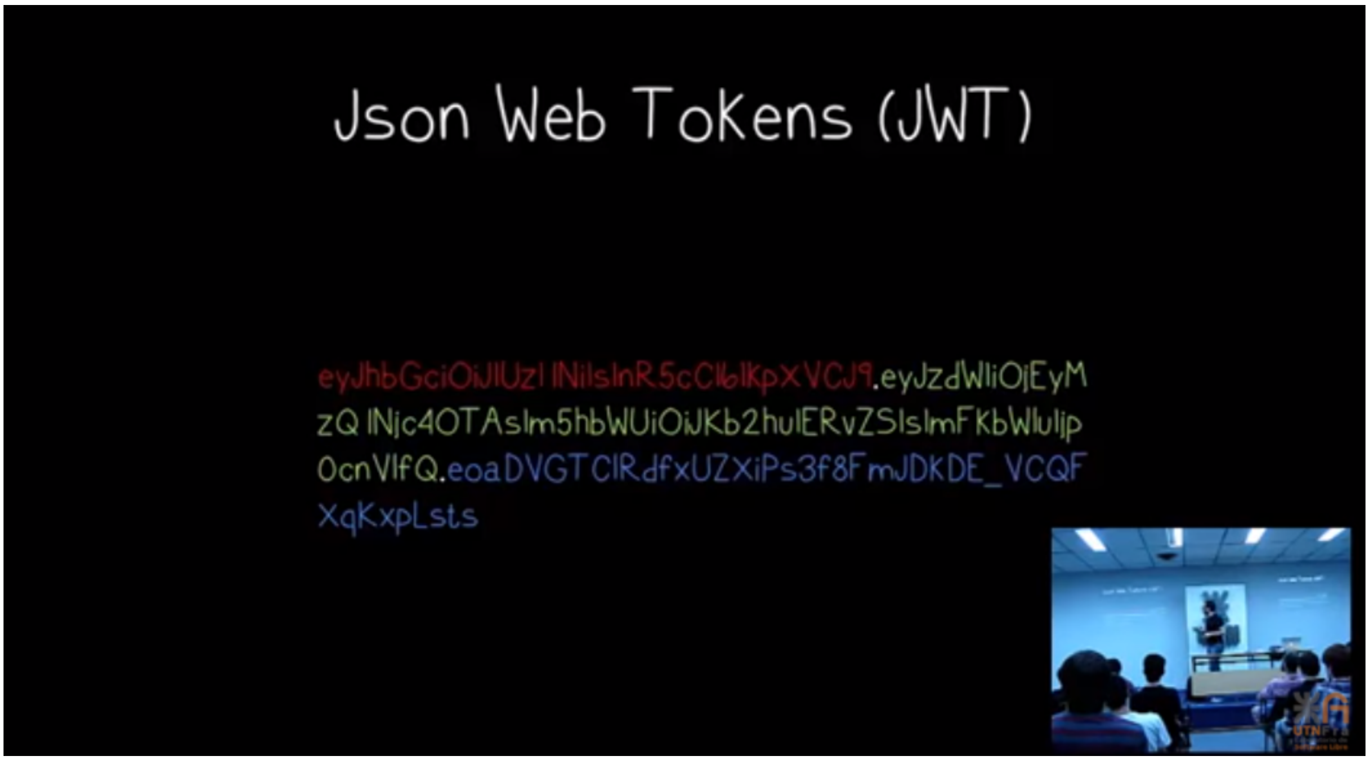
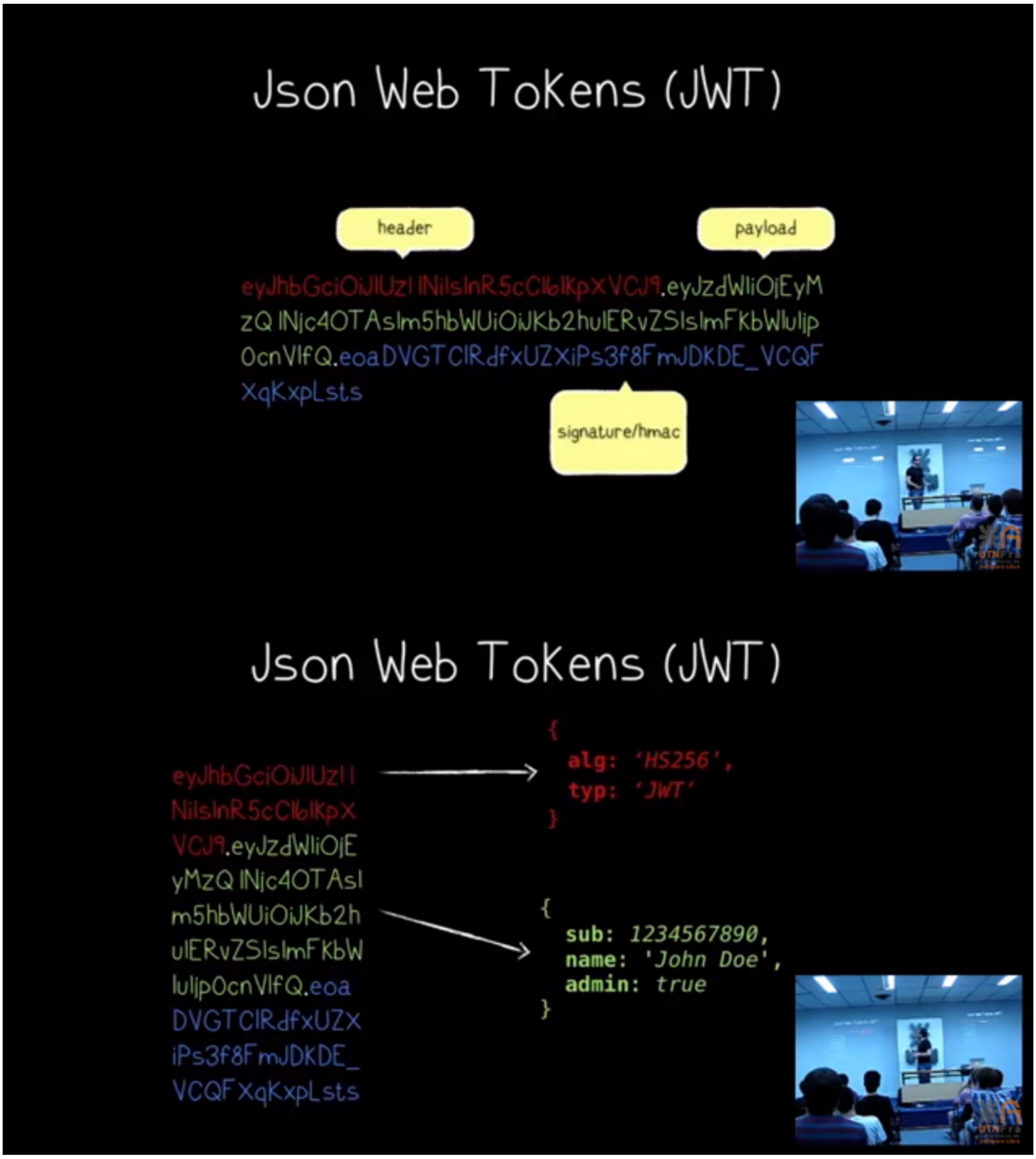
En este artículo vamos a revisar apuntes sobre [JWT JSON Web Token]

