Guía de Estudio en Español

CompTIA Security+ (SY0-701) — Capítulo 10

Understanding Cryptography and PKI — Criptografía e Infraestructura de Clave Pública

---

Este material es un **resumen conceptual elaborado con palabras propias**, pensado como guía de apoyo para estudiar junto al libro original en inglés. No es una traducción ni una reproducción del texto original; sigue la misma estructura y orden de secciones del libro, incluyendo recuadros equivalentes a los 'Remember This!' originales, para facilitar el repaso en español.

Incluye: resumen por secciones, tablas comparativas, glosario EN↔ES y preguntas de práctica originales. Es uno de los capítulos más densos del libro — tomate tu tiempo.

1. Introducción a Conceptos de Criptografía

- **Integridad:** garantiza que los datos no fueron modificados; se verifica comparando hashes.
- **Confidencialidad:** los datos solo son legibles por usuarios autorizados; se logra con cifrado (encryption).
- **Cifrado simétrico:** misma clave para cifrar y descifrar.
- **Cifrado asimétrico:** dos claves (pública y privada) como par matemáticamente relacionado, requiere PKI.
- **Esteganografía:** oculta datos dentro de otros archivos (ej. en el espacio blanco de una imagen).
- **Autenticación:** valida una identidad. **No repudio:** impide que alguien niegue haber realizado una acción.
- **Firma digital:** aporta autenticación, no repudio e integridad — es un hash del mensaje cifrado con la clave privada del remitente.

2. Integridad con Hashing

El **hashing** es un algoritmo matemático que produce una cadena hexadecimal de longitud fija (hash) a partir de datos. La misma entrada siempre produce el mismo hash. Se usa comparando dos hashes: si son iguales, los datos mantuvieron integridad.

■ Recordá esto!

El hashing verifica la integridad de datos como correos, archivos descargados o guardados en disco. Un hash es un número hexadecimal creado con un algoritmo de hashing.

2.1 Hash vs. checksum

Un **checksum** es mucho más corto (a veces 1-2 bits) y solo sirve para detectar errores rápidamente (ej. paridad en RAID-5, dígito verificador de tarjetas de crédito) — no es criptográficamente seguro, a diferencia de hashes fuertes como SHA-3.

2.2 Algoritmos de hashing

Algoritmo	Tamaño / estado
MD5	128 bits; comprometido desde 2004, ya no se considera seguro criptográficamente (pero se usa aún como checksum rápido).
SHA-1	160 bits; ya no aprobado para la mayoría de usos criptográficos.
SHA-2 (SHA-256, SHA-512, etc.)	Mejora de SHA-1; sigue en uso amplio.
SHA-3	Alternativa a SHA-2 creada fuera de la NSA; mismos tamaños de salida (224 a 512 bits).

2.3 HMAC

El **HMAC (Hash-based Message Authentication Code)** combina un hash (ej. MD5 o SHA-256) con una clave secreta compartida entre emisor y receptor. Aporta integridad *y* autenticidad: si un atacante modifica el mensaje, no puede recalcular el HMAC correcto porque no conoce la clave secreta. TLS e IPsec suelen usar variantes como HMAC-SHA256.

■ Recordá esto!

El hashing es una función de una sola vía: no se puede revertir para recrear los datos originales. Las contraseñas suelen almacenarse como hash, a menudo con 'sal' (salt) agregada antes de aplicar el hash.

2.4 Colisiones de hash

Una **colisión** ocurre cuando dos entradas distintas generan el mismo hash — indeseable, porque permite que un atacante use una contraseña distinta a la original y aun así 'coincida'. MD5 es especialmente vulnerable a colisiones.

3. Ataques de Contraseña

Online: intenta descubrir una contraseña contra un sistema activo (logs muestran Event ID 4625 en Windows). **Offline:** intenta descubrirla desde una base de datos o captura de paquetes robada.

Ataque	Cómo funciona
Diccionario	Prueba palabras de una lista/diccionario (incluye contraseñas comunes como '12345').
Fuerza bruta	Prueba todas las combinaciones posibles de caracteres.
Password spraying	Prueba UNA contraseña contra MUCHAS cuentas, luego repite con otra — evade bloqueos por intentos fallidos.
Pass the hash	Usa el hash robado de la contraseña directamente para autenticarse, sin necesidad de descifrarlo (vulnerable en NTLM).
Birthday attack	Explota la paradoja del cumpleaños para encontrar colisiones de hash más rápido de lo esperado.
Rainbow table	Compara hashes robados contra una base de datos precomputada de hashes/contraseñas.

