

C:\Users\Admin\Documents\Tecnicatura en programación\Segundo cuatrimestre\Programación II\TP\ProyectoTP8\src\interfaces\TarjetaCredito.java

```
1 package interfaces;
2
3
4 public class TarjetaCredito implements PagoConDescuento {
5     private static final double TASA_DESCUENTO = 0.05; // 5% de descuento
6
7     @Override
8     public void procesarPago(double monto) {
9         System.out.println("Procesando pago con Tarjeta de Crédito por $" + monto);
10        System.out.println("Pago con Tarjeta de Crédito exitoso.");
11    }
12
13    @Override
14    public double aplicarDescuento(double monto) {
15        System.out.println("Aplicando descuento de " + (TASA_DESCUENTO * 100) + "%...");
16        return monto * (1 - TASA_DESCUENTO);
17    }
18 }
19
20
```

C:\Users\Admin\Documents\Tecnicatura en programación\Segundo cuatrimestre\Programación II\TP\ProyectoTP8\src\interfaces\Producto.java

```
1
2 package interfaces;
3
4 import java.util.ArrayList;
5 import java.util.List;
6
7
8 public class Producto implements Pagable {
9     private String nombre;
10    private double precio;
11
12    public Producto(String nombre, double precio) {
13        this.nombre = nombre;
14        this.precio = precio;
15    }
16
17    public String getNombre() {
18        return nombre;
19    }
20
21    @Override
22    public double calcularTotal() {
23        return precio;
24    }
25
26    @Override
27    public String toString() {
28        return nombre + " ($" + precio + ")";
29    }
30 }
```

C:\Users\Admin\Documents\Tecnicatura en programación\Segundo cuatrimestre\Programación II\TP\ProyectoTP8\src\interfaces\Pedido.java

```
1 package interfaces;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 public class Pedido implements Pagable {
7     private List<Producto> productos;
8     private Cliente cliente;
9     private String estado;
10
11     public Pedido(Cliente cliente) {
12         this.productos = new ArrayList<>();
13         this.cliente = cliente;
14         this.estado = "PENDIENTE";
15         // Notificar al cliente al crear el pedido
16         notificarCambioEstado();
17     }
18
19     public void agregarProducto(Producto p) {
20         this.productos.add(p);
21     }
22
23     public void setEstado(String nuevoEstado) {
24         this.estado = nuevoEstado;
25         notificarCambioEstado();
26     }
27
28     private void notificarCambioEstado() {
29         String mensaje = "Su pedido ahora está en estado: " + this.estado;
30         if (cliente != null) {
31             cliente.notificar(mensaje); // La clase Cliente implementa Notificable
32         }
33     }
34
35     @Override
36     public double calcularTotal() {
37         double total = 0;
38         for (Producto p : productos) {
39             total += p.calcularTotal();
40         }
41         return total;
42     }
43
44     public Cliente getCliente() {
45         return cliente;
46     }
47 }
```

```
46     }  
47 }
```

C:\Users\Admin\Documents\Tecnicatura en programación\Segundo cuatrimestre\Programación II\TP\ProyectoTP8\src\interfaces\PayPal.java

```
1 package interfaces;
2
3 // PayPal.java
4 public class PayPal implements Pago {
5     @Override
6     public void procesarPago(double monto) {
7         System.out.println("Procesando pago con PayPal por $" + monto);
8         // Lógica de integración con PayPal...
9         System.out.println("Pago con PayPal exitoso.");
10    }
11 }
12
```

C:\Users\Admin\Documents\Tecnicatura en programación\Segundo cuatrimestre\Programación II\TP\ProyectoTP8\src\interfaces\PagoConDescuento.java

```
1 package interfaces;
2
3 public interface PagoConDescuento extends Pago {
4     /** Aplica un descuento al monto total. */
5     double aplicarDescuento(double monto);
6 }
```

C:\Users\Admin\Documents\Tecnicatura en programación\Segundo cuatrimestre\Programación II\TP\ProyectoTP8\src\interfaces\Pago.java

```
1 package interfaces;
2
3 public interface Pago {
4     /** Procesa el pago de un monto dado. */
5     void procesarPago(double monto);
6 }
```

C:\Users\Admin\Documents\Tecnicatura en programación\Segundo cuatrimestre\Programación II\TP\ProyectoTP8\src\interfaces\Pagable.java

```
1 package interfaces;
2
3 public interface Pagable {
4     /** Calcula el valor total de un producto o pedido. */
5     double calcularTotal();
6 }
7
8
```

C:\Users\Admin\Documents\Tecnicatura en programación\Segundo cuatrimestre\Programación II\TP\ProyectoTP8\src\interfaces\Notificable.java

```
1 package interfaces;
2
3 public interface Notificable {
4     /** Envía una notificación de un evento */
5     void notificar(String mensaje);
6 }
```

