

Aprendiz:

Henry Andrés Morales Garzón

Valentina Vargas Sanchez

Instructor

Andrés Rubiano Cucarian

CENTRO DE SERVICIOS FINANCIEROS

SENA- REGIONAL DISTRITO CAPITAL

ANALISIS Y DESARROLLO DEL SOFTWARE

FICHA: 2627062

2024

Tipos de pruebas de software y sus herramientas disponibles

Tipo de prueba	Herramientas
Pruebas de unidad: Se centran en verificar el funcionamiento individual de cada componente o unidad del software.	JUnit (para Java), NUnit (para .NET), Pytest (para Python), Jasmine (para JavaScript).
Pruebas de integración: Comprueban la interoperabilidad entre diferentes módulos o unidades del software.	Postman (para API), SoapUI (para servicios web), TestNG (para Java), Mocha (para JavaScript).
Pruebas de sistema: Validan el comportamiento del sistema completo en relación con los requisitos especificados.	Selenium WebDriver (para pruebas de GUI), Appium (para pruebas de aplicaciones móviles), Robot Framework (para pruebas de aceptación de sistemas).
Pruebas de aceptación: Se realizan para asegurar que el software cumple con los criterios de aceptación del cliente o usuario final.	Cucumber (para pruebas de comportamiento), SpecFlow (para .NET), Behave (para Python), Jasmine (para JavaScript).
Pruebas de regresión: Se ejecutan para verificar que las modificaciones realizadas en el software no hayan introducido nuevos errores o afectado negativamente a funcionalidades existentes.	Selenium WebDriver, Ranorex, TestComplete, Telerik Test Studio.
Pruebas de rendimiento: Evalúan el desempeño del software en términos de velocidad, eficiencia y escalabilidad bajo diferentes condiciones de carga.	JMeter, LoadRunner, Gatling, Apache Benchmark.
Pruebas de seguridad: Buscan identificar vulnerabilidades y garantizar que el software sea resistente a ataques maliciosos.	OWASP ZAP (para pruebas de seguridad automatizadas), Burp Suite, Nessus, Acunetix.

Pruebas usabilidad: Se enfocan en la experiencia del usuario, evaluando la facilidad de uso, la accesibilidad y la satisfacción general al interactuar con el software.	UserTesting (para pruebas de usuario reales), UsabilityHub, Optimal Workshop, Crazy Egg.
--	--

Pruebas que se adaptan al proyecto

- **Las pruebas de unidad.**

Debido a que nuestro proyecto se divide en módulos estas pruebas nos ayudan a:

- **Identificación temprana de errores:** Ayuda a detectar errores en componentes individuales en software antes de que se integren en su sistema completo.
- **Mejora de la calidad del código:** Al hacer las pruebas se debe considerar como se comporta el código en diferentes situaciones, lo que hace que el código sea más limpio y modular.

- **Las pruebas de integración.**

Nuestro proyecto tiene pequeños sistemas que comparten varios módulos, así que estas pruebas nos aseguran:

- **Validación del comportamiento del sistema completo:** Las pruebas de integración verifican que los diferentes componentes del sistema funcionen correctamente juntos como un todo, garantizando que el software cumpla con los requisitos funcionales y de rendimiento.
- **Detección temprana de problemas de interoperabilidad:** Identifican problemas de comunicación o interoperabilidad entre los distintos módulos o sistemas que componen la aplicación, permitiendo corregirlos antes de que afecten al funcionamiento del sistema en su totalidad.
- **Facilitan la detección de fallos ocultos:** Las pruebas de integración revelan fallos que pueden pasar desapercibidos durante las pruebas unitarias, ya que simulan el entorno real en el que operará el software.
- **Mejora la mantenibilidad del sistema:** Al garantizar que los componentes estén bien integrados y funcionen correctamente en conjunto, las pruebas de integración facilitan el mantenimiento del sistema a lo largo del tiempo.

