



---

## 2. MARCO TEÓRICO

### 2.1. CARACTERÍSTICAS, VENTAJAS, DESVENTAJAS DE LAS BASES DE DATOS RELACIONALES Y METODOLOGÍAS.

El concepto de base de datos surge de la necesidad de organizar y almacenar grandes cantidades de información para después realizar consultas de forma eficaz. En la actualidad las industrias generan una gran cantidad de información la cual debe poderse analizar y modificar en su caso, de manera rápida y accesible.

Una base de datos es una colección de datos relacionados, diseñados para cumplir con las necesidades de información de una organización

Los sistemas de administración de bases de datos han evolucionado hacia sistemas de Administración de bases de datos relacionales. Una base de datos relacional es un modelo organizado de entidades que posee características que tienen relaciones entre ellas. Una base de datos relacional bien diseñada provee información de un proceso y nos permite almacenar y recuperar información de manera eficiente, esta capacidad es una ventaja ya que al almacenar información y recuperarla, la información se mantiene íntegra.

A través de un enfoque formal, la base de datos se podría definir como un conjunto de datos estructurados, fiables y homogéneos, organizados independientemente en una máquina, accesibles en tiempo real, compartibles por usuarios concurrentes que tienen necesidades de información diferente y no predecible en el tiempo.

Para cumplir con las expectativas del concepto de bases de datos se requiere tomar en consideración los siguientes puntos:

- Seguir una metodología.
- El modelado de los datos.
- Una estructura formal (tablas, columnas y filas).



- Un gestor de bases de datos.

### **2.1.1. Sistemas de Bases de Datos.**

Un sistema de bases de datos es una forma de organización que a través de una computadora pueda contener registros, es decir una colección de archivos de datos con los cuales se puedan realizar múltiples operaciones.

El registro de los datos queda de forma permanente en varios archivos y se utilizan diversas aplicaciones para extraer los registros o consultas o añadir registros a los archivos adecuados.

### **2.1.2. Ventajas del uso de una Base de Datos**

- Se puede acceder a cualquier dato en todo momento.
- Es integra.
- Evita la duplicidad de registros, es no redundante.
- Integridad referencial, elimina todos los registros relacionados dependientes.
- Define unívocamente a todos los demás atributos de la tabla a través de una llave primaria.
- Establece la relación existente en dos tablas a través de una llave foránea.
- Favorece la normalización por ser más comprensible y aplicable.
- Factores de seguridad.
- Esta organizada en tablas.
- Poseen una llave primaria.



### 2.1.3. Desventajas del uso de una Base de Datos

- Presentan deficiencias con datos gráficos, multimedia.
- No se manipulan de forma manejable los bloques de texto como algún tipo de dato.
- Complejidad debido a la cantidad de operaciones y a la capacidad del sistema manejador de bases de datos.
- Costo de la base de datos.
- Tiempo de ejecución sobre todo si es un sistema en tiempo real.
- Si la base de datos es centralizada existe pérdida de datos en caso de falla.
- Fallas de funcionamiento.
- Problemas de seguridad.

### 2.1.4. Conceptos Básicos

El modelo entidad-relación, es un modelo que consiste en objetos llamados entidades las cuales son una percepción de los datos en el mundo real, y de relaciones entre estos. Se desarrolló para representar la estructura lógica de los datos a manera de un esquema de la empresa y su estructura. La importancia del modelo entidad-relación es la representación del significado de los datos.

El modelo de datos utiliza tres conceptos básicos para su desarrollo, que son los conjuntos de entidades, los conjuntos de relaciones y los atributos.

#### 2.1.4.1. Conjunto de Entidades

Una entidad es un objeto que se distingue de todos los demás objetos, tiene un conjunto de características que hacen propio al objeto y se le asignan algunos valores



## MARCO TEÓRICO

---

para algún conjunto de propiedades que lo puedan identificar de otro objeto de manera unívoca.

Una entidad se representa a través de un conjunto de atributos, dichos atributos describen características que tiene cada objeto de un conjunto de entidades. Al designar un atributo para cada conjunto de entidades, se manifiesta que la base de datos almacena información de la misma especie y que concierne a cada entidad del conjunto de entidades, sin embargo cada entidad tiene su propio valor para cada uno de sus atributos.

Un conjunto de entidades, debe ser del mismo tipo y sus características son que comparten las mismas propiedades o atributos.

### **2.1.4.2. Conjunto de Relaciones.**

Una relación es una asociación entre diferentes entidades; un conjunto de relaciones es una asociación de relaciones del mismo tipo que puede tener de 2 o más conjuntos de entidades.

Los conjuntos de entidades que participan en un conjunto de relaciones generalmente son distintos, esto implica que el rol de cada uno no se especifica normalmente, sin embargo existen conjuntos de entidades que participan en una relación más de una vez con diferentes roles, este tipo de conjunto de relaciones, generalmente se llama recursivo y es necesario especificar el rol de cada entidad como parte de la relación.

### **2.1.5. Cardinalidad.**

El modelo de datos debe tener la capacidad de representar el número de ocurrencias de objetos que se dan en una relación, la correspondencia de cardinalidad o razón de



cardinalidad, expresa el número de entidades a las que otra entidad se asocia a través de un conjunto de relaciones.

La cardinalidad normalmente se expresa como “uno” o “muchos” y las relaciones se dan de la siguiente manera:

### 2.1.5.1. Uno a Uno

Una entidad de A se asocia únicamente con una entidad de B, y una entidad de B se asocia únicamente con una entidad de A como se muestra en la **Figura. 2.1.5.1.1.**

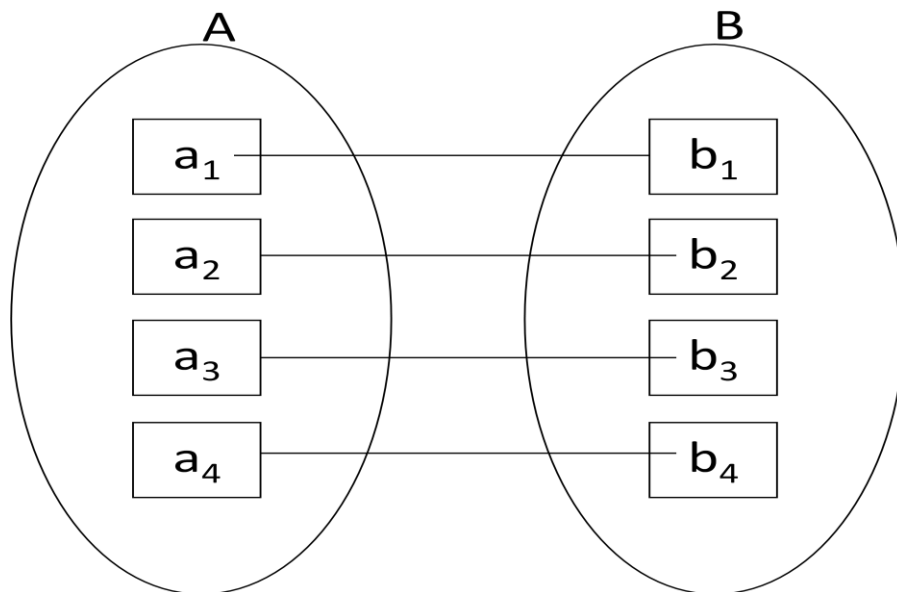


Figura 2.1.5.2.1.1.1 de Cardinalidad Uno a Uno.

### 2.1.5.2. Uno a muchos

Una entidad de A se asocia con cualquier entidad de B, sin embargo la entidad de B sólo se puede asociar con una única entidad de A, como se muestra en la **Figura 2.1.5.2.1.**

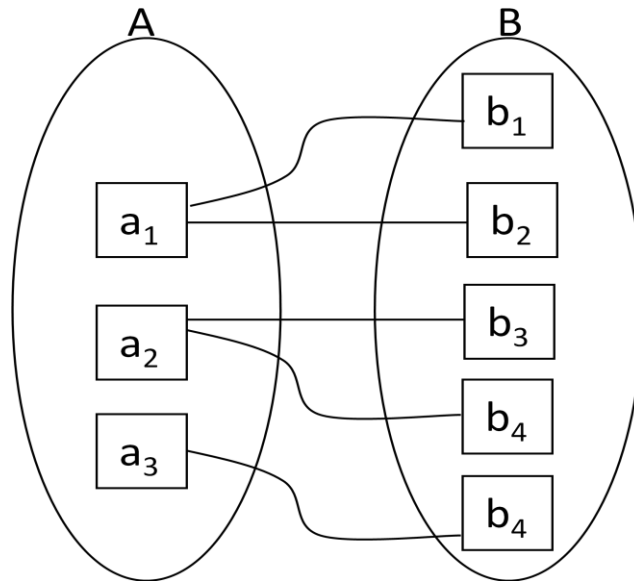


Figura 2.1.5.2.1. de Cardinalidad Uno a Muchos.

