

# Universidad Nacional de la Matanza

Departamento:

Ingeniería e Investigaciones Tecnológicas

Cátedra:

## Fundamentos de TIC's

JEFE DE CÁTEDRA:

Mg. Artemisa Trigueros

UNIDAD NRO: 6

**PARTE A:**

### **INTRODUCCIÓN AL SOFTWARE DE LOS SISTEMAS DE COMPUTACIÓN**

COLABORACIÓN:

**DOCENTES DE LA CÁTEDRA**

CICLO LECTIVO: **2020**

# INTRODUCCIÓN AL SOFTWARE DE LOS SISTEMAS DE COMPUTACIÓN

## Índice

<b>TEMA</b>	
1.	INTRODUCCIÓN .....3
2.	ARCHIVOS .....3
2.1.	CONCEPTOS BÁSICOS .....3
2.2.	OPERACIONES CON ARCHIVOS.....6
2.3.	TIPOS DE ARCHIVOS .....6
2.4.	ORGANIZACIÓN DE ARCHIVOS.....7
2.5.	ARCHIVOS DE TEXTO – ARCHIVOS BINARIOS .....10
3.	BASES DE DATOS .....11
3.1.	SISTEMA GESTOR DE BASE DE DATOS.....12
3.2.	BASES DE DSTOS RELACIONALES.....13
3.3.	USUARIOS DE LA BASE DE DATOS.....16
3.4.	APLICACIONES DE LAS BASES DE DATOS .....17
4.	SISTEMAS OPERATIVOS.....20
4.1.	INTRODUCCIÓN.....20
4.2.	FUNCIONES DE LOS SISTEMAS OPERATIVOS .....21
4.3.	TIPOS DE SISTEMAS OPERATIVOS.....22
4.4.	EJEMPLOS DE SISTEMAS OPERATIVOS .....23
5.	LENGUAJES DE PROGRAMACIÓN .....23
5.1.	INTRODUCCIÓN.....23
5.2.	LENGUAJE DE MÁQUINA .....23
5.3.	LENGUAJE ENSAMBLADOR O SIMBÓLICO .....24
5.4.	LENGUAJE DE ALTO NIVEL .....24
5.5.	TRADUCTORES: COMPILADORES E INTÉRPRETES .....25
6.	COMPUTACIÓN EN LA NUBE (CLOUD COMPUTING).....27

# INTRODUCCIÓN AL SOFTWARE DE LOS SISTEMAS DE COMPUTACIÓN

## 1. INTRODUCCIÓN

**El software está constituido por un conjunto de elementos intangibles llamados programas que el hardware ejecuta.**

Dentro del software se incluyen: lenguajes de programación, interfaces de usuarios, sistemas operativos, las herramientas o utilidades, etc.

Una primera clasificación nos permite distinguir tres grandes grupos:

<b>De Sistema</b>	Conjunto de programas imprescindibles para el funcionamiento del hardware, más aquellos programas que facilitan el uso del sistema y optimizan sus recursos. Forman parte de este software:
	<b>1. Sistema Operativo:</b> conjunto de programas que se encargan de gestionar todos los elementos del sistema (tanto de software como de hardware); ocultando los detalles técnicos al usuario. Se trata con más detalle mientras se avanza con esta Unidad.
	<b>2. Software de Programación.</b> Programas y utilidades que se utilizan en la construcción de aplicaciones de usuarios.
	<b>3. Software de diagnóstico y mantenimiento.</b> Utilizado para poner a prueba los equipos, encontrar fallas, etc.
<b>De Aplicación</b>	Programas que realizan una tarea específica. Se los puede agrupar en:
	<b>1. Estándar:</b> aplicaciones de uso general para ser utilizadas por gran número de usuarios, por ejemplo: planillas de cálculo, procesadores de textos, etc.
	<b>2. A medida:</b> aplicaciones específicas desarrolladas para pocos usuarios y para un fin específico. Por ejemplo: un programa solicitado por la NASA.
<b>Software en la Nube (Cloud Computing)</b>	La Computación en la Nube permite a las personas y empresas acceder a servicios de tecnología desde cualquier parte y en cualquier momento a través de su conexión a Internet.

## 2. ARCHIVOS

### 2.1. CONCEPTOS BÁSICOS

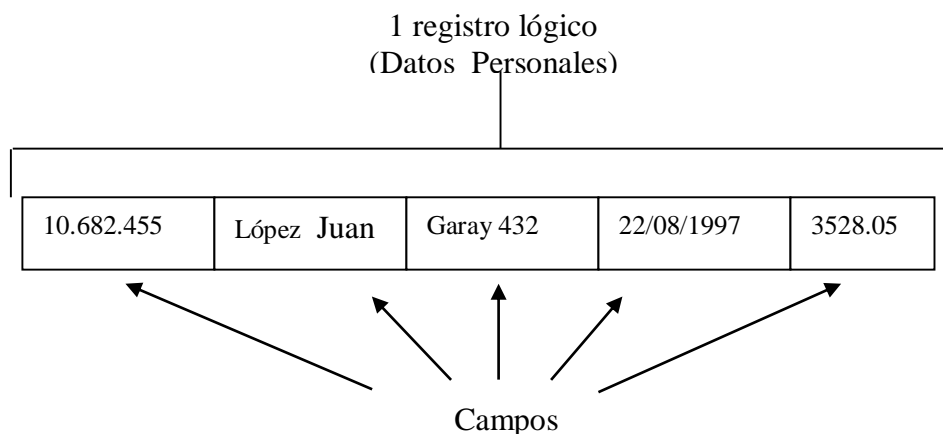
La necesidad de almacenar y procesar grandes volúmenes de datos utilizando como herramienta el computador dio lugar al uso de los denominados archivos informáticos.

**Un archivo es un conjunto ordenado de datos que tienen entre sí una relación lógica y residen en soportes de información, también llamados memorias secundarias auxiliares.**

Un archivo está compuesto por estructuras de datos más simples llamadas **registros**. Todos los registros de un archivo son del mismo tipo, es decir, tienen la misma estructura.

Cada registro está formado por **campos**, los cuales pueden ser de diferentes tipos, conteniendo información referente a una característica en particular dentro del archivo.

Por ejemplo en un archivo de personal, cada registro contiene los campos con los datos de cada empleado (número de DNI, nombre y apellido, dirección, fecha de ingreso, sueldo, etc.).



Se llama **registro lógico** al conjunto de información identificable acerca de uno de los elementos del archivo. Ejemplo: registro de Datos\_Personales

Se llama **registro físico o bloque** a la cantidad de información que se transfiere físicamente en cada operación de acceso (lectura o escritura) sobre el archivo. Ejemplo: sector del disco rígido.

La transferencia bloque a bloque, se realiza entre una zona de memoria principal asociada a las entradas/salidas llamada buffer y el registro físico o bloque.

El tamaño y formato del registro lógico los define el programador y el tamaño del registro físico viene dado por las características físicas del computador.

En general, un bloque contiene uno o más registros lógicos (registros bloqueados), pero puede ocurrir que un registro lógico ocupe más de un registro físico (registros expandidos).

Se denomina **factor de bloqueo** al número de registros lógicos que contiene cada registro físico.

Ejemplo:

Sea un disco rígido con sectores de 512 bytes. En él, se almacenó un archivo que contiene registros lógicos de 100 bytes cada uno. El factor de bloqueo se calcula como:

$$\text{Factor de bloqueo} = \text{Tamaño del registro físico} / \text{tamaño del registro lógico}$$

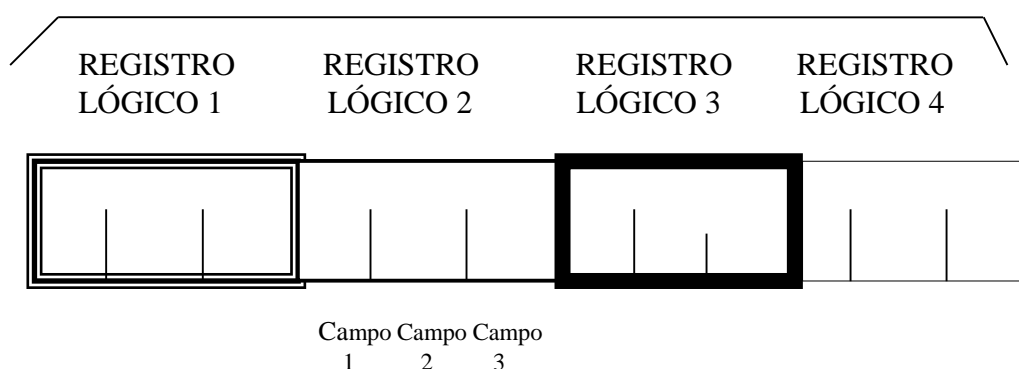
$$\text{Factor de bloqueo} = 512 \text{ bytes} / 100 \text{ bytes} = 5$$

**El factor de bloqueo indica cuántos registros lógicos se traen a memoria cada vez que se accede al disco para leer un registro físico.**

En el ejemplo “sobran” 12 bytes en cada registro físico. Es importante diseñar el registro lógico de manera tal que el número de bytes sobrantes sea lo más pequeño posible.

En un archivo con registros bloqueados, en una lectura se transfieren a la memoria central varios registros lógicos a la vez, pero para leer un registro expandido es necesario realizar más de un acceso. Es importante tener en cuenta el factor de bloqueo cuando se diseña un archivo, ya que el bloqueo de registros mejora la velocidad de los procesos de entrada/salida y el aprovechamiento de la capacidad del soporte.

REGISTRO FÍSICO: Bloque (Factor de bloqueo = 3)



Los registros se pueden identificar de manera única a través de un campo o conjunto de campos, denominados **llave** o **clave** del archivo y por lo tanto no puede aparecer repetido en otro registro diferente. Un archivo puede tener una, varias o ninguna clave en sus registros, por ejemplo en un archivo de personal el campo D.N.I. se podría tomar como llave, el registro quedaría totalmente identificado, no pasa lo mismo si tomamos como llave el campo Apellido y Nombre ya que pueden existir empleados con el mismo Apellido y Nombre.

<b>DIRECCIÓN DE UN REGISTRO</b>	<b>LÓGICA (SOFTWARE)</b>	ES LA POSICIÓN RELATIVA QUE OCUPA EN EL ARCHIVO.
	<b>FÍSICA (HARDWARE)</b>	ES LA POSICIÓN REAL O EFECTIVA DONDE SE ENCUENTRA DICHO REGISTRO EN EL SOPORTE.

Ejemplo:

<b>DIRECCIÓN LÓGICA</b>	REGISTRO 1		REGISTRO 2		REGISTRO 3		REGISTRO 4	
NOMBRE DE LOS CAMPOS	DNI	NOMBRE	DNI	NOMBRE	DNI	NOMBRE	DNI	NOMBRE
CONTENIDO	112233	JUAN	4568	ANA	9874	HUGO	2359	ELSA

En el disco rígido este archivo está almacenado en la siguiente:

**DIRECCIÓN FÍSICA:** Cara: 2 Pista: 4 Sector: 7

El Sistema Operativo ha de realizar la transformación de la dirección lógica usada en los programas en la dirección física con la que se direcciona al soporte.

