

Identifica els tipus de credencials més utilitzats

Índex

- [Usuari i contrasenya](#)
- [Tokens de sessió](#)
- [Tokens d'autenticació](#)
- [Claus Asimètriques](#)
- [Certificats digitals a partir de claus asimètriques.](#)
- [Bibliografia](#)

Usuari i contrasenya

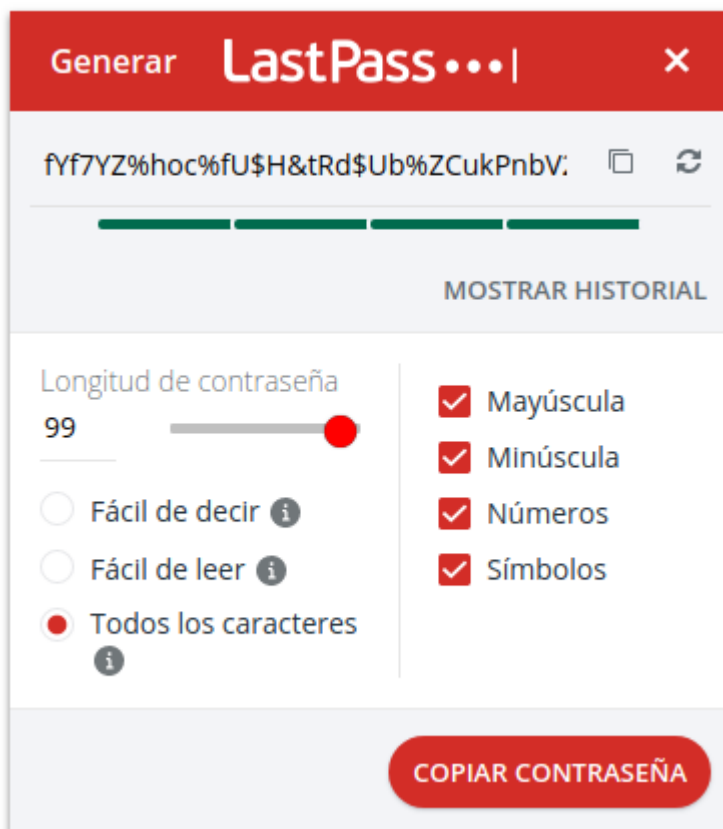
User:

tperaira

Password:

FYf7YZ%hoc%fU\$H&tRd\$Ub%ZCukPnbV240bcWP^hJjIPDP9FpxCsS!98s\$NmCIXQV#H#VB2iuQ1TY2
2ztLTNpGy#vLb2wh5hRjX

Contrasenya generada amb el generador de LastPass.



L'usuari és simplement l'inicial del meu nom i el meu primer cognom, ho veig idoni per identificar un usuari i és molt simple de recordar.

He triat generar una contrasenya amb la màxima longitud permesa pel generador perquè contra més llarga i complex millor. Com utilitzo un gestor de contrasenyes no haig de fer l'esforç de recordar el password i es fa impossible de petar mitjançant diccionaris i força bruta. Únicament haig de recordar la meua contrasenya mestra.

Tokens de sessió

Token de sessió mitjançant JWT.

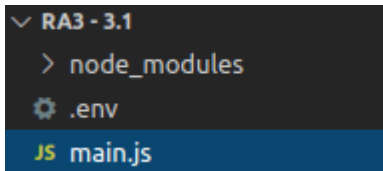
Token de sessió obtingut:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VyX2lkIjozLCJ1c2VyX25pY2tuYW1lIjoidHBldmFpcmEiLCJ1c2VyX21haWwiOiJhMjEwMjlyNkIjYmJlbnN0aXR1dG1vbnRpbGl2aS5jYXQiLCJpYXQiOiJlY2MzUyNzE4MjQsImV4cCI6MTY0OTY3MTgyNH0.8xERi829ABDUVRxQyaOGNixP8kYXDeRBbG0cJ3fqDxw
```

Dades descriptades:

```
{
  "user_id": 1,
  "user_nickname": "tperaira",
  "user_mail": "a2102226@institutmontilivi.cat",
  "iat": 1635271824,
  "exp": 1649671824
}
```

He generat el token amb JavaScript, creant els següents fitxers:



1. .env: Inclou el secret.

