1. Descripción de tablas

**1. Alumnos**

* **Descripción**: Almacena la información personal y académica de los alumnos.
* **Campos**:
  + ID: Identificador único del alumno.
  + Nombres: Nombres del alumno.
  + Apellido1 y Apellido2: Apellidos del alumno.
  + Semestre: Semestre en el que está inscrito.
  + CantidadCreditos: Total de créditos académicos obtenidos.
  + Reprobadas: Cantidad de materias que el alumno ha reprobado.
  + Grupo: Grupo académico al que pertenece.
  + Especialidad: Especialidad en la que está matriculado.
  + Telefono, Direccion, ImagenPerfil, SeguroSocial, Correo: Información de contacto y otros datos relevantes.

**2. Maestros**

* **Descripción**: Guarda la información personal de los maestros que imparten clases.
* **Campos**:
  + ID: Identificador único del maestro.
  + Nombres, Apellido1, Apellido2: Información personal.
  + Especialidad: Área de especialización del maestro.
  + Telefono, Correo: Datos de contacto.

**3. JefesACargo**

* **Descripción**: Contiene información sobre los jefes que supervisan a otros empleados o programas.
* **Campos**:
  + ID: Identificador único del jefe.
  + Nombre, Apellido1, Apellido2: Información personal.
  + Puesto: Cargo o puesto que ocupa.
  + Correo1, Correo2: Información de contacto.

**4. Encargados**

* **Descripción**: Almacena datos sobre las personas encargadas de supervisar y gestionar diversas tareas.
* **Campos**:
  + ID: Identificador único del encargado.
  + Nombre, Apellido1, Apellido2: Información personal.
  + Cargo: Cargo que desempeña.
  + Telefono, Correo: Datos de contacto.

**5. AdministradoresDB**

* **Descripción**: Guarda la información de los administradores de la base de datos.
* **Campos**:
  + ID: Identificador único del administrador.
  + Nombre, Apellido1, Apellido2: Información personal.
  + Rol: Rol o nivel de acceso que tiene en el sistema.
  + Usuario: Nombre de usuario para acceso al sistema.
  + Email: Correo electrónico del administrador.

**6. Usuarios**

* **Descripción**: Contiene información sobre los usuarios del sistema, incluyendo alumnos, maestros y encargados.
* **Campos**:
  + ID: Identificador único del usuario.
  + Usuario: Nombre de usuario único.
  + TipoUsuario: Indica si es un alumno, maestro, encargado o administrador del sistema.
  + Contraseña: Contraseña cifrada para acceso al sistema.
  + FechaCreacion, FechaVencimiento: Fechas de creación y expiración de la cuenta.
  + AlumnoID, MaestroID, EncargadoID: Referencias a las tablas correspondientes, permitiendo identificar a qué tipo de usuario corresponde.

**7. Organizaciones**

* **Descripción**: Almacena información sobre las organizaciones con las que los alumnos pueden tener relación, especialmente en el contexto de programas de servicio social.
* **Campos**:
  + ID: Identificador único de la organización.
  + NombreOrganizacion: Nombre de la organización.
  + NombreEncargado, Apellido1Encargado, Apellido2Encargado: Datos del encargado en la organización.
  + Cargo, Telefono, Direccion, ImagenPerfil: Información de contacto y otros datos relevantes.

**8. EncargadosServicioSocial**

* **Descripción**: Guarda la información de los encargados de los programas de servicio social.
* **Campos**:
  + ID: Identificador único del encargado de servicio social.
  + Nombre: Nombre del encargado.
  + Cargo: Cargo que desempeña.
  + Permisos: Descripción de los permisos que tiene en su rol.

**9. ProgramasServicioSocial**

* **Descripción**: Contiene información sobre los diferentes programas de servicio social disponibles.
* **Campos**:
  + ID: Identificador único del programa.
  + NombrePrograma: Nombre del programa.
  + Duracion: Duración del programa en meses.
  + Locacion: Ubicación del programa.
  + Objetivos, Actividades: Descripción de lo que se espera lograr y las actividades a realizar.
  + FechaInicio, FechaVencimiento: Fechas relevantes del programa.
  + chk\_Fechas: Restricción para asegurar que la fecha de inicio sea anterior a la de vencimiento.

**10. AlumnosOrganizaciones**

* **Descripción**: Establece las relaciones entre los alumnos, las organizaciones y los programas de servicio social en los que participan.
* **Campos**:
  + AlumnoID, OrganizacionID, EncargadoID, ProgramaID: Referencias a las tablas correspondientes, identificando las relaciones.
  + FechaInicio, FechaFin: Fechas de inicio y fin de la participación del alumno en la organización.

**11. HistorialProgramasAlumnos**

* **Descripción**: Guarda el historial de los programas en los que han participado los alumnos.
* **Campos**:
  + ID: Identificador único del historial.
  + AlumnoID: Referencia al alumno.
  + ProgramaID: Referencia al programa.
  + FechaInicio, FechaFin: Fechas relevantes del programa.

**12. SolicitudesProgramas**

* **Descripción**: Almacena las solicitudes de programas realizadas por los alumnos o encargados.
* **Campos**:
  + ID: Identificador único de la solicitud.
  + ProgramaID: Referencia al programa solicitado.
  + CantidadPersonas: Número de personas solicitadas para el programa.

**13. Bimestres**

* **Descripción**: Registra la asistencia de los alumnos en los bimestres de sus programas.
* **Campos**:
  + ID: Identificador único del registro.
  + AlumnoID, ProgramaID: Referencias a las tablas correspondientes.
  + Bimestre1, Bimestre2, Bimestre3: Estado de la asistencia en cada bimestre.
  + PDF: Almacena un archivo PDF relacionado con el bimestre (ej. comprobantes).

**14. Tutores**

* **Descripción**: Almacena información sobre los tutores que supervisan a los alumnos.
* **Campos**:
  + ID: Identificador único del tutor.
  + Nombre, Apellido1, Apellido2: Datos personales del tutor.
  + NoControl: Número de control del tutor.
  + Puesto: Cargo o puesto que ocupa.
  + JefeACargoID: Referencia al jefe a cargo del tutor.
  + Correo: Información de contacto.

**15. HistorialCreditos**

* **Descripción**: Registra los cambios en la cantidad de créditos de los alumnos.
* **Campos**:
  + ID: Identificador único del registro.
  + AlumnoID: Referencia al alumno.
  + FechaCambio: Fecha en que se realizó el cambio.
  + CantidadCreditosAnterior, CantidadCreditosNuevo: Créditos antes y después del cambio.

**16. HistorialAsignacionProgramas**

* **Descripción**: Almacena el historial de asignación de programas a los alumnos.
* **Campos**:
  + ID: Identificador único del registro.
  + AlumnoID, ProgramaID: Referencias a las tablas correspondientes.
  + FechaAsignacion: Fecha en que se asignó el programa.
  + EstadoAnterior, EstadoNuevo: Registro del estado del programa antes y después de la asignación.

2. Instrucciones para Ejecutar el Contenedor en Docker

**2.1 Requisitos Previos**

* Tener Docker instalado en tu máquina. Puedes descargarlo desde Docker.

**2.2 Construcción de la Imagen Docker**

1. Navega al directorio donde se encuentra tu archivo baseservicio.sql y tu **Dockerfile**:

cd D:\ITE\10mo\GIT\SISSA\_ite\sqlcode

1. Construye la imagen Docker:

docker build -t sissa\_ite .

**2.3 Ejecución del Contenedor**

Ejecuta el contenedor PostgreSQL y expón el puerto 5432:

docker run --name sissa\_ite -p 5432:5432 -d sissa\_ite

**2.4 Conexión a la Base de Datos**

Para acceder a la base de datos en el contenedor:

1. Conéctate a PostgreSQL utilizando psql o cualquier cliente compatible.
2. Si usas psql, puedes entrar al contenedor y conectarte a la base de datos:

docker exec -it sissa\_ite psql -U postgres -d baseservicio

1. Verifica que las tablas se hayan creado correctamente:

\dt

3. Instrucciones para Usar Git

**3.1 Clonación del Repositorio**

Clona el repositorio en tu máquina local:

git clone https://github.com/robealex/nombre\_repositorio.git

cd nombre\_repositorio

**3.2 Subida de Cambios a GitHub**

1. Agrega los archivos al área de preparación:

git add .

1. Realiza un commit:

git commit -m "Agregando Dockerfile y archivos SQL"

1. Sube los cambios a GitHub:

git push origin main

# Alters a tablas una vez instaladas

**Documentación SQL de Modificaciones en Tablas**

1. **Borrar Restricción**
   * ALTER TABLE Alumnos DROP CONSTRAINT chk\_cantidadcreditos;
     + Elimina la restricción chk\_cantidadcreditos en la tabla Alumnos, que anteriormente limitaba los valores permitidos en la columna CantidadCreditos.
2. **Alteración de la Tabla Alumnos**
   * ALTER TABLE Alumnos
     + Se configura cada columna para no permitir valores nulos (SET NOT NULL) en varias columnas de Alumnos, lo cual garantiza que esos datos deben estar presentes en cada registro.
     + Se agregan restricciones para asegurar la integridad de los datos:
       - chk\_CantidadCreditos: limita CantidadCreditos entre 0 y 100.
       - chk\_Reprobadas: asegura que el campo Reprobadas sea igual o mayor a 0.
       - chk\_TelefonoAlumnos: valida que Telefono contenga exactamente 10 dígitos.
3. **Alteración de la Tabla Maestros**
   * ALTER TABLE Maestros
     + Configura columnas como no nulas y agrega una restricción chk\_TelefonoMaestros para que Telefono cumpla con el formato de 10 dígitos.
4. **Alteración de la Tabla JefesACargo**
   * ALTER TABLE JefesACargo
     + Establece las columnas como no nulas, garantizando que se complete la información de cada jefe a cargo.
5. **Alteración de la Tabla Encargados**
   * ALTER TABLE Encargados
     + Similar a JefesACargo, se configuran las columnas para no permitir valores nulos y se añade chk\_TelefonoEncargados, que verifica que Telefono tenga exactamente 10 dígitos.
6. **Alteración de la Tabla AdministradoresDB**
   * ALTER TABLE AdministradoresDB
     + Define todas las columnas esenciales como no nulas, asegurando datos completos en cada registro de administrador.
7. **Alteración de la Tabla Usuarios**
   * ALTER TABLE Usuarios
     + Asegura que los campos Usuario, TipoUsuario y Contraseña no tengan valores nulos, manteniendo la integridad de la información de cada usuario.
8. **Alteración de la Tabla Organizaciones**
   * ALTER TABLE Organizaciones
     + Configura columnas para no permitir valores nulos y establece chk\_TelefonoOrganizaciones, que garantiza que Telefono sea un número de 10 dígitos.
9. **Alteración de la Tabla EncargadosServicioSocial**
   * ALTER TABLE EncargadosServicioSocial
     + Define las columnas Nombre y Cargo como no nulas para asegurar su completitud.
10. **Alteración de la Tabla ProgramasServicioSocial**
    * ALTER TABLE ProgramasServicioSocial
      + Configura varias columnas como no nulas, lo cual asegura que todos los datos del programa estén presentes.
11. **Alteración de la Tabla AlumnosOrganizaciones**
    * ALTER TABLE AlumnosOrganizaciones
      + Define FechaInicio y FechaFin como no nulas para mantener una trazabilidad completa de la participación de alumnos en organizaciones.
12. **Alteración de la Tabla HistorialProgramasAlumnos**
    * ALTER TABLE HistorialProgramasAlumnos
      + Configura las fechas de inicio y fin como no nulas para mantener la integridad histórica.
13. **Alteración de la Tabla ProgramasEncargados**
    * ALTER TABLE ProgramasEncargados
      + Define las columnas EncargadoID y ProgramaID como no nulas para garantizar asociaciones completas entre encargados y programas.
14. **Alteración de la Tabla SolicitudesProgramas**
    * ALTER TABLE SolicitudesProgramas
      + Asegura que CantidadPersonas no contenga valores nulos, lo cual es esencial para cada solicitud de programa.
15. **Alteración de la Tabla Bimestres**
    * ALTER TABLE Bimestres
      + Configura cada columna de bimestre como no nula para garantizar el registro de cada período académico.
16. **Alteración de la Tabla Tutores**
    * ALTER TABLE Tutores
      + Define como no nulas varias columnas claves en la información de tutores.
17. **Alteración de la Tabla HistorialCreditos**
    * ALTER TABLE HistorialCreditos
      + Configura como no nulas las columnas relacionadas con los créditos académicos históricos para un seguimiento completo de los cambios.
18. **Alteración de la Tabla HistorialAsignacionProgramas**
    * ALTER TABLE HistorialAsignacionProgramas
      + Asegura que las columnas EstadoAnterior y EstadoNuevo no contengan valores nulos para un registro preciso del estado de las asignaciones.
19. **Modificación Adicional en JefesACargo**
    * ALTER TABLE JefesACargo ADD COLUMN Telefono VARCHAR(15);
      + Agrega la columna Telefono para almacenar el número telefónico de cada jefe a cargo, configurado con tipo VARCHAR(15) para flexibilidad en formato.

# Restricciones, triggers y funciones

**1. Restricciones de Unicidad para Evitar Duplicados**

* **Alumnos, Maestros y JefesACargo**:
  + ALTER TABLE Alumnos ADD CONSTRAINT uniq\_Alumnos\_nombre\_apellidos UNIQUE (Nombres, Apellido1, Apellido2);
  + ALTER TABLE Maestros ADD CONSTRAINT uniq\_Maestros\_nombre\_apellidos UNIQUE (Nombres, Apellido1, Apellido2);
  + ALTER TABLE JefesACargo ADD CONSTRAINT uniq\_JefesACargo\_nombre\_apellidos UNIQUE (Nombre, Apellido1, Apellido2);
    - Estas restricciones aseguran que no haya duplicación de nombres completos dentro de cada tabla, permitiendo la identificación única de personas.
* **Organizaciones**:
  + ALTER TABLE Organizaciones ADD CONSTRAINT uniq\_Organizaciones\_nombre UNIQUE (NombreOrganizacion);
    - Asegura que cada organización sea única, evitando registros duplicados de la misma organización.
* **Teléfonos**:
  + Ejemplo: ALTER TABLE Alumnos ADD CONSTRAINT uniq\_Alumnos\_telefono UNIQUE (Telefono);
    - En las tablas Alumnos, Maestros, JefesACargo, Encargados y Organizaciones, se añade una restricción para evitar números de teléfono duplicados, garantizando la unicidad de los contactos.

**2. Función y Trigger para Verificar la Cantidad de Personas en un Programa**

* **Función verificar\_cantidad\_personas**
  + CREATE OR REPLACE FUNCTION verificar\_cantidad\_personas() RETURNS TRIGGER
    - Calcula la cantidad de personas inscritas en un programa (AlumnosOrganizaciones) y verifica que al intentar agregar nuevas personas en SolicitudesProgramas, el total no supere el límite de 50. Si el límite se supera, lanza una excepción.
* **Trigger trg\_verificar\_cantidad\_personas**
  + CREATE TRIGGER trg\_verificar\_cantidad\_personas
    - Activa la función verificar\_cantidad\_personas antes de insertar un registro en SolicitudesProgramas, asegurando el cumplimiento de la restricción.

**3. Función y Trigger para Verificar la Cantidad de Créditos de un Alumno**

* **Función verificar\_creditos\_alumno**
  + CREATE OR REPLACE FUNCTION verificar\_creditos\_alumno() RETURNS TRIGGER
    - Recupera el total de créditos de un alumno y verifica que tenga al menos 60 créditos antes de permitir su participación en un programa. Si el alumno no cumple, lanza una excepción.
* **Trigger trg\_verificar\_creditos\_alumno**
  + CREATE TRIGGER trg\_verificar\_creditos\_alumno
    - Activa verificar\_creditos\_alumno antes de insertar en AlumnosOrganizaciones, garantizando el control de créditos requeridos.

**4. Función y Trigger para Evitar que un Alumno Sea Su Propio Encargado o Maestro**

* **Función evitar\_autoservicio**
  + CREATE OR REPLACE FUNCTION evitar\_autoservicio() RETURNS TRIGGER
    - Previene que un alumno sea su propio encargado o maestro, lanzando una excepción si el AlumnoID coincide con EncargadoID o MaestroID.
* **Trigger trg\_evitar\_autoservicio**
  + CREATE TRIGGER trg\_evitar\_autoservicio
    - Ejecuta evitar\_autoservicio antes de la inserción en AlumnosOrganizaciones para impedir autorreferencias inadecuadas.

**5. Función y Trigger para Verificar Fechas de Inicio y Fin en Programas**

* **Función verificar\_fechas\_programa**
  + CREATE OR REPLACE FUNCTION verificar\_fechas\_programa() RETURNS TRIGGER
    - Valida que la FechaInicio de un programa sea menor que FechaFin. Si no se cumple esta condición, lanza una excepción.
* **Trigger trg\_verificar\_fechas\_programa**
  + CREATE TRIGGER trg\_verificar\_fechas\_programa
    - Activa verificar\_fechas\_programa antes de una inserción o actualización en ProgramasServicioSocial para asegurar la consistencia temporal en las fechas de los programas.