***Tecnológico de Costa Rica***

***Sede Regional San Carlos***

***Escuela Ingeniería en Computación***

***Desarrollo de Aplicaciones para Dispositivos Móviles***

***Tarea #2***

***“Investigación e Internacionalización ”***

***Responsable:***

***Henry Solís Chacón 2013085706***

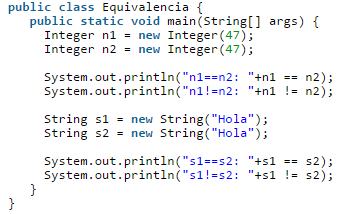
***Hellen Rojas Rojas 2013083934***

***11 de setiembre del 2015***

***Santa Clara, San Carlos***

**Diferencia entre .equals() y “==”**

**Ejemplo:**



**Resultado al ejecutarlo:**

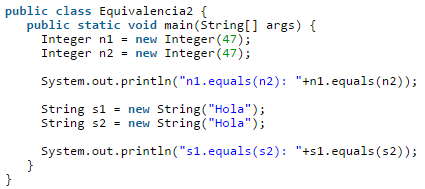
n1==n2: false  
n1!=n2: true  
s1==s2: false  
s1!=s2: true

*Tomado de “Piensa en java desde cero”*

El operador **==** realiza la comparación a nivel de objeto. Esto se debe a que los operadores **==** comparan referencias o sea la ubicación en memoria del objeto y no los valores (Letras, números...). La referencia de n1 está almacenada en un lugar de la memoria y la de n2 en otra, lo mismo para s1 y s2.

Recuerda que cada vez que se crea un objeto con new se le asigna un espacio de

memoria.



**Resultado al ejecutarlo:**

n1.equals(n2): true  
s1.equals(s2): true

*Tomado de “Piensa en java desde cero”*

Para comparar objetos, no de origen de memoria o posteriores, tenemos el método **equals()**, que se puede utilizar siempre, ya que está incluido en la clase java.lang.Object que por defecto heredan todas las clases. Este método compara valores en lugar de referencias, como en el ejemplo anterior. Donde lo que comparamos es **“Hola”** y **47**

**References**

Felinfo: Java, Linux, Virtualización. Open Source. (n.d.). Retrieved March 08, 2016, from http://felinfo.blogspot.com/2013/04/diferencia-entre-comparar-con-y-con.html

Piensa En Java Desde Cero. (n.d.). Retrieved March 08, 2016, from http://piensaenjavadesdecero.blogspot.com/2011/02/diferencias-entre-metodo-equals-y.html