https://doi.org/10.22201/dgtic.30618096e.2025.3.4.143 **Vol. 3, Núm. 4.** octubre-diciembre 2025, págs. 86 - 94

# Implementación asistida por inteligencia artificial generativa de un certificado digital en un servidor web institucional restringido

**Información del reporte:**



Para citar este reporte técnico:

Olguín Hernández, M.E. (2025). Implementación asistida por inteligencia artificial generativa de un certificado digital en un servidor web institucional restringido. *Cuadernos Técnicos Universitarios de la DGTIC, 3* (4) páginas (86 - 94). https://doi.org/10.22201/dgtic.30618096e.2025.3.4.143

**Miriam Esther Olguin Hernández**
Centro de Investigaciones sobre América del Norte
Universidad Nacional Autónoma de México
molguinh@unam.mx
ORCID: 0009-0008-8053-2870

## Resumen

Durante el periodo reciente, se abordó la instalación de un certificado digital en un servidor web institucional que operaba en un entorno con acceso restringido. La actividad se desarrolló sin asistencia directa de personal especializado en seguridad informática. Se seleccionó un método compatible con distribuciones basadas en Unix que no requieren permisos de administrador, lo que permitió cumplir con los requerimientos técnicos mediante herramientas automatizadas. A lo largo del proceso, se consultaron recursos tecnológicos accesibles que facilitaron la interpretación de instrucciones, la solución de errores frecuentes y la verificación de resultados. Esta experiencia puso en evidencia la posibilidad de ejecutar procedimientos complejos mediante recursos digitales guiados, sin comprometer la integridad del entorno. Los hallazgos obtenidos contribuyeron al fortalecimiento de las competencias técnicas del personal técnico académico en materia de protección de servicios web institucionales. Permitieron adquirir experiencia práctica en la generación de certificados digitales sin privilegios de administrador, comprender métodos de validación de dominios como webroot frente a restricciones operativas y aprovechar herramientas automatizadas como acme.sh para gestionar certificados en entornos limitados. Además, se logró sistematizar un procedimiento replicable que puede ser de utilidad para otras entidades universitarias con recursos limitados o sin personal especializado en seguridad web.

https://doi.org/10.22201/dgtic.30618096e.2025.3.4.143 **Vol. 3, Núm. 4.** octubre-diciembre 2025, págs. 87 - 94

**Palabras clave:**

Entornos UNIX, seguridad web, ciberseguridad, modelos generativos.

***Abstract***

*During the recent period, the installation of a digital certificate on an institutional web server operating in a restricted-access environment was addressed. The activity was carried out without the direct assistance of specialized computer security personnel. A method compatible with Unix-based distributions that does not require administrator privileges was selected, allowing technical requirements to be met using automated tools. Throughout the process, accessible technological resources were consulted, which facilitated the interpretation of instructions, the resolution of common errors and the verification of results. This experience demonstrated the possibility of executing complex procedures using guided digital resources without compromising the integrity of the environment. The findings contributed to strengthening the technical competencies of academic technical staff in the protection of institutional web services. They provided practical experience in generating digital certificates without administrator privileges, understanding domain validation methods such as webroot in the face of operational restrictions and leveraging automated tools such as acme.sh to manage certificates in limited environments. In addition, a replicable procedure was systematized, which could be useful for other university entities with limited resources or without specialized web security personnel.*

***Keywords:***

*UNIX environments, web security, cybersecurity, generative models.*

# 1. INTRODUCCIÓN

En el ámbito universitario, la seguridad de los sitios web institucionales es fundamental para preservar la confidencialidad, integridad y disponibilidad de la información. La adopción de certificados digitales se ha consolidado como una estrategia clave, respaldada por estudios que destacan la necesidad de mecanismos criptográficos robustos ante el aumento de ataques a instituciones públicas (Stallings, 2022; Anderson, 2020).

En la Universidad Nacional Autónoma de México (UNAM), diversas dependencias operan entornos digitales en servidores virtuales sin privilegios de administrador, lo que limita la ejecución de acciones técnicas avanzadas. Esta condición, sumada al acompañamiento limitado de expertos en ciberseguridad o la falta de personal especializado en esta área, dificulta el cumplimiento de estándares de protección web y limita la ejecución autónoma de procesos técnicos críticos, como la instalación de parches de seguridad, la gestión directa de certificados digitales o la configuración de *firewalls* a nivel de sistema. Estas limitaciones pueden aumentar el riesgo de vulnerabilidades explotables si no se cuenta con mecanismos alternativos (Anderson, 2020; Kumar & Sharma, 2018).

