Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
EN CIENCIA E ENXEÑARÍA DE DATOS

Deseño e implementación dunha plataforma web para a análise e visualización de chíos sobre a COVID-19 en EEUU

Estudante: Patricia Bardanca Rojo

Dirección: Tirso Varela Rodeiro

Diego Seco Naveiras

A Coruña, September de 2023.

A Lola

Agradecementos

Gustaríame agradecer en primeiro lugar aos meus titores, Diego e Tirso, por guiarme en todo momento e pola súa implicación no traballo. A súa orientación e sabiduría foron inestimables ao longo deste proceso, sinto unha profunda gratitude pola vosa paciencia, dedicación e inspiración.

En segundo lugar, agradecerlle á miña familia, pilar fundamental na miña vida, non teño palabras suficientes para describir canto aprecio o voso apoio incondicional. A papá, a mamá e a miña irmá Lucía, por estar conmigo desde o inicio, estivestes ao meu carón, brindándome amor, alento e comprensión. Cada logro que alcanzo é tamén un logro voso, e este traballo de fin de grao non sería posible sen o voso constante alento e confianza en min. E, á miña amiga Lucía, que sepas que cho agradezo da misma maneira, pois para min xa eres familia.

Quería dar grazas tamén aos compañeiros que me levo desta etapa, por facer deste grao moito máis divertido e levadeiro. Lembrarei sempre as risas nas clases, as axudas cos traballos e cos exames nas mesas da fic e as aventuras fóra do campus. Levarei conmigo as lembranzas e os vínculos que formamos. Este grao non sería o mesmo sen vós, e estou agradecida por cada momento que compartimos. En especial, quero darlle as grazas a Pedro, por confiar sempre en min e apoiarme como o que máis, contigo todo é moito máis fácil.

E, en último lugar, grazas a ti, Lola. Quedoume pendiente ensiñarche a escribir e a leer pero espero, que de algunha maneira, che chegue este traballo e estés moi orgullosa de min por finalizar esta etapa. Lévote conmigo sempre.

Resumo

O presente traballo de fin de grao centrouse no deseño e desenvolvemento dunha plataforma web dedicada á análise e visualización de mensaxes en Twitter (chíos) relacionadas coa COVID-19 nos Estados Unidos, basado na necesidade de comprender o impacto e a percepción pública da pandemia a través da recollida e análise de datos en redes sociais.

O proceso estruturouse en varios pasos clave. Inicialmente, definíronse os requisitos funcionais e estableciuse o alcance do proxecto, para continuar coa obtención dun conxunto de datos adecuado ao problema. Seguidamente, procedeuse coa realización dunha análise preliminar dos datos dispoñibles, seguida dunha análise máis profunda que se centrou particularmente na detección e análise de sentimentos expresados nos chíos.

Posteriormente, ideáronse bocetos para posibles interfaces de usuario que a plataforma podería adoptar, o que permitiu visualizar como os usuarios interactuarían coa aplicación, e desenvolveuse un modelado conceptual que serviu como base para a implementación. Finalmente, desenvolveuse e implementouse a plataforma.

En canto ás tecnoloxías empregadas, elixíronse as seguintes: PostgreSQL, coa extensión PostGIS, empregouse como base de datos para almacenar o conxunto de datos recollidos. A extensión PostGIS permitiu almacenar, xestionar e analizar información xeo-referenciada de xeito sinxelo e integrado coa base de datos principal. O framework Flask, xunto co ORM SQLAlchemy (con extensión GeoAlchemy) foi a elección para definir os modelos de datos e crear a aplicación web. Para mellorar a interacción nas vistas finais, empregáronse tecnoloxías como Leaflet, HTML, CSS e JavaScript. Ademais, para analizar os sentimentos nos chíos, utilizáronse ferramentas de procesamento de linguaxe natural e análise de sentimento. O proxecto tamén fixo uso de PowerBI para a visualización efectiva do contido xerado.

O traballo abarcou diversos aspectos esenciais para proporcionar unha ferramenta eficiente e significativa na comprensión da percepción pública durante a pandemia. Todo isto xestionouse seguindo unha metodoloxía iterativa e incremental, permitindo unha adaptación flexible ás necesidades en constante evolución e asegurando a calidade do resultado final ao longo do tempo.

Abstract

The present final degree work focused on the design and development of a web platform dedicated to the analysis and visualization of Twitter messages (tweets) related to COVID-19 in the United States, based on the need to understand the impact and public perception of the pandemic through the collection and analysis of social media data.

The process was structured in several key steps. Initially, the functional requirements were defined and the scope of the project was established, followed by obtaining a data set appropriate to the problem. Next, a preliminary analysis of the available data was performed, followed by a deeper analysis that focused particularly on the detection and analysis of sentiments expressed in the tweets.

Subsequently, sketches were devised for possible user interfaces that the platform could adopt, which allowed visualizing how users would interact with the application, and a conceptual modeling was developed to serve as the basis for the implementation. Finally, the platform was developed and implemented.

In terms of the technologies used, were chosen: PostgreSQL, with the PostGIS extension, was used as the database to store the collected dataset. The PostGIS extension allowed storing, managing and analyzing geo-referenced information in a simple and integrated way with the main database. The Flask framework, together with the SQLAlchemy ORM (with GeoAlchemy extension), was chosen to define the data models and create the web application. To improve interaction in the final views, technologies such as Leaflet, HTML, CSS and JavaScript were used. In addition, to analyze the sentiments in the tweets, natural language processing and sentiment analysis techniques were implemented. The project also made use of PowerBI for effective visualization of the generated content.

The work covered various aspects essential to provide an efficient and meaningful tool in understanding public perception during the pandemic. All of this was managed following an iterative and incremental methodology, allowing for flexible adaptation to evolving needs and ensuring the quality of the end result over time.

Palabras chave:

- Análise e visualización de datos
- Análise de sentimentos
- Ciencia e enxeñaría de datos
- Covid-19
- Aplicación web
- Xeoinformática
- PostgreSQL
- Flask
- Pyhton
- Leaflet
- PowerBI

Keywords:

- Data analysis and visualization
- Sentiment analysis
- Data science and engineering
- Covid-19
- Web application
- Geoinformatics
- PostgreSQL
- Flask
- Pyhton
- Leaflet
- PowerBI

Índice Xeral

Índice de Figuras

Índice de Táboas

Capítulo 1

Introdución

Neste capítulo, preséntase tanto a motivación que xace baixo a realización deste proxecto como os obxectivos que o mesmo debe cumprir.

1.1 Motivación

A epidemia da COVID-19, que se orixinou en Wuhan, China, en decembro de 2019, tivo un impacto global sen precedentes, afectando ao mundo enteiro a nivel económico, sociolóxico e sanitario [1]. A rápida propagación do virus xerou unha gran cantidade de información e opinións, expresadas principalmente a través das redes sociais, converténdose estas nunha fonte fundamental para obter novas sobre a evolución da situación da epidemia en diferentes rexións.

No medio da incerteza e da necesidade de manterse informado, as redes sociais, como Twitter, agora coñecido como X [2], permitíronlle á xente compartir novas, actualizacións e opinións sobre a COVID-19. Desde o inicio do brote en Wuhan ata a súa expansión global, as redes sociais enchéronse de mensaxes, publicacións e debates sobre a evolución da epidemia, medidas de prevención, recursos sanitarios, políticas gobernamentais e outros temas relacionados. A velocidade e a accesibilidade das redes sociais fixeron posible que calquera persoa con conexión a internet participara na conversa en torno á COVID-19, xerando unha gran cantidade de información e opinións de diversa índole.

Ademais, as redes sociais tamén foron utilizadas para expresar preocupacións e críticas sobre a xestión da pandemia por parte dos gobernos e sistemas de saúde. Xeráronse debates sobre estratexias de contención, medidas de mitigación, distribución de recursos e equidade no acceso á atención médica, entre outras cuestións, que influíron nas decisións tomadas e na percepción pública da situación.

O obxectivo principal do proxecto é realizar unha análise xeoespacial de chíos presentes nun conxunto de datos que contén información sobre a COVID-19 nos Estados Unidos. Un

chío é cada unha das mensaxes que publican os usuarios en Twitter. A análise busca estudar a evolución do interese e da percepción da poboación estadounidense ao longo do tempo, así como identificar as áreas xeográficas onde a epidemia suscitou maior preocupación, no referente ás respostas á enfermidade expresadas en Twitter. Tamén busca examinar as diferenzas significativas na percepción da crise entre diferentes estados ou rexións, así como as relacións existentes entre os temas discutidos polas persoas en Twitter e o seu impacto no sentimento público. O obxectivo é determinar se certos temas ou etiquetas poden contribuír a xerar unha impresión positiva ou negativa da situación e se ocorren cambios na opinión pública.

Con base en todo isto, desenvolveuse unha aplicación web, coa idea de dispoñer da información condensada para que os usuarios poidan analizar en retrospectiva tanto a evolución da epidemia como a evolución do estado de ánimo das persoas, proporcionando mapas e outros tipos de visualizacións para que poidan realizar consultas con respecto á mesma e navegar a través de diferentes filtros sobre a información presente, así como realizar análises xeográficas.

1.2 Obxectivos

Como obxectivo principal deste proxecto, estableceuse a creación dunha plataforma web que permitirá a análise e visualización dos chíos de Twitter. Esta plataforma enfocarase en comprender o impacto e a percepción pública da pandemia a través da recollida e análise de mensaxes en redes sociais.

Ademais, este obxectivo central componse de obxectivos secundarios que se presentan a continuación:

- **Realizar unha análise exploratoria completa dos datos recollidos a partir das mensaxes en redes sociais.** Nesta etapa, os datos en cru serán preparados, preprocesados e visualizados antes de seren examinados en busca de patróns e tendencias útiles. Este proceso permitirá unha comprensión máis profunda do comportamento dos usuarios a través da exploración dos datos.

- **Realizar unha análise de sentimentos nas mensaxes recollidas.** Empregar técnicas de procesamento de linguaxe natural e análise de sentimentos para identificar e comprender as emocións e opinións expresadas nas mensaxes relacionadas coa COVID-19. Isto permitirá capturar a percepción pública e as reaccións emocionais ante a pandemia.

- **Modelar os datos.** Utilizar técnicas de modelado de datos para transformar os datos recollidos en información detallada sobre a evolución da COVID-19 nos Estados Unidos. Para abordar esta transformación, é imprescindible organizar os datos de xeito

que se teñan en conta os aspectos máis relevantes dos comentarios en Twitter sobre a propagación da enfermidade.

- **Desenvolver unha plataforma web que permita aos usuarios realizar consultas sobre os datos.** É de suma importancia despregar unha interface que sexa áxil, práctica, intuitiva e, sobre todo, sinxela. En busca deste obxectivo, a aplicación debe contar cunha páxina na que se reflicta, de forma ordenada e detallada, información sobre a evolución da pandemia ao longo do tempo e xeograficamente, ademais de permitir aos usuarios explorar gráficos e estatísticas que amosen as emocións predominantes nas mensaxes analizadas ao longo do tempo e en diferentes rexións. Ademais, esta información poderá ser utilizada xunto cos datos xeo-referenciados para obter unha comprensión máis completa da percepción pública e das tendencias emocionais relacionadas coa COVID-19 en diferentes áreas dos Estados Unidos.

Capítulo 2

Fundamentos tecnolóxicos

Nesta sección, realizarase unha revisión sobre a literatura relacionada co tema do proxecto, que se pode ver no Apartado 2.1, co obxectivo de ofrecer un panorama actualizado das investigacións, estudos e desenvolvementos anteriores que trataron aspectos similares ao do traballo presente. A través desta revisión, establécese o contexto no cal se inscribe o proxecto e identifícanse as tendencias, enfoques, tecnoloxías e resultados máis relevantes neste campo, así como áreas de oportunidade e brechas que o proxecto busca abordar. Por outro lado, comentaranse as tecnoloxías utilizadas para o desenvolvemento total do proxecto, dende tecnoloxías de procesamento e análise de datos, ata ferramentas de visualización e de integración, que serán enumeradas xunto cunha breve explicación no Apartado 2.2.

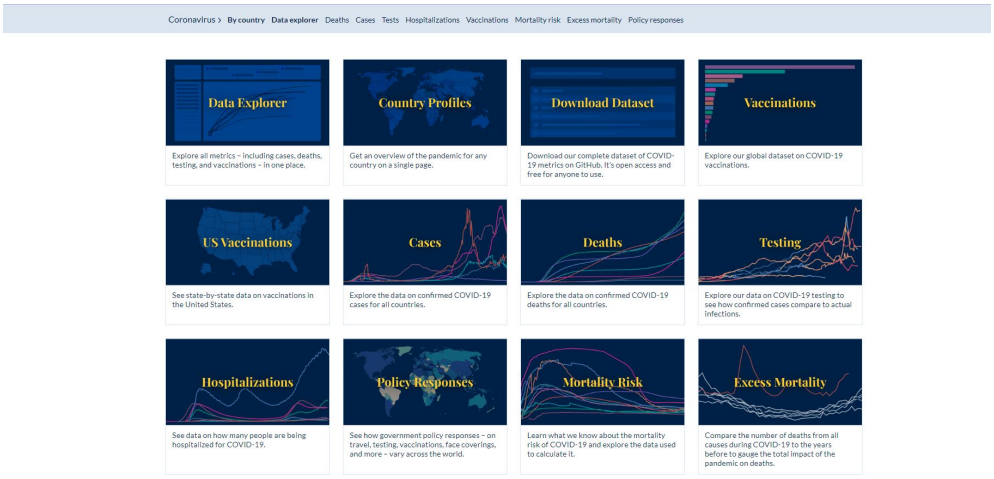
2.1 Estado da arte

A partir dos obxectivos detallados no Apartado 1.2, mergullámosnos agora en alternativas con obxectivos similares, enriquecendo o coñecemento sobre experiencias xa desenvoltas neste ámbito e detectando puntos débiles que o noso proxecto contempla.

A pandemia da COVID-19 tivo un impacto importante na sociedade mundial, o que se viu directamente reflectido na gran cantidade de ferramentas de todo tipo que se construíron para aportar solucións e abordar os novos desafíos. Estas ferramentas foron creadas co obxectivo de ofrecer solucións, mitigar efectos adversos e adaptarse á nova realidade imposta pola pandemia. Seleccionáronse, entón, as ferramentas máis parecidas ao proxecto desenvolvido neste TFG, as cales se comentan a continuación.

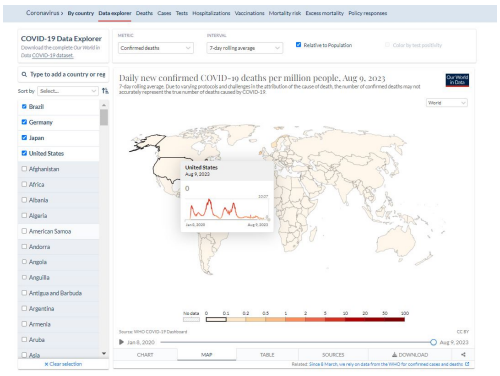
A primeira ferramenta é **COVID-19 Data Explorer** [3] que pertence a Our World in Data [4], proxecto en liña que ten como obxectivo presentar datos e análises enriquecidas sobre unha ampla variedade de temas globais, dende a saúde e a educación ata o medio ambiente e o desenvolvemento económico. Foi fundado por Max Roser en 2011 e enfócase en proporcionar visualizacións interactivas e narrativas baseadas en datos confiables e verificables. No

caso concreto de COVID-19 Data Explorer, ofrece información e perspectivas relacionadas coa COVID-19, incluídos datos sobre casos, mortes e vacinas, así como artigos de investigación e actualizacións de noticias. A páxina web está deseñada para ser un recurso integral para aqueles interesados na investigación e os datos sobre a COVID-19, incluíndo visualizacións interactivas dos datos da COVID-19, como mapas e gráficos, para axudar aos usuarios a comprender a propagación e o impacto do virus. A páxina web tamén proporciona ligazóns a artigos de investigación e actualizacións de noticias relacionadas, que poden utilizarse para informar a toma de decisións e o desenvolvemento de políticas. Ademais, actualízase periodicamente cos datos e as investigacións máis recentes, o que garante que os usuarios teñan acceso a información máis actualizada.



(a) Páxina principal de COVID-19 Data Explorer

(b) Visión en cadro da páxina de Data Explorer



(c) Visión en mapa da páxina de Data Explorer

Figura 2.1: Páxina web COVID-19 Data Explorer [3]

Se se entra en detalle, nada máis abrir a páxina ofrécese unha multitude de opcións pa-

ra elixir en que subtema da COVID-19 se desexa indagar. Esta primeira páxina móstrase na imaxe 2.1a da figura 2.1. Se se elixe, por exemplo, mostrar a páxina de Data Explorer, que é a que mostra a visión máis xeral, como se ve na figura 2.1b., dende esta páxina aínda se pode acceder a un menú xeral, que se mostra tamén na parte superior da figura 2.1b, ademais de numerosos controis na propia páxina que permiten escoller a granularidade temporal, os países sobre os que mostrar a información, a escala dos datos (lineal ou logarítmica) e o indicador que se prefire explorar, entre os que hai unha ampla variedade de opcións relacionadas con número de mortes, vacinacións ou casos, entre outras. Ademais, permite visualizar os datos nun mapa do mundo (imaxe 2.1c), para poder acceder fácilmente ás gráficas de cada país, solo con pulsar na súa rexión.

A segunda ferramenta é **SensePlace2** [5], que presenta un enfoque de análise xeovisual para apoiar o coñecemento situacional dos eventos de crise utilizando datos de Twitter. O enfoque céntrase en aproveitar a información xeográfica dos chíos, desenvolver esquemas de indexación e proporcionar métodos de interface visual que permitan comprender os compoñentes de lugar, tempo e tema das situacións en evolución.

A funcionalidade de SensePlace2 abarca cinco compoñentes clave: o panel de consultas, a visualización e control da liña do tempo, a lista de chíos, o mapa de chíos e a vista do historial. O panel de consultas permite especificar termos para búsquedas baseadas en texto. A visualización e o control da liña do tempo posibilitan o filtrado temporal. A lista de chíos mostra chíos relevantes cos seus metadatos. O mapa de chíos ofrece unha visión xeral da distribución xeográfica, cun mapa de calor e ubicacións destacadas. A vista do historial almacena consultas para reutilización e permite axustar os filtros. Para lograr isto, conta cun rastrexador de Twitter distribuído integrado cunha base de datos PostgreSQL para o almacenamento. O rastrexador utiliza a API de Twitter para recuperar chíos relevantes a palabras clave introducidas polo usuario (como 'terremoto' ou 'ciclón') nun período de sete días. Todo isto amósase na figura 2.2.

Ambas ferramentas comparten obxectivos similares cos descritos para o noso proxecto e son bastante completas, atractivas e intuitivas para o usuario. Con todo, no caso do COVID-19 Data Explorer, non se considera o análise das redes sociais, unha parte clave. Ademais, non se realiza un estudo de sentimentos, senón que se traballa unicamente con datos obxectivos sobre casos confirmados. Por outra banda, o SensePlace2 alíñase máis cos obxectivos da nosa aplicación, xa que permite a busca de chíos relacionados coa pandemia e utiliza xeorreferencias para representalos nun mapa de calor, o cal se pode filtrar temporalmente. Non obstante, non ofrece unha visión tan holística como a que se persigue. É necesaria unha nova proposta que satisgafa esa idea, que proporcione máis posibilidades de visualización e interacción, ademais de incluír unha análise de sentimentos que permita explorar outra percepción importante da pandemia: as emocións que esta suscitou nas redes sociais.

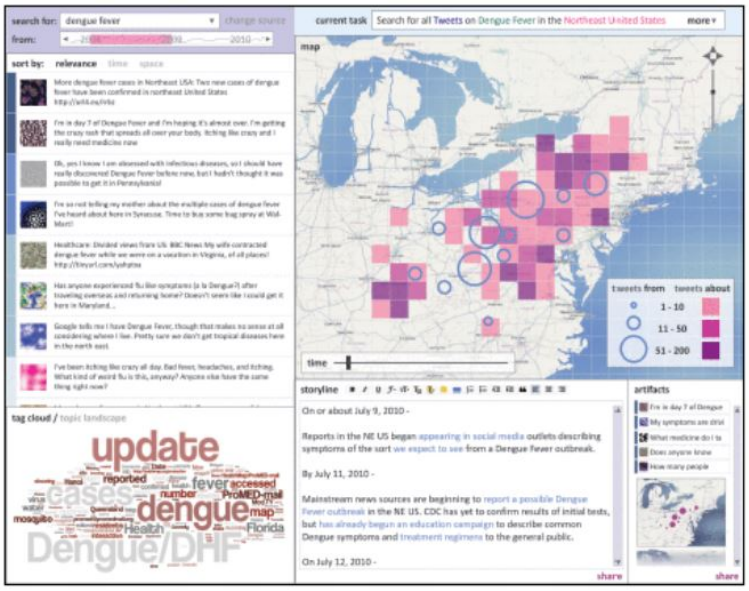


Figura 2.2: Páxina principal da aplicación SensePlace2
[5]

Un problema fundamental que se detecta na segunda ferramenta e que tamén está presente noutros traballos explorados [6] [7], é o feito de traballar en tempo real cos datos obtidos a través da API de Twitter, que debido a cambios recentes na política [8], ao pasar a ser de pago, deixou fora o acceso a moitas persoas: "Con estos precios muchos de los desarrolladores que hoy cuentan con productos o proyectos dependientes de la API de Twitter simplemente han quedado sin muchas posibilidades de acceso a la plataforma o, en el mejor de los casos, tendrán que revisitar los costos de sus servicios para poder justificar un pago de 42.000 dólares" [9]. De feito, este un dos problemas que se tivo para iniciar o proxecto que estamos a tratar, solventado finalmente coa utilización de técnicas de *web scrapping*.

2.2 Tecnoloxías utilizadas

- **Selenium [10]**: Ferramenta amplamente utilizada para a automatización de probas de páxinas web. Proporciona unha API que permite controlar un navegador web, interactuar con elementos da páxina, enviar entradas e verificar resultados. É moi útil para automatizar tarefas repetitivas de navegación web ou probas funcionais.

- **Python [11]**: Linguaxe de programación de alto nivel, coñecida pola súa sintaxe clara e lexible. Utilízase en diversos ámbitos, desde desenvolvemento web ata análise de datos

e intelixencia artificial.

- **SQLAlchemy [12] e GeoAlchemy [13]**: SQLAlchemy é unha biblioteca de Object-Relational Mapping (ORM) para Python, que permite interactuar con bases de datos relacionais usando obxectos Python en vez de escribir consultas SQL directamente. GeoAlchemy é unha extensión que engade características específicas de xeolocalización, permitindo o uso de tipos de datos xeoespaciais en bases de datos.

- **PostgreSQL [14] e PostGIS [15]**: Sistema de xestión de bases de datos relacional de código aberto e altamente extensible. Ofrece características avanzadas e soporta tipos de datos xeoespaciais, o que o fai popular para aplicacións que requiren recursos de localización. PostGIS é unha extensión espacial que engade soporte para datos xeoespaciais e funcionalidades xeoespaciais avanzadas, como almacenamento e análise de datos xeoespaciais.

- **Flask [16]**: Framework lixeiro para desenvolvemento web en Python. Facilita a creación de aplicacións web e APIs de forma sinxela e flexible.

- **HTML [17], CSS [18] e JS [19]**: Tecnoloxías fundamentais para o desenvolvemento web. HTML utilízase para estruturar o contido, CSS para a estilización e deseño, e para a interactividade e funcionalidades dinámicas.

- **Leaflet.js [20]**: Biblioteca JavaScript de código aberto para creación de mapas interactivos. Utilízase amplamente para mostrar datos xeoespaciais en aplicacións web.

- **Servizo de Integración de SQL Server (SSIS) [21]:** Plataforma deseñada para axudar na integración de datos de varias fontes, na transformación destes datos segundo sexa necesario e na súa carga nun destino final, como unha base de datos ou un ficheiro.

- **MicroStrategy [22]**: Plataforma de análise de negocios que permite explorar e visualizar datos para a toma de decisións. Ofrece recursos para crear paneis interactivos e informes.

- **Power BI [23]**: Ferramenta de análise de datos e visualización desenvolvida por Microsoft. Permíteche conectar a diversas fontes de datos, crear modelos e crear informes interactivos e paneis de control. Power BI é moi utilizado para transformar datos en información visual comprensible.

- **Pandas [24] e GeoPandas [25]**: Pandas é unha biblioteca Python para análise e manipulación de datos. Ofrece estruturas de datos flexibles para traballar con táboas e series temporais. GeoPandas é unha extensión do Pandas que engade soporte para datos xeoespaciais, combinando capacidades de análise de datos con xeolocalización.

- **Seaborn [26], Plotly [27] e Matplotlib [28]**: Seaborn é coñecida polos seus gráficos estatísticos atractivos e de fácil creación. Plotly proporciona a capacidade de crear gráficos interactivos e visualizacións de datos dinámicas en Python. Matplotlib ofrece un control detallado sobre a creación de gráficos estáticos, dinámicos e interactivos, permitindo unha personalización avanzada. Estas bibliotecas son esenciais para representar datos de maneira efectiva e comunicar ideas de maneira clara e visualmente atractiva, cada unha coas súas características propias. Por iso se foron necesarias as tres, según os diferentes obxectivos a acadar.

- **VADER Sentiment [29], NLTK [30] e TextBlob [31]**: Bibliotecas de Natural Language Processing (NLP) en Python. VADER Sentiment utilízase para análise de sentimentos en texto, Natural Language Toolkit (NLTK) é unha biblioteca completa para tarefas de NLP e TextBlob simplifica tarefas de procesamento de texto.

- **NumPy [32] e Math [33]**: NumPy é unha biblioteca fundamental para computación numérica en Python. Ofrece soporte para matrices multidimensionais e funcións matemáticas. O módulo Math ofrece funcións matemáticas máis básicas.

- **Requests [34]**: Utilízase para facer peticións HTTP en Python, permitindo interactuar con APIs e páxinas web.

- **CSV [35]**: Proporciona funcionalidade para ler e escribir ficheiros Comma-Separated Value (CSV), un formato común para datos tabulares.

- **re [36]**: O módulo ofrece soporte a expresións regulares en Python, permitindo a busca e manipulación de patróns en cadeas de texto.

- **WordCloud [37]**: Utilízase para crear nubes de palabras a partir dun texto, mostrando visualmente as palabras máis frecuentes.

- **Chardet [38]**: Emprégase para a detección de codificación de caracteres en texto, axudando á interpretación correcta de datos con diferentes codificacións.

Capítulo 3

Metodoloxía e planificación

Neste capítulo analizaranse as pezas claves que permitiron o correcto desenvolvemento do proxecto. Esta análise comprenderá varios apartados que detallarán cada aspecto esencial. No Apartado 3.1, examinarase en profundidade a metodoloxía elixida, resaltando os motivos que nos levaron á súa selección e subliñando os beneficios que pode aportar ao contexto do noso proxecto. No Apartado 3.2, mergullarémonos na planificación inicial, describindo as tarefas esenciais identificadas e os recursos necesarios para a execución do proxecto, así como os cambios que puideron ocorrer na planificación, os contratempos que se enfrontaron e a lóxica detrás destas modificacións. Por último, no Apartado 3.3, ofrécese un resumo dos custos relacionados cos recursos utilizados ao longo do proxecto.

3.1 Metodoloxía de desenvolvemento

A metodoloxía elixida é unha variante iterativa incremental [39], onde se incorporan as mellores prácticas de Scrum [40], aínda que non se aplica na súa totalidade debido á limitación de contar só cunha alumna como desenvolvedora. Este enfoque de desenvolvemento de software céntrase no crecemento progresivo da funcionalidade do sistema. En lugar de intentar construír o sistema completo dende o inicio, como nas metodoloxías tradicionais, o proxecto nas metodoloxías áxiles divídese en bloques temporais máis pequenos, coñecidos como iteracións, e cada iteración constrúese sobre a anterior, permitindo un avance gradual.

O proceso dunha metodoloxía iterativa incremental, o cal podemos ver na figura 3.1, inicia cunha implementación simple dos requisitos establecidos e mellora de forma iterativa a través de versións sucesivas nas iteracións seguintes, ata acadar a implementación completa do sistema. Ao longo do desenvolvemento, o enfoque fundamental reside en introducir novas funcionalidades e melloras baseadas no valor proporcionado a través de comentarios de revisión e avaliación. Por esta razón, o núcleo deste enfoque radica en aprender de cada ciclo de desenvolvemento e adaptarse ás cambiantes necesidades tanto do proxecto como dos

usuarios. Esta foi unha das principais razóns para escoller esta metodoloxía, ademais da non secuencialidade das iteracións, que permite traballar en varias iteracións ao mesmo tempo, mentres que unhas non interfiran noutras, o que aumenta a eficiencia e permite unha maior adaptabilidade ao ritmo cambiante das necesidades do proxecto.

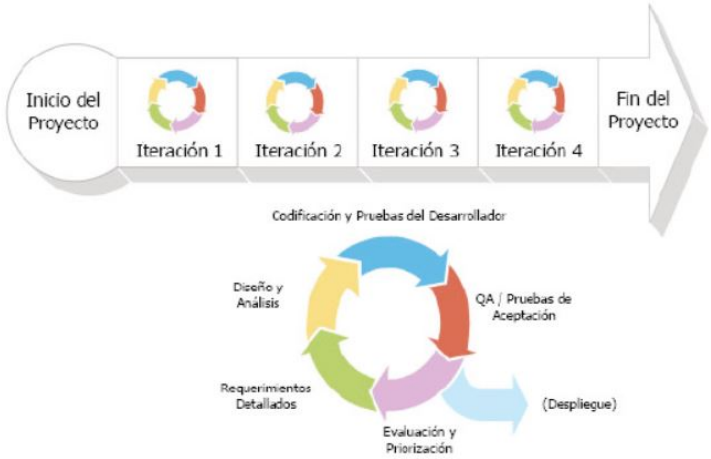


Figura 3.1: Funcionamento dun proxecto iterativo incremental
[41]

No caso concreto deste proxecto, cada unha das iteracións comprende un período de 2 semanas, cada unha das cales engloba as catro fases que se relatan a continuación:

- **Planificación e deseño**: Defínense os obxectivos SMART a conseguir no proxecto [42]. Os obxectivos Specific, Measurable, Achievable, Relevant and Timely (SMART) son específicos, medibles, alcanzables, relevantes e limitados no tempo. Durante esta etapa, establécese o alcance do traballo para a iteración en curso, definindo as metas a acadar e os diferentes procesos necesarios para logralas, así os requisitos, as medidas de aceptación ou as ferramentas a empregar. A planificación adáptase a medida que surxen novos obxectivos e circunstancias cambiantes, o que asegura unha maior flexibilidade e eficacia na execución do proxecto.

- **Implementación**: Lévanse a cabo os procesos marcados na primeira fase, de acordo á definición que se lles estableceu. Nesta fase, créase novo software para incluír novas funcionalidades e modifícase o anterior, se fose necesario, integrando todo correctamente no sistema que se ten ata o momento. A implementación implica a tradución do deseño e das especificacións en código funcional, seguindo os procedementos establecidos.

