

Documentación de la Base de Datos para el Sistema de Gestión Académica de CEVAM

1. Introducción

Este documento detalla el diseño de la base de datos para el sistema de gestión académica de CEVAM. El propósito principal de esta aplicación es centralizar y optimizar la gestión académica y administrativa de estudiantes y profesores, enfocándose en el seguimiento del progreso en libros y unidades. La solución busca resolver la desorganización de la información y la ineficiencia en el seguimiento del avance académico, mejorando la eficiencia interna y la experiencia educativa general para estudiantes y profesores.

2. Objetivos del Sistema

- **Centralización de la Información:** Consolidar datos de estudiantes, profesores, libros y progreso en un único lugar.
- **Optimización del Seguimiento Académico:** Permitir a profesores marcar unidades completadas y a estudiantes visualizar su progreso.
- **Mejora de la Eficiencia Operativa:** Reducir la carga administrativa del personal y el tiempo dedicado a buscar y organizar información.
- **Personalización de la Experiencia Educativa:** Ofrecer un seguimiento más claro y personalizado a los estudiantes.
- **Escalabilidad:** Proporcionar una base sólida para manejar un creciente número de estudiantes y profesores en el futuro.

3. Diagrama Entidad-Relación (DER) / Diagrama de Base de Datos

erDiagram

ROLES {

INT id_rol PK

VARCHAR nombre_rol UNIQUE

}

USUARIOS {

INT id_usuario PK

VARCHAR nombre

VARCHAR apellido

VARCHAR correo UNIQUE

```
    VARCHAR contraseña_hash
    VARCHAR cedula_identidad UNIQUE
    VARCHAR numero_celular UNIQUE
    INT id_rol FK
    TIMESTAMP fecha_creacion
    TIMESTAMP fecha_actualizacion
}
```

```
LIBROS {
    INT id_libro PK
    VARCHAR nombre_libro
    VARCHAR nivel
    TEXT descripcion
    TIMESTAMP fecha_creacion
    TIMESTAMP fecha_actualizacion
}
```

```
UNIDADES {
    INT id_unidad PK
    INT id_libro FK
    VARCHAR nombre_unidad
    INT numero_unidad
    TEXT descripcion
}
```

```
TIPOS_CONTENIDO {
    INT id_tipo_contenido PK
    VARCHAR nombre_tipo UNIQUE
}
```

}

```
TIPOS_MATERIAL {  
    INT id_tipo_material PK  
    VARCHAR nombre_material UNIQUE  
}
```

```
CONTENIDOS_UNIDAD {  
    INT id_contenido_unidad PK  
    INT id_unidad FK  
    INT id_tipo_contenido FK  
    INT id_tipo_material FK  
    VARCHAR titulo  
    TEXT descripcion  
    VARCHAR url_recurso  
    TEXT contenido_texto  
    TIMESTAMP fecha_creacion  
    TIMESTAMP fecha_actualizacion  
}
```

```
ESTUDIANTES_LIBROS {  
    INT id_estudiante_libro PK  
    INT id_usuario_estudiante FK  
    INT id_libro FK  
    TIMESTAMP fecha_asignacion  
}
```

```
PROGRESO_UNIDADES {
```

```

    INT id_progreso PK
    INT id_usuario_estudiante FK
    INT id_unidad FK
    DATE fecha_completado
    TEXT comentarios_profesor
}

```

```

PROFESORES_ESTUDIANTES {
    INT id_profesor_estudiante PK
    INT id_usuario_profesor FK
    INT id_usuario_estudiante FK
    TIMESTAMP fecha_asignacion
}

```

```

ROLES ||--o{ USUARIOS : tiene
USUARIOS ||--o{ ESTUDIANTES_LIBROS : asignado_a
USUARIOS ||--o{ PROGRESO_UNIDADES : registra
USUARIOS ||--o{ PROFESORES_ESTUDIANTES : enseña_a
LIBROS ||--o{ UNIDADES : contiene
LIBROS ||--o{ ESTUDIANTES_LIBROS : asignado_a
UNIDADES ||--o{ PROGRESO_UNIDADES : tiene_progreso
UNIDADES ||--o{ CONTENIDOS_UNIDAD : tiene_contenido
TIPOS_CONTENIDO ||--o{ CONTENIDOS_UNIDAD : clasifica_contenido
TIPOS_MATERIAL ||--o{ CONTENIDOS_UNIDAD : clasifica_material
USUARIOS }o--|| PROFESORES_ESTUDIANTES : gestiona

