

UNIVERSIDAD PRIVADA DEL NORTE



**UPN**  
UNIVERSIDAD  
PRIVADA  
DEL NORTE

**CONTENIDO:**

**INFORME**

# FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

---

<<Desarrollo de una Plataforma de Gestión Académica para el Instituto CIMAS:  
Un Enfoque en Accesibilidad y Precios Accesibles.>>

## **Autores:**

FERNANDEZ GALLO, ANGELA GUADALUPE (% 100)

GONZALES DURAND, BRANN DY NYCOLE (% 100)

LLUNCOR VILCHES, CRISTHIAN ANDRE (% 100)

GOMEZ CALDERON, JEFFERSON ANDY (% 100)

QUISPE HUARCAYA, VICTOR JOSE (% 100)

## **Curso:**

Base de Datos

## **Docente:**

CESAR EDINSON AGUIRRE RODRIGUEZ

Chorrillos – Perú

2024-1

## CONTENIDO

|   |    |
|---|----|
| CONTENIDO.....  | 2  |
| RESUMEN .....   | 4  |
| ABSTRACT .....  | 5  |
| 1. CAPÍTULO I: INTRODUCCIÓN .....   | 6  |
| 1.1. MOTIVACIÓN DEL PROYECTO.....   | 6  |
| 1.2. PROPUESTAS .....   | 6  |
| 1.2.1. Propuesta 01: Sistema de Gestión Académica Tradicional .....                 | 6  |
| 1.2.2. Propuesta 02: Uso de Software de Gestión de Cursos Externo .....             | 6  |
| 1.2.3. Propuesta 03: Desarrollo de una Plataforma Propia Similar a Blackboard ..... | 6  |
| 1.2.4. IMPACTOS .....   | 6  |
| 1.2.4.1. Impacto social .....   | 7  |
| 1.2.4.2. Impacto cultural .....   | 7  |
| 1.2.4.3. Impacto político .....   | 7  |
| 1.2.4.4. Impacto ambiental .....  | 7  |
| 1.2.4.5. Impacto ético .....  | 7  |
| 1.2.4.6. Impacto económico .....  | 7  |
| 2. CAPÍTULO II: MARCO TEÓRICO .....   | 8  |
| 2.1. ANTECEDENTES TEÓRICOS .....  | 8  |
| 2.2. BASES TEÓRICAS .....   | 8  |
| 2.3. ANÁLISIS DEL PROBLEMA .....  | 9  |
| 2.3.1. IDENTIFICACIÓN Y FORMULACION DEL PROBLEMA .....                              | 9  |
| 2.3.2. DEFINICIÓN DE OBJETIVOS .....  | 10 |
| Objetivo general .....  | 10 |
| Objetivos específicos.....  | 10 |
| 2.3.3. ALCANCE DE LA SOLUCIÓN .....   | 10 |
| 3. CAPITULO III: HERRAMIENTAS DE INGENIERÍA .....                                   | 11 |
| 3.1. Herramientas de Desarrollo .....   | 11 |
| 3.2. Herramientas de Gestión de Proyectos .....                                     | 11 |
| 4. CAPITULO IV: GENERACIÓN DE SOLUCIONES.....                                       | 12 |
| 4.1. Alternativas de Solución .....   | 12 |
| 4.2. Cronograma del Proyecto .....  | 12 |
| 4.3. Modelado del Negocio .....   | 13 |
| 5. CAPÍTULO V: METODOLOGÍA DE DESARROLLO .....                                      | 14 |
| MIGRACION AL GESTOR DE BASE DE DATOS .....  | 22 |

|   |    |
|---|----|
| POLITICAS DE SEGURIDAD DE USUARIO .....                 | 26 |
| PLAN DE BACKUPS Y RECUPERACION DE FALLOS .....          | 30 |
| 6.    CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES ..... | 34 |
| 6.1.    CONCLUSIONES .....                              | 34 |
| 6.2.    RECOMENDACIONES.....                            | 34 |
| 7.    INDICES .....                                     | 36 |

## RESUMEN

El presente trabajo tiene como objetivo diseñar y desarrollar una base de datos para el Instituto de Educación Superior María de Los Ángeles CIMA'S (CIMAS), una institución con 34 años de experiencia y reconocido prestigio en Lima. La base de datos permitirá gestionar eficientemente la administración académica, facilitando el registro de alumnos en cursos accesibles y mejorando la experiencia educativa tanto para alumnos como para docentes. Este proyecto surge de la necesidad de modernizar y optimizar la administración educativa del instituto, proporcionando acceso fácil a la información a través de una plataforma similar a Blackboard.

La institución, licenciada bajo la R.M. 442-2023, se dedica a formar profesionales emprendedores altamente calificados, comprometidos con el cambio en el mundo laboral. Nuestra misión es ofrecer educación de calidad, consolidando nuestra imagen como líderes en la formación de profesionales competentes y emprendedores. Nos orientamos hacia carreras profesionales y de extensión con un enfoque empresarial y humanístico, respondiendo a las necesidades del mercado con integridad, innovación, trabajo en equipo, y valores de justicia, honestidad y respeto.

Palabras clave: base de datos, gestión académica, Blackboard, Instituto CIMAS, administración educativa.

## ABSTRACT

The objective of this work is to design and develop a database for the María de Los Ángeles CIMA'S Higher Education Institute (CIMAS), an institution with 34 years of experience and recognized prestige in Lima. The database will allow efficient management of academic administration, facilitating the registration of students in accessible courses and improving the educational experience for both students and teachers. This project arises from the need to modernize and optimize the educational administration of the institute, providing easy access to information through a platform similar to Blackboard.

The institution, licensed under the R.M. 442-2023, is dedicated to training highly qualified entrepreneurial professionals, committed to change in the world of work. Our mission is to offer quality education, consolidating our image as leaders in the training of competent and entrepreneurial professionals. We are oriented towards professional and extension careers with a business and humanistic approach, responding to the needs of the market with integrity, innovation, teamwork, and values of justice, honesty and respect.

Keywords: database, academic management, Blackboard, CIMAS Institute, educational administration.

## 1. CAPÍTULO I: INTRODUCCIÓN

### 1.1. MOTIVACIÓN DEL PROYECTO

La motivación principal para la elección de este proyecto es la necesidad de modernizar y optimizar la gestión académica en el Instituto CIMAS. Actualmente, el instituto enfrenta desafíos relacionados con la eficiencia en la administración de cursos y alumnos, la accesibilidad de la información y la mejora de la experiencia educativa. Implementar una plataforma similar a Blackboard permitirá resolver estas problemáticas, proporcionando un sistema centralizado y eficiente que beneficie tanto a estudiantes como a profesores.

### 1.2. PROPUESTAS

#### 1.2.1. Propuesta 01: Sistema de Gestión Académica Tradicional

- Ventajas:
  - Familiaridad con métodos tradicionales de administración.
  - Menor costo inicial de implementación.
- Desventajas:
  - Baja eficiencia y propensión a errores humanos.
  - Dificultad en la gestión de grandes volúmenes de datos.
  - Acceso limitado y no centralizado.

#### 1.2.2. Propuesta 02: Uso de Software de Gestión de Cursos Externo

- Ventajas:

Implementación rápida con herramientas ya desarrolladas.  
Actualizaciones y soporte por parte del proveedor.

- Desventajas:
  - Dependencia de terceros y posible falta de personalización.
  - Costos recurrentes por licencias y suscripciones.

#### 1.2.3. Propuesta 03: Desarrollo de una Plataforma Propia Similar a Blackboard

- Ventajas:
  - Total, personalización según las necesidades del instituto.
  - Integración directa con procesos y sistemas existentes.
  - Mejora significativa en la administración y accesibilidad de la información.
- Desventajas:
  - Mayor costo inicial de desarrollo e implementación.
  - Requiere personal especializado para desarrollo y mantenimiento.

#### 1.2.4. IMPACTOS

Describir los impactos de que generará su Proyecto en los diferentes contextos.

#### **1.2.4.1. Impacto social**

La implementación de una base de datos eficiente mejorará la calidad educativa y el acceso a la información, beneficiando a estudiantes y docentes del instituto.

#### **1.2.4.2. Impacto cultural**

Facilitar el acceso a cursos y recursos educativos contribuirá al enriquecimiento cultural y académico de la comunidad del Instituto CIMAS.

#### **1.2.4.3. Impacto político**

Un sistema educativo mejor gestionado fortalece la imagen institucional y puede influir positivamente en políticas educativas locales.

#### **1.2.4.4. Impacto ambiental**

Digitalizar procesos académicos reduce el uso de papel y otros recursos físicos, contribuyendo a prácticas más sostenibles.

#### **1.2.4.5. Impacto ético**

La transparencia y eficiencia en la gestión educativa promueve prácticas éticas y responsables dentro del instituto.

#### **1.2.4.6. Impacto económico**

Optimizar la administración educativa puede reducir costos operativos y mejorar la asignación de recursos, generando ahorros a largo plazo.



## 2. CAPÍTULO II: MARCO TEÓRICO

### 2.1. ANTECEDENTES TEÓRICOS

El diseño y la implementación de bases de datos son aspectos fundamentales en el desarrollo de sistemas de información. La teoría de bases de datos se centra en la organización, almacenamiento y recuperación de datos, y su correcto manejo es crucial para el éxito de cualquier sistema informático. La modelación de bases de datos relacionales, basada en el modelo de datos relacional propuesto por Edgar F. Codd en 1970, es una de las técnicas más utilizadas. Este modelo organiza los datos en tablas (relaciones), lo que permite una manipulación flexible y eficiente mediante el uso de SQL (Structured Query Language).

### 2.2. BASES TEÓRICAS

#### **Modelamiento de Bases de Datos Relacionales:**

- *Concepto de Modelo Relacional:* El modelo relacional se basa en la teoría de conjuntos y proporciona una representación tabular de los datos. Las tablas consisten en filas y columnas, donde cada fila representa un registro y cada columna, un atributo del registro.
- *Normalización:* Proceso de organizar los datos en una base de datos para reducir la redundancia y mejorar la integridad. Las formas normales son etapas de este proceso, desde la Primera Forma Normal (1NF) hasta la Quinta Forma Normal (5NF).
- *Diagramas Entidad-Relación (ER):* Herramienta de modelado que permite visualizar la estructura de la base de datos mediante entidades (tablas) y sus relaciones.

