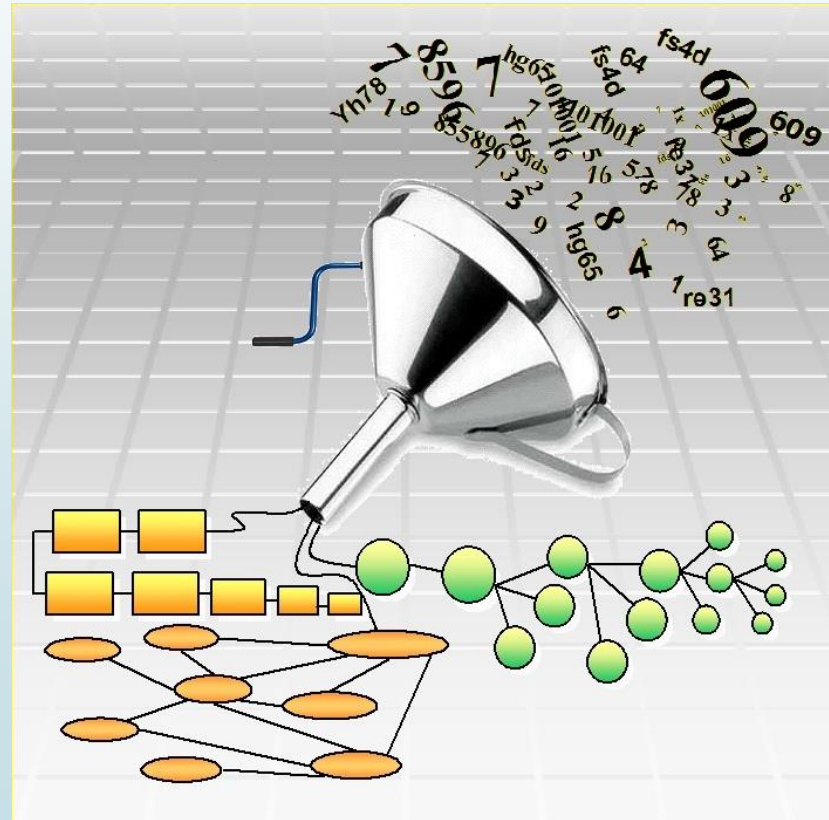


Tema 1. Introducción a las estructuras de datos





Tema 1 competencia específica

Conoce y comprende las diferentes estructuras de datos, su clasificación y forma de manipularlas para buscar la manera más eficiente de resolver problemas.



Tema 1 competencias genéricas:

- Capacidad de análisis y síntesis
- Habilidad en el manejo de equipo de cómputo
- Capacidad para trabajar en equipo
- Habilidad para buscar y analizar información proveniente de fuentes diversas
- Capacidad de aplicar los conocimientos en la práctica.

TEMA	SUBTEMA
1. Introducción a las estructuras de datos	<ul style="list-style-type: none">1.1 Clasificación de las estructuras de datos1.2 Tipos de datos abstractos (TDA)1.3 Ejemplos de TDA's1.4 Manejo de memoria<ul style="list-style-type: none">1.4.1 Memoria estática1.4.2 Memoria dinámica1.5 Análisis de algoritmos<ul style="list-style-type: none">1.5.1 Complejidad en el tiempo1.5.2 Complejidad en el espacio1.5.3 Eficiencia de los algoritmos.

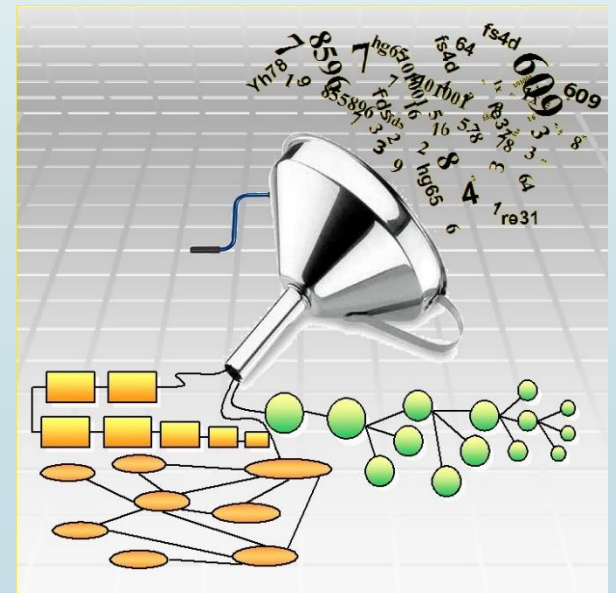


Introducción

- Muchos algoritmos requieren una representación apropiada de los datos para lograr ser eficientes.
- Esta representación junto con las operaciones permitidas se llama

