

**ULP**  
**Virtual**

# **Tecnicatura Universitaria en Desarrollo de Software**

## **Programación II**

### **Guía 2**

### **Ejercicios**



Universidad de  
**LA PUNTA**



GOBIERNO DE  
**SAN LUIS**

## EJERCICIOS DE APRENDIZAJE

En este módulo vamos a poner en práctica nuestros conocimientos fundamentales de la programación orientada a objetos utilizando algunas clases de API de Java, como String, Scanner y Math; aprenderemos como instancias objetos de algunas de ellas, utilizar sus métodos, invocar métodos estáticos entre otras; por eso es importante que hayas leído con atención la guía y hayas hecho los ejercicios de ejemplos que en ella aparecen. Como el objetivo es aprender a utilizar las nuevas clases del API de Java y repasar con las estructuras de programación vistas en la guía anterior, vamos a crear clases con métodos estáticos.

### Videos



Te sugerimos ver los videos relacionados con este tema, antes de empezar los ejercicios, los podrás encontrar en tu aula virtual o en nuestro canal de YouTube.

Dificultad Baja Dificultad Media Dificultad Alta

*En un nuevo proyecto de nombre “repasoA” crear una clase de nombre GuiaA y dentro de esta, crear un método estático para cada ejercicio propuesto; luego pruébelo desde el método main de una clase PruebaRepasoA.*

1. Escribir un método que pida dos números enteros por teclado y calcule la suma de los dos mostrando por consola la suma de los mismos.
2. Escribir un método que pida tu nombre, lo guarde en una variable y lo muestre por pantalla.
3. Escribir un método que pida una frase y la muestre toda en mayúsculas y después toda en minúsculas. Nota: investigar los métodos toUpperCase() y toLowerCase() de la clase String.
4. Escribir un método que dada una cantidad de grados centígrados se debe mostrar su equivalente en grados Fahrenheit. La fórmula correspondiente es:  $F = 32 + (9 * C / 5)$ .
5. Escribir un método que lea un número entero por teclado y muestre por pantalla el doble, el triple y la raíz cuadrada de ese número. Nota: investigar el método Math.sqrt().

*En un nuevo proyecto de nombre “repasoB” crear una clase de nombre GuiaB y dentro de esta, crear un método estático para cada ejercicio propuesto; luego pruébelo desde el método main de una clase PruebaRepasoB.*

### Condicionales en Java

6. Implementar un método que reciba por parámetro dos números enteros y determine cuál es el mayor y lo muestre por pantalla.
7. Crear un método que reciba por parámetro un numero y determine si es par o impar.
8. Crear un método que pida una frase por teclado y si esa frase es igual a “eureka” pondrá un mensaje de Correcto, sino mostrará un mensaje de Incorrecto. Nota: investigar el método equals() de String.
9. Realizar un método que solo permita introducir frases o palabras de 8 caracteres de largo. Si el usuario ingresa una frase o palabra de 8 de largo se deberá de imprimir un mensaje por pantalla que diga “CORRECTO”, en caso contrario, se deberá imprimir “INCORRECTO”. Nota: investigar el método lenght() de String.
10. Escriba un método que pida una frase o palabra y valide si la primera letra de esa frase es una ‘A’. Si la primera letra es una ‘A’, se deberá de imprimir un mensaje por pantalla que diga “CORRECTO”, en caso contrario, se deberá imprimir “INCORRECTO”.
11. Considera que estás desarrollando una web para una empresa que fabrica motores (suponemos que se trata del tipo de motor de una bomba para mover fluidos). En un nuevo método definir una variable tipoMotor y permitir que el usuario ingrese un valor entre 1 y 4. El método debe mostrar lo siguiente:
  - Si el tipo de motor es 1, mostrar un mensaje indicando “La bomba es una bomba de agua”.
  - Si el tipo de motor es 2, mostrar un mensaje indicando “La bomba es una bomba de gasolina”.
  - Si el tipo de motor es 3, mostrar un mensaje indicando “La bomba es una bomba de hormigón”.
  - Si el tipo de motor es 4, mostrar un mensaje indicando “La bomba es una bomba de pasta alimenticia”.
  - Si no se cumple ninguno de los valores anteriores mostrar el mensaje “No existe un valor válido para tipo de bomba”

*En un nuevo proyecto de nombre “repasoC” crear una clase de nombre GuiaC y dentro de esta, crear un método estático para cada ejercicio propuesto; luego pruébelo desde el método main de una clase PruebaRepasoC.*

### Bucles y sentencias de salto break y continue

12. Escriba un método que valide si una nota ingresada por teclado está entre 0 y 10, sino está entre 0 y 10 la nota se pedirá de nuevo hasta que la nota sea correcta.

13. Escriba un método en el cual se ingrese un valor límite positivo, y a continuación solicite números al usuario hasta que la suma de los números introducidos supere el límite inicial.

14. Realizar un método que pida dos números enteros positivos por teclado y muestre por pantalla el siguiente menú:

MENU
1. Sumar
2. Restar
3. Multiplicar
4. Dividir
5. Salir
Elija opción:

El usuario deberá elegir una opción y el programa deberá mostrar el resultado por pantalla y luego volver al menú. El método deberá ejecutarse hasta que se elija la opción 5. Tener en cuenta que, si el usuario selecciona la opción 5, en vez de salir del programa directamente, se debe mostrar el siguiente mensaje de confirmación: ¿Está seguro que desea salir (S/N)? Si el usuario selecciona el carácter 'S' termina, caso contrario se vuelve a mostrar el menú.

15. Escriba un método que lea 20 números. Si el número leído es igual a cero se debe salir del bucle y mostrar el mensaje "Se capturó el número cero". El método deberá calcular y mostrar el resultado de la suma de los números leídos, pero si el número es negativo no debe sumarse. Nota: recordar el uso de la sentencia break.

16. Escribir un método que simule el funcionamiento de un dispositivo RS232, este tipo de dispositivo lee cadenas enviadas por el usuario. Las cadenas deben llegar con un formato fijo: tienen que ser de un máximo de 5 caracteres de largo, el primer carácter tiene que ser X y el último tiene que ser una O.

Las secuencias leídas que respeten el formato se consideran correctas, la secuencia especial "&&&&&" marca el final de los envíos (llamémosla FDE), y toda secuencia distinta de FDE, que no respete el formato se considera incorrecta. Al finalizar el proceso, se imprime un informe indicando la cantidad de lecturas correctas e incorrectas recibidas. Para resolver el ejercicio deberá investigar cómo se utilizan los siguientes métodos de la clase String: Substring(), Length(), equals().

17. Crear un método que dibuje un cuadrado de N elementos por lado utilizando el carácter “\*”.

Por ejemplo, si el cuadrado tiene 4 elementos por lado se deberá dibujar lo siguiente:

```
* * * *  
*      *  
*      *  
* * * *
```

El valor de N se ingresará por teclado y en este caso utilice la estructura **while** para dibujarlo.

18. Realizar un método que lea 4 números (comprendidos entre 1 y 20) e imprima el número ingresado seguido de tantos asteriscos como indique su valor. Por ejemplo:

```
5 *****  
3 ***  
11 *****  
2 **
```