### 2.1.5.3. Muchos a uno

Una entidad de A se asocia únicamente con una entidad de B, sin embargo la entidad de B se puede asociar con cualquier número de entidades de A, como se muestra en la **Figura 2.1.5.3.1.**

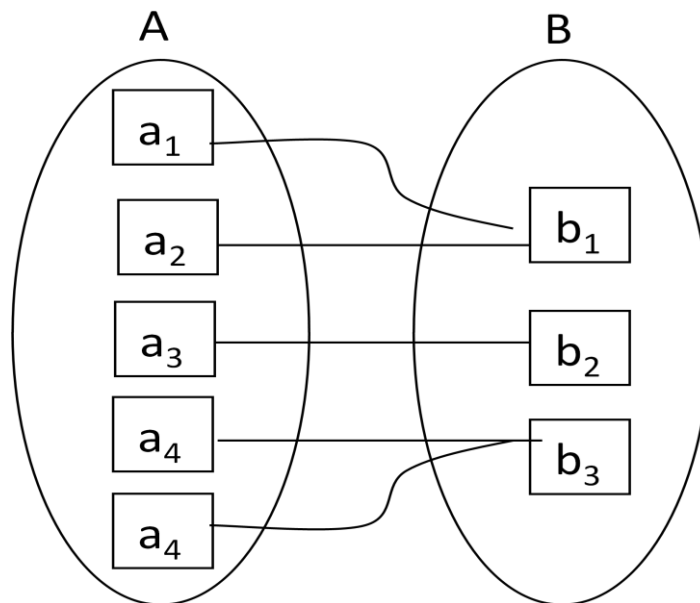
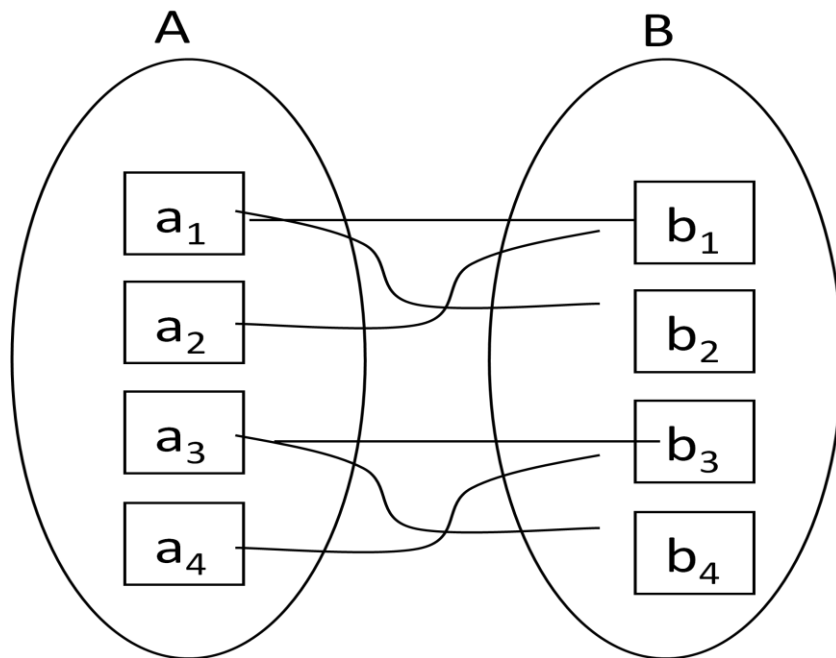


Figura 2.1.5.3.1. Cardinalidad Muchos a Uno.



**2.1.5.4. Muchos a Muchos**

Una entidad de A se puede asociar con cualquier entidad de B y las entidades de B se pueden asociar a cualquier entidad en A, como se muestra en la **Figura 2.1.5.4.1.**



**Figura 2.1.5.4.1. Cardinalidad Muchos a Muchos**

La correspondencia de cardinalidades apropiada para un conjunto de relaciones depende de la situación que se esté modelando.

**2.1.6. Modelo Relacional**

Una base de datos relacional consiste en un conjunto de tablas a las cuales se les asigna un nombre exclusivo. Cada tabla representa una relación entre un conjunto de valores, esto implica que una tabla es equivalente a una relación del modelo Entidad-Relación.



## MARCO TEÓRICO

---

El modelo relacional se encarga de la estructura de la información, la manipulación de la información y la integridad de los datos, y esto lo hace a través de dos conceptos importantes, el álgebra relacional y el cálculo relacional.

### 2.1.6.1. Álgebra Relacional.

El álgebra relacional es un conjunto de operadores que toman relaciones como sus operandos y regresan una relación como resultado. Las operaciones son unión, intersección, diferencia, producto, restricción, proyección, agrupación y división, más un operador para renombrar atributos.

La finalidad principal del álgebra relacional es la recuperación de datos, además de permitir la escritura de expresiones relacionales. A su vez, estas expresiones atienden diversos fines, como la recuperación, la actualización, algunas restricciones de integridad, y permite controlar la concurrencia, así como la autorización a acceder a algunos datos, es decir las restricciones de seguridad.

Los operadores algebraicos básicos son:

- **Unión** (Union). La unión de dos relaciones compatibles  $R_1$  y  $R_2$  es una nueva relación  $R_3$ , también compatible, cuyo esquema es igual al esquema de  $R_1$  y  $R_2$ , y cuya extensión está formada por la agrupación, sin repetición, de las extensiones de  $R_1$  y  $R_2$ .
- **Diferencia** (Minus). La diferencia de dos relaciones compatibles  $R_1$  y  $R_2$  es una nueva relación  $R_3$ , también compatible, cuyo esquema es igual al esquema de  $R_1$  y  $R_2$ , y cuya extensión está formada por aquellos registros de la relación  $R_1$  que no se encuentran en la relación  $R_2$ .
- **Selección** (Select). La selección sobre una relación  $R_1$  mediante una cualificación  $Q$  es una nueva relación  $R_2$ , cuyo esquema es igual a  $R_1$ , y cuya extensión está formada por todos aquellos registros de  $R_1$  que satisfacen la cualificación  $Q$ .





## MARCO TEÓRICO

---

- **Proyección** (Project). La proyección sobre una relación R1 con esquema R1.ai, R1.aj,..., R1.az mediante un subesquema  $S(R1) \equiv R1.am, R1.an, \dots, R1.ap$  donde  $am \geq ai$  y  $ap \leq az$  es una nueva relación R2, cuyo esquema es igual al subesquema S(R1), y cuya extensión es igual a todos los registros de R1 sin repetición sobre el subesquema S(R1).
- **Producto Cartesiano** (Product). El producto cartesiano de dos relaciones R1 y R2 no necesariamente compatibles es una nueva relación R3, cuyo esquema es igual a la concatenación de los esquemas R1 y R2, y cuya extensión está formada por el conjunto de registros que se obtiene de concatenar cada uno de los registros de R1 con todos y cada uno de los registros de R2.

Los operadores algebraicos avanzados son:

- **Intersección** (Intersect). La intersección de dos relaciones compatibles R1 y R2 es una nueva relación R3, también compatible, cuyo esquema es igual al esquema de R1 y R2, y cuya extensión está formada por el conjunto de registros que son comunes a R1 y R2.
- **Reunión** (Join). La reunión de dos relaciones R1 y R2 no necesariamente compatibles, pero en las que existe al menos un atributo con el dominio común, sobre una cualificación Q, es una nueva relación R3 cuya intención está formada por la concatenación de las intenciones de R1 y R2, y cuya extensión está formada por los registros que resultan del producto cartesiano de  $R1 \times R2$  que satisfacen la cualificación Q.
- **División** (Division). La división de una relación R1 con esquema R1.ai, R1.aj,..., R1.az entre una relación R2 de esquema R1.am, R1.an,..., R1.ap donde  $am \geq ai$  y  $ap \leq az$ , es una nueva relación R3 cuyo esquema es igual a la diferencia del esquema de R1 menos el esquema de R2, y cuya extensión es igual a todos los registros de R1 sin repetición para los cuales está presente toda la extensión de la relación R2.



### 2.1.6.2. Cálculo Relacional

El cálculo relacional es un lenguaje de consulta que simplemente proporciona una notación para establecer la definición de la relación deseada, y esto lo hace en términos de relaciones dadas de tipo declarativo. El cálculo relacional es meramente descriptivo.

El cálculo se presenta en dos versiones, cálculo de tuplas y cálculo de dominios. La diferencia entre ellos es que las variables del cálculo de tuplas abarcan relaciones mientras que las variables del cálculo de dominios abarcan dominios.

Una tupla se define como una función finita que asocia unívocamente los nombres de los atributos de una relación con los valores de una instancia de la misma, es decir, es una fila de una tabla relacional. Un atributo es una columna; al número de tuplas se le llama cardinalidad y al número de atributos se le denomina grado.

Un dominio es un conjunto de valores, de donde se toman los valores de atributos específicos de relaciones específicas.

### 2.1.7. Formas Normales.

La normalización es un proceso que involucra organizar los datos de una base de datos estableciendo las relaciones que existen entre ellas de acuerdo a reglas diseñadas para proteger los datos, que éstas sean flexibles y eliminar la redundancia y las dependencias respecto a las aplicaciones que las usan.



### 2.1.7.1. Primera Forma Normal.

Un dominio es atómico si se considera que los elementos del dominio son unidades indivisibles. Se dice que el esquema de una relación  $R$  está en la primera forma normal ó 1FN si los dominios de todos los atributos de  $R$  son atómicos.<sup>15</sup>

### 2.1.7.2. Segunda Forma Normal.

Se dice que un atributo o conjunto de atributos tiene dependencia funcional de otro u otros si a cada uno de los primeros le corresponde sólo uno de los segundos.

