

## TP N°4: Introducción a Java

La siguiente guía cubre los contenidos vistos en las clases teórica **5. Introducción a Java**

### ~~Ejercicio 1~~

Verificar que cuenta con el intérprete y el compilador de Java y que las versiones de ambos son al menos 1.8.

```
$ java -version
$ javac -version
```

### Ejercicio 2

Explicar por qué no compilan los siguientes programas (en papel). Luego verificar en máquina el error obtenido y, de ser posible, corregirlo.

```
public class Ej2_1 {
    public static void main(String args[]) {
        double numero1;
        Double numero2;
        double resultado = numero1 + numero2;
    }
}
```

```
public class Ej2_2 {
    public static void main(String args[]) {
        String a = "hola";
        a = 32;
        System.out.println("El valor de a es: " + a);
    }
}
```

```
public class Ej2_3 {
    public static void main(String args[]) {
        int a;
        int b = a + 5;
    }
}
```

```
public class Ej2_4 {
    public static void main(String args[]) {
        Double a = 1.3;
        Float b = 1.3;
    }
}
```

```
public class Ej2_5 {  
    public static void main(String args[]) {  
        int a = 1;  
        if (a) {  
            System.out.println(",.");  
        }  
    }  
}
```

```
public class Ej2_6 {  
    public static void main(String args[]) {  
        long x = 3;  
        long y = 2;  
        int rta;  
        rta = sumar(x, y);  
        System.out.println(rta);  
    }  
  
    private static int sumar(int a, int b) {  
        return a + b;  
    }  
}
```

```
public class Ej2_7 {  
    public static void main(String args[]) {  
        if (3 > 2) {  
            break;  
        }  
    }  
}
```

### Ejercicio 3

Escribir un programa que tome de la línea de comandos un mensaje y lo imprima en la salida estándar de la siguiente manera: cada palabra deletreada (separada internamente con guiones) y en una nueva línea.

Ejemplo: Al invocar `java Deletrear buen comienzo de Java` debe obtenerse en la salida estándar lo siguiente:

```
b-u-e-n  
c-o-m-i-e-n-z-o  
d-e  
J-a-v-a
```

¿Cómo deberá llamarse la clase? ¿Y el archivo fuente?

### Ejercicio 4

Compilar el siguiente programa. Explicar qué hace y probar que funcione correctamente.

```
public class Ej4 {  
    public static void main(String[] args) {  
        Integer valor1 = Integer.valueOf(args[0]);  
        Integer valor2 = Integer.valueOf(args[1]);  
        Integer suma = valor1 + valor2;  
        System.out.println(suma.toString());  
    }  
}
```

Luego, contestar las siguientes preguntas:

- ¿`valueOf` es un método de instancia o de clase? ¿Para qué sirve?
- ¿`toString` es un método de instancia o de clase? ¿Para qué sirve? ¿Cuál es su equivalente en Ruby?
- ¿Hubiera sido lo mismo invocar a `System.out.println(suma)`? Si es así, ¿por qué no hay un error de compilación al pasarle un `Integer` a un método que recibe `String`?

### Ejercicio 5

Escribir un programa que dada una frase detecte si es palíndromo. Se consideran frases palíndromos aquellas que se pueden leer tanto de izquierda a derecha como de derecha a izquierda, ignorando espacios y diferencias en mayúsculas y minúsculas.

Ejemplo: Al invocar `java Palindromo Anita lava la tina` debe obtenerse en la salida estándar:

```
ES PALINDROMO
```

Ejemplo: Al invocar `java Palindromo Hola Mundo` debe obtenerse en la salida estándar:

```
NO ES PALINDROMO
```

### Ejercicio 6

Indique la salida de los siguientes programas:

```
public class Change {  
    public static void main(String args[]) {  
        System.out.println(2.00 - 1.10);  
    }  
}
```

```
public class LastLaugh {
    public static void main(String args[]) {
        System.out.print("H" + "a");
        System.out.print('H' + 'a');
    }
}
```

```
public class Abc {
    public static void main(String[] args) {
        String letters = "ABC";
        char[] numbers = { '1', '2', '3' };
        System.out.println(letters + " easy as " + numbers);
    }
}
```

### Ejercicio 7

Escribir un programa que reciba por línea de comandos una lista de números enteros en el intervalo [0, 100] e imprima por salida estándar un histograma de los mismos. El histograma debe tener 10 intervalos de clase.

Ejemplo: Al invocar `java Histograma 5 58 2 12 26 81 82 83 45 47 71` una posible salida podría ser:

```

| | | | | | | | | |
| * | | | | | * | | * | |
| * | * | * | | * | * | | * |
+---+---+---+---+---+---+---+---+---+
0   10  20  30  40  50  60  70  80  90  100
```

### Ejercicio 8

Indicar y justificar el resultado que se obtiene al ejecutar el siguiente programa:

```
public class Ej8 {
    public static void main(String[] args) {
        Double valores[] = {1.1, 2.2, 3.3};
        for(Double valor : valores) {
            valor *= 2;
            System.out.printf("El valor es %g\n", valor);
        }
        for(Double valor : valores) {
            valor *= 2;
            System.out.printf("El valor es %g\n", valor);
        }
    }
}
```

**Ejercicio 9**

Representar el estado en que quedaría la memoria al ejecutar el siguiente programa:

```
public class Ej9 {
    public static void main(String[] args) {
        Integer[][] matriz = new Integer[3][4];
        for(int i = 0; i < 3; i++) {
            for(int j = 0; j < 4; j++) {
                matriz[i][j] = 3 - j + i;
            }
        }
    }
}
```

**Ejercicio 10**

Indicar y justificar el resultado que se obtiene al ejecutar el siguiente programa:

```
public class Ej10 {
    public static void main(String[] args) {
        Integer[][] matriz = new Integer[2][2];
        matriz[0][0] = 1;
        matriz[0][1] = 2;
        matriz[1][0] = 3;
        matriz[1][1] = 4;
        for(Integer[] fila : matriz) {
            for(Integer columna : fila) {
                System.out.print(columna);
            }
            fila[1] = fila[0] + 3;
            System.out.println();
        }
        for(Integer[] fila : matriz) {
            for(Integer columna : fila) {
                System.out.print(columna);
            }
            System.out.println();
        }
    }
}
```