

Universidad Internacional de La Rioja

Facultad de Ingeniería y Tecnología

# Máster Universitario en Análisis y Visualización de Datos Masivos / Visual Analitics & Big Data

Actividad HDFS, Spark SQL y MLlib

|  |  |
| --- | --- |
| Actividad de estudio presentado por: | Juan David Escobar Escobar |
| Tipo de trabajo: | Actividad |
| Modalidad: | Individual |
| Profesor/a: | Yamila García Martínez |
| Fecha: | Enero 2022 |

Índice de contenidos

[Máster Universitario en Análisis y Visualización de Datos Masivos / Visual Analitics & Big Data 1](#_Toc90307828)

[1. Introducción 8](#_Toc90307829)

[2. Metodología 9](#_Toc90307830)

[3. Descripción problemas 13](#_Toc90307831)

[4. Formato JSON propuesto 18](#_Toc90307832)

[5. Propuestas mejoras del conjunto de datos 19](#_Toc90307833)

[6. Conclusiones 21](#_Toc90307834)

[7. Referencias bibliográficas 22](#_Toc90307835)

Índice de tablas

[**Table 1.** Muestra de datos valores duplicados por CrimeId. 9](#_Toc90310023)

[**Table 2.** Muestra de datos columnas redundantes. 10](#_Toc90310024)

[**Table 3.** Muestra de datos valores atípicos columna AddressType. 11](#_Toc90310025)

[**Table 4.** Muestra de datos valores atípicos columna City. 11](#_Toc90310026)

[**Table 5.** Estadística valores Missing columnas Range, City y State. 12](#_Toc90310027)

[**Table 6.** Muestra de datos valores atípicos columna AddressType. 13](#_Toc90310028)

[**Table 7.** Valores atípicos columna Disposition. 13](#_Toc90310029)

[**Table 8.** Error de almacenamiento columna derecha. 14](#_Toc90310030)

# Planteamiento del problema

# 

# Graphical user interface, text Description automatically generated

# Formulación del problema

# Problema general

# ¿Cómo se relaciona el COVID-19 con la educación, el estilo de vida social y la salud mental de los estudiantes, en la India?

# Problema especifico

# ¿Cómo impacta el COVID-19 con la educación, el estilo de vida social y la salud mental de los estudiantes?

# Formulación de hipótesis

# El COVID-19 se relaciona significativamente con el impacto en los cambios de estilo de vida, creación de nuevas formas educativas y la afectación de la salud mental de los estudiantes y las personas en la India.

# Objetivo general

# Determinar como se relaciona la COVID-19 con la educación, el estilo de vida social y la salud mental de los estudiantes, en la India.

# Marco Metodológico

# Variables

# Graphical user interface, application Description automatically generated

# Metodología

# El método utilizado en la presente investigación es cuantitativo observacional (Encuesta 19 variables Liker scale).

# Sub grupos de la encuesta

# Tipo de estudio

# Básico descriptivo correlacional. (Experimental longitudinal??)

# Población, muestra y muestreo

# Población

# La población esta constituida por 1182 individuos de diferentes edades y centros de educativos en la capital de India Delhi.

# Métodos de análisis de datos

# Análisis descriptivos,

# Para hallar las tablas, graficas, medidas de tendencia central, dispersión y análisis de datos se utilizo el Software R-Studio.

# Análisis inferencial,

# Para la prueba de hipótesis se utilizo Chi cuadrado utilizando el software R-Studio??

# Tablas, graficas, métricas e interpretación

# Considerar las preguntas de la encuesta

# Table Description automatically generated

# Chart Description automatically generated

# Interpretación de las tablas y las graficas

# Text Description automatically generated with medium confidence

# Chart, bar chart Description automatically generated

# Análisis discusión de resultados

# Se hace una comparación entre los resultados y el problema planteado

# Conclusiones

# En función a los objetivos formulados

# Recomendaciones

# En función a los objetivos formulados

# Anexos

# En función a los objetivos formulados

# Encuesta

# En función a los objetivos formulados

# Base de datos

# En función a los objetivos formulados

# Bibliografía

https://www.youtube.com/watch?v=CuKr7GzohbI

https://www.youtube.com/watch?v=VWOvYPTrrNA

https://datos.gob.es/es/documentacion/guia-

<https://datos.gob.es/es/documentacion/guia-practica-de-introduccion-al-analisis-exploratorio-de-datos>

