

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

    public static Connection getConnection() throws SQLException { 1 usage
        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
```

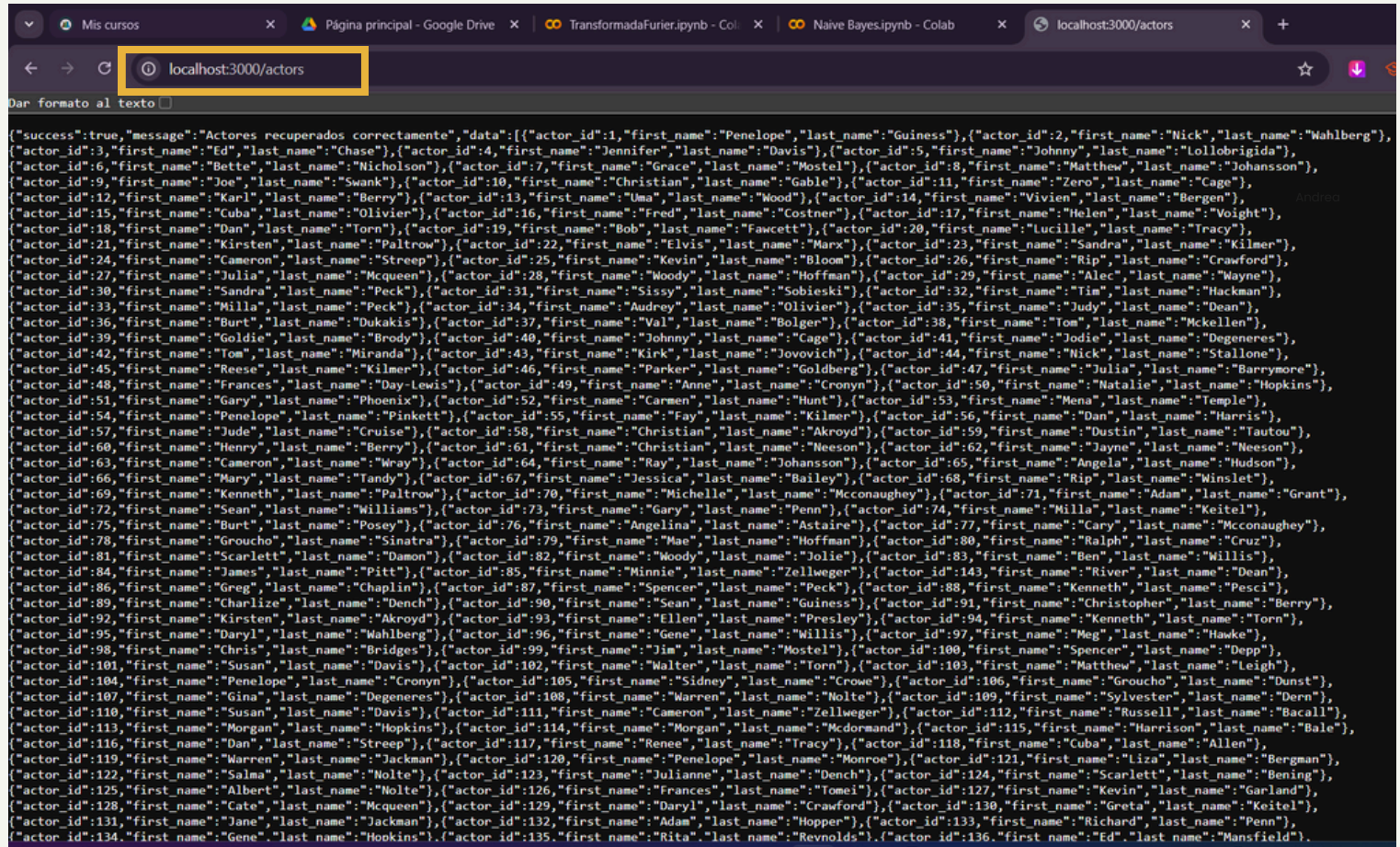
Fig. 8

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



The screenshot shows a web browser with the address bar displaying 'localhost:3000/actors'. The page content is a large JSON array of actor objects. Each object contains an 'actor_id', 'first_name', and 'last_name'. The actors listed include Penelope Guinness, Nick Wahlberg, Ed Chase, Jennifer Davis, Johnny Lollobrigida, Bette Nicholson, Grace Mostel, Matthew Johansson, Joe Swank, Christian Gable, Zero Cage, Karl Berry, Uma Wood, Vivien Bergen, Cuba Olivier, Fred Costner, Helen Voight, Dan Torn, Bob Fawcett, Lucille Tracy, Kirsten Paltrow, Elvis Marx, Sandra Kilmer, Cameron Streep, Kevin Bloom, Rip Crawford, Julia McQueen, Woody Hoffman, Alec Wayne, Sandra Peck, Sissy Sobieski, Tim Hackman, Milla Audrey, Judy Dean, Tom Dukakis, Kirk Bolger, Tom McKellen, Goldie Brody, Johnny Cage, Jodie Degeneres, Dan Miranda, Kirk Jovovich, Nick Stallone, Reese Kilmer, Parker Goldberg, Julia Barrymore, Frances