

TEMAS DE INTRO I

- Pilas (datos/estructuras de control)
- Filas (datos/estructuras de control)
- Modularización y Parámetros
- Variables
- Método de Desarrollo
- Funciones
- Arreglos
- Ordenamientos
- Matrices y Tipo String

Matrices

Arreglos de más de una dimensión

Const

InicioFila=1;

FinFila=3;

InicioCol=1;

FinCol=5;

Type

Matriz: Array [inicioFila..FinFila, inicioCol..finCol] of **Integer;**

Var

Mat: Matriz;

InicioFila

FinFila

InicioCol

FinCol

Mat[1,5]

Se accede por celda indicando cada dimensión →

Matrices

Arreglos de más de una dimensión

Const

InicioFila=1;

FinFila=3;

InicioCol=1;

Fincol=5;

Type

Matriz: Array [inicioFila..finFila, inicioCol..finCol] of **integer**;

Var

Mat: Matriz;

inicifila					44
finfila					
	inicioCol				finCol

Mat[1,5]:= 44

Matrices

Const

InicioFila=1;

FinFila=3;

InicioCol=1;

Fincol=5;

Type

Matriz: Array [inicioFila..finFila, inicioCol..finCol] of integer;

Var

Mat: Matriz;

InicioFila					54
FinFila					
	InicioCol				FinCol

Mat[1,5]:= 44

Mat[1,5]:= Mat[1,5] + 10;

CUAL DE AMBAS GRAFICAS ESTA BIEN ??????

Const

AñoInicio=1;

AñoFin=4;

Cuat1=1;

Cuat2=2

Type

Materias: Array [AñoInicio..AñoFin,Cuat1..Cuat2] of TipoPrimitivo;

1				
2				
	1	2	3	4

1		
2		
3		
4		
	1	2

Se puede “graficar” de las dos maneras, pero luego se debe respetar la convención, no mezclar los índices

Const

AñoInicio=1;

AñoFin=4;

Cuat1=1;

Cuat2=2

Type

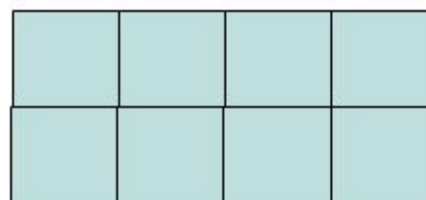
Materias: Array [AñoInicio..AñoFin,Cuat1..Cuat2] of TipoPrimitivo;

1				
2				
	1	2	3	4

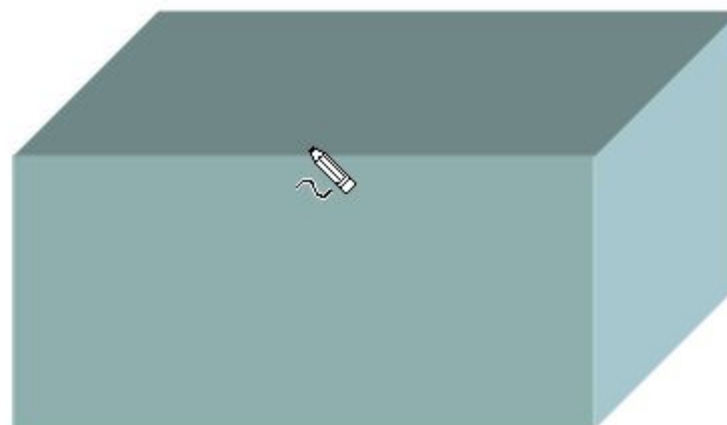
1		
2		
3		
4		
	1	2

Se pueden graficar?

Var **Nombre2D** : array [**Inicio1**..**Fin1**, **Inicio2**..**Fin2**] of **TipoPri**;



Var **Nombre3D** : array [**Inicio1**..**Fin1**, **Inicio2**..**Fin2**, **Inicio3**..**Fin3**] of **TipoPri**;

