**Nombre: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Cédula: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pts Obt: \_\_\_\_\_\_\_\_\_\_\_ Nota: \_\_\_\_\_\_\_\_\_\_\_\_\_**

**Instrucciones generales**

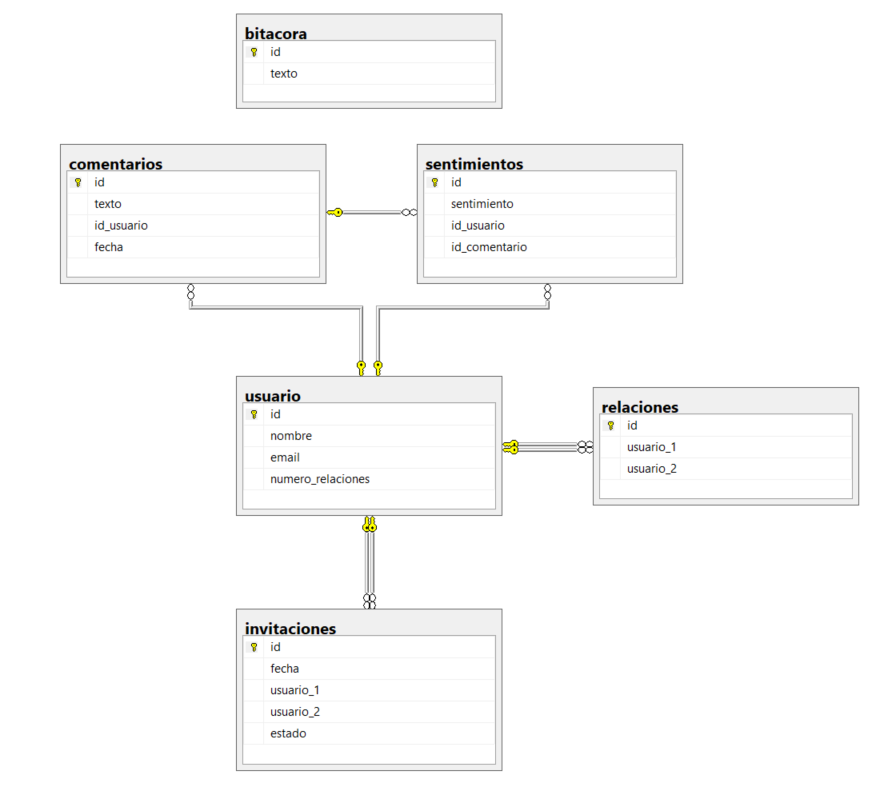
* El examen es en parejas, cualquier caso de plagio o consultas entre grupos se penalizará con una nota de 0 para todos los implicados.
* Desarrolle el código SQL que solucione lo solicitado, se entregará un archivo .sql con el script.

**Parte única.** Lea detenidamente el caso presentado en este examen y desarrolle en PostgresSQL o SQL SERVER cada uno de los procedimientos solicitados a continuación. **Valor 100pts**

# **Descripción del caso**

Una red social es una estructura social compuesta por un conjunto de actores (tales como individuos u organizaciones) que están relacionados de acuerdo a algún criterio (relación profesional, amistad, parentesco, etc.). Normalmente se representan simbolizando los actores como nodos y las relaciones como líneas que los unen. El tipo de conexión representable en una red social es una relación diádica o lazo interpersonal. ​Las redes sociales se han convertido, en pocos años, en un fenómeno global, se expanden como sistemas abiertos en constante construcción de sí mismos, al igual que las personas que las utilizan.

El diagrama desarrollado es el siguiente:



Dentro de los requerimientos a seguir se encuentran los siguientes puntos a desarrollar:

1. Escribir un trigger o disparador que inserte un registro en la tabla bitácora cada vez que ocurre alguna acción de inserción o actualización en la tabla invitaciones. **10pts**
2. Escribir un procedimiento almacenado o función que realice la invitación de amistad de un usuario hacia otro. Es importante validar que no sea el mismo usuario, o que ya la relación de amistad exista. **15pts.**
3. Escribir un trigger o disparador que actualice la cantidad de relaciones de amistad en la tabla usuario, recuerde que debe actualizar la cantidad de ambos usuarios porque es asimétrica la relación. Cuando el estado de una invitación cambia al estado **‘A’**. **10pts**
4. Escribir un procedimiento almacenado o función que verifique y actualice las cantidades de relaciones correctas entre los usuarios con el fin de evitar números erróneos en la cantidad de relaciones. **10pts.**
5. Escribir un procedimiento almacenado o función que simule la eliminación de una relacion de amistad. Esto conllevo actualizar la invitación de amistad entre el usuario 1 y el usuario 2 y a la eliminación de la relacion. **15pts.**
6. Escribir un trigger o disparador que evite no se inserte una nueva invitación que haya sido previamente rechazada o que las personas ya son amigas. **10pts**
7. Escribir un procedimiento almacenado o función que obtenga todas las invitaciones pendientes que tenga mas de 3 meses al día de hoy. **10pts.**
8. Escribir un procedimiento almacenado o función que obtenga el top 3 de las relaciones de amistad por los comentarios negativos 1 y positivos 0. **20pts.**

*“El mayor error que una persona puede cometer es tener miedo de cometer un error.” - Elbert Hubbard.*