

DEPARTAMENTO:	Ciencias de la computación	CARRERA:	Software		
ASIGNATURA:	Pruebas de software	NIVEL:	6to	FECHA:	25/10/2025
DOCENTE:	Luis Alberto Castillo Salinas	PRÁCTICA N°:	1	CALIFICACIÓN:	

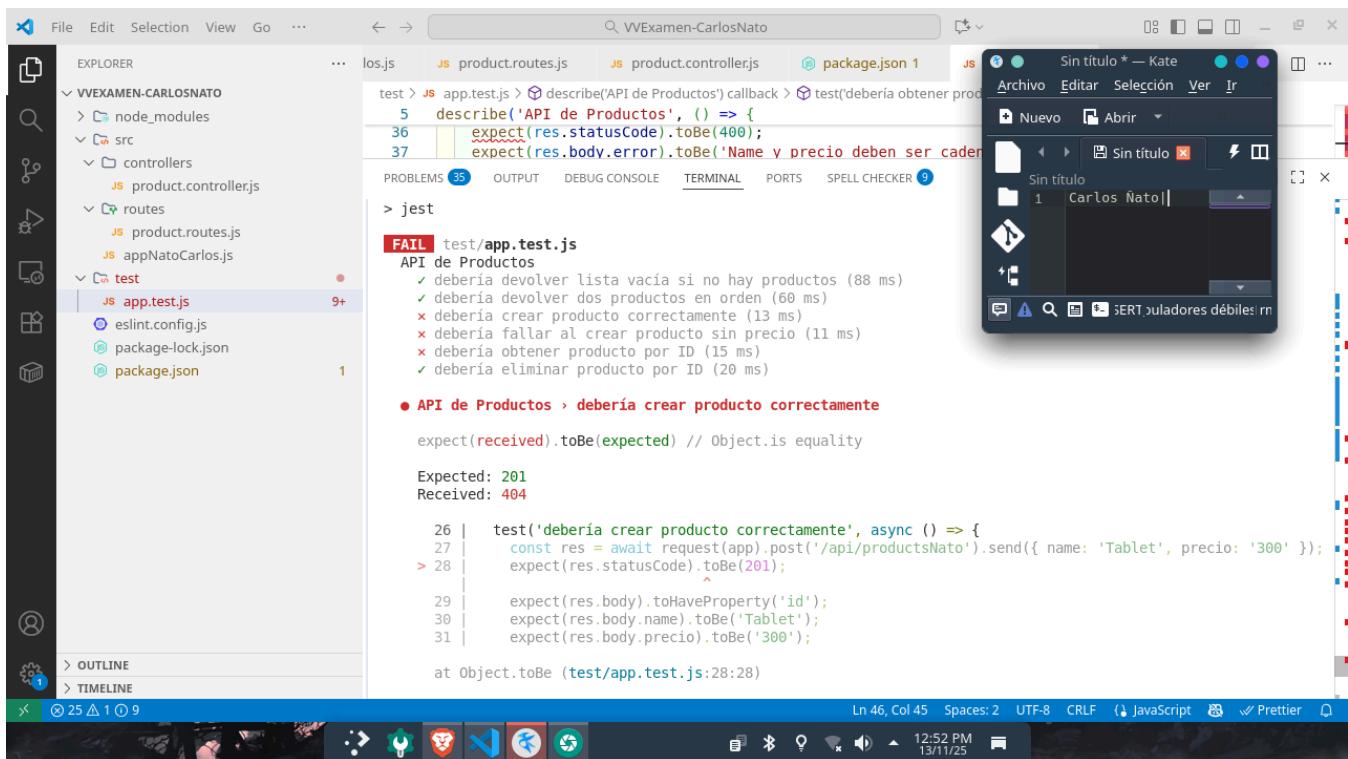
Tema: Verificación y Validación en API REST – Node.js

