



Instituto Tecnológico
Nacional de México



Instituto Tecnológico
del valle de Oaxaca



Reporte Crud

Mayo 2025

Ingeniería en Tecnologías de la Información y las
Comunicaciones

Realizado por: Oscar Javier Agudo Juarez

Oracle y java



La dependencia de junit-jupiter es fundamental para que puedas escribir y ejecutar tus pruebas unitarias para las operaciones CRUD de CustomerDAOImpl, asegurando que la lógica de acceso a datos funcione correctamente con la base de datos Oracle.

```
<dependency>
  <groupId>com.oracle.database.jdbc</groupId>
  <artifactId>ojdbc11</artifactId>
  <version>21.9.0.0</version> <!-- Compatible con Oracle 21c -->
</dependency>
<dependency>
  <groupId>org.junit.jupiter</groupId>
  <artifactId>junit-jupiter</artifactId>
  <version>5.10.0</version>
  <scope>test</scope>
</dependency>
```

Para la conexión con oracle se proporciona un método estático getConnection() para establecer una conexión con la base de datos Oracle.

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

public class DatabaseConnection {
    private static final String URL = "jdbc:oracle:thin:@//localhost:1521/xes";
    private static final String USER = "co";
    private static final String PASSWORD = "codigo26";

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

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Oracle y java



CustomerDAOImpl.java

- Es la implementación concreta de la interfaz CustomerDAO.
- Utiliza JDBC para interactuar con la base de datos Oracle.
- Cada método implementa la lógica SQL correspondiente para las operaciones CRUD:
 - add: Ejecuta una sentencia INSERT para añadir un nuevo cliente. [cite: CustomerDAOImpl.java]
 - getById: Ejecuta una sentencia SELECT para recuperar un cliente por su customer_id.
 - getAll: Ejecuta una sentencia SELECT para recuperar todos los clientes.
 - update: Ejecuta una sentencia UPDATE para modificar los datos de un cliente.
 - delete: Ejecuta una sentencia DELETE para eliminar un cliente.
- Utiliza PreparedStatement para prevenir inyección SQL y mejorar el rendimiento.
- Maneja ResultSet para procesar los resultados de las consultas.

Oracle y java




CustomerDAO.java

- Es una interfaz que define el contrato para las operaciones de acceso a datos (DAO) para la entidad Customer.
- Declara los siguientes métodos:
 - `add(Customer customer)`: Para añadir un nuevo cliente a la base de datos.
 - `getById(Long id)`: Para recuperar un cliente por su ID.
 - `getAll()`: Para recuperar una lista de todos los clientes.
 - `update(Customer customer)`: Para actualizar un cliente existente.
 - `delete(Long id)`: Para eliminar un cliente por su ID.
- Todos los métodos declaran que pueden lanzar una `SQLException`.



CustomerDAOTest.java y SimpleInsertTest.java

- Ambas clases son pruebas JUnit 5 para verificar la funcionalidad de CustomerDAOImpl. [cite: CustomerDAOTest.java, SimpleInsertTest.java]
 - Utilizan @BeforeAll para configurar la conexión a la base de datos y la instancia de CustomerDAO antes de que se ejecuten todas las pruebas. [cite: CustomerDAOTest.java, SimpleInsertTest.java]
 - Utilizan @AfterAll para cerrar la conexión a la base de datos después de que todas las pruebas hayan finalizado. [cite: CustomerDAOTest.java, SimpleInsertTest.java]
 - CustomerDAOTest.java:
 - Contiene un método testCRUDOperations() que prueba el ciclo completo de Crear, Leer, Actualizar y Borrar para un cliente. [cite: CustomerDAOTest.java]
 - Verifica la inserción, recuperación, actualización y eliminación exitosa de un cliente utilizando aserciones JUnit como assertNotNull, assertEquals y assertNull. [cite: CustomerDAOTest.java]
 - También verifica la recuperación de todos los clientes y la presencia del cliente de prueba en la lista. [cite: CustomerDAOTest.java]
 - SimpleInsertTest.java:
 - Contiene un método testSimpleInsert() enfocado específicamente en la operación de inserción y lectura básica. [cite: SimpleInsertTest.java]
 - Genera un customerId único utilizando System.currentTimeMillis() para evitar conflictos en pruebas consecutivas. [cite: SimpleInsertTest.java]
 - Inserta un cliente, lo recupera y verifica su existencia y detalles. [cite: SimpleInsertTest.java]
 - ★ ■ Imprime mensajes en la consola para mostrar el progreso y los resultados de la prueba. [cite: SimpleInsertTest.java]
 - Muestra el número total de clientes en la base de datos. [cite: SimpleInsertTest.java]
- 

