La Sénia
Institut d'Educació Secundària
Paiporta

Unión Europea
Fondo Social Europeo
El FSE invierte en tu futuro

Estructuras de Datos

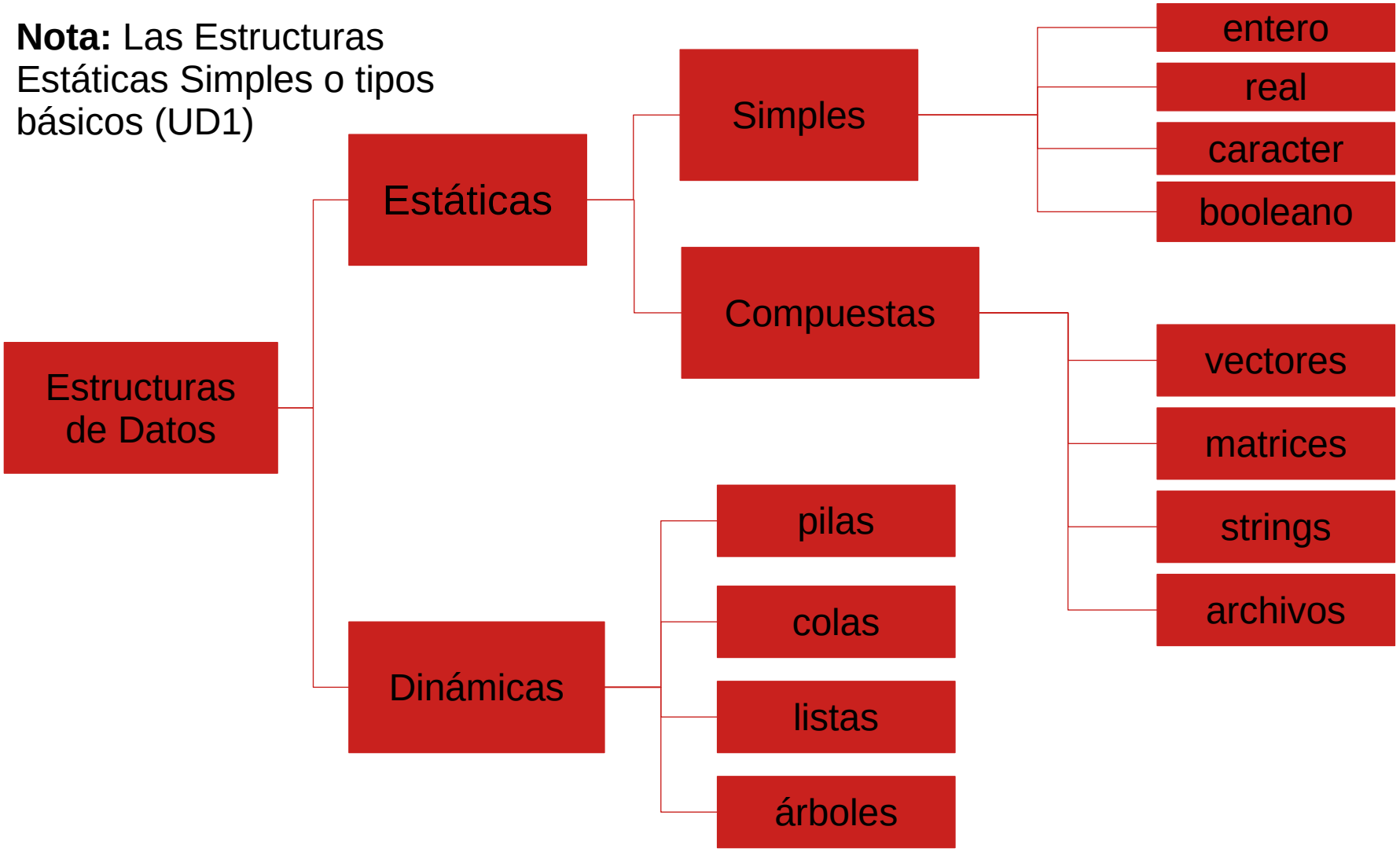
Curso 2023 - 2024

1. Estructuras de Datos

- **Estáticas**:
  - Ocupan tamaño en memoria **fijo**.
  - Se definen con anterioridad a la ejecución del programa.
  - Su dimensión no puede modificarse.
- **Dinámicas**:
  - Ocupan un espacio en memoria **variable** (puede crecer o decrecer)
  - Se definen la estructura sin reservar espacio.