1. Endpoints de productos (2.0 pts)

```

VVExamen-CarlosNato
JS appNatoCarlos.js JS product.routes.js JS product.controller.js JS app.test.js 9+ eslint.config.js
src > routes > JS product.routes.js > ...
3   getAllProducts,
4   UpdateProduct,
5   createProduct,
6   deleteProductById,
7 } = require('../controllers/product.controller');
8
9 const router = express.Router();
10
11 // GET para obtener todos los productos
12 router.get('/CarlosNato', getAllProducts);
13
14 // PUT para obtener un producto por ID
15 router.get('/CarlosNato/:id', UpdateProduct);
16
17 // POST para crear un nuevo producto
18 router.post('/CarlosNato', createProduct);
19
20 // DELETE para eliminar un producto por ID
21 router.delete('/CarlosNato/:id', deleteProductById);
22
23 module.exports = router;

```



```

FAIL - test/app.test.js
API de Productos
  ✓ debería devolver lista vacía si no hay productos (88 ms)
  ✓ debería devolver los productos en orden (60 ms)
  ✘ debería crear producto correctamente (13 ms)
    × debería fallar al crear producto sin precio (11 ms)
  ✘ debería obtener producto por ID (15 ms)
  ✓ debería eliminar producto por ID (20 ms)

● API de Productos > debería crear producto correctamente

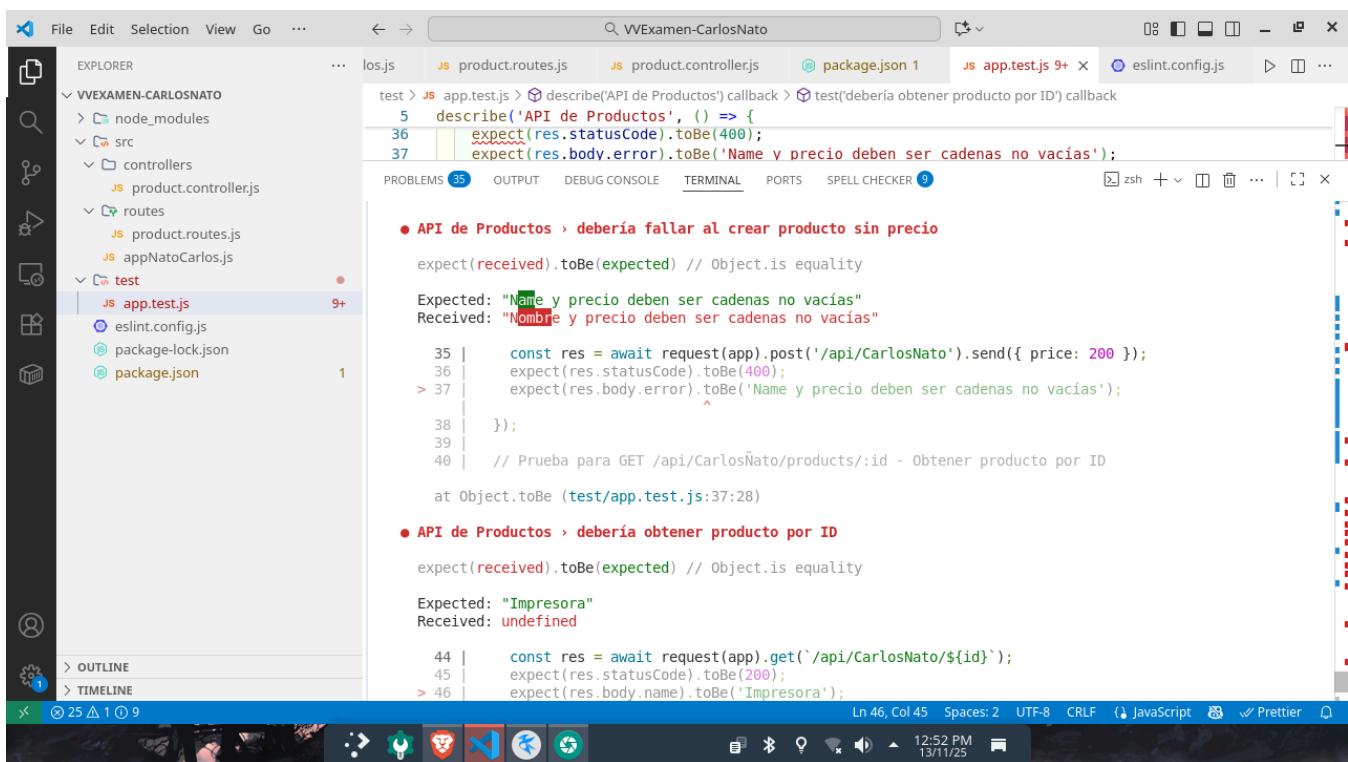
expect(received).toBe(expected) // Object.is equality

Expected: 201
Received: 404

26 |   test('debería crear producto correctamente', async () => {
27 |     const res = await request(app).post('/api/productsNato').send({ name: 'Tablet', precio: '300' });
> 28 |     expect(res.statusCode).toBe(201);
      |
29 |     expect(res.body).toHaveProperty('id');
30 |     expect(res.body.name).toBe('Tablet');
31 |     expect(res.body.precio).toBe('300');

at Object.toBe (test/app.test.js:28:28)
  
