Módulo 3: Defensas, controles y contramedidas

Lección 1: Autenticación y Gestión de Identidades

Objetivos de la Lección

Al finalizar, el estudiante podrá:

1. **Comparar** los factores de autenticación “algo que sabes, que tienes o que eres”, evaluando fortalezas y debilidades.
2. **Diseñar** un esquema de autenticación multifactor (MFA) apropiado para distintos escenarios de riesgo.
3. **Evaluar** las tendencias actuales en gestión de identidades (SSO (**Single Sign-On**), gestores de contraseñas, cookies y passkeys) y su impacto en la privacidad y la experiencia de usuario.

Introducción a la Lección

La **autenticación** es la primera línea de defensa: confirma que quien intenta acceder a un recurso es realmente quien dice ser. Cuando esa verificación falla—por contraseñas filtradas o biometría falsificada—, se abren las puertas a ataques de escalado de privilegios, fraude y robo de datos. En esta lección profundizaremos en los factores de autenticación, los errores más comunes (falsos positivos/negativos), la lógica de un buen **MFA** y las tendencias que están transformando la gestión de identidades.

Desarrollo del Tema

Factores de autenticación: “algo que sabes, que tienes o que eres”

Más Información: https://support.microsoft.com/en-us/topic/what-is-multifactor-authentication-e5e39437-121c-be60-d123-eda06bddf661 y en https://csrc.nist.gov/glossary/term/Multi_Factor_Authentication

1. Algo que sabes — factor de conocimiento

- **Definición:** credencial memorizable conocida exclusivamente por el legítimo usuario.
- **Ejemplos concretos:**
    - Contraseña con *passphrase* de 14+ caracteres.
    - PIN de 6 dígitos en cajeros.
    - Respuesta a “Nombre de tu primer tutor” (pero esto puede ser una mala práctica si la respuesta es pública).
- **Buenas prácticas técnicas:**
    - Política de *length* > *complexity*:
        - **¿Qué significa?**: priorizar la **longitud de la contraseña** sobre la mezcla artificial de caracteres.
            - Ejemplo: *UnaFraseDeContraseñaSegura2025!* es más segura y fácil de recordar que *P@5s!*.
        - **Recomendación**:
            - Usar **frases largas y únicas** (mínimo 8–16 caracteres).
            - Evitar la **rotación forzada frecuente**, porque induce patrones inseguros (*Password2025*, *Password2026*).
        - Basado en guías de **NIST SP 800-63B**.
    - Hash + sal + *pepper* en el servidor

- **Hashing:** aplicar una función unidireccional sobre la contraseña antes de guardarla. Ej: Argon2id, recomendado sobre SHA-256 o MD5.
- **Salt:** valor aleatorio único (≥ 32 bytes) añadido a cada contraseña antes de aplicar Hashing, para evitar ataques con tablas rainbow.
- **Pepper:** secreto adicional (guardado fuera de la BD, ej. en un HSM o variable segura del servidor) que se combina con la contraseña antes del hash → agrega otra capa en caso de fuga de la base de datos.
- **¿Qué es Argon2id?**
    - Argon2id es un algoritmo moderno de hash de contraseñas, diseñado para ser seguro contra ataques de fuerza bruta y cracking con GPU/ASIC. En adicción combina lo mejor de **Argon2i** (resistente a ataques de canal lateral) y **Argon2d** (resistente a ataques con GPU).
    - Ejemplo de **Argon2id** en **Python**

```
from argon2 import PasswordHasher
import os       #se utiliza os para generar valores aleatórios seguros

# Generamos un 'salt' de 32 bytes con un CSPRNG (Un CSPRNG (Cryptographically Secure Pseudo-Random Number Generator) es un generador de números pseudoaleatorios diseñado para ser seguro en contextos criptográficos.) del sistema operativo.
# Nota: con PasswordHasher, argon2-cffi maneja el salt internamente; aquí lo creamos
# solo para ilustrar el concepto de 'salt' único por contraseña.

.salt = os.urandom(32)

# Inicializamos el hasher Argon2id con parámetros "fuertes".
```

```
# - time_cost: número de iteraciones (a mayor valor, más costoso para un atacante).
# - memory_cost: memoria usada en KB (64 MB). Alta memoria dificulta ataques con GPU/ASIC.
# - parallelism: hilos usados en paralelo (aprovecha CPUs modernas).

ph = PasswordHasher(
    time_cost=100000,   # iteraciones
    memory_cost=65536,  # memoria en KB (~65 MB)
    parallelism=4       # paralelismo (número de hilos)
)

# Contraseña del usuario. Recomendado: frase larga y única (length > complexity).

password = "UnaFraseSegura2025!"

# 'Pepper' = secreto global del servidor (no se guarda en la base de datos).
# Se concatena a la contraseña antes de hashear para agregar una capa extra de defensa
# en caso de fuga de la base de datos. Debe almacenarse en un HSM (Un HSM (Hardware Security Module) es un dispositivo físico especializado que genera, protege y gestiona claves criptográficas.) o en variables seguras.

pepper = "claveSecretaServidor"

# Combinamos la contraseña con el 'pepper'.

password_final = password + pepper
```

```
# Calculamos el hash Argon2id. El valor devuelto incluye metadatos
(parámetros, salt, hash).
# Este es el valor que se guarda en la base de datos (NUNCA la
contraseña en texto claro).
hash_guardado = ph.hash(password_final)

# Mostramos el hash resultante (por ejemplo:
$argon2id$v=19$m=65536,t=100000,p=4$...)

print(hash_guardado)
```

