Facilitador(a): Emilio Batista

Asignatura: **Herramientas de la Programación Aplicada II**

* Estudiante: Terry He | 8-1021-2180
* Jhuomar Barría | 9-766-196
* Geremi Tejeira | 9-768-42

Grupo: 1IL128

Fecha: 10/5/2025

1. **TÍTULO** **DE LA EXPERIENCIA**: Guía de Laboratorio Tarea: **Creación de los Métodos**, y programa principal, incluir las librerías.
2. **TEMAS:**

Creación de Métodos en java.

1. **OBJETIVO(S):**

Conocer la forma de creación de métodos, en este caso crearemos para sumar, restar, multiplicar, dividir

1. **METODOLOGÍA SUGERIDA:**
2. Se trabajará de manera individual o grupo de tres
3. La actividad debe ser desarrollada durante la sesión de clases o en casa
4. Presentar el análisis y diseño de los problemas. (uso IDE y codificación de su programa)
5. La solución de los problemas debe ser en codificación java
6. Anexar la prueba de corrida del programa para validar la solución, y captura de las pantallas
7. Para ejecutar la solución, utilizar el lenguaje Java.
8. La entrega de la solución se hará mediante plataforma Moodle/Dropbox/OneDrive, Teams
9. **ENUNCIADO:**

Confeccionar un programa en java que realice 6 métodos, para sumar, restar, multiplicar, leer los números, Dividir dos números (verificar la división x cero A/B), e imprimir los resultados, debe incluir títulos del reporte, nombre completo, cedula, numero de grupo y asignatura. Revisar el formato en la última página para referencia de la confección del programa.

1. **PROCEDIMIENTO:**

Indicar todos los pasos necesarios para realizar la experiencia. Se pueden mostrar con esquemas, dibujos, imágenes, fotos, entre otros.

* 1. **Crear el o los programas o nombre del proyecto**
  2. **Crear la clase**
  3. **Captura del código fuente, captura de las pantallas de la corrida**
  4. **Codificación en Java, prueba del programa**

1. **RECURSOS:**

*Guía de actividad, apuntes de clases, aplicación.es de Celular Java Nide (apps).*

1. **RESULTADOS (OPCIONAL, DE ACUERDO CON LA ACTIVIDAD):**

Sección donde se colocan los datos obtenidos con la experiencia. Se formatea de acuerdo con la salida deseada: conjunto de preguntas, tablas, gráficos, preguntas abiertas, entre otros.

|  |  |  |
| --- | --- | --- |
| ***Análisis y Diseño*** | | |
| El propósito de este proyecto es desarrollar una aplicación en lenguaje Java, utilizando el IDE Visual Studio Code, que implemente una serie de métodos para realizar operaciones aritméticas básicas. El programa debe estar compuesto por seis métodos:   1. Leer dos números desde entrada. 2. Sumar los dos números. 3. Restar los dos números. 4. Multiplicar los dos números. 5. Dividir dos números, incluyendo verificación de división por cero. 6. Imprimir los resultados.   El uso de métodos permite organizar el código de forma modular, facilitando su lectura, mantenimiento y reutilización. Además, esta estructura favorece el aprendizaje de los principios fundamentales de la programación orientada a objetos. | | |
| *Entradas* | *Proceso* | *Salida* |
| Crear el programa principal, y la clase llamada OperacionMatematica, que debe contener 5 métodos sumar, restar, leer números, multiplicar, dividir | Obtener los valores de la sumar, resta, producto, división, lectura e impresión de los resultados | Imprimir los resultados, el reporte debe llevar su nombre, cedula, Nombre de la facultad, Universidad, numero grupo fecha y año. |
| ***Escribir en esta sección descripción de las variables usadas en su programa.*** | | |
| **En la clase CalculadoraGUI:**   1. ***Variables de instancia (privadas):***  * *campoA: JTextField para el primer número* * *campoB: JTextField para el segundo número* * *resultadoLabel: JLabel para mostrar el resultado*  1. ***Variables locales en el método calcular:***  * *a: int - primer número ingresado* * *b: int - segundo número ingresado* * *resultado: String - almacena el resultado de la operación* * *operacion: String - parámetro que indica qué operación realizar* * *ex: NumberFormatException - variable para manejar errores de formato*   **En la clase OperacionMatematica:**   1. ***Variables de método (parámetros):***  * *a: int - primer número para las operaciones* * *b: int - segundo número para las operaciones*   **En la clase Main:**   * *No hay variables declaradas, solo usa el método main con el parámetro args (String[])* | | |

