



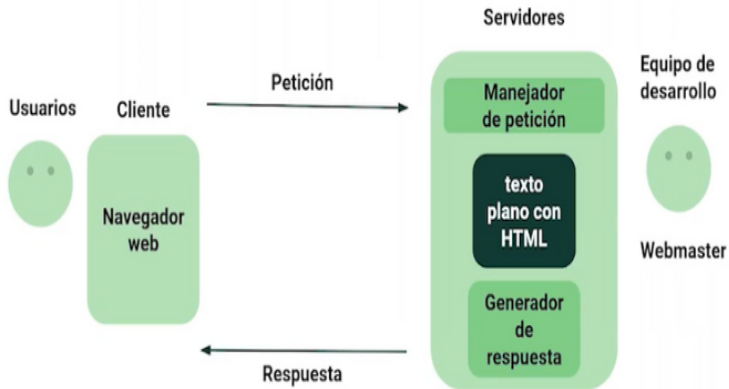
# Flujo de desarrollo devops

Juan Sebastián Méndez Rondón - Santiago Pineda Quintero - Diego Pérez

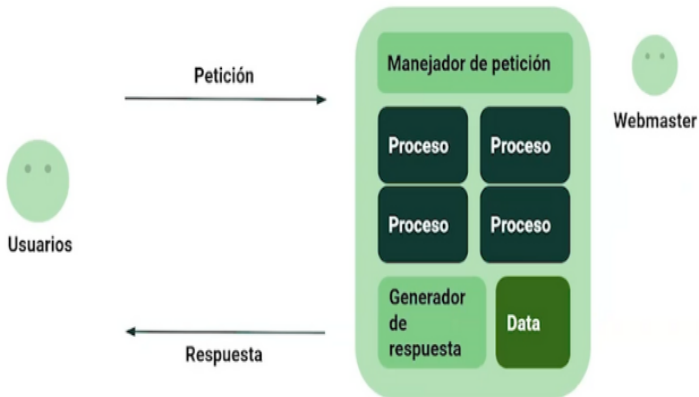
Grupo de Control y Procesamiento Digital de Señales - (GCPDS)  
Universidad Nacional de Colombia  
Manizales, Colombia



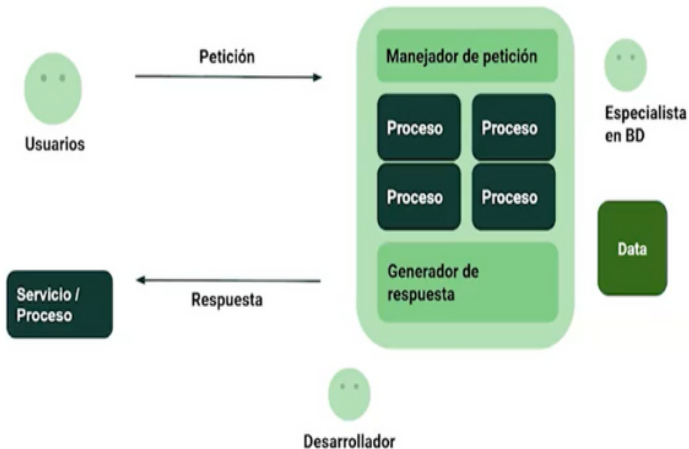
- Evolución de las arquitecturas
- Devops
- Tareas dentro de la cultura devops
- Ciclo de vida de una puesta a productivo



