

WOMEN WHO
CODE®
/medellín

FROM HERO TO SUPERHERO

BACKEND CON NODEJS

AGRADECER A...

Nuestro patrocinador y tutores voluntarios

PATROCINADOR

softserve

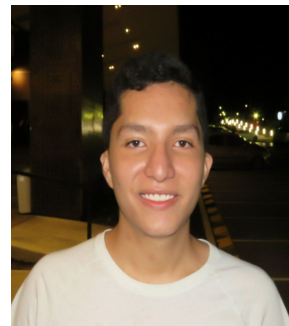
TUTORES VOLUNTARIOS



Edwin



Habid



Jean



Jose

NUESTROS VALORES

01 **PUNTUALIDAD**
El tiempo de todos es oro

02 **ORDEN**
Dejar todo mejor de lo que encontramos: limpio y ordenado.

03 **RESPECTO**
Como invitados respetamos las reglas de nuestros anfitriones, en este caso Softserve

04 **COLABORACIÓN**
Apoyarnos y ayudarnos para terminar como un solo equipo

VIVIR LA CULTURA WWCODE

Que tus actos hablen más que tus palabras

01

Theoretical Lesson

Autenticación y sesiones

02

Features of the Topic

Autenticación

Autorización

Sesiones

03

TIPS

Stateless vs Stateful

04

PRACTICAL EXERCISES

Login y autenticación

TABLE OF CONTENTS

GLOSARIO

Repasemos términos que usaremos antes de
iniciar

Glosario

Autenticación

01

La autenticación es el proceso de identificar a los **usuarios** y **garantizar** que los mismos **sean** quienes **dicen** ser.

Autorización

02

La autorización es lo que define a qué **recursos** de sistema el usuario **autenticado** podrá **acceder**.

Token

03

La tokenización se refiere al proceso de **sustitución** de un elemento de datos **sensible** por un **equivalente no sensible** denominado token, que no tiene un significado o valor extrínseco o explotable

Sesión

04

Una sesión es un intercambio de información interactiva semipermanente, también conocido como diálogo, una conversación o un encuentro, **entre dos o más** dispositivos de comunicación, o entre un ordenador y usuario

Iniciemos

¿Cuáles elementos de seguridad queremos tener en nuestro sistema?

Autenticación

La autenticación evita que cualquiera pueda entrar en un determinado sistema o iniciar sesión en alguna plataforma de forma indebida, sin que realmente sea el usuario legítimo que tiene el poder para hacerlo.



