



**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

**ALGORITMO COMPUTACIONAL PARA EL ANÁLISIS Y TOMA DE DECISIONES  
EN DATOS DE LUGARES CON MAYOR INCIDENCIA DE COVID-19, UTILIZANDO  
ESTRATEGIAS DE MACHINE LEARNING**

Corporación Universitaria Remington.  
Facultad de ingeniería  
Tecnología en desarrollo de software

Estudiantes:  
Oscar Castro.  
Tutor: Juan Carlos Briñez de León  
Opción de Trabajo de grado Seminario-Diplomado.  
2024.

## **Dedicatoria**

A mis padres, cuyo amor y sacrificio han sido la brújula que me guiaron durante todo este viaje universitario. A mis hermanos, por su apoyo y por ser mi fuente de fortaleza en los momentos de adversidad. Y sobre todo a esas personas que, con su presencia y ánimo, hicieron posible que llegara tan lejos, les dedico este logro con profundo agradecimiento y gratitud. Que este trabajo sea testimonio de la huella que cada uno ha dejado en mi vida y en mi formación como persona y profesional.

## Tabla de Contenidos

### Contenido

Tabla de Contenidos.....	3
Resumen .....	4
Palabras clave .....	4
Marco conceptual y contextual.....	5
Pregunta problema: .....	6
Acercamiento a los datos:.....	6
Objetivos:.....	6
Objetivo general.....	6
Objetivos específicos.....	7
.....	10
Procesamiento de los datos.....	12
Modelo de toma de decisiones.....	15
Conclusiones.....	19
Referencias .....	20

## Resumen

La propagación descontrolada del COVID-19 en China, Italia, Irán y la República de Corea resultó en un porcentaje más alto de casos en comparación con países como Francia, España, Alemania, Emiratos Árabes Unidos y otras naciones europeas. Varios factores contribuyeron a esta notable diferencia en las tasas de incidencia.

En China, donde se originó el brote, la falta de preparación inicial y la demora en implementar medidas de contención efectivas permitieron que el virus se propagara rápidamente. La alta densidad poblacional en algunas regiones y la intensa movilidad interna fueron facilitadores adicionales de la transmisión (Huang et al., 2020).

Italia fue uno de los primeros países europeos en registrar un gran número de casos debido a la tardanza en imponer medidas de distanciamiento social, la alta proporción de población anciana vulnerable y la arraigada cultura de reuniones familiares frecuentes (Remuzzi & Remuzzi, 2020).

En Irán, las autoridades fueron lentas en reconocer la gravedad de la situación y tardaron en implementar medidas de contención adecuadas. Además, la celebración de eventos masivos y la negativa inicial a cancelar las peregrinaciones religiosas aceleraron la propagación del virus (Abdi, 2020).

La República de Corea experimentó un brote significativo relacionado con la Iglesia de Shincheonji, cuyos miembros se reunían en lugares cerrados y algunos ocultaron inicialmente sus afiliaciones, dificultando el rastreo de contactos (Lee et al., 2020).

En contraste, países como Francia, España, Alemania y los Emiratos Árabes Unidos, si bien tuvieron un número considerable de casos, implementaron medidas de contención más rápidamente y contaban con sistemas de salud más preparados, lo que pudo haber contribuido a una menor tasa de incidencia en comparación con los primeros países mencionados (Legido-Quigley et al., 2020).

## Palabras clave

Covid-19, Datasets, Análisis de datos, Machine learning, Clasificación, regresión, Clustering.

## Marco conceptual y contextual

La propagación de enfermedades infecciosas como el COVID-19 está influenciada por diversos factores, que incluyen aspectos sociales, culturales, demográficos, políticos y de preparación sanitaria. Comprender estos factores es crucial para analizar las diferencias en las tasas de incidencia entre países y regiones, y para mejorar la respuesta ante futuras pandemias. La propagación de la infecciosas como el COVID-19 no es cuestión de pura suerte. Hay varios factores de por medio que influyeron y tenerlos claros es la clave para entender por qué en unos países se desmadró más que en otros.

Los factores socioculturales jugaron un papel bien malo. Porque es que las costumbres y tradiciones de algunas sociedades le facilitaron al virus regarse como pólvora. Piensen en esas rumbas y fiestas con toda esa gente amontonada, o las reuniones familiares de los domingos donde todo el mundo se abraza y se da besos.

También está lo demográfico. Es lógico que en sitios repoblados y donde la gente se la pasa en la calle para arriba y para abajo, pues el riesgo de contagio aumenta un montón. Y ni se diga de los países con hartas personas viejitas, que son las más vulnerables si les da duro.

Por otro lado, la preparación y respuesta de los gobiernos fue decisiva. Hubo naciones que se hicieron los locos y se demoraron en ponerle el cuidado a la situación, o peor, nunca tomaron medidas drásticas. Ahí sí que la embarraron con su falta de transparencia y de actuar.