- **Las pruebas de sistema.**

Estas pruebas son necesarias en nuestro proyecto ya que va muy ligado con el cumplimiento de requisitos con el usuario y del sistema en su conjunto, las razones son las siguientes:

- **Validación del cumplimiento de requisitos:** Las pruebas de sistema garantizan que el software cumpla con todos los requisitos del usuario y

del sistema, lo que incluye funcionalidades específicas, rendimiento, seguridad y usabilidad.

- **Identificación de problemas de extremo a extremo:** Estas pruebas evalúan el comportamiento del sistema completo en un entorno simulado o realista, lo que permite identificar problemas que pueden surgir cuando los diferentes componentes interactúan entre sí.
- **Detección temprana de problemas de integración:** Al evaluar el sistema completo, las pruebas de sistema ayudan a detectar problemas de integración entre los diferentes módulos o sistemas que componen la aplicación, permitiendo corregirlos antes de la implementación en producción.
- **Validación de la usabilidad y la experiencia del usuario:** Estas pruebas permiten evaluar la facilidad de uso y la experiencia del usuario final, asegurando que el software sea intuitivo y satisfactorio para los usuarios.

- **Pruebas de rendimiento.**

Esta prueba es necesaria debido que nuestra web ofrece servicios en línea y requiere tiempos de carga rápidos, por eso estas pruebas nos garantiza lo siguiente:

- **Garantía de rendimiento óptimo:** Las pruebas de rendimiento ayudan a identificar y corregir cuellos de botella, ineficiencias y problemas de escalabilidad que pueden afectar el rendimiento del software.
- **Mejora de la experiencia del usuario:** Un rendimiento deficiente puede resultar en tiempos de carga lentos, retrasos en la respuesta del sistema y una experiencia de usuario insatisfactoria. Las pruebas de rendimiento garantizan que el software responda rápidamente a las acciones del usuario, lo que mejora su experiencia y satisfacción.
- **Optimización de recursos:** Identificar y resolver problemas de rendimiento ayuda a optimizar el uso de recursos, como CPU, memoria y ancho de banda.

- **Pruebas de seguridad.**

Nuestra web es una plataforma de comercio electrónico el cual maneja datos sensibles es por eso por lo que se necesita estas pruebas que nos ofrece lo siguiente:

- **Protección de datos sensibles:** Las pruebas de seguridad ayudan a identificar y mitigar vulnerabilidades que podrían exponer datos sensibles, como información personal, financiera o de negocios, a accesos no autorizados o filtraciones.
- **Prevención de ataques cibernéticos:** Al identificar y corregir vulnerabilidades de seguridad antes de la implementación en producción, las pruebas de seguridad reducen la superficie de ataque y disminuyen el riesgo de ser comprometido por ataques como inyecciones SQL, cross-site scripting (XSS) o ataques de denegación de servicio (DDoS).

- **Cumplimiento de regulaciones y estándares:** Las pruebas de seguridad ayudan a garantizar que el software cumpla con las regulaciones y estándares de seguridad aplicables, como el Reglamento General de Protección de Datos (GDPR), la Ley de Privacidad y Seguridad de la Información de la Industria de Tarjetas de Pago (PCI DSS) o las normas ISO/IEC 27001.
- **Pruebas de usabilidad.**
 Al tener una interfaz que muestra variedad de datos necesitamos saber si se muestra de manera intuitiva, fácil de usar Y satisfagan las necesidades de los usuarios, es por eso por lo que esta prueba nos ayuda a:
 - **Mejora de la experiencia del usuario:** Las pruebas de usabilidad ayudan a identificar y corregir problemas de diseño, navegación y flujo de trabajo que pueden dificultar la experiencia del usuario.
 - **Aumento de la eficiencia y productividad:** Un software con una interfaz intuitiva y fácil de usar permite a los usuarios completar tareas de manera más rápida y eficiente.
 - **Reducción de errores y tiempo de aprendizaje:** Una interfaz de usuario clara y bien diseñada reduce la probabilidad de cometer errores y el tiempo necesario para aprender a usar el software.
 - **Aumento de la retención de usuarios:** Los usuarios tienden a abandonar un software si encuentran dificultades para usarlo o si no satisface sus necesidades.

