

# **Maestría en Análisis y Visualización de Datos Masivos**

## Análisis grupal tendencias D3

### HERRAMIENTAS DE VISUALIZACIÓN

Integrantes del grupo:

Hendrick Rolando Calderón Aguirre – Matrícula: 5185535 – 447942

Pedro David Gabriel Wong – Matrícula: 5160143

Oscar Canseco Salas – Matrícula: 5182949 - 445615

## Índice general

Índice de ilustraciones	I
Introducción	II
Objetivos	III
General	III
Específicos	III
Hoja de control	IV
Contenido	1
Estructura general	4
Código fuente	5
1.    limpiezaCodigosPais.ipynb	5
2.    index.html	12
3.    main.css	14
4.    main.js	15
Conclusiones	23
Recomendaciones	24
Bibliografía	25

## **Índice de ilustraciones**

### **Gráficos**

Gráfico 1. Mapa coroplético mundial de contagios de COVID-19	2
Gráfico 2. Contagios y fallecimientos por COVID-19	3

### **Tablas**

Tabla 1. Descripción de la estructura general de la actividad	4
---	---

### **Imágenes**

Imagen 1. Estructura general de la actividad	4
--	---

## Introducción

La visualización es un fenómeno humano tratado por estudiosos provenientes de múltiples ámbitos científicos, lo que ha contribuido a la conformación desde distintos enfoques de un marco teórico-práctico amplio. En general, se destacan tres perspectivas de análisis asociadas: cognitiva, tecnológica, y comunicativa.

Por un lado la perspectiva cognitiva crea representaciones visuales para apoyar distintas actividades es un acto que se emprende con frecuencia, con la finalidad de comprender las relaciones que se dan en un contexto particular, y ayuda a la comprensión del mundo externo a partir del pensamiento y el razonamiento.

La perspectiva tecnológica conserva diversos estudios sobre las interacciones que ocurren entre los humanos y las computadoras describen cómo las visualizaciones externas pueden amplificar, reforzar o aumentar la cognición

Finalmente la perspectiva comunicativa considera una tarea del proceso comunicativo, por medio del cual se transforman los datos abstractos y los fenómenos complejos de la realidad en mensajes visibles, y que lleva a un proceso de descubrimiento del conocimiento. (Torres, 2009)

En el presente trabajo, se consultaron los principios de Gestalt para poder brindar una serie de visualizaciones con respecto a una serie de data sets recopilados con respecto a la presente situación ocasionada por el virus SARS-COV-2.

El conjunto de datos utilizado fue recuperado de <https://covid19.who.int/>, y cuentan con registros del periodo comprendido entre el 03/01/2020 al 29/11/2021. Se decidió utilizar este data-set debido a que la actual pandemia, es un fenómeno de alto interés público y que puede brindar más información e interpretación para el usuario que desee consultarlo y pueda informarse acerca de las tendencias actuales que esta ha tenido.

## **Objetivos**

### **General**

Crear una visualización interactiva en D3.js que permita analizar tendencias presentes a nivel mundial debido a la enfermedad COVID-19 tales como contagios y defunciones.

### **Específicos**

1. Investigar un dataset que permita desarrollar una visualización interactiva con la librería D3 de JavaScript.
2. Desarrollar un script de programación en Python que permita limpiar y agrupar los datos para su correcta interpretación.
3. Programar un mapa coroplético que represente por medio de una escala de color rojo la cantidad de contagios de COVID-19 a nivel mundial.
4. Programar un gráfico de barras interactivo que represente la cantidad de contagios y fallecidos por país a causa del COVID-19.

## Hoja de control

Integrante	Actividad desarrollada	Fecha
Pedro Gabriel y Oscar Canseco	<ul style="list-style-type: none"> <li>Investigación y propuesta del dataset WHO-COVID-19-global-data.csv. Recuperado de: <a href="https://covid19.who.int/table">https://covid19.who.int/table</a> el 29 de noviembre de 2021</li> </ul>	29/11/2021
Hendrick Calderón	<p>Generación de la estructura inicial del proyecto:</p> <ul style="list-style-type: none"> <li>resources: main.js, main.css</li> <li>README.txt</li> <li>WHO-COVID-19-global-data-alpha3.csv</li> <li>WHO-COVID-19-global-data.csv</li> <li>WHO-COVID-19-summary.csv</li> <li>index.html</li> <li>limpiezaCodigosPais.ipynb</li> <li>Análisis grupal tendencias D3.docx</li> </ul> <p>Programación inicial de la visualización:</p> <ul style="list-style-type: none"> <li>Map</li> <li>BarPlot</li> </ul>	05/12/2021
Pedro Gabriel	<ul style="list-style-type: none"> <li>Programación de final del archivo CSS.</li> <li>Actualización de la visualización.</li> </ul>	19/12/2021
Oscar Canseco	<ul style="list-style-type: none"> <li>Elaboración escrita del informe y testeo de visualizaciones.</li> <li>Programación final de la visualización.</li> </ul>	27/12/2021
Hendrick Calderón	<ul style="list-style-type: none"> <li>Elaboración audiovisual y demostración de las visualizaciones.</li> </ul>	27/12/2021

Referencia del trabajo colaborativo: <https://github.com/heroca60/tendencias-d3.git>

## Contenido

Se realizó un cambio al data-set original (WHO-COVID-19-global-data.csv), haciendo una copia con configuración Alpha-3 debido a que el data-set original cuenta con configuración Alpha-2. Esto es debido a que los polígonos utilizados para dibujar las visualizaciones en el mapa global, cuentan con un enlace externo a un archivo que presenta los países con los países en Alpha-3. Para ello, se utilizó Python.

En este mismo código, al finalizar la transformación ya mencionada, se hizo una agrupación en cuanto a los nuevos casos y casos de defunción por cada país, generando un nuevo archivo llamado *WHO-COVID-19-summary.csv*.

