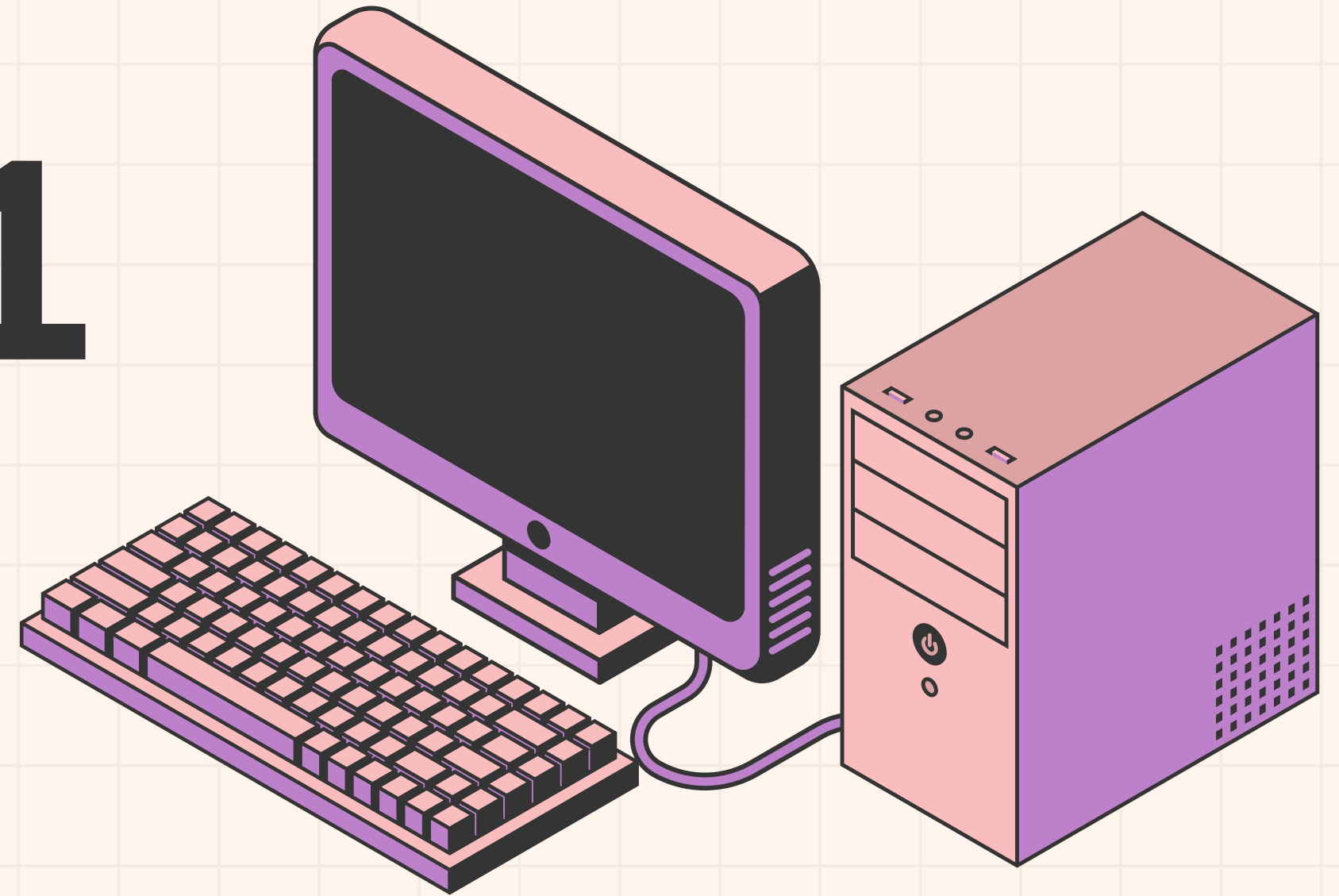


ACTIVIDAD 1

INTRODUCCIÓN A DEVOPS

Integrantes:

- Chowdhury Gomez, Junal Johir
- La Torre Vasquez, Andres Sebastian
- Zapata Inga, Janio Adolfo



Preguntas de reflexión parte 1

¿Por que surgio la necesidad de DevOps en el desarrollo de software?



