INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE CÓMPUTO

Prototipo de análisis de datos ciudadanos relacionados a contaminación ambiental.

Trabajo Terminal: No. 2021-B051

Integrantes:

Hernández Clemente Samantha

Medina Flores Susana

Olivares Conchillos Leonel

Director de TT:

Zagal Flores Eswart Roberto.

Contenido.

1. Recapitulación

La contaminación ambiental en Ciudad de México es uno de los temas más preocupantes, existen sitios web basados en el monitoreo de calidad del aire y de emisiones atmosféricas, donde podemos visualizar información estadística acerca de este tema.

Pero en redes sociales existen datos que pueden ayudar a identificar denuncias de carácter social relacionadas al medio ambiente, sin embargo esta información no ha sido analizada.

El reto técnico es obtener, limpiar y analizar información sobre la calidad del aire que nos arroje un valor sobre la contaminación en la Ciudad de México y nos permita generar datos sobre la contaminación a partir de las denuncias ciudadanas en las redes sociales.

1.1 Delimitación de los temas.

Tránsito vehicular

La Secretaría del medio ambiente (SEDEMA) declaró que las fuentes móviles generan el 78% de partículas de óxido de nitrógeno.

Pirotecnia

La Secretaría del medio ambiente (SEDEMA) afirma que es la principal generadora de contingencias ambientales en época invernal.

Incendios

Las emisiones de fuentes naturales, el 86% fueron de compuestos orgánicos volátiles (COV).

“El tema de la contaminación atmosférica es tratado frecuentemente en Twitter, describiendo situaciones o eventos que son fuente de contaminación [11]”

2. Objetivo General

Desarrollar un prototipo de software que permita analizar descriptivamente publicaciones extraídas de Twitter relacionadas a situaciones de contaminación del medio ambiente en Ciudad de México a fin de obtener una caracterización espacial y temporal sobre eventos que generan contaminación del aire como es el caso de la pirotecnia, el tráfico y los incendios.

3. Solución propuesta

“Prototipo de análisis de datos ciudadanos relacionados a contaminación ambiental”

4. Herramientas utilizadas

NumPy

pandas

python NLTK

spaCy

GENSIM
topic modelling for humans

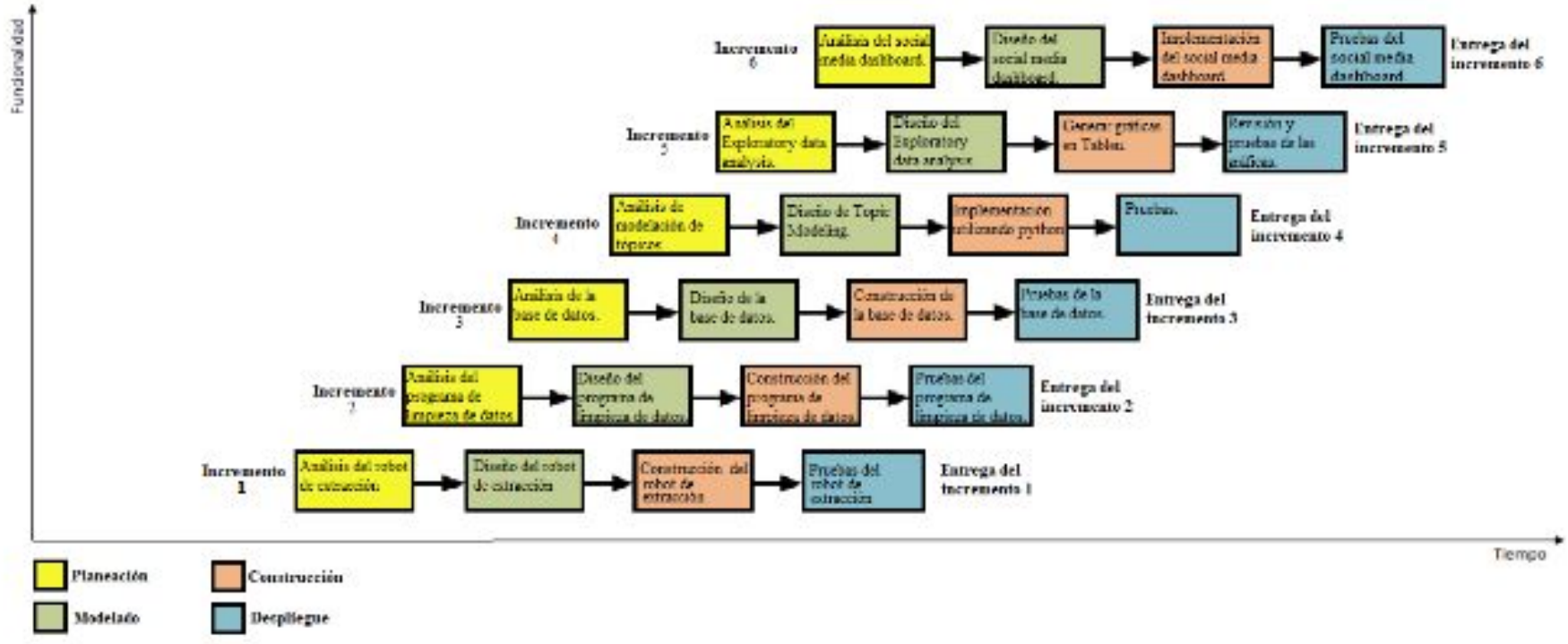
- Tweepy
- Snscrape

MySQL™

+ableau

5. Implementación
Metodología incremental.

- **Incremento 1:** Extracción de datos sociales.
- **Incremento 2:** Limpieza de datos.
- **Incremento 3:** Base de datos.
- **Incremento 4:** Análisis de datos - Topic Modeling.
- **Incremento 5:** Topic Modeling con Tableau.
- **Incremento 6:** Dashboard.



5.1 Extracción de datos.

5.1.1 Parámetros de búsqueda

Se hizo una búsqueda de tweets ideales que aportan información necesaria para visualizar palabras clave que caracterizan a cada tema. (tema, hashtag, mención de la alcaldía desde donde se está denunciando)

Fig 1. Tweet ideal relacionado a pirotecnia

Fig 2. Tweet ideal relacionado a incendios

5.1.2 Creación de querys

```
topics = [
        "(pirotecnia OR cohete OR 'fuegos artificiales')",
        "(tránsito OR tráfico)",
        "(incendio OR humo OR fuego)"
    ]
```

Fig 3. Querys para la búsqueda de los temas seleccionados.

```
new_query = f'{query} geocode:{coordenadas},{radio} since:{start_date_time} until:{end_date_time} {meta_search}'
new_query = f'{query} {mun_query} since:{start_date_time} until:{end_date_time} {meta_search}'
```

Fig 4. Querys para la búsqueda de los temas seleccionados.

