



# Segunda Entrega TFM – COVID-19

INESDI

**Programa:** Máster en Business Intelligence &  
Data Management (Online)



**Autores:** Grupo 6 COVID - Grupo I (Proyecto Inesdi)

**Tutor:** Pier Paolo Rossi

- Amaia Miranda Ulloa
- Fabián Ascheri Aguerre
- José Chavarría Montero
- Juan Carlos Valcuende Aláez
- Patricia Peña Torres

10 de julio de 2023

# Tabla de Contenidos

INTRODUCCIÓN .....	3
DESCRIPCIÓN DE FUENTES DE INFORMACIÓN .....	4
Datos de Recuento de casos COVID y Defunciones.....	4
Fuente 1: Casos diarios y muertes por fecha modificada a la OMS .....	4
Fuente 2: Ultimos recuentos notificados de casos y muertes .....	5
Datos de Vacunación .....	5
Fuente 3: Datos de Vacunación.....	6
Fuente 4: Tipos de Vacunas.....	7
Otras Fuentes .....	8
Fuente 5: DATA ON TESTING FOR COVID-19 BY WEEK AND COUNTRY.....	8
Fuente 6: Data on hospital and ICU admission rates and current occupancy for COVID-19 .....	9
Fuente 7: Datos sobre casos diarios registrados por estado en USA .....	11
Fuente 8: Países.....	11
LIMPIEZA, TRANSFORMACIÓN Y ENRIQUECIMIENTO DE DATOS .....	13
Fuente 1: OMS Daily cases and deaths by date reported to WHO .....	13
Fuente 2: OMS Latest reported counts of cases and deaths .....	21
Fuente 3: Datos de Vacunación.....	25
Fuente 4: Tipos de Vacunas .....	31
Fuente 5: Data on testing for COVID-19 by week and country .....	36
Fuente 6: Data on hospital and ICU admission rates and current occupancy for COVID-19 .....	40
Fuente 7: Datos sobre casos diarios registrados por estado en USA .....	45
BASE DE DATOS Y PROCEDIMIENTO DE CARGA.....	56
Script para creación de BBDD .....	56
Modelo de datos de la BBDD – TFM COVID-19 .....	59
Notebook Jupyter de Conexión y Carga de Ficheros a la BBDD – TFM COVID-19 .....	60
CONCLUSIONES .....	63
ANEXOS.....	64

## INTRODUCCIÓN

En el primer entregable de este proyecto de TFM se realizó una definición del proyecto, incluyendo una propuesta de valor luego de entender las necesidades de los potenciales usuarios. Estas necesidades se definieron mediante entrevistas, las cuales se resumieron en una sección que denominamos "insights".










Adicionalmente, en el primer entregable se realizó una exploración y un perfilado inicial de las distintas fuentes de información, lo que permitió definir claramente aquellas que se estarían utilizando para el modelo de datos definitivo.

Este segundo avance que presentamos contiene las siguientes etapas dentro de la construcción de un modelo que permitirá brindar información relevante al potencial usuario. Dicho esto, el entregable se compone de las siguientes tres secciones:

1. Descripción de Fuentes de Información: en esta sección se explica en detalle cada una de las fuentes de información utilizadas, desde el contenido mismo de cada fuente, hasta el tipo de dato que se obtiene de cada columna.
2. Limpieza, Transformación y Enriquecimiento de Datos: en este paso se explica en detalle todo el proceso de depuración de los datos, para lo que se utilizó principalmente la librería Pandas de Python.
3. Base de Datos y Procedimiento de Carga: este capítulo considera la estructura de base de datos de SQL que se propone, así como el script de creación de la misma y el procedimiento de carga de información.

Finalmente, todo nuestro proyecto TFM se encuentra almacenado en el siguiente repositorio público de Github, para fácil acceso de cualquier persona interesada:

<https://github.com/patriciaapenat/TFM.git>

main	2 branches	0 tags	Go to file	Add file	<> Code
<hr/>					
	JuankyHub Add files via upload	f651d32 12 hours ago	45 commits		
	.ipynb_checkpoints	Actualització de testingcovid19	2 weeks ago		
	Ficheros_Depurados	Delete aa	12 hours ago		
	Notebooks	Add files via upload	12 hours ago		
	DEFMODTestingCovid19.csv	Actualització de testingcovid19	2 weeks ago		
	DataSets_TFM_COVID_19.ipynb	limpieza de datos testingcovid	2 weeks ago		
	README.md	Initial commit	3 weeks ago		
	Testingcovid19.ipynb	Actualització de testingcovid19	2 weeks ago		
	datos bbdd.ipynb	tfm sesión estructura bbdd	3 weeks ago		

# DESCRIPCIÓN DE FUENTES DE INFORMACIÓN

## Datos de Recuento de casos COVID y Defunciones

Consta de 2 ficheros que se cargan semanalmente desde la OMS <https://covid19.who.int/data>

Los recuentos de casos nuevos y muertes se calculan restando los recuentos totales acumulativos anteriores del recuento actual. Estos recuentos se actualizan gradualmente a lo largo del día a medida que se dispone de más información. Los recuentos diarios de casos nuevos y muertes se completan a las 23:59CET/CEST de cada día. Debido a las diferencias en los métodos de notificación, las horas límite, la consolidación de datos retrospectivos y los retrasos en la notificación, es posible que la cantidad de casos nuevos no siempre refleje los totales diarios publicados por países, territorios o áreas individuales. Debido a la tendencia reciente de los países que realizan ejercicios de conciliación de datos que eliminan un gran número de casos o muertes de sus recuentos totales, dichos datos pueden reflejarse como números negativos en los recuentos de nuevos casos/nuevas muertes, según corresponda. Esto ayudará a los usuarios a identificar cuándo se producen dichos ajustes.

### Fuente 1: Casos diarios y muertes por fecha modificada a la OMS

<https://covid19.who.int/WHO-COVID-19-global-data.csv>

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Descripción</i>
PAIS	VARCHAR(50)	País, territori, àrea
PAIS_ISO3	VARCHAR(3)	Código de país ISO Alpha-3
PAIS_ISO2	VARCHAR(2)	Código de país ISO Alpha-2
FECHA_NOTIFICACION	DATE	Fecha de notificación a IOMS
OMS_REGION	VARCHAR(50)	Oficinas regionales de la OMS: Los Estados miembros de la OMS se agrupan en seis regiones de la OMS: Oficina Regional para África ( <b>AFRO</b> ), Oficina Regional para las Américas ( <b>AMRO</b> ), Oficina Regional para el Sur- este Asiático ( <b>SEARO</b> ), Oficina Regional para Europa ( <b>EURO</b> ), Oficina Regional para el Mediterráneo Oriental ( <b>EMRO</b> ) y Oficina Regional para el Pacífico Occidental ( <b>WPRO</b> ).
CASOS_NUEVOS	INTEGER	Nuevos casos confirmados. Se calcula restando el recuentoacumulado anterior del recuentoacumulado de casos actual.*
CASOS_ACUM	INTEGER	Casos confirmadosacumuladosnotificados en la OMS hastaahora.
MUERTES_NUEVAS	INTEGER	Nuevasmuertesconfirmadas. Se calcula restando las defuncionesacumuladasanteriores de las defuncionesacumuladas actuales.*
MUERTES_ACUM	INTEGER	Las muertesconfirmadasacumuladas se han notificado a la OMS hastaahora.

\*Los usuarios tienen que tener en cuenta que, además de capturar nuevos casos y muertes notificados un día cualquiera, las actualizaciones se hacen retrospectivamente para corregir los recuentos de los días anteriores según sea necesario en función de la información posterior recibida.

## Fuente 2: Ultimos recuentos notificados de casos y muertes

<https://covid19.who.int/WHO-COVID-19-global-table-data.csv>

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Descripción</i>
PAIS_ISO_3	VARCHAR(3)	Código de país ISO Alpha-3
OMS_REGION	VARCHAR(50)	Región de l'OMS
CASOS_ACUM_TOTAL	INTEGER	Casos confirmados acumulados notificados en la OMS hasta ahora.
CASOS_ACUM_TOTAL_POR_100000_HAB	DOUBLE	Casos confirmados acumulados notificados a la OMS hasta la fecha por cada 100.000 habitantes.
CASOS_NUEVOS_INFORMADOS_UL T_7_DIAS	INTEGER	Se han notificado nuevos casos confirmados en los últimos 7 días. Se calcula restando el recuento acumulado anterior (8 días antes) del recuento acumulado de casos actual.
CASOS_NUEVOS_INFORMADOS_UL T_7_DIAS_POR_100000_HAB	DOUBLE	Nuevos casos confirmados notificados en los últimos 7 días por cada 100.000 habitantes.
CASOS_NUEVOS_INFORMADOS_UL T_24H	INTEGER	Se han notificado nuevos casos confirmados en las últimas 24 horas. Se calcula restando el recuento acumulado anterior de casos del recuento acumulado actual.
MUERTES_ACUM_INFORMADAS_U LT_7_DIAS	INTEGER	Las muertes confirmadas acumuladas que se han notificado a la OMS hasta ahora.
MUERTES_ACUM_TOTAL_POR_100 000_HAB	DOUBLE	Muertes confirmadas acumuladas notificadas a la OMS hasta la fecha por cada 100.000 habitantes.
MUERTES_NUEVAS_INFORMADAS_ ULT_7_DIAS	DOUBLE	Se han notificado nuevas muertes confirmadas en los últimos 7 días. Se calcula restando el recuento acumulado anterior de defunciones (8 días antes) del recuento acumulado actual de defunciones.
MUERTES_NUEVAS_INFORMADAS_ ULT_7_DIAS_POR_100000_HAB	DOUBLE	Nuevas muertes confirmadas notificadas en los últimos 7 días por cada 100.000 habitantes.
MUERTES_NUEVAS_INFORMADAS_ ULT_24H	INTEGER	Se han notificado nuevas muertes confirmadas en las últimas 24 horas. Se calcula restando el recuento acumulado anterior de defunciones del recuento acumulado actual de defunciones.

