



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ FACULTAD DE INGENIERÍA DE
SISTEMAS COMPUTACIONALES LICENCIATURA EN CIBERSEGURIDAD

Programación I

Laboratorio #2

PREPARADO POR:

Angeline Urriola 4-834-1980

Daniela Sánchez 4-836-173

PROFESOR:

Napoleón Ibarra

GRUPO:

2S3111

FECHA:

7/09/2025

Visual Studio Code

Usamos Visual Studio Code para escribir y ejecutar los códigos JavaScript que se nos asignaron. Visual Studio Code es un editor de código fuente muy popular y ligero, que ofrece herramientas integradas como resaltado de sintaxis, autocompletado y depuración, lo que facilita la creación y el manejo eficiente de programas en JavaScript. Además, su gran comunidad y la posibilidad de agregar extensiones hacen que sea muy adaptable a nuestras necesidades diarias de trabajo.

NetBeans

También utilizamos NetBeans para desarrollar nuestros códigos. NetBeans es un entorno de desarrollo integrado (IDE) completo que soporta múltiples lenguajes de programación, incluido JavaScript, y ofrece funcionalidades como gestión de proyectos, depuración y edición avanzada, lo que ayuda a mejorar la productividad al programar. Nos resulta especialmente útil para manejar proyectos más grandes y complejos, ya que organiza todo de forma clara y nos permite trabajar de manera más ordenada.

JOptionPane

Nos pidieron que utilizáramos JOptionPane para la entrada y salida de datos en nuestros programas. JOptionPane es una clase de la biblioteca Swing de Java que permite crear ventanas emergentes o diálogos gráficos, los cuales facilitan la interacción con el usuario de forma visual y sencilla. A través de estas ventanas, podemos solicitar datos al usuario mediante cuadros de texto o mostrar información y mensajes de forma clara. Esto hace que la comunicación entre el programa y el usuario sea más intuitiva, en lugar de usar solo la consola de texto, haciendo los programas más amigables y fáciles de usar. Además, aprender a manejar JOptionPane nos ayuda a familiarizarnos con el desarrollo de interfaces gráficas básicas en Java.

Problemas

Desarrollo de problemas en Java. Valor 70 Puntos

1. Escriba un código que lea la hora en notación de 24 horas y que imprima en notación de 12: por ejemplo, si la entrada es 13:45, la salida será 1:45 PM. El programa debe solicitar al usuario final que introduzca cinco caracteres para especificar una hora: por ejemplo, las 9 en punto se debe introducir así: 09:00. Valor 15 puntos

Para este problema se utilizó Visual Code, a continuación, el código:

```
..> Users > angel > OneDrive > Desktop > UTP ANGE > Programación 1 > LAB 2 > hora.java > hora > main(String[])
1 import javax.swing.JOptionPane;
2 public class hora {
3     Run | Debug
4     public static void main(String[] args)
5     {
6         try
7         {
8             String horai = JOptionPane.showInputDialog(message:"Hora en formato de 24h:");
9
10            if (horai == null) {
11                return;
12            }
13
14            if (horai.length() != 5) {
15                JOptionPane.showMessageDialog(parentComponent:null, message:"Error");
16                return;
17            }
18
19            if (horai.charAt(index:2) != ':') [
20                JOptionPane.showMessageDialog(parentComponent:null, message:"Error");
21            ]
22
23            int hrs = Integer.parseInt(horai.substring(beginIndex:0, endIndex:2));
24            int min = Integer.parseInt(horai.substring(beginIndex:3, endIndex:5));
25
26            if (hrs < 0) {
27                JOptionPane.showMessageDialog(parentComponent:null, message:"Error");
28                return;
29            }
30
31            if (hrs > 23) {
32                JOptionPane.showMessageDialog(parentComponent:null, message:"Error");
33                return;
34            }
35
36            if (min < 0) {
```

```

37     JOptionPane.showMessageDialog(parentComponent:null, message:"Error");
38     return;
39   }
40
41   if (min > 59) {
42     JOptionPane.showMessageDialog(parentComponent:null, message:"Error");
43     return;
44   }
45
46   String jornada;
47   if (hrs >= 12) {
48     jornada = "pm";
49   } else {
50     jornada = "am";
51   }
52
53   int horad = hrs % 12;
54   if (horad == 0) {
55     horad = 12;
56   }
57
58   String horaid = String.format(format:"%02d", horad) + ":" + String.format(format:"%02d", min) + " " + jornada;
59   JOptionPane.showMessageDialog(parentComponent:null, "Hora en formato 12h: " + horaid);
60   catch (Exception e) {
61     JOptionPane.showMessageDialog(parentComponent:null, message:"Error");
62   }
63 }
64 }
```

