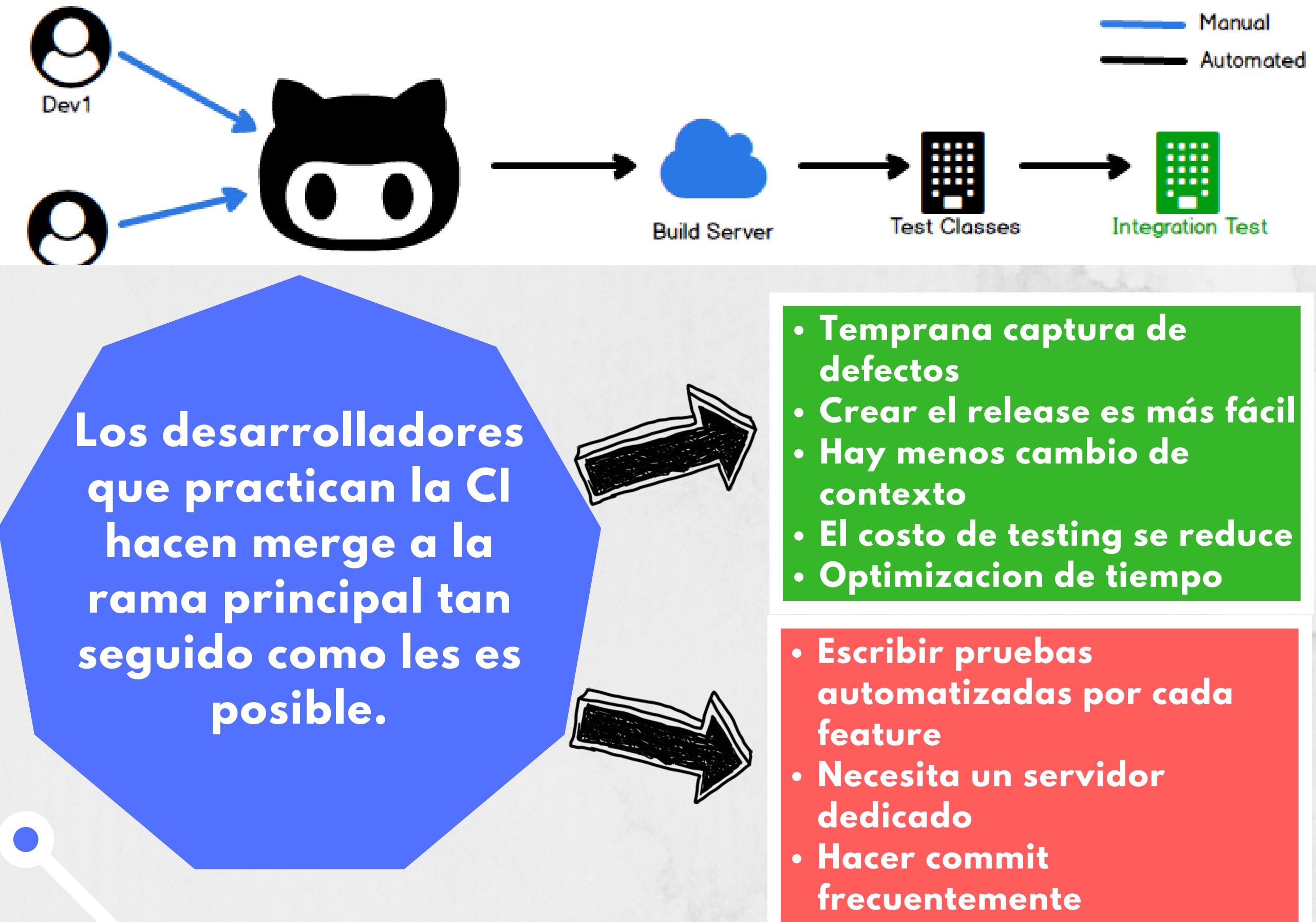


# Estrategias de Despliegue

## Continuous integration [Integración Continua]



