



Escuela de habilidades digitales “**Coderhouse**”

## **Proyecto Integracion Educa Chile**

"Este proyecto ha sido elaborado con el propósito de obtener el Certificado MYSQL. A lo largo de su desarrollo, hemos focalizado nuestros esfuerzos en optimizar la gestión de bases de datos, aplicando prácticas y conceptos clave para asegurar un manejo eficiente y seguro de la información."

**Profesor:** Camilo Redondo  
**Ayudantía:** José Ignacio López  
**Alumno:** Ever Martin Salinas  
**Comision#:** 47365

## ÍNDICE

1. Introducción	3
2. Descripción	4
3. Objetivo	5
4. Situacion Problematica	6
5. Diagrama Entidad-Relación	7
6. Descripción de tablas	8
7. Creación de tablas	11
8. Inserción de datos	13
9. Vista Principales	16
10. Funciones	17
11. Stored Procedures	18
12. Triggers	20
13. Usuarios Permisos	22
14. Transacciones	22
15. Reportes Power Bi	24
16. Herramientas Utilizadas	27
17. Comentarios Finales	27

## **1. Introducción**

El proyecto Integración Educa Chile DB SQL tiene como objetivo optimizar la gestión de información y comunicación en el ámbito educativo a través de una base de datos integral. Diseñada para profesionales, estudiantes y usuarios, la herramienta facilita el acceso a datos sobre profesionales, estudiantes, cursos, calificaciones y contenido educativo. Busca mejorar la toma de decisiones, la administración y la calidad de los servicios escolares, convirtiéndose en un recurso valioso para potenciar la calidad de la educación y fortalecer la interacción en la comunidad escolar.

## 2. Descripción

El proyecto se centra en proporcionar servicios en el entorno escolar a través de una herramienta integral. Su objetivo principal es desarrollar una base de datos que mejore la gestión de información y la comunicación en la escuela, simplificando el acceso a datos sobre profesionales, estudiantes, cursos y calificaciones. La iniciativa busca optimizar la toma de decisiones, la administración y la calidad de los servicios educativos, convirtiéndose en un recurso valioso para potenciar la calidad y la interacción en la comunidad escolar.

### 3. Objetivo

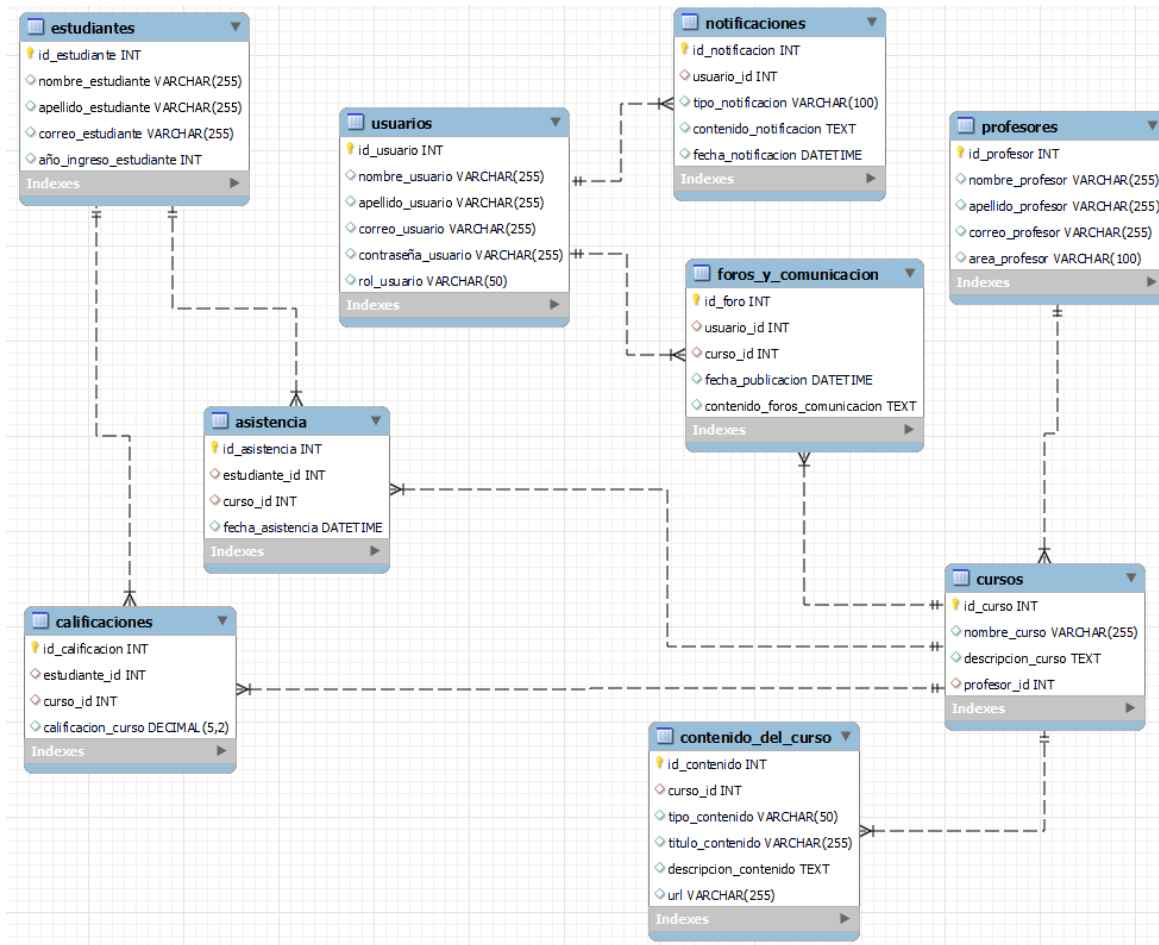
La base de datos tiene como objetivos principales optimizar la gestión de información y comunicación en el entorno escolar, enriquecer la experiencia educativa mediante el acceso eficiente a recursos y apoyo personalizado, facilitar la interacción entre profesionales y usuarios, y contribuir a la mejora continua de la calidad de los servicios escolares. A través de la recopilación estructurada de datos sobre profesionales, estudiantes, cursos y calificaciones, se busca impulsar la eficiencia, la toma de decisiones y la colaboración, creando un entorno educativo más efectivo y centrado en las necesidades de la comunidad escolar.

#### 4. Situación Problemática

La situación problemática abordada por la base de datos se centra en varios desafíos clave en el entorno educativo:

1. **Acceso a la información:** Simplificar el acceso a información dispersa sobre cursos, profesores, estudiantes y calificaciones para facilitar la toma de decisiones y la administración eficiente.
2. **Gestión de datos dispersos:** Enfrentar la dificultad común de gestionar datos dispersos almacenados en diversos sistemas, centralizando y estructurando la información para mejorar la administración y reducir errores.
3. **Comunicación efectiva:** Mejorar la comunicación entre profesionales, estudiantes y demás actores en la comunidad escolar mediante herramientas que faciliten el intercambio eficiente y oportuno de información.
4. **Optimización de la administración:** Automatizar procesos y proporcionar información actualizada en tiempo real para optimizar la administración de cursos, horarios, calificaciones y recursos.
5. **Enriquecimiento de la experiencia educativa:** Contribuir a una experiencia de aprendizaje más rica y personalizada para los estudiantes al almacenar información sobre contenido educativo y calificaciones.
6. **Mejora de la interacción entre partes interesadas:** Fomentar una interacción más efectiva entre profesionales, estudiantes y otros involucrados mediante un sistema de información centralizado, contribuyendo así a un entorno educativo más centrado en las necesidades de la comunidad escolar.

## 5. Diagrama Entidad-Relación



## 6. Descripción de tablas

**Tabla de Usuarios:**

Tipo de Clave	Columna	Tipo de Datos	Descripción
PK	id_usuario	INT AUTO_INCREMENT	Identificador único para cada usuario.
	nombre_usuario	VARCHAR(255)	Nombre del usuario.
	apellido_usuario	VARCHAR(255)	Apellido del usuario.
	correo_usuario	VARCHAR(255)	Correo electrónico del usuario.
	Contraseña_usuario	VARCHAR(255)	Contraseña del usuario (hash por seguridad).
	rol_usuario	VARCHAR(50)	Rol o función del usuario en el sistema.

**Tabla Profesores:**

Tipo de Clave	Columna	Tipo de Datos	Descripción
PK	id_profesor	INT AUTO_INCREMENT	Identificador único para cada profesor.
	nombre_profesor	VARCHAR(255)	Nombre del profesor.
	apellido_profesor	VARCHAR(255)	Apellido del profesor.
	correo_profesor	VARCHAR(255)	Correo electrónico del profesor.
	area_profesor	VARCHAR(100)	Área o especialización del profesor.

**Tabla Cursos:**

Tipo de Clave	Columna	Tipo de Datos	Descripción
PK	id_curso	INT AUTO_INCREMENT	Identificador único para cada curso.
	nombre_curso	VARCHAR(255)	Nombre del curso.
	descripción_curso	TEXT	Descripción detallada del curso.
FK	profesor_id	INT	El ID del profesor que imparte el curso.

**Tabla Estudiantes:**

Tipo de Clave	Columna	Tipo de Datos	Descripción
PK	id_estudiante	INT AUTO_INCREMENT	Identificador único para cada estudiante.
	nombre_estudiante	VARCHAR(255)	Nombre del estudiante.
	apellido_estudiante	VARCHAR(255)	Apellido del estudiante.
	correo_estudiante	VARCHAR(255)	Correo electrónico del estudiante.
	año_ingreso_estudiante	INT	Año de ingreso del estudiante.

**Tabla Calificaciones:**



Tipo de Clave	Columna	Tipo de Datos	Descripcion
PK	id_calificación	INT AUTO_INCREMENT	Identificador único para cada calificación.
FK	estudiante_id	INT	El ID del estudiante al que pertenece la calificación.
FK	curso_id	INT	El ID del curso al que pertenece la calificación.
	calificación_curso	DECIMAL(5, 2)	La calificación o nota del estudiante en el curso.

**Tabla Contenido\_del\_Curso:**

Tipo de Clave	Columna	Tipo de Datos	Descripcion
PK	id_contenido	INT AUTO_INCREMENT	Identificador único para cada contenido de curso.
FK	curso_id	INT	El ID del curso al que pertenece el contenido.
	tipo_contenido	VARCHAR(50)	Tipo de contenido (por ejemplo, video, texto, presentación).
	título_contenido	VARCHAR(255)	Título o nombre del contenido.
	descripción_contenido	TEXT	Descripción detallada del contenido.
	url	VARCHAR(255)	URL o enlace relacionado con el contenido.

**Tabla Foro\_y Comunicación:**

Tipo de Clave	Columna	Tipo de Datos	Descripcion
PK	id_foro	INT AUTO_INCREMENT	Identificador único para cada publicación en el foro.
FK	usuario_id	INT	El ID del usuario que realiza la publicación.
FK	curso_id	INT	El ID del curso al que pertenece la publicación.
	fecha_publicación	DATETIME	Fecha y hora de la publicación.
	contenido_foros_comunicación	TEXT	Contenido de la publicación en el foro o comunicación.

**Tabla Asistencia**

Tipo de Clave	Columna	Tipo de Datos	Descripción
PK	id_asistencia	INT AUTO_INCREMENT	Se utiliza para rastrear la asistencia de los estudiantes
FK	estudiante_id	INT	Hace referencia al id del estudiante que asiste a la clase.
FK	curso_id	INT	Hace referencia al id del curso al que pertenece la asistencia.
	fecha_asistencia	DATETIME	Almacena la fecha y hora de la asistencia.

**Tabla Notificaciones**

Tipo de Clave	Columna	Tipo de Datos	Descripción
PK	id_notificacion	INT AUTO_INCREMENT	Se utiliza para almacenar notificaciones enviadas a los usuarios de la plataforma educativa.
FK	usuario_id	INT	Hace referencia al id del usuario al que se envía la notificación.
	tipo_notificacion	INT	Almacena el tipo de notificación, como "nueva tarea" o "recordatorio".
	contenido_notificacion	TEXT	Contiene el contenido de la notificación.
	fecha_notificacion	DATETIME	Almacena la fecha y hora de la notificación.

## 7. Creación de tablas

```
CREATE SCHEMA IF NOT EXISTS Integración_Educa_Chile;
USE Integración_Educa_Chile;
-- Crear la tabla Usuarios
CREATE TABLE IF NOT EXISTS Usuarios (
  id_usuario INT AUTO_INCREMENT PRIMARY KEY,
  nombre_usuario VARCHAR(255),
  apellido_usuario VARCHAR(255),
  correo_usuario VARCHAR(255),
  contraseña_usuario VARCHAR(255),
  rol_usuario VARCHAR(50)
);
-- Crear la tabla Profesores
CREATE TABLE IF NOT EXISTS Profesores (
  id_profesor INT AUTO_INCREMENT PRIMARY KEY,
  nombre_profesor VARCHAR(255),
  apellido_profesor VARCHAR(255),
  correo_profesor VARCHAR(255),
  area_profesor VARCHAR(100)
);
-- Crear la tabla Cursos
CREATE TABLE IF NOT EXISTS Cursos (
  id_curso INT AUTO_INCREMENT PRIMARY KEY,
  nombre_curso VARCHAR(255),
  descripcion_curso TEXT,
  profesor_id INT,
  FOREIGN KEY (profesor_id) REFERENCES Profesores(id_profesor)
);
-- Crear la tabla Estudiantes
CREATE TABLE IF NOT EXISTS Estudiantes (
  id_estudiante INT AUTO_INCREMENT PRIMARY KEY,
  nombre_estudiante VARCHAR(255),
  apellido_estudiante VARCHAR(255),
  correo_estudiante VARCHAR(255),
  año_ingreso_estudiante INT
);
-- Crear la tabla Calificaciones
CREATE TABLE IF NOT EXISTS Calificaciones (
  id_calificacion INT AUTO_INCREMENT PRIMARY KEY,
  estudiante_id INT,
  curso_id INT,
  calificacion_curso DECIMAL(5, 2),
  FOREIGN KEY (estudiante_id) REFERENCES Estudiantes(id_estudiante),
  FOREIGN KEY (curso_id) REFERENCES Cursos(id_curso)
);
```

```

-- Crear la tabla Contenido del Curso
CREATE TABLE IF NOT EXISTS Contenido_del_Curso (
    id_contenido INT AUTO_INCREMENT PRIMARY KEY,
    curso_id INT,
    tipo_contenido VARCHAR(50),
    titulo_contenido VARCHAR(255),
    descripcion_contenido TEXT,
    url VARCHAR(255),
    FOREIGN KEY (curso_id) REFERENCES Cursos(id_curso)
);

-- Crear la tabla Foros y Comunicación (publicación)
CREATE TABLE IF NOT EXISTS Foros_y_Comunicacion (
    id_foro INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT,
    curso_id INT,
    fecha_publicacion DATETIME,
    contenido_foros_comunicacion TEXT,
    FOREIGN KEY (usuario_id) REFERENCES Usuarios(id_usuario),
    FOREIGN KEY (curso_id) REFERENCES Cursos(id_curso)
);

-- Crear la tabla Asistencia
CREATE TABLE IF NOT EXISTS Asistencia (
    id_asistencia INT AUTO_INCREMENT PRIMARY KEY,
    estudiante_id INT,
    curso_id INT,
    fecha_asistencia DATETIME,
    FOREIGN KEY (estudiante_id) REFERENCES Estudiantes(id_estudiante),
    FOREIGN KEY (curso_id) REFERENCES Cursos(id_curso)
);

-- Crear la tabla Notificaciones
CREATE TABLE IF NOT EXISTS Notificaciones (
    id_notificacion INT AUTO_INCREMENT PRIMARY KEY,
    usuario_id INT,
    tipo_notificacion VARCHAR(100),
    contenido_notificacion TEXT,
    fecha_notificacion DATETIME,
    FOREIGN KEY (usuario_id) REFERENCES Usuarios(id_usuario)
);

```

## 8. Inserción de datos

-- Tabla Estudiantes

**INSERT INTO** Estudiantes

(nombre\_estudiante, apellido\_estudiante, correo\_estudiante, año\_ingreso\_estudiante)

**VALUES**

('Juan', 'Pérez', 'juan@example.com', 2022),  
('María', 'López', 'maria@example.com', 2021),  
('Carlos', 'Gómez', 'carlos@example.com', 2023),  
('Ana', 'Rodríguez', 'ana@example.com', 2021),  
('Luis', 'Hernández', 'luis@example.com', 2022),  
('Laura', 'García', 'laura@example.com', 2021),  
('Pedro', 'Martínez', 'pedro@example.com', 2023),  
('Sofía', 'Díaz', 'sofia@example.com', 2020),  
('David', 'Torres', 'david@example.com', 2021),  
('Isabel', 'Rivera', 'isabel@example.com', 2022);

-- Tabla Usuarios

**INSERT INTO** Usuarios

(nombre\_usuario, apellido\_usuario, correo\_usuario, Contraseña\_usuario, rol\_usuario) **Values**

Juan,Pérez,juan@email.com,clave123,Estudiante  
María,López,maria@email.com,contraseña456,Profesor  
Carlos,Gómez,carlos@email.com,clave789,Estudiante  
Ana,Rodríguez,ana@email.com,password123,Administrador  
Luis,Hernández,luis@email.com,secret123,Estudiante  
Laura,García,laura@email.com,pass456,Estudiante  
Pedro,Martínez,pedro@email.com,clave123,Profesor  
Sofía,Díaz,sofia@email.com,contraseña789,Estudiante  
David,Torres,david@email.com,secret456,Estudiante  
Isabel,Rivera,isabel@email.com,pass123,Profesor

-- Tabla Profesores

**INSERT INTO** Profesores

(nombre\_profesor, apellido\_profesor, correo\_profesor, area\_profesor) **VALUES**

('Juan', 'Ortega', 'juan@profesor.com', 'Matemáticas'),  
('María', 'González', 'maria@profesor.com', 'Historia'),  
('Carlos', 'Sánchez', 'carlos@profesor.com', 'Ciencias'),  
('Ana', 'Rodríguez', 'ana@profesor.com', 'Inglés'),  
('Luis', 'Hernández', 'luis@profesor.com', 'Economía'),  
('Laura', 'García', 'laura@profesor.com', 'Informática'),  
('Pedro', 'Martínez', 'pedro@profesor.com', 'Arte'),  
('Sofía', 'Díaz', 'sofia@profesor.com', 'Física'),  
('David', 'Torres', 'david@profesor.com', 'Química'),  
('Isabel', 'Rivera', 'isabel@profesor.com', 'Literatura');

-- Tabla Cursos

**INSERT INTO** Cursos

(nombre\_curso, descripción\_curso, profesor\_id) **Values**

Matemáticas Básicas, Un curso introductorio de matemáticas., 1

Historia del Arte, Explora la historia del arte a lo largo de los siglos.,<sup>2</sup>  
 Ciencias de la Computación, Un curso sobre programación y desarrollo de software.,<sup>3</sup>  
 Inglés Avanzado, Clases de inglés para hablantes **no** nativos.,<sup>1</sup>  
 Economía **101**, Introducción a los principios económicos.,<sup>4</sup>  
 Programación en Python, Aprende a programar en Python desde cero.,<sup>3</sup>  
 Arte Moderno, Explora el arte moderno y contemporáneo.,<sup>2</sup>  
 Física Aplicada, Un curso de física práctica para estudiantes de ciencias.,<sup>5</sup>  
 Química Orgánica, Estudia los compuestos químicos orgánicos.,<sup>3</sup>  
 Literatura Mundial, Lecturas y análisis de obras literarias de todo el mundo.,<sup>2</sup>  
 -- Tabla Foros\_y\_Comunicacion  
**INSERT INTO** Foros\_y\_Comunicacion  
 (usuario\_id, curso\_id, fecha\_publicacion, contenido\_foros\_comunicacion)  
**VALUES**  
 (1, 1, '2023-10-23 09:00:00', 'Este es un mensaje de ejemplo en el curso 1.'),  
 (2, 2, '2023-10-23 10:15:00', 'Mensaje en el curso 2 para discutir temas.'),  
 (3, 1, '2023-10-23 11:30:00', 'Otro mensaje en el curso 1 sobre el tema principal.'),  
 (4, 3, '2023-10-23 12:45:00', 'Mensaje en el curso 3 para intercambiar ideas.'),  
 (5, 2, '2023-10-23 14:00:00', 'Mensaje de prueba en el curso 2.'),  
 (6, 4, '2023-10-23 15:15:00', 'Este mensaje se publicó en el curso 4.'),  
 (7, 1, '2023-10-23 16:30:00', 'Mensaje en el curso 1 para debates abiertos.'),  
 (8, 5, '2023-10-23 17:45:00', 'Mensaje de discusión en el curso 5.'),  
 (9, 3, '2023-10-23 19:00:00', 'Otro mensaje en el curso 3 sobre temas importantes.'),  
 (10, 2, '2023-10-23 20:15:00', 'Mensaje final en el curso 2 para hoy.');  
 -- Tabla Contenido del Curso  
**INSERT INTO** Contenido\_del\_Curso  
 (curso\_id, tipo\_contenido, título\_contenido, descripción\_contenido, url) **VALUES**  
 (1, 'Video', 'Introducción al curso', 'Video de introducción al curso.',  
 'https://www.ejemplo.com/video1.mp4'),  
 (2, 'Documento', 'Lección 1', 'Documento sobre la primera lección.',  
 'https://www.ejemplo.com/leccion1.pdf'),  
 (3, 'Quiz', 'Prueba de conocimientos', 'Realiza esta prueba para evaluar tus conocimientos.',  
 'https://www.ejemplo.com/quiz1.html'),  
 (1, 'Video', 'Tema 1', 'Video sobre el primer tema del curso.',  
 'https://www.ejemplo.com/video2.mp4'),  
 (2, 'Documento', 'Lección 2', 'Documento sobre la segunda lección.',  
 'https://www.ejemplo.com/leccion2.pdf'),  
 (3, 'Quiz', 'Prueba final', 'Prueba final para evaluar lo aprendido en el curso.',  
 'https://www.ejemplo.com/quiz2.html'),  
 (1, 'Video', 'Resumen del curso', 'Video de resumen del curso.',  
 'https://www.ejemplo.com/video3.mp4'),  
 (2, 'Documento', 'Lección 3', 'Documento sobre la tercera lección.',  
 'https://www.ejemplo.com/leccion3.pdf'),  
 (3, 'Quiz', 'Examen final', 'Examen final para obtener la certificación.',  
 'https://www.ejemplo.com/examen.html');  
 -- Tabla Calificaciones

```
INSERT INTO calificaciones (estudiante_id, curso_id, calificacion_curso)
VALUES
```

```
(1, 1, 90.50),
(2, 2, 85.00),
(3, 1, 92.75),
(4, 3, 78.25),
(5, 2, 88.00),
(6, 4, 95.75),
(7, 1, 87.50),
(8, 5, 76.00),
(9, 3, 91.25),
(10, 2, 89.75);
```

```
-- Tabla Asistencia
```

```
INSERT INTO Asistencia (estudiante_id, curso_id, fecha_asistencia) VALUES
```

```
(1, 1, '2023-10-23 09:00:00'),
(2, 2, '2023-10-23 10:15:00'),
(3, 1, '2023-10-23 11:30:00'),
(4, 3, '2023-10-23 12:45:00'),
(5, 2, '2023-10-23 14:00:00'),
(6, 4, '2023-10-23 15:15:00'),
(7, 1, '2023-10-23 16:30:00'),
(8, 5, '2023-10-23 17:45:00'),
(9, 3, '2023-10-23 19:00:00'),
```

```
-- Tabla Notificaciones
```

```
INSERT INTO Notificaciones
```

```
(usuario_id, tipo_notificacion, contenido_notificacion, fecha_notificacion) VALUES
```

```
(1, 'Nueva tarea', 'Tienes una nueva tarea asignada.', '2023-10-23 09:00:00'),
(2, 'Recordatorio', 'Recordatorio de la próxima reunión.', '2023-10-23 10:15:00'),
(3, 'Nueva tarea', 'Tienes una nueva tarea para completar.', '2023-10-23 11:30:00'),
(4, 'Aviso importante', 'Aviso importante para todos los estudiantes.', '2023-10-23
12:45:00'),
(5, 'Nueva tarea', 'Tarea adicional para la próxima semana.', '2023-10-23 14:00:00'),
(6, 'Recordatorio', 'Recordatorio de la fecha límite de un proyecto.', '2023-10-23
15:15:00'),
(7, 'Aviso importante', 'Cambios en el horario de clases.', '2023-10-23 16:30:00'),
(8, 'Nueva tarea', 'Nueva tarea de investigación asignada.', '2023-10-23 17:45:00'),
(9, 'Recordatorio', 'Recordatorio de la reunión de padres.', '2023-10-23 19:00:00'),
(10, 'Aviso importante', 'Cambios en el calendario académico.', '2023-10-23 20:15:00'),
```

## 9. Vista Principales

Usuarios con filtros **WHERE**

```
CREATE VIEW VistaUsuarios_Filtrados AS
```

```
SELECT id_usuario, nombre_usuario, apellido_usuario, correo_usuario, rol_usuario  
FROM Usuarios
```

```
WHERE rol_usuario = 'estudiante';
```

