**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

**Fecha:** 02/5/2025

**Grupo:** 1IL -128

1. **TÍTULO** **DE LA EXPERIENCIA**: Guía de Laboratorio Tarea: Manejo Ide, Lectura, impresión, y librerías en java, en uno solo taller crear 4 programas en java, separado o crear uno que realice todo el trabajo.
2. **TEMAS:**

Uso del IDE en Java, lectura de datos e impresión de resultados.

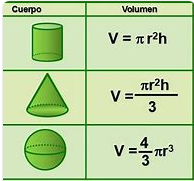
1. **OBJETIVO(S):**

Conocer las formas básicas lectura de datos, impresión de resultados y manejo del ide. Como crear un proyecto en java, como crear una clase, verificar algunos errores de sintaxis y la compilación del programa.

1. **METODOLOGÍA SUGERIDA:**
2. Se trabajará de manera individual.
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 Seudocódigo o codificación java
6. Anexar la prueba de escritorio para validar la solución, o 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 1 proyecto en Java, que permita confeccionar varios métodos en una clase, para asignar valores a variables de entrada a dos valores A, B, e imprimir los resultados. Para los datos usar datos fijos como variables, e imprimir los resultados, usar el Scanner (para la lectura), y la pantalla emergente como incluir las librerías asociadas para la lectura. Incluir el problema área del rectángulo, área del triángulo (base x altura/2), como también crear la clase Volumen (volumen para el cono, volumen del cilindro, volumen de la esfera). **Leer los datos por consola para las figuras geométricas**.

* **Glosario de 15** palabras relacionadas con el primer tema del capítulo 1,2 (UML, Do en java, while en java, constante en java,
* Dns 1.1.1.1 dns 9.9.9.9 ,
* Que es rdp, como usar el goto en java.
* Que es vector y de ejemplo en java.
* Cuña auid y año de antigüedad.
* Quien es Gary Mckinnon.
* Mecanismo anticitera y año de antigüedad. Resumir en 15 lineas
* El martillo de Londres (resumen que y año que se estima)
* Conclusión del proyecto de laboratorio.



