

# Memoria Explicativa Proyecto WordHub

## 1. Base de datos.

La base de datos es llamada igual que el proyecto “WordHub”, la base de datos consta de tres tablas: usuarios, publicaciones y comentarios.

SQL DE LA BASE DE DATOS:

```
CREATE TABLE usuarios (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,  
    nombre VARCHAR(50) NOT NULL,  
    email VARCHAR(100) NOT NULL UNIQUE,  
    contraseña VARCHAR(100) NOT NULL
```

```
);
```

```
CREATE TABLE publicaciones (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,  
    usuario_id INT,  
    titulo VARCHAR(100) NOT NULL,  
    contenido TEXT NOT NULL,  
    fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (usuario_id) REFERENCES usuarios(id)
```

```
);
```

```
CREATE TABLE comentarios (
```

```
    id INT AUTO_INCREMENT PRIMARY KEY,  
    publicación_id INT,  
    usuario_id INT,  
    contenido TEXT NOT NULL,  
    fecha_creacion TIMESTAMP DEFAULT CURRENT_TIMESTAMP,  
    FOREIGN KEY (publicación_id) REFERENCES publicaciones(id),  
    FOREIGN KEY (usuario_id) REFERENCES usuarios(id)
```

```
);
```

```
ALTER TABLE comentarios
```

```
DROP FOREIGN KEY comentarios_ibfk_1;
```

```
ALTER TABLE comentarios
```

```
ADD CONSTRAINT comentarios_ibfk_1
```

```
FOREIGN KEY (publicación_id) REFERENCES publicaciones(id)
```

```
ON DELETE CASCADE;
```

```
ALTER TABLE publicaciones
```

```
ADD CONSTRAINT fk_publicaciones_usuario
```

```
FOREIGN KEY (usuario_id) REFERENCES usuarios(id)
```

```
ON DELETE CASCADE;
```

```
ALTER TABLE comentarios
```

```
ADD CONSTRAINT fk_comentarios_usuario
```

```
FOREIGN KEY (usuario_id) REFERENCES usuarios(id)
```

```
ON DELETE CASCADE;
```

```
-- Índice en la columna usuario_id de la tabla publicaciones
```

```
CREATE INDEX idx_usuario_id ON publicaciones(usuario_id);
```

```
-- Índice en la columna publicación_id de la tabla comentarios
```

```
CREATE INDEX idx_publicacion_id ON comentarios(publicación_id);
```

```
-- Índice en la columna usuario_id de la tabla comentarios
```

```
CREATE INDEX idx_usuario_id_comentarios ON comentarios(usuario_id);
```

```
-- Índice en la columna email de la tabla usuarios (para acelerar el inicio de sesión)
```

```
CREATE INDEX idx_email ON usuarios(email);
```

Como se puede ver las relaciones son de la tabla publicaciones con usuario\_id a la tabla usuario al campo id, y de la tabla comentarios son publicación\_id a la tabla publicación campo id y usuario\_id a la tabla usuario campo id.

## 2. API

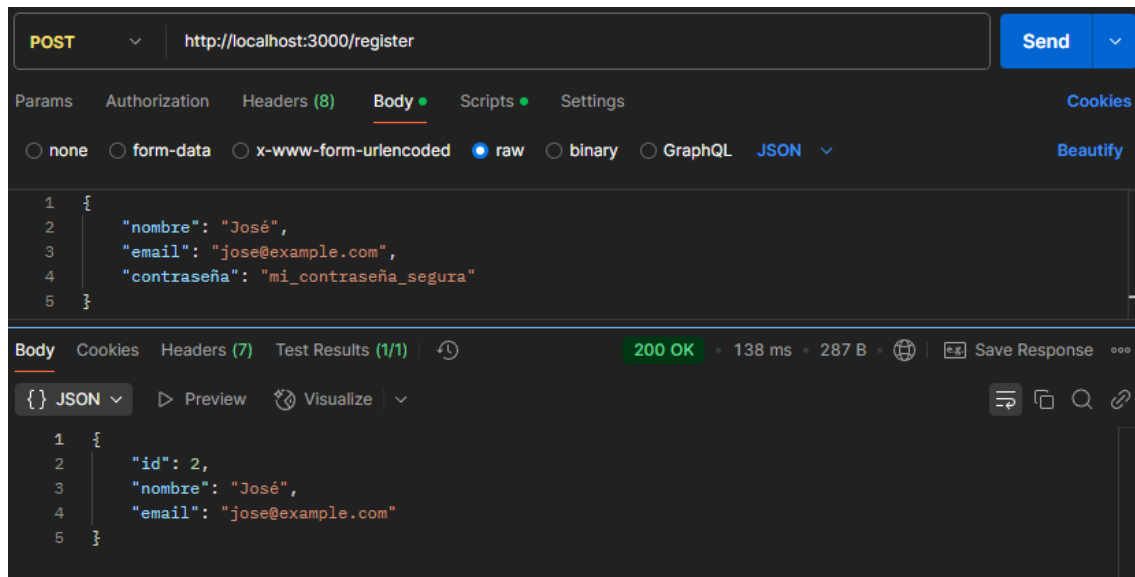
El api está realizado con node.js, para su funcionamiento se han utilizado los siguientes paquetes: express express-validator mysql body-parser bcryptjs jsonwebtoken dotenv

