



JAVA SIN IDE



MÓDULO: PRO401-9523-225081-ONL-TALLER DE APLICACIONES MÓVILES

SEMANA: 3

Docente: IVAN AYALA AYALA

Estudiante: CLAUDIO ZUÑIGA CONTRERAS

Índice

INTRODUCCIÓN	3
DESARROLLO	4
DOCUMENTACIÓN CODIGO.....	4
EJECUCIÓN EN CMD	6
REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES.....	6
Requerimientos Funcionales	6
Requerimientos No Funcionales	7
HISTORIAS DE USUARIOS Y CRISTERIOS DE ACEPTACIÒN	8
CONCLUSIÓN	9
GITHUB.....	9
BIBLIOGRAFÍA.....	9

INTRODUCCIÓN

El presente trabajo tiene como objetivo comprender y aplicar el proceso de compilación y ejecución de un programa en lenguaje Java sin el uso de un entorno de desarrollo integrado (IDE). Para ello, se utilizó el compilador oficial `javac` incluido en el JDK de JavaSE, lo que permite transformar el código fuente en bytecode ejecutable por la Máquina Virtual de Java (JVM). Este enfoque fomenta el aprendizaje práctico de los fundamentos del ciclo de vida de un programa en Java, desde la escritura del código en un editor de texto simple hasta su ejecución en consola. Asimismo, se documentó cada línea del código y se estructuró un repositorio en GitHub, incorporando los requerimientos, historias de usuario y el seguimiento detallado del proceso.

DESARROLLO

Para esta actividad, genera una aplicación que pida los siguientes datos en un programa java (sin herramienta IDE). Para ello, utiliza la declaración de las variables como String e int, según corresponda, además de Scanner (System.in) y System.out de java.

DOCUMENTACIÓN CODIGO

```
import java.util.Scanner; // Importamos la librería Scanner para leer datos desde teclado
```

```
// Declaramos la clase principal llamada Vehiculo
```

```
public class Vehiculo {
```

```
    // Método principal del programa, punto de entrada de la aplicación
```

```
    public static void main(String[] args) {
```

```
        // Se crea un objeto Scanner para capturar la entrada del usuario desde consola
```

```
        Scanner sc = new Scanner(System.in);
```

```
        // ----- Variables del usuario -----
```

```
        String nombre; // Almacena el nombre del usuario
```

```
        int edad; // Almacena la edad del usuario
```

```
        // ----- Variables del vehículo -----
```

```
        String marca; // Almacena la marca del vehículo
```

```
        String modelo; // Almacena el modelo del vehículo
```

```
        String cilindrada; // Almacena la cilindrada del vehículo
```

```
        String combustible; // Almacena el tipo de combustible
```

```
        int capacidad; // Almacena la cantidad de pasajeros del vehículo
```

```
        // ----- Captura de datos del usuario -----
```

```
        System.out.print("Ingrese su nombre: "); // Muestra mensaje en pantalla
```

```
        nombre = sc.nextLine(); // Lee el nombre escrito por el usuario
```

```
        System.out.print("Ingrese su edad: "); // Solicita la edad
```

```
        edad = sc.nextInt(); // Captura la edad ingresada como número entero
```

```

sc.nextLine(); // Limpia el buffer de entrada

// ----- Captura de datos del vehículo -----

System.out.print("Ingrese la marca: "); // Solicita la marca del vehículo

marca = sc.nextLine(); // Captura la marca


System.out.print("Ingrese el modelo: "); // Solicita el modelo del vehículo

modelo = sc.nextLine(); // Captura el modelo


System.out.print("Ingrese la cilindrada: "); // Solicita la cilindrada

cilindrada = sc.nextLine(); // Captura la cilindrada


System.out.print("Ingrese el tipo de combustible: "); // Solicita el tipo de combustible

combustible = sc.nextLine(); // Captura el tipo de combustible


System.out.print("Ingrese la capacidad en pasajeros: "); // Solicita la capacidad

capacidad = sc.nextInt(); // Captura la capacidad en pasajeros

// ----- Salida de datos del usuario -----

System.out.println("\n--- Información del Usuario ---"); // Título de sección

System.out.println("Nombre: " + nombre); // Imprime el nombre ingresado

System.out.println("Edad: " + edad + " años"); // Imprime la edad ingresada


// ----- Salida de datos del vehículo -----

System.out.println("\n--- Información del Vehículo ---"); // Título de sección

System.out.println("La marca que ha ingresado es: " + marca); // Muestra la marca

System.out.println("El modelo que ha ingresado es: " + modelo); // Muestra el modelo

System.out.println("La cilindrada que ha ingresado es: " + cilindrada); // Muestra la cilindrada

System.out.println("El tipo de combustible es: " + combustible); // Muestra el combustible

System.out.println("Tiene una capacidad de " + capacidad + " pasajeros."); // Muestra capacidad
}

}

