



Proyecto de Sistemas de Bases de Datos I

Plataforma de gestión de academias interdisciplinarias en Unidad Educativa
Particular Daniel Comboni

Sistemas de Bases de Datos I
Segundo Término 2025-2026

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

Materia: Sistemas de Bases de Datos 1	Facultad de Ingeniería en Electricidad y Computación
Profesor: MSc. Irene M. Cheung Ruiz	ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Última Revisión: Fecha	Pág. 1 de 18



Índice

Tabla de contenido

<i>Integrantes</i>	<i>3</i>
<i>Título del Proyecto</i>	<i>3</i>
<i>Objetivos del Negocio del cual se basa el Proyecto</i>	<i>3</i>
<i>Descripción General</i>	<i>3</i>
<i>Descripciones Funcionales</i>	<i>3</i>
<i>Modelo Conceptual</i>	<i>7</i>
<i>Modelo Lógico</i>	<i>7</i>
<i>Reportes</i>	<i>8</i>
<i>Diccionario de Datos</i>	<i>10</i>
<i>Referencias</i>	<i>12</i>

Materia: Sistemas de Bases de Datos 1	Facultad de Ingeniería en Electricidad y Computación
Profesor: MSc. Irene M. Cheung Ruiz	ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Ultima Revisión: Fecha	Pág. 2 de 18

Integrantes

- Toledo Arcos Adrian Enrique
- Apolo Balcazar Diego Gonzalo
- Campozano Bravo Frank Jaren
- Motoche Motoche David Sebastian

Título del Proyecto

Plataforma de gestión de academias interdisciplinarias en Unidad Educativa Particular Daniel Comboni

Etapas de Desarrollo: Análisis del Sistema

Objetivos del Negocio del cual se basa el Proyecto

Reemplazar el antiguo sistema basado en Excel mediante la creación de un sistema de base de datos que optimice la recepción de información para los miembros de la academia (entrenadores, coordinadores).

Etapas de Desarrollo: Análisis del Sistema

Descripción General

Llevar un registro de los estudiantes registrados en las academias del colegio (basquetbol, fútbol sala, voleibol, cheerleader, robótica y boxeo), cada una teniendo también subcategorías. Guardar los datos del estudiante, sus pagos de la pensión, entre otros datos. Todo esto con una plataforma de fácil acceso para los docentes encargados y autoridades.

Etapas de Desarrollo: Análisis del Sistema

Descripciones Funcionales

1. Definir los procesos especificados en el desarrollo del proyecto.

Nombre del Rol	Descripción
Coordinador	Es el responsable de la gestión general de las academias, incluyendo la aprobación del registro y la supervisión de los entrenadores .
Entrenador/Docente	Es el encargado de impartir la enseñanza en una academia específica. Puede registrar y editar información de entrenamientos/clases y marcar la participación de los estudiantes en competencias.
Representante	Es la persona responsable del estudiante. Puede consultar las deudas pendientes y las cláusulas de pago del estudiante.
Sistema (Plataforma)	El sistema automatizado de bases de datos que gestiona los registros, los pagos, y genera los reportes.

2. Identificar cada uno de los diferentes roles de las personas y sistemas externos involucrados en los procesos y Detallar cada uno de los roles.		
Paso	Rol Involucrado	Detalle de la Función
1. Ingreso de Datos	Representante / Coordinador	Se ingresa la información completa del Estudiante (nombre, fecha de nacimiento, promedio, curso, etc.) y la del Representante (nombre, teléfono, correo)8.
2. Selección de Academia	Representante / Coordinador	Se selecciona una o más Academias (Baloncesto, Robótica, etc.) a las que se inscribe el estudiante. El sistema asocia el ID_estudiante con el ID_academia a través de la relación de inscripción.
3. Verificación de Cupo	Sistema	El sistema verifica el campo n_estudiantes en la tabla Academia para asegurar que el cupo máximo no se haya excedido.
4. Asignación Financiera	Sistema	El sistema crea automáticamente un registro en la tabla Pago con el monto de la pensión y una Deuda inicial si corresponde.
Paso	Rol Involucrado	Detalle de la Función
5. Aprobación y Activación	Coordinador	El Coordinador revisa y aprueba el registro. El sistema activa el acceso del estudiante a la academia.

