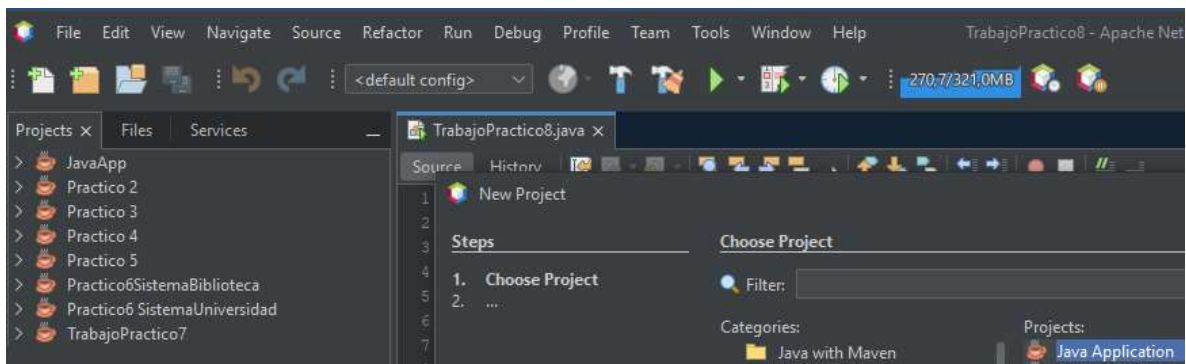


Trabajo Práctico 8 - Interfaces y Excepciones en Java

Paso 1: Configuración del proyecto en NetBeans

Abro NetBeans y creo un nuevo proyecto:

- **File -> New Project -> Java with Ant -> Java Application**
- **Project Name: TrabajoPractico8**



Paso 2: Creación de las interfaces y clases para el sistema E-commerce

Interfaz Pagable

Para crear la interfaz Pagable:

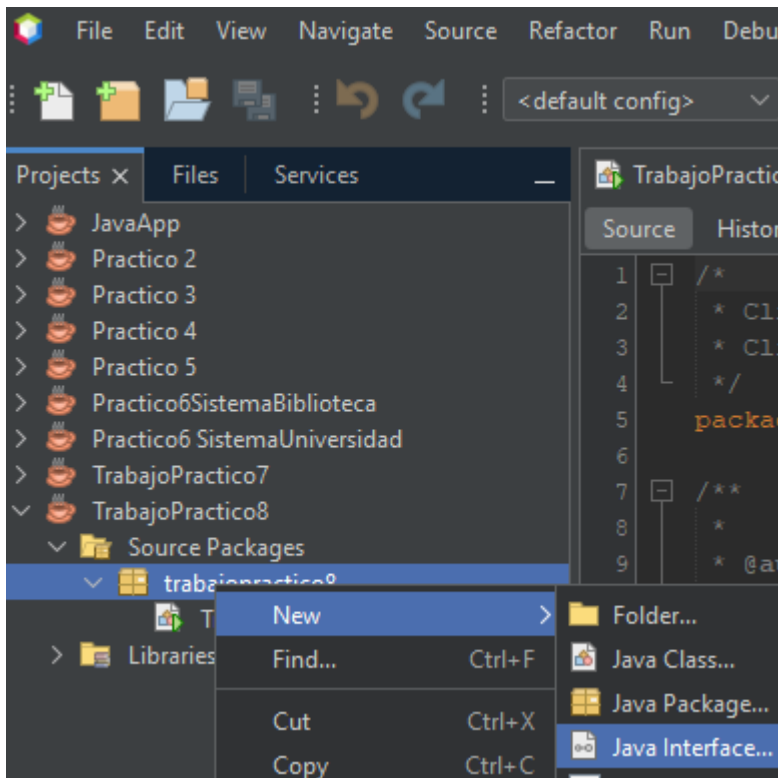
- **Clic derecho sobre trabajopractico8.ecommerce**
- **New -> Java Interface**
- **Nombre: Pagable**
- **Click en Finish**

