

# Documentació JavaScript

Hugo Córdoba, Nil Díaz

---



# ÍNDEX

<b>JAVASCRIPT:</b>	<b>3</b>
HEADER.EJS	3
1. Comprovació inicial:	3
2. Funció showPasswordPrompt:	4
3. Funció checkPassword:	4
LIST.EJS	5
1. Funció disableHeaderLinks:	5
2. Funció resetAccess:	6
DB.JS	6
1. Importacions:	7
2. Creació de la connexió:	7
3. Exportació:	7
APP.JS	10
Dependencies:	11
Models:	11
Relacions:	11
Rutes:	11
Servidor:	11
Chatbot:	11
<b>Bootstrap:</b>	<b>12</b>
Introduccio:	12
index.ejs:	12

# JAVASCRIPT:

## HEADER.EJS

En aquest script hem desenvolupat un sistema senzill de protecció amb contrasenya per controlar l'accés a una secció concreta de la nostra web.

Vam analitzar que no és gaire correcte que un usuari qualsevol pugui accedir a l'apartat de gestió d'incidències i vam pensar aquesta solució perquè no pogués accedir ningú en aquest apartat sense saber unes credencials específiques.

```
<script>
  const correctPassword = "12345";

  // Verificar si ya se ha validado La contraseña
  if (sessionStorage.getItem("accessGranted") === "true") {
    document.getElementById("incidencies-link").style.pointerEvents = "none";
    document.getElementById("projecte-link").style.pointerEvents = "none"; // Desactivar el enlace
  }

  function showPasswordPrompt() {
    document.getElementById("password-prompt").style.display = "block";
    document.getElementById("main-content").style.display = "none";
    document.getElementById("incidencies-link").style.pointerEvents = "none";
    document.getElementById("projecte-link").style.pointerEvents = "none"; // Desactivar el enlace
  }

  function checkPassword() {
    const enteredPassword = document.getElementById("password").value;
    if (enteredPassword === correctPassword) {
      sessionStorage.setItem("accessGranted", "true"); // Guardar en sessionStorage que el acceso fue concedido
      window.location.href = "/incidencies";
    } else {
      document.getElementById("error-message").style.display = "block";
    }
  }
</script>
```

### 1. Comprovació inicial:

Definim la contrasenya correcta amb la constant *correctPassword*, que en aquest cas és "12345". A l'inici del codi, comprovem si l'usuari ja ha passat la validació de la contrasenya. Ho fem mirant si al *sessionStorage* hi ha guardat "accessGranted" amb el valor "true".

Si la validació ja s'ha fet, immediatament desactivem els enllaços amb els id *"incidencies-link"* i *"projecte-link"*, mitjançant la propietat CSS *pointerEvents = "none"*. Això evita que s'hi pugui clicar novament.

```
const correctPassword = "12345";

// Verificar si ya se ha validado la contraseña
if (sessionStorage.getItem("accessGranted") === "true") {
  document.getElementById("incidencies-link").style.pointerEvents = "none";
  document.getElementById("projecte-link").style.pointerEvents = "none"; // Desactivar el
  enlace
}
```

## 2. Funció showPasswordPrompt:

En aquest apartat del codi seria la crida per a mostrar el quadre per a introduir les credencials necessàries per a accedir als apartats ocults.

Primer de tot el que volem que faci és mostrar l'element amb id *"password-prompt"*. Després volem que el contingut principal del *main* que teníem anteriorment quedi amagat i això ho aconseguim realitzant el *"none"* del *main-content*.

També desactivem els enllaços *"incidencies-link"* i *"projecte-link"* per assegurar-nos que no siguin clicables mentre es demana la contrasenya, perquè ens pot donar error a l'hora de mostrar contingut en el *main* que perjudiqui la visualització de la nostra aplicació.

➡ Amb això obliguem l'usuari a introduir la contrasenya abans de poder seguir navegant en els nostres apartats ocults.

```
function showPasswordPrompt() {
  document.getElementById("password-prompt").style.display = "block";
  document.getElementById("main-content").style.display = "none";
  document.getElementById("incidencies-link").style.pointerEvents = "none";
  document.getElementById("projecte-link").style.pointerEvents = "none"; // Desactivar el enlace
}
```

## 3. Funció checkPassword:

Principalment, l'ús d'aquesta funció és per a comprovar que la contrasenya introduïda per l'usuari, comprova que sigui la correcta o no, en cas que no ho sigui, mostrarà un missatge.