## Datos de Vacunación

Consta de 2 ficheros que se cargan semanalmente desde la OMS <https://covid19.who.int/data>

Un fichero con las actualizaciones semanales sobre la introducción y administración de vacunas por países, territorios y áreas. Estos datos se recopilan de numerosas fuentes, incluidos informes directos de los Estados miembros, la revisión de la OMS de datos oficiales disponibles públicamente o datos recopilados y publicados por sitios de terceros como Our World in Data . Los datos publicados por sitios de terceros no han sido validados por la OMS, y la OMS no puede comentar sobre su precisión o integridad. Se esperan diferencias en los conteos en comparación con otras fuentes debido a los diferentes criterios de inclusión y tiempos de corte de datos.

Las dosis totales administradas, las personas vacunadas con al menos una dosis y las personas vacunadas por completo son totales acumulados desde el inicio de la vacunación en el país respectivo, hasta la última actualización de datos. Las dosis totales administradas se refieren a dosis únicas y pueden no ser iguales al número total de personas vacunadas, según el régimen de dosis específico (las personas reciben dosis múltiples). Las dosis totales administradas por 100 habitantes pueden exceder las 100, por ejemplo, cuando más de la mitad de la población recibe las dos dosis de vacuna requeridas en un régimen de dos dosis. Las tasas <0,001 por 100 habitantes pueden redondearse a 0. Cuando se utilizan múltiples vacunas en un país/territorio/área, la fecha de inicio que se muestra es equivalente a la fecha de inicio de la primera vacuna introducida. No se tienen en cuenta las suspensiones (temporales o no) del despliegue de la vacunación.

Un segundo fichero con la información de los tipos de vacunas utilizadas por los diferentes países. La mención de empresas específicas o de productos vacunales de ciertos fabricantes no implica que la OMS los apruebe o recomiende de preferencia a otros de naturaleza similar que no se mencionan. Salvo excepciones limitadas, los nombres de los productos patentados se distinguen por letras mayúsculas iniciales.

### Fuente 3: Datos de Vacunación

<https://covid19.who.int/who-data/vaccination-data.csv>

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Descripción</i>
PAIS	VARCHAR(50)	País, territorio, área
PAIS_ISO3	VARCHAR(3)	Código de país ISO Alpha-3
CONTINENTE	VARCHAR(50)	Descripción del Continente
ISO_CONTINENTE	VARCHAR(2)	Código de Continente ISO Alpha-2
OMS_REGION	VARCHAR(50)	Oficinas regionales de la OMS: Los Estados miembros de la OMS se agrupan en seis regiones de la OMS: Oficina Regional para África ( <b>AFRO</b> ), Oficina Regional para las Américas ( <b>AMRO</b> ), Oficina Regional para el Sur- este Asiático ( <b>SEARO</b> ), Oficina Regional para Europa ( <b>EURO</b> ), Oficina Regional para el Mediterráneo Oriental ( <b>EMRO</b> ) y Oficina Regional para el Pacífico Occidental ( <b>WPRO</b> ).
FUENTE_DATOS	VARCHAR(50)	Indica la fuente de los datos: - REPORTING: Datos reportados por los Estados miembros, o procedentes de informes oficiales - OWID: Datos procedentes de OurWorld in Data: <a href="https://ourworldindata.org/covid-vaccinations">https://ourworldindata.org/covid-vaccinations</a>
FECHA_ULT_ACTUALIZACION	DATE	Fecha de la última actualización
TOTAL_VACUNACIÓN_ACUM	DOUBLE	Total acumulado de dosis de vacunas administradas
NPER_VACUNADAS_1DOSIS	INTEGER	Número acumulado de personas vacunadas con al menos una dosis
TOTAL_VACUNACION_PER100	DOUBLE	Total acumulado de dosis de vacunas administradas por cada 100 habitantes
NPER_VACUNADAS_1DOSIS_PER100	DOUBLE	Personas acumuladas vacunadas con al menos una dosis por cada 100 habitantes

NPER_VACUNADAS_DOSIS_FULL	INTEGER	Número acumulado de personas completamente vacunadas
NPER_VACUNADAS_DOSIS_FULL_PER100	DOUBLE	Número acumulado de personas completamente vacunadas por cada 100 habitantes
FECHA_PRIMERA_VACUNA	DATE	Fecha de las primeras vacunaciones. Equivalente a la fecha de inicio/lanzamiento de la primera vacuna administrada en un país.
N_TIPOS_VACUNAS_USADAS	INTEGER	Número de tipos de vacunas utilizadas por país, territorio, área
NPER_CON_DOSIS_ADIDICIONAL	DOUBLE	Las personas recibieron dosis de refuerzo o adicional
NPER_CON_DOSIS_ADIDICIONAL_PER100	DOUBLE	Las personas recibieron dosis de refuerzo o adicional por cada 100 habitantes

#### Fuente 4: Tipos de Vacunas

<https://covid19.who.int/who-data/vaccination-metadata.csv>

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Descripción</i>
PAIS_ISO3	VARCHAR(3)	Código de país ISO Alpha-3
CONTINENTE	VARCHAR(50)	Descripción del Continente
ISO_CONTINENTE	VARCHAR(2)	Código de Continente ISO Alpha-2
NOMBRE_VACUNA	VARCHAR(100)	Nombre corto combinado de la vacuna: "Empresa - Nombre del producto"
NOMBRE_TIPO_VACUNA	VARCHAR(90)	Nombre o etiqueta del producto de la vacuna, o tipos de vacuna (si no tiene nombre).
NOMBRE_COMPAÑÍA	VARCHAR(90)	Autorización de comercialización del titular del producto vacunal.
FECHA_PRIMERA_VACUNA	DATE	Fecha de las primeras vacunaciones. Equivalente a la fecha de inicio/lanzamiento de la primera vacuna administrada en un país.
FECHA_INICIO_VACUNACION	DATE	Fecha de inicio/lanzamiento de la vacunación con tipos de vacuna (excluye las vacunas durante los ensayos clínicos).
FECHA_FIN_VACUNACION	DATE	Fecha de finalización del despliegue de la vacuna
FUENTE_DATOS	VARCHAR(50)	Indica la fuente de datos - REPORTING: Datos reportados por los Estados miembros, o procedentes de informes oficiales - OWID: Datos procedentes de OurWorld in Data: <a href="https://ourworldindata.org/covid-vaccinations">https://ourworldindata.org/covid-vaccinations</a>

## Otras Fuentes

### Fuente 5: DATA ON TESTING FOR COVID-19 BY WEEK AND COUNTRY

<https://opendata.ecdc.europa.eu/covid19/testing/>

Las cifras que se muestran para la tasa de pruebas semanales por cada 100 000 habitantes y la positividad de las pruebas semanales (%) se basan en varias fuentes de datos.

El número de casos semanales por utilizado para estimar la positividad de la prueba semanal por país o región subnacional se basa en los datos recopilados por ECDC Epidemic Intelligence. Las fuentes de información son Ministerios de Salud o Institutos Nacionales de Salud Pública (sitios web, cuentas oficiales de twitter o cuentas oficiales de Facebook), y los datos obtenidos se cotejan sistemáticamente con datos de OMS. Hay más información disponible en este enlace.

La fuente principal del total de pruebas por país o región subnacional por semana son los datos agregados presentados por los Estados miembros a TESSy. Sin embargo, cuando no estaba disponible, como solía ser el caso antes de la pandemia, el ECDC recopiló datos de fuentes públicas en línea. Estos datos se han recuperado automática o manualmente ("web-scraped") diariamente de fuentes públicas en línea nacionales/oficiales de países de la UE/EEE. Cabe señalar que existen varias limitaciones para este tipo de datos. Los datos raspados no están disponibles para todas las variables y/o países debido a la variabilidad del contenido en los sitios web nacionales.

Además, el proceso de recopilación de datos requiere una adaptación constante para evitar series temporales interrumpidas (es decir, debido a la modificación de las páginas del sitio web, tipos de datos).

