

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA

Arquitectura de Computadoras II
Sección B

PROYECTO FINAL

NexoVisión

Informe técnico

**Diseño, montaje e implementación de un clúster de
Edge Computing para inferencia de IA local autónoma**

Presentado por

Christopher José Hernández Gramajo
Carné 0900-23-8598

Docente

Alejandro Alberto García Arrecis

16 de octubre de 2026

1. Resumen, objetivos y alcance

1.1 Resumen

Se implementó un clúster de inferencia local con una HP ProBook como nodo coordinador y procesador, y una HP Pavilion como worker remoto. Una cámara integrada en la ProBook proporciona fotogramas; el coordinador los comprime como JPEG y publica tareas en una cola compartida. Ambos workers ejecutan el modelo en CPU y devuelven detecciones y tiempos al coordinador. Un panel local muestra CPU, RAM, tráfico de red y disponibilidad del agente de cada equipo.

En la prueba de carga monitorizada, una ventana de 355.243 segundos produjo 6,561 resultados, equivalentes a **18.47 resultados por segundo**. La ProBook procesó 3,890 y la Pavilion 2,671. En un ensayo separado de continuidad, durante 52.86 segundos sin resultados del worker remoto, el master entregó 790 inferencias y posteriormente se registraron nuevamente resultados de ambos nodos.

Estos resultados demuestran procesamiento distribuido, observación de recursos y continuidad frente a la ausencia del worker. No demuestran por sí solos aceleración frente a un único nodo, recuperación de todos los fotogramas pendientes ni alta disponibilidad de todos los componentes.

1.2 Objetivo general

Diseñar, montar y evaluar un sistema local de inferencia de visión por computadora que distribuya procesamiento entre dos nodos CPU, utilice una red Ethernet privada y mantenga la entrega de resultados ante la ausencia del worker remoto.

1.3 Objetivos específicos

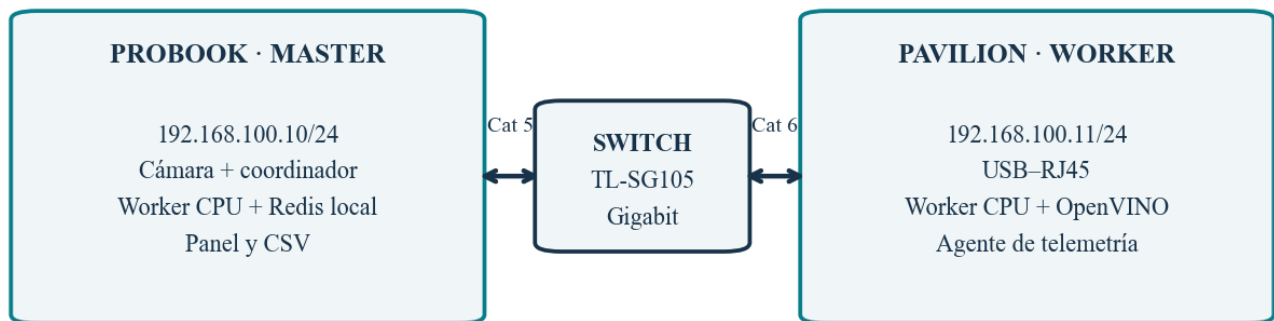
- Interconectar los nodos mediante switch Gigabit y direccionamiento IPv4 estático en una subred dedicada.
- Ejecutar un modelo ligero en CPU y enviar fotogramas mediante un middleware local de tareas.
- Registrar latencia, throughput y recursos con un método de medición explícito.
- Simular la ausencia del worker, observar continuidad y documentar su reincorporación y limitaciones.
- Preparar una demostración operativa sin comunicación externa a Internet y conservar instrucciones reproducibles.

1.4 Alcance académico y condición de hardware

Se seleccionó el caso de visión por computadora del enunciado. La demostración debe utilizar productos apropiados para las etiquetas del modelo y una cámara conectada directamente al clúster. La instrucción de operar sin comunicación externa a Internet se refiere al sistema demostrado; las descargas iniciales se realizaron durante la preparación.

El enunciado establece el uso de plataformas de borde de bajo consumo. Para la implementación se emplearon dos computadoras portátiles, de acuerdo con la aclaración docente recibida durante el desarrollo. Esta autorización se distingue de las condiciones expresadas en el documento original. El alcance del trabajo comprende la arquitectura de inferencia local seleccionada; la documentación escrita de dicha aclaración no se incluye entre las evidencias disponibles.

2. Arquitectura física y lógica



Ethernet privado 192.168.100.0/24 · enlaces verificados a 1000 Mb/s

Redis en loopback · acceso remoto por túnel SSH · sin nube de inferencia

Figura 1. Arquitectura física y servicios del clúster.

La subred dedicada es 192.168.100.0/24. Cada equipo dispone de una IP fija y se conecta al switch TP-Link TL-SG105 de 5 puertos Gigabit; este no requiere conexión al router para la operación del clúster. La ProBook usa su interfaz Ethernet integrada con cable Cat 5 y la Pavilion un adaptador USB-RJ45 con cable Cat 6. Los registros verificaron enlaces negociados de 1000 Mb/s en ambos extremos.

