

# MLOps

## Introducción

# DevOps: Introducción

DevOps, un término que escuchamos cada vez más en las empresas con frases como “nosotros hacemos DevOps” o “usamos herramientas DevOps”, DevOps es la contracción de las palabras “Development” (“Desarrollo”) y “Operations” (“Operaciones”).



# DevOps: Introducción

DevOps es una cultura que es diferente de las culturas corporativas tradicionales y requiere un cambio de mentalidad, procesos y herramientas. Suele asociarse con prácticas de integración continua (CI) y entrega continua (CD), que son prácticas de ingeniería de software, pero también con Infraestructura como código (IaC), que consiste en codificar la estructura y configuración de la infraestructura.

# DevOps: Introducción

El término DevOps fue introducido en 2007-2009 por Patrick Debois, Gene Kim y John Willis, y representa la combinación de Desarrollo (Dev) y Operaciones (Ops). Ha dado lugar a un movimiento que aboga por reunir a los desarrolladores y las operaciones en equipos. Esto brinda valor comercial agregado a los usuarios de manera más rápida, lo que lo hace más competitivo en el mercado.

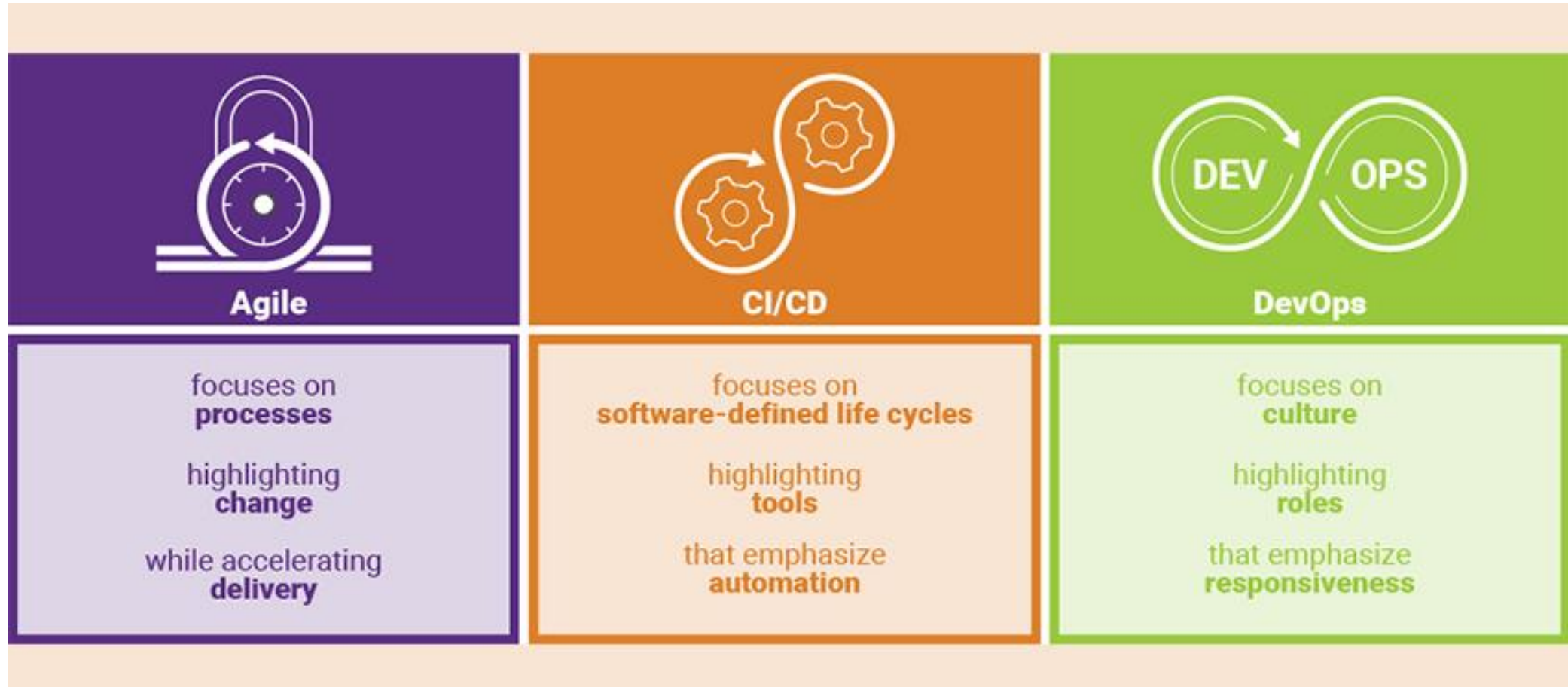
# DevOps: Introducción

La cultura DevOps es un conjunto de prácticas que reducen las barreras entre los desarrolladores, que quieren innovar y entregar más rápido, y las operaciones, que quieren garantizar la estabilidad de los sistemas de producción y la calidad de los cambios que realizan en el sistema.