#### **Implementación en SGBD (Sistemas de Gestión de Bases de Datos):**

- *MS SQL Server:* Es un sistema de gestión de bases de datos relacional desarrollado por Microsoft. Proporciona un entorno robusto para el almacenamiento y gestión de datos, con características avanzadas de seguridad, replicación y recuperación ante desastres.
- *SQL (Structured Query Language):* Lenguaje estándar para el manejo de bases de datos relacionales. Incluye comandos para la definición (DDL), manipulación (DML) y control de datos (DCL).

#### **Desarrollo de Aplicaciones:**

- *C# y Java:* Lenguajes de programación ampliamente utilizados para el desarrollo de aplicaciones de escritorio y web. Ambos lenguajes son orientados a objetos y

*proporcionan una fuerte tipificación, lo que mejora la fiabilidad y mantenibilidad del código.*

- *IDE (Entornos de Desarrollo Integrados): Herramientas como Visual Studio y Eclipse facilitan el desarrollo de software al integrar editores de código, depuradores y herramientas de gestión de proyectos en una sola interfaz.*

### **Metodologías Ágiles:**

*Scrum: Marco de trabajo ágil para la gestión de proyectos que permite la entrega incremental y continua de productos. Incluye roles específicos (Scrum Master, Product Owner, Equipo de Desarrollo) y eventos (Sprint, Daily Scrum, Sprint Review, Sprint Retrospective) para facilitar la colaboración y mejorar la productividad.*

### **Evaluación de Alternativas:**

*La elección de una solución tecnológica debe considerar tanto los beneficios inmediatos como los costos a largo plazo. En el caso del Instituto CIMAS, se evaluaron tres alternativas: un sistema de gestión académica tradicional, el uso de software de gestión de cursos externo, y el desarrollo de una plataforma propia similar a Blackboard. La evaluación incluyó aspectos como la eficiencia, la personalización, los costos y la dependencia de terceros.*

*Estas bases teóricas proporcionan el fundamento necesario para el diseño, implementación y gestión eficaz de bases de datos y sistemas de información. A través del uso adecuado de estas teorías y herramientas, se pueden desarrollar soluciones robustas y eficientes que mejoren la administración y accesibilidad de la información en instituciones educativas como el Instituto CIMAS.*

## **2.3. ANÁLISIS DEL PROBLEMA**

### **2.3.1. IDENTIFICACIÓN Y FORMULACION DEL PROBLEMA**

El Instituto de Educación Superior María de Los Ángeles CIMA'S es una institución de reconocido prestigio, cuyo propósito es formar profesionales emprendedores altamente calificados y que sean agentes de cambio en el mundo laboral. Fundada en la ciudad de Lima y licenciada con la R.M. 442-2023, el instituto se destaca por su compromiso con la excelencia académica y la innovación educativa.

El problema radica en la falta de un sistema centralizado y eficiente para administrar la información de cursos y estudiantes. Actualmente, los datos se manejan de manera dispersa y principalmente de forma manual, lo que conduce a errores en los registros, ineficiencias en la comunicación entre departamentos y una dificultad considerable en el acceso a datos críticos para la toma de decisiones. Como resultado, se experimentan

retrasos en la matriculación, asignación incorrecta de recursos y una experiencia estudiantil menos óptima.

Una alternativa viable para resolver estos problemas sería la implementación de un sistema de gestión académica integrado. Este sistema centralizado permitiría la automatización de procesos clave como la matriculación, la gestión de horarios y la generación de reportes académicos. Además, facilitaría el acceso seguro y eficiente a la información, mejorando la transparencia y la colaboración entre departamentos. Mediante la adopción de tecnologías modernas, el Instituto CIMAS podría mejorar la experiencia tanto para estudiantes como para el personal administrativo, asegurando un flujo de trabajo más eficiente y una gestión académica más efectiva

### 2.3.2. DEFINICIÓN DE OBJETIVOS

#### Objetivo general

Desarrollar una base de datos centralizada y eficiente para la gestión académica del Instituto CIMAS.

#### Objetivos específicos

- Facilitar el registro y administración de estudiantes y cursos.
- Mejorar la accesibilidad y disponibilidad de la información académica.
- Optimizar los procesos administrativos relacionados con la gestión educativa.

### 2.3.3. ALCANCE DE LA SOLUCIÓN

- La solución propuesta se enfocará en desarrollar una base de datos que permita la gestión integral de alumnos, cursos, y recursos educativos. No se cubrirán aspectos externos a la gestión académica directa, como la administración financiera del instituto.

### 3. CAPITULO III: HERRAMIENTAS DE INGENIERÍA

En este capítulo se describirán las herramientas de ingeniería que se utilizarán para el diseño y desarrollo de la base de datos y la plataforma similar a Blackboard para el Instituto CIMAS.

#### 3.1. Herramientas de Desarrollo

- **Sistemas de Gestión de Bases de Datos (DBMS):** Se utilizará **MySQL** debido a su fiabilidad, rendimiento y popularidad en la gestión de bases de datos relacionales.
- **Entornos de Desarrollo Integrados (IDE):** **Visual Studio Code** será el principal IDE para la codificación, aprovechando su amplia gama de extensiones y soporte para múltiples lenguajes de programación.
- **Lenguajes de Programación:** Se emplearán **PHP** para la lógica del servidor y **JavaScript** junto con **HTML** y **CSS** para el desarrollo del frontend.
- **Frameworks de Desarrollo Web:** **Laravel** para el backend, proporcionando una estructura robusta y segura para la aplicación web, y **Bootstrap** para el frontend, facilitando el diseño de una interfaz de usuario receptiva y moderna.
- **Herramientas de Diseño de Bases de Datos:** Se utilizará **MySQL Workbench** para la modelación y diseño visual de la base de datos.
- **Sistemas de Control de Versiones:** **Git** será la herramienta de control de versiones para gestionar el código fuente del proyecto.

#### 3.2. Herramientas de Gestión de Proyectos

- **Metodologías Ágiles:** Se empleará **Scrum** para la gestión del proyecto, permitiendo un desarrollo iterativo y una entrega continua de funcionalidades.
- **Software de Gestión de Proyectos:** **Jira** será utilizado para la gestión de tareas, la asignación de recursos y el seguimiento del progreso del proyecto.
- **Herramientas de Comunicación y Colaboración:** **Slack** y **Trello** se utilizarán para facilitar la comunicación y colaboración entre los miembros del equipo de desarrollo.

## 4. CAPITULO IV: GENERACIÓN DE SOLUCIONES

### 4.1. Alternativas de Solución

Se evaluaron tres propuestas para mejorar la gestión académica del Instituto CIMAS:

- **Propuesta 01: Sistema de Gestión Académica Tradicional**
  - **Ventajas:**
    - Familiaridad con métodos tradicionales de administración.
    - Menor costo inicial de implementación.
  - **Desventajas:**
    - Baja eficiencia y propensión a errores humanos.
    - Dificultad en la gestión de grandes volúmenes de datos.
    - Acceso limitado y no centralizado.
- **Propuesta 02: Uso de Software de Gestión de Cursos Externo**
  - **Ventajas:**
    - Implementación rápida con herramientas ya desarrolladas.
    - Actualizaciones y soporte por parte del proveedor.
  - **Desventajas:**
    - Dependencia de terceros y posible falta de personalización.
    - Costos recurrentes por licencias y suscripciones.
- **Propuesta 03: Desarrollo de una Plataforma Propia Similar a Blackboard**
  - **Ventajas:**
    - Total, personalización según las necesidades del instituto.
    - Integración directa con procesos y sistemas existentes.
    - Mejora significativa en la administración y accesibilidad de la información.
  - **Desventajas:**
    - Mayor costo inicial de desarrollo e implementación.
    - Requiere personal especializado para desarrollo y mantenimiento.

### 4.2. Cronograma del Proyecto

Se establecerá un cronograma detallado para cada fase del proyecto, incluyendo:

- **Fase 1: Análisis y Recolección de Requerimientos**
  - Duración: 2 semanas
- **Fase 2: Diseño de la Base de Datos y la Plataforma**
  - Duración: 4 semanas
- **Fase 3: Desarrollo e Implementación**
  - Duración: 8 semanas
- **Fase 4: Pruebas y Validación**
  - Duración: 3 semanas
- **Fase 5: Implementación y Capacitación**
  - Duración: 2 semanas
- **Fase 6: Mantenimiento y Soporte**
  - Duración: 4 semanas

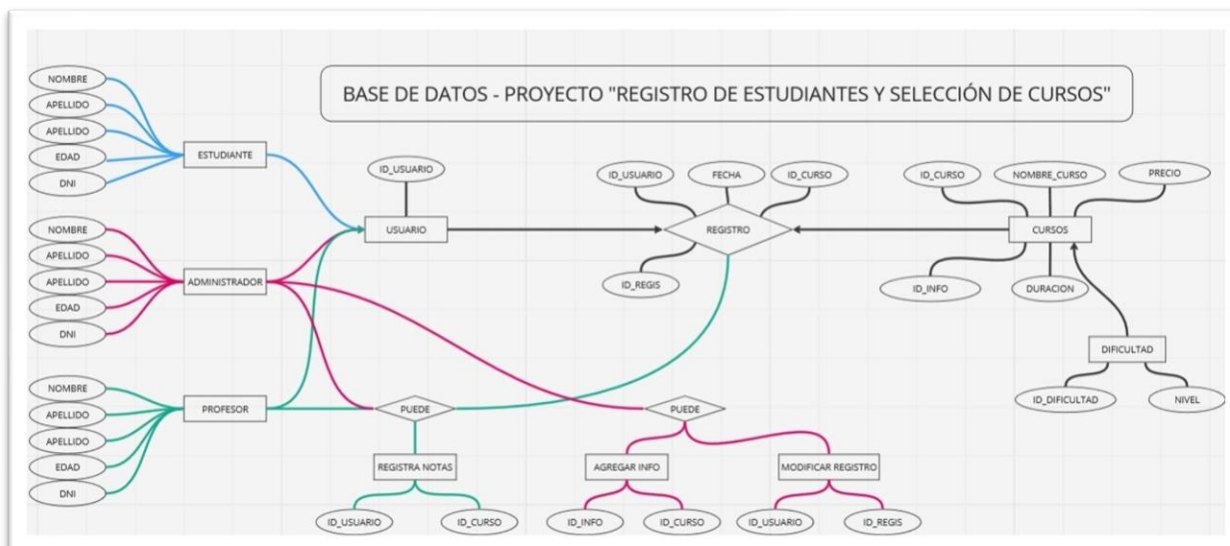
### 4.3. Modelado del Negocio

El modelado del negocio incluirá:

- **Análisis de Procesos:** Identificación y documentación de los procesos académicos y administrativos del Instituto CIMAS.
- **Modelado de Datos:** Diseño de un modelo de datos que represente la estructura de la información educativa y administrativa del instituto.
- **Modelado de Casos de Uso:** Descripción de las interacciones de los usuarios con el sistema propuesto, identificando los casos de uso más relevantes.

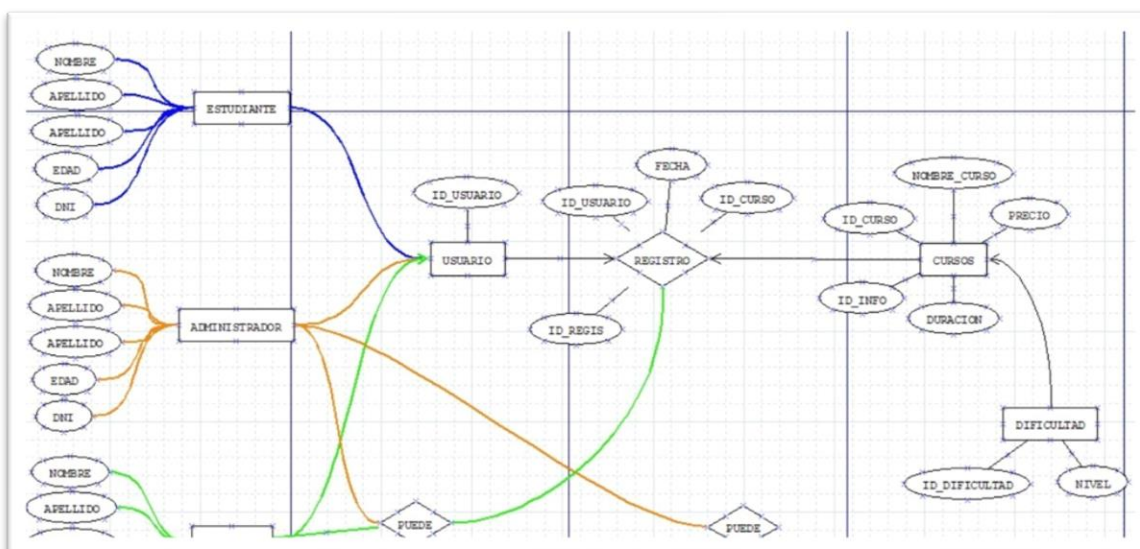
## 5. CAPÍTULO V: METODOLOGÍA DE DESARROLLO

Imagen N°01. MODELO CONCEPTUAL



Fuente: PROPIA

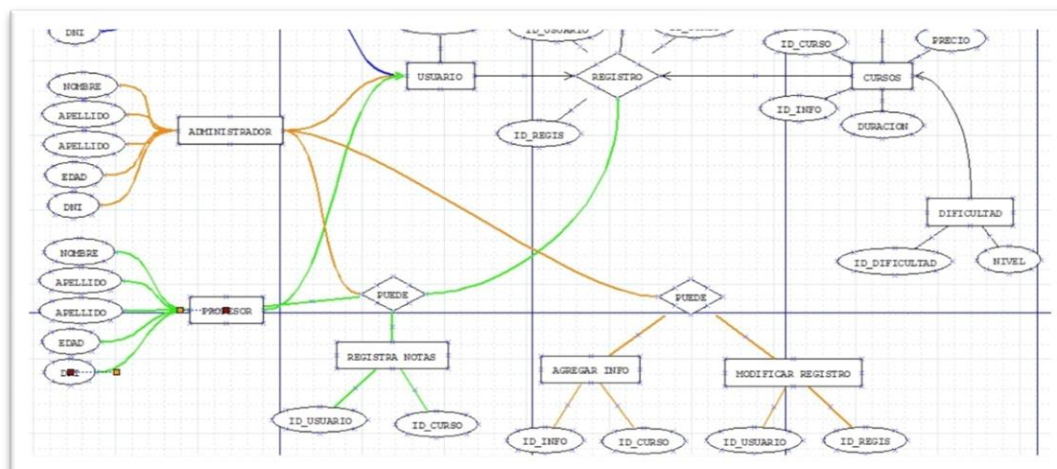
Imagen N°02. MODELO CONCEPTUAL



Fuente: PROPIA



Imagen N°03. **MODELO CONCEPTUAL**



Fuente: PROPIA

Cuadro N°01. **NORMALIZACION**

| 0FN   | 1FN  | 2FN   |
|---|--|---|
| <b>USUARIOS</b><br>codigoAlumno<br>nombreAlumno<br>apellido1Alumno<br>apellido2Alumno<br>emailAlumno<br>edadAlumno<br>dniAlumno<br>codigoProfesor<br>nombreProfesor<br>apellido1Profesor<br>apellido2Profesor<br>emailProfesor<br>dniProfesor<br>codigoAdmin<br>nombreAdmin<br>apellido1Admin<br>apellido2Admin<br>emailAdmin<br>dniAdmin | <b>ALUMNOS</b><br>PK codigoAlumno<br>nombreAlumno<br>apellido1Alumno<br>apellido2Alumno<br>emailAlumno<br>edadAlumno<br>dniAlumno<br><br><b>PROFESORES</b><br>PK codigoProfesor<br>nombreProfesor<br>apellido1Profesor<br>apellido2Profesor<br>emailProfesor<br>dniProfesor<br><br><b>ADMINISTRADOR</b><br>PK codigoAdmin<br>nombreAdmin<br>apellido1Admin<br>apellido2Admin<br>emailAdmin<br>dniAdmin | <b>USUARIO</b><br>PK ID_Usuario<br>Email<br>Password<br>Nombre<br>Apellido1<br>Apellido2<br>Edad<br>Dni<br>ID_Rol FK<br><br><b>ROL</b><br>PK ID_ROL<br>Nombre<br><br><b>ESTUDIANTE</b><br>PK ID_Estudiante<br>ID_Usuario FK<br><br><b>Profesor</b><br>PK ID_Profesor<br>ID_Usuario FK<br><br><b>Administrador</b><br>PK ID_Administrador<br>ID_Usuario FK |

Fuente:

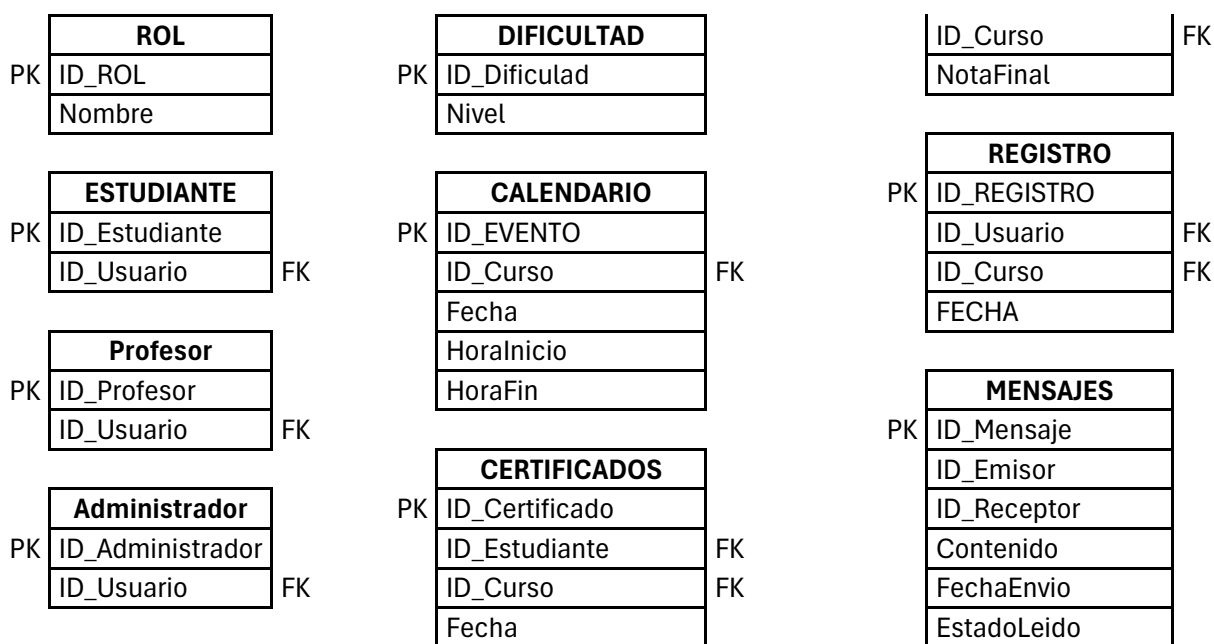


**Cuadro N°02. NORMALIZACION**

| 0FN   |  | 1FN  |  | 2FN   |  |
|---|--|--|--|---|--|
| <b>USUARIOS</b><br>codigoAlumno<br>nombreAlumno<br>apellido1Alumno<br>apellido2Alumno<br>emailAlumno<br>edadAlumno<br>dniAlumno<br>codigoProfesor<br>nombreProfesor<br>apellido1Profesor<br>apellido2Profesor<br>emailProfesor<br>dniProfesor<br>codigoAdmin<br>nombreAdmin<br>apellido1Admin<br>apellido2Admin<br>emailAdmin<br>dniAdmin |  | <b>ALUMNOS</b><br>PK codigoAlumno<br>nombreAlumno<br>apellido1Alumno<br>apellido2Alumno<br>emailAlumno<br>edadAlumno<br>dniAlumno<br><br><b>PROFESORES</b><br>PK codigoProfesor<br>nombreProfesor<br>apellido1Profesor<br>apellido2Profesor<br>emailProfesor<br>dniProfesor<br><br><b>ADMINISTRADOR</b><br>PK codigoAdmin<br>nombreAdmin<br>apellido1Admin<br>apellido2Admin<br>emailAdmin<br>dniAdmin |  | <b>USUARIO</b><br>PK ID_Usuario<br>Email<br>Password<br>Nombre<br>Apellido1<br>Apellido2<br>Edad<br>Dni<br>ID_Rol FK<br><br><b>ROL</b><br>PK ID_ROL<br>Nombre<br><br><b>ESTUDIANTE</b><br>PK ID_Estudiante<br>ID_Usuario FK<br><br><b>Profesor</b><br>PK ID_Profesor<br>ID_Usuario FK<br><br><b>Administrador</b><br>PK ID_Administrador<br>ID_Usuario FK |  |