<p>3. Detallar cada una de las reglas de procedimiento haciendo uso del prototipo.</p> <p>1.- Autenticación y Acceso (Regla #5): El sistema requiere que los usuarios se identifiquen. El prototipo lo muestra con el ícono de "Cuenta" y el nombre del usuario logueado, "Diego" , indicando una sesión activa.</p> <p>2.-Validación Académica (Reglas #4, #6): El sistema verifica el promedio del estudiante (mínimo 8.5) antes de la inscripción. Esto se refleja en la columna "Promedio" de la tabla de Estudiantes y es una condición para la función "Agregar estudiante".</p> <p>3.-Validación Administrativa (Regla #3): Secretaría debe validar la matrícula antes de que un alumno pueda ser añadido usando la función "Agregar estudiante".</p> <p>4.-Registro de Asistencia (Regla #9): El docente registra la asistencia en cada sesión. El prototipo tiene una pestaña dedicada de "Asistencia" para esta función específica.</p> <p>5.-Control de Pagos y Deudas (Reglas #7, #10): El sistema gestiona el estado de pensiones. La pestaña "Deudas" permite al docente verificar quiénes tienen pagos pendientes y aplicar la restricción de participación.</p>

6.-Límite de Inscripción (Regla #8): El sistema bloquea las inscripciones después de dos meses. Esto significa que la función **"Agregar estudiante"** debe desactivarse automáticamente pasada esa fecha límite.

4. Detallar las restricciones que posea el sistema.


1.-Restricción Académica: Un estudiante no puede inscribirse ni permanecer en la academia si su promedio general es inferior a 8.5.

2.- Restricción Administrativa: Un estudiante no puede ser agregado a una academia si no está previamente matriculado en la unidad educativa (validado por Secretaría).

3.-Restricción Financiera: Un estudiante no puede participar en las actividades (y se bloquea su registro de asistencia) si acumula más de dos meses de atraso en las pensiones.

4.-Restricción Temporal: La inscripción a las academias se bloquea automáticamente para los estudiantes después de los dos primeros meses del inicio de clases.

5.- Restricción de Acceso (Roles): El acceso a la información está segregado por roles. Un docente (como "Diego") solo puede ver y gestionar los estudiantes, horarios y asistencias de las academias que tiene asignadas (ej. "Baloncesto"), pero no de otras (ej. "Fútbol" o "Robótica").