## 2.2. OPERACIONES CON ARCHIVOS

<b>CREACIÓN</b>	Primera operación sobre un archivo donde se describen los datos y sus características. Se diseña el archivo.
<b>CONSULTA O RECUPERACIÓN</b>	Se realiza a nivel de registro para obtener la información contenida en él, por ejemplo consultar el domicilio de un empleado.
<b>MANTENIMIENTO O ACTUALIZACIÓN</b>	<p>Una vez creado el archivo puede ser necesario realizar distintas operaciones a nivel de registro:</p> <ul style="list-style-type: none"> <li>• <i>Inserción o Alta</i> de un registro nuevo, por ejemplo: se incorpora un empleado nuevo.</li> <li>• <i>Modificación o Cambio</i> de un registro por cambios en uno o varios campos del mismo, por ejemplo: cambia el domicilio de un empleado.</li> <li>• <i>Eliminación, borrado de un registro o Baja</i>, por ejemplo un empleado que se da de baja. Puede hacerse de 2 formas: <ol style="list-style-type: none"> <li>1. Por marca o borrado lógico: colocar en un campo un valor que será interpretado por los programas de aplicación como registro sin validez.</li> <li>2. Eliminación real: hacer inaccesible el registro o bien ocupar su espacio con otros registros.</li> </ol> </li> </ul>
<b>BORRADO O DESTRUCCIÓN</b>	Se elimina la información y la estructura del archivo. Finaliza la existencia del archivo.

## 2.3. TIPOS DE ARCHIVOS

Clasificación según el uso que se hace de los archivos

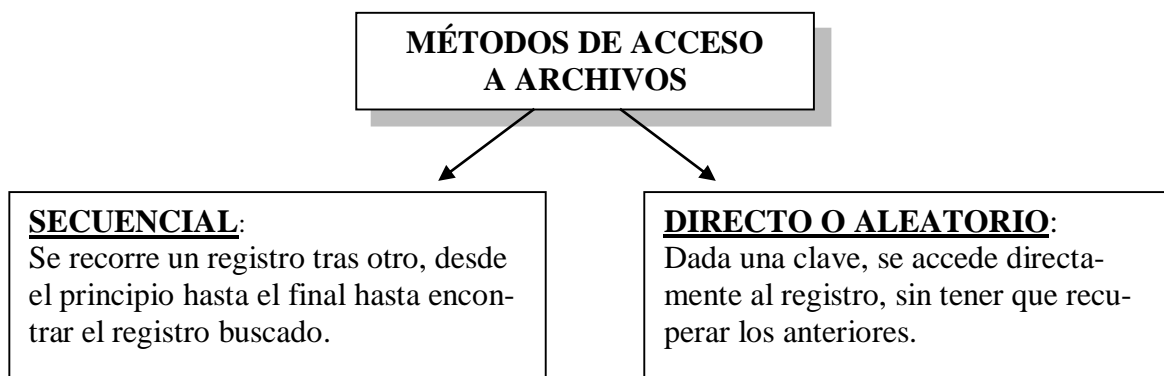
<b>PERMANENTES</b>	<p>Contienen información necesaria para el funcionamiento de una aplicación. Su vida es larga.</p> <ul style="list-style-type: none"> <li>• Archivos maestros o de situación. Refleja el estado actual de los datos, se actualiza constantemente para reflejar cada nueva situación. Ej. Estado de cuentas de un banco. Alumnos Activos de la Universidad, Padrón Electoral.</li> <li>• Archivos constantes. Su información permanece prácticamente inamovible, en general se utilizan de consulta. Ej. Archivo de códigos postales.</li> <li>• Archivos históricos. Contienen datos que fueron actuales en tiempos anteriores. Se obtienen de los maestros cuando se dejan fuera de uso para futuros estudios estadísticos o consultas. Ej. Archivo de las cuentas canceladas. Alumnos que pasaron por la Universidad.</li> </ul>
<b>TEMPORALES</b>	Contienen información necesaria para un proceso específico. Tienen una vida efímera y una vez realizada su función se cancelan.

Clasificación según el tamaño de los registros:

<b>LARGO FIJO</b>	Todos los registros tienen la misma longitud. Por ej: 100 bytes. Son más simples de manejar.
<b>LARGO VARIABLE</b>	Los registros tienen longitud variable. Necesitan que el 1er. Campo contenga el largo del registro. Ej: 5 PARIS 14 CONSTANTINOPLA. Aprovechan el espacio en forma más eficiente.

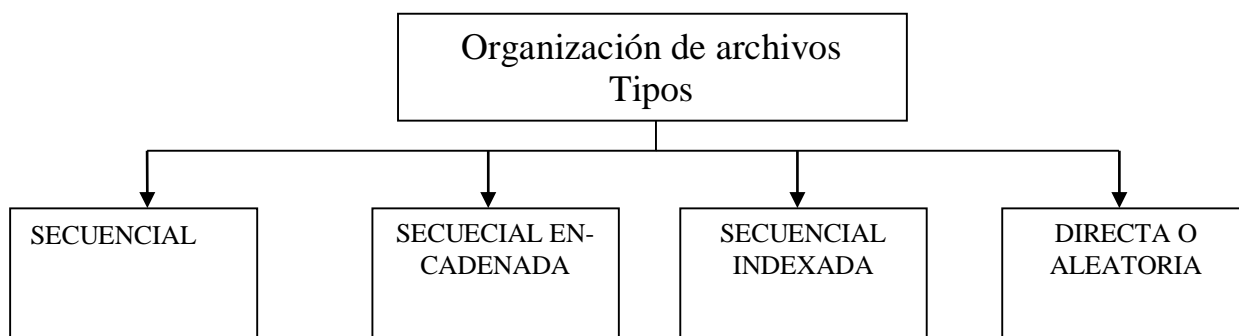
## 2.4. ORGANIZACIÓN DE ARCHIVOS

Usualmente el computador necesita acceder a los archivos ya sea para recuperar la información o para grabarla. El *método de acceso* determina como pueden recuperarse los registros.



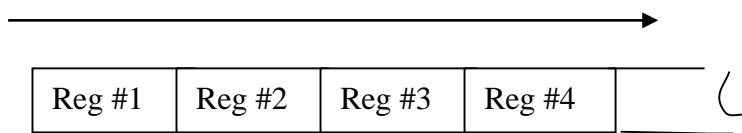
El acceso a un archivo está íntimamente ligado a la forma como están dispuestos los registros en el soporte material, por ejemplo un archivo con organización secuencial no podrá ser accedido de forma directa. Cuando se crea un archivo es necesario especificar qué organización tendrá, ya que esto va a determinar qué tipo de acceso podemos utilizar.

Los tipos de organización de archivos son básicamente:



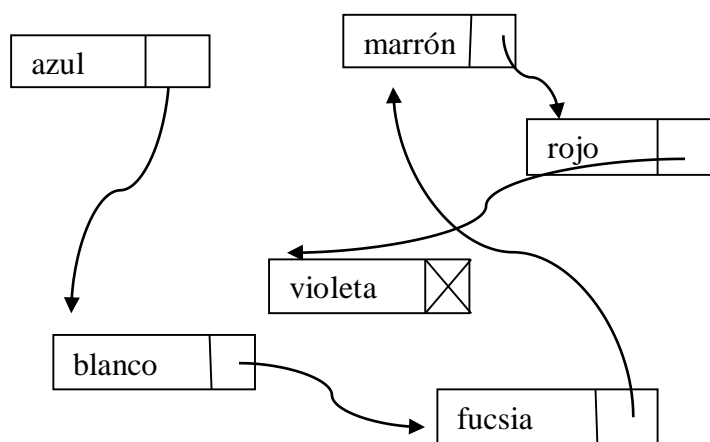
### 2.4.1. ORGANIZACIÓN SECUENCIAL

Los registros se almacenan uno después de otro, sin dejar espacio entre ellos y ordenados según una clave de clasificación.



### 2.4.2. ORGANIZACIÓN SECUENCIAL ENCADENADA

Cada registro contiene, además de los campos de información, un puntero con la dirección del registro siguiente según el orden lógico del archivo, el puntero del último registro contiene una dirección nula.



Las direcciones físicas donde se encuentran los registros son arbitrarias, pero los punteros permiten recorrer el archivo en su secuencia lógica.

### 2.4.3. ORGANIZACIÓN SECUENCIAL INDEXADA

Un archivo con esta organización consta de tres zonas o áreas:

Zona o área	Contenido
De registros	Posee una organización secuencial pura. Contiene todos los registros de datos, ordenados según un campo clave. Está dividida en una serie de segmentos o tramos lógicos, formados por registros consecutivos.
De índices	Es una estructura, con organización secuencial pura, creada y gestionada por el sistema, con una cantidad de registros muy inferior al total de registros del archivo. Cada registro está formado por solo dos campos: un campo <i>clave</i> o <i>llave</i> (contiene la clave más alta de cada segmento) y otro campo <i>dirección</i> (contiene la dirección de comienzo de dicho segmento). Por cada segmento o tramo en la zona de registros, existe un registro en la zona de índices. El sistema accede primero a la zona de índices y a través de esta accede directamente a un segmento de la zona de registros.
De desbordamiento (overflow)	Contiene los nuevos registros que no pueden ser agregados al área de registros, ya que para ello hay que reorganizar el archivo por tratarse de una estructura secuencial pura. En esta zona los registros están desordenados, ya que cada registro nuevo se añade al final de la misma.

Ejemplo de un archivo secuencial indexado:



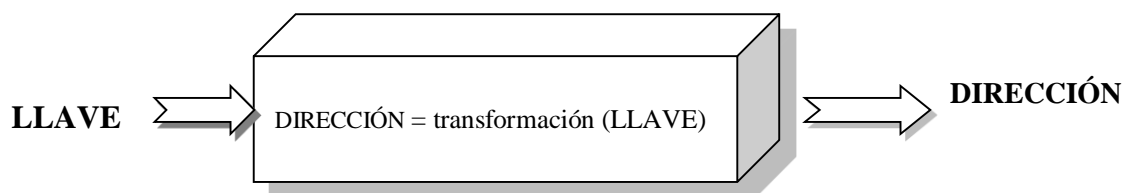
Zona de registros

	<b>Legajo (Clave)</b>	<b>Apellido y nombre</b>	<b>Domicilio</b>	<b>Bloque</b>
1	8640	Lopez Juan	Carlos Calvo 1832	1
2	9522	Areco Mirta	Azcuénaga 345	
3	10482	Lordi Marcelo	Arieta 3467	
4	12478	Blanco Pedro	Illia 2412	2
5	28453	Zamudio Pablo	Pichincha 765	
6	30125	Avalo Liliana	Varela 321	3
7	31877	Zavala Nora	Ombú 467	
8	32012	Miguez Pablo	Cabildo 1234	
9	32489	Caruso José	Santa Fé 456	
10	32546	Marin Pedro	Lavalle 54	

Zona de índices	
Última Clave del Bloque	Dirección de Inicio del Bloque
10482	1
28453	4
32546	6

#### 2.4.4. ORGANIZACIÓN DIRECTA O ALEATORIA.

En la organización indexada, el índice relaciona la clave con la dirección, a expensas de usar tablas adicionales que ocupan lugar y requieren mantenimiento. En la organización directa o aleatoria, la posición de un registro en el almacenamiento masivo, se calcula aplicando una fórmula o algoritmo matemático al valor del campo clave.