- Clave simetrica: es yo encripto y desencripto con la misma clave
- Clave asimetrica: yo tengo mi clave privada y las otras apps tienen una clave publica
- JWS Signature
- JWE Encription



Json Web Tokens (JWT)

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOjEyM
zQ1Njc4OTAsIm5hbWUiOiJKb2huIERvZSIsImFkbWluIjp
0cnVlfQ.eoaDVGTClRdfxUZXiPs3f8FmJDKDE_VCQF
XqKxpLsts
```



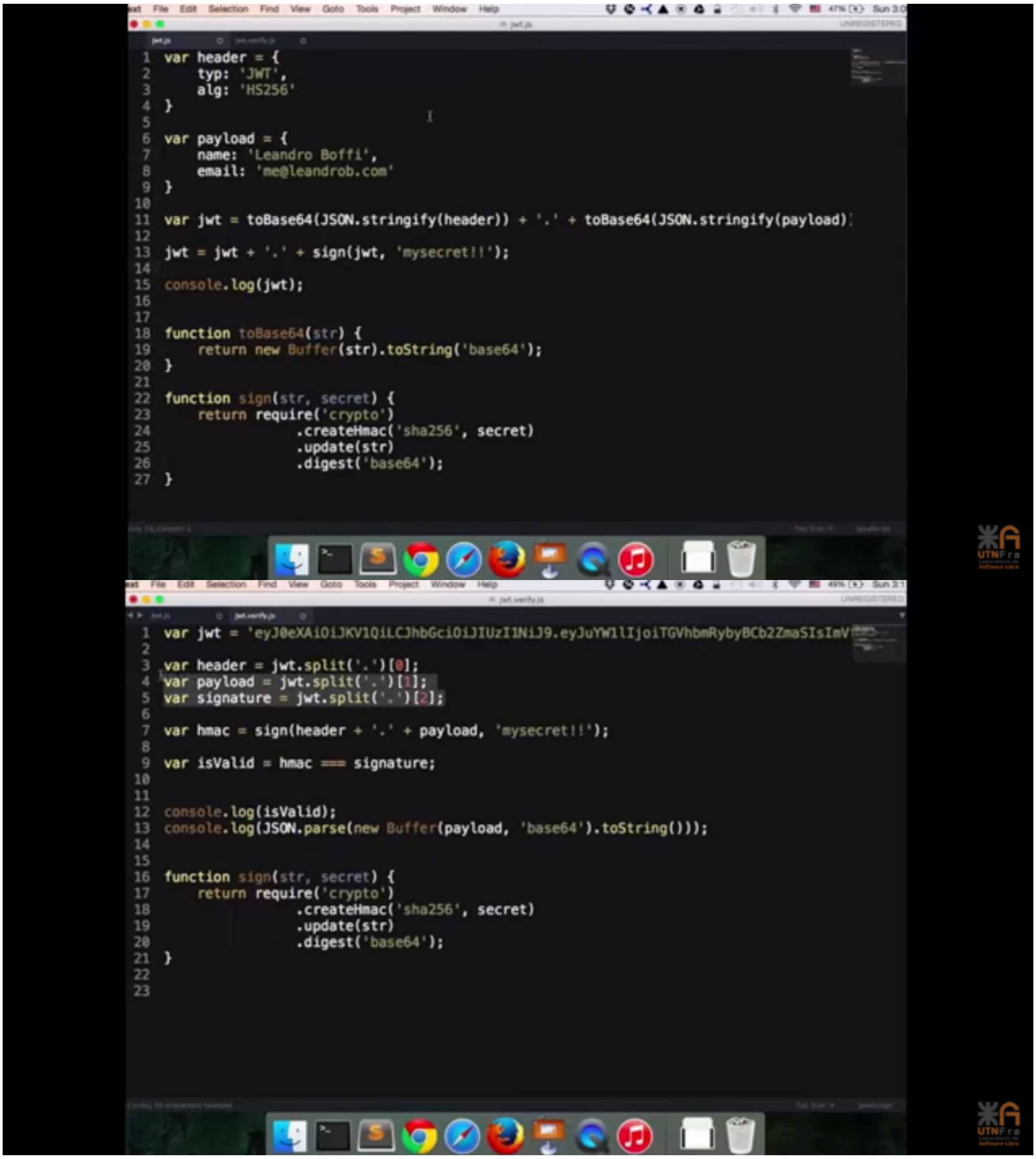
Json Web Tokens (JWT)

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiEyM
zQ1Njc4OTAsIm5hbWUiOiJKb2huIERvZSIsImFkbWluIjp
0cnVlfQ.eoaDVGTClRdfxUZXiPs3f8FmJDKDE_VCQF
XqKxpLsts
```

Json Web Tokens (JWT)

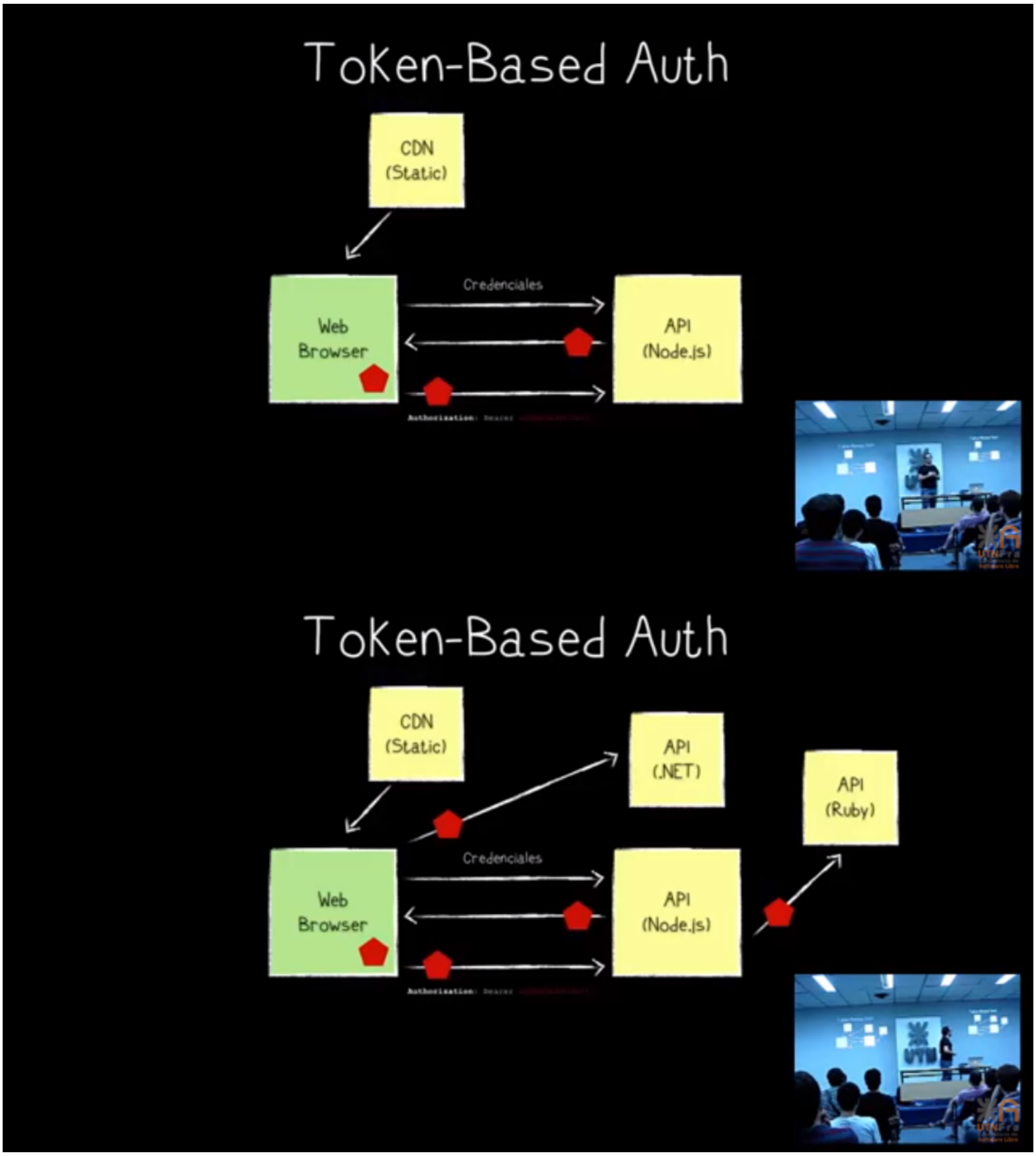
```
eyJhbGciOiJIUzI1
NiIsInR5cCI6IkpX
VCJ9.eyJzdWIiOiE
yMzQ1Njc4OTAsI
m5hbWUiOiJKb2h
uIERvZSIsImFkbW
luIjp0cnVlfQ.eoa
DVGTClRdfxUZX
iPs3f8FmJDKDE_
VCQFXqKxpLsts
```

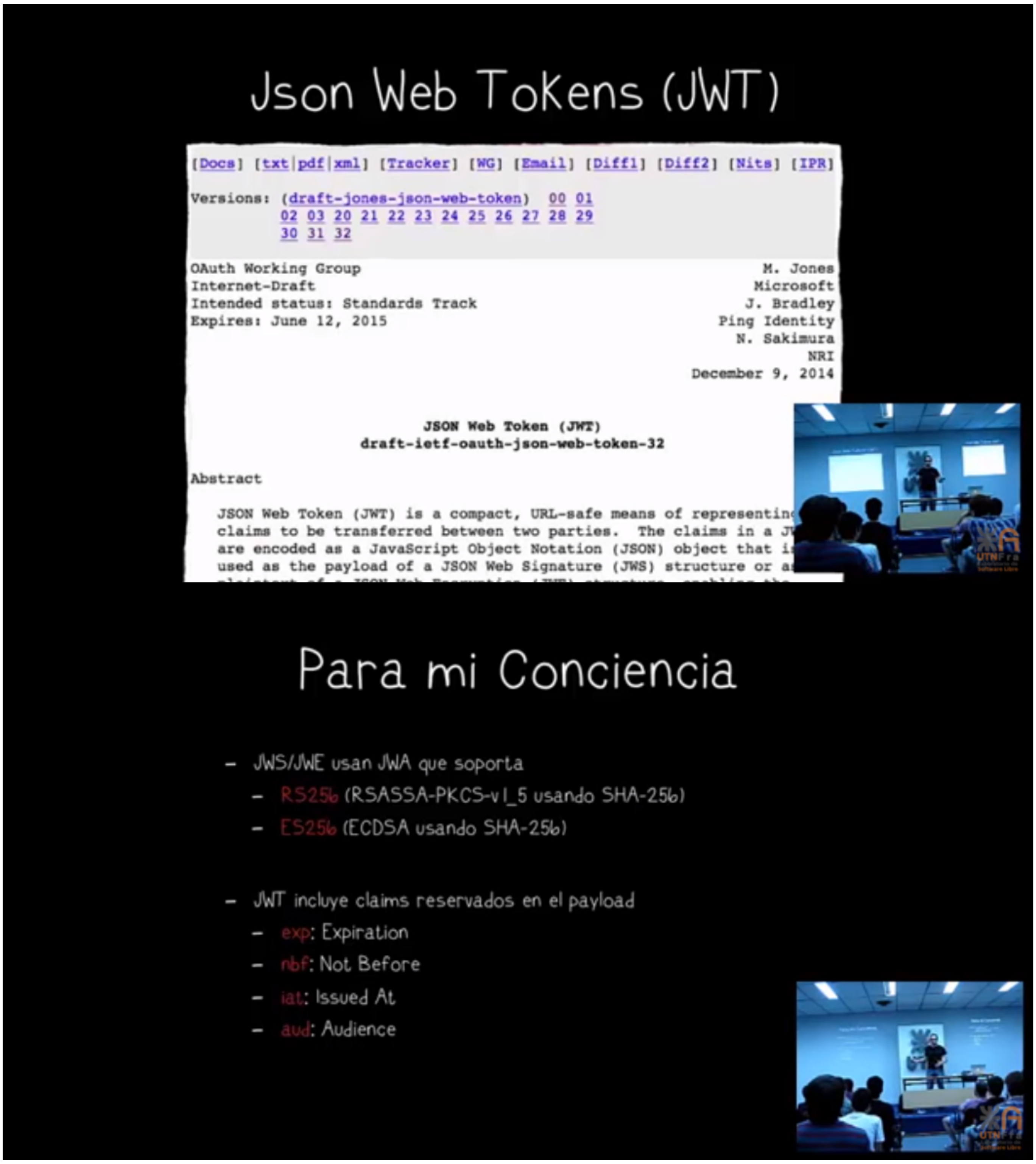
```
{
  alg: 'HS256',
  typ: 'JWT'
}

{
  sub: 1234567890,
  name: 'John Doe',
  admin: true
}
```



```
var header = {
    typ: 'JWT',
    alg: 'HS256'
}

var payload = {
    name: 'Leandro Boffi',
    email: 'me@leandrob.com'
}

var jwt = toBase64(JSON.stringify(header)) + '.' + toBase64(JSON.stringify(payload)

jwt = jwt + '.' + sign(jwt, 'mysecret!!');

console.log(jwt);


function toBase64(str) {
    return new Buffer(str).toString('base64');
}

function sign(str, secret) {
    return require('crypto')
            .createHmac('sha256', secret)
            .update(str)
            .digest('base64');
}
```

```
var jwt = 'eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJuYW1lIjoiTGVhbmRybyBCb2ZmaSIsImV

var header = jwt.split('.')[0];
var payload = jwt.split('.')[1];
var signature = jwt.split('.')[2];

var hmac = sign(header + '.' + payload, 'mysecret!!');

var isValid = hmac === signature;


console.log(isValid);
console.log(JSON.parse(new Buffer(payload, 'base64').toString()));


function sign(str, secret) {
    return require('crypto')
            .createHmac('sha256', secret)
            .update(str)
            .digest('base64');
}
```


