Prueba técnica desarrollador Backend

Para la empresa Computers S.A se necesita realizar un software para el manejo de pedidos online de productos de tecnología. El gerente de la empresa desea tener un inventario de los productos y quiere almacenar el nombre, descripción, cantidad existente y para los pedidos quiere tener control de cuáles fueron los productos elegidos por el usuario, cantidad de cada producto, dirección entrega y estado del pedido (Realizado, En Bodega, En camino, Entregado, Cancelado). El gerente deja muy en claro las siguientes reglas que se deben cumplir en la solución:

1. Los nombres de los productos no se pueden repetir.
2. No se pueden realizar más de 2 pedidos simultáneamente a una misma dirección.
3. Solo se puede cancelar el pedido si se está en los estados de Realizado o En Bodega.
4. La dirección debe cumplir la siguiente estructura: (CALLE | CARRERA) (#ro) (A-B-C) (#) (#ro) (#ro). Ejemplos: CALLE 54 A # 76 23, CARRERA 90 B # 80 76.
5. Solo se pueden hacer 5 productos por pedido.
6. Al momento de hacer el pedido se debe tener el control del stock de cada producto, dado el caso de que no se cumpla con el stock al realizar el pedido se debe de realizar el respectivo manejo a nivel de estados http.

Usted deberá realizar el modelo de datos y los servicios necesarios para cumplir con los requerimientos previamente descritos. A continuación se listan los entregables y aspectos que se esperan:

1. El código y documentación deberá ser subido a un repositorio de Github.
2. La documentación deberá estar en un archivo README.md.
3. Se deberá documentar la implementación e inicialización del proyecto.
4. En la documentación se deberá evidenciar el modelo de datos en alguna herramienta de modelado como Draw.io.
5. Se deberá tener un archivo donde se ponga el script SQL generador de las tablas.
6. Se deberá utilizar las siguientes tecnologías:
    a. NodeJS con el framework Hapi.js.
    b. PostgreSQL para el gestor de BD.
    c. La documentación de los servicios deberá ser con Swagger utilizando la librería hapi-swagger.
    d. Todo servicio que requiera entrada de datos ya sea en query, params o payload se deberá validar con la librería joi.
    e. Para las consultas a la BD se recomienda utilizar Sequelize (Opcional)
7. A nivel de arquitectura se recomienda utilizar el siguiente repositorio para la estructura del proyecto, si se elige no utilizarla se deberá plantear una arquitectura básica como por ejemplo:
    a. Rutas-Controladores-Casos de Uso-Modelos
    b. Rutas-Controladores-Modelos

    c. Otra

   y se deberá de documentar.
8. Los servicios deben probarse en la herramienta Postman y se deben exportar y subir el archivo generado al repositorio.
9. Es de suma importancia que cada uno de los resultados posibles de cada servicio quede documentado con su respectivo estado http, se puede basar de la documentación oficial de IANA, no hay problema si se utilizan los que no están asignados a nivel de IANA, pero en caso de utilizarlos se debe de documentar en el servicio, todo esta documentación se puede realizar con Swagger.
10. Se recomienda luego de terminado el proyecto y la documentación realizar el levantamiento del proyecto con los pasos descritos por usted en la documentación para comprobar que efectivamente se puede levantar el proyecto.
11. Se debe utilizar la versión 18.13.0 o superior de Nodejs y especificarse en la documentación.

Tiempo máximo de realización: Jueves 7 de septiembre