

Manual Ava

Por: Jhonathan Miceli Landa 183381

¿Para qué sirve?

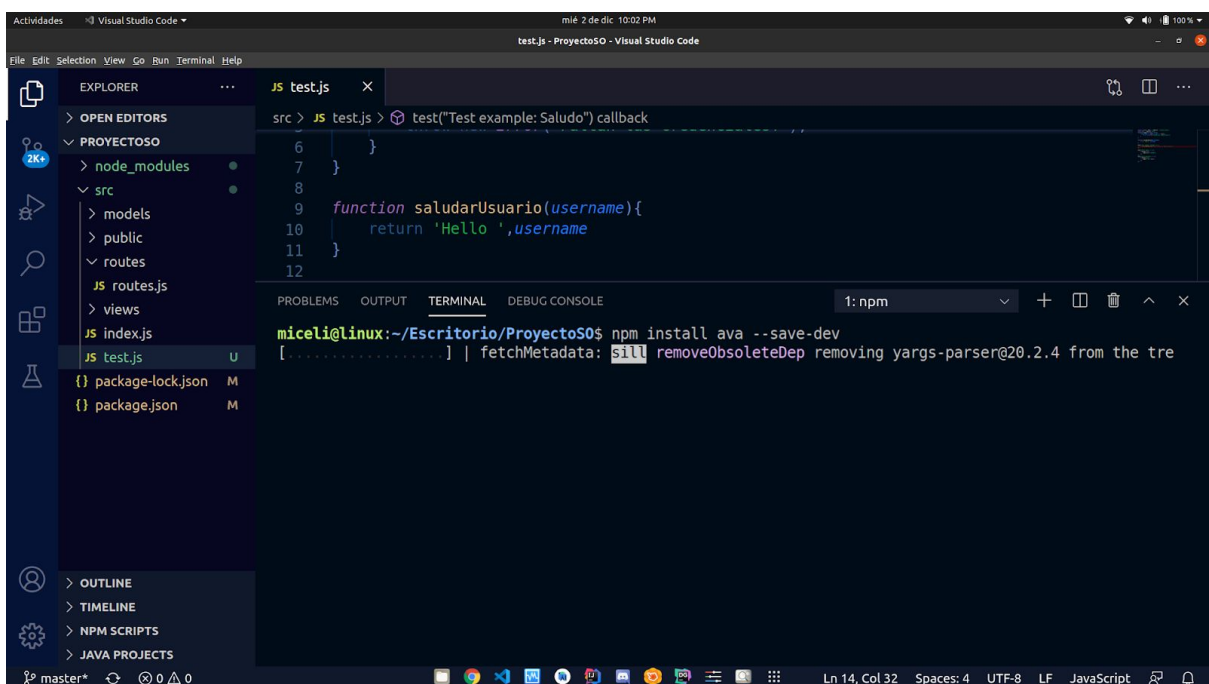
AVA es un *test runner* para Node.js que ofrece una API concisa, un detallado informe de errores, y soporte para nuevas características de lenguaje y aislamiento de procesos. AVA es el más adecuado para probar módulos Node.js y aplicaciones de servidor, pero quizás no para probar aplicaciones de IU. Entre sus capacidades, AVA ofrece una función para marcar una prueba como una tarea pendiente. Cada archivo de prueba ejecutado es un proceso separado. Otros beneficios incluyen el minimalismo y la velocidad de AVA, la sintaxis de prueba simple y la capacidad de ejecutar pruebas al mismo tiempo. AVA también tiene soporte para funciones asíncronas y observables, un observable es un tipo que se puede usar para modelar fuentes de datos basadas en *push*, como los eventos DOM.

Tipo de herramienta: Para la Ejecución y Registro de pruebas.

- Proceso de elaboración de pruebas.

Instalación Ava

Instalar AVA usando npm. Antes que nada, deberás tener [Node.js](#) instalado. Ava se instala con el siguiente comando desde la terminal: **npm install ava --save-dev**



The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays a project structure with folders like 'node_modules', 'src', 'models', 'public', 'routes', 'views', and files like 'routes.js', 'index.js', 'test.js', 'package-lock.json', and 'package.json'. The main editor area shows a JavaScript file named 'test.js' with the following code:

```
src > JS test.js > test("Test example: Saludo") callback
6   }
7   }
8
9   function saludarUsuario(username){
10     return 'Hello ',username
11   }
12
```

The TERMINAL panel at the bottom shows the command 'npm install ava --save-dev' being executed. The output indicates that the package is being installed and that 'yargs-parser@20.2.4' is being removed from the tree.

```
miceli@linux:~/Escritorio/ProyectoSO$ npm install ava --save-dev
[.....] | fetchMetadata: sill removeObsoleteDep removing yargs-parser@20.2.4 from the tree
```

una vez instalado debería aparecer en el “package-lock.json” de tu proyecto de esta manera:

```
"ava": {
  "version": "3.13.0",
  "resolved": "https://registry.npmjs.org/ava/-/ava-3.13.0.tgz",
  "integrity": "sha512-yzky+gark5PdsFFlZ4CnBVxm/OgBUWtn9vAsSSnuooVJNOk5ER17HJXVeUzy63LIt06Zy34oThcn+2ZqgMs7SA==",
  "dev": true,
```