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 A LA ACTIVIDAD):**

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

|  |  |  |  |
| --- | --- | --- | --- |
| ***Análisis y Diseño*** | | | |
| *Entradas* | *Proceso* | | *Salida* |
| **Variables fijas:**   * A = 5 * B = 3   **Datos ingresados por el usuario según la opción del menú:**   * Base y altura para área del rectángulo * Base y altura para área del triángulo * Radio y altura para volumen del cono * Radio y altura para volumen del cilindro * Radio para volumen de la esfera   **Opción del menú:**   * Número entero del 0 al 5 para seleccionar la operación | **Mostrar valores fijos y su suma.**  **Mostrar menú principal y leer la opción del usuario.**  **Según la opción seleccionada:**   * Solicitar los datos necesarios (base, altura, radio, etc.) * Llamar al método correspondiente de la clase Areas o Volumenes para realizar el cálculo: * Área del rectángulo: base × altura * Área del triángulo: (base × altura) / 2 * Volumen del cono: (π × radio² × altura) / 3 * Volumen del cilindro: π × radio² × altura * Volumen de la esfera: (4/3) × π × radio³   **Repetir el menú hasta que el usuario elija salir.** | | **En consola:**   * Muestra los valores fijos y su suma al inicio. * Muestra el menú principal. * Muestra el resultado del cálculo solicitado (área o volumen) en cada operación. * Muestra mensaje de despedida al salir. |
| ***Programa: Escribir su programa en esta sección Programa principal*** | | | |
| *import java.util.Scanner;*  *public class MenuPrincipal {*  *public static void main(String[] args) {*  *Scanner scanner = new Scanner(System.in);*  *// Variables fijas de ejemplo*  *double A = 5;*  *double B = 3;*  *System.out.println("Valores fijos de ejemplo: A = " + A + ", B = " + B);*  *System.out.println("A + B = " + (A + B));*  *boolean continuar = true;*  *while (continuar) {*  *System.out.println("\n=== MENÚ PRINCIPAL ===");*  *System.out.println("1. Área del rectángulo");*  *System.out.println("2. Área del triángulo");*  *System.out.println("3. Volumen del cono");*  *System.out.println("4. Volumen del cilindro");*  *System.out.println("5. Volumen de la esfera");*  *System.out.println("0. Salir");*  *System.out.print("Seleccione una opción: ");*  *String input = scanner.nextLine();*  *int opcion;*  *try {*  *opcion = Integer.parseInt(input);*  *} catch (NumberFormatException e) {*  *opcion = -1;*  *}*  *switch (opcion) {*  *case 1:*  *System.out.print("Ingrese la base: ");*  *double baseRect = Double.parseDouble(scanner.nextLine());*  *System.out.print("Ingrese la altura: ");*  *double alturaRect = Double.parseDouble(scanner.nextLine());*  *double areaRect = Areas.areaRectangulo(baseRect, alturaRect);*  *System.out.println("Área del rectángulo: " + areaRect);*  *break;*  *case 2:*  *System.out.print("Ingrese la base: ");*  *double baseTri = Double.parseDouble(scanner.nextLine());*  *System.out.print("Ingrese la altura: ");*  *double alturaTri = Double.parseDouble(scanner.nextLine());*  *double areaTri = Areas.areaTriangulo(baseTri, alturaTri);*  *System.out.println("Área del triángulo: " + areaTri);*  *break;*  *case 3:*  *System.out.print("Ingrese el radio: ");*  *double radioCono = Double.parseDouble(scanner.nextLine());*  *System.out.print("Ingrese la altura: ");*  *double alturaCono = Double.parseDouble(scanner.nextLine());*  *double volCono = Volumenes.volumenCono(radioCono, alturaCono);*  *System.out.println("Volumen del cono: " + volCono);*  *break;*  *case 4:*  *System.out.print("Ingrese el radio: ");*  *double radioCil = Double.parseDouble(scanner.nextLine());*  *System.out.print("Ingrese la altura: ");*  *double alturaCil = Double.parseDouble(scanner.nextLine());*  *double volCil = Volumenes.volumenCilindro(radioCil, alturaCil);*  *System.out.println("Volumen del cilindro: " + volCil);*  *break;*  *case 5:*  *System.out.print("Ingrese el radio: ");*  *double radioEsf = Double.parseDouble(scanner.nextLine());*  *double volEsf = Volumenes.volumenEsfera(radioEsf);*  *System.out.println("Volumen de la esfera: " + volEsf);*  *break;*  *case 0:*  *continuar = false;*  *System.out.println("¡Hasta luego!");*  *break;*  *default:*  *System.out.println("Opción no válida.");*  *}*  *}*  *scanner.close();*  *}*  *}* | | | |
| ***En esta otra puede incluir los métodos utilizados*** | | | |
| ***Método: Áreas.java***  *public class Areas {*  *// Método para calcular el área del rectángulo*  *public static double areaRectangulo(double base, double altura) {*  *return base \* altura;*  *}*  *// Método para calcular el área del triángulo*  *public static double areaTriangulo(double base, double altura) {*  *return (base \* altura) / 2;*  *}*  *}* | | | |
| ***Método: Volúmenes.java***  *public class Volumenes {*  *// Método para calcular el volumen del cono*  *public static double volumenCono(double radio, double altura) {*  *return (Math.PI \* Math.pow(radio, 2) \* altura) / 3;*  *}*  *// Método para calcular el volumen del cilindro*  *public static double volumenCilindro(double radio, double altura) {*  *return Math.PI \* Math.pow(radio, 2) \* altura;*  *}*  *// Método para calcular el volumen de la esfera*  *public static double volumenEsfera(double radio) {*  *return (4.0 / 3.0) \* Math.PI \* Math.pow(radio, 3);*  *}*  *}* | | | |
| ***En esta sección capture las pantallas de la corrida de su programa.*** | | | |
|  | |  | |
|  | | | |
|  | | | |