Se realizó un video donde se hace una demostración y explicación del presente trabajo, dicho video puede ser es ubicado por medio del siguiente enlace:

<https://drive.google.com/file/d/1k5ir4DD71msXLznfq6qTcMaDX71IRK4I/view?usp=sharing>

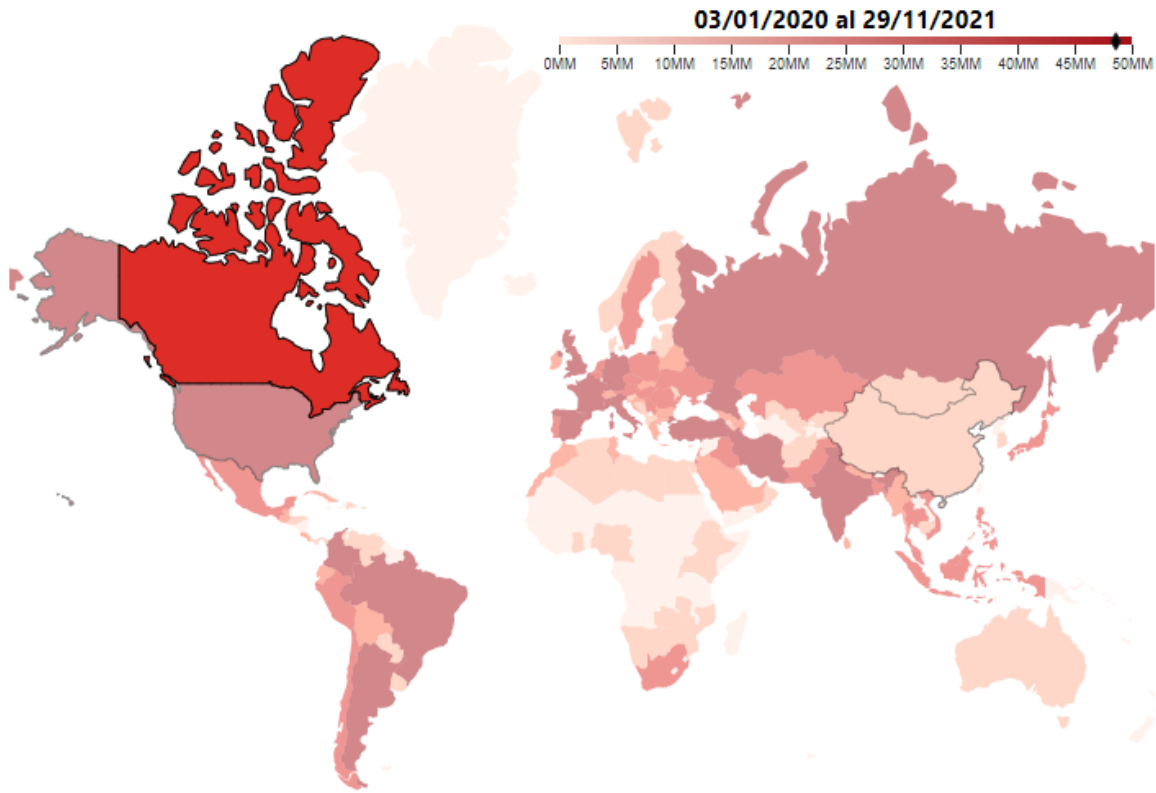
Así mismo, se puede encontrar el repositorio de GitHub con todos los archivos mencionados en el siguiente enlace:

<https://github.com/heroca60/tendencias-d3>

Para el desarrollo del siguiente trabajo se utilizó la versión 6 de D3.

La visualización cuenta con dos gráficos, en primer lugar se puede observar:

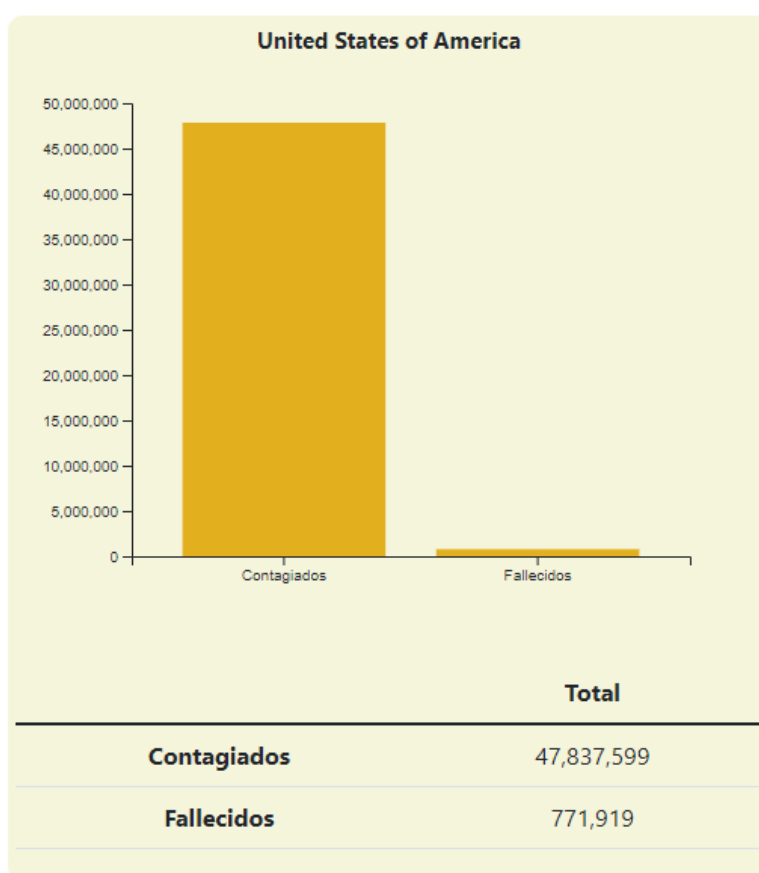
Gráfico 1. Mapa coroplético mundial de contagios de COVID-19



En el *Gráfico 1. Mapa coroplético mundial de contagios de COVID-19*, el usuario puede observar a nivel mundial, el comportamiento de contagiados y fallecidos por país. La escala de colores utilizada es directamente proporcional al número de personas contagiadas, es decir, a mayor número de contagios, el color del país será de un tono más oscuro. La visualización cuenta con la característica de que posicionarse sobre un país, se tendrá un enfoque visual sobre la selección y haciendo un desenfoque del resto.



Gráfico 2. Contagios y fallecimientos por COVID-19



Al proporcionar un click sobre un país del gráfico 1, se desplegará del lado derecho, el detalle de dicha selección mostrado en un gráfico de barras el número de contagios y fallecimientos (Gráfica 2). El eje y, número de personas, tiene un rango dinámico que se irá adecuando según las unidades que cada país muestre.

