

PROGRAMACIÓN II

TRABAJO PRÁCTICO 8: Interfaces y excepciones

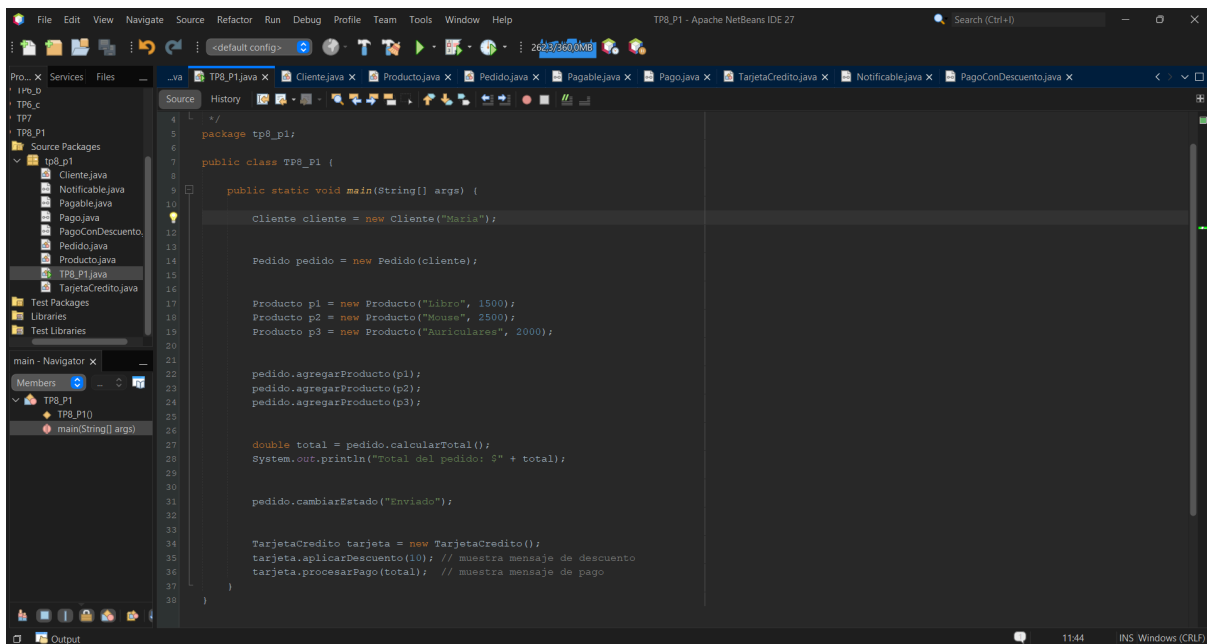
Alumna: Garcia Galfione, Fiorella

Enlace a repositorio en Github: <https://github.com/FioreGG/UTN-TUPaD-P2.git>

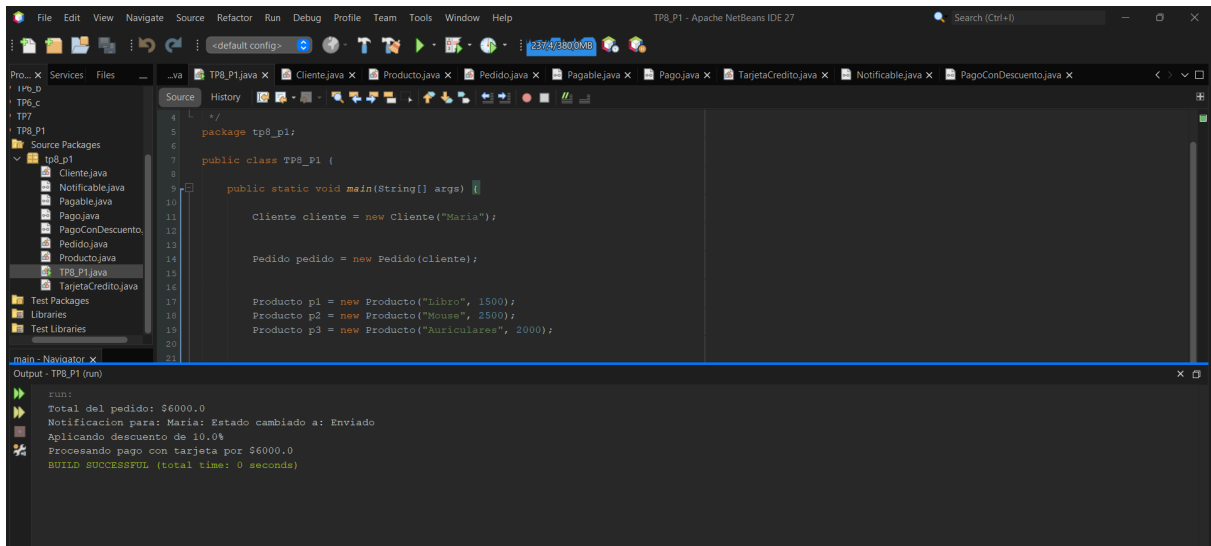
PARTE 1

1. Crear una interfaz Pagable con el método calcularTotal().
2. Clase Producto: tiene nombre y precio, implementa Pagable.
3. Clase Pedido: tiene una lista de productos, implementa Pagable y calcula el total del pedido
4. Ampliar con interfaces Pago y PagoConDescuento para distintos medios de pago (TarjetaCredito, PayPal), con metodos procesarPago(double) y aplicarDescuento(double).
5. Crear una interfaz Notificable para notificar cambios de estado. La clase Cliente implementa dicha interfaz y Pedido debe notificarlo al cambiar de estado.

Main y resultado por consola del ejercicio:



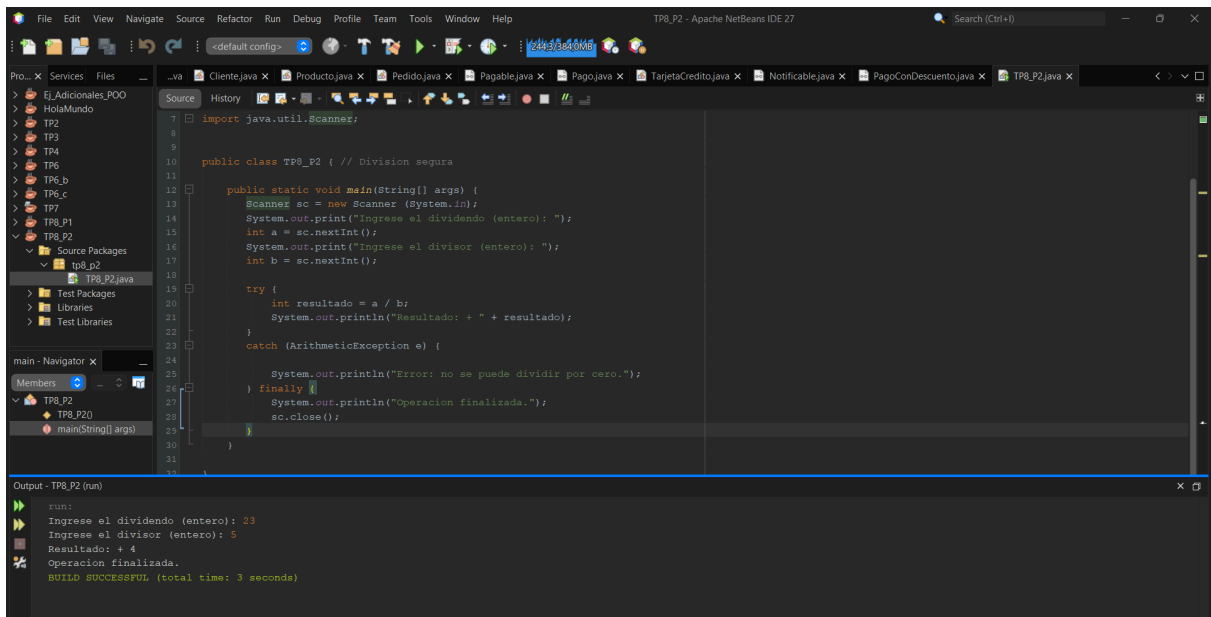
```
1 package tp8_p1;
2
3 public class TP8_P1 {
4
5     public static void main(String[] args) {
6
7         Cliente cliente = new Cliente("Maria");
8
9         Pedido pedido = new Pedido(cliente);
10
11         Producto p1 = new Producto("Libro", 1500);
12         Producto p2 = new Producto("Mouse", 2500);
13         Producto p3 = new Producto("Auriculares", 2000);
14
15         pedido.agregarProducto(p1);
16         pedido.agregarProducto(p2);
17         pedido.agregarProducto(p3);
18
19         double total = pedido.calcularTotal();
20         System.out.println("Total del pedido: $" + total);
21
22         pedido.cambiarEstado("Enviado");
23
24         TarjetaCredito tarjeta = new TarjetaCredito();
25         tarjeta.aplicarDescuento(10); // muestra mensaje de descuento
26         tarjeta.procesarPago(total); // muestra mensaje de pago
27     }
28 }
```

