APLICACIÓN WEB CON NODEJS

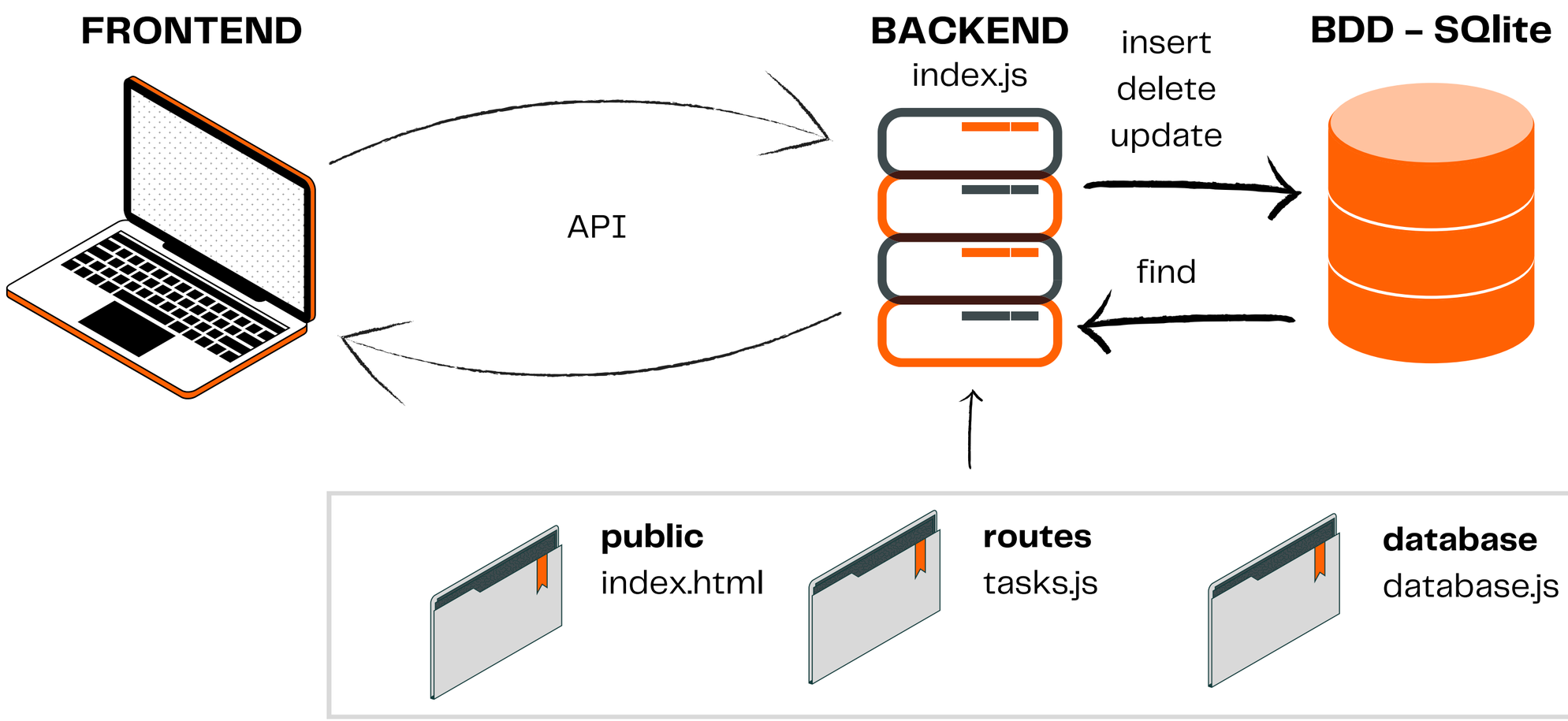
SOMOSCODENAUTAS.COM

CODENAUTAS

LA ESCUELA TECNOLÓGICA DE AUCA

ESTRUCTURA DE LA APP

CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA



PASOS EN LA CREACIÓN DEL PROYECTO

CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA

- En el directorio donde vamos a crear el proyecto ejecutamos el comando **npm init –y**.
- Vamos a ver como se genera el archivo de configuración del proyecto: **package.json**.
- Instalamos en el directorio del proyecto el paquete de Node.js **express** para crear el servidor local, con **npm install express**. Tras esta instalación veremos como en el archivo de configuración del proyecto se crea la dependencia de express node.
- Instalamos los paquetes **nodemon** y **body-parser**. Nodemon se utiliza para poder actualizar la ejecución de nuestra aplicación cada vez que realicemos un cambio, mientras que body-parser facilita la comunicación en formato JSON entre el backend y el frontend.