Podemos ver que el código funciona correctamente porque a la hora de ejecutarlo sale de esta manera:



2. Crear un código que calcule su costo diario al conducir su automóvil, de modo que pueda estimar cuánto dinero puede ahorrar si comparte los viajes en automóvil, lo cual también tiene otras ventajas, como la reducción de emisiones de carbono y mostrar el costo de la congestión de tráfico. La aplicación debe recibir como entrada la siguiente información y mostrar el costo por día para el usuario por conducir al trabajo. Valor 15 puntos

- a. Total de kilómetros conducido por día.
- b. Costo por litro de combustible.
- c. Promedio de kilometro por litros.
- d. Cuota de estacionamiento por día.

e. Peaje por día.

f. ¿Cuánto es el ahorro?

Código del problema 2:

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Projects View:** Shows the project structure with a package named `com.miran.costodiaricoconducion` containing a source file `CostoDiarioConduccion.java`.
- Source Editor:** Displays the Java code for calculating daily toll costs. It includes imports for `JOptionPane` and `DecimalFormat`, and a main method that prompts for kilometers driven, cost per liter, average kilometers per liter, and parking fees, then calculates total cost.
- Output View:** Shows the command line output for running the application.

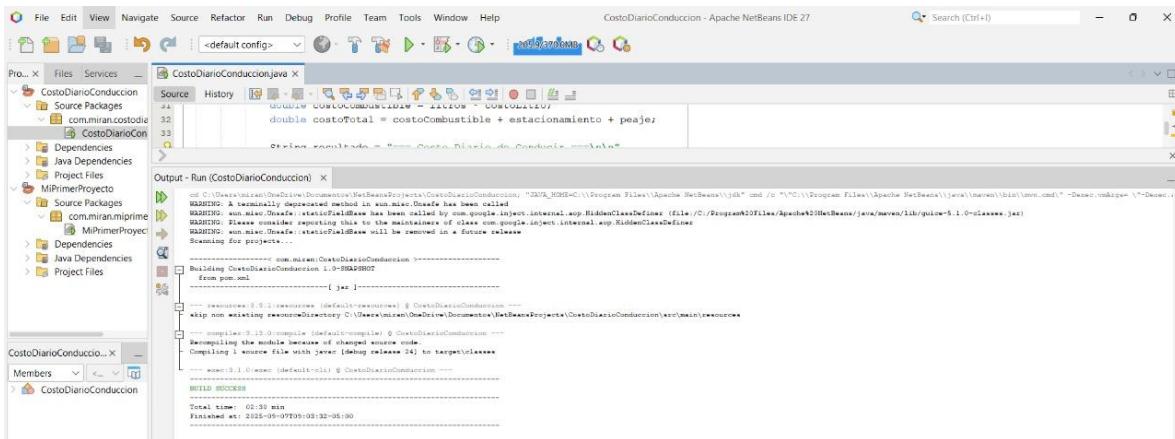
```
CostoDiarioConduccion - Apache NetBeans IDE 27
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
CostoDiarioConduccion - Apache NetBeans IDE 27
Search (Ctrl+F)
Projects X Files Services
CostoDiarioConduccion
  Source Packages
    com.miran.costodiaricoconducion
      CostoDiarioConduccion.java
  Dependencies
  Java Dependencies
  Project Files
MiPrimerProyecto
  Source Packages
    com.miran.miprimerproyecto
      MiPrimerProyecto.java
  Dependencies
  Java Dependencies
  Project Files
CostoDiarioConducción.java - Navigator X
Members <empty>
CostoDiarioConducción.java - Source X
Source History
1 /**
2  * Click nbfs://nbhost/SystemFileSystem/Templates/licenses/license-default.txt to change this license.
3  *
4  */
5 package com.miran.costodiaricoconducion;
6
7 /**
8  * 
9  * @author Daniela
10 */
11 import javax.swing.JOptionPane;
12 import java.text.DecimalFormat;
13
14 public class CostoDiarioConduccion {
15     public static void main(String[] args) {
16         DecimalFormat df = new DecimalFormat("0.00");
17
18         try {
19             double kmDia = Double.parseDouble(JOptionPane.showInputDialog("Kilómetros conducidos por dia:"));
20             double costoLitro = Double.parseDouble(JOptionPane.showInputDialog("Costo por litro de combustible (B/.):"));
21             double kmPorLitro = Double.parseDouble(JOptionPane.showInputDialog("Promedio de Kilómetros por litro:"));