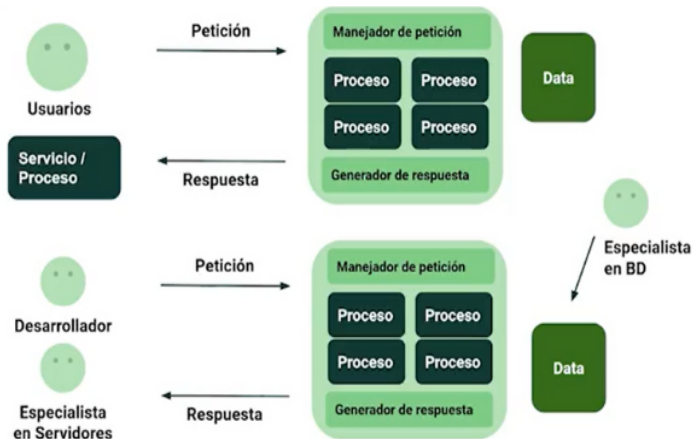
arquitectura base



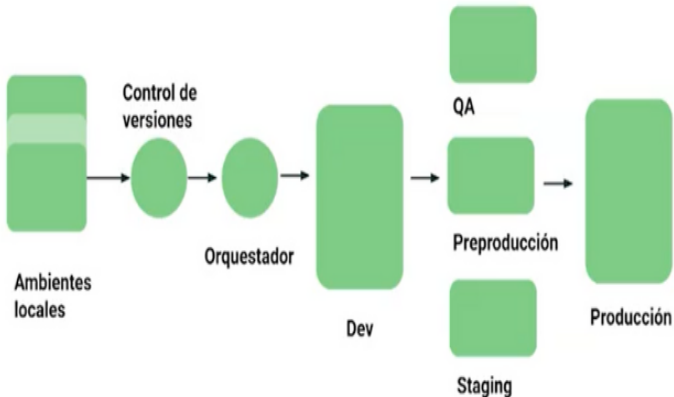
modelado de servicios



modelado base de datos



ambientes



entorno moderno basico

# ¿Qué tiene que ver esto con devops?



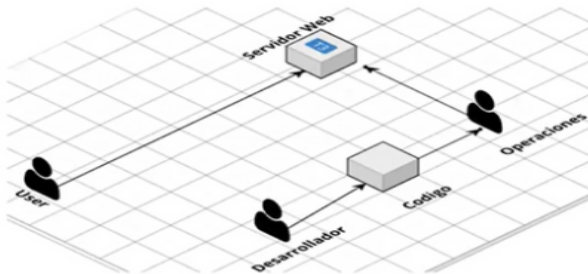
## ¿Qué es devops?

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

## ¿De donde surge?

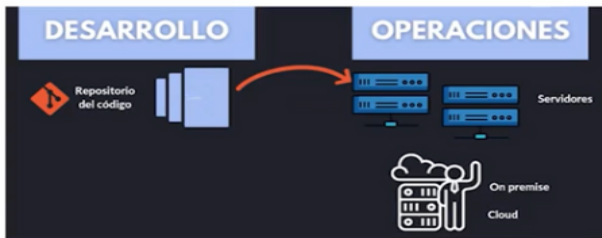
"Devops surge de la interacción de 2 equipos para buscar un objetivo mas grande que ellos mismos"

