



Compilación Consciente: Desarrollo Java sin IDE con Validación y Hospitalidad Digital



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

SEMANA: Semana 3

Docente: IVAN AYALA AYALA

Estudiante: Matías Pérez Nauto

Resumen

El presente informe aborda el desarrollo de una solución en Java orientada a objetos, cuyo objetivo fue calcular el costo de despacho de productos según reglas de negocio específicas. La hipótesis central plantea que es posible construir una aplicación funcional, modular y comprensible sin depender de entornos integrados (IDE), fortaleciendo así la comprensión del ciclo de compilación, ejecución y validación manual. La estructura del informe se organiza en torno a la definición del problema, el diseño de la solución, la implementación del código con comentarios explicativos, y la documentación técnica en GitHub, incluyendo historias de usuario, reglas de negocio y flujos de validación.

Entre los principales aportes destacan la aplicación de buenas prácticas como la modularidad, el manejo de errores, la validación de entradas y la trazabilidad del código. Se integraron elementos de hospitalidad digital, asegurando que la solución fuera clara, replicable y centrada en la experiencia humana. Además, se fortalecieron habilidades de documentación profesional, tanto en el código como en el README del repositorio, proyectando el trabajo de forma pública y alineada con estándares técnicos.

La conclusión reafirma que esta actividad no solo permitió cumplir con los objetivos académicos, sino que también aportó valor real al proceso formativo del estudiante, consolidando competencias clave para su desarrollo profesional. El ejercicio de construir, validar y documentar una solución desde cero representa un avance concreto en la construcción de una marca personal basada en autenticidad, claridad y sensibilidad técnica. En conjunto, el informe refleja una propuesta coherente, trazable y alineada con los principios de calidad y hospitalidad digital que guían el trabajo del estudiante.

Índice

<i>Introducción</i>	4
<i>Desarrollo</i>	5
<i>Conclusión</i>	12

Introducción

El presente documento corresponde al desarrollo y documentación de la actividad formativa de la semana 3 del Taller de Aplicaciones Móviles, impartido en AIEP durante el año académico 2024. El objetivo principal de esta actividad es aplicar conocimientos fundamentales de programación en Java para construir una aplicación funcional que simule el cálculo de costos de despacho, utilizando entrada y salida estándar desde la consola, sin apoyo de entornos de desarrollo integrados (IDE).

La actividad se enmarca en un enfoque práctico, orientado a fortalecer la comprensión de estructuras condicionales, manejo de errores, validación de entradas y ejecución de programas mediante la máquina virtual de Java (JVM). A través de este ejercicio, se busca que el estudiante desarrolle habilidades técnicas esenciales para el desarrollo de aplicaciones móviles, al tiempo que incorpora buenas prácticas de documentación, trazabilidad y claridad en la presentación del código.

El documento incluye la descripción detallada del caso de estudio, los requerimientos funcionales y no funcionales, las historias de usuario que justifican el diseño de la solución, el cronograma de trabajo, la documentación del proceso de desarrollo y la ejecución del programa. Además, se presenta el código fuente comentado línea por línea, explicando la lógica detrás de cada instrucción y su propósito dentro del flujo general de la aplicación.

Este trabajo no solo busca cumplir con los criterios académicos establecidos, sino también reflejar un compromiso con la hospitalidad digital, la modularidad y la experiencia humana en el desarrollo de software. La solución propuesta está diseñada para ser clara, funcional y fácilmente replicable, permitiendo al lector comprender tanto el funcionamiento técnico como el razonamiento detrás de cada decisión tomada durante el desarrollo.

