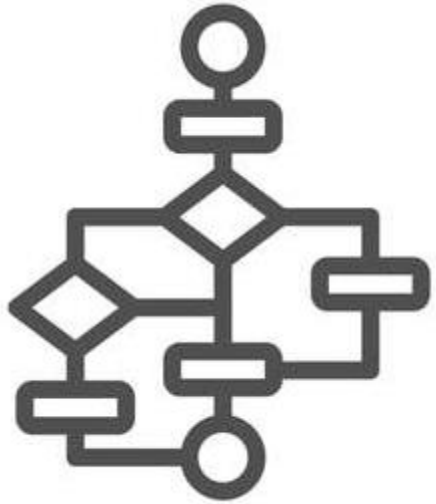
Florida

1º DAM/DAW EDE

U3. UML Comportamiento: actividades

3 - UML Actividades. Arrays

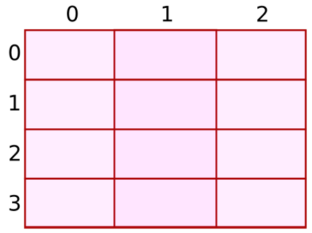


Pascual Martínez

¿Qué es un array?

- Un array es una **estructura de datos** o, dicho de otro modo, un **tipo de dato estructurado**.
- Consta de una **serie o colección de elementos**.
- En función de cuántas **dimensiones** tenga puede ser:
  - Unidimensional: **vector**.

  - Bidimensional: **matriz (tabla)**.



  - N-dimensionales: …

Declaración de arrays. Posiciones

- Un array se declara, básicamente, del siguiente modo:
  - **Unidimensional.** Ejemplo JavaScript:

    let nombre_array = [valor_0, valor_1, …, valor_n];

  - **Bidimensional.** Ejemplo JavaScript (array de arrays):

    let nombre_array = [

    [valor_0_0, valor_0_1, …, valor_0_n],

    [valor_1_0, valor_1_1, …, valor_1_n],

    [… , … , …, …],

    [valor_n_0, valor_n_1, …, valor_n_n]

    ];

array

índice	0	1	2	3	4	5	6	7	8	9
contenido	13	21	34	92	0	6	76	4	0	84

matriz

índice \ índice	0	1	2	3	4	5	6	7	8
0	76	4	0	84	21	34	92	0	6
1	1	2	3	4	5	6	7	8	9
2	21	34	92	0	6	76	4	0	84

contenido

- Por defecto, **un array empieza en la posición o índice 0**.

Asignación/Consulta de valores en arrays

- Dada una posición de un array, puedo:
  - **Asignar** un valor en esa posición:
    - **Unidimensional.** Ejemplo JS:

      Dado un vector:

      - vector[0] = 8;
      - vector[0] = “Jorge”;
      - vector[5] = 14;

    - **Bidimensional.** Ejemplo JS:

      Dada una matriz:

      - matriz[0][0] = 8;
      - matriz[0][0] = “Jorge”;
      - matriz[5][3] = 14;

  - **Consultar** el valor de esa posición:
    - **Unidimensional**: Ejemplo JS:
      - console.log(vector);
      - console.log(vector[0]);
      - variable = vector[5];
    - **Bidimensional. Ejemplo** JS:
      - console.log(matriz);
      - console.log(matriz[0][0]);
      - variable = matriz[5][3];

Longitud de arrays

- Resulta muy útil conocer la longitud de un array:
  - **Unidimensional**:
    - Ejemplo JS:

      longitud = vector.length;

      ultimo_elemento = vector.length **- 1**;

  - **Bidimensional**:
    - Ejemplo JS:

      numero_filas = matriz.length;

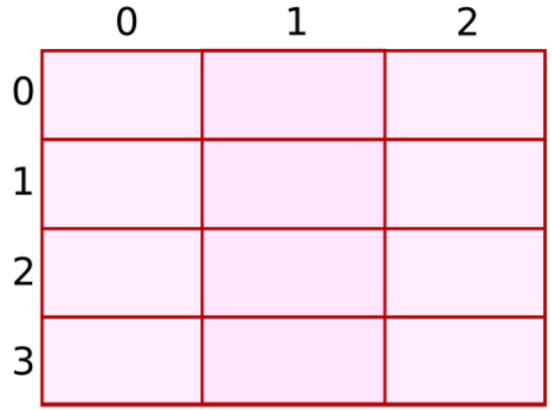
      longitud_columna = matriz.length;

      longitud_fila_0 = matriz[0].length;