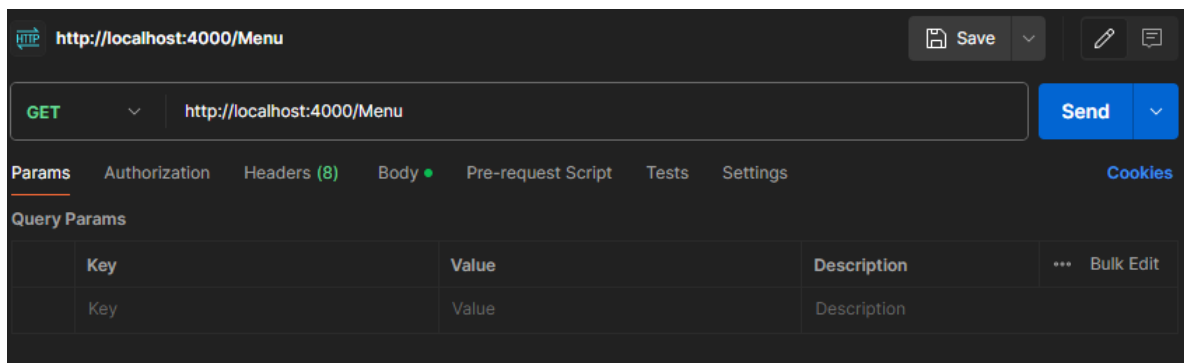
Prueba de integración en nuestro proyecto con Postman.

Método Get (mostrar registros)

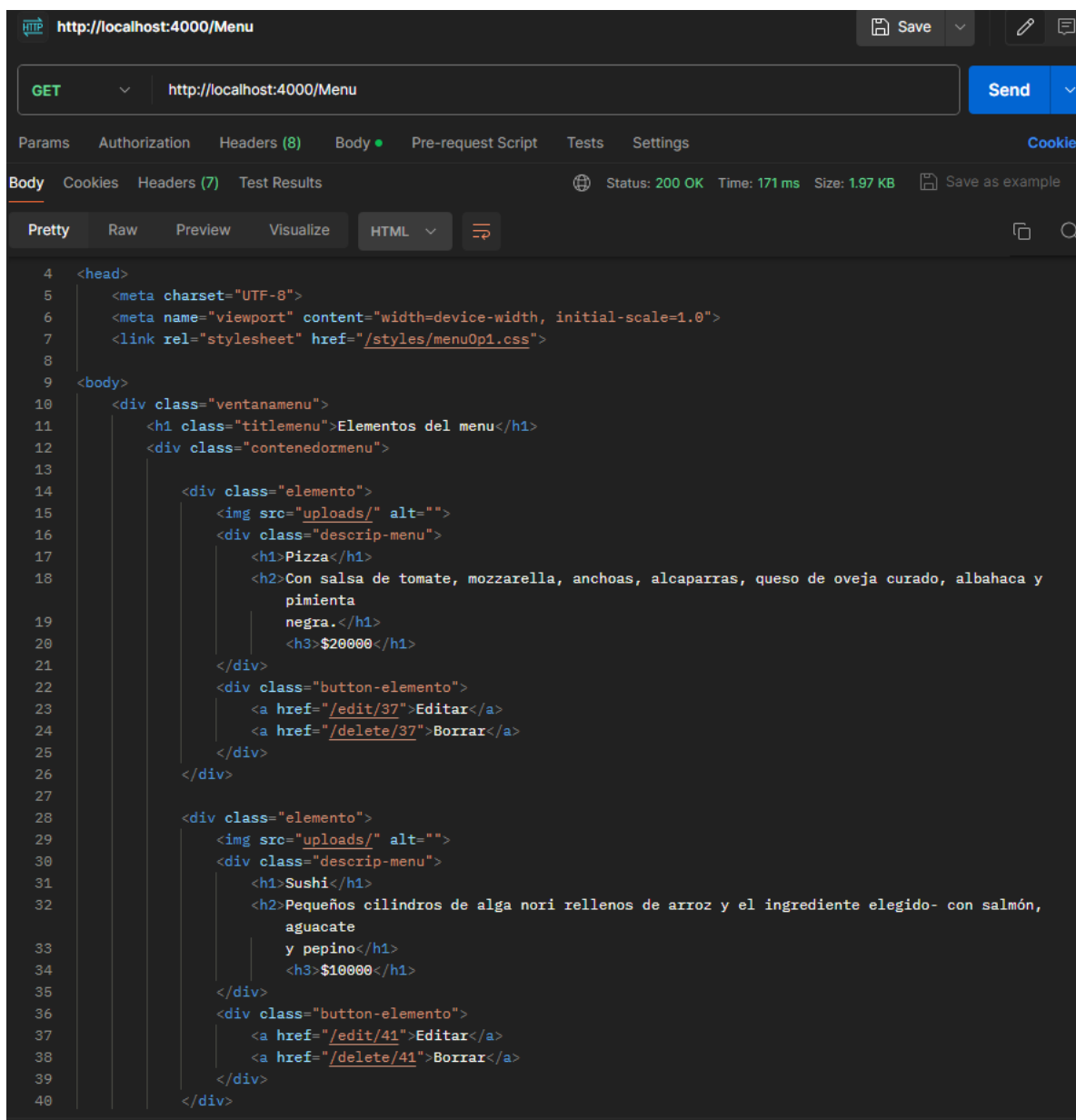
- El método get que usamos en nuestro proyecto pide los registros a nuestra base de datos Sql, asignándole la ruta “/Menú” para esta función.

```
router.get('/Menu',(req,res)=>{
  conexion.query('SELECT * FROM menu',(error,results)=>{
    if(error){
      throw error
    }else{
      res.render('menu',{results:results})
    }
  })
})
```

