
WOMEN WHO
CODE®
/medellín

FROM HERO TO SUPERHERO

BACKEND CON NODEJS

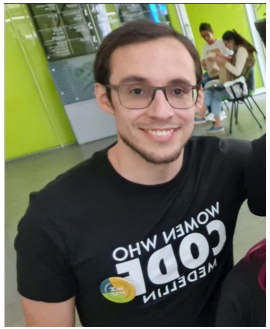
AGRADECER A...

Nuestro patrocinador y tutores voluntarios

PATROCINADOR

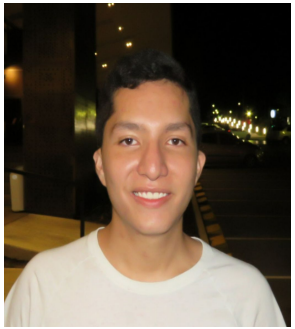
softserve

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 **RESPETO**

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

TABLE OF CONTENTS

GLOSARIO

Repasemos términos qee usaremos antes de iniciar

Glosario

01

Autenticacion

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

02

Autorizacion

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

03

Token

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

04

Sesión

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

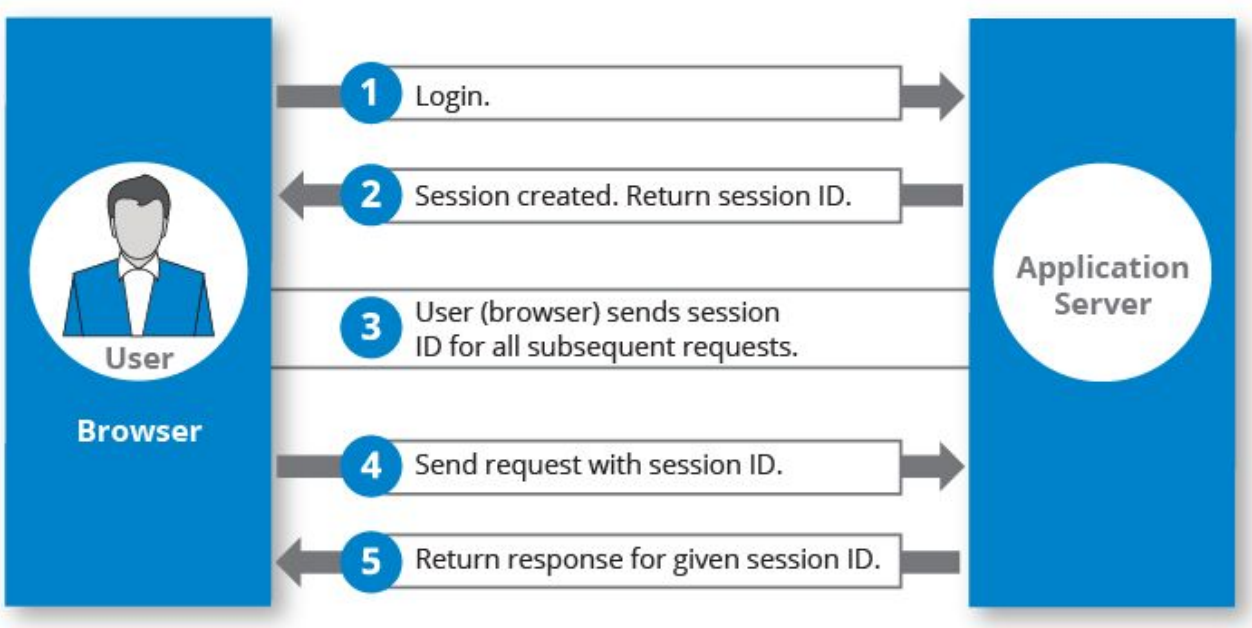
La autenticación evita que cualquiera pueda entrar en un determinado sistema o iniciar sesión en alguna plataforma de forma indebida, sin que realmente sea el usuario legítimo que tiene el poder para hacerlo.

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

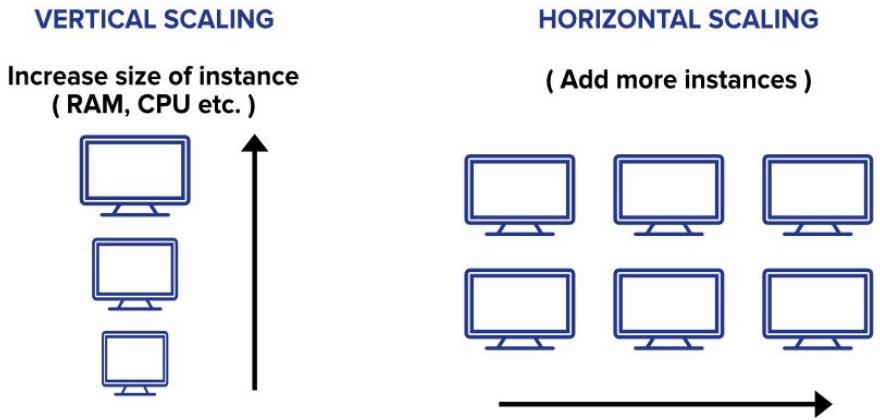
Por ejemplo, en una aplicación de comercio electrónico, una sesión con estado puede almacenar el carrito de compras del usuario para que se pueda acceder a él en diferentes páginas