Elemento	ProBook · master	Pavilion · worker
Hostname	arq2-master	arq2-worker
IP Ethernet	192.168.100.10/24	192.168.100.11/24
Interfaz	enp2s0	enxec9a0c000d47
Procesamiento	Coordinador y worker local	Worker remoto
Persistencia de tareas/resultados	Redis local	Acceso a Redis por túnel SSH
Observación	Panel y CSV	Agente HTTP de telemetría

2.1 Recorrido de un fotograma

1. OpenCV captura el fotograma desde la cámara de la ProBook.
2. El coordinador codifica JPEG con calidad 80 y lo transforma a base64.
3. Celery publica la tarea `arq2.infer_frame` en la cola `cluster`.
4. Un worker disponible decodifica la imagen y ejecuta OpenVINO en CPU.
5. El worker devuelve hostname, detecciones y tiempos mediante el backend Redis.
6. El coordinador recoge el resultado, dibuja cajas sobre el fotograma asociado y escribe una fila CSV.

2.2 Decisiones de diseño

La cola compartida permite que cualquiera de los workers tome nuevas tareas. Se usó concurrencia 1 por worker y prefetch 1 para limitar reservas anticipadas. Esto no significa utilizar un solo núcleo: OpenVINO puede utilizar múltiples hilos dentro del proceso de inferencia.

Redis permanece enlazado a loopback en el master. Un túnel SSH inverso ofrece al worker un puerto local que reenvía hacia Redis de la ProBook. Así se evita exponer directamente Redis en la LAN, a costa de introducir dependencia del proceso SSH. La solución implementa continuidad del procesamiento con más de un consumidor; no introduce redundancia del switch, Redis ni coordinador.

3. Plataforma, software y servicios

3.1 Equipos utilizados

Plataforma	Características documentadas
HP ProBook 460 G11	Intel Core Ultra 7 155U; 14 procesadores lógicos reportados por <code>lscpu</code> , diferenciados de los núcleos físicos. 16 GB de RAM; Ubuntu y entorno Python 3.12.
HP Pavilion x360 15-dq0xxx	Intel Core i5-8265U, 12 GB de RAM, HDD WDC de 1 TB comercial; Ubuntu 24.04 con escritorio Xubuntu y Python 3.12.3.
Switch	TP-Link TL-SG105 de 5 puertos Gigabit, identificado mediante el empaque del equipo.
Cableado	ProBook al switch: Cat 5. Pavilion al switch: Cat 6, mediante adaptador USB-RJ45. Ambos enlaces negociaron 1000 Mb/s.

La categoría del cable hacia ProBook se registra como Cat 5, sin asumir que sea Cat 5e. La velocidad negociada de 1000 Mb/s es un resultado observado de esta instalación, no una certificación del cableado ni una garantía de su rendimiento en cualquier longitud o condición.

La Pavilion utiliza una partición Linux de aproximadamente 100.369 GiB, conservando Windows en otra partición y reutilizando la EFI existente sin formatearla. La preparación desacopló la memoria Optane. El procedimiento de instalación no forma parte del flujo normal de operación del clúster.

3.2 Componentes de software

- **YOLO11n / Ultralytics:** detección de objetos con un modelo ligero, versión de Ultralytics utilizada 8.4.177.
- **OpenVINO:** ejecución del modelo exportado en CPU; archivos del modelo almacenados en ambos nodos.
- **OpenCV y NumPy:** captura, compresión y reconstrucción de fotogramas, y visualización de resultados.
- **Celery y Redis:** distribución de tareas y recuperación de resultados por una cola común.
- **OpenSSH:** administración y transporte del puerto Redis entre equipos mediante Ethernet.
- **psutil y monitor local:** recolección de recursos, panel web sin bibliotecas externas y escritura CSV.
- **Git:** historial local de código, manifiestos de paquetes, resultados y documentación.

3.3 Matriz de servicios

Servicio	Protocolo y puerto	Dependencia / alcance
Redis master	TCP 127.0.0.1:6379	Broker y backend; debe estar activo antes de los workers.
SSH Pavilion	TCP 192.168.100.11:22	Túnel supervisado por <code>systemd</code> ; worker remoto independiente de la sesión.
Redis visto en Pavilion	TCP 127.0.0.1:6379	Reenvío remoto al loopback del master por SSH.
Agente de monitor	HTTP 192.168.100.11:8765	Telemetría de Pavilion, independiente de la sesión Celery si se abre localmente.
Panel local	HTTP 127.0.0.1:8766	Navegador en ProBook y CSV de ambos equipos.
Cola cluster	Mensajería Celery sobre Redis	Coordinador y los dos workers usan el mismo nombre de cola.

4. Implementación y controles operativos

4.1 Confirmación y distribución de tareas

La aplicación configura mensajes y resultados en formato JSON, acepta únicamente este formato y conserva los resultados durante 300 segundos. La configuración implementada se presenta a continuación:

```
app.conf.update(  
    task_serializer="json",  
    result_serializer="json",  
    accept_content=["json"],  
    worker_prefetch_multiplier=1,  
    task_acks_late=True,  
    task_reject_on_worker_lost=True,  
    result_expires=300,  
)
```

`task_acks_late` confirma la tarea después de ejecutarla. `task_reject_on_worker_lost` permite rechazar/reencolar tareas cuando se pierde el proceso de ejecución bajo un worker cuyo proceso principal puede detectar esa pérdida. No garantiza recuperación inmediata si el nodo completo o su red desaparecen.

