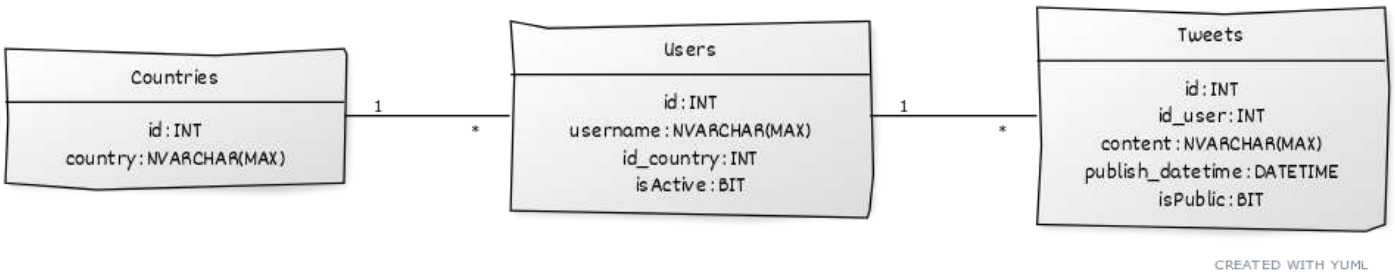
Modelo de parcial Programación II by kron

Diagrama de la base de datos (script.sql):



Endpoints a realizar:

- **GET /api/tweets** -> Devuelve todos los tweets en base a los filtros indicados via **query parameters** (si no se envia ningún filtro, devolver todo). Se deben agregar los siguientes filtros:
    - user -> filtra por el id del usuario
    - country -> filtra por los usuarios los cuales pertenezcan a x país

  BONUS
    - date_start -> fecha de inicio de tweet
    - date_send -> fecha de fin de tweet

- **PUT /api/tweets/{id}** -> modifica la fecha y el contenido del tweet indicado por **path parameter** e informa al usuario con un mensaje de Success(200) o Error (400 o 500). Se debe enviar la siguiente info via **query parameters**:
    - content -> el contenido del tweet a actualizar

- **DELETE /api/tweets/{id}** -> marca el tweet como no público (ver diagrama) e informa al usuario con un mensaje de Success(200) o Error (400 o 500). Se debe enviar el id por **path parameter.**

BONUS

- **POST /api/tweets/** -> recibe un tweet por body (recomendado usar DTO), agrega el tweet a la base de datos y devuelve el tweet creado.

VALIDACIONES:

- **Get:** Solo se deben traer tweets públicos.
- **Put:** Solo se pueden modificar tweets que sean públicos y que no superen la semana de antigüedad.
- **Delete:** Solo se pueden eliminar tweets que sean públicos.
- **Post:** Solo se pueden agregar tweets que no superen el día y hora de la fecha.

El mayor puntaje se consigue usando DTOs, servicio y async-await.