- Usamos la ruta completa en la herramienta Postman para hacer la prueba.



- El resultado que nos arroja Postman es en HTML donde se evidencia que efectivamente muestra los registros existentes.



- La base de datos con los registros existentes

SQL File 4* menu menu menu menu x

Limit to 1000 rows

1 • `SELECT * FROM bd_foodplus.menu;`

Result Grid

	idmenu	nombre_menu	descripcion_menu	precio_menu	img_menu	fechacreacion
▶	37	Pizza	Con salsa de tomate, mozzarella, anchoas, a...	20000	NULL	1714851636489
	41	Sushi	Pequeños cilindros de alga nori rellenos de ar...	10000	NULL	1714852152630
*	NULL	NULL	NULL	NULL	NULL	NULL

menu 1 x

Apply Revert

Método Post (agregar registros)

- El método post en nuestro proyecto agrega los datos seleccionados en un único registro dentro de nuestra base de datos SQL.

```
router.post("/savemenu",upload.single("image"),(req,res)=>{
  const nombre_menu = req.body.nombre_menu
  const descripcion_menu = req.body.descripcion_menu
  const precio_menu = req.body.precio_menu

  conexion.query('INSERT INTO menu SET ?', {nombre_menu:nombre_menu, descripcion_menu:descripcion_menu, precio_menu:precio_menu, fechacreacion:Date.now()}, (error,result)=>{
    if(error){
      console.log(error)
    }else{
      res.redirect('/Restaurante')
    }
  })
})
```

- En Postman aparte de agregar la ruta para realizar la prueba, se ponen los datos que se desean agregar al nuevo registro.