Day-Lewis, Anne Cronyn, Natalie Hopkins, Gary Phoenix, Carmen Hunt, Mena Temple, Penelope Pinkett, Fay Kilmer, Dan Harris, Jude Cruise, Christian Akroyd, Dustin Tautou, Henry Berry, Christian Neeson, Jayne Neeson, Cameron Wray, Ray Johansson, Angela Hudson, Mary Tandy, Jessica Bailey, Rip Winslet, Kenneth Paltrow, Michelle McConaughey, Adam Grant, Sean Williams, Gary Penn, Milla Keitel, Burt Posey, Angelina Astaire, Cary McConaughey, Groucho Sinatra, Mae Hoffman, Ralph Cruz, Scarlett Damon, Woody Jolie, Ben Willis, James Pitt, Minnie Zellweger, River Dean, Greg Chaplin, Spencer Peck, Kenneth Pesci, Charlize Dench, Sean Guinness, Christopher Berry, Daryl Wahlberg, Gene Willis, Meg Hawke, Chris Bridges, Jim Mostel, Spencer Depp, Susan Davis, Walter Torn, Matthew Leigh, Penelope Cronyn, Sidney Crowe, Groucho Dunst, Gina Degeneres, Warren Nolte, Sylvester Dern, Susan Davis, Cameron Zellweger, Russell Bacall, Dan Streep, Renee Tracy, Cuba Allen, Warren Jackman, Penelope Monroe, Liza Bergman, Salma Nolte, Julianne Dench, Scarlett Bening, Albert Nolte, Frances Tomei, Kevin Garland, Cate McQueen, Daryl Crawford, Greta Keitel, Jane Jackman, Adam Hopper, Richard Penn, Gene Hookins, Rita Reynolds, Ed Mansfield.