ACADEMIAS
Diego 

UNIDAD EDUCATIVA



DANIEL COMBONI

Baloncesto

Kickboxing


Futbol

Robótica

Cheerleader

Voleibol

Contáctanos:
secretaria@uepdc.edu.ec
+593 98 765 4321



Diego

BALONCESTO

23 estudiantes

DOCENTE

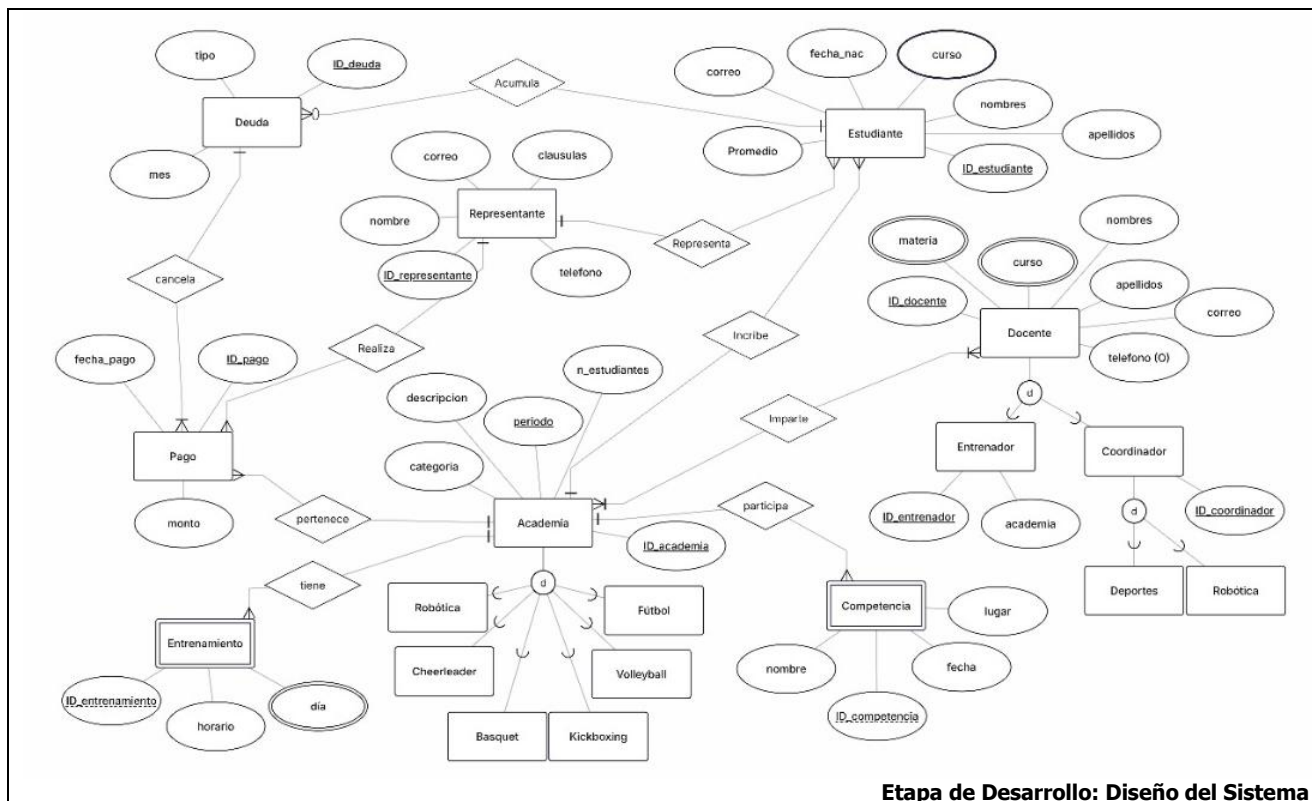
Braulio Sánchez
braulsancheza@docente.uepdc.edu.ec
0987654321

Estudiantes	Asistencia	Horario	Deudas														
Docente encargado Braulio Erick Sánchez Arellano Editar Lenin Alberto Pérez (coordinador) Editar		Horario Hora de entrenamiento: 14:30 – 17:00 Editar <table border="1"> <thead> <tr> <th>Lunes</th><th>Martes</th><th>Miércoles</th><th>Jueves</th><th>Viernes</th><th>Sábado</th><th>Domingo</th></tr> </thead> <tbody> <tr> <td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table>		Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo							
Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo											
Estudiantes Agregar estudiante Quitar estudiante Actualizar datos de estudiante																	
Academia Modificar mensualidad Suspender academia																	

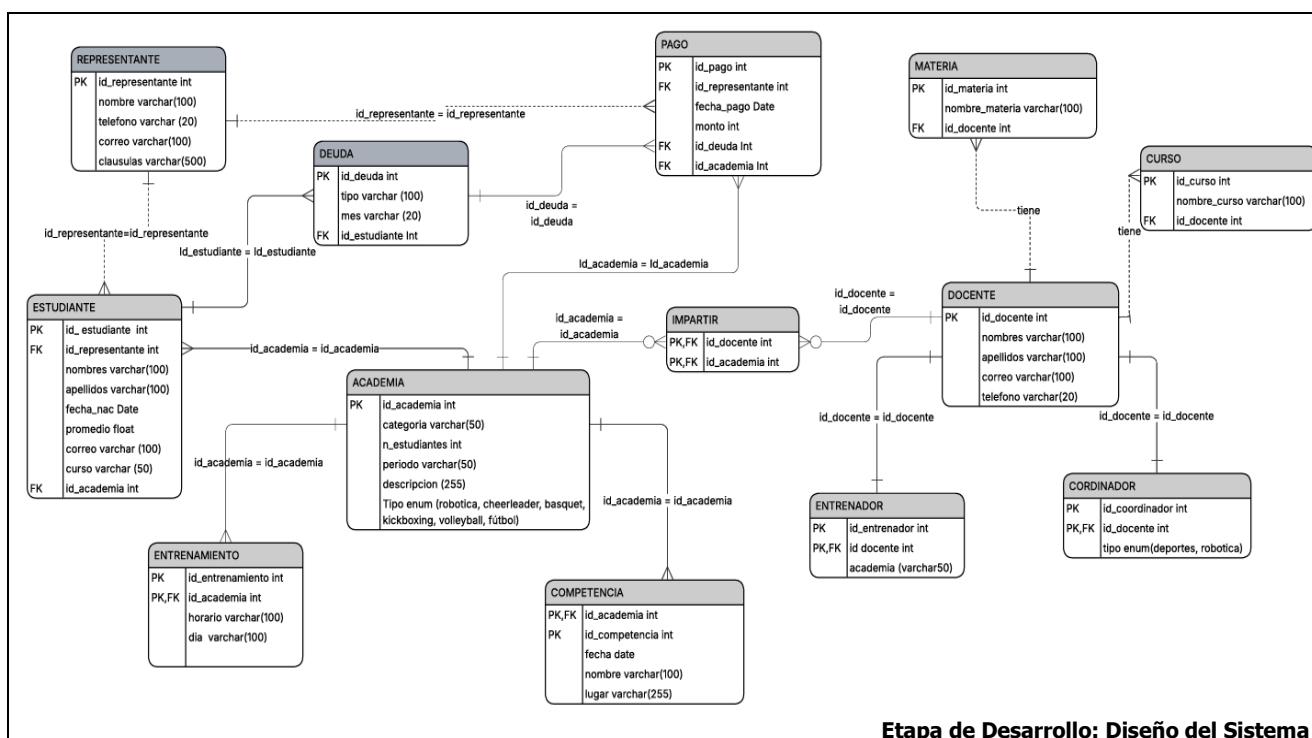
[illegible]