■ Recordá esto!

Los ataques online adivinan contraseñas contra un sistema en línea; los offline, contra un archivo descargado. Los ataques de spraying intentan evadir las políticas de bloqueo de cuenta. Los ataques de cumpleaños explotan colisiones en algoritmos de hashing.

3.1 Salting y key stretching

El **salting** agrega caracteres aleatorios a la contraseña antes de aplicar el hash, evitando ataques de rainbow table. El **key stretching** va más allá: aplica el hash repetidamente (con la sal) para consumir más tiempo y recursos, frustrando a los atacantes. Técnicas comunes: **bcrypt** (basado en Blowfish), **PBKDF2** (usa HMAC, hasta 1.000.000 de iteraciones) y **Argon2** (ganador de la competencia PHC 2015).

■ Recordá esto!

Bcrypt, PBKDF2 y Argon2 son técnicas de key stretching que ayudan a prevenir ataques de fuerza bruta y de rainbow table. Salan la contraseña con bits adicionales y luego la procesan con un algoritmo criptográfico.

4. Confidencialidad con Cifrado (Encryption)

Texto plano (plaintext) es legible; **texto cifrado (ciphertext)** no lo es. Se aplica a **datos en reposo** (almacenados), **datos en tránsito** (viajando por la red) y **datos en uso** (procesados en memoria — normalmente descifrados temporalmente para poder trabajarlos).

4.1 Cifrado simétrico

Usa la **misma clave** para cifrar y descifrar (también llamado clave secreta o de sesión). Ejemplo simplificado: un cifrado de sustitución que mueve cada letra 'X' espacios. Nunca se debe reutilizar la misma clave para todo — reutilizarla debilita significativamente la seguridad.

■ Recordá esto!

El cifrado simétrico usa la misma clave para cifrar y descifrar datos en ambos extremos de la transmisión.

4.2 Cifrado por bloques vs. por flujo

Por bloques (block cipher): cifra datos en bloques de tamaño fijo (ej. 64 o 128 bits) — ideal cuando se conoce el tamaño de los datos (ej. un archivo). **Por flujo (stream cipher):** cifra un bit o byte a la vez — más eficiente para datos de tamaño desconocido o continuo (ej. audio/video en vivo). Nunca se debe reutilizar una clave con un cifrado de flujo.

4.3 Algoritmos simétricos comunes

Algoritmo	Tipo	Detalle
AES	Bloque, 128 bits	Claves de 128/192/256 bits. Rápido, eficiente y fuerte — el estándar actual (NIST).
3DES	Bloque, 64 bits	Mejora de DES, más lento que AES; se usa cuando el hardware no soporta AES.
Blowfish	Bloque, 64 bits	Rápido, incluso más que AES-256 en algunos casos; reemplazo general de DES.
Twofish	Bloque, 128 bits	Relacionado con Blowfish; claves de 128/192/256 bits.

■ Recordá esto!

AES es un cifrado simétrico de bloque fuerte que cifra en bloques de 128 bits, con claves de 128, 192 o 256 bits. 3DES cifra en bloques de 64 bits; NIST eligió AES como estándar actual, pero 3DES sigue usándose en hardware legacy.

5. Cifrado Asimétrico

Usa un par de claves matemáticamente relacionadas: **pública** (se comparte libremente, embebida en un certificado) y **privada** (nunca se comparte). Regla clave:

- Lo cifrado con la clave **pública** solo lo descifra la **privada** correspondiente.
- Lo cifrado con la clave **privada** solo lo descifra la **pública** correspondiente.

Es muy segura pero consume muchos recursos, por eso casi siempre se usa solo para el **intercambio de claves (key exchange)**: compartir de forma segura una clave simétrica que luego hace el trabajo pesado de cifrar los datos.

■ Recordá esto!

Solo la clave privada correspondiente puede descifrar información cifrada con la clave pública. Solo la clave pública correspondiente puede descifrar información cifrada con la clave privada. Varios métodos de cifrado asimétrico requieren un certificado digital y una PKI.

