	<b>VICERRECTORADO DOCENTE</b>	<b>Código:</b> GUIA-PRL-001
	CONSEJO ACADÉMICO	<b>Aprobación:</b> 2016/04/06
<b>Formato:</b> Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

<b>CARRERA:</b> Ing. Computación.		<b>ASIGNATURA:</b> Programación Aplicada.
<b>NRO. PRÁCTICA:</b>	1	<b>TÍTULO PRÁCTICA:</b> Investigación de las versiones de java.
<b>OBJETIVO ALCANZADO:</b> Identificar los cambios importantes de Java Diseñar e Implementar las nuevas técnicas de programación Entender la cada una de las características nuevas en Java		
<b>ACTIVIDADES DESARROLLADAS</b>  <h2 style="text-align: center;">Java 11.</h2> <p>Lanzada en septiembre del 2018 es la versión LTS(Long Term Suport ). Esto significa que tendrá soporte y actualizaciones durante 3 años, en lugar de solo 6 como el resto de las versiones no LTS. Además a partir de Java 11 en adelante Oracle no proveerá la JDK(Java Development Kit) de forma libre para uso comercial. Es decir, si deseas usar Java 11 para uso comercial deberás pagar una licencia pero claro eso no implica que java deje de ser gratuito. Considerando que las versiones anteriores como java 8,9,10 no deberás pagar nada.</p> <p>Java String.</p> <p>Java 11 introduce algunos métodos nuevos para nuestros Strings tales como:</p> <ul style="list-style-type: none"> <li>- <code>isBlank()</code>. Devuelve true si la cadena esta vacia o solamente contiene espacios en blanco.No podemos confundirnos con <code>isEmpty()</code>, que devuelve true solo si la longitud de la cadena es 0.  Var k = "";  System.out.println(k.isBlanck());//true  System.out.println(k.isEmpty());//false</li> <li>- <code>Stream lines()</code>. Devuelve un streams con las líneas que forman la cadena.</li> <li>- <code>Strip()</code>. Remueve los caracteres en blanco. Devolviéndonos una nueva cadena resultando de eliminar espacios a izquierda y derecha. Se diferencia con <code>trim</code> en el concepto de espacios en blanco. <ul style="list-style-type: none"> <li>• <code>String stripLeading()</code>. Devuelve una nueva cadena resultante de eliminar espacios a izquierda</li> <li>• <code>String stripTrailing()</code>. Devuelve una nueva cadena resultante de eliminar espacios a derecha.</li> </ul> </li> <li>- <code>Character.isWhitespace()</code> nos permite saber si un carácter es considerado un blanco.</li> <li>- <code>String repeat(int n)</code>. Repite el string las veces que se le indique, devolviéndonos una cadena resultado de concatenar la cadena actual, tantas veces como se indica en el parámetro.</li> </ul>		

## Comparación de StringBuilder.

Desde java 11 es posible comparar dos StringBuilder sin necesidad de transformarlos en String, ya que la clase dispone del método `compareTo`:

- `int compareTo(StringBuilder another)`/ devuelve 1 si la cadena pasada como parámetro es menor(lexicográficamente) que la principal, -1 si es mayor y 0 si son iguales.

## Variables Locales para los parámetros Lambda

En Java 10 se introdujo el uso de **var** con inferencia del tipo. Ahora podemos usar esto en la declaración de lambdas.

```
1 (var s1, var s2) -> s1 + s2
```

Esto a su vez permite el uso de anotaciones en los parámetros de la función por ejemplo `@NotNull`

```
1 (@NotNull var s1, @NotNull var s2) -> s1 + s2
2
3 @NotNull var foo = new Foo()
```

Si se usa **var** se debe usar en todas las variables. Es decir no podríamos hacer esto combinando un var con un int por ejemplo

```
1 (var s1, int s2) -> s1 + s2 // no compila
```

## Lectura y escritura de String en archivos


En Java 11 se agregaron estos dos métodos que nos simplificarán la lectura y escritura de texto sobre archivos.

`String readString(Path path)`. Devuelve un String con el contenido completo de un fichero.

- `readString(Path)`
- `readString(Path, Charset)`

`Path writeString(Path path, CharSequence csq, OpenOptions)`. Escribe en el fichero la secuencia de caracteres especificada en el segundo parámetro utilizando codificación UTF-8.

