

**Tecnicatura Universitaria en Programación**

**Base de Datos II**

**Equipo # 89**

**Integrantes:**

**Urbañski, Joel Román - Legajo 27714**

**Carracedo, Sebastián Horacio - Legajo 25843**

**Strozzi, Nicolas Ezequiel - Legajo 31952**

**Medina, Rodrigo – Legajo 28167**

**Explicación del Sistema**

El sistema desarrollado tiene como finalidad gestionar de manera ordenada y eficiente la actividad diaria de un gimnasio. A través de la base de datos se administran los socios, los tipos de pase disponibles (por día, 8 veces al mes y libre mensual), las clases que se dictan, las inscripciones de los socios a dichas clases y las asistencias registradas.

El objetivo principal es controlar el acceso de los socios según el tipo de pase adquirido, asegurando que solo puedan ingresar aquellos que poseen un pase vigente y con usos disponibles. Para lograrlo, el sistema implementa una serie de reglas de negocio mediante procedimientos almacenados y triggers, que automatizan el control y previenen inconsistencias.

Cada tipo de pase posee su propia lógica de funcionamiento:

* El pase diario permite un solo ingreso el mismo día del pago.
* El pase de 8 veces habilita ocho ingresos durante el mes calendario.
* El pase libre otorga acceso ilimitado durante el mes vigente.

Además, el sistema administra las clases del gimnasio (como funcional, spinning o boxeo), registrando inscripciones y controlando los cupos máximos establecidos para cada una. Mediante un trigger, se evita superar el límite de lugares disponibles, garantizando un uso adecuado de los recursos.

De esta forma, la base de datos no solo facilita el registro de información, sino que también automatiza controles y valida condiciones importantes para el funcionamiento del gimnasio, logrando una gestión más precisa y confiable.

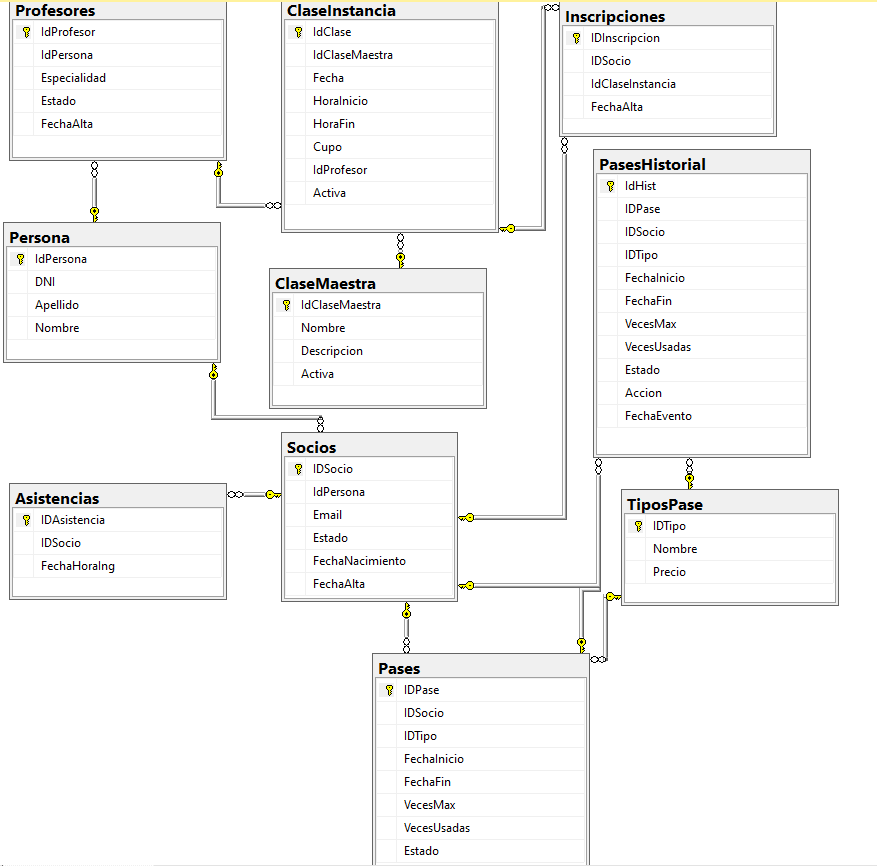
**Funcionalidades Principales**

El sistema desarrollado ofrece una serie de funcionalidades que permiten gestionar de forma integral las operaciones del gimnasio y garantizar el cumplimiento de las reglas de negocio definidas. Entre las principales se destacan:

* **Gestión de socios:** permite registrar, consultar y mantener actualizada la información personal y el estado de cada socio.
* **Control de pases:** administra los tres tipos de pase disponibles (diario, 8 veces al mes y libre mensual), verificando vigencia y cantidad de usos restantes.
* **Registro de asistencias:** automatiza el ingreso de los socios y descuenta los usos correspondientes según el tipo de pase adquirido.
* **Gestión de clases e inscripciones:** permite registrar nuevas clases, establecer cupos máximos e inscribir socios, controlando que no se superen los límites.
* **Automatización mediante procedimientos almacenados y triggers:** asegura la ejecución correcta de las reglas del negocio, evitando ingresos sin pase vigente o con cupos completos.
* **Vistas de control y reportes:** facilitan la visualización de información relevante, como los pases activos y el nivel de ocupación de las clases.

Estas funcionalidades integradas permiten un manejo eficiente de los datos del gimnasio, reduciendo errores y garantizando la coherencia en todos los procesos de registro y control.

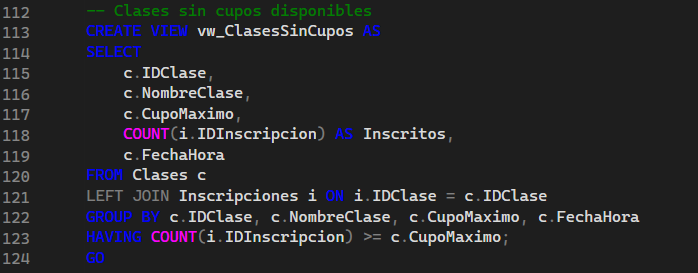
**Diagrama de Entidad Relación**



**Vista 1 – vw\_ClasesSinCupos**

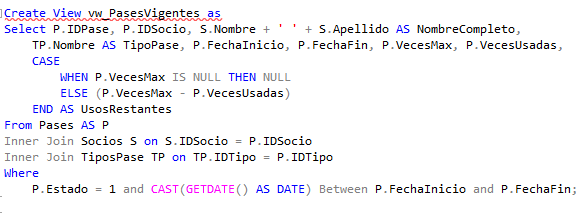
La vista vw\_ClasesSinCupos muestra todas las clases del gimnasio que ya alcanzaron su límite máximo de inscripciones.  
Su objetivo es permitir al administrador detectar rápidamente qué clases no tienen más lugares disponibles, evitando nuevas inscripciones y ayudando a planificar horarios adicionales o clases alternativas.

Se realiza un LEFT JOIN entre las tablas Clases e Inscripciones, agrupando por cada clase y contando la cantidad de socios inscriptos.  
Mediante la cláusula HAVING COUNT(i.IDInscripcion) >= c.CupoMaximo, la vista solo devuelve aquellas clases donde la cantidad de inscriptos es igual o superior al cupo máximo definido.



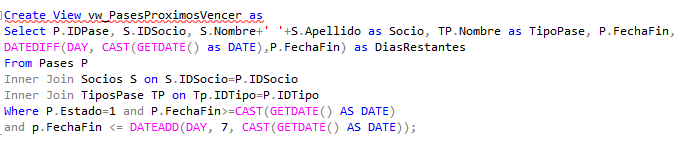
**Vista 2 – vw\_PasesVigentes**

Se utiliza la vista vw\_PasesVigentes, la cual muestra todos los pases activos y vigentes a la fecha actual.  
Solamente podrán visualizarla los administradores y recepcionistas del gimnasio.  
La vista contiene la siguiente información:  
IDPase, IDSocio, Nombre completo del socio, Tipo de pase, Fechas de inicio y fin, Cantidad máxima de usos, Veces utilizadas y Usos restantes.  
En el caso de los pases libres (sin límite de ingresos), el campo *UsosRestantes* devuelve *NULL*.  
Su principal utilidad es controlar qué socios están habilitados para ingresar al gimnasio y cuántas entradas les quedan disponibles según su tipo de pase.

