# FICHA DE TRABAJO PARA LOS EJERCICIOS 2 Y 3 DE LA TAREA

Apellidos y nombre:

Luis Fernández Vidal

#### Instrucciones:

En la primera captura debes incluir también, como fondo, tu login en la plataforma, donde se pueda observar tu nombre y tu foto, para así comprobar que el trabajo lo has realizado tú.

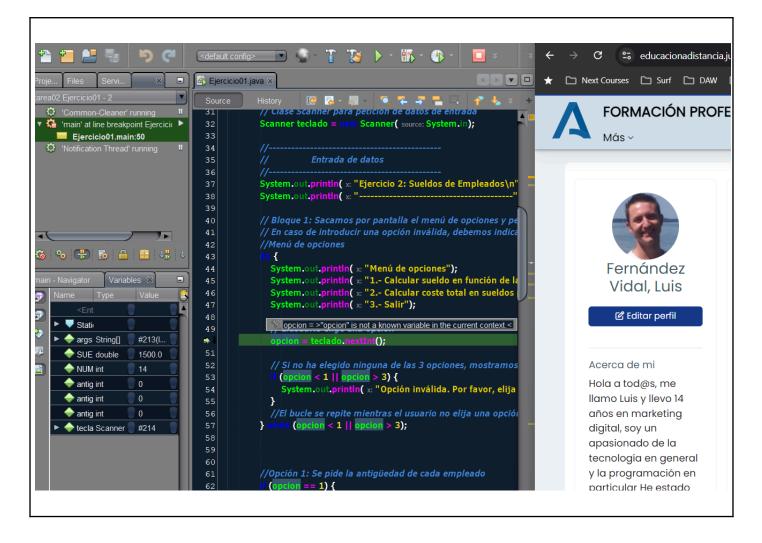
Una vez hayas terminado de cumplimentar el documento, genera un PDF a partir de él para evitar problemas de formato a la hora de corregirlo. ¡Y no olvides incluir ese documento en el paquete junto con el resto de tu proyecto! (En el zip generado exportando tu proyecto de NetBeans y que has nombrado incluyendo tu nombre y apellidos, añade el pdf que has obtenido rellenando el modelo de ficha, como en el documento de ejemplo).

0

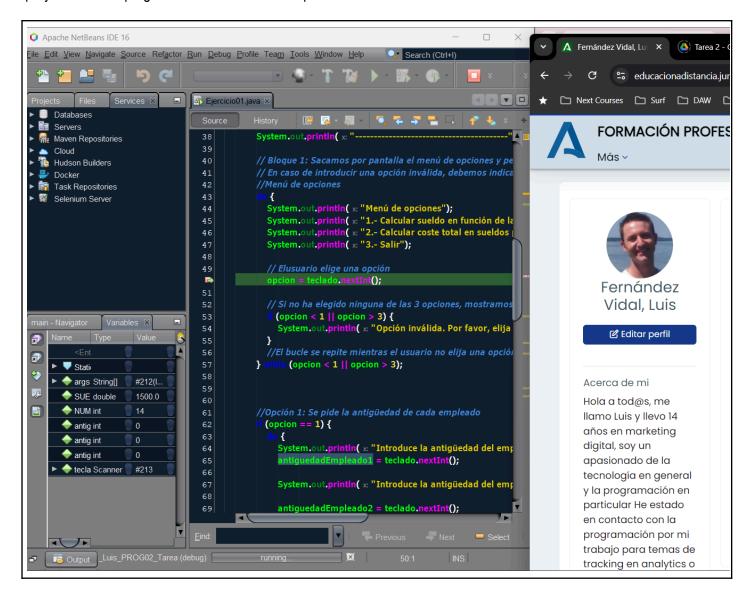
## **EJERCICIO 2**

1.- Establecer un punto de ruptura en la línea donde se lea la opción de menú elegida por el usuario.

