ISBN: 978-628-96613-3-0. ISSN: 2414-6390. Digital Object Identifier: https://dx.doi.org/10.18687/LACCEI2026.1.1.2572

Biometric-Gated SD-VPN Gateway for Secure Remote PLC (Siemens S7-1200) Control with Microsegmentation and Conversational Monitoring

J. E. Huarancay Huaranga[1]; J. A. Quispe Ticllacuri[2]; M. C. Matute Villegas[3]
[1,2,3]Universidad Tecnológica del Perú, Lima, Perú, U19202016@utp.edu.pe, U18302407@utp.edu.pe, C24978@utp.edu.pe

Abstract*– In industrial automation, PLCs such as the Siemens S7-1200 are essential components for process management; however, the need for remote access to these devices introduces vulnerabilities such as unauthorized access. Although VPNs are used to protect the communication channel, their authentication is often limited to digitized passwords or manually granted permissions. In addition, industrial monitoring systems lack simple and interactive remote query mechanisms, limiting the ability to verify PLC status, access history, or the user who performed an operation. This study proposes the implementation of a secure remote control system based on a Raspberry Pi 5 acting as a gateway running ZeroTier to establish secure connectivity and microsegmentation to a Siemens S7-1200 PLC. Access is restricted through facial biometric authentication implemented with an anti-spoofing module, MediaPipe, and Teachable Machine. A conversational monitoring workflow developed in n8n is also integrated to query PLC status via WhatsApp. Validation included 240 biometric attempts (120 authorized and 120 unauthorized), achieving an accuracy of 95.42%, FAR of 0.00%, and FRR of 9.17% under a confidence threshold of 0.95. For PLC control, 50 Put/Get operations were performed, achieving 100% success, with an average end-to-end latency of 192.40 ms (p95 = 238.68 ms). These results demonstrate that the proposed architecture enables secure remote PLC access with biometric authentication, maintaining communication performance compatible with non-critical industrial environments.

Keywords*-- PLC, zeroTier, raspberry pi, facial biometrics, microsegmentation.

Gateway SD-VPN con control biométrico para el control remoto seguro de un PLC (Siemens S7-1200), con microsegmentación y monitoreo conversacional.

J. E. Huarancay Huaranga[1]; J. A. Quispe Ticllacuri[2]; M. C. Matute Villegas[3]
[1,2,3]Universidad Tecnológica del Perú, Lima, Perú, U19202016@utp.edu.pe, U18302407@utp.edu.pe , C24978@utp.edu.pe

Resumen–En la automatización industrial, los PLC como el Siemens S7-1200 son componentes esenciales en la gestión de procesos; sin embargo, la necesidad de acceso remoto a estos dispositivos se expone a vulnerabilidades como los accesos no autorizados. Aunque las VPN se emplean para proteger el canal de comunicación, su autenticación suele limitarse al uso de contraseñas digitalizadas o permisos manuales. Además, los sistemas de supervisión industrial carecen de mecanismos de consulta remota simples e interactivos, limitando la verificación del estado del PLC, el historial de acceso o el usuario que realizó la operación. Este estudio propone la implementación de un sistema de control remoto seguro basado en una Raspberry Pi 5 que actúa como Gateway ejecutando ZeroTier para establecer conectividad segura y microsegmentación hacia un PLC Siemens S7-1200. El acceso se restringe mediante una autenticación biométrica facial implementada con un módulo anti-spoofing, MediaPipe y Teachable Machine. Se integra además un flujo de monitoreo conversacional desarrollado en n8n para consultar el estado del PLC mediante WhatsApp. La validación incluyó 240 intentos biométricos (120 autorizados y 120 no autorizados), obteniéndose un accuracy de 95.42%, FAR de 0.00% y FRR de 9.17% bajo un umbral de confianza de 0.95. Para el control del PLC se realizaron 50 operaciones Put/Get, alcanzando 100% de éxito; con una latencia end-to-end promedio de 192.40 ms (p95 = 238.68 ms). Estos resultados evidencian que la arquitectura propuesta permite acceso remoto al PLC con autenticación biométrica, manteniendo un desempeño de comunicación compatible en entornos industriales no críticos.

Palabras clave-- PLC, zeroTier, raspberry pi, biometría facial, microsegmentación.

I. Introducción

La seguridad y confiabilidad de los Controladores Lógicos Programables (PLC) resultan esenciales para mantener la continuidad operativa en sectores críticos como energía, manufactura, minería y transporte. Sin embargo, estudios recientes indican que su diseño original se centró en la robustez del proceso más que en la protección frente a amenazas externas, lo que los convierte en objetivos frecuentes de accesos no autorizados cuando se exponen a redes abiertas o entornos de telemantenimiento [1]. Informes especializados en ciberseguridad industrial destacan que gran parte de las amenazas actuales se dirigen a PLC, DCS (Distributed Control System), SCADA (Supervisory Control And Data Acquisition) y HMI (Human Machine Interface) mediante accesos remotos inseguros, falta de segmentación y configuraciones deficientes, incrementando la vulnerabilidad operativa en entornos reales [2].

Diversos estudios han explorado alternativas para mejorar el acceso remoto seguro en sistemas industriales. Por ejemplo, algunas propuestas implementan arquitecturas SCADA basadas en Red Privada Virtual (VPN) móviles para operar estaciones de bombeo distribuidas, mostrando que es posible mantener estabilidad incluso con enlaces variables siempre que exista un canal cifrado confiable [3]. Otros trabajos analizan túneles VPN dedicados para supervisión industrial, demostrando que estos esquemas preservan la disponibilidad de los dispositivos y la integridad de los datos aun cuando el operador se conecta desde fuera de la red local del PLC [4]. Sin embargo, estos mismos estudios coinciden en que una proporción significativa de incidentes en OT (Operational Technology) ocurre debido a configuraciones erróneas del acceso remoto o al uso de VPN que no verifican adecuadamente la identidad del operador, evidenciando que el cifrado por sí solo no es suficiente [5].

En respuesta a estas limitaciones, han surgido tecnologías más flexibles como ZeroTier, que permite crear redes privadas virtuales definidas por software y puede integrarse con firewalls como OPNsense para proteger entornos IoT e industriales con menor complejidad de despliegue [6]. Asimismo, comparaciones entre OpenVPN, WireGuard e IPSec analizan su rendimiento en términos de latencia, carga computacional y nivel de seguridad, proporcionando criterios para seleccionar la solución más adecuada en aplicaciones industriales donde la disponibilidad es crítica [7].

Paralelamente, la visión por computador ha permitido que técnicas biométricas sean más accesibles gracias a plataformas de bajo costo. Google Teachable Machine ha sido aplicada exitosamente en tareas de clasificación visual en contextos agrícolas como el reconocimiento del estado del cacao [8] y del mango [9], demostrando su utilidad en entornos reales sin necesidad de programación especializada [10]. Por otro lado, MediaPipe se ha consolidado como un framework robusto para el análisis de landmarks faciales en tiempo real, con aplicaciones que incluyen monitoreo clínico [11], detección de movimientos anómalos [12] y control de interfaces mediante gestos [13]. Investigaciones recientes también han evaluado su uso para analizar la atención en entornos educativos mediante procesamiento multimodal de baja latencia [14]. De manera complementaria, las plataformas de automatización como n8n (nodemation) permiten coordinar procesos, registrar eventos y enlazar módulos inteligentes en arquitecturas distribuidas.

Algunos estudios demuestran su capacidad para gestionar flujos complejos y mantener trazabilidad de operaciones críticas mediante nodos configurables [15]. Otros trabajos resaltan su utilidad para integrar monitoreo y supervisión automatizada en sistemas IoT (Internet of Things) y de control [16].

