El futuro digital es de todos
MinTIC

Recuerda marcar tu asistencia

Asistencia en LMS

MISIÓN TIC 2022
unab
VIGILADA MINEDUCACIÓN
Hechos QUE CONECTAN

Semana 5

Sesión 03

JWT

Ing Edwin Garcia

El futuro digital es de todos

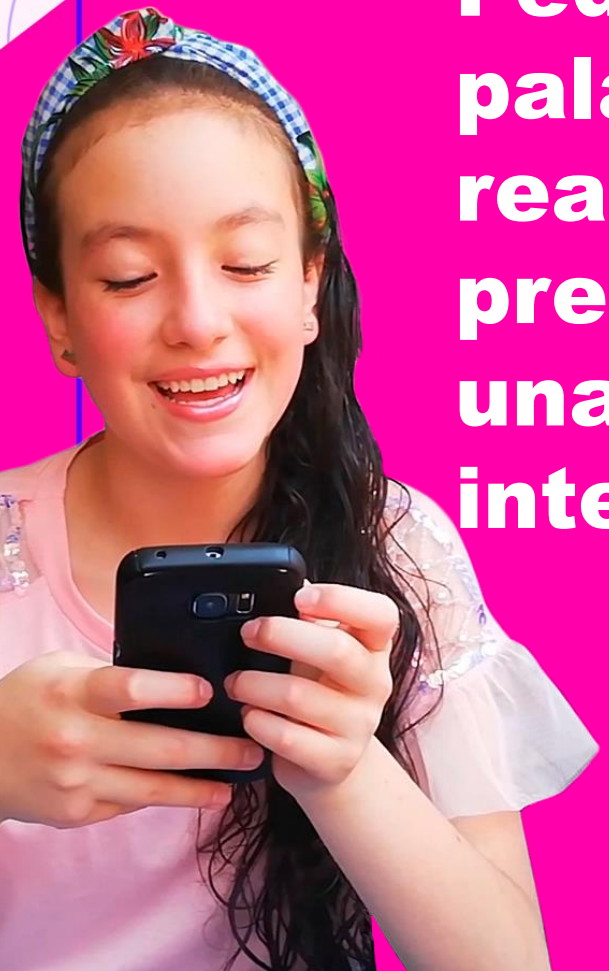
MinTIC

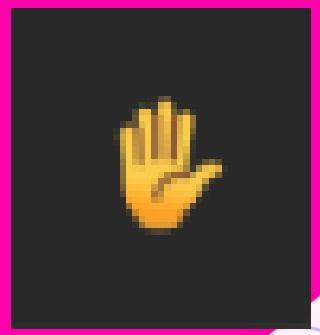
Hechos QUE CONECTAN

Recuerden marcar su asistencia

Asistencia en LMS

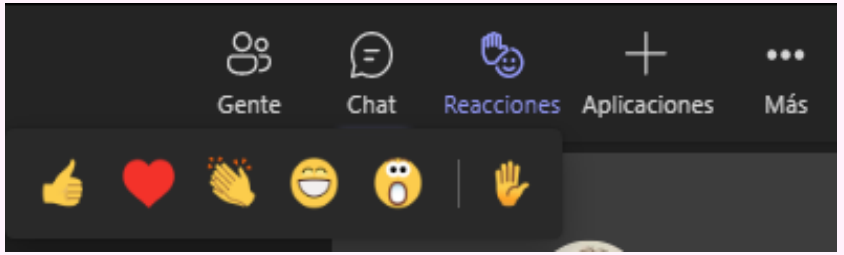
Pedir la palabra para realizar una pregunta o una intervención

