



Instrucciones

- La PEC se realizará de manera individual.
- La fecha de entrega es el 4 de Enero de 2021 a las 23:59.

Nombre y apellidos: Miguel Pérez Caro

Nivel Básico (5 puntos)

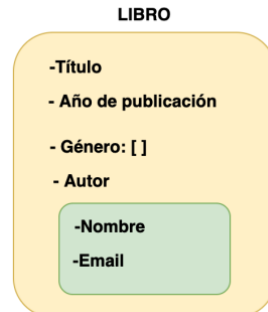
Responde a las siguientes preguntas referentes a aspectos teóricos de los modelos de BD vistos durante el módulo (Documentales, Orientada a Columnas, Clave-Valor y de Grafos). Para resolver el cuestionario, marcar la respuesta deseada en **negrita**.

Preguntas:

1. Indique la respuesta **correcta**:
 - a. Las BD NoSQL ofrecen una resolución de problemas más optima que las BD Relacionales en casos de usos muy específicos.
 - b. Las BD NoSQL ofrecen una resolución de problemas más optima que las BD Relacionales siempre.
 - c. **El modelo relacional se considera obsoleto y debería de ser sustituido cuanto antes por los modelos más flexibles que ofrecen las bases de datos NoSQL.**
 - d. La persistencia políglota ofrece una mejor variedad de modelos de datos, optimizando mejor cada caso de uso, y además no supone una mayor administración ni mantenimiento.
2. Indique la respuesta **correcta**:
 - a. Los sistemas AP nunca disponen de consistencia.
 - b. **En el teorema CAP, buscamos garantizar la P ya que es muy común en sistemas distribuidos.**
 - c. El modelo BASE aparece de una mayor exigencia de las propiedades ACID.
 - d. El teorema CAP defiende que se puede conseguir consistencia junto con una alta disponibilidad, es decir, que la consistencia es compatible con la gestión de datos altamente distribuidos.



3. Supongamos que tenemos el siguiente diseño de agregado en MongoDB:



Si el campo Género es una lista de elementos, ¿cómo podemos filtrar los libros que tienen el género "Acción" o "Drama"? Indique la respuesta **correcta**:

- a. `db.libros.find({genero: "Accion", genero: "Drama"}, {})`
 - b. `db.libros.find({genero.value: "Accion", genero.value: "Drama"}, {})`
 - c. `db.libros.find({genero: "Accion" OR genero: "Drama"}, {})`
 - d. **`db.libros.find({genero: {$in: ["Accion", "Drama"]}}, {})`**
4. Indique la respuesta **correcta**, respecto a MongoDB:
- a. El particionamiento basado en rango distribuye de forma más uniforme que el basado en hash.
 - b. El particionamiento basado en hash recupera documentos secuenciales más rápido que el basado en rango.
 - c. El particionamiento basado en rango tiene mejor escalabilidad si definimos la shard key como incremental.
 - d. **El particionamiento basado en hash distribuye de forma más uniforme que el basado en rango.**
5. Indique la respuesta **correcta**:
- a. Cassandra es un sistema AP, por lo que no se puede configurar para que promocioe la consistencia.
 - b. El modelo de datos de Cassandra se crea primero pensando en una estructura general, y por último en las consultas.
 - c. Para replicar los datos, Cassandra sigue el modelo master-slave, como MongoDB.
 - d. **Un factor de replicación = 3 indica que tendremos tres copias de un dato (contando el original).**



6. Supongamos que tenemos la siguiente definición de familia de columnas:

```
CREATE TABLE tweet (  
  tweet_id uuid PRIMARY KEY,  
  author text,  
  body text  
);
```

¿Qué ocurre si ejecutamos la consulta *SELECT * FROM tweet where author='mmonzon'*?