|  |
| --- |
| ***Prueba de Escritorio: escriba aquí captura de las pantallas*** |
| ***Prueba de Escritorio***   1. ***Inicio del Programa***    * *Se inicia la aplicación*    * *Se muestra la interfaz gráfica con:*       + *Campo para primer número*      + *Campo para segundo número*      + *Botones: Sumar, Restar, Multiplicar, Dividir*      + *Etiqueta para mostrar resultado* 2. ***Caso de Prueba 1: Suma***   *Texto  El contenido generado por IA puede ser incorrecto.*   1. ***Caso de Prueba 2: Resta***   *Texto  El contenido generado por IA puede ser incorrecto.*   1. ***Caso de Prueba 3: Multiplicación***   *Texto  El contenido generado por IA puede ser incorrecto.*   1. ***Caso de Prueba 4: División***   *Texto  El contenido generado por IA puede ser incorrecto.*   1. ***Caso de Prueba 5: División por Cero***   *Texto  El contenido generado por IA puede ser incorrecto.*   1. ***Caso de Prueba 6: Entrada Inválida***   *Texto  El contenido generado por IA puede ser incorrecto.* |

|  |
| --- |
| ***Codificación en Lenguaje Java*** |
| **Main.java**  *public class Main {*  *public static void main(String[] args) {*  *javax.swing.SwingUtilities.invokeLater(() -> {*  *new CalculadoraGUI().setVisible(true);*  *});*  *}*  *}* |
| **CalculadoraGUI.java**  import javax.swing.\*;  import java.awt.\*;  import java.awt.event.\*;  public class CalculadoraGUI extends JFrame {      private JTextField campoA, campoB;      private JLabel resultadoLabel;      public CalculadoraGUI() {          setTitle("Calculadora");          setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);          setSize(480, 520);          setLocationRelativeTo(null);          getContentPane().setBackground(new Color(18, 22, 34)); // Fondo general oscuro          // Panel central con fondo y bordes redondeados          JPanel panelCentral = new JPanel() {              @Override              protected void paintComponent(Graphics g) {                  super.paintComponent(g);                  Graphics2D g2 = (Graphics2D) g;                  g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);                  g2.setColor(new Color(32, 36, 48));                  g2.fillRoundRect(0, 0, getWidth(), getHeight(), 32, 32);              }          };          panelCentral.setOpaque(false);          panelCentral.setLayout(new BoxLayout(panelCentral, BoxLayout.Y\_AXIS));          panelCentral.setBorder(BorderFactory.createEmptyBorder(0, 0, 0, 0));          // Encabezado morado degradado          JPanel header = new JPanel() {              @Override              protected void paintComponent(Graphics g) {                  super.paintComponent(g);                  Graphics2D g2 = (Graphics2D) g;                  g2.setRenderingHint(RenderingHints.KEY\_ANTIALIASING, RenderingHints.VALUE\_ANTIALIAS\_ON);                  GradientPaint gp = new GradientPaint(0, 0, new Color(168, 85, 247), 0, getHeight(),                          new Color(232, 121, 249));                  g2.setPaint(gp);                  g2.fillRoundRect(0, 0, getWidth(), getHeight(), 32, 32);              }          };          header.setPreferredSize(new Dimension(400, 80));          header.setMaximumSize(new Dimension(Integer.MAX\_VALUE, 80));          header.setOpaque(false);          header.setLayout(new BorderLayout());          JLabel titulo = new JLabel("Calculadora", SwingConstants.CENTER);          titulo.setFont(new