Una tabla está en Segunda Forma Normal ó 2FN cuando está en 1FN y todo atributo que no pertenece a la clave primaria tiene una dependencia funcional de la clave completa y no de parte de ella. Luego, si la clave principal está formada por un solo atributo y ya está en 1FN, ya estará en 2FN.<sup>16</sup>

### 2.1.7.3. Tercer Forma Normal.

Se dice que hay dependencia funcional transitiva entre dos atributos cuando un atributo que no pertenece a la clave primaria permite conocer el valor de otro atributo.

Una tabla está en Tercera Forma Normal ó 3FN si está en 2FN y no existen atributos que no pertenezcan a la clave primaria que puedan ser conocidos mediante otro atributo que no forma parte de la clave primaria, es decir, no hay dependencias funcionales transitivas.<sup>17</sup>

---

<sup>15</sup> Silberschatz, Abraham, et al., *Fundamentos de Bases de Datos*. (España, McGraw-Hill, 2002)

<sup>16</sup> DATE, C. J., *Introducción a los sistemas de Bases de Datos*.(Prentice Hall, 2003)

<sup>17</sup> DATE, C. J., *Introducción a los sistemas de Bases de Datos*.(Prentice Hall, 2003)



### 2.1.7.4. Cuarta Forma Normal.

La relación  $R$  está en 4FN si y solamente si siempre que existan subconjuntos  $A$  y  $B$  de los atributos de  $R$ , tales que la Dependencia multivaluada no trivial<sup>18</sup>  $A \twoheadrightarrow B$  se satisfaga, entonces todos los atributos de  $R$  son también dependientes *funcionalmente* de  $A$ .<sup>19</sup>

### 2.1.7.5. Quinta Forma Normal.

Una relación  $R$  está en 5FN, también llamada forma normal de proyección-junta si y solamente si cada dependencia de junta no trivial, si y solamente si una de las proyecciones  $A, B, \dots, Z$  es la proyección identidad de  $R$ , válida para  $R$  que está implicada por las claves candidatas de  $R$ .<sup>20</sup>

### 2.1.8. Metodologías de Desarrollo.

A partir del nacimiento del concepto de bases de datos, surge la necesidad de contar con una metodología de desarrollo para generar un software capaz de manejar la gran cantidad de información que surge de la creación de la base de datos. La Ingeniería de software es la disciplina que provee de métodos y técnicas para desarrollar y mantener software. La Ingeniería de software sistematiza procesos con el fin de minimizar los riesgos y los fallos en la ejecución del objetivo a desarrollar.

---

<sup>18</sup> Una Dependencia Multivalorada  $A \twoheadrightarrow B$  es trivial si  $A$  es un superconjunto de  $B$  o la unión de  $A$  y  $B$

<sup>19</sup> DATE, C. J., *Introducción a los sistemas de Bases de Datos*.(Prentice Hall, 2003)

<sup>20</sup> DATE, C. J., *Introducción a los sistemas de Bases de Datos*.(Prentice Hall, 2003)



## MARCO TEÓRICO

---

Esta área toma como referencia las fases del ciclo de vida del desarrollo de cualquier tipo de sistema de información y las aplica a diversas áreas como negocios, investigación, logística, Internet, etc.

Para lograr el objetivo deseado fue necesaria la recomendación de uso de técnicas que aseguraran en lo posible que los procedimientos de diseño de las aplicaciones como la documentación asociada estuvieran formalizados y reglamentados a manera de que cualquier persona pudiera leer y llegado el caso modificar cualquier aplicación sin necesidad de preguntar a su creador.

### **2.1.8.1. Metodología Merise.**

Merise es una metodología integrada de análisis, concepción y gestión de proyectos, desarrollada en Francia. Esta metodología se va a desarrollar enfocada a la aplicación como si ésta se estuviera moviendo en 3 ejes: Abstracción, Decisión y Vida.

En el ciclo de abstracción, es el nivel conceptual el que indica lo que el sistema tiene que resolver, trabaja con los objetos, entidades y describe propiedades; en el nivel organizativo indica quién lo hace, cuándo y dónde, transforma el modelo conceptual en estructural, y provee la lógica de los datos; en el nivel técnico indica cómo se hace, se concreta la estructura final en función del gestor a utilizar.

En el ciclo de decisión se plantea el sistema, las personas que están involucradas en la toma de decisiones y se toma la iniciativa de realizar o no el proyecto. Se incluyen los recursos humanos con los que cuenta la empresa y se recoge una serie de información para la toma de decisiones.

En el ciclo de vida, se tiene ya el producto desarrollado y se implanta, se generan resultados, como se puede observar en la **Figura 2.1.8.1.1**

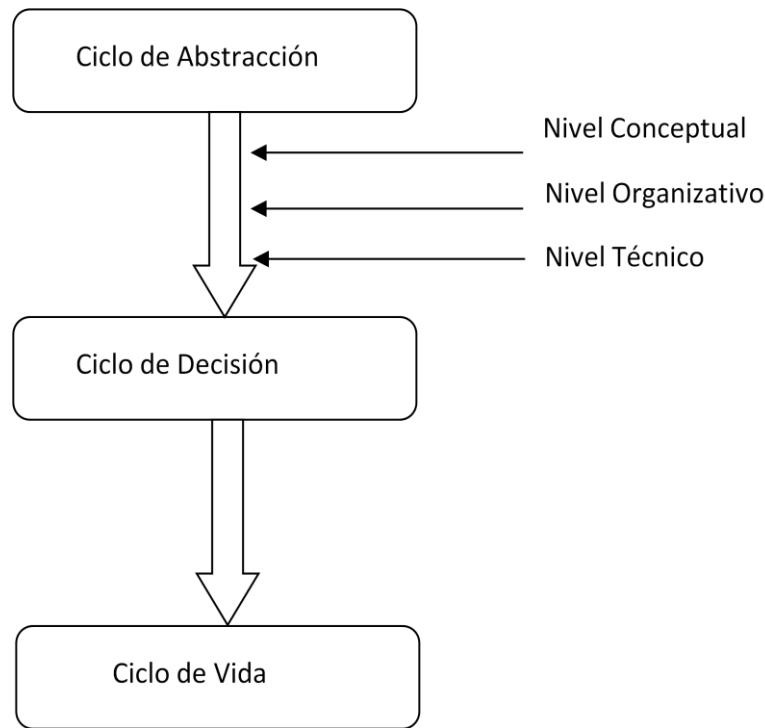


Figura 2.1.8.1.1. Metodología Merise.

### 2.1.8.2. Metodología Ssadm.

Análisis de sistemas estructurados y Método de Diseño (SSADM). Es un enfoque de sistemas para el análisis y diseño de sistemas de información. SSADM fue producido por la Agencia Nacional de Telecomunicaciones, del gobierno del Reino Unido dedicada a la utilización de la tecnología a partir de 1980.

Las tres técnicas más importantes que se utilizan en SSADM son las siguientes:

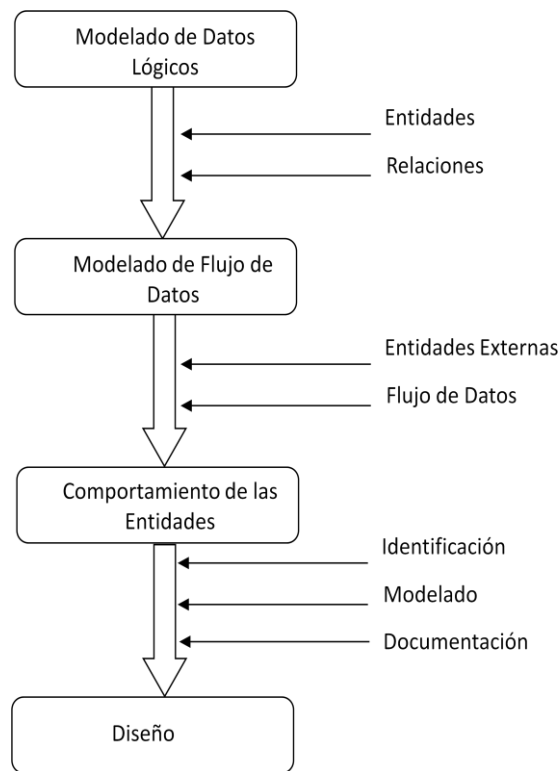
- Modelado de datos lógicos: Este es el proceso de identificación, modelado y documentación de los requisitos de datos del sistema que se está diseñando. Los datos se dividen en entidades (cosas de las que una empresa necesita para registrar la información) y relaciones (las asociaciones entre las entidades).
- Modelado de flujo de datos: Este es el proceso de identificación, modelado y documentación de cómo los datos se mueven en torno a un sistema de



información. En el modelado de flujo de datos se examinan los procesos (actividades que transforman los datos de una forma a otra), almacenes de datos (la zona de espera de los datos), las entidades externas (lo que envía o recibe datos de un sistema), y los flujos de datos (rutas porque los datos pueden fluir).

- Comportamiento de las entidades modeladas: Este es el proceso de identificación, modelado y documentación de los acontecimientos que afectan a cada entidad y la secuencia en que ocurren estos eventos.

La **Figura 2.1.8.2.1**. Muestra la metodología SSADM.



