

PROYECTO FINAL INTEGRADOR

EDGEVISION

Inteligencia artificial local sobre un clúster heterogéneo

ORANGE PI 4 PRO + UBUNTU VM | YOLO11-NANO + QWEN3-VL + RPC

4 integrantes • 4 sectores • demostración física + evidencia experimental

Una empresa que no quiera pagar IA por tokens necesita otra ruta

Solución: inferencia dentro de una red privada, sin enviar fotografías a APIs externas.

El problema

Dependencia de nube; consumo por token; acceso a Internet; transferencia de datos a terceros; menos control operativo.

La propuesta

Una Orange Pi ARM64 y Ubuntu x86-64 procesan video, fotografías y texto dentro de un laboratorio local.

La evidencia

Red privada operativa; YOLO local; Qwen3-VL interpretando imágenes; asignación RPC demostrada con Qwen2.5.

Mensaje del catedrático: el proyecto resuelve una necesidad empresarial, no simplemente "montar un clúster".

Cuatro sectores, una demostración completa

Cada integrante explica una capa distinta del sistema y conoce el recorrido extremo a extremo.

01

Contexto y hardware

2:00 min

02

Sistemas, virtualización y red

2:00 min

03

IA, cuantización y RPC

3:00 min

04

YOLO, portal, resultados y continuidad

3:00 min

Problema empresarial, Edge Computing y hardware

El porqué y el con qué

Definir la necesidad, justificar el enfoque local y presentar el montaje físico.

Edge Computing no es una marca: es una decisión de arquitectura

Acercamos el cómputo al origen del dato; los modelos y los servicios quedan bajo control del equipo.



Sin API externa para la inferencia $\Rightarrow$ sin costos, mantenimiento o limitaciones de rendimiento.

El laboratorio utiliza hardware de bajo consumo y una VM

Switch de 5 puertos, dos cables CAT6 y webcam USB conectada directamente al nodo ARM64.

Orange Pi 4 Pro

ARM64 · 8 núcleos
4 GB RAM

Worker RPC y YOLO

Laptop Win11

16 GB RAM · 6 núcleos

Anfitrión VirtualBox

Ubuntu Server

3 vCPU asignadas
x86-64

Portal, VLM y
supervisor

Conectividad

Switch Gigabit
Cables CAT6 + webcam

LAN privada y
captura

Por qué combinar ARM64 con x86-64

Dos arquitecturas distintas: un clúster experimental heterogéneo para observación de inferencia.

ARM64 · Orange Pi

Aproxima la IA al sensor. Ejecuta YOLO en CPU y ofrece un dispositivo remoto por RPC a través de Ethernet.

x86-64 · Ubuntu

Aloja los modelos GGUF, el portal, llama-server y el mecanismo de supervisión / respaldo.

Interconexión · Gigabit

RPC transmite trabajo y resultados por TCP. La red introduce latencia: dos CPUs no garantizan mayor velocidad.

Diferencia clave: colaboración lógica de cálculo ≠ fusión física de procesadores.

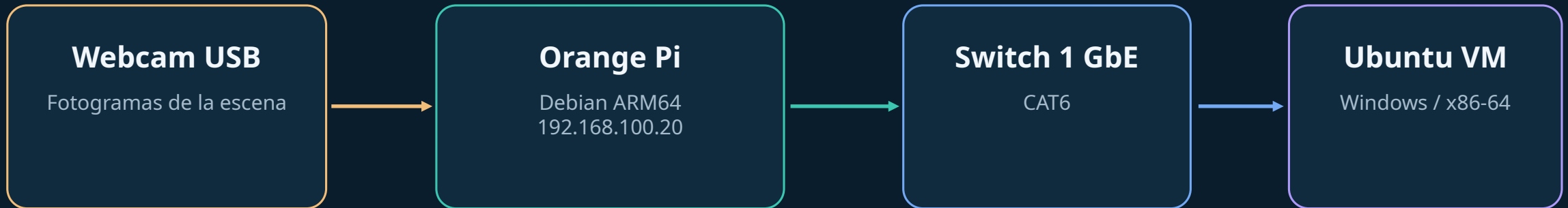
Virtualización, Linux y red del clúster

Cómo está montado

Demostrar direccionamiento IP, puertos, SSH, conectividad y arranque de servicios.

Montaje físico: origen → Worker → Master

La webcam está en la Orange Pi; Windows hospeda la VM y consulta el portal local.



Puente Realtek + subred privada 192.168.100.0/24 + puertos exclusivos por servicio.