- **Probas**: As probas son moi importantes, pois permiten verificar e validar o software desenvolto. Nesta fase, realízanse as probas necesarias para asegurarse de que o softwa-

re cumpre cos requisitos establecidos e funciona de forma adecuada. Isto inclúe probas de unidade para comprobar o funcionamento das partes individuais do software, probas de integración para verificar a interacción entre as distintas partes e probas de sistema para avaliar o comportamento global do software. As probas permiten identificar erros e corrixir problemas antes da implementación final, asegurando así a calidade do produto final.

- **Revisión**: Lévase a cabo cos titores do proxecto nunha reunión presencial ou por Teams cada 2 semanas, comentando os avances e os obxectivos cumplidos así como as problemáticas que puido haber e a resolución de dúbidas de cara a seguinte iteración. Esta fase é fundamental no proxecto, para ter un punto de apoio crítico para evaluar o progreso, asegurarse de que o proxecto esté alineado cos obxectivos e realizar os axustes necesarios en función da retroalimentación recibida. Ademais, esta interacción cos titores permite un enriquecimiento do coñecemento e unha constante mellora do enfoque do proxecto.

3.1.1 Ferramentas de apoio á metodoloxía

- Balsamiq [43]: Ferramenta de software que se utiliza para a creación de prototipos no deseño de produtos dixitais, permitindo crear representacións visuais simplificadas das interfaces de usuario. Esta ferramenta é moi útil nas primeiras etapas do deseño, xa que permite crear bocetos rápidos. Foi utilizada para crear os *mockups* de deseño da aplicación, axudando a planificar a disposición dos elementos da interface e a visualizar os fluxos de usuario antes de comezar co desenvolvemento completo.

- DBeaver [44]: Ferramenta multiplataforma que proporciona unha interface gráfica para interactuar cunha ampla variedade de bases de datos, o que facilita a administración, a consulta e a visualización de datos almacenados en sistemas de bases de datos. Foi utilizada para comprobar a conexión á base de datos, así como o desplegue das táboas e realizar consultas sobre elas, de cara ao desenvolvemento da aplicación web.

- Draw.io [45]: Ferramenta de diagramación que permite crear diagramas de fluxo, procesos e outros gráficos. Empregouse para a representación do diagrama de Gantt e para os modelos conceptuais do proxecto.

- OneDrive [46]: Ferramenta de almacenamento na nube, que se utilizou para gardar distintas versión de código e probas, así como os *mockups*, os modelos conceptuais ou o seguimento das reunións cos titores.

- Overleaf [47]: Plataforma en liña que permite aos usuarios crear, editar e colaborar en documentos de LaTeX de forma sinxela. LaTeX é un sistema de composición de textos

que se emprega normalmente para crear documentos científicos, técnicos e académicos con un alto nivel de calidade tipográfica. Foi a ferramenta utilizada para a elaboración da memoria do proxecto.

- Visual Code Studio [48]: Ferramenta sinxela de manexar que ofrece unha gran vantaxe: a capacidade de traballar con diversos linguaxes de programación, como R e Python, e incluso de interactuar con cadernos Jupyter. En VSCode esribiuse todo o código deste proxecto, tanto da parte de ciencia de datos como da parte de enxeñaría de datos.

3.2 Planificación e seguimento

3.2.1 Planificación

A planificación consiste na elaboración dun plan detallado que servirá como guía para a execución e control das tarefas relacionadas co proxecto, así como a xestión de recursos, tanto humanos como materiais. Defínense os obxectivos do proxecto, así como as tarefas e os recursos necesarios para acadalos. Da mesma maneira, defínese o tempo contemplado para a execución da tarefa e os custos asociados, asígnanse as responsabilidades e establécese a vía de comunicación entre as persoas involucradas no proxecto.

Neste caso, a planificación do proxecto realizouse en dúas reunións claves. Na primeira reunión, abordouse a idea xeral do proxecto e exploráronse diversas vías de acción. Neste punto, estableceuse un primeiro enfoque conceptual e identificáronse as áreas principais a ter en conta, pero aínda era necesario profundizar no tema e definir mellor os obxectivos e as tarefas a realizar.

A segunda reunión, que tivo lugar despois dunha investigación máis profunda e a lectura de documentación relevante, marcou un paso crucial no proceso de planificación. Neste momento, xa se contaba cunha comprensión máis clara do contexto e dos desafíos do proxecto. Durante esta reunión, delineáronse con maior precisión os obxectivos, os alcances e as estratexias específicas a seguir. A discusión centrouse nunha idea máis refinada e establecéronse as bases para a execución do proxecto.

Estas dúas reunións de planificación serviron como pilares fundamentais para construír unha estrutura sólida e coherente para o proxecto, permitindo que as ideas iniciais evolucionasen cara a un enfoque máis estratéxico e concreto.

Despois destas dúas primeiras reunións de definición do proxecto, acordouse abordar as tarefas seguindo unha metoloxía iterativa incremental, dividindo o traballo en 6 iteracións que, á súa vez, tamén seguirían a execución de diferentes etapas, como se explicou no Apartado 3.1. A continuación, detállase o que se estableceu inicialmente para facer en cada unha desas iteracións.

- **Iteración 1.** Destinada a realizar probas coa API básica de Twitter e a leer documentación para conseguir un manexo fluído no momento no que se aceptara a licencia académica solicitada á aplicación (que era aproximadamente de 2 semanas según os usuarios do foro de Twitter). Necesitábase a licencia académica porque esta tiña moitas funcionalidades que a básica non tiña, moitas en canto a eficiencia mais a fundamental era que permitía recolectar chíos xeolocalizados, algo que a básica non contemplaba e que era fundamental para o proxecto, basado nun Sistema de Información Xeográfica (SIG).

- **Iteración 2.** Pensada para a obtención do *dataset* de chíos relacionados con diversos temas, co obxectivo de poder estudalos en conxunto, levando a cabo unha primeira exploración exhaustiva do conxunto de datos obtido, analizando os seus patróns, características e posibles problemas. Isto permite ter unha visión inicial sobre a información que temos e comezar a pensar en como podería estruturarse a base de datos. Nesta iteración, tamén se elabora un esbozo da futura estructura da base de datos, tendo en conta os datos recopilados e as necesidades do noso proxecto.

- **Iteración 3.** Unha vez completada a exploración inicial e tendo a estrutura da base de datos en mente, comeza o traballo máis substancial. Nesta fase, aplícase un proceso Extract, Transform, Load (ETL) para depurar os datos e transformalos nun formato máis axeitado para o análise posterior. Identifícanse e elimínanse datos duplicados, valores incorrectos ou faltantes, e aplícanse diversas transformacións para homoxeneizar os datos.

  Ademais, nesta etapa, impleméntase a estrutura definida previamente para a base de datos. Isto inclúe a creación de táboas, definición de relacións entre elas e a organización xeral dos datos dentro da base de datos.

- **Iteración 4.** Neste punto do proxecto, procédese cunha análise detallada dos datos que foron previamente procesados, utilizando diversas técnicas e ferramentas para explorar as relacións entre variables nos datos, identificando patróns, tendencias ou posibles ideas que poidan axudar a comprender mellor a información contida no *dataset*. Esta análise profunda pode incluír a creación de gráficas, histogramas e outras representacións visuais para visualizar as relacións entre os datos.

  Ademais, nesta etapa, comézase a traballar nos primeiros deseños do visor web. Isto implica a definición da arquitectura da interface, o deseño das primeiras páxinas e unha primeira visión do deseño da estrutura xeral.

- **Iteración 5**. Remátase coa parte da análise, profundizando nas emocións e percepcións dos usuarios, algo moi atractivo e interesante, por medio dunha análise de sentimentos. Esta parte da análise precisa do seu propio proceso de limpeza, para adecuar os datos

a este tipo de análise concreto. Por outro lado, comézanse a considerar as funcionalidades específicas que o visor debe proporcionar aos usuarios, como filtros interactivos, búsqueda de datos e posibles formas de representación gráfica dos resultados da análise xeral do conxunto de datos, realizándose os *mockups* das páxinas restantes.

- **Iteración 6.** Unha vez que a análise dos datos está completada e se ten unha comprensión clara do contido que presentar na aplicación web, avánzase coa etapa de implementación. Nesta fase, comézanse a crear as primeiras páxinas do noso visor web. Utilizanse as tecnoloxías apropiadas para desenvolver a interface do usuario, asegurando a súa intuitividade e que sexa fácil de navegar.

  Ademais, nesta iteración, trabállase en como representar un cadro de mando que englobe as representacións obtidas do proceso de análise. Isto implica a creación de gráficas, gráficos e outros elementos visuais que amosen as conclusións da análise de maneira clara e efectiva. Estúdase como organizar e presentar esta información de forma que sexa comprensible para os usuarios finais, e que lles permita obter ideas valiosas sen esforzo.

- **Iteración 7.** Nesta etapa avanzada do noso proxecto, céntranse os esforzos na implementación das páxinas que quedan pendentes na aplicación web, dando continuidade á creación de novas páxinas e funcionalidades, asegurándose de que a interface do usuario sexa coherente e fluida en todo o sistema.

  Un dos aspectos clave nesta iteración é a implementación do cadro de mando previamente deseñado. Créanse as representacións gráficas e visuais que compoñen o cadro de mando, mostrando os resultados da análise de maneira organizada e comprensible. Este cadro de mando pode incluír gráficas de tendencias, comparacións entre variables ou calquera outra información relevante para os usuarios.

  Unha vez completada esta última iteración, realízase unha revisión global do proxecto para garantir que todas as pezas encaixan correctamente. Compróbase que as páxinas, funcionalidades e o cadro de mando estean integrados de forma sinxela e eficiente, creando unha experiencia de usuario fluída e coherente. Por último, procédese coa redacción da memoria do proxecto.

Podemos visualizar de maneira máis clara a planificación do proxecto, coas súas correspondentes iteracións e o tempo estimado, no diagrama de Gantt da figura 3.2. As tarefas planificadas móstranse en azul.

3.2.2 Seguimento

Aínda que a o ideal sería ceñirse á planificación e estimación inicial dun proxecto, isto non é sempre posible, debido a diversos factores que poden influír no desenvolvemento do mesmo. A realidade é que os proxectos están expostos a incertezas, cambios nas circunstancias e imprevistos que poden alterar a traxectoria previamente establecida.

No caso particular deste proxecto, houbo certos problemas técnicos, sobre todo no inicio debido a grandes cambios que sufriu a plataforma (agora X), aos que adicionalmente se lle sumou a compaxinación das obligacións laborais da alumna nunha empresa externa cos estudos na universidade, así como a eventos como o periodo de exames na facultade e motivos personais de enfermidade, que impactaron directamente no progreso do proxecto.

Para explicar un pouco mellor os desvíos no tempo por problemas técnicos, vamos a enumerar de novo as iteracións nas que surxiron problemas, explicando o motivo dos mesmos e o retraso que conlevaron.

- **Iteración 2**. Esta iteración conlevou un retraso importante, debido a modificacións na API de Twitter, que pasou a ser de pago [49]. Ante este imprevisto, despois dunha reunión cos profesores, prodeceuse a explorar dous *datasets* con chíos doutros anos, proporcionados polos titores. Pero rapidamente se detectou un problema, que xa estaba contemplado despois de leer artigos sobre o índice de datos xeorreferenciados na plataforma. O problema nos conxuntos estudados era que o porcentaxe de chíos con referencia xeográfica, algo indispensábel para a nosa plataforma web, era apenas do 3% do total, supoñendo apenas 100 chíos de temas moi variados ou incluso algúns sin ningún tema definido, complicando enormemente o obxectivo definido para o proxecto.

  A solución foi buscar datos xeorrefenciados sobre un tema, que xa estiveran recopilados por alguén que os poidera recuperar no seu día coa API, mais esto non foi tarea fácil, posto que a única información que se compartía eran os ids dos tweets. Por tanto, sen ter acceso á API, voltábamos a estar no punto de partida. Definiuse entón un tema único, a COVID-19, para que fora máis sinxelo e plantexouse unha alternativa de *web scrapping*, comentada na sección 4.1, para poder conseguir información dos chíos sen necesidade de ter licencia para a API, a partir dos ids. Esto supuxo un retraso adicional de formación neste ámbito, posto que no grao foi visto moi superficialmente, pero deu resultado e obtemos un conxunto de datos xeorreferenciado, a nivel de cidade, sobre a COVID-19 en EEUU.

- **Iteración 3**. Problemas coa xeorreferenciación dos chíos a partir do nome da cidade, posto que hai cidades co mesmo nome en estados distintos e, da forma que se estaba a proceder, esta problemática non estaba contemplada. Foi algo sinxelo de resolver, mais supuxo unha demora con respecto á estimación inicial para a iteración.

- **Iteración 6**. Unha das páxinas do visor web non presentaba o contido de maneira intuitiva para o usuario, debido á maneira na que están xeorreferenciados os datos, a nivel de cidade, de xeito que non se apreciaban visualmente todos os chíos porque os que pertencían a unha mesma cidade, estaban a ser representados polas mesmas coordenadas, superpóndose. A solución foi agregar unha variable delta pequena, para que os datos puideran aparecer sutilmente diferenciados na representación visual.

- **Iteración 7**. A ferramenta plantexada para o cadro de mando na iteración 6 foi Microstrategy. Solicitouse unha licencia para poder empregala, pero esta demoraba moito. Por tanto, solicitouse á empresa na que traballa a alumna, se se podía utilizar unha licencia interna da empresa. A empresa estivo de acordo pero á hora de comentarlle a integración na páxina web, aconsellaron empregar outra ferramenta, por si durante este mes se facían reformas que afectaran a posterior visualización na defensa do traballo. Por tanto, aínda que o traballo incialmente se fixo en Microstrategy e vai estar documentado na memoria, houbo que facer un desenvolvemento noutra ferramenta, replicando o de Microstrategy na medida do posible, para poder integralo no visor web e permitir a interacción do usuario. A ferramenta elixida foi Power BI, ao ser unha ferramenta que xa fora vista no grao e permitía partir dunha base coñecida, ademais do seu destacado desempeño e posición no cadro de Gartner [50].

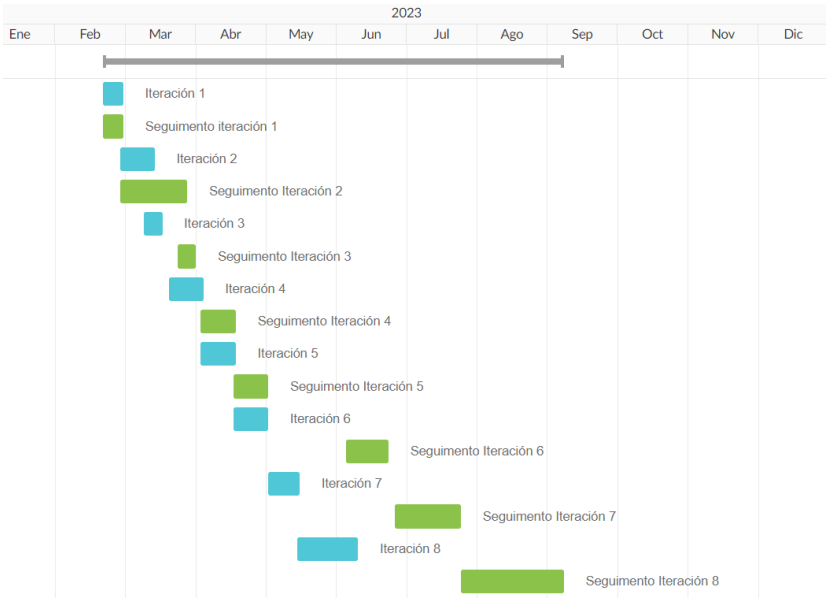


Figura 3.2: Diagrama de Gantt

Todos estos contratempos técnicos, sobre todo iniciais, conxuntamente cos retrasos persoais conlevaron a que o proxecto que estaba previsto para entregar en xuño, tivera que alar-

garse ata a seguinte convocatoria en setembro. Tamén podemos ver o diagrama de Gantt do seguimento real do proxecto, na figura 3.2, coas tarefas en cor verde.

3.3 Custos

3.3.1 Custos recursos humanos

Os custos humanos comprenden os custos asociados as persoas implicadas no proxecto. Neste caso, son tres persoas, a estudante e dous titores.

Para calcular os custos asociados á estudante, vaise partir do soldo medio dunha enxeñeira en España [51], que é de aproximadamente 24956€ anuais. Si se ten en conta que o proxecto de fin de grao comprende un total de 12 créditos *ETS* e que a cada crédito lle pertence unha dedicación de 25 horas de traballo, podemos estimar que o alumno dedicou un total de 300 horas. Como o salario medio é de 24956€ e sábese que nun ano hai 1776 horas laborais, o soldo dunha enxeñeira de datos junior en España é de 14.05€/h. Por tanto, o custo total por esta parte é de 4215€.

Para calcular os custos asociados aos titores, temos que ter en conta os seus postos educativos, un deles como profesor axudante doutor e outro como profesor interino de sustitución [52], dando lugar a soldos de 30732 e de 29408, respectivamente, o que fan un soldo de 17.30€/h e de 16.56€/h. Sabendo que houbo un total de 7 reunións de revisión con duración aproximada de 1h e, a maiores, unha reunión de apoio para a organización da memoria de 1h, temos un custo total por esta parte de 270.88€.

Temos como custo humano un total de 4485.88€, tal como se representa na táboa 3.1.

Recursos	Custo
Humanos	4485.88€
Materiais	800€
Custo total	**5285.88€**

Táboa 3.1: Táboa custos recursos humanos

3.3.2 Custos recursos materiais

Os custos materiais comprenden, neste caso, o custo do ordenador portátil da alumna, que foi o único material empregado, un HP Pavilion Laptop 15-cs2xxx cun procesador i5, valorado en 800€, aproximadamente. Polo tanto, o custo material ascende a un total de 800€, como se pode ver na táboa 3.1, onde podemos ver tamén o custo total do proxecto.

Capítulo 4

Análise

Neste capítulo, realizarase unha análise detallada dos datos. Comezarase obtendo os datos necesarios e facendo unha análise preliminar para continuar cunha limpeza e preprocesado, incluindo a modificación de tipos de datos, tratamento de valores nulos e inclusión de novas columnas de valor. A continuación, levarase a cabo unha análise exploratoria das variables, máis detallada, onde se realizarán análises estatísticas e representacións visuais das variables cuantitativas e cualitativas. Finalmente, rematarase o capítulo cunha análise de sentimentos, abordando as fases da análise, as ferramentas utilizadas e o proceso de análise en Python.

4.1 Obtención dos datos

Unha parte fundamental neste proxecto foi a obtención do conxunto de datos, pois debíase buscar un conxunto de datos sobre o que traballar. Neste proceso, atravesáronse distintas fases e desafíos que se detallan a continuación.

Primeiramente, comezouse coa solicitude dunha licenza para acceder á API de Twitter. Mentres se agardaba a súa concesión, comezouse a traballar cunha versión básica da API, a cal tiña as súas limitacións. Esta versión permitía obter menos chíos ao día e non proporcionaba coordenadas xeográficas (só deixaba coñecer se o chío en cuestión estaba xeorreferenciado ou non), que eran esenciais para a análise. Para obter os chíos, tivéronse que realizar buscas por palabras clave e extraelos en lotes de 100 chíos de cada vez. Esta vía non era unha boa opción polas desvantaxes que tiña e só se tratou para poder ir coñecendo a API e traballar con ela para que despois o tratamento coa API académica fose máis sinxelo, ademais de investigar a cantidade de chíos que estaban xeorreferenciados, posto que nunha primeira documentación sobre o tema xa se prevía que esta porcentaxe era extremadamente baixa [53], o que podía complicar o proxecto. Podemos ver na figura 4.1, como foi a evolución dos chíos xeorreferenciados nos últimos anos, tanto con coordenadas precisas como con xeorreferencias de lugar, a nivel de cidade. Ante esta problemática, que se confirmaba na proba feita inicialmente coa

API básica, unha solución plantexada era recuperar todo tipo de xeorreferencias e non só as que tiñan coordenadas precisas, xa que se poderían obter moitos máis datos, posto que na actualidade hai maior riqueza de etiquetas de lugar ca de coordenadas GPS nos chíos.

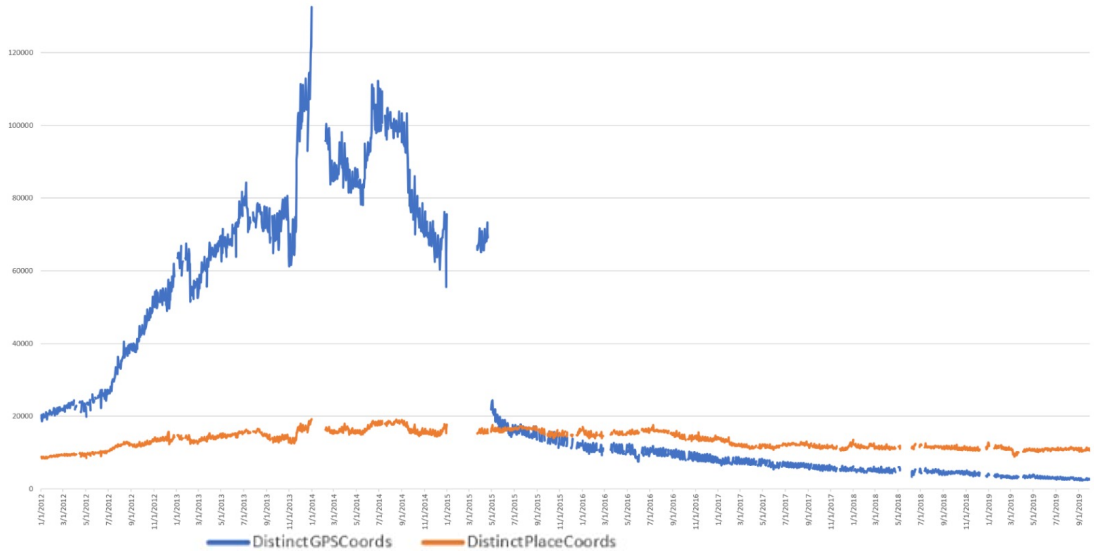


Figura 4.1: Evolución ao longo de 7 anos da cantidade de chíos xeoetiquetados [54]

De todas formas, presentouse un novo problema. A licenza non daba chegado dado que a situación da aplicación cambiou en febreiro de 2023, cando Twitter modificou a súa API e comezou a requerir unha tarifa polo seu uso [49]. Esta nova limitación levou a explorar alternativas para obter os datos necesarios.

Os titores proporcionaran dous conxuntos de datos para considerar. Un deles datábase do ano 2012 e consistía en 337 arquivos con aproximadamente un millón de chíos cada un. Non obstante, nunha primeira análise viuse que só un pequeno porcentaxe destes estaban xeorreferenciados, un aspecto crucial para o noso análise. Ademais, os temas destes chíos eran variados e non específicos, o que dificultaba máis atopar unha cantidade suficiente que cumprira os requisitos de tema concreto e interesante, con xeorreferenciamento.

O segundo conxunto de datos, que xa databa do ano 2019, tamén presentaba desafíos, xa que o porcentaxe de xeorreferenciación era bastante baixo e as coordenadas non se concentraban en ningún país, polo que tamén dificultaban o obxectivo de ver a evolución sobre algún tema nalgún sitio concreto. Toda a información sacada destes arquivos represéntase na táboa 4.1, sendo o primeiro arquivo comentado nombrado como *data12*, o segundo como *data19*, *GPS* para representar os chíos con coordenadas exactas e *Lugar* para representar os que teñen coordenadas a nivel de lugar.

Podemos ver que a porcentaxe é moi baixa e aínda habería que filtrar por algún tema,

polo que no caso do *data19*, non obteríamos unha cantidade de rexistros suficiente. Porén, no caso do *data12*, se temos en conta que este arquivo só é unha partición do conxunto e que hai ao redor de 300 arquivos máis con características similares, se sería posíbel obter unha cantidade considerable de rexistros, despois de filtrar por unha temática. O principal inconvinte sería a temporalidade, pois son datos do ano 2012, que están bastante desactualizados e non proporcionan a mellor perspectiva no uso actual da ferramenta e nas tendencias actuais.

	data12	**data19**
GPS	3.430	171
Lugar	9.981	4.932
Cantidade total de chíos	1.000.000	650.968
Porcentaxe GPS + Lugar sobre o total	**1.34%**	**0.78%**

Táboa 4.1: Análise de dous arquivos proporcionados polos titores

Despois de sopesar as opcións, tomouse a decisión de investigar na internet e centrar a busca de datos no tema da COVID-19. Dado o tempo xa perdido no proceso de busca de datos, viuse óptimo marcar un tema para buscar. Elixiuse este debido á abundancia de datos e á relevancia deste fenómeno, ademais do seu carácter recente. Foi así que se atopou un conxunto de datos en liña que ofrecía IDs de chíos relacionados coa COVID-19 en Estados Unidos [55].

Estes IDs de chíos constituíron o punto de partida para a obtención de datos. Empregáronse técnicas de *web scraping* con Selenium [10] para acceder ás páxinas individuais de cada chío, usando os IDs correspondentes. A través da estrutura HTML das páxinas, puidose extraer información relevante dos chíos, como se ve no trozo de código de exemplo que se amosa a continuación:

```
def get_data(tweet_ids, driver, n):

    bar = IncrementalBar("Screapeando chíos: \t",
    max=len(tweet_ids),
                         suffix="%(index)d/%(max)d
    (%(percent).1f%%) - ETA:%(eta)ds")
    with open(f"./chíos_dataset_{n}.csv", "a", newline="",
    encoding="utf-8") as archivo_csv:
        escritor_csv = csv.DictWriter(archivo_csv, fieldnames=[
        "tweet_id", "usuario", "fecha", "chío", "me_gusta",
        "retweets", "citas", "ciudad", "estado"])

        escritor_csv.writeheader()

```

```
        for tweet_id in tweet_ids:
            tweet_data = {"tweet_id": tweet_id, "usuario": None,
            "fecha": None, "chío": None, "me_gusta": None,
            "retweets": None, "citas": None, "ciudad": None,
            "estado": None}

    driver.get(f"https://twitter.com/twitter/status/{tweet_id}")
            driver.implicitly_wait(10)

            try:
                usuario_elemento = driver.find_element(
                    By.XPATH,
                    "//div[@class="css-901oao css-1hf3ou5 r-18u37iz
                    r-37j5jr r-1wvb978 r-a023e6 r-16dba41 r-rjixqe
                    r-bcqeeo r-qvutc0"]//span[@class="css-901oao css
                    -16my406 r-poiln3 r-bcqeeo r-qvutc0"]",)
                tweet_data["usuario"] = usuario_elemento.text

            except:
                tweet_data["usuario"] = None
```

Cabe destacar a maiores que o proceso de extracción se fixo en catro partes, dividindo 12000 ids en 4 bloques distintos, que se procesaron por separado, pois dunha soa vez era computacionalmente moi costosa, levaba ao redor de 1 semana conseguir tódolos datos, e daba moitos problemas co arquivo unha vez se quería abrir ao rematar o proceso.

O proceso de obtención deste conxunto de datos presentou os seus desafíos, dada a restrición da API de Twitter e á baixa porcentaxe de chíos xeorreferenciados, así como o costo computacional que supuxo a extracción final dos chíos mediante *web scrapping*. Con todo, o resultado foi un conxunto de datos valioso e relevante, o cal proporcionou a base necesaria para realizar un análise xeográfico da COVID-19 nos Estados Unidos, con posibles implicacións para futuras pandemias.

4.2 Exploración inicial do conxunto de datos

Unha vez se obtén o conxunto de datos, da forma que se viu no Apartado 4.1, é fundamental realizar una visualización e análise inicial dos datos para adquirir unha comprensión preliminar da súa naturaleza e características. Isto permite identificar patróns, tendencias e posibles anomalías que poidan influir na análise posterior.

