



# *Segunda Fase P.I*

Fabricio Agüero Mata - C20097

Jose Guerra Rodriguez - C33510

Sebastián Hernández Porras - C23770



# Protocolo mancomunado

**Tenedor se conecta al Servidor (Nuevo tenedor):**

Tenedor hace broadcast del siguiente mensaje:

**Request:** GET /servers

Los servidores contestan directamente al tenedor que lo solicitó:

**Response:** ServerName | ip | {lista dibujos}

**Servidor se conecta al Tenedor:**

Los servidores envían a los tenedores en broadcast:

**Response:** ServerName | ip | {lista dibujos}

Ejemplo:

ServidorA | 172.16.123.51 | ballena, perro, gato

- Al ser un mensaje de broadcast y los servidores estar en la misma red estos recibirán ServerName | ip | {lista dibujos}, las cuales debe ignorar.

# *Tipos de Solicitudes y Respuestas*

**Petición ASCII-art tenedor a servidor:**

**Request:** GET /figure/{nombre\_figura}

**Respuesta del servidor:**

Si la figura existe: (Se envía solo el ASCII-art)

```
(\__/\n(='.'=)\n(\")_(\")
```

**Si la figura no existe:** el Tenedor tiene la tabla de figuras actualizada por lo que no hará la solicitud a ningún Servidor y retorna un código de error: **404 Not Found**

```
404 Not Found
```

**Servidor notifica muerte al Tenedor:**

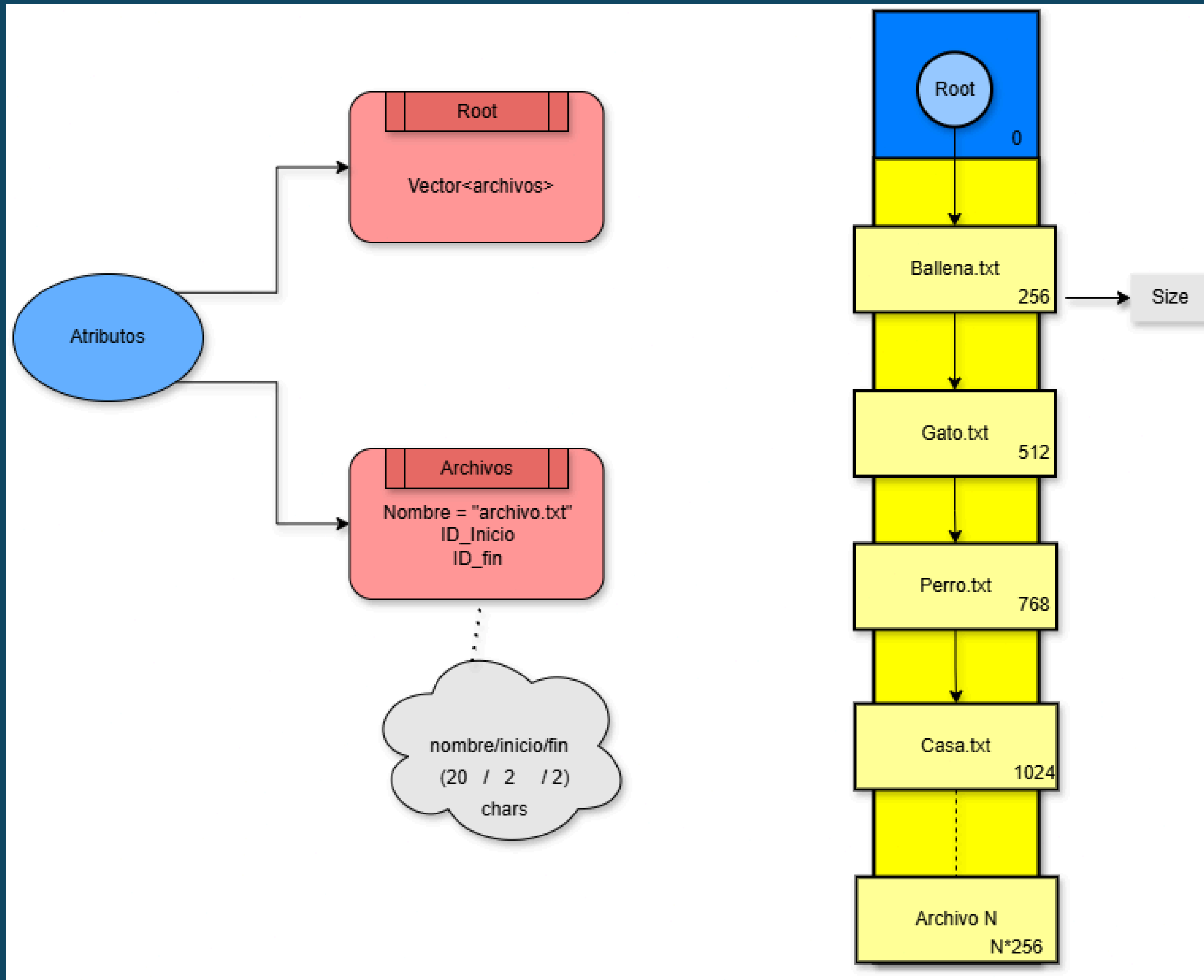
El servidor manda el siguiente mensaje (broadcast) y se “muere”.

**Request:** Shutdown ServerName

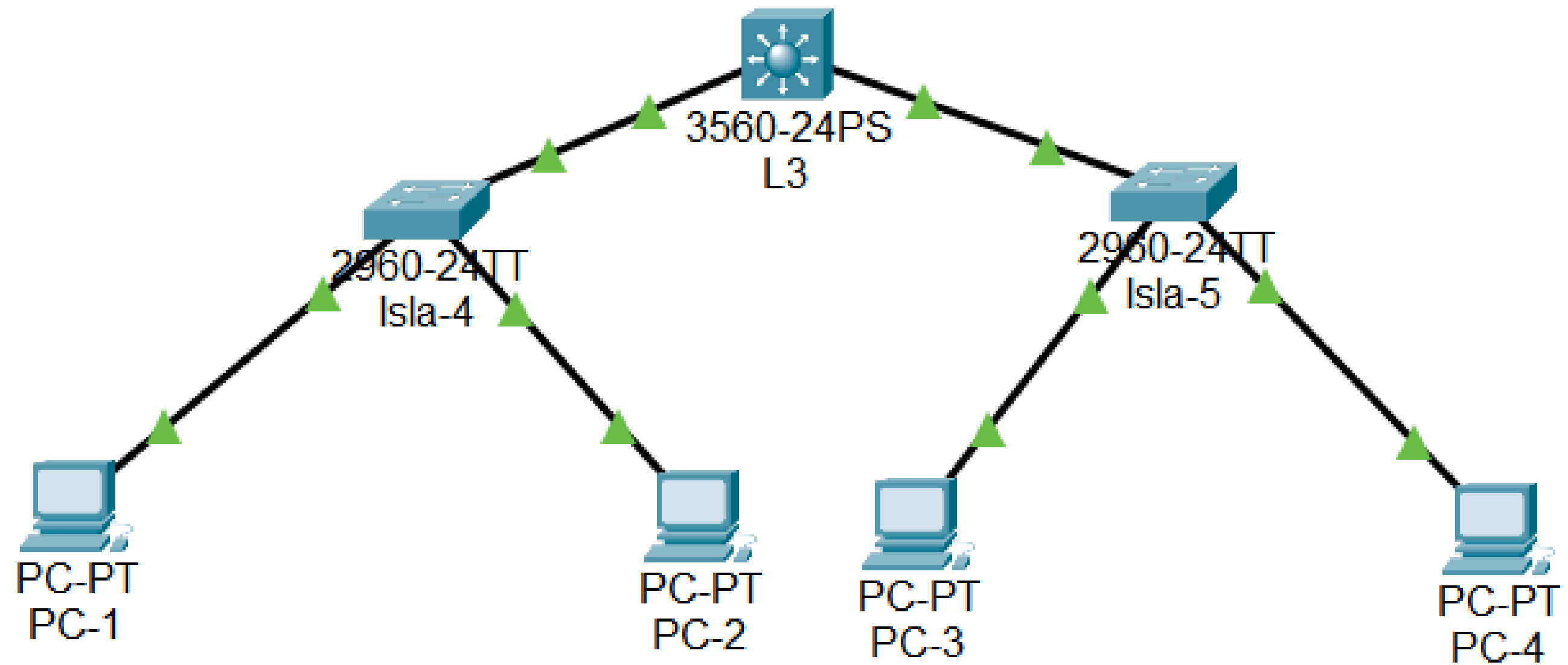
```
Shutdown ServidorA
```

- Como es un broadcast, cuando a un servidor le llega la petición “Shutdown ServerName” la ignora.
- Cada Tenedor se hará cargo de limpiar su registro de dibujos al recibir la notificación de la muerte de algún servidor.

# Diseño File System

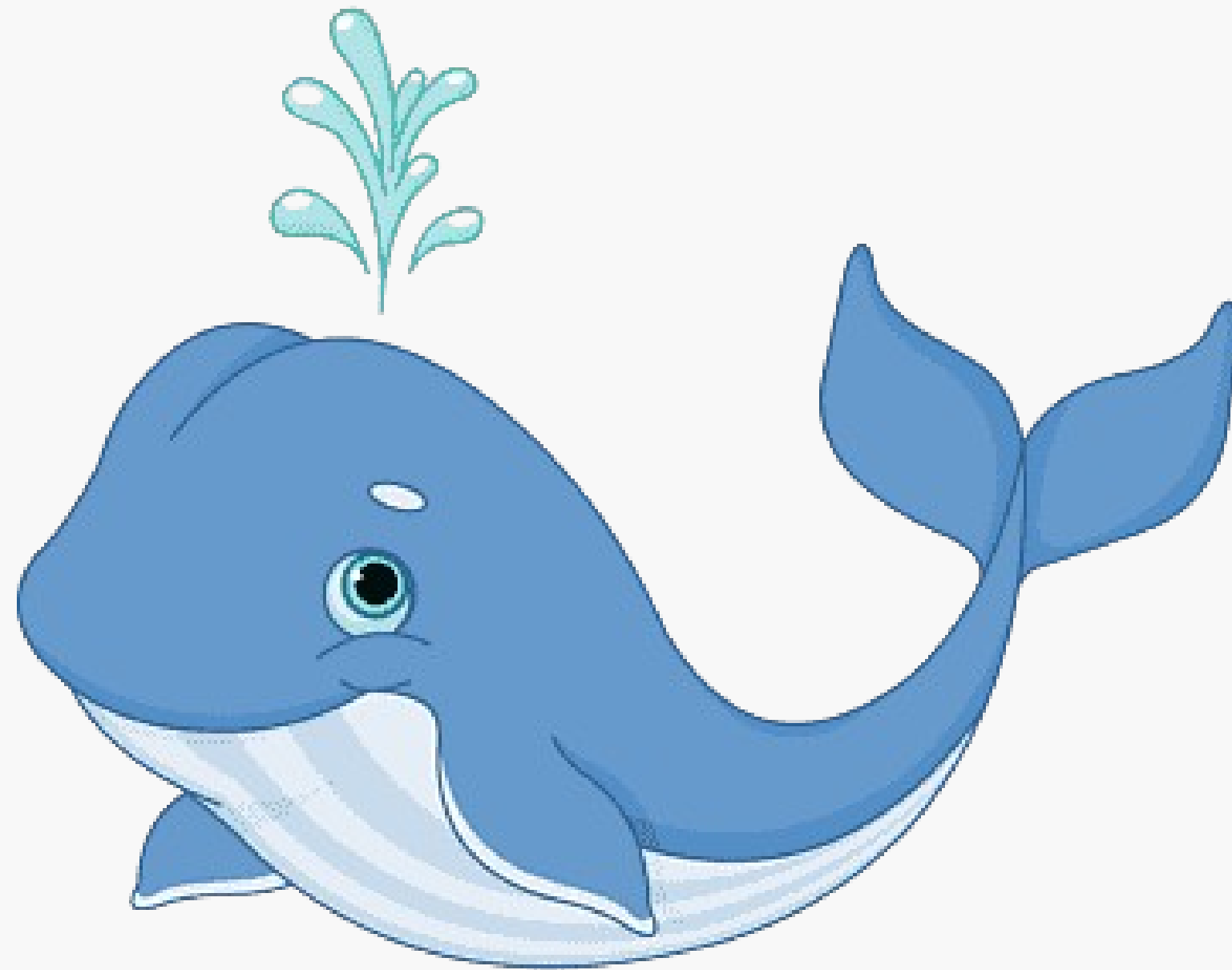


# Packet Tracer



# *Simulación y Server-Cliente*

---





*Gracias!*

