Contenido

[Propósito. 2](#_Toc359488597)

[Alcance. 2](#_Toc359488598)

[Patrones Arquitectónicos 2](#_Toc359488599)

[Estructuración del sistema. 3](#_Toc359488600)

[Capas 4](#_Toc359488601)

[Vista arquitectónica del despliegue – Niveles de hardware 5](#_Toc359488602)

[Accesos de Base de Datos 6](#_Toc359488603)

[Data provider 6](#_Toc359488604)

[DataSets 6](#_Toc359488605)

[Consultas SQL 7](#_Toc359488606)

# Propósito.

Se pretende que la descripción de la arquitectura brinde una visión global y comprensible del diseño general del framework desarrollado, exponiendo la arquitectura implementada, la estructura del sistema, las capas, consultas y procedimientos almacenados, niveles de hardware y el mapeo de la base de datos.

# Arquitectura Cliente/Servidor

Luego de realizar el análisis del sistema, los subsistemas que lo componen y sus relaciones, se observa necesario realizar una separación de intereses entre las diferentes capas del sistema.

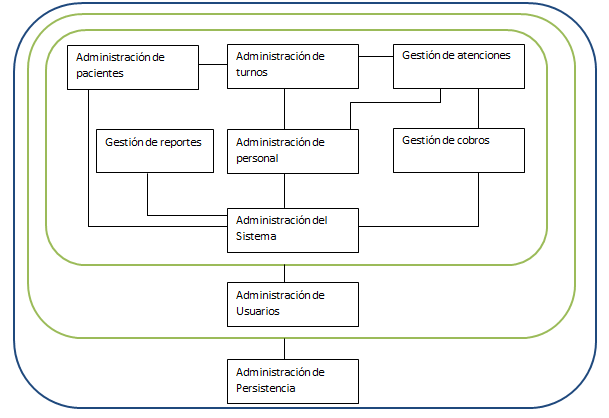
Los requerimientos no funcionales planteados en relación a tiempo de respuesta y desempeño del sistema llevan a minimizar en lo posible las llamadas necesarias entre capas para completar las peticiones.

Las capas a considerar son: Cliente / Lógica de negocio / Administración de Datos y Servicios Web. Esta última capa resuelve el requerimiento en relación al entorno web requerido para la consulta de días y horarios de atención por profesional y/o especialidad, e información general de la clínica.

# Estructuración del sistema.

El sistema se dividirá en los siguientes subsistemas:

* Administración de pacientes: contempla la administración de la información de los pacientes y sus respectivas historias clínicas.
* Administración de turnos: Incluye la administración de información de los turnos y la gestión de la información de las atenciones.
* Administración de personal: administración de los datos del personal y los profesionales, asignación de profesionales a consultorio y especialidades
* Gestión de cobros: Administración de la información de los cobros efectuados a cada atención.
* Gestión de atenciones: considera la gestión de datos sobre las atenciones médicas realizadas a los pacientes.
* Gestión de reportes: gestiona las tareas vinculadas a la elaboración y presentación de reportes y documentos.
* Administración de Sistemas: Contempla la administración de soporte.
* Administración de Usuarios: Contempla la administración de usuarios y sesiones.
* Administración de Persistencia: Gestiona las tareas vinculadas a la persistencia de los objetos en una base de datos relacional.



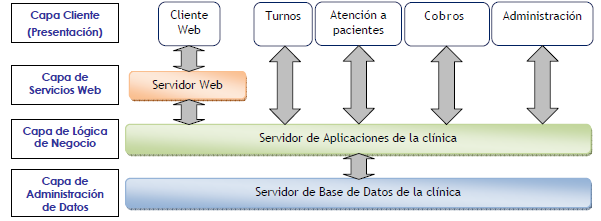
# Capas

Las capas a considerar son: Cliente / Lógica de negocio / Administración de Datos y Servicios Web. Esta última capa resuelve el requerimiento en relación al entorno web requerido para la consulta de días y horarios de atención por profesional y/o especialidad, e información general de la clínica.

Los puestos en la capa cliente responden a la distribución y organización de los principales requerimientos y funcionalidades del sistema. Los puestos clientes identificados son:

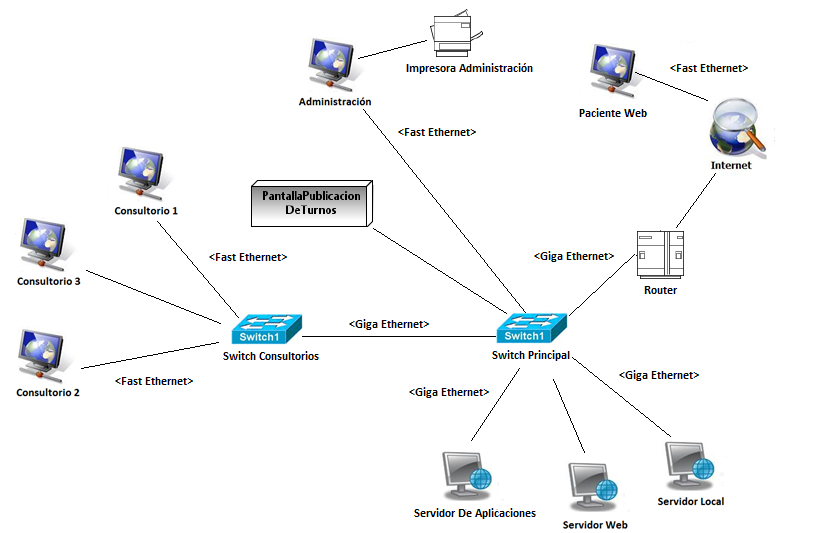
* Cliente Web. Puesto referido a la consulta a través del entorno Web sobre días y horarios de atención e información general de la clínica.
* Turnos. Puesto que comprende la administración y definición de horarios y turnos.
* Atención a pacientes. Puesto vinculado con la atención administrativa a pacientes: otorgamiento de turnos, recepción de pacientes y cobro, cancelaciones respectivas.
* Atención médica. Puesto cliente referido al registro de las atenciones médicas realizadas en los consultorios.
* Administración. Puesto vinculado con la gestión de datos de profesionales, obras sociales y administración general del sistema.

Debe tenerse especial cuidado con la capa de administración de datos para evitar que se convierta en un cuello de botella al resolver las transacciones.



# Niveles de hardware

A continuación se presenta la vista arquitectónica del despliegue, donde se visualizarán los niveles de hardware definidos para el sistema. En el diagrama de despliegue se observarán los nodos que conforman la topología de hardware sobre la que se ejecutará el sistema.



# Accesos de Base de Datos

ADO.NET es un conjunto de [componentes del software](http://es.wikipedia.org/wiki/Componente_de_software) que serán usados para acceder a datos y a servicios de datos. Es una parte de la biblioteca de clases base que están incluidas en el Microsoft [.NET](http://es.wikipedia.org/wiki/.NET) Framework. Se utilizará para acceder y modificar los datos almacenados en un [Sistema Gestor de Bases de Datos](http://es.wikipedia.org/wiki/Sistema_Gestor_de_Bases_de_Datos) Relacionales.

ADO.NET consiste en dos partes primarias:

### Data provider

Estas clases proporcionan el acceso a una fuente de datos, como [Microsoft SQL Server](http://es.wikipedia.org/wiki/Microsoft_SQL_Server).

Cada fuente de datos tiene su propio conjunto de objetos del proveedor, pero cada uno tiene un conjunto común de clases de utilidad:

* [Connection](http://es.wikipedia.org/w/index.php?title=Connection_(ADO.NET)&action=edit&redlink=1): utilizado para comunicarse con la fuente de datos.
* [Command](http://es.wikipedia.org/w/index.php?title=Command_(ADO.NET)&action=edit&redlink=1): se utiliza para realizar una lectura, actualización, o borrado de datos relacionales.
* [Parameter](http://es.wikipedia.org/w/index.php?title=Parameter_(ADO.NET)&action=edit&redlink=1): Describe un parámetro para un command. Un ejemplo común es un parámetro para ser usado en un [procedimiento almacenado](http://es.wikipedia.org/wiki/Procedimientos_almacenados).
* [DataAdapter](http://es.wikipedia.org/w/index.php?title=DataAdapter_(ADO.NET)&action=edit&redlink=1): utilizado para transferir data entre una fuente de datos y un objeto DataSet
* [DataReader](http://es.wikipedia.org/w/index.php?title=DataReader_(ADO.NET)&action=edit&redlink=1): Es una clase usada para procesar una lista grande de resultados.

### DataSets

Los objetos DataSets, es un grupo de clases que describen una simple [base de datos relacional](http://es.wikipedia.org/wiki/Base_de_datos_relacional) en memoria.

# Consultas SQL

Se utilizará procedimientos almacenados. Un procedimiento es un [programa](http://es.wikipedia.org/wiki/Programa_(computaci%C3%B3n)) (o [procedimiento](http://es.wikipedia.org/wiki/Subrutina)) el cual es almacenado físicamente en una [base de datos](http://es.wikipedia.org/wiki/Base_de_datos). La ventaja de un procedimiento almacenado es que al ser ejecutado, en respuesta a una petición de usuario, es ejecutado directamente en el motor de bases de datos, el cual usualmente corre en un servidor separado. Como tal, posee acceso directo a los datos que necesita manipular y sólo necesita enviar sus resultados de regreso al usuario, deshaciéndose de la sobrecarga resultante de comunicar grandes cantidades de datos salientes y entrantes.

Usos típicos para procedimientos almacenados incluyen la validación de datos siendo integrados a la estructura de base de datos o encapsular un proceso grande y complejo. El último ejemplo generalmente ejecutará más rápido como un procedimiento almacenado que de haber sido implementado como, por ejemplo, un programa corriendo en el sistema cliente y comunicándose con la base de datos mediante el envío de consultas [SQL](http://es.wikipedia.org/wiki/SQL) y recibiendo sus resultados.

# Mapeo de Base Datos