22             double estacionamiento = Double.parseDouble(JOptionPane.showInputDialog("Cuota de estacionamiento por dia (B/.):"));
23             double peaje = Double.parseDouble(JOptionPane.showInputDialog("Peaje por dia (B/.):"));
24
25             if (kmPorLitro <= 0) {
26                 JOptionPane.showMessageDialog(null, "El rendimiento debe ser mayor que 0.", "Error", JOptionPane.ERROR_MESSAGE);
27                 return;
28             }
29
30             double litros = kmDia / kmPorLitro;
31             double costoCombustible = litros * costoLitro;
32             double costoTotal = costoCombustible + estacionamiento + peaje;
33
34         } catch (NumberFormatException e) {
35             JOptionPane.showMessageDialog(null, "Debe ingresar solo números.", "Error", JOptionPane.ERROR_MESSAGE);
36         }
37
38         double ahorro = Double.parseDouble(JOptionPane.showInputDialog("¿Cuanto podria ahorrar compartiendo el viaje? (B/.):"));
39         JOptionPane.showMessageDialog(null, "Ahorro estimado: B/. " + df.format(ahorro));
40
41     }
42
43 }
44
45
46
47
48
49
50
51
52
53
54 }
```

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Projects View:** Shows the project structure with the same package and source file as the previous screenshot.
- Source Editor:** Displays the completed Java code. It adds a string variable `resultado` to store the output message, which is then displayed using `JOptionPane.showMessageDialog`. It also handles the case where input is not a number.
- Output View:** Shows the command line output for running the application, including the calculated total cost and estimated savings.

```
CostoDiarioConducción.java - Source X
Source History
31         double costoTotal = costoCombustible + estacionamiento + peaje;
32
33         String resultado = "==== Costo Diario de Conducir ====\n";
34         resultado += "Kilómetros por dia: " + kmDia + " km\n";
35         resultado += "Costo por litro: B/. " + df.format(costoLitro) + "\n";
36         resultado += "Rendimiento: " + kmPorLitro + " km/l\n";
37         resultado += "Combustible: B/. " + df.format(costoCombustible) + "\n";
38         resultado += "Estacionamiento: B/. " + df.format(estacionamiento) + "\n";
39         resultado += "Peaje: B/. " + df.format(peaje) + "\n";
40         resultado += "Costo total diario: B/. " + df.format(costoTotal);
41
42         JOptionPane.showMessageDialog(null, resultado);
43
44         double ahorro = Double.parseDouble(JOptionPane.showInputDialog("¿Cuanto podria ahorrar compartiendo el viaje? (B/.):"));
45         JOptionPane.showMessageDialog(null, "Ahorro estimado: B/. " + df.format(ahorro));
46
47     } catch (NumberFormatException e) {
48         JOptionPane.showMessageDialog(null, "Debe ingresar solo números.", "Error", JOptionPane.ERROR_MESSAGE);
49     }
50
51
52
53
54 }
```