Para continuar con las pruebas debemos crear un archivo test.js en la raíz del proyecto y requerir “ava” de la siguiente manera:

```
const test = require('ava')
```

Ahora debemos importar las funciones que deseamos probar en mi caso son 3 que deseo poner a prueba, una que verifique username que no sea de tipo null, otra de password que no sea null y por último una función que retorne un saludo con el nombre del usuario.

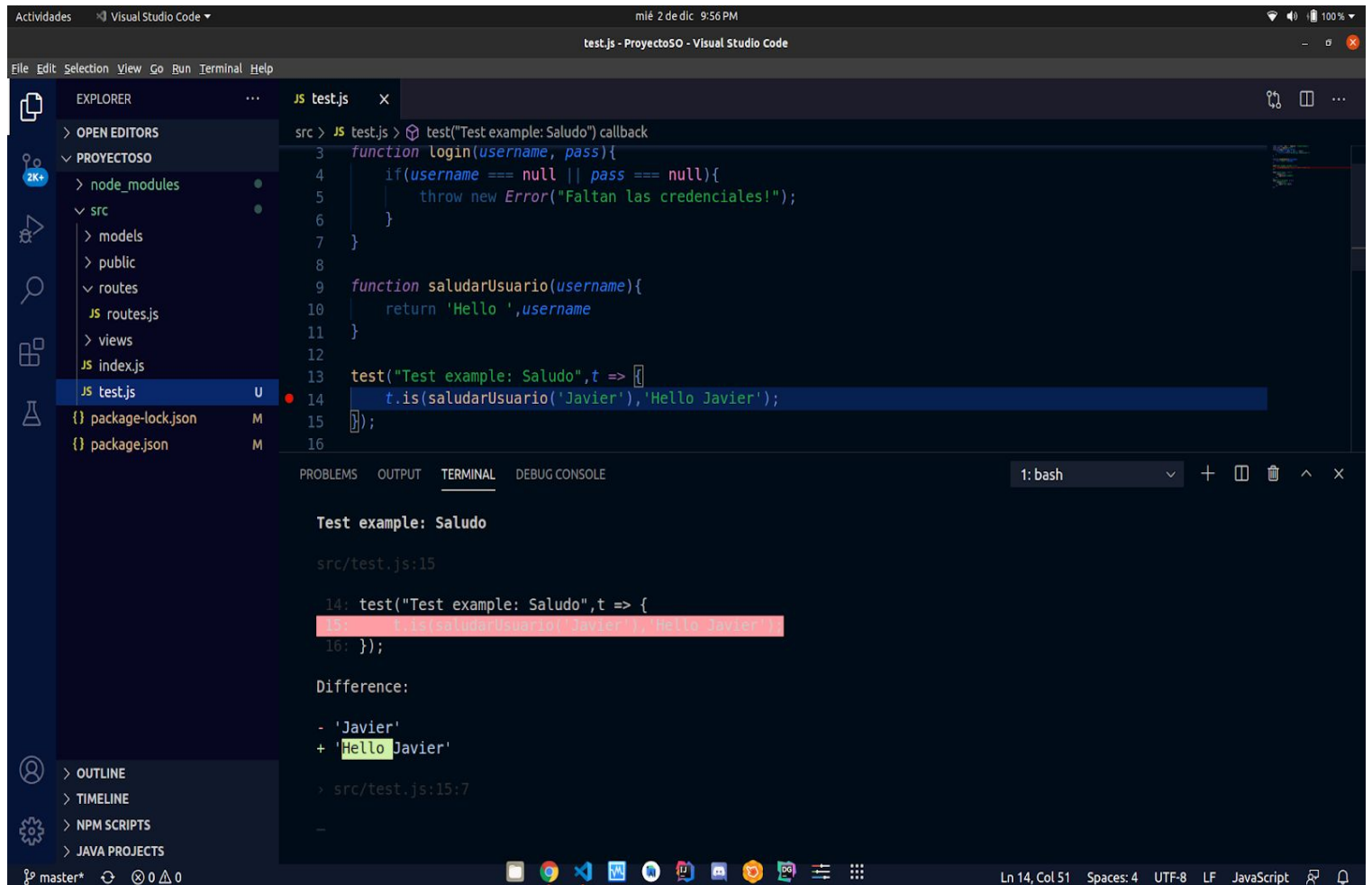
Los tests se definen como una llamada a la función test con la descripción del test y a continuación un callback (asíncrono si se desea) con el objeto que controla los tests (llamado “t”, normalmente).

