

UNIDAD TEMÁTICA 1: Introducción a JAVA

PRACTICOS DOMICILIARIOS INDIVIDUALES #7

EJERCICIO #1

Observa el siguiente código:

```
String s = "1";
while (s != "1000") {
    s += "0";
}
```

El mismo representa un bucle infinito ya que la comparación con el operador == entre el objeto s y el literal "1000" jamás devuelve True.

- ¿Por qué esto es así? ¿Qué diferencia representa con otros lenguajes como C#?
Esto pasa ya que en Java el "==" solo se usa para tipos primitivos. Para comparar Strings u otros objetos, se debe usar el método .equals();

EJERCICIO #2

Observa los siguientes fragmentos de código:

1.

```
String s1 = "Hola";
String s2 = "Hola";
if ( s1 == s2 ) {
    System.out.println( "True" );
} else {
    System.out.println( "False" );
}
```

2.

```
String s1 = new String("Hola");
String s2 = "Hola";
if ( s1 == s2 ) {
    System.out.println( "True" );
} else {
    System.out.println( "False" );
}
```

- ¿Qué sentencia se ejecuta como resultado del If en cada caso y por qué? Busca en internet referencias que te permitan explicar las respuestas a las preguntas anteriores. Asegúrate que las mismas sean autoritativas y cítalas adecuadamente en tu respuesta.
- Propone una versión correcta del código del Ejercicio #1.

Ayuda: investiga cómo funciona el método intern() de la clase String.

a)

En el caso 1 se ejecuta la sentencia "True", mientras que en el caso 2, imprime "False". Esto se da ya que Java tiene una tabla interna que almacena cadenas (método intern), con el fin de ahorrar memoria. En el caso 1, ambas variables apuntan al mismo espacio en memoria, es decir, al String "Hola" almacenado en la tabla de cadenas interna. Mientras que el segundo caso, crea un nuevo espacio en memoria para guardar una String, por ende, devuelve como resultado "False", ya que apuntan a distintos lugares de la memoria.

b) En el ejercicio #1 se debería cambiar el "(s != "1000")" por "s.equals("1000");"