Código ejecutado



3. Escribir un código que permita ingresar el precio unitario de un producto vendido y la cantidad de venta; después calcular el precio parcial (precio unitario * cantidad), mostrar un descuento del 5%, 10%, 20%, 30%, 40%, 50%, sólo si el vendedor ha seleccionado la opción de aplicar descuento, al final mostrar el precio neto (precio parcial - descuento). Valor 20 Puntos

Código del tercer problema:

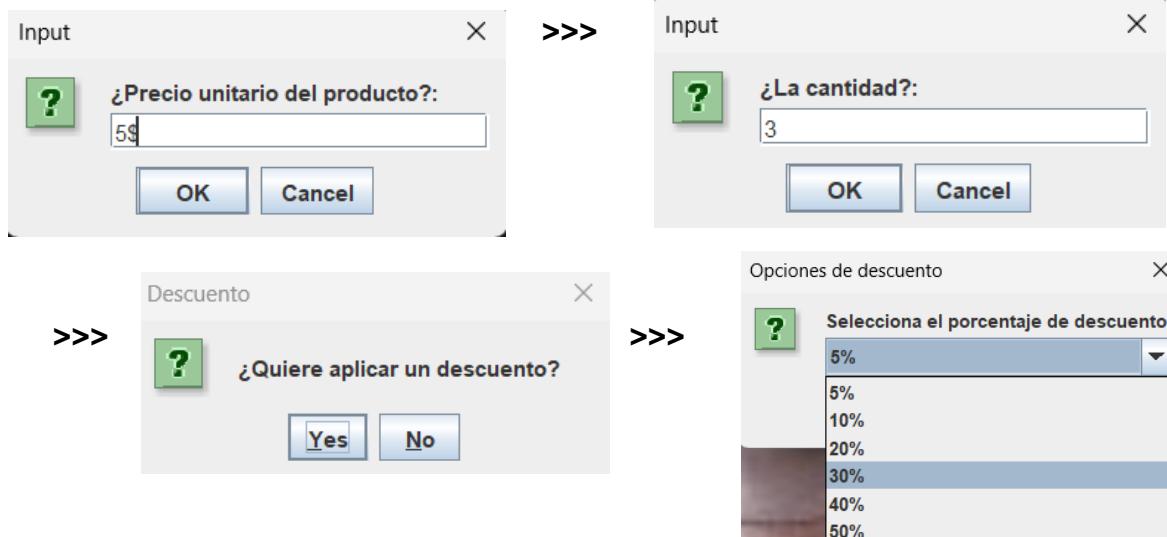
```
1 import javax.swing.JOptionPane;
2 import java.text.DecimalFormat;
3
4 public class pn {
5     Run | Debug
6     public static void main(String[] args) {
7         try {
8             DecimalFormat df = new DecimalFormat(pattern:"#,##0.00");
9
9             String entradaPrecio = JOptionPane.showInputDialog(message:"¿Precio unitario del producto?:");
10
11             if (entradaPrecio == null) return;
12
13             entradaPrecio = entradaPrecio.replace(target:"", replacement:"").trim();
14
15             double precioUni = Double.parseDouble(entradaPrecio);
16
17             int cant = Integer.parseInt(JOptionPane.showInputDialog(message:"¿La cantidad?:"));
18
19             double precioPa = precioUni * cant;
20
21             int desapl = JOptionPane.showConfirmDialog(parentComponent:null, message:"¿Quiere aplicar un descuento?", title:"Descuento",
22                 JOptionPane.YES_NO_OPTION);
23
24
25             double des = 0;
26
27             if (desapl == JOptionPane.YES_OPTION) {
28                 String[] opcion = {"5%", "10%", "20%", "30%", "40%", "50%"};
29                 String seleccion = (String) JOptionPane.showInputDialog(parentComponent:null, message:"Selecciona el porcentaje de descuento:",
30                     title:"Opciones de descuento", JOptionPane.QUESTION_MESSAGE, icon:null, opcion[0]);
31
32                 if (seleccion != null) {
33                     int porcentaje = Integer.parseInt(seleccion.replace(target:"%", replacement:""));
34                     des = precioPa * porcentaje / 100.0;
35                 }
36             }
37         }
38     }
39 }
```

```

37     double precioNet = precioPa - des;
38
39     String mensaje = "Precio unitario: $" + df.format(precioUni) + "\nCantidad: " + cant + "\nPrecio parcial: $" + df.format(precioPa) +
40             "\nDescuento aplicado: $" + df.format(des) + "\nPrecio neto: $" + df.format(precioNet);
41     JOptionPane.showMessageDialog(parentComponent:null, mensaje);
42
43 } catch (Exception e) {
44     JOptionPane.showMessageDialog(parentComponent:null, message:"Error");
45 }
46
47 }