Dos direcciones privadas, servicios independientes

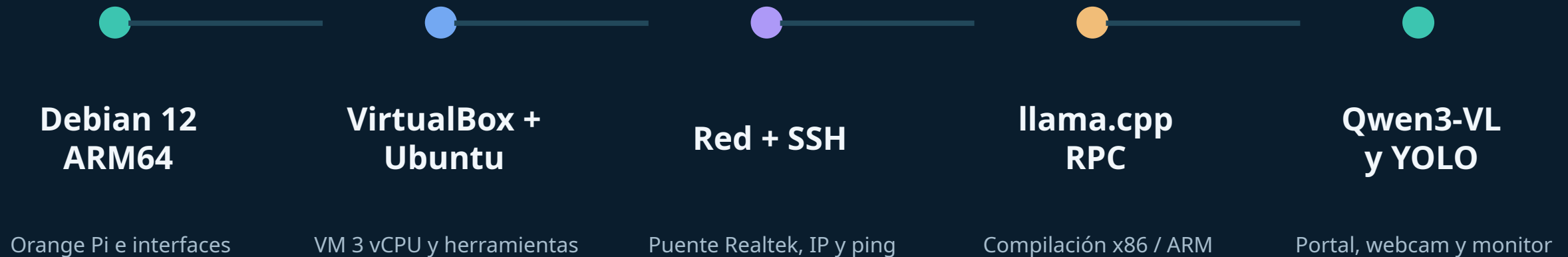
El cómputo utiliza 192.168.100.0/24. El NAT es de administración y descargas.

Equipo / servicio	IP o interfaz	Puerto	Responsabilidad
Ubuntu · Master	enp0s3 · 192.168.100.10	8000	Portal web
Ubuntu · VLM	127.0.0.1	8081	llama-server Qwen3-VL
Orange Pi · RPC	end0 · 192.168.100.20	50052	Worker de ggml-rpc-server
Orange Pi · YOLO	192.168.100.20	9200	Webcam y stream MJPEG
Orange Pi · métricas	192.168.100.20	9100	Uso de CPU y memoria
Windows → Ubuntu	127.0.0.1	18000	Redirección NAT al portal

SSH: puerto 22. No exponer RPC, cámara ni portal a una WAN pública.

Hitos reales de instalación y recuperación

Cada incidencia permitió validar una capa de la arquitectura.



Incidencias resueltas: puente Wi-Fi incorrecto, IP perdida, sintaxis CMake, hash ONNX y dimensiones 480x640.

La red dejó de ser una hipótesis: respondió el clúster

Ubuntu llegó al Worker y consultó sus servicios de IA.

PING LAN

4 / 4

Paquetes recibidos · 0% perdidos

RPC TCP

50052

Conexión satisfactoria

WEBCAM

/dev/video0

YOLO informa online=true

Comprobaciones reproducibles

```
ping -c 4 192.168.100.20  
nc -vz 192.168.100.20 50052  
curl http://192.168.100.20:9200/status
```

Inteligencia artificial local + RPC distribuido

La parte técnica principal

Modelos GGUF, cuantización, CPU local/remota, evidencias y latencia.

Tres modelos / componentes; cada uno resuelve algo distinto

LLM de prueba, VLM multimodal y detector de objetos se ejecutan con motores diferentes.

Qwen2.5 · 0.5B

Texto local y benchmark

GGUF · Q4_K_M

Prueba RPC inicial

Qwen3-VL · 2B

Interpretación de fotografías

GGUF · Q4_K_M + mmpoj Q8

VLM del portal

YOLO11-Nano

Detección con webcam

ONNX Runtime CPU

Boxes + clases + conf.

Del píxel a la respuesta en lenguaje natural

No es solo reconocimiento de clases: Qwen3-VL responde una instrucción textual sobre la fotografía.



La percepción visual se convierte en representaciones que el modelo de lenguaje utiliza para generar tokens de respuesta.

Cuantización Q4: ejecutar IA con memoria limitada

Menos bytes por peso, menor uso de RAM y posible pérdida de fidelidad frente a representaciones de mayor precisión.

