TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE NUEVO LAREDO

"Con la ciencia por la humanidad…"

INGENIERÍA EN SISTEMAS COMPUTACIONALES

Analítica de Big Data

PROYECTO FINAL

ACTIVIDAD EN REDES SOCIALES: LACTANCIA MATERNA Y FÓRMULA

Ing. José Antonio Espino López

Equipo "Los Nuggets"

Alumno(s):

Valeria Margarita Espinoza Sánchez
18100168

José Arcadio Rodríguez Matta
18100227

Nuevo Laredo, Tam. Noviembre, 2022

ÍNDICE

INTRODUCCIÓN

Se ha demostrado que la lactancia materna es la mejor y más completa nutrición para los recién nacidos, ya que este método promueve la salud del bebé y favorece su crecimiento (Academia Americana de Pediatría [AAP], 2005). Los medios masivos de comunicación han sido considerados como medios universales con el potencial de influir en las normas sociales. Así, este estudio pretende explorar cuantitativa y cualitativamente, la actividad que se presenta acerca de la alimentación mediante lactancia materna y mediante fórmula.

Durante la presentación del presente estudio, se expondrán en primer lugar los objetivos generales y específicos que se pretenden lograr al finalizar la analítica de los datos recopilados en distintos formatos, estos, en su mayoría se componen de comentarios y publicaciones de distintas redes sociales, incluidos foros de discusión, Twitter, Facebook, etc.

En el proceso del estudio se hará uso de distintas herramientas que servirán para hacer un análisis de los datos recopilados, entre estas se encuentran Python, Anaconda, Jupyter Lab, y algunas librerías que harán posible analizar más de 270, 000 registros de comentarios públicos en las redes sociales durante el periodo de tres meses a partir del 20 de Junio del 2022, etiquetados específicamente con las palabras clave "breastfeeding" y "formula milk", teniendo como resultado un conjunto de indicadores y gráficas que se puedan interpretar y comparar de forma que se obtengan conclusiones acerca de este tópico.

OBJETIVO

La realización del proyecto busca cumplir los siguientes objetivos:

- Buscar el día con mayor actividad dentro de un plazo de 3 meses.
- Obtener el día de la semana con más publicaciones realizadas.
- Ver los países que generan mayor cantidad de contenido.
- Conocer la hora en la cual se genera mayor contenido relacionado.

Se espera que mediante dichas métricas se pueda obtener un punto de vista más preciso acerca de la influencia de las redes sociales, específicamente los comentarios o posts en redes sociales, sobre las decisiones de las madres acerca de la alimentación temprana de un niño, esto es, por ejemplo, la decisión entre la lactancia materna y por fórmula.

Este estudio podría tener un gran impacto a futuro, ya que el uso de redes sociales es exponencial en función de la edad de las madres modernas, es decir, veremos una brecha generacional en la crianza entre las madres que hayan nacido durante épocas modernas en donde abunda la información en internet, y las madres que tenían métodos distintos al criar a sus hijos durante épocas pasadas, generalmente estos métodos se pasaban de generación en generación.

MARCO TEÓRICO

LIMPIEZA DE DATOS

El Data Cleansing sirve para analizar, identificar y corregir datos en bruto que están desordenados, equivocados y mal procesados. El proceso de limpieza de datos trata de completar los valores faltantes, corregir errores y determinar si toda la información está en las filas y columnas correctas.

El análisis de datos de una compañía siempre debe comenzar con un proceso de limpieza de datos exhaustivo para tomar decisiones estratégicas. Además, el procesamiento y limpieza de datos es fundamental para un análisis de datos eficiente, preciso y efectivo.

- ➔ La limpieza de datos en la big data aplicada a los negocios elimina los principales errores e inconsistencias que aparecen cuando se incorporan múltiples fuentes de datos en un solo conjunto de datos.
- ➔ El uso de herramientas para Data Cleansing o limpieza de datos hará que todos los miembros de tu equipo sean más eficientes al momento de obtener rápidamente los datos que realmente necesitan.
- ➔ Los métodos de limpieza de datos te brindan menos errores, y eso significa, clientes más felices y trabajadores menos frustrados.
- ➔ Las diferentes funciones de la limpieza de datos te permiten comprender mejor qué se pretende hacer con los datos y saber de dónde provienen.
- ➔ Las etapas de limpieza de datos te permiten mejorar la calidad de los datos.
- ➔ La utilización de la tecnología del Data Cleansing te otorga una mejor eficiencia y productividad interna. Cuando la información se limpia adecuadamente, revela información valiosa sobre las necesidades y los procesos.

HERRAMIENTAS PARA EL DESARROLLO DEL PROYECTO

JUPYTERLAB

JupyterLab es una aplicación web que ofrece todo un entorno de trabajo interactivo ideal para trabajar en el ámbito científico. Su utilidad más impactante es la posibilidad de crear Jupyter Notebooks, que combinan diversos elementos como código

interactivo, textos, ecuaciones, imágenes y otras fuentes de datos que se pueden manejar en proyectos científicos de todo tipo.

JupyterLab es extremadamente potente y flexible, ya que permite usar una increíble cantidad de fuentes desde texto, imágenes, hojas de cálculo, HTML, PDF, LaTeX y mucho más. Paralelamente permite trabajar con código escrito en cantidad de lenguajes populares como Python, R o Scala, editando y ejecutando el código sin salirnos de la propia aplicación. Además es capaz de usar diversos tipos de consolas de comandos y almacenar scripts para su ejecución.

Con todo, es una herramienta excelente para integrar en diversos flujos de trabajo de data-science, ya que es capaz de ajustarse a las necesidades más diversas de los profesionales. Además integra toda una serie de herramientas modernas que permiten incrementar la colaboración entre los profesionales, publicar y revisar código entre personas de todo el mundo, en flujos como los que nos proporcionan GitHub o GitLab.

PYTHON

Python es un lenguaje de programación de alto nivel que se utiliza para desarrollar aplicaciones de todo tipo. A diferencia de otros lenguajes como Java o .NET, se trata de un lenguaje interpretado, es decir, que no es necesario compilarlo para ejecutar las aplicaciones escritas en Python, sino que se ejecutan directamente por el ordenador utilizando un programa denominado interpretador, por lo que no es necesario "traducirlo" a lenguaje máquina.

Python es un lenguaje sencillo de leer y escribir debido a su alta similitud con el lenguaje humano. Además, se trata de un lenguaje multiplataforma de código abierto y, por lo tanto, gratuito, lo que permite desarrollar software sin límites. Con el paso del tiempo, Python ha ido ganando adeptos gracias a su sencillez y a sus amplias posibilidades, sobre todo en los últimos años, ya que facilita trabajar con inteligencia artificial, big data, machine learning y data science, entre muchos otros campos en auge.

ANACONDA

Anaconda es una suite de código abierto que abarca una serie de aplicaciones, librerías y conceptos diseñados para el desarrollo de la ciencia de datos con Python. Se trata de una distribución de Python que básicamente funciona como un gestor de entorno, de paquetes y que posee una colección de más de 720 cuya característica primordial es que son de código abierto.

Anaconda Distribution se agrupa en cuatro sectores o soluciones tecnológicas:

- Anaconda Navigator: Interfaz gráfica de Anaconda Python
- Anaconda Project
- Librerías de Ciencia de Datos
- Conda: Gestor de código del Anaconda Python.

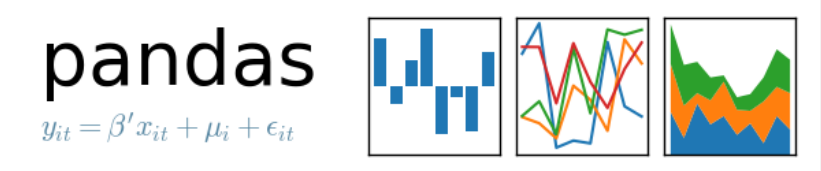
Características de Anaconda Python

Esta suite para la ciencia de datos con Python cuenta con una gran cantidad de características entre las que podemos resaltar las siguientes:

- Libre, de código abierto, cuenta con una documentación bastante detallada y una gran comunidad.
- Multiplataforma
- Permite instalar y administrar paquetes, dependencias y entornos para la Ciencias de Datos con Python de una manera muy sencilla.
- Ayuda a desarrollar proyectos de Ciencia de datos utilizando diversos entornos de desarrollo como Jupyter, JupyterLab, Spyder y RStudio.
- Cuenta con herramientas como Dask y Numba para analizar datos.
- Simplifica de manera acelerada la implementación de proyectos de Ciencia de Datos.

PANDAS

Pandas es una librería de Python especializada en el manejo y análisis de estructuras de datos.



Las principales características de esta librería son:

Define nuevas estructuras de datos basadas en los arrays de la librería NumPy pero con nuevas funcionalidades.

- ➔ Permite leer y escribir fácilmente ficheros en formato CSV, Excel y bases de datos SQL.
- ➔ Permite acceder a los datos mediante índices o nombres para filas y columnas.
- ➔ Ofrece métodos para reordenar, dividir y combinar conjuntos de datos.
- ➔ Permite trabajar con series temporales.
- ➔ Realiza todas estas operaciones de manera muy eficiente.

Pandas dispone de tres estructuras de datos diferentes:

- Series: Estructura de una dimensión.
- DataFrame: Estructura de dos dimensiones (tablas).
- Panel: Estructura de tres dimensiones (cubos).

Estas estructuras se construyen a partir de arrays de la librería NumPy, añadiendo nuevas funcionalidades.

NUMPY

NumPy es un módulo de Python. El nombre es un acrónimo de Python Numérico. Es una librería que consiste en objetos de matrices multidimensionales y una colección de rutinas para procesar esas matrices.