Sesiones con estado

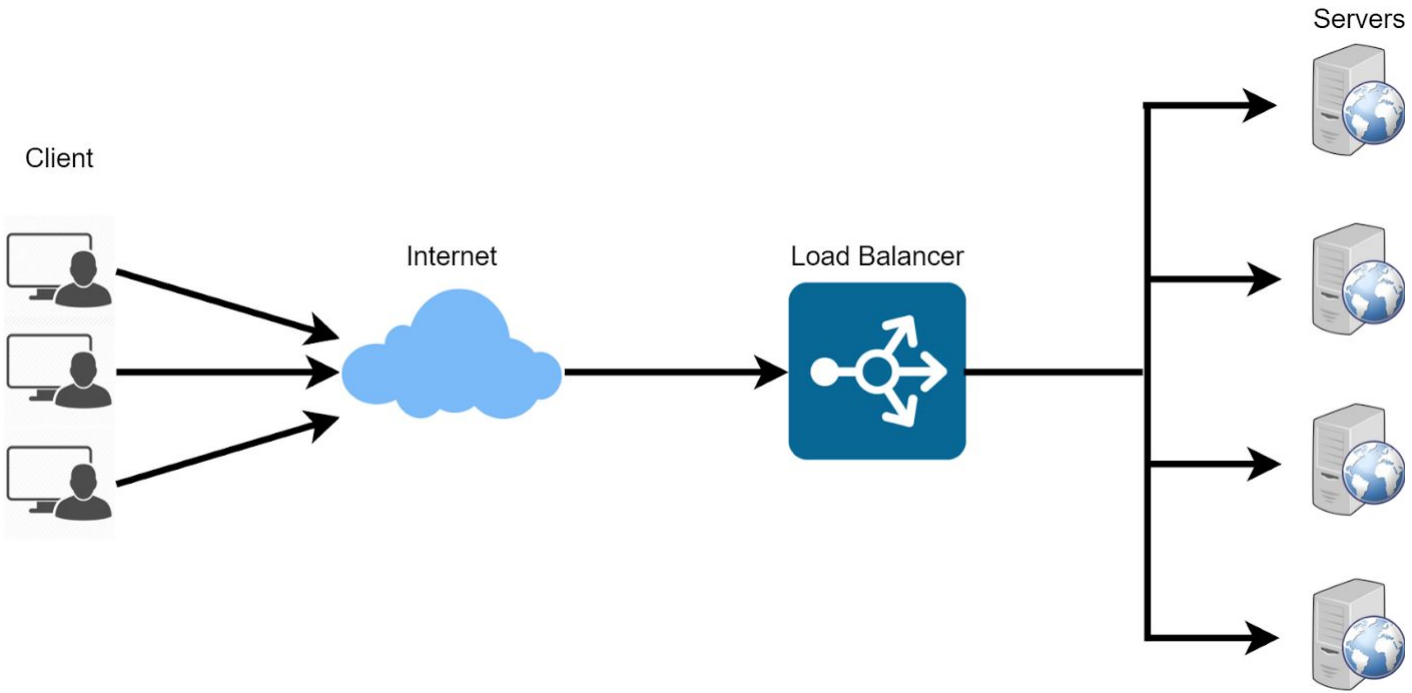
Puedes almacenar preferencias, configuraciones o historial de un usuario en la sesión, lo que permite adaptar la interfaz y el contenido según las necesidades y preferencias específicas de cada usuario