```

Ln 46, Col 45 Spaces: 2 UTF-8 CRLF (JavaScript) Prettier



```

FAIL - test/app.test.js
API de Productos > debería obtener producto por ID
  ✓ debería devolver lista vacía si no hay productos (88 ms)
  ✓ debería devolver los productos en orden (60 ms)
  ✘ debería fallar al crear producto sin precio (13 ms)
    × debería obtener producto por ID (15 ms)
  ✘ debería eliminar producto por ID (20 ms)

● API de Productos > debería obtener producto por ID

expect(received).toBe(expected) // Object.is equality

Expected: "Name y precio deben ser cadenas no vacías"
Received: "Nombre y precio deben ser cadenas no vacías"

35 |   const res = await request(app).post('/api/CarlosNato').send({ price: 200 });
36 |   expect(res.statusCode).toBe(400);
> 37 |   expect(res.body.error).toBe('Name y precio deben ser cadenas no vacías');
      |
38 | );
39 | // Prueba para GET /api/CarlosNato/products/:id - Obtener producto por ID
40 | 

at Object.toBe (test/app.test.js:37:28)

● API de Productos > debería crear producto correctamente

expect(received).toBe(expected) // Object.is equality

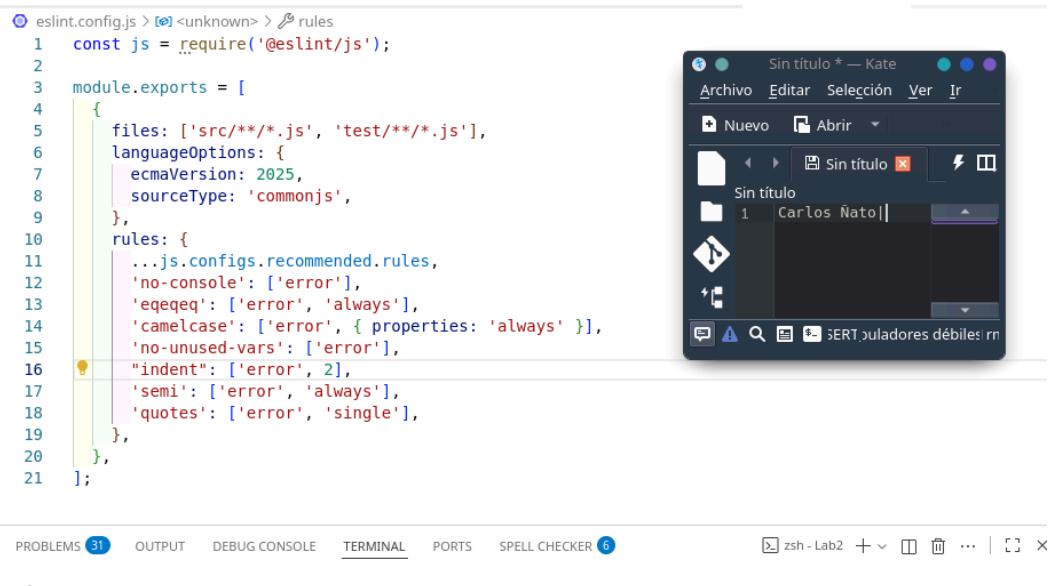
Expected: "Impresora"
Received: undefined

44 |   const res = await request(app).get(`/api/CarlosNato/${id}`);
45 |   expect(res.statusCode).toBe(200);
> 46 |   expect(res.body.name).toBe('Impresora');

  
```