Durante el mes de mayo de 2025, se enfrentó la necesidad de implementar un certificado SSL en un entorno operativo restringido, bajo un esquema sin permisos de superusuario, para un sitio institucional universitario adscrito a una Red Internacional de Investigación, alojado en un servidor con dominio *.unam.mx*. Esta situación reflejó una problemática frecuente: ¿cómo garantizar la seguridad web institucional en entornos con acceso técnico restringido y limitada capacidad operativa en materia de ciberseguridad?

https://doi.org/10.22201/dgtic.30618096e.2025.3.4.143 **Vol. 3, Núm. 4.** octubre-diciembre 2025, págs. 88 - 94

Ante este escenario, se optó por herramientas accesibles y procedimientos replicables, apoyados por inteligencia artificial generativa (IA), lo que permitió obtener una alternativa de solución viable y segura. Su uso facilitó la comprensión de instrucciones, la gestión de errores comunes y la conclusión exitosa del proceso técnico, incluso en ausencia de acompañamiento especializado.

El objetivo de la intervención técnica, presentada en este reporte, es construir e implementar un certificado digital válido en un servidor institucional operado sin privilegios de administrador, mediante una guía detallada y replicable para sistemas basados en GNU Linux, apoyada en el uso de inteligencia artificial generativa aplicada como recurso de soporte técnico autónomo.

## 2. DESARROLLO TÉCNICO

La implementación del certificado digital se llevó a cabo en un servidor institucional, administrado por la Dirección General de Tecnologías de Información y Comunicación (DGTIC), en un entorno con acceso limitado y sin privilegios de superusuario (ver Figura 1).

**Figura 1**

*Ambiente de trabajo*


Dado este contexto, se optó por una solución compatible con entornos de usuario estándar. Se seleccionó el cliente acme.sh, ya que permite generar certificados digitales válidos sin requerir instalación como *root* y ha sido documentado en la literatura técnica como una herramienta ligera y efectiva para contextos restringidos (Kumar & Sharma, 2018).

De forma complementaria, se utilizó ChatGPT (modelo GPT-4o; OpenAI, 2024), como recurso de apoyo para guiar cada fase del procedimiento técnico. A partir de una necesidad concreta (generar un certificado

https://doi.org/10.22201/dgtic.30618096e.2025.3.4.143 **Vol. 3, Núm. 4.** octubre-diciembre 2025, págs. 89 - 94

digital sin privilegios de administrador), se inició un proceso iterativo de consulta y validación, donde la herramienta fue proporcionando instrucciones ajustadas a cada contexto del servidor.

La formulación precisa de preguntas técnicas (incluyendo la descripción de los comandos utilizados y los mensajes de error obtenidos en consola) permitió interpretar correctamente instrucciones, comprender el funcionamiento de herramientas como acme.sh y adaptar procedimientos convencionales a un entorno restringido. Este enfoque no sólo orientó la solución, sino que facilitó a la IA identificar el punto exacto del fallo y proponer ajustes adecuados al entorno operativo.

Ante errores inesperados, como conflictos de permisos, fallos de validación o estructuras de directorios no reconocidas, se reformularon las consultas, lo que derivó en una mejora progresiva de las respuestas. Esta interacción continua con la IA facilitó una comprensión operativa del proceso y permitió alcanzar de manera autónoma la emisión del certificado digital, sin intervención de personal especializado ni privilegios de administrador, lo cual representó una alternativa viable y documentada para fortalecer la seguridad web institucional.

La generación del certificado requirió condiciones mínimas, como acceso al servidor vía SSH, ubicación del directorio raíz, permisos de lectura y escritura y un cliente SFTP para la transferencia de archivos, lo que permitió ejecutar el procedimiento de forma segura y autónoma, sin afectar la estabilidad del sistema.

**Verificación del sistema operativo**

Al ingresar al servidor mediante el cliente PuTTY para la conexión SSH, se verificó la distribución del sistema operativo ejecutando el comando cat /etc/os-release en la terminal Bash. Esta comprobación fue clave, ya que algunas herramientas de instalación requieren bibliotecas o entornos específicos para funcionar correctamente. En este caso, se descartó el uso de Certbot, una herramienta escrita en Python, debido a que la distribución CentOS Linux 7 no contaba con Python 3, una dependencia común en sistemas más recientes.

