# 4. MARCO CONCEPTUAL

Para delimitar el marco conceptual en el que se desarrolla este proyecto hay que hacer dos distinciones, uno es el marco teórico correspondiente a las redes de datos tipo LAN con tecnología Ethernet, el otro es el marco de referencia de programas de simulación ya existentes respecto a sistemas de comunicaciones; esto se complementará explicando brevemente nuestra plataforma de trabajo para el desarrollo del software .NET

## 4.1 REDES DE DATOS LAN TECNOLOGÍA ETHERNET

Una red datos es un sistema de intercomunicación e intercambio de información entre terminales o periféricos, las redes se originaron por la necesidad de compartir información de forma dinámica entre computadores, dando la opción que varios usuarios dispusieran de ella de forma sencilla y rápida. De esta forma el compartir información fue indispensable para el eficiente uso de las nuevas tecnologías. Hoy en día existen muchas clases de redes de datos. Estas están clasificadas según su tamaño, su método de conexión, relación funcional, tipo de transmisión, topología y según la tecnología en las que fueron implementadas que corresponde a los protocolos de intercambio de información. Entre la clasificación según el tamaño y según la tecnología tenemos:

|  |
| --- |
| * **Tipos de red según su tamaño** |
| * + Red de área personal (*PAN*) |
| * + Red de área local (*LAN*) |
| * + Red de área metropolitana (*MAN*) |
| * + Red de área amplia (*WAN*) |

Tabla Clasificación de redes según su tamaño

|  |
| --- |
| * **Tipos de red según su tecnología** |
| Ethernet IEEE 802.3 |
| Token Ring IEE 802.5 |
| Wi-Fi IEEE802.11 |
| Bluetooth IEE 802.15 |

Tabla Clasificación de redes según su tecnología

El objetivo de nuestro proyecto de grado se enfoca en la redes de Datos tipo LAN con tecnología Ethernet, usando como medio de transmisión cable de par trenzado UTP.

Un protocolo de comunicación es un conjunto de reglas y convenciones que son usadas como parámetros para establecer un canal de intercambio de información entre dos dispositivos, estos datos serán interpretados correctamente por cada dispositivo, según los protocolos que originaron los paquetes de datos y de la arquitectura del dispositivo. Esto con el fin que diferentes componentes del dispositivo entre los que se creó el canal de comunicación hablen un mismo idioma.

**Capa física**: Señal y transmisión binaria

**Capa enlace de datos**: Direccionamiento físico

**Capa de red**: Direccionamiento Lógico

**Capa Transporte**: Conexión entre dispositivos

**Capa sesión**: Comunicación entre dispositivos

**Capa de representación**: Representación de Datos

**Capa de aplicación**: Aplicaciones de usuarios

Ilustración 1 Modelo OSI

Para el estudio de este tipo de redes usaremos el modelo de referencia OSI (Open System Interconnection), este es usado como marco de referencia de la arquitectura de este tipo de redes.

Este modelo de referencia está dividido en siete capas, cada una tiene determinada el manejo de un conjunto de Protocolos que son usados para establecer un canal de comunicación entre dispositivos, estas capas se aprecian en la grafica N°1

Cada una de estas capas permite fraccionar el proceso de comunicación en una red de datos.

Dentro de las funciones y características de cada una de estas capas tenemos lo siguiente:

### 4.1.1 Capa física

La capa física es la encargada de las conexiones físicas entre los equipos de la red. En esta se definen las características y los valores nominales (como voltajes, polaridad, tiempo de duración del bit y otras) del medio por el que se transfiriere la información.

### Capa de enlace

Esta es la capa encargada de recibir los bits provenientes de la capa física y comprobar si durante la transmisión del mensaje hubo alguna variación de este. Esta capa fragmenta los mensajes que se están recibiendo, los analiza y determina si hay algún error en el mensaje recibido, por medio de protocolos con algoritmos que gestionan dichos bits erróneos, perdidos o duplicados. En esta capa también se maneja la dirección física de los dispositivos, esta sirve cómo identificación para cada puerto Ethernet, este número es único e irrepetible por cada uno.