- **¿Qué hace este programa?**

- Este script hashea una contraseña usando Argon2id (vía argon2-cffi) aplicando la buena práctica de agregar un “pepper” (secreto del servidor). Genera también un salt de 32 bytes (a modo ilustrativo; argon2-cffi maneja el salt internamente cuando usamos PasswordHasher). El resultado es un hash seguro que es lo que deberías guardar en la base de datos, nunca la contraseña en texto claro.

- **Errores habituales:** lluvia de preguntas de reto fáciles de responder vía OSINT, uso de recordatorios “¿Olvidaste tu contraseña?” que exponen correos secundarios inseguros.

2. Algo que tienes — factor de posesión

- **Definición:** objeto físico o digital que solo el usuario debería controlar.

- Esto puede ser lo que sería un token de autenticación.

    - **¿Qué es un Token?**

        - Un **token de autenticación** es un **dispositivo o software** que genera o almacenas credenciales únicas para validar la identidad del usuario.

        - No depende solo de memoria (como contraseñas), sino de tener físicamente el objeto.
        - Se usa mucho en MFA (Multi-Factor Authentication) junto con algo que sabes o eres.
        - **Ejemplo:** Dispositivo que genera códigos de acceso de 6 dígitos cada 30 segundos, esto como “Microsoft Authenticator”.
- **Sub-categorías:**
    - **Token OTP desconectado** (hardware fob como RSA SecurID).
        - **Definición:** dispositivo físico (tipo llavero) que genera códigos de un solo uso (OTP) en intervalos de tiempo. No necesita internet ni estar conectado al servidor.
        - **Ejemplo: RSA SecurID** → muestra un número que cambia cada 60 segundos y sirve para autenticarse en portales corporativos o bancos.
    - **Token OTP software** (TOTP/HOTP en app móvil).
        - **Definición:** aplicación que genera códigos OTP en el celular.
            - TOTP: basado en tiempo (cambia cada 30 s).
            - HOTP: basado en contador (cambia al solicitarlo).
        - **Ejemplo real: Google Authenticator**, **Microsoft Authenticator**, **Authy** = estos generan códigos de 6 dígitos que se ingresan al iniciar sesión en Gmail, Facebook o GitHub.
    - **Llave criptográfica** (FIDO2, PIV, smart-card con chip).
        - **Definición -** dispositivo físico (USB, NFC o tarjeta con chip) que almacena claves criptográficas de forma segura y permite autenticación fuerte sin necesidad de recordar contraseñas.
            - **Dicho sea de paso** que NFC (**Near Field Communication** o *Comunicación de Campo Cercano)* es una comunicación inalámbrica de corto alcance usado para pagos móviles, llaves electrónicas y Autenticación.