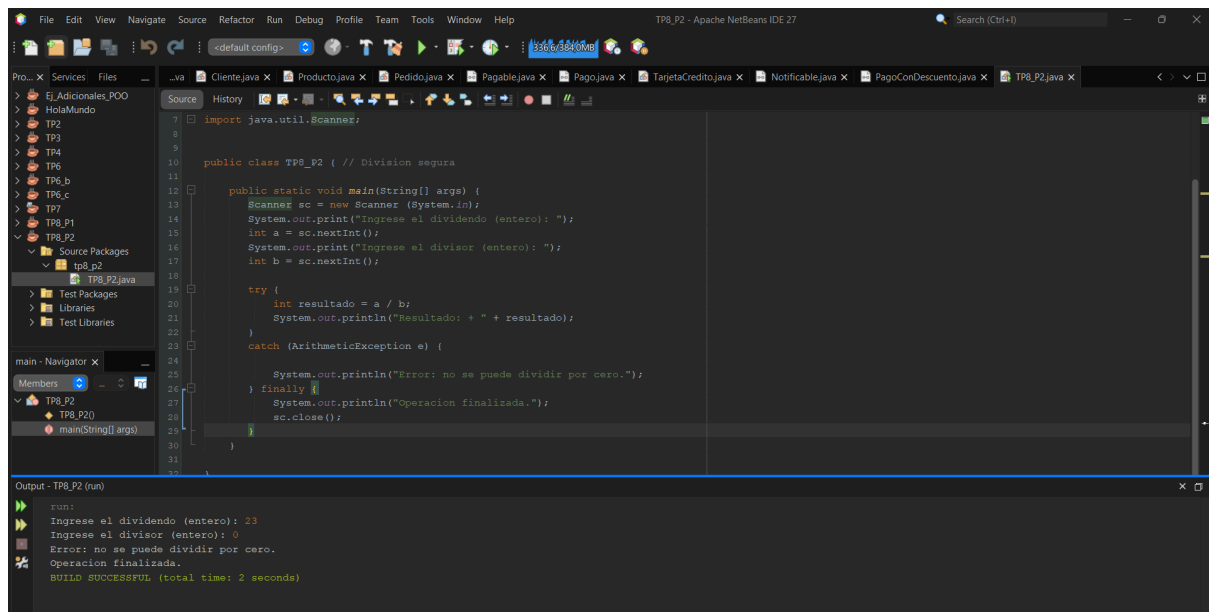


PARTE 2: Ejercicios sobre Excepciones

1. División segura

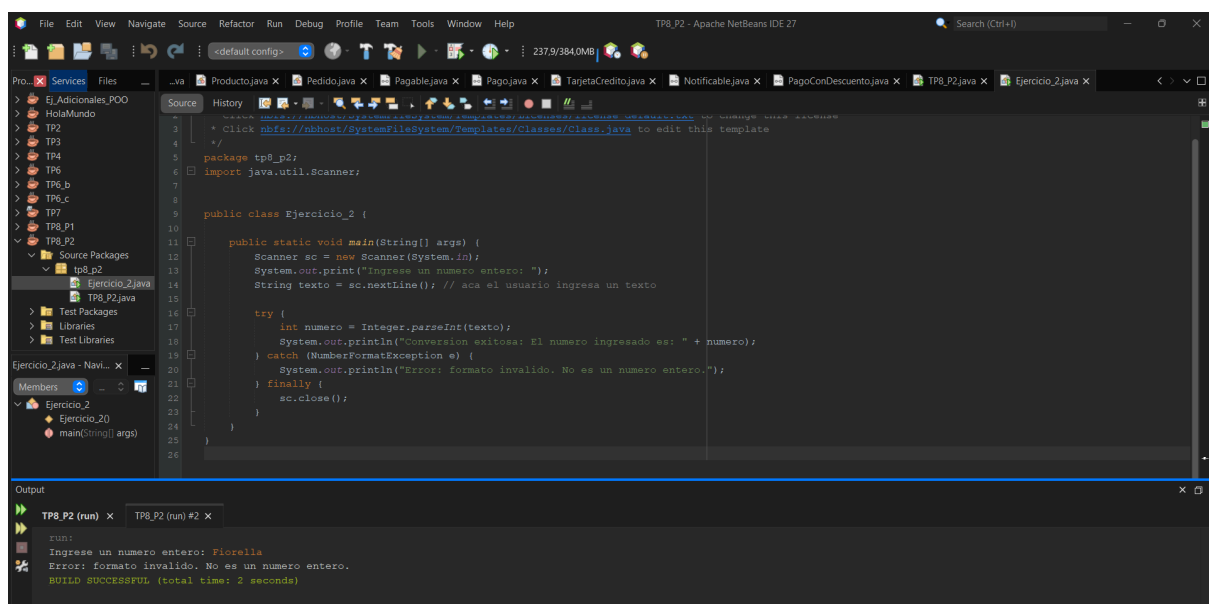
- Solicitar dos números y dividirlos. Manejar `ArithmeticException` si el divisor es cero.





