Portada

Integrantes del grupo:

- Rubén Gómez Olivencia

Índice

- 1. Introducción
- 2. Obtención de los datos
  - 2.1. Descargar el dataset
  - 2.2. Definición del dataset
- 3. Diseño de la base de datos
  - 3.1. Diseño conceptual
  - 3.2. Diseño físico
    - 3.2.1. Elección del modelo
    - 3.2.2. Crear las colecciones
- 4. Despliegue de la base de datos
  - 4.1. Conexión a servidores
    - 4.1.1. Conexión local
    - 4.1.2. Conexión remota
  - 4.2. Creación de base de datos y colección
    - 4.2.1. Creación en local
    - 4.2.2. Creación en remoto
  - 4.3. Despliegue de datos
    - 4.3.1. Despliegue en local
    - 4.3.2. Despliegue en remoto
- 5. Consultas a base de datos
  - 5.1. Consultas básicas
    - 5.1.1. Mostrar todas las sidrerías que existen en el País Vasco
    - 5.1.2. Asadores que hay en bilbao
  - 5.2. Consultas avanzadas
    - 5.2.1. Buscar restaurantes que oferten menú vegetariano
    - 5.2.2. Búsqueda geoespacial de restaurantes a 2 kilómetros de una posición dada
- 6. MongoDB Charts
  - 6.1. Restaurantes, Sidrerías y Asadores
  - 6.2. Sidrerías por territorio
  - 6.3. Restaurantes con estrellas Michelin
- 7. Conclusiones

1. Introducción

Queremos conocer los distintos tipos de restaurantes, asadores y sidrerías que existen en Euskadi para poder realizar actividades grupales para ir a comer a estos restaurantes. Es interesante diferenciarlos por el tipo de restaurantes que son, para así poder realizar un sorteo de a qué tipo de restaurante ir.

Dado que pueden existir personas que son **vegetarianas**, también resulta decisivo poder identificar los lugares en los que se ofertan menús de este tipo.

Por último, también va a interesar qué locales permiten realizar **bodas** para de esta manera poder ofrecerlo a los clientes.

2. Obtención de los datos

Para obtener los datos de los restaurantes, asadores y sidrerías que existen en Euskadi se ha utilizado la página web creada por el Gobierno Vasco llamada "Open Data Euskadi" cuya URL es la siguiente https://www.opendata.euskadi.eus/inicio/. Se ha utilizado este portal dado que es una fuente de datos oficial y que contiene datos fidedignos de información turística como la que nos interesa.

Dentro de este portal de transparencia existe mucha información como la que nos interesa, que concratamente podemos visitar en el siguiente enlace. En esta web vamos a poder ver que podemos descargarnos los datos en distintos formatos, siendo el que nos interesa el formato JSON. Concretamente, la URL completa del fichero JSON es la siguiente: https://www.opendata.euskadi.eus/contenidos/ds_recursos_turisticos/restaurantes_asador_sidr con el que obtenemos un fichero llamado **restaurantes.json**.

2.1. Descargar el dataset

Ahora que sabemos la URL del fichero es momento de descargar el dataset original para poder hacer uso de él.

Lo primero que hay que hacer es cargar las librerías necesarias:

- **requests**: Librería HTTP para realizar peticiones.
- **json**: Librería para hacer uso de ficheros JSON.

In [1]:
```
# Se usa la librería requests, para hacer peticiones a webs
import requests as re
# Importar Librería json de Python porque se va a descargar fichero en form
import json
```

Creamos una variable con la URL original del fichero y realizamos la petición de descarga:

In [2]:
```
# Se indica la url del fichero de la página web
url = 'https://www.opendata.euskadi.eus/contenidos/ds_recursos_turisticos/r
```

```
# Se realiza la petición
respuesta = re.get(url)
```

Tras realizar la petición, es el momento de guardar los datos en un fichero:

In [3]:
```
# Guardar el fichero de la página web en local (en nuestro pc)
with open('./fichero_datos.json', 'wb') as fichero:
    fichero.write(respuesta.content)
# No hace falta, pero no está de más.
fichero.close()
```

Con el dataset descargado, ya podemos proseguir al siguiente paso.

2.2. Definición del dataset

Una vez descargado el fichero, podemos observar que el dataset obtenido contiene 58 campos para cada uno de los restaurantes que existen en la información obtenida de Open Data Euskadi. Tras realizar un análisis del fichero, vemos que muchos de los campos están vacios o que no nos interesan para nuestro objetivo (ya que es un dataset genérico reutilizado por varios apartados de Open Data), y por tanto nos vamos a quedar con el siguiente conjunto de datos del fichero original. Los campos que nos interesan son:

- **documentName**: Este es el nombre del campo que contiene el nombre del restaurante.
- **turismDescription**: Descripción del restaurante.
- **restorationType**: Tipo de local: si es restaurante, asador o sidrería.
- **phone**: Teléfono de contacto del establecimiento.
- **tourismEmail**: E-mail de contacto.
- **web**: Página web del restaurante.
- **address**: Dirección donde está situado.
- **postalCode**: Código postal
- **municipality**: Municipio en el que se sitúa el restaurante.
- **territory**: Territorio dentro de la Comunidad Autónoma del País Vasco.
- **capacity**: Capacidad permitida.
- **michelinStar**: Si el restaurante cuenta con alguna estrella Michelin.
- **latwgs84**: Latitud de la geoposición donde está situado.
- **lonwgs84**: Longitud de la geoposición donde está situado.

Con este conjunto de datos podremos obtener la información que nos interesa.

3. Diseño de la base de datos

Tras analizar el fichero y determinar cuál es el conjunto de datos que nos interesa, vamos a realizar el diseño de la base de datos que utilizaremos posteriormente para usarlo en MongoDB.

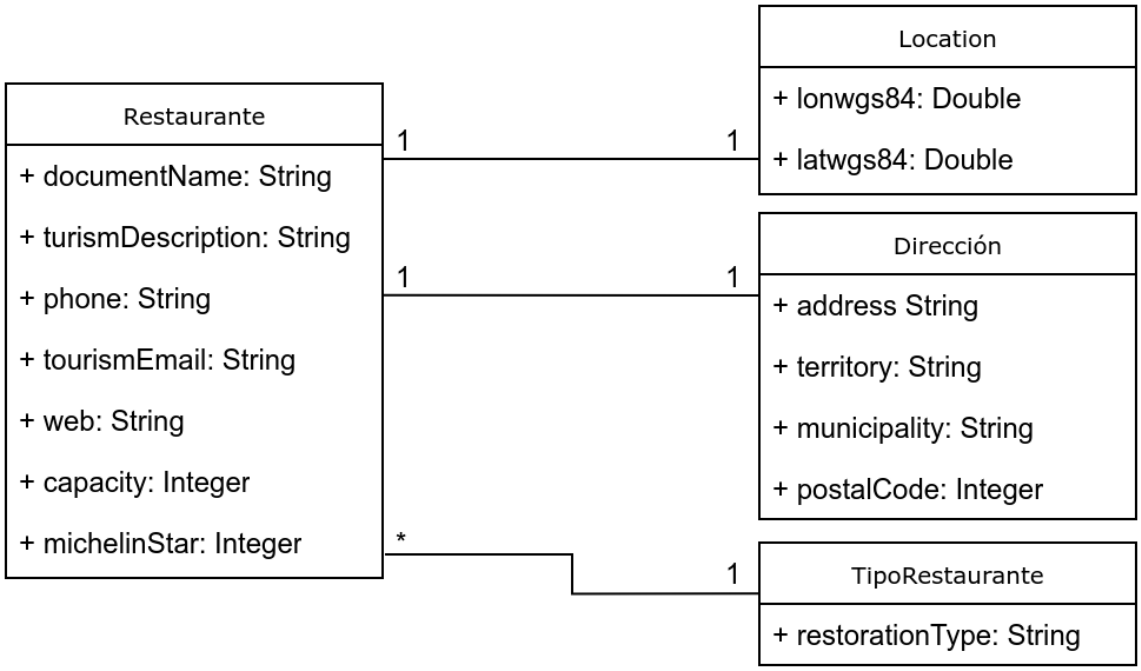
3.1.Diseño conceptual

Con los datos que nos interesa, vamos a realizar el diseño conceptual de los mismos mediante el **lenguaje unificado de modelado** (UML).

Analizando los datos podemos observar que existen cuatro objetos:

- **Restaurante**: El objeto restaurante, con cierta información acerca del mismo.
- **Location**: El objeto que indica una geo-posición.
- **Dirección**: El objeto de tipo dirección.
- **TipoRestaurante**: El objeto de los tipos de restaurante que existen.

Con estos cuatro objetos el diseño conceptual quedaría de la siguiente forma:



Con esto ya tendríamos el modelado de datos creado y pasaremos al siguiente paso que es el diseño físico de los mismos.