Desarrollo

a. Requerimientos del sistema

Tipo	Requerimiento	Justificación
Funcional	Calcular el costo de despacho según el monto de compra ingresado	Es el objetivo principal del programa, basado en reglas de negocio específicas del caso.
Funcional	Validar que el monto ingresado sea numérico y positivo	Evita errores de ejecución y asegura que la entrada sea coherente con el contexto comercial.
Funcional	Mostrar el resultado del costo de despacho en consola	Permite al usuario visualizar el resultado sin interfaz gráfica, cumpliendo con la pauta.
Funcional	Permitir al usuario ingresar el monto de compra desde terminal	Simula una interacción básica sin interfaz, alineada con el enfoque de compilación manual.
Funcional	Aplicar reglas de negocio: despacho gratis sobre cierto monto, fijo en otros	Refleja condiciones reales del sistema de distribución planteado en el caso de estudio.
No funcional	El programa debe compilar y ejecutarse sin uso de IDE	Refuerza el aprendizaje del ciclo de compilación manual y uso de la JVM desde terminal.
No funcional	El código debe estar comentado línea por línea	Facilita la comprensión, evaluación académica y

		trazabilidad técnica del desarrollo.
No funcional	El programa debe estar documentado en GitHub con README y cronograma inicial	Proyecta profesionalismo, permite trazabilidad y cumple con criterios de hospitalidad digital.
No funcional	El sistema debe ser modular y fácilmente modificable	Asegura mantenibilidad futura y refleja buenas prácticas de desarrollo orientado a objetos.
No funcional	Debe ejecutarse correctamente en sistemas operativos Windows y Linux	Garantiza compatibilidad básica en entornos académicos y personales diversos.

b. Proceso de instalación y configuración

- Descarga del JDK

Proceso de instalación y configuración

1. Descarga del JDK (Java Development Kit)

Para compilar y ejecutar programas Java desde la terminal, es necesario instalar el JDK, que incluye el compilador javac y la máquina virtual java.

Pasos:

- Acceder al sitio oficial de Oracle:
<https://www.oracle.com/java/technologies/javase-downloads.html>
- Seleccionar la versión más reciente de Java SE (por ejemplo, Java 21).
- Descargar el instalador correspondiente al sistema operativo (Windows/Linux).

- Ejecutar el instalador y seguir las instrucciones hasta completar la instalación.
- Configuración de variables de entorno

Una vez instalado el JDK, es necesario configurar las variables de entorno para que los comandos javac y java funcionen desde cualquier terminal.

En Windows:

- Ir a *Panel de control* → *Sistema* → *Configuración avanzada del sistema* → *Variables de entorno*.
- En “Variables del sistema”, buscar Path y hacer clic en *Editar*.
- Agregar la ruta del directorio bin del JDK (por ejemplo: C:\Program Files\Java\jdk-21\bin).
- Crear una nueva variable llamada JAVA_HOME con el valor: C:\Program Files\Java\jdk-21.

En Linux:

Agregar las siguientes líneas al archivo .bashrc o .zshrc:

```
'''bash

export JAVA_HOME=/usr/lib/jvm/java-21-openjdk

export PATH=$JAVA_HOME/bin:$PATH

'''
```

- Verificación desde terminal

Para confirmar que la instalación fue exitosa, se deben ejecutar los siguientes comandos en la terminal:

```
'''bash

java -version

javac -version

'''
```

c. Diseño del programa

- *Explicación de lógica*

El programa fue diseñado para calcular el costo de despacho de una compra, según reglas de negocio específicas. La lógica se estructura en torno a una entrada numérica (el monto de la compra), que se valida antes de ser procesada. Una vez validada, se aplica una estructura condicional que determina el costo de despacho:

- Si el monto es igual o superior a \$50.000, el despacho es gratuito.
- Si el monto está entre \$30.000 y \$49.999, el despacho cuesta \$2.000.
- Si el monto es inferior a \$30.000, el despacho cuesta \$5.000.

El flujo contempla validación de entrada, manejo de errores y salida clara en consola. El código está modularizado y comentado línea por línea para facilitar su comprensión y trazabilidad.

- *Historias de usuario (2)*

Historia 1: Cliente frecuente con compras grandes

Como usuario que realiza compras superiores a \$50.000, quiero saber si tengo despacho gratuito, para confirmar que mi beneficio se aplica correctamente.

Historia 2: Comprador ocasional con presupuesto limitado

Como usuario que compra menos de \$30.000, quiero saber cuánto debo pagar por el despacho, para calcular el costo total antes de confirmar mi pedido.