http://localhost:4000/savemenu

POST http://localhost:4000/savemenu

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL JSON Beautify

```
1 {
2   "nombre_menu": "Hamburguesa",
3   "descripcion_menu": "Con 200 g de carne Angus, pan artesanal, salsa de queso azul, cebolla morada, tocineta
4   "precio_menu": 30000
5 }
```

- ```
http://localhost:4000/Menu

GET http://localhost:4000/Menu Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies Headers (7) Test Results Status: 200 OK Time: 9 ms Size: 2.59 KB Save as example

Pretty Raw Preview Visualize HTML




15
16 <div class="descrip-menu">
17 <h1>Pizza</h1>
18 <h2>Con salsa de tomate, mozzarella, anchoas, alcaparras, queso de oveja curado, albahaca y
19 pimienta
20 negra.</h1>
21 <h3>$20000</h1>
22 </div>
23 <div class="button-elemento">
24 Editar
25 Borrar
26 </div>
27
28 <div class="elemento">
29
30 <div class="descrip-menu">
31 <h1>Sushi</h1>
32 <h2>Pequeños cilindros de alga nori rellenos de arroz y el ingrediente elegido- con salmón,
33 aguacate
34 y pepino.</h1>
35 <h3>$10000</h1>
36 </div>
37 <div class="button-elemento">
38 Editar
39 Borrar
40 </div>
41 </div>
42
43 <div class="elemento">
44
45 <div class="descrip-menu">
46 <h1>Hamburguesa</h1>
47 <h2>Con 200 g de carne Angus, pan artesanal, salsa de queso azul, cebolla morada, tocineta
48 caramelizada y lechuga fresca.</h1>
49 <h3>$30000</h1>
50 </div>
51 <div class="button-elemento">
52 Editar
```

- | Result Grid  |             |                                                   |             |                |               |                                      |
|--------------|-------------|---------------------------------------------------|-------------|----------------|---------------|--------------------------------------|
| Filter Rows: |             | Edit:                                             |             | Export/Import: |               | Wrap Cell Content: <a href="#">↕</a> |
| idmenu       | nombre_menu | descripcion_menu                                  | precio_menu | img_menu       | fechacreacion |                                      |
| 37           | Pizza       | Con salsa de tomate, mozzarella, anchoas, a...    | 20000       | NULL           | 1714851636489 |                                      |
| 41           | Sushi       | Pequeños cilindros de alga nori rellenos de ar... | 10000       | NULL           | 1714852152630 |                                      |
| 42           | Hamburguesa | Con 200 g de carne Angus, pan artesanal, s...     | 30000       | NULL           | 1715105569690 |                                      |
| *            | NULL        | NULL                                              | NULL        | NULL           | NULL          |                                      |



- La vista de nuestra página web con el registro nuevo.

## Elementos del menu

|                                                                                   |                                                                                                                                                                         |                                     |
|-----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
|  | <h3>Pizza</h3> <p>Con salsa de tomate, mozzarella, anchoas, alcaparras, queso de oveja curado, albahaca y pimienta negra.</p> <p><b>\$20000</b></p>                     | <div>Editar</div> <div>Borrar</div> |
|  | <h3>Sushi</h3> <p>Pequeños cilindros de alga nori rellenos de arroz y el ingrediente elegido- con salmón, aguacate y pepino</p> <p><b>\$10000</b></p>                   | <div>Editar</div> <div>Borrar</div> |
|  | <h3>Hamburguesa</h3> <p>Con 200 g de carne Angus, pan artesanal, salsa de queso azul, cebolla morada, tocineta caramelizada y lechuga crespas</p> <p><b>\$30000</b></p> | <div>Editar</div> <div>Borrar</div> |

