



# Ciencia de Datos y BigData

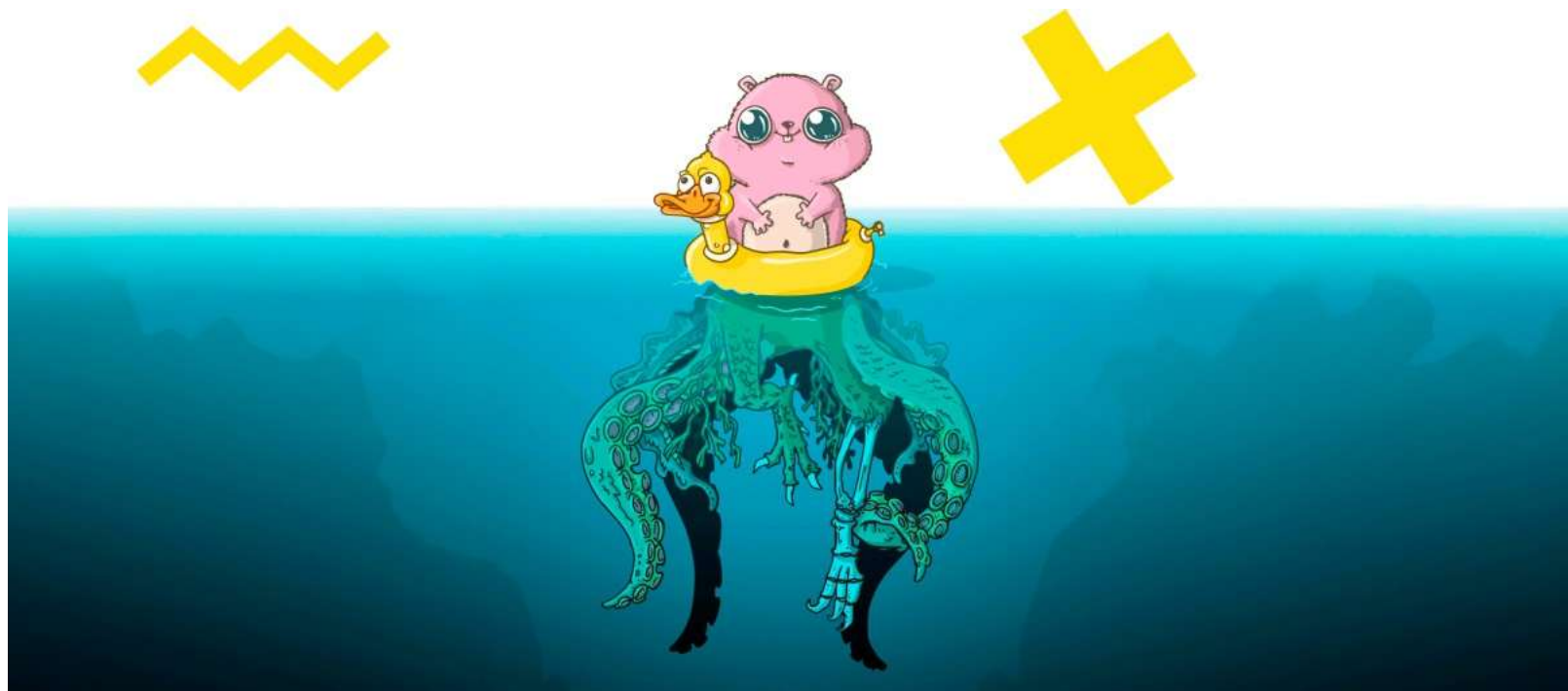
## Flujo de Desarrollo DevOps

**Dr. José Ramón Iglesias**

DSP-ASIC BUILDER GROUP  
Director Semillero TRIAC  
Ingeniería Electronica  
Universidad Popular del Cesar

- 1 [Conceptos base](#)
- 2 [Evolución de las arquitecturas](#)
- 3 [DevOps](#)
- 4 [Tareas dentro de la cultura DevOps](#)
- 5 [Ciclo de vida de una puesta a productivo](#)

- 1 [Conceptos base](#)
- 2 [Evolución de las arquitecturas](#)
- 3 [DevOps](#)
- 4 [Tareas dentro de la cultura DevOps](#)
- 5 [Ciclo de vida de una puesta a productivo](#)





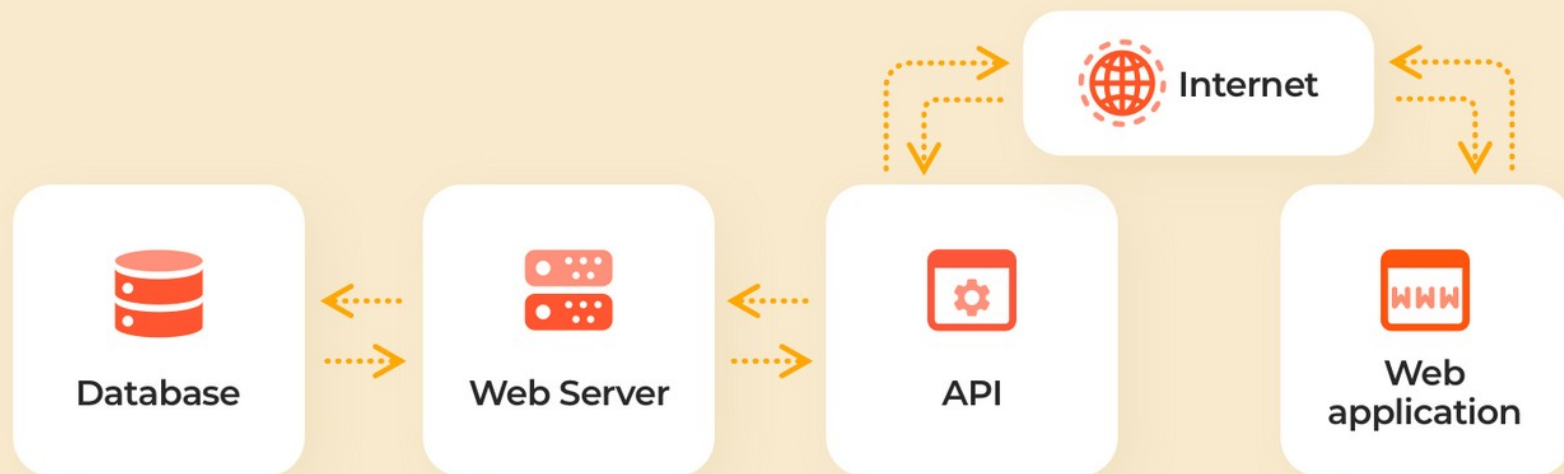
**Frontend**

**vs.**

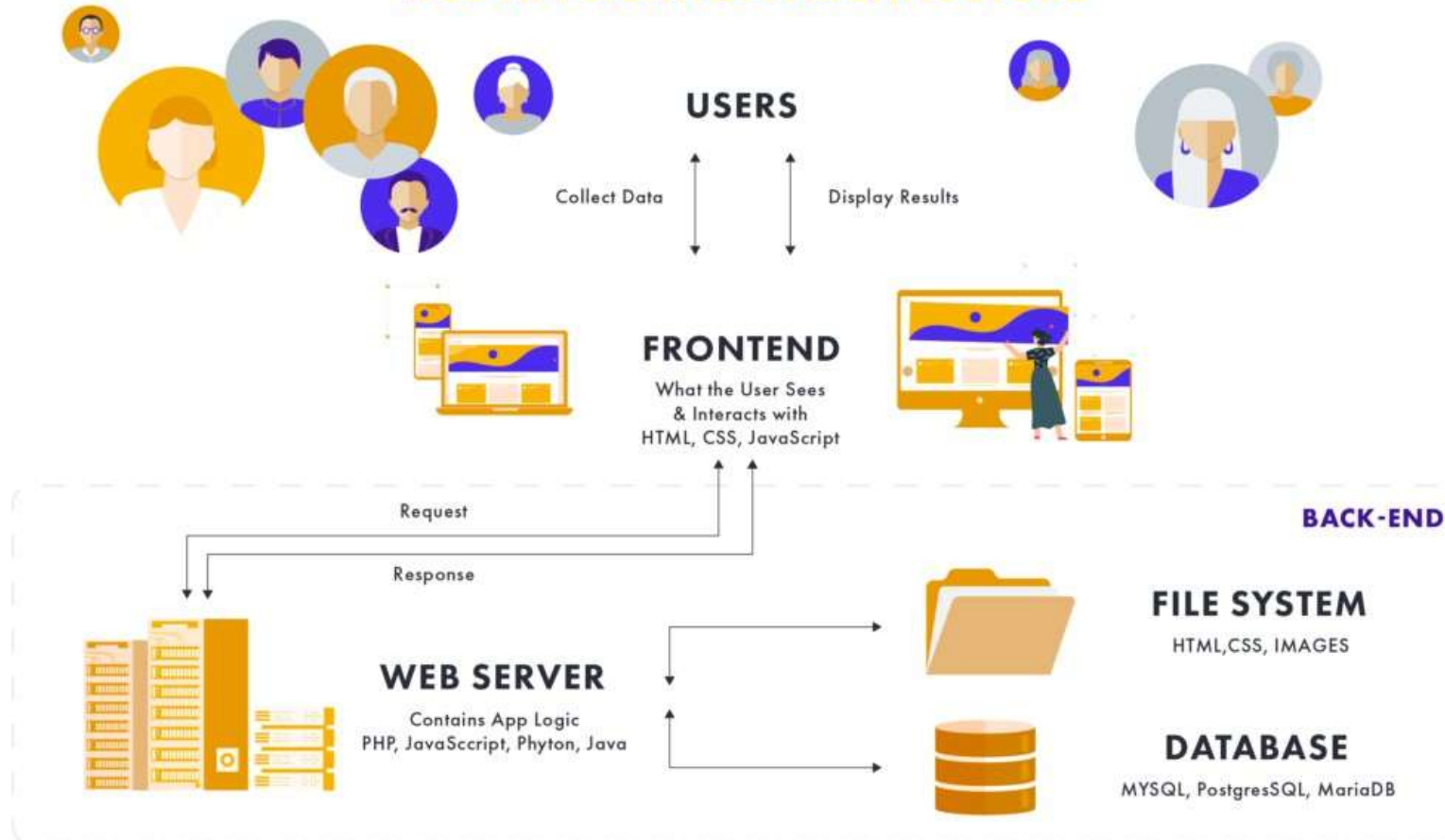


**Backend**

# What is an API?

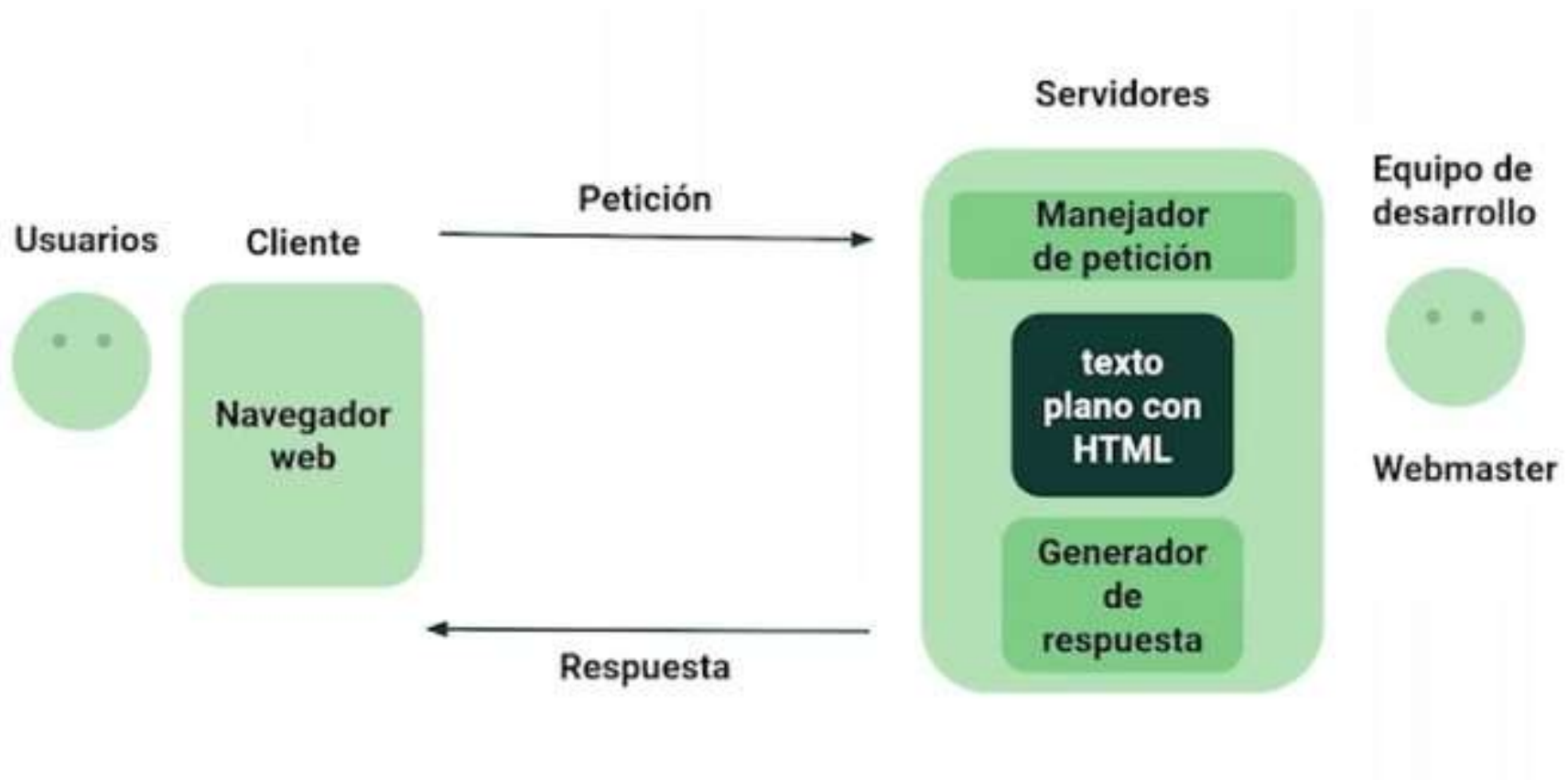


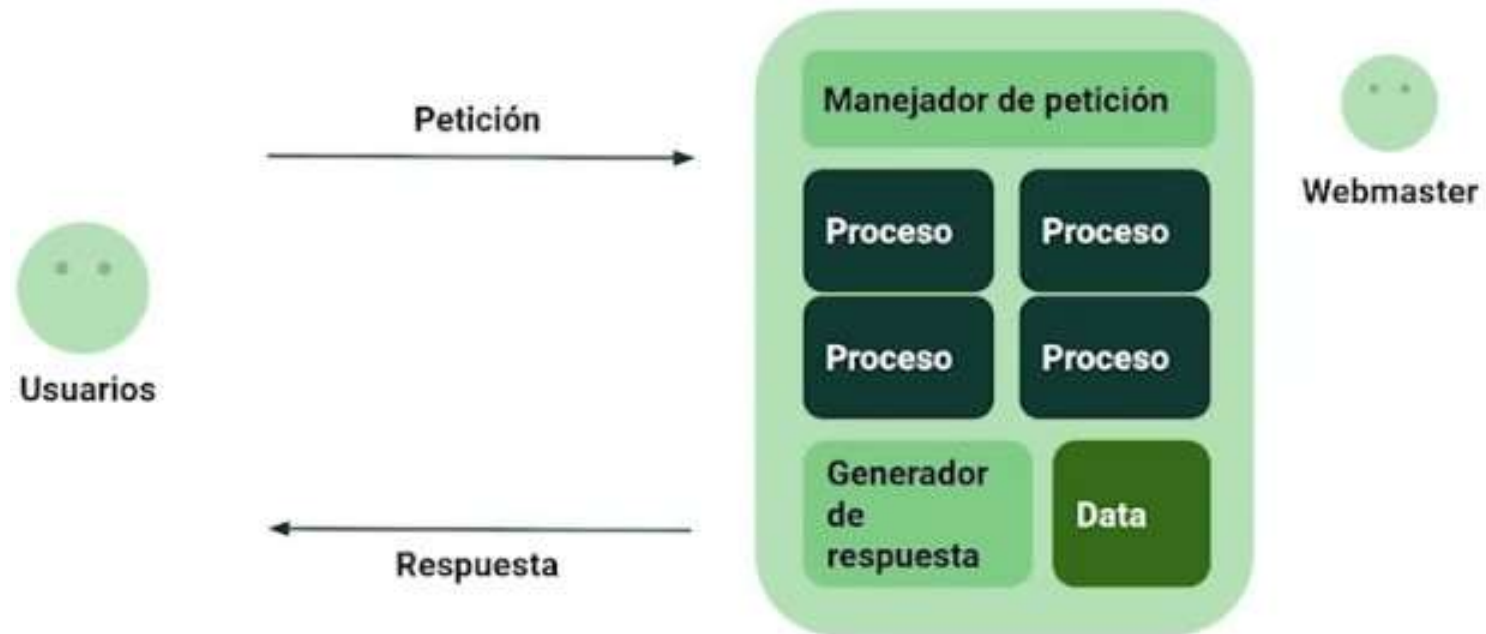
## WEB APPLICATION ARCHITECTURE

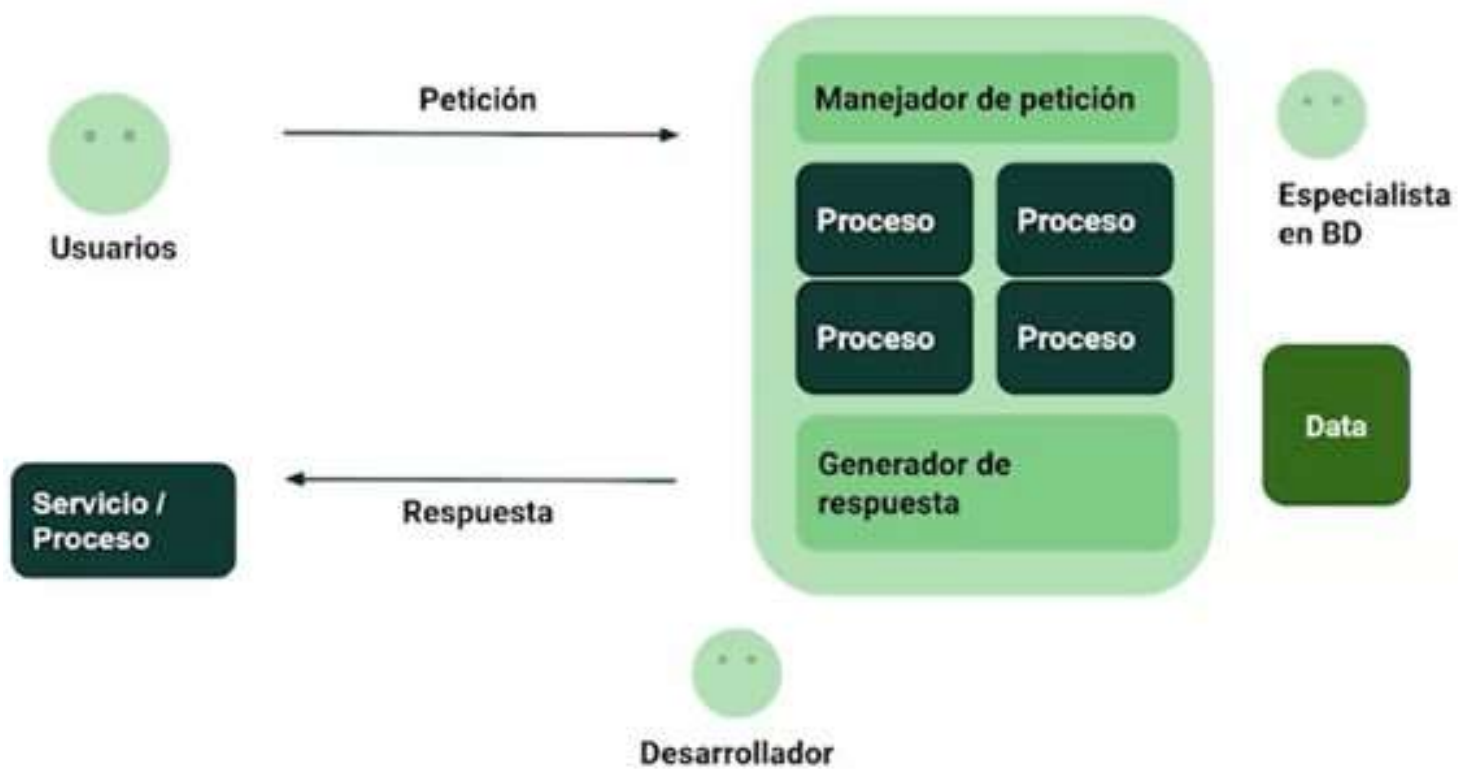


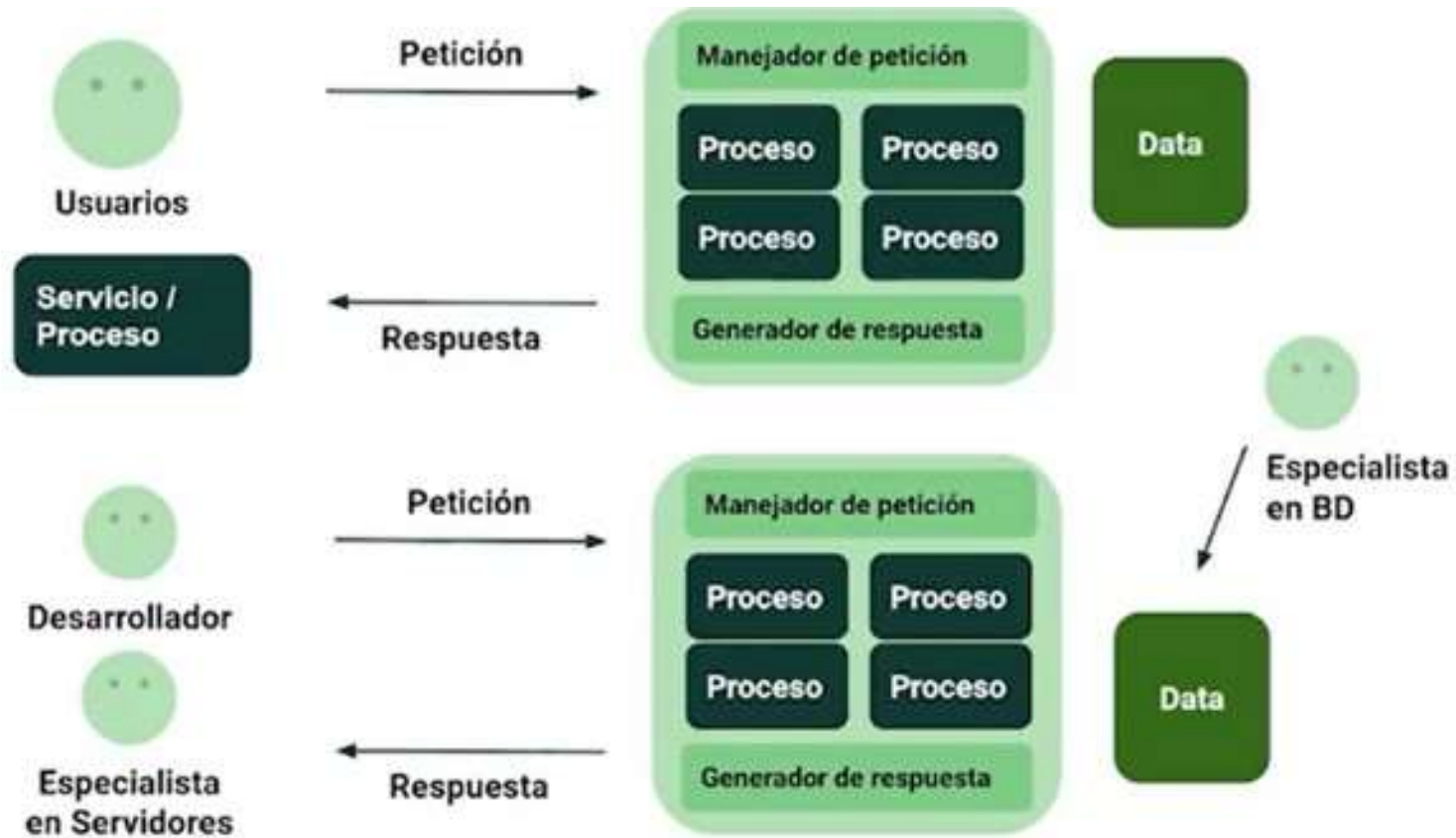
- 1 [Conceptos base](#)
- 2 [Evolución de las arquitecturas](#)
- 3 [DevOps](#)
- 4 [Tareas dentro de la cultura DevOps](#)
- 5 [Ciclo de vida de una puesta a productivo](#)

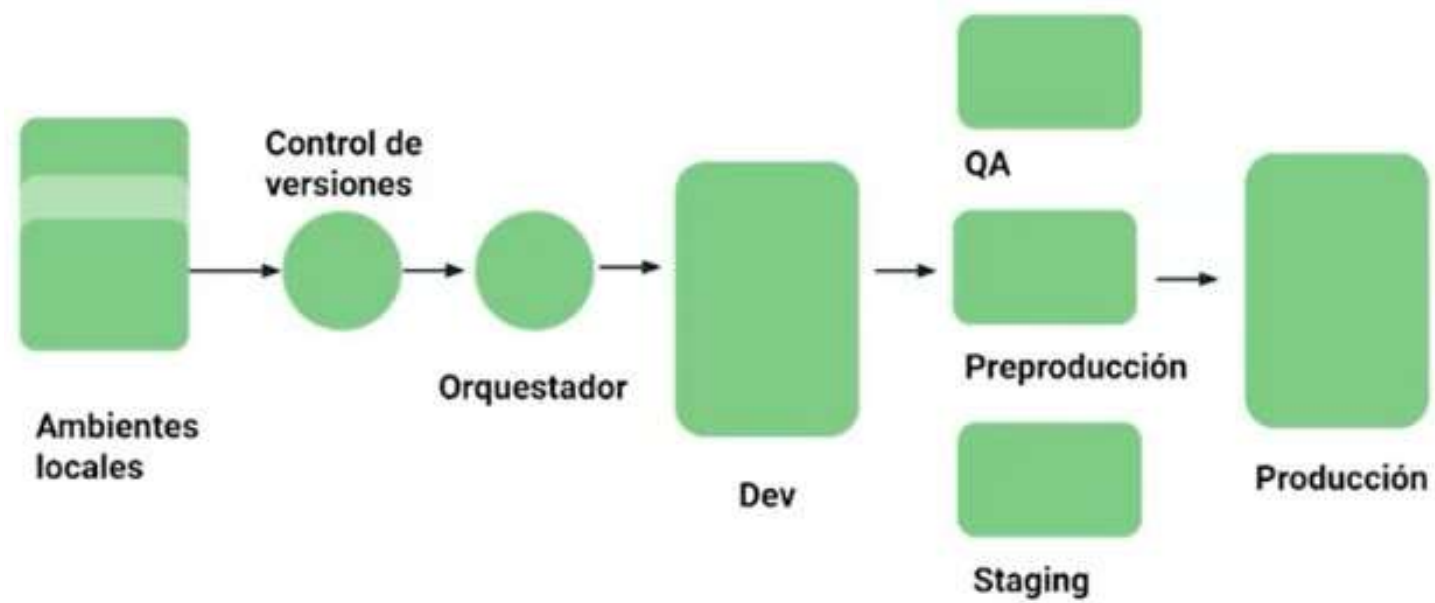












- 1 [Conceptos base](#)
- 2 [Evolución de las arquitecturas](#)
- 3 [DevOps](#)
- 4 [Tareas dentro de la cultura DevOps](#)
- 5 [Ciclo de vida de una puesta a productivo](#)

**¿Qué tiene que ver esto  
con DevOps?**

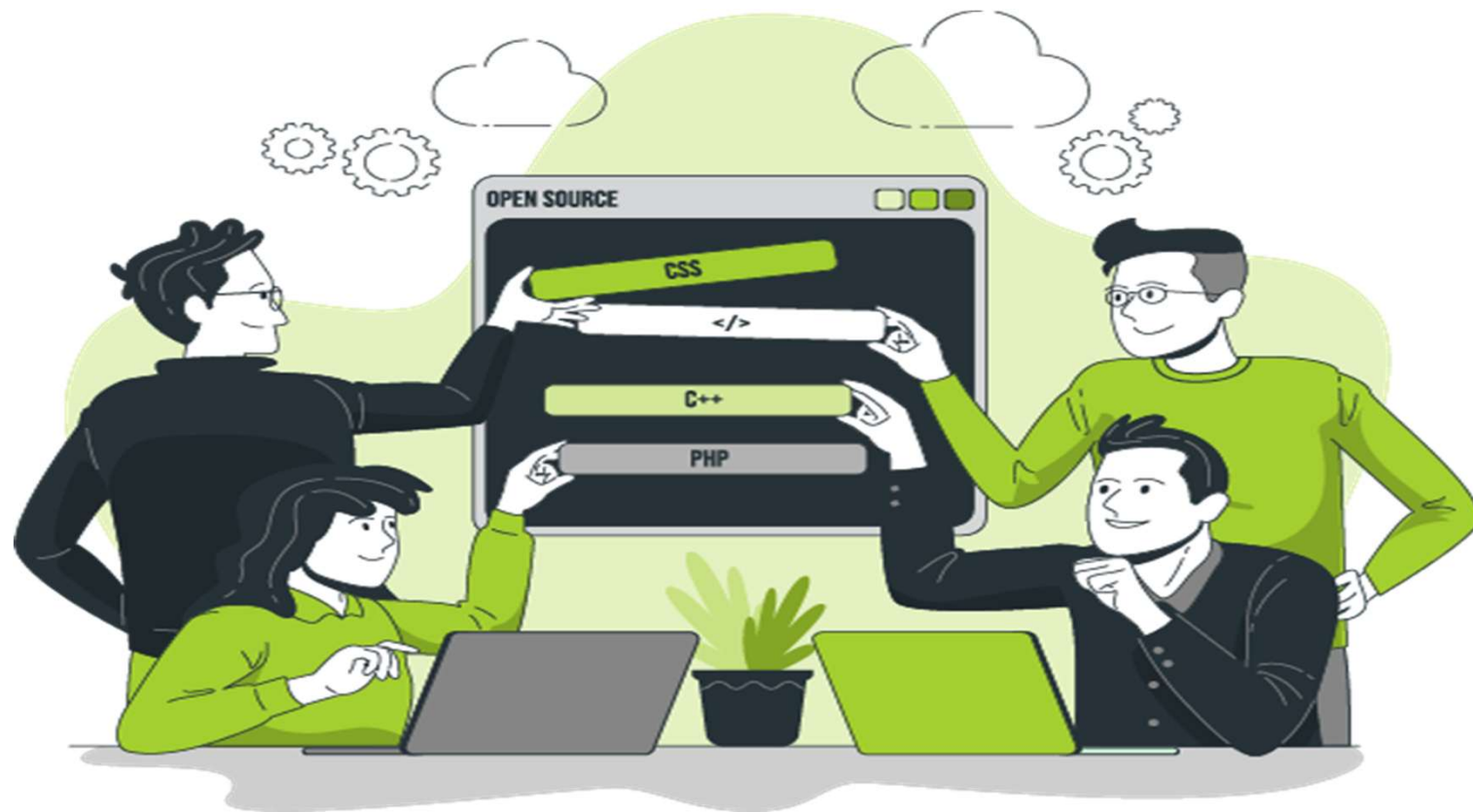
# ¿Qué es DevOps?

