Resumen Teórico

Gestión de Datos – 2C – 2021

Unidad 1: "Grafos"

Concepto

- Un grafo es una pareja G = (V, A), donde V es un conjunto de puntos, llamados vértices, y A es un conjunto de pares de vértices, llamadas aristas. Grafo es conjunto de vértices y arcos que se relacionan.
- Componentes (Teoría de Grafos Computacional):
    - Nodos: son los vértices
    - Relaciones: son las aristas o arcos
    - Grado: cantidad de arcos que salen de un vértice (grado positivo), cantidad de arcos que llegan (grado negativo).

Objetivo

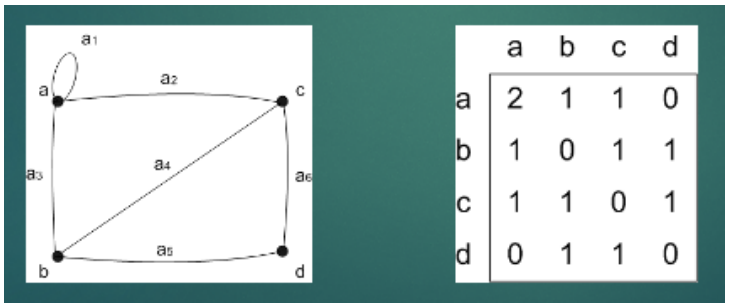
- Los grafos son estructuras abstractas, es decir, que no existen realmente, sino que modelizan virtualmente un problema real.
- Los grafos son utilizados para representar relaciones entre distintos elementos.

Representación Computacional

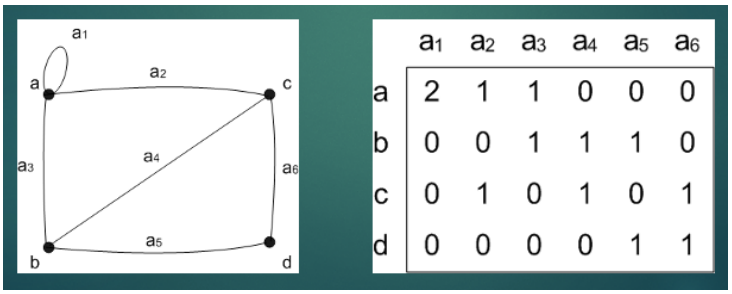
- **Estatica**: se construyen sobre estructuras computacionales rigidas. Representación de las relaciones de un grafo mediante una matriz. se denomina estática, cuando el espacio consumido para representar computacionalmente al grafo es invariable y fijo respecto a la cantidad de nodos y vértices a representar, esto es que son consideradas todas las ocurrencias de relaciones que puedan producirse entre todos los nodos, reservando el espacio para dicha ocurrencia potencial.
- **Dinámica**: se caracterizan por acompañar la dinámica del grafo, esto es que el espacio utilizado por la representación va cambiando en función de como va cambiando el grafo. No consideran todas las posibilidades de relación posible, sino que solo se representa lo que ocurre en ese momento.

Representación Estática

- **Matriz de adyacencia**: se asocia cada fila y cada columna a cada nodo del grafo, siendo los elementos de la matriz la relación entre los mismos, tomando como valores la cantidad de aristas que los unen y 0 en caso contrario.
- **Matriz de incidencia**: se asocia cada fila con un nodo y cada columna con una arista del grafo, siendo los elementos de la matriz la relación un 1 si dicho nodo es incidente con dicha arista, y 0 en caso contrario.



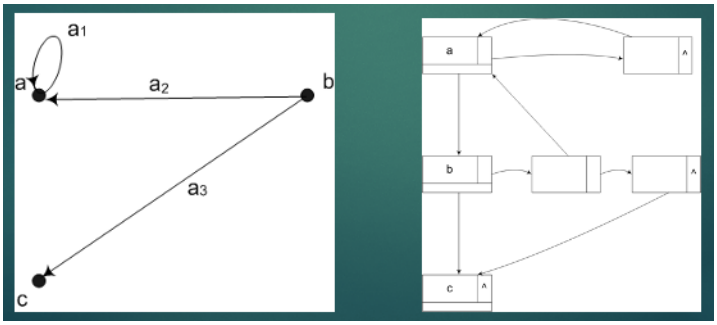
	a	b	c	d
a	2	1	1	0
b	1	0	1	1
c	1	1	0	1
d	0	1	1	0



	a_1	a_2	a_3	a_4	a_5	a_6
a	2	1	1	0	0	0
b	0	0	1	1	1	0
c	0	1	0	1	0	1
d	0	0	0	0	1	1

Representación Dinámica

- **Listas de adyacencia**: se asocia a cada nodo del grafo una lista que contenga todos aquellos nodos que sean adyacentes a el.



Caracterización

- **Grafo Libre**: es el grafo en el cual no existen arcos, o sea, que todos los vértices son aislados.
- **Grafo Completo**: es el grafo en el cual cada vértice esta conectado a todos los vértices que componen el grafo, incluido el mismo.
- **Grafo Regular**: grafo es regular de determinado grado g, si cada vértice tiene grado g, o sea que todos los vértices tienen el mismo grado g.
- **Grafo Simple**: un grafo es simple si a lo sumo un arco une dos vértices cualesquiera, esto es, que existe solo una arista que une a dos vértices específicos.
- **Grafo Complejo**: inverso al grafo simple. Es aquel donde puede existir mas de un arco que vincule dos vértices cualesquiera.

- **Grafo Conexo**: un grafo es conexo si para cualquier par de vértices existe al menos un camino posible entre ellos.
- **Grafo No Conexo**: se considera no conexo a un grafo donde un grupo de vértices no esta conectado con el resto de los vértices. Inverso a conexo.
- **Grafo Complementario**: dos grafos son complementarios si están compuesto por los mismos vértices, y el conjunto de aristas son todas aquellas que le faltan al "padre" para ser un grafo completo.