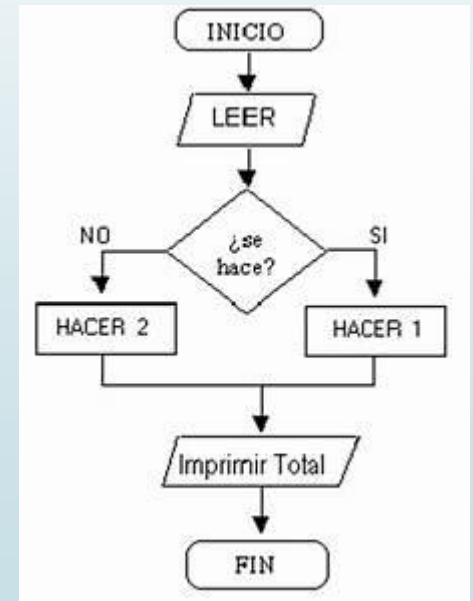
Estructura de datos.

- ▶ Una Estructura de datos, es una colección de datos almacenados, que se caracteriza por su organización y las operaciones que la componen.
- ▶ Una estructura de datos está compuesta por tipos de datos simples o primitivos(entero, real, carácter, etc.).



Usos de las estructuras de datos

- Manejo eficiente de grandes cantidades de datos.
- Son la clave para crear algoritmos eficientes.
- Ayuda en la organización para el diseño de software

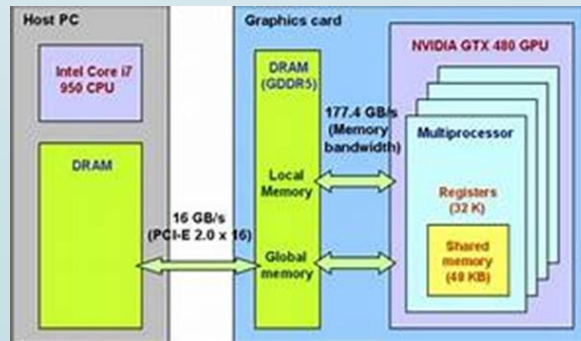


Objetivo de las estructuras de datos:

- Facilitar la organización de los datos con eficiencia



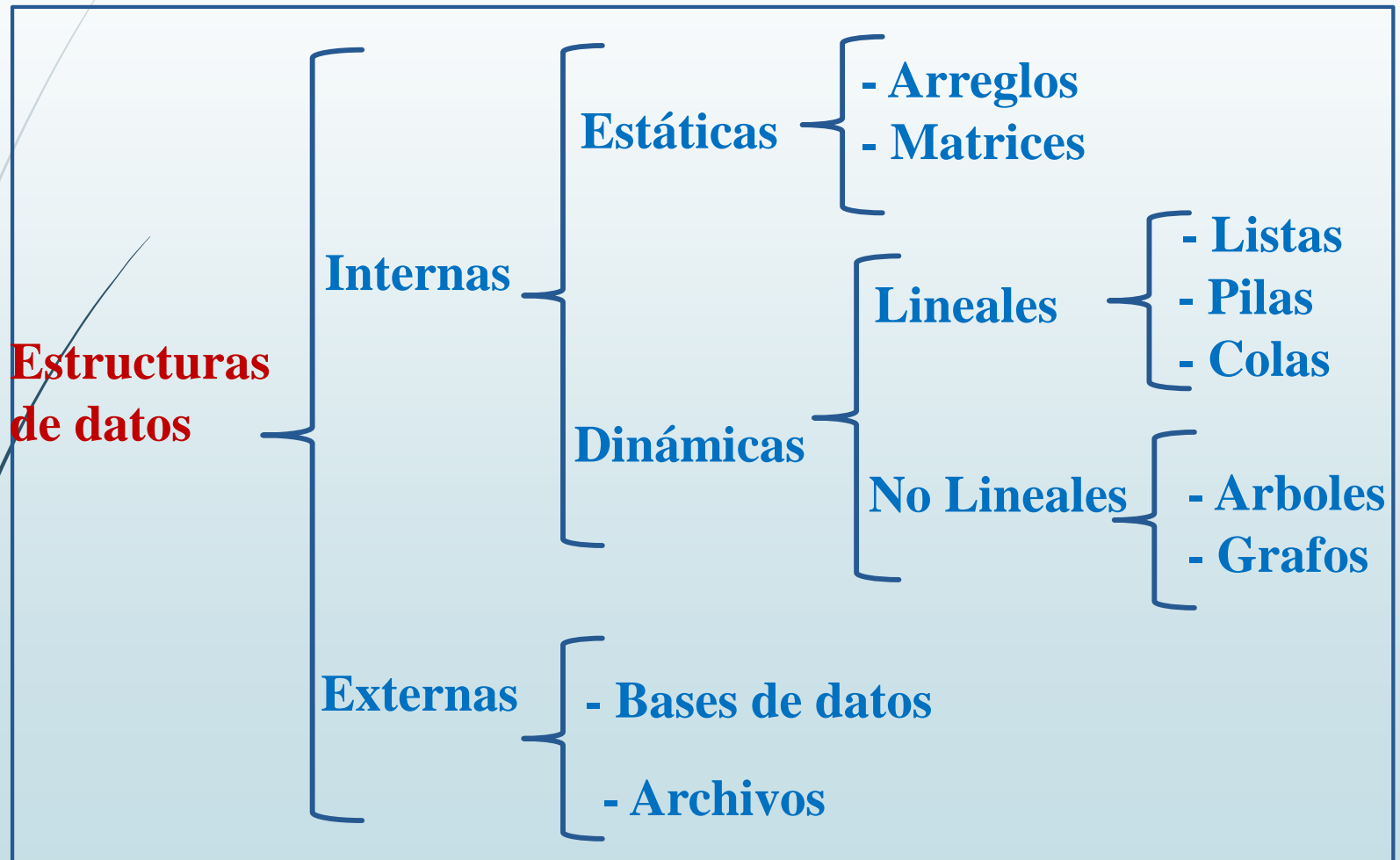
- Eficiencia
la mínima cantidad de recursos (espacio de memoria y tiempo de proceso)



CPU Time



1.1 Clasificación de las estructuras de datos





1.1 Clasificación de las estructuras de datos

- Estructuras de datos lineales:
arreglos, pilas, colas, listas enlazadas.
- Estructuras de datos no lineales.
árboles, grafos.

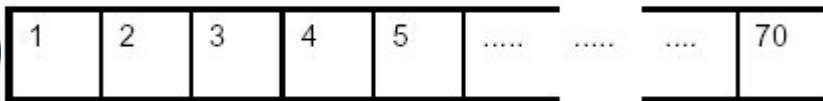


Estructuras de datos lineales:

Las estructuras lineales de datos se caracterizan porque sus elementos están en secuencia, relacionados en forma lineal, uno luego del otro. Cada elemento de la estructura puede estar conformado por uno o varios subelementos o campos que pueden pertenecer a cualquier tipo de dato, pero que normalmente son tipos básicos.

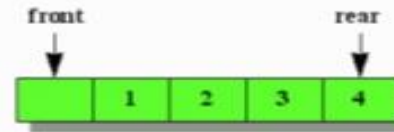
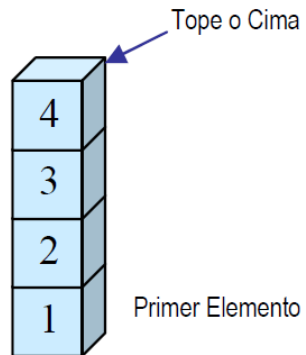
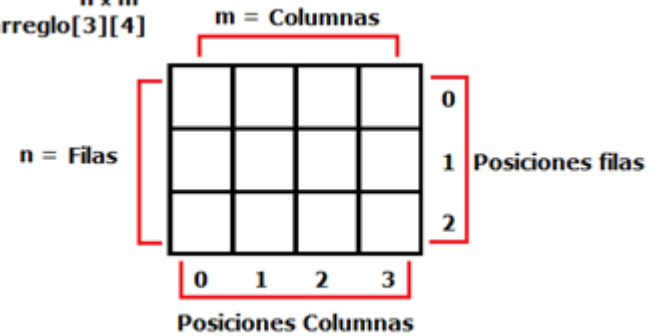
Estructuras de datos lineales