Se utilizó dos gráficos no figurativos (mapa coroplético y gráfico de barras) ya que los gráficos no figurativos tienen la ventaja de presentarse de forma fácil y sencilla de reflejan patrones y tendencias procedentes de un data-set, así como las relaciones que se pueden establecer entre ellos. Es de suma importancia resaltar que el modo de visualizar los datos es abstracto y no guarda algún parecido real con lo que se ha presentado. (UNIR, 2021)

## Estructura general

Imagen 1. Estructura general de la actividad

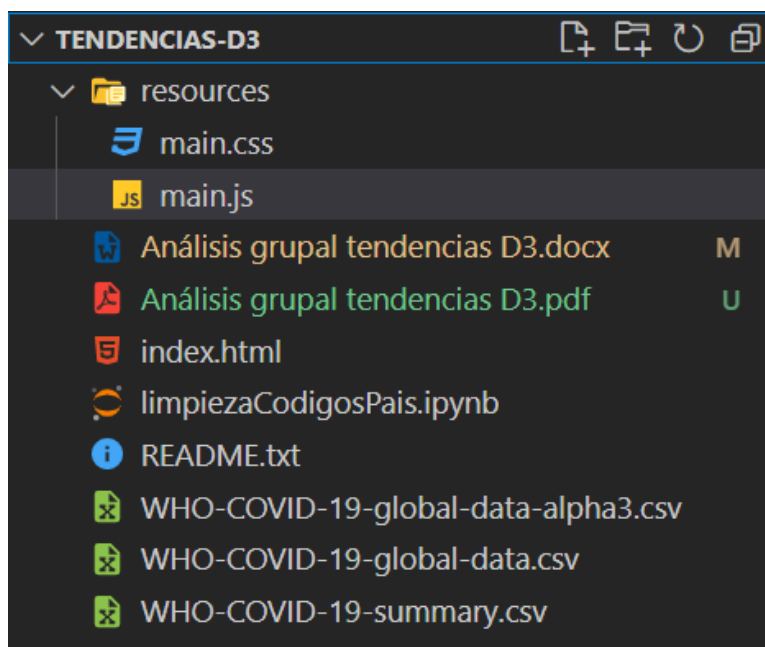


Tabla 1. Descripción de la estructura general de la actividad

Elemento	Descripción
resources	Carpeta que contiene los archivos main.css y main.js
main.css	Archivo de hoja de estilos en cascada
main.js	Archivo de javascript de contiene la programación utilizando la librería D3.js
Análisis grupal tendencias D3.docx y Análisis grupal tendencias D3.pdf	Archivo general de explicación de dicha actividad.
index.html	Archivo web que contiene la visualización.

Continuación Tabla 1.

limpiezaCodigosPais.ipynb	Archivo de Jupyter notebook que contiene programación en Python para la limpieza y agrupación de datos.
README.txt	Archivo de explicación básica de la estructura de la actividad.
WHO-COVID-19-global-data.csv	Dataset original recuperado de: <a href="https://covid19.who.int/table">https://covid19.who.int/table</a> el 29 de noviembre de 2021
WHO-COVID-19-global-data-alpha3.csv	Archivo csv generado con codificación Alpha 3 por medio del script del archivo limpiezaCodigosPais.ipynb
WHO-COVID-19-summary.csv	Archivo csv generado con la agrupación de contagios y fallecidos por COVID-19.

Fuente: elaboración propia

## Código fuente

### 1. limpiezaCodigosPais.ipynb

```
path = 'WHO-COVID-19-global-data.csv'
import pandas as pd
import numpy as np
# leyendo el archivo csv original
df = pd.read_csv('WHO-COVID-19-global-data.csv')
# imprimiendo únicamente los encabezados
df.head()
# Visualizando la composición del dataframe = 165189 filas y 8 columnas
df.shape
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['AD'], 'AND')
df['Country_code'] = df['Country_code'].replace(['AE'], 'ARE')
df['Country_code'] = df['Country_code'].replace(['AF'], 'AFG')
df['Country_code'] = df['Country_code'].replace(['AG'], 'ATG')
df['Country_code'] = df['Country_code'].replace(['AI'], 'AIA')
```

```

df['Country_code'] = df['Country_code'].replace(['AL'], 'ALB')
df['Country_code'] = df['Country_code'].replace(['AM'], 'ARM')
df['Country_code'] = df['Country_code'].replace(['AN'], 'ANT')
df['Country_code'] = df['Country_code'].replace(['AO'], 'AGO')
df['Country_code'] = df['Country_code'].replace(['AQ'], 'ATA')
df['Country_code'] = df['Country_code'].replace(['AR'], 'ARG')
df['Country_code'] = df['Country_code'].replace(['AS'], 'ASM')
df['Country_code'] = df['Country_code'].replace(['AT'], 'AUT')
df['Country_code'] = df['Country_code'].replace(['AU'], 'AUS')
df['Country_code'] = df['Country_code'].replace(['AW'], 'ABW')
df['Country_code'] = df['Country_code'].replace(['AX'], 'ALA')
df['Country_code'] = df['Country_code'].replace(['AZ'], 'AZE')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['BA'], 'BIH')
df['Country_code'] = df['Country_code'].replace(['BB'], 'BRB')
df['Country_code'] = df['Country_code'].replace(['BD'], 'BGD')
df['Country_code'] = df['Country_code'].replace(['BE'], 'BEL')
df['Country_code'] = df['Country_code'].replace(['BF'], 'BFA')
df['Country_code'] = df['Country_code'].replace(['BG'], 'BGR')
df['Country_code'] = df['Country_code'].replace(['BH'], 'BHR')
df['Country_code'] = df['Country_code'].replace(['BI'], 'BDI')
df['Country_code'] = df['Country_code'].replace(['BJ'], 'BEN')
df['Country_code'] = df['Country_code'].replace(['BL'], 'BLM')
df['Country_code'] = df['Country_code'].replace(['BM'], 'BMU')
df['Country_code'] = df['Country_code'].replace(['BN'], 'BRN')
df['Country_code'] = df['Country_code'].replace(['BO'], 'BOL')
df['Country_code'] = df['Country_code'].replace(['BR'], 'BRA')
df['Country_code'] = df['Country_code'].replace(['BS'], 'BHS')
df['Country_code'] = df['Country_code'].replace(['BT'], 'BTN')
df['Country_code'] = df['Country_code'].replace(['BV'], 'BVT')
df['Country_code'] = df['Country_code'].replace(['BW'], 'BWA')
df['Country_code'] = df['Country_code'].replace(['BY'], 'BLR')
df['Country_code'] = df['Country_code'].replace(['BZ'], 'BLZ')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['CA'], 'CAN')
df['Country_code'] = df['Country_code'].replace(['CC'], 'CCK')
df['Country_code'] = df['Country_code'].replace(['CD'], 'COD')
df['Country_code'] = df['Country_code'].replace(['CF'], 'CAF')
df['Country_code'] = df['Country_code'].replace(['CG'], 'COG')
df['Country_code'] = df['Country_code'].replace(['CH'], 'CHE')
df['Country_code'] = df['Country_code'].replace(['CI'], 'CIV')
df['Country_code'] = df['Country_code'].replace(['CK'], 'COK')
df['Country_code'] = df['Country_code'].replace(['CL'], 'CHL')

```

```

df['Country_code'] = df['Country_code'].replace(['CM'], 'CMR')
df['Country_code'] = df['Country_code'].replace(['CN'], 'CHN')
df['Country_code'] = df['Country_code'].replace(['CO'], 'COL')
df['Country_code'] = df['Country_code'].replace(['CR'], 'CRI')
df['Country_code'] = df['Country_code'].replace(['CU'], 'CUB')
df['Country_code'] = df['Country_code'].replace(['CV'], 'CPV')
df['Country_code'] = df['Country_code'].replace(['CW'], 'CUW')
df['Country_code'] = df['Country_code'].replace(['CX'], 'CXR')
df['Country_code'] = df['Country_code'].replace(['CY'], 'CYP')
df['Country_code'] = df['Country_code'].replace(['CZ'], 'CZE')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['DE'], 'DEU')
df['Country_code'] = df['Country_code'].replace(['DJ'], 'DJI')
df['Country_code'] = df['Country_code'].replace(['DK'], 'DNK')
df['Country_code'] = df['Country_code'].replace(['DM'], 'DMA')
df['Country_code'] = df['Country_code'].replace(['DO'], 'DOM')
df['Country_code'] = df['Country_code'].replace(['DZ'], 'DZA')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['EC'], 'ECU')
df['Country_code'] = df['Country_code'].replace(['EE'], 'EST')
df['Country_code'] = df['Country_code'].replace(['EG'], 'EGY')
df['Country_code'] = df['Country_code'].replace(['EH'], 'ESH')
df['Country_code'] = df['Country_code'].replace(['ER'], 'ERI')
df['Country_code'] = df['Country_code'].replace(['ES'], 'ESP')
df['Country_code'] = df['Country_code'].replace(['ET'], 'ETH')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['FI'], 'FIN')
df['Country_code'] = df['Country_code'].replace(['FJ'], 'FJI')
df['Country_code'] = df['Country_code'].replace(['FK'], 'KLK')
df['Country_code'] = df['Country_code'].replace(['FM'], 'FSM')
df['Country_code'] = df['Country_code'].replace(['FO'], 'FRO')
df['Country_code'] = df['Country_code'].replace(['FR'], 'FRA')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['GA'], 'GAB')
df['Country_code'] = df['Country_code'].replace(['GB'], 'GBR')
df['Country_code'] = df['Country_code'].replace(['GD'], 'GRD')
df['Country_code'] = df['Country_code'].replace(['GE'], 'GEO')
df['Country_code'] = df['Country_code'].replace(['GF'], 'GUF')
df['Country_code'] = df['Country_code'].replace(['GG'], 'GGY')
df['Country_code'] = df['Country_code'].replace(['GH'], 'GHA')
df['Country_code'] = df['Country_code'].replace(['GI'], 'GIB')

```

```

df['Country_code'] = df['Country_code'].replace(['GL'], 'GRL')
df['Country_code'] = df['Country_code'].replace(['GM'], 'GMB')
df['Country_code'] = df['Country_code'].replace(['GN'], 'GIN')
df['Country_code'] = df['Country_code'].replace(['GP'], 'GLP')
df['Country_code'] = df['Country_code'].replace(['GQ'], 'GNQ')
df['Country_code'] = df['Country_code'].replace(['GR'], 'GRC')
df['Country_code'] = df['Country_code'].replace(['GS'], 'SGS')
df['Country_code'] = df['Country_code'].replace(['GT'], 'GTM')
df['Country_code'] = df['Country_code'].replace(['GU'], 'GUM')
df['Country_code'] = df['Country_code'].replace(['GW'], 'GNB')
df['Country_code'] = df['Country_code'].replace(['GY'], 'GUY')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['HK'], 'HKG')
df['Country_code'] = df['Country_code'].replace(['HM'], 'HMD')
df['Country_code'] = df['Country_code'].replace(['HN'], 'HND')
df['Country_code'] = df['Country_code'].replace(['HR'], 'HRV')
df['Country_code'] = df['Country_code'].replace(['HT'], 'HTI')
df['Country_code'] = df['Country_code'].replace(['HU'], 'HUN')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['ID'], 'IDN')
df['Country_code'] = df['Country_code'].replace(['IE'], 'IRL')
df['Country_code'] = df['Country_code'].replace(['IL'], 'ISR')
df['Country_code'] = df['Country_code'].replace(['IM'], 'IMN')
df['Country_code'] = df['Country_code'].replace(['IN'], 'IND')
df['Country_code'] = df['Country_code'].replace(['IO'], 'IOT')
df['Country_code'] = df['Country_code'].replace(['IQ'], 'IRQ')
df['Country_code'] = df['Country_code'].replace(['IR'], 'IRN')
df['Country_code'] = df['Country_code'].replace(['IS'], 'ISL')
df['Country_code'] = df['Country_code'].replace(['IT'], 'ITA')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['JE'], 'JEY')
df['Country_code'] = df['Country_code'].replace(['JM'], 'JAM')
df['Country_code'] = df['Country_code'].replace(['JO'], 'JOR')
df['Country_code'] = df['Country_code'].replace(['JP'], 'JPN')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['KE'], 'KEN')
df['Country_code'] = df['Country_code'].replace(['KG'], 'KGZ')
df['Country_code'] = df['Country_code'].replace(['KH'], 'KHM')
df['Country_code'] = df['Country_code'].replace(['KI'], 'KIR')
df['Country_code'] = df['Country_code'].replace(['KM'], 'COM')
df['Country_code'] = df['Country_code'].replace(['KN'], 'KNA')