Cursos con **LIMIT**

```
CREATE VIEW VistaCursos_Limit AS
```

```
SELECT id_curso, nombre_curso, descripcion_curso  
FROM Cursos
```

```
LIMIT 5;
```

Promedio de calificaciones por curso con **GROUP BY**

```
CREATE VIEW VistaPromedio_Calificaciones AS
```

```
SELECT Cursos.id_curso, nombre_curso, AVG(calificacion_curso) as  
promedio_calificaciones
```

```
FROM Cursos
```

```
INNER JOIN Calificaciones ON Cursos.id_curso = Calificaciones.curso_id
```

```
GROUP BY Cursos.id_curso, nombre_curso;
```

Estudiantes y sus cursos con **INNER JOIN**

```
CREATE VIEW VistaEstudiantes_Cursos AS
```

```
SELECT Estudiantes.id_estudiante, nombre_estudiante, apellido_estudiante, nombre_curso  
FROM Estudiantes
```

```
INNER JOIN Calificaciones ON Estudiantes.id_estudiante = Calificaciones.estudiante_id
```

```
INNER JOIN Cursos ON Calificaciones.curso_id = Cursos.id_curso;
```

Notificaciones por usuario con filtro **WHERE**

```
CREATE VIEW VistaNotificaciones_PorUsuario AS
```

```
SELECT Usuarios.id_usuario, nombre_usuario, tipo_notificacion, fecha_notificacion  
FROM Usuarios
```

```
INNER JOIN Notificaciones ON Usuarios.id_usuario = Notificaciones.usuario_id
```

```
WHERE tipo_notificacion = 'Nueva tarea';
```



## 10. Funciones

USE Integracion\_Educa\_Chile;

/\*

*Funciones: CalcularPromedioCalificacionesEstudiante*

*Descripción: Calcula el promedio de calificaciones para un estudiante dado.*

\*/

DELIMITER //

CREATE FUNCTION CalcularPromedioCalificacionesEstudiante(estudiante\_id\_param  
INT) RETURNS DECIMAL(5, 2)

DETERMINISTIC

BEGIN

DECLARE promedio DECIMAL(5, 2);

-- Utilizando INNER JOIN para obtener calificaciones del estudiante específico

SELECT AVG(calificacion\_curso) INTO promedio

FROM Calificaciones AS cal

INNER JOIN Estudiantes AS est ON cal.estudiante\_id = est.id\_estudiante

WHERE est.id\_estudiante = estudiante\_id\_param;

RETURN promedio;

END //

DELIMITER ;

-- Calcular el promedio de calificaciones segun tus necesidades

SELECT CalcularPromedioCalificacionesEstudiante(2) AS

PromedioCalificacionesEstudiante;

/\*

-- Crear función para obtener el total de notificaciones de un tipo específico

-- Obtener el total de notificaciones de tipo 'aviso importante'

\*/

DELIMITER //

CREATE FUNCTION ObtenerTotalNotificacionesPorTipo(tipo\_notificacion\_param  
VARCHAR(100)) RETURNS INT

DETERMINISTIC

BEGIN

-- Variable para almacenar el total de notificaciones

DECLARE total INT;

-- Utilizar COUNT para contar el número de notificaciones del tipo específico

SELECT COUNT(\*) INTO total

FROM Notificaciones

WHERE tipo\_notificacion = tipo\_notificacion\_param;

-- Devolver el total de notificaciones

RETURN total;

END //

DELIMITER ;

```
-- Puedes cambiar el tipo de notificación según tus necesidades
SELECT ObtenerTotalNotificacionesPorTipo('aviso importante') AS
TotalAvisosImportantes;
```

## 11. Stored Procedures

```
USE Integracion_Educa_Chile;
/*
```

*Stored Procedures para ordenar una tabla*

*Ordenar una tabla según un campo específico y especificar si el orden es ascendente o descendente.*

```
*/
```

```
DELIMITER //
```

```
CREATE PROCEDURE OrdenarTabla(
    IN tabla_nombre VARCHAR(100),
    IN campo_ordenamiento VARCHAR(100),
    IN orden VARCHAR(10)
)
```

```
BEGIN
```

```
-- Variable para construir la consulta SQL dinámicamente
```

```
SET @sql_query = CONCAT('SELECT * FROM ', tabla_nombre, ' ORDER BY ',
campo_ordenamiento, ' ', orden);
```

```
-- Ejecutar la consulta SQL dinámica
```

```
PREPARE stmt FROM @sql_query;
```

```
EXECUTE stmt;
```

```
DEALLOCATE PREPARE stmt;
```

```
END //
```

```
DELIMITER ;
```

```
-- Uso del S.P. OrdenarTabla
```

```
CALL OrdenarTabla('Usuarios', 'nombre_usuario', 'ASC');
```

```
/*
```

*Stored Procedure para gestionar registros en una tabla*

*Stored Procedure permite insertar o eliminar registros en una tabla específica del proyecto.*

```
*/
```

```
DELIMITER //
```

```
CREATE PROCEDURE GestionarRegistros(
    IN accion VARCHAR(10),
    IN tabla_nombre VARCHAR(100),
    IN id_registro INT
)
```

```
BEGIN
```

```
-- Variable para construir la consulta SQL dinámicamente
```

```

SET @sql_query = "";
-- Determinar la acción a realizar (insertar o eliminar)
CASE accion
    WHEN 'INSERT' THEN
        SET @sql_query = CONCAT('INSERT INTO ', tabla_nombre, ' (nombre_usuario,
apellido_usuario, correo_usuario, contraseña_usuario, rol_usuario) VALUES
("NuevoNombre", "NuevoApellido", "nuevo@correo.com", "nuevacontraseña",
"NuevoRol");');
    WHEN 'DELETE' THEN
        SET @sql_query = CONCAT('DELETE FROM ', tabla_nombre, ' WHERE
id_usuario = ', id_registro, ');');
END CASE;
-- Ejecutar la consulta SQL dinámica
PREPARE stmt FROM @sql_query;
EXECUTE stmt;
DEALLOCATE PREPARE stmt;
END //
DELIMITER ;
-- Uso del S.P. GestionarRegistros para insertar un nuevo registro
CALL GestionarRegistros('INSERT', 'Usuarios', NULL);
-- Uso del S.P. GestionarRegistros para eliminar un registro específico
CALL GestionarRegistros('DELETE', 'Usuarios', 11); -- Cambia el valor 1 por el ID del
registro que deseas eliminar