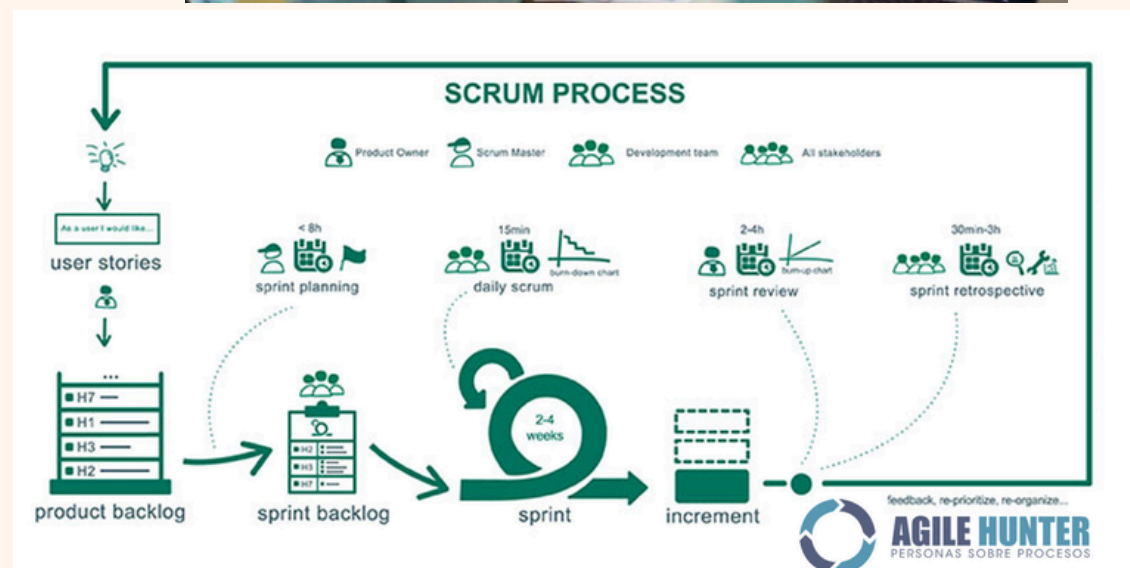
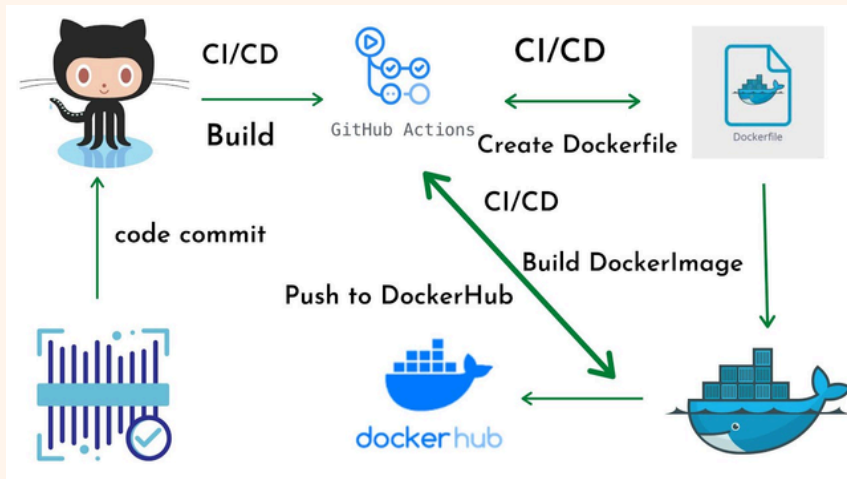
- Se genero una brecha entre los equipos de desarrollo y operaciones:
 - Desarrollo buscaba rapidez en la entrega de nuevas funciones
 - Operaciones priorizaba la estabilidad y seguridad del sistema
- Esta fricción causaba retrasos en los lanzamientos y problemas al implementar software
- DevOps surgió para integrar ambos equipos, reducir fricciones y permitir entregas mas rápidas y confiables

Explica como la falta de comunicacion y coordinacion entre los equipos de desarrollo y operaciones en el pasado llevo a la creacion de DevOps



- Desarrollo y operaciones trabajaban sin comunicación efectiva
- Los desarrolladores entregaban código sin considerar el entorno de producción
- El equipo de operaciones recibía software incompleto o difícil de implementar
- Esto provocaba:
 - Problemas en producción
 - Desgaste y frustración en los equipos
 - Procesos lentos e ineficientes
- DevOps nació para romper estos silos, fomentar la colaboración y mejorar la integración entre ambos equipos

¿Que significa que DevOps no se trata solo de herramientas, individuos o procesos?