Sesiones con estado - Retos

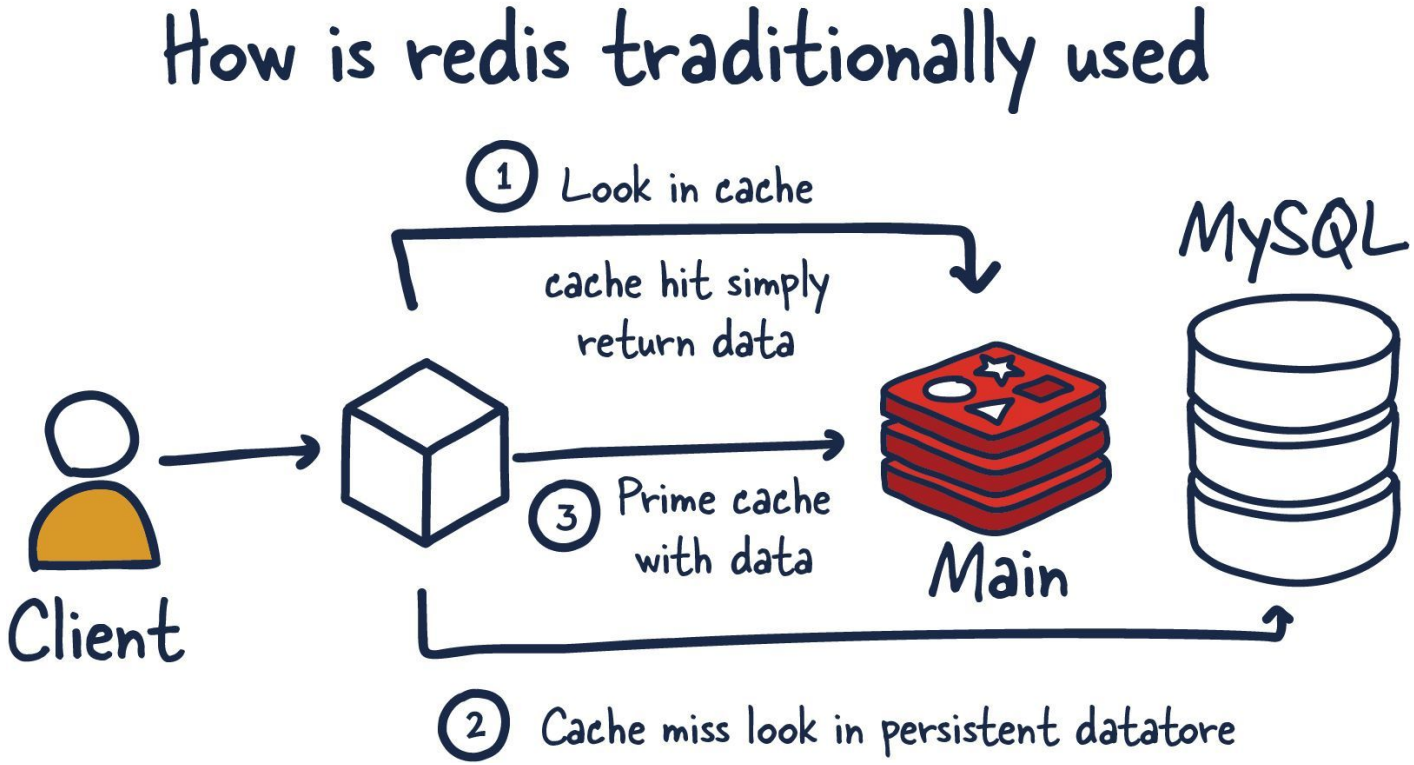
Las sesiones con estado pueden requerir un **mayor uso de recursos** y pueden ser más **difíciles de escalar** horizontalmente en comparación con las sesiones sin estado.



