cómo esta problemática era mayor cuando se combinaban sistemas centrales con personales, aparecieron protocolos para la conectividad entre bases de datos. El estándar más utilizado es el **ODBC (Open Database Connectivity)** de Microsoft que sirve tanto para conectar bases de datos como para que las aplicaciones puedan acceder a diferentes bases de datos. Aparece el concepto de “fuente de datos”, en el que incluso el SGBD que lo gestiona puede pasar desapercibido.

El ODBC es una **API (Applications Programming Interface)** para enlazar aplicaciones con una base de datos. Los programas se conectan a una base de datos sin tener que usar los comandos y características específicas del SGBD. Los comandos ODBC se utilizan en los programas y luego se traducen, por la interfaz ODBC, en los comandos específicos que hay sobre el SGBD. Esto permite además que los programas se puedan portar de SGBD a SGBD con un mínimo de cambios de código, ya que el usuario indica qué fuente de datos ODBC está utilizando, lo que permite un cambio y adaptación a otros SGBD, especialmente los nuevos SGBD y versiones que van apareciendo.

Modelo de base de datos orientado a objetos

Según Orallo (2002), a mediados de los años 80, algunas aplicaciones exigían mayor expresividad en los datos con los que trabajaban. Por ejemplo, las bases de datos médicas, requerían mayor flexibilidad en la manera en la que los datos se representaban y eran accedidos.

Sistemas de Gestión de Bases de Datos Orientados a Objetos (SGBDOO, o simplemente, SGBDO) a principios de los años 90 empezaron a aparecer en el mercado, a partir de compañías como Objectivity.

“El modelo de la base de datos orientado a objetos define la base de datos como una colección de objetos utilizados en la programación orientada a objetos (es decir, que emplean lenguajes como Smalltalk, C++ o Java, por ejemplo). Este modelo utiliza tablas también, pero no solo se limita a ellas y permite almacenar información muy detallada sobre cada objeto.

Los objetos se dotan de un conjunto de características propias, que a su vez se diferencian de objetos similares. Los objetos similares pueden agruparse en una clase y cada objeto de esta es una instancia. Las clases intercambian datos entre sí a través de métodos (mensajes). “
(<https://ayudaleyprotecciondatos.es/bases-de-datos/modelos/> - 03/2023)

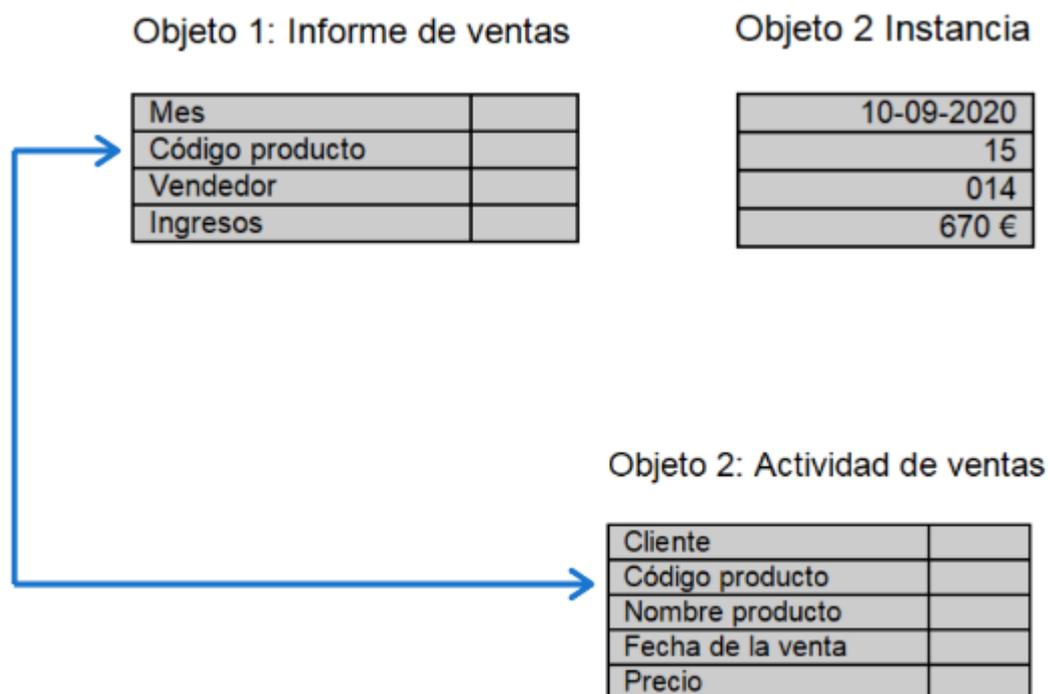


Figura 13 - Modelo de base de datos orientado a objetos
(<https://ayudaleyprotecciondatos.es/bases-de-datos/modelos/> - 03/2023)

En otras palabras, este modelo almacena la información sobre una entidad como un objeto persistente y no como una fila en una tabla. Lo que implica que es más eficiente en términos de requerimientos de espacio en disco, en especial con la consultas, ya que, en lugar de almacenarlas, utiliza una serie de punteros a los objetos seleccionados. Además, asegura que los usuarios puedan manipular los datos sólo de las maneras en las que el programador haya especificado.

Modelo de base de datos relacionales orientadas a objetos

Podría concluirse, que por las características descritas anteriormente, el modelo de bases de datos orientado a objetos, debería de haber superado en la práctica al modelo relacional. No obstante, después de más 15 años, el mercado de las bases de datos orientadas a objetos no supone más de un 5% del mercado de las relacionales

“El modelo objeto-relacional se definió, en los años 80, como una extensión objetual del modelo relacional, permitiendo la definición de nuevos tipos y de relaciones de herencia, entre otras cosas. Permite a las organizaciones continuar usando sus sistemas existentes y sus datos, sin realizar prácticamente cambios y les permiten empezar a utilizar gradualmente características orientadas a objetos, especialmente si se hace en conjunción con aplicaciones desarrolladas en entornos también orientados a objetos.” (Orallo, (2002).

Como escribió Orallo (2002), surge, entre otras cosas, por la necesidad imperiosa de tratar nuevos tipo de datos, por ejemplo, audio, imágenes y video; además de tipos definidos por el usuario con sus propiedades.

Como ejemplo, este modelo utiliza el protocolo para la conectividad **JDBC** (**J**ava **D**ata**B**ase **C**onnectivity). Originalmente fue pensado para interrelacionar bases de datos relacionales con Java, pero fue incorporando algunos detalles objeto-relacionales. JDBC es una API (Applications Programming Interface) desarrollada por Sun Microsystems para conectar Java con las bases de datos, con una filosofía inspirada en ODBC. De hecho es usual conectar un programa Java con una base de datos directamente o a través de ODBC, dependiendo de si el fabricante de la base de datos ha creado los drivers necesarios para Java.

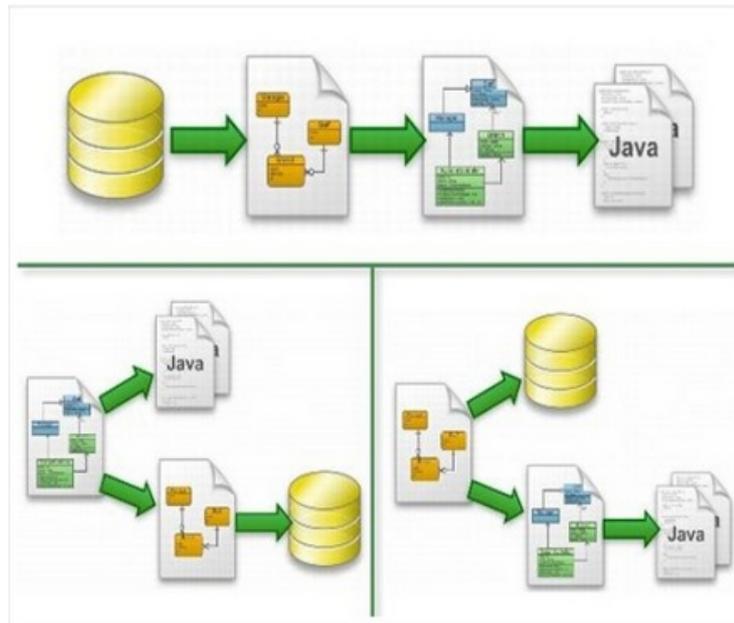


Figura 14 - Modelo de base de datos relacional orientada a objetos (<https://basededatosutim.wordpress.com/2014/03/21/objeto-relacionales/> - 03/2023)

Modelo de base de datos paralelas y distribuidas

Alrededor de los años 80 comenzaron a desarrollarse los sistemas de bases de datos paralelas sobre una serie de procesadores de propósito general funcionando en paralelo (por ejemplo IBM, Tandem, Oracle, Informix, Sybase y AT&T basan sus sistemas paralelos en este sistema). Pero al aumentar la capacidad de proceso, surgieron inconvenientes con respecto a otros límites de las máquinas: cantidad de memoria, cantidad de discos que podían conectarse al mismo procesador y la cantidad de procesadores corriendo en paralelo, en otras palabras, en la práctica, con el aumento de la información de una gran base de datos en un único sistema paralelo, hace que no pueda dar abasto con toda la información que tiene que almacenarse, ordenarse y consultarse. A raíz esto, entra en escena, el paradigma de las bases de datos distribuidas: los datos se almacenan en dos o más computadoras, llamadas nodos, y estos nodos están conectados en una red.

Esto debe realizarse de manera transparente para los usuarios. Ellos deben creer que están trabajando con un único sistema.

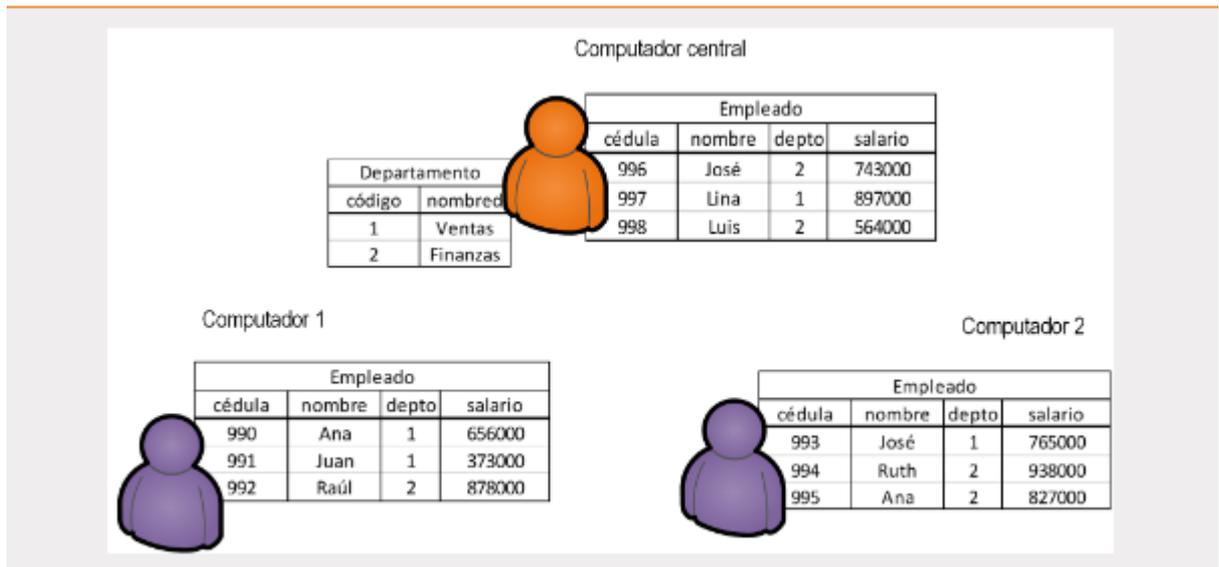


Figura 15 - Modelo de Base de datos paralela y distribuida

(<https://www.redalyc.org/journal/2570/257051186010/html/> -03/2023)

Modelo de base de datos Multimedia

Ante el crecimiento del uso de datos multimedia en las distintas aplicaciones, las cuales además de almacenar texto, necesitan almacenar imágenes, audio, animaciones y/o video. De este requerimiento surge diseñar bases de datos apropiadas para el almacenamiento y recuperación de la información multimedia.