      ……

      longitud_fila_n = matriz[n].length;

      longitud_total = ∑ (longitud_fila_x)



Si longitud_fila_0 = longitud_fila_1 = …. = longitud_fila_n → **longitud_total = longitud_columna * longitud_fila_x**

Recorrido de arrays unidimensionales

- Utilizaremos la estructura de control **FOR**, donde:
  - **Inicialización**: indicaremos en qué componente del array empezará el recorrido. (**i = 0**)
  - **Condición**: mientras se cumpla la condición continuamos el recorrido. Si no se cumple, finalizamos el recorrido. (¿Nos queda algún elemento por recorrer? **¿ i<vector.length ?**)
  - **Incremento o paso**: avanzamos al siguiente elemento del recorrido. (**i = i + 1**)

Ejemplo JavaScript:

```
for (i = 0; i < vector.length; i++)
{
  actividad;
}
```

Recorrido de arrays unidimensionales

7	9	1	0	12	50
0	1	2	3	4	5

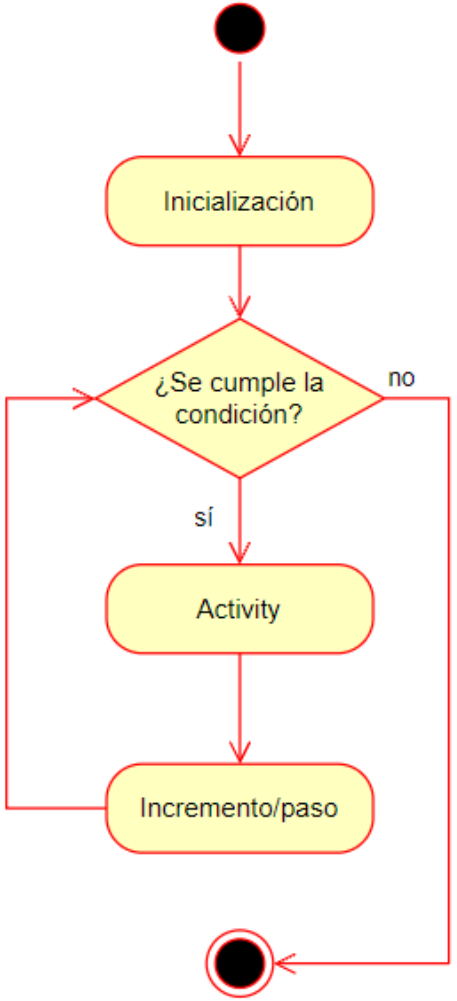
Ejemplo JavaScript:

```
for (i = 0; i < vector.length; i++)

{

  actividad;

}
```

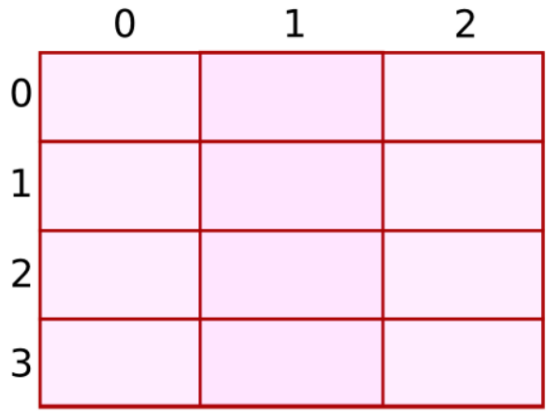


Recorrido de arrays bidimensionales

- Utilizaremos **2 estructuras de control FOR anidadas**:

Ejemplo JavaScript:

```
//Recorremos cada fila
//matriz.length = número de filas = longitud columna
for (i = 0; i < matriz.length; i++)
{
        //Recorremos cada elemento dentro de una fila
        //matriz[i].length = longitud de cada fila
        for (j = 0; j < matriz[i].length; j++)
                {
                        actividad;
                }
}
```



Recorrido de arrays bidimensionales