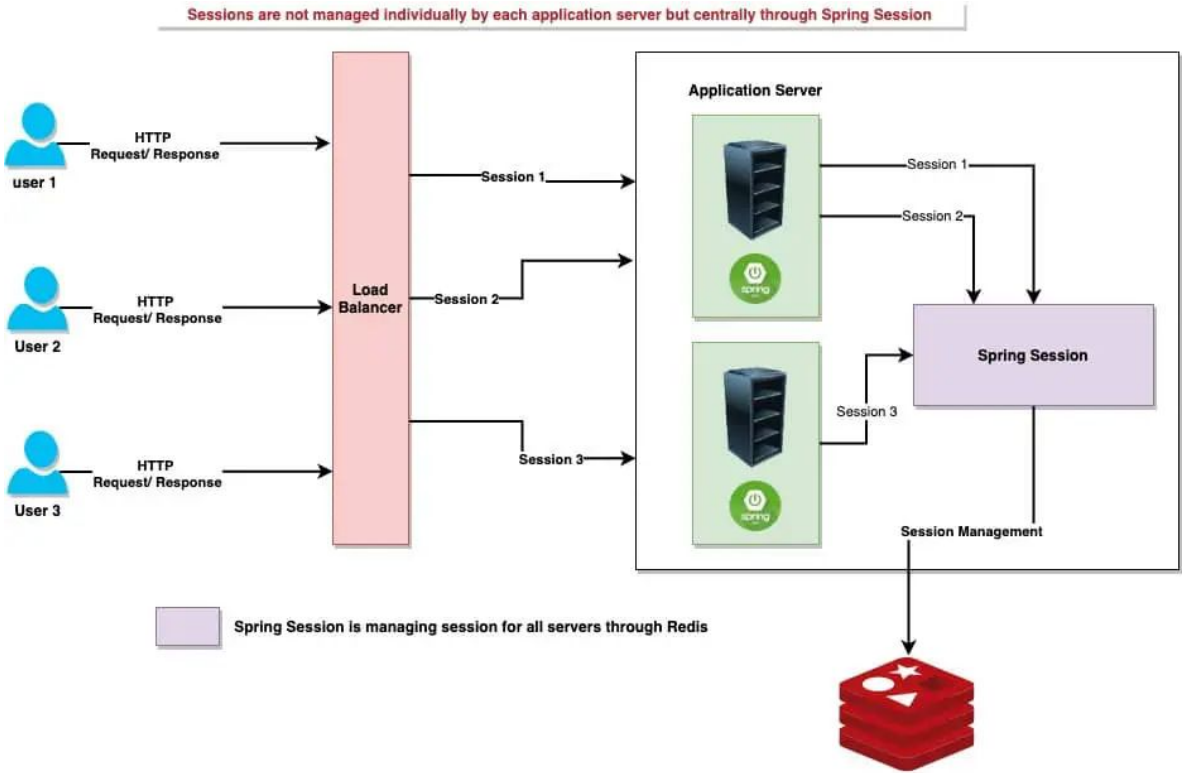
Representación arquitectónica sistema



Redis


How is redis traditionally used
1 Look in cache
cache hit simply return data
MySQL
Client
3 Prime cache with data
Main
2 Cache miss look in persistent datatore

Redis - Stateful session



Sesiones sin estado (Stateless)

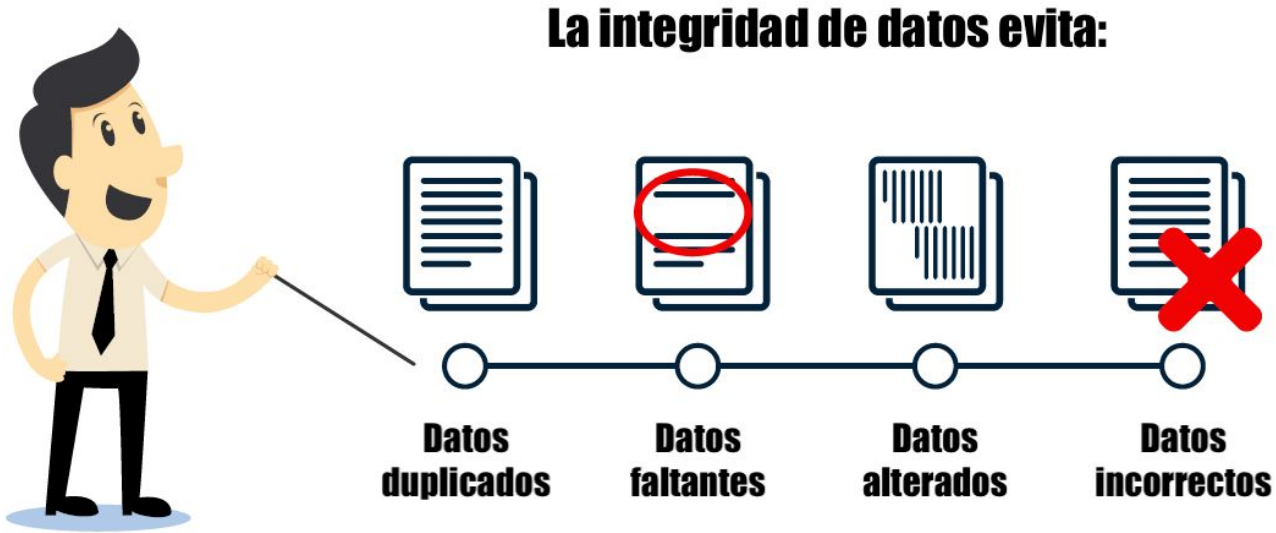
¿Como funcionan?

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).