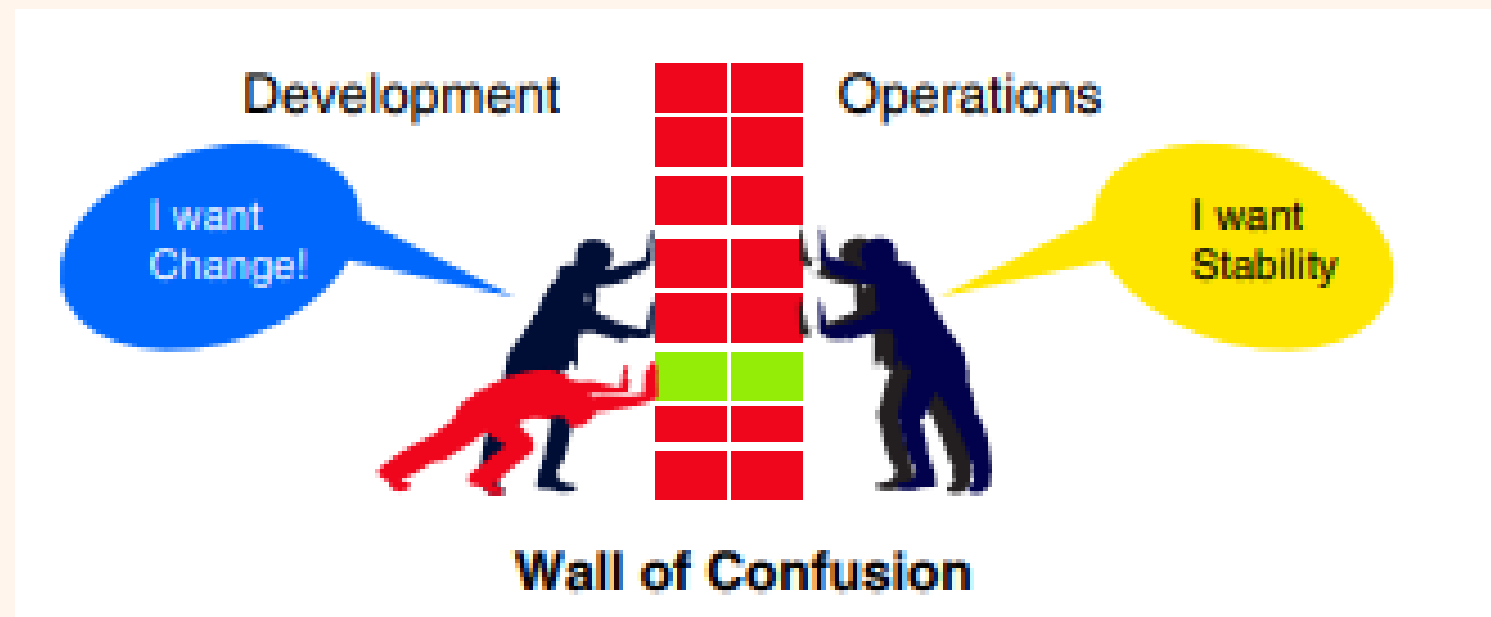
- no basta con usar tecnologías como docker, github o CI/CD
- no se trata solo de contratar o formar "equipos DevOps"
- no es simplemente adoptar procesos agiles como Scrum
- DevOps requiere:
 - Un cambio cultural dentro de la organización
 - Colaboración efectiva entre desarrollo y operaciones
 - Automatización estratégica para optimizar procesos sin generar burocracia innecesaria
- sin estos elementos, las herramientas y metodologías pueden generar mas carga operativa en lugar de mejorar la eficiencia

