# 5. descripción general y diagrama en bloques

Nuestro proyecto de simulara los aspectos más importante que ocurren en una red de datos tipo LAN con tecnología ethetnet, para el desarrollo de ese simulador vamos a utilizar la plataforma de desarrollo .NET ya que esta herrmienta cuenta con todos los aspectos tecnológicos y relevantes para hacer usos de estos. Unos de los aspectos mas relevantes en la escogencia de este software, es el poder ser una base de desarrollo para futuros mejoras de nuestro proyecto, por otra parte el posser la herramientas necesarias para elaborar una plataorma multiusuario y de esta forma cumplir con nuestro objetivo de ser un sistema mutiusuario, para ser usado de forma didáctica sindo un sistema de hanbito académico.

El diseño general para el desarrollo de simulador tendrá la metodología descrita a continuación:

#### 5.1 DESARROLLO DE LA APLICACIÓN EN CAPAS:

Uno de los objetivos principales de esta aplicación, es que sea extensible sencillamente, ya que servirá como base para implementaciones especificas, o para la actualización de nuevas tecnologías. Cuando sea necesario implementar un nuevo módulo, no se tendrá que volver a escribir todo el código; el diseño de esta aplicación debe permitir la reutilización del código, y que sólo sea necesario codificar nuevas implementaciones, permitiendo que la aplicación sea extensible de una forma rápida para futuros proyectos que requieran ser complementadas.

Se podrá conseguir hacer funcionar la aplicación en forma rápida, haciendo caso omiso de ningún patrón de desarrollo de software por capas, dejando todos los módulos que la compongan altamente acoplados entre ellos, esto conllevaría a que sería imposible la extensión de esta aplicación, o que fuera necesaria volver a escribir gran parte de código, por lo que se podria correr el riesgo que la aplicación quede limitada a las funcionalidades que se van a realizar con este proyecto.

Otro inconveniente que se encuentra al no utilizar el diseño por capas, sería cuando la tecnología de .NET este obsoleta, o sean creados o actualizados nuevos módulos en el BCL, sería imposible estar a la altura de estos cambios, esto conllevaría a lo que ocurre con muchas aplicaciones, que después de un tiempo sería obsoleta porque no puede ser utilizada debido a un cambio de tecnología( como el cambio de Windows XP a Windows Vista), o si se desea utilizar nuevas tecnologías es imposible acondicionarlas, debido a que todos los componentes de la aplicación están altamente acoplados, y un cambio en cualquiera de ellas involucra un cambio en todos los componentes de la aplicación.

El otro inconveniente que encontramos al no diseñar por capas, es la dificultad para realizar pruebas unitarias a los módulos que componen la aplicación, porque estos, entre si están altamente acoplados.

Por estas razones se usará un modelo en capas en donde esté altamente diferenciado, la visualización, el modelo de negociación, y el acceso a los datos.

La visualización es la capa encargada de la interfaz con el usuario, y su única responsabilidad será la de recibir cualquier evento que sea originado por el usuario, o desplegar la información necesaria al usuario.

En este caso se utilizará para la capa de visualización, ventanas de Windows para la representación de la información.

La capa de negociación es la encargada del manejo de cálculos y procesos de los datos, esta capa no conoce la existencia de una visualización, o una capa de acceso a datos, su única función es la de realizar toda la lógica de negociación correspondiente. Esta capa será el núcleo de la aplicación en donde verdaderamente estará ocurriendo toda la lógica de la simulación.

En la capa de acceso a datos, está la abstracción de la fuente de datos, que va hacer usada por la capa de Negociación, esta capa obtendrá los datos de la topología de la red, desde un archivo, o una base datos o usando .NET Remoting para, obtener los datos de red de la capa de datos de una aplicación remota del simulador.

Con esta división por capas, es sencilla la extensión de la aplicación debido a que están altamente desacoplados los módulos principales, de esta manera si en un futuro se requiere presentar este simulador en un ambiente WEB, lo único necesario será implementar la capa de visualización en un entorno WEB, la capa de negociación y el acceso a datos quedarán intactas.