**Instalación de acme.sh sin privilegios de root**

La herramienta para emitir y gestionar certificados SSL seleccionada fue acme.sh, la cual es un cliente ligero y completamente escrito en shell (Bash) que permite emitir y renovar certificados SSL/TLS desde Let's Encrypt y otras autoridades certificadoras, sin necesidad de contar con privilegios de superusuario (*root*). Esta herramienta es especialmente útil en entornos restringidos, ya que no depende de Python ni de bibliotecas externas, a diferencia de otras soluciones como Certbot. Su funcionamiento se basa en el protocolo ACME (Automatic Certificate Management Environment) y permite realizar validaciones y emisiones automatizadas de certificados, siendo ideal para servidores con acceso limitado.

A partir de la descripción detallada del entorno operativo proporcionada, que incluía el tipo de servidor, la ausencia de privilegios de administrador y la estructura de directorios donde se alojaban los archivos del sitio web, se solicitó a ChatGPT (modelo GPT-4o) orientación específica para llevar a cabo la instalación de acme.sh de forma segura y compatible con dichas restricciones. Con base en esta información, la herramienta generó una secuencia de comandos adaptada al contexto, lo que permitió ejecutar correctamente cada paso.

https://doi.org/10.22201/dgtic.30618096e.2025.3.4.143 Vol. 3, Núm. 4. octubre-diciembre 2025, págs. 90 - 94

Como parte del inicio de la instalación de acme.sh, la indicación de ChatGPT fue crear un directorio dentro del entorno del usuario en la ruta donde se encuentra alojado el sitio web (/home/sitiowebinstitucional/), con el fin de almacenar el *script* y sus archivos de configuración: mkdir -p ~/.acme.sh

Con base en el análisis del entorno y la ruta establecida, se indicó la instrucción adecuada para obtener el *script* principal desde su repositorio oficial en GitHub, almacenándolo directamente en el directorio recién creado: curl -o ~/.acme.sh/acme.sh https://raw.githubusercontent.com/acmesh-official/acme.sh/master/acme.sh

Posteriormente, se dieron los permisos de ejecución respectivos: chmod +x ~/.acme.sh/acme.sh

**Configuración de la autoridad certificadora (CA)**

Finalizada la instalación de acme.sh, de acuerdo con las indicaciones proporcionadas, se procedió a configurar la autoridad certificadora predeterminada. Aunque esta herramienta utiliza por defecto ZeroSSL, la sugerencia generada ante la consulta realizada indicó cambiarla debido a problemas recurrentes de visibilidad y acceso a los archivos locales que esa opción ocasionaba en entornos restringidos.

Como alternativa, se recomendó establecer a Let's Encrypt como autoridad certificadora principal, lo cual permitió una integración más fluida y sin restricciones adicionales. La instrucción correspondiente fue: ~/.acme.sh/acme.sh –set-default-ca –server letsencrypt

**Emisión del certificado SSL**

El siguiente paso consistió en generar el certificado digital para el dominio correspondiente al sitio web institucional. Para ello, se empleó el método webroot (validación mediante archivos temporales colocados en el directorio público del sitio), que permite demostrar el control sobre el dominio sin modificar la configuración del servidor: ~/.acme.sh/acme.sh –issue –webroot /home/sitiowebinstitucional/htdocs -d www.sitiowebinstitucional.unam.mx

La elección del método webroot respondió a las limitaciones técnicas del entorno, particularmente la falta de privilegios de administrador. Dado que no era posible modificar la configuración de red ni suspender servicios activos como Apache (servidor web que gestiona las solicitudes HTTP), se optó por esta alternativa, que permitió validar el dominio sin intervenir en los puertos del servidor.

**Creación manual del directorio del dominio (en caso necesario)**

En algunos entornos, la ejecución del comando anterior no crea automáticamente el directorio correspondiente al dominio en la ruta de acme.sh. Por ello, la instrucción fue verificar su existencia y, en caso de ausencia, ejecutar: mkdir -p ~/.acme.sh/www.sitiowebinstitucional.unam.mx_ecc. Esto aseguró que los archivos generados (.cer, .key, .csr) pudieran almacenarse correctamente para su posterior uso.

