Examen Unidad 1

Programación Orientada a Objetos

Alumno: Evelyn de los Ángeles López Ávila.

Fecha: 14/febrero/2025

Objetivo:

El objetivo de este caso práctico es que los estudiantes universitarios aprendan a utilizar las herramientas de depuración de NetBeans para identificar y corregir errores en un programa Java. A través de esta actividad, los estudiantes desarrollarán habilidades para establecer puntos de interrupción, inspeccionar variables y controlar el flujo de ejecución del programa.

Descripción:

Se proporcionará a los estudiantes un programa Java que contiene varios errores comunes, tanto de lógica como de sintaxis. Los estudiantes deberán importar el código en NetBeans y utilizar las herramientas de depuración para identificar y corregir los errores presentes. El programa simula una simple calculadora que realiza operaciones básicas como suma, resta, multiplicación y división.

Instrucciones:

1. **Importar el proyecto en NetBeans:**
    - Cree un nuevo proyecto en NetBeans y añada una nueva clase llamada Calculadora.java.
    - Copie y pegue el siguiente código en la clase Calculadora.java.
2. **Analizar el código:**
    - Revise el código proporcionado y observe su estructura y funcionalidad prevista.
3. **Establecer puntos de interrupción:**
    - Identifique las secciones del código donde sospecha que pueden existir errores.
    - Haga clic en el margen izquierdo de la línea de código para establecer un punto de interrupción (breakpoint).

4. **Iniciar la depuración:**
    - Ejecute el programa en modo depuración seleccionando "Depurar" > "Depurar proyecto principal" o presionando Ctrl + F5.
5. **Inspeccionar variables y flujo de ejecución:**
    - Utilice las herramientas de NetBeans para inspeccionar los valores de las variables en tiempo de ejecución.
    - Avance paso a paso por el código utilizando las opciones "Step Into" (F7) y "Step Over" (F8).
6. **Identificar y corregir errores:**
    - Observe los comportamientos anómalos y los valores incorrectos de las variables.
    - Realice las correcciones necesarias en el código y vuelva a ejecutar el programa para verificar que los errores hayan sido solucionados.

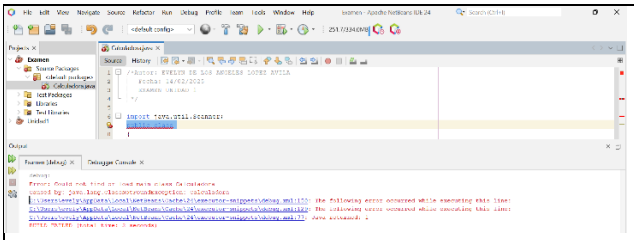
```java
import java.util.Scanner;

Public class {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        boolean continuar = true

        while (continuar) {
            System.out.println("Seleccione una operación:";
            System.out.println("1. Suma");
            System.out.println("2. Resta");
            System.out.println("3. Multiplicación");
            System.out.println("4. División");
            System.out.println("5. Salir");
            int opcion = scanner.nextInt();

            if (opcion = 5)
                continuar = false;
                System.out.println("Saliendo...");
                break;


            System.out.print("Ingrese el primer número: ");
            double num1 = scanner.nextDouble();
            System.out.print("Ingrese el segundo número: ");
            double num2 = scanner.nextDouble();
```

```
27
28              //double resultado = 0;
29              boolean error = false;
30
31              switch (opcion) {
32                  case 1:
33                      resultado = num1 + num2;
34                      break;
35                  case 2:
36                      resultado = num1 - num2;
37                      break;
38                  case 3:
39                      resultado = num1 * num2;
40                      break;
41                  case 4:
42                      if (num2 != 0) {
43                          resultado = num1 / num2;
44                      } else {
45                          System.out.println("Error: División por cero.");
46                          error = true;
47                      }
48                      break;
49                  default:
50                      System.out.println("Opción no válida.);
51                      error = true;
52              }
```

```
53
54              if (!error) {
55                  System.out.println("El resultado es: " + resultado);
56              }
57          }
58
59          scanner.close();
60      }
61  }
```

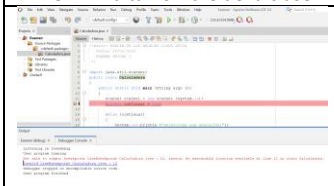
Resultado

En la siguiente tabla agregue tantos renglones como sean necesarias.
Cada error deberá anexarse a un renglón indicando el numero de línea y el código con error en la columna 1, en la columna 2 el error solucionado.
El penúltimo renglón deberá contener el código completo (corregido).
El último renglón deberá contener evidencia de la ejecución.



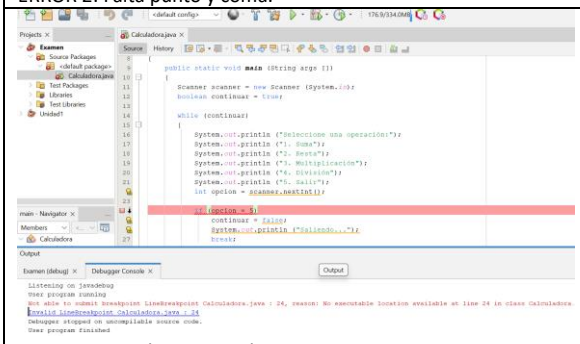
```
5
6   import java.util.Scanner;
7   public class Calculadora
8   {
```

ERROR 1: Falta nombre de la clase.



```
11          Scanner scanner = new Scanner (System.in);
12          boolean continuar = true;
```

ERROR 2: Falta punto y coma.



```
23                  if (opcion == 5)
                        continuar = false;
```

ERROR 3: Uso de = en vez de ==

```
23
24          if (opcion = 5)
                continuar = false;
26              System.out.println ("Saliendo...");
27              break;
28
```

```
23
24          if (opcion == 5){
                continuar = false;
26              System.out.println ("Saliendo...");
27              break;
28          }
```

ERROR 4: Falta de llaves al usar if.

```
34
35          //double resultado = 0
36          boolean error = false;
```

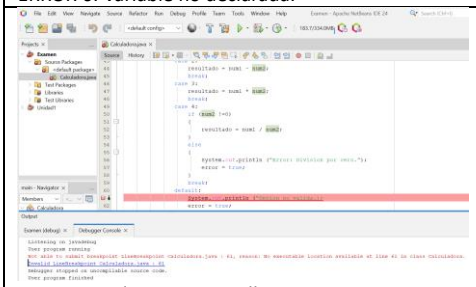
```
34
35              double resultado = 0;
36              boolean error = false;
```

ERROR 5: Variable hecha comentario y no declarada.

```
40          case 1:
41              resultado = num1 + num2;
42              break;
43          case 2:
44              resultado = num1 - num2;
45              break;
46          case 3:
47              resultado = num1 * num2;
48              break;
49          case 4:
50              if (num2 !=0)
51              {
52                  resultado = num1 / num2;
53              }
```

```
40          case 1:
41              resultado = num1 + num2;
42              break;
43          case 2:
44              resultado = num1 - num2;
45              break;
46          case 3:
47              resultado = num1 * num2;
48              break;
49          case 4:
50              if (num2 !=0)
51              {
52                  resultado = num1 / num2;
53              }
```

ERROR 6: Variable no declarada.