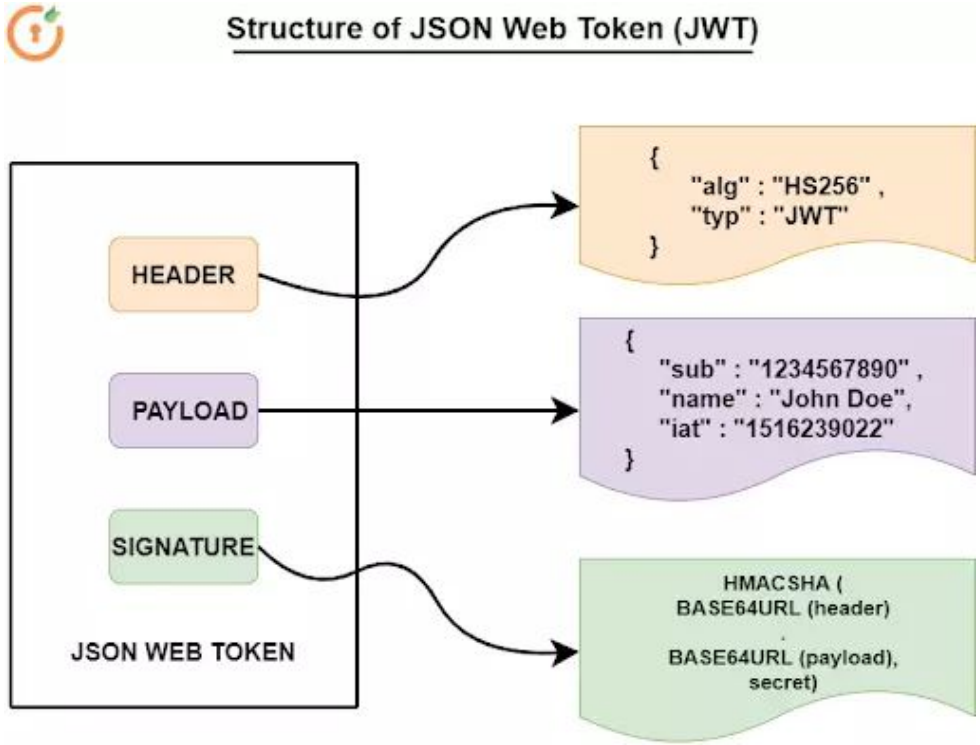
Stateless session - Retos

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

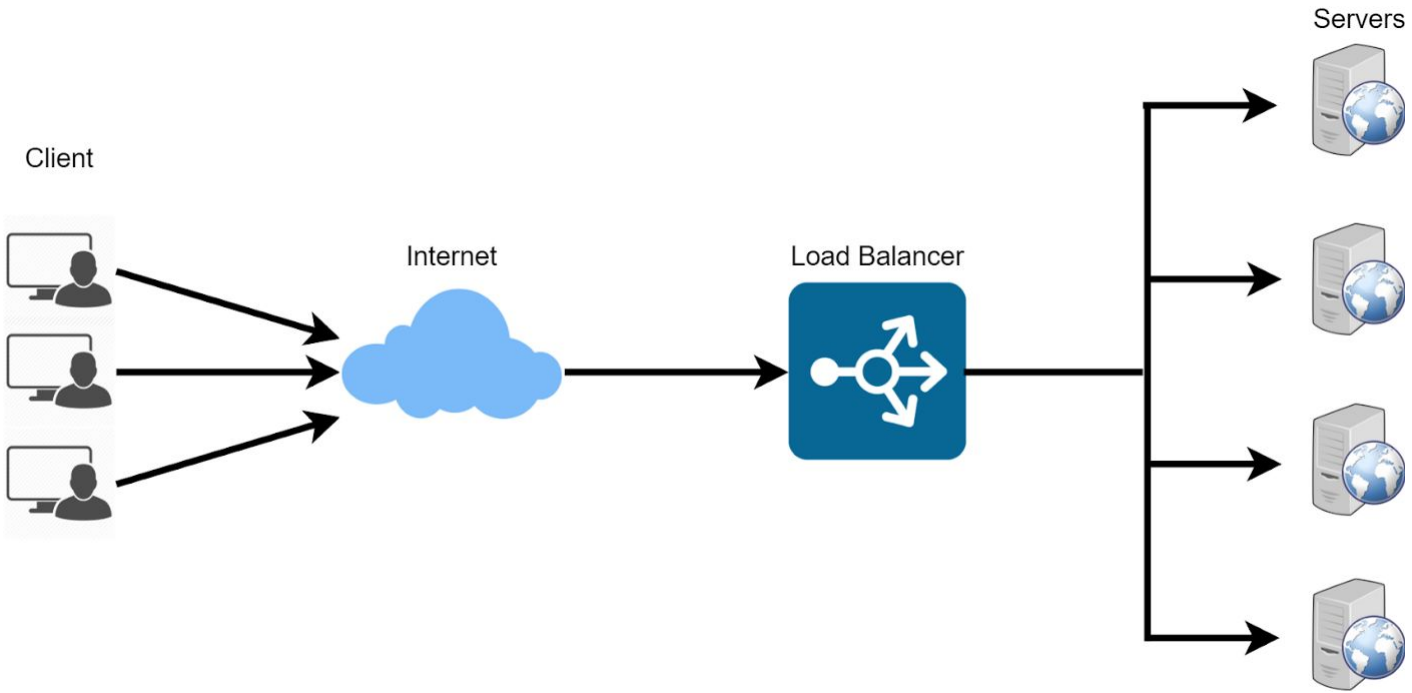
