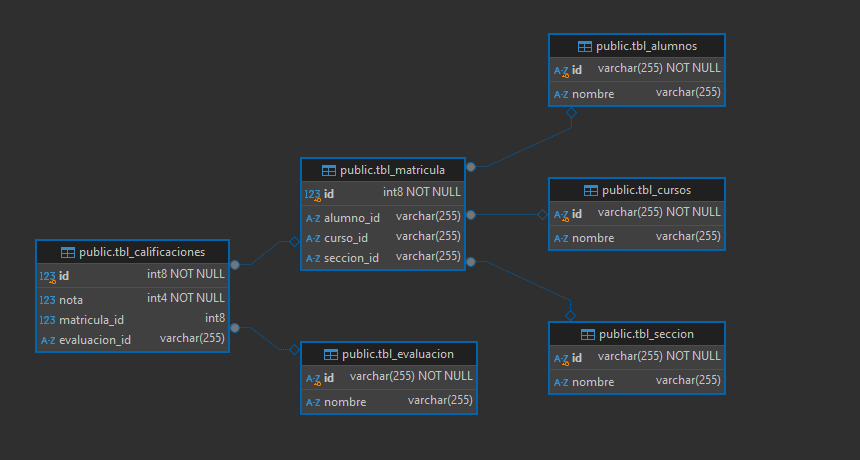
**Programa Expedition Trainee – UTP**

Alvaro Sara Guevara

1. **Documentación de los trabajos realizados**
   1. **Casos de Uso**
      1. **Caso de Uso 1: Registrar Calificaciones**

* **Actor**: Administrador de Calificaciones.
* **Descripción**: El administrador envía una solicitud con la información de calificación en formato JSON al sistema.
* **Acciones**:
  1. El administrador inicia sesión y obtiene un token de autorización.
  2. El sistema valida el token.
  3. El sistema recibe el JSON con los detalles de la calificación.
  4. El sistema almacena la calificación en la base de datos.
     1. **Caso de Uso 2: Obtener Calificaciones por Curso y Sección**
* **Actor**: Profesor/Administrador.
* **Descripción:** Un usuario autorizado solicita las calificaciones de los estudiantes por curso y sección.
* **Acciones**:
  1. El usuario inicia sesión y obtiene un token de autorización.
  2. El sistema valida el token.
  3. El usuario solicita las calificaciones de un curso y sección específicos.
  4. El sistema recupera y devuelve la lista de estudiantes con sus calificaciones.
  5. **Diccionario de Datos y Modelo Entidad-Relación**

****

* + 1. **Tablas Principales:**
  1. **Alumno:** Contiene los datos de los estudiantes.
     + - * id: Integer (PK)
         * nombre: String
  2. **Curso:** Almacena los datos de los cursos.
     + - * id: String (PK)
         * nombre: String
  3. **Sección:** Define las secciones de cada curso.
     + - * id: String (PK)
         * nombre: String
  4. **Evaluación:** Define las evaluaciones del curso.
     + - * id: Integer (PK)
         * nombre: String
  5. **Calificación:** Registra las calificaciones de los estudiantes.
     + - * id: Integer (PK)
         * evaluacion\_id: Integer (FK)
         * matricula\_id: Integer (FK)
         * nota: Float
  6. **Matrícula:** Registra la relación entre Alumno, Curso y Sección
     + **id:** Integer(PK)
     + **alumno\_id:** Varchar(FK)
     + **curso\_id:** Varchar(FK)
     + **seccion\_id:** Varchar(FK)

1. **Inventario de Librerías y Componentes**

* **Spring Boot**: Para la construcción del backend en Java.
* **Spring Security**: Para la autenticación y manejo de tokens de autorización.
* **PostgreSQL**: Base de datos relacional.
* **Hibernate**: ORM para mapear las entidades de la base de datos.
* **Maven**: Para la gestión de dependencias y construcción del proyecto.
* **Lombok:** Reduce el código boilerplate
* **Springboot starter security:** Facilita la implementación de autenticación y autorización.
* **Springboot starter web:** Permite crear aplicaciones web y servicios RESTful

1. **Catálogo de los Servicios Web Implementados**
   1. **Servicio 1: Registrar Calificación**

URL: POST /api/calificaciones/notas

Descripción: Recibe un JSON con la calificación de un estudiante y lo almacena en la base de datos.

Parámetros:

Id\_alumno: ID del alumno.

Id\_evaluacion: ID de la evaluación.

nota: Nota obtenida.

Id\_curso: ID del curso.

Id\_seccion: ID de la sección.

* 1. **Servicio 2: Obtener Calificaciones por Curso**

URL: GET /api/calificaciones/curso/ {curso\_id}

Descripción: Devuelve una lista de estudiantes con sus calificaciones por curso.

Parámetros:

Id\_curso: ID del curso.

* 1. **Servicio 2: Obtener Calificaciones por Sección**

URL: GET /api/calificaciones/seccion/ {seccion\_id}

Descripción: Devuelve una lista de estudiantes con sus calificaciones por sección.

Parámetros:

Id\_seccion: ID de la sección.