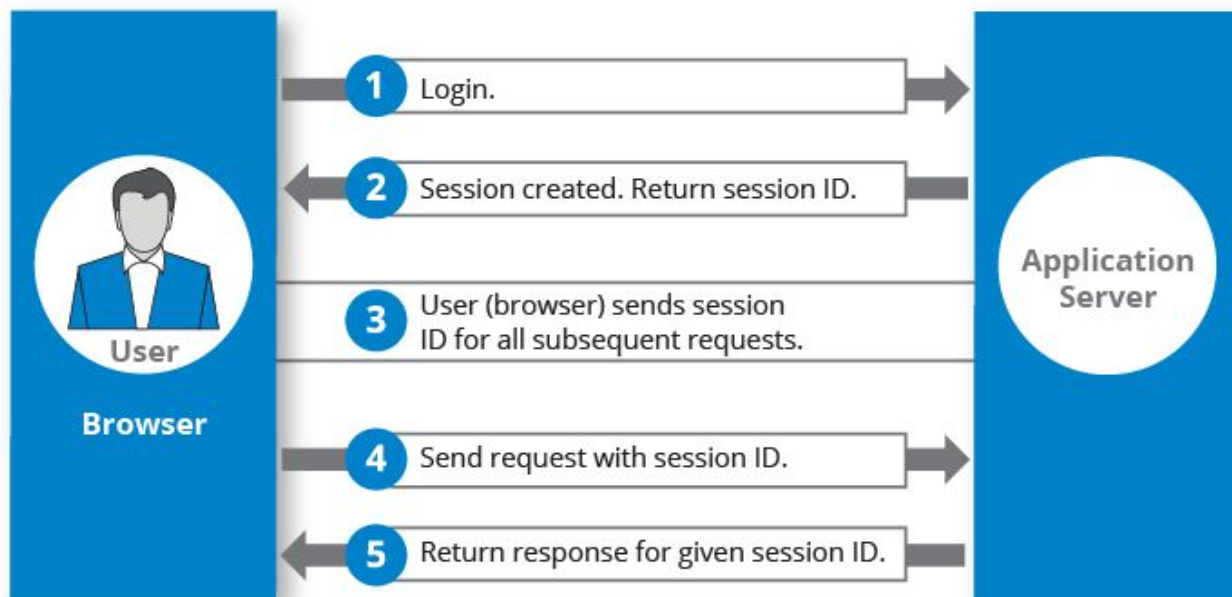
Autorización

Un sistema de software tiene muchas funcionalidades que deben ser protegidas para evitar abusos o ataques por parte de actores malintencionados



Sesiones

Una sesión es un intercambio de información interactiva.



Tipos de Sesiones



Una sesión con estado: Al menos una de las partes comunicantes necesita salvar información sobre el historial de sesión para ser capaz de comunicarse

Sesión sin estado: La comunicación consta de peticiones independientes con respuestas.

Sesiones con estado (Stateful)

¿Como funcionan?

Sesiones con estado

Se usa para controlar la interacción de los usuarios con aplicaciones

Por ejemplo, en una aplicación de comercio electrónico, una sesión con estado puede almacenar el carrito de compras del usuario para que se pueda acceder a él en diferentes páginas



Sesiones con estado

Puedes almacenar preferencias, configuraciones o historial de un usuario en la sesión, lo que permite adaptar la interfaz y el contenido según las necesidades y preferencias específicas de cada usuario

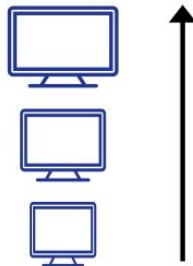


Sesiones con estado - Retos

Las sesiones con estado pueden requerir un **mayor uso de recursos** y pueden ser más **difíciles de escalar** horizontalmente en comparación con las sesiones sin estado.

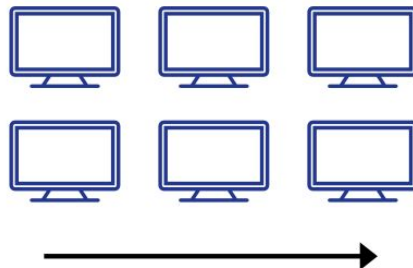
VERTICAL SCALING

Increase size of instance
(RAM, CPU etc.)