Por tanto, comézase cargando o CSV cos datos en Python, coa axuda da libraría pandas, que permite cargar datos de diversas fontes. Neste caso, emprégase o comando *pd.read_csv ("ruta_do_arquivo_csv")*. Nunha primeira instancia, estudarase o tamaño, o nome das colum-

nas, o tipo de datos e a cantidade de valores non nulos, así como unha primeira visualización dos datos, para a cal se decidiu mostrar os primeiros e últimos valores do conxunto. Todo isto vese na figura 4.2.

```
# Primeiros valores do dataset
dfCovid.head()
```
✓ 0.0s — Python

	tweet_id	usuario	fecha	tweet	me_gusta	retweets	citas	ciudad	estado
0	1238459356271828992	@brownsugar1966	2020-03-13T13:38:14.000Z	Good morning Friday it's a beautiful day keep...	1	0	0	North Lauderdale	FL
1	1226702953949560834	NaN	NaN	NaN	0	0	0	NaN	NaN
2	1241429053372071938	@barhogphil	2020-03-19T23:13:43.000Z	https://twitter.com/bananashannahs/status/1240...	0	1	0	San Antonio	TX
3	1250661116046385152	@JeriKMiller1	2020-04-15T02:48:31.000Z	Repubs all over America just keep insisting on...	0	0	0	NaN	Wisconsin
4	1255607013851254790	@mistreatedrq	2020-04-25T23:30:06.000Z	105'den çok daha fazla aile mağdur durumda emi...	7	1	0	Naperville	IL

```
# Últimos valores do dataset
dfCovid.tail()
```
✓ 0.0s — Python

	tweet_id	usuario	fecha	tweet	me_gusta	retweets	citas	ciudad	estado
12295	1252615236953219072	@BartNLutherKing	2020-04-21T15:08:39.000Z	Trump ain't no f*cking president. He's a KKKlo...	2	1	0	North Little Rock	AR
12296	1258263171749629955	@Ldallett	2020-05-07T01:56:50.000Z	There is no legitimate reason for the 2nd amen...	0	0	0	Manhattan	NY
12297	1221866954765398016	@mreddy940	2020-01-27T03:33:19.000Z	Here we go this is finally happening the outbr...	0	0	0	San Jose	CA
12298	1245778768523939841	@LatinaFarmerUSA	2020-04-02T18:22:58.000Z	#SocialDistancing Does not & Shouldn't equal #...	1	0	0	NaN	NaN
12299	1255186060319109121	@MarkDasher	2020-04-28T15:01:52.000Z	And here we go	0	0	0	NaN	Michigan

(a) Información sobre o conxunto de datos

```
# Tamaño dataset
dfCovid.shape
```
✓ 0.0s — Python

```
(12300, 9)
```

```
# Coñecer o nome das columnas
dfCovid.columns
```
✓ 0.3s — Python

```
Index(['tweet_id', 'usuario', 'fecha', 'tweet', 'me_gusta', 'retweets',
       'citas', 'ciudad', 'estado'],
      dtype='object')
```

```
# Información sobre tipos de obxectos e non nulos
dfCovid.info()
```
✓ 0.1s — Python

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12300 entries, 0 to 12299
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   tweet_id  12300 non-null  int64
 1   usuario   8953 non-null   object
 2   fecha     9025 non-null   object
 3   tweet     8856 non-null   object
 4   me_gusta  12300 non-null  object
 5   retweets  12300 non-null  int64
 6   citas     12300 non-null  int64
 7   ciudad    7632 non-null   object
 8   estado    8963 non-null   object
dtypes: int64(3), object(6)
memory usage: 865.0+ KB
```

(b) Visualización do conxunto de datos

Figura 4.2: Análise inicial do conxunto de datos

A partir desta primeira exploración, pódese observar que o *dataset* componse de 12300 rexistros para cada un dos cales se proporciona información con respecto a 9 variables. Os nomes das columnas (variables) realmente é algo que se elexiu durante a *web scrapping*, pero procédese a explicar o significado de cada unha, considerando tamén na explicación o tipo de datos e a cantidade de non nulos que se viron na exploración, aínda que estos se estudarán

con máis profundidade no Apartado 4.3.

- **tweet_id:** Conforma o identificador único de cada chío, polo que é un valor irrepetible por cada rexistro. Os ids en Twitter constrúense en base a unha aproximación Snowflake[56], solución para xerar ID únicos en sistemas distribuídos. Twitter usa este enfoque en chíos, mensaxes directas, listas, etc. Conxuntamente, os tres compoñentes que se explican a continuación, conforman o "Tweet ID" de 19 díxitos , asegurando que cada tuit teña un identificador único e permitindo que Twitter rastrexee e xestione os chíos na súa plataforma de forma eficiente [57].

  - Os primeiros 41 bits (bits 0-40): Os primeiros 41 *bits* do "Tweet ID" representan un valor de tempo único en milisegundos desde un punto de referencia específico, coñecido como "epoch" (xeralmente é un momento no pasado). Estes bits codifican a data e hora exacta na que se creou o chío. Como resultado, os chíos máis recentes terán un valor de tempo maior nestes bits.

  - Seguintes 10 bits (bits 41-50): Estes bits coñécense como "worker ID" e representan un identificador único asignado á máquina ou servidor que xerou o *Tweet ID*. Isto axuda a evitar conflitos de identificación cando múltiples servidores xeran *Tweet IDs* ao mesmo tempo.

  - Últimos 12 bits (bits 51-62): Estes *bits* denomínanse "sequence number" e utilízanse para manexar chíos que se xeran no mesmo milisegundo e desde a mesma máquina. Cada chío dentro do mesmo milisegundo recibirá un número de secuencia diferente para garantir a unicidade do *id*.

  Un exemplo de id é *1238459356271828992*. Dado que este *id* se utiliza para distinguir chíos individuais e non para cálculos matemáticos, é máis apropiado considerar que este dato é de tipo obxecto e non un enteiro, como está definido no *dataframe*. O tipo desta columna cambiarase posteriormente para asegurar a integración e unicidade dos datos.

- **usuario**: Conforma o nome único de usuario. O nome de usuario en Twitter é coñecido como "handle" ou "username". É o identificador único que escolle cada usuario para representar a súa conta en Twitter. Este precédese polo símbolo "@" e pode conter letras, números e guións baixos. É importante elixir un nome axeitado e representativo para a identidade en liña na plataforma, porque é a forma en que outras persoas poden atoparte e interactuar contigo na plataforma. Tamén aparece na URL do perfil (twitter.com/username) e utilízase para a mención en chíos, permitindo que outros usuarios identifiquen facilmente a outro en conversas. É un dato de tipo obxecto, polo que xa está correcta a definición da columna, e conta con 8953 rexistros non nulos, o cal equivale a un 0.26% de nulos.

- **fecha**: Data de creación do chío. A data dun chío é a marca de tempo que indica cando se publicou esa mensaxe na plataforma de Twitter. É importante porque fornece información sobre a cronoloxía da conversa e axuda aos usuarios a comprender cando ocorreron certos eventos ou interaccións na plataforma. A data amósase xunto co contido da mensaxe e frecuentemente inclúe detalles como o día, o mes, o ano e a hora na que se realizou a publicación. Neste caso, a data segue a convención internacionalmente recoñecida chamada ISO 8601 [58]. Esta convención é amplamente utilizada para representar datas e horas de forma consistente e estandarizada, facendo máis fácil a comprensión e a intercambio de información entre diferentes sistemas e rexións. Un exemplo para "2020-04-28T15:01:52.000Z" do que significa cada parte sería:

  - *2020-04-28*: Esta é a parte da data, co ano (2020), o mes (abril) e o día (28) representados nesa orde.

  - *T*: O "T" separa a parte da data da parte da hora.

  - *15:01:52.000*: Esta é a parte da hora, coa hora (15), os minutos (01), os segundos (52) e os milisegundos (000) representados nesa orde.

  - *Z*: O "Z" ao final indica que a hora está en Coordinated Universal Time (UTC), que é un estándar de tempo de referencia global. "Z" representa a abreviatura de "Zulu", que é a forma de pronunciar a letra "Z" no alfabeto fonético da OTAN e utilízase para indicar UTC.

  Por tanto, a data e hora "2020-04-28T15:01:52.000Z" representan o momento exacto o 28 de abril de 2020, ás 15:01:52 (hora, minutos e segundos) en UTC.

  Para este campo temos o tipo de dato obxecto e 9025 rexistros non nulos, que equivalen a un 0.25% de nulos. Os nulos estudaranse máis tarde na sección 4.3.2, así como o tipo de dato na sección 4.3.1, que será necesario cambiar a un formato *datetime*[59].

- **me gusta:** Os "me gusta" en Twitter son un recurso que permite aos usuarios mostrar o aprezo ou interese por un chío en particular. Funcionan como un indicador de que un usuario atopou o contido do chío valioso, interesante ou relevante. Cando un usuario fai clic no icono do corazón abaixo dun chío, este marcase como "me gusta". Para este campo non hai valores nulos, así como para os dous que se explicarán a continuación. Isto débese á configuración do *scrapper* empregado na extracción dos chíos, ao que se lle sinalou que se non atopaba o campo, establecese a cero o reconto. É dicir, se nun chío, non detectaba o campo de "Me gusta", estableciuse o seu valor a cero, como se ve no seguinte código.

```
1     try:
2         me_gusta_elemento = driver.find_element(By.XPATH,
3             f"//a[@href="/{tweet_data["usuario"[1:]}/status/
```

```
            {tweet_id}/likes"]//span[@class="css-901oao css-16my406
            r-poiln3 r-bcqeeo r-qvutc0"]",)

        tweet_data["me_gusta"] = me_gusta_elemento.text

    except:
        tweet_data["me_gusta"] = 0

```

Isto fíxose porque os chíos con 0 "me gusta" non tiñan este campo. Ademais, se non detectaba o campo, daba problemas a extracción e pódese recuperar igualmente os que son nulos, isto explicarase con máis detalle no seguinte Apartado 4.3.

O que si é importante destacar é que o tipo de dato está mal asignado, posto que é un valor númerico e non de tipo obxecto. Isto modificarase máis adiante.

- **retweets e citas:** Os "retweets" e as "citas" son función similares á de "me gusta" e, para non repetir os usos, o que se vai mostrar é unha táboa coas principais diferenzas entre as tres funcionalidades (táboa 4.2).

  Neste caso, para os "retweets" e "citas", o tipo de dato asignado se é correcto e o problema dos nulos é o mesmo comentado anteriormente para os "me gusta".

Me gusta	**Retweets**	**Citas**
Expresión de aprecio e acordo.	Comparten contido directamente.	Comparten contido con comentario adicional.
Interacción rápida e sen comentarios.	Amplifican e difunden rapidamente.	Enriquecen o contido orixinal.
Aparecen no teu perfil; visíbel para outros.	Aparecen no teu perfil e no *feed* de seguidores.	Aparecen no teu perfil e no *feed* de seguidores.
Indican aprobación sen compartir directamente.	Amplían o alcance dun chío.	Agregan perspectiva personal ao "retweet".
Sinalan participación en conversas.	Sinalan participación en conversas.	Sinalan opinión persoal co contido.

Táboa 4.2: Comparación de me gusta, retweets e citas en Twitter.

- **ciudad:** Refírese ao nome da localidade ou área urbana específica desde a que se escibiu o chío. Campo fundamental para conseguir o visor web, posto que permite xeorrefe-

renciar os chíos a partir do seu valor. Trátase dun campo de tipo obxecto e ten 7632 non nulos, que equivale a un 0.38% de nulos sobre os rexistros totais recuperados.

- **estado:** Refírese á división política máis grande nos Estados Unidos, coñecida como "estado". Os Estados Unidos están compostos por 50 estados diferentes, cada un co seu propio goberno e leis rexionais. Cada estado ten o seu propio nome e adoita representarse mediante a súa abreviatura de dúas letras, como "CA" para California, "NY" para Nova York, etc. A información do estado no conxunto de datos permite identificar en que parte do país se atopa a cidade específica. Neste caso, tamén é un campo de tipo obxecto e ten menos nulos que o campo "ciudad", cunha cifra de de 0.28% nulos (8963 rexistros non nulos).

4.3 Limpeza e preprocesado

A limpeza de datos, tamén coñecida como *Data Cleaning* o *Data Cleansing*, é un proceso que involucra detectar, eliminar, corrixir ou transformar calquera anomalía, perturbación ou irrelevancia dos datos. Antes de proceder cunha análise máis detallada, compre facer unha limpeza e preprocesado adecuado arreglando erros que detectamos na análise incial, tales como a modificación dalgúns tipos de datos, que se verá no Apartado 4.3.1. O por que dalgúns valores nulos, que tamén se verá no Apartado 4.3.2. Ademais, verase noutro Apartado 4.3.3, a creación de novas columnas que aporten valor, xunto cunha explicación do seu por que.

4.3.1 Modificación tipo de datos

Como se viu no Apartado 4.2, no *dataset* hai algunhas columnas que non teñen o tipo de dato ben definido. É importante que o tipo de dato dun *dataset* estea ben definido por varias razóns cruciais. Primeiro, o tipo de dato adecuado permite unha interpretación correcta dos valores contidos na columna, evitando erros de análise ou cálculos incorrectos. En segundo lugar, o tipo de dato ben definido é esencial para a realización de operacións estatísticas e de cálculo. Por exemplo, se unha columna debería conter valores numéricos, pero está almacenando texto, como é o caso da columna "me gusta", as operacións matemáticas ou estatísticas realizadas nesa columna non terán sentido e poderían levar a resultados irreais ou incorrectos. Outra razón importante é a eficiencia de almacenamento e procesamento. Cada tipo de dato ten unha representación en memoria e en disco específica. Se os tipos de datos non están ben definidos, podería haber un consumo innecesario de recursos, como memoria e capacidade de almacenamento. Ademais, a definición clara dos tipos de datos tamén facilita a limpeza e a preparación dos datos. Cando se sabe que tipo de información se agarda en cada columna, é máis sinxelo identificar e tratar valores atípicos, valores en falta ou datos inconsistentes. Por último, a definición precisa dos tipos de datos tamén é crucial para a interoperabilidade

e a compartición de datos entre diferentes plataformas e ferramentas. Diferentes sistemas informáticos poden tratar os tipos de datos de maneira diferente, polo que unha definición correcta asegura a consistencia e a correcta interpretación dos datos en diferentes entornos.

Os campos que tiñan mal definido o tipo de dato eran "tweet_id", "fecha" e "me gusta". O primeiro está definido como númerico cando debería ser de tipo obxecto e o segundo e o terceiro están definidos como obxecto cando deberían ser un *datetime* e un numérico, respectivamente. Amósase o cambio na figura 4.5. No caso de "me gusta", hai que eliminar a coma (,) propia do tipo de dato obxecto para poder pasalo a formato "int64".

```
# cambiar tipo de dato de tweet_id a object
dfCovid['tweet_id'] = dfCovid['tweet_id'].astype(object)

# cambiar tipo de dato de fecha a datetime
dfCovid['fecha'] = pd.to_datetime(dfCovid['fecha'])

# cambiar tipo de dato de me_gusta a int
dfCovid['me_gusta'] = dfCovid['me_gusta'].str.replace(',', '').astype('int64')

✓ 0.0s
```

Figura 4.3: Cambio do tipo de datos de "tweet_id", "fecha" e "me gusta"

```
dfCovid.info()
✓ 0.0s
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12300 entries, 0 to 12299
Data columns (total 9 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   tweet_id  12300 non-null  object
 1   usuario   8953 non-null   object
 2   fecha     9025 non-null   datetime64[ns, UTC]
 3   tweet     8856 non-null   object
 4   me_gusta  12300 non-null  int64
 5   retweets  12300 non-null  int64
 6   citas     12300 non-null  int64
 7   ciudad    7632 non-null   object
 8   estado    8963 non-null   object
dtypes: datetime64[ns, UTC](1), int64(3), object(5)
memory usage: 865.0+ KB
```

Figura 4.4: Novos tipos de datos

Figura 4.5: Modificación dos tipos de datos

4.3.2 Tratamento valores nulos

Tratar os valores nulos é crucial na análise de datos e na construción de modelos estatísticos e de *machine learning* por varias razóns importantes. Os valores nulos poden distorsionar os resultados, afectar a precisión dos modelos, levar a conclusións incorrectas, dificultar a comparabilidade, perder información valiosa e causar sesgos. A forma en que se manexan os valores nulos pode afectar a integridade das análises e dos modelos, polo que se deben utilizar técnicas como a imputación, interpolación ou eliminación de valores nulos de maneira axeitada para cada contexto. Vaise explicar brevemente en que consiste cada técnica, así como os posibles motivos de datos faltantes.

Tipos de ausencia de datos

"MCar," "MAR" e "MNAR" [60] son termos que se usan no contexto do manexo de valores nulos en conxuntos de datos. Estes termos describen diferentes patróns e razóns polas cales os valores poden estar ausentes nun conxunto de datos.

1. **MCAR (*Missing Completely At Random*):** Neste escenario, a ausencia de valores é completamente aleatoria. Iso significa que a probabilidade de que un valor estea ausente é a

mesma para todas as observacións. Neste caso, os valores nulos poden tratarse con técnicas xerais de imputación ou eliminación sen introducir sesgos.

2. **MAR (*Missing At Random*):** Neste caso, a ausencia de valores non é aleatoria en si mesma, pero está relacionada con outras variables observadas no conxunto de datos. Iso significa que a probabilidade de que un valor estea ausente depende das variables observadas. É posíbel abordar esta situación con técnicas de imputación e análise estatística que consideren as variables relacionadas.

3. **MNAR (*Missing Not At Random*):** Neste escenario, a ausencia de valores non é aleatoria e está relacionada con información non observada ou información que non foi recopilada no conxunto de datos. Este é o caso máis desafiante, xa que a ausencia de valores pode introducir sesgos nas análises e modelos se non se aborda axeitadamente.

Técnicas de imputación de datos

Hai moitas técnicas de imputación e hai que adaptarse ao contexto de cada problema. Aquí explícanse dúas das máis habituais, que dentro de si poden abarcar distintas técnicas.

1. **Imputación de Valores (*Imputation*):** A imputación implica estimar ou calcular valores para as entradas faltantes dun conxunto de datos. Isto implica a utilización de diferentes métodos para preencher os valores nulos co obxectivo de manter a integridade dos datos. Pódese usar información doutros datos ou estatísticas para realizar esta estimación.

2. **Eliminación de Valores (*Value Deletion*):** A eliminación de valores faltantes implica retirar ou ignorar as observacións que conteñen valores nulos do conxunto de datos. Isto pode ser feito cando se considera que a eliminación non afecta significativamente ás conclusións do análise ou modelo. Con todo, esta técnica debe usarse con precaución, xa que a eliminación de datos pode levar á perda de información valiosa e introducir sesgos nos resultados. A eliminación pode ser apropriada cando a porcentaxe de valores nulos é pequena e non afecta significativamente ás análises.

Solución para tratar os nulos do proxecto

No caso deste proxecto, hai nulos en todas as columnas, agás en "tweet_id", porque é un valor único e irrepetible, adxudicado a cada chío.

O motivo principal de datos faltantes é que estos datos son do 2020 e moitos usuarios eliminaron dende entón a súa conta, ou limitaron o acceso a ela. Isto deriva en que durante a extracción non se puido acceder as páxinas dos chíos, o que derivou en moitos valores nulos abarcando filas completas, afectando a todas as variables, coa excepción do "tweet_id". Esta problemática detectouse xa durante a extracción e móstrase a continuación a cantidade de rexistros afectados (figura 4.6), para os cales se realizou novamente a extracción de información tentando comprobar si todos estaban afectados pola problemática comentada. O

resultado foi positivo, no sentido de que se atopou que os rexistros que tiñan nulos en todas as columnas (excepto en "tweet_id") eran consecuencia de páxinas ás que non se accedía por erros de "páxina non encontrada" ou "a conta de usuario foi eliminada".

```
# Descubrir rexistros que teñen usuario nulo, fecha nula, tweet nulo, ciudad nula y estado nulo e me_gusta cero, retweets cero e citas cero
condicions = ((dfCovid['usuario'].isnull() & dfCovid['fecha'].isnull() & dfCovid['tweet'].isnull() &
               dfCovid['ciudad'].isnull() & dfCovid['estado'].isnull()) &
              (dfCovid['me_gusta'] == 0) & (dfCovid['retweets'] == 0) & (dfCovid['citas'] == 0))

registros_cumplen_condicions = dfCovid[condicions]
registros_cumplen_condicions.head(10)

dfCovid = dfCovid[~condicions]
```

✓ 0.0s Python

	tweet_id	usuario	fecha	tweet	me_gusta	retweets	citas	ciudad	estado
1	1226702953949560834	NaN	NaT	NaN	0	0	0	NaN	NaN
7	1237608361891221508	NaN	NaT	NaN	0	0	0	NaN	NaN
9	1258059680292765698	NaN	NaT	NaN	0	0	0	NaN	NaN
20	1237484136111247365	NaN	NaT	NaN	0	0	0	NaN	NaN
29	1239050850070401025	NaN	NaT	NaN	0	0	0	NaN	NaN
30	1254653157310754816	NaN	NaT	NaN	0	0	0	NaN	NaN
39	1241305929829756928	NaN	NaT	NaN	0	0	0	NaN	NaN
43	1254369975655190529	NaN	NaT	NaN	0	0	0	NaN	NaN
46	1255521924698046464	NaN	NaT	NaN	0	0	0	NaN	NaN
49	1235728097246396416	NaN	NaT	NaN	0	0	0	NaN	NaN

```
dfCovid.shape
```

✓ 0.0s Python

```
(9025, 9)
```

Figura 4.6: Tratamento dos nulos por chíos que xa non existen

Como se pode ver na figura 4.6, o *dataset* pasou de 12300 rexistros a 9025 coa eliminación. Porén, segue habendo presenza de nulos, que xa non se relacionan coa eliminación anterior e hai que descubrir de onde veñen. Nun primeiro lugar, miráronse as filas para as que o valor de usuario estivera a nulo e se se atopaban nulos a simple vista noutras variables, para esas filas, por comprobar se había algunha relación, mais esta non se atopou. Como segunda alternativa, miráronse as páxinas ás que levaba o "tweet_id" para ver se eran nomes especiais, que non fose o *scrapper* capaz de extraelos. Esta alternativa tamén fracasou, polo que se deduciu que simplemente na extracción non se chegou a conseguir ese contido, polo tempo asignado á recarga de cada páxina. A solución foi rechealos á man, pois apenas sucedía en 72 casos. Como resultado, só quedaban con nulos as columnas de "ciudad" e "estado". Tras un estudo similar, detectouse que os rexistros para os que ambas columnas eran nulos, eran chíos que non estaban xeorreferenciados. Aínda que para a análise xeográfica estos campos son esenciais, non se eliminaron os rexistros porque si podían aportar información para análises con outras variables. Todos os casos no que o estado era nulo debíase a este motivo, non obstante, para a cidade seguía a haber nulos. Pasaba algo similar ao caso de "usuario", pero eran máis rexistros. Por tanto, non se rechearon a man, colléronse os ids destes chíos para volver a intentar a extracción e conseguiuse pasar de 1376 rexistros nulos a 893. Por falta de tempo, decidiuse deixar así, conseguindo un *dataset* sen nulos, agás 893 no campo de cidade.

4.3.3 Inclusión columnas de valor

Por último, nesta etapa de limpeza e preprocesado, decidiuse crear algunha columna nova, de cara a facilitar a análise posterior. A maioría das columnas creáronse a partir de "fecha", para o que previamente se renomeou esta a "fecha_hora". Na figura 4.7 pódese ver o proceso de creación de cada columna. As columnas creadas foron:

- **fecha:** Campo creado a partir de "fecha_hora". Permite ter a fecha por separado, sen a hora, facilitando as análises que se fagan sobre o campo.
- **hora:** Campo creado a partir de "fecha_hora". Permite facer análises solo sobre o campo de hora, independientemente das datas.
- **fecha_id:** Campo creado a partir de "fecha_hora". Converte o formato a YYYYMMDD. Por exemplo a "fecha_hora" con valor "2020-03-13 13:38:14+00:00", pasa a ser "20230313". Decidiuse construír esta columna como unha mellora na eficiencia, no almacenamento e na simplicidade no manexo de datas. Por exemplo, as operacións de comparación ou de mínimo e máximo, son máis sinxelas.
- **dia_semana:** Campo creado a partir de "fecha_hora". Creouse para unha análise concreta sobre a cantidade de chíos en función do día da semana, porque este campo simplificaría a análise.
- **hashtags:** Campo creado a partir de chío, tamén para facilitar análises sobre os *hashtags*.

```python
# cambiar o nome da columna de fecha a fecha_hora
dfCovid.rename(columns={'fecha': 'fecha_hora'}, inplace=True)

# obter unha columna coa fecha sola e coa hora sola a partir da columna fecha_hora
dfCovid['fecha'] = [d.date() for d in dfCovid['fecha_hora']]
dfCovid['hora'] = [d.time() for d in dfCovid['fecha_hora']]

# sacar a partir de fecha_hora a fecha_id
for fecha_hora in dfCovid['fecha']:
    fecha_id = fecha_hora.strftime('%Y%m%d')
    dfCovid['fecha_id'] = fecha_id

# sacar a partir de fecha_hora o dia da semana
dfCovid['dia_semana'] = dfCovid['fecha_hora'].dt.day_name()
```

✓ 0.0s

```python
import re

# Lista que almacenará os hashtags
all_hashtags = []
# Recorremos os tweets extraendo os hashtags, si non hai hashtags engadimos 'No hashtags'
for tweet in dfCovid['tweet']:
    tweet_str = str(tweet)  # Convert to string explicitly
    tweet_hashtags = re.findall(r"#(\w+)", tweet_str)
    if tweet_hashtags:
        all_hashtags.append(tweet_hashtags)
    else:
        all_hashtags.append(['No hashtags'])

# Añadimos a columna 'hashtags' ao DataFrame
dfCovid['hashtags'] = all_hashtags
```

Figura 4.7: Proceso de creación de novas columnas

4.4 Análise exploratoria de variables: exploración estatística e visual

Según a natureza das variables, estas poden ser cuantitativas ou cualitativas. A forma de exploración varía en función desta distinción, utilizando técnicas estatísticas e gráficas adecuadas para cada tipo de variable.

Para as variables cuantitativas levarase a cabo unha exploración mediante:

- Estatísticas descritivas: Calcula medidas como a media, a mediana e a desviación estándar para resumir a distribución dos datos.
- Histogramas: Representa gráficamente a distribución de frecuencias dos valores.
- Diagramas de dispersión: Amosa a relación entre dúas variables cuantitativas.
- Boxplots: Proporciona información sobre a mediana, curtosis e posibles valores atípicos.

Pola súa parte, para as variables cualitativas empregaranse:

- Frecuencias e porcentaxes: Calcula a frecuencia e o porcentaxe de cada categoría.
- Gráficos de barras: Representa graficamente as frecuencias das categorías nun gráfico de barras.
- Gráficos circulares (Pie Charts): Amosa as proporcións relativas de cada categoría.
- Táboas de continxencia: Explora a relación entre dúas variables categóricas.

4.4.1 Variables cuantitativas

Miramos as estatísticas descriptivas para as tres variables cuantitativas do conxunto de datos, que son "me gusta", "retweets" e "citas". Amósase o resultado na figura 4.8, onde se poden observar patróns e comportamentos semellantes que achegan luz sobre cómo os usuarios interactúan co contido na plataforma. Estas tres métricas representan a interacción e o compromiso dos usuarios cos chíos e ofrecen información valiosa sobre a popularidade e a relevancia do contido.

	me_gusta	retweets	citas
count	8856.000000	8856.000000	8856.000000
mean	4.717254	0.964544	0.140244
std	50.888539	10.528072	2.426464
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	2.000000	0.000000	0.000000
max	3505.000000	362.000000	113.000000

Figura 4.8: Estatísticas variables cuantitativas

En primeiro lugar, todas as variables presentan distribucións asimétricas cara á dereita, onde a maioría dos chíos teñen valores baixos. Isto suxire que a maior parte do contido na plataforma non alcanza niveis significativos de interacción por parte dos usuarios. A mediana en todas as variables é 0, indicando que a metade dos chíos ten 0 en cada unha destas métricas.

Ademais, as desviacións estándar son relativamente altas, especialmente en "me gusta". Isto indica que hai chíos que reciben un número considerablemente alto de interaccións, o que pode deberse a contido viral, relevante ou altamente atractivo para a audiencia.

Aínda que as medias son valores relativamente baixos, é importante notar que estes valores non reflicten a totalidade do conxunto de datos. Os chíos con números excepcionalmente altos destas métricas poden influír na media, pero son relativamente raros en comparación coa maioría dos chíos que reciben interaccións mínimas.

	tweet_id	usuario	fecha_hora	tweet	me_gusta	retweets	citas	ciudad	estado	fecha	hora	fecha_id	dia_semana
5042	1241076828665597952	@johnnybananas	2020-03-20 17:32:07+00:00	Hey @espn next deal needs to be with @MTV and ...	3505	362	73	Fullerton	CA	2020-03-20	17:32:07	20200320	Friday
5042	1241076828665597952	@johnnybananas	2020-03-20 17:32:07+00:00	Hey @espn next deal needs to be with @MTV and ...	3505	362	73	Fullerton	CA	2020-03-20	17:32:07	20200320	Friday
2121	1251951014736883712	@Adam_Waltz	2020-04-19 19:09:16+00:00	Rally to re-open Arizona happening now with hu...	134	46	113	Phoenix	AZ	2020-04-19	19:09:16	20200419	Sunday
8078	1251594554341556224	@SamanthaJoRoth	2020-04-18 19:32:49+00:00	JUST IN — Florida Governor Ron DeSantis to Mak...	636	214	113	Washington	DC	2020-04-18	19:32:49	20200418	Saturday

(a) Información dos chíos máis relevantes

```
Hey @espn next deal needs to be with @MTV and show classic seasons of @challengeMTV citando a Scoop: ESPN, scrambling to fill its coronavirus-depleted scheudule,
has struck a deal to show classic WrestleMania events from the WWE on Sundays....