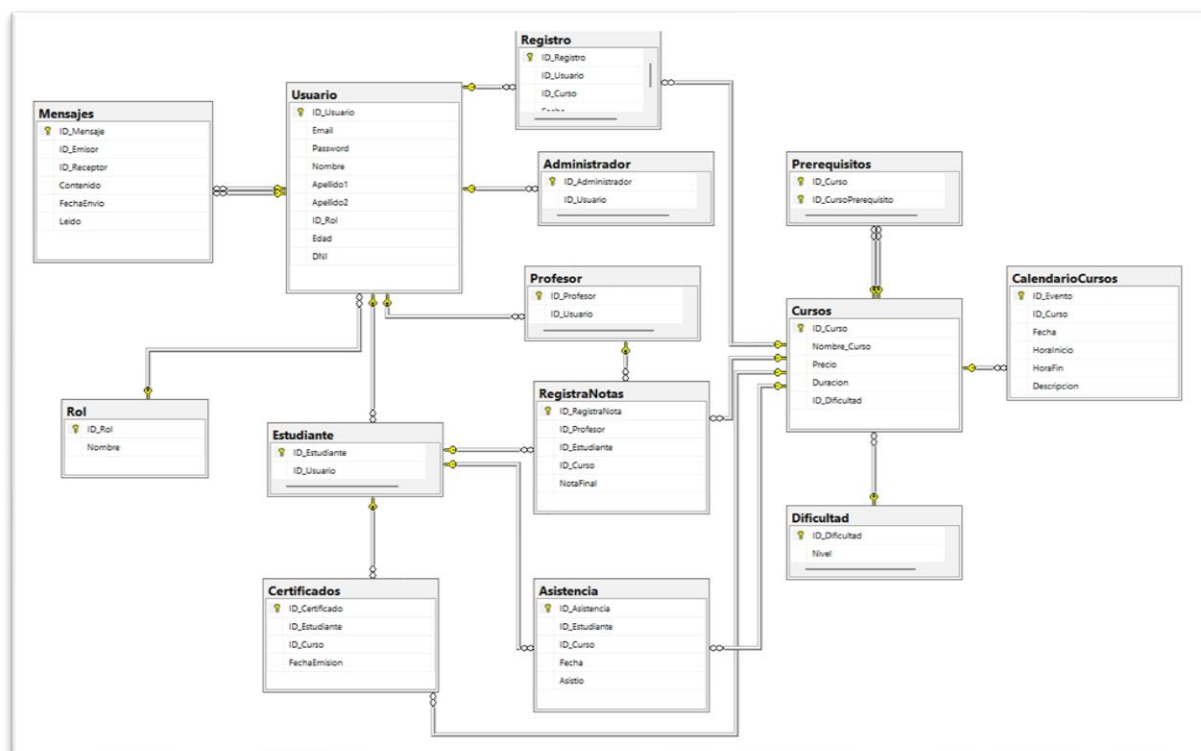
**Fuente: PROPIA**
**Cuadro N°03. NORMALIZACIÓN**

|  |                      |  |                        |   |                   |
|--|----------------------|--|------------------------|---|-------------------|
| PK ID_Usuario<br>Email<br>Password<br>Nombre<br>Apellido1<br>Apellido2<br>Edad<br>Dni<br>ID_Rol FK | <b>USUARIO</b>       | PK ID_Curso<br>nombreCurso<br>Precio<br>Duracion<br>ID_Dificultad FK | <b>CURSOS</b>          | PK ID_Asistencia<br>ID_Estudiante FK<br>ID_Curso FK<br>Fecha<br>EstadoAsistio | <b>ASISTENCIA</b> |
| PK ID_Curso<br>ID_CursoPrerequisito PK   | <b>PREREQUISITOS</b> | PK ID_RegistrarNota<br>ID_Profesor FK<br>ID_Estudiante FK            | <b>REGISTRAR NOTAS</b> |   |                   |



Fuente: PROPIA

Imagen N°04. MODELO LOGICO



Fuente: PROPIA

## MODELO FISICO

```
CREATE DATABASE ProyectoFinal;
```

```
USE ProyectoFinal;
```

```
-- Tabla Rol
```

```
CREATE TABLE Rol (
    ID_Rol INT NOT NULL,
    Nombre VARCHAR(20) NOT NULL
);
```

```
-- Tabla Usuario
```

```
CREATE TABLE Usuario (
    ID_Usuario INT NOT NULL,
    Email VARCHAR(100) UNIQUE,
    Password VARCHAR(255),
    Nombre VARCHAR(50),
    Apellido1 VARCHAR(50),
    Apellido2 VARCHAR(50),
    ID_Rol INT DEFAULT 1, -- SE DEBE CREAR LOS ROLES.
    Edad INT NULL,
    DNI VARCHAR(20) NULL
);
```

```
-- Tabla Estudiante
```

```
CREATE TABLE Estudiante (
    ID_Estudiante INT NOT NULL,
    ID_Usuario INT NOT NULL
);
```

```
-- Tabla Administrador
```

```
CREATE TABLE Administrador (
    ID_Administrador INT NOT NULL,
    ID_Usuario INT NOT NULL
);
```

```
-- Tabla Profesor
```

```
CREATE TABLE Profesor (
    ID_Profesor INT NOT NULL,
    ID_Usuario INT NOT NULL
);
```

```
-- Tabla Dificultad
```

```
CREATE TABLE Dificultad (
    ID_Dificultad INT NOT NULL,
    Nivel VARCHAR(50)
);
```

```
-- Tabla Cursos
```

```
CREATE TABLE Cursos (
    ID_Curso INT NOT NULL,
    Nombre_Curso VARCHAR(100),
    Precio DECIMAL(10, 2),
    Duracion VARCHAR(50),

```

```

        ID_Dificultad INT
    );

-- Tabla Registro
CREATE TABLE Registro (
    ID_Registro INT NOT NULL,
    ID_Usuario INT,
    ID_Curso INT,
    Fecha DATE
);

-- Tabla RegistraNotas
CREATE TABLE RegistraNotas (
    ID_RegistraNota INT NOT NULL,
    ID_Profesor INT,
    ID_Estudiante INT,
    ID_Curso INT,
    NotaFinal DECIMAL(5, 2)
);

-- Tabla Prerequisitos
CREATE TABLE Prerequisitos (
    ID_Curso INT NOT NULL,
    ID_CursoPrerequisito INT NOT NULL
);

-- Tabla Certificados
CREATE TABLE Certificados (
    ID_Certificado INT NOT NULL,
    ID_Estudiante INT,
    ID_Curso INT,
    FechaEmision DATE
);

-- Tabla Asistencia
CREATE TABLE Asistencia (
    ID_Asistencia INT NOT NULL,
    ID_Estudiante INT,
    ID_Curso INT,
    Fecha DATE,
    Asistio BIT
);

-- Tabla Mensajes
CREATE TABLE Mensajes (
    ID_Mensaje INT NOT NULL,
    ID_Emisor INT,
    ID_Receptor INT,
    Contenido TEXT,
    FechaEnvio DATETIME,
    Leido BIT DEFAULT 0
);

-- Tabla CalendarioCursos
CREATE TABLE CalendarioCursos (
    ID_Evento INT NOT NULL,

```

```

    ID_Curso INT,
    Fecha DATE,
    HoraInicio TIME,
    HoraFin TIME,
    Descripcion TEXT
);

-- #####--
#####

-- Llaves Rol
ALTER TABLE Rol ADD PRIMARY KEY (ID_Rol);

-- Llaves Usuario
ALTER TABLE Usuario ADD PRIMARY KEY (ID_Usuario);

ALTER TABLE Usuario
ADD CONSTRAINT FK_Rol FOREIGN KEY (ID_Rol) REFERENCES Rol(ID_Rol);

-- Llaves Estudiante
ALTER TABLE Estudiante ADD PRIMARY KEY (ID_Estudiante);

ALTER TABLE Estudiante
ADD CONSTRAINT FK_Estudiante_Usuario FOREIGN KEY (ID_Usuario) REFERENCES
Usuario(ID_Usuario);

-- Llaves Administrador
ALTER TABLE Administrador ADD PRIMARY KEY (ID_Administrador);

ALTER TABLE Administrador
ADD CONSTRAINT FK_Administrador_Usuario FOREIGN KEY (ID_Usuario)
REFERENCES Usuario(ID_Usuario);

-- Llaves Profesor
ALTER TABLE Profesor ADD PRIMARY KEY (ID_Profesor);

ALTER TABLE Profesor
ADD CONSTRAINT FK_Profesor_Usuario FOREIGN KEY (ID_Usuario) REFERENCES
Usuario(ID_Usuario);

-- Llave Dificultad
ALTER TABLE Dificultad ADD PRIMARY KEY (ID_Dificultad);

-- Llaves Cursos
ALTER TABLE Cursos ADD PRIMARY KEY (ID_Curso);

ALTER TABLE Cursos
ADD CONSTRAINT FK_Cursos_Dificultad FOREIGN KEY (ID_Dificultad)
REFERENCES Dificultad(ID_Dificultad);

-- Llaves Registro
ALTER TABLE Registro ADD PRIMARY KEY (ID_Registro);

```

```

ALTER TABLE Registro
ADD CONSTRAINT FK_Registro_Usuario FOREIGN KEY (ID_Usuario) REFERENCES
Usuario(ID_Usuario);

ALTER TABLE Registro
ADD CONSTRAINT FK_Registro_Cursos FOREIGN KEY (ID_Curso) REFERENCES
Cursos(ID_Curso);

-- Llaves RegistraNotas
ALTER TABLE RegistraNotas ADD PRIMARY KEY (ID_RegistraNota);

ALTER TABLE RegistraNotas
ADD CONSTRAINT FK_RegistraNotas_Profesor FOREIGN KEY (ID_Profesor)
REFERENCES Profesor(ID_Profesor);

ALTER TABLE RegistraNotas
ADD CONSTRAINT FK_RegistraNotas_Estudiante FOREIGN KEY (ID_Estudiante)
REFERENCES Estudiante(ID_Estudiante);

ALTER TABLE RegistraNotas
ADD CONSTRAINT FK_RegistraNotas_Curso FOREIGN KEY (ID_Curso) REFERENCES
Cursos(ID_Curso);

-- Llaves Prerequisitos

ALTER TABLE Prerequisitos ADD PRIMARY KEY (ID_Curso,
ID_CursoPrerequisito);

ALTER TABLE Prerequisitos
ADD CONSTRAINT FK_Prerequisitos_Curso FOREIGN KEY (ID_Curso) REFERENCES
Cursos(ID_Curso);

ALTER TABLE Prerequisitos
ADD CONSTRAINT FK_Prerequisitos_CursoPrerequisito FOREIGN KEY
(ID_CursoPrerequisito) REFERENCES Cursos(ID_Curso);

-- Llaves Certificados
ALTER TABLE Certificados ADD PRIMARY KEY (ID_Certificado);

ALTER TABLE Certificados
ADD CONSTRAINT FK_Certificados_Estudiante FOREIGN KEY (ID_Estudiante)
REFERENCES Estudiante(ID_Estudiante);

ALTER TABLE Certificados
ADD CONSTRAINT FK_Certificados_Curso FOREIGN KEY (ID_Curso) REFERENCES
Cursos(ID_Curso);

-- Llaves Asistencia
ALTER TABLE Asistencia ADD PRIMARY KEY (ID_Asistencia);

ALTER TABLE Asistencia
ADD CONSTRAINT FK_Asistencia_Estudiante FOREIGN KEY (ID_Estudiante)
REFERENCES Estudiante(ID_Estudiante);

ALTER TABLE Asistencia

```

```
ADD CONSTRAINT FK_Asiistencia_Curso FOREIGN KEY (ID_Curso) REFERENCES
Cursos(ID_Curso);

-- Llaves Mensajes
ALTER TABLE Mensajes ADD PRIMARY KEY (ID_Mensaje);

ALTER TABLE Mensajes
ADD CONSTRAINT FK_Mensajes_Emisor FOREIGN KEY (ID_Emisor) REFERENCES
Usuario(ID_Usuario);

ALTER TABLE Mensajes
ADD CONSTRAINT FK_Mensajes_Receptor FOREIGN KEY (ID_Receptor) REFERENCES
Usuario(ID_Usuario);

-- Llaves CalendarioCursos
ALTER TABLE CalendarioCursos ADD PRIMARY KEY (ID_Evento);

ALTER TABLE CalendarioCursos
ADD CONSTRAINT FK_CalendarioCursos_Curso FOREIGN KEY (ID_Curso)
REFERENCES Cursos(ID_Curso);
```

## MIGRACION AL GESTOR DE BASE DE DATOS

Se establecerá un plan detallado para la migración de las bases de datos actuales al gestor de bases de datos SQL Server, asegurando la integridad, disponibilidad y rendimiento óptimo de los datos durante y después del proceso de migración.

Este plan aplica a todas las bases de datos que serán migradas al SQL Server de la organización. Incluye la preparación, ejecución y verificación de la migración.

Teniendo como objetivos:

- Garantizar una transición suave y eficiente al SQL Server.
- Minimizar el tiempo de inactividad y la interrupción de los servicios.
- Asegurar la integridad y consistencia de los datos durante el proceso de migración.
- Optimizar el rendimiento de las bases de datos en el entorno de SQL Server.

Las fases para la migración al gestor de base de datos a realizar serán:

**Cuadro N°04. Fases para la migración al gestor de base de datos**

| Preparación para la Migración   |  |
|---|--|
| <ul style="list-style-type: none"> <li>• <b>Evaluación de la Base de Datos Actual:</b> <ul style="list-style-type: none"> <li>○ Identificar todas las bases de datos y sus dependencias.</li> <li>○ Evaluar el tamaño, estructura y contenido de cada base de datos.</li> <li>○ Revisar los esquemas, procedimientos almacenados, funciones y triggers.</li> </ul> </li> <li>• <b>Planificación de la Migración:</b> <ul style="list-style-type: none"> <li>○ Establecer un cronograma detallado con todas las fases de la migración.</li> <li>○ Designar un equipo de migración, asignando roles y responsabilidades.</li> <li>○ Definir las ventanas de tiempo para la migración para minimizar la interrupción del servicio.</li> </ul> </li> <li>• <b>Infraestructura del SQL Server:</b> <ul style="list-style-type: none"> <li>○ Configurar el entorno del SQL Server, asegurando que cumple con los requisitos de hardware y software.</li> <li>○ Hay que asegurar que las configuraciones de seguridad y red estén correctamente establecidas.</li> </ul> </li> </ul> |  |
|   |  |
| Ejecución de la Migración   |  |
| <ul style="list-style-type: none"> <li>• <b>Respaldo de Datos:</b> <ul style="list-style-type: none"> <li>○ Realizar una copia de seguridad completo de las bases de datos actuales antes de iniciar la migración.</li> <li>○ Verificar la integridad de los backups.</li> </ul> </li> <li>• <b>Migración de Esquema:</b> <ul style="list-style-type: none"> <li>○ Exportar los esquemas de las bases de datos actuales.</li> <li>○ Importar los esquemas en el SQL Server utilizando herramientas como SQL Server Management Studio (SSMS).</li> </ul> </li> <li>• <b>Migración de Datos:</b></li> </ul>   |  |
|   |  |



- Utilizar herramientas de migración de datos como SQL Server Integration Services (SSIS), el Asistente de Importación y Exportación de SQL Server, o scripts personalizados para transferir los datos.

- Validar la integridad y consistencia de los datos migrados.

- **Migración de Procedimientos Almacenados, Funciones y Triggers:**

- Exportar todos los procedimientos almacenados, funciones y triggers de la base de datos actual.
- Importar y probar cada uno en el SQL Server para asegurar que funcionan correctamente.

### Verificación y Validación

- **Pruebas de Integridad de Datos:**

- Realizar comparaciones entre los datos originales y los datos migrados para asegurar la integridad y consistencia.
- Verificar que no haya pérdida de datos ni corrupción.

- **Pruebas de Funcionamiento:**

- Ejecutar pruebas de funcionamiento para asegurar que todas las aplicaciones y servicios dependientes funcionan correctamente con la nueva base de datos.
- Probar todos los procedimientos almacenados, funciones y triggers para asegurar su correcto funcionamiento.

- **Pruebas de Rendimiento:**

- Realizar pruebas de rendimiento para asegurar que la base de datos en SQL Server cumple con los requisitos de rendimiento.
- Ajustar las configuraciones de SQL Server y optimizar las consultas según sea necesario.

### Implementación en Producción

- **Preparativos Finales:**

- Informar a todos los usuarios y partes interesadas sobre la fecha y hora de la migración final.
- Realizar un backup completo final de las bases de datos antes de la migración en producción.
- **Migración Final:**
  - Ejecutar la migración siguiendo el plan detallado.
  - Monitorear el proceso de migración en tiempo real para identificar y solucionar cualquier problema que surja.
- **Post-Migración:**
  - Realizar una verificación final de los datos y el funcionamiento.
  - Monitorear el rendimiento y solucionar cualquier problema que surja durante el período de estabilización.

### Documentación y Formación

- **Documentación del Proceso:**
  - Documentar todos los pasos y procedimientos realizados durante la migración.
  - Registrar cualquier problema encontrado y las soluciones aplicadas.
- **Formación:**
  - Proporcionar capacitación al personal de TI y a los usuarios clave sobre el uso y administración del nuevo entorno SQL Server.
  - Actualizar los manuales y guías de usuario según sea necesario.

### Mantenimiento y Soporte Continuo

- **Monitoreo Regular:**
  - Implementar un sistema de monitoreo para supervisar el rendimiento y la salud del SQL Server.
  - Realizar mantenimientos preventivos y ajustes de rendimiento regularmente.
- **Soporte Técnico:**

- Establecer un canal de soporte técnico para resolver cualquier problema relacionado con la base de datos.
- Programar revisiones periódicas para asegurar el correcto funcionamiento del entorno SQL Server.

## POLITICAS DE SEGURIDAD DE USUARIO

Las políticas de seguridad de usuario son pautas y prácticas diseñadas para proteger la integridad, confidencialidad y disponibilidad de los datos dentro de tu sistema. Aquí te proporciono una estructura básica de cómo podrías abordar este tema:

### 1. Seguridad de Datos:

- Requerir contraseñas fuertes que incluyan una combinación de letras, números y caracteres especiales.
- Cumplir con la protección de datos personales.

### 2. Autenticación Multifactor:

- Implementar la autenticación multifactor para aumentar la seguridad durante el inicio de sesión.

### 3. Control de Acceso:

- Definir roles y privilegios de usuario basados en los permisos de funciones.

### 4. Auditoría y Monitoreo:

- Establecer registros de auditoría para supervisar el acceso y las actividades de los usuarios.

### 5. Educación:

- Capacitar a los usuarios sobre las mejores prácticas de seguridad informática y la importancia de proteger la información confidencial.

### 6. Plan de contingencia:

- Establecer un plan de respuesta a incidentes para manejar rápidamente cualquier brecha de seguridad o violación de datos.

Cuadro N°05. Usuarios y roles de base de datos

| Administrador de la Base de Datos |
|-----------------------------------|
|-----------------------------------|

- **Nombre:** user\_admin
- **Rol:** sysadmin
- **Responsabilidades:** Gestión completa de la base de datos, incluyendo configuración, mantenimiento y solución de problemas.
- **Acceso:** Control total sobre todas las bases de datos y servidores SQL Server.

#### Usuario de Mantenimiento de Datos

- **Nombre:** MMTO\_TI
- **Roles:** db\_datawriter, db\_datareader, db\_ddladmin
- **Responsabilidades:** Realización de operaciones de inserción, modificación, eliminación y consultas de datos.
- **Acceso:** Permisos específicos para gestionar datos y estructuras de las bases de datos necesarias.

