

# **1º EVALUACIÓN - PROYECTO BASES DE DATOS NO RELACIONALES.**

**I.E.S Punta Del Verde.**

**Santiago Gutiérrez Romero.**

**“RED SOCIAL”**

## **Documento explicativo del proyecto en MongoDB.**

### **Contexto:**

Este proyecto está basado en una base de datos llamada “g2\_proyecto”, la cual contiene una colección llamada “RedSocial”. Los diferentes datos contenidos en los documentos están referidos a información que los propios usuarios de una red social como las que tenemos hoy en día han introducido durante su uso, entre esta información encontramos información personal, como dirección, teléfono o código postal, información referida a cuando los usuarios crearon su cuenta en la red social, o información referida a los nombres de usuario que tienen los usuarios.

### **Objetivo:**

El objetivo de este proyecto es simular como una red social guarda los datos de sus usuarios. Los datos de los usuarios para una red social es uno de los elementos más importantes de la empresa por lo que es primordial tenerlos organizados, seguros y con facilidad de acceso y modificación de los mismos.

### **Big Data:**

En este caso la naturaleza de este proyecto se podría considerar de tipo Big Data ya que la combinación de datos que alberga esta colección, 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.

### **Tipos de campos utilizados en la colección “RedSocial”:**

Los diferentes campos utilizados en esta colección enfocado a una red social ficticia serían los siguientes:

1. String: Para introducir caracteres alfabéticos.
2. Boolean: Para almacenar datos de tipo booleano.
3. Double: Para almacenar valores de punto flotante. En nuestro caso valores numéricos enteros
4. Arrays: Para almacenar listas múltiples de valores.
5. Object: Para identificar cada documentos.
6. Date: Para almacenar fechas.
7. Null: Para almacenar valores nulos.
8. Nested: Contiene documentos anidados dentro un campo.

## Descripción de los campos utilizados en la colección "RedSocial":

Los diferentes campos que hemos utilizados en esta colección contienen la información descrita a continuación:

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 red social.
9. Dirección: [STRING]. Contiene información sobre la dirección del usuario.
10. CodPost: [DOUBLE]. Contiene información sobre el código postal del usuario.
11. Telefono: [DOUBLE]. Contiene información sobre el número de teléfono del usuario.

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, código postal o dirección.

## Operadores utilizados en la realización de consultas:

A continuación se listaran la totalidad de operadores que se ha utilizado a lo largo de la realización de las diferentes consultas aplicadas a la colección "RedSocial"

Query selectors:

1. Comparadores:
  - 1.1. \$eq
  - 1.2. \$gt
  - 1.3. \$gte
  - 1.4. \$lt
  - 1.5. \$lte
  - 1.6. \$ne
2. Lógicos:
  - 2.1. \$and

- 2.2. \$not
- 2.3. \$or
- 3. Dirigidos a elementos:
  - 3.1. \$exists
  - 3.2. \$type
- 4. Para evaluación de contenido:
  - 4.1. \$regex
- 5. Enfocados a arrays:
  - 5.1. \$size

### **Inserción de datos en la colección:**

Para la inserción de datos hemos utilizado el método “insertmany”, concretamente `db.RedSocial.insertMany([ ARRAY DE DOCUMENTOS ])`, con dicho método conseguimos insertar los documentos en la colección especificándolos todos ellos en un array de documentos, de esta manera no tenemos que hacer uso de varios “insertOne” por lo que de esta manera ahorramos tiempo.