En China, por ejemplo, siendo el inicio de todo, les fue muy mal por no estar listos y tardarse un rato en controlar esa vaina. Sumado al gentío y la movilidad que hay por allá, era un horno listo para la corona.

Los italianos tampoco se salvaron. Como eran de los primeros afectados en Europa, ya hubieran debido reaccionar más rápido con el distanciamiento y todo el cuento. Pero con lo uniditos que son y la cultura de las comidas familiares pues no les fue muy bien tampoco. En Irán tampoco ayudó que se hicieran los loros y tardaran en tomar cartas en el asunto. Encima permitieron tanta peregrinación religiosa masiva, que fue como echarle más leña al fuego.

Corea del Sur también la supo embarrar con ese brote por la tal Iglesia de Shincheonji, donde los miembros se reunían a cantarle al Señor encerraditos. Para rematar, algunos ocultaron que asistían, complicando todo ese relajó.

En cambio, países como Francia, España, Alemania y los Emiratos actuaron más rápido para contener el lío y contaban con sistemas de salud más robustos, entonces la vaca no los pateó tan duro como a los otros.

En fin, que todos estos factores sociales, culturales y demás tienen que quedarnos bien claros, para que, si llega a pasar de nuevo una chambonada así, sepamos cómo mitigar el daño de manera más efectiva.

**Pregunta problema:**

¿Cómo desarrollar una estrategia computacional para ayudar en Los Lugares con Mayor Incidencia de COVID-19, a partir de algoritmos de Machine Learning?

**Acercamiento a los datos:**

- recopilar datos relevantes sobre la incidencia de COVID-19. Esto puede incluir datos de casos confirmados, pruebas realizadas, tasas de positividad, datos demográficos, movilidad de la población, medidas de contención implementadas, entre otros.
- Una vez que tengas los datos, es necesario limpiarlos y prepararlos para su análisis. Esto puede implicar la eliminación de valores atípicos, la imputación de datos faltantes, la normalización de variables y la codificación de variables categóricas, entre otros pasos.
- Identifica las características más relevantes que puedan influir en la incidencia de COVID-19 en un lugar determinado.

**Objetivos:**

**Objetivo general.**

El objetivo principal de este trabajo es analizar y visualizar datos relacionados con la incidencia de COVID-19 en diferentes países de todo el mundo, utilizando conjuntos de datos disponibles en Kaggle. A través del análisis de estos datos, se busca identificar y

determinar los países que han experimentado la mayor presencia de COVID-19 durante el transcurso de la pandemia, utilizando estrategias de machine learning.

### **Objetivos específicos.**

- Recopilar conjuntos de datos relevantes sobre la incidencia de COVID-19 a nivel mundial disponibles en Kaggle, que incluyan información detallada sobre casos confirmados, fallecimientos, recuperaciones y otras métricas relevantes.

### **Desarrollo e implementación del aprendizaje**

El conjunto de datos nos muestra diferentes variables entre ellas están:

País: Lista de países visitados por la persona.

Edad: Clasificación del grupo de edad según los estándares de la OMS.

Síntomas: Los cinco síntomas principales del COVID-19 como por ejemplo fiebre, cansancio, dificultad para respirar, tos seca y dolor de garganta).

Otros síntomas: Indica si la persona experimenta otros síntomas como dolores, congestión nasal, secreción nasal o diarrea.

Gravedad: El nivel de gravedad de los síntomas (leve, moderado o grave).

Contacto: Si la persona ha estado en contacto con otro paciente con COVID-19.

Enfermedad por coronavirus: Esta es la variable objetivo que indica si la persona tiene o no la enfermedad por coronavirus.

Para llevar a cabo dicha investigación se cargaron los datos para dicho análisis y recopilaron los datos.

```
Cargando archivos desde el pc

[.] Import pandas as pd
from google.colab import files
uploaded = files.upload()
for filename in uploaded.keys():
    Dataset = pd.read_csv(filename, sep=',')

Dataset.head(7)
```

	Fever	Tiredness	Dry-Cough	Difficulty-in-Breathing	Sore-Throat	None_Sympton	Pains	Nasal-Congestion	Runny-Nose	Diarrhea	...	Gender_Male	Gender_Transgender	Severity_Mild	Severity_Moderate	Severity_None	Severity_Severe	Contact_Dont-Know	Contact_No	Contact_Yes	Country
0	1	1	1	1	1	0	1	1	1	1	...	1	0	1	0	0	0	0	0	1	China
1	1	1	1	1	1	0	1	1	1	1	...	1	0	1	0	0	0	0	1	0	China
2	1	1	1	1	1	0	1	1	1	1	...	1	0	1	0	0	0	1	0	0	China
3	1	1	1	1	1	0	1	1	1	1	...	1	0	0	1	0	0	0	0	1	China
4	1	1	1	1	1	0	1	1	1	1	...	1	0	0	1	0	0	0	1	0	China
5	1	1	1	1	1	0	1	1	1	1	...	1	0	0	1	0	0	1	0	0	China
6	1	1	1	1	1	0	1	1	1	1	...	1	0	0	0	0	1	0	0	1	China