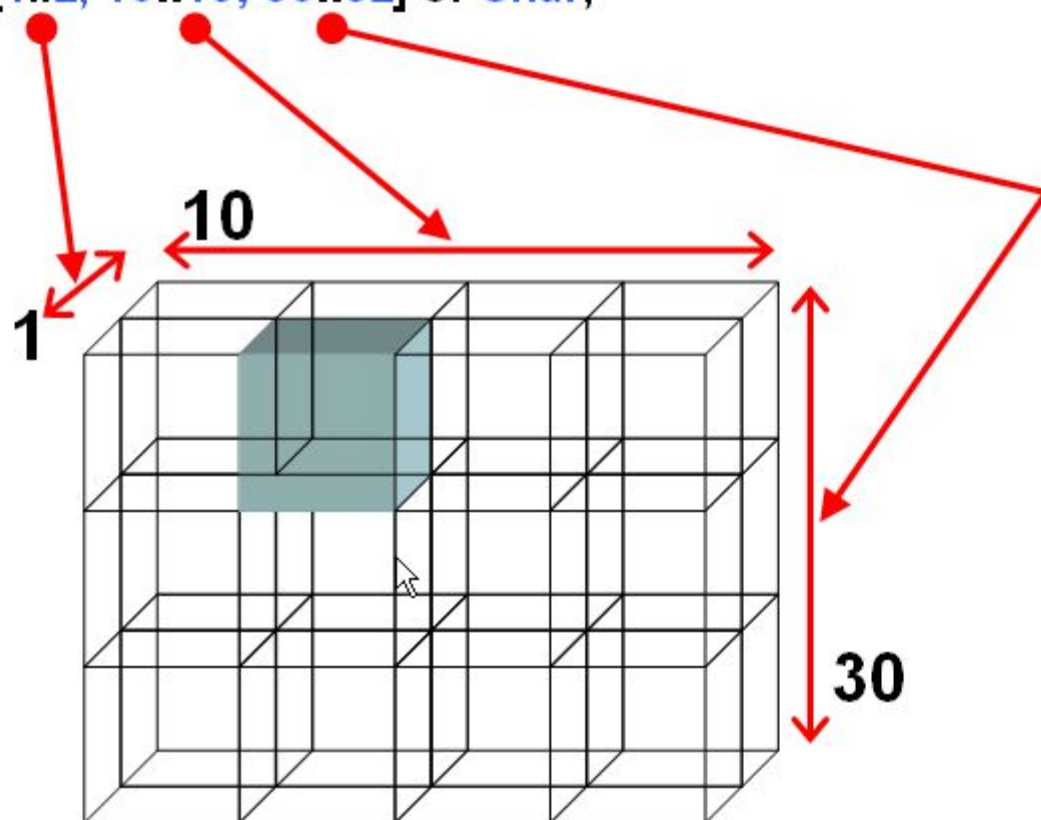


Var **Nombre4D** : array [**Inicio1**..**Fin1**, **Inicio2**..**Fin2**, **Inicio3**..**Fin3** , **Inicio4**..**Fin4**] of **TipoPri**;