Fig. 10

Nota:

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

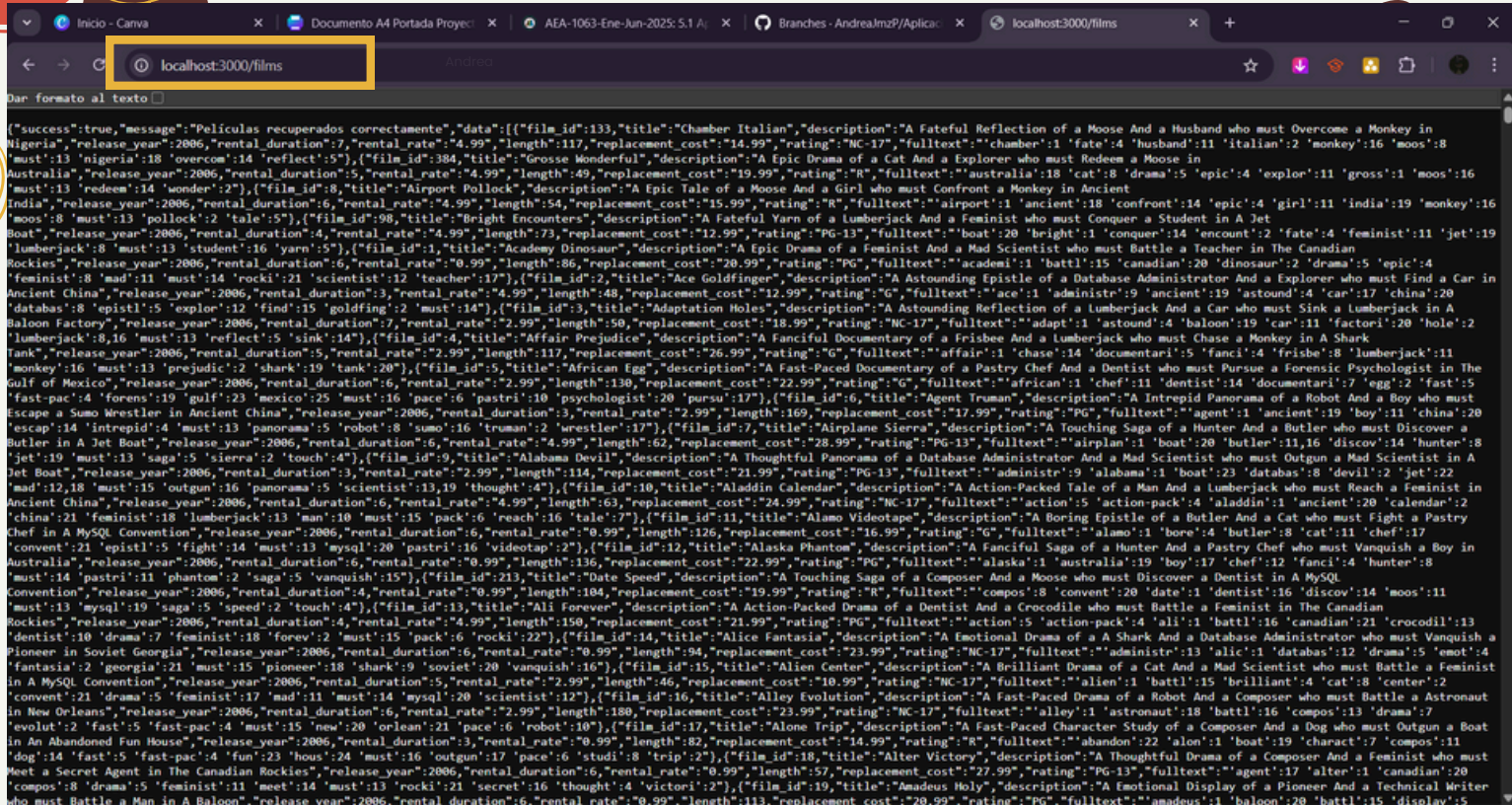


Fig. 11

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":"G","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_duration":3,"rental_rate":"4.99","length":null,"replacement_cost":"19.99","rating":"G","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":"G","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":"G","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



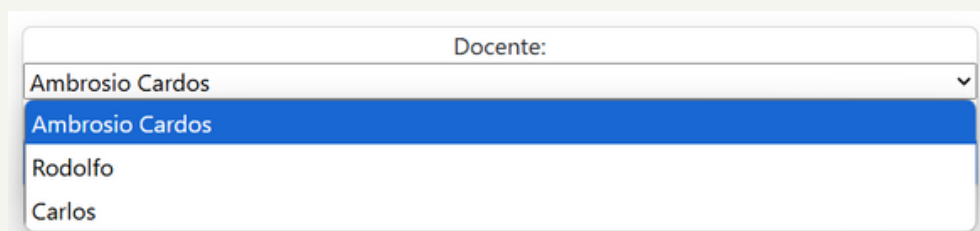