FP16



16 bits por peso (referencia conceptual)

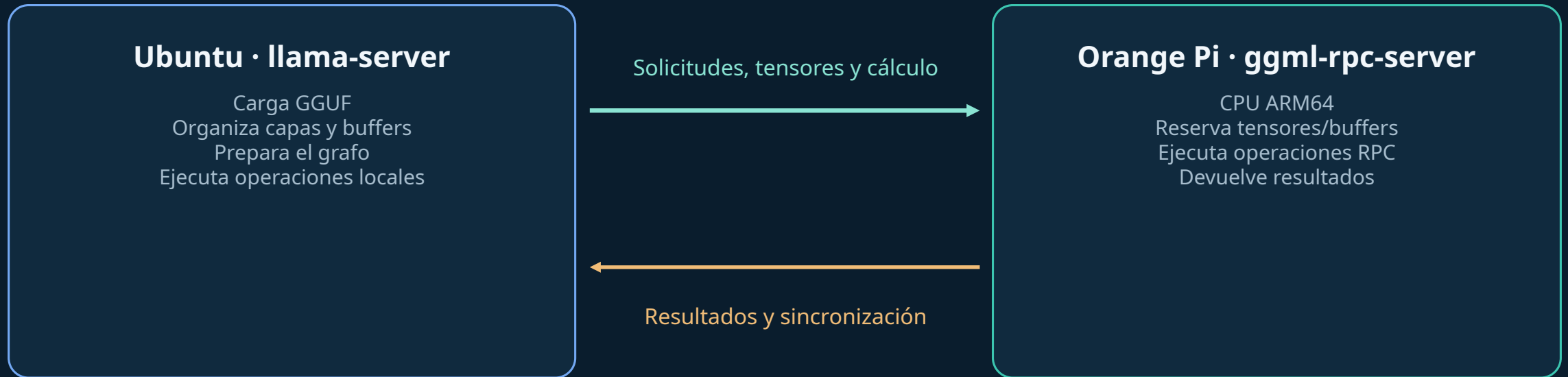
Q4



- Menor tamaño de pesos GGUF
- Ejecutable en CPU sin GPU dedicada
- Se conserva mmproj para la entrada visual
- Contexto y buffers siguen ocupando RAM
- La calidad debe medirse con imágenes reales

Un mismo motor coordina cálculo sobre dos CPUs

La llamada remota mueve operaciones y tensores; NO fusiona procesadores ni elimina el costo de red.



Protocolo RPC en TCP 50052 • Ethernet Gigabit • CPU local + dispositivo RPC0

Qué ocurre cuando habilitamos `--rpc ... -ngl 8`

Cinco etapas verificables: disponibilidad, asignación, buffers, operación y respuesta.

- 1 Conectar** llama.cpp descubre RPC0 en 192.168.100.20:50052
- 2 Asignar** Se decide qué capas o tensores permanecen remotos
- 3 Reservar** Memoria y buffers de cálculo locales/remotos
- 4 Calcular** Grafo GGML alterna cálculo y transferencias TCP
- 5 Responder** El servidor genera tokens y entrega JSON al portal

-ngl 8 = intento de descargar capas; los logs demuestran la asignación efectiva.

Qwen2.5 sí mostró asignación remota `RPC0`

Captura real del laboratorio con capas y buffers asignados al dispositivo remoto.

```
0.00.480.798 D llama_kv_cache: layer 8: dev = CPU
0.00.480.799 D llama_kv_cache: layer 9: dev = CPU
0.00.480.801 D llama_kv_cache: layer 10: dev = CPU
0.00.480.802 D llama_kv_cache: layer 11: dev = CPU
0.00.480.803 D llama_kv_cache: layer 12: dev = CPU
0.00.480.803 D llama_kv_cache: layer 13: dev = CPU
0.00.480.805 D llama_kv_cache: layer 14: dev = CPU
0.00.480.806 D llama_kv_cache: layer 15: dev = CPU
0.00.480.807 D llama_kv_cache: layer 16: dev = CPU
0.00.480.810 D llama_kv_cache: layer 17: dev = RPC0
0.00.480.813 D llama_kv_cache: layer 18: dev = RPC0
0.00.480.814 D llama_kv_cache: layer 19: dev = RPC0
0.00.480.816 D llama_kv_cache: layer 20: dev = RPC0
0.00.480.817 D llama_kv_cache: layer 21: dev = RPC0
0.00.480.818 D llama_kv_cache: layer 22: dev = RPC0
0.00.480.819 D llama_kv_cache: layer 23: dev = RPC0
0.00.480.822 I llama_kv_cache: CPU KV buffer size = 0,00 MiB
0.00.480.824 I llama_kv_cache: RPC0[192.168.100.20:50052] KV buffer size = 0,00 MiB
0.00.480.829 I llama_kv_cache: size = 12,00 MiB ( 1024 cells, 24 layers, 1/1 seqs), K (f16): 6,00
iB, V (f16): 6,00 MiB
0.00.544.378 I sched_reserve: RPC0[192.168.100.20:50052] compute buffer size = 33,01 MiB
0.00.544.379 I sched_reserve: CPU compute buffer size = 38,27 MiB
0.00.550.756 I common_memory_breakdown_print: | memory breakdown [MiB] | total free self m
del context compute unaccounted |
0.00.550.846 I common_memory_breakdown_print: | - RPC0 (192.168.100.20:50052) | 3844 = 3844 + ( 241 =
205 + 3 + 33) + -241 |
0.00.650.349 I llama_model_loader: - kv 13: qwen2.attention.layer_norm_rms_epsilon f32 =
0,000001
0.00.728.025 I llama_prepare_model_devices: using device RPC0 (192.168.100.20:50052) (unknown id) - 3844 M
B free
0.01.106.819 I print_info: n_layer = 24
0.01.106.820 I print_info: n_layer_all = 24
```

