

EL TINTO BENDITO

JESUS ANIBAL BAENA ARCILA
EDWIN ARMANDO ESPINOSA ZAMBRANO
JOHANN SANTIAGO NIÑO MEDINA
OSCAR DANIEL PIÑEROS HERNÁNDEZ
JUAN DIEGO VASQUEZ SOSSA
JORGE HERNÁN VICQUE SÁNCHEZ

SERGIO ARTURO MEDINA CASTILLO
FORMADOR

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
MISION TIC 2022
DESARROLLO DE SOFTWARE
2021

TABLA DE CONTENIDO

1. APLICACIÓN BACKEND.	3
1.1. Evidencia del Backend en Github.	6
1.2. Ejecución del método GET en Postman.	8
1.3. Ejecución del método GET en Postman con @Path("/{Id_Producto}").	8
1.4. Ejecución de método POST en Postman.	9
1.5. Verificación de la ejecución de método POST en Postman.	9
1.6. Verificación de la ejecución de método POST en MySQL.	10
1.7. Ejecución del método PUT en MySQL sobre el producto con id=3.	10
1.8. Ejecución del método DELETE sobre el producto con id=4.	12
2. APLICACIÓN FRONTEND.	14
2.1. Evidencia del Frontend en Github.	14
2.2. Consumo del método GET.	15
2.3. Creación de un producto nuevo – Método POST.	16
2.4. Edición de un producto Método PUT.	18
2.5. Borrado de un registro Método DELETE.	21

EL TINTO BENDITO

SPRINT 5

1. APLICACIÓN BACKEND.

Para la aplicación Backend se trabajó con la tabla PRODUCTOS, cuya estructura se muestra a continuación:

1 • DESC PRODUCTOS

Result Grid						
		Filter Rows:			Export:	Wrap Cell Content:
	Field	Type	Null	Key	Default	Extra
▶	id_prod	int	NO	PRI	NULL	auto_increment
	nombre_prod	varchar(30)	NO		NULL	
	desc_prod	varchar(100)	NO		NULL	
	id_categoria	int	YES	MUL	NULL	
	url_img_prod	varchar(30)	YES		NULL	
	precio_prod	double	YES		NULL	
	disp_prod	varchar(1)	NO		NULL	
	estado_prod	varchar(1)	NO		NULL	

Se desarrolló la clase pública ProductosService.java para manejar los servicios del backend. Su código se muestra a continuación:

```
1  package restful.service;
2
3  import java.sql.PreparedStatement;
4  import java.sql.ResultSet;
5  import java.sql.Statement;
6  import java.sql.SQLException;
7  import java.util.ArrayList;
8  import restful.model.ProductosModel;
9  import restful.model.ConexionBD;
10 import javax.swing.JOptionPane;
11
12 public class ProductosService {
13     public ArrayList<ProductosModel> getProductos() {
14         ArrayList<ProductosModel> lista = new ArrayList<>();
15         ConexionBD conn = new ConexionBD();
16
17         String sql = "SELECT * FROM productos";
18
19         try {
20             Statement stm = conn.getCon().createStatement();
21             ResultSet rs = stm.executeQuery(sql);
22             while (rs.next()) {
23                 //Crear un objeto de la clase Productos
24                 ProductosModel productos = new ProductosModel();
25                 System.out.println(rs.getInt("id_prod"));
26                 productos.setId_prod(rs.getInt("id_prod"));
27                 productos.setNombre_prod(rs.getString("nombre_prod"));
28                 productos.setDesc_prod(rs.getString("desc_prod"));
29                 productos.setId_categoria(rs.getInt("id_categoria"));
30                 productos.setUrl_img_prod(rs.getString("url_img_prod"));
31                 productos.setPrecio_prod(rs.getFloat("precio_prod"));
32                 productos.setDisp_prod(rs.getString("disp_prod"));
33                 productos.setEstado_prod(rs.getString("estado_prod"));
34                 //Agregar objeto al arreglo de objetos
35                 lista.add(productos);
36             }
37         } catch (SQLException exception) {
38             System.out.println("Se presentó un error: " + exception.getMessage());
39             return null;
40         }
41     }
42     if (lista.isEmpty()) {
43         System.out.println("La tabla está vacía");
44     }
45
46     return lista;
47 }
48 }
```

```
50 public ProductosModel getProducto(int id) {
51
52     ProductosModel producto = new ProductosModel();
53
54     ConexionBD conex = new ConexionBD();
55     String Sql = "SELECT * FROM productos WHERE id_prod = ?";
56
57     try {
58
59         PreparedStatement pstm = conex.getCon().prepareStatement(Sql);
60         pstm.setInt(1, id);
61
62         ResultSet rs = pstm.executeQuery();
63
64         while (rs.next()) {
65             producto.setId_prod(rs.getInt("id_prod"));
66             producto.setNombre_prod(rs.getString("nombre_prod"));
67             producto.setDesc_prod(rs.getString("desc_prod"));
68             producto.setId_categoria(rs.getInt("id_categoria"));
69             producto.setUrl_img_prod(rs.getString("url_img_prod"));
70             producto.setPrecio_prod(rs.getFloat("precio_prod"));
71             producto.setDisp_prod(rs.getString("disp_prod"));
72             producto.setEstado_prod(rs.getString("estado_prod"));
73         }
74     } catch (SQLException exception) {
75         System.out.println("Se presentó un error: " + exception.getMessage());
76     }
77
78     return producto;
79 }
80 }
```

```
82 public ProductosModel addProducto(ProductosModel producto) {
83     ConexionBD conex = new ConexionBD();
84     String Sql = "INSERT INTO productos (nombre_prod, desc_prod, id_categoria, url_img_prod, precio_prod, disp_prod, estado_prod) ";
85     Sql = Sql + "values (?, ?, ?, ?, ?, ?, ?)";
86
87     try {
88         PreparedStatement pstm = conex.getCon().prepareStatement(Sql);
89
90         pstm.setString(1, producto.getNombre_prod());
91         pstm.setString(2, producto.getDesc_prod());
92         pstm.setInt(3, producto.getId_categoria());
93         pstm.setString(4, producto.getUrl_img_prod());
94         pstm.setFloat(5, producto.getPrecio_prod());
95         pstm.setString(6, producto.getDisp_prod());
96         pstm.setString(7, producto.getEstado_prod());
97
98         pstm.executeUpdate();
99
100     } catch (SQLException exception) {
101         System.out.println("Se presentó un error: " + exception.getMessage());
102         return null;
103     }
104     return producto;
105 }
106 }
```

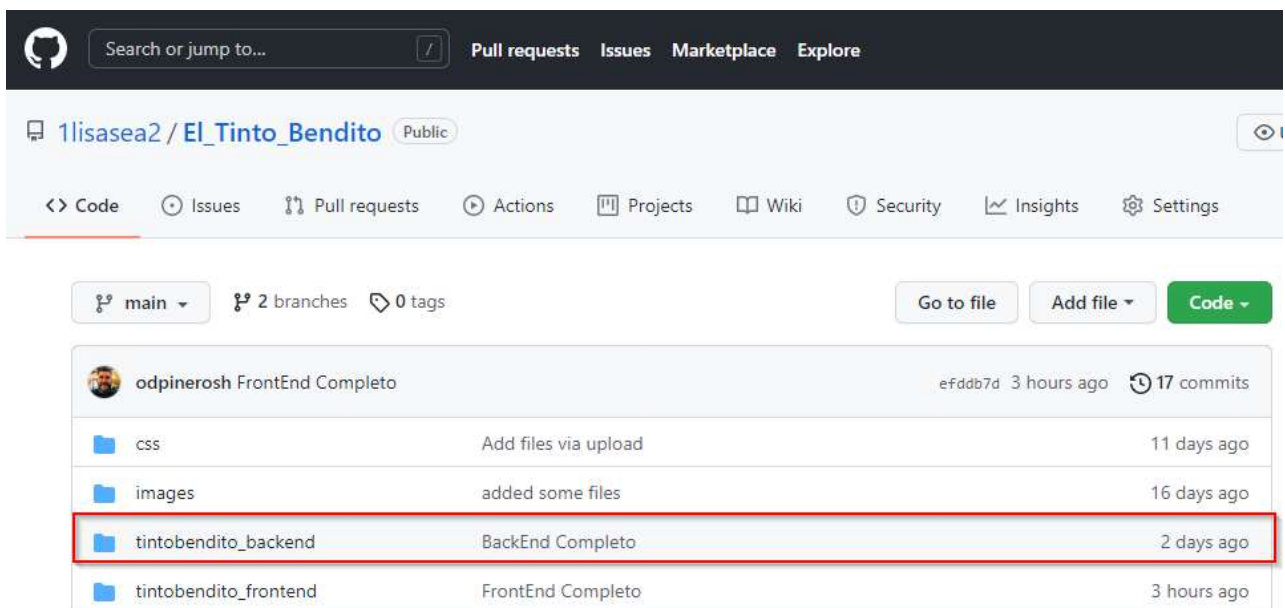


```

107 public ProductosModel updateProducto(ProductosModel producto) {
108     ConexionBD conn = new ConexionBD();
109     String sql = "UPDATE productos SET nombre_prod=?, desc_prod=?, id_categoria=?, url_img_prod=?, precio_prod=?, disp_prod=?, estado_prod=? ";
110     sql = sql + "WHERE id_prod= ?";
111     try {
112         PreparedStatement pstmt = conn.getCon().prepareStatement(sql);
113         pstmt.setString(1, producto.getNombre_prod());
114         pstmt.setString(2, producto.getDesc_prod());
115         pstmt.setInt(3, producto.getId_categoria());
116         pstmt.setString(4, producto.getUrl_img_prod());
117         pstmt.setFloat(5, producto.getPrecio_prod());
118         pstmt.setString(6, producto.getDisp_prod());
119         pstmt.setString(7, producto.getEstado_prod());
120         pstmt.setInt(8, producto.getId_prod());
121
122         pstmt.executeUpdate();
123     } catch (SQLException excepcion) {
124         System.out.println("Ha ocurrido un error al modificar: " + excepcion.getMessage());
125         return null;
126     }
127     return producto;
128 }
129
130 public String delProducto(int id) {
131     ConexionBD conn = new ConexionBD();
132
133     String sql = "DELETE FROM productos WHERE id_prod = ?";
134     try {
135         PreparedStatement pstmt = conn.getCon().prepareStatement(sql);
136         pstmt.setInt(1, id);
137
138         pstmt.executeUpdate();
139     } catch (SQLException excepcion) {
140         System.out.println("Ha ocurrido un error al eliminar: " + excepcion.getMessage());
141         return "\\Accion\\:\\Error\\";
142     }
143     return "\\Accion\\:\\Registro Borrado\\";
144 }
145 }
146

