
	<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>FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES</b>	
<b>CARRERA:</b> COMPUTACIÓN		<b>ASIGNATURA:</b> Programación Aplicada	
<b>NRO. PRÁCTICA:</b>	1	<b>TÍTULO PRÁCTICA:</b> Investigación de la versión de Java 9	
<b>OBJETIVO ALCANZADO:</b> Identificar los cambios importantes de Java 9. Diseñar e Implementar las nuevas técnicas de programación. Entender la cada uno de las características nuevas en Java 9.			
<b>ACTIVIDADES DESARROLLADAS</b>			
1. Revisar la teoría y conceptos de Java 9.			
2. Diseñar e implementar las características de Java 9.			
3. Probar su funcionamiento y rendimiento dentro de los equipos de cómputo.			
4. Realizar práctica codificando los códigos de las nuevas características de Java.			
<b>RESULTADO(S) OBTENIDO(S):</b> <div style="text-align: center;"><b>JAVA 9</b></div> <p>Java SE 9 fue lanzada el 21 de septiembre del 2017. A esta versión de java se le agregaron varias funciones como:</p> <ul style="list-style-type: none"> <li>• JShell.</li> <li>• Métodos de fabrica Inmutables para List, Set, Map.</li> <li>• Métodos Privados en Interfaces.</li> <li>• Sistema de módulos.</li> <li>• Recolector de basura G1 por defecto</li> <li>• Javadoc</li> <li>• Archivos Jar multi versión</li> </ul> <p><b>JShell Herramienta línea de comandos</b>          La herramienta que fue introducida como jshell, nos permite ejecutar y probar cualquier clase, interfaz, objetos, etc. de una forma muy sencilla.</p>			

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

```

C:\WINDOWS\system32\CMD.exe - jshell

C:\>jshell
| Welcome to JShell -- Version 11.0.2
| For an introduction type: /help intro

jshell> class Calculadora{
...> public static void sumar(int a,int b){
...> int c = a + b;
...> System.out.print("La suma de "+a+" y "+b+" es igual a "+c);
...> }
...> public static void restar(int a,int b){
...> int c = a - b;
...> System.out.print("La resta de "+a+" y "+b+" es igual a "+c);
...> }
...> }
| created class Calculadora

jshell> Calculadora.sumar(4,6)
La suma de 4 y 6 es igual a 10
jshell> Calculadora.restar(4,6)
La resta de 4 y 6 es igual a -2
jshell> _

```

### Métodos de fabrica Inmutables para List, Set, Map.

Se utiliza para crear listas de tipo inmutables, esto significa que no cambiaran su valor una vez instanciadas. Nos ayuda a mejorar la forma de utilizar listas que necesitamos que sean fijas. En Java 8 se creaba primero la lista y luego se creaba otra con la sintaxis de Collections.unmodifiableList(lista). Con Java 9 podemos crear listas vacías o con datos con List.of('Datos o dejar en Blanco').

```

jshell> List numerosInmutablesOff = List.of("uno","dos","tres");
numerosInmutablesOff ==> [uno, dos, tres]

```

### Métodos Privados en Interfaces.

En java 8 se puede hacer la implementación de métodos en interfaces, estos pueden ser predeterminados y estáticos, pero no privados. Con el objetivo de evitar código redundante y de mayor reutilización se introdujo los métodos privados.

```
public interface imprimible {
    /*
    public default void encender() {
        System.out.println("encendiendo el dispositivo");
    }

    public default void apagar() {
        System.out.println("apagando el dispositivo");
    }
    */
    public default void encender() {
        formatear("encendiendo el dispositivo");
    }

    public default void apagar() {
        formatear("apagando el dispositivo");
    }

    private static void formatear( String mensaje) {
        System.out.println("*****");
        System.out.println(mensaje);
        System.out.println("*****");
    }

    public void imprimir();
}
```

### Sistema de módulos

Nos permite llevar mejor la dependencia de paquetes, donde la forma de programación es diferente, se tiene módulos que tienen paquetes que pueden ser dependientes a otros módulos, ayudándonos a mejorar la organización de paquetes dentro de nuestro programa.

### Recolector de basura G1 por defecto


El recolector de basura G1 usa ambas estrategias la paralela y la concurrente. Usa threads concurrentes mientras la aplicación está funcionando buscando los objetos vivos y usa la estrategia paralela para realizar la recolección y compactación rápidamente manteniendo las pausas bajas. El recolector de basura G1 también divide la memoria en regiones de memoria catalogadas como de objetos jóvenes y objetos viejos. Las regiones de objetos jóvenes las recolecta en cada pausa, para las zonas de objetos viejos tiene cierta flexibilidad para recolectar muchas o pocas como la estimación de tiempo que le llevará hacerlo le permita para cumplir con el objetivo de tiempo de pausa configurado dado que permite ajustar según preferencia el límite de tiempo máximo deseado para las pausas.

### Javadoc

Ahora la documentación Javadoc se genera con marcado de HTML 5 e incluye un cuadro de búsqueda para encontrar más fácilmente tipos y métodos.

### Archivos Jar multi versión

Esta función nos ayuda en el apartado de las librerías, donde los desarrollos de las mismas pueden generar un único archivo jar donde pueda tener unas clases para una o varias versiones, por ejemplo, si se tiene las clases A, B, C y D compatibles con Java 6, el desarrollador puede elegir que para la versión nueve la clase A y B sean una optimizadas.

	<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		

#### **CONCLUSIONES:**

Se ha podido observar que existen mejoras en la versión 9 de Java, donde se implementan herramientas que a los programadores nos ayudan para que nuestro código sea mas limpio y organizado. Se podría decir que cada vez se busca llevar una mejor optimización de las funciones e implementar nuevas metodologías de programación.

#### **RECOMENDACIONES:**

Existen temas nuevos, que talvez no podamos desarrollarlo como estudiantes, ya que se vuelve un poco abstracto al no conocer los nuevos términos.

#### **REFERENCIAS:**

Journaldev Java 9 features with examples <https://www.journaldev.com/13121/java-9-features-with-examples>  
 Picodotdev Novedades de Java 9, más allá de la modularidad <https://picodotdev.github.io/blog-bitix/2017/09/novedades-de-java-9-mas-alla-de-la-modularidad/>

**Nombre de estudiante:** Paul Idrovo

**Firma de estudiante:** \_\_\_\_\_