```

```

df['Country_code'] = df['Country_code'].replace(['KP'], 'PRK')
df['Country_code'] = df['Country_code'].replace(['KR'], 'KOR')
df['Country_code'] = df['Country_code'].replace(['KW'], 'KWT')
df['Country_code'] = df['Country_code'].replace(['KY'], 'CYM')
df['Country_code'] = df['Country_code'].replace(['KZ'], 'KAZ')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['LA'], 'LAO')
df['Country_code'] = df['Country_code'].replace(['LB'], 'LBN')
df['Country_code'] = df['Country_code'].replace(['LC'], 'LCA')
df['Country_code'] = df['Country_code'].replace(['LI'], 'LIE')
df['Country_code'] = df['Country_code'].replace(['LK'], 'LKA')
df['Country_code'] = df['Country_code'].replace(['LR'], 'LBR')
df['Country_code'] = df['Country_code'].replace(['LS'], 'LSO')
df['Country_code'] = df['Country_code'].replace(['LT'], 'LTU')
df['Country_code'] = df['Country_code'].replace(['LU'], 'LUX')
df['Country_code'] = df['Country_code'].replace(['LV'], 'LVA')
df['Country_code'] = df['Country_code'].replace(['LY'], 'LBY')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['MA'], 'MAR')
df['Country_code'] = df['Country_code'].replace(['MC'], 'MCO')
df['Country_code'] = df['Country_code'].replace(['MD'], 'MDA')
df['Country_code'] = df['Country_code'].replace(['ME'], 'MNE')
df['Country_code'] = df['Country_code'].replace(['MF'], 'MAF')
df['Country_code'] = df['Country_code'].replace(['MG'], 'MDG')
df['Country_code'] = df['Country_code'].replace(['MH'], 'MHL')
df['Country_code'] = df['Country_code'].replace(['MK'], 'MKD')
df['Country_code'] = df['Country_code'].replace(['ML'], 'MLI')
df['Country_code'] = df['Country_code'].replace(['MM'], 'MMR')
df['Country_code'] = df['Country_code'].replace(['MN'], 'MNG')
df['Country_code'] = df['Country_code'].replace(['MO'], 'MAC')
df['Country_code'] = df['Country_code'].replace(['MP'], 'MNP')
df['Country_code'] = df['Country_code'].replace(['MQ'], 'MTQ')
df['Country_code'] = df['Country_code'].replace(['MR'], 'MRT')
df['Country_code'] = df['Country_code'].replace(['MS'], 'MSR')
df['Country_code'] = df['Country_code'].replace(['MT'], 'MLT')
df['Country_code'] = df['Country_code'].replace(['MU'], 'MUS')
df['Country_code'] = df['Country_code'].replace(['MV'], 'MDV')
df['Country_code'] = df['Country_code'].replace(['MW'], 'MWI')
df['Country_code'] = df['Country_code'].replace(['MX'], 'MEX')
df['Country_code'] = df['Country_code'].replace(['MY'], 'MYS')
df['Country_code'] = df['Country_code'].replace(['MZ'], 'MOZ')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3

```



```

df['Country_code'] = df['Country_code'].replace(['NA'], 'NAM')
df['Country_code'] = df['Country_code'].replace(['NC'], 'NCL')
df['Country_code'] = df['Country_code'].replace(['NE'], 'NER')
df['Country_code'] = df['Country_code'].replace(['NF'], 'NFK')
df['Country_code'] = df['Country_code'].replace(['NG'], 'NGA')
df['Country_code'] = df['Country_code'].replace(['NI'], 'NIC')
df['Country_code'] = df['Country_code'].replace(['NL'], 'NLD')
df['Country_code'] = df['Country_code'].replace(['NO'], 'NOR')
df['Country_code'] = df['Country_code'].replace(['NP'], 'NPL')
df['Country_code'] = df['Country_code'].replace(['NR'], 'NRU')
df['Country_code'] = df['Country_code'].replace(['NU'], 'NIU')
df['Country_code'] = df['Country_code'].replace(['NZ'], 'NZL')
df['Country_code'] = df['Country_code'].replace(['OM'], 'OMN')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['PA'], 'PAN')
df['Country_code'] = df['Country_code'].replace(['PE'], 'PER')
df['Country_code'] = df['Country_code'].replace(['PF'], 'PYF')
df['Country_code'] = df['Country_code'].replace(['PG'], 'PNG')
df['Country_code'] = df['Country_code'].replace(['PH'], 'PHL')
df['Country_code'] = df['Country_code'].replace(['PK'], 'PAK')
df['Country_code'] = df['Country_code'].replace(['PL'], 'POL')
df['Country_code'] = df['Country_code'].replace(['PM'], 'SPM')
df['Country_code'] = df['Country_code'].replace(['PN'], 'PCN')
df['Country_code'] = df['Country_code'].replace(['PR'], 'PRI')
df['Country_code'] = df['Country_code'].replace(['PS'], 'PSE')
df['Country_code'] = df['Country_code'].replace(['PT'], 'PRT')
df['Country_code'] = df['Country_code'].replace(['PW'], 'PLW')
df['Country_code'] = df['Country_code'].replace(['PY'], 'PRY')
df['Country_code'] = df['Country_code'].replace(['QA'], 'QAT')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['RE'], 'REU')
df['Country_code'] = df['Country_code'].replace(['RO'], 'ROU')
df['Country_code'] = df['Country_code'].replace(['RS'], 'SRB')
df['Country_code'] = df['Country_code'].replace(['RU'], 'RUS')
df['Country_code'] = df['Country_code'].replace(['RW'], 'RWA')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['SA'], 'SAU')
df['Country_code'] = df['Country_code'].replace(['SB'], 'SLB')
df['Country_code'] = df['Country_code'].replace(['SC'], 'SYC')
df['Country_code'] = df['Country_code'].replace(['SD'], 'SDN')
df['Country_code'] = df['Country_code'].replace(['SE'], 'SWE')
df['Country_code'] = df['Country_code'].replace(['SG'], 'SGP')

