

FECHAS Y HORAS

- Veamos algunas operaciones habituales con estos tipos. Comencemos por la creación de objetos de tipo fecha y hora:

```
LocalDate fecha1 = LocalDate.of(2014, Month.MAY, 23);
LocalDate fecha2 = LocalDate.of(2014, 5, 23);
LocalTime hora1 = LocalTime.of(11, 00);
LocalTime hora2 = LocalTime.of(10, 59, 59);
```

- Además del método of, se pueden crear objetos con la fecha y hora del sistema en el momento en que se invoca al método. Por ejemplo,

```
LocalDate fecha3 = LocalDate.now();
LocalTime hora3 = LocalTime.now();
```

- Dada una fecha o una hora, podemos acceder a una parte de ella, como por ejemplo el día de una fecha o los minutos de una hora:

```
LocalDate hoy = LocalDateTime.now();
Integer dia = hoy.getDayOfMonth();
DayOfWeek diaSemana = hoy.getDayOfWeek();
LocalDate ahora = LocalDateTime.now();
Integer minutos = ahora.getMinute();
```

- También podemos sumar o restar un periodo de tiempo a una fecha u hora:

```
LocalDate f1 = LocalDate.of(2008, Month.FEBRUARY, 29);
LocalDate f2 = f1.plusYears(1);
System.out.println("Un año después..." + f2);
```

- Otra operación habitual es obtener el tiempo transcurrido entre dos fechas:

```
LocalDate fechaNacimiento = LocalDate.of(2013, 12, 3);
LocalDate hoy = LocalDate.now();
Period p = fechaNacimiento.until(hoy);
Period p = Period.between(fechaNacimiento, hoy); // También así
System.out.println ("El periodo entre las dos fechas es " + p);
System.out.println ("Los años del periodo son " + p.getYears());
System.out.println ("Los meses del periodo son " + p.getMonths());
System.out.println ("Los días de periodo son " + p.getDays());
```

- De manera similar para las horas, pero usando el tipo Duration:

```
LocalTime hora1 = LocalTime.of(15, 30);
LocalTime hora2 = LocalTime.of(15, 45);
Duration d = Duration.between(hora1, hora2);
System.out.println("Duración - " + d);
System.out.println ("Los segundos de la duración son " +
d.getSeconds());
```

- Para comparar dos fechas u horas, utilizamos los métodos isBefore, isAfter e isEqual:

```
LocalDate f1 = LocalDate.of (2016, Month.SEPTEMBER, 19);
LocalDate f2 = LocalDate.of (2017, Month.JANUARY, 13);
System.out.println("¿Es f1 anterior a f2? " + f1.isBefore(f2));
System.out.println("¿Es f1 posterior a f2? " + f1.isAfter(f2));
System.out.println("¿Es f1 igual a f2? " + f1.isEqual(f2));
```

- Finalmente, cuando mostramos fechas y horas en la pantalla, queremos que aparezcan en un formato concreto. Los siguientes ejemplos muestran varias formas de hacerlo:

```
LocalDate fecha = LocalDate.of(2016, 12, 3);
String fechaFormateada = fecha.format(DateTimeFormatter.ofPattern("dd-MM-yyyy"));
System.out.println("Fecha: " + fechaFormateada);
LocalTime hora = LocalTime.now();
String horaFormateada = hora.format(DateTimeFormatter.ofPattern("HH:mm:ss:n"));
System.out.println("Hora: " + horaFormateada);
```

- La salida que se produce al ejecutar el código anterior es la siguiente:

```
Fecha: 03-12-2016
Hora: 19:00:19:56000000
```

-