4.2 Coordinador e instrumentación

Se añadió la opción `--queue` y se cambió el envío para dirigir tareas a una cola concreta. Cada solicitud se asocia al fotograma original y a un reloj monótono. La medición se inicia antes de publicar, para incluir el coste del envío, y termina al recoger el resultado:

```
sent = time.perf_counter()  
task = infer_frame.apply_async(  
    args=[jpeg_base64, args.imgsz, args.conf],  
    queue=args.queue,  
    expires=args.timeout,  
)
```

El número de tareas pendientes se limita para evitar acumulación ilimitada. En la prueba de carga se ejecutó `--queue cluster --timeout 30 --max-pending 2`. Para el ensayo de ausencia del worker se usó un timeout de 5 segundos después del calentamiento; un límite corto puede descartar tareas lentas. La expiración no cancela por sí sola una inferencia que ya comenzó.

4.3 Telemetría

El agente de Pavilion consulta CPU, RAM, uptime del sistema operativo, tráfico y velocidad del enlace, y procesos Celery. El panel del master recoge esa telemetría por HTTP y sus propios recursos localmente. El objetivo es una muestra por segundo; una consulta remota fallida puede alargar el periodo efectivo hasta superar un segundo, por lo que el análisis usa los timestamps reales.

CPU es el porcentaje del equipo completo, promediado entre los procesadores lógicos. Las tasas RX/TX incluyen todo el tráfico de la interfaz y se expresan en megabits por segundo. La latencia del agente es RTT HTTP de telemetría, distinto del RTT de inferencia. El contador Celery incluye procesos padre e hijo; dos procesos no significan dos nodos independientes ni estado ready confirmado.

Las muestras sin respuesta no reutilizan valores antiguos ni se sustituyen por cero. Los eventos Inicio, Desconecto, Reconecto y Fin registran los clics del usuario; no detectan automáticamente la acción física.

5. Metodología de evaluación

5.1 Pruebas realizadas

Ensayo	Propósito	Evidencia disponible
Inferencia local	Verificar modelo, cámara y ejecución CPU	CSV y resúmenes iniciales; condiciones distintas de la prueba final.
Tarea remota única	Verificar recorrido ProBook → Pavilion → ProBook	Resultados Celery con hostname <code>arq2-worker</code> .
Video remoto y compartido	Verificar flujo continuo y reparto	CSV 120554, 121210 y ensayos posteriores.
Ausencia y retorno de worker	Observar continuidad al desconectar Ethernet	<code>cluster_20261016_121746.csv</code> y registro descriptivo del procedimiento.
Carga con monitoreo	Relacionar tareas y utilización de recursos	<code>cluster_20261016_142136.csv</code> , CSV de monitor y eventos.

Las pruebas iniciales con detecciones de persona y televisión permitieron verificar el funcionamiento de la arquitectura. Posteriormente se documentó la detección de una taza, presentada en el anexo B como evidencia cualitativa del procesamiento de objetos. Los archivos CSV de inferencia almacenan la cantidad de detecciones, pero no sus clases ni imágenes; por ello, no permiten evaluar la precisión del modelo.

5.2 Definiciones de métricas

Métrica	Definición y límite
Inferencia	Tiempo reportado por el modelo durante ejecución; no incluye toda la tubería.
Tiempo dentro del worker	Decodificación, carga si corresponde, inferencia y construcción de respuesta.
RTT de inferencia	Desde antes de publicar hasta recoger el resultado; incluye esperas y transporte, sin equiparlo al ping.
Throughput observado	Resultados recogidos por segundo de la ventana seleccionada; no es FPS nativo de captura.
CPU/RAM	Recursos del equipo completo, no únicamente del proceso del modelo.
Disponibilidad de muestras	Consultas respondidas / intentos del monitor; no es disponibilidad integral del servicio.

5.3 Sincronización y selección de la ventana

La prueba monitorizada se delimitó entre las marcas Inicio y Fin: de 14:21:58.319 a 14:27:53.562 del 16 de octubre, con desplazamiento local −06:00, equivalentes a 20:21:58.319–20:27:53.562 UTC. La duración fue de 355.243 s. El coordinador empezó antes y terminó después, de modo que se excluyeron resultados fuera de las marcas. Su arranque antecede aproximadamente 21 s a Inicio.

El CSV del coordinador no tiene zona horaria; se interpretó como hora local del master usando el desplazamiento explícito de los eventos del monitor. P95 se calculó mediante interpolación lineal con índice $(n-1) \times 0.95$. Esta selección evita mezclar reposo con carga, pero no reemplaza un protocolo comparativo idéntico entre configuraciones.

No se utilizaron los resultados de este ensayo para afirmar un factor de aceleración frente a un único nodo. Esa conclusión exigiría repetir el mismo flujo y condiciones de captura, monitoreo, calentamiento y carga con uno y dos nodos.

6. Resultados de carga monitorizada

En el intervalo seleccionado se recogieron **6,561 resultados** en 355.243 s: **18.47 resultados/s**. La ProBook completó 3,890 tareas (59.3 %) y la Pavilion 2,671 (40.7 %). El archivo completo contiene 7,011 resultados; las estadísticas siguientes corresponden únicamente a la ventana marcada.