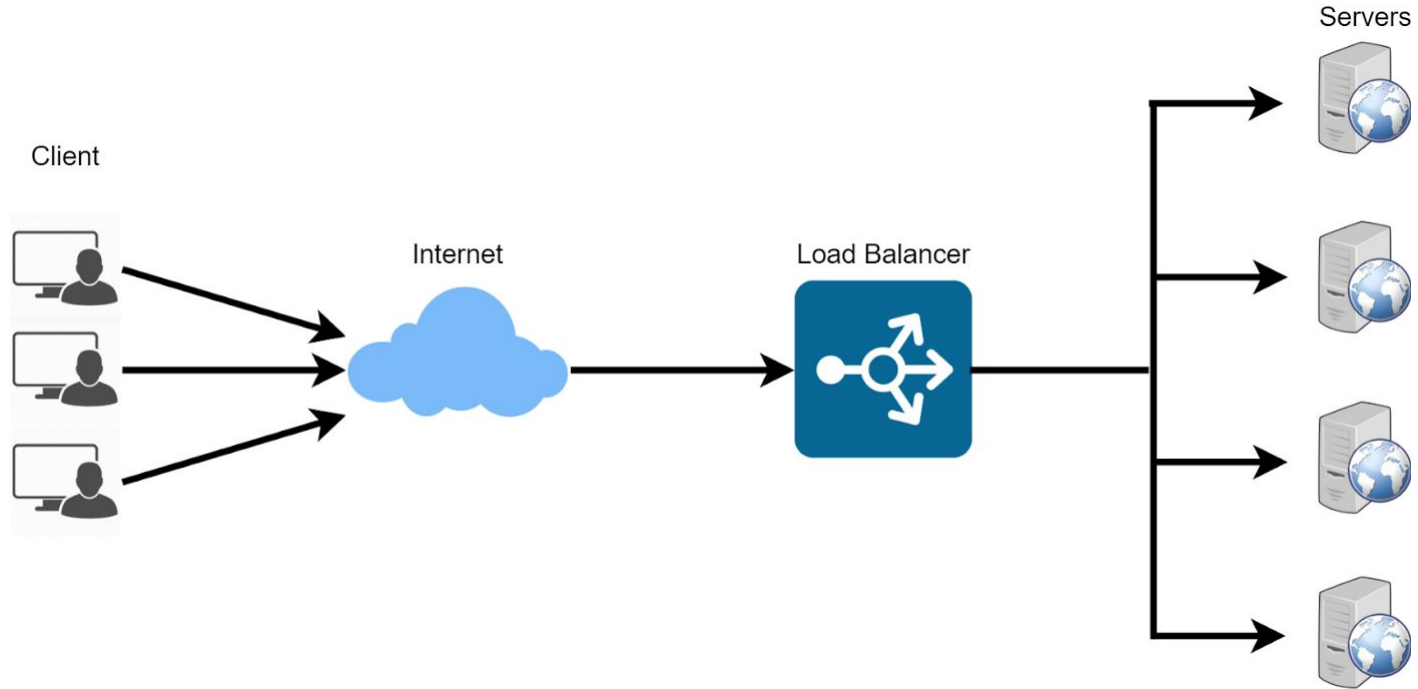


HORIZONTAL SCALING

(Add more instances)