- **Express:**  
Express es un marco de aplicaciones web para Node.js, diseñado para crear y manejar aplicaciones web y APIs de manera sencilla y eficiente. Es uno de los marcos más populares debido a su simplicidad y flexibilidad.
- **express-validator:**  
es un paquete de Node.js que proporciona un conjunto de middlewares para validar y desinfectar datos en aplicaciones Express. Permite asegurarse de que los datos enviados por los usuarios cumplen ciertos requisitos antes de procesarlos, como formatos válidos de email, longitud de campos, o valores específicos, ayudando a prevenir errores y vulnerabilidades en la aplicación.
- **Mysql:**  
es un paquete de Node.js que permite interactuar con bases de datos MySQL. Proporciona una interfaz para conectar con la base de datos y ejecutar consultas SQL.
- **Body-parser:**  
es un middleware para Express que analiza el cuerpo de las solicitudes HTTP entrantes y lo convierte en un objeto JSON accesible desde req.body
- **Bcryptjs:**  
es una biblioteca para Node.js que implementa el algoritmo bcrypt, utilizado para el cifrado seguro de contraseñas.
- **Jsonwebtoken:**  
es una biblioteca que permite generar y verificar JSON Web Tokens (JWT), que son un estándar para la creación de tokens de autenticación.
- **Dotenv:**  
dotenv es una biblioteca que permite cargar variables de entorno desde un archivo .env en process.env. Esto es útil para almacenar información sensible, como credenciales de base de datos, claves secretas y configuraciones que no deseas exponer en el código fuente.

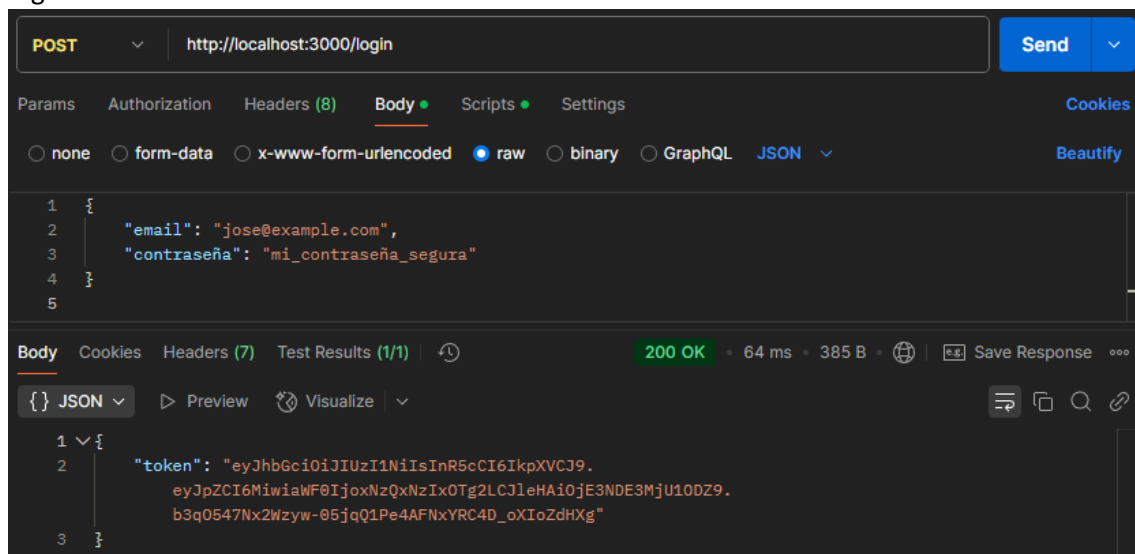
## USO DE LA API CON POSTMAN

POST:

Registrar usuarios:



login:



se devuelve un token que será necesario en las siguientes funciones.