#### Usuario de Solo Consultas

- **Nombre:** user\_Gen
- **Rol:** db\_datareader
- **Responsabilidades:** Realización de consultas a las bases de datos.
- **Acceso:** Permisos de solo lectura para evitar modificaciones accidentales o no autorizadas.

#### Usuario de Backup

- **Nombre:** user\_TI
- **Rol:** db\_backupoperator
- **Responsabilidades:** Realización de copias de seguridad y restauración de bases de datos.
- **Acceso:** Permisos específicos para ejecutar tareas de backup y restore sin acceso adicional innecesario.

Se comparte la lista de políticas de seguridad de usuarios del sistema de gestión de base de datos MS SQL SERVER 2022

Cuadro N°06. Usuarios y roles de base de datos

| Acceso y Autenticación  |
|---|
| <ul style="list-style-type: none"> <li>• <b>Credenciales de Acceso:</b> Todos los usuarios deben tener credenciales únicas. Está prohibido compartir cuentas de usuario.</li> <li>• <b>Contraseñas:</b> Las contraseñas deben tener al menos 12 caracteres y contener una combinación de letras mayúsculas, minúsculas, números y caracteres especiales. Deben cambiarse cada 90 días.</li> </ul>   |
| Uso Aceptable   |
| <ul style="list-style-type: none"> <li>• <b>Acceso Autorizado:</b> Los usuarios solo deben acceder a la base de datos SQL Server para actividades relacionadas con sus funciones laborales.</li> <li>• <b>Prohibición de Software No Autorizado:</b> Está prohibido instalar o ejecutar software no autorizado en el entorno de la base de datos.</li> </ul>  |
| Protección de Datos   |
| <ul style="list-style-type: none"> <li>• <b>Encriptación:</b> Todos los datos sensibles deben estar encriptados tanto en reposo (usando Transparent Data Encryption - TDE) como en tránsito (usando SSL/TLS).</li> <li>• <b>Respaldo de Datos:</b> Se realizarán copias de seguridad diarias y se almacenarán en una ubicación segura diferente de la base de datos principal.</li> <li>• <b>Cifrado de Backups:</b> Las copias de seguridad deben estar cifradas para proteger los datos en caso de pérdida o robo.</li> </ul> |
| Monitoreo y Auditoría   |
| <ul style="list-style-type: none"> <li>• <b>Registro de Actividades:</b> Todas las actividades en SQL Server deben ser registradas en un log de auditoría.</li> <li>• <b>Auditoría de SQL Server:</b> Utilizar las capacidades de auditoría nativas de SQL Server para rastrear eventos importantes.</li> <li>• <b>Revisión de Logs:</b> Los logs de auditoría serán revisados semanalmente para detectar cualquier actividad sospechosa.</li> </ul>  |

### Gestión de Incidentes

- **Respuesta a Incidentes:** Cualquier sospecha de incidente de seguridad debe ser reportada inmediatamente al equipo de seguridad de la información.
- **Plan de Recuperación:** Se mantendrá un plan de recuperación ante desastres actualizado y probado anualmente.
- **Procedimientos de Contención:** Incluir procedimientos específicos para la contención rápida de amenazas detectadas en SQL Server.

### Formación y Concienciación

- **Capacitación:** Todos los usuarios con acceso a SQL Server deben completar una capacitación anual en seguridad de la información.
- **Concienciación Continua:** Se llevarán a cabo campañas de concienciación sobre seguridad periódicamente.

### Cumplimiento y Revisión

- **Cumplimiento:** El incumplimiento de estas políticas puede resultar en medidas disciplinarias, incluyendo la terminación del empleo.
- **Revisión de Políticas:** Estas políticas serán revisadas y actualizadas al menos una vez al año o cuando sea necesario debido a cambios en el entorno de seguridad o regulaciones.

### Seguridad Física

- **Acceso Físico:** El acceso físico a los servidores que ejecutan SQL Server debe estar restringido a personal autorizado. Las salas de servidores deben contar con sistemas de seguridad adecuados, como cerraduras y cámaras de vigilancia.

## PLAN DE BACKUPS Y RECUPERACION DE FALLOS

### Importancia:

Implementar un plan de backups y recuperación es crucial para proteger nuestros datos críticos y mantener la operación continua del negocio. Esto asegura la integridad y disponibilidad de nuestros sistemas y datos frente a amenazas cibernéticas.

### Objetivos:

- Garantizar la integridad y disponibilidad de los datos.
- Minimizar el tiempo de inactividad y pérdida de datos en caso de fallo.
- Establecer procedimientos claros para la creación, verificación y restauración de copias de seguridad

### Detalles del Plan:

- **Identificación de Riesgos Potenciales:**
  - Identificamos activos críticos: datos, aplicaciones y sistemas.
- **Frecuencia de Backups:**
  - Determinamos la frecuencia basada en la necesidad e integridad de datos.
- **Almacenamiento de Backups:**
  - Utilizamos almacenamiento seguro
  - Consideramos opciones como almacenamiento en la nube, discos externos y servidores dedicados.
- **Procedimientos de Recuperación:**
  - Documentamos procedimientos paso a paso para restaurar datos en casos de fallos.
- **Pruebas y Validación:**
  - Realizamos pruebas regulares para asegurar la eficacia de los procedimientos.
- **Actualización y Mantenimiento:**
  - Mantenemos el plan actualizado conforme evolucionan sistemas y requisitos operativos en función a nuevas amenazas y tecnologías.

### Implementación de Firewall:

- **Tipo de Firewall:**
  - Consideramos firewalls de hardware, software, de próxima generación (NGFW) o en la nube según necesidades de seguridad.
- **Configuración y Mantenimiento:**
  - Configuramos reglas de firewall adecuadas y realizamos actualizaciones regulares para responder rápidamente a amenazas.
- **Integración con Otras Medidas de Seguridad:**
  - Integramos el firewall con antivirus, sistemas de detección y respuesta a incidentes (EDR) y políticas de seguridad robustas.

#### **Tipo de Backup a Realizar:**

Implementar un plan de backups y recuperación es crucial para proteger nuestros datos críticos y mantener la operación continua del negocio. Esto asegura la integridad y disponibilidad de nuestros sistemas y datos frente a amenazas cibernéticas.

- **Backup Completo:** Realiza una copia de seguridad completa de la base de datos. Frecuencia: semanal (domingos a las 3:00 AM).
- **Remoto:** Una copia de las copias de seguridad se almacenará en una ubicación remota segura, utilizando almacenamiento en la nube cifrado.

La creación de Backups se realizará de la siguiente manera:

- **Administrador de Backup:** El usuario user\_TI será responsable de la ejecución de las tareas de backup.
- **Scripts de Backup:** Utilizar scripts automatizados para la creación de backups, programados mediante el Agente SQL Server (SQL Server Agent).
- **Verificación de Backups:** Comprobar la integridad de las copias de seguridad después de cada creación utilizando la opción RESTORE VERIFYONLY.
- Configurar notificaciones por correo electrónico para informar al equipo de TI sobre el estado de los backups (éxito o fallo).

El procedimiento de recuperación de datos se realizará en las siguientes fases:

- **Evaluación de Fallos:**
  - Determinar la causa del fallo y evaluar el alcance del daño.



- Decidir el tipo de recuperación necesario (completa, diferencial o de registro de transacciones).
- **Recuperación de Datos:**
  - Backup Completo: Restaurar la copia de seguridad completa más reciente.
  - Backup Diferencial: Restaurar la copia de seguridad completa más reciente y luego aplicar la copia de seguridad diferencial más reciente.
  - Registro de Transacciones: Aplicar las copias de seguridad del registro de transacciones en secuencia hasta el punto de falla.
- **Pruebas de Recuperación:**
  - Realizar pruebas de recuperación periódicas (trimestrales) para asegurarse de que los procedimientos de recuperación funcionen correctamente y los datos se puedan restaurar sin problemas.

Responsabilidades en el plan de backup de base de datos

- **Administrador de la Base de Datos (user\_admin):** Responsable de la supervisión general del plan de backups y recuperación, así como de la ejecución de pruebas de recuperación.
- **Usuario de Backup (user\_TI):** Responsable TI (Supervisor TI – VICTOR QUISPE H.) de la ejecución diaria de las tareas de backup y verificación de la integridad de estos.
- **Equipo de TI:** Responsable de la infraestructura de almacenamiento de backups y soporte durante la recuperación de fallos.

Documentación y Registro del backup de base de datos

- **Registro de Backups:** Mantener un registro detallado de todas las operaciones de backup, incluyendo fechas, tipos de backup, ubicación de almacenamiento y resultados de la verificación.
- **Documentación de Procedimientos:** Mantener documentación actualizada de todos los procedimientos de backup y recuperación, incluyendo scripts utilizados y configuraciones del Agente SQL Server.

Revisión y Actualización de las políticas del backup de base de datos.

- **Revisión Periódica:** Revisar y actualizar el plan de backups y recuperación al menos una vez al año o cuando haya cambios significativos en la infraestructura de la base de datos.

- **Actualización de Procedimientos:** Ajustar los procedimientos según los resultados de las pruebas de recuperación y las auditorías internas.

Seguridad de Backups de base de datos.

- **Encriptación:** Todas las copias de seguridad deben estar cifradas utilizando una clave de cifrado fuerte para proteger los datos sensibles.
- **Acceso Restringido:** El acceso a las copias de seguridad debe estar restringido a personal autorizado y debe ser auditado regularmente.

Imagen N°05.