Donde pasamos a ver Información de la estructura de dataset.

```
Dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 316800 entries, 0 to 316799
Data columns (total 27 columns):
#   Column                                     Non-Null Count  Dtype
---  ---                                     -
0   Fever                                     316800 non-null  int64
1   Tiredness                                316800 non-null  int64
2   Dry-Cough                                316800 non-null  int64
3   Difficulty-in-Breathing                  316800 non-null  int64
4   Sore-Throat                              316800 non-null  int64
5   None_Sympton                             316800 non-null  int64
6   Pains                                     316800 non-null  int64
7   Nasal-Congestion                         316800 non-null  int64
8   Runny-Nose                               316800 non-null  int64
9   Diarrhea                                  316800 non-null  int64
10  None_Experiencing                        316800 non-null  int64
11  Age_0-9                                   316800 non-null  int64
12  Age_10-19                                 316800 non-null  int64
13  Age_20-24                                 316800 non-null  int64
14  Age_25-59                                 316800 non-null  int64
15  Age_60+                                   316800 non-null  int64
16  Gender_Female                             316800 non-null  int64
17  Gender_Male                               316800 non-null  int64
18  Gender_Transgender                       316800 non-null  int64
19  Severity_Mild                             316800 non-null  int64
20  Severity_Moderate                        316800 non-null  int64
21  Severity_None                             316800 non-null  int64
22  Severity_Severe                           316800 non-null  int64
23  Contact_Dont-Know                        316800 non-null  int64
24  Contact_No                                316800 non-null  int64
25  Contact_Yes                               316800 non-null  int64
26  Country                                   316800 non-null  object
dtypes: int64(26), object(1)
memory usage: 65.3+ MB
```

Para luego seguir con la limpieza de los datos al igual que eliminar duplicados o identificarlos de igual forma se eliminan al igual se corrigen errores del formato para poder llevar a cabo el análisis en este caso analizamos la columna Country el cual nos arroja datos como variables de tipo texto, pero como no podemos graficar nombres debemos convertirlos a números para que sea más fácil

```

Limpieza de datos

[] # Identificar valores faltantes
print(Dataset.isnull().sum())

# Eliminar filas con valores faltantes
Dataset = Dataset.dropna()
Dataset.head(7)

Fever 0
Tiredness 0
Dry-Cough 0
Difficulty-in-breathing 0
Sore-Throat 0
None_Symptom 0
Pains 0
Nasal-Congestion 0
Runny-Nose 0
Diarrhea 0
... Gender_Male 0
Gender_Transgender 0
Severity_Mild 0
Severity_Moderate 0
Severity_None 0
Severity_Severe 0
Contact_Dont-Know 0
Contact_No 0
Contact_Yes 0
Country
other: 1064

```

	Fever	Tiredness	Dry-Cough	Difficulty-in-breathing	Sore-Throat	None_Symptom	Pains	Nasal-Congestion	Runny-Nose	Diarrhea	...	Gender_Male	Gender_Transgender	Severity_Mild	Severity_Moderate	Severity_None	Severity_Severe	Contact_Dont-Know	Contact_No	Contact_Yes	Country
0	1	1	1	1	1	0	1	1	1	1	...	1	0	1	0	0	0	0	0	1	China
1	1	1	1	1	1	0	1	1	1	1	...	1	0	1	0	0	0	0	1	0	China
2	1	1	1	1	1	0	1	1	1	1	...	1	0	1	0	0	0	1	0	0	China
3	1	1	1	1	1	0	1	1	1	1	...	1	0	1	0	0	0	0	0	1	China
4	1	1	1	1	1	0	1	1	1	1	...	1	0	0	1	0	0	0	1	0	China
5	1	1	1	1	1	0	1	1	1	1	...	1	0	0	1	0	0	1	0	0	China
6	1	1	1	1	1	0	1	1	1	1	...	1	0	0	0	0	1	0	0	1	China

```

[] # Corregir errores de formato (por ejemplo, convertir a mayúsculas)
Dataset['Country'] = Dataset['Country'].str.lower()
Dataset.head()

# Realizando opciones de variable categóricas
Dataset['Country'].unique()

array(['china', 'italy', 'iran', 'republic of korea', 'france', 'spain',
       'germany', 'usa', 'other-eur', 'other'], dtype=object)

[] # Convirtiendo a número
Conversion = {'china':1, 'italy':2, 'iran':3, 'republic of korea':4, 'france':5, 'spain':6, 'germany':7, 'usa':8, 'other-eur':9, 'other':10}
Dataset['Country'] = Dataset['Country'].replace(Conversion)
Dataset.head(7)