```
TOKEN_SECRET=1ae3dd89ffa39029cac5b5ef63cbe2e29dc5c94e324bc7fd827774b81412fdd8a52317ce49b94b520e03efef858fed2115f1ef17d6420898f83450bf029a75
```

He generat el secret amb la comanda:

```
require('crypto').randomBytes(64).toString('hex')
```

2. main.js: Inclou el codi que genera l'access token i el llegeix.

```
const jwt = require('jsonwebtoken');
const dotenv = require('dotenv');
```

```
// Creació del TOKEN_SECRET
//console.log(require('crypto').randomBytes(64).toString('hex'));

// get config vars
dotenv.config();

// access config var
process.env.TOKEN_SECRET;

let token = generateAccessToken({
  user_id: 1,
  user_nickname: "tperaira",
  user_mail: "a2102226@institutmontilivi.cat"
});
console.log(token);

var decoded = jwt.verify(token, process.env.TOKEN_SECRET);
console.log(decoded)

function generateAccessToken(user) {
  return jwt.sign(
    user,
    process.env.TOKEN_SECRET,
    {
      algorithm: "HS256",
      expiresIn: 4 * 60 * 60 * 1000
    }
  );
}
```

Tokens d'autenticació

fe25c8a6-ac6f-492f-8374-0cd18eae7c03-1650998502723

Es tracta d'un UUID més la data de caducitat en mil·lisegons. Per l'aplicació que he pensat on es fa servir, l'usuari necessita tenir la data de caducitat, tenir-la en el token li facilita la vida.

Per generar el token ho he fet amb Java.

He creat el fitxer Main.java:

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package com.tperaira.ra3_31_uuid;

import java.time.LocalDateTime;
```

```
import java.time.ZoneId;
import java.util.UUID;

/**
 *
 * @author tperaira
 */
public class Main {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        UUID uuid = UUID.randomUUID();
        LocalDateTime expiration = LocalDateTime.now().plusMonths(6); //
        Expira en 6 meses
        String token = uuid + "-" +
        expiration.atZone(ZoneId.systemDefault()).toInstant().toEpochMilli();
        System.out.println("token: " + token);
    }

}
```

Claus Asimètriques

Crearem el parell de claus asimètriques amb l'eina *OpenSSL* i el sistema *RSA*.

Podem fer servir aquest parell de claus per encriptar contingut d'un servidor. Tot el contingut que s'hagi encriptat amb la clau pública, es podrà desxifrar amb la clau privada, i viceversa.

La clau privada serà de 4096 bits. Encriptarem la clau a partir del pass phrase que li donem.

Generem la clau privada:

```
openssl genrsa -des3 -out key_private.pem 4096
```

- **genrsa:** Genera una clau privada RSA.
- **-des3:** Opció per xifrar la clau privada amb un xifrat específic.
- **-out key_private.pem:** Guarda la clau en un fitxer.
- **4096:** la mida de la clau a generar en bits.

Pass phrase usat:

Stimulus-radial-vicarage

Generem la clau pública:

```
openssl rsa -in key_private.pem -outform PEM -pubout -out key_public.pem
```

- **rsa:** Comanda per processar les claus RSA.
- **-in key_private.pem:** Especifica el nom del fitxer d'entrada que inclou la clau privada. Si la clau està xifrada amb un pass phrase, es demanarà després.
- **-outform PEM:** especifica el format de sortida, les opcions són DER|NET|PEM.
- **-pubout:** per defecte s'imprimeix la clau privada, amb aquesta opció s'imprimeix la clau pública.
- **-out key_public.pem:** Guarda la clau en un fitxer.

