

The background of the slide is a close-up, slightly blurred image of a green printed circuit board (PCB). It features intricate black circuit traces, various electronic components like capacitors and resistors, and a series of pins along the edges. The lighting is soft, creating a professional and technical aesthetic.

# Sistemas Distribuidos

---

TAREA #1

FRANCISCO ABARCA

ALEXANDER RUZ

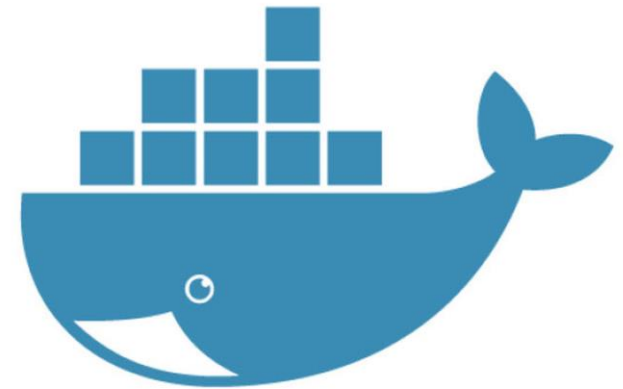
# Temario

- Arquitectura
- Implementación
  - Diseño conceptual
  - Comportamiento de sockets
    - Envío de datos
    - Heartbeat
  - Dockerfiles
  - Docker-compose

# Arquitectura

---

LAS TECNOLOGÍAS BASES SON PYTHON  
(LÓGICA) Y DOCKER (DESPLIEGUE)



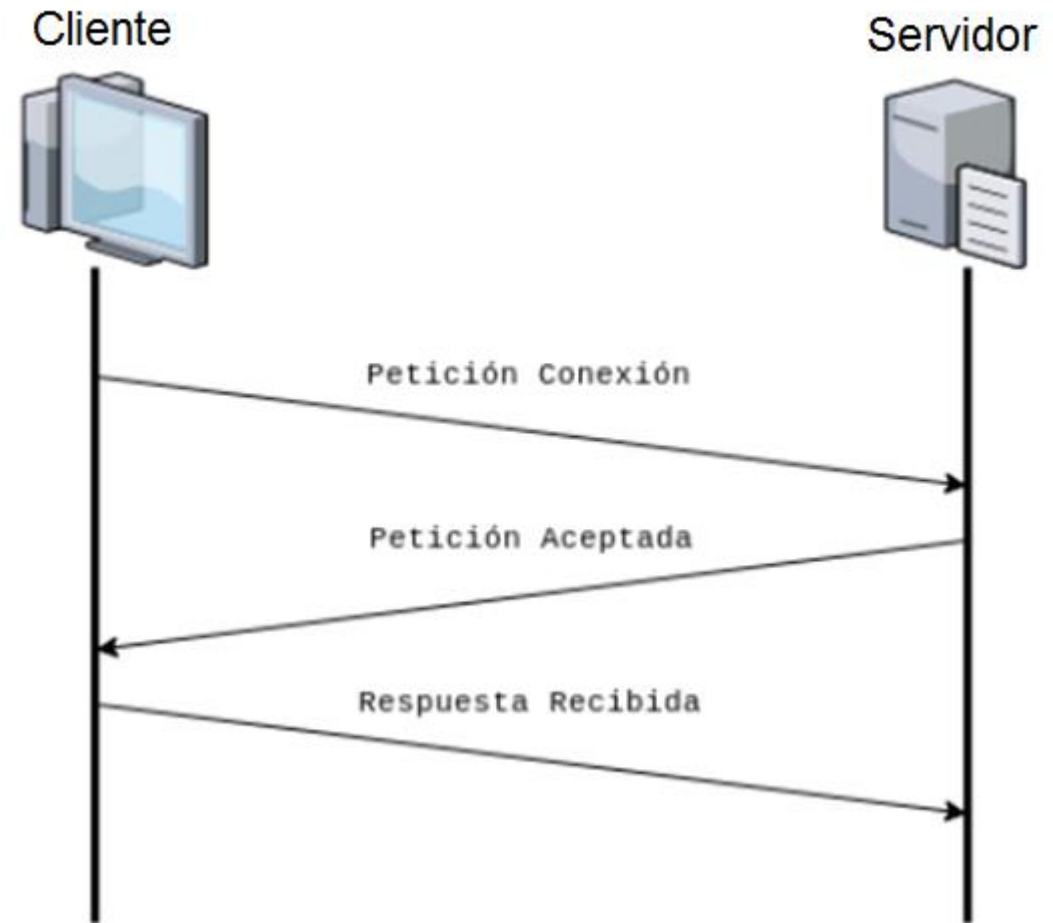


# Cliente-Servidor

Se realiza una conexión en el puerto 5000.

Hay comunicación entre ambos.

Ambos tienen un archivo con información.



# Dockerfiles

El ejecutable de cliente y servidor se empaquetan en un contenedor diferente para cada uno.

## Cliente

```
1 FROM python:latest
2
3 ADD cliente.py /cliente/
4
5 WORKDIR /cliente/
```

## Servidor

```
1 FROM python:latest
2
3 ADD servidor.py /servidor/
4
5 WORKDIR /servidor/
```

# Docker-compose

---

Docker-compose busca instrucciones en este archivo, que luego ejecuta según lo que se busca.

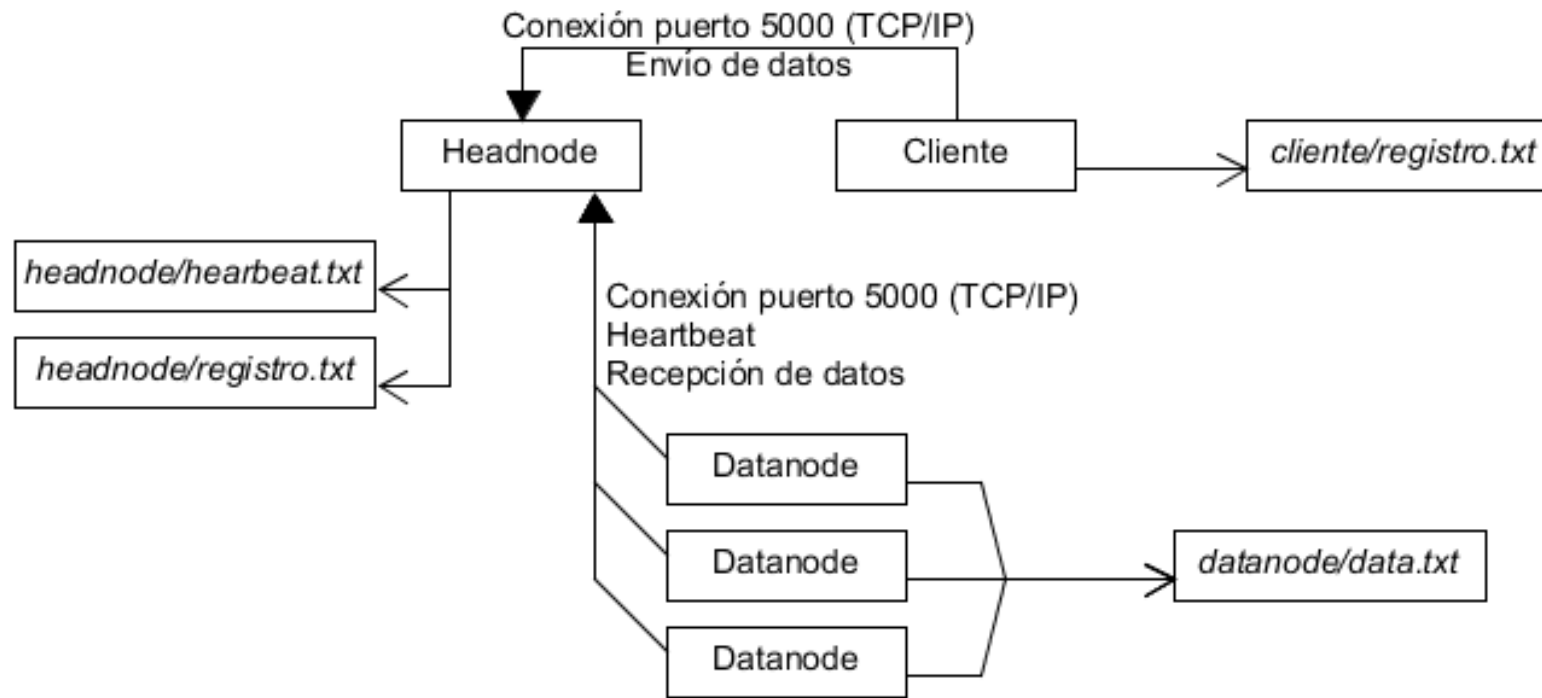
```
1  version: "3"
2
3  services:
4    server:
5      build: servidor/
6      command: python3 servidor.py
7      ports:
8        - "5000:5000"
9      volumes:
10       - ./servidor:/servidor/
11
12   client:
13     build: cliente/
14     command: python3 cliente.py
15     volumes:
16       - ./cliente:/cliente/
17     depends_on:
18       - server
```