La tasa de notificación de 14 días de nuevos casos de COVID-19 se basa en los datos recopilados por ECDC Epidemic Intelligence de varias fuentes y se ve afectada por la estrategia de prueba local, la capacidad del laboratorio y la eficacia de los sistemas de vigilancia. Por lo tanto, la comparación de la situación epidemiológica de la COVID-19 entre países no debe basarse únicamente en estas tasas. Sin embargo, a nivel de país individual o regional, este indicador puede ser útil para monitorear la situación nacional a lo largo del tiempo.

Las políticas de pruebas y el número de pruebas realizadas por cada 100 000 personas varían notablemente a lo largo del UE/EEE y presumiblemente aún más entre terceros países. Las pruebas más exhaustivas conducirán inevitablemente a la detección de más casos.

#### Interpretation of COVID-19 data

La tasa de notificación de 14 días de nuevos casos de COVID-19 debe usarse en combinación con otros factores, incluidas las políticas de prueba, la cantidad de pruebas realizadas, la positividad de la prueba, el exceso de mortalidad y las tasas de ingresos hospitalarios y en la Unidad de Cuidados Intensivos (UCI), al analizar la situación epidemiológica en un país. La mayoría de estos indicadores se presentan para los Estados miembros de la UE/EEE en el informe Panorama general del país.

Incluso cuando se utilizan varios indicadores en combinación, las comparaciones entre países deben hacerse con cautela y experiencia epidemiológica relevante.

Columna	Descripción
ISO 3	3-letter ISO country code



year_week	yyyy-Www
level	National (archived dataset with national subnational data to week 36, 2022 is available on ECDC's website)
new_cases	Number of new confirmed cases
tests_done	Number of tests done
population	Numeric
testing_rate	Testing rate per 100,000 population
positivity_rate	Weekly test positivity (%): 100 x Number of new confirmed cases/number of tests done per week
testing_data_source	- Country API - Country GitHub - Country website - Manual webscraping - Other - Survey - TESSy: data provided directly by Member States to ECDC via TESSy

Fuente 6: Data on hospital and ICU admission rates and current occupancy for COVID-19

<https://www.ecdc.europa.eu/en/publications-data/download-data-hospital-and-icu-admission-rates-and-current-occupancy-covid-19>

Datos sobre las tasas de admisión hospitalaria y de UCI y la ocupación actual por COVID-19.

<b>Nombre del Campo</b>	<b>Tipo de Datos</b>	<b>Descripción</b>
COUNTRY	String	Pais 3-letter ISO country code
INDICATOR	String	<p>▣ Ocupación hospitalaria diaria (número de pacientes con COVID-19 en el hospital en un día determinado)</p> <p>▣ Ocupación diaria de la UCI (número de Pacientes con COVID-19 en UCI en un dado día)</p> <p>▣ Nuevos ingresos hospitalarios semanales por 100k (tarifa semanal de nuevo admisiones de pacientes con COVID-19 por 100 000 habitantes)</p> <p>▣ Nuevos ingresos semanales en UCI de pacientes con COVID-19 por cada 100k Cuerda (tasa semanal de nuevas admisiones por 100 000 habitantes)</p>
DATE	YYYY -MM - DD	Fecha para los indicadores de ocupación diaria
YEAR_WEEK	YYYY -Www	Fecha
VALUE	Numeric	Número de pacientes o nuevos ingresos

		por 100 000 habitantes
SOURCE	String	Fuente categórica de los datos: <a href="#">TESSy</a> : datos proporcionados directamente por Los Estados miembros al ECDC a través de TESSy <a href="#">Country_API</a> <a href="#">Country_Github</a> <a href="#">Country_Website</a> <a href="#">External_Github</a> <a href="#">CCI</a> <a href="#">Vigilancia</a> <a href="#">Other_Websit</a>

### Descripción y descargo de responsabilidad:

Los archivos de datos descargables contienen información sobre hospitalización y Unidad de Cuidados Intensivos (UCI) tasas de admisión y ocupación actual por COVID-19 por fecha y país. Cada fila contiene el datos correspondientes para una fecha determinada (día o semana) y por país. El archivo se actualiza semanalmente. Tú puede utilizar los datos de acuerdo con la política de derechos de autor del ECDC.

### Fuente

Las cifras mostradas sobre las tasas de hospitalización y admisión en UCI y la ocupación actual son basado en varias fuentes de datos. La fuente principal son los datos basados en casos presentados por los Estados miembros. Sin embargo, cuando no está disponible, y especialmente para la ocupación actual, el ECDC recopila datos de Fuentes públicas en línea.

Los datos que se muestran se han recuperado automática o manualmente ("web-scraped") diariamente de Fuentes en línea públicas nacionales/oficiales de países de la UE/EEE. Cabe señalar que hay varias limitaciones a este tipo de datos. Los datos raspados no están disponibles para todas las variables y/o países debido a la variabilidad del contenido en los sitios web nacionales. Además, el proceso de recopilación de datos requiere una adaptación constante para evitar series temporales interrumpidas (es decir, debido a la modificación del sitio web) páginas, tipos de datos). Los criterios de admisión en hospitales y UCI, y las políticas para informar estos datos difiere entre países y a lo largo del tiempo, lo que puede resultar en estimaciones sesgadas derivadas de tales datos.

### Interpretación de los datos de COVID-19

La tasa de notificación de 14 días de los nuevos casos de COVID-19 se basa en los datos recopilados por el ECDC Epidemic Intelligence de varias fuentes y se ven afectados por la estrategia de prueba local, la capacidad de los laboratorios y la eficacia de los sistemas de vigilancia. Comparando la epidemiología

Por lo tanto, la situación con respecto a COVID-19 entre países no debe basarse en estas tasas solo. Sin embargo, a nivel de país individual, este indicador puede ser útil para monitorear la situación nacional a lo largo del tiempo.

Las políticas de pruebas y el número de pruebas realizadas por cada 100 000 personas varían notablemente a lo largo del UE/EEE y presumiblemente aún más entre terceros países. Pruebas más extensas inevitablemente llevar a que se detecten más casos.

La tasa de notificación de 14 días de nuevos casos de COVID-19 debe usarse en combinación con otros factores que incluyen políticas de prueba, número de pruebas realizadas, positividad de la prueba, exceso de mortalidad y tasas de ingresos hospitalarios y en UCI, al analizar la situación epidemiológica de un país.

La mayoría de estos indicadores se presentan para los Estados miembros de la UE/EEE en el informe Panorama general del país. Incluso cuando se usan varios indicadores en combinación, se deben hacer comparaciones entre países con precaución y experiencia epidemiológica relevante.

Fuente: [2021-01-13 Variable Dictionary and Disclaimer hosp icu all data.pdf \(europa.eu\)](#)

Fuente 7: Datos sobre casos diarios registrados por estado en USA

[https://github.com/CSSEGISandData/COVID-19/tree/master/csse\\_covid\\_19\\_data/csse\\_covid\\_19\\_daily\\_reports\\_us](https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_daily_reports_us)

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Descripción</i>
PROVINCE_STATE	String	Nombre del estado en USA
CONFIRMED	Integer	Casos confirmados acumulados por estado
DEATH	Integer	Numero de individuos muertos acumulados por estado
RECOVERED	Float	Número de individuos recuperados acumulados por estado
ACTIVE	Float	Acumulado de casos confirmados que no an sido resueltos (Casos activos = número de casos totales-total de individuos recuperados -total de individuos muertos)
ISO3	String	Código oficial de identificador asignado a cada país
DATE	DATETIME	Fecha en que se realizo el nuevo registro

#### Descripción:

La fuente de información fue los archivos descargables que contiene datos de los casos acumulados registrados diariamente relativo a COVID-19 en cada estado referente a casos confirmados, individuos fallecidos, recuperados y casos activos

#### Fuente

La fuente de datos fue el repositorio de datos sobre COVID-19 del Centro para Sistemas de Ciencia e Ingeniería de la Universidad John Hopkings en GitHub. Los datos en este repositorio fueron recolectados por el Centro de Recursos de Coronavirus de dicha Universidad

Fuente 8: Países

<https://gist.github.com/wipodev/9596693c07e1152dae03f2e3e294c493>

<i>Nombre del Campo</i>	<i>Tipo de Datos</i>	<i>Descripción</i>
PAIS_ISO3	VARCHAR(3)	Código de país ISO Alpha-3
PAIS_ISO2	VARCHAR(2)	Código de país ISO Alpha-2
PAIS_NOM	VARCHAR(100)	Decriptivo del Nombre del País

COD_CONTINENTE	VARCHAR(2)	Código de Continente
CONTINENTE	VARCHAR(100)	Descriptivo del Nombre del Continente
OMS_REGION	VARCHAR(5)	Oficinas regionales de la OMS: Los Estados miembros de la OMS se agrupan en seis regiones de la OMS: Oficina Regional para África ( <b>AFRO</b> ), Oficina Regional para las Américas ( <b>AMRO</b> ), Oficina Regional para el Sur- este Asiático ( <b>SEARO</b> ), Oficina Regional para Europa ( <b>EURO</b> ), Oficina Regional para el Mediterráneo Oriental ( <b>EMRO</b> ) y Oficina Regional para el Pacífico Occidental ( <b>WPRO</b> ).
DESC_OMS_REGION	VARCHAR(100)	Descriptivo de las Oficinas regionales de la OMS
PAIS_NOM_2	VARCHAR(100)	Decriptivo del Nombre del País con alguna variedad en la descrpción del Nombre del País.