```

	Fever	Tiredness	Dry-Cough	Difficulty-in-breathing	Sore-Throat	None_Symptom	Pains	Nasal-Congestion	Runny-Nose	Diarrhea	...	Gender_Female	Gender_Male	Severity_Mild	Severity_Moderate	Severity_None	Severity_Severe	Contact_Dont-Know	Contact_No	Contact_Yes	Country
0	1	1	1	1	1	0	1	1	1	1	...	0	1	1	0	0	0	0	0	1	1
1	1	1	1	1	1	0	1	1	1	1	...	0	1	1	0	0	0	0	1	0	1
2	1	1	1	1	1	0	1	1	1	1	...	0	1	1	0	0	0	1	0	0	1
3	1	1	1	1	1	0	1	1	1	1	...	0	1	0	1	0	0	0	0	1	1
4	1	1	1	1	1	0	1	1	1	1	...	0	1	0	1	0	0	0	1	0	1
5	1	1	1	1	1	0	1	1	1	1	...	0	1	0	1	0	0	1	0	0	1
6	1	1	1	1	1	0	1	1	1	1	...	0	1	0	0	0	1	0	0	1	1

7 rows x 26 columns

Aplicando estrategias de Clustering para la segmentación de los datos seguidos de sus centroides el modelo

```

[] # Preparando los datos para el clustering
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Transformando el dataset a formato numérico
Dataset_numerico = np.array(Dataset)

# Agrupar el número de grupos
k = 8

# Normalizando y entrenando el modelo
Modelo_Cluster = KMeans(n_clusters=k, init='k-means++', random_state=42)
Modelo_Cluster.fit(Dataset_numerico)

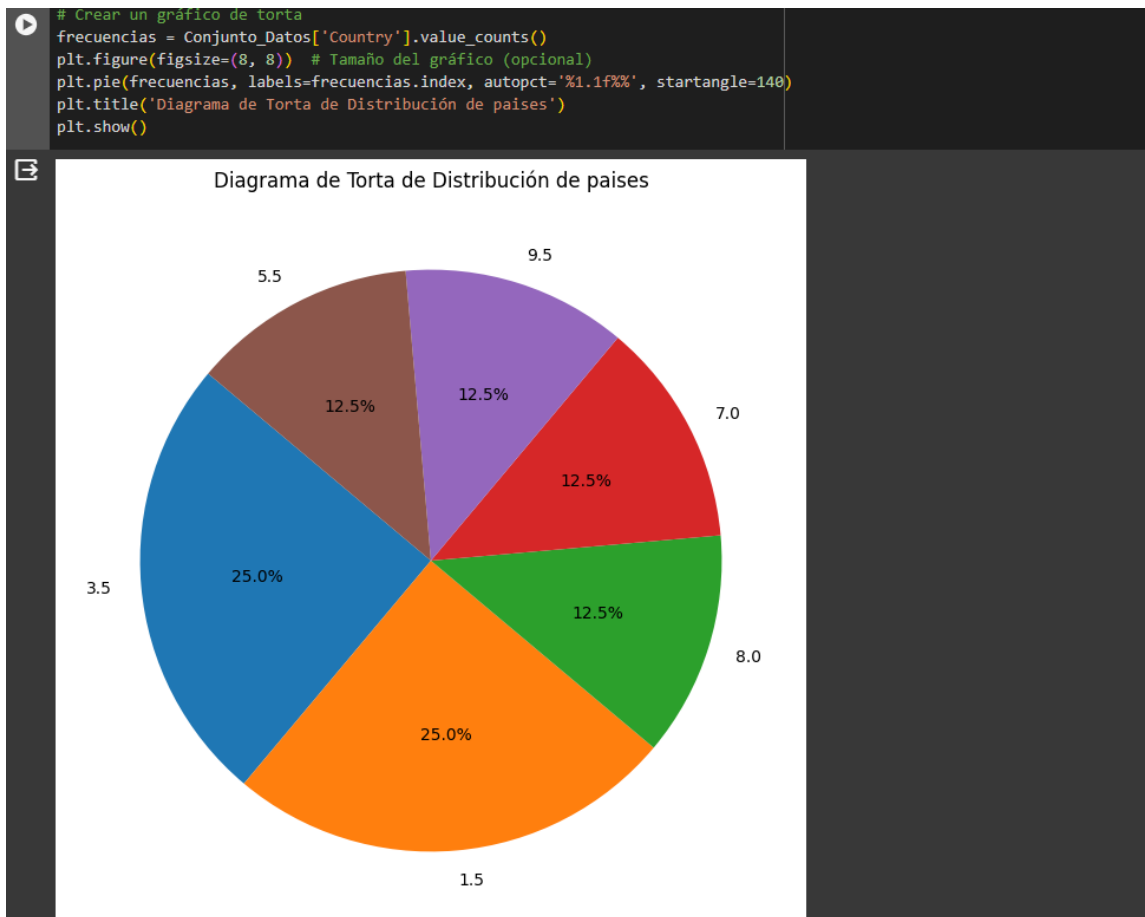
# Prediciendo el cluster de grupos en el conjunto de datos
Predicciones = Modelo_Cluster.fit_predict(Dataset_numerico)

# Agregando el cluster como columna nueva al dataset
Dataset_2 = Dataset.copy()
Dataset_2['Grupo'] = Predicciones
Dataset_2.head(12)