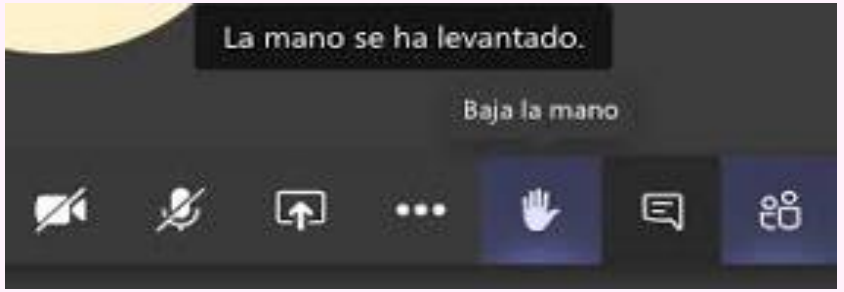




El futuro digital es de todos

MinTIC





Misión TIC 2022

unab
VIGILADA MINEDUCACIÓN

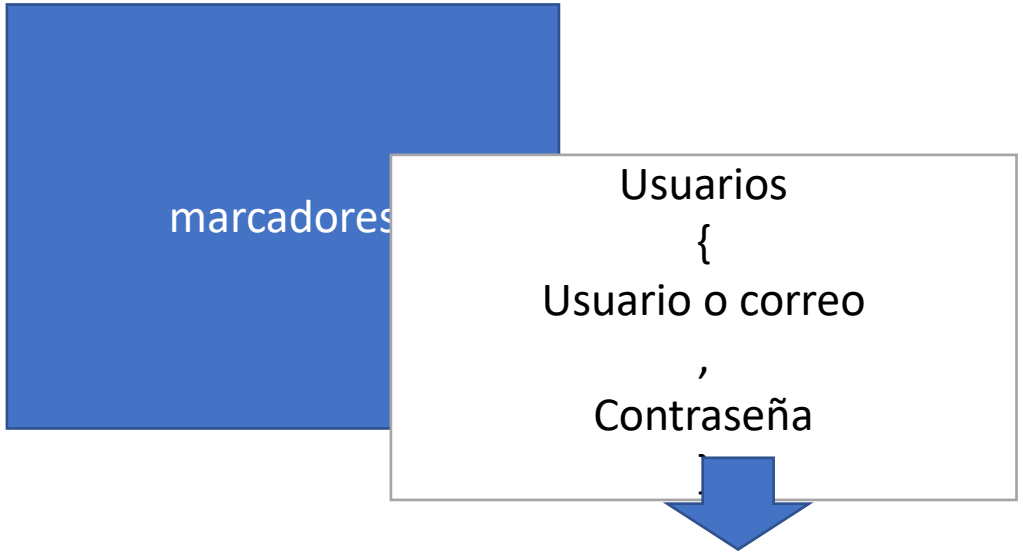
Hechos QUE CONECTAN

El futuro digital es de todos
MinTIC
Mision TIC 2022
unab
VIGILADA MINEDUCACIÓN
Hechos QUE CONECTAN

bcryptJS

Bcrypt es una función de hashing de passwords diseñado por Niels Provos y David Maxieres, basado en el cifrado de Blowfish.

usuario	contraseña
usu@mintic.co	12345 => #5JEWIOJWEFjwErwJRWe
usu2@correo.co	67890 =>EWRU)##MW#LFW#RWO



Lleva incorporado un valor llamado **salt**, que es un fragmento aleatorio que se usará para generar el hash asociado a la password, y se guardará junto con ella en la base de datos. Así se evita que dos passwords iguales generen el mismo hash y los problemas que ello conlleva, por ejemplo, ataque por fuerza bruta a todas las passwords del sistema a la vez.

El futuro digital es de todos
MinTIC

- Abcde => edcba => cbaed => deabc => aedbc =>

Mision TIC 2022
unab
VIGILADA MINEDUCACIÓN
Hechos QUE CONECTAN

El futuro digital es de todos
MinTIC

```
> npm i --save bcryptjs
```

Mision TIC 2022
unab
VIGILADA MINEDUCACIÓN
Hechos QUE CONECTAN

El futuro digital es de todos
MinTIC

```
const mongoose = require('mongoose')

const usuarioSchema = mongoose.Schema({
    nombre :{
        type: String,
        required : true,
        trim : true
    },
    correo : {
        type: String,
        required: true,
        trim : true
    },
    contrasena : {
        type : String,
        required : true,
        trim : true
    }
})

module.exports = mongoose.model("usuario", usuarioSchema)
```

Mision TIC 2022
unab VIGILADA MINEDUCACIÓN
Hechos QUE CONECTAN

El futuro digital es de todos
MinTIC

```
const usuarioModel = require('../models/usuarioScheme')
const bcryptjs = require('bcryptjs')

const usuarioGuarda= async (req,res)=>{
    console.log("guardando usuario")
    console.log(req.body)
    try{

        const usuario = new usuarioModel(req.body)
        usuario.contrasena = await bcryptjs.hash(usuario.contrasena,10)
        await usuario.save()
        res.status(200).json({msj: "Usuario Creado "})
    }catch(error){ console.log("usuario ctrl error" + error)  }
}
```

```
module.exports = {
    usuarioGuarda
}
```

