FUNCIONAMIENTO DE ALGORITMOS

RSA (Rivest–Shamir–Adleman)

Tipo: Cifrado asimétrico.
Usos típicos: Encriptar claves, firmar digitalmente, comunicaciones seguras (HTTPS, por ejemplo).

Cómo funciona:

- Se basa en matemáticas de números primos grandes.
- Se generan dos claves:
  - **Pública:** se comparte.
  - **Privada:** se guarda en secreto.
- Lo que se cifra con una se descifra con la otra.
  - Ejemplo: tú publicas tu clave pública; cualquiera puede enviarte un mensaje cifrado, pero solo tú (con tu clave privada) puedes leerlo.
- La seguridad depende de lo difícil que es factorizar números enormes (romperlos en sus primos).

Ventajas: muy seguro, útil para intercambiar información sin compartir claves antes.
Desventajas: es lento, no sirve para cifrar archivos grandes directamente.

MD5 (Message Digest 5)

Tipo: Hash o resumen criptográfico (no cifrado).
Usos típicos: Verificar integridad de datos (que un archivo no cambió).

Cómo funciona:

- Toma **una entrada de cualquier tamaño** (texto, archivo, etc.).
- Produce **una salida de 128 bits** (32 caracteres hexadecimales).
- Ejemplo:
  "hola" → 4d186321c1a7f0f354b297e8914ab240
- Siempre que metas “hola”, da ese mismo resultado.
- Pero si cambias una letra, cambia totalmente el hash.

Importante:
Ya no es seguro para uso criptográfico (se pueden crear dos archivos distintos con el mismo hash → colisión).
Solo se usa hoy para verificación básica, no para proteger datos.

Base64 (B64)

Tipo: Codificación, no cifrado ni hash.
Usos típicos: Transmitir datos binarios (imágenes, archivos) como texto (por ejemplo, en correos o JSON).

Cómo funciona:

- Convierte los bytes en caracteres imprimibles (A–Z, a–z, 0–9, +, /).
- Divide los datos en bloques de 6 bits y los mapea a esos 64 símbolos.
- Ejemplo:
  Texto: "Hola" → SG9sYQ==
  (el "==" es relleno para completar bloques).
- Cualquiera puede decodificarlo, no hay clave.

No protege la información, solo la hace "transportable".