



UNIVERSIDAD DE SONSONATE

IMPLANTACIÓN DE SISTEMAS

Jueves, 17 de Julio de 2025.

DOCENTE

MARLIN YAJAIRA MOZ MARÍN



marlin.moz@usonsonate.edu.sv



7803 - 9583

SINAPSIS Y WHATSAPP



CLAVE
IMP-SIS-G2

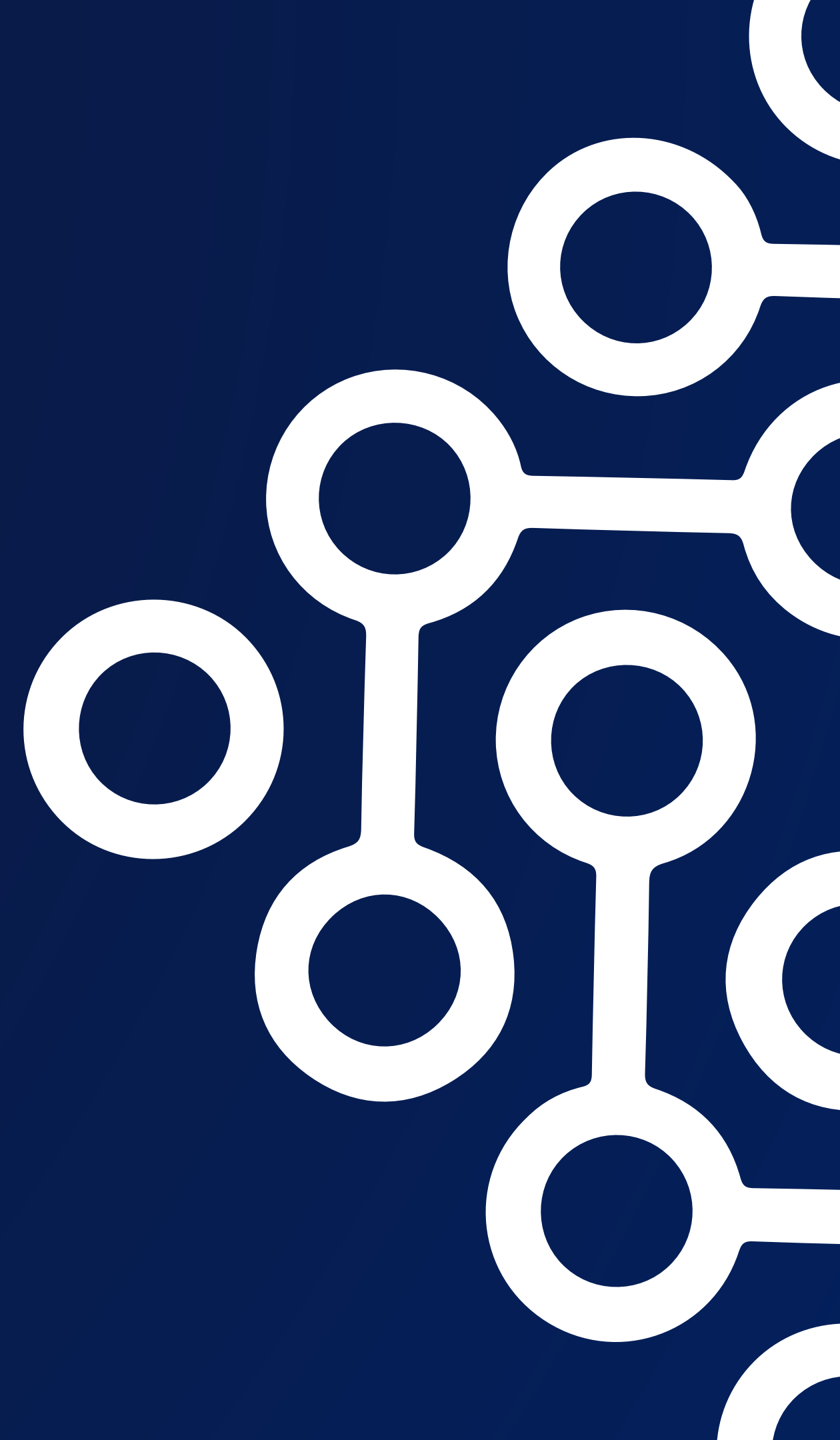


Implantación de Sistemas - G02
Grupo de WhatsApp



GENERALIDADES

NOMBRE DE LA ASIGNATURA	IMPLANTACION DE SISTEMAS
NÚMERO DE ORDEN	38
CÓDIGO	I04-2-18
PRERREQUISITO	<ul style="list-style-type: none">▪ ANÁLISIS DE SISTEMAS▪ DESARROLLO DE SOFTWARE II
NÚMERO DE HORAS POR CICLO	80
HORAS TEÓRICAS SEMANALES PRESENCIALES	1
HORAS TEÓRICAS SEMANALES NO PRESENCIALES	1
HORAS PRÁCTICAS SEMANALES PRESENCIALES	2
HORAS PRÁCTICAS SEMANALES NO PRESENCIALES	0
DURACIÓN DEL CICLO	20 SEMANAS
DURACIÓN DE LA HORA CLASE	50 MINUTOS
UNIDADES VALORATIVAS	4
CICLO ACADÉMICO	VIII





MODELO DE EVALUACIÓN

EVALUACIÓN

PORCENTAJE

- ✓ PRIMER PARCIAL
- ✓ SEGUNDO PARCIAL
- ✓ 2 LABORATORIOS
- ✓ PARCIAL FINAL

20 %

20 %

20 %

40 %

100 %

¿QUÉ ES IMPLANTACIÓN DE SISTEMAS?

La implantación de sistemas es el proceso de integrar y desplegar una nueva solución tecnológica (software, hardware o procesos) en una organización, asegurando que sea adoptada con éxito por los usuarios y que cumpla con los objetivos del negocio.



— ¿QUÉ ES IMPLANTACIÓN DE SISTEMAS?

Incluye etapas clave como:

1. **Planificación:** Alinear el sistema con las necesidades de la empresa.
2. **Instalación y configuración:** Adaptarlo a la infraestructura existente.
3. **Pruebas y ajustes:** Garantizar su funcionamiento óptimo.
4. **Capacitación:** Preparar a los equipos para su uso.
5. **Soporte continuo:** Resolver problemas post-implantación.

— CONTENIDO DE LA MATERIA

UNIDAD 1 - FUNDAMENTOS DE INTEGRACIÓN Y DESPLIEGUE

- 1.0 - Introducción y conceptos fundamentales.
 - 1.1 - Definición de integración y despliegue.
 - 1.2 - Beneficios y retos en entornos modernos.
- 2.0 - Manejo de la terminal.
 - 2.1 - Comandos básicos y avanzados.
 - 2.2 - Navegación de sistemas de archivos y ejecución de scripts.



— CONTENIDO DE LA MATERIA

UNIDAD 1 - FUNDAMENTOS DE INTEGRACIÓN Y DESPLIEGUE

- 3.0 - Monitoreo de procesos y performance.
- 3.1 - Identificación y control de procesos.
- 3.2 - Herramientas para el análisis del rendimiento.
- 4.0 - Protocolos y herramientas de red.
- 4.1 - Introducción a los protocolos de red.
- 4.2 - Diagnóstico de red con herramientas básicas.
- 4.3 - Análisis de conectividad y transferencia de datos.



— CONTENIDO DE LA MATERIA

UNIDAD 2 - CONTROL DE VERSIONES Y USO DE CONTENEDORES

- 1.0 - Administración del versionamiento.
- 1.1 - Principios del control de versiones.
- 1.2 - Uso de sistemas como Git / GitHub / GitLab.
- 2.0 - Estrategias de versionamiento.
- 2.1 - Ramas, fusiones y resolución de conflictos.
- 2.2 - Flujos de trabajo en equipos de desarrollo.



— CONTENIDO DE LA MATERIA

UNIDAD 2 - CONTROL DE VERSIONES Y USO DE CONTENEDORES

3.0 - Contenedores.

3.1 - Introducción a Docker.

3.2 - Creación y configuración de imágenes y contenedores.

3.3 - Gestión de redes y volúmenes en contenedores.