En ese sentido, esta investigación propone una arquitectura que integra autenticación facial mediante Teachable Machine y MediaPipe, una Raspberry Pi 5 como pasarela de microsegmentación y un túnel ZeroTier para asegurar un canal cifrado punto a punto con el PLC Siemens S7-1200. La autenticación se realiza en la laptop del operador y la Raspberry Pi reencamina el tráfico industrial hacia el PLC a través de su interfaz Ethernet. La comunicación se establece mediante python-snap7 con configuraciones generadas en TIA Portal [17], mientras que la microsegmentación garantiza que solo el tráfico autorizado alcance al controlador [18] y ZeroTier mantiene un enlace privado entre el operador y la puerta de enlace (Gateway) [19]. Además, n8n registra quién accede al sistema, asocia cada acceso con el usuario autenticado y genera un registro temporal de las sesiones, siguiendo los mecanismos de orquestación y monitoreo automatizado descritos en la literatura [15]. Con esta integración, el sistema asegura el tráfico, verifica la identidad del operador y reduce significativamente el riesgo de accesos indebidos en entornos industriales [1], [12], [13].

Finalmente, la validación del sistema se realiza mediante pruebas de latencia, autenticación biométrica y monitoreo conversacional, evaluando su comportamiento tanto en escenarios locales como remotos. Este análisis permite determinar cómo la combinación de VPN definida por software, biometría facial y microsegmentación mejora la seguridad del acceso remoto al PLC sin comprometer la disponibilidad ni el rendimiento de la red. Adicionalmente, para delimitar el alcance de la propuesta, se define el siguiente modelo de amenaza. Se considera un atacante con acceso a Internet capaz de: intentar acceso no autorizado por credenciales, incorporarse indebidamente al dominio de la red VPN, ejecutar ataques de repetición (replay) sobre tokens HTTP de autorización, realizar spoofing biométrico mediante imágenes, y efectuar movimiento lateral hacia activos OT una vez dentro del dominio de red. A partir del análisis del estado del arte, se identifica una brecha científica: los trabajos revisados se centran en el cifrado del canal VPN o en el rendimiento de comunicación, pero no integran explícitamente separación entre autenticación biométrica del operador y autorización dinámica de red, ni implementan microsegmentación condicionada a validación de identidad con trazabilidad operativa automatizada. En este contexto, los aportes principales del presente trabajo son: diseño de un Gateway biométrico que separa autenticación lógica de autorización de red; implementación de microsegmentación dinámica hacia el PLC condicionada a verificación facial válida; e incorporación de trazabilidad operativa mediante monitoreo conversacional automatizado.

II. Metodología

El objetivo principal es condicionar toda operación remota sobre un PLC a dos salvaguardas: verificación biométrica previa de la identidad del operador y transporte cifrado mediante una VPN superpuesta que aísla el dominio de control del resto de la red [1]. El alcance comprende la configuración del PLC con un bit de permiso en un bloque de datos (DB), el cual habilita o bloquea la secuencia Ladder del semáforo; sin autorización válida, no se permite el inicio ni el cambio de estados [17]. La arquitectura considera la conectividad física PLC a una Raspberry Pi 5 en la red local industrial y la conectividad lógica laptop a la Raspberry Pi 5 mediante ZeroTier, permitiendo operación remota desde otra red manteniendo cifrado extremo a extremo [6]. La autenticación biométrica se implementa con Teachable Machine y MediaPipe, priorizando un enfoque de bajo costo, baja latencia y reproducibilidad en entornos académicos e industriales [10]. Una vez validado el usuario, Python y Snap7 modifican el valor booleano (bool) del DB en el PLC, habilitando temporalmente la secuencia de operación [18].

El proceso industrial donde se efectuó la validación se desarrolló sobre el módulo de automatización Daedalus PS1, mostrado en la Fig. 1, el cual integra un PLC Siemens S7-1200, una HMI y terminales de entrada/salida, permitiendo reproducir condiciones reales de operación y probar la efectividad del control remoto autorizado mediante biometría.

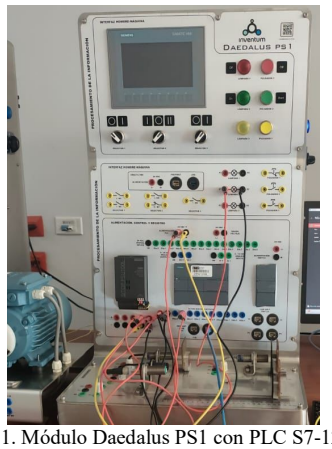

Fig. 1. Módulo Daedalus PS1 con PLC S7-1200.

A. *Entorno de hardware y software*

El entorno físico está conformado por un PLC Siemens S7-1200 con dirección IP (Internet Protocol) 192.168.0.1, Rack 0, Slot 1 y una Raspberry Pi 5 que opera como pasarela con eth0 dedicada a la LAN industrial y wlan0 a Internet [17]. La laptop del operador ejecuta Windows 11 y corre el script main.py en Visual Studio Code, integrando OpenCV para captura y preprocesamiento, TensorFlow/Keras para inferencia del modelo, MediaPipe para detección de

landmarks faciales y Python con Snap7 para la escritura del bool correspondiente al DB del PLC [13]. ZeroTier crea un dominio privado con un Network ID propio; su controlador aprueba o rechaza nodos antes de que intercambien tráfico, aportando un control de acceso ligero y facilitando el despliegue sobre redes con NAT (Network Address Translation) [6]. La selección de ZeroTier sobre alternativas como WireGuard u OpenVPN se fundamenta en su operación sin necesidad de abrir puertos en el router ni mantener infraestructura de servidor dedicado, su compatibilidad nativa con el Raspberry Pi 5 y su despliegue simplificado sobre redes. Estudios comparativos entre estas tecnologías indicaron que ZeroTier prioriza la facilidad de implementación y la operación transparente sobre NAT frente al rendimiento bruto, características determinantes para el caso de uso propuesto [7].

B. Flujo general de operación

Primero se habilita Acceso PUT/GET en el PLC y se crea un DB global con el bit de permiso (DB=1, byte=0, bit=0), desactivando el acceso optimizado para exponer offset compatibles con aplicaciones externas [17]. Posteriormente se configura la red local: el módem asignado a la IP 192.168.0.4, la Raspberry Pi en 192.168.0.8 mediante eth0, y el enlace físico PLC al Raspberry Pi 5 para el tránsito de control; wlan0 queda reservada exclusivamente a la VPN [7]. Luego se crea la red ZeroTier, se instalan los clientes en la laptop y en la Raspberry Pi, se aprueban nodos desde el controlador y se asignan IP gestionadas (Managed IPs), quedando así el canal cifrado y con control de membresía [6]. En biometría, Teachable Machine se usa para entrenar el modelo con las clases correspondientes; MediaPipe detecta rostro y landmarks en tiempo real y main.py determina la autorización. Si el resultado es válido, Python y Snap7 escribe el bool en el DB y la programación Ladder habilita la operación del semáforo [10]. Como mecanismo de endurecimiento, la pasarela gobierna el paso de tráfico ZeroTier al eth0 mediante un servicio HTTP minimalista que permanece cerrado por defecto y solo abre el puente tras una autorización biométrica válida; este servicio acepta comandos 1 (habilitar), 0 (bloquear) y q (estado seguro) [4]. El sistema integra un módulo de monitoreo en n8n, donde cada autenticación biométrica válida genera una notificación HTTP con el usuario, la hora y el estado del permiso. El operador puede consultar el estado del PLC mediante WhatsApp, y el agente en n8n recupera los registros desde Google Sheets, proporcionando trazabilidad y supervisión remota en tiempo real [15].

