

# Sistemas Distribuidos

---

ECSDI

Curso 2021/2022

CS-FIB-UPC 



# Preliminares

---

1. Bajad el código de <https://github.com/bejar/ECSDI2022/> o haced:  
`git clone https://github.com/bejar/ECSDI2022.git`
2. Bajad el documento de laboratorio, seguiremos la explicacion de como funciona Flask en el capítulo 3
3. Ejemplos de flask en `Examples/flask`

# Sistemas distribuidos

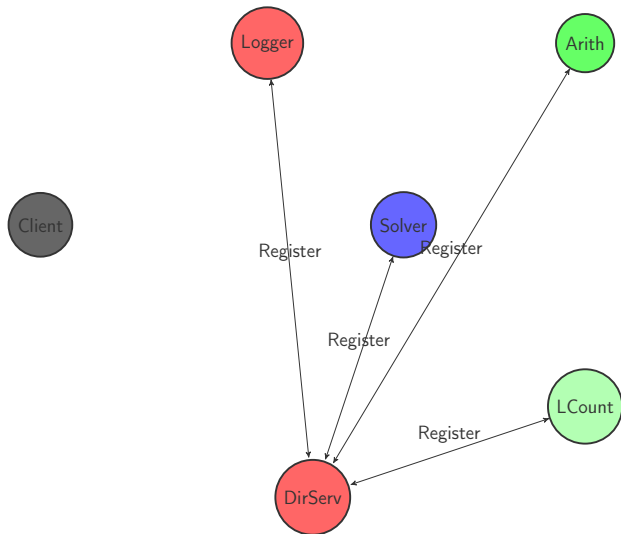
---

- ⊙ Ejemplos de sistemas distribuidos sencillos
- ⊙ Implementados sobre API REST mediante el framework Flask
- ⊙ Usamos REST como pasarela
- ⊙ El objetivo es exponer las interioridades de la comunicación y coordinación de agentes/servicios