        - **Ejemplo:**
        - **YubiKey (FIDO2)** → usada para login en Google, GitHub, Microsoft.
        - **PIV (Personal Identity Verification) cards** en agencias gubernamentales de EE.UU.
        - **Smart-cards** con chip y PIN para acceso a redes corporativas.
    - **Certificado X.509 alimentado por HSM** (firma de correos o VPN).
        - **Definición:** certificado digital que valida identidad bajo el **estándar X.509**, pero almacenado en un HSM (Hardware Security Module) para máxima seguridad.
        - **Ejemplo real:**
            - Firma digital de correos electrónicos con S/MIME.
            - Autenticación en una VPN corporativa mediante certificados emitidos por la empresa.
- **Ataques típicos y mitigaciones:**
    - **SIM-Swap para robar SMS-OTP**
        - **Ataque:**
            - El atacante convence al operador móvil (por ingeniería social o corrupción interna) de transferir tu número telefónico a otra SIM.
            - Así, recibe tus **códigos OTP enviados por SMS** y puede entrar a tus cuentas.
        - **Mitigación:**
            - **Evitar SMS-OTP** como único factor (es considerado inseguro por NIST y OWASP).
            - Migrar a **TOTP por app autenticadora** o **notificación push** en móvil (ej. Microsoft Authenticator).
            - Configurar el número con un **PIN de portabilidad** o clave adicional en el operador.
        - **Ejemplo Real:** Ataques de SIM-swap muy usados en fraudes financieros y robo de cuentas de criptomonedas.
    - **Duplicado de llave FIDO2**

        - Una **llave FIDO2** es un **dispositivo físico** de seguridad (tipo USB, NFC o Bluetooth) que permite autenticarse en servicios digitales sin necesidad de contraseñas o como segundo factor.
        - **Ataque**
            - Teóricamente se intentaría copiar la **clave privada** almacenada en la llave física (ej. YubiKey).
            - Sin embargo, **es prácticamente imposible**, ya que la clave privada **nunca sale del chip** ni se exporta.
            - Además, para usarla puede requerir **PIN local** o **biometría (huella/rostro)** en el dispositivo.
        - **Mitigación**
            - El propio diseño de **FIDO2/WebAuthn** protege contra duplicación.
            - Se recomienda tener **llaves de respaldo** registradas para recuperación de cuenta (en caso de pérdida física).
        - **Ejemplo Real**
            - Empresas como Google y Microsoft exigen llaves FIDO2 a sus empleados para mitigar phishing y robo de credenciales.

3. Algo que eres — factor inherente/biométrico

- **Definición:** rasgos físicos o conductuales estadísticamente únicos.
- **Tipos y especificidad:**
    - **Huella digital**
        - **Definición:** patrón único de crestas y valles en los dedos.
        - **Riesgo:** puede ser engañada con moldes de silicona si no hay detección de vitalidad (liveness detection).
        - **Ejemplos:**

            - Touch ID (Apple).
            - Sensores en laptops corporativas (ThinkPad, Dell, HP).
    - **Iris/Retina**
        - **Definición:** escaneo de patrones únicos en el iris o la retina (vasos sanguíneos).
        - **Limitaciones:** puede fallar con cambios de iluminación o usuarios con problemas oculares.
        - **Ejemplos:**
            - Escáner de iris en Samsung Galaxy S8/S9.
            - Sistemas biométricos de aeropuertos internacionales.
    - **Reconocimiento facial 3-D con proyección de puntos (Face ID)**
        - **Definición:** sistema que proyecta y analiza un patrón de puntos infrarrojos en el rostro para generar un mapa 3D.
        - **Riesgo:** la Tasa de Falsos Rechazos (FRR) aumenta si el usuario usa mascarilla, gafas oscuras o barba nueva.
        - **Ejemplos:**
            - Apple Face ID (iPhone, iPad Pro).
            - Windows Hello con cámara infrarroja.
- **Reglas de privacidad:** GDPR art. 9 exige base legal explícita y medidas “state-of-the-art” para proteger biometría; se recomienda almacenarla localmente y derivar *templates* irreversibles.

