



# “SQLite POSTMAN”

**:: UNIDAD 3: Fundamentos de programación de dispositivos móviles**

**:: ACTIVIDAD COMPLEMENTARIA 1**

**MATERIA: :: PROGRAMACIÓN DE DISPOSITIVOS MÓVILES ::**

**CLAVE: 1668**

**LICENCIATURA EN INFORMÁTICA**

**PLAN 2012**

**REALIZO: Emmanuel Alejandro Pérez Hernández**

**No. 423142118**

**Grupo: 8691**

**ASESOR: MARTINEZ FERNANDEZ JUAN MANUEL**

**domingo, 2 de noviembre de 2025**

## **UNIDAD 3. Actividad Complementaria 1**

### **a) Investiga qué es SQLite y sus ventajas.**

SQLite es un sistema de gestión de bases de datos relacional (RDBMS) que se distingue por ser:

- Autocontenido: No requiere un proceso servidor separado
- Sin configuración: No necesita instalación o configuración compleja
- Basado en archivos: Toda la base de datos se almacena en un único archivo
- Ligero: La biblioteca completa ocupa menos de 500KB
- Sin dependencias externas: Funciona independientemente

Ventajas de SQLite:

1. Portabilidad: El archivo de base de datos puede moverse entre sistemas
2. Bajo consumo de recursos: Ideal para dispositivos embebidos y móviles
3. Cero configuración: No requiere administración
4. Transacciones ACID: Cumple con las propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad)
5. Compatibilidad multiplataforma: Funciona en Windows, Linux, macOS, iOS, Android, etc.
6. Lenguaje SQL completo: Soporta la mayoría del estándar SQL
7. Licencia de dominio público: Libre de regalías

Casos de uso típicos:

- Aplicaciones móviles
- Dispositivos embebidos/IoT
- Prototipado rápido
- Almacenamiento local en navegadores
- Aplicaciones de escritorio
- Caché de datos



**b) Investiga qué es POSTMAN.**

Postman es una plataforma de colaboración para el desarrollo de APIs que proporciona:

Funcionalidades principales:

1. Cliente HTTP/REST: Interfaz gráfica para enviar solicitudes a APIs
2. Pruebas de APIs: Permite probar endpoints y verificar respuestas
3. Colecciones: Organizar y agrupar solicitudes relacionadas
4. Entornos: Configurar diferentes ambientes (desarrollo, producción, testing)
5. Automatización: Ejecutar pruebas automatizadas
6. Documentación: Generar documentación automática de APIs
7. Monitoreo: Supervisar el rendimiento de APIs
8. Colaboración: Compartir colecciones con equipos de trabajo

Características técnicas:

- Soporte múltiples métodos HTTP: GET, POST, PUT, DELETE, PATCH, etc.
- Gestión de headers y parámetros
- Scripts de pre-request y tests (JavaScript)
- Variables y entornos dinámicos
- Importar/exportar colecciones
- Integración con CI/CD
- Soporte para GraphQL, WebSockets, gRPC

Ventajas de Postman:

1. Interfaz intuitiva: Fácil de usar sin necesidad de escribir código
2. Colaboración en equipo: Compartir y trabajar conjuntamente
3. Automatización: Pruebas regresivas automáticas
4. Documentación viva: Siempre actualizada



5. Ecosistema completo: Todo el ciclo de vida de APIs
6. Comunidad activa: Miles de colecciones públicas disponibles

Casos de uso:

- Desarrollo y testing de APIs
- Documentación técnica
- Automatización de pruebas
- Monitoreo de servicios
- Debugging de integraciones
- Enseñanza de conceptos de APIs



**c) Crea un endpoint "agrega\_todo" que reciba datos con POST.**

```
// Endpoint para ver todos
app.get('/todos', (req, res) => {
  db.all('SELECT * FROM todos', (err, rows) => {
    if (err) {
      return res.status(500).json({ error: err.message });
    }
    res.json(rows);
  });
});
```