5.1 Certificados digitales

Un **certificado digital** incluye la clave pública y datos del propietario. Las **autoridades certificadoras (CA)** los emiten y gestionan. Elementos comunes: número de serie, emisor, fechas de validez, sujeto (subject), clave pública, uso de la clave, y atributos como CN, O, L, S, C.

Claves efímeras (ephemeral): duran poco y se recrean por sesión, aportando **perfect forward secrecy** (comprometer una clave no compromete sesiones pasadas). **Claves estáticas:** duran más (ej. la vigencia de un certificado). **ECC (criptografía de curva elíptica):** requiere menos procesamiento; ideal para dispositivos con poca energía (una clave ECC de 256 bits equivale en seguridad a una RSA de 3072 bits).

Longitud de clave: a mayor longitud, mayor fuerza. RSA soporta 1024/2048/4096 bits; NIST ya no recomienda 1024 bits — el mínimo actual es 2048.

6. Ofuscación

Técnicas que hacen que algo sea difícil de entender sin necesariamente cifrarlo.

Técnica	Qué hace
Esteganografía	Oculta datos dentro de otros archivos (audio, imagen, video), sin cifrarlos.
Tokenización	Reemplaza datos sensibles con tokens no sensibles; el dato original se guarda en una bóveda separada (común en pagos con tarjeta).
Masking (enmascaramiento)	Oculta parcial o totalmente un dato con símbolos (ej. mostrar solo los últimos 4 dígitos de una tarjeta).

■ Recordá esto!

Esteganografía, tokenización y masking son técnicas de ofuscación para proteger datos sensibles. La esteganografía oculta mensajes dentro de otros archivos; la tokenización reemplaza datos sensibles con tokens; el masking oculta parcial o totalmente los datos.

7. Uso de Protocolos Criptográficos

Uso	Qué cifra	Qué descifra
Firma digital de correo	Clave privada del remitente	Clave pública del remitente
Cifrado de correo	Clave pública del destinatario	Clave privada del destinatario
Cifrado de sitio web (TLS)	Clave pública del sitio (cifra la clave simétrica)	Clave privada del sitio; luego la clave simétrica cifra la sesión

■ Recordá esto!

Saber qué clave cifra y cuál descifra ayuda a responder preguntas del examen. Por ejemplo, si algo se cifra con una clave privada, se trata de una firma digital.

8. Protegiendo el Correo Electrónico

8.1 Firma digital

Proceso: 1) se calcula el hash del mensaje; 2) se cifra ese hash con la clave privada del remitente (esto ES la firma digital); 3) se envía el mensaje + la firma; 4) el receptor descifra la firma con la clave pública del remitente para revelar el hash, y lo compara con el hash que calcula sobre el mensaje recibido. Si coinciden, se confirma **autenticación** (lo envió quien dice), **no repudio** (no puede negar haberlo enviado) e **integridad** (no fue modificado).

■ Recordá esto!

Una firma digital es un hash de un mensaje cifrado con la clave privada del remitente. El receptor descifra la firma con la clave pública del remitente para revelar el hash. Si tiene éxito, brinda autenticación, no repudio e integridad.

8.2 Cifrado de correo

Con solo asimétrico: el remitente cifra con la clave pública del destinatario; el destinatario descifra con su propia clave privada. En la práctica, la mayoría de las apps combinan ambos: usan asimétrico para compartir de forma segura una clave de sesión simétrica, y luego usan esa clave simétrica (mucho más rápida) para cifrar el contenido real.

■ Recordá esto!

Al cifrar un mensaje de correo, la clave pública del destinatario cifra y el destinatario usa su clave privada para descifrar el mensaje.

S/MIME: estándar más popular para firmar y cifrar correo, combina cifrado simétrico y asimétrico usando certificados y PKI. Puertos típicos: 995 (POP3-TLS), 587 (SMTP-TLS), 993 (IMAP-TLS).

9. Cifrado de Transporte HTTPS

TLS reemplaza a SSL (que tiene vulnerabilidades conocidas y no debe usarse). TLS requiere certificados emitidos por una CA. Combina cifrado asimétrico (para el intercambio de clave) con simétrico (para cifrar la sesión).

■ Recordá esto!

TLS es el reemplazo de SSL y requiere certificados emitidos por una CA. TLS cifra tráfico HTTPS, pero también puede cifrar otros tipos de tráfico.

