

Integración y entrega continua de software

2025-2

Duración: 24 horas

Especialización en Desarrollo de Software

Horarios:

- viernes 18:00 - 21:00 (13-305)
- sábado 8:00 - 11:00 (13-405)

Docente



Daniel Hoyos Ospina

ML Engineer - Factored

Ingeniero de Sistemas - EAFIT


Magíster en Inteligencia Analítica de Datos – UniAndes

dhoyoso@eafit.edu.co

<https://github.com/dhoyoso>

<https://www.linkedin.com/in/daniel-hoyos-ospina-534a41175/>

REQUISITOS PREVIOS

- Computador personal con acceso a internet y permisos de administrador.
 - Aplicativos previamente instalados:
 - Python 3.12 – pip
 - Git
 - Docker
 - VisualStudioCode (recomendado) o cualquier otro ambiente de desarrollo / editor de código en el que te sientas cómodo.
- 


Objetivo del curso:

“ Comprender los **frameworks** y **prácticas** conducentes a la **automatización** de procesos dentro del **ciclo de vida del software**, que permiten la integración de los mismos desde el **desarrollo** hasta la **operación** de un sistema ”

Contenido del curso:

Unidad 1: Fundamentos de DevOps


Duración: 6 horas

- ¿De dónde venimos? – Ciclo tradicional desarrollo de software
 - ¿Qué es DevOps?
 - Historia, estadísticas, valor para el negocio y casos de éxito.
 - Principios, pilares, prácticas y cultura DevOps.
 - Introducción a las Herramientas DevOps.
 - Introducción a CI/CD con GitHub y GitHub Actions.
- 

Contenido del curso:

Unidad 2: Dominando la Integración Continua (CI)


Duración: 6 horas

- Por qué la automatización es indispensable.
 - Pruebas automatizadas.
 - Anatomía de un pipeline de CI: Code, Build, Test, Release.
 - Gestión de artefactos y secretos.
 - Deuda técnica y cómo CI ayuda a gestionarla.
- 

Contenido del curso:

Unidad 3: Entrega Continua y Despliegue Continuo (CD)


Duración: 6 horas

- ¿Qué es y por qué necesitamos CD?
 - Delivery vs. Deployment.
 - El Pipeline de CD.
 - Etapas y Pruebas Clave en CD (Más allá de CI).
 - Herramientas que orquestan la entrega.
 - Estrategias Fundamentales para Despliegues Seguros.
 - Rollbacks: Nuestro plan B automatizado.
 - El Rol de la Nube en CD.
 - Contenedores, Kubernetes, Serverless.
 - Observabilidad
 - Infraestructura como Código (IaC) y Seguridad en CD.
 - Contexto importa: Adaptando el pipeline.
 - El futuro de las operaciones: NoOps vs. NewOps/SRE.
 - Midiendo el éxito: Métricas DevOps (DORA).
- 

Contenido del curso:

Unidad 4: DevOps Avanzado: Estrategias, Tendencias y Futuro

Duración: 6 horas

- DORA, IA y GenAI en DevOps y sus empresas.
 - Ejemplos de pipelines CI/CD en la industria (Referenciación).
 - DevOps: Más allá de CI/CD: DevSecOps, MLOps, IA y el Futuro.
 - DevSecOps, LLMOps, MLOps, entre otros.
 - Inteligencia Artificial en CI/CD: Optimización de código, pruebas y despliegues.
 - DevOps para software embebido y edge computing
 - Otros dominios de interés para DevOps
 - ¿Cómo cambiará DevOps en los próximos 5-10 años?
 - Cierre.
- 

Presentación

- Nombre
- Empresa - cargo
- Fun Fact
- Expectativa curso



Acuerdos

- Cuestionar
- Aprendizaje mutuo
- Uso de IA
- Comunicación – Teams o Correo



Evaluación del curso:

(6 septiembre) - 10% - Taller en parejas Entregable 1: Introducción a CI/CD con GitHub y GitHub Actions

(13 septiembre) - 25% - Taller en grupo Entregable 2: Pipeline de CI con Python, GitHub Actions y Herramientas Open Source

(20 septiembre) - 30% - Taller en grupo Entregable 3: Despliegue Continuo (CD) con Proveedores Cloud.

(26 septiembre) - 20% - Proyecto final en grupos (ir adelantando desde el 13 de sept)

(27 septiembre) - 15% - Foro de discusión (DORA, IA y GenAI en DevOps y sus empresas) – Lectura previa de recursos base, material adicional y discusión en clase - **nota individual**)

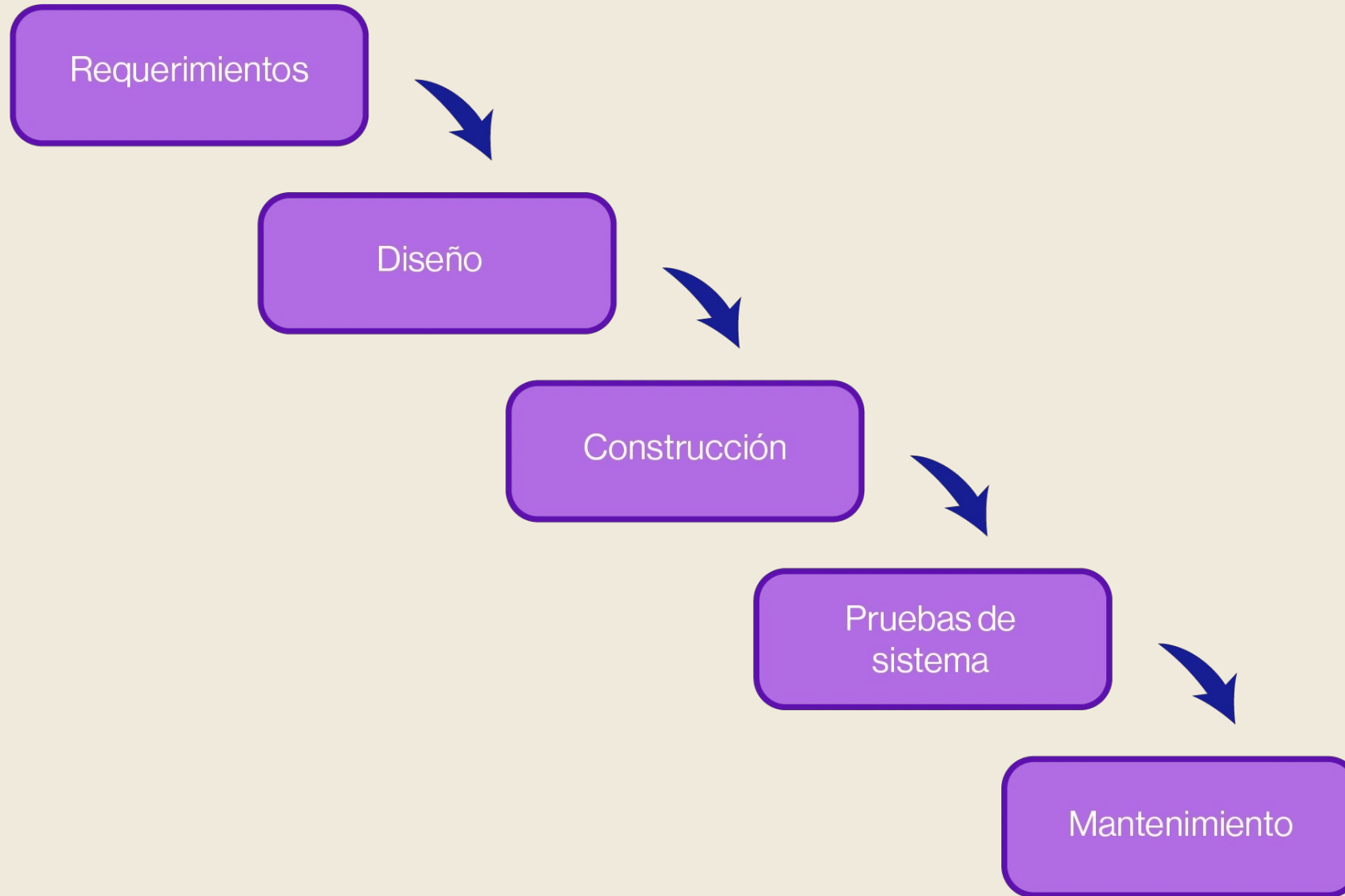
- Las fechas de los talleres son tentativas, ya que planeamos realizarlos durante el horario habitual de clase.
- Cada taller tendrá una fecha límite de entrega, que se definirá en la sesión correspondiente.
- Esta fecha aplicará si no se alcanza a completar el taller durante el tiempo asignado en clase.