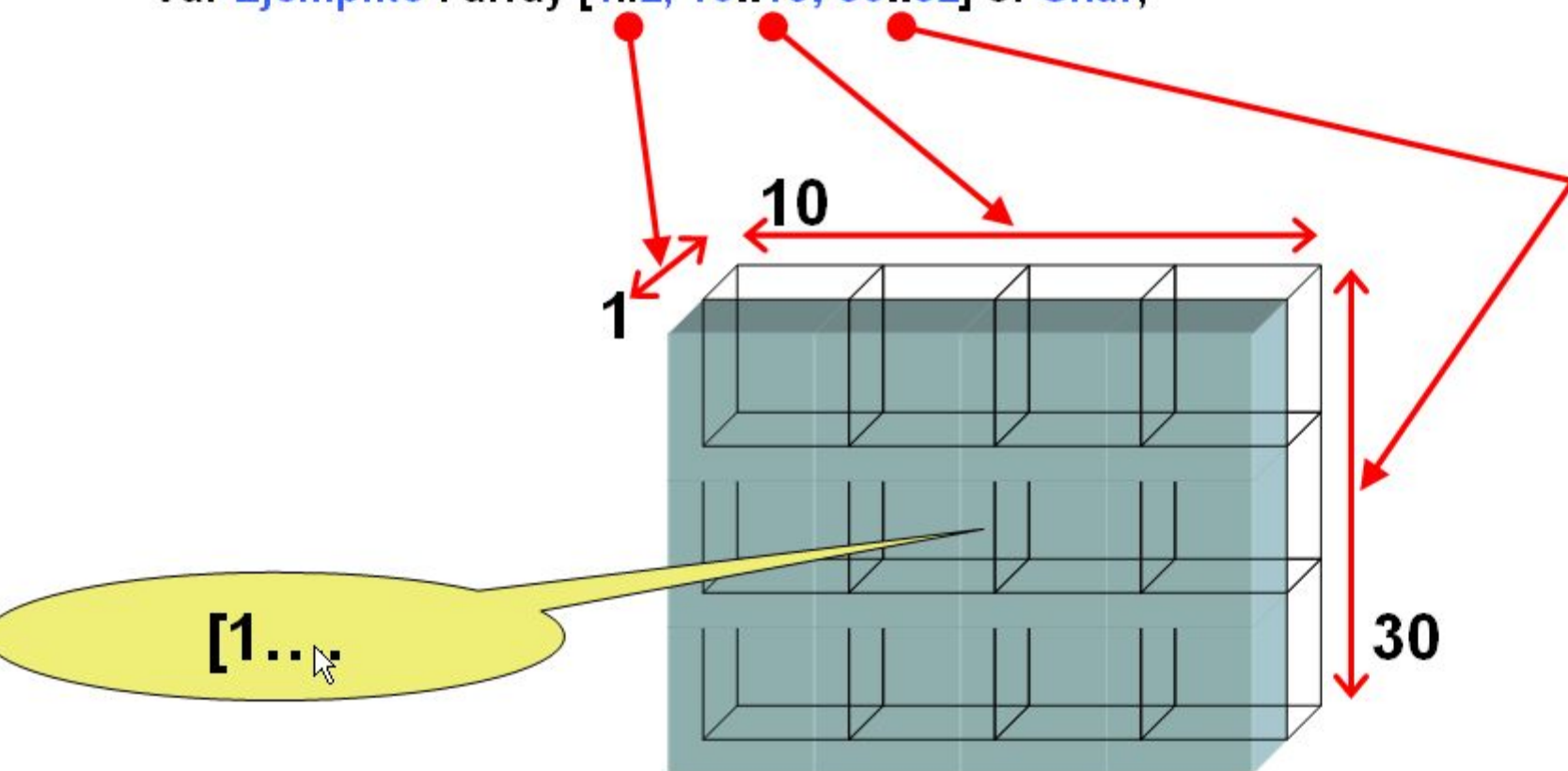
Ejemplo de tres dimensiones

Var **Ejemplito** : array [1..2, 10..13, 30..32] of Char;