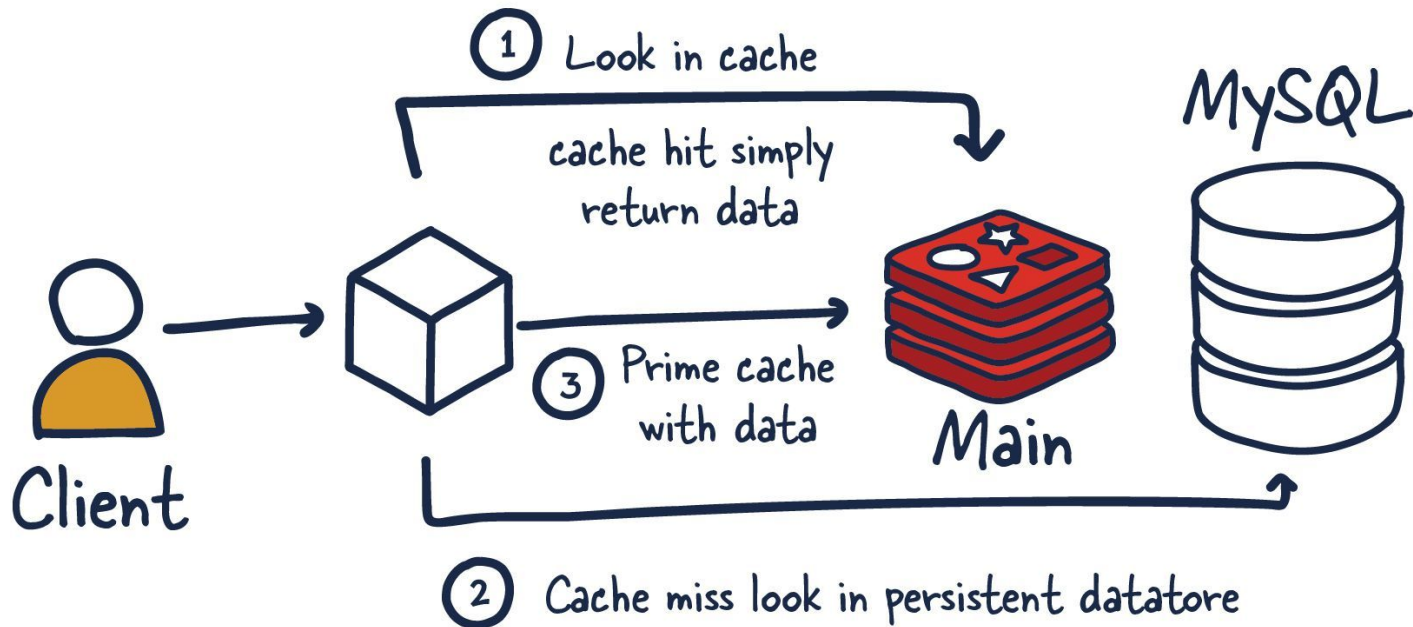


Representación arquitectónica sistema



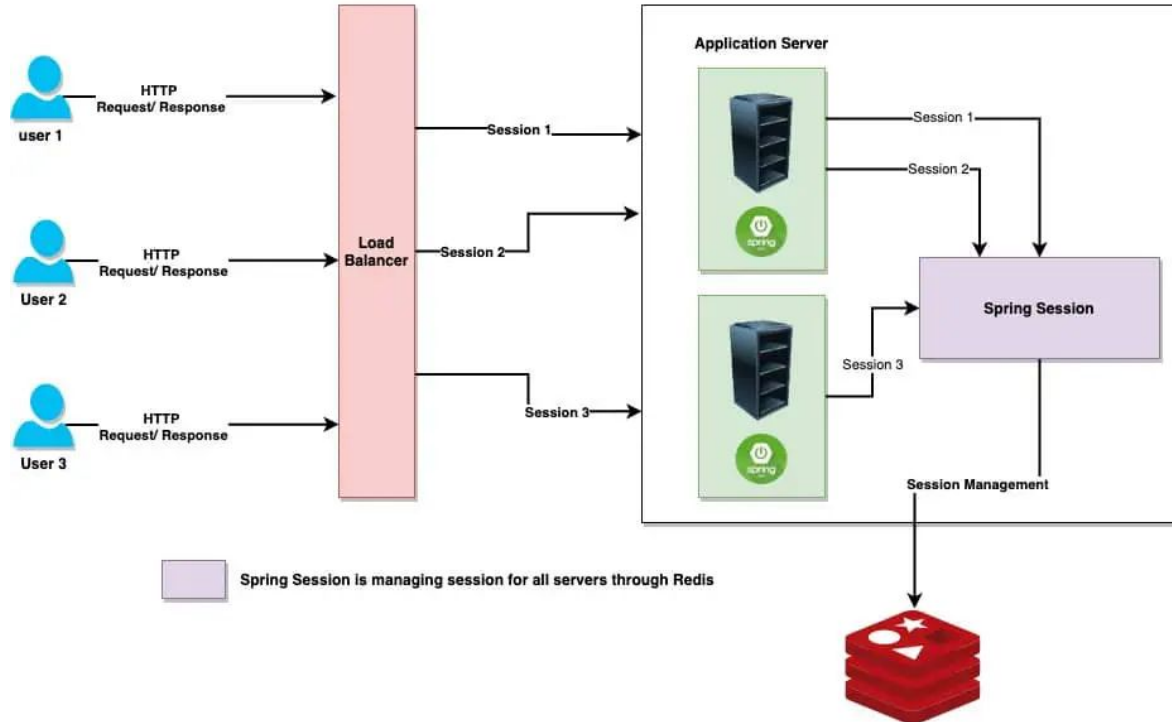
Redis

How is redis traditionally used



Redis - Stateful session

Sessions are not managed individually by each application server but centrally through Spring Session



Sesiones sin estado (Stateless)

¿Como funcionan?

Stateless session - Retos

Confidencialidad: Nada aparte del servidor tendría que ser capaz de interpretar la información de sesión.



Stateless session - Retos

Integridad de datos: Nada aparte del servidor tendría que manipular la información de sesión (accidentalmente o maliciosamente).

La integridad de datos evita:



Stateless session - Retos

Autenticidad: Nada aparte del servidor tendría que ser capaz de iniciar sesiones válidas.



