|  |  |
| --- | --- |
| Módulo | Almacenamiento e Integración de Datos |
| Nombre y apellidos | Román López Seoane |
| * Fecha entrega | 05/04/2020 |
|  | |



Hoja de respuestas

Caso Práctico 1

# Requisitos

Para la realización de este caso práctico se establecen una serie de requisitos:

* El sistema debe permitir realizer búsquedas de ofertas de trabajo.
* Los portales de ofertas en los que se realizarán búsquedas serán configurables por parte de cada usuario.
* Las características o criterios de búsqueda serán configurables por parte de cada usuario.
* La aplicación se conectará a los portales y recuperará entradas asociadas a cada usuario.
* Cada entrada devuelta tendrá la siguiente estructura:
  + Nombre del Portal Web.
  + Contenido principal de la oferta.
  + Características que cumple la oferta en base a los criterios definidos por el usuario.
  + Enlace a la misma.
* El sistema guardará una copia de la oferta para asegurarse que no sea eliminada del portal.
* Será necesario gestionar grandes cantidades de usuarios, portales, requisitos de búsqueda y ofertas.

# Aspectos a tener en cuenta durante la definición del sistema

A la hora de definir y seleccionar una tecnología específica para el modelo de datos y para la capa de distribución, se han tenido en cuenta las siguientes consideraciones:

* Se gestionarán grandes cantidades de información y se dará soporte a multiples usuarios. Necesitamos capacidad para poder escalar la solución que definamos.
* Se generan ofertas de trabajo constantemente, por lo que la velocidad de procesamiento será un aspecto clave.
* Los origenes de datos serán diversos y su definición queda fuera del alcance del sistema a implementar. Los formatos y protocolos de comunicación serán diversos.
* El formato de representación de una oferta en cada portal será diferente.
* En el futuro podrán aparecer nuevos portales o desaparecer los ya existentes.
* Los formatos de representación de la información podrán variar con el tiempo.

# Modelo de datos: Relacional VS NO-SQL

A la hora de establecer la estrategia a seguir, lo primero será definir el modelo de datos. La primera decisión pasa por discerner entre modelos relacionales y NO-SQL.

En el caso que se nos presenta, el modelo de datos deberá ser flexible, atendiendo a los aspectos definidos en el punto inmadiatamente anterior. Necesitamos poder almacenar estructuras de datos (que modelarán las ofertas recuperadas por cada usuario) diversas y que podrán cambiar en el futuro. Los modelos relacionales no se adaptarían bien en este contexto debido a que se basa en deficiones estrictas y rígidas. Sería necesario definir un modelo estándar y adaptar la información recogida desde cada origen de datos para hacerlo encajar. Esto require una complejidad añadida y procesamiento de datos adicional. Así, para poder agregar todos los datos y gestionarlos como una unidad, se necesitan almacenamientos de información que sean lo suficientemente flexibles como para albergar datos heterogéneos. En este sentido, las bases de datos relacionales no constituyen una solución adecuada, dado que el tipo de dato esperado es homogéneo, regular y con una estructura previamente definida en el esquema de la base de datos.

Por otro lado, necesitamos una tecnología que nos permita escalar la solución ante la aparición de entornos más exigentes (más usuarios, portales, etc), por tanto el escalado horizontal sera un aspecto clave, no solo para incrementar la capacidad computacional, sino para dotar al sistema de alta disponibilidad mediante una arquitectura distribuida. Por el momento, no existen alternativas tecnológicas basadas en modelos relacionales que nos permitan dar respuesta a este requerimiento.

Teniendo en cuenta lo antes expuesto, nos decantaremos por la utilización de modelos NO-SQL.

A la hora de seleccionar una tecnología concreta dentro de la familia de BBDDs NO-SQL, tendremos que tener en cuenta aspectos como las características de almacenamiento y representación de la información, las capacidades de acceso posteriores y las posibilidades existentes en relación a la distribución y replicación de los datos.

En el contexto del problema que se plantea, la mejor alternativa pasar por utilizar una base de datos orientada a documentos. La información a gestionar (ofertas de empleo) será fácilmente representada en forma de documentos, por ejemplo utilizando estructuras JSON. Así mismo, en contraposición con los modelos basados en clave-valor, nos vendrá bien poder acceder al contenido de cada documento a la hora de interrogar al modelo de datos.

Por otro lado, las bases de datos orientadas a documentos, se encuentran dentro de las familas de bases de datos orientadas hacia agregados. El uso de agregados proporciona algunas ventajas, como la posibilidad de operar sobre clústeres, dado que el agregado se convierte en la unidad natural para la replicación y la distribución.

