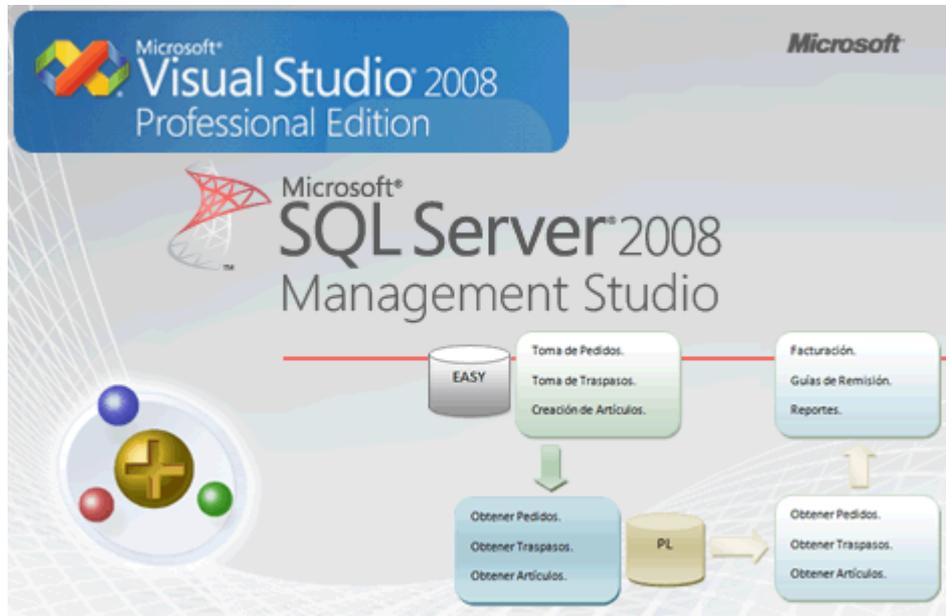


# CAPÍTULO IV

---



## RESUMEN

En este capítulo daremos a conocer como es el funcionamiento de las diferentes bases de datos que la aplicación tiene en uso, esto es el caso de las bases de datos EASY y PL, estas dos bases interactúan de forma dependiente en diferentes tablas, tales como son los datos de los clientes, artículos, códigos de barras entre otras. El estudio que se realiza a estas bases de datos es principalmente por el problema de la concurrencia que existe al momento de utilizar las diferentes aplicaciones que usan las mismas tablas.

Para un correcto funcionamiento de las aplicaciones y la base de datos se recomienda instalar y configurar correctamente los servidores de aplicaciones y base de datos. Estas configuraciones se encuentran detalladas en los anexos e instalaciones, como también se da a conocer un proceso de cómo optimizar la base de datos, este proceso genera y crea índices si las diferentes consultas

las necesitan. Estos índices deben de ser administrados y revisados por lo menos cada tres meses para que los índices no sean un problema.

Mas adelante se podrá observar un modelo de los procesos que la aplicación realizará al transcurso del desarrollo de la tesis, como también las diferentes tablas afectadas entre las dos bases de datos.

Se debe tomar en cuenta que la base de datos principal de la empresa donde se desarrolló la aplicación es la de ERP, esto fue analizado y procesado de una forma que los procesos que se corren no afecten al desempeño de los procesos de la empresa.

## 4. Análisis del las Bases de Datos de EASY y PL

A continuación estudiaremos las bases de datos que usaremos en el proceso de la implementación y ejecución de la tecnología WPF en estudio.

### 4.1. Análisis del uso de varias BDD's en servidores distintos.

Para comenzar a realizar el análisis de la estructura y el funcionamiento de las bases de datos que se usaran para la ejecución del aplicativo daremos a conocer el funcionamiento que tienen las diferentes bases de datos, claro que cada base de datos corren en distintos servidores.

La base de Datos de EASY la misma que es el ERP de la empresa y donde se almacenaran informaciones como la parametrización de los artículos, kardex de los artículos, toma de pedidos para clientes particulares, toma de traspasos para farmacias propias, y el proceso comercial y financiero que no necesitamos para el estudio que realizaremos.