Font("Arial", Font.BOLD, 30));          titulo.setForeground(Color.WHITE);          titulo.setBorder(BorderFactory.createEmptyBorder(18, 0, 0, 0));          header.add(titulo, BorderLayout.CENTER);          panelCentral.add(header);          panelCentral.add(Box.createRigidArea(new Dimension(0, 20)));          // Primer número          JLabel labelA = new JLabel("Primer número");          labelA.setFont(new Font("Arial", Font.BOLD, 16));          labelA.setForeground(new Color(232, 232, 255));          labelA.setAlignmentX(Component.LEFT\_ALIGNMENT);          campoA = new JTextField();          campoA.setFont(new Font("Arial", Font.PLAIN, 16));          campoA.setMaximumSize(new Dimension(Integer.MAX\_VALUE, 40));          campoA.setBackground(new Color(36, 41, 54));          campoA.setForeground(new Color(200, 200, 210));          campoA.setCaretColor(new Color(200, 200, 210));          campoA.setBorder(BorderFactory.createCompoundBorder(                  BorderFactory.createLineBorder(new Color(60, 60, 80), 1, true),                  BorderFactory.createEmptyBorder(8, 12, 8, 12)));          campoA.setToolTipText("Ingrese el primer número");          // Segundo número          JLabel labelB = new JLabel("Segundo número");          labelB.setFont(new Font("Arial", Font.BOLD, 16));          labelB.setForeground(new Color(232, 232, 255));          labelB.setAlignmentX(Component.LEFT\_ALIGNMENT);          campoB = new JTextField();          campoB.setFont(new Font("Arial", Font.PLAIN, 16));          campoB.setMaximumSize(new Dimension(Integer.MAX\_VALUE, 40));          campoB.setBackground(new Color(36, 41, 54));          campoB.setForeground(new Color(200, 200, 210));          campoB.setCaretColor(new Color(200, 200, 210));          campoB.setBorder(BorderFactory.createCompoundBorder(                  BorderFactory.createLineBorder(new Color(60, 60, 80), 1, true),                  BorderFactory.createEmptyBorder(8, 12, 8, 12)));          campoB.setToolTipText("Ingrese el segundo número");          panelCentral.add(Box.createRigidArea(new Dimension(0, 10)));          panelCentral.add(labelA);          panelCentral.add(campoA);          panelCentral.add(Box.createRigidArea(new Dimension(0, 18)));          panelCentral.add(labelB);          panelCentral.add(campoB);          panelCentral.add(Box.createRigidArea(new Dimension(0, 28)));          // Panel de botones          JPanel panelBotones = new JPanel(new GridLayout(2, 2, 18, 18));          panelBotones.setOpaque(false);          JButton btnSumar = new JButton("Sumar");          btnSumar.setBackground(new Color(139, 92, 246));          btnSumar.setForeground(Color.WHITE);          btnSumar.setFont(new Font("Arial", Font.BOLD, 18));          btnSumar.setFocusPainted(false);          btnSumar.setBorder(BorderFactory.createLineBorder(new Color(139, 92, 246), 1, true));          JButton btnRestar = new JButton("Restar");          btnRestar.setBackground(new Color(168, 85, 247));          btnRestar.setForeground(Color.WHITE);          btnRestar.setFont(new Font("Arial", Font.BOLD, 18));          btnRestar.setFocusPainted(false);          btnRestar.setBorder(BorderFactory.createLineBorder(new Color(168, 85, 247), 1, true));          JButton btnMultiplicar = new JButton("Multiplicar");          btnMultiplicar.setBackground(new Color(219, 39, 119));          btnMultiplicar.setForeground(Color.WHITE);          btnMultiplicar.setFont(new