Clasificación

- **Según la presencia de dirección / el tipo de relación que implementan**:
    - **Dirigidos**: son aquellos en los cuales loas arcos que vinculan a los vértices tienen una dirección definida, la cual marca una jerarquía en la relación modelizada. Cuando tienen sentido.
    - **No Dirigidos**: son aquellos donde los arcos no tienen una dirección definida que marque propiedad en la relación modelizada. Cuando no tienen sentido.
- **Según restricciones de sus relaciones**:
    - **Restrictos**: grafos en los cuales la relación que se modela no debe cumplir las propiedades de reflexividad, simetría y transitividad. Debe ser Anti-equivalente.
    - **Irrestrictos**: grafos en los cuales no se aplica ninguna restricción a la relación.

Camino, Paso y Ciclo

- **Camino**: un camino entre dos nodos se establece cuando existe una vinculación directa o indirecta entre ambos. Es decir, vincular mediante arcos, independientemente del sentido.
- **Paso**: un paso entre dos nodos se produce cuando existe un camino entre ambos, pero con sentido preestablecido. Partiendo de A se llega a B.
- **Ciclo**: un ciclo entre dos nodos es un paso o un camino donde el origen y el destino son iguales.

Búsqueda

- **Búsqueda en Profundidad**: esta técnica se caracteriza por avanzar en profundidad, es decir sin mantener un orden jerárquico de evaluación. Se van abriendo y abriendo en profundidad, luego se vuelve y se va haciendo lo mismo.
- **Búsqueda en Anchura**: evalúa todos los destinos de todos los arcos que parten del vértice origen.

Unidad 1: “Estructura de Datos”

Concepto

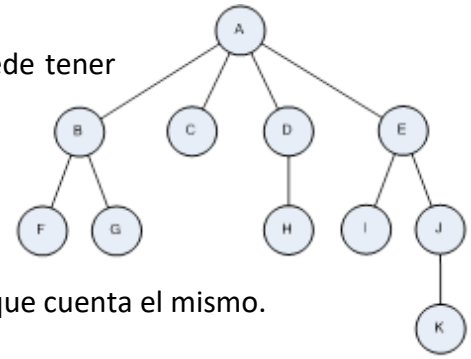
- Una estructura de datos es un grafo dirigido y restricto, con las características de unicidad en sus relaciones, esto es que en orden de predecesor, cada nodo solo puede tener un nodo predecesor a el.
- Estructuras de datos son utilizadas para modelar problemas reales al igual que los grafos. Pero porque son grafos restrictos unívocos, se simplifica su administración.

Clasificación

- **Estructuras Biunívocas**: son univocas en ambos sentidos de la relación manteniendo uno o ningún predecesor y uno o ningún sucesor.
    - **Listas**: estructura de datos que tiene una dinámica abierta, esto es que dentro de una lista, a la hora de realizar un alta se debe recorrer toda la lista.
        - **Lineal**: ultimo puntero del nodo a NULL.
        - **Circular**: ultimo puntero apunta al primero.
        - **Doblemente Enlazada**: todos los nodos apuntan al siguiente y al anterior.
    - **Pilas**: estructura de datos que su dinámica de ingreso y egreso es de tipo LIFO. Por mismo extremo se hace ingreso y egreso.
    - **Colas**: estructura de datos que se caracteriza por privilegiar el orden y la jerarquía en una estructura manteniendo una dinámica FIFO.
- **Estructuras Univocas**: son solo univocas manteniendo un solo predecesor, pero pudiendo tener mas de un sucesor.
    - **Arboles**: la unicidad se cumple en un solo sentido. Un solo predecesor, pero varios sucesores. Tienen grado.

Unidad 1: “Arboles”

Conceptos



- **Grado**: el grado es la máxima cantidad de hijos o subárboles que puede tener cada nodo.
- **Nivel**: el nivel es la posición en la que se encuentra cada nodo con respecto a la raíz de este. La raíz es nivel 0.
- **Profundidad**: la profundidad de un árbol es la cantidad de niveles con que cuenta el mismo.

Representación Computacional

- **Estática**: un árbol se representa en forma estática a través de un vector.
- **Dinámica**: un árbol se representa en forma dinámica a través de un conjunto de nodos de igual tipo vinculados entre si a través de punteros o links.

Características

- **Completo**: un árbol completo es aquel en el cual todos los nodos cumplen el grado o son hojas.
- **Balanceado**: un árbol esta balanceado si todos los subárboles desde la raíz pesan lo mismo, o sea tienen la misma cantidad de elementos.
- **Perfectamente Balanceado**: un árbol esta perfectamente balanceado cuando esta balanceado en todos sus niveles.