Relación de Países del Mundo identificados por los códigos ISO2 e ISO3 definidos por la Organización Internacional de Normalización (ISO), así como el continente al que pertenecen y la Oficina Regional de la OMS a la que son miembros.

# LIMPIEZA, TRANSFORMACIÓN Y ENRIQUECIMIENTO DE DATOS

## Fuente 1: OMS Daily cases and deaths by date reported to WHO

El fichero se puede encontrar en el siguiente enlace:

[https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data Frame covid daily oms.ipynb](https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data%20Frame%20covid%20daily%20oms.ipynb)

### Ficheros de la organización Mundial de la Salud

- <https://covid19.who.int/data>
- <https://ourworldindata.org/coronavirus#coronavirus-country-profiles>

### Descarga de datos

Casos diarios y muertes por fecha notificados a la OMS: <https://covid19.who.int/WHO-COVID-19-global-data.csv>

### Información del Dataset

Los usuarios deben tener en cuenta que, además de capturar nuevos casos y muertes notificadas en un día determinado, las actualizaciones se realizan retrospectivamente para corregir los recuentos de días anteriores según sea necesario en función de la información recibida posteriormente.

Consulte "Datos agregados diarios de recuento de casos y muertes" más arriba para obtener más detalles sobre el cálculo de nuevos casos/muertes.

### Exploración

Exloración de los casos diarios y muertes por fecha notificados a la OMS.

Al dataframe lo llamaremos "df\_covid\_daily".

```
In [1]: import pandas as pd
        pip install pycountry
        import pycountry

Requirement already satisfied: pycountry in c:\users\joschava\anaconda3\lib\site-packages (22.3.5)
Requirement already satisfied: setuptools in c:\users\joschava\anaconda3\lib\site-packages (from pycountry) (63.4.1)
```

```
In [2]: df_covid_daily = pd.read_csv('https://covid19.who.int/WHO-COVID-19-global-data.csv')
        df_covid_daily.head()
```

```
Out[2]:
```

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
0	2020-01-03	AF	Afghanistan	EMRO	0	0	0	0
1	2020-01-04	AF	Afghanistan	EMRO	0	0	0	0
2	2020-01-05	AF	Afghanistan	EMRO	0	0	0	0
3	2020-01-06	AF	Afghanistan	EMRO	0	0	0	0
4	2020-01-07	AF	Afghanistan	EMRO	0	0	0	0

```
In [3]: df_covid_daily.shape
```

```
Out[3]: (301701, 8)
```

```
In [4]: df_covid_daily.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301701 entries, 0 to 301700
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Date_reported          301701 non-null object
1   Country_code           300428 non-null object
2   Country                301701 non-null object
3   WHO_region             301701 non-null object
4   New_cases              301701 non-null int64
5   Cumulative_cases       301701 non-null int64
6   New_deaths             301701 non-null int64
7   Cumulative_deaths      301701 non-null int64
dtypes: int64(4), object(4)
memory usage: 18.4+ MB
```

## Transformación

Transformamos el campo de fecha (DATE\_REPORTED) que aparecen con tipo de datos cadena de caracteres

```
In [5]: df_covid_daily['Date_reported'] = pd.to_datetime(df_covid_daily['Date_reported'])
```

## Transformación

Verificamos que se ha modificado el tipo de dato a fecha

```
In [6]: df_covid_daily.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 301701 entries, 0 to 301700
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Date_reported          301701 non-null datetime64[ns]
1   Country_code           300428 non-null object
2   Country                301701 non-null object
3   WHO_region             301701 non-null object
4   New_cases              301701 non-null int64
5   Cumulative_cases       301701 non-null int64
6   New_deaths             301701 non-null int64
7   Cumulative_deaths      301701 non-null int64
dtypes: datetime64[ns](1), int64(4), object(3)
memory usage: 18.4+ MB
```

```
In [7]: df_covid_daily.head()
```

```
Out[7]:
```

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
0	2020-01-03	AF	Afghanistan	EMRO	0	0	0	0
1	2020-01-04	AF	Afghanistan	EMRO	0	0	0	0
2	2020-01-05	AF	Afghanistan	EMRO	0	0	0	0
3	2020-01-06	AF	Afghanistan	EMRO	0	0	0	0
4	2020-01-07	AF	Afghanistan	EMRO	0	0	0	0

## Exploración

Observamos si tenemos valores duplicados

```
In [8]: df_covid_daily.duplicated().values.any()
Out[8]: False
```

## Exploración

Observamos si tenemos valores nulos

```
In [9]: df_covid_daily.isnull().values.sum()
Out[9]: 1273
```

## Exploración

Identificamos dónde están los valores nulos

```
In [10]: df_covid_daily.isnull().sum()
Out[10]: Date_reported      0
Country_code      1273
Country            0
WHO_region        0
New_cases         0
Cumulative_cases  0
New_deaths        0
Cumulative_deaths  0
dtype: int64
```

```
In [11]: df_covid_daily_NaN = df_covid_daily[df_covid_daily['Country_code'].isnull()]
df_covid_daily_NaN
```

```
Out[11]:
```

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
182039	2020-01-03	NaN	Namibia	AFRO	0	0	0	0
182040	2020-01-04	NaN	Namibia	AFRO	0	0	0	0
182041	2020-01-05	NaN	Namibia	AFRO	0	0	0	0
182042	2020-01-06	NaN	Namibia	AFRO	0	0	0	0
182043	2020-01-07	NaN	Namibia	AFRO	0	0	0	0
...	...	...	...	...	...	...	...	...
183307	2023-06-24	NaN	Namibia	AFRO	0	171310	0	4091
183308	2023-06-25	NaN	Namibia	AFRO	0	171310	0	4091
183309	2023-06-26	NaN	Namibia	AFRO	0	171310	0	4091
183310	2023-06-27	NaN	Namibia	AFRO	0	171310	0	4091
183311	2023-06-28	NaN	Namibia	AFRO	0	171310	0	4091

## Transformación

Sustituimos los códigos de país nulos por su respectivo código ISO.

```
In [13]: df_country = pd.read_csv("C:/Users/joschava/Dropbox/TFM COVID/Limpieza de Archivos Fuente/Paises_Region_OMS.csv")
df_country
```

```
Out[13]:
```

	PAIS_ISO3	PAIS_ISO2	PAIS_NOM	COD_CONTINENTE	CONTINENTE	OMS_REGION	DESC_OMS_REGION
0	AGO	AO	Angola	AF	Africa	AFRO	Africa
1	BDI	BI	Burundi	AF	Africa	AFRO	Africa
2	BEN	BJ	Benin	AF	Africa	AFRO	Africa
3	BFA	BF	Burkina Faso	AF	Africa	AFRO	Africa
4	BWA	BW	Botswana	AF	Africa	AFRO	Africa
...	...	...	...	...	...	...	...
244	VCT	VC	Saint Vincent and the Grenadines	NA	North America	AMRO	América
245	VGB	VG	British Virgin Islands	NA	North America	AMRO	América
246	BLM	BL	San Bartolomé	NA	North America	AMRO	América
247	MAF	MF	San Martín	NA	North America	AMRO	América
248	SXM	SX	San Martín	NA	North America	AMRO	América

```
In [14]: df_covid_daily_NaN_merge = pd.merge(left=df_covid_daily_NaN, right=df_country, how='left', left_on='Country', right_on='PAIS_NOM')
df_covid_daily_NaN_merge
```

```
Out[14]:
```

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths	PAIS_ISO3	PAIS_ISO2	PAIS_NOM
0	2020-01-03	NaN	Namibia	AFRO	0	0	0	0	NAM	NM	Namibia
1	2020-01-04	NaN	Namibia	AFRO	0	0	0	0	NAM	NM	Namibia
2	2020-01-05	NaN	Namibia	AFRO	0	0	0	0	NAM	NM	Namibia
3	2020-01-06	NaN	Namibia	AFRO	0	0	0	0	NAM	NM	Namibia
4	2020-01-07	NaN	Namibia	AFRO	0	0	0	0	NAM	NM	Namibia
...	...	...	...	...	...	...	...	...	...	...	...
1268	2023-06-24	NaN	Namibia	AFRO	0	171310	0	4091	NAM	NM	Namibia
1269	2023-06-25	NaN	Namibia	AFRO	0	171310	0	4091	NAM	NM	Namibia
1270	2023-06-26	NaN	Namibia	AFRO	0	171310	0	4091	NAM	NM	Namibia
1271	2023-06-27	NaN	Namibia	AFRO	0	171310	0	4091	NAM	NM	Namibia
1272	2023-06-28	NaN	Namibia	AFRO	0	171310	0	4091	NAM	NM	Namibia