**Verificación de archivos generados**

Para validar la correcta emisión del certificado, se ingresó al directorio: cd ~/.acme.sh/www.sitiowebinstitucional.unam.mx_ecc

En este punto, se esperaban los archivos: www.sitiowebinstitucional.unam.mx.key, www.sitiowebinstitucional.unam.mx.cer, fullchain.cer y ca.cer. La generación de estos elementos confirmó el éxito de la operación y su disponibilidad para empaquetado y entrega institucional.

https://doi.org/10.22201/dgtic.30618096e.2025.3.4.143 **Vol. 3, Núm. 4.** octubre-diciembre 2025, págs. 91 - 94

**Empaquetado de certificados**

Con los archivos generados, se procedió a agruparlos en un archivo comprimido:

tar czf certs-sitiowebinstitucional.tar.gz www.sitiowebinstitucional.unam.mx.key fullchain.cer www.sitiowebinstitucional.unam.mx.cer

Este paquete se entregó a la DGTIC mediante una solicitud realizada a través del sitio https://soporte.dc.unam.mx, la cual fue gestionada por el responsable TIC del área universitaria correspondiente.

**Renovación del certificado**

Aunque el certificado generado tiene una validez limitada, acme.sh permite su renovación automatizada mediante el siguiente comando: ~/.acme.sh/acme.sh –renew -d www.sitiowebinstitucional.unam.mx

En caso de necesitar renovación inmediata por problemas técnicos, la herramienta sugiere utilizar la opción forzada: ~/.acme.sh/acme.sh –renew –force -d www.sitiowebinstitucional.unam.mx

**Verificación del contenido del certificado**

Finalmente, se realizaron pruebas de validación del contenido del certificado digital generado para asegurar su correcto formato y vigencia. Se indicó el uso de los siguientes comandos dentro del directorio /home/sitiowebinstitucional/.acme.sh/www.sitiowebinstitucional.unam.mx_ecc/:

openssl x509 -in fullchain.cer -text -noout

openssl x509 -enddate -noout -in fullchain.cer

Este paso fue importante para verificar que los archivos fueran funcionales y estuvieran listos para ser instalados en el entorno institucional.

## 2.1 ERRORES TÉCNICOS RECURRENTES

**Ausencia de Python 3**

Al intentar instalar certbot como primera opción, el sistema arrojó el mensaje `python3: command not found`, lo que evidenció la falta de esta dependencia en el entorno operativo. Por ello, se descartó el uso de certbot y ChatGPT (modelo GPT-4o) optó utilizar acme.sh, herramienta que no requiere Python y se ajusta a sistemas sin privilegios administrativos: curl https://get.acme.sh | sh

***Script* certbot-auto obsoleto**

Al intentar ejecutar certbot-auto, el sistema devolvió errores 404 y fallos de permisos, confirmando su desuso oficial, por lo que se eliminó esta opción del flujo operativo y se reforzó la selección de acme.sh como solución principal.

**Falta de crontab**

Durante la instalación de acme.sh, el sistema sugirió habilitar *crontab* para automatizar la renovación, pero no se contaba con acceso a este componente, por lo que se emitió el certificado manualmente sin cron: ~/.acme.sh/acme.sh –issue –webroot /home/sitiowebinstitucional/htdocs -d www.sitiowebinstitucional.unam.mx. Esta alternativa respondió a una necesidad operativa inmediata, sin requerir configuración adicional por parte del administrador del sistema.

https://doi.org/10.22201/dgtic.30618096e.2025.3.4.143 Vol. 3, Núm. 4. octubre-diciembre 2025, págs. 92 - 94

**Emisión inicial con ZeroSSL**

El primer intento de emisión generó certificados bajo ZeroSSL, autoridad por defecto en acme.sh, pero sus archivos no fueron localizables en el directorio esperado, por lo que se modificó la configuración para usar Let's Encrypt como autoridad principal y se reemitió el certificado exitosamente: ~/.acme.sh/acme.sh –set-default-ca –server letsencrypt

~/.acme.sh/acme.sh –issue –force –webroot /home/sitiowebinstitucional/htdocs -d www.sitiowebinstitucional.unam.mx

**Ausencia de archivos .cer y .key tras la emisión**

Luego de emitir el certificado, los archivos esperados no se encontraron en el directorio de almacenamiento, lo que generó dudas sobre si el proceso se había completado correctamente, por lo que se ejecutó una renovación forzada con acme.sh, lo cual regeneró los archivos en la ruta prevista: ~/.acme.sh/acme.sh –renew –force -d www.sitiowebinstitucional.unam.mx

## 2.2 METODOLOGÍA

La implementación del certificado digital en un entorno universitario sin privilegios de administrador implicó un proceso dividido en cuatro etapas: diseño, configuración, instalación y pruebas. Cada fase fue desarrollada con base en criterios técnicos fundamentados, priorizando herramientas ligeras, seguras y replicables. Para resolver dudas, validar comandos y superar errores operativos, se recurrió al uso de inteligencia artificial generativa como recurso de apoyo adaptativo, lo que permitió orientar cada paso con base en las condiciones reales del servidor. La metodología aplicada respondió a la necesidad de una solución replicable, segura y adaptada a entornos con acceso limitado.