Crecimiento

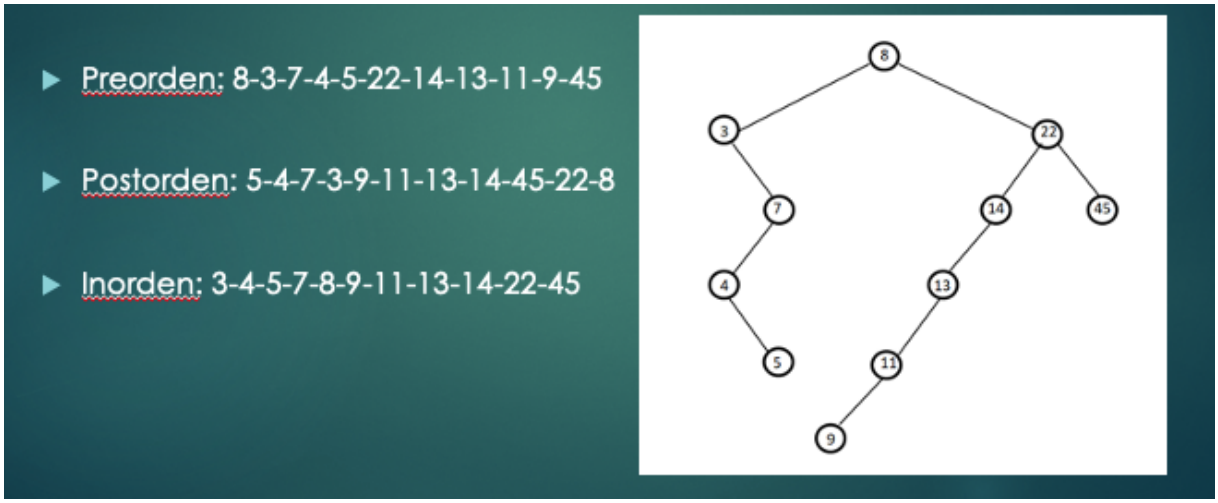
- El crecimiento de un árbol es exponencial en función del grado de este, o sea, que en cada nivel puede crecer en función del grado definido.
- Max Elementos = grado elevado a los niveles - 1

Búsqueda

- Búsqueda en arboles se realizar por niveles y no por elementos. Necesitamos tantas preguntas como niveles tenga el árbol.
- En un árbol la búsqueda es logarítmica. Comparado con una estructura de datos básica donde es lineal.

- **Árbol Binario de Búsqueda**: árbol de grado 2 diseñado para buscar como método alternativo a una lista, donde los elementos menores se ingresan a la izquierda y los mayores a la derecha.

Barridos

- Un barrido de un árbol es la forma de leer el mismo.
    - **Preorden**: el nodo se lee apenas se llega al mismo.
    - **Postorden**: el nodo se lee cuando se va del mismo y no se va a regresar.
    - **Inorden**: el nodo se lee cuando se cambia de rama en el árbol.
- 

Árbol de Expresión

- Una expresión puede representarse y resolverse a partir de un árbol, a estos se los llama arboles de expresión. (Expresión = cuenta matemática)
- Si el árbol se barre inorden se obtiene la expresión matemática en notación Infijo, si se lo barre postorden se obtiene en notación polaca inversa o postfijo.

Unidad 2: “Métodos de Clasificación”

Concepto

- **Objetivo**: dado un conjunto de valores desordenados, devolver un conjunto ordenado de menor a mayor o de mayor a menor.
- **Registros**: los números a ordenar no son valores aislados, si no que son parte de una colección de datos denominada registro. Cada registro contiene una clave que es el valor a ser ordenado, y el resto del registro contiene los datos satélites.
- **Estabilidad**: ordenamiento se considera estable si mantiene el orden relativo que tenían originalmente los elementos con claves iguales.

Clasificación

- **In Situ**: transforman una estructura de datos usando una cantidad extra de memoria, siendo esta pequeña y constante. Los que no son “in situ” utilizan una gran cantidad de memoria extra para transformar una entrada.
- **Métodos Internos**: si el archivo a ordenar cabe en memoria principal.
- **Métodos Externo**: si el ordenamiento del archivo se realizar desde un disco u otro dispositivo que no es memoria principal.

Complejidad

- Estudia la complejidad de ejecutar un algoritmo. La clase de complejidad “P” es el conjunto de problemas de decisión que pueden ser resueltos en tiempo polinómico. Los “NP” no.
- El orden de complejidad se describe como O(función).
- Para evaluar complejidad en un algoritmo se evalúan principalmente la cantidad de comparaciones realizadas.

Métodos

- **Bubble Sort**: método mas simple, pero a la vez mas lento. Realizar pasadas sobre los datos, donde en cada paso, los elementos adyacentes son comparados e intercambiados si es necesario. Peor tiempo de ejecución ya que en el peor de los casos es de $O(n^2)$.

- **Selection Sort**: busca el elemento mas pequeño del array y lo intercambia con el que esta en primera posición. Luego busca el segundo mas pequeño y lo coloca en segunda, y así continua hasta que este todo el array ordenado. Bueno para archivos con registros grandes y claves pequeñas. Orden de complejidad $O(n^2)$.
- **Insertion Sort**: idea de ordenamiento parcial, considera que todo a la izquierda ya esta ordenado. Elige un elemento y lo saca, y mueve todos los elementos restantes hacia la derecha. Va pasando y cuando encuentra elemento mas pequeño, lo intercambia con el sacado. Orden de complejidad $O(n^2)$.
- **Shell Sort**: toma números a una distancia n, y va intercambiando estos números para ordenarlos empezando por los que se encuentran mas lejanos.
- **Merge Sort**: partiendo desde dos listas, voy tomando el primero de cada una y voy moviendo hasta ordenar en una nueva lista. Primero divide la lista en 2. Es un algoritmo recursivo.
- **Quick Sort**: algoritmo consiste en 4 pasos: primero elige un elemento como pivote, después compara todos los elementos con el pivote generando dos conjunto mayores y menores.
- **BSort**: variante de Quick Sort donde el funcionamiento del método es igual pero solo cambia la elección del pivote y en este caso es el elemento central.
- **MeanSort**: variante de Quick Sort donde el funcionamiento del método es igual pero solo cambia la elección del pivote que en este caso es el elemento mas próximo a la media.
- **Heap Sort**: se aplica sobre arboles de orden parcial cargados en un vector. Agarra el primero con el ultimo y los va intercambiando hasta tener todo ordenado.