— CONTENIDO DE LA MATERIA

UNIDAD 3 - AUTOMATIZACIÓN, DESPLIEGUE Y MONITOREO

- 1.0 - Proveedores de nube.
 - 1.1 - Introducción a servicios de nube.
 - 1.2 - Configuración de entornos AWS, Azure y Google Cloud.
- 2.0 - Herramientas de CI/DC.
 - 2.1 - Introducción a la integración y despliegue continuo.
 - 2.2 - Uso de herramientas como GitHubActions, GitLab CI/CD.
- 3.0 - Contenedores de orquestación.
- 4.0 - Monitoreo de infraestructura,



UNIDAD 1

FUNDAMENTOS DE INTEGRACIÓN Y DESPLIEGUE



Objetivo: Comprender los conceptos básicos de integración y despliegue de software, mediante el uso de terminales, monitoreo de procesos y desempeño, utilizando herramientas de red para gestionar eficientemente el ciclo de vida del software.

UN POCO DE HISTORIA

Era Prehistórica

Integración manual.

Despliegue con cintas magnéticas o tarjetas perforadas.

1960s - 1980s

Revolución ágil y CI

Propuesta de Martin Fowler (2006) para la integración continua.

Despliegue con virtualización (VMware) y primeros entornos cloud.

2000s

1990s

Nacimiento del control de versiones

Integración con Sistemas como CVS.

Despliegue con FTP y scripts básicos.

2010s - Presente

Era moderna

Integración con Git, contenedores (Docker).

Despliegue con Kubernetes, serverless, CI/CD automatizado.

—— ¿QUÉ ES INTEGRACIÓN?

Es el proceso de unir componentes, módulos o sistemas independientes para que trabajen de forma conjunta como un sistema completo.

Su objetivo es asegurar que las diferentes partes del software o de la infraestructura se comuniquen correctamente, compartan datos y funcionen sin conflictos.