Resultado de que si se modificó(utilizando la clase SimpleInsertTest)



```
"C:\Program Files\Java\jdk-24\bin\java.exe" ...  
Insertando customer con ID: 1748658256704  
✓ Customer insertado  
✓ Customer verificado: Test User  
✉ Email: test@example.com  
👤 Total customers en base: 393  
🗑 Customer de prueba eliminado  
  
Process finished with exit code 0
```

02



Postgres-TypeScript

Para la conexión se necesito la instalación de bun para ello se ejecuto los siguientes comandos

```
powershell -c "irm bun.sh/install.ps1|iex"  
npm install -g bun # the last `npm` command you'll ever need
```

Ya una vez instalado, ejecutamos los siguientes comando dentro de la carpeta de mi programa

```
bun .init  
bun add drizzle-orm pg bun dotenv - Para usar Drizzle ORM con PostgreSQL y Bun,  
junto con dotenv para las variables de entorno
```

```
C:\Users\usuario\Documents\Developer\JS\JavaScript_postgres>bun add drizzle-orm pg bun dotenv  
bun add v1.2.12 (32a47ae4)  
error: moving "@oven/bun-windows-x64-baseline" to cache dir failed  
ERROR: Operation not permitted (WSetInformationFile())  
From: 3e3d9bffff2de5ae-00000012.bun-windows-x64-baseline  
To: @oven/bun-windows-x64-baseline@1.2.12@001  
  
error: InstallFailed extracting tarball from @oven/bun-windows-x64-baseline  
C:\Users\usuario\Documents\Developer\JS\JavaScript_postgres>bun add drizzle-orm pg  
bun add v1.2.12 (32a47ae4)  
  
installed drizzle-orm@0.43.1  
installed pg@8.15.6  
  
[2.43s] done  
  
C:\Users\usuario\Documents\Developer\JS\JavaScript_postgres>bun add dotenv  
bun add v1.2.12 (32a47ae4)  
  
installed dotenv@16.5.0  
  
[119.00ms] done  
  
C:\Users\usuario\Documents\Developer\JS\JavaScript_postgres>bun add bun  
bun add v1.2.12 (32a47ae4)
```

En mi caso
se ejecuta
por parte por
q me genero
un error

02



Postgres-TypeScript

El proyecto sigue una estructura de capas (Controladores, Servicios, Repositorios, Esquemas) que promueve la separación de responsabilidades y facilita la mantenibilidad.

- **Controllers:** Manejan las solicitudes HTTP y coordinan con los servicios.
- **Services:** Contienen la lógica de negocio y orquestan las operaciones de los repositorios.
- **Repositories:** Interactúan directamente con la base de datos (Drizzle ORM).
- **Schemas:** Definen la estructura de los datos y se utilizan para la validación.

Funcionalidades Implementadas

- Gestión de Actores:
- Obtener todos los actores.
- Obtener un actor por ID.
- Agregar un nuevo actor (con validación de esquema).
- Eliminar un actor por ID.
- Actualizar un actor por ID.

Gestión de Películas:

- Obtener todas las películas.
- Obtener una película por ID.
- Agregar una nueva película.
- Eliminar una película por ID.
- Actualizar una película por ID.