Existen muchos métodos para generar las direcciones de los registros, dependiendo del archivo concreto y de la relación claves-direcciones se elegirá el método que asegure que las direcciones estarán dentro del rango permitido y distribuidas de modo que existan pocos *sinónimos* (registros con diferentes claves pero con la misma dirección física). El problema es la elección de la transformación o métodos de direccionamiento. Se analizarán varios de ellos:

- **Direccionamiento directo:** la dirección es el valor de la clave sin ninguna transformación. El problema es que tiene que haber tantas direcciones como valores de las claves y éstas deben ser numéricas. La ventaja: garantiza que no se produzcan sinónimos.

Ejemplo: una fábrica que tiene menos de 100 empleados (números de Legajos entre 0 y 100). Podríamos crear un archivo con 100 registros donde el número de legajo se utilice directamente como la dirección de un registro. Así el registro con los datos del empleado legajo 025 estará en la dirección 25, si bien parece el método ideal, su aplicación está muy limitada. Sería muy ineficaz tomar el número de DNI como clave, el cual tiene 8 dígitos, ya que necesitaríamos un archivo con 99.999.999 registros de los cuales utilizaríamos menos de 100.

- **Direccionamiento asociado.** Se utiliza cuando las claves son alfanuméricas y consiste en asociar a cada llave una dirección lógica por medio de una tabla. No produce sinónimos pero necesita espacio adicional para la tabla.

- **Direccionamiento calculado (hashing).** La dirección de cada registro se obtiene al aplicarle a la clave ciertos cálculos. Por ejemplo:

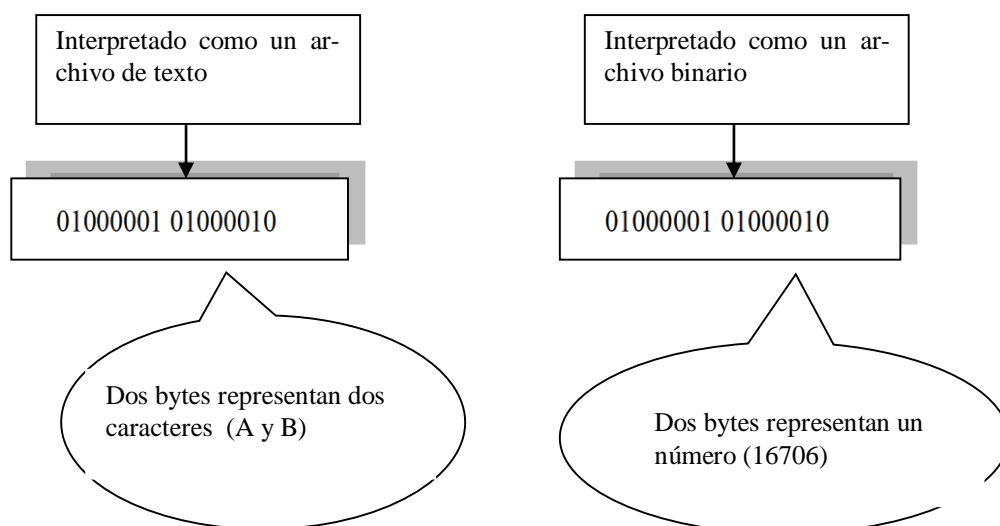
*Hashing de residuo de división (o división por un número primo).* Siendo N el número máximo estimado de registros en el archivo y P el primo más próximo a N, se divide la llave por P y se utiliza como dirección del registro el **resto** de la operación. Ejemplo:

Si N es 42, P (primo más próximo a N) = 41, calculamos las direcciones:

Llave	Dirección
3408	5
1232	2
8202	2(sinónimo)

## 2.5. ARCHIVOS DE TEXTO – ARCHIVOS BINARIOS

Un archivo almacenado en un dispositivo de almacenamiento es una secuencia de bits que puede ser interpretado por un programa de aplicación como un archivo de texto o un archivo binario.



- **Archivos de textos:** es un archivo de caracteres, números enteros, punto flotante o cualquier otra estructura de datos. Para almacenarlos deberán ser convertidos a sus formatos equivalentes de caracteres.

Si una cadena de caracteres se envía a la impresora, esta toma 8 bits, los interpreta como un byte y lo decodifica en el sistema de codificación de la impresora (ASCII o EBCDIC).

Si es un carácter imprimible, la impresora lo imprimirá, de no serlo, se realizará otra acción, por ejemplo la impresión de un espacio, el avance de una línea, etc.

- **Archivos Binarios:** Es una colección de datos (enteros, punto flotante, un carácter, etc.) almacenados en el formato interno de la computadora.

Estos datos son significativos solo si son interpretados correctamente por un programa.

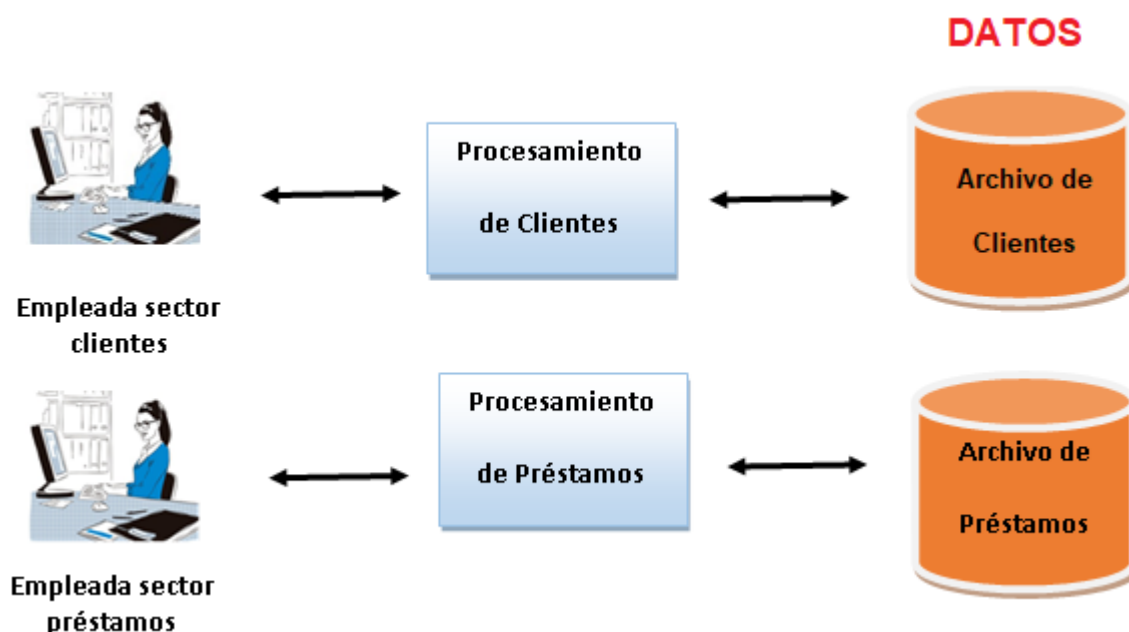
Si son numéricos 2 o más bytes se consideran un elemento de datos, por ejemplo una PC que utiliza 2 bytes para almacenar un entero y 4 bytes para un flotante. Si los datos son textuales un carácter se representa con 1 byte.

### 3. BASES DE DATOS

Si se considera que una de las aplicaciones más importantes de una computadora es el almacenamiento, recuperación y mantenimiento de grandes cantidades de información, los archivos como aplicación convencional no son del todo eficientes.

Los archivos se diseñan de acuerdo con los programas, donde el programador decide si debe haber archivos, cuántos deben ser, qué organización tendrán, qué información contendrán, qué programas actuarán sobre ellos y cómo lo harán.

En la siguiente figura puede verse un ejemplo abreviado referido a un Banco.



En este Banco, hay un sector de **Clientes** que guarda los datos necesarios de todas las personas que son clientes del Banco y allí deberán dirigirse aquellos que quieran serlo. Por otra parte, se encuentra el sector de **Préstamos** donde cada persona que necesite solicitar un préstamo deberá dirigirse allí. En este sector, la persona que atiende le solicita también los mismos datos personales que en el sector de clientes, además de asignarle el monto del préstamo, el número de cuotas, la fecha en que solicita el préstamo y la fecha de vencimiento de las cuotas entre otros.

Como se observa, con un sistema tradicional de archivos, cada sector de una misma empresa será el responsable de crear y mantener los datos necesarios, aun cuando éstos estén duplicados.

De esta manera, se pueden encontrar los siguientes problemas:

Problema	Características
Actualización de la Información	La actualización puede resultar costosa cuando se tiene información total o parcialmente duplicada en archivos diferentes. Esto conduce a inconsistencia de datos.

Redundancia	Consiste en tener datos que no aportan información, porque pueden ser deducidos de otros.
Rigidez en la búsqueda	No siempre el modo de acceso que tiene el archivo es el más eficiente, no pudiendo cambiarse.
Dependencia con los programas	Cualquier cambio en la estructura del archivo implica una modificación de los programas que lo tratan.
Confidencialidad	La confidencialidad consiste en evitar el acceso a determinados usuarios.
Seguridad	La seguridad consiste en que los datos no puedan ser modificados por usuarios no autorizados.

Las *bases de datos* surgen como una alternativa válida a los sistemas de archivos, que facilita la manipulación de grandes cantidad de información, para hacerla segura y accesible a una variedad de usuarios para una variedad de propósitos.

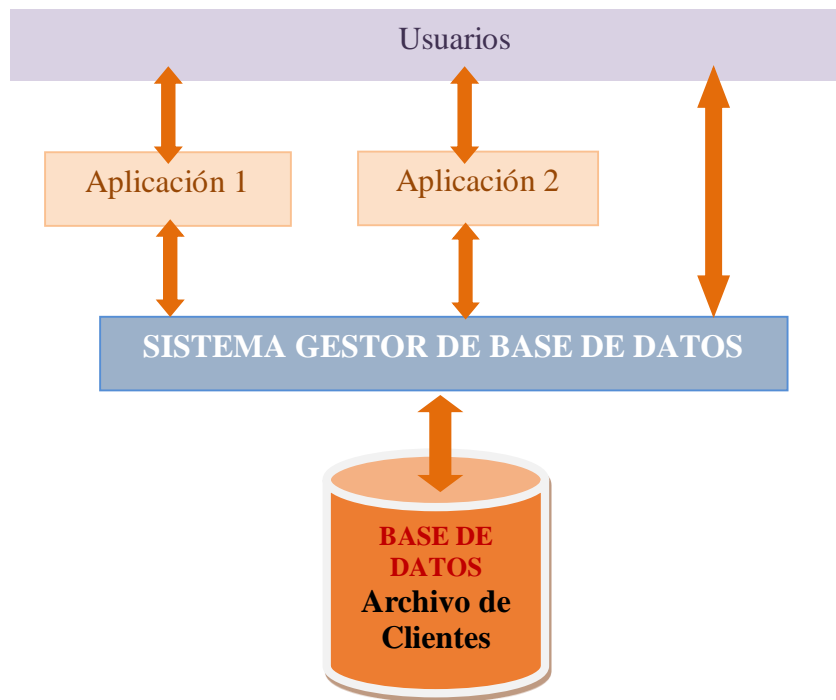
***Una base de datos es un sistema formado por un conjunto de datos organizados de tal manera que se controla el almacenamiento de datos redundantes, los datos resultan independientes de los programas que los usan y se pueden acceder a ellos de diversas formas.***

Los requisitos de una buena base de datos son:

- Varios usuarios pueden acceder a la base de datos, y cada uno accederá a determinada información.
- Se controlará el acceso de los usuarios asegurando confiabilidad y seguridad.
- Los datos se almacenan sin redundancia, excepto en casos especiales (redundancia aceptable).
- Se accede de distintas maneras, flexibilizando las búsquedas.
- Se puede modificar los contenidos, las relaciones o agregar nuevos datos sin afectar los programas que usan la base de datos.
- Existe una interfaz de la base de datos que permite usarla de forma cómoda y sencilla.