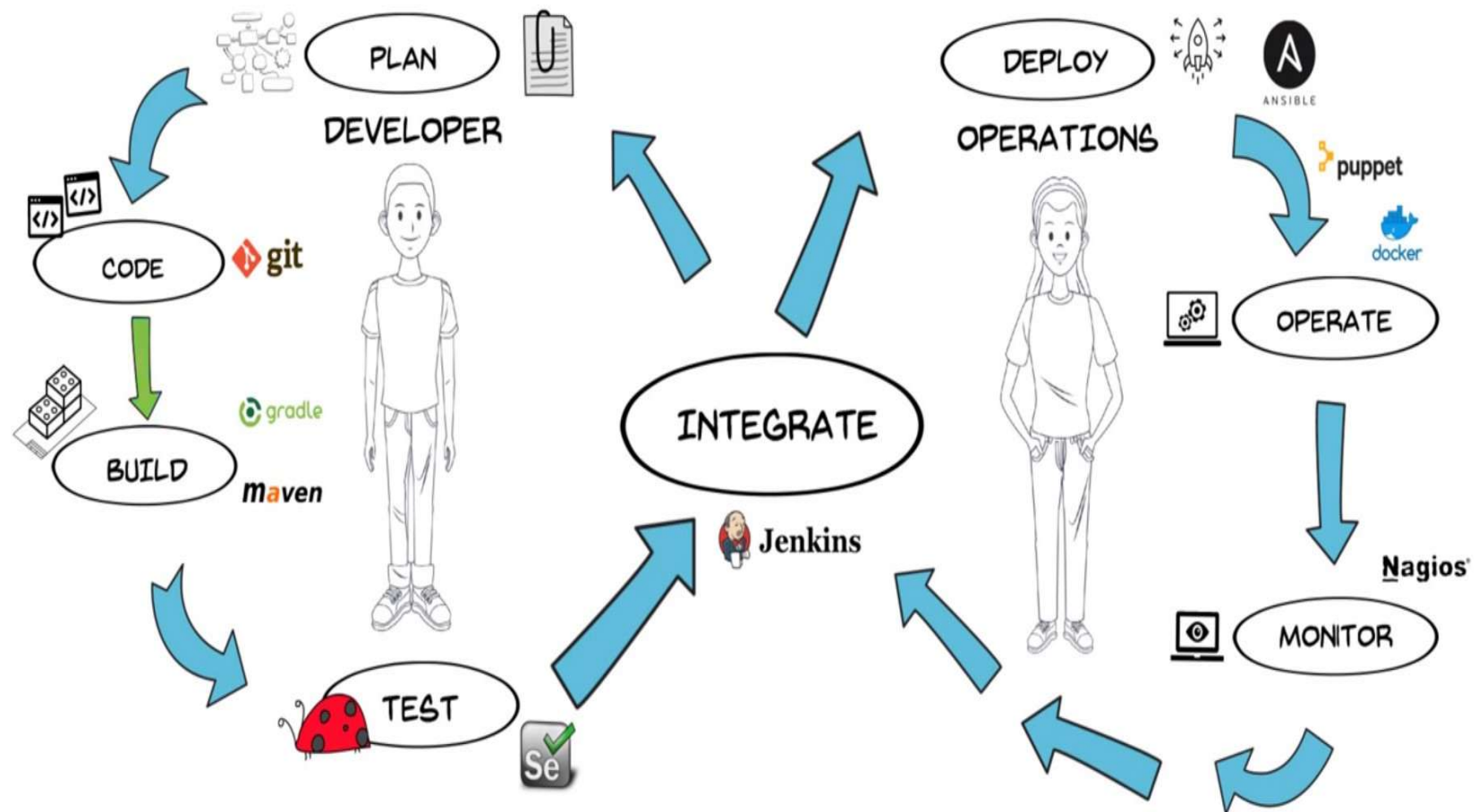
Cultura de mejora continua en el desarrollo y puesta a productivo de una aplicación o servicio



# ¿De dónde surge?

“DevOps surge de la interacción de 2 equipos para buscar un objetivo mas grande que ellos mismos”





- 1 [Conceptos base](#)
- 2 [Evolución de las arquitecturas](#)
- 3 [DevOps](#)
- 4 [Tareas dentro de la cultura DevOps](#)
- 5 [Ciclo de vida de una puesta a productivo](#)

# Tareas frecuentes



Automatización  
de despliegue



Automatización  
de infraestructura

## Tareas frecuentes

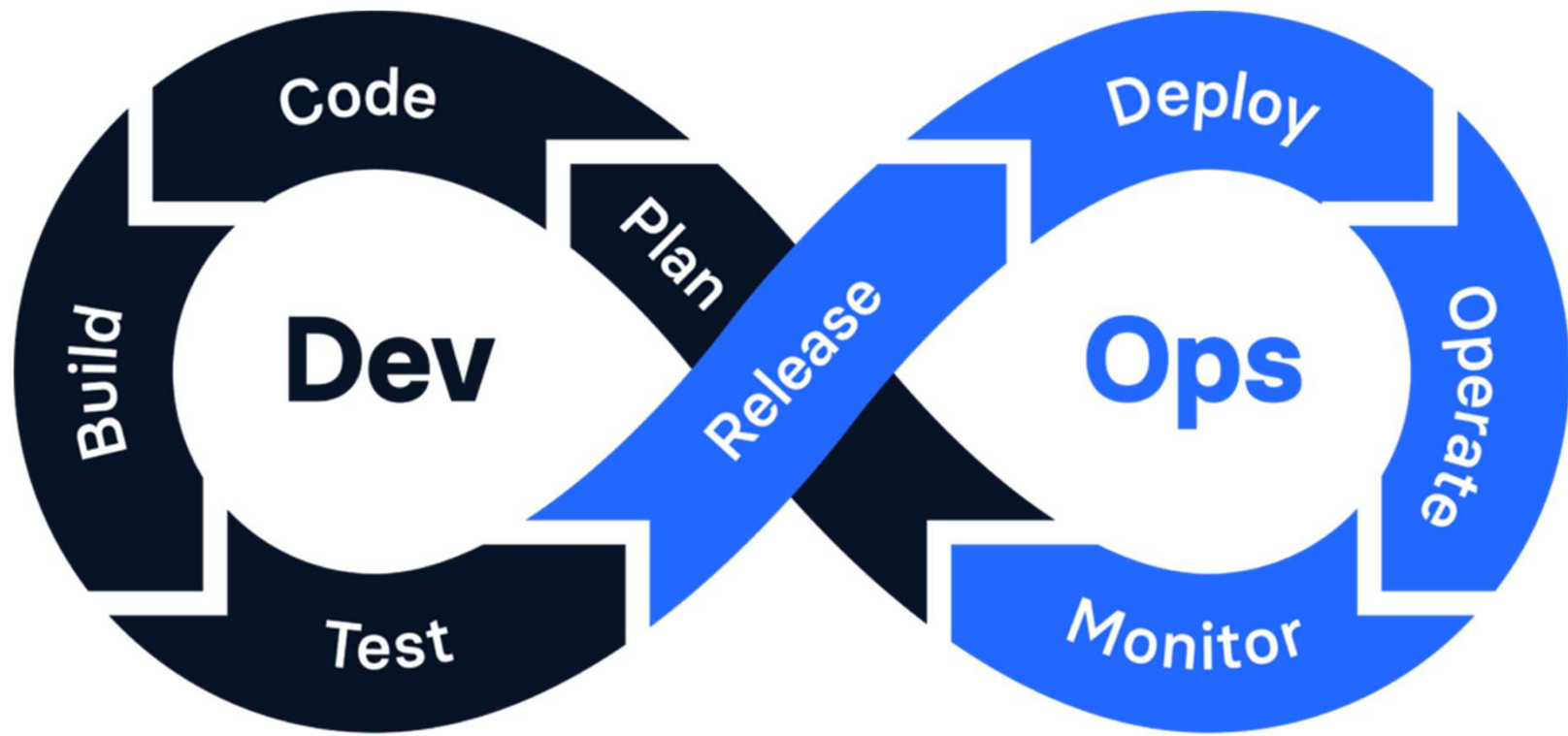


Implementación de  
herramientas de  
análisis y monitoreo



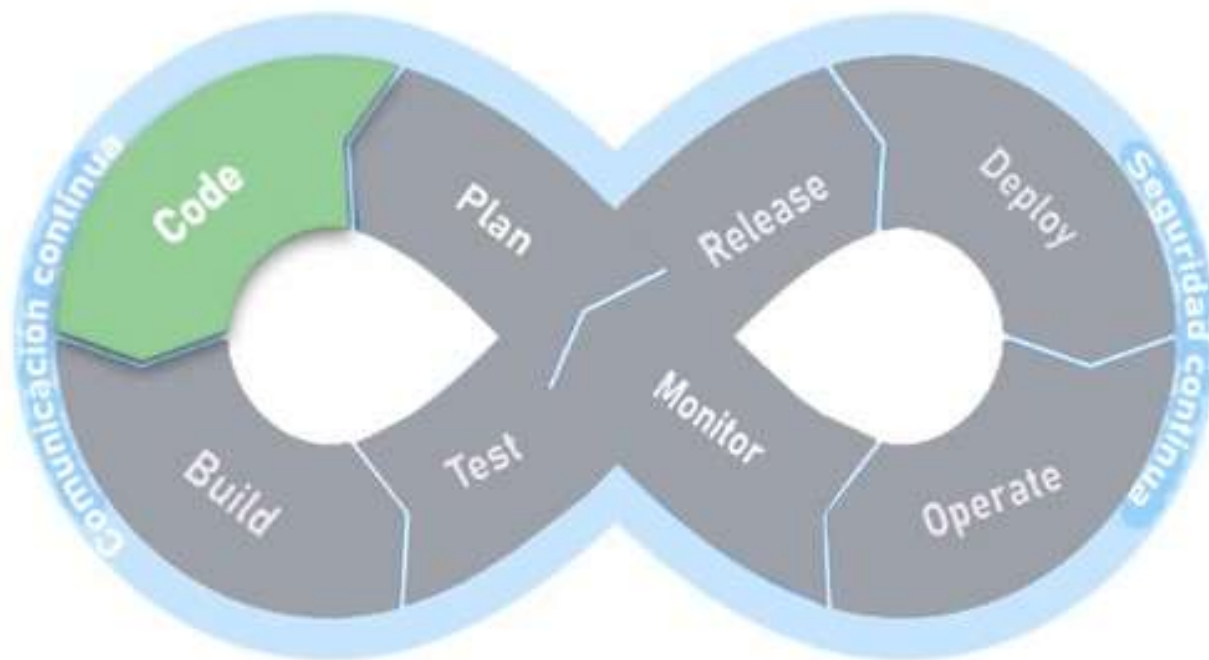
Implementación de  
pruebas y seguridad

- 1 [Conceptos base](#)
- 2 [Evolución de las arquitecturas](#)
- 3 [DevOps](#)
- 4 [Tareas dentro de la cultura DevOps](#)
- 5 [Ciclo de vida de una puesta a productivo](#)