C:\Users\Admin\Documents\Tecnicatura en programación\Segundo cuatrimestre\Programación II\TP\ProyectoTP8\src\interfaces\DemoInterfaces.java

```
1 package interfaces;
2
3 public class DemoInterfaces {
4     public static void main(String[] args) {
5         // 1. Crear Cliente y Pedido
6         Cliente cliente = new Cliente("Ana Torres", "ana.t@email.com");
7         Pedido pedido = new Pedido(cliente);
8
9         // 2. Agregar Productos
10        pedido.agregarProducto(new Producto("Laptop", 100));
11        pedido.agregarProducto(new Producto("Mouse Inalámbrico", 25.50));
12
13        // 3. Calcular Total y Notificar
14        double totalOriginal = pedido.calcularTotal();
15        System.out.println("\n--- Total del Pedido ---");
16        System.out.println("Total Original: $" + totalOriginal);
17
18        // 4. Pago con Descuento (TarjetaCredito)
19        TarjetaCredito pagoTC = new TarjetaCredito();
20        double totalConDescuento = pagoTC.aplicarDescuento(totalOriginal);
21        System.out.println("Total con Descuento: $" + totalConDescuento);
22        pagoTC.procesarPago(totalConDescuento);
23
24        // 5. Notificar cambio de estado
25        pedido.setEstado("ENVIADO");
26
27        // 6. Pago sin Descuento (PayPal)
28        System.out.println("\n--- Nuevo Pedido con PayPal ---");
29        Pedido pedido2 = new Pedido(new Cliente("Juan Pérez", "juan@mail.net"));
30        pedido2.agregarProducto(new Producto("Libro Java", 40.00));
31
32        double total2 = pedido2.calcularTotal();
33        System.out.println("Total a pagar con PayPal: $" + total2);
34        new PayPal().procesarPago(total2);
35        pedido2.setEstado("ENTREGADO");
36    }
37 }
38
```

C:\Users\Admin\Documents\Tecnicatura en programación\Segundo cuatrimestre\Programación II\TP\ProyectoTP8\src\interfaces\Cliente.java

```
1
2 package interfaces;
3
4 public class Cliente implements Notificable {
5     private String nombre;
6     private String email;
7
8     public Cliente(String nombre, String email) {
9         this.nombre = nombre;
10        this.email = email;
11    }
12
13    @Override
14    public void notificar(String mensaje) {
15        System.out.println("-> NOTIFICACIÓN para Cliente " + nombre + " (" + email + "): " + mensaje);
16    }
17 }
18
```

C:\Users\Admin\Documents\Tecnicatura en programación\Segundo cuatrimestre\Programación  
II\TP\ProyectoTP8\src\excepciones\EdadInvalidaException.java

```
1 package excepciones;
2
3
4 public class EdadInvalidaException extends Exception {
5     public EdadInvalidaException(String mensaje) {
6         super(mensaje);
7     }
8 }
```

C:\Users\Admin\Documents\Tecnicatura en programación\Segundo cuatrimestre\Programación II\TP\ProyectoTP8\src\excepciones\DemoExcepciones.java

```
1 package excepciones;
2
3 import java.io.BufferedReader;
4 import java.io.FileNotFoundException;
5 import java.io.FileReader;
6 import java.io.IOException;
7 import java.util.InputMismatchException;
8 import java.util.Scanner;
9
10 public class DemoExcepciones {
11
12     public static void main(String[] args) {
13         Scanner scanner = new Scanner(System.in);
14
15         // 1. División segura
16         System.out.println("\n--- 1. División Segura ---");
17         try {
18             System.out.print("Ingrese el numerador: ");
19             int num1 = scanner.nextInt();
20             System.out.print("Ingrese el divisor: ");
21             int num2 = scanner.nextInt();
22             System.out.println("Resultado: " + (num1 / num2));
23         } catch (ArithmeticException e) {
24             System.out.println("Error: No se puede dividir por cero.");
25         } catch (InputMismatchException e) {
26             System.out.println("Error: La entrada debe ser un número entero.");
27             scanner.nextLine(); // Limpiar el buffer
28         }
29
30         // 2. Conversión de cadena a número
31         System.out.println("\n--- 2. Conversión de Cadena ---");
32         try {
33             System.out.print("Ingrese un texto para convertir a número: ");
34             String texto = scanner.next();
35             int numero = Integer.parseInt(texto);
36             System.out.println("Número convertido con éxito: " + numero);
37         } catch (NumberFormatException e) {
38             System.out.println("Error: La cadena ingresada no es un formato de número válido.");
39         }
40
41         // 4. Excepción personalizada
42         System.out.println("\n--- 4. Excepción Personalizada ---");
43         try {
44             System.out.print("Ingrese su edad: ");
45             int edad = scanner.nextInt();
```

```
46     verificarEdad(edad);
47     System.out.println("Edad válida.");
48 } catch (EdadInvalidaException e) {
49     System.out.println("Error al validar la edad: " + e.getMessage());
50 } catch (InputMismatchException e) {
51     System.out.println("Error: La edad debe ser un número entero.");
52     scanner.nextLine(); // Limpiar el buffer
53 }
54
55 // 3 y 5. Lectura de archivo y try-with-resources
56 System.out.println("\n--- 3 y 5. Lectura de Archivo (try-with-resources) ---");
57 leerArchivo("src/excepciones/ejemplo.html");
58 scanner.close();
59 }
60
61
62 public static void verificarEdad(int edad) throws EdadInvalidaException {
63     if (edad < 0 || edad > 120) {
64         throw new EdadInvalidaException("La edad (" + edad + ") debe estar entre 0 y 120.");
65     }
66 }
67
68
69 public static void leerArchivo(String nombreArchivo) {
70     // Usa try-with-resources para garantizar que BufferedReader se cierre
71     try (BufferedReader br = new BufferedReader(new FileReader(nombreArchivo))) {
72         String linea;
73         System.out.println("Contenido del archivo:");
74         while ((linea = br.readLine()) != null) {
75             System.out.println(linea);
76         }
77     } catch (FileNotFoundException e) {
78         System.out.println("Error: Archivo no encontrado. Verifique la ruta.");
79     } catch (IOException e) {
80         System.out.println("Error de I/O al leer el archivo: " + e.getMessage());
81     }
82 }
83 }
```