# Ejemplo



Servicio web con dos equipos

# Ejemplo



DevOps es representado  
por la línea roja

# Tareas frecuentes



Automatización  
de despliegue



Automatización  
de infraestructura

## Tareas frecuentes

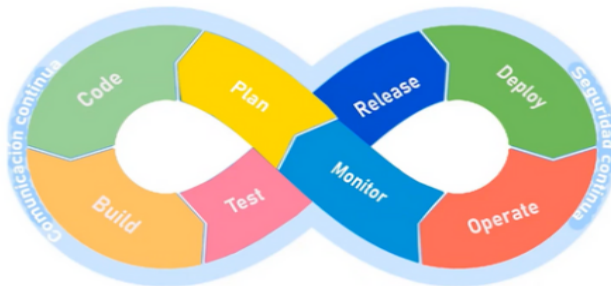


Implementación de  
herramientas de  
análisis y monitoreo

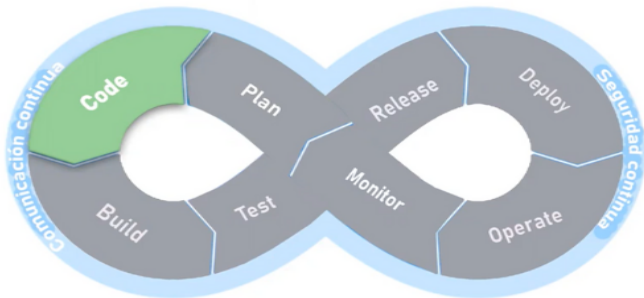


Implementación de  
pruebas y seguridad

# Ciclo de vida



## Fase de Code



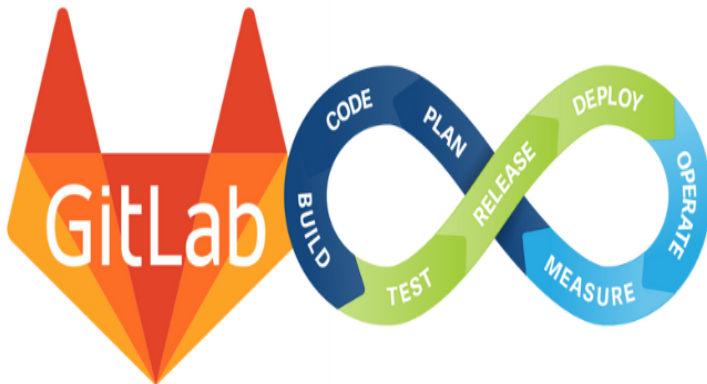


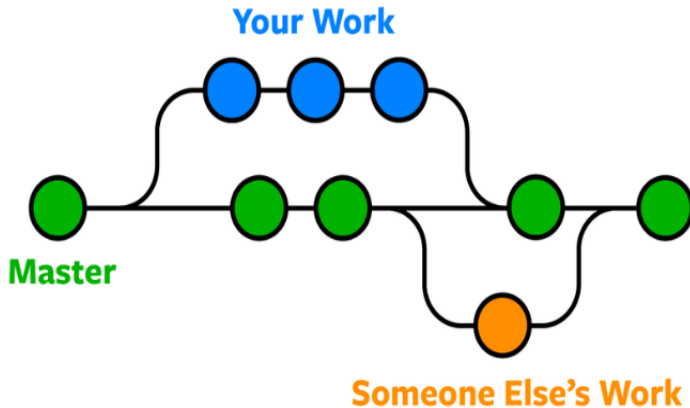
# Control de versiones



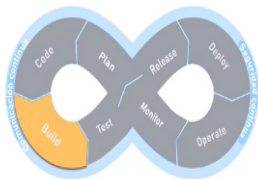
Git









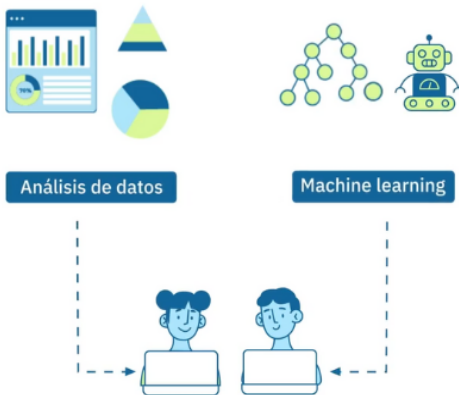
## Ciclo de vida



✓ 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
```











# Aislamiento

# Reproducibilidad

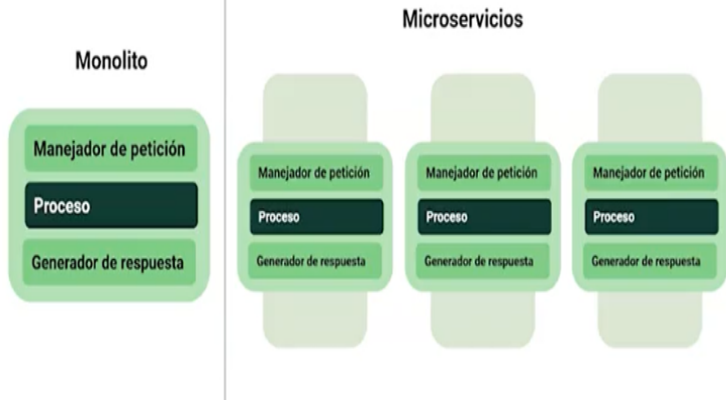
# Organización

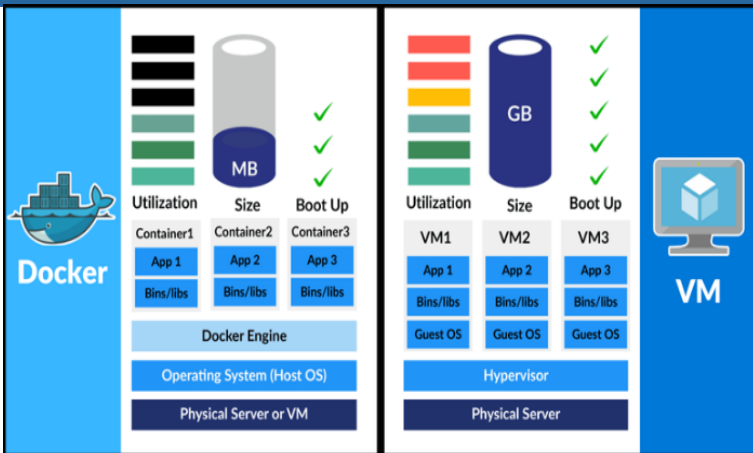
## DOCKER



## VM

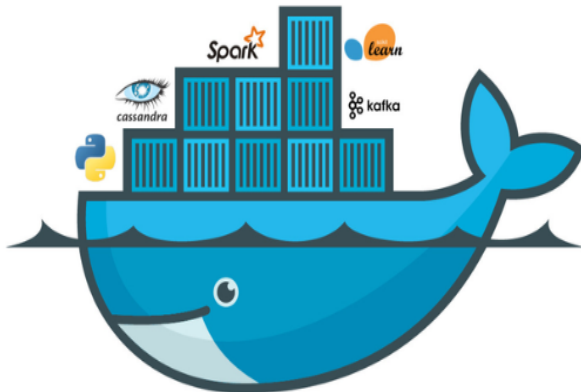








Virtual Machines	Docker
Each VM runs its own OS	All containers share the same Kernel of the host
Boot up time is in minutes	Containers instantiate in seconds
VMs snapshots are used sparingly	Images are built incrementally on top of another like layers. Lots of images/snapshots
Not effective diffs. Not version controlled	Images can be diffed and can be version controlled. Dockerhub is like GITHUB
Cannot run more than couple of VMs on an average laptop	Can run many Docker containers in a laptop.
Only one VM can be started from one set of VMX and VMDK files	Multiple Docker containers can be started from one Docker image