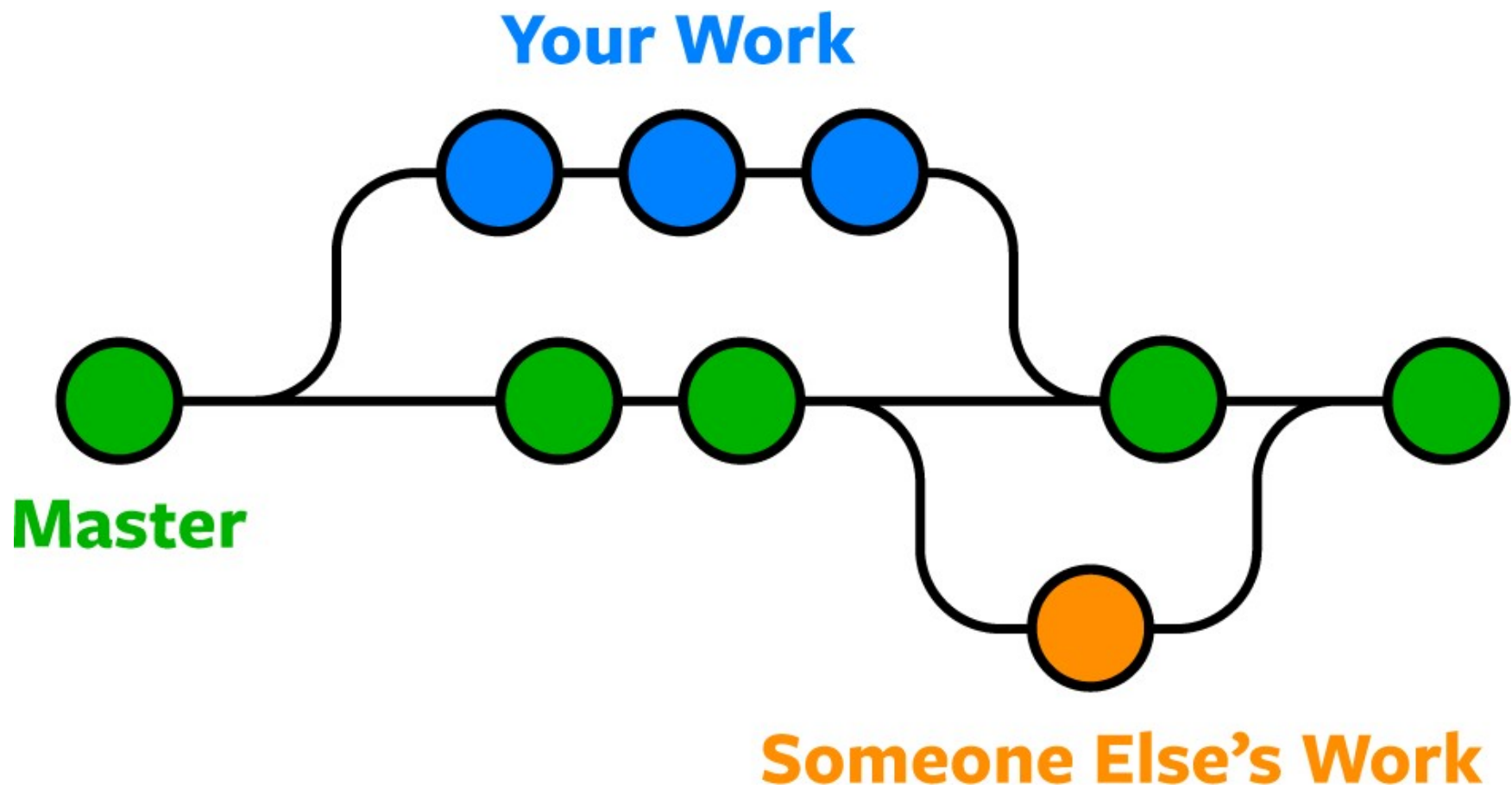


## Fase de Code



# Control de versiones



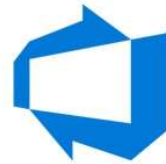


**GitHub**

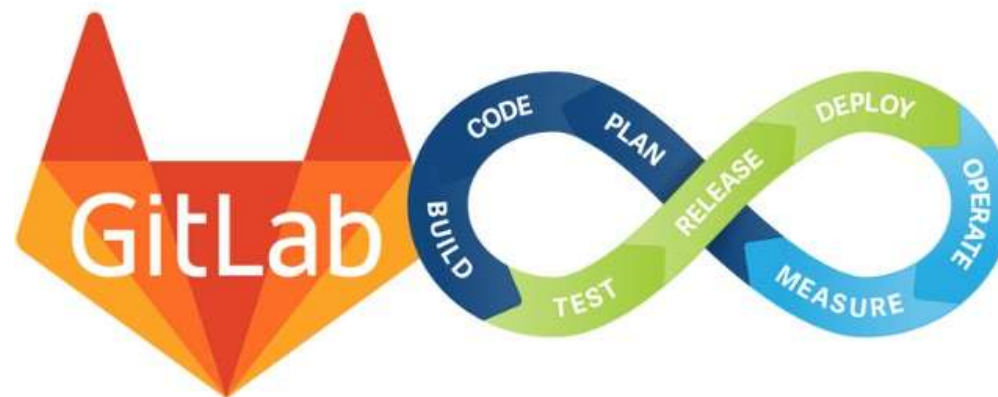
 **Bitbucket**



**GitLab**

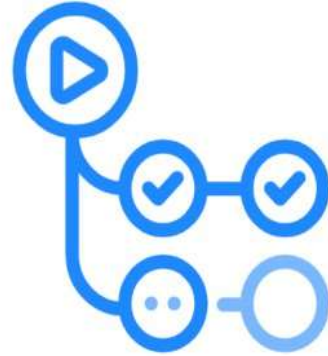


**Azure DevOps**



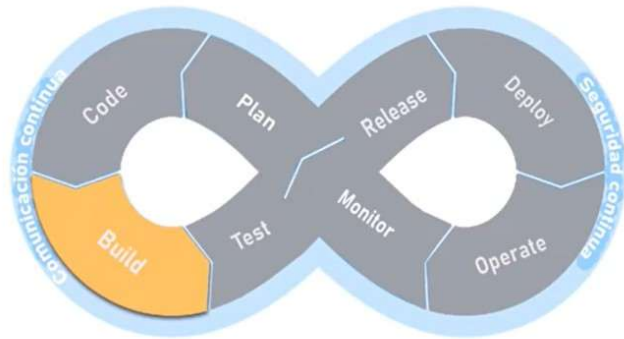
# Azure DevOps





Create CI/CD pipelines  
with Github Actions

## Ciclo de vida



✓ This GitLab CI configuration is valid. [Learn more](#)

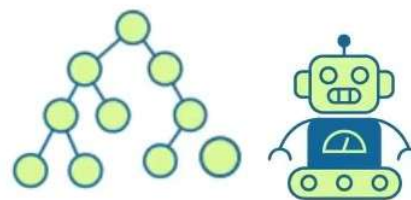
`.gitlab-ci.yml` 268 Bytes

```
1 image: python:3.6-alpine
2
3 pages:
4   script:
5     - pip install sphinx
6     - pip install sphinx_rtd_theme
7     - pip install recommonmark
8     - sphinx-build -d _build/doctrees . _build/html
9     - mv _build/html public
10  artifacts:
11    paths:
12      - public
13  only:
14    - master
```