```

4. Descripción Detallada de las Tablas y sus Relaciones

A continuación, se describe cada tabla, sus campos y las relaciones que establece con otras tablas en la base de datos:

4.1. Roles

- **Propósito:** Almacena los distintos tipos de usuarios que interactuarán con la aplicación, permitiendo una gestión de permisos y funcionalidades basada en roles.
- **Campos:**
 - id_rol (INT, PK, AUTO_INCREMENT): Identificador único del rol.
 - nombre_rol (VARCHAR(50), NOT NULL, UNIQUE): Nombre del rol (ej., 'Administrador', 'Profesor', 'Estudiante').
- **Relaciones:**
 - **Uno a Muchos** con Usuarios: Un rol puede ser asignado a múltiples usuarios.

4.2. Usuarios

- **Propósito:** Contiene la información fundamental de todos los usuarios del sistema (estudiantes, profesores y administradores).
- **Campos:**
 - id_usuario (INT, PK, AUTO_INCREMENT): Identificador único del usuario.
 - nombre (VARCHAR(100), NOT NULL): Nombre del usuario.
 - apellido (VARCHAR(100)): Apellido del usuario.
 - correo (VARCHAR(100), NOT NULL, UNIQUE): Correo electrónico del usuario, utilizado para el inicio de sesión y debe ser único.
 - contraseña_hash (VARCHAR(255), NOT NULL): Hash seguro de la contraseña del usuario.
 - cedula_identidad (VARCHAR(20), UNIQUE): Número de cédula de identidad del usuario, debe ser único.
 - numero_celular (VARCHAR(20), UNIQUE): Número de teléfono celular del usuario, debe ser único.
 - id_rol (INT, FK): Clave foránea que referencia a la tabla Roles, indicando el rol del usuario.
 - fecha_creacion (TIMESTAMP): Fecha y hora en que se creó el registro del usuario.
 - fecha_actualizacion (TIMESTAMP): Última fecha y hora en que se actualizó el registro del usuario.
- **Relaciones:**
 - **Muchos a Uno** con Roles: Cada usuario pertenece a un único rol.
 - **Uno a Muchos** con Estudiantes_Libros: Un usuario (estudiante) puede tener varios libros asignados.
 - **Uno a Muchos** con Progreso_Unidades: Un usuario (estudiante) registra progreso en varias unidades.

- **Uno a Muchos** con Profesores_Estudiantes: Un usuario puede ser profesor de varios estudiantes o un estudiante puede ser asignado a un profesor.

4.3. Libros

- **Propósito:** Almacena los detalles de los libros de inglés utilizados en CEVAM.
- **Campos:**
 - id_libro (INT, PK, AUTO_INCREMENT): Identificador único del libro.
 - nombre_libro (VARCHAR(255), NOT NULL): Título del libro.
 - nivel (VARCHAR(50), NOT NULL): Nivel del libro (ej., 'Adolescentes', 'Adultos').
 - descripcion (TEXT): Descripción opcional del libro.
 - fecha_creacion (TIMESTAMP): Fecha y hora de creación del registro del libro.
 - fecha_actualizacion (TIMESTAMP): Última fecha y hora de actualización del registro del libro.
- **Relaciones:**
 - **Uno a Muchos** con Unidades: Un libro contiene múltiples unidades.
 - **Uno a Muchos** con Estudiantes_Libros: Un libro puede ser asignado a múltiples estudiantes.

4.4. Unidades

- **Propósito:** Detalla cada una de las unidades que componen los libros.
- **Campos:**
 - id_unidad (INT, PK, AUTO_INCREMENT): Identificador único de la unidad.
 - id_libro (INT, FK): Clave foránea que referencia a la tabla Libros, indicando a qué libro pertenece la unidad.
 - nombre_unidad (VARCHAR(255), NOT NULL): Nombre o título de la unidad.
 - numero_unidad (INT, NOT NULL): Número secuencial de la unidad dentro de su libro.
 - descripcion (TEXT): Descripción opcional de la unidad.
 - **Restricción UNIQUE:** (id_libro, numero_unidad) asegura que no haya dos unidades con el mismo número dentro del mismo libro.
- **Relaciones:**
 - **Muchos a Uno** con Libros: Varias unidades pertenecen a un mismo libro.
 - **Uno a Muchos** con Progreso_Unidades: Una unidad puede tener el progreso registrado por múltiples estudiantes.
 - **Uno a Muchos** con Contenidos_Unidad: Una unidad puede tener múltiples recursos o contenidos asociados.

