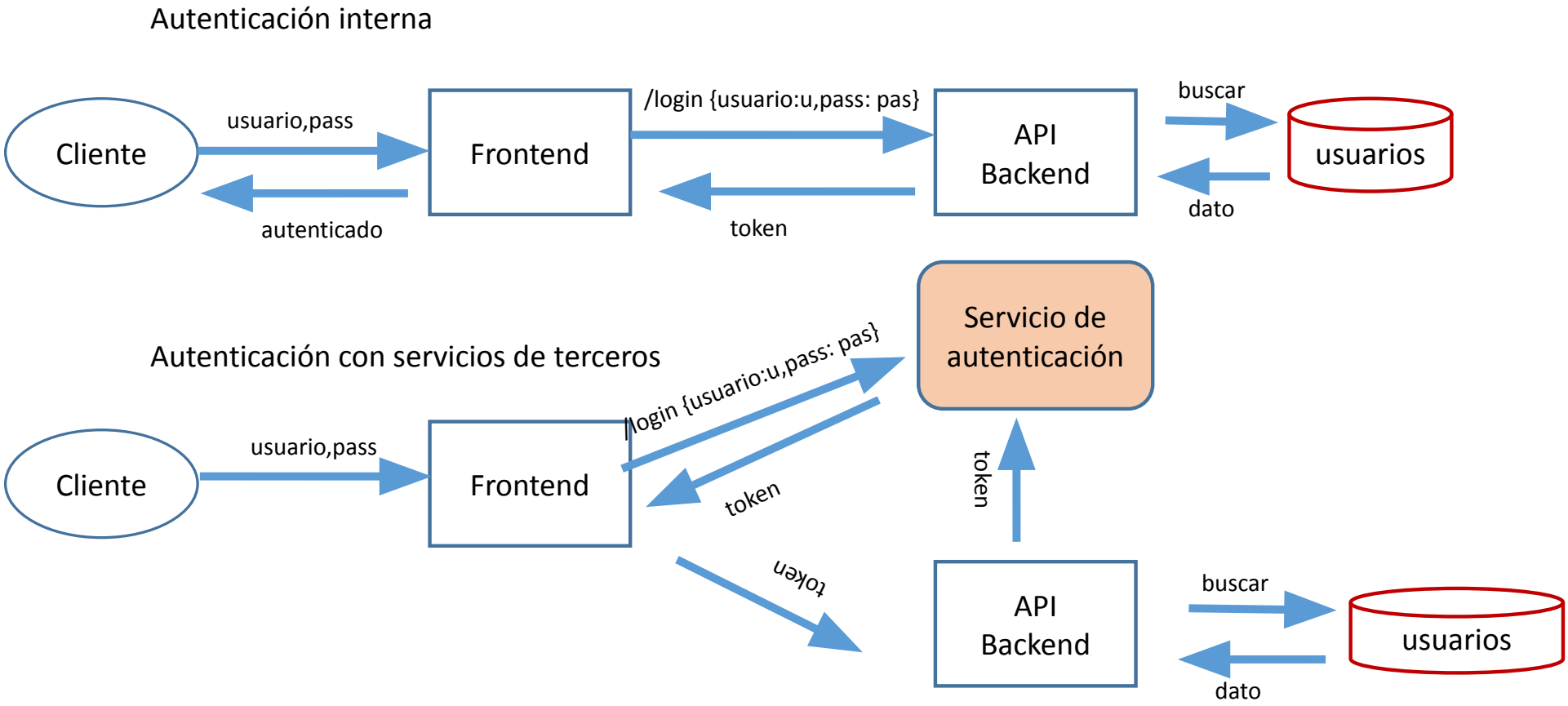
Autenticación en el backend

Algunas formas de autenticación



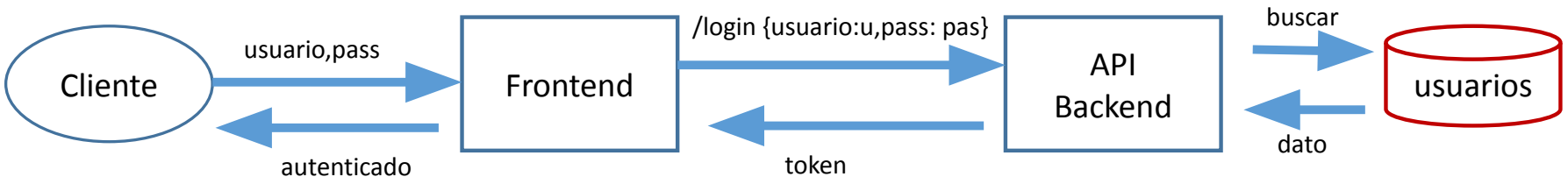
JSON Web Token (JWT)

- Es un estándar abierto (RFC 7519) que define una forma compacta de transmitir información segura entre las partes, como un objeto JSON
- Funcionamiento en un esquema de autorización:
  - El servidor genera un JWT que certifica la identidad del usuario y lo envía al cliente.
  - El cliente enviará el token de vuelta al servidor para cada solicitud posterior, por lo que el servidor sabe que la solicitud proviene de una identidad particular autorizada.

