



**UNIVERSIDAD TECNOLÓGICA DE PANAMÁ**  
**CENTRO REGIONAL DE CHIRIQUÍ**  
**LICENCIATURA EN CIBERSEGURIDAD**



**Laboratorio Práctico2**

**Asignatura:**

Programación 1

**Código:**

2S31111

**Integrantes:**

Magdalena Gonzalez 4-819-1590

Irvin Martinez 4-834-1736

**Profesor:**

Napoleón Ibarra

**PRIMER SEMESTRE**

**2025**

## Desarrollo de problemas en Java. Valor 70 Puntos

Nota: contemple dentro de su código cálculos con decimales, a su vez 2 cifras significativas después del punto. Utilice JOptionPane (entrada, salida).

- 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**

```
package com.mycompany.conversorhora;
```

```
import java.util.Scanner;
```

```
import javax.swing.JOptionPane;
```

```
import javax.swing.JPanel;
```

```
import java.awt.Color;
```

```
import javax.swing.JLabel;
```

```
public class ConversorHora {
```

```
    public static void main(String[] args) {
```

```
        // 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.
```

```
        Scanner in = new Scanner(System.in);
```

```

boolean continuar = true;

do{

    try{

        String hora = JOptionPane.showInputDialog("Introduce la hora en formato 24h (HH:MM):
");

        if (hora == null){

            continuar = false;

            break;

        }

        if (hora.length()!=5 || hora.charAt(2)!=':'){

            throw new IllegalArgumentException("Formato incorrecto. " + hora + " Recuerda que
es HH:MM");

        }

        int     horaNum     =     Integer.parseInt(hora.substring(0,2)),     horaMin     =
Integer.parseInt(hora.substring(3,5));

        if(horaNum<0 || horaNum>23 || horaMin<0 || horaMin>59){

            throw new IllegalArgumentException("La hora " + hora + " no está en el rango. :(");

        }

        String tiempo = (horaNum<12) ? "AM" : "PM";

        int horaDoce = (horaNum == 0) ? 12 : (horaNum > 12 ? horaNum - 12 : horaNum);

        double horaDecimal = horaNum + horaMin / 60.0;

```

```

String horaDecimalStr = String.format("%.2f", horaDecimal);

String mensaje = "Hora en formato 12h: " + horaDoce + ":" + String.format("%02d",
horaMin) + " " + tiempo
                + "\n Hora en decimal (24h): " + horaDecimalStr;

// Crear panel para cambiar color según hora
JPanel panel = new JPanel();
if(horaNum >= 6 && horaNum < 18){
    panel.setBackground(new Color(255, 223, 186)); // color día
} else {
    panel.setBackground(new Color(50, 50, 80)); // color noche
}

JLabel label = new JLabel(mensaje);
label.setForeground((horaNum >= 6 && horaNum < 18) ? Color.BLACK : Color.WHITE);
panel.add(label);

JOptionPane.showMessageDialog(null, panel, "Conversión de hora ",
JOptionPane.INFORMATION_MESSAGE);

} catch (Exception e){

    JOptionPane.showMessageDialog(null, "Error: " + e.getMessage(), "Entrada inválida",
JOptionPane.ERROR_MESSAGE);

}

} while(continuar);

```

```
}  
}
```



ConversorHora.java

**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?**

```
package com.mycompany.viaje;
```

```
import javax.swing.JOptionPane;
```

```
public class recibeDatos {
```

```
    public static double pedir(String mensaje) {
```

```
while (true) {  
    try {  
        String entrada = JOptionPane.showInputDialog(mensaje);  
  
        if (entrada == null) {  
            JOptionPane.showMessageDialog(null, "Invalido: " + entrada);  
            System.exit(0);  
        }  
        double valor = Double.parseDouble(entrada);  
        if (valor < 0) {  
            JOptionPane.showMessageDialog(null, "Valores negativos no :c ");  
        } else {  
            return valor;  
        }  
    } catch (NumberFormatException e) {  
        JOptionPane.showMessageDialog(null, "Numero invalido");  
    }  
}  
  
}
```