Tokens

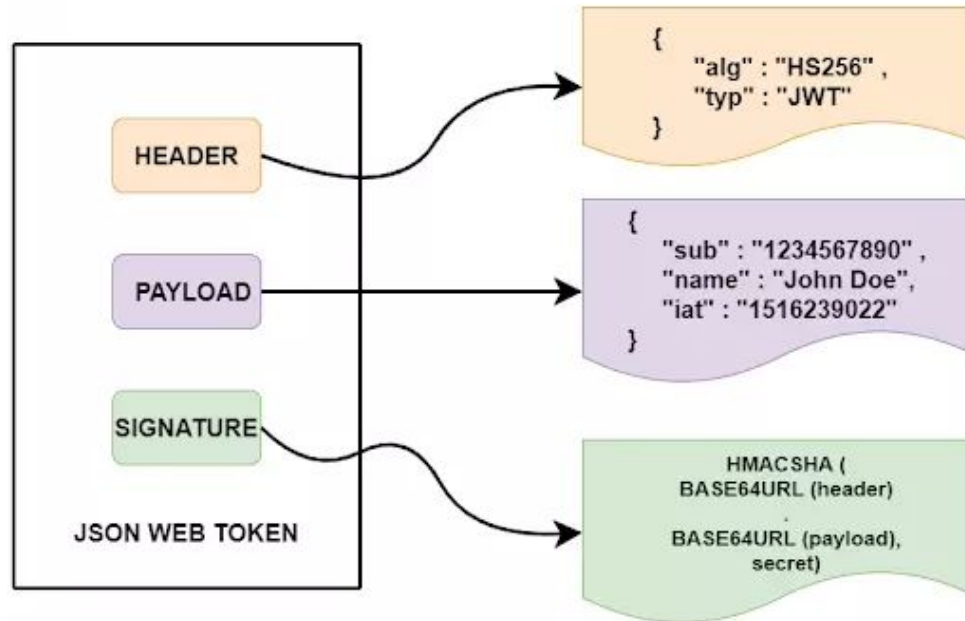
La tokenización se refiere al proceso de sustitución de un elemento de datos sensible por un equivalente no sensible denominado token, que no tiene un significado o valor extrínseco o explotable



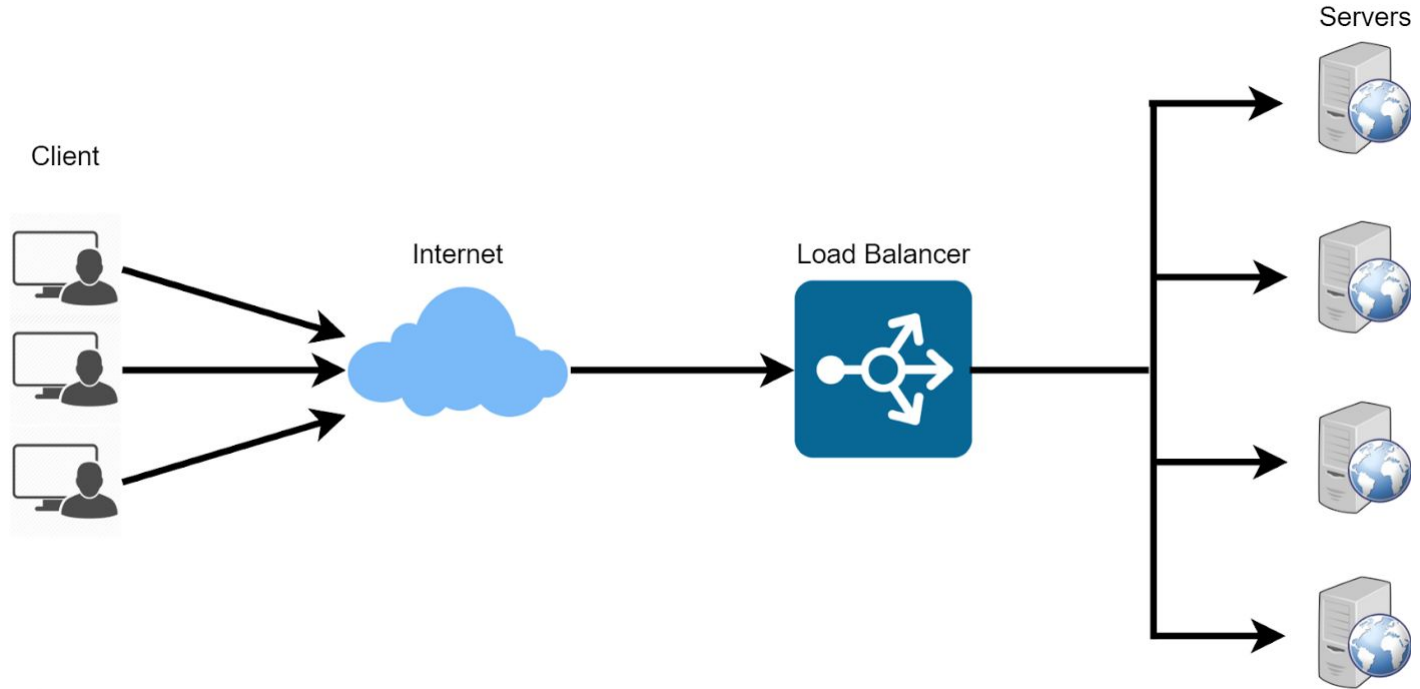
Json Web Token (JWT)