4.5. Tipos_Contenido

- **Propósito:** Cataloga las grandes categorías de contenido pedagógico, como se muestra en la estructura de contenidos proporcionada (Grammar, Listening, Speaking, Reading, Writing).
- **Campos:**
 - id_tipo_contenido (INT, PK, AUTO_INCREMENT): Identificador único del tipo de contenido.
 - nombre_tipo (VARCHAR(100), NOT NULL, UNIQUE): Nombre de la categoría de contenido (ej., 'Grammar').
- **Relaciones:**
 - **Uno a Muchos** con Contenidos_Unidad: Un tipo de contenido puede estar asociado a múltiples recursos de unidades.

4.6. Tipos_Material

- **Propósito:** Cataloga los tipos específicos de materiales educativos dentro de cada categoría de contenido (ej., Explanations, Videos, Exercises, Practice tests, Samples).
- **Campos:**
 - id_tipo_material (INT, PK, AUTO_INCREMENT): Identificador único del tipo de material.
 - nombre_material (VARCHAR(100), NOT NULL, UNIQUE): Nombre del tipo de material (ej., 'Explanations', 'Videos').
- **Relaciones:**
 - **Uno a Muchos** con Contenidos_Unidad: Un tipo de material puede ser utilizado en múltiples recursos de unidades.

4.7. Contenidos_Unidad

- **Propósito:** Relaciona las unidades con los recursos educativos específicos (videos, audios, textos, ejercicios) que las componen, permitiendo una gestión detallada de los materiales por unidad. Esta tabla responde a la necesidad de gestionar contenidos como "Recursos Favoritos" o que los profesores puedan "subir" o "editar" contenidos.
- **Campos:**
 - id_contenido_unidad (INT, PK, AUTO_INCREMENT): Identificador único del contenido de la unidad.
 - id_unidad (INT, FK): Clave foránea que referencia a la tabla Unidades, indicando a qué unidad pertenece este contenido.
 - id_tipo_contenido (INT, FK): Clave foránea que referencia a la tabla Tipos_Contenido, categorizando el contenido (ej., Grammar).
 - id_tipo_material (INT, FK): Clave foránea que referencia a la tabla Tipos_Material, especificando el tipo de material (ej., Videos).
 - titulo (VARCHAR(255), NOT NULL): Título descriptivo del recurso.
 - descripcion (TEXT): Descripción del recurso.

- url_recurso (VARCHAR(255)): URL si el recurso es un archivo externo (video, audio, PDF).
- contenido_texto (TEXT): El contenido del recurso si es texto directo (ej., una explicación escrita, un ejercicio de texto).
- fecha_creacion (TIMESTAMP): Fecha y hora de creación del recurso.
- fecha_actualizacion (TIMESTAMP): Última fecha y hora de actualización del recurso.
- **Restricción UNIQUE:** (id_unidad, id_tipo_contenido, id_tipo_material, titulo) asegura que un recurso específico no se duplique en una unidad.
- **Relaciones:**
 - **Muchos a Uno** con Unidades: Varios contenidos pueden pertenecer a una misma unidad.
 - **Muchos a Uno** con Tipos_Contenido: Un contenido es de un tipo de contenido específico.
 - **Muchos a Uno** con Tipos_Material: Un contenido es de un tipo de material específico.

4.8. Estudiantes_Libros

- **Propósito:** Registra qué libros están asignados a cada estudiante, permitiendo a los estudiantes visualizar los libros que están cursando y sus unidades.
- **Campos:**
 - id_estudiante_libro (INT, PK, AUTO_INCREMENT): Identificador único de la asignación.
 - id_usuario_estudiante (INT, FK): Clave foránea que referencia al id_usuario de la tabla Usuarios (solo para usuarios con rol de Estudiante).
 - id_libro (INT, FK): Clave foránea que referencia al id_libro de la tabla Libros.
 - fecha_asignacion (TIMESTAMP): Fecha en que se asignó el libro al estudiante.
 - **Restricción UNIQUE:** (id_usuario_estudiante, id_libro) impide que un estudiante tenga el mismo libro asignado dos veces.
- **Relaciones:**
 - **Muchos a Uno** con Usuarios (rol Estudiante): Un estudiante puede tener múltiples libros asignados.
 - **Muchos a Uno** con Libros: Un libro puede ser asignado a múltiples estudiantes.