Mision TIC 2022
unab VIGILADA MINEDUCACIÓN
Hechos QUE CONECTAN

El futuro digital es de todos
MinTIC

routes > JS usuariosRoute.js > ...

```
1  const express = require('express')
2  const router =  express.Router()
3  const usuariosCtrl = require('../controllers/usuariosCtrl')
4
5  router.post('/', usuariosCtrl.usuarioGuarda)
6
7  module.exports = router
8
```

Mision TIC 2022
unab VIGILADA MINEDUCACIÓN
Hechos QUE CONECTAN

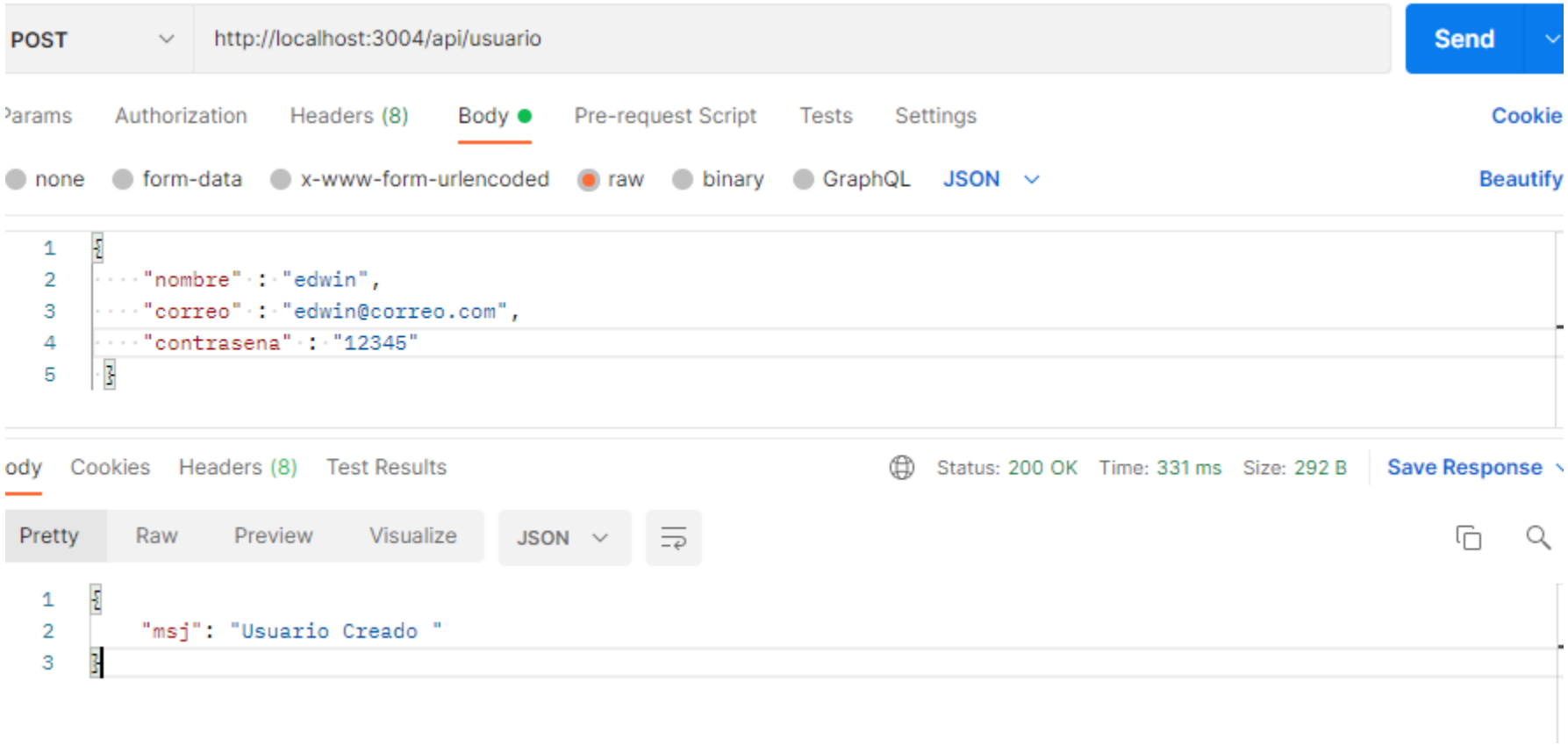
El futuro digital es de todos
MinTIC

- En index.js

```
app.use("/api/usuario/",require('./routes/usuariosRoute'))
```