C. Plan de validación y métricas

La validación incluye precisión biométrica mediante matriz de confusión con datos no vistos, asegurando robustez ante variaciones de iluminación y pose [10]. Se mide la tasa de éxito de la comunicación Put/Get y los tiempos de ciclo PLC al Raspberry Pi 5 para evaluar la estabilidad del enlace industrial [18]. Se cuantifica además latencia y pérdidas con y sin VPN para determinar el impacto del túnel ZeroTier en el tiempo de respuesta y su conveniencia para acceso remoto a PLC [7]. El banco de pruebas empleado se muestra en la Fig. 2, donde se observa la laptop con el cliente biométrico y la Raspberry Pi 5 configurada como pasarela hacia el PLC.

Fig. 2. Banco de pruebas del sistema biométrico VPN PLC.

En la Tabla I se resume el protocolo de validación y los criterios de éxito para biometría, anti-spoofing, control PLC, red y monitoreo conversacional. Se definieron tamaños de muestra y condiciones de prueba (distancias, umbral y ventana temporal). Para el control PLC, el éxito se define por la lectura consistente del bit de permiso en el DB y la transición esperada en Ladder, en el cambio de la salida del semáforo (Q0.0). Asimismo, se establecieron métricas de desempeño (éxito Put/Get, latencia end-to-end y percentiles de RTT con pérdida y jitter) para asegurar reproducibilidad.

TABLA I
PROTOCOLO DE VALIDACIÓN EXPERIMENTAL Y CRITERIOS DE ÉXITO

Métrica	Cómo se calcula	Criterio de éxito
Accuracy	Accuracy = (TP+TN)/N	Accuracy Matriz de confusión
FAR / FRR	FAR = FP/(FP+TN) FRR = FN/(TP+FN)	FAR y FRR bajo umbral=0.95
Anti-spoofing	Spoof detectado = Bloq/Total Falso rechazo = Rech/Total real	% spoof detectado % falso rechazo
Control PLC	Éxito = OK/N Latencia por operación end-to-end	% éxito y latencia (prom/p95) en LAN/VPN
Red	RTT prom/p95/p99; jitter; pérdida (%)	RTT(i) − RTT(i-1)

III. ARQUITECTURA DEL SISTEMA

La arquitectura propuesta integra tres componentes principales: un cliente biométrico encargado de la autenticación facial, un túnel VPN que proporciona conexión en red, y una Raspberry Pi 5 configurada como Gateway de seguridad que gestiona el acceso hacia el PLC Siemens S7-1200. El funcionamiento interno del sistema se organiza en un flujo lógico que enlaza todos los módulos responsables del control remoto. La Fig. 3 ilustra este proceso, donde el cliente ejecuta la autenticación biométrica y envía la orden HTTP; ZeroTier valida la membresía del nodo y enruta el tráfico a través del dominio virtual; la Raspberry Pi 5 recibe la autorización y activa el puente entre sus interfaces inalámbrica wlan0 y cableada eth0; finalmente, el PLC actualiza el bit de permiso en el DB1, lo que condiciona la ejecución de la secuencia Ladder programada para el proceso.

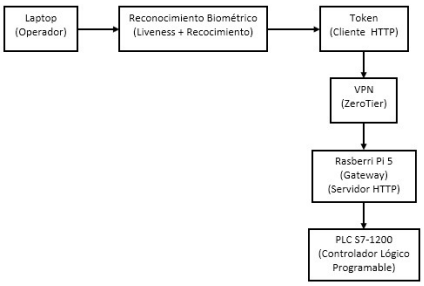

Fig. 3. Diagrama funcional del sistema (elaboración asistida por IA).

Como se muestra en la Fig. 4, el proceso de autorización comienza con el reconocimiento facial realizado localmente en la laptop del operador; posteriormente, el sistema utiliza la VPN para transportar de manera segura la solicitud hacia la pasarela, donde se habilita temporalmente el enlace hacia la red industrial.

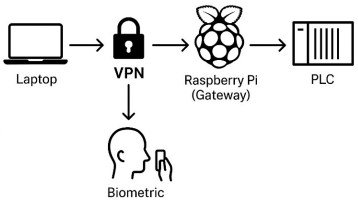

Fig. 4. Arquitectura general del Sistema (elaboración asistida por IA).

La infraestructura física se complementa con la pasarela implementada en la Raspberry Pi 5, que actúa como punto intermedio entre el túnel cifrado y la red industrial cableada [20]. Como se aprecia en la Fig. 5, esta configuración permite aislar el PLC del resto de la red y ejercer un control granular sobre el flujo de tráfico autorizado.

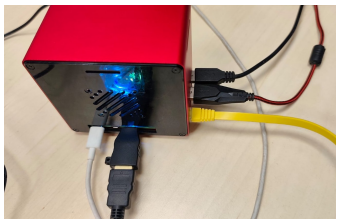

Fig. 5. Raspberry Pi 5 operando como Gateway entre VPN y red industrial.

IV. Sistema de Autenticación Biométrica

A. *Adquisición y preparación de los datos*

Se estructuró inicialmente el dataset utilizando Teachable Machine con múltiples etiquetas (“Jardel”, “Anthonny”, “Hombre”, “Mujer” y “ambiente”) con el fin de explorar capacidad de clasificación y robustez ante variaciones de iluminación, pose y expresión facial [10]. No obstante, para el sistema final de autenticación, las clases operativas se redefinieron bajo un esquema de verificación binaria: UsuarioAutorizado y NoAutorizado, alineado con buenas prácticas en sistemas biométricos. Se recolectaron aproximadamente 200 muestras por clase, asegurando balance y división en conjuntos de entrenamiento y validación. Durante la ejecución en tiempo real, el sistema evalúa la similitud bajo un umbral de confianza de 0.95; si la probabilidad estimada es inferior a dicho umbral, el intento se clasifica como no autorizado [8]. En términos de ética y privacidad, se obtuvo consentimiento informado del usuario cuyas muestras fueron empleadas en el entrenamiento, y el sistema no almacena imágenes posteriores a la inferencia, conservando únicamente registros de eventos para fines de auditoría.

Las Fig. 6 y Fig. 7 muestran la respuesta del sistema durante la verificación biométrica: en la Fig. 6 se observa la detección de persona no entrenada y el bloqueo automático; en la Fig. 7 se aprecia la identificación correcta del usuario autorizado y la habilitación del proceso previo al envío del token hacia la pasarela.

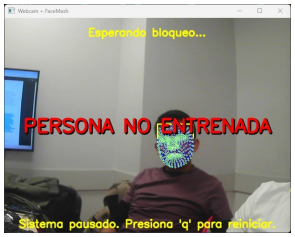

Fig. 6. Detección facial en usuario no registrado y bloqueo del sistema.

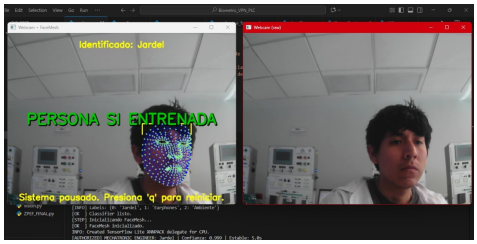

Fig. 7. Reconocimiento facial exitoso en usuario autorizado.

B. *Entrenamiento con Teachable Machine*

Se eligió TM (Teachable Machine) por su flujo rápido y la exportación directa a TFLite, lo que habilita dispositivos de borde con recursos moderados, y diversos estudios comparativos señalan sus ventajas frente a pipelines basados en MobileNet y YOLO para tareas de entrada rápida y datasets pequeños, manteniendo un desempeño adecuado para clasificación binaria [10]. El modelo y labels se integraron en main.py con keras/tensorflow y se probaron localmente antes del despliegue en la Raspberry Pi [21].