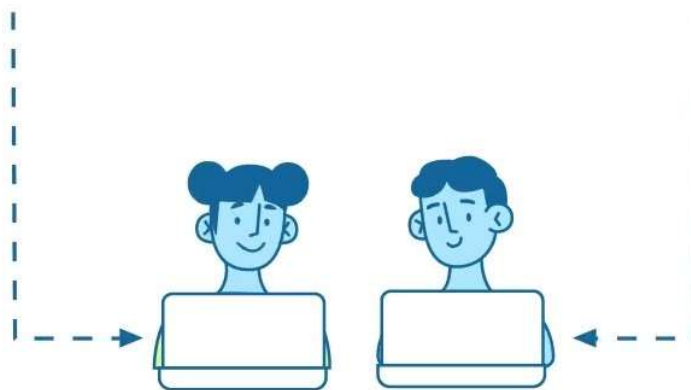




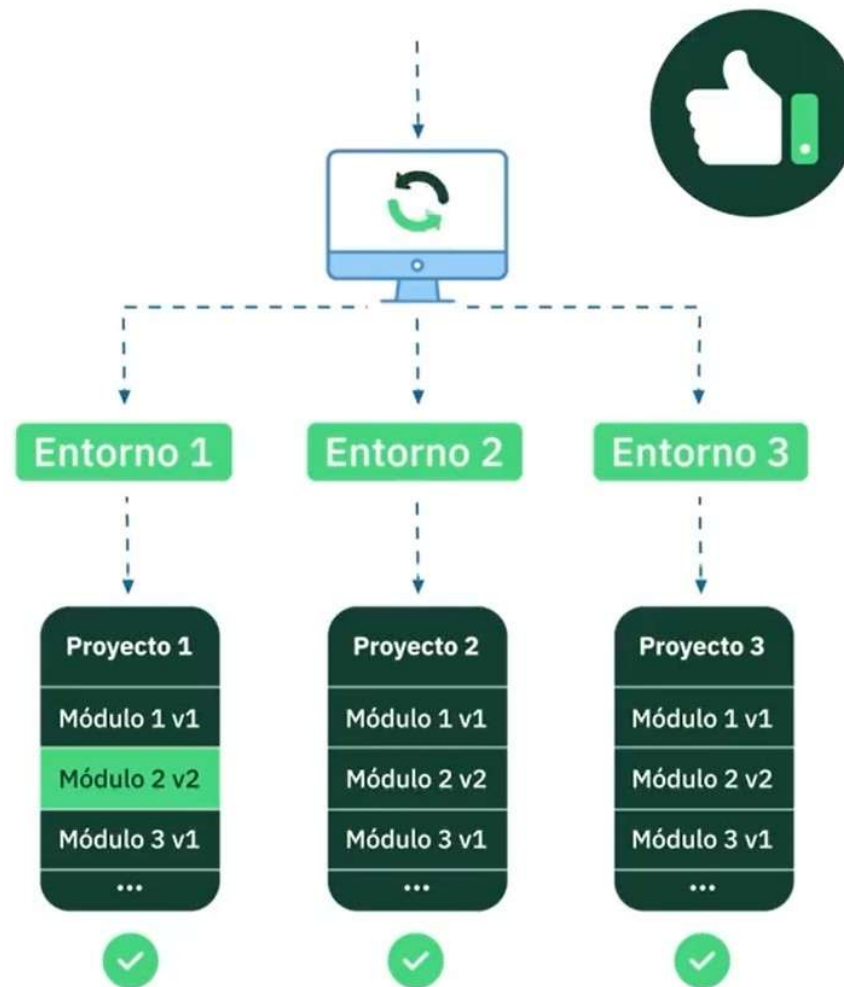
Análisis de datos



Machine learning



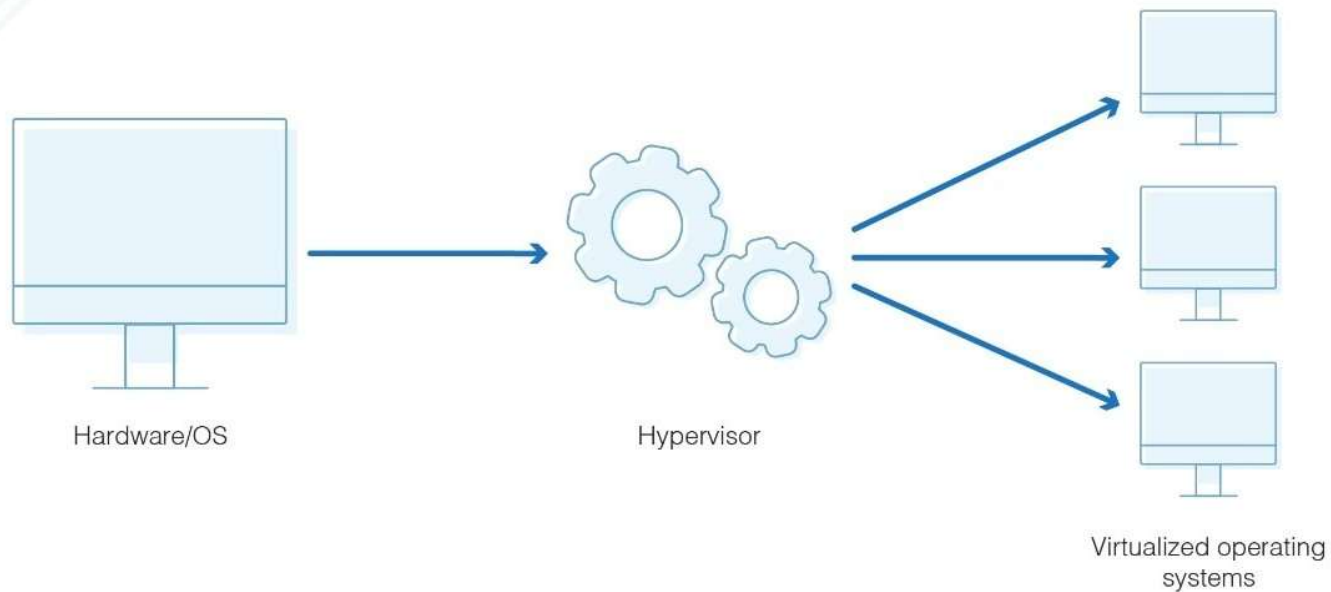




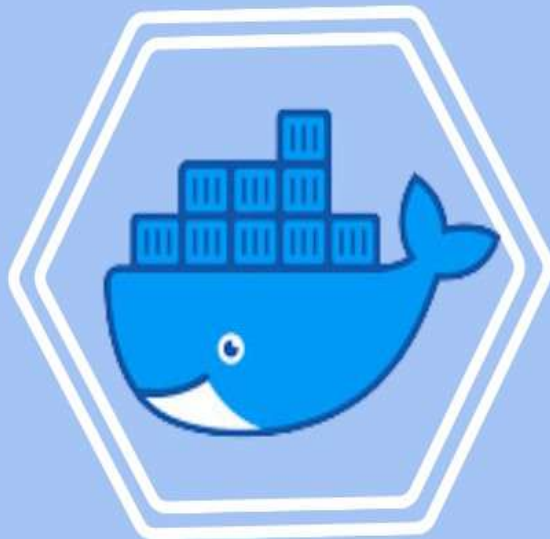
# ¿Qué se busca?

“AISLAMIENTO - REPRODUCIBILIDAD - ORGANIZACIÓN”