Arre = arreglo[1..70] de enteros

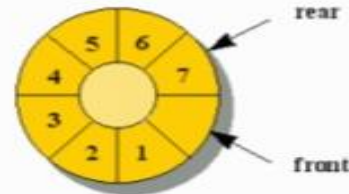


NTE = $70 - 1 + 1 = 70$ elementos

$n \times m$
arreglo[3][4]



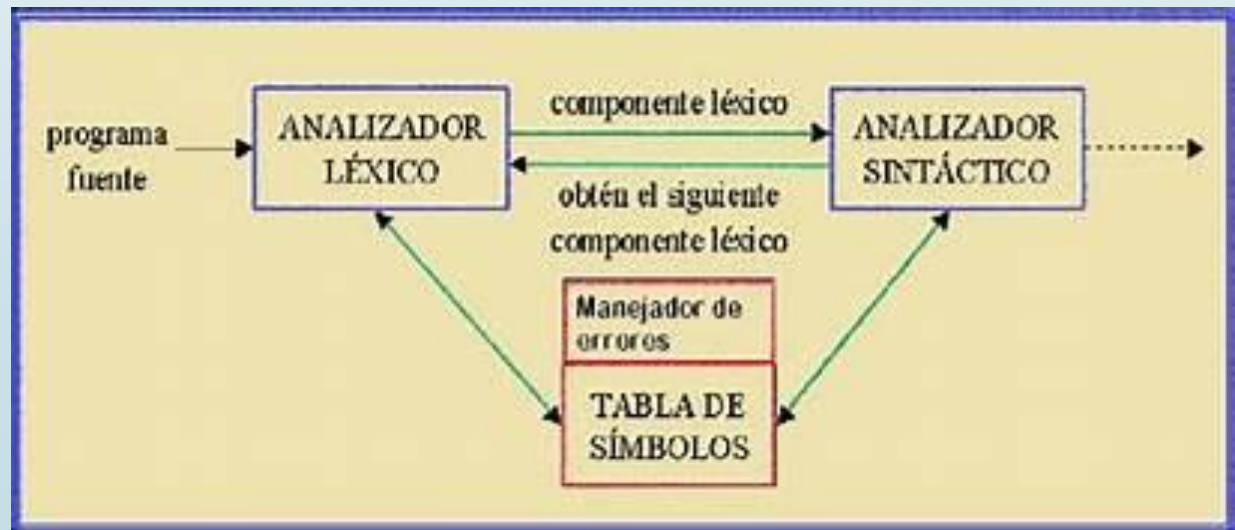
linear queue



circular queue

Entre las aplicaciones que tienen estas estructuras se puede mencionar:

- El desarrollo de compiladores de lenguajes de programación que están conformados por varios subprogramas con finalidades más específicas, como por ejemplo: el analizador de léxico que genera la tabla de símbolos.



```
IF cuenta = sueldo THEN jefe:= justo;
```

El analizador léxico la separa en la siguiente secuencia de tokens:

IF cuenta = sueldo THEN jefe := justo

Y les asigna su atributo, habitualmente por medio de un código numérico cuyo significado se ha definido previamente.

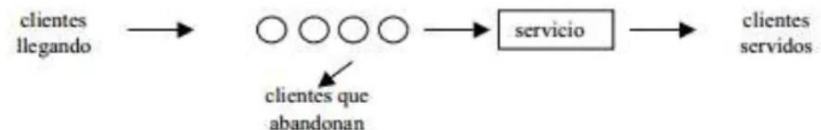
Token	Atributo	Observaciones
IF	20	Palabra reservada
cuenta	1	Identificador
=	15	Operador de comparación
sueldo	1	Identificador
THEN	21	Palabra reservada
jefe	1	Identificador
:=	10	Asignación
Justo	1	Identificador
;	27	Separador de sentencias

Tabla de Símbolos		
Token	Lexema	Palabra Reservada
pro	programa	Yes
int	int	Yes
char	char	Yes
float	float	Yes
leer	leer	Yes
imp	imprimir	Yes
+	+	Yes
-	-	Yes
*	*	Yes
/	/	Yes
=	=	Yes
ter	terminar	Yes
min	mientras	Yes
si	si	Yes
sino	sino	Yes
"	"	Yes
,	,	Yes
:	:	Yes
((Yes
))	Yes
{	{	Yes
}	}	Yes
&	&	Yes
&&	&&	Yes
		Yes
		Yes

- La simulación discreta de sistemas a través del computador, donde la mayoría de los paquetes de simulación digital ofrecen lenguajes de simulación que soportan las primitivas para el manejo de colas y sus diferentes versiones.