Nombre	Mejor caso	Caso medio	Peor caso	Estable	Comentarios
Bubble Sort	$O(n)$	$O(n^2)$	$O(n^2)$	Si	El más lento de todos. Uso pedagógico.
Selection Sort	$O(n^2)$	$O(n^2)$	$O(n^2)$	Si	Apto si queremos que consumir siempre la misma cantidad de tiempo.
Insertion Sort	$O(n)$	$O(n)$	$O(n^2)$	Si	Conveniente cuando el array esta casi ordenado.
Shell Sort	$O(n^{5/4})$	$O(n^{3/2})$	$O(n^2)$	No	Dependiente de la secuencia de incrementos.
Merge Sort	$O(n\log n)$	$O(n\log n)$	$O(n\log n)$	Si	Adecuado para trabajos en paralelo.
Heap Sort	$O(n\log n)$	$O(n\log n)$	$O(n\log n)$	No	El método acotado en el tiempo muy utilizado para grandes volúmenes de datos
QuickSort	$O(n\log n)$	$O(n\log n)$	$O(n^2)$	No	El más rápido en la práctica. Implementado en gran cantidad de sistemas.

Unidad 3: "Índices"

Concepto - Objetivo

- El objetivo es crear una estructura adicional a la tabla que permita mantenerlos los datos ordenados en función de alguna clave.
- El índice puede formar parte de la tabla o ser una estructura adicional a la tabla sobre la cual se crea, generando espacio adicional y la necesidad de incorporar y eliminar valores en ambos sitios.

Accesos

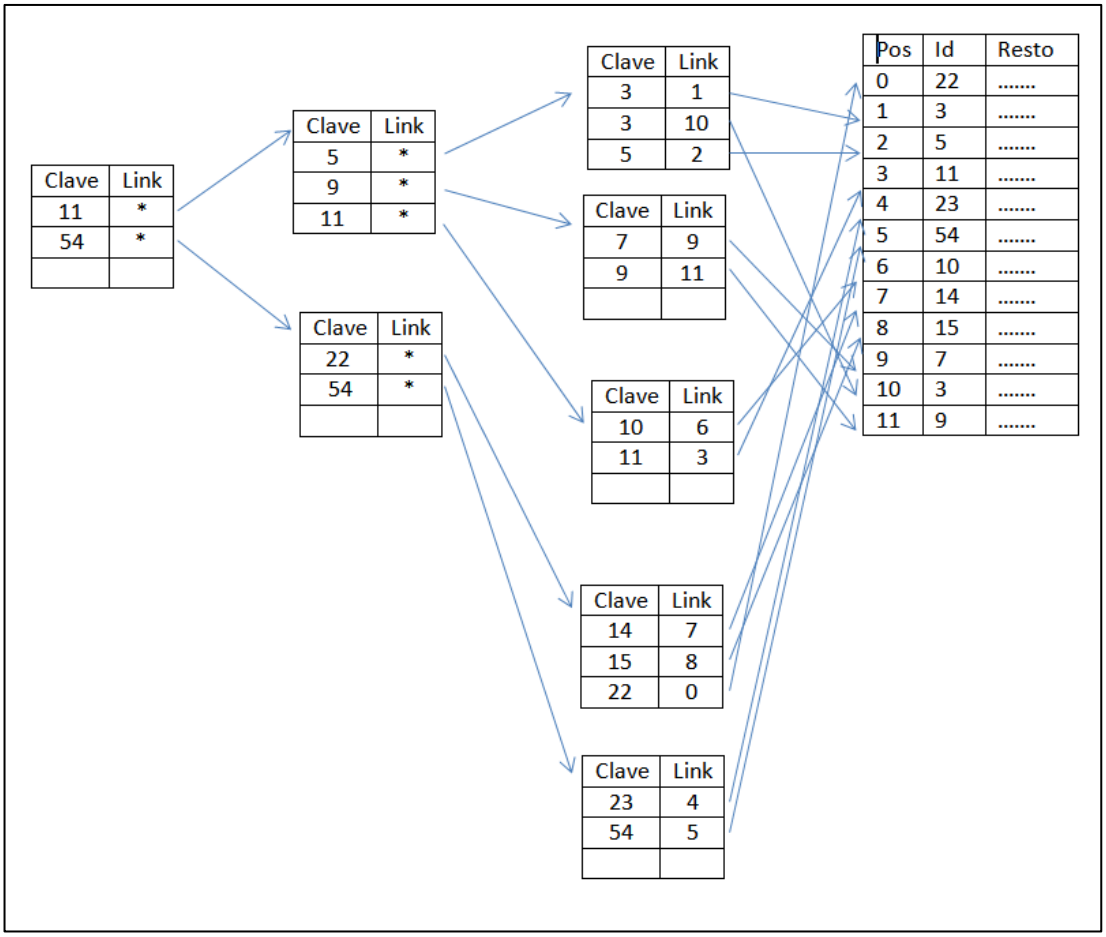
- Existen diversas formas de acceder a los datos:
    - **Secuencial**: el acceso se realiza en función al modo en que ingresaron los datos.
    - **Secuencial Indexado**: el acceso se realiza en función de alguna clave que fue definida.
    - **Directo o Random**: el acceso es en forma directa a una clave sin realizar ningún recorrido.

Métodos - Hashing

- **Hashing**: trabaja sobre el concepto de una tabla y una función hash. Dicha función se utiliza para convertir algún tipo de dato en un pequeño numero que puede servir como huella digital de ese dato.
- Las buenas funciones hash tienen las siguientes cualidades:
    - **Evitar Colisiones**: se cumple dado un conjunto de valores de entrada, la salida es distinta.
    - **Tiende a distribuir las claves uniformemente**.
    - **Es fácil de calcular**.
- **Método**: crea una tabla donde en la primera dimensión coloca las claves y en la segunda las posiciones relativas. Función hash recibe como entrada la clave a almacenar y devuelve un valor numérico que corresponde a la posición en la cual debería ir dicha clave en la tabla.
- **Colisiones**: si una clave genera una posición en la tabla ya ocupada. Varias técnicas de resolución de colisiones:
    - **Encadenamiento**: cada casilla referencia a una lista de los registros insertados que colisionan en la misma casilla. Es decir, una lista en cada posición.
    - **Direccionamiento Abierto**: se busca otra posición dentro de la tabla. Existen varios métodos:
        - **Sondeo lineal:** busca hasta encontrar una posición vacía.
        - **Sondeo Cuadrático:** ubica colisión examinando cierta posición a una distancia espcif.
        - **Hashing doble:** aplica función hash por segunda vez a la clave, usando func. distinta.