```



```

df['Country_code'] = df['Country_code'].replace(['SH'], 'SHN')
df['Country_code'] = df['Country_code'].replace(['SI'], 'SVN')
df['Country_code'] = df['Country_code'].replace(['SJ'], 'SJM')
df['Country_code'] = df['Country_code'].replace(['SK'], 'SVK')
df['Country_code'] = df['Country_code'].replace(['SL'], 'SLE')
df['Country_code'] = df['Country_code'].replace(['SM'], 'SMR')
df['Country_code'] = df['Country_code'].replace(['SN'], 'SEN')
df['Country_code'] = df['Country_code'].replace(['SO'], 'SOM')
df['Country_code'] = df['Country_code'].replace(['SR'], 'SUR')
df['Country_code'] = df['Country_code'].replace(['SS'], 'SSD')
df['Country_code'] = df['Country_code'].replace(['ST'], 'STP')
df['Country_code'] = df['Country_code'].replace(['SV'], 'SLV')
df['Country_code'] = df['Country_code'].replace(['SX'], 'SXM')
df['Country_code'] = df['Country_code'].replace(['SY'], 'SYR')
df['Country_code'] = df['Country_code'].replace(['SZ'], 'SWZ')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['TC'], 'TCA')
df['Country_code'] = df['Country_code'].replace(['TD'], 'TCD')
df['Country_code'] = df['Country_code'].replace(['TF'], 'ATF')
df['Country_code'] = df['Country_code'].replace(['TG'], 'TGO')
df['Country_code'] = df['Country_code'].replace(['TH'], 'THA')
df['Country_code'] = df['Country_code'].replace(['TZ'], 'TZA')
df['Country_code'] = df['Country_code'].replace(['TJ'], 'TJK')
df['Country_code'] = df['Country_code'].replace(['TK'], 'TKL')
df['Country_code'] = df['Country_code'].replace(['TL'], 'TLS')
df['Country_code'] = df['Country_code'].replace(['TM'], 'TKM')
df['Country_code'] = df['Country_code'].replace(['TN'], 'TUN')
df['Country_code'] = df['Country_code'].replace(['TO'], 'TON')
df['Country_code'] = df['Country_code'].replace(['TR'], 'TUR')
df['Country_code'] = df['Country_code'].replace(['TT'], 'TTO')
df['Country_code'] = df['Country_code'].replace(['TV'], 'TUV')
df['Country_code'] = df['Country_code'].replace(['TW'], 'TWN')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['UA'], 'UKR')
df['Country_code'] = df['Country_code'].replace(['UG'], 'UGA')
df['Country_code'] = df['Country_code'].replace(['US'], 'USA')
df['Country_code'] = df['Country_code'].replace(['UY'], 'URY')
df['Country_code'] = df['Country_code'].replace(['UZ'], 'UZB')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['VA'], 'VAT')
df['Country_code'] = df['Country_code'].replace(['VC'], 'VCT')
df['Country_code'] = df['Country_code'].replace(['VE'], 'VEN')

```

```

df['Country_code'] = df['Country_code'].replace(['VG'], 'VGB')
df['Country_code'] = df['Country_code'].replace(['VI'], 'VIR')
df['Country_code'] = df['Country_code'].replace(['VN'], 'VNM')
df['Country_code'] = df['Country_code'].replace(['VU'], 'VUT')
# Sustituyendo de la columna 'Country_code' los códigos de país alpha-2 por
código alpha-3
df['Country_code'] = df['Country_code'].replace(['WF'], 'WLF')
df['Country_code'] = df['Country_code'].replace(['WS'], 'WSM')
df['Country_code'] = df['Country_code'].replace(['YE'], 'YEM')
df['Country_code'] = df['Country_code'].replace(['YT'], 'MYT')
df['Country_code'] = df['Country_code'].replace(['ZA'], 'ZAF')
df['Country_code'] = df['Country_code'].replace(['ZM'], 'ZMB')
df['Country_code'] = df['Country_code'].replace(['ZW'], 'ZWE')
# rectificando valores nulos
country_null = df[df.Country_code.isnull()]
country_null.head()
df['Country_code'] = df['Country_code'].replace([np.nan], 'NAM')
# rectificando valores nulos
country_null = df[df.Country_code.isnull()]
country_null.head()
# exportando DataFrame sin índices a un nuevo archivo csv con códigos de
país alpha-3
df.to_csv("WHO-COVID-19-global-data-alpha3.csv", index = False)
# Agrupando países
dfSummary = df.groupby(['Country_code'])[['New_cases',
'New_deaths']].apply(sum)
dfSummary.head()
# exportando DataFrame con índices a un nuevo archivo csv con códigos de
país alpha-3
dfSummary.to_csv("WHO-COVID-19-summary.csv", index= True)

```

## 2. index.html

```

<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Load d3.js -->
  <script src="https://d3js.org/d3.v6.js"></script>
  <script defer src="resources/main.js"></script>

```

```

    <!-- CSS only -->
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.cs
s" rel="stylesheet"
    integrity="sha384-
1BmE4kWBq78iYhF1dvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
crossorigin="anonymous">
    <link rel="stylesheet" type="text/css" href="resources/main.css" />
    <title>Tendencias en D3</title>
</head>

<body>

    <div class="collapse" id="navbarToggleExternalContent">
        <div class="bg-dark p-4">
            <h5 class="text-white h4">Collapsed content</h5>
            <span class="text-muted">Toggleable via the navbar brand.</span>
        </div>
    </div>
    <nav class="navbar navbar-dark bg-dark">
        <div class="container-fluid">
            <button class="navbar-toggler" type="button" data-bs-
toggle="collapse"
                data-bs-target="#navbarToggleExternalContent" aria-
controls="navbarToggleExternalContent"
                aria-expanded="false" aria-label="Toggle navigation">
                <span class="navbar-toggler-icon"></span>
            </button>
        </div>
    </nav>
    <div>
        <h2>
            COVID-19
        </h2>
    </div>
    <br>
    <div class="left">
        <!-- Create an element where the map will take place -->
        <svg id="my_dataviz" width="700" height="600">
        </svg>

        <div class="datascale" id="datascale">
        </div>
    </div>
    <div class="right" id="barplot">

```

```

    <p id="barplotTitle"></p>
    <svg id="my_barplot">
    </svg>
    <table class="table">
      <thead>
        <tr>
          <th scope="col"></th>
          <th scope="col">Total</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <th scope="row">Contagiados</th>
          <td><p id="contagiados"></p></td>
        </tr>
        <tr>
          <th scope="row">Fallecidos</th>
          <td><p id="fallecidos"></p></td>
        </tr>
      </tbody>
    </table>
  </div>
</body>

</html>