**GLOSARIO**

1. **UML (Lenguaje Unificado de Modelado):** Es un lenguaje estándar utilizado para modelar sistemas de software mediante diagramas visuales que representan la estructura, comportamiento e interacciones del sistema
2. **Do en Java:** Es una estructura de control de bucle que primero ejecuta un bloque de código y luego verifica una condición para determinar si debe repetirse, garantizando que el bloque se ejecute al menos una vez.
3. **While en Java:** Es una estructura de bucle que repite continuamente un bloque de código mientras se cumpla una condición booleana especificada, verificando la condición antes de cada iteración.
4. **Constante en Java:** Se refiere a una variable cuyo valor no puede modificarse después de su inicialización, declarada usando la palabra clave final y cuyo nombre por convención se escribe en mayúsculas.
5. **DNS 1.1.1.1:** Es un servicio público de resolución de nombres de dominio ofrecido por Cloudflare, reconocido por su alto rendimiento y enfoque en la privacidad del usuario.
6. **DNS 9.9.9.9:** Es un servidor DNS público operado por Quad9 que proporciona resolución de nombres con protección integrada contra dominios maliciosos y sitios de phishing.
7. **RDP (Remote Desktop Protocol):** Es un protocolo desarrollado por Microsoft que permite a los usuarios conectarse y controlar de forma remota otra computadora a través de una interfaz gráfica.
8. **Go to en Java:** Aunque Java no incluye una instrucción goto tradicional, se puede simular su funcionalidad mediante el uso de etiquetas combinadas con las sentencias break y continue para controlar el flujo del programa.
9. **“Vector” en Java:** En Java, un Vector es una clase que implementa una lista dinámica (similar a un ArrayList), pero **sincronizada**.   
   **Ejemplo:** Vector<String> nombres = new Vector<>(Arrays.asList("Ana", "María", "Luis", "Carlos"));   
   [Ana, luis, María, Carlos]
10. **Cuña auid Y año de antiguedad:** La cuña de Aiud es un objeto metálico misterioso encontrado en Rumanía, que data de hace al menos 40,000 años y está hecho de una aleación de aluminio desconocida para la humanidad, lo que sugiere orígenes extraterrestres.
11. **Quien es Gary Mckinnon:** Gary McKinnon es un administrador de sistemas y hacker escocés, nacido en febrero de 1966 en Glasgow, conocido por haber llevado a cabo lo que las autoridades estadounidenses describieron como "el mayor ataque informático militar de todos los tiempos”. Esto al acceder sin autorización a 97 computadoras pertenecientes a la NASA y al Departamento de Defensa de los Estados Unidos Entre febrero de 2001 y marzo de 2002.
12. **Mecanismo anticitera y año de antigüedad:** El Mecanismo de Anticitera es un antiguo dispositivo mecánico descubierto en 1901 en un naufragio cerca de la isla griega de Anticitera. Data aproximadamente del año 150 a.C., por lo que tiene más de 2,000 años de antigüedad. Su compleja maquinaria de engranajes de bronce era capaz de predecir posiciones astronómicas y eclipses con notable precisión. También podía mostrar los ciclos del Sol, la Luna, los planetas conocidos entonces y el calendario olímpico griego. Estaba compuesto por más de 30 engranajes entrelazados, ensamblados con una precisión sorprendente para su época, lo que demuestra un avanzado conocimiento en mecánica e ingeniería. El mecanismo estaba alojado en una caja de madera con inscripciones explicativas. Usaba engranajes de bronce dispuestos en un sistema muy preciso. Girando una manivela, el usuario podía simular el movimiento del cielo. El Mecanismo de Anticitera es considerado una muestra asombrosa de la ciencia helenística, comparado por algunos como una especie de “reloj suizo” de la Antigüedad. Su diseño muestra que los griegos tenían un conocimiento profundo del movimiento celeste y la mecánica de engranajes.
13. **Martillo de Londres resumen y año que se estima:** El **Martillo de Londres** es un artefacto que fue encontrado en 1938 en Londres, Texas, Estados Unidos. Este objeto consiste en un martillo con una cabeza de hierro fundido y un mango de madera. Lo peculiar de este hallazgo es que el martillo estaba incrustado en una piedra de 100 millones de años de antigüedad, lo que ha generado controversia y teorías sobre su origen.
14. **Scanner en Java:** Es una clase de la biblioteca java.util que permite leer entradas del usuario desde la consola, como números o cadenas de texto. Se utiliza comúnmente para capturar datos dinámicos en programas interactivos.
15. **Método en Java:** Es un bloque de código que se puede ejecutar cuando es llamado por su nombre. Permite reutilizar código y organizar la lógica del programa en partes funcionales, como los métodos para calcular áreas o volúmenes.