Indique la respuesta **correcta**:

- a. Devuelve correctamente los tweets cuyo autor es 'mmonzon'.
 - b. Devuelve error porque el campo 'author' debe ser la única PRIMARY KEY de la tabla.
 - c. **Devuelve error porque el campo 'author' no se encuentra indexado.**
 - d. Ninguna de las anteriores.
7. Indique la respuesta **correcta**, respecto a BD Clave-Valor y Riak:
- a. Teóricamente, dentro del valor del agregado, se puede almacenar cualquier tipo de dato.
 - b. La clave de un agregado depende del modelo de negocio a tratar.
 - c. El SGBD no se hace responsable de los datos que se almacenan dentro de este.
 - d. **Todas las anteriores son correctas.**
 - e. Ninguna respuesta es correcta.
8. Si se trata de insertar un objeto clave-valor, sin clave, en Riak, ¿qué ocurre? Indique la respuesta **correcta**:
- a. No permite insertarlo, ya que no estamos especificando la clave.
 - b. Permite insertarlo, pero generando una clave interna que no podemos recuperar ni consultar.
 - c. **Permite insertarlo, pero generando una clave interna que sí se puede recuperar y consultar.**
 - d. Ninguna de las anteriores.
9. Respecto a BD de Grafos y Neo4J, ¿cómo se pueden eliminar nodos de un grafo? Indique la respuesta **correcta**:
- a. Eliminando el nodo directamente.
 - b. Eliminando una relación que tenga el nodo y después eliminar el nodo.
 - c. Eliminar las relaciones entrantes al nodo y después eliminar el nodo.
 - d. Eliminar las relaciones salientes y después eliminar el nodo.
 - e. **Ninguna de las anteriores.**



10. En Neo4J, sobre las etiquetas y propiedades de nodos y relaciones, indique la respuesta correcta:
- Los nodos y las relaciones solo pueden tener 1 etiqueta cada uno.
 - Los nodos pueden tener varias etiquetas, pero las relaciones solo una.**
 - Los nodos solo pueden tener una etiqueta, pero las relaciones pueden tener varias.
 - Los nodos y las relaciones pueden tener múltiples etiquetas, pero solo una propiedad.
 - Los nodos y las relaciones pueden tener múltiples etiquetas y propiedades.
 - Ninguna de las anteriores es correcta.

Nivel Intermedio (3,5 puntos)

Enunciado 1: Cassandra (1,5 puntos)

Basándonos en el modelo de datos de Tweetssandra visto en clase, implementar las siguientes operaciones utilizando CQL. Se puede realizar o en la terminal (cqlsh) o en el notebook de Python. En el primer caso, adjuntar la consulta y captura de pantalla. En el segundo caso, entregar el notebook con los nuevos cambios (puede exportarse a HTML para su mejor corrección).

- Actualizar** la familia de columnas "users" para hacer incluir las direcciones de correo de los usuarios. **(0,5 puntos)**
- Realizar las modificaciones oportunas para permitir que un usuario comente el tweet de otro. **(0,5 puntos)**
- Demostrar el funcionamiento de las dos modificaciones anteriores insertando datos y realizando algunas consultas que puedan llevar a cabo esa demostración. **(0,5 puntos)**

La respuesta a este ejercicio se encuentra en el archivo pec-ej-cassandra-bdnc.html incluido en el zip.

Enunciado 2: Riak (2 puntos)

Tras ver algunos casos prácticos con Riak, tanto utilizando curl como utilizando Python, realizar el ejercicio práctico adjunto a esta entrega, en el cual se trata de simular que un concesionario nos contrata y nos pide que le realicemos una implementación de Riak en sus sistemas.

Para la entrega, basta con exportar el notebook resultante del ejercicio en formato HTML, de tal forma que se puedan evaluar los comandos lanzados, los resultados y las posibles explicaciones.

La respuesta a este ejercicio se encuentra en el archivo pec-ej-riak-python-concesionario.html incluido en el zip.