Etapas de Desarrollo: Diseño del Sistema

Modelo Conceptual



Modelo Lógico





Reportes

1. Nombres de estudiantes con deudas en una academia específica.

Este reporte busca el nombre y el monto de la deuda de los estudiantes que pertenecen a una academia de categoría 'Deportiva' y que tienen deudas.

Álgebra Relacional:

π nombre_E, deudas (σ deudas > 0 \wedge categoria = 'Deportiva')
(ESTUDIANTE \bowtie ACADEMIA))

Explicación:

1. **Join :** Combina ESTUDIANTE y ACADEMIA usando id_academia.
2. **Selección:** Filtra el resultado para las filas donde deudas > 0 y categoria = 'Deportiva'.
3. **Proyección:** Muestra solo las columnas nombre y deudas.

2. Nombre de los docentes que también son coordinadores.

Este reporte genera una lista de los nombres de los docentes que también aparecen en la tabla COORDINADOR, ordenada alfabéticamente.

Álgebra Relacional:

τ nombre_D asc (π nombre_D (DOCENTE \bowtie COORDINADOR))

Explicación:

1. **Join Natural:** Combina DOCENTE y COORDINADOR usando la columna común (id_docente). El resultado *solo* incluirá a las personas que están en ambas tablas.
2. **Proyección:** Extrae solo la columna nombre del resultado.
3. **Ordenamiento:** Ordena la lista de nombres alfabéticamente.

3. Número de estudiantes por cada categoría de academia

Este reporte cuenta cuántos estudiantes hay en cada tipo de academia, agrupando los resultados por el nombre de la categoría.

Álgebra Relacional:

γ categoria; count(id_estudiante) \rightarrow n_estudiantes (ACADEMIA \bowtie ESTUDIANTE)

Explicación:

1. **Join natural:** Combina ACADEMIA y ESTUDIANTE usando id_academia.
2. **Agrupamiento:** Agrupa las filas por categoria y cuenta (COUNT) los id_estudiante en cada grupo.
3. **Proyección:** Muestra la categoria y el conteo (num_estudiantes).

4. Estudiantes de Alto Rendimiento que Requieren Ayuda Financiera

Este reporte ayuda a identificar a los estudiantes que sobresalen académicamente (alto promedio) pero que podrían estar teniendo dificultades financieras (tienen deudas pendientes). Esto es crucial para la asignación de becas y la retención del talento superior.

Álgebra Relacional:

π nombre_E, promedio, deudas, nombre_R, telefono (σ promedio $\geq 80 \wedge$ deudas > 0 (ESTUDIANTE \bowtie REPRESENTANTE))

Explicación:

1. **Natural Join:** Combina la información de las tablas ESTUDIANTE y REPRESENTANTE usando su columna común id_representante
2. **Selección:** Filtra las filas resultantes para incluir solo aquellos estudiantes que tienen un promedio ≥ 80 Y tienen deudas > 0 .
3. **Proyección:** Muestra únicamente las columnas nombre_E, promedio, deudas, nombre_R y telefono para el reporte final.