# DevOps: Introducción

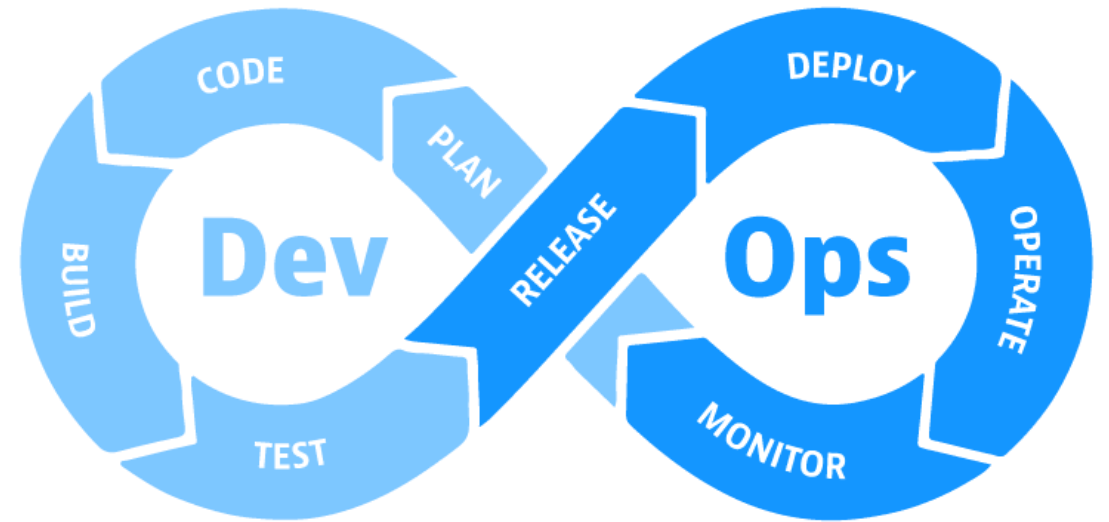
La cultura DevOps es también la extensión de los procesos ágiles (Scrum, XP, etc.), lo que permite reducir los tiempos de entrega y ya involucra a desarrolladores y equipos comerciales. Sin embargo, a menudo se ven obstaculizados por la no inclusión de Ops en los mismos equipos.

# DevOps: Introducción



# DevOps: Introducción

La comunicación y este vínculo entre Dev y Ops permite un mejor seguimiento de los despliegues de producción de extremo a extremo y despliegues más frecuentes y de mayor calidad, ahorrando dinero para la empresa.





# DevOps: Introducción

Para facilitar esta colaboración y mejorar la comunicación entre Dev y Ops, hay varios elementos clave en los procesos:

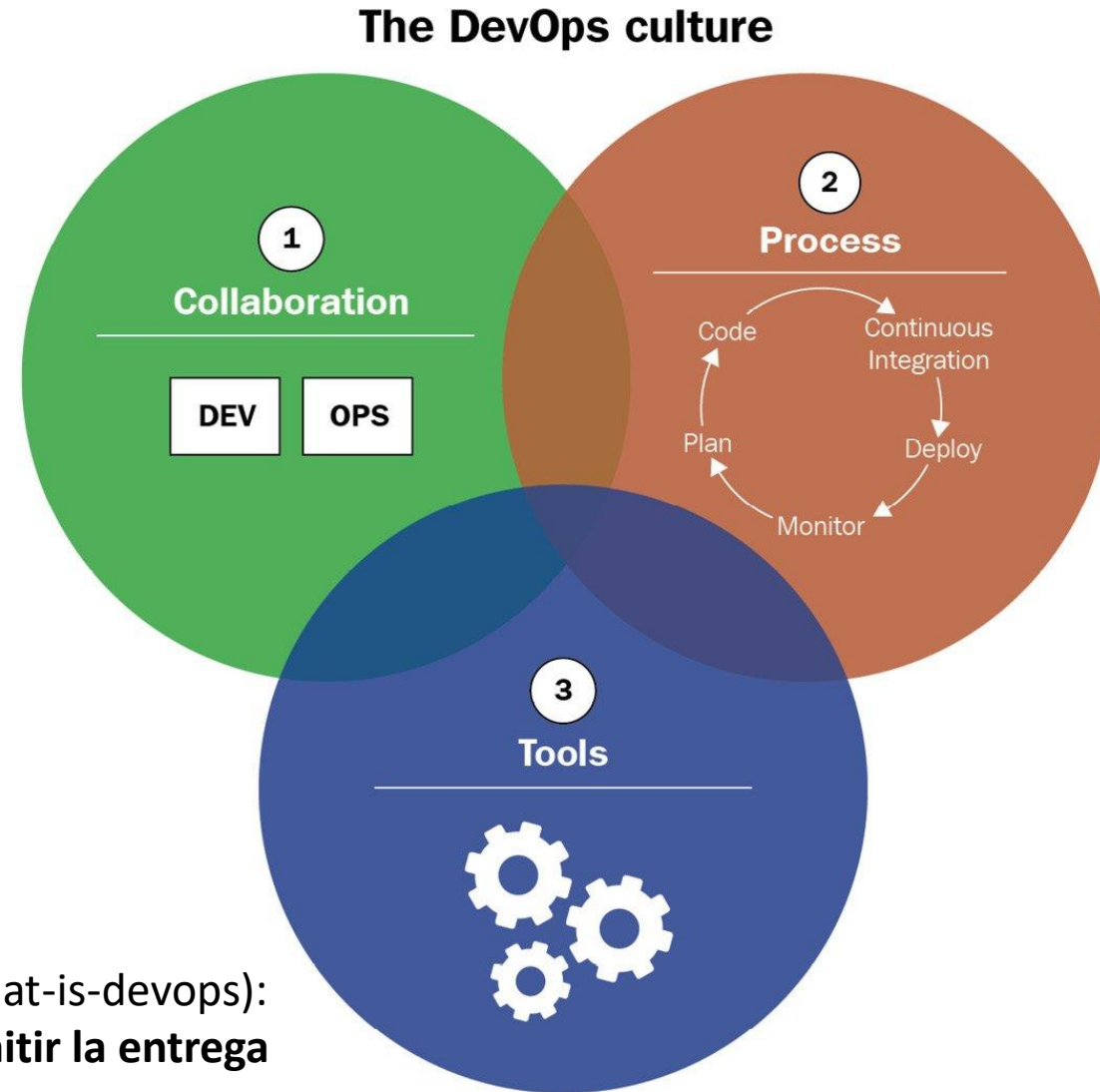
- Despliegues de aplicaciones más frecuentes con integración y entrega continua (llamados CI/CD).
- La implementación y automatización de pruebas unitarias y de integración, con un proceso enfocado al diseño dirigido por comportamiento (BDD) o diseño dirigido por pruebas (TDD).
- La implementación de un medio para recopilar comentarios de los usuarios.
- Monitoreo de aplicaciones e infraestructura.