Métodos – Arboles B

- Tipo de árbol M-ario destinado a la creación de índices físicos para el acceso a la información. Busca minimizar las operaciones de entrada y salida hacia el disco.
- **Tipos de nodo**:
  - **Nodo Hoja**: tiene un componente de dato donde van los valores de las claves ordenados de menor a mayor y un componente de puntero que contiene posición de los datos.
  - **Nodo Raíz**: tiene un componente de dato donde van los valores de las claves ordenados de menor a mayor, y un componente puntero que apunta al nodo que contiene claves menores o iguales.
- **Operaciones**:
  - **Búsqueda**: parecido a buscar en un árbol binario. Tomamos decisión multicamino en base al numero de hijos del nodo.
  - **Inserción**: comenzamos en la raíz y realizamos una búsqueda para el. Búsqueda sin éxito terminara en un nodo hoja. Si se llega a la hoja y no hay lugar, se hace un Split, proceso donde se divide en 2 dejando la mitad de los elementos en cada uno.
  - **Eliminación**: comenzamos en la raíz y realizamos la búsqueda. Si cuando se elimina el nodo queda vacío, se fusiona con nodo anterior.



Unidad 4: “Compresión de los Datos”

Tipos de Algoritmos

- **Con Pérdida**: son algoritmos que pierden información al comprimir y que por dicha perdida no son reversibles. Este tipo de algoritmos se utiliza para los archivos multimedia.
- **Sin Pérdida**: son algoritmos que comprimen archivos sin perder información, por lo cual son reversibles, dado que permiten volver al estado original del archivo. Se utilizan para todos los archivos menos los de multimedia.

Concepto de Compresión

- **Fundamento**: es posible comprimir un archivo, ya que el alfabeto ASCII fue creado de longitud fija de forma tal que por mas que un carácter se repita mucho mas que otro el espacio que ocupa es el mismo. Pero no todos los archivos utilizan los 256 caracteres especificados en el archivo ASCII, por lo cual se desperdician bits en su representación.
- **Compresión de Archivos** -> Algoritmo de Huffman
- **Compresión Multimedia** -> modificar su codificación, recodificando la resolución para disminuir su calidad utilizando menos bits.

Algoritmo de Huffman

- Algoritmo de compresión de datos sin perdida, que es muy eficiente con archivos de texto.
- El algoritmo busca identificar cada uno de los caracteres distintos en el conjunto de archivos a comprimir, y le asigna un código de longitud variable según la frecuencia. Cuantas mas veces aparezca un carácter en los archivos, la longitud de su código va a ser mas pequeña.
- **Proceso de Compresión**:
    1. Se lee el archivo y se identifican todos los caracteres distintos que lo componen.
    2. Caracteres identificados se almacenan conjunto a la cantidad de repeticiones en un vector.
    3. Luego se ordena el vector por la cantidad de repeticiones en forma descendente.
    4. Por ultimo, se crea un árbol binario dividiendo el vector de a dos en función de la cantidad de repeticiones de cada carácter.
    5. Luego por convención se define que la izquierda representa un 0 y derecha un 1-
    6. Se vuelve a leer el archivo y para cada carácter se busca en el árbol el código de bits que representará ese carácter.

- **Proceso de Descompresión**:
    1. Para descomprimir se parte el archivo comprimido y se realiza la lectura del mismo para obtener los caracteres.
- Es necesario guardar el vector en archivo comprimido, por ese motivo los archivos muy pequeños no se comprimen.

Unidad 5: “Tipos y Modos de Procesamiento”

Conceptos

- **Tipos de procesamiento**: es la forma en la que se procesaran los Request del usuario y el modo en que se diagrama el diseño de la aplicación a realizar en función de este tipo.
- **Modos de procesamiento**: es la forma mediante la cual se van a organizar los recursos y el procesamiento de los datos dentro del computador.

Tipos de Procesamiento

- **Batch**: es un procesamiento que se caracteriza por procesar por lotes de instrucciones.
    - Los lotes no tienen control ni supervisión directa del usuario. Ejecución no precisa interacción.
    - Tareas repetitivas sobre grandes conjuntos de información.
    - Especifican su funcionamiento mediante scripts.
- **Tiempo Real**: es un procesamiento que se realiza en función de los pedidos del usuario en el momento en que fueron realizados sobre los datos reales.
    - Se recibe un proceso y se responde inmediatamente, es muy utilizado en la televisión en vivo.
    - Constituido para aplicaciones muy especificas.
- **Interactivo**: es un procesamiento que para el usuario se realiza en tiempo real, pero realmente se esta ejecutando en forma asincrónica.
    - Proceso que requiere la interacción con el usuario.
    - Inicialmente soportado por grandes computadoras por un proceso “Time Sharing”.
    - Modo de procesamiento es mezcla entre tiempo real y batch. Persona piensa que esta ejecutando en tiempo real pero en realidad es batch.
    - **Time Sharing**: uso de tiempo compartido de CPU en forma concurrente entre muchos usuarios.

