

UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA

***La Universidad Católica de Loja***

**PRACTICUM 1.1**

**Docente:**

Ing. Danilo Rubén Jaramillo Hurtado

**Estudiantes:**

Roberto Narváez

Juan Yanza

**Tema:**

Proyecto Integrador de Saberes

**PERÍODO ACADÉMICO OCTUBRE 2019 – FEBRERO 2020**

**Loja-Ecuador**

# ÍNDICE

[ÍNDICE II](#_Toc30020879)

[Índice de tablas IV](#_Toc30020880)

[Resumen V](#_Toc30020881)

[Problemática VI](#_Toc30020882)

[Objetivos VII](#_Toc30020883)

[Objetivo general VII](#_Toc30020884)

[Objetivos especíﬁcos VII](#_Toc30020885)

[Metodología VIII](#_Toc30020886)

[Introducción 1](#_Toc30020887)

[Marco Referencial (APA) 2](#_Toc30020888)

[Programación Funcional 2](#_Toc30020889)

[Base de Datos 2](#_Toc30020890)

[GitHub 2](#_Toc30020891)

[VEGA – LITE 4](#_Toc30020892)

[¿Qué es ODS? 4](#_Toc30020893)

[Lenguaje SQL 5](#_Toc30020894)

[Librería Pandas 5](#_Toc30020895)

[Lenguaje Python 6](#_Toc30020896)

[Modelo Entidad Relación 6](#_Toc30020897)

[Modelo Lógico 7](#_Toc30020898)

[Modelo Físico 7](#_Toc30020899)

[SQL Developer 8](#_Toc30020900)

[Solución 9](#_Toc30020901)

[Validación 20](#_Toc30020902)

[Conclusiones 21](#_Toc30020903)

[Bibliografía 22](#_Toc30020904)

[Anexos 23](#_Toc30020905)

**ÍNDICE DE ILUSTRACIONES**

[Ilustración 1 Repositorio GitHub 3](#_Toc30020906)

[Ilustración 2: Dataset proyecto 10](#_Toc30020907)

[Ilustración 3- Modelo Conceptual 13](#_Toc30020908)

[Ilustración 4-Modelo Lógico 14](#_Toc30020909)

[Ilustración 5 Limpieza ods\_1\_2 17](#_Toc30020910)

[Ilustración 6 Generar csv - días 18](#_Toc30020911)

[Ilustración 7 Publicaciones por día 18](#_Toc30020912)

[Ilustración 8 Publicaciones por hora 19](#_Toc30020913)

# Índice de tablas

[Tabla 1 Tipo de datos 11](#_Toc30020914)

[Tabla 2 - Diccionario de datos 16](#_Toc30020915)

# Resumen

# Problemática

A partir de un dataset se debe de diseñar un modelo de base de datos el cual se lo validará posteriormente mediante consultas sql por lo cual la base del proyecto es el modelado de la base de datos. Lo imprescindible es el modelo conceptual del cual se partirá, para lo cual se deben de tener conceptos básicos para superar dicha etapa.

Al tener una gran cantidad de datos se debe de tener consideraciones de limpiado del ods debido a que posteriormente puede generar errores por campos nulos o vacíos que tienen definido en el modelo físico un tipo de dato en específico o un dominio predeterminado.

Se debe de buscar integridad en los datos por lo cual se deben implementar distintas normas de normalización de la base de datos para que no existan datos inconsistentes o repetidos.

Del mismo modo en base al dataset se generarán nuevos archivos usando la parte de programación funcional y reactiva analizar un volumen de datos desde una IDE o consola no es lo óptimo por lo que se debe de acudir a herramientas en las cuales se facilite la visualización mediante gráficos estadísticos.

# Objetivos

## Objetivo general

* Realizar un análisis exploratorio de datos a un data set

## Objetivos especíﬁcos

* Construir un modelo relacional de base datos que represente a las entidades que se encuentran presentes en el dataset.
* Construir un conjunto de consultas SQL para extraer datos almacenados en la base de datos con el ﬁn de proporcionar información relevante.
* Utilizar conceptos de programación funcional para realizar análisis exploratorio de datos.
* Construir un conjunto de visualizaciones que permitan ver el resultado de los análisis realizados.

# Metodología

En grupo de dos personas se nos asignó un dataset el cual deberá ser analizado y tratado para así poder conseguir los resultados esperado.

El manejo de versiones y documentación se lo llevara mediante la plataforma Github y su respectiva wiki, para ello, un integrante del grupo creara una tarea grupal a través de GitClassrom[[1]](#footnote-2). Seguidamente el segundo integrante procede a ingresar mediante un enlace de invitación al repositorio grupal.

Para el proceso del proyecto definimos tres fases de trabajo, en las cuales se deberán seguir las acciones siguientes:

A continuación, se describe cada fase y acción:

**Planificación:** Se identifica cada tarea y proceso, esfuerzo y tiempo que conllevara cada tarea y el cronograma con su respectiva fecha por tarea.

**Ejecución:** Esta es la fase que más tiempo y esfuerzo conlleva puesto que es donde se trabajará el dataset, y seguidamente se migraran los datos al SGBD.

**Análisis:** Se realiza el diagrama entidad relación, y el modelo conceptual de la base de datos.

**Diseño:** Se identifica la estructura necesaria para nuestra base de datos, implementando las claves principales y foráneas.

**Construcción:** Desarrollo de Scripts SQL con el fin de ir construyendo las tablas y demás estructuras de la base de datos.

**Cierre:** Se presenta los resultados esperados y se defiende el proyecto.

El proyecto que realizamos tiene una metodología de tipo cuantitativa puesto que se hizo uso de técnicas de programación funcional para así obtener resultados en forma de gráficas, tiene un diseño experimental puro ya que a partir de nuestras acciones como limpieza de dataset, consultas y aplicación de programación funcional obtuvimos los resultados.