### **3.1. SISTEMA GESTOR DE BASE DE DATOS**

Se denomina *Sistema de Gestión de Base de Datos* (SGBD o DBMS, *Data Base Management System*), al conjunto de software destinado a la creación, gestión, control y manipulación de la información sobre una base de datos. Los SGBD tienen como propósito registrar y mantener información.



### 3.2. BASES DE DATOS RELACIONALES

Una base de datos relacional es una colección de elementos de datos organizados en un conjunto de tablas formalmente descritas desde la que se puede acceder a los datos.

#### 3.2.1. Componentes del modelo relacional:

##### ➤ Tablas o entidades

Las tablas son sistemas de clasificación constituidos por filas horizontales y columnas verticales que permiten agrupar datos y presentarlos de forma ordenada. La tabla lleva el nombre de la entidad que representa.

Si tomamos el ejemplo de una base de datos que guarde la información del Departamento de Ingeniería de la UNLaM acerca de los alumnos, los docentes, las carreras y las aulas donde se dictan, se tomaría como tablas ALUMNO, PROFESOR y CARRERA, que representan respectivamente a una entidad.

##### ➤ Atributos

Un atributo es una unidad básica e indivisible de información acerca de una entidad que sirve para identificarla o describirla.

Si continuamos con el ejemplo anterior, atributos de la tabla ALUMNO son: DNI, Apellido y nombre, fecha de nacimiento, domicilio, carrera que cursa (Informática, Industrial o Electrónica), etc.

##### ➤ Registros o tuplas

En una base de datos, la información de cada tabla se almacena en registros, y cada atributo, en campos de dicho registro. Existen distintos tipos de registros dentro de la misma base de

datos, ya que cada tabla necesitará una estructura distinta, pero todos los registros de una misma tabla deben ser del mismo tipo.

Siguiendo con el ejemplo anterior, la estructura del registro de la tabla ALUMNO va a ser muy distinta del registro de la entidad CARRERAS, ya que ambas entidades tienen distintos atributos.

**Tabla: Alumno**

DNI	Apellido	Legajo	F.Nacimiento	Cod. Carrera
45236587	Robles	1941	20/02/2001	20
43201897	Mariani	0122	18/05/2000	10
42568032	Juárez	3251	15/12/1999	30

} Registros

**Tabla: Carreras**

Cód. Carrera	Nombre
10	Informática
20	Electrónica
30	Civil
40	Industrial
50	Mecánica

} Registros

➤ **Claves:** *superclave*, *clave candidata*, *clave principal o primaria* y *clave foránea o ajena*.

Se llama **superclave** de una entidad a un atributo o conjunto de atributos que permite identificar de forma única a un registro de una entidad.

Si de una **superclave** no se puede obtener ningún subconjunto que a su vez sea **superclave**, se dice que dicha **superclave** es **clave candidata**.

De todas las **claves candidatas** existentes, el diseñador de la base de datos, escogerá una que individualizará de forma única a cada *registro* de la *entidad*. Esta clave se denomina **clave principal o primaria**.

Cuando existe una referencia entre dos entidades, esto es, cuando un campo o conjunto de campos de una de las entidades es la clave de otra, se la llama **clave foránea o ajena**. Esta clave foránea permite localizar una entidad a partir de otra.

En el ejemplo anterior, en la entidad ALUMNO algunas **superclaves** pueden ser:

- DNI
- Legajo
- DNI + Apellido + Nombre
- DNI + Apellido + Nombre + Fecha de nacimiento

Las **claves candidatas** serían todas aquellas que no permitan encontrar un subconjunto que pueda ser a su vez **superclave**:

- DNI
- Legajo

Si estas *claves candidatas*, se escoge DNI. Entonces DNI es la *clave primaria o principal*.

La entidad ALUMNO tiene un atributo **Cod. Carrera**. En la entidad CARRERAS, el código de la carrera es la clave primaria o principal. Por lo tanto, en la entidad ALUMNO el atributo **Cod. Carrera** es una *clave foránea*, ya que relaciona a esta entidad con la entidad CARRERA.

### ➤ Relaciones

Las entidades por sí solas no describen la realidad de un sistema de información. Además de identificar objetos, hay que establecer las asociaciones existentes entre ellos. Esto es una *relación*: la existencia de *algo común* entre entidades.

Siguiendo con el ejemplo, existe la *relación* entre las entidades ALUMNO – CARRERAS (el alumno cursa una carrera).

<b>DNI</b>	Apellido	Nombre	F.Nacimiento	<b>Cod. Carrera</b>
45236587	Robles	Mariano	20/02/2001	20
43201897	Mariani	Rocío	18/05/2000	10
42568032	Juárez	Ana	15/12/1999	30

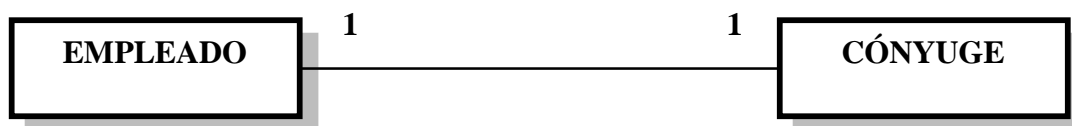
<b>Cod. Carrera</b>	Nombre
10	Informática
20	Electrónica
30	Civil
40	Industrial
50	Mecánica

En el ejemplo se observa en color rojo a las *claves principales* de cada una de las tablas (“**DNI**” en la tabla ALUMNO y “**Cod. Carrera**” en la tabla CARRERAS) y en verde a la *clave foránea*, en esta caso se encuentra en la tabla ALUMNO y es “**Cod. Carrera**”. Este campo es el que permite la relación entre las tablas.

### ➤ Cardinalidad

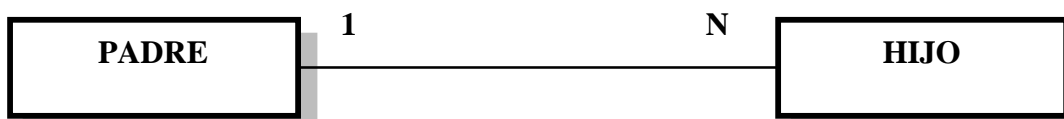
La cardinalidad de una relación representa la participación en la relación de cada una de las entidades afectadas. La cardinalidad siempre se evalúa de a dos entidades. Existen tres tipos posibles

✓ **1:1 (una a una):** A cada registro de una entidad le corresponde no más de un registro de la otra y viceversa. Es **biunívoca**.



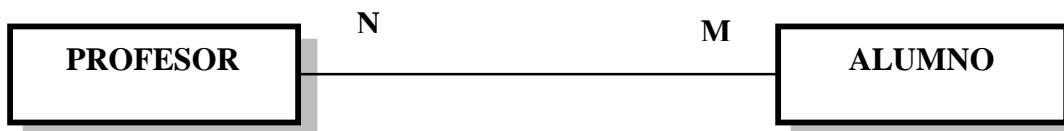
La relación EMPLEADO – CONYUGE (un empleado puede estar casado con una única persona)

✓ **1:N (una a muchas):** A cada registro de la primera entidad le pueden corresponder varios registros de la segunda, y a cada registro de la segunda le corresponde no más de uno de la primera.



La relación PADRE – HIJO (un padre puede tener muchos hijos, pero éstos sólo tendrán un padre)

✓ **N:N (muchas a muchas):** A cada registro de la primera entidad le pueden corresponder varios registros de la segunda y viceversa. No son usadas en Bases de Datos Relacionales.



La relación PROFESOR – ALUMNO (un profesor da clase a muchos alumnos y un alumnos tiene varios profesores)

### 3.3. USUARIOS DE LA BASE DE DATOS

Tipo de Usuario	Características	Responsabilidades
Administrador de Base de Datos	Encargado control general del sistema base de datos.	<ul style="list-style-type: none"> <li>- Decidir el contenido de la información.</li> <li>- Decidir las estructuras de almacenamiento y la estrategia de acceso.</li> <li>- Vincularse con el resto de los usuarios de la base de datos.</li> <li>- Definir controles de autorización y validación.</li> </ul>



		<ul style="list-style-type: none"> <li>- Definir estrategia de respaldo y recuperación de datos por fallas del sistema.</li> <li>- Controlar el rendimiento y utilización de la base.</li> <li>- Responder a los cambios en los requerimientos.</li> </ul>
<b>Programador de aplicaciones</b>	Escriben los programas que utilizan las Bases de Datos.	<ul style="list-style-type: none"> <li>- Realizar tareas de creación, borrado, consulta, recuperación o modificación de datos.</li> <li>- Diseñan aplicaciones para apoyar al usuario final.</li> </ul>
<b>Usuario Final</b>	Accede a la Base de Datos desde una terminal.	<ul style="list-style-type: none"> <li>- Utiliza los programas creados por el programador.</li> </ul>

### 3.4. APLICACIONES DE LAS BASES DE DATOS

#### 3.4.1. Big Data:

##### ➤ ¿Qué es Big Data?

“Big data son datos que contienen una mayor variedad y que se presentan en volúmenes crecientes y a una velocidad superior”. Gartner (aprox. 2001)<sup>1</sup>

En términos generales se puede decir que es la tendencia existente en el avance de la tecnología que ha abierto las puertas hacia un nuevo enfoque de entendimiento y toma de decisiones, la cual es utilizada para describir enormes cantidades de datos (estructurados, no estructurados y semi estructurados) que tomaría demasiado tiempo y sería muy costoso cargarlos a un base de datos relacional para su análisis.

De tal manera que, el concepto de Big Data aplica para toda aquella información que no puede ser procesada o analizada utilizando procesos o herramientas tradicionales.