```

### 3. main.css

```

body{
  text-align: center;
}
div{
  border-radius: 10px;
}

p{
  margin: 0px;
}

h2{
  font-weight: bold;
}

.left{
  float: left;
}

```

```

    position: relative;
    width: 60%;
    padding: 5px;
}

.right{
    background-color: beige;
    text-align: center;
    float: left;
    position: relative;
    width: 40%;
    padding: 5px;
}

.legendTitle{
    font-size: 15px;
    font-weight: bold;
}

.legendAxis{
    font-size: 9px;
}

```

#### 4. main.js

```

/*
declarando constante COVID que contendrá el resumen general de contagiados
y fallecidos
*/
COVID = []
// variable que albergará el país seleccionado del mapa
sumary = [
    { group: "Contagiados", value: 1 },
    { group: "Fallecidos", value: 1 }
]
// leyendo archivo CSV y cargando su contenido al array COVID
d3.csv("WHO-COVID-19-summary.csv").then((res) => {
    res.forEach(element => {
        COVID.push(element)
    });
})

// Definiendo parámetros de ancho y alto para del SVG del mapa
const svg = d3.select("svg"),
    width = +svg.attr("width"),
    height = +svg.attr("height");

```

```

// Realizando la proyección y el escalado
const path = d3.geoPath();
const projection = d3.geoMercator()
  .scale(115)
  .center([0, 20])
  .translate([width / 2, height / 2]);

// Definiendo la escala de colores para el mapa
// La escala de color responde a la cantidad de contagiados por país
const data = new Map();
const colorScale = d3.scaleThreshold()
  .domain([100000, 500000, 1000000, 5000000, 10000000, 25000000,
50000000])
  .range(d3.schemeReds[5]);

//*****Begin: Legenda del Mapa *****/
const legendWidth = Math.min(width * 0.50, 400);
// Asignando el svg para la escala de datos
const svg_legend = svg.append("g")
  .attr("class", "legendWrapper")
  .attr("transform",
    "translate(" + (width * 0.73) + "," + (16) + ")");

//*****End: Legenda del Mapa *****/

//*****BAR PLOT*****/
// definiendo margenes y ancho y largo del barplot
const margin = {
  top: 30,
  right: 30,
  bottom: 70,
  left: 60
},
width_bp = 460 - margin.left - margin.right,
height_bp = 400 - margin.top - margin.bottom;

// Asignando el svg para el barplot sobre el div #barplot
const svg_bp = d3.select("#my_barplot")
  // .append("svg")
  .attr("width", width_bp + margin.left + margin.right)
  .attr("height", height_bp + margin.top + margin.bottom)
  .append("g")
  .attr("transform",

```

```

    `translate(${margin.left},${margin.top})`);

// Inicializando valores para el eje X
const x = d3.scaleBand()
    .range([0, width_bp])
    .padding(0.2);
const xAxis = svg_bp.append("g")
    .attr("transform", `translate(0,${height_bp})`);

// Inicializando valores para el eje Y
const y = d3.scaleLinear()
    .range([height_bp, 0]);
const yAxis = svg_bp.append("g")
    .attr("class", "myYaxis");

//*****BAR PLOT*****/

// Cargando información externa
Promise.all([
    //Obteniendo todas las coordenadas del póligo para cada país del mundo
    d3.json("https://raw.githubusercontent.com/holtzy/D3-graph-gallery/master/DATA/world.geojson"),
    //Obteniendo el resumen de reportes y fallecidos por país
    d3.csv("WHO-COVID-19-summary.csv", function (d) {
        data.set(d.Country_code, +d.New_cases)
    })).then(function (loadData) {
        let topo = loadData[0]

        let mouseOver = function (d) {
            d3.selectAll(".Country")
                .transition()
                .duration(200)
                .style("opacity", .5)
            d3.select(this)
                .transition()
                .duration(200)
                .style("opacity", 1)
                .style("stroke", "black")
        }

        let mouseLeave = function (d) {
            d3.selectAll(".Country")
                .transition()

```

```

        .duration(200)
        .style("opacity", .8)
    d3.select(this)
        .transition()
        .duration(200)
        .style("stroke", "transparent")
    }

    // Dibujando el mapa
    svg.append("g")
        .selectAll("path")
        .data(topo.features)
        .enter()
        .append("path")
        // Dibujando cada país
        .attr("d", d3.geoPath()
            .projection(projection)
        )
        // Asignando color a cada país
        .attr("fill", function (d) {
            d.total = data.get(d.id) || 0;
            return colorScale(d.total);
        })
        .style("stroke", "transparent")
        .attr("class", function (d) { return "Country" })
        .style("opacity", .8)
        //Asignando evento cuando el puntero del mouse está sobre el
        área cada polígono del país
        .on("mouseover", mouseOver)
        //Asignando evento cuando el puntero se retira del área del
        polígono del país
        .on("mouseleave", mouseLeave)
        //Cuando el usuario proporciona click sobre el polígono se
        redibuja el barplot
        .on("click", function (d, i) {
            drawBarplot(i.id);
            drawLegendPointer(i.id);
        })

    // Dibuja la legenda del Mapa
    drawMapLegend();
})

function drawBarplot(i) {
    //Se busca la información relacionada con el país seleccionando

```



```

//dentro del array inicial COVID
let res = COVID.find(pais => pais.Country_code === i)
//Limpia el array
sumary = []
//Asignando los elementos encontrados en la búsqueda
//al array sumary
sumary.push(
  {
    'group': 'Contagiados',
    'value': res.New_cases
  },
  {
    'group': 'Fallecidos',
    'value': res.New_deaths
  }
)

// Actualizando el eje x en función del país seleccionado
x.domain(sumary.map(d => d.group))
xAxis.call(d3.axisBottom(x))

// Actualizando el eje y en función del país seleccionado
// reescalando el eje dependiendo de la cantidad de personas contagiadas
var contagiados = parseInt(sumary[0].value, 10)
if (contagiados <= 100000) {
  max = 100000
} else if (contagiados > 100000 && contagiados <= 500000) {
  max = 500000
} else if (contagiados > 500000 && contagiados <= 1000000) {
  max = 1000000
} else if (contagiados > 1000000 && contagiados <= 5000000) {
  max = 5000000
} else if (contagiados > 5000000 && contagiados <= 10000000) {
  max = 10000000
} else if (contagiados > 10000000 && contagiados <= 25000000) {
  max = 25000000
} else if (contagiados > 25000000 && contagiados <= 50000000) {
  max = 50000000
}
y.domain([0, max]);
// Asignando animación
yAxis.transition().duration(1000).call(d3.axisLeft(y));

var u = svg_bp.selectAll("rect")
  .data(sumary)