Repetir el proceso para cada interfaz:

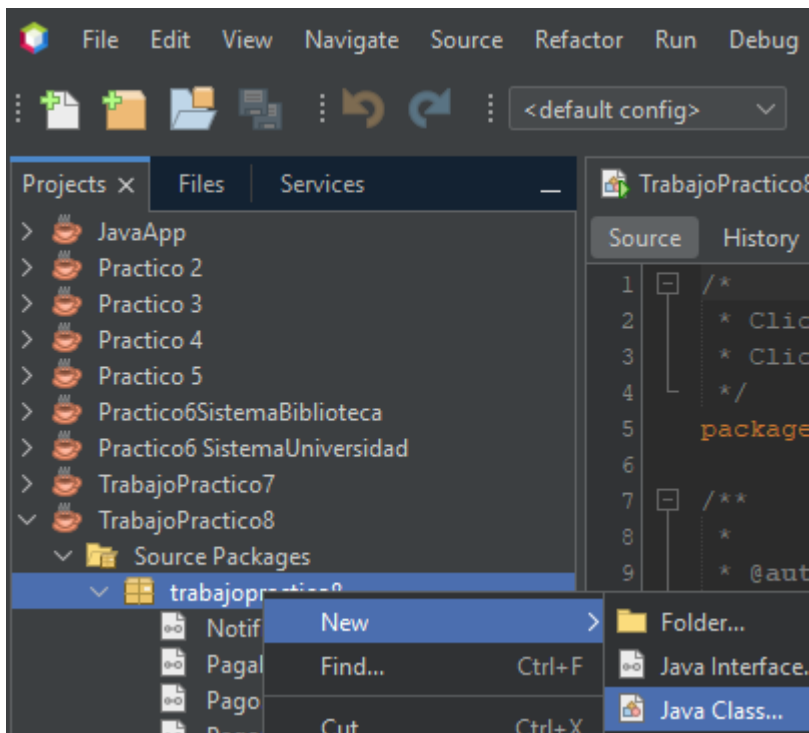
Notificable.java

Pago.java

PagoConDescuento.java



Clase Producto



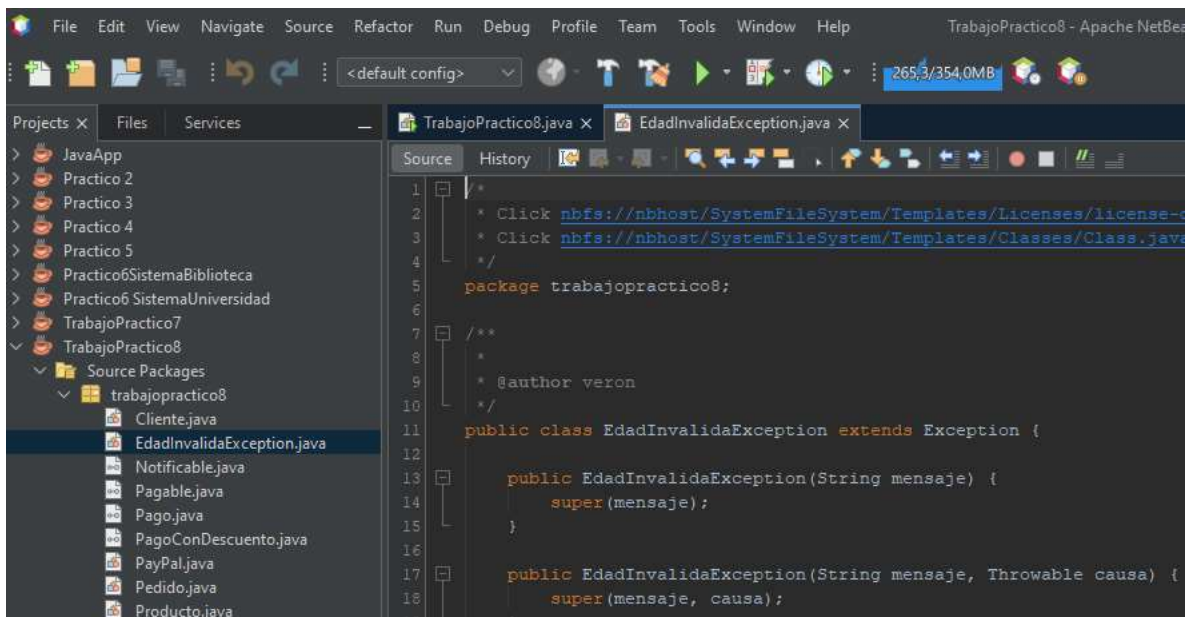
Repetir para el resto de las clases: Clase Cliente - Clase Pedido - Clase TarjetaCredito - Clase PayPal