Un artículo de Alvez y Vecchietti (2006) define que el propósito de los Sistemas de Gestión de Bases de Datos Multimedia (**SGBDMM**), es:

- organizar y estructurar información multimedia (hipertextos, imágenes, animaciones, video, audio) para su almacenamiento y posterior recuperación.
- consultar los datos multimedia y textuales representados en diferentes formatos, de responder a las consultas y presentar los resultados en términos audiovisuales,
- desarrollar la presentación de los datos audiovisuales con diferentes requerimientos de calidad, manejar la dependencia del tiempo y la presentación sincronizada de los datos multimediales.

Siguiendo con el artículo (Alvez y Vecchietti,2006, pág 2), a continuación se enumeran algunos modelos propuestos para SGBDMM:

- **Algebraico:** Las operaciones contienen un mecanismo para combinar y expresar relaciones temporales y espaciales entre los videos, como también para asociar información descriptiva. Se implementa en un prototipo escrito en C + + con un conjunto de clases para manejar las funciones básicas y el procesamiento de los videos.
- **Logical Hypervideo Data Model (LHDM):** representa objetos de video de niveles múltiples, denominados “objetos calientes” (hot objects). Para la representación de asociaciones semánticas se proponen los videos hyperlinks y para los datos de los videos vinculados con estas propiedades los hyper video. La implementación se realizó en una base de datos prototipo denominada web-based video database system (WVT DB) usando un código escrito en Java.
- **Basado en un sistema de base de datos orientada a objetos:** utiliza DISIMA (base de datos para imágenes). Ha propuesto una extensión al lenguaje de consulta OQL (Object Query Language) denominado MOQL (Motion Object Query Language) que permite consultas a las imágenes como también definir las especificaciones de una presentación de imágenes y/o videos.
- **VIDEX:** implementado en el sistema de información multimedia SMOOTH. Propone un modelo de clases y relaciones entre clases definidas en UML, que contiene las relaciones espaciales y temporales entre las clases y los segmentos multimedia que se quieren representar. SMOOTH está implementado sobre la opción Objeto-Relacional de ORACLE.

Este tipo de bases de datos tienen mucha relación con los SGBD objeto relacionales (SGBD) y los orientados a objetos (SGBDOO) ya que pueden almacenar tipos de datos multimedia.

Modelo de base de datos Nosql (Not Only SQL)

A principios de los años 2000, debido al crecimiento acelerado de los datos a almacenar, la concurrencia de usuarios, los millones de transacciones

que se realizan al día contra la base de datos, la necesidad de obtener y mostrar información de manera confiable y rápida, dejan al descubierto las limitaciones del modelo relacional con respecto al almacenamiento masivo de datos, ya que es necesario disponer de servidores muy potentes.

Se clasifican según el tipo de estructura de datos o el esquema en para almacenar los datos

- Basadas en documentos: almacenan información como si fueran documentos. Por lo general utilizan formato JSON o XML. El funcionamiento es muy similar a las clave-valor, sólo que en este caso el atributo clave es el nombre que se le pone al fichero.
- Orientadas a grafos: las relaciones entre nodos pueden tener atributos. Son recomendables para sistemas que se puedan representar en forma de red de manera sencilla.
- Orientadas a columnas: almacenan toda la información en columnas de esta forma las lecturas son muy rápidas, pero se sacrifica mucho tiempo para las escrituras, por lo que no son recomendables a no ser que el número de lecturas sea muy superior al número de escrituras.
- Clave-valor: es la forma más usada, se almacena una clave, y un valor. El valor se suele almacenar como un tipo "BLOB" de esta forma el sistema es independiente al tipo de datos que se quiere almacenar. Su principales características son muy rápidas para las operaciones de lectura/escritura y fácilmente escalables particionando la clave, es decir, se pueden almacenar las distintas claves en distintos servidores dependiendo de su valor inicial. “ (Antaño, Castro y Valencia 2014).

Base	Usuarios
MongoDB	CISCO, FourSquare, MetLife, MTV.
Cassandra	Facebook, Twitter, Spotify, Microsoft, Instragram, GE, Disney, Adobe, ebay.
Big Table	Google
Dynamo	Amazon
Project Voldemort	LinkedIn

Figura 16 - Marcas usuarias de bases NoSQL según (Antaño, Castro y Valencia 2014)

Modelo de base de datos e Internet

A continuación, se detalla brevemente el avance de los sistemas de base de datos a partir de los años 90.

1994 -1997

- se crean las primeras bases de datos en línea, debido a que el incremento del uso de Internet se plasmó en un crecimiento exponencial de las bases de datos. Información obtenida en (<https://www.timetoast.com/timelines/historia-y-evolucion-de-las-bd-03/2023>)

●

1997-2000

- nuevas tecnologías. El aumento en la inversión de negocios por internet incrementó la demanda de conectores de BD (Frontpage, Active Server Pages, Java Servlets, Dreamweaver, ColdFusion, Enterprise Java Beans, y Oracle Developer 2000). El uso de CGI, GCC, MySQL, Apache, y otros sistemas, trajo la solución Open-Source al internet. Con el incremento en el uso de tecnologías Point-of-sale, procesamiento de transacciones en línea, y análisis online, el procesamiento maduró. Se aplica XML al procesamiento de BD. Información obtenida en (<https://www.timetoast.com/timelines/historia-y-evolucion-de-las-bd-03/2023>)

2010

- Sistemas xml: Las bases de datos XML (Extensible Markup Language) forman un subconjunto de bases de datos utilizando el lenguaje XML, que es abierto, legible para humanos y máquinas, ampliamente usado. Información obtenida en (<https://www.timetoast.com/timelines/historia-y-evolucion-de-las-bd-03/2023>))
- Big Data o Datos masivos: es un término que hace referencia a una cantidad de datos tal que supera la capacidad del software convencional para ser capturados, administrados y procesados en un tiempo razonable. El volumen de los datos masivos crece constantemente. Muchos de estos datos no están estructurados y no cumplen con el esquema entidad-relación. Información obtenida de (<https://www.timetoast.com/timelines/evolucion-de-las-bases-de-dat-os-ff60b26f-f1c2-46e3-9dba-3433b60c6bc5> - 03/2023)
- Base de datos en la nube: Los servicios de las bases de datos consisten en un componente de administración que controla las instancias de cada base de datos subyacente utilizando una API de servicios. La API de servicios se expone al usuario final, y permite a los usuarios realizar operaciones de mantenimiento y ampliar sus instancias de la base de datos. Las bases de datos están en los servidores de los proveedores del servicio en la nube. (<https://www.timetoast.com/timelines/evolucion-de-las-bases-de-datos-ff60b26f-f1c2-46e3-9dba-3433b60c6bc5> - 03/2023)

2020

- En la actualidad, las tres grandes compañías que dominan el mercado de las bases de datos son IBM, Microsoft y Oracle. Por su parte, en el campo de internet, la compañía que genera gran cantidad de información es Google. Aunque existe una gran variedad de software que permiten crear y manejar bases de datos con gran facilidad, como por ejemplo LINQ, que es un proyecto de Microsoft que agrega consultas nativas semejantes a las de SQL a los lenguajes de la plataforma .NET. Información obtenida de (<https://www.timetoast.com/timelines/evolucion-de-los-sistemas-de-bases-de-datos-cef83116-5c5a-4f55-810c-968162eb0552-03/2023>)
-

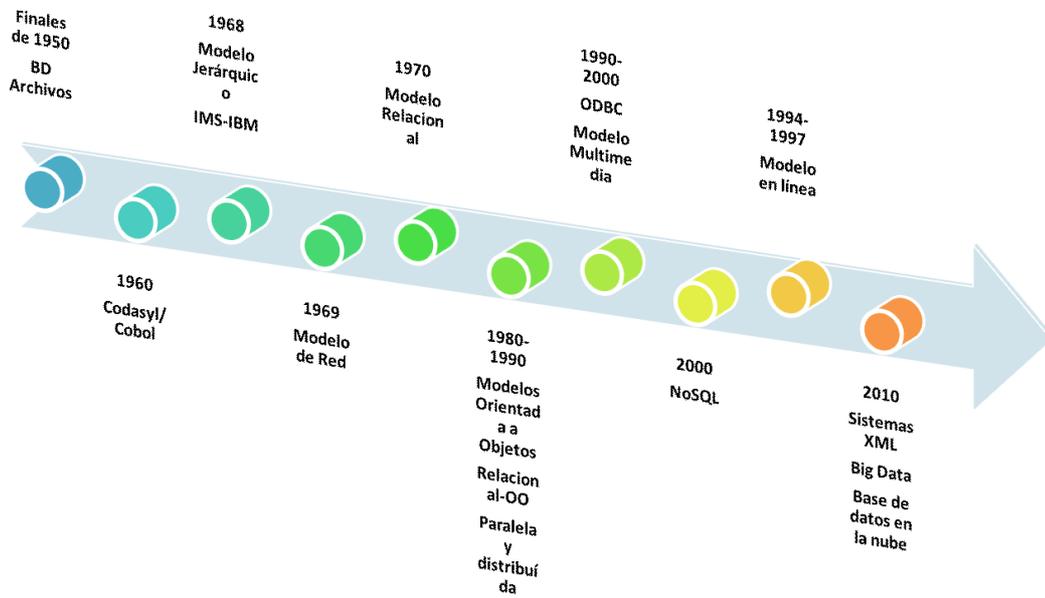


Figura 17 - Modelos de base de datos entre 1950 y 2010

CAPÍTULO 4 - MIGRACIÓN DE BASES DE DATOS HEREDADAS A UNA BASE DE DATOS RELACIONAL - DESARROLLO PROPUESTO.

El proyecto de cambio de sistemas en la Secretaría de Administración del Poder Judicial de la Provincia de Buenos Aires, surge por la necesidad de actualizar la tecnología, dado que el avance exponencial de la misma en los últimos años, provocó que dichas aplicaciones quedaran obsoletas, y por lo tanto hacía dificultosa la tarea diaria de los usuarios. Este cambio, no sólo incluye los programas sino que también la información que ellos contienen, considerada de vital importancia para la organización. El principal desafío del proyecto es efectuar una migración exitosa de los datos, es decir que los mismos deben estar disponibles en las nuevas aplicaciones y deben poder manipularse.

Las áreas y sectores afectados son:

- Área de Compras y Contrataciones e Inmuebles
- Área de Contaduría
- Área de Tesorería
- Sector de Registro Patrimonial
- Sector de Soporte informático de la Secretaría

Los sistemas que intervienen en este proyecto son del año 1993 y están desarrollados en Clipper 5.0 con tablas DBF y, se ejecutan sobre DOS bajo una red Novell. Cada área y sector tienen su aplicación.

Cada una de las aplicaciones tiene sus propios datos, almacenados en diferentes carpetas de red, de manera que la información de cada sistema es independiente. Esto provoca, entre otras cosas, redundancias e inconsistencias. Por ejemplo, cada sistema utiliza su propia tabla de proveedores, con lo cual, es inevitable que se repitan los proveedores (redundancia) y además cabe la posibilidad que no coincida la información del proveedor en las diferentes tablas (inconsistencia).

El sector de Soporte Informático es el encargado de llevar a cabo la actualización de tecnologías, programas y datos.

La figura 18 representa gráficamente la organización de los sistemas por área/sector.

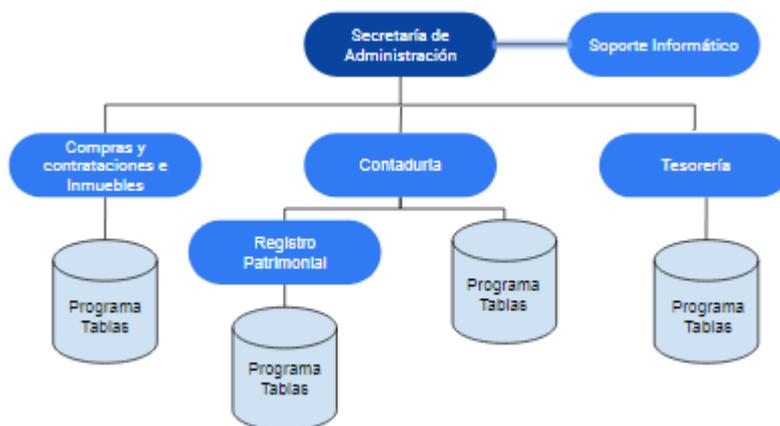


Figura 18 - Organización de las aplicaciones por área/sector

Los programas se ejecutan en la unidad virtual G, pero físicamente hacen referencia a una ubicación distinta en el disco. A los usuarios de cada área/sector, se les asigna un contexto en la red, en este caso Novell. Este contexto, entre otras cosas, “mapea” la unidad G correspondiente.