Tokens

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

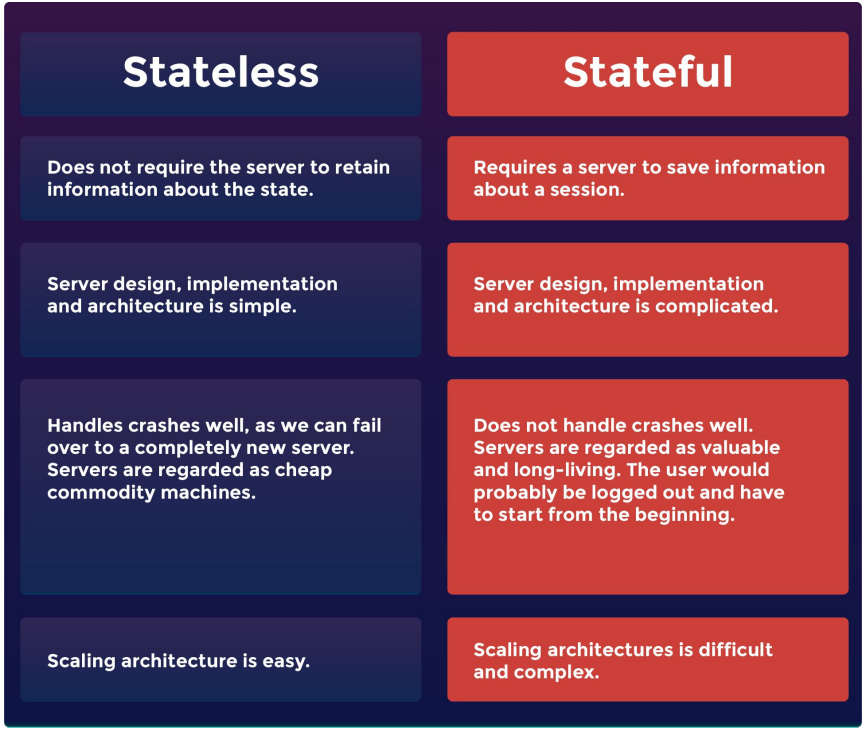
Json Web Token (JWT)



