

TECNICATURA  
UNIVERSITARIA  
EN PROGRAMACIÓN  
UTN-FRC



Facultad Regional Córdoba

# TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN

## LABORATORIO DE COMPUTACIÓN

Nivelación

Material Teórico



## ÍNDICE

1. DATO, INFORMACIÓN Y CONOCIMIENTO .....	2
2. SISTEMAS DE INFORMACIÓN .....	4
3. CONCEPTO DE BASE DE DATOS .....	7
4. Sistema de Base de Datos .....	11
5. Componentes de un Sistema de Base de Datos .....	11
6. Niveles de abstracción de datos .....	14
7. Modelo de Datos .....	16
8. Modelo Relacional .....	17
9. Campos Claves .....	21
Clave Principal .....	21
Clave Externa .....	21
10. Tipos de relaciones .....	22
Relación Uno a Uno .....	22
Relación Uno a Varios .....	22
Relación Varios a Varios .....	22
11. Pautas para el diseño de una base de datos .....	23
12. Diferentes Tipos de Tablas .....	25
13. Normalización de datos .....	26
14. Grados de Normalización .....	27
15. Integridad de datos .....	28
16. Bibliografía .....	32

## 1. DATO, INFORMACIÓN Y CONOCIMIENTO

Como habrán podido observar en la presentación de la asignatura, las Bases de Datos son el hilo conductor del temario de la misma, y no podemos trabajar con algo que no conocemos, comenzaremos por las definiciones fundamentales que hacen al tema, es decir vamos a definir y distinguir los conceptos de **dato**, **información** y **conocimiento**.

¿Qué son **Datos**? Son hechos aislados y en bruto, los cuales situados en un contexto significativo y mediante una o varias operaciones de procesamiento, permiten obtener resultados que son objeto de dicho procesamiento. La finalidad básica al recopilar y procesar datos es reproducir información.

Una empresa recibe una cantidad ilimitada de datos provenientes de fuente internas y externas, los que, si fueran procesados sin una finalidad informativa su abundancia sería excesiva. En consecuencia, solo podrá obtenerse la cantidad apropiada de datos mediante un sistema adecuado de información.

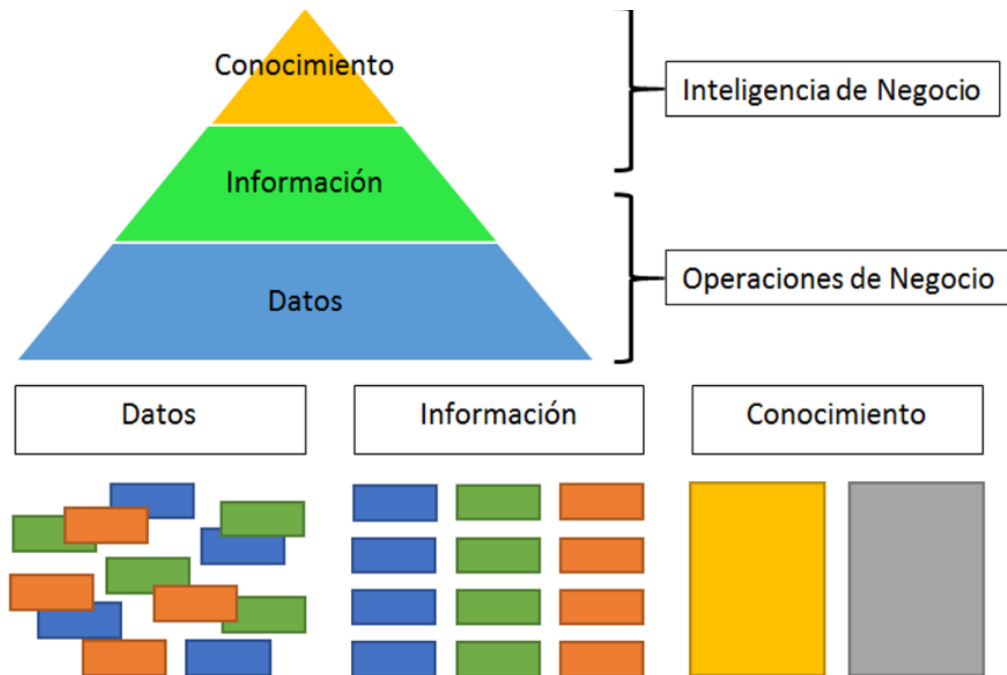
Podemos deducir entonces que **Información** es el conjunto de datos organizados, procesados y contextualizados que se transforman en un mensaje con un sentido para un receptor.

La información es un acontecimiento o una serie de ellos que llevan un mensaje y que, al ser percibido por el receptor mediante alguno de sus sentidos, amplía sus conocimientos. Solo el destinatario puede evaluar la significación y la utilidad de la información recibida.

La función primordial de la información, y por lo tanto, la de un Sistema de Información, consiste en aumentar el **conocimiento** del usuario, o en reducir su incertidumbre, para la toma de decisiones dentro de una organización.

Por su parte el **conocimiento** es un proceso complejo que dependen de múltiples factores ya que en el mismo se produce una mezcla de experiencias y valores que permiten interpretar la información y hacen al saber, es decir permiten analizar cómo sirve la información y así transformarla, siendo muy útil para la acción.

Es importante destacar que si bien se origina y aplica en la mente de los conocedores. En las organizaciones con frecuencia no sólo se encuentra dentro de documentos o almacenes de datos, sino que también se traduce a rutinas organizativas, procesos, prácticas, y normas (Carrión, 2017).



**Gráfico 1:** Elaboración propia

El conocimiento se deriva de la información, así como la información se deriva de los datos. Para que la información se convierta en conocimiento es necesario que el experto o conocedor realice procesos complejos como la comparación de esta información con otro conocimiento, deducción o inducción de la misma, análisis de variables, evaluación de distintos escenarios, predicción de las consecuencias, diálogo o debate con otros expertos o conocedores afines al área, entre otros.

De forma general, los datos son la materia prima bruta y es en el momento en el que el usuario les atribuye algún significado especial que pasan a convertirse en información. Cuando los especialistas elaboran o encuentran un modelo, haciendo que la interpretación que surge entre la información y ese modelo represente un valor agregado, entonces se transforman al conocimiento.

Veamos unos ejemplos para poder entender mejor la diferencia entre estos conceptos:

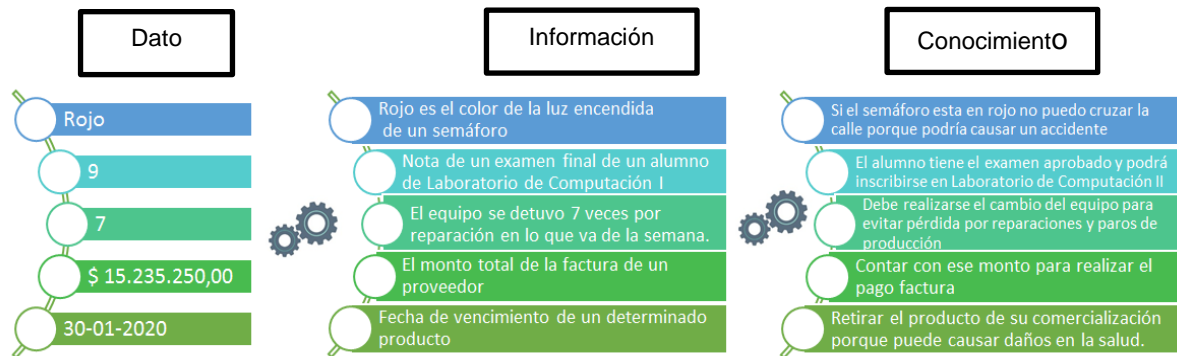


Gráfico 2: Elaboración propia

El objeto de esta asignatura en el ciclo de nivelación y posteriormente en el trayecto pedagógico de la carrera, tiene como finalidad que los estudiantes puedan comprender cómo se almacenan datos para su proceso y cómo se transforman en información, quedando fuera de su alcance lo relacionado con la gestión del conocimiento.

## 2. SISTEMAS DE INFORMACIÓN

Es importante dar inicio a este apartado explicando que el proceso de transformar un conjunto de datos en información es realizado por un Sistema de Información.

Ahora bien, antes de continuar es importante aclarar cuestiones que hacen a la definición de Sistemas.

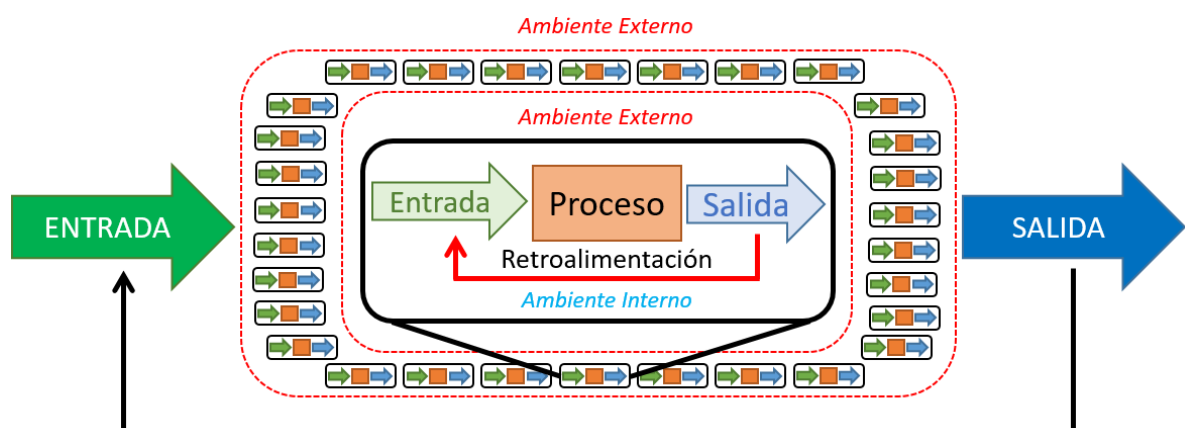


Gráfico 3: Extraído de el Blog Salmón

Un sistema es un conjunto de elementos que se interrelacionan entre sí y con su ambiente (sistema abierto) con un objetivo en común. Esta definición se desprende de la Teoría General de los Sistemas (TGS) impulsada a mediados del siglo XX en gran medida por los aportes del biólogo austriaco Ludwig von

Bertalanffi, que luego llegó a inspirar desarrollos en distintas disciplinas constituyendo el amplio campo de la sistémica o de las ciencias de los sistemas, incluyendo especialidades como la cibernética, la teoría de la información, la teoría de juegos, la teoría del caos o la teoría de las catástrofes, entre otros.