Endpoint POST agrega\_todo

**d) Crea una tabla "todos" en SQLite (id, todo, created\_at).**

```
// Crear tabla todos
db.run(`
  CREATE TABLE IF NOT EXISTS todos (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    todo TEXT NOT NULL,
    created_at DATETIME DEFAULT CURRENT_TIMESTAMP
)
`, (err) => {
  if (err) {
    console.error(`Óòi Error creando tabla: ${err}`);
  } else {
    console.log(`Óéà Tabla "todos" lista`);
  }
});
```

Tabla SQLite todos

**e) Modifica el endpoint para guardar los datos en la tabla.**

```
// f) GUARDAR EN LA BASE DE DATOS
db.run('INSERT INTO todos (todo) VALUES (?)', [todo], function(err) {
  if (err) {
    return res.status(500).json({
      error: 'Error del servidor: ' + err.message
    });
  }
  console.log(`Óéà TODO guardado ID: ${this.lastID}`);

  // h) RESPUESTA CON ESTADO 201
  res.status(201).json({
    success: true,
    id: this.lastID,
    todo: todo,
    message: 'TODO guardado exitosamente'
  });
});
```

Guardar datos en tabla

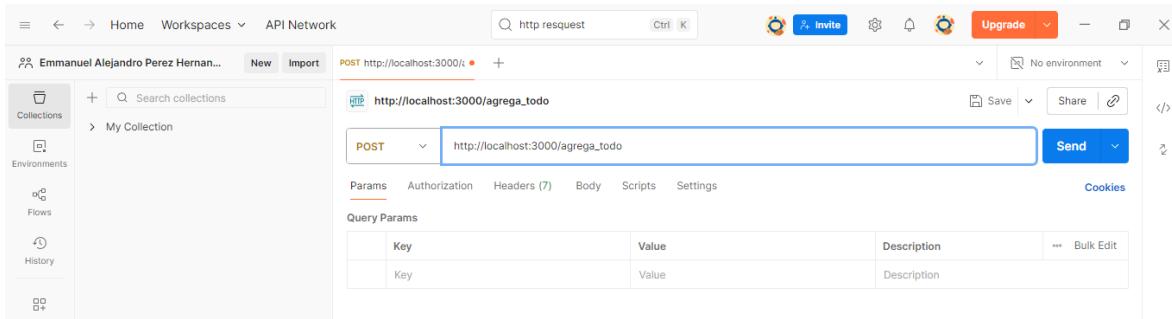
```
// Endpoint para ver todos
app.get('/todos', (req, res) => {
  db.all('SELECT * FROM todos', (err, rows) => {
    if (err) {
      return res.status(500).json({ error: err.message });
    }
    res.json(rows);
  });
});

// Endpoint de prueba
app.get('/', (req, res) => {
  res.json({
    message: '-fÜC API de TODOS funcionando!',
    endpoints: [
      'POST /agrega_todo': 'Agregar nuevo TODO',
      'GET /todos': 'Ver todos los TODOS'
    ]
  });
});
```

