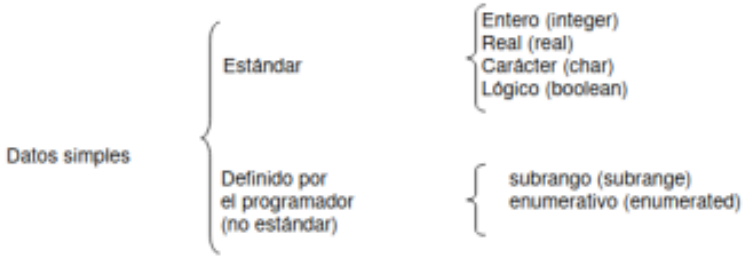
11.- Organización lógica de los datos. Estructuras estáticas.

Conceptos: dato, estructura de datos

Tipos de datos **simples**: entero, real, lógico, carácter

Tipos de datos **compuestos**: cadena de caracteres, registros, arrays, ficheros

Tema relacionado con programación



2.4.- Tipos de estructuras de datos

Los tipos de datos se pueden organizar en diferentes estructuras, atendiendo al espacio que ocupan:

Estructuras Estaticas: el tamaño ocupado en la memoria se define antes de que el programa se ejecute y no puede modificarse durante la ejecución del programa. Están implementadas en casi todos los lenguajes y son: registros, arrays, conjuntos, uniones y ficheros.

Estructuras Dinámicas: no tienen las limitaciones en el tamaño de memoria ocupada propias de las estructuras estáticas. Su tamaño puede cambiar en tiempo de ejecución, tanto para aumentar como para decrecer. Las más comunes son las listas (enlazadas, pilas, colas), árboles (binarios, B, B+, …) y grafos.

este tema

Clasificación de la organización lógica de los datos

11.- Organización lógica de los datos. Estructuras estáticas.

Array: declaración, recorrido, asignación, lectura/escritura

Actualización de un array:
Añadir al final (append)
Insertar un elemento (en cualquier posición) → desplazar
Borrar elemento → desplazar otros

Pera	Manzana	Fresa		

Insertar "Uva" en la tercera posición; "Fresa" se desplaza una posición.

Pera	Manzana	**Uva**	**Fresa**	

Pera	Manzana	Uva X	Fresa	

Borrar elemento Manzana; desplazamiento de los elementos a la derecha

Pera	Uva	Fresa		

11.- Organización lógica de los datos. Estructuras estáticas.

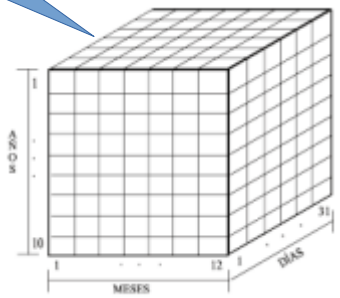
3.1.2.- Arrays bidimensionales: matrices

Un array de dos dimensiones necesita de **dos índices** para poder acceder a todas posiciones que lo forman. Están formados por datos del mismo tipo en las dos dimensiones. Es el caso de la representación de una matriz bidimensional, por ejemplo *Entero A[3,4]* donde la matriz se denomina A y cuenta con tres filas y cuatro columnas.

A[1,1]	A[1,2]	A[1,3]	A[1,4]
A[2,1]	A[2,2]	A[2,3]	A[2,4]
A[3,1]	A[3,2]	A[3,3]	A[3,4]

En este caso se dispone de dos índices para referenciar los elementos

En **memoria se almacena de forma lineal** (dos alternativas)
- Orden de fila mayor (C, Basic, Cobol, Pascal)
- Orden de columna mayor (FORTRAN)



Ejemplo de array tridimensional

Puede ser de cualquier dimensión =número de índices

11.- Organización lógica de los datos. Estructuras estáticas.

Registro: declaración, asignación de contenido, lectura/escritura

Tiene un número fijo de componentes (campos)
Campo clave
Tipos de datos: simples o compuestos

(*) Pueden ser de longitud fija, variable o indefinida

```
Reg_Pelicula es tipo  REGISTRO
        Entero cod_pelicula
        Cadena Titulo
Fin Reg_Pelicula
```

11.- Organización lógica de los datos. Estructuras estáticas.

Fichero: conjunto de información relacionada susceptible de almacenar en memoria secundaria. Identificador único.

Elementos: registro, campo, apuntador → reg.activo, EOF

Atributos: nombre, ubicación, tipo, tamaño, fecha/hora, protección

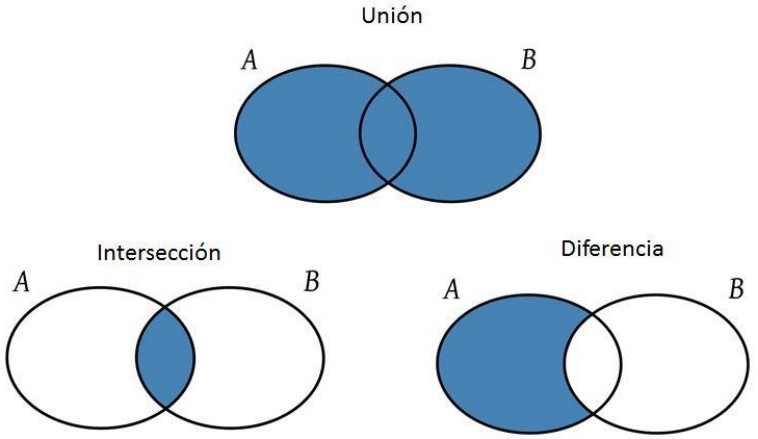
Operaciones

- Crear estructura fichero
- Abrir (modo de apertura, según lenguaje)
- Leer un registro
- Escribir registro
- Cerrar fichero
- Renombrar fichero
- Copiar fichero
- Editar fichero (modificar contenido)
- *Indexar un fichero*

Organización del fichero: relación directa con soporte de almacenamiento

- Secuencial
- Secuencial indexada
- Directa o aleatoria

11.- Organización lógica de los datos. Estructuras estáticas.



Ej. Pascal

```
var
  letras: set of char;
begin
  letras := ['a', 'b', 'c', 'd'];
end
```

Conjuntos: Concepto matemático

Su implementación depende de cada lenguaje de programación

Operaciones sobre conjuntos: **unión, diferencia, intersección, pertenencia**