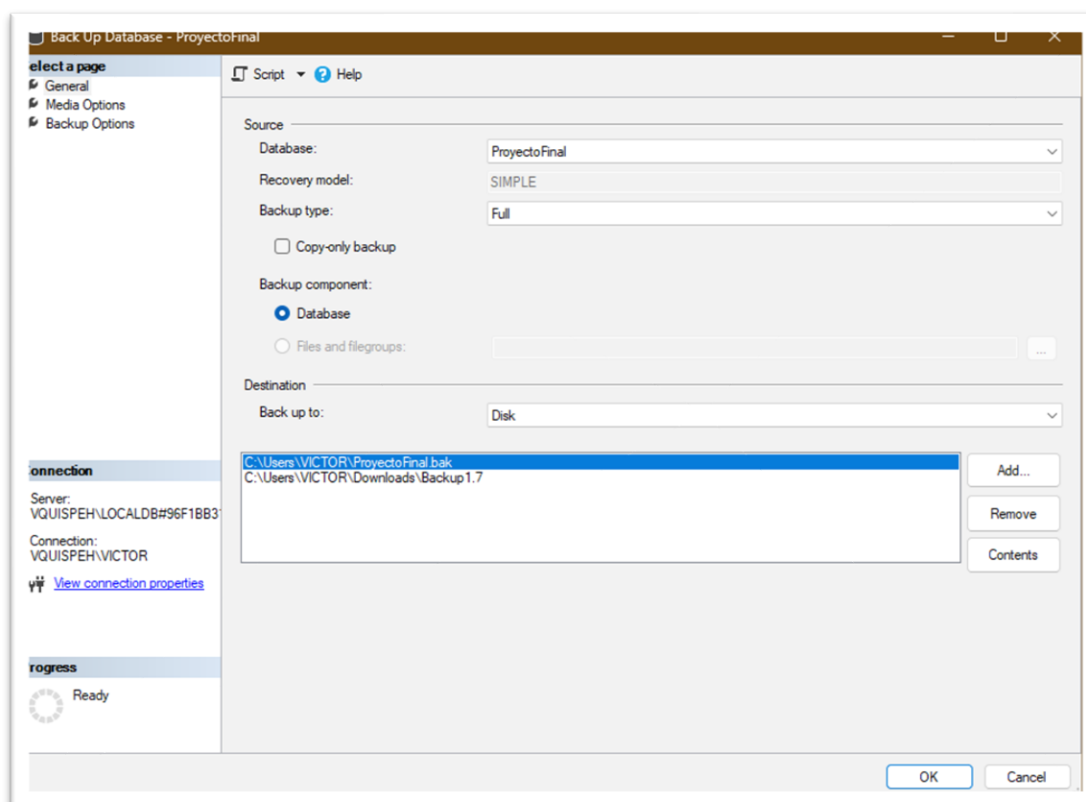


Imagen N°06. Fuente: SQL

En resumen, es fundamental tener políticas de seguridad y un plan de backups para proteger nuestros datos y asegurar que estén siempre disponibles y seguros, incluso en situaciones difíciles.

## 6. CAPÍTULO VI: CONCLUSIONES Y RECOMENDACIONES

### 6.1. CONCLUSIONES

Las conclusiones de un proyecto deben reflejar los logros y resultados obtenidos a lo largo del desarrollo del mismo. En el contexto del proyecto "Desarrollo de una Plataforma de Gestión Académica para el Instituto CIMAS", podrías considerar las siguientes conclusiones basadas en el contenido del documento:

1. **Mejora en la Gestión Académica:** La implementación de la plataforma centralizada de gestión académica ha demostrado ser una solución efectiva para reducir los errores en la inscripción de estudiantes, la asignación de cursos y el registro de calificaciones.
2. **Eficiencia Operativa:** La automatización de procesos administrativos ha permitido un uso más eficiente del tiempo y los recursos del personal administrativo, lo que ha resultado en una mayor eficiencia operativa.
3. **Acceso a Información:** La plataforma desarrollada ha facilitado el acceso rápido y preciso a la información relevante, beneficiando tanto a estudiantes como a docentes y personal administrativo en sus tareas diarias.
4. **Comunicación y Coordinación:** La mejora en la comunicación interna ha llevado a una mejor coordinación de las actividades académicas, asegurando una respuesta más efectiva a las necesidades y preocupaciones de los estudiantes.
5. **Satisfacción de los Usuarios:** La implementación de la plataforma ha aumentado la satisfacción de los usuarios finales (estudiantes, docentes y personal administrativo) gracias a la accesibilidad y la facilidad de uso del sistema.

### 6.2. RECOMENDACIONES

Las recomendaciones deben ofrecer sugerencias prácticas para la mejora continua y la sostenibilidad del sistema implementado. Aquí hay algunas recomendaciones basadas en el proyecto:

1. **Capacitación Continua:** Es fundamental continuar con la capacitación del personal administrativo y docente en el uso de la plataforma para asegurar que todos puedan aprovechar al máximo sus funcionalidades.
2. **Actualización y Mantenimiento:** Se recomienda establecer un plan regular de actualización y mantenimiento del sistema para asegurar que permanezca eficiente y seguro ante nuevas demandas y posibles amenazas de seguridad.
3. **Ampliación de Funcionalidades:** Considerar la incorporación de nuevas funcionalidades que puedan surgir de las necesidades identificadas por los usuarios, como módulos adicionales para la gestión de recursos educativos o la integración con herramientas de aprendizaje en línea.

4. **Monitoreo y Evaluación:** Implementar un sistema de monitoreo y evaluación constante para identificar áreas de mejora y asegurar que la plataforma continúe cumpliendo con los objetivos establecidos.
5. **Fomentar la Retroalimentación:** Crear canales efectivos para que los usuarios puedan proporcionar retroalimentación continua sobre el sistema, permitiendo así ajustar y mejorar la plataforma en función de sus necesidades y sugerencias.

| OBJETIVO GENERAL   | CONCLUSIONES   |
|--|--|
| Diseñar y desarrollar una plataforma de gestión académica accesible y de bajo costo para el Instituto CIMAS, que facilite la administración académica y mejore la experiencia educativa de alumnos y docentes. | Se espera que la plataforma incremente la eficiencia, reduzca errores humanos, y facilite el acceso a la información, contribuyendo a una mejor experiencia educativa. |
| OBJETIVOS ESPECIFICOS  |  |
| - Crear una base de datos eficiente para el manejo de información académica  | Diseñar y desarrollar una base de datos robusta que permita almacenar y gestionar toda la información académica, asegurando integridad y accesibilidad de los datos.   |
| - Implementar funcionalidades que permitan el registro y la administración de cursos de manera efectiva.   | Gestionar registros y llevar un control eficiente de las actividades académicas.   |
| - Reducir el uso de recursos físicos, promoviendo la sostenibilidad ambiental.   | Digitalizar o automatizar los procesos administrativos para minimizar el uso de papel y otros recursos físicos, contribuyendo a la reducción del impacto ambiental.    |
| - Asegurar la integridad y seguridad de los datos académicos.  | Implementar medidas de seguridad que protejan la integridad de los datos almacenados en la plataforma.   |
| - Proveer un sistema de notificaciones y comunicación entre alumnos y docentes.  | Integrar un sistema de comunicación interna que permita a alumnos y docentes.  |
| Estimación de costos   |  |
| <ul style="list-style-type: none"> <li>- Desarrollo de software: 500</li> <li>- Infraestructura de equipos: 500</li> <li>- Hosting o VPS: 500</li> <li>- Capacitación: 200</li> <li>- Soporte: 200</li> </ul>  | Total estimado:<br>\$ 2000   |
| Tiempo de desarrollo   |  |
| - Fase 1   | Análisis y Recolección de Requerimientos (2 semanas)   |
| - Fase 2   | Diseño de la Base de Datos y la Plataforma (4 semanas)   |
| - Fase 3   | Desarrollo e Implementación (8 semanas)  |
| - Fase 4   | Pruebas y Validación (3 semanas)   |
| - Fase 5   | Implementación y Capacitación (2 semanas)  |
| - Fase 6   | Mantenimiento y Soporte (4 semanas)  |
| <b>TOTAL TIEMPO ESTIMADO</b>   | Aproximadamente 6 meses  |

## 7. INDICES

Cuadro N°07.

### CREACION DE TABLAS

```

1  -- Tabla Rol
2  CREATE TABLE Rol (
3      ID_Rol INT NOT NULL,
4      Nombre VARCHAR(20) NOT NULL
5  );
6
7  -- Tabla Usuario
8  CREATE TABLE Usuario (
9      ID_Usuario INT NOT NULL,
10     Email VARCHAR(100) UNIQUE,
11     Password VARCHAR(255),
12     Nombre VARCHAR(50),
13     Apellido1 VARCHAR(50),
14     Apellido2 VARCHAR(50),
15     ID_Rol INT DEFAULT 1, -- SE DEBE CREAR LOS ROLES
16     Edad INT NULL,
17     DNI VARCHAR(20) NULL
18 );
19
20 -- Tabla Estudiante
21 CREATE TABLE Estudiante (
22     ID_Estudiante INT NOT NULL,
23     ID_Usuario INT NOT NULL
24 );
25
26 -- Tabla Administrador
27 CREATE TABLE Administrador (
28     ID_Administrador INT NOT NULL,
29     ID_Usuario INT NOT NULL
30 );
31
32 -- Tabla Profesor
33 CREATE TABLE Profesor (
34     ID_Profesor INT NOT NULL,
35     ID_Usuario INT NOT NULL
36 );
37
38 -- Tabla Dificultad
39 CREATE TABLE Dificultad (
40     ID_Dificultad INT NOT NULL,
41     Nivel VARCHAR(50)
42 );
43
44 -- Tabla Cursos
45 CREATE TABLE Cursos (
46     ID_Curso INT NOT NULL,
47     Nombre_Curso VARCHAR(100),
48     Precio DECIMAL(10, 2),
49     Duracion VARCHAR(50),
50     ID_Dificultad INT
51 );
52
53 -- Tabla Registro
54 CREATE TABLE Registro (
55     ID_Registro INT NOT NULL,
56     ID_Usuario INT,
57     ID_Curso INT,
58     Fecha DATE
59 );