Modos de Procesamiento

- **Centralizado**: es el modo de procesamiento que concentra todos los recursos en un computador central.
    - Todos los recursos o programas se encuentran en un computador.
    - El procesamiento se realiza solo en un lugar y recursos encapsulados.
    - Usuarios se conectan desde computador sin recursos.
- **Descentralizado**: los recursos físicos se descentralizan en diferentes computadoras interconectadas entre si.
    - Todos los recursos están distribuidos en diferentes computadoras.
    - Procesamiento se realiza en diferentes equipos que componen la red.
    - Usuarios se conectan con computadoras con recursos propios.
- **Distribuido**: es similar al descentralizado, pero donde los recursos se distribuyen por funcionalidad o ubicación geográfica.
    - **Funcional**: se produce distribución en forma funcional, es decir por una funcionalidad.
    - **Geográfica**: se produce una distribución en forma geográfica.
- **Satelital**: es el modo de procesamiento que funciona en forma aislada sin una red de computadoras ni conexión externa.
    - Procesamiento “Stand Alone”, donde una computadora esta aislada del resto y procede en forma mono usuario.
    - Comunicación mediante dispositivos externos y periféricos.
    - Muy utilizada en el esquema de retail.

Unidad 5: “Estructura de Data Base Managment System”

Conceptos

- **DB**: conjunto de datos interrelacionados que se ajustan a una serie de modelos preestablecidos que recogen información de interés de objetos del mundo real.
- **DBMS**: software encargado de gestionar los datos de la DB. Proporciona mecanismos de acceso a los datos para almacenar, definir y recuperar información de forma eficiente.

Propiedades (ACID)

- **A**: Atomicidad es la propiedad que asegura que una operación se ha realizado o no, y por lo tanto ante un fallo del sistema no puede quedar a medias. Es imposible de separar en pasos intermedios.
- **C**: Consistencia es la propiedad que asegura que solo se empieza aquello que se puede terminar. Solo se ejecuta lo que no rompe con las reglas de integridad de la base de datos.
- **I**: Aislamiento es la propiedad que asegura que una operación no puede afectar a otras. Cada transacción es independiente de las demás.
- **D**: Durabilidad es la propiedad que asegura la persistencia de los datos, es decir que una vez realizada una operación no se podrá deshacer aunque falle el sistema y los datos sobreviven.

Arquitectura ANSI

- Arquitectura de una DBMS se compone de tres niveles denominados como “capas”. Arquitectura mas utilizada.
- **Capas**:
    - **Interno o Físico**: forma en que la base de datos representa físicamente los datos.
    - **Conceptual o Lógico**: datos que se almacenan y como están relacionados entre si.
    - **Externo o de Usuario**: describe parte de la base que es relevante para el usuario.
- **Características**:
    - Permite vistas de usuario independientes y personalizadas.
    - Oculta los detalles físicos de almacenamiento a los usuarios.
    - El administrador de la base de datos debe ser capaz de cambiar las estructuras.
    - Estructura interna de la base de datos no debería verse afectada por cambios en los aspectos físicos de almacenamiento.

- **Componentes**:
    - **IPL**: "*initial program loader*" es un programa de carga inicial que permite levantar el servicio del DBMS.
    - **User Manager**: es el modulo encargado de manejar los perfiles, usuarios y roles de acceso.
    - **File Manager**: es el modulo encargado de administración lógica de los archivos que la componen.
    - **Disk Manager**: es el modulo encargado de la administración física de la información persistida.

Nivel Interno de Almacenamiento

- **Segmentación**: divide la memoria en segmentos, cada uno de los cuales tiene una longitud variable.
- **Paginación**: divide la memoria en paginas, cada una de las cuales es de longitud fija.

Lógica de Almacenamiento

- **Clustering**: técnica de agrupamiento que permite unificar objetos en función de algún criterio establecido.
    - **Intra File**: los objetos se agrupan en función de la pertenencia a un conjunto predeterminado. Usado para los datos secuenciales.
    - **Inter File**: los objetos se agrupan en función a la relación existentes entre los objetos independientemente que pertenezcan a diferentes conjuntos. Usado para los índices y PK asociadas a las FK que existan.

Almacenamiento de Archivos

- Dado que hay un solo formato de archivos, es necesario identificar de alguna forma el contenido del mismo para poder tipificarlo y administrarlo en forma diferencial.
- **Header**: es la cabecera de un archivo siendo el conjunto de caracteres que se colocan al inicio del mismo y que permiten definir el contenido que continua.
- **Extensión**: esta relacionada con su tipología y de hecho con la aplicación destinada a su apertura y administración.

Unidad 6: “Inteligencia de Negocios y Tecnologías OLAP”

Inteligencia de Negocios

- Consiste en la transformación de datos en información y esta en conocimiento, con la intención de mejorar al máximo el proceso de toma de decisiones de la organización.
- Conjunto de metodologías, herramientas y estructuras de almacenamiento.

Datos, Información y Conocimiento

- **Dato**: mínima unidad semántica de información que por si mismo no tiene ningún valor. Para brindar información necesitan que se los vincule con alguna relación. Provienen de manera interna o de manera externa. Pueden ser objetivos o subjetivos, y de tipo cualitativo o cuantitativo.
- **Información**: conjunto de datos procesados o relacionados con un significado especifico. Si a los datos se les añade relaciones, se convierten en información.
- **Conocimiento**: es la fusión de valores, información y experiencia. Es el marco conceptual adecuado para la incorporación de nueva información. Para que la información se convierta en conocimiento, se deben llevar adelante acciones de búsqueda de conexiones.

Tecnologías OLAP y OLTP

- Evolución de la informática trajo con si mismo la acumulación masiva de datos. Hace falta un modo de estructurar la información y los datos que aporte una nueva perspectiva sobre los mismos.
- **OLAP**: “Online Analytical Processing”, llamado Modelo Relacional, debido a que analiza y relaciona la información analizada.
- **OLTP**: “Online Transaction Processing”, llamado Modelo Transaccional, debido a que se basa en la ejecución de un conjunto de transacciones para obtener el resultado esperado.

