

TPI - Procesador de Envíos

El e-commerce CompraloSiPodes actualmente ofrece a sus clientes productos de diferentes rubros, donde cada compra es entregada al cliente final a través de un solo operador logístico (EntregaYa) que tiene cobertura en zona centro del país (Córdoba, San Luis, Santa Fe).

Con el objetivo de expandir los límites de cobertura de entrega de productos, y así mejorar el volumen de ventas, se decidió contratar un nuevo operador logístico que solo opera en zona norte del país (Tucuman, Jujuy, Salta, Catamarca, etc).

Actualmente, las ventas realizadas por día se exportan en un archivo *.csv que se envía al operador logístico para generar las órdenes de envío (No existe API). El cliente no tiene posibilidad de hacer un seguimiento de su producto mientras se encuentra en el operador logístico.

API Procesador de Envíos

Se solicita desarrollar una API Backend que se encargue de generar órdenes de envíos en diferentes operadores logísticos de acuerdo al área de cobertura de cada uno y la dirección destino el envío, y llevar el seguimiento de estado del envío en tiempo real.

- Endpoints a desarrollar.
 - POST /envios: Permite dar el alta de envíos en el sistema y a su vez genera la orden de envío en el operador logístico que corresponda según el área de cobertura.
 - GET /envios/{id_envio}: Consulta de envío creado.
 - POST /envios/{id_envio}/novedades: Alta de novedades de cambio de estado que se van a recibir desde el Operador Logístico.
 - GET/POST/DELETE /operadores: ABM operadores logísticos.
- Datos del Envío (De referencia, cada grupo define su contrato) :
 - Dirección Origen.
 - Dirección Destino.
 - Contacto Comprador.
 - Estado del Envío.
 - Detalle del producto.
 - Datos del Operador Logístico seleccionado para el envío.

Se deben persistir en una base de datos (motor a elección) la info de los envíos y los operadores logísticos con los datos necesarios para invocar a la API según corresponda y las zonas de cobertura de cada uno.

El área de cobertura de cada operador logístico se define en base a la provincia de la dirección destino del envío. (Por lo tanto en la tabla de operadores logísticos debe existir un listado de provincias en las cuales tiene cobertura ese operador).

API Operador Logístico (Centro/Norte):

Se solicita desarrollar una API Backend que permita el alta de órdenes de envíos, la asignación de repartidores para la entrega de productos a destinatarios y el registro de entregas de órdenes de envíos como así la notificación vía Webhook.

- Endpoints a desarrollar
 - POST /ordenes_envio/: Alta orden de envío.
 - GET /ordenes_envio/{orden_envio}: Consulta de la orden de envío.
 - POST /ordenes_envio/{orden_envio}/repartidor: Permite asignar el repartidor disponible.
 - POST /ordenes_envio/{orden_envio}/entrega: Se registra la entrega del producto persistiendo la fecha de entrega. (Esto debe disparar la notificación al Procesador de envíos para que actualice el estado)
 - GET/POST/DELETE /repartidores. ABM Repartidores.
- Datos del Envío (De referencia, cada grupo define su contrato) :
 - Dirección Origen.
 - Dirección Destino.
 - Contacto Comprador.
 - Estado del Envío.
 - Detalle del producto.
 - Datos del repartidor asignado a la entrega del producto.
- Estados de Envío.
 - Creado: Cuando se crea el pedido.
 - En tránsito: Cuando una orden de envío se asigna a un repartidor.
 - Entregado. Cuando un repartidor notifica que el producto se entregó.

Se deben persistir en una base de datos (motor a eleccion) la info de los envíos y de los repartidores.

Consideraciones

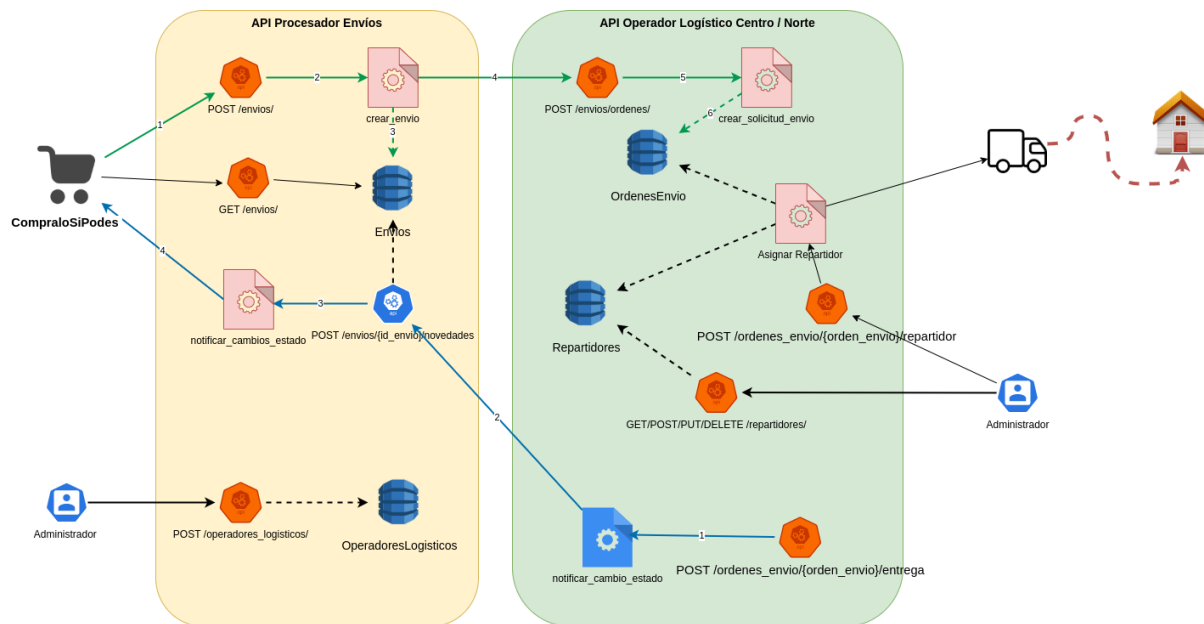
- Formar grupos de no más de 3 personas.
- Cada grupo puede elegir con qué tecnología va a trabajar, pero si debe desplegar su API en las cuentas de AWS.

- Cada grupo debe generar un repositorio, y dar a los profes acceso.
- No vamos a desarrollar FrontEnd, por lo tanto todas las pruebas al momento de la presentación final se realizarán desde Postman.
- Cada grupo debe trabajar con un Mockup de las APIs del resto para poder avanzar.
- Los grupos asignados a Operador Logístico deben compartir sus credenciales OAuth 2.0 al grupo de Procesador de envíos para la presentación final.
- Deben utilizar Auth0 como servidor de autorización.

Para la presentación final cada grupo debe:

- Entregar el Swagger de la API que le corresponda.
- Tener su API desplegada en una cuenta educativa de AWS.
- La API debe implementar autenticación OAuth 2.0. Flujo Client Credentials.
- Realizar la demostración desde un Postman de los flujos planteados.

Diagrama Simplificado de la arquitectura



Calendario checkpoints del TPI

| Fecha | Semana | Día | Tema |
|------------|--------|--------|--|
| 21/10/2021 | 12 | Jueves | Swagger / Open API |
| 28/10/2021 | 13 | Jueves | ABMs (Operador Logístico / Repartidor según corresponda) |
| 4/11/2021 | 14 | Jueves | Alta de Orden de Envío / Webhook |
| 11/11/2021 | 15 | Jueves | Seguridad (OAuth 2.0: ClientCredentials) |
| 18/11/2021 | 16 | Jueves | Presentación Final |