**Diseño**

Durante esta etapa, se identificaron los requerimientos del entorno: sistema CentOS 7, sin acceso *root* ni instalación de dependencias complejas como Python. Con base en ello, se descartaron herramientas como certbot y se eligió acme.sh, una alternativa ligera, portable y adecuada para usuarios sin privilegios elevados (Kumar & Sharma, 2018; Scherf, 2021). La decisión se apoyó también en buenas prácticas documentadas por la comunidad de Let's Encrypt (ISRG, 2025).

**Configuración**

Se validaron los elementos mínimos necesarios para el procedimiento: sistema operativo compatible, ubicación del directorio raíz del sitio, acceso SSH (mediante el cliente PuTTY) y permisos de escritura. Se seleccionó el método de validación *webroot*, que permite demostrar el control del dominio sin modificar los puertos del servidor ni detener servicios activos como Apache (servidor HTTP). Asimismo, se configuró a Let's Encrypt como autoridad certificadora predeterminada, en lugar de ZeroSSL, debido a problemas de accesibilidad y visibilidad de archivos detectados en pruebas preliminares.

**Instalación**

La instalación se llevó a cabo en un entorno controlado, empleando un *script* accesible sin permisos de *root*. La herramienta seleccionada permitió generar y organizar los archivos requeridos (.key, .cer, .csr) de forma autónoma. Las instrucciones específicas se obtuvieron en función de consultas detalladas realizadas a un modelo de lenguaje generativo, el cual adaptó cada paso a las restricciones del servidor.

https://doi.org/10.22201/dgtic.30618096e.2025.3.4.143 Vol. 3, Núm. 4. octubre-diciembre 2025, págs. 93 - 94

Esta asistencia permitió superar errores comunes, como conflictos de permisos, rutas incorrectas o incompatibilidades entre herramientas.

**Pruebas**

Los certificados generados fueron verificados en cuanto a formato, vigencia y funcionamiento. Una vez entregados a la DGTIC, se habilitó el acceso seguro al sitio web institucional. Esta validación se realizó siguiendo buenas prácticas señaladas por Stallings (2022), quien destaca el uso de certificados digitales confiables y del protocolo HTTPS como elementos clave para proteger las comunicaciones web.