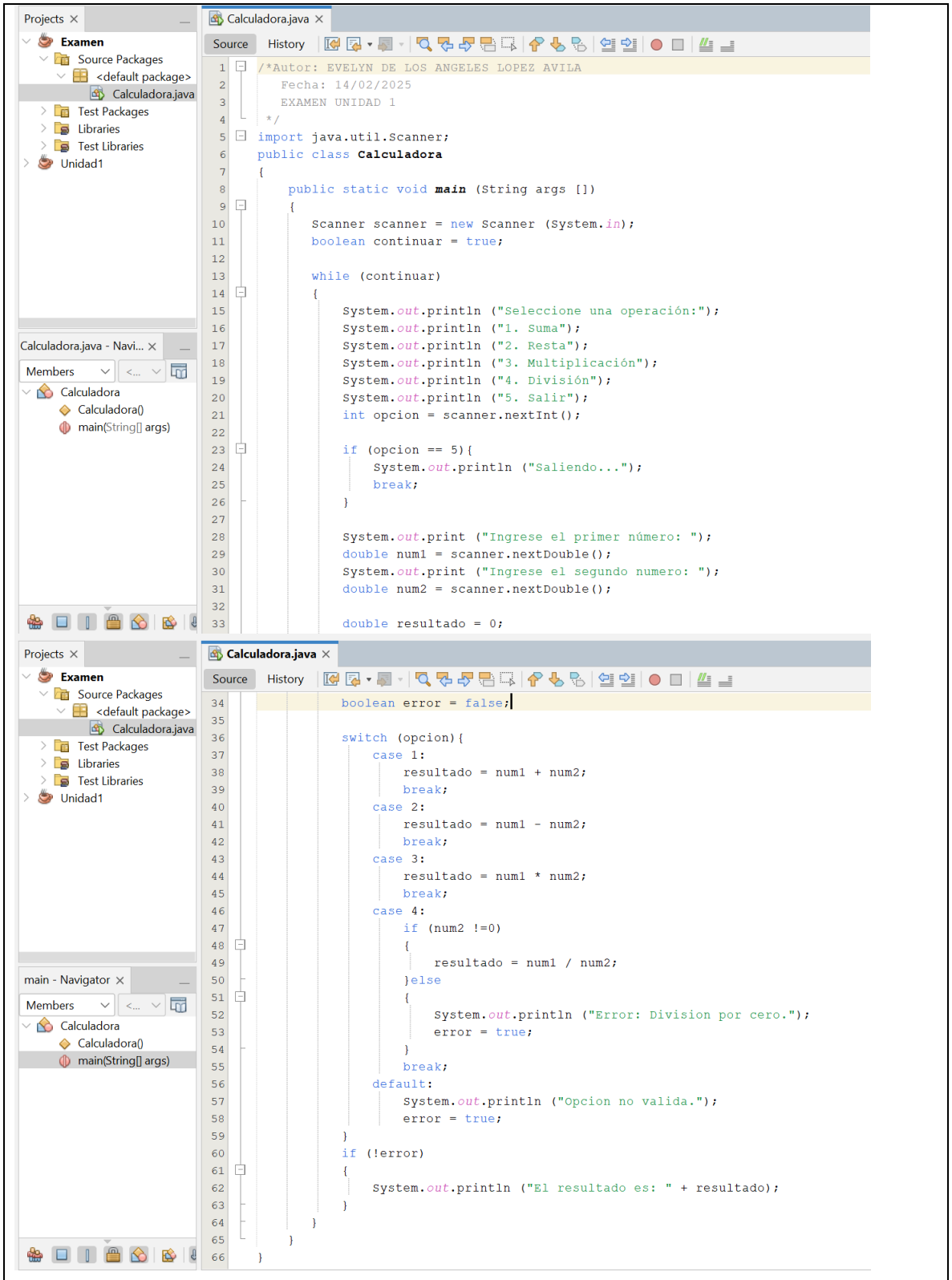
```
60          default:
61              System.out.println ("Opcion no valida.");
62              error = true;
```

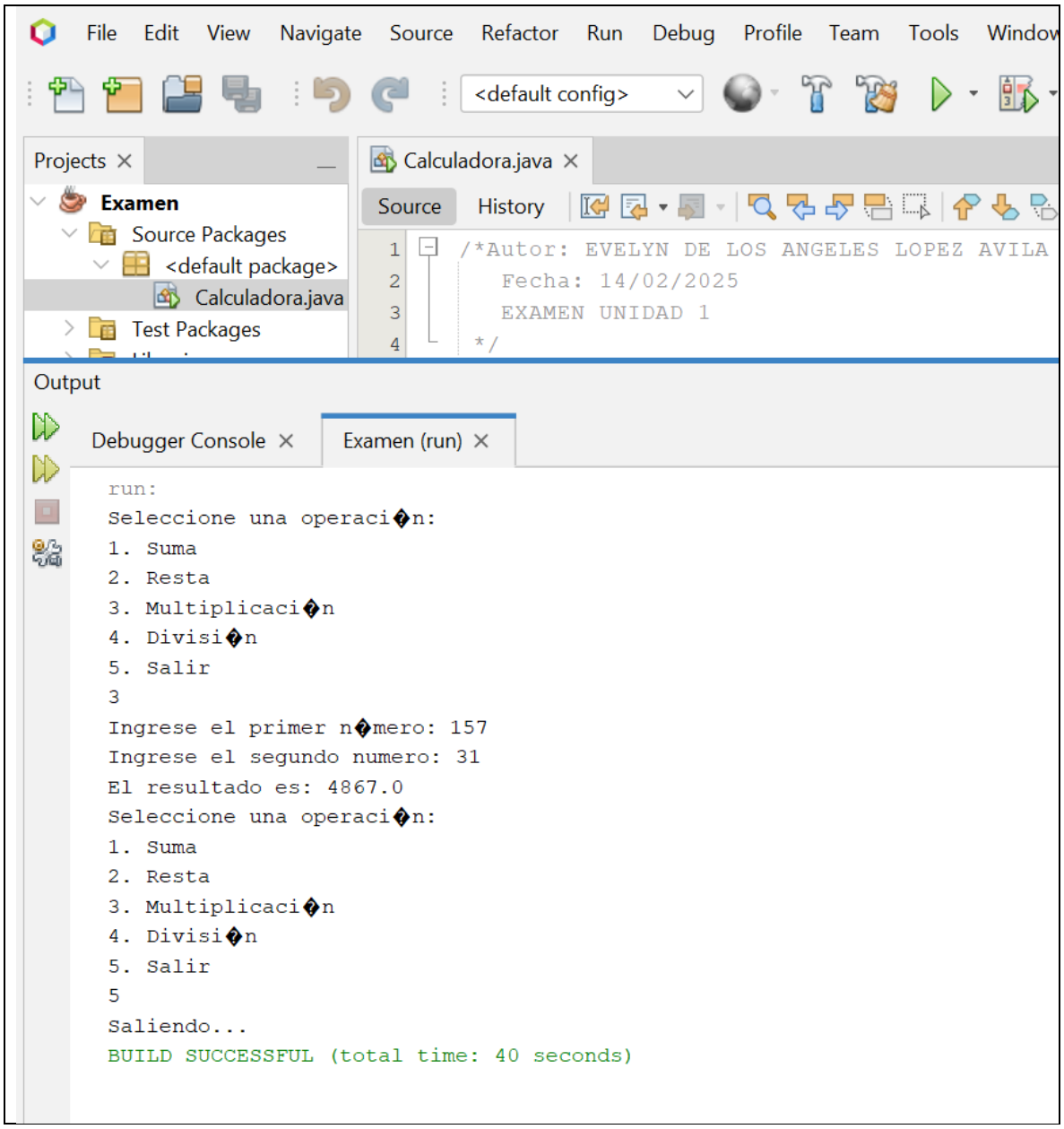
ERROR 7: Falta cerrar comillas.

```
23              if (opcion == 5){
24                  continuar = false;
25                  System.out.println ("Sal
```

```
23              if (opcion == 5){
24                  System.out.println ("Saliendo...");
25                  break;
```

ERROR 8: El valor asignado nunca se ocupa.


