

App distribuidora

MÓDULO: Taller de aplicaciones móviles

SEMANA: 3

Docente: Ivan Ayala Ayala

Estudiantes: Nicolás Brevis Neira

Índice

[Introducción 3](#_Toc208086213)

[Desarrollo 4](#_Toc208086214)

[Listado de Requerimientos. 7](#_Toc208086215)

[Requerimientos Funcionales 7](#_Toc208086216)

[Requerimientos No Funcionales 8](#_Toc208086217)

[Historias de Usuario 9](#_Toc208086218)

[HU1 – Registro de usuario 9](#_Toc208086219)

[HU2 – Ingreso de datos de compra 9](#_Toc208086220)

[HU3 – Cálculo de despacho 10](#_Toc208086221)

[Pantallazos 11](#_Toc208086222)

[Conclusión 12](#_Toc208086223)

[Bibliografía: 13](#_Toc208086224)

# **Introducción**

El trabajo tiene como objetivo desarrollar una aplicación en Java que calcule automáticamente el costo del despacho de productos según reglas de negocio.

Se busca que el estudiante comprenda el proceso completo sin depender de un IDE, usando la línea de comandos y el compilador javac.

El caso de estudio fue una distribuidora de alimentos que cobra según el valor de la compra y la distancia.

La aplicación solicita datos del usuario como nombre, edad, valor de compra y kilómetros recorridos, y muestra los datos ingresados junto con el cálculo del envío.

Se documentó el código línea por línea y se creó un repositorio en GitHub para almacenar toda la documentación y evidencia del proceso.

Además, se elaboraron los requerimientos funcionales y no funcionales, así como las historias de usuario, fomentando la planificación y organización del proyecto.

La actividad integra conocimientos de programación, documentación, control de versiones y buenas prácticas, preparando al estudiante para proyectos académicos y profesionales.

# **Desarrollo**

Primero descargué e instalé el JDK desde el sitio oficial de Oracle. Luego configuré las variables de entorno JAVA\_HOME y actualicé el PATH para poder utilizar los comandos javac y java desde cualquier ubicación en la terminal.  
Posteriormente, creé una carpeta llamada DistribuidoraProject, dentro de la cual guardé un archivo llamado Distribuidora.java con el código fuente de la aplicación.  
Una vez guardado el archivo, abrí la terminal y me posicioné en la carpeta del proyecto para compilar el programa utilizando javac Distribuidora.java. Esto generó el archivo Distribuidora.class. Luego, ejecuté el programa con java Distribuidora, ingresé los datos solicitados y comprobé que se imprimieran correctamente en la salida, junto con el cálculo del costo de despacho. Finalmente se subí todo a github.

Código utilizado:

import java.util.Scanner;

public class Distribuidora {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

// Pedimos nombre y edad (requisito)

System.out.print("Ingrese su nombre: ");

String nombre = sc.nextLine();

System.out.print("Ingrese su edad: ");

int edad = sc.nextInt();

// Pedimos valor de compra y kilómetros

System.out.print("Ingrese el valor de la compra (ej: 35000): ");

int valorCompra = sc.nextInt();

System.out.print("Ingrese la distancia en kilometros: ");

int kilometros = sc.nextInt();

// Mostramos los datos ingresados con el formato solicitado

System.out.println("La edad que ha ingresado es: " + edad);

System.out.println("El nombre que ha ingresado es: " + nombre);

System.out.println("el valor de compra que ha ingresado es: " + valorCompra);

System.out.println("la distancia en kilometros es: " + kilometros);

// Cálculo del costo de despacho según reglas de negocio (con un supuesto claro)

int costoDespacho;

if (valorCompra > 50000) {

if (kilometros <= 20) {

costoDespacho = 0; // servicio gratuito dentro de 20 km

} else {

// Supuesto: si supera 50k pero queda fuera de 20 km, aplicamos tarifa 150/km.

costoDespacho = 150 \* kilometros;

}

} else if (valorCompra >= 25000) {

costoDespacho = 150 \* kilometros;

} else {

costoDespacho = 300 \* kilometros;

}

System.out.println("Costo de despacho: $" + costoDespacho + " pesos");

sc.close();

}

}

### Listado de Requerimientos.

### ****Requerimientos Funcionales****

El sistema debe permitir al usuario ingresar su **nombre y edad** mediante consola.

El sistema debe permitir ingresar el **valor de la compra y la distancia en kilómetros.**

El sistema debe mostrar en pantalla los datos ingresados por el usuario.

El sistema debe calcular automáticamente el costo del despacho según las siguientes reglas:

Compras sobre $50.000 → **envío gratis** (si es dentro de 20 km).  
Compras entre $25.000 y $49.999 → **$150 por km recorrido.**  
Compras menores a $25.000 → **$300 por km recorrido.**

El sistema debe mostrar en pantalla el resultado del cálculo del despacho.

El sistema debe ejecutarse en la **JVM (Java Virtual Machine)** desde línea de comando, sin usar un IDE.

El sistema debe estar documentado y almacenado en un repositorio de **GitHub.**

### ****Requerimientos No Funcionales****

**Portabilidad:** El programa debe ejecutarse en cualquier dispositivo con la JVM instalada (PC o dispositivos móviles compatibles).

**Usabilidad:** La interacción debe ser clara y sencilla, solicitando datos mediante mensajes en consola.

**Eficiencia:** El cálculo del costo de despacho debe realizarse de manera inmediata al ingresar los datos.

**Mantenibilidad:** El código debe estar documentado línea por línea para facilitar futuras modificaciones.

**Disponibilidad:** El código y la documentación deben estar siempre accesibles en el repositorio de GitHub.

**Seguridad:** El acceso a la aplicación se plantea mediante cuentas Gmail para garantizar un registro controlado (supuesto).