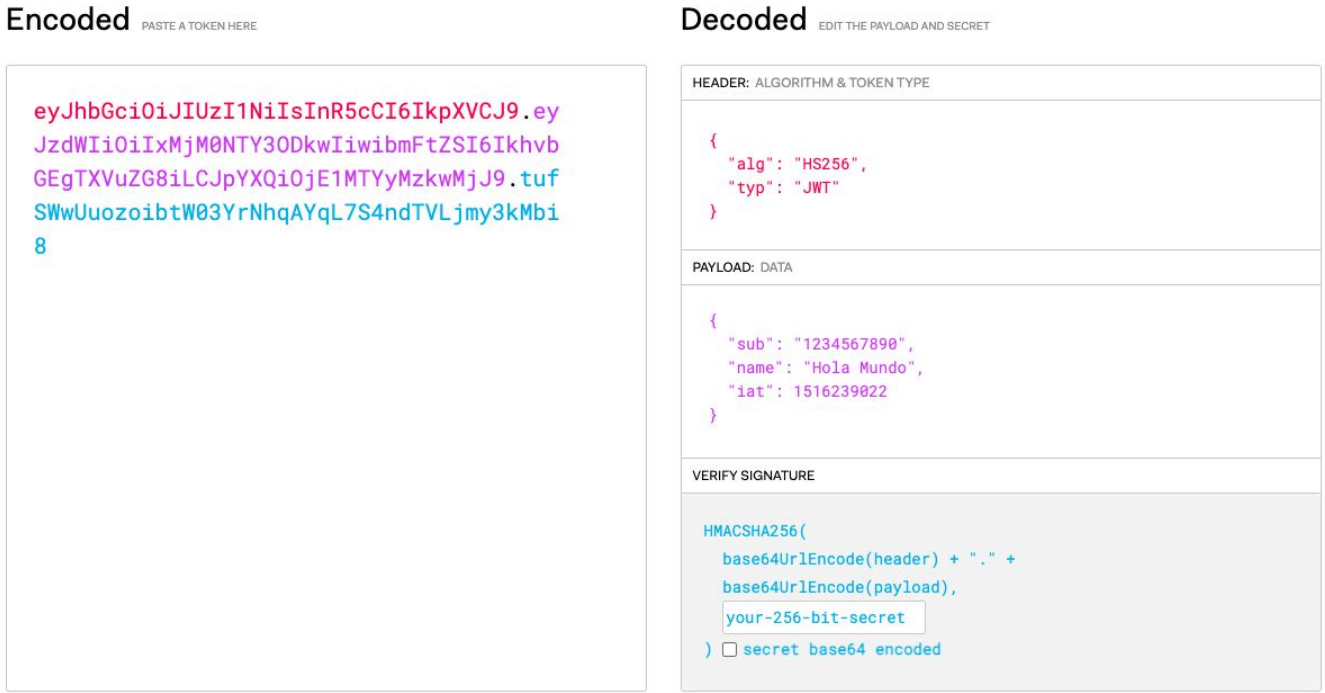
Representación arquitectónica sistema



Stateless vs Stateful



JWT - Ejemplos

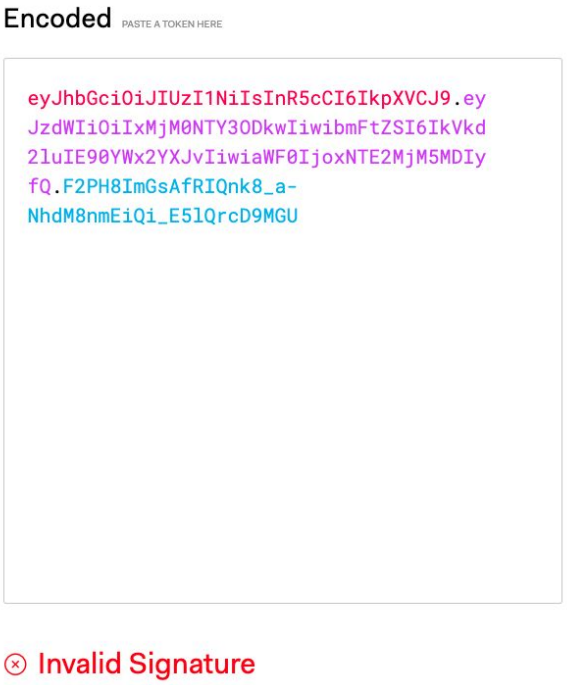


WOMEN WHO CODE /medellín

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

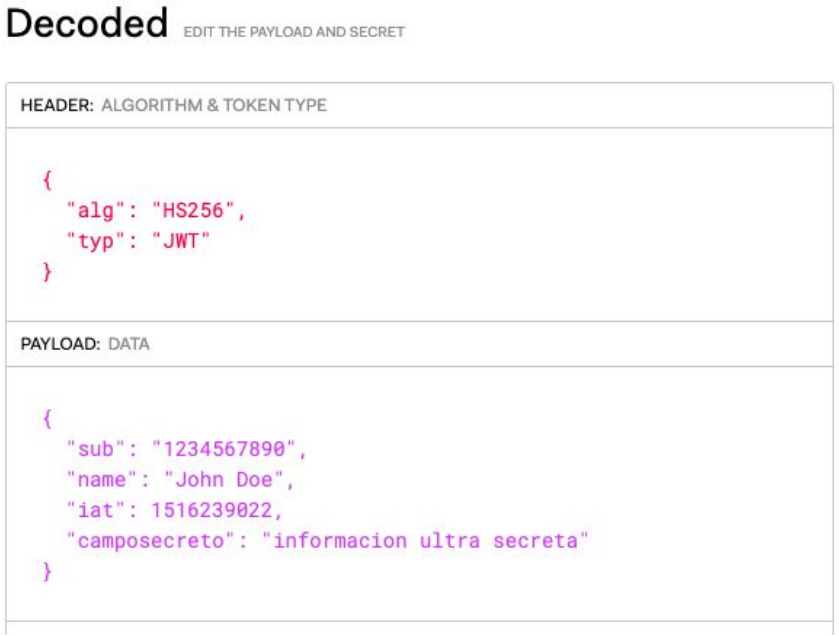


JWT - Integridad de datos y Autenticidad

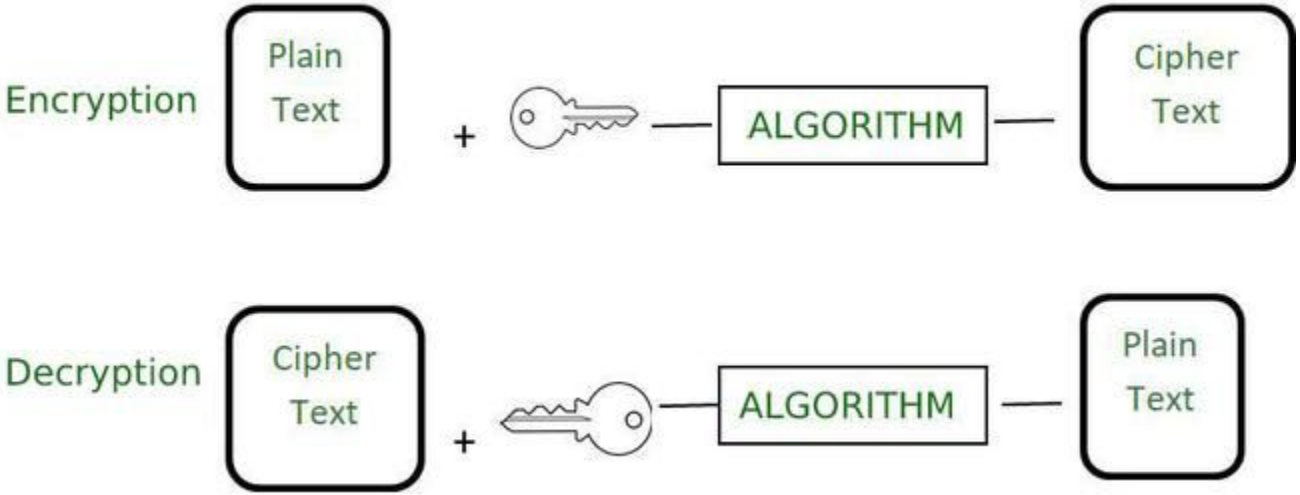