1273 rows x 15 columns

```
In [15]: df_covid_daily_NaN = df_covid_daily_NaN_merge[['Date_reported', 'PAIS_ISO2', 'Country', 'WHO_region', 'New_cases', 'Cumulative_cases', 'New_deaths', 'Cumulative_deaths']]
df_covid_daily_NaN
```

```
Out[15]:
```

	Date_reported	PAIS_ISO2	Country	WHO_region	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
0	2020-01-03	NM	Namibia	AFRO	0	0	0	0
1	2020-01-04	NM	Namibia	AFRO	0	0	0	0
2	2020-01-05	NM	Namibia	AFRO	0	0	0	0
3	2020-01-06	NM	Namibia	AFRO	0	0	0	0
4	2020-01-07	NM	Namibia	AFRO	0	0	0	0
...	...	...	...	...	...	...	...	...
1268	2023-06-24	NM	Namibia	AFRO	0	171310	0	4091
1269	2023-06-25	NM	Namibia	AFRO	0	171310	0	4091
1270	2023-06-26	NM	Namibia	AFRO	0	171310	0	4091
1271	2023-06-27	NM	Namibia	AFRO	0	171310	0	4091
1272	2023-06-28	NM	Namibia	AFRO	0	171310	0	4091

```
In [18]: df_covid_daily_NaN.rename(columns = {'PAIS_ISO2': 'Country_code'}, inplace = True)
df_covid_daily_NaN
```



Out[18]:

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
0	2020-01-03	NM	Namibia	AFRO	0	0	0	0
1	2020-01-04	NM	Namibia	AFRO	0	0	0	0
2	2020-01-05	NM	Namibia	AFRO	0	0	0	0
3	2020-01-06	NM	Namibia	AFRO	0	0	0	0
4	2020-01-07	NM	Namibia	AFRO	0	0	0	0
...	...	...	...	...	...	...	...	...
1268	2023-06-24	NM	Namibia	AFRO	0	171310	0	4091
1269	2023-06-25	NM	Namibia	AFRO	0	171310	0	4091
1270	2023-06-26	NM	Namibia	AFRO	0	171310	0	4091
1271	2023-06-27	NM	Namibia	AFRO	0	171310	0	4091
1272	2023-06-28	NM	Namibia	AFRO	0	171310	0	4091

1273 rows × 8 columns

## Transformación

Validamos que no hay nulos en el df.

```
In [19]: df_covid_daily_NaN.isnull().sum()
```

```
Out[19]: Date_reported      0
Country_code      0
Country           0
WHO_region        0
New_cases         0
Cumulative_cases  0
New_deaths        0
Cumulative_deaths 0
dtype: int64
```

## Transformación

Eliminamos registros con nulos en el df original (df\_covid\_daily).

```
In [21]: df_covid_daily_dropna = df_covid_daily.dropna()
df_covid_daily_dropna
```

Out[21]:

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
0	2020-01-03	AF	Afghanistan	EMRO	0	0	0	0
1	2020-01-04	AF	Afghanistan	EMRO	0	0	0	0
2	2020-01-05	AF	Afghanistan	EMRO	0	0	0	0
3	2020-01-06	AF	Afghanistan	EMRO	0	0	0	0
4	2020-01-07	AF	Afghanistan	EMRO	0	0	0	0
...	...	...	...	...	...	...	...	...
301696	2023-06-24	ZW	Zimbabwe	AFRO	0	265413	0	5707
301697	2023-06-25	ZW	Zimbabwe	AFRO	0	265413	0	5707
301698	2023-06-26	ZW	Zimbabwe	AFRO	0	265413	0	5707
301699	2023-06-27	ZW	Zimbabwe	AFRO	0	265413	0	5707
301700	2023-06-28	ZW	Zimbabwe	AFRO	0	265413	0	5707

300428 rows × 8 columns

```
In [22]: df_covid_daily_transformed = pd.concat([df_covid_daily_dropna,df_covid_daily_NaN])
df_covid_daily_transformed
```

Out[22]:

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
0	2020-01-03	AF	Afghanistan	EMRO	0	0	0	0
1	2020-01-04	AF	Afghanistan	EMRO	0	0	0	0
2	2020-01-05	AF	Afghanistan	EMRO	0	0	0	0
3	2020-01-06	AF	Afghanistan	EMRO	0	0	0	0
4	2020-01-07	AF	Afghanistan	EMRO	0	0	0	0
...	...	...	...	...	...	...	...	...
1268	2023-06-24	NM	Namibia	AFRO	0	171310	0	4091
1269	2023-06-25	NM	Namibia	AFRO	0	171310	0	4091
1270	2023-06-26	NM	Namibia	AFRO	0	171310	0	4091
1271	2023-06-27	NM	Namibia	AFRO	0	171310	0	4091
1272	2023-06-28	NM	Namibia	AFRO	0	171310	0	4091

301701 rows × 8 columns

## Transformación

Validamos misma cantidad de registros que df inicial y comprobamos que no haya nulos.

```
In [23]: df_covid_daily_transformed.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 301701 entries, 0 to 1272
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   Date_reported    301701 non-null  datetime64[ns]
1   Country_code     301701 non-null  object
2   Country          301701 non-null  object
3   WHO_region       301701 non-null  object
4   New_cases        301701 non-null  int64
5   Cumulative_cases 301701 non-null  int64
6   New_deaths       301701 non-null  int64
7   Cumulative_deaths 301701 non-null  int64
dtypes: datetime64[ns](1), int64(4), object(3)
memory usage: 20.7+ MB
```

```
In [24]: df_covid_daily_transformed.isnull().sum()
```

```
Out[24]: Date_reported    0
Country_code            0
Country                 0
WHO_region              0
New_cases               0
Cumulative_cases        0
New_deaths              0
Cumulative_deaths       0
dtype: int64
```

## Transformación

Nuestra base de datos incluirá códigos ISO3 para los países, por lo que debemos llamar la tabla de dimension "País" y sustituir los códigos ISO2 por sus respectivos ISO3.

```
In [43]: df_covid_daily_merge = pd.merge(left=df_covid_daily_transformed, right=df_country, how='left', left_on='Country_code', right_on='Country_code')
df_covid_daily_merge
```

```
Out[43]:
```

	Date_reported	Country_code	Country	WHO_region	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths	PAIS_ISO3	PAIS_ISO2	PAIS_ISO3
0	2020-01-03	AF	Afghanistan	EMRO	0	0	0	0	AFG	AF	Afgh
1	2020-01-04	AF	Afghanistan	EMRO	0	0	0	0	AFG	AF	Afgh
2	2020-01-05	AF	Afghanistan	EMRO	0	0	0	0	AFG	AF	Afgh
3	2020-01-06	AF	Afghanistan	EMRO	0	0	0	0	AFG	AF	Afgh
4	2020-01-07	AF	Afghanistan	EMRO	0	0	0	0	AFG	AF	Afgh
...	...	...	...	...	...	...	...	...	...	...	...
301696	2023-06-24	NM	Namibia	AFRO	0	171310	0	4091	NAM	NM	NM
301697	2023-06-25	NM	Namibia	AFRO	0	171310	0	4091	NAM	NM	NM
301698	2023-06-26	NM	Namibia	AFRO	0	171310	0	4091	NAM	NM	NM
301699	2023-06-27	NM	Namibia	AFRO	0	171310	0	4091	NAM	NM	NM
301700	2023-06-28	NM	Namibia	AFRO	0	171310	0	4091	NAM	NM	NM

```
In [44]: df_covid_final = df_covid_daily_merge[['Date_reported', 'PAIS_ISO3', 'New_cases', 'Cumulative_cases', 'New_deaths', 'Cumulative_deaths']]
df_covid_final.rename(columns = {'PAIS_ISO3':'Country_code'}, inplace = True)
df_covid_final
```

```
Out[44]:
```

	Date_reported	Country_code	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
0	2020-01-03	AFG	0	0	0	0
1	2020-01-04	AFG	0	0	0	0
2	2020-01-05	AFG	0	0	0	0
3	2020-01-06	AFG	0	0	0	0
4	2020-01-07	AFG	0	0	0	0
...	...	...	...	...	...	...
301696	2023-06-24	NAM	0	171310	0	4091

## Transformación

Identificamos valores nulos relacionados a códigos ISO2 especiales que no aparecen en la tabla de parámetros de país.