Es un módulo de extensión para Python, escrito en su mayor parte en C. Esto asegura que las funciones y funcionalidades matemáticas y numéricas precompiladas de NumPy garantizan una gran velocidad de ejecución.

NumPy es un paquete de procesamiento de matrices de uso general. Proporciona un objeto de matriz multidimensional de alto rendimiento, y herramientas para trabajar con estas matrices. Es el paquete fundamental para la computación científica con Python. Además de sus obvios usos científicos, NumPy también puede ser usado como un eficiente contenedor multidimensional de datos genéricos.

Características de NumPy

- Está escrito en C, lo que le proporciona una velocidad muy alta, y cuando trabajamos con grandes conjuntos de datos la velocidad es un punto fundamental.
- Incluye funciones para operaciones de muchos tipos, matemáticas, de lógica, de ordenación, estadísticas, de entrada y salida para leer y escribir ficheros, etcétera. Es una librería bastante amplia y unida a Pandas ambas en conjunto son muy potentes..
- Es muy usada en el mundo del Data Science, cualquier persona que trabaje en este mundo, probablemente usará a diario la librería NumPy.

PLOTLY

Es una biblioteca de gráficos para Python, interactiva y basada en el navegador. Construida sobre plotly.js, se trata de una biblioteca de gráficos declarativa de alto nivel. plotly.js se entrega con más de 30 tipos de gráficos, incluyendo gráficos científicos, gráficos 3D, gráficos estadísticos, mapas SVG, gráficos financieros y más.

Se encuentra disponible para trabajar con su conjunto de herramientas en los lenguajes de programación Python, R y JavaScript. Por supuesto, cada librería con sus peculiaridades dada la adaptación a la gramática y las características de cada uno de los lenguajes.

Existen múltiples maneras de evaluar la calidad de una librería. Después de probar Plotly y conociendo su enorme popularidad (recordamos que el plugin Data Plotly para QGIS integra esta librería para visualizar datos dentro de su interfaz) podrían destacar algunos aspectos como:

- **Sencillez**. Los gráficos no pretenden ser espectaculares, más bien son sencillos a pesar de las múltiples posibilidades de personalización. En su diseño no existen elementos gráficos pesados, que llamen la atención. Se trata de visualizaciones modernas, muy en la línea de ggplot2 para R, por ejemplo, o Seaborn para Python.

- **Integración**. La ventaja de utilizar Plotly es su integración en múltiples lenguajes y plataformas, así como la facilidad que ofrece para compartir los gráficos, mapas y visualizaciones de datos creados con esta librería.

- **Variedad**. El abanico de posibilidades de visualización y personalización de los gráficos y los mapas es abrumador. Existen cientos de gráficos distintos que aportan una gran flexibilidad a la hora de realizar visualizaciones de datos.

- **Interactividad**. Este es quizás el punto más interesante de todos. Los gráficos y mapas son interactivos. Podemos jugar con el hover y el zoom, la selección múltiple y única de elementos, entre otros… Básicamente Plotly genera figuras habilitadas para interaccionar con los datos plasmados en el gráfico o mapa, desplazarse por ellos e ir al detalle. Con ello, se ofrece la posibilidad de conocer en mayor profundidad los datos, destacar algunos de ellos, modificar sus visualizaciones, etcétera, en comparación con un gráfico estático.

MATPLOTLIB

Matplotlib es una librería de Python especializada en la creación de gráficos en dos dimensiones. Permite crear y personalizar los tipos de gráficos más comunes, entre ellos:

- Diagramas de barras
- Histograma
- Diagramas de sectores
- Diagramas de caja y bigotes
- Diagramas de violín
- Diagramas de dispersión o puntos
- Diagramas de líneas
- Diagramas de áreas
- Diagramas de contorno
- Mapas de color

y combinaciones de todos ellos.

JUSTIFICACIÓN

Dentro del marco de la analítica de datos, uno de los objetivos más importantes que se encuentran asociados a este proyecto es la creación de un modelo de datos que sirva a manera de una plantilla, misma que se podrá aplicar nuevamente a versiones actualizadas de los sets de datos utilizados. Esto con el propósito de tener una visión más completa del tópico que se presenta en este documento, de esta forma se pretende llegar a conclusiones más acertadas con respecto a la actividad que se genera sobre estos temas.

Algunos de los nuevos conocimientos que el presente proyecto puede aportar en el presente y futuro son los siguientes:

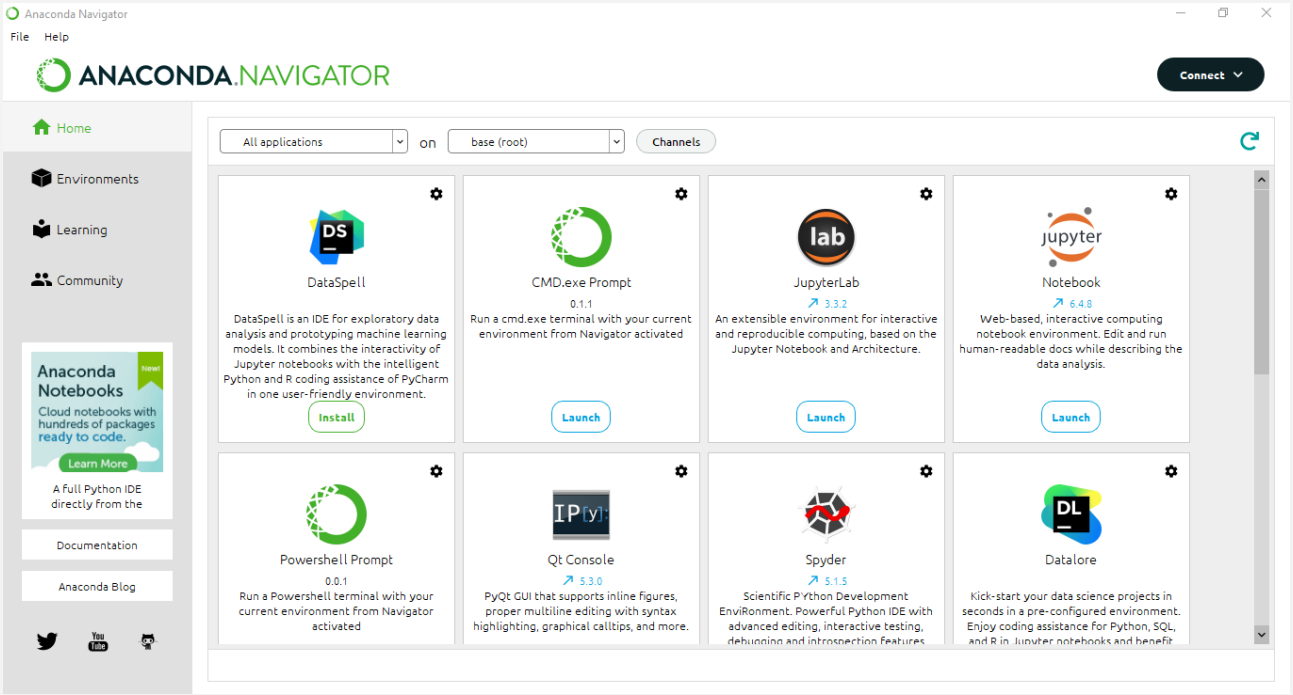
- La relación entre las palabras o frases que se utilizan con respecto a un tema en específico en una época determinada del año.
- El área geográfica desde la cual se generan en mayor cantidad un cierto tipo de posts, comentarios, o cualquier tipo de contenido textual en redes sociales.
- La decisión o tendencias populares respecto a la alimentación temprana según el contenido que se consume en redes sociales.
- Cómo es la actividad que se presenta en ambos temas dentro de las redes sociales.

Se hará uso de tecnologías que tienen un gran impacto en la actualidad, creando así un modelo que puede ser mantenible y escalable durante el tiempo, ya que las herramientas que se utilizan son relativamente recientes, así como de gran demanda y uso por la comunidad de analítica de los datos.

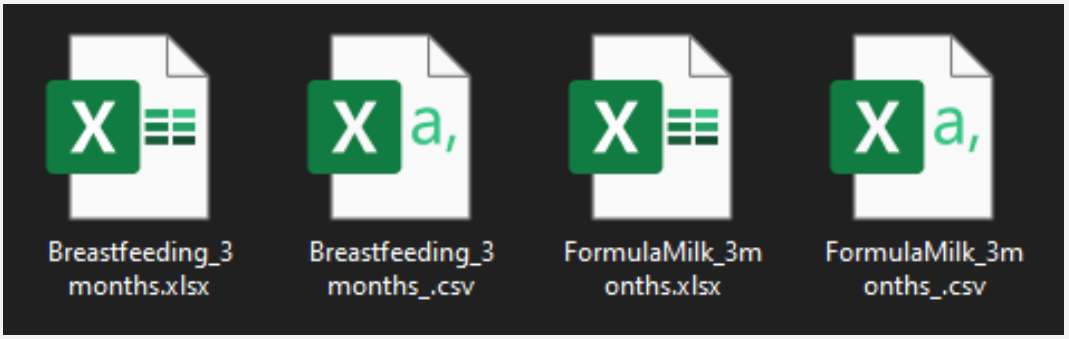
DESARROLLO Y RESULTADOS

El proyecto de desarrolla en el siguiente entorno:

- Anaconda, utilizando JupyterLab, escrito en Python con las librerías **Pandas, Matplotlib, Numpy y Plotly** para el muestreo y análisis de los datos



Conjuntos de datos (Datasets)



Los datasets obtenidos abarcan un periodo de tres meses, desde julio hasta septiembre. Estos datasets tienen dos formatos, el primero es .xlsx que es el formato de Excel y el .csv que pertenece a un archivo donde la información está separada por comas, conteniendo las siguientes columnas:

- **status_id**: ID numérico único asignado a cada registro.
- **created_at**: La fecha y hora en que se publicó

- **text**: El texto del comentario/publicación
- **display_text_width**: Largo del texto del post (número de caracteres)
- **country**: País donde se publicó
- **day**: Día de la semana en donde se publicó

Muestra de la información:

A	B	C	D	E	F
status_id	created_at	text	display_text_width	country	day
1538998114270461 95	2022-06-20 21:31:58 UTC	@TaylorLyonsMSJ HAHAHAHA. Sister, y	181	United States	lunes
1539010386506162 17	2022-06-20 22:20:44 UTC	Gloria Dudney starting the #TNBFSym2	160	United States	lunes
1539026425377869 82	2022-06-20 23:24:28 UTC	@KateNicholl @BelTel @SuzyJourno I'	74	United Kingdom	lunes
1539065335239819 26	2022-06-21 01:59:05 UTC	on the fence, a part of me is DONE brea	117	United States	martes
1539100120696115 20	2022-06-21 04:17:18 UTC	Lupig pag buntis aning breastfeeding u	55	Republic of the Philippines	martes
1539127189697970 17	2022-06-21 06:04:52 UTC	@PoolsideGaGirl @Docmaker63 @Ka	76	United Kingdom	martes
1539128130929147 90	2022-06-21 06:08:36 UTC	@PoolsideGaGirl @Docmaker63 @Ka	34	United Kingdom	martes
1539246405717573 63	2022-06-21 13:58:35 UTC	....and suddenly I miss breastfeeding (	39	South Africa	martes
1539259459918106 62	2022-06-21 14:50:27 UTC	Our breastfeeding journey has been ha	145	United States	martes

Análisis de los datasets referentes a Breastfeeding

Para el desarrollo del proyecto, primeramente se realizó la importación de las librerías **numpy, pandas, plotly** y **malplotlib**, que permitirán manipular datos y obtener información en base a ellos.

Se debe de realizar un recorrido para obtener los documentos que se encuentran en la carpeta donde están almacenados los datasets, con el fin de ver que éstos sean detectados de forma correcta por Python.

```
[2]: import numpy as numpy #librería de algebra lineal
     import pandas as panda #librería para el procesamiento de datos
     #librería para crear gráficas
     import plotly.express as plotly #sofisticadas
     import matplotlib.pyplot as matplot #sencillas
     import re #librería para hacer uso de expresiones regulares

     import os #se importan los datasets con los que se trabajaran
     for dirname, _, filenames in os.walk('C:/Users/Valeria/Documents/9no Semestre/Analítica de Big Data/Proyecto Final/Breastfeeding/'):
         for filename in filenames:
             print(os.path.join(dirname, filename))

     C:/Users/Valeria/Documents/9no Semestre/Analítica de Big Data/Proyecto Final/Breastfeeding/Breastfeeding_3months.xlsx
     C:/Users/Valeria/Documents/9no Semestre/Analítica de Big Data/Proyecto Final/Breastfeeding/Breastfeeding_3months_.csv
     C:/Users/Valeria/Documents/9no Semestre/Analítica de Big Data/Proyecto Final/Breastfeeding/FormulaMilk_3months.xlsx
     C:/Users/Valeria/Documents/9no Semestre/Analítica de Big Data/Proyecto Final/Breastfeeding/FormulaMilk_3months_.csv
```

Al verificar la existencia de los archivos, se seleccionará el dataset con el cual se requiera trabajar, por lo tanto, se eligió *Breastfeeding_3months_.csv*

En el siguiente apartado se realiza la lectura del archivo csv, y para comprobar la información se utiliza el método Head() para mostrar una parte de esta (por defecto devuelve los primeros 5 registros):

```
[4]: #Se realiza la lectura del archivo que se analizará
     dataset = panda.read_csv('C:/Users/Valeria/Documents/9no Semestre/Analítica de Big Data/Proyecto Final/Breastfeeding/Breastfeeding_3months_.csv')

[5]: dataset.head() #Se muestra una parte de la información
```

[5]:

	status_id	created_at	text	display_text_width	country	day
0	1538941385809747968	2022-06-20 17:46:33 UTC	Manic Mondays \n\nFrom the 16th century onwar...	140	NaN	Monday
1	1538941854649032705	2022-06-20 17:48:24 UTC	southern softie commented on MailOnline: What ...	196	NaN	Monday
2	1538942219645861892	2022-06-20 17:49:51 UTC	MomToBe Women's Rayon Maternity Dress/Easy Bre...	163	NaN	Monday
3	1538942452723335169	2022-06-20 17:50:47 UTC	@AlfredMwandagha Good for a breastfeeding bunn...	33	NaN	Monday
4	1538942645011030016	2022-06-20 17:51:33 UTC	We know more about cow's milk than human milk....	140	NaN	Monday

Posteriormente, como parte de la obtención de los datos, se comprueban los tipos de datos que contiene cada campo:

```
[6]: dataset.info() #Devuelve la información de las columnas

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 238045 entries, 0 to 238044
Data columns (total 6 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   status_id           238045 non-null  int64
 1   created_at          238045 non-null  object
 2   text                228058 non-null  object
 3   display_text_width  238045 non-null  int64
 4   country             485 non-null     object
 5   day                 238045 non-null  object
dtypes: int64(2), object(4)
memory usage: 10.9+ MB
```

También se buscan la cantidad de registros que tengan alguna columna en blanco / nulos:

```
[7]: dataset.isna().sum() #Busca las columnas que tengan valores nulos

[7]: status_id                0
     created_at               0
     text                  9987
     display_text_width       0
     country             237560
     day                      0
     dtype: int64
```

- ➔ Para la columna *created_at*, se encontró que tiene un tipo de dato “object”, por lo tanto hay que volver a comprobar con el método **type().**
- ➔ Al devolver string, se concluye que tiene que convertirse el tipo de dato de dicha columna, ya que sabemos que es una fecha y hora, para dicha tarea se utiliza **to_datetime()**:

```
[8]: #Devuelve el tipo de objeto en la columna 'created_at'
     type(dataset.at[0, 'created_at'])
     #El valor que devuelve es str

[8]: str

[9]: dataset['created_at'] = panda.to_datetime(dataset['created_at']) #Cambia de string datetime a datetime

[10]: dataset['created_at'].head() #Muestra las primeras fechas en formato string datetime

[10]: 0   2022-06-20 17:46:33+00:00
      1   2022-06-20 17:48:24+00:00
      2   2022-06-20 17:49:51+00:00
      3   2022-06-20 17:50:47+00:00
      4   2022-06-20 17:51:33+00:00
      Name: created_at, dtype: datetime64[ns, UTC]
```

Ahora a manera informativa, se obtienen todos los países que se encuentren en el dataset (sin repetir, mediante el método **unique()** a la columna ***country***):

```
[27]: dataset['country'].unique()

[27]: array([nan, 'United States', 'United Kingdom',
             'Republic of the Philippines', 'South Africa', 'Zimbabwe', 'India',
             'Rwanda', 'Canada', 'Ireland', 'Australia', 'Kenya', 'Vietnam',
             'Malaysia', 'East Timor', 'Denmark', 'Indonesia', 'Nigeria',
             'The Netherlands', 'Italy', 'Pakistan', 'Germany', 'Jamaica',
             'Switzerland', 'Norway', 'Singapore', 'Spain', 'Mexico',
             'Kingdom of Saudi Arabia', 'Ghana', 'Uganda', 'Thailand', 'Cyprus',
             'Trinidad and Tobago', 'Botswana', 'Portugal', 'Namibia',
             'New Zealand', 'Republic of Slovenia', 'Ethiopia',
             'United Arab Emirates', 'Latvia', 'Maldives', 'Zambia',
             'Dominican Republic', "People's Republic of China"], dtype=object)
```

Para el siguiente bloque se utiliza la librería **PyCountry**, una biblioteca de Python para completar los datos de los países a partir de la base de datos de países para preparar el conjunto de datos para su comparación o agregación con otros.

- ➔ La variable ***mapeo*** se llena con el código en tres siglas de cada país
- ➔ La variable ***mapeo_banderas*** se llena con la bandera de cada país

```
import pycountry
mapeo = {country.name: country.alpha_3 for country in pycountry.countries} #alpha_3 cambia el nombre del país por sus 3 siglas
mapeo_banderas = {country.name: country.flag for country in pycountry.countries} #Obtiene las banderas de los países
```

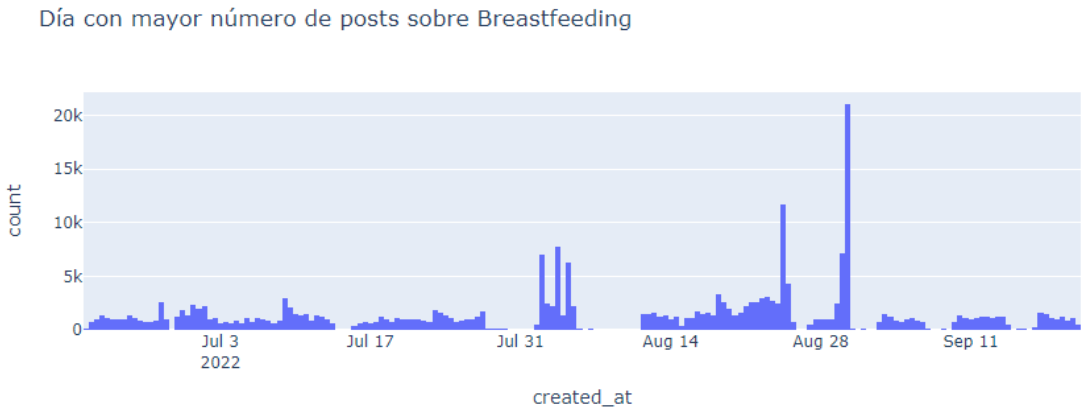
```
#Se renombran los nombres de los países a los cuales no se pueden obtener su código
dataset['country'].replace(['Republic of the Philippines', 'Vietnam', 'East Timor','The Netherlands',
'Kingdom of Saudi Arabia', 'Republic of Slovenia', "People's Republic of China"],
['Philippines', 'Viet Nam', 'Timor-Leste', 'Netherlands', 'Saudi Arabia', 'Slovenia', 'China'],regex = True, inplace = True)
#regex permite expresiones regulates e inplace permite aplicar los cambios directamente al dataset
```