npm install nodemon –D
npm install body-parser

ORGANIZACIÓN DE LOS ARCHIVOS

CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA

```
task-manager/
├── database/
│   └── database.db             # Archivo SQLite que contiene la base de datos
│   └── database.js             # Archivo con la lógica de conexión a la BDD
├── routes/
│   └── tasks.js                # Archivo con las rutas para la API REST
├── public/
│   ├── index.html              # Archivo principal del frontend (HTML y JS)
│   └── styles.css              # Archivo de estilos CSS (opcional, si se extraen los estilos del HTML)
├── node_modules/               # Dependencias instaladas con npm
├── .gitignore                  # Archivos y carpetas a ignorar por Git
├── index.js                    # Archivo principal del servidor Node.js
├── package.json                # Configuración del proyecto y dependencias
└── package-lock.json           # Detalles de las dependencias instaladas
```

CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA

Base de datos para nuestra aplicación

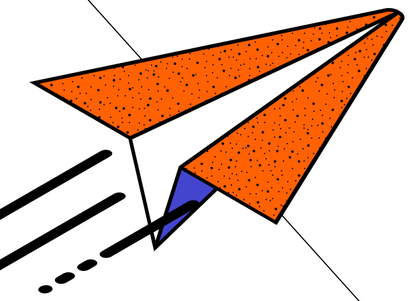
SQLite es una biblioteca en lenguaje C que implementa un motor de base de datos SQL pequeño, rápido, autónomo, de alta confiabilidad y con todas las funciones.

SQL es uno de los lenguajes de base de datos más utilizado del mundo.

SQLite está integrado en todos los teléfonos móviles y en la mayoría de los ordenadores y viene incluido en muchas otras aplicaciones que utilizamos todos los días.

Instalación

npm install sqlite3



CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA

ARCHIVO PACKAGE.JSON

Nuestro archivo de configuración de proyecto debe quedar así:

Añadimos la característica **“type”: “module”** para la importación de módulos y dependencias en la versión moderna de javascript ECMAScript Modules.

Añadimos el script **“start”: “nodemon index.js”** para ejecutar la aplicación con la dependencia nodemon y utilizando el comando **npm start.**

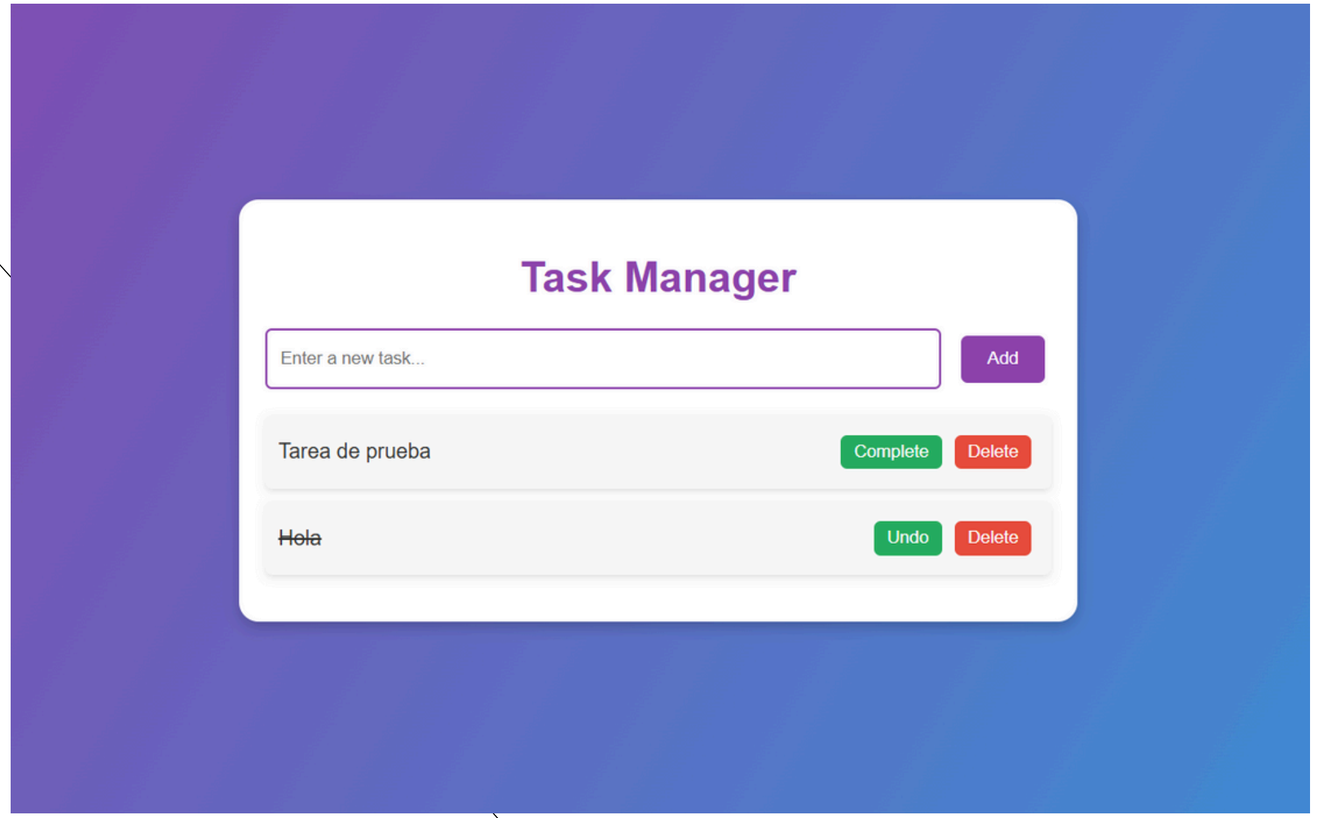
```
{
  "name": "tasks-manager",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "type": "module",
  ▷ Debug
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "nodemon index.js"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "body-parser": "^1.20.3",
    "express": "^4.21.2",
    "sqlite3": "^5.1.7"
  },
  "devDependencies": {
    "nodemon": "^3.1.9"
  }
}
```

CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA

CÓDIGO DE LA APP

APP PARA REGISTRAR TAREAS

CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA



INDEX.JS

CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA

```
import express from 'express';
import path from 'path';
import { fileURLToPath } from 'url';
import tasksRouter from './routes/tasks.js';

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

// Obtener la ruta del directorio actual
const __filename = fileURLToPath(import.meta.url);
const __dirname = path.dirname(__filename);

// Middleware para parsear JSON
app.use(express.json());

// Servir archivos estáticos (index.html y otros desde la carpeta "public")
app.use(express.static(path.join(__dirname, 'public')));

// Rutas de la API
app.use('/api/tasks', tasksRouter);

// Ruta principal para servir la vista
Tabnine | Edit | Test | Explain | Document | Ask
app.get('/', (req, res) => {
  res.sendFile(path.join(__dirname, 'public', 'index.html'));
});

// Iniciar el servidor
Tabnine | Edit | Test | Explain | Document | Ask
app.listen(port, () => {
  console.log(`Servidor corriendo en http://localhost:${port}`);
});
```

DATABASE.JS

CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA

```
import sqlite3 from 'sqlite3';

const db = new sqlite3.Database('./database/database.db', (err) => {
  if (err) {
    console.error('Error al conectar con la base de datos:', err.message);
  } else {
    console.log('Conectado a la base de datos SQLite.');
    db.run(`
      CREATE TABLE IF NOT EXISTS tasks (
        id INTEGER PRIMARY KEY AUTOINCREMENT,
        title TEXT NOT NULL,
        completed INTEGER DEFAULT 0
      )
    `, (err) => {
      if (err) {
        console.error('Error al crear la tabla:', err.message);
      } else {
        console.log('Tabla `tasks` verificada/existente.');
      }
    });
  }
});

export default db;
```

ROUTES.JS

CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA

```
import express from 'express';
import db from '../database/database.js';

const router = express.Router();

// Obtener todas las tareas
Tabnine | Edit | Test | Explain | Document | Ask
router.get('/', (req, res) => {
  db.all('SELECT * FROM tasks', [], (err, rows) => {
    if (err) {
      res.status(500).json({ error: err.message });
      return;
    }
    res.json(rows);
  });
});

// Crear una nueva tarea
Tabnine | Edit | Test | Explain | Document | Ask
router.post('/', (req, res) => {
  const { title, description } = req.body;
  db.run(
    'INSERT INTO tasks (title) VALUES (?)',
    [title, description],
    function (err) {
      if (err) {
        res.status(500).json({ error: err.message });
        return;
      }
      res.status(201).json({ id: this.lastID, title });
    }
  );
});
```

```
// Actualizar una tarea
Tabnine | Edit | Test | Explain | Document | Ask
router.put('/:id', (req, res) => {
  const { completed } = req.body; // Asegúrate de que completed es un boolean
  const id = req.params.id;

  console.log(`Actualizar tarea ID: ${id}, Estado completado: ${completed}`);

  if (typeof completed !== 'boolean') {
    res.status(400).json({ error: 'El campo completed debe ser booleano' });
    return;
  }

  db.run(
    'UPDATE tasks SET completed = ? WHERE id = ?',
    [completed ? 1 : 0, id], // Convertimos booleano a entero para SQLite
    function (err) {
      if (err) {
        res.status(500).json({ error: err.message });
        return;
      }
      if (this.changes === 0) {
        res.status(404).json({ error: 'Tarea no encontrada' });
        return;
      }
      res.json({ id, completed });
    }
  );
});
```