[https://es.wikipedia.org/wiki/ISO\\_3166-1\\_alfa-2](https://es.wikipedia.org/wiki/ISO_3166-1_alfa-2)

- El código XA está siendo utilizado por Suiza, como código de país por las Islas Canarias, a pesar de que IC ya está reservado con dicho propósito.<sup>23</sup>
- El código XI está siendo utilizado por el Gobierno del Reino Unido, como el prefijo de código de país del número EORI de Irlanda del Norte.<sup>24</sup>
- El código XK está siendo utilizado por la Comisión Europea,<sup>25</sup> el FMI, la SWIFT,<sup>26</sup> el CLDR y otras organizaciones como código de país provisional para Kosovo.<sup>27</sup>
- El código XN está siendo utilizado por la Organización Mundial de la Propiedad Intelectual como indicador para el Instituto Nórdico de Patentes, una organización internacional a la que pertenecen Dinamarca, Islandia, Noruega y Suecia.<sup>28</sup>
- El código XU está siendo utilizado por la Organización Mundial de la Propiedad Intelectual como indicador para la Unión Internacional para la Protección de las Obtenciones Vegetales
- El código XV está siendo utilizado por la Organización Mundial de la Propiedad Intelectual como indicador para el Instituto Visegrad de Patentes
- El código XX está siendo utilizado por la Organización Mundial de la Propiedad Intelectual como un indicador para estados desconocidos, otras entidades u organizaciones

```
In [45]: df_covid_final.isnull().sum()
```

```
Out[45]: Date_reported      0
Country_code      6365
New_cases         0
Cumulative_cases   0
New_deaths         0
Cumulative_deaths  0
dtype: int64
```

```
In [46]: df_covid_final_NaN = df_covid_final[df_covid_final['Country_code'].isnull()]
df_covid_final_NaN
```

```
Out[46]:
```

	Date_reported	Country_code	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
31825	2020-01-03	NaN	0	0	0	0
31826	2020-01-04	NaN	0	0	0	0
31827	2020-01-05	NaN	0	0	0	0
31828	2020-01-06	NaN	0	0	0	0
31829	2020-01-07	NaN	0	0	0	0
...	...	...	...	...	...	...
246957	2023-06-24	NaN	0	1220	0	6
246958	2023-06-25	NaN	0	1220	0	6
246959	2023-06-26	NaN	0	1220	0	6
246960	2023-06-27	NaN	0	1220	0	6
246961	2023-06-28	NaN	0	1220	0	6

```
In [47]: df_covid_final['Country_code'].fillna('Z99', inplace=True)
df_covid_final
```

C:\Users\joschava\AppData\Local\Temp\ipykernel\_25704\3015219144.py:1: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)

```
df_covid_final['Country_code'].fillna('Z99', inplace=True)
```

```
Out[47]:
```

	Date_reported	Country_code	New_cases	Cumulative_cases	New_deaths	Cumulative_deaths
0	2020-01-03	AFG	0	0	0	0
1	2020-01-04	AFG	0	0	0	0
2	2020-01-05	AFG	0	0	0	0
3	2020-01-06	AFG	0	0	0	0
4	2020-01-07	AFG	0	0	0	0
...	...	...	...	...	...	...
301696	2023-06-24	NAM	0	171310	0	4091

```
In [48]: df_covid_final.isnull().sum()
```

```
Out[48]: Date_reported      0
Country_code      0
New_cases         0
Cumulative_cases   0
New_deaths         0
Cumulative_deaths  0
dtype: int64
```

```
In [49]: df_covid_final.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 301701 entries, 0 to 301700
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   Date_reported    301701 non-null  datetime64[ns]
1   Country_code     301701 non-null  object
2   New_cases        301701 non-null  int64
3   Cumulative_cases 301701 non-null  int64
4   New_deaths       301701 non-null  int64
5   Cumulative_deaths 301701 non-null  int64
dtypes: datetime64[ns](1), int64(4), object(1)
memory usage: 16.1+ MB
```

## Exportamos

Por último, procedemos con la descarga de los datos.

```
In [50]: df_covid_final.to_csv("C:/Users/joschava/Dropbox/TFM COVID/Limpieza de Archivos Fuente/Daily cases and deaths by date reported to
```

## Fuente 2: OMS Latest reported counts of cases and deaths

El fichero se puede encontrar en el siguiente enlace:

[https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data\\_Frame\\_Latest%20reported%20counts%20of%20cases%20and%20deaths.ipynb](https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data_Frame_Latest%20reported%20counts%20of%20cases%20and%20deaths.ipynb)

### Ficheros de la organización Mundial de la Salud

- <https://covid19.who.int/data>
- <https://ourworldindata.org/coronavirus#coronavirus-country-profiles>

### Descarga de datos

Último reporte de casos y muertes notificados a la OMS: <https://covid19.who.int/WHO-COVID-19-global-table-data.csv>

### Información del Dataset

Los usuarios deben tener en cuenta que, además de capturar nuevos casos y muertes notificadas en un día determinado, las actualizaciones se realizan retrospectivamente para corregir los recuentos de días anteriores según sea necesario en función de la información recibida posteriormente.

Consulte "Datos agregados diarios de recuento de casos y muertes" más arriba para obtener más detalles sobre el cálculo de nuevos casos/muertes.

### Exploración

Exploración del último reporte de casos y muertes notificados a la OMS. Al dataframe lo llamaremos "df\_latest\_covid".

```
In [3]: import pandas as pd
```

```
In [14]: df_latests_covid = pd.read_csv("C:/Users/joschava/Dropbox/TFM COVID/Limpieza de Archivos Fuente/Latest reported counts of cases & deaths per 100,000 population.csv")
df_latests_covid.head()
```

Out[14]:

	Name	WHO Region	Cases - cumulative total	Cases - cumulative total per 100000 population	Cases - newly reported in last 7 days	Cases - newly reported in last 7 days per 100000 population	Cases - newly reported in last 24 hours	Deaths - cumulative total	Deaths - cumulative total per 100000 population	Deaths - newly reported in last 7 days	Deaths - newly reported in last 7 days per 100000 population	Deaths - newly reported in last 24 hours
0	Global	NaN	761402282	9768.405152	534869	6.862098	2692	6887000	88.356717	4243	0.054436	25
1	United States of America	Americas	102697566	31026.206000	152968	46.214000	0	1117054	337.476000	2084	0.630000	0
2	China	Western Pacific	99238143	6744.989000	503	0.034000	0	120894	8.217000	70	0.005000	0
3	India	South-East Asia	44707525	3239.665000	9407	0.682000	0	530841	38.467000	28	0.002000	0
4	France	Europe	38677413	59467.703000	42503	65.350000	3	161857	248.860000	122	0.188000	0

```
df_latests_covid = pd.read_csv("https://covid19.who.int/WHO-COVID-19-global-table-data.csv", sep=',', header='infer', index_col=False)
df_latests_covid.head()
```

```
In [15]: df_latests_covid.shape
```

Out[15]: (238, 12)

```
In [16]: df_latests_covid.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 238 entries, 0 to 237
Data columns (total 12 columns):
#   Column                                     Non-Null Count  Dtype
---  -
0   Name                                     238 non-null    object
1   WHO Region                             237 non-null    object
2   Cases - cumulative total                 238 non-null    int64
3   Cases - cumulative total per 100000 population  237 non-null    float64
4   Cases - newly reported in last 7 days      238 non-null    int64
5   Cases - newly reported in last 7 days per 100000 population  237 non-null    float64
6   Cases - newly reported in last 24 hours      238 non-null    int64
7   Deaths - cumulative total                 238 non-null    int64
8   Deaths - cumulative total per 100000 population  237 non-null    float64
9   Deaths - newly reported in last 7 days      238 non-null    int64
10  Deaths - newly reported in last 7 days per 100000 population  237 non-null    float64
11  Deaths - newly reported in last 24 hours      238 non-null    int64
dtypes: float64(4), int64(6), object(2)
memory usage: 22.4+ KB
```

## Exploración

Observamos si tenemos valores duplicados

```
In [17]: df_latests_covid.duplicated().values.any()
```

Out[17]: False

## Exploración

Observamos si tenemos valores nulos

```
In [18]: df_latests_covid.isnull().values.sum()
```

Out[18]: 5

## Exploración

Identificamos dónde están los valores nulos

```
In [25]: df_latests_covid.isnull().sum()
```

```
Out[25]: Name 0
WHO Region 1
Cases - cumulative total 0
Cases - cumulative total per 100000 population 1
Cases - newly reported in last 7 days 0
Cases - newly reported in last 7 days per 100000 population 1
Cases - newly reported in last 24 hours 0
Deaths - cumulative total 0
Deaths - cumulative total per 100000 population 1
Deaths - newly reported in last 7 days 0
Deaths - newly reported in last 7 days per 100000 population 1
Deaths - newly reported in last 24 hours 0
dtype: int64
```

```
In [26]: df_covid_final_NaN = df_latests_covid[df_latests_covid.isnull().any(axis=1)]
df_covid_final_NaN
```

```
Out[26]:
```

	Name	WHO Region	Cases - cumulative total	Cases - cumulative total per 100000 population	Cases - newly reported in last 7 days	Cases - newly reported in last 7 days per 100000 population	Cases - newly reported in last 24 hours	Deaths - cumulative total	Deaths - cumulative total per 100000 population	Deaths - newly reported in last 7 days	Deaths - newly reported in last 7 days per 100000 population	Deaths - newly reported in last 24 hours
0	Global	NaN	761402282	9768.405152	534869	6.862098	2692	6887000	88.356717	4243	0.054436	25
230	Other	Other	764	NaN	0	NaN	0	13	NaN	0	NaN	0

