CRUD API en Postman: Marco Teórico y Ejercicio Práctico

Julian F. Latorre

Índice

1. Introducción

Este documento proporciona un marco teórico sobre CRUD (Create, Read, Update, Delete) API y su implementación en Postman, seguido de un ejercicio práctico utilizando Express.js.

2. Marco Teórico

2.1. ¿Qué es CRUD?

CRUD es un acrónimo que representa las cuatro operaciones básicas que se pueden realizar en la mayoría de las bases de datos y sistemas de almacenamiento:

- **Create (Crear):** Agregar nuevos datos.
- **Read (Leer):** Recuperar o consultar datos existentes.
- **Update (Actualizar):** Modificar datos existentes.
- **Delete (Eliminar):** Borrar datos existentes.

2.2. API RESTful

Una API RESTful es una interfaz que utiliza solicitudes HTTP para acceder y manipular datos. Las operaciones CRUD se mapean a métodos HTTP de la siguiente manera:

- Create: POST
- Read: GET
- Update: PUT o PATCH
- Delete: DELETE

2.3. Postman

Postman es una plataforma de colaboración para el desarrollo de API que permite a los usuarios crear, probar, documentar y compartir API. Algunas características clave de Postman incluyen:

- Interfaz gráfica para enviar solicitudes HTTP.
- Capacidad para organizar solicitudes en colecciones.
- Entornos para manejar diferentes configuraciones.
- Pruebas automatizadas y scripts pre/post-solicitud.
- Generación de documentación de API.
- Monitoreo y reportes.

3. Ejercicio Práctico: CRUD API con Express

3.1. Configuración del Proyecto

Primero, creamos un nuevo proyecto de Node.js e instalamos las dependencias necesarias:

```
mkdir crud-api-express
cd crud-api-express
npm init -y
npm install express body-parser
```

3.2. Creación del Servidor Express

Creamos un archivo `server.js` con el siguiente contenido:

```
const express = require('express');
const bodyParser = require('body-parser');

const app = express();
const PORT = 3000;

app.use(bodyParser.json());

let users = [
  { id: 1, name: 'Juan', email: 'juan@example.com' },
  { id: 2, name: 'María', email: 'maria@example.com' }
];

// Rutas CRUD aquí

app.listen(PORT, () => {
  console.log(`Servidor corriendo en http://localhost:${PORT}`);
});
```

3.3. Implementación de Rutas CRUD

Ahora, implementaremos las rutas CRUD para la entidad üsuario":

```
// CREATE
app.post('/users', (req, res) => {
  const newUser = {
    id: users.length + 1,
    name: req.body.name,
    email: req.body.email
  };
  users.push(newUser);
  res.status(201).json(newUser);
});

// READ (todos los usuarios)
app.get('/users', (req, res) => {
  res.json(users);
});
```

```
// READ (usuario específico)
app.get('/users/:id', (req, res) => {
  const user = users.find(u => u.id === parseInt(req.params.id));
  if (!user) return res.status(404).send('Usuario no encontrado');
  res.json(user);
});

// UPDATE
app.put('/users/:id', (req, res) => {
  const user = users.find(u => u.id === parseInt(req.params.id));
  if (!user) return res.status(404).send('Usuario no encontrado');

  user.name = req.body.name;
  user.email = req.body.email;
  res.json(user);
});

// DELETE
app.delete('/users/:id', (req, res) => {
  const userIndex = users.findIndex(u => u.id === parseInt(req.
    params.id));
  if (userIndex === -1) return res.status(404).send('Usuario no
    encontrado');

  users.splice(userIndex, 1);
  res.status(204).send();
});
```

3.4. Pruebas con Postman

Para probar nuestra API, seguiremos estos pasos en Postman:

3.4.1. Configuración del Entorno

1. Crear un nuevo entorno llamado ÇRUD API Local". 2. Añadir una variable `baseUrl` con el valor `http://localhost:3000`.

3.4.2. Creación de una Colección

1. Crear una nueva colección llamada ÇRUD API Users". 2. Dentro de esta colección, crear una carpeta para cada operación CRUD.

3.4.3. Configuración de Solicitudes

CREATE (POST)

- Método: POST
- URL: {{baseUrl}}/users
- Headers: Content-Type: application/json
- Body (raw JSON):

```
{
  "name": "Ana",
  "email": "ana@example.com"
}
```

READ (GET)

- Método: GET
- URL: {{baseUrl}}/users

READ Específico (GET)

- Método: GET
- URL: {{baseUrl}}/users/1

UPDATE (PUT)

- Método: PUT
- URL: {{baseUrl}}/users/1
- Headers: Content-Type: application/json
- Body (raw JSON):

```
{
  "name": "Juan Actualizado",
  "email": "juan.actualizado@example.com"
}
```

DELETE

- Método: DELETE
- URL: {{baseUrl}}/users/2

3.5. Ejecución de Pruebas

1. Iniciar el servidor Express ejecutando `node server.js`. 2. En Postman, seleccionar el entorno ÇRUD API Local". 3. Ejecutar cada solicitud en el orden: CREATE, READ, UPDATE, DELETE. 4. Verificar las respuestas y el estado de los datos después de cada operación.