UNIVERSIDAD TECNOLÓGICA DE PANAMÁ

FACULTAD DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

LICENCIATURA EN CIBERSEGURIDAD

PROGRAMACION 1

LABORATORIO 2

PREPARADO POR

ADRIAN JIMENEZ 4-839-2413

JUSTIN HE 8-1045-2230

A CONSIDERACIÓN DE:

NAPOLEON IBARRA

GRUPO: 2S3111

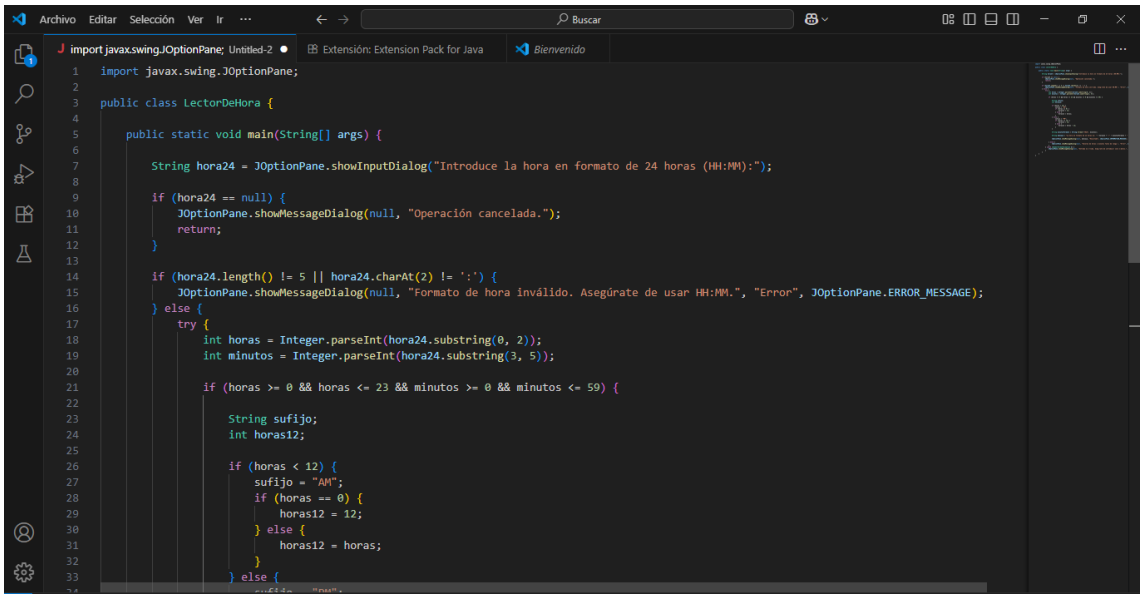
FECHA: 8-9-25

Programa 1: Leer la hora asignada e imprimirla

Objetivo

Leer la hora que un usuario le asigna al programa desarrollado en notación de 24 horas, para luego imprimirla en 12.

