

# **Educación y Desarrollo Cultural de Monterrey**

Fase 2: Caso practico

Tópico desarrollo de aplicaciones web

Erick Hernandez Abrego 31945

Campus Miguel Aleman

Diseño grafico

Guadalupe, N.L. a 13 de junio del 202

## Descripción del Backend API

Se implementó un sistema CRUD completo para la entidad principal **borradores**. Las funcionalidades incluyen:

- Crear un nuevo borrador (crear.php)
- Leer todos los borradores (leer.php)
- Leer un borrador por ID (leer\_uno.php)
- Actualizar un borrador existente (actualizar.php)
- Eliminar un borrador (eliminar.php)

Todos los scripts usan sentencias preparadas con PDO para prevenir inyecciones SQL.

### Estructura de archivos del backend:

api/

└─ borradores/

└─ crear.php

└─ leer.php

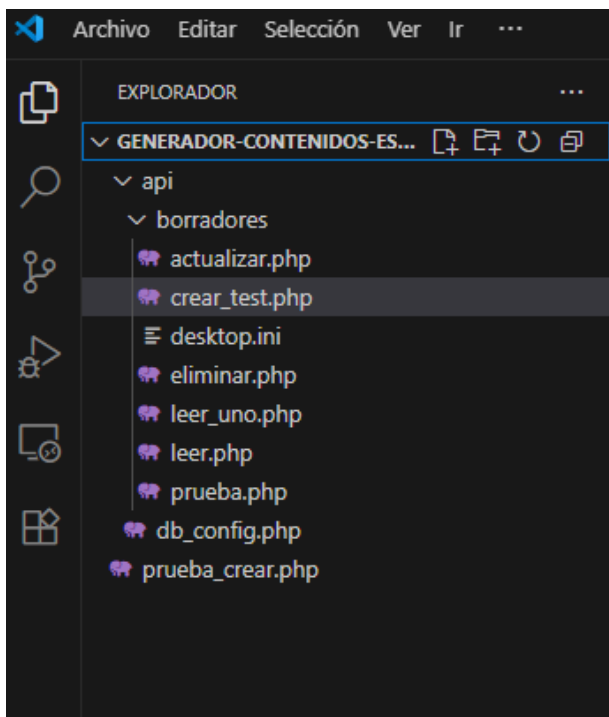
└─ leer\_uno.php

└─ actualizar.php

└─ eliminar.php

└─ config/

└─ database.php



### Ejemplo de Seguridad: Sentencia Preparada

Fragmento del archivo crear.php que usa PDO:

```
$query = "INSERT INTO borradores (titulo, contenido, tipo_contenido, tono)
        VALUES (:titulo, :contenido, :tipo_contenido, :tono)";
```

```
$stmt = $conn->prepare($query);
```

```
$stmt->bindParam(":titulo", $data->titulo);
```

```
$stmt->bindParam(":contenido", $data->contenido);
```

```
$stmt->bindParam(":tipo_contenido", $data->tipo_contenido);
```

```
$stmt->bindParam(":tono", $data->tono);
```

```
) {
    $query = "INSERT INTO borradores (titulo, contenido, tipo_contenido, tono)
            VALUES (:titulo, :contenido, :tipo_contenido, :tono)";

    $stmt = $conn->prepare($query);

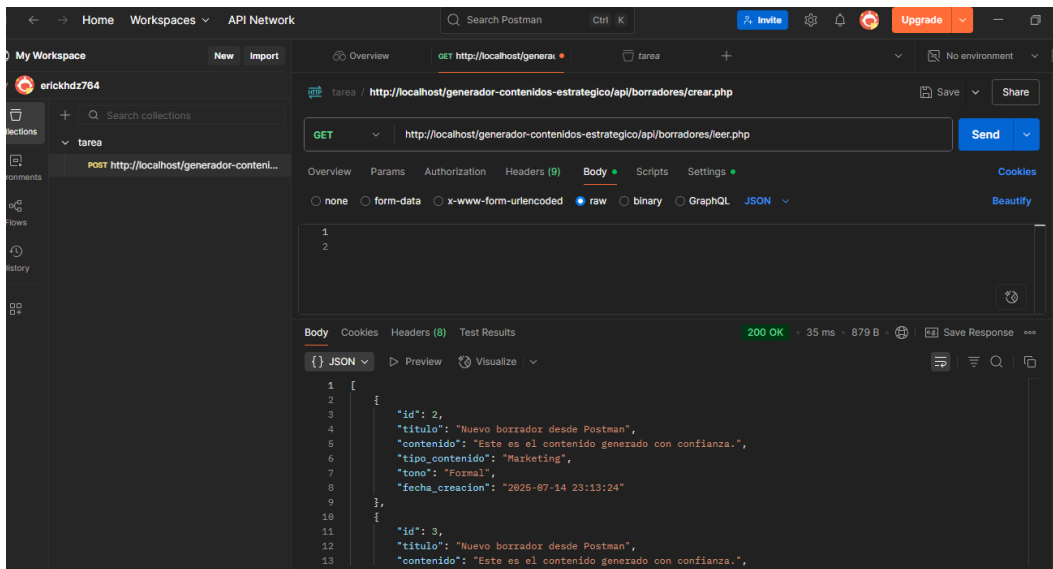
    $stmt->bindParam(":titulo", $data->titulo);
    $stmt->bindParam(":contenido", $data->contenido);
    $stmt->bindParam(":tipo_contenido", $data->tipo_contenido);
    $stmt->bindParam(":tono", $data->tono);
```