```

1.1. Evidencia del Backend en Github.



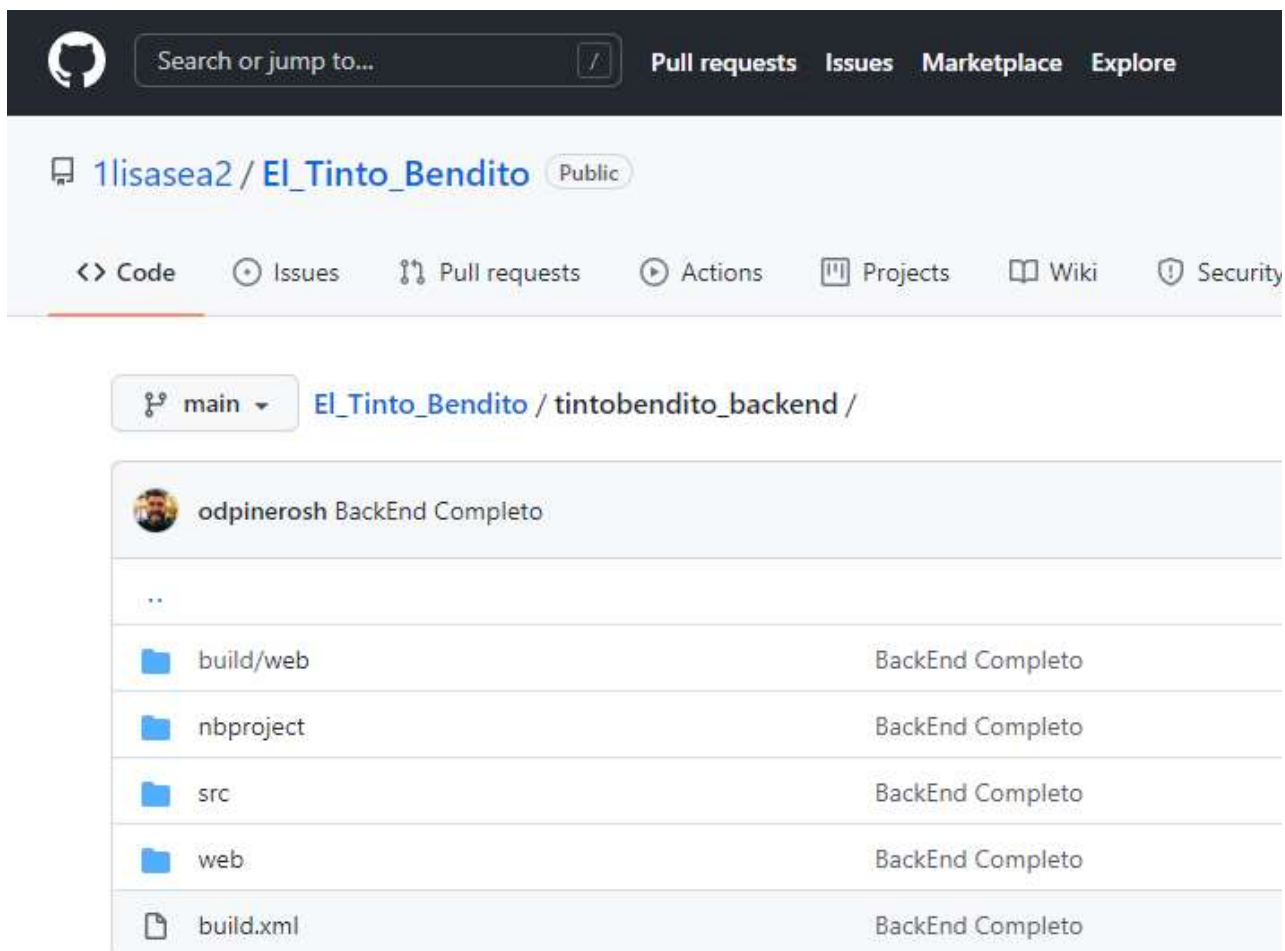
Search or jump to... / Pull requests Issues Marketplace Explore

1lisasea2 / El_Tinto_Bendito Public

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main 2 branches 0 tags Go to file Add file Code

Commit Message	Description	Time Ago
odpinerosh FrontEnd Completo	efddb7d 3 hours ago 17 commits	
css	Add files via upload	11 days ago
images	added some files	16 days ago
tintobendito_backend	BackEnd Completo	2 days ago
tintobendito_frontend	FrontEnd Completo	3 hours ago



The screenshot shows the GitHub interface for the repository 'El_Tinto_Bendito' by user '1lisasea2'. The repository is public. The main branch is 'main'. The repository contains a directory structure with folders 'build/web', 'nbproject', 'src', and 'web', and a file 'build.xml'. All items are marked as 'BackEnd Completo'.

Search or jump to... Pull requests Issues Marketplace Explore

1lisasea2 / El_Tinto_Bendito Public

<> Code Issues Pull requests Actions Projects Wiki Security

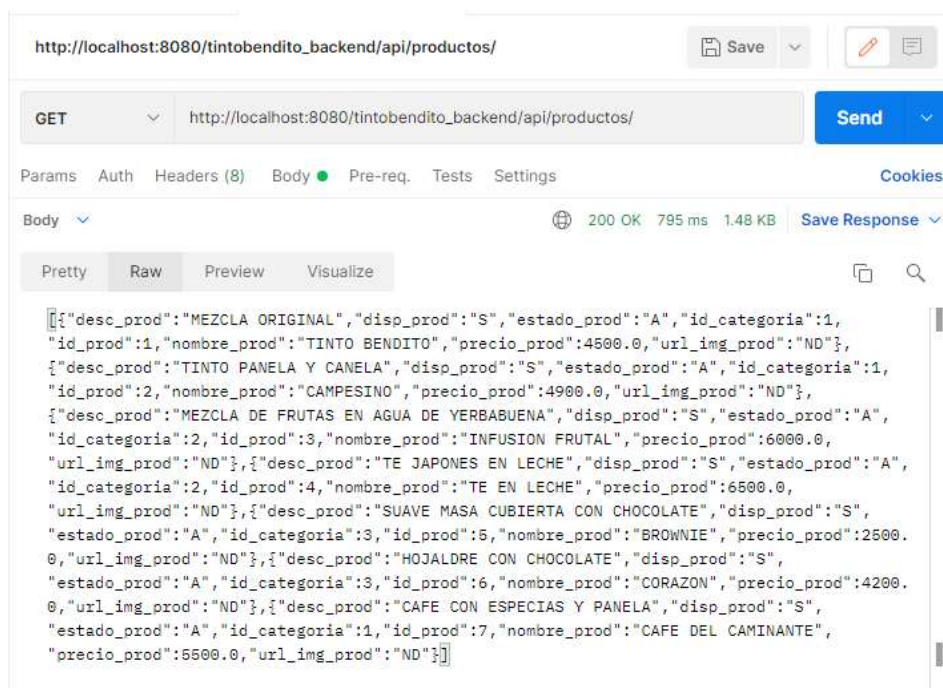
main El_Tinto_Bendito / tintobendito_backend /

