

“Sistema de Gestión de Expedientes MP – Manual Técnico Detallado”

Subtítulo:

Documentación exhaustiva para desarrolladores, administradores de sistemas e ingenieros de soporte sobre la arquitectura, implementación y mantenimiento del Sistema de Gestión de Expedientes del Ministerio Público (MP).

Elaborado por:

Jorge Noel García Valenzuela

Fecha:

23 de noviembre del 2025

Introducción

El presente documento describe los aspectos técnicos del Sistema de Gestión de Expedientes, una aplicación web desarrollada para optimizar la administración de expedientes de la Dirección de Investigaciones Criminalísticas (DICRI) y los indicios asociados.

La aplicación implementa flujos de trabajo para el registro, la aprobación y el rechazo de expedientes, junto con una gestión detallada de los indicios por parte de personal técnico y coordinadores. El sistema incorpora un robusto control de acceso y seguridad basado en roles.

Objetivos del Sistema:

- Administrar el ciclo de vida de los expedientes, incluyendo la gestión de sus estados (Pendiente, Aprobado, Rechazado).
- Facilitar el registro y la vinculación de indicios específicos a cada expediente.
- Asegurar la integridad y el control de las operaciones mediante un esquema de permisos basado en roles de usuario (Técnico / Coordinador).
- Proporcionar herramientas de reporte y análisis estadístico, con capacidad de filtrado por estado y rango de fechas.

Tecnologías utilizadas

Frontend: React + Vite, TailwindCSS

Estado / contexto: React Context API ([AuthContext](#))

Backend: Node.js + Express

Base de datos: SQL Server

Conexión a BD: [mssql](#)

Reportes: [chart.js](#) / [react-chartjs-2](#), [jsPDF](#) + [jspdf-autotable](#), [xlsx](#)

Control de versiones: Git + GitHub

 22jgarcia	chore: inicial proyecto strutura para backend/frontend folders y giti...		38e8389 · 2 days ago	 2 Commits
 backendMP	chore: initial projet setup		2 days ago	
 frontend	chore: initial project setup		2 days ago	
 scripts	chore: inicial proyecto strutura para backend/frontend ...		2 days ago	
 .gitignore	chore: inicial proyecto strutura para backend/frontend ...		2 days ago	

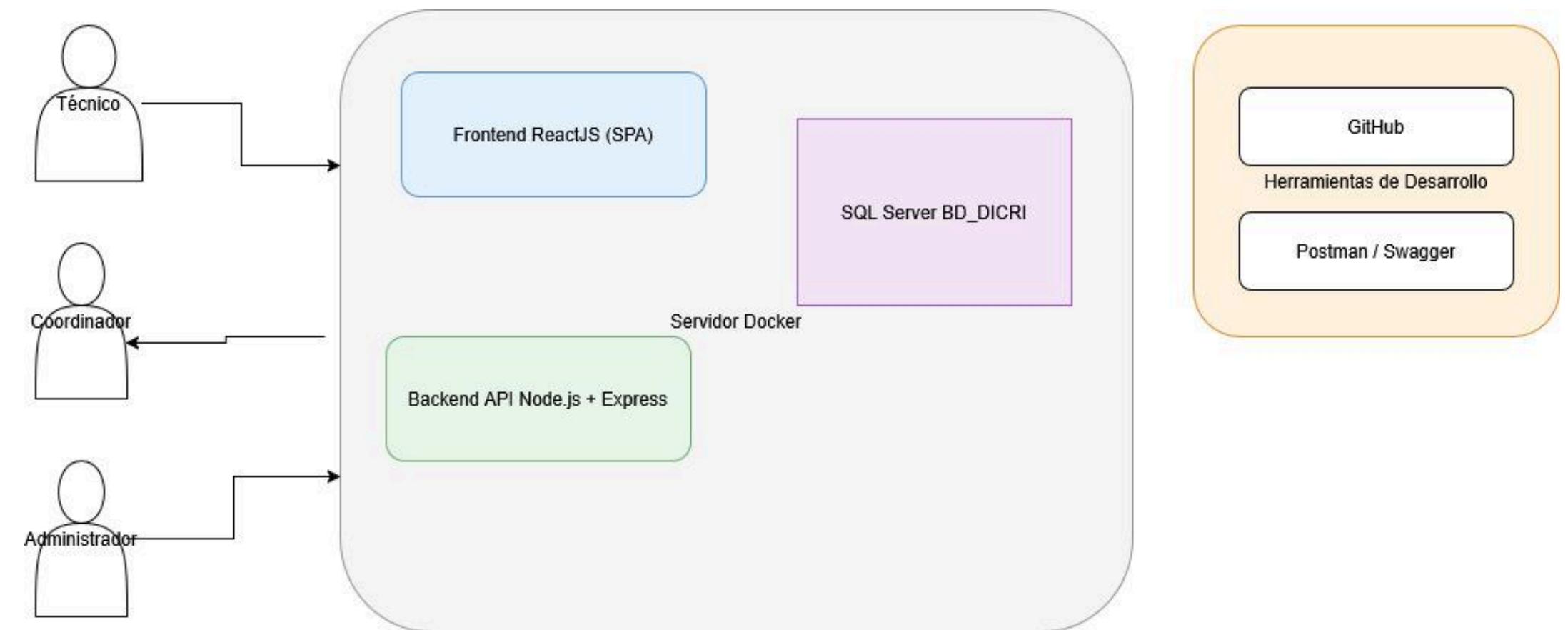
Arquitectura general del sistema

El Cliente React se comunica con la API REST (<http://localhost:3000>), la cual utiliza procedimientos almacenados en SQL Server para las operaciones CRUD.

Endpoints expuestos:

- [/login](#)
- [/expedientes](#), [/expedientes/:id](#), [/expedientes/:id/indicios](#)
- [/expedientes/:id/aprobar](#), [/expedientes/:id/rechazar](#)
- [/indicios](#)
- [/reportes](#), [/reportes/filtro](#), [/reportes/detallado](#)

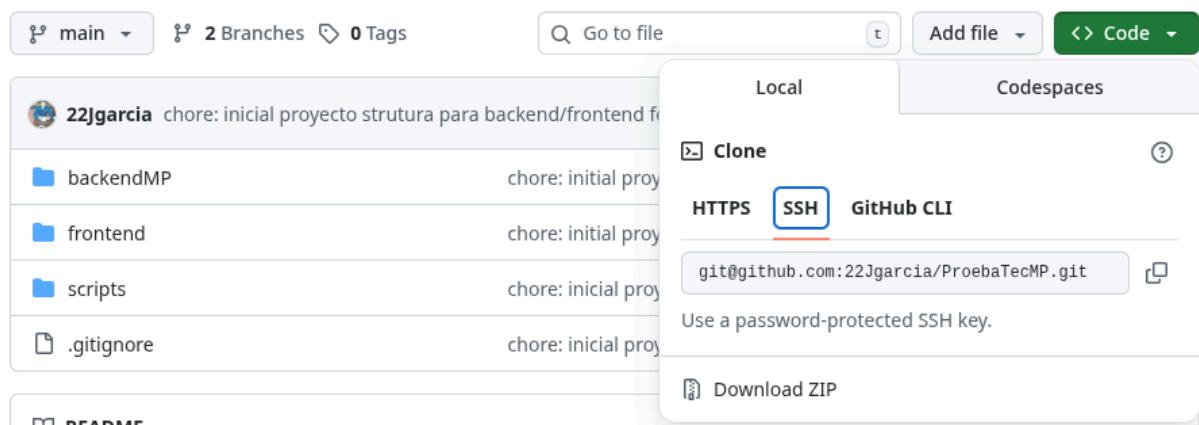
La API usa procedimientos almacenados en SQL Server para las operaciones CRUD.



Instalación y ejecución

Backend

1. Clonar repo:



2. Instalar dependencias:

```
npm install
```

3. Configurar conexión a SQL Server en config/db.js:

servidor, usuario, contraseña, base de datos.

4. Ejecutar servidor:

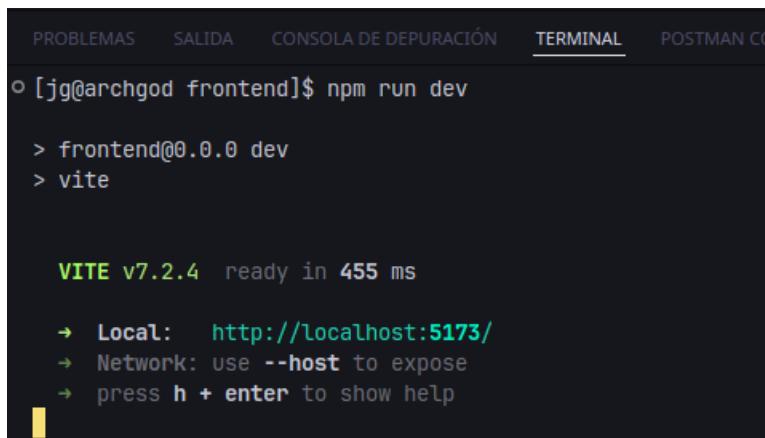
```
npm run dev
```

5. El backend queda escuchando en <http://localhost:3000>.

A screenshot of a terminal window titled 'TERMINAL'. It shows the command 'npm run dev' being executed. The output indicates that nodemon is starting the server.js file, watching for changes, and the server is now running at http://localhost:3000.

Frontend

1. Clonar repo o ir a carpeta:
cd frontendMP
npm install
npm run dev
2. La app se sirve en <http://localhost:5173>.



```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL POSTMAN CO
○ [jg@archgod frontend]$ npm run dev

> frontend@0.0.0 dev
> vite

VITE v7.2.4 ready in 455 ms

→ Local: http://localhost:5173/
→ Network: use --host to expose
→ press h + enter to show help
```

Modelo de datos y procedimientos almacenados

Tablas principales

Usuarios

- id, nombre, rol

Expedientes

- id, numero, descripcion, estado,
- fecha_registro,
- justificacion_rechazo,
- usuario_creacion, usuario_aprobacion, usuario_rechazo,
- fecha_aprobacion

Indicios

- id, expediente_id (FK),
- descripcion, cantidad,
- color, tamano, peso, ubicacion,
- tecnico

Indicios
123 id
123 expediente_id
AZ descripcion
123 cantidad
AZ color
AZ tamano
AZ peso
AZ ubicacion
AZ tecnico

Expedientes
123 id
AZ numero
AZ descripcion
AZ estado
① fecha_registro
AZ justificacion_rechazo
AZ usuario_creacion
① fecha_aprobacion
AZ usuario_aprobacion
AZ usuario_rechazo

Usuarios
123 id
AZ nombre
AZ rol

Procedimientos almacenados

SP_RegistrarExpediente

SP_ListarExpedientes

SP_ObtenerExpediente

SP_AprobarExpediente

SP_RechazarExpediente

SP_ReporteExpedientesPorEstado

SP_ReporteExpedientesDetallado

SP_RegistrarIndicio

Backend

Rutas de usuarios

Archivo: [routes/usuarios.routes.js](#)

- **POST /usuarios/login**

Recibe **nombre**, busca en tabla **Usuarios**, retorna { id, nombre, rol }.



```
1 const { Router } = require("express");
2 const controller = require("../controllers/usuarios.controller");
3
4 const router = Router();
5
6 router.get("/", controller.listarUsuarios);
7 router.post("/login", controller.login);
8
9 module.exports = router;
10
```

Rutas de expedientes

Archivo: [routes/expedientes.routes.js](#)

- **GET /expedientes**
- **POST /expedientes** (solo coordinador)
- **GET /expedientes/:id**
- **GET /expedientes/:id/indicios**
- **PUT /expedientes/:id/aprobar**

- PUT /expedientes/:id/rechazar

El middleware `requireRole` basado en header `rol` y `usuario`.

```

● ● ●
1 function requireRole(rolesPermitidos = []) {
2   return (req, res, next) => {
3     const rol = req.headers["rol"];
4
5     if (!rol) {
6       return res.status(401).json({ error: "Rol no enviado en headers" });
7     }
8     if (!rolesPermitidos.includes(rol)) {
9       return res.status(403).json({ error: "Acceso denegado: rol no autorizado" });
10    }
11    next();
12  };
13}
14
15 module.exports = requireRole;
16
17

```

Rutas de reportes

`GET /reportes` → resumen por estado

`GET /reportes/filtro` → con filtros `estado`, `fechainicio`, `fechaFin`

`GET /reportes/detallado` → listado detallado (para PDF/Excel)

```

● ● ●
1 const { Router } = require("express");
2 const controller = require('../controllers/reportes.controller');
3 const requireRole = require("../middleware/roles");
4
5 const router = Router();
6
7 router.get("/", requireRole(["coordinador"]), controller.listarReportes);
8 router.get("/estados", requireRole(["coordinador"]), controller.reportePorEstado);
9 router.get("/filtro", requireRole(["coordinador"]), controller.reporteFiltro);
10 router.get("/detallado", requireRole(["coordinador"]), controller.reporteDetallado);
11
12 module.exports = router;
13

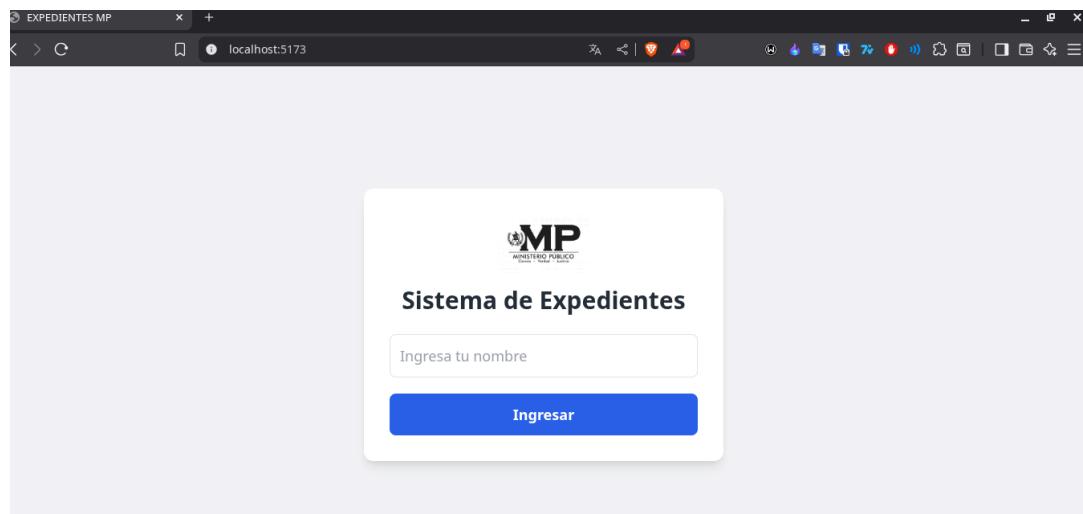
```

Frontend

Login

Archivo: `Login.jsx`

- Input de nombre.
- Llamada a `/usuarios/login`.
- Guarda `{nombre, rol}` en `localStorage` y en `AuthContext`.
- Navega a `/expedientes`.



```
1 import { useContext, useState } from "react";
2 import { AuthContext } from "../context/AuthContext";
3 import { useNavigate } from "react-router-dom";
4 import logoMP from "../assets/Logo.jpg";
5
6 export default function Login() {
7   const { login } = useContext(AuthContext);
8   const navigate = useNavigate();
9
10  const [nombre, setNombre] = useState("");
11
12  const handleLogin = async () => {
13    const res = await fetch("http://localhost:3000/usuarios/login", {
14      method: "POST",
15      headers: { "Content-Type": "application/json" },
16      body: JSON.stringify({ nombre }),
17    });
18
19    const data = await res.json();
20
21    if (data.error) {
22      alert("Usuario no encontrado");
23      return;
24    }
25
26    // GUARDAR usuario completo
27    login(data.nombre, data.rol);
28
29    navigate("/expedientes");
30  };
31  return (
32    <div className="min-h-screen flex items-center justify-center bg-gray-100">
33      <div className="bg-white shadow-lg p-8 rounded-lg w-full max-w-md text-center">
34        <img src={logoMP} alt="MP" className="w-28 mx-auto mb-4 opacity-90" />
35
36        <h1 className="text-3xl font-bold text-gray-800 mb-6">
37          Sistema de Expedientes
38        </h1>
39
40        <input
41          type="text"
42          className="w-full p-3 border rounded-lg mb-5 text-lg"
43          placeholder="Ingresa tu nombre"
44          value={nombre}
45          onChange={(e) => setNombre(e.target.value)}
46        />
47
48        <button
49          onClick={handleLogin}
50          className="w-full py-3 bg-blue-600 text-white rounded-lg text-lg font-semibold hover:bg-blue-700 transition"
51        >
52          Ingresar
53        </button>
54      </div>
55    </div>
56  );
57}
```

Listado de expedientes

Archivo: [Expedientes.jsx](#)

Muestra cards con:

Número

Descripción

Estado con badge de color

“Registrado por ...”

Fecha de registro

Al hacer clic, navega a [/expedientes/:id](#).

The screenshot shows a web browser window titled "EXPEDIENTES MP" with the URL "localhost:5173/expedientes". The page has a dark header with the text "SISTEMA EXPEDIENTES" and navigation links for "Expedientes", "Reportes", "Nuevo Expediente", "Reporte Detallado", and "Salir". Below the header, the title "Gestión de Expedientes" is displayed. The main content area contains six cards, each representing an expediente:

- EXP-1010** (APROBADO) - testtt 14
Registrado por: **Coordinador**
Fecha de registro: 23/11/2025, 12:19:25 p. m.
- EXP-1009** (PENDIENTE) - expediente siuuu
Registrado por: **Coordinador**
Fecha de registro: 23/11/2025, 10:02:05 a. m.
- EXP-011** (PENDIENTE) - testttt
Registrado por: **Coordinador**
Fecha de registro: 23/11/2025, 1:05:44 a. m.
- EXP-010** (APROBADO) - expediente tessstss
Registrado por: **Coordinador**
Fecha de registro: 23/11/2025, 12:31:52 a. m.
- EXP-009** (RECHAZADO) - test de usuariocreacion
Registrado por: **Coordinador**
Fecha de registro: 22/11/2025, 11:55:30 p. m.
- EXP-008** (APROBADO) - test5
Registrado por:
Fecha de registro: 22/11/2025, 9:37:49 p. m.

```
1 import { useEffect, useState } from "react";
2 import { Link } from "react-router-dom";
3
4 export default function Expedientes() {
5   const [lista, setLista] = useState([]);
6
7   useEffect(() => {
8     const cargar = async () => {
9       const res = await fetch("http://localhost:3000/expedientes");
10      const data = await res.json();
11      setLista(data);
12    };
13    cargar();
14  }, []);
15
16  const badgeClass = (estado) => {
17    switch (estado.toLowerCase()) {
18      case "pendiente":
19        return "bg-gray-200 text-gray-700 border-gray-300";
20      case "aprobado":
21        return "bg-gray-100 text-green-800 border-green-300";
22      case "rechazado":
23        return "bg-gray-100 text-red-800 border-red-300";
24      default:
25        return "bg-gray-200 text-gray-700 border-gray-300";
26    }
27  };
28
29  return (
30    <div className="p-8 max-w-6xl mx-auto">
31      <h2 className="text-3xl font-bold text-slate-900 tracking-tight mb-8">
32        Gestión de Expedientes
33      </h2>
34
35      {lista.length === 0 && (
36        <p className="text-gray-600 text-sm">No hay expedientes registrados.</p>
37      )}
38
39      <div className="grid sm:grid-cols-2 lg:grid-cols-3 gap-6">
40        {lista.map((exp) => (
41          <Link
42            to={`/expedientes/${exp.id}`}
43            key={exp.id}
44            className=""
45            block
46            bg-white
47            border border-slate-200
48            rounded-lg
49            p-6
50            shadow-sm
51            transition-all
52            duration-200
53            hover:shadow-xl
54            hover:-translate-y-1.5
55            hover:border-slate-400
56          " >
57            {/* Header */}
58            <div className="flex justify-between items-center mb-3">
59              <h3 className="text-xl font-semibold text-slate-900 tracking-tight">
60                EXP-{exp.id.toString().padStart(3, "0")}
61              </h3>
62
63              <span
64                className={`${`px-2.5 py-1 text-xs font-medium rounded border ${badgeClass(
65                  exp.estado
66                )}`}`}
67              >
68                {exp.estado.toUpperCase()}
69              </span>
70            </div>
71
72            {/* Descripción */}
73            <p className="text-sm text-slate-700 mb-6 leading-snug line-clamp-3">
74              {exp.descripcion}
75            </p>
76
77            <p className="mt-1 text-gray-600 text-sm">
78              Registrado por:{" "}
79              <span className="font-semibold">{exp.usuario_creacion}</span>
80            </p>
81
82            {/* Fecha */}
83            <p className="text-xs text-slate-500 border-t pt-3">
84              Fecha de registro:{` `}
85              <span className="text-slate-700">
86                {new Date(exp.fecha_registro).toLocaleString()}
87              </span>
88            </p>
89          </Link>
90        ))
91      </div>
92    </div>
93  );
94}
95
```

Detalle del expediente

Archivo: [DetalleExpediente.jsx](#)

Muestra:

Número, descripción, estado, fecha.

Botones para coordinador:

Aprobar ([PUT /expedientes/:id/aprobar](#))

Rechazar con modal y justificación ([PUT /rechazar con justificacion](#))

Lista de indicios registrados.

Formulario para **técnico**:

descripción, cantidad, color, tamaño, peso, ubicación

Se envía a [POST /indicios](#).

The screenshot shows a browser window with the title 'EXPEDIENTES MP'. The address bar indicates the URL is 'localhost:5173/expedientes/1'. The main content area displays the following information:

Expediente EXP-001

Descripción: Prueba expediente
Estado: RECHAZADO
Fecha: 2025-11-21T21:38:43.753Z

Buttons: Aprobar (green) and Rechazar (red).

Indicios Registrados

Documento incautado	listo
Cantidad: 3 Color: — Tamaño: — Peso: — Ubicación: — Técnico que registró: —	Cantidad: 1 Color: — Tamaño: — Peso: — Ubicación: — Técnico que registró: —

Indicios UI:

The screenshot shows a browser window with the title 'EXPEDIENTES MP'. The address bar indicates the URL is 'localhost:5173/expedientes/1'. The main content area displays the following form:

Registrar Indicio

Form fields:

- Descripción: (input field)
- Cantidad: 1 (input field)
- Color: (input field)
- Tamaño: (input field)
- Peso: (input field)
- Ubicación: (input field)

Button: Agregar Indicio (blue button at the bottom right).

Reporte resumido

Archivo: [Reporte.jsx](#)

Filtros:

Estado

Fecha inicio / fin

Tabla con columnas: estado y cantidad.

Gráfica [Doughnut](#) por estado.

Botones de exportación (PDF, Excel, CSV, Copiar).

The screenshot shows a web application interface titled 'Reporte de Expedientes'. At the top, there is a search bar with fields for 'Estado' (Todos), 'Desde' (dd/mm/aaaa), 'Hasta' (dd/mm/aaaa), and buttons for 'Aplicar' and 'Limpiar'. Below the search bar are four export buttons: PDF, Excel, CSV, and Copiar. A table displays the count of documents by status: Aprobado (6), Pendiente (2), and Rechazado (5). To the right of the table is a donut chart illustrating the distribution of these states. The legend indicates: Aprobado (yellow), Pendiente (teal), and Rechazado (pink).

Estado	Cantidad
Aprobado	6
Pendiente	2
Rechazado	5

Reporte detallado

Archivo: [ReporteDetallado.jsx](#)

Tabla con columnas:

número, descripción, estado, fecha_registro,

usuario_creacion, justificación, usuario_aprobación, usuario_rechazo, fecha_aprobación...

Botones de exportación reutilizando [ReportButtons](#).

The screenshot shows a detailed report page titled 'Reporte Detallado de Expedientes'. It features a table with columns: Número, Descripción, Estado, Fecha Registro, Usuario Crea, and Justificación Rechazo. The first row of data is shown: EXP-014, testtt 14, Aprobado, 23/11/2025, 12:19:25, 0.0%, Coordinador, por no me gusto. Below the table are four export buttons: PDF, Excel, CSV, and Copiar.

Número	Descripción	Estado	Fecha Registro	Usuario Crea	Justificación Rechazo
EXP-014	testtt 14	Aprobado	23/11/2025, 12:19:25 0.0%	Coordinador	por no me gusto

EXP-005	as	Aprobado	22/11/2025, 12:43:18 p. m.		Lorem Ipsum
EXP-004	test-expediente	Rechazado	22/11/2025, 12:23:16 a. m.		—
EXP-002	Prueba de expediente sis	Rechazado	22/11/2025, 12:22:08 a. m.		—
3	test	Rechazado	22/11/2025, 12:20:46 a. m.		test rechazo
EXP-001	Prueba expediente	Rechazado	21/11/2025, 3:38:43 p. m.		Lorem Ipsum



Seguridad y control de acceso

Autenticación ligera: login por nombre contra tabla **Usuarios**.

Autorización por rol:

AuthContext guarda usuario y rol.

Rutas frontend protegidas con RutaPrivada.

Backend usa middleware requireRole(["coordinador"] o ["técnico"]).

Rutas accesibles:

Técnico: expedientes, detalle, registrar indicios.

Coordinador: todo + crear, aprobar, rechazar, reportes.