# What Is a Hypervisor?

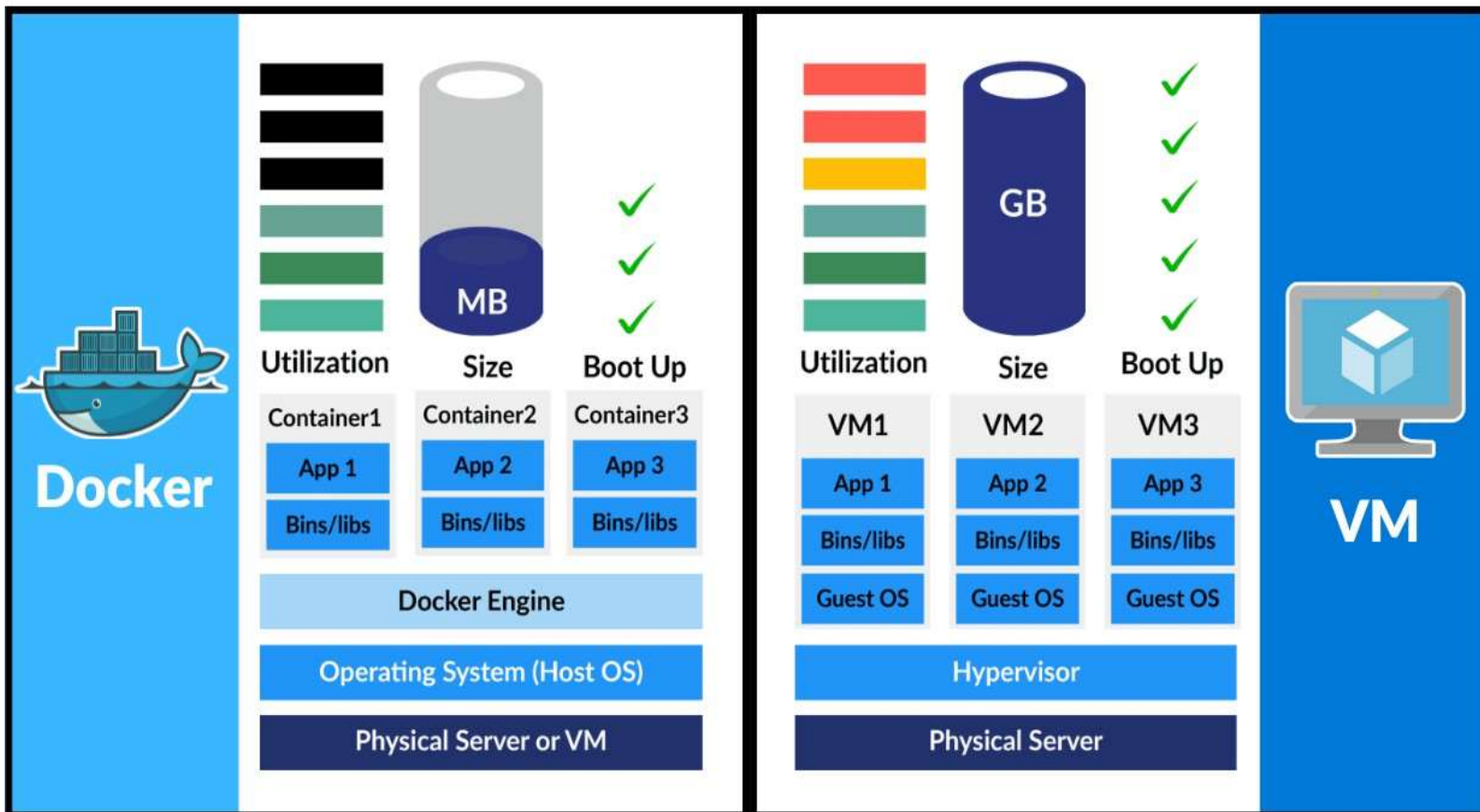


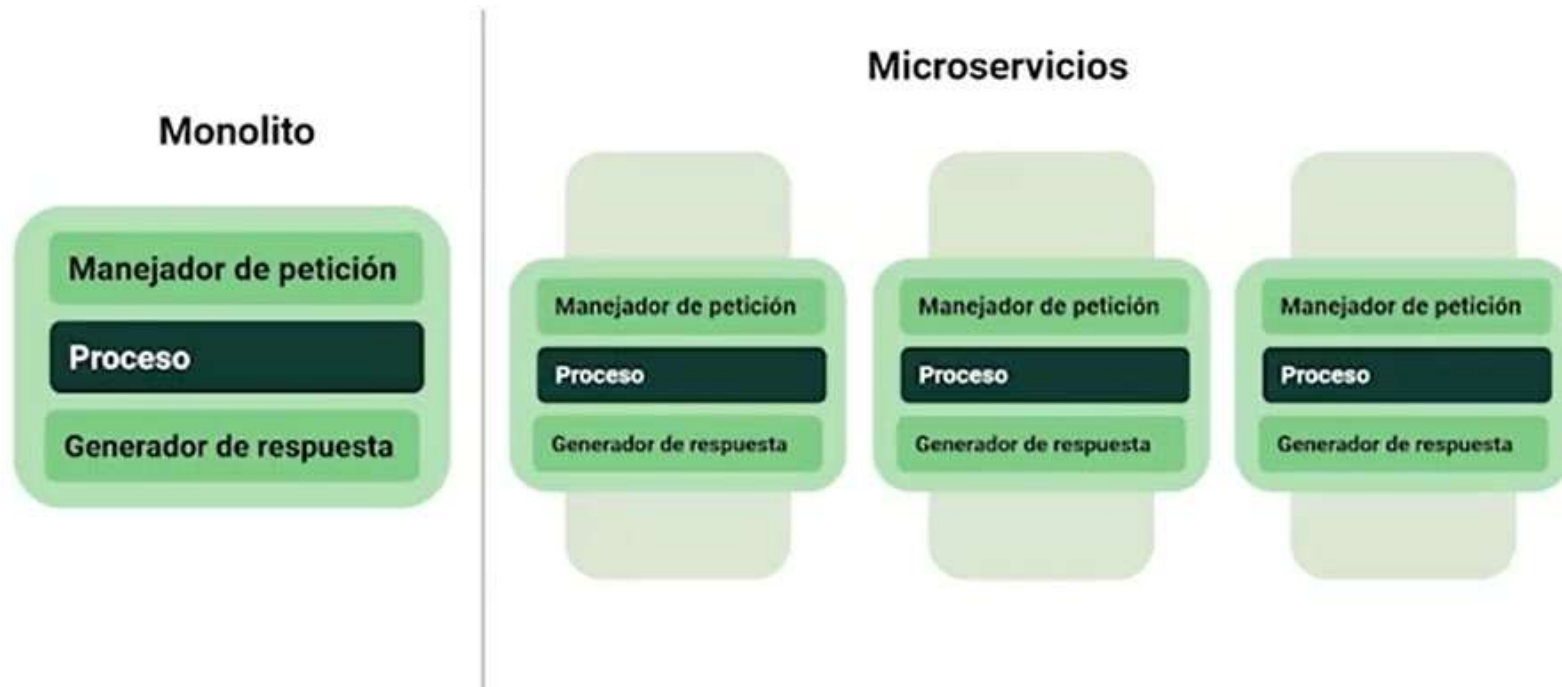
**DOCKER**



**VM**

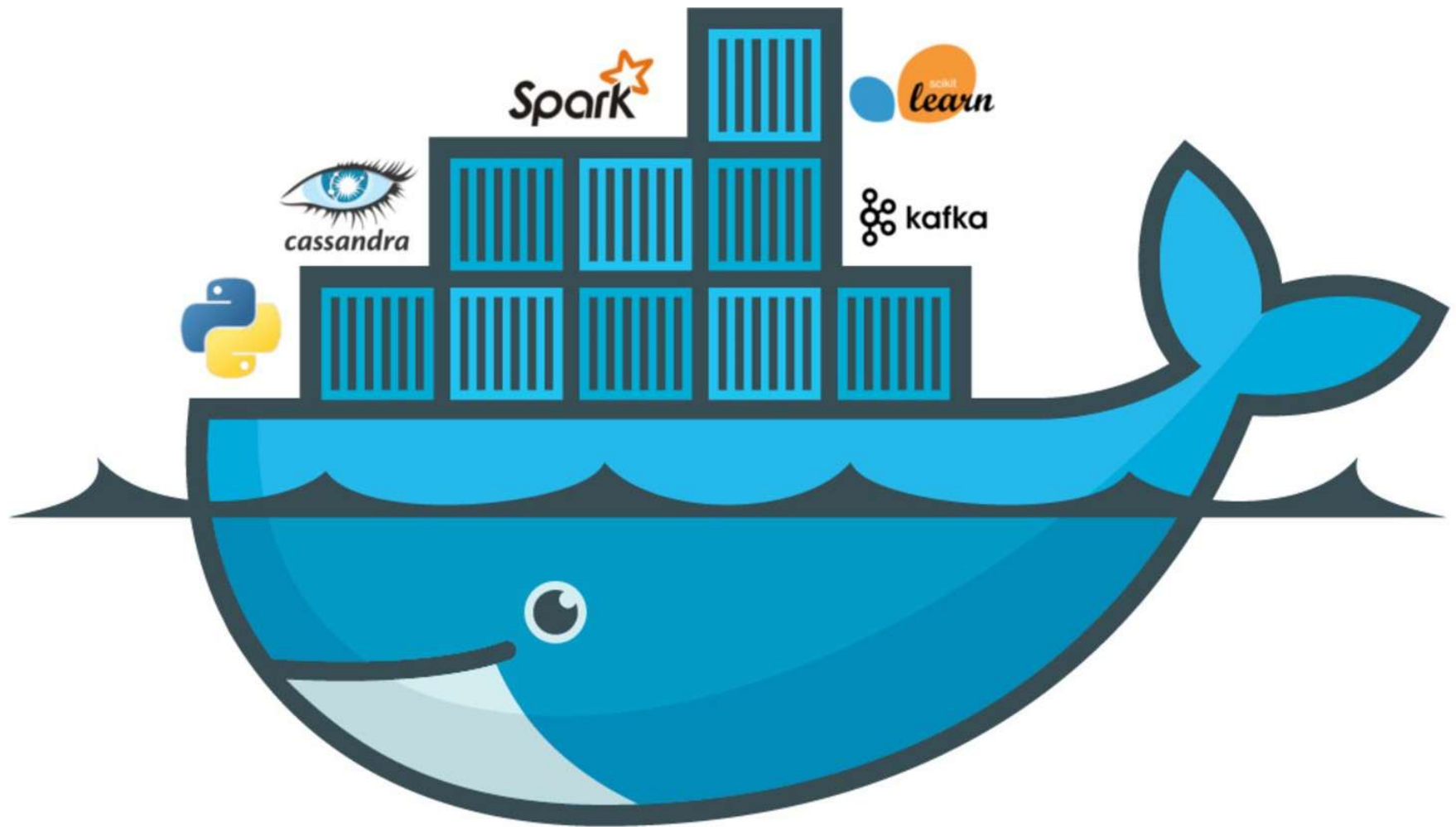








<b>Máquinas Virtuales</b>	<b>Docker</b>
Cada VM ejecuta su propio sistema operativo	Todos los contenedores comparten el mismo kernel del host
El tiempo de arranque es de minutos	Los contenedores se inician en segundos
Los snapshots no son efectivos ni están versionados	Las imágenes pueden versionarse.
No se pueden ejecutar más de un par de VMs en una laptop promedio	Se pueden ejecutar muchos contenedores Docker en una laptop




#20191210.2 Update azure-pipelines.yml for Azure Pipe...

on FabrikamFiber


Run new

Summary

Environments

Triggered by  Steve Danielson

FabrikamFiber

 main 

b2f795e

Today at 12:56 PM

Duration:

1m 9s

Tests:

[Get started](#)

Changes:

2 commits

Work items:

1 linked

Artifacts:

1 published

Stages

Jobs

Build

1 job completed

41s

Deploy

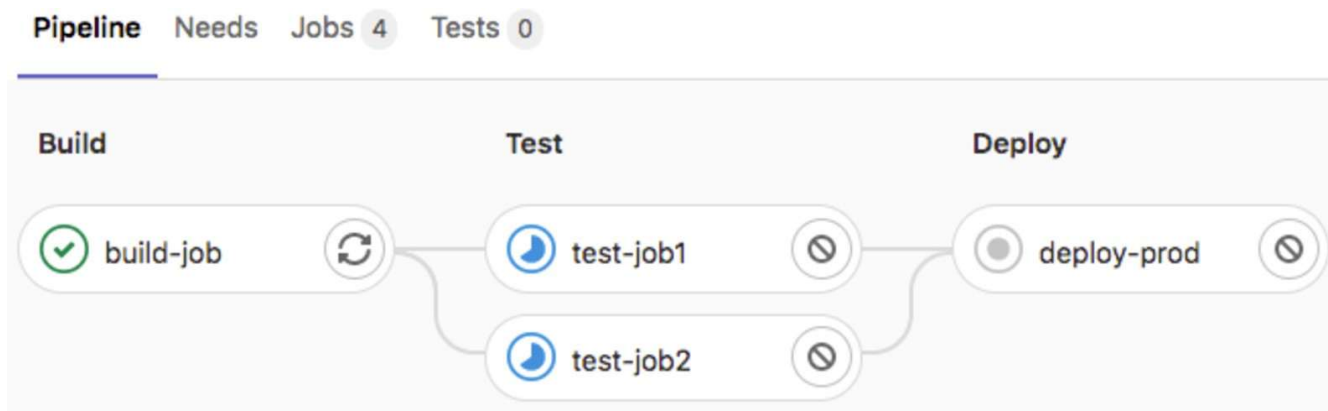
1 job completed

13s


1 artifact




## Supervisión en Azure DevOps

43/52



Supervisión en gitlab

 .gitlab-ci.yml 312 Bytes

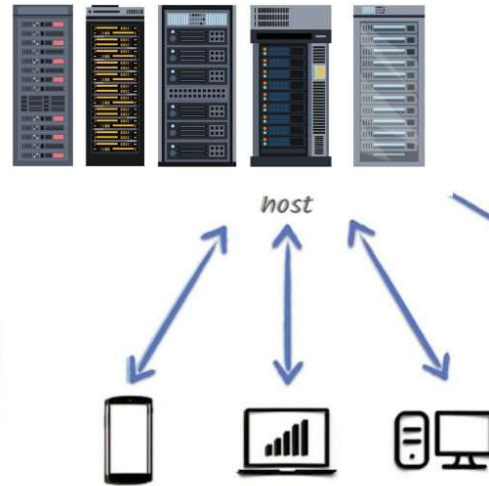
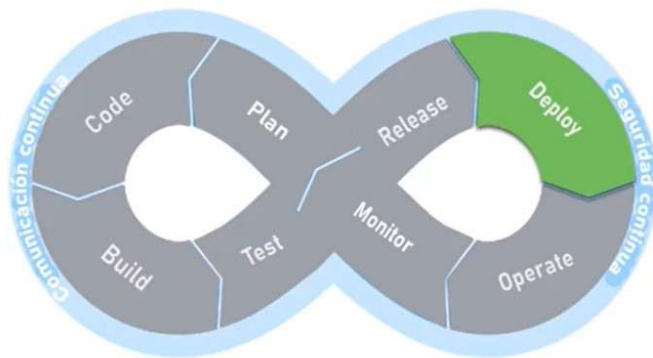


EditWeb IDEReplaceDelete

```
1 image: python:3.6
2
3 stages:
4   - build
5   - test
6
7 build:
8   stage: build
9   script:
10    - echo "Building"
11    - mkdir build
12    - touch build/info.txt
13   artifacts:
14     paths:
15       - build/
16
17 test:
18   stage: test
19   script:
20    - echo "Testing"
21    - test -f "build/info.txt"
```

Definición estructura de pruebas

## Ciclo de vida

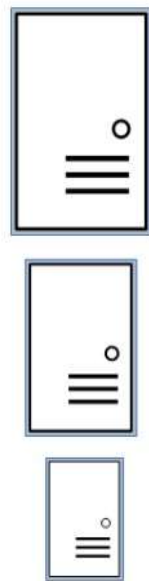


*Deployment*

Etapa de despliegue

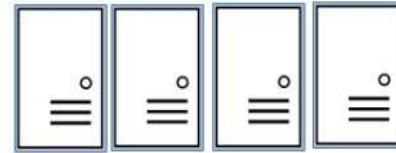
## ESCALAMIENTO VERTICAL

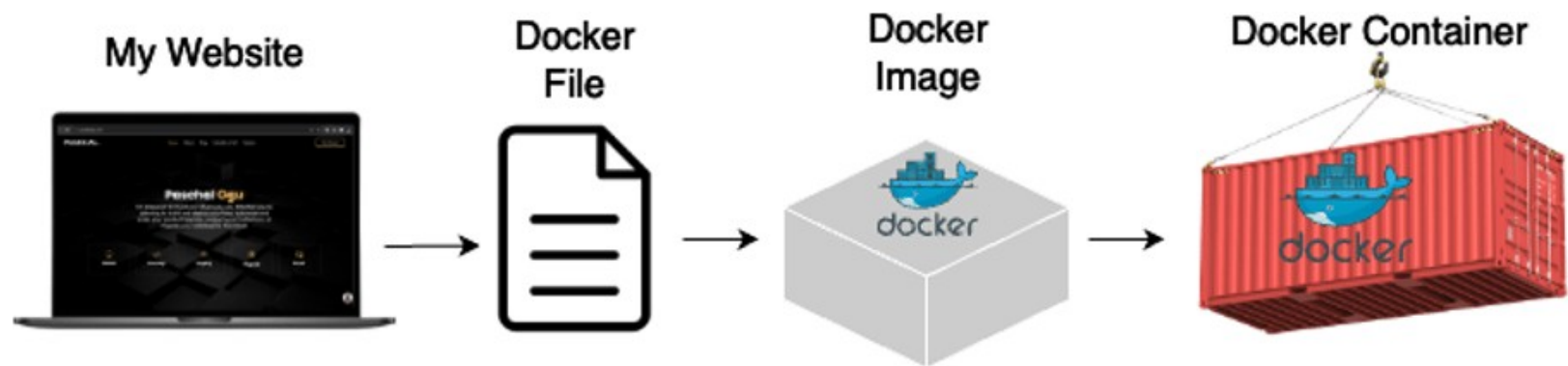
(incrementas tamaño de la instancia CPU, Memoria, etc.)