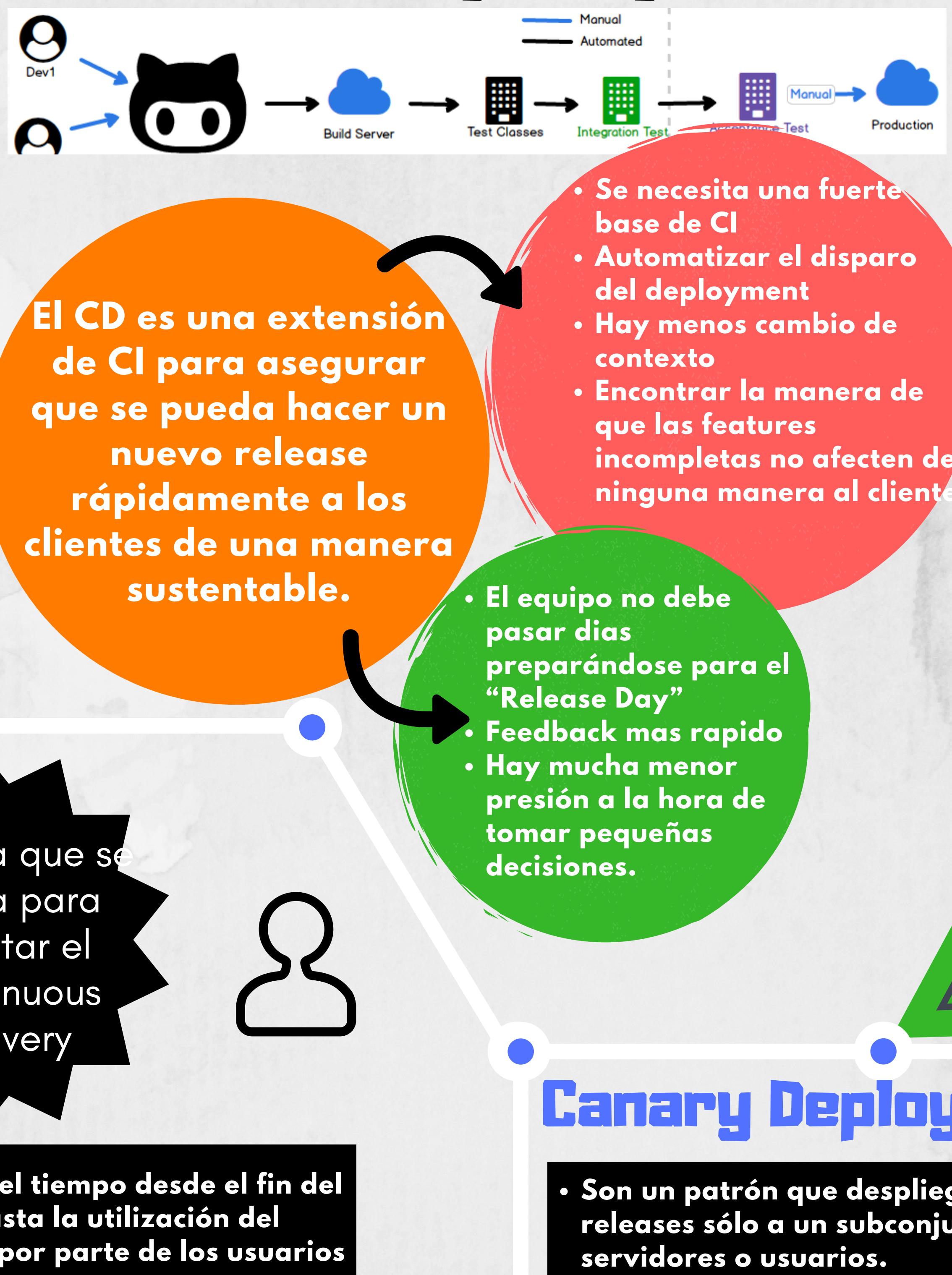
GRUPO N° 6

- Abanto, Diego
- Bracamonte, Gabriel
- Crespo, Mickaela
- García, Eliana
- Pérez, Juan