Métrica	ProBook	Pavilion
Inferencia media	17.83 ms	45.34 ms
Inferencia P95	22.97 ms	71.19 ms
Tiempo medio dentro del worker	21.88 ms	54.35 ms
RTT medio desde el coordinador	40.78 ms	84.04 ms
RTT P95 desde el coordinador	68.42 ms	133.58 ms
CPU media	21.04 %	52.55 %
RAM media	35.21 %	27.77 %

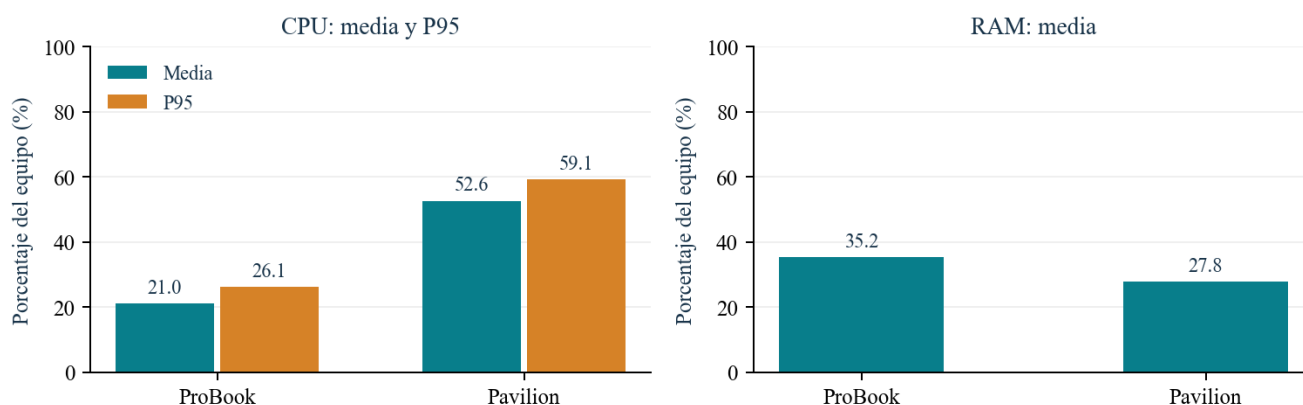


Figura 2. Recursos del equipo completo dentro de la ventana marcada; P95 de CPU y media de RAM.

Los agentes respondieron 355 de 355 muestras por equipo en ese intervalo y reportaron enlace UP a 1000 Mb/s. La telemetría de Pavilion tuvo RTT HTTP medio de 2.65 ms y P95 de 5.34 ms. La CPU P95 fue 26.10 % en ProBook y 59.13 % en Pavilion. Las tasas medias RX/TX estuvieron aproximadamente entre 3.93 y 4.21 Mb/s por dirección en estos registros, muy por debajo de la velocidad nominal del enlace; esto no elimina la sobrecarga de mensajería, serialización o esperas.

El reparto desigual es compatible con nodos heterogéneos y una cola común. La ProBook tuvo inferencia más rápida y mayor participación. La utilización de CPU no equivale a medición energética: no se midieron potencia eléctrica ni eficiencia en vatios por inferencia.

7. Continuidad ante ausencia del worker

Durante el ensayo de continuidad se retiró y se conectó nuevamente el cable Ethernet de Pavilion. El archivo `cluster_20261016_121746.csv` contiene 1,274 resultados: 1,032 del master y 242 del worker. El mayor intervalo entre resultados de Pavilion delimita una ausencia observada de 52.863787 s; el archivo no registra de forma independiente el instante de la desconexión física.

Medida	Resultado
Último resultado remoto antes del intervalo	12:18:22.446435, hora local del CSV
Primer resultado remoto posterior	12:19:15.310222
Intervalo sin resultados remotos	52.863787 s
Inferencias de master dentro del intervalo	790
Frecuencia de entrega durante el intervalo	14.94 resultados/s
Mayor separación entre resultados en ese tramo	154.966 ms

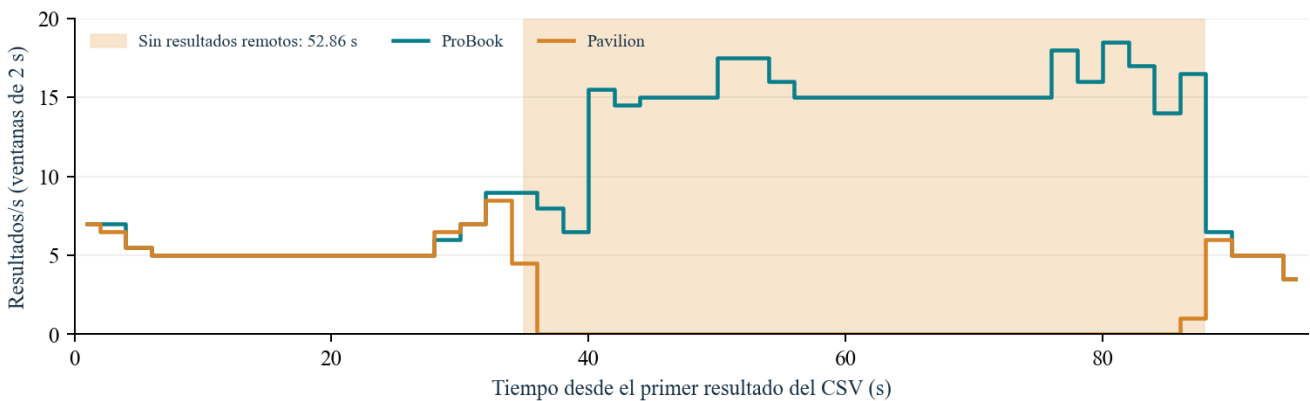


Figura 3. Resultados por segundo, agrupados en ventanas de 2 s. La franja indica ausencia observada de resultados remotos, no el instante exacto de la desconexión física.