9.1 Proceso simplificado del handshake TLS

- 1. El cliente solicita una sesión HTTPS.
- 2. El servidor responde con su certificado (incluye su clave pública).
- 3. El cliente genera una clave simétrica de sesión y la cifra con la clave pública del servidor.
- 4. El cliente envía la clave simétrica cifrada al servidor.

- 5. El servidor la descifra con su clave privada. Ahora ambos conocen la clave de sesión.
- 6. Toda la sesión se cifra con esa clave simétrica.

9.2 Ataques de downgrade

Un **ataque de downgrade** fuerza al sistema a usar un protocolo más débil (ej. de TLS a SSL) para luego explotar esa debilidad (ej. el ataque POODLE), habilitando después un ataque on-path. La defensa es simplemente **deshabilitar SSL** y las suites de cifrado obsoletas en el servidor.

■ Recordá esto!

Los administradores deben deshabilitar suites de cifrado y protocolos débiles en los servidores. Cuando un servidor soporta tanto suites fuertes como débiles, los atacantes pueden lanzar ataques de downgrade para explotar la más débil.

10. Blockchain

Un **libro contable público, distribuido y descentralizado (open public ledger)**. Cada 'bloque' incluye: información de la transacción, información de las partes (mediante firma digital, no nombres reales) y un hash único. Cada bloque también incluye el hash del bloque anterior, formando la 'cadena'. Un bloque se agrega tras: 1) ocurrir una transacción, 2) ser verificada por una red de computadoras, 3) registrarse correctamente en un bloque, y 4) asignársele un hash único.

11. Identificando Limitaciones

- **Recursos vs. seguridad:** cifrar todo consume más espacio, memoria y procesamiento (~40% más de espacio es típico); hay que equilibrar costo y necesidad.
- **Velocidad:** se busca rapidez al cifrar/descifrar datos, pero lentitud deliberada al hashear contraseñas (para frustrar ataques de fuerza bruta).
- **Tamaño / overhead computacional:** dispositivos pequeños necesitan criptografía 'liviana' con menor huella de memoria.
- **Entropía:** aleatoriedad del algoritmo; a mayor entropía, mayor seguridad.
- **Predictibilidad:** generadores de números pseudoaleatorios son deterministas (riesgo); los generadores verdaderamente aleatorios usan factores ambientales (ej. ruido atmosférico).
- **Claves débiles:** claves cortas se descifran más fácilmente (ej. RSA de 1024 bits ya no se recomienda).
- **Longevidad:** cuánto durará un algoritmo antes de volverse obsoleto (ej. DES fue reemplazado por AES).
- **Reutilización de claves:** nunca reutilizar claves, especialmente con cifrados de flujo (fue el problema de WEP).

Ataques de texto plano conocido/elegido: si un atacante conoce parte del texto plano y su versión cifrada correspondiente, puede deducir el método de cifrado. Un **ataque de solo texto cifrado** (sin conocer nada del texto plano) normalmente solo funciona contra algoritmos débiles.

12. Componentes de una PKI

Una **PKI (Public Key Infrastructure)** es el conjunto de tecnologías para solicitar, crear, gestionar, almacenar, distribuir y revocar certificados digitales. Permite que dos partes se comuniquen de forma segura sin haberse conocido antes.

12.1 Autoridad certificadora (CA)

Emite, gestiona, valida y revoca certificados. La confianza se basa en el concepto de **raíz de confianza (root of trust)**: si el certificado raíz de la CA está en el almacén de certificados raíz confiables del sistema operativo/navegador, todos los certificados que emite son confiables.

12.2 Modelo de confianza jerárquico

La **CA raíz (root CA)** — mantenida offline por seguridad — emite certificados a **CAs intermedias** (online), que a su vez emiten certificados finales (leaf) a usuarios finales, organizaciones o dispositivos. El **encadenamiento de certificados** combina todos los certificados desde la raíz hasta el certificado final.

12.3 Solicitud de certificados (CSR)

Proceso: 1) se genera un par de claves pública/privada (ej. con OpenSSL); 2) se crea una **solicitud de firma de certificado (CSR)** con formato PKCS#10, que incluye la clave pública (nunca la privada) y datos del solicitante; 3) la CA valida la identidad y emite el certificado con la clave pública embebida. Una **autoridad de registro (RA)** puede ayudar a recolectar esta información, pero nunca emite certificados por sí misma.