Cuando el token no es válido se retorna un código de error http 401

JWT - Codificado

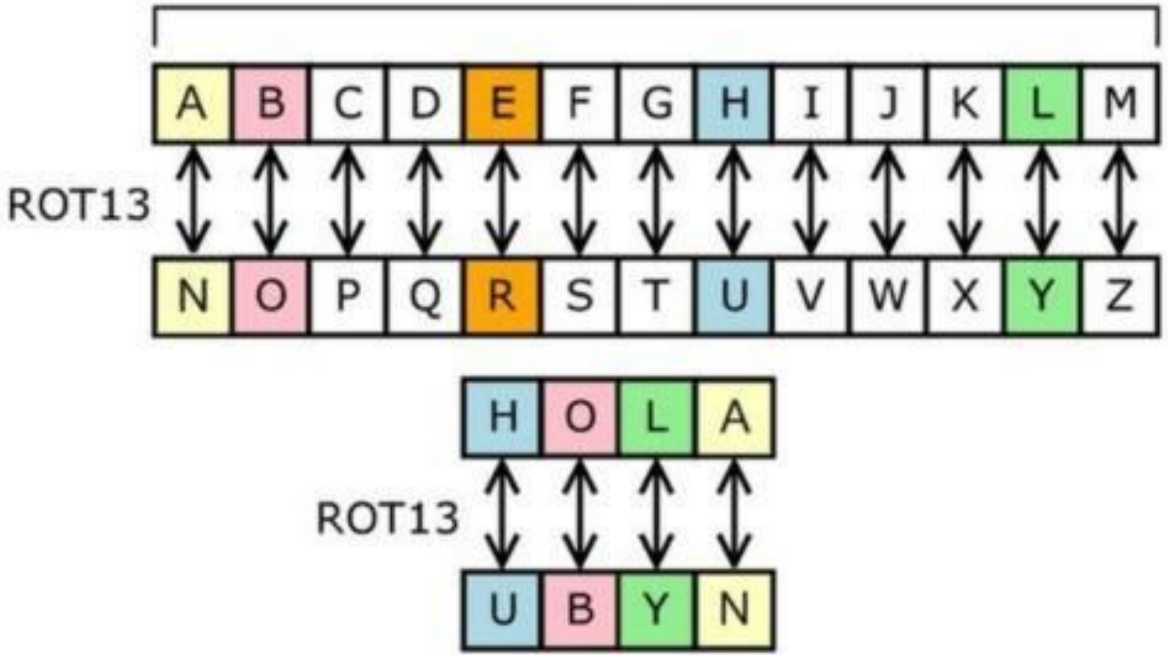
Es posible ver la información del token sin conocer la llave privada



Encriptado


Encryption
Plain Text
+
ALGORITHM
Cipher Text
Decryption
Cipher Text
+
ALGORITHM
Plain Text

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