Teoría de colas

Un sistema de colas se puede describir como: “clientes” que llegan buscando un servicio, esperan si este no es inmediato, y abandonan el sistema una vez han sido atendidos. En algunos casos se puede admitir que los clientes abandonan el sistema si se cansan de esperar. El término “cliente” se usa con un sentido general y no implica que sea un ser humano, puede significar piezas esperando su turno para ser procesadas o una lista de trabajo esperando para imprimir en una impresora en red.





A screenshot of the Windows Services console. The 'Cola de impresión' service is highlighted with a blue selection bar. A red oval is drawn around the 'Cola de impresión' row. A red arrow points from the 'Cola de impresión' service name to the 'Cola de impresión' text in the section header below. The table lists several services with their names, descriptions, status, startup type, and logon account.

Cliente de seguridad	Maneja el...	Iniciado	Automático	Sistema local
Cliente DHCP	Administra l...	Iniciado	Automático	Sistema local
Cliente DNS	Resuelve y...	Iniciado	Automático	Servicio de red
Cliente Web	Habilita los...	Iniciado	Automático	servicio local
Cola de impresión	Carga archi...		Automático	Sistema local
Compatibilidad de e...	Proporcion...	Iniciado	Manual	Sistema local
Conexiones de red	Administra ...	Iniciado	Manual	Sistema local

Cola de impresión

➤ La realización de sistemas operativos para las computadoras, los cuales hacen un uso intensivo de las estructuras lineales, ya que internamente se soportan en los sistemas operativos, las colas de ejecución para los dispositivos, las pilas de llamadas a los subprogramas de cualquier programa, las listas de usuarios en los sistemas operativos multiusuarios, etc.

AGif - UNREGISTERED

```
class program {  
    void turn (n) {  
        iterate(n) {  
            turnleft();  
        }  
    }  
  
    program() {  
        turn(3);  
        turnoff();  
    }  
}
```

turnoff();

PILA DE LLAMADAS

► Estructuras de datos no lineales:

Cada elemento puede estar enlazado a cualquier otro componentes. Se trata de estructuras de datos en las que cada elemento puede tener varios sucesores y/o varios predecesores.

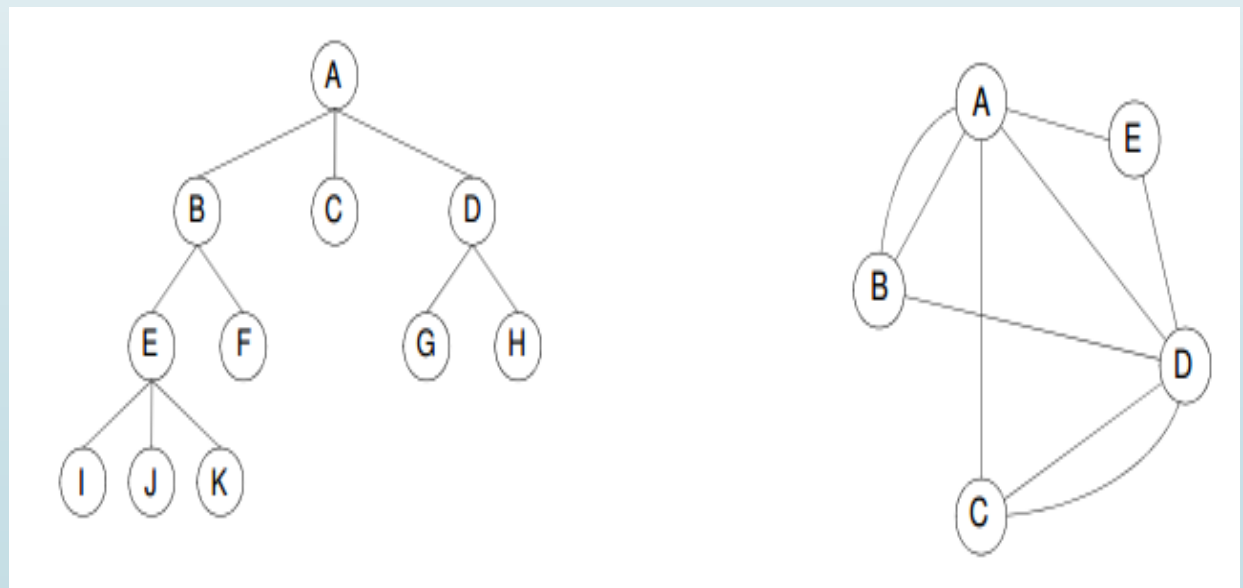
- Árboles.
- Grafos.