package com.mycompany.viaje;

```

public class Calculos {

    private double kmDia, costoLitros, kmLitro, estacionar, peaje;


    public Calculos(double kmDia, double costoLitros, double kmLitro, double estacionar,
double peaje){

        this.kmDia= kmDia;

        this.costoLitros= costoLitros;

        this.kmLitro= kmLitro;

        this.estacionar= estacionar;

        this.peaje= peaje;

    }


    public double costoxDia(){

        return (kmDia/kmLitro)* costoLitros + estacionar + peaje;

    }


    public double ahorro(int personas){

        return personas > 1 ? costoxDia() - (costoxDia()/personas):0;

    }

}

package com.mycompany.viaje;

import javax.swing.*;

import java.util.Scanner;

```

```

public class Viaje {

    public static void main(String[] args) {

        // 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?

        Scanner in = new Scanner(System.in);

        double km= recibeDatos.pedir("Inserta los kilometros conducidos por dia: "),

        costoLitros = recibeDatos.pedir("Inserta el costo por litro de combustible: "),

        kmLitros = recibeDatos.pedir("¿Cual es el rendimiento de km por litro?: "),

        estacionamiento = recibeDatos.pedir("¿Valor del estacionamiento?: "),

        peaje= recibeDatos.pedir("Inserta el peaje: ");
    }
}

```



```

Calculos calculo = new Calculos(km, costoLitros, kmLitros, estacionamiento, peaje);

double costo = calculo.costoxDia();

int personas = (int) recibeDatos.pedir("¿Con cuantas personas compartes el camino
(incluyendote)?");

double ahorro = calculo.ahorro(personas);

JOptionPane.showMessageDialog(null,

    "DATOS INTRODUCIDOS:\n" +

        "Kilómetros por día: " + km + "\n" +

        "Costo por litro: " + costoLitros + "\n" +

        "Km/L: " + kmLitros + "\n" +

        "Estacionamiento: " + estacionamiento + "\n" +

        "Peaje: " + peaje + "\n\n" +

        "RESULTADOS:\n" +

        "Costo diario: $" + String.format("%.2f", costo) + "\n" +

        "Ahorro compartido con " + personas + " personas: $" + String.format("%.2f",
ahorro)

    );

}

}

```

- 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**

```
import javax.swing.JOptionPane;

public class precioProductoTryCatch {

    public static void main(String [] args) {

        double precioParcial = 0, precioNeto = 0, precioUnitario = 0, descuento = 0 ;

        int cantidad = 0;

        String respuesta;

        boolean validez = false;

        // Pedir precio unitario

        while (true) {

            try {

                String inputPrecio = JOptionPane.showInputDialog(null, "Ingrese el precio unitario del producto:");

                if (inputPrecio == null) {

                    JOptionPane.showMessageDialog(null, "Debe ingresar un valor.");

                    continue;

                }

                precioUnitario = Double.parseDouble(inputPrecio);

                if (precioUnitario > 0) { // Validar que sea positivo

                    break;

                } else {

                    JOptionPane.showMessageDialog(null, "El precio debe ser mayor que 0.");

                }

            }

        }

    }

}
```

```

    }

    } catch (NumberFormatException e) {

        JOptionPane.showMessageDialog(null, "Entrada no válida. Debe ingresar un
número.");

    }

}

// Pedir cantidad

while (true) {

    try {

        String inputCantidad = JOptionPane.showInputDialog(null, "Ingresa la cantidad
de venta:");

        if (inputCantidad == null) {

            JOptionPane.showMessageDialog(null, "Debe ingresar un valor.");

            continue;

        }

        cantidad = Integer.parseInt(inputCantidad);

        if (cantidad > 0) { // Validar que sea positivo

            break;

        } else {

            JOptionPane.showMessageDialog(null, "La cantidad debe ser mayor que 0.");

        }

    } catch (NumberFormatException e) {