Rally to re-open Arizona happening now with hundreds of cars honking. @abc15 #Covid_19

JUST IN — Florida Governor Ron DeSantis to Make a Major Announcement Regarding COVID-19 in a 4:30 PM news conference.
```

(b) Visión dos chíos

Figura 4.9: Chíos con máis me gusta, retweets e citas

Procédese a explorar os chíos con valores máis altos nas tres métricas, ademais de estudar se se tratan dos mesmos chíos, que é o esperado. Pódese ver na figura 4.9, que o chío con máis "me gusta", tamén é o que ten máis "retweets" e que para o caso do chío con máis citados, hai un empate entre dous con 113 citados. Por cuestións de espazo, non se poden expoñer aquí todos os chíos con valores altos para estas métricas pero a maioría coinciden en que pertencen a chíos de usuarios verificados e persoaxes con profesións na televisión. No exemplo dos tres chíos que vemos, os que posúen os valores máximos das métricas, tratánse de textos escritos por usuarios verificados e, en dous casos, persoas que se dedican á televisión. Os chíos datan de marzo e abril e pertencen a estados distintos, estados que como veremos máis adiante, pertencen a estados con alta actividade en Twitter.

Como a maioría de valores son cero, se intentamos graficar os datos, non van a aportar valor, verase todo concentrado no cero. Por eso, non se utilizaron histogramas e os *boxplots* si se empregaron pero non son nada intuitivos. De feito, a caixa característica deste tipo de gráfico non se chega a ver pola disparidade entre os valores máximos das métricas e os valores "habituais" concentrados en cero (ver figura 4.10). Aínda que estes valores altos se consideran *outliers*, non se van eliminar, porque xa se estudou o motivo, son chíos que por ser escritos por usuarios verificados e con moito alcance de seguidores, conseguen máis interaccións.

O que si se puido estudar foi a correlación entre as variables, mediante gráficos de dispersión e un mapa de calor, onde se mostran as correlacións entre variables numéricas, como se ve na figura 4.11. Entre "me_gusta" e "retweets" si se pode ver unha relación lineal, se pasamos por alto os valores atípicos que se saen desta relación. Si observamos o mapa de calor, presentan unha correlación de 0.82, algo que tamén podemos intuír no gráfico de dispersión. Porén, para o resto de relacións, nos gráficos de dispersión non se visualiza ningunha tenden-

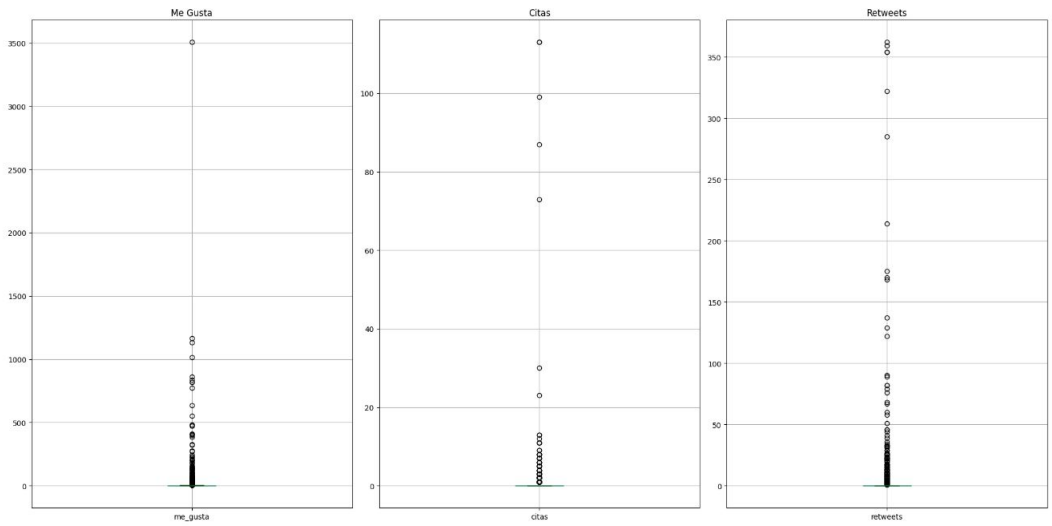


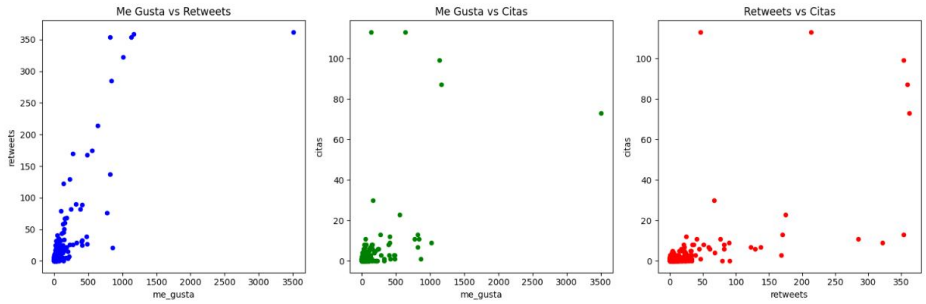
Figura 4.10: Boxplot das variables cuantitativas

cia e no mapa de calor, aínda que se amosan correlacións do 0.62 e do 0.60, estas poden vir xustificadas en gran parte pola enorme cantidad de chíos con respecto ao total que teñen todas as interaccións a cero. Isto é algo que tamén afecta á relación de "me_gusta" e "retweet" pero en menor medida, posto que si podemos ver certa tendencia lineal no gráfico de dispersión. É fundamental explorar diversas técnicas de visualización, por esto mesmo.

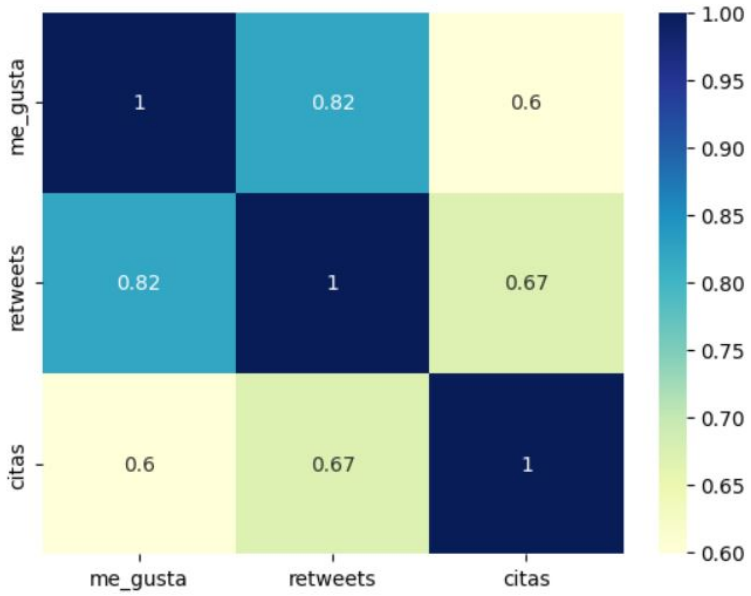
Con esto, remátase coa exploración entre variables cuantitativas aínda que, posteriormente, se volverán a explorar en conxunto coas cualitativas para estudar relacións entre unhas e outras.

4.4.2 Variables cualitativas

No comezo, é prudente examinar a cantidade de valores únicos presentes nas variables categóricas. Isto débese a que en certos casos as variables poden conter unha gran cantidade de categorías, o que podería dificultar a claridade do análise. Na figura 4.12, podemos ver que a maoiría das variables categóricas presentan unha variabilidade diversa na cantidade de valores únicos que posúen. Estes valores revelan detalles interesantes sobre as características dos datos. Por exemplo, a variable "fecha" presenta só 120 valores únicos, o que pode indicar que se trata dunha variable categórica con categorías limitadas e ben definidas. En contraste, as variables "tweet" e "usuario" posúen un gran número de valores únicos, o que pode indicar unha gran variabilidade nos textos dos chíos e nos usuarios que xeran os contidos. Ademais, a variable "dia_semana" ten 7 valores únicos, que corresponde aos días da semana, unha relación intuitiva e esperada. Chama bastante a atención a variable "estado", posto que posúe 102 valores diferentes, cando só debería haber ao sumo 50 estados, pois son a cantidade que



(a) Gráficos de dispersión entre cada par de variables



(b) Mapa de calor de correlación entre variables

Figura 4.11: Correlación entre variables cuantitativas

conforma Estados Unidos. Facendo unha pequena investigación, compróbase que non todos os estados teñen o mesmo formato, algúns están escritos con abreviaturas e outros co nome completo. Isto arránxase cun mapeo entre os valores da columna cos valores dun arquivo que conten en cada liña o nome completo do estado seguido da abreviatura, separados por unha coma. Este arquivo xerouse a partir dun artigo [61]. Algo importante que se notou no proceso foi a existencia do suposto estado de "DC". "DC" realmente é Washinton D.C., oficialmente Distrito de Columbia, e trátase da capital dos Estados Unidos. Está ubicada entre as fronteiras dos estados de Virginia e Maryland e non se comporta como estado, senón qué un distrito que depende directamente do goberno federal. Mantense, por tanto, no *dataset*, pero é importante que quede clara a distinción [62]. O resultado final foi que a cantidade de valores únicos da columna pasou a 52 (ver figura 4.13), o esperado, posto que unha posibilidade son os nulos que non se eliminaron, o resto son os 50 posibles estados e o distrito federal, polo que vemos tamén que hai, polo menos, un chío en cada estado. Procédese agora a ver a frecuencia das categorías de cada variable cualitativa para ver a súa presencia dentro do conxunto de datos. Vanse estudar as variables "fecha_id", "hora", "usuario", "ciudad", "estado", "hashtags" e "dia_semana" pero, por motivos de espazo, que ademais viuse que son variables con moitas categorías distintas, só se mostrarán aquí as análises de frecuencia para "estado", "hashtags" e "dia_semana".

```
Cantidade de valores únicos en 'tweet_id': 8855
Cantidade de valores únicos en 'usuario': 7867
Cantidade de valores únicos en 'tweet': 8809
Cantidade de valores únicos en 'ciudad': 2096
Cantidade de valores únicos en 'estado': 102
Cantidade de valores únicos en 'fecha': 120
Cantidade de valores únicos en 'hora': 8103
Cantidade de valores únicos en 'fecha_id': 120
Cantidade de valores únicos en 'dia_semana': 7
Cantidade de valores únicos en 'hashtags': 4127
```

Figura 4.12: Valores únicos para cada variable cualitativa

```
Valores únicos en 'estado':

 ['FL' nan 'TX' 'WI' 'IL' 'NY' 'CA' 'IN' 'MD' 'CO' 'PA' 'MI' 'MO' 'MN' 'VA'
 'NC' 'DC' 'AZ' 'OK' 'KY' 'GA' 'OR' 'LA' 'NM' 'WA' 'HI' 'UT' 'ME' 'SC'
 'MA' 'NJ' 'KS' 'MS' 'OH' 'TN' 'AK' 'NV' 'SD' 'AL' 'CT' 'VT' 'NH' 'DE'
 'WV' 'IA' 'NE' 'AR' 'RI' 'MT' 'WY' 'ID' 'ND']

Cantidad de valores únicos: 52
```

Figura 4.13: Valores únicos para "estado" tras o mapeo

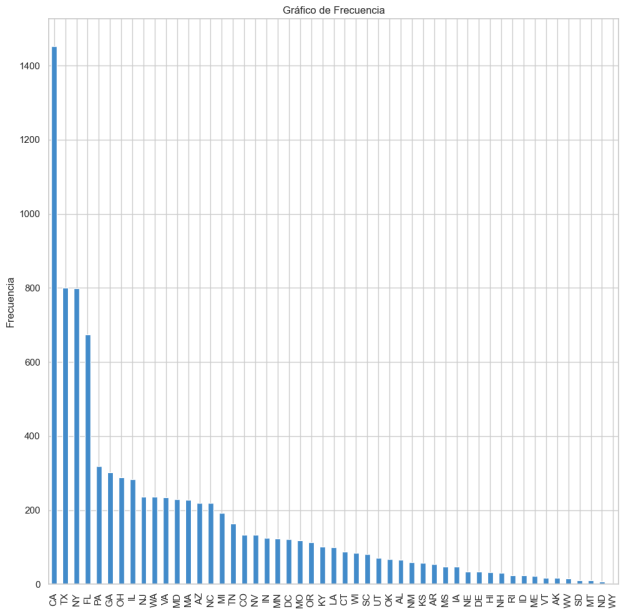
En primeiro lugar, na figura 4.14, podemos ver a análise das categorías de "estado". Para facelo, empregouse a función *value_counts()* de Python. Esta función permite contar a frecuencia de cada categoría na variable e obter así unha visión clara de cantas veces aparece cada unha. Ademais, calculouse a porcentaxe de cada categoría en relación co total de observacións, o que proporcionou unha comprensión máis completa da distribución proporcional das categorías. Esta información resultou esencial para identificar aquelas categorías que eran dominantes e aquelas que eran menos frecuentes no conxunto de datos, aínda que na imaxe 4.14a, só se mostran as 10 categorías máis frecuentes.

Para comunicar estes resultados de maneira efectiva, empregáronse gráficos de barras, como se ve na imaxe 4.14b, donde si se visualizaron todas as categorías. Neste tipo de grá-

ficos, cada categoría represéntase cunha barra vertical, sendo a altura da barra directamente proporcional á frecuencia ou ao porcentaxe da categoría correspondente. Esta representación gráfica simplifica a interpretación ao permitir unha comparación rápida e sinxela entre as diferentes categorías. Durante a exploración dos datos, detectouse que as catro categorías máis frecuentes abarcan o 41.43% do total, case a metade. Isto resulta relevante se recordamos que hai un total de 51 estados (incluído o Distrito de Columbia, que non é un estado).

	Frecuencia	Porcentaje (%)
CA	1413	16.07
NY	789	8.97
TX	776	8.82
FL	666	7.57
WA	355	4.04
PA	315	3.58
GA	299	3.40
OH	285	3.24
IL	278	3.16
VA	231	2.63

(a) Frecuencia por estado en número e en porcentaxe



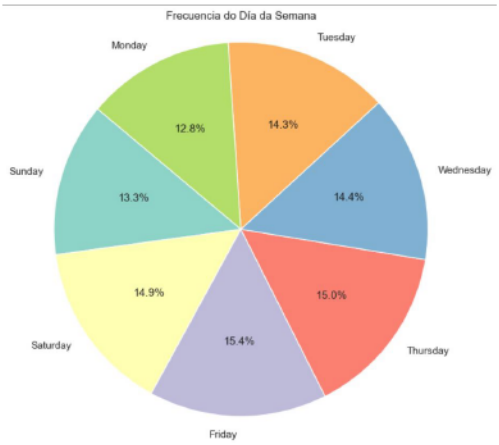
(b) Gráfico de barras de frecuencia de estados

Figura 4.14: Frecuencias e porcentaxes das categorías da columna estado

Seguindo coa análise, agora de "dia_semana", vemos na figura 4.15 a mesma visión estatística ca que se viu para "estado", aínda que desta vez, ao tratarse de poucas categorías, móstrase a táboa completa 4.15a. Para a parte gráfica, empregouse neste caso un gráfico de torta, que se ve na imaxe imaxe 4.15b. Nesta exploración, esperábamos ver unha distinción entre a frecuencia dos días máis notoria, esperábase máis concentración de chíos durante a fin de semana, mais están todos os días cunha porcentaxe bastante similar. O día con menor actividade é luns (12.8%) e o día con maior actividade venres (15.4%), coincidindo o primeiro co fin do descanso e inicio da semana laboral e o segundo todo o contrario, algo que pode ser un factor directo.

	Frecuencia	Porcentaje (%)
Monday	1133	12.793586
Tuesday	1264	14.272809
Wednesday	1277	14.419603
Thursday	1326	14.972900
Friday	1362	15.379404
Saturday	1317	14.871274
Sunday	1177	13.290425

(a) Frecuencia por días da semana en número e porcentaxe



(b) Gráfico de torta da frecuencia de cada día da semana

Figura 4.15: Distribución de frecuencias dos días da semana

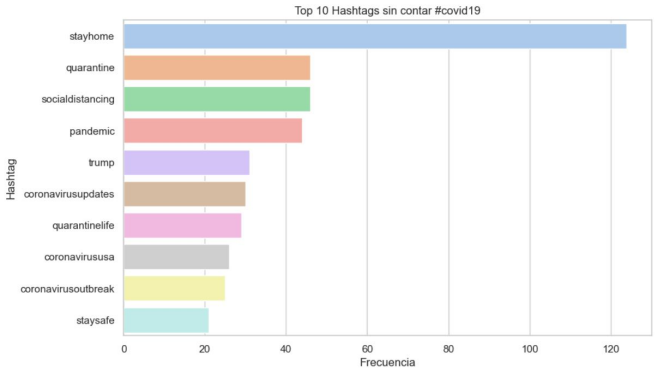
Por último, procédese co estudo da variable "hashtags", no que tamén houbo que facer cambios. Se se examinaba a variable tal cual, atopábanse moitos *hashtags* no top de frecuencias que viñan a querer expresar o mesmo, coma #Covid-19, #covid19, #coronavirus, etc. Realizouse un proceso para homoxeneizar os *hashtags*, de tal maneira que, por exemplo, neste caso pasaron todas as variantes a coñecerse como #covid19. Ademais disto, pasáronse todos a minúsculas. Vese un exemplo de código a continuación, aínda que se levaron a cabo máis cambios noutros *hashtags*, pero o procedemento foi o mesmo.

```
def homogenize_hashtag(hashtag):
        hashtag = hashtag.lower()
        hashtag = re.sub(r"\b(covid19|covid2019|covid_19|
        covid19outbreak|covid|coronavirus|covid19|corona)\b",
        "covid19", hashtag, flags=re.IGNORECASE)
        return hashtag
```

Unha vez se fixo a homoxeneización dos *hashtags*, procedeuse ao estudo das súas frecuencias. Na táboa da imaxe 4.16a amósanse os 10 *hashtags* con máis aparicións. Destaca que un 45.31% dos chíos non conteñen ningún *hashtag* e #covid19 ocupa a segunda posición, con un 14.5%. Entre os *hashtags* máis frecuentes tamén figuran #stayhome e #quarantine, que fan referencia ao illamento social, xunto con outros como #socialdistancing e #pandemic, os cales tamén reflicten a natureza da situación. Algúns *hashtags* como #trump enfócanse nun contexto máis político. Para a representación gráfica, para a cal neste caso se utilizou a libraría Seaborn de Python, consideráronse os 10 *hashtags* máis frecuentes, excluíndo #covid19, así como os chíos sen *hashtags*, con fins de comparación (imaxe 4.16b).

Hashtag	Frecuencia	Porcentaje (%)
no hashtags	6311	45.31
covid19	2019	14.50
stayhome	124	0.89
quarantine	46	0.33
socialdistancing	46	0.33
pandemic	44	0.32
trump	31	0.22
coronavirusupdates	30	0.22
quarantinelife	29	0.21
coronavirususa	26	0.19

(a) Frecuencia dos *hashtags* en número e porcentaxe



(b) Gráfico de barras da frecuencia de cada *hashtag*

Figura 4.16: Distribución dos datos da columna *hashtags*

4.4.3 Representacións gráficas de interés

Neste apartado, mostraranse outras gráficas de interés que se realizaron durante a análise, algunhas de relacións entre variables. Aínda que, unha vez máis, por falta de espazo, non se poden ensinar todas, presentarase as máis representativas e relevantes.

En primeiro lugar, algo fundamental, estúdase a evolución da cantidade de actividade, de chíos ao longo do tempo, que se pode ver na figura 4.17. O que destaca na representación é un pico na segunda semana de marzo, que coincide no día 13, cando o presidente Trump anunciou o estado de alerta [63]. De maneira xeral, vemos que según a actividade en Twitter, a pandemia comezou a facerse notar nas redes a partir de marzo, cun pico a finais de xaneiro, que coincide coa aparición do primeiro caso de coronavirus en Estados Unidos [64]. Polo xeral, no resto das semanas, aínda que hai picos, mantense bastante estable a cantidade, notando certo aumento a mediados de abril, despois de pasado un mes da declaración do estado de alerta, e certo descenso a partir de entón.

Outra representación interesante para ver evolución é a dos *hashtags*. Se estudamos a evolución dos *hashtags* no tempo, podemos investigar cando e como comezou a gañar popularidade un determinado *hashtag*, se houbo momentos de maior uso, e se a súa relevancia diminuíu ou se mantivo constante ao longo do tempo. Para isto, debido a gran cantidade de valores, representaranse unicamente os máis empregados. Por un lado, na imaxe 4.18a, visualízase o gran protagonista, #covid19. Pódense distinguir máis ou menos catro etapas: dende comezos de ano ata finais de marzo, o seu uso foi en aumento; a partir de entón, durante a última semana de marzo e a primeira semana de abril, baixou para volver a subir ata finais de mes, donde volve a caer en descenso. Podemos ver un comportamento bastante similar ao da evolución da cantidade de chíos.

Por outro lado, se se mira a figura 4.18, pódense observar os seguintes *hashtags* máis utilizados, onde se poden ver outros comportamentos. Para empezar, estos, aínda que foron

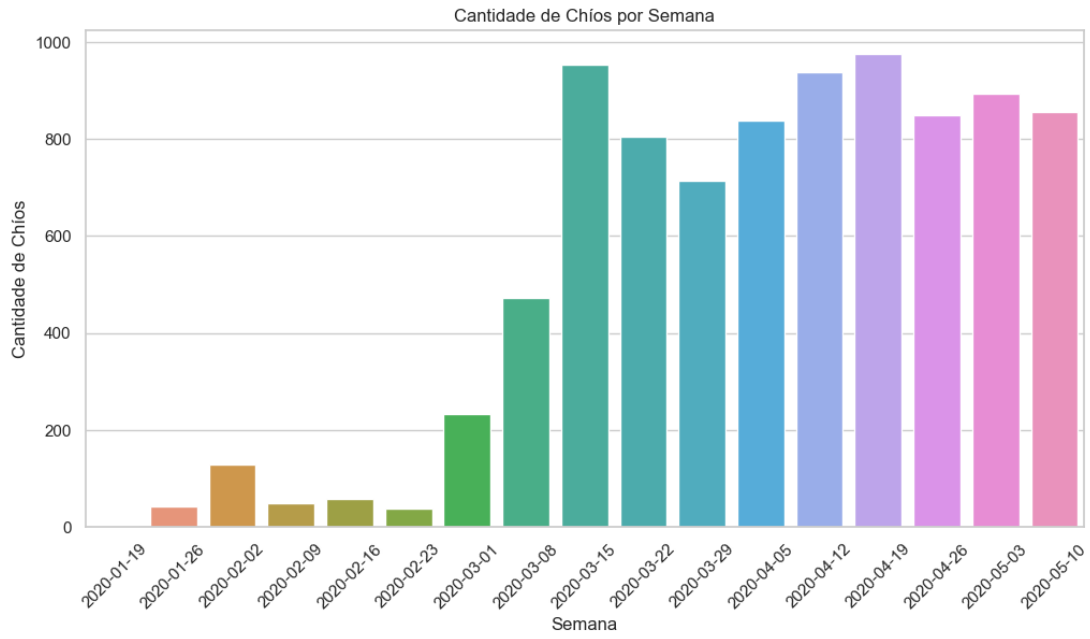


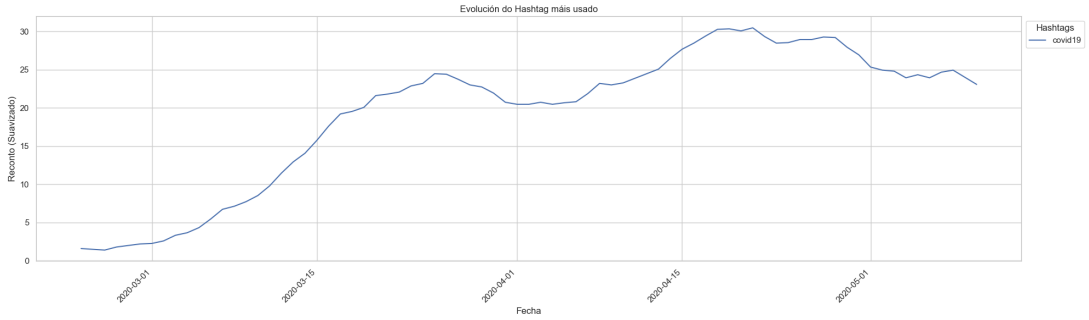
Figura 4.17: Evolución dos chíos ao longo do tempo

relevantes, comezaron a xurdir a partir de abril, durante as primeiras tres semanas según cal se mire. Son, na maioría, sobre a nova realidade nese tempo, polo que se entende que aparecerán nesas datas. Todos se manteñen bastante constantes, destacan #stayhome, que segue unha curva descendente ao largo de todo o tempo e #trump, co que pasa todo o contrario. Este último pode ser atribuída a varios factores, incluíndo a natureza polarizante e controvertida da figura de Donald Trump, a súa participación en eventos políticos clave como as eleccións presidenciais dos Estados Unidos e a extensa cobertura mediática que xeraba.

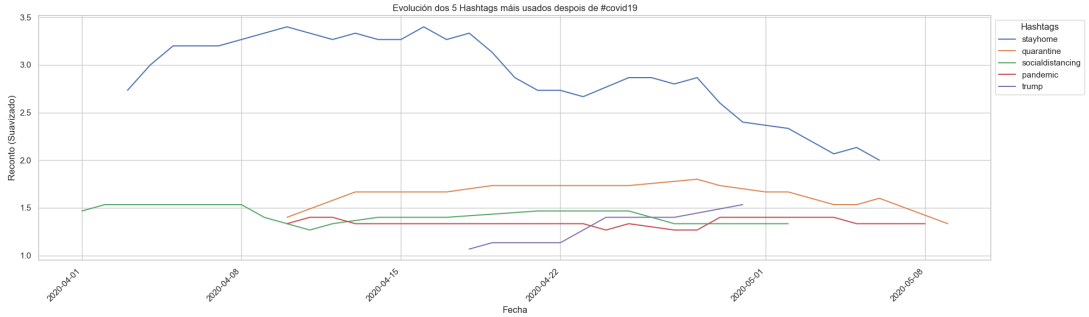
Outras figuras de interés como a distribución dos chíos según día da semana e hora ou a evolución que seguiron según o estado, atópanse no Apartado A.

4.5 Análise de sentimentos

No contexto da pandemia da COVID-19, a análise de sentimentos converteuse nunha ferramenta esencial para comprender a percepción pública arredor desta crise global. Unha análise de sentimentos é unha técnica no campo do procesamento da linguaxe natural que ten como obxectivo determinar a polaridade emocional dun texto, isto é, se o texto expresa sentimentos positivos, negativos ou neutros. Esta análise realízase utilizando algoritmos e modelos de aprendizaxe automática que son capaces de clasificar automaticamente o ton emocional dun fragmento de texto, xa sexa unha frase, un parágrafo ou un documento completo. O presente análise tivo como obxectivo examinar a actitude xeral dos usuarios de Twitter cara



(a) Evolución dos hashtag Covid19



(b) Evolución de hashtags relevantes

Figura 4.18: Evolución dos hashtags ao longo do tempo

á pandemia, identificando tendencias emocionais e patróns clave nun conxunto de datos de chíos relacionados co COVID-19.

4.5.1 Fases da análise

Para levar a cabo esta análise hai que comprender tres etapas diferenciadas:

1. **Tokenización e procesamento de texto**: A etapa de tokenización implicou dividir o texto en palabras individuais ou tokens significativos. Este proceso permitiu unha representación axeitada do contido textual para análise posterior. Ademais, aplicáronse técnicas de procesamento de texto, como a eliminación de *stopwords* (palabras comúns sen un significado contextual) e a lematización (redución de palabras á súa forma base), para simplificar o texto e mellorar a calidade da análise.

2. **Análise de sentimentos**: A fase central da análise consistiu en determinar o sentimento asociado con cada chío. Utilizáronse modelos de análise de sentimentos previamente entrenados que clasifican o texto en categorías emocionais, como positivo, negativo ou neutro. Estes modelos baséanse en técnicas de procesamento da linguaxe natural e aprendizaxe automática.

3. **Visualización e extracción de ideas**: Unha vez que se avaliaron os sentimentos dos chíos, procedeuse á visualización de resultados. Creáronse gráficos e visualizacións que amosaban a distribución de sentimentos ao longo do tempo ou en relación con eventos específicos relacionados coa pandemia. Isto permitiu identificar picos emocionais e tendencias ao longo do período analizado.

4.5.2 Ferramentas empregadas

Aínda que, como se comentou, xeralmente utilízanse modelos de aprendizaxe automática para levar a cabo un análise de sentimentos, no contexto deste traballo decidíronse empregar tres bibliotecas de análise de sentimentos previamente preentrenadas: SIA[30], VaderSentiment [29] e TextBlob [31]. Elixíronse debido á súa facilidade de uso e á capacidade de proporcionar resultados rápidos e precisos sen a necesidade de entrenamento propio. O feito de facer o mesmo análise con tres librarías distintas foi para comparar os resultados e elixir a mellor opción posible. Estas bibliotecas xa foron adestradas en grandes volumes de datos para recoñecer patróns e características asociadas aos sentimentos positivos, negativos e neutros nos textos. Isto permite obter resultados de análise de sentimentos sen ter que pasar polo proceso de adestramento de modelos desde cero. Cada unha destas bibliotecas ten a súa propia abordaxe para calcular os sentimentos, permitindo obter unha comprensión completa das opinións e actitudes expresadas nos chíos relacionados co COVID-19. Aínda que estas bibliotecas previamente entrenadas ofrecen unha vantaxe en canto á súa rapidez e facilidade de uso, tamén teñen as súas limitacións que hai que ter en conta:

- **Contexto específico**: Estas bibliotecas non están especificamente adestradas para o noso dominio particular, que é a percepción dos usuarios sobre a pandemia de COVID-19. Isto pode levar a resultados menos precisos, xa que algúns termos ou frases utilizadas en relación coa pandemia poden ter unha connotación diferente ao que estas bibliotecas esperan.

- **Ambigüidade lingüística**: As bibliotecas poden ter dificultades en capturar subtextos, sarcasmo ou ironía, o que pode levar a interpretacións incorrectas dos sentimentos expresados nos chíos.

- **Diversidade na entonación**: As expresións emocionais son complexas e poden variar amplamente. Estas bibliotecas adoitan clasificar os sentimentos en categorías básicas (positivo, negativo, neutro), pero hai unha ampla variedade de tonos emocionais intermedios que poden ser máis complexos de capturar.

Para abordar estas limitacións, tívose en conta:

- **Revisión manual**: Realizar revisións manuais de subconxuntos de chíos para validar os resultados e identificar posibles erros ou inexactitudes na clasificación de sentimentos.
- **Integración de coñecemento**: Complementar a análise de sentimentos co coñecemento experto sobre a pandemia e os temas relacionados, para interpretar con precisión os resultados no contexto adecuado.
- **Ensemble de modelos**: Combina-la saída de varias bibliotecas ou modelos para axudar a obter unha visión máis completa dos sentimentos expresados nos chíos.

Sabendo esto, coméntanse brevemente as tres ferramentas utilizadas: SentimentIntensityAnalyzer (SIA) de NLTK [30], VaderSentimentIntensityAnalyzer [29] e TextBlob [31]. Cada unha ofrece un enfoque único para avaliar os sentimentos.

SentimentIntensityAnalyzer (SIA) de NLTK

Esta valiosa ferramenta é unha parte esencial do Natural Language Toolkit (NLTK). O SentimentIntensityAnalyzer aproveita a técnica VADER (Valence Aware Dictionary and sEntiment Reasoner) para determinar a polaridade emocional dun texto combinando un dicionario léxico de palabras con polaridades asignadas e regras sintácticas intelixentes para analizar o contexto emocional das palabras. Esta técnica é especialmente útil na análise de sentimentos en chíos sobre a COVID-19 debido á súa capacidade para manexar textos curtos e expresivos.

VaderSentimentIntensityAnalyzer

VADER é tan impactante que tamén ten a súa propia implementación, chamada VaderSentimentIntensityAnalyzer. Esta ferramenta, similar ao SIA de NLTK, céntrase en proporcionar resultados precisos para textos sociais e curtos, como os que tratamos (chíos). A forza de VaderSentimentIntensityAnalyzer radica na súa capacidade para detectar o impacto emocional incluso en contextos informais, como emoticonas, sinais de exclamación e xerga. Iso faino ideal para capturar as emocións sutís e a actitude xeral nos chíos que xiran ao redor da pandemia.

TextBlob

Utilizando un enfoque basado no clasificador Naive Bayes, TextBlob simplifica a análise de sentimentos ao máximo. Con só unhas poucas liñas de código, pódese pasar un chío a TextBlob e obter a polaridade e subxectividade estimadas do texto. Aínda que non é tan especializado como VADER, é unha boa opción para unha análise rápida de sentimentos.

4.5.3 Proceso de análise en Python

Unha vez se ten certo contexto en canto a etapas do proceso e ferramentas utilizadas, procédese a amosar funcións claves empregadas e imaxes relevantes dos resultados en Python.

En primeiro lugar, móstrase o seguinte código con parte da función empregada para o preprocesamento dos chíos, que é unha peza clave para a análise. Isto debeuse a que as bibliotecas utilizadas están preparadas para recibir textos nun determinado formato, no cal lles é máis sinxelo centrarse nas palabras importantes para prever o sentimento. Como se pode ver, elimínanse palabras e caracteres non desexados, así como a puntuación (aínda que hai bibliotecas que saben traballar baixo textos puntuados). Ademais, pásase todo o texto a minúsculas e tokenizase, para pasar o texto a palabras, coñecidas como *tokens*. Isto é útil para simplificar a tarefa de análise e destacar as partes esenciais do texto.