odpinerosh BackEnd Completo

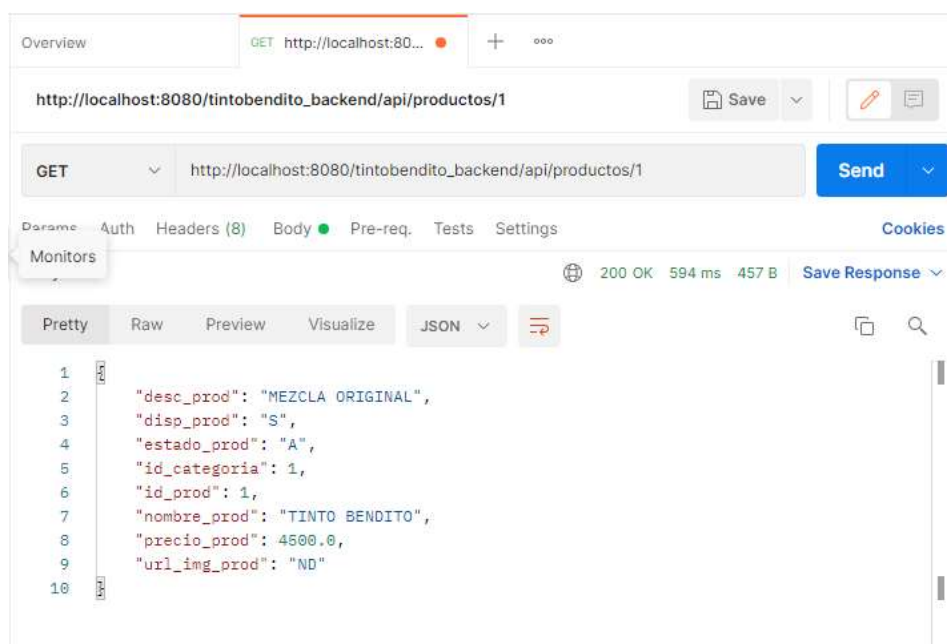
..

build/web	BackEnd Completo
nbproject	BackEnd Completo
src	BackEnd Completo
web	BackEnd Completo
build.xml	BackEnd Completo

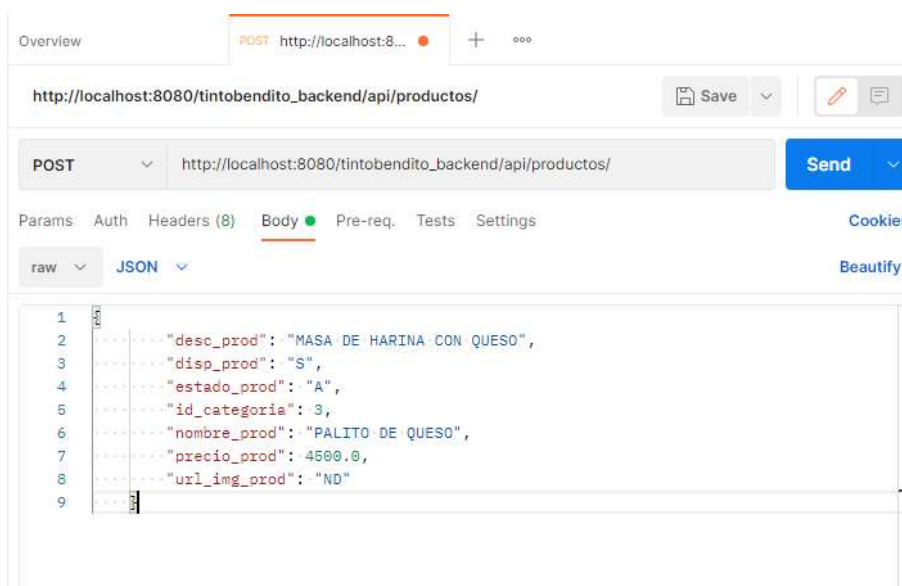
1.2. Ejecución del método GET en Postman.



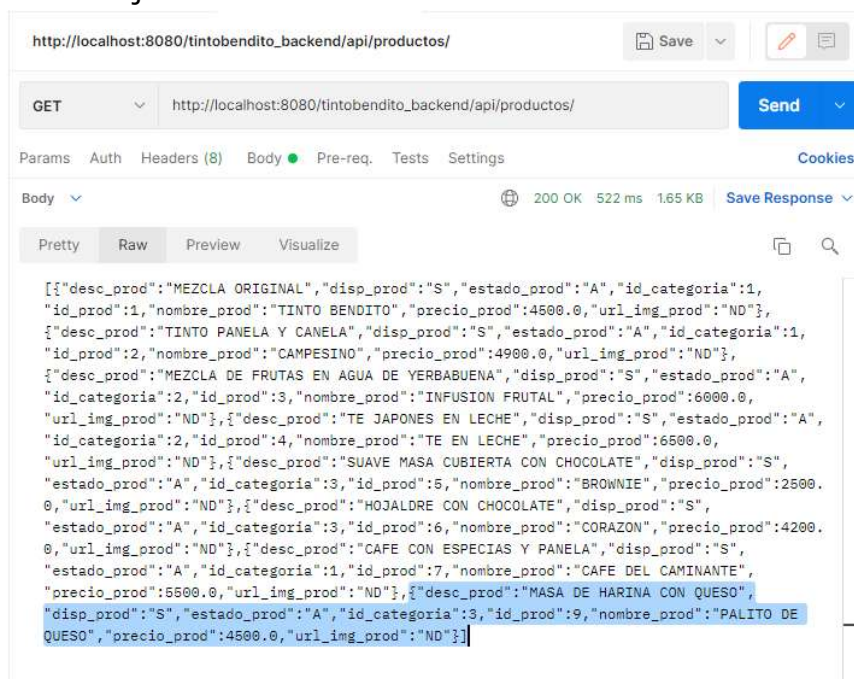
1.3. Ejecución del método GET en Postman con @Path("/{Id_Producto}").



1.4. Ejecución de método POST en Postman.



1.5. Verificación de la ejecución de método POST en Postman.



1.6. Verificación de la ejecución de método POST en MySQL.

Query 1 x

Limit to 1000 rows

1 • **SELECT * FROM PRODUCTOS**

id_prod	nombre_prod	desc_prod	id_categoria	url_img_prod	precio_prod	disp_prod	estado_prod
1	TINTO BENDITO	MEZCLA ORIGINAL	1	ND	4500	S	A
2	CAMPESINO	TINTO PANELA Y CANELA	1	ND	4900	S	A
3	INFUSION FRUTAL	MEZCLA DE FRUTAS EN AG...	2	ND	6000	S	A
4	TE EN LECHE	TE JAPONES EN LECHE	2	ND	6500	S	A
5	BROWNIE	SUAVE MASA CUBIERTA C...	3	ND	2500	S	A
6	CORAZON	HOJALDRE CON CHOCOLATE	3	ND	4200	S	A
7	CAFE DEL CAMINANTE	CAFE CON ESPECIAS Y PA...	1	ND	5500	S	A
9	PALITO DE QUESO	MASA DE HARINA CON QU...	3	ND	4500	S	A
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

Result Grid | Form Editor | Field Types

1.7. Ejecución del método PUT en MySQL sobre el producto con id=3.

- Verificación de la existencia del registro con el método GET en Postman.