```

EJECUCIÓN EN CMD

Una vez creado el código en Bloc de Notas (.txt), se guarda como extensión (.java) en esta ocasión es Vehiculo.java, en CMD se realiza la conversión utilizando Javac como se muestra en el pantallazo, luego se realiza la ejecución del archivo recién creado por Javac que es (Vehiculo.class) y este pide la información al usuario y la compila al final de esta.

```
E:\AEIP\Segundo Semestre 2025\TALLER DE APLICACIONES MOVILES\S3>javac Vehiculo.java
E:\AEIP\Segundo Semestre 2025\TALLER DE APLICACIONES MOVILES\S3>java Vehiculo
Ingrese su nombre: Claudio
Ingrese su edad: 37
Ingrese la marca: TOYOTA
Ingrese el modelo: COROLLA
Ingrese la cilindrada: 1000cc
Ingrese el tipo de combustible: 95
Ingrese la capacidad en pasajeros: 5

--- Información del Usuario ---
Nombre: Claudio
Edad: 37 años

--- Información del Vehículo ---
La marca que ha ingresado es: TOYOTA
El modelo que ha ingresado es: COROLLA
La cilindrada que ha ingresado es: 1000cc
El tipo de combustible es: 95
Tiene una capacidad de 5 pasajeros.

E:\AEIP\Segundo Semestre 2025\TALLER DE APLICACIONES MOVILES\S3> LISTO
```

REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES

Requerimientos Funcionales

El sistema debe permitir al usuario ingresar su nombre y edad antes de registrar los datos del vehículo.

El sistema debe solicitar al usuario la siguiente información del vehículo:

- Marca
- Modelo
- Cilindrada
- Tipo de combustible
- Capacidad de pasajeros

El sistema debe mostrar en pantalla la información ingresada por el usuario, tanto personal como del vehículo.

El sistema debe ejecutarse desde la línea de comandos utilizando el compilador javac y la máquina virtual java.

El sistema debe generar una salida estructurada y clara que diferencie los datos del usuario y del vehículo.

El programa debe validar la secuencia de entrada para evitar errores comunes (por ejemplo, limpiar el buffer tras leer números).

Requerimientos No Funcionales

Usabilidad:

El sistema debe ser fácil de utilizar, mostrando mensajes claros en la consola.

Portabilidad:

El programa debe poder ejecutarse en distintos sistemas operativos (Windows, Linux, macOS) siempre que tengan instalado el JDK.

Mantenibilidad:

El código debe estar comentado línea por línea para facilitar su comprensión y futuras modificaciones.

Eficiencia:

La ejecución del programa no debe superar los 2 segundos en hardware estándar.

Seguridad:

El programa no debe almacenar ni transmitir los datos ingresados; solo mostrarlos en consola.

Escalabilidad:

El diseño del programa debe permitir agregar fácilmente nuevas funcionalidades (por ejemplo, más atributos del vehículo).

HISTORIAS DE USUARIOS Y CRISTERIOS DE ACEPTACIÓN

ID	HISTORIA DE USUARIO	CRISTERIO DE ACEPTACION
HU1	Como usuario, quiero ingresar mi nombre y edad para que el sistema me identifique antes de registrar el vehículo.	<ul style="list-style-type: none"> - El sistema solicita nombre y edad al inicio. - Los datos ingresados se muestran al final junto con los del vehículo.
HU2	Como usuario, quiero ingresar los datos de mi vehículo para que el sistema los muestre de forma clara.	<ul style="list-style-type: none"> - El sistema solicita marca, modelo, cilindrada, tipo de combustible y capacidad. - El sistema imprime en pantalla los valores ingresados.
HU3	Como usuario, quiero que el programa muestre la información de manera organizada para distinguir mis datos personales de los del vehículo.	<ul style="list-style-type: none"> - La salida debe mostrar un bloque con los datos del usuario y otro con los del vehículo.
HU4	Como administrador o estudiante, quiero ejecutar el programa sin usar un IDE, solo con <i>"javac"</i> y java, para practicar el proceso de compilación manual.	<ul style="list-style-type: none"> - El programa se compila con <i>"javac"</i> NombreArchivo.java. - El programa se ejecuta con java <i>"NombreArchivo"</i>.
HU5	Como desarrollador, quiero que el código esté documentado línea por línea, para comprender fácilmente la lógica y realizar modificaciones futuras.	<ul style="list-style-type: none"> - Cada línea del código debe tener un comentario claro y conciso. - El repositorio debe incluir documentación detallada.

CONCLUSIÓN

La realización de este trabajo permitió comprender la importancia de conocer las bases del lenguaje Java y su proceso de compilación sin depender de herramientas gráficas como los IDEs. Al trabajar directamente con el compilador `javac` y la JVM, se refuerzan los conceptos esenciales de programación, manejo de archivos, rutas y comandos en consola. Además, el desarrollo del programa contribuyó a aplicar buenas prácticas de documentación, tanto en el código como en el repositorio, garantizando la trazabilidad y mantenibilidad del proyecto. Finalmente, este aprendizaje no solo fortalece las competencias técnicas en Java, sino que también entrega una base sólida para enfrentar proyectos más complejos en entornos de desarrollo profesional.

GITHUB

Enlace para revisión

<https://github.com/dracilidan001>

BIBLIOGRAFÍA

- Oracle. (2024). *Java SE Development Kit 8 Downloads*. Recuperado de: <https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>
- Sierra, K., & Bates, B. (2005). *Head First Java*. O'Reilly Media.
- Eckel, B. (2006). *Thinking in Java* (4th Edition). Prentice Hall.