5.1.3 Robot de Extracción

```
INICIO
    si el programa se ejecutó con los argumentos archivo de query y archivo de resultados
        si se puede abrir el archivo de querys
            imprimir "Which method will you use? Tweepy or Snscrape (t/s)?:"
            si (method=t)
                escribir el día de búsqueda
                escribir el mes de búsqueda
                para i=0 hasta i=23 con i++
                    llamar a la función search_tweets_tweepy()
                    almacenar los datos en la variable tweets_found.
                    Imprimir los Tweets encontrados.
                fin para
                ir a FIN
            si no, si (method=s)
                escribir la fecha de inicio de búsqueda.
                escribir la fecha de fin de búsqueda.
                llamar a la función search_tweets_snscrape()
                almacenar los datos en la variable tweets_found.
                Imprimir los Tweets encontrados.
                ir a FIN
            fin si
        si no
            imprimir "The file with the querys doesn't exist."
            ir a FIN
        fin si
    si no
        imprimir "Introduce the query_file and file_name of the search in the console's args."
        ir a FIN
    fin si
FIN
```

Fig 5. Pseudocódigo del Robot de extracción

Fig 6. Ejemplo de denuncias extraídas

5.2 Limpieza y Transformación de datos.

5.2.1 Limpieza de los datos

```
# eliminamos registros repetidos
df = df.drop_duplicates(['pubID'])
# obtener el municipio correcto de los registros extraídos por coordenadas
df = df.apply(mun_request, axis=1)
# eliminar registros que están fuera de la CDMX
df = df.drop(df[df['geoID'] == 0].index)
# cambiaremos los tipos de datos bool a 0 y 1 para su uso en MySQL
df['authorVerified'] = df['authorVerified'].apply(lambda x: 1 if x else 0)
# eliminamos registros con posibles datos nulos de los campos importantes.
cabeceras = list(df.columns)
cabeceras.remove('likeCount')
cabeceras.remove('replyCount')
cabeceras.remove('retweetCount')
cabeceras.remove('followersCount')
cabeceras.remove('followingCount')
for columna in cabeceras:
    df = df[df[columna].notna()]
```

Fig 7. Código de limpieza de datos

5.2.2 Transformación de los datos

5.2.2.1 Tokenización y lematización

Convertir un texto en una lista de tokens (palabras), pero omitiendo las palabras conectoras, que se conocen como stopwords.

Se corrigen las palabras mal escritas y se lematizan para regresarlas a su base.

```python
def tokens_tweet(tweet):
    pattern = r'''(?x)          # set flag to allow verbose regexps
            (?:[A-Z]\.)+        # abbreviations, e.g. U.S.A.
            | \w+(?:-\w+)*      # words with optional internal hyphens
            | \$?\d+(?:\.\d+)?%? # currency and percentages, e.g. $12.40, 82%
            | [][.,;"'?():-_`]  # these are separate tokens; includes ], [
    '''
    # obtenemos la lista de stopwords
    stop_words = stopwords.words('spanish')
    # Convertir todo el texto en minúsculas
    tweet = tweet.lower()
    # Remover menciones, hashtags, links y saltos de linea
    tweet = re.sub("@[A-Za-z0-9_]+","", tweet)
    tweet = re.sub("#\S+","", tweet)
    tweet = re.sub(r"http\S+", "", tweet)
    tweet = re.sub(r"www.\S+", "", tweet)
    tweet = re.sub(r'\n', '', tweet)
    # Tokenización
    tokens_tweet = nltk.regexp_tokenize(tweet, pattern)
    tokens_tweet = [token for token in tokens_tweet if len(token) > 1]
    # Remover los stopwords
    interesting_tokens = [w for w in tokens_tweet if not w in stop_words]
    # corregir palabras
    spell = Speller(lang='es')
    interesting_tokens = [spell(w) for w in interesting_tokens]
    # Lematización
    nlp = spacy.load("es_dep_news_trf")
    doc = nlp(' '.join(interesting_tokens))
    interesting_tokens = [w.lemma_ for w in doc]

    return interesting_tokens
```

Fig 8. Código de Tokenización y Lematización

5.2.2.2 Validar Ubicación

Se definió una función que utiliza la API de google maps para poder determinar el municipio correcto de un tweet que fue buscado por coordenadas.

```python
def mun_request(row):
    geo_mun = {
        'Azcapotzalco': 2,
        'Coyoacán': 3,
        'Cuajimalpa de Morelos': 4,
        'Gustavo A. Madero': 5,
        'Iztacalco': 6,
        'Iztapalapa': 7,
        'La Magdalena Contreras': 8,
        'Milpa Alta': 9,
        'Álvaro Obregón': 10,
        'Tláhuac': 11,
        'Tlalpan': 12,
        'Xochimilco': 13,
        'Benito Juárez': 14,
        'Cuauhtémoc': 15,
        'Miguel Hidalgo': 16,
        'Venustiano Carranza': 17
    }
```

```python
    if row['typeQuery'] == 'coordenadas':
        url = f"https://maps.googleapis.com/maps/api/geocode/json?latlng={row['l
atitude']},{row['longitude']}&key={credentials.GOOGLE_MAPS_KEY}"
        res = requests.get(url)
        elements = res.json()
        mun = ''
        if  re.search("CDMX", elements['plus_code']['compound_code']):
            for i in elements['results']:
                if i['address_components'][0]['long_name'] in geo_mun.keys():
                    mun = i['address_components'][0]['long_name']
                    row['geoID'] = geo_mun[mun]
                    row['geoName'] = mun
                    break
        else:
            row['geoID'] = 0
    return row
```