CARACTERÍSTICAS CLAVE



AUTOMATIZACIÓN



FRECUENCIA



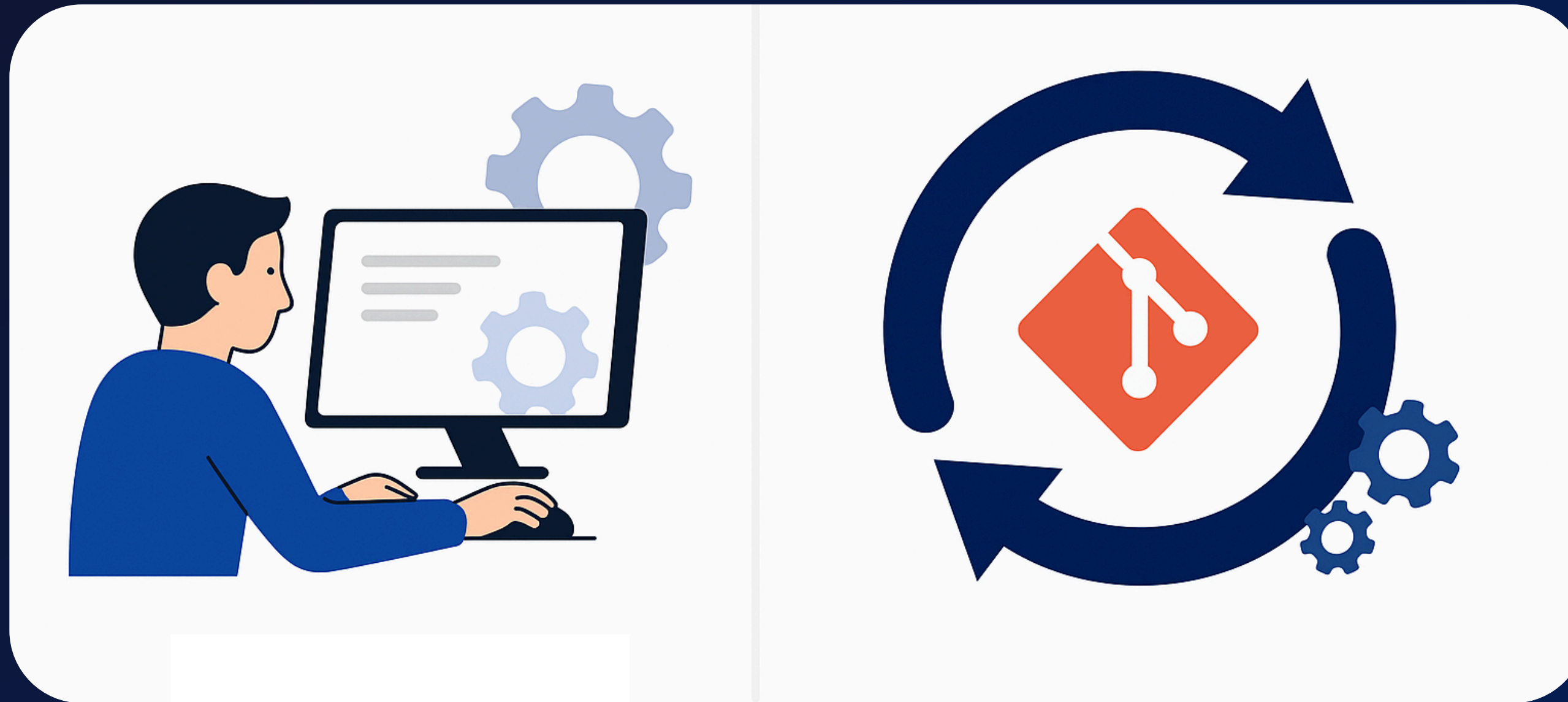
FEEDBACK RÁPIDO



ENTORNO CONSISTENTE



INTEGRACIÓN TRADICIONAL VS INTEGRACIÓN MODERNA



Aspecto

Integración tradicional

Integración moderna

1- Frecuencia de Integración

Big Bang Integration: se integra al final de ciclos largos.

Integración continua: múltiples commits diarios.

Alto riesgo de conflictos. (merge hell)

Errores detectados y corregidos al instante.

2- Automatización

Proceso manual.

Flujo de trabajo automáticos.

Propenso a errores y lento.

Repetible y confiable.

3- Cultura y Colaboración

Equipos en silos.

Equipos colaborativos. (Dev+QA+Ops)

Culpa cruzada cuando algo falla.

