

1. ¿Por qué surgió la necesidad de DevOps en el desarrollo de software?

- surgió como respuesta a problemáticas en los ciclos tradicionales de desarrollo de software.
- barreras entre los equipos
- La detección tardía de errores provocaba retrabajos costosos y retrasos en la entrega.
- Por que se requería una metodología que fuera mas rápida, eficiente y exista una mejor integración entre los equipos de desarrollo

2. Explica cómo la falta de comunicación y coordinación entre los equipos de desarrollo y operaciones en el pasado llevó a la creación de DevOps.

- En los modelos tradicionales, los equipos trabajaban de forma aislada, lo que resultaba en un flujo secuencial donde los errores se detectan tarde, al final del proceso, lo que prolongaba el ciclo de desarrollo

3. Describe cómo el principio de mejora continua impacta tanto en los aspectos técnicos como en los culturales de una organización.

- Estas prácticas permiten a los equipos identificar de forma temprana obstáculos, ajustar prioridades y mantener una comunicación constante.
- Genera una cultura orientada a la colaboración y a la responsabilidad compartida no solo mejora la eficiencia operativa, sino que también crea un ambiente de trabajo más motivador e innovador.

4. ¿Qué significa que DevOps no se trata solo de herramientas, individuos o procesos?

- DevOps es una metodología integral que se trata de una Fomentar la comunicación, la transparencia y la responsabilidad compartida entre todos los involucrados

5. Según el texto, ¿cómo contribuyen los equipos autónomos y multifuncionales a una implementación exitosa de DevOps?

los equipos multifuncionales facilita la comunicación entre equipos autónomos aprovechando su conocimiento en ambas áreas promoviendo el trabajo colaborativo

1. ¿Qué significa "desplazar a la izquierda" en el contexto de DevSecOps y por qué es importante?

- Antes implementar la seguridad era algo que se implementaba al final del ciclo del desarrollo, sin embargo al tener el enfoque de desplazar a la izquierda se implementa la seguridad durante el ci/cd pipeline. De esta manera se implementa la seguridad en cada nueva versión del proyecto.

2. Explica cómo IaC mejora la consistencia y escalabilidad en la gestión de infraestructuras.

al automatizar el despliegue mediante scripts evitamos errores manuales

3. ¿Cuál es la diferencia entre monitoreo y observabilidad? ¿Por qué es crucial la observabilidad en sistemas complejos?

La observabilidad combina tres pilares principales, uno de ellos es el monitoreo que recolección de métricas del sistema evaluar el estado de salud de la aplicación. observabilidad para detectar problemas antes de que afecten a los usuarios finales.

4. ¿Cómo puede la experiencia del desarrollador impactar el éxito de DevOps en una organización?

- La experiencia del desarrollador puede influir en términos de la calidad de desarrollo que realice (buenas prácticas), además de su experiencia trabajando en equipo mejora la comunicación durante el ciclo de desarrollo.

5. Describe cómo InnerSource puede ayudar a reducir silos dentro de una organización.

InnerSource aplica las prácticas del código abierto dentro de una empresa, permitiendo que los equipos colaboren más allá de sus departamentos. En lugar de trabajar de manera aislada, los desarrolladores pueden acceder al código, sugerir mejoras y compartir soluciones con otros equipos

6. ¿Qué rol juega la ingeniería de plataformas en mejorar la eficiencia y la experiencia del desarrollador?

herramientas, entornos y automatizaciones que mejora el flujo de trabajo, estandariza procesos y reducir tareas repetitivas