

SISTEMA DE GESTIÓN PARA ACADEMIA DE FÚTBOL

PREPARADO PARA:
CoderHouse

PREPARADO POR:
Bianca D'Amelio

19 DE MAYO DE 2025

CONTENIDO

| | |
|--|----|
| INTRODUCCIÓN | 3 |
| OBJETIVO | 3 |
| SITUACIÓN PROBLEMÁTICA | 3 |
| USUARIO FINAL Y NIVEL DE APLICACIÓN DEL ANÁLISIS | 3 |
| DESCRIPCIÓN DE LAS TABLAS DE LA BASE DE DATOS | 3 |
| DIAGRAMA DE ENTIDAD-RELACIÓN | 7 |
| VISTAS DE LA BASE DE DATOS | 8 |
| DESCRIPCIÓN DE LAS FUNCIONES DE LA BASE DE DATOS | 9 |
| STORED PROCEDURES DE LA BASE DE DATOS | 10 |
| TRIGGERS DE LA BASE DE DATOS | 12 |

INTRODUCCIÓN

Este proyecto se centra en el desarrollo de una base de datos para una academia de fútbol, con el objetivo de centralizar y gestionar eficientemente la información relativa a jugadores, entrenamientos y partidos. Utilizando SQL, el sistema está diseñado para mejorar la administración y proporcionar acceso rápido y organizado a los datos necesarios para la operación diaria de la academia.

OBJETIVO

El objetivo principal de este proyecto es crear una base de datos estructurada y funcional que:

- Facilite la administración de la información de jugadores y entrenadores.
- Permita la planificación y seguimiento sistemático de entrenamientos y partidos.
- Mejore la eficiencia operativa mediante el uso efectivo de consultas y reportes SQL.

SITUACIÓN PROBLEMÁTICA

Actualmente, la academia enfrenta desafíos con la dispersión de datos y la falta de un sistema centralizado, lo que resulta en dificultades para acceder y manejar la información de forma rápida y precisa. La ausencia de una base de datos estructurada limita la capacidad de realizar seguimientos efectivos y de tomar decisiones informadas basadas en datos completos y actualizados.

USUARIO FINAL Y NIVEL DE APLICACIÓN DEL ANÁLISIS

Los usuarios finales de esta base de datos son principalmente los administradores y entrenadores de la academia. Utilizarán el sistema para:

- Ingresar y actualizar datos de jugadores y entrenadores.
- Programar y registrar detalles de entrenamientos y partidos.
- Extraer informes que ayuden a evaluar el progreso y necesidades del plantel.

Los entrenadores y administradores podrán utilizar consultas SQL para acceder a la información necesaria para la planificación diaria y la gestión de jugadores, haciendo el proceso más eficiente y menos propenso a errores.

DESCRIPCIÓN DE LAS TABLAS DE LA BASE DE DATOS

A continuación, se detallan las tablas que conforman la base de datos desarrollada, incluyendo su propósito y estructura de campos.

Tabla 1: Jugadores

Descripción: Contiene información esencial sobre los jugadores inscritos en la academia, incluyendo detalles personales y de categoría.

- **Id_Jugador** (INT, PRIMARY KEY): Identificador único para cada jugador.
- **Nombre_jugador** (VARCHAR): Nombre del jugador.
- **Apellido_jugador** (VARCHAR): Apellido del jugador.
- **Fecha_Nacimiento** (DATE): Fecha de nacimiento del jugador.

- **Id_Categoria** (INT, FOREIGN KEY): Referencia a la categoría del jugador.

Tabla 2: Padres_Representantes

Descripción: Almacena información de contacto de los padres o representantes legales de los jugadores.

- **Id_PR** (INT, PRIMARY KEY): Identificador único para cada padre o representante.
- **Id_Jugador** (INT, FOREIGN KEY): Clave foránea que vincula cada padre o representante a un jugador específico en la tabla Jugadores.
- **Nombre_PR** (VARCHAR): Nombre del padre o representante.
- **Apellido_PR** (VARCHAR): Apellido del padre o representante.
- **Telefono_PR** (VARCHAR): Número de teléfono de contacto.
- **Correo_PR** (VARCHAR): Dirección de correo electrónico.

