**PRÁCTICA DE DATOS ABIERTOS Y VISUALIZACIÓN DINÁMICA**

Por: Manuela Larrea Gómez

1. **Directorio del proyecto:**

**A blue and yellow snake logo

Description automatically generatedHome.py:** Script con la landing page.

**requirements.txt:** Archivo con las librerías necesarias

**folder html" Icon - Download for free – Iconduckpages:** Carpeta con la paginación de la aplicación.

**A blue and yellow snake logo

Description automatically generatedProveniencia.py**: Script de visualizaciones dinámicas.

**A blue and yellow snake logo

Description automatically generatedArtistas.py**: Script de visualizaciones dinámicas.

**A blue and yellow snake logo

Description automatically generatedA blue and yellow snake logo

Description automatically generatedCulturas.py**: Script de visualizaciones dinámicas.  
**A blue and yellow snake logo

Description automatically generated** **Evolucion Temporal.py**: Script de visualizaciones dinámicas.

**Hardcoded Folder Icons in macOS | Antranig VartanianProyecto Jupyter - Wikipedia, la enciclopedia libreclean\_data.ipynb:** Notebook con el proceso de limpieza de los datos.

**utils:**

**A blue and yellow snake logo

Description automatically generatedendpoints.py**: Script con los endpoints de las APIs usadas.

**A blue and yellow snake logo

Description automatically generatednationality\_mapper.py**: Clase creada para apoyar el proceso de limpieza de datos.

**A blue and yellow snake logo

Description automatically generatedfilters.py**: script con los filtros usados para las visualizaciones dinámicas.

folder html" Icon - Download for free – Iconduck

**data:** Carpeta con los datos.

**met\_object.feather**: Salidas de las consultas a la API del MET**.  
countries.feather:** Salidas de las consultas a la API de Google Maps. **clean\_data.feather:**Data set limpio

1. **Requisitos del sistema:**

|  |  |
| --- | --- |
| **Paquete o librería** | **Versión** |
| Requests | 2.31.0 o superior |
| Json | 0.9.14 o superior |
| Pathlib | 1.0.1 o superior |
| Numpy | 1.24.3 o superior |
| Pandas | 2.1.4 o superior |
| Missingno | 0.4.2 o superior |
| matplotlib | 3.8.0 o superior |
| Fuzzywuzzy | 0.18.0 o superior |
| Streamlit | 1.31.0 o superior |
| Altair | 5.2.0 o superior |

1. **Instrucciones de uso:**

La aplicación se encuentra desplegada en la siguiente dirección:

Asimismo, los archivos con las salidas de ejecución guardadas son:

clean\_data.html

En caso de que se quiera reproducir el proyecto localmente y de forma manual, se deberá ejecutar con el siguiente orden:

1. Ejecutar el siguiente comando: ‘pip install -r requirements.txt’, fijando el puntero del directorio en el directorio raíz.

**NOTA**: El fichero requirements tiene todos los paquetes que deben ser previamente instalados.

1. Ejecutar el archivo clean\_data.ipynb

**NOTA:** Los datos recolectados con las consultas a las API están alojados en la carpeta data (met\_object.feather y countries.feather). Sin embargo, el script de limpieza esta diseñado para que verifique la existencia de estos datos y en caso de no encontrarlos, procede a ejecutar los endpoints.

1. Ejecutar el script \_\_\_ para desplegar la aplicación en local.

**Observaciones finales:**

Para el proyecto, se optó por guardar los datos en archivos con formato .feather. Esta elección no fue casual, sino que se basó en varias ventajas significativas que este formato ofrece. En primer lugar, los archivos .feather son a menudo más compactos que sus equivalentes en otros formatos populares, como .csv o .xlsx. Esta característica de tamaño reducido resultó ser especialmente útil al subir la carpeta de datos al repositorio de GitHub, ya que permitió una carga eficiente y sin problemas.

Además, el formato .feather, desarrollado por Wes McKinney, es conocido por su velocidad excepcional de lectura y escritura, lo que facilita un flujo de trabajo eficiente, especialmente cuando se manejan grandes conjuntos de datos. Por último, su compatibilidad con múltiples lenguajes de programación, incluyendo Python y R, añade una capa extra de versatilidad.

