Practica 3

gRPC y comunicación entre microservicios

PONDERACIÓN: 3

Horas Aproximadas: 8

Universidad San Carlos de Guatemala

Facultad de ingeniería.

Ingeniería en ciencias y sistemas

Contenido

Contexto

La práctica se basa en el proyecto **Delivereats**, una plataforma de delivery diseñada con arquitectura de microservicios.

En esta ocasión se le solicita que realice una demostración de la comunicación entre los microservicios de ordenes y catalogo para garantizar una definición correcta del core de negocio del proyecto.

Objetivos

General

El objetivo de la práctica es Implementar un mecanismo de validación de datos en tiempo real entre microservicios utilizando el protocolo **gRPC**, asegurando que el proceso de creación de órdenes en **Delivereats** cumpla con las reglas de negocio y la consistencia de datos sin acoplamiento de persistencia.

específicos:

Que el estudiante sea capaz de:

- Definir un contrato de servicio mediante **Protocol Buffers (.proto)** para la verificación de catálogos y precios.
- Desarrollar un servidor gRPC en el **Restaurant-Catalog-Service** que exponga métodos de consulta de productos y disponibilidad.
- Implementar un cliente gRPC en el **Order-Service** que actúe como interceptor de lógica antes de la persistencia de una orden.
- Garantizar que cada microservicio gestione su propia base de datos de forma independiente, cumpliendo con la arquitectura de sistemas distribuidos.

Alcance

Para esta práctica, deberás trabajar en la intersección de dos servicios clave definidos en el alcance del proyecto:

Restaurant-Catalog-Service (Servidor gRPC)

- Lógica de Verificación: Un procedimiento que reciba una lista de IDs de productos y un ID de restaurante para validar que:
    - Los productos existen en la base de datos de este servicio.
    - Los productos pertenecen efectivamente al restaurante indicado.
    - Los precios actuales coinciden con los solicitados por el cliente.
- **Base de Datos de Catálogo:** Debe contener los registros de menús y precios actualizados.

Order-Service (Cliente gRPC)

- **Flujo de Creación de Orden:** Modificar el endpoint de "Realizar Orden" para que, antes de guardar en su base de datos, realice una llamada remota al servidor de catálogo.
- **Manejo de Estados de Error:** Si la validación gRPC falla (por precio incorrecto o falta de stock), el servicio debe rechazar la creación de la orden y notificar al frontend.

Contrato de Comunicación

- Definición de mensajes ValidationRequest y ValidationResponse que permitan el intercambio de información estructurada de manera eficiente.

Entregables

Documentación:

- **Bitácora de Pruebas (Logs):** Captura de pantalla o archivo de texto que demuestre:
  - Validaciones exitosas (Orden creada-5).
  - Validaciones fallidas (Orden rechazada por precio o producto inexistente-5).

Funcionalidades:

- **Solicitud de creación de pedido:** El Restaurant-Catalog-Service valida que cada producto solicitado exista en su base de datos y pertenezca realmente al restaurante seleccionado.
- **Verificación de existencia y pertenencia:** El Restaurant-Catalog-Service valida que cada producto solicitado exista en su base de datos y pertenezca realmente al restaurante seleccionado.
- Validación de disponibilidad: El catálogo confirma si los ítems del menú están marcados como disponibles antes de autorizar la continuación del pedido
- **Notificación de error al usuario:** El sistema genera una respuesta clara hacia el cliente indicando el motivo por el cual no se pudo procesar su pedido

Despliegue inicial:

- Archivos .yml configurados para levantar ambos servicios y sus respectivas bases de datos, permitiendo la visibilidad de red necesaria para gRPC

Si el estudiante considera otra solución viable esta debe ser documentada y justificada

Consideraciones

- Se debe hacer uso de un repositorio en la nube para realizar la entrega de su práctica (Gitlab, Github, Bitbucket, etc.) con el nombre SA_PRACTICAS_CARNET y con una carpeta con el título PRACTICA3, se recomienda trabajar en el repositorio del proyecto y al finalizar crear un tag con el siguiente nombre v0.3.0 y trasladar el trabajo realizado a la carpeta PRACTICA3 del repositorio de practicas.
- La practica se debe presentar de forma local **(NO DESPLIEGUES EN NUBE)**
- Agregar al auxiliar al repositorio de GitHub cómo colaborador: Samashoas
- **Se trabajará de manera individual.**
- Las copias completas/parciales serán merecedoras de una nota de 0 puntos, los responsables serán reportados al catedrático de la sección y a la Escuela de Ciencias y Sistemas.
- La práctica tiene una nota aprobatoria de 61 nota mínima para optar a trabajar en parejas y **es requisito entregarla y calificarla para tener derecho a calificación de la fase 1.**

Cronograma

Tarea	Fecha
Asignacion practica	05/02/2026
Fecha de entrega	11/02/2026
Fecha de calificación	12/02/2026, 14/02/2026

Rúbrica de Calificación

Criterio	Descripción	Valor
Orquestación Docker	Uso de archivos .yml para levantar ambos servicios y sus bases de datos independientes.	10
Aislamiento de Persistencia	Verificación de que cada microservicio maneja su propia base de datos (Database-per-service) sin acceso directo a tablas ajenas.	10
Definición de Contrato	Implementación correcta de mensajes en archivos .proto	10
Acoplamiento de Contrato	Definición precisa de tipos de datos para garantizar que ambos servicios se comuniquen sin errores de tipos.	10
Consistencia de Datos entre Servicios	Garantía de que la orden solo se crea si el catálogo confirma la validez de los datos; si no hay respuesta o es negativa, la orden no debe persistirse.	10
Existencia y Pertenencia	El sistema valida que los productos existan y correspondan al restaurante seleccionado.	10
Disponibilidad	Confirmación de que los ítems están marcados como activos/disponibles antes de procesar el pedido.	10
Respuesta al Usuario	Notificación clara al cliente sobre el motivo específico del rechazo (error de precio, stock o producto inexistente).	10
Logs: Éxito	Evidencia de una orden creada tras validación exitosa (mínimo 5 ejemplos).	7.5

Logs: Fallo	Evidencia de órdenes rechazadas por inconsistencias detectadas (mínimo 5 ejemplos).	7.5
Preguntas	**Documentación**	5

Valores

En el desarrollo de la práctica, se espera que cada estudiante demuestre honestidad académica y profesionalismo. Por lo tanto, se establecen los siguientes principios:

1. **Originalidad del Trabajo**
    - Cada estudiante o equipo debe desarrollar su propio código y/o documentación, aplicando los conocimientos adquiridos en el curso.
2. **Prohibición de Copias y Plagio**
    - Si se detecta la copia total o parcial del código, documentación o cualquier otro entregable, la calificación será de **0 puntos**.
    - Esto incluye la reproducción de código entre compañeros, la reutilización de proyectos de semestres anteriores o el uso de código externo sin la debida referencia.
3. **Uso Responsable de Recursos Externos**
    - El uso de bibliotecas, frameworks y ejemplos de código externos está permitido, siempre y cuando se referencian correctamente y se comprendan plenamente. (Consultar con el catedrático su política)
4. **Revisión y Detección de Plagio**
    - Se podrán utilizar herramientas automatizadas y revisiones manuales para identificar similitudes en los proyectos.
    - En caso de sospecha, el estudiante deberá justificar su código y demostrar su desarrollo individual o en equipo. Si este extremo no es comprobable la calificación será de **0 puntos**.

Al detectarse estos aspectos se informará al catedrático del curso quien realizará las acciones que considere oportunas.