Imagen de estos tres elementos de autenticación

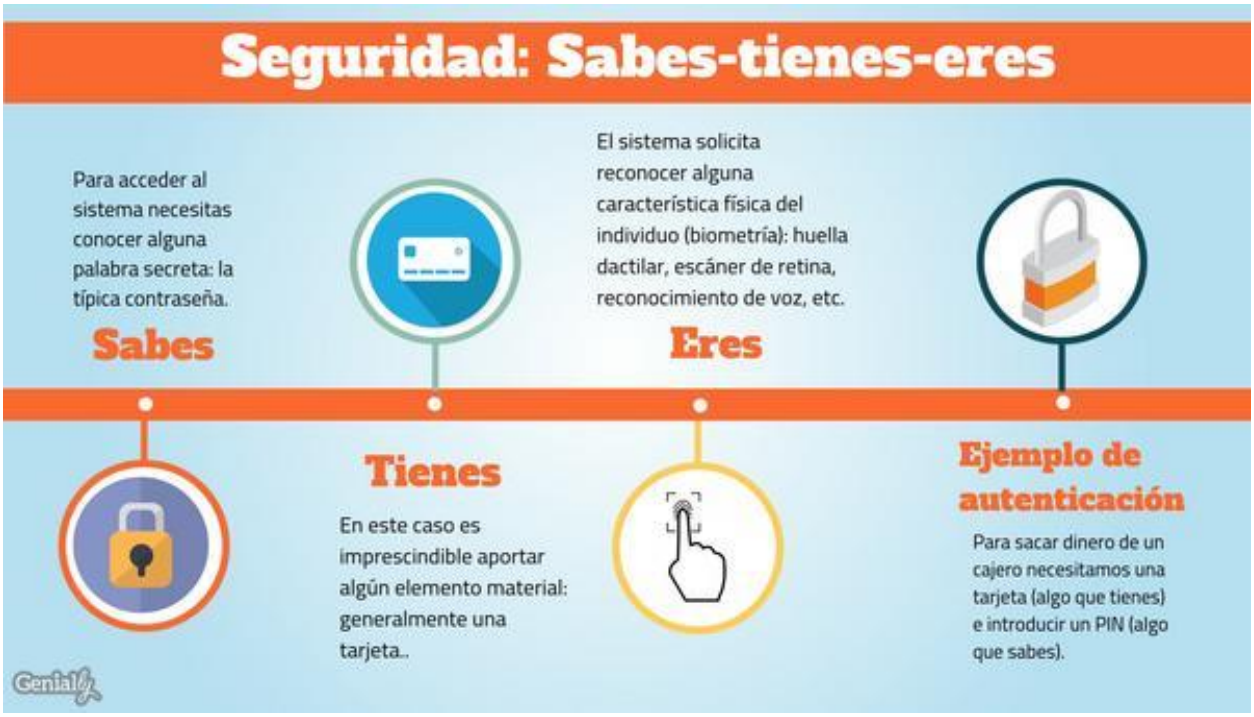

Imagen1: Algo que sabes, tienes y eres, https://view.genially.com/57fc8beeb6c0375ecc3e5b05/interactive-content-infografia-sabes-tienes-eres

Falsos positivos y falsos negativos

- **Falso positivo**
    1. **Definición completa:** evento en que un sistema acepta como legítimo a un impostor; medido por **FAR (False Acceptance Rate)**.
    2. **Consecuencias:** escalado de privilegios, fraude.
    3. **Caso real:** sensor de huella no muy costoso desbloquea teléfono con huella parcial de otra persona (FAR elevado).
- **Falso negativo**
    1. **Definición:** usuario legítimo es rechazado; cuantificado por **FRR (False Rejection Rate)**.
    2. **Impacto:** soporte costoso, abandono de servicio.
    3. **Caso real:** lector facial falla con cambios de peinado -> FRR alto obliga a fallback inseguro (PIN de 4 dígitos).

- **Equal Error Rate (EER)**
    1. **Definición:**
        - Es el punto en el que FAR = FRR.
        - Sirve como métrica global de precisión de un sistema biométrico.
        - Mientras más bajo es el EER, mejor es el equilibrio entre seguridad y usabilidad.
    2. **Ejemplo Numérico Simple**
        - Sistema A → EER = 5% (acepta a muchos impostores y rechaza a usuarios legítimos).
        - Sistema B → EER = 0.2% (mucho más preciso y confiable).
- **Parámetros de calibración biométrica:**
    1. **Score threshold** – bajar umbral reduce FRR pero eleva FAR.
    2. **Temporización** – limitar ventanas de autenticación a 10 s reduce ataques de replay.

Autenticación multifactor (MFA)

1. ¿Qué es MFA?