The screenshot shows a web form titled "Encuesta de Evaluación Docente". It contains the following fields:

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

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

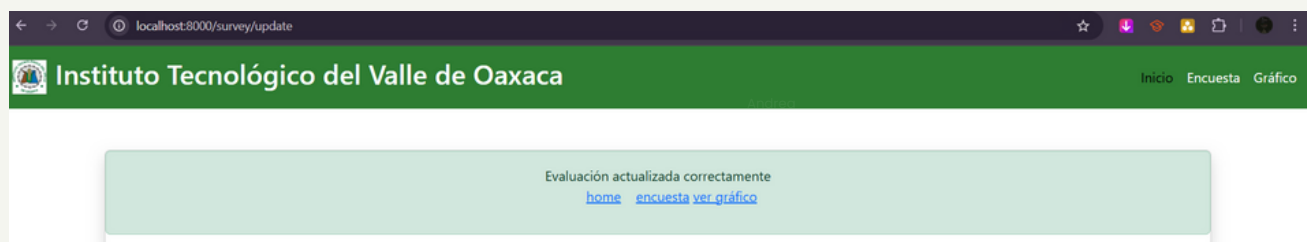


Fig. 32

Comprobamos que se actualizó (ahora está bien escrito)

Docente:

Ambrosio Cardoso ▼

Ambrosio Cardoso

Rodolfo

Carlos

Fig. 33

Eliminamos uno de la encuesta

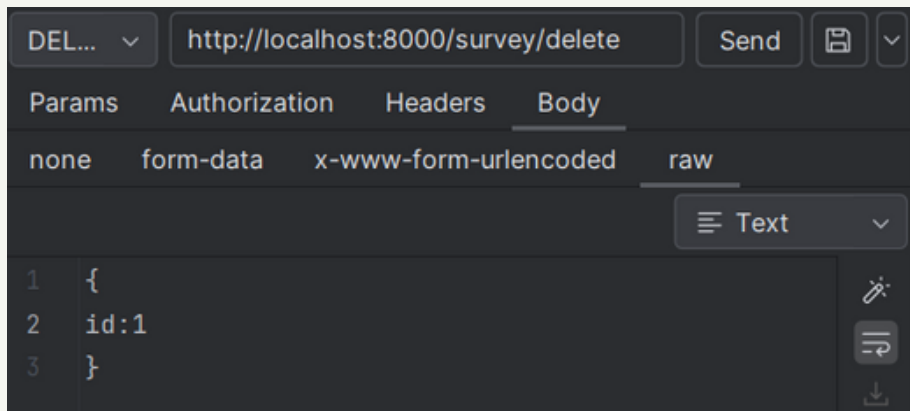


Fig. 34

Verificamos que se haya eliminado

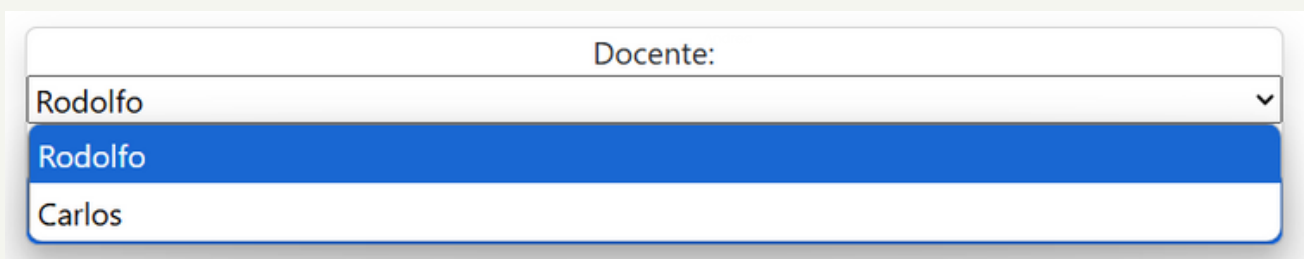


Fig. 35

Fig. 33