<https://www.youtube.com/watch?v=CIIzowIm9ZQ>

<https://miasesordetesis.com/como-identificar-y-definir-las-dimensiones-de-las-variables/>

<https://www.youtube.com/watch?v=788WtRruUlU>

<https://www.researchgate.net/publication/347935769_COVID-19_and_its_impact_on_education_social_life_and_mental_health_of_students_A_Survey>

<https://www.kaggle.com/kunal28chaturvedi/covid19-and-its-impact-on-students>

Con este proceso se pretende que el consumidor aprenda y reutilice las técnicas propuestas, mediante un ejercicio practico. El ejercicio de este paso a paso detalla herramientas libres de uso, y un conjunto de datos de uso publico adjunto en el siguiente repositorio de [GitHub](https://github.com/juadaves91/bd-bigdata-unir) almacenado en formato CSV.

Para llevar a cabo el proceso de limpieza se opto por utilizar las herramientas que se usan con frecuencia por los analista y científicos de datos, para la recolección, procesamiento y explotación de datos, a continuación, se listas las herramientas y librerías utilizadas:

**Herramientas**: Jupyter Nootebooks Versión 6.4.2 (<https://jupyter.org/>), Python versión 3.0 (<https://www.python.org/>), MongoDB (<https://www.mongodb.com/>), Visual Studio Code (<https://code.visualstudio.com/>), Anaconda Studio (<https://www.anaconda.com/>), Spark versión 3.2.0 (<https://spark.apache.org/>, Atom (<https://atom.io/>).

**Librerías y Módulos**:

* Os (<https://docs.python.org/3/library/os.html>).
* Chardet (<https://chardet.readthedocs.io/en/latest/>).
* Pandas (<https://pandas.pydata.org/>).
* Matplotlib (<https://matplotlib.org/>).
* Seaborn (<https://seaborn.pydata.org/>).
* Numpy (<https://numpy.org/>).
* Datetime (<https://docs.python.org/3/library/datetime.html>).
* Re (<https://docs.python.org/3/library/re.html>).
* Pyspark,Pyspark.sql y Pyspark.types (<https://spark.apache.org/docs/latest/api/python/>)
* Json (<https://docs.python.org/3/library/json.html>),
* Pymongo y MongoClient (<https://pymongo.readthedocs.io/en/stable/>).

**Artefactos**:

* Dataset original (CSV) (<https://github.com/juadaves91/bd-bigdata-unir/blob/main/Limpieza%20Datos%20Dataset%20-%20CSV.pdf>).
* Dataset Limpio (JSON) (<https://raw.githubusercontent.com/juadaves91/bd-bigdata-unir/main/ESCOBAR_ESCOBAR_JUAN_DAVID.json>).
* Notebook (Limpieza de un Dataset - CSV) (<https://github.com/juadaves91/bd-bigdata-unir/blob/main/Limpieza%20Datos%20Dataset%20-%20CSV.pdf>) .
* Documento paso a paso (<https://github.com/juadaves91/bd-bigdata-unir/blob/main/Paso%20a%20paso%20limpieza%20de%20un%20Dataset.pdf>).
  1. Fases proceso de limpieza

**Fase1. Interpretación de datos crudos (Raw Zone):** Se les llama datos crudos a aquellos archivos, o fuentes de datos originales, a los cuales les vamos a hacer un trabajo de limpieza, para llevar a cabo este primer paso se tienen definidas algunas funciones, cuyo principal objetivo es la lectura de los datos según el formato en el cual se encuentre la información, por ejemplo (JSON, AVRO, ORC, TXT, CSV, BINARIO, PARQUET o conexión directa a una Base de datos), para este ejemplo se trabaja con un archivo en formato CSV, a continuación se detallan los pasos que se encuentran asociados a esta primera fase:

1. **Definir formato de codificación del archivo:** Es importante establecer un tipo de codificación con el cual se va a trabajar el archivo fuente, nuestro Dataset original **data\_act\_01.rtf**, contiene datos en el idioma ingles, cada archivo será codificado en UTF-8 (Unicode Transformation Format), ya que cuenta con 135.000 caracteres y esto nos genera tranquilidad de poder incluir caracteres especiales como por ejemplo la letra Ñ y la tildé.
2. Read\_csv():Este método pre construido se encarga de leer el Dataset original **data\_act\_01.rtf,** de acuerdo a las características del mismo y en codificación UTF-8 is\_valid\_encoding\_csv().
3. **Análisis y descripción de los datos:** Procedemos a validar la calidad de la información que tenemos, a través del método df\_pandas.limit(4), **damos un primer vistazo a nuestros datos, posterior a ello utilizamos el método** df\_pd.info(),el cual nos indicara la cantidad de filas, columnas, registros leídos, nulos, tipos de datos y la memoria ocupada por el objeto que almacena la los datos Dataframe, esta dos funciones nos dan una visión general sobre las características, esquema, correlación de los tipos de datos inferidos y calidad de los datos.

Para continuar analizando nuestro conjunto de datos utilizamos la función pd\_df.describe()el cual nos muestra las siguientes métricas (conteo, valores únicos, frecuencia, ultimo valor, primer valor, mediana, desviación estándar, valor mínimo, valor máximo y representación del 25%, 50% y 75% de los datos).

El siguiente paso será graficar un histograma para cada una de las variables numéricas, con la finalidad de identificar sesgos de datos en los valores, y como se distribuyen los datos, este paso se recomienda ejecutar cada vez que se haga una transformación o limpieza en el Dataset

**Fase 2. Datos procesados (Processed Zone):** En este paso ya tenemos la información almacenada en un objeto en memoria llamado Dataframe, a través de los pasos anteriormente descritos, la zona de datos procesados es una zona intermedia entre la lectura y el almacenamiento o formateo resultante de los datos, en esta fase realizaremos algunos ajustes y transformaciones para tener una aproximación de los datos limpios y formateados como es requerido. En esta fase es importante documentar o registrar un log de los cambios efectuados sobre los datos, para llevar una trazabilidad de estos y poder consultarla cuando se requiera.

1. **Validación de duplicados:** Se procede con la ejecución del comando df\_pd = df\_pd.drop\_duplicates() para proceder con la eliminación de registros duplicados, adicional si identificamos un campo identificador de fila, procedemos a validar los registros duplicados asociados al mismo Identificador, mediante la función list\_duplicates = get\_duplicates(df, df\_pk), luego procedemos a eliminar aquellos que sean exactamente iguales, en el caso que no lo sea, se validan formatos, sintaxis o posibles errores en los datos provenientes del archivo fuente.
2. **Ajuste de tipos de variables:** En este paso procedemos a validar cada variable con el tipo de valor correspondiente, es decir las variables numéricas y cualitativas. El siguiente paso será la validación de formatos de fecha, patrones de texto, tiempo, moneda, entre otros, para realizar estas validaciones podemos apoyarnos en expresiones regulares, librerías para el manejo de fechas y tiempo como lo es Datetime, algunas de las funciones definidas en nuestro script son validate\_format\_timestamp (df, "DateColumn", col\_format = "yyyy-mm-dd") y validate\_format\_text (df, “ColName”, reg = r'^[A-Z]{3}').
3. **Estandarización de datos:** Definir un formato estándar para todos los campos tipo fecha, esto implica una trasformación para cada columna tipo fecha, pero nos ayudará en procesos posteriores, idealmente el formato de fecha se debe ajustar al tipo por defecto que utiliza la fuente destino. Otra buena practica es que todos los valores tipo texto los convertiremos en mayúsculas o minúsculas, excepto si se identifica un formato estándar y con calidad, por último eliminaremos los espacios en blanco a la derecha e izquierda con la función trimp().
4. **Detección y tratamiento de datos ausentes:** Para esta labor se creo una función llamadacount\_missings(df), la cual nos retorna los valores perdidos de las columnas del Dataset. Para aquellas columnas tipo numéricas que tienen mas de un 50% de los datos perdidos, lo recomendado es eliminarlas, para las columnas tipo texto podemos rellenar los espacios vacíos con un valor Dummy definido, como por ejemplo el valor “-99”.
5. **Identificación de datos atípicos:** Aquellos valores que no concuerdan con la descripción o el tipo de dato de una variable pueden afectar la calidad de los datos, tareas de análisis y pueden considerarse como datos anómalos. Lo que haremos será detectarlos para disminuir el daño en fases posteriores, para llevar a cabo la detección se identifican y se agrupan las variables según su tipología numéricas y categóricas, por medio de la función distinct() y la validación de la distribución de los datos mediante gráficos de histogramas, bigotes, como draw\_hist\_num\_vars(list\_var\_num) y draw\_hist\_cat\_vars(list\_var\_cat), dichas funciones nos permiten tener una visión mas clara de los valores almacenados en cada una de las variables, a continuación se detalla una descripción de los valores atípicos encontrados para cada variable.
6. **Redefinir el esquema:** Para concluir nuestra fase datos procesados, se creará una copia del Dataframe original, el cual almaceno los datos con un esquema inferido de forma automática, para el nuevo Dataframe le especificaremos un nuevo esquema ajustado a la naturaleza de los datos y variables asociadas, ya que contamos con la tranquilidad de que validamos los tipos de datos y formatos de estos en pasos previos, en este paso se ignoraran las variables redundantes. Ahora ya tenemos un Dataset en memoria listo para ser formateado y almacenado en la siguiente fase.