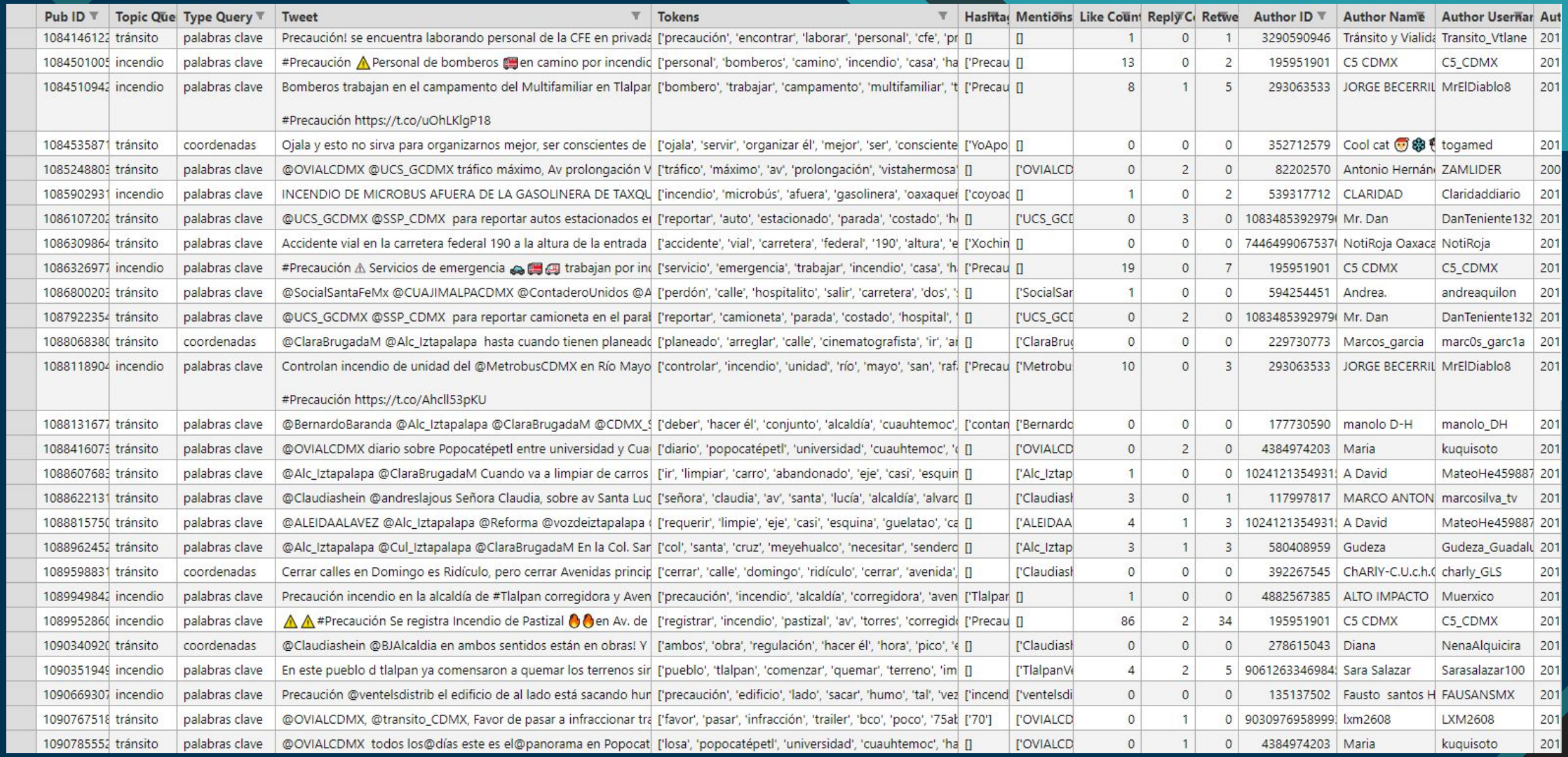
Fig 9. Código de función para determinar municipios

5.2.2.3 Transformar Fecha

Para obtener la fecha y hora correcta de los tweets se hizo lo siguiente:

```
# modificamos las fechas de creación del tweet de acuerdo a los horarios de verano
if año == 19:
    df['pubDate'] = df['pubDate'].apply(lambda x: datetime.datetime.strptime(x, '%Y-%m-%d %H:%M:%S') + datetime.timedelta(hours=1) \
                                        if  datetime.datetime.strptime(x, '%Y-%m-%d %H:%M:%S') > \
                                            datetime.datetime(2019,4,7,2) and \
                                            datetime.datetime.strptime(x, '%Y-%m-%d %H:%M:%S') < \
                                            datetime.datetime(2019,11,3,2) \
                                        else x)
elif año == 20:
    df['pubDate'] = df['pubDate'].apply(lambda x: datetime.datetime.strptime(x, '%Y-%m-%d %H:%M:%S') + datetime.timedelta(hours=1) \
                                        if  datetime.datetime.strptime(x, '%Y-%m-%d %H:%M:%S') > \
                                            datetime.datetime(2020,4,5,2) and \
                                            datetime.datetime.strptime(x, '%Y-%m-%d %H:%M:%S') < \
                                            datetime.datetime(2020,10,25,2) \
                                        else x)
```

