



El turno de la noche

En Bas nuestra misión es transformar la movilidad de las ciudades, por eso queremos que cada pasajero llegue seguro y a tiempo a casa, sin que el conductor tenga que preocuparse por nada más que manejar, y que el área de operaciones tenga control en todo momento. Para lograrlo necesitamos tu talento para diseñar soluciones cercanas al usuario, que conecten tecnología con las experiencias cotidianas de pasajeros y conductores.

Esta prueba te lleva por el recorrido completo de una operación real en Bas. Queremos que imagines el **momento a momento** que vive un pasajero al finalizar su turno de trabajo, lo que sucede paralelamente en nuestra compañía y cómo tu solución puede responder a esos retos.

Recientemente adquirimos un nuevo cliente que necesita enviar cada noche sus trabajadores desde la oficina hacia sus casas usando transporte corporativo. Cada día los trabajadores pueden cambiar, por eso necesitamos tu ayuda.

El cierre de turno

Los empleados terminan su turno a las 10pm y la empresa debe enviarnos el listado de pasajeros antes de esa hora y así contar con tiempo suficiente para planear la operación. Cada bus desde su patio hasta la sede empresarial tarda en promedio de 45min. Adjunto te compartimos un dataset fake (passengers.csv) que simula lo que enviaría la empresa. Con esta información:

- Diseña el mecanismo de recepción de datos que garantice que recibiremos el listado de pasajeros en nuestro sistema, el punto de llegada. Puede ser un endpoint REST, un script que procesa .CSVs subidos a una carpeta (S3, Firebase Storage, Google Drive) o el mecanismo que consideres más sencillo y escalable cuando sean miles de pasajeros.
- ¿Cuánto tiempo mínimo necesitaría tu sistema para procesar, planear y asignar rutas, considerando un margen de seguridad antes de que los buses deban salir del patio?

Planeación de rutas

Una vez recibimos los datos, debemos empezar a planear la operación. Los buses y vans de nuestros operadores aliados tiene capacidades de: 8, 15, 19, 20 y 40 pasajeros. La meta es que cada ruta sea lo más eficiente posible, reduciendo tiempo y costos, pero asegurando comodidad y puntualidad para los usuarios. Con esta información:

- Agrupa pasajeros en rutas según proximidad de sus destinos. Puedes apoyarte en Google Maps API, OSRM o cualquier librería de clustering. El objetivo es resolver una variante del *Vehicle Routing Problem (VRP)*, asignando pasajeros a buses según capacidad y cercanía.
- Calcula cuántos buses necesitamos para transportar a todos los pasajeros. Aquí buscamos optimizar el uso de la flota, minimizando el número de vehículos sin sobrepasar su capacidad.
- Genera la ruta que seguirá el conductor para el descenso de los pasajeros. Para definir el orden de paradas puedes aplicar enfoques del *Traveling Salesman Problem (TSP)* y apoyarte en algoritmos como Dijkstra o A* para calcular trayectos más cortos entre los puntos.

En camino

Ahora que los buses están en marcha, necesitamos diseñar cómo interactúan pasajeros y conductores con nuestra plataforma. El pasajero debe poder validar que tiene un cupo confirmado en el bus asignado. El conductor debe contar con un listado digital que le muestre qué pasajeros deben abordar y en qué orden dejarlos. Para lograr esto:

- Genera el output de datos (JSON o el de tu preferencia) para la app del pasajero que le permita confirmar su viaje de forma simple. El pasajero debe saber en qué bus se subirá, con un identificador claro (número de bus, placa o código generado por el sistema).
- Genera el output de datos para la app del conductor que le permita visualizar el listado de pasajeros asignados a su bus. Este listado debe estar sincronizado con la agrupación previa (VRP).
- Genera el output de datos para la app del conductor que indique el listado de paradas en el orden calculado previamente. Aquí es clave que la

información venga del algoritmo de enrutamiento (TSP + Dijkstra/A*) para que el conductor solo deba seguir la secuencia y no calcular nada manualmente.

Entregables

1. **Técnico**
 - Código y respuestas para cada uno de los puntos de la prueba
2. **Presentación:** Breve (máximo 5 slides) explicación que cubra:
 - Lógica de la solución y cómo se generan los outputs.
 - Decisiones técnicas (por qué eligió X algoritmo o librería).
3. **Fecha de entrega:** Viernes 5 de septiembre 8pm.

¡Buen viaje! 😊