- `writeString(Path, CharSequence, OpenOption...)`
- `writeString(Path, CharSequence, Charset, OpenOption...)`

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Ejemplo.

```
String uri = "c:\\temp\\datos.tct";
Path path = Paths.get(uri);
String cad = "Cadena de prueba para fichero";
Files.writeString(path, cad);
```

## Acceso a clases internas

En Java 11 se agregaron nuevos métodos para acceder via reflexión a los métodos internos.

La clase `java.lang.Class` introduce tres métodos `getNestHost()`, `getNestMembers()`, y `isNestmateOf()`.

En Java puedes hacer algo como esto en donde desde una clase interna accedes a un método privado de la clase externa.


## Eliminación de código no deseable

A partir de Java 11 Oracle se ha vuelto más decidido en eliminar características que no eran recomendadas. Habitualmente las despreciaba, pero ahora se ha decidido directamente en eliminarlas. Por ejemplo ha eliminado `Thread.destroy()` y `Thread.stop(Throwable)`. Así que ahora hay que estar más atentos a la compatibilidad con versiones previas al actualizar Java y a los ajustes que debemos implementar en nuestro código.

### 2) Implementaciones del código en Java de algunos nuevos métodos.

Método `isBlank`.

```
/**
 * isBlank() Este nuevo Metodo retorna un valor booleano según si el
 * string contiene solo espacios en blanco y si esta vacío. Si el string
 * esta vacío nos devolverá un true y false de caso contrario.
 *
 * empty si la longitud de la cadena es 0 nos da un true y si no un false si la longitud es mayor a
0
 */
System.out.println(" ".isBlank());
System.out.println("").isEmpty();
String s = "Estudiantes";
System.out.println(s.isBlank());
System.out.println();
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Teniendo como resultado:

```

System.out.println(" ".isBlank());
System.out.println("").isEmpty());
String s = "Estudiantes";
System.out.println(s.isBlank());
System.out.println();

/**
 * est ups.edu.ec.Vista.Principal > main >
 *
 * Output - Run (Java_11) X Search Results
 *
 * --- exec-maven-plugin:1.5.0:exec (default-cli) @ Java_11 ---
 * true
 * true
 * false

```

Método lines().

```

/**
 * lines() Este método devolverá un array de string 'spliteando' el texto por líneas.
 */
String palabra = "Explicacion\nJava 11\n Tarea numero 1";
System.out.println("palabra");
System.out.println(palabra.lines().collect(Collectors.toList()));
System.out.println();

```

Teniendo como resultado

```

5      * lines() Este metodo devolvera un array de string 'spliteando' el texto
6      */
7      System.out.println("Lines");
8      String palabra = "Explicacion\nJava 11\n Tarea numero 1";
9      System.out.println("palabra");
10     System.out.println(palabra.lines().collect(Collectors.toList()));
11     System.out.println();
12
13     /**
14     * est ups.edu.ec.Vista.Principal > main >
15     *
16     * Output - Run (Java_11) X Search Results
17     *
18     * --- exec-maven-plugin:1.5.0:exec (default-cli) @ Java_11 ---
19     * Lines
20     * palabra
21     * [Explicacion, Java 11, Tarea numero 1]

```

Metodo strip.

```

/**
 * strip()
 * stripLeadding(), stripTrailing() Remueve los caracteres en blanco
 * A diferencia del viejo trim(), este nuevo metodo strip si tiene una cuanta de caracteres
Unicode.
 * Unicode: Es un estandar de codificación de caracteres diseñando para facilitar el tratamiento
informático.
 */

```

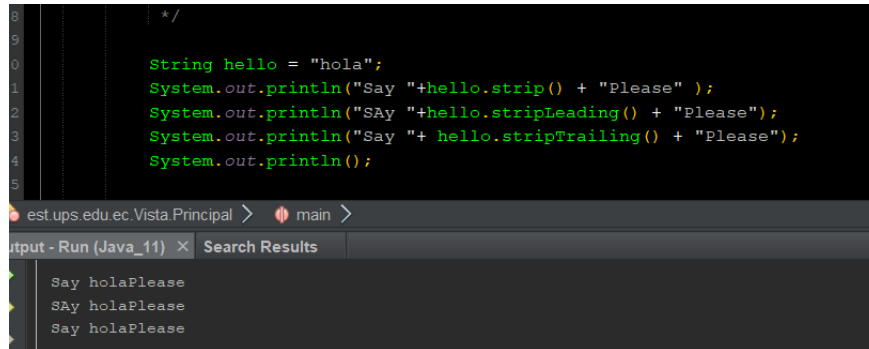
```