1. **BIBLIOGRAFÍA:**

* Deitel, P. & Deitel, H. (2017). *Java: How to Program (11th Edition).* Pearson Education.
* Schildt, H. (2018). *Java: The Complete Reference (11th Edition).* McGraw-Hill Education.
* Oracle. (s.f.). *The Java™ Tutorials*. Recuperado de: https://docs.oracle.com/javase/tutorial/
* Oracle. (s.f.). *Class Scanner*. Recuperado de: <https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html>
* Oracle. (s.f.). *Class JOptionPane*. Recuperado de: <https://docs.oracle.com/javase/8/docs/api/javax/swing/JOptionPane.html>
* GeeksforGeeks. (2023). *Java Scanner Class*. Recuperado de: <https://www.geeksforgeeks.org/scanner-class-in-java/>
* Programiz. (s.f.). *Java Methods*. Recuperado de: <https://www.programiz.com/java-programming/methods>
* Stack Overflow. (2010). *How to use Scanner in Java?* Recuperado de: <https://stackoverflow.com/questions/5287538/how-to-use-scanner-in-java>
* Stack Overflow. (2012). *How to use JOptionPane to show message dialogs?* Recuperado de: <https://stackoverflow.com/questions/7080205/how-to-use-joptionpane-to-show-message-dialogs>
* Khan Academy. (s.f.). *Área y volumen de figuras geométricas*. Recuperado de: <https://es.khanacademy.org/math/geometry>
* Stewart, J. (2012). *Cálculo de una variable*. Cengage Learning.
* GitHub. (2021). *Java Geometry Calculator Example*. Recuperado de: <https://github.com/rodrigorgs/geometry-calculator>
* YouTube. (2019). *Java Tutorial for Beginners [Video]*. Programming with Mosh. Recuperado de: <https://www.youtube.com/watch?v=eIrMbAQSU34>
* YouTube. (2020). *Java Swing Tutorial: Simple Dialogs with JOptionPane [Video]*. Bro Code. Recuperado de: <https://www.youtube.com/watch?v=5o3fMLPY7qY>
* TutorialsPoint. (s.f.). *Java - Methods*. Recuperado de: <https://www.tutorialspoint.com/java/java_methods.htm>

Tabla

El contenido generado por IA puede ser incorrecto.

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 en Java constituyó una valiosa oportunidad para aplicar y consolidar conocimientos fundamentales de programación orientada a objetos, así como para profundizar en el diseño e implementación de soluciones informáticas orientadas a la resolución de problemas matemáticos. El trabajo en equipo fue un pilar esencial durante todas las etapas del proyecto, permitiendo la integración de diversas perspectivas y habilidades, lo que enriqueció tanto el proceso como el resultado final. Desde la fase de análisis y diseño, el equipo colaboró activamente en la definición de los requerimientos funcionales y en la estructuración del programa mediante clases independientes y especializadas. La creación de las clases Areas y Volumenes permitió encapsular de manera eficiente los métodos para el cálculo de áreas y volúmenes de figuras geométricas, promoviendo la reutilización del código y facilitando futuras ampliaciones o modificaciones. La clase principal, encargada de la interacción con el usuario, fue diseñada para ofrecer una experiencia intuitiva y robusta, gestionando adecuadamente la entrada de datos y la presentación de resultados.

Durante la implementación, se aplicaron buenas prácticas de programación, como el uso de métodos estáticos para operaciones matemáticas, el manejo de excepciones para validar las entradas del usuario y una clara separación entre la lógica de negocio y la interfaz de usuario. El uso de la clase Scanner para la lectura de datos por consola permitió una interacción directa y sencilla, mientras que la impresión de resultados en consola garantizó la claridad y accesibilidad de la información generada. Este proyecto también representó un ejercicio práctico de aplicación de fórmulas matemáticas para el cálculo de áreas y volúmenes, integrando conceptos de geometría y álgebra en un entorno de programación real. La correcta implementación de estas fórmulas y su validación mediante pruebas colaborativas aseguraron la precisión y confiabilidad de los resultados obtenidos.

El trabajo en equipo se evidenció en la distribución equitativa de tareas, la revisión cruzada de código, la resolución conjunta de problemas y la toma de decisiones consensuada. La comunicación constante y el apoyo mutuo permitieron superar los desafíos técnicos y organizativos surgidos durante el desarrollo, fortaleciendo no solo las competencias técnicas individuales, sino también las habilidades interpersonales y de gestión de proyectos. Este proyecto no solo cumplió con los objetivos académicos y técnicos propuestos, sino que también fomentó un ambiente de aprendizaje colaborativo y profesional. El resultado es una aplicación modular, eficiente y fácil de usar, que puede servir como base para desarrollos futuros y como ejemplo de buenas prácticas en el desarrollo de software en equipo.