# 

[1. Algoritmos HASH 1](#_Toc56672044)

[1.1. Práctica: Integridad de archivos 2](#_Toc56672045)

[1.2. Algoritmos rotos 2](#_Toc56672046)

[2. Métodos de autenticación 3](#_Toc56672047)

[2.1. ¿Qué es la autenticación? 3](#_Toc56672048)

[2.2. Políticas de contraseñas 4](#_Toc56672049)

[2.2.1. Práctica: Política de contraseña 4](#_Toc56672050)

[2.3. Plataformas para la gestión de contraseñas 6](#_Toc56672051)

[Keepass: es un programa de gestión de contraseñas local; es gratuito y uno de los más usados. 6](#_Toc56672052)

[3. Certificado digital 6](#_Toc56672053)

[3.1. ¿Qué es un certificado digital? 6](#_Toc56672054)

[3.2. Infraestructura de Clave Pública (PKI) 7](#_Toc56672055)

[3.2.1. Elementos funcionales de una PKI: 7](#_Toc56672056)

[3.2.1.1. Básicos 7](#_Toc56672057)

[3.2.1.1.1. Autoridad de certificación(CA) 7](#_Toc56672058)

[3.2.1.1.2. Autoridad de registro(RA) 8](#_Toc56672059)

[3.2.1.1.3. Almacén de certificados y claves 8](#_Toc56672060)

[3.2.1.2. Opcionales 8](#_Toc56672061)

[3.2.1.2.1. Autoridad de sellado de tiempo(TSA) 8](#_Toc56672062)

[3.2.1.2.2. Servidor de revocación 8](#_Toc56672063)

[3.2.2. Ejemplo del procedimiento de emisión de un certificado 8](#_Toc56672064)

# 

# Algoritmos HASH

Los algoritmos hash se utilizan para garantizar la integridad de la información. Un algoritmo Hash coge una cadena, mensaje o fichero cualquiera, y calcula un número con una longitud determinada. En un algoritmo determinado la longitud del hash será siempre la misma, independientemente de cuál sea el tamaño de la entrada.

Los algoritmos **hash** son algoritmos de una única sentido; es decir: una vez que se obtiene la cadena final, no se puede revertir el proceso. Es precisamente por eso por lo que se utilizan para garantizar que aquello que se ha enviado entre el emisor y el receptor no se ha modificado en el transcurso de la comunicación, manteniéndose, por lo tanto, la integridad.

Algunos de los algoritmos **Hash** más conocidos son: MD5, SHA1, SHA-256, …

El hash md5 de estas cadenas es:

Esto es una prueba de concepto: 789757984e4e86c9c6c00c1047d03bc

Esto es otra prueba de concepto: 390a4362024be38457ffe7b32daa3bce

Como ves, aunque las cadenas de entrada sean similares, los hashes son totalmente diferentes.

Sin embargo, los algoritmos no están exentos de ser vulnerables o ser rotos. Imaginemos que se ha podido calcular el mismo hash pero con dos ficheros diferentes; a eso se le denomina colisión. En estos casos, los algoritmos se convierten en débiles ya que su propoósito ha sido alterado. Han sido rotos, y, por consiguiente, son no confiables.

#### 

## Práctica: Integridad de archivos

1. Descarga un archivo, por ejemplo de [aquí](https://code.visualstudio.com/download).
2. Busca el código en la página de descarga.
3. Instala 7ZIP.
4. Haz click en el botón derecho sobre el archivo descargado.
5. Pincha en: CRC-SHA.
6. Elige el sistema en el que aparezca el código en la página de descarga, y genera el código.
7. Copia los códigos de los pasos 2 y 6 en el Notepad+; hacemos doble click en uno de los dos; si son iguales, se iluminan ambos.



## Algoritmos rotos

Los algoritmos de MD5 y SHA1 son algoritmos rotos.

