



UNIVERSIDAD TECNOLÓGICA DE PANAMÁ

**FACULTAD DE INGENIERÍA DE SISTEMAS
COMPUTACIONALES**



LICENCIATURA EN CIBERSEGURIDAD

PROGRAMACION 1

PARCIAL 1

PREPARADO POR
ADRIAN JIMENEZ 4-839-2413

A CONSIDERACIÓN DE: NAPOLEON
IBARRA

GRUPO: 2S3111

FECHA: 1-10-2025

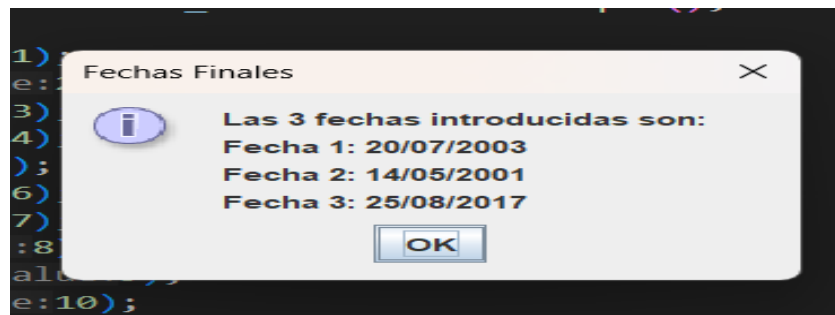
PROBLEMA 1

```
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import javax.swing.JOptionPane;

public class Problema1 {

    private static final Map<String, Integer> MAPA_MESES = new HashMap<>();
    static {
        MAPA_MESES.put(key:"enero", value:1);
        MAPA_MESES.put(key:"febrero", value:2);
        MAPA_MESES.put(key:"marzo", value:3);
        MAPA_MESES.put(key:"abr", value:4);
        MAPA_MESES.put(key:"mayo", value:5);
        MAPA_MESES.put(key:"junio", value:6);
        MAPA_MESES.put(key:"julio", value:7);
        MAPA_MESES.put(key:"agosto", value:8);
        MAPA_MESES.put(key:"septiembre", value:9);
        MAPA_MESES.put(key:"octubre", value:10);
        MAPA_MESES.put(key:"noviembre", value:11);
        MAPA_MESES.put(key:"diciembre", value:12);
    }

    Run main [Debug main]
    public static void main(String[] args) {
        List<String> fechasFormateadas = new ArrayList<>();
        DateTimeFormatter formatoFinal = DateTimeFormatter.ofPattern(pattern:"dd/MM/yyyy");
        final int NUM_FECHAS = 3;
        for (int i = 1; i <= NUM_FECHAS; i++) {
            LocalDate fecha = obtenerFechaValida(" " + i);
            if (fecha != null) {
                fechasFormateadas.add(fecha.format(formatoFinal));
            }
        }
        JOptionPane.showMessageDialog(null, fechasFormateadas);
    }
}
```