3.2. Diseño físico

Tras realizar el diseño conceptual, es el momento de realizar el diseño físico para posteriormente realizar el despliegue en la base de datos.

3.2.1 Elección del modelo

A pesar del diseño conceptual creado previamente, en el que hemos diferenciado cuatro objetos separados, se ha decidido optar por un **modelo flexible** a la hora de realizar el diseño físico de la base de datos.

Este modelo flexible nos va a permitir tener todos los datos embebidos dentro de un mismo documento que a la hora de realizar la búsqueda de datos nos dará **mayor rendimiento**.

Por todo ello, se unificarán los cuatro objetos dentro de un mismo documento que contendrá los 14 campos que hemos ido analizando a lo largo de este documento.

3.2.2. Crear las colecciones

Teniendo en cuenta el paso anterior, en el que los datos de cada restaurante se contendrá en un único documento, se ha decidido crear una única colección en la que contener todos los documentos de todos los restaurantes.

Dicha colección se llamará **restaurantes** que estará integrada dentro de la base de datos **salidas_grupales**.

Para crear la colección tendremos que realizar los siguientes pasos:

```
use salidas_grupales
db.createCollection("restaurantes")
```

Tal como se puede ver, son los comandos necesarios para hacer uso de una base de datos (que en caso de no existir se crea) y la creación de una colección con el nombre **restaurantes** elegido.

Estos mismos pasos están añadidos dentro del fichero **fisico.js** que va adjunto a este documento.

4. Despliegue de la base de datos

Tras realizar todo el análisis previo es el momento de realizar el despliegue en la base de datos. El sistema gestor de base de datos elegido es MongoDB. MongoDB es un sistema de base de datos **NoSQL** y orientado a documentos.

Para la realización de este despliegue se han utilizado dos servidores:

- **Servidor local**: Desplegado a través de un contenedor **Docker** exponiendo los puertos necesarios.
- **Servidor remoto**: Servidor desplegado en la nube de Amazon (AWS) a través del proveedor MongoDB Atlas

A continuación se explica cómo realizar las conexiones a ambos servidores.

4.1. Conexión a servidores

En este apartado se va a explicar cómo realizar la conexión a servidores MongoDB a través del lenguaje de programación Python. Antes de realizar la conexión es necesario tener la librería **pymongo** instalada en nuestro sistema. Para ello realizaremos la instalación mediante el comando de sistema `pip`.

In [4]:
```
# Para realizar la conexión necesitamos de la API de python.
!pip install pymongo
```

```
Defaulting to user installation because normal site-packages is not writeab
le
Requirement already satisfied: pymongo in /home/yuki/.local/lib/python3.10/
site-packages (3.12.3)
WARNING: There was an error checking the latest version of pip.
```

In [5]:
```python
# importamos la librería para poder hacer uso de ella:
import pymongo
```

4.1.1. Conexión local

En lugar de realizar una instalación al uso, se ha decidido realizar la instalación del servidor a través de un contenedor **Docker**, haciendo uso de **docker-compose** para el contenedor con el SGBD. Para ello se ha utilizado un fichero **compose.yaml** con el siguiente contenido de despliegue:

```yaml
services:
  mongo:
    image: mongo:5
    environment:
      MONGO_INITDB_ROOT_USERNAME: root
      MONGO_INITDB_ROOT_PASSWORD: example
    ports:
      - 27017:27017
    restart: always
```

Si nuestro sistema está correctamente configurado, al realizar `docker-compose up` levantará el contenedor con el servidor que hemos configurado:

- **MongoDB** versión 5 escuchando en el puerto 27017 del anfitrión.

Tras esto tenemos que formar la URI de conexión indicando:

- **Protocolo de conexión**: El protocolo que utilizaremos para realizar la conexión, en este caso **mongodb://**.
- **Usuario de conexión**: A la hora de realizar la conexión, el usuario que vamos a utilizar. En nuestro caso **root**.
- **Contraseña del usuario**: La contraseña necesaria para que el usuario pueda realizar la conexión.
- **Host**: Servidor al que realizaremos la conexión. En este caso, al ser localhost: **127.0.0.1**.
- **Opciones**: Opciones extra para que reintente las escrituras.

Y posteriormente realizaremos la conexión con la función "MongoClient" de pymongo.

In [6]:
```python
# protocol: protocolo de acceso
protocol = "mongodb://"

# user: nombre de usuario de la base de datos
user = "root"

# password: contraseña del usuario
password = "example"

# host: server name
host = "127.0.0.1"
```

```
# options: opciones extra de la conexión
options = "/?retryWrites=true&w=majority"

# formamos la URI de conexión al servidor local
uri_mongo = protocol + user + ":" + password + "@" + host + options

# Conectar al servicio MongoDB
con_local = pymongo.MongoClient(uri_mongo)
```

Con esto ya tendríamos la conexión al servidor local realizada.

4.1.2. Conexión remota

Para la conexión remota se va a utilizar el proveedor Mongo Atlas, donde hemos creado una cuenta y se ha realizado el despliegue de una instancia en los servidores de Amazon (AWS).

De manera similar a la conexión local, para poder realizar la conexión al servidor remoto debemos conocer los siguientes datos:

Tras esto tenemos que formar la URI de conexión indicando:

- **Protocolo de conexión**: El protocolo que utilizaremos para realizar la conexión, en este caso **mongodb+srv://**.
- **Usuario de conexión**: A la hora de realizar la conexión, el usuario que vamos a utilizar. En nuestro caso **root**.
- **Contraseña del usuario**: La contraseña necesaria para que el usuario pueda realizar la conexión.
- **Host**: Servidor al que realizaremos la conexión, que es un servidor remoto.
- **Opciones**: Opciones extra para que reintente las escrituras.

Y tras esto realizaremos la conexión al servidor remoto.

In [7]:
```
# protocol: protocolo de acceso
protocol = "mongodb+srv://"

# user: nombre de usuario de la base de datos
user = "root"

# password: contraseña del usuario
password = "example"

# host: server name
host = "cluster0.odzyc.mongodb.net"

# options: opciones extra de la conexión
options = "/?retryWrites=true&w=majority"

# formamos la URI de conexión al servidor local
uri_mongo = protocol + user + ":" + password + "@" + host + options

# Conectar al servicio MongoDB
con_remoto = pymongo.MongoClient(uri_mongo)
```

4.2. Creación de base de datos y colección

En este apartado vamos a hacer uso del diseño físico que hemos creado en apartados anteriores para desplegarlo en los dos servidores que tenemos actualmente. Para ello, y tal como se ha dicho previamente, se va hacer uso del **modelo flexible**.

4.2.1. Creación en local

Tal como se ha dicho previamente, vamos a crear la base de datos **salidas_grupales** y dentro una colección llamada **restaurantes** donde posteriormente crearemos los documentos:

In [8]:
```
# creamos la base de datos
db_local = con_local.salidas_grupales
# creamos la colección
collection_local = db_local.restaurantes
```

4.2.2. Creación en remoto

Al igual que en local, creamos la base de datos y la colección en el servidor remoto:

In [9]:
```
# creamos la base de datos
db_remoto = con_remoto.salidas_grupales
# creamos la colección
collection_remoto = db_remoto.restaurantes
```

4.3. Despliegue de datos

Ahora es el momento de realizar el despliegue de datos dentro de la colección, con los datos que hemos obtenido y con el dataset que nos interesa tras realizar el análisis del dataset original.

Al igual que los pasos que hemos realizado en este apartado, realizaremos el despliegue en los dos servidores: local y remoto.

Para no repetir código, se va a hacer de una variable donde guardar los inserts que se van a realizar en ambos servidores:

In [10]:
```
inserts = [{"documentName": "1860 Tradición", "turismDescription": "<p>El r
```

4.3.1. Despliegue en local

Para realizar la inserción en local se va a hacer uso de las variables previamente creadas, para insertar los datos en la colección creada:

In [11]:
```
collection_local.insert_many(inserts)
```

Out[11]:
```
<pymongo.results.InsertManyResult at 0x7f91cb196780>
```

Con esto ya tendremos los datos en la base de datos local.

4.3.2. Despliegue en remoto

Para el servidor remoto realizamos el mismo paso, pero usando la variable para la colección remota:

In [12]:
```
collection_remoto.insert_many(inserts)
```

Out[12]:
```
<pymongo.results.InsertManyResult at 0x7f919c42da40>
```

Y de nuevo, ya tendremos los datos insertados en el servidor remoto.

5. Consultas a base de datos

Una vez realizado el despliegue en los servidores, junto con la inserción de los datos, es el momento de realizar las consultas necesarias para obtener los datos. Las consultas se van a realizar sobre el servidor local, pero dado que en ambos servidores los datos son idénticos, el resultado debe ser el mismo si se utiliza el servidor remoto.