- ➔ Al dataset se agregan dos columnas: **country_code** y **country_flag** y se llenan con el contenido de **mapeo** (códigos en tres siglas) y **mapeo_banderas** (imágenes de banderas) respectivamente.
- ➔ Se comprueba dicha información consultando las dos columnas con **unique()**

```
[44]: dataset = dataset.assign(country_code = None) #Se agrega la columna country_code
      dataset = dataset.assign(country_flag = None) #Se agrega la columna country_flag
      for index in dataset.index:   #Realiza recorrido por filas del dataset
          country = dataset.at[index, 'country']  #country es igual al elemento en index de la columna country
          dataset.at[index, 'country_code'] = mapeo.get(country) #Se coloca en el dataset las siglas del país
          dataset.at[index, 'country_flag'] = mapeo_banderas.get(country) #Se coloca en el dataset las banderas

[45]: dataset['country_code'].unique() #Obtiene de la columna country_code los elementos no repetidos, siendo 45 países

[45]: array([None, 'USA', 'GBR', 'PHL', 'ZAF', 'ZWE', 'IND', 'RWA', 'CAN',
             'IRL', 'AUS', 'KEN', 'VNM', 'MYS', 'TLS', 'DNK', 'IDN', 'NGA',
             'NLD', 'ITA', 'PAK', 'DEU', 'JAM', 'CHE', 'NOR', 'SGP', 'ESP',
             'MEX', 'SAU', 'GHA', 'UGA', 'THA', 'CYP', 'TTO', 'BWA', 'PRT',
             'NAM', 'NZL', 'SVN', 'ETH', 'ARE', 'LVA', 'MDV', 'ZMB', 'DOM',
             'CHN'], dtype=object)

[46]: dataset['country_flag'].unique() #Obtiene de la columna country_code los elementos no repetidos, siendo 45 países

[46]: array([None, 'US', 'GB', 'PH', 'ZA', 'ZW', 'IN', 'RW', 'CA', 'IE', 'AU',
             'KE', 'VN', 'MY', 'TL', 'DK', 'ID', 'NG', 'NL', 'IT', 'PK', 'DE',
             'JM', 'CH', 'NO', 'SG', 'ES', 'MX', 'SA', 'GH', 'UG', 'TH', 'CY',
             'TT', 'BW', 'PT', 'NA', 'NZ', 'SI', 'ET', 'AE', 'LV', 'MV', 'ZM',
             'DO', 'CN'], dtype=object)
```

Con el siguiente bloque se crea un histograma, que permite ver el día con mayor número de posts generados en relación al tema *Breastfeeding*. Para ello se utiliza **Plotly** con el método .**histogram** donde se usa la columna **created_at** para obtener los días donde se realizaron publicaciones a lo largo de 3 meses.

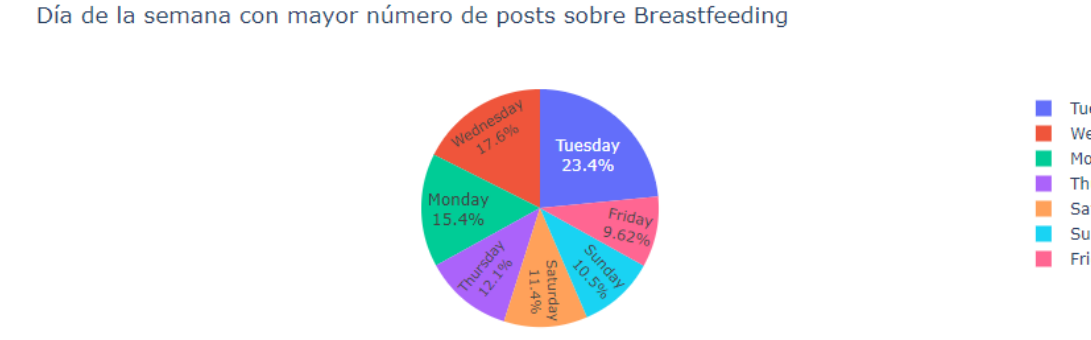
```
[93]: figura = plotly.histogram(dataset, x = 'created_at', title='Día con mayor número de posts sobre Breastfeeding')
      figura.show()
```



Con la siguiente gráfica se hace un análisis sobre el día de la semana con mayor número de posts sobre *Breastfeeding*, para ello:

- ➔ En la primera línea se genera una agrupación utilizando **day** y su tamaño.
- ➔ En la segunda se ordenaron de acuerdo al día que mayor número de posts generados
- ➔ Por último, se genera la gráfica utilizando **Plotly** y su método de **pie**, colocando los valores que se graficarán.

```
[92]: pie = dataset.groupby(['day']).size().reset_index(name = 'size')
      pie.sort_values(by='size', ascending=False ,inplace=True)
      figura = plotly.pie(pie, values='size', names='day',
                  title='Día de la semana con mayor número de posts sobre Breastfeeding',
                  hover_data=['size'])
      figura.update_traces(textposition='inside', textinfo='percent+label')
      figura.show()
```



En el siguiente bloque de código se genera un conjunto de datos de los países con mayor número de publicaciones realizadas. Para ello:

- ➔ Se crea una agrupación llamada **dtMundial**, donde se agrupa utilizando **country**, **country_code** y **country_flag**, obteniendo su tamaño.

➔ Se organizan los valores de **dtMundial** por la columna **size**.

➔ Por último, se muestra una previsualización de los valores de la agrupación.

```
[85]: dtMundial = dataset.groupby(['country', 'country_code', 'country_flag']).size().reset_index(name = 'size')
      dtMundial.sort_values(by='size', ascending=False ,inplace=True)

[86]: dtMundial.head()
```

[86]:

	country	country_code	country_flag	size
41	United States	USA	US	213
40	United Kingdom	GBR	GB	81
15	Kenya	KEN	KE	23
32	South Africa	ZAF	ZA	18
10	India	IND	IN	17

Al generar la agrupación, se crea la gráfica de un mapa mundial que mostrará la ubicación geográfica de los países donde se hacen más publicaciones sobre *Breastfeeding*, utilizando **size** para poder graficar sobre el mapa.

```
[87]: figura = plotly.scatter_geo(dtMundial, locations="country_code", color="country",
                                  hover_name="country", size="size",
                                  projection="natural earth1",title="Países con más publicaciones de Breastfeeding")
      figura.show()
```

Para el siguiente bloque de código, se crea una agrupación de la cantidad de publicaciones que se realizan en cada hora de un día, tomando en cuenta:

➔ De la columna **created_at** se agrupa utilizando la hora.

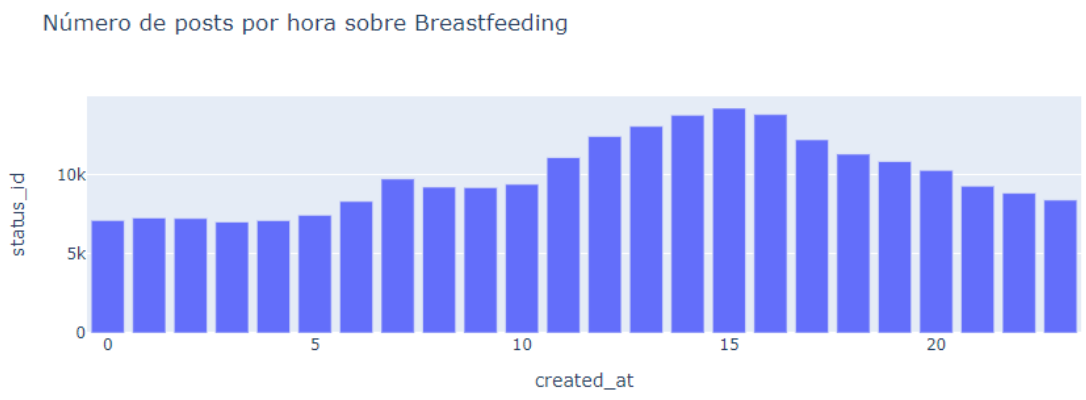
➔ En base a la agrupación, se cuentan los elementos utilizando **status_id**.

```
[88]: dt=dataset.groupby([dataset['created_at'].dt.hour])['status_id'].count()
      print(dt)
```

```
created_at
0      7102
1      7259
2      7226
3      7000
4      7095
5      7431
6      8300
7      9714
8      9202
9      9171
10     9375
11    11064
12    12398
13    13030
14    13734
15    14172
16    13775
17    12183
18    11280
19    10814
20    10256
21     9263
22     8817
23     8384
Name: status_id, dtype: int64
```

Con este bloque se genera la gráfica de barras que permite ver las horas del día con mayor número de publicaciones, utilizando **status_id** como los números a graficar y **created_at** como las 24 horas del día.

```
[90]: figura = plotly.bar(dt, y='status_id', title="Número de posts por hora sobre Breastfeeding")
      figura.show()
```



Análisis de los datasets referentes a Formula Milk

Para realizar el análisis de los datasets sobre *Formula Milk*, se utilizó otro archivo para mantener separados y organizados los bloques para el proyecto. Debido a ello, se procederá a explicar nuevamente los pasos realizados para obtener las gráficas y análisis de la información.