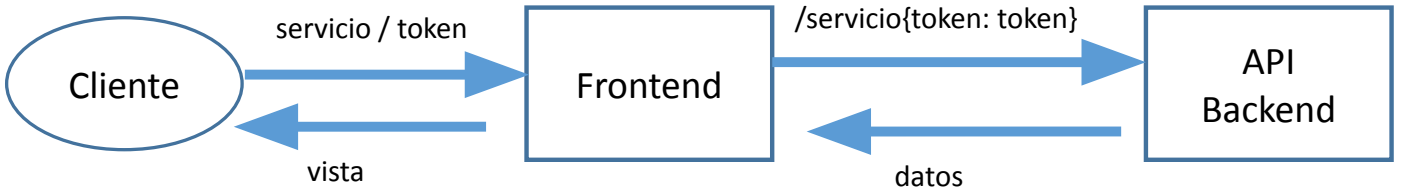
Aprender más en: https://jwt.io/

Autorización de servicios

Autenticación



Autorización



https://www.npmjs.com/package/passport

Passport

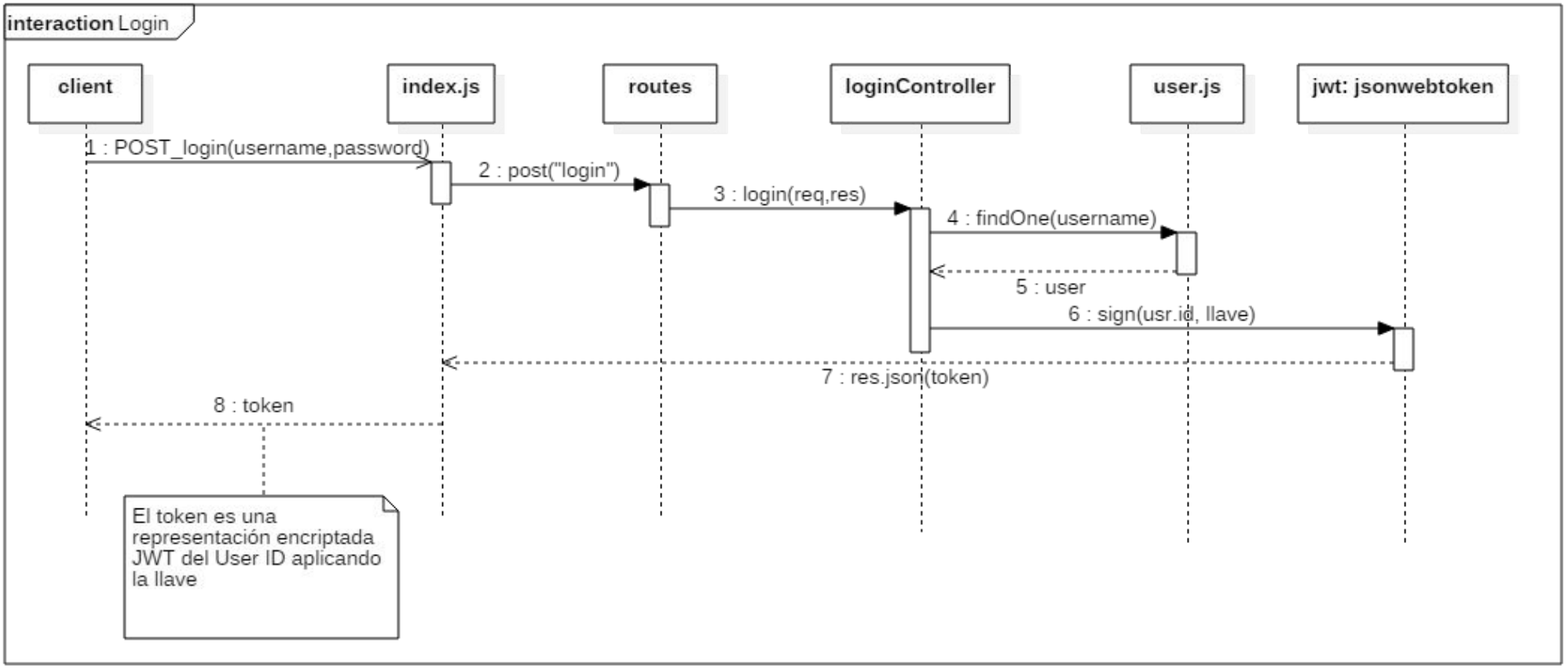
Passport is **Express**-compatible authentication middleware for **Node.js**.

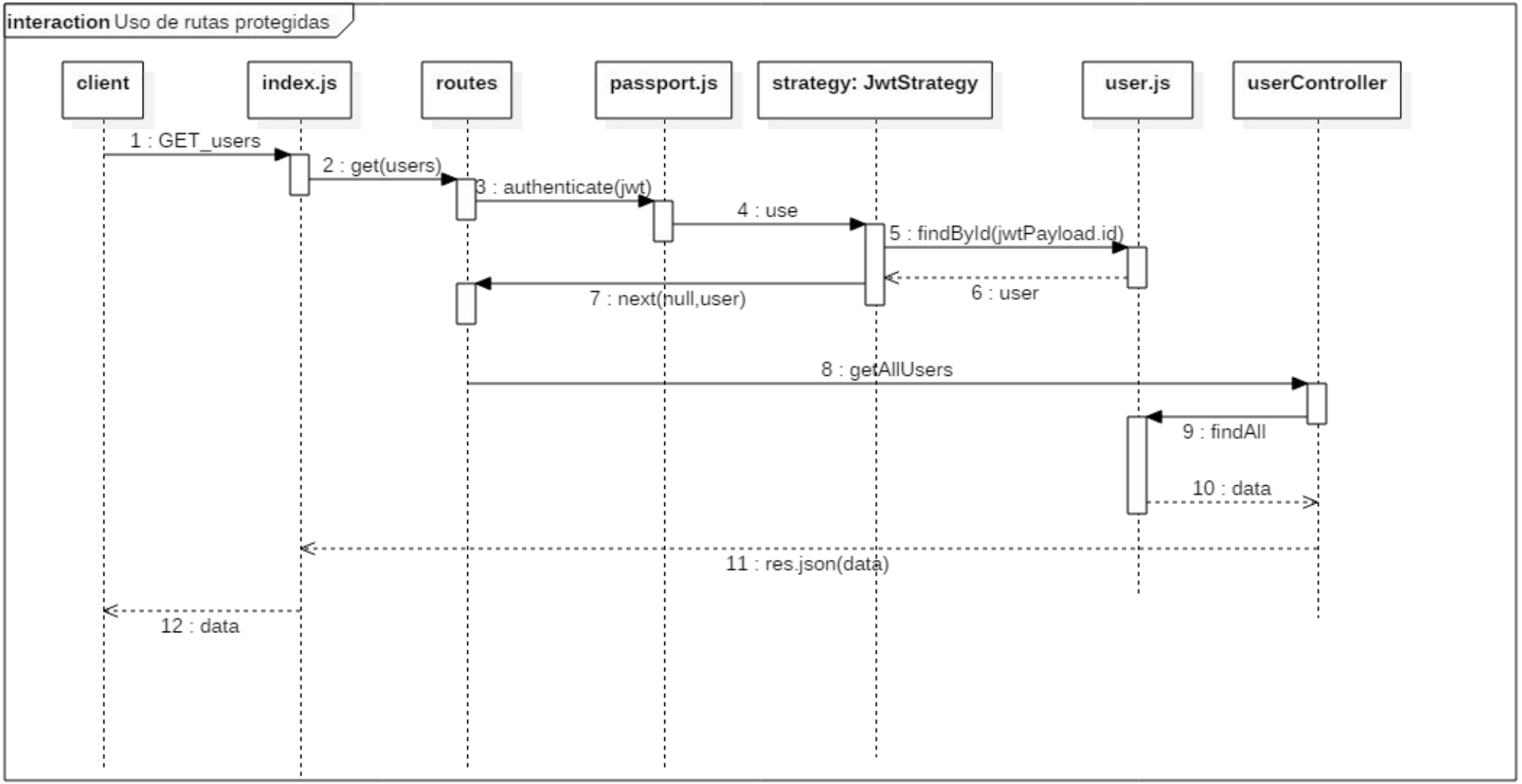
Passport's sole purpose is to authenticate requests, which it does through an extensible set of plugins known as *strategies*. Passport does not mount routes or assume any particular database schema, which maximizes flexibility and allows application-level decisions to be made by the developer. The API is simple: you provide Passport a request to authenticate, and Passport provides hooks for controlling what occurs when authentication succeeds or fails.