Etapas de Desarrollo: Diseño del Sistema

Materia: Sistemas de Bases de Datos 1	Facultad de Ingeniería en Electricidad y Computación
Profesor: MSc. Irene M. Cheung Ruiz	ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Última Revisión: Fecha	Pág. 9 de 18

Diccionario de Datos

Se debe de detallar cada una de las relaciones del modelo lógicos, siga el siguiente ejemplo.

Tabla: Academia			
Atributo	Tipo de Dato	Dominio	Descripción
ID_Academia	Int	Contiene todos los caracteres posibles	Identificador único de cada academia.
Nombre_a	VarChar(255)	Contiene todas las letras del alfabeto	Guarda el nombre de la academia.
Categoria	VarChar(30)	Contiene todos los caracteres posibles separando los numéricos de los alfabéticos por un guion	Guarda la categoría de las academias según el límite de edad
Tipo_a	Enum(robótica, cheerleader, basquet, kickboxin, volleyball, futbol)	Contiene una opción elegir la especialización	Identificar el área de especialidad de la academia

Tabla: Docente			
Atributo	Tipo de Dato	Dominio	Descripción
Id_docente	Int	Contiene todos los caracteres posibles	Identificador unico de cada docente
Nombre	VarChar(255)	Contiene todos los caracteres posibles	Guarda el nombre de cada docente
Curso	VarChar(30)	Contiene todos los caracteres posibles	Guarda el curso donde imparte catedra
Materia	VarChar(255)	Contiene todos los caracteres posibles	Representa la materia impartida

Tabla: Entrenador			
Atributo	Tipo de Dato	Dominio	Descripción
Id_Entrenador	Int	Contiene todos los caracteres posibles	Identificador único de cada entrenador
Nombre_e	VarChar(255)	Contiene todas las letras del alfabeto	Guarda el nombre del entrenador
Academia	VarChar(30)	Contiene todas las letras del alfabeto	Guarda la academia donde se especializa el entrenador

Tabla: Coordinador			
Atributo	Tipo de Dato	Dominio	Descripción
Id Coordinador	Int	Contiene todos los caracteres posibles	Identificador único para cada Coordinador
Nombre_Coord	VarChar(255)	Contiene todas las letras del alfabeto	Guarda el nombre del coordinador
Tipo_c	Enum	Contiene una opción elegir la especialización del coordinador	Identificar el área de especialización del coordinador

Tabla: Entrenamiento			
Atributo	Tipo de Dato	Dominio	Descripción
ID_Entrenamiento	Int	Contiene todos los caracteres posibles	Identificador único de cada sesión
Fecha/Hora	DateTime	Formato fecha abarca desde el periodo actual	Guarda la fecha del estudiante
Tabla: Representante			
Atributo	Tipo de Dato	Dominio	Descripción
Id_Representante	Int	Contiene todos los caracteres posibles	Identificador único del representante.
Nombre_R	VarChar(255)	Contiene todas las letras del alfabeto	Guarda el nombre del representante
Clausulas	VarChar(255)	Contiene todas las letras del alfabeto	Indica el tipo de clausula que falta por cumplir
Teléfono	VarChar(30)	Contiene todos los caracteres posibles	Guarda el número de telf. del representante
Correo	VarChar(255)	Contiene todos los caracteres posibles	Guarda el correo del representante.
Tabla: Estudiante			
Atributo	Tipo de Dato	Dominio	Descripción
Id_Estudiante	Int	Contiene todos los caracteres posibles	Identificador único de cada estudiante
Nombre_e	VarChar(255)	Contiene todas las letras del alfabeto	Guarda el nombre del estudiante
Correo	VarChar(255)	Contiene todos los caracteres posibles	Guarda la dirección de correo del estudiante
Deudas	boolean	Contiene todos los caracteres posibles	Nos da verdadero o falso dependiendo de si el representante realizo el pago
Curso	VarChar(30)	Contiene todos los caracteres posibles	Guarda el curso escolar que está cursando el estudiante
Fecha_nac	Date	Contiene todas las letras del alfabeto	Guarda la fecha de nacimiento para no tener un valor volátil como la edad