Código Desarrollado



Explicación del código

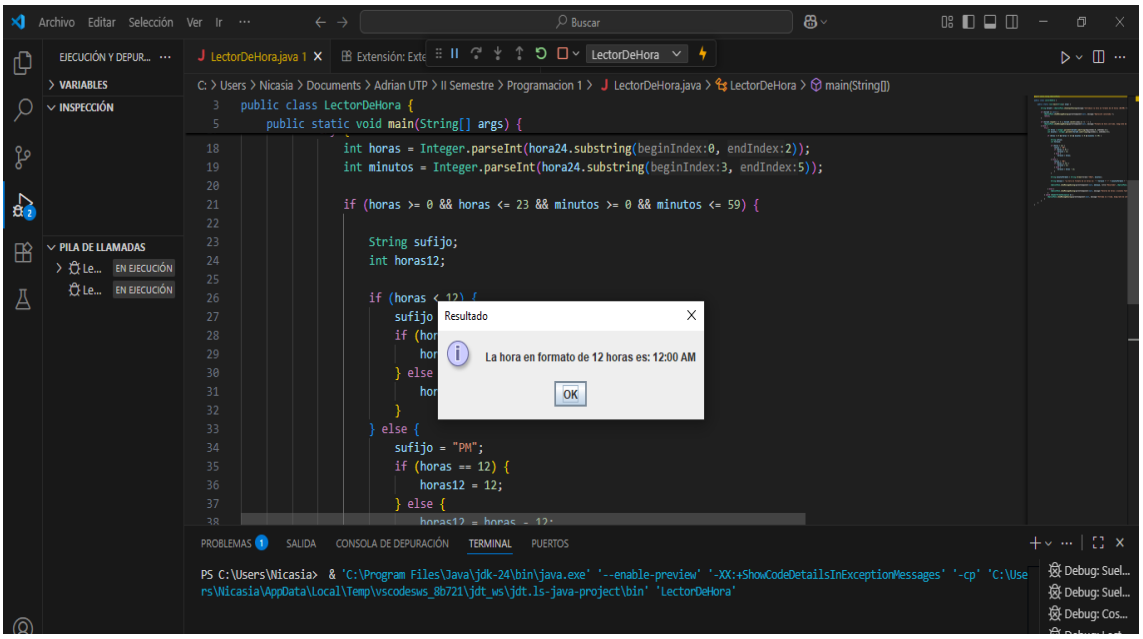
Primeramente se coloca el comando **import javax.swing.JOptionPane** porque es fundamental para se pueda abrir las ventanas a la hora de ejecutar el código.

Se utiliza el comando **JOptionPane.showInputDialog** para pedir la hora al usuario en una ventana. También **JOptionPane.showMessageDialog** para mostrar los mensajes de error en otra ventana si se escribe mal la hora.

También se ingresa el comando **Integer.parseInt** para convertir las horas y los minutos que el usuario escribió a números, para poder hacer los cálculos y **.substring** para cortar el texto de la hora para separar las horas de los minutos.

Y aparte también se utiliza el comando **.length** para verificar que la hora que se ingresó tenga 5 caracteres.

Ejecución del código



Programa 2: Costo diario de conducir y ahorro al compartir

Objetivo

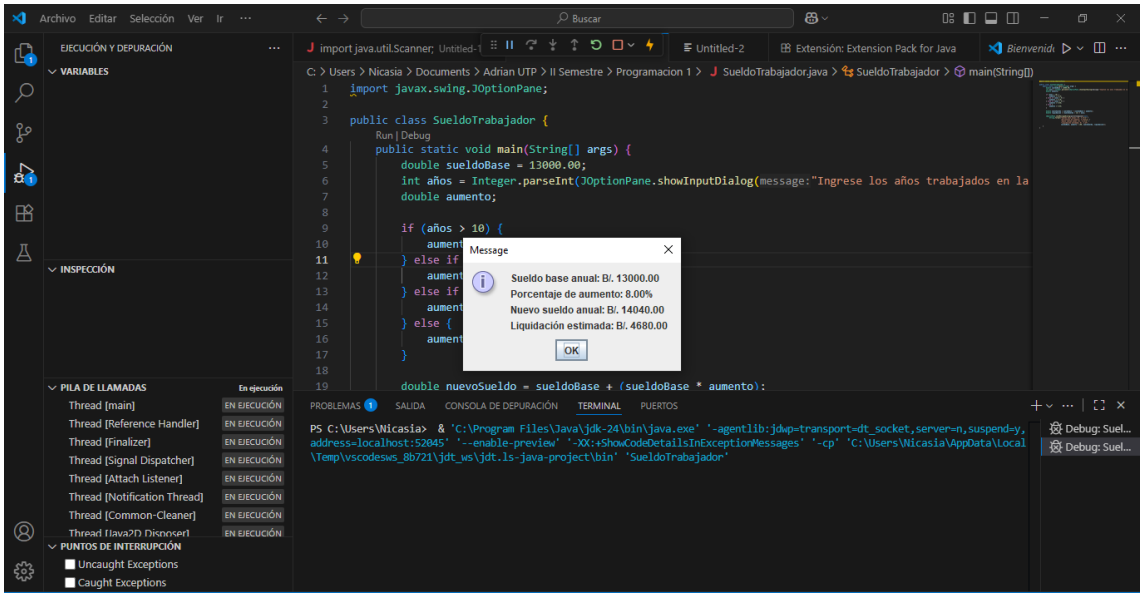
Calcular el costo total diario de usar el automóvil para ir al trabajo y estimar el ahorro por persona si se comparte el viaje. Se consideran combustible, estacionamiento y peajes.

Código fuente: CostoAuto.java

Explicación del código

- Se piden los datos al usuario con JOptionPane.showInputDialog(...).
- Se convierten a double/int con Double.parseDouble(...) y Integer.parseInt(...).
- Se calculan litrosConsumidos, costoCombustible y costoTotal.
- Si hay más de 1 persona, se divide el costoTotal entre personas para obtener costoCompartido.
- Se calcula el ahorro como la diferencia entre manejar solo y el costo por persona compartiendo.
- Se muestran los resultados con 2 decimales usando String.format("%.2f").

Ejemplo de prueba rápida

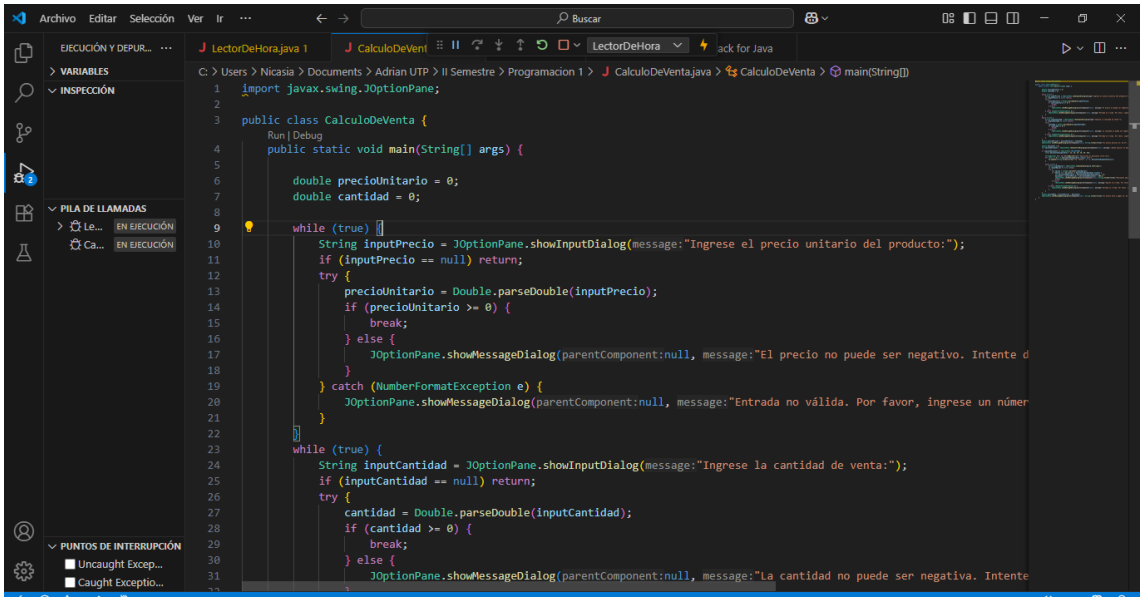


Programa 3: Calcular precio de un producto

Objetivo

Calcular el precio de un producto, y agregar un descuento solo si el vendedor lo requiera.