Overview GET http://localhost:80...

http://localhost:8080/tintobendito_backend/api/productos/4

GET http://localhost:8080/tintobendito_backend/api/productos/4

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

Body 200 OK 284 ms 459 B Save Response

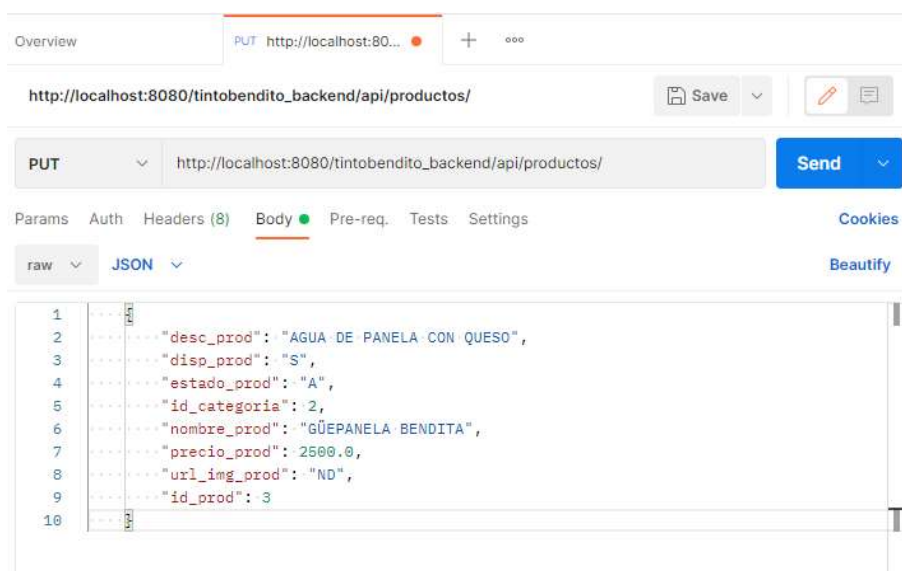
Pretty Raw Preview Visualize JSON

```

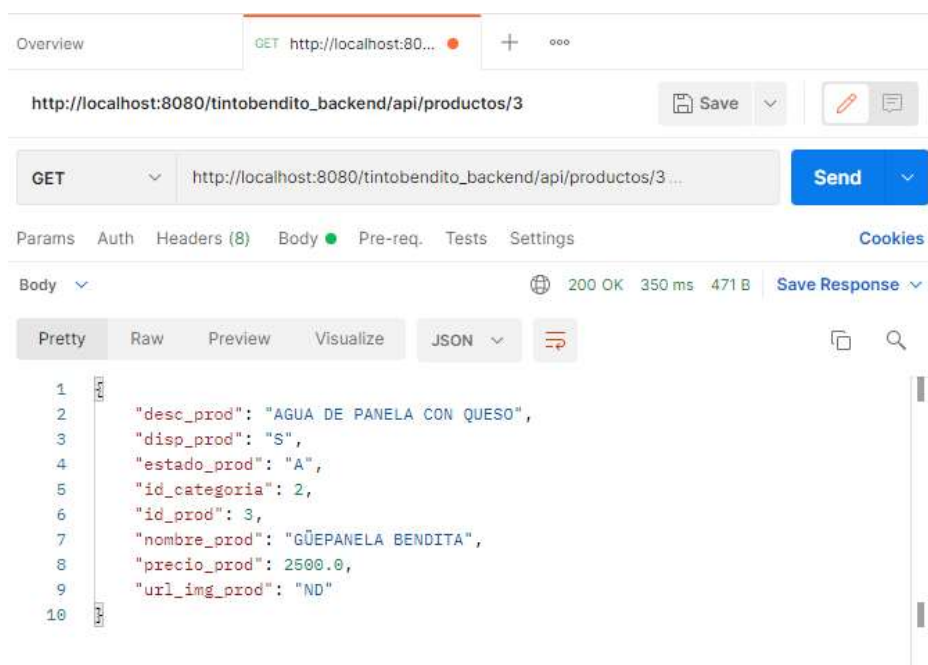
1  {
2    "desc_prod": "TE JAPONES EN LECHE",
3    "disp_prod": "S",
4    "estado_prod": "A",
5    "id_categoria": 2,
6    "id_prod": 4,
7    "nombre_prod": "TE EN LECHE",
8    "precio_prod": 6500.0,
9    "url_img_prod": "ND"
10 }

```

- Ejecución del método PUT: Se cambiará la descripción del producto, el nombre y la categoría.



- Verificación de la ejecución del método PUT con el método GET en Postman.



- Verificación de la ejecución del método PUT en MySQL.

Query 1 x

Limit to 1000 rows

1 • **SELECT * FROM PRODUCTOS WHERE ID_PROD = 3**

Result Grid

	id_prod	nombre_prod	desc_prod	id_categoria	url_img_prod	precio_prod	disp_prod	estado_prod
▶	3	GÜEPANELA BENDITA	AGUA DE PANELA CON QU...	2	ND	2500	S	A
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid

1.8. Ejecución del método DELETE sobre el producto con id=4.

- Verificación de la existencia del registro con el método GET en Postman.

Overview GET http://localhost:80...