■ Recordá esto!

Normalmente se solicita un certificado mediante una CSR. El primer paso es crear la clave privada basada en RSA, que se usa para generar la clave pública. Luego se incluye la clave pública en la CSR y la CA la embebe en el certificado. La clave privada nunca se envía a la CA.

12.4 Actualización y revocación

Los certificados vencen según sus fechas de validez y se pueden renovar. Se **revocan** antes de vencer si, por ejemplo, se filtra la clave privada o se compromete la CA.

- **CRL (Certificate Revocation List):** lista pública de números de serie revocados; se descarga y cachea, por lo que puede quedar desactualizada.
- **OCSP (Online Certificate Status Protocol):** consulta en tiempo real a la CA (respuesta: good/revoked/unknown) — más actualizado, pero genera más tráfico.
- **OCSP stapling:** el propio servidor 'grapa' una respuesta OCSP firmada y con sello de tiempo al certificado durante el handshake, evitando que el cliente consulte directamente a la CA.

■ Recordá esto!

Las CAs revocan certificados por varias razones, como el compromiso de la clave privada o de la propia CA. La CRL incluye una lista de certificados revocados y es de acceso público. Como alternativa a la CRL está el OCSP, que devuelve respuestas como good, revoked o unknown.

Certificate pinning: el servidor envía una lista de hashes de claves públicas válidas; el cliente las recuerda y las compara en conexiones futuras, detectando intentos de suplantación con certificados fraudulentos.

Key escrow: copia de seguridad de una clave privada guardada en un lugar seguro para recuperación (ej. un agente de recuperación de claves). **KMS (Key Management System):** sistema centralizado que gestiona todo el ciclo de vida de las claves — generación, almacenamiento (a menudo en HSMs), distribución, rotación y destrucción.

13. Tipos y Formatos de Certificados

Tipo	Uso
Máquina/computadora	Identifica un dispositivo dentro de un dominio.
Usuario	Cifrado, autenticación, tarjetas inteligentes.
Email	Cifrado y firma digital de correos.
Firma de código (code signing)	Verifica que un script o ejecutable no fue modificado.
Autofirmado (self-signed)	No emitido por una CA confiable; usado por CAs privadas internas.
Raíz (root)	El certificado de más alto nivel de la CA raíz.
Wildcard	Empieza con *; sirve para múltiples subdominios del mismo dominio raíz.
SAN (Subject Alternative Name)	Sirve para múltiples dominios distintos de la misma organización.
Validación de dominio (DV)	Confirma que el solicitante controla el dominio DNS.
Validación extendida (EV)	Pasos adicionales de verificación más allá de DV.

13.1 Formatos de archivo

Formato	Tipo	Uso típico
CER	ASCII	Formato base ASCII para certificados.
DER	Binario	Formato base binario para certificados.
PEM	ASCII (con headers)	El más usado; sirve para casi cualquier tipo de certificado.
P7B (PKCS#7)	ASCII base64	Comparte claves públicas; nunca incluye la clave privada.
P12 / PFX (PKCS#12)	Binario	Contiene la clave privada; suele estar cifrado.

■ Recordá esto!

CER es un formato ASCII y DER es binario. PEM es el formato de certificado más usado y sirve para casi cualquier tipo. Los archivos P7B se usan comúnmente para compartir claves públicas. Los archivos P12 y PFX se usan comúnmente para contener la clave privada.

14. Glosario de Términos Clave (Inglés ↔ Español)

Inglés	Español	Definición breve
Hashing	Hashing / función hash	Genera una cadena fija a partir de datos, para verificar integridad.
Hash collision	Colisión de hash	Dos entradas distintas producen el mismo hash.
Salting	Salteo (salting)	Agregar caracteres aleatorios a una contraseña antes de aplicar el hash.
Key stretching	Estiramiento de clave	Aplicar el hash repetidamente para dificultar ataques de fuerza bruta.
Symmetric / asymmetric encryption	Cifrado simétrico / asimétrico	Misma clave para cifrar/descifrar / par de claves pública-privada.
Public key / private key	Clave pública / privada	Par de claves usadas en cifrado asimétrico.
Digital signature	Firma digital	Hash cifrado con la clave privada del remitente; da autenticación, integridad y no repudio.
Non-repudiation	No repudio	Impide que alguien niegue haber realizado una acción.
Steganography	Esteganografía	Ocultar datos dentro de otros archivos.
Tokenization	Tokenización	Reemplazar datos sensibles con tokens no sensibles.
Masking	Enmascaramiento (masking)	Ocultar parcial o totalmente datos sensibles.
Certificate Authority (CA)	Autoridad certificadora	Emite, gestiona y revoca certificados digitales.
Root of trust	Raíz de confianza	Base de confianza que sostiene la validez de los certificados.
Certificate Signing Request (CSR)	Solicitud de firma de certificado	Solicitud enviada a una CA para obtener un certificado.
Certificate Revocation List (CRL)	Lista de revocación de certificados	Lista pública de certificados revocados.

