

TECNOLÓGICO NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO DEL VALLE DE
OAXACA



INGENIERÍA EN TECNOLOGÍAS DE LA
INFORMACIÓN Y COMUNICACIÓN

APLICACIÓN CON CONEXIÓN A BASE DE DATOS

TALLER DE BASE DE DATOS

ANDREA ITZEL JIMÉNEZ PÉREZ

IT4A

INSTALACIÓN Y CONFIGURACIÓN DE DRIVERS

ORACLE CON JAVA

OJDBC11.JAR

Conectarse a una base de datos Oracle

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection { 1usage
    private static final String URL = "jdbc:oracle:thin:@//localhost:1521/XE";
    private static final String USER = "co"; 1usage
    private static final String PASSWORD = " "; 1usage

    public static Connection getConnection() throws SQLException { 1usage
        return DriverManager.getConnection(URL, USER, PASSWORD);
    }
}
```

Fig. 1

Nota:

Configurar la URL de conexión: La URL puede variar dependiendo de tu entorno. Ejemplos:

Para Oracle XE: jdbc:oracle:thin:@localhost:1521:xe

Para un servicio en red:

jdbc:oracle:thin:@//host:puerto/nombre_servicio

Se agrega el driver JDBC al proyecto, incluye el archivo ojdbc11.jar en el classpath, donde se agrega la dependencia al archivo pom.xml

```
<dependency>
  <groupId>com.oracle.database.jdbc</groupId>
  <artifactId>ojdbc11</artifactId>
  <version>21.9.0.0</version> <!-- Compatible con Oracle 21c -->
</dependency>
```

Fig. 2

COMPROBACIÓN DEL CRUD

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Pro
=== Añadir cliente ===
Cliente añadido: Andrea Jiménez

=== Buscar cliente por ID ===
Cliente encontrado: Andrea Jiménez | Email: AndreaJ@test.com

=== Actualizar cliente ===
Cliente actualizado: Andrea J
```

Fig. 3

```
ID: 392 | Nombre: Adam Miller | Email: adam.miller@internalmail
ID: 1001 | Nombre: email@test.com | Email: Nombre
ID: 1002 | Nombre: Andrea J | Email: Andrea@test.com
ID: 1 | Nombre: Tammy Bryant | Email: tammy.bryant@internalmail
ID: 2 | Nombre: Roy White | Email: roy.white@internalmail
```

Fig. 4

```
=== Eliminar cliente ===
Cliente eliminado (ID: 1002)
```

Fig. 5

Una vez eliminado el cliente verificamos en la lista

```
ID: 392 | Nombre: Adam Miller | Email: adam.miller@internalmail
ID: 1001 | Nombre: email@test.com | Email: Nombre
ID: 1 | Nombre: Tammy Bryant | Email: tammy.bryant@internalmail
ID: 2 | Nombre: Roy White | Email: roy.white@internalmail
```

Fig. 6

JAVASCRIPT CON POSTGRES

BUN

Comando para instalar en windows

```
powershell -c "irm bun.sh/install.ps1|iex"
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\andre> powershell -c "irm bun.sh/install.ps1|iex"
##### 100.0%
Bun 1.2.14 was installed successfully!
The binary is located at C:\Users\andre\.bun\bin\bun.exe

To get started, restart your terminal/editor, then type "bun"
```

Fig. 7

Comprobar que Bun se instaló correctamente

```
PS C:\Users\andre> bun --version
1.2.14
```

