

PROYECTO:

Sistema Integral de Seguimiento GPS en Autobuses

Este proyecto busca desarrollar un sistema completo para monitorear, analizar y gestionar la información proveniente de autobuses utilizando datos GPS simulados.

El proyecto abordará todas las etapas del ciclo de vida del dato: captura, almacenamiento, procesamiento, análisis, distribución, uso, mantenimiento y archivado.

1. Captura o generación

Simulación manual o automática de datos GPS

- Clase GPSTData: busId, timestamp, latitude, longitude, speed.
- Generar datos GPS para 3 autobuses, intervalos de 1 minuto.
- Guardar datos en CSV `gps_data.csv`.

Se definirá una clase específica, denominada `GPSTData`, que incluirá información relevante como identificación única del autobús (`busId`), tiempo registrado en formato ISO (`timestamp`), y coordenadas geográficas (`latitude`, `longitude`) junto con la velocidad (`speed`).

Se generarán automáticamente registros simulados mediante un programa en Java, generando datos realistas para un conjunto de tres autobuses durante un periodo continuo de 60 minutos.

Estos datos se producirán con intervalos regulares de un minuto y se almacenarán en formato CSV (`gps_data.csv`) para facilitar su posterior gestión y procesamiento.

2. Almacenamiento

- `ArrayList<GPSData>`.
- Explicar almacenaría en un sistema real (BD relacional).

Los datos generados se cargarán en memoria mediante estructuras dinámicas eficientes como `ArrayList<GPSData>`, lo que permite flexibilidad y fácil acceso a los registros almacenados.

Se detallará, mediante comentarios explicativos en el código, cómo podría implementarse el almacenamiento en entornos de producción reales, incluyendo bases de datos relacionales como MySQL o PostgreSQL, y opciones NoSQL tales como MongoDB o Firebase.

Esta explicación será esencial para entender las ventajas y limitaciones de cada método en contextos específicos.

3. Procesamiento

- Procesar el archivo CSV:
- Filtrar los datos por autobús y rango horario.

Se desarrollará una lógica para validar cada registro capturado, asegurando la integridad y precisión de los datos almacenados.

Esto incluirá la revisión de rangos aceptables para las coordenadas geográficas, verificación de velocidades coherentes y comprobación estricta del formato y validez del timestamp.

Además, el sistema deberá identificar y gestionar adecuadamente datos incompletos, inconsistentes o nulos, garantizando la calidad y fiabilidad del análisis posterior.

4. Análisis

- Calcular la **velocidad media** por autobús.
- Detectar paradas (velocidad = 0).
- Agrupar y contar el número de paradas por ruta.

En esta fase se llevarán a cabo cálculos estadísticos y análisis operativos con los datos procesados.

Se determinará la velocidad media registrada para cada autobús a lo largo del período observado, permitiendo evaluar el rendimiento general del transporte.

También se identificará y contabilizará la cantidad de paradas efectuadas, definidas específicamente por momentos en los que la velocidad es cero.

Estos resultados podrán visualizarse opcionalmente mediante reportes impresos o archivos externos para facilitar la toma de decisiones operativas.

5. Distribución o Compartición

- Exportar posición de cada autobús en formato **JSON**: **bus01_status.json**

```
{  
    "busId": "BUS01",  
    "latitude": 40.4172,  
    "longitude": -3.7045,  
    "timestamp": "2025-03-25T08:45:00"  
}
```

El sistema exportará periódicamente la última posición conocida de cada autobús en formato **JSON** estructurado, con atributos claros como el identificador del autobús, sus coordenadas actuales, y la marca de tiempo correspondiente.

Estos archivos JSON se generarán individualmente para cada autobús, facilitando así su uso por parte de otras aplicaciones o servicios web externos que puedan consumir esta información en tiempo real.

Este método simula un escenario típico de integración en sistemas inteligentes de transporte urbano.

6. Uso

- Mostrar en consola o con una interfaz gráfica simple (JavaFX o Swing) el trayecto de cada autobús.
- Opcional: simular un usuario que consulta el estado actual de un bus mediante un menú interactivo.

La información recopilada y procesada se presentará mediante salidas por consola, permitiendo visualizar de manera clara y detallada el trayecto y el comportamiento de cada autobús registrado en el sistema.

Se propone como mejora adicional la implementación de una interfaz gráfica básica utilizando tecnologías JavaFX o Swing, la cual ofrecerá una visualización interactiva de trayectos y posiciones actuales.

También será posible simular un entorno de usuario interactivo para consultar el estado actualizado de los autobuses mediante opciones de menú sencillas.

7. Mantenimiento / Actualización

- Modificar registros simulando cambio de recorrido.

El proyecto permitirá simular cambios en los recorridos mediante la edición selectiva de ciertos registros previamente almacenados, imitando escenarios reales donde los autobuses deben modificar rutas por diversas razones operativas.

Tras estos cambios, los datos deberán ser reprocesados, asegurando que todos los análisis y reportes generados reflejen adecuadamente la información actualizada.

Este procedimiento ayudará a comprender el impacto operativo de cambios repentinos en el transporte público.

8. Archivado / Eliminación

- Carpeta archivados con CSV de días anteriores.
- Simular la eliminación automática de archivos antiguos (más de 7 días).

Finalmente, se implementará un mecanismo automático de archivado para gestionar de manera eficiente y organizada los archivos CSV históricos generados en días anteriores, ubicándolos dentro de una *carpeta* específica llamada "**archivados**".

Complementariamente, se simulará un proceso automático de limpieza que eliminará de manera segura y efectiva aquellos archivos cuya antigüedad supere los siete días, optimizando así el espacio de almacenamiento disponible y asegurando que el sistema mantenga un rendimiento óptimo a largo plazo.

Entrega

Se deberá subir todo el proyecto a GitHub y sólo deberás indicar en la tarea el enlace a tu repositorio con el proyecto de Digitalización.

- Código Java bien estructurado y comentado.
- Archivo CSV con datos simulados.
- Informe por consola o JSON con estadísticas.
- Explicación breve de cómo se aplicó cada etapa del ciclo de vida del dato.

Rúbrica del Proyecto

Sistema Integral de Seguimiento GPS en Autobuses

Criterio de Evaluación	Aspectos evaluados	Puntos
1. Captura o Generación de Datos	Creación clase GPSTData completa	0.5
	Generación automática y realista de datos GPS simulados	0.5
	Guardado correcto en formato CSV	0.5
2. Almacenamiento	Uso adecuado de ArrayList<GPSTData>	0.5
	Comentarios explicativos sobre almacenamiento en bases reales (relacional/NoSQL)	0.5
3. Procesamiento	Validación eficaz de coordenadas, velocidad, formatos de fecha/hora	0.5
	Detección y manejo de errores y datos nulos o inconsistentes	0.5
	Filtrado correcto por autobús y rango horario	0.5
4. Análisis	Cálculo correcto de velocidad media por autobús	0.5
	Detección y conteo adecuado de paradas (velocidad = 0)	0.5
	Agrupación clara por autobús/ruta	0.5
	Presentación estructurada del análisis	0.5

5. Distribución o Compartición	Exportación estructurada en formato JSON	0.5
	Creación correcta de archivos JSON individuales por autobús	0.5
6. Uso	Visualización clara del trayecto en consola	0.5
	Implementación opcional interfaz gráfica o menú interactivo	0.5
7. Mantenimiento / Actualización	Modificación correcta simulando cambios en recorridos	0.5
	Reprocesamiento efectivo y actualización correcta de informes	0.5
8. Archivado / Eliminación	Archivado organizado de CSV antiguos	0.5
	Eliminación automática segura de archivos antiguos (más de 7 días)	0.5
Evaluación General del Proyecto	Código estructurado, legible y bien comentado	+ 0.5
	Explicación clara sobre cómo se aplicó cada etapa del ciclo del dato	+ 0.5
Total		10