Tabla 3: Entrenadores

Descripción: Registra a todos los entrenadores, incluyendo su información de contacto y la categoría que entrenan.

- **Id_Entrenador** (INT, PRIMARY KEY): Identificador único para cada entrenador.
- **Nombre_entrenador** (VARCHAR): Nombre del entrenador.
- **Apellido_entrenador** (VARCHAR): Apellido del entrenador.
- **Correo_entrenador** (VARCHAR): Correo electrónico del entrenador.
- **Telefono_entrenador** (VARCHAR): Número de teléfono del entrenador.
- **Id_Categoria** (INT, FOREIGN KEY): Categoría que entrena.

Tabla 4: Categoria

Descripción: Define las diferentes categorías de edad o nivel en las que se clasifican los jugadores.

- **Id_Categoria** (INT, PRIMARY KEY): Identificador único para cada categoría.
- **Nombre_Categoria** (VARCHAR): Nombre descriptivo de la categoría (e.g., Sub-12, Sub-15).

Tabla 5: Entrenamientos

Descripción: Gestiona los detalles de los entrenamientos programados, incluyendo información sobre quién los dirige y cuándo se realizan.

- **Id_Entrenamiento** (INT, PRIMARY KEY): Identificador único para cada sesión de entrenamiento.
- **Id_Categoria** (INT, FOREIGN KEY): Identifica la categoría de jugadores que participan en el entrenamiento.
- **Id_Entrenador** (INT, FOREIGN KEY): Entrenador responsable de la sesión.
- **Fecha_Entrenamiento** (DATE): Fecha en que se lleva a cabo el entrenamiento.
- **Hora_Entrenamiento** (TIME): Hora a la que comienza el entrenamiento.

Tabla 6: Evaluacion_Jugadores

Descripción: Registra evaluaciones periódicas de jugadores, indicando el rendimiento y áreas de mejora.

- **Id_Evaluacion_Jugadores** (INT, PRIMARY KEY): Identificador único de cada evaluación.
- **Id_Jugador** (INT, FOREIGN KEY): Jugador al que se evalúa.
- **Id_Entrenador** (INT, FOREIGN KEY): Entrenador que realiza la evaluación.
- **Calificacion_Evaluacion** (ENUM): Calificación asignada al jugador.
- **Id_Habilidad** (INT, FOREIGN KEY): Habilidad específica evaluada.
- **Comentario** (TEXT): Observaciones adicionales del entrenador.
- **Fecha_Evaluacion** (DATE): Fecha en que se realizó la evaluación.

Tabla 7: Habilidades

Descripción: Define distintas habilidades que se evalúan en los jugadores, como técnica, física, táctica, y psicológica (podría agregarse más habilidades a evaluar a futuro).

- **Id_Habilidad** (INT, PRIMARY KEY): Identificador único de cada habilidad.
- **Nombre_Habilidad** (VARCHAR): Descripción de la habilidad.

Tabla 8: Partidos

Descripción: Registra los partidos disputados, incluyendo detalles sobre las categorías, resultados y fechas.

- **Id_Partidos** (INT, PRIMARY KEY): Identificador único para cada partido.
- **Nombre_Partido** (VARCHAR): Nombre o título del partido.
- **Id_Categoria** (INT, FOREIGN KEY): Categoría que participa en el partido.
- **Fecha** (DATE): Fecha del partido.
- **Resultado** (VARCHAR): Resultado del partido.
- **Id_Estado_Resultados** (INT, FOREIGN KEY): Estado del resultado del partido.

Tabla 9: Estado_Resultados

Descripción: Describe los posibles estados de resultados de los partidos.

- **Id_Estado_Resultados** (INT, PRIMARY KEY): Identificador único para cada estado de resultado.
- **Nombre_Estado_Resultados** (VARCHAR): Descripción del estado (ejemplo: Ganado, Perdido, Empatado, otros).

Tabla 10: Finanzas_Ingresos

Descripción: Gestiona los registros de todos los ingresos financieros de la academia, diferenciando entre ingresos directamente relacionados con los jugadores y otros ingresos externos.