El objeto “t” soporta múltiples operaciones, siendo “is” la más básica. En este caso is pide que sus dos argumentos sean iguales entre sí.

```
JS test.js
src > JS test.js > test("Test example: Saludo") callback
1  const { router, use } = require('./routes/routes')
2  const test = require('ava')
3
4  function saludarUsuario(username){
5    return 'Hello ',username
6  }
7
8  function login(username, pass){
9    if(username === null || pass === null){
10     throw new Error("Faltan las credenciales!");
11    }
12  }
13
14  test("Test example: Saludo",t => {
15    t.is(saludarUsuario('Hello Javier'),'Hello Javier');
16  });
17
18  test("Error Login", t => {
19    t.throws(()=>{
20      login(null,"123456");
21    });
22  });
23  test("Error password", t => {
24    t.throws(()=>{
25      login("test",null);
26    });
27  });
```

Para poder hacer el test a nuestras funciones debemos ejecutar el siguiente comando: **npm run test**, o si definen su propio comando en los Scripts del package.json entonces se ejecuta ese.

Si alguna de las funciones cuenta con un error Ava se encargará de mostrar la línea exacta y cuál es el error.



```
src > JS test.js > test("Test example: Saludo") callback
3 function login(username, pass){
4   if(username === null || pass === null){
5     throw new Error("Faltan las credenciales!");
6   }
7 }
8
9 function saludarUsuario(username){
10   return 'Hello ',username
11 }
12
13 test("Test example: Saludo",t => {
14   t.is(saludarUsuario('Javier'),'Hello Javier');
15 });
16
```

Test example: Saludo

src/test.js:15

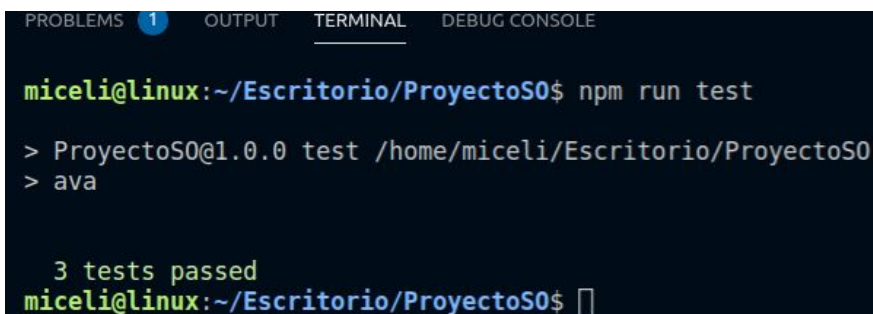
```
14: test("Test example: Saludo",t => {
15:   t.is(saludarUsuario('Javier'),'Hello Javier');
16: });
```

Difference:

```
- 'Javier'
+ 'Hello Javier'
```

> src/test.js:15:7

Por otro lado, si las pruebas que se le aplican a las funciones son examinadas y aprobadas correctamente, ava se encarga de mostrar por la consola los resultados.



```
miceli@linux:~/Escritorio/ProyectoS0$ npm run test

> ProyectoS0@1.0.0 test /home/miceli/Escritorio/ProyectoS0
> ava

  3 tests passed
miceli@linux:~/Escritorio/ProyectoS0$
```

The screenshot shows the Visual Studio Code editor with a file named `test.js` open. The code defines a `saludarUsuario` function and two tests using the `test` function. The first test checks if `saludarUsuario('Hello Javier')` returns `'Hello Javier'`. The second test checks if `login(null, '123456')` throws an error. The terminal at the bottom shows the command `npm run test` being executed, which runs the tests using `ava`. The output indicates that all 3 tests passed.

```
mié 2 de dic 9:57 PM
test.js - ProyectoS0 - Visual Studio Code

JS test.js x
src > JS test.js > test("Test example: Saludo") callback
6   }
7   }
8
9   function saludarUsuario(username){
10     return 'Hello ',username
11   }
12
13   test("Test example: Saludo",t => {
14     t.is(saludarUsuario('Hello Javier'),'Hello Javier');
15   });
16
17   test("Error Login", t => {
18     t.throws(()=>{
19       login(null,"123456");
20     });
21   });
22 }
```

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: bash

```
miceli@linux:~/Escritorio/ProyectoS0$ npm run test
> ProyectoS0@1.0.0 test /home/miceli/Escritorio/ProyectoS0
> ava

 3 tests passed
miceli@linux:~/Escritorio/ProyectoS0$
```

Ln 14, Col 32 Spaces: 4 UTF-8 LF JavaScript

Si todo salió bien, debería ver un mensaje que indica que pasó una prueba.

Bibliografía:

<https://github.com/avajs/ava>
https://github.com/avajs/ava-docs/blob/master/es_ES/readme.md#documentaci%C3%B3n

Link herramienta:

<https://github.com/avajs/ava#usage>