The background is a dark gray field filled with various blue and white line-art icons. These include a globe with speech bubbles, a large elephant (the Hadoop logo) inside a circular flow diagram, a city skyline, a bar chart, a line graph, a pie chart, a speech bubble with a network diagram, a play button, a photo icon, and several binary code strings (1s and 0s).

# Actividad Hadoop

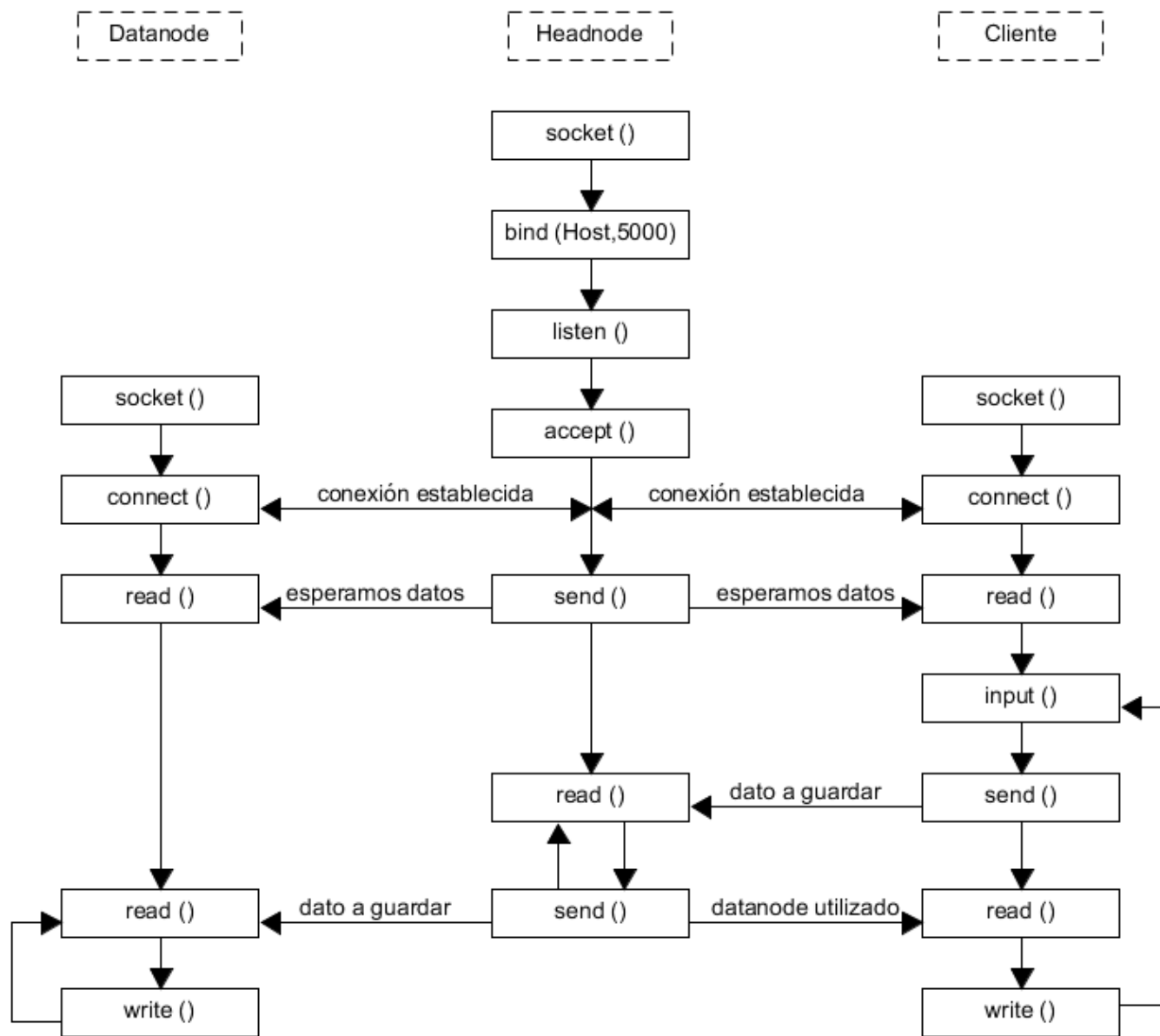
---



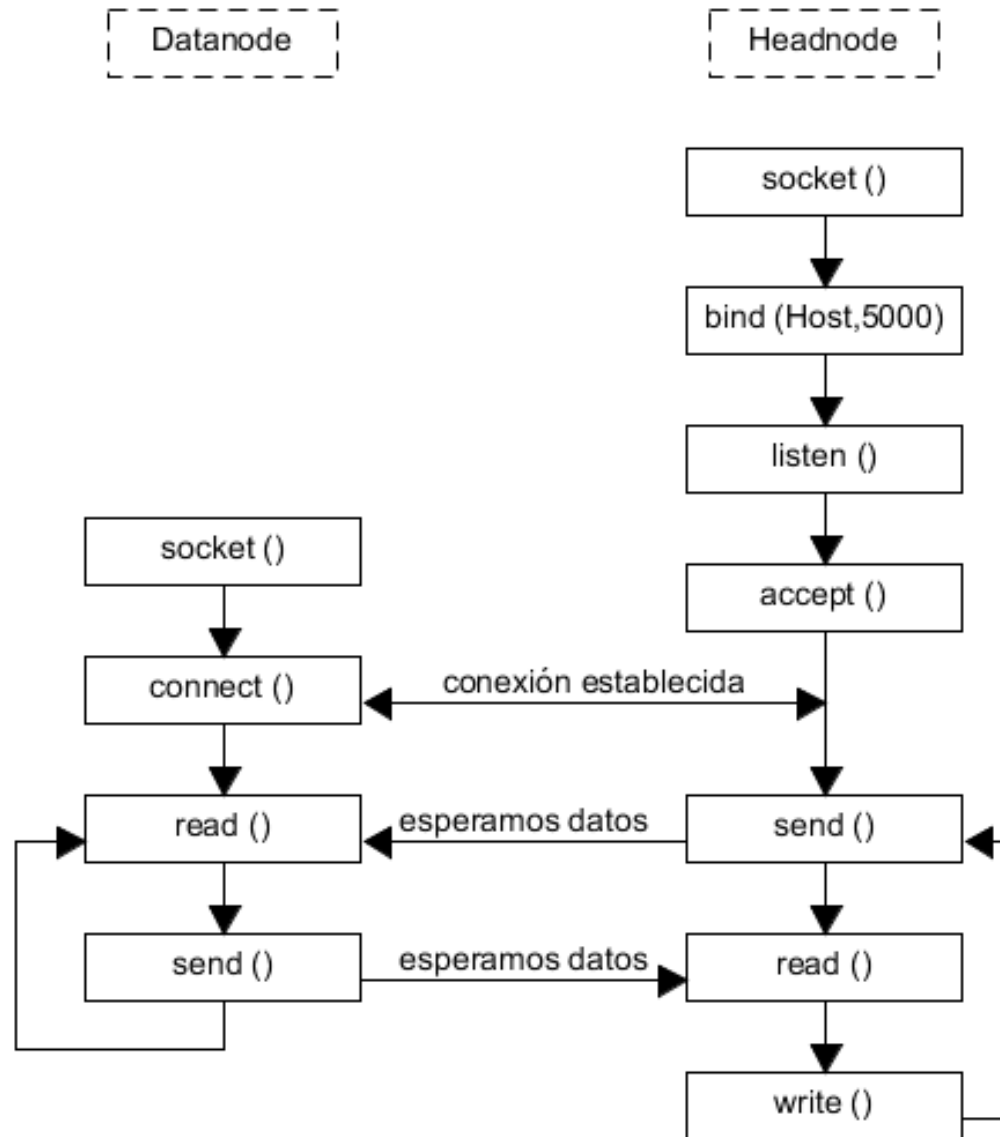


# Diseño conceptual

---