Código



```java
import javax.swing.JOptionPane;

public class CalculoDeVenta {
    Run | Debug
    public static void main(String[] args) {

        double precioUnitario = 0;
        double cantidad = 0;

        while (true) {
            String inputPrecio = JOptionPane.showInputDialog(message:"Ingrese el precio unitario del producto:");
            if (inputPrecio == null) return;
            try {
                precioUnitario = Double.parseDouble(inputPrecio);
                if (precioUnitario >= 0) {
                    break;
                } else {
                    JOptionPane.showMessageDialog(parentComponent:null, message:"El precio no puede ser negativo. Intente d
                }
            } catch (NumberFormatException e) {
                JOptionPane.showMessageDialog(parentComponent:null, message:"Entrada no válida. Por favor, ingrese un númer
            }
        }
        while (true) {
            String inputCantidad = JOptionPane.showInputDialog(message:"Ingrese la cantidad de venta:");
            if (inputCantidad == null) return;
            try {
                cantidad = Double.parseDouble(inputCantidad);
                if (cantidad >= 0) {
                    break;
                } else {
                    JOptionPane.showMessageDialog(parentComponent:null, message:"La cantidad no puede ser negativa. Intente
```

Explicación

Usamos **JOptionPane.showInputDialog** para pedir el precio y la cantidad en una ventana emergente. Este método siempre devuelve un **String o null** si el usuario cancela.

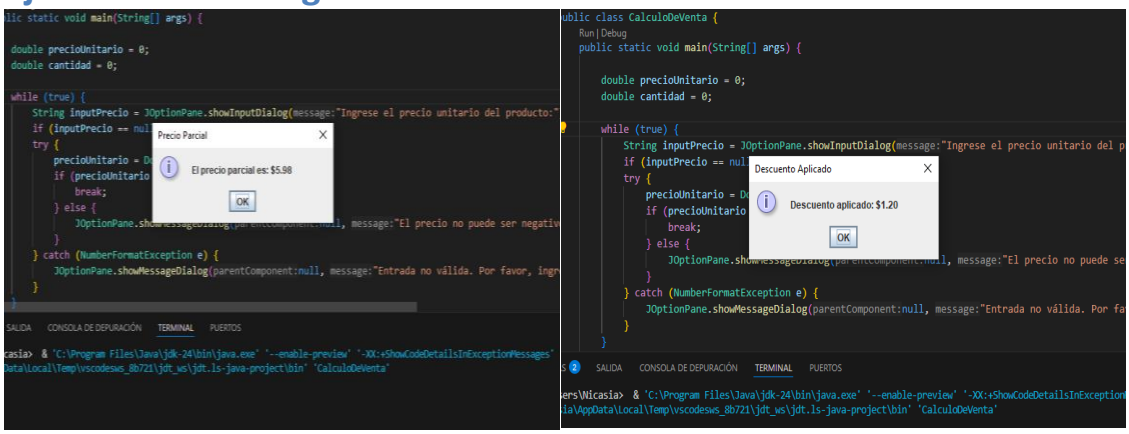
Para mostrar resultados o errores, usamos **JOptionPane.showMessageDialog**. Esto presenta la información en una ventana emergente.