### 4.1.3 Capa de red

Esta capa determina cual es el mejor camino que debe usarse para enrutar los paquetes transmitidos, aquí se incorpora la dirección lógica, origen y destino de la información. Existen diferentes algoritmos tanto dinámicos como estáticos, que determinan cual es la mejor ruta, dependiendo de diferentes variables como son tiempo, longitud del camino y estado del mismo.

### 4.1.4 Capa de transporte

Esta es la primera capa donde hay comunicación directa entre host. El objetivo es transportar eficientemente la información. Aquí se identifica cual fue el proceso o programa para lo que fue enviada la información. Esto es posible, ya que toda aplicación que se ejecuta en el host transfiere su información a través de un puerto identificado y distinguido de forma única, así si el host está ejecutando varias aplicaciones la capa de red sabe a cual pertenece cada paquete de información recibido y enviado.

### 4.1.5 Capa de sesión

De forma complementaria a la capa de transporte, la capa de sesión maneja servicios adicionales para la mejor gestión de la transmisión de la información, Por ejemplo, puede manejar “tokens (objetos abstractos y únicos) para controlar las acciones de participantes o puede hacer checkpoints (puntos de recuerdo) en las transferencias de datos”.

### 4.1.6 Capa de presentación

En esta capa se gestionan protocolos y funciones comunes para muchas aplicaciones tales como traducciones entre caracteres, códigos de números, etc.

### 4.1.7 Capa de aplicación

Finalmente en esta capa son gestionados los protocolos usados por las aplicaciones individuales.

Estos son los aspectos más relevantes que debemos tener en cuenta a la hora de comenzar nuestra investigación para el desarrollo de nuestro simulador. Existe mucha información referente a este tema, en esta se analiza más detalladamente cada una de las características de los protocolos de red existentes. Para nuestro objetivo es importante dar a conocer los aspectos relevantes desde nuestro punto de partida para el desarrollo de nuestro software.

## 4.2 TEORÍA DE OTROS SIMULADORES

## Software actual para la simulación de redes de comunicaciones

Para ubicar puntualmente el software de simulación de nuestro proyecto hicimos un estudio detallado de algunas de las principales y más importantes herramientas de software para la simulación de redes de comunicaciones, estas son:

* FLAN es un software de propósito general para la simulación de redes de comunicación, este programa hace el análisis de las redes asociando su estructura basada en nodos y enlaces, con bloques simples, por medio de los cuales se puede entender el funcionamiento especialmente de los protocolos de enrutamiento que maneja la capa de red. La limitación es que el usuario debe contar con conocimientos básicos de programación en Java, en el momento de definir características y parámetros de los dispositivos.
* PACKET TRACER es un simulador gráfico de redes desarrollado y utilizado por Cisco como herramienta de entrenamiento para obtener la certificación CCNA (Cisco Certified Network Associate), este ofrece como ventaja principal el análisis de la simulación de acuerdo al modelo OSI de capas, pero tiene la limitación que es un software de tipo propietario y habría que pagar la licencia para su utilización además que su uso es exclusivo para dicho entrenamiento. Además no permite crear topologías de red que involucren la implementación de tecnologías diferentes a Ethernet.
* OPNET MODELER este paquete de simulación es muy utilizado en la industria del diseño de redes de comunicaciones, muy utilizado en la investigación y desarrollo, es un software de tipo propietario, diseñado especialmente para la industria por lo que el tiempo para el aprendizaje y manejo de este software es largo y riguroso.
* KIVA es un simulador de redes basado en Java que permite especificar diferentes esquemas de redes de datos y simular el encaminamiento de paquetes a través de dichas redes, esta es una herramienta software orientada principalmente a simular el comportamiento del protocolo IP, y especialmente para el estudio del tratamiento de los datagramas y el encaminamiento de los mismos por una red. En la versión actual, la interfaz de usuario está implementada con un conjunto de clases, las cuales deben ejecutarse en el equipo del usuario, además, cada vez que se desee trabajar con éste programa se deben descargar varios archivos para poder instalar el programa.
* COMNET III es una herramienta comercial orientada al diseño, configuración y estudio de las redes de comunicaciones, desarrollado por CACI Products Inc. Por medio de este programa es posible crear topologías de redes complejas, configurar varias tecnologías, protocolos y dispositivos de red, para hacer un análisis detallado del funcionamiento y del rendimiento de redes tipo LAN, MAN y WAN. Como desventaja principal tenemos que es un software propietario. Por ser una de las herramientas de simulación más completas del mercado, la programación de los parámetros de los dispositivos y enlaces de la red tiende a ser compleja. Además de los conocimientos sobre el manejo y el diseño de redes de comunicaciones, se requieren conocimientos en otras áreas como por ejemplo la estadística.

