

# Trabajo con Fechas (Programación)

---

## 1. `createDate`

### Descripción:

Crea un objeto `Date` representando el instante actual del sistema.

### Parámetros de entrada:

- Ninguno

### Ejemplo de comportamiento:

- Entrada: N/A
  - Salida: `Thu Jan 17 12:00:00 GMT 2025` (o la fecha/hora actual al momento de la ejecución).
- 

## 2. `compareDates`

### Descripción:

Compara dos fechas y determina si la primera es anterior a la segunda.

### Parámetros de entrada:

- `date1` (`Date`): Fecha a comparar.
- `date2` (`Date`): Fecha de referencia.

### Ejemplo de comportamiento:

- Entrada: `date1 = 2025-01-01, date2 = 2025-01-15`
  - Salida: `true`
  - Entrada: `date1 = 2025-01-20, date2 = 2025-01-15`
  - Salida: `false`
- 

## 3. `addDaysToDate`

### Descripción:

Suma un número específico de días a un objeto `Date`.

### Parámetros de entrada:

- `date` (`Date`): Fecha base.
- `days` (`int`): Número de días a agregar (puede ser negativo).

### Ejemplo de comportamiento:

- Entrada: `date = 2025-01-01, days = 10`

- Salida: 2025-01-11
  - Entrada: `date = 2025-01-01, days = -5`
  - Salida: 2024-12-27
- 

## 4. formatDate

### Descripción:

Convierte un objeto `Date` a una cadena de texto con formato `yyyy-MM-dd`.

### Parámetros de entrada:

- `date (Date)`: Fecha a formatear.

### Ejemplo de comportamiento:

- Entrada: `date = 2025-01-15`
  - Salida: "2025-01-15"
- 

## 5. calculateDateDifference

### Descripción:

Calcula la diferencia en días entre dos fechas.

### Parámetros de entrada:

- `date1 (Date)`: Primera fecha.
- `date2 (Date)`: Segunda fecha.

### Ejemplo de comportamiento:

- Entrada: `date1 = 2025-01-01, date2 = 2025-01-15`
  - Salida: 14
  - Entrada: `date1 = 2025-01-15, date2 = 2025-01-01`
  - Salida: 14
- 

## 6. getCurrentLocalDate

### Descripción:

Obtiene la fecha actual utilizando `LocalDate`.

### Parámetros de entrada:

- Ninguno

### Ejemplo de comportamiento:

- Entrada: N/A
  - Salida: 2025-01-17 (o la fecha actual al momento de la ejecución).
- 

## 7. createSpecificLocalDate

### Descripción:

Crea un objeto `LocalDate` con un año, mes y día específicos.

### Parámetros de entrada:

- `year (int)`: Año.
- `month (int)`: Mes (1-12).
- `day (int)`: Día del mes (1-31).

### Ejemplo de comportamiento:

- Entrada: `year = 2025, month = 1, day = 15`
  - Salida: 2025-01-15
- 

## 8. addDays

### Descripción:

Suma un número específico de días a un objeto `LocalDate`.

### Parámetros de entrada:

- `date (LocalDate)`: Fecha base.
- `days (int)`: Número de días a agregar (puede ser negativo).

### Ejemplo de comportamiento:

- Entrada: `date = 2025-01-01, days = 10`
  - Salida: 2025-01-11
  - Entrada: `date = 2025-01-01, days = -5`
  - Salida: 2024-12-27
- 

## 9. isBefore

### Descripción:

Determina si una fecha `LocalDate` es anterior a otra.

### Parámetros de entrada:

- `date1 (LocalDate)`: Fecha a comparar.
- `date2 (LocalDate)`: Fecha de referencia.

### Ejemplo de comportamiento:

- Entrada: `date1 = 2025-01-01, date2 = 2025-01-15`
  - Salida: `true`
  - Entrada: `date1 = 2025-01-20, date2 = 2025-01-15`
  - Salida: `false`
- 

## 10. `calculatePeriodBetween`

### Descripción:

Calcula la diferencia en años, meses y días entre dos fechas `LocalDate`.

### Parámetros de entrada:

- `start` (`LocalDate`): Fecha inicial.
- `end` (`LocalDate`): Fecha final.

### Ejemplo de comportamiento:

- Entrada: `start = 2020-01-01, end = 2025-01-01`
  - Salida: `5 años, 0 meses, 0 días`
  - Entrada: `start = 2025-01-01, end = 2024-01-01`
  - Salida: `-1 años, 0 meses, 0 días`
- 

## 11. `getCurrentLocalTime`

### Descripción:

Obtiene la hora actual utilizando `LocalTime`.

### Parámetros de entrada:

- Ninguno

### Ejemplo de comportamiento:

- Entrada: N/A
  - Salida: `12:00` (o la hora actual al momento de la ejecución).
- 

## 12. `addMinutes`

### Descripción:

Suma un número específico de minutos a un objeto `LocalTime`.

### Parámetros de entrada:

- `time` (`LocalTime`): Hora base.
- `minutes` (`int`): Minutos a agregar (puede ser negativo).

**Ejemplo de comportamiento:**

- Entrada: `time = 10:30, minutes = 45`
  - Salida: `11:15`
  - Entrada: `time = 10:30, minutes = -45`
  - Salida: `09:45`
- 

## 13. `isTimeBefore`

**Descripción:**

Determina si una hora `LocalTime` ocurre antes que otra.