Structure of JSON Web Token (JWT)



Representación arquitectónica sistema



Stateless vs Stateful

Stateless	Stateful
Does not require the server to retain information about the state.	Requires a server to save information about a session.
Server design, implementation and architecture is simple.	Server design, implementation and architecture is complicated.
Handles crashes well, as we can fail over to a completely new server. Servers are regarded as cheap commodity machines.	Does not handle crashes well. Servers are regarded as valuable and long-living. The user would probably be logged out and have to start from the beginning.
Scaling architecture is easy.	Scaling architectures is difficult and complex.

JWT - Ejemplos

Encoded

PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkhvbmGEgTXVuZG8iLCJpYXQiOiE1MTYyMzkwMjJ9.tufSWwUuoZoibtW03YrNhqAYqL7S4ndTVLjmy3kMbi8
```

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

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

PAYLOAD: DATA

```
{
  "sub": "1234567890",
  "name": "Hola Mundo",
  "iat": 1516239022
}
```

VERIFY SIGNATURE

```
HMACSHA256(
  base64UrlEncode(header) + "." +
  base64UrlEncode(payload),
  your-256-bit-secret
) ☐ secret base64 encoded
```

JWT - Integridad de datos y Autenticidad

Es posible conocer cuando el token ha sido alterado o incluso corrompido

Es posible conocer cuando el token ha sido generado usando una llave privada distinta a la que usa nuestro servicio

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpVkd2luIE90YWx2YXJvIiwiaWF0IjoxNTE2MjM5MDIyfQ.F2PH8ImGsAfRIQnk8_a-NhdM8nmEiQi_E5lQrcD9MGU
```

⊗ Invalid Signature

JWT - Integridad de datos y Autenticidad

Cuando el token no es válido se retorna un código de error http 401



JWT - Codificado

Es posible ver la información del token
sin conocer la llave privada

Decoded

[EDIT THE PAYLOAD AND SECRET](#)