```
1   def preprocesamento_tweets(df):
2   # Crear unha lista para almacenar os chíos preprocesados
3   tweets_preprocesados = []
4
5   for i, tweet in enumerate(df["tweet"]):
6       # Eliminar caracteres non desexados, mencións e hashtags
7       tweet = `` ``.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z
    \t])|(\w+:\/\/\S+)|(#[A-Za-z0-9]+)", `` ``, tweet).split())
8
9       # Convertir a minúsculas e tokenizar
10      tweet = nltk.word_tokenize(tweet.lower())
11
12      # Filtrar palabras vacías (stopwords)
13      palabras_vacias = set(stopwords.words("english"))
14      tweet = [palabra for palabra in tweet if palabra not in
    palabras_vacias]
```

Móstrase na figura 4.19, un dos chíos do conxunto de datos, antes e despois do proprocesamento. Neste exemplo, é mostrado o proceso de preprocesamento de texto nun chío. O chío orixinal, con palabras como “Good morning Friday it”s a beautiful day keeping the ones that”s affected by this Coronavirus uplifted in prayers and those that”s not please stay safe”, transformase nunha versión procesada: “good morning Friday beautiful day keeping ones affected Coronavirus uplifted prayers please stay safe”. Durante o preprocesamento, elimináronse palabras non esenciais, reducíronse palabras e caracteres, normalizouse o uso de minúsculas e realizouse a tokenización para dividir o texto en palabras individuais. Aínda que houbo cambios, a mensaxe central do chío mantívose intacta, resaltando a importancia de manterse seguro durante a pandemia e expresando bos desexos. Este proceso simplificado e centrado facilita a análise de sentimento ao destacar as palabras clave que inflúen na tonalidade xeral da mensaxe.

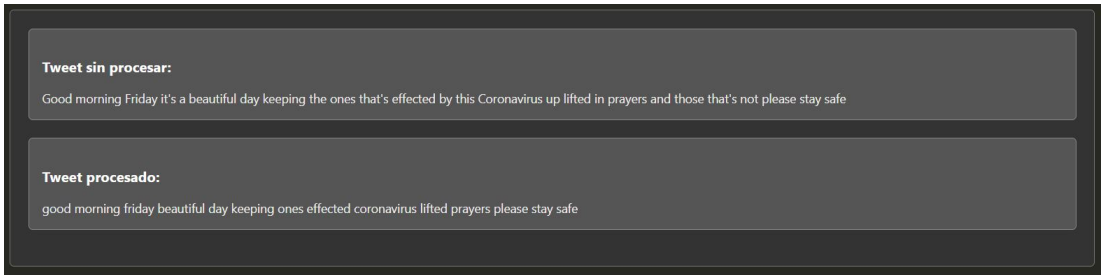


Figura 4.19: Chío orixinal e preprocesado

Antes de iniciar a análise, realizouse un estudo das palabras máis frecuentes nas conversas de Twitter. Esta valiosa información pode ser apreciada na imaxe 4.20, na cal se presenta de maneira visual unha nube de palabras que pon de manifesto as frecuencias relativas das palabras. Unha nube de palabras é unha ferramenta visual que permite identificar rapidamente as palabras que aparecen con maior frecuencia nun texto ou conxunto de textos, mostrándoas en función de cantas veces aparecen, o que significa que as palabras máis frecuentes serán máis grandes e destacadas. Esta visualización ofrece unha forma intuitiva de identificar os temas clave, conceptos e preocupacións que prevalecen no texto sen ter que lelo na súa totalidade. A través desta representación gráfica, é posible identificar as palabras que son máis comúns nas conversas en liña. Ademais, na imaxe 4.21 é posible observar en detalle as frecuencias das 10 palabras máis utilizadas. Estas palabras clave capturan unha ampla variedade de temas e preocupacións relacionadas coa pandemia. "Covid" e "pandemic" destacan na discusión central sobre a enfermidade e a súa propagación, mentres que "people" e "home" reflicten o impacto humano e as implicacións no entorno doméstico. A mención de "Trump" suxire debates en torno ás políticas e enfoques gobernamentais. "Like" e "time" indican a comparación de situacións pasadas e presentes. "Need" e "work" sinalan desafíos económicos e laborais, mentres que "today" pon de relevo eventos e novidades actuais. Xuntas, estas palabras ofrecen unha instantánea das conversas en liña sobre a pandemia e os seus múltiples aspectos.

Por último, con mor de indagar máis neste aspecto, tamén se visualizou un gráfico de frecuencias para bigramas e trigramas, mostrando só os 5 máis empregados. Como se ve na figura 4.22, os bigramas más comúns fan referencia ás situacións máis destacadas durante a pandemia como "quedarse na casa", "distanciamento social" ou "test positivo", e no caso de trigramas, a pesar de ser menos comúns e ser moitas veces expansión dos brigramas anteriores, téñense exemplos como "orden de quedarse na casa", "xente testada positivo" ou "practicar distanciamento social".

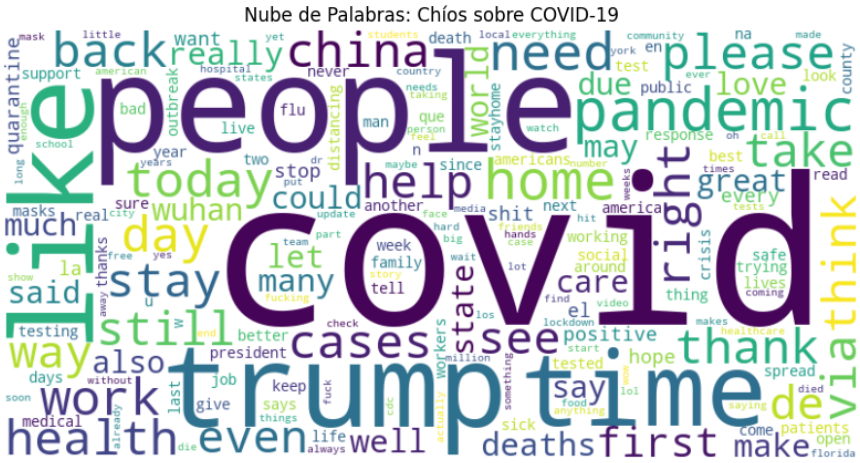


Figura 4.20: Nube de palabras

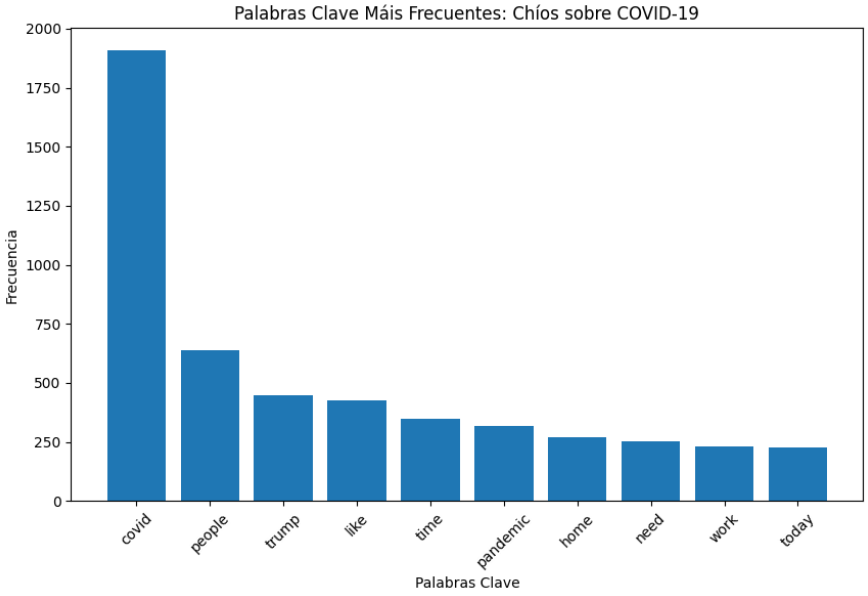


Figura 4.21: Frecuencias dos termos máis utilizados

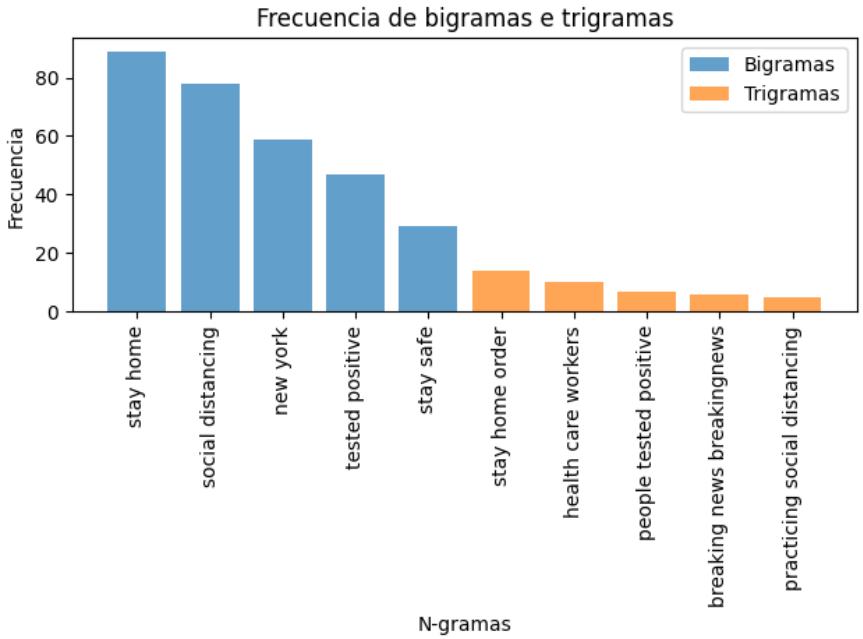


Figura 4.22: Frecuencias dos brigramas e trigramas máis utilizados

Despois de ter os chíos preprocesados, procedeuse á análise coas tres librarías.

Na táboa 4.3, represéntase a análise das tres novas columnas, que representan o valor *compound* asignado a cada chío. Segue a ser unha análise estatística de tres columnas cuantitativas novas. As cifras en cada fila representan diferentes estadísticas sobre as puntuacións de sentimento xeradas. A columna "mean" mostra o promedio das puntuacións. Estas son lixeiramente máis altas para TextBlob (0.058070) en comparación con NLTK e VaderSentiment (0.038960), o que podería indicar que tende a dar puntuacións máis positivas en xeral. Porén, se se mira o o percentil do 75%, vese que o valor de TextBlob é moi baixo (0.169028) en comparación con NLKT e VaderSentiment (0.401900 e 0.415300, respectivamente), algo que tamén se mide no Rango Intercuartil (IQR). Isto podería suxerir que asigna puntuacións relativamente baixas incluso a chíos que poderían ter un certo grao de positividade, posiblemente TextBlob tenda a etiquetar máis chíos como neutros ou podería ter dificultades para capturar niveis máis elevados de positividade en comparación con outras bibliotecas. A columna "std" representa a desviación estándar. Una desviación estándar alta indica maior variabilidade de puntuacións. Aquí, as tres bibliotecas teñen desviacións estándar similares. Pola súa parte, as columnas "min" e "max" mostran o valor mínimo e máximo das puntuacións. E vese que, polo xeral, as puntuacións de todas as ferramentas están nun rango similar e van dende moi negativas ata moi positivas.

No proceso de analizar o sentimento de cada chío, as bibliotecas NLKT e VaderSentiment, ofrecen uns resultados e TextBlob outros, aínda que todas coinciden nun valor. As primerias

Métricas	NLTK	VaderSentiment	TextBlob
Count	8856	8856	8856
Mean	0.038960	0.041315	0.041315
Std	0.460283	0.464984	0.464984
Min	-0.981700	-0.975200	-0.975200
25%	-0.273200	-0.273200	-0.273200
50%	0.000000	0.000000	0.000000
75%	0.401900	0.415300	0.415300
Max	0.985900	0.985400	0.985400

Táboa 4.3: Estadísticas de NLTK, VaderSentiment e TextBlob.

ofrecen valores de:

- Positividade (pos): Esta métrica representa a proporción de palabras no texto que se consideran positivas en termos de sentimento. Canto maior sexa este valor, máis positivo se considera o texto.
- Negatividade (neg): Similar á positividade, esta métrica representa a proporción de palabras que se consideran negativas en termos de sentimento. Un valor máis alto indica un sentimento máis negativo.
- Neutralidade (neutro): Representa a proporción de palabras neutras no texto. Un valor alto indica un texto máis neutral.
- Compound: Esta é unha métrica agregada que combina as puntuacións de positividade, negatividade e neutralidade nun só número. É especialmente útil para obter unha puntuación xeral do sentimento do texto. Un valor positivo indica sentimento positivo, un valor negativo indica sentimento negativo e un valor preto de cero suxire neutralidade.

Mentres tanto, TextBlob ofrece unha avaliación tamén en canto á subxectividade:

- Subxectividade: Mide o grao de subxectividade presente no texto, isto é, canto é unha expresión de opinións persoais en lugar de feitos obxectivos. A subxectividade mídese nun rango de 0 a 1, onde 0 é completamente obxectivo e 1 é completamente subxectivo.

Leváronse a cabo probas coa libraría TextBlob para probar o funcionamento da subxectividade, algo que so esta proporcionaba, pero ao ser textos con certa ironía ou moi emocionais

na maioría dos casos, observouse unha alta tendencia á subxectividade. O *compound* das primeiras e a polaridade de TextBlob, considéranse o mesmo, un valor entre -1 e 1 que nos indica o grao de positividade ou negatividade do texto, sendo este o valor que sacamos para a predicción do sentimento de cada chío, a partir de cada unha das bibliotecas. Cabe destacar, que se quixo dividir a polaridade en negativo, neutro e positivo e que se fixo baixo criterio propio, pois no rango de -1 a 1 tense claro que -1 é o máis negativo e 1 o máis positivo, pero dependendo dunha persoa ou outra o criterio para marcar un umbral que estableza a partir de que valor un chío é positivo, negativo ou neutro pode cambiar significativamente. No caso deste proxecto, comezouse establecendo o umbral nun 0.40 (-0.40, por tanto). Fixéronse varias probas con varios umbrales para ver o comportamento da distribución de sentimentos pero non se percibiu nada destacable, rematando por marcar un umbral de 0.30, como se ve no seguinte código.

```
# pasar compound a positivo, negativo, neutro cun umbral
umbral = 0.3
dfCovid["NLTK"] = dfCovid["NLTK"].apply(lambda x: "Positivo" if x >
    umbral else ("Negativo" if x < -umbral else ``Neutro"))
dfCovid["TextBlob"] = dfCovid["TextBlob"].apply(lambda x:
    ``Positivo" if x > umbral else ("Negativo" if x < -umbral else
    "Neutro"))
dfCovid["VaderSentiment"] = dfCovid["VaderSentiment"].apply(lambda
    x: "Positivo" if x > umbral else ("Negativo" if x < -umbral else
    "Neutro"))
```

O resultado foi para os tres casos, como se ve na figura 4.23, asignacións moi neutrales case na metada dos rexistros, incluso máis dun 75% sobre o total, no caso de TextBlob. Isto remata por confirmar as hipótesis feitas durante a análise estadística, TextBlob tende a dar puntuacións máis neutrales en comparacións coas outras bibliotecas. Algo destacable ademais é que nun contexto como o que nos incumbe, haxa maior porcentaxe de sentimentos positivos ca negativos, aínda que pode ser entendible no sentido de que os usuarios nas redes sociais, en xeral, tenden a mostrar moito máis as emocións positivas e soen ter maior impacto as negativas [65].

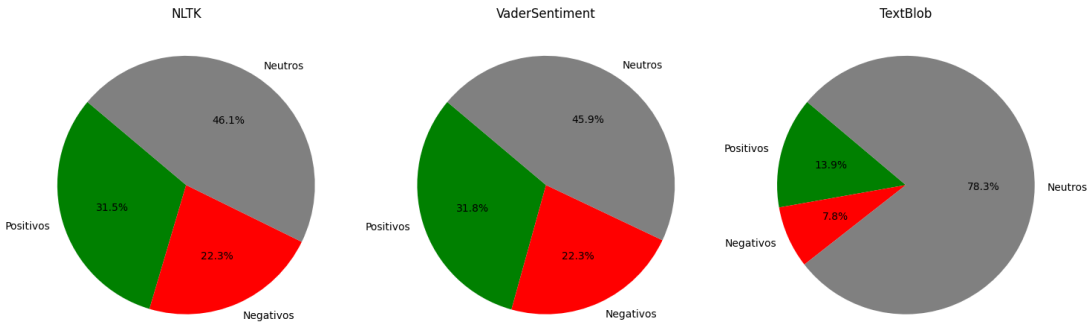


Figura 4.23: Porcentaxe sentimentos por libraría

Como se comentou nun principio, tanto VaderSentiment como SIA utilizan o enfoque de VADER (Valence Aware Dictionary and Sentiment Reasoner), o que significa que ambas se basan no mesmo léxico e enfoque para asignar puntuacións. As diferencias que poidan xurdir, probablemente se deban a variacións na implementación, na configuración e nas características adicionais que cada biblioteca incorpora. Xa se intuía algo disto nos resultados e confírmase a hipótese de que asignan *compounds* moi parecidos, mirando a matriz de correlación, que se construíu para as tres librerias e se mostra na táboa 4.4. Gráficamente, tamén se pode comprobar nos diagramas de dispersión da figura 4.24, na que se ve de xeito moi claro unha correlación positiva lineal entre NLTK e VaderSentiment. É importante ter esto en conta, á hora de estudar os resultados proporcionados por cada libraría, porque estas dúas bibliotecas realmente son como ter unha única.

Bibliotecas	**NLTK**	**VaderSentiment**	**TextBlob**
NLTK	1.000000	0.996694	0.525717
VaderSentiment	0.996694	1.000000	0.525423
TextBlob	0.525717	0.525423	1.000000

Táboa 4.4: Correlación entre NLTK, VaderSentiment e TextBlob.

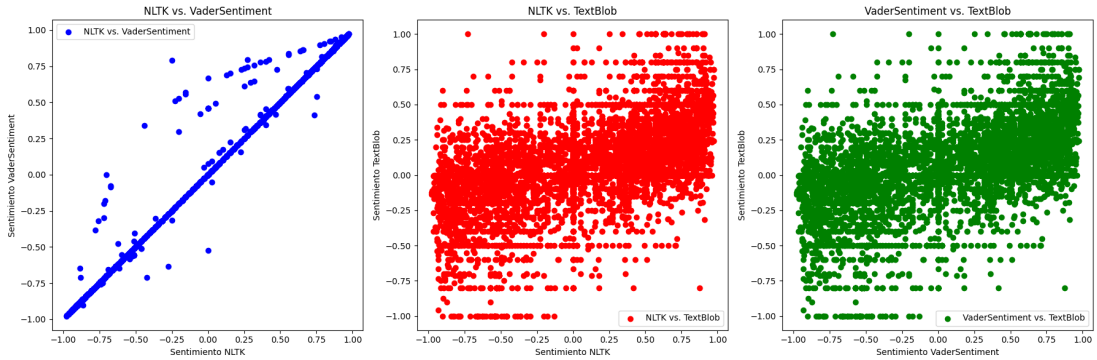


Figura 4.24: Gráficos de dispersión entre pares de librarías para a análise de sentimentos

Á hora de elixir que libraría era máis adecuada e proporcionaba mellores predicciónns para os chíos, fíxose unha proba á man. Esta consistiu nun etiquetado manual de 200 chíos, en base ao criterio propio, para ver con que libraría se coincidía máis. O proceso non foi sinxelo, porque un factor moi destacable é a subxectividade deste procedemento, como se vía xa en TextBlob. No entanto, chegouse a coincidir nun 39% dos casos con esta mesma libraría, 32% con VaderSentiment e 30% con NLTK. Como se pode ver, aínda que se concordou máis veces coa primeira, non é un valor moi notorio do total, é dicir, concordouse case tantas veces coas outras. Porén, a solución final para asignar un valor único de sentimento aos chíos, foi facer un promedio das tres puntuacións de *compound*, como se mostra no código a continuación:

```
1 dfCovid["avg_sent"] = ( (dfCovid["NLTK"] +
      dfCovid["VaderSentiment"])/2 + dfCovid["TextBlob"] )/2
```

Para rematar con esta análise, leváronse a cabo varias visualizacións de distinto tipo para explorar este novo valor en conxunto con outras variables do conxunto de datos. Comentaranse aquí dúas delas, que se mostrarán no Apéndice A, a figura A.5, onde se ve a evolución temporal de cada sentimento, e a figura A.6, onde se mostra a frecuencia de cada sentimento nos dez estados con máis actividade. Vese, que en todo momento e para todos os estados, predomina un sentimento neutral e, polo xeral, aínda que a veces se equiparan, o sentimento positivo tamén predomina sobre o negativo. Para os estados que se mostran, sempre se cumple esa observación. Cabe destacar que os momentos temporais nos que o sentimento negativo predomina lixeiramente sobre o positivo, ou se manteñen iguais, coinciden cos primeiros meses e en datas como a confirmación do primeiro positivo en Estados Unidos ou a declaración do estado de alarma.

Capítulo 5

Deseño e implementación da aplicación

Este capítulo céntrase en explicar aspectos relacionados co desenvolvemento da aplicación, proporcionando unha visión completa, dende os requisitos iniciais ata as probas finais de funcionamento, pasando polo deseño e a implementación das súas compoñentes clave.

5.1 Requisitos

Nesta sección, detallaranse os requisitos clave que guían o desenvolvemento e a execución do proxecto. Os requisitos son esenciais para comprender completamente o que se espera e garantir que todos os aspectos do proxecto se aborden de maneira efectiva. Os requisitos divídense en dúas categorías principais: requisitos funcionais e requisitos non funcionais.

5.1.1 Requisitos funcionais

Como se ve na táboa 5.1, mediante HU, nesta sección recóllense os requisitos funcionais da aplicación, que consisten nas funcións ou servizos que a aplicación web pon a disposición do usuario.

Táboa 5.1: Táboa de historias de usuario

ID	Nome	Descrición
HU-01	Selección de fecha	Os usuarios poderán escoller unha data específica utilizando un selector de data na páxina de inicio.

. . . . . *(continúa na páxina seguinte)* . . . . . .

Táboa 5.1 – *(vén da páxina anterior)*

ID	Nome	Descrición
HU-02	Selección de tipo de mapa	Os usuarios poderán seleccionar un tipo de mapa na páxina de inicio, que con un botón de "Enviar" serán dirixidos á pá páxina correspondente.
HU-03	Selección de estado	Os usuarios poderán escoller un estado utilizando un selector na páxina de "Mapa de Puntos" e "Mapa de Interaccións" e "Cadro de mando"
HU-04	Selección de cidade	Os usuarios poderán escoller unha cidade utilizando un selector na páxina de "Mapa de Puntos" e "Mapa de Interaccións".
HU-05	Selección de intervalo de tempo	Os usuarios poderán escoller unha data específica utilizando un selector de data na páxina de "Mapa de Puntos", "Mapa de Calor", "Mapa Coroplético" e "Mapa de Palabras" e "Cadro de mando"
HU-06	Selección de palabras/hashtags	Os usuarios poderán seleccionar palabras ou hashtags na páxina de "Mapa de Palabras".
HU-07	Mostrar marcadores	Amosaranse marcadores no mapa segundo os filtros de data, estado e cidade seleccionados na páxina de "Mapa de Puntos".
HU-08	Mostrar palabra/hashtag máis usado	Amosarase a palabra ou hashtag máis utilizado en cada estado, segundo os filtros seleccionados na páxina de "Mapa de Palabras".

. . . . . *(continúa na páxina seguinte)* . . . . . .

Táboa 5.1 – *(vén da páxina anterior)*

ID	Nome	Descrición
HU-09	Mostrar mapa de puntos	Xerarase un mapa de puntos, cada punto asóciase a un chío e represéntase cun marcador con información sobre el.
HU-10	Mostrar mapa de calor	Xerarase un mapa de calor baseado nos datos de ubicación e o intervalo de tempo elixido na páxina de "Mapa de Calor".
HU-11	Mostrar mapa coroplético	Xerarase un mapa coroplético en función do sentimento dos datos e o intervalo de tempo na páxina de "Mapa Coroplético".
HU-12	Mostrar mapa de palabras	Xerarase un mapa coas palabras máis frecuentes en cada estado.
HU-13	Mostrar mapa de hashtags	Xerarase un mapa cos hashtags máis frecuentes en cada estado.
HU-14	Mostrar marcadores de interaccións	Amosaranse marcadores con tamaños variables segundo o número de interaccións, baseado nos filtros seleccionados na páxina de "Mapa de Interaccións".
HU-15	Mostrar mapa de interaccións	Xerarase un mapa de puntos, cada punto asóciase a un chío e o tamaño do marcador é determinado según a cantidade de interaccións.
HU-16	Mostrar cadro de mando	Xerarase un cadro de mando con información de relevancia para interactuar co usuario.

. . . . . *(continúa na páxina seguinte)* . . . . . .

Táboa 5.1 – *(vén da páxina anterior)*

ID	Nome	Descrición
HU-17	Visualizar evolución dos chíos	Observar a evolución da cantidade de chíos ao longo dun periodo temporal.
HU-18	Mostrar cantidade de chíos	Coñecer a cantidade de chíos representados.
HU-19	Visualizar porcentaxe de chíos por cidade	Obter información sobre a cantidade de chíos para cada unha das cidades.
HU-20	Coñecer sentimentos	Observar por estado ou por cidade a cantidade de sentimentos de cada tipo.

5.1.2 Requisitos non funcionais

Os requisitos non funcionais definen as características ou propiedades que o sistema debe ter en canto a aspectos como a seguridade e o almacenamento, entre outros.

Os principais requisitos non funcionais do proxecto son os seguintes:

- As consultas realizadas na aplicación deben executarse de maneira eficiente, reducindo, na medida do posible, os tempos de espera nas execucións da plataforma. Isto é especialmente relevante dado que a aplicación se destina ao análise xeoespacial e temporal de chíos. A resposta rápida ás consultas permite unha análise áxil dos datos.
- Deseño dunha interface de usuario sinxela e intuitiva para a aplicación web. Isto pode axudar á comprensión por parte dos usuarios, facilitando a exploración e visualización dos datos de chíos sobre a COVID-19. A interface debe ser deseñada tendo en mente as necesidades específicas de análise xeoespacial e temporal, para que os usuarios poidan interaccionar coa información de forma eficaz. Os gráficos e mapas deben comunicar efectivamente as tendencias e patróns dos datos, permitindo aos usuarios comprender a información sen esforzo.
- Eficiencia de recursos e desempeño. A aplicación debe estar optimizada en termos de uso de recursos e desempeño, aínda que actualmente non se engadan datos adicionais. Isto asegura que a aplicación funcione de forma fluída e eficiente, permitindo un procesamento rápido e análise efectiva dos datos históricos de chíos sobre COVID-19.

5.2 Deseño

5.2.1 Arquitectura tecnolóxica do sistema

O sistema estrutúrase segundo un modelo de tres capas, caracterizado pola súa natureza secuencial e unha estrita xerarquía. Cada capa só posúe coñecemento da capa directamente inferior, o que establece un enfoque lineal na interacción. A comunicación entre estas capas prodúcese de forma asíncrona na súa maior parte, o que á súa vez fomenta unha maior escalabilidade do sistema. As tres compoñentes principais nas que se divide son: o Sistema de Xestión de Base de Datos con SQL (PostgreSQL e PostGIS), o Servizo Web (Flask, SQLAlchemy, GeoAlchemy e Python) e o Cliente Web (Leaflet, HTML, CSS, JavaScript, Requests, Flask e Python).

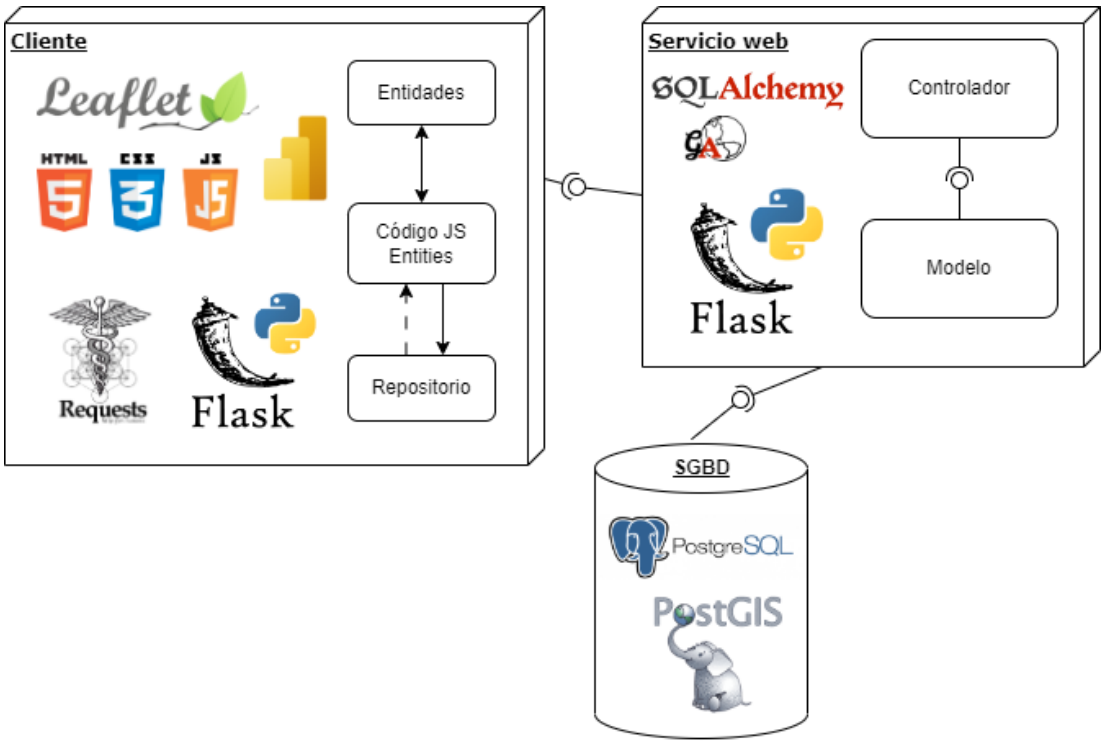


Figura 5.1: Arquitectura do sistema

Estes conceptos están ilustrados na figura 5.1 e explícanse máis detalladamente a continuación.

Sistema de Xestión de Base de Datos con SQL

A base da arquitectura alicerza sobre unha sólida capa de base de datos. Esta capa é responsable de administrar e almacenar todos os datos críticos dos chíos sobre COVID. Utiliza

unha combinación de PostgreSQL e PostGIS para xestionar tanto a información relacional como a xeoespacial; e proporciona un repositorio seguro e fiable onde os datos se almacenan e recuperan de forma eficiente. A súa conexión estreita co controlador garante que os datos se manexen de maneira coherente e segura ao longo do sistema.

Servidor Web (modelo e controlador)

O servidor web é o núcleo funcional do sistema e está construído utilizando o *framework* Flask xunto con SQLAlchemy e Python. Flask proporciona unha estrutura flexible para crear aplicacións web, mentres que SQLAlchemy facilita a interacción coa base de datos de forma eficiente, empregando tamén a súa extensión GeoAlchemy, para o manexo de datos xeoespaciais. Nesta capa, defínense dous compoñentes clave: o modelo e o controlador. O modelo define a estrutura dos datos e as relacións na base de datos asegurándose de que os datos estean dispoñibles e listos para ser procesados polo controlador, mentres que o controlador manexa a lóxica de negocio e orquestra as solicitudes do cliente. Utiliza un enfoque RESTful para comunicarse co repositorio no cliente web, accedendo aos datos solicitados de forma eficiente e mantendo a integridade do sistema.

Cliente Web (Repositorio, Entidades e Código JS de Entidades)

A capa de cliente web xoga un papel crucial na experiencia do usuario. Desglósase en tres compoñentes: o repositorio, as entidades e o código JS. O primeiro actúa como unha capa de abstracción, permitindo ao usuario final acceder e enviar datos ao sistema a través de solicitudes HTTP. Esta capa mantén unha comunicación constante co controlador no servizo web, a través dunha API REST, asegurando que os datos flúan. As entidades, pola súa parte, representan os modelos visuais (*templates*) que amosan a información ao usuario final, empregando HTML, CSS e JavaScript. Estas interfaces de usuario énchense con datos procedentes do modelo. O código JS encárgase da lóxica de presentación e engade interactividade ás entidades, permitindo que os datos se presenten e actualicen en tempo real, ofrecendo unha experiencia dinámica e atractiva para o usuario. Por exemplo, manexa a capacidade de filtrar ou actualizar datos sen recargar a páxina. Emprégase Leaflet para integrar os mapas interactivos e JavaScript para crear unha experiencia interactiva enriquecedora.

Pódese ver ver, que desde o almacenamento e xestión de datos na capa de base de datos, pasando pola lóxica de negocio e control na capa de servizo web, ata a visualización e interacción dinámica na capa de cliente web, cada capa xoga un papel vital no funcionamento integral do sistema.

5.2.2 Deseño da aplicación

Esta sección focalizarase en resaltar as ferramentas, arquivos e elementos que permiten a implementación de cada un dos compoñentes do modelo do sistema definido na sección anterior.

Compoñentes do Servidor - Flask e SQLAlchemy

- **models.py**: Este ficheiro xoga un papel central na parte do servidor. Aquí defínense as clases que representan as táboas da base de datos utilizando SQLAlchemy. Cada clase corresponde a unha entidade na base de datos, e os atributos da clase mapéanse ás columnas da táboa. Permite xestionar as relacións e operacións na base de datos de maneira eficiente.

- **views.py**: Na capa do servidor, especificamente na parte do controlador, atópase este arquivo. Neste ficheiro, defínense as funcións (vistas) que manexan as solicitudes entrantes. Estas interactúan co modelo definido en *models.py* utilizando SQLAlchemy para realizar operacións de lectura e escritura na base de datos. Logo, devolven respostas adecuadas ao cliente.

Compoñentes do Cliente - Plantillas HTML, JavaScript e CSS

- **Plantillas HTML (Entidades)**: As plantillas HTML actúan como a interface visual para o usuario. Estas plantillas atópanse nunha carpeta designada, chamada "templates'. Aquí defínese como se presenta a información ao usuario. Utilízase Jinja2, o motor de plantillas de Flask, para xerar contido dinámico nas plantillas. Estas plantillas incorporan código JavaScript e CSS.

- **Código JavaScript e CSS**: Dentro das plantillas HTML, inclúese código JavaScript para engadir interactividade e realizar solicitudes asíncronas ao servidor. O código JavaScript úsase para actualizar partes específicas da páxina sen recargala, proporcionando unha experiencia máis fluída e dinámica. Ademais, emprégase CSS para dar estilo e mellorar a aparencia visual das plantillas.

5.3 Modelo conceptual de datos

Nesta sección mostrarase o modelado conceptual no que se basou o proxecto. Un mapa conceptual radica na necesidade de transmitir información complexa de maneira comprensible e accesible. O noso sistema involucra múltiples capas, compoñentes e relacións, dende a adquisición e análise de datos até a visualización nunha aplicación web. Estes elementos

poden ser difíciles de entender na súa totalidade mediante descricións verbais ou mesmo diagramas individuais. O mapa proporciona unha vista panorámica e resumida da arquitectura xeral do sistema. Utilizouse un modelo entidade-relación, diagrama de fluxo que ilustra como as entidades, personas, obxectos ou conceptos, se relacionan entre si dentro dun sistema. O modelo represéntase na figura 5.2 e as entidades contempladas explícanse a continuación.

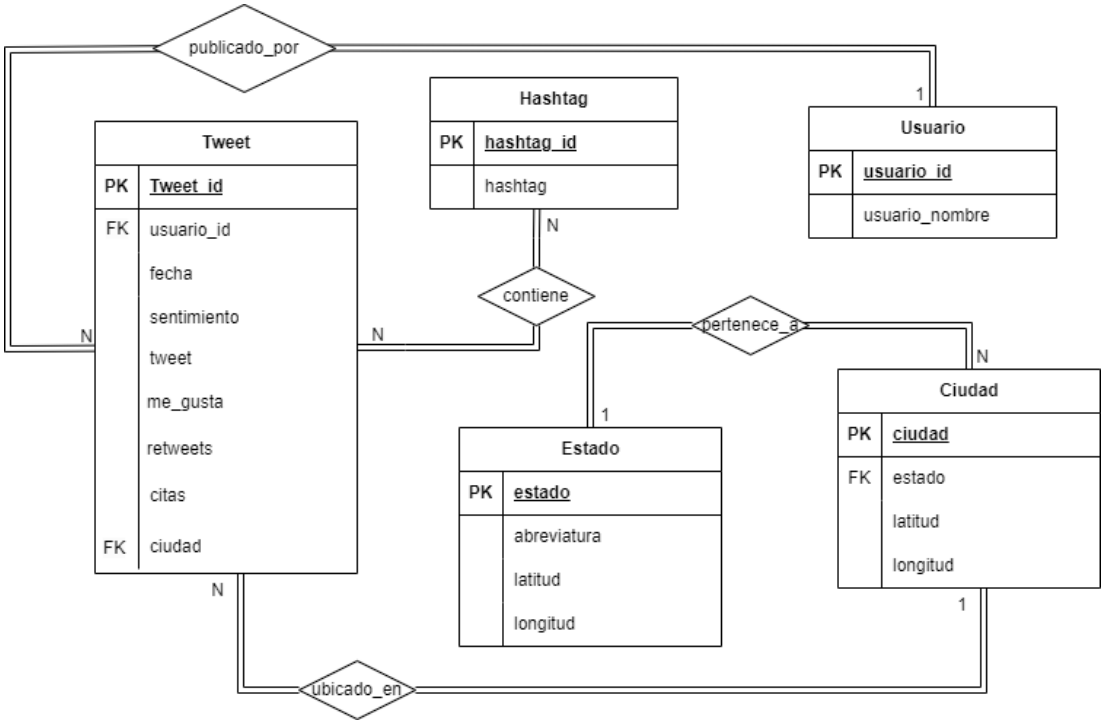


Figura 5.2: Modelo entidade-relación do proxecto

- **Tweet**. Contén información esencial sobre cada publicación realizada por un usuario. Cada chío está identificado polo seu "tweet_id" único e está composto por detalles como a data, o contido do chío, a cantidade de "me gusta", "retweets" e "citas". Os chíos son a base do análise e permiten entender as tendencias e a difusión de información en relación coa pandemia.
- **Hashtag**. Etiquetas ou palabras clave que se usan nos chíos para categorizalos e resaltar temas específicos. Cada *hashtag* vincúlase cos chíos relevantes. Aínda que non teñan unha identificación única, son esenciais para explorar os temas de discusión e avaliar a súa frecuencia e popularidade.
- **Usuario**. Entidade que representa aos individuos que publicaron chíos relacionados co tema da COVID. Cada usuario está caracterizado polo seu id de usuario, ademais de ter o atributo nome único.

- **Estado**. Representa as diferentes rexións xeográficas que forman parte do país. Cada estado ten un nome e unha abreviatura únicos. Esta entidade é crucial para vincular os chíos con información xeoespacial, permitindo visualizar e analizar como a conversa sobre COVID se distribúe en diferentes partes do país.

- **Ciudad**. Representa os lugares nos que se atopan os usuarios que publican os chíos. Cada cidade está identificada polo seu nome único. A información das cidades é importante para comprender as localizacións xeográficas dos usuarios e como se relacionan cos outros compoñentes do sistema.

5.4 API REST

Unha API é un conxunto de regras e protocolos que permiten a diferentes aplicacións comunicarse entre si. En esencia, actúa como un intermediario que permite que distintos sistemas se conecten e compartan datos e funcionalidades de maneira controlada e segura. As APIs son fundamentais para a construción de aplicacións, xa que facilitan a integración de servizos e a creación de solucións máis complexas a partir de compoñentes máis simples.

Os *endpoints* son URL específicos nunha API que se utilizan para acceder a diferentes recursos ou realizar accións. Cada *endpoint* representa unha funcionalidade ou recurso particular dentro da aplicación ou servizo proporcionado pola API. Estes *endpoints* permiten aos desenvolvedores enviar solicitudes (usualmente utilizando métodos HTTP como GET, POST, PUT ou DELETE) para interactuar coa API e obter a información necesaria ou realizar operacións específicas.

No caso do noso proxecto, móstranse na táboa 5.2, os *endpoints* que compoñen a páxina, así como unha breve descripción da acción que se realiza ao acceder a cada un, según o método empregado.

5.5 Interface de usuario

Previamente a proceder co desenvolvemento do visor web, levouse a cabo a elaboración de varios deseños que serviran de base para contemplar a organización e a presentación que se pretendía acadar. Estes deseños coñécense como *mockups*, trátanse de maquetas estátaticas sen funcionalidade real, serven para visualizar, comunicar e axustar o deseño antes de entrar na fase de programación, o que conduce a un desenvolvemento máis eficiente, aforro de tempo e un producto final que cumple na súa totalidade coas necesidades e expectativas dos usuarios. Por motivos de espazo, aínda que se levou a cabo a elaboración de 7 deseños, nesta sección só se mostrarán dous, podendo igualmente visualizar os restantes no apéndice B.

Endpoint	Método	Acción
/	GET	Acceso á páxina principal
/puntos	GET, POST	Acceso á páxina do mapa de puntos (GET). Obter datos filtrados de rango de fecha (POST).
/punto	POST	Selección dun punto.
/heatmap	GET, POST	Acceso á páxina do mapa de calor (GET). Obter datos filtrados de rango de fecha (POST).
/coropletico	GET, POST	Acceso á páxina do mapa coroplético (GET). Obter datos filtrados de rango de fecha (POST).
/palabras	GET, POST	Acceso á páxina do mapa de palabras/-hashtags (GET). Obter datos filtrados de rango de fecha (POST).
/interacciones	GET, POST	Acceso á páxina do mapa de interaccións (GET). Obter datos filtrados de rango de fecha (POST).
/cuadro	POST	Acceso á páxina do mapa do cadro de mando de Power BI.

Táboa 5.2: *Endpoints* das páxinas que compoñen o visor web

En primeiro lugar, ao acceder á web, móstrase unha páxina inicial que non posúe información en si. Isto é, a páxina principal, como se pode observar na figura 5.3, presenta un filtro de selector de datas, así como un selector de "tipo de mapa", xunto cun mapa básico dos Estados Unidos. A idea desta páxina é que o usuario poida decidir dende o primeiro momento o período temporal que desexa visualizar e a maneira na que quere visualizar a información. A imaxe que acompaña os selectores é puramente estética, sen función interactiva, servindo só para dar unha idea do mapa sobre o que a información será representada de distintas formas.

Ao resto das páxinas pódese acceder tanto dende un menú como dende esta páxina principal, coa vantaxe de que desta última forma podemos visualizar os datos filtrados por un periodo desde o comezo. De outro xeito, accedendo dende o menú, aparecerán representados todos os datos, aínda que a posibilidade dun selector que os filtre seguirá a existir. O seguinte deseño realizado foi o da páxina do mapa de puntos, como se pode ver na figura 5.4, que ten, ao igual que todas as páxinas da aplicación, un selector temporal. Ademais, pódense observar dous filtros: un selector de estado e un selector de cidade. A idea para esta páxina foi representar os chíos na súa correspondente localización, pintados en vermello, azul ou verde, segundo o sentimento etiquetado neles e, para mellorar a visualización, debido á destacable cantidade de puntos, empregáronse os selectores de localización para amosar só os puntos de interese, sen necesidade de visualizar todos. Para rematar, asignáronse marcadores a cada chío, que permitiran coñecer información relevante sobre estes.

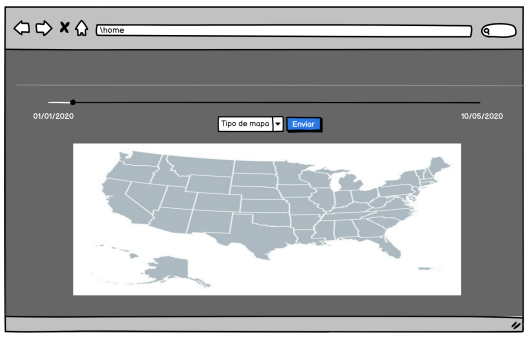


Figura 5.3: Mockup da páxina de inicio da aplicación

Figura 5.4: Mockup do mapa de puntos

5.6 Implementación

Nesta sección, coméntanse en detalle as partes máis complexas da implementación do sistema, como o proceso ETL e o funcionamento xeral das consultas.

5.6.1 ETL

Un traballo de proceso ETL é un conxunto de tarefas deseñadas para extraer, transformar e cargar datos dunha fonte a un destino nun sistema de información.

1. **Extracción (Extract)**: Nesta etapa, os datos obtéñense dunha ou varias fontes, que poden ser bases de datos, arquivos, servizos web ou outras fontes de información. A extracción implica recuperar os datos necesarios para a análise ou a aplicación específica.

2. **Transformación (Transform)**: Unha vez extraídos os datos, é común que necesiten ser limpos, reorganizados ou enriquecidos para ser útiles no destino final. As transformacións poden incluír a conversión de formatos, a eliminación de datos duplicados ou erróneos, o cálculo de novas métricas e calquera outra manipulación necesaria para que os datos sexan coherentes e significativos.

3. **Carga (Load)**: Nesta fase, os datos transformados córganse no destino final, que soe ser un almacén de datos, unha base de datos operativa ou un sistema de análise. A carga implica gardar os datos na estrutura adecuada do destino para que poidan ser consultados e utilizados segundo as necesidades do negocio.

A continuación explícase o proceso en dous bloques ben diferenciados: importación e transformación no software de analítica de datos e carga e axuste de datos no software de visualización de datos; e importación, transformación e carga no sistema xestor de bases de datos.

No primeiro caso, abórdase a importación da base de datos de viaxes en Python. Tal e como se mencionou na sección 4.2, xunto cunha explicación do proceso de limpeza realizado, decidiuse empregar a función “pd.read_csv” en VSCode.

No segundo caso, enfócase na carga de datos do modelo conceptual definido na sección 5.3 no sistema xestor de base de datos, PostgreSQL. A ferramenta empregada para crear a base de datos e operar con ela, foi DBeaver. O primeiro paso foi crear a conexión, para posteriormente crear a nova base de datos. Unha vez se testea a conexión, xa se pode ver baixo a sección das bases de datos. Pódese expandir para visionar as táboas, por agora ningunha. O seguinte paso é a creación do proxecto en Flask e a definición das entidades no arquivo ‘models.py’. A continuación, móstrase un exemplo.

```
mkdir proy_covid
cd proy_covid
python3 -m venv venv
source venv/bin/activate
pip install Flask Flask-SQLAlchemy Flask-Migrate geoalchemy2
```

```
class Usuario(db.Model):
    __tablename__ = 'usuario'
    usuario_id = db.Column(db.Integer, primary_key=True)
    usuario = db.Column(db.String)
```

Séguese por configurar en Flask a conexión a través do arquivo 'config.py', da seguinte maneira:

```
app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] =
    'postgresql://usuario:contraseña@localhost/nombre_basedatos'
db.init_app(app)
```

Unha vez, definidos os modelos e a conexión, realízase a propagación a PostgreSQL, utilizando o comando *flask db migrate -m "Creación de táboas"*. As migracións aseguran que a estrutura da base de datos se manteña sincronizada coas definicións dos modelos, o que é fundamental para o desenvolvemento e a evolución do proxecto. A medida que se realizaron cambios nos modelos, repetíronse estes pasos para manter a base de datos actualizada. Para cargar os datos nas táboas, utilizouse un script cun bucle *for* que recorre as liñas de archivos CSV, resultantes da limpeza en Python, para asignar os valores dos campos que se utilizan para crear un novo obxecto "Tweet" da clase.

Ao importar datos usando Flask e PostgreSQL, atopouse un problema recurrente co manexo dos tipos de datos. Durante a importación dos datos en PostgreSQL, atopáronse dificultades causadas por mor de que o formato dos datos non cumpría cos requisitos establecidos polo sistema de xestión da base de datos. Ademais, tamén se tiveron problemas para dividir as columnas nos conxuntos de datos importados.

Na táboa "ciudad", atopouse un reto relacionado cos datos almacenados nos campos de "latitude" e "lonxitude" dentro da base de datos. Estes campos conteñen valores numéricos decimais que proceden do ficheiro CSV. Con todo, o ficheiro utiliza comas para separar a parte enteira da parte decimal nestes valores.

Inicialmente, escolleuse o tipo de dato "DecimalField" para estes campos, que é o axeitado para manexar números decimais nunha base de datos. Mais, durante o proceso de importación dos datos, atopouse un erro que indicaba que había un número maior de columnas do que se esperaba. A razón detrás deste erro residía na forma en que os datos estaban a ser separados no ficheiro CSV. Neste caso, a coma utilizada como separador no ficheiro estaba a

ser interpretada pola base de datos como un novo delimitador de columna. Como resultado, os valores numéricos que contiñan comas na súa representación decimal eran divididos en máis columnas do previsto, o que xeraba o erro. Para superar este inconvinte, optouse por importar os datos destas columnas como texto en PostgreSQL. Logo, executouse unha consulta SQL que substituíu as comas por puntos nos valores dos campos de "latitude" e "lonxitude". Ademais, aplicouse unha conversión para cambiar o tipo de dato de texto a "double precision", que é máis axeitado para números decimais con alta precisión. Deste xeito, corrixiuse a interpretación errónea das comas como delimitadores de columna e adecuáronse os datos na base de datos de maneira exitosa.

```
UPDATE public."ciudad"
SET longitud = REPLACE(longitud, ',', '.')::double precision;
```

Por último, no caso da importación no software de visualización, tamén se realizou a través dun csv. Como se explicou na sección 3.2.2, traballouse con dúas ferramentas. Primeiramente, o desenvolvemento levouse a cabo en Microstrategy e posteriormente en Power BI, tras a negativa da empresa de utilizalo no desenvolvemento final. Para o primeiro, os datos cargáronse mediante un proceso ETL, que será comentado na seguinte sección. No caso de Power BI, simplemente se fixo clic na pestaña "Inicio" e seleccionouse a operación "Obtener Datos", elixindo a opción "Texto o CSV". Finalmente, abriuse unha pestaña que mostraba as táboas de datos importadas. Seleccionouse opción "Cargar datos" para completar o proceso de importación, a pesar de que dende esta pestaña se podían transformar os datos e seguir un proceso ETL, mais para este caso, como xa estaba feito previamente na outra ferramenta, os csv cargados xa foron os resultantes de dito proceso en Microstrategy.

Como ben se explicou, Microstrategy comprendiu un proceso ETL a parte. Isto debeuse que nesta ferramenta estaba configurada como único orixe de datos SQL Server. A elección desta ferramenta de visualización debeuse á súa constante utilización no ámbito laboral por parte da alumna. Solicitouse o seu uso á empresa, por iso o uso quedaba suxeito as condicións coas que estaba configurada. Aínda que de xeito similar ao realizado en PostgreSQL, para as táboas en SQL Server, xa se podían cargar os datos directamente limpos do CSV, levouse a cabo un proceso ETL cunha nova ferramenta, Servizo de Integración de SQL Server, plataforma que permite xerar solucións de integración de datos de alto rendemento, entre as que se inclúen paquetes de ETL para o almacenamento de datos. O proceso ou solución realizada comprendiu tres fases, que se corresponden coas tres etapas do proceso ETL e que se desenvolveu cada unha nun paquete distinto. Cabe previamente a explicar o concepto de solución, de paquetes, de conexións e de compoñentes.

Unha solución en SSIS é un conxunto organizado de paquetes, conexións, configuracións e outros elementos relacionados que forman un proxecto de integración de datos. Os paquetes son as unidades fundamentais de traballo en SSIS. Un paquete é unha colección de fluxos de

control, conexións, tarefas e transformacións que definen un proceso de integración de datos. Cada paquete pode conter unha serie de tarefas que se executan secuencialmente ou de forma paralela. As conexións son configuracións que permiten que os paquetes se conecten a fontes de datos ou destinos. Son esenciais para que os paquetes poidan obter datos, transformalos e cargalos en diferentes orixes ou destinos. Os tipos de conexións utilizadas neste proxecto son:

- **Conexións OLE DB**: Pertmite conectarte a orixes e destinos que admiten OLE DB (Object Linking and Embedding, Database). Utilizáronse para a conexión coas táboas de SQL Server creadas no proceso.
- **Conexións de ficheiro plano**: Usadas para acceder a ficheiros de texto plano para a extracción ou carga de datos. Utilizáronse no primeiro paso de extracción, para conectarse aos archivos CSV.

En canto ás compoñentes, son as pezas fundamentais que conforman os paquetes de integración de datos. Cada compoñente ten unha función específica e axuda a definir as accións que se deben realizar durante a extracción, transformación e carga dos datos. A continuación, explicaranse os compoñentes empregados:

- **Orixe de ficheiro plan**: Este compoñente permite extraer datos de ficheiros de texto plano, como CSV. Pódense configurar os detalles do ficheiro, como a ruta do ficheiro, o tipo de delimitador, o formato dos campos, etc. Emprega a conexión de ficheiro plano comentada anteriormente.
- **Orixe OLDB (OLE DB Source)**: O compoñente "OLE DB Source" permite a extracción de datos dende fontes de datos que admiten a tecnoloxía OLE DB, como bases de datos SQL Server. Pódese especificar a conexión OLE DB, a táboa ou a consulta da que se queren obter os datos.
- **Destino OLDB (OLE DB Destination)**: O compoñente "OLE DB Destination" permite cargar datos nunha fonte de datos que admita OLE DB, como unha base de datos SQL Server. Pódese definir a conexión OLE DB, a táboa na que se queren cargar os datos e mapear as columnas dos datos de entrada coas columnas da táboa de destino.
- **Sentenza SQL**: O compoñente "Sentenza SQL" permite executar unha sentenza SQL directamente contra unha base de datos ou outra fonte de datos.

O proceso levado a cabo foi o mesmo ca en Python, con consultas xa sobre as táboas, co último compoñente dos comentados. Por temas de espazo, todas as capturas relacionadas con este proceso, imaxes dos paquetes, das conexións e dalgunha consulta, mostraranse no Apartado C.

5.6.2 Implementación de consultas en Flask

Todas consultas lánzanse dende o *backend* cunha estructura semellante, consistente na creación da consulta, conexión co Sistema de Xestión de Base de Datos SQLAlchemy [12] e execución da petición. A continuación preséntanse dúas das consultas do proxecto:

- **Creación do motor:** O motor é unha parte esencial de SQLAlchemy e úsase para establecer e xestionar as conexións cunha base de datos. "Create_engine" é unha función que se utiliza para crear un obxecto de motor de base de datos. Recibe como argumento unha cadea de conexión que especifica onde e como conectarse á base de datos. Como se ve no código, a cadea de conexión neste caso é "postgresql://postgres:postgres@localhost:5432/postgres" Unha vez que se chama a "create_engine" coa cadea de conexión, crea un obxecto de motor que está listo para conectarse á base de datos PostgreSQL especificada cando se realiza algunha operación.

```
1        engine =
     create_engine("postgresql://postgres:postgres@localhost:5432
2        /postgres")
3
```

- **Obtención dos datos para o mapa coroplético:** A consulta está deseñada para obter datos relacionados coa data dun chío, o estado, a xeometría e o sentimento asociado ao tweet. Utilízanse alias para acurtar os nomes das táboas e constrúese a consulta utilizando as funcións "select", "select_from" e "group_by".

```
1    # Construír a consulta
2    consulta = select([
3        tweet.c.fecha, estado.c.estado, estado.c.geometry,
     tweet.c.Sentimiento,
4    ]).select_from(
5        join(
6            join(
7                tweet, ciudad, tweet.c.ciudad == ciudad.c.ciudad),
8            estado, ciudad.c.Estado == estado.c.Estado)
9    ).group_by(
10       tweet.c.fecha, estado.c.estado, estado.c.geometry,
     tweet.c.Sentimiento)
```

- **Obtención dos datos para mapa de *hashtags*:** Ten a misma estructura que a consulta anterior, só que recupera outros datos. Neste caso, para conseguir información dos hashtags máis frecuentes nos chíos, para un rango de datas. Aquí só se recupera a data, o chío, o hashtag e as coordenadas, o proceso de conseguir o *hashtag* máis común en cada periodo, lévase a cabo na plantilla a partir destes datos.

```
    # Construír a consulta
    consulta = select([
        tweet.c.fecha, estado.c.estado, estado.c.latitud,
    estado.c.longitud, hashtag.c.hashtag,
    ]).select_from(
        join(
            join(
                tweet, ciudad, tweet.c.ciudad == ciudad.c.ciudad),
            estado, ciudad.c.Estado == estado.c.Estado),
        hashtag, tweet."Tweet\_id" == hashtag."Tweet\_id")
    ).group_by(
        tweet.c.fecha, estado.c.estado, estado.c.latitud,
    estado.c.longitud, hashtag.c.hashtag
    )
```

- **Envío dos datos ás plantillas**: Para que as plantillas poidan recuperar os datos, empréganse o código seguinte, da mesma maneira para todas as consultas, adaptando os datos e sempre recuperando a fecha mínima e máxima da consulta para poder construír o selector temporal:

```
        # Execución da consulta
        result = engine.connect().execute(text(consulta))

        # Obter rango de fechas para o selector
        min_date = select([func.min(Tweet.c.fecha)]).scalar()
        max_date_query = select([func.max(Tweet.c.fecha)]).scalar()

        # Construcción da lista de marcadores para pasala á
    plantilla
        markers = []
        for row in result:
            markers.append({'fecha': row[0], 'estado': row[1],
    latitud': row[2], 'longitud': row[3], 'hashtag': row[4]})

        # As datas pásanse a srtftime
        min_date = min_date.strftime("%Y-%m-%d")
        max_date = max_date.strftime("%Y-%m-%d")
```

- Obtención dos datos e filtrado temporal na plantilla: Para todas as plantillas se segue o mesmo proceso, comentado a continuación brevemente.

  O código JavaScript que se mostra a continuación deste texto é utilizado para crear un mapa interactivo cun control desprazable de intervalo de datas. Os datos dos marcadores obtéñense en formato JSON desde o servidor e almacénanse na variable *markers*. A

continuación, créanse copias dos datos orixinais e establécense as datas mínima, máxima, inicial e final. As datas iniciais e finais son definidas polo usuario, por defecto o seu valor correspóndese co da data mínima e máxima. Utilízase a biblioteca Leaflet [20] para o mapa e o complemento ionRangeSlider [66] para o control desprazable. Cando o usuario cambia o valor do desprazable, chámase á función "updateMarkersFromSlider" para actualizar os marcadores no mapa segundo o novo intervalo de datas seleccionado.

