Investigación sobre el protocolo HTTPS

---

1. Breve historia y evolución

HTTPS (HyperText Transfer Protocol Secure) es la versión segura del protocolo HTTP, desarrollado inicialmente por Tim Berners-Lee en 1991 para la transferencia de información en la web.

- **1994:** Netscape introdujo HTTPS como combinación de HTTP con SSL (Secure Sockets Layer) para cifrar la comunicación entre navegadores y servidores.
- **2000s:** Evolución de SSL a TLS (Transport Layer Security), mejorando la seguridad y la integridad de los datos transmitidos.
- **2014-2020:** Google y otros grandes actores impulsan el HTTPS como estándar de facto, promoviendo certificados gratuitos y cifrado obligatorio en sitios web.

Hoy en día, HTTPS es esencial para proteger la privacidad de los usuarios, prevenir ataques de intermediarios (MITM) y garantizar la autenticidad de los sitios web.

---

2. Funcionamiento y uso en desarrollo web

HTTPS funciona sobre el protocolo TCP y utiliza TLS para cifrar la comunicación. Su flujo básico es el siguiente:

1. El cliente (navegador) solicita una conexión segura al servidor.
2. El servidor responde con un **certificado digital** que contiene su clave pública.
3. El cliente verifica el certificado con una **autoridad de certificación (CA)**.
4. Ambos establecen una **clave de sesión compartida** mediante un proceso de intercambio seguro.
5. Toda la información intercambiada se cifra con esa clave, garantizando confidencialidad e integridad.

Usos en desarrollo web: - Protege formularios de registro, login y pagos online. - Garantiza que las APIs y servicios web transmitan datos de forma segura. - Mejora la confianza del usuario y el posicionamiento en buscadores (SEO).

---

3. Ventajas y posibles riesgos de seguridad

Ventajas	Riesgos de seguridad
Cifrado de extremo a extremo de datos	Certificados caducados o mal configurados
Autenticación del servidor	Ataques de intermediario si el cliente no verifica correctamente el certificado
Integridad de datos	Vulnerabilidades en versiones antiguas de TLS/SSL
Mejora la confianza del usuario	Phishing si se usan certificados válidos en sitios maliciosos

---

4. Ejemplos prácticos de uso

a) Navegación segura Acceder a un sitio web con HTTPS (`https://www.ejemplo.com`) muestra un candado en el navegador, indicando que la conexión es segura.

b) Prueba con `curl` en la terminal

```
curl -I https://www.google.com
```

Salida esperada:

```
HTTP/2 200
date: Fri, 26 Sep 2025 12:00:00 GMT
content-type: text/html; charset=ISO-8859-1
...
```

Esto demuestra que la página responde sobre HTTPS correctamente.

c) Implementación en un servidor web (Apache)

```
<VirtualHost *:443>
    ServerName www.ejemplo.com
    DocumentRoot /var/www/html
    SSLEngine on
    SSLCertificateFile /etc/ssl/certs/ejemplo.crt
    SSLCertificateKeyFile /etc/ssl/private/ejemplo.key
</VirtualHost>
```

Con esta configuración, el sitio sirve contenido de forma segura con HTTPS.

---

5. Recursos adicionales

- RFC 2818 – HTTP over TLS
- Let's Encrypt – Certificados TLS gratuitos
- OWASP – Guía de seguridad HTTPS

---

6. Reflexión personal

HTTPS no solo protege la información, sino que se ha convertido en un estándar ético y técnico en la web moderna. Su correcta implementación refleja la responsabilidad del desarrollador y la importancia de priorizar la privacidad y seguridad del usuario.