Tecnología OLTP

- Están caracterizadas en que muchos usuarios crean, actualizan o retienen registros individuales. Entonces las bases OLTP están optimizadas para las actualizaciones de las transacciones.
- **Características**:
    - Su ejecución se basa en transacciones
    - Conforman el 99% de los sistemas existentes

- Son sistemas operativos
- Procesan datos
- Los datos se almacenan normalizados
- Registran datos nivel de detalle de cada transacción
- Los datos son volátiles

Tecnología OLAP

- Son usadas por analistas y gerentes que quieren vistas de alto nivel de los datos.
- Las bases de datos OLAP son actualizadas por bloques, generalmente de múltiples fuentes, y provee poderosas aplicaciones multiusuario de poder analítico. Bases optimizadas para el análisis.
- Herramientas OLAP buscan ser compatibles con otras herramientas de análisis.
- Datos provienen de otros sistemas y aplicaciones, pero son capturados por la aplicación OLAP. Se deben duplicar y almacenar en otras fuentes para el análisis de manera activa.
- **Estructura Modelo OLAP**:
  - Ejecución: aplicaciones de gran tamaño y utilizadas para el análisis de futuros inciertos. Se deben tener duplicidad de los datos para rápido acceso.
  - Múltiples fuentes de datos
  - Filtrado de datos
  - Ajuste y modificación de datos
  - Actualización y consistencia de datos
  - Historia de los datos
  - Distintas perspectivas o vistas
  - Actualización de datos
- **Características**:
  - Ejecución basada en el análisis
  - Conforman el 1% de los sistemas existentes
  - Son sistemas para la toma de decisiones
  - Procesan información
  - La información se almacena desnormalizada
  - Registran información global por patrones conocidos como dimensiones
  - La información es persistente

Bases de Datos Multidimensionales

- Las dimensiones determinan la estructura de la información almacenada y definen adicionalmente caminos de consolidación.
- La información almacenada se presenta como variables que a su vez están caracterizadas por una o mas dimensiones.
- La información puede analizarse dentro del cubo formado por la intersección de las dimensiones de la variable particular.

Dispersión de Datos

- **Definición**: A medida que se agregan dimensiones a una Base de Datos Multidimensional, el numero de celdas crece rápidamente. Lo que sucede es que las bases de datos empiezan a tener un gran numero de celdas vacías, que se denomina dispersión de datos.
- **Hipercubo**: la información se guarda implícitamente en un gran y único cubo, presentando los datos al usuario en un formato de hipercubo, donde todos los datos en la aplicación aparecen como una sencilla estructura multidimensional.
- **Multicubo**: la información se almacena dividiendo los datos en grupos mas pequeños y densos, donde la base de datos multidimensional consiste en un numero de objetos separados normalmente con diferentes dimensiones.

Base de Datos Relacional vs Base de Datos Multidimensional

	Base de Datos Relacional	Base de Datos Multidimensional
Deposito de datos, acceso y visión	Relacional Tablas de Columnas e hileras Lenguajes SQL con ampliaciones Herramientas de terceros que usan API	Dimensional Arreglos: Hipercubo, Multicubo Tecnología de matriz dispersa Propietario de hoja de calculo
Utilización e incorporación	OLTP Motor RBDMS Profundización a nivel de detalle Desempeño de consultas: rango amplio	OLAP Motor multidimensional Profundización a nivel de resumen/adición Desempeño de consultas: rápido
Tamaño y actualización de bases de datos	Gigabytes a terrabytes El deposito de indices y el retiro de normas que incrementan tamaño Consulta y cargas paralelas Actualizacion durante uso	Gigabytes Compresion y adicion de datos dispersos Dificil actualizar durante uso; los cambios pequeños pueden requerir reorganizacion

Unidad 6: “Data Warehouse”

Concepto y Objetivo

- La inteligencia de negocio consiste en la transformación de datos en información y, esta, en conocimiento, con la intención de mejorar al máximo el proceso de toma de decisiones de la organización.
- **Data Warehouse**: es una base de datos corporativa cuya característica principal es la integración y el filtrado de información de una o varias fuentes, que luego procesara para su análisis desde diferentes puntos de vista y con una gran velocidad de respuesta.
- Colección de datos históricos e integrados diseñada para soportar el procesamiento informático para la toma de decisiones estratégicas que no utilizan para la operatoria diaria.
- **Objetivo**: conversión de los datos de las aplicaciones del ambiente transaccional en datos integrados de gran calidad.

Características

- **Esta orientado a sujetos**: modelo operacional orientado a los sujetos mayores de la organización.
- **Es integrado**: datos se integran antes de ingresar en DW.
- **Es temático**: solo se añaden datos que se necesitan en el proceso de generación de conocimiento del negocio.
- **Es variante en el tiempo**: los datos en DW varía en el tiempo.
- **Es simple de manejar**: no se hacen updates de los datos, solo se cargan o acceden.
- **No es volátil**: almacén de información de DW se puede leer pero no admite ninguna modificación.

Funcionalidades de un DW

- **Acceso a Fuentes**: incluye los procesos que se aplican en las bases de datos fuentes a los datos que se transferirán.
- **Carga**: abarca los procesos de extracción, depuración, conversión y carga de los datos.
- **Almacenamiento**: abarca la arquitectura que se necesita para incluir varias vistas en DW. Modo en que se organizan los datos para su posterior muestra.
- **Consultas**: ambiente de consultas permite que el usuario dirija el análisis y la producción de reportes.
- **Metadatos**: toda información descriptiva sobre el contexto. Deben estar disponibles para el análisis.