**Fase 3. Datos resultados (Result Zone):** En este paso ya tenemos la información almacenada en un objeto en memoria llamado Dataframe limpio y transformado, listo para materializarse en un formato determinado y usarse, para finalizar nuestra ultima etapa de limpieza de datos vamos a ejecutar los pasos de análisis y correlación de datos, formateo del Dataset y almacenamiento en una base de datos tipo MongoDB.

1. **Correlación de variables:** En esta etapa se determina la relación lineal entre dos o mas variables, mediante la correlación encontrada la cual fluctúa entre 1 y -1, cuando la correlación tiende a cero quiere decir que los valores son independientes y no se relacionan directamente, para analizar la correlación creamos un Dataframe que contenga únicamente las variables numéricas, posteriormente utilizamos la función df\_correl.corr() y lo graficamos mediante un grafico de calor para analizar la correlación de los valores de cada variable seaborn.heatmap()
2. **Formato y conversión Dataset resultante:** El Dataset resultante será convertido a un formato JSON, para ello utilizaremos la función df.toJSON() y lo almacenaremos en un nuevo objeto Dataframe el cual será convertido a una lista o vector para adicionar dos objetos o diccionarios con las columnas y filas resultantes de la limpieza , el resultado será un objeto JSON con pares llave valor, se debe evaluar que tipo de relaciones pueden existir al interior del JSON uno a uno, uno a muchos, muchos a muchos o muchos a uno, ya que esto nos dará un indicio de como estructurar el JSON.
3. **Almacenamiento en fuente de datos:** Para nuestro ejemplo vamos a generar una conexión a nuestra BD local de MongoDB, luego crearemos una nueva base de datos a la cual le insertaremos una nueva colección o tabla, la cual es la lista o vector que contiene las dos primeras filas o documentos de nuestro Dataset.

# Descripción problemas

**Problema 1. Identificadores duplicados por columna CrimeId**

**¿Qué o cual es el problema?** Se identifican valores duplicadosen la columna CrimeId, la cual, por su nombre, tipo de dato nos da a entender que es el valor llave primaria del conjunto de datos, aún así no tenemos la certeza total de que este campo sea único, para esta variable se presentan valores duplicados para **160950496** y **160913455**. El problema es que no podemos identificar cual es el registro único ya que el resto de las columnas almacena información diferente, lo cual pudo haberse almacenado de manera incorrecta.

**¿Cómo se ha detectado el problema?** Uno de los pasos incorporados a la metodología para limpiar un Dataset es la validación de registros duplicados por todos los valores o combinaciones posibles asociados a las columnas y por llave primaria, la forma en como se ha encontrado dicho problema se encuentra descrito en la Fase 2 de la metodología punto numero 4 del paso a paso.