Crear publicaciones:

POST

http://localhost:3000/publicaciones

Send

ParamsAuthorizationHeaders (9)Body ●Scripts ●SettingsCookies

Headers8 hidden

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MiwiWF0lJox...				
	Key	Value	Description			

ParamsAuthorizationHeaders (9)Body ●Scripts ●SettingsCookies

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

```
1 {
2   "titulo": "Mi primera publicación",
3   "contenido": ";Hola a todos! Esta es mi primera publicación."
4 }
5
```

BodyCookiesHeaders (7)Test Results (1/1)200 OK82 ms357 B

{ } JSONPreviewVisualize

```
1 {
2   "id": 2,
3   "usuario_id": 2,
4   "titulo": "Mi primera publicación",
5   "contenido": ";Hola a todos! Esta es mi primera publicación."
6 }
```

Comentarios

POST

http://localhost:3000/publicaciones/:2/comentarios

Send

Params ●AuthorizationHeaders (9)Body ●Scripts ●SettingsCookies

Headers8 hidden

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6MiwiWF0lJox...				
	Key	Value	Description			

ParamsAuthorizationHeaders (9)Body ●Scripts ●SettingsCookies

noneform-datax-www-form-urlencodedrawbinaryGraphQLJSON

```
1 {
2   "contenido": ";Excelente publicación!"
3 }
4
```

BodyCookiesHeaders (7)Test Results (1/1)200 OK79 ms320 B

{ } JSONPreviewVisualize

```
1 {
2   "id": 1,
3   "publicación_id": "2",
4   "usuario_id": 2,
5   "contenido": ";Excelente publicación!"
6 }
```

**PUT**

## Editar publicación

**PUT** ▼

**URL:** http://localhost:3000/publicaciones/2?Authorization=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCk

**Send** ▼

---

**Tabs:** Params • Authorization Headers (9) Body • Scripts • Settings Cookies

**Headers Section:** Headers ⚙️ 8 hidden

<input type="checkbox"/>	Key	Value	Description	
<input checked="" type="checkbox"/>	Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9....		Bulk Edit Presets ▼
<input type="checkbox"/>	Key	Value	Description	

The screenshot displays a REST client interface with a top toolbar containing radio buttons for 'none', 'form-data', 'x-www-form-urlencoded', 'raw' (selected), 'binary', and 'GraphQL', along with a 'JSON' dropdown and a 'Beautify' button. The main area shows a JSON request body:

```
1 {
2   "titulo": "Mi publicación actualizada",
3   "contenido": "He actualizado el contenido de mi publicación."
4 }
5
```

Below the request, a status bar indicates a successful response: '200 OK' in a green box, followed by '88 ms', '347 B', a globe icon, a 'Save Response' button, and a three-dot menu. The 'Body' tab is active, showing a 'JSON' dropdown, 'Preview', and 'Visualize' options. The response body is displayed as a JSON object:

```
1 {
2   "id": "2",
3   "titulo": "Mi publicación actualizada",
4   "contenido": "He actualizado el contenido de mi publicación."
5 }
```

Editar comentario:

PUT

▼

http://localhost:3000/comentarios/2

Send

▼

ParamsAuthorizationHeaders (9)Body ●Scripts ●SettingsCookies

Headers

8 hidden

	Key	Value	Description	...	Bulk Edit	Presets ▼
<input checked="" type="checkbox"/>	Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9....				
	Key	Value	Description			

Editar usuario

PUT

https://security-killer.ddns.net:3000/usuarios/2

Send

Params

Authorization

Headers (9)

Body

Scripts

Settings

Cookies

Headers

8 hidden

Key	Value	Description	Bulk Edit	Presets
Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6ImMywialWF0IjpbXzQxODUyNjI...			
Key	Value	Description		

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

1

2

3

4

5

6

```
{
  "nombre": "jmartin",
  "email": "jmartin@gmail.com",
  "contraseña": "*****"
}
```

Body

Cookies

Headers (7)

Test Results (1/1)

200 OK

451 ms

292 B

Save Response

{}

JSON

Preview

Visualize

1

2

3

4

5

```
{
  "id": "2",
  "nombre": "jmartin",
  "email": "jmartin@gmail.com"
}
```

GET

Todas las publicaciones:

GET

http://localhost:3000/publicaciones

Send

Params

Authorization

Headers (6)

Body

Scripts

Settings

Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body

Cookies

Headers (7)

Test Results (1/1)

200 OK

17 ms

406 B

Save Response

{}

JSON

Preview

Visualize

1

2

3

4

5

6

7

8

9

```
[
  {
    "id": 2,
    "usuario_id": 2,
    "titulo": "Mi publicación actualizada",
    "contenido": "He actualizado el contenido de mi publicación.",
    "fecha_creacion": "2025-03-11T18:42:57.000Z"
  }
]
```

Comentarios de una publicación:

REST API basics: CRUD, test & variable / **Get data** Save Share

**GET** ▼  Send ▼

Params Authorization Headers (6) Body Scripts ● Settings Cookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body Cookies Headers (7) Test Results (1/1) 200 OK • 15 ms • 365 B • Save Response ...

