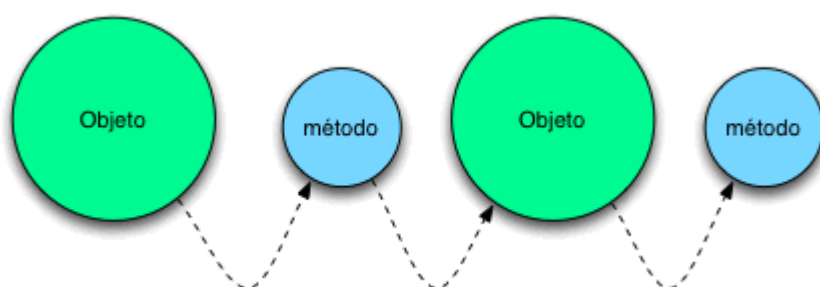


Trabajamos todos los días con Java y uno de las cosas que uno echa de menos a veces es que existan más interfaces fluidas a nivel del lenguaje. ¿Qué es una fluid interface? . Son interfaces o clases que cuando invocamos a un método concreto nos devuelve el mismo objeto modificado . De tal forma que podemos volver a solicitar otro método del mismo objeto y encadenar más operaciones.



Fluid interfaces y StringBuilder

Una de las clases más habituales a la hora de usar programación fluida es StringBuffer vamos a ver un ejemplo sencillo.

```
package com.arquitecturajava;

public class Principal {

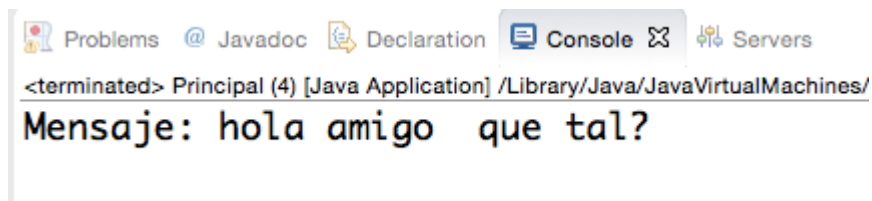
    public static void main(String[] args) {

        StringBuilder cadena = new StringBuilder("?hola");
        cadena.delete(0, 1).append(" que tal?").insert(4, " amigo ");
        System.out.println("Mensaje: " + cadena);
    }
}
```

```
}
```

```
}
```

El resultado será:



Fluid interfaces y LocalDateTime

Otra de las clases que soporta este tipo de interface es LocalDateTime vamos a verlo:

```
package com.arquitecturajava;

import java.time.LocalDateTime;
import java.time.temporal.ChronoUnit;

public class Principal2 {

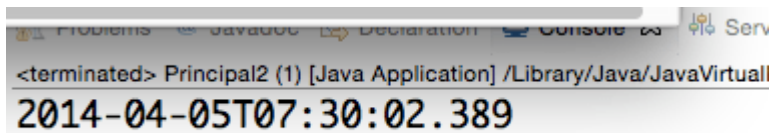
    public static void main(String[] args) {

        LocalDateTime fecha = LocalDateTime.now()
            .withDayOfMonth(1).withYear(2014)
            .plusWeeks(5).minus(3, ChronoUnit.HOURS);
        System.out.println(fecha);
    }
}
```

```
}
```

```
}
```

El resultado será:



The screenshot shows a console window with the following text: `<terminated> Principal2 (1) [Java Application] /Library/Java/JavaVirtualMachines/1.8.0_65-b12/Contents/Home/bin/java -Djava.class.path=. Principal2` followed by the timestamp `2014-04-05T07:30:02.389`.

Como podemos ver las APIs son muy sencillas y directas de utilizar lo cual ayuda sobremanera a los desarrolladores.

Otros artículos relacionados: [Java BigInteger](#) , [Java Equals and hashCode](#) , [Java8 Date Time API](#)