Datos

- **Migración de Datos**
    - Trasladar los datos desde los sistemas seleccionados de origen hasta el stage de DW. Solo se moverán los datos solicitados por los usuarios.
    - Se deben migrar los datos referenciales y transaccionales. Mas importante es entender donde se ubicarán los datos y cuales se reubicarán.
- **Depuración de Datos**
    - Consiste en corregir para estandarizar el formato y completar cualquier valor requerido por DW.
- **Conversión de Datos**
    - El objetivo de la conversión de los datos es cambiar los datos con el formato y la estructura requeridos por el DW.
    - Proceso que reduce el numero de elementos de datos que se cargan desde stage a DW.
- **Carga de Datos**
    - La renovación completa comienza truncando las tablas en Data Warehouse y luego cargándolas con todos los datos requeridos.
    - La renovación incremental identifica los cambios que se produjeron en los datos origen desde la ultima vez que se cargo DW, y luego inserta, actualiza o borra registros como se solicite.
- **Conciliación de Datos**
    - Identifica los problemas de datos que si no se les diera importancia pasarían los controles de prevención. Provee veracidad e identifica datos que no concuerdan.
    - Se analiza la calidad de los datos como también la cantidad de datos.
    - **Tipos**:
        - **Conciliación Completa**: al finalizar cada proceso de carga se realiza una conciliación completa comparando información con sistema de origen.
        - **Conciliación Por Fase**: se realiza después de cada etapa del flujo del proceso de datos.

Data Marts

- Se denomina a las vistas multidimensionales de cada área.
- Dara Marts se ajusta mejor a las necesidades que tiene una parte especifica de un negocio, mas que a las de toda una organización.

- Data Marts deben su popularidad a que se disminuyen de manera significativa los costos asociados a su creación y operación.

Modelo Estrella

- Modelo de datos conformados por dos tipos de tablas, de hechos y las de dimensiones.
- **Tabla de Hechos**: registra medidas o métricas de un evento especifico, generalmente consisten en valores numéricos y claves foráneas que referencian a tablas de datos dimensionales que guardan información descriptiva.
- **Tabla de Dimensiones**: las dimensiones pueden definir una amplia variedad de características. Tablas tienen un bajo numero de registros comparada con la Tabla de Hechos, pero cada registro tiene numerosos atributos.

Unidad 6: “Data Mining”

Concepto

- Conjunto de técnicas que se utilizan para la obtención de la información implícita en grandes bases de datos. Exploran las bases en busca de patrones ocultos.
- Se encarga de buscar patrones de interés ocultos, que luego permiten la anticipación de futuros acontecimientos gracias a la predicción y el pronostico.

Características

- Bases de Datos de Gran Volumen:
    - **Gran cantidad de columnas**: mayor nivel de análisis y de detalle por gran cantidad de combinaciones.
    - **Gran cantidad de Filas**: disminuye la cantidad de error de estimación y desvíos. Más información histórica.
- Para que se pueda ejecutar y cumplir con su objetivo:
    - **Recolección de datos en gran escala**: unifica contenidos de todas las bases de datos disponibles.
    - **Alta Tecnología y gran almacenamiento**: al procesar muchos datos, se debe contar con veloces procesadores y gran capacidad de almacenamiento.
    - **Algoritmos de Data Mining**: funciona mediante la aplicación de diversas herramientas algorítmicas.
- Oportunidades de Negocio:
    - **Predicción automatizada de tendencias y comportamientos**
    - **Obtención automatizada de modelos previamente desconocidos**

Ventajas del DM

- Contribuye a la toma de decisiones estratégicas y proporciona un sentido automatizado para identificar información clave.
- Permite a los usuarios dar prioridad a decisiones y acciones e indica los factores que tienen una mayor incidencia, que segmentos no lo son y que unidades de negocio son sobrepasadas.
- Genera modelos descriptivos, permitiendo explorar, visualizar y comprender los datos, patrones y relaciones.
- Genera modelos predictivos, permiten ver relaciones no descubiertas e identificarlas.

Herramientas Algorítmicas

- **Redes Neuronales**
  - Modelos predecibles de características no lineales que aprenden a través del entrenamiento y semejan la estructura de una red neuronal biológica.
  - El entrenamiento de la red consiste en probar entradas que se propagan hasta que llega a la capa de salida que lo devuelve como solución a la entrada.
  - **Algoritmos de Optimización**
    - **Ascenso a Colina**: genera soluciones cada vez mas complejas de manera iterativa, y se repite hasta que no se pueda encontrar una mutación que provoque un incremento en la aptitud de la solución actual. Realiza siempre la mejor elección en cada paso.
    - **Recocido Simulado**: función de aptitud define una solución candidata, añade concepto de temperatura que es cantidad numérica global que disminuye de manera gradual. La aptitud de la nueva solución se compara con la anterior y se remplaza si es mejor. Cuando temperatura alcanza el cero el sistema termina.
- **Algoritmos Genéticos**
  - Técnicas de optimización con un diseño basado en el concepto de evolución y que utilizan procesos como las combinaciones genéticas, las mutaciones y la selección natural.
  - Tratan de encontrar la mejor solución a un problema dado entre un conjunto de soluciones posibles. Mecanismos que se asemejan a la evolución biológica.
  - Trabajan sobre el concepto de la mutación, ya que comienza con una población de soluciones, luego se elige a los mejores y se combinan para generar nuevos. Se repite hasta encontrar la mas optima.
- **Arboles de Decisión**
  - Estructura cuya forma representa la copa de un árbol mediante un conjunto de decisiones. Estas decisiones son las que generan las reglas que clasifican un conjunto de datos que se segmentan mediante búsquedas arboladas.
  - Técnica de programación que permite analizar decisiones secuenciales basadas en el uso de resultados y probabilidades asociadas.
  - Se utilizan en los sistemas expertos, que se basan en grandes bases de datos, donde se cargan reglas de decisión que encuentran su fundamento en la experiencia e los expertos.