

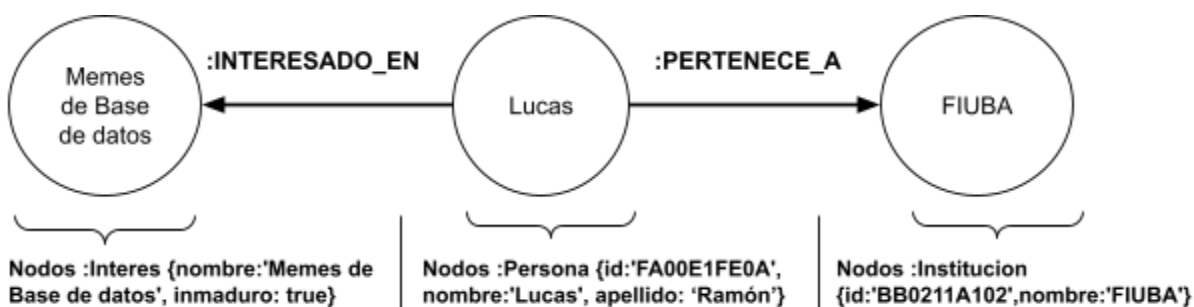
Base de Datos (75.15 / 95.05 / TA044)

Evaluación Integradora - 8 de julio de 2025 - 20251C2

NoSQL		AR		NoSQL		Padrón: Apellido: Nombre: Hojas entregadas:
Recup		Conc.		Proc.		
Nota: <input type="checkbox"/> Aprobado <input type="checkbox"/> Insuficiente						

Criterio de aprobación: El examen está compuesto por 6 ítems, cada uno de los cuales se corrige como B/B-/Reg/Reg-/M. El examen se aprueba con nota mayor o igual a 4 (cuatro) y la condición de aprobación es desarrollar al menos un ítem bien (B/B-) entre los 3 ejercicios de procesamiento de consultas y NoSQL y tener al menos un 60% del examen correcto.

1. (*NoSQL*) Un conocido investigador está intentando analizar el grado de inmadurez de diversas instituciones. Su justificación es simple: “*Porque sí*”. Volcó el resultado de su relevamiento en una base Neo4J donde registró a qué instituciones pertenece cada persona y qué intereses tiene:



Escriba una consulta que devuelva el nombre de las instituciones con alto grado de inmadurez. Esto se determina cuando hay más de 20 personas pertenecientes a la institución para los que más del 50% de sus intereses están registrados como inmaduros. Considere que puede haber muchas personas con mismo nombre y apellido, pero no con mismo id.

2. (*Álgebra Relacional*) La empresa *StalkerSearch* ha sido contratada por el famoso compositor Johann Sebastian Mastrobeiro para detectar stalkers que lo acechan debido a su gran popularidad. La empresa posee la información de los celulares de distintos usuarios que indican la ubicación que tuvieron en base a la geolocalización:

- Ubicacion(id, fecha, hora, id_celular, coord_x, coord_y, id_ciudad)

Mediante una consulta de **álgebra relacional**, devuelva los ids de usuarios que en 2025 visitaron las mismas ciudades que Johann Sebastian visitó en ese mismo año. El id de celular de Johann Sebastian es 12345. Los stalkers pueden haber visitado otras ciudades además de las de Johann.

3. (NoSQL) La plataforma de juegos “Vapor” guarda en una base de datos MongoDB los logros (Achievements) que tienen sus juegos y los jugadores que consiguieron esos logros. Hay un documento por cada logro (los juegos suelen tener muchos logros) con la estructura:

```
{
  id_logro: 123, nombre : "Gran Asasino", juego: "Doom Forever"
  usuarios: [
    {username: "tirubito", conseguido: 2020-10-01, pais:"ARG" },
    {username: "fairy", conseguido: 2020-10-02, pais:"USA" },
    (...) ]
}
```

Genere una consulta que devuelva los 5 países con la mayor cantidad de juegos en los que un usuario del país consiguió al menos un logro. Los documentos de la respuesta tendrán la forma:

```
{pais: "USA", juegos_con_logro: 70}
```

4. (Recuperación) Un SGBD implementa el algoritmo de recuperación UNDO/REDO con checkpoint activo. Los valores iniciales de A, B y C son 0
Se produce un corte en el sistema y el log tiene los valores que se muestran a la derecha.
Indique para cada ítem (A, B y C) cuál es el valor que seguramente tiene en disco (si es que hay seguridad de tener uno) o cuales son los posibles valores que puede tener (si es que no se está seguro)
- 01 (BEGIN, T1);
02 (WRITE T1, A, 0, 1);
03 (BEGIN, T2);
04 (WRITE T1, B, 0, 2);
05 (BEGIN CKPT, T1, T2);
06 (COMMIT, T1);
07 (WRITE T2, A, 1, 3);
08 (WRITE T2, C, 0, 4);
09 (BEGIN, T3);
10 (COMMIT, T2);
11 (WRITE T3, B, 2, 5);
5. (Concurrencia) Un SGBD maneja concurrencia con el método de timestamps. En cierto momento una transacción con timestamp 40 intenta acceder a un ítem X. Indique para cada una de las siguientes situaciones si puede o no leer el ítem y si puede o no escribirlo:
- READ_TS(X) = 35 y WRITE_TS(X) = 35
 - READ_TS(X) = 45 y WRITE_TS(X) = 35
 - READ_TS(X) = 35 y WRITE_TS(X) = 45
 - READ_TS(X) = 45 y WRITE_TS(X) = 45

6. (*Proc. Cons.*) La compañía de envíos *Lentti* quiere premiar a sus mejores repartidores. Para ello, inicialmente ejecuta la siguiente consulta en su base de datos con todos los pedidos:

```
SELECT *  
FROM repartidores r INNER JOIN pedidos p USING (id_repartidor)  
WHERE p.calificacion = 10 OR p.calificacion = 9
```

Posteriormente, agrupará por id de repartidor y premiará a los que mayor cantidad de pedidos hicieron con un hermoso señalador de libros con el logo de la empresa.

Estime el **costo** de resolver la primera consulta, teniendo la siguiente información de catálogo:

REPARTIDORES	PEDIDOS
$n(\text{repartidores}) = 10,000$	$n(\text{pedidos}) = 10,000,000$
$B(\text{repartidores}) = 5,000$	$B(\text{pedidos}) = 1,000,000$
	$V(\text{id_repartidor}, \text{pedidos}) = 10,000$
	$V(\text{calificacion}, \text{pedidos}) = 5$

No hay índices en las tablas y se cuenta con 200 bloques de memoria, el join quiere hacerse usando el método de **loops anidados por bloque** (aunque no resultara ser la mejor opción).

7. (*Migraciones*) (**Optativo**, puede mejorar la nota, **no** reemplaza ningún ejercicio): Dé un breve ejemplo de cómo se podría necesitar una migración, y cuales son los pasos que seguiría para hacerla.