Ln 46, Col 45 Spaces: 2 UTF-8 CRLF (JavaScript) Prettier

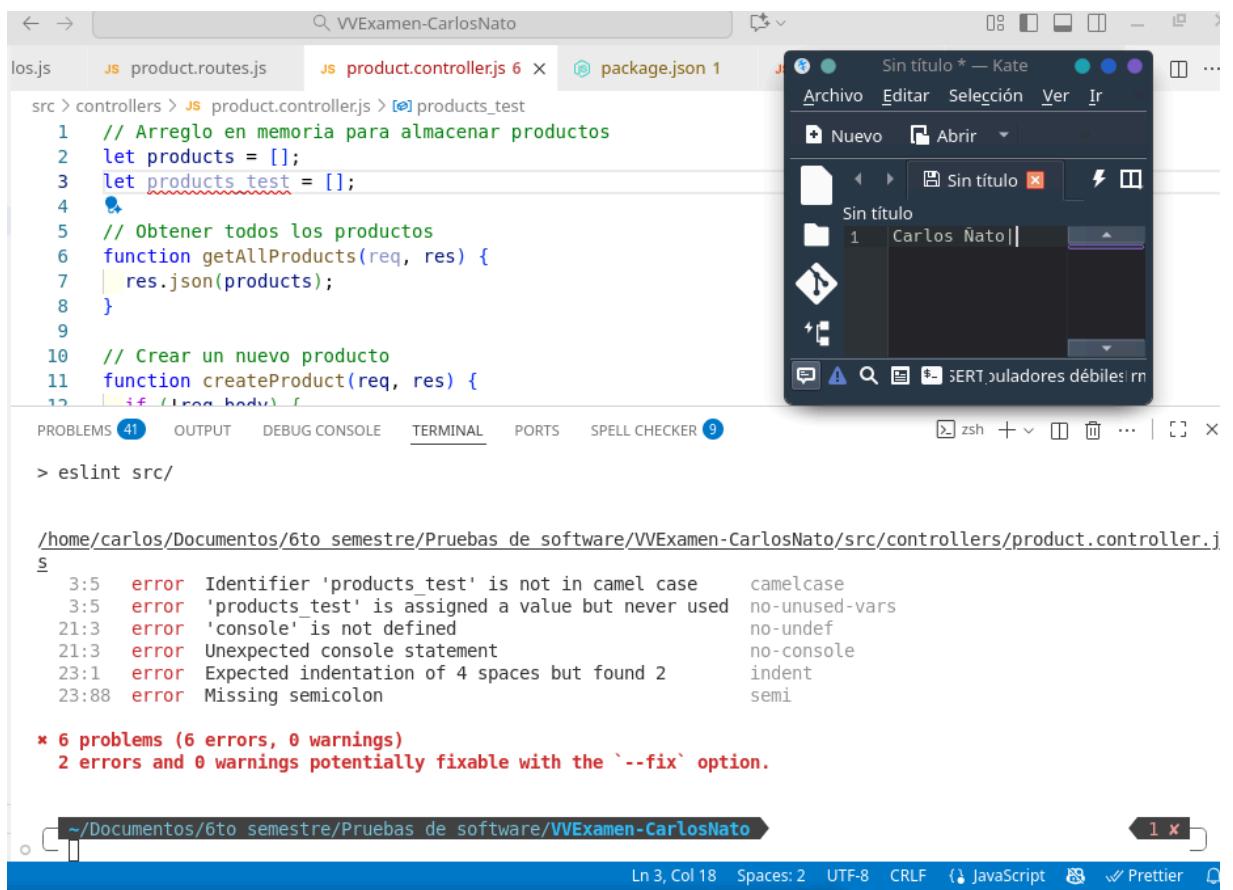
2. Reglas de verificación con ESLint (1.5 pts)