**Figura 2.1.8.2.1. Metodología SSADM**

### 2.1.8.3. Metodología Yourdon.

Esta metodología consiste en diseñar paso a paso sistemas detallados. La principal herramienta del diseño estructurado es que se genera el diagrama de estructura, donde



## MARCO TEÓRICO

---

se muestran los componentes de los diferentes procedimientos del programa, su orden jerárquico y los datos relacionados con ellos.

El diagrama de estructura es un diagrama de árbol que define la arquitectura de un programa y muestra los procedimientos y sus interrelaciones. En este diagrama se utilizan bloques básicos que representan los componentes del procedimiento y las flechas que muestran cómo se conectan.

Esta metodología se compone de 4 pasos en el proceso de diseño.

- Diagrama de flujo de datos: Se representa el problema de diseño como el flujo de datos a través de un sistema, el cual se compone de procesos que transforman los datos, y estos a su vez definen los componentes del programa.
- Trazar el diagrama de estructura: Se representa el diseño del programa como una jerarquía de componentes de procedimiento. Este diagrama se deriva del diagrama de estructura. Proporciona estrategias de diseño para el análisis de transformación y el análisis de transacción, estas estrategias ayudan a desarrollar el diseño jerárquico así como un proceso paso a paso de la transformación de cada estrategia.
  - El flujo de la información divide al diagrama de flujo en tres partes, rama aferente que es la entrada de los datos, transformación central, que es el proceso lógico y la rama deferente que es la salida de los datos.
- Evaluación del diseño: En este punto se utilizan dos técnicas que son la de acoplamiento y la de cohesión. En el acoplamiento se mide el grado de independencia entre los componentes de los procedimientos en el diagrama de estructura. En la cohesión se mide la fuerza de las relaciones entre los elementos dentro de un módulo.
- Preparación del diseño para la implantación: En esta parte se divide el diseño del programa lógico en unidades físicas de implantación llamadas unidades de carga, es decir, es un diseño físico del programa.



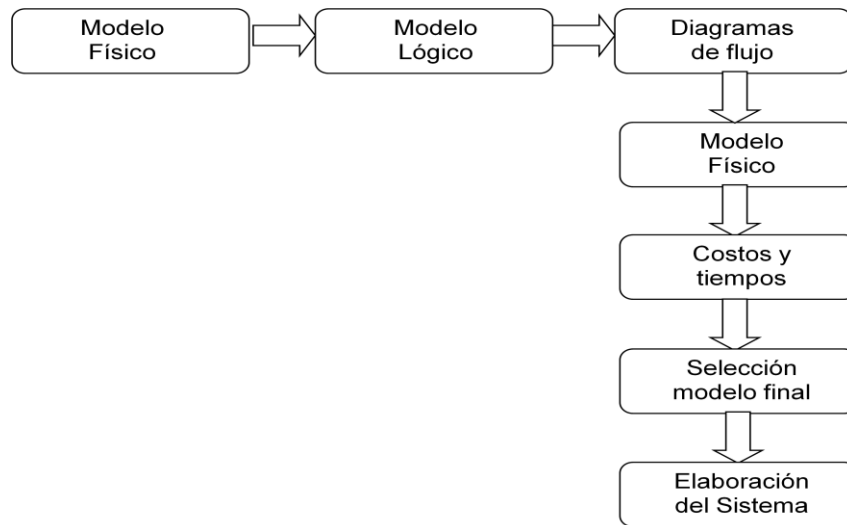


**2.1.8.4. Metodología de análisis De Marco**

La metodología de DeMarco consta de los siguientes pasos:

- Construir el modelo físico.
- Construir el modelo lógico a partir del físico.
- Construir el modelo lógico del nuevo sistema, esto es, construir una especificación estructurada que incluye los diagramas de flujo de datos, el diccionario de datos y las especificaciones de los procesos.
- Crear los nuevos modelos físicos.
- Estimar costos y tiempos para cada modelo.
- Seleccionar un modelo.
- Realizar los subsistemas en paquetes.

Vea **Figura 2.1.8.4.1.**



**Figura 2.1.8.4.1. Metodología de diseño estructurado de Yourdon/DeMarco.**



---

## 2.2. CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DEL OSCOMMERCE



En la actualidad oscommerce es una solución tecnológica, para tiendas online de comercio electrónico que cuenta con una gran cantidad de herramientas y características. Además de sus grandes ventajas que nos proporciona; se encuentra disponible de forma gratuita como una solución de código abierto basado bajo **GNU (Licencia Pública General)**.

oscommerce además de ser una fuente abierta de software se basa en el enfoque modular de e-commerce. Desde el año 2000, miles de usuarios han implementado este modelo como solución confiable que nos brinda la ventaja de ser altamente personalizable. En sus características generales posee un carrito de compra que permite la venta directa a través de internet, la gestión del sitio y mantenimiento del mismo en línea de manera sencilla y sin necesidad de mucho mantenimiento o costos por licenciamiento.

### 2.2.1. Filosofía de oscommerce

Una de las ventajas del oscommerce es que cuenta con una extensa comunidad que da soporte a la construcción del proyecto ya que ofrece a las personas que desarrollan y que tienen conocimiento en el área la oportunidad de intercambiar ideas, conocimientos y trabajar en conjunto para mejorar el sistema. La creación de



## MARCO TEÓRICO

---

contribuciones que permitan herramientas adicionales a las herramientas con las que se cuentan.

Los desarrolladores de oscommerce tienen su principal motivación en el desarrollo de este sistema de código abierto que se origina de diferentes formas que incluye el trabajo en el software para diversión, aprendizaje y satisfacer las necesidades propias para las cuales está diseñado.

El equipo de desarrolladores que realizan hoy este gran sistema es una comunidad activa que está en crecimiento y comprometida con el mejoramiento del mismo para que sea una solución para los usuarios. La opción de utilizar el software es la opción de aprender del mismo, participar y compartir.

Esta es la razón por la cual el software de código abierto tiene éxito.

### 2.2.2. Definición de Open Source

**Open Source (Código abierto)** se le conoce al software distribuido y desarrollado libremente. Tiene un punto de vista orientado a compartir el código.

La idea del concepto de *código abierto* es: cuando los programadores pueden leer, modificar, desarrollar, mejorar y redistribuir el código fuente de un programa. Los usuarios lo adaptan a sus necesidades, corrigen sus errores a una velocidad impresionante, mayor a la aplicada en el desarrollo de software convencional o cerrado, dando como resultado la producción de un mejor software.

El software libre, *open source* tiene una serie de requisitos<sup>21</sup> necesarios para que un programa pueda considerarse dentro de este movimiento, éstos son:

- Libre redistribución: el software debe poder ser regalado o vendido libremente.

---

<sup>21</sup> <http://www.opensource.org/docs/osd>



## MARCO TEÓRICO

---

- El código fuente debe estar incluido u obtenerse libremente.
- Trabajos derivados de la redistribución de modificaciones debe estar permitida.
- Integridad del código fuente del autor: las licencias pueden requerir que las modificaciones sean redistribuidas sólo como parches.
- Sin discriminación de áreas de iniciativa: los usuarios comerciales no pueden ser excluidos.
- Distribución de la licencia: deben aplicarse los mismos derechos a todo el que reciba el programa.
- La licencia no debe ser específica de un producto: el programa no puede licenciarse solo como parte de una distribución mayor.
- La licencia no debe restringir otro software: la licencia no puede obligar a que algún otro software que sea distribuido con el software abierto deba también ser de código abierto.
- La licencia debe ser tecnológicamente neutral: no debe requerirse la aceptación de la licencia por medio de un acceso por clic de ratón o de otra forma específica del medio de soporte del software.

### 2.2.3. Requerimientos para oscommerce

Hoy en día los requerimientos que necesita oscommerce se pueden considerar como requerimientos estándar de aplicaciones web; ya que se puede instalar en cualquier servidor que tenga disponible PHP, APACHE y que disponga de acceso a una base de datos de MySQL.

Pero también como cualquier script, requiere de algunas características particulares que son:

- Versión de PHP 4.1 o superior.
- Contar con una Base de datos MySQL 4 o superior.



## MARCO TEÓRICO

---

- Un administrador de base de datos como phpMyAdmin.
- Servidor web Apache.
- Espacio que requiere para instalación es de 10MB.
- Disponibilidad de uso de archivos .htaccess para personalizar características de APACHE.
- Mod\_rewrite activado para usar direcciones amigables optimizadas.

Esto incluye servidores compartidos, servidores dedicados, e incluso la instalación local en cualquier equipo que estén usando Linux, Unix, Mac OS o Microsoft Windows.

### 2.2.4. Características principales de oscommerce

Las características principales en la instalación de oscommerce que se obtienen al cargar la herramienta son:

- Compatible con cualquier versión de PHP 4.1 en adelante.
- Al finalizar la instalación todas las funciones que contiene quedan habilitadas.
- Back-End orientado a objetos.
- Diseño modular.
- Permite la carga de más de un idioma.
- Permite la carga de más de una denominación de moneda.
- Diseño 100% personalizable.
- Almacena pedidos, clientes y productos en una base de datos de fácil consulta vía web.
- Posibilidad de implementar un certificado de seguridad SSL.
- Se puede navegar por las categorías de una forma fácil e intuitiva.
- Facilidad de poder habilitar y deshabilitar categorías y/o productos, sin límite de niveles.