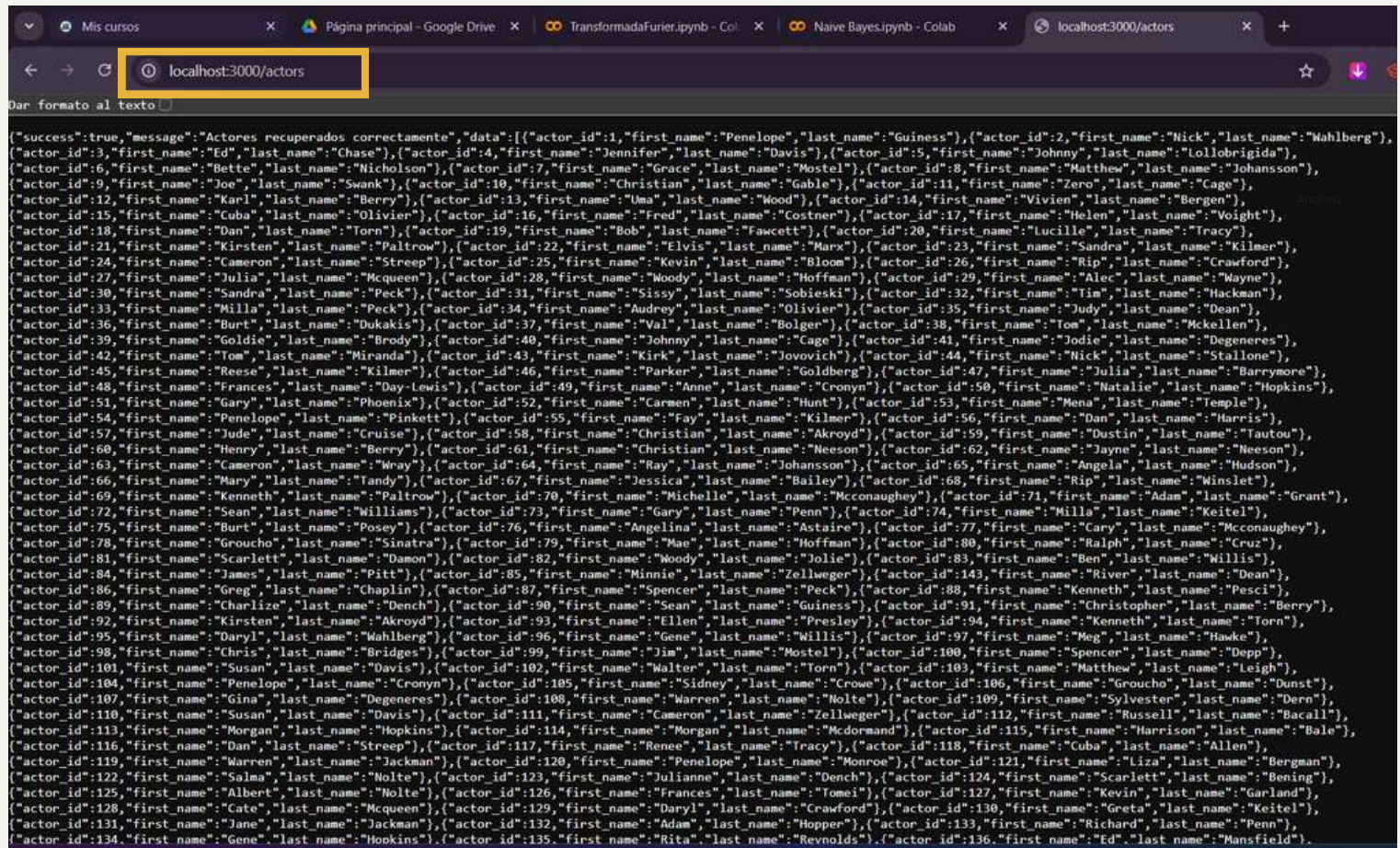
Fig. 8

Configurar Bun como tu runtime por defecto
para scripts o ejecutar archivos directamente

```
C:\Users\andre\Documents\GitHub\javascript_postgres>bun run server.ts
🚀 Servidor escuchando en http://localhost:3000
```

