

EduFeedback

Manual de instalación

2ºDAW/2ºDAM

Adrián Álvarez Cotta
Álvaro García Marchena
Francisco Garrido Ribeiro
Juan Calixto del Hoyo

Índice

Introducción.....	1
Requisitos previos.....	1
Configuración de la base de datos.....	1
Cómo ejecutar.....	2
Endpoints principales (usados por el frontend).....	2
Notas de arquitectura (MVC reducido).....	3
Troubleshooting rápido.....	3



Introducción

EduFeedback es una Aplicación web para compartir mensajes anónimos entre alumnos, profesores y moderadores dentro de clases.

Este repositorio contiene dos subproyectos:

- buzón_app — Frontend Angular (SPA)
- buzón_api — Backend Node.js + Express + MySQL (API REST)

Backend: se sigue una estructura MVC reducida (separación en routes, controllers, services y models). Esto facilita mantenimiento y pruebas.

Frontend: Angular con servicios HTTP en src/app/services que llaman exactamente a los endpoints listados más abajo.

Requisitos previos

- Node.js 18+ (se usó Node 22 en desarrollo)
- npm
- MySQL. Se puede importar buzón_anonimo.sql para crear esquema y datos de ejemplo.
- Angular CLI para desarrollo del frontend

Configuración de la base de datos

1. Crear la base de datos buzón_anonimo e importar el fichero SQL:
2. Usando cliente MySQL:

```
mysql -u root -p
CREATE DATABASE buzón_anonimo;
USE buzón_anonimo;
source buzón_anonimo.sql;
```

O bien con phpMyAdmin:

- importar buzón_anonimo.sql en la base buzón_anonimo.
 - Ajusta credenciales en buzón_api/server.js si tu instalación MySQL no usa root sin contraseña.
3. Variables sensibles (correo, contraseñas de servicios) no deben guardarse en el repo. Actualmente el backend usa valores de ejemplo para notificaciones; para producción configura variables de entorno o un archivo .env y modifica la carga en server.js.



Cómo ejecutar

Recomendación: abrir dos terminales (uno para backend y otro para frontend).

- Backend (buzon_api):

```
cd c:\ruta\Buzon-Anonimo-PI\buzon_api
npm install
node server.js
# o, si prefieres npm start, añade en package.json: "start": "node server.js" y usa
npm start
```

El servidor por defecto escucha en <http://localhost:3000>.

- Frontend (buzon_app):

```
cd c:\ruta\Buzon-Anonimo-PI\buzon_app
npm install
npx ng serve
# accesible en http://localhost:4200
```

Si usas Docker o despliegue remoto, adapta los pasos anteriores a tu entorno.

Endpoints principales (usados por el frontend)

POST /api/login — body: { correo, pass }
POST /api/registro — body: { correo, pass, rol, nombre }
PUT /api/usuario/password — body: { id_user, nuevaPass }
GET /api/clases/:id/:rol — obtener clases del usuario
POST /api/crear-clase — body: { nombre, id_creador }
POST /api/unirse-clase — body: { id_user, codigo }
GET /api/mensajes/:idClase — obtener mensajes de una clase
POST /api/mensaje — body: { id_clase, id_autor, texto }
DELETE /api/mensaje/:id_mensaje — eliminar mensaje
GET /api/clase/:idClase — info básica de clase
DELETE /api/clase/:id_clase — borrar clase y mensajes
PUT /api/clase/codigo/:id_clase — rotar código temporal
GET /api/clase-usuarios/:id_clase — listar usuarios de la clase
DELETE /api/clase/:id_clase/usuario/:id_user — expulsar usuario

Nota: el frontend actual (buzon_app/src/app/services) llama a las rutas anteriores; si las cambias debes actualizar también los servicios del frontend.



Notas de arquitectura (MVC reducido)

Estructura que encontrarás en buzón_api:

server.js — punto de entrada y montaje de routers
routes/ — definición de rutas (p. ej. routes/authRoutes.js)
controllers/ — manejo de peticiones y respuestas
services/ — reglas de negocio y orquestación
models/ — acceso directo a la base de datos (consultas)

Este refactor se hizo de forma incremental para mantener compatibilidad con el frontend. Si añades recursos nuevos, sigue esta separación.

Troubleshooting rápido

- Error MODULE_NOT_FOUND al arrancar: revisa que no falten archivos en routes/, controllers/ o services y que no haya cambios de nombres.
- Error de conexión a MySQL: verifica credenciales, host y que el servicio mysqld esté corriendo.
- Errores FK al insertar: los IDs referenciados deben existir (crear usuarios/clases antes de insertar mensajes). No agregues datos de prueba en producción.
- Comandos útiles (PowerShell):

```
# comprobar puerto 3000
netstat -ano | Select-String ":3000"
# listar procesos node
Get-Process -Name node -ErrorAction SilentlyContinue | Select-Object
Id,ProcessName
```