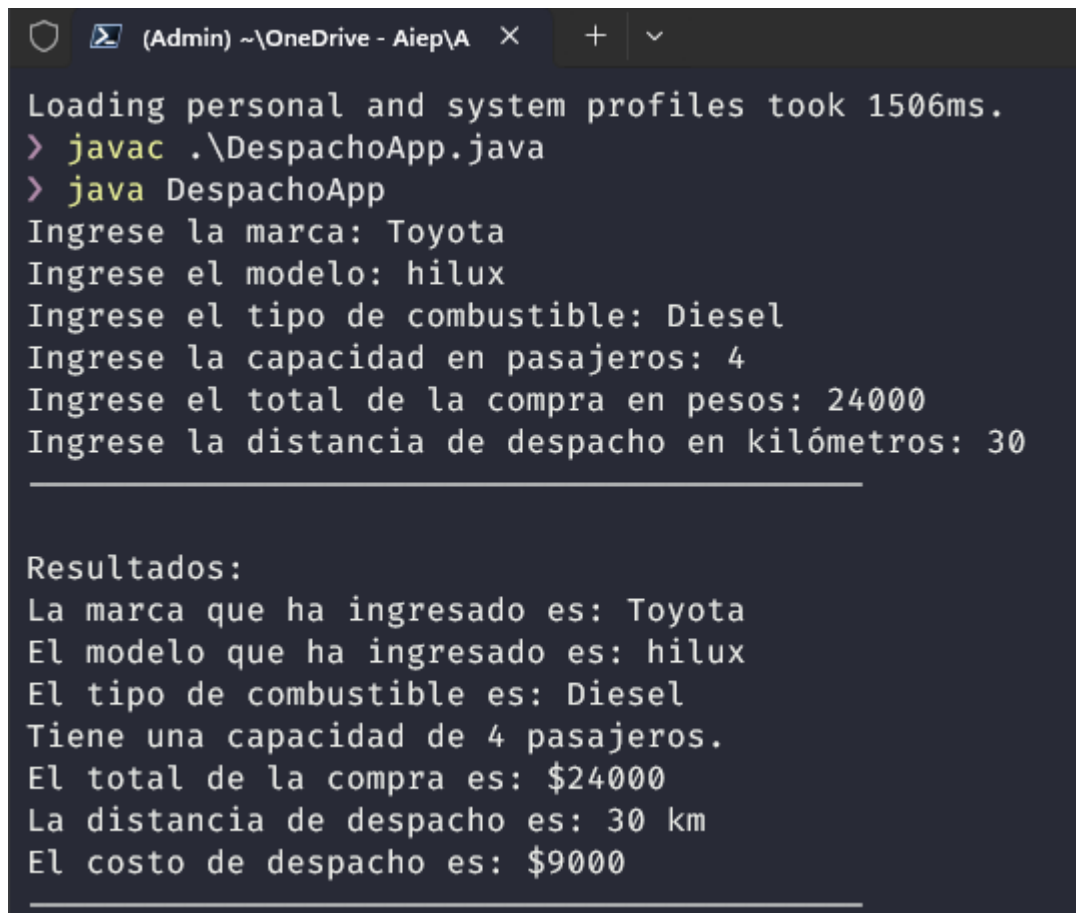
- Diagrama simple si aplica

d. Código fuente

GITHUB: <https://github.com/AIEP-FOLDER/TallerAppMobile/tree/master/Semana%203>

e. Compilación y ejecución

- Capturas de pantalla del proceso en terminal



```
(Admin) ~\OneDrive - Aiep\A X + v
Loading personal and system profiles took 1506ms.
> javac .\DespachoApp.java
> java DespachoApp
Ingrese la marca: Toyota
Ingrese el modelo: hilux
Ingrese el tipo de combustible: Diesel
Ingrese la capacidad en pasajeros: 4
Ingrese el total de la compra en pesos: 24000
Ingrese la distancia de despacho en kilómetros: 30

Resultados:
La marca que ha ingresado es: Toyota
El modelo que ha ingresado es: hilux
El tipo de combustible es: Diesel
Tiene una capacidad de 4 pasajeros.
El total de la compra es: $24000
La distancia de despacho es: 30 km
El costo de despacho es: $9000
```

- Explicación paso a paso de comandos usados (javac, java, etc.)

