

EJERCICIOS DE REPASO UNIDAD 2

1.- Corrige el siguiente código y explica el porqué de los errores:

CÓDIGO CON ERRORES

```
public class Casting1 {  
    public static void main(String[] args) {  
  
        byte edadJuan=20;  
        byte edadPedro=edadJuan+1;  
  
        short sueldoBase= 1980;  
        short complementos= 400;  
  
        short sueldoTotal;  
        sueldoTotal=sueldoBase+complementos;  
    }  
}
```

CÓDIGO CORREGIDO CON CASTING

```
public class Casting1 {  
    public static void main(String[] args) {  
  
        byte edadJuan=20;  
        byte edadPedro= (byte) edadJuan + 1;  
  
        short sueldoBase= 1980;  
        short complementos= 400;  
  
        short sueldoTotal;  
        sueldoTotal= (short) sueldoBase+complementos;  
    }  
}
```

Según la [documentación de Java](#), la JVM no es capaz de hacer operaciones aritmeticas con tipos numericos más pequeños que el int, así que al intentar sumar estos números, Java los convierte automáticamente en enteros. Para arreglar lo tenemos dos soluciones, podemos utilizar un casting o podemos cambiar el tipo de dato a integer.

2.- ¿Por qué se usa más a menudo el tipo int cuando por ahorro de memoria serían más adecuados otros tipos?

Existen dos motivos de comodidad y otro de rendimiento, como bien hemos visto en el ejercicio anterior, la JVM tiene limitaciones a la hora de trabajar con tipos más pequeños que el int, esto obliga a los programadores que utilizan estos tipos de datos a utilizar castings.

Otro motivo es que la mayoría de métodos que vienen con Java están preparados para recibir y devolver datos de tipo entero, nuevamente los programadores tendrían que estar constantemente utilizando castings.

En cuanto a rendimiento se refiere, actualmente la mayoría de CPU están preparadas para funcionar con instrucciones de 64 bits, los tipos short y byte utilizan instrucciones de 32 bits, por esto, aunque a simple vista estos dos tipos son más ligeros, a las CPU modernas les resulta más eficiente trabajar con instrucciones "nativas" (es decir de 64 bits). No obstante el tipo int utiliza más espacio en memoria RAM.

En conclusión, es más cómodo y seguro para el programador utilizar el tipo int, además en algunos casos puede otorgar un mayor rendimiento en la CPU.