Projects
Examen
Source Packages
<default package>
Calculadora.java
Test Packages
Libraries
Test Libraries
Unidad1
Calculadora.java - Navi...
Members
Calculadora
Calculadora()
main(String[] args)
Calculadora.java
Source
History
/*Autor: EVELYN DE LOS ANGELES LOPEZ AVILA
Fecha: 14/02/2025
EXAMEN UNIDAD 1
*/
import java.util.Scanner;
public class Calculadora
{
public static void main (String args [])
{
Scanner scanner = new Scanner (System.in);
boolean continuar = true;
while (continuar)
{
System.out.println ("Seleccione una operación:");
System.out.println ("1. Suma");
System.out.println ("2. Resta");
System.out.println ("3. Multiplicación");
System.out.println ("4. División");
System.out.println ("5. Salir");
int opcion = scanner.nextInt();
if (opcion == 5){
System.out.println ("Saliendo...");
break;
}
System.out.print ("Ingrese el primer número: ");
double num1 = scanner.nextDouble();
System.out.print ("Ingrese el segundo numero: ");
double num2 = scanner.nextDouble();
double resultado = 0;
Projects
Examen
Source Packages
<default package>
Calculadora.java
Test Packages
Libraries
Test Libraries
Unidad1
main - Navigator
Members
Calculadora
Calculadora()
main(String[] args)
Calculadora.java
Source
History
boolean error = false;
switch (opcion){
case 1:
resultado = num1 + num2;
break;
case 2:
resultado = num1 - num2;
break;
case 3:
resultado = num1 * num2;
break;
case 4:
if (num2 !=0)
{
resultado = num1 / num2;
}else
{
System.out.println ("Error: Division por cero.");
error = true;
}
break;
default:
System.out.println ("Opcion no valida.");
error = true;
}
if (!error)
{
System.out.println ("El resultado es: " + resultado);
}
}
}
}


File Edit View Navigate Source Refactor Run Debug Profile Team Tools Windov
<default config>
Projects
Examen
Source Packages
<default package>
Calculadora.java
Test Packages
Calculadora.java
Source History
1 /*Autor: EVELYN DE LOS ANGELES LOPEZ AVILA
2 Fecha: 14/02/2025
3 EXAMEN UNIDAD 1
4 */
Output
Debugger Console
Examen (run)
run:
Seleccione una operaci&#65533;n:
1. Suma
2. Resta
3. Multiplicaci&#65533;n
4. Divisi&#65533;n
5. Salir
3
Ingrese el primer n&#65533;mero: 157
Ingrese el segundo numero: 31
El resultado es: 4867.0
Seleccione una operaci&#65533;n:
1. Suma
2. Resta
3. Multiplicaci&#65533;n
4. Divisi&#65533;n
5. Salir
5
Saliendo...
BUILD SUCCESSFUL (total time: 40 seconds)

Valor = 40

Curso: Programación Orientada en Objetos

Tarea: Examen

Ver todos los envíos

Evelyn de los Angeles López Ávila
evelynlopav16@gmail.com
Fecha de entrega: 15 de febrero de 2...

Cambiar usuario

1 de 29 Reiniciar preferencias de tabla

Página 1 de 6

Entrega

Enviado para calificar
Calificado
La tarea fue enviada 57 mins 13 segundos antes de la fecha límite
Los estudiantes pueden editar este envío

Examen Unidad1.pdf 14 de febrero de 2025, 23:02

Comentarios (0)

Calificación

Calificación sobre 40

30.00

Calificación actual en el libro
30.00

Comentarios de retroalimentación

Errores del código:

ERROR	Corregido (Si/No)
Mayúscula en public	No
Falta nombre de la clase	Si
Falta punto y coma	Si
Falta paréntesis	No
Corregir == en condición opcion = 5	Si

Notificar a estudiantes GUARDAR CAMBIOS GUARDAR Y MOSTRAR SIGUIENTE REINICIAR

INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA

PROGRAMACIÓN ORIENTADA A OBJETOS

INGENIERÍA INFORMÁTICA

EVELYN DE LOS ANGELES LÓPEZ ÁVILA

GRUPO 210 A

ROGELIO ENRIQUE TELONA TORRES

PRÁCTICAS UNIDAD 1

SAN ANDRÉS TUXTLA, VER. A 14 DE FEBRERO DEL 2025

PRÁCTICA 1

Alumna: Evelyn de los Ángeles López Ávila. **Fecha:** 13/02/2025

Actividad: Depuración de Aplicaciones en NetBeans

Objetivo: Aprender a utilizar las herramientas de depuración de NetBeans para identificar y corregir errores en el código.

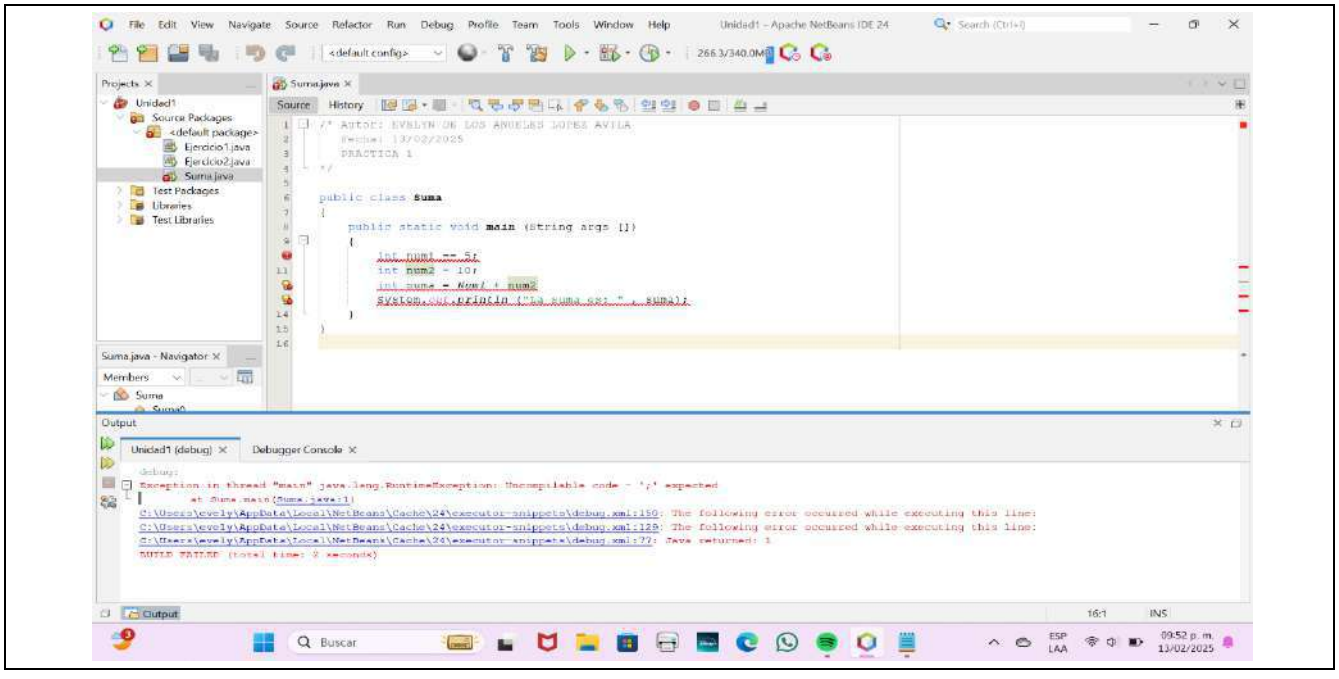
Descripción:

1. Los estudiantes cargarán un proyecto con errores lógicos conocidos y utilizarán las herramientas de depuración para identificar y corregir dichos errores.
2. Cada estudiante deberá depurar el código proporcionado, identificar el error lógico y proponer una solución.
3. Agregue captura del proceso de depuración, de los errores encontrados y de la solución de cada uno de los errores encontrados. De igual manera, después de corregir los errores agregue pantalla de ejecución del código.