Fig 10. Código para asignar fecha y hora correcta



Pub ID	Topic Que	Type Query	Tweet	Tokens	Hashta	Mentions	Like Coun	Reply C	Retwe	Author ID	Author Name	Author Usernar	Aut
1084146122	tránsito	palabras clave	Precaución! se encuentra laborando personal de la CFE en privada	['precaución', 'encontrar', 'laborar', 'personal', 'cfe', 'pr	[]	[]	1	0	1	3290590946	Tránsito y Vialida	Transito_Vtlane	201
1084501005	incendio	palabras clave	#Precaución ⚠Personal de bomberos 🚒en camino por incendio	['personal', 'bomberos', 'camino', 'incendio', 'casa', 'ha	['Precau	[]	13	0	2	195951901	C5 CDMX	C5_CDMX	201
1084510942	incendio	palabras clave	Bomberos trabajan en el campamento del Multifamiliar en Tlalpar #Precaución https://t.co/uOhLKlgP18	['bombero', 'trabajar', 'campamento', 'multifamiliar', 't	['Precau	[]	8	1	5	293063533	JORGE BECERRIL	MrElDiablo8	201
1084535871	tránsito	coordenadas	Ojala y esto no sirva para organizarnos mejor, ser conscientes de	['ojala', 'servir', 'organizar él', 'mejor', 'ser', 'consciente	['YoApo	[]	0	0	0	352712579	Cool cat 🤠❄🍁	togamed	201
1085248803	tránsito	palabras clave	@OVIALCDMX @UCS_GCDMX tráfico máximo, Av prolongación V	['tráfico', 'máximo', 'av', 'prolongación', 'vistahermosa'	[]	['OVIALCD	0	2	0	82202570	Antonio Hernán	ZAMLIDER	200
1085902931	incendio	palabras clave	INCENDIO DE MICROBUS AFUERA DE LA GASOLINERA DE TAXQU	['incendio', 'microbús', 'afuera', 'gasolinera', 'oaxaquer	['coyoac	[]	1	0	2	539317712	CLARIDAD	Claridaddiario	201
1086107202	tránsito	palabras clave	@UCS_GCDMX @SSP_CDMX para reportar autos estacionados er	['reportar', 'auto', 'estacionado', 'parada', 'costado', 'h	[]	['UCS_GCD	0	3	0	1083485392979	Mr. Dan	DanTeniente132	201
1086309864	tránsito	palabras clave	Accidente vial en la carretera federal 190 a la altura de la entrada	['accidente', 'vial', 'carretera', 'federal', '190', 'altura', 'e	['Xochin	[]	0	0	0	7446499067537	NotiRoja Oaxaca	NotiRoja	201
1086326977	incendio	palabras clave	#Precaución ⚠ Servicios de emergencia 🚓🚒🚚 trabajan por inc	['servicio', 'emergencia', 'trabajar', 'incendio', 'casa', 'h	['Precau	[]	19	0	7	195951901	C5 CDMX	C5_CDMX	201
1086800203	tránsito	palabras clave	@SocialSantaFeMx @CUAJIMALPACDMX @ContaderoUnidos @A	['perdón', 'calle', 'hospitalito', 'salir', 'carretera', 'dos', '	[]	['SocialSar	1	0	0	594254451	Andrea.	andreaquilon	201
1087922354	tránsito	palabras clave	@UCS_GCDMX @SSP_CDMX para reportar camioneta en el paral	['reportar', 'camioneta', 'parada', 'costado', 'hospital', '	[]	['UCS_GCD	0	2	0	1083485392979	Mr. Dan	DanTeniente132	201
1088068380	tránsito	coordenadas	@ClaraBrugadaM @Alc_Iztapalapa hasta cuando tienen planeado	['planeado', 'arreglar', 'calle', 'cinematografista', 'ir', 'ar	[]	['ClaraBrug	0	0	0	229730773	Marcos_garcia	marc0s_garc1a	201
1088118904	incendio	palabras clave	Controlan incendio de unidad del @MetrobusCDMX en Río Mayo #Precaución https://t.co/Ahcll53pKU	['controlar', 'incendio', 'unidad', 'río', 'mayo', 'san', 'raf	['Precau	['Metrobu	10	0	3	293063533	JORGE BECERRIL	MrElDiablo8	201
1088131677	tránsito	palabras clave	@BernardoBaranda @Alc_Iztapalapa @ClaraBrugadaM @CDMX_S	['deber', 'hacer él', 'conjunto', 'alcaldía', 'cuauhtemoc',	['contan	['Bernardo	0	0	0	177730590	manolo D-H	manolo_DH	201
1088416073	tránsito	palabras clave	@OVIALCDMX diario sobre Popocatépetl entre universidad y Cua	['diario', 'popocatépetl', 'universidad', 'cuauhtemoc', '	[]	['OVIALCD	0	2	0	4384974203	Maria	kuquisoto	201
1088607683	tránsito	palabras clave	@Alc_Iztapalapa @ClaraBrugadaM Cuando va a limpiar de carros	['ir', 'limpiar', 'carro', 'abandonado', 'eje', 'casi', 'esquin	[]	['Alc_Iztap	1	0	0	1024121354931	A David	MateoHe459887	201
1088622131	tránsito	palabras clave	@Claudiashein @andreslajous Señora Claudia, sobre av Santa Luc	['señora', 'claudia', 'av', 'santa', 'lucía', 'alcaldía', 'alvarc	[]	['Claudiash	3	0	1	117997817	MARCO ANTON	marcosilva_tv	201
1088815750	tránsito	palabras clave	@ALEIDAALAVEZ @Alc_Iztapalapa @Reforma @vozdeiztapalapa (	['requerir', 'limpie', 'eje', 'casi', 'esquina', 'guelatao', 'ca	[]	['ALEIDAA	4	1	3	1024121354931	A David	MateoHe459887	201
1088962452	tránsito	palabras clave	@Alc_Iztapalapa @Cul_Iztapalapa @ClaraBrugadaM En la Col. San	['col', 'santa', 'cruz', 'meyehualco', 'necesitar', 'sendero	[]	['Alc_Iztap	3	1	3	580408959	Gudeza	Gudeza_Guadalu	201
1089598831	tránsito	coordenadas	Cerrar calles en Domingo es Ridículo, pero cerrar Avenidas princip	['cerrar', 'calle', 'domingo', 'ridículo', 'cerrar', 'avenida',	[]	['Claudiash	0	0	0	392267545	ChARlY-C.U.c.h.(	charly_GLS	201
1089949842	incendio	palabras clave	Precaución incendio en la alcaldía de #Tlalpan corregidora y Aven	['precaución', 'incendio', 'alcaldía', 'corregidora', 'aven	['Tlalpar	[]	1	0	0	4882567385	ALTO IMPACTO	Muerxico	201
1089952860	incendio	palabras clave	⚠⚠#Precaución Se registra Incendio de Pastizal 🔥🔥en Av. de	['registrar', 'incendio', 'pastizal', 'av', 'torres', 'corregid	['Precau	[]	86	2	34	195951901	C5 CDMX	C5_CDMX	201
1090340920	tránsito	coordenadas	@Claudiashein @BJAlcaldia en ambos sentidos están en obras! Y	['ambos', 'obra', 'regulación', 'hacer él', 'hora', 'pico', 'e	[]	['Claudiash	0	0	0	278615043	Diana	NenaAlquicira	201
1090351949	incendio	palabras clave	En este pueblo d tlalpan ya comensaron a quemar los terrenos sir	['pueblo', 'tlalpan', 'comenzar', 'quemar', 'terreno', 'im	[]	['TlalpanVe	4	2	5	9061263346984	Sara Salazar	Sarasalazar100	201
1090669307	incendio	palabras clave	Precaución @ventelsdistrib el edificio de al lado está sacando hun	['precaución', 'edificio', 'lado', 'sacar', 'humo', 'tal', 'vez	['incend	['ventelsdi	0	0	0	135137502	Fausto santos H	FAUSANSMX	201
1090767518	tránsito	palabras clave	@OVIALCDMX, @transito_CDMX, Favor de pasar a infraccionar tra	['favor', 'pasar', 'infracción', 'trailer', 'bco', 'poco', '75ab	['70']	['OVIALCD	0	1	0	9030976958999	lxm2608	LXM2608	201
1090785552	tránsito	palabras clave	@OVIALCDMX todos los@días este es el@panorama en Popocat	['losa', 'popocatépetl', 'universidad', 'cuauhtemoc', 'ha	[]	['OVIALCD	0	1	0	4384974203	Maria	kuquisoto	201

Fig 11. Dataset de datos limpios.