4.9. Progreso_Unidades

- **Propósito:** Almacena el registro del progreso de cada estudiante en unidades específicas, indicando cuándo se completó una unidad y

permitiendo notas del profesor. Esta es una funcionalidad esencial para el seguimiento académico.

- **Campos:**
 - `id_progreso` (INT, PK, AUTO_INCREMENT): Identificador único del registro de progreso.
 - `id_usuario_estudiante` (INT, FK): Clave foránea que referencia al `id_usuario` de la tabla Usuarios (solo para usuarios con rol de Estudiante).
 - `id_unidad` (INT, FK): Clave foránea que referencia al `id_unidad` de la tabla Unidades.
 - `fecha_completado` (DATE, NOT NULL): Fecha en que la unidad fue marcada como completada.
 - `comentarios_profesor` (TEXT): Notas opcionales del profesor sobre el desempeño del estudiante en esa unidad.
 - **Restricción UNIQUE:** (`id_usuario_estudiante`, `id_unidad`) asegura que un estudiante solo pueda marcar una unidad como completada una vez.
- **Relaciones:**
 - **Muchos a Uno** con Usuarios (rol Estudiante): Un estudiante puede tener progreso en múltiples unidades.
 - **Muchos a Uno** con Unidades: Una unidad puede tener progreso registrado por múltiples estudiantes.

4.10. Profesores_Estudiantes

- **Propósito:** Relaciona a los profesores con los estudiantes que tienen a su cargo. Esto es fundamental para que un profesor pueda ver el progreso de "todos sus estudiantes en un dashboard".
- **Campos:**
 - `id_profesor_estudiante` (INT, PK, AUTO_INCREMENT): Identificador único de la asignación profesor-estudiante.
 - `id_usuario_profesor` (INT, FK): Clave foránea que referencia al `id_usuario` de la tabla Usuarios (solo para usuarios con rol de Profesor).
 - `id_usuario_estudiante` (INT, FK): Clave foránea que referencia al `id_usuario` de la tabla Usuarios (solo para usuarios con rol de Estudiante).
 - `fecha_asignacion` (TIMESTAMP): Fecha en que se estableció la relación profesor-estudiante.
 - **Restricción UNIQUE:** (`id_usuario_profesor`, `id_usuario_estudiante`) asegura que un profesor solo sea asignado una vez a un estudiante.
- **Relaciones:**
 - **Muchos a Uno** con Usuarios (rol Profesor): Un profesor puede gestionar a múltiples estudiantes.
 - **Muchos a Uno** con Usuarios (rol Estudiante): Un estudiante puede ser gestionado por un profesor.

5. Consideraciones Adicionales

- **Seguridad:** Se contempla el almacenamiento de contraseñas en texto plano, lo cual es una buena práctica de seguridad.
- **Escalabilidad:** El diseño modular con tablas separadas para roles, libros, unidades, tipos de contenido y materiales permite que la base de datos crezca y se adapte a futuras funcionalidades sin requerir cambios estructurales mayores.
- **Rendimiento:** Se han incluido índices (CREATE INDEX) en campos clave (id_rol, id_libro, id_usuario_estudiante, id_unidad, id_tipo_contenido, id_tipo_material) para optimizar el rendimiento de las consultas, especialmente aquellas que implican búsquedas y uniones frecuentes.
- **Flexibilidad de Contenidos:** La inclusión de las tablas Tipos_Contenido, Tipos_Material y Contenidos_Unidad permite una estructura flexible para categorizar y asociar diversos recursos educativos a las unidades, preparando la aplicación para futuras funcionalidades como "Recursos Favoritos" y la edición de contenidos por parte de los profesores.
- **Restricciones de Unicidad:** Se han aplicado restricciones UNIQUE a correo, cedula_identidad y numero_celular en la tabla Usuarios para garantizar que estos identificadores clave sean únicos para cada persona en el sistema.

Esta estructura de base de datos proporciona una base robusta y escalable para el sistema de gestión académica de CEVAM, cubriendo todas las funcionalidades prioritarias y sentando las bases para futuras expansiones.