## MARCO TEÓRICO

---

- Buscador de productos, por categoría, marca y/o fabricante, o por palabra.
- Gestión de Marcas, productos agotados, novedades, productos más vendidos.
- Listas globales o por categorías de los productos más vistos y más vendidos.
- Enumeración de productos de cada categoría.
- Control de stock de los productos.
- Administración de clientes y pedidos.
- Posible implementación de un boletín de noticias o novedades.
- Descripciones de productos basados en HTML.
- Configuración de diversas formas de pago, el sistema soporta diversas formas de pago online y off-line como, contra reembolso, transferencia bancaria, cobro en tienda, giro, PayPal, Western Union, Rapidpago, Pagofacil, tarjeta de crédito.
- Configuración de diversas formas de envío, así como el cálculo automático de los mismos en función del destino del envío, forma del envío, peso, seguro.
- Mailing personalizados a clientes. Esto permite el envío de correos electrónicos a clientes, pudiendo utilizar datos de la ficha del cliente para personalizar los mismos. También permite enviar en el mailing listados de productos para informar de ofertas y novedades.
- Recepción de pedidos, cuando un cliente realiza un pedido desde la tienda online, este y el administrador de la tienda recibe una confirmación del mismo. Así mismo los pedidos quedan registrados en la base de datos para poder realizar una mejor administración de ellos posteriormente.
- Gestión de publicidad mediante banners fácilmente administrables.
- Compatibilidad con navegadores como: Explorer, FireFox, Opera, Safari.
- Personalización de apariencia, imágenes, textos y mensajes.

### 2.2.5. Ventajas de oscommerce

Algunas de las ventajas de oscommerce son:

- Se adapta a la venta de cualquier producto.



## MARCO TEÓRICO

---

- Requerimientos muy comunes.
- Uso de herramientas de licencia GNU.
- Instalación sencilla.
- Todas las herramientas de instalación necesarias contenidas en un solo archivo.
- Contribuciones gratuitas disponibles desde el sitio.
- Sistema de debugging, en donde se plantean los errores y se lleva un control para que sean resueltos. (Issue Tracker)
- Extensa documentación, soporte en foros y artículos.
- Evitan errores de programación al desarrollar un sistema como este hecho a la medida.
- Todas las versiones estables están probadas y en uso por miles de comercios.
- El sitio se gestiona desde un panel de control.
- Los usuarios finales no necesitan de requerimientos especiales para poder acceder a la aplicación solo necesitan una computadora dispositivo móvil que se conecte a internet.
- Es personalizable.
- Y muchas características se le pueden agregar o quitar según las necesidades.

### 2.2.6. Desventajas de oscommerce

Algunas de las desventajas que podemos encontrar en oscommerce son:

- Solo una imagen por producto.
- No cuenta con un sistema de plantillas lo que dificulta el poder modificar el diseño.
- La edición del diseño y plantillas se hace a través de código.
- La personalización del diseño es complicada.
- Para hacer cualquier modificación en el funcionamiento del sitio hay que entenderlo y tener conocimientos de PHP.



## MARCO TEÓRICO

---

- Poca documentación en español.
- Depende de las personas que voluntariamente ofrezcan su tiempo, dinero, y energía para mantener el proyecto funcionando.
- El servidor web debe soportar PHP 4.1 o superior.
- El servidor web debe tener una base de datos MySQL.
- El servidor web debe permitir la ejecución de scripts CGI.
- No es un sistema a la medida y el usuario debe de ajustarse a lo que la solución brinda.
- Las contribuciones o herramientas extras deben de adaptarse e investigarse para poderlas adaptar al sistema.
- Para poder tener más de un lenguaje debemos de crear y cargar los archivos en los directorios correspondientes.
- Instalar los add-ons o contribuciones que proporcionan los foros toma mucho tiempo, dinero y algunas veces no funcionan correctamente.

### 2.3. CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE PHP

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas dinámicas. Se usa principalmente para la interpretación del lado del servidor pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica.

Se ejecuta en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores.

PHP puede ejecutar un conjunto de operaciones con un script CGI, como procesar la información de formularios, generar páginas con contenidos dinámicos, operar con bases de datos, o mandar y recibir cookies.





## MARCO TEÓRICO

---

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta.

Aunque todo en su diseño está orientado a facilitar la creación de sitios web, es posible crear aplicaciones con una interfaz gráfica para el usuario. Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el Script solicitado que generará el contenido de manera dinámica. El resultado es enviado por el intérprete al servidor, quien a su vez lo envía al cliente.

### 2.3.1. Características de PHP



Algunas de las características del lenguaje PHP son:

- **Multiplataforma:** PHP funciona tanto en sistemas Unix o Linux con servidor web Apache así como en sistemas Windows con Microsoft Internet Information Server, de forma que el código generado por cualquiera de éstas plataformas no debe ser modificado al pasar a la otra plataforma.  
Sin embargo, las mejores prestaciones del lenguaje se obtienen trabajando en un entorno Unix o Linux con servidor web Apache.

La principal ventaja que aporta el uso conjunto de PHP y Apache es que, de esta manera, se puede compilar el intérprete PHP como un módulo de Apache, consiguiendo así que la velocidad de ejecución de una página PHP sea elevada



y que el consumo de recursos sea bajo ya que el intérprete PHP se carga una sola vez en memoria.

- **Ejecución en Servidor:** Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página PHP.



Figura 2.3.1.1. Esquema del funcionamiento de PHP

En la **Figura 2.3.1.1.** Se muestra el esquema del funcionamiento de PHP<sup>22</sup>

<sup>22</sup> <http://www.adrformacion.com/cursos/PHP/leccion1/tutorial1.html>



## MARCO TEÓRICO

---

Una página PHP no es más que un programa escrito en PHP que genera código HTML.

Cuando un navegador solicita al servidor web una página PHP, antes de enviar dicha página al cliente, se la pasa al intérprete del mismo, éste la interpreta y es el resultado de esta interpretación del programa, contenido en la página PHP, lo que termina llegando al cliente.

Características de PHP:

- Es un lenguaje basado en herramientas con licencia de software libre.
- Sintaxis cómoda. Cuenta con una sintaxis similar a la de C, C++ o Perl. Lo más destacado ocurre a nivel semántico: el tipado es muy poco estricto. Es decir, cuando creamos una variable no tenemos que indicar de qué tipo es, pudiendo guardar en ella datos de cualquier tipo. Esto es muy flexible y cómodo para el desarrollador, aunque los errores que se cometen pueden ser graves y difíciles de corregir al reducirse las posibilidades del intérprete para detectar incompatibilidades entre variables.
- Soporta objetos y herencia. Tiene soporte para la programación orientada a objetos, es decir, es posible crear clases para la construcción de objetos, con sus constructores. Además soporta herencia, aunque no múltiple.
- Cuenta con una extensa librería de funciones que facilitan enormemente el trabajo de los desarrolladores.
- Compatibilidad con bases de datos
- Expansión. PHP Está alcanzando unos niveles de uso tan elevados que hacen que su conocimiento sea algo indispensable para los profesionales del desarrollo en Internet.



### 2.3.2. Ventajas de PHP

Algunas de las ventajas por las que usamos PHP son las siguientes:

- **Costo:**  
Es un software de distribución libre, y con documentación muy amplia en internet, por lo general no tardamos en encontrar tutoriales y guías gratuitas acerca de cómo utilizar este lenguaje.
- **Sencillez y Versatilidad:**  
PHP es un lenguaje de una sintaxis muy simple, y fácil de aprender, además posee una gran variedad de funciones que pueden ser utilizadas para mejorar el rendimiento de nuestros programas.
- **Seguridad:**  
PHP es un lenguaje de uso muy común en la web, además de ser libre, esto significa que una inmensa comunidad de programadores que utilizan este lenguaje están cooperando para la mejora del motor de PHP, por lo cual es cada vez más seguro y estable a medida que pasa el tiempo y aumenta su versión, otra ventaja es que en internet se pueden encontrar muchos tips para evitar errores que puedan convertirse en bugs peligrosos en nuestros sitios web.
- **Rapidez.**  
  
PHP generalmente es utilizado como módulo de Apache, lo que lo hace extremadamente veloz. Esta completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria
- **Soporte para bases de datos, flash, y dinamismo en páginas web, que a pesar de que también encontramos estas características en otros lenguajes, PHP hace más fácil la implementación de ellas, por su versatilidad y sencillez.**



### 2.3.3. Desventajas de PHP

Las desventajas observadas en PHP son:

- No posee una abstracción de base de datos estándar, sino bibliotecas especializadas para cada motor (a veces más de una para el mismo motor).
- No posee adecuado manejo de internacionalización. Por su diseño dinámico no puede ser compilado y es muy difícil de optimizar.
- Por sus características favorece la creación de código desordenado y complejo de mantener.
- PHP no obliga a quien lo usa a seguir una determinada metodología en el momento de programar (muchos otros lenguajes tampoco lo hacen), aún estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable.