Fig. 9

Poner en un navegador la dirección que nos apareció más el nombre de la tabla con los datos que deseemos ver



```
{
  "success": true,
  "message": "Actores recuperados correctamente",
  "data": [
    {
      "actor_id": 1,
      "first_name": "Penelope",
      "last_name": "Guinness"
    },
    {
      "actor_id": 2,
      "first_name": "Nick",
      "last_name": "Wahlberg"
    },
    {
      "actor_id": 3,
      "first_name": "Ed",
      "last_name": "Chase"
    },
    {
      "actor_id": 4,
      "first_name": "Jennifer",
      "last_name": "Davis"
    },
    {
      "actor_id": 5,
      "first_name": "Johnny",
      "last_name": "Lollobrigida"
    },
    {
      "actor_id": 6,
      "first_name": "Bette",
      "last_name": "Nicholson"
    },
    {
      "actor_id": 7,
      "first_name": "Grace",
      "last_name": "Mostel"
    },
    {
      "actor_id": 8,
      "first_name": "Matthew",
      "last_name": "Johansson"
    },
    {
      "actor_id": 9,
      "first_name": "Joe",
      "last_name": "Swank"
    },
    {
      "actor_id": 10,
      "first_name": "Christian",
      "last_name": "Gable"
    },
    {
      "actor_id": 11,
      "first_name": "Zero",
      "last_name": "Cage"
    },
    {
      "actor_id": 12,
      "first_name": "Karl",
      "last_name": "Berry"
    },
    {
      "actor_id": 13,
      "first_name": "Uma",
      "last_name": "Wood"
    },
    {
      "actor_id": 14,
      "first_name": "Vivien",
      "last_name": "Bergen"
    },
    {
      "actor_id": 15,
      "first_name": "Cuba",
      "last_name": "Olivier"
    },
    {
      "actor_id": 16,
      "first_name": "Fred",
      "last_name": "Costner"
    },
    {
      "actor_id": 17,
      "first_name": "Helen",
      "last_name": "Voight"
    },
    {
      "actor_id": 18,
      "first_name": "Dan",
      "last_name": "Torn"
    },
    {
      "actor_id": 19,
      "first_name": "Bob",
      "last_name": "Fawcett"
    },
    {
      "actor_id": 20,
      "first_name": "Lucille",
      "last_name": "Tracy"
    },
    {
      "actor_id": 21,
      "first_name": "Kirsten",
      "last_name": "Paltrow"
    },
    {
      "actor_id": 22,
      "first_name": "Elvis",
      "last_name": "Marx"
    },
    {
      "actor_id": 23,
      "first_name": "Sandra",
      "last_name": "Kilmer"
    },
    {
      "actor_id": 24,
      "first_name": "Cameron",
      "last_name": "Streep"
    },
    {
      "actor_id": 25,
      "first_name": "Kevin",
      "last_name": "Bloom"
    },
    {
      "actor_id": 26,
      "first_name": "Rip",
      "last_name": "Crawford"
    },
    {
      "actor_id": 27,
      "first_name": "Julia",
      "last_name": "Mcqueen"
    },
    {
      "actor_id": 28,
      "first_name": "Woody",
      "last_name": "Hoffman"
    },
    {
      "actor_id": 29,
      "first_name": "Alec",
      "last_name": "Wayne"
    },
    {
      "actor_id": 30,
      "first_name": "Sandra",
      "last_name": "Peck"
    },
    {
      "actor_id": 31,
      "first_name": "Sissy",
      "last_name": "Sobieski"
    },
    {
      "actor_id": 32,
      "first_name": "Tim",
      "last_name": "Hackman"
    },
    {
      "actor_id": 33,
      "first_name": "Milla",
      "last_name": "Audrey"
    },
    {
      "actor_id": 34,
      "first_name": "Audrey",
      "last_name": "Olivier"
    },
    {
      "actor_id": 35,
      "first_name": "Judy",
      "last_name": "Dean"
    },
    {
      "actor_id": 36,
      "first_name": "Burt",
      "last_name": "Dukakis"
    },
    {
      "actor_id": 37,
      "first_name": "Val",
      "last_name": "Bolger"
    },
    {
      "actor_id": 38,
      "first_name": "Tom",
      "last_name": "McKellen"
    },
    {
      "actor_id": 39,
      "first_name": "Goldie",
      "last_name": "Brody"
    },
    {
      "actor_id": 40,
      "first_name": "Johnny",
      "last_name": "Cage"
    },
    {
      "actor_id": 41,
      "first_name": "Todie",
      "last_name": "Degeneres"
    },
    {
      "actor_id": 42,
      "first_name": "Tom",
      "last_name": "Miranda"
    },
    {
      "actor_id": 43,
      "first_name": "Kirk",
      "last_name": "Jovovich"
    },
    {
      "actor_id": 44,
      "first_name": "Nick",
      "last_name": "Stallone"
    },
    {
      "actor_id": 45,