5.3 Base de datos.

5.3.1 Modelo de la base de datos

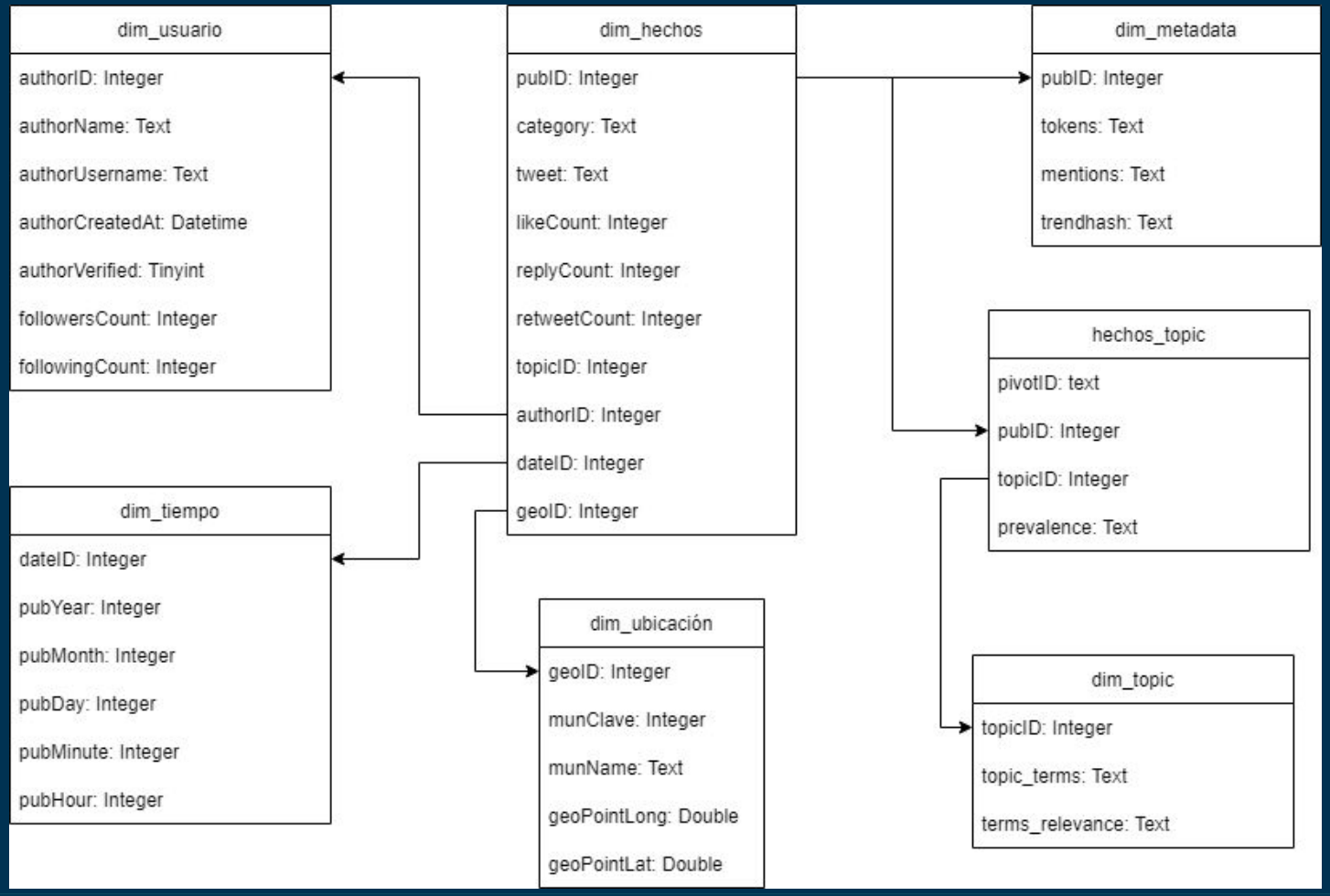


Fig 12. Modelo de la base de datos.

5.3.2 Creación de las tablas

Se crearon las diferentes tablas en archivos csv para poder cargarlas manualmente a MySQL.

```
INICIO
    Abrir archivos limpios
    Agregar los archivos  a un solo dataframe
    Crear diferentes dataframes correspondientes a cada tabla
    Guardar los dataframes como archivos csv
    Cargar los dataframes a MySQL.
FIN
```

Fig 13. Pseudocódigo de creación de base de datos.

