

Fechas

LocalDate

-Enseñar la fecha actual:

```
LocalDate localDate = LocalDate.now();
```

```
LocalDate localDateOf = LocalDate.of(2022, 10, 10);
```

-Añadir días:

```
LocalDate datePlus = localDateOf.plusDays(7); (suma 7 días a la fecha)
```

-Restar:

```
LocalDate dateMinus = localDateOf.minusDays(7); (resta 7 días a la fecha).
```

Devolver true/false si una fecha es anterior/posterior a otra:

```
boolean isBefore = LocalDate.of(2022, 10, 10).isBefore(LocalDate.of(2022, 8, 20));  
System.out.println(isBefore); // false  
boolean isAfter = LocalDate.of(2022, 10, 10).isAfter(LocalDate.of(2022, 8, 20));  
System.out.println(isAfter); // true
```

LocalTime

```
LocalTime localTime = LocalTime.now();
```

```
LocalTime hora = LocalTime.of(6, 30); (enseña 6:30)
```

-También se pueden sumar y restar:

```
LocalTime localTimePlus = hour.plus(1, ChronoUnit.HOURS); (devuelve 7:30)
```

```
LocalTime localTimeMinus = hour.minus(60, ChronoUnit.SECONDS); (devuelve 6:29)
```

-Compara tiempos:

```
boolean isBeforeHour = LocalTime.parse("08:30").isBefore(LocalTime.parse("10:20"));
```

LocalDateTime

-Combina ambos.

```
LocalDateTime ahora = LocalDateTime.now(); (devuelve tanto hora como fecha)
```

```
LocalDateTime localDateTimeOf = LocalDateTime.of(2022, Month.AUGUST, 20, 8, 30);  
(selecciona la fecha que queremos)
```

-Sumar/restar tiempo/días:

```
LocalDateTime localDateTimePlus = localDateTimeOf.plusDays(5);
```

```
LocalDateTime localDateTimeMinus = localDateTimePlus.minusMinutes(10);
```

Period

-Obtienes la diferencia entre dos fechas:

```
int diffDays = Period.between(fechaInicio, fechaFin).getDays();
```

Duration

-Funciona igual que el Period para horas.

```
long diffSeconds = Duration.between(startLocalTime, endLocalTime).getSeconds();
```

DateTimeFormatter

-Puedes dar formato a la fecha utilizando la clase DateTimeFormatter:

```
DateTimeFormatter f = DateTimeFormatter.ofPattern(pattern: "dd-MM-yyyy");  
fechaVencimiento = LocalDate.parse(text: teclado.next(), formatter: f);
```

Idioma:

```
String idiomaLocal = System.getProperty("user.language");
```

Arrays

-Declaración:

tipo[] nombre; ----- int[] coches;

-Puedes asignar la longitud ahora y rellenar el array después:

int[] coches = new int[5]; (El array tendrá una longitud de 5)

coches[0] = 23; (el coche en la posición 0 tiene el número 23)

-O declarar directamente el contenido del array:

int [] coches = {23, 32, 41, 2, 18}; (array de 5 valores ya asignados)

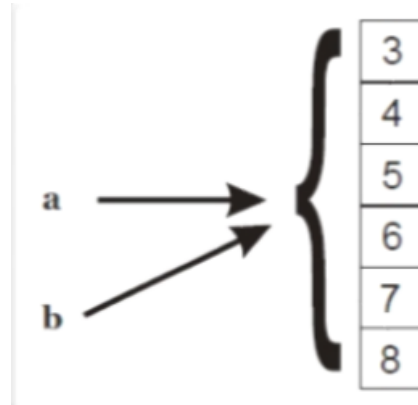
-Para saber la longitud de un array

coches.length (el resultado sería 5)

Asignar dos arrays al mismo dato:

```
int[] a;  
int[] b=new int[]{3,4,5,6,7,8};  
a=b;
```

Si modifico a/b se modificará el otro.



usamos:

-Usar el comparador ==

int a[]={3,3,3};

int b[]={3,3,3};

int c[] = b;

System.out.println(a==b); (es false)

System.out.println(b==c); (es true)

For-each

Aquí “números” sustituye la posición x del array nums y cada vez que completa avanza. No se pueden modificar valores del array.

```
int nums[]={1,2,3,4,5};  
int sum=0;  
for(int numeros:nums){  
    sum+=numeros;  
}
```

Arrays multidimensionales

Arrays de dos dimensiones:

int[][] posicion = new int[3][5]

00	01	02	03	04
10	11	12	13	14
20	21	22	23	24

Array de objeto

Objeto Alumno

atributos: String nombre, int edad

Alumno[] alum = new Alumno("Pedro", 24)

Clase Arrays

La clase Arrays tiene diversos metodos, para ver todos podemos usar:

Arrays.metodós(argumentos);

fill:

-Rellena el array con un valor.

Arrays.fill(coche, 4); (todo el array se rellena con el valor 4)

-También podemos rellenar desde un rango:

Arrays.fill(coche,2,6,-3); (del elemento 2 al 5 se rellena a -3)

sort

-Ordena el array de menor a mayor

Arrays.sort(coche)

-También podemos ordenar desde un rango:

Arrays.sort(coche, 2, 5) (ordena solo los elementos entre 2 y 5 siendo 5 excluyente)

equals

-Comparar dos arrays y devuelve true or false, este si compara el contenido

```
int a[] = {2,3,4,5,6};
int b[] = {2,3,4,5,6};
int c[] = a;
System.out.println(a==b); //false
System.out.println(Arrays.equals(a,b)); //true
System.out.println(a==c); //true
System.out.println(Arrays.equals(a,c)); //true
```

binarySearch

-Busca un elemento dentro de un array ordenado:

Arrays.sort(coche)

Arrays.binarySearch(coche, 23); (Devuelve 0)

copyOf

-Copia un array a otro:

```
int a[] = {1,2,3,4,5,6,7,8,9};
int b[] = Arrays.copyOf(a, a.length); //b es {1,2,3,4,5,6,7,8,9}
int c[] = Arrays.copyOf(a, 12); //c es {1,2,3,4,5,6,7,8,9,0,0,0}
int d[] = Arrays.copyOf(a, 3); //d es {1,2,3}
```

copyOfRange

-Copia entre rangos:

```
int a[] = {1,2,3,4,5,6,7,8,9};
int b[] = Arrays.copyOfRange(a, 3,6); //b vale {4,5,6}
```

Metodo Main

-El método main funciona como un array de Strings:

```
public static void main(String[] args) {
```

Para meter argumentos al array debemos ir a las propiedades del proyecto y separar cada elemento con espacios, todo el texto entre “ ”