Font("Arial", Font.BOLD, 18));          btnMultiplicar.setFocusPainted(false);          btnMultiplicar.setBorder(BorderFactory.createLineBorder(new Color(219, 39, 119), 1, true));          JButton btnDividir = new JButton("Dividir");          btnDividir.setBackground(new Color(139, 92, 246));          btnDividir.setForeground(Color.WHITE);          btnDividir.setFont(new Font("Arial", Font.BOLD, 18));          btnDividir.setFocusPainted(false);          btnDividir.setBorder(BorderFactory.createLineBorder(new Color(139, 92, 246), 1, true));          btnSumar.setCursor(Cursor.getPredefinedCursor(Cursor.HAND\_CURSOR));          btnRestar.setCursor(Cursor.getPredefinedCursor(Cursor.HAND\_CURSOR));          btnMultiplicar.setCursor(Cursor.getPredefinedCursor(Cursor.HAND\_CURSOR));          btnDividir.setCursor(Cursor.getPredefinedCursor(Cursor.HAND\_CURSOR));          panelBotones.add(btnSumar);          panelBotones.add(btnRestar);          panelBotones.add(btnMultiplicar);          panelBotones.add(btnDividir);          panelCentral.add(panelBotones);          panelCentral.add(Box.createRigidArea(new Dimension(0, 20)));          // Resultado          resultadoLabel = new JLabel(" ", SwingConstants.CENTER);          resultadoLabel.setFont(new Font("Arial", Font.BOLD, 18));          resultadoLabel.setForeground(new Color(232, 121, 249));          resultadoLabel.setAlignmentX(Component.CENTER\_ALIGNMENT);          panelCentral.add(resultadoLabel);          // Acciones de los botones          btnSumar.addActionListener(e -> calcular("sumar"));          btnRestar.addActionListener(e -> calcular("restar"));          btnMultiplicar.addActionListener(e -> calcular("multiplicar"));          btnDividir.addActionListener(e -> calcular("dividir"));          // Layout principal          setLayout(new GridBagLayout());          add(panelCentral);      }      private void calcular(String operacion) {          try {              int a = Integer.parseInt(campoA.getText());              int b = Integer.parseInt(campoB.getText());              String resultado = "";              switch (operacion) {                  case "sumar":                      resultado = "Resultado: " + OperacionMatematica.sumar(a, b);                      break;                  case "restar":                      resultado = "Resultado: " + OperacionMatematica.restar(a, b);                      break;                  case "multiplicar":                      resultado = "Resultado: " + OperacionMatematica.multiplicar(a, b);                      break;                  case "dividir":                      resultado = "Resultado: " + OperacionMatematica.dividir(a, b);                      break;              }              resultadoLabel.setText(resultado);          } catch (NumberFormatException ex) {              resultadoLabel.setText("Por favor, ingrese números válidos.");          }      }  } |
| **OperacionMatematica.java**  public class OperacionMatematica {      public static int sumar(int a, int b) {          return a + b;      }      public static int restar(int a, int b) {          return a - b;      }      public static int multiplicar(int a, int b) {          return a \* b;      }      public static String dividir(int a, int b) {          if (b == 0) {              return "Error: división por cero";          } else {              return String.format("%.2f", (double) a / b);          }      }  } |