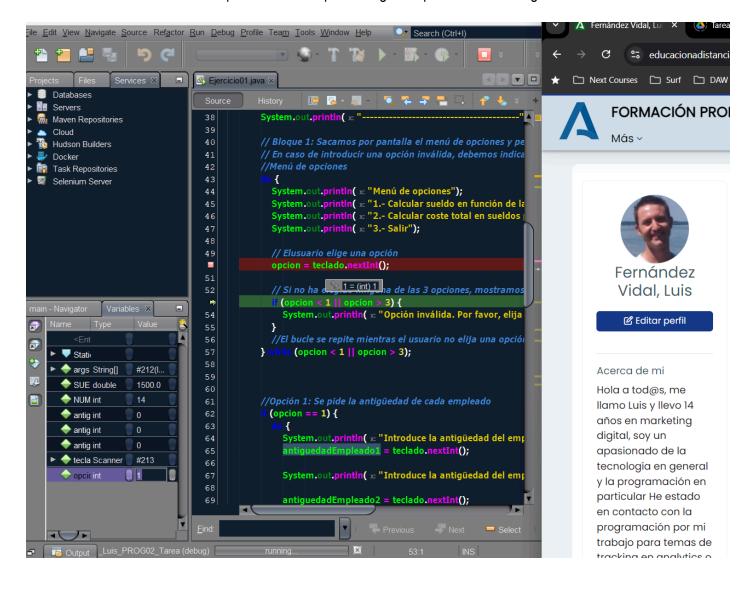
Muestra una captura de pantalla del breakpoint junto con tu perfil en la plataforma.



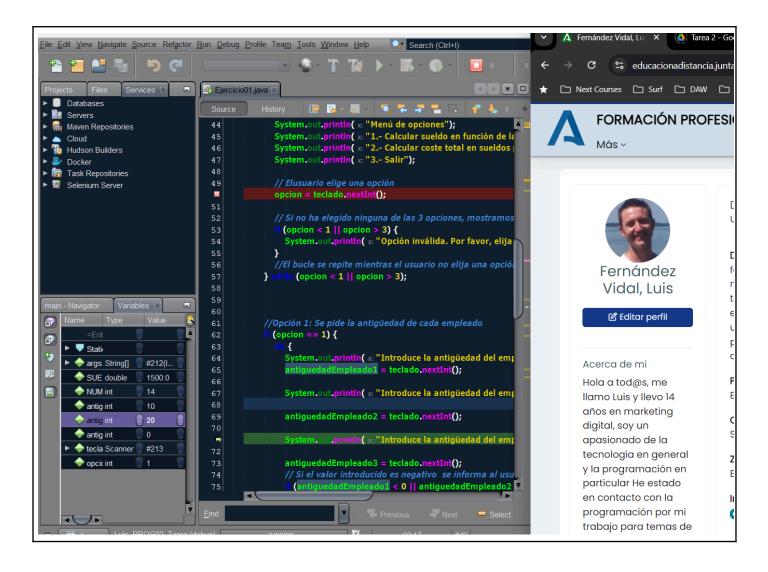
2.- Ejecuta en el programa modo depuración. Muestra la ventana de variables y sitúala a la izquierda debajo de los proyectos con el programa detenido en el breakpoint creado anteriormente.



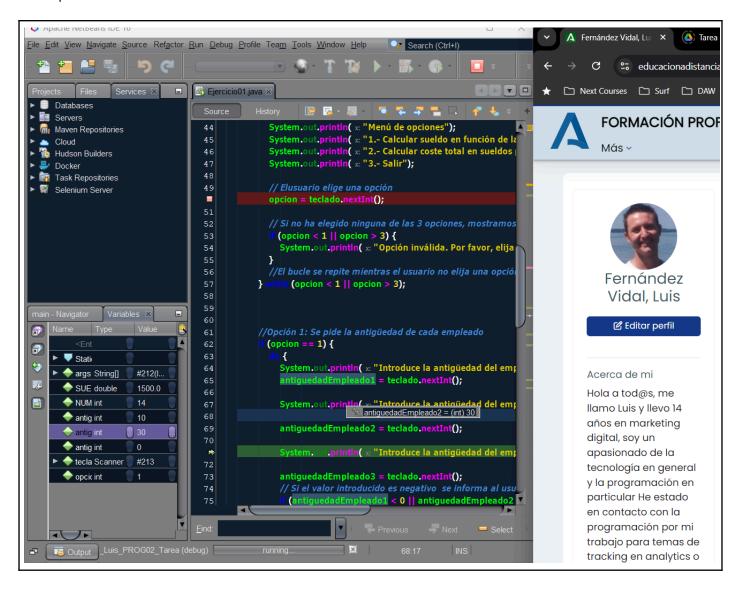
3.- Muestra el valor de la variable que has utilizado para recoger la opción de menú elegida.



