


Ejercicios de aplicación

Rol del arquitecto en equipos ágiles

Es hora de que pongas en práctica todo lo aprendido. 

Este apartado tiene el objetivo de ayudarte a seguir potenciando tus habilidades, por lo que a continuación encontrarás diferentes **desafíos** que podrás resolver de forma independiente y a tu ritmo.

Más adelante conseguirás las resoluciones para que valides tus respuestas y puedas monitorear tu progreso. 

¡Manos a la obra!

1. Desafío

Consigna: Diseña un plan de arquitectura adaptable para un sistema de mensajería en tiempo real que pueda responder a los cambios constantes del entorno ágil. Define las responsabilidades del arquitecto en cada sprint y crea un enfoque de colaboración con el equipo de desarrollo para asegurar que el diseño arquitectónico sea flexible y sostenible.

Pasos:

1. Define las responsabilidades del arquitecto en equipos ágiles, destacando cómo debe colaborar con los desarrolladores y otros stakeholders en cada sprint.
2. Esboza una estrategia de arquitectura para un sistema de mensajería en tiempo real que incorpore principios de modularidad y escalabilidad.
3. Desarrolla un plan de integración continua, explicando cómo el arquitecto puede supervisar la evolución de la arquitectura sin interrumpir la adaptabilidad del equipo.

4. ¿Dónde se lleva a cabo?

Puedes realizar este desafío en una herramienta de diagramación como **Lucidchart** o **Miro** para visualizar la arquitectura y el plan de integración. Documenta el plan de arquitectura y el rol del arquitecto en un archivo de texto, utilizando **Word**, **Google Docs**, o **Notion**.

5. Tiempo de dedicación

Ejercicios de aplicación

Rol del arquitecto en equipos ágiles

2 Horas.

6. Recursos

- Consulta el **manual del módulo** sobre el rol del arquitecto en entornos ágiles.
- Documentación del **Manifiesto Ágil** y guías sobre integración continua en sistemas de arquitectura modular.
- Opcional: Accede a recursos en línea sobre patrones de arquitectura escalable para sistemas de mensajería en tiempo real y sobre cómo implementar planes de integración continua.

7. Plus

Opcional: Para profundizar en el desafío, agrega una propuesta para implementar monitoreo y automatización en el sistema de mensajería. Puedes incluir métricas clave que el arquitecto debe revisar en cada sprint para asegurar que el sistema sea eficiente y confiable.

8. Condición

Esta práctica o ejercitación **no requiere ser entregada y/o evaluada** por el mentor. No obstante puedes compartir tus resultados con el resto de los bootcampers y construir conocimiento en conjunto.

9. Resolución del ejercicio:

Paso 1: Definición de las Responsabilidades del Arquitecto en Equipos Ágiles

En un entorno ágil, el arquitecto debe asumir un rol de facilitador y mentor, apoyando al equipo y colaborando en cada sprint. Sus responsabilidades principales incluyen:

1. **Colaboración y Comunicación:** El arquitecto debe trabajar de cerca con el equipo de desarrollo y otros stakeholders para definir y ajustar la arquitectura según las necesidades del cliente y los cambios en los requisitos. Esto implica participar en las reuniones de planificación de sprint y en las retrospectivas, así como comunicarse con los desarrolladores para entender sus desafíos y necesidades.

2. **Facilitación de la Adaptabilidad Arquitectónica:** En cada sprint, el arquitecto evalúa cómo la arquitectura puede adaptarse para soportar nuevas funcionalidades o cambios sin afectar la estabilidad del sistema. Para un sistema de mensajería en tiempo real, esto incluye asegurar que la arquitectura permita la escalabilidad y el manejo eficiente de múltiples usuarios concurrentes.
3. **Supervisión del Cumplimiento de Atributos de Calidad:** El arquitecto debe definir los atributos de calidad clave para el sistema, como rendimiento, seguridad y usabilidad, y verificar su cumplimiento en cada sprint. Esto implica establecer métricas de rendimiento y realizar revisiones de código y diseño para asegurar que el sistema cumpla con los estándares de calidad.
4. **Apoyo en la Implementación de Integración Continua (CI):** El arquitecto supervisa la implementación de integración continua, garantizando que el código pueda ser probado y desplegado rápidamente sin interrumpir el flujo de trabajo del equipo. La CI permite que el sistema de mensajería evolucione con seguridad y eficiencia en cada sprint.
5. **Documentación de la Evolución Arquitectónica:** Aunque en un entorno ágil la documentación tiende a ser ligera, el arquitecto debe documentar cambios clave en la arquitectura para que el equipo mantenga una referencia clara y consistente del diseño, lo cual es útil para futuras iteraciones y para nuevos miembros del equipo.