```

```

//Dibujando las barras y asignando animaciones de transición
u.join("rect")
    .transition()
    .duration(1000)
    .attr("x", d => x(d.group))
    .attr("y", d => y(d.value))
    .attr("width", x.bandwidth())
    .attr("height", d => height_bp - y(d.value))
    .attr("fill", "rgb(223, 163, 0, 0.863)")

//Asignando dinámicamente el título del gráfico
d3.select("#barplotTitle")
    .text(res.Country)
    .attr("class", "legendTitle")

//Asignando dinámicamente el resumen del país
d3.select("#contagiados")
    .text(parseInt(res.New_cases).toLocaleString('es-MX'))
d3.select("#fallecidos")
    .text(parseInt(res.New_deaths).toLocaleString('es-MX'))
}

// Función encargada de dibujar la leyenda del mapa
function drawMapLegend() {
    /* -- Definición del gradiente de forma lineal -- */
    var def_lg = svg.append("defs")
        .append("linearGradient")
        .attr("id", "def_linear_gradient")
        .attr("x1", "0%").attr("y1", "0%")
        .attr("x2", "100%").attr("y2", "0%");
    // Color de finalización del gradiente
    def_lg.append("stop")
        .attr("offset", "0%")
        .style("stop-color", d3.max(colorScale.range()))
        .style("stop-opacity", 1)
    // Color de inicio del gradiente
    def_lg.append("stop")
        .attr("offset", "100%")
        .style("stop-color", d3.min(colorScale.range()))
        .style("stop-opacity", 1)
    /* -- */

    /* -- Dibujando el rectángulo -- */
    svg_legend.append("rect")
        .attr("class", "legendRect")

```

```

        .attr("x", -legendWidth / 2)
        .attr("y", 0)
        .attr("width", legendWidth)
        .attr("height", 5)
        .style("fill", "url(#def_linear_gradient)");
    /* -- */

    /* -- Título de la leyenda -- */
    svg_legend.append("text")
        .attr("class", "legendTitle")
        .attr("x", 0)
        .attr("y", -5)
        .style("text-anchor", "middle")
        .text("03/01/2020 al 29/11/2021");
    /* -- */

    /* -- Leyenda de la escala de los datos -- */
    // Escala para cantidades
    var Leg_xScale = d3.scaleLinear()
        .domain([0, 50000000])
        .range([-legendWidth / 2, legendWidth / 2])
    // Axis para dar formato a los datos (abreviación de Millones)
    var Leg_xAxis = d3.axisBottom(Leg_xScale).tickFormat(function (d) {
        return d / 1000000 + "MM";
    })

    //Configura el eje X
    svg_legend.append("g")
        .attr("class", "legendAxis")
        .attr("transform", "translate(0," + (5) + ")")
        .call(Leg_xAxis);
    /* -- */
}

// Función drawLegendPointer: dibuja sobre la escala de color
// un puntero con la cantidad de casos para una rápida identificación.
function drawLegendPointer(i) {
    // Filtra los datos específicos del país seleccionado
    let res = COVID.find(pais => pais.Country_code === i);
    // El valor a representar en el apuntados, es la suma de los nuevos
    casos
    var value = parseInt(res.New_cases) + parseInt(res.New_deaths);

    /* -- Leyenda de la escala de los datos -- */
    var PtScale = d3.scaleLinear()

```

```

        .range([-legendWidth / 2, legendWidth / 2])
        .domain([0, 50000000]);
    // Definición del objeto apuntador (Diamante)
    var symb = d3.symbol().type(d3.symbolDiamond).size(30);
    // Dibuja el apuntador
    svg_legend.selectAll("path")
        .attr("d", symb)
        .attr("fill", "black")
        .attr("transform", "translate(" + PtScale(value) + ", -2)");
}

function showTitleBarPlot(res) {
}

```

## Conclusiones

1. Los paquetes de pandas y numpy de Python proveen las herramientas necesarias para la correcta preparación de un dataset.
2. Un mapa coroplético es un mapa temático coloreado bajo la misma gama cromática de un color.
3. Para dibujar un mapa con D3 es necesario obtener todos los puntos de geolocalización del país para trazar el polígono respectivo.
4. La librería D3 cuenta con una amplia gama de ejemplos para el correcto desarrollo de visualizaciones.
5. D3 permite asociar múltiples eventos a una visualización.
6. El evento “mouseover” hace referencia al momento en que el puntero del mouse está sobre un elemento en particular.
7. El evento “mouseleave” hace referencia al momento en que el puntero del mouse se retira del área del elemento sobre el que se posicionó.
8. Los eventos manejados por javascript son los mismos que se pueden asociar directamente con la librería D3.js.

## Recomendaciones

1. Independientemente de la herramienta de visualización que se utilice no hay que olvidar los principios gestaltistas para el correcto desarrollo de la visualización.
2. Considerar que el correcto proceso para desarrollo de una visualización consta de Investigación, Selección de datos, Creación del Boceto, Elaboración, Percepción y Conocimiento.
3. Para crear un mapa con D3 es necesario utilizar todos los puntos geolocalización de los países, una alternativa para este caso particular es el siguiente enlace:  
<https://raw.githubusercontent.com/holtzy/D3-graph-gallery/master/DATA/world.geojson>
4. Para profundizar sobre los eventos que se pueden asociar a una visualización es recomendable considerar la documentación correcta, como por ejemplo <https://developer.mozilla.org/es/docs/Web/Events>

## Bibliografía

- Torres, D. (2009). Aproximaciones a la visualización como disciplina científica. *ACIMED*, 20(6). Obtenido de [http://scielo.sld.cu/scielo.php?script=sci\\_arttext&pid=S1024-94352009001200005](http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352009001200005)
- UNIR. (2020). *Tema 6: Definición y tipologías de gráficos*. (p. 2-27). Visualización Interactiva de la Información. Material no publicado.