# Introducción

Para poder entender y analizar de una mejor forma nuestra dataset, se debe tener en cuenta diversos conceptos, entre ellos; la red social Twitter, ODS, base de datos, programación funcional y reactiva y entornos de apoyo de proyectos (Vega, Github, OneDrive).

Twitter es conocido por ser una de las plataformas sociales más grandes del mundo, ésta es un servicio de comunicación bidireccional con el que puedes compartir información de diversos tipos de una forma veloz, sencilla y gratuita.

Según (WebEmpresa, 2018) Twitter:

Es un formato muy completo, que permite obtener de forma inmediata información en formato de titular, rápido de leer y rápido de escribir.

Esta plataforma tiene carácter de autopublicación basado en la inmediatez de sus mensajes y es una red perfectamente estructurada para compartir experiencias y vivencias en el momento en que suceden.

El tipo de contenido que se publica en Twitter es de diversa naturaleza: podemos encontrarnos mensajes personales, fotografías, infografías, información corporativa, noticias, eventos, descuentos, publicidad, etc.

ODS también llamado (del inglés Operational Data Store) es un tipo de dbase de datos que se usa comúnmente como área lógica provisional para un almacén de datos, es decir mientras los datos se encuentran en el ODS éstos pueden ser borrados, arreglados por redundancia y revisados.

# Marco Referencial (APA)

## Programación Funcional

Según (Hoyos & Puertas, 2017):

Se aclara que el mecanismo escogido es el de una paralelización básica mediante la mónada Eval, ya que en la actualidad, existen otros enfoques que buscan mejorar el desempeño mediante el uso de mecanismos más sofisticados. La mónada Eval define varias operaciones que hacen posible la paralelización en la evaluación de expresiones. Con la operación rpar se establece que el argumento debe ser evaluado en forma paralela, y con la operación rseq se fuerza una evaluación secuencial de los argumentos. (p.67)

## Base de Datos

Para (Corzo, Portillo, & Zamora, 2006)

La propuesta para el análisis sintáctico es el uso de un diccionario de palabras validas para el dominio de la aplicación. Dicho diccionario esta constituido por una tabla en una base de datos que contiene palabras propias del dominio. La estructura gramatical de la instrucción en lenguaje natural es descompuesta en palabras y símbolos de puntuación tomados cada uno de ellos como elementos constituyentes durante la fase de análisis semántico. (p.3)

## GitHub

En nuestro proyecto el servicio de Github cumple un papel muy importante puesto que gracias a éste se pudo mantener el proyecto compartido de forma grupal en todo momento, facilitando la realización del proyecto integrador de una forma ordenada.

Según (B, 2019) GitHub es:

es un sistema de gestión de proyectos y control de versiones de código, así como una plataforma de red social diseñada para desarrolladores. ¿Pero para qué se usa GitHub? Bueno, en general, permite trabajar en colaboración con otras personas de todo el mundo, planificar proyectos y realizar un seguimiento del trabajo.

[GitHub](https://github.com/) es también uno de los [repositorios online más grandes](https://octoverse.github.com/) de trabajo colaborativo en todo el mundo.

Tal como se muestra en la **ilustración 1** se puede observar el repositorio que se utilizó en la plataforma GitHub.