Agregar producto

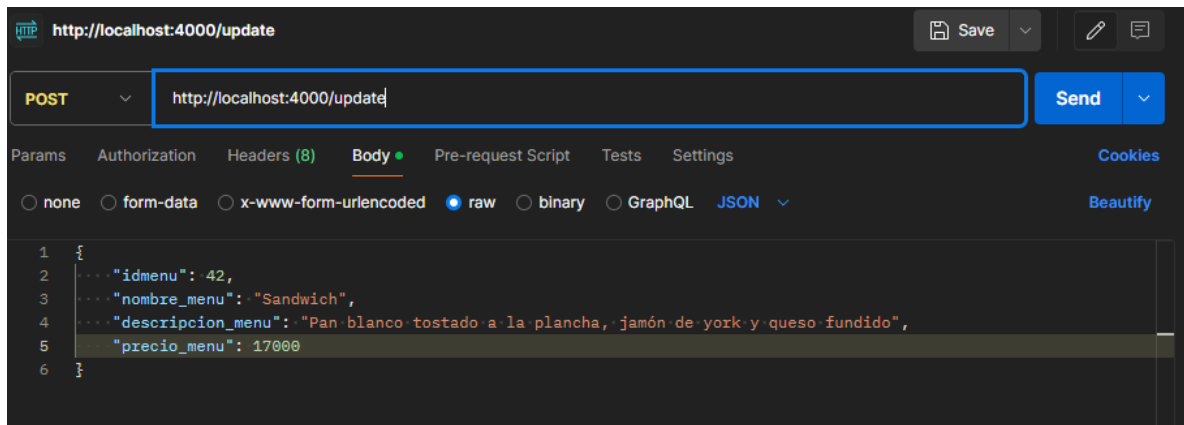
### Método Post (actualizar registros)

- Este otro método post en nuestro proyecto funciona para actualizar los datos de un registro ya existente, en este caso se usó post y no put debido a la estructura de nuestra página web, sin embargo, funciona de la misma manera.

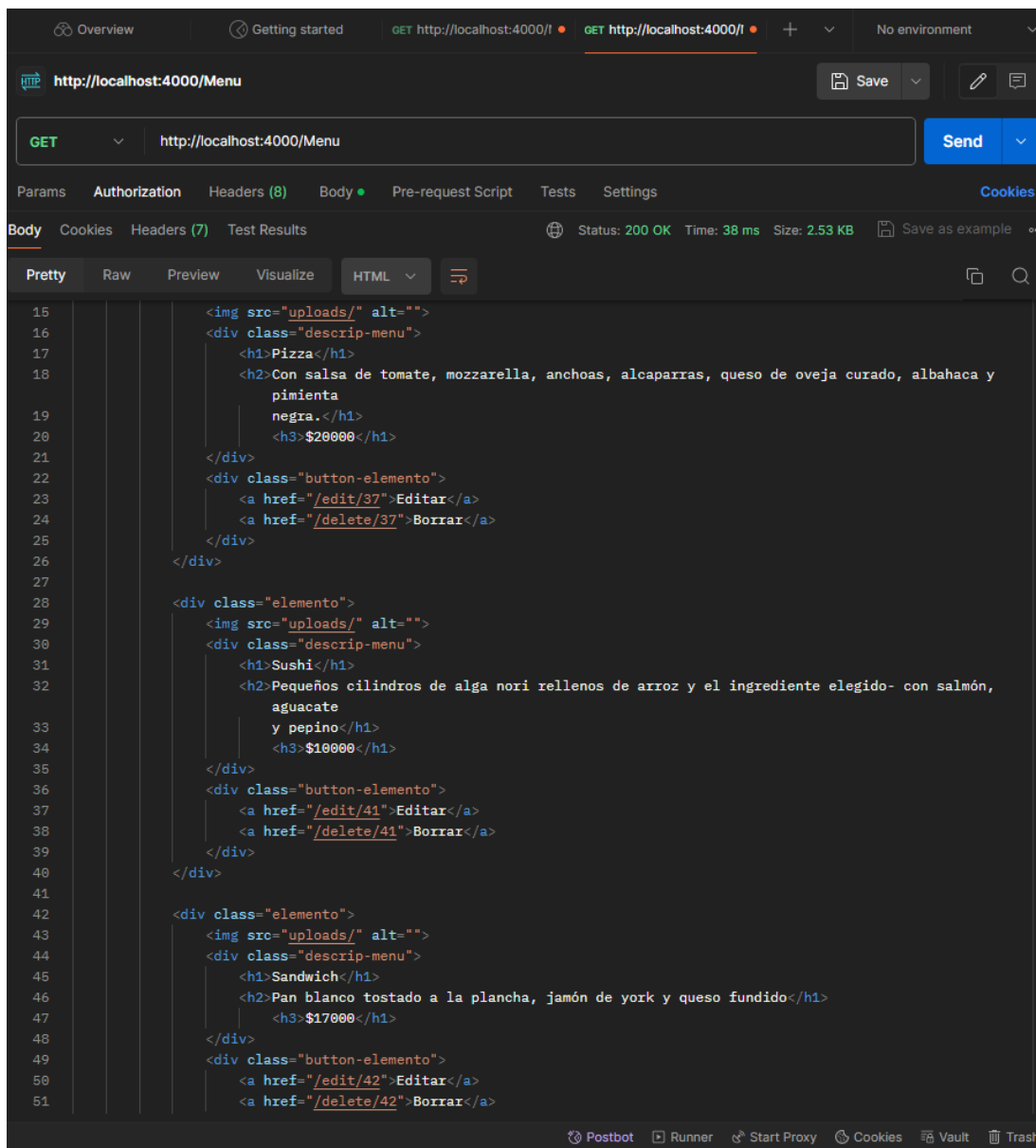
```
router.post("/update", upload.single("image"), (req, res) => {
 const id = req.body.idmenu
 const nombre_menu = req.body.nombre_menu
 const descripcion_menu = req.body.descripcion_menu
 const precio_menu = req.body.precio_menu

 conexion.query('UPDATE menu SET ? WHERE idmenu =?', [{nombre_menu:nombre_menu, descripcion_menu:descripcion_menu, precio_menu:precio_menu, fechacreacion:Date.now(), id}], (error, results) => {
 if(error){
 console.log(error)
 }else{
 res.redirect('/Restaurante')
 }
 })
})
```

- En Postman se pone la ruta respectiva y aparte se pone el id del registro que queremos cambiar junto con los datos que queremos actualizar.



- Se usa el método get para mostrar los registros y ver que si se hizo la actualización de los datos del último registro (42).




- También se evidencia el cambio en la base de datos en el registro con el id número 42.

| Result Grid |        |              |                                                   |             |          |                |
|-------------|--------|--------------|---------------------------------------------------|-------------|----------|----------------|
|             |        | Filter Rows: |                                                   |             | Edit:    | Export/Import: |
|             | idmenu | nombre_menu  | descripcion_menu                                  | precio_menu | img_menu | fecha creacion |
|             | 37     | Pizza        | Con salsa de tomate, mozzarella, anchoas, a...    | 20000       | NULL     | 1714851636489  |
|             | 41     | Sushi        | Pequeños cilindros de alga nori rellenos de ar... | 10000       | NULL     | 1714852152630  |
|             | 42     | Sandwich     | Pan blanco tostado a la plancha, jamón de y...    | 17000       | NULL     | 1715106086278  |
| *           | NULL   | NULL         | NULL                                              | NULL        | NULL     | NULL           |

- Vista de la página web con el cambio realizado.

### Elementos del menu




#### Pizza

Con salsa de tomate, mozzarella, anchoas, alcaparras, queso de oveja curado, albahaca y pimienta negra.

**\$20000**

Editar

Borrar




#### Sushi

Pequeños cilindros de alga nori rellenos de arroz y el ingrediente elegido- con salmón, aguacate y pepino