Viernes: 6 pm a 9 pm (**receso** 7:30 pm – 15 min)
Sábados: 8 am a 11 am (**receso** 9:30 am – 15 min)

Total horas: 24

Unidad 1: Fundamentos de DevOps

¿De dónde venimos?



¿De dónde venimos?

Equipos independientes (silos)

Objetivos independientes ¿Cuáles?

Procesos manuales

Problemas de Integración

Problemas de calidad

Ciclos y tiempos largos x
releases o versiones grandes

Desconfianza
Entre equipos

Caídas grandes por
grandes releases

=

< Innovación



Innovación o transformación digital

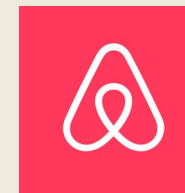
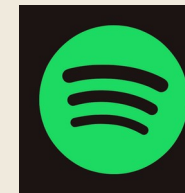
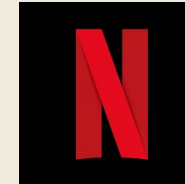
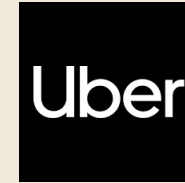
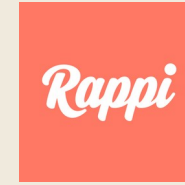
¿PLUS o NECESIDAD?

Mundo Tradicional



VS

Mundo Digital



Ciclo tradicional desarrollo de software

- El ciclo tradicional de desarrollo de software **impone grandes restricciones** cuando las organizaciones **necesitan lanzar ágilmente** nuevas iniciativas o productos digitales.
- El ciclo tradicional **impide a las compañías evolucionar y mantenerse a la vanguardia tecnológica** en los productos y servicios que ofrecen.
- Esto conlleva a un **alto costo** basado en el reproceso, falta de cohesión y comunicación de los equipos.



Los retos del mundo digital

Entorno

- Usuarios más exigentes (usabilidad y tiempos de respuesta).
- Múltiples plataformas, dispositivos, versiones y canales.
- Cantidad de usuarios.
- Cantidad de transacciones por usuario (miles o millones de transacciones por segundo).
- Competitividad – Innovación constante.

Infraestructura

- Escalabilidad y elasticidad.
- Alta disponibilidad.
- Seguridad.

End 2 end

- Arquitecturas distribuidas - aplicaciones interconectadas entre sí (transaccionales, analítica, legacy, etc.).
- Tamaño de los equipos de desarrollo y operaciones.
- Esquemas de contribución.

¿Qué buscamos?

- Un conjunto de prácticas que buscan que el **proceso de innovación**, desde la creación de requerimientos, hasta la puesta en producción **sea más rápido**.
- **Ciclos cortos** de innovación, pasar de meses a días.
- **¿Y cómo lo hacemos?** Eliminando los conflictos, entre negocio y desarrollo con Metodologías Ágiles, y entre los equipos de desarrollo y operaciones con **DevOps**.



Entonces... ¿Qué es DevOps?

- Equipos **colaborativos y multifuncionales**.
- Equipos con los mismos **objetivos**.
- Procesos **automáticos**.
- Entregas de valor en **ciclos cortos**.
- Reducción, mitigación y rápida **corrección de errores**.
- **Confianza** entre equipos.






¿Qué es DevOps?

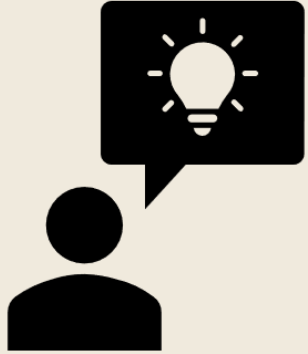
Definición

DevOps es la convergencia de **filosofías culturales, prácticas y herramientas tecnológicas** para aumentar la capacidad de una organización de **entregar aplicaciones y servicios rápidamente, a un menor costo y generando un mayor valor.**

Objetivo

Acortar el ciclo de vida del desarrollo de sistemas y proporcionar una **entrega continua de software** con una **alta calidad**. DevOps busca **fusionar el desarrollo de software y las operaciones de TI**, empleando **herramientas automatizadas** y fomentando la **colaboración**.





Si el objetivo es generar valor rápido... Entonces:

- Crea la infraestructura en la consola.
- Maneja un solo ambiente.
- No incluyas validaciones de seguridad.
- No monitorees.
- Despliega directamente en producción.

¡NO!

Otra forma de hacerlo es...

- | | |
|----------------------------|---------------|
| → Arquitectura. | → Monitoreo. |
| → Infraestructura. | → |
| → Separación de ambientes. | Conocimiento. |
| → Automatización. | → Equipo. |
| → Seguridad. | → |
| | Herramientas. |
| | → Pruebas. |

**Evitando
reprocesos entre
Dev y Ops.**

Historia y evolución de DevOps

1 ———— Finales de los 80 y Principios de los 90

Antes de DevOps, el modelo era tradicional, también conocido como **modelo en cascada** era lento y rígido, con **equipos aislados** y **procesos manuales** que dificultaban la flexibilidad y la adaptación.