# DevOps: Introducción

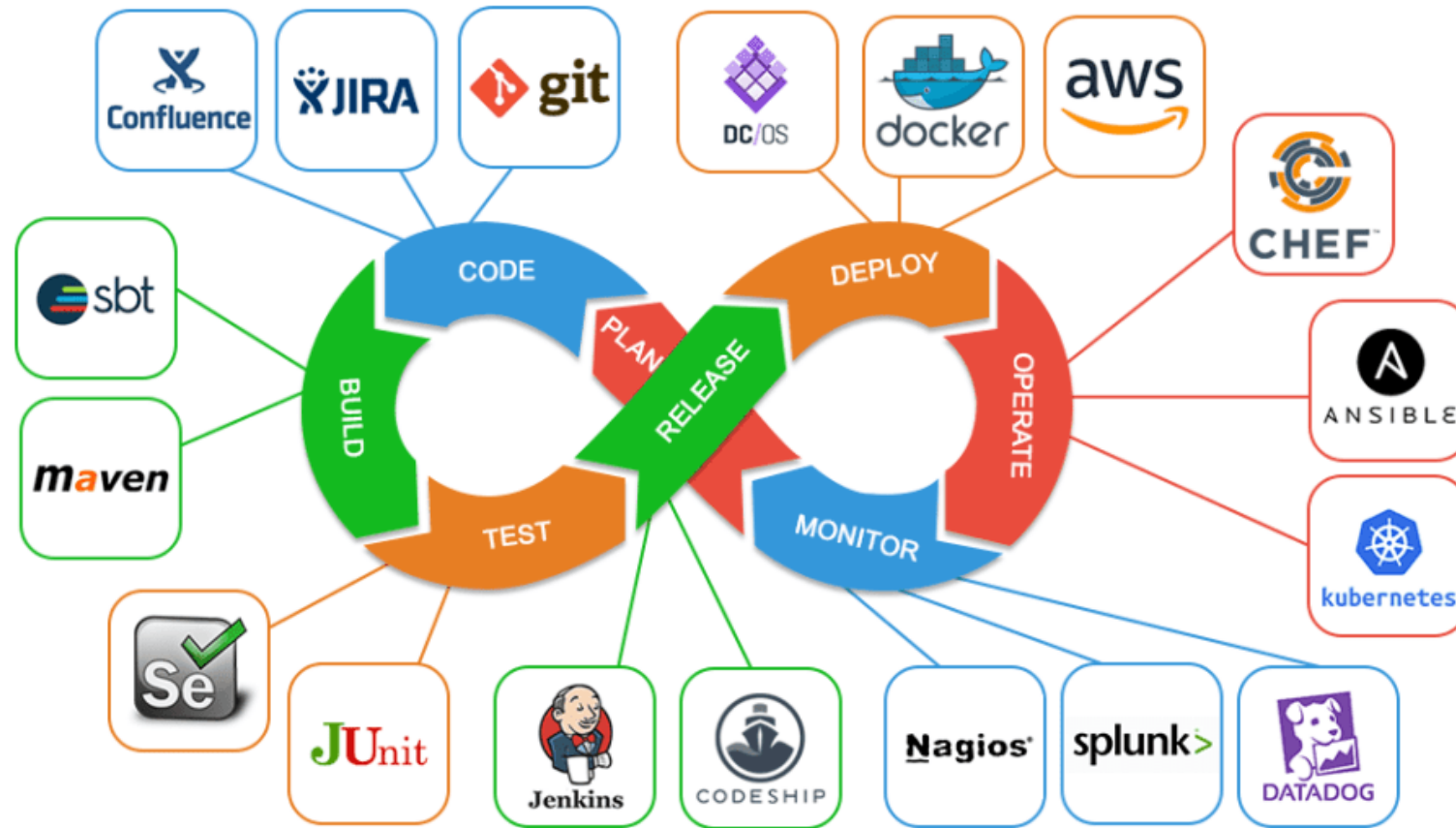
La cultura DevOps se basa en tres ejes:

- La cultura de la colaboración
- Procesos
- Herramientas

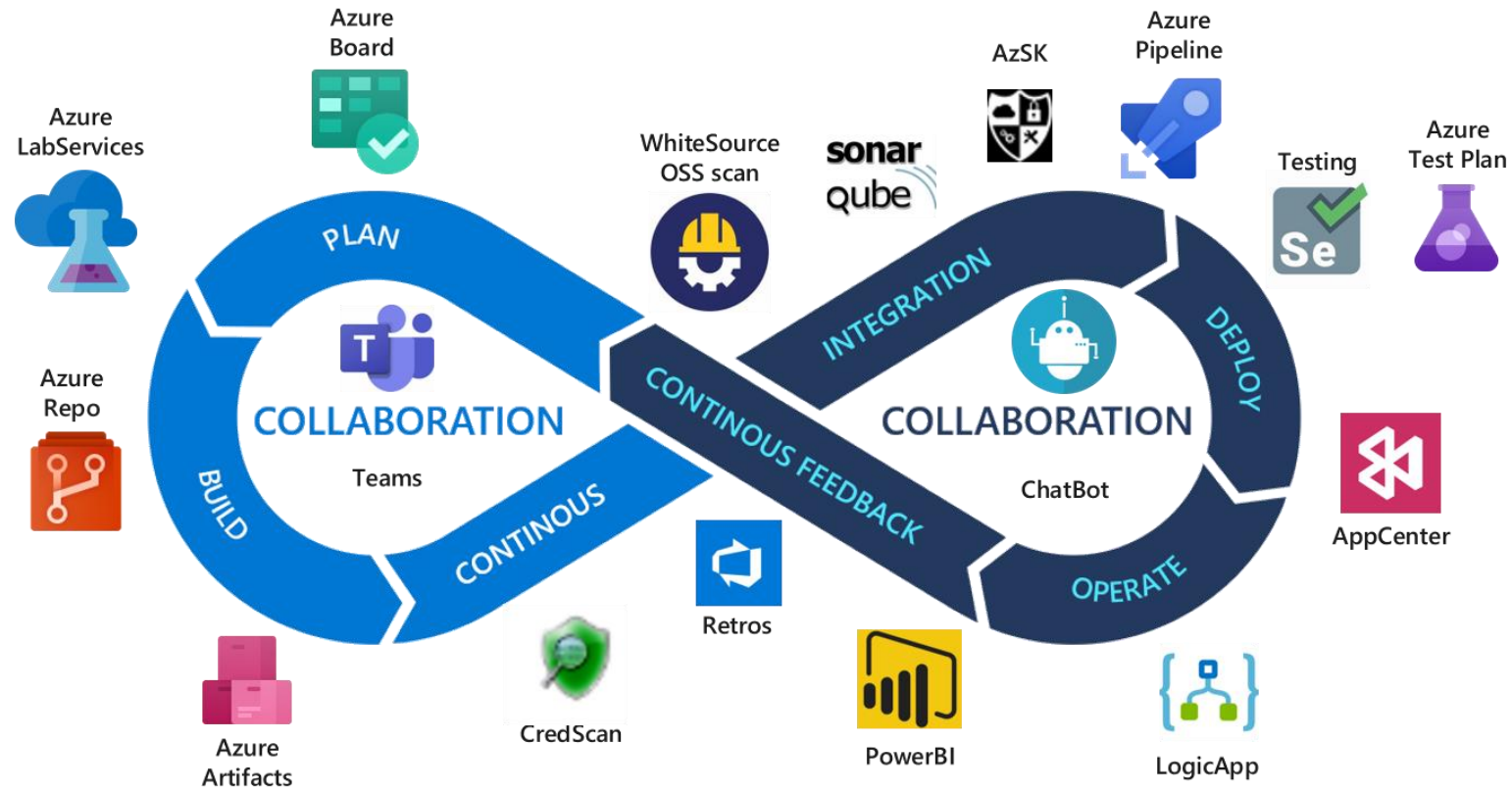
Definición de Donovan Brown (<http://donovanbrown.com/post/what-is-devops>):  
"DevOps es la unión de personas, procesos y productos para permitir la entrega continua de valor a nuestros usuarios finales".



