|  |  |  |  |
| --- | --- | --- | --- |
| **CONSTRUCCION DE SOFTWARE.**  mANEJO DE EXCEPCIONES Y ERRORES EN JAVA. | | | |
| Tecnológico de Antioquia | Freddy Alexander Ramírez Rivera | freddy.ramires@tdea.edu.co | 11 - Mayo - 2025 |

|  |  |
| --- | --- |
| 1. **Manejo De Excepciones Y Errores En JAVA** | **En Java,** errores **y** excepciones:son problemas que ocurren durante la ejecución de un programa. Aunque ambos pueden interrumpir el flujo del programa, se manejan de manera diferente:  **Errores (**Error**)**: Son fallos graves que normalmente no pueden ser manejados por el programa, como la falta de memoria o errores internos de la JVM.  **Excepciones (**Exception**)**: Son situaciones inesperadas que pueden ser controladas, como intentar dividir por cero o acceder a un archivo que no existe.  Cláusulas try y catch  La estructura try-catch permite manejar excepciones en Java, evitando que el programa falle por completo.  try {  // Código que puede generar una excepción  } catch (TipoDeExcepcion nombreVariable) {  // Código para manejar la excepción  } finally {  // (Opcional) Código que se ejecuta siempre, haya excepción o no  }  Significado de cada parte:  try: Contiene el código propenso a generar excepciones.  catch: Captura y maneja la excepción especificada.  finally (opcional): Se ejecuta siempre, ocurra o no una excepción, puede ser útil para liberar recursos.  TipoDeExcepcion: Es el tipo específico de excepción que se desea capturar (por ejemplo, ArithmeticException, IOException, etc.).  NombreVariable: Es el nombre que le damos a la excepción capturada, que puede usarse dentro del bloque catch para obtener más detalles.  Ejemplo 1: Manejo de división por cero java  public class EjemploTryCatch {  public static void main(String[] args) {  try {  int resultado = 10 / 0; // Esto genera una excepción  } catch (ArithmeticException e) {  System.out.println("Error: División por cero.");  }  }  }  Salida: Error: División por cero.  Ejemplo 2: Manejo de lectura de archivos  import java.io.File;  import java.io.FileNotFoundException;  import java.util.Scanner;  public class LeerArchivo {  public static void main(String[] args) {  try {  File archivo = new File("texto.txt");  Scanner lector = new Scanner(archivo);  while (lector.hasNextLine()) {  System.out.println(lector.nextLine());  }  lector.close();  } catch (FileNotFoundException e) {  System.out.println("Error: Archivo no encontrado.");  }  }  }  **Salida (si el archivo no existe):** Error: Archivo no encontrado. |
|  |  |