

ÍNDICE

1. Introducción al caso de negocio	2
2. Recomendaciones e instrucciones	5
3. Cuestiones sobre el caso de negocio MongoDB	6

1. INTRODUCCIÓN AL CASO DE NEGOCIO

El caso de negocio es el mismo que se ha preparado en el caso de repaso:

Una empresa multinacional decide llevar a cabo un proyecto basado en el internet de las cosas (IoT) con dos objetivos principales:

- Optimizar las condiciones laborales de sus trabajadores en sus puestos de trabajo en oficina, especialmente en época estival.
- Optimizar costes mediante una mejor gestión del consumo eléctrico.

Para ello, selecciona dos de sus sedes en España, concretamente las de Valladolid y Sevilla, como sedes piloto.

El desarrollo del proyecto implica la instalación de dos sensores en cada una de las sedes, los cuales recogen en tiempo real información sobre:

- ❖ Sensor 1: temperatura (°C) y humedad relativa (%) exteriores.
- ❖ Sensor 2: emisiones de CO₂ (kg CO₂ / m²) y consumo eléctrico (kW).

Nota: el sensor que recoge las emisiones de CO₂ envía el acumulado en kg CO₂ / m² de los últimos 15 minutos. Tras el envío de este acumulado se resetea a cero para volver a comenzar otro acumulado durante los próximos 15 minutos, y así sucesivamente.

Estos sensores medirán estos cuatro parámetros entre los días 1 y 14 de julio de 2020, para posteriormente analizar resultados a partir de los datos obtenidos.

Para determinar los valores para los sensores de medición de emisiones de CO₂ y consumo eléctrico, hay que basarse en los datos proporcionados por los enlaces siguientes:

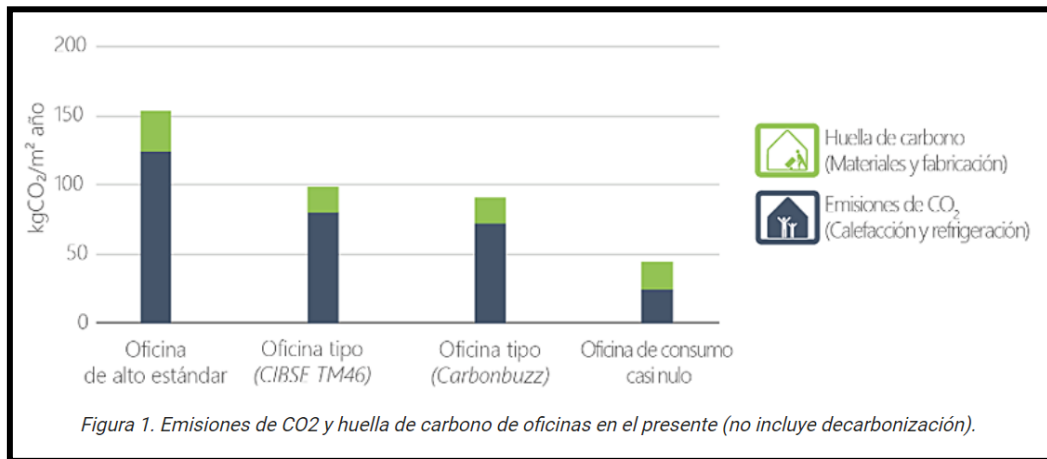


Figura 1. Emisiones de CO₂ y huella de carbono de oficinas (sin incluir decarbonización).

Fuente: <https://www.construible.es/comunicaciones/influencia-del-diseno-arquitectonico-las-emisiones-co2-edificios-oficinas-analisis-cinco-casos-pioneros-reino-unido-suiza>

Nota: se adjunta también un enlace en el que se ofrece un intervalo de consumo eléctrico medio por metro cuadrado útil en un edificio de oficinas:

<https://www.enectiva.cz/es/blog/2015/06/ideas-energia-edificio-de-oficinas/#:~:text=Los%20fines%20de%20semana%2C%20la,5%20Kwh%2Fm%C2%B2%20por%20a%C3%B1o.>

En el ámbito técnico, la solución incluye la utilización de un conjunto de tecnologías, entre las que están:

1. Linux
2. StreamSets
3. MongoDB

Recapitulando, los pasos de la práctica resueltos durante el caso de repaso han sido:

- Introducción teórica al *internet of things* (IoT), protocolos y redes de telecomunicaciones.
- Formato de los dos ficheros que contienen los datos de los cuatro sensores de Valladolid y los cuatro sensores de Sevilla, y que actúan como si fuesen *data loggers* que envían datos a un *topic* de Kafka (en este caso, un fichero en lugar de un *topic* de Kafka).
- Construcción de un *pipeline* de *streamsets* conectado al fichero que actúa como *topic* de Kafka (este fichero recoge los datos enviados desde los dos ficheros de Valladolid y Sevilla), que tras un pequeño procesado de la fecha, unifica los datos procedentes de las dos ciudades en una única colección de MongoDB llamada “datos_sensores” de la BBDD “sensores_IoT”.

Por tanto, ya están cargados todos los datos ingestados desde los sensores en la colección de MongoDB.

La directiva ha enviado una serie de 10 cuestiones para que los analistas de datos respondan, con el objetivo de proporcionar un informe para que se puedan tomar decisiones acerca del estudio.

Estas cuestiones implican la consulta de la colección para obtener diferentes *insights*, así como la realización de inserciones o actualizaciones de la BBDD “sensores_IoT”.

2. RECOMENDACIONES E INSTRUCCIONES

Como recomendación para resolver el caso de negocio, es aconsejable crear una colección de prueba que contenga, por ejemplo, cuatro documentos de Valladolid y cuatro de Sevilla. De esta manera, si se aplica comando erróneo y se actualiza lo que no se quiere en la colección, simplemente habrá que hacer un **drop()** de la colección y volver a crearla ejecutando la sentencia *insert* que se presenta a continuación.

Como ejemplo de la colección de prueba, se adjunta esta con los ocho primeros documentos del *dataset* ingestado en la colección “datos_sensores” mediante el flujo de *streamsets*.

```
db.test.insert([{"timestamp":"2020-07-01T00:00:00Z","sensor_id":1,"location_id":2,"medidas":[{"tipo_medida":"Temperatura","valor":22.08,"unidad":"°C"}, {"tipo_medida":"Humedad_relativa","valor":34.92,"unidad":"%"}]}, {"timestamp":"2020-07-01T00:00:00Z","sensor_id":2,"location_id":2,"medidas":[{"tipo_medida":"Emision_CO2","valor":2.055,"unidad":"gCO2/m2"}, {"tipo_medida":"Consumo_electrico","valor":0.00269,"unidad":"kWh/m2"}]}, {"timestamp":"2020-07-01T00:15:00Z","sensor_id":1,"location_id":2,"medidas":[{"tipo_medida":"Temperatura","valor":21.12,"unidad":"°C"}, {"tipo_medida":"Humedad_relativa","valor":37.7,"unidad":"%"}]}, {"timestamp":"2020-07-01T00:15:00Z","sensor_id":2,"location_id":2,"medidas":[{"tipo_medida":"Emision_CO2","valor":2.102,"unidad":"gCO2/m2"}, {"tipo_medida":"Consumo_electrico","valor":0.00272,"unidad":"kWh/m2"}]}, {"timestamp":"2020-07-01T00:00:00Z","sensor_id":1,"location_id":1,"medidas":[{"tipo_medida":"Temperatura","valor":16.61,"unidad":"°C"}, {"tipo_medida":"Humedad_relativa","valor":83.74,"unidad":"%"}]}],
```

```
{ "timestamp": "2020-07-01T00:00:00Z", "sensor_id": 2, "location_id": 1, "medidas": [ { "tipo_medida": "Emision_CO2", "valor": 1.572, "unidad": "gCO2/m2" }, { "tipo_medida": "Consumo_electrico", "valor": 0.00188, "unidad": "kWh/m2" } ] },
{ "timestamp": "2020-07-01T00:15:00Z", "sensor_id": 1, "location_id": 1, "medidas": [ { "tipo_medida": "Temperatura", "valor": 15.75, "unidad": "°C" }, { "tipo_medida": "Humedad_relativa", "valor": 83.08, "unidad": "%" } ] },
{ "timestamp": "2020-07-01T00:15:00Z", "sensor_id": 2, "location_id": 1, "medidas": [ { "tipo_medida": "Emision_CO2", "valor": 1.626, "unidad": "gCO2/m2" }, { "tipo_medida": "Consumo_electrico", "valor": 0.00146, "unidad": "kWh/m2" } ] } ] }
```

3. CUESTIONES SOBRE EL CASO DE NEGOCIO MONGODB

CASO 1: crear una nueva colección llamada “sensores” e introducir en ella cuatro documentos, correspondientes a cada uno de los sensores que se encuentran instalados en este momento: dos en Sevilla y dos en Valladolid.