02



Postgres-TypeScript

Este para mostrar todos los actores

```
const actorRouter = new Hono();

actorRouter.get( path: '/actores', async (): Promise<Response> => {
  const { status, body } = await ActorController.getAll();
  return new Response(JSON.stringify(body), {
    status: status,
    headers: { 'Content-Type': 'application/json' }
  });
});
```

Este para agregar y mostrar los que se agregaron(solo el primero), como parámetros pide el primer nombre y el apellido, que en este caso se agregan alguno con valores específicos

```
actorRouter.get( path: '/actores/agregar', async (): Promise<Response> => {
  const { status, body } = await ActorController.add({"first_name": "Constantine", "last_name": "Castro"});
  await ActorController.add({"first_name": "Leon", "last_name": "Castro"});
  await ActorController.add({"first_name": "Leon S.", "last_name": "Kennedy"});

  return new Response(JSON.stringify(body), {
    status: status,
    headers: { 'Content-Type': 'application/json' }
  });
});
```

02



Postgres-TypeScript

Elimina por id cualquier autor en la base de datos y lo muestra

```
actorRouter.get( path: '/actors/eliminar/:id', async (c : Context<BlankEnv, '/actors/eliminar/:id',...
  const id :number = Number(c.req.param( key: 'id'));
  const { status, body } = await ActorController.delete(id);
  return new Response(JSON.stringify(body), {
    status: status,
    headers: { 'Content-Type': 'application/json' }
  });
});
```

Va a actualizar el autor mediante el id y los parámetros serán separados por / simulando la petición

```
actorRouter.get( path: '/actors/actualizar/:id/:first_name/:last_name', async (c : Context<BlankE
  const id :number = Number(c.req.param( key: 'id'));
  const first_name :string = c.req.param( key: 'first_name');
  const last_name = c.req.param('last_name');
  const { status, body } = await ActorController.update(id, { first_name, last_name });
  return new Response(JSON.stringify(body), {
    status: status,
    headers: { 'Content-Type': 'application/json' }
  });
});
```

02



Evidencias

Se muestran todos los actores

```
localhost:3000/actors
Impresión con formato estilístico
{"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":"Helen","last_name":"Voight"},{"actor_id":22,"first_name":"Helen","last_name":"Voight"},
{"actor_id":23,"first_name":"Helen","last_name":"Voight"},{"actor_id":24,"first_name":"Helen","last_name":"Voight"},
{"actor_id":25,"first_name":"Helen","last_name":"Voight"},{"actor_id":26,"first_name":"Helen","last_name":"Voight"},
{"actor_id":27,"first_name":"Helen","last_name":"Voight"},{"actor_id":28,"first_name":"Helen","last_name":"Voight"},
{"actor_id":29,"first_name":"Helen","last_name":"Voight"},{"actor_id":30,"first_name":"Helen","last_name":"Voight"}]}
```

Muestra que se agrego el primero y si vamos al /actors se muestra que si se agregaron

```
localhost:3000/actors/agregar
Impresión con formato estilístico
{"success":true,"message":"Actor creado","data":[{"actor_id":209,"first_name":"Constantine","last_name":"Constantine"}]}

{"actor_id":209,"first_name":"Constantine","last_name":"Constantine"},
{"actor_id":210,"first_name":"Leon","last_name":"Castro"},{"actor_id":211,"first_name":"Leon S.,"last_name":"Kennedy"}]}
```

02



Evidencias

Actualizar

```
localhost:3000/actors/actualizar/210/Actualizado/Leon-Castro

Impresión con formato estilístico ☐

{"success":true,"message":"Actor actualizado correctamente","data":
[{"actor_id":210,"first_name":"Actualizado","last_name":"Leon-Castro"}]}
```

Si se actualizo

```
{"actor_id":210,"first_name":"Actualizado","last_name":"Leon-Castro"}]}
```