**¿Por qué crees que se trata de un problema?** Al analizar la estructura de la información podemos identificar que proviene de una tabla de una base de datos SQL, normalizada, acompañada de algunos campos redundantes, que quizá fueron agregados mediante la materialización de una vista, al tener llaves primarias duplicadas, podemos inferir que la información asociada a cada registro puede estar duplicada o no estar consistente.

**Ejemplo extraído del Dataset:**

**Tabla 1.** Muestra de datos valores duplicados por CrimeId.

**Table

Description automatically generated**

**Forma de solucionarlo:** Una de las maneras de solucionar el problema es yendo a la fuente de datos y validar con el dueño de la información proveída al equipo de datos cual es la información correcta acompañado de las correctivas validaciones de lo que pudo haber sucedido al momento de generar el Dataset original “archivo fuente CSV”. Otra de las maneras de solucionarlo es crear una columna calculada llamada llave primaria compuesta, la cual podría ser la concatenación del campo (CrimeId + CallDateTime8) y así tener un identificador mas detallado por cada registro, es importante contar con un documento diccionario de datos que nos describa el esquema y condiciones de cada campo de la fuente de datos.

**Problema 2. Hallazgo columnas redundantes OffeceDate, CallTime**

**¿Qué o cual es el problema?** Las columnas OffeceDate y CallTime contienen la información asociada a la columna CallDateTime, por lo cual se consideran como redundantes, esto puede ocupar espacio de almacenamiento innecesario.

**¿Cómo se ha detectado el problema?** En la fase 3 paso 6 de la metodología, se realiza un análisis de formatos y estandarización de la información, allí se identifican 3 variables de tipo tiempo, las cuales almacenan la misma información de forma redúndate.

**¿Por qué crees que se trata de un problema?** Almacenar información redundante no ayuda a optimizar el espacio de almacenamiento, puede generar confusión al momento de interpretar los datos.

**Ejemplo extraído del Dataset:**

**Table 2.** Muestra de datos columnas redundantes.

**Table

Description automatically generated with medium confidence**

**Forma de solucionarlo:** La forma de solucionarlo es eliminando las columnas redundantes, dichas columnas se pueden generar en procesos o pasos posteriores, mediante la generación de columnas derivadas o calculadas, las cuales se crean de acuerdo a la necesidad del consumidor, por ejemplo en el caso de las fechas podemos crear variables como el año, día o mes, con el fin de particional nuestra información en un sistema HDFS o posiblemente crear filtros de búsqueda para ese conjunto de datos mediante una visualización o Dashboard.

**Problema 3. Estandarización de variables tipo texto**

**¿Qué o cual es el problema?** En ciertas ocasiones encontramos que los valores asociados a un mismo valor se pueden duplicar, ya que los datos pueden ser recopilados de campos abiertos que los usuarios pueden rellenar de forma arbitraria o que se almacenaron de forma incorrecta.

**¿Cómo se ha detectado el problema?** En la fase 2 paso numero 8 mediante se identificaron valores atípicos para las variables categóricas por medio del análisis de los gráficos de histograma.

**¿Por qué crees que se trata de un problema?** Puede alterar las métricas construidas para una categoría ya que la información no se encuentra homogenizada y almacenada de manera consistente para una variable tipo categórica.

**Ejemplo extraído del Dataset:**

**Table 3.** Muestra de datos valores atípicos columna AddressType.

**Logo

Description automatically generated with medium confidence**

**Table 4.** Muestra de datos valores atípicos columna City.

****

**Forma de solucionarlo:** Estos valores atípicos en ocasiones son difíciles de encontrar, para ello podemos apoyarnos en los gráficos de histograma, funciones de agrupación y Distinct Count de los datos, para solucionarlo debemos buscar un estándar para los datos de texto, en nuestro caso hicimos una trasformación de los datos a mayúsculas, eliminamos los espacios con la función Trim y actualizamos la sintaxis de aquellos que tengan errores.

**Problema 4. Valores ausentes para las variables Range, City y State**

**¿Qué o cual es el problema?** Para la variable Range existe mas de un 50% de los datos ausentes o perdidos, la variable ciudad cuenta con 321 datos nulos y State 3.

**¿Cómo se ha detectado el problema?** En la fase 3 paso numero 7 se identifican datos ausentes para las variables mencionadas.