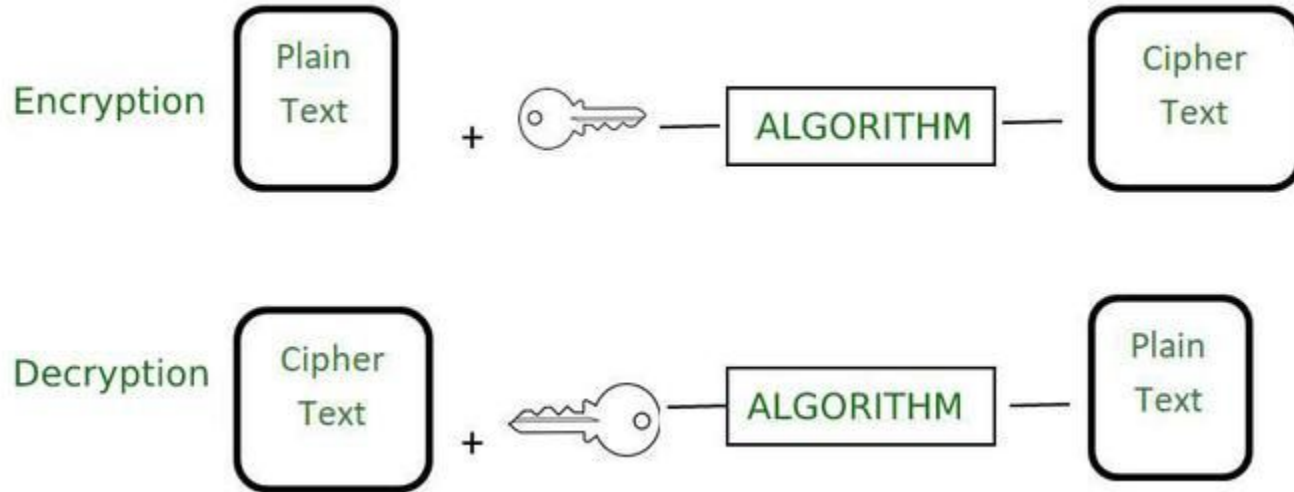
HEADER: ALGORITHM & TOKEN TYPE

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

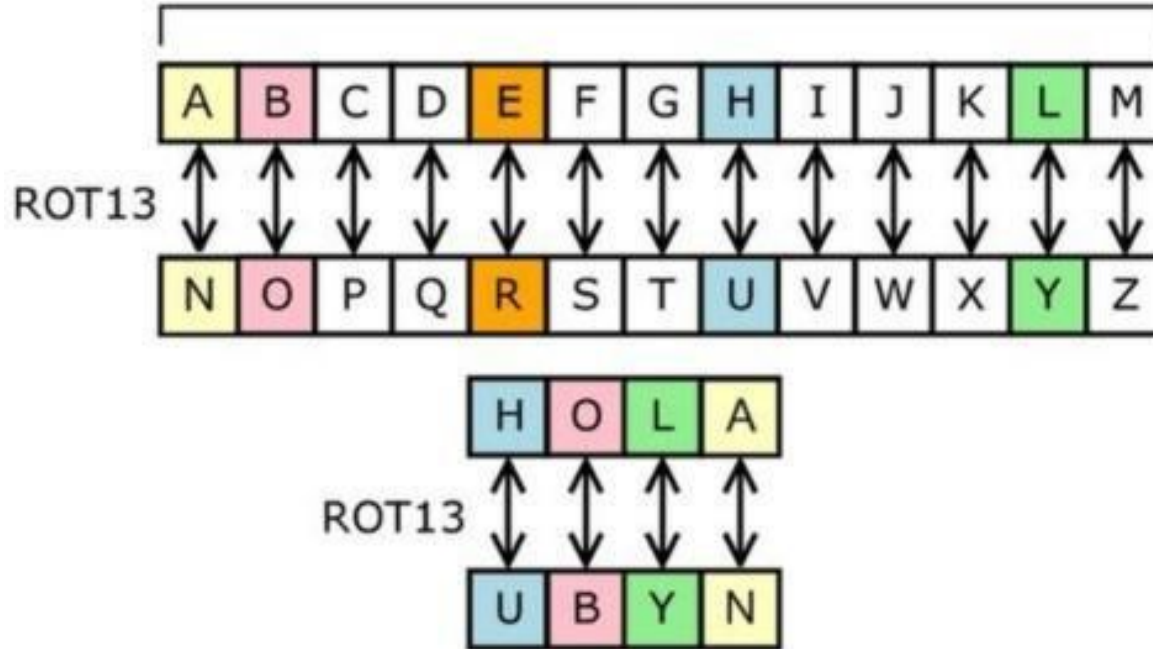
PAYLOAD: DATA

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "iat": 1516239022,  
  "camposecreto": "informacion ultra secreta"  
}
```

Encriptado



Encriptado - Ejemplo Cifrado del Cesar



JWT - La información es pública

En los JWT la información del payload es pública. **Nunca**
agregar información confidencial a estos tokens



Implementemos JWT en nodejs

Let's Play



TAREA PARA LA CLASE 9

- Modificar el modelo de User para el password.
- Modificar el endpoint de login para que reciba el id del usuario y el password; y lo compare con el password guardado en la base de datos antes de crear el token de sesión