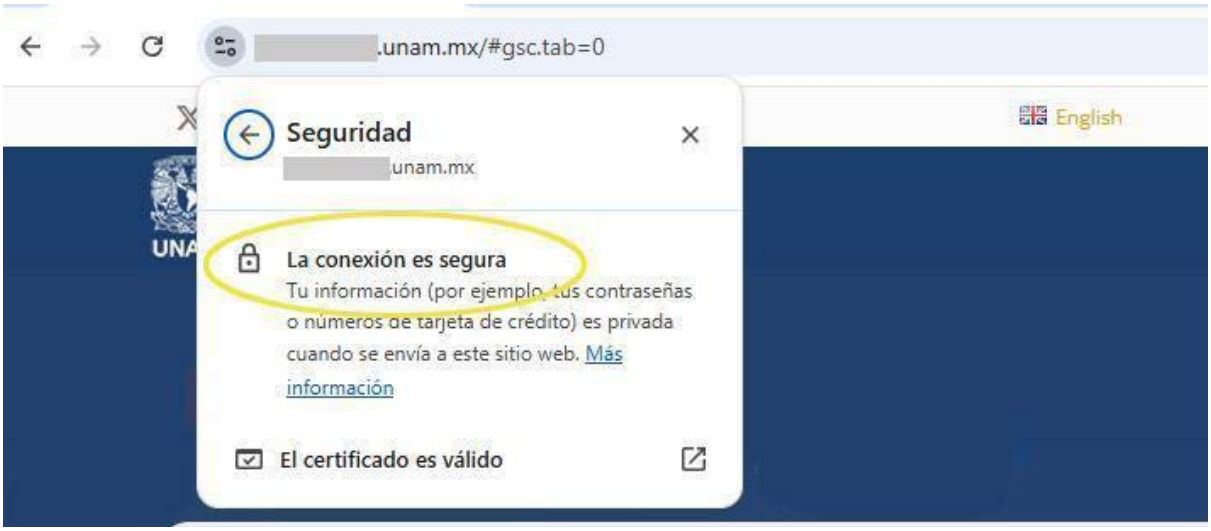
## 3. RESULTADOS

La instalación del certificado digital en el sitio web institucional se completó con éxito, logrando la emisión, validación y activación de un certificado válido mediante el cliente acme.sh. Esta acción resolvió la problemática técnica planteada, al habilitar el protocolo HTTPS en un entorno sin privilegios de administrador.

Los archivos generados (.key, .cer, fullchain.cer) fueron empaquetados y entregados a la DGTIC, donde el administrador del servidor realizó su instalación. Posteriormente, se verificó el acceso seguro al sitio, la validez del certificado mediante comandos de OpenSSL y se navegó en el entorno web https://www.sitiowebinstitucional.unam.mx (ver Figura 2).

**Figura 2**

*Sitio Web institucional (seguro)*




Durante el procedimiento se identificaron fallos como la emisión inicial por ZeroSSL, la ausencia de Python 3 y la falta de acceso a *crontab*, los cuales fueron solucionados mediante ajustes técnicos y el uso de inteligencia artificial generativa como herramienta de apoyo.

https://doi.org/10.22201/dgtic.30618096e.2025.3.4.143 **Vol. 3, Núm. 4.** octubre-diciembre 2025, págs. 94 - 94

En comparación con opciones descartadas como *certbot*, acme.sh ofreció mayor compatibilidad y autonomía en entornos restringidos, lo que justificó su elección. Los resultados alcanzados confirman que la solución implementada fue efectiva, segura y replicable en condiciones similares.

## 4. CONCLUSIONES

Gracias al apoyo del uso de ChatGPT con el modelo GPT-4o como recurso de soporte técnico autónomo se pudo lograr el objetivo mencionado del reporte.

Se logró la generación, validación y entrega de un certificado SSL confiable, utilizando acme.sh como herramienta ligera y compatible con entornos sin acceso *root*, aplicando el método *webroot* para la verificación del dominio. Esta implementación permitió habilitar el protocolo HTTPS en el sitio web institucional, con lo cual se reforzó la confidencialidad e integridad de la información transmitida, alineándose con los estándares de seguridad web actuales.

El proceso técnico evidenció que es posible ejecutar procedimientos avanzados de protección web en servidores basados en GNU/Linux, sin necesidad de privilegios de administrador ni de dependencias complejas como Python o Certbot. Además, la automatización mediante *scripts* en shell (.sh) simplificó significativamente las tareas de emisión y renovación de certificados.

El acompañamiento proporcionado por la inteligencia artificial generativa favoreció el aprendizaje autónomo, orientó con precisión cada etapa del proceso y permitió reducir el margen de error durante la ejecución. Este enfoque híbrido, que combina herramientas de código abierto y asistencia por IA, representa una alternativa eficaz y replicable en otras dependencias universitarias con recursos limitados o sin personal dedicado a la seguridad informática.

Se recomienda documentar este procedimiento en la base de conocimientos institucional, fomentar la capacitación continua en temas generales de ciberseguridad y promover espacios de formación impartidos por especialistas. Finalmente, se sugiere considerar la incorporación sistemática de asistentes basados en inteligencia artificial generativa como recurso complementario para el diagnóstico, orientación y resolución de tareas técnicas en entornos operativos restringidos.

## REFERENCIAS

Anderson, R. (2020). *Security Engineering: A guide to building dependable distributed systems (3.ª ed.)*. Wiley.

Internet Security Research Group. (2025). Getting Started – Let's Encrypt. Recuperado de https://letsencrypt.org/getting-started/

Kumar, R., & Sharma, P. (2018). Comparative study of SSL Certificate Generation Techniques. *International Journal of Computer Applications,* 179(7), 1–4.

OpenAI. (2024). ChatGPT (modelo GPT-4o) [IA generativa]. https://chat.openai.com/

Scherf, T. (2021). *Obtain certificates with acme.sh*. ADMIN Magazine, 65. Recuperado de https://www.admin-magazine.com/Archive/2021/65/Obtain-certificates-with-acme.sh

Stallings, W. (2022). *Cryptography and Network Security: Principles and Practice (8ª ed.)*. Pearson Education.