Eliminar

```
localhost:3000/actors/eliminar/210

Impresión con formato estilístico ☐

{"success":true,"message":"Actor eliminado correctamente","data":
[{"actor_id":210,"first_name":"Actualizado","last_name":"Leon-Castro"}]}
```

```
{
  "actor_id": 192, "first_name": "John", "last_name": "Suvari",
  "actor_id": 193, "first_name": "Burt", "last_name": "Temple",
  "actor_id": 194, "first_name": "Meryl", "last_name": "Allen",
  "actor_id": 195, "first_name": "Jayne", "last_name": "Silverstone",
  "actor_id": 196, "first_name": "Bela", "last_name": "Walken",
  "actor_id": 197, "first_name": "Reese", "last_name": "West",
  "actor_id": 198, "first_name": "Mary", "last_name": "Keitel",
  "actor_id": 199, "first_name": "Julia", "last_name": "Fawcett",
  "actor_id": 200, "first_name": "Thora", "last_name": "Temple",
  "actor_id": 201, "first_name": "Leonardo", "last_name": "DiCaprio",
  "actor_id": 206, "first_name": "Nuevo", "last_name": "Actor",
  "actor_id": 207, "first_name": "Nuevo1", "last_name": "Actor1",
  "actor_id": 208, "first_name": "Actualizado", "last_name": "actor1",
  "actor_id": 209, "first_name": "Constantine", "last_name": "Constantine",
  "actor_id": 211, "first_name": "Leon S.", "last_name": "Kennedy"}]}
```

02



Postgres-TypeScript

Es el mismo código para films lo único diferente es en actualizar se piden mas datos y el id del lenguaje es recibido y convertido a number

```
filmRouter.get( path: '/films/actualizar/:id/:title/:description/:release_year/:language_id', async (c: Context<BlankEnv, /films/actualizar/:id/:...>) : Promise<Response> => {
  const id : number = Number(c.req.param( key: 'id'));
  const title : string = c.req.param( key: 'title');
  const description : string = c.req.param( key: 'description');
  const release_year : string = c.req.param( key: 'release_year');
  const language_id : number = Number(c.req.param( key: 'language_id'));
  const { status, body } = await FilmController.update(id, { title, description, release_year, language_id});
  return new Response(JSON.stringify(body), {
    status: status,
    headers: { 'Content-Type': 'application/json' }
  });
});
export default filmRouter; Show usages
```

Datos que se agregaran

```
filmRouter.get( path: '/films/agregar', async () : Promise<Response> => {
  const { status, body } = await FilmController.add({title: 'Inception', 'description': 'A mind-bending thriller', 'release_year': '2010', 'language_id': 1});
  await FilmController.add({title: 'The Matrix', 'description': 'A sci-fi classic', 'release_year': '1999', 'language_id': 1});
  await FilmController.add({title: 'Interstellar', 'description': 'A space exploration epic', 'release_year': '2014', 'language_id': 1});

  return new Response(JSON.stringify(body), {
    status: status,
    headers: { 'Content-Type': 'application/json' }
  });
});
```


02



Evidencias

Se muestran todas las películas

```
localhost:3000/films
Impresión con formato estilístico
{"success":true,"message":"Películas recuperados correctamente","data":[{"film_id":133,"title":"Chamber Italian","description":"A Fateful Reflection of a Moose And a Husband who must Overcome a Monkey in Nigeria","release_year":2006,"language_id":1,"rental_duration":7,"rental_rate":"4.99","length":117,"replacement_cost":"14.99","rating":"NC-17","last_update":"2013-05-26 14:50:58.951","special_features":["Trailers"],"fulltext":"'chamber':1 'fate':4 'husband':11 'italian':2 'monkey':16 'moos':8 'must':13 'nigeria':18 'overcom':14 'reflect':5}],{"film_id":384,"title":"Grosse Wonderful","description":"A Epic Drama of a Cat And a Explorer who must Redeem a Moose in Australia","release_year":2006,"language_id":1,"rental_duration":5,"rental_rate":"4.99","length":49,"replacement_cost":"19.99","rating":"R","last_update":"2013-05-26 14:50:58.951","special_features":["Behind the Scenes"],"fulltext":"'australia':18 'cat':8 'drama':5 'epic':4 'explor':11 'gross':1 'moos':16 'must':13 'redeem':14 'wonder':2}],{"film_id":8,"title":"Airport Pollock","description":"A Epic Tale of a Moose And a Girl who must Confront a Monkey in Ancient India","release_year":2006,"language_id":1,"rental_duration":6,"rental_rate":"4.99","length":54,"replacement_cost":"15.99","rating":"R","last_update":"2013-05-26 14:50:58.951","special_features":["Trailers"],"fulltext":"'airport':1 'ancient':18 'confront':14 'epic':4 'girl':11 'india':19 'monkey':16 'moos':8 'must':13 'pollock':2 'tale':5}],{"film_id":98,"title":"Bright Encounters","description":"A Fateful Yarn of a Lumberjack And a Feminist who must Conquer a"}]
```

