Challenge Tech Support Analyst - Solvento
Cabrera Ábrego Raúl Andrés

Introducción:

Este documento acompaña el desarrollo de un sistema de planificación de entregas para Solvendo, una tienda de comercio electrónico. El objetivo principal del proyecto es demostrar la habilidad de programación para resolver problemas técnicos comunes en un entorno productivo. El script resultante gestiona la logística de asignación de viajes y camiones, integrando los procesos necesarios para la organización eficiente de las entregas diarias.

Supuestos y Justificaciones

Lenguaje de Programación: Para este challenge, se eligió Python por su simplicidad y familiaridad, así como su sencilla conectividad en Colab.

Base de Datos: La solución presenta una simulación de la conexión a una base de datos utilizando estructuras de datos de Python. Aunque no se integra una base de datos real, se proporcionan ejemplos claros de cómo se realizaría la conexión y la ejecución de consultas, cumpliendo con la restricción de no usar un ORM.

Versionamiento: Se utiliza un repositorio público basado en GIT para el control de versiones, cumpliendo con la práctica estándar de desarrollo y permitiendo un seguimiento transparente del historial de cambios del código.

Frontend: Se decidió implementar una visualización simple de los resultados utilizando DataFrames de pandas y widgets interactivos de IPython en lugar de un frontend tradicional. Esto permite una rápida visualización e interacción dentro del entorno de desarrollo de Jupyter Notebook, cumpliendo con la necesidad de ver los resultados en forma de lista y adaptándose a las restricciones del entorno de ejecución.

Modelado de Clases y Funciones: Las clases y funciones se diseñaron según el modelo de datos proporcionado, con ajustes y extensiones donde fue necesario para garantizar la funcionalidad y eficiencia del sistema. Esto incluye la gestión de los estados de las compras y los viajes, la capacidad de peso de los camiones, y el número de códigos postales permitidos por viaje.

Planificación de Entregas: Para la planificación de viajes y asignación de camiones, se desarrollaron algoritmos que tienen en cuenta los códigos postales, la capacidad máxima de peso, y la disponibilidad de los camiones. La funcionalidad satisface los requerimientos con la flexibilidad de manejar variaciones en la cantidad de paradas y en la capacidad de carga.

Optimización: Aunque se priorizó la funcionalidad y la correctitud del código, se tuvieron en cuenta la optimización de las estructuras de datos en memoria y la

eficiencia algorítmica. La solución busca un equilibrio entre la complejidad y la eficiencia operativa.

Metodología

El proceso de desarrollo siguió una metodología iterativa, comenzando con la definición de estructuras de datos y luego avanzando hacia la implementación de la lógica específica del dominio. Cada etapa fue probada informalmente para asegurar la funcionalidad antes de pasar a la siguiente.

Clases

Item

Representa un artículo en el catálogo de la tienda. Contiene información sobre el ID, nombre, peso y precio del artículo.

Purchase

Refleja una compra hecha por un cliente. Incluye detalles como ID de la compra, nombre del cliente, artículos comprados (junto con la cantidad), fecha de la compra, dirección de entrega, estado de la compra y una fecha de entrega planificada.

Truck

Simboliza un camión de la flota de entrega. Almacena información como el número de matrícula, el peso máximo que puede llevar, la dirección del camión, los días de trabajo y su disponibilidad.

TripState (Enum)

Una enumeración que define los posibles estados de un viaje: planeado, en progreso, completado y cancelado.

Trip

Encapsula un viaje de entrega. Contiene el ID del viaje, los códigos postales a visitar, los camiones asignados, la fecha/hora de salida y el estado actual del viaje.

Address

Define una dirección postal, incluyendo calle, ciudad, código postal y país.

Funciones y Simulación de Base de Datos

get_all_items()

Devuelve una lista de todos los artículos de la base de datos simulada.

get_purchase_items(purchase_id)

Dada una ID de compra, recupera los artículos asociados con esa compra específica.

add_item_to_purchase(purchase_id, item_id, quantity)
Permite agregar un ítem adicional a una compra existente.

plan_trips(purchases, max_stops=3)
Organiza los viajes agrupando las compras por dirección y limitando el número de paradas por viaje.

assign_trucks_to_trips(trips, trucks, items_db)
Asigna camiones a viajes planificados, teniendo en cuenta el peso total y la disponibilidad del camión.

Visualización de Datos
Utiliza pandas para crear DataFrames de los componentes de datos (artículos, compras, camiones, viajes) y los presenta en un formato tabular claro y estructurado.

Widgets de Interacción
show_trip_details(change)
Muestra los detalles de un viaje seleccionado a través de un widget de dropdown, incluyendo los artículos de la compra y el camión asignado.