



## **Reto Hackathon SIIGO**

### **Problema**

Se tiene un grupo de tiendas/empresas que están contenidas en un esquema multitenant, con una o más sucursales, atendiendo servicios de facturación y gestión de inventario.

Las tiendas/empresas requieren que se pueda facturar en sucursales los productos que están en bodega, cada producto debe tener un IVA (19%). El sistema debe ser tolerante a fallos si se cierra la ventana del navegador o se cae la conexión a internet, se debe guardar el borrador de la factura, para una vez restablecido el servicio se pueda continuar facturando.

El sistema debe manejar una gestión de inventarios, actualizando en tiempo real el stock por bodega, y no debe permitir facturar productos que no se encuentran disponibles, también debe notificar un evento a los proveedores informando que un producto está llegando a tope mínimo de unidades en el stock (como sugerencia 10 unidades). Si se realiza una venta de un producto que no está en el stock la solicitud queda encolada, para cuando el stock esté disponible, el sistema genera un evento notificando que la orden/factura se ha procesado e informa a la sucursal que debe hacer la entrega de este, retirándolo de bodega, para posterior entrega (manual) al cliente)

Cuando se desee realizar una factura, se deben escoger los productos mediante una funcionalidad de búsqueda de productos, esta debe contar con un esquema que permita predecir el producto a escoger, y se deben almacenar temporalmente datos de preferencia de búsqueda.

### **Objetivo**

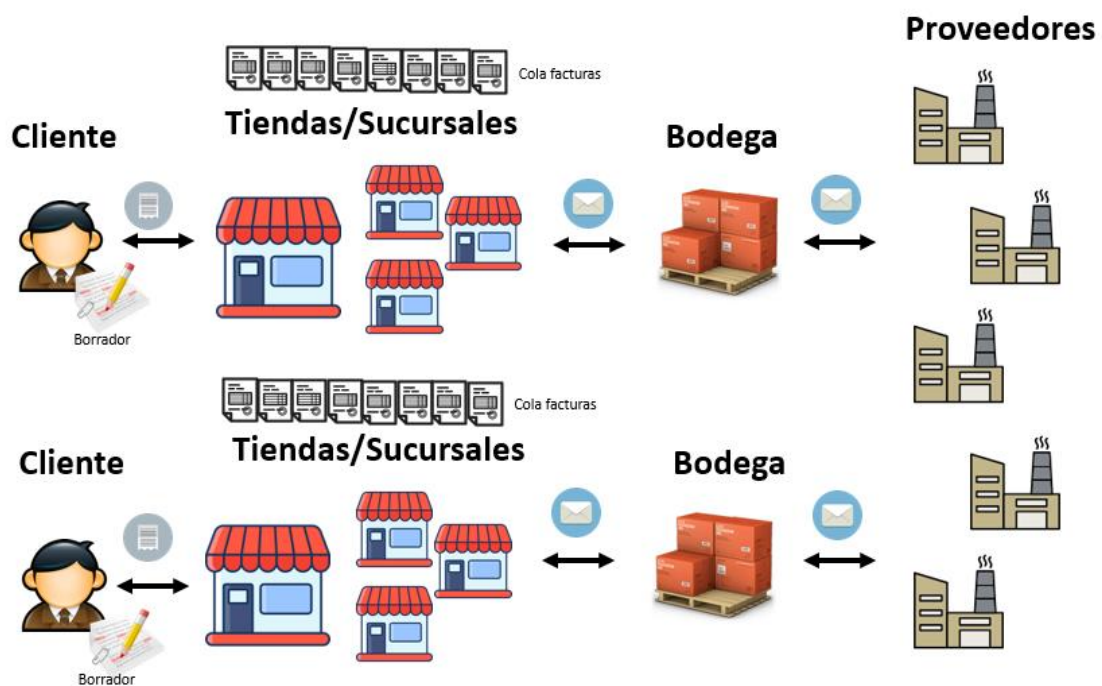
Crear la solución óptima para el problema con una base de arquitectura escalable y funcional, teniendo en cuenta los volúmenes de lectura de inventario y de escritura de facturas. Proponer una base de desarrollo empleando infraestructura en la nube óptima y sostenible.

### **Consideraciones**

- El sistema debe tener una interfaz para crear, revisar las facturas y el stock de los productos
- Las notificaciones deben ser en tiempo real (al momento de generar una orden, o que llegue al límite del stock)
- Se debe poder actualizar y cargar datos de productos desde un archivo Excel (.csv)



- El sistema debe tener la capacidad de recibir un volumen alto de transacciones (inventarios y facturas)



#### Criterios de evaluación

- Modelamiento datos
- Arquitectura y servicios
- UX/UI
- Desarrollo en Backend
- Desarrollo en Frontend
- Solución óptima al problema



- Cumplimiento
- Trabajo en equipo
- Infraestructura –Nube (Plus adicional)

#### CRITERIOS DE DESEMPATE

Para definir el equipo ganador, en caso de empate se tomará en su orden los siguientes criterios:

1. Tiempo de entrega: se verificará el tiempo de entrega en cada una de las jornadas asignando un punto en cada jornada a la pareja que haya entregado primero su prueba.
2. Documentación de líneas de código: si persiste el empate se dará un punto adicional por cada prueba al grupo que utilice comentarios para documentar las líneas de los algoritmos realizados.
3. Documentación técnica: si persiste el empate se dará un punto adicional al grupo que haya realizado en mayor detalle la documentación de análisis para la solución de cada prueba.