**CELLWD – DOCUMENTACION APIS**  
  
   
  
  
  
 **DESARROLLO DE SOFTWARE WEB BACK-END**

**WILSON EDUARDO GOMEZ MORA** [**weduardogomez@ucompensar.edu.co**](mailto:weduardogomez@ucompensar.edu.co) **PAULA DAYANA MORA ACOSTA**   
[**pdmora@ucompensar.edu.co**](mailto:pdmora@ucompensar.edu.co) **DOCENTE :** [**CARLOS HUGO NEIVA REYES**](https://virtual.ucompensar.edu.co/user/profile.php?id=249254) **FUNDACION UNIVERSITARIA COMPENSAR   
 26/09/2024**Contenido  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Introducción   
En el desarrollo de aplicaciones web, las API (Interfaz de Programación de Aplicaciones) juegan un papel fundamental al permitir la interacción entre el cliente y el servidor. Las API facilitan el intercambio de datos mediante el uso de métodos HTTP estándar como GET, POST, PUT y DELETE, que permiten realizar operaciones de lectura, creación, actualización y eliminación de recursos en una base de datos. A lo largo de este documento, explicaremos cómo se han aplicado estos métodos en un proyecto de venta de celulares, detallando su función y el manejo de la información desde el servidor hasta la base de datos.  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
¿Qué son las API GET, POST, PUT y DELETE

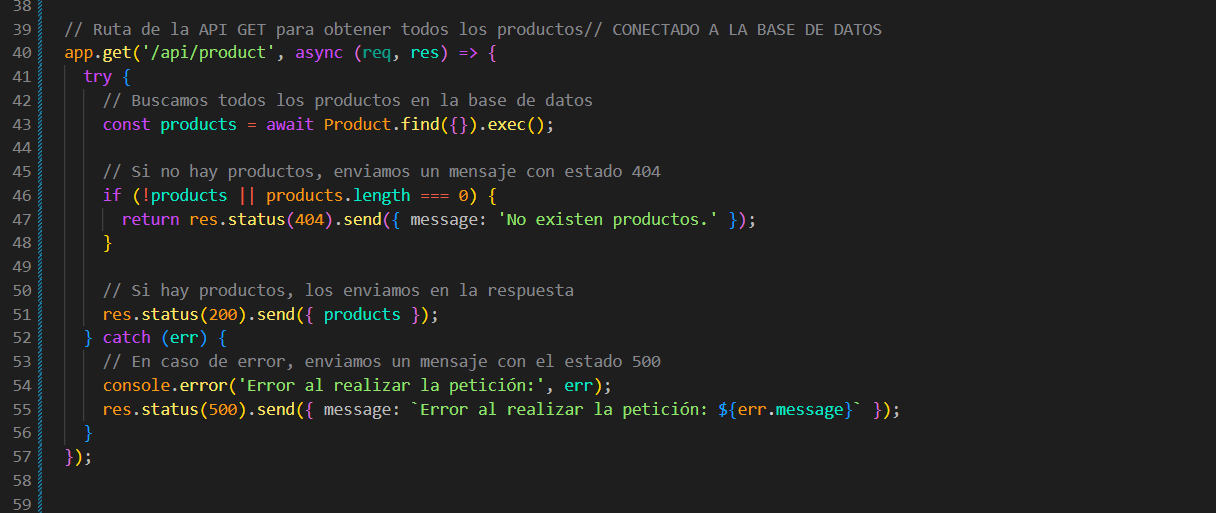
**GET**: Este método se utiliza para solicitar y obtener información de un recurso específico del servidor, como una lista de productos o un producto en particular. Es una operación de solo lectura.

**POST**: El método POST se usa para enviar datos al servidor con el fin de crear un nuevo recurso, como la adición de un nuevo producto en una tienda en línea.

**PUT**: Este método se utiliza para actualizar un recurso existente en el servidor. A diferencia de POST, que crea nuevos recursos, PUT sobrescribe los datos existentes con la información proporcionada.

**DELETE**: Como su nombre lo indica, DELETE se emplea para eliminar un recurso existente en el servidor.  
  
**Aplicación en el código**  
1. GET (Obtener productos)

El método GET se usa en dos rutas diferentes:

1. Una ruta para obtener todos los productos almacenados en la base de datos:  
     
     
     
    **Documentación POSTMAN:** Primero debemos tener presente el servidor en donde está corriendo nuestro programa, el cual es : <http://localhost:3000/> y adicional debemos agregar la api creada la cual es “ api/producto : ( <http://localhost:3000/api/product> ) .   
     
   **Abrimos el POSTMAN USANDO EL METODO GET:**

Captura de pantalla de computadora

Descripción generada automáticamente  
Y luego copiamos la dirección del servidor junto al de la api, adicional le damos send:   
  
Una captura de pantalla de una computadora

Descripción generada automáticamente  
Este nos muestra todos los productos creados, actualmente hay 12 productos los cuales están conectados a la base de datos, si se realiza un cambio en POSTMAN en la base de datos también se refleja.   
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente  
Adicional podemos comprobarlo en el servidor, se debe tener en cuenta que debe estar corriendo el programa del node.js.  
Una vez comprobado copiamos la dirección del servidor junto con la api.  
  