Prueba técnica

- Capas 17-23: RPC0
- Buffer RPC: 33.01 MiB
- Buffer CPU: 38.27 MiB
- Worker: 192.168.100.20:50052
- Modelo: Qwen2.5 0.5B

El procesamiento distribuido no siempre acelera

Medición real con un modelo pequeño: el costo de comunicación se hizo visible.

Solo Ubuntu



10.4 tokens/s

Ubuntu + RPC



8.6 tokens/s

~17% menos velocidad en la prueba RPC: intercambio de datos + sincronización + CPU ARM64.

Separar configuración distribuida de evidencia completada

La VLM puede arrancar con RPC, pero el watchdog volvió a `Solo VM` en las pruebas aportadas.



Regla académica: no etiquetar una respuesta local como multimodal distribuida.

YOLO, portal, evidencia visual y continuidad

Del hardware al usuario

El sistema recibe video, detecta objetos, captura una imagen y la interpreta con un VLM.

YOLO11-Nano se ejecuta directamente en la Orange Pi

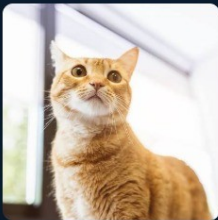
Webcam USB → OpenCV → ONNX Runtime → recuadros, clase y confianza.



Un portal web ofrece dos modos de visión en la misma red

Imagen - Qwen3-VL | Cámara en vivo - YOLO-Nano | Monitoreo de CPU y RAM.

Analizar fotografía



prueba.jpg

Instrucción para el modelo multimodal

Describe detalladamente qué aparece en esta imagen, sus objetos principales y cualquier texto visible. Responde en español, sin inventar información.

Analizar imagen localmente

Análisis completado. Guarda una captura para la evidencia.

Resultado de la inferencia

La imagen muestra un gato de pelaje de color naranja con manchas más oscuras en su cuerpo. El gato tiene una expresión de sorpresa o curiosidad, con los ojos grandes y muy abiertos, y su cabeza está ligeramente levantada. Se puede observar que tiene una nariz rosada y largas barbas blancas. El gato está en primer plano, y el fondo es muy claro y desenfocado, lo que sugiere que está en una habitación con luz natural. No hay texto visible en la imagen.

Respaldo CPU local - 245.38 s - Foto 39d7577a4f565abc

Monitoreo de CPU y RAM

CPU Ubuntu VM 2.5% RAM 21.2%	CPU Orange Pi 6.4% RAM 8.6%
--	---

El respaldo VLM funciona en frío: si RPC falla, Qwen3-VL puede recargarse en Ubuntu y la petición se reinicia. No garantiza continuidad instantánea. Las métricas de CPU por sí solas no demuestran distribución: conserva los registros de asignación RPC.

El portal, RPC y la webcam no tienen autenticación. Deben estar accesibles solo dentro del laboratorio; nunca abras puertos 50052, 9200 ni 8000 hacia Internet.

- Subir imagen JPG/PNG
- Escribir una pregunta
- Ver respuesta del VLM
- Cámara + detecciones
- Capturar fotograma
- Estado del clúster

Una fotografía real se analizó sin una API de IA externa

El VLM identificó el gato naranja y generó texto en español.

Imagen · Qwen3-VL

Cámara en vivo · YOLO-Nano

Analizar fotografía



prueba.jpg

Instrucción para el modelo multimodal

Describe los objetos y animales visibles en esta imagen en un máximo de tres frases. Indica sus características principales, sin inventar información.