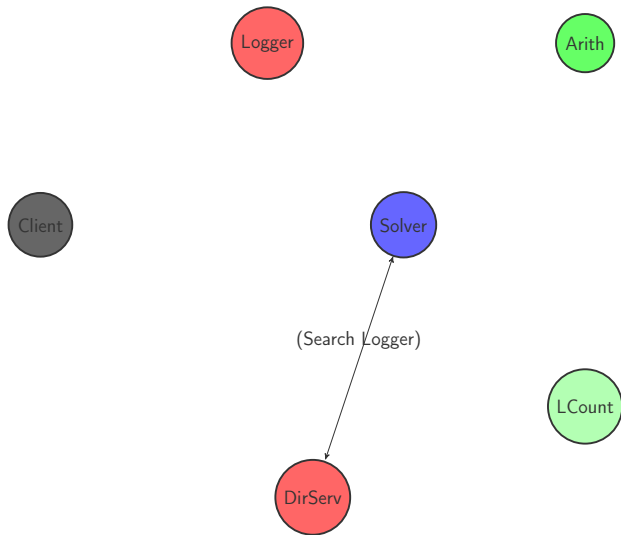
Resolución versión abierta

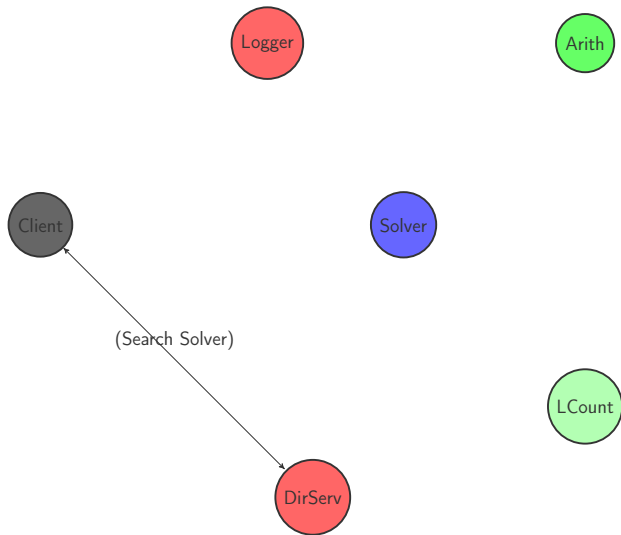
---

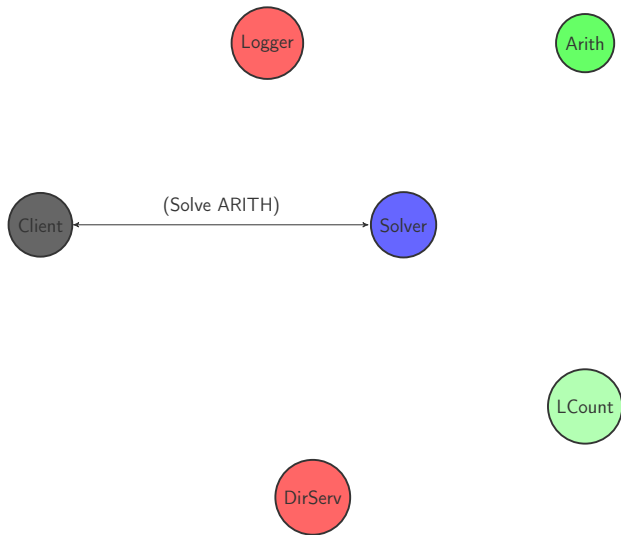
- ⊙ Resolución de tareas simples mediante servicios distribuidos
- ⊙ **Infraestructura:** Servicio de descubrimiento (DirectoryService)
- ⊙ **Frontend:** Múltiples instancias de resolvidor de tareas
- ⊙ **Backend:** Múltiples instancias de resolvidores
- ⊙ **Cliente:** Cliente web para interactuar con el servicio de resolución
- ⊙ **Versión Abierta** porque todos los agentes están públicos en el servicio de directorio
- ⊙ Nada impide solicitar a los resolvidores específicos que hagan tareas