Se realizó la importación de las librerías **numpy, pandas, plotly** y **malplotlib**, que permitirán manipular datos y obtener información en base a ellos.

Se debe de realizar un recorrido para obtener los documentos que se encuentran en la carpeta donde están almacenados los datasets, con el fin de ver que éstos sean detectados de forma correcta por Python.

```
[5]: import numpy as numpy #librería de algebra lineal
     import pandas as panda #librería para el procesamiento de datos
     #librería para crear gráficas
     import plotly.express as plotly #sofisticadas
     import matplotlib.pyplot as matplot #sencillas
     import re #librería para hacer uso de expresiones regulares

     import os #se importan los datasets con los que se trabajaran
     for dirname, _, filenames in os.walk('C:/Users/Valeria/Documents/9no Semestre/Analítica de Big Data/Proyecto Final/Breastfeeding/'):
         for filename in filenames:
             print(os.path.join(dirname, filename))

C:/Users/Valeria/Documents/9no Semestre/Analítica de Big Data/Proyecto Final/Breastfeeding/Breastfeeding_3months.xlsx
C:/Users/Valeria/Documents/9no Semestre/Analítica de Big Data/Proyecto Final/Breastfeeding/Breastfeeding_3months_.csv
C:/Users/Valeria/Documents/9no Semestre/Analítica de Big Data/Proyecto Final/Breastfeeding/FormulaMilk_3months.xlsx
C:/Users/Valeria/Documents/9no Semestre/Analítica de Big Data/Proyecto Final/Breastfeeding/FormulaMilk_3months_.csv
```

Al verificar la existencia de los archivos, se seleccionará el dataset con el cual se requiera trabajar, por lo tanto, se eligió *FormulaMilk_3months_.csv*

En el siguiente apartado se realiza la lectura del archivo csv, y para comprobar la información se utiliza el método Head() para mostrar una parte de esta (por defecto devuelve los primeros 5 registros):

```
[7]: #Se realiza la lectura del archivo que se analizará
     dataset = panda.read_csv('C:/Users/Valeria/Documents/9no Semestre/Analítica de Big Data/Proyecto Final/Breastfeeding/FormulaMilk_3months_.csv')

[8]: dataset.head() #Se muestra una parte de la información
```

[8]:

	status_id	created_at	text	display_text_width	country	day
0	1.538674e+18	2022-06-20 00:02:41 UTC	I have 67 days worth of milk stocked up for my...	267.0	NaN	Monday
1	1.538675e+18	2022-06-20 00:07:19 UTC	Free Day & Night Toddler Milk Samples - HA...	267.0	NaN	Monday
2	1.538676e+18	2022-06-20 00:12:58 UTC	I need help with formula milk & diapers. W...	144.0	NaN	Monday
3	1.538677e+18	2022-06-20 00:14:12 UTC	Bill Gates is amazing.\nHe launches his brand ...	139.0	NaN	Monday
4	1.538677e+18	2022-06-20 00:16:21 UTC	🚨 IN STOCK ALERT 🚨\n\nSimilac 22.8 oz. Milk-...	208.0	NaN	Monday

Posteriormente, como parte de la obtención de los datos, se comprueban los tipos de datos que contiene cada campo:

```
[9]: dataset.info() #Devuelve la información de las columnas

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43559 entries, 0 to 43558
Data columns (total 6 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   status_id           41205 non-null  float64
 1   created_at          41205 non-null  object
 2   text                38195 non-null  object
 3   display_text_width  41205 non-null  float64
 4   country             71 non-null     object
 5   day                 41205 non-null  object
dtypes: float64(2), object(4)
memory usage: 2.0+ MB
```

También se buscan la cantidad de registros que tengan alguna columna en blanco / nulos:

```
[11]: dataset.isna().sum() #Busca las columnas que tengan valores nulos

[11]: status_id             2354
      created_at            2354
      text                  5364
      display_text_width    2354
      country              43488
      day                   2354
      dtype: int64
```

- ➔ Para la columna *created_at*, se encontró que tiene un tipo de dato "object", por lo tanto hay que volver a comprobar con el método **type().**
- ➔ Al devolver string, se concluye que tiene que convertirse el tipo de dato de dicha columna, ya que sabemos que es una fecha y hora, para dicha tarea se utiliza **to_datetime()**:

```
[12]: #Devuelve el tipo de objeto en la columna 'created_at'
      type(dataset.at[0, 'created_at'])
      #El valor que devuelve es str

[12]: str

[13]: dataset['created_at'] = panda.to_datetime(dataset['created_at']) #Cambia de string datetime a datetime

[14]: dataset['created_at'].head() #Muestra las primeras fechas en formato string datetime

[14]: 0   2022-06-20 00:02:41+00:00
      1   2022-06-20 00:07:19+00:00
      2   2022-06-20 00:12:58+00:00
      3   2022-06-20 00:14:12+00:00
      4   2022-06-20 00:16:21+00:00
      Name: created_at, dtype: datetime64[ns, UTC]
```

Ahora a manera informativa, se obtienen todos los países que se encuentren en el dataset (sin repetir, mediante el método **unique()** a la columna ***country***):

```
[15]: dataset['country'].unique()

[15]: array([nan, 'United States', 'Australia', 'United Kingdom', 'Canada',
             'Republic of the Philippines', 'Botswana', 'Zimbabwe', 'Ghana',
             'Mexico', 'Pakistan', 'South Africa', 'Nigeria', 'Ireland'],
            dtype=object)
```

Para el siguiente bloque se utiliza la librería **PyCountry**, una biblioteca de Python para completar los datos de los países a partir de la base de datos de países para preparar el conjunto de datos para su comparación o agregación con otros.

- ➔ La variable ***mapeo*** se llena con el código en tres siglas de cada país
- ➔ La variable ***mapeo_banderas*** se llena con la bandera de cada país
- ➔ Al dataset se agregan dos columnas: **country_code** y **country_flag** y se llenan con el contenido de **mapeo** (códigos en tres siglas) y **mapeo_banderas** (imágenes de banderas) respectivamente.

```
[20]: import pycountry
      mapeo = {country.name: country.alpha_3 for country in pycountry.countries} #alpha_3 cambia el nombre del país por sus 3 siglas
      mapeo_banderas = {country.name: country.flag for country in pycountry.countries} #Obtiene las banderas de los países

[21]: #Se renombran los nombres de los países a los cuales no se pueden obtener su código
      dataset['country'].replace(['Republic of the Philippines'],
      ['Philippines'],regex = True, inplace = True)
      #regex permite expresiones regulates e inplace permite aplicar los cambios directamente al dataset

[22]: dataset = dataset.assign(country_code = None) #Se agrega la columna country_code
      dataset = dataset.assign(country_flag = None) #Se agrega la columna country_flag
      for index in dataset.index:   #Realiza recorrido por filas del dataset
          country = dataset.at[index, 'country']  #country es igual al elemento en index de la columna country
          dataset.at[index, 'country_code'] = mapeo.get(country) #Se coloca en el dataset las siglas del país
          dataset.at[index, 'country_flag'] = mapeo_banderas.get(country) #Se coloca en el dataset las banderas
```

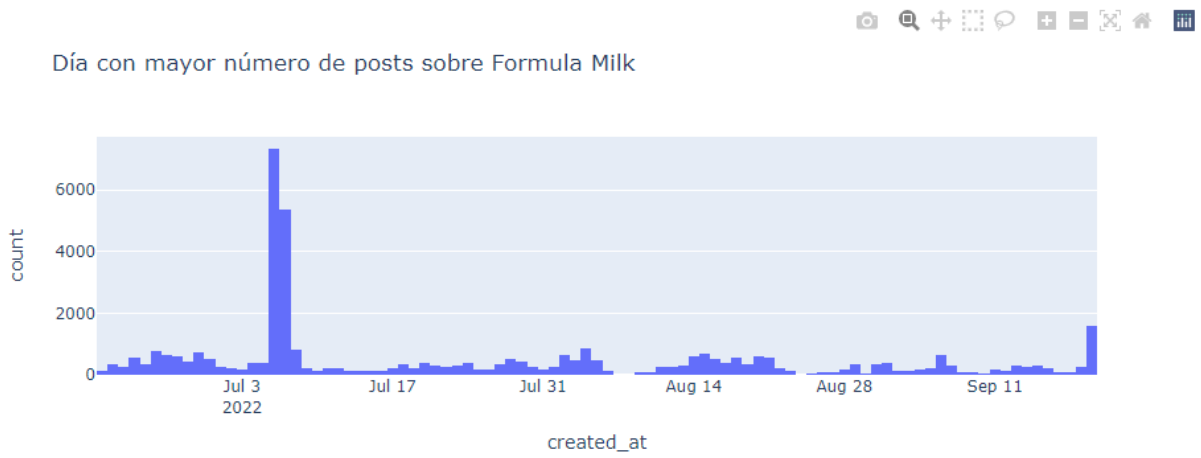
- ➔ Se comprueba dicha información consultando las dos columnas con **unique()**

```
[23]: dataset['country_code'].unique() #Obtiene de la columna country_code los elementos no repetidos, siendo 45 países

[23]: array([None, 'USA', 'AUS', 'GBR', 'CAN', 'PHL', 'BWA', 'ZWE', 'GHA',
             'MEX', 'PAK', 'ZAF', 'NGA', 'IRL'], dtype=object)

[24]: dataset['country_flag'].unique() #Obtiene de la columna country_code los elementos no repetidos, siendo 45 países

[24]: array([None, 'US', 'AU', 'GB', 'CA', 'PH', 'BW', 'ZW', 'GH', 'MX', 'PK',
             'ZA', 'NG', 'IE'], dtype=object)
```

Con el siguiente bloque se crea un histograma, que permite ver el día con mayor número de posts generados en relación al tema *Formula Milk*. Para ello se utiliza **Plotly** con el método .**histogram**

donde se usa la columna **created_at** para obtener los días donde se realizaron publicaciones a lo largo de 3 meses.