5.1. Consultas básicas

A continuación se van a detallar dos consultas básicas que son necesarias realizar para nuestro interés.

5.1.1. Mostrar todas las sidrerías que existen en el País Vasco

Una de las consultas más habituales, sobre todo en la época de febrero, es saber las sidrerías que existen para poder organizar una comida al comienzo de la temporada. Es por ello que a continuación se explican los detalles:

- **Descripción de la consulta**: Queremos consultar las sidrerías que existen en el País vasco.
- **Resultado esperado**: Se espera un listado de los nombres, dirección completa (con código postal, municipio y territorio) y teléfono de cada sidrería que existe en el País Vasco.

La consulta ejecutada directamente sobre **MongoDB** sería:

```
use salidas_grupales
db.restaurantes.find({"restorationType" : "Sidreria"},
{"documentName":1,"address":1,"postalCode":1,"municipality":1,"territ
```

De esta manera obtendríamos los datos. A continuación se muestra el código necesario para realizar la consulta en Python y la salida de los datos obtenidos:

In [13]:
```
for sidreria in collection_local.find({"restorationType" : "Sidreria"},{"do
    print(sidreria)
```

{'documentName': 'ARAIA SAGARDOTEGIA', 'phone': 945304763, 'address': 'Call
e Santsaerreka, 26 - bajo', 'postalCode': 1250, 'municipality': 'Asparren
a', 'territory': 'Araba/Álava'}
{'documentName': 'Aburuza', 'phone': 943692452, 'address': 'Barrio Goibur
u', 'postalCode': 20150, 'municipality': 'Aduna', 'territory': 'Gipuzkoa'}
{'documentName': 'Aginaga', 'phone': 943366710, 'address': 'Barrio Aginaga,
s/n', 'postalCode': 20170, 'municipality': 'Usurbil', 'territory': 'Gipuzko
a'}
{'documentName': 'Akarregi', 'phone': 943330713, 'address': 'Barrio Akarreg
i, 5', 'postalCode': 20120, 'municipality': 'Hernani', 'territory': 'Gipuzk
oa'}
{'documentName': 'Akelenea', 'phone': 943333333, 'address': 'Camino Oialum
e, 57', 'postalCode': 20115, 'municipality': 'Astigarraga', 'territory': 'G
ipuzkoa'}
{'documentName': 'Algorri', 'phone': 943865617, 'address': 'Barrio de Santi
ago, 1', 'postalCode': 20750, 'municipality': 'Zumaia', 'territory': 'Gipuz
koa'}
{'documentName': 'Alorrenea', 'phone': 943336999, 'address': 'Camino Alorre
ne, 4', 'postalCode': 20115, 'municipality': 'Astigarraga', 'territory': 'G
ipuzkoa'}
{'documentName': 'Altuna', 'phone': 943554917, 'address': 'Caserío Galarrag
a, barrio Lategi', 'postalCode': 20130, 'municipality': 'Urnieta', 'territo
ry': 'Gipuzkoa'}
{'documentName': 'Altzueta', 'phone': 943551502, 'address': 'Osinaga, 7',
'postalCode': 20120, 'municipality': 'Hernani', 'territory': 'Gipuzkoa'}
{'documentName': 'Amaike', 'phone': 944832667, 'address': 'Doctor José Zald
ua, 31', 'postalCode': 48920, 'municipality': 'Portugalete', 'territory':
'Bizkaia'}
{'documentName': 'Amebi', 'phone': 943162523, 'address': 'Zubierreka, 9',
'postalCode': 20210, 'municipality': 'Lazkao', 'territory': 'Gipuzkoa'}
{'documentName': 'Ander', 'phone': 944476666, 'address': 'Andalucia, 1', 'p
ostalCode': 48015, 'municipality': 'Bilbao', 'territory': 'Bizkaia'}
{'documentName': 'Ankapalu Berria', 'phone': 944558197, 'address': 'Logroño
Behekoa, s/n', 'postalCode': 48195, 'municipality': 'Larrabetzu', 'territor
y': 'Bizkaia'}
{'documentName': 'Aristizabal', 'phone': 943492714, 'address': 'Txalaka bid
ea, 4 (Iturriotz)', 'postalCode': 20180, 'municipality': 'Oiartzun', 'terri
tory': 'Gipuzkoa'}
{'documentName': 'Armentegi', 'phone': 945132101, 'address': 'Mendibe, 4',
'postalCode': 1007, 'municipality': 'Vitoria-Gasteiz', 'territory': 'Araba/
Álava'}
{'documentName': 'Arratzain', 'phone': 943366663, 'address': 'Kale Zahar Au
zoa, 21', 'postalCode': 20170, 'municipality': 'Usurbil', 'territory': 'Gip
uzkoa'}
{'documentName': 'Arriaga', 'phone': 944165670, 'address': 'Santa María, 1
3', 'postalCode': 48005, 'municipality': 'Bilbao', 'territory': 'Bizkaia'}
{'documentName': 'Astarbe', 'phone': 943551527, 'address': 'Txoritokieta bi
dea, 13', 'postalCode': 20115, 'municipality': 'Astigarraga', 'territory':
'Gipuzkoa'}
{'documentName': 'Aulia', 'phone': 943806066, 'address': 'Barrio Guadalupe
12', 'postalCode': 20250, 'municipality': 'Legorreta', 'territory': 'Gipuzk
oa'}
{'documentName': 'Axpe', 'phone': 946168285, 'address': 'Caserío Axpe, barr
io Atxondo', 'postalCode': 48270, 'municipality': 'Markina-Xemein', 'territ
ory': 'Bizkaia'}
{'documentName': 'Ayoberri', 'phone': 946764257, 'address': 'Carretera de B
utrón, barrio Mendiondo', 'postalCode': 48610, 'municipality': 'Urduliz',
'territory': 'Bizkaia'}
{'documentName': 'Añota', 'phone': 943812092, 'address': 'Barrio Elosiaga,
s/n', 'postalCode': 20730, 'municipality': 'Azpeitia', 'territory': 'Gipuzk
oa'}
{'documentName': 'Baleio', 'phone': 943491340, 'address': 'Ananedes Bidea.C
aserío Baleione. Iturrioz', 'postalCode': 20180, 'municipality': 'Oiartzu
n', 'territory': 'Gipuzkoa'}

{'documentName': 'Barkaiztegi', 'phone': 943451304, 'address': 'Paseo de Barkaiztegi, 42, Martutene', 'postalCode': 20014, 'municipality': 'Donostia / San Sebastián', 'territory': 'Gipuzkoa'}
{'documentName': 'Beharri', 'phone': 943431631, 'address': 'Calle Narrica, 22', 'postalCode': 20003, 'municipality': 'Donostia / San Sebastián', 'territory': 'Gipuzkoa'}
{'documentName': 'Bereziartua', 'phone': 943555798, 'address': 'Camino de Iparralde, 16', 'postalCode': 20115, 'municipality': 'Astigarraga', 'territory': 'Gipuzkoa'}
{'documentName': 'Calonge', 'phone': 943213251, 'address': 'Paseo Padre Orkolaga, 8', 'postalCode': 20008, 'municipality': 'Donostia / San Sebastián', 'territory': 'Gipuzkoa'}
{'documentName': 'Eguzkitza', 'phone': 943672613, 'address': 'Barrio Usabal, 25', 'postalCode': 20400, 'municipality': 'Tolosa', 'territory': 'Gipuzkoa'}
{'documentName': 'El Cartero', 'phone': 946806674, 'address': 'Bº Molinar, 20', 'postalCode': 48891, 'municipality': 'Karrantza Harana/Valle de Carranza', 'territory': 'Bizkaia'}
{'documentName': 'Elexalde', 'phone': 944571614, 'address': 'Barrio Elexalde s/n', 'postalCode': 48960, 'municipality': 'Galdakao', 'territory': 'Bizkaia'}
{'documentName': 'Elorrabi', 'phone': 943336990, 'address': 'Osinaga bailara, 13', 'postalCode': 20120, 'municipality': 'Hernani', 'territory': 'Gipuzkoa'}
{'documentName': 'Elutxeta', 'phone': 943556981, 'address': 'Caserío Elutxeta, 34', 'postalCode': 20130, 'municipality': 'Urnieta', 'territory': 'Gipuzkoa'}
{'documentName': 'Etxe Zuri', 'phone': 943882049, 'address': 'Barrio Erreka lde', 'postalCode': 20212, 'municipality': 'Olaberria', 'territory': 'Gipuzkoa'}
{'documentName': 'Etxeberria', 'phone': 943555697, 'address': 'Sagardotegizeharra, 5', 'postalCode': 20115, 'municipality': 'Astigarraga', 'territory': 'Gipuzkoa'}
{'documentName': 'Eula', 'phone': 943552744, 'address': 'Barrio Lategi, 19', 'postalCode': 20130, 'municipality': 'Urnieta', 'territory': 'Gipuzkoa'}
{'documentName': 'Galtzagorri', 'phone': 944116111, 'address': 'Juan de la Cosa, 20', 'postalCode': 48004, 'municipality': 'Bilbao', 'territory': 'Bizkaia'}
{'documentName': 'Gartziategi', 'phone': 943469674, 'address': 'Paseo de Martutene, 139', 'postalCode': 20115, 'municipality': 'Astigarraga', 'territory': 'Gipuzkoa'}
{'documentName': 'Gaztañaga', 'phone': 943591968, 'address': 'Barrio Buruntza', 'postalCode': 20140, 'municipality': 'Andoain', 'territory': 'Gipuzkoa'}
{'documentName': 'Goiko Lastola', 'phone': 943553272, 'address': 'Barrio Ereñotzu, 89', 'postalCode': 20120, 'municipality': 'Hernani', 'territory': 'Gipuzkoa'}
{'documentName': 'Gorozika', 'phone': 946309150, 'address': 'Bº Gorozika s/n', 'postalCode': 48392, 'municipality': 'Muxika', 'territory': 'Bizkaia'}
{'documentName': 'Gurutzeta', 'phone': 943552242, 'address': 'Oialume Bidea, 63', 'postalCode': 20115, 'municipality': 'Astigarraga', 'territory': 'Gipuzkoa'}
{'documentName': 'Iarritu', 'phone': 945386131, 'address': 'Barrio Arrotegui, 114', 'postalCode': 1470, 'municipality': 'Amurrio', 'territory': 'Araba/Álava'}
{'documentName': 'Ibarra', 'phone': 946731100, 'address': 'Barrio Ibarra, s/n', 'postalCode': 48340, 'municipality': 'Amorebieta-Etxano', 'territory': 'Bizkaia'}
{'documentName': 'Intxaurrondo', 'phone': 943292074, 'address': 'Zubiaurre, 72', 'postalCode': 20015, 'municipality': 'Donostia / San Sebastián', 'territory': 'Gipuzkoa'}
{'documentName': 'Iparragirre', 'phone': 943550328, 'address': 'Barrio Osiñaga', 'postalCode': 20120, 'municipality': 'Hernani', 'territory': 'Gipuzko