2 ———— 2009

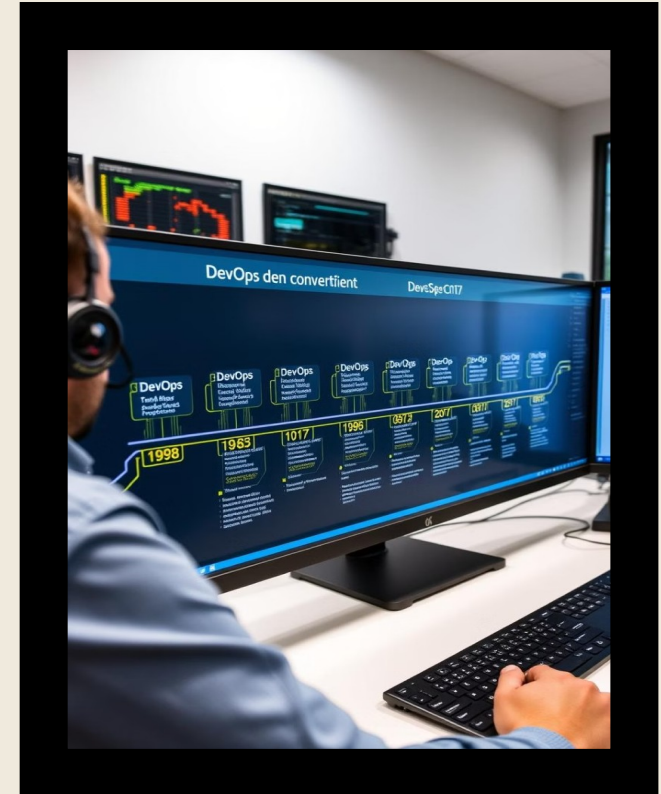
Entre 2007 y 2009, **surgió DevOps como respuesta** a las limitaciones del modelo tradicional, liderado por **Patrick Debois**. En 2009, se acuñó el término y se celebró la primera conferencia DevOpsDays en Bélgica.

3 ———— 2014

Entre 2010 y 2014, **DevOps creció** gracias a la **adopción de integración continua, entrega continua y computación en la nube**.

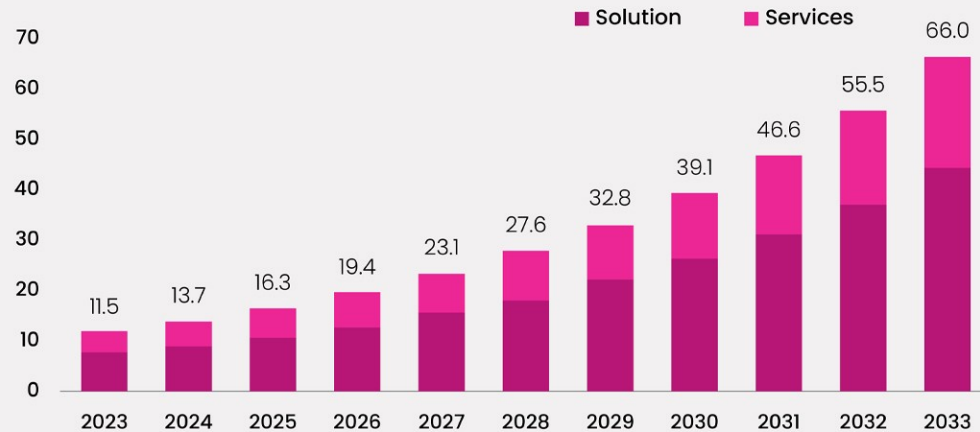
4 ———— 2015 al día de hoy

Desde 2015, DevOps ha evolucionado a una **filosofía global con tendencias** como **DevSecOps** e IA/ML (**LLMOps y MLOps**), y **GitOps (IaC, Configs)**. Además, la **ingeniería de plataforma** y las tecnologías nativas de la nube han impulsado su desarrollo hacia una entrega más rápida y



Adopción y estadísticas del Mercado Global de DevOps

Global DevOps Market
Size, 2023 - 2033 (USD Billion)



BACANCY®

DevOps Growth Statistics

DevOps has led to 60% of developers releasing code 2x faster.

Microsoft's \$12 billion allocation should cement Azure DevOps as the second-leading player, with an exceeding 25% share.

DORA 2023 report reveals that DevOps takes less than one day to change, leading to faster results; it takes less than 1 hour to recover from deployment failure and a 5% change failure rate.

Companies that adopted DevOps practices reported 2.8 times more frequent software deployments in 2023.

\$80,168

Salario anual de un DevOps promedio según encuestas de StackOverflow Global

BACANCY®

DevOps Enhancement

99%

of developers say DevOps has been beneficial for their organizations.

Career

48% of respondents were raised

Delivery

61% of experts state it helps them deliver high-quality products

Faster delivery

2.5 x times faster market launch

49%

of developer say it enhances their development frequency

Gráficos y datos tomados de:

DevOps Statistics: Key Trends and Insights in 2025 - Bacancy Technology, acceso: marzo 19, 2025,

<https://www.bacancytechnology.com/blog/devops-statistics>



Automatización

La automatización en DevOps optimiza el ciclo de vida del desarrollo mediante **pipelines de CI/CD e IaC**, acelerando la entrega, reduciendo errores y mejorando la eficiencia.



Colaboración

La colaboración en DevOps fomenta **equipos multifuncionales con autonomía**, flujos de trabajo claros y herramientas eficaces para mejorar la calidad del software y la resolución de problemas.



Mejora Continua

La mejora continua en DevOps promueve **experimentación, retroalimentación** constante y métricas como DORA para optimizar procesos, **fomentar el aprendizaje** y mejorar la entrega de software.



Enfoque en el Usuario

Los equipos de DevOps emplean ciclos de retroalimentación cortos y monitorización en tiempo real para desarrollar **productos centrados en el usuario, alineando objetivos comerciales y satisfacción del cliente**.

La importancia de la **cultura** y la **colaboración** en DevOps

Fomentar una cultura de DevOps requiere varias prácticas recomendadas:

- Comenzar poco a poco con **proyectos manejables**.
- Derribar los silos y fomentar **equipos multifuncionales**.
- Fomentar la **comunicación abierta y la transparencia**.
- Construir **confianza** y seguridad psicológica.
- **Automatizar** tareas repetitivas.
- Alentar el **aprendizaje continuo** y la experimentación.
- Implementar **reuniones periódicas**.

En esencia, **DevOps es una transformación cultural**; si bien las herramientas y los procesos son importantes, **sin la cultura adecuada de colaboración, confianza y mejora continua, los beneficios de DevOps no se pueden realizar por completo.**

