

2º EVALUACIÓN – PROYECTO BASES DE DATOS (NOSQL). I.E.S. PUNTA DEL VERDE.

**Santiago Gutiérrez
Romero.**

“DATA BROKERS”

1. Introducción y contexto:

El siguiente proyecto esta basado en una empresa de “DATA BROKERS” este tipo de empresas se dedican a la compra y venta de datos, aunque posteriormente a su venta modifican dichos datos de tal manera que los proveen de más información para poder aumentar su precio.

En este caso estamos trabajando con tres colecciones llamadas Personas, redes sociales y barrios. La colección que nuestra empresa ficticia ha comprado ha sido la de Personas la cual contiene información personal en cuanto a la actividad en internet de diferentes individuos, entre esta información se encuentran las redes sociales a las que están suscritas estas personas y el código postal de la ciudad y barrio en el que viven. El procedimiento que llevaría a cabo esta empresa sería tomar la colección de Personas y anexarle información referida a su barrio y ciudad (colección barrios) lo cual se puede hacer gracias al código postal que se tiene de la persona, además de información referente a las redes sociales en las que está suscrita dicha persona (colección redes sociales) lo cual se consigue a través de un campo que contiene la colección personas en el cual se indican códigos que referencian a redes sociales en concreto.

Una vez que la empresa consigue enriquecer la cantidad de datos de la colección personas y hacer abstracciones a partir de dichos datos ya podría venderlos por un precio mayor al que lo compró.

2. Objetivo proyecto:

Con la realización de este proyecto nos planteamos el objetivo de simular el trabajo que realizan este tipo de empresas, para lograrlo utilizamos la tecnología MONGODB la cual nos permite manejar una gran cantidad de datos de diferentes orígenes.

3. Big Data:

En este caso la naturaleza de este proyecto se podría considerar de tipo Big Data ya que la combinación de datos que albergan estas colecciones, como su volumen, variabilidad y crecimiento a lo largo del tiempo serían literalmente imposibles de tratar mediante tecnologías de bases de datos relacionales, es por ello que hacemos uso de una tecnología de bases de datos no relacionales como así lo es MongoDB.

4. Tipos de campos utilizados en el proyecto:

Los diferentes tipos de campos que se han utilizado en las tres colecciones de este proyecto serían los siguientes:

1. String: Para introducir caracteres alfabéticos.
2. Boolean: Para almacenar datos de tipo booleano.
3. Arrays: Para almacenar listas múltiples de valores.
4. ObjectID: Para identificar cada documento/ objeto.

5. Date: Para almacenar fechas.
6. Null: Para almacenar valores nulos.
7. Nested: Contiene documentos anidados dentro un campo.

5. Descripción de los campos utilizados en las colecciones:

COLECCIÓN PERSONAS:

1. Nombre: [STRING]. Contiene el nombre de los usuarios.
 2. Apellidos: [STRING]. Contiene el apellido de los usuarios.
 3. Publicaciones: [DOUBLE]. Contiene el número de publicaciones en total realizadas por los usuarios.
 4. NombresUsuarios: [ARRAY]. Contiene los nombres de usuario con los que cuenta cada cliente de la red social.
 5. NumPublicaciones: [NESTED]. Contiene el desglose de publicaciones que ha hecho cada usuarios, entre las que podemos encontrar fotos, videos y frases.
 6. HorasDeUso: [DOUBLE]. Contiene el número de horas que el usuario a invertido en el uso de la red social.
 7. Premium: [BOOLEAN]. Contiene información sobre si el usuario tiene membresía premium o no.
 8. FechaIngreso: [DATE]. Contiene la fecha exacta de cuando el usuario creo su cuenta en la primera red social.
 9. Dirección: [STRING]. Contiene información sobre la dirección del usuario.
 10. CodPost: [DOUBLE]. Contiene información sobre el código postal del usuario.
 11. CodRedSS: [ARRAY]. Identificadores de las redes sociales a las que se pertenece.
- Todos estos campos no aparecen en la totalidad de documentos ya que es posible que un usuario no haya realizado aún ninguna publicación, o que todavía no haya hecho uso de la red social, o que haya decidido no especificar su información referente a su número de teléfono o dirección.

