

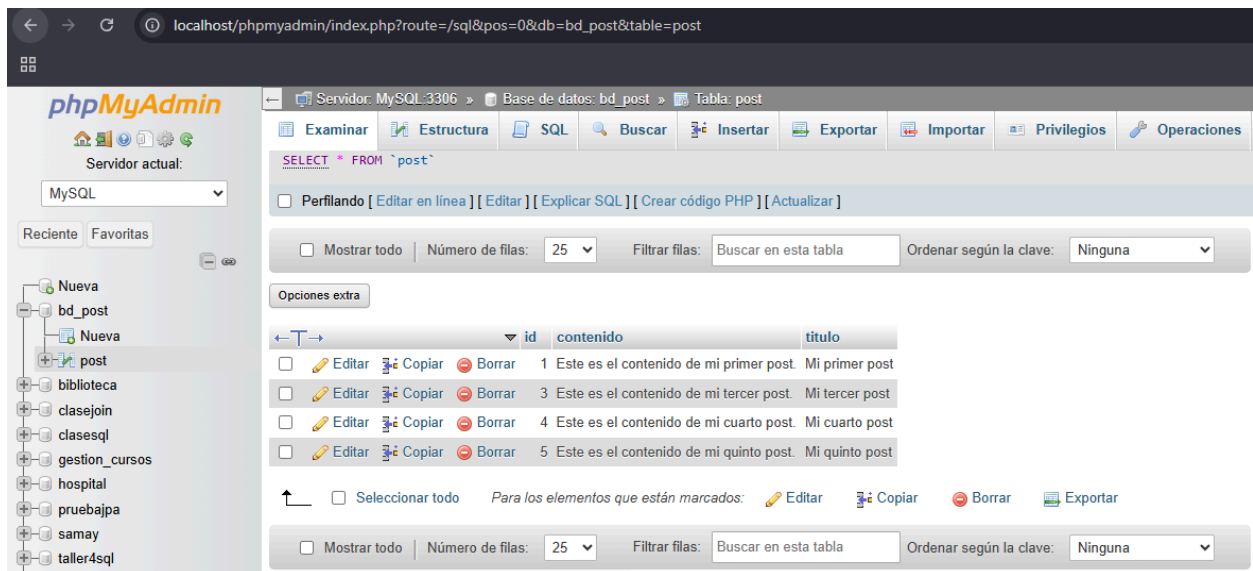
Captura de la base de datos luego de ejecutarse el script en java



The screenshot shows the phpMyAdmin interface for a MySQL server. The left sidebar displays the database structure, with 'bd\_post' selected. The main panel shows the 'Base de datos: bd\_post' view. A table named 'post' is listed with 1 row and 3 columns: 'id' (InnoDB, utf8mb4\_spanish\_ci, 16.0 KB), 'contenido' (MyISAM, utf8mb4\_spanish\_ci, 16.0 KB), and 'titulo' (MyISAM, utf8mb4\_spanish\_ci, 16.0 KB). The table is currently empty.

Tabla	Acción	Filas	Tipo	Cotejamiento	Tamaño	Residuo a depurar
post	Examinar	1	InnoDB	utf8mb4_spanish_ci	16.0 KB	-

Captura de los registros que quedaron al final, en la entidad post, tras los requests hechos en postman.



The screenshot shows the phpMyAdmin interface for the 'post' table. The left sidebar displays the database structure, with 'post' selected. The main panel shows the 'Tabla: post' view. The table contains 5 records. The records are displayed in a table with columns 'id', 'contenido', and 'titulo'. The records are: 1, 3, 4, and 5. The table is currently empty.

id	contenido	titulo
1	Este es el contenido de mi primer post.	Mi primer post
3	Este es el contenido de mi tercer post.	Mi tercer post
4	Este es el contenido de mi cuarto post.	Mi cuarto post
5	Este es el contenido de mi quinto post.	Mi quinto post

## GET BY ID

The screenshot shows the Postman interface with a workspace named "Juan's Workspace". The "API Post" collection is selected, and a "GET New Request" is highlighted. The request is configured with the method "GET" and the URL "localhost:8080/posts/3". The response is a 200 OK status with a 73 ms response time and 252 B of data. The response body is displayed in JSON format:

```
1 {
2   "id": 3,
3   "titulo": "Mi tercer post",
4   "contenido": "Este es el contenido de mi tercer post."
5 }
```

## GET ALL

The screenshot shows the Postman interface with a workspace named "Juan's Workspace". The "API Post" collection is selected, and a "GET New Request" is highlighted. The request is configured with the method "GET" and the URL "localhost:8080/posts". The response is a 200 OK status with a 46 ms response time and 521 B of data. The response body is displayed in JSON format as an array of objects:

```
1 [
2   {
3     "id": 1,
4     "titulo": "Mi primer post",
5     "contenido": "Este es el contenido de mi primer post."
6   },
7   {
8     "id": 3,
9     "titulo": "Mi tercer post",
10    "contenido": "Este es el contenido de mi tercer post."
11  },
12  {
13    "id": 4,
14    "titulo": "Mi cuarto post",
15    "contenido": "Este es el contenido de mi cuarto post."
16  },
17  {
18    "id": 5,
19    "titulo": "Mi quinto post",
20    "contenido": "Este es el contenido de mi quinto post."
21  }
22 ]
```

A notification bubble in the bottom left corner states: "You've made new changes. Post an update to inform others."

## DELETE BY ID

The screenshot shows the Postman application interface. At the top, there's a navigation bar with 'Home', 'Workspaces', and 'API Network'. Below this, the 'Juan's Workspace' is selected, showing a list of requests under the 'API Post' collection. The 'DEL New Request' is highlighted. The main panel shows the configuration for this request: the method is 'DELETE' and the URL is 'localhost:8080/posts/2'. The 'Params' tab is active, showing a table for 'Query Params' with columns 'Key' and 'Value'. The 'Body' tab is also visible, showing a status of '200 OK'. The bottom bar includes options for 'Raw', 'Preview', and 'Visualize'.

Home Workspaces API Network

Search Postman Ctrl K

Juan's Workspace New Import

API Post GET Untitled Requ GET New Reques POST New Reque: GET New Reques DEL N

API Post / New Request

DELETE localhost:8080/posts/2

Params Authorization Headers (7) Body Scripts Settings

Query Params

Key	Value
Key	Value

Body Cookies Headers (4) Test Results 200 OK

Raw Preview Visualize