      "first_name": "Reese",
      "last_name": "Kilmer"
    },
    {
      "actor_id": 46,
      "first_name": "Parker",
      "last_name": "Goldberg"
    },
    {
      "actor_id": 47,
      "first_name": "Julia",
      "last_name": "Barrymore"
    },
    {
      "actor_id": 48,
      "first_name": "Frances",
      "last_name": "Day-Lewis"
    },
    {
      "actor_id": 49,
      "first_name": "Anne",
      "last_name": "Cronyn"
    },
    {
      "actor_id": 50,
      "first_name": "Natalie",
      "last_name": "Hopkins"
    },
    {
      "actor_id": 51,
      "first_name": "Gary",
      "last_name": "Phoenix"
    },
    {
      "actor_id": 52,
      "first_name": "Carmen",
      "last_name": "Hunt"
    },
    {
      "actor_id": 53,
      "first_name": "Mena",
      "last_name": "Temple"
    },
    {
      "actor_id": 54,
      "first_name": "Penelope",
      "last_name": "Pinkett"
    },
    {
      "actor_id": 55,
      "first_name": "Fay",
      "last_name": "Kilmer"
    },
    {
      "actor_id": 56,
      "first_name": "Dan",
      "last_name": "Harris"
    },
    {
      "actor_id": 57,
      "first_name": "Jude",
      "last_name": "Cruise"
    },
    {
      "actor_id": 58,
      "first_name": "Christian",
      "last_name": "Akroyd"
    },
    {
      "actor_id": 59,
      "first_name": "Dustin",
      "last_name": "Tautou"
    },
    {
      "actor_id": 60,
      "first_name": "Henry",
      "last_name": "Berry"
    },
    {
      "actor_id": 61,
      "first_name": "Christian",
      "last_name": "Neeson"
    },
    {
      "actor_id": 62,
      "first_name": "Jayne",
      "last_name": "Neeson"
    },
    {
      "actor_id": 63,
      "first_name": "Cameron",
      "last_name": "Wray"
    },
    {
      "actor_id": 64,
      "first_name": "Ray",
      "last_name": "Johansson"
    },
    {
      "actor_id": 65,
      "first_name": "Angela",
      "last_name": "Hudson"
    },
    {
      "actor_id": 66,
      "first_name": "Mary",
      "last_name": "Tandy"
    },
    {
      "actor_id": 67,
      "first_name": "Jessica",
      "last_name": "Bailey"
    },
    {
      "actor_id": 68,
      "first_name": "Rip",
      "last_name": "Winslet"
    },
    {
      "actor_id": 69,
      "first_name": "Kenneth",
      "last_name": "Paltrow"
    },
    {
      "actor_id": 70,
      "first_name": "Michelle",
      "last_name": "McConaughey"
    },
    {
      "actor_id": 71,
      "first_name": "Adam",
      "last_name": "Grant"
    },
    {
      "actor_id": 72,
      "first_name": "Sean",
      "last_name": "Williams"
    },
    {
      "actor_id": 73,
      "first_name": "Gary",
      "last_name": "Penn"
    },
    {
      "actor_id": 74,
      "first_name": "Milla",
      "last_name": "Keitel"
    },
    {
      "actor_id": 75,
      "first_name": "Burt",
      "last_name": "Posey"
    },
    {
      "actor_id": 76,
      "first_name": "Angelina",
      "last_name": "Astaire"
    },
    {
      "actor_id": 77,
      "first_name": "Cary",
      "last_name": "McConaughey"
    },
    {
      "actor_id": 78,
      "first_name": "Groucho",
      "last_name": "Sinatra"
    },
    {
      "actor_id": 79,
      "first_name": "Mae",
      "last_name": "Hoffman"
    },
    {
      "actor_id": 80,
      "first_name": "Ralph",
      "last_name": "Cruz"
    },
    {
      "actor_id": 81,
      "first_name": "Scarlett",
      "last_name": "Damon"
    },
    {
      "actor_id": 82,
      "first_name": "Woody",
      "last_name": "Jolie"
    },
    {
      "actor_id": 83,
      "first_name": "Ben",
      "last_name": "Willis"
    },
    {
      "actor_id": 84,
      "first_name": "James",
      "last_name": "Pitt"
    },
    {
      "actor_id": 85,
      "first_name": "Minnie",
      "last_name": "Zellweger"
    },
    {
      "actor_id": 86,
      "first_name": "Greg",
      "last_name": "Chaplin"
    },
    {
      "actor_id": 87,
      "first_name": "Spencer",
      "last_name": "Peck"
    },
    {
      "actor_id": 88,
      "first_name": "Kenneth",
      "last_name": "Pesci"
    },
    {
      "actor_id": 89,
      "first_name": "Charlize",
      "last_name": "Dench"
    },
    {
      "actor_id": 90,
      "first_name": "Sean",
      "last_name": "Guinness"
    },
    {
      "actor_id": 91,