```

```

24 -- Tabla RegistraNotas
25 CREATE TABLE RegistraNotas (
26     ID_RegistraNota INT NOT NULL,
27     ID_Profesor INT,
28     ID_Estudiante INT,
29     ID_Curso INT,
30     NotaFinal DECIMAL(5, 2)
31 );
32
33 -- Tabla Prerequisitos
34 CREATE TABLE Prerequisitos (
35     ID_Curso INT NOT NULL,
36     ID_CursoPrerequisito INT NOT NULL
37 );
38
39 -- Tabla Certificados
40 CREATE TABLE Certificados (
41     ID_Certificado INT NOT NULL,
42     ID_Estudiante INT,
43     ID_Curso INT,
44     FechaEmision DATE
45 );
46
47 -- Tabla Asistencia
48 CREATE TABLE Asistencia (
49     ID_Asistencia INT NOT NULL,
50     ID_Estudiante INT,
51     ID_Curso INT,
52     Fecha DATE,
53     Asistio BIT
54 );
55
56 -- Tabla Mensajes
57 CREATE TABLE Mensajes (
58     ID_Mensaje INT NOT NULL,
59     ID_Emisor INT,
60     ID_Receptor INT,
61     Contenido TEXT,
62     FechaEnvio DATETIME,
63     Leido BIT DEFAULT 0
64 );
65
66 -- Tabla CalendarioCursos
67 CREATE TABLE CalendarioCursos (
68     ID_Evento INT NOT NULL,
69     ID_Curso INT,
70     Fecha DATE,
71     HoraInicio TIME,
72     HoraFin TIME,
73     Descripcion TEXT
74 );

```

Fuente: PROPIA

Imagen N°07. IMPLEMENTACIÓN DE LLAVES PRIMARIAS Y FORANEAS

```

1  -- Llaves Rol
2  ALTER TABLE Rol ADD PRIMARY KEY (ID_Rol);
3
4  -- Llaves Usuario
5  ALTER TABLE Usuario ADD PRIMARY KEY (ID_Usuario);
6
7  ALTER TABLE Usuario
8  ADD CONSTRAINT FK_Rol FOREIGN KEY (ID_Rol) REFERENCES Rol(ID_Rol);
9
10
11 -- Llaves Estudiante
12 ALTER TABLE Estudiante ADD PRIMARY KEY (ID_Estudiante);
13
14 ALTER TABLE Estudiante
15 ADD CONSTRAINT FK_Estudiante_Usuario FOREIGN KEY (ID_Usuario) REFERENCES Usuario(ID_Usuario);
16
17
18 -- Llaves Administrador
19 ALTER TABLE Administrador ADD PRIMARY KEY (ID_Administrador);
20
21 ALTER TABLE Administrador
22 ADD CONSTRAINT FK_Administrador_Usuario FOREIGN KEY (ID_Usuario) REFERENCES Usuario(ID_Usuario);
23
24
25 -- Llaves Profesor
26 ALTER TABLE Profesor ADD PRIMARY KEY (ID_Profesor);
27
28 ALTER TABLE Profesor
29 ADD CONSTRAINT FK_Profesor_Usuario FOREIGN KEY (ID_Usuario) REFERENCES Usuario(ID_Usuario);
30
31 -- Llave Dificultad
32 ALTER TABLE Dificultad ADD PRIMARY KEY (ID_Dificultad);
33
34 -- Llaves Cursos
35 ALTER TABLE Cursos ADD PRIMARY KEY (ID_Curso);
36
37 ALTER TABLE Cursos
38 ADD CONSTRAINT FK_Cursos_Dificultad FOREIGN KEY (ID_Dificultad) REFERENCES Dificultad(ID_Dificultad);
39
40 -- Llaves Registro
41
42 ALTER TABLE Registro ADD PRIMARY KEY (ID_Registro);
43
44 ALTER TABLE Registro
45 ADD CONSTRAINT FK_Registro_Usuario FOREIGN KEY (ID_Usuario) REFERENCES Usuario(ID_Usuario);
46
47 ALTER TABLE Registro
48 ADD CONSTRAINT FK_Registro_Cursos FOREIGN KEY (ID_Curso) REFERENCES Cursos(ID_Curso);
49
50 -- Llaves RegistraNotas
51 ALTER TABLE RegistraNotas ADD PRIMARY KEY (ID_RegistraNota);
52
53 ALTER TABLE RegistraNotas
54 ADD CONSTRAINT FK_RegistraNotas_Profesor FOREIGN KEY (ID_Profesor) REFERENCES Profesor(ID_Profesor);
55
56 ALTER TABLE RegistraNotas
57 ADD CONSTRAINT FK_RegistraNotas_Estudiante FOREIGN KEY (ID_Estudiante) REFERENCES Estudiante(ID_Estudiante);
58
59 ALTER TABLE RegistraNotas
60 ADD CONSTRAINT FK_RegistraNotas_Curso FOREIGN KEY (ID_Curso) REFERENCES Cursos(ID_Curso);
61

```

Fuente: PROPIA



Imagen N°08. IMPLEMENTACIÓN DE LLAVES PRIMARIAS Y FORANEAS

```

62 -- Llaves Prerequisitos
63
64 ALTER TABLE Prerequisitos ADD PRIMARY KEY (ID_Curso, ID_CursoPrerequisito);
65
66 ALTER TABLE Prerequisitos
67 ADD CONSTRAINT FK_Prerequisitos_Curso FOREIGN KEY (ID_Curso) REFERENCES Cursos(ID_Curso);
68
69 ALTER TABLE Prerequisitos
70 ADD CONSTRAINT FK_Prerequisitos_CursoPrerequisito FOREIGN KEY (ID_CursoPrerequisito) REFERENCES Cursos(ID_Curso);
71
72 -- Llaves Certificados
73 ALTER TABLE Certificados ADD PRIMARY KEY (ID_Certificado);
74
75 ALTER TABLE Certificados
76 ADD CONSTRAINT FK_Certificados_Estudiante FOREIGN KEY (ID_Estudiante) REFERENCES Estudiante(ID_Estudiante);
77
78 ALTER TABLE Certificados
79 ADD CONSTRAINT FK_Certificados_Curso FOREIGN KEY (ID_Curso) REFERENCES Cursos(ID_Curso);
80
81 -- Llaves Asistencia
82 ALTER TABLE Asistencia ADD PRIMARY KEY (ID_Asistencia);
83
84 ALTER TABLE Asistencia
85 ADD CONSTRAINT FK_Asistencia_Estudiante FOREIGN KEY (ID_Estudiante) REFERENCES Estudiante(ID_Estudiante);
86
87 ALTER TABLE Asistencia
88 ADD CONSTRAINT FK_Asistencia_Curso FOREIGN KEY (ID_Curso) REFERENCES Cursos(ID_Curso);
89
90 -- Llaves Mensajes
91 ALTER TABLE Mensajes ADD PRIMARY KEY (ID_Mensaje);
92
93 ALTER TABLE Mensajes
94 ADD CONSTRAINT FK_Mensajes_Emisor FOREIGN KEY (ID_Emisor) REFERENCES Usuario(ID_Usuario);
95
96 ALTER TABLE Mensajes
97 ADD CONSTRAINT FK_Mensajes_Receptor FOREIGN KEY (ID_Receptor) REFERENCES Usuario(ID_Usuario);
98
99 -- Llaves CalendarioCursos
100 ALTER TABLE CalendarioCursos ADD PRIMARY KEY (ID_Evento);
101
102 ALTER TABLE CalendarioCursos
103 ADD CONSTRAINT FK_CalendarioCursos_Curso FOREIGN KEY (ID_Curso) REFERENCES Cursos(ID_Curso);

```

## Imagen N°09. CONSULTAS

```

1  --ROLES
2  INSERT INTO Rol (ID_Rol, Nombre) VALUES (1, 'Estudiante'),(2, 'Administrador'),(3, 'Profesor');
3
4  -- NIVELES DE DIFICULTAD
5  INSERT INTO Dificultad (ID_Dificultad, Nivel) VALUES (1, 'Básico'),(2, 'Intermedio'),(3, 'Avanzado');
6
7  -----
8  INSERT INTO Usuario (Email, Password, Nombre, Apellido1, Apellido2)
9  VALUES ('ejemplo1@correo.com', '123', 'victor', 'test', 'test');
10
11 SELECT * FROM Usuario WHERE Email = 'ejemplo1@correo.com';
12
13 INSERT INTO Cursos (ID_Curso, Nombre_Curso, Precio, Duracion, ID_Dificultad)
14 VALUES (010130, 'Curso de SQL Básico', 99.99, '2 meses', 1);
15
16 INSERT INTO Cursos (ID_Curso, Nombre_Curso, Precio, Duracion, ID_Dificultad)
17 VALUES (010131, 'Curso de SQL Avanzado', 149.99, '3 meses', 2);
18
19 INSERT INTO Cursos (ID_Curso, Nombre_Curso, Precio, Duracion, ID_Dificultad)
20 VALUES (010132, 'Curso de Programación en Python', 199.99, '4 meses', 3);
21
22
23 SELECT * FROM Dificultad;
24 SELECT * FROM Prerequisitos;
25 INSERT INTO Prerequisitos (ID_Curso, ID_CursoPrerequisito)
26 VALUES (010131, 010130);
27
28
29 --consulta
30 SELECT
31     c.ID_Curso AS ID_Curso_Curso,
32     c.Nombre_Curso AS Nombre_Curso_Curso,
33     p.ID_CursoPrerequisito AS ID_Curso_Requisito,
34     rc.Nombre_Curso AS Nombre_Curso_Requisito
35 FROM
36     Cursos c
37     LEFT JOIN
38     Prerequisitos p ON c.ID_Curso = p.ID_Curso
39     LEFT JOIN
40     Cursos rc ON p.ID_CursoPrerequisito = rc.ID_Curso;

```

Imagen N°010.

|   | Nombre_Alumno | Apellido1_Alumno | Apellido2_Alumno | Nombre_Curso        | NotaFinal | Nombre_Profesor |
|---|---------------|------------------|------------------|---------------------|-----------|-----------------|
| 1 | victor        | test             | test             | Curso de SQL Básico | 15.50     | cesar           |

Imagen N°011.

|   | ID_Profesor | Nombre_Profesor | Apellido_Profesor | Email               |
|---|-------------|-----------------|-------------------|---------------------|
| 1 | 1           | cesar           | aguirre           | ejemplo2@correo.com |

Imagen N°012.

|   | ID_Curso_Curso | Nombre_Curso_Curso              | ID_Curso_Requisito | Nombre_Curso_Requisito |
|---|----------------|---------------------------------|--------------------|------------------------|
| 1 | 10130          | Curso de SQL Básico             | <null>             | <null>                 |
| 2 | 10131          | Curso de SQL Avanzado           | 10130              | Curso de SQL Básico    |
| 3 | 10132          | Curso de Programación en Python | <null>             | <null>                 |