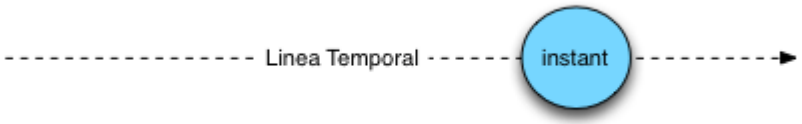
Llevamos mas de de una década trabajando con la clase Date y la clase Calendar en Java . Mucha gente cuando empieza se ha encontrado con verdaderos problemas a la hora de gestionar estas clases unidas a la clase DateFormat. A partir de Java 8 las cosas se van a simplificar ya que tenemos un nuevo API de DateTime mucho más intuitivo. Una de las clases fundamentales a la hora de trabajar con Java 8 es la clase Instant que define un instante en el tiempo.



Vamos a ver como funciona en el código :

```
package com.arquitecturajava.fechas;

import java.time.Duration;
import java.time.Instant;

public class PrincipalInstante {

 public static void main(String[] args) {

 Instant instanteA=Instant.now();

 try {
 Thread.sleep(3000);
 } catch (InterruptedException e) {
 // TODO Auto-generated catch block
 e.printStackTrace();
```

```
 }

 Instant instanteB= Instant.now();
 System.out.println(Duration.between(instanteA, instanteB));

 }

}
```

Su funcionamiento es trivial ya que especifica un punto concreto en la linea de tiempo concreto con el cual podemos trabajar. En este caso nos imprimirá "PT3.011S" que es el tiempo transcurrido entre ambos instantes.

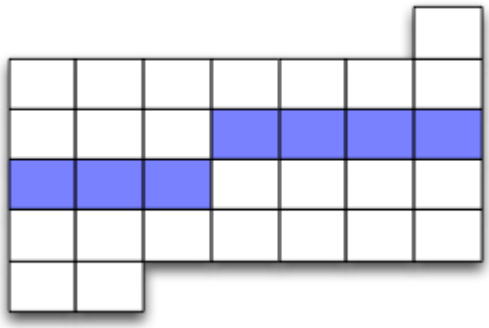
Java 8 Date Time y LocalTime

La otra clase importante y que viene a substituir al Calendario es LocalDate que nos permite definir fechas y trabajar con ellas de una forma bastante más directa que el Calendario.

```
package com.arquitecturajava.fechas;

import java.time.LocalDate;
import java.time.temporal.ChronoField;

public class PrincipalLocalDate {

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

```
 LocalDate finAño=LocalDate.of(214, 12, 31);

 LocalDate navidad=finAño.minusDays(6);
 System.out.println(navidad.get(ChronoField.DAY_OF_MONTH));

 }
}
```

El programa imprimirá 25 por pantalla. Por último otra de las clases que me ha parecido que aporta es la clase Period que define un intervalo de tiempo entre dos fechas y nos permite trabajar con el de forma sencilla.



Vamos a verla en código:

```
package com.arquitecturajava.fechas;

import java.time.LocalDate;
import java.time.Period;

public class PrincipalPeriodo {
```

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

LocalDate fechaA = LocalDate.of(1978, 8, 26);
LocalDate fechaB = LocalDate.of(1988, 9, 28);
Period period = Period.between(fechaA, fechaB);
System.out.printf("Periodo %s y %s"
+ "hay %d años, %d meses"
+ " y %d dias%n", fechaA, fechaB,
period.getYears(),
period.getMonths(),
period.getDays());
}
}
```

Esto nos imprimirá el siguiente mensaje por pantalla :

Periodo 1978-08-26 y 1988-09-28hay 10 años, 1 meses y 2 dias

Estos cambios los pedíamos todos los desarrolladores hace tiempo y por fin han llegado. Esperemos que este API ayude a todo el mundo a trabajar con las fechas de una forma más natural.

Otros artículos relacionados

- ForEach vs Iterator
- Expresiones Lambda
- JavaStreams