a'}
{'documentName': 'Irigoien', 'phone': 943550333, 'address': 'Camino de Iparralde, 12', 'postalCode': 20115, 'municipality': 'Astigarraga', 'territory': 'Gipuzkoa'}
{'documentName': 'Isastegi', 'phone': 943652964, 'address': 'Aldaba Txiki, 15', 'postalCode': 20400, 'municipality': 'Tolosa', 'territory': 'Gipuzkoa'}
{'documentName': 'Itsasburu', 'phone': 943556879, 'address': 'Osiñaga Auzoa, 10', 'postalCode': 20120, 'municipality': 'Hernani', 'territory': 'Gipuzkoa'}
{'documentName': 'Iturrieta', 'phone': 945445385, 'address': 'Barrio Arraga, s/n.', 'postalCode': 1169, 'municipality': 'Aramaio', 'territory': 'Araba/Álava'}
{'documentName': 'Izeta', 'phone': 943131693, 'address': 'Barrio Urdaneta', 'postalCode': 20809, 'municipality': 'Aia', 'territory': 'Gipuzkoa'}
{'documentName': 'Katxarro', 'phone': 946550279, 'address': 'San Fausto Kalea, 17', 'postalCode': 48230, 'municipality': 'Elorrio', 'territory': 'Bizkaia'}
{'documentName': 'Kupeltegi', 'phone': 945130627, 'address': 'Bekolasa, 1', 'postalCode': 1015, 'municipality': 'Vitoria-Gasteiz', 'territory': 'Araba/Álava'}
{'documentName': 'La Sidre', 'phone': 945131565, 'address': 'Larrintzar, 3.', 'postalCode': 1195, 'municipality': 'Vitoria-Gasteiz', 'territory': 'Araba/Álava'}
{'documentName': 'Laka-Erdi', 'phone': 946139178, 'address': 'Alto de Millioi (Ctra. Markina - Lekeitio)', 'postalCode': 48710, 'municipality': 'Berriatua', 'territory': 'Bizkaia'}
{'documentName': 'Larrarte', 'phone': 943555647, 'address': 'Muñagorri Enea Baserria, s/n', 'postalCode': 20155, 'municipality': 'Astigarraga', 'territory': 'Gipuzkoa'}
{'documentName': 'Larre-Gain', 'phone': 943555846, 'address': 'Barrio Ereñozu, 58', 'postalCode': 20120, 'municipality': 'Hernani', 'territory': 'Gipuzkoa'}
{'documentName': 'Lezama', 'phone': 944556368, 'address': 'Barrio Garaioltza, 146', 'postalCode': 48196, 'municipality': 'Lezama', 'territory': 'Bizkaia'}
{'documentName': 'Lizeaga', 'phone': 943468290, 'address': 'Gartziategi Baserria, Martutene Pasealekua, 139', 'postalCode': 20155, 'municipality': 'Astigarraga', 'territory': 'Gipuzkoa'}
{'documentName': 'Mina', 'phone': 943555220, 'address': 'Txoritokieta, 50', 'postalCode': 20155, 'municipality': 'Astigarraga', 'territory': 'Gipuzkoa'}
{'documentName': 'Mizpiradi', 'phone': 943593954, 'address': 'Barrio Leizotz, s/n', 'postalCode': 20140, 'municipality': 'Andoain', 'territory': 'Gipuzkoa'}
{'documentName': 'Oarso', 'phone': 943515956, 'address': 'Zubiaurre, 8 (trasera)', 'postalCode': 20100, 'municipality': 'Errenteria', 'territory': 'Gipuzkoa'}
{'documentName': 'Oianume', 'phone': 943556683, 'address': 'Barrio Ergoien, 18', 'postalCode': 20130, 'municipality': 'Urnieta', 'territory': 'Gipuzkoa'}
{'documentName': 'Oiarbide', 'phone': 943553199, 'address': 'Oiarbide Bidea, z/g', 'postalCode': 20155, 'municipality': 'Astigarraga', 'territory': 'Gipuzkoa'}
{'documentName': 'Oiharte', 'phone': 943501013, 'address': 'Irukaketa gaina', 'postalCode': 20214, 'municipality': 'Zerain', 'territory': 'Gipuzkoa'}
{'documentName': 'Oilurta Azpi', 'phone': 943643708, 'address': 'Jaizubia auzoa, 45', 'postalCode': 20280, 'municipality': 'Hondarribia', 'territory': 'Gipuzkoa'}
{'documentName': 'Ola', 'phone': 943623130, 'address': 'Barrio Meaka, s/n', 'postalCode': 20304, 'municipality': 'Irun', 'territory': 'Gipuzkoa'}
{'documentName': 'Olagi', 'phone': 943887726, 'address': 'Altzaga bidea, 1', 'postalCode': 20248, 'municipality': 'Altzaga', 'territory': 'Gipuzkoa'}