Responsabilidad compartida.

4- Gestión de errores

Errores detectados tarde y costosos.

Pruebas fallan rapido y dan feedback inmediato.

Rollback complicados.

— CONCLUSIÓN

La integración moderna reduce riesgos, acelera la entrega de valor y mejora la calidad del software mediante:

- Automatización (menos errores humanos).
- Frecuencia (cambios pequeños y gestionables).
- Cultura colaborativa (equipos alineados con objetivos comunes).

—— ¿QUÉ ES DESPLIEGUE?

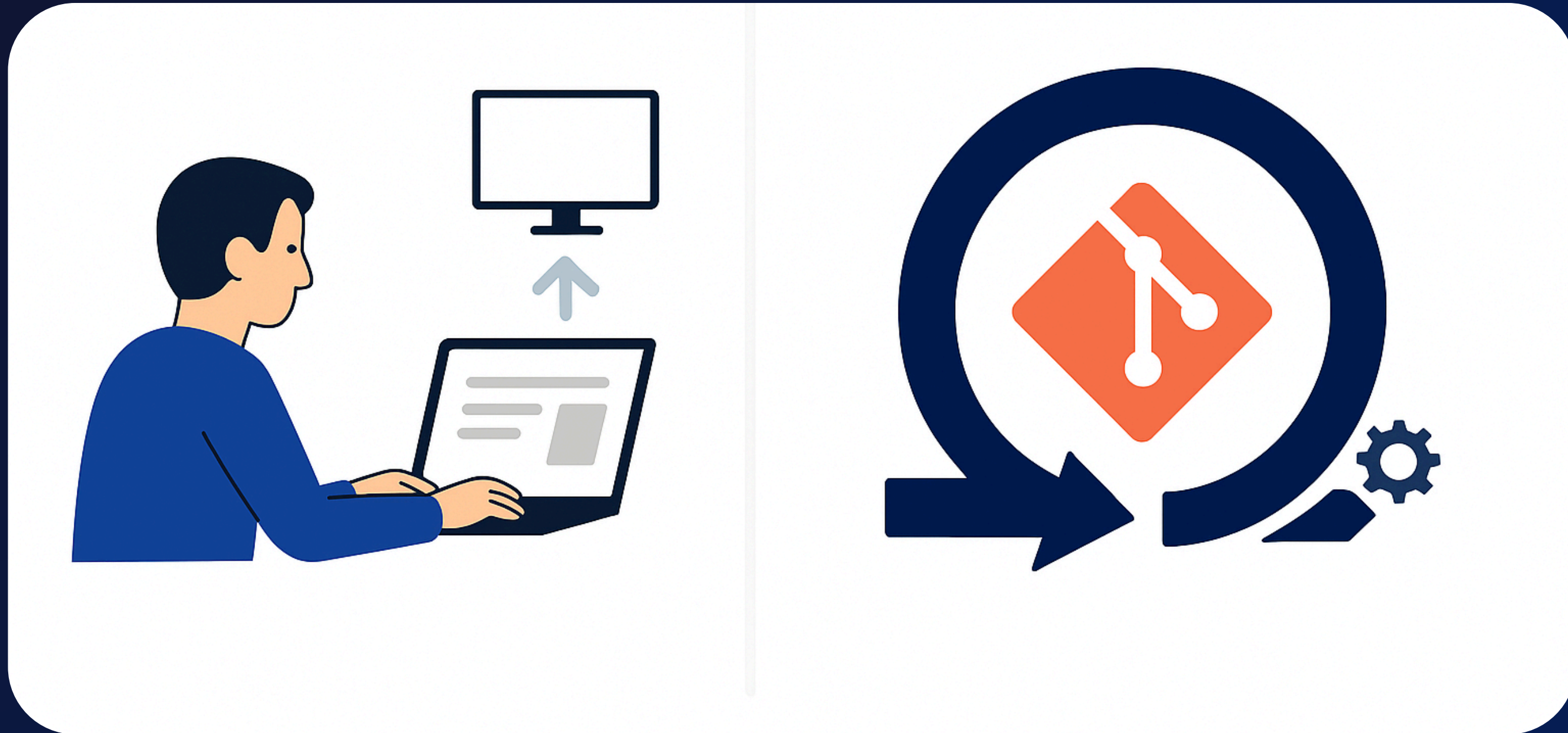
El despliegue (o deployment) es el proceso de liberar e instalar una versión de software en un entorno específico (como testing, staging o producción), haciendo que esté disponible para los usuarios finales.