Beneficios cuantificados de DevOps

Métrica	Enfoque Tradicional	Enfoque DevOps	Mejora/Reducción %	Fuente
Frecuencia de Implementación	Semanal/Mensual	Múltiples veces al día	200x más frecuente	DevOps Case Studies: Lessons from the Industry - Nucamp, acceso: marzo 19, 2025, https://www.nucamp.co/blog/coding-bootcamp-back-end-with-python-and-sql-devops-case-studies-lessons-from-the-industry
Tiempo de Recuperación	Horas/Días	Menos de un día	24x más rápido	
Tasa de Fallos en los Cambios	Mayor	Menor	3x menor	
Tiempo de Comercialización	Mayor	Menor	Reducción del 50%	DevOps Success Stories: Real-Life Case Studies Attract Group, acceso: marzo 19, 2025, https://attractgroup.com/blog/devops-success-stories-real-life-case-studies/
Tiempo de Implementación	Días	Minutos	Reducción del 75%	5 Real-World DevOps Case Studies You Can Learn From by Mihir ..., acceso: marzo 19, 2025, https://mihirpopat.medium.com/5-real-world-devops-case-studies-you-can-learn-from-4c963d09150f
Tiempo dedicado a Incidencias	Mayor	Menor	21% menos	DevOps Statistics You Should Know in 2025 StrongDM, acceso: marzo 19, 2025, https://www.strongdm.com/blog/devops-statistics
Tiempo dedicado a Mejoras	Menor	Mayor	33% más	
Calidad de la Implementación	Menor	Mayor	63% de mejora	DevOps Success Stories: Real-Life Case Studies Attract Group, acceso: marzo 19, 2025, https://www.attractgroup.com/blog/devops-success-stories-real-life-case-studies/
Tiempo de Entrega	Mayor	Menor	26 veces más rápido	DevOps Success Stories: Real-Life Case Studies Attract Group, acceso: marzo 19, 2025, https://attractgroup.com/blog/devops-success-stories-real-life-case-studies/

Casos de éxito

Empresa	Industria	Desafíos Previos a DevOps	Prácticas/Herramientas DevOps Implementadas	Resultados Clave/Impacto Empresarial
Capital One	Servicios Financieros	Ciclos de lanzamiento lentos, equipos aislados	Pruebas automatizadas, integración continua, DevSecOps	Mayor velocidad de implementación, mejora de la calidad, <u>reducción de riesgos de seguridad</u>
Etsy	Comercio Electrónico	Tiempo de inactividad frecuente, equipos aislados	Equipos multifuncionales, colaboración, herramientas de implementación interna	Mayor estabilidad, ciclos de lanzamiento reducidos, <u>mejora de la satisfacción del cliente</u>
Netflix	Streaming de Contenido	Riesgos por infraestructura a un solo nivel, fallos de BD	Automatización, CI/CD, arquitectura de microservicios, ingeniería de resiliencia	Mayor velocidad de implementación, mayor resiliencia del sistema, innovación continua. <u>Su cultura de "operar lo que construyes" es un factor clave en su éxito.</u>
DocuSign	Firma Electrónica	Uso de metodología Agile con desafíos en la implementación	Colaboración mejorada, automatización, arquitectura de microservicios	Velocidades de implementación aceleradas, mayor confiabilidad y <u>resiliencia del sistema, mejor escalabilidad</u>
Amazon	Comercio Electrónico/Cloud	Procesos tradicionales lentos y frecuentes interrupciones	Automatización, propiedad, agilidad, AWS	<u>Alta frecuencia de implementaciones, innovación continua, liderazgo en la nube</u>

Datos curiosos

AMAZON → Despliegues cada 11.6 segundos (o incluso más rápido en algunos equipos). Miles de despliegues al día.

Fuente: Artículo "Going Faster Than You Thought Possible" (2011, All Things Distributed, blog de Werner Vogels, CTO de Amazon). <https://www.allthingsdistributed.com/2011/08/devops-at-amazon.html> Nota: Aunque es una fuente antigua, este dato se ha citado ampliamente y sigue siendo representativo de la filosofía de Amazon., Entrevista con Werner Vogels, 2016, **que indica 50 millones de despliegues al año** <https://www.infoq.com/news/2016/05/amazon-50m-deploys-year/>

→ Equipos "Dos Pizzas": Equipos pequeños (generalmente 6-10 personas), autónomos y responsables.

Fuente: Concepto ampliamente atribuido a Jeff Bezos y discutido en numerosas presentaciones de AWS re:Invent y artículos sobre la cultura de Amazon. Nota: No hay una única fuente "oficial", pero es un concepto central en la organización de Amazon.

NETFLIX → "Chaos Monkey" y Simian Army: Herramientas que introducen fallos aleatorios en producción para probar la resiliencia.

Fuente: Blog de Tecnología de Netflix (numerosos artículos sobre Chaos Engineering) y presentaciones de Adrian Cockcroft (ex-arquitecto de nube de Netflix). Ejemplo: <https://netflixtechblog.com/chaos-engineering-upgraded-878d3e2a4625>

GOOGLE → "Post-mortems" sin culpa: Cultura de aprendizaje a partir de los errores, sin buscar culpables.

Fuente: Libro "Site Reliability Engineering" (Google) y blog de Google Cloud. "Blameless Postmortems and a Just Culture", Google Blog.

Pilares DevOps

Configuration Management

Gestionar con buenas prácticas el código fuente y compilados de la aplicación. Ejemplos:

- Versionamiento de código
- Versionamiento de dependencias.
- Versionamiento de artefactos.

Continuous Testing

Probar todos los cambios a través de pruebas automatizadas. Ejemplos:

- Pruebas unitarias.
- Pruebas modulares.
- Pruebas de seguridad.
- Pruebas de performance.

Continuous Integration

Integrar de manera automática todos los cambios en el código fuente. Ejemplo:

- Integrar código de diferentes devs en un repo central con validaciones automáticas.

Continuous Deployment

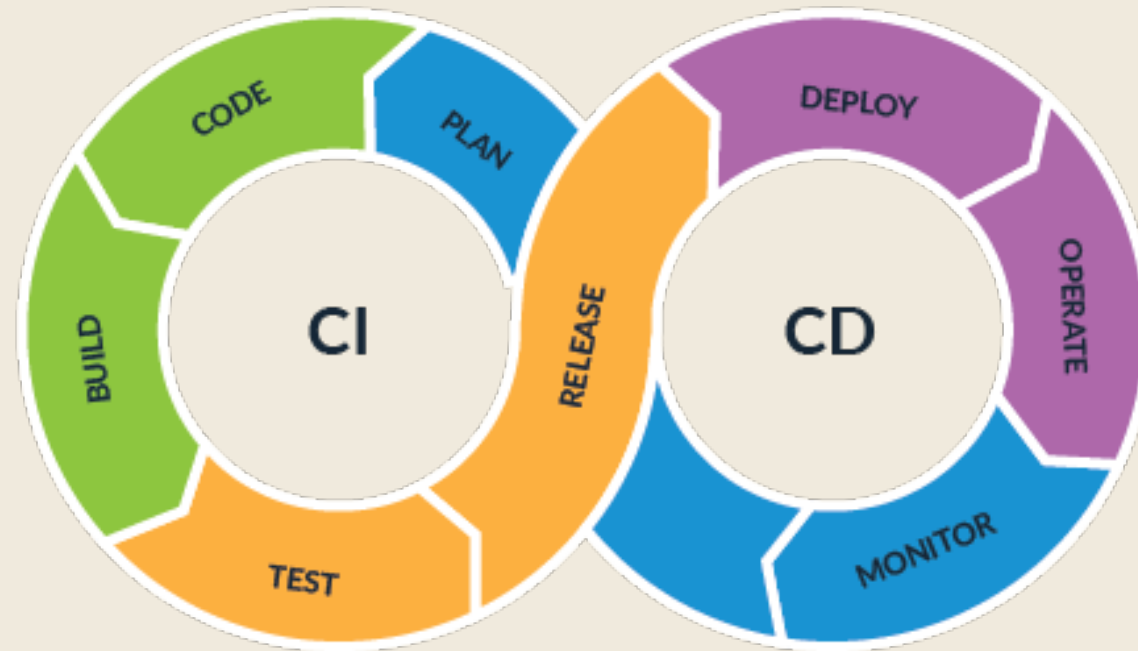
Instalar y distribuir cambios de manera automática en todos los ambientes. Ejemplo:

- Configuración y ejecución automática de tareas de despliegue repetibles en cada ambiente.