http://localhost:8080/tintobendito_backend/api/productos/4

GET http://localhost:8080/tintobendito_backend/api/productos/4

Params Auth Headers (8) Body Pre-req. Tests Settings Cookies

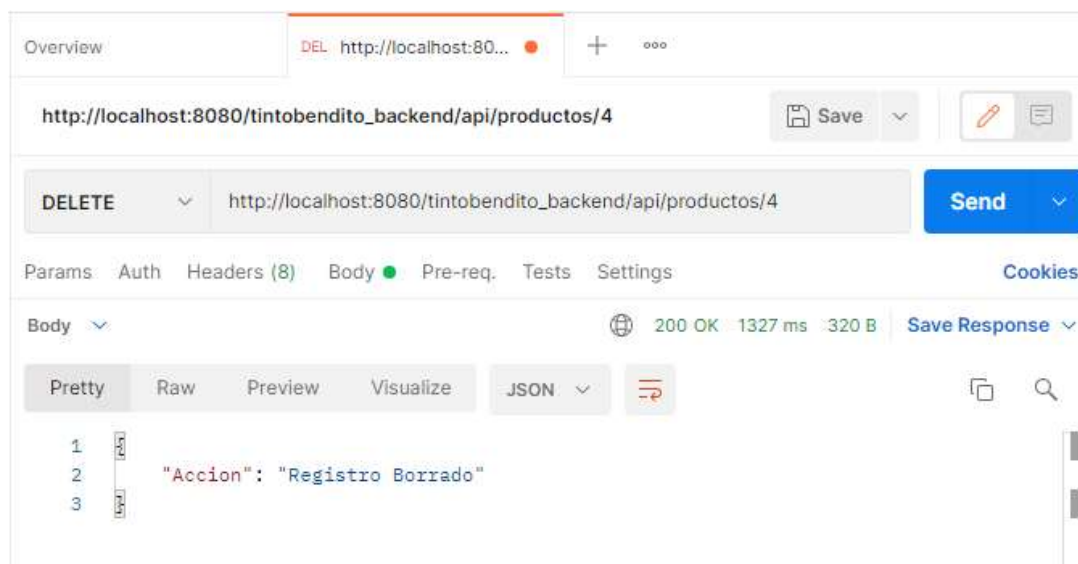
Body 200 OK 284 ms 459 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "desc_prod": "TE JAPONES EN LECHE",
3    "disp_prod": "S",
4    "estado_prod": "A",
5    "id_categoria": 2,
6    "id_prod": 4,
7    "nombre_prod": "TE EN LECHE",
8    "precio_prod": 6500.0,
9    "url_img_prod": "ND"
10 }
```

- Ejecución del Método DELETE en Postman.



- Verificación de la ejecución en MySQL.

Query 1 x

Limit to 1000 rows

1 • **SELECT * FROM PRODUCTOS**

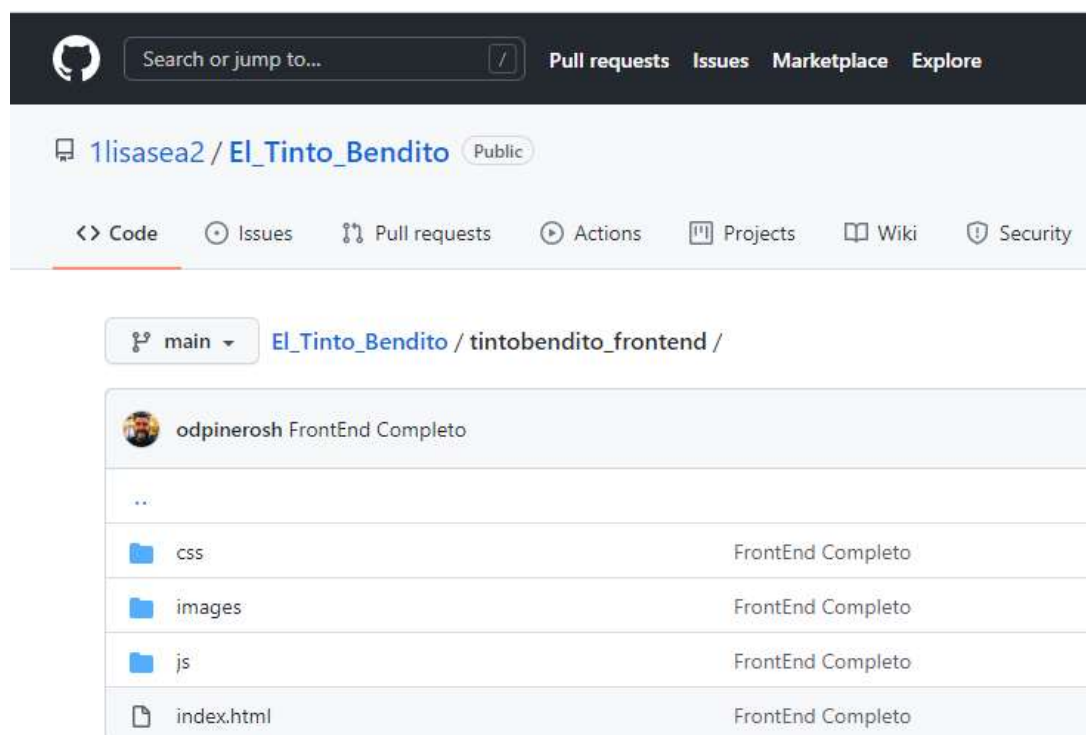
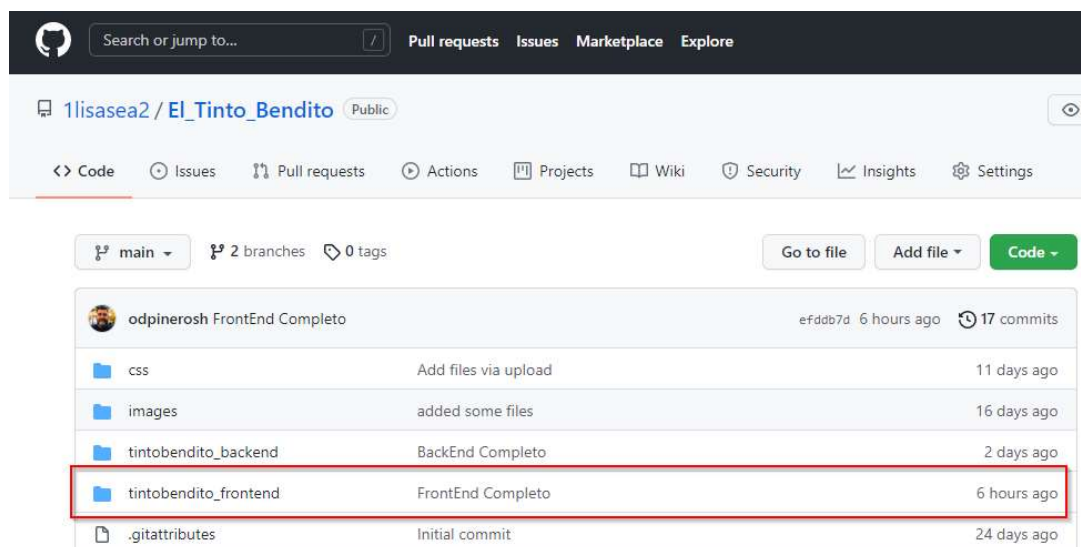
Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

	id_prod	nombre_prod	desc_prod	id_categoria	url_img_prod	precio_prod	disp_prod	estado_prod
▶	1	TINTO BENDITO	MEZCLA ORIGINAL	1	ND	4500	S	A
	2	CAMPESINO	TINTO PANELA Y CANELA	1	ND	4900	S	A
	3	GÜEPANELA BENDITA	AGUA DE PANELA CON QU...	2	ND	2500	S	A
	5	BROWNIE	SUAVE MASA CUBIERTA C...	3	ND	2500	S	A
	6	CORAZON	HOJALDRE CON CHOCOLATE	3	ND	4200	S	A
	7	CAFE DEL CAMINANTE	CAFE CON ESPECIAS Y PA...	1	ND	5500	S	A
	9	PALITO DE QUESO	MASA DE HARINA CON QU...	3	ND	4500	S	A
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid | Form Editor | Field Types

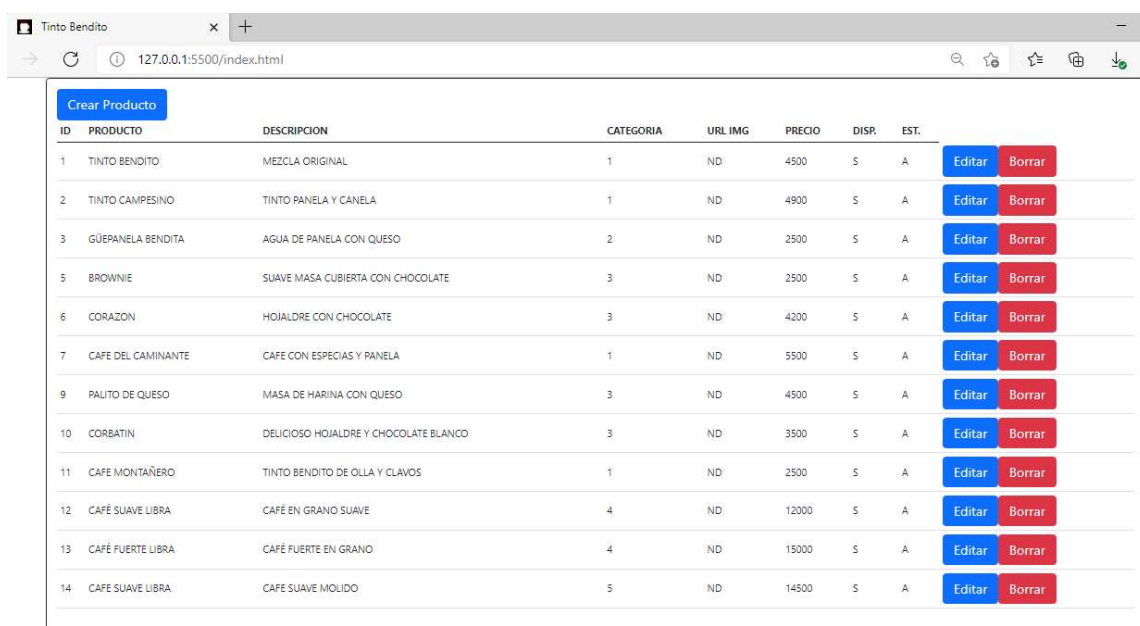
2. APLICACIÓN FRONTEND.

2.1. Evidencia del Frontend en Github.



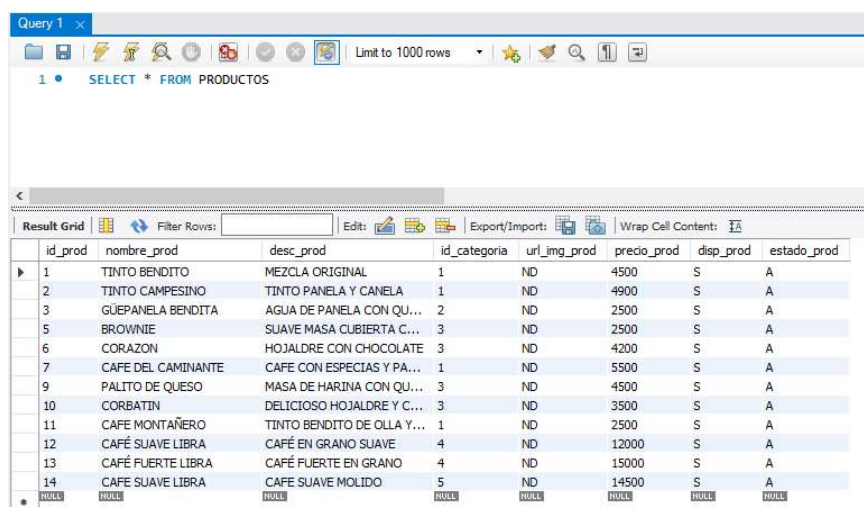
2.2. Consumo del método GET.

En la siguiente imagen se observa como el Frontend está consumiendo el servicios GET ofrecido por la API para mostrar la lista de los producto.



ID	PRODUCTO	DESCRIPCION	CATEGORIA	URL IMG	PRECIO	DISP.	EST.		
1	TINTO BENDITO	MEZCLA ORIGINAL	1	ND	4500	S	A	Editar	Borrar
2	TINTO CAMPESINO	TINTO PANELA Y CANELA	1	ND	4900	S	A	Editar	Borrar
3	GÜEPANELA BENDITA	AGUA DE PANELA CON QUESO	2	ND	2500	S	A	Editar	Borrar
5	BROWNIE	SUAVE MASA CUBIERTA CON CHOCOLATE	3	ND	2500	S	A	Editar	Borrar
6	CORAZON	HOJALDRE CON CHOCOLATE	3	ND	4200	S	A	Editar	Borrar
7	CAFE DEL CAMINANTE	CAFE CON ESPECIAS Y PANELA	1	ND	5500	S	A	Editar	Borrar
9	PALITO DE QUESO	MASA DE HARINA CON QUESO	3	ND	4500	S	A	Editar	Borrar
10	CORBATIN	DELICIOSO HOJALDRE Y CHOCOLATE BLANCO	3	ND	3500	S	A	Editar	Borrar
11	CAFE MONTAÑERO	TINTO BENDITO DE OLLA Y CLAVOS	1	ND	2500	S	A	Editar	Borrar
12	CAFÉ SUAVE LIBRA	CAFÉ EN GRANO SUAVE	4	ND	12000	S	A	Editar	Borrar
13	CAFÉ FUERTE LIBRA	CAFÉ FUERTE EN GRANO	4	ND	15000	S	A	Editar	Borrar
14	CAFE SUAVE LIBRA	CAFE SUAVE MOLIDO	5	ND	14500	S	A	Editar	Borrar

Este contenido existe efectivamente en la tabla PRODUCTOS de la base de datos, tal y como se evidencia a continuación realizando la consulta con MySQL.



Query 1

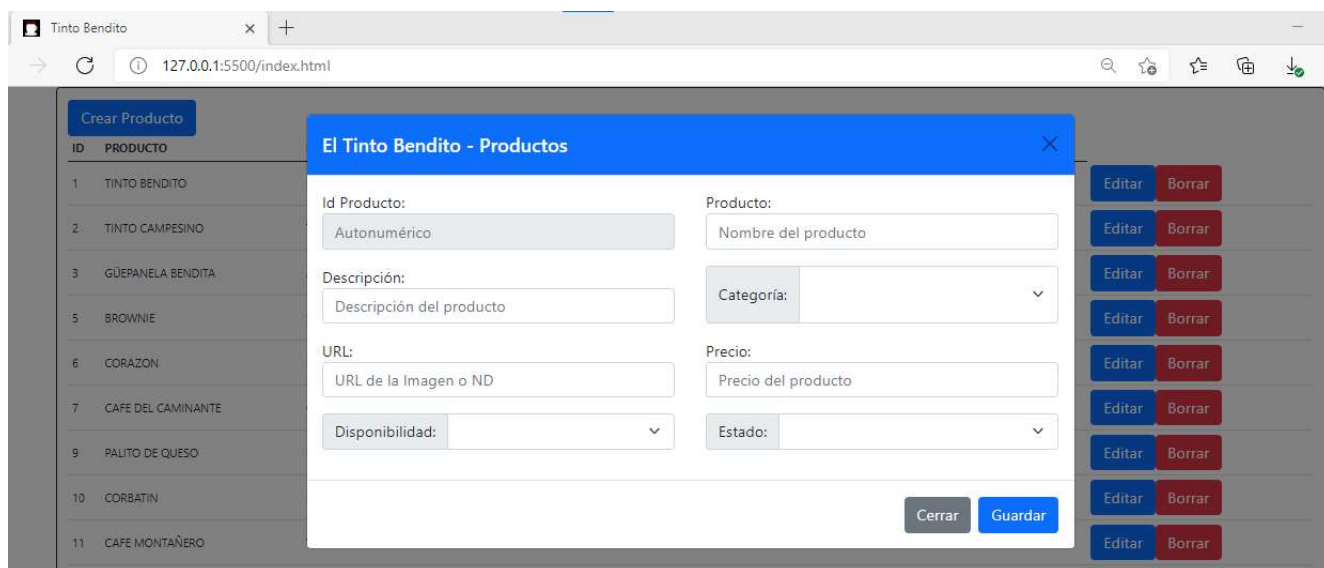
Limit to 1000 rows

1 • SELECT * FROM PRODUCTOS

	id_prod	nombre_prod	desc_prod	id_categoria	url_img_prod	precio_prod	disp_prod	estado_prod
1	1	TINTO BENDITO	MEZCLA ORIGINAL	1	ND	4500	S	A
2	2	TINTO CAMPESINO	TINTO PANELA Y CANELA	1	ND	4900	S	A
3	3	GUEPANELA BENDITA	AGUA DE PANELA CON QU...	2	ND	2500	S	A
5	5	BROWNIE	SUAVE MASA CUBIERTA C...	3	ND	2500	S	A
6	6	CORAZON	HOJALDRE CON CHOCOLATE	3	ND	4200	S	A
7	7	CAFE DEL CAMINANTE	CAFE CON ESPECIAS Y PA...	1	ND	5500	S	A
9	9	PALITO DE QUESO	MASA DE HARINA CON QU...	3	ND	4500	S	A
10	10	CORBATIN	DELICIOSO HOJALDRE Y C...	3	ND	3500	S	A
11	11	CAFE MONTAÑERO	TINTO BENDITO DE OLLA Y...	1	ND	2500	S	A
12	12	CAFÉ SUAVE LIBRA	CAFÉ EN GRANO SUAVE	4	ND	12000	S	A
13	13	CAFÉ FUERTE LIBRA	CAFÉ FUERTE EN GRANO	4	ND	15000	S	A
14	14	CAFE SUAVE LIBRA	CAFE SUAVE MOLIDO	5	ND	14500	S	A

2.3. Creación de un producto nuevo – Método POST.