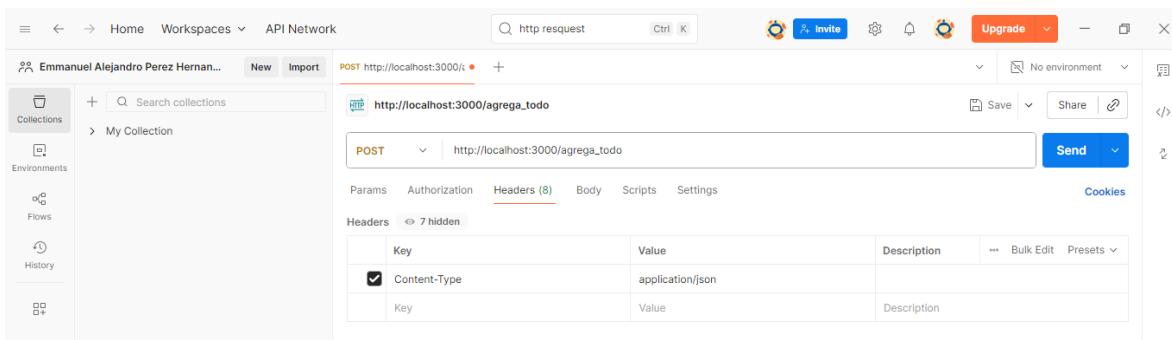
Modificar endpoint para guardar



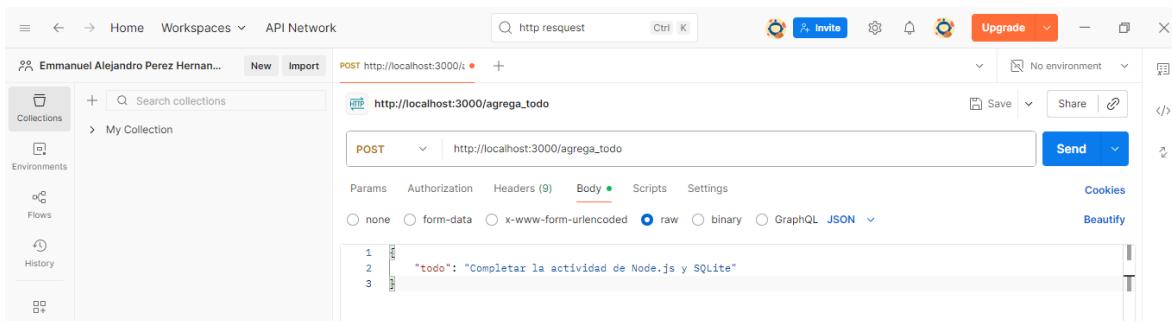
f) Usa POSTMAN para enviar una petición y verificar que los datos se guarden.



### POSTMAN - Método POST y URL configurados



### POSTMAN - Headers con Content-Type



### POSTMAN - Body con JSON



POST http://localhost:3000/agrega\_todo

```

1 {
2   "todo": "Completar la actividad de Node.js y SQLite"
3 }

```

Body Cookies Headers (8) Test Results

201 Created 206 ms 387 B

```

1 {
2   "success": true,
3   "id": 1,
4   "todo": "Completar la actividad de Node.js y SQLite",
5   "message": "TODO guardado exitosamente"
6 }

```

## POSTMAN - Respuesta 201 exitosa

GET http://localhost:3000/todos

Body Cookies Headers (8) Test Results

200 OK 21 ms 364 B

```

1 [
2   {
3     "id": 1,
4     "todo": "Completar la actividad de Node.js y SQLite",
5     "created_at": "2025-11-02 22:38:56"
6   }
7 ]

```

## POSTMAN - GET todos mostrando datos guardados

```

C:\Windows\system32\cmd.exe - node server.js
    console.log('-----');
});
C:\Users\Emma\Desktop\ proyecto-todos>node server.js
-----
[{"id": 1, "todo": "Completar la actividad de Node.js y SQLite", "created_at": "2025-11-02 22:38:56"}]
-----
```

## CMD - Terminal con logs del servidor



**g) Devuelve un JSON con estado HTTP 201.**

```
// h) RESPUESTA CON ESTADO 201
res.status(201).json({
  success: true,
  id: this.lastID,
  todo: todo,
  message: 'TODO guardado exitosamente'
});
```

RESPUESTA CON ESTADO 201



## I. REFERENCIAS BIBLIOGRÁFICAS

- sqlite3 — DB-API 2.0 interfaz para bases de datos SQLite — documentación de Python - 3.8.20.* (2024). Python.org.  
<https://docs.python.org/es/3.8/library/sqlite3.html#:~:text=SQLite%20es%20una%20biblioteca%20de,2.0%20descrita%20por%20PEP%20249.&text=Tambi%C3%A9n%20se%20puede%20agregar%20el,de%20datos%20en%20memoria%20RAM.&text=Para%20obtener%20los%20datos%20luego,lista%20de%20todos%20los%20registro>  
s.
- SQLite: La Base de Datos Embebida.* (2025). SG Buzz. <https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida>
- Javier. (2023, August 24). *¿Qué es Postman? ¿Cuáles son sus principales ventajas?* - *Formadores IT.* Formadores IT. <https://formadoresit.es/que-es-postman-cuales-son-sus-principales-ventajas/#:~:text=Fomenta%20y%20facilita%20la%20colaboraci%C3%B3n,de%20trabajo%20en%20diferentes%20proyectos>
- (2025). Youtube.com. <https://www.youtube.com/watch?v=jAe-Jv4FDFM>