La entrega de resultados del master continuó durante el intervalo y luego reaparecieron resultados de Pavilion. Esta evidencia respalda continuidad del servicio ante la ausencia del worker y posterior reincorporación. La recuperación de tareas pendientes y del túnel SSH no se puede deducir automáticamente del CSV.

La ventana observada no mide exactamente la duración física del corte: incluye lo ocurrido entre dos resultados remotos. Tampoco acredita recuperación de todos los fotogramas; el coordinador registra resultados exitosos y puede descartar tareas vencidas. Cero IDs duplicados entre las filas registradas no prueba ejecución exactamente una vez.

La ausencia de un consumidor permite que el otro siga tomando tareas nuevas. Sin embargo, una tarea ya reservada por un nodo desconectado depende del mecanismo de visibilidad de Redis para ser recuperada, cuyo valor predeterminado suele ser una hora salvo configuración distinta. Se documenta esta diferencia entre continuidad del video y garantía de entrega de cada tarea.

7.1 Supervisión y recuperación automática

Después del ensayo de continuidad registrado en CSV, se incorporó supervisión mediante systemd. La configuración separa el túnel SSH, el worker remoto y el agente de monitoreo, de modo que su funcionamiento no depende de mantener terminales abiertas.

Equipo	Servicio	Función
ProBook	<code>arq2-tunnel.service</code>	Establecimiento y reinicio del túnel SSH inverso.
Pavilion	<code>arq2-worker.service</code>	Ejecución y supervisión del worker de Celery.
Pavilion	<code>arq2-monitor-agent.service</code>	Ejecución y supervisión del agente HTTP de telemetría.

El túnel utiliza autenticación por clave, verificación de la identidad del servidor y ejecución no interactiva. Los controles de conexión de SSH permiten detectar la pérdida del enlace y systemd reinicia el cliente cuando termina, con un intervalo de reintento de cinco segundos. Celery reintenta la conexión al broker. Los tres servicios quedaron habilitados para iniciarse con sus respectivos equipos; el ensayo de arranque completo desde un reinicio permanece pendiente.

En un ensayo operativo posterior, se desconectó Pavilion y se mantuvo el procesamiento en ProBook. Al restablecer el enlace, se observó el retorno de nuevas tareas remotas sin reiniciar manualmente el túnel ni Celery. El tiempo de retorno se estimó en aproximadamente diez segundos desde la reconexión del cable. Esta cifra corresponde a una observación del ensayo, no a una medición calculada a partir de los CSV originales, una media estadística o una garantía de recuperación.

Inicialmente, el agente de monitoreo estaba detenido, aunque Celery continuaba procesando. Tras instalar el agente como servicio independiente, se repitió la desconexión y se confirmó el regreso automático tanto de las tareas como de la telemetría. Esta validación fue comunicada durante la ejecución del ensayo; no se incorporaron nuevos CSV para reconstruir de forma independiente sus tiempos.

La recuperación automática documentada se limita al túnel, el worker remoto y su agente. No implica la recuperación de todos los fotogramas pendientes ni el arranque automatizado de la cámara, el panel o el worker local. Los resultados históricos del apartado anterior corresponden a la configuración previa y se conservan sin modificación.

8. Operación autónoma, seguridad y limitaciones

8.1 Demostración sin Internet

La arquitectura utiliza modelos y servicios locales. Las pruebas realizadas mediante las direcciones de la red cableada evidenciaron continuidad del procesamiento. La validación visual del funcionamiento sin Internet requiere que ambos nodos mantengan el Wi-Fi desactivado y que el switch permanezca sin conexión al router. Los archivos CSV de rendimiento no permiten verificar por sí solos estas condiciones, por lo que dicha validación se considera pendiente.

La secuencia de la demostración final consiste en mostrar topología y direcciones, evidenciar aislamiento externo, ejecutar detecciones de productos y observar CPU/RAM/red. Posteriormente se desconecta únicamente Pavilion para comprobar la continuidad de resultados del master. Al restablecer el enlace, se observa la recuperación de las tareas remotas y de la telemetría sin intervención manual.

El panel usa recursos web incluidos en el script y no necesita CDN. El modelo debe permanecer en `yololln_openvino_model/` de ambos equipos; los inventarios de dependencias no contienen sus binarios. La instalación de paquetes requiere Internet durante preparación o un repositorio local adicional si se exige reinstalación offline, requisito no expresado en el caso de uso.

8.2 Controles de exposición

- Red privada dedicada, sin puerta de enlace en la conexión cableada.
- Redis en loopback; acceso remoto a través del túnel SSH cifrado sobre Ethernet.
- Panel en loopback y agente enlazado solo a la IP privada de Pavilion.
- Serialización JSON; sin contraseñas ni claves privadas dentro del código versionado.
- Demostración con objetos y resguardo de imágenes para evitar exposición innecesaria de datos personales.

El monitor es una herramienta del laboratorio, no un servicio público autenticado. La supervisión del túnel utiliza `systemd` y autenticación SSH por clave. Los keepalives detectan desconexiones; el reinicio del cliente corresponde a `systemd`. La clave privada se conserva fuera del código y del paquete documental.