Al hacer clic sobre el botón “Crear Producto”, la aplicación muestra una ventana modal con los campos vacíos con el fin de que el usuario proporcione los datos del nuevo producto, tal y como se ve en la siguiente imagen.

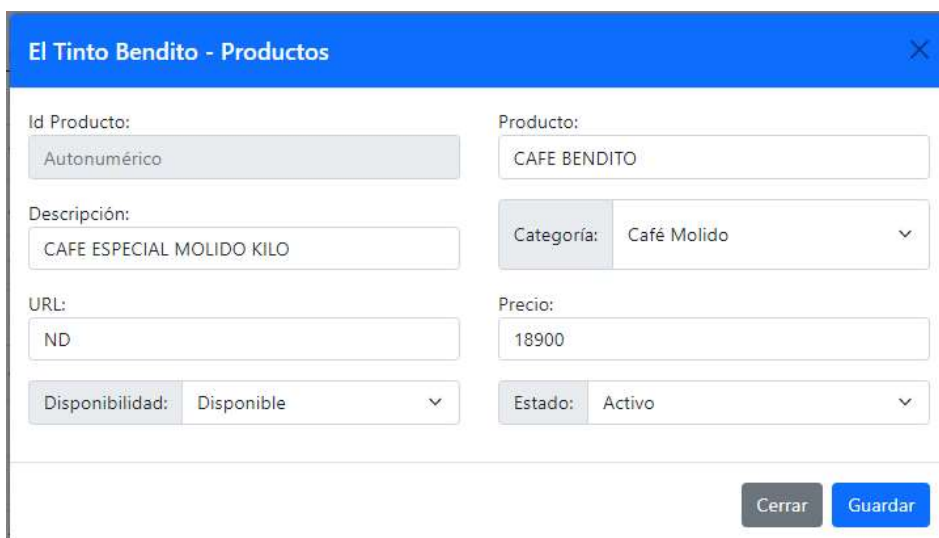


The screenshot shows a web browser window with the URL 127.0.0.1:5500/index.html. A modal titled "El Tinto Bendito - Productos" is open. On the left, there is a sidebar with a "Crear Producto" button and a list of products. The modal contains the following fields:

- Id Producto:** Autonumérico
- Producto:** Nombre del producto
- Descripción:** Descripción del producto
- Categoría:** (dropdown menu)
- URL:** URL de la Imagen o ND
- Precio:** Precio del producto
- Disponibilidad:** (dropdown menu)
- Estado:** (dropdown menu)

At the bottom right of the modal are buttons for "Cerrar" and "Guardar". To the right of the modal, a list of existing products is visible, each with "Editar" and "Borrar" buttons.

A continuación, se muestra la ventana con los campos diligenciados antes de hacer clic sobre el botón “Guardar”.



The screenshot shows the same modal form, but now with the following data entered:

- Id Producto:** Autonumérico
- Producto:** CAFE BENDITO
- Descripción:** CAFE ESPECIAL MOLIDO KILO
- Categoría:** Café Molido
- URL:** ND
- Precio:** 18900
- Disponibilidad:** Disponible
- Estado:** Activo

The "Cerrar" and "Guardar" buttons remain at the bottom right.

Al hacer clic sobre el botón “Guardar”, la aplicación consume el servicio POST, agregando el registro a la base de datos. La aplicación muestra una ventana modal informando del éxito de la operación.



Al hacer clic sobre el botón “Aceptar” de esta modal, se ejecuta la actualización de la página, mostrando la lista de productos actualizada, tal y como se muestra en la imagen a continuación.

Crear Producto								
ID	PRODUCTO	DESCRIPCION	CATEGORIA	URL IMG	PRECIO	DISP.	EST.	
1	TINTO BENDITO	MEZCLA ORIGINAL	1	ND	4500	S	A	Editar Borrar
2	TINTO CAMPESINO	TINTO PANELA Y CANELA	1	ND	4900	S	A	Editar Borrar
3	GÜEPANELA BENDITA	AGUA DE PANELA CON QUESO	2	ND	2500	S	A	Editar Borrar
5	BROWNIE	SUAVE MASA CUBIERTA CON CHOCOLATE	3	ND	2500	S	A	Editar Borrar
6	CORAZON	HOJALDRE CON CHOCOLATE	3	ND	4200	S	A	Editar Borrar
7	CAFE DEL CAMINANTE	CAFE CON ESPECIAS Y PANELA	1	ND	5500	S	A	Editar Borrar
9	PALITO DE QUESO	MASA DE HARINA CON QUESO	3	ND	4500	S	A	Editar Borrar
10	CORBATIN	DELICIOSO HOJALDRE Y CHOCOLATE BLANCO	3	ND	3500	S	A	Editar Borrar
11	CAFE MONTAÑERO	TINTO BENDITO DE OLLA Y CLAVOS	1	ND	2500	S	A	Editar Borrar
12	CAFÉ SUAVE LIBRA	CAFÉ EN GRANO SUAVE	4	ND	12000	S	A	Editar Borrar
13	CAFÉ FUERTE LIBRA	CAFÉ FUERTE EN GRANO	4	ND	15000	S	A	Editar Borrar
14	CAFE SUAVE LIBRA	CAFE SUAVE MOLIDO	5	ND	14500	S	A	Editar Borrar
15	CAFE BENDITO	CAFE ESPECIAL MOLIDO KILO	5	ND	18900	S	A	Editar Borrar

El producto se puede verificar también en la tabla, consultado por MySQL, como se muestra en la imagen a continuación.

Query 1

Limit to 1000 rows

1 • SELECT * FROM PRODUCTOS

	id_prod	nombre_prod	desc_prod	id_categoria	url_img_prod	precio_prod	disp_prod	estado_prod
1	1	TINTO BENDITO	MEZCLA ORIGINAL	1	ND	4500	S	A
2	2	TINTO CAMPESINO	TINTO PANELA Y CANELA	1	ND	4900	S	A
3	3	GÜEPANELA BENDITA	AGUA DE PANELA CON QU...	2	ND	2500	S	A
5	5	BROWNIE	SUAVE MASA CUBIERTA C...	3	ND	2500	S	A
6	6	CORAZON	HOJALDRE CON CHOCOLATE	3	ND	4200	S	A
7	7	CAFE DEL CAMINANTE	CAFE CON ESPECIAS Y PA...	1	ND	5500	S	A
9	9	PALITO DE QUESO	MASA DE HARINA CON QU...	3	ND	4500	S	A
10	10	CORBATIN	DELICIOSO HOJALDRE Y C...	3	ND	3500	S	A
11	11	CAFE MONTAÑERO	TINTO BENDITO DE OLLA Y...	1	ND	2500	S	A
12	12	CAFÉ SUAVE LIBRA	CAFÉ EN GRANO SUAVE	4	ND	12000	S	A
13	13	CAFÉ FUERTE LIBRA	CAFÉ FUERTE EN GRANO	4	ND	15000	S	A
14	14	CAFÉ SUAVE LIBRA	CAFÉ SUAVE MOLIDO	5	ND	14500	S	A