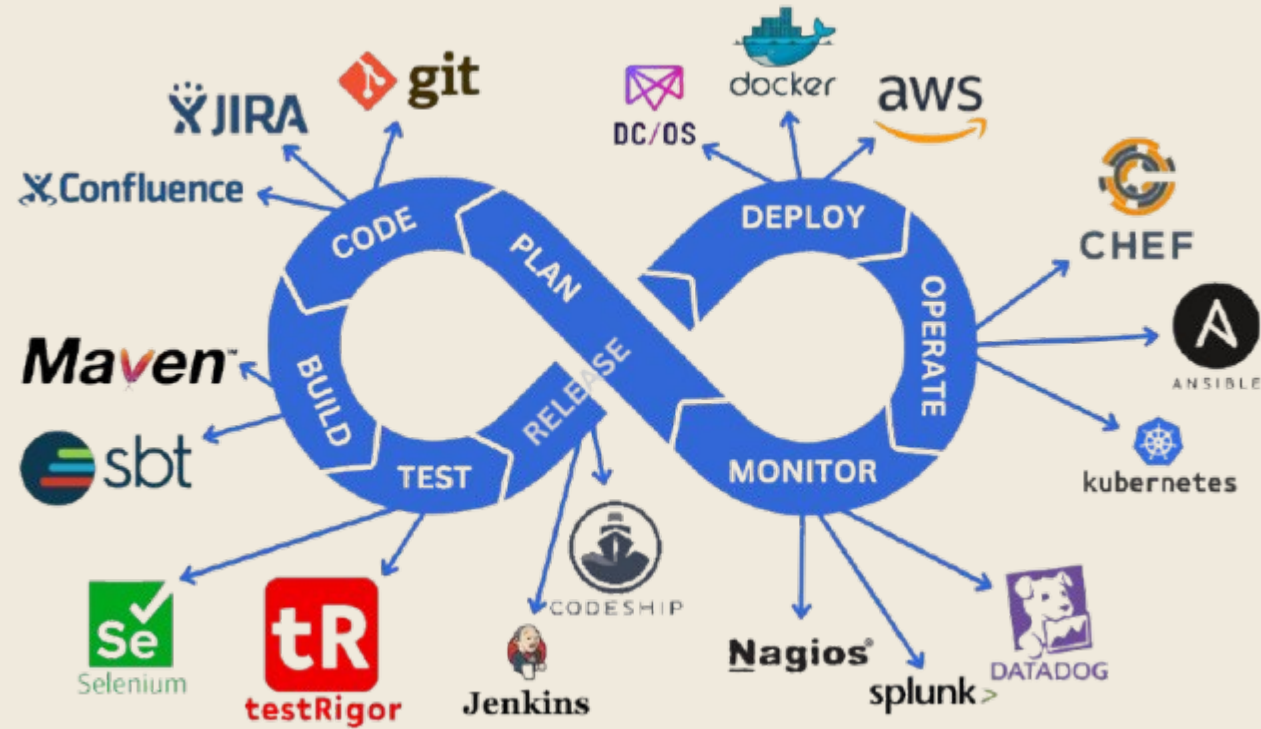
Release Management

Gestionar las entregas de aplicaciones desde la creación del requerimiento o historia de usuario hasta la salida a producción.


Prácticas DevOps



Introducción a Herramientas DevOps



ACTIVIDAD EN GRUPOS DE 4


- Dentro de sus organizaciones o una que encuentren en internet, ¿utilizan DevOps? O ¿se identifican con los dolores del modelo tradicional? – Lístenlos todos en un Slide
 - ¿Cómo son los pipelines de CI/CD en sus empresas? ¿Qué pasos tienen? – Graficar en un Slide
 - ¿Cuáles son las principales herramientas que utilizan en los cada paso de los pipeline de CI/CD de sus empresas? (Categorízalas en qué etapa del pipeline están) – Graficar en un Slide
 - Describan los beneficios y desventajas que perciben en sus empresas por usar CI/CD.
- 

GitHub

y

Github Actions

- Plataforma en la nube que permite a los **usuarios almacenar, compartir y editar código**.
- **Colaborar en un proyecto compartido**, sin preocuparse de que los cambios afectarán al trabajo de los colaboradores **antes de que esté listo para integrarlos**.

- Plataforma para **automatizar flujos de trabajo** de desarrollo de software.
 - **Pipelines de CI/CD** es uno de tantos flujos de trabajo automatizados que se pueden crear.
- 

¿Por qué Github Actions?