Analizar imagen localmente

Análisis completado. Guarda una captura para la evidencia.

Resultado de la inferencia

La imagen muestra un gato de pelaje de color naranja con manchas más oscuras, que tiene una expresión de sorpresa o curiosidad. Su pelaje es de textura suave y liso, y su cabeza está ligeramente inclinada hacia arriba. El gato tiene grandes ojos de color amarillo claro y orejas erguidas, lo que le da una apariencia alerta y atenta.

Respaldo CPU local · 240.8 s · Foto 39d7577a4f565abc

Estado del clúster

Orange Pi 4 Pro

192.168.100.20 · RPC:50052

Disponible

Qwen3-VL multimodal

VM Ubuntu · servicio interno 8081

Disponible

Webcam USB / YOLO-Nano

Orange Pi · 192.168.100.20:9200

Disponible

Ruta de inferencia VLM

Worker desconectado (watchdog)

Solo VM

Recuperaciones del VLM

4

Reactivar RPC

Usar solo VM

Modo activo: rpc

Monitoreo de CPU y RAM

CPU Ubuntu VM

1.1%

RAM 19.1%

CPU Orange Pi

5.5%

RAM 8.9%

El respaldo VLM funciona en frío: si RPC falla, Qwen3-VL puede recargarse en Ubuntu y la petición se reinicia. No garantiza continuidad instantánea. Las métricas de CPU por sí solas no demuestran distribución: conserva los registros de asignación RPC.

El portal, RPC y la webcam no tienen autenticación. Deben estar accesibles solo dentro del laboratorio; nunca abras puertos 50052, 9200 ni 8000 hacia Internet.

Evidencia del laboratorio

- Imagen: gato naranja
- Respuesta en español
- Análisis completado
- Tiempo: 240.8 s
- Ruta: Solo VM

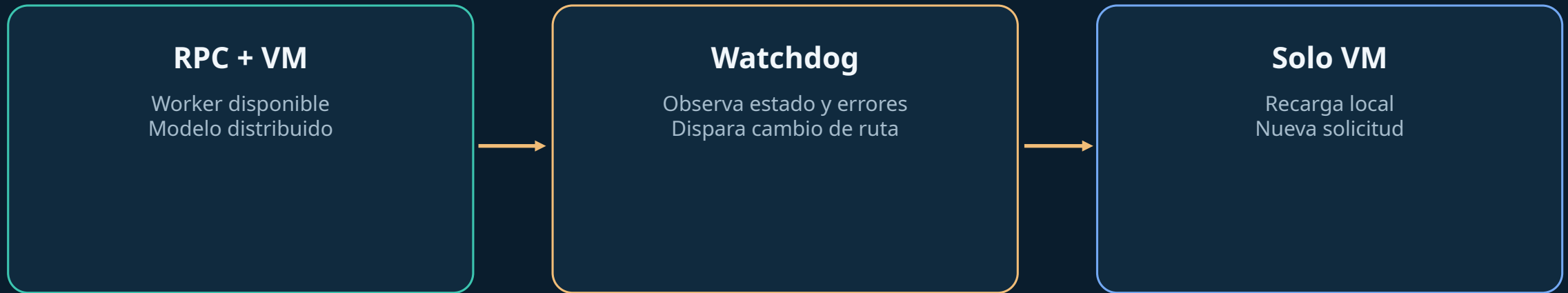
Resultado: funcional; no distribuido en esta prueba.

EDGEVISION | SECTOR 4

25

Continuidad lógica: watchdog y respaldo en frío

Se protege la posibilidad de ejecutar nuevas inferencias; no se promete continuidad sin interrupciones.



En el portal se registraron cuatro recuperaciones, pero falta probar de forma controlada la caída y una nueva respuesta.

Qué pudimos probar y cuál es la limitación

No confundir implementación, configuración y verificación de extremo a extremo.

Capacidad	Evidencia	Conclusión
LAN privada	Ping 4/4, TCP 50052	Operativa
YOLO en Orange Pi	/dev/video0, ONNX, 1.26 FPS	Operativa
Imagen a Qwen3-VL	Gato descrito, 245.38 s	Local validada
RPC Qwen2.5	Capas/buffer RPC0	Asignación probada
Qwen3-VL RPC	Reserva RPC0 + retorno a local	No validada E2E
HA y desconexión WAN	Pruebas reportadas por equipo	Falta acta/captura de control

Secuencia compacta para la clase y para grabar evidencia

Evitar depender de una inferencia en vivo de cuatro minutos: mostrar el proceso real grabado cuando sea necesario.