Nivel Avanzado (1,5 puntos)

En Neo4J, importar la base de datos de "Movie Graph", e indicar el código necesario (con pantallazo de ejecución mostrándolo) para resolver las siguientes cuestiones:

1. Obtener las películas anteriores al año 2000 en las que ha participado "Kevin Bacon". Devolver título y año de lanzamiento. **(0,3 puntos)**

La query ejecutada es la siguiente:

```
MATCH (kevin:Person {name: "Kevin Bacon"})-[:ACTED_IN]->(kevinMovies)
WHERE kevinMovies.released < 2000
RETURN kevinMovies.title AS Título, kevinMovies.released AS `Año de Lanzamiento`
```

El resultado al ejecutar la query se puede observar en la siguiente captura de pantalla.

neo4j\$ MATCH (kevin:Person {name:"Kevin Bacon"})-[:ACTED_IN]->(kevinMovies) WHERE kevinMovies.released < 2000 RETURN kevinMovies.title AS Título, kevinMovies.released AS `Año de Lanzamiento`

"Título"	"Año de Lanzamiento"
"Apollo 13"	1995
"A Few Good Men"	1992

2. Indicar los actores que han trabajado bajo el rol de "Morpheus", y la película en la que actuaron bajo ese rol. Devolver: nombre de actor y título de la película. **(0,3 puntos)**

La query ejecutada es la siguiente:

```
MATCH (p)-[r:ACTED_IN]->(m)
WHERE "Morpheus" IN r.roles
RETURN p.name AS Actor, m.title AS Título
```

El resultado al ejecutar la query se puede observar en la siguiente captura de pantalla.

neo4j\$ MATCH (p)-[r:ACTED_IN]->(m) WHERE "Morpheus" IN r.roles RETURN p.name AS Actor, m.title AS Título

"Actor"	"Título"
"Laurence Fishburne"	"The Matrix Revolutions"
"Laurence Fishburne"	"The Matrix Reloaded"
"Laurence Fishburne"	"The Matrix"



3. Obtener el valor medio que le han dado los revisores a las películas revisadas. Devolver: título de película y puntuación media. **(0,3 puntos)**

La query ejecutada es la siguiente:

```
MATCH (p)-[r:REVIEWED]->(m)
RETURN m.title AS Título, avg(r.rating) AS Puntuación
```

El resultado al ejecutar la query se puede observar en la siguiente captura de pantalla.

```
neo4j$ MATCH (p)-[r:REVIEWED]->(m) RETURN m.title AS Título, avg(r.rating) AS Puntuación
```

"Título"	"Puntuación"
"The Replacements"	75.66666666666667
"The Da Vinci Code"	66.5
"Jerry Maguire"	92.0
"Cloud Atlas"	95.0
"The Birdcage"	45.0
"Unforgiven"	85.0

4. Obtener Top 5 actores que han realizado más películas. **(0,3 puntos)**

La query ejecutada es la siguiente:

```
MATCH (p)-[r:ACTED_IN]->(m)
RETURN p.name AS Actor, COUNT(m) AS `Películas Realizadas`
ORDER BY COUNT(m) DESC LIMIT 5
```

El resultado al ejecutar la query se puede observar en la siguiente captura de pantalla.

```
neo4j$ MATCH (p)-[r:ACTED_IN]->(m) RETURN p.name AS Actor, COUNT(m) AS `Películas Realizadas` ORDER BY COUNT(m) DESC LIMIT 5
```

"Actor"	"Películas Realizadas"
"Tom Hanks"	12
"Keanu Reeves"	7
"Meg Ryan"	5
"Hugo Weaving"	5
"Jack Nicholson"	5



5. ¿Existe alguna persona que ha dirigido y actuado en la misma película? **(0,3 puntos)**

La query ejecutada es la siguiente:

```
MATCH (p)-[:ACTED_IN]->(movies)<-[:DIRECTED]-(p)
```

```
RETURN p.name AS `Actor-Director`, movies.title AS Título
```

El resultado al ejecutar la query se puede observar en la siguiente captura de pantalla.

```
neo4j$ MATCH (p)-[:ACTED_IN]->(movies)<-[:DIRECTED]-(p) RETURN p.name AS `Actor-Director`, movies.title AS Título
```



Table



Text



Code

"Actor-Director"	"Título"
"Tom Hanks"	"That Thing You Do"
"Clint Eastwood"	"Unforgiven"
"Danny DeVito"	"Hoffa"