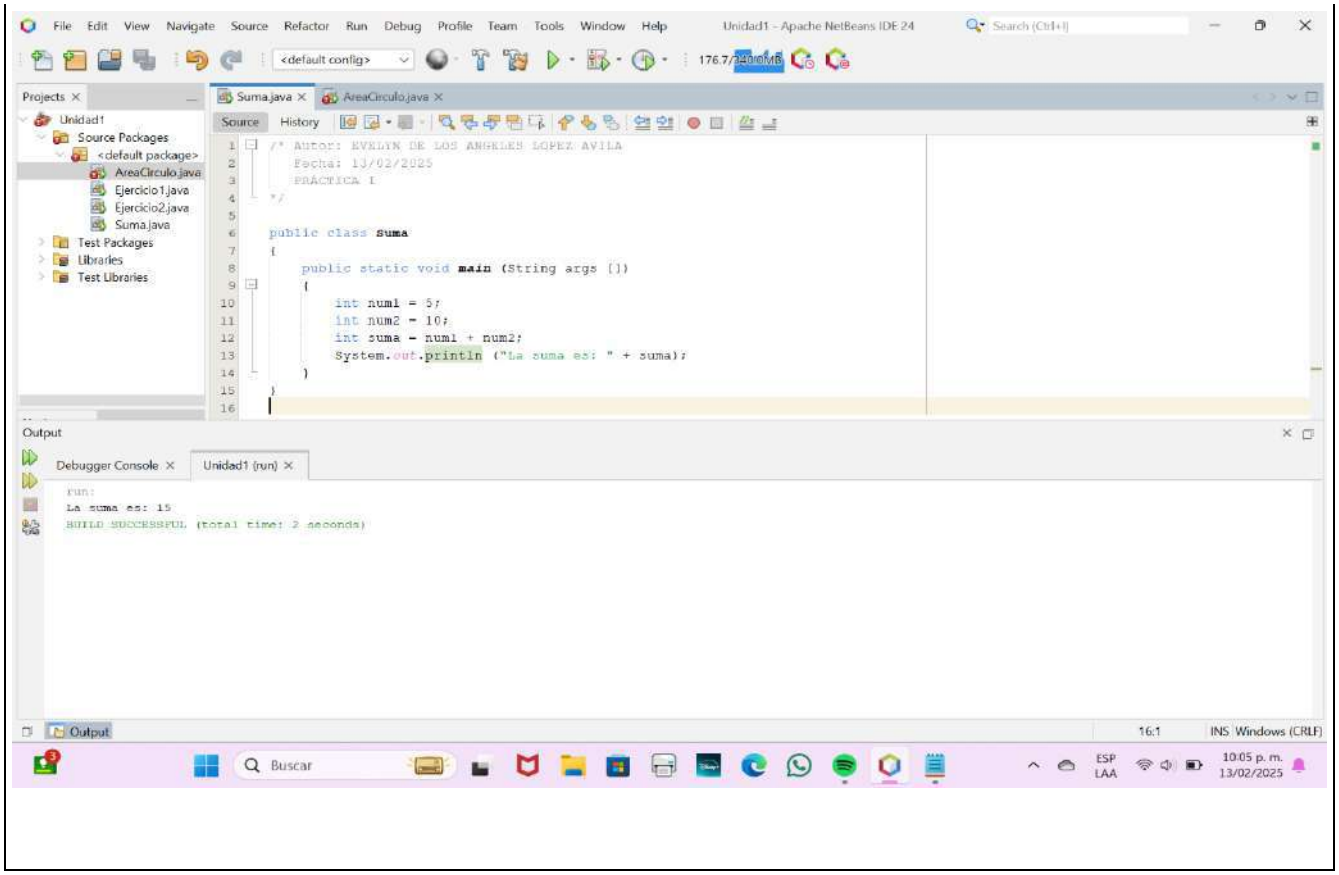
Código:

```java
public class Suma {
    public static void main(String args) {
        int num1 == 5;
        int num2 = 10;
        int suma = Num1 + num2
        System.out.println("La suma es: " , suma);
    }
}
```

Pantalleo:



PRÁCTICA 1



Errores detectados:

1. En el código original no colocan los corchetes “[ ]” en el método main.
2. Utiliza asignación “==” en lugar de igual “=”.
3. Usa una variable no identificada (Coloca Num1 en lugar de num1).
4. Falta punto y coma “;”.
5. Utiliza “,” en vez de “+”.

Soluciones:

1. Colocar los corchetes de la forma correspondiente en el paréntesis: (String args [ ]).
2. Cambiar la asignación por el símbolo de igual.
3. Quitarle la letra mayúscula y colocar “num1”.
4. Colocar punto y coma “;”.
5. Colocarle el símbolo “+” ya que sin este no concatena.

PRÁCTICA 1

Rubrica (Autoevaluación):

Criterio de Evaluación	Sí	No
1. Identificación de Errores		
- El estudiante identificó correctamente todos los errores presentes en el código.		
2. Corrección de Errores		
- El estudiante corrigió adecuadamente todos los errores identificados.		
3. Funcionamiento del Código		
- El código ejecuta correctamente la tarea para la cual fue diseñado.		

PRÁCTICA 2

Alumna: Evelyn de los Ángeles López Ávila. **Fecha:** 13/02/2025

Actividad: Depuración de Aplicaciones en NetBeans

Objetivo: Aprender a utilizar las herramientas de depuración de NetBeans para identificar y corregir errores en el código.

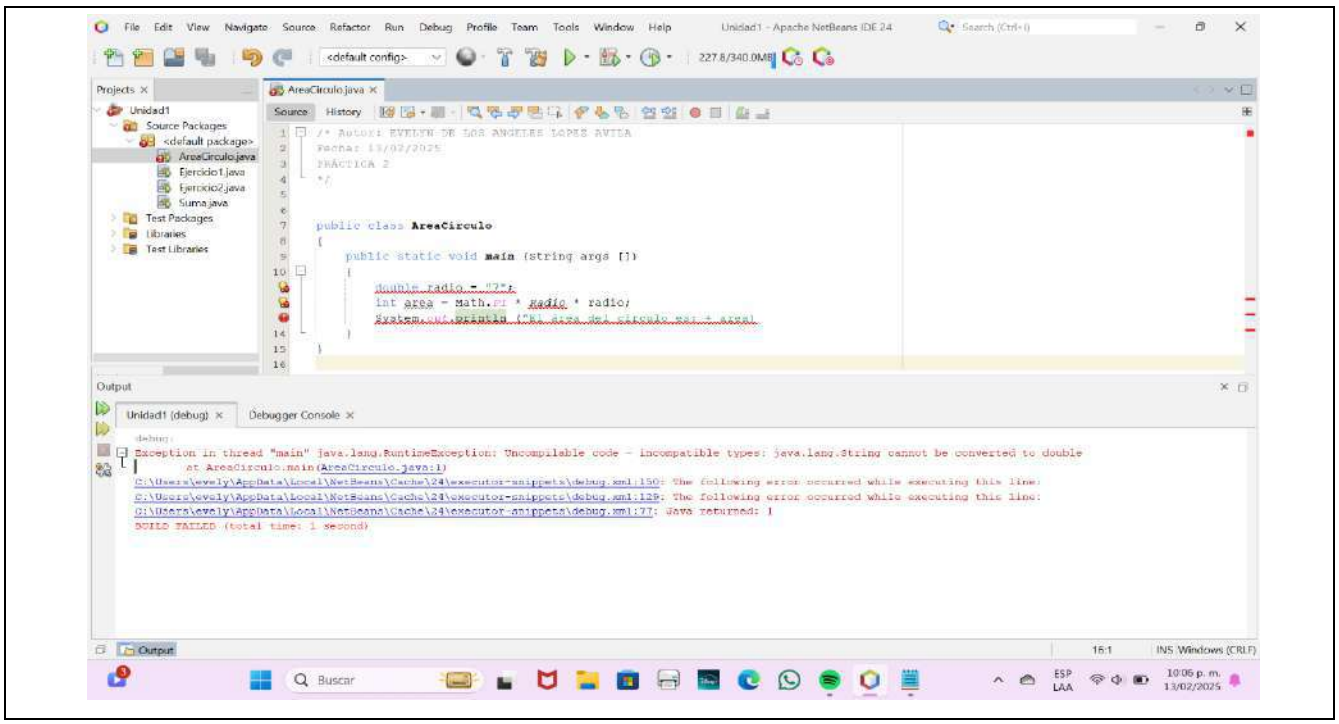
Descripción:

1. Los estudiantes cargarán un proyecto con errores lógicos conocidos y utilizarán las herramientas de depuración para identificar y corregir dichos errores.
2. Cada estudiante deberá depurar el código proporcionado, identificar el error lógico y proponer una solución.
3. Agregue captura del proceso de depuración, de los errores encontrados y de la solución de cada uno de los errores encontrados. De igual manera, después de corregir los errores agregue pantalla de ejecución del código.