Clau privada.

```
-----BEGIN RSA PRIVATE KEY-----
Proc-Type: 4, ENCRYPTED
DEK-Info: DES-EDE3-CBC, 09BCE3529D1C9F69

5Mb3C1+aAbkDPHoMs5S3eGN+omh5pc/vmeAL1a8+e5TrSNxs fWMgmedh+BvzZ2JO
8P4ny6KzfXLc4YM5xcJXz07qNk3bf4pnNlXxCuP+TktWHOUiBS8X1BoLKwdpCiR8
HZm4ShM8jbDrBmiVGuPB55AvjkQBW+CvAjqg0PE14b2v6ZStR3LPSNPihZnaBBc6
puvAk+UwpzV26ELbGTyvVNdhdFIWJlXmUHIITtLnuvqL1rLb5GXCU0P5Mkdne4py7
40fZ0HWKIiNkX7D54P7NmKu3BxYXrNbh1J4mFrKgizel+tH10UBq72kpqUQ4kt9Q
RokWoN4IC35BrMWk+/YtSR8EcotazfswYYLlghmm0RbRPXmfXKgfK81a0A6qpCb78
WQOadcnqKJfJeLdHbWx28uHaIU6XmoNokVlRVCf9XHB8s1KRy+1P+G1h5BRCQRF
jIc7lObI5coXbnOxoYvYFMISauIJUaSlboH2sy9uN2azW2Cr8e8K09e22MDbK4Xc
ev7NZKWEXoW/5q1/FnPUDDpDjbHDYLo3VpUJkSpl1bjzzNuzA+6fFTPBTaX5ihrn
295OH1vYtlnFM04ymWAjdUx2pkWHE4DzFbHBfev194n21x3QPlYEIWSxLE3YJwCL
omLdAasL1d8UxJj8n1Mzkp6Uq9V42ShYPHW5YGF9ozFRlWx/hExdvEvRUdFUq/+f
bokStKzTRRE1ozXFJyovU4EB9BCvEMRV8QBN2k4iHQUM6vrourkTGQSpbEzI8EDA
XiTJequm305jGwvYvLT6NUQWo19+AAATQ/FgUixpLuPPE0iwG3NuZ6lbAXOLXLz1W
S5ACI1CnbA8THFqUJaVoDpyzdoyQ3AWA83NpYLgrewvP+ZJVxnPWTa2GR+LUP3wh
seTQ2vsKG2kofn+QRS8/tD+CpMa0jRPG0gypTKy0f7cKf4A38Bz6rtyRXb0nks+6
VL0L3YxRgtPX4FbYZ4eYNUcmIlnZwVwK3o+NBDrn2HOF10cYotsgmbuTTF+iooJo
Fm5dIy7yZhq6TBQC1YiJY7D5G68To6Xcj3brSJqfOefK99os8E7YgSefAwRbcQyJ
DISV31zt0TBIZ+TDpdD7bIgVVvGISW6NRFYBe9ei6zUdUrEMGDz2pwysPoSigtiU
XghLhKQ4QvUvLmIs71SMOzueHF+mY9gRTIylnTIsUK4Z8opyAEsXsT+BP62t6Mpd
EZEUjMSZLLAL/NTHD86lfsIDOhJwr+NxazRB90qriQtWVGsbrRA6WBw5pCFQGPJI
WnECfj99Q6KWq1rGvLB9yzYUOhcUegPfTn2RN4pcOSK0t2JpokLdqunjvxXpQGrF
Xf2y3ivW4g+yVM/UjPaujpHgTp2YAh0v6ME0D7Tj4v03xvny8US/x0RB0eD82Hh4
pWMVAb93N3SriYfquvxVismbbbWl03S3aZFGAVK8HtbRlXBJ2zhwycx8EW1xrzn9
HlrO9/yxP11PbDqT2kKysQyUG1Uz1PAUTbRAGNhIaPMSA8H1VZWPNqT+w+8nAW//
8vYr7ILbs+wZtBtaSnjbDurVdWKreA0ncTXvuxzy6m8vDC901NBFHjyl6oIsZ+8o
dXgLHNRGd2ET6wpTcfMwNok/EkftiTwfSYJDC+HgcU4gXws0sLK7/QetnqxOCcVv
izWV4MCD9aAQOS9Z9nkoOyLs+68pUpBeke8ElahlApuXptoBJjkNSp2YS3nJW09p
V75vIppWHRw/tb4j3PoQg9a6H4BxENZIn3d6do+zVke/ja4h1YRV/9UvBYzWAICy
9kjLXHjorRvw1lW2Ymhhu6Sgoz/N9awlyuifWVK6LKgD69kzBSfLsNOrpX44gDFQ
GIUbDS1stShg+k/KKuaG+Z9R29D9qCtgdFt3luEzn5vsd2FKGRk+udOSjq7Fr8mu
70d+CtZDMLn8gefW0yJfOkY2sQJhuRgHTiXtzL7wsQbY11tqNjvWsUd8nbW14ECf
WDyvbOBp59Ll/P/cmDkxR3r1KV2kjVseOibmzud+RdgHELrrBMD4E7FEM/HSp/bm
8r6kZO5mumxcYizupfzUeAxapaqFym35ghprFpjA5lLUcxBwQ00kzNJHjFnGqHu6
j0Tvi7ZEGbTVaS5GG1vXfLhY8u6/3xIYTemcUU3+R2OFd/8EN9yz1EK5FSJLsac0
O2GUduBp+h2BCxsituCEg4AgHmMozTW7UlgkjBQaUhu6dcXqJlUT+5aHvCBzhpm2
wYEnxSdqRk5EpVwgMxIZntvRRpQgWiKgWpvvRAGqOk4En3cihfcJ6212/JZCa8qm
/5fbmP2dkBwBoxRlHk36t6JpmLjYBnhM/pmXhZ2Dsu55K/xQXkj+3iwOufLHYII+
wtEYBkPkGWhq/9kiuusTWBmdi3b+0mCrG1lR0IMaUA9GjwTkT1/6HDtAGvuQVCq4
Xr1/OZ6SvPc5zBWqWrEZVLOEE1r8S+WbYR2p6ULRskfdYHHmeuzIMpz5AZWZLnnb
```

```
pojCpBZrB2+lpwHgWrwaZ0kJvBzv4otvQxfzT/3x50drDEK3Bcp8+8HGTPAVn2iF
IRJAHirKTMLY7mBmXvNFbrk0L6F9Lvpz2e8dHuo2gG8iBxpX/UF3v4TfX2siEVMp
GkvOYdU6TmKf39hJd/PPzTCto1kvUMDUziIMAvzq7D17Iztc4E+BKPR+NwyiHUHS
zYpSk/nf3FKhQZH2VzuMaGPRuCQyk9Lv119NSLfMJ1ZcV7KsJjegoacFBUOzfeVb
lTd9Qn/viip6OYMX3vhYMKov19lvyszagmqhlQUbY5/gYyZ1DhYbcxvzs8q6HrHd
znphetBKTsWXsBMLE8PRXv56j11FxPoLN2jGP4T/K+TdZqvbzNAJSWwsODKMgx3
TTxTTGr9YB/k+B4zZQXARFsQ8sTVyxnfYraUcY0e3Y/fw0Cn6AXm6WXj+L/kDtjT
rJ+Rj27CN5XwrfqjCPs9AXnUSucGtedgDW+q0Y8APelertVA+Qv9ydoRf7ceB8ud
Gmet75+Ev9V4e5bYOTz6ZrGl6gLhE52ENlrb2Gamu9QSI1qs7OwbFnq7GuNXJIuy
LInVrJl3Edslh2d4pPZvP3kWZMtvjnmrF+FUCmxQ+3731XwKAJmMUvVshbGPDV/F
-----END RSA PRIVATE KEY-----
```