15	15	CAFÉ BENDITO	CAFÉ ESPECIAL MOLIDO KILO	5	ND	18900	S	A
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2.4. Edición de un producto Método PUT.

Para editar un registro existente, el usuario debe hacer clic sobre el botón “Editar” correspondiente a la fila de la tabla que muestra el producto. El clic sobre dicho botón hace que la aplicación muestre una ventana modal cuyos elementos vienen precargados con los datos del registro. Para este ejemplo, se editará el registro con ID 5, así:

Tinto Bendito

127.0.0.1:5500/index.html

Crear Producto

ID	PRODUCTO
1	TINTO BENDITO
2	TINTO CAMPESINO
3	GÜEPANELA BENDITA
5	BROWNIE
6	CORAZON
7	CAFE DEL CAMINANTE
9	PALITO DE QUESO
10	CORBATIN
11	CAFE MONTAÑERO

El Tinto Bendito - Productos

Id Producto: 5

Producto: BROWNIE

Descripción: SUAVE MASA CUBIERTA CON CHOCOLATE

Categoría: Pastelería

URL: ND

Precio: 2500

Disponibilidad: Disponible

Estado: Activo

Cerrar Guardar

Editar Borrar

Editar Borrar

Editar Borrar

Editar Borrar

Editar Borrar

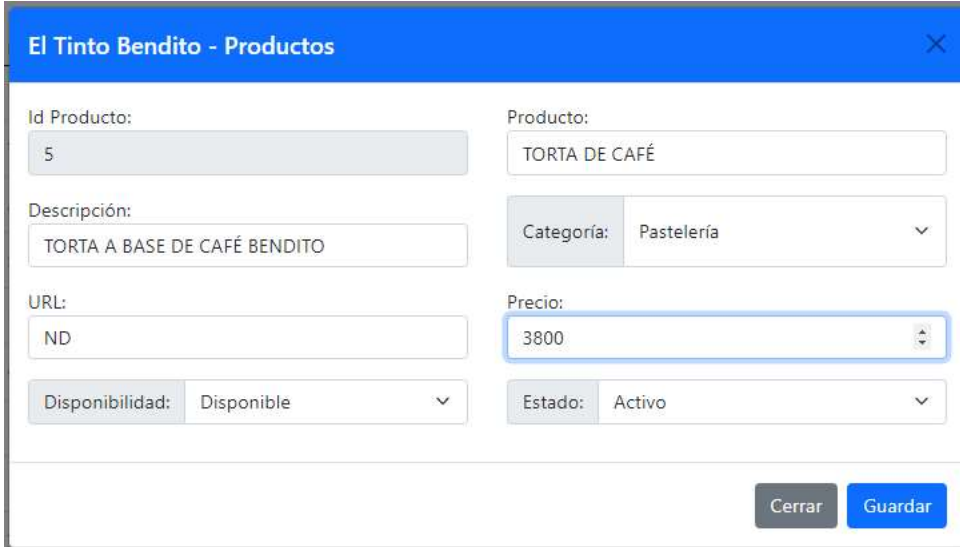
Editar Borrar

Editar Borrar

Editar Borrar

Editar Borrar

Se cambia el nombre, la descripción y el precio del producto.

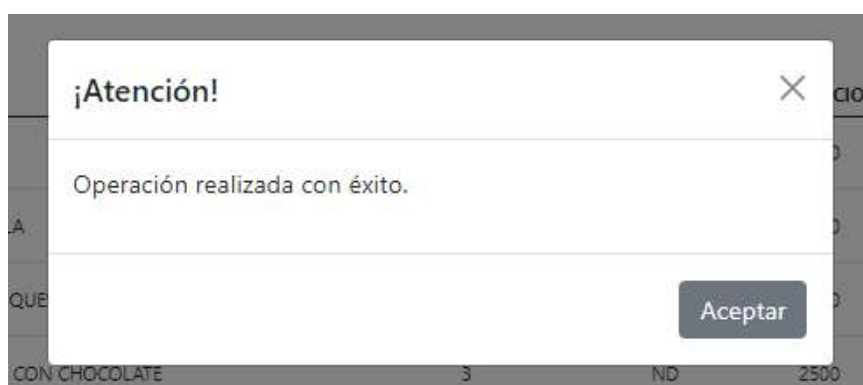


The screenshot shows a web form titled "El Tinto Bendito - Productos". The form contains the following fields and values:

- Id Producto:** 5
- Producto:** TORTA DE CAFÉ
- Descripción:** TORTA A BASE DE CAFÉ BENDITO
- Categoría:** Pastelería
- URL:** ND
- Precio:** 3800
- Disponibilidad:** Disponible
- Estado:** Activo

At the bottom right of the form are two buttons: "Cerrar" and "Guardar".

Al hacer clic sobre el botón "Guardar", la aplicación consume el servicio PUT, modificando el registro en la base de datos. La aplicación muestra una ventana modal informando del éxito de la operación:



Al hacer clic sobre el botón “Aceptar” de esta modal se ejecuta la actualización de la página mostrando la lista de productos actualizada, tal y como se muestra en la imagen a continuación:

Crear Producto								
ID	PRODUCTO	DESCRIPCION	CATEGORIA	URL IMG	PRECIO	DISP.	EST.	
1	TINTO BENDITO	MEZCLA ORIGINAL	1	ND	4500	S	A	Editar Borrar
2	TINTO CAMPESINO	TINTO PANELA Y CANELA	1	ND	4900	S	A	Editar Borrar
3	GÜEPANELA BENDITA	AGUA DE PANELA CON QUESO	2	ND	2500	S	A	Editar Borrar
5	TORTA DE CAFÉ	TORTA A BASE DE CAFÉ BENDITO	3	ND	3800	S	A	Editar Borrar
6	CORAZON	HOJALDRE CON CHOCOLATE	3	ND	4200	S	A	Editar Borrar
7	CAFE DEL CAMINANTE	CAFE CON ESPECIAS Y PANELA	1	ND	5500	S	A	Editar Borrar
9	PALITO DE QUESO	MASA DE HARINA CON QUESO	3	ND	4500	S	A	Editar Borrar
10	CORBATIN	DELICIOSO HOJALDRE Y CHOCOLATE BLANCO	3	ND	3500	S	A	Editar Borrar
11	CAFE MONTAÑERO	TINTO BENDITO DE OLLA Y CLAVOS	1	ND	2500	S	A	Editar Borrar
12	CAFÉ SUAVE LIBRA	CAFÉ EN GRANO SUAVE	4	ND	12000	S	A	Editar Borrar
13	CAFÉ FUERTE LIBRA	CAFÉ FUERTE EN GRANO	4	ND	15000	S	A	Editar Borrar
14	CAFE SUAVE LIBRA	CAFE SUAVE MOLIDO	5	ND	14500	S	A	Editar Borrar
15	CAFE BENDITO	CAFE ESPECIAL MOLIDO KILO	5	ND	18900	S	A	Editar Borrar

Esto mismo puede verificarse en la tabla por medio de MySQL, tal y como se muestra en la siguiente imagen:

Query 1 x

Limit to 1000 rows

1 • SELECT * FROM PRODUCTOS

	id_prod	nombre_prod	desc_prod	id_categoria	url_img_prod	precio_prod	disp_prod	estado_prod
▶	1	TINTO BENDITO	MEZCLA ORIGINAL	1	ND	4500	S	A
	2	TINTO CAMPESINO	TINTO PANELA Y CANELA	1	ND	4900	S	A
	3	GÜEPANELA BENDITA	AGUA DE PANELA CON QU...	2	ND	2500	S	A
	5	TORTA DE CAFÉ	TORTA A BASE DE CAFÉ BE...	3	ND	3800	S	A
	6	CORAZON	HOJALDRE CON CHOCOLATE	3	ND	4200	S	A
	7	CAFE DEL CAMINANTE	CAFE CON ESPECIAS Y PA...	1	ND	5500	S	A
	9	PALITO DE QUESO	MASA DE HARINA CON QU...	3	ND	4500	S	A
	10	CORBATIN	DELICIOSO HOJALDRE Y C...	3	ND	3500	S	A