```

	Fever	Tiredness	Dry-Cough	Difficulty-in-breathing	Sore-Throat	None_Symptom	Pains	Nasal-Congestion	Runny-Nose	Diarrhea	...	Gender_Male	Severity_Mild	Severity_Moderate	Severity_None	Severity_Severe	Contact_Dont-Know	Contact_No	Contact_Yes	Country	Grupo
0	1	1	1	1	1	0	1	1	1	1	...	1	1	0	0	0	0	0	1	1	7
1	1	1	1	1	1	0	1	1	1	1	...	1	1	0	0	0	0	1	0	1	7
2	1	1	1	1	1	0	1	1	1	1	...	1	1	0	0	0	0	1	0	0	7
3	1	1	1	1	1	0	1	1	1	1	...	1	1	0	0	0	0	0	1	1	7
4	1	1	1	1	1	0	1	1	1	1	...	1	0	1	0	0	0	1	0	0	7
5	1	1	1	1	1	0	1	1	1	1	...	1	0	1	0	0	1	0	0	1	7
6	1	1	1	1	1	0	1	1	1	1	...	1	0	0	0	0	1	0	0	1	7
7	1	1	1	1	1	0	1	1	1	1	...	1	0	0	0	0	1	0	1	0	7
8	1	1	1	1	1	0	1	1	1	1	...	1	0	0	0	0	1	0	0	1	7
9	1	1	1	1	1	0	1	1	1	1	...	1	0	0	0	0	1	0	0	1	7
10	1	1	1	1	1	0	1	1	1	1	...	1	0	1	0	0	1	0	1	0	7
11	1	1	1	1	1	0	1	1	1	1	...	1	0	0	1	0	1	0	0	1	7
12	1	1	1	1	1	0	1	1	1	1	...	1	0	0	0	0	1	0	0	1	7

Para poder determinar cuáles son los lugares de el dataset con mayor índice de contagiados se realizaron gráficas y diagramas donde se denominan como: china:1, italy:2, iran:3 , republic of korean:4, france:5, spain:6 , germany:7 ,uae:8 , other-eur:9 ,other:10



La grafica de torta nos indica que los países 1.5 a 3.5 tienen un 25% lo que nos indica que son los lugares con más índice de contagios al igual que los otros diagramas para poder tener datos más claros y exactos



## Procesamiento de los datos

Para poder realizar un procesamiento de datos se inicia cargando y visualizando datos en Python donde podemos descargarlos desde paginas como kaggle donde después podemos usar la herramienta de Google colab donde se realice un programa para la manipulación de los datos en Python

```
#Para cargar los datos
import pandas as pd
from google.colab import files
uploaded = files.upload()
for filename in uploaded.keys():
    Conjunto_Datos = pd.read_csv(filename, sep=',')
Conjunto_Datos.head()
```

Elección de archivos: Sin archivos seleccionados Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.  
Saving diabetes (2).csv to diabetes (2) (4).csv

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Analizando los datos:

```
#Información de la estructura de datos
Conjunto_Datos.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  ---                ---
0   Pregnancies            768 non-null    int64
1   Glucose                768 non-null    int64
2   BloodPressure          768 non-null    int64
3   SkinThickness          768 non-null    int64
4   Insulin                768 non-null    int64
5   BMI                    768 non-null    float64
6   DiabetesPedigreeFunction 768 non-null    float64
7   Age                    768 non-null    int64
8   Outcome                768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Una vez teniendo los datos cargados podemos analizar o modificar el archivo dependiendo de la necesidad o el proceso a realizar ya sea quitando elementos como columnas o cambiando rango de valores.