Área/Sector	Unidad G	Ubicación datos
Contaduría	c:\contadu\adminis	g:\dbf
Compras y Contrataciones e Inmuebles	c:\contrata\adminis	g:\dbf
Tesorería	c:\tesoreria\adminis	g:\dbf
Registro Patrimonial	c:\contadu\rpcl\adminis	g:\dbf

Figura 19 - Mapeo de la unidad G

4.1- INCONVENIENTES DETECTADOS EN LAS APLICACIONES EN MS-DOS

Los inconvenientes técnicos que a continuación se detallan, determinaron que era imprescindible y urgente reemplazar las aplicaciones y el almacenamiento de datos.

- Manejo del teclado: como las aplicaciones se ejecutaban en un entorno MS-DOS, no podía utilizar el mouse. Sólo el teclado. Tarea muy incómoda a la hora de interactuar con el usuario.
- Manipulación de la información: el problema principal se presentó ante la necesidad de obtener información adicional, es decir datos que no se obtienen de la aplicación propiamente dicha, a saber, estadísticas, pedidos de informes para presentar en distintos organismos, o simplemente para su evaluación. Ese manejo externo, se hacía con el utilitario DBU (Data Base Utility), en MS-DOS. Generalmente, era una tarea engorrosa dado que el ambiente es muy poco práctico.

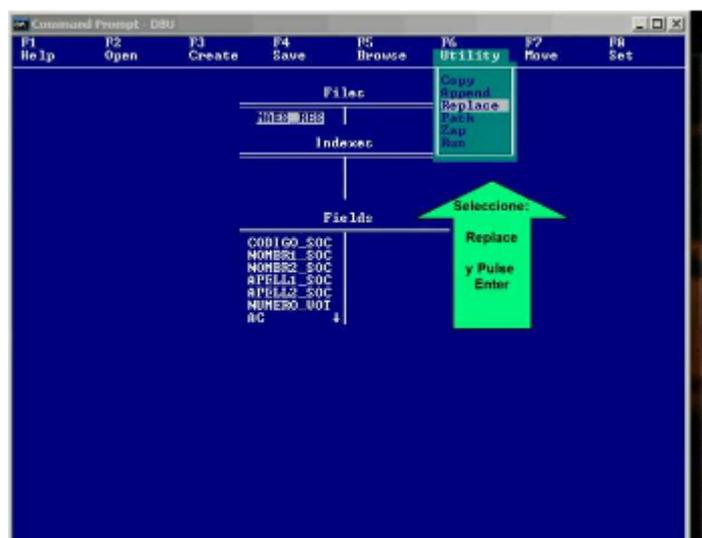


Figura 20 - Ambiente DBU

- Manejo de índices de las tablas y campos memo: cada tabla de datos DBF llevaba asociado, al menos, un archivo de índice (extensiones NTX o CMX) y en el caso de tener campos del tipo memo, un archivo con extensión DBT. Dado que los índices de las tablas DBF eran también archivos, era habitual que se corrompieran, lo que implicaba ejecutar procesos de indexación para regenerarlos. La duración de estos procesos dependía del tamaño de la tabla a la cual indexa. Si la misma era de gran tamaño, la ejecución podía insumir horas. Como consecuencia de esto, se generaba una demora en las tareas del usuario, ya que durante la indexación no podía utilizarse la aplicación en las cuales intervenían las tablas y los índices.

Con respecto a los campos del tipo memo, se crea el archivo con extensión DBT para aquellas tablas que los contienen. Cuando el tamaño de algunos de los archivos DBT crecía en demasía, comienzan a perderse los datos, lo que significaba realizar una depuración de la información de la tabla principal DBF para poder disminuir el tamaño del archivo DBT.

Ejemplo:

ARCHIVO	TIPO
PLANI_A3.DBF	TABLA DE DATOS
PLANI_A3.CMX	ÍNDICE
PLANI_A3.DBT	CAMPOS MEMO

- La crisis Y2K (año 2000): El formato utilizado en todas las aplicaciones era dd/mm/yy. El principal problema surgió cuando se trataba con fechas del año 2000, Clipper 5.0 asume que yy=00 corresponde al año 1900, yy=01 al 1901, etc. Esto obligó a diseñar rutinas especiales específicamente para todos los procesos que involucran fechas, tales como, ordenamientos o comparaciones, para poder evitar datos erróneos y resultados inconsistentes.
- Manejo de impresoras: estas aplicaciones desarrolladas en Clipper 5.0 manejaban las impresiones enviándolas directamente al puerto paralelo de la impresora del equipo, esperando que sean LPT1,LPT2,etc. Con la modernización de los equipos, desaparecieron los puertos paralelos y las nuevas impresoras adquiridas se conectan al equipo a través de los puertos USB y directamente imprimen mediante la RED. Como solución a este problema se optó por redirigir las impresiones a un puerto USB. Para ello se desarrollaron procesos por lotes (extensión BAT) con el comando de NET USE de MS-DOS NET USE que debían ejecutar previo a la utilización de la aplicación.

```
REM Para liberar el puerto de alguna otra captura
NET USE LPT1 /D

REM Captura de la impresora
NET USE LPT1 \\Soporte1\HPLaserJET

REM Soporte1: Host del recurso compartido (impresora)
REM HPLaserJET: Nombre del recurso compartido
```

Figura 21 - Captura de Impresora

Esto ocasionó muchos inconvenientes. Por ejemplo, los usuarios olvidaban ejecutar el proceso de captura de la impresora y lo detectaban en el momento de imprimir con lo cual debían cerrar la aplicación, capturar la impresora y volver a realizar todos los pasos hasta llegar a la pantalla en la que estaban trabajando (solución muy poco práctica). Además, en varias oportunidades, fallaba la captura y tampoco podían imprimir. Todo esto, significaba retrasar el trabajo y generar malestar en los usuarios por la pérdida de tiempo.

Este problema llegó a una instancia en que no bastaba la redirección de la impresión a un puerto USB. Ya no se podía imprimir en impresoras de marca EPSON (Clipper 5.0 no reconocía los códigos de control de impresión). Con lo cual se actualizaron algunos listados (los más frecuentes), en los programas fuentes, reemplazando los códigos de impresión originales por los códigos equivalentes que ofrecía la marca HP. No fue una tarea fácil, ya que insumió mucho tiempo de investigación y pruebas.

- Aparición de los procesadores de 64 bits: estas aplicaciones, que eran de 16 bits, dejaron de funcionar en equipos con procesadores de 64 bits. En 64 bits sólo funcionan programas de 32 y 64 bits.

Nuevamente surgieron graves problemas, no podían ejecutarse los programas de gestión. Luego de una ardua tarea de investigación y en tiempo récord se decidió utilizar máquinas virtuales HyperV (MV), donde se instaló Windows 7.

A cada usuario de los programas de gestión (más de 70 personas), se les creó en sus equipos la MV que les permitía el acceso al módulo

correspondiente (Compras y Contrataciones e Inmuebles, Contaduría, Tesorería, Servicios Públicos,).

Esta solución, si bien permitió seguir trabajando, consumía mucha memoria, entonces su funcionamiento dependía de la cantidad de aplicaciones ejecutándose al momento de querer usar la MV. En caso que no funcionara, había que reiniciar las máquinas y sólo ejecutar esta aplicación.

4.2- SOLUCIÓN PROPUESTA Y DESARROLLADA

La propuesta consistió en construir aplicaciones básicamente con las siguientes características:

- acceder a datos centralizados en una base de datos
- conectarse entre sí evitando cargar información compartida varias veces.
- aprovechar al máximo las herramientas propias del SQL: integridad de los datos, definición de claves, índice, triggers, store procedures, seguridad, transacciones.
- brindar una interfase amigable con el usuario
- acceder a las utilidades del ambiente Windows, desde utilizar un mouse a la administración de impresoras

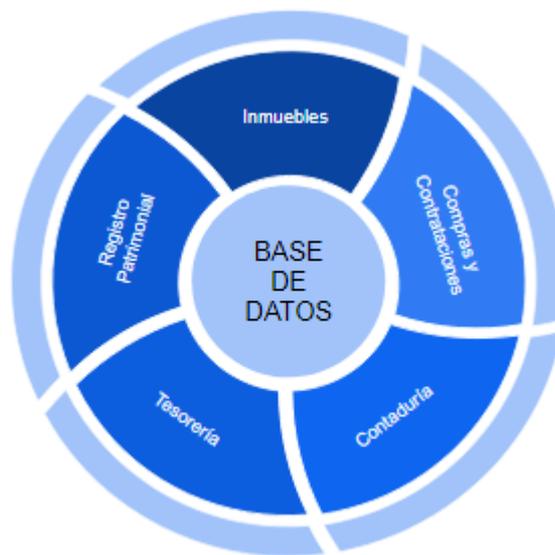


Figura 22 - Organización de las aplicaciones bajo Windows

Debido a que el tiempo y recursos eran escasos, se decidió reemplazar los sistemas por aplicaciones que se ejecuten bajo windows, desarrolladas en Borland Developer Studio 2006 utilizando bases de datos relacionales (Microsoft SQL) bajo Windows. La implementación y puesta en marcha se hizo progresivamente.

Aplicación	Año (puesta en marcha)
Inmuebles	2016
Contaduría-Rendiciones	2017
Compras y Contrataciones	2019
Contaduría -Pedidos de Fondos	2022
Contaduría-Alquileres	2022

4.3- ETAPAS EN LA MIGRACIÓN

4.3.1- Relevamiento de la información

El relevamiento de la información estuvo a cargo del sector de Soporte Informático. Dado que este sector también era el responsable del mantenimiento de las aplicaciones a reemplazar, esta etapa no presenta mayores dificultades.

4.3.2- Transformación/mapeo de datos

En esta etapa se definieron las estructuras de las aplicaciones nuevas utilizando Microsoft Sql Server Management Studio.

El SQL, ofrece una variedad de tipos de datos mucho mayor que la que está disponible para las tablas DBF, se analizaron, en algunos casos, si era necesario o conveniente redefinir tipos de datos. Por ejemplo:

Se reemplazó el tipo Alfanumérico (dbf) por el VARCHAR de SQL. La ventaja de usar el tipo VARCHAR es que el tamaño de almacenamiento de un valor VARCHAR es la longitud real de los datos almacenados más 2 bytes.

VARCHAR(n), donde n define la longitud de la cadena.

VARCHAR(max), donde max es la longitud máxima del tipo VARCHAR (2 GB), se utilizó para los campos memo .

En las tablas de SQL se incorporan campos tipo VARBINARY(max), que se utilizan para almacenar archivos (pdf,word,excel..etc). Esto representa una nueva funcionalidad en los sistemas a implementar.

Otra característica de las tablas principales en SQL es que tienen definido un campo ID (generalmente autoincremental) que identifica unívocamente el registro y que es usado para relacionarse con otras tablas.

4.3.3- Depuración de datos

Como se ha mencionado anteriormente, los sistemas datan del año 1993. En el año 2000, el Poder Judicial de la Provincia de Buenos Aires se dividió en Administración de Justicia (formado por Corte Suprema de Justicia, Cámaras de Apelación, Jueces de Primera Instancia y demás tribunales y jueces que establezca la ley) y Ministerio Público (formado por Procurador y Subprocurador General de la Suprema Corte de Justicia, Fiscales de Cámaras, Agentes Fiscales, Asesores de Menores y Defensores de Pobres y Ausentes, Fiscal de Casación y del Defensor de Casación.).

La primera depuración realizada fue sólo migrar los datos correspondientes a Administración de Justicia.

Otra depuración que se hizo fue la unificación de los proveedores. En los sistemas originales, los proveedores se identificaban con un código. En las aplicaciones nuevas se define como identificador de la tabla el cuit del proveedor. Cada sistema, como se explica al comienzo de este capítulo, tenía su tabla de proveedores. Entonces la tarea se definió básicamente en 2 pasos:

- armar un listado con la unión de las tablas de proveedores, relevando el cuit para cada uno. Como resultado, se encontraron muchas inconsistencias (misma razón social con más de un cuit, proveedores con un mismo cuit con diferente razón social, etc)
- en base al resultado del punto anterior, estudiar cada caso para reducir la información a un solo registro por proveedor.

4.3.4- Procesos de Migración.

Los procesos de migración se desarrollaron, en su mayoría, en Borland Developer Studio 2006, además en Clipper 5.0 y stored procedure de SQL.

Aplicación de Gestión de Inmuebles.