**Parámetros de entrada:**

- `time1 (LocalTime)`: Hora a comparar.
- `time2 (LocalTime)`: Hora de referencia.

**Ejemplo de comportamiento:**

- Entrada: `time1 = 10:30, time2 = 11:00`
  - Salida: `true`
  - Entrada: `time1 = 12:30, time2 = 11:30`
  - Salida: `false`
- 

## 14. `getCurrentLocalDateTime`

**Descripción:**

Obtiene la fecha y hora actual utilizando `LocalDateTime`.

**Parámetros de entrada:**

- Ninguno

**Ejemplo de comportamiento:**

- Entrada: N/A
  - Salida: `2025-01-17T12:00` (o la fecha y hora actual al momento de la ejecución).
- 

## 15. `addHours`

**Descripción:**

Suma un número específico de horas a un objeto `LocalDateTime`.

**Parámetros de entrada:**

- `dateTime` (`LocalDateTime`): Fecha y hora base.
- `hours` (`int`): Número de horas a agregar (puede ser negativo).

**Ejemplo de comportamiento:**

- Entrada: `dateTime = 2025-01-01T10:00`, `hours = 5`
  - Salida: `2025-01-01T15:00`
  - Entrada: `dateTime = 2025-01-01T10:00`, `hours = -5`
  - Salida: `2025-01-01T05:00`
- 

## 16. `subtractMonths`

**Descripción:**

Resta un número específico de meses a un objeto `LocalDateTime`.

**Parámetros de entrada:**

- `dateTime` (`LocalDateTime`): Fecha y hora base.
- `months` (`int`): Número de meses a restar.

**Ejemplo de comportamiento:**

- Entrada: `dateTime = 2025-06-15T10:00`, `months = 2`
  - Salida: `2025-04-15T10:00`
  - Entrada: `dateTime = 2025-01-15T10:00`, `months = 2`
  - Salida: `2024-11-15T10:00`
- 

## 17. `getZonedDateTime`

**Descripción:**

Obtiene la fecha y hora actuales en una zona horaria específica.

**Parámetros de entrada:**

- `zoneId` (`String`): Identificador de zona horaria (por ejemplo, `"Europe/Madrid"`).

**Ejemplo de comportamiento:**

- Entrada: `zoneId = "Europe/Madrid"`
  - Salida: `2025-01-17T12:00+01:00[Europe/Madrid]`
  - Entrada: `zoneId = "America/New_York"`
  - Salida: `2025-01-17T06:00-05:00[America/New_York]`
-

## 18. `convertZone`

### Descripción:

Convierte un objeto `ZonedDateTime` de una zona horaria a otra.

### Parámetros de entrada:

- `dateTime` (`ZonedDateTime`): Fecha y hora en la zona horaria original.
- `targetZoneId` (`String`): Identificador de la zona horaria de destino.

### Ejemplo de comportamiento:

- Entrada: `dateTime = 2025-01-17T12:00+01:00[Europe/Madrid]`, `targetZoneId = "America/New_York"`
  - Salida: `2025-01-17T06:00-05:00[America/New_York]`
- 

## 19. `getFirstDayOfMonth`

### Descripción:

Obtiene el primer día del mes de una fecha `LocalDate` dada.

### Parámetros de entrada:

- `date` (`LocalDate`): Fecha base.

### Ejemplo de comportamiento:

- Entrada: `date = 2025-01-15`
  - Salida: `2025-01-01`
  - Entrada: `date = 2025-12-31`
  - Salida: `2025-12-01`
- 

## 20. `getLastDayOfMonth`

### Descripción:

Obtiene el último día del mes de una fecha `LocalDate` dada.

### Parámetros de entrada:

- `date` (`LocalDate`): Fecha base.

### Ejemplo de comportamiento:

- Entrada: `date = 2025-01-15`
- Salida: `2025-01-31`
- Entrada: `date = 2024-02-15`

- Salida: `2024-02-29` (año bisiesto).
- 

## 21. `isLeapYear`

### Descripción:

Verifica si el año de una fecha `LocalDate` dada es bisiesto.

### Parámetros de entrada:

- `date` (`LocalDate`): Fecha base.

### Ejemplo de comportamiento:

- Entrada: `date = 2024-01-01`
  - Salida: `true`
  - Entrada: `date = 2025-01-01`
  - Salida: `false`
- 

## 22. `measureExecutionTime`

### Descripción:

Mide el tiempo de ejecución de una tarea representada como un objeto `Runnable`.

### Parámetros de entrada:

- `task` (`Runnable`): Tarea cuya duración se medirá.

### Ejemplo de comportamiento:

- Entrada: `task = () -> Thread.sleep(100)`
  - Salida: Duración de aproximadamente 100 ms.
- 

## 23. `parseDate`

### Descripción:

Convierte una cadena de texto en un objeto `LocalDate` utilizando un formato específico.

### Parámetros de entrada:

- `date` (`String`): Fecha en formato texto.
- `format` (`String`): Formato esperado de la fecha (por ejemplo, `"yyyy-MM-dd"`).

### Ejemplo de comportamiento:

- Entrada: `date = "2025-01-15", format = "yyyy-MM-dd"`
- Salida: `2025-01-15`



- Entrada: `date = "15-01-2025", format = "yyyy-MM-dd"`
- Salida: Lanzará una excepción `DateTimeParseException`.