```
#Análisis de los datos
Conjunto_Datos.describe()
```

	User ID	Age	AnnualSalary	Purchased
count	1000.000000	1000.000000	1000.000000	1000.000000
mean	500.500000	40.106000	72689.000000	0.402000
std	288.819436	10.707073	34488.341867	0.490547
min	1.000000	18.000000	15000.000000	0.000000
25%	250.750000	32.000000	46375.000000	0.000000
50%	500.500000	40.000000	72000.000000	0.000000
75%	750.250000	48.000000	90000.000000	1.000000
max	1000.000000	63.000000	152500.000000	1.000000

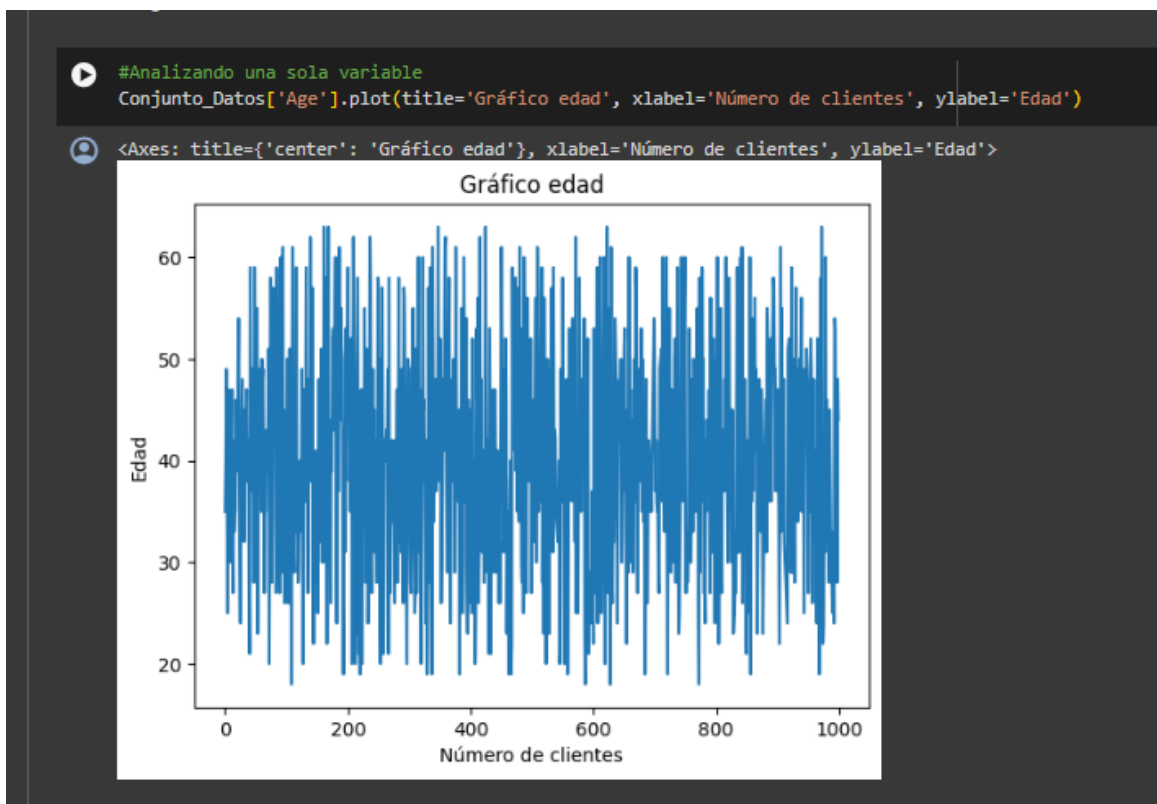
```
#Quitando columnas indeseadas
Conjunto_Datos = Conjunto_Datos.drop(['User ID'],axis=1)
#resumen de los datos
Conjunto_Datos.head(20)
```

	Gender	Age	AnnualSalary	Purchased
0	Male	35	20000	0
1	Male	40	43500	0
2	Male	49	74000	0
3	Male	40	107500	1
4	Male	25	79000	0
5	Female	47	33500	1
6	Female	46	132500	1
7	Male	42	64000	0
8	Female	30	84500	0
9	Male	41	52000	0
10	Male	42	80000	0
11	Male	47	23000	1
12	Female	32	72500	0
13	Female	27	57000	0
14	Female	42	108000	1
15	Female	33	149000	1
16	Male	35	75000	0
17	Male	35	53000	0
18	Male	46	79000	1
19	Female	39	134000	1

Utilizar el algoritmo anterior para cargar los datasets, identificar la información que contiene, describir estadísticamente, remover variables y hacer análisis

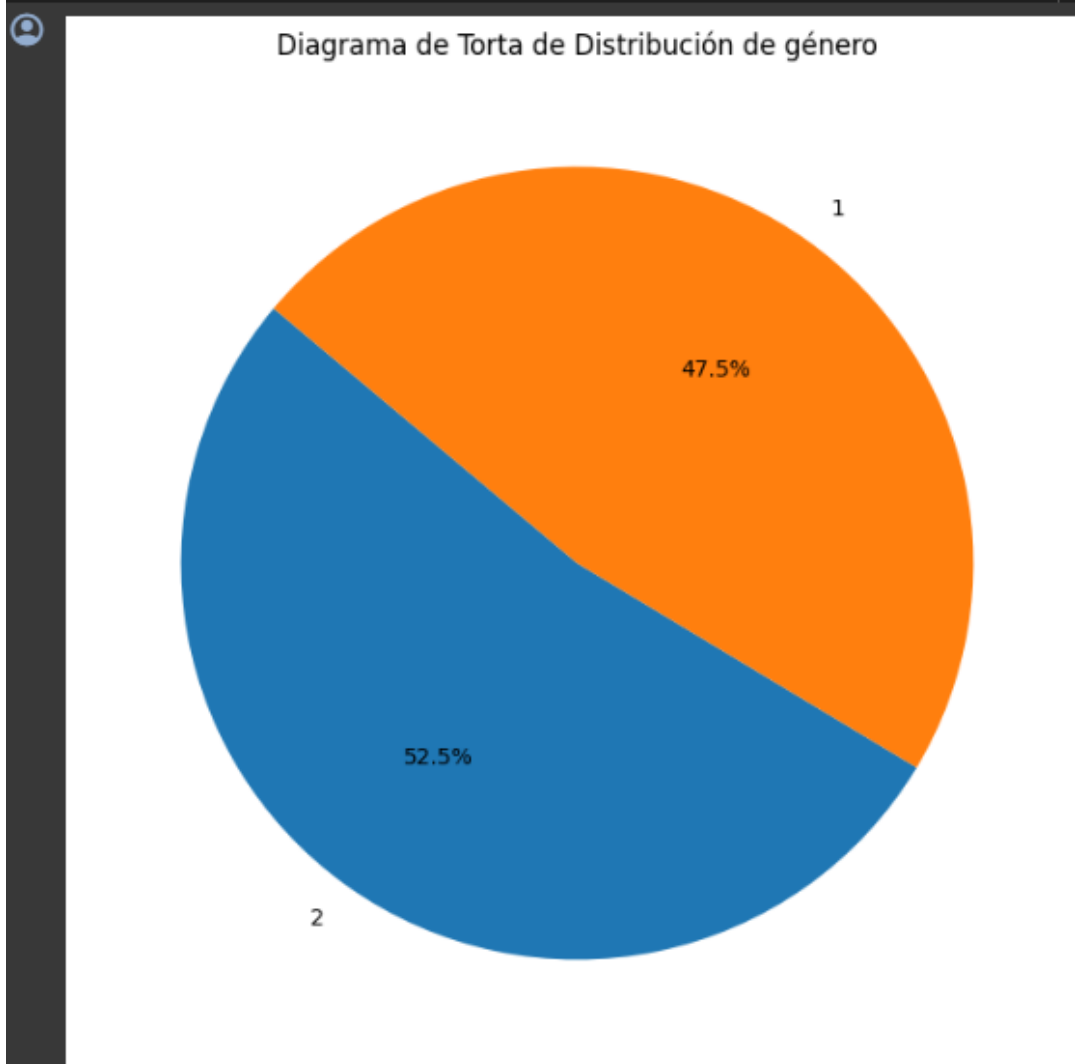
### Modelo de toma de decisiones

Para poder realizar una toma de decisiones utilizando los datos proporcionados ya sean en kaggle u otra pagina lo que tenemos que hacer es analizar las variables como por ejemplo en este caso que se realizó un análisis para determinar qué porcentaje de clientes son más propensos a adquirir un auto se tuvo en cuenta la edad los ingresos anuales y el genero estos fueron los resultados:



```
#Cargando librerías
import seaborn as sns
import matplotlib.pyplot as plt

# Crear un gráfico de torta
frecuencias = Conjunto_Datos['Gender'].value_counts()
plt.figure(figsize=(8, 8)) # Tamaño del gráfico (opcional)
plt.pie(frecuencias, labels=frecuencias.index, autopct='%1.1f%%', startangle=140)
plt.title('Diagrama de Torta de Distribución de género')
plt.show()
```



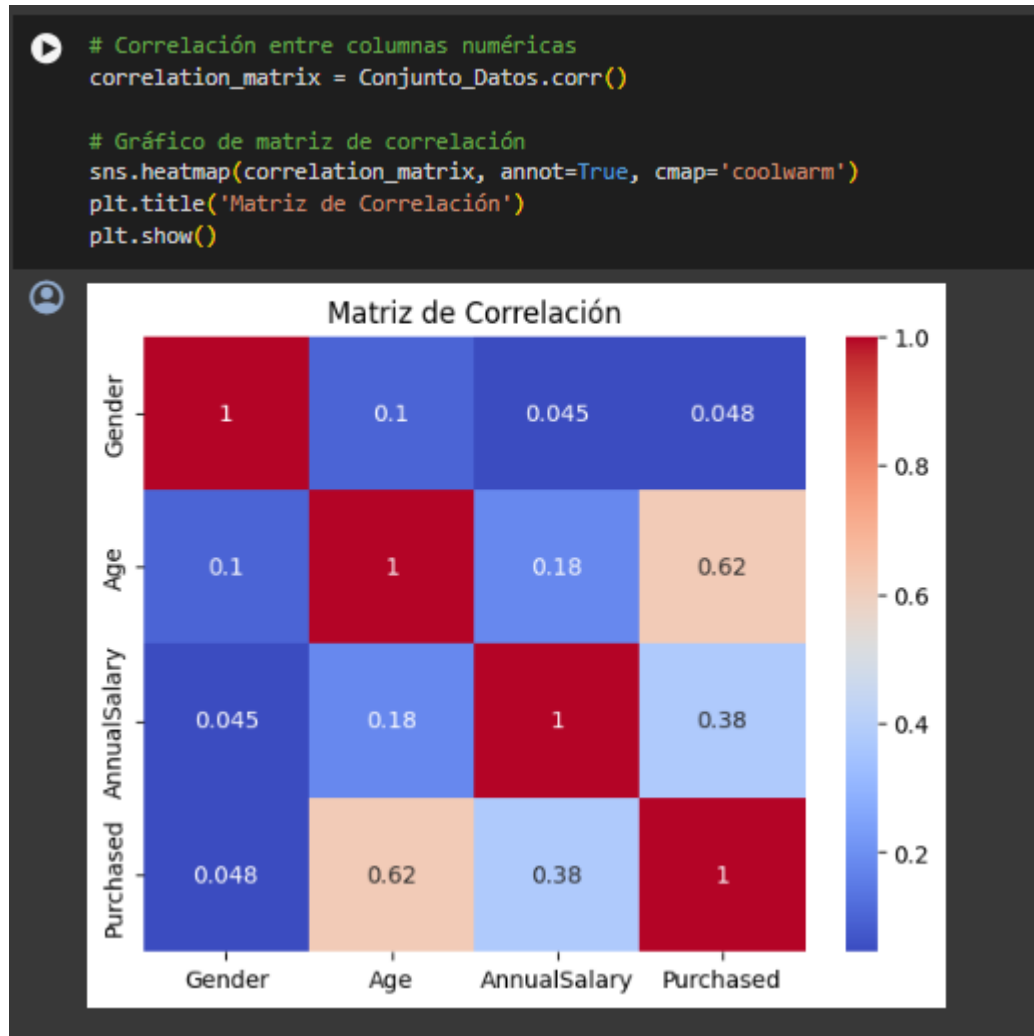
En el diagrama de torta se evidencia como 1 a los hombres y 2 a las mujeres se realizo este cambio para poder determinar mejores resultados a la hora de sacar un análisis

