# Seguridad

Existen varios sistemas de autenticación en una aplicación web, el sistema más tradicional utilizado hasta el momento funcionaba guardando los datos de la sesión de usuario y lo almacenaba posteriormente esa información en una base de datos como pueden ser Mysql o sql, luego para autenticar al usuario propiamente dicho se realizaba una request a esa base de datos y se comparaba estos valores para verificar si el usuario es quien dice ser. Esta metodología conlleva algunas desventajas propias de la implementación como ser la lentitud, el gasto de recursos entre otras cosas.

Las nuevas tendencias en cuanto al desarrollo web moderno se refiere, es la autenticación por medio de Token, a continuación se detalla el funcionamiento básico de la metodología

El funcionamiento es el siguiente. El usuario se autentica en nuestra aplicación, bien con un par usuario/contraseña, A partir de entonces, cada petición HTTP que haga el usuario va acompañada de un Token en la cabecera. Este Token no es más que una firma cifrada que permite a nuestro API identificar al usuario. Pero este Token no se almacena en el servidor, si no en el lado del cliente (por ejemplo en localStorage o sessionStorage) y el API es el que se encarga de descifrar ese Token y redirigir el flujo de la aplicación en un sentido u otro. Como los tokens son almacenados en el lado del cliente, no hay información de estado y la aplicación se vuelve totalmente escalable. Podemos usar el mismo API para diferentes apliaciones (Web, Mobile, Android, iOS, ...) solo debemos preocuparnos de enviar los datos en formato JSON y generar y descifrar tokens en la autenticación y posteriores peticiones HTTP a través de un middleware.

También nos añade más seguridad. Al no utilizar cookies para almacenar la información del usuario, podemos evitar ataques CSRF (Cross-Site Request Forgery) que manipulen la sesión que se envía al backend. Por supuesto podemos hacer que el token expire después de un tiempo lo que le añade una capa extra de seguridad.

# Autenticación con JSON Web Tokens

En este caso particular se utilizó JWT para autenticar ***pickupmeal.com,***  a continuación se detalla la forma en que se genera este token.

El formato de un JWT está compuesto por 3 string separados por un punto. Donde cada String tiene un significado

La primera parte del token corresponde al **Header,** esta consta a su vez de otras dos partes, el tipo de token utilizado y el algoritmo por el cual éste es generado.

La segunda parte corresponde al **payload** está compuesto por los llamados JWT Claims donde irán colocados la atributos que definen nuestro token. Exiten varios que puedes consultar aquí, los más comunes a utilizar son:

* sub: Identifica el sujeto del token, por ejemplo un identificador de usuario.
* iat: Identifica la fecha de creación del token, válido para si queremos ponerle una fecha de caducidad. En formato de tiempo UNIX
* exp: Identifica a la fecha de expiración del token. Podemos calcularla a partir del iat. También en formato de tiempo UNIX.

Ademas de estos datos se pueden añadir mas campos, generalmente, en esta parte del token van ubicados los atributos del **User**como el nombre, apellido , dirección entre otras cosas

Como ultima parte del JWT ubicamos la **firma** Está formada por los anteriores componentes (Header y Payload) cifrados en Base64 con una clave secreta (almacenada en nuestro backend). Así sirve de Hash para comprobar la autenticidad del usuario.