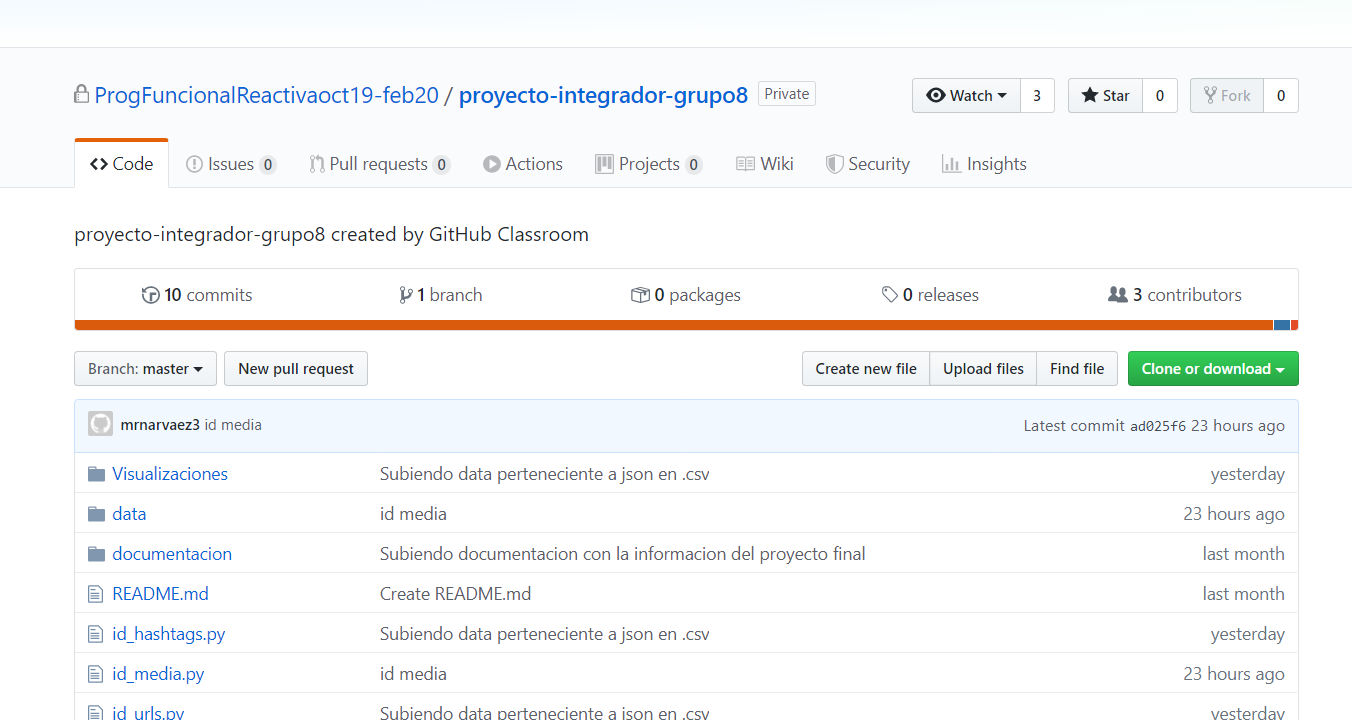


Ilustración 1 Repositorio GitHub

## VEGA – LITE

Según (Satyanarayan, Moritz, Wongsuphasawat, & Heer, 2017):

Los lenguajes de alto nivel también pueden habilitar la búsqueda y la inferencia sobre espacio de visualizaciones. Por ejemplo, Wongsuphasawat et al. [30] introdujo Vega-Lite para impulsar el navegador de visualización Voyager. Por proporcionando un área de superficie más pequeña que el lenguaje Vega de nivel inferior, Vega-Lite hace que la enumeración sistemática y la clasificación de las transformaciones de datos y las codificaciones visuales sean más manejables (p.1)

Es necesario manejar esta herramienta para la visualización de datos especialmente en el caso de tener demasiada información, ya que dispone de cuadros o formas y funciones propias amigables para una buena visualización. Mediante la parte funcional generáramos archivos .csv que podrán ser leídos por vega mediante sus propios scripts en la creación de un .html que muestra las gráficas pedidas.

## ¿Qué es ODS?

(Castillo, Morata, & del Arbol, 2005) mencionan que:

ODS es un sistema que recopila la información existente en los Sistemas Operacionales y la ofrece a los Sistemas Informacionales o directamente a los Usuarios Finales de Negocio. Para ello existe una plataforma común donde conviven tanto la información procedente de los Sistemas Operacionales (Subsistema ODS Replicación) como la información integrada en Sistemas Informacionales (Subsistema ODS Analítico).

## Lenguaje SQL

Según(Rouse, 2015)**:** (Moye, 2015)

SQL son las siglas de (Structured Query Lengugage), es conocido por ser un lenguaje de programación estándar e interactivo para la obtención de información desde una base de datos y para actualizarla. También permite realizar consultas sobre una base de datos, éstas consultas toman la forma de un lenguaje de comandos que permite seleccionar, insertar, actualizar y más. También dispone de una interfaz de programación.

## Librería Pandas

(Moye, 2015) Indique que:

Pandas es una librería de python destinada al análisis de datos, que proporciona unas estructuras de datos flexibles y que permiten trabajar con ellos de forma muy eficiente. Pandas ofrece las siguientes estructuras de datos:

**Series:**Son arrays unidimensionales con indexación (arrays con índice o etiquetados), similar a los diccionarios. Pueden generarse a partir de diccionarios o de listas.

**DataFrame:** Son estructuras de datos similares a las tablas de bases de datos relacionales como SQL.

**Panel, Panel4D y PanelND:** Estas estructuras de datos permiten trabajar con más de dos dimensiones. Dado que es algo complejo y poco utilizado trabajar con arrays de más de dos dimensiones no trataremos los paneles en estos tutoriales de introducción a Pandas.

## Lenguaje Python

Es un lenguaje de programación orientado a objetos y tiene una forma de utilizar muy sencilla. Es de código abierto es decir se lo puede encontrar en su propia web. Este lenguaje es comparable y basado en otro como Perl, Scheme o Java, pero consta de ciertas características que lo han hecho sumamente popular entre los programadores, sobre porque es un lenguaje elegante, limpio y minimalista.

Python tiene la funcionalidad de programar en varios estilos dentro de la llamada programación multiparadigma: estructurada, funcional, orientada a objetos o a aspectos. Sin embargo, sus principales beneficios van más allá de esas posibilidades del desarrollo web. Es un software muy versátil y útil para la automatización de procesos con el fin de ahorrarte complicaciones (Sevilla, 2018).

En nuestro caso que debemos trabajar con grandes volúmenes de datos como es el dataset “ods\_1\_2.csv”, el lenguaje de programación es perfecto ya que el procesamiento y extracción en ese sentido es sumamente efectivo.

## Modelo Entidad Relación

(Sánchez, 2004) manifiesta que:

Una base de datos se puede ver de diferentes formas. Cada programa que accede a la base de datos manipula sólo ciertos datos y estructuras. Así cada programa posee una visión de la base de datos. La unión de todos los datos y sus relaciones forman el llamado esquema conceptual. Mientras que el esquema físico representa el almacenamiento de los datos y sus formas de acceso. (p.12)

En el modelo conceptual las entidades que existen son todas las que se pueden identificar en el ods que utilizamos, y que tras emitir algunos criterios o lógicas de la base de datos pueden pasar a ser más o menos entidades según se considere.

## Modelo Lógico

Según (Sánchez, 2004):

El modelo conceptual es independiente del DBMS que se vaya a utilizar. El lógico depende de un tipo de SGBD en particular. El modelo lógico es más cercano al ordenador, es más cercano al usuario el modelo conceptual, el lógico forma el paso entre el informático y el sistema. (p.16)

En el modelo lógico es evidenciable la cardinalidad de las entidades que quedan tras realizar la técnica de la normalización de datos, se determinan todas las relaciones o recursividad en una misma entidad.

## Modelo Físico

Según (IBM, s.f.):

Un modelo de datos físico es un modelo específico de bases de datos que representa objetos de datos relacionales (por ejemplo, tablas, columnas, claves principales y claves externas) y sus relaciones. Un modelo de datos físico se puede utilizar para generar sentencias DDL que, después, se pueden desplegar en un servidor de base de datos.

## SQL Developer

Según (Fernández, 2008):

SQL Developer es una herramienta grafica gratuita que proporciona Oracle para que no sea necesario utilizar herramientas de terceros, tiene la función de ejecutar consultas o scripts SQL, DML o DDL, sobre base de datos Oracle. Utiliza una apariencia y funcionalidad similar a otras herramientas de este tipo.

Además, en las últimas versiones ha incorporado mejoras como permitir conectar con bases de datos no Oracle, como SQLServer, MySQL o Access. La conexión con *MySQL* o *SQLServer* se realiza a través de **JDBC**, y de manera bastante sencilla.

Una vez se establece la conexión se pueden explorar los objetos de las bases de datos como si se tratase de una de Oracle, seguidamente permite ejecutar sobre estas bases de datos sentencias SQL, sin embargo, en cuanto a funcionalidades mas avanzadas como la creación de estructuras de este tipo de conexión se llega a limitar.

# Solución

**Análisis de datos**

El análisis de datos se encarga de examinar un conjunto de datos con el objetivo de obtener conclusiones sobre la información para poder tomar decisiones o simplemente ampliar los conocimientos sobre diversos temas de interés social.

Hoy en día muchas industrias y empresas de alto prestigio usan el análisis de datos para sacar conclusiones y decidir acciones a implementar siempre pensando en su beneficio propio, Cabe mencionar que la ciencia también usa el análisis de datos para comprobar o descartar teorías o modelos existentes.

Según lo analizado en los datos que hemos tratado para el proyecto decidimos importarlos desde CSV como origen de archivo el de (UNICODE UTF-8) tal como se muestra en la **ilustración 2** se puede visualizar caracteres en el campo de tweets y también cantidades numéricas bastante grandes.

Además, el dominio y tipos de datos decidimos que hay tweets con muchos caracteres especiales y debemos de utilizar tipos de datos que acepten dichos caracteres.

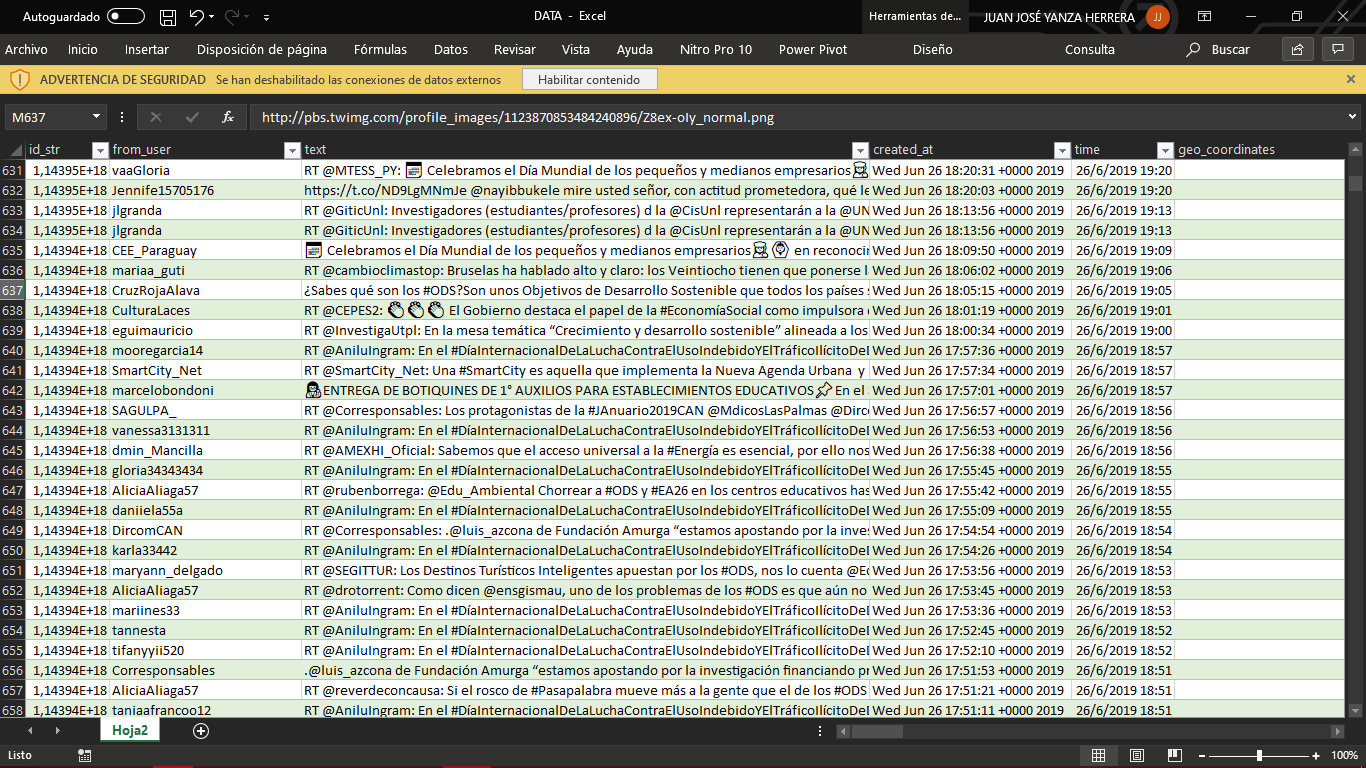


Ilustración 2: Dataset proyecto

Tipo de datos definidos en el dataset:

|  |  |  |
| --- | --- | --- |
| **Columna** | **Tipo** | **Descripción** |
| Id\_str | String | Representa al identificador único de un Tweet |
| From\_user | String | El nombre del usuario, tal como lo ha definido |
| Text | String | El texto, en UTF-8, de la publicación. |
| Created\_at | String | Fecha UTC cuando se creó este Tweet |
| Time | DateTime | Fecha |
| Geo coordinates | String | Puede ser nulo. Representa la ubicación geográfica de este Tweet según lo informado por el usuario o la aplicación del cliente. |
| User\_lang | String | Idioma con el que el usuario configuró su cuenta. |
| In\_reply\_to\_user\_id\_str | String | Puede ser nulo. Si el Tweet es una respuesta, este campo contendrá la el ID del autor del Tweet original. Esto no siempre será necesariamente el usuario mencionado directamente en el Tweet. |
| In\_reply\_to\_screen\_name | String | Puede ser nulo. Si el Tweet es una respuesta, este campo contendrá el nombre de pantalla del autor original del Tweet |
| from\_user\_id\_str | String | La cadena representa al identificador único para este usuario |
| in\_reply\_to\_status\_id\_str | String | Puede ser nulo. Si el Tweet es una respuesta, este campo contendrá la representación de cadena del ID del Tweet original. |
| Source | String | Aplicación utilizada para publicar el Tweet.  (Formato HTML) |
| profile\_image\_ur | String | URL que apunta a la imagen de perfil del usuario |
| user\_followers\_count | Int | El número de seguidores que esta cuenta tiene actualmente. Bajo ciertas condiciones de coacción, este campo indicará temporalmente "0" |
| user\_friends\_count | Int | El número de usuarios que esta cuenta está siguiendo (también conocido como sus "seguidores"). Bajo ciertas condiciones de coacción, este campo indicará temporalmente "0" |
| user\_location | String | Puede ser nulo. La ubicación definida por el usuario en el perfil de su cuenta. |
| status\_url | String | URL que apunta al tweet original |
| entities\_str | String | Entidades que se han analizado del texto del  Tweet. (Formato JSON). Mayor información la pueden [encontrar aquí: https:// developer.twitter.com/ en/docs/tweets/datadictionary/overview/ entities-object](https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/entities-object) |

Tabla 1 Tipo de datos

**Base de Datos de tweets relacionados a ODS**

Dado un ods es un caso de aplicación en el cual se establecen relaciones de tweets y usuarios en donde una depende de otra en este caso tweet depende de la existencia de usuario, en donde podemos determinar que tweet es una entidad débil.

El ODS consta de varias columnas de las cuales se puede abstraer información para la elaboración de un modelado de base de datos. Primeramente, determinando los elementos de la vida real tal como son los usuarios y tweets siendo estas las primeras entidades en ser determinadas, luego se abstraen entidades que pueden ser reales o también abstractas tales como **entities\_str** en donde se encuentra un JSON con aún más información en cuyas *keys* de dicho diccionario se toma como entidades propias de un tweet o un usuario. Una vez identificadas las identidades queda por definir los atributos es decir el contenido de cada entidad.

Contienen información específica los cuales pueden ser de distintos tipos como fecha, numéricos o texto. Un ejemplo de estos es: *time*, *text*, *user\_followers\_count* cuyo tipo de datos son fecha, texto y numérico respectivamente.

**Descripción de Entidades**

**Usuario:** cada usuario genera uno o varios tweets, esta entidad se identifica por el atributo **from\_user\_id\_str** consta de varios elementos propios tales como seguidores y seguidos, localización del usuario, y la url de la foto de perfil del usuario. Según el id de cada usuario se puede encontrar un tweet diferente que puede ser una respuesta o un RT.

**Tweet:** es la entidad principal la cual se la identifica por el atributo **id\_str**, tiene una dependencia de la entidad usuario ya que si no existen usuarios no se pueden dar un tweet. Tweet tiene varias entidades las cuales depende de esta las cuales son media – hashtag – menciones – url que se encuentran en entities\_str en formato JSON. Dicha información contiene el texto del tweet y los hashtags utilizados en cada uno de ellos. Además de contener enlaces o URLs de archivos multimedia, de fotos de perfil de un usuario entre otros datos que deben de estar en cada entidad correspondiente.

**Media:** es comúnmente usado en tweets para compartir videos, imágenes, enlaces externos (tipo media), al igual que otras entidades los cuales se almacenan en la misma (URL\_MEDIA) contiene datos como urls y el tipo de contenido que son las urls como “photo”, y urls exclusivos de twitter como el display\_url URL visible que ayuda a los usuarios a anticipar el contenido de la página de destino a la que se dirigen, el expanded\_ url es el enriquecimiento de URL ampliado y mejorado expande automáticamente los URL acortados que se incluyen en el cuerpo de un Tweet.

**Hashtag:** una herramienta muy usada de twitter tiene la función de entrar a una categoría mediante una palabra clave seguida del carácter **#**. Al dar click sobre esta palabra sin espacios redirecciona a una lista de tweets con la misma temática es un identificador en específico para filtrar contenido de interés lo cual puede ser útil contener hashtags para cada tweet único.

**Url:** en un tweet es posible colocar direcciones web por los usuarios, estas urls redireccionan a otra página web y se identifican con url\_id cada url tiene un enlace único que redirecciona a distintos tipos de contenido tales como fotos de perfil en caso de los tweets, o urls referenciales.

**Menciones:** Una mención es cualquier Tweet que contenga el nombre de usuario de otra persona en alguna parte del cuerpo del Tweet, está identificada por **id\_mentions**. Estos mensajes, así como todas tus respuestas, se reúnen en la pestaña de Notificaciones.

Considerando las siguientes relaciones del modelo conceptual se tiene que los tweets pueden ser respuestas o retweets por lo tanto se produce una recursividad o reflexividad en la entidad tweet. Tal como se muestra en la **ilustración 3**.

Imagen que contiene objeto, reloj, sostener, hombre

Descripción generada automáticamente

Ilustración 3- Modelo Conceptual

Imagen que contiene texto, mapa

Descripción generada automáticamente

Además, en las relaciones binarias 1..1 con 1..\* se replica la llave primaria como llave foránea en la parte de 1 a \* representando a la clave primaria de la parte 1..1. dado el caso que en este modelo no se encuentran relaciones de muchos a muchos no es necesario crear una entidad intermedia. De acuerdo a la **ilustración 4**.

Una captura de pantalla de una red social

Descripción generada automáticamente

Ilustración 4-Modelo Lógico

**Diccionario de datos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TWEET** | | | | |
| **PK/FK** | **Atributo** | **Tipo de dato** | **Tamaño** | **Descripción** |
| Llave primaria | id\_str | NUMERIC | 60 | Representa al identificador único de un Tweet |
|  | text | Nvarchar | 5000 | El texto, en UTF-8, de la publicación. |
|  | created\_at | Varchar2 | 40 | Fecha UTC cuando se creó este Tweet. |
|  | geo\_cordinates | Varchar2 | 20 | Puede ser nulo. Representa la ubicación geográfica de este Tweet según lo informado por el usuario o la aplicación del cliente. |
|  | in\_reply\_to\_screen\_name | Varchar2 | 30 | Puede ser nulo. Si el Tweet es una respuesta, este campo contendrá el nombre de pantalla del autor original del Tweet |
|  | in\_reply\_to\_status\_id\_str | Varchar2 | 20 | Puede ser nulo. Si el Tweet es una respuesta, este campo contendrá la representación de cadena del ID del Tweet original. |
|  | statuts\_url | Varchar2 | 20 | URL que apunta al tweet original |
| Llave foránea | From\_user\_id\_str1 | Varchar2 | 20 | El nombre del usuario, tal como lo ha definido |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **USUARIO** | | | | |
| **PK/FK** | **Atributo** | **Tipo de dato** | **Tamaño** | **Descripción** |
|  | from\_user | Varchar2 | 60 | El nombre del usuario, tal como lo ha definido |
|  | user\_lang | Varchar2 | 4 | Idioma con el que el usuario configuró su cuenta. |
| Llave primaria | from\_user\_id\_str | Numeric | 60 | La cadena representa al identificador único para este usuario. |
|  | profile\_image\_url | Varchar2 | 60 | URL que apunta a la imagen de perfil del usuario. |
|  | user\_followers\_count | Numeric | 1000 | El número de seguidores que esta cuenta tiene actualmente. Bajo ciertas condiciones de coacción, este campo indicará temporalmente "0". |
|  | user\_friends\_count | Numeric | 1000 | El número de usuarios que esta cuenta está siguiendo (también conocido como sus "seguidores"). |
|  | user\_location | Varchar2 | 60 | Puede ser nulo. La ubicación definida por el usuario en el perfil de su cuenta. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **URL** | | | | |
| **PK/FK** | **Atributo** | **Tipo de dato** | **Tamaño** | **Descripción** |
| Llave primaria | url\_id | Numeric | 20 | Código con el que se identifica la tabla URL |
| Llave foránea | Id\_str4 | Numeric | 60 | Representa al identificador único de un Tweet |
|  | source | Varchar2 | 40 | Aplicación utilizada para publicar el Tweet.  (Formato HTML) |
|  | profile\_image\_url | Varchar2 | 60 | URL que apunta a la imagen de perfil del usuario |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MENCIONES** | | | | |
| **PK/FK** | **Atributo** | **Tipo de dato** | **Tamaño** | **Descripción** |
| Llave primaria | Id\_mentions | Numeric | 60 | Identificador de las menciones |
| Llave foránea | id\_str3 | Varchar2 | 60 | Representa al identificador único de un Tweet. |
|  | Screen\_name | Varchar2 | 20 | Identificador de la pantalla del nombre. |
|  | name | Varchar2 | 20 | Nombre propio de la mención. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **HASHTAG** | | | | |
| **PK/FK** | **Atributo** | **Tipo de dato** | **Tamaño** | **Descripción** |
| Llave foránea | id\_str2 | Numeric | 60 | Representa al identificador único de un Tweet. |
|  | text\_h | NVarchar | 500 | El texto, en UTF-8, de la etiqueta hashtag. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MEDIA** | | | | |
| **PK/FK** | **Atributo** | **Tipo de dato** | **Tamaño** | **Descripción** |
| Llave foránea | id\_str1 | Numeric | 60 | Código con el que se identifica la tabla URL |
|  | media\_url | Varchar2 | 70 | URL que apunta al dato tipo(media). |
|  | display\_url | Varchar2 | 60 | La URL visible es la URL que los usuarios de motores de búsqueda ven en un anuncio PPC. |
|  | type | Varchar2 | 40 | Indica el tipo de archivo que es; imagen, video, etc… |
|  | expanded\_url | NVarchar | 150 | Código URL completo. |

Tabla 2 - Diccionario de datos

**Limpieza de datos mediante programación funcional.**

Mediante análisis de cada columna del ods determinamos su tipo de dato y su dominio, posteriormente usando programación funcional importando la librería pandas para manejo de datos y visualizaciones comenzamos a buscar valores nulos en cada columna y como observaciones se tiene que las columnas **text, from\_user y id\_str** no contenían ninguna fila o registro nulo.