INTRODUCCIÓN:

Continuous Integration, Continuous Delivery y Continuous Deployment son estrategias de la Ingeniería de Software utilizadas por los desarrolladores para agilizar y automatizar el proceso de despliegue. Se presentarán características, ventajas y desventajas de cada una de ellas con el fin de comprender la importancia y el beneficio de su implementación.

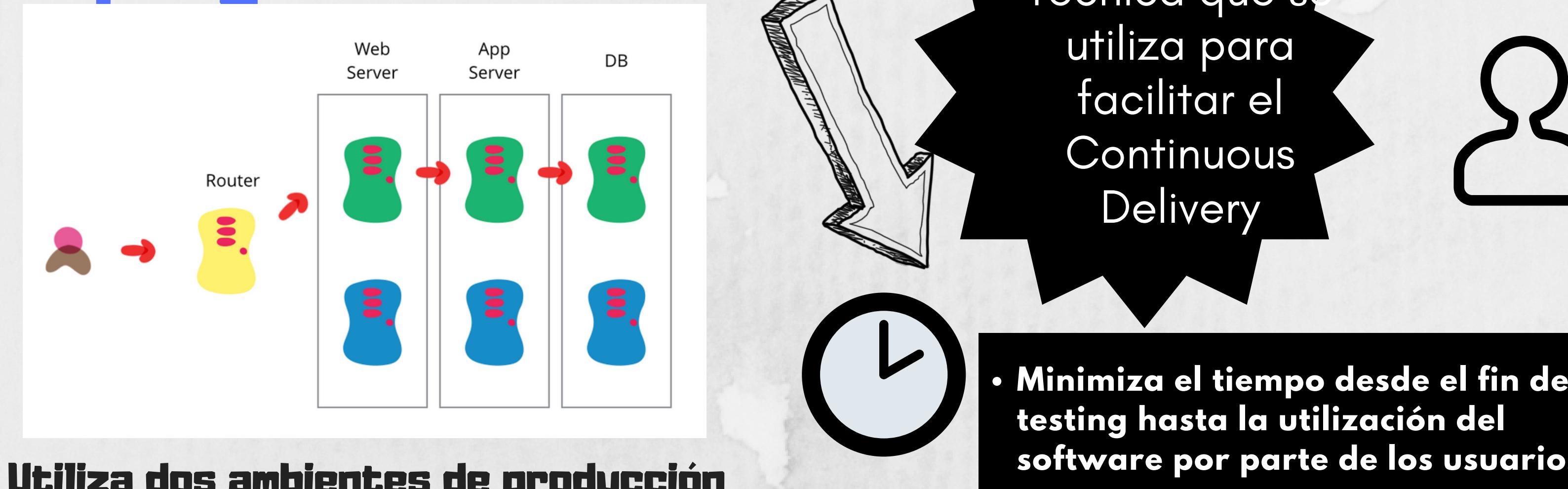
## Continuous delivery [Entrega Continua]



## Continuous deployment [Despliegue Continuo]



## Blue-Green Deployment [Despliegue Azul-Verde]



**Utiliza dos ambientes de producción**

1. el 1er ambiente se encuentra en "Live-Production", el cliente los utiliza activamente de manera diaria,
2. en el otro ambiente se prepara el nuevo release y se realizan las últimas instancias de testina.

**Ventaja:** Esta técnica da una forma rápida de rollback ya que si algo sale mal en el ambiente recientemente puesto en "Live Production" se puede volver al que se acaba de sacar de producción.