Muestra que se agrego el primero y si vamos al /actors se muestra que si se agregaron

```
localhost:3000/films/agregar
Impresión con formato estilístico
{"success":true,"message":"pelicula creado","data":[{"film_id":1008,"title":"Inception","description":"A mind-bending thriller","release_year":2010,"language_id":1,"rental_duration":3,"rental_rate":"4.99","length":null,"replacement_cost":"19.99","rating":"G","last_update":"2025-05-30 22:59:19.001637","special_features":null,"fulltext":"'bend':5 'incept':1 'mind':4 'mind-bend':3 'thriller':6"}]}
```

```
localhost:3000/films/1008
Impresión con formato estilístico
{"success":true,"message":"Pelicula encontrado","data":[{"film_id":1008,"title":"Inception","description":"A mind-bending thriller","release_year":2010,"language_id":1,"rental_duration":3,"rental_rate":"4.99","length":null,"replacement_cost":"19.99","rating":"G","last_update":"2025-05-30 22:59:19.013741","special_features":null,"fulltext":"'bend':5 'incept':1 'mind':4 'mind-bend':3 'thriller':6"}]}
```

```
localhost:3000/films/1009
Impresión con formato estilístico
{"success":true,"message":"Pelicula encontrado","data":[{"film_id":1009,"title":"The Matrix","description":"A sci-fi classic","release_year":1999,"language_id":1,"rental_duration":3,"rental_rate":"4.99","length":null,"replacement_cost":"19.99","rating":"G","last_update":"2025-05-30 22:59:19.013741","special_features":null,"fulltext":"'classic':7 'fi':6 'matrix':2 'sci':5 'sci-fi':4"}]}
```

```
localhost:3000/films/1010
Impresión con formato estilístico
{"success":true,"message":"Pelicula encontrado","data":[{"film_id":1010,"title":"Interstellar","description":"A space exploration epic","release_year":2014,"language_id":1,"rental_duration":3,"rental_rate":"4.99","length":null,"replacement_cost":"19.99","rating":"G","last_update":"2025-05-30 22:59:19.01787","special_features":null,"fulltext":"'epic':5 'explor':4 'interstellar':1 'space':3"}]}
```

02



Evidencias

Actualizar

```
localhost:3000/films/actualizar/1010/otraPeli/peliculafunny/2020/1
Impresión con formato estilístico
{"success":true,"message":"Película actualizado correctamente","data":
[{"film_id":1010,"title":"otraPeli","description":"peliculafunny","release_year":2020,"language_id":1,"rental_duration":3,"ren
tal_rate":4.99,"length":null,"replacement_cost":19.99,"rating":"G","last_update":"2025-05-30
23:12:39.40309","special_features":null,"fulltext":"'otrap':1 'peliculafunni':2"}]}
```

Si se actualizo

```
localhost:3000/films/1010
Impresión con formato estilístico
{"success":true,"message":"Película encontrado","data":
[{"film_id":1010,"title":"otraPeli","description":"peliculafunny","release_year":2020,"language_id":1,"rental_duration":3,"ren
tal_rate":4.99,"length":null,"replacement_cost":19.99,"rating":"G","last_update":"2025-05-30
23:12:39.40309","special_features":null,"fulltext":"'otrap':1 'peliculafunni':2"}]}
```