## Transformación

Ambas líneas podemos eliminarlas de nuestro DF por ser "total" y "otros" (no podremos analizar o extraer nada de esta información).

```
In [27]: df_latest_covid_transformed = df_latests_covid.dropna()
df_latest_covid_transformed
```

```
Out[27]:
```

	Name	WHO Region	Cases - cumulative total	Cases - cumulative total per 100000 population	Cases - newly reported in last 7 days	Cases - newly reported in last 7 days per 100000 population	Cases - newly reported in last 24 hours	Deaths - cumulative total	Deaths - cumulative total per 100000 population	Deaths - newly reported in last 7 days	Deaths - newly reported in last 7 days per 100000 population	Deaths - newly reported in last 24 hours
1	United States of America	Americas	102697566	31026.206	152968	46.214	0	1117054	337.476	2084	0.630	0
2	China	Western Pacific	99238143	6744.989	503	0.034	0	120894	8.217	70	0.005	0
3	India	South-East Asia	44707525	3239.665	9407	0.682	0	530841	38.467	28	0.002	0
4	France	Europe	38677413	59467.703	42503	65.350	3	161857	248.860	122	0.188	0
5	Germany	Europe	38338298	46098.129	21329	25.646	0	170493	205.001	51	0.061	0
...	...	...	...	...	...	...	...	...	...	...	...	...
233	Holy See	Europe	26	3213.844	0	0.000	0	0	0.000	0	0.000	0
234	Tokelau	Western Pacific	5	370.370	0	0.000	0	0	0.000	0	0.000	0

## Transformación

Validamos

```
In [29]: df_latest_covid_transformed[df_latest_covid_transformed.isnull().any(axis=1)]
```

```
Out[29]:
```

	Name	WHO Region	Cases - cumulative total	Cases - cumulative total per 100000 population	Cases - newly reported in last 7 days	Cases - newly reported in last 7 days per 100000 population	Cases - newly reported in last 24 hours	Deaths - cumulative total	Deaths - cumulative total per 100000 population	Deaths - newly reported in last 7 days	Deaths - newly reported in last 7 days per 100000 population	Deaths - newly reported in last 24 hours
--	------	------------	--------------------------	--	---------------------------------------	---	---	---------------------------	---	--	--	--

## Transformación

Llamamos tabla de países para obtener ISO3

```
In [46]: df_country = pd.read_csv("C:/Users/joschava/Dropbox/TFM COVID/Limpieza de Archivos Fuente/paises_region_oms_v2.csv")
df_country
```

Out[46]:

	PAIS_ISO3	PAIS_ISO2	PAIS_NOM	COD_CONTINENTE	CONTINENTE	OMS_REGION	DESC_OMS_REGION	PAIS_NOM_2
0	AGO	AO	Angola	AF	Africa	AFRO	Africa	Angola
1	BDI	BI	Burundi	AF	Africa	AFRO	Africa	Burundi
2	BEN	BJ	Benin	AF	Africa	AFRO	Africa	Benin
3	BFA	BF	Burkina Faso	AF	Africa	AFRO	Africa	Burkina Faso
4	BWA	BW	Botswana	AF	Africa	AFRO	Africa	Botswana
...	...	...	...	...	...	...	...	...
244	VCT	VC	Saint Vincent and the Grenadines	NA	North America	AMRO	América	Saint Vincent and the Grenadines
245	VGB	VG	British Virgin Islands	NA	North America	AMRO	América	British Virgin Islands
246	BLM	BL	San Bartolomé	NA	North America	AMRO	América	Saint Barthélemy
247	MAF	MF	San Martín	NA	North America	AMRO	América	Saint Martin
248	SXM	SX	San Martín	NA	North America	AMRO	América	San Martín

## Transformación

Cruzamos las tablas y luego limpiamos los registros

```
In [49]: df_latest_covid_transformed_merge = pd.merge(left=df_latest_covid_transformed, right=df_country, how='left', left_on='Name', right_on='PAIS_NOM')
df_latest_covid_transformed_merge
```

Out[49]:

	Name	WHO Region	Cases - cumulative total	Cases - cumulative total per 100000 population	Cases - newly reported in last 7 days	Cases - newly reported in last 7 days per 100000 population	Cases - newly reported in last 24 hours	Deaths - cumulative total	Deaths - cumulative total per 100000 population	Deaths - newly reported in last 7 days	Deaths - newly reported in last 7 days per 100000 population	Deaths - newly reported in last 24 hours	PAIS_ISO3	PAIS_NOM
0	United States of America	Americas	102697566	31026.206	152968	46.214	0	1117054	337.476	2084	0.630	0	USA	
1	China	Western Pacific	99238143	6744.989	503	0.034	0	120894	8.217	70	0.005	0	CHN	
2	India	South-East Asia	44707525	3239.665	9407	0.682	0	530841	38.467	28	0.002	0	IND	
3	France	Europe	38677413	59467.703	42503	65.350	3	161857	248.860	122	0.188	0	FRA	
4	Germany	Europe	38338298	46098.129	21329	25.646	0	170493	205.001	51	0.061	0	DEU	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

```
In [51]: df_latest_covid_transformed_merge['PAIS_ISO3'].fillna('Z99', inplace=True)
df_latest_covid_transformed_merge['PAIS_ISO2'].fillna('Z9', inplace=True)
df_latest_covid_transformed_merge['PAIS_NOM'].fillna('Otros', inplace=True)
```

```
In [54]: df_latest_covid_final = df_latest_covid_transformed_merge[['PAIS_ISO3', 'WHO Region', 'Cases - cumulative total', 'Cases - cumulative total per 100000 population', 'Cases - newly reported in last 7 days', 'Cases - newly reported in last 7 days per 100000 population', 'Cases - newly reported in last 24 hours', 'Deaths - cumulative total', 'Deaths - cumulative total per 100000 population', 'Deaths - newly reported in last 7 days', 'Deaths - newly reported in last 7 days per 100000 population', 'Deaths - newly reported in last 24 hours']]
df_latest_covid_final
```

Out[54]:

	PAIS_ISO3	WHO Region	Cases - cumulative total	Cases - cumulative total per 100000 population	Cases - newly reported in last 7 days	Cases - newly reported in last 7 days per 100000 population	Cases - newly reported in last 24 hours	Deaths - cumulative total	Deaths - cumulative total per 100000 population	Deaths - newly reported in last 7 days	Deaths - newly reported in last 7 days per 100000 population	Deaths - newly reported in last 24 hours
0	USA	Americas	102697566	31026.206	152968	46.214	0	1117054	337.476	2084	0.630	0
1	CHN	Western Pacific	99238143	6744.989	503	0.034	0	120894	8.217	70	0.005	0
2	IND	South-East Asia	44707525	3239.665	9407	0.682	0	530841	38.467	28	0.002	0
3	FRA	Europe	38677413	59467.703	42503	65.350	3	161857	248.860	122	0.188	0
4	DEU	Europe	38338298	46098.129	21329	25.646	0	170493	205.001	51	0.061	0
...	...	...	...	...	...	...	...	...	...	...	...	...
231	VAT	Europe	26	3213.844	0	0.000	0	0	0.000	0	0.000	0
232	TKL	Western Pacific	5	370.370	0	0.000	0	0	0.000	0	0.000	0
233	PCN	Western Pacific	4	8000.000	0	0.000	0	0	0.000	0	0.000	0
234	PRK	South-East Asia	0	0.000	0	0.000	0	0	0.000	0	0.000	0
235	TKM	Europe	0	0.000	0	0.000	0	0	0.000	0	0.000	0

236 rows × 12 columns



## Exportamos


Por último, procedemos con la descarga de los datos.

```
In [55]: df_latest_covid_final.to_csv("C:/Users/joschava/Dropbox/TFM COVID/Limpieza de Archivos Fuente/Latest reported counts of cases and
```

## Fuente 3: Datos de Vacunación

El fichero se puede encontrar en el siguiente enlace:

[https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data\\_Frame\\_vacunaciones\\_oms.ipynb](https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data_Frame_vacunaciones_oms.ipynb)

 Open in Colab

Clonación del Repositorio GitHub de **TFM**

```
In [2]: ! git clone https://github.com/patriciaapenat/TFM.git
```

```
Cloning into 'TFM'...
remote: Enumerating objects: 161, done.
remote: Counting objects: 100% (57/57), done.
remote: Compressing objects: 100% (56/56), done.
remote: Total 161 (delta 24), reused 0 (delta 0), pack-reused 104
Receiving objects: 100% (161/161), 7.78 MiB | 10.05 MiB/s, done.
Resolving deltas: 100% (69/69), done.
```

Importo algunas librerías de pandas

```
In [30]: import pandas as pd
import numpy as np
```