# Comportamiento de sockets (Envío de datos)



# Comportamiento de sockets (Heartbeat)

---

# Dockerfiles

---



## Headnode

```
FROM python:latest  
ADD Headnode.py /headnode/  
WORKDIR /headnode/  
EXPOSE 5000
```



## Datanode

```
FROM python:latest  
ADD Datanode.py /datanode/  
WORKDIR /datanode/
```



## Cliente

```
FROM python:latest  
ADD Cliente.py /cliente/  
WORKDIR /cliente/
```

# Docker-compose

---

cliente:

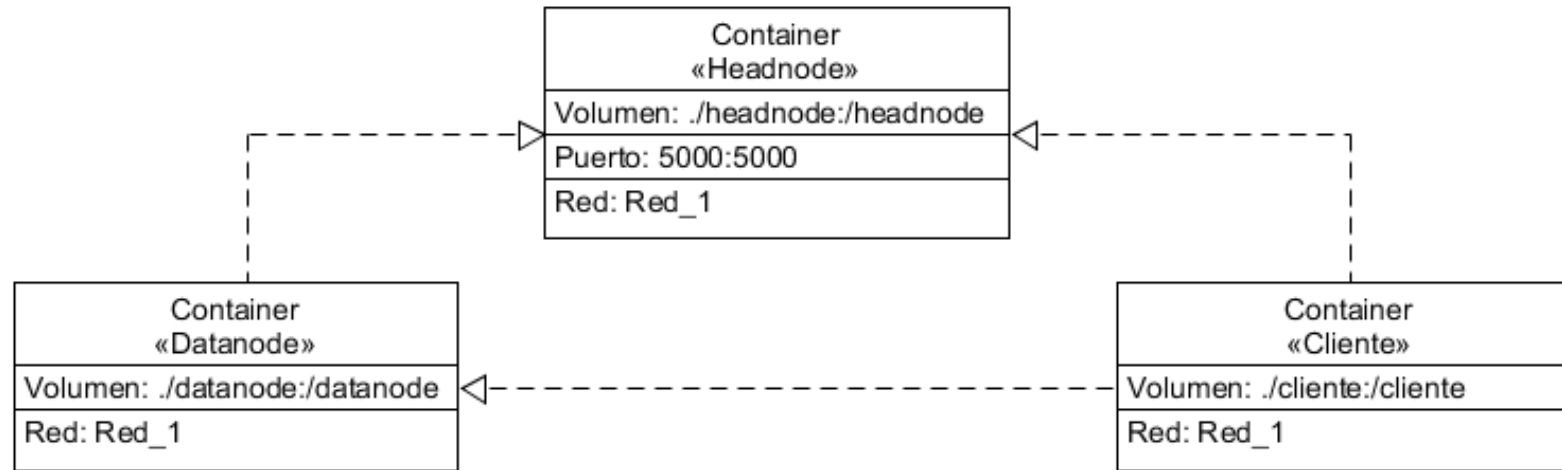
```
container_name: Cliente
stdin_open: true
tty: true
build: cliente/
command: python ./Dummy.py
network_mode: host
volumes:
  - ./cliente:/cliente
depends_on:
  - headnode
  - datanode
```

datanode:

```
build: datanode/
command: python ./Datanode.py
network_mode: host
volumes:
  - ./datanode:/datanode
depends_on:
  - headnode
```

headnode:

```
container_name: Headnode
stdin_open: true
tty: true
build: headnode/
command: python ./Headnode.py
ports:
  - "5000:5000"
network_mode: host
volumes:
  - ./headnode:/headnode
```



# Modelo de containers

---