

Parcialito 6 - NoSQL - CORRECCIÓN RÁPIDA

● Graded

Student

Juana Rehl

Total Points

8.5 / 10 pts

Question 1

1- NoSQL - MongoDB

2 / 2 pts

✓ - 0 pts Correcto

- 0.75 pts Realiza mal el unwind

- 1 pt Error agrupación

💬 Ojo que cuando hacés el unwind, lo tenés que hacer por cada hashtag.text, luego agrupás

Question 2

2- NoSQL - MongoDB

1.5 / 3 pts

✓ + 1.5 pts Resuelve *algo* (pokemon)

✓ + 2 pts Devuelve el resultado correcto

✓ - 0.5 pts Documentos de salida con esquema incorrecto

4

- 1.5 pts Utiliza alguna etapa (`$<stage>`) que hace un JOIN

✓ - 1 pt Agrupa el dataset completo

2

Hacer un agregamiento de todo el dataset no es una buena idea. Imaginemos si es demasiado grande: ni siquiera se puede tener el documento en un solo nodo

3

✓ - 0.5 pts El código requiere especificar para cada estadística cada intervalo.

1

Se debería buscar generalizar de alguna forma no tener que especificar para cada estadística cada uno de los intervalos para definir los grupos

Question 3

3- NoSQL - Neo4j

2.5 / 2.5 pts

✓ - 0 pts Correcto

- 2.5 pts No cumple con la consigna

- 1.5 pts Mal uso de las herramientas del lenguaje

Question 4

4 - NoSQL - Neo4j

2.5 / 2.5 pts

- 0 pts Correcto

- 1 pt No consigue todos los goleadores
- 1 pt Mal uso de las herramientas del lenguaje
- 0 pts Uso innecesario del CALL procedure

Question assigned to the following page: [1](#)

Juana Rehl

112185

Parcialito 6

Ejercicios MongoDB

1-

[

{

\$addFields: {
 hour: { \$hour: "\$created_at" }
}

},

{

\$unwind: {
 path: "\$entities.hashtags"
}

},

{

\$group: {
 _id: {
 hour: "\$hour",
 text: "\$entities.hashtags.text"
 },
 sumatoriaHashtag: {
 \$sum: 1
 }
}

},

{

\$sort: {
 sumatoriaHashtag: -1
},
{

{

\$group: {
 _id: "\$_id.hour",
 texto: { \$first: "\$_id.text" },
 cantidad: { \$first: "\$sumatoriaHashtag" }
}

},

{

\$sort: {
 _id: 1
}

}

]



Question assigned to the following page: [2](#)

2- [{ \$group: { _id: null, avgAttack: { \$avg: "\$attack" }, avgDefense: { \$avg: "\$defense" }, pokes: { \$push: { name: "\$pokename", attack: "\$attack", defense: "\$defense" } } } }, { \$unwind: { path: "\$pokes" } }, { \$group: { _id: null, normal: { \$push: { \$cond: [{ \$lte: ["\$pokes.attack", "\$avgAttack"] }, "\$pokes.name", "\$noval"] } } }, avanzado: { \$push: { \$cond: [{

Question assigned to the following page: [2](#)

```

$and: [
  {
    $gte: [
      "$pokes.attack",
      "$avgAttack"
    ]
  },
  {
    $lt: [
      "$pokes.attack",
      {
        $multiply: ["$avgAttack", 2]
      }
    ]
  }
],
"$pokes.name",
"$noval"
]
}
},
legendario: {
  $push: {
    $cond: [
      {
        $gte: [
          "$pokes.attack",
          {
            $multiply: ["$avgAttack", 2]
          }
        ]
      },
      "$pokes.name",
      "$noval"
    ]
  }
},
normal_def: {
  $push: {
    $cond: [
      {
        $lt: [
          "$pokes.defense",
          "$avgDefense"
        ]
      }
    ]
  }
}

```

Question assigned to the following page: [2](#)

```

        },
        "$pokes.name",
        "$noval"
    ]
}
},
{
avanzado_def: {
$push: {
$cond: [
{
$and: [
{
$gte: [
"$pokes.defense",
"$avgDefense"
]
},
{
{
$lt: [
"$pokes.defense",
{
$multiply: [
"$avgDefense",
2
]
}
]
}
]
},
{
"$pokes.name",
"$noval"
]
}
},
{
legendario_def: {
$push: {
$cond: [
{
$gte: [
"$pokes.defense",
{
$multiply: ["$avgDefense", 2]
}
]
}
],
}
},

```

Question assigned to the following page: [2](#)

```
    "$pokes.name",
    "$noval"
]
}
}
},
{
$project: {
    _id: 0,
    attack: {
        normal: "$normal",
        avanzado: "$avanzado",
        legendario: "$legendario"
    },
    defense: {
        normal: "$normal_def",
        avanzado: "$avanzado_def",
        legendario: "$legendario_def"
    }
}
}
]

```

Questions assigned to the following page: [4](#) and [3](#)

Ejercicios Neo4j

3-

```
MATCH (g:Game {stage: "final"})-[:PLAYED_AT]-(s:Stadium)
WHERE NOT EXISTS {
    MATCH (p:Player)-[sc:SCORED_IN]->(g)
    WHERE sc.minute_regulation <91
}
RETURN g.date, g.name, s.city, s.name
```



4-

```
MATCH (p:Player)-[:SCORED_IN]->(g:Game)-[:PART_OF]->(wc:WorldCup)
WITH p, wc, COUNT(*) AS cantGoles

WITH wc, MAX(cantGoles) AS maximoGoleador
MATCH (p:Player)-[:SCORED_IN]->(g:Game)-[:PART_OF]->(wc)
WITH p, wc, COUNT(*) AS cantGoles, maximoGoleador
WHERE cantGoles=maximoGoleador ★

RETURN wc.name AS FIFAWorldCup ,p.first_name AS Nombre, p.last_name AS Apellido,
cantGoles AS Goles
ORDER BY wc.year DESC
```