Árboles.

- Cada elemento sólo puede estar enlazado con su predecesor y sus sucesores. Puede tener varios sucesores.

Grafos.

- Cada elemento puede estar enlazado a cualquier otro.

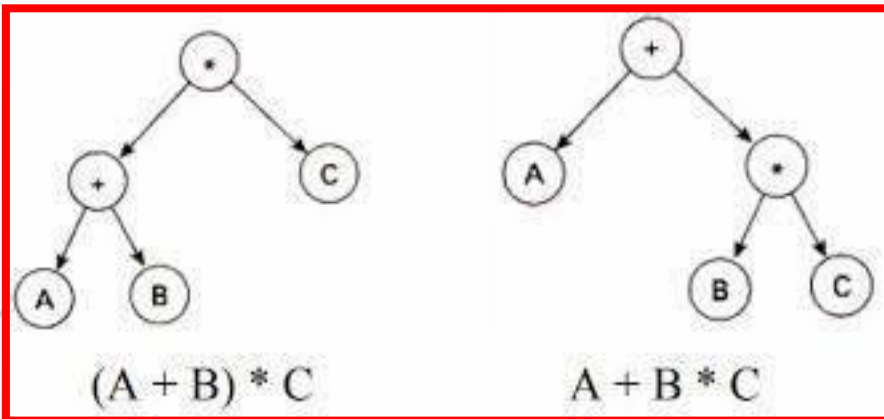




aplicaciones de Estructuras de datos no lineales:

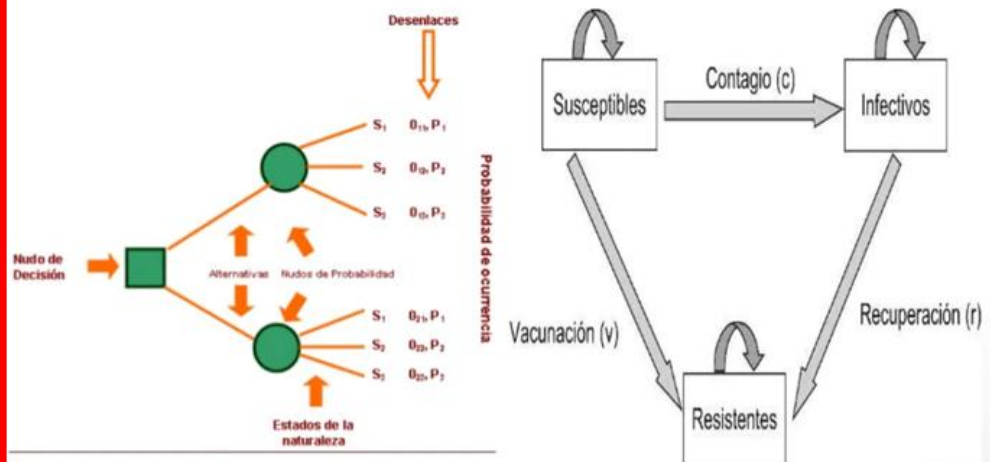
- Árboles:

Evaluación de expresiones algebraicas, búsqueda de elementos, diseño de compiladores, sistemas expertos, manejo de directorios, probabilidad.



Modelos usados en la simulación

Los 2 tipos de modelos más ampliamente utilizados en el ámbito descrito son los árboles de decisión y los modelos de Markov.



• Grafos:

Representación de caminos o rutas de transporte, determinar tiempos mínimos y máximos de un proceso, flujo de control de un programa, representación de circuitos electrónicos analógicos y digitales, representación de redes e computadoras.