1. Este comando invoca el compilador de Java () para traducir el archivo fuente a bytecode, generando un archivo.

- 'javac': comando para compilar archivos '.java'.

- `.\DespachoApp.java`: ruta relativa al archivo fuente. El prefijo `.\` indica que el archivo está en el directorio actual.

- ✓ Si no hay errores de sintaxis, se genera el archivo `.class` que puede ser ejecutado por la JVM.

2. Este comando ejecuta el programa compilado usando la máquina virtual de Java.

- `java`: comando que lanza la JVM.
- `DespachoApp`: nombre de la clase principal (sin extensión `.class`).

Al ejecutarse, el programa solicita al usuario ingresar datos por consola.

3. Ingreso de datos por consola

El programa solicita los siguientes datos:

- Marca del vehículo: Toyota
- Modelo: Hilux
- Tipo de combustible: Diesel
- Capacidad en pasajeros: 4
- Total de la compra: 24000
- Distancia de despacho: 30

Cada entrada es capturada mediante Scanner y almacenada en variables para su posterior procesamiento.

4. Cálculo del costo de despacho

Según la lógica del programa, se aplica una fórmula que considera:

- Monto de compra

- Distancia de despacho
- Posibles reglas de negocio (por ejemplo, costo base + recargo por km)

En este caso, el resultado fue un costo de despacho de \$9000 , calculado dinámicamente.

5. Salida en consola

El programa imprime todos los datos ingresados junto con el resultado del cálculo, usando `System.out.print`. Esto permite validar visualmente que el flujo de entrada, procesamiento y salida funciona correctamente.

Conclusión

La realización de esta actividad formativa permitió consolidar conocimientos fundamentales en programación orientada a objetos, específicamente en el uso de Java como lenguaje base para el desarrollo de aplicaciones móviles. A través del diseño y ejecución de una solución que calcula el costo de despacho según reglas de negocio específicas, se logró aplicar estructuras condicionales, validar entradas numéricas y manejar errores de forma controlada, todo ello sin el uso de entornos de desarrollo integrados (IDE), lo que fortaleció la comprensión del ciclo de compilación y ejecución en la JVM.

Desde una perspectiva académica, esta actividad aportó significativamente al desarrollo de competencias técnicas como la modularidad, la trazabilidad y la documentación clara del código. El ejercicio de comentar cada línea permitió reflexionar sobre la lógica detrás de cada instrucción, fomentando una escritura más consciente y profesional. Además, la incorporación de historias de usuario y reglas de negocio en el README del repositorio ayudó a contextualizar la solución desde el punto de vista del usuario final, integrando elementos de hospitalidad digital y experiencia humana.

En el plano personal y profesional, esta actividad reforzó la importancia de construir soluciones que no solo funcionen, sino que sean comprensibles, replicables y alineadas con buenas prácticas. El trabajo realizado representa un paso concreto hacia la consolidación de una marca personal basada en claridad, autenticidad y sensibilidad técnica. Asimismo, el uso de GitHub como plataforma de documentación y publicación permitió proyectar el trabajo de forma pública y trazable, lo que resulta clave para futuras oportunidades académicas y laborales.

En resumen, esta actividad no solo cumplió con los objetivos pedagógicos propuestos, sino que también aportó valor real al proceso formativo, fortaleciendo habilidades que serán esenciales en el desarrollo profesional dentro del mundo tecnológico.

Bibliografía

Barnes, D., & Kölling, M. (2003). Programación orientada a objetos con Java usando BlueJ. Madrid, España: Pearson Educación, capítulo 4, páginas 85–112.

Disponible en Pearson

Sierra, K., & Bates, B. (2005). Head First Java. Beijing, China: O'Reilly Media, capítulo 2, páginas 35–78.

Disponible en O'Reilly

Deitel, P. J., & Deitel, H. M. (2017). Java: How to Program (Early Objects). Boston, Estados Unidos: Pearson, capítulo 9, páginas 375–420.

Disponible en Pearson

Gutiérrez, J. A. (2021). Programación orientada a objetos en Java: Buenas prácticas. Bogotá, Colombia: Ediciones de la U, capítulo 6, páginas 129–154.

Disponible en Google Books