Mision TIC 2022
unab
VIGILADA MINEDUCACIÓN
Hechos QUE CONECTAN

El futuro digital es de todos
MinTIC



Mision TIC 2022
unab
VIGILADA MINEDUCACIÓN
Hechos QUE CONECTAN

JWT

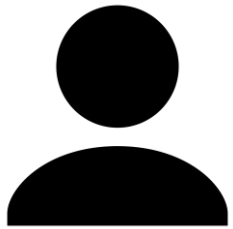
- JWT (JSON Web Token) es un estándar qué está dentro del documento RFC 7519.
- En el mismo se define un mecanismo para poder propagar entre dos partes, y de forma segura, la identidad de un determinado usuario, además con una serie de claims o privilegios.

- Estos privilegios están codificados en objetos de tipo JSON, que se incrustan dentro de del payload o cuerpo de un mensaje que va firmado digitalmente.

Token JWT

En la práctica, se trata de una cadena de texto que tiene tres partes codificadas en Base64, cada una de ellas separadas por un punto, como la que vemos en la imagen siguiente:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6I
kpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.ikFGEvw-Du0f30vBaA742D_wqPA5BBHXgUY6
wwqab1w
```



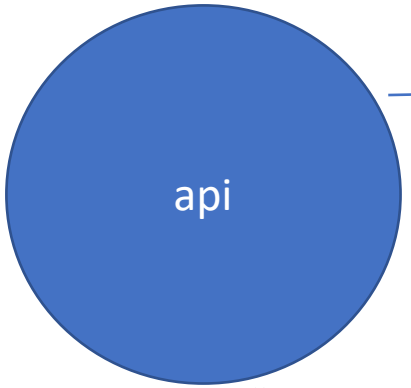
Acceso usu,pass

modificacion
“jwt token”

consultar
Jwt token

Palabra secreta

EWR)R)R#$#OJR”#R#”$O#”$J#”O$J”#$Ñ#”$”#$



Ok
no

“jwt token”

Token{
Header,
Payload,
sign
}

”#$Ñ#”$”#$

firma

tipo

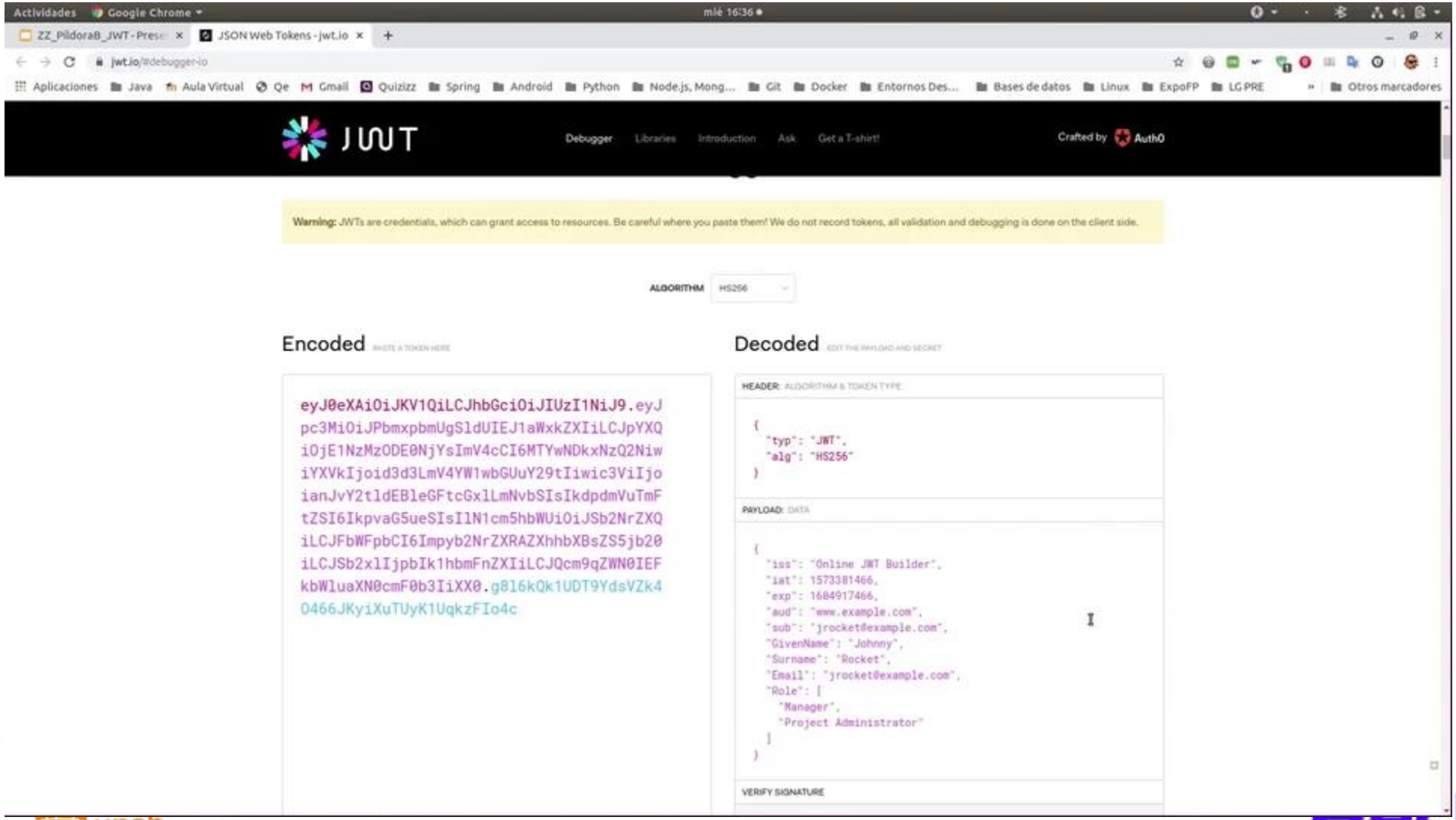
EWR)R)R#$#OJR”#R#”$O#”$J#”O$J”#$Ñ#”$”#$

Datos
expiracion

Token	-	expir

- Podemos utilizar un debugger online para decodificar ese token y visualizar su contenido. Si accedemos al mismo y pegamos dentro el texto completo, se nos mostrará lo que contiene:

El futuro digital es de todos
MinTIC



Mision TIC 2022
unab
VIGILADA MINEDUCACIÓN
Hechos QUE CONECTAN

- Un token se compone de tres partes:
- **Header**: encabezado dónde se indica, al menos, el algoritmo y el tipo de token, que en el caso del ejemplo anterior era el algoritmo HS256 y un token JWT.
- **Payload**: donde aparecen los datos de usuario y privilegios, así como toda la información que queramos añadir, todos los datos que creamos convenientes.
- **Signature**: una firma que nos permite verificar si el token es válido, y aquí es donde radica el quid de la cuestión, ya que si estamos tratando de hacer una comunicación segura entre partes y hemos visto que podemos coger cualquier token y ver su contenido con una herramienta sencilla, ¿dónde reside entonces la potencia de todo esto?

Ciclo de vida de un token JWT

El futuro digital es de todos
MinTIC

```
npm i --save jsonwebtoken
```

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

```
const payload ={
    usuario : { id : usuario.id}
}
```

Mision TIC 2022
unab
VIGILADA MINEDUCACIÓN
Hechos QUE CONECTAN

El futuro digital es de todos
MinTIC

```
jwt.sign(
    payload,
    "palabra secreta",
    {
        expiresIn:3600
    },
    (error,token)=>{
        if(error) throw error
        res.status(200).json({token :token, msj: "Acceso concedido"})
    }
    )
```

Mision TIC 2022
unab
VIGILADA MINEDUCACIÓN
Hechos QUE CONECTAN

- https://openwebinars.net/blog/que-es-json-web-token-y-como-funciona/
- https://es.stackoverflow.com/questions/424831/bcrypt-nodejs-hash-de-contrase%C3%B1a
- https://es.stackoverflow.com/questions/424831/bcrypt-nodejs-hash-de-contrase%C3%B1a
- https://www.youtube.com/watch?v=bhmTEdEt1a4

Bibliografía

Practicar los temas tratados

Nos vemos en la siguiente sesión

El futuro digital es de todos

MinTIC

Muchas Gracias

www.mintic.gov.co

MisionTIC2022

unab
VIGILADA MINEDUCACIÓN