Teniendo esto en cuenta, se selecciona MONGO DB como BBDD.

# Arquitectura de distribución y replicación de los datos

En el apartado anterior se exponia la posibilidad de desplegar entornos distribuidos como características de las bases de datos basadas en agregados. Esta organización de la información conlleva algunas ventajas, como la gestión de los clústeres, pues se utilizan los agregados como unidad de replicación y distribución de los datos.

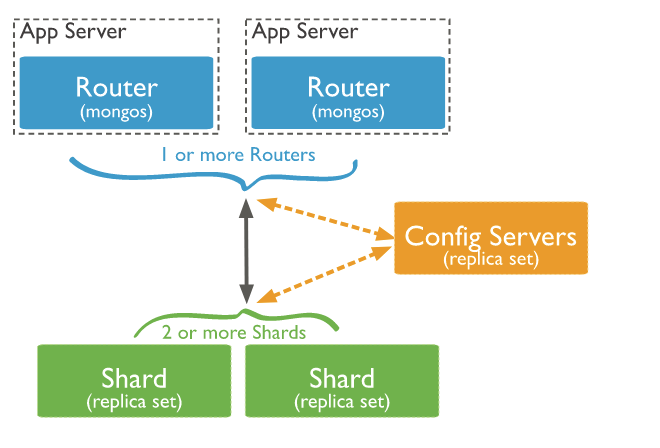
El proyecto actual deberá dar respuesta a necesidades de procesamiento elevadas y que podrán crecer en el futuro, por lo que definir un modelo tolerante a fallos y escalable será fundamental. En este caso, definiremos un sistema escalable haciendo uso de las capacidades que MONGO DB ofrece en este sentido:

* Capacidad para dividir conjuntos de datos en múltiples nodos. **Sharding**.
* Posibilidad de escalar horizontalmente cuando es requerido.

En nuestro escenario necesitamos aprovisionar varios servidores MONGO DB y configurar el sharding sobre los datos. No vamos a realizar una definición exacta del número de nodos y sus características debido a que no se dispone de información suficiente, pero si podremos definir la arquitectura a seguir que debería ser dimensionada en función de las necesidades reales.

Nuestra arquitectura se basará en los siguientes componentes:

* **Shards**: cada uno de ellos contiene un subconjunto de datos. Permite distribuir los documentos gestionados entre diferentes nodos, repartiendo la carga entre cada uno de ellos.
* **Mongos**: actúan como routers para gestionar las consultas desde la capa de aplicación.
* **Config servers**: Almacenan los metadatos y configuraciones del clúster.



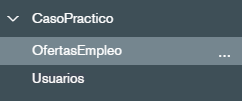
Esta estructurá permitirá incluir nuevos nodos en la cada Router y en la cada Shard para poder mejorar la tolerancia a fallos y escalar el sistema. Así mismo, en la capa Config Servers configuraríamos al menos 2 servidores, algo posible con esta tecnología, para evitar la existencia de un único punto de fallo.

Con esta estrategia Mongo distribuye la carga de lectura y escritura a los diferentes Shard, de modo que cada uno procesa un subconjunto de operaciones dentro del clúster. Así mismo, las consultas deberían incorporar shard keys, de modo que los Router no tengan que hacer un broadcast a todos los shard. En cambio redirigirían las query a los nodos que disponen de la información.

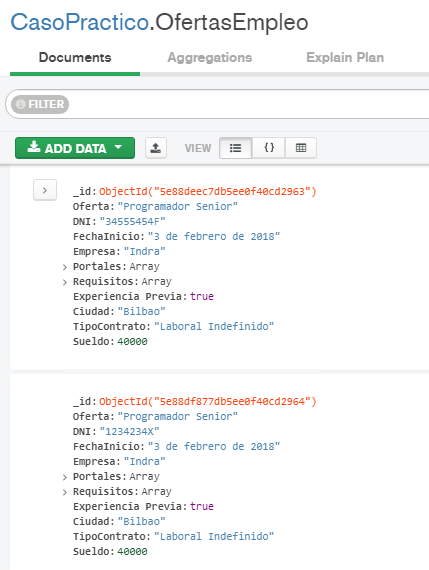
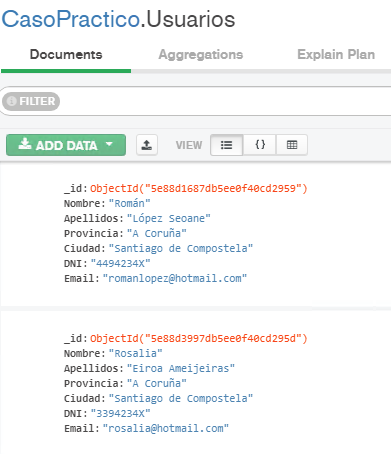
En relación a la **tolerancia a fallos**, esta arquitectura nos permitiría seguir operando incluso si alguno de los servidores Shard no se encuentra disponible. Si es cierto que, en principio, no se podría acceder al subconjunto de información afectada. Para solventar este problema, configuraríamos los Shard (al menos un subconjunto de los mismos) como replica set, de modo que podrían asumir la gestión de las consultas destinadas al nodo o nodos no disponibles. Otra alternativa igualmente válida pasaría por configurar los Config Servers como replica sets.