Su objetivo es garantizar que los cambios desarrollados se implementen de forma segura, rápida y sin interrupciones.

CARACTERÍSTICAS CLAVE

- ✓ **ETAPA FINAL DEL CICLO DE VIDA**
Codificación → Pruebas → Integración
- ✓ **TIPOS DE ENTORNOS**
Desarrollo → Testing → Staging → Producción.
- ✓ **MANUAL O AUTOMATIZADO**

DESPLIEGUE TRADICIONAL VS DESPLIEGUE MODERNO



Aspecto

Integración tradicional

Integración moderna

1- Frecuencia de despliegue

Esporadico (mensual o trimestral) con lanzamientos grandes o versiones complejas.

Frecuente y continuo: se puede desplegar en cualquier momento sin interrupciones.

2- Proceso

Manual, usando FTP, copiando archivos, comandos remotos.
Requiere intervención humana.

Menor intervención humana y mayor velocidad.

3- Riesgo y errores

Alto riesgo de errores humanos.
Dificultad para detectar fallas rápidamente.

Bajo riesgo gracias a validaciones automáticas.
Feedback inmediato ante fallos.

4- Tiempo de recuperación

Lento y complicado
Rollback pueden tardar horas o requerir intervención urgente del equipo.

Rollbacks rápidos.
Uso de feature flags o blue-green deployments.

5- Herramientas

Dependencia de scripts locales o documentación.

Herramientas de automatización.