{'documentName': 'Ordo-Zelai', 'phone': 943491686, 'address': 'Txalaka bide
a, 3 - Ugaldetxo Auzoa', 'postalCode': 20180, 'municipality': 'Oiartzun',
'territory': 'Gipuzkoa'}
{'documentName': 'Otegi', 'phone': 943365029, 'address': 'Caserío Solla-Ene
a, barrio Ilarratzueta', 'postalCode': 20160, 'municipality': 'Lasarte-Ori
a', 'territory': 'Gipuzkoa'}
{'documentName': 'Otsu Enea', 'phone': 943556894, 'address': 'Osinaga Auzo
a', 'postalCode': 20120, 'municipality': 'Hernani', 'territory': 'Gipuzko
a'}
{'documentName': 'Petritegi', 'phone': 943457188, 'address': 'Petritegi Bid
ea, 6', 'postalCode': 20155, 'municipality': 'Astigarraga', 'territory': 'G
ipuzkoa'}
{'documentName': 'R. Zabala', 'phone': 943690774, 'address': 'Caserío de Ga
ragartza', 'postalCode': 20150, 'municipality': 'Aduna', 'territory': 'Gipu
zkoa'}
{'documentName': 'Rezola', 'phone': 608143332, 'address': 'Santio Zeharra,
12', 'postalCode': 20155, 'municipality': 'Astigarraga', 'territory': 'Gipu
zkoa'}
{'documentName': 'Romeral', 'phone': 943344211, 'address': 'Eskalantegi, 4
2', 'postalCode': 20110, 'municipality': 'Pasaia', 'territory': 'Gipuzkoa'}
{'documentName': 'Rufino', 'phone': 943552739, 'address': 'Akarregi Baserri
a', 'postalCode': 20120, 'municipality': 'Hernani', 'territory': 'Gipuzko
a'}
{'documentName': 'Saizar', 'phone': 943373995, 'address': 'Barrio Kale-zaha
r, 39', 'postalCode': 20170, 'municipality': 'Usurbil', 'territory': 'Gipuz
koa'}
{'documentName': 'San Roque', 'phone': 944851085, 'address': 'Anbia, 24',
'postalCode': 48901, 'municipality': 'Barakaldo', 'territory': 'Bizkaia'}
{'documentName': 'Sarasola', 'phone': 943919328, 'address': 'Camino de Oiar
bide, 14', 'postalCode': 20115, 'municipality': 'Astigarraga', 'territory':
'Gipuzkoa'}
{'documentName': 'Sarasola', 'phone': 943690283, 'address': 'Calle Bellabar
a - Carretera Alkiza', 'postalCode': 20159, 'municipality': 'Asteasu', 'ter
ritory': 'Gipuzkoa'}
{'documentName': 'Satxota', 'phone': 943835738, 'address': 'Barrio Santiag
o, 26', 'postalCode': 20809, 'municipality': 'Aia', 'territory': 'Gipuzko
a'}
{'documentName': 'Setien', 'phone': 943551014, 'address': 'Gurutzeta Berri,
"Moko" - Oztaran auzoa', 'postalCode': 20130, 'municipality': 'Urnieta', 't
erritory': 'Gipuzkoa'}
{'documentName': 'Sidrería Goikoetxea', 'phone': 943682175, 'address': 'Elb
arrena, 4 - Aroztegi Baserria', 'postalCode': 20490, 'municipality': 'Lizar
tza', 'territory': 'Gipuzkoa'}
{'documentName': 'Tximista', 'phone': 943881128, 'address': 'Gudarien etorb
idea, 2', 'postalCode': 20240, 'municipality': 'Ordizia', 'territory': 'Gip
uzkoa'}
{'documentName': 'Txindurri-Iturri', 'phone': 943199389, 'address': 'Barrio
Mardari, 12', 'postalCode': 20829, 'municipality': 'Deba', 'territory': 'Gi
puzkoa'}
{'documentName': 'Txinparta', 'phone': 943376698, 'address': 'Txikierdi Auz
oa, 7. Centro Comercial Urbil.', 'postalCode': 20170, 'municipality': 'Usur
bil', 'territory': 'Gipuzkoa'}
{'documentName': 'Txirrita', 'phone': 943467638, 'address': 'San Bartolomé,
32', 'postalCode': 20007, 'municipality': 'Donostia / San Sebastián', 'terr
itory': 'Gipuzkoa'}
{'documentName': 'Txomin', 'phone': 943451964, 'address': 'Pº Aintzieta,6.
LOIOLA', 'postalCode': 20014, 'municipality': 'Donostia / San Sebastián',
'territory': 'Gipuzkoa'}
{'documentName': 'Unzueta', 'phone': 945445095, 'address': 'Zabola auzoa,
2', 'postalCode': 1169, 'municipality': 'Aramaio', 'territory': 'Araba/Álav
a'}
{'documentName': 'Urbitarte', 'phone': 943180119, 'address': 'Urbitarte',
'postalCode': 20211, 'municipality': 'Ataun', 'territory': 'Gipuzkoa'}
{'documentName': 'Urdaira', 'phone': 943372691, 'address': 'Barrio Aginag

```
a', 'postalCode': 20170, 'municipality': 'Usurbil', 'territory': 'Gipuzko
a'}
{'documentName': 'Usabal', 'phone': 943674316, 'address': 'Polígono Usabal,
22', 'postalCode': 20400, 'municipality': 'Tolosa', 'territory': 'Gipuzko
a'}
{'documentName': 'Uxarte', 'phone': 946308815, 'address': 'Bº Montorra, 6',
'postalCode': 48340, 'municipality': 'Amorebieta-Etxano', 'territory': 'Biz
kaia'}
{'documentName': 'Zapiain', 'phone': 943330033, 'address': 'Rekalde Etxea,
Calle de Nagusia 96', 'postalCode': 20155, 'municipality': 'Astigarraga',
'territory': 'Gipuzkoa'}
{'documentName': 'Zelaia', 'phone': 943555851, 'address': 'Barrio Martindeg
i, 29', 'postalCode': 20120, 'municipality': 'Hernani', 'territory': 'Gipuz
koa'}
{'documentName': 'de Algorta', 'phone': 944604177, 'address': 'Comporte,
3.', 'postalCode': 48992, 'municipality': 'Getxo', 'territory': 'Bizkaia'}
```

5.1.2. Asadores que hay en Bilbao

Otra de las consultas habituales es los asadores que hay en una ciudad concreta, para este ejemplo vamos a utilizar Bilbao:

- **Descripción de la consulta**: Queremos consultar los asadores que hay en Bilbao, pero sólo nos interesa el nombre del asador, la dirección y el teléfono.
- **Resultado esperado**: Se espera un listado de sólo el nombre, dirección y teléfono de los asadores de Bilbao.

Al igual que se ha realizado antes, para ejecutar la consulta directamente sobre **MongoDB** sería:

```
use salidas_grupales
db.restaurantes.find({"restorationType" : "Asador",
"municipality":"Bilbao"},
{"documentName":1,"address":1,"phone":1,"_id":0})
```

De esta manera obtendríamos los datos. A continuación se muestra el código necesario para realizar la consulta en Python y la salida de los datos obtenidos:

In [14]:
```
for asador in collection_local.find({"restorationType" : "Asador", "municip
    print(asador)
```

```
{'documentName': 'Arraiz', 'phone': 944431919, 'address': 'Monte Arraiz, 11
6'}
{'documentName': 'Guetaria', 'phone': 944243923, 'address': 'Colón de Larre
ategui, 12'}
{'documentName': 'Ibañez de Bilbao', 'phone': 944233034, 'address': 'Ibáñez
de Bilbao, 6'}
{'documentName': 'La Gabarra', 'phone': 944477062, 'address': 'Botica Viej
a, 18'}
{'documentName': 'Zuria', 'phone': 944246080, 'address': 'Uribitarte , 7'}
```

5.2. Consultas avanzadas

Aparte de las consultas básicas, es necesario realizar ciertas consultas más avanzadas en los datos como las siguientes:

5.2.1. Buscar restaurantes que oferten menú vegetariano

Tal como se ha comentado al inicio, existen personas vegetarianas en nuestros grupos, y por tanto es interesante conocer restaurantes que oferten este tipo de menús. Debido a que los menús de este tipo se pueden llamar de manera distinta, haremos uso de **expresiones regulares** para realizar la búsqueda de los términos "vegetariano", "vegetariana", "vegano" y "vegana", pero ignorando si está en mayúsculas o minúsculas.

- **Descripción de la consulta**: Queremos consultar los restaurantes que ofertan menús vegetarianos.
- **Resultado esperado**: Se espera un listado con el nombre del restaurante, la descripción del mismo, la dirección, el municipio y el teléfono.

Siguiendo con las consultas anteriores, para ejecutar la consulta directamente sobre **MongoDB** sería:

```
use salidas_grupales
db.restaurantes.find({"restorationType":"Restaurante",
"turismDescription": /(vegetarian[ao]|vegan[ao])/},
{"documentName":1,"turismDescription":1,"address":1,"municipality":1,'
```

De esta manera obtendríamos los datos. A continuación se muestra el código necesario para realizar la consulta en Python y la salida de los datos obtenidos:

In [15]:
```
# importamos la librería de expresiones regulares
import re
# queremos buscar por la palabra "vegetariano", "vegetariana", "vegano" y "
regx = re.compile("(vegetarian[ao]|vegan[ao])", re.IGNORECASE)

for vegano in collection_local.find({"restorationType" : "Restaurante", "tu
    print(vegano)
```