The screenshot shows the Visual Studio Code interface. On the left, the code editor displays an ESLint configuration file named `eslint.config.js`. The file contains configuration for file paths, language options, and various ESLint rules. On the right, a file browser window titled "Sin título * — Kate" is open, showing a folder structure with files like `Sin título` and `Carlos Nato`. Below the editor, the status bar shows "PROBLEMS 31", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", "PORTS", "SPELL CHECKER 6", and "Time: 0.785 s, estimated 1 s".

```
eslint.config.js > [?] <unknown> > ⚙ rules
1  const js = require('@eslint/js');
2
3  module.exports = [
4    {
5      files: ['src/**/*.js', 'test/**/*.js'],
6      languageOptions: {
7        ecmaVersion: 2025,
8        sourceType: 'commonjs',
9      },
10     rules: {
11       ...js.configs.recommended.rules,
12       'no-console': ['error'],
13       'eqeqeq': ['error', 'always'],
14       'camelcase': ['error', { properties: 'always' }],
15       'no-unused-vars': ['error'],
16       "indent": ['error', 2],
17       'semi': ['error', 'always'],
18       'quotes': ['error', 'single'],
19     },
20   },
21 ];
```

Evidencia PDF:



The screenshot shows a code editor interface with several tabs open: los.js, product.routes.js, product.controller.js (which has 6 errors), package.json, and another partially visible file. The product.controller.js tab contains the following code:

```
src > controllers > js product.controller.js > products_test
1 // Arreglo en memoria para almacenar productos
2 let products = [];
3 let products test = [];
4 
5 // Obtener todos los productos
6 function getAllProducts(req, res) {
7   res.json(products);
8 }

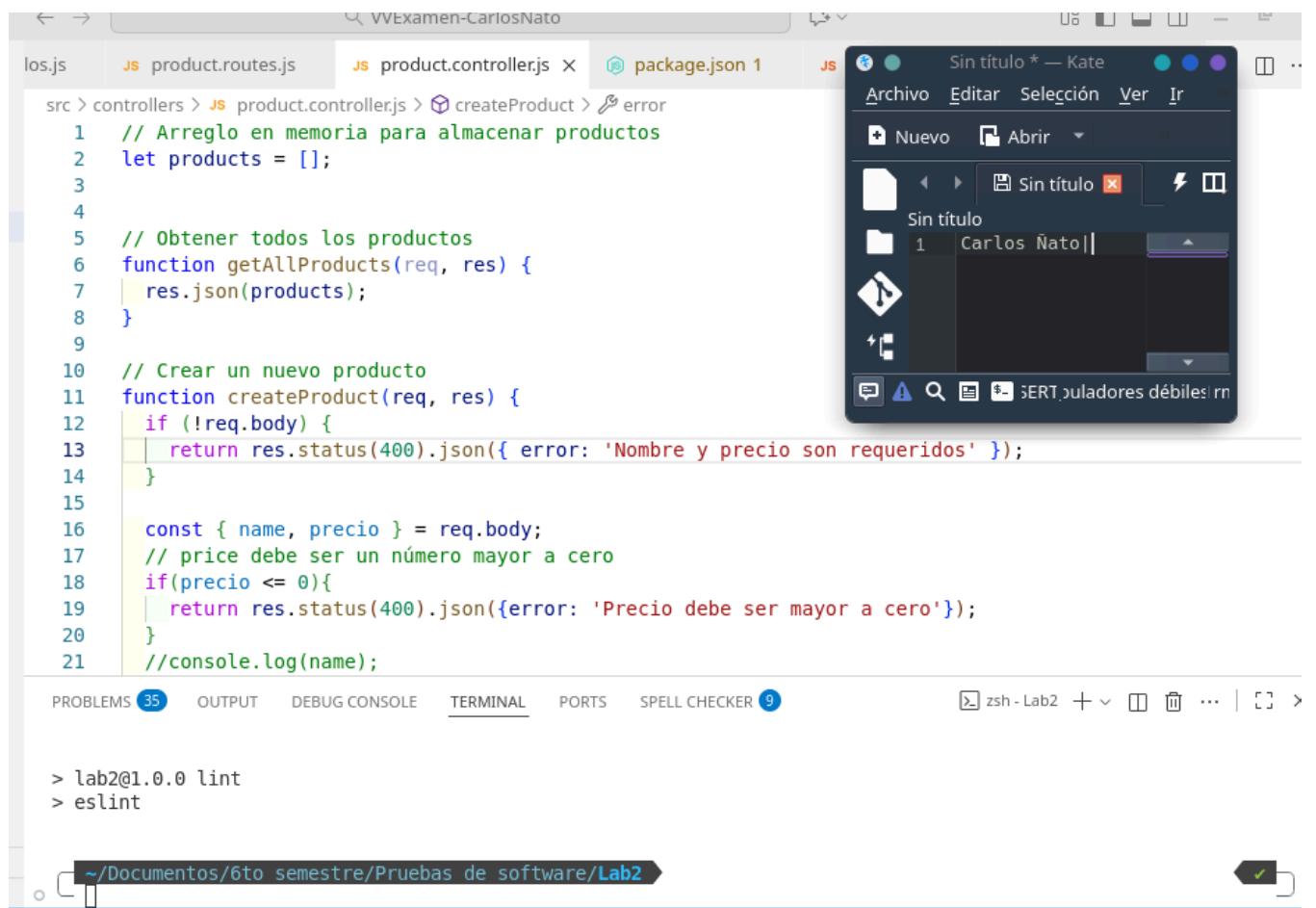
10 // Crear un nuevo producto
11 function createProduct(req, res) {
12   if (req.body) {
13     products.push(req.body);
14     res.json(products);
15   }
16 }
```