1. **BIBLIOGRAFIA:**

***Documentación Oficial:***

1. *Oracle Java Documentation*

* *https://docs.oracle.com/en/java/*
* *Especialmente útil para:*
* *Swing GUI Framework*
* *Manejo de excepciones*
* *Operaciones matemáticas básicas*

***Tutoriales y Cursos:***

1. *Java Swing Tutorial - Oracle*

* *https://docs.oracle.com/javase/tutorial/uiswing/*
* *Excelente para aprender sobre la interfaz gráfica*

1. *W3Schools Java Tutorial*

* *https://www.w3schools.com/java/*
* *Bueno para conceptos básicos de Java*

***Stack Overflow:***

1. *Preguntas relevantes sobre Swing:*

* *https://stackoverflow.com/questions/tagged/java-swing*
* *Ejemplos de implementación de calculadoras*
* *Soluciones a problemas comunes de GUI*

1. *Preguntas sobre manejo de excepciones:*

* *https://stackoverflow.com/questions/tagged/java+exceptions*
* *Cómo manejar errores en operaciones matemáticas*

***GitHub:***

1. *Proyectos de ejemplo de calculadoras:*

* [*https://github.com/topics/java-calculator*](https://github.com/topics/java-calculator)
* [*https://github.com/topics/java-swing-calculator*](https://github.com/topics/java-swing-calculator)
* *Proyectos similares que pueden servir como referencia*

***Videos Tutoriales:***

1. *YouTube:*

* *"Java Swing GUI Tutorial" por Bro Code*
* *"How to Create a Calculator in Java" por ProgrammingKnowledge*
* *"Java GUI Tutorial" por freeCodeCamp*

***Páginas Web:***

1. *GeeksforGeeks*

* [*https://www.geeksforgeeks.org/java-swing-tutorial/*](https://www.geeksforgeeks.org/java-swing-tutorial/)
* *Artículos sobre implementación de GUI en Java*

1. *JavaTpoint*

* [*https://www.javatpoint.com/java-swing*](https://www.javatpoint.com/java-swing)
* *Tutoriales detallados sobre Swing*

***Recursos para Diseño UI/UX:***

1. *Material Design Guidelines*

* [*https://material.io/design*](https://material.io/design)
* *Para mejorar la apariencia de la interfaz*

1. *Color Theory Resources*

* [*https://colorhunt.co/*](https://colorhunt.co/)
* *Para la selección de colores en la interfaz*

***Herramientas de Desarrollo:***

1. *Maven Documentation*

* [*https://maven.apache.org/guides/*](https://maven.apache.org/guides/)
* *Para la gestión de dependencias y construcción del proyecto*

1. *IntelliJ IDEA Documentation*

* [*https://www.jetbrains.com/idea/documentation/*](https://www.jetbrains.com/idea/documentation/)
* *Si se usa este IDE*

***Foros y Comunidades:***

1. *Reddit r/javahelp*

* [*https://www.reddit.com/r/javahelp/*](https://www.reddit.com/r/javahelp/)
* *Comunidad activa para resolver dudas*

1. *Java Forum*

* [*https://coderanch.com/*](https://coderanch.com/)
* *Foro especializado en Java*

***Libros:***

1. *"Head First Java" por Kathy Sierra*

* *Excelente para principiantes*

1. *"Java Swing" por Marc Loy*

* *Específico para desarrollo de interfaces gráficas*

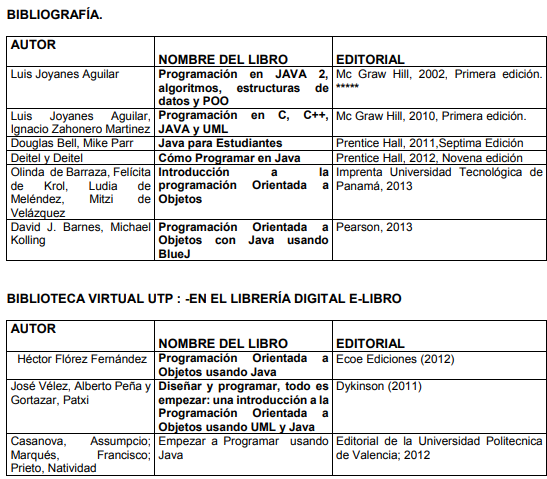
***Recursos para Testing:***

1. *JUnit Documentation*

* [*https://junit.org/junit5/*](https://junit.org/junit5/)
* *Para implementar pruebas unitarias*

1. *Java Testing Tutorials*

* *https://www.vogella.com/tutorials/JUnit/article.html*
* *Guías sobre testing en Java*