# Colecciones Mongo DB

Se crearán dos colecciones, una representará a los usuarios y otra a las ofertas de empleo.



Para explicar el formato de cada una de ellas, se muestran algunos ejemplos de documentos ya creados (usuarios y ofertas):

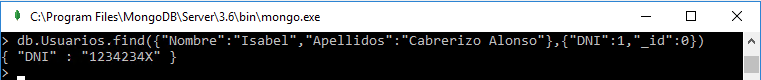


Como nexo de union utilizaremos el DNI. Otra opción sería incorporar el \_id de cada usuario en las ofertas correspondientes. También podríamos optar por un solo documento, pero estaríamos duplicando constantemente la información de los usuarios.

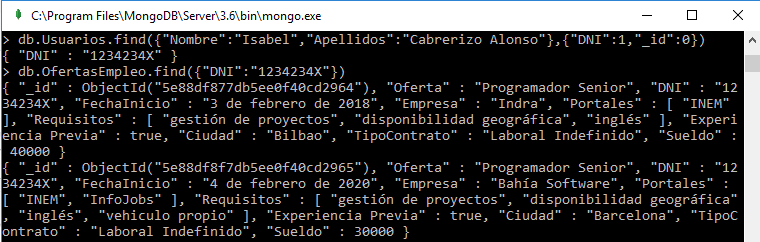
# Consultas solicitadas

1. ***Recuperar todas las ofertas de trabajo que ha solicitado el usuario con nombre “Isabel” y apellidos “Cabrerizo Alonso”.***

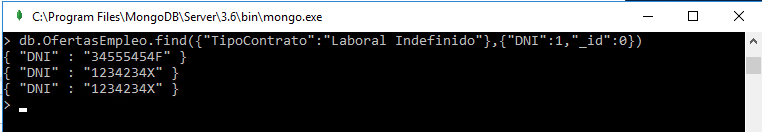
En primer lugar necesitamos el DNI del usuario correspondiente:

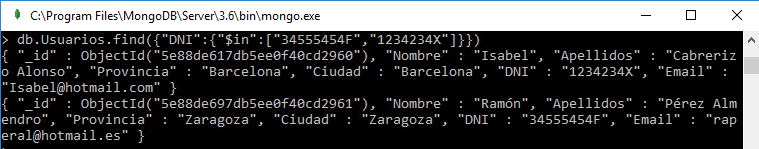


Con esta información podemos recuperar las ofertas correspondientes aplicando una segunda consulta:

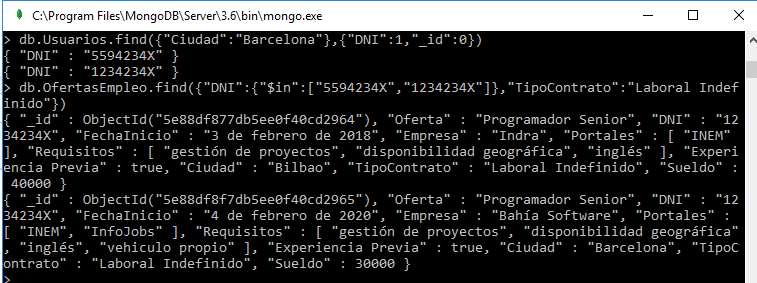


1. ***Recuperar todos los usuarios que han realizado una solicitud para la oferta que tiene como destino “Laboral Indefinido”.***

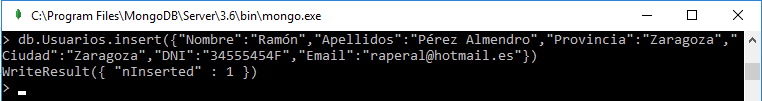
******

******

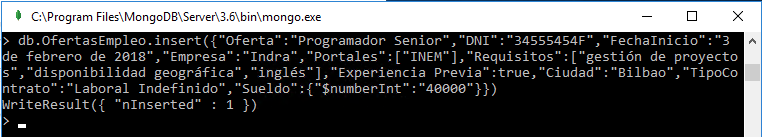
1. ***Recuperar todos los usuarios que viven en Barcelona y han realizado una solicitud para la oferta que tiene como destino de trabajo Madrid.***