Basado en lo anterior, podemos decir que un Sistema de Información (S.I.) es un conjunto de componentes que interaccionan entre sí para alcanzar un fin determinado, el cual es satisfacer las necesidades de información de una organización para la toma de decisiones en todos sus niveles, reduciendo la incertidumbre. Estos componentes pueden ser personas, datos, actividades o recursos materiales en general, los cuales procesan la información y la distribuyen de manera adecuada.

Un Sistema de Información realiza cuatro actividades principales:

1. **Entrada de datos:** Los datos que ingresan deben ser pertinentes para la organización y para el objeto del S.I. Es muy importante poder identificar cuáles son los estrictamente necesarios. De esta manera se previenen errores alusivos a la excesiva abundancia, o a la escases de los mismos. Estos errores deben ser minimizados dado que genera altos costos, pérdida de tiempo e incertidumbre.
2. **Almacenamiento de datos:** Es necesario contar siempre con S.I. que permitan recuperar en forma consistente y confiable los datos ingresados. Los datos en este almacenamiento son datos persistentes es decir los que están guardados en un almacenamiento no volátil, en concreto guardados en una **Base de Datos**.
3. **Procesamiento:** Se lleva a cabo cuando se genera la información tanto de los datos convenientemente seleccionados y almacenados en los dos procesos anteriores.
4. **Salida de información:** Es el objetivo principal del sistema. Para que esta salida del sistema sea de utilidad a los administradores y personas involucradas en la toma de decisiones, la información debe tener determinadas características las que se muestran en la siguiente tabla elaborada por Stair & Reynolds, 2015:

**Salida de información**

<b>Accesible</b>	Que los usuarios puedan acceder a ella de una manera fácil
<b>Exacta</b>	Libre de errores
<b>Completa</b>	Que contenga todos los hechos relevantes
<b>Económica</b>	El costo debe ser accesible
<b>Flexible</b>	Que pueda utilizarse para una gran variedad de propósitos
<b>Relevante</b>	Debe ser importante para las personas que toman las decisiones
<b>Segura</b>	Estar protegida de usuarios no autorizados
<b>Simple</b>	Sin complejidades que enturbien su significado.
<b>Oportuna</b>	En el momento en que se necesita
<b>Verificable</b>	Que pueda ser comprobable la exactitud de la misma
<b>Confiable</b>	Que el método de recolección y procesamiento sea confiable de igual modo que su fuente

**Tabla 1:** Tabla de elaboración propia, información extraída de Stair & Reynolds,2015.

Antes de avanzar, es necesario volver al Gráfico 3 (pag, 4) donde se puede observar, cómo un sistema está compuesto de otros sistemas menores es decir de subsistemas. Y que al tener la característica de sistemas abiertos, puede suceder que la salida de un sistema de información (que es información) sea la entrada (dato) de otro sistema de información.

Para esta asignatura lo importante es centrarnos en el almacenamiento de datos persistentes (aquellos que están guardados en un almacenamiento no volátil) sin, por supuesto, olvidarnos del **todo** ya que siempre se está trabajando con un sistema más amplio. Es decir, cómo se va a almacenar (o cómo están almacenados) los datos que ingresan en un S.I. de tal forma que esté disponible para el resto de los procesos que se realizan en el mismo.

Un ejemplo de Sistema de Información que se puede nombrar es el Sistema de Procesamiento de Transacciones (TPS: Transaction Processing System) que realiza y registra las transacciones rutinarias diarias necesarias para el funcionamiento de una empresa y se encuentran en el nivel más bajo en la jerarquía organizacional. Por ejemplo, el sistema que funciona en la línea de cajas de un supermercado.



Imagen 1: Ejemplo Sencillo

Otros sistemas a los cuales se puede hacer alusión son: El de Control de Proceso de Negocio (BPM: Business Process Control); el de Colaboración Empresarial (ERP: Enterprise Resource Plannig); el de Información de Gestión (MIS: Management Information System); el de Apoyo a la Toma de Decisiones (DSS: Decision Support System); el de Información Ejecutiva (EIS: Executive Information System), entre otros.

### 3. CONCEPTO DE BASE DE DATOS

Una **base de datos** es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Actualmente con el avance de la informática las bases de datos están almacenadas en medios digitales.



Imagen 2:  
Extraído de la página de  
123RF



Imagen 3:  
Extraído de EUROINNOVA

Se podría decir entonces, que una Base de Datos es una colección de datos relacionados con un tema particular, que está constituida por cierto conjunto de datos persistentes y que son utilizados por los sistemas de aplicaciones de una empresa (o cualquier organización) determinada.

Existen bases de datos **estructuradas** y no **estructuradas**, en el ciclo de Nivelación se van a estudiar solo las primeras. Es importante tener en cuenta que:

- Cada unidad de dato es indivisible de dicha base de datos y se encuentra almacenado en un **campo**.
- Un conjunto de campos referido a una misma y única persona, cosa o transacción en particular forman lo que se llama un **registro**.
- Un conjunto de registros y campos relacionados con el mismo tema (mismo conjunto de personas, cosas o transacciones) de una organización se agrupan en una misma **tabla de datos**.



### 3.a Elementos de una Base de Datos

Continuando con lo explicado en el párrafo anterior vamos a definir los elementos de una Base de datos estructurada:

**Campo** es el espacio de almacenamiento de un dato en particular, la pieza más pequeña de información de la que se compone una base de datos, que es la parte indivisible, contiene un único dato.

**Registro** es un conjunto de campos que describen un mismo objeto o entidad. Un registro es, en medios electrónicos de almacenamiento, el equivalente a una ficha. Cada registro está compuesto por los mismos campos, y en la misma disposición; sólo cambia el contenido, pero permanece invariable la longitud y la ubicación de cada uno de los datos en todos los registros. Esto siempre y cuando se esté hablando de base de datos relacionales las cuales conservan una estructura determinada.

**Tabla** es una colección de registros idénticos en cuanto a su formato, que se agrupan. La denominación de Tabla se debe a su organización en forma de filas y columnas que presentan los datos. Cada fila o renglón contiene los datos de un único registro, dispuestos uno al lado de otro, por lo que, siguiendo hacia abajo, en cada columna encontraremos un determinado campo de cada uno de los registros de la tabla.

Veamos estos conceptos con un ejemplo: si queremos guardar artículos que están a la venta en un supermercado: fideos secos tallarín Masolio de 500 gms a \$50,00.- se deberá guardar “fideos secos tallarín” en un campo al cual le podemos llamar descripción, “Masolio” en un campo llamado marca, “50,00” en un campo precio, “500” en tamaño, “gramos” en unidad de medida, y así sucesivamente. Así mismo si se quiere guardar otro artículo, por ejemplo: arvejas en lata Arcos por 250 gms a \$45,00.- cada dato se almacenaría en su campo respectivo.

Entonces para guardar los datos de cada artículo tendremos los Campos:

- Descripción
- Marca
- Tamaño
- Unidad de medida
- Precio

Un registro sería el conjunto de todos los datos de un artículo en particular.

Tabla a		Tabla b	
Campos	Registro 1	Campos	Registro 2
Descripción	Fideos secos tallarín	Descripción	Lata arvejas
Marca	Masolio	Marca	Arcos
Tamaño	500	Tamaño	250
Unidad Medida	Gramos	Unidad Medida	Gramos
Precio	\$50,00	Precio	\$45,00

Tabla 2 a - b: Elaboración propia

La colección de esos registros (unidos a otros que se pueden agregar) forma una tabla. De la cual se puede distinguir perfectamente cada fila como registro y cada columna como campo:

Descripción	Marca	Tamaño	Unidad Medida	Precio
Fideos secos tallarín	Masolio	500	Gramos	\$50,00
Lata arvejas	Arcos	250	Gramos	\$45,00
Mayonesa	Rica	250	Gramos	\$30,00
Puré de tomates	Arcos	500	Gramos	\$55,00

Gráfico 4: Elaboración propia

Los nombres de los campos aparecen al tope de la tabla, describen la información que contiene a cada uno de ellos, y son el medio de referencia del dato contenido en el campo.

A las tablas también se las llama entidades porque representan una cosa, persona o transacción y a cada campo se lo llama atributo ya que describen a esa entidad. Por lo cual dada la definición tendríamos con nuestro ejemplo la tabla o Entidad de Artículos y sus atributos (descripción, marca, tamaño, unidad medida, precio).