COLECCIÓN REDES SOCIALES:

1. Nombre: [STRING]. Contiene el nombre de la RRSS.
2. numeroUsuarios: [DOUBLE]. Número de usuarios de la RRSS
3. usuariosDeBajaXdia: [DOUBLE]. Media de usuarios que se da de baja diariamente.
4. usuariosNuevosXdia: [DOUBLE]. Media de usuarios que se dan de alta diariamente.
5. fechaCreacion: [DATE]. Fecha en la que se creo la red social.
6. DePago: [BOOLEAN]. Indica si la red social es de pago o gratuita.
7. cuotaMensual: [DOUBLE]. Indica el precio mensual que pagaría el usuario por la membresía de la red social.

8. horasUsoMedioXdia: [DOUBLE]. Media de las horas que los usuarios suelen hacer de esta red social diariamente.

9. CodRedSS: [DOUBLE]. Identificador de la red social.

COLECCIÓN BARRIOS:

1. Nombre: [STRING]. Contiene el nombre del barrio

2. CodPost: [DOUBLE]. Contiene el código postal del barrio.

3. Ciudad: [STRING]. Contiene el nombre de la ciudad en la que se encuentra el barrio.

4. Habitantes: [DOUBLE]. Contiene el número de habitantes del barrio.

5. rentaMedia: [DOUBLE]. Contiene la renta media del barrio.

6. Operadores utilizados en la realización de consultas:

A continuación se listarán la totalidad de operadores que se han utilizado durante la realización de este proyecto:

1. \$lookup:

<https://docs.mongodb.com/manual/reference/operator/aggregation/lookup/index.html>

2. \$unwind:

<https://docs.mongodb.com/manual/reference/operator/aggregation/unwind/index.html>

3. \$project:

<https://docs.mongodb.com/manual/reference/operator/aggregation/project/index.html>

4. \$group:

<https://docs.mongodb.com/manual/reference/operator/aggregation/group/index.html>

5. \$sum:

<https://docs.mongodb.com/manual/reference/operator/aggregation/sum/index.html>

6. \$cmp:

<https://docs.mongodb.com/manual/reference/operator/aggregation/cmp/index.html>

7. \$subtract:

<https://docs.mongodb.com/manual/reference/operator/aggregation/subtract/index.html>

8. \$divide:

<https://docs.mongodb.com/manual/reference/operator/aggregation/divide/index.html>

9 \$avg:

<https://docs.mongodb.com/manual/reference/operator/aggregation/avg/index.html>

10. \$bucket:

<https://docs.mongodb.com/manual/reference/operator/aggregation/bucket/index.html>

11. \$push:

<https://docs.mongodb.com/manual/reference/operator/update/push/index.html>

12. \$match:

<https://docs.mongodb.com/manual/reference/operator/aggregation/match/index.html>

13. \$switch:

<https://docs.mongodb.com/manual/reference/operator/aggregation/switch/index.html>

14. \$gt:

<https://docs.mongodb.com/manual/reference/operator/aggregation/gt/index.html>

15. \$gte:

<https://docs.mongodb.com/manual/reference/operator/aggregation/gte/index.html>

16. \$lt:

<https://docs.mongodb.com/manual/reference/operator/aggregation/lt/index.html>

17. \$lte:

<https://docs.mongodb.com/manual/reference/operator/aggregation/lte/index.html>

18. \$set:

<https://docs.mongodb.com/manual/reference/operator/update/set/index.html>

19. \$unset:

<https://docs.mongodb.com/manual/reference/operator/aggregation/unset/index.html>

20. \$eq:

<https://docs.mongodb.com/manual/reference/operator/aggregation/eq/index.html>

21. \$sort:

<https://docs.mongodb.com/manual/reference/operator/aggregation/sort/index.html>

22. \$out:

<https://docs.mongodb.com/manual/reference/operator/aggregation/out/index.html>

23. \$merge:

<https://docs.mongodb.com/manual/reference/operator/aggregation/merge/index.html>

24. \$trunc:

<https://docs.mongodb.com/manual/reference/operator/aggregation/trunc/index.html>

25. \$concat:

<https://docs.mongodb.com/manual/reference/operator/aggregation/concat/index.html>