Agafem el valor introduït al camp de contrasenya (input amb id *"password"*). Si la contrasenya coincideix amb la que hem definit, guardem la informació d'accés concedit al *sessionStorage* (clau *"accessGranted" = "true"*). Després redirigim l'usuari automàticament a la pàgina de */incidencies*.

Si la contrasenya no és correcta, mostrem un missatge d'error (l'element amb id `"error-message"`).

```
function checkPassword() {
  const enteredPassword = document.getElementById("password").value;
  if (enteredPassword === correctPassword) {
    sessionStorage.setItem("accessGranted", "true"); // Guardar en sessionStorage que el acceso fue concedido
    window.location.href = "/incidencies";
  } else {
    document.getElementById("error-message").style.display = "block";
  }
}
```

## LIST.EJS

En aquest script hem creat dues funcions per gestionar l'estat dels enllaços del nostre header i el sistema d'accés de la sessió.

```
<script>
  // Funció per desactivar els enllaços del header
  function disableHeaderLinks() {
    document.getElementById("projecte-link").style.pointerEvents = "none";
    document.getElementById("incidencies-link").style.pointerEvents = "none";
  }

  // Funció per resetar l'accés (per tancar sessió)
  function resetAccess() {
    sessionStorage.removeItem("accessGranted");
    window.location.href = "/";
  }
</script>
```

### 1. Funció disableHeaderLinks:

Aquesta funció la fem servir per desactivar els enllaços del header quan volem impedir que l'usuari accedeixi a certes seccions de la web. Concretament, el que fem és:

Accedir a l'element amb l'id `"projecte-link"` i li establim la propietat CSS `pointerEvents` a `"none"`, cosa que fa que aquest enllaç deixi de ser clicable per a l'usuari. Farem també el mateix amb l'element que té l'id `"incidencies-link"` per la mateixa raó.

➡ Amb això, evitem que els usuaris puguin interactuar amb aquests enllaços quan no tenen els permisos necessaris.

```
// Funció per desactivar els enllaços del header
function disableHeaderLinks() {
  document.getElementById("projecte-link").style.pointerEvents = "none";
  document.getElementById("incidencies-link").style.pointerEvents = "none";
}
```

## 2. Funció resetAccess:

Aquesta funció serveix per a quan l'usuari tanca la sessió o volem reiniciar el seu accés. El que fem aquí és:

Eliminar la clau *"accessGranted"* del *sessionStorage*, així ens assegurem que es destrueix la informació d'accés temporal i també volem redirigir l'usuari cap a la pàgina d'inici, perquè així torni al punt de partida després del logout.

➡ Aquesta funció ens ajuda a gestionar la seguretat i el tancament correcte de sessió.

```
function resetAccess() {
  sessionStorage.removeItem("accessGranted");
  window.location.href = "/";
}
```

## DB.JS

En aquest fitxer hem creat la configuració de la connexió a la base de dades utilitzant Sequelize..

```
// src/db.js
const { Sequelize } = require('sequelize');
require('dotenv').config();

const sequelize = new Sequelize(

  process.env.MYSQL_DATABASE,

  process.env.MYSQL_USER,

  process.env.MYSQL_PASSWORD,

  {
    host: process.env.MYSQL_HOST,

    dialect: 'mysql',
  }
);

module.exports = sequelize;
```

## 1. Importacions:

Primer importem el mòdul Sequelize de la llibreria sequelize.

També fem *"require('dotenv').config();"* per carregar automàticament les variables d'entorn definides al fitxer .env. Això ens permet tenir les dades delicades com l'usuari i la contrasenya fora del codi font.

## 2. Creació de la connexió:

Instanciem un nou objecte Sequelize passant-hi quatre paràmetres:

Nom de la base de dades (process.env.MYSQL\_DATABASE).

Usuari de la base de dades (process.env.MYSQL\_USER).

Contrasenya de la base de dades (process.env.MYSQL\_PASSWORD).