Token-Based Auth
CDN (Static)
Web Browser
Credenciales
API (Node.js)
Authorization: Bearer
Token-Based Auth
CDN (Static)
API (.NET)
API (Ruby)
Web Browser
Credenciales
API (Node.js)
Authorization: Bearer



Json Web Tokens (JWT)

```
[Docs] [txt|pdf|xml] [Tracker] [WG] [Email] [Diff1] [Diff2] [Nits] [IPR]

Versions: (draft-jones-json-web-token)  00 01
          02 03 20 21 22 23 24 25 26 27 28 29
          30 31 32

OAuth Working Group                                             M. Jones
Internet-Draft                                                 Microsoft
Intended status: Standards Track                              J. Bradley
Expires: June 12, 2015                                     Ping Identity
                                                             N. Sakimura
                                                                     NRI
                                                        December 9, 2014


                         JSON Web Token (JWT)
                   draft-ietf-oauth-json-web-token-32

Abstract

   JSON Web Token (JWT) is a compact, URL-safe means of representin
   claims to be transferred between two parties.  The claims in a J
   are encoded as a JavaScript Object Notation (JSON) object that i
   used as the payload of a JSON Web Signature (JWS) structure or a
```

Para mi Conciencia

- JWS/JWE usan JWA que soporta
  - RS256 (RSASSA-PKCS-v1_5 usando SHA-256)
  - ES256 (ECDSA usando SHA-256)
- JWT incluye claims reservados en el payload
  - exp: Expiration
  - nbf: Not Before
  - iat: Issued At
  - aud: Audience

Video

Adjunto PDF Oficial de JWT Cookbook: jwt-handbook.pdf

ArquitecturaIBM Consulting