0:00-0:50

P1

Necesidad empresarial + hardware

0:50-1:45

P2

Diagrama LAN, IPs, ping y servicios

1:45-2:55

P3

Modelos, RPC0, benchmark 10.4 vs 8.6

2:55-5:00

P4

YOLO + captura + respuesta Qwen3-VL + conclusión

Los archivos y enlaces cuentan tanto como la demostración

El PDF asigna 10 puntos al informe escrito y 10 a defensa/coevaluación. La clase agrega entregables para feria.

- **Informe grupal PDF**

Archivo formal + código/evidencias

- **4 videos individuales**

5-7 min cada integrante (PDF)

- **Video principal de feria**

Hasta ~10 min (explicado en clase)

- **Video de reflexión**

Aprendizajes, dificultades y resultados

- **Autoevaluación / pares**

Excel oficial de Canvas; notas reales 1-100

- **Entrega y enlaces**

Canvas por cada miembro; feria; permisos correctos

Responder exactamente al catedrático, sin inventar componentes

El foco oral fue problema real, modelo local, clúster, evidencia, defensa y participación.

Puntos que enfatiza el docente

Justificación empresarial; dos arquitecturas; caso visual; inferencia por CPU; claridad técnica; autoevaluación; pares; defensa grupal.

Advertencias documentales

La rúbrica escrita menciona pfSense, portal cautivo y normativa. En la clase se aclaró que pfSense no era obligatorio para este caso.

La redundancia física del switch NO está implementada; el failover del VLM es lógico.

Software funcional no equivale a entrega académica cerrada

Tareas de verificación final: documentar las pruebas y completar anexos antes de Canvas.

Confirmado

Qwen3-VL local, YOLO, red, portal y RPC Qwen2.5

Reportado

Reinicio y funcionamiento sin Internet: guardar evidencia

Pendiente

Qwen3-VL remoto E2E; monitor 2.0.5 en rollback

Por entregar

PDF final, videos, Excel y enlaces de feria

EDGEVISION CONECTA HARDWARE, RED E IA LOCAL

- Demostramos visión por computadora en ARM64.
- Interpretamos fotografías con un VLM local cuantizado.
- Verificamos comunicación RPC y asignación remota con Qwen2.5.
- Identificamos límites reales: memoria, latencia, calidad y supervisión.

Chequeo de red, servicios, inferencia y evidencia

Ejecutar en el nodo indicado; no pegar comandos destructivos durante la defensa.

UBUNTU

```
ip -4 -br addr  
ping -c 4 192.168.100.20  
ss -ltnp | grep -E "8000|8081"
```

ORANGE PI

```
ip -4 -br addr show end0  
systemctl status edgevision-rpc  
systemctl status edgevision-camera
```

RPC / YOLO

```
curl http://192.168.100.20:9200/status  
ss -ltnp | grep -E "50052|9200|9100"  
journalctl -u edgevision-rpc -n 30
```

EVIDENCIAS

```
tail -n 3 evidencias/inferencias.jsonl  
grep -Ei "RPC0|compute buffer|offload" \  
evidencias/modelo-vlm.log
```

Cuatro verificaciones permiten sostener la defensa técnica

Registrar hora, comando, estado anterior/posterior y captura vinculada al informe.

Prueba	Acción	Evidencia requerida
Arranque	Reiniciar ambos nodos	IPs, 50052, 9200 y 9100 estables
Sin Internet	Desconectar solamente WAN	YOLO y Qwen3-VL responden en LAN
Fallo worker	Detener RPC durante prueba controlada	Nueva inferencia local finalizada
RPC Qwen3-VL	Activar RPC y analizar fotografía	Respuesta rpc-primary + log RPC0 asociado

Una configuración posible sin prueba no es equivalente a una inferencia confirmada.

Documentos y capturas que sustentan esta exposición

Evitar referencias genéricas y conservar la evidencia exacta de cada afirmación.

Documento oficial

proyecto final requerimientos.pdf

Fases, restricciones, tres casos de uso y rúbrica

Indicaciones orales

class.txt

Aclaraciones sobre pfSense, feria, videos y entrega

Bitácora técnica

Informe_Tecnico_Integral_EdgeVision_2026.docx

Comandos, cronología y matriz de evidencias

Registros reales

capturas EdgeVision + inferencias.jsonl

Tiempos, modos, pantallas y resultados RPC