Documentación de la API

1. Introducción

Este sistema web está construido con un **frontend** en **React**, un **backend** basado en **Node.js + Express** y una **base de datos PostgreSQL**. La comunicación se realiza a través de **API RESTful** y se sigue una **arquitectura en 3 capas (cliente ↔ servidor ↔ base de datos)**.

Funciones clave:

- Autenticación y autorización con JWT.
- Carga y gestión de archivos (PDFs, imágenes, Excel).
- CRUD completo para usuarios, PDFs, noticias e inventario.
- Procesamiento de ficheros Excel para estadísticas de días festivos.

2. Convenciones generales

- **URL base:** `https://observatorio-api-dhp4.vercel.app`
- **Autenticación:** Token JWT en header `Authorization: Bearer <token>`
- **Formato de error estándar:**

```
{ "error": "Mensaje descriptivo", "code": 400 }
```

- **Códigos de estado HTTP:**

Código	Significado
200	OK
201	Creado
400	Datos o parámetros incorrectos
401	Token ausente o inválido
403	Sin permisos suficientes
404	Recurso no encontrado
500	Error interno del servidor

- **Paginación:**
  - Parámetros: `?page=1&limit=20`
  - Respuesta ejemplo:

```
{
  "data": [ ... ],
  "page": 1,
  "limit": 20,
  "total": 100
}
```

- **Carga de archivos:**
  - Usar `multipart/form-data` para archivos binarios (PDF, imágenes, Excel).
  - Tamaño máximo recomendado: 10MB por archivo.
- **Versionado:**
  - Actualmente sin prefijo, pero se recomienda `/v1/` para futuras versiones.

3. Endpoints por módulo

3.1 PDFs Front `/pdfs-front`

Acción	Método	URL	Auth	Admin
Subir metadatos de PDF	POST	/pdfs-front/	Requerido	No
Obtener todos los PDFs	GET	/pdfs-front/	Requerido	No
Eliminar PDF por ID	DELETE	/pdfs-front/:id	Requerido	No
Actualizar PDF por ID	PUT	/pdfs-front/:id	Requerido	No

Acción	Método	URL	Auth	Admin

Ejemplo: Subir metadatos

Request:

```
{
  "title": "Ejemplo",
  "fileUrl": "https://...",
  "category": "General"
}
```

Response 201:

```
{ "id": 1, "title": "Ejemplo", "fileUrl": "https://...", "category": "General", "url": "https://..." }
```

Error 400/500:

```
{ "error": "Faltan campos obligatorios", "code": 400 }
```

3.2 Usuarios `/users`

Acción	Método	URL	Auth	Admin
Obtener tu usuario	GET	/users/me	Requerido	No
Actualizar tu usuario	PUT	/users/me	Requerido	No
Eliminar tu usuario	DELETE	/users/me	Requerido	No
Listar usuarios (admin)	GET	/users/	Requerido	Sí
Actualizar usuario (admin)	PUT	/users/:id	Requerido	Sí
Eliminar usuario (admin)	DELETE	/users/:id	Requerido	Sí

Ejemplo: Obtener usuario propio

Request header:

```
Authorization: Bearer <token>
```

Response 200:

```
{ "id": 1, "name": "Ana", "email": "ana@dominio.com", "role": "user" }
```

Error 401:

```
{ "error": "Token inválido o ausente", "code": 401 }
```

3.3 Noticias `/news`

Acción	Método	URL	Auth	Admin
Crear noticia	POST	/news/	Requerido	Sí
Obtener todas	GET	/news/	No	No
Obtener por ID	GET	/news/:id	No	No
Actualizar por ID	PUT	/news/:id	Requerido	Sí
Eliminar por ID	DELETE	/news/:id	Requerido	Sí

Ejemplo: Crear noticia

Request header:

```
Authorization: Bearer <token>
Content-Type: application/json
```