Grafo del Recorrido del pizzero
partiendo del negocio Tooginós
al domicilio de un cliente:





1.2 Tipos de datos abstractos (TDA)

- Un **TDA** consiste en una colección de valores y un conjunto de operaciones sobre esos valores.
- La colección de valores y las operaciones sobre esos valores pueden implementarse mediante una **estructura de datos** particular.



Proceso de programación:

- Modelación: solución al problema; modelo matemático y algoritmo informal.
- TDA: algoritmo en pseudocódigo sobre datos y sus operaciones.
- Estructura de datos: lenguaje de programación.



Modelo
Matemático

algoritmo
Informal

TDAs


algoritmo
pseudocódigo
o

Estructuras de
datos

Programa en
Java



1.3 Ejemplos de TDA's

- Vectores
 - Matrices
 - Números racionales
 - Números complejos
- 



► Para trabajar con un TDA se debe:

1. Especificar

2. Implementar



➤ **Especificación:**

Establecer la interfaz con el usuario del tipo (“lo que necesita saber el usuario”)

Decir qué es sin decir nada sobre cómo se hace. Se trata de dar la lista de operaciones necesarias y especificarlas.



➤ **Implementación:**


Elegir la representación de los valores.

Implementar las operaciones

Debe ser estructurada, legible y eficiente



1.3 Ejemplos de TDA's

- Vectores
 - Matrices
 - Números racionales
 - Números complejos
- 



Ejemplo: TDA Arreglo

Especificación:

1) Tipo de datos : enteros

2) tamaño n

3) operaciones:

Leer.- lee por teclado los valores del arreglo,

Sumar.- suma el contenido del arreglo.

Buscar.- busca un elemento x en el arreglo y retorna su posición.

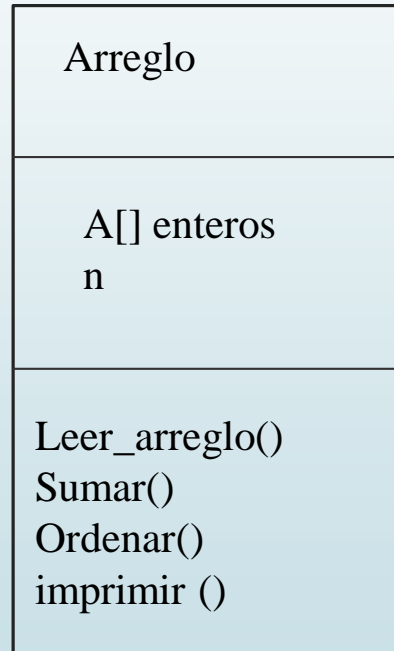
Imprimir.- muestra el contenido del arreglo.



Ejercicio:

Elaborar un diagrama UML para la clase **ARREGLO** sus operaciones: lectura del arreglo, sumar su contenido, buscar un elemento, imprimir el contenido del arreglo.

Diagrama UML de la clase Arreglo:






1.4 Manejo de memoria

- 1.4.1 Memoria estática
- 1.4.2 Memoria dinámica
- **Actividad extra clase...**

Introducción



- La administración de memoria de una computadora es una tarea fundamental debido a que la cantidad de memoria es limitada.
- El sistema operativo es el encargado de administrar la memoria del sistema y compartirla entre distintos usuarios y/o aplicaciones.

- 
- En un lenguaje de programación, el Run Time System (RTS) es el encargado de administrar la memoria para cada programa en ejecución.
 - La ejecución de un programa requiere que diversos elementos se almacenen en la memoria:
 - Código del programa (instrucciones)
 - Datos
 - Permanentes
 - Temporales
 - Direcciones para controlar de flujo de ejecución del programa.



Tarea

Actividad 1.1

Para cada tipo (estática, dinámica) de memoria investiga:

- Métodos de asignación

- Utilización de memoria

- Liberación de memoria

- Ventajas, desventajas

- Diferencias, . . .

Con la información encontrada, elabora un **cuadro comparativo de los tipos de memoria**.



Fuentes de información



- 1. Aho A.V., Hopcroft J.E., Ullman J.D. (1988) *Estructuras de Datos y Algoritmos*. Addison Wesley.
- 2 Cairo, O. y Guardati, S. (2006) *Estructura de Datos, Tercera Edición*. México: Mc Graw Hill.
- 3 Euan J., Cordero L., *Estructuras de datos*, México Limusa, 1989
- 4. Saara Base, *Algoritmos Computacionales. Introducción al análisis de algoritmos*, Adison Wesley.