```

## 12. Triggers

```
/*
Triggers: En esta tabla se utiliza para registrar acciones realizadas en la tabla Usuarios.
SHOW TRIGGERS;
DROP TRIGGER IF EXISTS After_insert_usuarios;
*/
-- Tabla de Log para Usuarios (Log_Usuarios):
CREATE TABLE IF NOT EXISTS Log_Usuarios (
  id_log INT AUTO_INCREMENT PRIMARY KEY,
  accion_realizada VARCHAR(50),
  usuario_modificador VARCHAR(255),
  fecha_modificacion DATE,
  hora_modificacion TIME
);

/*
Este trigger se activa antes de que se realice una inserción en la tabla Usuarios.
*/
-- Trigger BEFORE para la tabla Usuarios (before_insert_usuarios):
DELIMITER //
CREATE TRIGGER before_insert_usuarios
BEFORE INSERT ON Usuarios
FOR EACH ROW
BEGIN
  -- Registra la información antes de la inserción
  INSERT INTO Log_Usuarios ( accion_realizada, usuario_modificador,
fecha_modificacion, hora_modificacion)
VALUES ( 'INSERT BEFORE USUARIOS', CURRENT_USER(),
CURRENT_DATE(), CURRENT_TIME());
END;
//
DELIMITER ;

/*
Este trigger se activa después de una inserción en la tabla Usuarios.
Se puede realizar acciones adicionales aquí si es necesario.
*/
-- Trigger AFTER para la tabla Usuarios (after_insert_usuarios):
DELIMITER //
CREATE TRIGGER after_insert_usuarios
AFTER INSERT ON Usuarios
FOR EACH ROW
BEGIN
  -- Insertar información en una tabla de registro (log)
```

```

    INSERT INTO Log_Usuarios ( accion_realizada, usuario_modificador,
fecha_modificacion, hora_modificacion)
    VALUES ( 'INSERT AFTER USUARIOS', CURRENT_USER(), CURRENT_DATE(),
CURRENT_TIME());
END;
//
DELIMITER ;

/*
    Esta tabla es similar a la tabla de log para usuarios, pero está destinada a las acciones
realizadas en la tabla Cursos.
*/
-- Tabla de Log para Cursos (Log_Cursos):
CREATE TABLE IF NOT EXISTS Log_Cursos (
    id_log INT AUTO_INCREMENT PRIMARY KEY,
    accion_realizada VARCHAR(50),
    usuario_modificador VARCHAR(255),
    fecha_modificacion DATE,
    hora_modificacion TIME
);

/*
    Trigger BEFORE para la tabla Cursos (before_update_cursos):
    Este trigger se activa antes de que se realice una actualización en la tabla Cursos.
    Captura la información relevante (ID de curso, acción 'UPDATE', usuario que realiza la
acción, fecha y hora) y la inserta en la tabla de log Log_Cursos.
*/
DELIMITER //
CREATE TRIGGER before_update_cursos
BEFORE UPDATE ON Cursos
FOR EACH ROW
BEGIN
    -- Registra la información antes de la actualización
    INSERT INTO Log_Cursos ( accion_realizada, usuario_modificador,
fecha_modificacion, hora_modificacion)
    VALUES ( 'UPDATE CURSOS ACTUALIZACION', CURRENT_USER(),
CURRENT_DATE(), CURRENT_TIME());
END;
//
DELIMITER ;

```

### 13. Usuarios Permisos

```
-- Crear un nuevo usuario llamado 'usuario_lectura' con contraseña 'lectura'
CREATE USER 'usuario_lectura'@'localhost' IDENTIFIED BY 'lectura';

-- Permisos de solo lectura en todas las tablas de la base de datos
'Integracion_Educa_Chile' al usuario 'usuario_lectura' desde 'localhost'
GRANT SELECT ON Integracion_Educa_Chile.* TO 'usuario_lectura'@'localhost';

-- Mostrar información sobre los usuarios en la base de datos 'mysql'
-- Ayuda a verificar si el usuario 'usuario_lectura' se creó correctamente
SELECT * FROM mysql.user;

-- Crear un nuevo usuario llamado 'lectura_insercion_modificacion' con contraseña '123'
CREATE USER 'lectura_insercion_modificacion'@'localhost' IDENTIFIED BY
'123';
```

