



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

FUNDAMENTOS DE BASES DE DATOS - 7094

P R Á C T I C A 8

EQUIPO:

DEL MONTE ORTEGA MARYAM MICHELLE - 320083527

SOSA ROMO JUAN MARIO - 320051926

CASTILLO HERNÁNDEZ ANTONIO - 320017438

ERIK EDUARDO GÓMEZ LÓPEZ - 320258211

FECHA DE ENTREGA:
29 DE OCTUBRE DE 2024

PROFESOR:
M. EN I. GERARDO AVILÉS ROSAS

AYUDANTES:
LUIS ENRIQUE GARCÍA GÓMEZ
KEVIN JAIR TORRES VALENCIA
RICARDO BADILLO MACÍAS
ROCÍO AYLIN HUERTA GONZÁLEZ



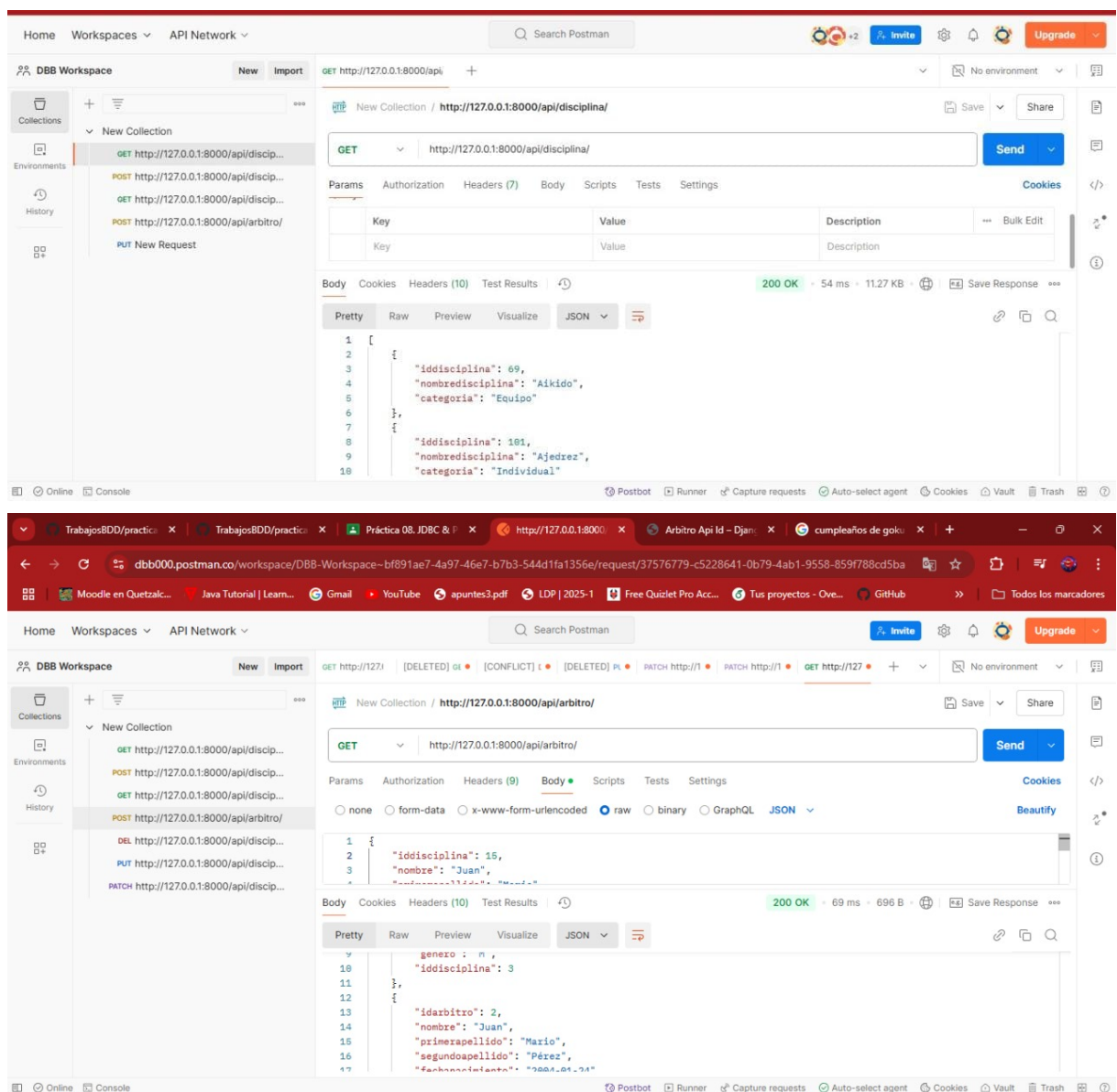
Práctica 8

Explicación del programa para operaciones CRUD

Para esta práctica, hemos escogido las tablas Árbitro y Disciplina.

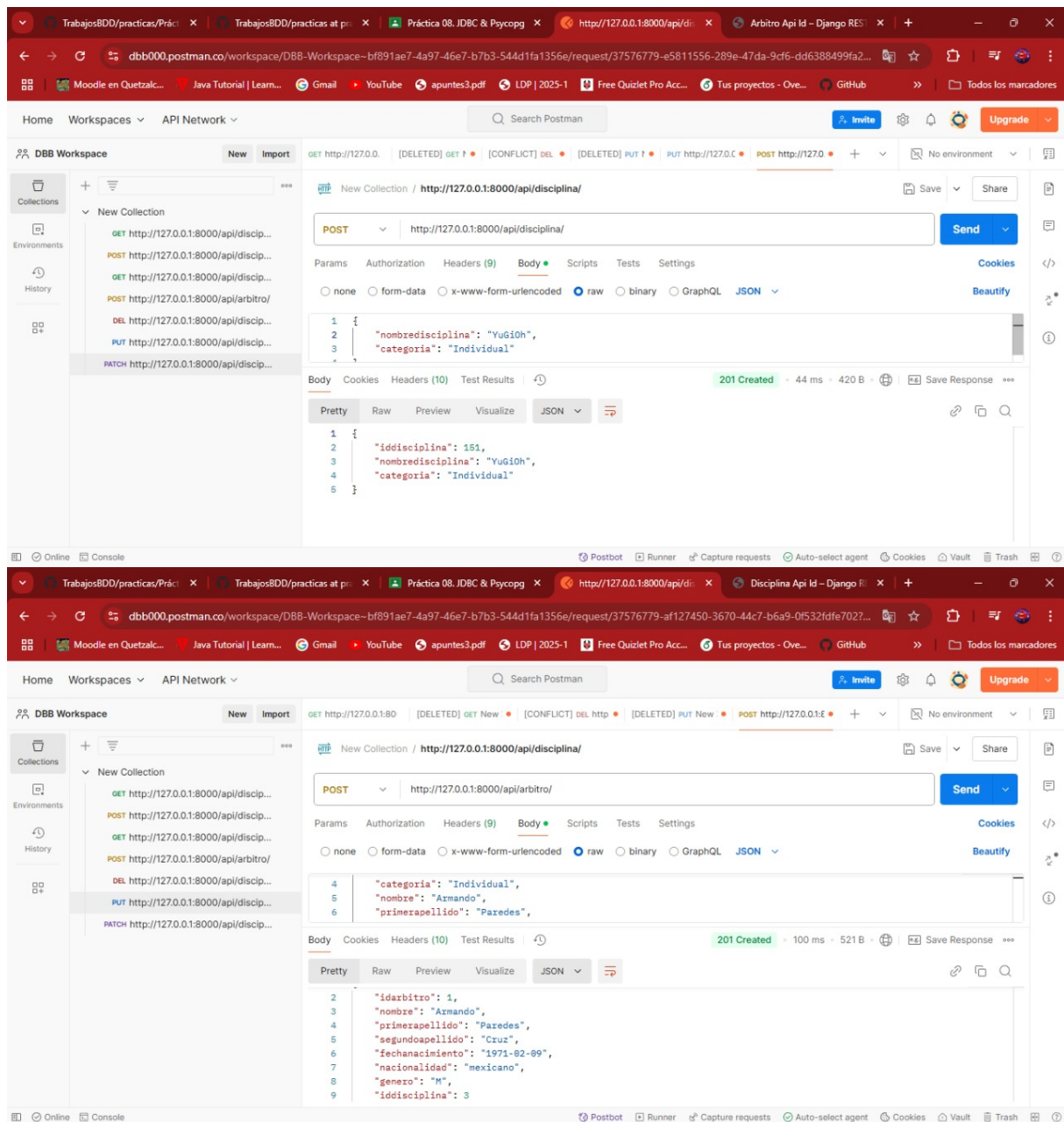
La manera en la que implementamos las operaciones CRUD para estas tablas es por medio de vistas.