- **Id_Ingreso** (INT, PRIMARY KEY): Identificador único para cada ingreso registrado.
- **Id_Conceptos_Ingresos** (INT, PRIMARY KEY): Identificador único de cada concepto de ingreso.
- **Id_Jugador** (INT, FOREIGN KEY, DEFAULT NULL): Identificador del jugador relacionado al ingreso, si aplica. (Si no, el campo puede permanecer como -1 para representar una "Entidad externa").
- **Descripcion** (VARCHAR): Descripción detallada del ingreso.
- **Fecha_ingreso** (DATE): Fecha en que se realizó el ingreso.

- **Monto** (DECIMAL): Cantidad del ingreso.

Tabla 11: Conceptos_Ingresos

Descripción: Define los diferentes conceptos bajo los cuales se pueden registrar los ingresos.

- **Id_Conceptos_Ingresos** (INT, PRIMARY KEY): Identificador único de cada concepto de ingreso.
- **Nombre_Conceptos_Ingresos** (VARCHAR): Nombre del concepto de ingreso.

Tabla 12: Finanzas_Egresos

Descripción: Gestiona los registros de todos los egresos financieros, donde la entidad que realiza el pago es siempre la academia.

- **Id_Egreso** (INT, PRIMARY KEY): Identificador único para cada egreso registrado.
- **Id_Conceptos_Egresos** (INT, FOREIGN KEY): Identificador único de cada concepto de egreso.
- **Descripcion** (VARCHAR): Descripción detallada del egreso.
- **Fecha_egreso** (DATE): Fecha en que se realizó el egreso.
- **Monto** (DECIMAL): Cantidad del egreso.

Tabla 13: Conceptos_Egresos

Descripción: Define los diferentes conceptos bajo los cuales se pueden registrar los egresos.

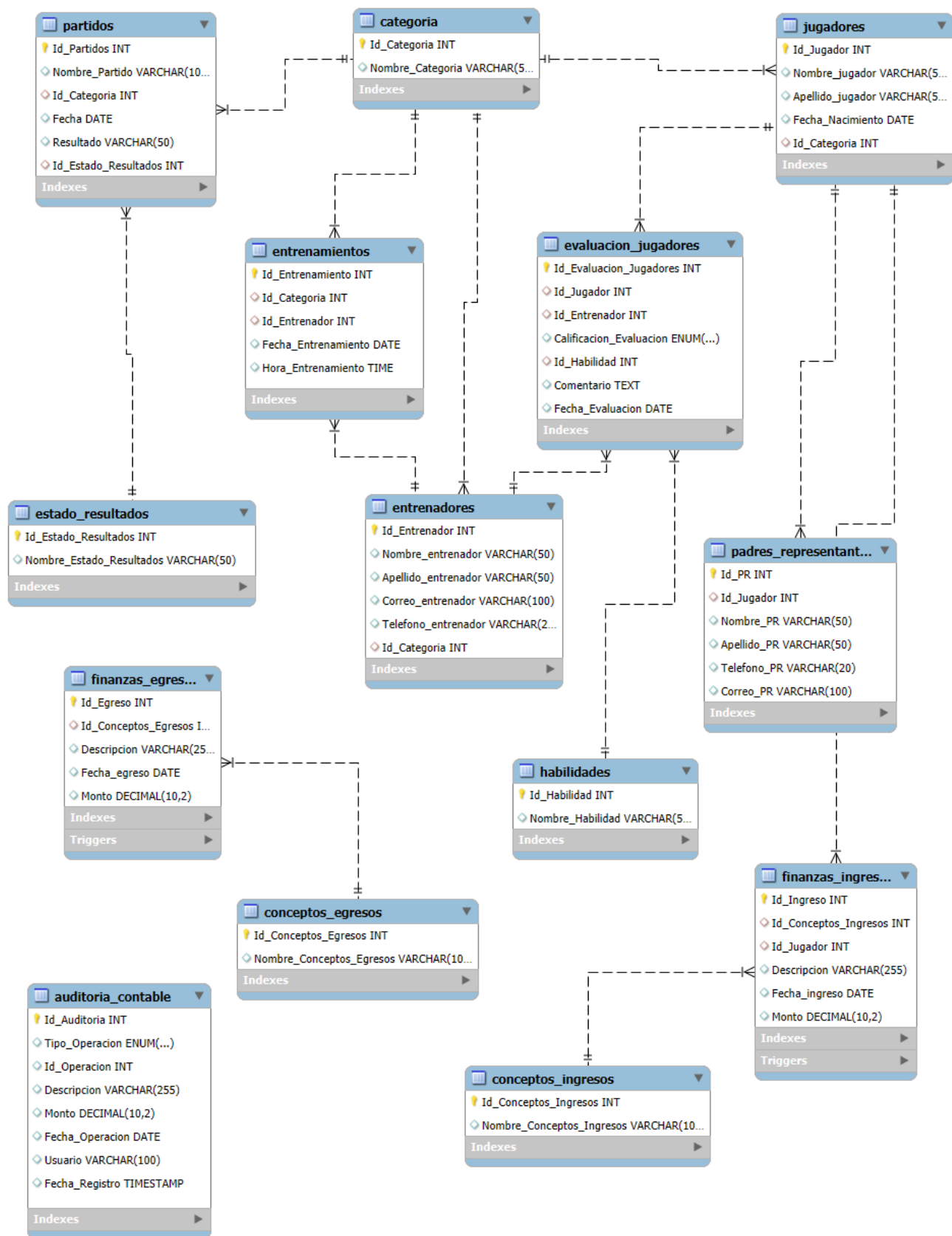
- **Id_Conceptos_Egresos** (INT, PRIMARY KEY): Identificador único de cada concepto de egreso.
- **Nombre_Conceptos_Egresos** (VARCHAR): Nombre del concepto de egreso.

Tabla 14: Auditoria_Contable

Descripción: Registra todas las operaciones contables realizadas en el sistema (ingresos y egresos), incluyendo el tipo de operación, monto, fecha, y usuario que ejecutó la acción. Esta tabla permite realizar un seguimiento detallado de cada transacción para efectos de auditoría y control financiero.

- **Id_Auditoria** (INT, PRIMARY KEY, AUTO_INCREMENT): Identificador único del registro de auditoría.
- **Tipo_Operacion** (ENUM): Tipo de operación registrada ('Ingreso' o 'Egreso').
- **Id_Operacion** (INT): ID del ingreso o egreso relacionado.
- **Descripcion** (VARCHAR): Detalle de la operación registrada.
- **Monto** (DECIMAL): Valor monetario de la operación.
- **Fecha_Operacion** (DATE): Fecha en que se realizó la operación financiera.
- **Usuario** (VARCHAR): Usuario que ejecutó la acción (capturado con CURRENT_USER()).
- **Fecha_Registro** (TIMESTAMP): Fecha y hora en que se registró la auditoría (capturada automáticamente)

DIAGRAMA DE ENTIDAD-RELACIÓN



VISTAS DE LA BASE DE DATOS

Para facilitar la consulta y visualización estructurada de la información, se han creado distintas vistas que reúnen datos relevantes desde múltiples tablas, adaptadas a las necesidades de los distintos usuarios del sistema.

A continuación, se describe cada vista implementada, indicando su propósito, los campos que devuelve y las tablas sobre las cuales se construye.

- **Vista 1: vista_detalle_jugadores**

Descripción: Muestra información completa de cada jugador, incluyendo su categoría y datos de contacto del padre o representante.

Objetivo: Centralizar la información clave de los jugadores para facilitar el contacto y la organización por categoría.

Tablas involucradas: Jugadores, Categoria, Padres_Representantes.

Columnas devueltas: Id_Jugador, Nombre_jugador, Apellido_jugador, Fecha_Nacimiento, Edad, Nombre_Categoria, Nombre_PR, Apellido_PR, Telefono_PR, Correo_PR.

- **Vista 2: vista_evaluaciones_jugadores**

Descripción: Resume las evaluaciones realizadas a cada jugador, incluyendo el entrenador evaluador, la habilidad evaluada y comentarios.

Objetivo: Dar seguimiento al desarrollo y desempeño de los jugadores en distintas habilidades.

Tablas involucradas: Evaluacion_Jugadores, Jugadores, Entrenadores, Habilidades.

Columnas devueltas: Id_Evaluacion_Jugadores, Nombre_jugador, Apellido_jugador, Nombre_entrenador, Apellido_entrenador, Nombre_Habilidad, Calificacion_Evaluacion, Comentario, Fecha_Evaluacion.

- **Vista 3: vista_resumen_partidos**

Descripción: Muestra un resumen de los partidos jugados por cada categoría, incluyendo la fecha, el resultado, el estado del partido, los puntos obtenidos en cada juego y el total acumulado de puntos de la categoría.

Objetivo: Brindar un análisis detallado del desempeño deportivo en cada encuentro y su impacto en el historial competitivo de la categoría.

Tablas involucradas: Partidos, Categoria, Estado_Resultados.

Columnas devueltas: Nombre_Partido, Nombre_Categoria, Fecha, Resultado, Nombre_Estado_Resultados, Puntos_Obtenidos, Puntos_Acumulados.

- **Vista 4: vista_entrenamientos_programados**

Descripción: Lista los entrenamientos agendados, indicando la categoría, el entrenador a cargo, la fecha y la hora.

Objetivo: Brindar una visión clara de la planificación deportiva por fecha y categoría.

Tablas involucradas: Entrenamientos, Categoría, Entrenadores.

Columnas devueltas: Id_Entrenamiento, Nombre_Categoría, Nombre_entrenador, Apellido_entrenador, Fecha_Entrenamiento, Hora_Entrenamiento

- **Vista 5: vista_finanzas_ingresos_por_concepto**

Descripción: Muestra los ingresos organizados por tipo de concepto (mensualidad, inscripción, uniforme, etc.), junto a la descripción, fecha y monto. Si el ingreso está asociado a un jugador, también se muestra su nombre.

Objetivo: Facilitar la revisión y análisis de los ingresos financieros agrupados por concepto.

Tablas involucradas: Finanzas_Ingresos, Conceptos_Ingresos, Jugadores

Columnas devueltas: Concepto, Fecha_ingreso, Descripcion, Monto, Nombre_jugador, Apellido_jugador.

- **Vista 6: vista_finanzas_egresos_por_concepto**

Descripción: Muestra los egresos financieros clasificados por tipo de concepto (pago de entrenador, administrativos, materiales, mantenimiento, etc.), junto a su fecha, monto y descripción.

Objetivo: Permitir un control claro de los gastos institucionales según su naturaleza.

Tablas involucradas: Finanzas_Egresos, Conceptos_Egresos.

Columnas devueltas: Concepto, Fecha_egreso, Descripcion, Monto.

- **Vista 7: vista_resumen_financiero_mensual**

Descripción: Presenta un resumen mensual de la situación financiera de la academia. Muestra los ingresos y egresos registrados en cada mes, calcula la diferencia entre ellos y determina si el resultado fue una “Ganancia” o una “Pérdida”.

Objetivo: Facilitar el análisis financiero periódico, permitiendo identificar meses con balance positivo o negativo y apoyar la toma de decisiones administrativas.

Tablas involucradas: Finanzas_Ingresos, Finanzas_Egresos

Columnas devueltas: Año, Mes, Total_Ingresos, Total_Egresos, Diferencia, Estado.

DESCRIPCIÓN DE LAS FUNCIONES DE LA BASE DE DATOS

El sistema incluye funciones personalizadas que se integran en vistas y procedimientos para facilitar cálculos automáticos, evaluaciones y comparaciones dentro de la base de datos.

- **Función 1: fn_edad_jugador**

Descripción: Calcula la edad actual del jugador.

Objetivo: Usar la edad en vistas informativas.

Tablas utilizadas: Jugadores

- **Función 2: fn_total_puntos_partidos**

Descripción: Devuelve la suma de puntos obtenidos por una categoría.

Objetivo: Evaluar el rendimiento deportivo.

Tablas utilizadas: Partidos, Estado_Resultados

- **Función 3: fn_total_ingresos**

Descripción: Suma el total de ingresos registrados.

Objetivo: Consultar la entrada total de dinero.

Tablas utilizadas: Finanzas_Ingresos

- **Función 4: fn_total_egresos**

Descripción: Suma el total de egresos registrados.

Objetivo: Consultar los gastos totales de la academia.

Tablas utilizadas: Finanzas_Egresos

- **Función 5: fn_estado_balance**

Descripción: Evalúa si el balance es positivo o negativo, devolviendo “Ganancia” o “Pérdida”.

Objetivo: Interpretar el resultado financiero general.

Tablas utilizadas: Usa internamente fn_total_ingresos() y fn_total_egresos()

STORED PROCEDURES DE LA BASE DE DATOS

Se han desarrollado procedimientos almacenados que permiten automatizar operaciones clave de la academia, estructurando la lógica interna de inserción y gestión de datos.

- **Stored Procedures 1: sp_inscribir_jugador**

Descripción:

Inserta un nuevo jugador en la base de datos y registra automáticamente a su padre o representante asociado, utilizando el ID del jugador recién creado.

Parámetros de entrada:

- p_nombre_jugador (VARCHAR): nombre del jugador
- p_apellido_jugador (VARCHAR): apellido del jugador
- p_fecha_nacimiento (DATE): fecha de nacimiento
- p_id_categoria (INT): ID de la categoría a la que pertenece
- p_nombre_pr (VARCHAR): nombre del representante
- p_apellido_pr (VARCHAR): apellido del representante
- p_telefono_pr (VARCHAR): teléfono del representante

- p_correo_pr (VARCHAR): correo electrónico del representante

Objetivo:

Simplificar el proceso de inscripción al registrar al jugador y su responsable legal en un solo paso.

Tablas manipuladas: Jugadores, Padres_Representantes.

- **Stored Procedures 2: sp_registrar_entrenamiento**

Descripción:

Registra un nuevo entrenamiento, indicando la categoría correspondiente, el entrenador responsable, la fecha y la hora.

Parámetros de entrada:

- p_id_categoria (INT): ID de la categoría
- p_id_entrenador (INT): ID del entrenador
- p_fecha (DATE): fecha del entrenamiento
- p_hora (TIME): hora del entrenamiento

Objetivo:

Agilizar la planificación y el registro formal de sesiones de entrenamiento por grupo.

Tablas manipuladas: Entrenamientos.

- **Stored Procedures 3: sp_insertar_ingreso**

Descripción:

Registra un ingreso financiero en el sistema, vinculado a un jugador y a un concepto definido (como mensualidad, inscripción, uniforme, etc.).

Parámetros de entrada:

- p_id_concepto (INT): ID del concepto de ingreso
- p_id_jugador (INT): ID del jugador asociado
- p_descripcion (VARCHAR): detalle o motivo del ingreso
- p_fecha (DATE): fecha del registro
- p_monto (DECIMAL): valor del ingreso

Objetivo:

Facilitar el registro estructurado de ingresos, garantizando coherencia y trazabilidad contable.

Tablas manipuladas: Finanzas_Ingresos-

- **Stored Procedures 4: sp_insertar_egreso**

Descripción:

Registra un egreso financiero institucional (sin asociarlo a un jugador), como pago de materiales, entrenadores o mantenimiento.

Parámetros de entrada:

- p_id_concepto (INT): ID del concepto de egreso
- p_descripcion (VARCHAR): motivo o uso del gasto
- p_fecha (DATE): fecha del egreso
- p_monto (DECIMAL): valor del egreso

Objetivo:

Garantizar el registro contable y ordenado de los gastos operativos o administrativos.

Tablas manipuladas: Finanzas_Egresos.

TRIGGERS DE LA BASE DE DATOS

Se implementaron triggers que permiten registrar automáticamente operaciones financieras en una tabla de auditoría cada vez que se insertan ingresos o egresos. Esto garantiza un control interno eficaz y una trazabilidad completa de las transacciones.

- **Trigger 1: trg_auditoria_ingreso**

Descripción:

Este trigger se activa automáticamente después de que se inserta un nuevo ingreso en la tabla Finanzas_Ingresos. Registra la operación en la tabla Auditoria_Contable.

Tipo de trigger: AFTER INSERT en Finanzas_Ingresos.

Acciones realizadas: Inserta un nuevo registro en Auditoria_Contable con los datos del ingreso y el usuario que lo ejecutó.

- **Trigger 2: trg_auditoria_egreso**

Descripción:

Este trigger se activa automáticamente después de que se inserta un nuevo egreso en la tabla Finanzas_Egresos. Registra la operación en la tabla Auditoria_Contable.

Tipo de trigger: AFTER INSERT en Finanzas_Egresos.

Acciones realizadas: Inserta un nuevo registro en Auditoria_Contable con los datos del egreso y el usuario que lo ejecutó.