**¿Por qué crees que se trata de un problema?** La variable Range es una columna que no contiene información por lo cual se convierte en un dato que no nos aporta valor, la variable City tiene un 30% de los datos ausentes, lo cual puede generar ruido al momento de generar métricas o KPI de los datos, también puede generar inconsistencias ya que hay información que no tiene una ciudad asociada.

**Ejemplo extraído del Dataset:**

**Table 5.** Estadística valores Missing columnas Range, City y State.

Table

Description automatically generated

**Forma de solucionarlo:** La variable Range se debe eliminar del conjunto de datos, para la variable City y State inicialmente se rellenan aquellos valores nulos por el valor Dummy definido **“-99”.**

**Problema 5. Valores atípicos para el campo OriginalCrimeTypeName**

**¿Qué o cual es el problema?** El campo almacena información alfanumérica, es un campo tipo texto, en el cual se identifican valores solo numéricos, decimales, direcciones IP, y algunos caracteres sin sentido.

**¿Cómo se ha detectado el problema?** En la Fase 2 paso 8, se identifica una gran cantidad de valores atípicos mediante el uso de gráficos que catalogan la información y a su vez apoyándonos de funciones SQL para agrupar la información.

**¿Por qué crees que se trata de un problema?** Hay registro de información inconsistente, sin sentido o significado alguno, lo cual puede generar incongruencia a la hora de analizar nuestra información.

**Ejemplo extraído del Dataset:**

**Table 6.** Muestra de datos valores atípicos columna AddressType.

**Table

Description automatically generated**

**Forma de solucionarlo:** La manera de solucionarlo podría ser eliminando u omitiendo estos registros de nuestro Dataset, pero como desconocemos el significado de aquellos valores, puede que tengamos que primero corrobóralo con el ente que nos provee esta información, ya que algunos pueden ser códigos que tengan un significado asociado.

**Problema 6. Hallazgo de valores atípicos en la variable Disposition**

**¿Qué o cual es el problema?** Los datos almacenados en dicha variable cumplen un patrón estándar de letras en mayúsculas entre 2 y 3 caracteres, se evidencia un dato “22” numérico el cual es atípico para la población de datos de esta variable y un dato Dummy “NotRecorded”

**¿Cómo se ha detectado el problema?** En la fase 2 paso numero 8, por medio de gráficos, histogramas, funciones de descripción de distribución de los datos, entre otros

**¿Por qué crees que se trata de un problema?** El conjunto de datos en la fase 3 tiene una etapa de estandarización de variables y esquema, en el cual el tipo de dato de esta variable puede que logre aceptar este valor, pero no coincide con la naturaleza de la información, por ello puede generar errores e inconsistencias en futuros análisis.

**Ejemplo extraído del Dataset:**

**Table 7.** Valores atípicos columna Disposition.

****

**Forma de solucionarlo:** El valor se asume como un valor errado o nulo, por lo cual se rellena con el valor Dummy establecido por defecto para este proceso “-99”

**Problema 7. Hallazgo de errores para las filas 5773, 8023 y 8475 cuyas columnas afectadas son: City, State, AgencyId y AddressType**

**¿Qué o cual es el problema?**  Las filas mencionadas en el problema tienen almacenados los valores correspondientes de cada columna una posición a la derecha comenzando desde la columna City hasta la columna AddressType, dejando esta ultima sin un valor asociado ya que se perdió por ser la ultima columna y este valor se ha movido una posición a la derecha

**¿Cómo se ha detectado el problema?** En la fase 2 paso numero 8, por medio de gráficos, histogramas, funciones de descripción de distribución de los datos, entre otros.

**¿Por qué crees que se trata de un problema?** Se identifica claramente que los valores de cada columna están asociados a la columna derecha, por lo cual se genera inconsistencia de la información.