ARTICULOS	
	descripcion
	marca
	tamaño
	unidad medida
	precio

Gráfico 5: Elaboración propia

En un sistema de información también se pueden encontrar datos que no son almacenados en una base de datos y se denominan transitorios (para distinguirlos de los persistentes), de los cuales podemos distinguir, datos de entrada o datos de salida.



**Datos de entrada:** Refieren a la información que entra al sistema por primera vez y qué, podría dar pie a una modificación de los datos persistentes o convertirse en parte de la base de datos, pero en principio no forma parte de la misma.

Un ejemplo de dato de entrada es el **porcentaje** de aumento en los precios de los productos, el usuario lo ingresa al sistema modifica el precio del producto que es un dato en un campo. En este caso el precio es persistente pero el porcentaje de aumento no.



**Datos de salida:** Refiere a mensajes y resultados que emanan del sistema, esta información podría derivarse de los datos persistentes, pero no se le considera en sí como parte de la base de datos.

Un ejemplo de dato de salida es el **total** de una compra que un cliente realizó, el precio unitario de cada producto está almacenado en un campo de una base de datos, la cantidad comprada se guarda en otro campo pero el total es calculado por el sistema y se muestra en pantalla o se imprime en el ticket pero no se guarda en la base de datos.

Es importante destacar que la distinción entre persistentes y transitorios no es rígida y nítida, sino que depende del contexto.

Transitorios		Persistentes
Entrada	Salida	Se encuentran guardados en la base de datos
Porcentaje de descuento	Monto final	Cantidad y precio del artículo
Legajo	Edad	Fecha de nacimiento del alumno
Mes de liquidación	Total a cobrar	Datos del empleado
Fecha	Cantidad total de artículos pedidos	Datos de los pedidos de esa fecha
Mes y año	Monto total de la facturación de ese mes y año	Datos de las facturas de ese mes y año (cantidades vendidas y precios de venta)

**Tabla 3:** Elaboración propia

## 4. SISTEMA DE BASE DE DATOS

Una base de datos no se encuentra aislada almacenada en un dispositivo físico, sino que está en interrelación con un conjunto de componentes que tienen como objetivo común el almacenamiento, el mantenimiento y la disposición de los datos para que un S.I. pueda obtener información a partir de ese **Sistema de Base de Datos**.

Los componentes de un Sistema de Base de Datos son la Base de Datos, el hardware, el software y los usuarios.

## 5. COMPONENTES DE UN SISTEMA DE BASE DE DATOS

### Base de Datos

Como se mencionó con anterioridad es el conjunto de datos pertenecientes a un mismo contexto, recolectados y almacenados sistemáticamente para su posterior uso. Teniendo en cuenta que en esta instancia solo se va a trabajar con Base de Datos **estructuradas**, es importante recordar que los tres elementos que la constituyen son: Campos, Registros y Tablas.

### Hardware o Equipos

Son los componentes físicos (equipos) necesarios para el almacenamiento de los datos y su posterior procesamiento:

- *Para la Base de Datos:* Volúmenes de almacenamiento, el procesador o procesadores y la memoria principal asociada que hacen posible la ejecución de los programas del sistema de bases de datos, entre otros.
- *Para la aplicación:* Se pueden clasificar en:
  - \* *Terminales inteligentes:* como puede ser una PC
  - \* *Terminales no inteligentes:* Aquellas en las que no se puede realizar ningún tipo de procesamiento.

### Software o Programas

Entre la base de datos física y los usuarios del sistema existen distintos niveles de software, desde un sistema operativo capaz de administrar todo el hardware necesario, como por ejemplo el **Sistema Gestor de Base de Datos** (**SGBD** o DBMS por su sigla en inglés: Database Management System) y un programa que refiere a las aplicaciones creadas por un programador para que los usuarios puedan ver la base de datos de una forma más amigable.

El **Sistema Gestor de Base de Datos** maneja todas las solicitudes de acceso a la base de datos formuladas por los usuarios como la adición, eliminación y obtención de datos, entre otros. Una de las funciones generales del SGBD es distanciar a los usuarios de detalles de la base de datos a nivel equipo.

Definitivamente es uno de los componentes de software más importante de todo el sistema. Es el responsable de mantener la estructura de la base de datos, que incluye:

- El mantenimiento de las relaciones entre los datos de la base de datos.
- La garantía de que los datos estén correctamente almacenados y de que no se infrinjan las reglas que definen las relaciones entre los datos.
- La recuperación de todos los datos hasta un punto coherente en caso de errores del sistema.

Ejemplos de SGBD son:

- Microsoft SQL Server: Sistema de manejo de bases de datos del modelo relacional desarrollado por Microsoft.
- Oracle Database: Sistema de gestión de base de datos objeto-relacional desarrollado por Oracle Corporation. Se le considera uno de los sistemas de bases de datos más completo y hasta fechas recientes contaba con un gran dominio del mercado.
- Postgre SQL: Sistema de gestión de base de datos orientado a objetos. Es de código abierto y se publica bajo licencia BSD. Su desarrollo corre a cargo de una comunidad de desarrolladores denominada PGDG (PostgreSQL Global Development Group).
- MySQL: Sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de 6 millones de descargas. Está desarrollado como software libre con licenciamiento dual, ya que las empresas que deseen usarlo en productos privativos deben adquirir una licencia específica.

## Usuarios

Son todas las personas que de algún modo se relacionan con la Base de Datos. Existen tres categorías de usuarios:

**Programador de aplicaciones:** Es quién se encarga de escribir los programas de aplicación que utiliza la base de datos. Estos programas operan sobre los datos en todas las formas acostumbradas: recuperación de información ya existente, inserción de información nueva, eliminación de datos ya existentes.

**Usuario final:** Es quién interactúa con el sistema desde una terminal en línea. Un usuario final puede tener acceso a la base de datos a través de una de las aplicaciones mencionadas anteriormente.

En una empresa con un sistema de base de datos, existe una persona identificada con esta responsabilidad central sobre los datos. Ese individuo es el **administrador de datos**, y es quien conoce la información y las necesidades de la empresa y que generalmente ocupa un cargo gerencial. Su labor es decidir en

primer término cuáles datos deben almacenarse y establecer políticas para mantener y manejar esos datos.

**Administrador de la Base de Datos (DBA).** Es el técnico responsable de poner en práctica las decisiones del administrador de datos, es un profesional en procesamiento y administración de datos y un experto en el uso de un SGBD particular. Su tarea es crear la base de datos en sí y poner en vigor los controles técnicos necesarios para apoyar las políticas establecidas por el administrador de datos. También debe garantizar el funcionamiento adecuado del sistema, para ello cuenta con un grupo de programadores de sistemas y otros asistentes técnicos.

En la siguiente Gráfico se ilustra la forma en que se integran los componentes principales de un sistema de Base de Datos:

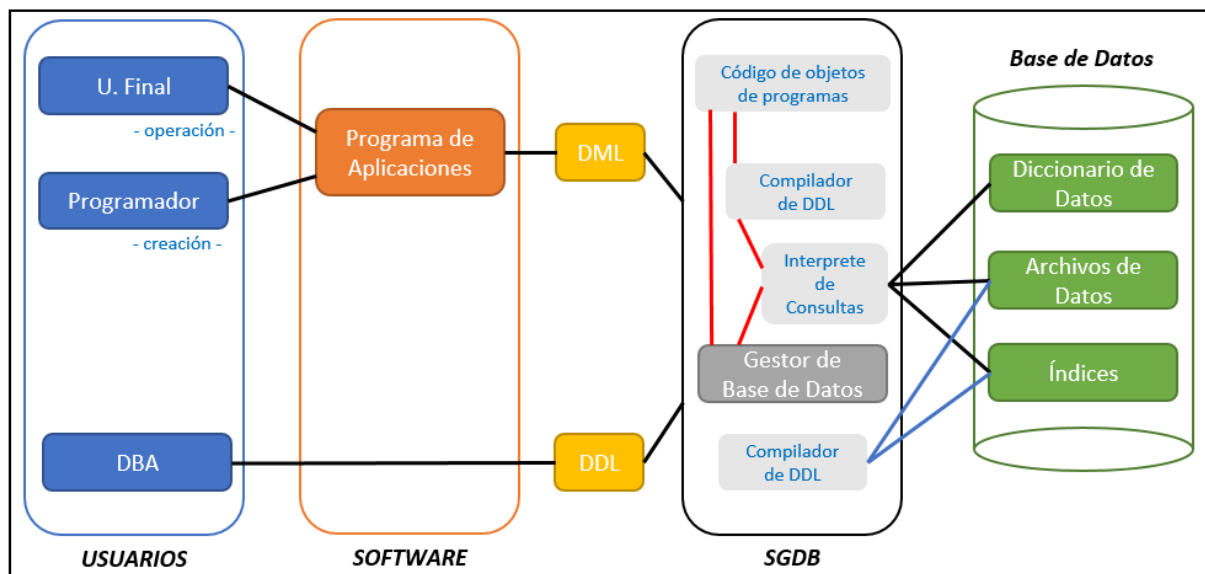


Gráfico 6: Elaboración propia

Aclaración de las abreviaciones utilizadas en el gráfico:

- **DML:** Es el lenguaje de manipulación de datos, que permite consultar datos, insertar nuevos datos, modificarlos o eliminarlos.
- **DDL:** Es el lenguaje de definición de datos. Con este lenguaje trabaja el DBA, definiendo o modificando la estructura de los almacenamientos u objetos de la base de datos.
- **Código objeto de programas:** Este código surge de la compilación de sentencias del lenguaje DML.
- **Compilador de DML:** Detecta errores de sintaxis en los comandos y participa en la generación del código objeto, a partir del programa fuente.
- **Compilador de DDL:** Interpreta y ejecuta las sentencias que el DBA le envía. Trabaja con el diccionario de datos incluyendo nuevos almacenamientos,

modificando o eliminando objetos existentes. También el DBA crea índices para el acceso a los datos.