**Confiabilidad:** Los cálculos deben ser precisos según las reglas de negocio establecidas.

**Escalabilidad:** El sistema debe poder adaptarse en el futuro para incluir más productos o reglas de negocio adicionales.

### Historias de Usuario

### ****HU1 – Registro de usuario****

**Como** cliente de la distribuidora, **quiero** registrarme en la aplicación con mi cuenta de Gmail, **para** acceder al servicio de compras y cálculo de despacho.

**Criterios de aceptación:**

El sistema debe aceptar un correo Gmail válido.

El registro debe ser confirmado con un mensaje en consola o por correo (supuesto).

### ****HU2 – Ingreso de datos de compra****

**Como** usuario de la aplicación, **quiero** ingresar mi nombre, edad, valor de compra y distancia, **para** que el sistema pueda calcular el costo de despacho correspondiente.

**Criterios de aceptación:**

El sistema debe solicitar los 4 datos de forma ordenada.

Al finalizar, debe mostrar los datos ingresados.

### ****HU3 – Cálculo de despacho****

**Como** comprador, **quiero** que el sistema calcule el costo de despacho según el monto y la distancia, **para** saber cuánto debo pagar por el envío de mis productos.

**Criterios de aceptación:**

Si la compra es sobre $50.000 y dentro de 20 km, el despacho debe ser gratuito.

Si la compra es entre $25.000 y $49.999, el despacho cuesta $150 por km.

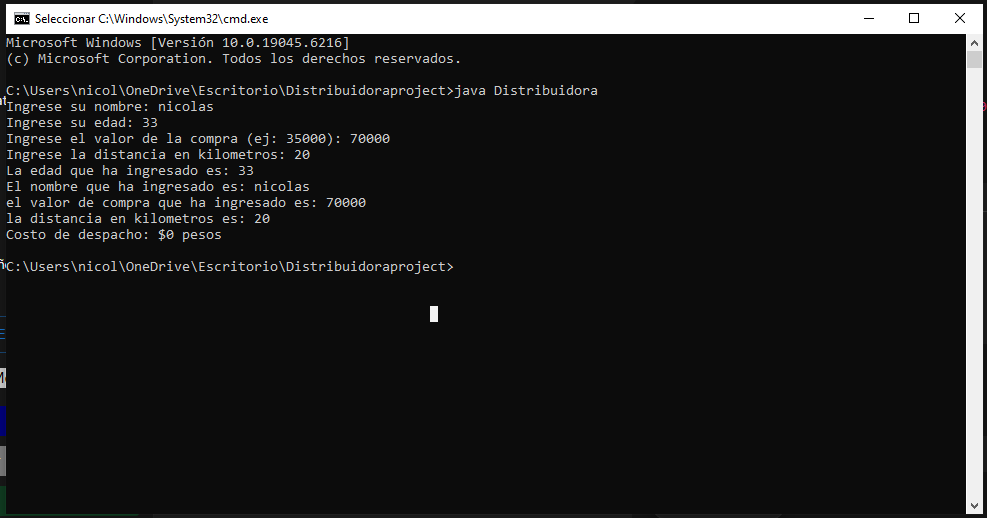
Si la compra es menor a $25.000, el despacho cuesta $300 por km.

El resultado debe mostrarse en pantalla.

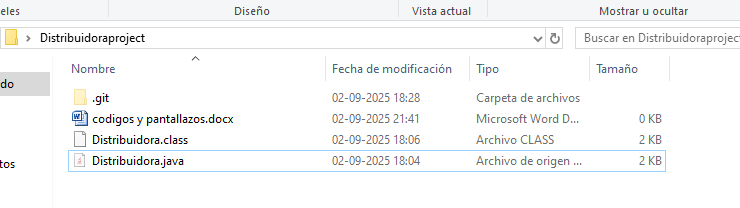
### Pantallazos

### 

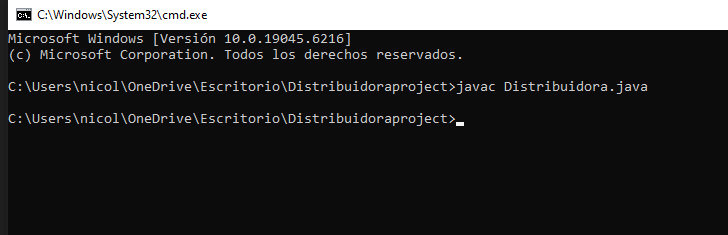
Ejecución del código



Archivos creados en la carpeta:



Se compila:



# **Conclusión**

Esta actividad permitió al estudiante comprender el ciclo completo de desarrollo en Java sin IDE, fortaleciendo la autonomía y el manejo de la línea de comandos.  
Se mejoraron habilidades en la interacción con el usuario, documentación del código, organización de proyectos y control de versiones mediante GitHub.  
El trabajo con requerimientos, historias de usuario y pantallazos fomentó la planificación, la verificación de resultados y la comunicación técnica clara.  
Además, permitió aplicar conocimientos a un caso práctico real, desarrollando capacidad de resolución de problemas y pensamiento crítico.  
En general, la actividad aporta significativamente a la vida académica y profesional del estudiante, brindando herramientas y metodologías útiles para futuros proyectos de desarrollo de software y documentación técnica.

# **Bibliografía:**

GitHub. (2025). *Hello World: GitHub Guides*. GitHub Docs. https://docs.github.com/en/get-started/quickstart/hello-world

GitHub. (2025). *Creating a new repository*. GitHub Docs. https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-new-repository