Tabla: Competencia			
Atributo	Tipo de Dato	Dominio	Descripción
Id_Competencia	Int	Contiene todos los caracteres posibles	Identificador único de cada competencia
Nombre	VarChar(255)	Contiene todas las letras del alfabeto	Guarda el nombre de la competencia
Fecha	Date	Formato de fecha que abarca el inicio y fin de la competencia	Guarda la fecha de la competencia
Lugar	VarChar(255)	Contiene todos los caracteres posibles	Guarda la ubicación de donde se realiza la competencia.
Tabla: Pago			
Atributo	Tipo de Dato	Dominio	Descripción
Id_Pago	Int	Contiene todos los caracteres posibles	Identificador único de cada pago
Deuda	Float	Contiene datos flotantes para abarcar los centavos	Guarda la deuda acumulada por el representante hasta el límite (monto*3)
Fecha_pago	Date	Formato que abarca desde 2025 hasta el fin del periodo	Guarda la fecha de la comprobación del pago
Monto	Float	Contiene datos flotantes para abarcar los centavos	Guarda el monto a depositar mensualmente.

Etapas de Desarrollo: Diseño del Sistema

Referencias

Nuestras referencias para el desarrollo de los avances, nacieron a partir de entrevistas realizadas al personal participe dentro de las academias (Administradora general y entrenador):

<https://drive.google.com/drive/folders/1F56U5APUfi4eZJ31HKmMGrdr8hjO40aq>

DESCRIPCIÓN DE TRIGGERS

Se han implementado dos disparadores (triggers) en la base de datos para automatizar la integridad referencial de los cupos en cada academia, eliminando la necesidad de cálculos manuales desde la aplicación.

- Nombre: `trg_incrementar_cupos`
 - Evento: Se activa AFTER INSERT (después de insertar) en la tabla estudiante.
 - Función: Incrementa automáticamente en 1 el campo `n_estudiantes` de la academia correspondiente al realizar una nueva inscripción.
- Nombre: `trg_decrementar_cupos`
 - Evento: Se activa AFTER DELETE (después de eliminar) en la tabla estudiante.
 - Función: Disminuye en 1 el contador de la academia cuando un estudiante es dado de baja, garantizando que el número de inscritos reflejado sea siempre real.

DESCRIPCIÓN DE STORED PROCEDURES

Se encapsularon las operaciones de manipulación de datos (CRUD) en procedimientos almacenados para garantizar la seguridad, integridad y atomicidad de las transacciones.

- Nombre: `sp_crear_estudiante`
 - Función: Inserta un nuevo registro en la tabla estudiante.
 - Lógica: Inicia una transacción, valida que el promedio se encuentre en el rango permitido (0-10) y ejecuta la inserción. En caso de fallo, aplica un ROLLBACK automático.
- Nombre: `sp_actualizar_estudiante`
 - Función: Modifica los datos de un estudiante existente.
 - Lógica: Recibe los parámetros modificables (nombres, apellidos, promedio, curso) y actualiza el registro específico mediante su ID, asegurando la persistencia de cambios válidos.
- Nombre: `sp_eliminar_estudiante`
 - Función: Elimina un estudiante del sistema.
 - Lógica: Ejecuta el borrado del registro por ID dentro de una transacción. Esta acción detona automáticamente el trigger `trg_decrementar_cupos`.

DESCRIPCIÓN DE ÍNDICES

Se crearon 5 índices no agrupados (Non-Clustered) para optimizar el tiempo de respuesta en las consultas recurrentes del sistema:

1. `idx_estudiante_apellido`: Optimiza la búsqueda de estudiantes por el campo 'apellidos', agilizando la generación de fichas y reportes sin requerir un escaneo completo de la tabla.

Materia: Sistemas de Bases de Datos 1	Facultad de Ingeniería en Electricidad y Computación
Profesor: MSc. Irene M. Cheung Ruiz	ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Última Revisión: Fecha	Pág. 13 de 18

2. idx_representante_telefono: Acelera la validación de duplicados al momento de la inscripción, utilizando el teléfono como criterio de búsqueda única.
3. idx_pago_fecha: Mejora el rendimiento de los reportes financieros filtrados por rangos temporales (campo fecha_pago).
4. idx_competencia_fecha: Permite el ordenamiento y filtrado cronológico eficiente en el módulo de eventos deportivos.
5. idx_docente_apellido: Facilita la búsqueda y asignación de docentes en los formularios administrativos mediante el campo 'apellidos'.

DESCRIPCIÓN DE USUARIOS Y ROLES

Se definió un esquema de seguridad basado en el principio de menor privilegio, con 5 perfiles de usuario:

Usuario	Rol	Permisos Asignados	Alcance
admin_sistema	Administrador	ALL PRIVILEGES	Control total de la base de datos.
secretaria	Operativo	SELECT, EXECUTE	Ver datos y ejecutar SPs de gestión de estudiantes.
rector	Gerencial	SELECT	Solo lectura de Vistas (Reportes). no tiene permiso para acceder directamente a las tablas originales donde se guardan los datos
cajero	Financiero	SELECT, INSERT	Gestión exclusiva de la tabla pago.
auditor	Auditoría	SELECT	Lectura de tablas críticas (pago, asistencia).

REQUISITOS E INSTALACIÓN

LIBRERÍAS NECESARIAS

El sistema requiere las siguientes bibliotecas de Python:

- streamlit
- mysql-connector-python
- pandas

Comando de instalación:

```
> pip install streamlit mysql-connector-python pandas
```

CONFIGURACIÓN DE LA BASE DE DATOS

PASO 1: Despliegue de Base de Datos

1. Abra su gestor SQL (ej. MySQL Workbench).
2. Ejecute el script SQL completo para crear la estructura de las 13 tablas, cargar los datos semilla y generar los objetos lógicos (Triggers, SPs, Vistas y Usuarios).

PASO 2: Configuración de Conectividad

1. Abra el archivo "CRUD.py".
2. Ubique la función "create_connection".
3. Actualice el parámetro 'password' con la credencial de su servidor MySQL local.

Materia: Sistemas de Bases de Datos 1	Facultad de Ingeniería en Electricidad y Computación
Profesor: MSc. Irene M. Cheung Ruiz	ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Última Revisión: Fecha	Pág. 14 de 18

GUÍA DE USO - MÓDULOS DEL SISTEMA

MÓDULO DE REPORTES

- Función: Tablero de visualización de información académica y financiera.
- Uso: Navegue a la sección "Reportes Globales".
- Resultado: El sistema consulta las Vistas SQL (vista_ficha_estudiante, vista_historial_pagos, etc.) para mostrar información consolidada en tiempo real. Estos datos son de solo lectura.

B. MÓDULO DE GESTIÓN DE ESTUDIANTES

Este módulo interactúa exclusivamente a través de Procedimientos Almacenados:

1. Registrar (Alta):

- Complete el formulario con los datos del alumno, representante y academia.
- Al guardar, el sistema llama a `sp_crear_estudiante`. Si el registro es exitoso, el cupo de la academia se descuenta automáticamente.

2. Actualizar (Modificación):

- Ingrese el ID del estudiante y los nuevos datos.
- El sistema ejecuta `sp_actualizar_estudiante` para modificar la información de manera segura.

3. Eliminar (Baja):

- Ingrese el ID del estudiante a dar de baja.
- Al confirmar, se ejecuta `sp_eliminar_estudiante`, lo que elimina el registro y libera un cupo en la academia (vía Trigger).

SOLUCIÓN DE PROBLEMAS FRECUENTES

ERROR: "Access denied for user 'root'@'localhost'"

- Solución: La contraseña en el script Python es incorrecta. Verifique la credencial en la función de conexión.

ERROR: "Can't connect to MySQL server"

- Solución: Verifique que el servicio de MySQL esté iniciado en XAMPP o en los servicios de Windows.

ERROR: Promedio inválido

- Solución: El Procedimiento Almacenado incluye una validación interna que rechaza promedios fuera del rango 0-10. Corrija el valor ingresado.

Materia: Sistemas de Bases de Datos 1	Facultad de Ingeniería en Electricidad y Computación
Profesor: MSc. Irene M. Cheung Ruiz	ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Última Revisión: Fecha	Pág. 15 de 18

JUSTIFICACIÓN TÉCNICA DEL PROYECTO

JUSTIFICACIÓN DE ÍNDICES (OPTIMIZACIÓN)

Se han implementado índices no agrupados para optimizar el rendimiento de las consultas más frecuentes dentro de la aplicación, reduciendo el costo computacional de las operaciones de lectura:

1. `idx_estudiante_apellido` y `idx_docente_apellido`
 - Tablas afectadas: Estudiante, Docente.
 - Justificación: El flujo de trabajo administrativo requiere búsquedas constantes de personas por sus apellidos para procesos como la toma de asistencia, edición de perfiles y asignación de cursos. La indexación de estos campos evita el escaneo completo de la tabla (Full Table Scan), mejorando la velocidad de respuesta en la interfaz de usuario.
2. `idx_representante_telefono`
 - Tabla afectada: Representante.
 - Justificación: Este índice agiliza la verificación de duplicados al momento de registrar nuevos tutores. Al ser un campo de búsqueda exacta, el índice permite validar la existencia del representante de manera inmediata antes de proceder con una inscripción.
3. `idx_pago_fecha` y `idx_competencia_fecha`
 - Tablas afectadas: Pago, Competencia.
 - Justificación: Los módulos de reportes financieros y cronogramas deportivos aplican filtros por rangos de fechas (mensuales o anuales). Estos índices permiten al motor de base de datos ordenar y filtrar temporalmente los registros de forma eficiente.

DESCRIPCIÓN DE TRIGGERS (INTEGRIDAD REFERENCIAL)

Se implementaron disparadores (Triggers) para automatizar la actualización de contadores y asegurar la consistencia de los datos en tiempo real:

1. `trg_incrementar_cupos`
 - Evento: AFTER INSERT en tabla Estudiante.
 - Función: Incrementa automáticamente el campo 'n_estudiantes' en la tabla Academia cada vez que se completa una inscripción. Esto elimina la necesidad de realizar conteos manuales (COUNT) repetitivos para verificar la disponibilidad de cupos.
2. `trg_decrementar_cupos`
 - Evento: AFTER DELETE en tabla Estudiante.
 - Función: Disminuye el contador de estudiantes de la academia correspondiente al eliminar un registro, garantizando que la cifra de inscritos mostrada en el sistema refleje siempre la realidad.

USO DE STORED PROCEDURES (SEGURIDAD Y LÓGICA DE NEGOCIO)

Las operaciones de manipulación de datos (CRUD) para la entidad Estudiante se encapsularon en Procedimientos Almacenados (`sp_crear_estudiante`, `sp_actualizar_estudiante`, `sp_eliminar_estudiante`) por las siguientes razones técnicas:

1. Integridad Transaccional (ACID): Se implementaron bloques de transacción (START TRANSACTION, COMMIT, ROLLBACK). Esto asegura que las operaciones sean atómicas; en

Materia: Sistemas de Bases de Datos 1	Facultad de Ingeniería en Electricidad y Computación
Profesor: MSc. Irene M. Cheung Ruiz	ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Última Revisión: Fecha	Pág. 16 de 18



caso de error durante la ejecución, se revierten todos los cambios para evitar datos corruptos o incompletos.

2. Seguridad: Se restringe el acceso directo a las tablas. El perfil de usuario "Secretaria" posee permisos de ejecución (EXECUTE) sobre los procedimientos, pero no permisos directos de INSERT o UPDATE sobre las tablas base, protegiendo la estructura ante errores humanos o inyecciones SQL.

3. Validación de Datos: Los procedimientos incluyen lógica de validación interna (ej. verificar rangos de promedios) antes de persistir la información.

USO DE REPORTES / VISTAS (ABSTRACCIÓN DE DATOS)

Se crearon Vistas SQL para simplificar consultas complejas y presentar la información denormalizada para los reportes del sistema:

1. vista_ficha_estudiante: Unifica la información dispersa en las tablas de Estudiante, Academia y Representante, permitiendo obtener el perfil completo del alumno en una sola consulta simple.

2. vista_historial_pagos: Cruza la información transaccional de pagos con los datos del representante y la categoría de la academia para facilitar la auditoría financiera.

3. vista_competencias_disciplina: Abstrae la relación entre las competencias y las academias, permitiendo visualizar los resultados deportivos junto con la disciplina correspondiente (obtenida del tipo de academia) sin requerir JOINS complejos en la capa de aplicación.

4. vista_asignacion_entrenadores: Facilita la visualización del staff docente, mostrando los nombres de los entrenadores y la academia asignada en un formato legible para el usuario final.

Materia: Sistemas de Bases de Datos 1	Facultad de Ingeniería en Electricidad y Computación
Profesor: MSc. Irene M. Cheung Ruiz	ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Última Revisión: Fecha	Pág. 17 de 18



Materia: Sistemas de Bases de Datos 1	Facultad de Ingeniería en Electricidad y Computación
Profesor: MSc. Irene M. Cheung Ruiz	ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL
Ultima Revisión: Fecha	Pág. 18 de 18