Para las otras 15 columnas se identificaron los valores nulos y se los reemplazó por strings vacíos correspondientemente con su tipo de dato, a los datos enteros se los reemplazó por 0. A los tipos de dato fecha se importó la librería “datetime” para a los registros nulos asignarles la fecha del ordenador o del sistema para lo cual no existirá inconveniente con la integridad de los datos. Tal como se muestra en la ***ilustración 5***.

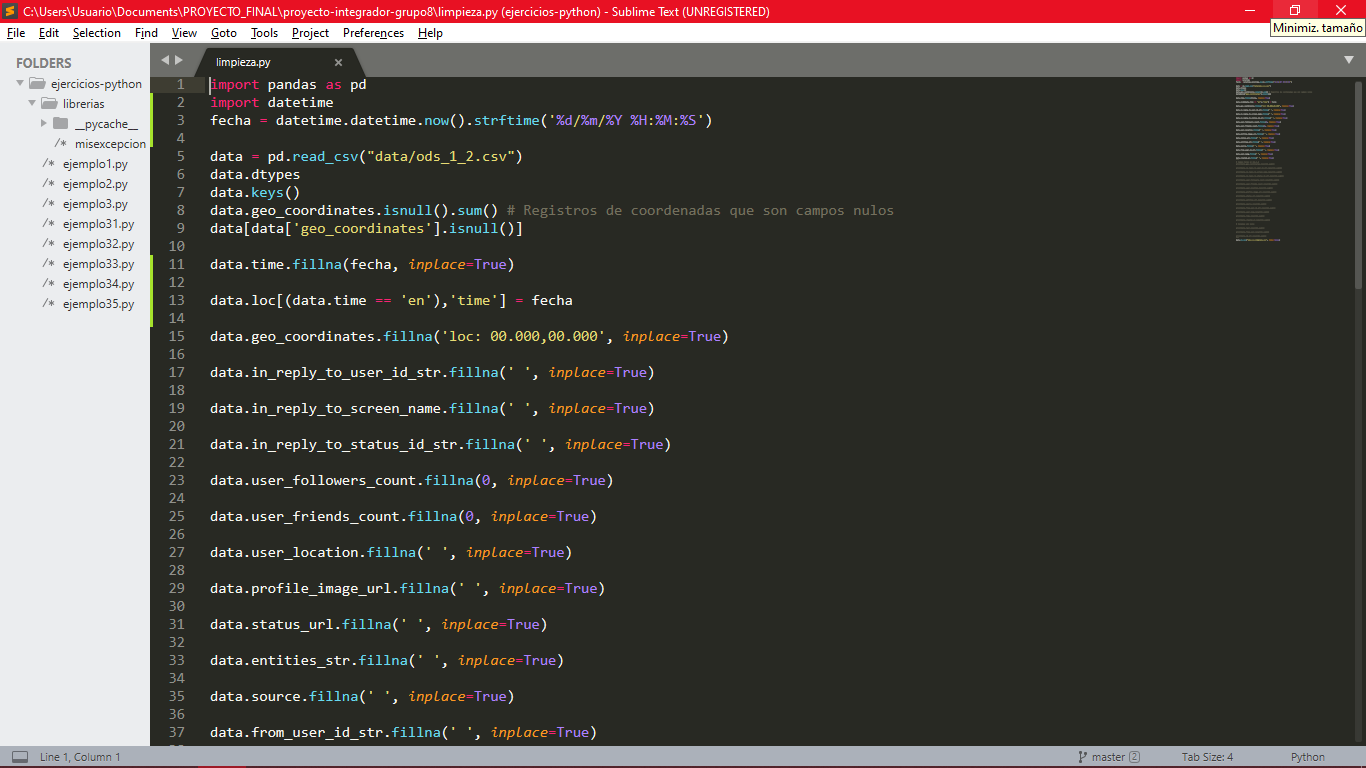


Ilustración 5 Limpieza ods\_1\_2

**Visualizaciones Tweets por día y por hora**

Para generar las visualizaciones generamos un .csv usando programación funcional de mano de las librerías datetime, collections y pandas. Se abrió el dataset limpio el mismo que fue transformado en diccionario para poder implementar map y separar las fechas usando map en la posición de las fechas. Como se presenta en la **ilustración 6** y la visualización de tweets por día y hora en las **ilustraciones** **7** y **8.**