```

```
JOptionPane.showMessageDialog(null, "Entrada no válida. Debe ingresar un  
número entero.");
```

```
}
```

```
}
```

```
precioParcial = precioUnitario * cantidad;
```

```
while (validez == false) {
```

```
    respuesta = JOptionPane.showInputDialog(null, "¿Mostrar descuento? s/n");
```

```
    boolean afirmacion = respuesta.equalsIgnoreCase("s");
```

```
    boolean negacion = respuesta.equalsIgnoreCase("n");
```

```
    if (afirmacion) {
```

```
        while (validez == false){
```

```
            try {
```

```
                descuento = Integer.parseInt(JOptionPane.showInputDialog(null, "¿Qué  
descuento aplicar? \n5%    10%    20%\n30%    40%    50%\nColoque el número sin  
%.");
```

```
                if (descuento == 5 || descuento == 10 || descuento == 20 || descuento == 30 ||  
descuento == 40 || descuento == 50) {
```

```
        descuento = descuento / 100.0;

        descuento = descuento * precioParcial;

        precioNeto = precioParcial - descuento;

        validez = true;

        JOptionPane.showMessageDialog(null, "Aplicando descuentos...");

    } else {

        JOptionPane.showMessageDialog(null, "Descuento no válido, vuelva a intentarlo.");

    }

} catch (NumberFormatException e) {

    JOptionPane.showMessageDialog(null, "Entrada no válida. Debe ingresar un número.");

}

}

} else if (negacion) {

    JOptionPane.showMessageDialog(null, "Continuando sin descuentos...");

    validez = true;

    precioNeto = precioParcial;

} else {
```

```

        JOptionPane.showMessageDialog(null, "Respuesta no válida. \n\nIntentelo de
nuevo.");
    }
}

```

```

        JOptionPane.showMessageDialog(null, String.format("El precio neto es: $%.2f",
precioNeto));
    }
}

```



precioProductoTryCatch.java

- 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**
- a. Si lleva más de 10 años en la empresa se le aplica un aumento del 15%.
  - b. Si lleva menos de 10 años pero más que 5 se le aplica un aumento del 10%.
  - c. 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.

```
import javax.swing.JOptionPane;
```

```

public class sueldoTrabajador {

    public static void main(String [] args) {

        /*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

```

- a. Si lleva más de 10 años en la empresa se le aplica un aumento del 15%.
- b. Si lleva menos de 10 años pero más que 5 se le aplica un aumento del 10%.
- c. 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. \*/

```
int tiempo = 0, mesesTrabajados = 0;

double sueldo = 13000, aumento = 0;

String renuncia = "", liquidado = "";

// Verificar el sueldo que le corresponde

while (true) {

    try {

        tiempo = Integer.parseInt(JOptionPane.showInputDialog(null, "Ingrese los años
trabajados en la empresa: "));

        if (tiempo > 0) {

            break;

        } else {

            JOptionPane.showMessageDialog(null, "Entrada no válida, ingrese un número
entero positivo.");

        }

    } catch (NumberFormatException e) {

        JOptionPane.showMessageDialog(null, "Entrada no válida, ingrese un número
entero.");

    }

}
```

```
}
```

```
if (tiempo >= 10) {
```

```
    aumento = sueldo * 0.15;
```

```
    sueldo = sueldo + aumento;
```

```
} else if (tiempo < 10 && tiempo >= 5){
```

```
    aumento = sueldo * 0.10;
```

```
    sueldo = sueldo + aumento;
```

```
} else if (tiempo < 5 && tiempo >= 3) {
```

```
    aumento = sueldo * 0.08;
```

```
    sueldo = sueldo + aumento;
```

```
} else if (tiempo < 3 && tiempo >= 0) {
```

```
    aumento = sueldo * 0.05;
```

```
    sueldo = sueldo + aumento;
```

```
}
```

```
// Verificar si renunció o es liquidado
```

```
while (true) {
```

```
    renuncia = JOptionPane.showInputDialog(null, "¿El trabajador renunció? s/n");
```

```
if (renuncia == null) {
```

```
    JOptionPane.showMessageDialog(null, "Debe responder \"s\" o \"n\".");
```

```
    continue;
```

```
}
```



```
if (renuncia.equalsIgnoreCase("s")) {  
    break;  
} else if (renuncia.equalsIgnoreCase("n")) {  
    liquidado = JOptionPane.showInputDialog(null, "¿El trabajador recibe  
liquidacion? s/n");
```

```
if (liquidado == null) {  
    JOptionPane.showMessageDialog(null, "Debe responder \"s\" o \"n\".");  
    continue;  
}
```

```
if (liquidado.equalsIgnoreCase("s") || (liquidado.equalsIgnoreCase("n"))) {  
    break;  
}
```

```
} else {  
    JOptionPane.showMessageDialog(null, "Entrada no válida, intente otra  
vez.\nDebe responder \"s\" o \"n\".");  
}  
}
```

```
boolean afirmacionRenuncia = renuncia.equalsIgnoreCase("s");
```

```
boolean negacionRenuncia = renuncia.equalsIgnoreCase("n");
```

```

boolean afirmacionLiquidado = liquidado.equalsIgnoreCase("s");

boolean negacionLiquidado = liquidado.equalsIgnoreCase("n");


//Si no renuncia ni se liquida se imprime el sueldo

if (negacionRenuncia && negacionLiquidado) {

    JOptionPane.showMessageDialog(null, String.format("El sueldo correspondido +
aumento es: \n$%.2f", sueldo));

// Si renuncia o se liquida se calcula

} else if (afirmacionRenuncia) {

while (true) {

    try {

        mesesTrabajados = Integer.parseInt(JOptionPane.showInputDialog(null,

            "Ingrese cuántos meses trabajó en este año: "));

        if (mesesTrabajados >= 0 && mesesTrabajados <= 12) {

            break;

        } else {

            JOptionPane.showMessageDialog(null, "Debe ingresar un número entre 0 y
12.");

        }

    } catch (NumberFormatException e) {

        JOptionPane.showMessageDialog(null, "Entrada no válida, ingrese un número
entero.");

    }
}