La colección deberá contener las siguientes claves:

- Ubicacion: tipo *string*, indica la ciudad en la que está instalado el sensor.
- “medidas_sensor”: *array* de objetos JSON. Cada JSON tendrá dos subclaves: “tipo_medida”, *string* que identifica lo que mide el sensor; “unidad”: *string* que identifica las unidades en las que mide el sensor cada tipo de medida.
- Coordenadas: *array* de numéricos. Los sensores de Valladolid tienen las coordenadas [41.638597, -4.740186] y los de Sevilla [37.409311, -5.949939].
- “fecha_instalación”: *timestamp*. Los dos sensores de Valladolid se instalaron el día 25 de mayo 2020 a las 10:30 AM, mientras que los de Sevilla se instalaron el 28 de mayo 2020 a las 11:30 AM.

- “location_id”: numérico que indica el ID de la ubicación a la que pertenece el sensor. Valor uno para los sensores de Valladolid y valor dos para los sensores de Sevilla.
- “tipo_sensor”: numérico que indica el tipo de sensor. Valor uno para el tipo de sensor que mide temperatura y humedad relativa. Valor dos para el tipo de sensor que mide emisión de CO2 y consumo eléctrico.

La colección se creará automáticamente al insertar los cuatro documentos en una misma consulta. Adjuntar captura de la consulta de inserción, así como el resultado tras ejecutarla.

CASO 2: mostrar el número de datos recogidos por el sensor que mide la temperatura en Valladolid, así como el número de datos recogidos por el de Sevilla entre los días 1 y 10 julio (incluido el día 10 entero).

Haciendo cuentas, se reciben cuatro datos por hora, 96 por día, y, por tanto, en 10 días debería haber 960 datos enviados por cada sensor. Comentar si no se han recibido datos de temperatura en algún intervalo de 15 minutos para alguno de los sensores.

CASO 3: se pide identificar si hay algún documento que pueda distorsionar los resultados del estudio. Por ello, se quiere identificar posibles valores erróneos de temperatura enviados por el sensor.

Se pide, por tanto, en primer lugar, identificar los documentos de la colección “datos_sensores” que tengan una temperatura superior a 55 °C, tanto para Valladolid como para Sevilla (contar el número de casos en cada ciudad y mostrar también los documentos erróneos).

A la hora de mostrar los documentos con errores, solamente se devolverán las claves: *timestamp*, “location_id” y de la clave medidas mostrar solo el objeto correspondiente a la temperatura, sin mostrar el objeto de humedad relativa.

Una vez se hayan identificado, se procede a eliminar estos documentos de la colección, es decir: para ese *timestamp* solo quedará el documento correspondiente a la emisión de CO2 y el consumo eléctrico.

Además de las anteriores, se deberá adjuntar también una captura que demuestre que se han eliminado correctamente dichos documentos, si es que existiera alguno.

CASO 4: buscar el valor mínimo de temperatura en Sevilla. Se considera que es un valor poco realista para Sevilla y que es necesario multiplicarlo por un factor 1,2 para hacerlo un valor algo más real.

Nota: una vez hallado el ID del objeto a actualizar, investigar el funcionamiento de la función **\$mul** para comprobar si puede utilizarse en este caso para actualizar el valor o es necesario utilizar otro modificador.

Se deberá adjuntar una captura de la consulta de búsqueda junto con el documento al que le corresponde la temperatura mínima de Sevilla. Además, también se adjuntará la consulta de actualización y, por último, se mostrará todo el documento ya actualizado.

CASO 5: obtener los tres máximos valores de consumo eléctrico en un día de fin de semana en ambas ciudades y comparar sus valores (se considera fin de semana a partir de las 16:00 del viernes y hasta las 7:45 del lunes).

Devolver para cada ciudad un documento con el *timestamp* en que se producen esos máximos, el día de la semana y la hora a las que corresponden esos máximos y el valor del consumo eléctrico (subclave “tipo_medida” “Consumo_electrico” + subclave “valor” + subclave “unidad”, es decir: no se quieren obtener los valores de ese día de emisión de CO2).