C. *Detección facial y landmarks con MediaPipe*

MediaPipe provee detección facial y landmarks con latencia apta para bucles de control ligeros y estabilidad suficiente ante cambios moderados de iluminación/pose [11]. Su uso en monitorización clínica de movilidad facial y en interfaces por gestos respalda precisión y robustez para escenarios interactivos en tiempo real [12]. Además, opera como sensor de presencia: si no hay rostro, el sistema se interpreta como “persona no” y se pausa la evaluación para evitar falsos disparos [14].

D. *Anti-spoofing mediante parpadeo y giro de cabeza*

Con el fin de evitar accesos falsos por imágenes estáticas, se desarrolló un módulo de verificación activa liveness.py que exige gestos fisiológicos voluntarios del operador antes de habilitar el reconocimiento facial. Estos gestos requeridos corresponden a un parpadeo y a un movimiento rotacional de la cabeza, ambos diseñados para demostrar presencia real del usuario. El sistema combina detección de parpadeo basada en el Eye Aspect Ratio (EAR) y estimación de giro de cabeza (yaw heurístico) a partir de los landmarks faciales proporcionados por MediaPipe [11], [12]. Ambos eventos deben producirse dentro de una ventana temporal determinada para validar la autenticidad del usuario. Si las condiciones no se cumplen o el tiempo expira, el módulo se reinicia automáticamente y retorna el sistema a estado seguro, asegurando que ninguna imagen del usuario autorizado pueda activar el permiso. Una vez detectados el parpadeo y el giro correctos, el módulo habilita la etapa siguiente de reconocimiento facial, en la cual se evalúa la identidad del operador mediante un clasificador entrenado [13]. Si la coincidencia biométrica es positiva, se genera un token válido que se envía al Gateway mediante el protocolo HTTP. Como medida contra ataques de *replay* por video, tras el reconocimiento facial el sistema ejecuta *antispoof_video.py*, que implementa unas pruebas basado en un patrón aleatorio de parpadeos y giro de cabeza en tiempo real. Dado que un video pregrabado no puede responder a estos desafíos, los intentos de suplantación son descartados. Solo si el patrón coincide, se valida la autenticidad del operador.

V. Red Privada Virtual y Control de Acceso

A. *Topología y políticas de acceso*

Se creó la red con Network ID 93afae59638329ed y se aprobaron los nodos; se asignaron Managed IPs internas para laptop y Raspberry, correspondientes a 10.242.143.6 y 10.242.43.135 respectivamente, manteniendo el PLC no expuesto a Internet [6]. El Network Controller permite aprobar y denegar altas de nodos y observar su estado, proporcionando un control de acceso adicional de bajo costo operativo [6], [7].

B. *Cliente del operador*

La laptop se unió a la red virtual, verificó rutas y asignación de IP interna, y aisló el tráfico de control para que fluya solo por la interfaz ZeroTier, reduciendo fuga por la red física [6]. Se estableció una línea base de latencia y se programaron pruebas periódicas para detectar degradaciones y reconexiones [7].

C. *Pasarela del Gateway*

La Raspberry Pi 5 se unió a la red, dejó wlan0 para Internet y la VPN, y eth0 dedicada a la comunicación con el PLC, con la microsegmentación, lo que evita choques entre tráfico de control e Internet y mantiene predecible el camino de datos hacia el PLC [7]. Sobre la pasarela se ejecuta zt_gateway_control.py, encargado de controlar por HTTP el estado del puente entre ZeroTier y la interfaz física eth0 del Gateway. Los comandos enviados están vinculados al módulo biométrico; el puente permanece cerrado por defecto y solo se habilita tras una autorización facial válida, cerrándose automáticamente ante no reconocimiento, ausencia de rostro o caída del túnel [4]. En la Fig. 8 se muestra parte del código disponible en un repositorio público de GitHub (https://github.com/jardelhh64ecommits/PROYECTO_BIOMETRIC_VPN_PLC_GATEWAY.git), el cual implementa la habilitación y el bloqueo del puente de red en la Raspberry Pi. Este código activa el reenvío de paquetes y la microsegmentación solo cuando la autenticación biométrica es válida; de lo contrario, bloquea el tráfico hacia el PLC en cualquier condición no autorizada.

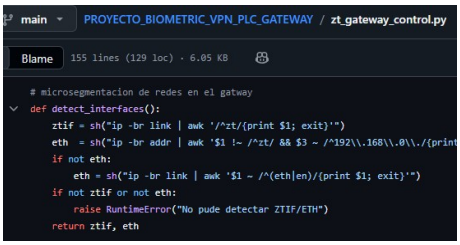

Fig. 8. Fragmento del código en la Raspberry Pi que controla el puente VPN al PLC según la autorización biométrica.

Reglas de microsegmentación (IP/puertos). En la implementación, zt_gateway_control.py expone un plano de control HTTP únicamente dentro del dominio ZeroTier, enlazado a la IP gestionada del Gateway (10.242.43.135) y escuchando en el puerto 8088. El endpoint acepta acciones {on, off, status} y no es accesible desde Internet al estar ligado a la interfaz ZeroTier. Para reducir superficie de ataque, se aplica deny-by-default: (I) el control HTTP se restringe a nodos autorizados (p.ej., lista blanca de la IP del operador 10.242.143.6); y (II) el plano de datos OT se limita al tráfico estrictamente necesario hacia el PLC, permitiendo únicamente S7comm sobre TCP/102 hacia 192.168.0.1 a través del Gateway, bloqueando cualquier otro puerto, protocolo o destino no requerido por la operación. Control de sesión y TTL del permiso. La habilitación del puente y del bit de permiso en el DB se considera una sesión temporal. Tras una autorización biométrica válida, el acceso se habilita por un intervalo (TTL, 30s). Al expirar el TTL, o ante no reconocimiento, ausencia de rostro, fallo de liveness.py o caída del túnel, el sistema retorna a estado seguro: cierra el puente (deshabilita NAT) y fuerza el bit de permiso del DB a 0, evitando accesos persistentes no supervisados [22].

D. *Autorización biométrica sobre túnel VPN*

El proceso de autorización se implementó sobre el dominio privado de ZeroTier, garantizando que todas las peticiones HTTP y los tokens de control se transporten por la VPN [6], [7]. En este esquema, la laptop ejecuta el módulo de envío de órdenes de control firmadas por la VPN mediante un código interno (llave_cliente.py) que actúa como cliente

HTTP encargado de enviar solicitudes hacia la Raspberry Pi 5. El cliente mantiene una variable global "token_activo" que solo se habilita cuando el sistema biométrico ha reconocido al usuario y superado el mecanismo de "anti-spoofing". Mientras "token_activo" permanece en "False", cualquier intento de enviar acciones HTTP es bloqueado localmente. Una vez validada la identidad, la función "activar_token()" cambia este valor a "True" y permite ejecutar solicitudes. El servidor interpreta estas solicitudes y, dependiendo del parámetro "action", ejecuta las acciones "on", "off" o "status" para habilitar, bloquear o consultar el estado del puente entre la interfaz ZeroTier y eth0 del Gateway [4]. Protección anti-replay del token HTTP. Para prevenir ataques de repetición, las órdenes HTTP incorporan un timestamp (ts) y un "nonce" único, firmados mediante HMAC (SHA-256) con una clave compartida K entre la laptop y el Gateway (K corresponde al token precompartido del sistema). Cada solicitud transporta {action, ts, nonce, sig}. El Gateway valida que ts se encuentre dentro de una ventana temporal acotada, que el "nonce" no haya sido utilizado previamente (almacenándolo con expiración), y que la firma coincida. Si alguna verificación falla, la orden se rechaza y el puente permanece cerrado (estado seguro). En este esquema, la clave K no se transmite en la URL; se usa únicamente para firmar y verificar las solicitudes. De este modo, el tráfico de control hacia el PLC solo se habilita bajo autenticación facial válida, aun cuando el túnel VPN permanezca activo, reforzando el principio de separación entre autenticación lógica y autorización de red. La arquitectura resultante impide que un cliente no autenticado interactúe con el PLC, incluso si pertenece al mismo dominio ZeroTier, cumpliendo los principios de microsegmentación segura y aislamiento de red descritos por Hrițcan et al. [6] y Kaniewski et al. [23].