El Big Data está basado en tres características denominadas “Las tres V”:

<b>Volu-men</b>	<p>La cantidad de datos importa.</p> <p>Con Big Data, hay que procesar grandes volúmenes de datos no estructurados que pueden tratarse de datos de valor desconocido, como feeds de datos de Twitter, flujos de clics de una página web o de aplicación para móviles, o de equipos con sensores.</p> <p>Para algunas organizaciones, esto puede suponer decenas de terabytes de datos, para otras, incluso cientos de Petabytes.</p>
<b>Velo-cidad</b>	<p>La velocidad es el ritmo al que se reciben los datos y (posiblemente) al que se utilizan. Por lo general, la mayor velocidad de los datos se transmite directamente a la memoria, en vez de escribirse en un disco. Algunos productos inteligentes habilitados para Internet funcionan en tiempo real o prácticamente en tiempo real y requieren una evaluación y actuación en tiempo real.</p>

<sup>1</sup> Gartner Inc. es una empresa consultora y de investigación de las tecnologías de la información con sede en Stamford, Connecticut, Estados Unidos.

<b>Va- rie- dad</b>	<p>La variedad hace referencia a los diversos tipos de datos disponibles.</p> <p>Los tipos de datos convencionales eran estructurados y podían organizarse claramente en una base de datos relacional.</p> <p>Con el auge del big data, los datos se presentan en nuevos tipos de datos no estructurados. Los tipos de datos no estructurados y semiestructurados, como el texto, audio o video, requieren un preprocesamiento adicional para poder obtener significado y habilitar los meta-datos.</p>
-----------------------------	---

Hace un tiempo que, a las tres V, se le han agregado dos más

<b>Va- lor</b>	<p>Convertir la información en conocimiento debe ser el propósito de cualquier estrategia de Big Data y esto se consigue a través de la analítica que ayuda en la toma de decisiones. Se trata en definitiva de utilizar los datos de manera rentable y a la vez eficiente. Tener datos es fácil, lo complicado y el factor verdaderamente diferencial es el uso que se da de estos para comprender, por ejemplo, el comportamiento de los clientes o alcanzar niveles óptimos en competitividad.</p>
<b>Ve- raci- dad</b>	<p>El Big Data hay que controlar necesariamente la integridad de los datos, ya que de esto dependerá del acierto de las decisiones. Una de la dimensiones más importantes también del Big Data señala a la fiabilidad de la información recogida eliminando cualquiera inexactitud o incertidumbre en la recopilación de estos, vinculándose este punto con el tratamiento de los datos o limpieza de estos. Esto supone un gran reto a medida que los datos crecen.</p>

### ➤ Un poco de historia

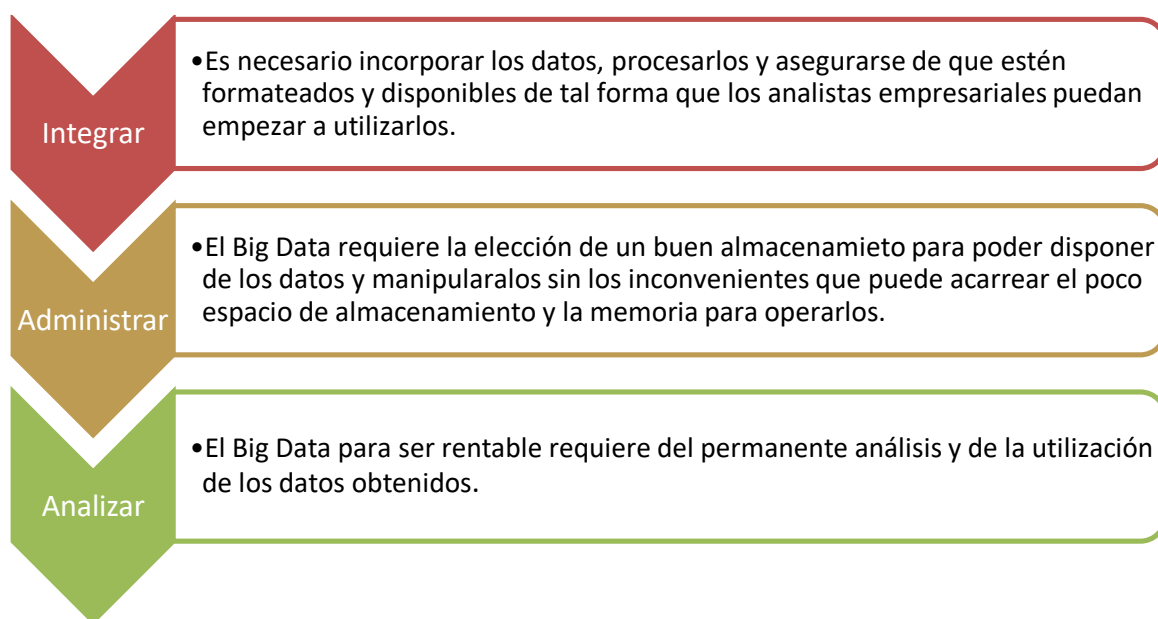
Si bien el concepto es nuevo, el manejo de grandes volúmenes de datos se remonta hacia los años 1960/1970 cuando se comenzaron a estructurar los datos en las bases de datos relacionales.

Hacia 2005, se comenzó a observar la gran cantidad de datos que se generaban a través de las redes sociales y servicios online. Eso hizo que se comenzara a trabajar para ver cual podría ser la utilidad que se le podía dar a esos datos, resultando ser útiles para gran cantidad de sectores tanto públicos como privados.

Con la llegada de Internet de las cosas, hay muchos dispositivos conectados a Internet que generan datos sobre patrones de uso de los clientes y el rendimiento de los productos, generando que estos datos bien procesados permitan tomar decisiones más certeras.

### ➤ Cómo funciona Big Data

Para que el Big Data aporte nuevas perspectivas a cualquier negocio que lo quiera implementar, requiere de tres acciones básicas:



➤ **Algunos Ejemplos de su uso**

<b>Desarrollo de productos</b>	Empresas como Netflix y Procter & Gamble usan big data para prever la demanda de los clientes. Construyen modelos predictivos para nuevos productos y servicios, clasificando atributos clave de productos anteriores y actuales, y modelando la relación entre dichos atributos y el éxito comercial de las ofertas.
<b>Mantenimiento predictivo</b>	Los factores capaces de predecir fallas mecánicas provienen de diversas fuentes profundamente ocultos entre datos estructurados (año del equipo, marca o modelo de una máquina) o entre datos no estructurados que cubren millones de entradas de registros, datos de sensores, mensajes de error y temperaturas de motor. Al analizar estos indicadores de problemas potenciales antes de que estos se produzcan.
<b>Experiencia del cliente</b>	El big data le permite recopilar datos de redes sociales, visitas a páginas web, registros de llamadas y otras fuentes para mejorar la experiencia de interacción, así como maximizar el valor ofrecido.
<b>Fraude y cumplimiento</b>	El big data le ayuda a identificar patrones en los datos que pueden ser indicativos de fraude, al tiempo que concentra grandes volúmenes de información para agilizar la generación de informes normativos.
<b>Aprendizaje automático</b>	El aprendizaje automático es un tema candente en la actualidad. Los datos, concretamente big data, es uno de los motivos de que así sea. Ahora, en lugar de programarse, las máquinas pueden aprender. Esto es posible gracias a la disponibilidad de big data para crear modelos de aprendizaje automático.
<b>Eficiencia operativa</b>	El big data permite analizar y evaluar la producción, la opinión de los clientes, las devoluciones y otros factores para reducir las situaciones de falta de stock y prever la demanda futura.
<b>Impulsa la innovación</b>	El big data puede ayudarle a innovar mediante el estudio de las interdependencias entre seres humanos, instituciones, entidades y procesos, y, posteriormente, mediante la determinación de nuevas maneras de usar dicha información.

### 3.4.2. Data Warehouse

### ➤ ¿Qué es Data Warehouse?

Un data Warehouse (almacén de datos), es un sistema de gestión de datos, diseñado para permitir y respaldar actividades de inteligencia empresarial (BI), especialmente análisis.

Los almacenes de datos se han diseñado exclusivamente para realizar consultas y análisis y, a menudo, contienen grandes cantidades de datos históricos. Los datos de un almacén de datos se obtienen normalmente de una amplia variedad de fuentes, como archivos de registro de las aplicaciones y aplicaciones de transacciones.

Sus capacidades analíticas permiten que las organizaciones obtengan información empresarial valiosa de sus datos para mejorar la toma de decisiones.

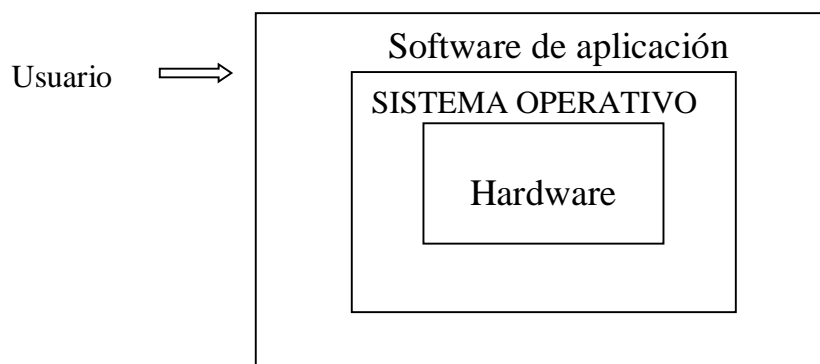
Un almacén de datos típico está compuesto por:

- Una base de datos relacional para almacenar y gestionar los datos.
- Una solución de extracción carga y transformación (ELT) para preparar los datos para el análisis.
- Funciones de análisis estadísticos, elaboración de informes y extracción de datos.
- Herramientas de análisis de clientes para visualizar y presentar los datos a los usuarios de empresa.
- Otras aplicaciones analíticas más avanzadas que generan información procesable aplicando algoritmos de aprendizaje automático e inteligencia artificial (IA).

## 4. SISTEMAS OPERATIVOS

### 4.1. INTRODUCCIÓN

El software es el elemento que permite que una computadora pueda almacenar, procesar y recuperar información. Se trata de aislar al usuario de la complejidad del hardware, añadiendo una capa de software sobre el hardware que se encarga de gestionar todos los elementos del sistema y facilitando de ese modo el trabajo con el computador. Esta capa de software, tan ligado al hardware, recibe el nombre de Sistema Operativo.



Un **Sistema Operativo** es un conjunto de programas que controlan el funcionamiento del hardware ocultando sus detalles al usuario, permitiéndole así trabajar con la computadora de una manera más fácil y segura.

Por otra parte, el computador posee un conjunto de elementos, necesarios para cumplir con su trabajo, llamados recursos, que deben ser racionalmente distribuidos y utilizados para un mejor rendimiento. El Sistema Operativo es el administrador de los recursos ofrecidos por el hardware.

Los principales recursos de un computador son: el procesador, la memoria principal, los dispositivos periféricos y la información (los datos).