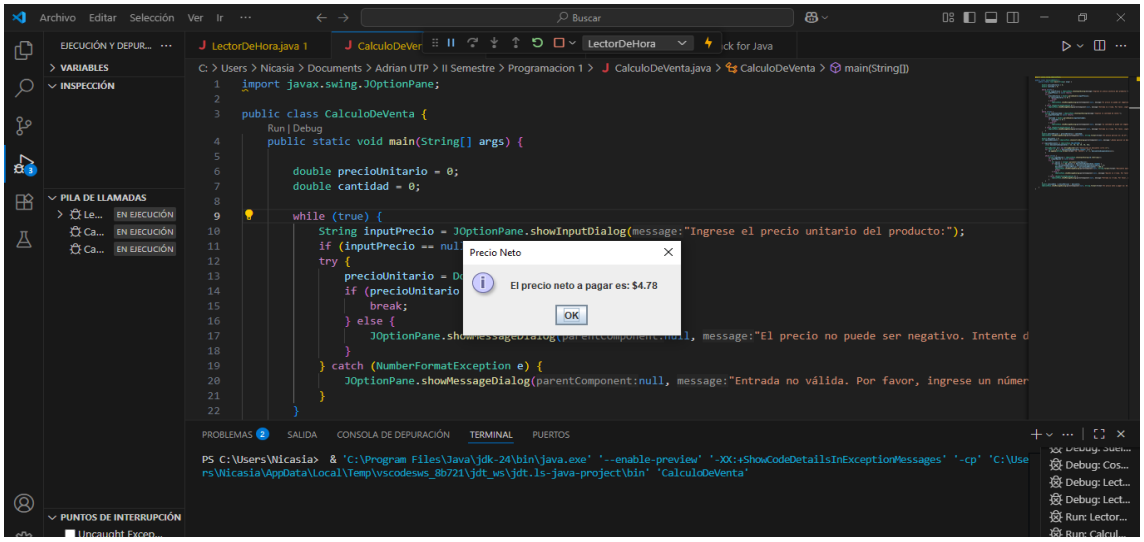
Para preguntar si el usuario quiere un descuento, se utiliza **JOptionPane.showConfirmDialog**, que muestra una ventana con las opciones "Sí" y "No".

Se usa **StringBuilder** para construir el texto del menú de descuentos.

Se mantienen los bucles **while** y los bloques **try-catch** para asegurar que las entradas de precio, cantidad y la opción de descuento sean números válidos.

Ejecución del código





Programa 4: Sueldo con aumento y liquidación

Objetivo

Calcular el nuevo sueldo anual de un trabajador que tiene un sueldo base de B/. 13,000 anuales, aplicando un aumento según la antigüedad y estimar una liquidación simple.

Criterios de aumento

- Más de 10 años: +15%
- Más de 5 y hasta 10 años: +10%
- Más de 3 y hasta 5 años: +8%
- 3 años o menos: +5%

Suposición para la liquidación (simplificada)

Se asume 1 sueldo mensual por cada año trabajado: liquidación = (nuevoSueldo / 12) × años. Esta fórmula es didáctica y simplificada para el ejercicio.

Código fuente: SueldoTrabajador.java

```java
import javax.swing.JOptionPane;

public class SueldoTrabajador {
    public static void main(String[] args) {
        double sueldoBase = 13000.00;
        int años = Integer.parseInt(JOptionPane.showInputDialog("Ingrese los años trabajados en la empresa:"));
        double aumento;

        if (años > 10) {
            aumento = 0.15;
        } else if (años > 5) {
            aumento = 0.10;
        } else if (años > 3) {
            aumento = 0.08;
        } else {
            aumento = 0.05;
        }

        double nuevoSueldo = sueldoBase + (sueldoBase * aumento);
        double liquidacion = (nuevoSueldo / 12) * años;

        JOptionPane.showMessageDialog(null,
            String.format("Sueldo base anual: B/. %.2f\n" +
                          "Porcentaje de aumento: %.2f%%\n" +
                          "Nuevo sueldo anual: B/. %.2f\n" +
                          "Liquidación estimada: B/. %.2f",
                          sueldoBase, aumento * 100, nuevoSueldo, liquidacion));
    }
}
```

Explicación pasó a paso

- Se solicita la antigüedad (años) vía JOptionPane.
- Según el valor de años, se asigna el porcentaje de aumento con if / else-if / else.
- Se calcula el nuevo sueldo anual: nuevoSueldo = sueldoBase + (sueldoBase × aumento).
- Se estima la liquidación: (nuevoSueldo / 12) × años.
- Se muestran los resultados formateados a 2 decimales.

Ejemplo de prueba rápida