- **Intérprete de consultas:** Traduce las consultas realizadas por un usuario, que conoce por ejemplo SQL, en órdenes de bajo nivel para que las ejecute el gestor de bases de datos.
- **Gestor de bases de datos:** Como se puede visualizar en el Gráfico, este módulo de programa del SGBD está relacionado a los siguientes elementos:  
\* código objeto, \* intérprete de consultas, \* índices y \*diccionario de datos.  
Ello se debe a que es el intermediario entre la base de datos almacenada físicamente y los módulos de programas. Es el que controla y brinda: los accesos de los usuarios, la seguridad definida por el DBA en el diccionario de datos, sin olvidarse de los accesos concurrentes de más de un usuario.
- **Archivos de datos:** Son los que almacenan los datos de las bases de datos. Junto al diccionario de datos y los índices conforman las estructuras de los datos que están en los medios de almacenamiento.
- **Diccionario de datos:** Es un conjunto de datos que hacen referencia a las estructuras de los archivos de datos, donde el DBA almacena su diseño, es decir almacena metadatos respecto a la estructura de la base de datos.
- **Índices:** Son archivos que permiten un acceso más rápido a los datos de la base de datos.

Si se presta atención al Gráfico anterior, se puede observar que entre el usuario final y la base de datos existen una serie de capas, es decir que dichos usuarios no tienen acceso directo a los datos. Uno de los propósitos principales de un sistema de bases de datos es proporcionar a los usuarios una visión abstracta de los mismos. Es decir, el sistema esconde ciertos detalles de cómo se almacenan y mantienen los datos, de ahí es que podemos encontrar los niveles de abstracción de datos.

## 6. NIVELES DE ABSTRACCIÓN DE DATOS

Para que el sistema sea útil debe recuperar los datos eficientemente. Esta preocupación ha conducido al diseño de estructuras de datos complejas para la representación de los datos en la base de datos. Como muchos usuarios de sistemas de bases de datos no están familiarizados con computadores, los desarrolladores esconden la complejidad de la estructura de datos a través de varios niveles de abstracción para simplificar la interacción de los usuarios con dichos datos y protegerlos de errores intencionados o no:



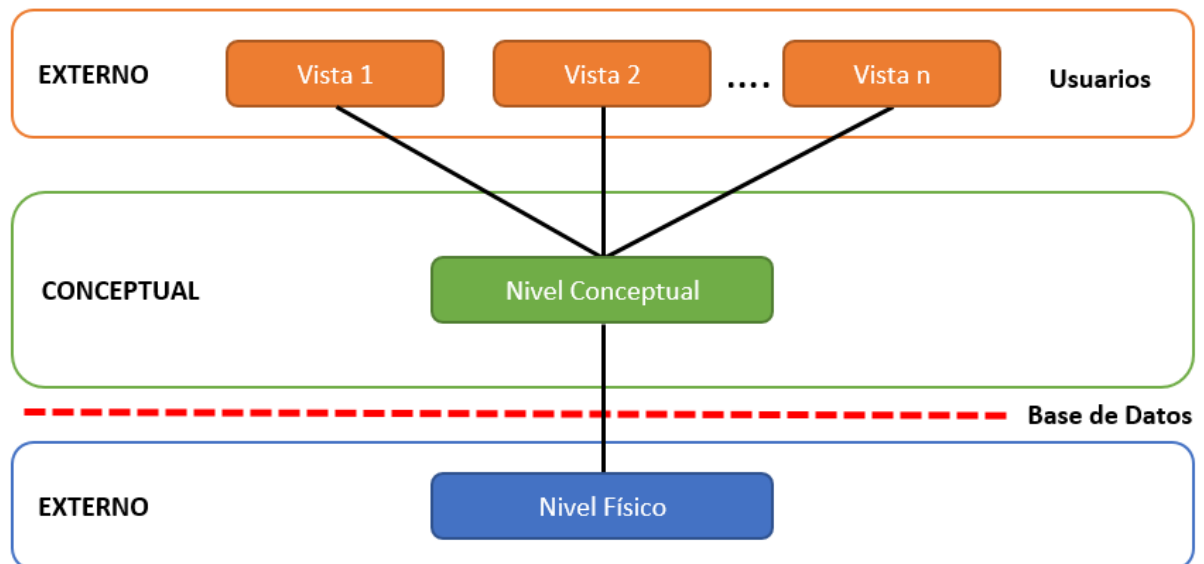


Gráfico 7: Elaboración propia

**Nivel físico:** Es el nivel más bajo de abstracción y describe cómo se almacenan realmente los datos en los dispositivos de almacenamiento secundario, por ejemplo en un hardware o disco duro.

En este nivel intervienen: el sistema operativo en su rol de gestión de almacenamiento secundario y su sistema de archivos.

En el nivel físico se describen en detalle las estructuras de datos complejas de bajo nivel.

**Nivel Conceptual o lógico:** Es un nivel más alto de abstracción y describe qué datos se almacenan en la base de datos y qué relaciones existen entre esos datos.

La base de datos completa se describe así en términos de un número pequeño de estructuras relativamente simples. Aunque la implementación de estructuras simples en el nivel lógico puede involucrar estructuras complejas del nivel físico, los usuarios del nivel lógico no necesitan preocuparse de esta complejidad.

Los administradores de bases de datos que deciden acerca de la información que se mantiene en la base de datos, usan el nivel lógico de abstracción.

**Nivel de vistas:** Es el nivel más alto de abstracción y refiere a los distintos segmentos que se pueden visibilizar de la base de datos completa.

A pesar del uso de estructuras más simples en el nivel lógico existe mayor complejidad, debido a la variedad de información almacenada en una gran base de datos. Considerando que muchos usuarios del sistema de base de datos no necesitan toda la información y solo deben acceder a una parte de la misma se simplifica el sistema mediante la abstracción del nivel de vistas. De esta manera, el sistema puede proporcionar muchas vistas para la misma base de datos.



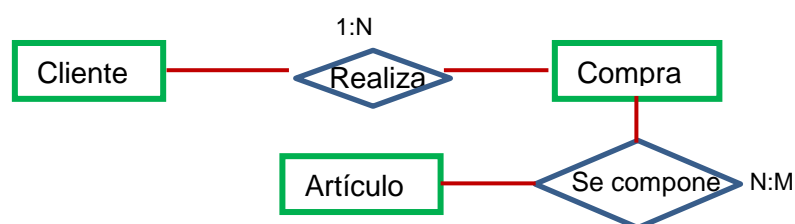
Volviendo al tema de los datos, una vez que se conoce la necesidad de la información subyacente en una organización determinada, antes de crear un almacenamiento de datos confiable se procede al modelaje de la estructura que contendrá los datos necesarios para el correcto funcionamiento del sistema de información que satisfaga dicha necesidad. Para ello es que se cuenta con el modelo de datos.

## 7. MODELO DE DATOS

El **Modelo de Datos** es una colección de herramientas conceptuales para describir los datos, las relaciones, la semántica y las restricciones de consistencia. Definir y diseñar el modelo de datos es generar una representación gráfica de la estructura que va a tener el almacenamiento de datos a un nivel lógico (ver niveles de abstracción en el apartado anterior). El trabajo de un especialista en S.I. en este punto es como el del arquitecto que dibuja un plano con todas las especificaciones y requerimientos de las necesidades de un cliente antes de la construcción de una casa.

A continuación se mencionan algunos ejemplos de Modelos de Datos:

**Modelo Entidad-Relación:** Está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamadas entidades, y de relaciones entre estos objetos. Las entidades (Tablas) son cualquier objeto, persona o transacción de la cual se necesitan guardar sus datos descriptivos o atributos representados en elipses. Estas entidades se representan en el modelo a través de un rectángulo con el nombre en su interior y las relaciones entre estas entidades se representan a través de rombos.



**Gráfico 8:** Elaboración propia

**Modelo de datos orientado a objetos:** Es una extensión del paradigma de programación orientado a objetos donde se representa el mundo real y sus problemas a través de objetos; cada uno de ellos posee un nombre, atributos y operaciones, y poseen las propiedades de polimorfismo, herencia y encapsulamiento. A diferencia de los lenguajes de programación, las bases de datos son persistentes.

**Modelos de Datos Semiestructurados:** Permite la especificación de los datos, donde los elementos de datos individuales del mismo tipo pueden tener diferentes conjuntos de atributos. Esto es diferente de los modelos de datos mencionados anteriormente, en los que cada elemento de datos de un tipo particular debe tener el mismo conjunto de atributos. El lenguaje de marcas extensible (XML, eXtensible Markup Language) se usa ampliamente para representar datos semiestructurados.

Históricamente, otros dos modelos de datos, el **Modelo de Datos de Red** y el **Modelo de Datos Jerárquico**, precedieron al modelo de datos relacional. Actualmente se usan muy poco, excepto en el código de bases de datos antiguo que aún está en servicio en algunos lugares.