{'documentName': '5ª Planta', 'turismDescription': ' La quinta planta del Teatro Campos de Bilbao alberga el restaurante 5&ordf Planta, que ofrece u na cocina creativa, basada en la vanguardia y las tendencias actuales. Cuen ta con men&uacute s ecol&oacute gicos, vegetarianos y para cel&iacute acos. Entre sus platos destacan el arroz cremoso con chipirones y hongos, las alm ejas rellenas de changurro empanado en panko y bacalao (al pil-pil, a la vi zca&iacute na, en salsa negra...). ', 'phone': 944079259, 'address': 'R estaurante del Teatro Campos Elíseos. Bertendona 3 , 5º Planta', 'municipal ity': 'Bilbao'}
{'documentName': 'Amalurra', 'turismDescription': ' Un amplio y luminoso edificio de l&iacute nea vanguardista alberga el restaurante Amalurra, que tambi&eacute n funciona como hotel rural. Adem&aacute s de cocina casera, l os comensales tambi&eacute n podr&aacute n disfrutar de men&uacute s vegeta rianos y platos t&iacute picos de la cocina vasca. En su exterior cuenta co n amplias zonas ajardinadas, que hacen del restaurante un lugar id&oacute n eo para la celebraci&oacute n de todo tipo de eventos. ', 'phone': 94610 9540, 'address': 'Bº La Reneja, 35', 'municipality': 'Artzentales'}
{'documentName': 'Arimendi', 'turismDescription': ' En el restaurante Ari mendi del Hotel Jardines de Uleta se trabaja una cocina tradicional, en la que se da prioridad a los productos locales. Especial menci&oacute n merece n sus carnes y pescados a la parrilla. Cuenta con m&aacute s de 120 refe rencias de las diversas regiones vin&iacute colas de Espa&ntilde a, as&iacu te como una amplia gama de vinos del mundo. El restaurante dispone de d os salones privados y sus pintxos han sido galardonados en concursos gastro n&oacute micos de renombre. Ofrecen un men&uacute vegetariano y otro para cel&iacute acos. ', 'phone': 945133131, 'address': 'Calle Uleta, 1', 'mu nicipality': 'Vitoria-Gasteiz'}
{'documentName': 'Bistró Guggenheim Bilbao', 'turismDescription': ' Prese nta un estilo de cocina libre, adaptando las ra&iacute ces y la cultura gas tron&oacute mica vasca con a un contexto de vanguardia. Las recetas y sals as antiguas se presentan evolucionadas y adaptadas a una cocina sabrosa, sa na en una experiencia gastron&oacute mica informal. Ofrece men&uacute infa ntil, men&uacute s de grupos y platos para vegetarianos, al&eacute rgicos y cel&iacute acos La terraza est&aacute situada en la plaza del Museo. El h orario de la terraza bar es de 9:30 a 20:30 h de martes a domingo. Mediod& iacute a: de martes a domingo de 13:00 a 15:30 h Noche: de jueves a s&aacu te bado de 20:30 a 22:30 h Los meses de julio y agosto abierto todos los d& iacute as mediod&iacute a y noche. ', 'phone': 944239333, 'address': 'A vda. Abandoibarra nº 2', 'municipality': 'Bilbao'}
{'documentName': 'Fadura', 'turismDescription': ' El Restaurante Fadura, cuyo nombre hace referencia a la zona de Getxo en la que se sit&uacute a, o frece una comida tradicional, pero haciendo un especial hincapi&eacute en el uso de productos ecol&oacute gicos en su cocina. Junto a su men&uacut e del d&iacute a,ofertan tambi&eacute n una opci&oacute n vegetariana. Ade m&aacute s, el restaurante es una escuela de hosteler&iacute a que ofrece c ursos tanto a adultos para propiciar su insercci&oacute n laboral, como a l o m&aacute s peque&ntilde os. ', 'phone': 946569755, 'address': 'Aveni da de los Chopos s/n, Getxo (Junto al polideportivo de Fadura)', 'municipal ity': 'Getxo'}
{'documentName': 'La Escuela', 'turismDescription': ' Restaurante de la E scuela de Hosteler&iacute a situado sobre Bilbao, en el monte Art xanda y rodeado de verde.   Ofrece una am plia variedad de men&uacute s adaptados a cada tipo de cliente. Buenas vist as de Bilbao. Men&uacute s especiales para celiacos, vegetarianos, jud&iacu te os, &aacute rabes y ni&ntilde os. ', 'phone': 944745200, 'address': 'Ctra. Enekuri-Artxanda Km 3', 'municipality': 'Bilbao'}
{'documentName': 'Rubiorena', 'turismDescription': ' El restaurante Rubio rena est&aacute ubicado en una zona de ocio de Beasain (G ipuzkoa).   Con Urki Balerdi a los fogones ofrece una coci na tradicional moderna y siempre con acertadas creaciones personales. Urki es especialista en menus vegetarianos y ademas su carta se adec&uacute a a cualquier tipo de menu. La carta esta tambi&eacute n orientada para celiaco s. ', 'phone': 943089519, 'address': 'Zaldizurrieta, 7', 'municipality':

```
'Beasain'}
{'documentName': 'Toki Alai', 'turismDescription': '<p>Ofrece  men&uacu
te  del d&iacute a de lunes a viernes por 11 &euro . </p><p>Adem&aacute s d
e la gran variedad de platos tradicionales, tambi&eacute n  hay platos
macrobi&oacute ticos, veganos y sin gluten para ayudarte a cuidar tu salud
todos los d&iacute as con el  men&uacute  del d&iacute a.</p><p>El fin
de semana adem&aacute s de la carta ofrecen sugerencias especiales.</p>',
'phone': 943888953, 'address': 'Herriko Plaza', 'municipality': 'Arama'}
```

5.2.2. Búsqueda geoespacial de restaurantes a 2 kilómetros de una posición dada

En algunas situaciones nos puede interesar no realizar grandes desplazamientos, y por tanto realizar una búsqueda de restaurantes sobre la posición en la que nos encontramos. Para ello necesitamos hacer búsquedas teniendo en cuenta la geoposición en la que nos encontramos. Para que estas búsquedas funcionen se ha tenido que crear un **índice** indicando cuál es el campo donde se encuentra la longitud y latitud (que en nuestro caso es el campo **location**) de la siguiente manera (aparece en el fichero **consultas.json**).

```
db.restaurantes.createIndex( { location: "2dsphere" } )
```

De esta manera las búsquedas funcionarán.

- **Descripción de la consulta**: Queremos consultar los restaurantes que están a una distancia de 2 kilómetros de un punto geo-espacial dado.
- **Resultado esperado**: Se espera un listado con el nombre del restaurante, la dirección y el teléfono (no se necesitan más datos porque ya sabemos la ciudad donde nos encontramos).

Siguiendo con las consultas anteriores, para ejecutar la consulta directamente sobre **MongoDB** sería:

```
use salidas_grupales
db.restaurantes.createIndex( { location: "2dsphere" } )
db.restaurantes.find({
    "location":
        { $geoNear:
          {
            $geometry: { type: "Point",  coordinates: [
-2.9388531, 43.2577024 ] },
            $maxDistance: 2000
          }
        }
    },{"documentName":1,"address":1,"phone":1,"_id":0})
```

De esta manera obtendríamos los datos. Se ha utilizado como posición en la que nos encontramos las coordenadas "-2.9388531, 43.2577024".

A continuación se muestra el código necesario para realizar la consulta en Python y la salida de los datos obtenidos:

In [16]:
```
# creamos el índice
collection_local.create_index([("location", pymongo.GEOSPHERE)])

# documentación sobre búsquedas geoespaciales con pymongo: https://pymongo.
from bson.son import SON
```

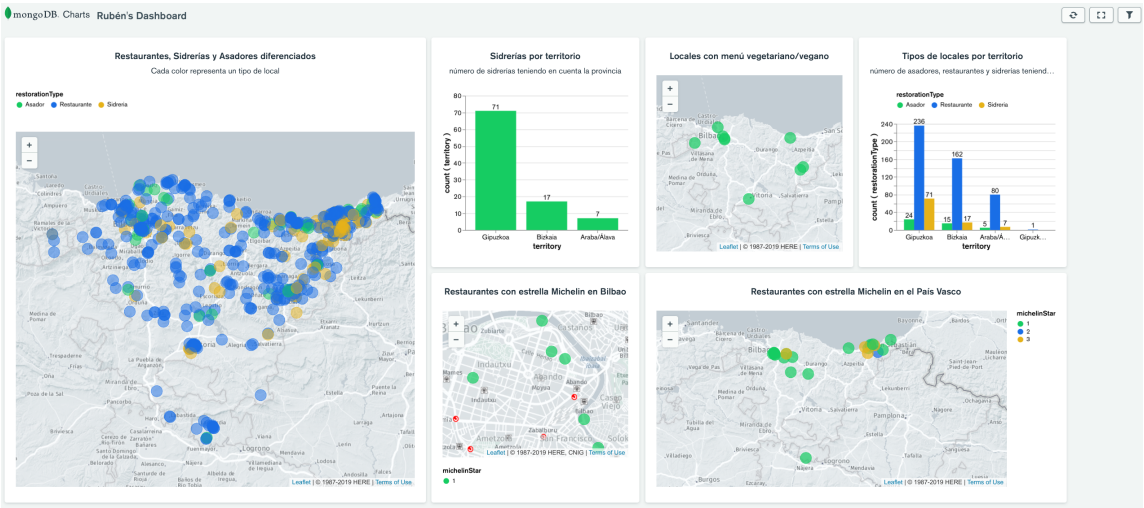
```
# búsqueda de restaurantes a menos de 2km de una coordenada dada:
query = {"location": SON([("$near",{"coordinates":[-2.9388531, 43.2577024]}
# realizamos la búsqueda
for cercano in collection_local.find(query,{"documentName":1,"address":1,"p
    print(cercano)
```