Simplicidad

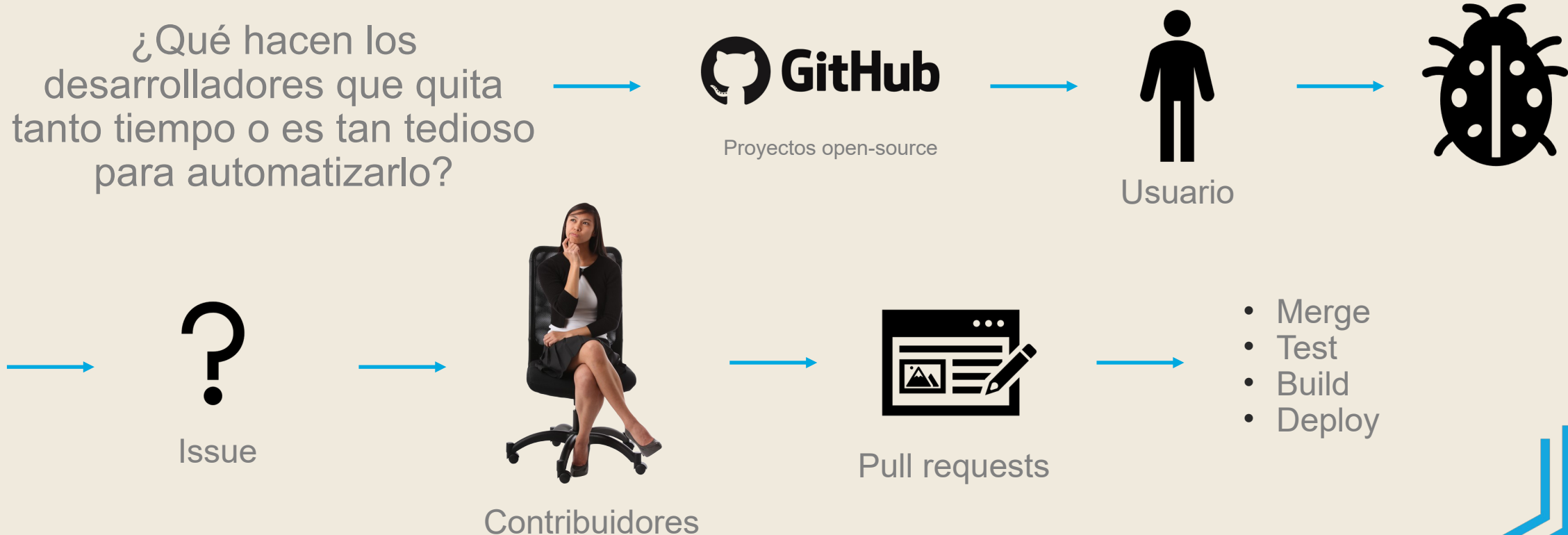
Integración

Capa Gratuita *

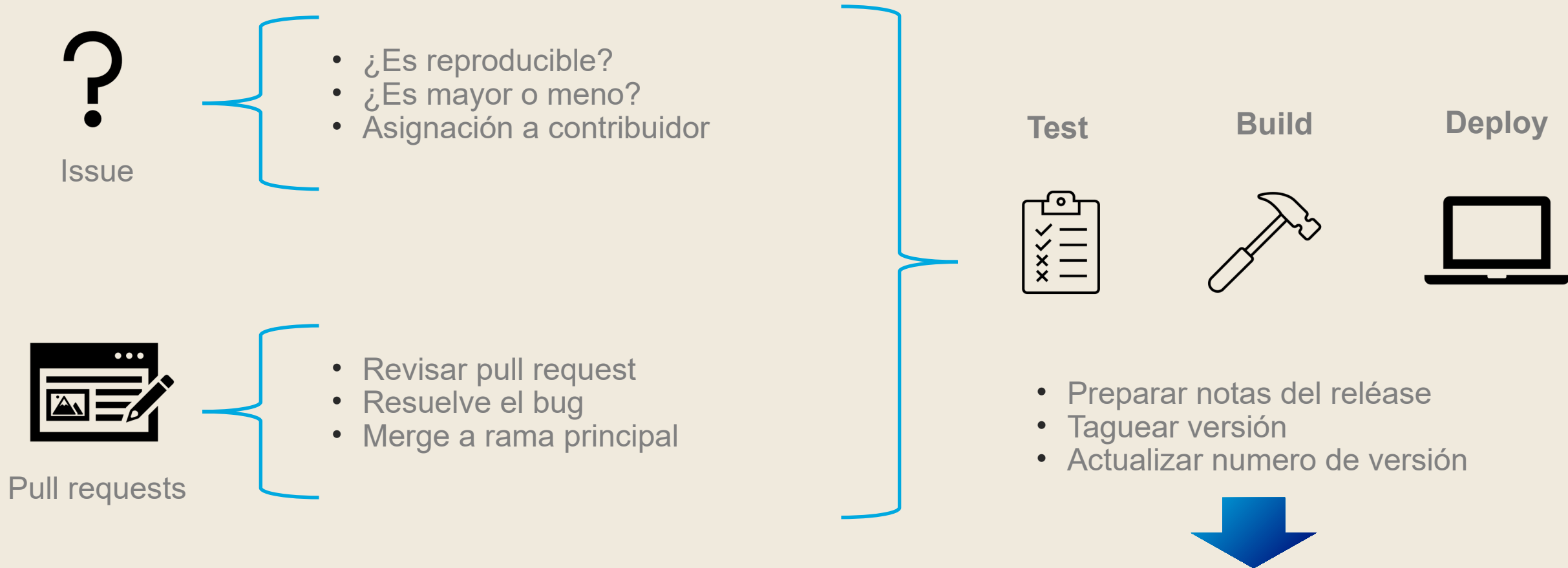
En repositorios públicos (500MB Almacenamiento y 2000 minutos de ejecución).

¿Qué tipos de flujos se automatizan?

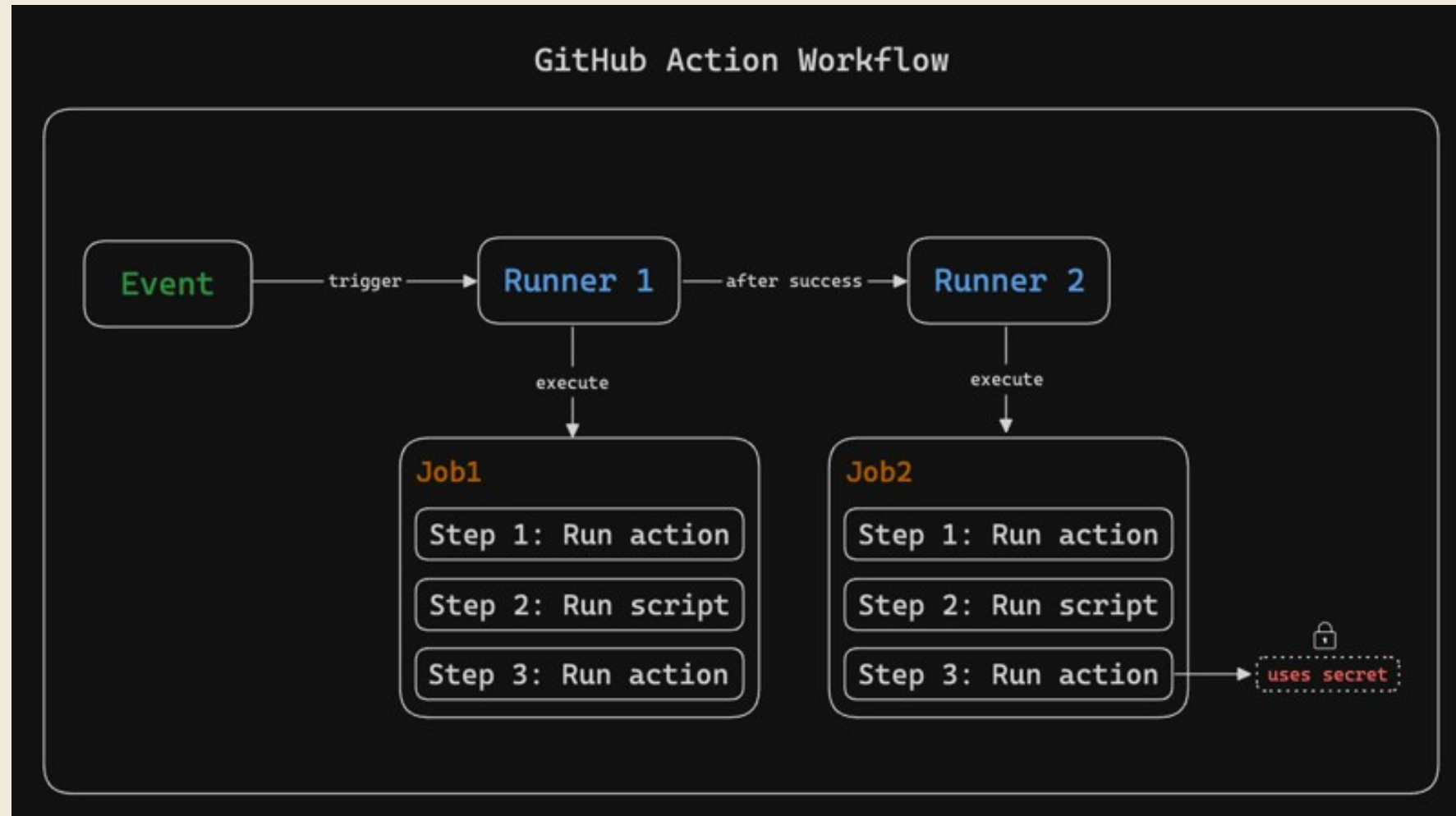
¿Qué hacen los desarrolladores que quita tanto tiempo o es tan tedioso para automatizarlo?