Clau pública.

```
-----BEGIN PUBLIC KEY-----
MIICIjANBgkqhkiG9w0BAQEFAAOCAg8AMIICCgKCAgEAXaNGOaFUjT8tgIFVdEy
adFkxxcccu6IViWYYkb5KCyaATTMXePtoXzxXTwWQ5updAzo1r39Ys1b7FxsV8NC
2v8s6UX48L8GC2FoYbq8dmqhtSuENZiAFye86+mWnHkcRJHQ+um3pl6ryj1wrRoE
pIWSWlJ/11bo9g48LLX5B8BySB+325CjYnw6X4FIxS2FvUob12P5e5MMxhnjldtw
mGO3ZRFZUUp8dArO7NuC46IeORgvR9SvGkNq5xEdz0zTULkBXbVfVu9qCWIRdRgV
kvNZ4iVw2Z2aOgGEMucPE1bnSBERlq/jPD4z03ERbxCo19p7ZWEADZWHZDN+KJLO
8sOx3zIBZltDWLGRy8R42DReiUtc+r92P6rC8CH1AIFx33JrPuGIdb0sVSR4cikip
BZ0fYd8pBdxqo0dkJrs1eY6bUKVTdY7S3w/6wAzFX3DPbQ1XWfwmUq9yMOGP8FU6
MM39HJBeUrTkBEWFTAUCIpQ+UGy3yqETevhevVW7Nk4qXAJxqVY7i/dBsa3qRdTe
mrqF0Va7SjEJRtZjPmnnIXR2yk451XndCW8sMOhk5bhYE/gTtRyGcjKbFR6VUWM
H8YAunOEzY+k5I6z4/816SVqtoDu2EPaKvj6n9+W5dryE2Cpmv9TrmhHzXb4kZQW
rAAvKp+ftRZS5a9HwnGxe4sCAwEAAQ==
-----END PUBLIC KEY-----
```