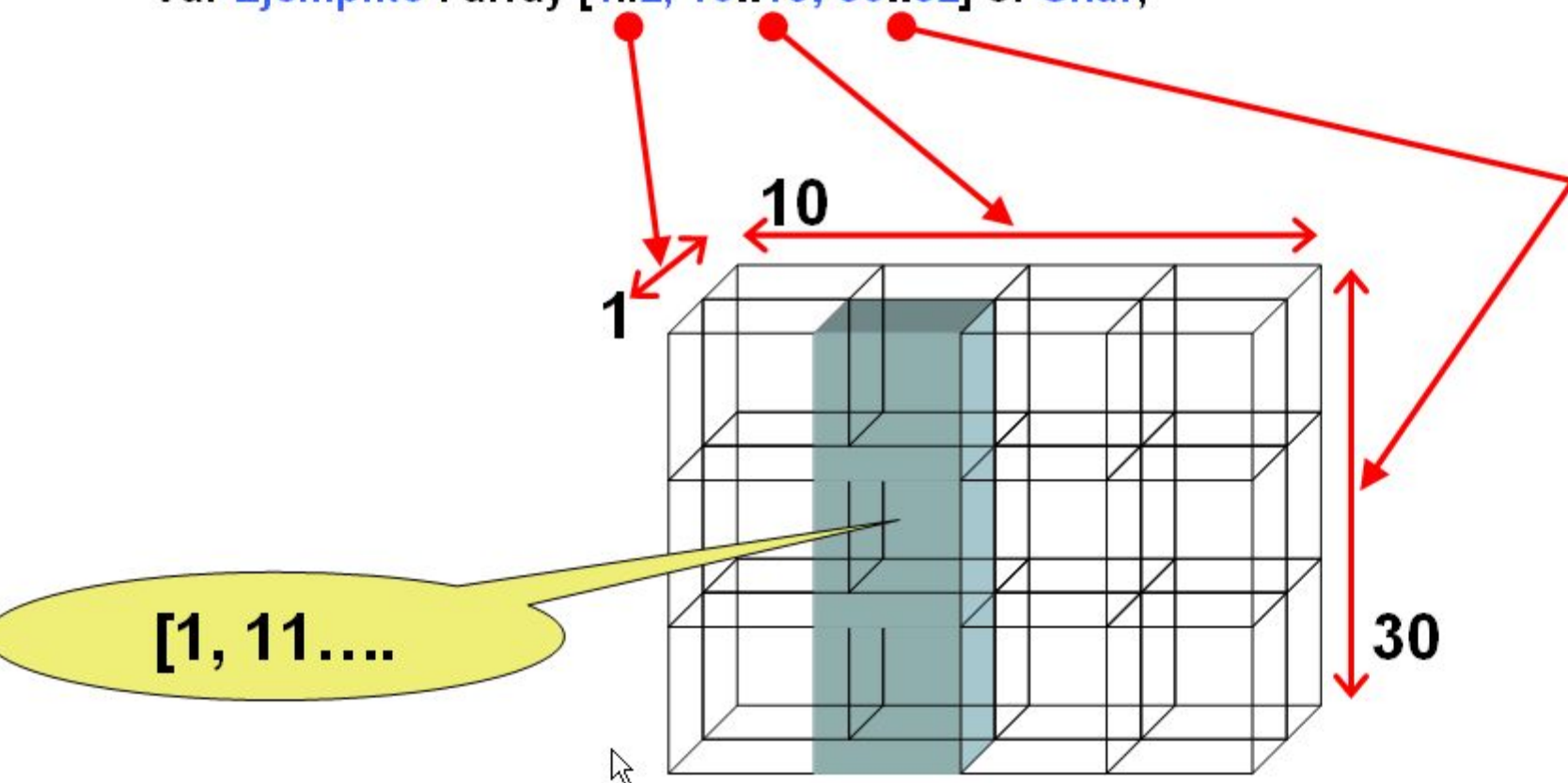
Ejemplo de tres dimensiones

Var **Ejemplito** : array [1..2, 10..13, 30..32] of Char;