```
[ ] #Analizando variables categóricas
Conjunto_Datos['Gender'].unique()
```

```
array(['MALE', 'FEMALE'], dtype=object)
```

```
▶ # Cambiando rango de valores del dataframe
Estrategia = {'MALE':1,'FEMALE':2}
Conjunto_Datos['Gender']=Conjunto_Datos['Gender'].map(Estrategia)
Conjunto_Datos.head(20)
```

	Gender	Age	AnnualSalary	Purchased
0	1	35	20000	0
1	1	40	43500	0
2	1	49	74000	0
3	1	40	107500	1
4	1	25	79000	0
5	2	47	33500	1
6	2	46	132500	1
7	1	42	64000	0
8	2	30	84500	0
9	1	41	52000	0
10	1	42	80000	0
11	1	47	23000	1
12	2	32	72500	0
13	2	27	57000	0
14	2	42	108000	1
15	2	33	149000	1
16	1	35	75000	0
17	1	35	53000	0
18	1	46	79000	1
19	2	39	134000	1



La matriz de correlación nos muestra como cuales son las que más relación tienen entre ellas en donde 0 significa que casi que esa variable casi no se asocia y 1 máxima asociación en donde el genero y la edad no tiene relación ya que es de 0.1 lo mismo con el salario anual pero los clientes con mas edad son mas asociados a realizar compras.

### **Conclusiones**

Para analizar y visualizar datos relacionados con la incidencia de COVID-19 en diferentes países de todo el mundo, utilizando conjuntos de datos disponibles en Kaggle. La elaboración de este trabajo pude determinar que la república de corea, china, Italia, irán fueron los mas afectados al inicio de la pandemia pues no estaban preparados para lo que se venia nadie lo estaba solo que otros actuaron más rápido no porque estos países tengan un 25% quiere decir que los otros estén mejor ya que muchos fueron los afectados por que los otros no se quedan atrás con un 12%.

## Referencias

Zeng, F., Huang, Y., Guo, Y., Yin, M., Chen, X., Xiao, L. y Deng, G. (2020). Asociación de marcadores inflamatorios con la gravedad de COVID-19: un metanálisis. *Revista Internacional de Enfermedades Infecciosas* , 96 , 467-474.

Remuzzi, A. y Remuzzi, G. (2020). COVID-19 e Italia: ¿qué sigue?. *La Lanceta* , 395 (10231), 1225-1228.

Abdi, M. (2020). Brote de enfermedad por coronavirus 2019 (COVID-19) en Irán: acciones y problemas. *Control de infecciones y epidemiología hospitalaria* , 41 (6), 754-755.

Liu, L., Wang, EZ y Lee, CC (2020). Impacto de la pandemia de COVID-19 en los mercados de petróleo crudo y bursátiles de EE. UU.: un análisis que varía en el tiempo. *Cartas de investigación energética* , 1 (1).

Legido-Quigley, H., Mateos-García, JT, Campos, VR, Gea-Sánchez, M., Muntaner, C. y McKee, M. (2020). La resiliencia del sistema sanitario español frente a la pandemia de COVID-19. *The Lancet salud pública* , 5 (5), e251-e252.

<https://www.kaggle.com/datasets/iamhungundji/covid19-symptoms-checker/data>