Status: build passing | coverage 99% | dependencies up to date

Install

```
$ npm install passport
```


interaction Login
client
index.js
routes
loginController
user.js
jwt: jsonwebtoken
1 : POST_login(username,password)
2 : post("login")
3 : login(req,res)
4 : findOne(username)
5 : user
6 : sign(usr.id, llave)
7 : res.json(token)
8 : token
El token es una representación encriptada JWT del User ID aplicando la llave


interaction Uso de rutas protegidas
client
index.js
routes
passport.js
strategy: JwtStrategy
user.js
userController
1 : GET_users
2 : get(users)
3 : authenticate(jwt)
4 : use
5 : findById(jwtPayload.id)
6 : user
7 : next(null,user)
8 : getAllUsers
9 : findAll
10 : data
11 : res.json(data)
12 : data

Instalar módulos en el backend

npm install jsonwebtoken

npm install passport

npm install passport-jwt

Crear el módulo passport.js

```
const passport = require('passport');
const passportJWT = require('passport-jwt');
const User = require("../models/user");
const llave = require("./llave");// ←—-En este archivo está la llave con la que se
                                    codifica el token
let ExtractJwt = passportJWT.ExtractJwt;
let JwtStrategy = passportJWT.Strategy;
let jwtOptions = {};
jwtOptions.jwtFromRequest = ExtractJwt.fromAuthHeaderAsBearerToken();
jwtOptions.secretOrKey = llave;
let strategy = new JwtStrategy(jwtOptions, function(jwt_payload, next) {
  let user = User.findById(jwt_payload.id ); // ←—-Aquí verifican el dato
                                                codificado en el token
  if (user) {
    next(null, user); //←- Si el token es válido el requerimiento pasa al
controlador
  } else {
    next(null, false);//←- Si el token es inválido enviará el “unauthorized”
  }
});
passport.use(strategy);
module.exports = passport;
```