# DevOps: Introducción



# DevOps: Introducción



# Evolución de la infraestructura y el desarrollo de software

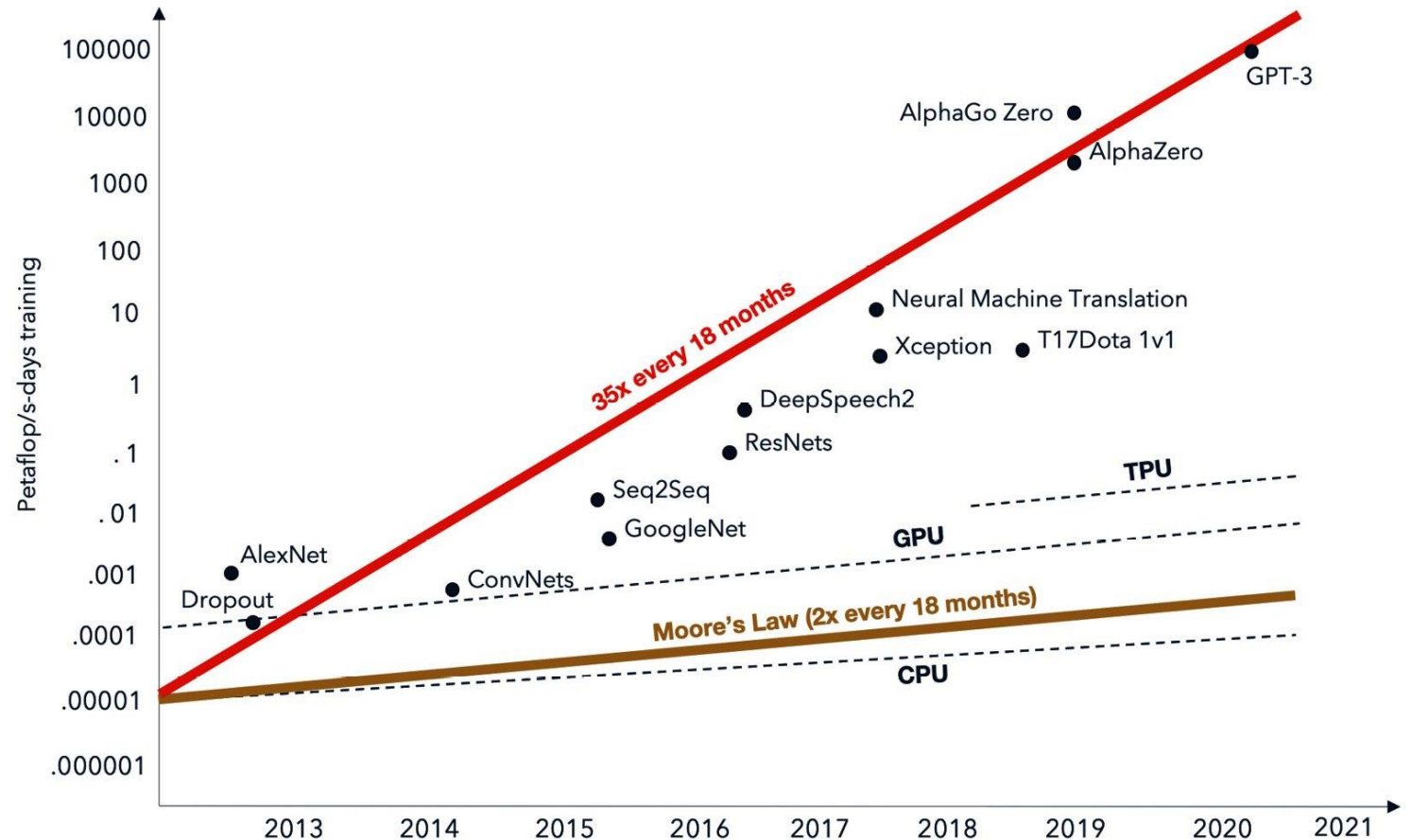
- Con la génesis de la era moderna de Internet (alrededor de 1995), fuimos testigos de un aumento en las aplicaciones de software, desde sistemas operativos como Windows 95 hasta el sistema operativo Linux y sitios web como Google y Amazon, que han estado sirviendo al mundo (en línea ) durante más de dos décadas.

# Evolución de la infraestructura y el desarrollo de software

- En la última década, muchas empresas a escala global y regional han catalizado la transformación de la nube. Como resultado, se ha producido un cambio de la informática localizada (empresas que tienen sus propios servidores y centros de datos) a la informática bajo demanda debido a la disponibilidad de servicios en la nube robustos y escalables. Ahora las empresas y organizaciones pueden proporcionar recursos a pedido en la nube para satisfacer sus necesidades de procesamiento de datos.

# Crecimiento de Machine Learning y Deep Learning

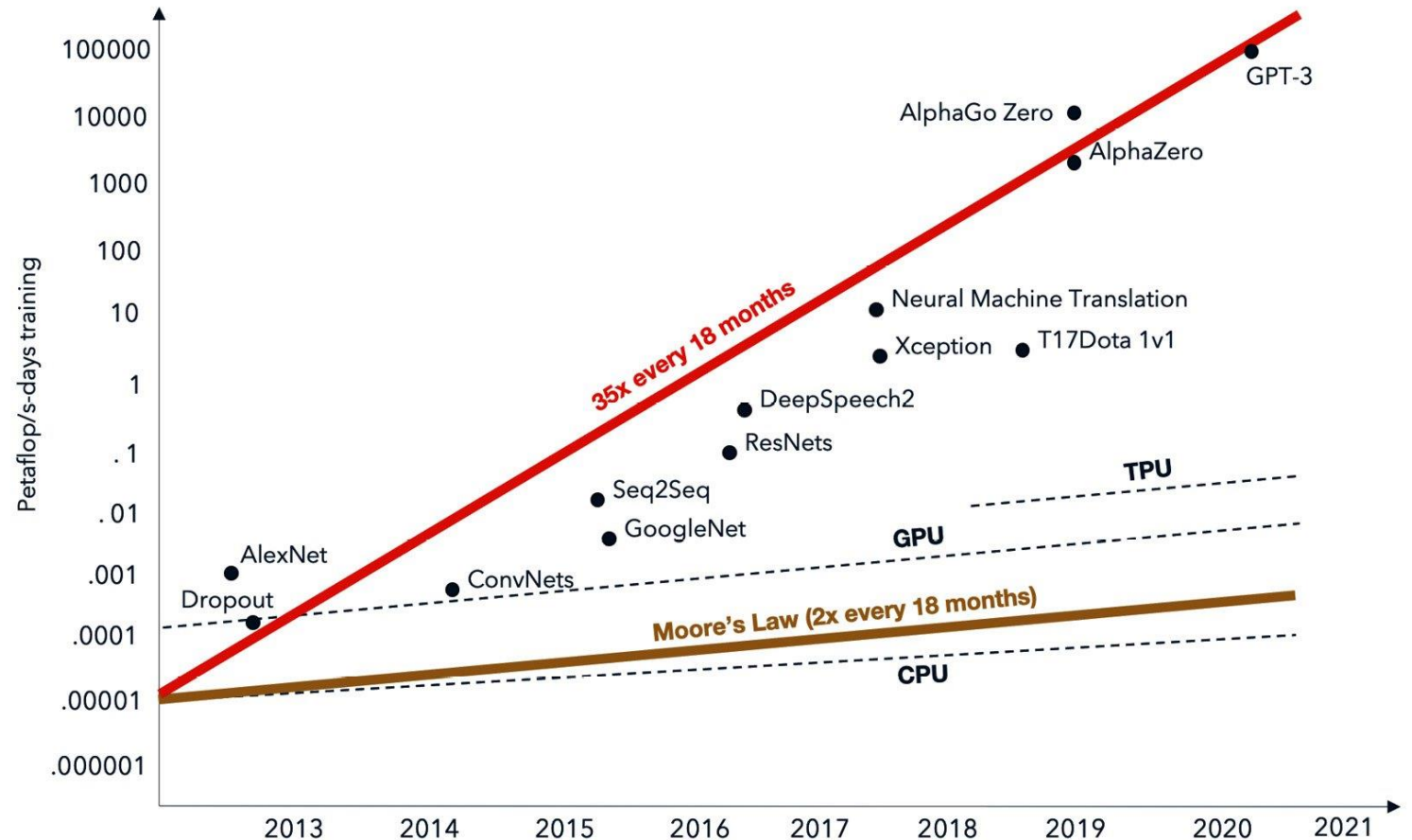
Los avances en el aprendizaje profundo son posibles gracias al crecimiento exponencial de la informática, que aumenta alrededor de 35 veces cada 18 meses. De cara al futuro, con tales demandas podemos encontrar obstáculos en términos de escalar la informática central para CPU, GPU o TPU.