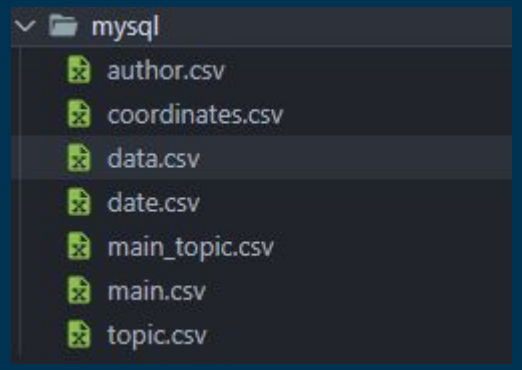


Fig 15. Tablas para la base de datos.

```python
df = pd.DataFrame()
for año in range(19,23):
    for mes in range (1,13):
        if mes < 10:
            df_aux = pd.read_csv(f'./csv_revisados/tweets_0{mes}{año}_c.csv')
        else:
            df_aux = pd.read_csv(f'./csv_revisados/tweets_{mes}{año}_c.csv')

        df = pd.concat([df,df_aux], sort=False, ignore_index=True)

main = df[['pubID', 'topicQuery', 'tweet', 'likeCount',
           'replyCount', 'retweetCount', 'authorID']]
data = df[['pubID', 'tokens', 'mentions', 'hashtags']]
author = df[['authorID', 'authorName', 'authorUsername',
           'authorCreatedAt', 'authorVerified', 'followersCount',
           'followingCount']]
date = df[['pubDate', 'pubYear', 'pubMonth', 'pubDay',
           'pubHour', 'pubMinute']]
coordinates = df[['geoID', 'geoName', 'longitude', 'latitude']]

idx = list(df.index)

main['tweet'] = main['tweet'].apply(lambda x: x.replace("\n", " "))
main['tweet'] = main['tweet'].apply(deEmojify)
main.insert(7, 'dateID', list(map(lambda x: 'd'+str(x+1), idx)))
main.insert(8, 'coordinateID', list(map(lambda x: 'c'+str(x+1), idx)))
author['authorName'] = author['authorName'].apply(deEmojify)
author['authorUsername'] = author['authorUsername'].apply(deEmojify)
date.insert(0, 'dateID', list(map(lambda x: 'd'+str(x+1), idx)))
coordinates.insert(0, 'coordinateID', list(map(lambda x: 'c'+str(x+1), idx)))

main.to_csv('./mysql/main.csv', index=False)
data.to_csv('./mysql/data.csv', index=False)
author.to_csv('./mysql/author.csv', index=False)
date.to_csv('./mysql/date.csv', index=False)
coordinates.to_csv('./mysql/coordinates.csv', index=False)
```

Fig 14. Código de creación de las tablas para la base de datos.

5.4 Topic Modeling - LDA

```python
def get_model(dictionary, corpus, texts, titulo):
    # Training the Model
    chunksize = len(texts)
    passes = 10
    iterations = 200
    eval_every = None
    temp = dictionary[0]

    NUM_TOPICS = get_topic_coherence_score(dictionary, corpus, texts, chunksize, passes, iterations)
    model = LdaModel(
        corpus=corpus,
        id2word=dictionary.id2token,
        chunksize=chunksize,
        alpha='auto',
        eta='auto',
        iterations=iterations,
        num_topics=NUM_TOPICS,
        passes=passes,
        eval_every=eval_every
    )

```

Fig 18. Código de creación de los modelos de LDA - topic modeling.

```python
# create n-grams
texts_with_bigrams = get_ngrams(texts, 2)
texts_with_trigrams = get_ngrams(texts, 3)

```

Fig 16. Código de creación de ngramas.

```python
# Create the dictionary
    dictionary = corpora.Dictionary(texts)
    dictionary.filter_extremes(no_below=int(len(texts)*0.01), no_above=0.5)
    corpus = [dictionary.doc2bow(text) for text in texts]
```

Fig 17. Código de creación de diccionario y corpus.

```
unigrama
        num_topics = 3 -> 0.4989
        num_topics = 4 -> 0.4946
bigramas
        num_topics = 4 -> 0.4883
        num_topics = 9 -> 0.4753
trigramas
        num_topics = 3 -> 0.5142
        num_topics = 4 -> 0.5067
```

Fig 19. Número de temas ideales para cada caso.

Fig 20. Nubes de palabras generadas a partir de topic modeling con trigrama para el tema 1.

- Advertencias para mejorar la circulación.
- Accidentes en avenidas.
- Principales alcaldías.
- Medios afectados.

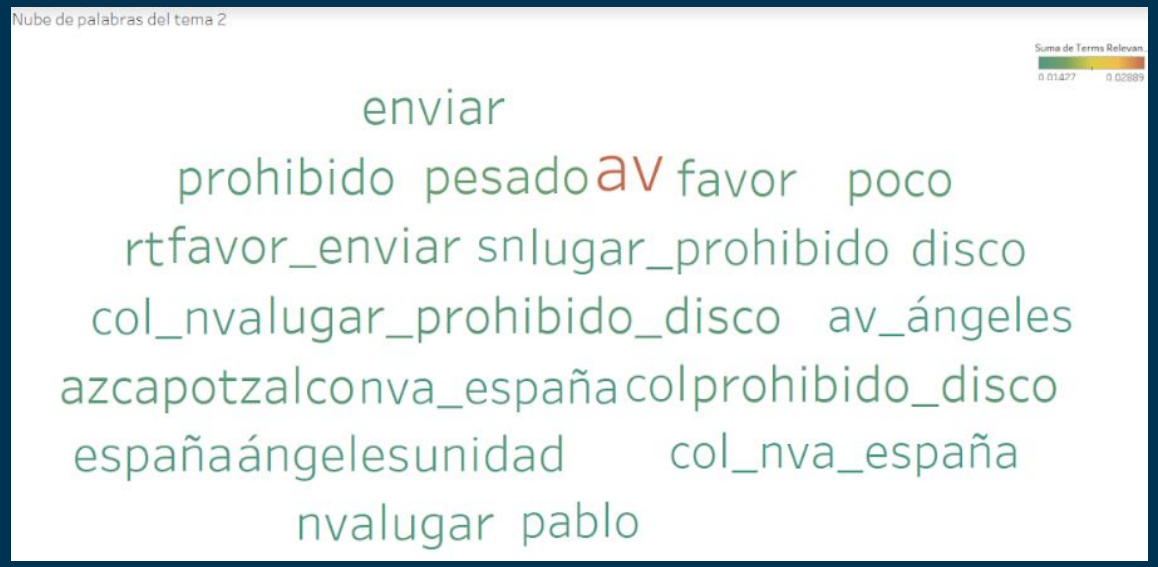


Fig 21. Nubes de palabras generadas a partir de topic modeling con trigrama para el tema 2.

- Solicitud de enviar unidades de tránsito.
- Denuncias sobre vehículos grandes estacionados en lugar prohibido obstruyendo el tránsito.
- Los reportes se concentran en la alcaldía de Azcapotzalco, colonia Nueva España avenida ángeles.

Fig 22. Mapa de la avenida ángeles, alcaldía Azcapotzalco.