      "first_name": "Christopher",
      "last_name": "Berry"
    },
    {
      "actor_id": 92,
      "first_name": "Daryl",
      "last_name": "Wahlberg"
    },
    {
      "actor_id": 93,
      "first_name": "Ellen",
      "last_name": "Presley"
    },
    {
      "actor_id": 94,
      "first_name": "Kenneth",
      "last_name": "Torn"
    },
    {
      "actor_id": 95,
      "first_name": "Daryl",
      "last_name": "Wahlberg"
    },
    {
      "actor_id": 96,
      "first_name": "Gene",
      "last_name": "Willis"
    },
    {
      "actor_id": 97,
      "first_name": "Meg",
      "last_name": "Hawke"
    },
    {
      "actor_id": 98,
      "first_name": "Chris",
      "last_name": "Bridges"
    },
    {
      "actor_id": 99,
      "first_name": "Jim",
      "last_name": "Mostel"
    },
    {
      "actor_id": 100,
      "first_name": "Spencer",
      "last_name": "Dopp"
    },
    {
      "actor_id": 101,
      "first_name": "Susan",
      "last_name": "Davis"
    },
    {
      "actor_id": 102,
      "first_name": "Walter",
      "last_name": "Torn"
    },
    {
      "actor_id": 103,
      "first_name": "Matthew",
      "last_name": "Leigh"
    },
    {
      "actor_id": 104,
      "first_name": "Penelope",
      "last_name": "Cronyn"
    },
    {
      "actor_id": 105,
      "first_name": "Sidney",
      "last_name": "Crowe"
    },
    {
      "actor_id": 106,
      "first_name": "Groucho",
      "last_name": "Dunst"
    },
    {
      "actor_id": 107,
      "first_name": "Gina",
      "last_name": "Degeneres"
    },
    {
      "actor_id": 108,
      "first_name": "Warren",
      "last_name": "Nolte"
    },
    {
      "actor_id": 109,
      "first_name": "Sylvester",
      "last_name": "Dern"
    },
    {
      "actor_id": 110,
      "first_name": "Susan",
      "last_name": "Davis"
    },
    {
      "actor_id": 111,
      "first_name": "Cameron",
      "last_name": "Zellweger"
    },
    {
      "actor_id": 112,
      "first_name": "Russell",
      "last_name": "Bacall"
    },
    {
      "actor_id": 113,
      "first_name": "Morgan",
      "last_name": "Hopkins"
    },
    {
      "actor_id": 114,
      "first_name": "Morgan",
      "last_name": "McDormand"
    },
    {
      "actor_id": 115,
      "first_name": "Harrison",
      "last_name": "Bale"
    },
    {
      "actor_id": 116,
      "first_name": "Dan",
      "last_name": "Streep"
    },
    {
      "actor_id": 117,
      "first_name": "Renee",
      "last_name": "Tracy"
    },
    {
      "actor_id": 118,
      "first_name": "Cuba",
      "last_name": "Allen"
    },
    {
      "actor_id": 119,
      "first_name": "Warren",
      "last_name": "Jackman"
    },
    {
      "actor_id": 120,
      "first_name": "Penelope",
      "last_name": "Monroe"
    },
    {
      "actor_id": 121,
      "first_name": "Liza",
      "last_name": "Bergman"
    },
    {
      "actor_id": 122,
      "first_name": "Salma",
      "last_name": "Nolte"
    },
    {
      "actor_id": 123,
      "first_name": "Julianne",
      "last_name": "Dench"
    },
    {
      "actor_id": 124,
      "first_name": "Scarlett",
      "last_name": "Bening"
    },
    {
      "actor_id": 125,
      "first_name": "Albert",
      "last_name": "Nolte"
    },
    {
      "actor_id": 126,
      "first_name": "Frances",
      "last_name": "Tomei"
    },
    {
      "actor_id": 127,
      "first_name": "Kevin",
      "last_name": "Garland"
    },
    {
      "actor_id": 128,
      "first_name": "Cate",
      "last_name": "Mcqueen"
    },
    {
      "actor_id": 129,
      "first_name": "Daryl",
      "last_name": "Crawford"
    },
    {
      "actor_id": 130,
      "first_name": "Greta",
      "last_name": "Keitel"
    },
    {
      "actor_id": 131,
      "first_name": "Jane",
      "last_name": "Jackman"
    },
    {
      "actor_id": 132,
      "first_name": "Adam",
      "last_name": "Hopper"
    },
    {
      "actor_id": 133,
      "first_name": "Richard",
      "last_name": "Penn"
    },
    {
      "actor_id": 134,
      "first_name": "Gene",
      "last_name": "Hookins"
    },
    {
      "actor_id": 135,
      "first_name": "Rita",
      "last_name": "Reynolds"
    },
    {
      "actor_id": 136,
      "first_name": "Ed",
      "last_name": "Mansfield"
    }
  ]
}
```