8.3 Alcance de disponibilidad y estándares

La arquitectura presenta puntos únicos de fallo en el master, el coordinador, Redis y el switch. La presencia de dos consumidores de tareas no constituye redundancia física de red ni alta disponibilidad integral. La evaluación desarrollada se limita a la continuidad del procesamiento ante la ausencia del worker remoto.

El direccionamiento utiliza un bloque privado definido en RFC 1918 y la interconexión se basa en Ethernet de la familia IEEE 802.3. La implementación corresponde a una red privada de laboratorio y no incluye la prestación comercial de conectividad ni una evaluación del espectro radioeléctrico. No se realizó un análisis jurídico específico de normativa guatemalteca o de disposiciones de la SIT, por lo que no se establece una conclusión de cumplimiento en ese ámbito.

9. Reproducibilidad, arranque y evidencias

9.1 Orden de puesta en marcha

1. Se encienden ambos nodos y el switch, y se comprueban las direcciones IP fijas y la conectividad Ethernet.
2. Se verifica Redis en ProBook mediante `redis-cli ping` y la disponibilidad del servicio SSH en Pavilion.
3. Se comprueba el servicio del túnel en ProBook y el servicio de Celery en Pavilion, que consume la cola `cluster`.
4. Se inicia el worker local en la misma cola y se verifica el mensaje `ready` de ambos procesos.
5. Se verifica el servicio del agente de monitoreo en Pavilion y se inicia el panel en ProBook.
6. Se ejecuta el coordinador en otra terminal local y se comprueba la recepción de resultados de ambos nodos.
7. Al finalizar, se cierra la cámara mediante `q`; los procesos manuales y los servicios se detienen de forma ordenada según el README.

Comprobación del túnel supervisado en ProBook:

```
systemctl status arq2-tunnel.service --no-pager
```

Comando del coordinador de carga:

```
QT_QPA_PLATFORM=xcb python -m src.coordinator \
  --queue cluster --timeout 30 --max-pending 2
```

El README del repositorio contiene los comandos completos por terminal y los pasos de recuperación. Los entornos `.venv` se crean de forma independiente; no se copian entre sistemas. Los inventarios `requirements-master.txt` y `requirements-worker.txt` contienen 64 y 56 entradas respectivamente; los nodos tienen responsabilidades distintas. Los archivos `config/requirements-*.txt`, `config/python-*.txt` y `config/model-sha256.txt` documentan dependencias e identifican el modelo, pero no garantizan por sí solos su reinstalación offline.

9.2 Evidencias experimentales conservadas

Archivo	Contenido
<code>cluster_20261016_121746.csv</code>	Ensayo de ausencia y retorno de resultados del worker.
<code>cluster_20261016_142136.csv</code>	Inferencias de la prueba de carga monitorizada.
<code>monitor_20261016_141955_915224.csv</code>	Recursos y disponibilidad observada de ambos agentes.
<code>monitor_eventos_20261016_141955_915224.csv</code>	Marcas Inicio/Fin de la prueba de carga.
<code>prueba-continuidad-20261016.md</code>	Método, resultados y límites del ensayo de continuidad.
<code>prueba-carga-monitorizada-20261016.md</code>	Cruce de inferencias y telemetría en la misma ventana.

10. Conclusiones

1. Se construyó un sistema distribuido que procesa fotogramas en las CPUs de ambas laptops mediante una red Ethernet privada y cola compartida. Los registros identifican a los dos nodos como ejecutores de tareas.
2. La prueba de carga marcada produjo 18.47 resultados/s durante 355.243 s, con CPU media de 21.04 % en master y 52.55 % en worker. La RAM media se mantuvo en 35.21 % y 27.77 %, respectivamente.
3. En el ensayo histórico, el master completó 790 inferencias durante 52.86 s sin resultados remotos. Posteriormente, con supervisión systemd, se comprobó recuperación automática de tareas y telemetría; el retorno de tareas se estimó en unos diez segundos en uno de los ensayos, sin constituir un tiempo garantizado.
4. El monitoreo local proporciona gráficas y CSV para relacionar recursos y rendimiento. Los datos no permiten concluir ahorro energético, precisión de detección, aceleración frente a un nodo único ni disponibilidad de componentes no redundantes.
5. La operación offline es compatible con la arquitectura porque la inferencia, el broker, el transporte y la visualización son locales. Debe acompañarse de evidencia visual final del aislamiento y de las detecciones de productos.

Referencias

1. Enunciado de la asignatura. *Proyecto Final arquí 2.pdf*, exportación de Canvas aportada por el autor, 16 de octubre de 2026. Fuente primaria de requisitos y rúbrica.
 2. Celery Project. *Configuration and defaults* y *Tasks*. <https://docs.celeryq.dev/en/stable/userguide/configuration.html> y <https://docs.celeryq.dev/en/stable/userguide/tasks.html>
 3. Celery Project. *Using Redis*; limitaciones de tiempo de visibilidad. <https://docs.celeryq.dev/en/stable/getting-started/backends-and-brokers/redis.html>
 4. Ultralytics. *Model Export with Ultralytics YOLO* y documentación de inferencia. <https://docs.ultralytics.com/modes/export/> y <https://docs.ultralytics.com/modes/predict/>
 5. Intel. *OpenVINO Documentation*. <https://docs.openvino.ai/>
 6. psutil. *Documentation*. <https://psutil.readthedocs.io/>
 7. Rekhter, Y. et al. *RFC 1918: Address Allocation for Private Internets*. IETF, 1996. <https://www.rfc-editor.org/rfc/rfc1918>
 8. IEEE. *IEEE 802.3 Ethernet Working Group*. <https://www.ieee802.org/3/>
- Las mediciones de este informe proceden de los CSV originales del proyecto.