{} JSON ▼ Preview Visualize ▼

```
1 [
2   {
3     "id": 1,
4     "publicación_id": 2,
5     "usuario_id": 2,
6     "contenido": "¡Excelente publicación!",
7     "fecha_creacion": "2025-03-11T18:57:10.000Z"
8   }
9 ]
```

DEL

Eliminar comentario:

**DELETE** ▼  Send ▼

Params ● Authorization Headers (7) Body Scripts ● Settings Cookies

Headers

	Key	Value	Description	...	Bulk Edit	Presets
<input checked="" type="checkbox"/>	Authorization	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...				
	Key	Value	Description			

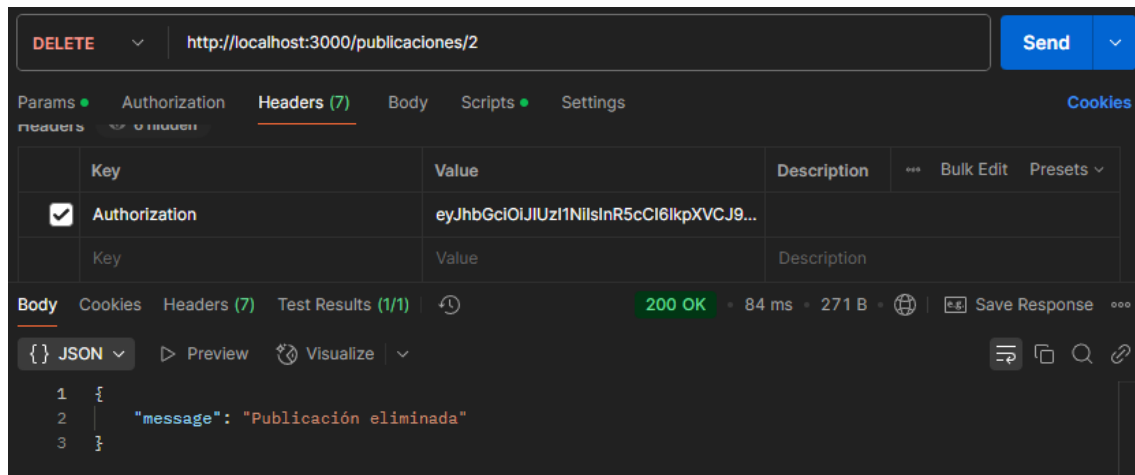
Body Cookies Headers (7) Test Results (1/1) 200 OK • 22 ms • 269 B • Save Response ...

{} JSON ▼ Preview Visualize ▼

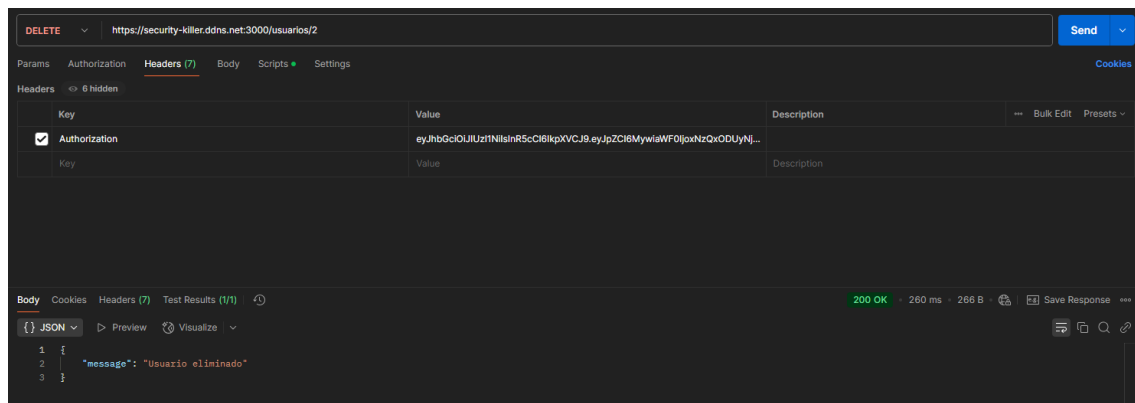
```
1 {
2   "message": "Comentario eliminado"
3 }
```



Eliminar publicación:



Eliminar usuario:



# IMPORTANTE.

En el repositorio de github está más detallado su uso, configuración, instalación y ejecución, si vas a ejecutarlo asegúrate de que has creado el fichero “.env” y has establecido las variables de entorno.

Repositorio de GitHub:

[https://github.com/jmarara529/API\\_WordHub.git](https://github.com/jmarara529/API_WordHub.git)

api hosteada en <https://security-killer.ddns.net:3000>