Un objecte de configuració on especifiquem:

host: La IP o URL del servidor MySQL (process.env.MYSQL\_HOST).

dialect: El tipus de base de dades; en el nostre cas 'mysql'.

➡ Amb això, ens assegurem que la connexió sigui dinàmica i pugui adaptar-se fàcilment a diferents entorns (producció, desenvolupament, etc.).

```
const sequelize = new Sequelize(  
  
  process.env.MYSQL_DATABASE,  
  process.env.MYSQL_USER,  
  process.env.MYSQL_PASSWORD,  
  {  
    host: process.env.MYSQL_HOST,  
    dialect: 'mysql',  
  }  
);
```

### 3. Exportació:

Finalment exportem l'objecte sequelize perquè pugui ser utilitzat a la resta de la nostra aplicació quan necessitem interactuar amb la base de dades.

```
// src/app.js  
  
const express = require('express');  
require('dotenv').config();  
const sequelize = require('./db');  
const path = require('path');  
  
const Incidencia = require('./models/Incidencia');  
const Departament = require('./models/Departament');  
const Tecnic = require('./models/Tecnic');  
const Actuacio = require('./models/Actuacio');  
  
// Relacions  
Departament.hasMany(Incidencia, { foreignKey: 'id_departament', onDelete: 'CASCADE', onUpdate: 'CASCADE' });  
Incidencia.belongsTo(Departament, { foreignKey: 'id_departament', onUpdate: 'CASCADE' });  
  
Tecnic.hasMany(Incidencia, { foreignKey: 'id_tecnic', onDelete: 'CASCADE', onUpdate: 'CASCADE' });  
Incidencia.belongsTo(Tecnic, { foreignKey: 'id_tecnic', onUpdate: 'CASCADE' });  
  
Incidencia.hasMany(Actuacio, { foreignKey: 'id_incidencia', onDelete: 'CASCADE', onUpdate: 'CASCADE' });  
Actuacio.belongsTo(Incidencia, { foreignKey: 'id_incidencia', onUpdate: 'CASCADE' });  
  
Tecnic.hasMany(Actuacio, { foreignKey: 'id_tecnic', onDelete: 'CASCADE', onUpdate: 'CASCADE' });  
Actuacio.belongsTo(Tecnic, { foreignKey: 'id_tecnic', onUpdate: 'CASCADE' });
```



```
// Rutes
const incidenciesRoutesEJS = require('./routes/incidenciesEJS.routes');
const departamentsRoutesEJS = require('./routes/departamentsEJS.routes');
const actuacionsRoutesEJS = require('./routes/actuacionsEJS.routes');

const app = express();
app.use(express.urlencoded({ extended: true }));
app.use(express.json());
app.use('/images', express.static(path.join(__dirname, 'public/images')));

// Configuració EJS
app.set('view engine', 'ejs');
app.set('views', path.join(__dirname, 'views'));

// Rutes EJS
app.use('/incidencies', incidenciesRoutesEJS);
app.use('/departaments', departamentsRoutesEJS);
app.use('/actuacions', actuacionsRoutesEJS);

// Ruta inicial
app.get('/', (req, res) => {
  res.render('index');
});
```

```
const port = process.env.PORT || 3000;

(async () => {
  try {
    await sequelize.sync();
    console.log('Base de dades sincronitzada (API JSON)');

    const existingDepartments = await Departament.findOne();
    if (!existingDepartments) {
      await Departament.bulkCreate([
        { nom_dpt: 'Matemàtiques' },
        { nom_dpt: 'Llengua Castellana' },
        { nom_dpt: 'Llengua Catalana' },
        { nom_dpt: 'Ciències Socials' },
        { nom_dpt: 'Ciències Naturals' },
        { nom_dpt: 'Educació Física' },
        { nom_dpt: 'Anglès' },
        { nom_dpt: 'Llengües Estrangeres' },
        { nom_dpt: 'Filosofia' },
        { nom_dpt: 'Tecnologia' },
        { nom_dpt: 'Informàtica' },
        { nom_dpt: 'Educació Plàstica i Visual' },
        { nom_dpt: 'Música' },
        { nom_dpt: 'Secretaria' },
        { nom_dpt: 'Administració' },
        { nom_dpt: 'Recursos Humans' },
        { nom_dpt: 'Direcció' },
        { nom_dpt: 'Coordinació de Cicles Formatius' },
        { nom_dpt: 'Biblioteca Escolar' },
      ]);
      console.log('Departaments inicials creats');
    }
  }
})
```

```

const existingTechnicians = await Tecnic.findOne();
if (!existingTechnicians) {
  await Tecnic.bulkCreate([
    { nom: 'Joan Pérez' },
    { nom: 'Maria López' },
    { nom: 'Carles Garcia' },
    { nom: 'Anna Martínez' },
    { nom: 'Pere Font' },
    { nom: 'Laura Vidal' },
    { nom: 'Marc Soler' },
    { nom: 'Clara Roca' },
    { nom: 'Jordi Serra' },
    { nom: 'Marta Bosch' },
  ]);
  console.log('Tècnics inicials creats');
}

const existingIncidents = await Incidencia.findOne();
if (!existingIncidents) {
  await Incidencia.bulkCreate([
    { id_departament: 1, id_tecnic: 1, descripcio: 'Error en la impressora', prioritat: 'Alta', estat: 'No resolt' },
    { id_departament: 2, id_tecnic: 2, descripcio: 'Error de xarxa', prioritat: 'Alta', estat: 'No resolt' },
    { id_departament: 3, id_tecnic: 3, descripcio: 'Error en la pàgina web', prioritat: 'Mitjana', estat: 'No resolt' },
    { id_departament: 4, id_tecnic: 4, descripcio: 'Cablejat desordenat', prioritat: 'Baixa', estat: 'No resolt' },
    { id_departament: 5, id_tecnic: 5, descripcio: 'Pantalla en mal estat', prioritat: 'Mitjana', estat: 'No resolt' },
  ]);
  console.log('Incidència inicials creats');
}