Inglés	Español	Definición breve
OCSP	OCSP	Protocolo de verificación de certificados en tiempo real.
Key escrow	Custodia de claves (key escrow)	Copia de respaldo de una clave privada para recuperación.
Perfect forward secrecy	Secreto perfecto hacia adelante	Comprometer una clave no compromete sesiones pasadas.
Wildcard / SAN certificate	Certificado wildcard / SAN	Cubre múltiples subdominios / múltiples dominios distintos.
Downgrade attack	Ataque de degradación	Fuerza el uso de un protocolo de seguridad más débil.
Blockchain	Cadena de bloques (blockchain)	Libro contable público, distribuido y descentralizado.

15. Preguntas de Práctica (Autoevaluación)

Preguntas de elaboración propia, en español, para repasar los conceptos del capítulo 10. Las respuestas están al final.

1. Una empresa publica un archivo de actualización junto con su hash SHA-3 en el sitio web. Un usuario lo descarga y calcula el hash del archivo descargado. ¿Qué está verificando si ambos hashes coinciden?

2. Un atacante intercepta un mensaje y logra recalcular el hash MD5 tras modificarlo, haciendo que la verificación de integridad parezca válida. ¿Qué técnica hubiera evitado este problema?

3. Un sistema detecta cientos de intentos fallidos de inicio de sesión, con una sola contraseña probada contra una larga lista de cuentas distintas, una por una. ¿Qué tipo de ataque es?

4. Una aplicación cifra un archivo grande de forma rápida y eficiente usando bloques de 128 bits. ¿Qué algoritmo es el más probable?

5. Lisa firma digitalmente un correo. ¿Qué clave usa para crear la firma, y qué clave usa Bart para verificarla?

6. Una empresa quiere cifrar datos en un dispositivo IoT con muy poca capacidad de procesamiento. ¿Qué método criptográfico asimétrico es el más adecuado?

7. Un navegador detecta que un servidor está forzando el uso de SSL en vez de TLS, y sospecha un posible ataque. ¿Cómo se llama este tipo de ataque?

8. Una organización necesita verificar en tiempo real si un certificado fue revocado, sin depender de una lista descargada que podría estar desactualizada. ¿Qué protocolo debería usar?

9. Una empresa pierde acceso a una clave privada crítica y no puede recuperar datos cifrados con ella. ¿Qué práctica de PKI habría evitado esta pérdida?

10. Un mismo certificado necesita cubrir 'correo.empresa.com' y 'ventas.empresa.com', ambos subdominios del mismo dominio raíz. ¿Qué tipo de certificado es el más adecuado?

Respuestas

1. La integridad del archivo — que no fue modificado ni corrompido durante la descarga.

2. HMAC — al usar una clave secreta compartida, el atacante no podría recalcular un hash válido sin conocer esa clave.

3. Un ataque de password spraying — diseñado para evadir las políticas de bloqueo de cuenta.

4. AES (Advanced Encryption Standard) — cifra en bloques de 128 bits y es rápido y eficiente.

5. Lisa usa su clave privada para crear (firmar) el hash; Bart usa la clave pública de Lisa para descifrar la firma y verificarla.

6. Criptografía de curva elíptica (ECC) — requiere mucho menos procesamiento que otros métodos asimétricos.

7. Un ataque de downgrade (degradación).

8. OCSP (Online Certificate Status Protocol) — da respuestas en tiempo real sobre el estado del certificado.

9. Key escrow (custodia de claves) — mantener una copia segura de la clave privada para recuperación.

10. Un certificado wildcard (ej. *.empresa.com), ya que ambos son subdominios del mismo dominio raíz.
