### **Guia de desplegament d’una aplicació web en node.js, SQLlite i express**

### **Pas 1: Configuració del Projecte**

Podeu utilitzar windows o linux per desenvolupar la pràctica.

A continuació us passo un link on us explica per a que serveix node.js (serà el nostre servirdor local lleuger per la pagina), que és i com s’instal·la a windows, linux i MAC.

<https://kinsta.com/es/blog/como-instalar-node-js/>

Assegura't de tenir Node.js i npm instal·lats, amb el comandament version tal com diu la guia anteriorment linkada. Després, crea una nova carpeta pel teu projecte i executa npm init per inicialitzar-lo. Instal·la les dependències necessàries:

**npm install express sqlite3**

### **Pas 2: Estructura de Carpetes**

Organitza el teu projecte de la següent manera:

Projecte

|-- public

| |-- styles.css

|-- views

| |-- index.html

|-- app.js

|-- database.db

|-- package.json

### **Pas 3: Arxius de Codi**

#### app.js (Servidor Node.js)

const express = require('express');

const sqlite3 = require('sqlite3');

const path = require('path');

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

const app = express();

const port = 3000;

// Configuració de SQLite

const dbPath = path.resolve(\_\_dirname, 'database.db');

const db = new sqlite3.Database(dbPath);

// Configuració d'express

app.use(express.static('public'));

app.use(bodyParser.urlencoded({ extended: true }));

// Rutes

app.get('/', (req, res) => {

res.sendFile(path.join(\_\_dirname, '/views/index.html'));

});

app.post('/addTask', (req, res) => {

const task = req.body.task;

// Inserir la tasca a la base de dades

db.run('INSERT INTO tasks (task) VALUES (?)', [task], (err) => {

if (err) {

return console.error(err.message);

}

console.log(`Tasca afegida: ${task}`);

res.redirect('/');

});

});

// Inicialització del servidor

app.listen(port, () => {

console.log(`Servidor en http://localhost:${port}`);

});

#### views/index.html (Formulari)

html

Copy code

<!DOCTYPE html>

<html lang="ca">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<link rel="stylesheet" href="/styles.css">

<title>Formulari de Tasques</title>

</head>

<body>

<h1>Formulari de Tasques</h1>

<form action="/addTask" method="POST">

<label for="task">Tasca:</label>

<input type="text" id="task" name="task" required>

<button type="submit">Afegir Tasca</button>

</form>

</body>

</html>

**(nota podeu canviar l’afegit de tasques a la base de dades per qualsevol item afegible en una base de dades, podeu ser originals :))**

#### public/styles.css (Estils, opcional)

body {

font-family: Arial, sans-serif;

text-align: center;

margin: 50px;

}

form {

display: flex;

flex-direction: column;

max-width: 300px;

margin: 0 auto;

}

label {

margin-bottom: 10px;

}

input {

padding: 8px;

margin-bottom: 20px;

}

button {

padding: 10px;

background-color: #4CAF50;

color: white;

cursor: pointer;

}

### **Pas 4: Base de Dades SQLite**

Instal·leu sqlite al vostre equip per a poder utilitzar el terminal de comandes.

Crea la base de dades SQLite executant la següent comanda a la terminal:

sqlite3 database.db

Després, dins la consola SQLite, executa la següent consulta per crear la taula tasks:

CREATE TABLE tasks (

id INTEGER PRIMARY KEY AUTOINCREMENT,

task TEXT NOT NULL

);

### **Pas 5: Execució**

Executa la teva aplicació amb:

node app.js

Visita http://localhost:3000 al teu navegador i hauries de veure el formulari. En afegir tasques, aquestes es guardaran a la base de dades SQLite.

Aquest és un exemple senzill que pots ampliar segons les teves necessitats. Recorda implementar mesures de seguretat, com la prevenció d'injeccions SQL, en aplicacions del món real.