



Instituto Superior Verbo Divino

Aplicaciones

CARRERA: Técnico Superior en Análisis en Sistemas

MATERIA: Investigación Operativa

COMISIÓN: “U”

PROFESOR: Jose F. Lutgen.

ESTUDIANTES: Villalba Mariano, Franco Nielsen Anker,
Danzel Burki, Ivan Mieres, Carlos Carisimo

FECHA: 30-10-2.025

Objetivo del proyecto

Desarrollar un sistema que simule y optimice la logística de distribución de productos desde varios almacenes hacia diferentes tiendas, considerando inventarios, rutas de entrega, tiempos de transporte y costos operativos.

El sistema permite:

- Monitorear niveles de inventario en tiempo real.
- Simular la operación logística para varios días.
- Evaluar métricas clave como entregas completadas, distancia recorrida y costos.
- Visualizar rutas de entrega y evolución de inventarios en un dashboard interactivo.

Descripción de los códigos

1 Inventory.py

- **Propósito:** Gestionar inventarios de los almacenes y predecir demanda futura.
- **Funciones principales:**
 - InventoryManagementSystem: clase que inicializa con datos históricos de inventario.
 - train_demand_forecaster: entrena un modelo RandomForest para predecir demanda.

- `create_features`: genera características temporales y codifica almacenes.
- `multi_echelon_inventory_optimization`: optimiza niveles de inventario para minimizar costos.
- `get_inventory_status`: devuelve un DataFrame con la evolución del inventario.

Uso: Se usa en el dashboard y en la simulación para generar y predecir niveles de inventario.

2 Simulation.py

- **Propósito:** Simular la operación logística en tiempo real usando eventos discretos (simpy).
- **Funciones principales:**
 - `RealTimeLogistics`: clase que simula entregas.
 - `delivery_process`: proceso de entrega para una ruta.
 - `operate_system`: ejecuta la simulación día a día, generando métricas de desempeño.

Uso: Permite evaluar cómo funcionan las rutas y el sistema logístico bajo diferentes escenarios.

3 Dashboard.py

- **Propósito:** Mostrar la información de la logística de forma interactiva usando **Streamlit**.
- **Funciones principales:**
 - Mostrar métricas clave: entregas completadas, distancia total, nivel de servicio, costos.
 - Graficar rutas de entrega en un mapa (Plotly Scattermapbox).
 - Mostrar evolución de inventarios por almacén en gráficos de línea.
 - Permitir al usuario seleccionar el número de días de simulación.

Uso: Interfaz visual para supervisores o gestores logísticos. Es interactivo y permite ver resultados de la simulación.

4 Main.py

- **Propósito:** Ejecutar la simulación completa en consola.
- **Funciones principales:**
 - `run_complete_simulation(days)`: simula la operación logística durante `days` días y devuelve métricas y conclusiones.

Uso: Para probar la simulación sin usar el dashboard. Sirve para debugging o pruebas automáticas.

5 Test.py

- **Propósito:** Probar que la clase `LogisticsOptimizer` funcione correctamente.
- **Funciones principales:**
 - `test_transport`: verifica que la solución de transporte se genere como un diccionario.

Uso: Garantiza que la parte de optimización básica funcione antes de ejecutar simulaciones completas.

Flujo del Proyecto

1. **Inventario:** `inventory.py` genera datos de inventario y predicciones de demanda.
2. **Optimización:** `optimizer.py` (ya existente) calcula rutas y transporte óptimo.
3. **Simulación:** `simulation.py` ejecuta el sistema día a día, registrando entregas y distancias.
4. **Dashboard:** `dashboard.py` permite al usuario visualizar métricas, rutas y niveles de inventario.
5. **Pruebas:** `test.py` asegura que la optimización básica funcione correctamente.
6. **Consola:** `main.py` permite correr simulaciones y obtener resultados en texto.

Conclusión

Este proyecto integra **optimización de rutas, simulación de operaciones y análisis de inventario** en un sistema interactivo. Permite tomar decisiones logísticas basadas en datos y visualizar resultados en tiempo real, facilitando la planificación y reducción de costos.