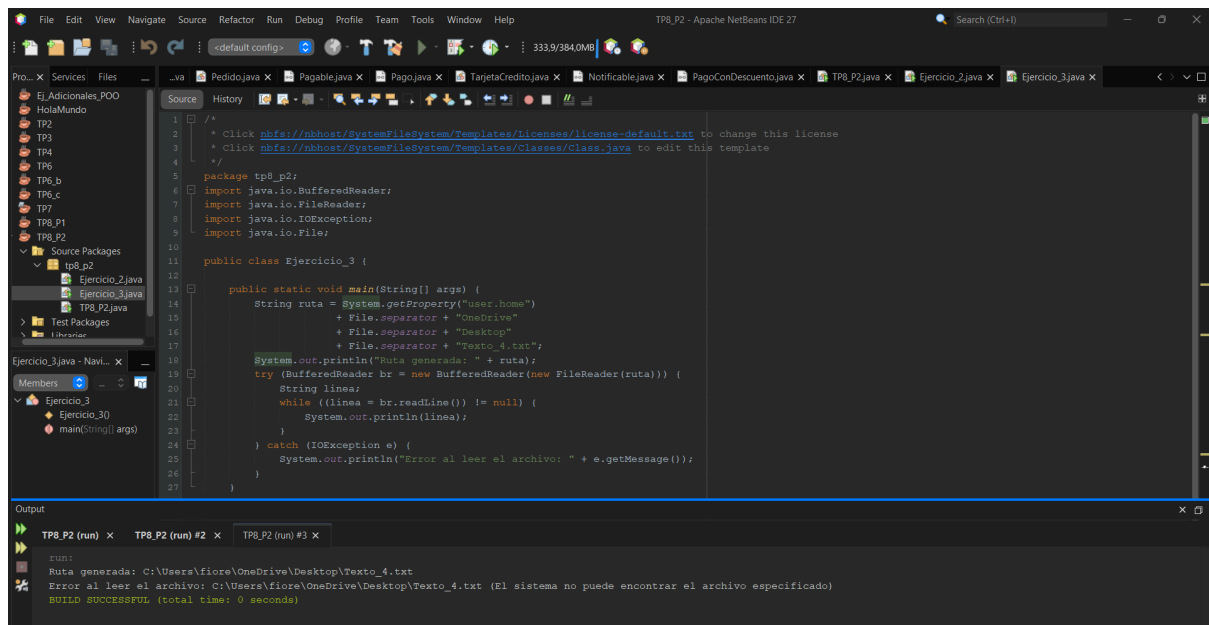
2. Conversión de cadena a número

- Leer texto del usuario e intentar convertirlo a int. Manejar NumberFormatException si no es válido.



3. Lectura de archivo

- Leer un archivo de texto y mostrarlo. Manejar FileNotFoundException si el archivo no existe.



4. Excepción personalizada

- Crear `EdadInvalidaException`. Lanzarla si la edad es menor a 0 o mayor a 120. Capturarla y mostrar mensaje.

5. Uso de try-with-resources

- Leer un archivo con `BufferedReader` usando try-with-resources. Manejar `IOException` correctamente.

Me resta completar puntos 4 y 5, que los voy a actualizar directamente en el repositorio.
PENDIENTE AMBOS PUNTOS