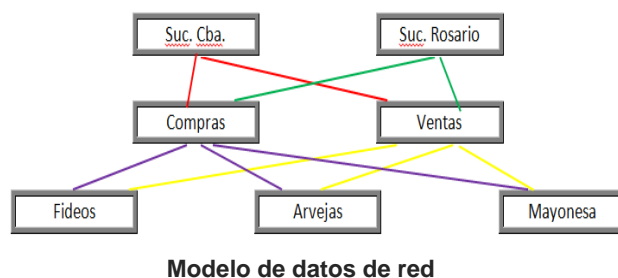


Gráfico 9: Elaboración propia

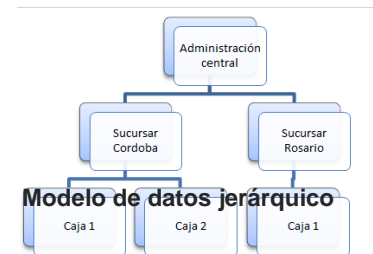


Gráfico 10: Elaboración propia

**Modelo Relacional:** Dada su trascendencia el mismo se trabaja en el apartado siguiente y es el que se utiliza para el ciclo de Nivelación de esta asignatura.

## 8. MODELO RELACIONAL

El modelo relacional es una metodología que se utiliza para modelar una realidad en particular. Esta metodología emplea una colección de herramientas conceptuales para describir datos, relaciones entre ellos, y restricciones de consistencia. Este término está referido a la forma en que se organiza la información de las distintas tablas que componen la Base de Datos.

*“Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Cada fila de la tabla representa una **relación** entre un conjunto de valores. Dado que cada tabla es un conjunto de dichas relaciones, hay una fuerte correspondencia entre el concepto de tabla y el concepto matemático de relación, del que toma su nombre el modelo de datos relacional” (García, s/d).*

El Modelo Relacional es un ejemplo de un modelo basado en registros. Los modelos basados en registros se denominan así porque la Base de Datos se

estructura en registros de formato fijo de varios tipos. Cada tabla contiene registros de un tipo particular. Cada tipo de registro define un número fijo de campos, o atributos. Las columnas de la tabla corresponden a los atributos del tipo de registro. (Ver el **Gráfico 6** anteriormente en el tema elementos de una base de datos)

Casi todos los productos de base de datos se basan en lo que se conoce como el enfoque relacional, el cual representa la tendencia dominante en el mercado actual.

Existe una amplia base teórica referida a las bases de datos relacionales y al álgebra relacional en el que está basado en modelo que escapa al alcance de lo que un Técnico Programador necesita para desarrollar su actividad profesional. Para aquellos que quieren profundizar sus conocimientos al respecto pueden utilizar el texto de Silberschatz, Korth & Sudarshan (2002) referenciado en la Bibliografía de este material.

En este apunte se abordará el tema de una manera práctica apuntando directamente al perfil profesional antes mencionado y no incursionando en las relaciones desde el punto de vista matemático. Aclarado este punto entonces, en un sistema relacional:

El usuario percibe los datos en forma de tablas relacionales, las cuales se relacionan entre sí a través de identificadores llamados clave.

Las operaciones al alcance del usuario (por ejemplo, para recuperación de datos) generan tablas nuevas a partir de las existentes. Así existe un operador para extraer un subconjunto de filas de una tabla determinada, y otro para extraer un subconjunto de las columnas (nivel de vistas, ver en niveles de abstracción desarrollado en párrafos anteriores).

Una base de datos relacional, es un sistema específicamente diseñado para el manejo de información, que ha sido previamente organizado en forma de una o varias tablas relacionadas entre sí mediante campos comunes.

Tabla: Artículos (presentada anteriormente en la página 9)

El diagrama muestra una tabla con 5 columnas: Descripción, Marca, Tamaño, Unidad Medida y Precio. Hay 4 filas de datos. A la izquierda de la tabla, el texto 'Registros' está acompañado de cuatro flechas rojas horizontales que apuntan a cada una de las cuatro filas de datos. Debajo de la tabla, el texto 'Campos' está acompañado de una línea horizontal púrpura que abarca todas las columnas, con cinco flechas púrpuras verticales que apuntan a cada una de las columnas. A la derecha de la tabla, el texto 'Nombres de Campos' está acompañado de una flecha verde horizontal que apunta a las encabezadas de las columnas.

Descripción	Marca	Tamaño	Unidad Medida	Precio
Fideos secos tallarín	Masolio	500	Gramos	\$50,00
Lata arvejas	Arcos	250	Gramos	\$45,00
Mayonesa	Rica	250	Gramos	\$30,00
Puré de tomates	Arcos	500	Gramos	\$55,00

**Gráfico 4:** Elaboración propia

En el apartado de componentes de una base de datos habíamos hablado de campos, tablas y registros, como se muestra en la figura anterior. Cuando se diseña una tabla para una base de datos en el modelo relacional se dejan de lado los registros y se grafica la tabla teniendo en cuenta su nombre, el nombre de los campos y los campos claves (de los que hablaremos más adelante). Además, es importante tener en cuenta, al momento de diseño y construcción de la base de datos en un SGBD el tipo de datos que guardará cada campo; como esto va a ser propio del software que se utilice genéricamente se va a trabajar con los siguientes tipos de datos:

- **Alfanumérico**: Implica que el campo va a guardar textos, número y/o símbolos.
- **Numérico**: Implica que se van a guardar números que luego pueden ser utilizados para cálculos matemáticos, se distingue si serán enteros o decimales: reales.
- **Fecha/hora**: Implica el guardado de fechas: día-mes-año y/o hora:minuto:segundo.
- **Booleano**: Implica el guardado de valores en un campo: verdadero o falso.

Teniendo en cuenta ello, la tabla anterior en el momento del diseño o en su nivel lógico de abstracción se ve cómo el Gráfico 5 de la página 10:

ARTICULOS	
<b>PK</b>	<b>codigo</b>
	descripcion
	marca
	tamaño
	unidad medida
	precio

**Gráfico 5:** Elaboración propia

Cuando se realiza el tratamiento de un problema de información en una organización o empresa, se empieza analizando la salida que un sistema de información debe brindar. Teniendo muy claro los detalles de esta salida, se deben diseñar las entradas necesarias para permitir el ingreso de datos (insumos de dicho sistema de información) y recién llegar a la tarea propia del programador que se pregunta: ¿Qué se va a almacenar? ¿Cómo se va a almacenar? ¿Cuál va a la

estructura del repositorio de esos datos? Estas preguntas son el punto de partida para el diseño de la base de datos.

Aquí es donde, sabiendo qué datos guardar, se debe determinar cuáles son las entidades que se distinguen de ese conjunto de datos, nos estamos refiriendo a cada persona, cosa o transacción involucrado en el almacenamiento de datos del sistema de información; cada una de esas entidades será una tabla y en cada una de ellas un campo para cada dato que describa la entidad. Así se llega a tener una cantidad de tablas separadas entre sí a las que se debe relacionar modelando la realidad de datos desde el que se partió desde el principio.

El siguiente Gráfico muestra un ejemplo de una base de datos relacional, no es necesario que se entienda en este momento todo lo referido a ella, solo es una muestra de lo que se va a llegar a hacer en parte de esta asignatura.

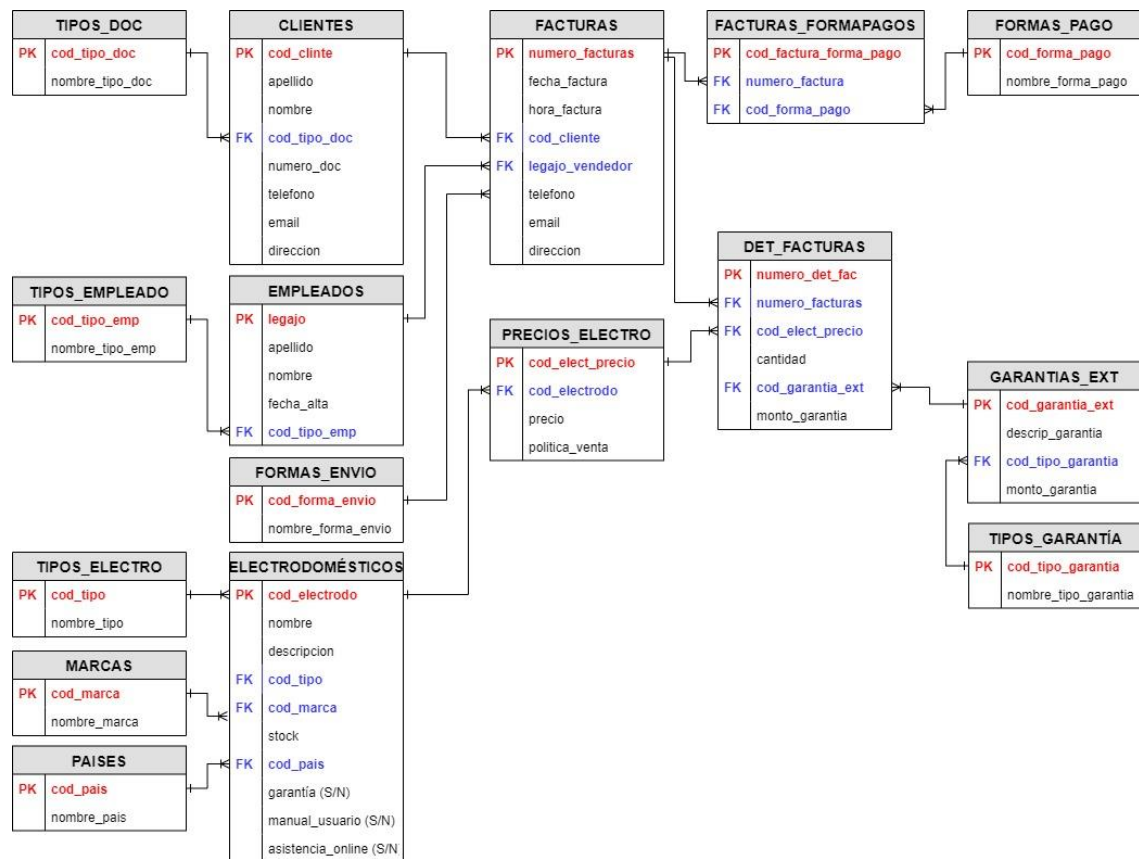


Gráfico 8: Elaboración propia

En este diagrama se pueden ver entidades que representan personas como clientes y vendedores, otras representan cosas como electrodomésticos, y otras transacciones como la factura y su detalle, se puede observar estructuras simples (tablas) que se unen entre sí a través de un campo en común. Cada tabla está referida a un único tema (clientes electrodomésticos, facturas, entre otros) y todos sus campos hacen referencia directa a ese tema y no a otro, a la vez cada campo guarda un único dato que es lógicamente indivisible.