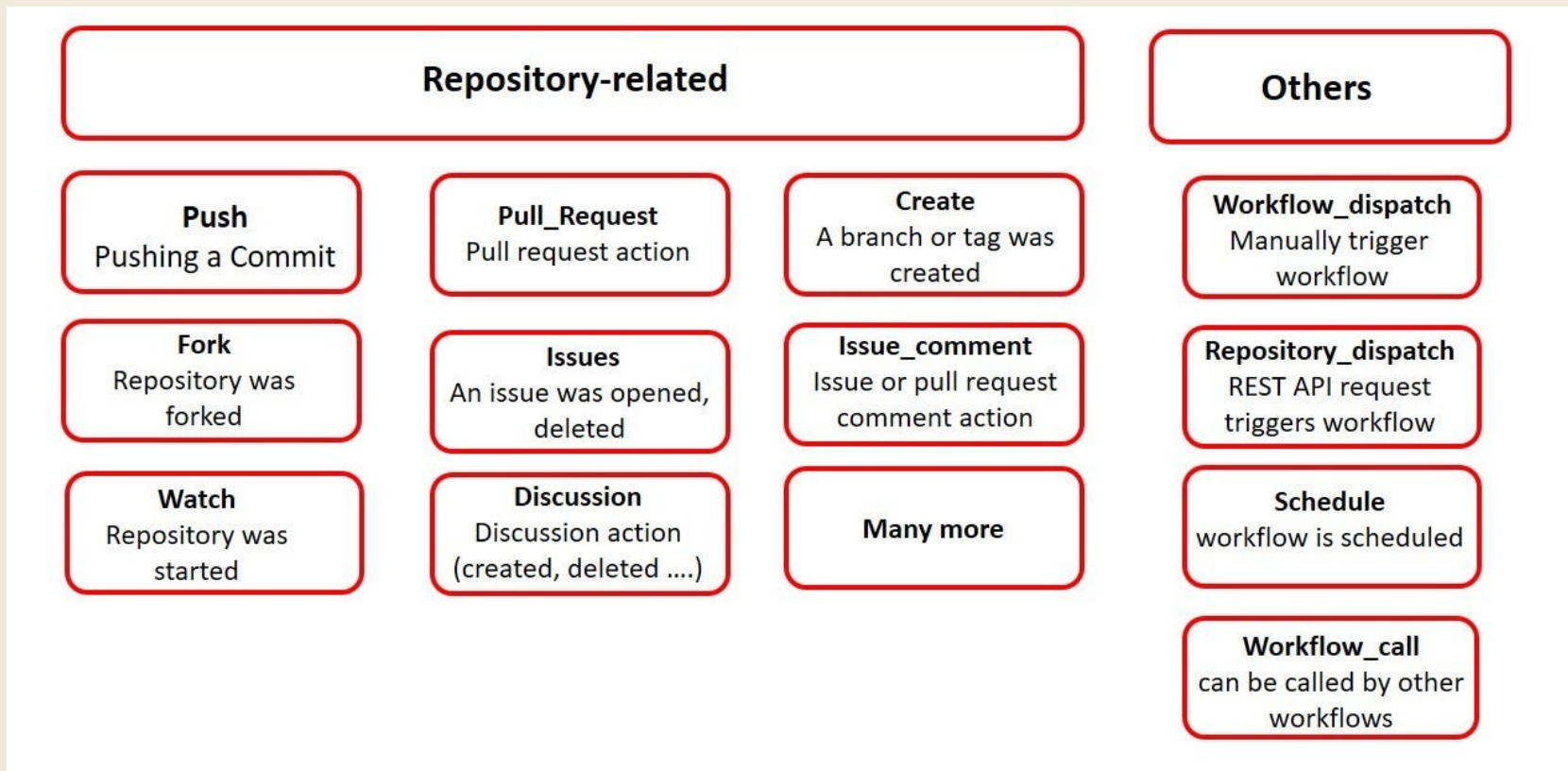
¿Qué tipos de flujos se automatizan?



¿Cómo?



