

## Parcialito VI: NoSQL

**Compromiso ético:** esta evaluación es domiciliaria, le permite evaluar su nivel de comprensión de la materia antes de la evaluación definitiva. Para que tenga validez usted debe garantizar que ha intentado resolverlo con su mejor esfuerzo; y que las respuestas sean el resultado únicamente de su trabajo y conocimiento individuales.

**Formato:** sólo aceptaremos trabajos presentados en formato pdf y editados en forma electrónica, no fotos, ni manuscritos ni otros formatos devenidos de otras aplicaciones.

Las respuestas deben ser **copiables (admitir CTRL+C) y ejecutables**, por lo que se considerarán **inválidas** respuestas que no lo permitan como una captura de pantalla o símil.

<b>Ejercicios MongoDB</b>	<b>2</b>
<b>Ejercicios Neo4j</b>	<b>3</b>

## Ejercicios MongoDB

Trabajando con el set de datos utilizado en el Taller MongoDB con la siguiente URI: <mongodb+srv://estudiante:bdd20251c@taller.zxyqk.mongodb.net/?readPreference=secondaryPreferred>

### Resuelva:

1. [sobre la base de **tweets**] La captura de tweets es toda en un mismo día: hallar el hashtag más utilizado en cada hora del día (y cuántos usos tuvo).
2. [sobre la base de **Pokémon**] Se quiere construir una nueva tabla con posibles enfrentamientos, para ello se desea categorizar a los Pokémon según su nivel de poder. Para esto, se quieren generar 3 grupos por las estadísticas de **ataque** y **defensa**. Cada grupo estará compuesto por una lista de los pókemon que pertenecen a ese grupo. Para el cálculo de los grupos se deberá utilizar el promedio de esa estadística y operar para obtener los intervalos de valores que definen a cada grupo (AVG):
  - $[0 ; AVG) \rightarrow$  grupo 1 “normal”
  - $[AVG ; 2\ AVG) \rightarrow$  grupo 2 “avanzado”
  - $> 2\ AVG \rightarrow$  grupo 3 “legendario”

```
[
  { _id: "attack",
    // grupos
    normal: [ /* pokemon con atk entre 0 y avg de atk */ ],
    avanzado: [ /* pokemon con atk entre avg y 2 avg de atk */ ],
    legendario: [ /* pokemon con atk mayor a 2 avg de atk */ ]
  },
  { _id: "defense", ... /* idem grupos pero con defensse */ },
]
```

Ayuda: se puede utilizar "\$noval" para no agregar un valor a una lista durante el uso del agregador \$push. Ejemplo:

```
{ $push: { $cond: [
  { falso / verdadero },
  "$campo_valido",
  "$noval" // no se agrega nada a la lista
]
}
```

## Ejercicios Neo4j

Trabajando con el set de datos utilizado en el [Taller Neo4j](#)

### Resuelva:

3. Encontrar fecha, equipos enfrentados (`Game.name`), ciudad y estadio donde se jugaron las finales que no tuvieron goles en los 90 minutos reglamentarios (finales sin goles ni en el primer ni en el segundo tiempo).

Comprobar el resultado con el [listado de las finales de Wikipedia](#),

Pista: estudiar la sintaxis `EXISTS` en Neo4j

<https://neo4j.com/docs/cypher-manual/current/subqueries/existential/>

4. Obtener, para cada mundial, los máximos goleadores.

Comprobar con el [listado de máximos goleadores de Wikipedia](#):

FIFA World Cup	Nombre	Apellido	Goles
"2022 FIFA Men's World Cup"	"Kyllian"	"Mbappé"	8
"2018 FIFA Men's World Cup"	"Harry"	"Kane"	6
"2014 FIFA Men's World Cup"	"James"	"Rodríguez"	6
"2010 FIFA Men's World Cup"	"Thomas"	"Müller"	5
"2010 FIFA Men's World Cup"	"Diego"	"Forlán"	5
"2010 FIFA Men's World Cup"	"Wesley"	"Sneijder"	5
"2010 FIFA Men's World Cup"	"David"	"Villa"	5
"2006 FIFA Men's World Cup"	"Miroslav"	"Klose"	5

...