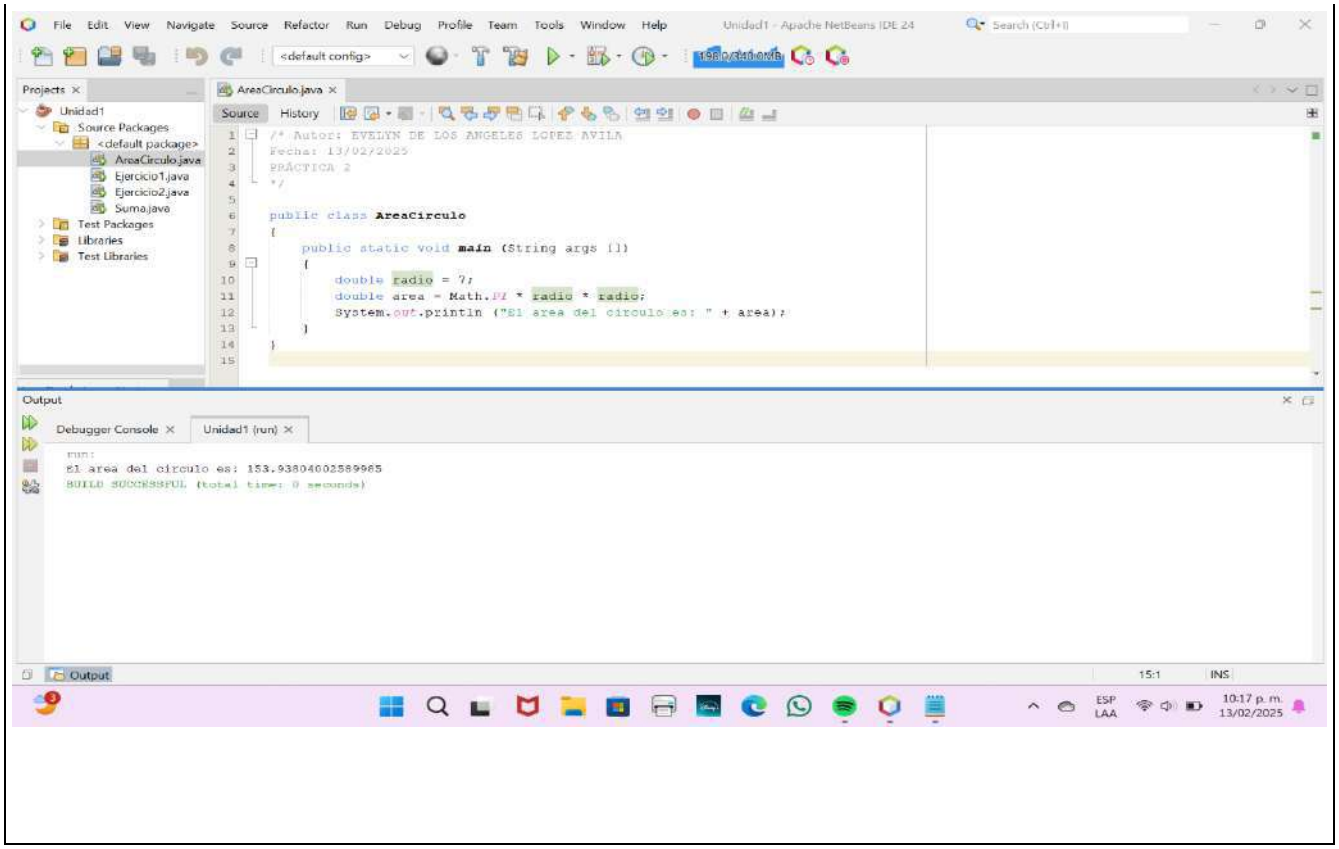
Código:

```
1  public class AreaCirculo {
2      public static void main(String[] args) {
3          double radio = "7";
4          int area = Math.PI * Radio * radio;
5          System.out.println("El área del círculo es:  + area)
6      }
7  }
8
```

Pantalleo:



PRÁCTICA 2



Errores detectados:

1. No corresponder poner el número 7 entre comillas dobles, ya que no es una cadena.
2. No se utiliza int para la variable área ya que no almacena números con decimales.
3. La variable “Radio” no está declarada.
4. Falta cerrar comillas.
5. Falta punto y coma “;”.

Soluciones:

1. Quitar las comillas.
2. Cambiar int y colocar double.
3. Colocar la variable existente: “radio”.
4. Cerrar comillas.
5. Colocar punto y coma.

PRÁCTICA 2

Rubrica (Autoevaluación):

Criterio de Evaluación	Sí	No
1. Identificación de Errores		
- El estudiante identificó correctamente todos los errores presentes en el código.		
2. Corrección de Errores		
- El estudiante corrigió adecuadamente todos los errores identificados.		
3. Funcionamiento del Código		
- El código ejecuta correctamente la tarea para la cual fue diseñado.		

PRÁCTICA 3

Alumna: Evelyn de los Ángeles López Ávila. **Fecha:** 13/02/2025

Actividad: Depuración de Aplicaciones en NetBeans

Objetivo: Aprender a utilizar las herramientas de depuración de NetBeans para identificar y corregir errores en el código.

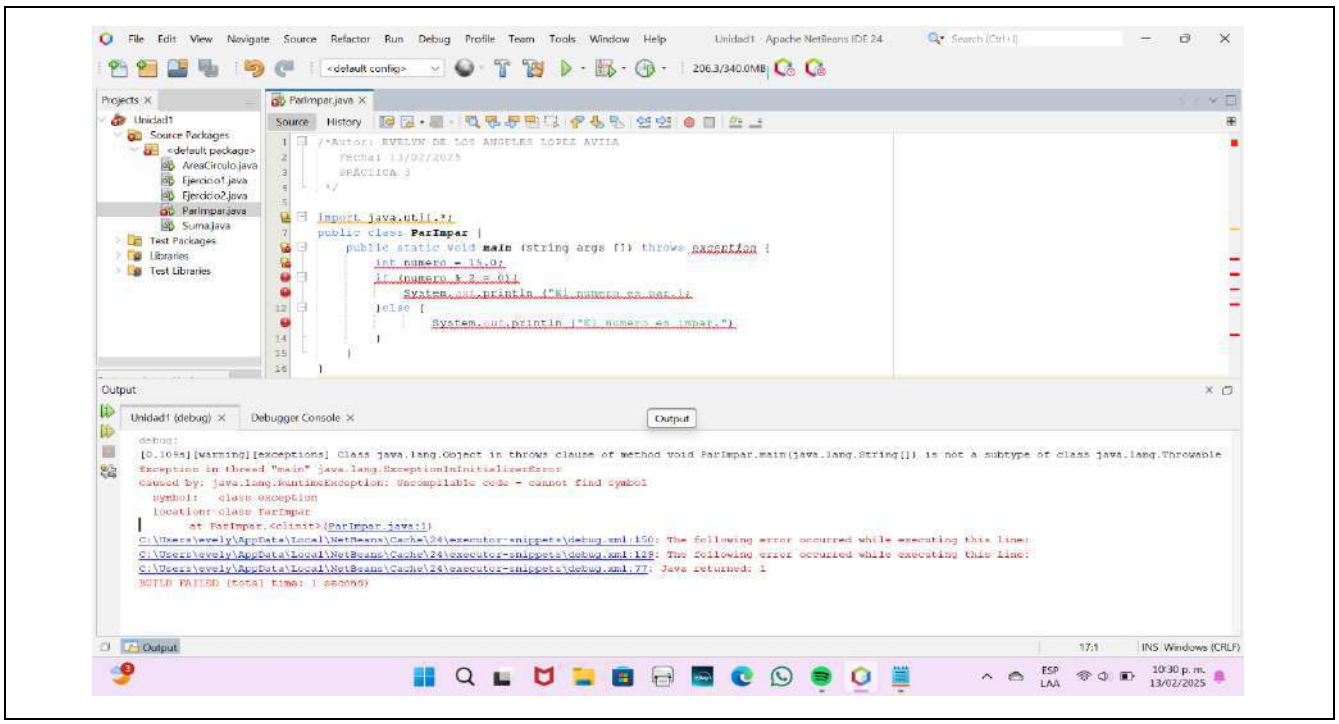
Descripción:

1. Los estudiantes cargarán un proyecto con errores lógicos conocidos y utilizarán las herramientas de depuración para identificar y corregir dichos errores.
2. Cada estudiante deberá depurar el código proporcionado, identificar el error lógico y proponer una solución.
3. Agregue captura del proceso de depuración, de los errores encontrados y de la solución de cada uno de los errores encontrados. De igual manera, después de corregir los errores agregue pantalla de ejecución del código.