```
1        var markers = {{ markers | tojson | safe }};
2        var filteredMarkers = markers.slice();
3        var minDate = new Date('{{ min_date }}'); var maxDate = new
    Date('{{ max_date }}');
4        var initial = new Date('{{ initial }}'); var end = new
    Date('{{ end }}');
5        var myMap = L.map('mapid').setView([37.09, -95.71], 4);
6
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png',
    {
7            attribution: 'Map data © <a
    href="https://openstreetmap.org">OpenStreetMap</a> contributors',
8            maxZoom: 18,
9        }).addTo(myMap);
10
11       $("#date-range-slider").ionRangeSlider({
12           type: 'double', grid: true,
13           min: minDate.getTime(), max: maxDate.getTime(),
14           from: initial.getTime(), to: end.getTime(),
15           onChange: function(data) {
16               updateMarkersFromSlider(data);
```

5.7 Probas

Nesta sección, amósanse as probas realizadas para para validar as implementacións anteriores do Apartado 5.6.

No desenvolvemento do proxecto, é fundamental realizar unha serie de probas exhaustivas para garantir a calidade e o funcionamento efectivo da aplicación. No caso deste proxecto, estas probas divídense en varias categorías para avaliar diferentes aspectos clave do proxecto.

En primeiro lugar, as probas de extracción e procesamento de datos foron esenciais. Verificouse a precisión e a exhaustividade da extracción de datos dos tweets relacionados coa COVID-19 nos Estados Unidos. Iso impolicou asegurarse de que todos os datos relevantes se recopilasen correctamente e a validación da limpeza e preprocesamento de datos para eliminar calquera dato irrelevante ou ruído, como se viu no Apartado 4.

Logo, as probas de análise de sentimento foron cruciais para avaliar a capacidade das bibliotecas de etiquetado de sentimentos seleccionadas para clasificar os mesmos nos chíos. Comprobouse a precisión desta clasificación, para cada unha, asegurando a correcta etiquetación dos sentimentos como positivos, negativos ou neutros. Para estas probas, levouse a cabo unha etiquetación manual, co fin de asegurarse de que os resultados do análise da aplicación coincidisen coas avaliacións humanas. Este proceso de etiquetación manual implicou a revisión e clasificación individual de cada un dos 200 chíos como "positivos", "negativos" ou "neutros" en base ao seu contido e tonalidade emocional. Estes chíos etiquetados manualmente foron posteriormente utilizados como un conxunto de datos de referencia para comparar coas saídas do análise de sentimento automatizado da aplicación e determinar a biblioteca máis afín.

No que respecta á xeración de mapas, leváronse a cabo unha serie de probas para os diferentes tipos de mapas que se querían amosar na páxina web. Por exemplo, para o caso do mapa coroplético xogouse co valor do umbral que determinaba intervalos de puntuación correspondían con cada sentimento. Para o mapa de puntos, realizouse un proceso incremental no que primeiro se intentou representar os marcadores no mapa sen máis. Logo, experimentouse cos filtros de estado e cidade para finalmente asignar unha cor ao marcador segundo o sentimento do chío.

Por último, para garantir que a aplicación sexa escalábel e de alto rendemento, realizáronse probas de rendemento e escalabilidade; e de interacción, como verificar a velocidade de carga da páxina, a capacidade de resposta dos gráficos e mapas en diferentes dispositivos, así como a facilidade de navegación nos mapas interactivos.

Ademais das probas comentadas, ao longo de todo o proceso, leváronse numerosas probas de aceptación cos titores, que foron validando o traballo e o correcto funcionamento da ferramenta.

Capítulo 6

Solución desenvolvida

Nesta sección, preséntase as páxinas que compoñen a solución desenvolvida. A idea da selección temporal é algo que se mantén para todas as páxinas, polo que, a partir de agora, sabendo que todas elas contan cun filtro deste tipo en forma de deslizador, tan só se comentarán os restantes filtros propios de cada páxina.

6.1 Páxina principal

A páxina principal destaca por ser o menú de acceso ao resto de páxinas, o lugar onde establecer os filtros para as visualizacións. Como se ve na figura 6.1 poderemos establecer o rango de datas apropiadas para a consulta e seleccionar o tipo de mapa a representar entre as opcións que se enuncian nas seguintes seccións.

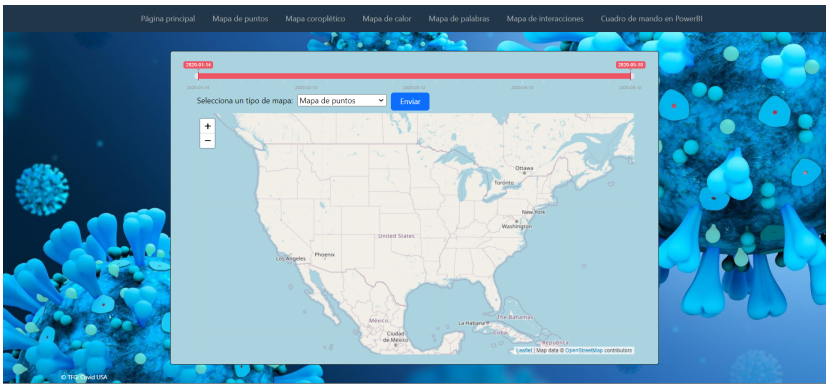
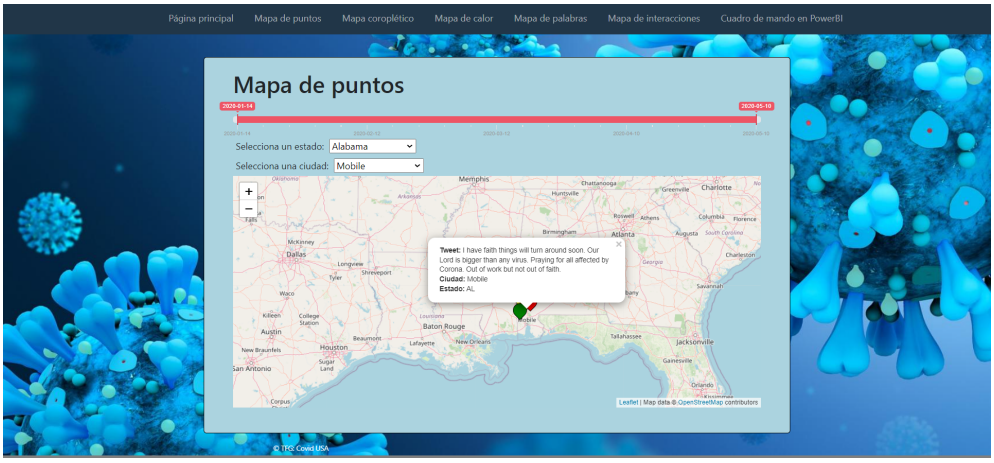


Figura 6.1: Páxina principal da aplicación web

6.2 Mapa de puntos

Esta visualización é a máis sinxela de todas, serve como punto de partida para as seguintes. Na figura 6.2a vemos como se proxectan os chíos en bruto nos mapas, e, ao poñer o rato sobre cada un dos puntos, como vemos na figura 6.2b podemos ver os detalles de cada chío, mostrando o contido deste, o estado e a cidade. Debido ás limitacións de Twitter, as coordenadas son obtidas a nivel de cidade, polo que para solucionar a superposición dos chíos da mesma cidade, aplícaselles un *delta* aleatorio para desplazalas e facilitar a visualización e o uso.

(a) Páxina do mapa de puntos



(b) Mapa de puntos filtrado por estado e cidade

Figura 6.2: Mapa de puntos

6.3 Mapa coroplético

Con isto, na seguinte páxina, represéntase un mapa coroplético, que é un mapa que permite mostrar de maneira sinxela a relación dunha área xeográfica cunha variable dos datos, dividindo estas e coloreándoas en función de dita variable. Como se ve no mapa da figura 6.3, emprégase o sentimento como variable, de xeito que, o usuario pode ver como varía este en función da rexión, sendo previamente filtrado por un rango de datas. O mockup correspondente a esta páxina atópase no Apartado C, nel podemos ver que inicialmente se plantexara utilizar 5 clases, pero, dados os resultados obtidos, non resultou sinxelo determinar os umbrais, polo que finalmente se fixeron únicamente 3 divisións, as cales se representan nunha lenda.

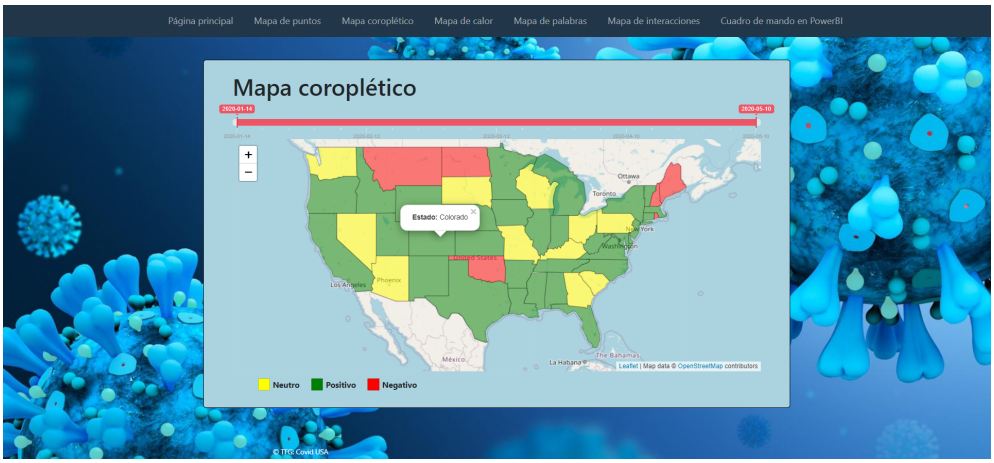


Figura 6.3: Mapa coroplético

6.4 Mapa de calor

Continuando coa idea do mapa de puntos e da visión temporal da evolución dos chíos sobre o mapa, desenvolveuse outra páxina que presenta a información como se ve na figura 6.4, cun mapa de calor no que se mide a densidade de chíos. Isto convérteo nunha visión máis atractiva e intuitiva para a mesma información, aínda que neste caso, debido ao tipo de mapa, non se accede á información en detalle de cada un dos chíos.

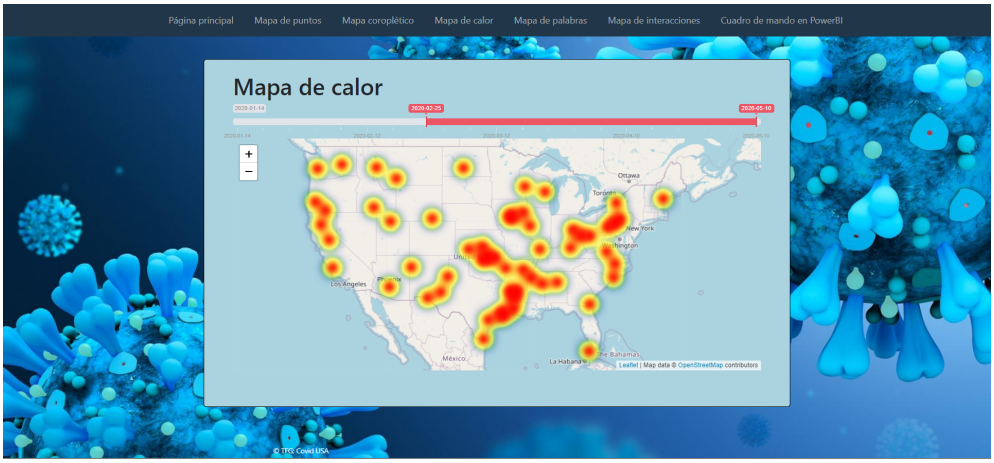


Figura 6.4: Mapa de calor

6.5 Mapa de palabras/hashtags

Seguindo coa liña do mapa coroplético e co obxectivo de continuar mostrando información relacionada coas emocións e expresións, a cuarta páxina da web proporciona información, para cada estado, da palabra ou *hashtag* máis frecuente, condicionado ao período temporal imposto. A selección de representación con palabras ou *hashtags* está suxeita a outro selector, como se ve na figura 6.6.

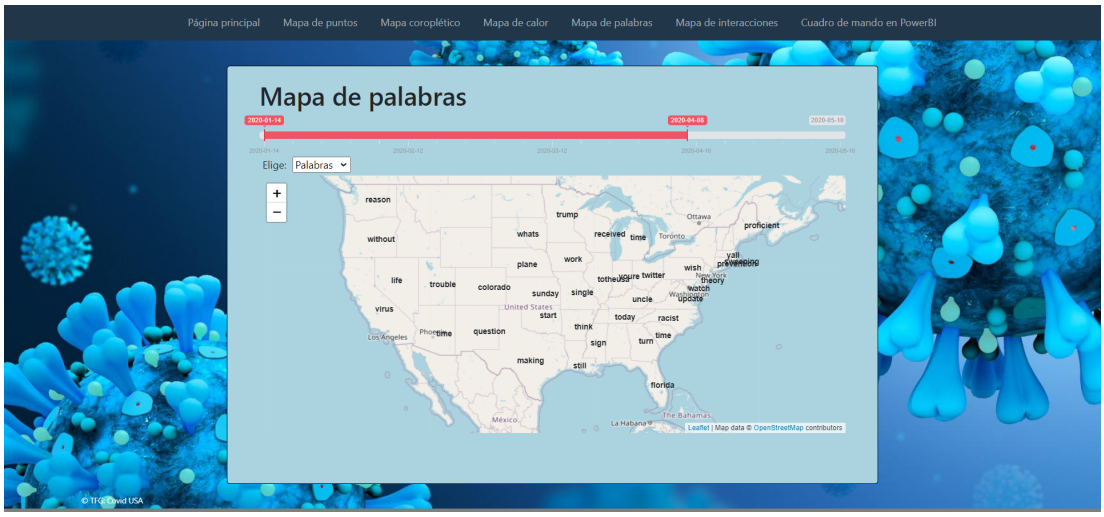
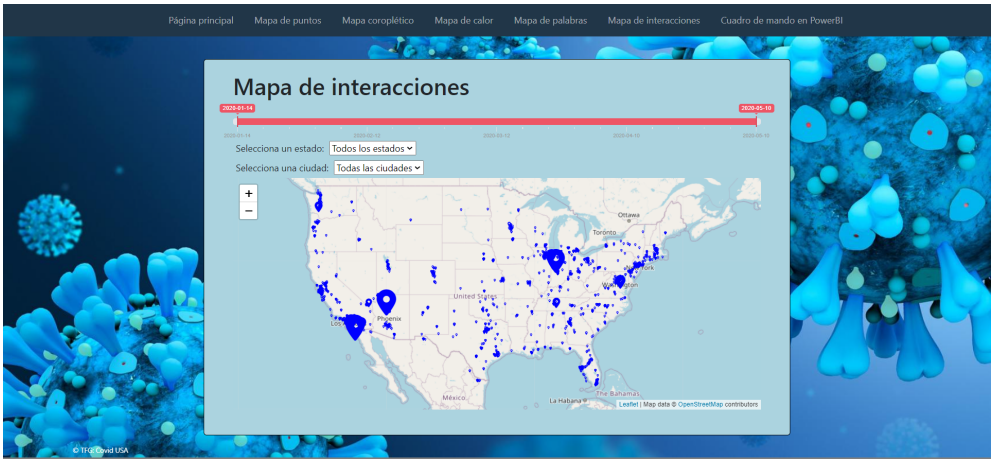


Figura 6.5: Mapa de palabras

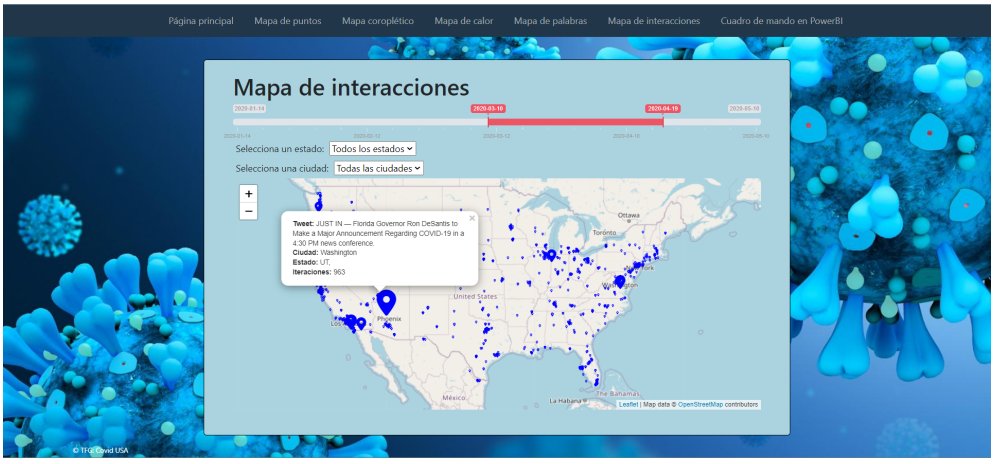
Figura 6.6: Mapa de palabras/hashtags

6.6 Mapa de interaccións

Na mesma liña que as páxinas anteriores anteriores, atópase o mapa de interaccións, que pode verse na figura 6.7, no que se fai énfasis nas interaccións dos usuarios cos chíos, que se leva a cabo mediante os "me gusta", "retweets" e "citas". A idea toma como base o mapa de puntos, cos mesmos filtros, pero añadindo peso sobre os máis relevantes, aumentando así o tamaño do marcador en función das interaccións conxuntas ("me gusta" + "retweets" + "citas") recibidas. Este mapa proporciona unha visión máis profunda sobre a interacción social en relación cos chíos, permitindo unha análise máis detallada das dinámicas de participación e interese dos usuarios nas distintas rexións e momentos temporais.



(a) Páxina do mapa de interaccións



(b) Visión detallada dun chío

Figura 6.7: Mapa de interaccións

6.7 Cadro de mando en Power BI

Para rematar esta sección, preséntase un enfoque algo diferente na última páxina da aplicación. Trátase dun completo cadro de mandos implementado en Power BI, que brinda ao usuario a capacidade de realizar unha ampla gama de filtros sobre os datos, os cales se presentan en diversas representacións gráficas, como se mostra na figura 6.8.

Este cadro de mandos ofrece aos usuarios unha experiencia interactiva na que poden seleccionar e explorar unha información mediante filtros, o que proporciona un profundo control sobre como desexan explorar e analizar os datos. Por un lado, proporciona un gráfico de evolución temporal da cantidade de chíos, que pode filtrarse por estado e por periodo. Por outra parte, proporciona un contador total de chíos, que permite ver de maneira directa a cantidade de información que se manexa para as representacións visuais. Os outros catro gráficos son similares en canto a idea, dous deles representan a cantidade de chíos por estado ou por cidade e os outros dous filtran eses chíos según o sentimento expresado neles. Convén mencionar que tanto o filtro temporal como o filtro de estado aplica a todos os gráficos, así como que varios deles se basan nun deseño inicial desenvolto en Python ao inicio do proxecto, o que demostra a versatilidade das ferramentas utilizadas e a sinerxia entre distintas tecnoloxías no proceso de visualización de datos. En última instancia, este enfoque destaca a importancia da flexibilidade e a personalización na visualización de datos, permitindo que cada usuario adapte a presentación segundo as súas necesidades e preferencias individuais.

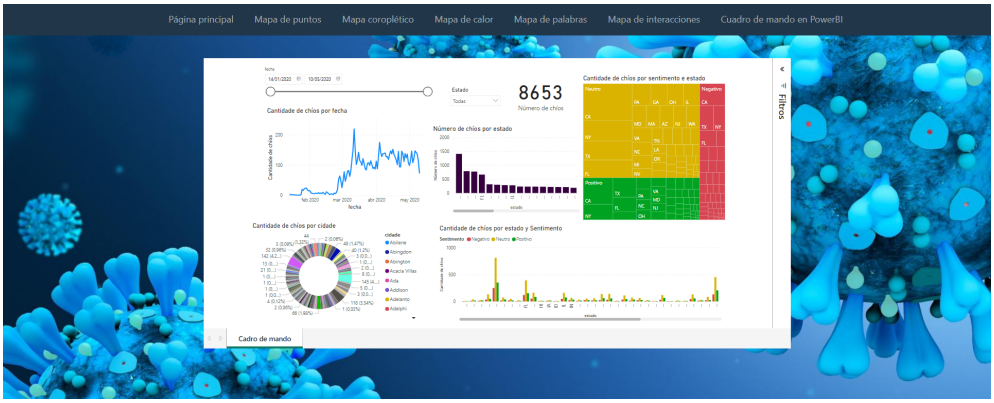


Figura 6.8: Cadro de mando en Power BI

Capítulo 7

Conclusións e traballo futuro

Neste último capítulo, analizaranse o impacto das decisións e solucións, destacando os logros alcanzados e as áreas que poden ser melloradas. Ademais, explorarase como este proxecto senta as bases para a evolución e expansión en futuras iteracións ou en contextos semellantes, presentando oportunidades emocionantes para a continuación do traballo.

7.1 Conclusións

Ao rematar este proxecto, podemos ver que se alcanzaron de maneira satisfactoria os obxectivos propostos incialmente. A meta principal consistía en deseñar e desenvolver un visor web capaz de presentar datos relacionados con chíos sobre COVID-19 nos Estados Unidos, permitindo unha comprensión profunda das tendencias e emocións expresadas en liña. Mediante unha combinación de análise de datos, análise de sentimentos, modelado e deseño de plataforma, alcanzáronse de maneira satisfactoria os obxectivos iniciais establecidos para o proxecto. A través destes esforzos, conseguiuse crear unha ferramenta valiosa para comprender a percepción pública e as emocións relacionadas coa COVID-19 nos Estados Unidos (facilmente ampliable a todo o mundo) axudando a informar decisións e estratexias futuras.

Na fase de análise, abordáronse tanto variables cualitativas como cuantitativas, o que permite obter unha comprensión completa dos patróns e tendencias nas conversas en liña relacionadas co COVID-19, ademais dunha análise de sentimentos, que aporta luz sobre as emocións predominantes nos chíos, o que engade un nivel máis profundo de perspicacia na opinión pública. Estes resultados cumpren co obxectivo de analizar e comprender a evolución das conversas en redor da pandemia.

En canto ao obxectivo do modelado, a elección de empregar SQLAlchemy demostrouse especialmente vantaxosa. O modelo flexible e baseado en obxectos que ofrece, permite unha adaptación eficiente aos datos históricos, o que aporta un valor significativo ao proxecto. A flexibilidade en termos de estrutura, relacións e mantemento, xunto coa capacidade de incor-

porar cambios futuros sen dificultade, permitiu non só modelar os datos históricos de maneira precisa e completa, senón tamén manter unha base de datos actualizada e relevante. Isto resulta crítico para asegurar que a plataforma sexa unha fonte de información útil e pertinente, garantindo que poida seguir proporcionando ideas valiosas e precisos no futuro.

En última instancia, o visor web resultante non só representa unha solución técnica exitosa, senón que tamén ten aplicacións concretas no mundo real. Desenvolveuse unha interface sinxela e práctica para que os usuarios poidan explorar e consultar eficazmente os datos sobre a evolución da pandemia en diferentes rexións e ao longo do tempo. Ademais, ao integrar datos xeorreferenciados coas emocións e opinións expresadas nos mensaxes sobre a COVID-19, pode obterse unha comprensión máis completa da percepción pública e das tendencias emocionais en diversas áreas dos Estados Unidos. Deste xeito, non só ofrece unha visión superficial, senón que profundiza na interacción entre a localización xeográfica e as emocións expresadas, o que enriquece a comprensión das reaccións públicas diante da COVID-19.

É esencial recoñecer o potencial duradeiro deste proxecto e o seu posible impacto en futuros contextos de pandemias. A medida que a nosa sociedade enfronta un novo brote de COVID-19 e considera a posibilidade de futuras pandemias, a plataforma que se desenvolveu pode desempeñar un papel crucial. É importante destacar que esta información reflicte opinións e emocións expresadas en Twitter, e non debe confundirse cos datos reais sobre a pandemia. A capacidade de identificar os sentimentos predominantes por estado a través de mapas coropléticos permite ás partes interesadas, como os profesionais de saúde pública e os analistas de políticas, comprender como as opinións varían xeograficamente. Iso pode ser valioso para a toma de decisións informadas sobre estratexias de comunicación e políticas relacionadas coa pandemia. A función de identificar as palabras ou *hashtags* máis frecuentes por estado brinda unha perspectiva adicional sobre as conversas locais. Isto pode ser de gran utilidade para as empresas, organizacións e profesionais de marketing que desexen axustar as súas estratexias segundo as tendencias locais. A característica de variar o tamaño dos marcadores segundo a interacción do tweet pode ser particularmente valiosa para xornalistas e medios de comunicación que busquen identificar contido moi influente en tempo real.

Un aspecto destacable deste proxecto é a percepción do crecemento ao longo do seu desenvolvemento, ao aplicar os coñecementos adquiridos en varias materias do grao e unir eses coñecementos para crear un proxecto completo. En termos de análise estatística, puxéronse en práctica os coñecementos adquiridos en asignaturas como *Modelización Estatística de Datos de Alta Dimensión (MEDAD)*. Para a análise de sentimentos e a extracción de información con expresións regulares, empregáronse os coñecementos de *Procesamento da Linguaxe Escrita (PLE)*. As habilidades adquiridas en *Modelaxe de Bases de Datos (MBD)*, *Bases de Datos Analíticas (BDA)*, *Representación e Xestión de Datos Espazo-Temporais (RXDET)* e conceptos de *Tecnoloxías de Integración (TI)* demostraron ser especialmente relevantes para o modelado

adecuado, os procesos ETL e a manipulación de datos espazo-temporais, que foron críticos neste proxecto. A integración destes coñecementos subliña a importancia dun enfoque multidisciplinario e da aplicación práctica dos conceptos académicos nun proxecto real.

7.2 Traballo futuro

A continuación explóranse varias liñas para continuar co traballo no futuro.

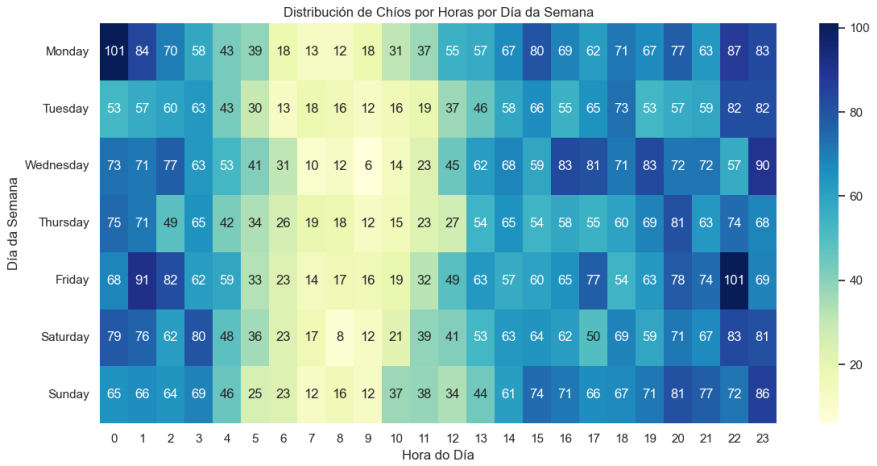
- **Escalabilidade e rendemento**: Asegurar que a plataforma sexa capaz de manexar volumes máis grandes de datos sen comprometer o rendemento, de modo que poida crecer cos requisitos futuros.