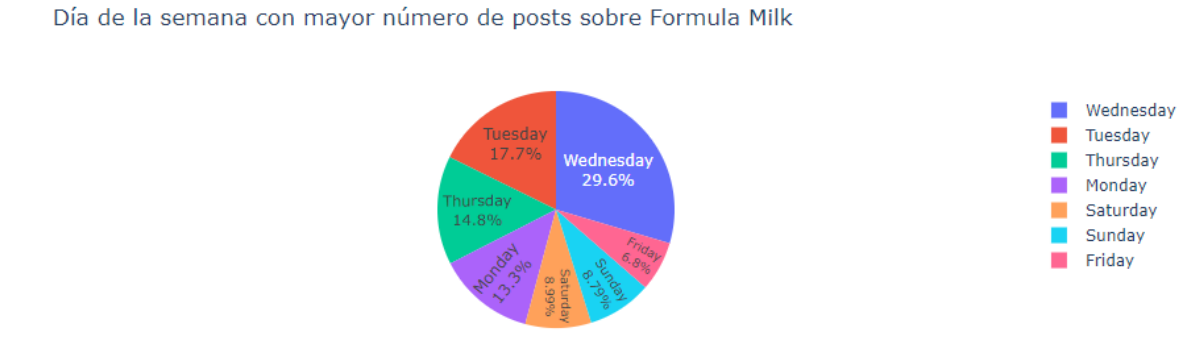
```
[40]: figura = plotly.histogram(dataset, x = 'created_at', title='Día con mayor número de posts sobre Formula Milk')
      figura.show()
```



Con la siguiente gráfica se hace un análisis sobre el día de la semana con mayor número de posts sobre *Formula Milk*, para ello:

- ➔ En la primera línea se genera una agrupación utilizando **day** y su tamaño.
- ➔ En la segunda se ordenaron de acuerdo al día que mayor número de posts generados
- ➔ Por último, se genera la gráfica utilizando **Plotly** y su método de **pie**, colocando los valores que se graficarán.

```
[41]: pie = dataset.groupby(['day']).size().reset_index(name = 'size')
      pie.sort_values(by='size', ascending=False ,inplace=True)
      figura = plotly.pie(pie, values='size', names='day',
                  title='Día de la semana con mayor número de posts sobre Formula Milk',
                  hover_data=['size'])
      figura.update_traces(textposition='inside', textinfo='percent+label')
      figura.show()
```



En el siguiente bloque de código se genera un conjunto de datos de los países con mayor número de publicaciones realizadas. Para ello:

- ➔ Se crea una agrupación llamada **dtMundial**, donde se agrupa utilizando **country**, **country_code** y **country_flag**, obteniendo su tamaño.
- ➔ Se organizan los valores de **dtMundial** por la columna **size**.
- ➔ Por último, se muestra una previsualización de los valores de la agrupación.

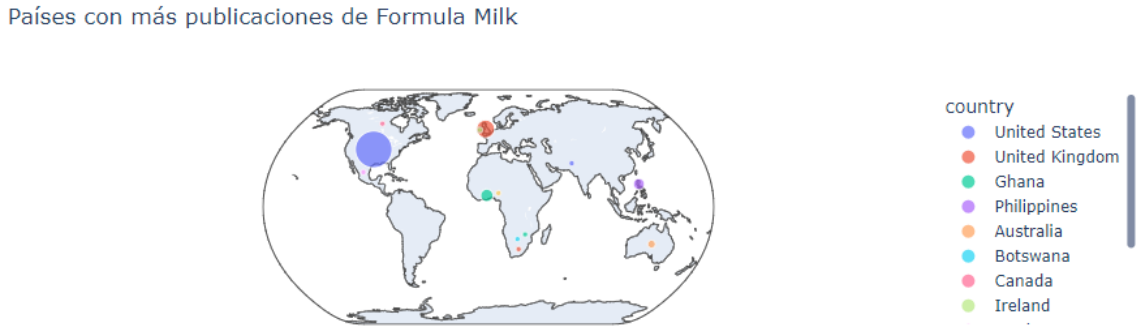
```
[34]: dtMundial = dataset.groupby(['country', 'country_code', 'country_flag']).size().reset_index(name = 'size')
      dtMundial.sort_values(by='size', ascending=False ,inplace=True)

[35]: dtMundial.head()
```

[35]:

	country	country_code	country_flag	size
11	United States	USA	US	42
10	United Kingdom	GBR	GB	10
3	Ghana	GHA	GH	5
8	Philippines	PHL	PH	4
0	Australia	AUS	AU	2

Al generar la agrupación, se crea la gráfica de un mapa mundial que mostrará la ubicación geográfica de los países donde se hacen más publicaciones sobre *Formula Milk*, utilizando **size** para poder graficar sobre el mapa.

```
[37]: figura = plotly.scatter_geo(dtMundial, locations="country_code", color="country",
                                  hover_name="country", size="size",
                                  projection="natural earth1", title="Países con más publicaciones de Formula Milk")
      figura.show()
```



Para el siguiente bloque de código, se crea una agrupación de la cantidad de publicaciones que se realizan en cada hora de un día, tomando en cuenta:

- ➔ De la columna **created_at** se agrupa utilizando la hora.

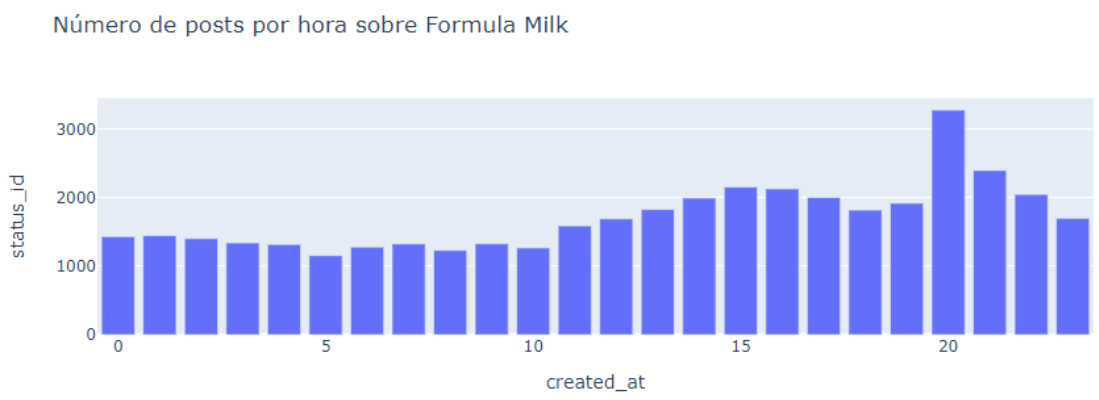
➔ En base a la agrupación, se cuentan los elementos utilizando **status_id**.

```
[25]: dt=dataset.groupby([dataset['created_at'].dt.hour])['status_id'].count()
      print(dt)
```

```
created_at
0.0     1437
1.0     1452
2.0     1407
3.0     1345
4.0     1322
5.0     1158
6.0     1284
7.0     1330
8.0     1235
9.0     1332
10.0    1272
11.0    1594
12.0    1697
13.0    1833
14.0    2001
15.0    2161
16.0    2135
17.0    2009
18.0    1825
19.0    1925
20.0    3288
21.0    2404
22.0    2054
23.0    1705
Name: status_id, dtype: int64
```

Con este bloque se genera la gráfica de barras que permite ver las horas del día con mayor número de publicaciones, utilizando **status_id** como los números a graficar y **created_at** como las 24 horas del día.

```
[39]: figura = plotly.bar(dt, y='status_id', title="Número de posts por hora sobre Formula Milk")
      figura.show()
```



COMPARATIVA DE RESULTADOS

Para tener una mejor visión de los resultados obtenidos, se realizó una tabla comparativa de ambos temas con respecto a las gráficas realizadas.

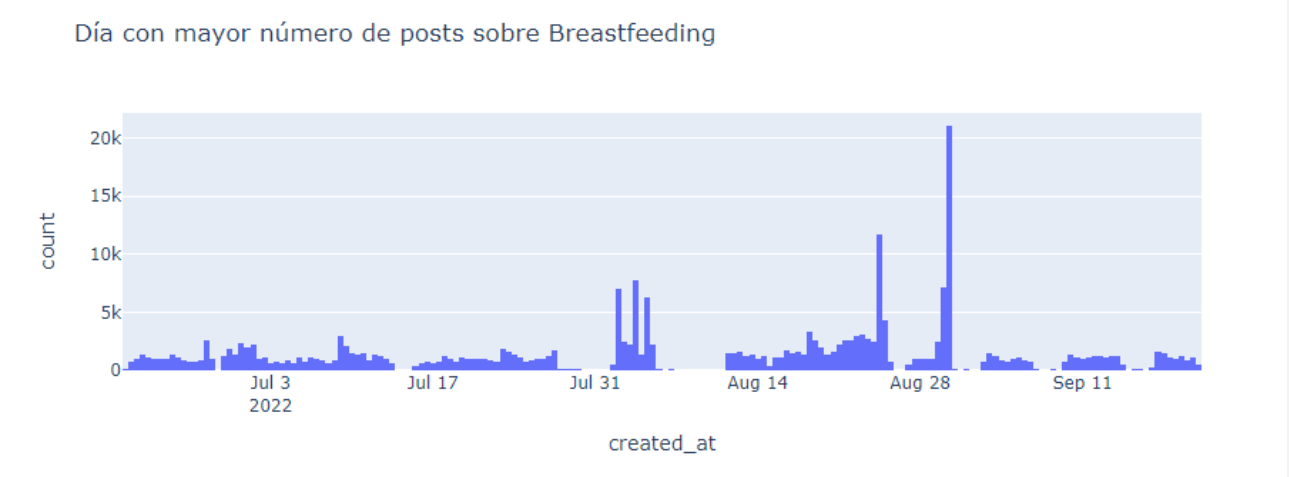
Gráfica	Breastfeeding	Formula Milk
Fecha con mayor número de posts	30-Agosto con 21,062 publicaciones	05-Julio con 7,343 publicaciones
Día de la semana con mayor número de posts	Martes con 23.4% con 55,812 publicaciones	Miércoles con 29.6% con 12,195 publicaciones
Países con mayor número de Posts	Estados Unidos 🇺🇸	Estados Unidos 🇺🇸
Número de posts por hora	3:00 pm con 14,172	8:00 pm horas con 3,288

