**Documento de Arquitectura de Software**

1. **Introducción**

* 1. **Propósito**

El propósito de este documento es generar el modelo de arquitectura del sistema de gestión de nodos para la empresa Medtronic. Se pretende capturar y transmitir las decisiones arquitecturales importantes que se han tomado en el desarrollo del proyecto.

* 1. **Alcance**

El alcance de este documento será enfocado en varias vistas: vista abstracta de alto nivel, vista lógica, vista de despliegue y vista de implementación; todas ellas de la aplicación SGNM.

* 1. **Definiciones, Acrónimos, y Abreviaturas**

Ver el [Glosario de la aplicación SGNM](https://drive.google.com/open?id=1_myKc3c3AKwRPoGps5xxZ48vzIQmPEioEmvz1Ty-dH8).

* 1. **Referencias**

[Casos de uso](https://drive.google.com/open?id=1zO8UUFz-P8bC-23Jmxn9wQXkuSFVcXKcDo3OigBg0AA), diagramas de paquetes, diagrama de clases, diagrama de despliegue, diagrama de componentes y diccionario tecnologías.

1. **Objetivos y Restricciones de Arquitectura**

El sistema es desarrollada a través de una web api en .net, JSON y REST, consumidos por un cliente web en Node JS además de un cliente móvil en Android / iOS.

1. **Vista Abstracta de Alto Nivel**

En esta parte se determina una visión abstracta de alto nivel, donde se muestra la organización fundamental, en conjunto con los servicios web desarrollados para SGNM; dentro de ella se encuentran los componentes de almacenamiento de datos, los servicios web usados, el modelo de datos y los protocolos bajo los cuales el sistema se soporta.



***Arquitectura de Alto Nivel***

1. **Vista Lógica**

**Descripción de la Vista Lógica**

Vemos claramente que, según lo descrito en este y otros documentos, la aplicación del Visor de Mapas es muy cercana al patrón de arquitectura MVC (Modelo Vista – Controlador), combinado con el patrón DAO (Data Access Object) para la configuración del acceso a datos. Esto debido al carácter web de la aplicación.

Los Servicios Web de Ventas son resultado de información consumida de la Base de Datos. La interfaz es producida por componentes en Javascript, CSS y HTML y a su vez en el cliente de Android con Java.

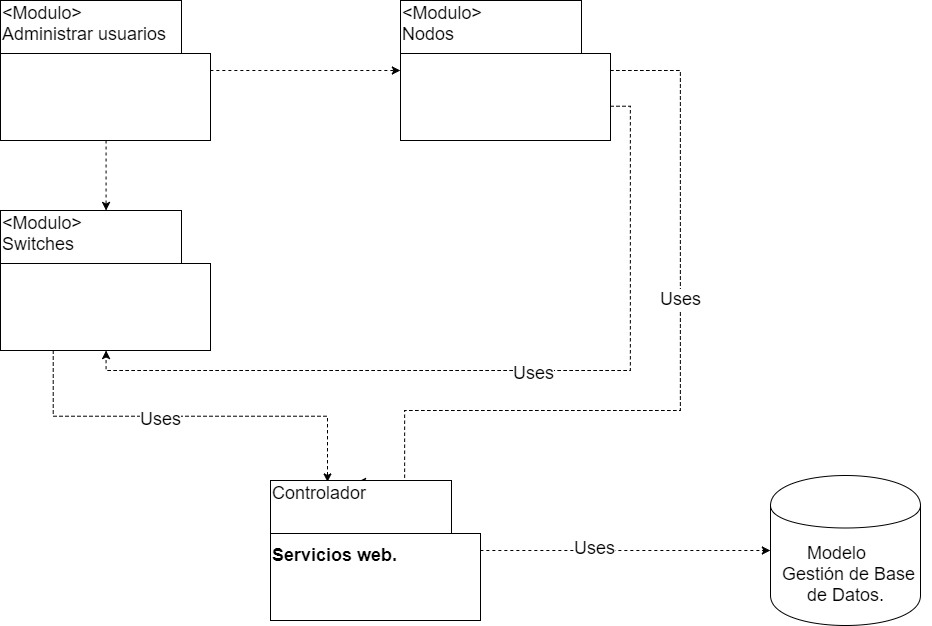
Según esto, el modelo de la aplicación se definirá de la siguiente forma:

**Modelo:** Configuración de la Base de Datos de acuerdo al Modelo de Datos propuesto (ver documento Modelo de Datos y casos de uso sobre Configuración de la Base de Datos).

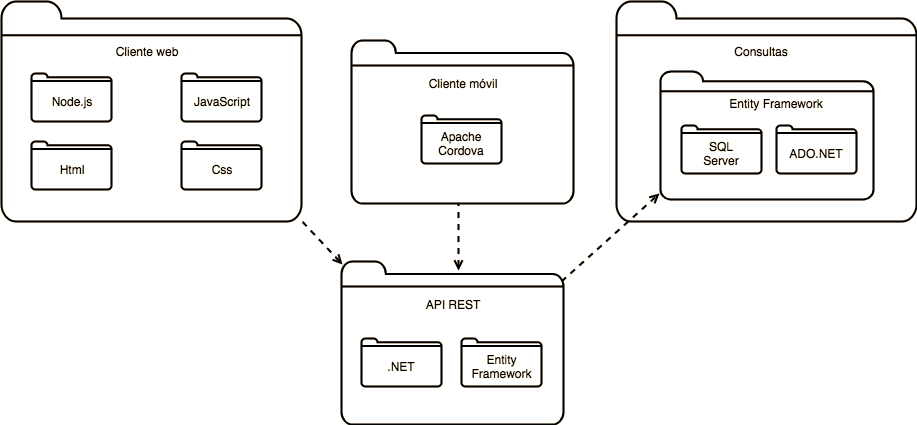
**Vista:** Interfaz de usuario en Javascript y HTML del Visor de Mapas.

**Controlador:** Servicios Web de Ventas, Productos y Usuarios.

Esto nos lleva al siguiente diagrama que explica mejor la interrelación de estos componentes:



La aplicación entonces se podría dividir arquitecturalmente en los siguientes paquetes:

***Diagrama de Paquetes***

A continuación, se describe de modo general los paquetes globales. Los demás paquetes pueden ser revisados en detalle en el modelo de casos de uso.

1. *Persistencia:* Contiene la información de configuración de la Base de Datos en SQL Server, y los datos como tal. Utiliza un conector SQLOLEDB para permitir acceso a los datos. Será el repositorio utilizado por los archivos .xsd, que serán la fuente de información que consumirá el servicio WFS.
2. *Negocio:* Contiene la configuración de los servicios web de ventas, productos y servicios. La información de cada servicio es mapeada de la base de datos en los archivos .xsd y referenciada en el archivo wfs\_configuration.xml, lo que permitirá consumir el servicio WFS, que no tiene resultados gráficos.
3. *Interfaz:* Contiene la presentación de los elementos de una manera amigable para el usuario. La Interfaz consume los servicios web de ventas, productos y servicios mediante una referencia del servidor de información, hecha en el archivo *ArchivoDeContexto.xml.* Este archivo hace referencia además a los componentes de la Interfaz Gráfica de Usuario y a las funcionalidades hechas en JavaScript, respaldado por Java.