# Crecimiento de Machine Learning y Deep Learning

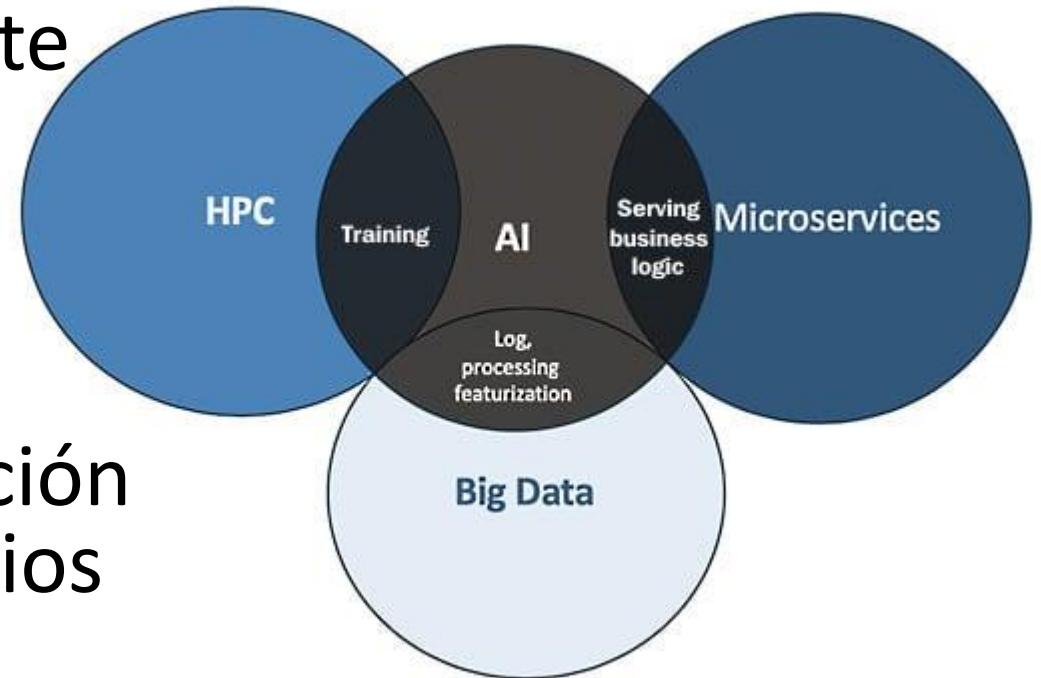
Esto nos ha obligado a buscar alternativas, como el aprendizaje distribuido, donde la computación para el procesamiento de datos se distribuye en múltiples nodos de computación. Hemos visto algunos avances en el aprendizaje distribuido, como el Aprendizaje Federado y Edge Computing. El aprendizaje distribuido se ha mostrado prometedor para satisfacer las crecientes demandas del aprendizaje profundo.





