

## Practica 4

# Actualización de Documentación y testeo de colas

---



**PONDERACIÓN: 3**

**Horas Aproximadas: 8**

Universidad San Carlos de Guatemala

Facultad de ingeniería.

## Ingeniería en ciencias y sistemas

## Contenido

Diseño de Arquitectura y Base de Datos para <i>Delivereats</i> .....	<b>¡Error! Marcador no definido.</b>
Contexto.....	2
Objetivos.....	2
General.....	2
específicos:.....	2
Alcance.....	2
Entregables.....	2
Documentación:.....	2
Funcionalidades:.....	3
Consideraciones.....	3
Cronograma .....	3
Rúbrica de Calificación .....	4
Valores.....	5
1. Originalidad del Trabajo .....	5
2. Prohibición de Copias y Plagio .....	5
3. Uso Responsable de Recursos Externos.....	5
4. Revisión y Detección de Plagio.....	5

## Contexto

La práctica se basa en la evolución del proyecto Delivereats hacia su Fase 2, migrando de una arquitectura básica hacia un sistema resiliente y escalable. En esta práctica el enfoque es preparar la documentación de la nueva infraestructura (Kubernetes, CI/CD, Bases de Datos actualizadas), actualizar la documentación general de la aplicación y realizar una Prueba de Concepto funcional de la mensajería asíncrona utilizando RabbitMQ o Kafka antes de implementar toda la lógica de negocio final.

## Objetivos

### General

El objetivo de la práctica es que el estudiante aplique de manera efectiva metodologías ágiles para la planificación y construcción de una arquitectura distribuida mediante la entrega continua de incrementos. A través de este enfoque, se busca que el estudiante desarrolle la capacidad de planificar y validar componentes críticos (como colas de mensajería) en pequeños ciclos, garantizando una transición segura hacia la Fase 2 del proyecto.

### específicos:

En esta práctica el equipo (pareja) debe entregar:

- Proponer un diagrama de despliegue enfocado en un clúster de Kubernetes.
- Actualizar la arquitectura a nivel de microservicios, comunicación y responsabilidades.
- Actualizar el esquema de la base de datos de la aplicación
- Implementar un canal de comunicación asíncrona básico entre dos microservicios existentes.

## Alcance

En esta práctica el equipo/estudiante debe entregar:

- **Documentación exhaustiva de los nuevos flujos asíncronos y la infraestructura.**
- **Código funcional mínimo:** El Order-Service y el Restaurant-Service deben ser capaces de enviarse un mensaje básico a través de una cola, sin necesidad de procesar la lógica de negocio completa de la orden aún.

## Entregables

### Documentación:

- Actualización de Requerimientos Funcionales
- Actualización de Requerimientos NO funcionales
- Actualización del Diagrama de Arquitectura de alto nivel
- Actualización del Diagrama de despliegue
- Actualización del Esquema de la base de datos
- Actualización del Diagrama de actividades
- Guía de despliegue en Kubernetes (paso a paso).
- Descripción del pipeline CI/CD a implementar y de las variables/secrets que se necesitarán.

- Descripción del flujo JWT (con ejemplos de peticiones si es posible).
- Estrategias de Rollout y Rollback a utilizar

#### Funcionalidades:

- Prueba de Concepto (PoC) RabbitMQ/Kafka:
  - *Productor*: Order-Service publica exitosamente un mensaje en la cola al crear una orden.
  - *Consumidor*: Restaurant-Service consume el mensaje de la cola y lo registra/imprime en consola.

## Consideraciones

- Se debe hacer uso del repositorio en la nube utilizado en la Fase 1. Una vez finalizada la práctica, el equipo deberá de agregar el trabajo realizado al repositorio del proyecto y crear un tag con el nombre **v1.1.0**.
- Agregar al auxiliar al repositorio de GitHub como colaborador: Samashoas
- **Se trabajará en PAREJAS. Ambos estudiantes deben evidenciar commits en el repositorio que demuestren un trabajo colaborativo y equitativo.**
- 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 cuartetos y **es requisito entregarla y calificarla para tener derecho a calificación de la fase 2.**

## Cronograma

Tarea	Fecha
Asignacion practica	19/02/2026
Fecha de entrega	25/02/2026
Fecha de calificación	26/01/2026

## Rúbrica de Calificación

La evaluación de la práctica busca medir el cumplimiento de los objetivos planteados, así como la correcta aplicación de los conocimientos técnicos y habilidades.

Descripción	Valor	Punteo
<b>Actualización de requerimientos funcionales y no funcionales</b>	7.5	
<b>Diagrama de despliegue (K8s)</b>	7.5	
<b>Diagrama Entidad Relación actualizado</b>	10	
<b>Diagramas de actividades / secuencia actualizados</b>	10	
<b>Guía de despliegue en K8s (paso a paso) y Estrategias de Rollout/Rollback</b>	10	
<b>Descripción del pipeline CI/CD (con variables/secrets) y Descripción del flujo JWT</b>	10	
<b>Productor: Order-Service publica mensaje en la cola exitosamente</b>	15	
<b>Consumidor: Restaurant-Service consume y registra el mensaje en consola</b>	15	
<b>Creación del tag v0.4.0 y uso colaborativo del repositorio</b>	7.5	
<b>Preguntas</b>	7.5	

## 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 mencionen 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.