

Base de datos (modelo relacional simple)

Tabla: users

Campo	Tipo	Descripción
id	INT (PK, autoincrement)	Identificador del usuario
username	VARCHAR(50)	Nombre de usuario único
email	VARCHAR(100)	Correo electrónico
password_hash	VARCHAR(255)	Contraseña encriptada
created_at	DATETIME	Fecha de creación
role	ENUM('reader', 'editor', 'admin')	Rol del usuario (permite control de acceso)

Tabla: posts

Campo	Tipo	Descripción
id	INT (PK, autoincrement)	Identificador del post
title	VARCHAR(255)	Título del artículo
content	TEXT	Cuerpo del artículo
author_id	INT (FK → users.id)	Autor del artículo
created_at	DATETIME	Fecha de creación
updated_at	DATETIME	Última edición

Endpoints REST API

Base URL: /api/v1

Autenticación

Método	Endpoint	Descripción
POST	/auth/register	Registrar nuevo usuario
POST	/auth/login	Iniciar sesión (retorna token JWT)

Usuarios

Método	Endpoint	Descripción
GET	/users/me	Obtener perfil del usuario autenticado
GET	/users/:id	Obtener perfil público de otro usuario
DELETE	/users/:id	Eliminar usuario (solo el propio o admin)

Artículos

Método	Endpoint	Descripción
GET	/posts	Listar todos los artículos
GET	/posts/:id	Ver un artículo por ID
POST	/posts	Crear nuevo artículo (<i>requiere ser editor</i>)
DELETE	/posts/:id	Eliminar artículo propio o si eres admin

Documentación completa endpoints

Bearer token: Será una key larga para garantizar la seguridad del sistema.

Método	Endpoint	Headers	Body (Entrada)	Response (Salida)
POST	/api/v1/auth/register	Content-Type: application/json	json { "username": "juanperez", "email": "juan@example.com", "password": "123456" }	201 Created json { "message": "Usuario registrado correctamente", "user": { "id": 5, "username": "juanperez", "email": "juan@example.com", "role": "reader", "created_at": "2025-10-19T10:30:00Z" } }
POST	/api/v1/auth/login	Content-Type: application/json	json { "email": "juan@example.com", "password": "123456" }	200 OK json { "message": "Inicio de sesión exitoso", "token": "eyJhbGciOiJIUzI1NilsInR5cCI6IkpXVCJ9...", "user": { "id": 5, "username": "juanperez", "role": "editor" } }
GET	/api/v1/users/me	Authorization: Bearer <token>	—	200 OK json { "id": 5, "username": "juanperez", "email": "juan@example.com", "role": "editor", "created_at": "2025-10-19T10:30:00Z" }
GET	/api/v1/users/:id	(opcional) Authorization: Bearer <token>	—	200 OK json { "id": 3, "username": "maria", "role": "editor", "created_at": "2025-09-10T08:20:00Z" }
DELETE	/api/v1/users/:id	Authorization: Bearer <token>	—	200 OK json { "message": "Usuario eliminado correctamente" }

GET	/api/v1/posts	(opcional) Authorization: Bearer <token>	—	200 OK json { "posts": [{ "id": 12, "title": "Nuevas tendencias en tecnología 2025", "content": "Resumen de las principales innovaciones...", "author": { "id": 3, "username": "maria" }, "created_at": "2025-10-17T09:00:00Z", "updated_at": "2025-10-18T14:12:00Z" }] }
GET	/api/v1/posts/:id	(opcional) Authorization: Bearer <token>	—	200 OK json { "id": 12, "title": "Nuevas tendencias en tecnología 2025", "content": "Resumen de las principales innovaciones...", "author": { "id": 3, "username": "maria" }, "created_at": "2025-10-17T09:00:00Z", "updated_at": "2025-10-18T14:12:00Z" }
POST	/api/v1/posts	Authorization: Bearer <token> Content-Type: application/json	json { "title": "Mi nuevo artículo", "content": "Texto del artículo." }	201 Created json { "message": "Artículo creado correctamente", "post": { "id": 14, "title": "Mi nuevo artículo", "content": "Texto del artículo.", "author": { "id": 5, "username": "juanperez" }, "created_at": "2025-10-19T11:00:00Z", "updated_at": "2025-10-19T11:00:00Z" } }
DELETE	/api/v1/posts/:id	Authorization: Bearer <token>	—	200 OK json { "message": "Artículo eliminado correctamente" }

Código	Descripción	Ejemplo de respuesta
400	Datos inválidos / faltantes	json { "error": "Datos inválidos o incompletos" }
401	No autenticado (sin token o inválido)	json { "error": "Token inválido o ausente" }
403	No autorizado (no tiene permisos)	json { "error": "Acceso denegado" }
404	No encontrado	json { "error": "Recurso no encontrado" }
500	Error interno del servidor	json { "error": "Error interno del servidor" }