

# **PROYECTO BASE DE DATOS NO RELACIONALES (NOSQL)**

-

## **ANÁLISIS DEL DISTRITO: BAHÍA DE CÁDIZ-LA JANDA**

**PROFESOR: ADOLFO SALTO SÁNCHEZ DEL CORRAL**

**ANDRÉS NARANJO JIMÉNEZ**

## Contenido

<b>ANTECEDENTES:</b> .....	2
<b>COLECCIONES</b> .....	3
<b>TRATAMIENTO DE LA INFORMACIÓN: MÉTODO FIND ( )</b> .....	4
<b>TRATAMIENTO DE LA INFORMACIÓN: MÉTODO AGGREGATION( )</b> .....	5

## ANTECEDENTES:

El siguiente proyecto se basa en el estudio de la provincia de Cádiz en tiempos de pandemia. Según la densidad de población y el número de hospitales y centros de salud de la provincia, esta está dividida en cuatro distritos sanitarios: Sierra de Cádiz, Campo de Gibraltar, Jerez-Costa Noroeste y Bahía de Cádiz- La Janda.



*Ilustración 1: Mapa dividido por las Áreas Sanitarias establecidas por la Junta de Andalucía.*

En este caso, el proyecto se va a centrar en la última, debido a su relevancia a nivel de pueblos que la conforman. Además, es el Área con mayor número de habitantes y se trata de la zona que más turismo recibe debido a sus kilómetros de costa.

La recolección de los datos se ha producido los días 17-11-2020, 18-11-2020 y 19-11-2020, en los cuales se ha consultado la página web del Instituto de Estadística y Cartografía de la Junta de

Andalucía, para analizar el impacto del virus y como está evolucionando la pandemia en la zona día a día.

Para el análisis del Área Bahía de Cádiz-La Janda, se ha realizado 3 colecciones diferentes dentro de una misma base de datos y el posterior tratamiento de los datos por medio del sistema de bases de datos NO relacionales, **MONGO DB**.

## COLECCIONES

La base de datos generada contiene 3 colecciones: Característica, Covid y Pueblo.

- La colección “caracteristica” está dentro del archivo JS, características.js. Los atributos que incluye esta colección son:

“\_id”: ID (num)

“Village/City”: Nombre de la ciudad

“Population”: Nº de habitantes (num)

“PC”: Postal Code (num)

“Mayor”: Partido político que gobierna

“Intereses”: gastronomía y monumentos

- La colección “covid” está dentro del archivo JS, CovidCadiz.js. Los atributos que incluyen esta colección son:

“\_id”: ID (num)

“Province”: Provincia

“Sanitary District”: Distrito sanitario

“Day”: día de la toma de muestra (date)

“Covid-19 affected”: incluye a total de contagiados, muertos y curados (num)

“Population(x1000)”: Nº de habitantes (num)

“CovidRate”: Tasa de Contagio (num)

“AboveSpanishAverage”: si el dato anterior esta sobre la media nacional o no (boolean).

- La colección “pueblo” está dentro del archivo JS, PueblosBC-LJ.js. Los atributos que incluyen esta colección son:

“\_id”: ID (num)

“Location”: Nombre del pueblo

“Sanitary District”: Distrito sanitario

“Day”: día de la toma de muestra (date)

“Total”: contagiados totales (num)

“Fatalities”: muertes (num)

“Recovered”: Curados (num)

“Population(x1000)”: Nº de habitantes (num)

“CovidRate”: Tasa de Contagio (num)

“AboveSpanishAverage”: si el dato anterior esta sobre la media nacional o no (boolean).

**En el caso de las colecciones de PueblosBC-LJ.js y CovidCadiz.js se han realizado mediante la recolección de datos durante los días 17-11-2020, 18-11-2020 y 19-11-2020.**

## TRATAMIENTO DE LA INFORMACIÓN: MÉTODO FIND ( )

El método find( ) nos permite, mediante los operadores de consulta, ubicar datos dentro de la base de datos.

En este proyecto se han usado:

- Los operadores comparación: **\$eq, \$gt, \$gte, \$lt, \$lte, \$ne**

*\*los operadores “\$in” ni “\$nin” no han sido usados debido a su carácter de funcionalidad sobre arrays y las colecciones de este proyecto no las contempla.*

- Los operadores lógicos: **\$and, \$not, \$nor, \$or**

- El operador Elemento: **\$exists**

- El operador de Evaluación: **\$regex**

- El operador Array: **\$elemMatch**

- Modificadores de consulta: **\$min, \$max, \$hint**: (obliga al optimizador de consultas a utilizar un índice específico para completar la consulta. Especifique el índice por el nombre del índice o por documento).

## TRATAMIENTO DE LA INFORMACIÓN: MÉTODO AGGREGATION( )

Las operaciones de agregación procesan registros de datos y devuelven resultados calculados. Las operaciones de agregación agrupan valores de varios documentos y pueden realizar una variedad de operaciones en los datos agrupados para devolver un solo resultado

En este proyecto se han usado:

- **\$match:** Empareja los datos con la instrucción introducida.
- **\$group:** Agrupa los registros con el mismo valor dependiendo del campo elegido.
- **\$lookup:** Realiza una unión externa izquierda a una colección no fragmentada en la misma base de datos para filtrar los documentos de la colección "unida" para su procesamiento.
- **\$unwind:** Deconstruye un campo de matriz a partir de los documentos de entrada para generar un documento para cada elemento.
- **\$project:** Se usa como filtro para las siguientes etapas.
- **\$sum:** Realiza el sumatorio de los valores seleccionados.
- **\$avg:** Realiza la media de los valores seleccionados.