Certificats digitals a partir de claus asimètriques.

Farem servir una altra vegada l'eina *OpenSSL*, aquest cop per generar un certificat digital.

Passos a seguir:

1. Creem un certificate authority (CA). Aquest CA serveix per signar digitalment cada certificat que es farà servir.

1. Creem un fitxer de sol·licitud de certificat (CSR).

```
openssl req -passout pass:throb-audio-dethrone -subj
"/C=SP/ST=Girona/L=Girona/O=Institut
Montilivi/OU=Ciberseguetat/CN=Montilivi
Testing/emailAddress=a2102226@institutmontilivi.cat" -keyout
ciberseguetat_montilivi_privkey.pem -new >
ciberseguetat_montilivi.ca.cert.csr
```

- req: La comanda procesa principalment sol·licituds de certificat en format PKCS. També pot crear certificats autosignats.
- -passout pass:throb-audio-dethrone: Pass phrase del fitxer.
- -subj: Informació del certificat amb les dades del propietari.
- -keyout: Nom del fitxer que contindrà la clau.
- -new: Genera una nova sol·licitud de certificat.

2. Crear el CA, `ciberseguretat_montilivi.ca.key`, on es guarda la clau privada. Això elimina la protecció amb contrasenya del CSR per no haver de posar la contrasenya cada vegada que es signi un certificat.

```
openssl rsa -passin pass:throb-audio-dethrone -in  
ciberseguretat_montilivi_privkey.pem -out  
ciberseguretat_montilivi.ca.key
```

3. Crear un certificat digital X.509 a partir del CSR. La comanda crearà el certificat signat amb la clau privada CA.

```
openssl x509 -in ciberseguretat_montilivi.ca.cert.csr -out  
ciberseguretat_montilivi.ca.cert -req -signkey  
ciberseguretat_montilivi.ca.key -days 365
```

```
* ciberseguretat_montilivi openssl x509 -in ciberseguretat_montilivi.ca.cert.csr -out cibersegure  
tat_montilivi.ca.cert -req -signkey ciberseguretat_montilivi.ca.key -days 365  
  
Signature ok  
subject=C = SP, ST = Girona, L = Girona, O = Institut Montilivi, OU = Ciberseguretat, CN = Montili  
vi Testing, emailAddress = a2102226@institutmontilivi.cat  
Getting Private key
```