1. Estructuras de Datos

Nota: Las Estructuras Estáticas Simples o tipos básicos (UD1)



2. Vectores

- **Vectores** o **arrays**: Almacenar datos del mismo tipo.
  - **Declaración del array**: igual que los tipos primitivos:

```
tipo [ ] nombre;
```

  - **Inicializar array**: indicar tamaño del array (elementos).

nombre = new tipo[*dimension*];

dimensión es un número entero positivo que indica el número de elementos.

int [ ] a = new int [10];

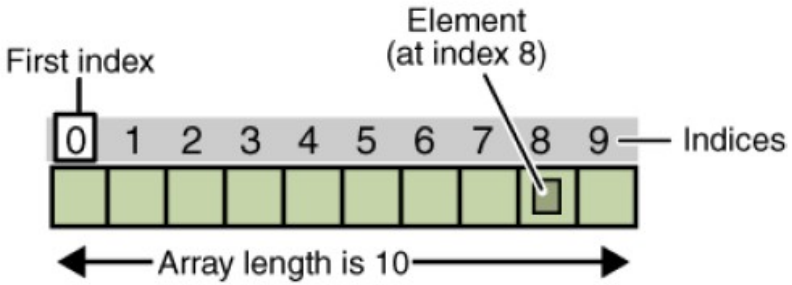
boolean [ ] b = new boolean [100];

char [ ] c = new char [4];

2. Vectores

El acceso a un elemento del array se realiza con **nombre[i]**

Se empieza por la posición **[0]**



One Dimensional array

Initialization `int a[] = new int [12];`

Value	1	2	3	4	5	6	7	8	9	10	11	12
Index	a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]	a[9]	a[10]	a11[11]

`System.out.print(a[5]);` Output: 6

Ejemplo.

```
public class EjemploArray1 {

    public static void main(String[] args){
        //Declaración de variables
        int[] intArray = new int[5];

        System.out.println("El primer elemento es: "+intArray[0]);

        System.out.println("El último elemento es: "+intArray[4]);

        intArray[1] = 8;

        System.out.println("El último elemento es: "+intArray[1]);


    }
}
```

Ejemplo.

```
public class EjemploArray2 {

    public static void main(String[] args){
        //Declaración de variables
        int[] intArray = {4,3,1,0,6,9};
        char[] charArray = {'a','e','i','o','u'};

        System.out.println(intArray[5]);

        System.out.println(charArray[2]);


    }
}
```

2. Vectores

Ejemplo:

- Crea un vector de enteros de 10 elementos.
- Realiza un bucle para inicializar sus valores imprimiendo la posición y el valor de cada elemento por pantalla.
- Crea otro bucle que recorra el vector de forma descendente e imprima también la posición y el valor de cada elemento por pantalla.

Ejemplo.

```
public class EjemploArray {

    public static void main(String[] args){
        //Declaración de variables
        int[] intArray = new int[10];

        System.out.println("El primer elemento es: "+intArray[0]);

        System.out.println("El último elemento es: "+intArray[4]);

        intArray[1] = 8;

        System.out.println("El último elemento es: "+intArray[1]);


    }
}
```

3. Clase Array

Java incluye la clase **Arrays** en java.util.Arrays

Métodos:

- `Arrays.sort(v)`: ordena los elementos del vector. Se pueden indicar posiciones de inicio y fin.
- `Arrays.equals(v1, v2)`: compara dos vectores.
- `Arrays.fill(v, val)`: rellena el vector v con el valor val.
- `Arrays.toString(v)`: devuelve una cadena que representa el contenido del vector.
- `Arrays.binarySearch(v, k)`: busca el valor k dentro del vector v (previamente ha de estar ordenado).