*Fuente:* [*https://towardsdatascience.com/the-best-format-to-save-pandas-data-414dca023e0d*](https://towardsdatascience.com/the-best-format-to-save-pandas-data-414dca023e0d)

**VISITANDO EL MET DE NUEVA YORK SIN SALIR DE MADRID**

Este proyecto se basa en la política de datos abiertos del Museo Metropolitano de Arte de Nueva York (The Met) y utiliza su API para acceder a su vasta colección de obras de arte. En febrero de 2017, The Met lanzó su Iniciativa de Acceso Abierto, que permite el uso irrestricto de todas las imágenes de obras de arte de dominio público y datos básicos de todas las obras en su colección bajo la licencia Creative Commons Zero (CC0). Esto significa que cualquier persona puede descargar, compartir y remezclar imágenes y datos sobre las obras de arte en la colección de The Met.

El proyecto también se apoya en la API de Google Maps, que proporciona una amplia gama de servicios y herramientas para crear y optimizar aplicaciones basadas en la ubicación. Esta combinación permite visualizar la información de las obras de arte en un contexto geográfico, proporcionando una nueva dimensión a la exploración de la colección de The Met.

Finalmente, para desplegar la aplicación, se utiliza Streamlit.

**Objetivos del proyecto**

El proyecto se centra en responder las siguientes preguntas:

1. ¿De dónde provienen las obras de arte del Met?
2. ¿Cuáles son los artistas que más le interesan al Met?
3. ¿Cuáles son las culturas que más le interesan al Met?
4. ¿Cuál es el departamento del museo que tiene más piezas?
5. ¿Cuál es la evolución temporal del tamaño de la colección del Met?
6. ¿Cuáles son las obras más importantes del Met?
7. ¿Qué porción de las obras de arte del Met son de dominio público?

**Extracción de los datos:**

Se realizó una serie de solicitudes HTTP GET a la API del Museo Metropolitano de Nueva York para extraer metadatos de su colección de arte, que consta de 481094 piezas. Estas solicitudes se implementaron mediante funciones de llamada a la API que recopilaron información detallada de cada obra de arte. Los metadatos de cada obra incluyen la fecha de creación, el país de origen, el nombre del artista, los materiales utilizados, el departamento del museo al que pertenece, la fecha de adquisición por parte del museo, y si es de dominio público, entre otros. Todos los datos recopilados de la API se serializaron y almacenaron en un archivo JSON.

Además, se implementó una función GET para interactuar con la API de Google Maps. Esta función se utilizó para obtener las coordenadas de latitud y longitud de los países mencionados en los metadatos de las obras de arte. Los resultados de estas solicitudes también se serializaron y almacenaron en un archivo JSON.

**Tratamiento y preparación de los datos:**

En términos generales, los dos desafíos principales en la fase de preprocesamiento de los datos recolectados fueron: La gestión de los valores faltantes (missing values y la normalización de los datos relativos a la nacionalidad de los artistas y los países de origen de las obras de arte.

En cuanto a la gestión de valores faltantes, se priorizó la reducción de la dimensionalidad del conjunto de datos (dataset), limitando estrictamente el proyecto a los objetivos descritos previamente. Además, se realizó un análisis de correlación de variables utilizando mapas de calor y matrices de nulidad para comprender la posible naturaleza e implicaciones de los datos faltantes.

Las decisiones más importantes (de eliminación e imputación) fueron:

* La mayoría de las columnas del conjunto de datos están predominantemente vacías. Para enriquecerlo, sería útil fusionarlo con otra fuente de datos. Sin embargo, según la investigación realizada, otros museos del mundo que disponen de API solo incluyen su colección en la base de datos y, por razones obvias, una misma obra no puede estar en dos museos al mismo tiempo. Por esta razón, se decidió centrar el proyecto en aquellos atributos que no superan el 60% de nulidad.
* No es extraño que el dataset tenga valores faltantes en *title*, porque existen en el sector obras artísticas sin título. Esta información se validó con la Casa de subastas Real de España. Tampoco es inesperado que las columnas de *'artistDisplayName'* y *'country’* tengan valores nulos, porque en el sector artístico es común que haya obras con artistas anónimos.
* De acuerdo con el datacard de la API, los atributos ‘*country’* y ‘*artistNationality’* responden de diferente forma a una misma pregunta: País de origen de la obra o el artista. El análisis exploratorio de los datos mostró que, en muchos casos, los valores faltantes de ‘*country’* estaban en ‘*artistNationality’*, y viceversa.
* Debido a que una obra puede pertenecer a varios artistas, y cada artista puede tener una nacionalidad distinta, se decide que la obra será de la nacionalidad predominante de los artistas involucrados. Si no existe una nacionalidad predominante, se imputa como 'Otro'
* Con el objetivo de unificar la información y, por ende, enriquecerla, se construyó una clase ‘*NationalityMapper’* compuesta por dos diccionarios, que permitió mapear las nacionalidades a los países y, a su vez, normalizar los nombres de los países. Los métodos incluidos en esta clase se apoyaron en funciones del script de limpieza que utilizaron expresiones regulares (regex) para normalizar los nombres de los países, ya que tenían muchos caracteres que dificultaban la lectura de los datos que se iban a fusionar con los datos extraídos de Google Maps.

*El detalle de la limpieza puede consultarse en* clean\_data.html

**Justificación de las visualizaciones:**

A continuación, se justifica el uso de las visualizaciones dinámicas desarrolladas para cada pregunta objetivo del proyecto y su ubicación en la aplicación desplegada.

1. **¿De dónde provienen las obras de arte del Met?**

Se utilizó un mapa interactivo que combina la información de la API de Google Maps y los datos del país de origen de las obras. Esta visualización geoespacial permite una comprensión intuitiva y global de la distribución geográfica de las obras de arte.

*Ubicación en la aplicación:*  🌎Proveniencia

1. **¿Cuáles son los artistas que más le interesan al Met?**

Se implementó un gráfico de barras dinámico que muestra el nombre del artista y el número total de obras. La interactividad permite al usuario personalizar la visualización según sus intereses, incluyendo o excluyendo artistas desconocidos, obras destacadas o no destacadas, obras de dominio público o protegidas por derechos de autor y una slider para acotar las obras por año de adquisición por parte del museo.

*Ubicación en la aplicación:* 👩‍🎨Artistas

1. **¿Cuáles son las culturas que más le interesan al Met?**

Se implementó un gráfico de barras dinámico que muestra el nombre del artista y el número total de obras. La interactividad permite al usuario personalizar la visualización según sus intereses, incluyendo o excluyendo obras que tienen culturas desconocidas, obras destacadas o no destacadas, obras de dominio público o protegidas por derechos de autor y una slider para acotar las obras por año de adquisición por parte del museo.

*Ubicación en la aplicación:* 🗿Culturas

1. **¿Cuál es el departamento del museo que tiene más piezas?**

Se implementó una casilla de verificación que permite dinamizar los demás gráficos. Esta funcionalidad permite al usuario explorar cómo la distribución de las piezas varía entre los diferentes departamentos del museo.

*Ubicación en la aplicación: En todas las paginaciones (como filtro).*

1. **¿Cuál es la evolución temporal del tamaño de la colección del Met?**

Se utilizó una línea de tiempo para mostrar el número de obras por año. Esta visualización permite al usuario entender cómo ha crecido la colección del Met a lo largo del tiempo y puede variar la escala temporal por medio de una slider.

*Ubicación en la aplicación:* 📅 Evolución temporal

1. **¿Cuáles son las obras más importantes del Met?**

Se implementó un filtro con checkbox que dinamiza los demás gráficos. Esta funcionalidad permite al usuario explorar cómo cambian las visualizaciones al seleccionar ‘Es Destacado’ o no.

*Ubicación en la aplicación: En todas las paginaciones (como filtro).*

**¿Qué porción de las obras de arte del Met son de dominio público?**

Se implementó un filtro con botón de radio que dinamiza los demás gráficos. Esta funcionalidad permite al usuario explorar cómo cambian las visualizaciones al centrarse en las obras de dominio público o las obras protegidas por derechos de autor.

*Ubicación en la aplicación: En todas las paginaciones (como filtro).*

**Explicación de los insights/conclusiones:**