**Data Warehouse**

El sistema Data Warehouse está formado por las siguientes partes:

* Capa de Conexión y Acceso a Datos.
* Capa de Servicios.
* Capa Controladora.
* Capa de Vistas.

A continuación se describe el detalle de cada Capa en el sistema

**Capa de Conexión y Acceso a Datos.**

En esta capa se realiza la conexión a la base de datos. Por ahora se prevé que los datos se encontraran en un servidor SQL de Microsoft, se están haciendo pruebas ya que el conector que ofrece Microsoft para conectarse a sus bases de datos tiene una configuración diferente a lo que comúnmente se espera de un conector para Java Database Conectivity.

En esta capa encontramos los Objetos de Acceso a Datos conocidos como DAO, clases especiales en las que se realizan consultas a la base de datos en el lenguaje SQL standard, se opto en este proyecto utilizar los DAO frente al API Java de Persistencia ya que es un sistema Monousuario, el sistema Data Warehouse es para llevar un monitoreo de las bases de datos de otros sistemas, no se van a ingresar datos en este sistema, solo se consultan.

**Capa de Servicios**

En la capa de los DAO solo obtendremos los resultados de las consultas, sin embargo no obtendremos detalles avanzados acerca de ellos, un ejemplo claro es la obtención del saldo de la empresa X en el banco A, quizá sea el único banco que maneje la empresa, pero en realidad muchas empresas manejan más de un banco, así que para conocer cuánto dinero en total se está manejando hay que saber cuántos bancos tiene la empresa X, cuales bancos son y que saldos tienen, esas operaciones simples las obtenemos usando los DAO, sin embargo los DAO solo hacen operaciones pequeñas, es aquí donde entran los Objetos Service.

Con un Objeto Service podemos conocer los saldos de una empresa utilizando la información de uno o varios DAO y además hacer las operaciones necesarias ya que hay que sumar los saldos de acuerdo a su tipo, con esto obtendremos los resultados.

**Capa Controladora**

Cuando estamos en un ambiente grafico muchas operaciones deben agruparse para generar el resultado final, es decir se necesitan invocar uno o más servicios dependiendo de las necesidades del cliente, además se necesita responder a sus solicitudes las cuales las realiza a través de la capa Vista.

La capa controladora sirve como un proxy de comunicación entre las peticiones del cliente y las operaciones del sistema, entre la capa Vista y la Capa de Servicios.

**Capa de Vistas**

Esta capa final muestra los resultados obtenidos en el sistema, utilizaremos los componentes gráficos del paquete swing de java para construir las interfaces graficas de usuario, Netbeans ya provee un diseñador gráfico, en el cual solo arrastramos y colocamos los componentes, así como asociar a los componentes manejadores de eventos para cuando el usuario interactúe con el sistema, esto sin embargo a pesar de ser lo más sencillo aquí entra la capa controladora, está pendiente realizar el diseño de esta capa.