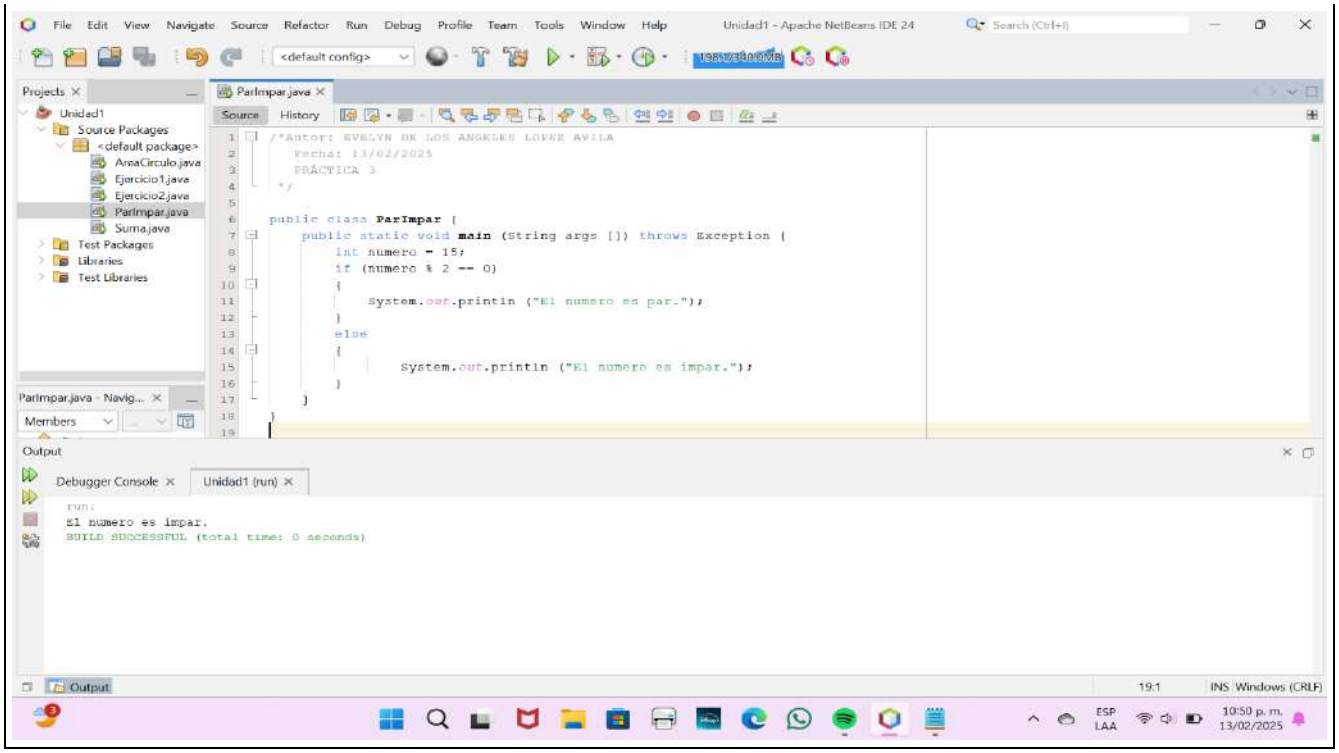
Código:

```java
import java.util.*;

public class ParImpar {
    public static void main(string[] args) throws Exception {
        int numero = 15.0;
        if (numero % 2 = 0) {
            System.out.println("El número es par.);
        } else {
            System.out.println("El número es impar.")
        }
    }
}
```

Pantalleo:



PRÁCTICA 3



Errores detectados:

1. No hace uso de la librería.
2. Error al escribir “throws exception”, ya que la palabara Exception comienza con mayúscula. (Error mío al escribir el código).
3. Un número entero tiene punto decimal innecesariamente.
4. Falta el operador de asignación == en lugar del igual =.
5. Faltan comillas dobles.
6. Falta punto y coma “;”.

Soluciones:

1. Quitar la librería, aunque dejarla tampoco afecta la ejecución del código.
2. Escribir correctamente “throws Exception”.
3. Quitarle el decimal.
4. Colocar el operador de asignación == para poder ejecutar la condición.
5. Colocar las comillas dobles.
6. Colocar punto y coma correspondientes.

PRÁCTICA 3

Rubrica (Autoevaluación):

Criterio de Evaluación	Sí	No
1. Identificación de Errores		
- El estudiante identificó correctamente todos los errores presentes en el código.		
2. Corrección de Errores		
- El estudiante corrigió adecuadamente todos los errores identificados.		
3. Funcionamiento del Código		
- El código ejecuta correctamente la tarea para la cual fue diseñado.		

PRÁCTICA 4

Alumna: Evelyn de los Ángeles López Ávila. **Fecha:** 13/02/2025

Actividad: Depuración de Aplicaciones en NetBeans

Objetivo: Aprender a utilizar las herramientas de depuración de NetBeans para identificar y corregir errores en el código.

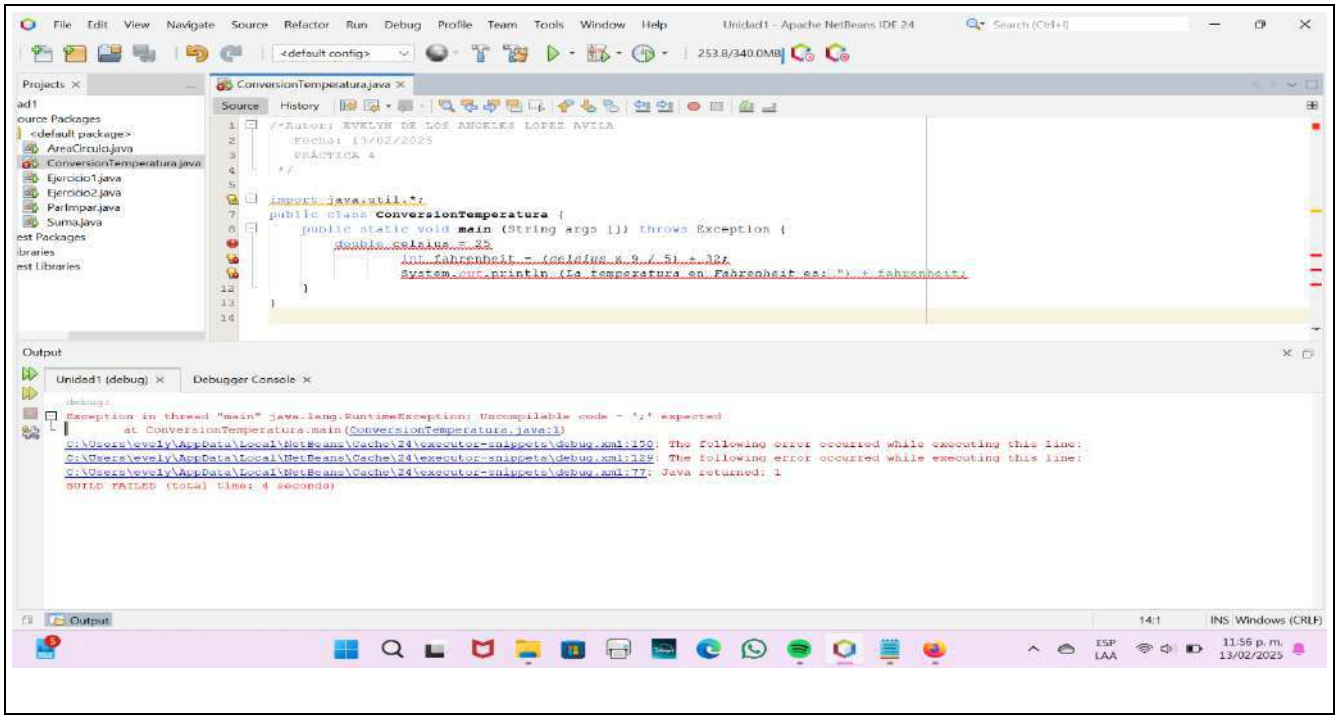
Descripción:

1. Los estudiantes cargarán un proyecto con errores lógicos conocidos y utilizarán las herramientas de depuración para identificar y corregir dichos errores.
2. Cada estudiante deberá depurar el código proporcionado, identificar el error lógico y proponer una solución.
3. Agregue captura del proceso de depuración, de los errores encontrados y de la solución de cada uno de los errores encontrados. De igual manera, después de corregir los errores agregue pantalla de ejecución del código.

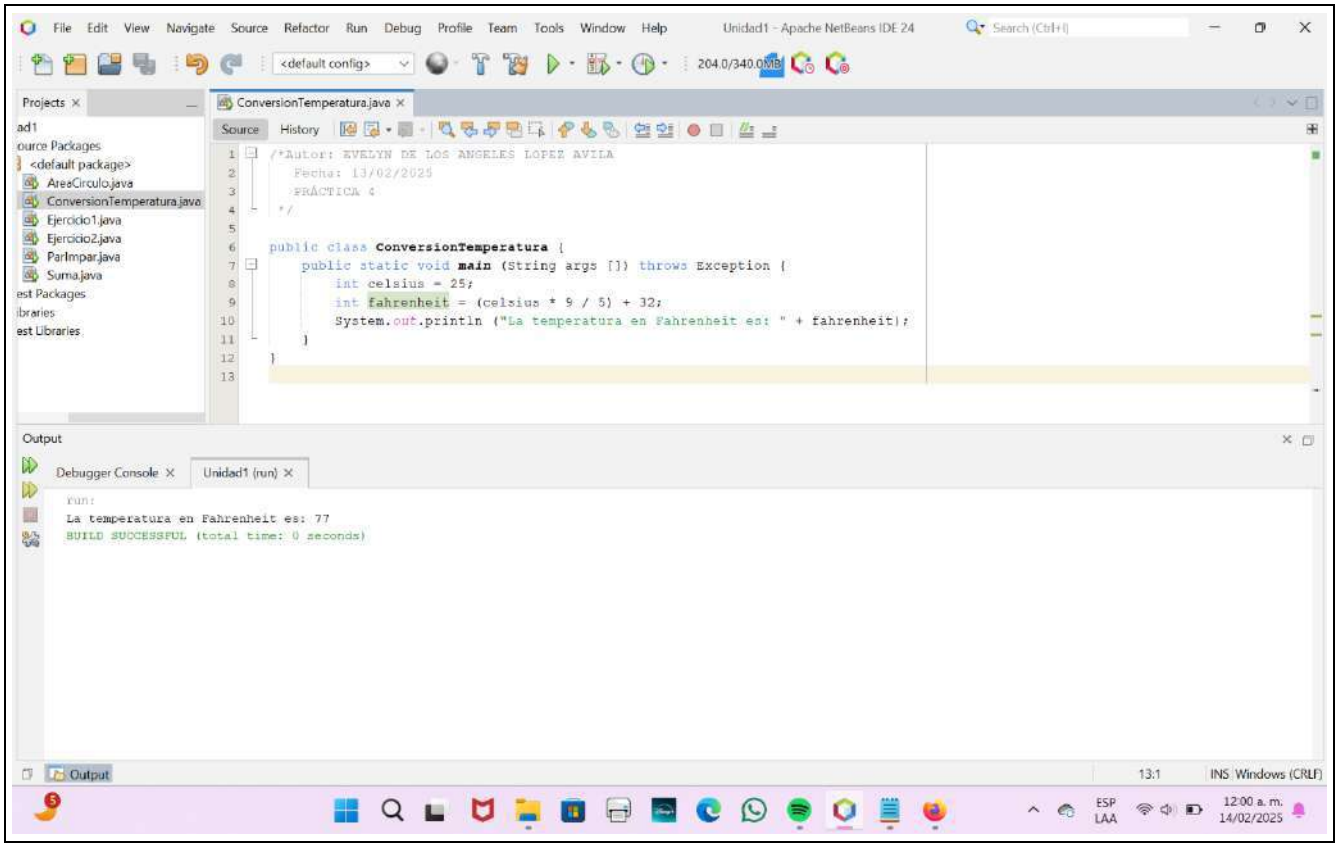
Código:

```java
import java.util.*;

public class ConversionTemperatura {
    public static void man(String[] args) throws Exception {
        double celsius = 25
         int fahrenheit = (celsius x 9 / 5) + 32;
         System.out.println(La temperatura en Fahrenheit es: " + fahrenheit);
    }
}
```

Pantalleo:



PRÁCTICA 4



Errores detectados:

1. No hace uso de la librería.
2. En el código original está escrito “man” en vez de “main”.
3. Utiliza double para un número entero.
4. Falta punto y coma.
5. Utiliza **x** para multiplicar en vez de ***.**
6. Mal colocación de comillas, faltó abrirlas.
7. Mal colocación de paréntesis (error mío de escritura).

Soluciones:

1. Quitar la librería.
2. Escribir correctamente el método main.
3. Usar int en vez de double ya que se trata de un número entero.
4. Agregar punto y coma “;”.
5. Utilizar ***** en vez de **x**, ya que ese es el operador aritmético correcto.

6. Colocar bien las comillas (abrirlas)
7. Colocar correctamente el paréntesis final.

Rubrica (Autoevaluación):

Criterio de Evaluación	**Sí**	**No**
1. Identificación de Errores		
- El estudiante identificó correctamente todos los errores presentes en el código.		
2. Corrección de Errores		
- El estudiante corrigió adecuadamente todos los errores identificados.		
3. Funcionamiento del Código		
- El código ejecuta correctamente la tarea para la cual fue diseñado.		

PRÁCTICA 5

Alumna: Evelyn de los Ángeles López Ávila. **Fecha:** 13/02/2025

Actividad: Depuración de Aplicaciones en NetBeans

Objetivo: Aprender a utilizar las herramientas de depuración de NetBeans para identificar y corregir errores en el código.

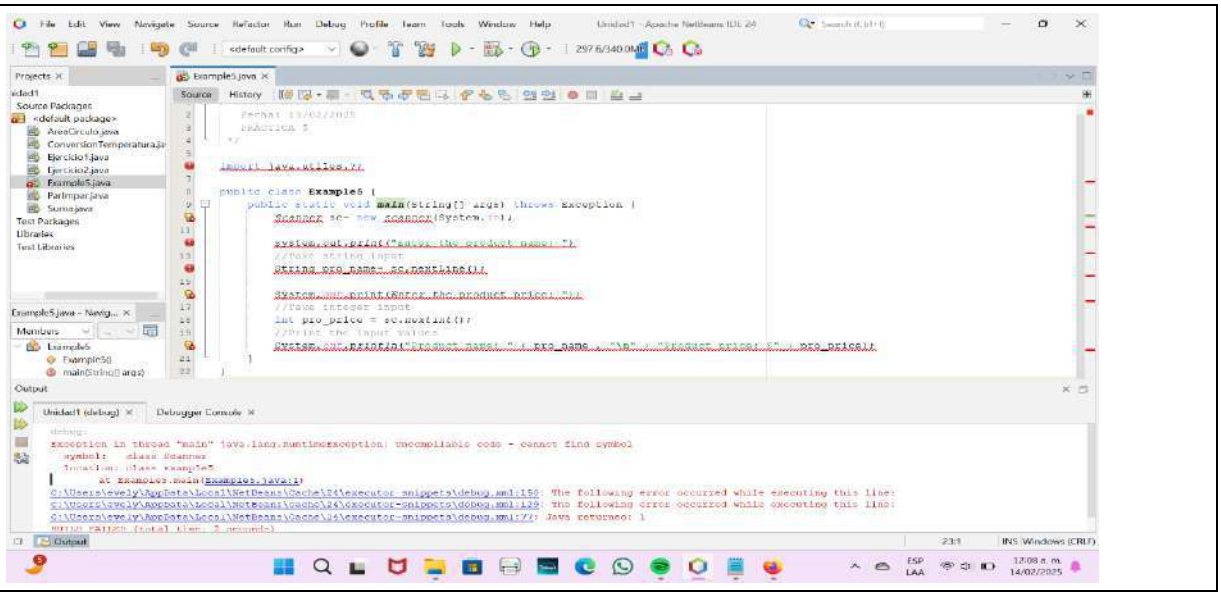
Descripción:

1. Los estudiantes cargarán un proyecto con errores lógicos conocidos y utilizarán las herramientas de depuración para identificar y corregir dichos errores.
2. Cada estudiante deberá depurar el código proporcionado, identificar el error lógico y proponer una solución.
3. Agregue captura del proceso de depuración, de los errores encontrados y de la solución de cada uno de los errores encontrados. De igual manera, después de corregir los errores agregue pantalla de ejecución del código.

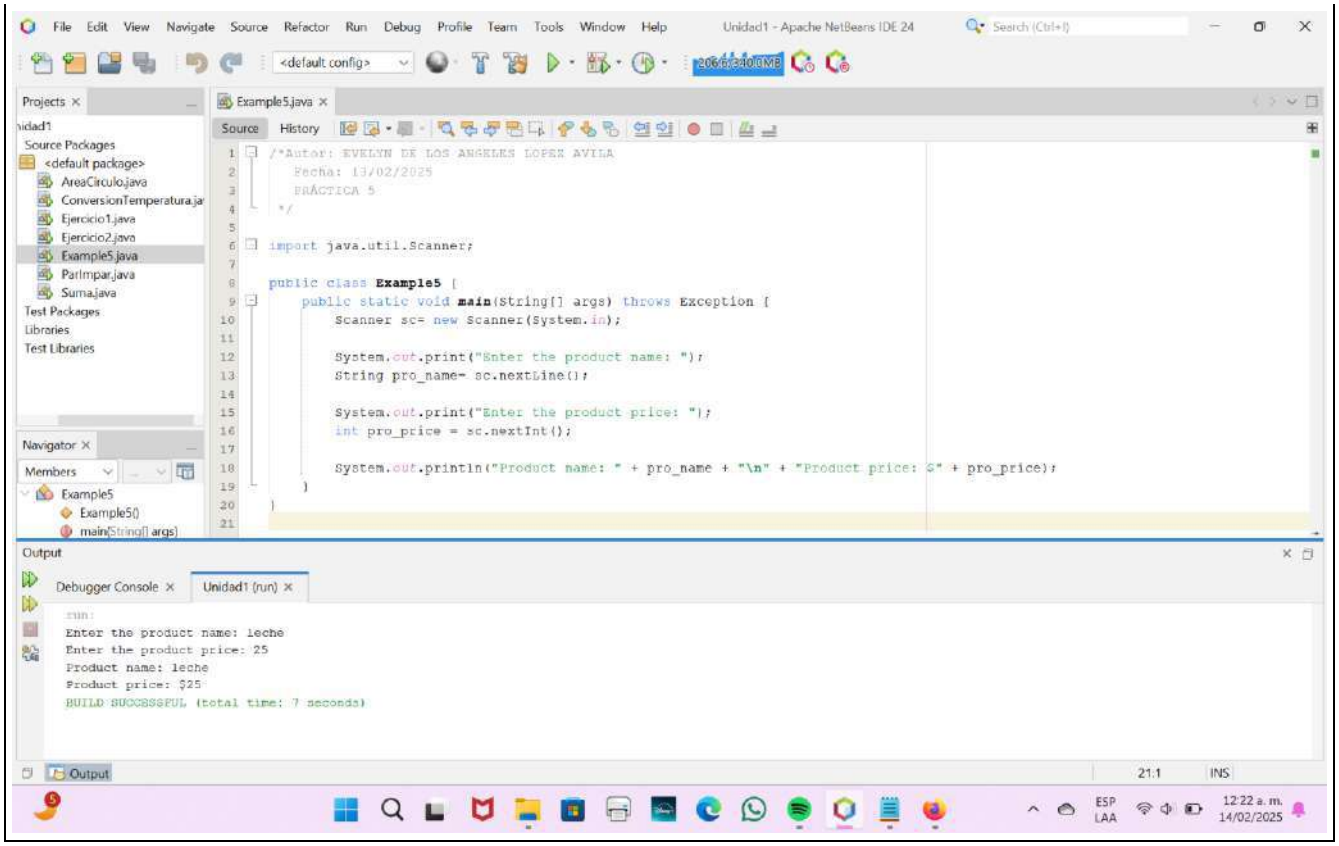
Código:

```java
import java.utiles.?;

public class Example5 {
    public static void man(String[] args) throws Exception {
        Scanner sc= new scanner(System.in);

        syStem.out.print("Enter the product name: ")
        //Take string input
        String pro_name= sc,nextLine();

        System.out.print(Enter the product price: ");
        //Take integer input
        int pro_price = sc.nextInt();
        //Print the input values
        System.out.println("Product name: " + pro_name , "\n" + "Product price: $" + pro_price);
    }
}
```

Pantalleo:



PRÁCTICA 5



Errores detectados:

1. Errores ortográficos en la escritura de la librería, coloca la palabra “utiles" en vez de “util", además de poner un signo de interrogación.
2. El código original escribe mal el método main, ya que coloca la palabra “men” en vez de “main”.
3. Scanner está escrito sin la S mayúscula.
4. La palabra System está mal escrita.
5. Falta punto y coma.
6. Coloca una coma en vez de punto en “sc.nextLine”.
7. Mal colocación de comillas dobles.
8. Uso de coma en vez de símbolo de suma lo que provoca que no concatene.

Soluciones:

1. Escribir de manera correcta la librería: import java.util.Scanner;
2. Corregir la palabra y escribir “main”.

3. Colocarle la “S” mayúscula a la palbara Scanner.
4. Escribir correctamente System
5. Colocar punto y coma “;”.
6. Cambiar la coma por un punto: sc.nextLine
7. Colocar de forma correcta las comillas dobles.
8. Utilizar el símbolo de suma +

Rubrica (Autoevaluación):

Criterio de Evaluación	**Sí**	**No**
1. Identificación de Errores		
- El estudiante identificó correctamente todos los errores presentes en el código.		
2. Corrección de Errores		
- El estudiante corrigió adecuadamente todos los errores identificados.		
3. Funcionamiento del Código		
- El código ejecuta correctamente la tarea para la cual fue diseñado.		

Curso: Programación Orientada en Objetos

Tarea: Practicas

Ver todos los envíos

Valor = 60

Evelyn de los Angeles López Ávila
evelynlopav16@gmail.com
Fecha de entrega: 14 de febrero de 2...

Cambiar usuario

1 de 30 Reiniciar preferencias de tabla

Página 1 de 16

Entrega

Enviado para calificar
Calificado
La tarea fue enviada 8 horas 1 min antes de la fecha límite
Los estudiantes pueden editar este envío

Prácticas1.pdf 14 de febrero de 2025, 09:58

Comentarios (0)

Calificación

Calificación sobre 60

60.00

Calificación actual en el libro
60.00

Comentarios de retroalimentación

¡Excelenteeeeee!

Quizá lo único que podría decir que falta, es indicar el número de línea de código.

Notificar a estudiantes GUARDAR CAMBIOS GUARDAR Y MOSTRAR SIGUIENTE REINICIAR