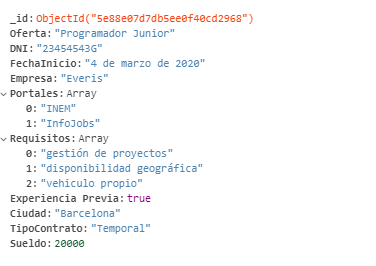
1. ***Añadir un usuario llamado “Ramón Pérez Almendro”, con DNI 34555454F, ciudad de origen: Zaragoza, email: raperal@hotmail.es.***

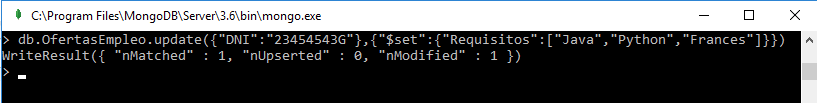


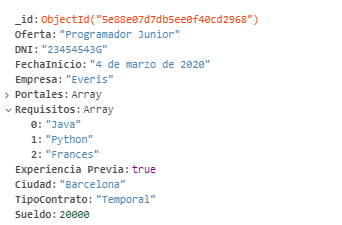
1. ***Añadir una oferta con los datos indicados.***

******

1. ***Modificar el documento asociado con un usuario que tiene como DNI 23454543G, de manera que en todas las peticiones aparezca, en el campo “Requisitos”, los valores “Java”, “Python”, “Francés”.***

******





Caso Práctico 2

# Creación de la BBDD

//NODOS "Director"

CREATE (n:Director {Nombre: "Juan Carlos Fresnadillo"});

CREATE (n:Director {Nombre: "Robert Zemeckis"});

CREATE (n:Director {Nombre: "Brad Bird"});

CREATE (n:Director {Nombre: "Francis Lawrence"});

CREATE (n:Director {Nombre: "Joel Schumacher"});

CREATE (n:Director {Nombre: "Betty Thomas"});

//NODOS "Pelicula"

CREATE (n:Pelicula {Nombre: "28 semanas después"});

CREATE (n:Pelicula {Nombre: "Contact"});

CREATE (n:Pelicula {Nombre: "Los Increíbles"});

CREATE (n:Pelicula {Nombre: "Constantine"});

CREATE (n:Pelicula {Nombre: "Batman y Robin"});

CREATE (n:Pelicula {Nombre: "Dr. Dolittle"});

//NODOS "Productora"

CREATE (n:Productora {Nombre: "Fox"});

CREATE (n:Productora {Nombre: "Warner Bros"});

CREATE (n:Productora {Nombre: "Pixar"});

//RELACIONES

Match (d:Director {Nombre : "Juan Carlos Fresnadillo"}),(p:Pelicula {Nombre : "28 semanas después"}) WITH d,p CREATE (d)-[:ES\_AUTOR\_DE]->(p)

Match (d:Director {Nombre : "Robert Zemeckis"}),(p:Pelicula {Nombre : "Contact"}) WITH d,p CREATE (d)-[:ES\_AUTOR\_DE]->(p)

Match (d:Director {Nombre : "Brad Bird"}),(p:Pelicula {Nombre : "Los Increíbles"}) WITH d,p CREATE (d)-[:ES\_AUTOR\_DE]->(p)

Match (d:Director {Nombre : "Francis Lawrence"}),(p:Pelicula {Nombre : "Constantine"}) WITH d,p CREATE (d)-[:ES\_AUTOR\_DE]->(p)

Match (d:Director {Nombre : "Joel Schumacher"}),(p:Pelicula {Nombre : "Batman y Robin"}) WITH d,p CREATE (d)-[:ES\_AUTOR\_DE]->(p)

Match (d:Director {Nombre : "Betty Thomas"}),(p:Pelicula {Nombre : "Dr. Dolittle"}) WITH d,p CREATE (d)-[:ES\_AUTOR\_DE]->(p)

Match (p:Pelicula {Nombre : "28 semanas después"}),(pr:Productora {Nombre : "Fox"}) WITH p,pr CREATE (p)-[:PRODUCIDA\_POR]->(pr)

Match (p:Pelicula {Nombre : "Contact"}),(pr:Productora {Nombre : "Warner Bros"}) WITH p,pr CREATE (p)-[:PRODUCIDA\_POR]->(pr)

