



Documentació Tècnica del Sistema de Gestió d'Incidències

Introducció

El present document té com a finalitat oferir una visió completa i detallada sobre el funcionament, la configuració i el manteniment d'un sistema web per a la **gestió d'incidències informàtiques**. Desenvolupat utilitzant tecnologies modernes com **Node.js**, **Express.js**, **MongoDB** i **EJS**, aquest projecte està pensat per ser desplegat en entorns educatius o organitzatius on sigui necessari **registrar, controlar i fer seguiment de problemes tècnics** de manera eficient.

El sistema ha estat dissenyat sota principis de modularitat i escalabilitat, amb una arquitectura clara i lògica que facilita tant el desenvolupament com la seva posterior extensió.

1. Arquitectura del Sistema

L'aplicació segueix l'arquitectura **MVC** (Model-Vista-Controlador), una estructura que permet separar les diferents capes del sistema per facilitar el manteniment, la reutilització del codi i la claredat del desenvolupament.

Components principals:

- **Client Web:** Qualsevol navegador modern pot accedir a la interfície gràfica (vistes renderitzades amb EJS).
- **Servidor Node.js (Express):** Gestiona les rutes, controla la lògica del sistema i connecta amb la base de dades.
- **MongoDB:** Emmagatzema la informació persistent del sistema com incidències, tècnics, departaments i logs d'activitat.

Exemple de Flux de Treball

1. Un usuari accedeix a l'URL `http://localhost:3000/incidencies/new`.
2. El servidor Express renderitza la vista EJS corresponent: `views/incidencies/new.ejs`.

3. En enviar el formulari, es fa una petició POST que és capturada per la ruta `/incidencies`.
4. Aquesta ruta crida el model `Incidencia.js` per tal d'inserir la informació a la base de dades MongoDB.

Aquest esquema garanteix una experiència fluida i permet una gestió eficaç del cicle de vida de cada incidència.

2. Requisits del Sistema

Per tal de garantir una execució òptima, el sistema necessita els següents components:

Requisits de Programari

Component	Versió Recomanada
Node.js	≥ 18.x
NPM	≥ 9.x
MongoDB	≥ 4.4
Sistema Operatiu	Linux, macOS o Windows
Navegador	Chrome, Firefox, Edge

Requisits Funcionals del Sistema

- Possibilitat de **crear, modificar i eliminar** incidències.
- Visualització d'un **registre de logs** amb detalls d'accés.
- Gestió d'usuaris amb rols: **administradors i tècnics**.
- Disponibilitat d'un **panell d'administració** per a una visió global.

3. Guia d'Instal·lació i Configuració

3.1 Instal·lació de Node.js

Pots descarregar l'instal·lador oficial des de:

<https://nodejs.org>

Selecciona la **versió LTS** per assegurar estabilitat. Un cop instal·lat, verifica que funciona correctament obrint el terminal i escrivint:

```
node -v  
npm -v
```

3.2 Descarregar el Projecte

Per obtenir una còpia del projecte:

```
git clone https://github.com/el-teu-usuari/gestio-incidencies.git  
cd gestio-incidencies
```

Instal·la totes les dependències definides al package.json:

```
npm install
```

3.3 Configuració d'Entorn

Per gestionar configuracions sensibles (com la connexió a la base de dades o el port del servidor), utilitzem un fitxer `.env`. Crea'l a l'arrel del projecte amb aquest contingut:

```
PORT=3000  
MONGODB_URI=mongodb://localhost:27017/incidenciesDB
```

Si utilitzes MongoDB Atlas, substitueix la URI per:

```
MONGODB_URI=mongodb+srv://usuari:contrasenya@nomdelcluster.mongodb.net/incidenciesDB
```

3.4 Connexió amb MongoDB

Assegura't que MongoDB estigui en funcionament. Si treballes en local, pots iniciar-lo amb:

```
mongod
```

Si prefereixes una base de dades remota (Atlas), assegura't de tenir accés i permisos suficients, i de configurar correctament l'IP al panell de MongoDB Atlas.

3.5 Execució del Projecte

Per iniciar el projecte en mode desenvolupament (amb autoreload):

npm run dev

En entorns de producció, es recomana:

npm start

L'aplicació estarà disponible a <http://localhost:3000>.

4. Organització del Codi

El projecte està organitzat de manera clara per facilitar el desenvolupament col·laboratiu:

```
src/
├── app.js           # Punt d'entrada principal de l'app
├── db.js           # Connexió a MongoDB
├── models/         # Models de dades amb Mongoose
│   ├── Incidencia.js
│   ├── Departament.js
│   ├── Tecnic.js
│   └── Log.js
├── routes/         # Rutes Express
│   ├── incidentsEJS.routes.js
│   ├── adminEJS.routes.js
│   └── logs.js
├── views/          # Vistes amb EJS
│   ├── admin/
│   │   └── logs.ejs
│   ├── incidencies/
│   │   ├── list.ejs
│   │   └── new.ejs
│   └── partials/
│       ├── header.ejs
│       └── footer.ejs
```

5. Manteniment del Sistema

5.1 Còpies de Seguretat

És essencial fer backups regulars de la base de dades. Un exemple de còpia manual seria:

```
mongodump --db incidenciesDB --out ./backups/incidencies-$(date +%F)
```

Per restaurar:

```
mongorestore --db incidenciesDB ./backups/incidencies-YYYY-MM-DD/incidenciesDB
```

Es recomana establir un **cron job** per a realitzar aquestes còpies de manera automatitzada.

5.2 Monitoratge i Logs

El sistema incorpora un middleware personalitzat `logMiddleware.js` que registra:

- Usuari (si n'hi ha)
- Ruta accedida
- Tipus de navegador
- Data i hora de l'accés

Aquest registre es desa a la col·lecció `logs` i pot ser consultat des del panell d'administració:

<http://localhost:3000/admin/logs>

6. Resolució de Problemes Comuns

Problema	Solució
Error: Failed to lookup view	Comprova que estàs utilitzant rutes relatives a <code>res.render()</code> . No incloguis <code>/</code> inicial.
MongoDB no es connecta	Verifica que <code>mongod</code> està en funcionament i que la URI és correcta.
Canvis a vistes no es veuen	Reinicia el servidor (<code>Ctrl + C</code> i <code>npm run dev</code>) o força el navegador a actualitzar (<code>Ctrl + Shift + R</code>).
Error de permisos a Linux	Comprova els permisos d'execució o executa amb <code>sudo</code> .
Els logs no es guarden	Assegura't que el middleware està registrat abans de definir les rutes.

Conclusió

Aquest document representa la **pedra angular** del projecte. Permet tant a nous membres de l'equip com a administradors externs **entendre i desplegar l'aplicació** sense dificultats. Gràcies a la seva arquitectura clara, el sistema és fàcilment escalable i adaptat a entorns educatius o corporatius on la gestió d'incidències és clau.

Amb la documentació adequada, les bones pràctiques de desenvolupament i l'ús de metodologies àgils com SCRUM, aquest projecte esdevé un exemple sòlid de com gestionar problemes tècnics de forma eficient i col·laborativa.