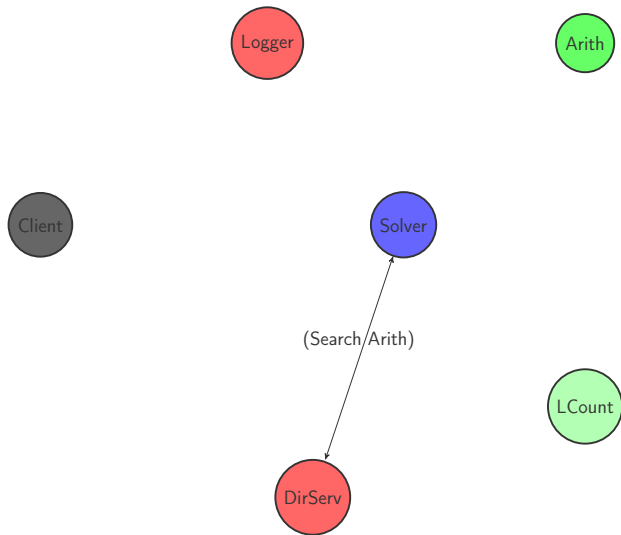


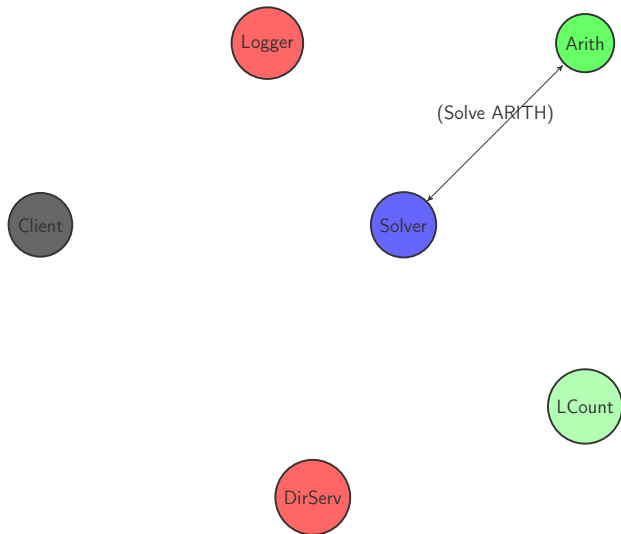


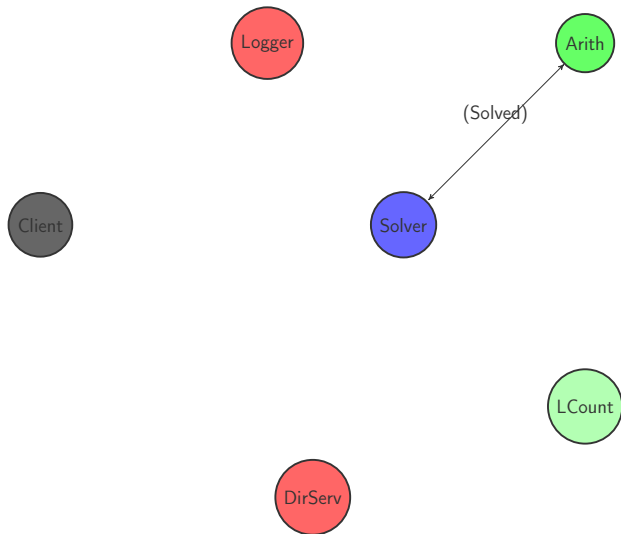


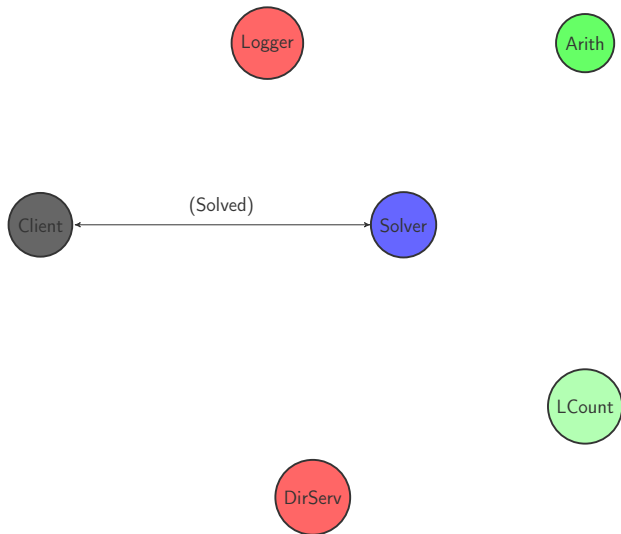


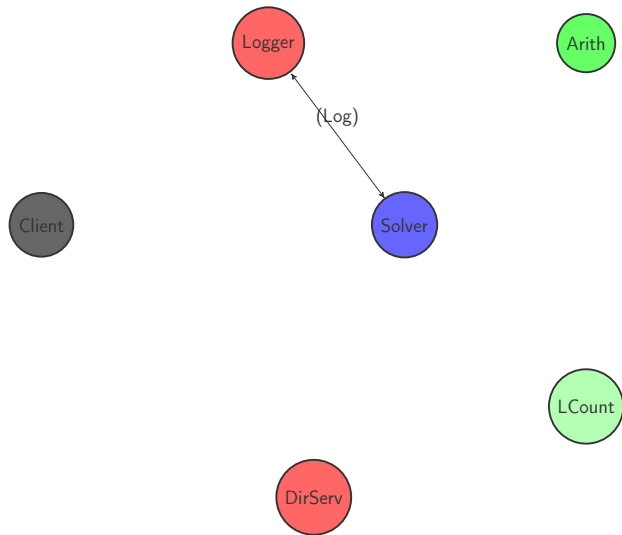














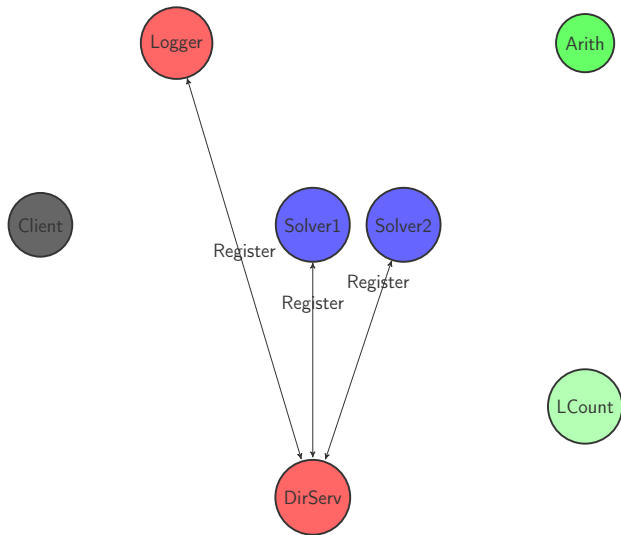
- ⊙ Primero ejecutaremos el sistema en local
- ⊙ Leed el `Readme.txt` del directorio `DistributedSolverOpen`
- ⊙ Poned en marcha los agentes en el orden indicado
- ⊙ Resolved algunos problemas

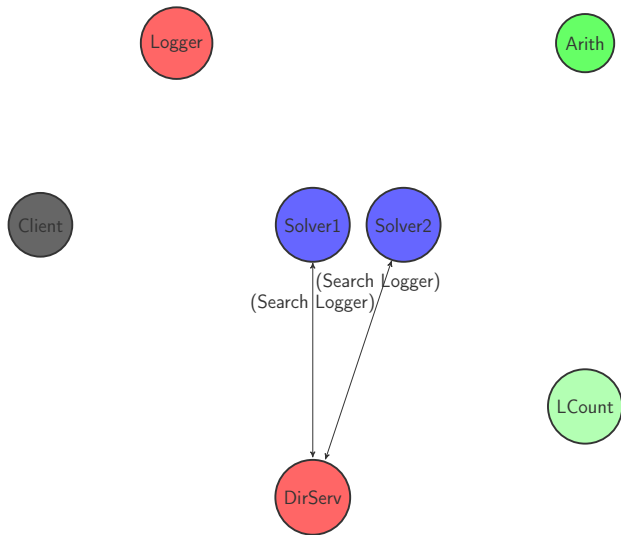
- ⊙ Esperad a que estén en marcha el servicio de directorio y logger
- ⊙ Poned en marcha agentes Solver, Arithmetic y LetterCounter
- ⊙ Cruzad los dedos
- ⊙ Resolved algunos problemas

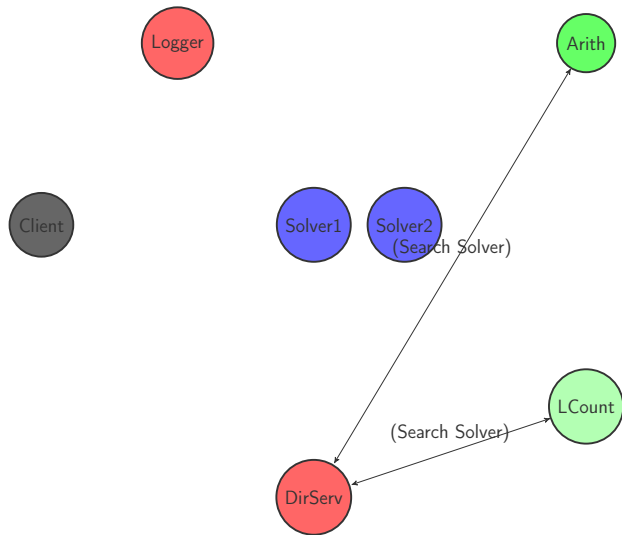
Resolución versión cerrada

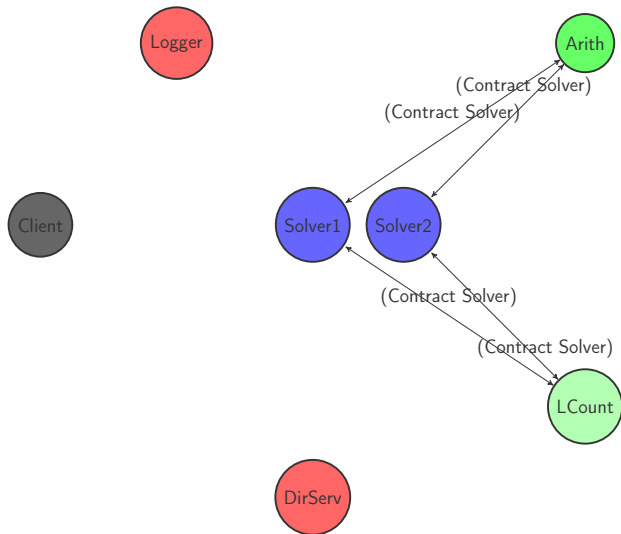
---

- ⊙ Resolución de tareas simples mediante servicios distribuidos
- ⊙ **Infraestructura:** Servicio de descubrimiento (DirectoryService)
- ⊙ **Frontend:** Múltiples instancias de resolutor de tareas
- ⊙ **Backend:** Múltiples instancias de resolvers
- ⊙ **Cliente:** Cliente web para interactuar con el servicio de resolución
- ⊙ No hemos cambiado los agentes :-)
- ⊙ Ahora los resolvers específicos se contratan con los resolvers genéricos, no son visibles a otros agentes

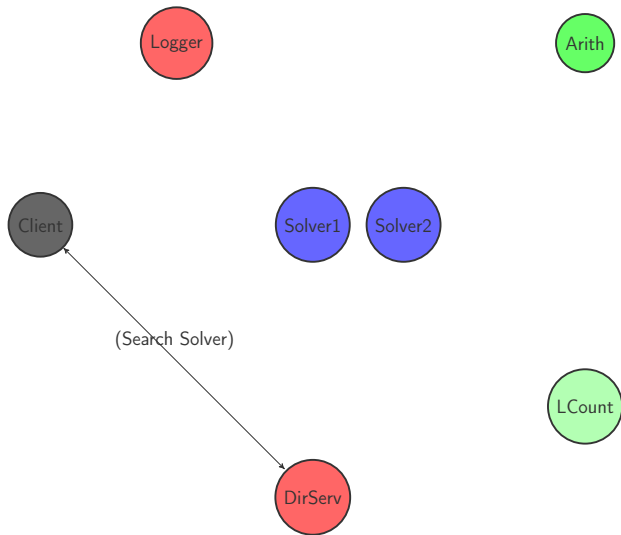


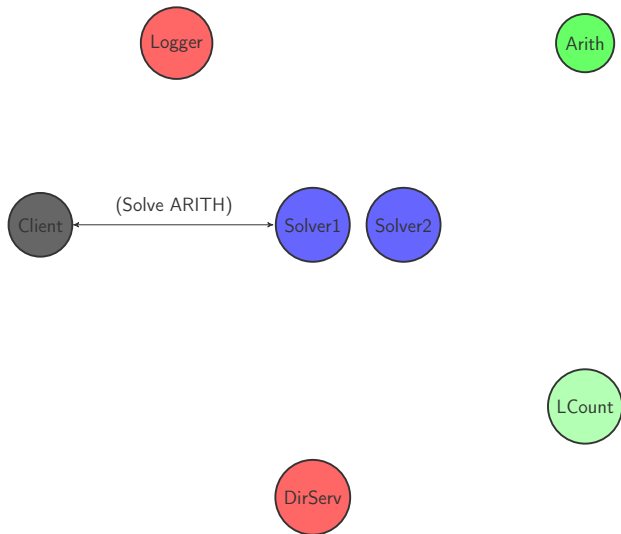


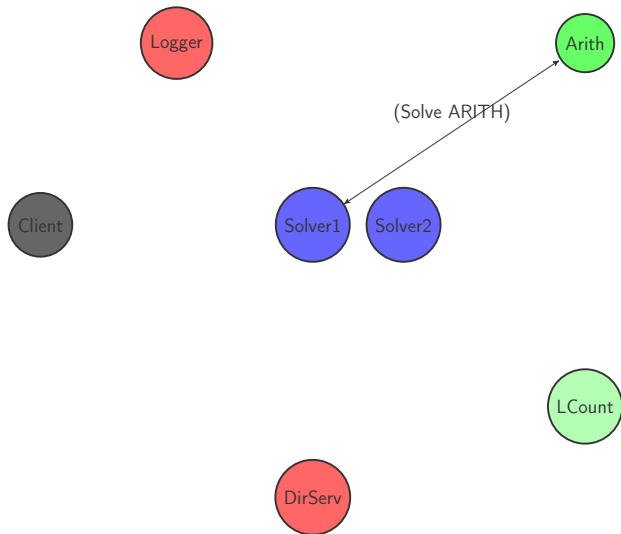


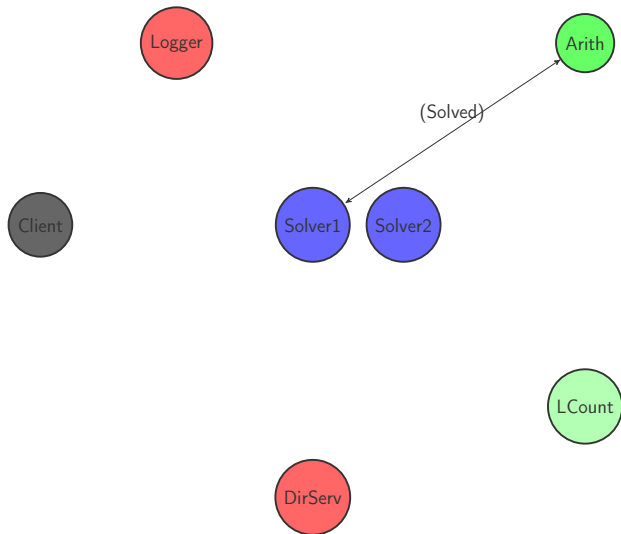


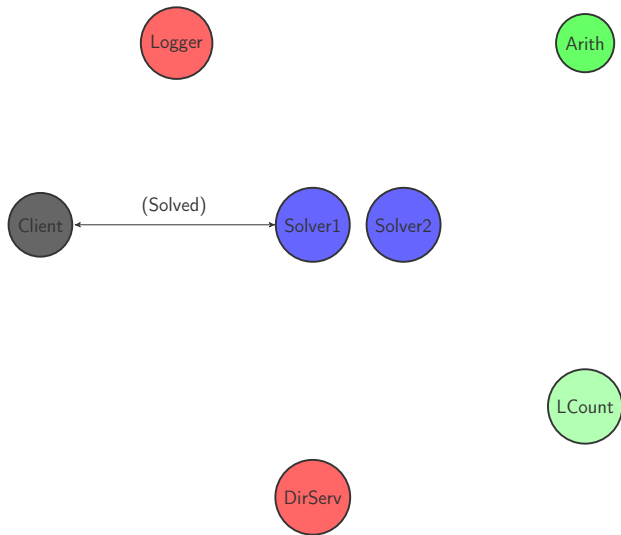


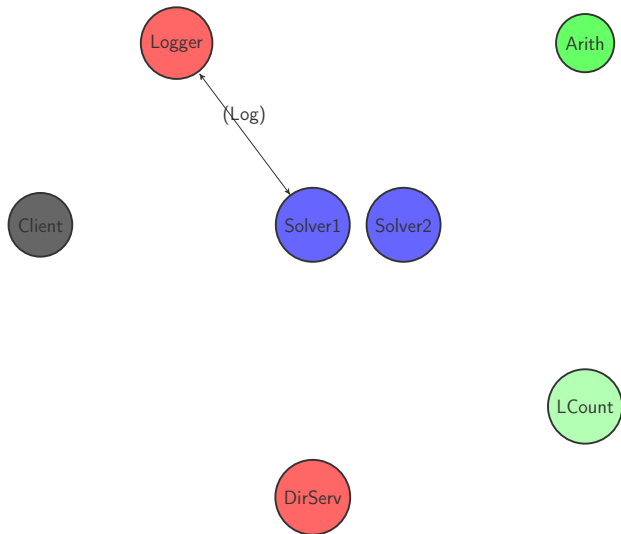












- ⊙ Leed el `Readme.txt` del directorio `DistributedSolverClosed`
- ⊙ Esperad a que estén en marcha el servicio de directorio y logger
- ⊙ Poned en marcha agentes `Solver` primero y `Arithmetic` y `LetterCounter` después
- ⊙ Cruzad los dedos
- ⊙ Resolved algunos problemas

# Ejercicio

---



- ⊙ Es un problema que solo haya un servicio de directorio que centralice toda la información
- ⊙ Podemos implementar una versión de SD que se conecte con otros SD cuando en una petición no encuentre lo que busca
- ⊙ Cada SD debería recibir al menos la dirección de otro SD en su inicio (atención a la topología de conexión)
- ⊙ Para evitar ciclos deberemos contar los saltos hechos en la petición