- Consulta para ver todos los elementos de una tabla: Esta query la definimos con **GET**. Por ejemplo, para Disciplina
 - **Consulta:** `Disciplina.objects.all()`
 - **Descripción:** Obtiene todas las instancias del modelo Disciplina de la base de datos.
 - **Respuesta:** Devuelve un JSON con la lista de disciplinas usando el serializador `DisciplinaSerializer`.



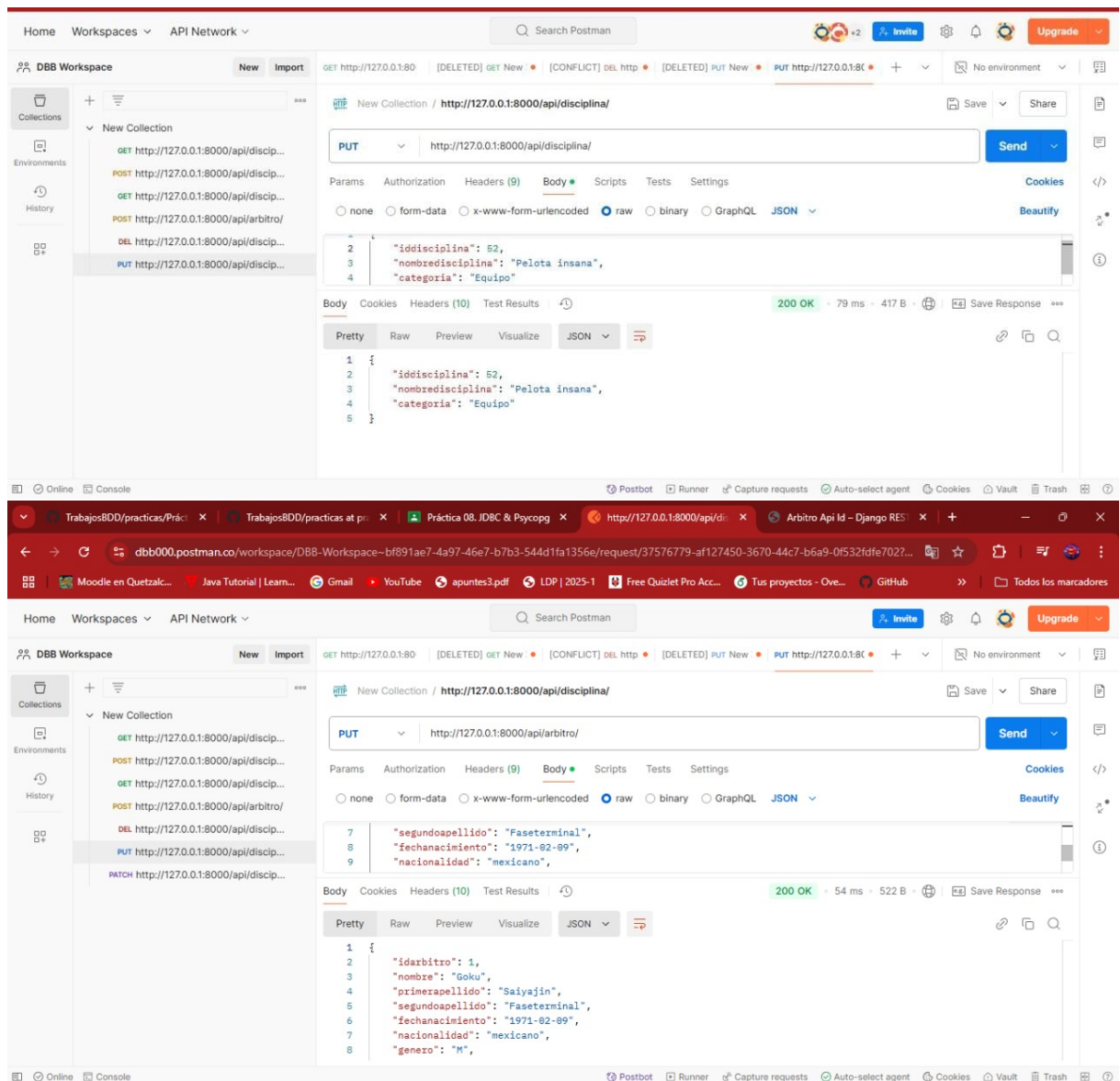
- Inserción: Realizaremos inserciones con el metodo **POST**:

- Se crea una nueva instancia.
- **Descripción:** Crea una nueva disciplina con los datos proporcionados en la solicitud.
- **Respuesta:** Devuelve la disciplina creada si los datos son válidos.



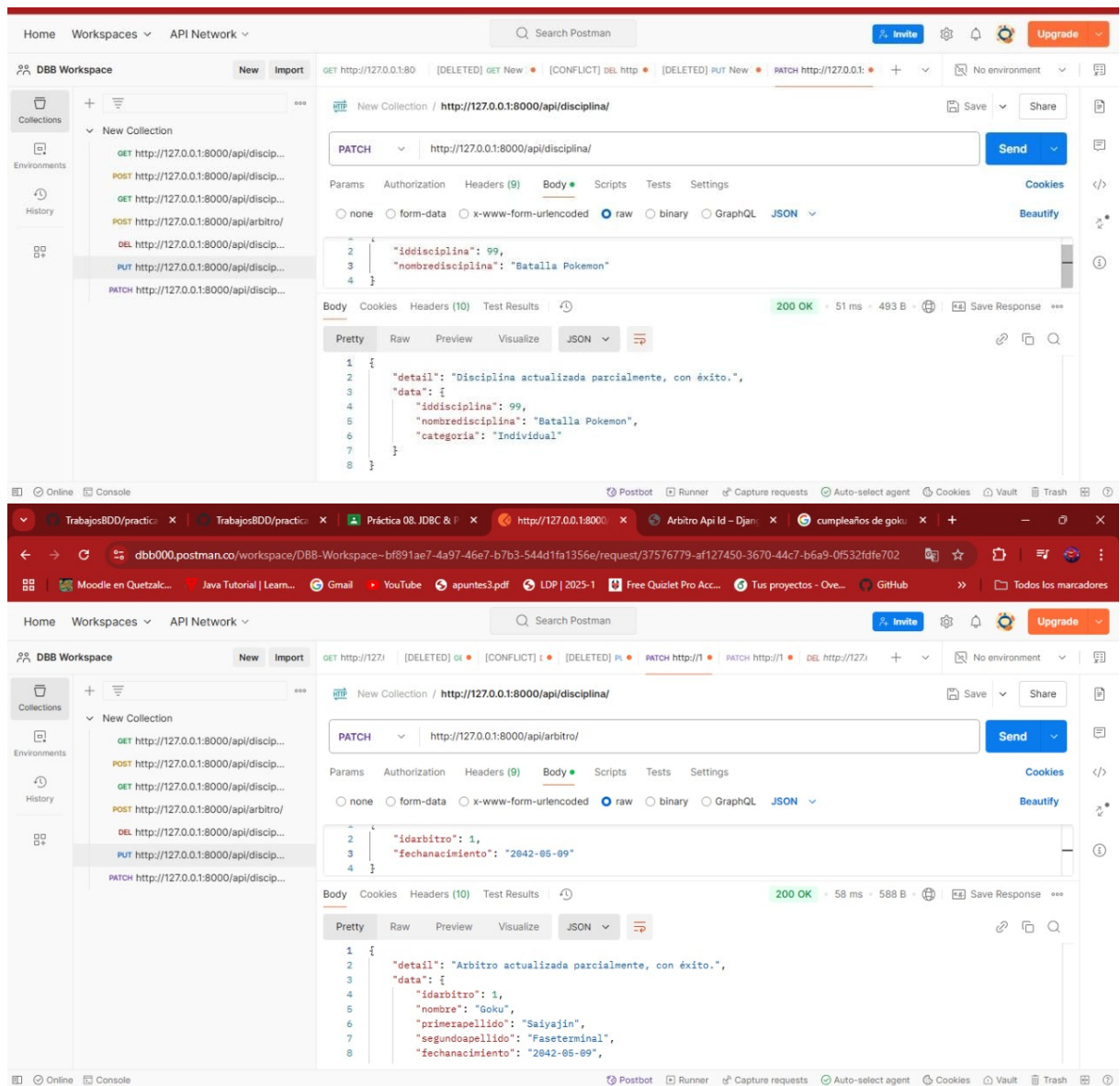
■ Actualización: La realizamos mediante **PUT**:

- **Consulta:** `Disciplina.objects.filter(iddisciplina=request.data['iddisciplina'])`
- **Descripción:** Actualiza una disciplina existente. Busca la disciplina por iddisciplina y actualiza sus datos.
- **Respuesta:** Devuelve la disciplina actualizada o un mensaje de error si no se encuentra.



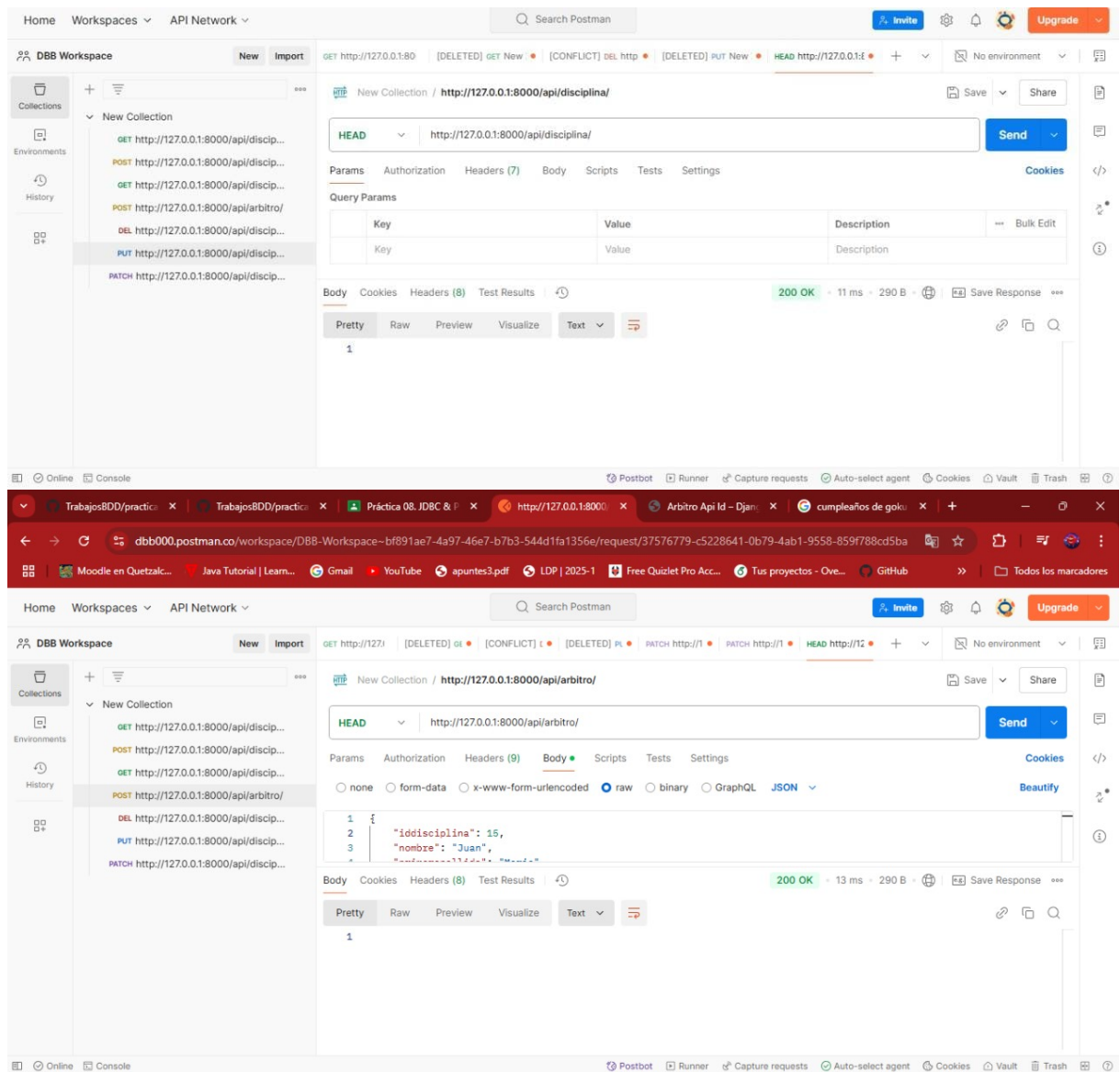
■ Método PATCH:

- **Consulta:** Similar al método put, pero permite actualizaciones parciales.
- **Descripción:** Actualiza parcialmente una disciplina existente.
- **Respuesta:** Devuelve la disciplina actualizada parcialmente o un mensaje de error si no se encuentra



■ Método **HEAD**:

- **Propósito:** Devuelve los metadatos de la solicitud.
- **Descripción:** Simplemente responde con el estado HTTP 200 para confirmar que el recurso está disponible.



■ Método DELETE:

- **Consulta:** Similar al método get, pero elimina la instancia.
- **Descripción:** Elimina una instancia específica por id.
- **Respuesta:** Devuelve un estado HTTP 204 si la eliminación es exitosa o un error si no se encuentra.

Home Workspaces API Network Search Postman

DBB Workspace New Import GET http://127.0.0.1:8000/api/ [DELETED] GET New Reque DEL http://127.0.0.1:8000/api/ + No environment

Collections + New Collection http://127.0.0.1:8000/api/disciplina/ Save Share

DELETE http://127.0.0.1:8000/api/disciplina/?iddisciplina=85 Send

Params Authorization Headers (9) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description
<input checked="" type="checkbox"/> Key		
<input checked="" type="checkbox"/> iddisciplina	85	
Key	Value	Description

Body Cookies Headers (9) Test Results 204 No Content 144 ms 307 B Save Response

Pretty Raw Preview Visualize Text 1

Online Console Postbot Runner Capture requests Auto-select agent Cookies Vault Trash

Home Workspaces API Network Search Postman

DBB Workspace New Import GET http://127/ [DELETED] GET [CONFLICT] [DELETED] PATCH http://1 PATCH http://1 DEL http://127 + No environment

Collections + New Collection http://127.0.0.1:8000/api/arbitro/ Save Share

DELETE http://127.0.0.1:8000/api/arbitro/?idarbitro=2 Send

Params Authorization Headers (9) Body Scripts Tests Settings Cookies Beautify

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```
1 {
2   "iddisciplina": 15,
3   "nombre": "Juan",
4   "apellido": "Perez"
5 }
```

Body Cookies Headers (9) Test Results 204 No Content 78 ms 307 B Save Response

Pretty Raw Preview Visualize Text 1

Online Console Postbot Runner Capture requests Auto-select agent Cookies Vault Trash