1. **RÚBRICAS:**

|  |  |  |
| --- | --- | --- |
| **Aspectos a Evaluar** | **Puntaje Máximo-100%** | **Puntaje Obtenido** |
| Estilo de escritura | **5** |  |
| Indentar | **5** |  |
| \*Claridad en la Escritura | **5** |  |
| Breve documentación a través de comentarios | **5** |  |
| Secuencia lógica. | **35** |  |
| Prueba: Captura de pantallas corrida | **25** |  |
| \*Código en Lenguaje Java | **10** |  |
| \*Optimización de instrucciones | **5** |  |
| \*Puntualidad en la entrega | **5** |  |
| Total | **100** |  |

1. **CONCLUSIÓN**

El desarrollo de este proyecto ha representado una experiencia valiosa para comprender de manera práctica los conceptos fundamentales de la programación estructurada y orientada a objetos utilizando el lenguaje Java. A lo largo de esta guía de laboratorio, se ha logrado aplicar los conocimientos teóricos adquiridos sobre la creación de métodos, la definición de clases, el uso de variables y la interacción con el usuario a través de la consola, consolidando así una base sólida para futuros retos en el área de programación. El enfoque principal del proyecto consistió en diseñar e implementar una clase denominada OperacionMatematica, que agrupara de manera modular los métodos requeridos para realizar operaciones aritméticas básicas: suma, resta, multiplicación y división, además de métodos adicionales para leer datos desde la entrada estándar e imprimir los resultados. Este enfoque modular permitió dividir el problema en partes más pequeñas y manejables, lo cual no solo mejora la comprensión del código, sino que también facilita su mantenimiento y evolución.

Durante la implementación del método de división, se abordó un aspecto crítico del manejo de errores: la verificación de la división entre cero. Este caso particular es una fuente común de errores en aplicaciones matemáticas, por lo cual fue necesario implementar una condición que impida la operación si el divisor ingresado por el usuario es igual a cero. Esta validación no solo evita que el programa falle inesperadamente, sino que además contribuye a que el usuario comprenda la importancia de los datos ingresados y su efecto sobre la ejecución del programa. El proyecto también permitió fortalecer la lógica algorítmica y la capacidad de traducir requerimientos funcionales en estructuras lógicas de programación. El uso de estructuras secuenciales en combinación con métodos independientes proporcionó una visión más clara de cómo deben organizarse los programas para lograr legibilidad y reutilización del código. Asimismo, se promovió el uso de buenas prácticas como nombrado significativo de variables y funciones, comentarios explicativos en puntos clave del código y separación de responsabilidades entre las distintas partes del programa.

Otro aspecto relevante fue la interacción con el entorno de desarrollo, en este caso Visual Studio Code y aplicaciones móviles como Java N-IDE. La familiarización con estos entornos facilitó la edición, compilación, ejecución y depuración del código, aspectos fundamentales para cualquier desarrollador en formación. Adicionalmente, se practicó la ejecución de pruebas de escritorio (pruebas manuales), donde se verificó que el programa cumpliera con los resultados esperados para diferentes valores ingresados, incluyendo escenarios normales y casos límite como el intento de dividir entre cero. La presentación final del proyecto incluyó información obligatoria como nombres de los integrantes, cédulas, nombre de la asignatura, grupo, fecha y título del reporte, reforzando también la importancia de la documentación y formalidad en el desarrollo de software, incluso en etapas académicas iniciales.

La elaboración de este laboratorio permitió afianzar de manera práctica los conocimientos fundamentales sobre el diseño y uso de métodos en Java, así como fomentar el pensamiento lógico y la estructura modular del código. A través de este ejercicio, se evidencia la importancia de planificar el diseño antes de codificar, de validar entradas, de separar la lógica en métodos específicos y de realizar pruebas exhaustivas para asegurar el correcto funcionamiento del sistema. Estos aprendizajes serán de gran utilidad en los siguientes niveles del curso y constituyen un paso esencial en la formación como programadores y futuros ingenieros de software.