Fig. 10

Nota:

En este caso se ocupó la base de datos de muestra de PostgreSQL llamado dvdRental

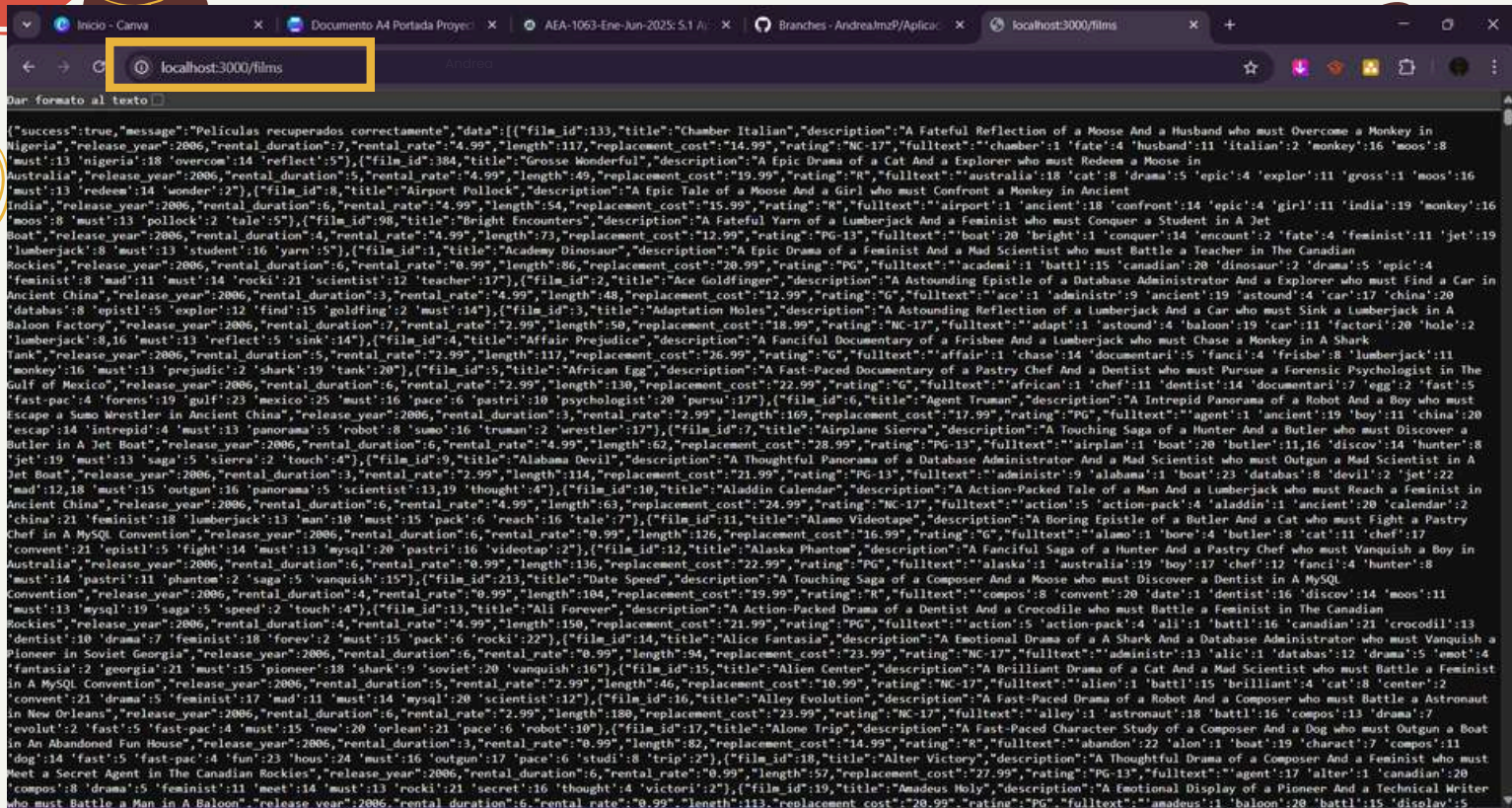


Fig. 11

COMPROBACIÓN DEL CRUD

Primero corremos el servidor

```
C:\Users\andre\Documents\GitHub\javascript_postgres>bun server.ts
Servidor escuchando en http://localhost:3000
```

Fig. 12

Agregamos un actor

```
PS C:\Users\andre\Documents\GitHub\javascript_postgres> curl.exe -X POST http://localhost:3000/actors
>> -H "Content-Type: application/json"
>> -d '{"first_name": "Tom", "last_name": "Hanks"}'
{"success":true,"message":"Recurso creado","data":{"actor_id":201,"first_name":"Tom","last_name":"Hanks"}}
```

Fig. 13

Verificamos que se creo el actor

```
{
  "actor_id": 177, "first_name": "Gene", "last_name": "Mckellen",
  "actor_id": 180, "first_name": "Jeff", "last_name": "Silverstone",
  "actor_id": 183, "first_name": "Russell", "last_name": "Close",
  "actor_id": 186, "first_name": "Julia", "last_name": "Zellweger",
  "actor_id": 189, "first_name": "Cuba", "last_name": "Birch",
  "actor_id": 192, "first_name": "John", "last_name": "Suvari",
  "actor_id": 195, "first_name": "Jayne", "last_name": "Silverstone",
  "actor_id": 198, "first_name": "Mary", "last_name": "Keitel",
  "actor_id": 201, "first_name": "Tom", "last_name": "Hanks"
}
```