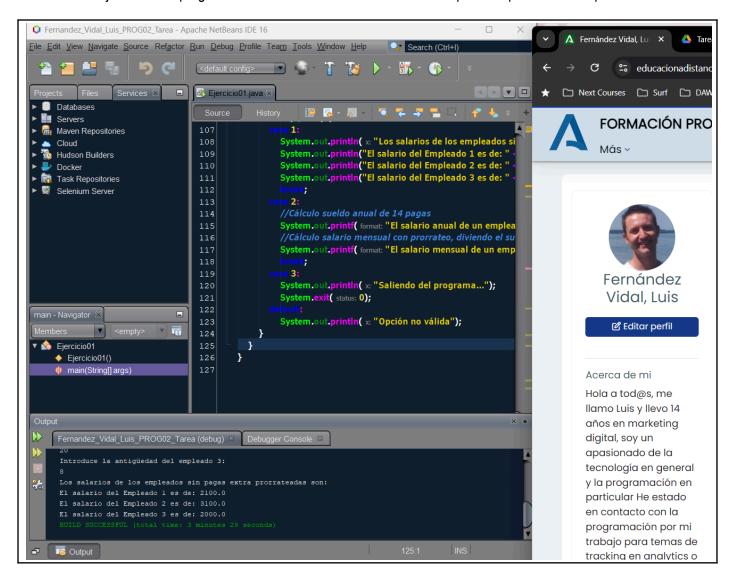
4.- Avanza paso a paso por la ejecución del programa hasta que éste solicite la antigüedad del "Empleado 2", introduce como valor para este dato 20. Muestra el valor de la variable colocando el cursor sobre la variable mediante la captura correspondiente.



5.- Utiliza el inspector de variables para cambiar el valor del dato de la antigüedad de 20 a 30. Muestra la captura correspondiente.



6.- Finaliza la ejecución del programa mostrando el resultado final con la captura de pantalla correspondiente.



#### **EJERCICIO 3**

1. Dados los siguientes bloques de código, indica cuál de ellos corresponde a una excepción y cual corresponde a una aserción, explicando el por qué de tu respuesta.

# Bloque de código 1:

```
int numerador = 10;
int denominador = 0;

try {
    int resultado = numerador / denominador;
    System.out.println("El resultado es: " + resultado);
} catch (ArithmeticException e) {
    System.out.println("Error: No se puede dividir por cero.");
}
```

Esta captura corresponde a la captura de una excepción, en este caso catch captura una excepcion aritmética cuando a int denominador se le da un valor de 0 e informa "Error: No se puede dividir por cero"

### Bloque de código 2:

```
int edad = -5;
assert edad > 0 : "La edad debe ser mayor que 0";
System.out.println("La edad es: " + edad);
```

## Respuesta:

En esta aserción se comprueba que la edad introducida no es negativa que será registrada en consola, en este caso no habrá mensaje para el desarrollador y se la ejecución.

2. ¿Para qué se usan las excepción y aserciones en java?. Justifica tu respuesta de acuerdo a lo explicado en las videoconferencias.

Las excepciones al contrario que los errores de compilación son errores que se dan durante la ejecución, por ejemplo cuando el usuario asigna valores a una de nuestras que no pueden ser gestionados, por ejemplo da una string se le pide que asigne un valor numérico. Esto crea excepciones que se capturan mediante el catch y se muestra el error/excepción

Las aserciones se utilizan para detectar errores en la fase de desarrollo, se crea una serie de aserciones dentro del proyecto para que el desarrollador habilita y crea para depurar el programa, y que una vez que vaya a salir a producción se eliminan.