ANÁLISIS DE RESULTADOS OBTENIDOS POR GRÁFICA

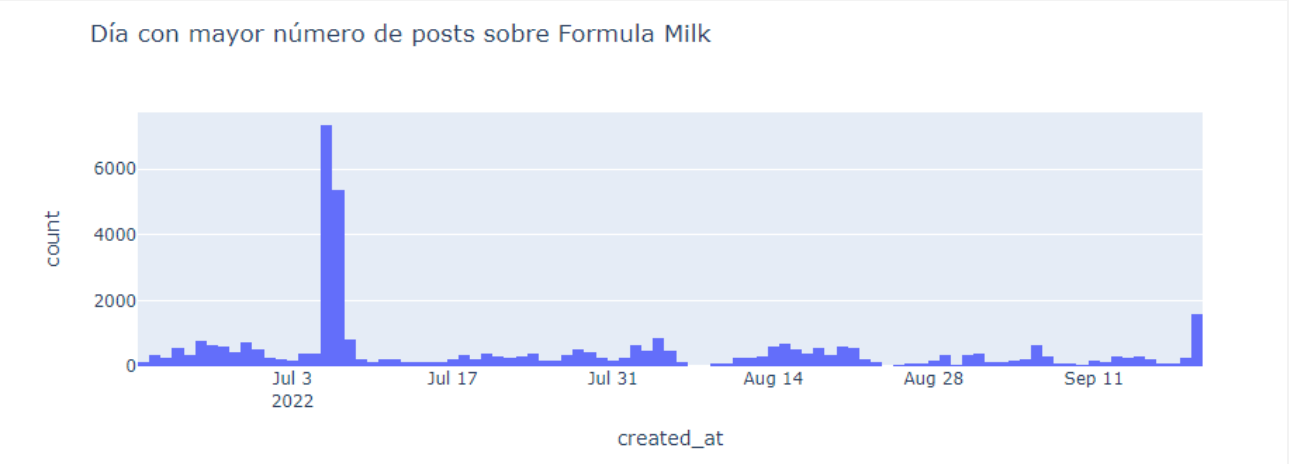
Fecha con mayor número de posts realizados

El histograma realizado representa todos los días que se realizaron publicaciones dentro de un periodo de tres meses, en este caso, los meses abarcan desde el 20 de junio hasta el 20 de septiembre, donde el eje x es la fecha de creación y el eje y la cantidad de posts realizados.

- Para *Breastfeeding*, el día con mayor cantidad de publicaciones realizadas es el día 30 de agosto.



- Para *Formula Milk*, el día con mayor cantidad de publicaciones realizadas es el día 05 de julio.



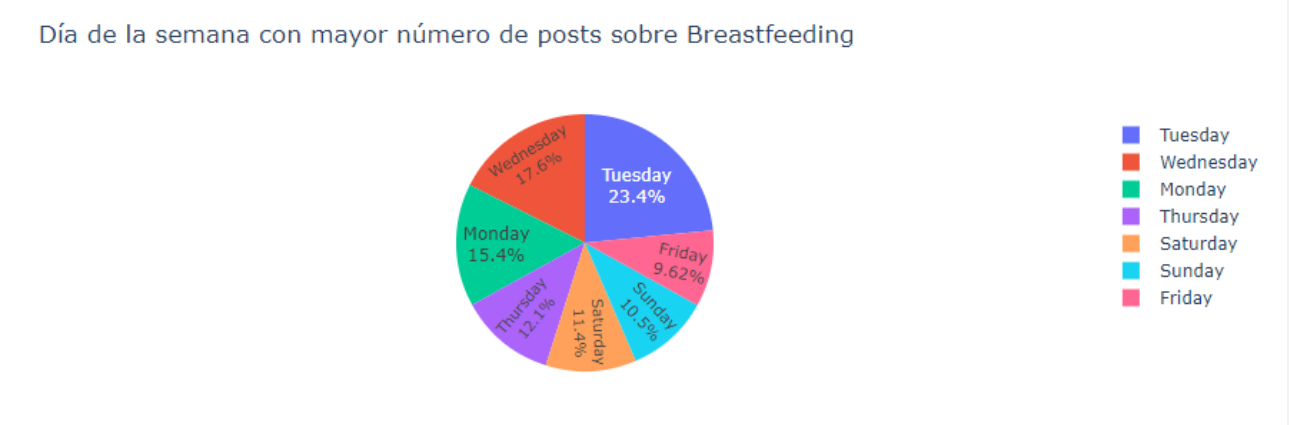
Al realizar el análisis de ambos histogramas, se llegó a las siguientes conclusiones:

- El tema de *Breastfeding* tuvo mayor actividad debido a que la primera semana de agosto es la semana para la lactancia materna. En general, mantuvo un nivel consistente de publicaciones.
- El tema de *Formula Milk* no tuvo tanta actividad, a excepción de un par de días de julio.

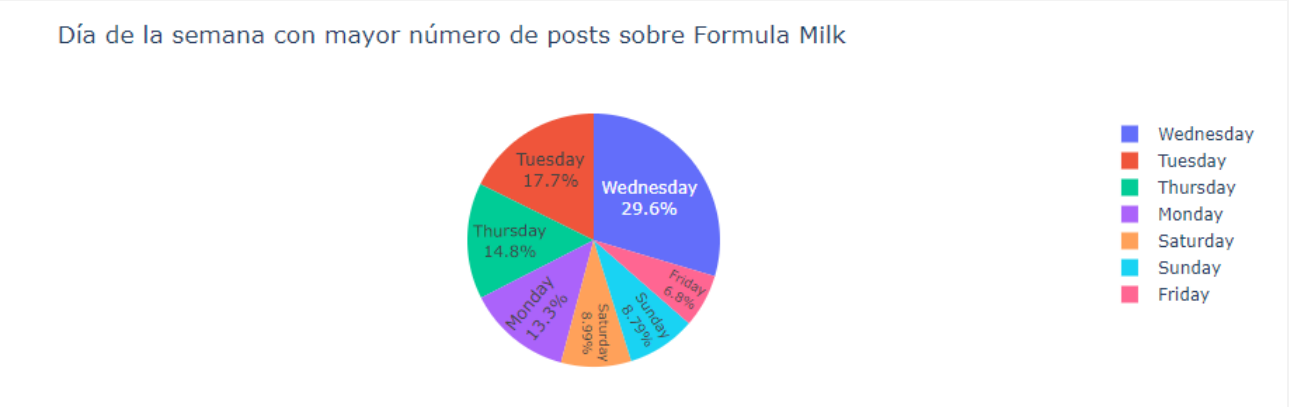
Día de la semana con mayor número de posts realizados

La gráfica de pie está formada por 7 partes, donde cada una de ellas representa cada uno de los días de la semana, con su porcentaje de la cantidad de posts realizados.

- Para *Breastfeeding*, el día de la semana con mayor cantidad de publicaciones realizadas es el día Martes.



- Para *Formula Milk*, el día de la semana con mayor cantidad de publicaciones realizadas es el día Miércoles.



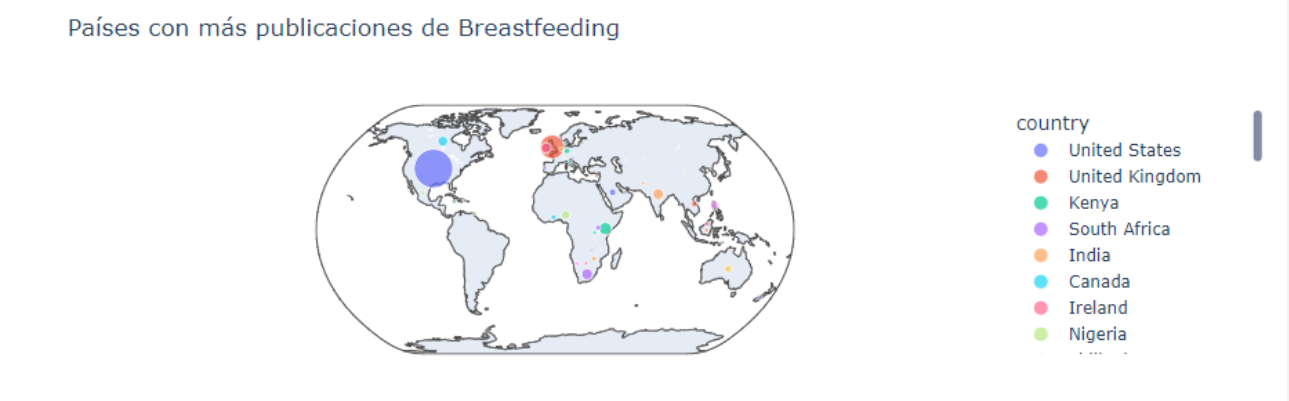
Al realizar el análisis de ambas gráficas de pie, se llegó a las siguientes conclusiones:

- En ambos tópicos, los días con menos actividad son los viernes, sábados y domingos.
- Ambos días de la semana que tienen más actividad, representan un poco más de ⅛ de la gráfica.