String hello = "hola";
System.out.println("Say "+hello.strip() + "Please" );

```

```
System.out.println("SAy "+hello.stripLeading() + "Please");
System.out.println("Say "+ hello.stripTrailing() + "Please");
System.out.println();
```

Teniendo como resultado:



```

8      */
9
10     String hello = "hola";
11     System.out.println("Say "+hello.strip() + "Please" );
12     System.out.println("SAy "+hello.stripLeading() + "Please");
13     System.out.println("Say "+ hello.stripTrailing() + "Please");
14     System.out.println();
15

```

est ups.edu.ec.Vista Principal > main >

Output - Run (Java\_11) X Search Results

```

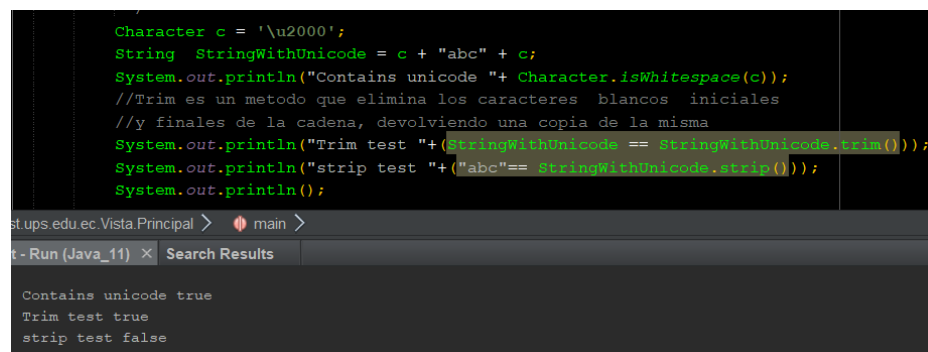
Say holaPlease
SAy holaPlease
Say holaPlease

```

**Método Character.**

```
/**
 * Character.isWhitespace nos permite saber si un caracter es considerado un blanco.
 */
Character c = '\u2000';
String stringWithUnicode = c + "abc" + c;
System.out.println("Contains unicode "+ Character.isWhitespace(c));
//Trim es un metodo que elimina los caracteres blancos iniciales y finales de la cadena,
devolviendo una copia de la misma
System.out.println("Trim test "+(stringWithUnicode == stringWithUnicode.trim()));
System.out.println("strip test "+("abc"== stringWithUnicode.strip()));
System.out.println();
```

Teniendo como resultado:



```

Character c = '\u2000';
String stringWithUnicode = c + "abc" + c;
System.out.println("Contains unicode "+ Character.isWhitespace(c));
//Trim es un metodo que elimina los caracteres blancos iniciales
//y finales de la cadena, devolviendo una copia de la misma
System.out.println("Trim test "+(stringWithUnicode == stringWithUnicode.trim()));
System.out.println("strip test "+("abc"== stringWithUnicode.strip()));
System.out.println();

```

est ups.edu.ec.Vista Principal > main >

Output - Run (Java\_11) X Search Results


```

Contains unicode true
Trim test true
strip test false

```

**Método repeat.**

```
/**
 * Repeat()
 *
 * Repite el string las veces que se le indique.
 */
String rep = "Hola".repeat(2);
System.out.println(rep);
System.out.println();
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Teniendo como resultado:

```
String rep = "Hola".repeat(2);
System.out.println(rep);
System.out.println();
```

est.ups.edu.ec.Vista.Principal > main >

ut - Run (Java\_11) × Search Results

HolaHola

A continuación, se dará a conocer las salidas de todos los métodos antes visto.