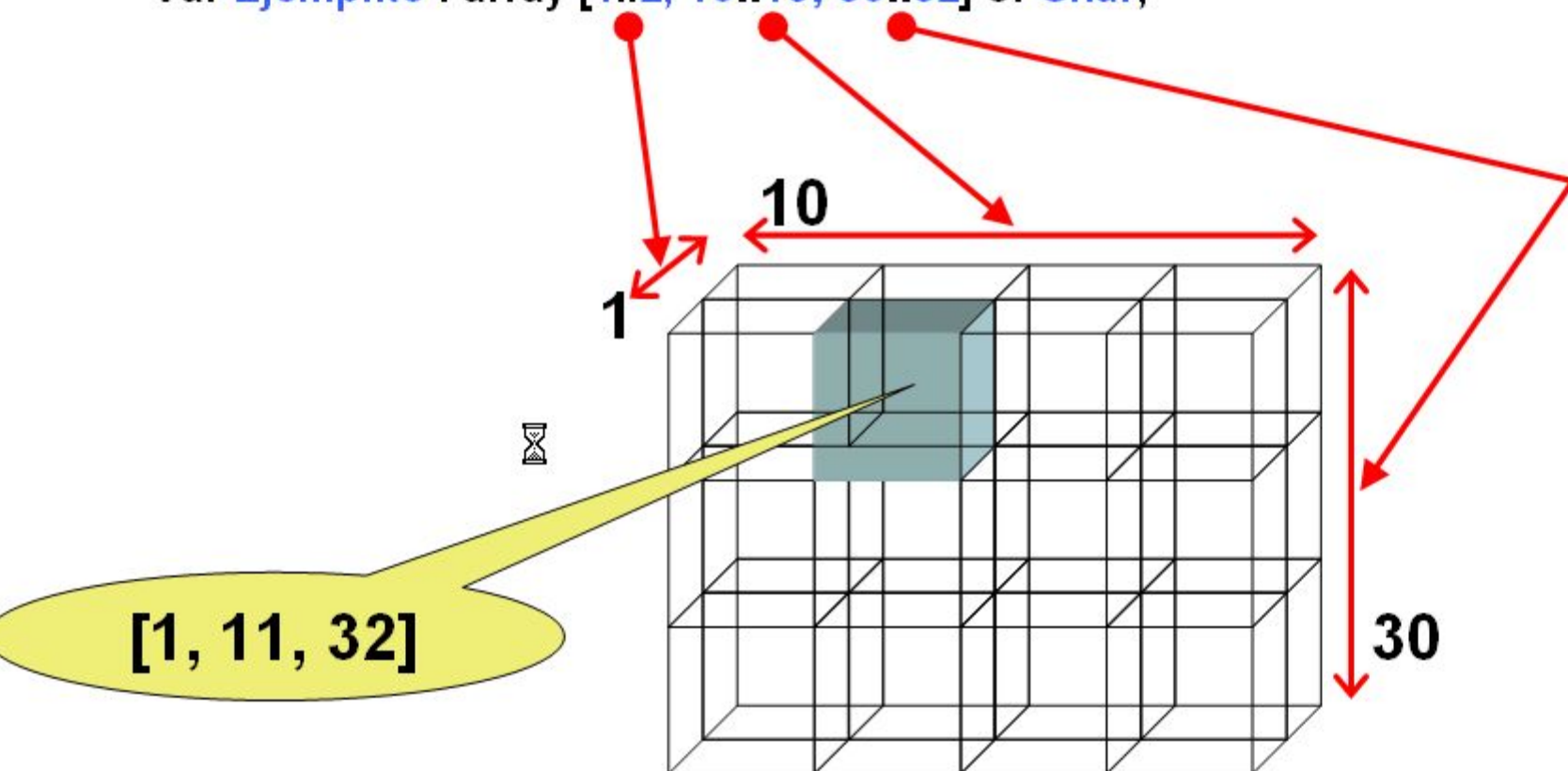
Ejemplo de tres dimensiones

Var **Ejemplito** : array [1..2, 10..13, 30..32] of Char;



Ejemplo de tres dimensiones

Var **Ejemplito** : array [1..2, 10..13, 30..32] of Char;



- Definir un matriz de dos dimensiones de caracteres y hacer un módulo para cargarla por teclado

```
program EjemploMatriz;
```

```
{.....}
```

```
const
```

```
    maxfila=2;
```

```
    maxcol=2;
```

```
type
```

```
    matriz= array [1..maxfila, 1..maxcol] of char;
```

```
procedure cargarFila (var mat:matriz, fi:integer);
```

```
var
```

```
    co: integer;
```

```
begin
```

```
    for co:=1 to maxcol do readln(mat[fi,co]);
```

```
end;
```

```
procedure cargarmatriz(var mat:matriz);
```

```
var
```

```
    fi, integer;
```

```
begin
```

```
    for fi:=1 to maxfila do cargarFila (mat, fi)
```

```
end;
```

- Implementar un módulo que pase como parámetro esa matriz y un dato a buscar y devuelva Verdadero ó Falso si el elemento está

```

Function estaenFila(mat: matriz; fil:integer; dato:char): boolean;
var
    col: integer;
begin
    col:=1;
    while (col<= maxcol) and (mat[fil, col]<> dato) do
        col:= col+1;
    estaenFila:= col <= maxcol
end;

```

```

Function estaEnMatriz (mat: matriz; dato: char): booelan
var
    fil: integer;
begin
    fil:=1;
    while (fil<=maxfila) and not estaenFila(mat,fil,dato) do
        fil:= fil +1;
    estaEnMatriz:= fil <= maxfila
end.

```

```
program EjemploMatriz;  
    {busca si un elemento esta en la matriz}
```

```
const
```

```
    maxfila=2;
```

```
    maxcol=2;
```

```
type
```

```
    matriz= array [1..maxfila, 1..maxcol] of char;
```

```
{módulos y funciones}
```

```
var
```

```
    mat: matriz
```

```
    dato:char
```

```
begin
```

```
    CargarMatriz(mat);
```

```
    readln(dato);
```

```
    if estaenMatriz (mat, dato)
```

```
    then writeln ('El elemento ingresado está' )
```

```
    else writeln ('El elemento ingresado NO está' );
```

```
end;
```


Tipo String

- Es una secuencia de caracteres (cadena) de longitud variable.
- `Var Palabra: string` (long. Variable con un máximo de 255)
- `Var Palabra: string[10]` (long. Variable con un máximo de 10)

String

```
Var nombre: string;
```

```
....
```

```
nombre:= 'Jorge' ;
```

Operador:

El operador '+' sirve para concatenar Strings.

Ejemplo:

...

ApellidoCliente := 'PEREZ';

NombreCliente := ApellidoCliente + ', ' + 'Juan';

Write(NombreCliente); → **PEREZ, Juan**

...

Operadores relacionales: (< <= > >= = <>)

Algunos ejemplos:

'a' < 'b' → True

'aa' < 'b' → True

'a' < 'ab' → True

'Z' < 'a' → True

'b' < 'ba' → True

Algunas funciones con String....

- `Length('abcde')` retorna 5
- `Copy ('1234567', 3, 2)` retorna '34'

program Ejemplo;

const

max=5;

type

palabras= array[1..max] of string;

var

Apellido, nombre, nomyape: palabras;

i: integer;

begin

for i:=1 to max do begin

writeln ('Ingrese apellido');

readln(apellido[i]);

writeln ('Ingrese nombre');

readln(nombre[i])

end;

for i:=1 to max do nomyape[i]:= nombre[i] + ' ' + apellido[i];

for i:=1 to max do writeln (nomyape[i])

end.

¿ Qué hace este programa?