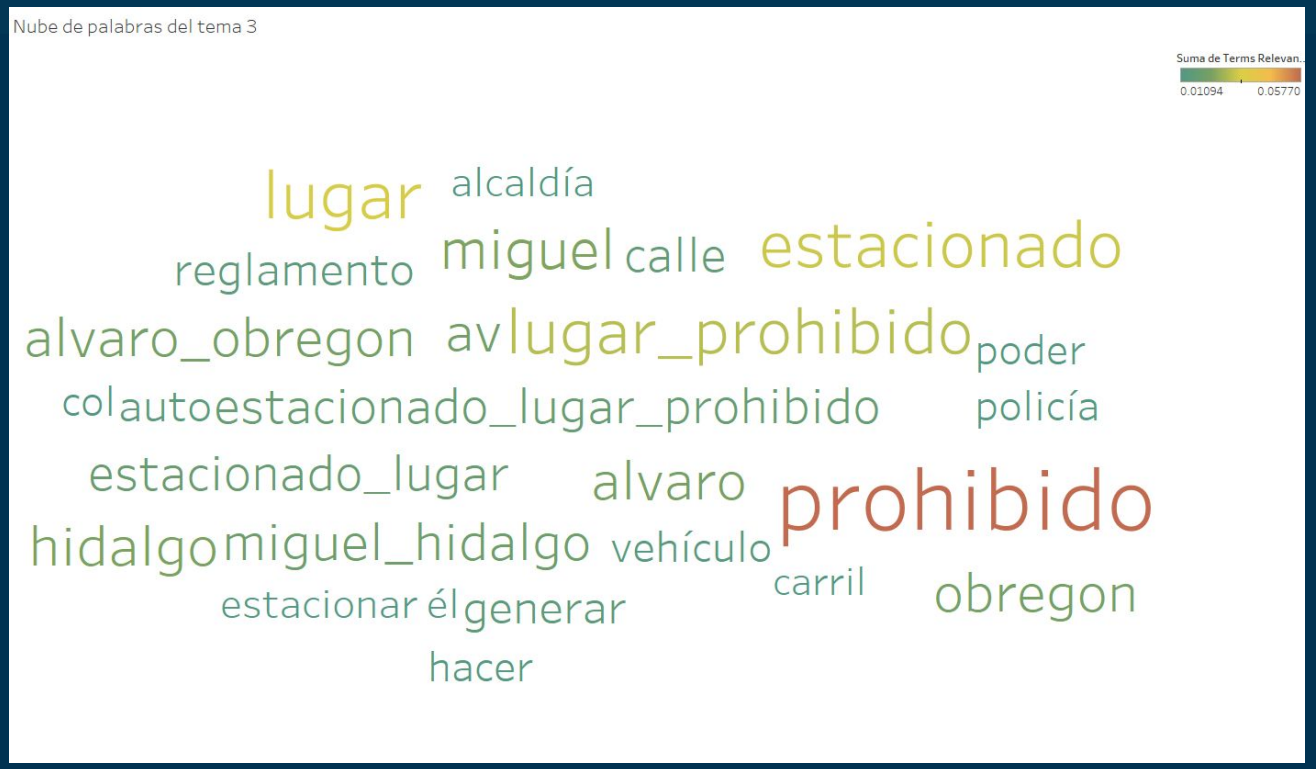


Fig 23. Nubes de palabras generadas a partir de topic modeling con trigrama para el tema 3.

- Zonas prohibidas para estacionarse.
- Faltas al reglamento de tránsito.
- Solicitudes de policías de tránsito.

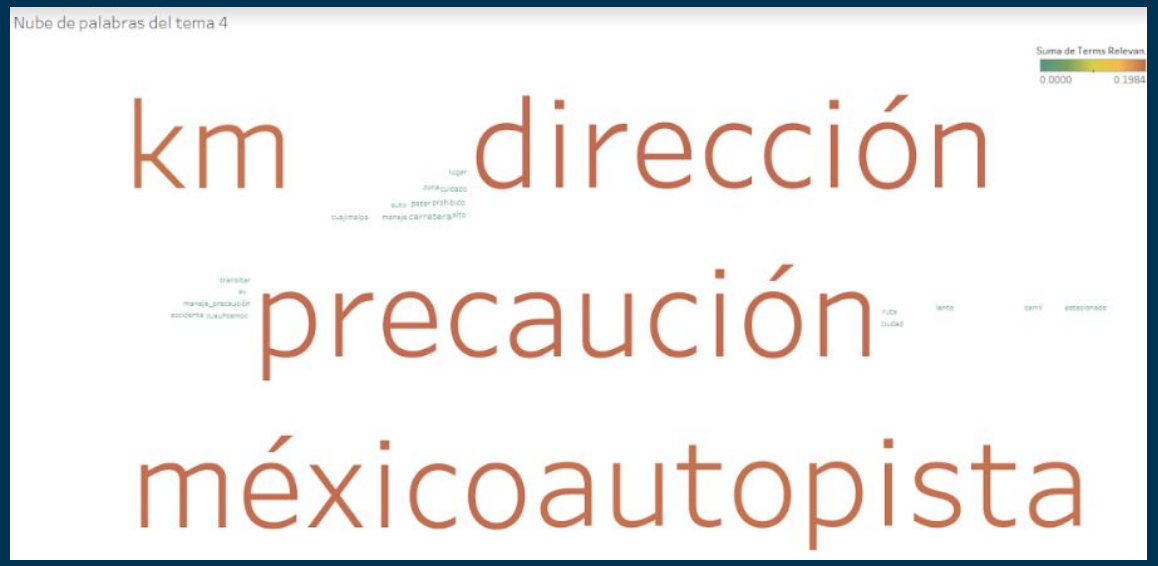


Fig 24. Nubes de palabras generadas a partir de topic modeling con trigrama para el tema 4.

- Advertencias para mejorar la circulación.
- Accidentes en autopistas saliendo de la ciudad.
- Autopistas como la México - Cuernavaca o México - Puebla.
- Lugar del accidente.