## 2.4. CARACTERÍSTICAS, VENTAJAS Y DESVENTAJAS DE MYSQL

MySQL es un sistema administrador de bases de datos relacionales<sup>23</sup> (RDBMS, Relational Data Base Management System) de código abierto. En sus inicios MySQL no contaba con soporte para algunas construcciones de SQL importantes como son las llaves foráneas, esto se reflejó en la oposición de los usuarios a su uso. Sin embargo, últimamente, MySQL cuenta con una gran cantidad de usuarios preocupados por mejorar este sistema en términos de licencia, rendimiento y facilidad de uso. Su aceptación se ha visto ayudada en parte por una amplia variedad de tecnologías como

---

<sup>23</sup> Una base de datos relacional almacena datos en tablas separadas en lugar de poner todos los datos en un gran almacén. Esto añade velocidad y flexibilidad.



## MARCO TEÓRICO

---

PHP, Perl, Python y similares que ha fomentado su uso por medio de módulos y extensiones estables y bien documentadas.

MySQL tiene un diseño multihilo que le permite soportar una gran carga de forma muy eficiente. MySQL fue creado por una empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. La parte "AB" del nombre de la compañía es el acrónimo del sueco "aktiebolag", o "stock company", o "sociedad anónima".

Las bases de datos son por lo general de utilidad, quizá la familia más útil de productos de software en informática moderna. Como muchos productos competidores, tanto gratuitos como comerciales, MySQL no es una base de datos hasta que se le asigna una estructura y forma. Podría pensar en esto como la diferencia entre una base de datos y un RDBMS (es decir, RDBMS mas requisitos de usuario es igual a una base de datos).

### 2.4.1. Características de MySQL.

La siguiente lista describe algunas de las características del software de base de datos MySQL.

- Proporciona sistemas de almacenamiento transaccional y no transaccional.
- Un sistema de reserva de memoria muy rápido basado en **threads (hilo de ejecución o subproceso)**.
- Joins muy rápidos usando un multi-join de un paso optimizado.
- Tablas hash en memoria, que son usadas como tablas temporales.
- Diversos tipos de columnas como son: enteros con/sin signo, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM, y tipos especiales.
- Registros de longitud fija y longitud variable.



## MARCO TEÓRICO

---

- Soporte completo para operadores y funciones en las cláusulas de consultas SELECT y WHERE.
- Soporte completo para las cláusulas SQL GROUP BY y ORDER BY.
- Soporte de funciones de agrupamiento (COUNT(), COUNT(DISTINCT ...), AVG(), STD(), SUM(), MAX(), MIN(), y GROUP\_CONCAT()).
- Soporte para LEFT OUTER JOIN y RIGHT OUTER JOIN cumpliendo estándares de sintaxis SQL y ODBC.
- Soporte para alias en tablas y columnas como lo requiere el estándar SQL.
- Comandos para obtener información acerca de la base de datos, el motor de base de datos, tablas e índices, así como comandos para determinar cómo el optimizador resuelve una consulta.
- Los nombres de funciones no colisionan con los nombres de tabla o columna. La única restricción es que para una llamada a una función, no se permiten espacios entre el nombre de función y el '(' a continuación.
- Proporciona soporte a grandes bases de datos.
- Se permiten hasta 64 índices por tabla. Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas.
- MySQL server tiene soporte para comandos SQL para checar, optimizar, y reparar tablas. Estos comandos están disponibles a través de la línea de comandos y el cliente MySQLcheck. MySQL también incluye myisamchk, una utilidad de línea de comandos muy rápida para efectuar estas operaciones en tablas MyISAM.

### 2.4.2. Ventajas de MySQL.

MySQL ofrece varias ventajas respecto a otros sistemas gestores de base de datos:

- Tiene licencia pública, permitiendo no solo la utilización del programa sino también la consulta y modificación de su código fuente. Resulta por tanto fácil de personalizar y adaptar a las necesidades concretas.



- El programa está desarrollado en C y C++, lo que facilita su integración en otras aplicaciones desarrolladas igualmente en esos lenguajes.
- Puede ser descargado gratuitamente de Internet (<http://www.MySQL.com>) haciendo uso de su licencia GLP (GENERAL PUBLIC LICENSE).
- Aunque MySQL es software libre, MySQL AB distribuye una versión comercial llamada "MySQL Enterprise Server", que se diferencia de la versión libre en el soporte técnico que ofrece, en la posibilidad de integrar este gestor en un software propietario y en algunas herramientas de interfaz gráfica de usuario.
- MySQL utiliza el lenguaje **SQL (Lenguaje de Consulta Estructurado)** que es el lenguaje de consulta más usado y estandarizado para acceder a bases de datos relacionales. Soporta la sintaxis estándar del lenguaje SQL para la realización de consultas de manipulación, creación y de selección de datos.
- Es un sistema cliente/servidor, permitiendo trabajar como servidor multiusuario y de subprocesamiento múltiple, es decir, cada vez que se establece una conexión con el servidor, el programa servidor crea un subproceso para manejar la solicitud del cliente, controlando el acceso simultáneo de un gran número de usuarios a los datos y asegurando el acceso solo a usuarios autorizados.
- MySQL dispone de un sistema sencillo de ayuda en línea, y de un monitor que permite realizar todas las operaciones desde la línea de comandos del sistema, sin necesitar ningún tipo de interfaz de usuario gráfica. Esto facilita la administración remota del sistema utilizando una herramienta como telnet.
- Es portable, es decir, puede ser llevado a cualquier plataforma informática. MySQL está disponible en más de veinte plataformas diferentes incluyendo las distribuciones más usadas de Linux, sistema operativo MacOS, Unix y Microsoft Windows.
- Es posible encontrar gran cantidad de software desarrollado sobre MySQL o que lo soporte. En concreto son de destacar diferentes aplicaciones *open source* para la administración de las bases de datos a través de un servidor web.
- Es estable, lo que significa que no es propenso a estropearse incluso con mucha carga. El código MySQL ha funcionado en proyectos desde mediados de 1996 sin ningún problema. Cuando el software de base de datos MySQL fue





distribuido entre un público más amplio, los nuevos usuarios rápidamente encontraron trozos de código no probados. Cada nueva versión desde entonces ha tenido pocos problemas de portabilidad incluso considerando que cada nueva versión ha tenido muchas nuevas funcionalidades.

- Es ligero, lo que significa que no requiere muchos recursos para instalarse o bien ejecutarse.
- Gestión de usuarios, manteniendo un muy buen nivel de seguridad en los datos. Las contraseñas son seguras porque todo el tráfico de contraseña está cifrado cuando se conecta con un servidor.

Todas estas características han hecho de MySQL uno de los sistemas gestores de bases de datos más utilizados en la actualidad, no solo por pequeñas empresas sino también por algunas grandes corporaciones, como pueden ser: Yahoo! Finance, Google, CISCO, MP3.com, Motorola, NASA, Silicon Graphics, Texas Instruments, A mediados del 2004 se estimaba que existían más de 5 millones de instalaciones activas del programa.

### **2.4.3. Desventajas de MySQL.**

- Una de las desventajas de MySQL es que es gratis para aplicaciones de código abierto, de lo contrario hay que pagar licencia comercial.
- La característica "select into table" no está implementada.
- El soporte para transacciones no es por defecto (se ha de activar un modo especial).
- Integridad referencial: aunque sí se admite la declaración de llaves foráneas en la creación de tablas, internamente no las trata de forma diferente al resto de campos.
- Mantenimiento y soporte: Debido a que MySQL es "open-source", no es posible contratar algún tipo de mantenimiento o soporte. Si se da el caso de que el manejador de base de datos falle, el usuario deberá resolver el problema sin poder



## MARCO TEÓRICO

---

contar con ayuda técnica de los desarrolladores de software. Como se comento con anterioridad esta desventaja se elimina si se contrata la licencia comercial.

Dar un juicio absoluto acerca de si un motor de base de datos es mejor que otro no es viable. Realmente, cada herramienta tiene fortalezas y debilidades que deben analizarse en el momento de decidir que RDBMS se va a escoger, partiendo, eso sí, del criterio de potencializar las fortalezas que puede tener cada motor de acuerdo al tipo de proyecto que se esté desarrollando; verificando, al mismo tiempo, que las posibles debilidades no sean elementos críticos en la aplicación que se va a desarrollar.

Desarrollar un estudio de este tipo a través de software libre trae ventajas no solo académicas, sino económicas, ya que este tipo de software está a disposición del público en general.

### 2.5. CONCEPTOS DE REDES

Una red de computadoras, es un sistema en el que se conectan entre sí varios equipos informáticos unidos entre sí por medio de dispositivos físicos que envían y reciben impulsos eléctricos, ondas electromagnéticas o cualquier otro medio para el transporte de datos con la finalidad de compartir información y periféricos.

El objetivo principal para la creación de una red de computadoras es compartir recursos y la información en la distancia, además de asegurar la confiabilidad, disponibilidad de la información, así como el aumento de la velocidad de la transmisión de los datos y la reducción del costo general de estas acciones al hacerlo de una forma eficiente es lo que le da a las redes su potencia y atractivo.



## MARCO TEÓRICO

---

Las redes de equipos se clasifican principalmente en:

Red de área local (**LAN**, Local Area Network), se caracteriza por ser una red que se limita a un área relativamente pequeña.

Red de área amplia (**WAN**, Wide Area Network), son redes informáticas que se extienden sobre un área geográfica extensa, no tiene limitaciones geográficas.

Esta se crea a través de la conexión de dos o más LAN físicamente aisladas y usa los servicios de conexión que proporciona la compañía telefónica. Los diversos servicios para conectar las LAN, soportan varias velocidades de comunicación. Se puede considerar Internet como la más grande WAN.