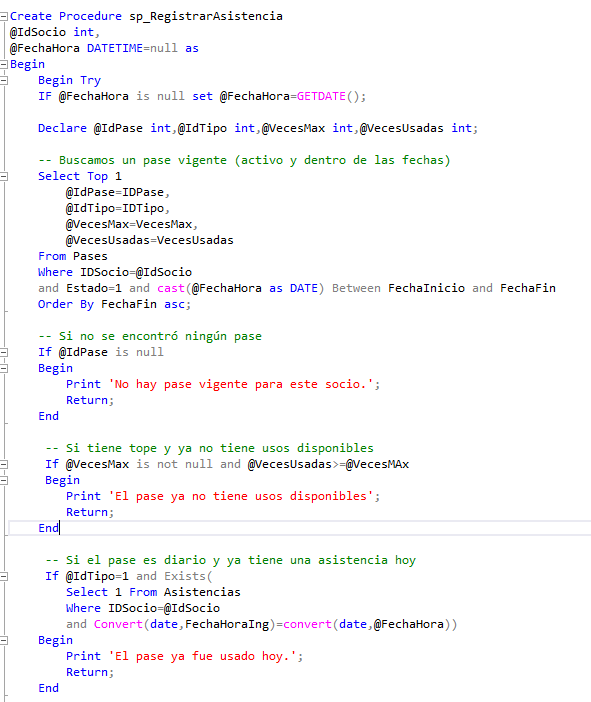


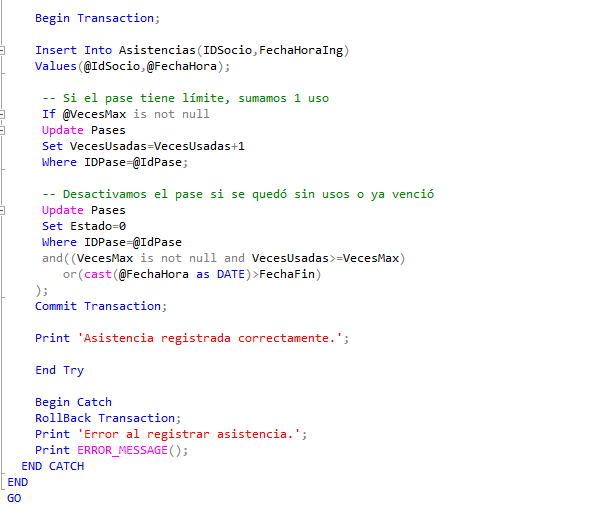
**Vista 3 – vw\_PasesProximosVencer**

Se utiliza la vista vw\_PasesProximosVencer, que permite identificar los pases que vencerán dentro de los próximos siete días.  
Está destinada a los administradores del gimnasio para anticipar las renovaciones y mantener la continuidad de los socios.  
La vista incluye los siguientes datos:  
IDPase, IDSocio, Nombre completo del socio, Tipo de pase, Fechas de inicio y fin, y Días restantes para el vencimiento.  
Gracias a esta información, el sistema puede generar alertas o recordatorios automáticos, facilitando la gestión comercial y administrativa del gimnasio.

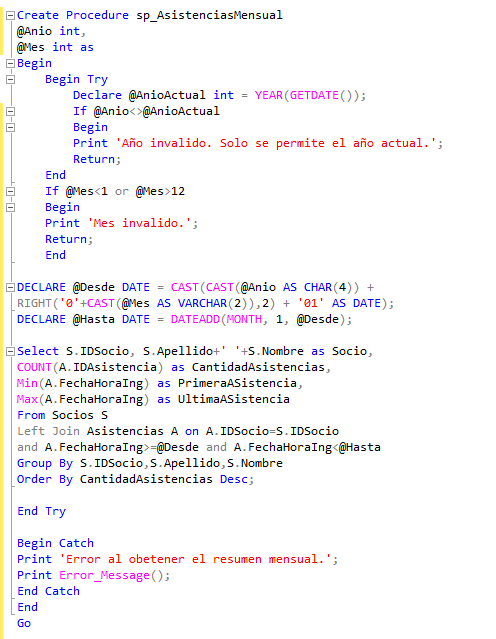


**Procedimiento Almacenado 1**  
Se utiliza el procedimiento almacenado llamado sp\_RegistrarAsistencia que permite registrar la asistencia de un socio. Valida que exista un pase vigente y activo, controla tope de usos (si corresponde) y, si el pase es DIARIO, evita más de un uso por día. Actualiza automáticamente los usos consumidos y desactiva el pase cuando se agota o vence. Maneja errores con TRY...CATCH.





**Procedimiento Almacenado 2**  
Se utiliza el procedimiento almacenado llamado sp\_Asistencias\_ResumenMensual que genera un resumen mensual de asistencias por socio. Valida año y mes, calcula el rango del mes (desde el primer día hasta el primer día del mes siguiente) y devuelve, por socio, la cantidad de asistencias, la primera y la última del período. Incluye socios sin asistencias (usa LEFT JOIN). Maneja errores con TRY...CATCH.

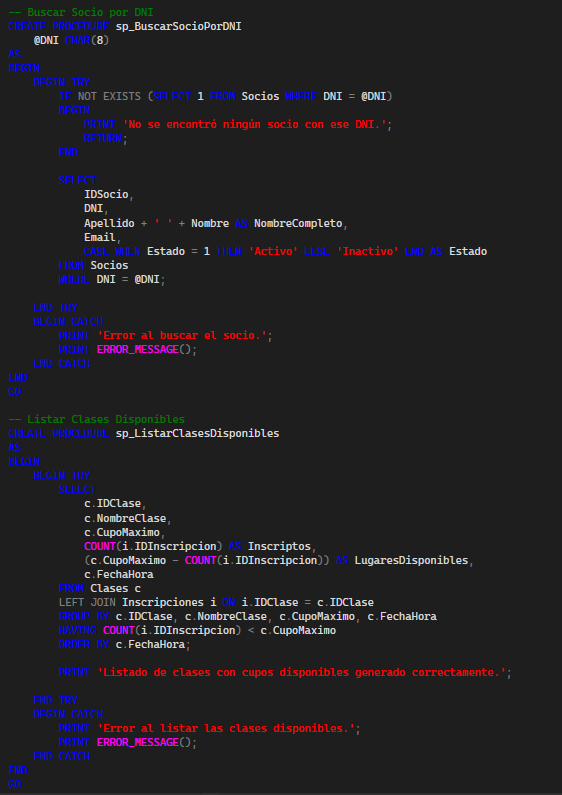


**Procedimiento: sp\_BuscarSocioPorDNI**

Permite buscar rápidamente un socio en la base de datos ingresando su número de DNI.  
Devuelve su nombre completo, email y estado (activo o inactivo).  
Es útil para atención al cliente o control de accesos en recepción.

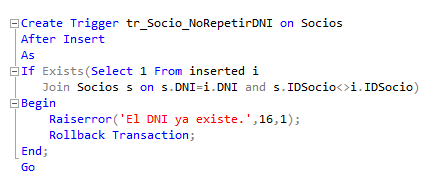
**Procedimiento: sp\_ListarClasesDisponibles**

Muestra todas las clases del gimnasio que aún tienen cupos disponibles, incluyendo la cantidad de lugares libres.  
Sirve para que el personal pueda ofrecer vacantes a nuevos socios o reubicar participantes en horarios menos saturados.



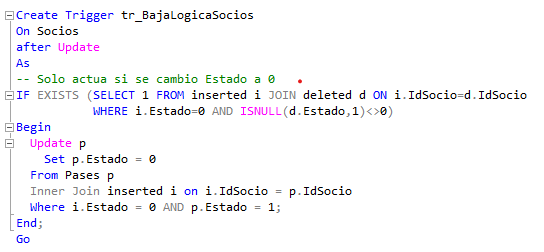
**Trigger: tr\_Socio\_NoREpetirDNI**

**Descripción:**  
Evita que se inserten socios con un DNI duplicado.  
**Propósito:**  
Garantiza la integridad de los datos en la tabla *Socios*, impidiendo la creación de registros repetidos con el mismo documento.



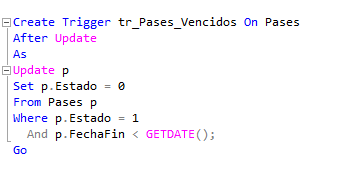
**Trigger: tr\_BajaLogicaSocios**

**Descripción:**  
Cuando un socio es dado de baja (Estado = 0), desactiva automáticamente todos sus pases activos.  
**Propósito:**  
Evitar inconsistencias entre el estado del socio y los pases asociados, garantizando que ningún socio inactivo tenga pases vigentes.



**Trigger: tr\_Pases\_Vencidos**

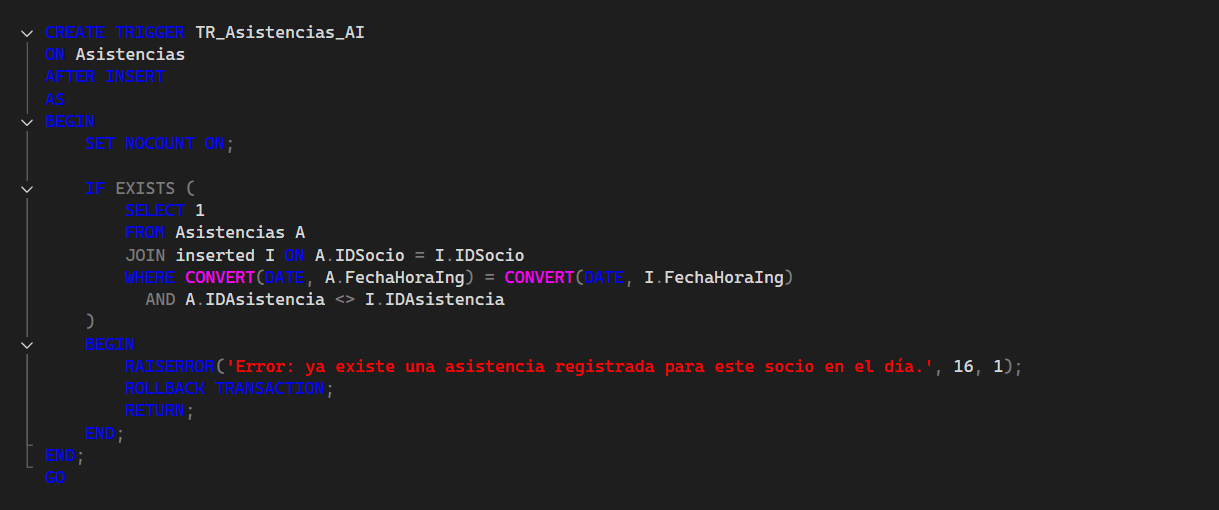
**Descripción:**  
Marca automáticamente los pases vencidos como inactivos cuando la fecha de fin ya pasó.  
**Propósito:**  
Mantener actualizada la tabla *Pases* y asegurar que solo los pases vigentes se consideren activos dentro del sistema.



**TR\_Asistencias\_AI**

**Tipo:** AFTER INSERT  
**Objetivo:** Evita registrar dos asistencias el mismo día para el mismo socio.

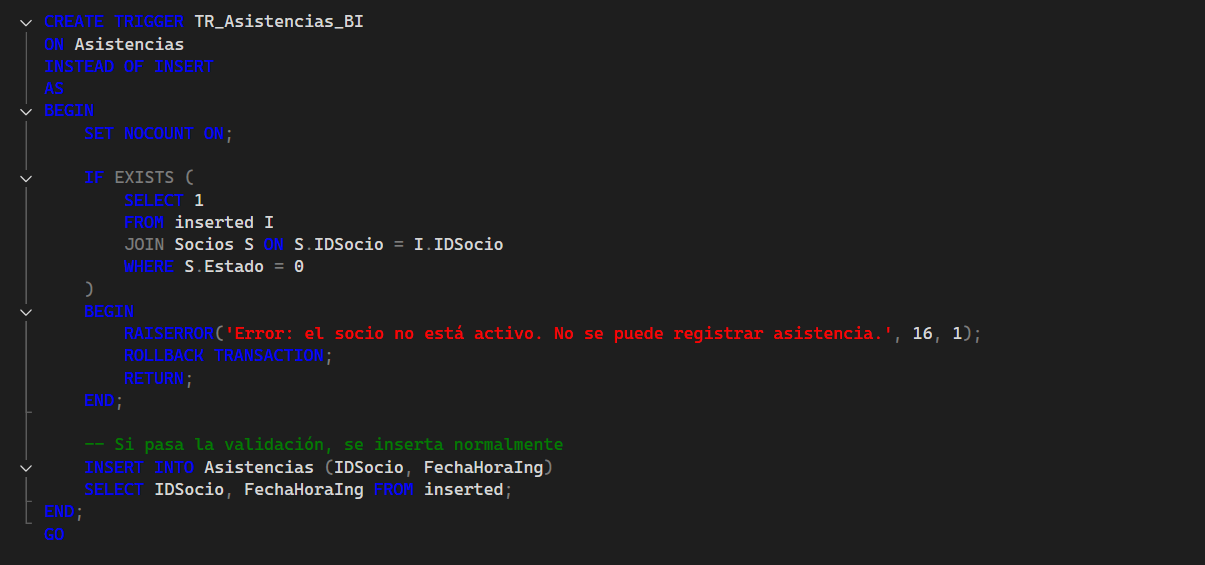
**Explicación:**  
Cada vez que se inserta una nueva asistencia, este trigger revisa si ya existe una asistencia con la misma fecha para ese socio.  
Si detecta un duplicado, cancela la operación con ROLLBACK.



**TR\_Asistencias\_BI**

**Tipo:** INSTEAD OF INSERT  
**Objetivo:** Evita insertar una asistencia si el socio no está activo.

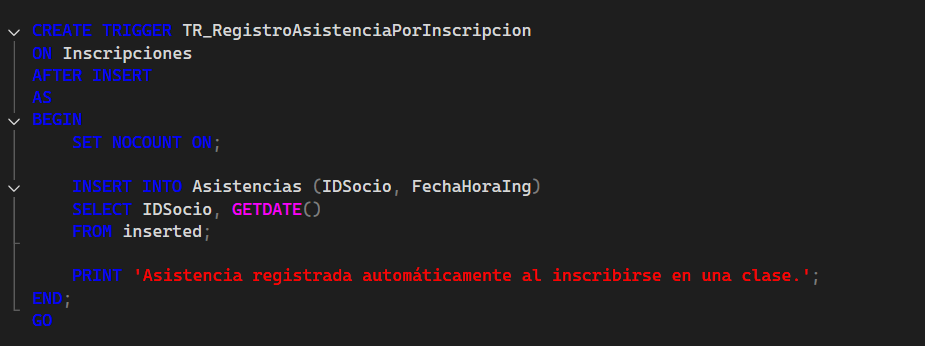
**Explicación:**  
Este trigger reemplaza la inserción normal (INSTEAD OF) y primero valida el estado del socio.  
Si el socio está activo, permite la inserción; si no, la bloquea.



**TR\_RegistroAsistenciaPorInscripcion**

**Tipo:** AFTER INSERT  
**Objetivo:** Inserta automáticamente una asistencia cuando un socio se inscribe a una clase.

**Explicación:**  
Cuando se registra una nueva inscripción en una clase, este trigger crea también un registro en Asistencias, simulando que el socio asistió al momento de la inscripción.



**TR\_ControlCuposInscripciones**

**Tipo:** AFTER INSERT  
**Objetivo:** Impide que se inscriban más socios que el cupo máximo de la clase.

**Explicación:**  
Después de una nueva inscripción, este trigger revisa si la cantidad total supera el cupo.  
Si lo supera, elimina la inscripción recién hecha y lanza un error.