Anexo A. Evidencia del montaje físico

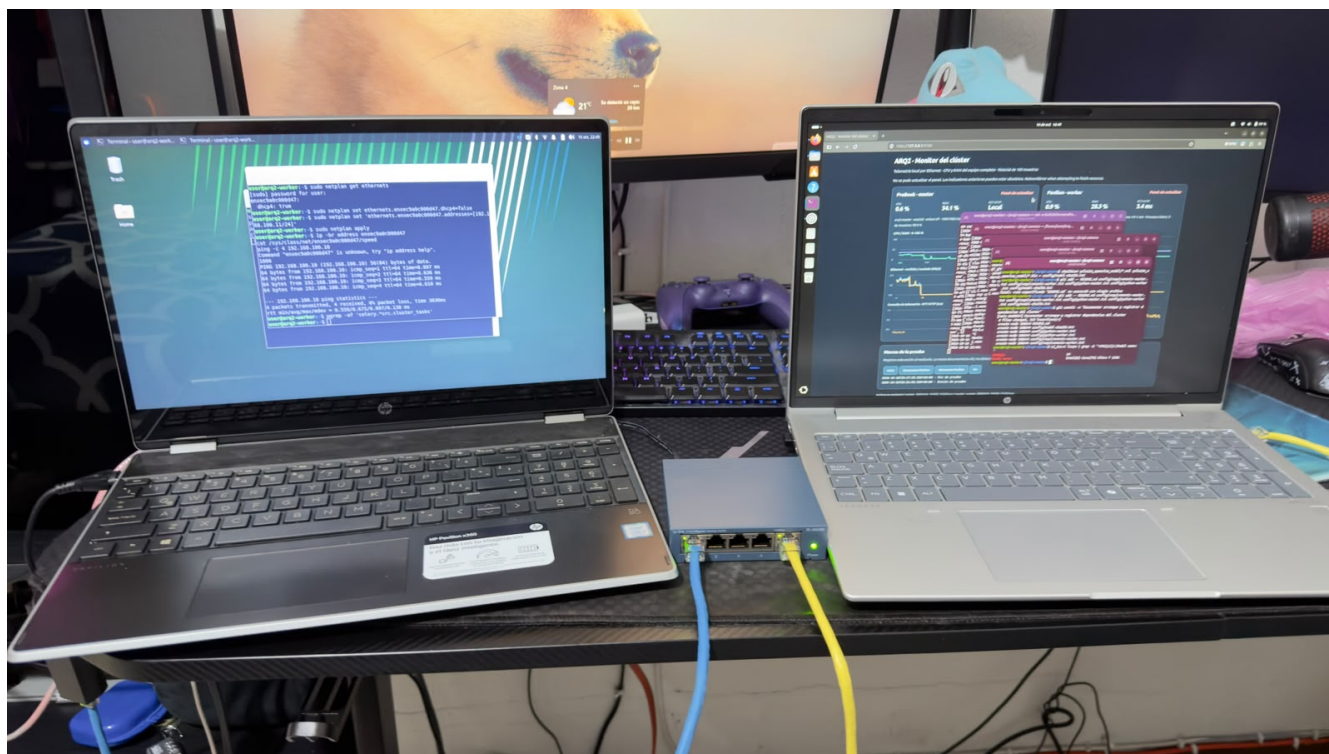


Figura 4. Montaje físico del clúster: Pavilion a la izquierda, ProBook a la derecha y switch entre ambos equipos. Fuente: elaboración propia.

La fotografía documenta los dos nodos y su interconexión mediante el switch. El equipo negro corresponde a Pavilion y el plateado a ProBook. El switch utilizado es un TP-Link TL-SG105 de cinco puertos Gigabit, identificado mediante su empaque durante la preparación del montaje.

El inventario del montaje registra un cable Cat 5 hacia ProBook y un cable Cat 6 hacia Pavilion. La velocidad negociada de 1000 Mb/s se verificó mediante las interfaces de red y el monitor. Este resultado corresponde a la instalación evaluada y no constituye una certificación del cableado.

La imagen permite identificar la disposición física de los equipos y las conexiones del montaje evaluado.

Anexo B. Detecciones y actividad de ambos workers



Figura 5. Detecciones de taza y persona y registros de ejecución de los workers remoto y local. Fuente: elaboración propia.

La ventana de cámara muestra una taza con etiqueta **cup** y confianza 0.71, y una persona con etiqueta **person** y confianza 0.61. Las terminales inferiores muestran tareas completadas por **arq2-worker** y **arq2-master**, lo que complementa la evidencia experimental de participación de ambos equipos.

El fotograma mostrado corresponde a **arq2-master**: inferencia 17.8 ms, RTT 33.5 ms y FPS indicado por la aplicación de 20.9. Son valores instantáneos de esta captura, no promedios de la prueba de carga analizada en el informe.

La detección de una taza es una evidencia cualitativa de funcionamiento con un objeto. El valor de confianza no representa una precisión del 71 % ni sustituye una evaluación sobre un conjunto etiquetado.

