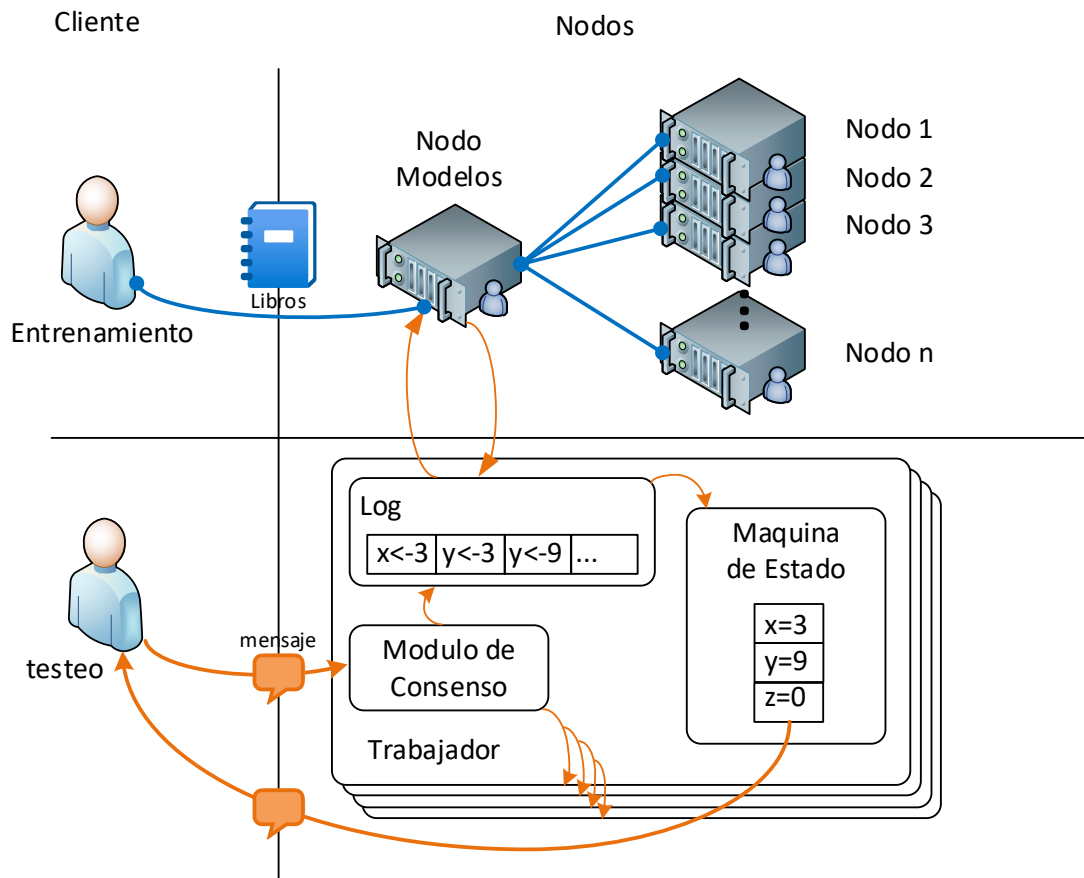


# Escuela de Ciencias de la Computación

## Practica 04 2024-II

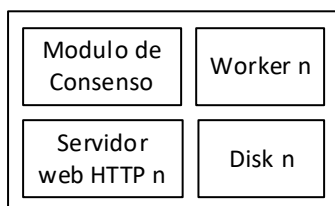
### CC4P1 Programación Concurrente y Distribuida

Se solicita crear un sistema de “backpropagation que se entrena con libros” distribuido con un algoritmo de consenso para el testeo.

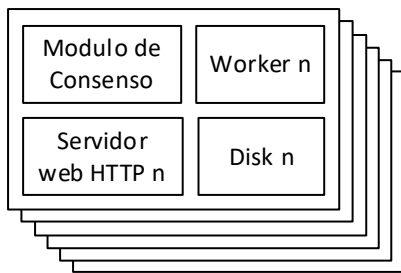


#### Workers (Trabajadores)

Los workers  $n$  tienen embebido un servidor web para poder administrarlo y el módulo de consenso. El módulo de consenso escoge tiene el líder de los workers que tiene el algoritmo de consenso. El Servidor web tiene por objetivo mostrar el monitor del respectivo worker  $n$ .

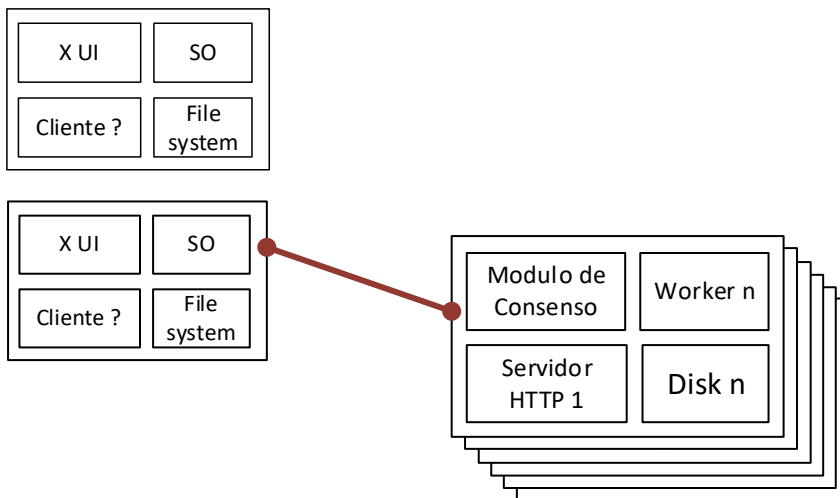


Se harán replicas de los archivos compartidos con el algoritmo de consenso.



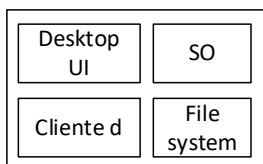
Cliente

Los clientes son desktop y móvil se conectará con el worker líder escogido mediante el algoritmo de consenso. Las funciones del “Cliente ¿” es poder subir archivos, listar y descargar archivos.

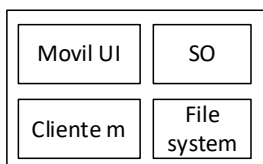


Los clientes son de dos tipos: desktop y móvil.

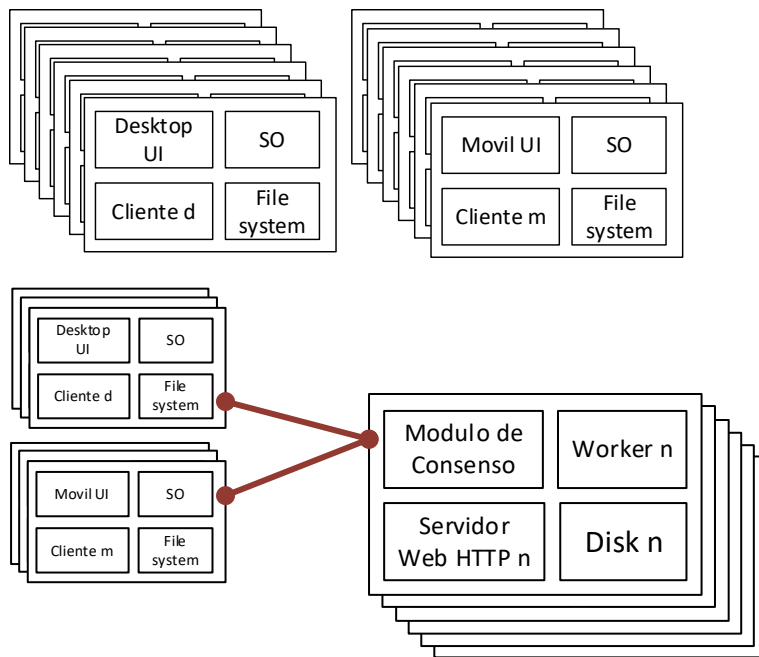
Cliente “Desktop d” se puede ejecutar



Cliente “Móvil m”



Pueden tener varios clientes desktop que suben, consultan y descargan archivos.



- Se pide escribir un código en = {LP1, LP2}, exponer y redactar un informe, por cada alumno adicional se agregará uno más {LP3, LP4, ...}.
- Se iniciará los nodos “workers n” luego se inician los clientes y recibirá las peticiones de Clientes.
- Sólo el módulo del entrenamiento de IA puede estar solo en Java versión mínima 8.
- Tomar como base las explicaciones y el código de clase.
- Exponer y Ejecutar en cluster, con = {LP1, LP2, ...}, para poder verificar el consenso.
- Para validar la consistencia de los archivos, se podrán visualizar en los monitores de servidores web de los “workers n”.
- Sistema Operativos, SO1 y SO2, donde  $SO1 \neq SO2$  y  $SO1 = SO2 = \{LP1, LP2, \dots\}$ .
- Graficar la arquitectura diseñada.
- Graficar el diagrama de protocolo.
- Usar solo Sockets de cada Lenguaje de programación.
- No usar websocket, socketio, frameworks, RabbitMQ, MQ, Librerías de Comunicación, etc.
- Explicar el desarrollo del programa puntualmente.
- Desplegar el programa en redes Lan y wifi.
- Evaluar el desempeño haciendo un script donde el cliente ingrese aleatoriamente 1000 nuevos archivos o más, para evaluar desempeño del algoritmo de consenso.
- Poder ingresar cada cliente, los registros uno por uno, y visualizarlos en los monitores de los “workers n”.
- Como base en los “workers n” realizar el código en terminal y sus monitores puede ser visualizados de otra PC con SO que tenga interfaz gráfica. Los clientes pueden tener interfaz gráfica para poder subir, listar y descargar archivos.
- Usar hilos para mejorar el desempeño y evitar corrupción de registros.

Subir en Univirtual.

- Comprimido consta
  - o Códigos fuente de extensión en el LP1, LP2 y LP3.
  - o PDF Informe.
  - o Presentación.
  - o Los grupos mayores a 2 alumnos desarrollaran un lenguaje de programación adicional LP3, ... para otro nodo.