Proyecto: Sistema de Entregas y Encomiendas (SISENCO)

Modalidad: Grupal (máximo 2 personas)

Tecnología: Java con Swing + MySQL en la nube u otra base de datos en la nube.

Producto final: Sistema ejecutable con documentación completa

# Objetivo del proyecto

Desarrollar un sistema que permita gestionar el proceso de atención al cliente, recepción, envío y entrega de encomiendas en una empresa. El sistema deberá manejar acceso por roles diferenciados (administrador y cajero-despachador), tener una navegación intuitiva y validar correctamente los datos ingresados.

# 1. Login (acceso al sistema) (Todos)

- El sistema debe iniciar con una pantalla de acceso.

- Solo usuarios activos podrán ingresar.

- Se debe validar usuario, clave y rol desde la base de datos.

- Roles permitidos: ADMINISTRADOR, CAJERO\_DESPACHADOR

# 2. Registro de usuarios y empleados (solo Miembros superiores)

- El administrador puede crear usuarios y empleados.

- Datos de usuario: usuario, clave, rol, estado.

- Datos del empleado: nombre, cédula, teléfono, correo, dirección, bodega asignada.

- Este módulo es solo para registrar a los cajeros-despachadores.

# 3. Registro de clientes (solo Usuarios)

- El cajero registra nuevos clientes.

- Datos: nombre, cédula, teléfono, dirección.

- Validar cédulas duplicadas.

# 4. Registro de encomiendas (solo CAJERO\_DESPACHADOR)

- Registrar cliente, paquete, peso, origen, destino.

- Estado inicial: PENDIENTE.

- Se guarda fecha de ingreso y usuario que atiende.

# 5. Gestión de estados

- Estados posibles: PENDIENTE, ENVIADO, EN TRÁNSITO, ENTREGADO, DEVUELTO, RETENIDO.

- Registrar fecha, usuario y observaciones por cada cambio.

# 6. Historial de movimientos

- Registrar trazabilidad completa de cada encomienda.

- Mostrar fechas, acciones y responsables.

# 7. Reportes y estadísticas (solo ADMINISTRADOR)

- Filtrar encomiendas por estado, fecha, bodega, cliente o empleado.

- Mostrar totales, comparativas y estadísticas mensuales.

# 8. Interfaz gráfica

- Java Swing.

- Botones y etiquetas legibles.

- Menú constante y diseño uniforme.

- Validaciones visuales y lógicas.

- Evitar múltiples ventanas simultáneas.

# 9. Validaciones obligatorias

- Validar campos vacíos.

- Validar formato de cédula, correo, teléfono, peso.

- Manejo de errores de conexión y datos inválidos.

# 10. Funcionalidad técnica esperada

- Conexión a MySQL en la nube.

- Manejo de excepciones.

- Acceso restringido por rol.

- Registro de acciones (trazabilidad).

# 11. Entregables obligatorios

Listas de requerimientos - modelado

1. Sistema ejecutable (.exe)

2. Manual técnico

3. Capturas del sistema

4. Repositorio en GitHub con README y commits

5. Video tutorial

6. Javadoc

7. Base de datos SQL no SQL

8. Informe final del proyecto

9 exportación a Excel o PDF

# 12. Puntos extra por innovación (opcional)

- Envío de correos automáticos

- Estadísticas visuales

- Códigos QR o de seguimiento

# 13. Defensa del proyecto

- Duración máxima: 10 minutos.

- Explicar módulos desarrollados, problemas resueltos, división de trabajo y aprendizajes.

- Mostrar el sistema en funcionamiento.

RUBRICA

# 1. Desarrollo del sistema (funcionalidades técnicas)

Puntaje máximo: 4.5 puntos

Criterios de evaluación:

* - Login funcional con roles diferenciados (Admin y Cajero-Despachador)
* - Registro completo de clientes y encomiendas con validaciones
* - Actualización de estados con trazabilidad (historial de movimientos)
* - Reportes filtrables por estado, usuario, fechas
* - Interfaz uniforme, intuitiva y con buena experiencia de usuario
* - Conexión funcional a base de datos remota
* - Manejo de errores y validación de datos correcta

# 2. Informe y documentación técnica

Puntaje máximo: 1 puntos

Criterios de evaluación:

* - Modelo entidad-relación bien definido
* - Capturas del sistema con explicación
* - Manual técnico y README organizados y completes
* - Video
* - Javadoc generado correctamente

# 3. Defensa del proyecto (presentación oral)

Puntaje máximo: 1 puntos

Criterios de evaluación:

* - Presentan las funciones clave del sistema
* - Demuestran dominio del código y del flujo del sistema
* - Explican la división del trabajo en el equipo
* - Responden preguntas con claridad y seguridad

# 4. Evidencia de avances (entregas previas / GitHub)

Puntaje máximo: 0.5 puntos

Criterios de evaluación:

* - Commits frecuentes y con mensaje claro en GitHub
* - Evidencias de avances en etapas anteriores
* - Organización del proyecto en versiones o ramas si aplica