La entidad principal de la aplicación es un trámite. Tiene asociado tipo de trámite, qué solicita, quién lo solicita (dependencias solicitantes), para quién se solicita (dependencias destino), qué rubro y un seguimiento con fechas de las diferentes etapas. Además, en el caso de alquileres maneja la información de los propietarios y que inmueble está involucrado y en el caso de una compra de inmueble, contiene los datos relacionados con la escritura del mismo.

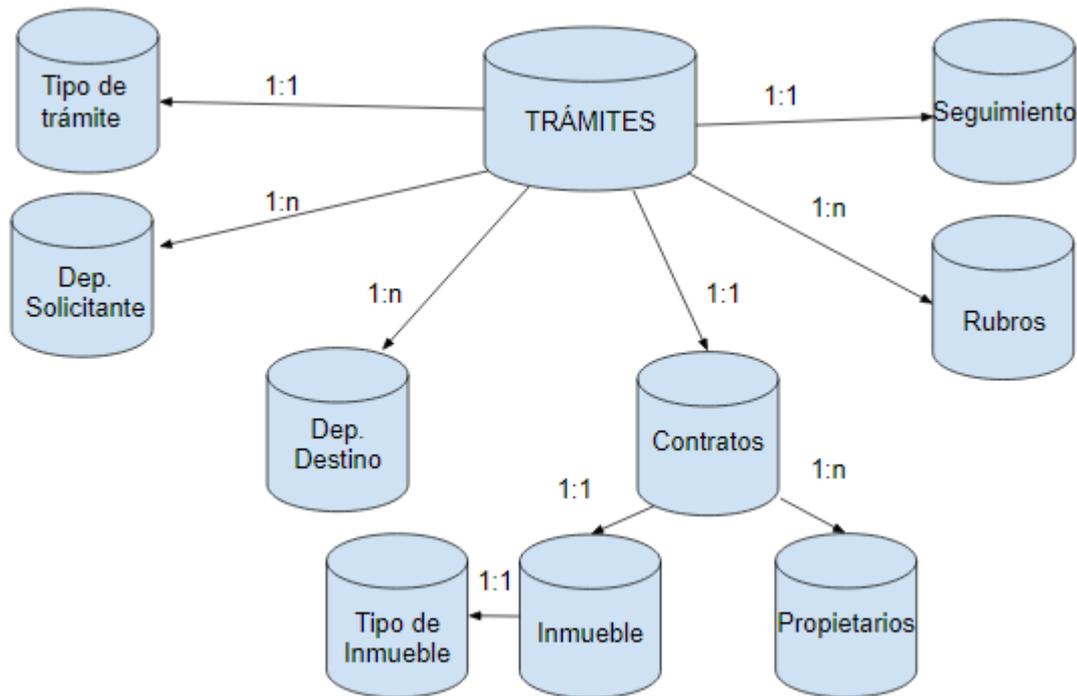


Figura 23 - Representación gráfica de las relaciones de un trámite

La migración definida para la aplicación, al ser la primera, fue la más costosa, en cuanto a tiempos de análisis, investigación y programación.

Los primeros procesos de migración correspondieron a las tablas básicas para la aplicación; aquellas a las cuales hacían referencia las tablas principales. Por ejemplo, proveedores, dependencias, tipo de trámite, rubros, provincias, etc.

Las tablas DBF del tipo código - descripción, se abrieron con excel y luego se importaron desde SQL.

La complejidad se dio con aquellas tablas a las que se les asociaron relaciones 1..n.

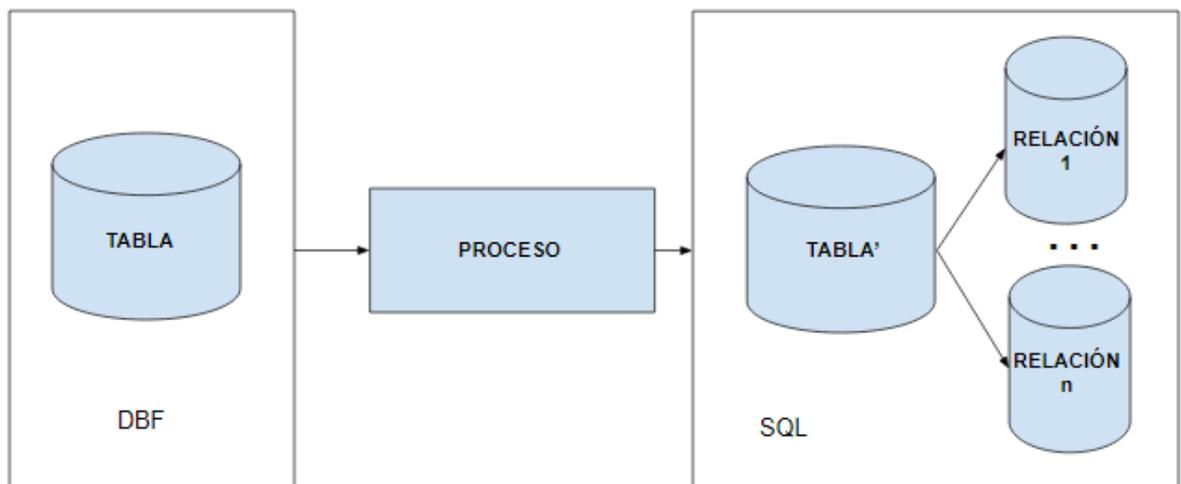


Figura 24 - Proceso para migrar tablas con n relaciones

Un caso típico es la tabla de proveedores DBF, contenía toda la información en un único registro. En el caso de domicilios, sólo permitía almacenar uno y los mismo ocurría con los rubros. En la nueva aplicación se agregaron tablas para tipos de domicilios del proveedor (comercial, legal, notificaciones, real) y rubros del proveedor. Dando así la posibilidad de asignar varios domicilios y varios rubros.

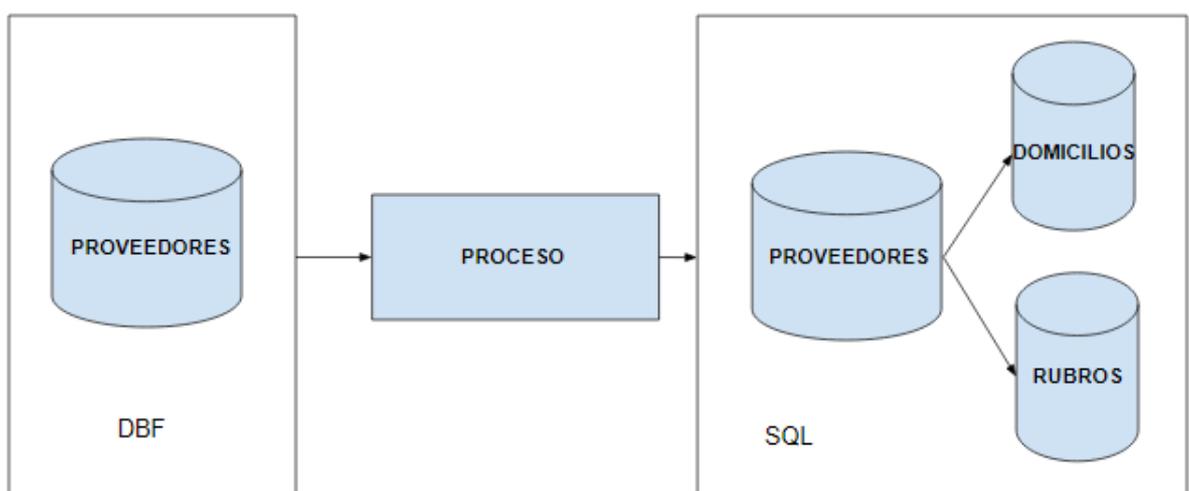


Figura 25 - Proceso para migrar PROVEEDORES

Además se detectaron problemas en el tratamiento de los campos tipo memo. No bastaba una simple asignación. Se resolvió recorriendo el campo origen, particionarlo en strings de 250 caracteres (hasta completar la longitud del

memo) para luego concatenarlo en un campo auxiliar. Por último se asignaba ese campo auxiliar al campo destino.

Por ejemplo, en la tabla ORDENES.DBF había dos campo memo, observ y lugar_ent

Previo al proceso, se agregaron a la tabla los campos obs1,obs2,obs3 de longitud 250, y los campos lugar_1 y lugar_2 de 250 caracteres. Cada uno de estos campos se instanciaron de la siguiente manera

```
obs1 = substr(observ,1,250)
obs2 = substr(observ,251,250)
obs3 = substr(observ,501)
```

```
lugar_1=substr(lugar_ent,1,250)
lugar_2=substr(lugar_ent,251)
```

Resumiendo, lo que se almacenó en el campo destino fue obs1+obs2+obs3 y lugar_1+lugar_2 respectivamente.

Aplicación Registro Único de Inmuebles

En el año 2016, debido a que los datos del área de Inmuebles habían sido migrados a SQL, la Secretaría de Administración junto con la Dirección de Arquitectura de la Secretaría de Planificación, implementaron un Registro único de Inmuebles. Se trata de una base de datos SQL en otro servidor que recolecta toda la información de los inmuebles propios y no propios a los cuales se accede por una aplicación web. Desde entonces la Secretaría de Administración tiene la responsabilidad de completar esta base de datos con todo lo referente a contratos, servicios, situación jurídica, etc. Técnicamente se establece una conexión entre los servidores, tarea imposible de hacer con las tablas DBF.

Aplicación para gestionar rendiciones en el área de Contaduría.

Básicamente, una rendición se refiere a toda información que las áreas y las Delegaciones de Administración deben cumplimentar para justificar el manejo de dinero. El sector de Rendiciones controla dicha información y, en caso de estar todo bien, la envía a la Delegación de Contaduría de la Provincia para su aprobación.

Hasta el año 2017 todo era un trabajo manual que se registraba en un archivo excel y se remitía toda la información en papel. Para agilizar este proceso, se generó una base de datos SQL con los datos importados desde el excel y se desarrolló una aplicación que gestiona todos estos datos y permite almacenar los documentos asociados (por ejemplo facturas).

De esta manera, el circuito se convirtió totalmente digital. La Delegación de Contaduría de la Provincia tiene acceso a la aplicación y hace su control sobre la base de datos y no sobre los papeles.

Aplicación de Gestión de Compras y Contrataciones

Los procesos de migración de datos para esta aplicación son los mismos que para Inmuebles, con ciertas adaptaciones, como los nombres de tablas.

Como en Inmuebles, la entidad principal es el trámite. Lo que los diferencia es que existen más funcionalidades en este sistema.

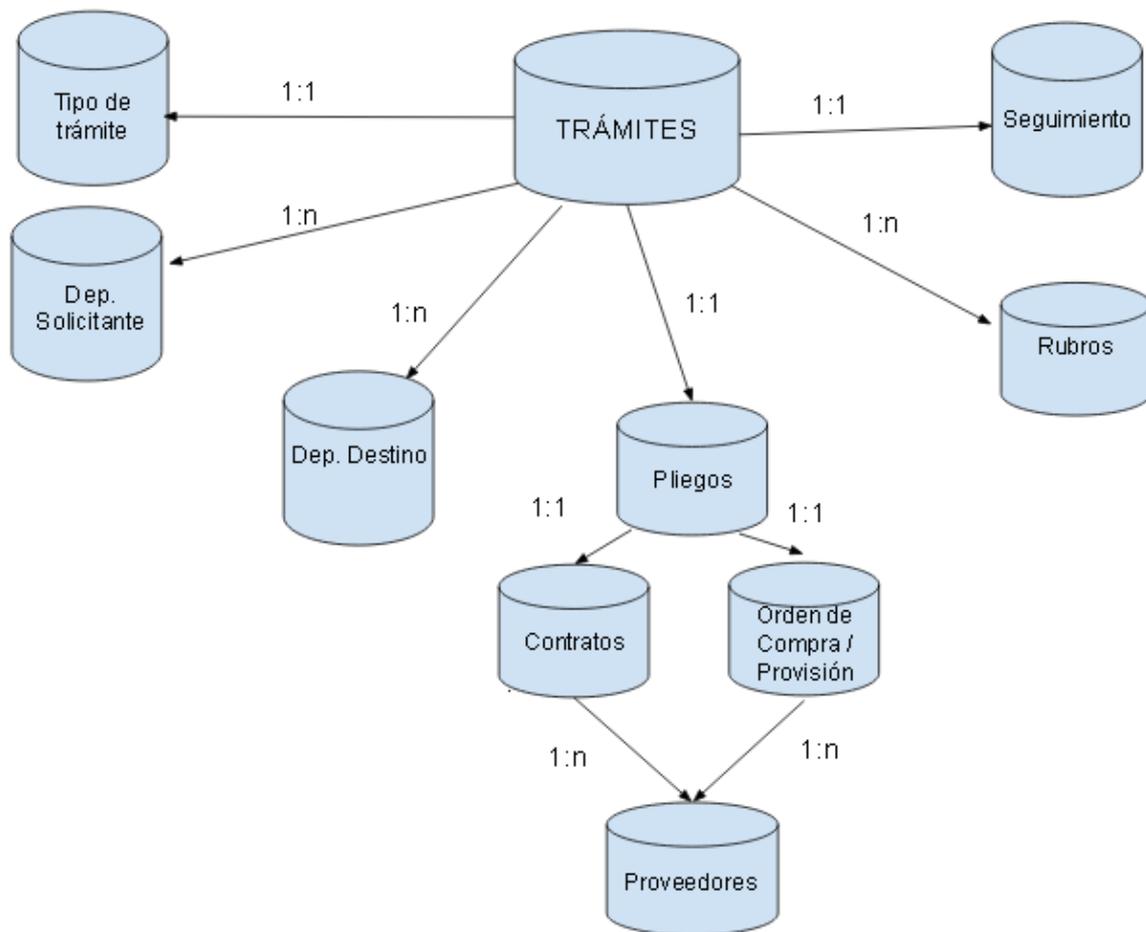


Figura 26 - Representación gráfica de las relaciones de un trámite

4.3.5- Pruebas

Una vez comprobado que el proceso de migración de datos funcionaba correctamente, se procedió a definir el plan de pruebas para todas las migraciones. Consistía en evaluar:

- La cantidad de registros de la tabla origen cuyos datos cumplieran la condición para migrar, debía ser la misma que en la tabla destino. Los procesos tenían contadores en el origen y en el destino y al final del proceso debían coincidir. Caso contrario, se analizaba la diferencia, se corregía y se ejecutaba nuevamente el proceso.
- La información migrada debía ser correcta, es decir, el valor de los datos, tanto del origen como el destino, debía coincidir. Para esto, se armaron

lotes de prueba, con todos los casos posibles, y luego se controlaba registro por registro. Si algún error era detectado, se corregía el proceso y se repetía la migración.

4.3.6- Simulación de carga.

El objetivo de esta simulación era comprobar que el aplicativo nuevo respondiera correctamente a las funcionalidades implementadas. En caso de detectar errores, debía corregirse el proceso correspondiente. Por otro lado, surgieron nuevas funcionalidades, producto de los beneficios de la base de datos relacional (nuevos campos, nuevos cálculos, nuevas consultas). Muchas de ellas se desarrollaron previo a la implementación y otras se programaron para una segunda etapa, con la aplicación en producción.

4.3.7- Cargas reales

En esta etapa, los usuarios ya disponían de la aplicación instalada, entonces se les solicitó que durante un tiempo, todo lo que registraba en el sistema anterior, se resolviera también en el nuevo. Se detectaron algunos errores los cuales fueron solucionados.

4.3.8- Revisión de resultados

Una vez comprobado que los datos habían sido migrados consistentemente y que la aplicación respondía correctamente a las funcionalidades, se procedió a la puesta en producción del nuevo sistema

El sistema de Inmuebles comenzó a funcionar el 1 de enero de 2016, el sistema de Rendiciones de Contaduría comenzó a funcionar en 1 de enero de 2017, en tanto que el sistema de Compras y Contrataciones comenzó a funcionar el 1 de enero de 2019.

4.4- Inconvenientes durante la migración de datos e implementaciones de los aplicativos

Los inconvenientes técnicos descritos anteriormente, que derivaron en la decisión de reemplazar los sistemas y el almacenamiento de los datos, no fueron los únicos. Se presentaron otros inconvenientes que complicaron las tareas y en cierta forma las retrasaron.

A continuación se enumeran los inconvenientes más significativos.

- Grupo de trabajo. El área de Soporte Técnico de la Secretaría de Administración está conformada por dos (2) personas. Hasta el día de hoy, ambas están abocadas a dar soporte a los usuarios de todos los sistemas que funcionan en la Secretaría. Este soporte es muy amplio, abarca desde cuestiones referidas a las aplicaciones hasta instalar impresoras, por ejemplo. Cabe aclarar que también reciben soporte los usuarios de las veinte (20) Delegaciones de la Secretaría de Administración que están repartidas en toda la Provincia de Buenos Aires. Para este proyecto se asignó una (1) persona que, en caso de ser necesario, era asistida por la segunda.
- Poco compromiso de los usuarios. Todos se quejaban por la complejidad en el trabajo con las aplicaciones anteriores, pero, argumentando falta de tiempo, no cumplían con las pruebas o tareas que se les solicitaban. Por otro lado, era poco el apoyo que daban los responsables de las áreas y sectores. Por ejemplo, las aplicaciones de Inmuebles y Compras y contrataciones, se pudieron poner en producción en el 2016 y 2017 respectivamente, en su tercer intento.
- Sistema Integrado de Gestión Administrativa Financiera (SIGAF). En el año 2017, la Contaduría General de la Provincia anunció que con el ejercicio 2018, todas las dependencias debían utilizar el sistema SIGAF reemplazando al Sistema Financiero y CRT, con el fin de un mayor control sobre el uso de las Finanzas Públicas Provinciales. La puesta en funcionamiento llevó meses, debido a que, a nivel general, no funcionaban correctamente ciertos procesos y en particular, para el Poder Judicial, al no ser una dependencia del Gobierno Provincial, faltaron algunas funcionalidades. Si bien estaban los “facilitadores” que asistían a los usuarios, en un primer momento, no había quién tuviera el conocimiento necesario para resolver.

A raíz de la implementación del SIGAF se debió modificar el proyecto original de la migración de las aplicaciones en DOS. Una de las razones fue la falta de recursos, ya que los mismos estaban asignados a dicha implementación. Otro motivo fue que las funcionalidades del SIGAF reemplazarían algunas de las aplicaciones en DOS y por lo tanto no era necesario reemplazarlas, por ejemplo el módulo de Registro Patrimonial.

- Módulo de Registro Patrimonial (RP). Como todas las dependencias de la Contaduría Gral de la Pcia, usaban los mismos sistemas para RP la migración de datos de las mismas estuvo a cargo del departamento de Sistemas de la Contaduría; pero para el Poder Judicial, fue necesario definir una migración especial. La Contaduría exigía datos que no manejaba el Poder Judicial, con lo cual se definieron valores default para resolverlo. Además, el Poder Judicial manejaba saldos iniciales (que representaban bienes y obras muy antiguas, que no podían ser identificados). La contaduría tuvo que desarrollar procesos especiales para poder incorporar dichos saldos, ya que la aplicación SIGAF no los contemplaba). El problema fundamental consistía en que recién luego de migrar los datos patrimoniales (bienes muebles, bienes inmuebles y obras), se podía desconectar la aplicación DOS.

CAPÍTULO 5 - RESULTADOS OBTENIDOS

A continuación, dada la experiencia obtenida durante todo el desarrollo e implementación del caso presentado, se describen los resultados tomando dos(2) puntos de vista: sobre el manejo de la información y sobre las nuevas aplicaciones.

5.1- Manejo de la información

En este trabajo se considera que el principal beneficio producido por la migración es que toda la información está en un mismo entorno: una base de datos relacional - SQL. Como se detalló en capítulos anteriores, se tiene acceso a todos los beneficios de este tipo de base: integridad de datos, relaciones entre tablas a través de la definición de claves, definición de procesos almacenados, definición de vistas, definición de consultas, velocidad para acceder a los datos a través de índices,etc, uso de transacciones, etc.

Otro beneficio es lograr interconectar bases de datos, tanto para el caso de que la información provenga de más de una base o se trate de información de otras Secretarías.

El siguiente ejemplo, corresponde a una aplicación web desarrollada entre la Secretaría de Administración y la Secretaría de Planificación. En la figura X , la información “expedientes relacionados”, “información catastral” y “servicios públicos”, se extrae on line de la base de datos de Inmuebles ubicada en un servidor de Administración, mientras el resto de los datos están en una base de datos en un servidor de Planificación

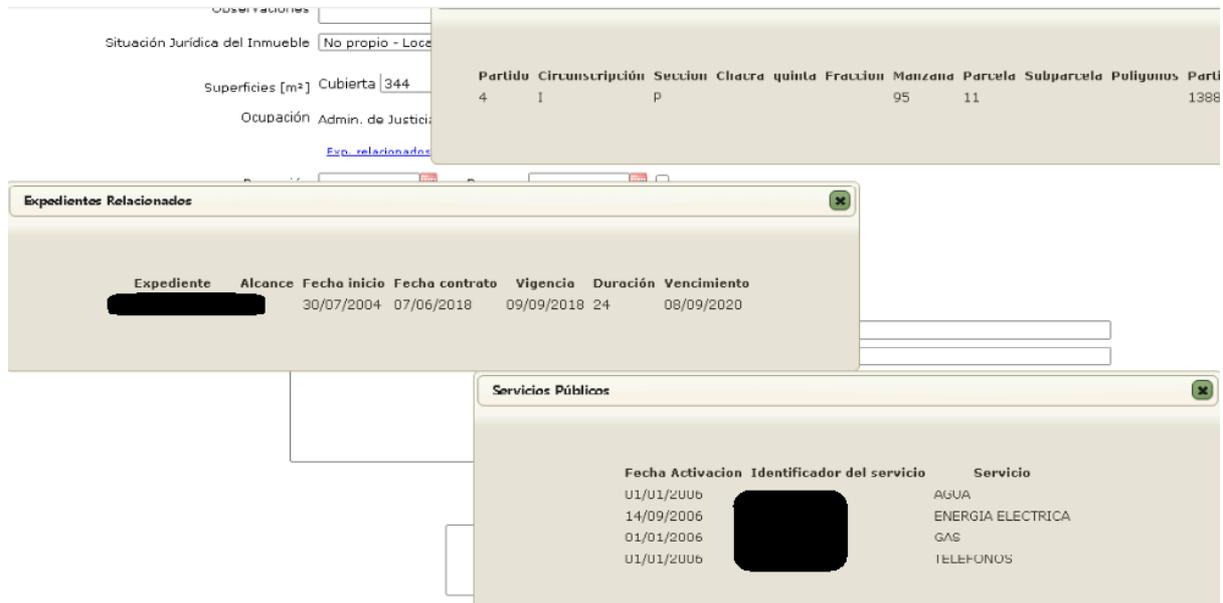


Figura 27 - Ejemplo de conexión entre bases de datos

Además se cuenta con la posibilidad de hacer consultas complejas para obtener información de manera efectiva y rápida. El siguiente ejemplo, genera una consulta relacionando cuatro(4) tablas, devolviendo como resultado campos pertenecientes a las respectivas tablas. A su vez, esta consulta puede manipularse como una tabla cualquiera.

```
SELECT CP.idcontrato, CP.proveedor, C.fechaContrato, C.vigencia, C.vencimiento, C.monto,
CAST(SUBSTRING(C.descripcion, 1, 255) AS varchar(255)) AS descripcion, TT.TIPO, T.nro_tramite,
T.anio_tramite, T.expte_prefijo, T.expte_nro, T.expte_anio, T.expte_alcance
FROM CONTRATO_PROVEEDORES AS CP
LEFT OUTER JOIN CONTRATO AS C ON CP.idcontrato = C.id
LEFT OUTER JOIN TRAMITES AS T ON C.idTramite = T.id
LEFT OUTER JOIN TIPO_TRAMITE AS TT ON TT.ID = T.idTipo
WHERE (T.idClasificacion = 5) AND (NOT (C.vigencia IS NULL)) AND (C.fechaDesocupa IS NULL) AND
(C.fechaRenovacion IS NULL) AND (C.FechaEscritura IS NULL) AND (C.monto > 0) AND (NOT
(C.fechaContrato IS NULL))
```

Figura 28 - Ejemplo de generación de consulta en SQL

Se utilizan vistas, principalmente para referenciar las tablas comunes, tales como DEPENDENCIAS, PROVEEDORES, DEPARTAMENTOS, ETC, ubicadas en una única base de datos. El ejemplo de la figura x define la vista DEPARTAMENTO en la

base de datos para el área de Contaduría, utilizando la tabla de DEPARTAMENTOS que está en otra base de datos, dentro del mismo servidor.

```
SELECT  cod_departamento, (CASE WHEN cod_departamento = 9800 THEN 'TESORERIA' ELSE
descripcion END) AS descripcion, nem, 'PRG' AS aop, (CASE WHEN cod_departamento = 9800 THEN
'090' ELSE aop_nro END) AS aop_nro, vigente, instalacion
FROM    SADM_REAL.dbo.DEPARTAMENTOS AS DEPARTAMENTOS_
```

Figura 29 - Ejemplo de Vistas

Otras ventajas son los Stored Procedures, triggers y functions, que permiten ejecutar procesos sobre las bases de datos independientemente de los programas fuente de las aplicaciones.

```
ALTER PROCEDURE [dbo].[ABMTIPOS PF]
@accion char(1),
@id integer,
@descripcion varchar(100),
@idtipoord integer
AS
BEGIN
if (@accion = 'A')
INSERT INTO [dbo].[TIPOS_PF]
([DESCRIPCION]
,[IDTIPOORD])
VALUES
(@descripcion
,@idtipoord)
else if (@accion = 'M')
UPDATE [dbo].[TIPOS_PF]
SET [DESCRIPCION] = @descripcion
,[IDTIPOORD] = @idtipoord
WHERE id=@id
else if (@accion = 'B')
DELETE FROM [dbo].[TIPOS_PF]
WHERE id=@id
END
```

Figura 30 - Ejemplo de Stored Procedure

```
ALTER TRIGGER [dbo].[borra_orden]
ON [dbo].[OP]
AFTER delete
AS
BEGIN
declare @nid integer
declare @nidPF integer
select @nid=id, @nidPF=IDPF from deleted
/*actualiza pedido de fondos */
if (not @nid is null)
begin
delete from PF_DETALLE
where IDPF=@nidPF and idopdetalle=@nid
end
/* actualiza tablas DIFERENCIAS */
UPDATE DIFERENCIAS
set IDOP=null
where IDOP=@nid
/* borra de la tabla diferencias_detalle*/
delete from diferencias_detalle
where idop=@nid
END
```

Figura 31 - Ejemplo de Trigger

```
ALTER FUNCTION [dbo].[fEstado]
(
    @id integer,
    @estado varchar(1)
)
RETURNS varchar(20)

begin
declare @Resultado varchar(20);
SET @Resultado = (SELECT case when t.estado = 'P' then 'Pendiente' else
case when t.estado = 'U' then 'Publicado' else
case when t.estado = 'A' then 'Para publicar' else
case when t.estado = 'D' then 'Adjudicado/Finalizado' else
case when t.estado = 'R' then 'Rechazado' else
case when t.estado = 'E' then 'Desierto' else
case when t.estado = 'S' then 'Se dejo sin efecto' else 'Otro' end
end
end
end
end
end
end
end
from TRAMITES t
where t.id=@id)
return @resultado
end
```

Figura 32 - Ejemplo de Function

5.2- Nuevas aplicaciones.

La migración de los datos al entorno relacional, es el puntapié inicial para el desarrollo de nuevas aplicaciones. La metodología general consiste en definir para cada área una base de datos y, como se mencionó anteriormente, un entorno global con las tablas comunes a todas las áreas, todas ubicadas dentro del mismo servidor.

Además de las bases de datos migradas, las aplicaciones implementadas contemplan la misma funcionalidad que las que se ejecutan en DOS. Pero utilizando las ventajas del modelo relacional y el ambiente WIndows, se fueron agregando nuevas funciones. Otro avance significativo son las consultas, ya que con la utilización de filtros pueden resolver situaciones tan complejas como se requiera.

5.2.1- Inmuebles

Esta aplicación gestiona toda la información de los inmuebles (propios y alquilados) relacionados con el Poder Judicial.

Internamente se conecta con diferentes bases de datos:

- base de datos de Publicaciones, desde inmuebles se puede publicar o quitar la publicación, los pliegos para pedidos de oferta en la página del Poder Judicial (www.scba.gov.ar)
- base de datos del área de Despacho de la Secretaría de Administración, para relacionar los trámites con los expedientes.
- base de datos de la Secretaría de Planificación, para obtener información específica de los inmuebles.
- base de datos de Contaduría, para actualizar ciertos cambios en los contratos de alquiler.

La figura 33 muestra gráficamente cómo son las conexiones de las bases de datos

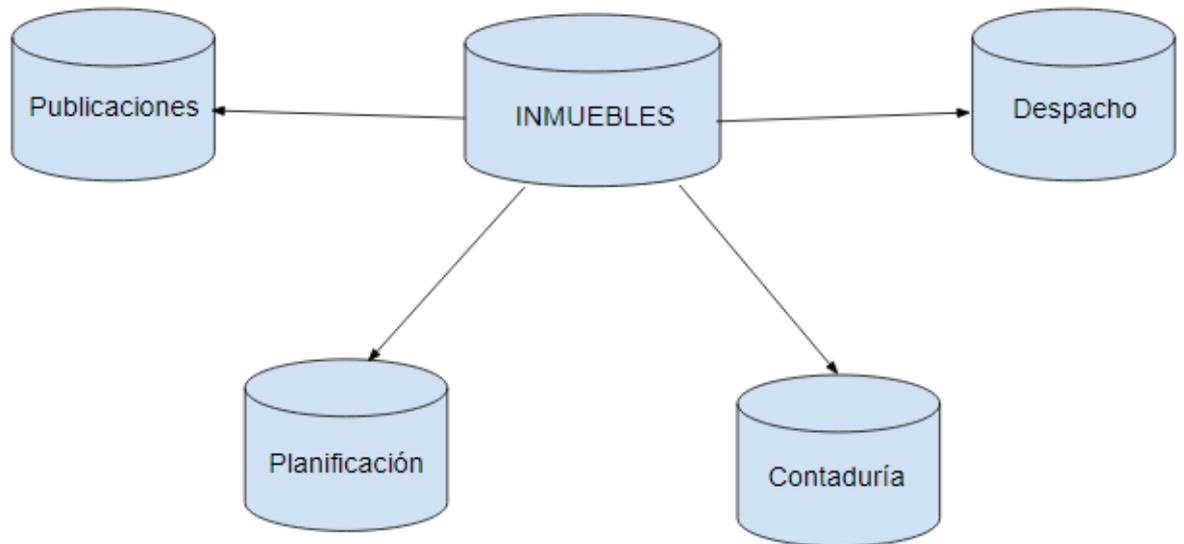


Figura 33 - Inmuebles - Conexiones entre Bases de Datos

A continuación la figura 34 muestra un ejemplo de un filtro para la consulta de contratos.

Consulta de Contratos

Clasificación: [] Servicio / Obj. de Locación: []

Propietario / Firma / Comodante: []

Depto. Judicial: [] Incluye Depto. completo Dependencia: []

Tipo Trámite: []

Vencimiento: [] y [] Desocupación: [] y [] Adjudicación: [] y [] Rel. locativa: [] y []

Vigencia: [] y [] Contrato: [] y [] Rescisión Anticipada: [] y []

Sólo vigentes al 20/04/2023

[Consultar] [Salir]

Trámite	Expte	Servicio	Fecha	F.Pago

CUIT	Proveedor	ACE/PRG	NRO	Dependencia	Departamento

Cantidad de Trámites: 0 Total \$: 0 Total m2: 0 Valor promedio m2: 0

Figura 34 - Inmuebles - Consulta de Contratos

5.2.2- Compras y Contrataciones

Esta aplicación gestiona toda la información referida a los procesos de compra de bienes muebles, contrataciones de servicios y obras del Poder Judicial.

Internamente se conecta con otra base de datos.

- base de datos de Publicaciones, desde Compras y contrataciones se puede publicar/despublicar, los pliegos para las distintas licitación en la página del Poder Judicial (www.scba.gov.ar).
- base de datos del área de Despacho de la Secretaría de Administración, para relacionar los trámites con los expedientes.

La figura 35 - muestra gráficamente cómo son las conexiones de las bases de datos

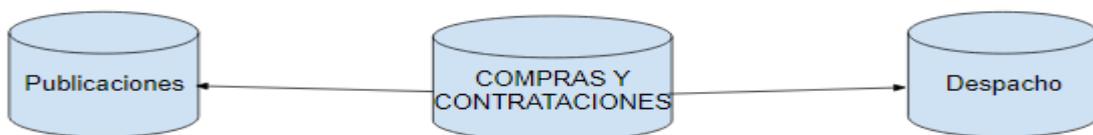


Figura 35 - Compras y contrataciones - Conexiones entre Bases de Datos

A continuación la figura 36 muestra un ejemplo de un filtro para la consulta de órdenes de compra.

Tipo	Orden Nº	Fecha	Expte	Alic.	Firma	Importe\$	Recepción	Pl.	
Cantidad de Ordenes: 0									
						\$:	0.00		

Figura 36 - Compras y contrataciones -Consulta de órdenes de compra

5.2.3- Contaduría.

En apartado 4.4, se describe la implementación del Sistema Integrado de Gestión Administrativa Financiera (SIGAF). Un pedido de fondos, es la solicitud a la Contaduría de la Provincia de Buenos Aires de fondos, a través del SIGAF, para pagar las obligaciones (alquileres, proveedores, obras, etc)

La aplicación desarrollada, tiene como objetivo poder cargar el detalle de los pedidos de fondos y el detalle de los alquileres pagados, ya que el SIGAF, sólo permite ingresar la información resumida. Es decir, el pedido de fondos en el SIGAF se hace por un importe, pero no permite desglosar cómo se utiliza ese importe.

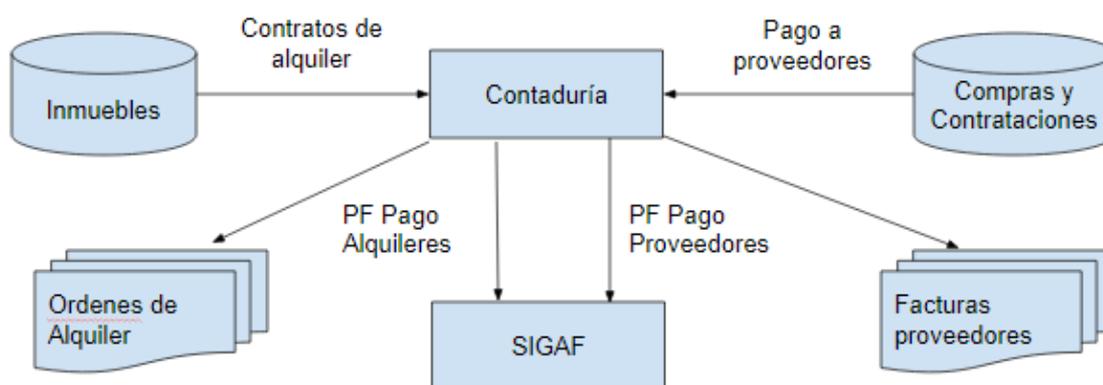
Cuáles son las funciones de esta aplicación:

- Contaduría-Inmuebles: en el sistema DOS, se registraban los contratos de alquiler en cada sistema, es decir, se duplica la información. En el nuevo sistema, la base de datos de Contaduría, genera las órdenes de pago con una consulta que “devuelve” la información de contratos vigentes que está disponible en la base de datos de Inmuebles. Es decir, dicha información está en un solo lugar.
- Contaduría-Compras y Contrataciones: en el sistema DOS, la información se duplica, dado que, como se dijo anteriormente, cada sistema tenía sus

propios datos. Por ejemplo, en la aplicación anterior de Compras y Contrataciones se cargaban facturas de proveedores para el pago, que luego eran nuevamente ingresadas al sistema de Contaduría. Como en el punto anterior, en el nuevo sistema, la información se carga a través de una consulta que se ejecuta en la base de datos de Contrataciones.

- Contaduría calcula, entonces, el importe total que debe remitirse al SIGAF para su aprobación, por cada pedido de fondos para alquileres y pago a proveedores respectivamente. Además almacena en su base datos, el detalle correspondiente a las órdenes de alquileres y pago a proveedores.

La figura 37 grafica la aplicación del área de Contaduría



PF: pedido de fondos

Figura 37 - Aplicación área Contaduría

Una de las ventajas de esta aplicación es poder filtrar la información por varios criterios combinados mediante consultas.

En el caso de Orden Pago de Alquileres (OP): por rango de fechas de emisión, por número de OP, por nro de pedido de fondos, por número asignado en SIGAF, por subresponsable (se refiere a la Delegación de Administración responsable) y por propietario. Además, permite ver la OP, modificar datos de la misma, modificar su número, imprimir el resultado de la consulta, reimprimir una OP en particular, y eliminar.

Subresponsable	Referencia	Fecha	Tipo de Pago	PF	Expediente	Período	Total	Ejercicio	Comprobante	Nº	Año	OPERADOR
----------------	------------	-------	--------------	----	------------	---------	-------	-----------	-------------	----	-----	----------

Figura 38 - Consulta de Alquileres - Aplicación de Contaduría

En el caso de pedido de fondos para pago a proveedores: por rango de fecha de emisión, por número de pedido de fondos, por PRD (Parte de Recepción Definitiva emitido por SIGAF), por número de comprobante o factura, por número asignado por SIGAF, por proveedor, por expediente del PRD, por expediente del pedido de fondos.

Emisión	Proveedor	Nro PRD	Año PRD	Responsable	Expte.	Importe	Multa	Comprobante	Comp.SIGAF	PF
---------	-----------	---------	---------	-------------	--------	---------	-------	-------------	------------	----

Figura 39 - Consulta de pagos - Aplicación de Contaduría

5.2.4- Tesorería

Para el área de tesorería, la información se maneja con el SIGAF, pero se generaron aplicaciones para controlar el detalle de los movimientos en el SIGAF.

Pago de acuerdos: En esta aplicación la base de datos de tesorería se conecta con la base de datos del área de Sueldos para liquidar y posteriormente pagar acuerdos al personal.

Subsidios: En esta aplicación la base de Tesorería se conecta con la base de datos de la Secretaría de Personal, para liquidar y posteriormente pagas los subsidios al personal

Ticketera: En esta aplicación de Tesorería se registra la información que cada una de las Delegaciones de Administración envía con respecto a los Jurados de Juicios por Jurado. La procesa y genera el archivo para remitir al Banco Provincia.

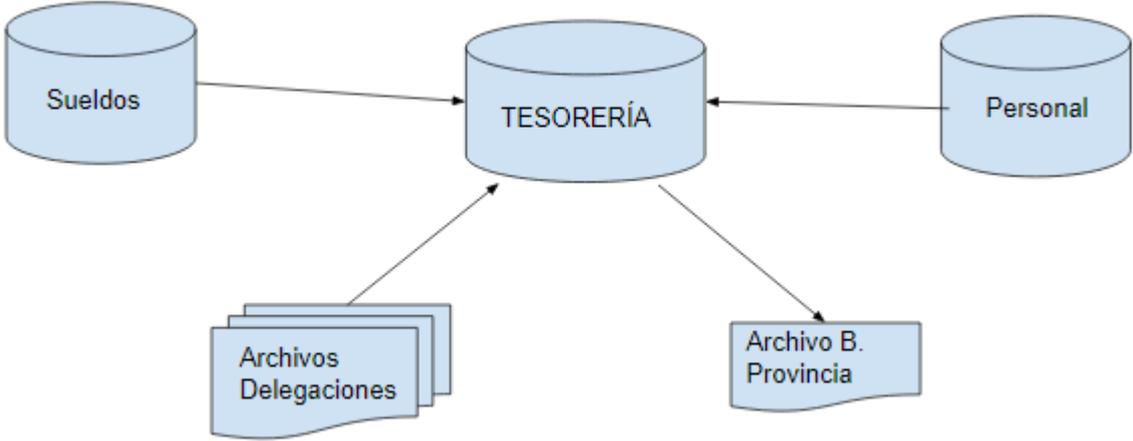


Figura 40 - Tesorería - Aplicaciones

CAPÍTULO 6 - CONCLUSIONES Y TRABAJOS FUTUROS

6.1- Conclusiones

El trabajo realizado fue una experiencia difícil, con muchos obstáculos, mucho esfuerzo, pocos recursos, pero con resultados sumamente positivos y gratificantes.

Si bien, en términos de tecnología, el cambio sigue estando lejos del avance informático de la actualidad pero, para el trabajo diario de la Secretaría de Administración resulta muy beneficioso. Desde la interfaz con el usuario hasta la optimización de los procesos para el manejo de datos.

Para evaluar el nivel de satisfacción de los usuarios con respecto al uso de las nuevas aplicaciones, se realizó una encuesta anónima de 10 preguntas. Con los resultados obtenidos se puede afirmar que la mayoría de los usuarios están conformes con el cambio. El Anexo I muestra gráficamente las respuestas y detalla las opiniones de los usuarios.

La encuesta, sobre un universo 46 usuarios de los distintos sistemas, arrojó los siguientes resultados:

Pregunta	Cantidad de Usuarios por aplicación				
	Inmuebles	Contrataciones	Contaduría	Registro único de Inmuebles	Registro Patrimonial
1. Indique de qué aplicaciones es usuario (puede indicar más de una)	10	18	20	6	4
	Si	No	Ns/Nc		
2. Considera que las aplicaciones actuales agilizaron su tarea (con respecto a los sistemas anteriores)?	95.70%	2.20%	2.10%		
3. Considera que las aplicaciones actuales son más fáciles de usar que los sistemas anteriores?	93.50%	2.20%	4.30%		
4. Considera que de las nuevas aplicaciones puede obtener más información que las anteriores?	87.00%	4.30%	8.70%		
5. Considera que las nuevas aplicaciones al estar interconectadas reducen la carga manual de datos?	91.30%	2.20%	6.50%		
6. Considera que las nuevas aplicaciones utilizan una interface más amigable con el usuario?	87.00%	2.10%	10.90%		
7. Considera que las nuevas aplicaciones tienen más funcionalidades que las anteriores?	89.10%	0.00%	10.90%		
8. Considera que las nuevas aplicaciones realmente mejoraron las tareas diarias del área?	97.80%	2.20%	0.00%		
9. Considera que no fue necesaria la implementación de las nuevas aplicaciones?	23.90%	67.40%	8.70%		
10. Qué opina de las nuevas aplicaciones?	Ver respuestas en el Anexo I				

Además se obtuvieron resultados agrupando los usuarios por aplicación y por respuesta

Respuestas afirmativas

Pregunta	Inmuebles	Contrataciones	Contaduría	Registro único de Inmuebles	Registro Patrimonial
	SI	SI	SI	SI	SI
2. Considera que las aplicaciones actuales agilizaron su tarea (con respecto a los sistemas anteriores)?	90.00%	88.89%	100.00%	100.00%	100.00%
3. Considera que las aplicaciones actuales son más fáciles de usar que los sistemas anteriores?	90.00%	88.89%	95.00%	100.00%	100.00%
4. Considera que de las nuevas aplicaciones puede obtener más información que las anteriores?	100.00%	88.89%	90.00%	66.67%	100.00%
5. Considera que las nuevas aplicaciones al estar interconectadas reducen la carga manual de datos?	100.00%	94.44%	95.00%	83.33%	75.00%
6. Considera que las nuevas aplicaciones utilizan una interface más amigable con el usuario?	100.00%	83.33%	95.00%	83.33%	75.00%
7. Considera que las nuevas aplicaciones tienen más funcionalidades que las anteriores?	100.00%	94.44%	80.00%	100.00%	100.00%
8. Considera que las nuevas aplicaciones realmente mejoraron las tareas diarias del área?	100.00%	94.44%	100.00%	100.00%	100.00%
9. Considera que no fue necesaria la implementación de las nuevas aplicaciones?	30.00%	22.22%	40.00%	33.33%	25.00%

Respuestas negativas

Pregunta	Inmuebles	Contrataciones	Contaduría	Registro único de Inmuebles	Registro Patrimonial
	No	No	No	No	No
2. Considera que las aplicaciones actuales agilizaron su tarea (con respecto a los sistemas anteriores)?	0.00%	5.56%	0.00%	0.00%	0.00%
3. Considera que las aplicaciones actuales son más fáciles de usar que los sistemas anteriores?	10.00%	5.56%	0.00%	0.00%	0.00%
4. Considera que de las nuevas aplicaciones puede obtener más información que las anteriores?	0.00%	5.56%	5.00%	0.00%	0.00%
5. Considera que las nuevas aplicaciones al estar interconectadas reducen la carga manual de datos?	0.00%	5.56%	0.00%	0.00%	0.00%
6. Considera que las nuevas aplicaciones utilizan una interface más amigable con el usuario?	0.00%	0.00%	5.00%	0.00%	0.00%
7. Considera que las nuevas aplicaciones tienen más funcionalidades que las anteriores?	0.00%	0.00%	0.00%	0.00%	0.00%
8. Considera que las nuevas aplicaciones realmente mejoraron las tareas diarias del área?	0.00%	5.56%	0.00%	0.00%	0.00%
9. Considera que no fue necesaria la implementación de las nuevas aplicaciones?	60.00%	66.67%	60.00%	33.33%	50.00%

Respuestas Ns/Nc

Pregunta	Inmuebles Ns/Nc	Contrataciones Ns/Nc	Contaduría Ns/Nc	Registro único de Inmuebles Ns/Nc	Registro Patrimonial Ns/Nc
2. Considera que las aplicaciones actuales agilizaron su tarea (con respecto a los sistemas anteriores)?	10.00%	5.56%	0.00%	0.00%	0.00%
3. Considera que las aplicaciones actuales son más fáciles de usar que los sistemas anteriores?	0.00%	5.56%	5.00%	0.00%	0.00%
4. Considera que de las nuevas aplicaciones puede obtener más información que las anteriores?	0.00%	5.56%	5.00%	33.33%	25.00%
5. Considera que las nuevas aplicaciones al estar interconectadas reducen la carga manual de datos?	0.00%	0.00%	5.00%	16.67%	25.00%
6. Considera que las nuevas aplicaciones utilizan una interface más amigable con el usuario?	0.00%	16.67%	0.00%	16.67%	25.00%
7. Considera que las nuevas aplicaciones tienen más funcionalidades que las anteriores?	0.00%	5.56%	20.00%	0.00%	0.00%
8. Considera que las nuevas aplicaciones realmente mejoraron las tareas diarias del área?	0.00%	0.00%	0.00%	0.00%	0.00%
9. Considera que no fue necesaria la implementación de las nuevas aplicaciones?	10.00%	11.11%	0.00%	33.33%	25.00%

6.2- Trabajos futuros

Estas nuevas aplicaciones están en constantes modificaciones, debido a que agregar nuevas funcionalidades a las mismas, resulta simple, en la mayoría de los casos. Además siempre surgen proyectos con otras áreas o dependencias.

Un proyecto a implementar a la brevedad consiste en que los sistemas almacenen los documentos en la nube y asimismo los recuperen. Actualmente los graban en las bases de datos locales, esto implica que el tamaño de las bases crezca enormemente e incida en la performance.

Otro proyecto es desarrollar las interfaces con un sistema web recientemente implementado por la Subsecretaría de Tecnología Informática, SGE (Sistema de Gestión de Expedientes) que los usuarios deben utilizar. De este modo, al intercambiar ambos sistemas las novedades, se evita la doble carga por parte de los usuarios.

BIBLIOGRAFÍA

<https://www.powerdata.es/migracion-de-datos>

Migración de datos: definición, desafíos y mejores prácticas para afrontarla - 11/2022)
-Migración de Bases de Datos

Serra, 2018 - Metodología para construcción de procesos de migración de datos en contexto de sistemas en desarrollo.

Rodríguez y Rosso, 2013. ProMiDa Proceso de Migración de Datos.

http://www07.ibm.com/th/events/ibmuni2009/downloads/Other/GTS_2009_Thailand_University_gts.pdf.

Orallo, 2002 . La Disciplina de los Sistemas de Bases de Datos. Historia, Situación Actual y Perspectivas. *Dep. de Sistemas Informáticos y Computación, Universidad Politécnica de Valencia*.

Cerón, A. B., Aviña, V. G. F., & Covarrubias, R. A. N. Actualización de Bases de Datos DBF a MySQL. *La computadora, herramienta indispensable en diversas áreas de conocimiento*, 117.

Ullman, J.D.; Widom, J.(1997) "A First Course in Database Systems" Prentice-Hall International, 1997

Gray, J.N. ,1981 - "The Transaction Concept: Virtues and Limitations" in Very Large DataBases, 1981.

Suárez, E. M. , 2008. ¿ Que es una base de datos relacional?. Universidad de Murcia, Murcia, España.

<http://victorgh84.blogspot.com/2015/09/base-de-datos-jerarquica.html>

Computación: Base de Datos Jerárquica - 02/2023

<https://ayudaleyprotecciondatos.es/bases-de-datos/modelos/> El Modelo base de datos: Definición y tipos - 03/2023

<https://basededatosutim.wordpress.com/2014/03/21/objeto-relacionales/>

Objeto-relacionales | Base De Datos UTIM - 03/2023

<https://www.redalyc.org/journal/2570/257051186010/html/> Procesamiento en paralelo y distribuido en dos SGBDS: un caso de estudio -03/2023

Alvez, C. E., & Vecchiatti, A. R. , 2006. SQL: 1999 y SQL/MM para el Modelado y Diseño de Bases de Datos Multimedia. In *VIII Workshop de Investigadores en Ciencias de la Computación*.

Antaño, A. C. M., Castro, J. M. M., & Valencia, R. E. C. , 2014. Migración de bases de datos sql a nosql. *Revista Tlamati, Especial*, 3, 144-148.

<https://www.timetoast.com/timelines/historia-y-evolucion-de-las-bd> Historia y evolución de las BD timeline - 03/2023

<https://www.timetoast.com/timelines/evolucion-de-los-sistemas-de-bases-de-datos-cef83116-5c5a-4f55-810c-968162eb0552> Evolución de los sistemas de bases de datos timeline | Timetoast - 03/2023

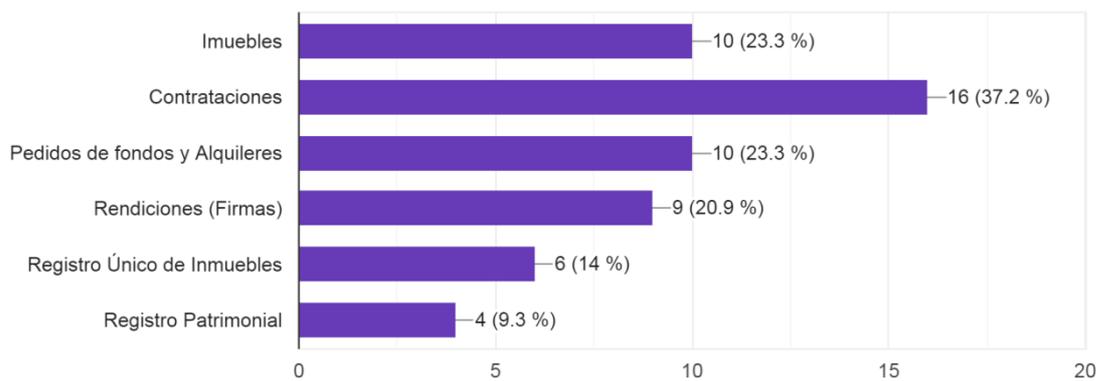
<https://www.timetoast.com/timelines/evolucion-de-las-bases-de-datos-ff60b26f-f1c2-46e3-9dba-3433b60c6bc5> Evolucion de las Bases de Datos timeline - 03/2023

ANEXO 1

Resultados de la encuesta realizada

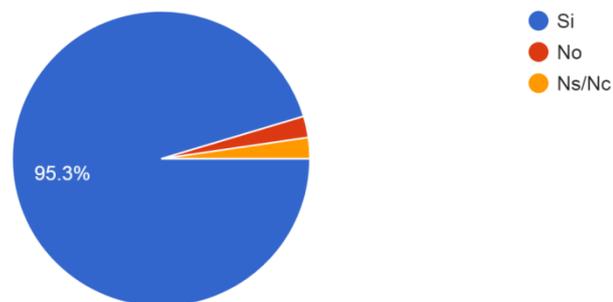
1. Indique de qué aplicaciones es usuario (puede indicar más de una)

43 respuestas



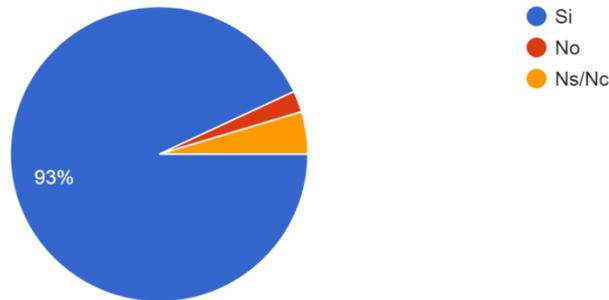
2. Considera que las aplicaciones actuales agilizaron su tarea (con respecto a los sistemas anteriores)?

43 respuestas



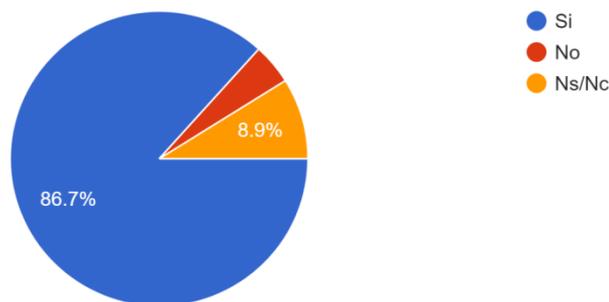
3. Considera que las aplicaciones actuales son más fáciles de usar que los sistemas anteriores?

43 respuestas



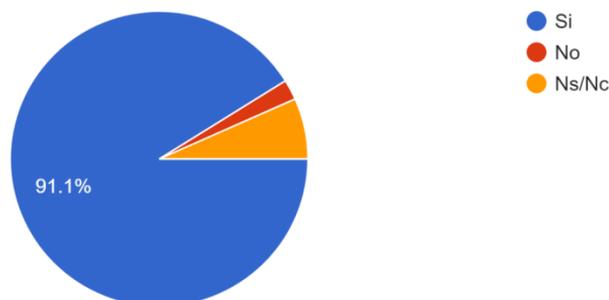
4. Considera que de las nuevas aplicaciones puede obtener más información que las anteriores?

45 respuestas



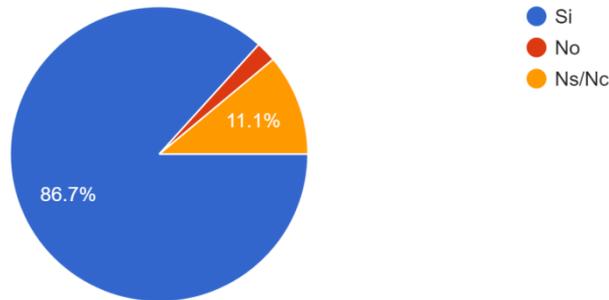
5. Considera que las nuevas aplicaciones al estar interconectadas reducen la carga manual de datos?

45 respuestas



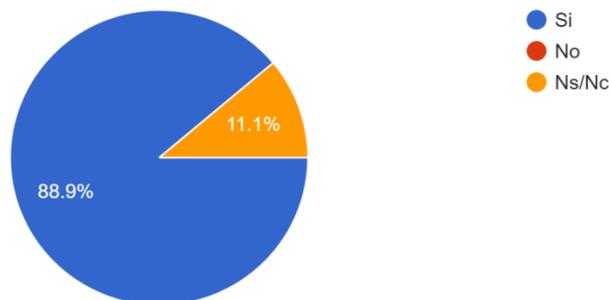
6. Considera que las nuevas aplicaciones utilizan una interface más amigable con el usuario?

45 respuestas



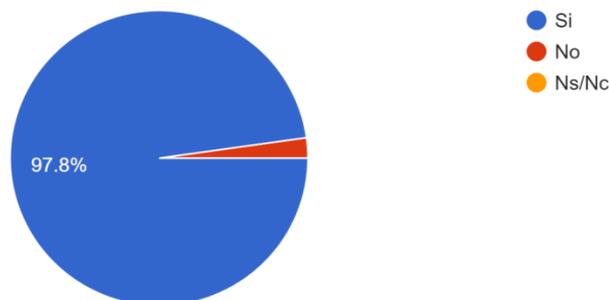
7. Considera que las nuevas aplicaciones tienen más funcionalidades que las anteriores?

45 respuestas



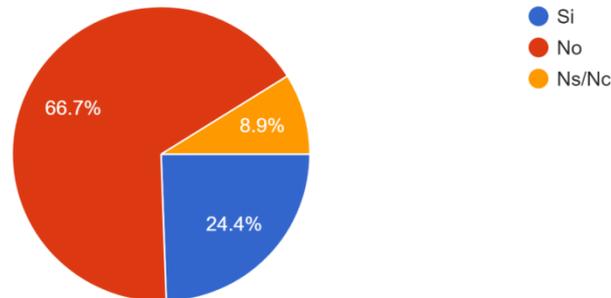
8. Considera que las nuevas aplicaciones realmente mejoraron las tareas diarias del área?

45 respuestas



9. Considera que no fue necesaria la implementación de las nuevas aplicaciones?

45 respuestas



10. Qué opina de las nuevas aplicaciones? (45 respuestas)

- Son de fácil operación, con tiempos de respuesta mucho más ágiles e información más clara y completa.
- Fáciles de usar
- Considero que son más dinámicas para mi tarea diaria
- Ahorro de tiempo
- Que agilizaron el trabajo diario
- fácil de usar
- Son geniales!
- son ágiles y sencillas para usar
- Que están más adaptadas a las necesidades del área de trabajo, logrando más eficiencia y eficacia al momento de utilizarlas y han tenido en cuenta todos los requerimientos para poder realizar la tarea diaria con menos errores y de forma más amigable.
- Antes que nada en la respuesta 9 considero que es necesaria la implementación del nuevo sistema que agiliza. En el caso de pedidos de fondos, sugiero que tenga visualizado que meses devenga la orden de compra o orden de provisión. Y si es entrega de bienes de consumo o bienes de uso ver el importe que resta a devengar en los contratos. Con esa observación, indico que ha mejorado bastante el sistema, siendo más agil y seguro. La seguridad se da en que se vincula los proveedores con su CUIT y luego que no se pueden agregar las órdenes de pago a pedidos de fondos cerrados.
- facilitaron el trabajo
- Muy Buenas, facilitan mucho la actividad en el área de contrataciones.
- Han facilitado las tareas y reducido el tiempo de resolución de las mismas.
- beneficio en la búsqueda y en armado de tablas
- facilitan la búsqueda de info mediante los filtros
- Resultaron muy beneficiosas dado que el atraso era muy marcado, las anteriores aplicaciones eran sobre DOS, además el equipo se adaptó fácilmente y trabajan conformes con su herramienta.

-
- Las aplicaciones de las cuales utilizo en mi labor diaria, brindan una mejor interfaz que las anteriores y por ello considero que son más prácticas para el usuario, ya sea alguien que conoce del uso de pc como alguien que tiene desconocimiento del mismo.
 - Las apps simplifican el trabajo cotidiano y reducen la utilización de papel.
 - Buena, con posibilidades de mejorar
 - que esta buenas y agilizan la tarea
 - Es una aplicación muy útil. Agiliza el trabajo diario.
 - Que a nivel información para toma de decisiones son más indicadas
 - Son simplificadoras de la tarea cotidiana, muy útiles y de fácil utilización.
 - Tanto el sistema de inmuebles como el Registro único de inmuebles gracias al registro constituyen una gran base de datos que facilitan y agilizan la labor diaria.
 - Permiten visualizar todo lo tramitado en el expediente
 - Son más prácticas y y por ende agilizan las tareas diarias respecto de su uso
 - facilitan el desarrollo del trabajo diario, agilizándolo
 - Muy buena
 - Mi opinión respecto de las nuevas aplicaciones es que no sirven a la hora de listar datos , o responder rápidamente a una consulta y que la aplicación anterior si y por eso es que sigue vigente en nuestro control diario
 - Muy eficientes y fáciles de utilizar para el trabajo
 - Son fáciles de usar y simplifican la tarea
 - Son buenas.
 - Son más rápidas para las consultas, y en la realización de las Órdenes, eso hace ahorro de tiempo para realizar tareas y estar más al día en el trabajo.
 - Agiliza el trabajo y redujo el uso de papel
 - En relación a la aplicación Contrataciones que utilizo diariamente desde el sector Ordenes de compra y Pases al pago -Área compras y Contrataciones -: considero a esta herramienta indispensable (tanto para la gestión como para el control de la misma).
 - Muy ágil y fácil de usar
 - Ágil. Práctica. Y accesible
 - Ágil y de fácil interpretación
 - Que muchas veces a pesar de buscar formas más modernas o prácticas terminan creando sistemas horribles que atrasan años a la forma tradicional
 - Muy buenas
 - Fue un buen cambio
 - Muchas en vez de simplificar, complican la tarea, habría que trabajar más en la simpleza y algo concreto
 - Excelente
 - Las nuevas aplicaciones agilizan el trabajo y visualizan los datos mejor que las anteriores.
 - Ayudan a mejorar y optimizar la gestión y tramitación diaria.
-