{'documentName': 'Etxaniz', 'phone': 944440004, 'address': 'Bombero Etxaniz, s/n'}
{'documentName': 'Yandiola', 'phone': 944133636, 'address': 'Plaza Arriquibar, 4 (Centro de Ocio y Cultura AlhóndigaBilbao)'}
{'documentName': 'Serantes II', 'phone': 944102699, 'address': 'Alameda de Urquijo, 51'}
{'documentName': 'Bermeo', 'phone': 944705700, 'address': 'Ercilla, 37-39 (Hotel Ercilla)'}
{'documentName': 'Serantes', 'phone': 944212129, 'address': 'Licenciado Poza, 16'}
{'documentName': '5ª Planta', 'phone': 944079259, 'address': 'Restaurante del Teatro Campos Elíseos. Bertendona 3 , 5º Planta'}
{'documentName': 'Zarate', 'phone': 944416521, 'address': 'Licenciado Poza, 65'}
{'documentName': 'Zurekin', 'phone': 946791440, 'address': 'Diputacion, 8'}
{'documentName': 'Restaurante Lasa', 'phone': 944240103, 'address': 'Diputación, 3'}
{'documentName': 'Le Bol Blanc', 'phone': 944416000, 'address': 'Gran Vía, 87 (H. Villa de Bilbao)'}
{'documentName': 'Guetaria', 'phone': 944243923, 'address': 'Colón de Larreategui, 12'}
{'documentName': 'Beraia', 'phone': 944052844, 'address': 'Alameda Recalde, 20'}
{'documentName': 'Ola Martín Berasategui', 'phone': 944652066, 'address': 'Erribera, 13'}
{'documentName': 'Club Náutico', 'phone': 944235500, 'address': 'Obispo Orueta, 2-4. Hotel López de Haro'}
{'documentName': 'Cafetería Abandoibarra', 'phone': 944035149, 'address': 'Avenida Abandoibarra, 4'}
{'documentName': 'Café Iruña', 'phone': 944237021, 'address': 'Jardines de Albia'}
{'documentName': 'Serantes III', 'phone': 944248004, 'address': 'Alameda Mazarredo, 75'}
{'documentName': 'La despensa de Etxanobe', 'address': 'Juan Ajuriaguerra, 8'}
{'documentName': 'Atelier Etxanobe', 'phone': 944421071, 'address': 'Juan Ajuriaguerra, 8'}
{'documentName': 'Aitor Rauleaga', 'phone': 944256345, 'address': 'Colón de Larreátegui Kalea, 9'}
{'documentName': 'Jauregia', 'phone': 944035151, 'address': 'Avda. Abandoibarra, 4 (Palacio Euskalduna)'}
{'documentName': 'Arriaga', 'phone': 944165670, 'address': 'Santa María, 13'}
{'documentName': 'Zortziko', 'phone': 944239743, 'address': 'Alda. Mazarredo, 17'}
{'documentName': 'Maider', 'phone': 944248990, 'address': 'Colón de Larreategui, 5'}
{'documentName': 'Aizian', 'phone': 944280039, 'address': 'Lehendakari Leizaola, 29. (H. Meliá Bilbao)'}
{'documentName': 'Cruz-Blanca', 'phone': 944395000, 'address': 'Lehendakari Leizaola, 2'}
{'documentName': 'Zuria', 'phone': 944246080, 'address': 'Uribitarte , 7'}
{'documentName': 'Porrue', 'phone': 944231313, 'address': 'Alameda Rekalde, 4'}
{'documentName': 'Le Café', 'phone': 944253300, 'address': 'Alameda de Mazarredo, 61'}
{'documentName': 'Beltz', 'phone': 944253300, 'address': 'Alameda Mazarredo, 61'}
{'documentName': 'Ibañez de Bilbao', 'phone': 944233034, 'address': 'Ibáñez de Bilbao, 6'}
{'documentName': 'Mina', 'phone': 944795938, 'address': 'Muelle Marzana, s/n'}
{'documentName': 'Larruzz Bilbao', 'phone': 944230820, 'address': 'Uribitarte, 24'}

```
{'documentName': 'Bistró Guggenheim Bilbao', 'phone': 944239333, 'address':
'Avda. Abandoibarra nº 2'}
{'documentName': 'Baita Bilbao', 'phone': 944242267, 'address': 'Alameda Ma
zarredo, 20'}
{'documentName': 'Víctor', 'phone': 944151678, 'address': 'Plaza Nueva, 2'}
{'documentName': 'Nerua', 'phone': 944000430, 'address': 'Avda. Abandoibarr
a, 2'}
{'documentName': 'La Gabarra', 'phone': 944477062, 'address': 'Botica Viej
a, 18'}
{'documentName': 'Dena-Ona', 'phone': 944470596, 'address': 'Rafaela Ybarr
a, 12'}
{'documentName': 'Arraiz', 'phone': 944431919, 'address': 'Monte Arraiz, 11
6'}
```


6. MongoDB Charts

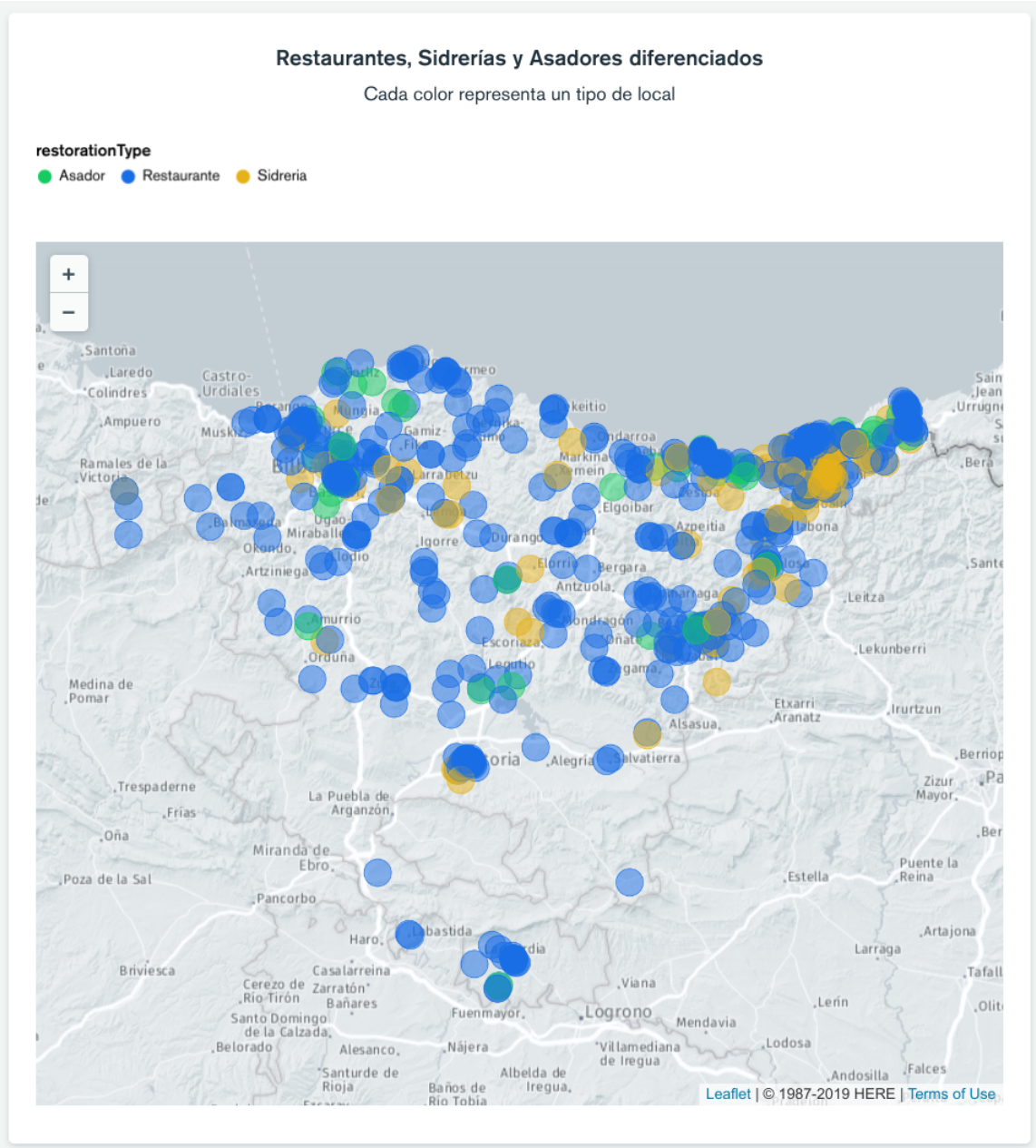
Aparte de las consultas, también se han decidido crear varias gráficas ya que de esta manera se tiene un acceso más visual a los datos. Las gráficas que se han creado se han hecho públicas a través del siguiente enlace: https://charts.mongodb.com/charts-prueba-vmkve/public/dashboards/62952659-02e8-45da-8754-3cefedb01fdd que nos mostrará el siguiente panel en el que podremos ver todas las gráficas creadas:



A continuación se van a detallar varias de las gráficas creadas.