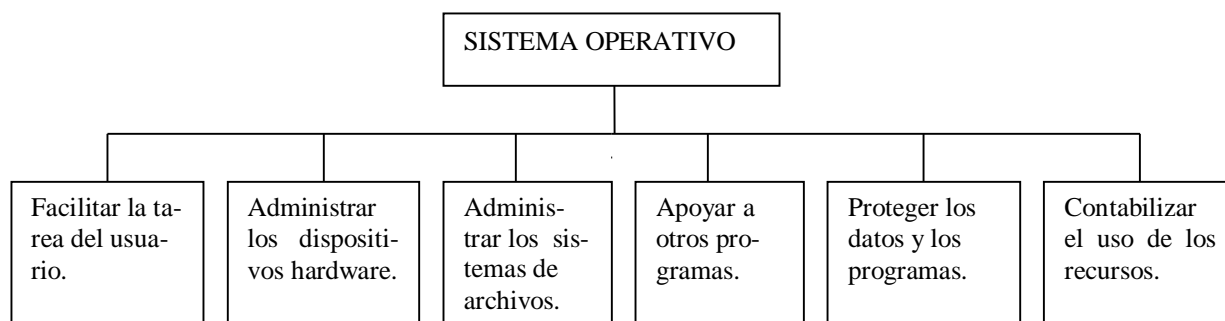
Sin un sistema operativo, un computador nunca podría empezar a funcionar, por lo tanto cuando se enciende un computador lo primero que ha de ocurrir es la carga del S. O. en la memoria principal. El *cargador* del sistema operativo es un módulo que tiene como función principal cargar el programa ejecutable en memoria.

Primero se ejecuta un programa de autodiagnóstico de encendido (CMOS), que identifica todos los dispositivos de hardware conectados, luego se ejecuta el *cargador* inicial, que carga un programa de autoarranque más eficiente, el mismo busca el S.O. y carga parte del mismo (parte residente) en la memoria principal.

Una vez que el ordenador ha puesto en marcha el Sistema Operativo mantiene parte de él en su memoria en todo momento.

#### 4.2. FUNCIONES DE LOS SISTEMAS OPERATIVOS

Las funciones que realiza un sistema operativo dependen del tipo de S.O., pero existen algunas que se pueden considerar en todos los S.O:

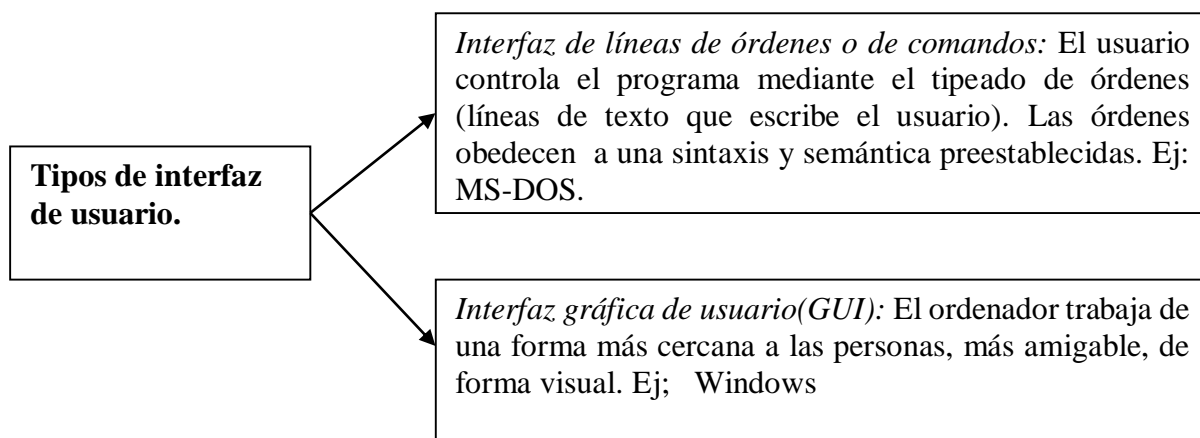


Podemos considerar que cada una de estas tareas es realizada por un módulo del S.O. Existe un módulo distinguido, denominado **Kernel**, es el núcleo del S.O., que sirve a todos los demás módulos. El Kernel es el módulo del S.O. de más bajo nivel, más ligado al hardware, siempre permanece en la memoria principal y, entre otras cosas, se encarga del manejo de interrupciones, la asignación de trabajos al procesador y proporciona una vía de comunicación entre los programas.

Para comunicarse con la computadora, los programas incluyen módulos para definir la interfaz del usuario. Cada programa de aplicación dispone de sus propios módulos de interfaz con el usuario y el S.O también dispone de su propia interfaz de usuario.

Otra de las funciones del S.O la cumple Distribuidor (dispatcher), es el que se encarga de poner en el PC la dirección física de la primera instrucción de los programas a ejecutar.

Las funciones principales del S.O. son controladas por el núcleo, pero la interfaz con el usuario es controlada y establecida por el *entorno (shell)* o *intérprete de órdenes*, es un módulo independiente del sistema operativo que acepta solicitudes de los usuarios (procesos) y las interpreta para el resto del S.O.



#### 4.3. TIPOS DE SISTEMAS OPERATIVOS

MONOTAREA O SERIE.	Fueron los primeros sistemas operativos, en ellos hasta que no finalizaba la ejecución de un programa no empezaba a ejecutarse otro. El rendimiento alcanzado con estos S. O. era muy bajo, debido a la existencia de tiempos en los cuales el procesador no realizaba ningún trabajo útil (por ejemplo, mientras se realizaban E/S).
MULTITAREA O MULTIPROGRAMADOS.	Capaces de ejecutar más de un programa al mismo tiempo. Son los más usados en la actualidad.
MONOUSUARIO.	Simples, permiten la conexión de un único usuario en un instante dado, por lo cual no necesitan realizar la gestión y control de varios usuarios. Pueden ser monotarea o multitarea.
MULTIUSUARIO.	Más de un usuario accede al computador al mismo tiempo. El S.O. debe ser también multitarea y establecer mecanismos de identificación, autenticación y control de los distintos usuarios. Además cada usuario puede ejecutar varios programas simultáneamente.
MULTIPROCESO.	Dos o más procesadores interconectados trabajando simultáneamente, formando un único ordenador.
EN TIEMPO REAL.	Para el control de aplicaciones en Tiempo Real (en las cuales el factor tiempo es crucial). Los S.O. en tiempo real deben ser capaces de responder a determinados eventos en plazos de tiempos previamente determinados. Muy usados en control industrial, control de vuelo, etc.

#### **4.4. EJEMPLOS DE SISTEMAS OPERATIVOS**

Algunos ejemplos de S.O. son:

- UNIX
- LINUX y todas sus versiones Ubuntu, Red Hat, Debian, etc..
- La familia de S.O. WINDOWS (10, 2000, CE, XP, VISTA, SERVER 2003, etc.).
- Sistemas operativos para celulares y tablets: Android y IOs.

### **5. LENGUAJES DE PROGRAMACIÓN**

#### **5.1. INTRODUCCIÓN**

Construir programas directamente utilizando el lenguaje de la máquina presenta grandes dificultades, para superar esto se han desarrollado otros lenguajes adecuados para programar cualquier computadora, más fáciles de aprender y utilizar, reduciendo la posibilidad de cometer errores y que puedan ser traducidos al lenguaje que entiende una máquina concreta.

**Un lenguaje de programación es un conjunto de símbolos junto a un conjunto de reglas para combinar dichos símbolos que se usan para escribir programas.**

Como cualquier tipo de lenguaje, se compone de un *léxico* predefinido (conjunto de símbolos permitidos), una *sintaxis* predefinida (reglas que indican como realizar la construcción del lenguaje) y una *semántica* (reglas que permiten determinar el significado de cualquier construcción del lenguaje).

Para que una computadora pueda ejecutar un programa escrito en un determinado lenguaje de programación, es necesaria primero una *traducción* del programa al lenguaje que entiende dicha computadora (lenguaje de máquina), según una serie de etapas.

#### **5.2. LENGUAJE DE MÁQUINA**

Fue el primer lenguaje utilizado en la programación de computadoras. Es el único lenguaje que entiende directamente un computador, por lo cual su estructura está totalmente ligada a los circuitos de la máquina y muy alejada de la forma de expresión y análisis de los problemas propios del hombre.

Frente a lo complicado que resulta programar en lenguaje de máquina, el código de máquina hace posible que el programador aproveche al máximo los recursos existentes.

Sus principales características son:

- Las instrucciones se expresan en binario como cadenas de ceros y unos, pudiéndose utilizar códigos intermedios (octal y hexadecimal), esto hace a los programas difíciles de entender y modificar.
- Los datos se referencian por medio de las direcciones de memoria donde se encuentran (no usan nombres de variables).
- Las instrucciones realizan operaciones muy simples, debiendo el programador expresar cada una de las operaciones que desee realizar solo con las instrucciones elementales que dispone.

- Poca *portabilidad*, un programa en lenguaje de máquina solo se puede ejecutar en el procesado para el cual está destinado, por estar íntimamente ligado a la CPU del ordenador.

.Ejemplo de un fragmento de código en lenguaje de máquina:

Instrucción de maquina en binario
0001 0001 0011 1011
1101 0011 1001 1110
1010 0101 1100 1101
1001 0111 0011 1111

### 5.3. LENGUAJE ENSAMBLADOR O SIMBÓLICO

Es el primer intento de sustituir el lenguaje de máquina por uno más cercano al hombre. Si bien presenta la mayoría de los inconvenientes del lenguaje de máquina (repertorio reducido de instrucciones, rígido formato de las instrucciones, baja portabilidad y fuerte dependencia del hardware), presenta las siguientes ventajas con respecto al mismo:

- Evita los códigos numéricos usando una notación simbólica o nemotécnica (generalmente constituidos por las abreviaturas de las operaciones en inglés) para representarlos. Por ejemplo: la suma se representa en la mayoría de los ensambladores por ADD.
- Direccionamiento simbólico. Los datos pueden identificarse con nombres, por ejemplo: FECHA, IMPORTE, IVA, etc., en lugar de direcciones binarias absolutas.

Al igual que el lenguaje de máquina goza de la ventaja de mínima ocupación de memoria y mínimo tiempo de ejecución en comparación con el resultado de la compilación del programa escrito en otros lenguajes. El lenguaje ensamblador hace corresponder a cada instrucción en ensamblador una instrucción en código máquina. Esta traducción es realizada por un programa traductor llamado **ensamblador**. El *ensamblador* es un traductor que traduce un programa escrito en Lenguaje Ensamblador a Lenguaje de Máquina.

Los programadores usan este lenguaje para afinar partes importantes de programas escritos en lenguajes de más alto nivel. El lenguaje ensamblador da al programador el control total de la máquina, permitiéndole generar un código compacto, rápido y eficiente. Ejemplo de un fragmento de código en lenguaje ensamblador:

INICIO:	ADD	B, 1
	MOV	A, B
	CMP	A, E
	JE	FIN
	JMP	INICIO
FIN:	END	

### 5.4. LENGUAJE DE ALTO NIVEL

El esfuerzo por hacer la tarea de programación independiente de la máquina condujo al desarrollo de lenguajes de alto nivel. Sus características principales son:



- Independencia de la arquitectura física del computador, lo cual permite utilizar un mismo programa en diferentes equipos (portabilidad), sin necesidad de conocer el hardware específico.
- Requiere de una traducción al lenguaje de máquina de la computadora donde va a ejecutarse.
- Se aproximan al lenguaje natural, para que el programa se pueda leer y escribir de una forma más sencilla. Las instrucciones vienen expresadas mediante texto, permitiendo incluir comentarios.
- Por lo general, una sentencia da lugar, al ser traducida a varias instrucciones en lenguaje de máquina.
- Se incluyen rutinas de uso frecuente, como las de Entrada/Salida, funciones matemáticas de uso frecuente (seno, coseno, conversión de entero a real), que figuran en una especie de librería del lenguaje, las cuales se pueden utilizar sin necesidad de programarlas cada vez.
- El lenguaje de alto nivel, a diferencia del de máquina y el ensamblador, no permite aprovechar totalmente los recursos internos de la máquina.