Fig. 14

Buscamos un actor

```
localhost:3000/actors/1
Dar formato al texto
{"success":true,"message":"Operación exitosa","data":{"actor_id":1,"first_name":"Penelope","last_name":"Guinness"}}
```

Fig. 15

Actualizamos un actor

```
PS C:\Users\andre\Documents\GitHub\javascript_postgres> curl.exe -X PUT http://localhost:3000/actors/1
>> -H "Content-Type: application/json"
>> -d '{"last_name": "Cruise"}'
{"success":true,"message":"Operación exitosa","data":{"actor_id":1,"first_name":"Penelope","last_name":"Cruise"}}
```

Fig. 16

```
localhost:3000/actors/1
Dar formato al texto
{"success":true,"message":"Operación exitosa","data":{"actor_id":1,"first_name":"Penelope","last_name":"Cruise"}}
```

Fig. 17

Eliminamos un actor

```
PS C:\Users\andre\Documents\GitHub\javascript_postgres> curl.exe -X DELETE http://localhost:3000/actors/201
>>
{"success":true,"message":"Operación exitosa","data":{"actor_id":201,"first_name":"Tom","last_name":"Hanks"}}
```

Fig. 18

```
localhost:3000/actors/201
Dar formato al texto
{"success":false,"message":"Actor no encontrado","data":[]}
```

Fig. 19

Agregamos una película

```
PS C:\Users\andre\Documents\GitHub\javascript_postgres> curl.exe -X POST http://localhost:3000/films
>> -H "Content-Type: application/json"
>> -d '{"title": "thunderbolts","description": "Newavengers","release_year": 2025,"language_id": 1}'
{"success":true,"message":"Recurso creado","data":{"film_id":1010,"title":"thunderbolts","description":"Newavengers","release_year":2025,"language_id":1,"rental_rate":4.99,"length":null,"replacement_cost":19.99,"rating":6,"fulltext":"'newaveng':2 'thunderbolt':1"}}
```

Fig. 20

Verificamos que se creo la película

```
{"success":true,"message":"Película encontrada","data":[{"film_id":1010,"title":"thunderbolts","description":"Newavengers","release_year":2025,"language_id":1,"rental_duration":3,"rental_rate":4.99,"length":null,"newaveng":2 'thunderbolt':1}]}
```

Fig. 21

Buscamos una película

```
localhost:3000/films/1
Dar formato al texto
{"success":true,"message":"Película encontrada","data":[{"film_id":1,"title":"Academy Dinosaur","description":"A Epic Drama of a Feminist And a Mad Scientist who must Battle a Teacher in The Rockies","release_year":2006,"rental_duration":6,"rental_rate":0.99,"length":86,"replacement_cost":20.99,"rating":"PG","fulltext":"'academi':1 'battl':15 'canadian':20 'dinosaur':2 'dra... 'feminist':8 'mad':11 'must':14 'rocki':21 'scientist':12 'teacher':17"}]}
```

Fig. 22

Actualizamos una película

```
PS C:\Users\andre\Documents\GitHub\javascript_postgres> curl.exe -X PUT http://localhost:3000/films/1010
>> -H "Content-Type: application/json"
>> -d '{"title": "New Avengers","description": "Newavengers","release_year": 2025,"language_id": 1}'
{"success":true,"message":"Película actualizada","data":{"film_id":1010,"title":"New Avengers","description":"Newavengers","release_year":2025,"language_id":1,"rental_rate":4.99,"length":null,"replacement_cost":19.99,"rating":6,"fulltext":"'aveng':2 'new':1 'newaveng':3"}}
```

Fig. 23

```
localhost:3000/films/1010
Dar formato al texto
{"success":true,"message":"Película encontrada","data":[{"film_id":1010,"title":"New Avengers","description":"Newavengers","release_year":2025,"language_id":1,"rental_duration":3,"rental_rate":4.99,"length":null,"replacement_cost":19.99,"rating":6,"fulltext":"'aveng':2 'new':1 'newaveng':3"}]}
```

Fig. 24

Eliminamos una película

```
PS C:\Users\andre\Documents\GitHub\javascript_postgres> curl.exe -X DELETE http://localhost:3000/films/1010
>>
{"success":true,"message":"Película eliminada","data":{"film_id":1010,"title":"New Avengers","description":"Newavengers","release_year":2025,"language_id":1,"rental_rate":4.99,"length":null,"replacement_cost":19.99,"rating":6,"fulltext":"'aveng':2 'new':1 'newaveng':3"}}
```

Fig. 25

Comprobamos que se eliminó la película

```
localhost:3000/films/1010
Dar formato al texto
{"success":false,"message":"Película no encontrada","data":[]}
```