Request body:

```
{
  "title": "Nueva noticia",
  "content": "Texto de la noticia",
  "imageUrl": "https://..."
}
```

Response 201:

```
{ "id": 1, "title": "Nueva noticia", "content": "Texto de la noticia", "imageUrl": "https://..." }
```

3.4 Excel Feed `/excel-feed`

Acción	Método	URL	Auth	Admin
Subir Excel	POST	/excel-feed/upload-excel	Requerido	Sí
Consultar estadísticas	GET	/excel-feed/	Requerido	No
Eliminar todo	DELETE	/excel-feed/all	Requerido	Sí

Subir Excel

- Usar `multipart/form-data` con campo `file`.

Consultar estadísticas

- Filtros soportados: `year`, `fromYear`, `toYear`, `municipality`, `bridgeName`, `month`.
- Ejemplo: `/excel-feed/?year=2023&municipality=Colima`

3.5 Inventario PDFs `/inventory`

Acción	Método	URL	Auth	Admin
Subir PDF	POST	/inventory/	Requerido	Sí
Listar todos	GET	/inventory/	Requerido	No
Eliminar por ID	DELETE	/inventory/:id	Requerido	Sí
Actualizar por ID	PUT	/inventory/:id	Requerido	Sí

Subir PDF

- Usar `multipart/form-data` con campo `file` y metadatos JSON.

3.6 Autenticación `/auth`

Acción	Método	URL	Auth
Registro	POST	/auth/register	No
Login	POST	/auth/login	No

Registro

Request body:

```
{ "name": "Ana", "email": "ana@dominio.com", "password": "Secreta123" }
```

Response 201:

```
{ "message": "Registro exitoso" }
```

Login

Request body:

```
{ "email": "ana@dominio.com", "password": "Secreta123" }
```

Response 200:

```
{ "message": "Login successful", "token": "eyJhbGciOi..." }
```

4. Manejo de errores estándar

Código	Motivo habitual
400	Datos o parámetros incorrectos
401	Token ausente o inválido
403	Sin permisos suficientes
404	Recurso no encontrado
500	Error interno del servidor

Ejemplo de error:

```
{ "error": "Token inválido o ausente", "code": 401 }
```

5. Seguridad y buenas prácticas

1. **JWT** con caducidad razonable (p. ej. 1 h) y refresco.
2. **Rate-limiting** en rutas de autenticación.
3. **CORS** restringido a dominios confiables.
4. **Validación** exhaustiva con express-validator o Zod.
5. **CSRF** protegido en formularios si se manejan cookies.
6. Cifrado de contraseñas con **bcrypt** (mínimo 10 salt rounds).
7. Subida de archivos a **S3**/Cloud Storage con URLs firmadas.
8. **HTTPS** obligatorio en producción.
9. **Límites de tamaño de archivos**: 10MB por archivo.
10. **Roles y permisos** claros en endpoints sensibles.

6. Diagrama lógico (texto)

- **Cliente (React)** – peticiones HTTP →
- **API Gateway (Express)** – rutas /auth, /users, /pdfs-front, etc. →
- **Servicios** (controladores) →
    - **Servicio Auth** (JWT, bcrypt)
    - **Servicio PDFs** (metadatos + almacenamiento S3)
    - **Servicio News** (CRUD + imágenes)
    - **Servicio Excel** (parsing XLSX)
- **Capa de acceso a datos** (ORM Sequelize/Prisma) → **PostgreSQL**

7. Ejemplo de flujo “Subir PDF”

1. React envía **POST /pdfs-front/** con título, URL y categoría.
2. Middleware valida token y cuerpo.
3. Controlador pdfsController.create guarda registro y responde 201.
4. URL devuelta se usa en el frontend para mostrar el nuevo documento.

Notas adicionales:

- Para endpoints de subida de archivos, asegúrate de enviar el token JWT en el header.
- Si tienes dudas sobre algún endpoint, revisa los ejemplos de request y response incluidos arriba.