Describe como el principio de mejora continua impacta tanto en los aspectos tecnicos como en los culturales de una organizacion

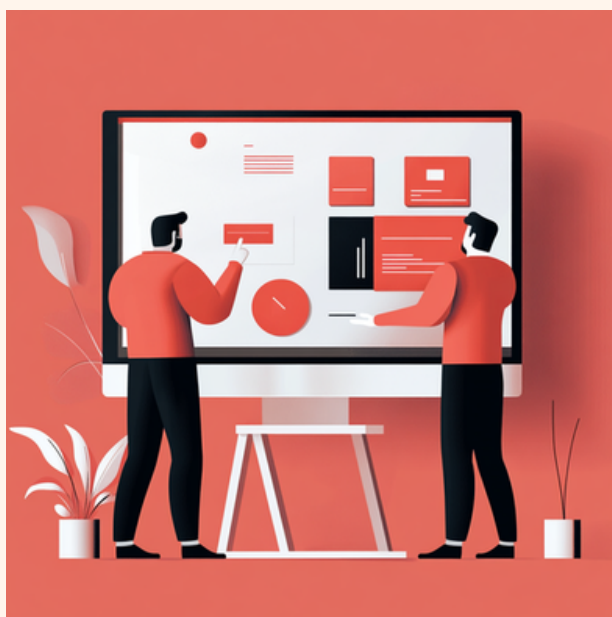


- **Impacto técnico:**
 - Mejora la confiabilidad, adaptabilidad y eficiencia del software
 - Optimiza procesos de entrega y monitoreo de métricas clave
 - Permite ajustes rápidos basados en datos de rendimiento
- **Impacto cultural:**
 - Fomenta la colaboración entre equipos
 - Impulsa una cultura de aprendizaje y experimentación
 - Ayuda a desmantelar silos organizacionales

Segun el texto, ¿como contribuyen los equipos autonomos y multifuncionales a una implementacion exitosa de DevOps?



- Eliminar barreras entre desarrollo, operaciones y QA
- Formar equipos con habilidades diversas como desarrollo, pruebas, despliegue, diseño
- Agilizar la toma de decisiones sin depender de múltiples aprobaciones
- Evitar cuellos de botella y tiempos de espera en la entrega de software
- Fomentar la responsabilidad y la colaboración dentro del equipo
- Permitir que el software se desarrolle, implemente y mantenga de manera eficiente y sin fricciones