Fig. 26

PHP CON MARIADB

CONFIGURACIÓN

Comando para ejecutar el archivo directamente

```
php --server localhost:8000 --docroot public  
router.php
```

```
PS C:\Users\andre\Documents\evaldocente-main> php --server localhost:8000 --docroot public router.php  
[Fri May 30 10:39:30 2025] PHP 8.4.7 Development Server (http://localhost:8000) started
```

Fig. 27



Fig. 28

PHP CON MARIADB

COMPROBACIÓN DEL CRUD



The screenshot shows a web form titled "Encuesta de Evaluación Docente". It contains the following fields:

- A text input field labeled "Nombre del docente:".
- A dropdown menu labeled "1. ¿Explica claramente los temas?" with the placeholder text "Seleccione".
- A dropdown menu labeled "2. ¿Fomenta la participación?" with the placeholder text "Seleccione".
- A dropdown menu labeled "3. ¿Es puntual y responsable?" with the placeholder text "Seleccione".
- A text area labeled "Comentarios adicionales:".
- A blue button at the bottom labeled "Enviar evaluación".

Por medio de la encuesta agregamos una evaluación

Fig. 29

Comprobamos que se agregaron nuestras evaluaciones



The screenshot shows a dropdown menu with the label "Docente:". The menu is open, displaying a list of names:

- Ambrosio Cardos
- Ambrosio Cardos
- Rodolfo
- Carlos

Fig. 30

Actualizamos una encuesta



POST

Params Authorization Headers **Body**

none form-data x-www-form-urlencoded raw

+ - ☒

Key	Value	Description
<input checked="" type="checkbox"/> id	1	
<input checked="" type="checkbox"/> docente	Ambrosio Cardoso	
<input checked="" type="checkbox"/> pregunta1	3	
<input checked="" type="checkbox"/> pregunta2	3	
<input checked="" type="checkbox"/> pregunta3	2	
<input checked="" type="checkbox"/> comentarios	:)	

Fig. 31



Fig. 32

Comprobamos que se actualizó (ahora está bien escrito)



Docente:

Ambrosio Cardoso ▼

Ambrosio Cardoso

Rodolfo

Carlos

Fig. 33

Eliminamos uno de la encuesta

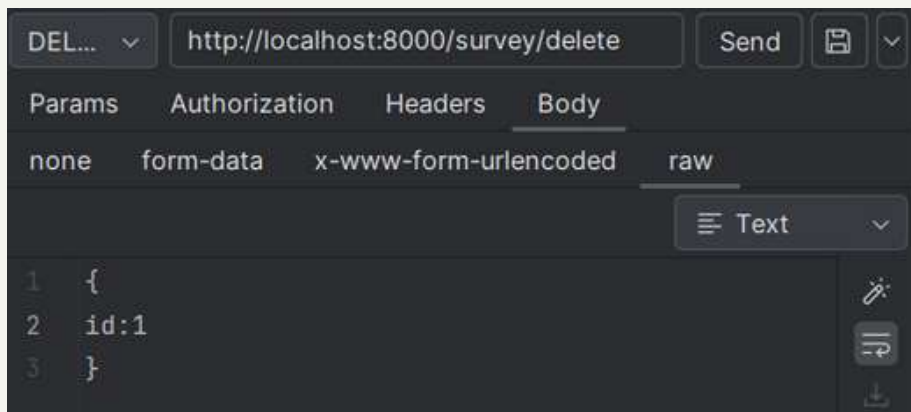


Fig. 34

Verificamos que se haya eliminado

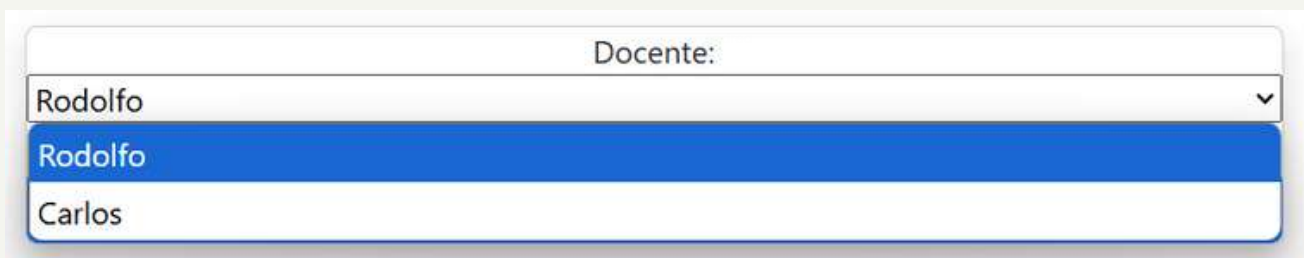


Fig. 35

Fig. 33