

En Java, la clase `Java Period` es parte del paquete `java.time` introducido en Java 8. Representa un período de tiempo en años, meses y días (en ese orden). `Period` te permite realizar operaciones y cálculos en fechas basados en estas unidades de tiempo.

Puedes crear una instancia de `Java Period` utilizando el método `of` que hace de factoría para construir un nuevo objeto pasándole como argumentos los años , meses y días que deseamos seleccionar para ese periodo de tiempo concreto. Una vez realizada esta primera operación lo siguiente es usar la clase `LocalDate` para generar una fecha a la cual añadir el periodo.

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

public class Main {
    public static void main(String[] args) {
        // Crear un período de 0 años, 2 meses y 3 días
        Period periodo = Period.of(0, 2, 3);

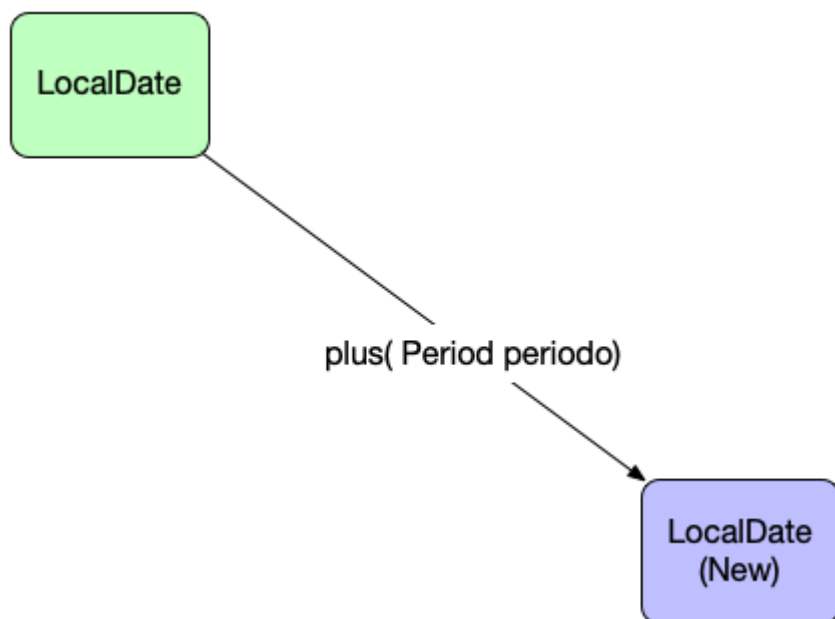
        // Aplicar el período a una fecha
        LocalDate fecha = LocalDate.now();
        // sumamos el los meses y los días con el método plus
        LocalDate nuevaFecha = fecha.plus(periodo);

        System.out.println("Fecha actual: " + fecha);
        System.out.println("Fecha después de aplicar el período: " +
nuevaFecha);
    }
}
```

Java Period y LocalDate Plus

Como se puede observar usamos el método `plus` de `LocalDate` para añadir ese periodo a la

fecha . Eso si recordemos que LocalDate es una clase inmutable así que en vez de simplemente añadir el periodo nos devolverá un nuevo objeto LocalDate con el Periodo añadido.



El resultado lo podemos ver en la consola :

Fecha actual: 2023-06-06

Fecha después de aplicar el período: 2023-08-09

Otros artículos relacionados

- [Java LocalDate y el manejo de Fecha](#)
- [Java Fechas y el principio SRP](#)
- [JPA Remove y Objetos gestionados](#)
- [Java 8 Date Time API](#)
- [Java String to Date utilizando Java 8](#)
- [Java 8](#)