6.1. Restaurantes, Sidrerías y Asadores

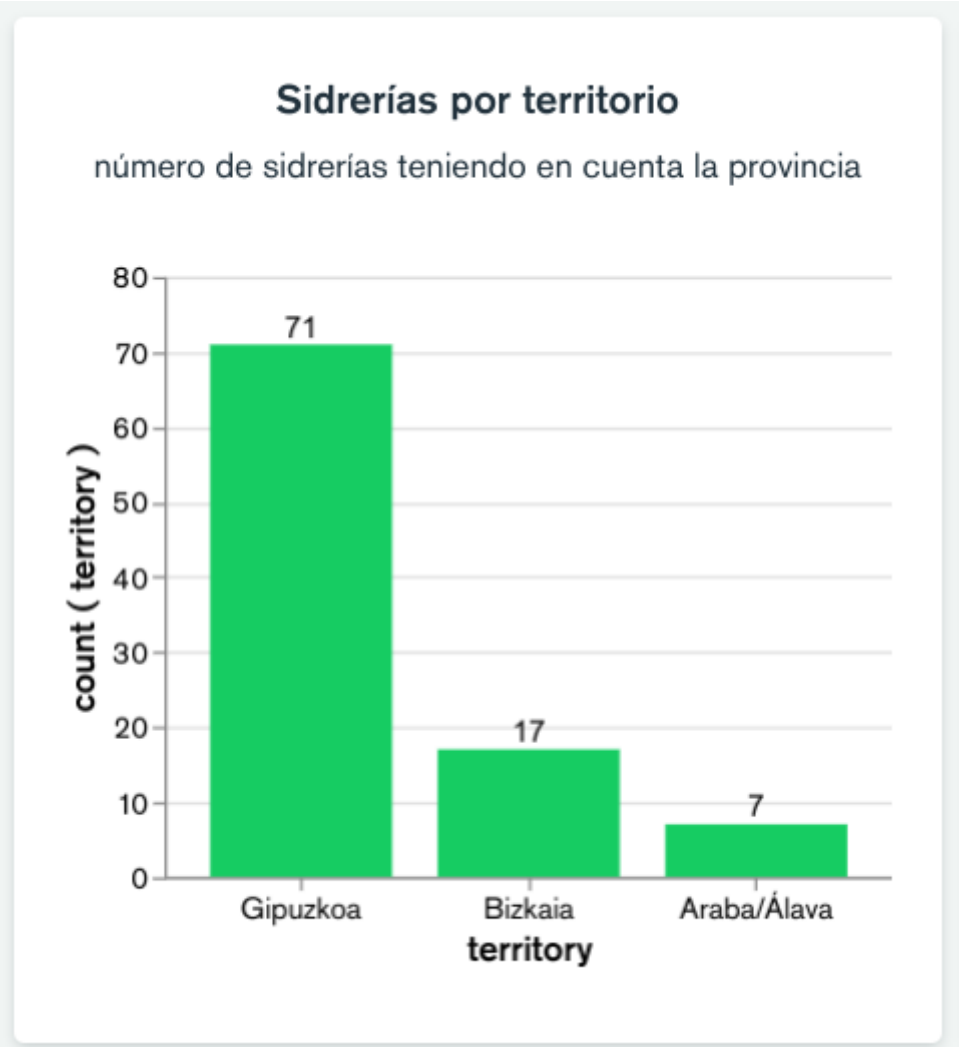
En este gráfico se han geolocalizado todos los datos obtenidos desde la web de OpenData diferenciando por el tipo de local que es: restaurante, sidrería o asador:



Esta gráfica, de tipo **Geospatial** y subtipo **Scatter**, se realiza de manera sencilla ya que la búsqueda es con todos los elementos (`{}`), indicando que las coordenadas están en el campo **location**, y diferenciando el color por el campo "**restorationType**" del dataset.

6.2. Sidrerías por territorio

Esta gráfica nos muestra el número de sidrerías que existen por territorio histórico, viendo claramente cómo Gipuzkoa es la que más tiene:

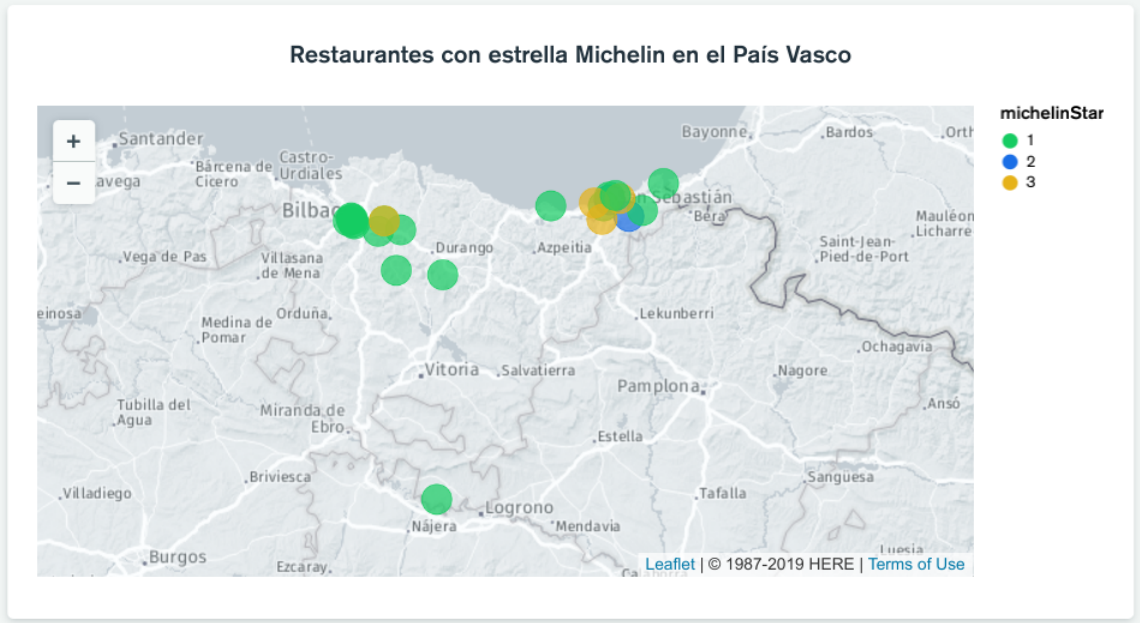


Para realizar esta gráfica se ha realizado la búsqueda `{"restorationType" : "Sidreria"}` y poniendo en la gráfica:

- **Eje X**: "territory", para diferenciar cada provincia
- **Eje Y**: un COUNT de los campos

6.3. Restaurantes con estrella Michelin

Para conocer cuáles son los mejores restaurantes de la comunidad autónoma, y que tienen el prestigio de contar con al menos una estrella Michelin se ha hecho la siguiente gráfica:



Para realizar esta gráfica se ha realizado la siguiente búsqueda: `{"michelinStar": {"$gt": 0}}`, de esta manera sólo nos aparecen los restaurantes que tienen una estrella Michelin o más. Al igual que la primera gráfica, es de tipo **Geospatial** y subtipo **Scatter**, y se ha tenido en cuenta el número de estrellas Michelin para que tengan color los datos obtenidos.

7. Conclusiones

Tal como se ha podido ver a lo largo de todo el documento, el hacer uso de datos del portal de transparencia de nuestra comunidad nos permite disponer de datos oficiales que podemos utilizar para crear nuestros propios datasets en bases de datos NoSQL como es MongoDB.

Tras realizar el análisis de los datos, se ha creado un conjunto de datos personalizado para posteriormente desplegarlos en nuestros servidores (tanto de manera local como remota) para poder realizar la búsqueda de información que nos interesaba. Para mostrar de manera más gráfica el conjunto de datos que hemos obtenido, se ha hecho uso de MongoDB Charts, dando un aspecto mucho más interesante a los datos.

Como conclusión final, el uso de bases de datos NoSQL nos ha permitido tener un conjunto de datos flexible con el que hemos podido dar respuesta a las preguntas que teníamos al inicio del documento.