Método isBlank

```
true
true
false
```

Método Lines .

```
Lines
palabra
[Explicacion, Java 11, Tarea numero 1]
```

Método strip.

```
Say holaPlease
SAy holaPlease
Say holaPlease
```

Character.

```
Contains unicode true
Trim test true
strip test false
```


Método repeat.

```
HolaHola
```

#### 4.Codificación y salida de nuevos métodos.

##### Codificación.

```
3 public class StringExample {
4
5     public static void main(String[] args) {
6
7         //isBlank() example
8         System.out.println(" ".isBlank());
9         String s = "Hi Word!";
10        System.out.println(s.isBlank());
11        String s1 = "";
12        System.out.println(s1.isBlank());
13        System.out.println();
14
15        // lines() example
16        String str = "Hi\nWord\nHi\nUniverse";
17        System.out.println(str);
18        System.out.println(str.lines().collect(Collectors.toList()));
19        System.out.println();
20
21        // strip() example
22        String hello = " HELLO ";
23        System.out.println("SAY" + hello.strip() + "PLEASE");
24        System.out.println("SAY" + hello.stripLeading() + "PLEASE");
25        System.out.println("SAY" + hello.stripTrailing() + "PLEASE");
26        System.out.println();
27
28        Character c = '\u2000';
29        String stringWithUnicode = c + "abc" + c;
30        System.out.println("Contain unicode " + Character.isWhitespace(c));
31        System.out.println("Trim test " + (stringWithUnicode == stringWithUnicode.trim() ));
32        System.out.println("Strip test " + ("abc"== stringWithUnicode.strip()));
33        System.out.println();
34
35        // repeat() example
36        String rep = "HI".repeat(2);
37        System.out.println(rep);
38        System.out.println();
39    }
40 }
```

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

#### Salida de este código.

```

true
false
true

Hi
Word
Hi
Universe
[Hi, Word, Hi, Universe]

SAYHELLOPLEASE
SAYHELLO PLEASE
SAY HELLOPLEASE

Contain unicode true
Trim test true
Strip test false

HIHI

```

#### RESULTADO(S) OBTENIDO(S):

Realizar una investigación adecuada sobre los cambios importantes de Java 11 entorno a métodos clases y procesos nuevos que se hayan implementado.

Poder desarrollar una codificación adecuada con las nuevas novedades de java 11.

Entender de una manera diferente nuevos procesos de codificación tales como lambda, métodos como trim(), isEmpty entre otros para poder desarrollar una comparación entre lo actual y lo antiguo en java 11.


#### CONCLUSIONES:

El correcto uso de los nuevos aplicativos en java 11 así sean métodos, clases y también la sintaxis para una simplificación de código y la obtención de un mismo resultado. Pero con menos código.

#### RECOMENDACIONES:

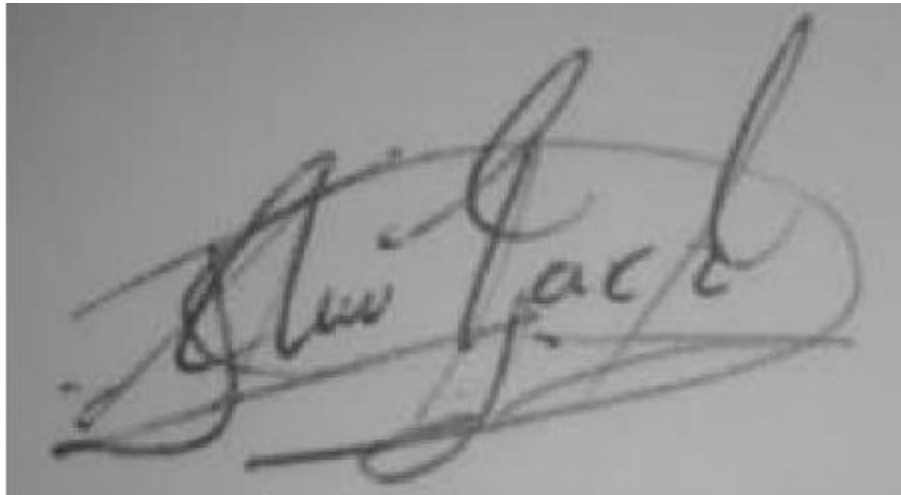
Trabajar mas con los nuevos métodos para estar actualizados con las versiones de java que cada ves se actualiza y es necesario conocer cada uno de sus novedades para poder implementarlos. Además, es necesario desarrollar las actividades planeadas del docente lo antes posible, de esta manera si tiene alguna duda durante algún proceso pueda ser consultado a tiempo.



	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

**Nombre de estudiante:** John Xavier Farez Villa.

**Firma de estudiante:**



**John Xavier Farez Villa.**