

Clases Date y Calendar en Java

Clase Date

La clase `Date`, representa como su nombre indica a `una fecha`.

Viene incluida en el `JDK` (`java.util.Date`), pero debemos importarla para utilizarla en nuestro código:

```
import java.util.Date;
```

Para crear una fecha deberemos instanciar un objeto de esta clase:

```
Date miFecha = new Date();
```

Si no le pasamos ningún parámetro al constructor “miFecha” será la fecha de hoy.

Clase Date

También podemos crear una fecha determinada pasándole al constructor los datos, por ejemplo 2 de Julio de 2022:

```
Date miFecha = new Date(122,6,2);
```

- 122: ¿porque 122? Esto es el año, tenemos que tener en cuenta que inicialmente empieza en 1900 y lo que hace es sumarle el valor que le demos $1900+122 = 2022$. Si ponemos 2022 en realidad, sería el año 3922.
- 6: este es el mes, pero si es el 6 ¿No debería ser Junio? No, porque empieza en 0 por lo que el 6 sería el séptimo mes, en este caso Julio.
- 2: este es el día, aquí es el día tal cual lo ponemos.

Date.getTime()

El método `getTime` devuelve un tipo `long` que representa una fecha y que son los milisegundos transcurridos desde el 1 de enero de 1970 hasta hoy (Unix Time)

Realmente el Unix Time son los segundos transcurridos (<https://www.unixtimestamp.com/>)

```
Date ahora = new Date();
```

```
long milisegundosTranscurridosDesde111970 = ahora.getTime();
```

Formatear una fecha con la clase `SimpleDateFormat`

Con la clase `SimpleDateFormat` podemos crear un formato de fecha para aplicarla a un objeto `Date`.

```
SimpleDateFormat miDateFormat = new SimpleDateFormat("dd MMMM yyyy");  
String fechaFormateada = miDateFormat.format(miFecha);
```

El método `format` de un objeto `SimpleDateFormat` devuelve un `String` con el formato de fecha indicado en el patrón pasado en el constructor `"dd MMMM yyyy"`.

Patrones de SimpleDateFormat


Existen multitud de patrones para dar formato a una fecha:

(<https://docs.oracle.com/en/java/javase/17/docs/api/java.base/java/text/SimpleDateFormat.html>)

Pero los más utilizados son:

Patrón	Resultado
dd	días
MM	Mes (número)
MMMM	Mes (nombre)
yy	Año corto
yyyy	Año completo

Ejemplos de Patrones de SimpleDateFormat

Patrón	Resultado
dd MMMM yyyy	05 abril 2022
dd-MM-yy	05-04-22
dd/MMMM/yy	05/abril/22
dd 'de' MMMM 'de' yyyy	05 de abril de 2022
dd de MMMM de yyyy	Error: patrón ilegal 

Clase Calendar

La clase `Calendar`, nos permite trabajar con fechas de una manera distinta a lo que lo haríamos con la clase `Date`.

Viene incluida en el `JDK` (`java.util.Calendar`), pero debemos importarla para utilizarla en nuestro código:

```
import java.util.Calendar;
```

Al ser un clase abstracta no podemos instanciarla pero podemos crear un calendario con el método `getInstance`:

```
Calendar miCalendario = Calendar.getInstance();
```

Si no le pasamos ningún parámetro al constructor obtendremos por defecto un calendario Gregoriano.

Clase `GregorianCalendar`

También podemos crea un calendario de esta forma:

```
Calendar miCalendario = new GregorianCalendar(2022,7,5);
```

La clase `GregorianCalendar` hereda de la clase `Calendar`. Debemos importarla para utilizarla en nuestro código:

```
import java.util.GregorianCalendar;
```

Métodos de la clase `Calendar`

Existen varios métodos para utilizar con objetos de la clase `Calendar`. Pero los más utilizados son:

Método	Qué hace
<code>getTime()</code>	Convierte un objeto <code>Calendar</code> en un objeto <code>Date</code> - <code>objetoCal.getTime()</code>
<code>setTime()</code>	Setea un objeto <code>Date</code> en un objeto <code>Calendar</code> - <code>objetoCal.setTime(date)</code>
<code>get(...)</code>	Obtener el valor de una fecha de un objeto <code>Calendar</code>
<code>set(...)</code>	Modificar el valor de una fecha de un objeto <code>Calendar</code>
<code>add(...)</code>	Para añadir o restar un valor específico de tiempo a una fecha <code>Calendar</code>