**Ejemplo extraído del Dataset:**

**Table 8.** Error de almacenamiento columna derecha.

**Table

Description automatically generated**

**Forma de solucionarlo:** Para el campo City se encuentra un valor atípico “S” y dos registros vacíos asociados a los identificadores con errores, estos serán reemplazados por el valor mas cercano “San Francisco”. La columna State tiene 3 valores nulos, los cuales serán reemplazados con el valor “CA”, que es el valor estándar o la media de esta variable y a su vez se identifica movimiento del este valor una columna a la derecha. La columna AgencyId tiene como factor común o media el valor “1”, aquí se registran 3 valores atípicos “CA”, dichos valores serán reemplazados por el valor “1”. Por ultimo la columna AddressType contiene 3 valores atípicos igual a “1” que ya fueron asignados respectivamente a la columna AgencyID, por lo cual estos valores se remplazan por valores nulos, y su equivalente valor Dummy que es “-99”.

Para entrar en detalle de cada una de las transformaciones anteriormente descritas refiérase al documento (<https://github.com/juadaves91/bd-bigdata-unir/blob/main/Paso%20a%20paso%20limpieza%20de%20un%20Dataset.pdf>) y al script o Jupyter notebook (<https://github.com/juadaves91/bd-bigdata-unir/blob/main/Limpieza%20Datos%20Dataset%20-%20CSV.pdf>).

# Formato JSON propuesto

A continuación, se detalla el formato de una colección o tabla y dos documentos o filas del Dataset, cabe mencionar que se identifica una relación de 1:1 entre la entidad Crimen y Agencia, por lo cual el valor de la agencia se encuentra en un nuevo documento. Para el JSON resultante se omitieron algunas columnas como Range la cual contenía mas del 50%, de los datos nulos, si por ejemplo se incluyeran datos en esta columna en un segundo Dataset con la misma estructura, no habría inconvenientes con adicionarla a nuestro formato JSON, y algunas columnas redundantes como CallTime y OffenseDate.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | [  {  "\_id": **ObjectId()**,  "CrimeId": **160913997**,  "OriginalCrimeTypeName": "Pay",  "CallDateTime": ISODate("2016-03-31T23:51:00.000-05:00"),  "Disposition": "ABA",  "Address": "1600 Block Of Green St",  "City": "ABA",  "State": "1",  "Agency": {  "\_id": **1**  },  "AddressType": "Premise Address"  },  {  "\_id": **ObjectId()**,  "CrimeId": **160920463**,  "OriginalCrimeTypeName": "Well Being Check",  "CallDateTime": ISODate("2016-04-01T05:49:00.000-05:00"),  "Disposition": "ND",  "Address": "Ada Ct/ofarrell St",  "City": "ND",  "State": "1",  "Agency": {  "\_id": **1**  },  "AddressType": "Intersection"  }  ] |

# Propuestas mejoras del conjunto de datos

Se listarán cada uno de los elementos propuestos para mejorar el conjunto de datos:

* **Codificación UTF-8:** Los valores o caracteres del conjunto de datos que se utilizo en el ejercicio son los utilizados para el idioma ingles, pero es importante tener un formato que nos sirva para abarcar un conjunto de caracteres especiales mas amplio como lo es la UTF-8, 16 o 32, el fichero actual tiene una codificación ASCCI (American Standard Code for International Interchange), ya que solo cuenta con 8 bits (128 caracteres) mientras que Unicode UTF (Unicode Transformation Format) cuenta con 135.ooo caracteres lo cual nos da la tranquilidad de incluir caracteres especiales del idioma español, tales como los caracteres tildes, ñ, entre otros.
* **Generación de hash MD5:** Es común generar un hash MD5, alfanumérico que representa la identidad e integridad del fichero, este código nos puede brindar la tranquilidad de compararlo con un MD5 generado por el analista de datos que lea el Dataset, con esto podemos validar que la información se leyó completamente y que no hay registros corruptos o que se este omitiendo información.
* **Validaciones previas:** Se puede ejecutar un proceso de pre-validación algo similar a lo que hicimos en el proceso de limpieza del Dataset, esto con la finalidad de ajustar los datos y minimizar errores en los mismos.
* **Estandarización de formatos:** Formatos como la fecha y algunas variables de tiempo pueden generar muchos dolores de cabeza para los analistas de datos, ya que no hay un estándar genérico, resulta ser una buena practica trabajar con un único formato de fecha cuando se esta trabajando un proceso ETL, al igual que para los textos, el manejar la información en un solo formato por ejemplo mayúsculas, minúsculas, sin espacios, homologar valores y errores de sintaxis podría ayudar a mejorar la calidad de los datos para los campos tipo texto.
* **Diccionario de datos:** Este archivo es muy usado para detallar los metadatos, características, llaves primarias, llaves foráneas, cantidad de registros, columnas, tipos de datos, si acepta o no nulos, la descripción, la fuente de datos de donde provienen, el dueño de la fuente, la fecha de creación o modificación, entre otros.
* **Archivo de propiedades (Nombre archivo | MD5 | Registros):** Este archivo contiene 3 datos el nombre del archivo, el MD5 que es el hash generado a partir de la lectura binaria del archivo de datos y por ultimo la cantidad de registros, este formato nos sirve para validar la integridad de los datos al momento de interpretarlos y llevar una trazabilidad de estos mediante un log.
* **Estándar de nombramiento**: Puede ser útil establecer un estándar de nombramiento, como por ejemplo <NOMBRE\_ARCHIVO\_YYYYMMDD\_CSV.csv>, ya que nos puede brindar información como por ejemplo el nombre, la fecha de creación, el tipo de archivo (CSV, JSON, AVRO, PARQUET o TXT) y en ocasiones un consecutivo el cual nos sirve para particionar la información en varios archivos de acuerdo con una cantidad de registros establecidos.
* **No incluir información redundante:** Evitar incluir información redúndate ya que esto ocupa mas espacio de almacenamiento y quizá no se utilice, dichas columnas calculadas, se pueden crear según la necesidad y no necesariamente sean aquellas que ya vienen en el Dataset.
* **No incluir columnas vacías:** Las columnas vacías no generan valor, por lo cual es recomendado no incluirlas.

# Conclusiones

La metodología explicada en este documento se apoya del Análisis Exploratorio de datos, la cual se compone de un conjunto de herramientas, pasos, técnicas estadísticas gráficos, entre otros para llevar a cabo el proceso fundamental para la interpretación básica de los datos y la limpieza de estos.

En este documento se detalla un proceso genérico basado en técnicas existentes y basadas en experiencia en el campo y se evidencian algunos problemas de calidad de datos que se presentan los cuales no son fáciles de tratar, en ocasiones debemos involucrar mas de un rol organizacional y en ocasiones hasta mas de un área organizacional que domina o es responsable una porción del negocio de donde proviene o se consume el dato, no solo esto puede interferir al momento de como manipulamos, limpiamos, estandarizamos o transformamos los datos, también es importante tener presente las reglas de negocio y el propósito o la usabilidad que se le dará a los datos, los escenarios son muy diversos (Dashboards, reportes, Procesos ETL, Bases de datos, informes, entre otros), por lo cual es importante mencionarlo.

# Referencias bibliográficas

datos.gob.es. (22 de 9 de 2021). *datos.gob.es.* Obtenido de Guía Práctica de Introducción al Análisis Exploratorio de Datos: https://datos.gob.es/es/documentacion/guia-practica-de-introduccion-al-analisis-exploratorio-de-datos.

Rocio Chavez Ciencia de Datos. (2020). Histogramas en Python [video]. YouTube.

AIEngineering. (2020). Data Cleaning and Analysis using Apache Spark [video]. YouTube.

Un Analista de Datos Peruano. (2019). Para qué me sirve la Desviación Estandar? [video]. YouTube.

orvizar TV. (2020). Pandas desde cero #3 |Limpiar Datos | Valores nulos | Eliminar datos | tutorial español [video]. YouTube.

Dattos.org. (2019). Gráficos en Python con Matplotlib, Seaborn y Ploty. YouTube.

Tomas\_IA. (2020). Limpieza de datos con Python [video]. YouTube.

Escuela de datos. (28 de 6 de 2017). *Escuela de datos*. Obtenido de Introducción a Pandas y Jupyter Notebook de Python: http://es.schoolofdata.org/2017/06/28/introduccion-a-pandas-y-jupyter-notebook-de-python/

Invarato, R. (20 de 4 de 2021). *jarroba.com*. Obtenido de Instalar Apache Spark en cualquier sistema operativo y aprender a programarlo con Python, Scala o Java: https://jarroba.com/instalar-apache-spark-en-cualquier-sistema-operativo-y-aprender-a-programarlo-con-python-scala-o-java/