Eliminar

```
localhost:3000/films/eliminar/1010
Impresión con formato estilístico
{"success":true,"message":"Película eliminado correctamente","data":
[{"film_id":1010,"title":"otraPeli","description":"peliculafunny","release_year":2020,"language_id":1,"rental_duration":3,"ren
tal_rate":4.99,"length":null,"replacement_cost":19.99,"rating":"G","last_update":"2025-05-30
23:12:39.40309","special_features":null,"fulltext":"'otrap':1 'peliculafunni':2"}]}

localhost:3000/films/1010
Impresión con formato estilístico
{"success":false,"message":"Película no encontrado","data":[]}
```

Mejoras que debí aplicar

debería usar DELETE para eliminar y PUT o PATCH para actualizar.

y en las películas solo mostrar los datos que estoy ocupando no todos

02



MariaDb - php

Para la instalación de php hice lo siguiente:

Descargar PHP:

- Ve al sitio oficial de PHP para Windows:
<https://windows.php.net/download/>
- Extraer los archivos en una ruta específica
- Configurar php.ini para que se habilite las extensiones:
 - 1.extension=curl
 - 2.extension=gd (para manipulación de imágenes)
 - 3.extension=mbstring (para manejo de cadenas multibyte)
 - 4.extension=pdo_mysql (para conectar con MySQL/MariaDB a través de PDO)
 - 5.extension=openssl
 - 6.extension=mysqli
- Por último se configura en las variables de entorno para agregarlo en el path

una vez hecho eso cree la base de datos evaluacion
y la tabla respuestas

Comando para ejecutar el php y haga la conexión

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


02



MariaDb - php

Los cambios que hice a lo que nos proporcionó el profesor es agregar dos botones en el header, cree dos archivos en views y uno para el de eliminar otro actualizar, los dos lo maje con form ya que se me hizo más práctico por el submit,

Para el código hice el metodo actualizar para que obtenga los datos del formulario y después llamar el método que cree para actualizar la base de datos

```
public static function actualizar() { no usages
    $docente = $_POST['teacher'] ?? '';
    $p1 = (int) ($_POST['pregunta1'] ?? 0);
    $p2 = (int) ($_POST['pregunta2'] ?? 0);
    $p3 = (int) ($_POST['pregunta3'] ?? 0);
    $comentarios = $_POST['comentarios'] ?? '';
    $conn = Database::getConnection();
    $surveyModel = new SurveyModel($conn);
    [$success, $message] = $surveyModel->update($docente, $p1, $p2, $p3, $comentarios);

    $response = (object)[
        'success' => $success,
        'message' => $message,
        'data' => []
    ];

    TemplateEngine::render( view: "response", ['response' => $response]);
}
```

```
public function update($docente, $p1, $p2, $p3, $comentarios) { 1 usage
    debug_to_console( data: "Actualizando evaluación para docente: " . $docente);
    $stmt = $this->conn->prepare(
        "UPDATE respuestas
        SET pregunta1 = ?, pregunta2 = ?, pregunta3 = ?, comentarios = ?
        WHERE docente = ?"
    );

    $stmt->bind_param("iiiss", $p1, $p2, $p3, $comentarios, $docente);

    if ($stmt->execute()) {
        return [true, "Evaluación actualizada correctamente."];
    } else {
        return [false, "Error al actualizar la evaluación: " . $stmt->error];
    }
}
```

02



MariaDb - php

para eliminar se obtiene el nombre del profesor ya que seleccionaría el nombre en vez del id (aun que es mas recomendado por el id, dependiendo de la cantidad de registros)

```
public function delete($docente) { 1 usage
    debug_to_console( data: "Eliminando evaluación para docente: " . $docente);

    $stmt = $this->conn->prepare(
        "DELETE FROM respuestas WHERE docente = ?"
    );

    $stmt->bind_param("s", $docente);

    if ($stmt->execute()) {
        return [true, "Evaluación eliminada correctamente."];
    } else {
        return [false, "Error al eliminar la evaluación: " . $stmt->error];
    }
}
```

```
public static function eliminar() { no usages

    $docente = $_POST['teacher'] ?? '';
    $conn = Database::getConnection();
    $surveyModel = new SurveyModel($conn);
    [$success, $message] = $surveyModel->delete($docente);

    $response = (object)[
        'success' => $success,
        'message' => $message,
        'data' => []
    ];

    TemplateEngine::render( view: "response", ['response' => $response]);
}
```