Si en un futuro la información que maneja el simulador es demasiado grande, en donde el modelo de acceso a datos de la configuración de la red, a través de archivos no pueda ser mantenida, y sea necesaria la utilización de una base de datos relacional, tan solo sería necesario cambiar la capa de acceso a datos, para que soportara este nuevo formato, las restantes capas quedan intactas, y la aplicación continuará funcionando normalmente.

Lo mismo ocurre al nivel de la capa de negociación, cuando se requiera hacer la implementación de un nuevo protocolo, o cambiar la lógica de algún dispositivo, esto no debe afectar las demás capas.

#### 5.2 COMUNICACIÓN ENTRE LAS DIFERENTES CAPAS.

Para la comunicación entre las 3 capas se va a usar la metodología Modelo Vista Presentación.

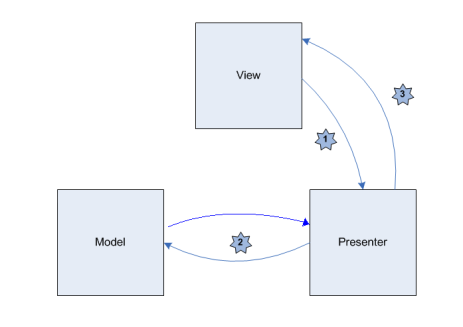
Esta metodología, es muy recomendable para desacoplar las capas de visualización de negociación, como también para que se puedan realizar pruebas unitarias tanto a la visualización como al modelo de negociación.

El modelo es el objeto donde residen los datos de la aplicación y provee métodos para acceder a ellos, en este también esta toda la lógica de negociación, el modelo no debe tener ninguna asociación con la interfaz con el usuario, no contiene una referencia a la vista, el modelo envía eventos cuando sus datos cambian, en el caso del simulador, van a existir diferentes modelos, cada uno para los diferentes dispositivos que existan y diferentes protocolos a implementar.

La vista es la encargada de mostrar los datos del modelo al usuario, en el caso del simulador la vista será una ventana de Windows, en donde estará el mapa de la topología de red, y la información de cada dispositivo, la vista notificará mediante eventos, cuando intervino el usuario.

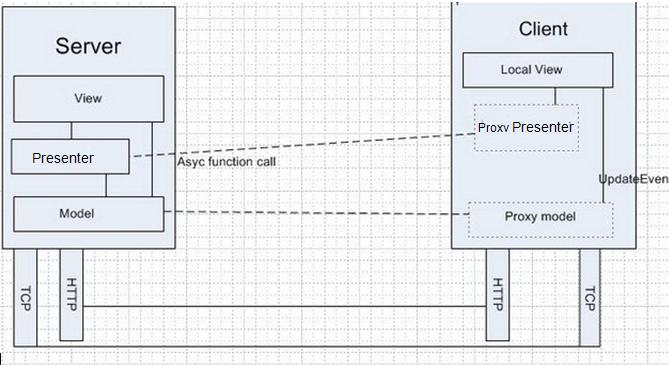
La presentación es la encargada de sincronizar y es la mediadora entre la vista y el modelo, la presentación conoce tanto al modelo como a la visualización y continuamente está escuchando los eventos de acción del usuario; una vez la presentación detecte este evento leerá los datos de la visualización y los transportará al modelo, la presentación también escucha los eventos de cambio del modelo, para que cuando ocurran, la visualización sea refrescada con los cambios que ocurrieron en el modelo.

En la aplicación se usará esta metodología por cada componente que exista, por lo que tendrán varios MVPs, y existirá un modelo global MVP, el cual contendrá varios MVPs , que en este caso será el mapa de la red.



Para la implementación remota entre diferentes sesiones del simulador se usara .NET remoting junto con el modelo MVP, en donde una sesión del simulador se convertirá en el servidor, y en este estará todo el modelo de red de todas las sesiones las cuales serán los clientes, en el servidor también se encontrará el presentador, y una vista, cada uno las demás sesiones que estén conectadas entre sí, solo serán visibles.

Mediante este modelo estableceremos la conexión entre diferentes sesiones del simulador, en donde será posible la comunicación entre diferentes componentes de red de las sesiones que estarán contenidas en un mismo modelo de red la cual se encontrará en el servidor.



la descripción general que tiene el presente proyecto de grado es la siguente: