

DiBiaseNet

DiBiaseNet es una webapp moderna para la gestión, consulta y visualización de documentación legislativa y bibliográfica del Concejo Municipal de San Carlos de Bariloche. Permite digitalizar, indexar y acceder a expedientes, normas, libros y materiales de interés municipal, con funcionalidades avanzadas de búsqueda, seguridad y automatización.

Tecnologías principales

- **Frontend:** React + Next.js (App Router), TypeScript, Tailwind CSS, CSS Modules
- **Backend:** Prisma ORM con PostgreSQL
- **Autenticación:** Clerk
- **UI:** Ant Design, Heroicons, react-icons
- **Procesamiento:** Zod (validación), SweetAlert2, XLSX, PapaParse, Marked, Highlight.js
- **APIs externas:** Google Sheets, Cloudinary (imágenes/PDF)
- **Gestión de datos:** Prisma, pg-promise, fs-extra

Estructura del proyecto

```
├── prisma/
│   ├── schema.prisma      # Esquema de base de datos
│   ├── migrations/        # Migraciones Prisma
│   └── seed.js            # Script de seed para proyectos
├── src/
│   ├── app/               # App Router y páginas principales
│   ├── knowledge/         # JSON de proyectos y datos
│   ├── components/       # Componentes UI (ChatBot, Libro, etc.)
│   ├── lib/              # Lógica de AI, validación, DB, loaders
│   └── api/              # Endpoints API
├── public/               # Assets estáticos
├── scripts/
│   └── setup.js          # Script de setup automatizado
└── README.md            # Este archivo
```

Instalación y configuración

1. Clona el repositorio:

```
git clone <repo-url> .
```

2. Ejecuta el setup automatizado:

```
node scripts/setup.js
```

- Crea `.env` si no existe
- Instala dependencias
- Genera cliente Prisma

3. Configura la conexión a la base de datos en `.env`:

```
DATABASE_URL="postgresql://usuario:contraseña@localhost:5432/nombre_db"
```

4. Ejecuta migraciones Prisma:

```
npx prisma migrate dev
```

5. Seed de datos reales:

```
node prisma/seed.js
```

Desarrollo

- Inicia el servidor de desarrollo:

```
npm run dev
```

- Accede a <http://localhost:3000>

Producción

- Build de producción:

```
npm run build
```

- Inicia en modo producción:

```
npm start
```

Funcionalidades detalladas

1. Gestión documental avanzada

- Acceso total a expedientes originales escaneados y normas en versión firmada
- Digitalización y despapelización: reducción de uso de papel y espacio físico
- Biblioteca digital de libros y materiales municipales

2. Búsqueda e indexación inteligente

- Búsqueda avanzada por número, tipo, autor, palabras clave y año
- Índices automáticos y filtros dinámicos
- Recuperación eficiente: reducción de tiempos de consulta en más de 60%

3. ChatBot asistente virtual

- ChatBot con IA (OpenRouter, OpenAI, Anthropic) para consultas legislativas
- Acceso y procesamiento de información desde el JSON principal ([src/knowledge/proyectos.json](#))
- Respuestas contextuales y personalizadas

4. Seguridad y control de acceso

- Autenticación robusta con Clerk
- Validación estricta de datos con Zod
- Manejo seguro de errores y logs
- Auditoría de accesos y gestión de permisos

5. Integración y automatización

- Integración con sistemas de gestión documental existentes
- APIs externas: Google Sheets, Cloudinary para imágenes y PDFs
- Scripts automatizados para setup y seed de datos

6. Analíticas y reportes

- Reportes de uso, accesos y frecuencia de consultas
- Estadísticas de satisfacción y eficiencia

7. Estructura modular y escalable

- Componentes reutilizables y lógica desacoplada
- Migraciones y seeds automatizados
- Soporte para ampliación de funcionalidades

Seguridad y buenas prácticas

- Consultas SQL protegidas con Prisma ORM
- Validación de entrada con Zod schemas
- Autenticación y control de acceso con Clerk

- Manejo seguro de errores y logs
 - Migraciones y seeds automatizados
-

Datos de interés

- JSON principal: `src/knowledge/proyectos.json` (fuente de verdad para proyectos)
 - Seed automatizado para poblar la base de datos desde JSON
 - Reducción de tiempos de consulta y costos de almacenamiento
-

Contribuciones

1. Haz fork del repositorio
 2. Crea una rama nueva (feature)
 3. Realiza commits de tus cambios
 4. Abre un pull request
-

Ejemplo de uso de endpoints y scripts

Seed de proyectos desde JSON

```
node prisma/seed.js
```

Setup automatizado

```
node scripts/setup.js
```

Sobre DiBiaseNet

DiBiaseNet automatiza la digitalización, indexación y almacenamiento, transformando la gestión documental en un proceso eficiente, seguro y sostenible. Desarrollada para optimizar la calidad legislativa y administrativa, facilita el acceso, consulta y manejo de regulaciones y expedientes, promoviendo la modernización y eficiencia en el trabajo diario.

¿Dudas o demostraciones? ¡Contáctanos!