- El **MFA** consiste en combinar **dos o más factores de autenticación** (algo que sabes, tienes o eres) para aumentar la seguridad. **Más información:** https://www.fortinet.com/lat/resources/cyberglossary/multi-factor-authentication

2. Imagen de MFA

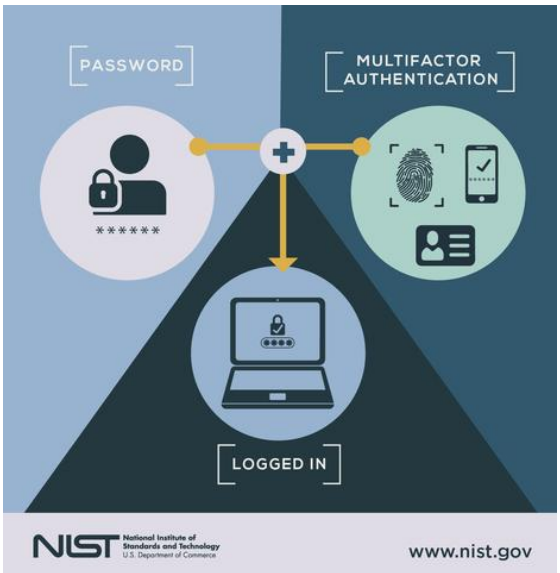

Imagen 2: Autenticación de MultiFactor, NIST (2025)

3. **Análisis de amenazas y contexto**
    A. **Nivel de garantía LOA/AAL según NIST SP 800-63B**
        - El NIST define **AAL (Authenticator Assurance Level)** como niveles de confianza en la autenticación:

Nivel	Significado	Ejemplo
AAL1 (bajo)	Un solo factor, poca resistencia a ataques.	Login con solo usuario + contraseña.
AAL2 (medio)	Requiere **MFA** (dos factores), protege contra robo básico de credenciales.	Contraseña + OTP por app móvil.
AAL3 (alto)	MFA con **dispositivos criptográficos o biometría fuerte**, resistente a ataques avanzados.	Llave FIDO2 con PIN biométrico + contraseña.

    B. **Mapeo a nivel de daño**
        - La elección del nivel depende de qué tan grave sería el impacto si hay una intrusión:
            - **Lectura de correos corporativos → AAL2 (medio):**
                a. Amenaza: acceso no autorizado a información interna.
                b. Mitigación: contraseña + TOTP (app móvil).
            - **Firma de pagos > $10 000 → AAL3 (alto):**
                a. Amenaza: fraude financiero significativo.
                b. Mitigación: autenticación con llave criptográfica (FIDO2 o smart-card) + biometría/PIN.
4. **Elección de combinaciones**
    - **Contraseña + push TOTP** = cobertura AAL2 (riesgo medio).

- **Llave FIDO2 + reconocimiento facial on-device** = AAL3 (máxima).
- **Passwordless Authentication**
  - **Passwordless** significa autenticarse **sin usar una contraseña tradicional** (algo que recuerdas), reemplazándola por factores más fuertes como **dispositivos criptográficos y biometría**.

5. **Arquitectura de flujos**
   - **Secuencial "All-or-Nothing"**
     - **Definición:** el sistema exige **todos los factores, en un orden predefinido**, para conceder acceso.
     - **Uso típico:** banca en línea o portales de alta seguridad.
     - **Ejemplo:**
       - Usuario ingresa **contraseña**.
       - Luego debe ingresar **OTP de app móvil**.
       - Finalmente, el sistema pide **biometría** si aplica.
   - **Step-Up**
     - **Definición:** el sistema pide **factores adicionales solo cuando detecta riesgo**.
     - **Mecanismos de detección:**
       - IP desconocida.
       - Intento de login desde otro país.
       - Transacción superior a cierto monto.
     - **Ejemplo:**
       - Acceso normal al correo desde tu PC habitual → solo pide contraseña.
       - Intento de transferir **$20,000** desde la misma cuenta → sistema exige OTP o llave FIDO2 adicional.
6. **Gestión de excepciones**
   - **Perdida de token físico:** proceso de verificación de identidad en persona o a través de vídeo + documento oficial.
7. **Experiencia de usuario y accesibilidad**
   - Siempre ofrecer códigos de recuperación offline.