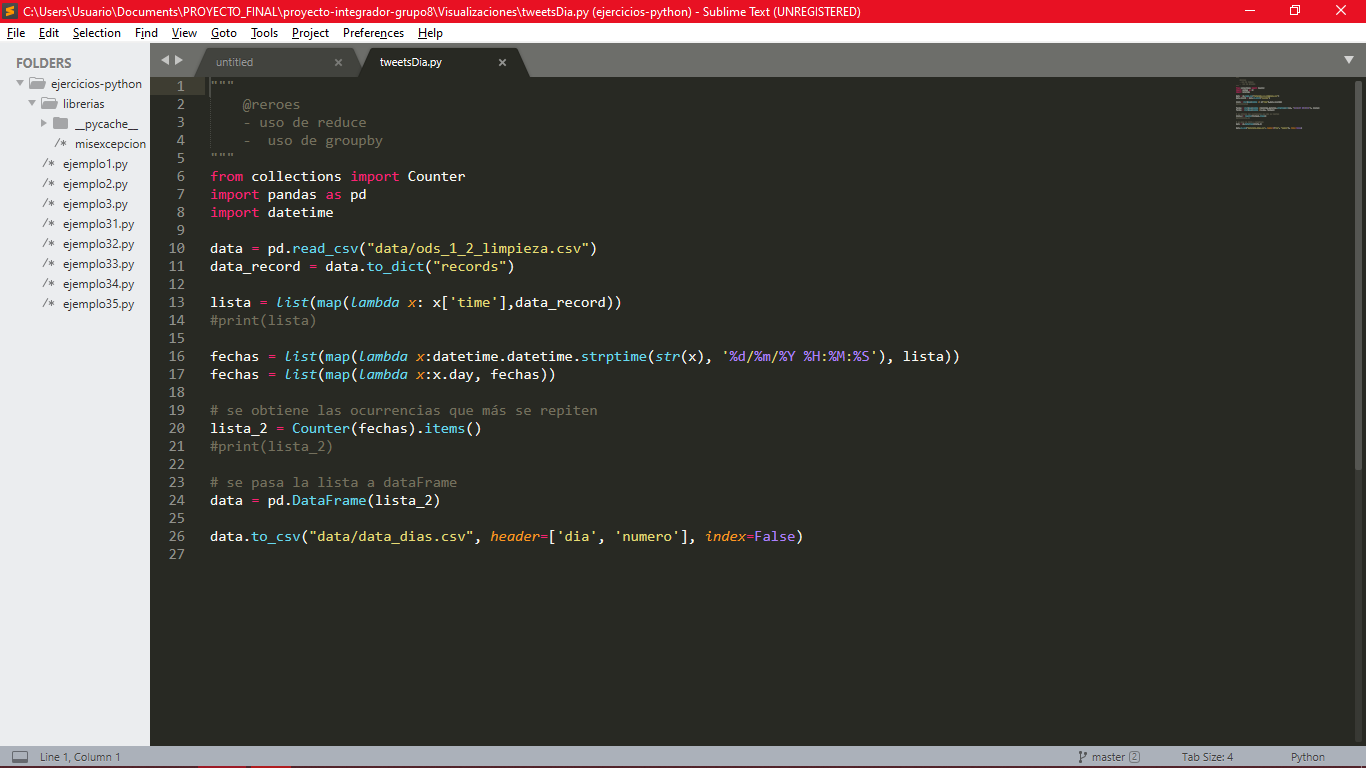


Ilustración 6 Generar csv – días



Ilustración 7 Publicaciones por día



Ilustración 8 Publicaciones por hora

**Normalización de datos**

La normalización de datos se lo conoce como el proceso de organizar los datos de una base de datos. En este proceso se incluye la creación de tablas y el establecimiento de relaciones entre ellas, bajo reglas diseñadas tanto para proteger los datos como para hacer que la base de datos sea más flexible y comprensible al eliminar la redundancia y las dependencias que no tienen mucha coherencia.

Los datos que se consideran redundantes desperdician el espacio de disco y crean problemas de mantenimiento, por ejemplo, si se deben cambiar datos que existen en muchos lugares, se deben cambiar de la misma forma exactamente en todas sus ubicaciones.

# Validación

# Conclusiones

# Bibliografía

Castillo, R., Morata, J., & del Arbol, L. (2005). Operational Data Store (ODS) - 933.pdf. *Actas Del III Taller Nacional de Minería de Datos y Aprendizaje*, 359–365. Retrieved from http://www.lsi.us.es/redmidas/CEDI/papers/933.pdf

Corzo, J. M. S., Portillo, D. D., & Zamora, J. A. C. (2006). Sistema de consultas en lenguaje natural para bases de datos. *CEUR Workshop Proceedings*, *220*.

Hoyos, J. G., & Puertas, A. (2017). Desempeño de los paradigmas funcional y orientado a objetos en arquitecturas multicore: Caso de estudio haskell-Java. *Informacion Tecnologica*, *28*(5), 65–74. https://doi.org/10.4067/s0718-07642017000500009

Sánchez, J. (2004). Diseño Conceptual de Bases de Datos guía de aprendizaje. *Creative Commons*, 1–25.

Satyanarayan, A., Moritz, D., Wongsuphasawat, K., & Heer, J. (2017). Vega-Lite: A Grammar of Interactive Graphics. *IEEE Transactions on Visualization and Computer Graphics*, *23*(1), 341–350. https://doi.org/10.1109/TVCG.2016.2599030

IBM. (s.f.). IBM. Obtenido de IBM®: https://www.ibm.com/support/knowledgecenter/es/SSRTLW\_9.6.0/com.ibm.datatools.core.ui.doc/topics/cphysmod.html

# Bibliografía

B, G. (13 de mayo de 2019). *Hostinger*. Obtenido de Hostinger: https://www.hostinger.es/tutoriales/que-es-github/

Fernández, C. (30 de Abril de 2008). *Dataprix*. Obtenido de Dataprix: https://www.dataprix.com/es/articulo/oracle-sql-developer

IBM. (s.f.). *IBM*. Obtenido de IBM®: https://www.ibm.com/support/knowledgecenter/es/SSRTLW\_9.6.0/com.ibm.datatools.core.ui.doc/topics/cphysmod.html

Moye, R. (30 de Octubre de 2015). *Jarroba.com*. Obtenido de Jarroba.com: https://jarroba.com/pandas-python-ejemplos-parte-i-introduccion/

Rouse, M. (Enero de 2015). *TechTarget*. Obtenido de https://searchdatacenter.techtarget.com/es/definicion/SQL-o-lenguaje-de-consultas-estructuradas

Sevilla, P. (07 de Agosto de 2018). *Intium*. Obtenido de Intium: https://initiumsoft.com/blog/que-es-el-lenguaje-de-programacion-python-y-para-que-sirve/

WebEmpresa. (01 de Marzo de 2018). *Webempresa.com*. Obtenido de Webempresa.com: https://www.webempresa.com/blog/que-es-twitter-como-funciona.html

# Anexos

1. <http://classroom.github.com>). [↑](#footnote-ref-2)