3. Clase Array

Ejemplo:

- Crea 3 vectores de enteros, dos de dimensión 10 y uno de dimensión 5 sin inicializar.
- Inicializa los vectores v1 y v2 con un for de forma que v1 tenga el valor del índice y v2 tenga el doble del valor del índice. Muestra v1 y v2 usando el método de Arrays.
- Comprueba si v1 y v2 son iguales y si no son iguales muestra los dos vectores con el método de Arrays
- Busca el elemento 7 en v1 con el método Arrays. Si se encuentra muestra un mensaje indicando la posición y si no muestra el vector con un método.
- Inicializa el vector v3 con el valor 12 e imprime el vector con el método adecuado.
- Mediante un for ascendente, modifica el vector restando a sus elementos el iterador+1. Muestra el nuevo vector. Ordena todo el vector y muestra el vector

4. Matrices

Una matriz puede verse como un vector bidimensional.

Accede a un elemento con el par [fila][columna]

Filas y Columnas tienen índice del 0 al N-1

Ejemplos:

f/c	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

```
m1[0][0]=1;
m1[0][1]=2;
m1[0][2]=3;
m1[1][0]=4;
m1[1][1]=5;
m1[1][2]=6;
m1[2][0]=7;
m1[2][1]=8;
m1[2][2]=9;
```

f/c	0	1
0	Primero	Segundo
1	tres	cuatro

```
m2[0][0]=”Primero”;
m2[0][1]=”Segundo”;
m2[1][0]=”tres”;
m2[1][1]=”cuatro”;
```

4. Matrices

Ejemplo:

- Crea una matriz de enteros indicando la dimensión 3x3.
- Codifica lo necesario para inicializarla de modo que añadas los números del 1 al 9 en cada una de las celdas.
- Imprime cada elemento e indicando su fila y columna. Ver la figura:

f/c	0	1	2
0	1	2	3
1	4	5	6
2	7	8	9

5. Cadenas (Strings)

La clase envoltorio **String** contiene métodos para trabajar con cadenas de caracteres:

String s = new String(“Hola Mundo”);

o String s = “Hola Mundo”;

Metodos:

`s.length`: Tamaño del String

`s.charAt(k)`: carácter de la posición k

`s.substring(k)`: subcadena desde la posición k al final

`s.substring(inicio, fin)`: subcadena desde el caracter inicio a fin

5. Cadenas

- `s.equals(String str)`: Dice si s y str son iguales
- `s.compare(String str)`: compara s y str

  (0 iguales; >0 s es mayor que str; <0 s es menor que str)
- `s=s.toUpperCase()`: devuelve s en MAYÚSCULAS
- `s=s.toLowerCase()`: devuelve s en minúsculas
- `s=s.trim()`: devuelve s eliminando los espacios anteriores y posteriores a la cadena, respetando los que forman parte de la misma.
- `s.split(Char c)`: Separa la cadena cada vez que encuetra el carácter c como elementos de un vector de String

5. Cadenas

Ejemplo:

Define la cadena “Hola. Este es un mensaje de PRUEBA”:

- Imprime la cadena y su longitud con `printf`.
- Obtener e imprimir el carácter situado en la posición 5.
- Obtener dos subcadenas de s, una desde la posición 6 (s1) y la otra desde la 11 hasta 23 (s2). Imprime con `printf` las dos subcadenas
- Buscar en s las subcadenas sub1 y “ “ (espacio) desde posición 5
- Compara las cadenas s y s1 e indica si son iguales o no
- Compara s1 con s, sub1 con s1 y sub1 con sub e indica cuál es igual, mayor o menor
- Pon s1 en mayúsculas y compárala con s. Luego comparas s1 con s ignorando Mayúsculas
- Muestra la longitud de s y concatena espacios delante y detrás (al menos 5). Imprime la nueva s y aplica el método trim mostrando el resultado.
- Por último, guarda en un vector el resultado de separar la cadena por el carácter espacio “ “ y muestra el resultado.

5. Cadenas

Ejercicios:

Vectores:

337 - La abuela María

171 - Abadías Pirenáicas

376 - Siete picos

Matrices:

151 - ¿Es matriz identidad?

160 – Matrices triangulares

Strings:

369 – Contando en la arena

427 – Yo soy tu…

467 – Polisílaba es polisílaba

446 – Abuelas falsas