- **Análise comparativa con outras pandemias**: Examinar e comparar os sentimentos e opinións expresadas durante outras pandemias, como a gripe H1N1 ou o Ébola, para identificar patróns e variacións nas reaccións do público.
- **Continuar a optimizar a experiencia do usuario**, incorporando retroalimentación dos usuarios para facer a plataforma máis intuitiva e accesible.
- **Expandir a capacidade de extraer información relevante**, como tendencias de saúde ou cambios políticos, para proporcionar ideas adicionais e diversificar a utilidade da plataforma. Ademais, mellorar a extracción actual, manexando, por exemplo, a información dos enlaces referenciados nos chíos ou o contido visual.
- **Predición de tendencias**: Explorar modelos de predición baseados en datos históricos e tendencias de sentimentos, ofrecendo unha perspectiva anticipada sobre os posibles cambios nas percepcións públicas e sentimentos durante futuros brotes ou pandemias.
- **Maior precisión na análise de sentimentos**: Ampliar o estudo con outros algoritmos de aprendizaxe automática como redes neurais ou modelos preentrenados; ou técnicas avanzadas como a representación de termos por TF-IDF, co obxectivo de mellorar a precisión e a captura dos sentimentos expresados nas mensaxes.
- **Integración de redes sociais adicionais**: Ampliar a plataforma para incluír análises de conversacións e sentimentos doutras redes sociais relevantes, como Instagram ou Facebook, para obter unha imaxe máis completa das actitudes públicas.

Apéndices

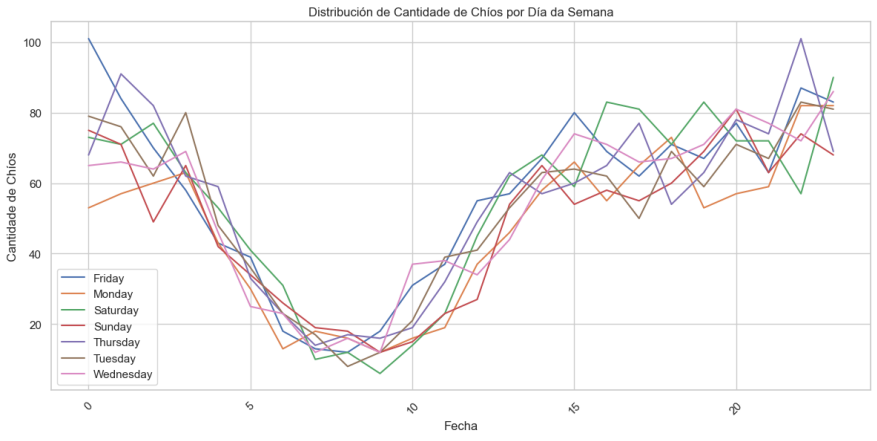
Apéndice A

Gráficas da análise

Nesta sección, amósanse as figuras do Apartado 4.4.3.



(a) Mapa de calor de chíos por días da semana e horas



(b) Gráfico de liñas de chíos por días da semana e horas

Figura A.1: Representacións da evolución dos chíos por día da semana e hora

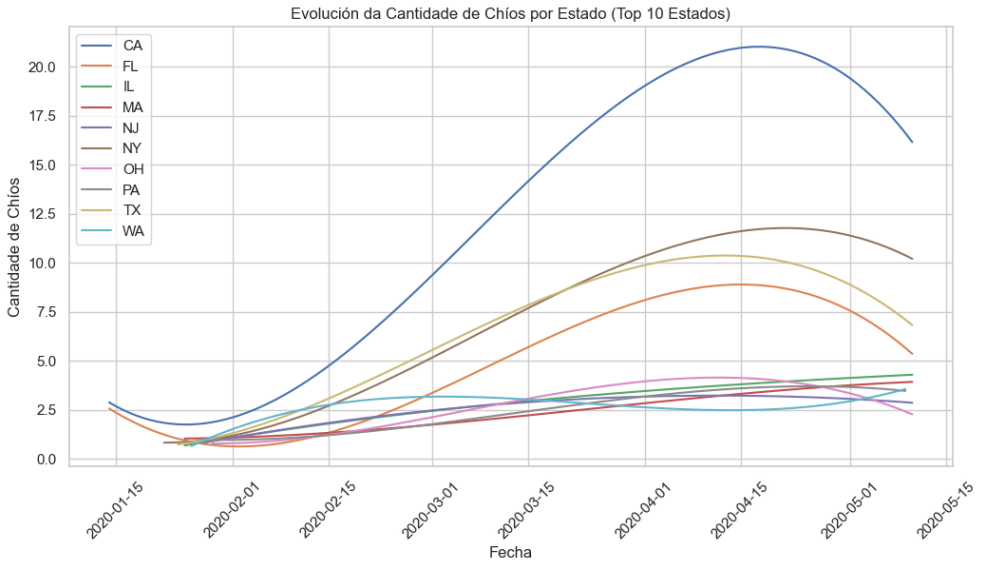


Figura A.2: Evolución dos chíos nos 10 estados máis activos en mensaxes de Twitter

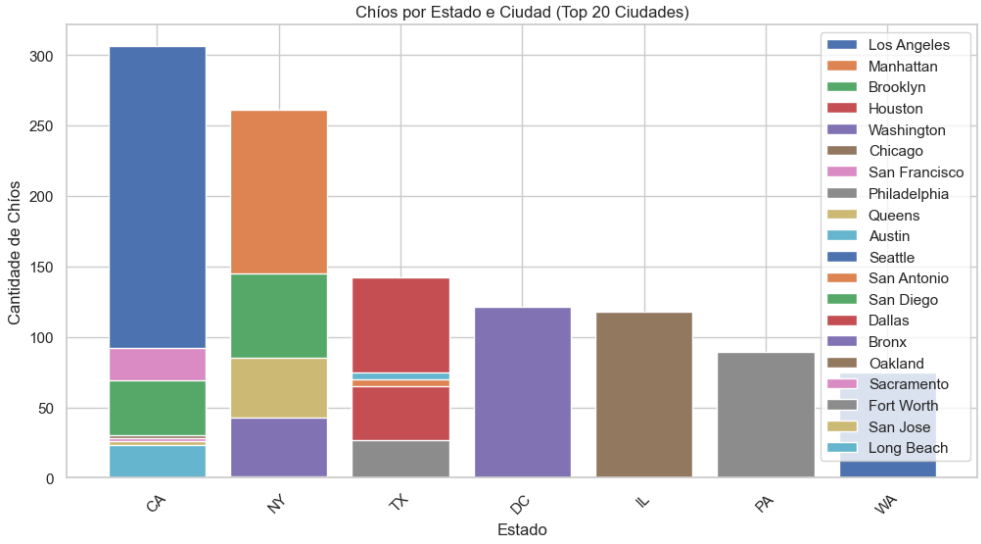


Figura A.3: Gráfico para ver si as 20 cidades máis activas pertencen aos 10 estados máis activos

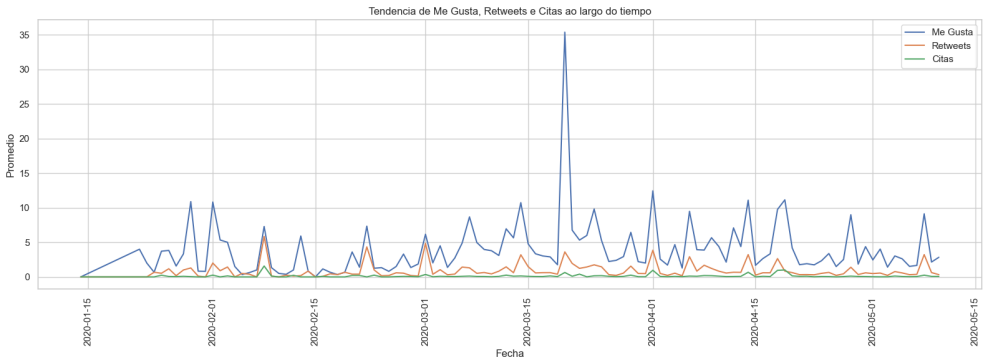


Figura A.4: Evolución das interaccións

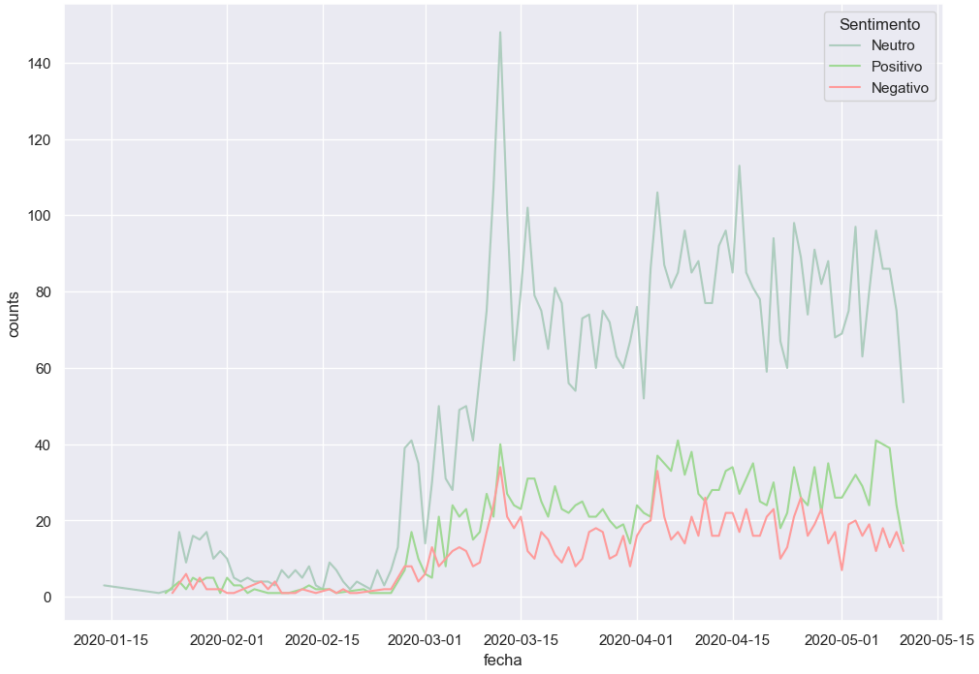


Figura A.5: Evolución temporal dos sentimentos

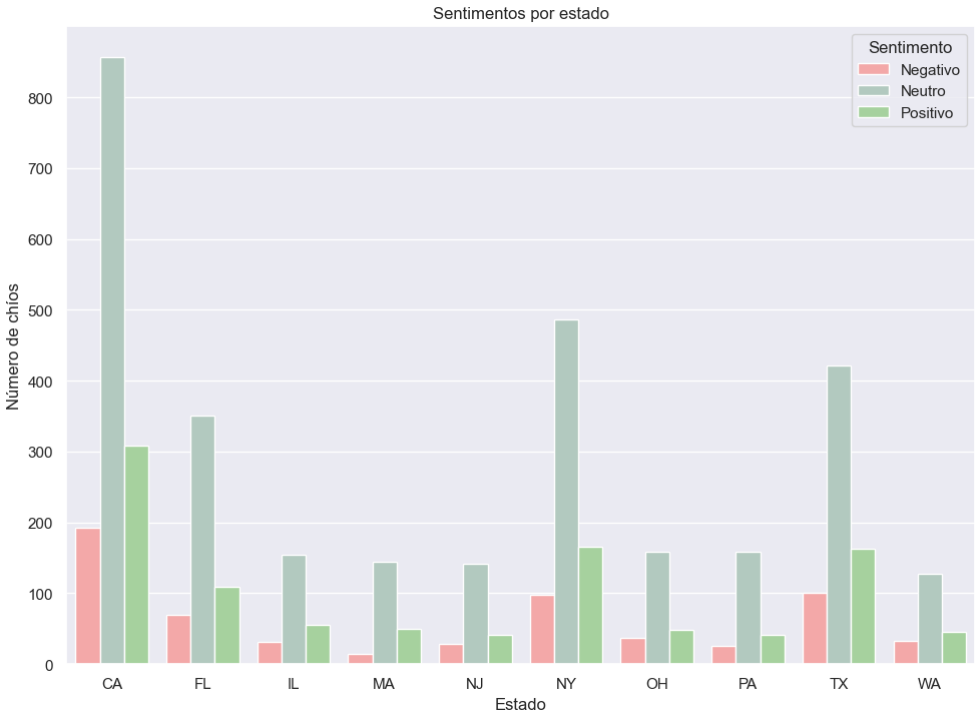


Figura A.6: Frecuencia para os 10 estados máis activos de cada sentimento

Apéndice B

Prototipos de pantalla

Neste capítulo, presenséntanse os deseños da interface que, por motivos de espazo, non se puideron representar no Apartado 5.5.

Figura B.1: Deseño da páxina do mapa coroplético

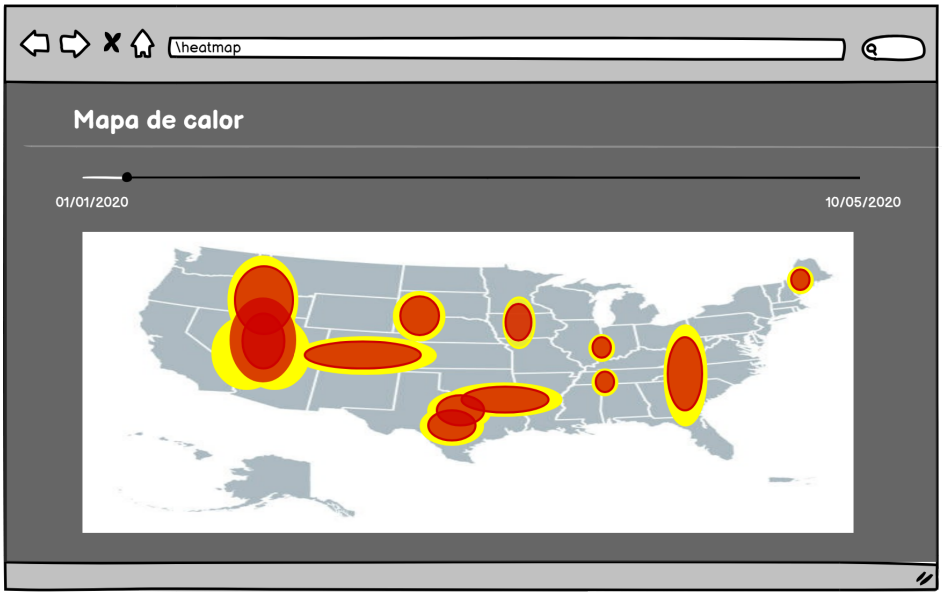


Figura B.2: Deseño da páxina do mapa de calor

Figura B.3: Deseño da páxina do mapa de termos

Figura B.4: Deseño da páxina do mapa de interaccións

Figura B.5: Deseño da páxina do cadro de mando

Apéndice C

Imaxes adicionais

Nesta sección veránse imaxes adicionais do proceso de ETL en Microstrategy que, por problemas de espazo, non se mostraron no Apartado 5.6.

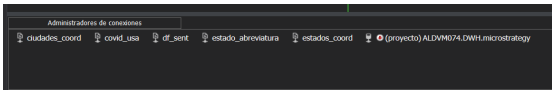


Figura C.1: Conexións empregadas

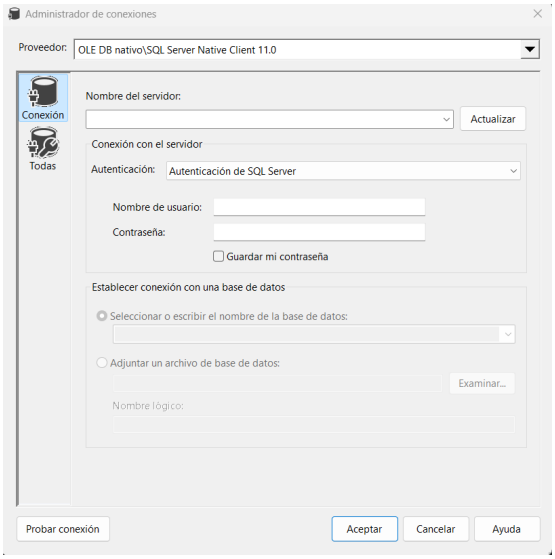
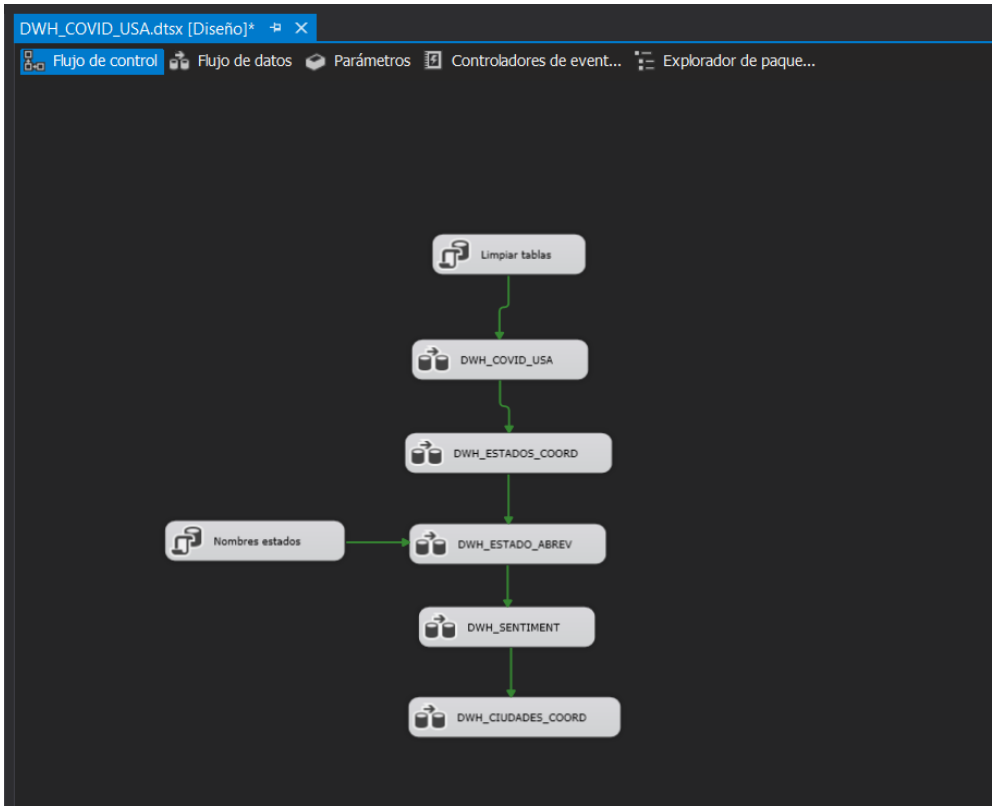
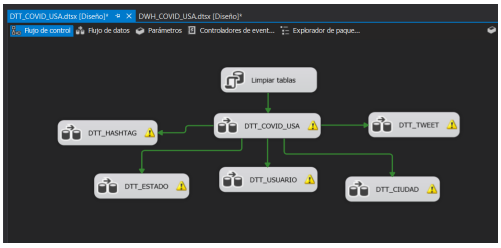


Figura C.2: Detalle de creación dunha conexión

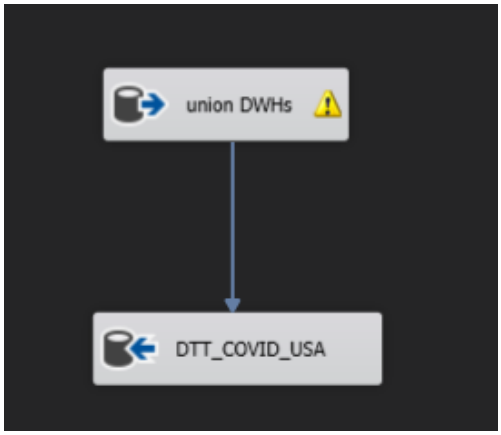
Figura C.3: Conexións do proxecto



(a) Paquete de DWH



(b) Paquete de DTT



(c) Visión detallada dun compoñente de DTT

Figura C.4: Visualización dos paquetes da solución de Microstrategy

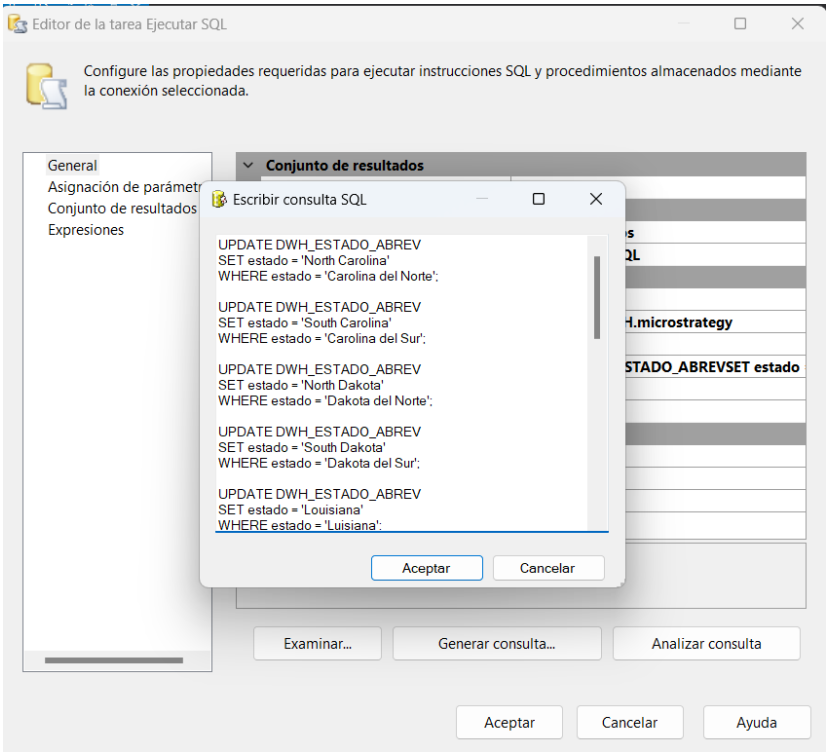


Figura C.5: Exemplo de sentenza no compoñente de sentenza SQL

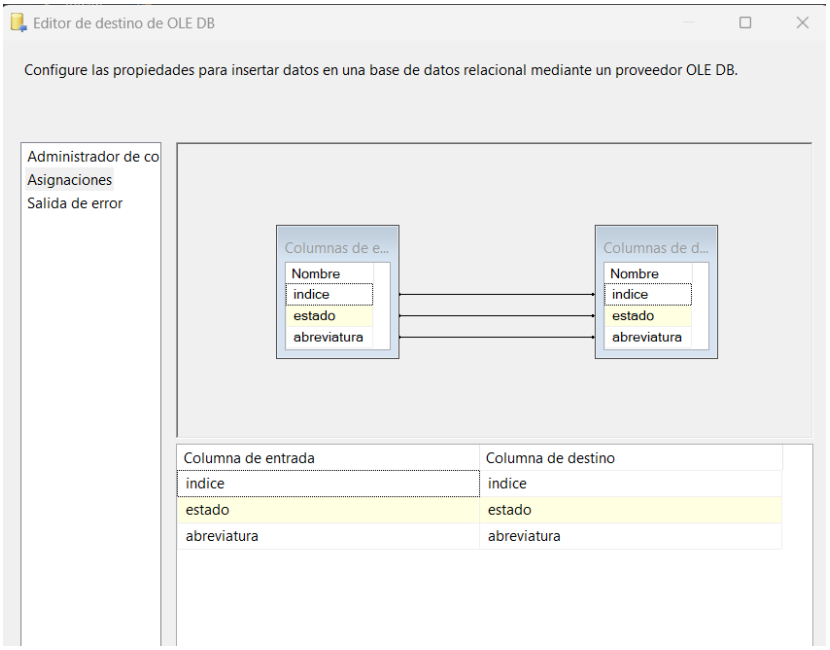
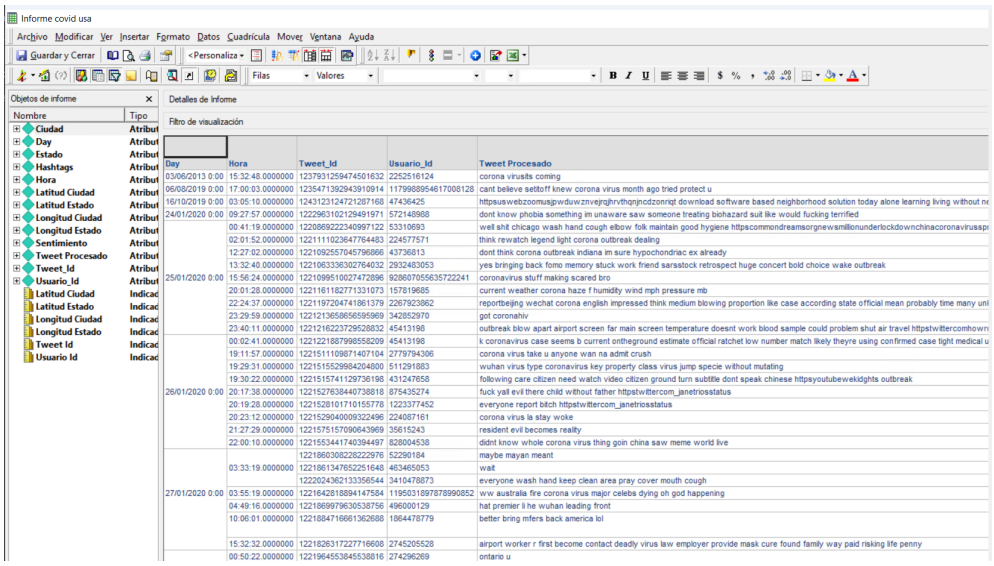
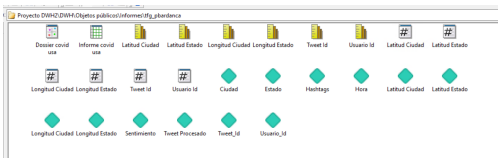


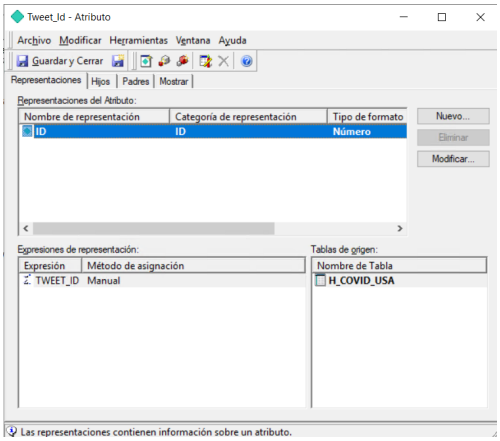
Figura C.6: Exemplo de asignacións entre orixe OLDB e destino OLDB



(a) Informe que alimenta o dosier en Microstrategy



(b) Obxectos creados en Microstrategy



(c) Detalle da creación dun obxecto

Figura C.7: Informe e obxectos en Microstrategy

Relación de Acrónimos

API Application Programming Interfaces. ii, 7–9, 14, 16, 19, 20, 22, 58, 61

CSS Cascading Style Sheets. 8, 57–59

CSV Comma-Separated Value. 9, 22, 65–67

ETL Extract, Transform, Load. 14, 64, 66, 80

GPS Global Positioning System. 20

HTML HyperText Markup Language. 8, 21, 58, 59

HTTP HyperText Transfer Protocol. 9, 58, 61

HU Historias de Usuario. 53

IQR Rango Intercuartil. 48

JS JavaScript. 8, 57–59, 69

NLP Natural Language Processing. 9

NLTK Natural Language Toolkit. 9, 44, 48, 51, 52

OLE DB Object Linking and Embedding, Database. 67

ORM Object-Relational Mapping. 8

REST Representational State Transfer. ii, 58, 61

SIG Sistema de Información Xeográfica. 14

SMART Specific, Measurable, Achievable, Relevant and Timely. 11

SQL Structured Query Language. 57, 67

SSIS Servizo de Integración de SQL Server. 8, 66

SXBD Sistema de Xestión de Base de Datos. 57, 68

TF-IDF Frecuencia de Termo – Frecuencia inversa de documento. 80

URL Uniform Resource Locator. 24, 61

Bibliografía

[1] C. Sánchez-Cantalejo, M. D. M. Rueda, M. Saez, I. Enrique, R. Ferri, M. Fuente, R. Villegas, L. Castro, M. A. Barceló, A. Daponte-Codina, N. Lorusso, and A. Cabrera-León, "Impact of covid-19 on the health of the general and more vulnerable population and its determinants: Health care and social survey-essoc, study protocol," *Int J Environ Res Public Health*, vol. 18, no. 15, p. 8120, Jul 2021.

[2] J. G. Fernández, "X: así es la nueva vida de twitter con musk," *EXPANSION*, Sep. 01 2023, (consultado o 04/09/2023). [En liña]. Dispoñible en: https://www.expansion.com/directivos/2023/09/01/64f12015e5fdeace6d8b45a1.html

[3] O. W. in Data. (2020, Mar. 01) Covid-19 data explorer. (consultado o 02/08/2023). [En liña]. Dispoñible en: https://shorturl.at/dxGO9

[4] Ministerio Para La Transición Ecológica Y El Reto Demográfico. Our world in data. Ministerio Para La Transición Ecológica Y El Reto Demográfico. (consultado o 04/08/2023). [En liña]. Dispoñible en: https://www.miteco.gob.es/es/ceneam/recursos/pag-web/our-world-in-data.html

[5] Autor/es, "Senseplace2: Geotwitter analytics support for situational awareness," *IEEE Conference Publication | IEEE Xplore*, 10 2011, (consultado o 04/08/2023). [En liña]. Dispoñible en: https://shorturl.at/ilrB4

[6] K. Zaballa *et al.*, *Social Response to COVID-19 SMART Dashboard: Proposal for Case Study*, 1 2022, pp. 154–165, (consultado o 04/08/2023).

[7] T. B. Team. (2022, 1) #covid 19 – twitter evolution. Tweet Binder. (consultado o 04/08/2023). [En liña]. Dispoñible en: https://www.tweetbinder.com/blog/covid-19-coronavirus-twitter/

[8] C. Stokel-Walker and M. S. Godoy. (2023, 3) La api de twitter cuesta 42 mil dólares al mes y deja fuera de juego a casi todo el mundo. (consultado o 04/08/2023). [En liña].

Dispoñible en: https://es.wired.com/articulos/api-de-twitter-cuesta-42-mil-dolares-al-mes-deja-fuera-de-juego-a-casi-todos

[9] J. R. González. (2023, 3) El nuevo sistema pago de api de twitter es el fin de los desarrolladores en la red social. (consultado o 04/08/2023). [En liña]. Dispoñible en: https://www.enter.co/especiales/dev/el-nuevo-sistema-pago-de-api-de-twitter-es-el-fin-de-los-desarrolladores-en-la-red-social/

[10] "Selenium," https://www.selenium.dev/, (consultado o 08/08/2023).

[11] "Welcome to python.org," https://www.python.org/, (consultado o 08/08/2023).

[12] "Sqlalchemy," https://www.sqlalchemy.org/, (consultado o 08/08/2023).

[13] "Geoalchemy 2 documentation — geoalchemy2 0.14.1.dev5+g80e738d documentation," https://geoalchemy-2.readthedocs.io/en/latest/, (consultado o 08/08/2023).

[14] "Postgresql," https://www.postgresql.org/, (consultado o 08/08/2023).

[15] "Home," https://postgis.net/, (consultado o 08/08/2023).

[16] "Welcome to flask — flask documentation (2.3.x)," https://flask.palletsprojects.com/en/2.3.x/, (consultado o 08/08/2023).

[17] "Html: Hypertext markup language | mdn," https://developer.mozilla.org/en-US/docs/Web/HTML, jul. 18, 2023.

[18] "Css | mdn," https://developer.mozilla.org/es/docs/Web/CSS, mar. 13, 2023.

[19] "Javascript | mdn," https://developer.mozilla.org/es/docs/Web/JavaScript, jul. 24, 2023.

[20] "Leaflet — an open-source javascript library for interactive maps," https://leafletjs.com/, (consultado o 08/08/2023).

[21] Chugugrace, "Sql server integration services - sql server integration services (ssis)," https://learn.microsoft.com/es-es/sql/integration-services/sql-server-integration-services?view=sql-server-ver16, (consultado o 09/08/2023).

[22] "Business intelligence y soluciones analíticas," https://www.microstrategy.com/es, (consultado o 08/08/2023).

[23] "Visualización de datos | microsoft power bi," https://powerbi.microsoft.com/es-es/, (consultado o 08/08/2023).

[24] "pandas - python data analysis library," https://pandas.pydata.org/, (consultado o 08/08/2023).

[25] "Geopandas 0.13.2 — geopandas 0.13.2+0.gd5add48.dirty documentation," https://geopandas.org/en/stable/, (consultado o 08/08/2023).

[26] "seaborn: statistical data visualization — seaborn 0.12.2 documentation," https://seaborn.pydata.org/, (consultado o 08/08/2023).

[27] "Plotly: Low-code data app development," https://plotly.com/, (consultado o 08/08/2023).

[28] "Matplotlib — visualization with python," https://matplotlib.org/, (consultado o 08/08/2023).

[29] "vadersentiment," https://pypi.org/project/vaderSentiment/, may 22, 2020.

[30] "Nltk :: Natural language toolkit," https://www.nltk.org/, (consultado o 08/08/2023).

[31] "Textblob: Simplified text processing — textblob 0.16.0 documentation," https://textblob.readthedocs.io/en/dev/, (consultado o 08/08/2023).

[32] "Numpy," https://numpy.org/, (consultado o 08/08/2023).

[33] "math — mathematical functions," https://docs.python.org/3/library/math.html, Python Documentation, (consultado o 08/08/2023).

[34] "requests," https://pypi.org/project/requests/, (consultado o 08/08/2023).

[35] "csv — csv file reading and writing," https://docs.python.org/3/library/csv.html, Python Documentation, (consultado o 08/08/2023).

[36] "re — operaciones con expresiones regulares," https://docs.python.org/es/3/library/re.html, Python Documentation, (consultado o 08/08/2023).

[37] "Wordcloud for python documentation — wordcloud 1.8.1 documentation," https://amueller.github.io/word_cloud/, (consultado o 08/08/2023).

[38] "chardet," https://pypi.org/project/chardet/, (consultado o 08/08/2023).

[39] Desarrollo iterativo incremental | realworld. (consultado o 02/08/2023). [En liña]. Dispoñible en: https://www.runroom.com/realworld/desarrollo-iterativo-incremental

[40] J. Arino and J. Arino, "Buenas prácticas de scrum," Scrum Master, Nov. 2021, (consultado o 02/08/2023). [En liña]. Dispoñible en: https://metodologiascrum.top/buenas-practicas-de-scrum/

[41] J. Bernal. (2022) Scrum – gestión Ágil de proyectos i. BLMovil. (consultado o 02/08/2023). [En liña]. Dispoñible en: https://www.blmovil.com/scrum-gestion-agil-de-proyectos-i/

[42] C. Muratet, "Objetivos smart o inteligentes: qué son, pasos para definirlos y ejemplos," www.inboundcycle.com, Apr. 2022, (consultado o 02/08/2023). [En liña]. Dispoñible en: https://www.inboundcycle.com/blog-de-inbound-marketing/objetivos-inteligentes-smart-que-son-pasos-para-definirlos

[43] "Balsamiq wireframes - industry standard low-fidelity wireframing software | balsamiq," https://balsamiq.com/wireframes/, (consultado o 02/08/2023).

[44] "Dbeaver community | free universal database tool," https://dbeaver.io/, (consultado o 02/08/2023).

[45] draw.io. (consultado o 02/08/2023). [En liña]. Dispoñible en: https://www.drawio.com/

[46] "Almacenamiento personal en la nube: Microsoft onedrive," https://www.microsoft.com/es-es/microsoft-365/onedrive/online-cloud-storage, (consultado o 02/08/2023).

[47] "Overleaf, online latex editor," https://www.overleaf.com/, (consultado o 02/08/2023).

[48] Microsoft, "Visual studio code - code editing. redefined," https://code.visualstudio.com/, Nov 2021, (consultado o 02/08/2023).

[49] C. Stokel-Walker and M. S. Godoy, "La api de twitter cuesta 42 mil dólares al mes y deja fuera de juego a casi todo el mundo," *WIRED*, Mar. 13 2023, (consultado o 02/08/2023). [En liña]. Dispoñible en: https://es.wired.com/articulos/api-de-twitter-cuesta-42-mil-dolares-al-mes-deja-fuera-de-juego-a-casi-todos

[50] D. B. Boldu, "Microsoft power bi es líder en el gartner magic quadrant," *Aglaia*, Apr. 2023, (consultado o 04/09/2023). [En liña]. Dispoñible en: https://aglaia.es/blog/power-bi/microsoft-power-bi-lider-en-gartner-magic-quadrant-2023/

[51] Glassdoor, "Sueldo: Ingeniero de datos junior en españa en 2023," https://www.glassdoor.es/Sueldos/ingeniero-de-datos-junior-sueldo-SRCH_KO0,25.htm, (consultado o 02/08/2023).

[52] "Táboas retributivas," https://www.udc.es/es/goberno/equipo_reitoral/xerencia/servizos/retribucions_seguridade_social_e_accion_social/taboas_retributivas/, (consultado o 02/08/2023).

[53] K. Leetaru, "Visualizing seven years of twitter's evolution: 2012-2018," *Forbes*, Mar. 04 2019, (consultado o 04/08/2023). [En liña]. Dispoñible

en: https://www.forbes.com/sites/kalevleetaru/2019/03/04/visualizing-seven-years-of-twitters-evolution-2012-2018/?sh=443bc89e7ccf

[54] "Visualizando oito anos da evolución de twitter: 2012-2019 - o proxecto gdelt," https://blog.gdeltproject.org/visualizing-eight-years-of-twitters-evolution-2012-2019/, (consultado o 05/08/2023).

[55] "Index of /geo-tagged_twitter_datasets/50_states_and_d.c/," https://yunhefeng.me/geo-tagged_twitter_datasets/50_States_and_D.C/, (consultado o 05/08/2023).

[56] A. Demircioğlu, "Twitter snowflake approach is cool," *Medium*, Dec. 18 2022, (consultado o 05/08/2023). [En liña]. Dispoñible en: https://atakde.medium.com/twitter-snowflake-approach-is-cool-3156f78017cb

[57] W. contributors, "Snowflake id," https://en.wikipedia.org/wiki/Snowflake_ID, Jul. 2023, (consultado o 05/08/2023).

[58] C. de Wikipedia, "Iso 8601," https://es.wikipedia.org/wiki/ISO_8601, Apr. 2023, (consultado o 05/08/2023).

[59] "datetime — tipos básicos de fecha y hora," https://docs.python.org/es/3/library/datetime.html, Python Documentation, (consultado o 05/08/2023).

[60] P. D. Allison, "Missing data," *European Radiology*, vol. 16, p. 966, 2005. [En liña]. Dispoñible en: https://api.semanticscholar.org/CorpusID:24362847

[61] colaboradores de Wikipedia, "Anexo:abreviaciones de los estados de estados unidos - wikipedia, la enciclopedia libre," https://es.wikipedia.org/wiki/Anexo:Abreviaciones_de_los_estados_de_Estados_Unidos, (consultado o 07/08/2023).

[62] "Categoría:washington d. c. - wikinoticias," https://es.wikinews.org/wiki/Categor%C3%ADa:Washington_D._C., (consultado o 07/08/2023).

[63] I. Pereira and A. Mitropoulos, "A year of covid-19: What was going on in the us in march 2020," *ABC News*, Mar. 07 2021, (consultado o 07/08/2023). [En liña]. Dispoñible en: https://abcnews.go.com/Health/year-covid-19-us-march-2020/story?id=76204691

[64] Cnnee, "Del primer caso a más de 400.000 muertes en un año: el fatal primer aniversario del covid-19 en ee.uu." *CNN*, Jan. 2021, (consultado o 07/08/2023). [En liña]. Dispoñible en: https://cnnespanol.cnn.com/2021/01/21/del-primer-caso-a-mas-de-400-000-muertes-en-un-ano-el-fatal-primer-aniversario-del-covid-19-en-ee-uu/

[65] V. Sabater, "Las emociones que se comparten en redes sociales son contagiosas," *La Mente Es Maravillosa*, Mar. 15 2021, (consultado o 07/08/2023). [En liña]. Dispoñible en: https://lamenteesmaravillosa.com/las-emociones-que-se-comparten-en-redes-sociales-son-contagiosas/

[66] D. I. IonDen.com, "Ion.rangeslider - jquery range slider | ionden.com," http://ionden.com/a/plugins/ion.rangeSlider/, (consultado o 09/08/2023).