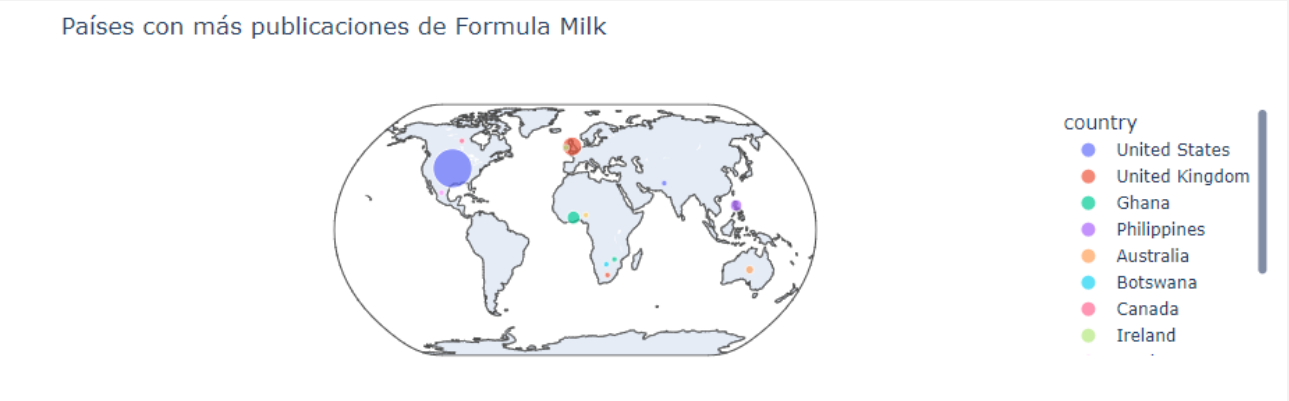
Países con mayor cantidad cantidad de publicaciones realizadas

La gráfica muestra un mapa mundial que ubica los países que realizaron la mayor cantidad de posts. Además, la cantidad de posts está representada por un círculo, mientras más grande sea, es mayor el número de publicaciones realizadas.

- Para *Breastfeeding*, el país con mayor cantidad de publicaciones realizadas es Estados Unidos.



- Para *Formula Milk*, el país con mayor cantidad de publicaciones realizadas es Estados Unidos.



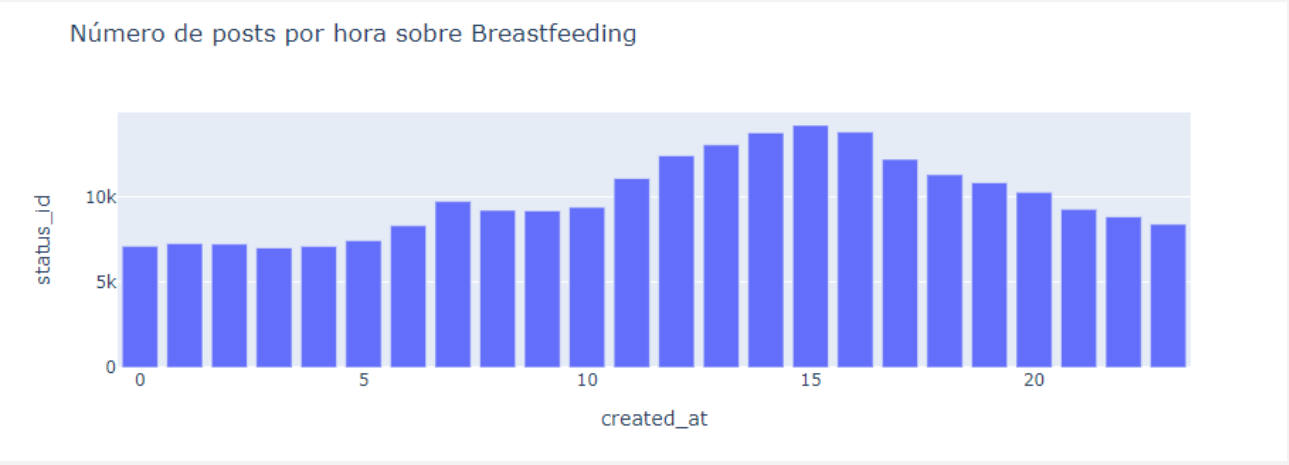
Al realizar el análisis de ambas gráficas, se llegó a las siguientes conclusiones:

- Los primeros dos países con mayor actividad para ambos temas son Estados Unidos y Reino Unido.

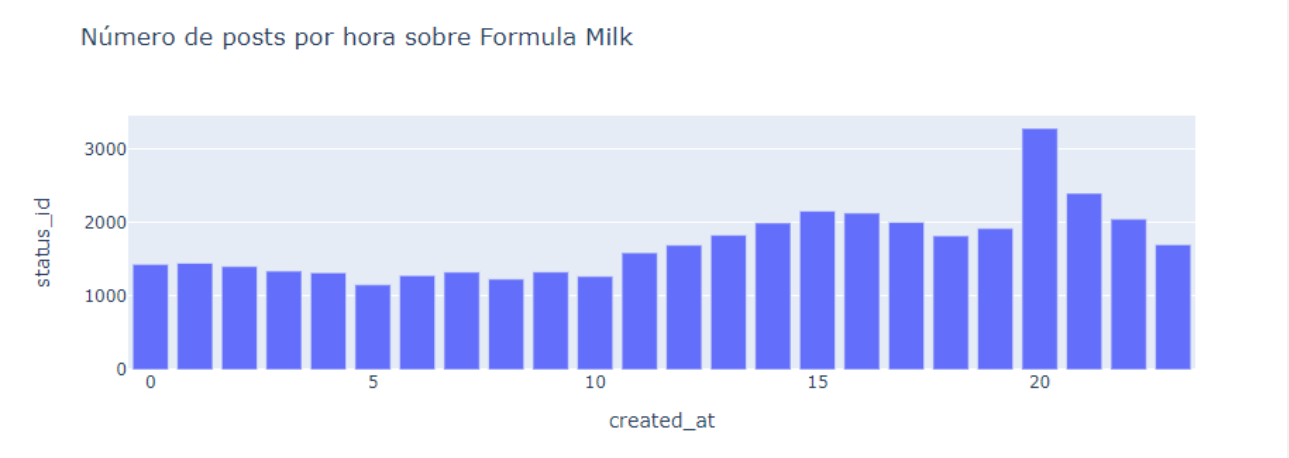
Horas con mayor cantidad de publicaciones realizadas

La gráfica de barras representa está conformada por las 24 horas de un día, donde el eje x son las horas y el eje y es la cantidad de publicaciones que se realizaron.

- Para *Breastfeeding*, la hora con mayor cantidad de publicaciones realizadas es las 03:00 pm.



- Para *Formula Milk*, la hora con mayor cantidad de publicaciones realizadas es las 08:00 pm.



Al realizar el análisis de ambas gráficas, se llegó a las siguientes conclusiones:

- Para el tema de *Breatsfeeding*, las horas con mayor actividad son entre las 02:00 pm y 04:00 pm.
- Para el tema de *Formula Milk*, solamente se tiene una hora con mayor actividad, siendo las 08:00 pm.
- *Breastfeeding* mantiene un nivel de actividad un poco más constante que *Formula Milk.*

Como observaciones finales, el tema de *Breastfeeding* tiene mayor actividad dentro de las redes sociales a comparación de *Formula Milk* ya que es un tema del cual no se suele hablar mucho con las personas cercanas, provocando que inicien una conversación sobre el mismo en la cual generan preguntas y comparten experiencias en sus procesos.

En general, ambos métodos son efectivos para la alimentación y nutrición de los bebés, pero sin duda alguna, debido al aumento de popularidad del tema de *Breastfeeding* es que se genera mayor actividad de la misma.

CONCLUSIONES

La elaboración de este proyecto ha significado un salto importante en la visión que tenemos tanto como usuarios y desarrolladores de software, las formas en que se puede extraer información acerca del comportamiento, tendencias o patrones a partir de aplicaciones que utilizamos en la vida cotidiana, representa una innovación en cuanto al estudio de los comportamientos y conductas.

Dentro del marco del desarrollo del presente proyecto, se aplicaron distintas áreas del conocimiento, como lo son la analítica de big data, la programación, la investigación cuantitativa, entre otras, que permitieron llegar a un conjunto de conclusiones que sirvieron como una respuesta a la problemática inicial.

REFERENCIAS

Alberca, A. S. (2020, October 4). La librería Matplotlib. Aprende Con Alf. La librería Matplotlib | Aprende con Alf

Alberca, A. S. (2022, June 14). La librería Pandas. Aprende Con Alf. La librería Pandas | Aprende con Alf

Data Cleansing. (n.d.). Data Cleansing: ¿cómo hacer la limpieza de datos?

Estévez, R. (2019, August 9). Visualizando datos espaciales con Plotly y Jupyter Notebook. Geomapik. Mapas con Plotly: visualizando datos espaciales - geomapik

Gonzalez, L. (2022, September 23). Librería NumPy. Aprende IA. Librería NumPy - Aprende IA

Rédac, T. (2022, August 1). Hacer Data Visualisation gracias a Plotly. Formation Data Science | DataScientest.com. Hacer Data Visualisation gracias a Plotly

Santander Universidades. (2022, septiembre 16). ¿Qué es Python? | Blog. Becas Santander. Python: qué es y por qué deberías aprender a utilizarlo

School, T. (2022, July 29). Ciencia de datos: Anaconda Python. Tokio School. Ciencia de datos: distribución con Anaconda Python | Tokio School