## 9. CAMPOS CLAVES

### Clave Principal

Una Clave Principal es uno o más campos (columnas) cuyo valor o valores identifican de manera exclusiva cada registro de una tabla. Las claves principales también se denominan restricciones de clave principal, porque restringen los valores que se pueden agregar a la tabla: evitan que se agregue una fila a la tabla, cuyas columnas de clave principal son iguales a los valores correspondientes de otra fila de dicha tabla. El campo clave sirve para localizar un registro dado.

Ejemplo: Escritores, se identifica en el campo `cod_escritor` con PK.

ESCRITORES	
PK	<b>cod_escritor</b>
	nombre
	apellido
	nacionalidad
	fecha_nac
	fecha_def

Gráfico 9: Elaboración propia

### Clave Externa

Una clave externa es una correspondencia entre un conjunto de campos (columnas) de una tabla y el conjunto de campos (columnas) de la clave principal de otra tabla. Las claves externas también reciben el nombre de restricciones de clave externa, porque restringen las filas de una tabla: garantizan que las filas que se agreguen a la tabla de claves externas tengan una fila correspondiente en la tabla de claves principales.

En el ejemplo Libros, se identifica en el campo `cod_escritor` con FK. Ósea significa que existe otra tabla donde `cod_escritor` es PK (tabla Escritores).

LIBROS	
PK	<b>legajo</b>
	titulo
	genero
	editorial
	edicion
	<b>cod_escritor</b> FK

Gráfico 10: Elaboración propia

## 10. TIPOS DE RELACIONES

Entre dos tablas de una base de datos se pueden dar los siguientes tipos de relaciones:

### Relación Uno a Uno

En una relación de uno a uno, cada registro en una tabla se puede relacionar con uno o ningún registro de la segunda tabla y viceversa, en un único registro de la segunda tabla se puede relacionar con solo un registro de la primera.

Ejemplo: Un producto de la Tabla Productos sólo puede tener un único registro coincidente en la Tabla Stocks y viceversa.

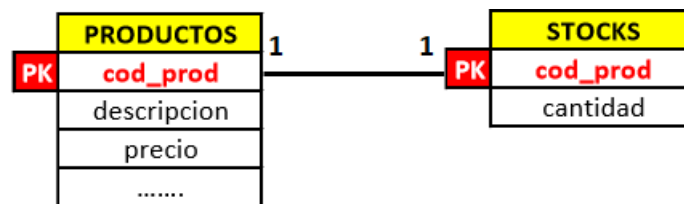


Gráfico 11 Elaboración propia

### Relación Uno a Varios

La relación uno a varios es el tipo de relación más común. En este tipo de relación, un registro de la primera tabla puede tener ninguno, uno o varios registros relacionados en la segunda tabla, pero no viceversa es decir la segunda tabla solo tendrá un registro relacionado en la primera.

Ejemplo: Cada editorial en la Tabla Editoriales podrá tener varios libros en la Tabla Libros. Pero un registro de la Tabla Libros sólo tiene un registro coincidente en la Tabla Editoriales.

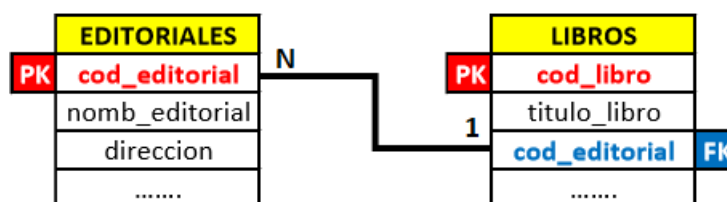


Gráfico 12: Elaboración propia

### Relación Varios a Varios

En una relación “varios a varios”, un registro de la primera tabla puede tener varios registros relacionados en la segunda tabla y un registro de la segunda tabla puede tener también varios registros relacionados en la primera.

Ejemplo: Un alumno tiene varios docentes y un docente puede tener varios alumnos.



Este tipo de relaciones no es válido en una base de datos normalizada y para ello se puede crear una tabla intermedia que al menos tenga las claves principales de ambas tablas como claves foráneas; en el ejemplo es la Tabla Cursos.

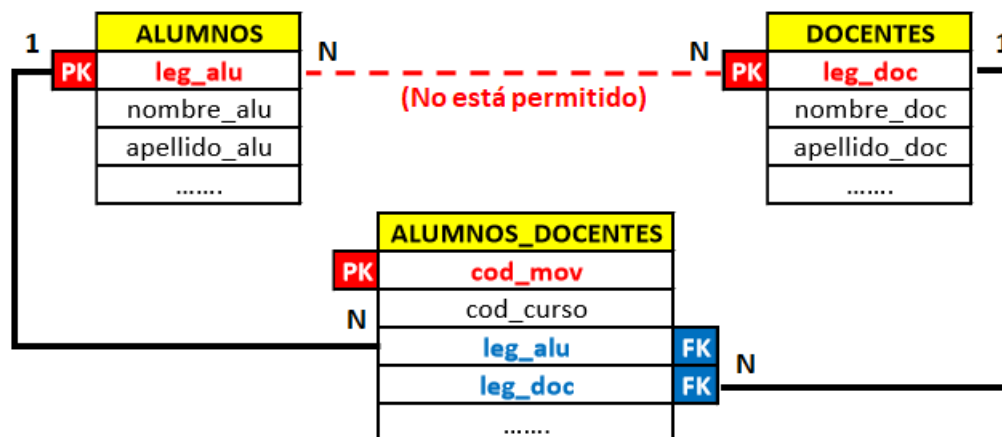


Gráfico 13: Elaboración propia

## 11. PAUTAS PARA EL DISEÑO DE UNA BASE DE DATOS

Un buen diseño de la Base de Datos es la pieza clave para que un S.I. realice las operaciones que desee de una forma efectiva, precisa y eficaz.

El diseño debe ser flexible, lógico y metódico. Algunos de los puntos importantes que definen una base de datos relacional efectiva son:

- Se adapta con facilidad para satisfacer los cambios futuros y los requerimientos de diseño.
- Sus diseños y relaciones de tablas son muy fáciles de comprender.
- Ofrece un desempeño aceptable y una utilización del disco muy adecuado.

A continuación, presentamos los pasos básicos a seguir para diseñar una base de datos:

### 1. Determinar la finalidad de la base de datos

El primer paso para diseñar una base de datos es determinar la finalidad de la Base de Datos y cómo se utiliza. Debe haberse identificado qué información desea obtener de la misma. A partir de ahí, puede determinarse sobre qué asuntos necesita almacenar hechos (las tablas) y qué hechos necesita almacenar sobre cada asunto (los campos de las tablas).

### 2. Determinar las tablas que se necesitan

Determinar las tablas puede ser el paso más complicado del proceso de diseño de la base de datos. Esto se debe a que los resultados que desea obtener de la base de datos (los informes que desea imprimir, los formularios que desea



utilizar, las preguntas para las que desea respuestas) no proporcionan pistas necesariamente acerca de la estructura de las tablas que los producen.

Lo más conveniente es realizar un boceto en papel de la base de datos y trabajar sobre el diseño primero. Al diseñar las tablas, se divide la información teniendo en cuenta los siguientes principios de diseño fundamentales:

- Una tabla no debe contener información duplicada y la información no debe duplicarse entre las tablas. Cuando cada dato está almacenado en una única tabla, se actualiza en un sólo lugar. Esto resulta más eficiente y elimina la posibilidad de que existan entradas duplicadas que contengan información diferente. Por ejemplo, el nombre y dirección del cliente se encuentra solo en la tabla clientes y no en la tabla facturas por más que posteriormente imprima esos datos en la facturación.
- Cada tabla debe contener información sobre un asunto. Cuando cada tabla contiene hechos sólo sobre un asunto, puede mantener la información acerca de cada asunto independientemente de otros asuntos. Por ejemplo, puede almacenar las direcciones de los clientes en una tabla diferente de las facturas de los clientes, de modo que pueda eliminar una factura y conservar a la vez la información sobre el cliente.

### **3. Determinar los campos que se necesitan**

Cada tabla contiene información acerca del mismo asunto, y cada campo de una tabla contiene hechos individuales sobre el asunto de la tabla. Por ejemplo, la tabla de Clientes puede incluir los campos del nombre de la empresa, la dirección, la ciudad, el país y el número de teléfono. Al realizar bocetos de los campos para cada tabla hay que tener en cuenta las siguientes sugerencias:

- Relacione cada campo con el asunto de la tabla.
- No incluya datos derivados ni calculados (datos que son el resultado de una expresión).
- Incluya toda la información que necesite.
- Almacene información en sus partes lógicas más pequeñas (por ejemplo, Nombre y Apellidos, en lugar del Nombre completo).