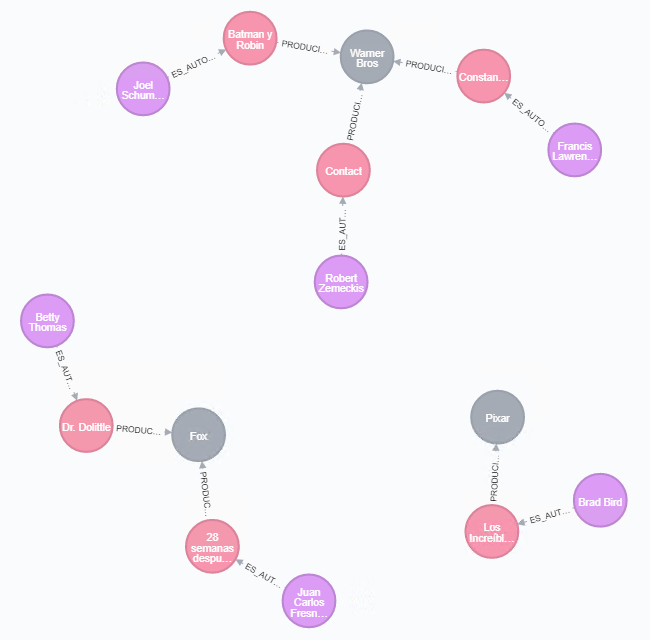
Match (p:Pelicula {Nombre : "Los Increíbles"}),(pr:Productora {Nombre : "Pixar"}) WITH p,pr CREATE (p)-[:PRODUCIDA\_POR]->(pr)

Match (p:Pelicula {Nombre : "Constantine"}),(pr:Productora {Nombre : "Warner Bros"}) WITH p,pr CREATE (p)-[:PRODUCIDA\_POR]->(pr)

Match (p:Pelicula {Nombre : "Batman y Robin"}),(pr:Productora {Nombre : "Warner Bros"}) WITH p,pr CREATE (p)-[:PRODUCIDA\_POR]->(pr)

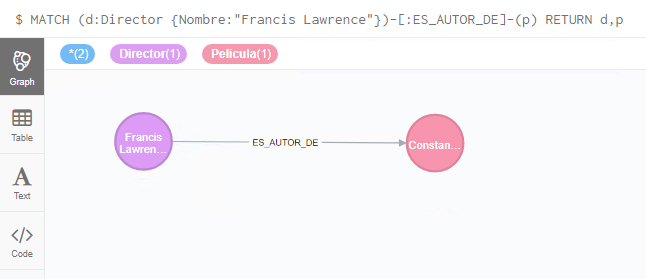
Match (p:Pelicula {Nombre : "Dr. Dolittle"}),(pr:Productora {Nombre : "Fox"}) WITH p,pr CREATE (p)-[:PRODUCIDA\_POR]->(pr)

Se muestran a continuación todos los nodos y relaciones una vez creados

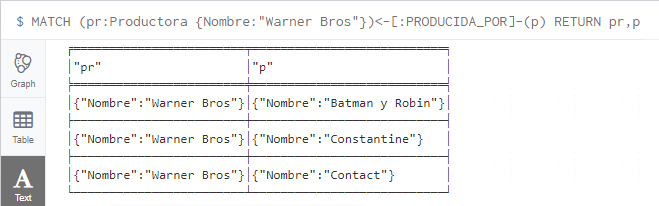


# Consultas

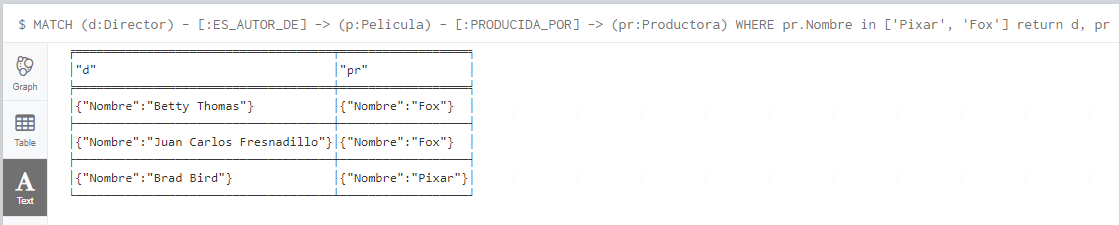
1. ***Recuperar todas las obras de Francis Lawrence.***

******

1. ***Recuperar las películas que están realizadas en la productora “Warner Bros”.***

******

1. ***Recuperar los directores que han producido en Fox y Pixar.***

******

1. ***Eliminar el nodo referido a “Brad Bird”.***