ROUTES.JS

CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA

```
// Eliminar una tarea
Tabnine | Edit | Test | Explain | Document | Ask
router.delete('/:id', (req, res) => {
  const id = req.params.id;

  db.run('DELETE FROM tasks WHERE id = ?', [id], function (err) {
    if (err) {
      res.status(500).json({ error: err.message });
      return;
    }
    if (this.changes === 0) {
      res.status(404).json({ error: 'Tarea no encontrada' });
      return;
    }
    res.status(204).send();
  });
});

export default router;
```

INDEX.HTML

CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Task Manager</title>
    <style>
        body {
            margin: 0;
            font-family: 'Arial', sans-serif;
            background: linear-gradient(120deg, #8e44ad, #3498db);
            color: white;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }

        .container {
            background: white;
            color: #333;
            width: 90%;
            max-width: 600px;
            padding: 20px;
            border-radius: 15px;
            box-shadow: 0 4px 10px rgba(0, 0, 0, 0.2);
        }

        h1 {
            text-align: center;
            color: #8e44ad;
        }

        form {
            display: flex;
            gap: 10px;
            margin-bottom: 20px;
        }

        input[type="text"] {
            flex: 1;
            padding: 10px;
            border: 2px solid #8e44ad;
            border-radius: 5px;
        }

        button {
            background: #8e44ad;
            color: white;
            border: none;
            padding: 10px 20px;
            margin: 5px;
            border-radius: 5px;
            cursor: pointer;
            transition: background 0.3s ease;
        }

        button:hover {
            background: #732d91;
        }

        .tasks {
            list-style: none;
            padding: 0;
        }

        .tasks li {
            display: flex;
            justify-content: space-between;
            align-items: center;
            background: #f9f9f9;
            margin-bottom: 10px;
            padding: 10px;
            border-radius: 5px;
            box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
        }

        .tasks li span {
            flex: 1;
        }

        .tasks li button {
            background: #e74c3c;
            padding: 5px 10px;
        }

        .tasks li button.complete {
            background: #27ae60;
        }

        .tasks li button:hover {
            opacity: 0.8;
        }
    </style>
</head>
```

INDEX.HTML

CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA

```
<body>
    <div class="container">
        <h1>Task Manager</h1>
        <form id="taskForm">
            <input type="text" id="taskTitle" placeholder="Enter a new task..." required>
            <button type="submit">Add</button>
        </form>
        <ul id="taskList" class="tasks"></ul>
    </div>

    <script>
        const taskForm = document.getElementById('taskForm');
        const taskTitle = document.getElementById('taskTitle');
        const taskList = document.getElementById('taskList');

        const API_URL = '/api/tasks';

        async function fetchTasks() {
            const response = await fetch(API_URL);
            const tasks = await response.json();
            renderTasks(tasks);
        }

        function renderTasks(tasks) {
            taskList.innerHTML = '';
            tasks.forEach(task => {
                const li = document.createElement('li');
                li.innerHTML = `
                    <span>${task.completed ? `<s>${task.title}</s>` : task.title}</span>
                    <button class="complete" onclick="toggleComplete(${task.id}, ${task.completed})">${task.completed ? 'Undo' : 'Complete'}</button>
                    <button onclick="deleteTask(${task.id})">Delete</button>
                `;
                taskList.appendChild(li);
            });
        }
```

INDEX.HTML

CODENAUTAS
LA ESCUELA TECNOLÓGICA DE AUCA

```
        taskForm.addEventListener('submit', async (e) => {
            e.preventDefault();
            const title = taskTitle.value;
            if (!title) return;

            await fetch(API_URL, {
                method: 'POST',
                headers: { 'Content-Type': 'application/json' },
                body: JSON.stringify({ title })
            });

            taskTitle.value = '';
            fetchTasks();
        });

        async function toggleComplete(id, completed) {
            console.log(`Toggling task ${id} to ${!completed}`); // Logging para debugging
            await fetch(`${API_URL}/${id}`, {
                method: 'PUT',
                headers: { 'Content-Type': 'application/json' },
                body: JSON.stringify({ completed: !completed }) // Enviar booleano correcto
            });
            fetchTasks(); // Refresca la lista
        }

        async function deleteTask(id) {
            await fetch(`${API_URL}/${id}`, { method: 'DELETE' });
            fetchTasks();
        }
        fetchTasks();
    </script>
</body>
</html>
```