	11	CAFE MONTAÑERO	TINTO BENDITO DE OLLA Y...	1	ND	2500	S	A
	12	CAFÉ SUAVE LIBRA	CAFÉ EN GRANO SUAVE	4	ND	12000	S	A
	13	CAFÉ FUERTE LIBRA	CAFÉ FUERTE EN GRANO	4	ND	15000	S	A
	14	CAFÉ SUAVE LIBRA	CAFÉ SUAVE MOLIDO	5	ND	14500	S	A
	15	CAFE BENDITO	CAFE ESPECIAL MOLIDO KILO	5	ND	18900	S	A
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2.5. Borrado de un registro Método DELETE.

Para borrar un registro existente, el usuario debe hacer clic sobre el botón "Borrar" correspondiente a la fila de la tabla que muestra el producto. Para el ejemplo se seleccionó el producto con ID 9. Cuando el usuario hace clic sobre el botón, la aplicación muestra una ventana modal de confirmación, así:

Crear Producto

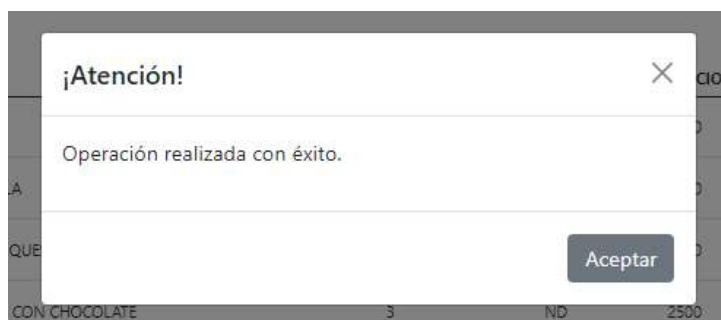
ID	PRODUCTO	DESCRIPCION	CATEGORIA	URL IMG	PRECIO	DISP.	EST.	
1	TINTO BENDITO	MEZCLA ORIGINAL				S	A	Editar Borrar
2	TINTO CAMPESINO	TINTO PANELA Y CANELA				S	A	Editar Borrar
3	GÜEPANELA BENDITA	AGUA DE PANELA CON QUE				S	A	Editar Borrar
5	TORTA DE CAFÉ	TORTA A BASE DE CAFÉ BEN				S	A	Editar Borrar
6	CORAZON	HOJALDRE CON CHOCOLATE	3	ND	4200	S	A	Editar Borrar
7	CAFE DEL CAMINANTE	CAFE CON ESPECIAS Y PANELA	1	ND	5500	S	A	Editar Borrar
9	PALITO DE QUESO	MASA DE HARINA CON QUESO	3	ND	4500	S	A	Editar Borrar

¡Atención!

¿Estás seguro de eliminar el producto 9 - PALITO DE QUESO?

OK Cancel

Para ejecutar el borrado, el usuario debe hacer clic sobre el botón OK. Esto finalmente consume el recurso DELETE, eliminando el registro. La aplicación muestra una ventana modal informando del éxito de la operación:



Al hacer clic sobre el botón "Aceptar" de esta modal se ejecuta la actualización de la página mostrando finalmente la lista de productos actualizada, tal y como se muestra en la imagen a continuación:

Tinto Bendito

127.0.0.1:5500/index.html

Crear Producto

ID	PRODUCTO	DESCRIPCION	CATEGORIA	URL IMG	PRECIO	DISP.	EST.		
1	TINTO BENDITO	MEZCLA ORIGINAL	1	ND	4500	S	A	Editar	Borrar
2	TINTO CAMPESINO	TINTO PANELA Y CANELA	1	ND	4900	S	A	Editar	Borrar
3	GÜEPANELA BENDITA	AGUA DE PANELA CON QUESO	2	ND	2500	S	A	Editar	Borrar
5	TORTA DE CAFÉ	TORTA A BASE DE CAFÉ BENDITO	3	ND	3800	S	A	Editar	Borrar
6	CORAZON	HOJALDRE CON CHOCOLATE	3	ND	4200	S	A	Editar	Borrar
7	CAFE DEL CAMINANTE	CAFE CON ESPECIAS Y PANELA	1	ND	5500	S	A	Editar	Borrar
10	CORBATIN	DELICIOSO HOJALDRE Y CHOCOLATE BLANCO	3	ND	3500	S	A	Editar	Borrar
11	CAFE MONTAÑERO	TINTO BENDITO DE OLLA Y CLAVOS	1	ND	2500	S	A	Editar	Borrar
12	CAFÉ SUAVE LIBRA	CAFÉ EN GRANO SUAVE	4	ND	12000	S	A	Editar	Borrar
13	CAFÉ FUERTE LIBRA	CAFÉ FUERTE EN GRANO	4	ND	15000	S	A	Editar	Borrar
14	CAFE SUAVE LIBRA	CAFE SUAVE MOLIDO	5	ND	14500	S	A	Editar	Borrar
15	CAFE BENDITO	CAFE ESPECIAL MOLIDO KILO	5	ND	18900	S	A	Editar	Borrar

Esto mismo puede verificarse en la tabla por medio de MySQL, tal y como se muestra en la siguiente imagen:

Query 1 x

Limit to 1000 rows

1 • SELECT * FROM PRODUCTOS

Result Grid

	id_prod	nombre_prod	desc_prod	id_categoria	url_img_prod	precio_prod	disp_prod	estado_prod
▶	1	TINTO BENDITO	MEZCLA ORIGINAL	1	ND	4500	S	A
	2	TINTO CAMPESINO	TINTO PANELA Y CANELA	1	ND	4900	S	A
	3	GÜEPANELA BENDITA	AGUA DE PANELA CON QU...	2	ND	2500	S	A
	5	TORTA DE CAFÉ	TORTA A BASE DE CAFÉ BE...	3	ND	3800	S	A
	6	CORAZON	HOJALDRE CON CHOCOLATE	3	ND	4200	S	A
	7	CAFE DEL CAMINANTE	CAFE CON ESPECIAS Y PA...	1	ND	5500	S	A
	10	CORBATIN	DELICIOSO HOJALDRE Y C...	3	ND	3500	S	A
	11	CAFE MONTAÑERO	TINTO BENDITO DE OLLA Y...	1	ND	2500	S	A
	12	CAFÉ SUAVE LIBRA	CAFÉ EN GRANO SUAVE	4	ND	12000	S	A
	13	CAFÉ FUERTE LIBRA	CAFÉ FUERTE EN GRANO	4	ND	15000	S	A
	14	CAFE SUAVE LIBRA	CAFE SUAVE MOLIDO	5	ND	14500	S	A
	15	CAFE BENDITO	CAFE ESPECIAL MOLIDO KILO	5	ND	18900	S	A
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL