

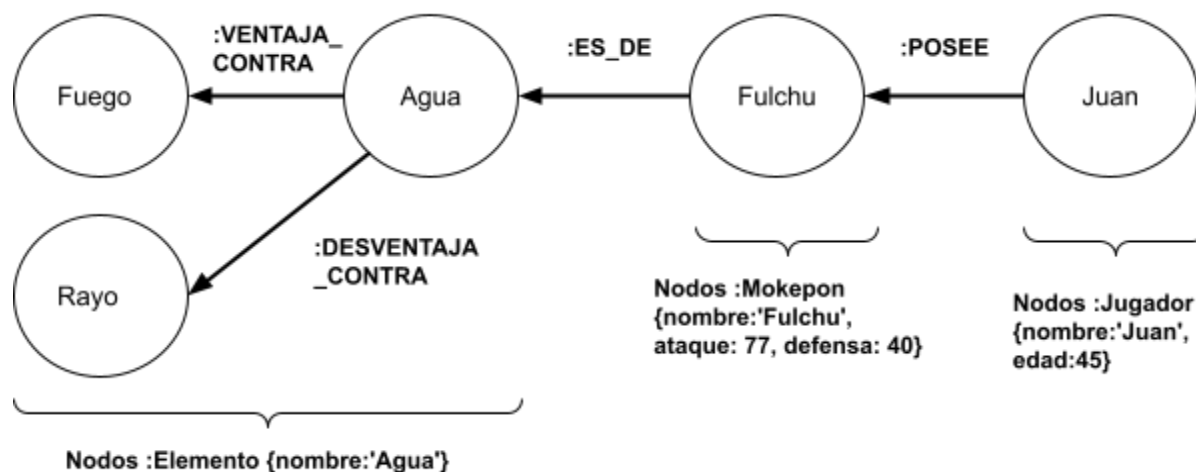
Base de Datos (75.15 / 95.05 / TA044)

Evaluación Integradora - 2 de julio de 2025 - 20251C1

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

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) Juan está jugando al juego online “*Mokepón™*”, donde debe armar un equipo de Mokepones™ para pelear con otros competidores. Las peleas son siempre entre 2 de los Mokepones™ poseídos por cada jugador. Cada Mokepón™ pertenece a un único elemento y los elementos pueden dar ventaja y desventaja con respecto a otros elementos:



Escriba una consulta en Cypher que recomiende dos Mokepones™ a utilizar por Juan para batallar contra otro jugador (parametrizado como de nombre “x”), buscando aquellos dos Mokepones™ con mayor poder de ataque combinado entre ellos, y para los que no haya un Mokepon™ del otro jugador que le genere desventaja a Juan.

2. (SQL) Escriba una consulta, dada la siguiente tabla, que devuelva el padrón de los alumnos que en todas sus materias tienen una nota de aprobación mejor que el promedio de notas de la materia:
- Notasaprobacion (padron, codigo, numero, fecha, nota)

3. (*NoSQL*) Buscando detectar usuarios que en realidad son bots, la empresa quiere obtener un histograma con la cantidad de cada puntaje que deja cada usuario en sus reseñas. Las reseñas están almacenadas en una base MongoDB en la que cada documento representa un restorán:

```
{
  id_resto: 46456, nombre : "Giuseppe Mangiuseppo",
  reseñas: [
    {username: "chipon", puntaje, 5 },
    {username: "focaccio", puntaje, 4 },
    (...) ]
}
```

Genere una consulta que devuelva el histograma pedido, con el siguiente formato de documento:

```
{username: "chipon", puntaje: 5,  cantidad_resto: 35}
```

4. (*Costos*) Considere la siguiente tabla con datos de los hoteles del país:

- Hoteles (id_hotel, nombre, estrellas, provincia, ciudad, direccion, contacto)

Si bien **en distintas provincias puede haber ciudades con el mismo nombre**, Alex no se da cuenta y ejecuta la siguiente consulta SQL buscando sólo por nombre de ciudad:

```
SELECT * FROM Hoteles WHERE ciudad = 'Bella Vista';
```

Conociendo los siguientes metadatos de la tabla:

HOTELES	
$n(\text{Hoteles}) = 2,000,000$	$B(\text{Hoteles}) = 900,000$
$V(\text{provincia}, \text{Hoteles}) = 20$	$V(\text{ciudad}, \text{Hoteles}) = 500$

- Calcule el **costo** de resolver la consulta, sabiendo que hay un índice combinado por las columnas (provincia, ciudad) que no es de clustering y tiene una altura de 4.
- Calcule el **costo** de resolver la misma consulta si el índice en cambio estuviera definido como (ciudad, provincia). Si el costo es el mismo, indicarlo.
- Para los dos ítems anteriores, estime la **cantidad de filas devueltas**.

5. (*Concurrencia*) Un SGBD maneja concurrencia con el método de timestamps. En cierto momento una transacción con timestamp 91 intenta acceder a un ítem X. Para cada una de las siguientes situaciones, indique un par de valores de READ_TS(X) y WRITE_TS(X) que permiten que ocurra la situación, o, en caso de que no existan pares de valores que la hagan posible, indique el por qué con una simple justificación:
- La transacción de timestamp 91 puede leer X pero no escribirlo
 - La transacción de timestamp 91 puede escribir X pero no leerlo
 - La transacción de timestamp 91 no puede ni leer ni escribir X
6. (*Recuperación*) Un SGBD implementa el algoritmo de recuperación REDO con checkpoint activo. Los valores iniciales de A, B y C son 1, 2 y 3 respectivamente.
La transacción T1 intenta grabar el valor 4 en A y el 5 en B, la transacción T2 intenta grabar el valor 6 en A y 7 en C y la transacción T3 intenta grabar 8 en C.
Se produce un corte en el sistema y el log (incompleto) tiene los valores que se muestran a la derecha.
- | |
|--------------------------|
| 01 (BEGIN, T1); |
| 02 (WRITE T1, A, __); |
| 03 (BEGIN, T2); |
| 04 (WRITE T1, B, __); |
| 05 (BEGIN CKPT, T1, T2); |
| 06 (COMMIT, T1); |
| 07 (WRITE T2, A, __); |
| 08 (WRITE T2, C, __); |
| 09 (BEGIN, T3); |
| 10 (COMMIT, T2); |
| 11 (WRITE T3, C, __); |
| 12 (COMMIT, T3); |
- Complete** el log llenando los valores que faltan y explique **cómo se llevará a cabo** el procedimiento de recuperación, indicando **hasta qué punto** del archivo de log se deberá retroceder, y **qué cambios** deberán ser realizados en disco y en el archivo de log.
7. (*Migraciones*) (**Optativo**, puede mejorar la nota, **no** reemplaza ningún ejercicio): Explique brevemente en sus palabras la diferencia entre un adaptador y un ORM para un lenguaje de programación.