

Microservicios y Docker

Implementacion Basica de Microservicios y Docker



PONDERACIÓN: 3

Horas Aproximadas: 12

Universidad San Carlos de Guatemala

Facultad de ingeniería.

Ingeniería en ciencias y sistemas

Índice

Índice	1
Competencias	2
Objetivos	2
General	2
Específicos	2
Descripción / Enunciado	2
Entregables	3
Consideraciones	3
Cronograma	3
Ejemplo	3
Evaluación	4
Valores	4
Referencias	4

Competencias

Capacidad para desarrollar, contenerizar, desplegar y mantener microservicios utilizando Docker, asegurando portabilidad, escalabilidad, aislamiento y eficiencia en ambientes de desarrollo, prueba y producción.

General

- Aplicar los conocimientos adquiridos a lo largo de la carrera de Ingeniería en Ciencias y Sistemas para generar software de alta calidad y escalable, a través de diferentes técnicas de desarrollo y utilizando las últimas tecnologías.

Específicos

- Que el estudiante aprenda a familiarizarse con la arquitectura de microservicios.
- Que el estudiante aprenda a construir los microservicios a través de docker.
- Que el estudiante aprenda a configurar su entorno de desarrollo.

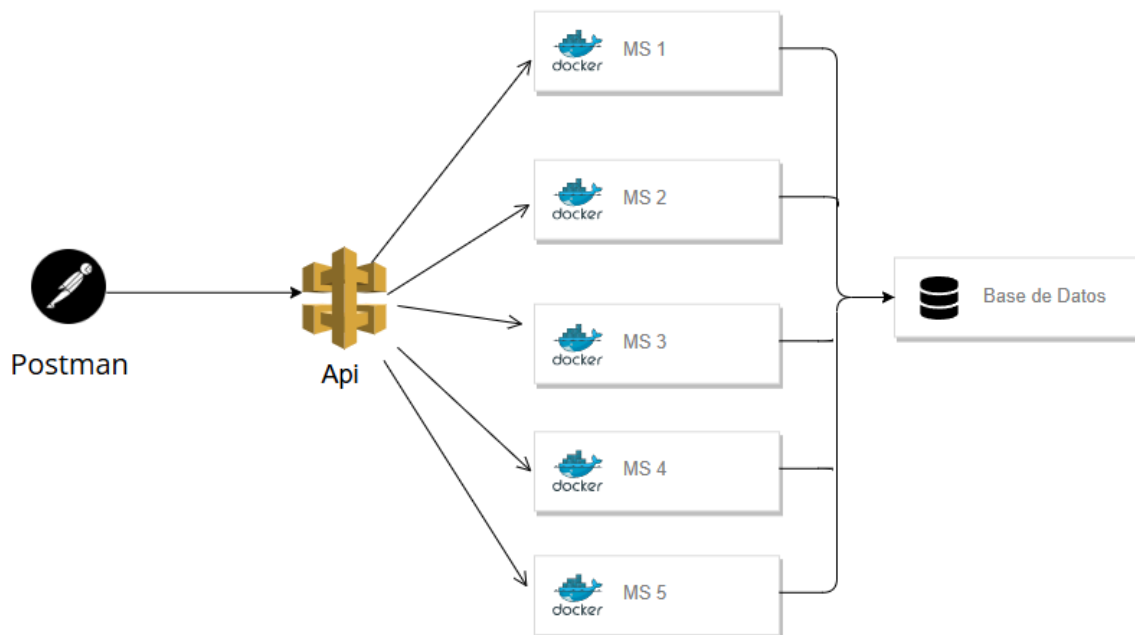
Enunciado

Se solicita que el estudiante realice la descripción de un proyecto con la temática de su elección; para dicho proyecto debe de utilizar la arquitectura basada en microservicios.

Debe de ser mínimo 4 microservicios.

Asi mismo, basándose en el diseño anterior, deberá de realizar lo siguiente:

- Implementar una versión básica de cada microservicio que incluya al menos un endpoint funcional.
- Configurar docker para cada microservicio.
- Debera de implementar un servicio como simulador de api gateway
- **Al menos dos microservicios deberán de implementar GraphQL**
- **Al menos dos lenguajes de programación distintos deberán de ser utilizados**



Nota: La imagen es plenamente ilustrativa. Recuerde seguir las estrategias y cumplir con las reglas de diseño de microservicios, así como hacer uso de las herramientas recomendadas para ello.

Entregables

Subir a UEDI el enlace del repositorio.

Consideraciones

- Se debe hacer uso de un repositorio en la nube para realizar la entrega de su proyecto (Gitlab, Github, Bitbucket, etc.)

- El lenguaje a utilizar es de elección libre según lo requiera el estudiante. Aun así, se recomienda el uso de javascript/typescript o python.
- 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.
 - Nombre del Repositorio: **Practicas-SA-<<SECCIÓN>>-<<CARNE>>** o
Crear carpeta dentro del repositorio con el nombre **P3** e incluir la documentación dentro de ella.
 - Agregar al auxiliar al repositorio, con el rol Developer:
 - o Sección A: **hkjvasquez**
 - o Sección B: **di3gini**

Cronograma

El cronograma permite planificar y organizar las diferentes etapas del desarrollo de la práctica

Ejemplo

Tarea	Fecha
Asignación práctica	15 septiembre 2025
Fecha de entrega	20 septiembre 2025
Fecha de calificación	21 septiembre 2025

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.

Se debe agregar una tabla con los aspectos a calificar con su respectiva puntuación.

Documentación	—	—	—
Contratos de Microservicios	10		
Dockerfiles optimizados	5		
Diagrama de Arquitectura	5		
Diagrama ER	5		
Documentación de GraphQL (que es, como utilizarlo, ejemplos, etc)	5		
Aplicación	—	—	—
Endpoints Microservicio 1	10		
Endpoints Microservicio 2	10		
Endpoints Microservicio 3 (GraphQL)	10		
Endpoints Microservicio 4 (GraphQL)	10		
Correcta implementación de arquitectura basada en microservicios	5		
Simulador de Api Gateway	5		
Comunicación correcta entre deploys de docker	10		
Preguntas	—	—	—
Pregunta 1	5		
Pregunta 2	5		
Punteo	—	—	—
Total	100		

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.

Referencias

[Utilización de Docker para la arquitectura de microservicios | AppMaster](#)