En febrero del 2017 ocurrió lo siguiente:



Romper el algoritmo MD5, hoy en día, le lleva a una computadora menos de 1 minuto; para romper el algoritmo SHA1, que tiene 9.223.372.036.854.775.808 ciclos, hace falta más tiempo:

* + Haciendo fuerza bruta, tardaría 12.000.000 de años con un solo procesador (GPU).
  + Utilizando un algoritmo específico para vulnerar SHA1, le llevaría 110 años a 1 procesador.

Sin embargo, Google utilizó 110 tarjetas para romperlo en 1 solo año. Es factible que empresas como Microsoft, IBM, Lenovo, Amazon, Google… e incluso algunos gobiernos puedan tener la capacidad de procesamiento de 110 tarjetas para hacer fuerza bruta y romper el algoritmo en un año; es por ello por lo que a medida que se desconfía de algoritmos ya rotos van implementándose otros nuevos. Si bien MD5 y SHA1 son vulnerables, estos ya están reemplazados por otros como SHA256 y SHA512 de la familia SHA2. El problema de esto es que también consumen más capacidad de procesamiento.

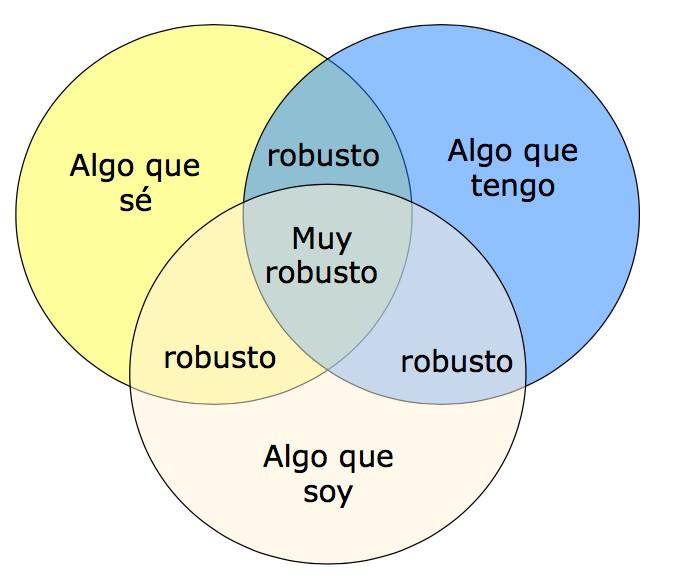
# Métodos de autenticación

## ¿Qué es la autenticación?

La autenticación es el proceso de confirmar que alguien es quien dice ser mediante la introducción de un identificador y de una contraseña que solo él sabe.

Existen varios métodos de autenticación:

* + - * Algo que sabemos, el PIN o la contraseña.
      * Algo que tenemos, tarjetas, token, fichero de llave, …
      * Algo que somos, huella biométrica digital, rostro, …



La combinación de diferentes métodos de autenticación mejora la seguridad de ésta. Es por eso por lo que plataformas como Google ofrecen ya hoy en día la autenticación mediante doble factor: la contraseña (algo que sé) y la activación mediante el móvil (algo que se tiene).

La autenticación por contraseña es el método más utilizado para el acceso a un sistema o servicio. Es necesario identificarse con el identificador del usuario y una contraseña para el acceso. Se trata de “algo que sabemos”.

A menudo se utilizan un identificativo y una contraseña muy fáciles de averiguar. Contraseñas como *12345678*, fechas de cumpleaños, nombres de mascotas… son las primeras contraseñas que un atacante utilizará para poder acceder al sistema o servicio. Como curiosidad, mira el [listado de las contraseñas más utilizadas](https://www.ncsc.gov.uk/static-assets/documents/PwnedPasswordsTop100k.txt) según el NCSC (Centro Nacional de Ciberseguridad del Reino Unido).