```

```

const existingTechnicians = await Tecnic.findOne();
if (!existingTechnicians) {
  await Tecnic.bulkCreate([
    { nom: 'Joan Pérez' },
    { nom: 'Maria López' },
    { nom: 'Carles Garcia' },
    { nom: 'Anna Martínez' },
    { nom: 'Pere Font' },
    { nom: 'Laura Vidal' },
    { nom: 'Marc Soler' },
    { nom: 'Clara Roca' },
    { nom: 'Jordi Serra' },
    { nom: 'Marta Bosch' },
  ]);
  console.log('Tècnics inicials creats');
}

const existingIncidents = await Incidencia.findOne();
if (!existingIncidents) {
  await Incidencia.bulkCreate([
    { id_departament: 1, id_tecnic: 1, descripcio: 'Error en la impressora', prioritat: 'Alta', estat: 'No resolt' },
    { id_departament: 2, id_tecnic: 2, descripcio: 'Error de xarxa', prioritat: 'Alta', estat: 'No resolt' },
    { id_departament: 3, id_tecnic: 3, descripcio: 'Error en la pàgina web', prioritat: 'Mitjana', estat: 'No resolt' },
    { id_departament: 4, id_tecnic: 4, descripcio: 'Cablejat desordenat', prioritat: 'Baixa', estat: 'No resolt' },
    { id_departament: 5, id_tecnic: 5, descripcio: 'Pantalla en mal estat', prioritat: 'Mitjana', estat: 'No resolt' },
  ]);
  console.log('Incidència inicials creats');
}

```

## APP.JS

Hem creat l'arxiu principal del nostre projecte, el qual:

Inicialitza l'aplicació amb Express, configura la connexió a la base de dades, defineix les rutes tant per a l'API com per a les vistes EJS, crea les relacions

entre els nostres models Sequelize, sincronitza la base de dades i crea dades inicials si encara no existeixen:

## **Dependencies:**

Express: Per gestionar el servidor web.

dotenv: Per llegir variables d'entorn des del .env.

sequelize: Importem la connexió que hem definit a db.js.

path: Per gestionar rutes de fitxers.

## **Models:**

Importem els models (Incidencia, Departament, Tècnic, Actuacio).

## **Relacions:**

Departament-Incidencia: Un departament pot tenir moltes incidències.

Tecnic-Incidencia: Un tècnic pot tenir moltes incidències.

Incidencia-Actuacio: Una incidència pot tenir moltes actuacions, i cada actuació pertany a una incidència.

## **Rutes:**

Rutes EJS: Per gestionar la interfície visual.

Rutes API: Per exposar la nostra API.

## **Servidor:**

Fem sequelize.sync() per sincronitzar la base de dades amb els models.

Si tot funciona correctament, escoltem al port definit des de .env o per defecte 3000, en cas d'error, capturem i mostrem l'error a consola.

## **Chatbot:**

Aquí tenim l'explicació del nostre chatbot, hem utilitzat una eina que es diu VoiceFlow, que es l'encarregat de gestionar el nostre chatbot, mitjançant un diagrama de casos depenent del que el usuari cliqui, mostrarà un contingut o un altre.

```

<script type="text/javascript">
(function(d, t) {
  var v = d.createElement(t), s = d.getElementsByTagName(t)[0];
  v.onload = function() {
    window.voiceflow.chat.load({
      verify: { projectID: '68089b1c3471dd67f6d121b0' },
      url: 'https://general-runtime.voiceflow.com',
      versionID: 'production',
      voice: {
        url: 'https://runtime-api.voiceflow.com'
      }
    });
  }
  v.src = "https://cdn.voiceflow.com/widget-next/bundle.mjs"; v.type = "text/javascript"; s.parentNode.insertBefore(v, s);
})(document, 'script');
</script>

```



# Bootstrap:

## Introduccio:

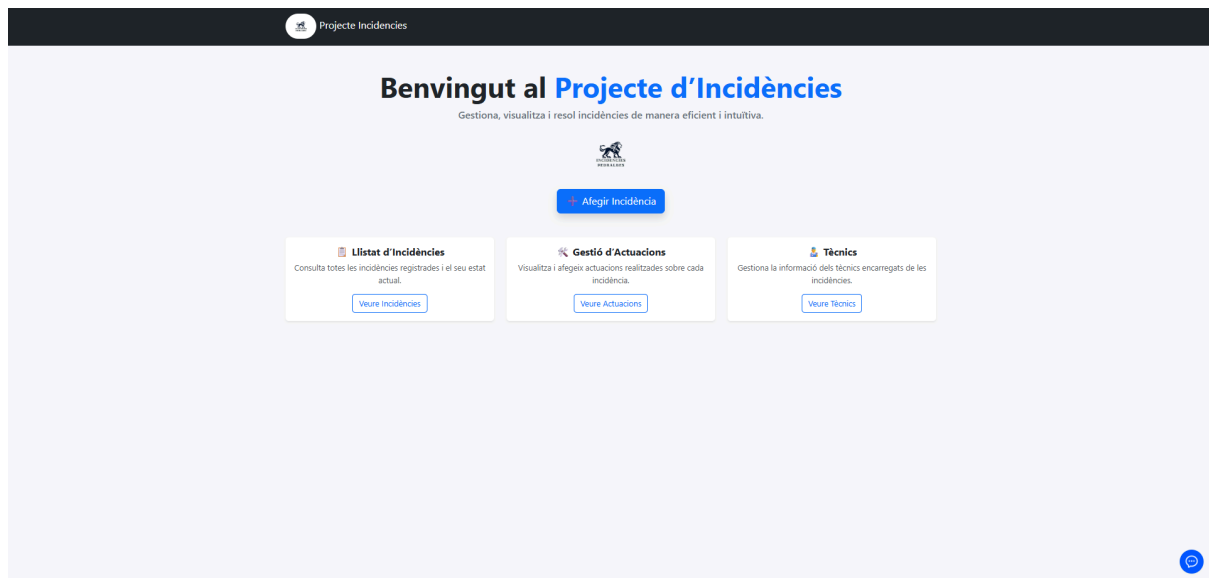
Bootstrap és un framework de CSS i JavaScript que permet crear interfícies web atractives, adaptatives (responsive) i coherents amb un esforç mínim.

Hem utilitzat aquesta tecnologia per poder donar un disseny atractiu als nostres fitxers ejs que s'encarreguen de donar la interactivitat visual per als usuaris que vulguin utilitzar la nostra aplicació de gestió d'incidències.



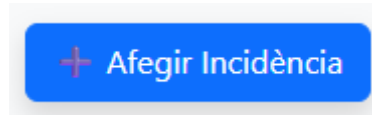
## index.ejs:

El fitxer index.ejs és la pàgina d'inici de l'aplicació i inclou diversos components visuals que aprofiten el potencial de Bootstrap.



Pel que fa a la tipografia, Bootstrap ofereix classes com `display-4`, utilitzada per destacar el títol principal amb una mida de lletra gran, o bé `lead`, que aplica un estil de paràgraf més destacat, ideal per a subtítols o frases introductòries. Altres classes com `fw-bold` o `fw-semibold` controlen el gruix de la lletra (negreta i semi-negreta, respectivament), mentre que `text-dark`, `text-secondary` i `text-primary` s'encarreguen de la coloració del text.

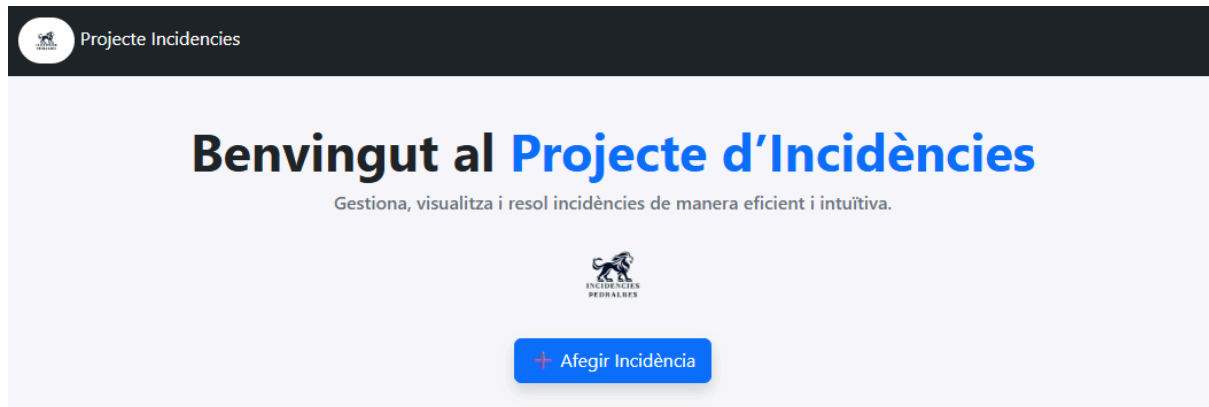
Un element clau d'interacció és el botó d'afegir una nova incidència. En aquest cas, s'utilitza el component de botó de Bootstrap (`btn`) amb combinacions com `btn-primary` per aplicar el color corporatiu, `btn-lg` per augmentar-ne la mida i `shadow` per afegir una lleugera ombra que dona sensació de relleu. Aquest estil es repeteix en altres botons amb variacions com `btn-outline-primary`, que crea un botó només amb el contorn acolorit.



El disseny responsiu, un dels punts forts de Bootstrap, es gestiona mitjançant el sistema de graells. La classe `row` serveix per agrupar columnes dins una mateixa línia horitzontal, mentre que `col-md-4` defineix columnes que ocupen una tercera part de l'ample de la pantalla en dispositius de mida mitjana o superior. Això assegura que el contingut s'adapti correctament a qualsevol dispositiu, mostrant-se en columnes separades a ordinadors i apilades a mòbils.

Dins d'aquestes columnes s'hi defineixen targetes (`card`), un component visual que permet mostrar informació agrupada d'una manera estèticament atractiva. A cada targeta se li aplica una ombra suau (`shadow-sm`) i s'elimina el contorn exterior (`border-0`) per donar-li un aspecte més net.

A més dels estils proporcionats per Bootstrap, també s'inclouen alguns estils en línia personalitzats, com és el cas de la imatge del logotip, que té una alçada fixada i un radi de cantonada per donar-li forma arrodonida. Tot això s'integra de manera coherent amb la resta d'elements proporcionats pel framework.



## list.ejs, edit.ejs i new.ejs:

A través dels fitxers de vista `list.ejs`, `new.ejs` i `edit.ejs`, hem implementat una estructura visual consistent, aprofitant les utilitats i components que Bootstrap proporciona.

Aquest fitxer presenta el llistat d'incidències. Bootstrap hi juga un paper clau en l'organització visual:

- El contingut està dins d'un `container` amb `py-5` per oferir espais equilibrats.
- S'utilitza `d-flex`, `justify-content-between` i `align-items-center` per alinear el títol i el botó de retorn, donant una estructura clara i responsive.
- L'enllaç per afegir una nova actuació es mostra com un botó primari gran (`btn btn-lg btn-primary mt-3 shadow`), reforçant la jerarquia visual.
- La taula principal fa ús de les classes `table`, `table-hover` i `align-middle`, i es troba dins d'una targeta (`card`) amb `shadow-sm` i `border-0`, cosa que afegeix un toc net i modern.
- El formulari de filtratge incorpora un `input` amb `form-control`, i la seva funcionalitat es complementa amb JavaScript per filtrar les files per ID d'incidència en temps real.






## Incidències pendents

[Veure actuacions](#)

ID	DESCRIPCIÓ	TIPUS D'INCIDENCIA	DEPARTAMENT	TÈCNIC	PRIORITAT	ESTAT	ACCIONS
1	Error en la impressora	Hardware	Matemàtiques	Joan Pérez	Alta	No resolt	 
2	Error de xarxa	Software	Llengua Castellana	Maria López	Alta	No resolt	 
3	Error en la pàgina web	Xarxa	Llengua Catalana	Carles Garcia	Mitjana	No resolt	 
4	Cablejat desordenat	Impressió	Ciències Socials	Anna Martínez	Baixa	No resolt	 
5	Pantalla en mal estat	Seguretat	Ciències Naturals	Pere Font	Mitjana	No resolt	 

 Tancar sessió

Aquest fitxer conté el formulari per crear una nova actuació, i presenta una estructura clara i centrada en l'usuari:

- El formulari està embolcallat dins d'una **card** amb **p-4 shadow**, que el diferencia visualment de la resta del contingut.
- S'utilitzen les classes **form-label**, **form-control** i **form-select** per donar consistència als inputs, potenciant la llegibilitat i l'accessibilitat.
- El selector de temps està automatitzat per generar opcions cada 5 minuts fins a 120 minuts.
- Els botons de submissió i cancel·lació segueixen la línia visual de Bootstrap (**btn btn-success** i **btn btn-secondary**), destacant la crida a l'acció i oferint una opció de retorn clara.



Nova Incidència

Descripció

Introdueix la descripció de la incidència

Departament

-- Selecciona un departament --

Tipus

-- Selecciona un tipus --

✓ Crear

✕ Cancel·lar

### 3. Fitxer `edit.ejs`: formulari de modificació d'incidències

El formulari d'edició d'incidències manté la coherència visual amb la resta de vistes:

- Es centra dins del `container` amb un `max-width` establert per assegurar una visualització equilibrada en pantalles grans i petites.
- S'hi utilitzen els mateixos components de formulari (`form-label`, `form-control`, `form-select`) per editar els atributs de l'incidència: descripció, prioritat, departament, tipus, tècnic i estat.
- Totes les opcions són preseleccionades segons les dades existents mitjançant condicions embegudes dins les etiquetes `<option>`, mantenint una experiència d'usuari contínua.
- El botó de submissió `btn btn-primary` es presenta dins d'un contenidor centrat (`text-center`) per reforçar la crida a l'acció.

projecte Incidències

## Editar Incidència

Descripció

Error en la impressora

Prioritat

Alta

Departament

-- Selecciona un Departament --

Tipus

-- Selecciona el Tipus d'incidència --

Tècnic

-- Selecciona un Tècnic --

Estat

No resolt

Actualitzar

## header.ejs i footer.ejs:

El fitxer header.ejs inclou l'estructura HTML bàsica, l'enllaç als estils de Bootstrap 5.3 i Bootstrap Icons, una barra de navegació amb logotip, i un formulari ocult de contrasenya amb control d'accés mitjançant JavaScript. Aquesta funcionalitat permet protegir pàgines específiques abans d'accedir-hi.

El fitxer footer.ejs tanca l'estructura HTML i carrega els scripts de Bootstrap necessaris per al bon funcionament dels components interactius.

Aquest canvi millora la claredat del projecte i facilita l'escalabilitat de l'aplicació.