Anexo C. Monitor durante procesamiento

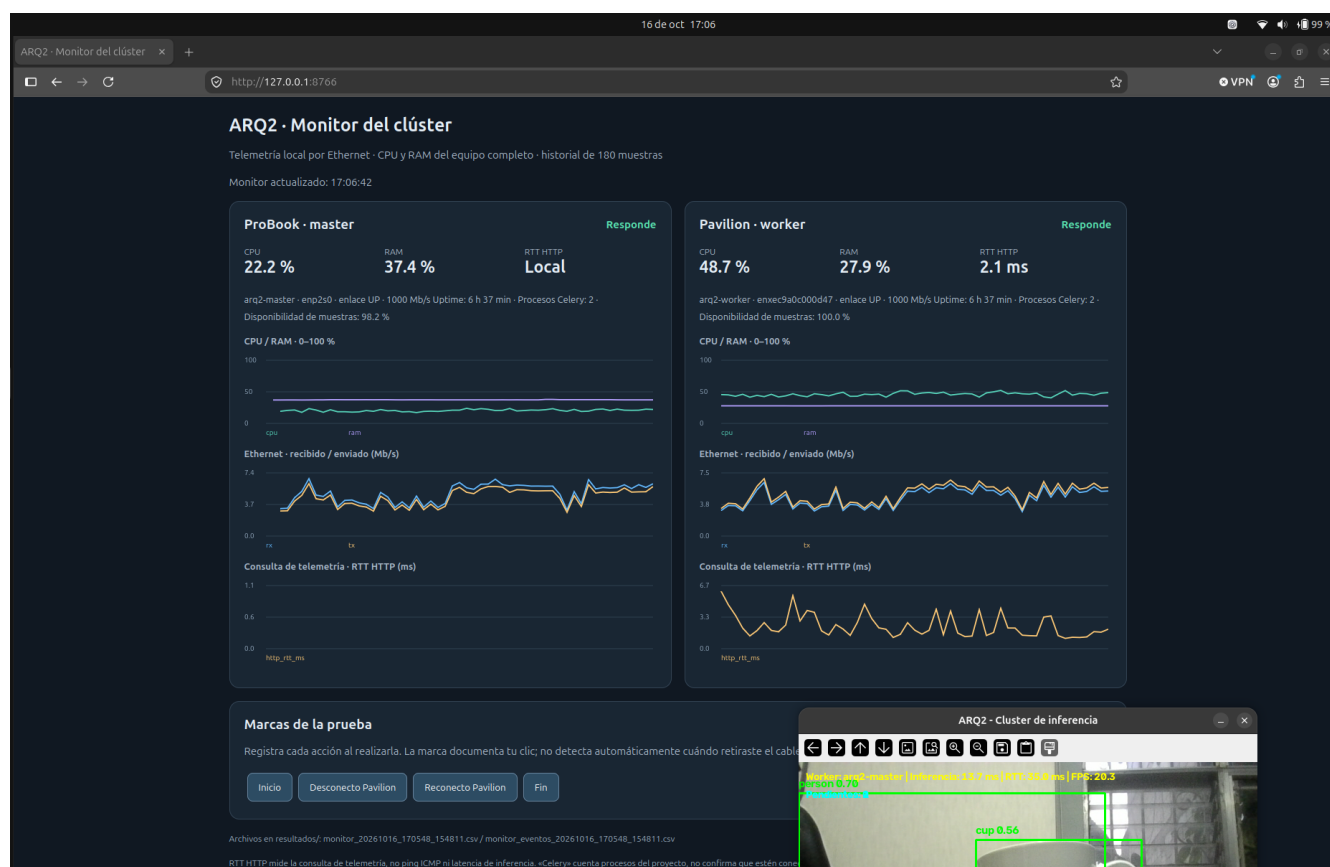


Figura 6. Panel local con ambos agentes en estado Responde y la cámara de detección activa. Fuente: elaboración propia.

El panel actualizado a las 17:06:42 muestra CPU de 22.2 % y RAM de 37.4 % en ProBook, y CPU de 48.7 % y RAM de 27.9 % en Pavilion. Ambos enlaces reportan 1000 Mb/s. El RTT HTTP del agente remoto es 2.1 ms; este valor no es la latencia de inferencia.

La captura pertenece a la sesión de monitoreo identificada como `monitor_20261016_170548_154811.csv`, diferente del ensayo formal de las 14:21–14:27 analizado en el cuerpo del informe. Sus valores instantáneos no reemplazan las medias ni percentiles calculados a partir de los CSV originales conservados.

La captura documenta la consulta simultánea de los recursos de ambos nodos durante el procesamiento de imágenes y complementa los registros experimentales del sistema.

Anexo D. Video de presentación y demostración

Como complemento audiovisual del informe, se incorpora el enlace al video de presentación del proyecto, publicado en YouTube por el autor.

Proyecto: Diseño, montaje e implementación de un clúster de Edge Computing para inferencia de IA local autónoma.

Autor: Christopher José Hernández Gramajo.

Carné: 0900-23-8598.

Enlace al video: <https://youtu.be/gdchqYcIsXI>

Página del proyecto en la feria

Consulta la ficha pública de NexoVisión y su presentación en la feria:

Enlace: <https://feria.hesm99.com/proyectos/36291fe5-557b-489e-a251-bba6d477e5ed>