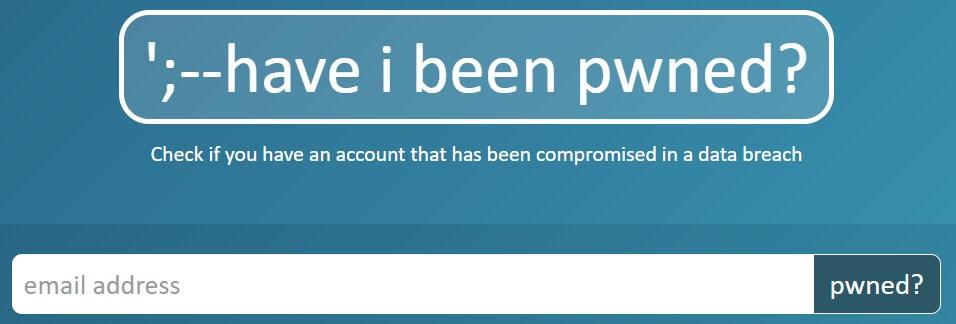
Para tener una contraseña robusta será necesario que al menos cumpla con las siguientes características:

* + **Complejidad**. Debe formarse por letras minúsculas y mayúsculas, números decimales y caracteres especiales.
  + **Longitud**. Debe tener al menos una longitud de 8 caracteres.
  + **Estructura**. No debe formar parte de una palabra de ningún diccionario.
  + **Caducidad**. No debe haberse usado con anterioridad.

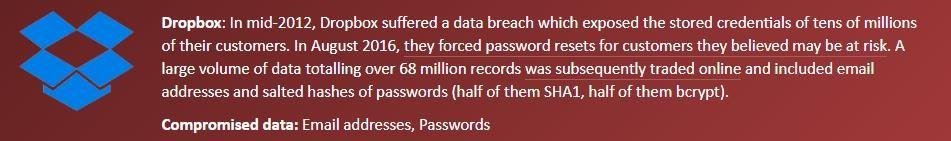


Actividad: 

Accede a <haveibeenpwned.com/> y comprueba si tu correo ha sido visto en algún diccionario.



Verás si tu cuenta ha sido hackeada y en qué servicios.



## Políticas de contraseñas

Si dejamos la gestión de las contraseñas en manos de los usuarios de una organización, la seguridad de ésta se verá claramente amenazada, ya que si las contraseñas son poco robustas el acceso a los equipos por parte de un intruso será muy fácil. Es por ello por lo que se hace necesario configurar e implantar una política de contraseñas que obligue a que la contraseña de cada usuario cumpla unos mínimos de seguridad.

### Práctica: Política de contraseñas

Para tener una política de contraseñas robusta, realiza los siguientes pasos.

1. Preparación e instalación de paquetes:

sudo dpkg --configure –a  
sudo apt-get update

sudo apt-get -y install libpam-pwquality cracklib-runtime

1. Cambia la fecha de renovación de contraseña editando el siguiente fichero: */etc/login.defs*

sudo nano /etc/login.defs

* + Cambia el parámetro PASS\_MAX\_DAYS a 30.

*PASS\_MAX\_DAYS 30*

1. Edita el siguiente fichero

sudo nano /etc/pam.d/common-password

Sustituye



*password requisite pam\_pwquality.so retry=3*

por lo siguiente:

password requisite pam\_pwquality.so retry=3 minlen=8 maxrepeat=3 ucredit=-1 lcredit=-1 dcredit=-1 ocredit=-1 difok=3 gecoscheck=1 reject\_username enforce\_for\_root

Los parámetros introducidos significan lo siguiente:

* + **retry=3**: Preguntar al usuario 3 veces por una contraseña válida.
  + **minlen=8** : Mínimo de caracteres en la contraseña.
  + **maxrepeat=3**: Los caracteres sólo pueden aparecer 3 veces consecutivamente.
  + **ucredit=-1** : Debe haber al menos un carácter en mayúscula.
  + **lcredit=-1** : Debe haber al menos un carácter en minúscula.
  + **dcredit=-1** : Debe haber al menos un digito.
  + **difok=3** : como mínimo tiene 3 caracteres diferentes a la contraseña anterior.
  + **gecoscheck=1**: No debe haber palabras que aparezcan en el nombre completo de la cuenta del usuario.
  + **reject\_username**: El nombre de usuario no puede estar en la contraseña ni tampoco a la inversa.
  + **enforce\_for\_root**: Aplica la política de contraseña también para el usuario root.