Privacidad, anonimato, pseudónimos e identidades múltiples

1. **Privacidad**
    - **Principio de minimización**
        - **Definición:** el proveedor de identidad (IdP, Identity Provider) solo debe compartir con el servicio destino (SP, Service Provider) los atributos mínimos necesarios para autorizar al usuario.
        - **Ejemplo:**
            - ❖ Un estudiante accede a la plataforma de la universidad.
            - ❖ En lugar de enviar nombre, apellido, fecha de nacimiento, correo, etc., el IdP puede emitir solo el student_id.
            - ❖ Así, el servicio reconoce al usuario sin exponer datos sensibles innecesarios.
        - **Beneficio:** reduce riesgo de fuga de información y protege la privacidad del usuario.
    - **Protocolos que aplican este principio**
        - **Idemix (IBM Identity Mixer):**
            - ❖ Permite que el usuario **demuestre atributos sin revelarlos todos**.
        - **Microsoft ION (Identity Overlay Network):**
            - ❖ Proyecto basado en **blockchain (DID, Decentralized Identifiers)**.
            - ❖ Implementa **credenciales verificables** con **divulgación selectiva**:

2. **Anonimato**
    - **Definición**
        - El **anonimato** busca que el usuario pueda autenticarse o usar un servicio **sin revelar su identidad real**.
        - Se centra en **ocultar datos personales**, permitiendo interacción sin rastreo directo de “quién” es el usuario.

- **Implementación práctica**
  - En las redes **Tor (.onion)** existe la autenticación a través de **"v3 Onion Services client authorization"**.
  - **¿Cómo funciona?**
    1. El cliente presenta un **hash de su clave pública** al servidor .onion.
    2. El servidor lo valida para permitir acceso.
    3. En ningún momento se comparten datos como nombre, correo o dirección IP → **solo claves criptográficas**.

3. **Pseudónimo permanente (pseudo-ID)**
   - **Definición**
     - Un **pseudo-ID** es una **identidad digital alternativa** que no expone directamente el nombre real del usuario.
     - Es **permanente** porque se reutiliza en múltiples interacciones, lo que permite construir historial y reputación.
   - **Usos Comunes**
     - Foros y comunidades online: usuarios con alias (ej. CyberWarrior92).
     - Criptomonedas: direcciones de Bitcoin o Ethereum (Criptomonedas).
     - Sistemas de e-voting: permiten identificar un voto válido sin revelar la identidad del votante.
4. **Identidades múltiples (role-based claims)**
   - **Definición**
     - Un mismo usuario puede tener **diferentes roles** o "identidades parciales" según el **contexto**.
     - Estos roles se expresan en los **claims (atributos)** que viajan en el token de autenticación.
       - **Ejemplo:** un profesor universitario que también está inscrito como estudiante de posgrado → la misma persona, pero con **roles distintos**.

- **Modelo técnico (OIDC + SCIM)**
  - **OIDC (OpenID Connect):** protocolo de autenticación que emite un **ID token** con claims (ej. role=student).
  - **SCIM (System for Cross-domain Identity Management):** estándar que permite **provisionar y sincronizar identidades y roles** entre sistemas.

Tendencias en gestión de identidades

1. **SSO (Single Sign-On)**
   A. **SSO (Single Sign-On)** es un mecanismo de autenticación que permite a un usuario acceder a **múltiples aplicaciones o servicios** utilizando **una sola credencial** (ej. una cuenta y contraseña, o un login con Google/Microsoft).
2. **Gestores de contraseñas**
   A. Los **gestores de contraseñas** son aplicaciones (ej. LastPass, Bitwarden, 1Password, KeePass) que almacenan de forma segura las credenciales del usuario y las sincronizan entre dispositivos.
   B. **Arquitectura de seguridad típica:**
      1. **Clave maestra local**
         - El usuario crea una contraseña maestra.
         - Esa contraseña nunca se guarda en la nube, se usa para derivar una clave de cifrado local.
      2. **Sincronización cifrada extremo a extremo (E2EE)**
         - Todas las contraseñas se cifran en el dispositivo antes de salir a la nube.
   C. **Funciones avanzadas**
      1. **Análisis de reutilización:** detecta si usas la misma contraseña en varias cuentas.
      2. **Auditoría de filtraciones:** integra servicios como Have I Been Pwned API para avisar si alguna de tus contraseñas aparece en bases de datos filtradas.

3. **Generador de contraseñas seguras:** crea claves largas y únicas para cada servicio.