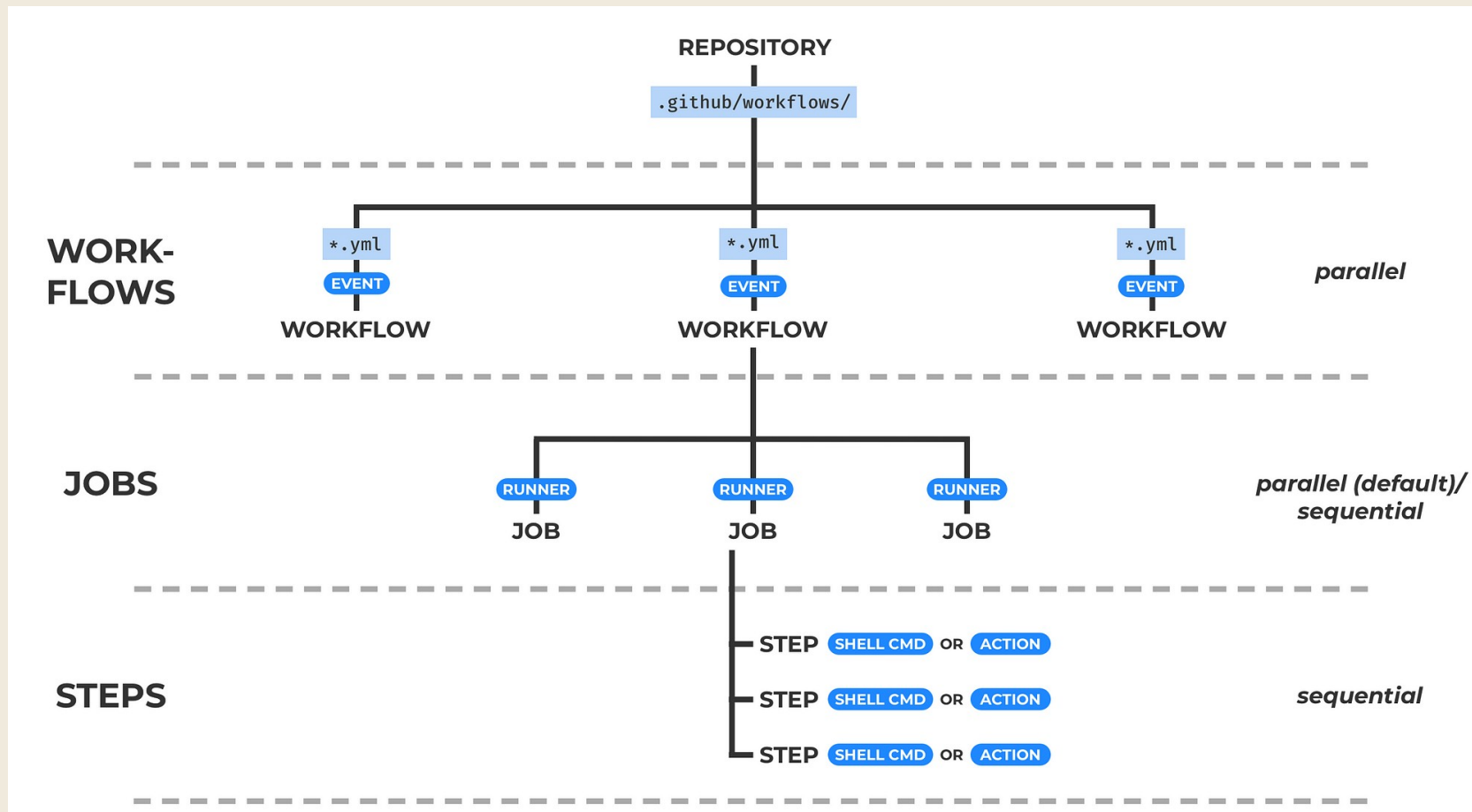
Eventos comunes



<https://medium.com/@sangeetv09/understanding-more-about-event-triggers-event-filters-and-activity-types-in-github-actions-8bfaec5b101e>

<https://docs.github.com/en/actions/writing-workflows/choosing-when-your-workflow-runs/events-that-trigger-workflows>

¿Cómo opera?



<https://itnext.io/getting-started-with-github-actions-fe94167dbc6d>

Ejemplo archivo .yaml

```
name: ??? # 1. Dale un nombre descriptivo a tu workflow

on:
  ??? : # 2. ¿Qué evento(s) quieres usar para disparar este workflow? (añade mínimo 2)
      # Piensa en cuándo quieres que se ejecute (push, pull_request, manualmente...).
  ??? : # 2b. (Opcional) ¿Quieres restringirlo a alguna rama específica?
      - main

jobs:
  imprimir_info: # No cambies este nombre, lo usaremos para evaluar
    runs-on: ??? # 3. ¿En qué sistema operativo quieres que se ejecute este job? (ubuntu-latest, windows-latest, macos-latest)

    steps:
      - uses: actions/checkout@v3 # Este paso clona tu repositorio. No lo modifiques.

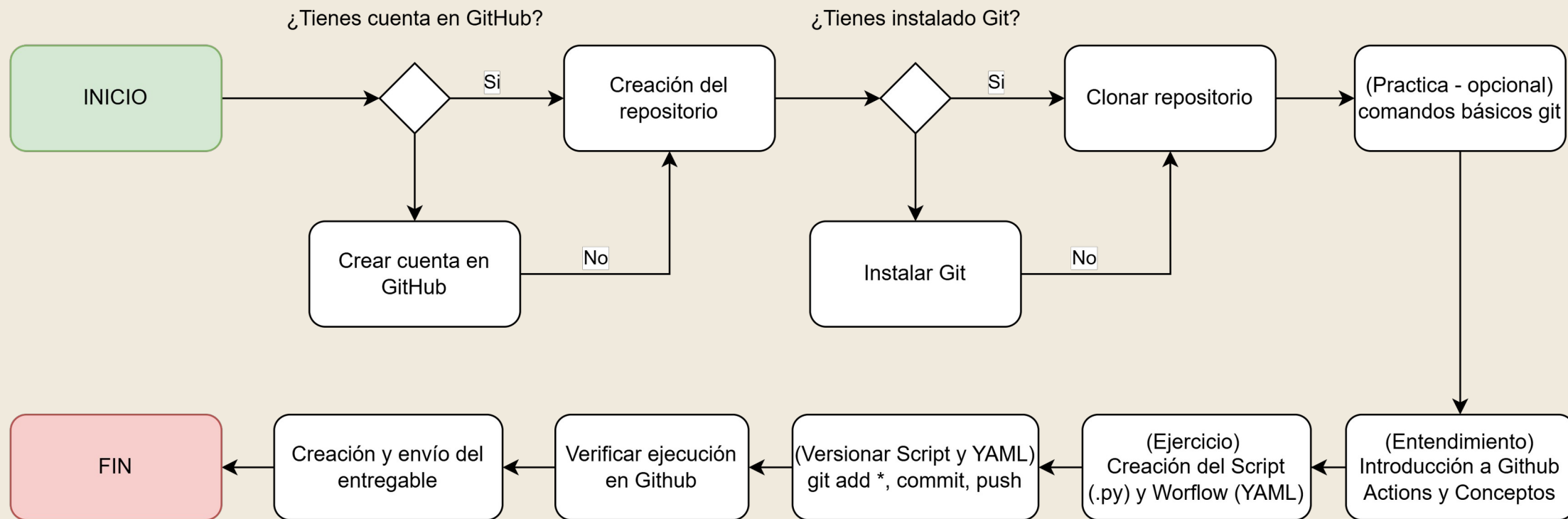
      - name: Imprimir Mensaje Personalizado
        # 4. Reemplaza [TU NOMBRE COMPLETO] con tu nombre.
        # 5. ¿Qué comando de shell usarías para obtener la fecha y hora?
        run: |
          echo "Hola, soy [TU NOMBRE COMPLETO] y este es mi primer workflow!"
          echo "La fecha y hora actual es: $(???)"

      - name: Configurar Python
        uses: actions/setup-python@v3
        with:
          python-version: ??? # 6. Elige una versión de Python (ej: '3.8', '3.9', '3.10', '3.11', '3.12').

      - name: Ejecutar script Python
        run: ??? #7. Completa para ejecutar el script de python. Pista: python <nombre_archivo>
```

TALLER EN PAREJAS ENTREGABLE 1

Introducción a CI/CD con GitHub y GitHub Actions



**LA ENTREGA SE RECIBE HASTA EL SÁBADO 06 DE SEPTIEMBRE
A LAS 11:59 PM**

TALLER EN PAREJAS ENTREGABLE 1

Introducción a CI/CD con GitHub y GitHub Actions

¿DUDAS?

¿PREGUNTAS?

¿INCONVENIENTES?




Preguntas y Discusión

¿Qué creen que pueden hacer con GitHub Actions?

¿Tienes alguna necesidad latente de automatización en tu trabajo de desarrollo de SW? ¿Cuál? ¿Qué opinas de la herramienta GitHub Actions, te ayudaría a resolverlo?

¿Les pareció sencillo o complejo la forma de automatizar con GitHub Actions?

¿Algo en particular que resaltar o consultar de este módulo de Fundamentos?



**Inspira
Crea
Transforma**

www.eafit.edu.co

VIGILADA | MINEDUCACIÓN

**UNIVERSIDAD
EAFIT**