**Desventaja:** Puede haber un problema de consistencia en la base de datos a la hora de volver al servidor que no tiene la actualización (se pueden idear mecanismos para evitarlo)

## Canary Deployments

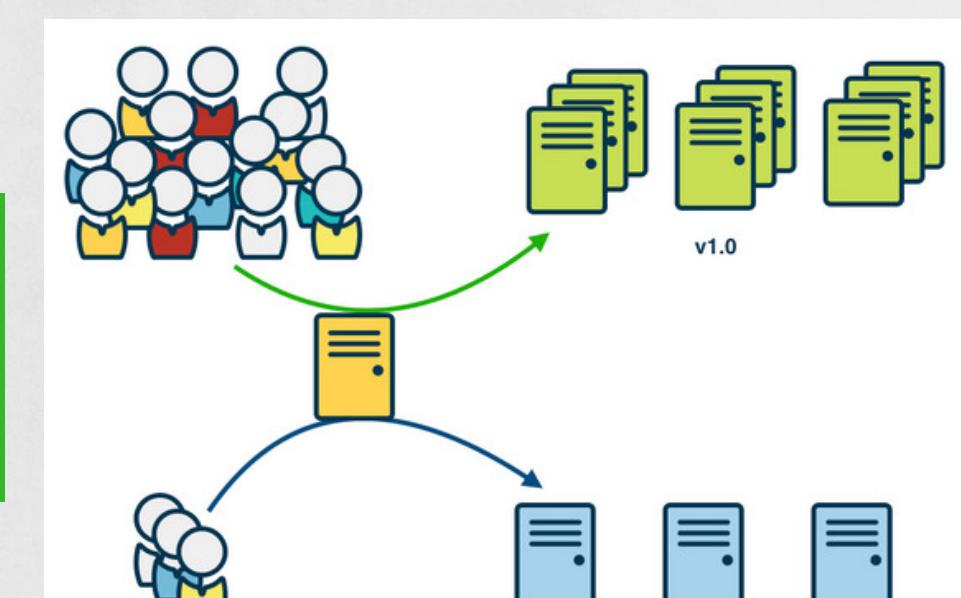
- Son un patrón que despliega releases sólo a un subconjunto de servidores o usuarios.

**PROCESO:**

- Deplegar al servidor canario
- Ejecutar tests
- Desplegar en el resto de servidores

**herramienta automatizada para continuous deployment**

Funciona como un sistema de "alerta temprana" reduciendo el impacto del error, ya que solo afecta a un subconjunto.



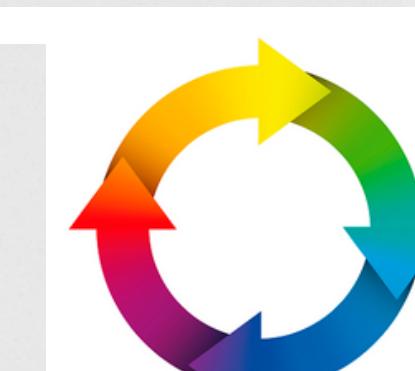
## Continuous deployment in 5 easy steps

Es un proceso mediante el cual todo el código que se escribe para una aplicación se implementa inmediatamente en producción

Reducción del tiempo del ciclo y la liberación de trabajo individual.

Para solucionar las desventajas, se proponen 5 pasos

- Servidor de integración continua
- Servidor de verificación de commits
- Script de implementación
- Alerta en tiempo real
- Análisis de la causa raíz



## CONCLUSION:

La implementación de las estrategias presentadas representan grandes beneficios en el desarrollo de software, ya que reducen costos, tiempos y complejidad, permitiendo destinar más esfuerzos en aspectos relacionados con la calidad del producto. Sin embargo, para gozar de estos beneficios, primero es necesario adaptar la forma de trabajo de todas las personas involucradas. Entender los conceptos de la implementación continua es fundamental para lograr un exitoso despliegue del producto.