3. **Cookies, sesiones y JWT**
   - **A. Seguridad de cookies**
     1. **SameSite=Strict** → la cookie solo se envía si la petición proviene del mismo dominio.
     2. **Secure (Seguro)** → solo se transmite en conexiones HTTPS → evita robo en redes inseguras.
     3. **HttpOnly** → la cookie no es accesible vía JavaScript → evita robo por **XSS (Cross-Site Scripting)**.
   - **B. Sesión**
     1. **Sesión (server-side):**
        - El servidor guarda la sesión en memoria o base de datos.
   - C. **JWT (JSON Web Token, stateless):**
     1. El token contiene claims firmados digitalmente.
4. **Autenticación adaptativa (Risk-Based Authentication, RBA)**
   - A. ¿Qué es **RBA**?
     1. La **RBA** es un tipo de autenticación inteligente que **adapta los controles de seguridad según el nivel de riesgo detectado en el login**.
   - **B. Señales que evalúa**
     1. **Reputación IP**
        - El sistema revisa si la IP desde donde te conectas está en listas negras de actividad maliciosa.
     2. **Geovelocidad**
        - Mide si los intentos de login desde distintas ubicaciones son **físicamente posibles**.
        - Ejemplo: te logueas en Madrid y 2 horas después en Nueva York → imposible viajar tan rápido → sospechoso.
     3. **Fingerprint de dispositivo**

- Analiza las características del dispositivo: navegador, sistema operativo, plugins, resolución de pantalla, etc.

4. **Tiempo de sesión**
    - Considera los horarios de acceso.
    - Ejemplo: si siempre entras de 8 a.m. a 6 p.m., pero ahora hay un login a las 3 a.m. → comportamiento anómalo.

Relación con Otros Conceptos

- Los factores de autenticación se conectan con los **controles técnicos** del Módulo C y con la **privacidad/no-repudio** que veremos en criptografía.
- Comprender falsos positivos/negativos ayuda a evaluar la **superficie de ataque** y alinear la sensibilidad biométrica con los riesgos identificados en el análisis de amenazas.

Resumen de la Lección

Examinamos los tres factores clásicos de autenticación y demostramos cómo los falsos positivos/negativos afectan su eficacia. Diseñamos un flujo MFA que equilibra seguridad con usabilidad y exploramos la compleja relación entre identidad, privacidad y roles múltiples. Finalmente, evaluamos tendencias como SSO, gestores de contraseñas, cookies seguras y passkeys, que apuntan hacia un futuro “passwordless” más robusto contra el phishing.

Actividad de la Lección

Laboratorio “Construye tu MFA” (90 min)

1. Forme equipos de cuatro y elija un contexto (e-commerce, clínica, universidad).
2. Diseñe un flujo MFA detallando: factores, backup y reset seguro.
3. Elabore un diagrama de secuencia (login → verificación → acceso).
4. Identifique posibles puntos de falla (FAR/FRR, pérdida de token) y proponga mitigaciones.

5. Entregue un PDF con diagrama y justificación (máx. 2 páginas).

Referencias Adicionales

- Fortinet. (s. f.). *Autenticación multifactor (MFA)*. Fortinet. https://www.fortinet.com/lat/resources/cyberglossary/multi-factor-authentication
- Genially. (2016). *Infografía: ¿Sabes? Tienes, eres*. https://view.genially.com/57fc8beeb6c0375ecc3e5b05/interactive-content-infografia-sabes-tienes-eres
- Microsoft. (2023). *What is multifactor authentication?* Microsoft Support. https://support.microsoft.com/en-us/topic/what-is-multifactor-authentication-e5e39437-121c-be60-d123-eda06bddf661
- National Institute of Standards and Technology. (s. f.). *Back to basics: Multi-factor authentication (MFA)*. NIST. https://www.nist.gov/itl/applied-cybersecurity/back-basics-multi-factor-authentication-mfa
- Pfleeger, C. P., Pfleeger, S. L., & Coles-Kemp, L. (2023). *Security in Computing* (6.ª ed.). Addison-Wesley.
- Stallings, W., & Brown, L. (2017). *Computer Security: Principles and Practice* (4.ª ed.). Pearson.
- NIST SP 800-63B (2023). *Digital Identity Guidelines – Authentication and Lifecycle Management*.
- FIDO Alliance. *WebAuthn & Passkeys Overview* (2024).
- OWASP. *Authentication Cheat Sheet* (2025).