

Escuela de Ciencias de la Computación
Practica 04 2025-II
CC4P1 Programación Concurrente y Distribuida

Implementación de un Sistema Distribuido para el entrenamiento de IA de reconocimiento de objetos, animales o personas que requiera entrenamiento.

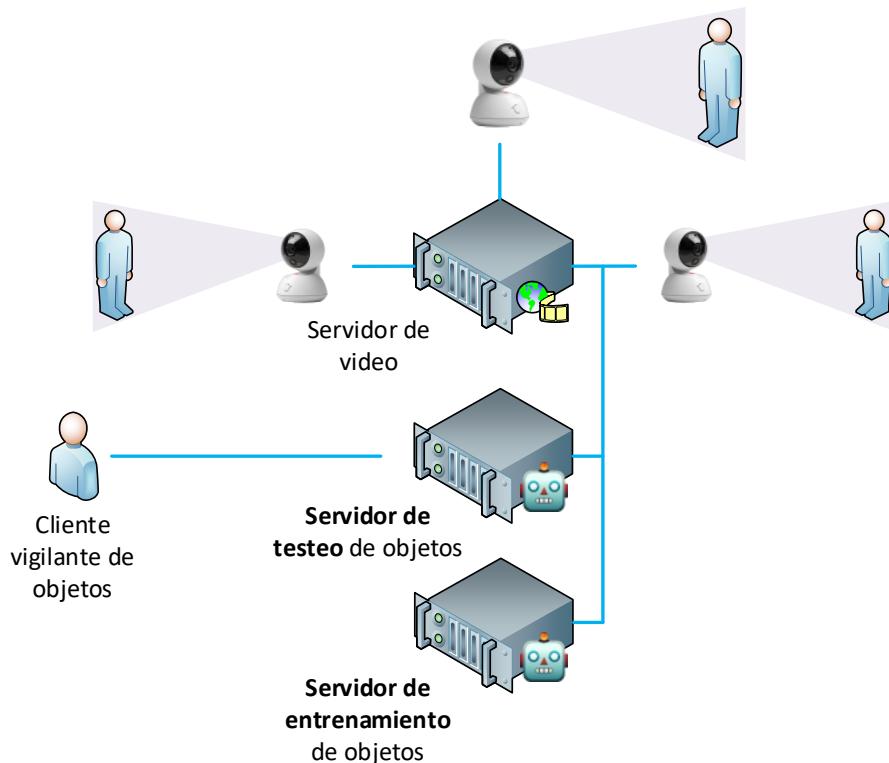
El objetivo de esta evaluación es diseñar e implementar un sistema distribuido que permita el entrenamiento y consumo de modelos de inteligencia artificial (IA) de manera paralela, distribuida y concurrente.

Entrenamiento Distribuido de Modelos de IA (Servidor de Entrenamiento)

Se debe desarrollar un sistema distribuido en el cual un cliente envíe datos de entrada (inputs) y salida (outputs) a un conjunto de servidores (nodos). Estos servidores pueden ser manera secuencial (se considerará si se procesa en paralelo distribuido tipo cpu), gestionando los recursos de manera distribuida y concurrente.

Requisitos:

- La parte del entrenamiento es de acuerdo al modelo del grupo.
- La carga de trabajo puede ser secuencial, en caso que sea paralelo distribuido la carga de trabajo debe distribuirse entre los nodos del sistema para optimizar el proceso de entrenamiento.
- Se debe garantizar la persistencia y accesibilidad de los modelos entrenados para su posterior consumo.
- Se tiene que entrenar con un grupo "n" de objetos, animales o personas que se pueda reconocer, se tomará en cuenta si mayor es el n para la evaluación, por ejemplo, n=2 puede reconocer perros y gatos.

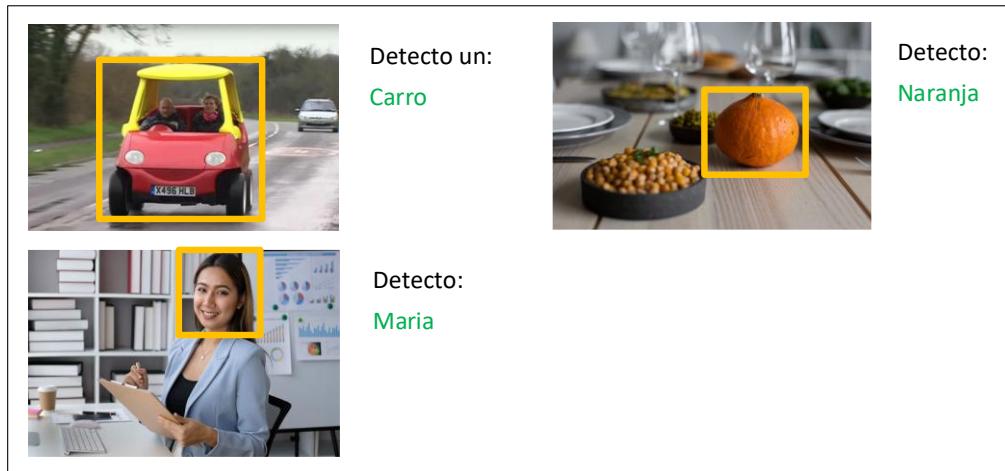


Consumo de Modelos de IA (Servidor de Testeo)

El sistema debe permitir que la cámara enésima use un modelo de IA modelo de IA previamente entrenado y según **identifique los objetos y guarde una imagen** (objetos, animales o personas) **en un archivo y agregue en un registro que objeto se acercó y su imagen**.

Requisitos:

- El servidor de testeo de objeto debe poder usar su modelo de IA y autónomamente reconocer los objetos entrenados.
- El modelo debe procesar la entrada de los frames del video y devolver la salida esperada de manera eficiente por cada cámara.
- Para un numero “c” de cámaras, mientras más cámaras se considerará



Vigilante de objetos (Cliente Vigilante)

El cliente vigilante de objetos puede ver visualmente el **registro que objetos identificados, su foto, la fecha, hora de la captura y la cámara** enésima que lo proceso el **servidor de testeo** de manera continua.

Encontro	Tipo	Encontro	Fecha Hora
	Mujer		01/11/2025 03:25
	Naranja		01/11/2025 15:20
	Loro		03/11/2025 2:30
	Carro		05/11/2025 14:00



- Se pide escribir un código en = {LP1,...}, puede ser con uno o más de un lenguaje de programación, exponer y redactar un informe, se tomará en cuenta si usa más lenguajes de programación.
- Se iniciará los nodos servidor de video, servidor de testeo de objetos y servidor de entrenamiento de objetos, luego se inician los clientes y recibirá las peticiones de Clientes.
- El módulo del entrenamiento de IA puede estar en el lenguaje de programación que decida.
- Tomar como base las explicaciones.
- Exponer y Ejecutar en cluster, con = {LP1,...}, u otro según sea el caso, ya sea virtual local o en las pcs de su grupo.
- Graficar la arquitectura diseñada.
- Graficar el diagrama de protocolo.
- Usar solo Socketsy el protocolo rtsp o similar de acuerdo su cámara ip.
- 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.
- Usar hilos para mejorar el desempeño y evitar corrupción de registros.

Subir en Univirtual.

- Comprimido consta
 - Códigos fuente de extensión en el LP1, LP2, LP3, ...
 - PDF Informe.
 - PDF Presentación.
 - Los grupos mayores a 2 alumnos pueden mejorar el “n” o “c”.