1. Guarda y reinicia.

$ sudo reboot

Una vez reiniciado el sistema, realiza la prueba con un nuevo usuario. Para ello

1. Crea un nuevo usuario

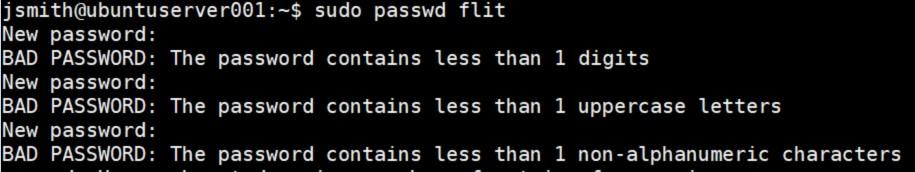
$ sudo useradd flit

1. Configura una contraseña para la nueva cuenta.

$ sudo passwd flit

1. Introduce diferentes passwords para comprobar los parámetros de complejidad configurados anteriormente: *contraseña*, *abc003*, *tilf944AS*

…



1. Finalmente, introduce una password válida; por ejemplo: Password123!



## Plataformas para la gestión de contraseñas

Hoy en día manejamos muchas cuentas online acerca de las que no recordamos si estábamos registrados o no y mucho menos de las contraseñas que hemos utilizado para acceder.



Existen varios programas para la gestión de contraseñas que permiten almacenar usuarios, contraseñas, URLs y algunas propiedades más. El objetivo es que nuestras contraseñas, al no tener que ser recordadas por nosotros, sean muy complejas. Ni siquiera es obligatorio que nosotros las pensemos, el propio software puede hacerlo. La única que sí tendremos que recordar es la maestra, y ésa tendrá que ser muy robusta. Hay dos tipos de gestores de contraseñas: locales y online. 

#### Keepass: es un programa de gestión de contraseñas local; es gratuito y uno de los más usados.

Si usas Keepass, tendrás que mover el archivo con las contraseñas en caso de querer usarlo en más de un dispositivo; poniéndolo en el Drive, por ejemplo. Ése es el lado malo de los gestores locales.

**1Password:** es un software online. [Vídeo promocional](https://www.youtube.com/watch?time_continue=112&amp;v=mcly2-b1W20)

El lado malo de los gestores online -*1password*, por ejemplo- es que te tienes que fiar de la empresa que te ofrece el servicio; ¿por qué no pueden ser atacados y ser tus contraseñas robadas?

# Certificado digital

## ¿Qué es un certificado digital?

En el cifrado asimétrico hemos visto que se utiliza un par de claves; la clave pública está accesible para cualquiera. Pero, ¿cómo podemos tener la certeza de que esa clave pública es de quien dice ser? Un certificado digital garantiza técnica y digitalmente la identidad en internet. Es un requisito a la hora de realizar peticiones o comunicaciones seguras.

## Infraestructura de Clave Pública (PKI)

PKI o Infraestructura de Clave Pública es una combinación de elementos que permiten cifrar, firmar y conseguir el no repudio de comunicaciones electrónicas.