CASO 6: se considera como perjudicial para la salud cualquier día en que el acumulado de las emisiones de CO₂ durante el día completo supere los 420 g CO₂/m². De los 14 días que comprende nuestro estudio, se pretende recuperar cuántos días se superó ese límite para Valladolid y, por otro lado, cuántos días se superó para Sevilla.

Se deben adjuntar tanto la captura del número de veces que se supera el límite para cada ciudad (clave “días_sobre_limite_permitido”), así como otra captura que muestre todos los documentos de Valladolid, por un lado, agrupados por la clave “dia_mes” y el acumulado total de emisión CO₂ bajo la clave “suma_Emision_CO2”, ordenados de mayor a menor emisión.

Presentar las mismas dos capturas para Sevilla y comentar los resultados. ¿Qué tienen los cuatro días de ambas ciudades en los que la emisión de CO₂ es más pequeña?

Las consultas serán idénticas para Valladolid y Sevilla, distinguiéndose únicamente por el filtrado sobre la clave “location_id”.

CASO 7: obtener la media de emisiones de CO2 de cada ciudad por separado en la hora punta de personas en la oficina, que se considera las 10 AM. Por tanto, se obtendrán por un lado los registros con hora 10 en el *timestamp*, que midan CO2 y que sean de Valladolid y obtener la media de emisiones de CO2 durante los 14 días en esa hora como la clave “Avg_Emission_CO2” (recordar que cada hora se reciben cuatro valores desde el sensor para cada ciudad). Mostrar los resultados ordenados por la clave “Avg_Emission_CO2”, mostrando como primer documento la media más baja. Redondear esta clave a dos decimales. Realizar la misma operativa con Sevilla.

CASO 8: Se ha descubierto que el sensor de temperatura de Valladolid mide 1,5 °C de más. Por ello, se pide actualizar todos los valores correspondientes a este sensor decrementando el valor de la temperatura en 1,5 °C.

Antes de realizar la actualización, se ordenarán mostrando primero el de mayor temperatura, y muestra los dos primeros documentos del sensor de Valladolid con la mayor temperatura. Solo mostrar las siguientes claves: *timestamp*, "location_id". Del *array* de medidas solo mostrar el primer ítem, por ejemplo: **"medidas" : [{ "tipo_medida" : "Temperatura", "valor" : 15.75, "unidad" : "°C" }] }**. No incluir el **ObjectId**)

Realizar el mismo proceso una vez que has actualizado los valores y realiza capturas de los dos documentos anteriores para comprobar que se han actualizado.

CASO 9: se quieren analizar los porcentajes de humedad relativa en horario laboral (8-18:00) de ambas ciudades. Recuperar por un lado los cinco documentos con los valores mínimos de humedad relativa en Sevilla y los cinco documentos con humedad relativa mínima en Valladolid por separado (ambos en horario laboral).

Sólo se quieren recuperar de estos documentos el *timestamp* y la subclave **`medidas.tipo_medida = Humedad_relativa`**, junto con el valor y la unidad asociados.

Analizar si los mínimos se producen siempre en la misma franja horaria (mismo intervalo de 1-2 horas, o es variable). Comentar si varían de una ciudad a otra.

CASO 10: se quieren actualizar todos los documentos para introducir en el *array* de medidas dos nuevos elementos que van a ser constantes y que van a tener el valor del precio del kWh y de la superficie total de la sede.

Solamente se añadirán a los documentos que tengan como medida el consumo eléctrico (los que tienen temperatura y humedad no).

- Los ítems a introducir son para Valladolid:

```
{"precio_kWh":0.102,"unidad":"€/kWh"}, {"superficie":450,"unidad":"m2"}
```

- Los ítems a introducir son para Sevilla:

```
{"precio_kWh":0.107,"unidad":"€/kWh"}, {"superficie":550,"unidad":"m2"}
```

Para comprobar que se ha actualizado correctamente la colección, muestra cuatro documentos de Sevilla y cuatro de Valladolid, correspondientes a los *timestamps*: **2020-07-01T08:00:00Z**, **2020-07-01T23:15:00Z**. Cada intervalo se envían dos documentos desde cada ciudad: **(Temperatura y Humedad_relativa) + (Emision_CO2 y Consumo_electrico)**.