### 2.5.1. Protocolos de redes

Actualmente el modo de funcionamiento y la estructura de las redes informáticas están determinados en diferentes protocolos, estándares y modelos, los más importantes son el modelo **OSI** y el **TCP/IP**, cada uno de ellos estructura el funcionamiento de una red a través de capas de una manera diferente.

#### 2.5.1.1. TCP/IP

Este modelo es implementado a nivel mundial, se compone de 4 capas a diferencia del OSI, pero aun así este modelo combina las funciones existentes en las 7 capas del modelo OSI. Su nombre se debe a los dos principales protocolos que lo conforman: **TCP (Protocolo de Control de la Transmisión)** y el protocolo **IP (Protocolo Internet)**.  
Vea la **Figura 2.5.1.1.1**.

# TCP/IP

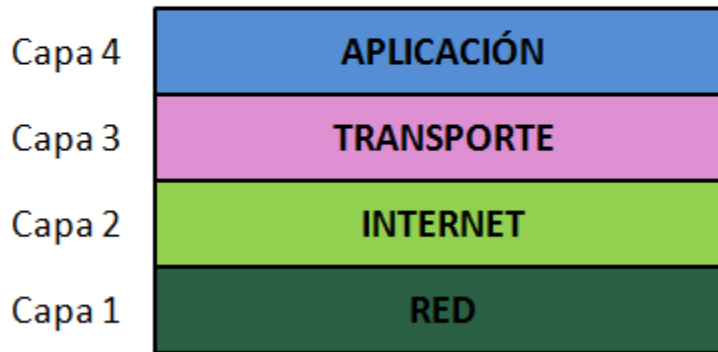


Figura 2.5.1.1.1. El modelo TCP/IP.

## 2.5.1.2. Modelo OSI

Debido a la complejidad de que diferentes dispositivos de conexión de red se integren y operen adecuadamente, se diseñó un modelo de siete capas diferentes llamado OSI (Modelo de interconexión de sistemas abiertos) para la elaboración de estándares de dispositivos de computación de redes.

Las primeras dos capas (la física y la de enlace de datos) se refieren al hardware de una LAN como los adaptadores de red y los cables Ethernet, las capas 3, 4 y 5 comprenden a los protocolos de comunicación, y las últimas dos capas son el **NOS (Sistema Operativo de red)**, que proporciona servicios y funciones de red al software de aplicación.

Para entender mejor acerca de las capas del modelo OSI, se da una breve explicación de cada una. Vea la **Figura 2.5.1.2.1.**

# Modelo OSI

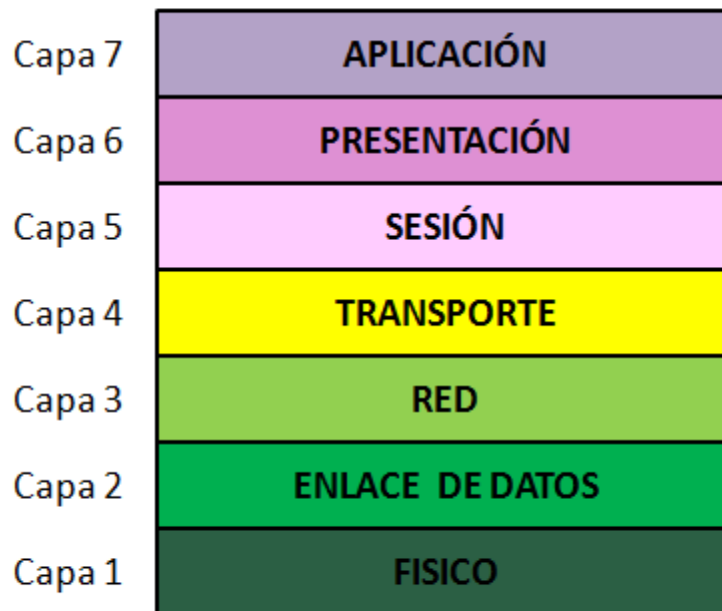


Figura 2.5.1.2.1. El modelo OSI.

## Capa física

Esta capa define la interfaz con el medio físico, incluyendo el cable de red. Se refiere a elementos como la intensidad de la señal, los voltajes indicados para la señal, la distancia de los cables y también a los tipos y especificaciones de cables.

## Capa de enlace de datos

Esta capa define el protocolo que detecta y corrige errores cuando se transmiten datos por el cable de red, a través de ella fluyen los datos de la red. Esta capa se encarga de



## MARCO TEÓRICO

---

reenviar los paquetes de información que han sido recibidos incorrectamente. Esta capa está dividida en dos subcapas: **MAC (Control de acceso al medio)** y **LLC (Control de enlace lógico)**.

### Capa de red

La capa de red define el modo en que se dirigen los datos de un nodo de red al próximo.

### Capa de transporte

La capa de transporte mantiene y proporciona el enlace de comunicaciones, es la encargada de responder si falla o se dificulta el establecimiento del enlace.

### Capa de sesión

Esta capa controla las conexiones de red entre nodos y es responsable de la creación, mantenimiento y terminación de la sesiones de red.

### Capa de presentación

La capa de transporte es la responsable de traducir los datos entre formatos específicos para asegurarse de que los datos puedan ser recibidos en un formato legible para el dispositivo al que se presenta.



---

### Capa de aplicación

Esta capa proporciona funciones a las aplicaciones de usuario y el administrador de red y al sistema operativo le ofrece servicios como la transferencia de archivos.

#### 2.5.2. Topología de redes

Se refiere principalmente a la organización física de los cables y otros componentes de la red, en otras palabras son los esquemas para la conexión de los equipos. Las cuatro topologías básicas son: bus, estrella, anillo y malla, de ahí parten los demás diseños.

##### 2.5.2.1. Características y ventajas de las topologías

###### Bus

- Dispositivos conectados a un cable común compartido.
- Es un método simple y común utilizado en las redes de equipos.
- Consta de un único cable llamado segmento central.
- En cada momento, solo puede haber un equipo enviando mensajes.
- No son responsables de pasar datos de un equipo al siguiente, si falla un equipo, no afecta al resto de la red.
- Es fácil de ampliar el bus.
- El uso del cable es económico.
- La red pierde rendimiento cuando el tráfico es muy fuerte.
- Si el cable se rompe puede afectar a muchos usuarios.
- Dificultad para aislar los componentes defectuosos conectados al bus, como resultado de la ausencia de puntos de concentración.

Vea la **Figura 2.5.2.1.1.**

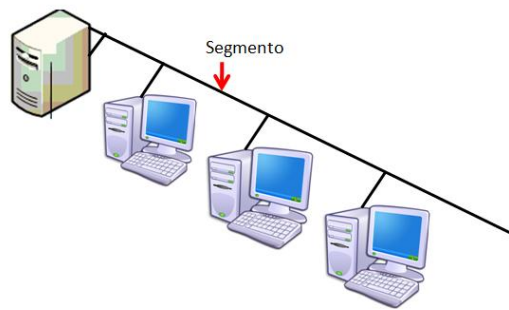


Figura 2.5.2.1.1. Red con topología en bus.

## Estrella

- Los segmentos de cable de cada equipo están conectados a un componente centralizado llamado hub.
- Centraliza los recursos y la gestión.
- Requiere una gran cantidad de cables en una instalación de red.
- Cuando el punto central falla, cae toda la red.
- Al utilizar esta topología es fácil modificar un sistema e incorporar nuevos equipos.
- El fallo de un equipo no afecta al resto de la red.

Vea la **Figura 2.5.2.1.2.**

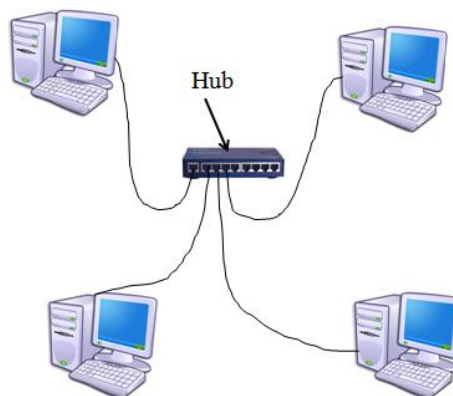


Figura 2.5.2.1.2. Red en estrella

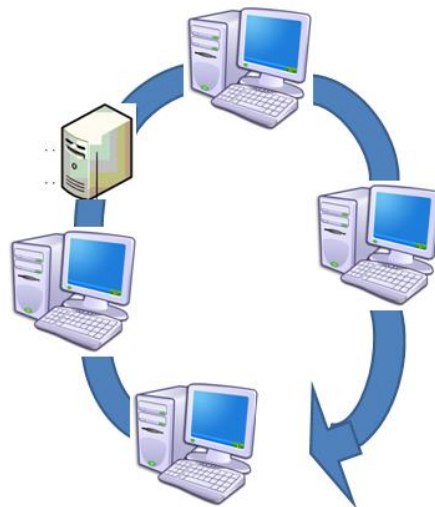




### Anillo

- Conecta equipos en un único círculo de cable.
- La señal viaja a través del bucle en una dirección, pasa a través de cada equipo que puede funcionar como repetidor para amplificarla y enviarla al equipo siguiente.
- Si falla un equipo puede tener impacto en toda la red.
- Ofrece un acceso equitativo a todos los equipos.
- El rendimiento se mantiene aunque haya muchos usuarios.
- La reconfiguración de la red interrumpe su funcionamiento.