The terminal below shows the output of running ESLint on the source directory:

```
> eslint src/
/home/carlos/Documentos/6to semestre/Pruebas de software/VVExamen-CarlosNato/src/controllers/product.controller.js
  3:5  error  Identifier 'products_test' is not in camel case      camelcase
  3:5  error  'products_test' is assigned a value but never used  no-unused-vars
21:3  error  'console' is not defined                            no-undef
21:3  error  Unexpected console statement                      no-console
23:1  error  Expected indentation of 4 spaces but found 2       indent
23:88 error  Missing semicolon                                semi

✖ 6 problems (6 errors, 0 warnings)
  2 errors and 0 warnings potentially fixable with the `--fix` option.
```

3. Validaciones de producto (1.5 pts)



The screenshot shows a code editor interface with several tabs open:

- los.js
- product.routes.js
- product.controller.js (active tab)
- package.json 1

The product.controller.js file contains the following code:

```
src > controllers > product.controller.js > createProduct > error
1 // Arreglo en memoria para almacenar productos
2 let products = [];
3
4
5 // Obtener todos los productos
6 function getAllProducts(req, res) {
7     res.json(products);
8 }
9
10 // Crear un nuevo producto
11 function createProduct(req, res) {
12     if (!req.body) {
13         return res.status(400).json({ error: 'Nombre y precio son requeridos' });
14     }
15
16     const { name, precio } = req.body;
17     // price debe ser un número mayor a cero
18     if(precio <= 0){
19         return res.status(400).json({error: 'Precio debe ser mayor a cero'});
20     }
21     //console.log(name);
```

The package.json file shows:

```
Sin título * — Kate
Archivo Editar Selección Ver Ir
Nuevo Abrir
Sin título
1 Carlos Ñato||
```

The terminal window shows the following command-line output:

```
> lab2@1.0.0 lint
> eslint
```

The status bar at the bottom indicates the current directory is `~/Documentos/6to semestre/Pruebas de software/Lab2`.