### 3. Pruebas de la API

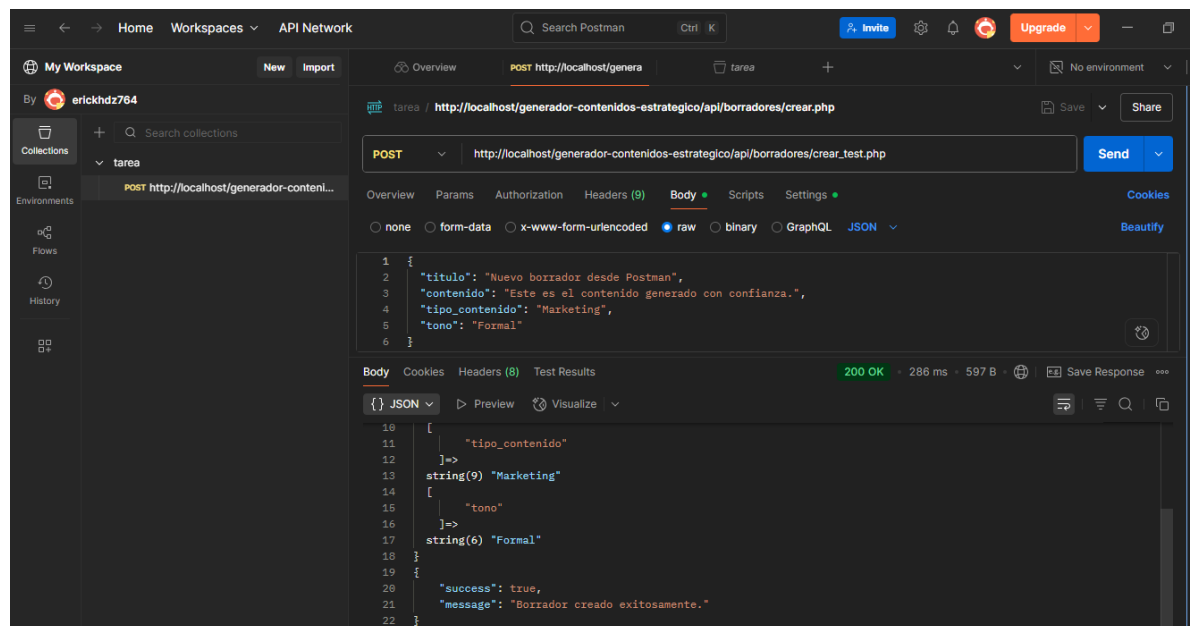
Los endpoints se probaron utilizando **Postman**. Se realizaron pruebas para verificar las funcionalidades de lectura, escritura, actualización y eliminación de datos.

Se adjuntaron capturas de pantalla mostrando los resultados de:

- Una petición **GET** exitosa que devuelve un arreglo JSON.



- Una petición **POST** con datos válidos que devuelve el mensaje de éxito: "Borrador creado exitosamente."



## **Desafíos y Soluciones**

- Error 403 Forbidden: Se resolvió verificando permisos de archivos, configuración del httpd.conf y pruebas con archivos sencillos.
- Base de datos no encontrada: Se corrigió el nombre en database.php.
- Validaciones de datos: Se corrigió la forma en que Postman enviaba los datos (JSON crudo)