La estructura a seguir en este estudio es la que indicaremos a continuación, allí se detalla el proceso a seguir como el funcionamiento esquemático del aplicativo en ejecución.

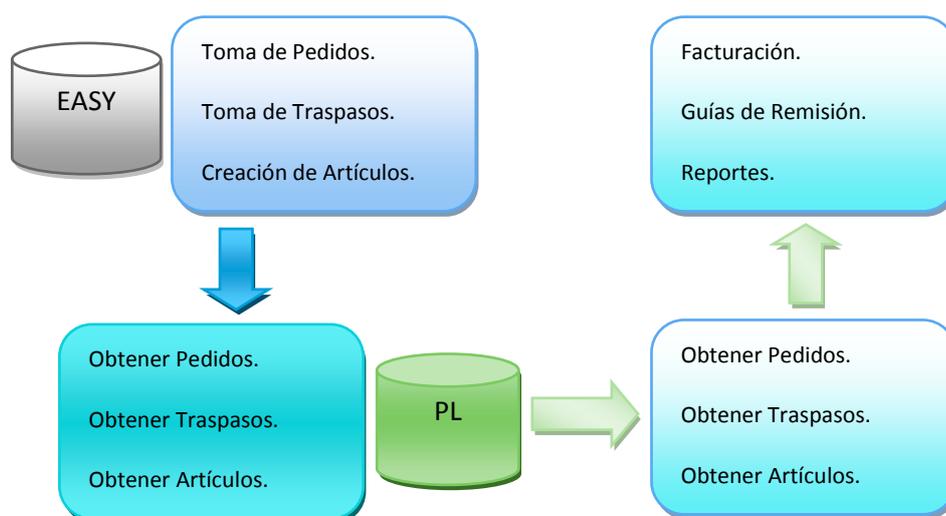


Figura 4.1: Análisis de Uso de Varios Base de Datos

Como se indica en la Figura 4.1, se representa el flujo de la información que se transmitirá desde las distintas bases de datos hacia el proceso de control de pedidos y trasposos en bodega central de la empresa Farmaenlace Cía. Ltda.

En la base de Datos *EASYGESTIONEMPRESARIAL* se almacenaran los datos de los pedido de los distintos clientes particulares que tiene la empresa, esta toma de pedidos lo realizan directamente a la empresa mediante un aplicación la misma que no es parte de nuestro estudio, estos pedidos son realizados y almacenados en la BDD, de igual forma son los trasposos de mercadería que se realizan automáticamente mediante una fórmula estadística que coge las ventas realizadas por farmacia y según esa información se arma el trasposo y se almacenan en la BDD.

Una vez que esta información se encuentra almacenada, un proceso de bodega los toma y los transforma en información apta para la nueva base de datos de TRANSPORTE.



The screenshot shows a software application window titled "CARGA DE PEDIDOS". At the top, there is a header with the title "CARGA DE PEDIDOS" and a subtitle "CARGA DE PEDIDOS". Below the header, there are two input fields: "Usuario: SISTEMAS" and "Fecha Proceso: 2010/06/01". The main area of the window contains a table with the following data:

MOV.	RUC/CEDULA	NOMBRE RAZON	FECHA	OBSERVACION	HORA
6564	1801910769001	FARMACIA LA FE	2010/06/01		17:05:57
6565	0501312581001	FARMACIA CHIMBACALLE	2010/06/01		17:05:57
6566	1711411825001	FARMACIA ALVAREZ	2010/06/01		17:05:57
6567	1711929362001	FARMADESCUENTOS CAROLITO	2010/06/01		17:05:57
6568	1791933915001	MEDISALUD	2010/06/01		17:05:57

At the bottom of the window, there is a toolbar with four icons: a printer, a gear, a wrench, and a person running.

Figura 4.2: Carga de Pedidos

Una vez almacenada en esta base de datos se procede a imprimir las hojas de despacho (Ver Figura 4.2), que serán entregados en Picking para que este los distribuya a las diferentes personas que sacan los pedidos.

Los pedidos que se asignan a dichas personas son registradas en el sistema mediante el modulo de Picking (Ver Figura 4.3), para que un mismo pedido no sea despachado varias veces, desde este proceso las personas comienzan a sacar la mercadería de las perchas guiandose en la hoja del pedido, una vez que terminan este proceso las personas dejan la mercadería substraída de la bodega a las personas de certificado.

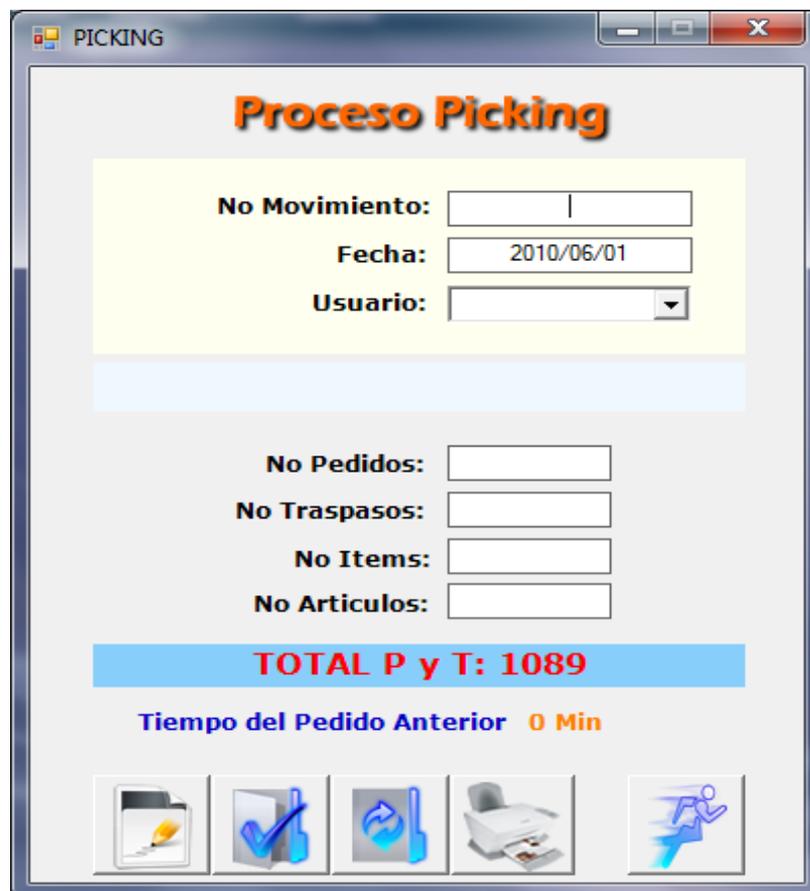


Figura 4.3: Proceso de Picking

DESPACHOS DE ARTICULOS

## Despachos a Realizar

No Movimiento:

**PEDIDOS** | **TRASPASOS**

Folio Referencia:

Usuario Digita:

Usuario Confirma:

Cliente:

Dir. Cliente:

Fecha Elabora:

Fecha Confirma:

Fecha Despacho: 2010/06/01 - 19:34:52

No Cajas:

No Fundas:

No Pacas:

Usuario: SISTEMAS | V. 1.5

**Figura 4.4:** Certificación de Mercadería

Luego de este proceso se procede a certificar los pedidos y trasposos mediante el modulo de despacho certificado (Ver Figura 4.4). Aquí se pasan los artículos como si fuese un punto de facturación de punto de venta, de tal forma que se controla la cantidad que se pidió con la cantidad que se envió al cliente, si las cantidades son erróneas el sistema indica el error tal como "El artículo no pertenece al pedido", "La cantidad a enviar es mayor a la cantidad pedida", "El artículo ya no pertenece al pedido".

Dando la seguridad y eficiencia al momento de enviar la mercadería a los clientes. Cuando el proceso a terminado se procede a facturar mediante el modulo de facturación (Ver Figura 4.5).

CÓDIGO	NOMBRE	UMPOS	ENTEROS	BONIFICA.	PVP	PVF	VTOTAL
--------	--------	-------	---------	-----------	-----	-----	--------

Subtotal: 0 % Desc: 0 Subtotal IVA 0%: 0 Subtotal IVA 12%: 0 IVA 12%: 0

Figura 4.5: Proceso de Facturación

Luego de este proceso de procede a entregar la mercadería a Transporte quienes registraran la guía de salida y la ruta para la entrega final de la mercadería.

#### 4.1.1. Como hacer que no cause conflicto la concurrencia a la BDD

Para el estudio de la concurrencia y conflicto de la información al momento de acceder a las diferentes bases de datos consideraremos un factor importante que es el servidor SQLSERVER, ya que desde la versión 2005 y 2008 los problemas de concurrencia los manejan automáticamente, solo se tendrá en cuenta los bloqueos que los diferentes procesos de acceso a la BDD causen ya que se está hablando de 400 personas accediendo al mismo tiempo a la BDD y unas 150 a las mismas tablas lo cual causaran bloqueos entre procesos si las consultas hacia la BDD son pesadas o de bastante tiempo de ejecución. Pero al momento de que esto sucede el sistema deja de acceder a la BDD reintentando varias veces en procesos distintos para evitar posibles colisiones.

Uno de los problemas primordiales de la saturación de la BDD son las respuestas que el servidor de a la ejecución de las consultas de los sistemas y

las peticiones que los usuarios realicen al servidor, para lo cual tendremos que realizar indexar las tablas de mayor concurrencia y así dar una respuesta más rápida y evitar los bloqueos entre procesos.

#### **4.1.2. Análisis de las conexiones COM+, OLE DB y ADO para el acceso a las BDD's EASY y PL**

A continuación daremos a conocer las conexiones de tipo COM+, OLE DB y ADO las mismas que son las más usadas y eficientes al momento de obtener la información de las diferentes BDD's que esta manipulen.

**Conexiones COM+.-** A continuación se describirá el funcionamiento primordial de una conexión COM+.

**El Modelo de Objetos Componentes.-** La clave de COM para la interoperabilidad de módulos consiste en una descripción exacta de los contratos entre módulos mediante la especificación binaria de punteros de interfaces.

**Bibliotecas de tipos.-** Los servidores COM utilizan recursos, o en ocasiones ficheros independientes, para la descripción de los servicios que ofrecen. Estas descripciones suelen crearse en un lenguaje llamado IDL(Interactive Data Language). Casi cualquier lenguaje de programación moderno puede crear las declaraciones necesarias para acceder una clase COM arbitraria a partir de la biblioteca de tipos correspondiente.

**Creación de servidores.-** COM permite alojar clases dentro de módulos de carga dinámica DLL(dinamic link library), o incluso dentro de ejecutables y aplicaciones de servicio.

**Automatización OLE.-** El tipo de interfaz IDispatch describe un mecanismo sencillo de ejecución de macros que se hizo muy popular por su aplicación en lenguajes interpretados.

**Modelos de concurrencia e implicaciones para el marshaling.-** Se conoce como marshaling la técnica que permite la transparencia, de cara al cliente, respecto a la ubicación física de un componente servidor. El conocimiento de esta técnica es sumamente importante para comprender muchas de las características del entorno COM+ y, naturalmente, para la administración de conexiones COM distribuidas.

**El entorno de ejecución COM+.-** COM+ implementa una serie de ideas inspiradas en la Aspect Oriented Programming, que permiten disminuir dramáticamente el número de líneas de código necesarias para el "montaje" de componentes. Además, el entorno COM+ aporta una serie de ventaja a los componentes y aplicaciones que hacen uso de él, como la posibilidad de ejecutarse como servicios de Windows.

**Administradores de recursos.-** COM+ administra automáticamente determinados recursos, como las conexiones a una base de datos por medio de ADO. También ofrece servicios, como el Shared Property Manager, para que los componentes no tengan que utilizar memoria global cuando tengan que comunicarse entre ellos.

**Object pooling, y sus implicaciones.-** Una de las características más populares de COM+ es la compartición de unas pocas instancias de objetos entre un número considerablemente mayor de clientes. Aquí se explican los requerimientos para el object pooling, y se presenta la técnica de application pooling.

**Objetos y transacciones.-** Otra característica importante de COM+ es el uso automático de transacciones. A más de existir transacciones a nivel de conexión también existen transacciones a nivel de componente dando mayor seguridad a la hora de guardar la información y reducir así bloqueos del servidor de BDD.

**Eventos en COM+ (loosely coupled events).**- Los eventos clásicos de COM, basados en la técnica de los puntos de conexión, implican que el emisor y sus posibles receptores son conscientes mutuamente de su existencia. El modelo de eventos de COM+ (eventos con acoplamiento débil), permite la programación de publicadores y suscriptores por separado. Además, la mayor parte del código necesario lo aporta el propio entorno de COM+. El modelo da soporte para filtros de mensajes, para la activación del componente por demanda y puede ser complementada con el uso de componentes en cola.

En conclusión diremos que la conexión mediante COM se lo hace con accesos directos al servidor usando el canal de TCP/IP esta información viaja encriptado para brindar seguridad la comunicación dejando que el servidor de componentes administre los procesos de conexión a la BDD.

**OLE DB.**- Denominado también *Object Linking and Embedding for Databases* es una tecnología desarrollada por Microsoft usada para tener acceso a diferentes fuentes de información, o bases de datos, de manera uniforme. También se puede decir que es un conjunto de interfaces COM que permiten el acceso a multitud de orígenes de datos. OLEDB está diseñado para proporcionar una funcionalidad óptima en una gran variedad de aplicaciones, pero no cumple el requisito de simplicidad; para esto surgió ADO (ActiveX Data Objects), que incluye un conjunto de clases que facilitan el acceso y manipulación de las interfaces OLEDB en nuestras aplicaciones.

**ADO.**- Denominado *ActiveX Data Objects* que es la evolución de OLE DB incluyendo más beneficios a la hora de acceder y manipular las interfaces de comunicación que nos brinda la tecnología.

ADO.NET proporciona acceso coherente a orígenes de datos como Microsoft SQL Server y XML, así como a orígenes de datos expuestos mediante OLE DB y ODBC. Las aplicaciones para usuarios que comparten datos pueden utilizar ADO.NET para conectar a estos orígenes de datos y recuperar, manipular y actualizar los datos contenidos.

### **4.1.3. Técnicas de mejoramiento del acceso hacia la BDD.**

Para que las aplicaciones puedan ejecutarse de una forma eficiente y eficaz se considera usar procediendo ya que estos se ejecutan directo en el servidor sin pasar por el canal información que no se desea se envíe, como es un *select* de un sin número de caracteres, lo aconsejable sería usar vistas para obtener esta información y retornar los datos necesarios para presentar al usuario.

A continuación daremos a conocer algunos procesos que nos ayudara a que las aplicaciones sean más optimas.

Usar Procedimientos cuando se trate de insertar, modificar, eliminar registros de una tabla o tablas relacionadas, con esto ganamos que la transacción se manipule en el servidor y no en un cliente que está a la espera de respuesta satisfactoria para realizar el commit a la transacción en curso, si no que el cliente reciba el OK de la operación caso contrario el error del servidor, operación que ya se puede obtener desde SQL SERVER 2005.

Si la transacción que se maneja se la hace desde el cliente es de suma importancia que la información viaje encriptado para evitar posibles filtraciones y manipulaciones de información a la hora de procesar la información en la BDD.

Si se trata de manejar consultas es importante usar vistas (VIEW) ya que estas igual que los procedimientos se ejecutan directo en el servidor, esta información resultante es la que viajara hacia el cliente para mostrar los resultados.

Usar las cadenas de conexiones encriptados y colocadas si es posible solo en el servidor de tal forma que las conexiones sean más seguras.