## ESCALAMIENTO HORIZONTAL

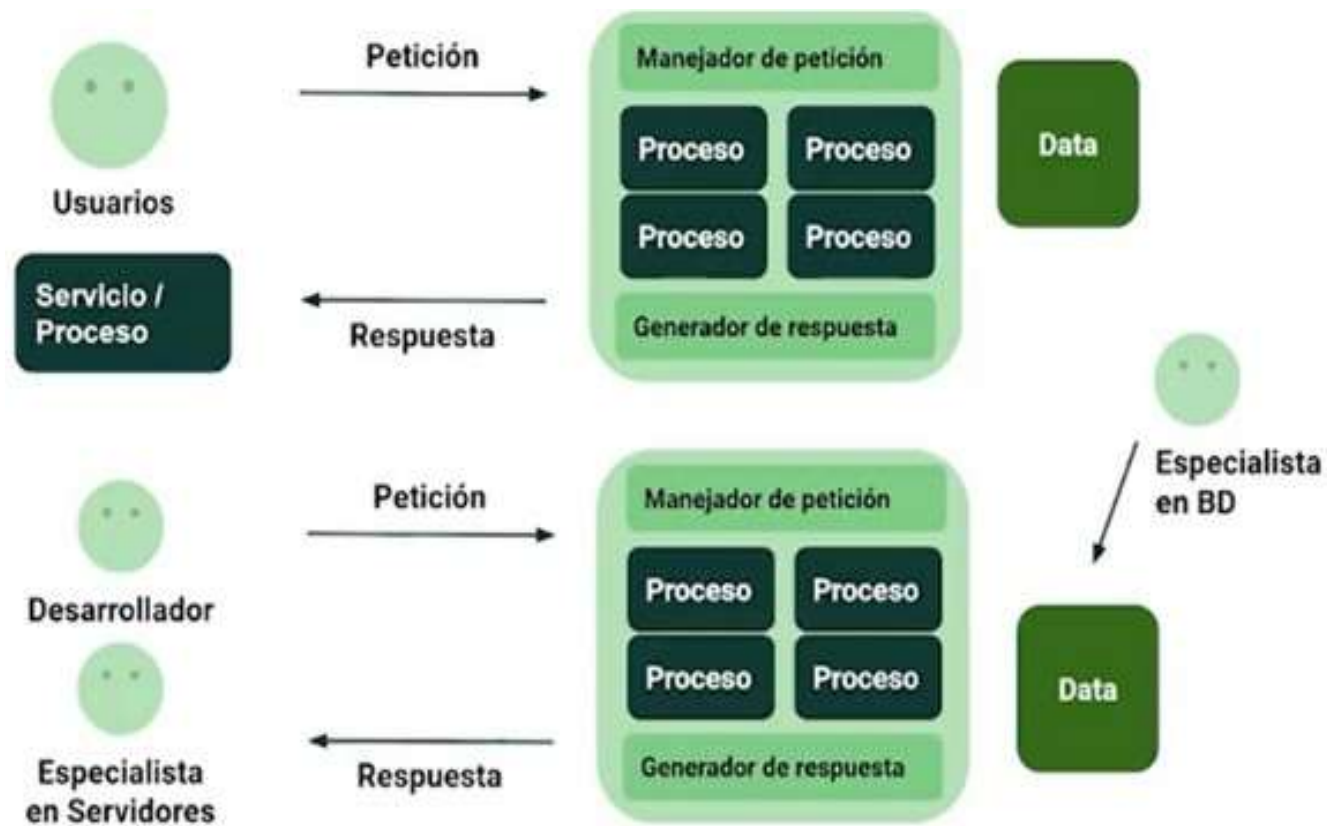
(Agregas más instancias)



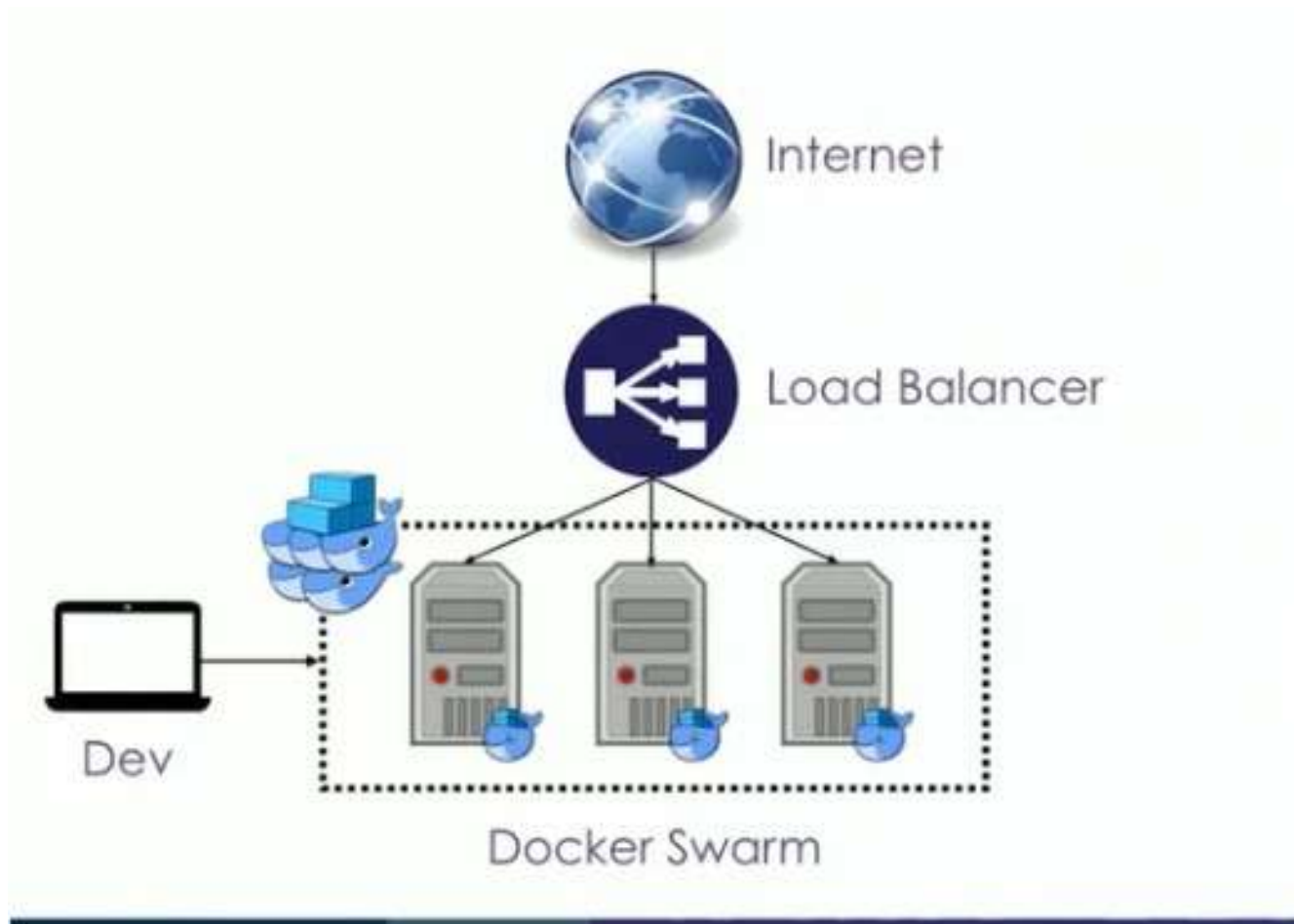


Uso de imágenes de docker

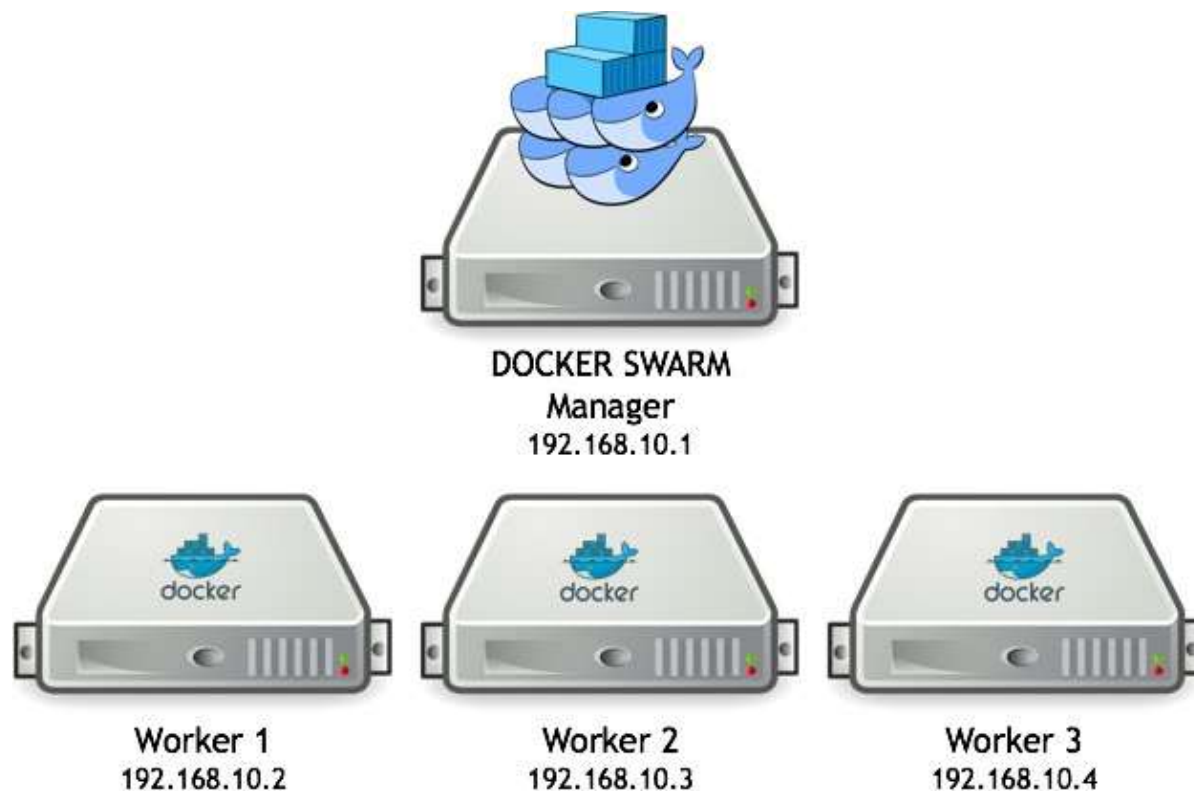




Ambientes



Uso de imágenes de docker



## Ciclo de vida