Todas estas características ponen de manifiesto un acercamiento a las personas y un alejamiento de la máquina, por eso los programas de alto nivel no pueden ser directamente interpretados por las computadoras, siendo necesario una traducción previa a lenguaje de máquina. Para ello se utilizan programas traductores, previamente desarrollados para cada computador, que se encargan de realizar dicho proceso.

Algunos de los más conocidos son: Fortran, Cobol, C++, Java, Lisp, Prolog, Modula-2, Visual Basic, PHP, XML.

## 5.5. TRADUCTORES: COMPILADORES E INTÉRPRETES

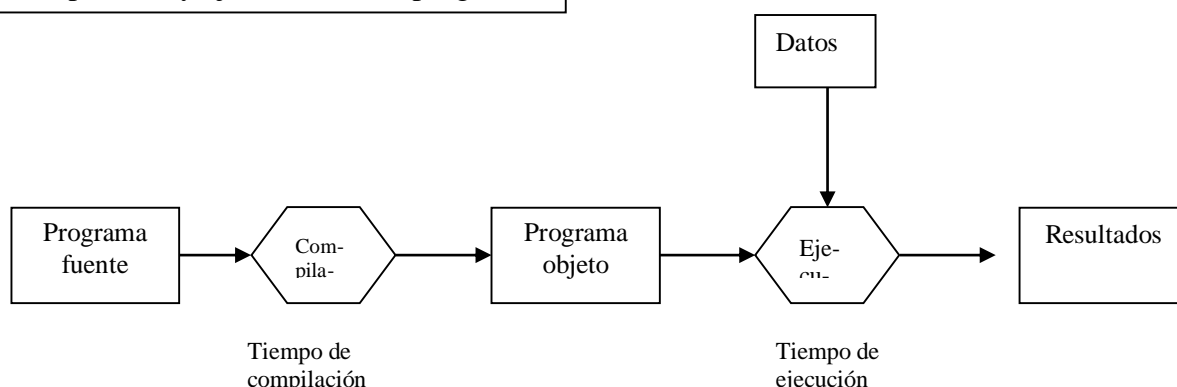
Un **traductor** es un metaprograma que toma como entrada un programa (o parte de él) escrito en lenguaje simbólico alejado de la máquina denominado **programa fuente** y proporciona como salida otro programa semánticamente equivalente, escrito en un lenguaje comprensible por la máquina denominado **programa objeto**.

Veremos dos tipos de traductores, los compiladores y los intérpretes, los cuales realizan la tarea de manera muy distinta.

Un **compilador** traduce completamente un programa fuente, generando un programa objeto escrito en lenguaje de máquina. En este proceso de traducción, se informa al usuario de errores en el programa fuente, sólo se crea el programa objeto si no se detectaron errores.

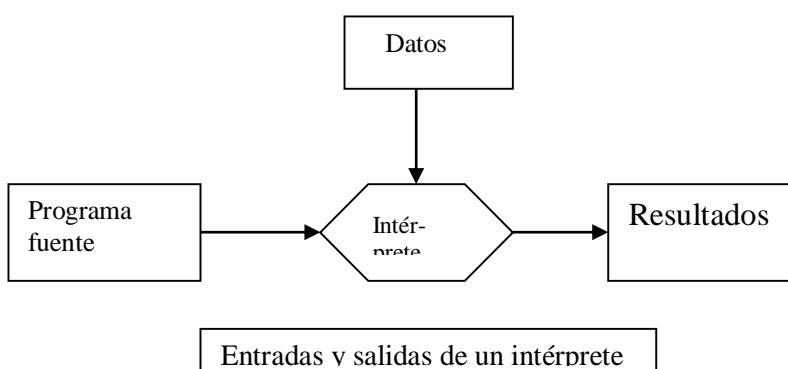
Piense el alumno en un traductor de una novela. Primero lee toda la obra, comprende sus personajes, su modo de expresarse, etc. Luego traduce el libro como una unidad.

### Compilación y ejecución de un programa:



Un **intérprete** permite que un programa fuente escrito en un determinado lenguaje vaya traduciéndose y ejecutándose directamente sentencia a sentencia por la computadora. Capta una sentencia fuente, la analiza y la interpreta, teniendo lugar su ejecución inmediata.

Ahora piense el alumno en un intérprete humano en cargado de traducir un discurso en el momento que este ocurre. El intérprete va traduciendo frase a frase, en tiempo real, (no espera a que el orador termine el discurso).



El intérprete no crea ningún programa objeto, por lo cual cada vez que se deba ejecutar el programa, se deberá traducir, en cambio, con un compilador, aunque sea más lenta, la traducción sólo debe realizarse una vez. La principal ventaja del intérprete respecto a los compiladores, es que resulta más fácil depurar los programas, ya que permite interrumpir en cualquier momento para conocer por ejemplo, los valores de las distintas variables o la instrucción fuente que acaba de ejecutarse, por estos motivos se los prefiere para fines pedagógicos.

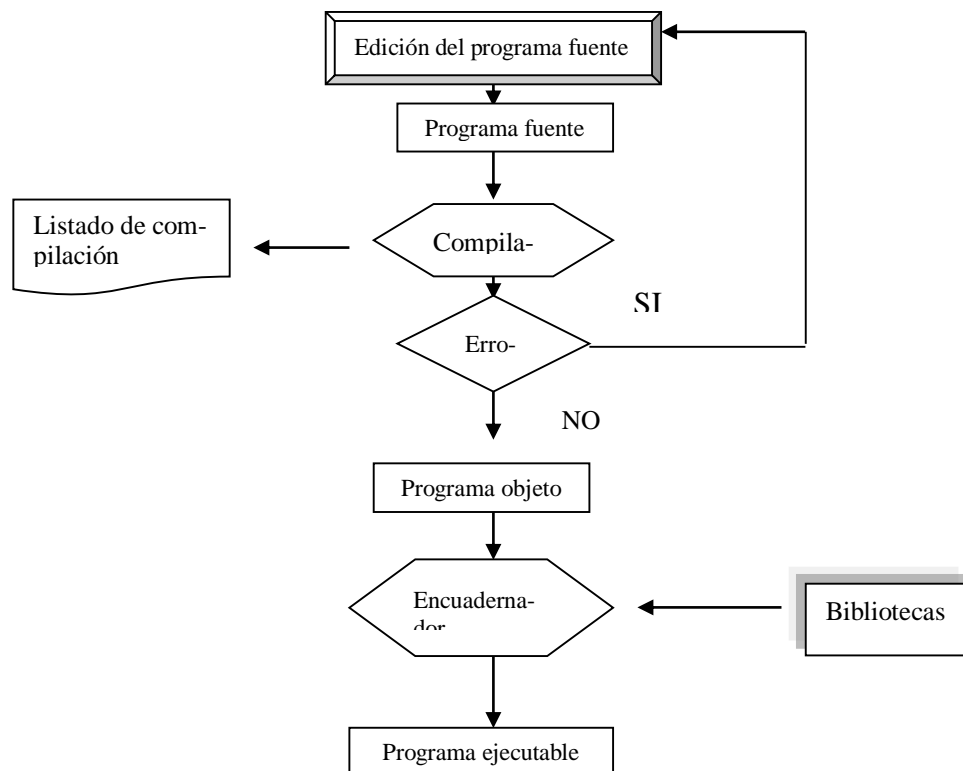
### 5.5.1. PROCESO DE COMPILACIÓN

Previo al proceso de compilación de un programa, se crea el programa fuente usando cualquier editor de texto. Para que la compilación sea posible el programa fuente debe residir en memoria simultáneamente con el compilador.

Si como resultado de la compilación se notifican errores, no se genera programa objeto, pero se genera un informe (listado de compilación) indicando la naturaleza de los mismos. Este último permitirá, volviendo al programa editor, corregir los errores y empezar de nuevo el proceso.

Terminada la compilación y obtenido el programa objeto, se lo somete a un proceso de montaje donde se enlazan los distintos módulos que lo componen (en caso de programas que poseen subprogramas, los cuales pueden ser compilados separadamente) y se incorporan las denominadas rutinas de bibliotecas (libraries), que contienen funciones prearmadas para facilitar la programación.

Este montaje es realizado por un programa denominado **montador o encuadernador o editor de enlace o linker**.



**Proceso de compilación de un programa**

## 6. COMPUTACIÓN EN LA NUBE (CLOUD COMPUTING)

### ➤ Concepto

*“La Computación en la Nube permite a las personas y empresas acceder a servicios de tecnología desde cualquier parte y en cualquier momento a través de su conexión a Internet”.*



Es uno de los pilares de la Industria 4.0, ya que promete varias ventajas atractivas para las empresas y los usuarios finales. La computación en la nube es un término general para la prestación de servicios alojados a través de Internet, y permite a las empresas y usuarios consumir recursos informáticos como una utilidad (igual que la electricidad) en lugar de tener que construir y mantener infraestructuras de computación en sus casas u oficinas.

## ➤ Ventajas y desventajas

Ventajas
<ul style="list-style-type: none"><li>• Menores costos de hardware</li><li>• Rendimiento</li><li>• Costos reducidos de software</li><li>• Actualizaciones instantáneas</li><li>• Aumento de la fiabilidad de los datos</li><li>• Facilitar la colaboración en grupo</li><li>• Independencia del dispositivo</li></ul>

- **Menores costos de hardware:** no es necesario tener una computadora de alta potencia y alto precio para ejecutar aplicaciones en la nube, dado que dichas aplicaciones no son ejecutadas localmente en la computadora. Tampoco es necesario disponer de gran capacidad de almacenamiento en el disco rígido. De esta manera, se disminuyen los costos en hardware.

- **Mejora del rendimiento:** dado que una menor cantidad de programas están siendo ejecutados localmente en la computadora, se consumirán menos recursos y la computadora tendrá un mejor rendimiento.

- **Costos reducidos de software:** gran cantidad de servicios en la nube son gratuitos. Los que no, generalmente se adquieren mediante mecanismos de suscripción, de manera mensual o anual, pudiendo ser cancelados en cualquier momento. En cambio, las licencias de los programas que se ejecutan localmente suelen ser más costosas y no están disponibles bajo mecanismos de suscripción.

- **Actualizaciones de software instantáneas:** otra de las ventajas del software en la nube es que ya no se enfrenta a la elección entre software obsoleto y los altos costos de actualización. Cuando la aplicación está basada en la web, las actualizaciones desde el punto de vista del usuario, se hacen en forma automática y simplemente estarán disponibles la próxima vez que acceda al servicio.

