Juan Carlos Carrasco Ruiz
Al02578566

Tabla de contenidos



## 1. Resumen ejecutivo

Descripción

Este sistema permite validar entregas de gasolina pegadas desde Excel, compararlas con la base de datos tiempos_entrega, mostrar visualmente los resultados y generar reportes. Implementado en Java (JSP + Servlets).

## 2. Problema identificado

En el contexto operativo de PEMEX, el proceso de registrar, validar y analizar entregas de gasolina se realiza de forma manual o desorganizada, lo que provoca errores humanos, duplicidad de datos, dificultad para auditar y pérdida de tiempo al no contar con validaciones automáticas contra los tiempos de entrega históricos, además de ayudar a los operadores a identificar las rutas más largas, lo cual ayuda a optimizar los tiempos de entrega.

## 3. Solución

Se desarrolló un sistema web de registro y validación de entregas de gasolina para uso interno en PEMEX, el cual permite a los operadores:

- Ingresar manualmente o pegar desde Excel las rutas establecidas.
- Validar automáticamente los datos ingresados (cliente, destino) contra catálogos oficiales.
- Verificar si existe un tiempo de entrega registrado entre cliente y destino, mostrando los registros correctos en verde y resaltando los inválidos.
- Visualizar, comparar y ordenar entregas válidas por su tiempo de entrega (horas) para facilitar análisis operativos.
- Evitar duplicidad, reducir errores humanos y automatizar procesos repetitivos.

Juan Carlos Carrasco Ruiz
Al02578566

## 4. Arquitectura

El sistema sigue una arquitectura en capas basada en el patrón MVC (Modelo–Vista–Controlador). Está diseñado para mantener una separación clara entre la lógica de negocio, la presentación y el acceso a datos.

## 5. Tecnologías utilizadas:

- Java EE (Servlets + JSP)
- HTML, CSS, JavaScript
- MySQL como sistema de gestión de base de datos
- JUnit para pruebas automatizadas
- GitHub + GitHub Actions para control de versiones e integración continua

## 6. Requerimientos

- Java JDK 23 o superior
- Glassfish 6.2.1
- Servidor MySQL 8.0.17 o superior
- JDBC Driver MySQL
- NetBeans IDE
- Navegador web

## 7.Instalación

a. ¿Cómo instalar el ambiente de desarrollo?

- Clonar el proyecto desde GitHub
- Abrir en NetBeans como proyecto web
- Configurar Glassfish y MySQL
- Crear la base de datos desde el script perote.sql

b. ¿Cómo ejecutar pruebas manualmente?

- Iniciar Glassfish desde NetBeans
- Ingresar a http://localhost:8080/perote/login
- Pegar datos desde Excel y presionar “Comparar con tiempos”

c. ¿Cómo implementar en la nube o local?

- Se puede empaquetar el proyecto en un WAR y subirlo a Heroku o ejecutar en un VPS con Tomcat.

Juan Carlos Carrasco Ruiz
Al02578566

## 8. Configuración

Configuración de BD: en Conexion.java
Parámetros editables: usuario, contraseña, URL de conexión
Instalación de todos los requerimientos anteriores

Uso

a. Para usuarios finales:
- Pegar entregas en el área habilitada
- Presionar botón “Comparar con tiempos”
- Validar qué entregas son correctas
- Ordenar tiempos en las rutas de entregas de mayor a menor

b. Para administradores:
- Consultar tabla tiempos_entrega
- Agregar nuevos clientes, destinos y productos si faltan

6. Contribución

a. Guía de contribución
- Clona este repo
- Crea una nueva rama
- Agrega tus cambios
- Envía un Pull Request a la rama develop

b. Estructura de ramas
- master: versión estable
- develop: en desarrollo

## 9. Roadmap

- Exportar entregas validadas a Excel o PDF.
- Filtrado por fecha, cliente o destino desde la interfaz.
- Dashboard visual con gráficas de desempeño por vehículo y destino.
- Login por roles (Administrador, Supervisor, Chofer).
- Historial de entregas validadas con opción de buscar y editar.
- Notificaciones o alertas si una entrega no coincide con los tiempos estándar.
- Integración con sistema de geolocalización para validación con GPS.
- API REST pública para consulta de entregas por sistemas externos.

El desarrollo de estas funciones dependerá de las necesidades del usuario final y del tiempo disponible para implementación.