E. Endurecimiento y monitoreo de red

El dominio ZeroTier se complementa con reglas de filtrado en el borde de lista de servidor y cliente para reducir superficie de ataque IoT [6]. Se monitoriza disponibilidad y reconexión del túnel y se registran eventos para analizar continuidad de servicio en escenarios reales [24], [25].

VI. Integración del control con PLC

Para validar el control remoto real sobre el proceso, se modificó en TIA Portal la salida Q0.0 por Q0.1 con el fin de simular una falla de campo de las salidas que se presentarían en un ámbito industrial. Este escenario permitió comprobar que el sistema puede ajustar la programación del PLC de forma remota y restaurar el estado esperado del actuador. En paralelo, se realizaron pruebas físicas mediante Python y snap7 para activar el bit correspondiente en el DB asociado a esa salida, confirmando la respuesta efectiva del hardware frente a la señal remota.

A. Configuración del DB y asignación de memoria

En TIA V16 se creó un DB global con el bit de permiso en (DB=1, byte=0, bit=0); se desactivó acceso optimizado para permitir Put/Get desde Python y snap7 con offsets explícitos [17]. Registrar los offsets y tipos de datos garantiza la correspondencia entre el PLC y las estructuras utilizadas en Python, evitando errores de lectura o escritura durante la operación [18].

B. Comunicación Put/Get mediante Python y snap7

El script main.py se conecta al PLC configurado en la IP 192.168.0.1, Rack 0 y Slot 1. Desde allí escribe el bit de permiso y realiza lecturas de estado para confirmar la operación; además, se registran tiempos de ciclo y tasa de éxito con el fin de evaluar la estabilidad del enlace [18]. La literatura respalda la integración entre módulos de visión computacional y PLC, así como el uso de Python como capa de orquestación, lo que valida este enfoque de bajo acoplamiento para aplicaciones industriales [26].

C. Lógica Ladder de permiso y seguridad ante fallos

La secuencia del semáforo, que incluye las etapas de arranque, rojo, amarillo, verde, reinicio y parada de emergencia, permanece bloqueada mientras el permiso no esté activo. Una vez habilitado el permiso, las transiciones se ejecutan mediante enclavamientos que evitan cambios indeseados provocados por posibles fluctuaciones en la biometría [17]. Tras cualquier evento anómalo o pérdida del túnel VPN, se requiere una cancelación manual para retornar a un estado seguro, evitando arranques inesperados al restablecerse la comunicación [1].

D. Pruebas del enlace y validación extremo a extremo

Las pruebas del enlace se centraron en evaluar la estabilidad de la comunicación entre la laptop, la VPN, la Raspberry Pi 5 y el PLC. Se midieron tiempos de respuesta, picos de latencia y pérdidas tanto en operación local como remota, lo que permitió caracterizar el impacto del túnel VPN en el enlace industrial [7]. Adicionalmente, se evaluaron reconexiones del túnel y su efecto en la continuidad del proceso. Cuando se detecta pérdida de la VPN o ausencia de rostro, el sistema pasa automáticamente al estado STOP y permanece bloqueado hasta que la autenticación biométrica y la comunicación con el PLC se restablecen satisfactoriamente [27].

VII. Monitoreo del proceso implementado en n8n

A. Instancia local de n8n

Para la ejecución de una instancia local de n8n (Nodemation), se instaló Node.js y el paquete de automatización n8n, obtenido desde su repositorio oficial de GitHub. Dicha instalación permite disponer de n8n de manera local y acceder a la herramienta mediante el puerto localhost 5678, todo ello ejecutado en un sistema operativo Windows, donde se encuentra nuestro servidor cliente. La base de datos de n8n, donde se almacenan los flujos, credenciales, ejecuciones, entre otros elementos, se gestiona mediante SQLite, lo que permite tener control total sobre estos registros.

Asimismo, posibilita mantener un número prácticamente ilimitado de flujos, dependiendo del tamaño de almacenamiento; esto contrasta con la versión en la nube de n8n que presenta limitaciones dependiendo del plan de servicio que se adquiera.

B. Monitoreo de los datos del Gateway con n8n

Esta plataforma n8n permite comunicarse con el Gateway a través del nodo Webhook, que permite recibir datos desde una URL. Los datos son enviados por un código en Python que son ejecutados en el Gateway. Este código envía valores 0 y 1, los cuales estos representan los estados activo e inactivo. Este registro sirve como base para alimentar al agente de IA y posibilitar la consulta del estado del PLC en tiempo real. Los valores transmitidos se almacenan en una hoja de Google Sheets con el fin de registrar el valor recibido junto con la hora y fecha de cada envío. Esto permite mantener una base de datos útil para el funcionamiento del agente de IA y posibilita el monitoreo en tiempo real del sistema.

C. Flujo de trabajo en n8n

Para el desarrollo del flujo de trabajo se aplicó una lógica bifurcada, la cual consistió en recibir los valores de 0 y 1, los cuales son guardados en la hoja de Google Sheet y posteriormente leídos al momento de realizar una consulta por WhatsApp. Después, dicha consulta textual llegada al agente, que finalmente respondía basándose en la información almacenada en la hoja de Google Sheets. Para este flujo se utilizaron los siguientes nodos:

a) El nodo Webhook es el encargado de recibir los valores de 0 y 1 que envía el sistema.

b) El nodo WhatsApp Trigger se encarga de enlazar el flujo con Meta para gestionar el envío y la recepción de mensajes desde el número configurado.

c) Los nodos Google Sheets Append y Google Sheets Read permiten registrar información en la hoja de cálculo y consultar los valores almacenados cuando se necesitan.

d) El nodo Code procesa y transforma los mensajes recibidos por WhatsApp para que el flujo continúe con datos organizados y coherentes.

e) El nodo IA Agent interpreta los mensajes procesados y aplica la lógica del modelo configurado para generar la respuesta adecuada dentro del flujo.

La estructura del flujo usada en el agente de IA permitió que el proceso de monitoreo se ejecutara de manera controlada. Tuyishime et al. [16] explican que n8n se considera una de las plataformas low code más versátiles para la integración de servicios mediante APIs REST, permitiendo ejecutar flujos complejos sin necesidad de programación extensiva. La Fig. 9 muestra el workflow implementado en n8n para el módulo de monitoreo inteligente, donde se visualizan los nodos utilizados y sus conexiones lógicas para recibir datos del Gateway, almacenarlos en Google Sheets y responder consultas realizadas por WhatsApp.

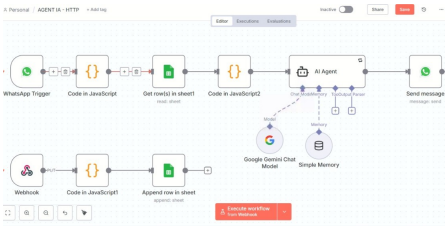

Fig. 9. Flujo de trabajo en n8n para el monitoreo del estado del PLC.

D. Funcionalidad del Agente de IA en n8n

El modelo Gemini Chat Model, el cual es de uso gratuito en n8n, se incorpora en el agente de IA que realiza el monitoreo del proceso. Este agente permite consultar sobre el estado del PLC de manera amigable sin seguir un proceso predefinido, además genera respuestas contextuales al usuario, debido al modelo puesto que a su vez interactúa con los valores almacenados en la hoja de Google Sheets, recibiendo así una respuesta precisa y no repetitiva. Barra et al. [15] afirman que la arquitectura agentiva permite coordinar múltiples subagentes responsables de tareas como toma de decisiones, generación de respuestas y gestión de la comunicación, todo dentro de un entorno unificado y sin intervención humana constante.

VIII. RESULTADOS

A. Resultados del módulo biométrico

La Tabla II resume el desempeño global del sistema biométrico (120 autorizados y 120 no autorizados), evaluado a 0.30–1.20 m con 10 s máximos por intento, umbral 0.95 y ventana de aceptación de 5 s. Se observa un FAR nulo y un FRR limitado, evidenciando un comportamiento conservador orientado a minimizar falsos aceptados.

TABLA II
MÉTRICAS BIOMÉTRICAS GLOBALES DEL SISTEMA

Métrica	Valor
N.º de intentos autorizados	120
N.º de intentos no autorizados	120
Accuracy (%)	95.42
FAR (%)	0.00 (0/120)
FRR (%)	9.17(11/120)
Umbral (confianza)	0.95
Ventana de Aceptación (s)	5

La Fig. 10 presenta la matriz de confusión correspondiente. Los errores se concentran en falsos rechazos (FN), mientras que no se registran falsos aceptados (FP=0), consistente con FAR=0.

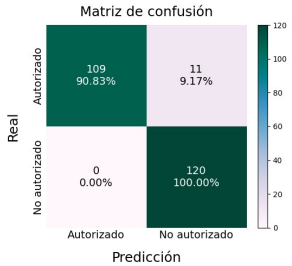

Fig. 10. Matriz de confusión del sistema biométrico.

B. *Robustez ante suplantación (anti-spoofing)*

La Tabla III presenta los resultados del módulo anti-spoofing (liveness.py) evaluado a distancias de 0.30, 0.60, 0.90 y 1.20 m, con 30 intentos por distancia para usuario real y 30 intentos por distancia usando fotografía. Se reporta la tasa de spoof detectado (bloqueos/total) y el falso rechazo sobre intentos reales (rechazos/total). Los resultados muestran detección completa de suplantación por foto en todas las distancias, mientras que el falso rechazo aumenta a mayor distancia, consistente con la degradación de calidad de captura.

TABLA III
EVALUACIÓN ANTI-SPOOFING EN FUNCIÓN DE LA DISTANCIA CON TASA DE DETECCIÓN Y FALSOS RECHAZOS

Distancia (m)	Usuario real: aceptados/30	Falso rechazo (%)	Ataque por foto: bloqueados/30	Spoof detectado (%)
0.30	30/30	0.00	30/30	100.00
0.60	30/30	0.00	30/30	100.00
0.90	30/30	0.00	30/30	100.00
1.20	26/30	13.33	30/30	100.00
Total	116/120	3.33	120/120	100.00

C. *Desempeño del control remoto PLC*

La Tabla IV reporta el desempeño del control remoto del PLC mediante operaciones Put/Get en LAN y sobre VPN (ZeroTier), con 50 ciclos por escenario. Put (ms) corresponde al tiempo de escritura del bit de permiso en el DB, mientras que Get (ms) corresponde al tiempo de lectura del mismo bit; End-to-end (ms) representa el tiempo total del ciclo de control. Para cada métrica se reporta el valor promedio y el percentil 95 (p95), que captura el peor caso típico bajo variabilidad de red. El criterio de éxito se define como escritura y lectura consistente del bit de permiso en el DB con la transición esperada en la lógica Ladder; el cambio de Q0.0 se emplea únicamente como verificación física secundaria.

TABLA IV
DESEMPEÑO DEL CONTROL REMOTO DEL PLC CON TASA DE ÉXITO Y LATENCIA EN LAN Y VPN

Métrica	LAN	VPN
N.º de intentos	50	50
Put (%)	100.00 (50/50)	100.00 (50/50)
Get (%)	100.00 (50/50)	100.00 (50/50)
Put (ms) prom/p95	6.71 / 12.72	97.07 / 121.04
Get (ms) prom/p95	2.99 / 4.63	47.82 / 68.99
End-to-end (ms) prom/p95	12.60 / 18.42	192.40 / 238.68

D. *Análisis comparativo por ruta de comunicación*

En la Tabla V muestra los resultados de latencia obtenidos en las cuatro rutas evaluadas. Las dos primeras pruebas fueron de conexiones físicas directas hacia el PLC, realizadas desde la laptop y desde la Raspberry Pi 5. Las otras dos rutas fueron evaluadas por conexión remota: la comunicación desde la laptop hacia la Raspberry Pi 5 y la comunicación desde la laptop hacia el PLC utilizando a la Raspberry Pi 5 como Gateway. Para cada ruta se realizaron 20 mediciones Internet Control Message Protocol (ICMP), a partir de las cuales se calcularon los valores mínimos, máximos y promedio.

TABLA V
RESULTADOS DE LATENCIA POR RUTA DE CONEXIÓN

Comunicación	Min (ms)	Max (ms)	Prom (ms)
Laptop a PLC por red LAN	2	3	2
Raspberry Pi a PLC mediante interfaz ETH0	0.039	0.077	0.054
Laptop a Raspberry Pi 5 a través de ZeroTier	38	409	158
Laptop a PLC vía Raspberry Pi 5 con ZeroTier	28	365	118

Los resultados muestran que todas las rutas establecieron comunicación con el PLC. Asimismo, todas las mediciones se completaron sin pérdida de paquetes, lo que confirma la correcta operación del acceso remoto propuesto. La Fig. 11 muestra la variación temporal de la latencia en las conexiones directas hacia el PLC. El enlace Laptop y PLC mantiene valores entre 2 y 3 ms, consistentes con una conexión Ethernet. Por otro lado, la Raspberry Pi logra latencias inferiores a 0.1 ms gracias a su interfaz eth0. Estos resultados indican que las conexiones embebidas proporcionan un enlace más estable y con menor variación en la latencia, adecuado para el uso de aplicaciones del control remoto del PLC.

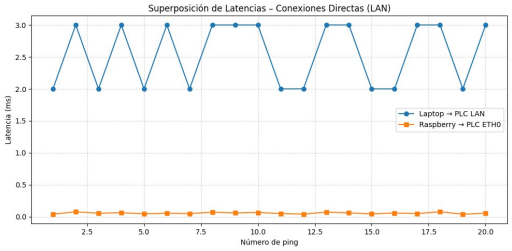

Fig. 11. Latencia en conexiones físicas Laptop–PLC y Raspberry–PLC.

La Fig. 12 muestra la variación temporal de la latencia en los enlaces que operan sobre ZeroTier. Ambas rutas remotas presentan fluctuaciones notorias y picos esporádicos propios del tránsito por Internet. Se observa que la ruta desde la laptop hacia la Raspberry Pi 5 y posteriormente al PLC presenta una estabilidad mayor en comparación con la ruta que llega únicamente a la Raspberry Pi 5, lo cual se aprecia en la comparación de los valores máximos y mínimos. Esto se debe a que la Raspberry Pi 5 actúa como un Gateway embebido con un enlace Ethernet microsegmentado hacia el PLC, lo cual reduce el promedio de la comunicación en comparación con el acceso remoto directo desde la laptop.

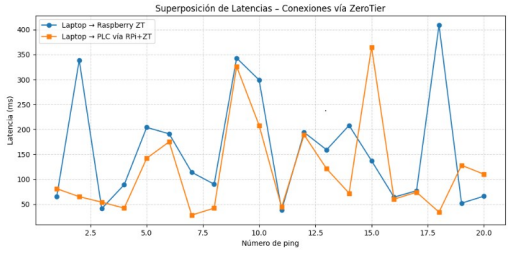

Fig. 12. Análisis temporal de latencia ZeroTier y comparación de valores extremos.

E. *Caracterización de red sobre ZeroTier*

La Tabla VI presenta la caracterización de red sobre ZeroTier para dos enlaces: laptop–Gateway y laptop–PLC vía Gateway, con 150 mediciones por enlace. Se reporta la pérdida de paquetes y el RTT en promedio, p95 y p99, donde los percentiles capturan el comportamiento en cola relevante para control remoto en OT. Asimismo, se incluye jitter (ms) como indicador de variabilidad temporal del RTT; valores mayores implican mayor inestabilidad del canal.

TABLA VI
MÉTRICAS DE RED SOBRE ZEROTIER: PÉRDIDA, RTT Y JITTER

Comunicación (ZeroTier)	Pérdida (%)	RTT prom (ms)	RTT p95 (ms)	RTT p99 (ms)	Jitter (ms)
Laptop a Raspberry Pi 5 con ZeroTier	0.00	12.25	30.00	61.29	7.21
Laptop a PLC vía Raspberry Pi 5 con ZeroTier	0.00	24.27	100.10	185.86	18.71

F. *Evaluación del rendimiento y robustez del Gateway*

Se evaluó la viabilidad del hardware propuesto mediante el análisis del consumo del Gateway durante la operación sostenida del sistema (autenticación biométrica, túnel ZeroTier activo y servicio HTTP). Los resultados muestran un uso promedio de CPU del 2.3 %, un consumo de RAM de 33 MB de 7933 MB y una temperatura de 52 °C. Además, se analizó la robustez frente a tres escenarios: cortes breves (10–30 s), cortes prolongados (2–5 min) y reinicio del dispositivo con el túnel activo. En los dos primeros, las reglas de iptables y el servicio HTTP se mantuvieron, y la conectividad con el PLC se restableció automáticamente sin nueva autenticación. En el tercer escenario, tras el reinicio, el reenvío de paquetes retornó a su estado desactivado. Estos resultados confirman el cumplimiento del principio de seguridad por defecto.

IX. DISCUSIÓN

Los resultados de este estudio experimental ofrecen respaldo empírico a una solución de seguridad biométrica aplicada al control y monitoreo de procesos industriales. La implementación de un modelo de reconocimiento facial basado en MediaPipe y Teachable Machine, junto con un algoritmo de detección de movimiento, constituye una herramienta eficaz para el reconocimiento seguro del operador en entornos industriales. A diferencia de estudios anteriores que aplicaron estos modelos para la detección de cacao [8], mango [9] o módulos educativos [4], [6], [7], [10], donde el objetivo fue la clasificación visual, en esta investigación se adaptan para un control de acceso como filtro de seguridad. En cuanto a la conectividad remota, la elección de ZeroTier junto con una Raspberry Pi 5 como pasarela se sustenta en evidencia previa sobre tecnologías VPN, su desempeño y costo de despliegue [7]. En este trabajo, la validación experimental demuestra que el acceso remoto al PLC es funcional tanto en LAN como sobre VPN, manteniendo éxito en operaciones Put/Get y permitiendo caracterizar el impacto en latencia y variabilidad mediante métricas de percentiles y jitter. Además, al incorporar control de tráfico en la pasarela y microsegmentación dinámica, la arquitectura no solo establece conectividad cifrada, sino que también limita el acceso a lo estrictamente necesario y fortalece la trazabilidad de las operaciones asociadas al PLC, en línea con enfoques de segmentación y gateways propuestos en la literatura [12], [13]. La latencia registrada sobre el túnel VPN es compatible con aplicaciones de monitoreo, trazabilidad, supervisión remota y control SCADA no crítico, donde los entornos OT aceptan tiempos de respuesta en el rango de 100–500 ms para estas operaciones en tiempo real. Para aplicaciones que demanden tiempos de respuesta inferiores a 10 ms, como control de movimiento o robótica de alta velocidad, se requerirían optimizaciones adicionales en el canal de comunicación o el uso de soluciones VPN de menor latencia. Como limitación, el desempeño biométrico puede degradarse a mayores distancias y bajo variaciones de pose e iluminación.

X. CONCLUSIONES

Este trabajo implementó un gateway de acceso remoto para PLC que integra autenticación facial con liveness.py y una pasarela basada en Raspberry Pi 5 sobre una VPN (ZeroTier), habilitando microsegmentación dinámica hacia un PLC Siemens S7-1200. En la evaluación biométrica se obtuvo un accuracy de 95.42%, FAR de 0.00% y FRR de 9.17% bajo umbral 0.95, mientras que el anti-spoofing por fotografía alcanzó 100% de detección con 3.33% de falsos rechazos globales. En el control PLC, las operaciones Put/Get lograron 100% de éxito en LAN y VPN, con una latencia end-to-end de 12.60 ms en LAN y 192.40 ms en VPN. Finalmente, el monitoreo conversacional mediante n8n posibilitó registrar autorizaciones y consultar estado vía WhatsApp, aportando trazabilidad operativa de sesiones. Por su bajo costo y uso de software libre, la propuesta es factible de replicar como solución piloto en laboratorios académicos y pymes, sujeta a validación adicional en escenarios de planta. No obstante, el estudio se validó en un banco de pruebas de laboratorio con condiciones y dataset biométrico acotados, por lo que la generalización puede degradarse ante variaciones más amplias de iluminación, pose y distancia; asimismo, el anti-spoofing fue evaluado principalmente frente a ataques por imagen y video, y el desempeño en VPN depende de la variabilidad de la red pública. Como trabajo futuro, se propone incorporar políticas de autorización por roles, reforzar endurecimiento OT y comparar experimentalmente ZeroTier con alternativas VPN bajo el mismo banco de pruebas y métricas.

AGRADECIMIENTOS

El primer autor dedica este trabajo a la memoria de su abuelo, Salvador Alejandro Huaranga Cristóbal, cuyo apoyo y motivación fueron fundamentales en su formación académica.

Declaración de divulgación sobre el uso de IA: Los autores declaran que se utilizaron herramientas de inteligencia artificial para generación de imágenes, y asistencia en la

generación de fragmentos de código. Todos los resultados generados con asistencia de IA fueron revisados, validados y editados por los autores.

Declaración de responsabilidad de los autores:
Los autores asumen la plena responsabilidad de la exactitud, originalidad e integridad de todo el contenido de este manuscrito, incluidos los componentes generados con ayuda de la Inteligencia Artificial.

REFERENCIAS

[1] W. Alsabbagh and P. Langendörfer, "Security of Programmable Logic Controllers and Related Systems: Today and Tomorrow," IEEE Open Journal of the Industrial Electronics Society, vol. 4, pp. 659–693, 2023, doi:10.1109/OJIES.2023.3335976.

[2] Kaspersky ICS CERT, "Threat landscape for industrial automation systems. Q1 2025," May 15, 2025. [Online]. Available: https://ics-cert.kaspersky.com/publications/reports/2025/05/15/threat-landscape-for-industrial-automation-systems-q1-2025/. [Accessed: Nov. 23, 2025].

[3] B. Qin and D. Yan, "Remote SCADA System Based on 3G VPN Services for Secondary Pressurization Pump Station," in 2010 International Conference on Intelligent System Design and Engineering Application, Changsha, China, 2010, pp. 132–135, doi:10.1109/ISDEA.2010.359.

[4] M. Bezenk and H. Korkmaz, "Remote Access Solution for Industrial Devices with VPN," in 2024 Innovations in Intelligent Systems and Applications Conference (ASYU), Ankara, Türkiye, 2024, pp. 1–8, doi:10.1109/ASYU62119.2024.10757061.

[5] Dragos, "2025 OT Cybersecurity Year in Review," 2025. [Online]. Available: https://www.dragos.com/ot-cybersecurity-year-in-review. [Accessed: Nov. 23, 2025].

[6] D.-F. Hriţcan, A. Graur, and D. Bălan, "Securing IoT Environments Using ZeroTier and OPNsense," in 2024 23rd RoEduNet Conference: Networking in Education and Research (RoEduNet), Bucharest, Romania, 2024, pp. 1–4, doi:10.1109/RoEduNet64292.2024.10722755.

[7] D.-F. Hriţcan, A. Graur, D.-G. Bălan, and E.-M. Timofte, "A Performance Analysis of VPN Technologies Used in an IoT Environment," Advances in Electrical and Computer Engineering, vol. 25, no. 2, pp. 81–88, 2025, doi:10.4316/AECE.2025.02009.

[8] F. G. Lopez and A. C. Fajardo, "Image-Based Classification of Healthy and Infected Cacao Pods Using Google Teachable Machine," in 2025 17th International Conference on Computer and Automation Engineering (ICCAE), Perth, Australia, 2025, pp. 131–136, doi:10.1109/ICCAE64891.2025.10980585.

[9] M. Pahati, L. C. De Jesus, R. Reyes, J. M. Villarroel, and M. R. Angeles, "Detecting Mango Leaf Diseases Using Google Teachable Machine for Sustainable Agriculture," in 2025 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Fukuoka, Japan, 2025, pp. 0611–0614, doi:10.1109/ICAIIC64266.2025.10920854.

[10] S. M. Saqib, M. Iqbal, T. Mazhar, T. Shahzad, K. Ouahada, and H. Hamam, "Effectiveness of Teachable Machine, mobile net, and YOLO for object detection: A comparative study on practical applications," Egyptian Informatics Journal, vol. 30, Art. no. 100680, Jun. 2025, doi:10.1016/j.eij.2025.100680.

[11] R. U. Karim, S. Mahdi, A. Samin, A. N. Zereen, M. Abdullah-Al Wadud, and J. Uddin, "Optimizing Stroke Recognition With MediaPipe and Machine Learning: An Explainable AI Approach for Facial Landmark Analysis," IEEE Access, vol. 13, pp. 32636–32660, 2025, doi:10.1109/ACCESS.2025.3550577.

[12] N. Abulail, A. Y. Owda, and M. Owda, "Real-Time Detection and Analysis of Facial Movement Disorders Using MediaPipe and OpenCV: A Streamlit-Based Framework," in 2025 12th International Conference on Information Technology (ICIT), Amman, Jordan, 2025, pp. 13–18, doi:10.1109/ICIT64950.2025.11049234.

[13] A. Kumar Raja, C. Sugandhi, G. Nymish, N. Sai Havish, and M. Rashmi, "Face Gesture Based Virtual Mouse Using MediaPipe," in 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1–6, doi:10.1109/I2CT57861.2023.10126453.

[14] J. Wang, S. Yuan, T. Lu, H. Zhao, and Y. Zhao, "Video-based real-time monitoring of engagement in E-learning using MediaPipe through multi-feature analysis," Expert Systems with Applications, vol. 288, Art. no. 128239, 2025, doi:10.1016/j.eswa.2025.128239.

[15] F. L. Barra, G. Rodella, A. Costa, A. Scalogna, L. Carenzo, A. Monzani, and F. Della Corte, "From prompt to platform: an agentic AI workflow for healthcare simulation scenario design," Advances in Simulation, vol. 10, Art. no. 29, May 2025, doi:10.1186/s41077-025-00357-z.

[16] A. Tuyishime, F. Basciani, L. Iovino, J. L. C. Izquierdo, J. Cabot, and A. Pierantonio, "Bridging Workflow Automation Tools and EMF Modeling Ecosystems," in 2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), Västerås, Sweden, 2023, pp. 893–897, doi:10.1109/MODELS-C59198.2023.00140.

[17] W. Koodtalang, T. Sangsuwan, and B. Noppakaow, "A Design of Automated Inspections of Both Shape and Height Simultaneously Based on Stereo Vision and PLC," in 2018 18th International Conference on Control, Automation and Systems (ICCAS), PyeongChang, South Korea, 2018, pp. 1290–1294.

[18] A. Truong-Duc-Duy and P. V. Tan, "A Python Framework for Model-Based Design, Commission and Monitor the Thermal Process," in 2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE), Kuala Lumpur, Malaysia, 2021, pp. 1–4, doi:10.1109/ICECCE52056.2021.9514203.

[19] P. Nguyen-Hoang and P. Vo-Tan, "Development An Open-Source Industrial IoT Gateway," in 2019 19th International Symposium on Communications and Information Technologies (ISCIT), Ho Chi Minh City, Vietnam, 2019, pp. 201–204, doi:10.1109/ISCIT.2019.8905157.

[20] A. Alvarez Oviedo, J. F. Mamani Villanueva, G. A. Echaiz Espinoza, J. M. M. Villanueva, A. O. Salazar, and E. R. L. Villarreal, "Design of a System for Driver Drowsiness Detection and Seat Belt Monitoring Using Raspberry Pi 4 and Arduino Nano," Designs, vol. 9, no. 1, Art. no. 11, 2025, doi:10.3390/designs9010011.

[21] S. Salim, M. M. A. Jamil, R. Ambar, W. S. W. Zaki, and S. Mohammad, "Learning Rate Optimization for Enhanced Hand Gesture Recognition using Google Teachable Machine," in 2023 IEEE 13th International Conference on Control System, Computing and Engineering (ICCSCE), Penang, Malaysia, 2023, pp. 332–337, doi:10.1109/ICCSCE58721.2023.10237148.

[22] L. Xia, B. Zhang, and J. Wang, "Research on Industrial Internet Data Collection Application Based on Raspberry Pi," in 2024 4th International Conference on Electronic Information Engineering and Computer Communication (EIECC), Wuhan, China, 2024, pp. 689–693, doi:10.1109/EIECC64539.2024.10929562.

[23] S. Kaniewski, L. Bechtel, P. Kneisel, M. Menth, and T. Heer, "Security Gateway for Automated Micro-Segmentation and VPN Encryption in Industrial Legacy Systems," IEEE Int. Conf. Emerging Technologies and Factory Automation (ETFA), Porto, Portugal, 2025, pp. 1–4, doi:10.1109/ETFA65518.2025.11205588.

[24] Z. Jia, D. Li, W. Zhang, and L. Pang, "5G MEC Gateway System Design and Application in Industrial Communication," in 2020 2nd World Symposium on Artificial Intelligence (WSAI), Guangzhou, China, 2020, pp. 5–10, doi:10.1109/WSAI49636.2020.9143280.

[25] V. K. Pandey, D. Sahu, S. Prakash, R. S. Rathore, P. Dixit, and I. Hunko, "A lightweight framework to secure IoT devices with limited resources in cloud environments," Scientific Reports, vol. 15, Art. no. 26009, 2025, doi:10.1038/s41598-025-09885-0.

[26] H. Benyezza, M. Bouhedda, I. Kratbi, A. Boukhtache, R. Kara, and Y. Aniba, "Automated Fruits Inspection and Sorting Smart System for Industry 4.0 Based on OpenCV and PLC," in 2023 2nd International Conference on Electronics, Energy and Measurement (IC2EM), Medea, Algeria, 2023, pp. 1–4, doi:10.1109/IC2EM59347.2023.10419533.

[27] H. Zhang, R. Zhang, and J. Sun, "Developing real-time IoT-based public safety alert and emergency response systems," Scientific Reports, vol. 15, Art. no. 29056, 2025, doi:10.1038/s41598-025-13465-7.