5.5 Análisis exploratorio con Tableau.

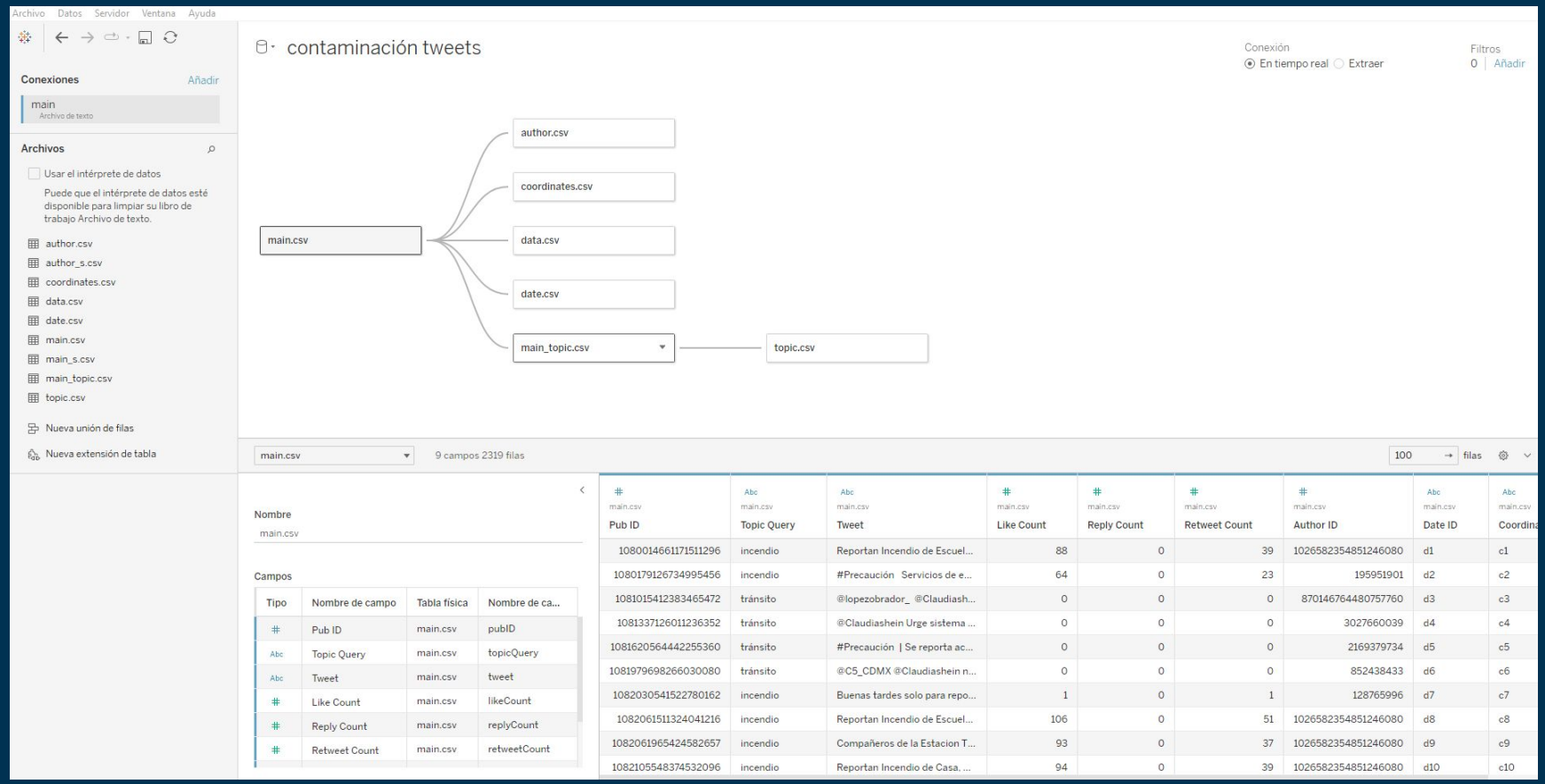


Fig 25. Captura de la base de datos en Tableau.

6. Visualización del Reporte

7. Conclusiones.

Este prototipo de software nos permitió realizar un análisis descriptivo en publicaciones extraídas de Twitter relacionados a contaminación ambiental, concretamente que afectan a la calidad del aire en la Ciudad de México, obteniendo una caracterización social, espacial y temporal sobre los tres temas seleccionados: tráfico, incendio y pirotecnia.

Con los resultados obtenidos en el análisis, se identificó el impacto que tienen estos eventos sobre los ciudadanos, a su vez la detección de zonas o alcaldías en las cuales suelen ocurrir con mayor frecuencia.

Referencias.

[1] M. Sammarco, R. Tse, G. Pau, G. Marfia, “Using geosocial search for urban air pollution monitoring,” Pervasive Mob. Comput., pp. 15-31, 2017. [En línea]. Disponible en: https://doi.org/10.1016/j.pmcj.2016.07.001

[2] K. A. Naranjo Análisis de correlación entre el índice de calidad del aire y el impacto en Twitter para la ciudad de Bucaramanga aplicando análisis de series temporales, extracción y procesamiento de lenguaje natural. [En línea]. Disponible en: http://hdl.handle.net/20.500.12749/15350.

[3] Porras Fresneda, Antonio Luis. (2019). Análisis del efecto en las redes sociales de las crisis de contaminación en Madrid. [En línea]. Disponible en: https://oa.upm.es/56734/

[4] G. Lorenzo, R. Salvatore, R. Francesco, V. Daniel, “From Tweets to Semantic Trajectories: Mining Anomalous Urban Mobility Patterns,” Citizen in Sensor Networks, pp. 26–35, 2013. [En línea]. Disponible en: https://doi.org/10.1007/978-3-319-04178-0_3

[5] Secretaría del Medio Ambiente de la Ciudad de México. (2020). “Calidad del aire en la Ciudad de México, Informe 2018,” Dirección General de Calidad del Aire, Dirección de Monitoreo de Calidad de Aire, Ciudad de México. [En línea]. Disponible en: http://www.aire.cdmx.gob.mx/descargas/publicaciones/informe-anual-calidad-del-aire-2018.pdf

Referencias.

[6] Sistema Nacional de Información Ambiental y de Recursos Naturales. (2015). Informe del Medio Ambiente. [En línea]. Disponible en :https://apps1.semarnat.gob.mx:8443/dgeia/informe18/tema/cap5.html#:~:text=Además%20de%20los%20efectos%20sobre,combinarse%20con%20el%20agua%20presente

[7] SEDEMA. (2021). Pide Sedema no quemar pirotecnia para evitar contingencias ambientales. [En línea]. Disponible en: https://www.sedema.cdmx.gob.mx/comunicacion/nota/pide-sedema-no-quemar-pirotecnia-paraevitar-contingencias-ambientales

[8] Instituto Nacional de Salud Pública. (2019). "Contaminación del aire, ambiente y salud." Instituto Nacional de Salud Pública, Ciudad de México. Disponible en: https://insp.mx/assets/documents/webinars/2021/CISP_Contaminación%20del%20aire%20(19%20oct)%204.pdf

[9] Allan, J. (Vol. Ed.), (2012). Topic detection and tracking: Event-based information organization: 12Springer Science & Business Media.

[10] Zheng, Y., Capra, L., Wolfson, O., & Yang, H. (2014). Urban computing: Concepts, methodologies, and applications. ACM Transactions on Intelligent Systems and Technology (TIST), 5(3), 38.

[11] Zagal-Flores, R., Felix Mata, M., Claramunt, C. (2018). From What and When Happen, to Why Happen in Air Pollution Using Open Big Data. In: R. Luaces, M., Karimipour, F. (eds) Web and Wireless Geographical Information

Sesión de preguntas y respuestas
Gracias.