Captura de pantalla de computadora

Descripción generada automáticamente  
En el servidor:   
Interfaz de usuario gráfica, Texto, Correo electrónico

Descripción generada automáticamente  
B. Una segunda ruta para obtener un producto específico por su ID:  
Texto

Descripción generada automáticamente  
  
Seguimos los mismos pasos, solo que debemos adicionar en la url el id único creado por POSTMAN del producto:   
  
Interfaz de usuario gráfica, Texto, Correo electrónico

Descripción generada automáticamente  
  
En POSTMAN:   
  
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente  
 **En el servidor:**   
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente  
Estas rutas permiten recuperar información desde la base de datos de productos, ya sea para listar todos o uno en particular, manejando respuestas con estados HTTP como 200 para éxitos y 404 o 500 en caso de errores.  
  
POST (Crear un nuevo producto) El método POST se utiliza para **crear un nuevo producto** en la base de datos:

Texto

Descripción generada automáticamente  
  
EN POSTMAN: Es de **suma** importancia crear un nuevo producto en POSTMAN usando las siguientes indicaciones:   
1. Se debe tener en cuenta que los 12 productos ya creados hacen referencia a las 12 tarjetas del índex, por ende hay dos datos que no pueden ser iguales al ingresar un nuevo producto :   
  
Logotipo

Descripción generada automáticamente  
  
Tomando como ejemplo el producto # 12 estos dos datos no se pueden repetir para que la creación sea exitosa, ya que el \_id es el numero único dado por POSTMAN en la base de datos , y el id es el numero único de la ubicación de la tarjeta en donde recaerá toda la información, dejando eso en claro se debe ingresar de la siguiente manera:   
  
Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente  
  
Cambiamos a POST y agregamos el url de el servidor y la api.   
  
Captura de pantalla de computadora

Descripción generada automáticamente  
  
Vamos a “**Body”**, seleccionamos **“raw”** y tenemos presente que este en formato **JSON,** una vez este esto preseleccionado vamos a usar el siguiente formado para ingresar un nuevo producto:   
………………………………………………………………………………………………..  
**{**

"id": 12,

"brand": "Motorola",

"price": 799.99,

"storage": "256GB",

"camera": "50MP",

"connect": "5G",

"ram": "12GB",

"screen": "AMOLED",

"warranty": "2years",

"image\_url": "https://i.blogs.es/602692/motorola-edge-30-pro-trasera/450\_1000.jpg",

"title": "Motorola Edge 30 Pro",

"description": "El Motorola Edge 30 Pro ofrece un excelente rendimiento con su procesador Snapdragon 8 Gen 1 y una pantalla OLED de 6.7 pulgadas.",

"\_\_v": 0  
}  
……………………………………………………………………………………………..  
Se debe copiar y pegar tal cual, junto con las llaves, cabe recalcar que la información que esta hace referencia a un producto que ya esta creado, mas sin embargo sirve como base para crear otro ya que indica la información que solicita.   
Una vez claro esto procedemos a agregar la información del producto número # 13 o cualquier otro numero que no sea ninguno del 1 al 12 ya que estos ya están creados.   
  
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente  
  
Y si le damos en send este se va a crear:   
  
Captura de pantalla de computadora

Descripción generada automáticamente  
 **Desde el servidor y la api:**   
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente  
  
**Desde la base de datos:**   
  
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente  
  
Esta operación valida los datos recibidos desde el cuerpo de la solicitud (req.body) y, si todo es correcto, guarda el nuevo producto en la base de datos.  
  
3. PUT (Actualizar un producto)

El método PUT se utiliza para **actualizar un producto existente** por su ID:  
  
Captura de pantalla de computadora

Descripción generada automáticamente  
  
Para actualizar un producto desde POSTMAN haremos el ejemplo con el producto # 13 recién creado :   
  
Lo primero que debemos hacer es buscar el dato **“\_id”** el cual es el dato único dado por el POSTMAN en la base de datos :   
  
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Una vez esto lo copiamos y lo pegamos junto a la url del servidor y la api en POSTMAN   
  
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente  
  
Y cambiamos uno de los datos o los que sean necesarios, **sin** alterar los ID.  
  
Captura de pantalla de computadora

Descripción generada automáticamente  
  
Y le damos en send:   
Captura de pantalla de computadora

Descripción generada automáticamente  
  
**En el servidor sin actualizar:**   
Interfaz de usuario gráfica, Texto

Descripción generada automáticamente  
  
**Actualizado:**   
Texto

Descripción generada automáticamente  
  
Este método toma el ID del producto a actualizar desde los parámetros de la URL (req.params.productId) y actualiza los campos con los datos proporcionados en el cuerpo de la solicitud (req.body).  
  
4. **DELETE (Eliminar un producto)**Por último, el método DELETE se utiliza para **eliminar un producto existente**:  
  
Captura de pantalla de computadora

Descripción generada automáticamente  
  
Para esto es mas sencillo que los anteriores procesos, vamos a buscar el “\_id” único de un producto para poderlo eliminar de la api y de la base de datos, vamos a usar el mismo producto, el número 13.   
  
Cambiamos a DELETE, copiamos el url del servidor con la api y el “ \_id” del producto que se quiere eliminar :   
  
Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente  
**Y damos send.**   
  
Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente  
  
Con estas estructuras, se han aplicado de manera efectiva los métodos de API para realizar operaciones básicas de gestión de productos dentro del proyecto CELLWD.