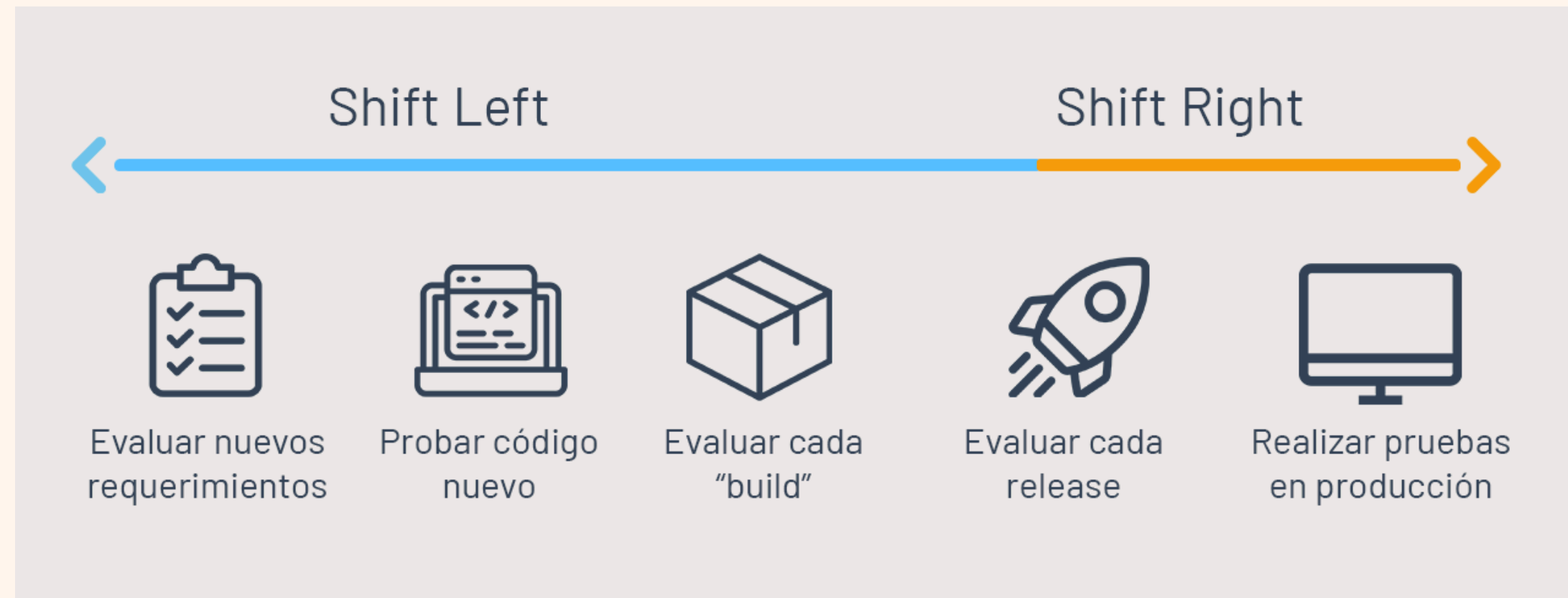


Preguntas de reflexion parte 2

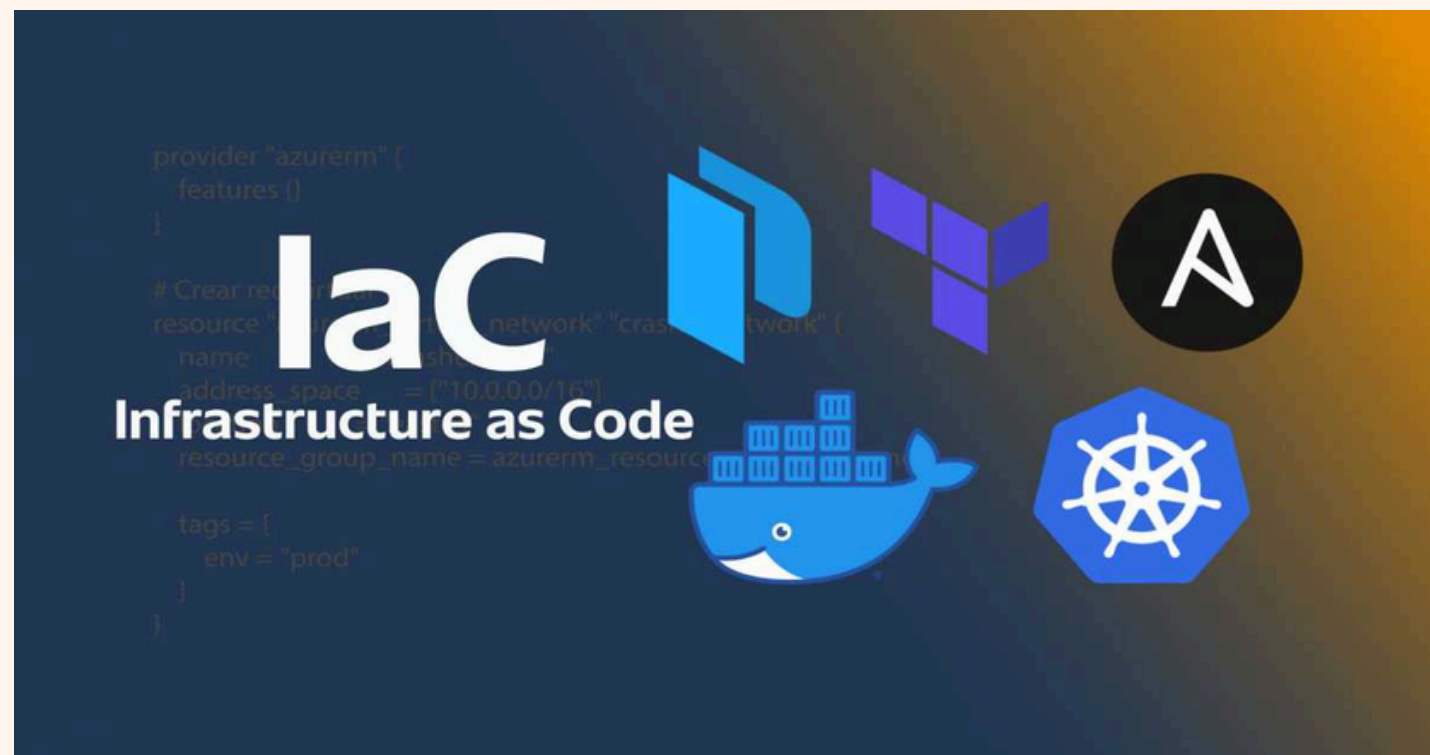
1.¿Qué significa "desplazar a la izquierda" en el contexto de DevSecOps y por qué es importante?

Es integrar la seguridad desde las primeras etapas del desarrollo en lugar de tratarla como un paso final antes del despliegue. Esto es importante porque:

- Permite identificar y corregir vulnerabilidades desde el inicio.
- Reduce costos y tiempo de corrección de errores de seguridad.
- Fomenta una cultura de seguridad en el equipo de desarrollo.



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



Permite gestionar la infraestructura mediante código en lugar de configuraciones manuales. Esto mejora la consistencia y escalabilidad ya sea con la automatización, consistencia, escalabilidad y trazabilidad.

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

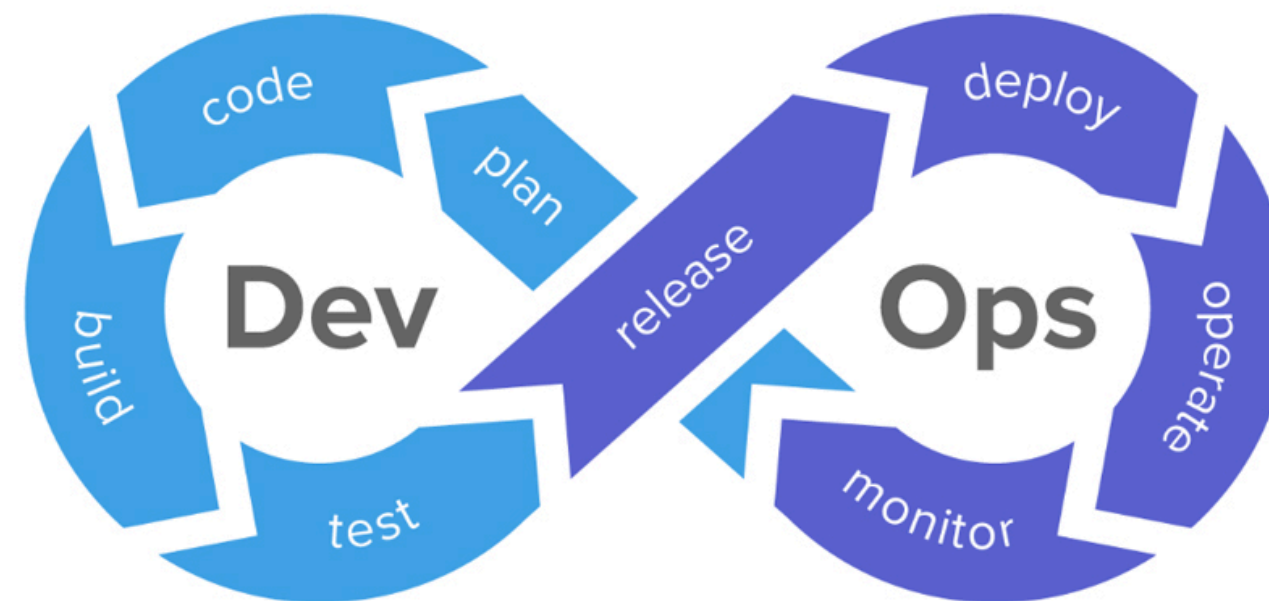
Monitoreo: Se enfoca en recopilar métricas predefinidas sobre el sistema (CPU, memoria, alertas).

Observabilidad: este enfoque nos permite ver el estado del sistema a partir de sus datos de salida como métricas, logs y trazabilidad.

- Permite detectar y solucionar problemas rápidamente en arquitecturas modernas (microservicios, contenedores).
 - Facilita el análisis de causa raíz de fallos inesperados.
- poco de texto

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

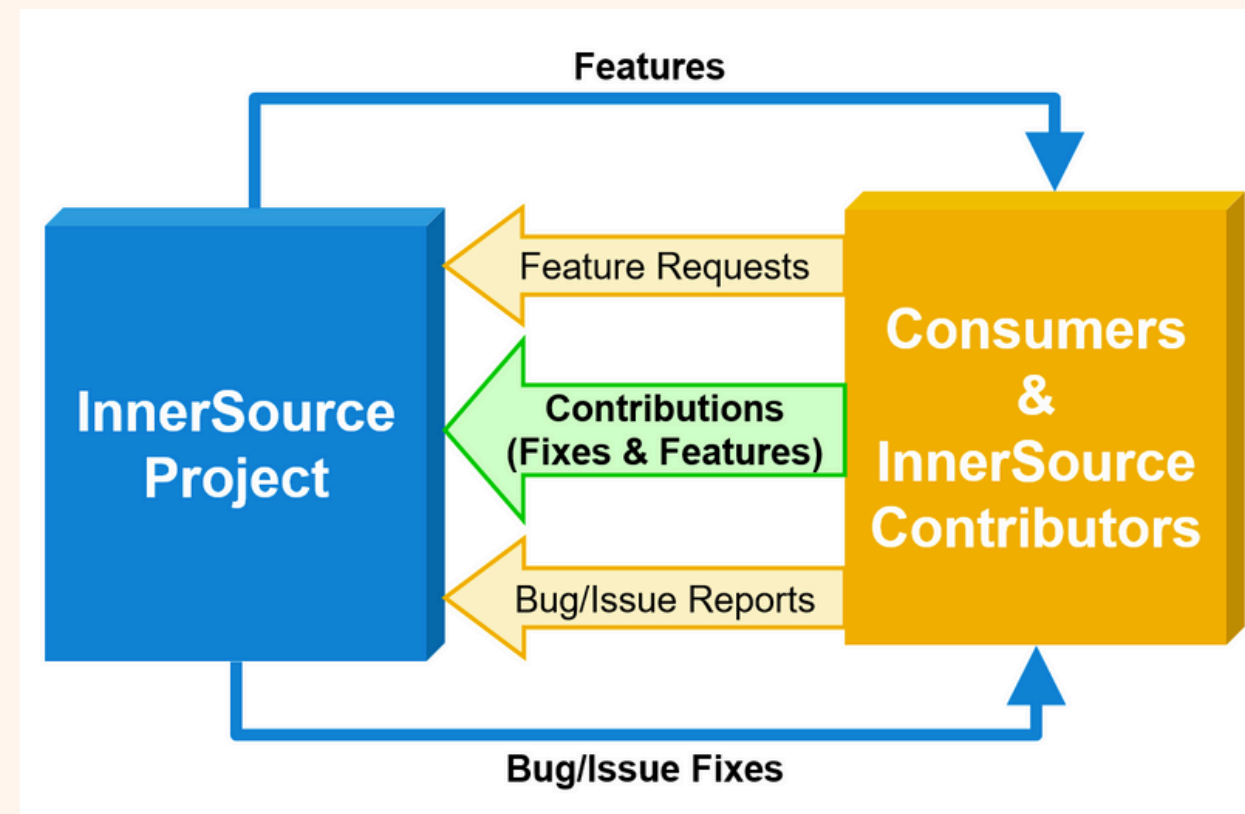
Al reducir silos entre los equipos , mejora la comunicación e impacta directamente en la calidad y velocidad del software desarrollado



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

Fomenta la colaboración abierta entre equipos mediante la contribución compartida a proyectos internos

Reduce la dependencia de equipos específicos al permitir que cualquier desarrollador contribuya a diferentes proyectos



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

La ingeniería de plataformas se enfoca en crear herramientas y servicios internos para mejorar el flujo de trabajo de los desarrolladores.

Algunos ejemplos puede ser Github Actions, Docker + Kubernetes y Grafana + Prometheus