- x509: Comanda per crear certificats amb format X.509. És un estàndard que defineix la sintaxis dels certificats i que no fa servir un algoritme en particular.
 - -in: Especifica el fitxer amb el certificat d'entrada.
 - -out: Especifica el nom del fitxer de sortida.
 - -req: Per defecte s'espera un certificat a la comanda. Amb aquesta opció s'espera una sol·licitud de certificat.
 - -signkey: Fa que el fitxer d'entrada es signi automàticament amb la clau privada que s'especifiqui.
 - -days: Especifica els dies que és vàlid el certificat.
4. Crear un fitxer PKCS#12 codificat que conté el certificat i la clau privada. PKCS#12 (també conegut com PKCS12 o PFX) és un format binari per emmagatzemar una cadena de certificats i una clau privada en un únic fitxer xifrat.

```
openssl pkcs12 -passout pass:toque-strident-stingy -export -  
nokeys -cacerts -in ciberseguretat_montilivi.ca.cert -out  
ciberseguretat_montilivi.ca.cert.p12 -inkey  
ciberseguretat_montilivi.ca.key
```

- pkcs12: Permet treballar amb fitxers codificats PKCS#12.
- -passout pass: Pass phrase per xifrar les claus privades.
- -export: Especifica que es crearà un fitxer PKCS#12 en lloc d'analitzar-lo.
- -nokeys: No imprimeix cap clau privada.
- -cacerts: Únicament retorna certificats CA.
- -in: Especifica el nom del fitxer PKCS#12 que s'ha d'analitzar.
- -out: El nom del fitxer on s'escriuen els certificats i claus privades.
- -inkey: Fitxer per llegir la clau privada.

2. Ja tenim el certificat CA (`ciberseguretat_montilivi.ca.cert`), el qual es pot instal·lar en un servidor web i un fitxer amb la clau privada (`ciberseguretat_montilivi.ca.key`) que podem fer servir per signar certificats d'usuari.

3. Crear un certificat digital per un usuari.

1. Crear un fitxer CSR per l'usuari. \

```
openssl req -passout pass:jackass-sedative-cherub -subj
"/C=SP/ST=Girona/L=Girona/O=Institut
Montilivi/OU=Ciberseguretat/CN=Montilivi Testing - Toni
Peraira/emailAddress=tperaira@institutmontilivi.cat" -keyout
ciberseguretat_montilivi_tperaira_privkey.pem -new >
ciberseguretat_montilivi_tperaira.cert.csr
```

2. Crear el fitxer amb la clau privada.

```
openssl rsa -passin pass:jackass-sedative-cherub -in
ciberseguretat_montilivi_tperaira_privkey.pem -out
ciberseguretat_montilivi_tperaira.key
```

3. Crear un nou certificat X.509 per l'usuari, el signem digitalment amb la clau privada de l'usuari i el certifiquem amb la clau privada CA.

```
openssl x509 -req -in ciberseguretat_montilivi_tperaira.cert.csr
-out ciberseguretat_montilivi_tperaira.cert -signkey
ciberseguretat_montilivi_tperaira.key -CA
ciberseguretat_montilivi.ca.cert -CAkey
ciberseguretat_montilivi.ca.key -CAcreateserial -days 365
```

4. Crear el fitxer PKCS#12 codificat. \

```
openssl pkcs12 -passout pass:demented-marry-flung -export -in
ciberseguretat_montilivi_tperaira.cert -out
ciberseguretat_montilivi_tperaira.cert.p12 -inkey
ciberseguretat_montilivi_tperaira.key
```

Bibliografia

<https://www.openssl.org/docs/man1.0.2/man1/genrsa.html>

<https://www.openssl.org/docs/man1.0.2/man1/openssl-rsa.html>

<https://www.openssl.org/docs/man1.0.2/man1/openssl-req.html>

<https://www.openssl.org/docs/man1.0.2/man1/x509.html>

<https://www.openssl.org/docs/man1.0.2/man1/pkcs12.html>

<https://www.ibm.com/docs/en/rpt/9.1.0?topic=overview-creating-digital-certificate-openssl>