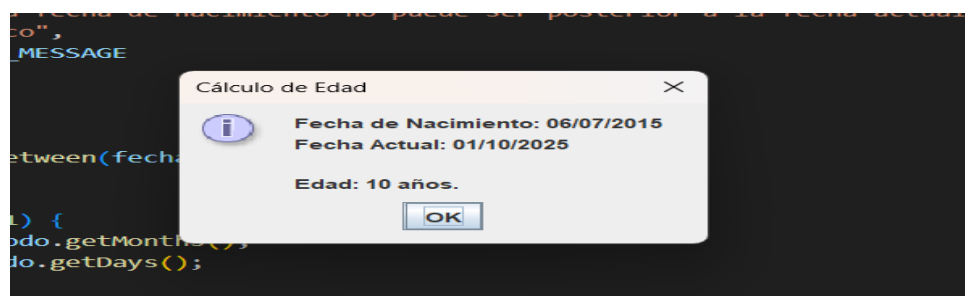


PROBLEMA 2

```

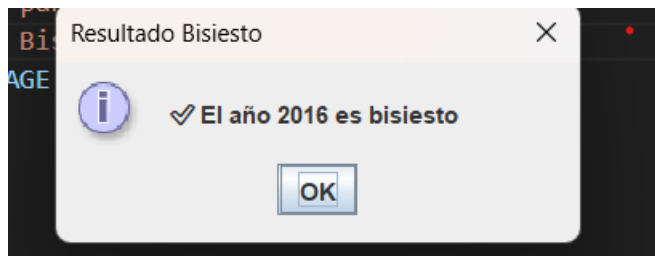
1  Users > adria > OneDrive > Documentos > UTP > II Semestre > Programación > J > Problema2.java > Java
2  import java.time.LocalDate;
3  import java.time.Period;
4  import java.time.Format.DateTimeFormatter;
5  import java.time.Format.DateTimeParseException;
6  import javax.swing.JOptionPane;
7
8  public class Problema2 {
9
10     private static final DateTimeFormatter FORMATO_ENTRADA = DateTimeFormatter.ofPattern(pattern:"dd/MM/yyyy");
11
12     Run main | Debug main
13     public static void main(String[] args) {
14
15         LocalDate fechaNacimiento = solicitarFecha(mensaje:"Introduce fecha de nacimiento:");
16         if (fechaNacimiento == null) return;
17
18         LocalDate fechaActual = solicitarFecha(mensaje:"Introduce fecha actual:");
19         if (fechaActual == null) return;
20
21         if (fechaNacimiento.isAfter(fechaActual)) {
22             JOptionPane.showMessageDialog(
23                 parentComponent:null,
24                 message:"Error: La fecha de nacimiento no puede ser posterior a la fecha actual.",
25                 title:"Error Lógico",
26                 JOptionPane.ERROR_MESSAGE
27             );
28             return;
29         }
30         Period periodo = Period.between(fechaNacimiento, fechaActual);
31         String resultado;
32         if (periodo.getYears() < 1) {
33             int totalMeses = periodo.getMonths();

```



PROBLEMA 3

```
C:\Users> adria > OneDrive > Documents > UTP > II Semestre > Programacion 1 > Problema3.java > La...
1 import javax.swing.JOptionPane;
2
3 public class Problema3 {
4
5     Run main | Debug main
6     public static void main(String[] args) {
7         String anioStr;
8         int anio = 0;
9         boolean entradaValida = false;
10
11         while (!entradaValida) {
12             anioStr = JOptionPane.showInputDialog(
13                 parentComponent:null,
14                 message:"Introduce un año para saber si es bisiesto :",
15                 title:"Verificador de Año Bisiesto",
16                 JOptionPane.QUESTION_MESSAGE
17             );
18             if (anioStr == null) return;
19             try {
20                 anio = Integer.parseInt(anioStr.trim());
21                 entradaValida = true;
22
23                 boolean esBisiesto = esAñoBisiesto(anio);
24
25                 String resultado;
26                 if (esBisiesto) {
27                     resultado = "✅ El año " + anio + " es bisiesto";
28                 } else {
29                     resultado = "❌ El año " + anio + "no es bisiesto.";
30                 }
31             }
32         }
33     }
34 }
```



PROBLEMA 4

```
C:\Users> adria > OneDrive > Documents > UTP > II Semestre > Programacion 1 > Problema4.java > Jav...
1 import javax.swing.JOptionPane;
2
3 public class Problema4 {
4
5     Run main | Debug main
6     public static void main(String[] args) {
7         String horasStr, tasaStr;
8         double horasTrabajadas = 0;
9         double tasaOrdinaria = 0;
10        boolean entradaValida = false;
11        while (!entradaValida) {
12            horasStr = JOptionPane.showInputDialog(parentComponent:null, message:"Introduce las HORAS semanales trabajadas:", title:"Entrada de Datos");
13            if (horasStr == null) return;
14            tasaStr = JOptionPane.showInputDialog(parentComponent:null, message:"Introduce la TASA horaria ordinaria (ej: 10.50 Balboas):", title:"Entrada de Datos");
15            if (tasaStr == null) return;
16            try {
17                horasTrabajadas = Double.parseDouble(horasStr);
18                tasaOrdinaria = Double.parseDouble(tasaStr);
19
20                if (horasTrabajadas < 0 || tasaOrdinaria <= 0) {
21                    JOptionPane.showMessageDialog(parentComponent:null, message:"Las horas y la tasa deben ser valores positivos válidos.", title:"Error");
22                    continue;
23                }
24                entradaValida = true;
25            } catch (NumberFormatException e) {
26                JOptionPane.showMessageDialog(parentComponent:null, message:"Error: Debes introducir valores numéricos válidos para horas y tasa.", title:"Error");
27            }
28        }
29    }
30 }
```