- **Aumento de la fiabilidad de los datos:** al trabajar localmente en la computadora, un fallo en el disco rígido puede provocar que se pierdan los datos del usuario. Las aplicaciones en la nube cuentan con sofisticados mecanismos de protección de los datos, copias de seguridad automáticas y datos replicados en diferentes centros de cómputos, lo que garantiza la integridad y disponibilidad de la información.

- **Facilitar la colaboración en grupo:** múltiples usuarios pueden colaborar fácilmente en documentos y proyectos. Debido a que los documentos están alojados en la nube, no en equipos individuales, cualquier persona con acceso a Internet puede trabajar colaborativamente con su grupo de trabajo. Esta es una de las características más destacables de la computación en la nube, permitiendo prácticas como el “trabajo a distancia”, en ciertos tipos de profesiones.

- **Independencia del dispositivo:** una persona puede cambiar de dispositivo y las aplicaciones y los datos seguirán estando en la nube. Puede utilizar un dispositivo portátil, y sus aplicaciones y datos estarán todavía disponibles. No hay necesidad de utilizar un dispositivo específico, un sistema operativo especial o un programa para ese dispositivo en particular. Sus documentos, datos y aplicaciones son los mismos sin importar qué computadora o dispositivo que esté utilizando para acceder a ellos.

## Desventajas

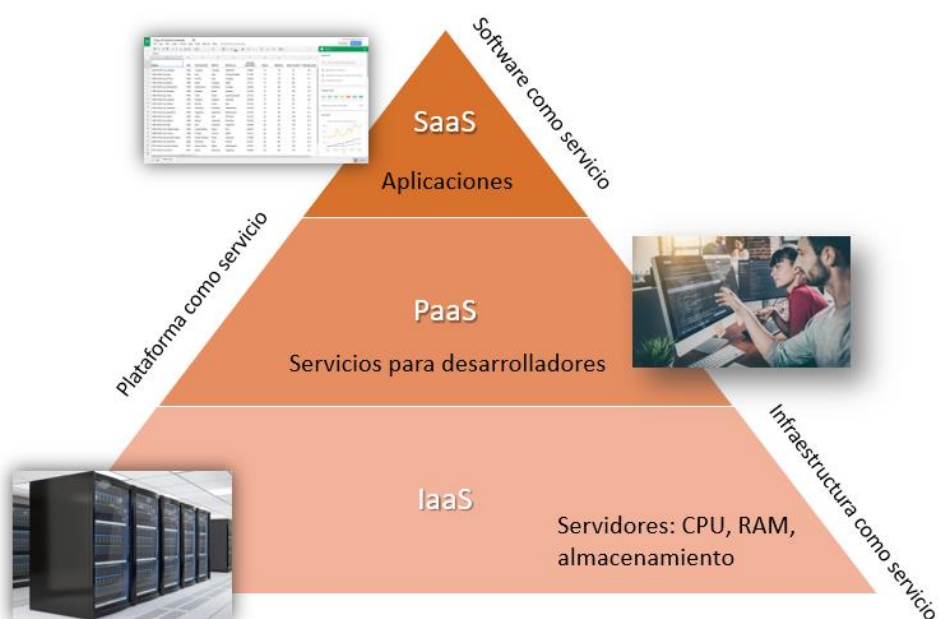
- Se requiere una conexión a Internet permanente
- No se tiene control sobre la disponibilidad y calidad del servicio
- Preocupación sobre la seguridad de los datos

**-Se requiere una conexión permanente a Internet:** la computación en la nube es imposible si no se puede conectar a Internet, dado que las aplicaciones y los datos del usuario se encuentran en servidores en la nube. Esto genera que cada vez exista una dependencia mayor con Internet. Afortunadamente, en los últimos años ha mejorado notoriamente la calidad y disponibilidad de las conexiones a Internet, sobre todo en dispositivos móviles, haciendo que esta desventaja sea cada vez menos relevante.

**-El usuario no tiene control sobre la disponibilidad y calidad del servicio:** cuando una persona utiliza aplicaciones instaladas localmente en su computadora, sabe que siempre estarán disponibles para su uso. En cambio, con los servicios en la nube, las decisiones sobre disponibilidad dependen del proveedor del servicio. Puede ocurrir que una aplicación no esté disponible para su uso porque se está realizando mantenimiento de la misma. Sin embargo, las empresas que proveen servicios en la nube suelen informar por adelantado a los usuarios cuando una aplicación no va a estar disponible, y además las ventanas de mantenimiento suelen ocurrir en horarios que no afecten a la mayoría de sus usuarios.

**-Preocupación sobre la seguridad de los datos:** la seguridad de los datos en la nube depende totalmente de las empresas que proveen servicios. Existen normas y políticas sobre cómo almacenar la información, con las cuales dichas empresas deben cumplir, pero uno podría no tener la certeza que siempre se cumplan. Almacenar los datos localmente les da control a los usuarios sobre los mecanismos de seguridad que quieran aplicar.

### ➤ Clasificación y ejemplos



Existen diferentes niveles de servicios que se pueden ofrecer en la nube. Cada nivel incluye un tipo de aplicaciones, generalmente dirigida a un tipo de usuario en particular. Se podrían representar dichos niveles en una pirámide, donde en la base se encuentran los servicios complejos asociados a proveer hardware para computación y almacenamiento, más arriba servicios para programadores y en la parte superior los servicios que están dirigidos a usuarios finales.

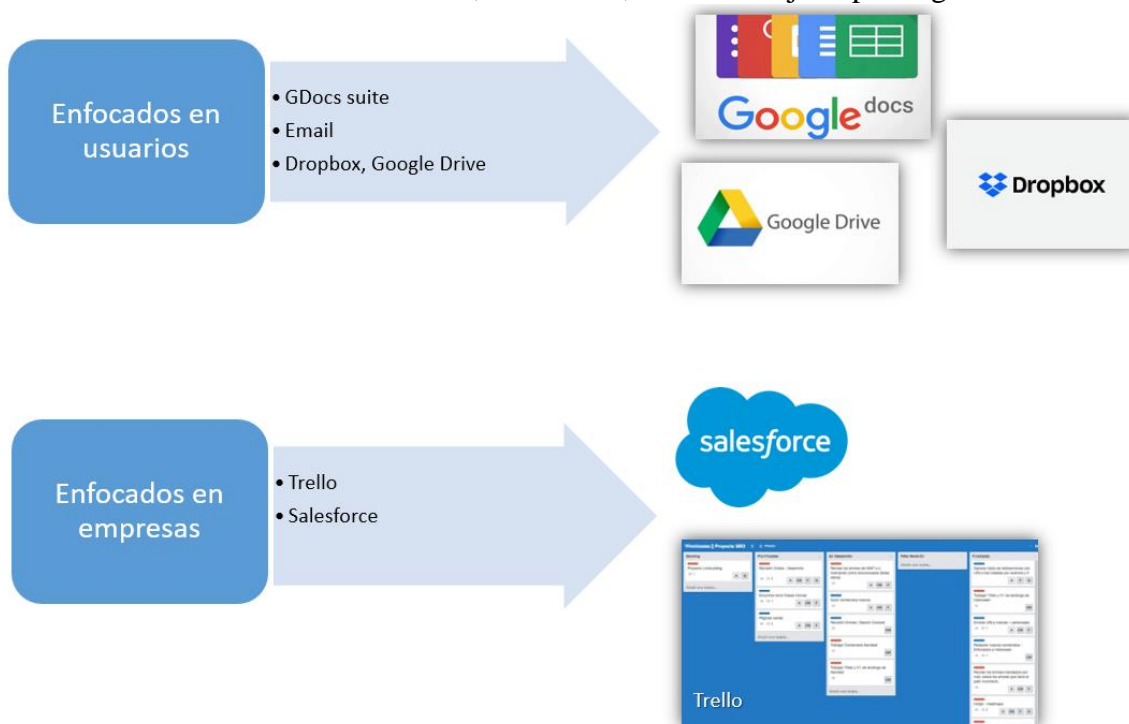


**Software como Servicio (SaaS – Software as a Service):** los usuarios utilizan las aplicaciones del proveedor que se ejecutan en una infraestructura de nube. Las aplicaciones son accesibles desde varios dispositivos a través de una interfaz de cliente ligera, como un navegador web o una interfaz de programa. El usuario no gestiona ni controla la infraestructura de nube subyacente incluyendo la red, servidores, sistemas operativos, almacenamiento e incluso la configuración de la aplicación.

Los ejemplos de SaaS pueden ser cotidianos para la mayoría de nosotros: sistemas de almacenamiento en la nube (Google Drive, Dropbox), suites de ofimática online (Google Docs, Office 365) o incluso servicios de email como Outlook o Gmail.

Además, existen servicios utilizados en las empresas, por ejemplo Trello, que permite organizar las tareas de un equipo.

Muchas empresas utilizan sistemas CRM (*Customer Relationship Management*) para gestionar a sus clientes. Uno de los más conocidos, Salesforce, funciona bajo el paradigma SaaS.





**Plataforma como Servicio (PaaS – Platform as a Service):** estos servicios están orientados a programadores y empresas que desarrollan software. Los usuarios pueden desplegar las aplicaciones creadas en la infraestructura alojada en la nube. El usuario no administra o controla la infraestructura de la nube incluyendo la red, servidores, sistemas operativos, o el almacenamiento, pero tiene control sobre las aplicaciones desplegadas y sobre los ajustes de configuración del entorno donde sus aplicaciones están siendo ejecutadas.

Las principales características asociadas a PaaS como solución de Cloud Computing se exponen a continuación:

- Facilita el despliegue de las aplicaciones del cliente, sin el costo y la complejidad derivados de la compra y gestión del hardware y de las capas de software asociadas.
- Permite que las aplicaciones o servicios alojados en la nube sean accesibles desde cualquier parte del mundo, sin que los usuarios tengan que preocuparse de toda la tecnología necesaria para que ello ocurra (servidores, dispositivos de red, etc.).
- Los servicios PaaS ofrecen interfaces (gráficas o de consola) para que los desarrolladores puedan gestionar fácilmente los recursos que utiliza su aplicación (servidores, contenedores, bases de datos, etc.).

Ejemplos:

- Amazon Web Services (AWS)
- Microsoft Azure
- Google Cloud Platform



**Infraestructura como Servicio (IaaS – Infrastructure as a Service):** en este tipo de servicio, el usuario/empresa está comprando el acceso al hardware de computación en bruto través de la red, como puede ser servidores o almacenamiento. Generalmente, se basa en el paradigma “pago bajo demanda”, en el cual el usuario paga por una capacidad de procesamiento/almacenamiento determinada, que puede ir variando según la demanda y el uso.

La ventaja de estos servicios es que el usuario/empresa puede evitar la complejidad y el costo de comprar y gestionar sus propios equipos físicos e infraestructura del centro de cómputos. Puede pagar por un re-

curso IaaS solo durante el tiempo que lo necesite.

A diferencia de PaaS, en IaaS el usuario tiene total libertad para gestionar el software (sistemas operativos, aplicaciones, etc.) que se ejecuta sobre la infraestructura por la cual está pagando.

Ejemplos:

- Digital Ocean: ofrece servicios de almacenamiento y bases de datos.



- Rackspace: ofrece servidores en la nube.
- Google Compute Engine: ofrece máquinas virtuales en la nube.