### **4. Identificar los campos con valores exclusivos**

Para conectar la información almacenada en tablas independientes (por ejemplo, para conectar a un cliente con todos los pedidos del cliente), cada tabla de la base de datos debe incluir un campo o un conjunto de campos que identifiquen de forma exclusiva cada registro individual de la tabla. Este campo o conjunto de campos se denomina clave principal.

Generalmente se acostumbra a identificar a cada fila o registro con un número de código o de serie, como por ejemplo número de empleado o código del

producto; a esta información de identificación la llamaremos CLAVE PRINCIPAL DE LA TABLA.

Cuando abrimos una base de datos existente, cada tabla tendrá su orden determinado inicialmente por su clave principal, y en base a la misma construiremos las relaciones entre las distintas tablas, puesto que se asocian datos de distintas tablas siempre en base a su clave principal.

### **5. Determinar las relaciones entre las tablas**

Ahora que se ha dividido la información en tablas y que se ha identificado los campos de clave principal, se necesita una forma de indicar cómo volver a reunir toda la información relacionada de un modo significativo. Para ello, debe definir relaciones entre las tablas.

### **6. Precisar el diseño.**

Una vez diseñadas las tablas, los campos y las relaciones que necesita, es el momento de estudiar el diseño y detectar los posibles fallos que pueden quedar. Es más sencillo cambiar el diseño de la base de datos ahora que una vez que haya rellenado las tablas con datos.

Es necesario utilizar SQL Server para crear las tablas, especificar relaciones entre las tablas e introducir algunos registros de datos en cada tabla. Se debe observar si se puede utilizar la base de datos para obtener las respuestas que desee.

### **7. Introducir datos y crear otros objetos de la base de datosg**

Cuando se considera que la estructura de las tablas cumple los objetivos de diseño descritos anteriormente, es el momento de comenzar a agregar los datos existentes a las tablas.

## **12. DIFERENTES TIPOS DE TABLAS**

Dentro de este conjunto de tablas relacionadas que conforman una Base de Datos, existen cuatro grupos bien diferenciados de Tablas:

Tablas Maestras o Fijas	Tablas de Movimientos o Transacciones	Tablas Auxiliares	Tablas Históricas
<ul style="list-style-type: none"> <li>• Aquellas que por su contenido se le realizan operaciones de Altas, Bajas y Modificaciones de sus atributos con muy poca frecuencia. Por ejemplo Tablas de Proveedores, Clientes, Alumnos, Profesores, etc.</li> </ul>	<ul style="list-style-type: none"> <li>• Las que por su contenido se le agregan permanentemente registros como resultado de las transacciones diarias y que se relacionan con otras tablas como las fijas y auxiliares. Por ejemplo: Facturación, Cobranzas, etc.</li> </ul>	<ul style="list-style-type: none"> <li>• Son las que por lo general contienen pocos campos y que sirven para evitar redundancia en las tablas Fijas o de Movimientos.</li> </ul>	<ul style="list-style-type: none"> <li>• Contienen backup de datos de las Bases de Datos, estos se realizan en forma periódica, según las características del sistema. Por ejemplo: todos los años se crean históricos de facturación de manera que podamos iniciar el nuevo año con la tabla de facturación vacía.</li> </ul>

Gráfico 14: Elaboración propia

### 13. NORMALIZACIÓN DE DATOS

La Normalización es el proceso mediante el cual se transforman datos complejos a un conjunto de estructuras de datos más pequeñas, que además de ser más simples y estables, son más fáciles de mantener. También se puede entender la normalización como una serie de reglas que sirven para ayudar a los diseñadores de bases de datos a desarrollar un esquema que minimice los problemas de lógica. Cada regla está basada en la que le antecede. La normalización también hace las cosas fáciles de entender. Las guías que la normalización provee crean el marco de referencia para simplificar una estructura de datos compleja.

Es decir, es un proceso de refinamiento de las estructuras de la base de datos para mejorar la velocidad a la que los datos puedan accederse, así como mejorar su integridad. La meta básica de la normalización es eliminar la información redundante de la base de datos.

La Normalización implica los siguientes procesos:

- Asegurarse de que los registros de cada tabla cuentan con un identificador único o clave principal de la tabla.
- Asegurarse de que cada campo representa una sola pieza de información. Por ejemplo, no almacene los nombres de ciudades y de los países en el mismo campo.
- Eliminar información redundante de las tablas. Cada registro de la base de datos deberá contener información única. Cada pieza de información deberá almacenarse en un solo lugar (a excepción de los campos clave).

- Eliminar los campos de repetición de grupos si existe la posibilidad de que se añaden más campos al grupo.
- Asegurarse de que en los campos no almacenen datos originados por cálculos.

Otra ventaja de la normalización de base de datos es el consumo del espacio. Una base de datos normalizada ocupa menos espacio en disco que una no normalizada. Hay menos repetición de datos, lo que tiene como consecuencia un menor uso del espacio en disco.

El proceso de normalización tiene un nombre y una serie de reglas para cada fase. Esto puede parecer un poco confuso al principio, pero poco a poco se va entendiendo el proceso, así como las razones para hacerlo de esta manera.

## 14. GRADOS DE NORMALIZACIÓN

En los años 70 Edgar F. Codd creó las bases de la Teoría de la normalización. A cada regla de la teoría la denominó forma normal. Codd creó las formas normales 1ª, 2ª y 3ª. Las formas normales proporcionan los criterios para determinar el grado de vulnerabilidad de una tabla a inconsistencias y anomalías lógicas.

•La regla establece que los campos (columnas) repetidas deben eliminarse, esto resuelve el problema de los encabezados de campo (columna) múltiples. Por ejemplo apellido y nombre o dirección. Esto ayuda a clarificar la base de datos y a organizarla en partes más pequeñas y más fáciles de entender.

**Primera  
Forma  
Normal**

**1**

•La regla establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas. Una dependencia parcial es un término que describe a aquellos datos que no dependen de la clave primaria de la tabla para identificarlos.

•Esto ayuda a clarificar la base de datos y a organizarla en partes más pequeñas y más fáciles de entender.

**Segunda Forma  
Normal**

**2**

•Una tabla está normalizada en esta forma, si todos los campos (columnas) que no son clave son funcionalmente dependientes por completo de la clave primaria y no hay dependencias transitivas. Una dependencia transitiva es aquella en la cual existen campos (columnas) que no son clave y que dependen de otros campos (columnas) que tampoco lo son.

•Cuando las tablas están en esta forma se previenen errores de lógica cuando se insertan o borran registros. Cada campo (columna) en una tabla está identificada de manera única por la clave primaria, y no debe haber datos repetidos.

**Tercera Forma  
Normal**

**3**

Gráfico 15: Elaboración propia

## 15. INTEGRIDAD DE DATOS

El término integridad de datos se refiere a la corrección y completitud de los datos en una base de datos. Cuando los contenidos de una base de datos se modifican al agregar, eliminar registros o modificar datos, la integridad de los datos almacenados puede perderse de muchas maneras diferentes.

Pueden añadirse datos no válidos a la base de datos, tales como un pedido que especifica un producto no existente.

Pueden modificarse datos existentes tomando un valor incorrecto, como por ejemplo si se reasigna un vendedor a una oficina no existente.

Los cambios en la base de datos pueden perderse debido a un error del sistema o a un fallo en el suministro de energía.

Los cambios pueden ser aplicados parcialmente, como por ejemplo si se añade un pedido de un producto sin ajustar la cantidad disponible para vender.

Una de las funciones importantes de un SGBD relacional es preservar la integridad de sus datos almacenados en la mayor medida posible. Los diferentes tipos de integridad de datos son:

### Integridad de dominio

La integridad de dominio (o columna) especifica un conjunto de valores de datos que son válidos para una columna y determina si se permiten valores nulos. La integridad de dominio se suele implementar mediante el uso del tipo de datos, el intervalo de los valores posibles permitidos en una columna, etc.

### Integridad de entidad

La integridad de entidad (o tabla) requiere que todas las filas de una tabla tengan un identificador exclusivo, conocido como clave principal.

### Integridad referencial

La integridad referencial asegura que siempre se mantienen las relaciones entre los registros de tablas relacionadas sean válidas y que no se eliminen ni modifiquen accidentalmente datos relacionados. Ayuda a garantizar que la información contenida en una tabla se corresponda con la información contenida en la tabla relacionada.

Gráfico 16: Elaboración propia

## 16. NORMALIZACIÓN DE FORMULARIOS

Si se quisiera crear una base de datos que guarde los datos de un formulario como el de la factura, se comenzará por determinar la finalidad de la base de datos, para ello se observa detenidamente la imagen del formulario.

<div style="border: 1px solid black; padding: 5px; display: inline-block; font-weight: bold; font-size: 1.2em;">C</div> <div style="border: 2px solid red; padding: 5px; display: inline-block; margin-left: 10px;">N° 0001 - 00000001</div>		<b>UNA Factura</b>							
<b>LOGO DE LA EMPRESA</b> <small>de tal persona</small>		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"><b>COMPROBANTE</b></td> <td style="width: 50%;"><b>F E C H A</b></td> </tr> <tr> <td style="text-align: center; font-weight: bold; font-size: 1.1em;">FACTURA</td> <td></td> </tr> </table>		<b>COMPROBANTE</b>	<b>F E C H A</b>	FACTURA			
<b>COMPROBANTE</b>	<b>F E C H A</b>								
FACTURA									
Domicilio Código postal - Teléfono - Provincia		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;">CUIT N°:</td> <td style="width: 50%;">XX-XXXXXXX-X</td> </tr> <tr> <td>ING. BRUTOS N°:</td> <td>XX-XXXXXXX-X</td> </tr> <tr> <td>In. Actividades:</td> <td>XX/XX/XXXX</td> </tr> </table>		CUIT N°:	XX-XXXXXXX-X	ING. BRUTOS N°:	XX-XXXXXXX-X	In. Actividades:	XX/XX/XXXX
CUIT N°:	XX-XXXXXXX-X								
ING. BRUTOS N°:	XX-XXXXXXX-X								
In. Actividades:	XX/XX/XXXX								
<b>RESPONSABLE MONOTRIBUTO</b>									
Señor(es): .....									
Domicilio: .....									
I V A	Resp. Ins. <input type="checkbox"/> Monotributo <input type="checkbox"/> Exento <input type="checkbox"/> No Respns. <input type="checkbox"/> Cons. Final <input type="checkbox"/>	CUIT N° .....							
Cond. de Venta	Contado	Cta. Cte.	Tarjeta						
		REMITO N° .....							
CANT	DETALLE	P. UNIT.	TOTAL						
2	Camas	423,50	847,00						
1	Placard	726,00	726,00						
<b>TOTAL \$</b>			<b>1573,00</b>						
<div style="display: flex; justify-content: space-between;"> <div> <b>PIE DE IMPRENTA</b>             N° 0001 - 00000051 al 000000100         </div> <div>           Fecha de Imp. Febrero 2010             Original Blanco            Duplicado Color         </div> </div>									

Se determinan las entidades que se observan, una para clientes otra para los artículos a la venta y otra para guardar la factura, para cada una crear un campo por cada atributo necesario y luego determinar las claves primarias.

CLIENTES	
PK	cod_cliente
	nombre
	direccion
	...
	...

FACTURAS	
PK	nro_factura
	fecha
	cod_cliente
	cod_articulo
	precio
	cantidad

ARTICULOS	
PK	cod_articulo
	descripcion
	precio
	...
	...

**Gráfico 20:** Elaboración propia

Establecemos las relaciones entre las tablas:



Gráfico 21: Elaboración propia

En la tabla facturas se observa lo siguiente: en una factura, se pueden vender varios artículos. Dentro de los campos de la tabla FACTURAS existe una relación **uno --> varios** entre los campos de la misma, lo que nos va a llevar a una repetición de datos (redundancia) entonces, es necesario, para este tipo de casos dos tablas: una tabla para todo lo que es uno que se le puede llamar **encabezado del formulario** (encabezados de facturas) y otra para los campos que son varios **detalles del formulario** (detalles de facturas).

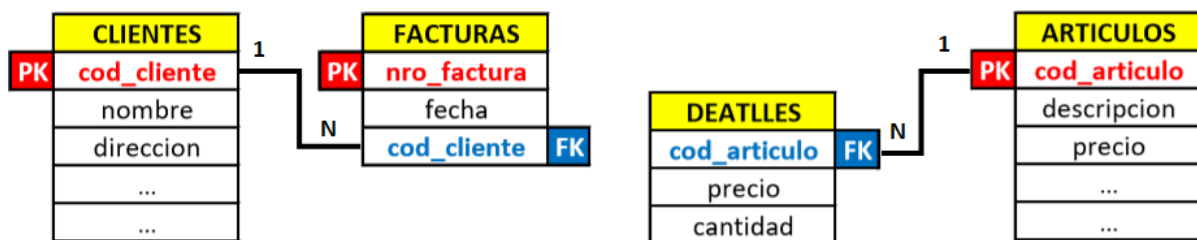


Gráfico 22: Elaboración propia

Quedan aún algunas cuestiones a resolver:

**UNO:** ¿Cómo unimos FACTURAS con DETALLES?

El análisis es el siguiente: una factura tiene varios detalles, entonces la clave primaria de FACTURAS se agrega como campo a DETALLES, como clave foránea, de tal forma que queda establecida la relación entre dichas tablas.

**DOS:** ¿Cuál es la clave primaria de DETALLES?

Aquí hay dos opciones:

- Una sería la combinación de dos campos: el campo **nro\_factura** y el campo **cod\_articulo** ya que suponemos que en una misma factura no se va a vender dos o más veces el mismo artículo dado que se tiene un campo cantidad para indicar que se venden más de un artículo en la misma factura.



- O bien, en el caso en que sea necesario tener varios registros con el mismo número de factura y el mismo código de artículo, se deberá agregar un campo entero no nulo e irrepetible, un campo autonumérico que podría llamarse por ejemplo `id_detalle`.

Solo una de estas dos opciones es válida, nunca las dos opciones.

Primero opción: una clave primaria compuesta

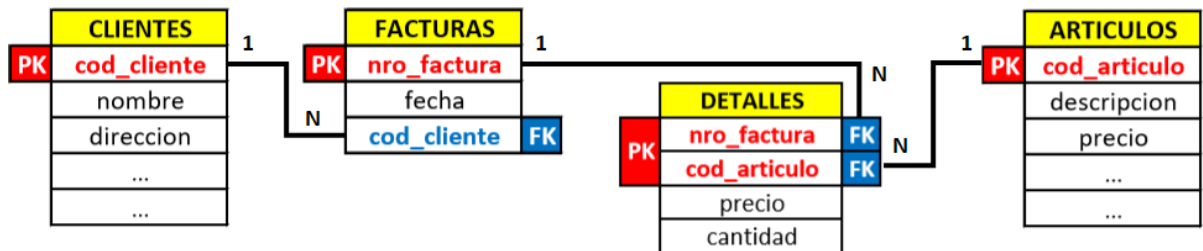


Gráfico 23: Elaboración propia

O bien, una clave primaria de un solo campo creada a tal fin:

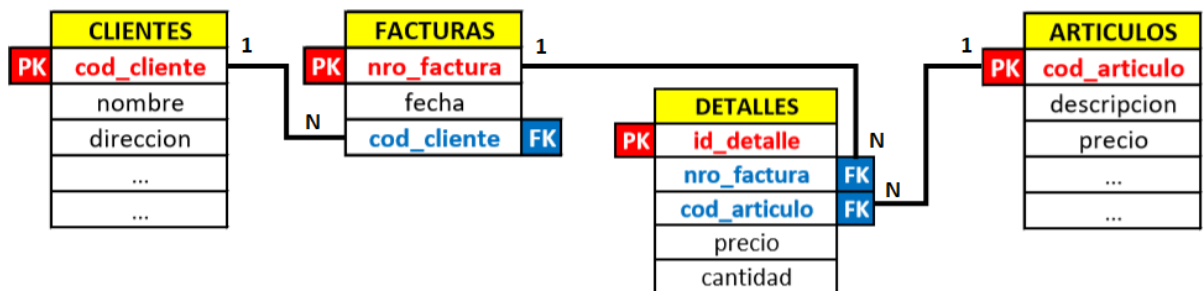


Gráfico 24: Elaboración propia



## 17. BIBLIOGRAFÍA

123RF. Base de datos Imágenes y Fotos de Stock. Consultado el 1/12/2020.  
Disponible en: [https://es.123rf.com/imagenes-de-](https://es.123rf.com/imagenes-de-archivo/base_de_datos.html?alttext=1&start=0&sti=nj6gdgrlqkxouorb64|&mediapopup=46737601)

[archivo/base\\_de\\_datos.html?alttext=1&start=0&sti=nj6gdgrlqkxouorb64|&mediapopup=46737601](https://es.123rf.com/imagenes-de-archivo/base_de_datos.html?alttext=1&start=0&sti=nj6gdgrlqkxouorb64|&mediapopup=46737601)

Carrión, J. (2017). Diferencia entre dato información y conocimiento. Recuperado de:

<http://iibi.unam.mx/voutssasmt/documentos/dato%20informacion%20conocimiento.pdf>

Ejemplo Sencillo. Ejemplo De Sistema De Procesamiento De Transacciones. Consultado el 1/12/2020. Disponible en: <https://unejemplosencillo.blogspot.com/2019/07/ejemplo-de-sistema-de-procesamiento-de.html>

El Blog Salmón. ¿Qué es la Teoría General de Sistemas? Consultado el 1/12/2020. Disponible en: <https://www.elblogsalmon.com/conceptos-de-economia/que-es-la-teoria-general-de-sistemas>

Euroinnova. Business School. Sistema Gestor de Bases de Datos. Consultado el 1/12/2020. Disponible en: <https://www.euroinnova.edu.es/cursos/sistemas-gestores-bases-datos>

García Gonzalez, V. La estructura de la base de datos relacionales. Sin datos. Disponible en:

<http://www.victorgarcia.org/pfc/relacional/estructura.php>

Silberschatz, A., Korth, H. & Sudarshan, S. (2002). Fundamentos de Bases de Datos. Madrid. Editorial Mc Graw Hill.

Stair, R. Reynolds, G. (2010). Principios de sistemas de información Un enfoque administrativo. México. Cengage Learning Editores.



### Atribución-NoComercial-SinDerivadas

Se permite descargar esta obra y compartirla, siempre y cuando no sea modificado y/o alterarse su contenido, ni se comercializarse. Referenciarlo de la siguiente manera:

Universidad Tecnológica Nacional Regional Córdoba. Material para la Tecnicatura en Programación modalidad virtual. Argentina.