Paso 3: Implementación de las excepciones

Excepción personalizada EdadInvalidaException

Para crear EdadInvalidaException:

- Clic derecho sobre trabajopractico8.excepciones
- New -> Java Class
- Nombre: EdadInvalidaException
- Click en Finish



```

1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java
4   */
5   package trabajopractico8;
6
7   /**
8    *
9    * @author veron
10   */
11  public class EdadInvalidaException extends Exception {
12
13      public EdadInvalidaException(String mensaje) {
14          super(mensaje);
15      }
16
17      public EdadInvalidaException(String mensaje, Throwable causa) {
18          super(mensaje, causa);
19      }
20  }
    
```

Clase ValidadorEdad



```

1  /*
2   *
3   * @author veron
4   */
5   public class ValidadorEdad {
6
7       public static void validarEdad(int edad) throws EdadInvalidaException {
8           if (edad < 0) {
9               throw new EdadInvalidaException("La edad no puede ser negativa: " + edad);
10          }
11          if (edad > 120) {
12              throw new EdadInvalidaException("La edad no puede ser mayor a 120: " + edad);
13          }
14          System.out.println("Edad valida: " + edad);
15      }
16  }
    
```

Ejecución

Salida

```

> Practico6SistemaBiblioteca
> Practico6 SistemaUniversidad
5 package trabajopractico8;

Output - TrabajoPractico8 (run) #2

run:
=== TRABAJO PRACTICO 8 - INTERFACES Y EXCEPCIONES ===

--- PARTE 1: SISTEMA E-COMMERCE ---
Productos en el pedido:
- Laptop - $1500.0
- Mouse - $25.0
- Teclado - $45.0
Total del pedido: $1570.0
Notificacion para Veronica Guirin (veronica@email.com): El estado de su pedido ha cambiado a: PROCESANDO
Notificacion para Veronica Guirin (veronica@email.com): El estado de su pedido ha cambiado a: ENVIADO
Descuento aplicado: $78.5
Procesando pago con tarjeta de credito por $1491.5
Procesando pago con PayPal (veronica@email.com) por $1570.0

--- PARTE 2: EJERCICIOS DE EXCEPCIONES ---

1. DIVISION SEGURA
Ingrese el dividendo:

```

Conclusión del trabajo

Este trabajo práctico demuestra exitosamente:

- Uso de interfaces:**
Se implementaron múltiples interfaces (Pagable, Notificable, Pago, PagoConDescuento) que permiten definir contratos de comportamiento común entre clases diferentes.
- Herencia múltiple con interfaces:**
La clase TarjetaCredito implementa PagoConDescuento que a su vez extiende Pago, mostrando cómo una clase puede heredar múltiples comportamientos.
- Manejo de excepciones:**
Se implementaron todos los tipos de manejo de excepciones solicitados:
 - ArithmeticException para división por cero
 - NumberFormatException para conversión de cadenas
 - FileNotFoundException para archivos inexistentes
 - Excepción personalizada EdadInvalidaException
 - Uso de try-with-resources para manejo automático de recursos
- Buenas prácticas:** Se aplicó try-with-resources, bloques catch específicos, y excepciones personalizadas para validaciones de negocio.