Vea la **Figura 2.5.2.1.3.**



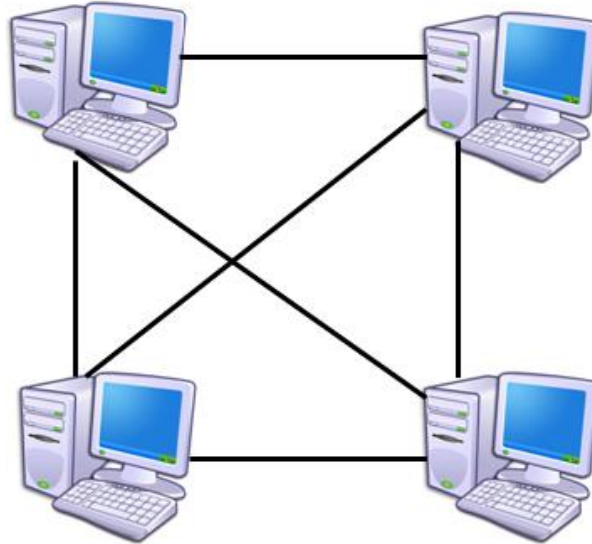
**Figura 2.5.2.1.3. Red en anillo.**

### Malla

- Ofrece una redundancia y fiabilidad superiores.
- Cada equipo está conectado a todos los demás equipos mediante cables separados.
- Resultan ser caras de instalar, debido a que utilizan mucho cableado.

- Facilidad para resolver problemas.

Vea la **Figura 2.5.2.1.4.**



**Figura 2.5.2.1.4.** Red con topología en malla.

### **Doble anillo**

- Permite que los datos se envíen en ambas direcciones.
- Los anillos no están conectados entre sí.
- Crea redundancia, si alguno de los anillos falla los datos se pueden transmitir por el otro.
- Es costoso debido a que se duplica la infraestructura que se requiere.

Vea la **Figura 2.5.2.1.5.**

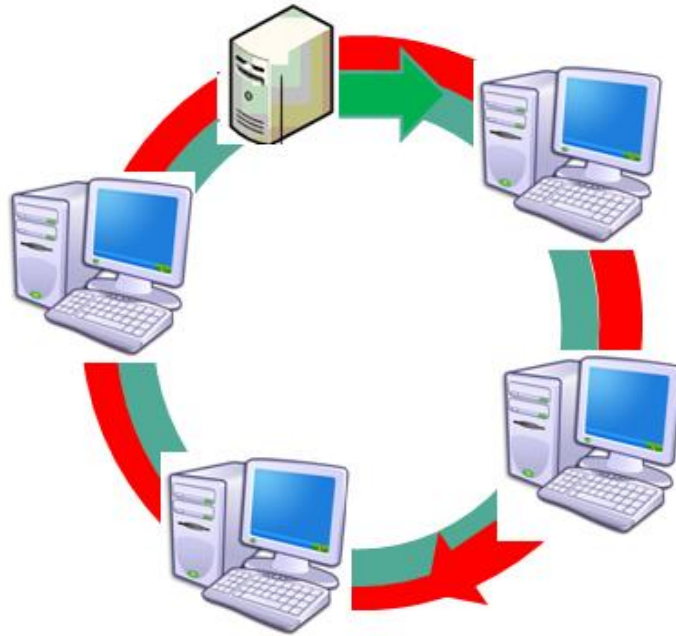


Figura 2.5.2.1.5. Red con topología en doble anillo.

### Estrella-bus

- Varias topologías en estrella están conectadas entre si con segmentos de bus lineales.
- Si un equipo cae, esto no afecta a toda la red, los demás equipos pueden seguir comunicándose.
- Depende de un hub, si deja de funcionar todos los equipos conectados a él dejaran de funcionar.

### Estrella-anillo

- Están centradas en un hub que contiene el anillo.
- Están conectados en forma de estrella al hub principal.



### 2.5.3. Dispositivos de conexión en redes

Existen diferentes dispositivos con el propósito de interconectar redes individuales, ampliar una red aislada, compartir o unir equipos informáticos y los recursos que contienen.

Estos periféricos son hubs, repetidores, bridges, switches, routers y gateways. A continuación se hace una explicación breve sobre cada uno de ellos.

Vea la **Figura 2.5.3**. Donde se ubican los dispositivos de conexión en redes.

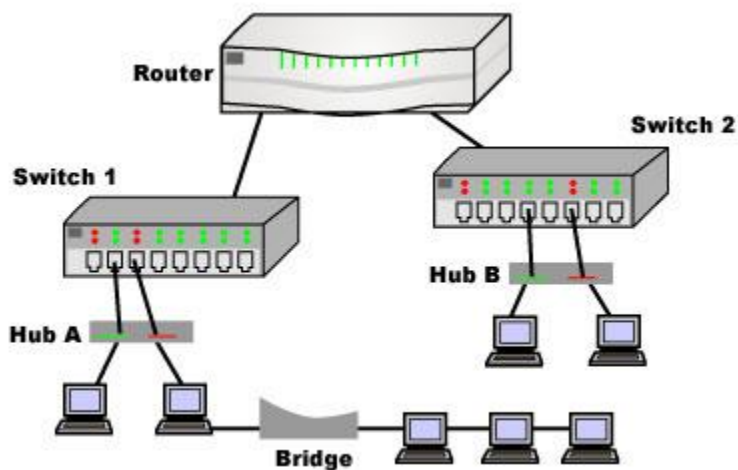


Figura 2.5.3. Dispositivos de conexión en redes.

#### 2.5.3.1. Hubs

Los hubs o concentradores, son dispositivos que son un punto central de conexión para nodos de red que están dispuestos de una topología física de estrella. Estos a su vez se encuentran separados físicamente de algún nodo de la red, aunque existen casos en los que algunos hubs se enchufan a un puerto de expansión en un nodo de red.



En un hub se pueden encontrar varios puertos en la parte trasera de la tarjeta, que le permite conectar el cable de otros nodos de red.

### **2.5.3.2. Repetidores**

Los repetidores son dispositivos de la capa física, que logran extender la longitud de la red, amplificar y retransmitir la señal de la red. Su principal objetivo es extender la longitud máxima permitida del cable de red. Este tipo de dispositivos hacen posibles que una serie de segmentos de cable logren comportarse como si fuera solo un cable.

Con los repetidores multipuertos se pueden conectar más de dos segmentos de cable de red, este tipo de repetidor logra crear una topología física de estrella.

### **2.5.3.3. Bridges**

Los puentes o bridges son dispositivos que conectan dos LAN separadas para crear lo que aparenta ser solo una LAN, también revisan la dirección asociada con cada paquete de información, si es la dirección correspondiente al otro segmento, el bridge podrá pasar el paquete al segmento de red. Si al revisar reconoce que esta dirección es la correspondiente a un nodo del segmento de red actual, procederá a no pasar el paquete al otro lado.

El objetivo principal de un bridge es lograr transmitir la información enviada por un nodo de una red al correspondiente segmento de otra red.

Los bridges también son utilizados para reducir la cantidad de tráfico de red de un segmento de red, esto se logra al dividir un segmento de red en dos y así poder conectarlos por medio de un bridge. Los bridges son un poco menos rápidos debido a que conmutan en el software e interconectan LANs con diferentes anchos de banda.



## MARCO TEÓRICO

---

Este tipo de dispositivos se pueden encontrar en diferentes formas y tamaños, en la mayoría de los casos, son dispositivos muy parecidos a una computadora con conectores a los que se conectan redes separadas, y otro tipo de bridges son computadoras con un adaptador para cada red que va a conectarse.

### **2.5.3.4. Switches**

Los conmutadores o switches son dispositivos que funcionan en la capa de enlace de datos del modelo OSI, su función es muy parecida como operan los bridges, pasan datos de un segmento a otro de acuerdo con la dirección de destino.

A diferencia de los bridges estos dispositivos son más rápidos debido a que conmutan en el hardware, mientras que los bridges lo hacen en el software. Soportan una densidad mayor de puertos, por eso se le considera un puente multipuertos. Funcionan como un filtro en la red, produciendo mayor rendimiento y seguridad en las redes de área local.

Disminuyen colisiones en los segmentos de red esto se debe porque ellos ofrecen un ancho de banda dedicado exclusivamente a cada segmento de red.

Algunos switches soportan la conmutación rápida, esta reduce la latencia y los retardos de la red. La latencia es la suma de retardos temporales de una red, los retardos son producidos por la transmisión y propagación de paquetes dentro de la red.

### **2.5.3.5. Routers**

Los routers son dispositivos que funcionan en la capa tres del modelo OSI, son utilizados para asegurar el direccionamiento de paquetes entre las redes informáticas y determinan la ruta más eficiente, cuando existe más de una ruta.



### 2.5.3.6. Gateways

Las puertas de enlace o gateways son dispositivos que permiten que los nodos de una red se comuniquen con diferentes protocolos y arquitecturas a todos los niveles de comunicación. Su función principal es traducir la información del protocolo empleado en una red al protocolo utilizado en la red de destino.