**\$10000**

Editar

Borrar



#### Sandwich

Pan blanco tostado a la plancha, jamón de york y queso fundido

**\$17000**

Editar

Borrar

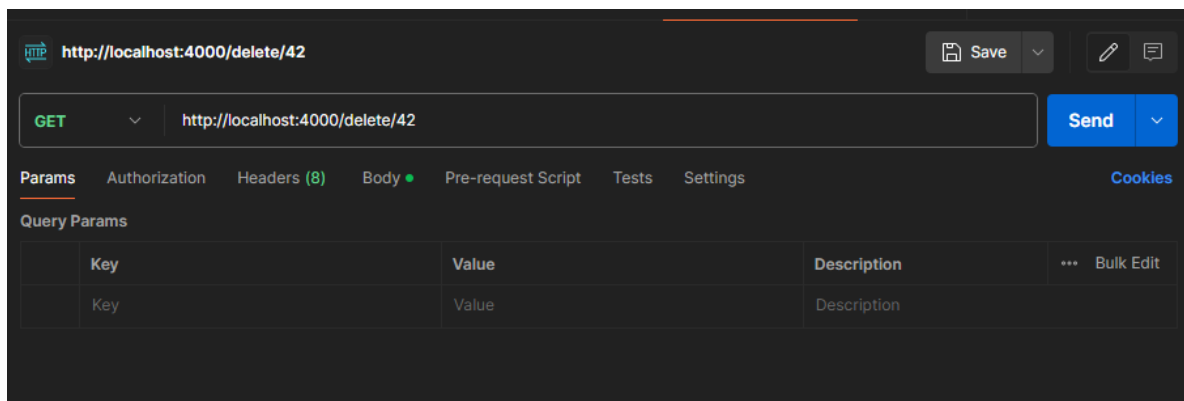
Agregar producto

### Método Get (Borrar registros)

- Este segundo método get en nuestro proyecto se usa para eliminar un registro según el id, se usó get y no delete debido a la estructura de nuestra página web, pero mantiene su función.

```
router.get('/delete/:idmenu',(req,res)=>{
 const id = req.params.idmenu
 conexion.query('DELETE FROM menu WHERE idmenu =?', [id], (error,results)=>{
 if(error){
 throw error
 }else{
 res.redirect('/Restaurante')
 }
 })
})
```

- En Postman se pone la ruta y seguido de la ruta el id del registro que se desea eliminar.



- Al usar el método get para ver los registros existentes se ve que efectivamente se borró uno de ellos.

```

10 <div class="ventanamenu">
11 <h1 class="titlemenu">Elementos del menu</h1>
12 <div class="contenedormenu">
13 <div class="elemento">
14
15 <div class="descrip-menu">
16 <h1>Pizza</h1>
17 <h2>Con salsa de tomate, mozzarella, anchoas, alcaparras, queso de oveja curado, albahaca y
18 pimienta
19 negra.</h1>
20 <h3>$20000</h1>
21 </div>
22 <div class="button-elemento">
23 Editar
24 Borrar
25 </div>
26 </div>
27
28 <div class="elemento">
29
30 <div class="descrip-menu">
31 <h1>Sushi</h1>
32 <h2>Pequeños cilindros de alga nori rellenos de arroz y el ingrediente elegido- con salmón,
33 aguacate
34 y pepino</h1>
35 <h3>$10000</h1>
36 </div>
37 <div class="button-elemento">
38 Editar
39 Borrar
40 </div>
41 </div>
42 </div>
43 <div class="cont-button"></div>
44 Agregar producto
45 </div>
46

```

- También la base de datos se puede visualizar que se borró el registro.

|   | idmenu | nombre_menu | descripcion_menu                                  | precio_menu | img_menu | fechacreacion |
|---|--------|-------------|---------------------------------------------------|-------------|----------|---------------|
| ▶ | 37     | Pizza       | Con salsa de tomate, mozzarella, anchoas, a...    | 20000       | NULL     | 1714851636489 |
|   | 41     | Sushi       | Pequeños cilindros de alga nori rellenos de ar... | 10000       | NULL     | 1714852152630 |
| ✱ | NULL   | NULL        | NULL                                              | NULL        | NULL     | NULL          |

- En la vista de nuestra página web se puede ver que se eliminó un registro.

## Elementos del menu



### Pizza

Con salsa de tomate, mozzarella, anchoas, alcaparras, queso de oveja curado, albahaca y pimienta negra.

**\$20000**

Editar

Borrar



### Sushi

Pequeños cilindros de alga nori rellenos de arroz y el ingrediente elegido- con salmón, aguacate y pepino

**\$10000**

Editar

Borrar

---

Agregar producto