Con el estudio de las características y componentes del software anteriormente expuestos y las necesidades particulares de la materia de integración de redes, definimos y delimitamos las características de software del simulador que desarrollaremos en nuestro proyecto. La ventaja más grande que tiene nuestro proyecto en comparación con los simuladores actuales, es el diseño y la plataforma sobre la que se va a diseñar el software, este se implementara sobre la plataforma .NET, diseñada por Microsoft este ambiente de diseño se escogió por contar con un “Entorno Común de Ejecución para Lenguajes” o CLR (Common Language Runtime), tal como se describe en breve.

## 4.3 .NET FRAMEWORK Y EL ENTORNO COMÚN DE EJECUCIÓN PARA LENGUAJES (CLR)

Uno de los ideales de la ciencia de la computación a través de los últimos años, ha sido el concepto de que un programa de computadora puede ser escrito para que sea ejecutado sin importar el Hardware con el que este compuesto.

Hasta cierto punto este ideal se ha logrado al poder abstraer el acceso al Hardware utilizando un sistema operativo, desde este punto de vista la función básica de un sistema operativo es encapsular la capa de Hardware de un sistema, para que un programador no tenga la necesidad de preocuparse por registros, memoria, y otros problemas que ocurren a nivel de Hardware.

En los últimos años, este problema volvió a surgir, pero esta vez ya no a nivel de Hardware, sino a nivel de un sistema operativo, en el mercado actual se encuentran diferentes sistemas operativos Win32, Solaris, OSs, inclusive de un mismo Proveedor como por ejemplo Microsoft contiene una serie de sistemas operativos diferentes (Win98, WinXp, WinVista). El problema radica ahora en darle al programador, otro nivel de abstracción en donde esté encapsulado el sistema operativo cuando se está desarrollando una aplicación. En muchos sentidos esto es exactamente lo que realiza Microsoft con .NET.

El Entorno Común de Ejecución para Lenguajes o CLR (Common Language Runtime) por sus siglas en ingles, el cual es uno de los pilares del Framework o marco de trabajo de .Net , forma un ambiente que abstrae el sistema operativo.

MARCO DE TRABAJO .NET

SISTEMA OPERATIVO

SISTEMA OPERATIVO

HADWARE

HADWARE

HADWARE

Ilustración 2 Ambiente de Abstracción del Sistema Operativo.

.NET abstrae los sistemas operativos, los cuales abstraen el acceso a Hardware, esto significa que al desarrollar software, se está escribiendo código encaminado al CLR y no directamente al sistema operativo, por lo que este software puede correr sobre diferentes sistemas operativos o cualquier plataforma que implemente el CLR. El CLR está presente en el núcleo del Framework de .NET, este provee un ambiente en donde las aplicaciones son ejecutadas, esto incluye conceptos como compilación, registro y hasta problemas de desarrollo.

Como su nombre lo implica el CLR está diseñado para soportar diferentes lenguajes de programación de forma en común entre ellos. Esto es muy interesante ya que los desarrolladores de software no tienen que aprender un nuevo lenguaje de programación, para trabajar sobre .NET.

Microsoft creó los siguientes lenguajes de programación encamidos para trabajar sobre .NET, los principales son: VB,C#,C++, JScript. Otras firmas están trabajando para desarrollar lenguajes de programación encaminados a .NET como COBOL, Phyton, y posiblemente JAVA.