6- Visibilidad y control

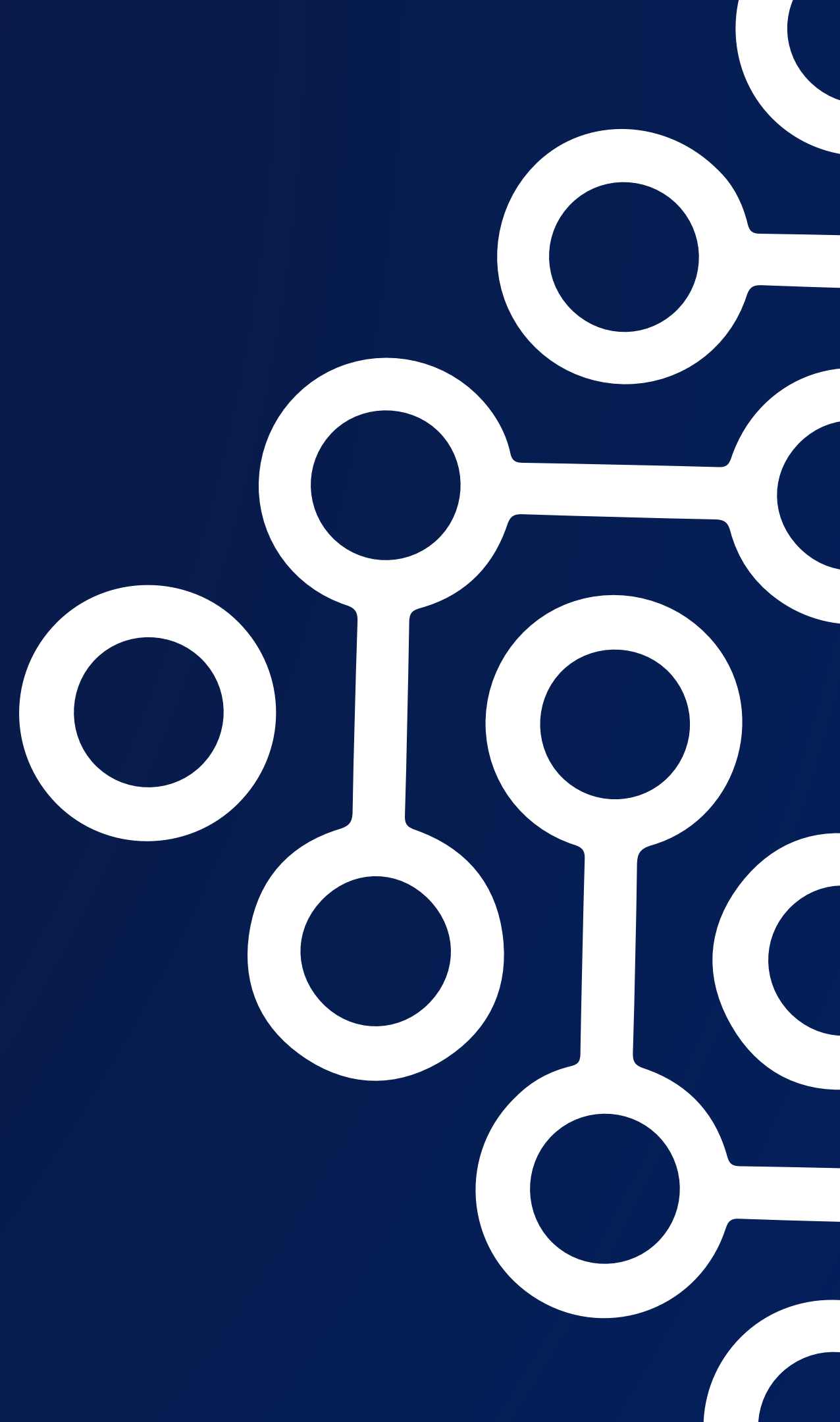
Difícil seguimiento de quien desplegó.
Poca trazabilidad y monitoreo.

Historial claro de despliegues.
Métricas, logs y monitoreo integrados.



— ¿SABÍAS QUE...?

¿Sabías que Netflix despliega código a producción más de 1,000 veces al día? Esto es posible gracias a las técnicas de integración y despliegue que aprenderán en esta materia.



CASO DE ESTUDIO

EJEMPLO 1 - EL CLÁSICO “FUNCIONA EN MI MAQUINA”

"Juan desarrolló una función de 'Carrito de compras' que funciona perfectamente en su laptop. Pero cuando se despliega a producción, la aplicación se crashea. ¿Qué pudo haber fallado?"

EJEMPLO 2 - INTEGRACIÓN ENTRE EQUIPOS

"El equipo A desarrolló el módulo de 'Login' y el equipo B el de 'Perfil de usuario'. Ambos funcionan perfectamente por separado, pero cuando se integran, los usuarios no pueden acceder a su perfil. ¿Qué falló en la integración?"

POSIBLES PROBLEMAS



EJEMPLO 1 - EL CLÁSICO “FUNCIONA EN MI MAQUINA”

- Diferentes versiones de dependencias
- Variables de entorno faltantes
- Base de datos de desarrollo vs producción
- Configuraciones específicas del entorno

EJEMPLO 2 - INTEGRACIÓN ENTRE EQUIPOS

- APIs incompatibles
- Diferentes formatos de datos
- Tokens de autenticación no compartidos
- Falta de comunicación entre equipos