



Mejora de uso de condicionales en java

Autor: Diego Hernández

Cláusulas de guarda

Es posible que para hacer una estructura condicional hagamos algo como lo siguiente

```
if( conditionA ){
  if(conditionB){
    if(conditionC){
      console.log("Instrucción realizada");
    }
  }
}
```

Este código pese a ser funcional, es difícil de leer y para escalarlo tendremos muchas dificultades (en caso de necesitar hacerlo).

Otra forma común es con una condicional múltiple "switch"

```
switch( variable ){
  case 1:{
    console.log("Instrucción 1");
    break;
  }
  case 1:{
    console.log("Instrucción 2");
    break;
  }
  default :{
    console.log("Instrucción 3");
    break;
  }
}
```

Si bien es una forma correcta de conseguir un objetivo, tenemos líneas redundantes sobre todo con "break".

Para mejorar esta escritura de código podemos utilizar las **cláusulas de guarda**, son un método que nos permitirá hacer código más *legible*, *escalable* y *testable*.

Aplicando cláusulas de guarda

Aunque suene difícil, son bastante sencillas de aplicar y su objetivo es reducir el nivel de indentación en el código. Para ello debemos destacar las condiciones más significativas para realizar cierto procedimiento. Veremos una condicional con los ejemplos anteriores y después otra con la cláusula de guarda.

Sin cláusula de guarda

```
if( Login ){
    console.log("Iniciar sesión");
} else {
    console.log("No se inicio sesión");
}
```

Con cláusula de guarda

```
if( Login ){
    console.log("Iniciar sesión");
    return ;
}
console.log("No se inicio sesión");
```

Es un ejemplo simple, pero vemos ya una diferencia en la limpieza del código. Rescatando del ejemplo anterior, tomamos una condicional y en caso de cumplirse detenemos el flujo de ejecución en la función con el uso de la palabra reservada **return**. Veamos un ejemplo, pero con más condiciones aun.

- *Sin cláusula de guarda*

```
if( Login ){
    console.log("Iniciar sesión");
} else if (!Login && Tries){
    console.log("Limite de intentos alcanzado");
} else {
    console.log("No se inició sesión");
}
```

- Con cláusula de guarda

```
if( Login ){  
    console.log("Iniciar sesión");  
    return ;  
}  
if ( Tries ){  
    console.log("Limite de intentos alcanzado");  
    return ;  
}  
console.log("No se inició sesión");
```

En este ejemplo agregamos una variable adicional para saber si los intentos se han superado y en el ejemplo sin cláusula de guarda vemos que ya hay un poco de redundancia en la variable login y Enel segundo ejemplo tenemos más visibilidad en esta variable, pues si no se cumple todos los demás casos son forzosamente cuando esta variable no es válida y tenemos más limpieza en el código.