```

Ejecución de código:



Resultado final:



4. Escriba un código que calcule el sueldo que le corresponde al trabajador de una empresa que cobra B/. 13.000 anuales, el programa debe realizar los cálculos en función de los siguientes criterios. Valor 20 puntos

- Si lleva más de 10 años en la empresa se le aplica un aumento del 15%.
- Si lleva menos de 10 años pero más que 5 se le aplica un aumento del 10%.
- Si lleva menos de 5 años pero más que 3 se le aplica un aumento del 8%.

d. Si lleva menos de 3 años se le aplica un aumento del 5%.

e. La liquidación de sus prestaciones en caso de renuncia o liquidación.

Código:

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Project:** CostoDiarioConduccion
- Source Packages:** com.miran.costodiarioconduccion, CalculoSueloTrabajador.java, CostoDiarioConduccion.java
- Code Editor:** CalculoSueloTrabajador.java

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
 */
package com.miran.costodiarioconduccion;

/*
 * @author Danirs
 */
import javax.swing.JOptionPane;
import java.text.DecimalFormat;

public class CalculoSueloTrabajador {
    public static void main(String[] args) {
        DecimalFormat df = new DecimalFormat("0.00");

        try {
            final double SALARIO_ANUAL = 13000.0;
            double aumento;

            int years = Integer.parseInt(JOptionPane.showInputDialog("Años de servicio del trabajador:"));

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

- Output:** Run (CostoDiarioConduccion)

The screenshot shows the Apache NetBeans IDE interface with the following details:

- Project:** CostoDiarioConduccion
- Source Packages:** com.miran.costodiarioconduccion, CalculoSueloTrabajador.java, CostoDiarioConduccion.java
- Code Editor:** CalculoSueloTrabajador.java

```
        } else {
            aumento = 0.05;
        }

        double montoAumento = SALARIO_ANUAL * aumento;
        double salarioNuevo = SALARIO_ANUAL + montoAumento;
        double mensual = salarioNuevo / 12;

        double prestaciones;
        if (years > 12) {
            prestaciones = (SALARIO_ANUAL / 12) * 12;
        } else {
            prestaciones = (SALARIO_ANUAL / 12) * years;
        }

        String resultado = "==== Liquidación de Sueldo ====\n" +
            "Años de servicio: " + years + "\n" +
            "Salario anual base: B/. " + df.format(SALARIO_ANUAL) + "\n" +
            "Aumento: " + (int) (aumento * 100) + "%\n" +
            "Monto del aumento: B/. " + df.format(montoAumento) + "\n" +
            "Nuevo salario anual: B/. " + df.format(salarioNuevo) + "\n" +
            "Salario mensual: B/. " + df.format(mensual) + "\n" +
            "==== Prestaciones ====\n" +
            "Liquidación: B/. " + df.format(prestaciones);

        JOptionPane.showMessageDialog(null, resultado);

    } catch (NumberFormatException e) {
        JOptionPane.showMessageDialog(null, "Debe ingresar solo números.", "Error", JOptionPane.ERROR_MESSAGE);
    }
}
```

- Output:** Run (CostoDiarioConduccion)

Resultado:

