

**CURSO DE SQL**

**COMISIÓN: 47365**

**ENTREGA DEL PROYECTO FINAL**

**Alumno:** Sabio Bezus Alejandro German

**Profesor:** Camilo Andres Redondo

**Tutor:** Lautaro Odoni

**AÑO 2023**

**ÍNDICE**

[**INTRODUCCIÓN**](#_41p2o12dddi6) **3**

[**OBJETIVO.**](#_hi7swx8bri7h) **4**

[**SITUACIÓN PROBLEMÁTICA.**](#_6hxr6efa48e1) **5**

[**MODELO DE NEGOCIO.**](#_cail5aecs738) **6**

[**DIAGRAMA DE ENTIDAD RELACIÓN.**](#_nkn41zmnd8cf) **7**

[**LISTADO DE TABLAS CON DESCRIPCIÓN DE ESTRUCTURA.**](#_64i5c8ap6a36) **9**

[**LISTADO DE VISTAS**](#_edun93p4xgnk) **20**

[**LISTADO DE FUNCIONES**](#_3m48rkw8hzpc) **23**

[**LISTADO DE STORED PROCEDURES**](#_30ler77cwbfp) **24**

[**LISTADO DE TRIGGERS.**](#_niu0z5jooqr5) **27**

[**USUARIOS MySQL Y PERMISOS.**](#_4ys4zb36ws3d) **32**

[**IMPLEMENTACIÓN DE SENTENCIAS DEL SUBLENGUAJE TCL.**](#_67z996rwgc5) **33**

[**INFORMES GENERADOS EN BASE A LA INFORMACIÓN DE LA BASE.**](#_aamx8mmo5hly) **35**

[**HERRAMIENTAS Y TECNOLOGÍAS USADAS.**](#_p6xga7ewqqt7) **38**

[**FUTURAS LÍNEAS.**](#_h22hvaonh6zb) **40**

# 

# 

# 

# **INTRODUCCIÓN**

En el contexto de mi proyecto final, he asumido la decisión estratégica de dar forma a una base de datos altamente especializada, concebida exclusivamente para la gestión integral de un gimnasio. Este emprendimiento no solo representa un paso significativo en mi formación académica, sino que también se erige como un pilar esencial en la optimización de la administración y el control meticuloso de diversas facetas críticas en el entorno empresarial de un gimnasio.

A lo largo de este apasionante trayecto, he dedicado tiempo y esfuerzo incansables al diseño y desarrollo de una serie de tablas, cada una meticulosamente estructurada para desempeñar un papel fundamental en la operación fluida y la eficiencia del gimnasio. Cada tabla, concebida con un propósito específico, se ha convertido en un componente esencial para el seguimiento detallado de información crucial en áreas que van desde la asistencia de los alumnos hasta el control del inventario y el análisis detallado de registros de salud y rendimiento.

Lo que comenzó como un conjunto inicial de 10 tablas durante el curso ha evolucionado considerablemente, expandiéndose a más de 15 tablas al culminar la etapa académica. Este conjunto ampliado no solo fortalece la capacidad de la base de datos para generar informes cruciales para la gestión eficiente del gimnasio, sino que también sienta las bases sólidas para una potencial implementación en una plataforma web integral.

La versatilidad de estas tablas no solo radica en su capacidad para brindar informes detallados que facilitan la toma de decisiones informadas, sino que también se vislumbra la posibilidad de aprovechar esta base de datos en el diseño e implementación de una página web. Esta plataforma web no solo ofrecerá a los usuarios del gimnasio un entorno amigable para su día a día, sino que también proporcionará una herramienta valiosa para los encargados y dueños del gimnasio, simplificando la tarea de monitoreo y administración continua. En esencia, se proyecta que esta base de datos no solo sea una herramienta de gestión interna sino también un activo estratégico que mejore la experiencia general tanto para usuarios como para administradores del gimnasio.

# **OBJETIVO**.

Este proyecto se articula en una tríada estratégica destinada a redefinir la gestión del gimnasio a través de la implementación de una base de datos sólida y relacional respaldada por un modelo de negocio perspicaz. Cada objetivo trazado se convierte en un pilar fundamental que se entrelaza para fortalecer la eficacia y eficiencia del gimnasio en su totalidad.

En el primer frente, mi compromiso radica en erigir la piedra angular de esta iniciativa: la creación de una base de datos robusta y relacional. Esta no será una simple colección de tablas y relaciones; será un cimiento estratégico que refleje la esencia de las operaciones diarias del gimnasio. Al abordar las complejidades inherentes a la gestión de alumnos, cursos, entrenadores y el seguimiento detallado de registros de salud y rendimiento, la estructura de la base de datos aspira a ser un habilitador esencial de la eficiencia operativa.

En el segundo frente, mi enfoque se despliega hacia el desarrollo de objetos que trascienden la mera creación, extendiéndose hacia la garantía de la integridad y seguridad de la base de datos. La implementación de procedimientos almacenados, disparadores y otras herramientas dinámicas se traducirá en una administración eficaz de la información almacenada, proporcionando a la base de datos la flexibilidad necesaria para evolucionar con las futuras dinámicas y expansiones del gimnasio.

El tercer y último frente concentra su energía en la implementación de consultas SQL como catalizadores para la generación de informes sustanciales. Estas consultas no solo extraerán datos; revelarán patrones y tendencias cruciales para decisiones informadas. El propósito es ir más allá de la acumulación de información, proporcionando herramientas analíticas que empoderen a los responsables del gimnasio a comprender y optimizar la dinámica operativa.

La amalgama de estos tres objetivos, lejos de ser un simple acto técnico, aspira a crear una infraestructura dinámica que no solo sustente sino transforme la gestión del gimnasio. Esta base de datos no es solo una herramienta, sino un aliado estratégico destinado a impulsar la experiencia del usuario y la eficacia administrativa del gimnasio. En resumen, este proyecto no solo representa una implementación técnica, sino una empresa que cataliza el progreso.

# **SITUACIÓN PROBLEMÁTICA.**

La ausencia de un sistema centralizado para la gestión de información en el gimnasio ha generado una red de desafíos multifacéticos que han impactado negativamente en varios aspectos clave de la operación diaria. Entre las problemáticas identificadas se encuentra la dificultad en el seguimiento de la asistencia de los alumnos, una variable esencial que informa sobre la participación y compromiso de los usuarios. La carencia de un mecanismo efectivo para evaluar los horarios con mayor concurrencia ha limitado la capacidad del gimnasio para potenciar áreas específicas y, a su vez, optimizar la utilización de recursos y espacios.

Otro punto crítico que ha afectado la eficiencia operativa radica en la gestión de pagos y cuotas. La falta de un sistema centralizado dificulta la identificación precisa de los pagos pendientes, generando potenciales inconvenientes tanto para el gimnasio como para los usuarios. La necesidad de optimizar estos procesos se convierte en un componente esencial para garantizar la sostenibilidad financiera y la satisfacción del cliente.

Adicionalmente, la carencia de un sistema centralizado ha limitado el seguimiento del progreso individual de los alumnos. La falta de integración entre datos como el peso, el índice de masa corporal (IMC) y el rendimiento durante las sesiones de entrenamiento ha impedido una evaluación holística y personalizada del progreso. Este vacío en la información obstaculiza la capacidad del gimnasio para adaptar las rutinas y programas de entrenamiento de manera eficaz, privando a los usuarios de una experiencia más personalizada y efectiva.

Este proyecto se configura como una respuesta estratégica a estas problemáticas identificadas, ofreciendo una solución integral que va más allá de la gestión de inventario y la generación de informes. Su alcance abarca la optimización de procesos internos, la maximización de recursos a través de un análisis detallado de la asistencia y la implementación de herramientas que faciliten el seguimiento del progreso individual, fomentando así una experiencia de usuario enriquecedora y una gestión eficiente del gimnasio. La plataforma resultante no solo aborda los problemas actuales, sino que sienta las bases para un crecimiento continuo y una operación más eficaz en el futuro.

# **MODELO DE NEGOCIO.**

En el tejido esencial de este proyecto, el modelo de negocio se despliega como una arquitectura estratégica que fusiona la visión integral del gimnasio con la ejecución operativa. La piedra angular de este modelo radica en la oferta de servicios que trascienden la mera actividad física, incorporando entrenamiento especializado, cursos específicos y un seguimiento personalizado que refleja el compromiso profundo del gimnasio con la salud y el bienestar de sus alumnos.

Este modelo se distingue por su enfoque proactivo en la comprensión profunda de las necesidades individuales de los usuarios. No se limita a proporcionar un espacio de ejercicios convencional; en cambio, se embarca en la misión de crear un entorno que nutra el progreso integral de los usuarios. Los servicios de entrenamiento van más allá del desarrollo físico, incorporando dimensiones motivacionales y estratégicas para mejorar la adherencia al ejercicio. La oferta de cursos especializados complementa este enfoque, proporcionando conocimientos especializados que abarcan desde técnicas avanzadas de entrenamiento hasta prácticas de bienestar general.

La gestión eficiente de recursos constituye otro pilar fundamental en este modelo de negocio. Este aspecto no solo implica la optimización de la infraestructura física del gimnasio, sino también la asignación estratégica de entrenadores y la adaptación de programas que respondan a las tendencias y demandas del mercado. La eficiencia en la gestión de recursos no solo impulsa la rentabilidad del gimnasio, sino que también contribuye a la satisfacción del cliente al garantizar una experiencia sin inconvenientes.

En paralelo, el control del inventario emerge como un componente estratégico adicional. Asegurando la disponibilidad de equipos y materiales necesarios, la implementación de un sistema de seguimiento de inventario no solo facilita el acceso a los recursos, sino que también apunta a la planificación a largo plazo para la adquisición de nuevos equipos y la mejora constante de las instalaciones.

El modelo también se distingue por su enfoque en el seguimiento integral de la salud y el rendimiento de los usuarios. Más allá de las métricas convencionales de pérdida de peso, se busca una evaluación profunda que incluya factores como el índice de masa corporal (IMC), registros de peso y el rendimiento en las rutinas. Este enfoque personalizado no solo impulsa la fidelización de los usuarios al demostrar resultados tangibles, sino que también facilita la adaptación continua de los servicios para satisfacer las cambiantes necesidades de la comunidad de usuarios.

# **DIAGRAMA DE ENTIDAD RELACIÓN.**

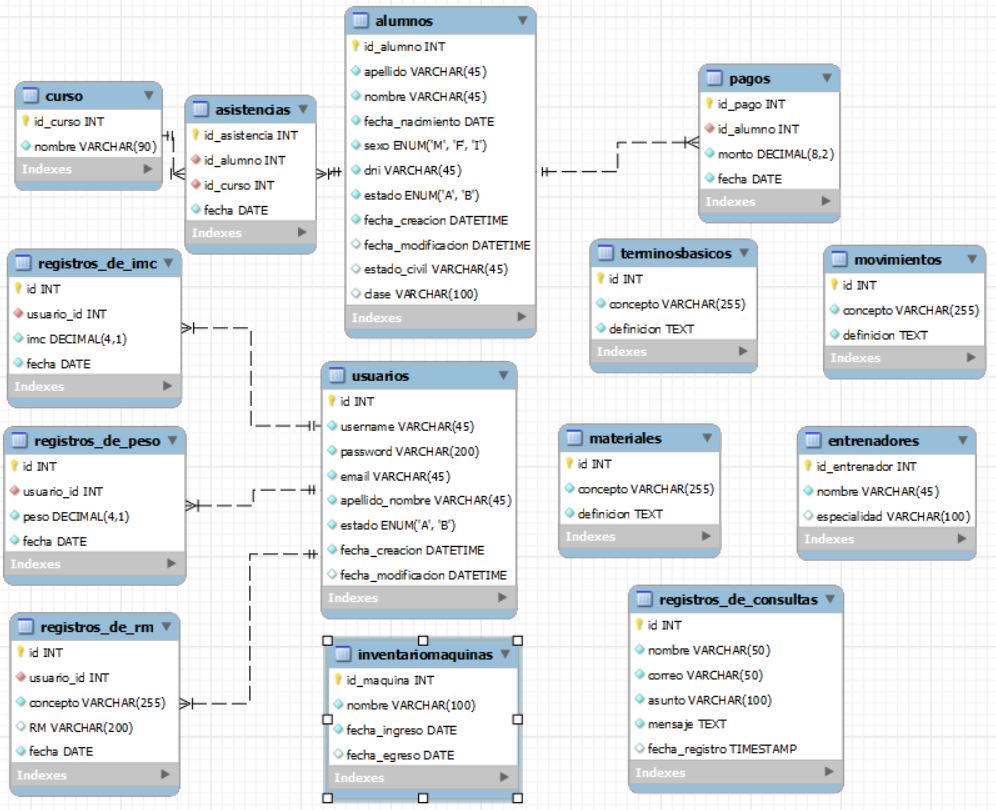
En una primera instancia, presentamos el Diagrama de Entidad Relación, una representación gráfica que encapsula la esencia estructural de la base de datos diseñada para el gimnasio. Este detallado diseño abarca un conjunto inicial de diez tablas, cada una de las cuales desempeña un papel crucial en la organización y gestión integral del establecimiento. Estas tablas, meticulosamente diseñadas para abordar aspectos específicos, se constituyen como los bloques fundamentales que sustentan la funcionalidad y eficiencia operativa del sistema.

Las diez tablas iniciales son las siguientes: Alumnos, Asistencias, Cursos, Entrenadores, InventarioMaquinas, Materiales, Movimientos, Pagos, Registros de Consultas, Registros de IMC, Registros de Peso, Registros de RM, Rutinas, TerminosBasicos y Usuarios. Cada una de estas tablas se entrelaza en un tejido sinérgico, contribuyendo de manera específica a la captura y organización de datos esenciales para la gestión integral del gimnasio.

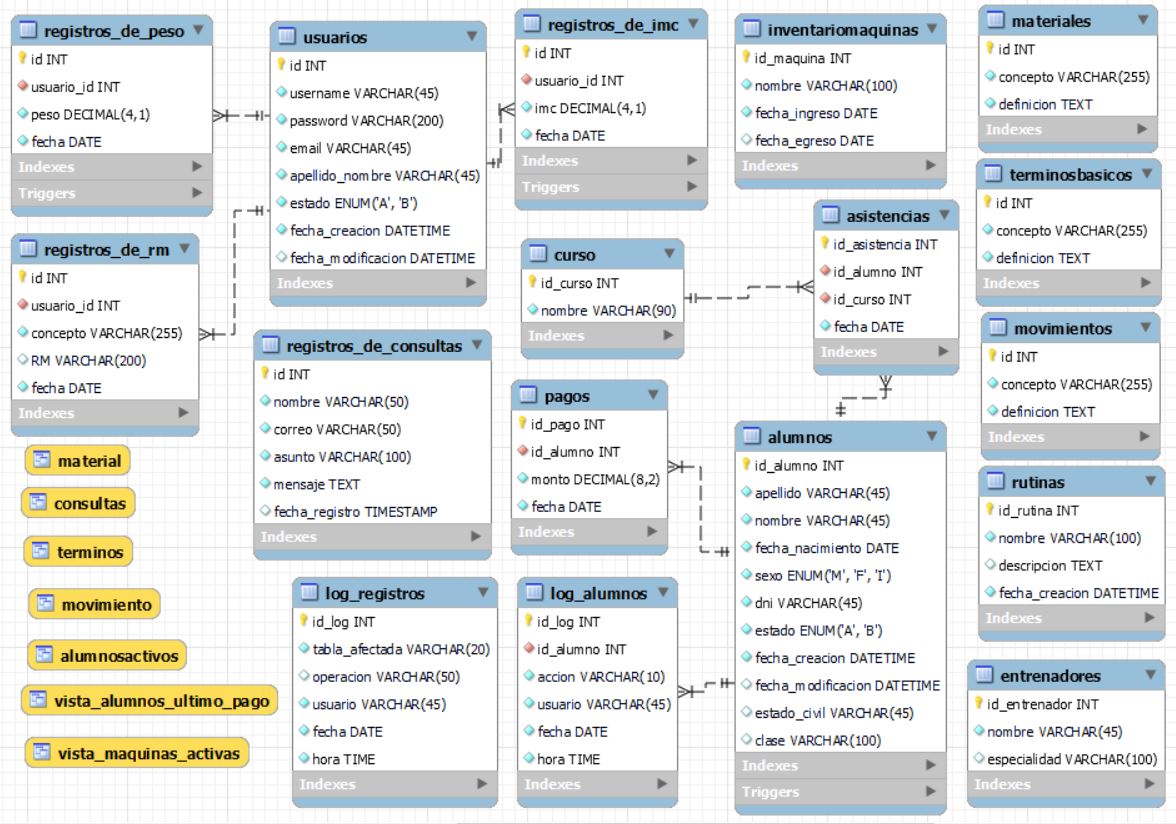
Posteriormente, se incorporan nuevas dimensiones al diagrama mediante la adición de más tablas, así como la inclusión de vistas y triggers. Esta expansión no solo enriquece la complejidad del diseño, sino que también proporciona herramientas adicionales para abordar situaciones específicas y optimizar procesos internos.

El Diagrama de Entidad Relación, más que una simple representación gráfica, sirve como un mapa detallado que orienta la navegación por la complejidad de la información. A través de esta herramienta visual, se busca ofrecer una visión panorámica de la interconexión y flujo de datos entre las diversas entidades, estableciendo así las bases para una gestión eficiente y una toma de decisiones informada en el entorno del gimnasio. La inclusión de vistas y triggers amplía la capacidad de la base de datos al permitir una manipulación más avanzada de los datos, mejorando la adaptabilidad y capacidad de respuesta del sistema ante diversas situaciones.

D.E.R. AL INICIO

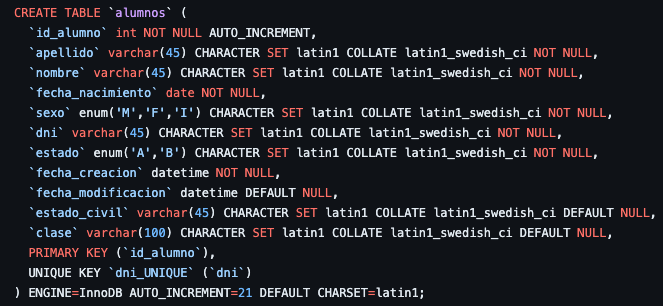


D.E.R. AL FINAL



# **LISTADO DE TABLAS CON DESCRIPCIÓN DE ESTRUCTURA.**

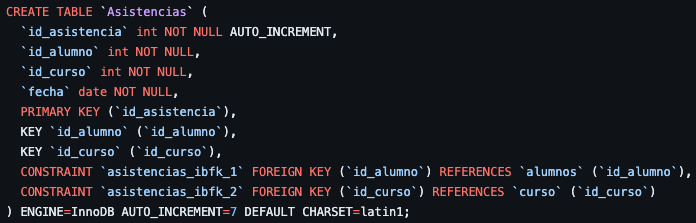
1. **Tabla: alumnos.**

****

* id\_alumno (ID): Este campo representa un identificador único para cada alumno en la base de datos. Es una clave primaria (PRIMARY KEY) y su tipo de dato es "int" (entero).
* apellido: Este campo almacena el apellido del alumno. Su tipo de dato es "varchar(45)" (cadena de caracteres de longitud máxima 45) y no puede estar vacío ("NOT NULL").
* nombre: Almacena el nombre del alumno. Su tipo de dato es "varchar(45)" y tampoco puede estar vacío.
* fecha\_nacimiento: Este campo registra la fecha de nacimiento del alumno. Su tipo de dato es "date" (fecha) y no puede estar vacío.
* sexo: Guarda información sobre el sexo del alumno y utiliza un tipo ENUM con valores 'M' (masculino), 'F' (femenino) e 'I' (indeterminado). No puede estar vacío.
* dni (ID): Es un campo que almacena el número de documento del alumno y también actúa como identificador único. Es una clave única (UNIQUE KEY) y su tipo de dato es "varchar(45)".
* estado: Indica el estado actual del alumno, donde 'A' significa activo y 'B' significa baja. Utiliza un tipo ENUM y no puede estar vacío.
* fecha\_creacion: Registra la fecha y hora en que se creó el registro del alumno. Su tipo de dato es "datetime" (fecha y hora) y no puede estar vacío.
* fecha\_modificacion: Almacena la fecha y hora de la última modificación del registro del alumno. Es de tipo "datetime" y permite valores nulos.
* estado\_civil: Guarda el estado civil del alumno. Su tipo de dato es "varchar(45)" y permite valores nulos.
* clase: Este campo registra la clase a la que está asignado el alumno, con un límite de 100 caracteres. Su tipo de dato es "varchar(100)" y permite valores nulos.

En resumen, la tabla "alumnos" se utiliza para almacenar información detallada sobre los alumnos del gimnasio, incluyendo sus datos personales, estado, y otros detalles relacionados con su membresía y participación en clases. El campo "id\_alumno" y "dni" actúan como identificadores únicos, mientras que "id\_alumno" es la clave primaria de la tabla.

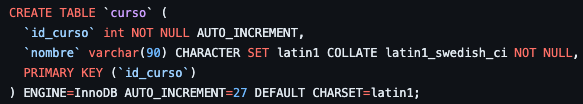
1. **Tabla: Asistencias.**

****

* **id\_asistencia (ID):** Este campo es un identificador único para cada registro de asistencia en la base de datos. Se genera automáticamente y es de tipo "int" (entero). Además, es una clave primaria (PRIMARY KEY), lo que garantiza su unicidad en la tabla.
* **id\_alumno:** Almacena el identificador único del alumno que está asociado a la asistencia. Este campo es de tipo "int" y no puede estar vacío ("NOT NULL"). Además, tiene una clave externa (FOREIGN KEY) que establece una relación con la tabla "alumnos" a través del campo "id\_alumno" en dicha tabla.
* **id\_curso:** Representa el identificador único del curso al que corresponde la asistencia. Similar al campo "id\_alumno", es de tipo "int" y no puede ser nulo ("NOT NULL"). También cuenta con una clave externa (FOREIGN KEY) que establece una relación con la tabla "curso" a través del campo "id\_curso" en dicha tabla.
* **fecha:** Indica la fecha en la que se registró la asistencia. Es de tipo "date" y no puede ser nulo ("NOT NULL").

Se utiliza para rastrear la asistencia de los alumnos a los diferentes cursos ofrecidos por el gimnasio. El campo "id\_asistencia" actúa como identificador único para cada registro de asistencia. Los campos "id\_alumno" e "id\_curso" están relacionados con las tablas "alumnos" y "curso", respectivamente, mediante claves foráneas. Esto permite vincular cada registro de asistencia a un alumno específico y al curso al que asistió. La columna "fecha" registra la fecha en que se tomó la asistencia. En conjunto, esta tabla es esencial para el seguimiento de la participación de los alumnos en los cursos del gimnasio.

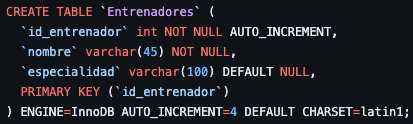
1. **Tabla: curso**

****

* **id\_curso (ID):** Este campo es un identificador único para cada curso en la base de datos. Es una clave primaria (PRIMARY KEY) y su tipo de dato es "int" (entero).
* **Nombre:** Almacena el nombre del curso, que describe el horario de un curso en el gimnasio. Su tipo de dato es "varchar(90)" (cadena de caracteres de longitud máxima 90) y no puede estar vacío ("NOT NULL").

En resumen, la tabla "curso" se utiliza para almacenar información sobre los cursos ofrecidos por el gimnasio y sus respectivos horarios. El campo "id\_curso" actúa como un identificador único y clave primaria para cada curso, mientras que "nombre" guarda el nombre del curso, que está relacionado con un horario específico. Esta tabla es importante para la gestión de turnos y asignación de alumnos a clases en el gimnasio.

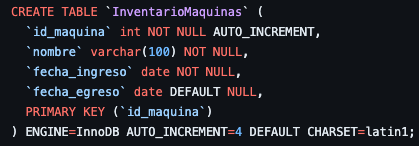
1. **Tabla: Entrenadores.**



* **id\_entrenador (ID):** Este campo es un identificador único para cada entrenador en la base de datos. Es de tipo "int" (entero) y se genera automáticamente con un incremento automático. Funciona como la clave primaria (PRIMARY KEY) de la tabla, garantizando su unicidad.
* **Nombre:** Almacena el nombre del entrenador. Este campo es de tipo "varchar(45)" (cadena de caracteres de longitud máxima 45) y no puede estar vacío ("NOT NULL"). Contiene la información esencial del nombre del entrenador.
* **Especialidad:** Describe la especialidad o área de enfoque del entrenador. Es un campo opcional que puede ser nulo, ya que se establece como "DEFAULT NULL". Si se completa, es de tipo "varchar(100)" (cadena de caracteres de longitud máxima 100).

Se utiliza para almacenar información sobre los entrenadores disponibles en el gimnasio. El campo "id\_entrenador" sirve como identificador único para cada entrenador. El campo "nombre" guarda el nombre del entrenador, proporcionando información identificativa clave. La columna "especialidad" es opcional y puede contener detalles adicionales sobre la especialidad o área de enfoque del entrenador. En conjunto, esta tabla es esencial para gestionar y organizar la información de los profesionales que brindan servicios de entrenamiento en el gimnasio.

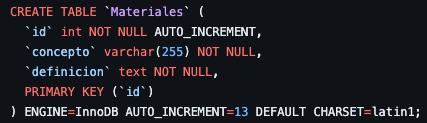
1. **Tabla: InventarioMaquinas.**

****

* **id\_maquina (ID):** Este campo es un identificador único para cada máquina en el inventario del gimnasio. Es de tipo "int" (entero) y se genera automáticamente con un incremento automático. Funciona como la clave primaria (PRIMARY KEY) de la tabla, garantizando su unicidad.
* **Nombre:** Almacena el nombre o descripción de la máquina. Este campo es de tipo "varchar(100)" (cadena de caracteres de longitud máxima 100) y no puede estar vacío ("NOT NULL"). Contiene información esencial para identificar y categorizar la máquina.
* **Fecha\_ingreso:** Indica la fecha en la que la máquina fue ingresada al inventario. Es de tipo "date" y no puede ser nulo ("NOT NULL"). Registra la fecha en la que la máquina se hizo parte del inventario del gimnasio.
* **Fecha\_egreso:** Representa la fecha en la que la máquina fue retirada o dejó de estar en uso. Este campo es opcional y puede ser nulo, ya que se establece como "DEFAULT NULL". Si se completa, es de tipo "date" y proporciona información sobre cuándo la máquina fue retirada del inventario.

Se utiliza para gestionar la información relacionada con las máquinas presentes en el gimnasio. El campo "id\_maquina" actúa como un identificador único para cada máquina. El campo "nombre" describe la máquina de manera detallada, mientras que "fecha\_ingreso" y "fecha\_egreso" registran las fechas importantes relacionadas con la incorporación y posible retiro de la máquina del inventario, respectivamente. Esta tabla es esencial para llevar un seguimiento del estado y la historia de las máquinas utilizadas en el gimnasio.

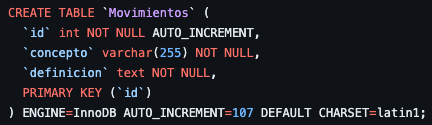
1. **Tabla: Materiales**

****

* **id (ID):** Este campo es un identificador único para cada material en la base de datos. Es una clave primaria (PRIMARY KEY) y su tipo de dato es "int" (entero).
* **Concepto:** Almacena el nombre o concepto del material utilizado en el gimnasio. Su tipo de dato es "varchar(255)" (cadena de caracteres de longitud máxima 255) y no puede estar vacío ("NOT NULL").
* **Definición:** Este campo guarda una descripción detallada o definición del material. Su tipo de dato es "text" y no puede estar vacío ("NOT NULL").

La tabla "Materiales" se utiliza para mantener un registro de los diferentes materiales y equipos utilizados en el gimnasio. Cada material tiene un identificador único en el campo "id" y está asociado con un nombre o concepto en el campo "concepto". La descripción detallada del material se almacena en el campo "definición". Esta tabla es útil para proporcionar información sobre los materiales a los usuarios del gimnasio.

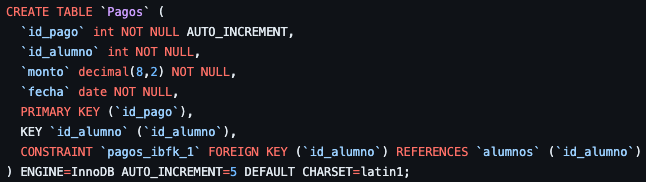
1. **Tabla: Movimientos**

****

* **id (ID):** Este campo es un identificador único para cada movimiento en la base de datos. Es una clave primaria (PRIMARY KEY) y su tipo de dato es "int" (entero).
* **Concepto:** Almacena el nombre o concepto del movimiento utilizado en el gimnasio. Su tipo de dato es "varchar(255)" (cadena de caracteres de longitud máxima 255) y no puede estar vacío ("NOT NULL").
* **Definición:** Este campo guarda una descripción detallada o definición del movimiento. Su tipo de dato es "text" y no puede estar vacío ("NOT NULL").

La tabla "Movimientos" se utiliza para mantener un registro de los diferentes movimientos utilizados en las sesiones de entrenamiento del gimnasio. Cada movimiento tiene un identificador único en el campo "id" y está asociado con un nombre o concepto en el campo "concepto". La descripción detallada del movimiento se almacena en el campo "definición". Esta tabla es útil para proporcionar información sobre los movimientos a los usuarios del gimnasio y para ayudar a los entrenadores a planificar y registrar los ejercicios en las rutinas de entrenamiento.

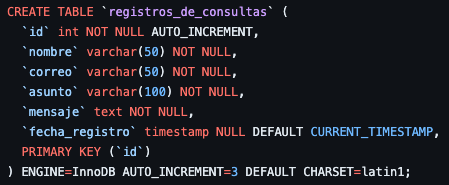
1. **Tabla: Pagos.**

****

* **id\_pago (ID):** Este campo es un identificador único para cada registro de pago en la base de datos. Es de tipo "int" (entero) y se genera automáticamente con un incremento automático. Funciona como la clave primaria (PRIMARY KEY) de la tabla, garantizando su unicidad.
* **id\_alumno:** Almacena el identificador único del alumno asociado al pago. Este campo es de tipo "int" y no puede estar vacío ("NOT NULL"). Además, tiene una clave externa (FOREIGN KEY) que establece una relación con la tabla "alumnos" a través del campo "id\_alumno" en dicha tabla.
* **monto:** Indica el monto del pago realizado. Es de tipo "decimal(8,2)" y no puede ser nulo ("NOT NULL"). Este formato decimal permite precisión en la representación de los montos de pago.
* **fecha:** Representa la fecha en la que se realizó el pago. Es de tipo "date" y no puede ser nulo ("NOT NULL"). Registra la fecha en que se efectuó la transacción financiera.

La tabla "Pagos" se utiliza para rastrear los pagos realizados por los alumnos del gimnasio. El campo "id\_pago" funciona como un identificador único para cada registro de pago. El campo "id\_alumno" está relacionado con la tabla "alumnos" mediante una clave externa (FOREIGN KEY), estableciendo así una conexión entre los pagos y los alumnos. La columna "monto" almacena la cantidad financiera asociada al pago, y "fecha" registra la fecha exacta en que se realizó la transacción. Esta tabla es esencial para llevar un seguimiento detallado de las transacciones financieras dentro del gimnasio y gestionar los pagos de los alumnos de manera efectiva.

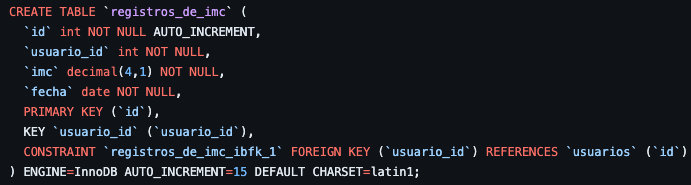
1. **Tabla: registros\_de\_consultas.**

****

* **id (ID):** Este campo es un identificador único para cada registro de consulta en la base de datos. Es una clave primaria (PRIMARY KEY) y su tipo de dato es "int" (entero).
* **nombre:** Almacena el nombre del remitente de la consulta. Su tipo de dato es "varchar(50)" (cadena de caracteres de longitud máxima 50) y no puede estar vacío ("NOT NULL").
* **correo:** Guarda la dirección de correo electrónico del remitente de la consulta. Su tipo de dato es "varchar(50)" y no puede estar vacío ("NOT NULL").
* asunto: Este campo almacena el asunto o tema de la consulta realizada. Su tipo de dato es "varchar(100)" y no puede estar vacío ("NOT NULL").
* **mensaje:** Contiene el texto del mensaje o consulta detallada realizada por el remitente. Su tipo de dato es "text" y no puede estar vacío ("NOT NULL").
* **fecha\_registro:** Este campo registra la fecha y hora en que se creó el registro de consulta. Tiene un valor predeterminado que es la marca de tiempo actual y puede ser nulo ("NULL") si no se especifica.

La tabla "registros\_de\_consultas" se utiliza para mantener un registro de las consultas realizadas por los usuarios o clientes del gimnasio. Cada consulta tiene un identificador único en el campo "id" y está asociada con el nombre del remitente, su dirección de correo electrónico, el asunto de la consulta y el mensaje detallado. Además, se registra la fecha y hora en que se creó el registro de consulta en el campo "fecha\_registro". Esta tabla es útil para gestionar y dar seguimiento a las consultas realizadas por los usuarios del gimnasio.

1. **Tabla: registros\_de\_imc**

****

* **id (ID):** Este campo es un identificador único para cada registro de IMC en la base de datos. Es de tipo "int" (entero) y se genera automáticamente con un incremento automático. Funciona como la clave primaria (PRIMARY KEY) de la tabla, garantizando su unicidad.
* **usuario\_id:** Almacena el identificador único del usuario asociado al registro de IMC. Este campo es de tipo "int" y no puede estar vacío ("NOT NULL"). Además, tiene una clave externa (FOREIGN KEY) que establece una relación con la tabla "usuarios" a través del campo "id" en dicha tabla.
* **imc:** Indica el valor del Índice de Masa Corporal registrado. Es de tipo "decimal(4,1)" y no puede ser nulo ("NOT NULL"). Este formato decimal permite precisión en la representación del IMC.
* **fecha:** Representa la fecha en la que se registró el IMC. Es de tipo "date" y no puede ser nulo ("NOT NULL"). Registra la fecha exacta en que se realizó la medición del Índice de Masa Corporal.

La tabla "Registros de IMC" se utiliza para almacenar información sobre el Índice de Masa Corporal de los usuarios. El campo "id" actúa como un identificador único para cada registro. El campo "usuario\_id" está relacionado con la tabla "usuarios" mediante una clave externa (FOREIGN KEY), estableciendo así una conexión entre los registros de IMC y los usuarios. La columna "imc" almacena el valor numérico del Índice de Masa Corporal registrado, y "fecha" registra la fecha exacta en que se realizó la medición. Esta tabla es esencial para realizar un seguimiento de la salud y el progreso físico de los usuarios en el gimnasio.

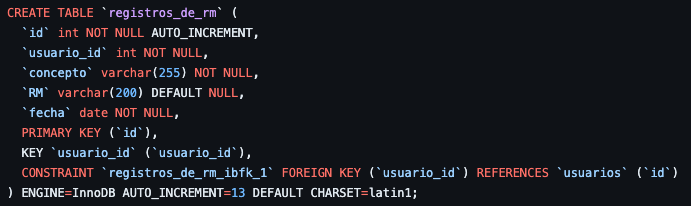
1. **Tabla: registros\_de\_peso.**

****

* **id (ID):** Este campo es un identificador único para cada registro de peso en la base de datos. Es de tipo "int" (entero) y se genera automáticamente con un incremento automático. Funciona como la clave primaria (PRIMARY KEY) de la tabla, garantizando su unicidad.
* **usuario\_id:** Almacena el identificador único del usuario asociado al registro de peso. Este campo es de tipo "int" y no puede estar vacío ("NOT NULL"). Además, tiene una clave externa (FOREIGN KEY) que establece una relación con la tabla "usuarios" a través del campo "id" en dicha tabla.
* **peso:** Indica el valor del peso registrado. Es de tipo "decimal(4,1)" y no puede ser nulo ("NOT NULL"). Este formato decimal permite precisión en la representación del peso.
* **fecha:** Representa la fecha en la que se registró el peso. Es de tipo "date" y no puede ser nulo ("NOT NULL"). Registra la fecha exacta en que se realizó la medición del peso.

La tabla "Registros de Peso" se utiliza para almacenar información sobre los registros de peso de los usuarios. El campo "id" actúa como un identificador único para cada registro. El campo "usuario\_id" está relacionado con la tabla "usuarios" mediante una clave externa (FOREIGN KEY), estableciendo así una conexión entre los registros de peso y los usuarios. La columna "peso" almacena el valor numérico del peso registrado, y "fecha" registra la fecha exacta en que se realizó la medición. Esta tabla es esencial para realizar un seguimiento de la salud y el progreso físico de los usuarios en el gimnasio, especialmente en términos de cambios en su peso corporal.

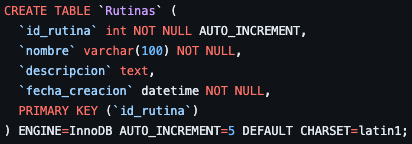
1. **Tabla: registros\_de\_rm.**

****

* **id (ID):** Este campo es un identificador único para cada registro de repeticiones máximas en la base de datos. Es de tipo "int" (entero) y se genera automáticamente con un incremento automático. Funciona como la clave primaria (PRIMARY KEY) de la tabla, garantizando su unicidad.
* **usuario\_id:** Almacena el identificador único del usuario asociado al registro de repeticiones máximas. Este campo es de tipo "int" y no puede estar vacío ("NOT NULL"). Además, tiene una clave externa (FOREIGN KEY) que establece una relación con la tabla "usuarios" a través del campo "id" en dicha tabla.
* **concepto:** Describe el concepto o ejercicio asociado al registro de repeticiones máximas. Es de tipo "varchar(255)" (cadena de caracteres de longitud máxima 255) y no puede estar vacío ("NOT NULL"). Proporciona información sobre el ejercicio específico para el cual se registraron las repeticiones máximas.
* **RM:** Indica el valor o resultado de las repeticiones máximas registradas. Es de tipo "varchar(200)" y puede ser nulo ("DEFAULT NULL"). Permite registrar el resultado de las repeticiones máximas de manera flexible, ya que puede contener información numérica o textual.
* **fecha:** Representa la fecha en la que se registraron las repeticiones máximas. Es de tipo "date" y no puede ser nulo ("NOT NULL"). Registra la fecha exacta en la que se realizaron las mediciones.

La tabla "Registros de RM" se utiliza para almacenar información sobre las repeticiones máximas registradas por los usuarios. El campo "id" actúa como un identificador único para cada registro. El campo "usuario\_id" está relacionado con la tabla "usuarios" mediante una clave externa (FOREIGN KEY), estableciendo así una conexión entre los registros de repeticiones máximas y los usuarios. Las columnas "concepto" y "RM" proporcionan detalles sobre el ejercicio específico y el resultado de las repeticiones máximas, respectivamente. La columna "fecha" registra la fecha exacta en que se realizaron las mediciones. Esta tabla es esencial para evaluar el progreso y el rendimiento de los usuarios en sus entrenamientos en el gimnasio.

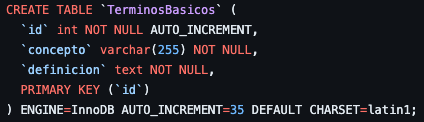
1. **Tabla: Rutinas.**

****

* **id\_rutina (ID):** Este campo es un identificador único para cada rutina en la base de datos. Es de tipo "int" (entero) y se genera automáticamente con un incremento automático. Funciona como la clave primaria (PRIMARY KEY) de la tabla, garantizando su unicidad.
* **nombre:** Almacena el nombre de la rutina. Este campo es de tipo "varchar(100)" (cadena de caracteres de longitud máxima 100) y no puede estar vacío ("NOT NULL"). Proporciona un nombre identificativo para la rutina.
* **descripción:** Contiene una descripción detallada de la rutina. Es de tipo "text", permitiendo una longitud de texto más extensa. Este campo puede ser nulo, ya que se establece como "DEFAULT NULL". Proporciona información adicional sobre la rutina, como los ejercicios incluidos, el enfoque de entrenamiento, etc.
* **fecha\_creacion:** Indica la fecha en la que se creó la rutina. Es de tipo "datetime" y no puede ser nulo ("NOT NULL"). Registra la fecha y hora exactas en que se creó la rutina.

La tabla "Rutinas" se utiliza para almacenar información sobre las rutinas de entrenamiento en el gimnasio. El campo "id\_rutina" actúa como un identificador único para cada rutina. El campo "nombre" proporciona un nombre descriptivo para la rutina, mientras que "descripcion" permite incluir detalles adicionales sobre la rutina. La columna "fecha\_creacion" registra la fecha y hora en que se creó la rutina. Esta tabla es esencial para organizar y gestionar las rutinas de entrenamiento ofrecidas en el gimnasio, proporcionando información clave para los usuarios y entrenadores.

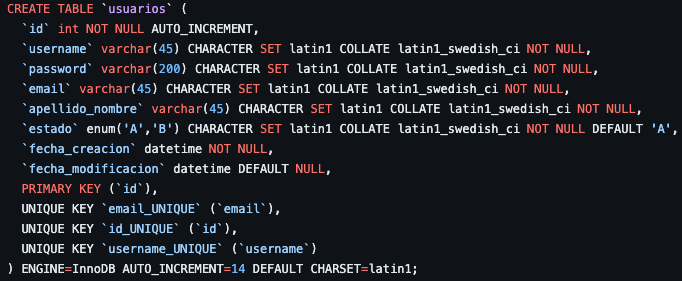
1. **Tabla: TerminosBasicos**

****

* **id (ID):** Este campo es un identificador único para cada término básico en la base de datos. Es de tipo "int" (entero) y se genera automáticamente con un incremento automático. Funciona como la clave primaria (PRIMARY KEY) de la tabla, garantizando su unicidad.
* **Concepto:** Almacena el concepto o término básico. Este campo es de tipo "varchar(255)" (cadena de caracteres de longitud máxima 255) y no puede estar vacío ("NOT NULL"). Proporciona un nombre identificativo para el término básico.
* **Definición**: Contiene la definición detallada del término básico. Es de tipo "text" y no puede ser nulo ("NOT NULL"). Proporciona una descripción extensa y clara del significado del término básico.

La tabla "Términos Básicos" se utiliza para almacenar información sobre los conceptos básicos relacionados con el gimnasio y el entrenamiento. El campo "id" actúa como un identificador único para cada término básico. La columna "concepto" almacena el nombre del término básico, mientras que "definicion" contiene una descripción detallada de su significado. Esta tabla es útil para proporcionar a los usuarios y entrenadores información educativa y clarificadora sobre los términos utilizados en el contexto del gimnasio.

1. **Tabla: usuarios.**

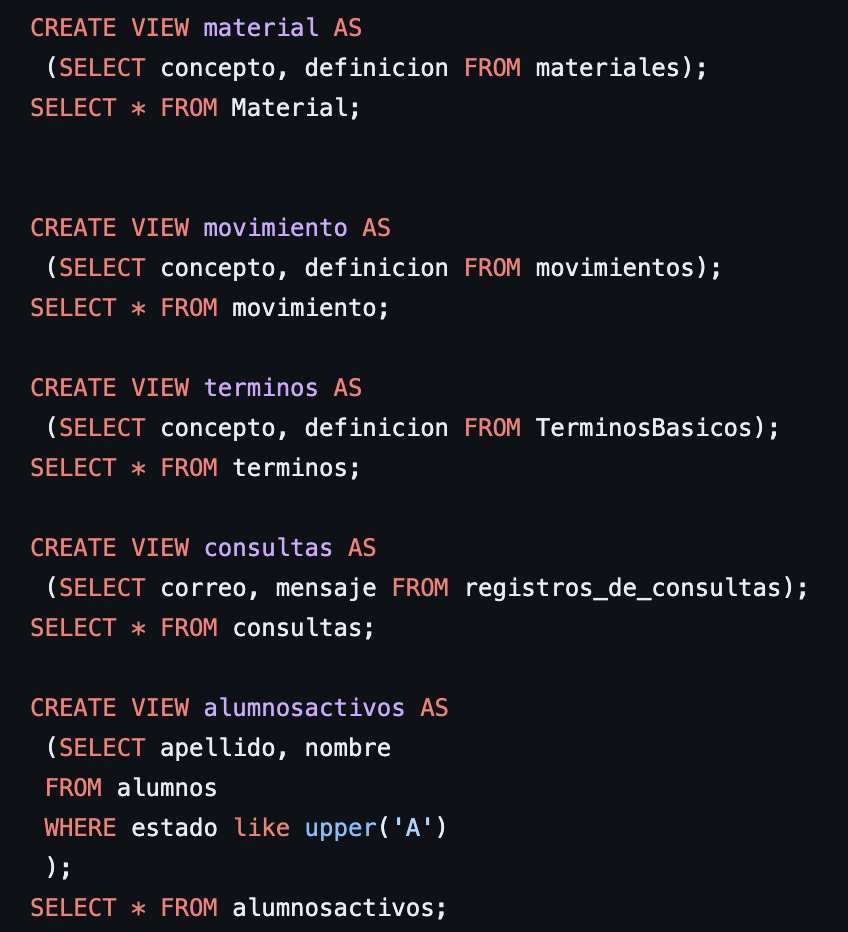
****

* **id (ID):** Este campo es un identificador único para cada usuario en la base de datos. Es de tipo "int" (entero) y se genera automáticamente con un incremento automático. Funciona como la clave primaria (PRIMARY KEY) de la tabla, garantizando su unicidad.
* **username:** Almacena el nombre de usuario único de cada usuario. Este campo es de tipo "varchar(45)" (cadena de caracteres de longitud máxima 45) y no puede estar vacío ("NOT NULL"). Proporciona un identificador único para el inicio de sesión.
* **password:** Contiene la contraseña del usuario. Este campo es de tipo "varchar(200)" y no puede estar vacío ("NOT NULL"). Almacena la contraseña de manera segura, y su formato es adecuado para contraseñas cifradas.
* **email:** Almacena la dirección de correo electrónico única de cada usuario. Este campo es de tipo "varchar(45)" y no puede estar vacío ("NOT NULL"). Proporciona una forma adicional de identificación única.
* **apellido\_nombre:** Contiene el apellido y el nombre del usuario. Este campo es de tipo "varchar(45)" y no puede estar vacío ("NOT NULL"). Permite almacenar el nombre completo del usuario.
* **estado:** Indica el estado del usuario, ya sea activo ('A') o inactivo ('B'). Es de tipo "enum" y no puede estar vacío ("NOT NULL"). Se establece por defecto en 'A' (activo) cuando se crea un nuevo usuario.
* **fecha\_creacion:** Indica la fecha y hora en la que se creó el usuario. Es de tipo "datetime" y no puede ser nulo ("NOT NULL"). Registra la fecha y hora exactas de la creación del usuario.
* **fecha\_modificacion:** Representa la fecha y hora de la última modificación del usuario. Es de tipo "datetime" y puede ser nulo, ya que se establece como "DEFAULT NULL". Registra la última vez que se realizaron cambios en la información del usuario.

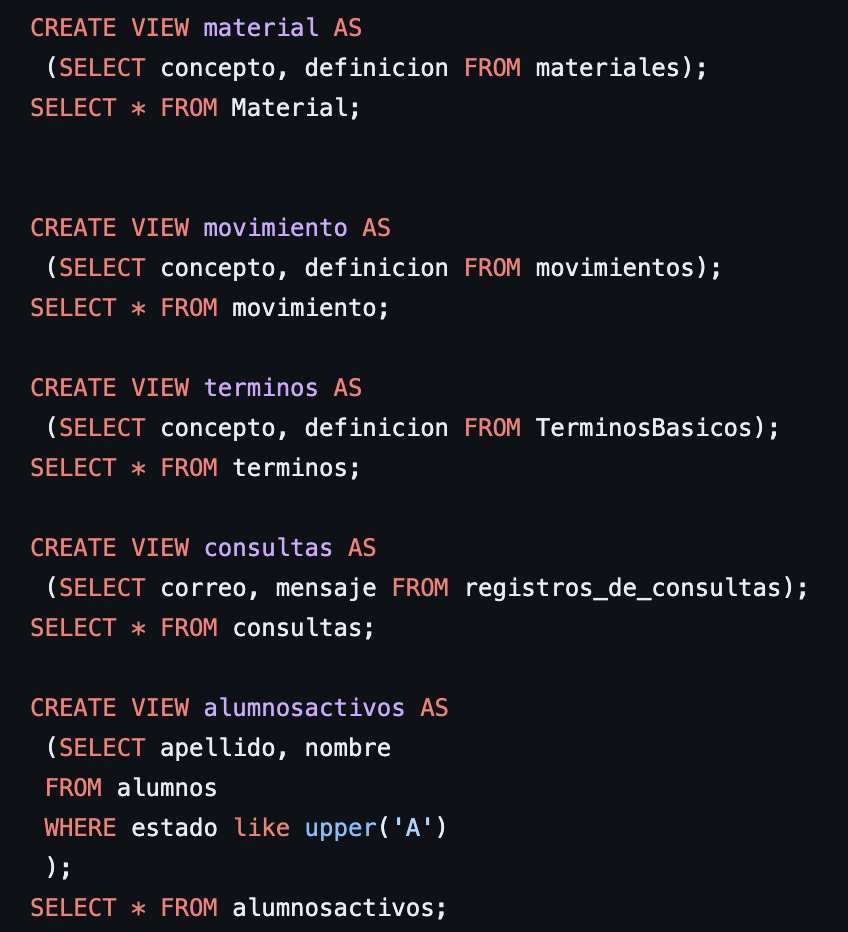
La tabla "Usuarios" se utiliza para almacenar información sobre los usuarios del sistema. El campo "id" actúa como un identificador único para cada usuario. Los campos "username" y "email" proporcionan identificadores únicos para el inicio de sesión. El campo "estado" permite gestionar el estado activo o inactivo de un usuario. Las columnas "fecha\_creacion" y "fecha\_modificacion" registran la información temporal asociada con la creación y la última modificación del usuario, respectivamente. Esta tabla es fundamental para la gestión de usuarios y la seguridad del sistema.

# **LISTADO DE VISTAS**

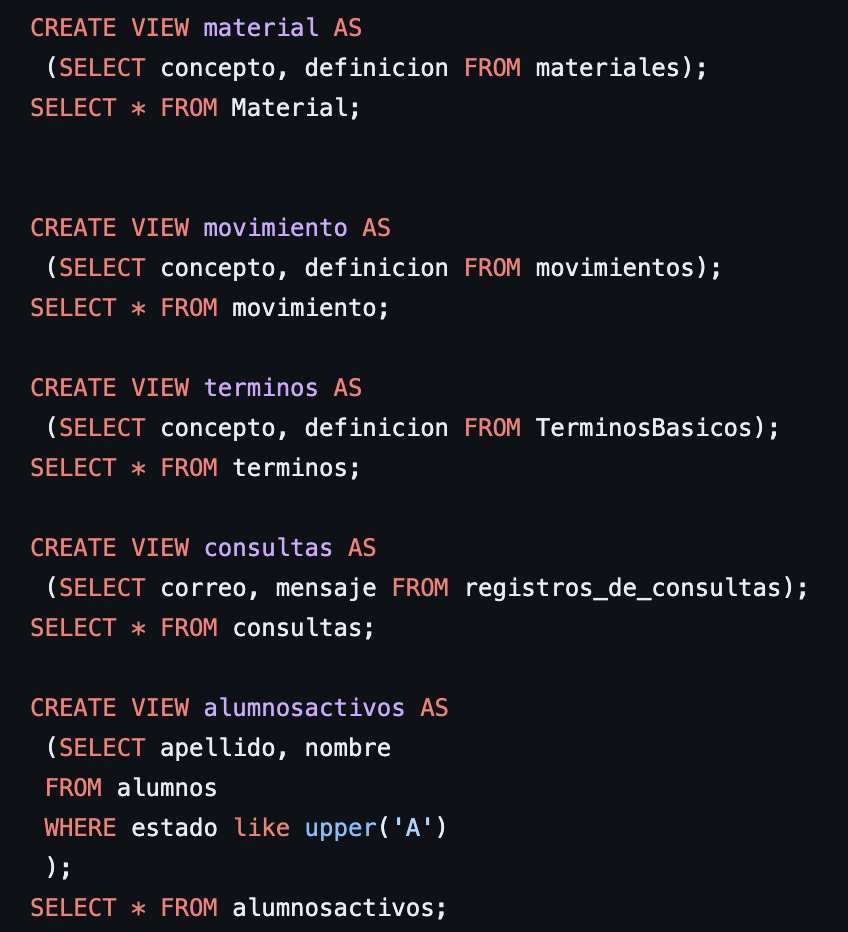
* Material:
  + Descripción: Vista que muestra los conceptos y definiciones de la tabla de materiales.
  + Objetivo: Proporcionar una visión simplificada de los materiales y sus definiciones.
  + Tablas: materiales



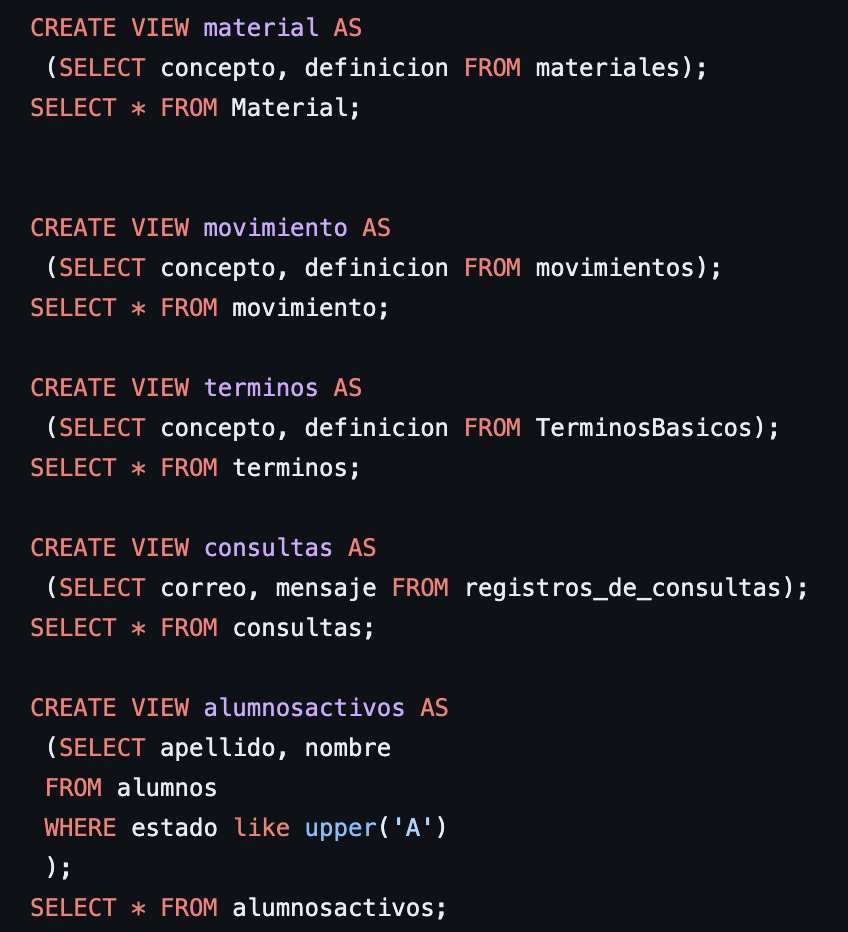
* Movimiento:
  + Descripción: Vista que presenta los conceptos y definiciones de la tabla de movimientos.
  + Objetivo: Ofrecer una perspectiva resumida de los movimientos y sus definiciones.
  + Tablas: movimientos



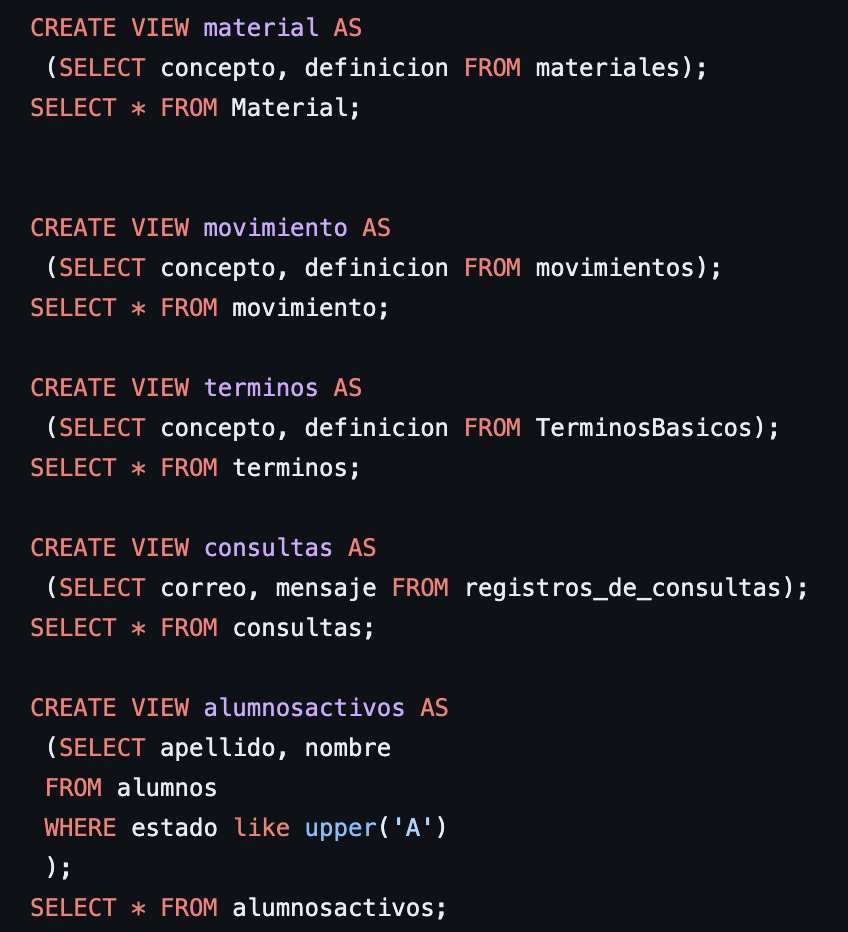
* Terminos:
  + Descripción: Vista que muestra los conceptos y definiciones de la tabla de términos básicos.
  + Objetivo: Proporcionar una visión simplificada de los términos básicos y sus definiciones.
  + Tablas: TerminosBasicos



* Consultas:
  + Descripción: Vista que extrae correos y mensajes de la tabla de registros de consultas.
  + Objetivo: Presentar una vista de las consultas realizadas junto con la información correspondiente.
  + Tablas: registros\_de\_consultas



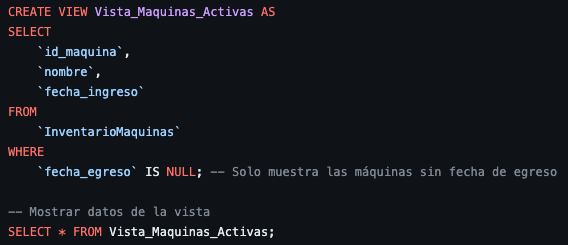
* Alumnos Activos:
  + Descripción: Vista que muestra apellidos y nombres de alumnos con estado activo.
  + Objetivo: Mostrar una lista simplificada de los alumnos activos.
  + Tablas: alumnos



* Vista\_Alumnos\_Ultimo\_Pago:
  + Descripción: se ha creado con el propósito de proporcionar una lista simplificada de los alumnos que tiene su abono al día o no.
  + Objetivo: Su principal objetivo es ofrecer una visión rápida y concisa de los alumnos que se encuentran actualmente activos en el entorno del gimnasio. Al simplificar la información a los campos esenciales de apellidos y nombres, la vista facilita una rápida identificación de los alumnos activos, optimizando la gestión y visualización de datos relevantes.
  + Tablas: relaciona la tabla “Alumnos” con la tabla de “Pagos”.



* Vista\_Maquinas\_Activas
  + Descripción: ha sido diseñada para presentar una lista simplificada de las máquinas activas en el inventario del gimnasio.
  + Objetivo: su objetivo principal es ofrecer una visión rápida y clara de las máquinas que aún están en uso, excluyendo aquellas que han sido retiradas. Al mostrar únicamente el ID de la máquina, su nombre y la fecha de ingreso, la vista facilita la identificación de las máquinas que contribuyen actualmente a la operatividad del gimnasio.
  + Tablas: InventarioMaquinas



# **LISTADO DE FUNCIONES**

* CountMovimientos
  + Descripción: Esta función cuenta el número total de registros en la tabla Movimientos.
  + Objetivo: Proporcionar la cantidad de movimientos registrados en la base de datos.
  + Tablas involucradas: Movimientos

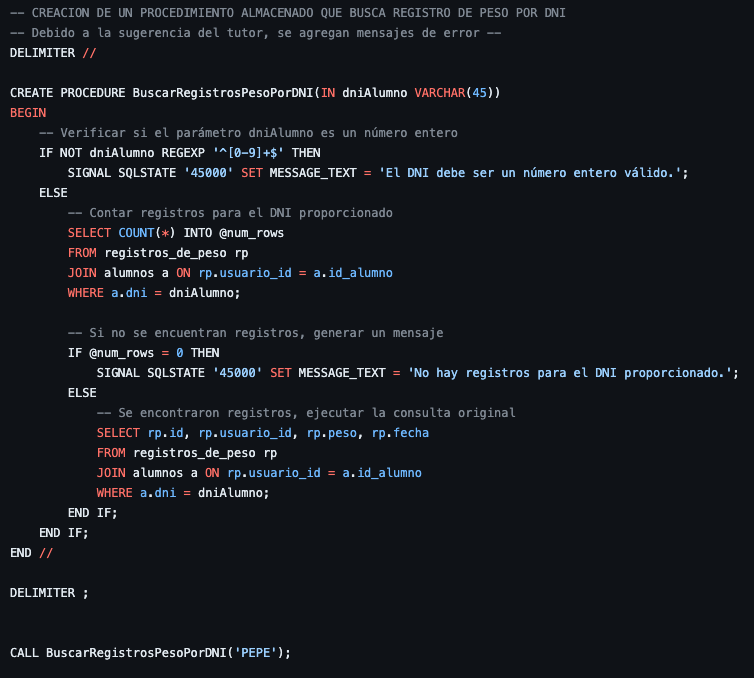


* ObtenerUltimoPesoPorDNI
  + Descripción: Esta función obtiene el último peso registrado para un alumno específico, identificado por su número de DNI.
  + Objetivo: Proporcionar el peso más reciente registrado para un alumno dado.
  + Tablas involucradas: registros\_de\_peso, alumnos.



# **LISTADO DE STORED PROCEDURES**

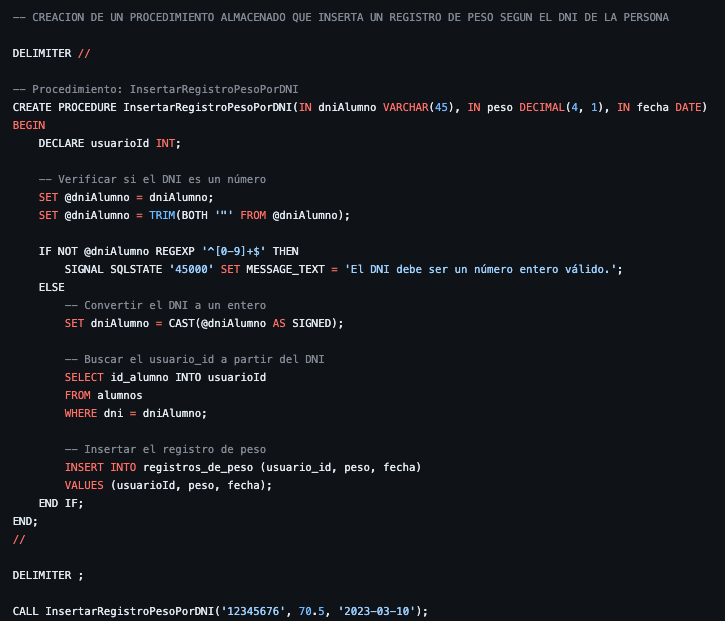
* BuscarRegistrosPesoPorDNI
  + Descripción: Este procedimiento busca registros de peso para un alumno específico identificado por su número de DNI.
  + Objetivo: Obtener información detallada de los registros de peso para un alumno dado.
  + Tablas involucradas: registros\_de\_peso, alumnos.



* BuscarRegistrosPesoPorDNIConOrden
  + Descripción: Este procedimiento busca registros de peso para un alumno específico identificado por su número de DNI y permite ordenar los resultados de manera ascendente o descendente.
  + Objetivo: Obtener información detallada de los registros de peso para un alumno dado con opciones de ordenamiento.
  + Tablas involucradas: registros\_de\_peso, alumnos



* InsertarRegistroPesoPorDNI
  + Descripción: Este procedimiento inserta un nuevo registro de peso para un alumno identificado por su número de DNI.
  + Objetivo: Agregar nuevos registros de peso a la base de datos.
  + Tablas involucradas: registros\_de\_peso, alumnos



# **LISTADO DE TRIGGERS.**

* Triggers para Auditoría en la Tabla "Alumnos"

Se han implementado dos triggers en la tabla "Alumnos" con el propósito de auditar cambios y acciones realizadas en los registros.

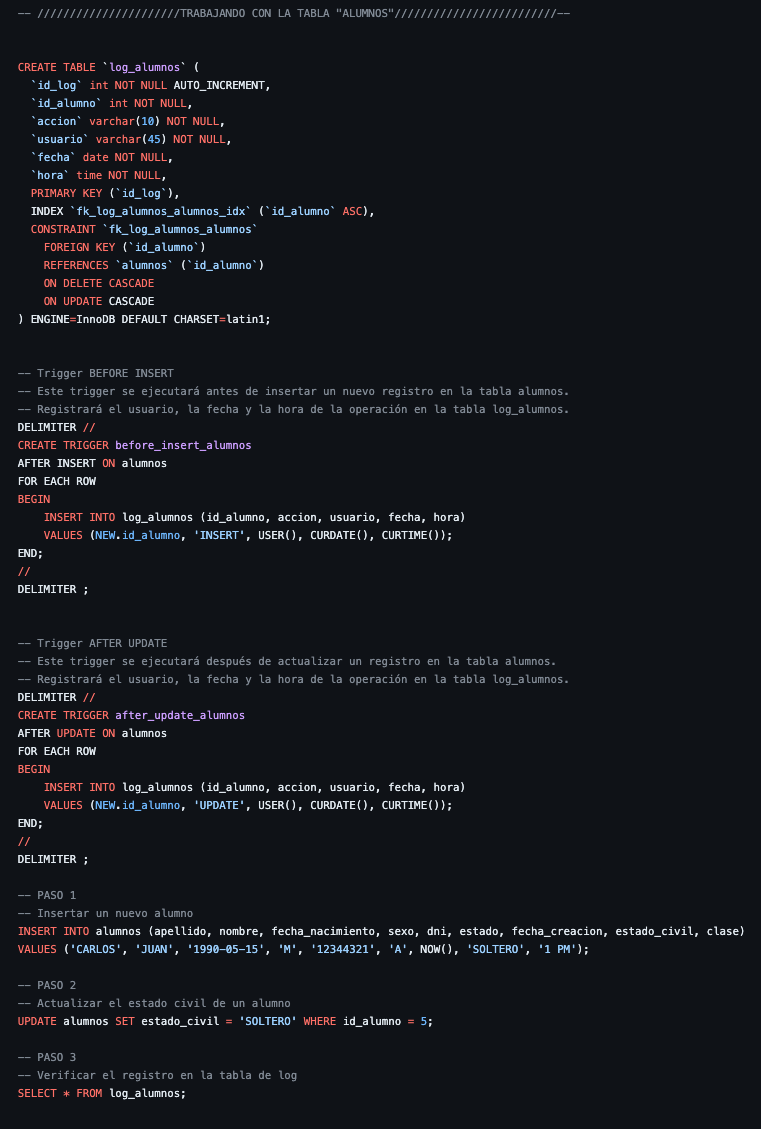
**Trigger BEFORE INSERT (before\_insert\_alumnos):** Este trigger se activa antes de insertar un nuevo registro en la tabla "Alumnos". Su función principal es registrar automáticamente en la tabla "log\_alumnos" los detalles de la acción, incluyendo el ID del alumno afectado, la acción (INSERT), el usuario que realiza la acción, la fecha y la hora exacta en que se ejecuta la operación.

**Trigger AFTER UPDATE (after\_update\_alumnos):** Este trigger se dispara después de que se realiza una actualización en un registro de la tabla "Alumnos". Similar al trigger BEFORE INSERT, registra en "log\_alumnos" información esencial, como el ID del alumno afectado, la acción (UPDATE), el usuario responsable, la fecha y la hora de la operación.

**Tabla "log\_alumnos":** La tabla "log\_alumnos" actúa como un registro de auditoría, almacenando información relevante sobre las acciones realizadas en la tabla "Alumnos". Cada registro en esta tabla incluye un ID único de log, el ID del alumno afectado, la acción realizada (INSERT o UPDATE), el usuario responsable, la fecha y la hora exacta de la acción.

**Operaciones de Ejemplo (PASO 1, 2, 3):**

El script incluye ejemplos prácticos que ilustran el funcionamiento de los triggers. En el "PASO 1", se inserta un nuevo alumno; en el "PASO 2", se actualiza el estado civil de un alumno; y finalmente, en el "PASO 3", se verifica el registro en la tabla de log ("log\_alumnos") para visualizar las acciones registradas. Estos ejemplos ayudan a comprender cómo los triggers contribuyen a mantener un registro histórico de las operaciones en la tabla "Alumnos".



* Triggers para Auditoría en las Tablas "Registros\_de\_Peso" y "Registros\_de\_IMC".

Se han implementado triggers en las tablas "Registros\_de\_Peso" y "Registros\_de\_IMC" con el objetivo de auditar las operaciones de inserción y actualización en ambas tablas.

**Triggers para la Tabla "Registros\_de\_IMC"**

**Trigger BEFORE INSERT (before\_insert\_imc):** Este trigger se ejecuta antes de insertar un nuevo registro en la tabla "Registros\_de\_IMC". Su función es registrar automáticamente en la tabla "log\_registros" los detalles de la operación, incluyendo la tabla afectada ("registros\_de\_imc"), la acción (BEFORE INSERT), el usuario que realiza la acción, la fecha y la hora exacta de la operación.

**Trigger AFTER UPDATE (after\_update\_imc):** Este trigger se activa después de actualizar un registro en la tabla "Registros\_de\_IMC". Al igual que el trigger BEFORE INSERT, registra en "log\_registros" información esencial como la tabla afectada, la acción (AFTER UPDATE), el usuario responsable, la fecha y la hora de la operación.

**Triggers para la Tabla "Registros\_de\_Peso"**

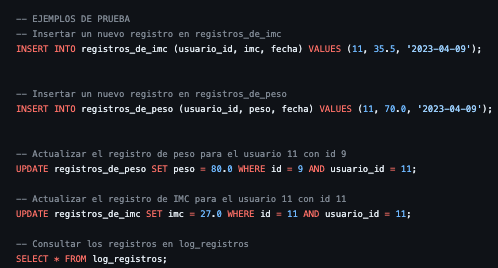
**Trigger BEFORE INSERT (before\_insert\_peso):** Similar al trigger antes mencionado para la tabla de IMC, este trigger se ejecuta antes de insertar un nuevo registro en la tabla "Registros\_de\_Peso". Registra en "log\_registros" los detalles de la operación, incluyendo la tabla afectada ("registros\_de\_peso"), la acción (BEFORE INSERT), el usuario que realiza la acción, la fecha y la hora exacta de la operación.

**Trigger AFTER UPDATE (after\_update\_peso):** Este trigger se dispara después de actualizar un registro en la tabla "Registros\_de\_Peso". Al igual que los triggers anteriores, registra en "log\_registros" información esencial sobre la tabla afectada, la acción (AFTER UPDATE), el usuario responsable, la fecha y la hora de la operación.

**Tabla "log\_registros":** La tabla "log\_registros" se utiliza como un registro de auditoría centralizado, almacenando información relevante sobre las acciones realizadas en las tablas "Registros\_de\_IMC" y "Registros\_de\_Peso". Cada registro en esta tabla incluye un ID único de log, la tabla afectada, la operación realizada, el usuario responsable, la fecha y la hora exacta de la acción.

**Operaciones de Ejemplo:** El script incluye ejemplos prácticos que ilustran el funcionamiento de los triggers. Se insertan nuevos registros en ambas tablas y se actualizan registros existentes. Luego, se verifica el registro en la tabla de log ("log\_registros") para visualizar las acciones registradas en ambas tablas. Estos ejemplos ayudan a comprender cómo los triggers contribuyen a mantener un registro histórico de las operaciones en las tablas "Registros\_de\_IMC" y "Registros\_de\_Peso".





# **USUARIOS MySQL Y PERMISOS.**

* Usuario 'usuario\_lectura'@'localhost':

**Identificación: '**usuario\_lectura' con acceso desde 'localhost'.

**Contraseña:** 'coderhouse'.

**Permisos:** SELECT en todas las tablas ('.'). Sin permisos de eliminación.

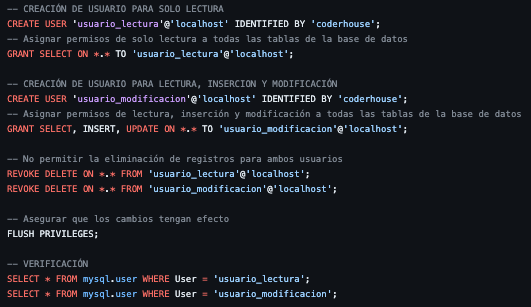
* Usuario 'usuario\_modificacion'@'localhost':

**Identificación:** 'usuario\_modificacion' con acceso desde 'localhost'.

**Contraseña:** 'coderhouse'.

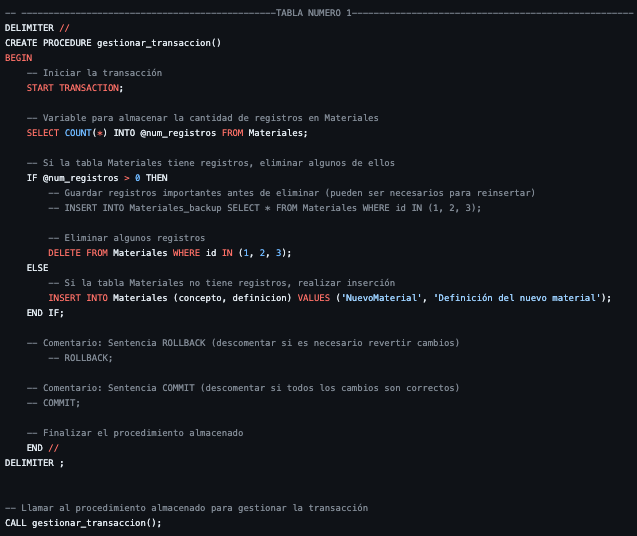
**Permisos:** SELECT, INSERT y UPDATE en todas las tablas ('.'). Sin permisos de eliminación.

Estos usuarios brindan acceso controlado para lectura o modificación, sin posibilidad de eliminar registros. Se ha ejecutado FLUSH PRIVILEGES para aplicar cambios y verificado la configuración.



# **IMPLEMENTACIÓN DE SENTENCIAS DEL SUBLENGUAJE TCL.**

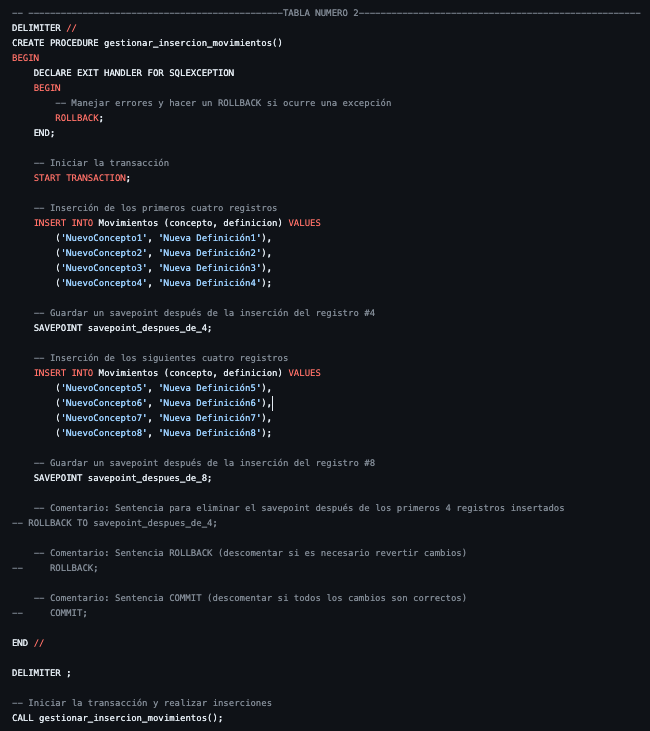
1. El procedimiento almacenado **gestionar\_transaccion** se utiliza para manejar transacciones en la tabla Materiales. En su lógica, verifica la existencia de registros en dicha tabla y realiza acciones específicas dependiendo de si hay o no registros presentes. En el caso de registros existentes, elimina algunos de ellos, y si no hay registros, realiza una inserción. Este procedimiento brinda un control transaccional, permitiendo realizar operaciones condicionales en la tabla Materiales según las necesidades del sistema, todo dentro de una única transacción.



1. El procedimiento denominado **gestionar\_insercion\_movimientos**, se encarga de manejar transacciones relacionadas con la tabla Movimientos. La lógica del procedimiento consiste en realizar dos conjuntos de inserciones de registros en la tabla. Después de la inserción de los primeros cuatro registros, se guarda un punto de guardado (savepoint) denominado savepoint\_despues\_de\_4. Luego, se realiza la inserción de los siguientes cuatro registros.

Incluye comentarios que indican cómo manejar situaciones específicas. Por ejemplo, se proporciona una sentencia para eliminar el savepoint después de la inserción de los primeros cuatro registros. Asimismo, se encuentran comentarios que indican cómo realizar un ROLLBACK o un COMMIT según sea necesario.

En caso de que se produzca una excepción (capturada mediante un manejador de excepciones), se ejecutará un ROLLBACK para deshacer cualquier cambio realizado en la transacción. Este procedimiento proporciona flexibilidad en la gestión de inserciones, permitiendo revertir cambios parciales en la base de datos si es necesario.



# **INFORMES GENERADOS EN BASE A LA INFORMACIÓN DE LA BASE.**

La generación de informes basados en la información almacenada en una base de datos desempeña un papel crucial en el análisis y la toma de decisiones en cualquier entorno empresarial. En el contexto de nuestro sistema de gestión integral para entrenamientos y salud, las tablas proporcionan un valioso repositorio de datos relacionados con alumnos, asistencias, pagos, ejercicios, y otros aspectos esenciales. La creación de informes a partir de esta información no solo proporciona una visión integral de la operación del sistema, sino que también se convierte en una herramienta fundamental para la optimización de servicios, la toma de decisiones informadas y la mejora continua.

A continuación, se presentan descripciones detalladas de las tablas clave en nuestra base de datos y cómo los informes generados a partir de estas pueden traducirse en valiosas acciones estratégicas. Cada tabla se examina en términos de los informes que podría ofrecer y cómo estos informes podrían aplicarse en la toma de decisiones para maximizar la eficiencia, satisfacción del usuario y rendimiento general del sistema.

* **Usuarios:** La tabla de usuarios contiene información esencial sobre los usuarios del sistema, incluyendo su nombre de usuario, contraseña, dirección de correo electrónico, nombre completo, estado (activo o inactivo), y fechas de creación y modificación. A partir de esta tabla, se podrían generar informes para auditar el acceso al sistema, rastrear cambios en las credenciales, y monitorear la actividad de los usuarios. Estos informes serían valiosos para la toma de decisiones relacionadas con la seguridad y la gestión de usuarios.

Cabe destacar que la intención de esta base de datos es ser aplicada para diseñar una página web, por lo cual es sumamente importante contar con una tabla para la gestión de los usuario de dicha página.

* **Alumnos:** La tabla de alumnos almacena datos clave sobre los estudiantes, como su nombre, fecha de nacimiento, género, DNI, estado civil, clase, y fechas de creación y modificación. A partir de esta información, podrían generarse informes para evaluar la demografía estudiantil, el rendimiento, y la participación en actividades. Estos informes serían útiles para la toma de decisiones relacionadas con la planificación del gimnasio, la asignación de recursos y el seguimiento del progreso de los alumnos.
* **Asistencias:** La tabla de asistencias registra la asistencia de los alumnos a diferentes cursos en diversas fechas. Los informes generados son necesarios para brindar información sobre la asiduidad de los alumnos, identificar patrones de asistencia, y evaluar la efectividad de los cursos. Estos informes serían valiosos para la toma de decisiones relacionadas con la mejora de la calidad del servicio.
* **Pagos:** La tabla de pagos contiene detalles sobre los pagos realizados por los alumnos, incluyendo el monto, la fecha y el alumno asociado. Los informes generados podrían ofrecer una visión financiera, permitiendo un seguimiento detallado de los ingresos, la identificación de tendencias de pago, y la gestión efectiva de los recursos financieros. Estos informes serían fundamentales para la toma de decisiones relacionadas con la planificación presupuestaria, la evaluación de la salud financiera y la toma de decisiones estratégicas.
* **Registros de IMC, Peso y RM:** Estas tablas registran información sobre índices de masa corporal (IMC), de peso y registros de pesos máximos levantados por los usuarios. Los informes generados podrían ofrecer datos sobre la salud y el progreso físico de los usuarios, siendo útiles para la toma de decisiones en el ámbito de la salud y el bienestar como así también con la planificación de rutinas de ejercicios.
* **Rutinas:** La tabla de rutinas almacena información sobre las rutinas de ejercicio, incluyendo nombre, descripción y fecha de creación. Estos informes serían esenciales para la toma de decisiones relacionadas con la planificación de programas de entrenamiento, la adaptación de rutinas según las preferencias de los usuarios y la mejora continua de los servicios ofrecidos.
* **Curso:** La tabla de cursos almacena información sobre los cursos ofrecidos. Los informes generados podrían proporcionar información sobre la demanda de cursos, teniendo en cuenta que horarios son los más concurridos y cuáles no, permitiendo potenciarlos a esos. Estos informes serían esenciales para el manejo de los cupos, poniendo límites en las inscripciones para que no se supere la capacidad permitida.
* **Entrenadores**: La tabla de entrenadores almacena detalles sobre los entrenadores, incluyendo nombre y especialidad.
* **Inventario de Máquinas:** La tabla de inventario de máquinas contiene información sobre las máquinas disponibles, incluyendo su nombre y fechas de ingreso y egreso. Los informes generados podrían ser útiles para manejar datos relevantes como la vida útil y la necesidad de mantenimiento. Estos informes serían esenciales para la toma de decisiones relacionadas con la gestión de inventario, la planificación de mantenimiento y la optimización de recursos.
* **Materiales, Movimientos y Términos Básicos:** estas 3 tablas desempeñan un papel crucial al proporcionar un registro detallado de diversos conceptos. Estos conceptos ofrecen a los usuarios valiosos conocimientos que pueden aplicar de manera efectiva durante la ejecución de rutinas. Esta información mejorada no solo permite una comprensión más profunda de las rutinas, sino que también contribuye a su ejecución eficiente. Además, estas tablas facilitan la labor de los entrenadores al brindarles recursos detallados y claros para explicar de manera efectiva las distintas rutinas a los usuarios. La información contenida en estas tablas enriquece la experiencia de los usuarios, fomentando un enfoque más informado y preciso hacia sus entrenamientos
* **Registros de Consultas:** La tabla de registros\_de\_consultas almacena información sobre las consultas realizadas, incluyendo el nombre, correo electrónico, asunto, mensaje y fecha de registro. Los informes generados podrían ofrecer datos sobre las consultas más frecuentes, el tiempo de respuesta y la satisfacción del usuario. Estos informes serían valiosos para la toma de decisiones relacionadas con la mejora del servicio al cliente, la optimización de respuestas y la identificación de áreas de interés. Además, con el pasar del tiempo, servirá analizar dichas consultas para poder realizar un listado de consultas “frecuentes” que ya se encuentren en una sección de la página para evitar al usuario tener que realizar la consulta y esperar por su respuesta, optimizando la atención al cliente.

# **HERRAMIENTAS Y TECNOLOGÍAS USADAS.**

La implementación y desarrollo de nuestro sistema de gestión integral para entrenamientos y salud se llevó a cabo empleando un conjunto robusto de herramientas y tecnologías modernas que garantizan un rendimiento eficiente y una experiencia de usuario óptima. Entre las tecnologías utilizadas se incluyen:

**1. Base de Datos MySQL:**

La base de datos subyacente está construida sobre MySQL, un sistema de gestión de bases de datos relacional ampliamente reconocido por su confiabilidad y escalabilidad. MySQL proporciona la estructura sólida necesaria para almacenar y gestionar datos críticos relacionados con alumnos, pagos, registros de salud, y otros aspectos del sistema.

**2. Lenguaje de Consulta SQL:**

La implementación de consultas SQL facilita la extracción eficiente de datos específicos de la base de datos. Las consultas SQL se utilizan para realizar operaciones como la recuperación de información detallada sobre alumnos, generación de informes, y seguimiento de registros de pagos, contribuyendo así a la toma de decisiones informada.

**3. Procedimientos Almacenados:**

Se emplean procedimientos almacenados para encapsular lógica de negocio compleja directamente en la base de datos. Esto mejora la eficiencia y seguridad del sistema al reducir el tráfico de red y permitir la ejecución de operaciones complejas de manera más eficiente.

**4. Triggers y Vistas:**

El uso de triggers y vistas proporciona capacidades adicionales para controlar y manipular datos en tiempo real. Los triggers se aplican para rastrear cambios en tablas específicas, como la creación de registros de log. Las vistas, por otro lado, se utilizan para presentar datos de manera simplificada, como en informes específicos. Esta elección estratégica de utilizar GitHub ha fortalecido la calidad y la coherencia de todo el proceso de desarrollo, contribuyendo de manera significativa al éxito general del proyecto.

**5. GitHub:**

GitHub ha sido la plataforma elegida para organizar y presentar el proyecto en su totalidad. Esta plataforma ha demostrado ser un recurso invaluable, proporcionando un entorno colaborativo que ha facilitado la coordinación y el seguimiento de las diversas etapas del proyecto. A través de GitHub, he estructurado el repositorio de manera organizada, proporcionando un README detallado que sirve como guía para comprender la arquitectura y funcionamiento del proyecto.

**6. MySQL Workbench:**

MySQL Workbench es una herramienta gráfica que se utiliza para el diseño visual de la base de datos, modelado de datos y administración del sistema MySQL. Proporciona una interfaz fácil de usar para la creación y modificación de esquemas de bases de datos.

7. Documentos de Google:

Además de las herramientas y tecnologías mencionadas anteriormente, es importante destacar que las dos pre-entregas y el documento final del proyecto fueron elaborados utilizando documentos de Google.

La combinación de estas herramientas y tecnologías contribuye a la creación de un sistema sólido, eficiente y fácilmente escalable, capaz de satisfacer las necesidades de gestión integral para entrenamientos y salud.

# **FUTURAS LÍNEAS**.

Con el firme propósito de dar continuidad al camino iniciado en Coderhouse, mi objetivo primordial es optimizar la base de datos que he desarrollado hasta ahora. Pretendo aplicar esta mejora de manera integral al proceso de creación de una página web que he venido gestando durante el último año, pero que, lamentablemente, ha sido postergada debido a limitaciones de tiempo. La visión que guía este proyecto es lograr que la administración completa del gimnasio se realice de forma eficiente a través de la plataforma web. Asimismo, busco ofrecer a los usuarios del gimnasio una experiencia más fluida y enriquecedora, permitiéndoles gestionar sus actividades y explorar los servicios del gimnasio de manera cómoda y eficaz directamente desde la página.