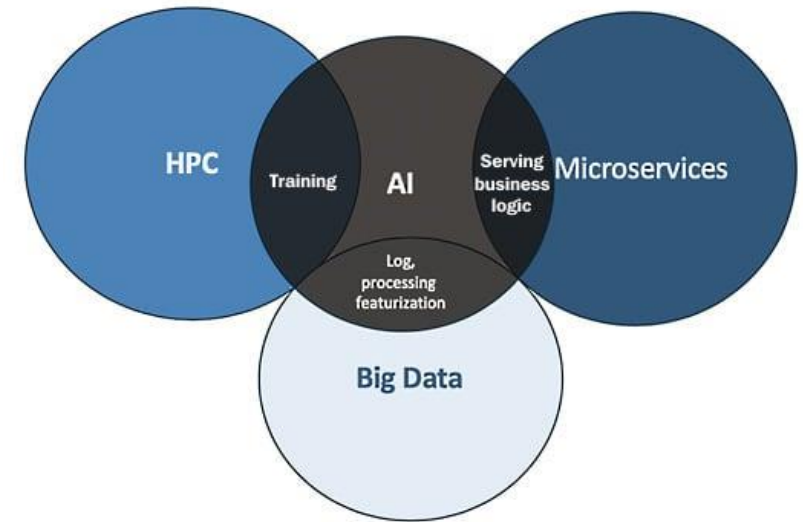
# Aplicaciones centradas en IA

Las aplicaciones se están volviendo centradas en la IA; lo vemos en múltiples industrias. Prácticamente todas las aplicaciones están comenzando a usar IA, y estas aplicaciones se ejecutan por separado en cargas de trabajo distribuidas como HPC (computación de alto rendimiento), microservicios y big data.



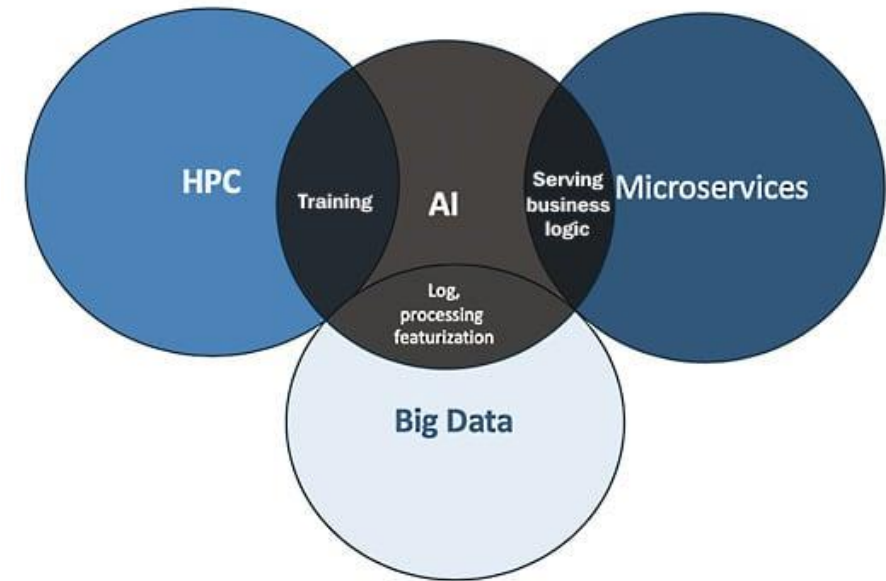
# Aplicaciones centradas en IA

- Al combinar HPC e IA, podemos habilitar los beneficios de la computación necesarios para entrenar modelos de aprendizaje profundo y ML.
- Con la superposición de big data e IA, podemos aprovechar la extracción de datos necesarios a escala para el entrenamiento del modelo de IA,
- Con la superposición de microservicios e IA podemos servir los modelos de IA para inferencia para mejorar las operaciones comerciales y el impacto. De esta forma, las aplicaciones distribuidas se han convertido en la nueva norma.



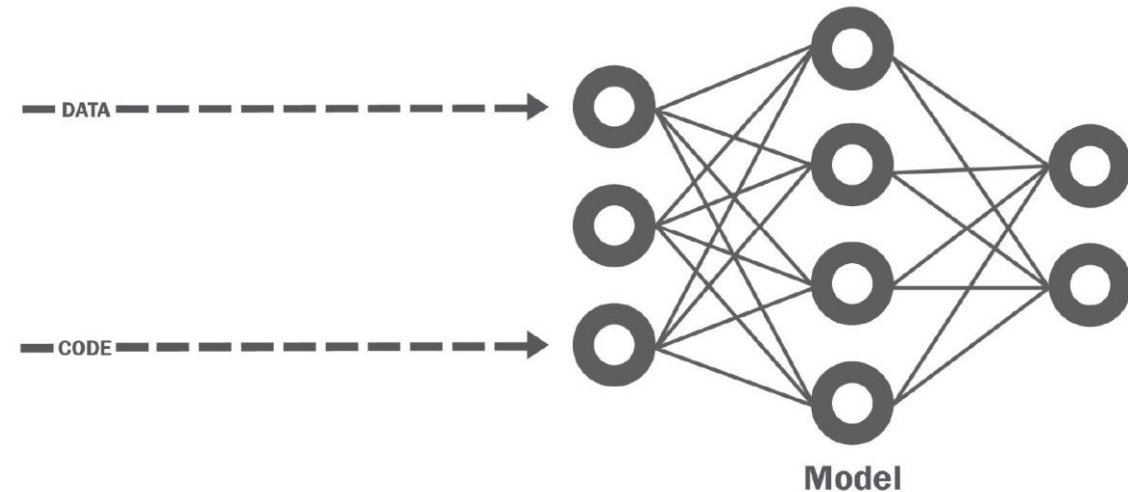
# Aplicaciones centradas en IA

- El desarrollo de aplicaciones centradas en IA a escala requiere una sinergia de aplicaciones distribuidas (HPC, microservicios y Big data) y para ello se requiere una nueva forma de desarrollar software.