```

```
}
```

```
double decimoTercero = sueldo / 12 * mesesTrabajados;
```

```
double vacaciones = sueldo / 30 * (2.5 * mesesTrabajados);
```

```
double totalRenuncia = decimoTercero + vacaciones;
```

```
JOptionPane.showMessageDialog(null, String.format("Total a recibir por renuncia:  
\nDécimo Tercero: $%.2f\nVacaciones: $%.2f\nTotal: $%.2f", decimoTercero, vacaciones,  
totalRenuncia));
```

```
} else if (afirmacionLiquidado) {
```

```
while (true) {
```

```
try {
```

```
mesesTrabajados = Integer.parseInt(JOptionPane.showInputDialog(null,
```

```
"Ingrese cuántos meses trabajó en este año: "));
```

```
if (mesesTrabajados >= 0 && mesesTrabajados <= 12) {
```

```
break;
```

```
} else {
```

```
JOptionPane.showMessageDialog(null, "Debe ingresar un número entre 0 y  
12.");
```

```
}
```

```
} catch (NumberFormatException e) {
```

```
JOptionPane.showMessageDialog(null, "Entrada no válida, ingrese un número  
entero.");
```

```
}  
}
```

```
double decimoTercero = sueldo / 12 * mesesTrabajados;  
double vacaciones = sueldo / 30 * (2.5 * mesesTrabajados);  
double indemnizacion = 0;
```

```
if (tiempo >= 1 && tiempo < 3) {  
    indemnizacion = sueldo;  
} else if (tiempo >= 3 && tiempo < 5) {  
    indemnizacion = 2 * sueldo;  
  
} else if (tiempo >= 5 && tiempo < 10) {  
    indemnizacion = 3 * sueldo;  
} else if (tiempo >= 10) {  
    indemnizacion = 6 * sueldo;  
}
```

```
double totalLiquidacion = decimoTercero + vacaciones + indemnizacion;
```

```
JOptionPane.showMessageDialog(null, String.format("Total a recibir por  
liquidación: \nDécimo Tercero: $%.2f\nVacaciones: $%.2f\nIndemnización: $%.2f\nTotal:  
$%.2f", decimoTercero, vacaciones, indemnizacion, totalLiquidacion));
```

```
}
```

}

}



sueldoTrabajador.java