```
.gitlab-ci.yml 312 Bytes
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"
```

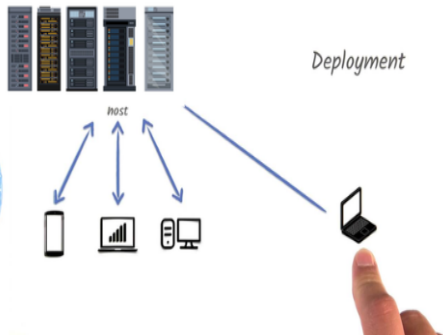
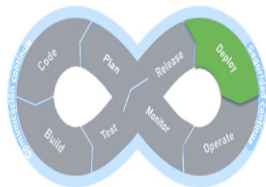
## Definición estructura de pruebas



## Supervisión en gitlab



## Ciclo de vida

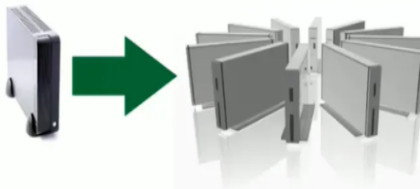


Etapa de despliegue

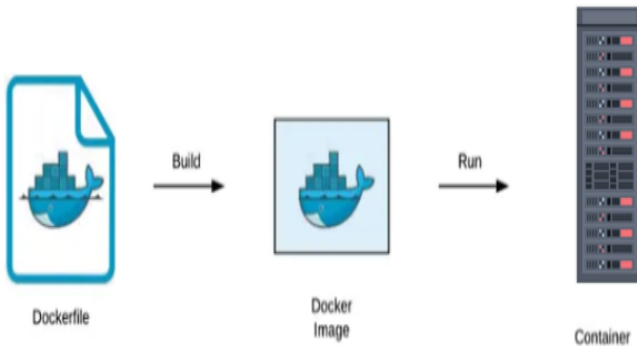
Escalar verticalmente



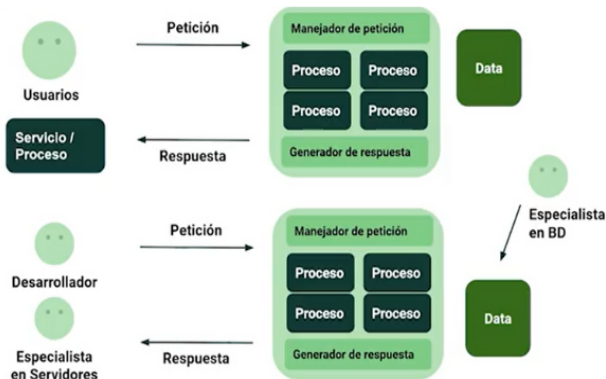
Escalar horizontalmente



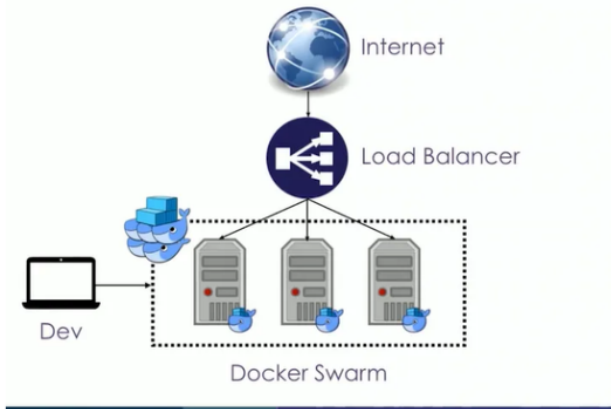
Etapas de despliegue



Uso de imagenes de docker

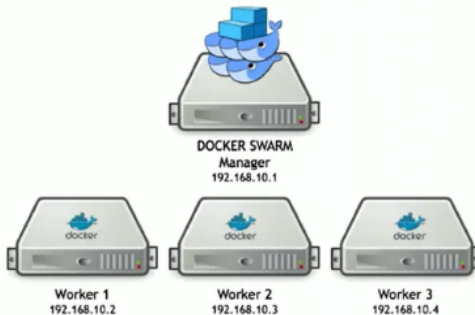


Uso de imagenes de docker



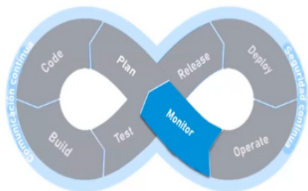
Uso de imagenes de docker

## Arquitectura



Uso de imagenes de docker

## Ciclo de vida



Uso de imagenes de docker