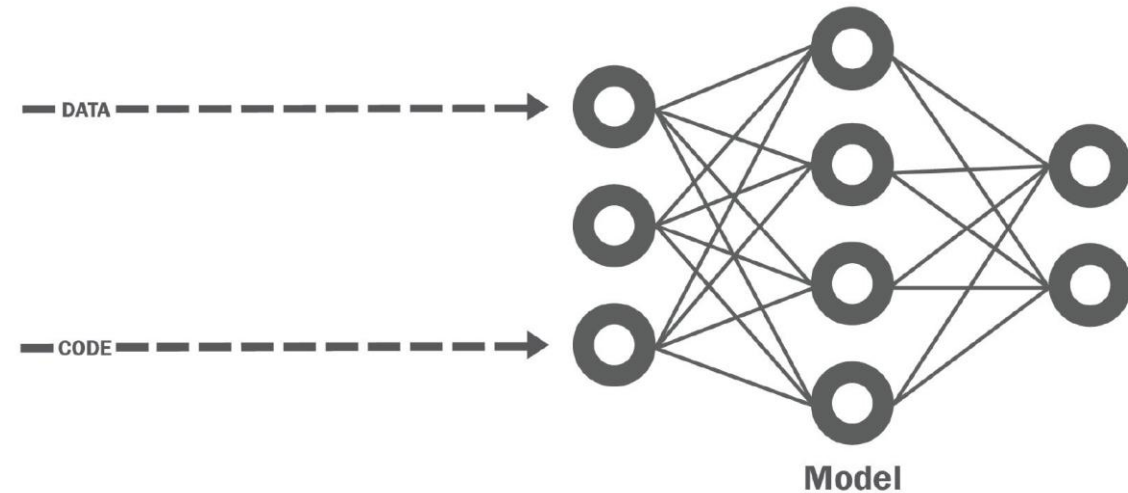
# Retos del desarrollo de software tradicional para Machine Learning

Machine Learning, no es solo código, como en el desarrollo de software tradicional, sino código más datos. Los datos son fundamentales para el modelo de ML, y el código nos permite ajustar los datos para que podamos obtener información de ellos.



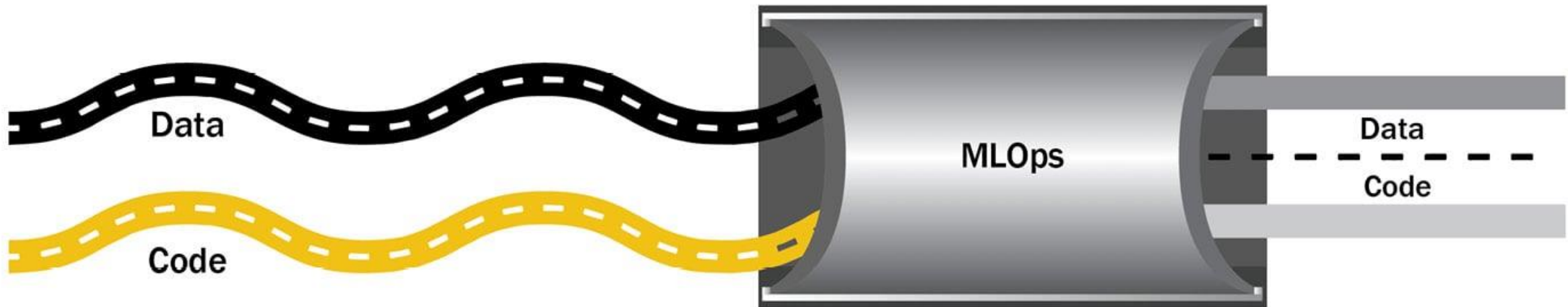
# Retos del desarrollo de software tradicional para Machine Learning

Debido a esta relación entre el código y los datos, se debe tener cuidado de unir los dos en el desarrollo para que evolucionen de manera controlada, hacia el objetivo común de un sistema ML robusto y escalable; los datos para el entrenamiento, las pruebas y la inferencia cambiarán con el tiempo, a través de diferentes fuentes, y deben cumplirse con un código cambiante.



# Retos del desarrollo de software tradicional para Machine Learning

Sin un enfoque sistemático de MLOps, puede haber divergencias en la forma en que evolucionan el código y los datos, lo que causa problemas en la producción, obstaculiza la implementación fluida y conduce a resultados que son difíciles de rastrear o reproducir:



# Retos del desarrollo de software tradicional para Machine Learning

- MLOps agiliza el proceso de desarrollo, implementación y monitoreo de aplicaciones de ML, unificando las contribuciones de los diferentes equipos involucrados y asegurando que todos los pasos del proceso se registren y se puedan repetir.
- MLOps habilita y empodera a los equipos de ciencia de datos y TI, para que puedan colaborar en la construcción y mantenimiento de sistemas de ML robustos y escalables.

# MLOps

El desarrollo de software es multidisciplinario y está cambiando para facilitar ML. MLOps es un nuevo enfoque para fusionar ML y el desarrollo de software mediante la combinación de diferentes dominios. MLOps combina ML, DevOps e ingeniería de datos, con el objetivo de construir, implementar y mantener los sistemas de ML en producción de manera confiable y eficiente. Por lo tanto, MLOps puede explicarse por esta intersección.



# MLOps