### 14. Transacciones

```
-- Iniciar una transacción para la tabla Profesores
START TRANSACTION;

-- Eliminar registros de la tabla Profesores si existen
DELETE FROM Profesores WHERE id_profesor IN (1, 2, 3);

-- Guardar un punto de guardado (Savepoint) después de la eliminación
SAVEPOINT despues_eliminar;

-- Sentencia Rollback comentada (sin ejecutar)
-- ROLLBACK TO despues_eliminar;

-- Sentencia Commit comentada (sin ejecutar)
-- COMMIT;

-- Iniciar una transacción para la tabla Cursos
START TRANSACTION;

-- Insertar ocho nuevos registros en la tabla Cursos
INSERT INTO Cursos (nombre_curso, descripcion_curso, profesor_id) VALUES
('Curso1', 'Descripción1', 1),
('Curso2', 'Descripción2', 2),
('Curso3', 'Descripción3', 3),
('Curso4', 'Descripción4', 4),
('Curso5', 'Descripción5', 5),
('Curso6', 'Descripción6', 6),
('Curso7', 'Descripción7', 7),
('Curso8', 'Descripción8', 8);

-- Guardar un punto de guardado (Savepoint) después de la inserción del registro #4
```

```
SAVEPOINT despues_insertar_4;

-- Guardar un punto de guardado (Savepoint) después de la inserción del registro #8
SAVEPOINT despues_insertar_8;

-- Sentencia para eliminar el Savepoint después de la inserción de los primeros 4 registros
(comentada)
-- RELEASE SAVEPOINT despues_insertar_4;

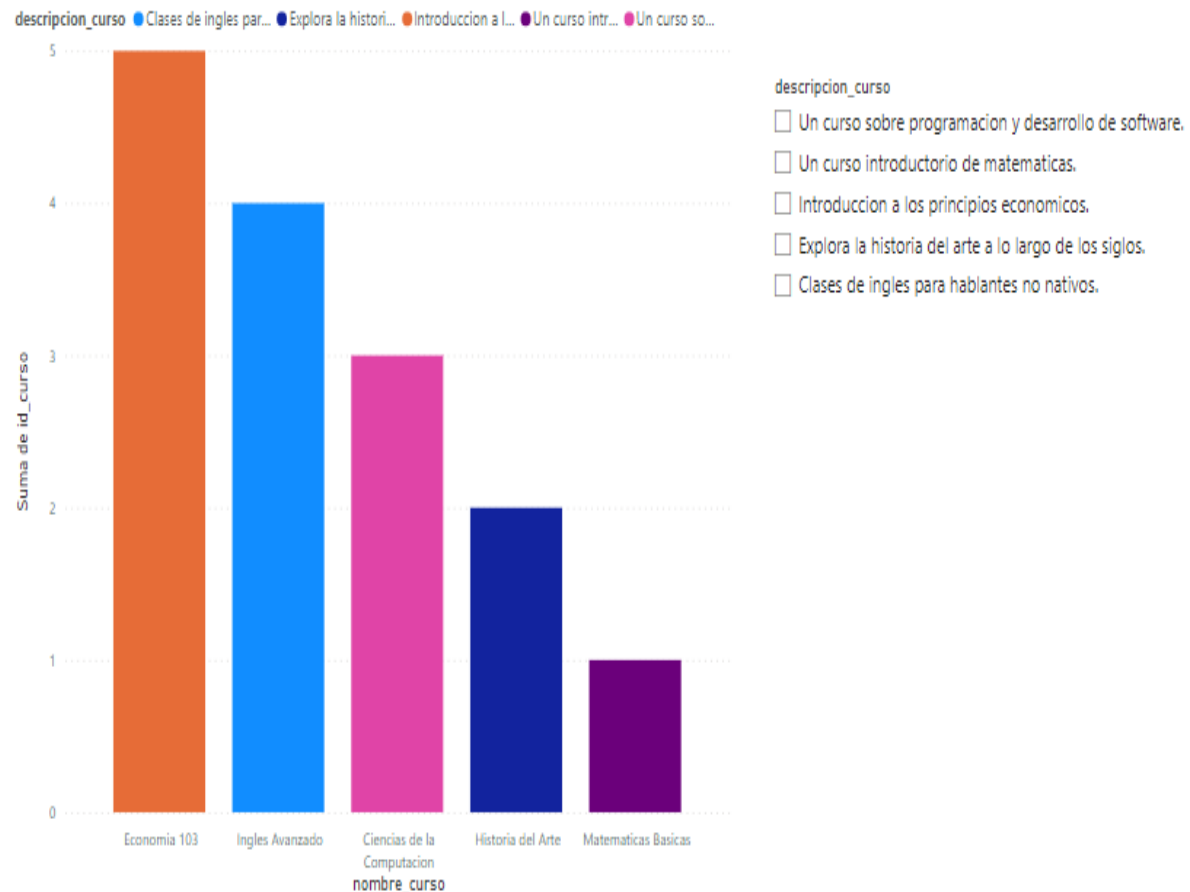
-- Confirmar la transacción
COMMIT;
```

## 15. Reportes Power BI

En Power BI, se generará un informe que destaca la información sobre cursos con límites específicos. Este reporte proporcionará una visión detallada de los cursos, centrándose en aquellos que tienen restricciones establecidas. La presentación en Power BI garantizará una visualización efectiva, permitiendo un análisis claro y conciso de los detalles asociados a cada curso y sus respectivos límites.

### viewscursos\_limit

Suma de id\_curso por nombre\_curso y descripcion\_curso

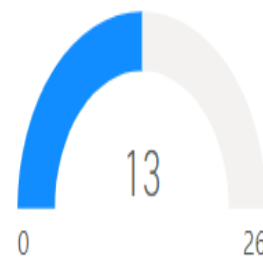
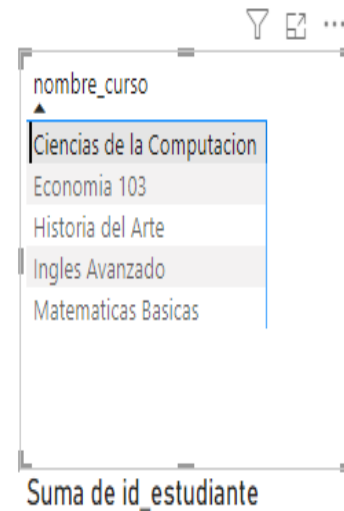
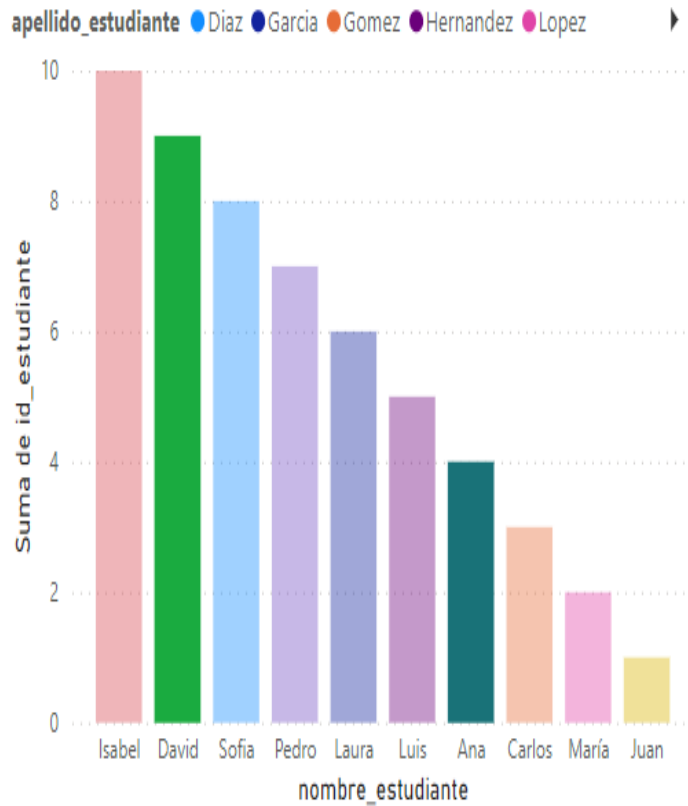




En Power BI, se elaborará un informe resaltando detalles relacionados con cursos y estudiantes. Este reporte ofrecerá una visión detallada de la interacción entre estudiantes y cursos, brindando información clave sobre la dinámica de la relación estudiantes-cursos.

### viewsestudiantes\_cursos

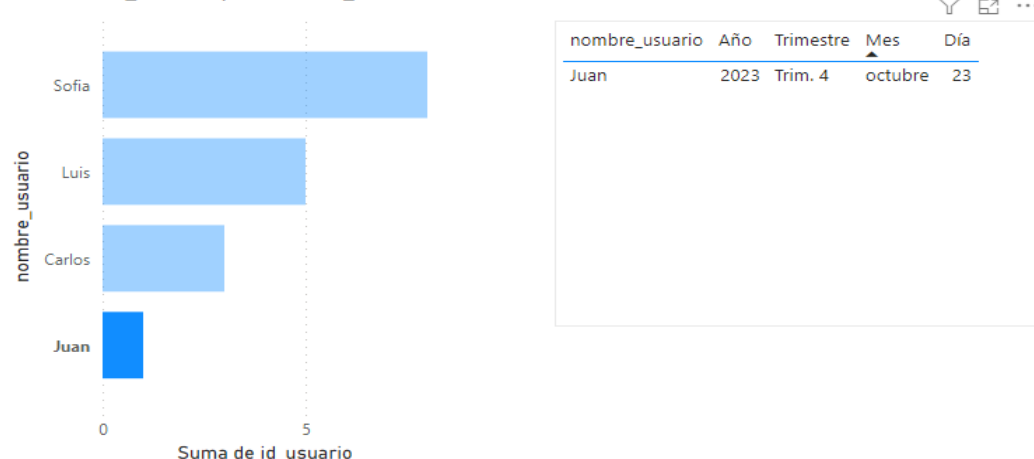
Suma de id\_estudiante por nombre\_estudiante y apellido\_estudiante



En Power BI, se creará un informe que resalta la información de notificaciones junto con los usuarios correspondientes. Este informe ofrecerá una visión detallada de las vistas de notificaciones por usuario, proporcionando una comprensión exhaustiva de la relación entre las notificaciones y los usuarios específicos.

### viewsnotificaciones\_porusuario

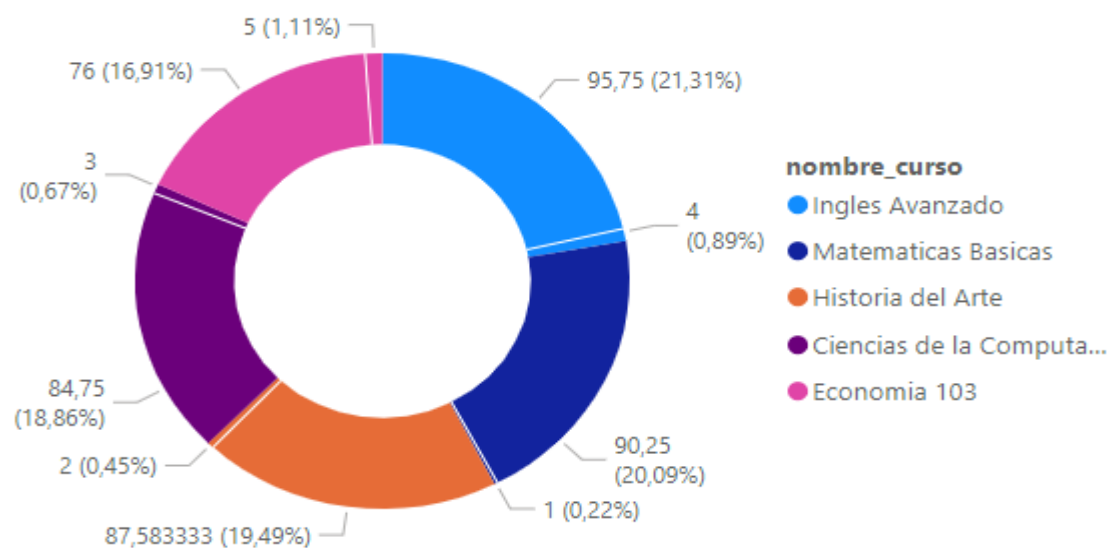
Suma de id\_usuario por nombre\_usuario



En Power BI, se elaborará un informe que resalta el promedio de calificaciones. Este reporte ofrecerá una visión detallada de las vistas que contienen información sobre el promedio de calificaciones, proporcionando una perspectiva exhaustiva de las evaluaciones en cuestión.

### viewspromedio\_calificaciones

Suma de promedio\_calificaciones y Suma de id\_curso por nombre\_curso



## 16. Herramientas utilizadas

Las herramientas empleadas en el proyecto fueron las siguientes:

1. **Draw.io:** Utilizado para la creación de diagramas y visualización de procesos.
2. **Trello:** Plataforma de gestión de proyectos basada en tableros Kanban para organizar tareas.
3. **Stack Overflow:** Recurso invaluable para buscar soluciones a problemas y obtener orientación en el desarrollo.
4. **DBeaver:** Herramienta de administración de bases de datos que facilita la conexión y manipulación de datos en diversas fuentes.
5. **Command-Line Client:** Empleado para ejecutar comandos directamente desde la línea de comandos, proporcionando flexibilidad en la gestión del proyecto.
6. **Visual Studio:** Utilizado como entorno de desarrollo integrado (IDE) para escribir, depurar y desplegar código de manera eficiente.
7. **Git Hub:** Plataforma esencial de control de versiones que facilita la colaboración y el seguimiento de cambios en el código fuente. Se utilizará para alojar y gestionar repositorios, relacionados con el proyecto, así como para llevar a cabo la revisión del código entre los miembros del equipo.

## 17. Comentarios Finales

El proyecto de Integración Educa Chile DB SQL se presenta como una solución integral para abordar desafíos cruciales en el entorno educativo. A medida que avanzamos en la implementación de esta base de datos, es esencial mantener un enfoque colaborativo y receptivo a las necesidades cambiantes de la comunidad escolar. La retroalimentación continua de los usuarios, tanto profesionales como estudiantes, será fundamental para perfeccionar y expandir la funcionalidad de la herramienta.

Además, se sugiere la implementación de sesiones de capacitación periódicas para garantizar que todos los usuarios puedan aprovechar al máximo las capacidades de la base de datos. Esto no solo promoverá la adopción efectiva de la herramienta, sino que también empoderará a la comunidad escolar para utilizarla como un recurso valioso en su día a día.

El éxito de este proyecto no solo se medirá por la eficiencia operativa y la optimización de la toma de decisiones, sino también por su impacto positivo en la calidad de la educación y la fortaleza de la comunidad escolar. Mantener un diálogo abierto con todas las partes interesadas será esencial para adaptar la base de datos a las necesidades cambiantes y garantizar su continuo valor a lo largo del tiempo.