Paso 2: Estrategia Arquitectónica para un Sistema de Mensajería en Tiempo Real

Arquitectura Modular y Escalable: El sistema de mensajería debe estar basado en microservicios para permitir una escalabilidad flexible y facilitar el mantenimiento de cada componente. Los microservicios principales incluyen:

- **Servicio de Autenticación y Autorización:** Gestiona el acceso seguro al sistema, asegurando que solo usuarios autenticados puedan acceder y enviar mensajes.
- **Servicio de Gestión de Usuarios:** Almacena y gestiona los perfiles y contactos de los usuarios, permitiendo la búsqueda y el agregado de

Ejercicios de aplicación

Rol del arquitecto en equipos ágiles

nuevos contactos.

- **Servicio de Mensajería en Tiempo Real:** Administra la comunicación en tiempo real entre los usuarios y maneja la entrega de mensajes, manteniendo una conexión constante mediante protocolos como WebSockets.
- **Servicio de Notificaciones:** Envía alertas y notificaciones push cuando llegan nuevos mensajes o se producen eventos importantes en la plataforma.
- **Servicio de Almacenamiento de Mensajes:** Almacena los historiales de mensajes, permitiendo a los usuarios consultar sus conversaciones previas. Utiliza una base de datos NoSQL para optimizar la recuperación rápida de datos.

Orquestación y Escalabilidad:

- Cada microservicio se ejecuta en contenedores Docker, lo que permite una fácil escalabilidad y orquestación con Kubernetes. Kubernetes supervisa la carga en cada servicio y ajusta los recursos según sea necesario.
- **Balanceadores de Carga** distribuyen las solicitudes entre los microservicios para asegurar que el sistema pueda manejar un alto número de usuarios concurrentes sin afectar el rendimiento.

Interfaz de Usuario y Experiencia de Usuario (Frontend):

- Una aplicación web responsiva en React permite una interfaz rápida y amigable para que los usuarios envíen mensajes, revisen contactos y reciban notificaciones en tiempo real.

Paso 3: Plan de Integración Continua (CI) y Supervisión

La implementación de un plan de integración continua es esencial para asegurar que los cambios en el código se integren y prueben de manera rápida y eficiente, permitiendo que el equipo adapte la arquitectura a los cambios de manera ágil.

1. Integración Continua (CI):

- **Automatización de Pruebas:** Cada cambio de código se somete a pruebas automáticas para verificar que no afecte la

estabilidad del sistema. Estas pruebas incluyen pruebas unitarias para cada microservicio y pruebas de integración para asegurar que los servicios interactúan correctamente.

- **Despliegue Automatizado:** El sistema utiliza pipelines automatizados para desplegar cambios en un entorno de prueba. Esto permite al equipo detectar y corregir errores antes de que lleguen al entorno de producción.

2. Supervisión y Monitoreo de Rendimiento:

- El arquitecto implementa herramientas de monitoreo (por ejemplo, Prometheus o New Relic) para supervisar el rendimiento en tiempo real. Estas herramientas rastrean el tiempo de respuesta de cada microservicio, el uso de recursos y la estabilidad de las conexiones en tiempo real.
- **Alertas Automatizadas:** En caso de que se detecten problemas de rendimiento o seguridad, el sistema envía alertas automáticas al equipo, permitiéndoles responder rápidamente a cualquier incidencia.

3. Revisión Continua de la Arquitectura:

- En cada sprint, el arquitecto revisa la arquitectura en conjunto con el equipo para identificar posibles mejoras o ajustes que respondan a los nuevos requisitos. Este proceso asegura que la arquitectura evolucione y se mantenga alineada con los objetivos del proyecto y la metodología ágil.

Resolución Completa de la Actividad

Comparación con Ejemplo Esperado:

1. **Responsabilidades del Arquitecto en Equipos Ágiles:** El arquitecto colabora activamente con el equipo, facilita la adaptabilidad de la arquitectura y asegura que la calidad y el rendimiento se mantengan.
2. **Estrategia Modular y Escalable:** La arquitectura de microservicios permite que el sistema sea flexible y responda a una alta demanda de usuarios en tiempo real.
3. **Plan de Integración Continua:** La CI facilita la implementación y el monitoreo continuo, asegurando que el sistema sea estable y se adapte a cambios con cada sprint.

Ejercicios de aplicación 🧐

Rol del arquitecto en equipos ágiles

Usa esta resolución como guía para comparar tus resultados y asegurar que tu enfoque arquitectónico esté alineado con los principios ágiles de flexibilidad, colaboración y mejora continua.