02



Evidencias

Primero agregar

Instituto Tecnológico del Valle de Oaxaca Inicio Encuesta Gráfico Actualizar Eliminar

Encuesta de Evaluación Docente

Nombre del docente:
martin

1. ¿Explica claramente los temas?
Nunca

2. ¿Fomenta la participación?
Nunca

3. ¿Es puntual y responsable?
Nunca

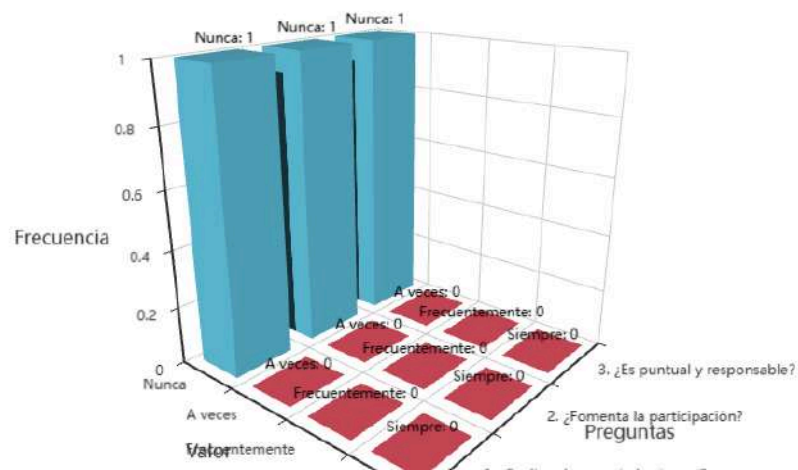
Comentarios adicionales:
profe

Enviar evaluación

Grafica sale asi

Docente:
martin

Ver gráfica



02



Evidencias

Ahora actualizar se notara que es una caja de selección y no una caja de texto

02



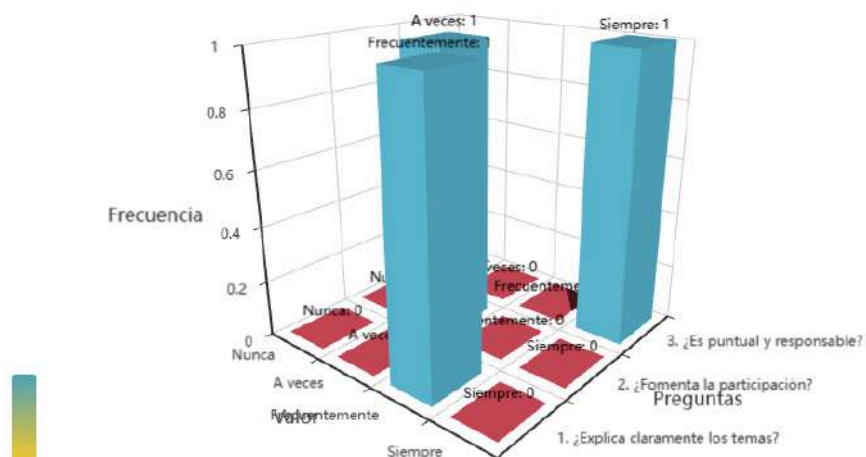
Evidencias

Grafica sale diferente por lo cual se nota que si se actualizo

Docente:

martin

Ver gráfica



02



Evidencias

Para eliminar solo pide le nombre del docente igual con una de seleccion

Encuesta de Evaluación Docente

Docente:

martin

▼

eliminar evaluación

La grafica no muestra nada por lo que se elimino correctamente

Docente:

▼

Ver gráfica