### Elementos funcionales de una PKI:

### Básicos

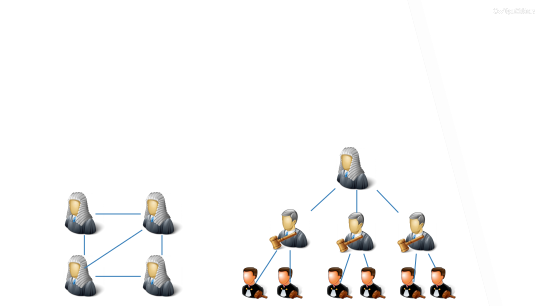
### Autoridad de certificación(CA)

Se trata de un tercero en quien se confía para: firmar y publicar certificados; revocar certificados y publicar la revocación; y recoger funciones de gestión y fechado. Las CAs pueden ser públicas o privadas (individuales).

Para firmar los certificados, lo hace con su clave privada. Su clave pública viene preinstalada en el S.O o navegador en algunos casos o será necesario descargarla e instalarla manualmente. Para confiar en su pública, se certifica a sí misma o la certifica otra CA.

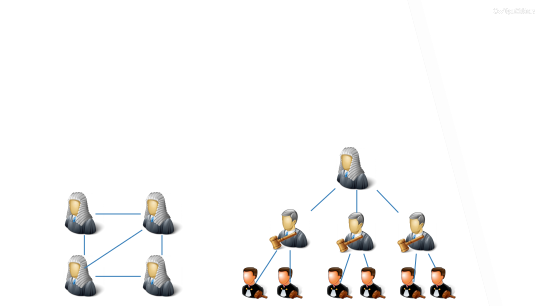
En cuanto a su organización, podemos dividir las CAs en las siguientes categorías:

* CA única: es fácil de mantener, pero es un punto vulnerable.
* Jerarquía de CAs: se establece un árbol de confianza en el que cada CA certifica a todas las CAs de su nivel inmediatamente inferior.



Se puede apreciar esta estructura en el certificado que nos muestran los sitios web en el navegador: “google.com”, por ejemplo.

* Malla de CA: Las CAs se certifican mutuamente, y cada sujeto confía en su CA.



He aquí un [listado](https://firmaelectronica.gob.es/Home/Empresas/Autoridades-Certificacion.html) de las principales autoridades de certificación. Como curiosidad, en esta [reciente noticia](https://elpais.com/politica/2019/11/05/actualidad/1572978141_937893.html) se aprecian algunos efectos del nuevo decreto según el cual el Gobierno Español podrá intervenir y a[sumir la “gestión directa” de redes y servicios de comunicaciones electrónicas](https://elpais.com/politica/2019/10/31/actualidad/1572509579_946126.html) en determinados casos excepcionales.

### Autoridad de registro(RA)

Es una autoridad delegada para la realización de algunas funciones:

* Recibir solicitudes de certificados.
* Generar las claves.
* Verificar la identidad del solicitante.
* Entregar el certificado al solicitante.

### Almacén de certificados y claves

Los certificados son públicos y deben estar siempre disponibles y ser guardados en un registro histórico. De esta manera, se podrá comprobar si están o no revocados. Ahora bien, las claves privadas asociadas sí deben estar protegidas, y, si se ven comprometidas, se revocará el certificado.

### Opcionales

### Autoridad de sellado de tiempo(TSA)

La TSA proporciona sellos de tiempo. Para ello, necesita que se registre el fechado; así, se asocia la huella digital de un documento a una fecha y hora concreta, permitiendo el no repudio.

- Necesita que se registre el fechado

- Documento que asocia la huella digital de un

documento a una fecha y hora concreta

- Permite el no repudio

### Servidor de revocación

Lugar donde se almacena la CRL (Lista de Revocación de Certificados). Lista de números de serie que han sido revocados y que, por lo tanto, ya no son válidos; ningún usuario del sistema deberá confiar en ellos.

### Ejemplo del procedimiento de emisión de un certificado

Supongamos que una persona quiere tener un certificado digital, y decide tramitarlo mediante la Fábrica Nacional de Moneda y Timbre, que, como hemos visto, es una de las principales autoridades de certificación