Instalo nuevas versiones del Módulo pycountry

```
In [31]: #!pip install pycountry==20.7.3
#!pip install pycountry-convert==0.7.2
```

## Acceso al dataset de la OMS que a partir de ahora se llamará **df\_vacuation**

```
In [4]: df_vacuation = pd.read_csv('https://covid19.who.int/who-data/vaccination-data.csv')
df_vacuation.head()
```

```
Out[4]:
```

	COUNTRY	ISO3	WHO_REGION	DATA_SOURCE	DATE_UPDATED	TOTAL_VACCINATIONS	PERSONS_VACCINATED_1PLUS_DOSE	TOTAL_
0	Albania	ALB	EURO	REPORTING	2023-05-14	3080679.0	1348396	
1	Chile	CHL	AMRO	REPORTING	2023-06-02	66273384.0	18106832	
2	Congo	COG	AFRO	REPORTING	2022-07-31	833210.0	695760	
3	Côte d'Ivoire	CIV	AFRO	REPORTING	2023-02-19	25263932.0	13568372	
4	Denmark	DNK	EURO	REPORTING	2023-06-11	14986863.0	4767119	

```
In [5]: df_vacuation.shape
```

```
Out[5]: (229, 16)
```

In [6]:

```
df_vacunacion.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 229 entries, 0 to 228
Data columns (total 16 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   COUNTRY                               229 non-null    object
1   ISO3                                  229 non-null    object
2   WHO_REGION                           229 non-null    object
3   DATA_SOURCE                         229 non-null    object
4   DATE_UPDATED                         229 non-null    object
5   TOTAL_VACCINATIONS                   228 non-null    float64
6   PERSONS_VACCINATED_1PLUS_DOSE        229 non-null    int64
7   TOTAL_VACCINATIONS_PER100            227 non-null    float64
8   PERSONS_VACCINATED_1PLUS_DOSE_PER100 229 non-null    float64
9   PERSONS_LAST_DOSE                    229 non-null    int64
10  PERSONS_LAST_DOSE_PER100             229 non-null    float64
11  VACCINES_USED                         0 non-null      float64
12  FIRST_VACCINE_DATE                    207 non-null    object
13  NUMBER_VACCINES_TYPES_USED            225 non-null    float64
14  PERSONS_BOOSTER_ADD_DOSE              211 non-null    float64
15  PERSONS_BOOSTER_ADD_DOSE_PER100       211 non-null    float64
dtypes: float64(8), int64(2), object(6)
memory usage: 28.8+ KB
```

Transformamos los **campos de fecha** que aparecen con tipo de datos cadena de caracteres a formato fecha

- 1.- DATE\_UPDATED
- 2.- FIRST\_VACCINE\_DATE

Transformamos los **campos de fecha** que aparecen con tipo de datos cadena de caracteres a formato fecha

- 1.- DATE\_UPDATED
- 2.- FIRST\_VACCINE\_DATE

In [7]:

```
df_vacunacion['DATE_UPDATED'] = pd.to_datetime(df_vacunacion['DATE_UPDATED']).dt.strftime('%d-%m-%Y')
df_vacunacion['FIRST_VACCINE_DATE'] = pd.to_datetime(df_vacunacion['FIRST_VACCINE_DATE']).dt.strftime('%d-%m-%Y')
```

Elimino la columna **VACCINES\_USED** ya que no hay valores....!!!

```
In [11]: df_vacuation.drop(['VACCINES_USED'], axis=1, inplace= True)
```

Observamos que no hay **ningún valor duplicado**

```
In [12]: df_vacuation.duplicated().values.any()
```

Out[12]: False

Observamos que hay **65 valores nulos**

```
In [13]: df_vacuation.isnull().values.sum()
```

Out[13]: 65

Identificamos en que **Variable** y la **suma de valores nulos**

```
In [14]: df_vacuation.isnull().sum()
```

```
Out[14]: COUNTRY                0
ISO3                0
WHO_REGION          0
DATA_SOURCE         0
DATE_UPDATED        0
TOTAL_VACCINATIONS   1
PERSONS_VACCINATED_1PLUS_DOSE  0
TOTAL_VACCINATIONS_PER100    2
PERSONS_VACCINATED_1PLUS_DOSE_PER100  0
PERSONS_LAST_DOSE        0
PERSONS_LAST_DOSE_PER100    0
FIRST_VACCINE_DATE      22
NUMBER_VACCINES_TYPES_USED   4
PERSONS_BOOSTER_ADD_DOSE     18
PERSONS_BOOSTER_ADD_DOSE_PER100  18
dtype: int64
```

Identificamos **el/los registros** donde se encuentran los **valores nulos**

- 1.-TOTAL\_VACCINATIONS
- 2.-TOTAL\_VACCINATIONS\_PER100
- 3.-FIRST\_VACCINE\_DATE
- 4.-NUMBER\_VACCINES\_TYPES\_USED
- 5.-PERSONS\_BOOSTER\_ADD\_DOSE
- 6.-PERSONS\_BOOSTER\_ADD\_DOSE\_PER100

```
In [15]: df_vacuation[df_vacuation['TOTAL_VACCINATIONS'].isnull()]
```

```
Out[15]:
```

	COUNTRY	ISO3	WHO_REGION	DATA_SOURCE	DATE_UPDATED	TOTAL_VACCINATIONS	PERSONS_VACCINATED_1PLUS_DOSE	TOTAL
155	Eritrea	ERI	AFRO	REPORTING	03-07-2022	NaN		0

```
In [16]: df_vacuation[df_vacuation['TOTAL_VACCINATIONS_PER100'].isnull()]
```

```
Out[16]:
```

	COUNTRY	ISO3	WHO_REGION	DATA_SOURCE	DATE_UPDATED	TOTAL_VACCINATIONS	PERSONS_VACCINATED_1PLUS_DOSE	TOTAL
155	Eritrea	ERI	AFRO	REPORTING	03-07-2022	NaN		0
201	Turkey	TUR	EURO	REPORTING	29-01-2023	139694693.0	50974980	

```
In [17]: df_vacuation[df_vacuation['FIRST_VACCINE_DATE'].isnull()]
```

Sustituimos los valores **NaN** por valores **0**

```
In [19]: df_vacuation=df_vacuation.fillna(0)
df_vacuation.head()
```

```
Out[19]:
```

	COUNTRY	ISO3	WHO_REGION	DATA_SOURCE	DATE_UPDATED	TOTAL_VACCINATIONS	PERSONS_VACCINATED_1PLUS_DOSE	TOTAL_V
0	Albania	ALB	EURO	REPORTING	14-05-2023	3080679.0	1348396	
1	Chile	CHL	AMRO	REPORTING	02-06-2023	66273384.0	18106832	
2	Congo	COG	AFRO	REPORTING	31-07-2022	833210.0	695760	
3	Côte d'Ivoire	CIV	AFRO	REPORTING	19-02-2023	25263932.0	13568372	
4	Denmark	DNK	EURO	REPORTING	11-06-2023	14986863.0	4767119	

Verificamos la instrucción y observamos que **no existen valores nulos**

```
In [20]: df_vacuation.isnull().sum()
```

```
Out[20]: COUNTRY          0
ISO3          0
WHO_REGION     0
DATA_SOURCE     0
DATE_UPDATED    0
TOTAL_VACCINATIONS  0
PERSONS_VACCINATED_1PLUS_DOSE  0
TOTAL_VACCINATIONS_PER100  0
PERSONS_VACCINATED_1PLUS_DOSE_PER100  0
PERSONS_LAST_DOSE  0
PERSONS_LAST_DOSE_PER100  0
FIRST_VACCINE_DATE  0
NUMBER_VACCINES_TYPES_USED  0
PERSONS_BOOSTER_ADD_DOSE  0
-----
dtype: object
```

Mediante la función **.columns** obtenemos el orden y el nombre de las columnas del **df\_vacuation**

```
In [21]: df_vacuation.columns
```

```
Out[21]: Index(['COUNTRY', 'ISO3', 'WHO_REGION', 'DATA_SOURCE', 'DATE_UPDATED',
'TOTAL_VACCINATIONS', 'PERSONS_VACCINATED_1PLUS_DOSE',
'TOTAL_VACCINATIONS_PER100', 'PERSONS_VACCINATED_1PLUS_DOSE_PER100',
'PERSONS_LAST_DOSE', 'PERSONS_LAST_DOSE_PER100', 'FIRST_VACCINE_DATE',
'NUMBER_VACCINES_TYPES_USED', 'PERSONS_BOOSTER_ADD_DOSE',
'PERSONS_BOOSTER_ADD_DOSE_PER100'],
dtype='object')
```

```
In [22]: df_vacuation.shape
```

```
Out[22]: (229, 15)
```

Renombramos el nombre de las Columnas del **df\_vacuation**

```
In [23]: df_vacuation.columns =['PAIS', 'ISO3', 'OMS_REGION', 'FUENTE_DE_DATOS', 'FECHA_ACTUALIZADA',
'TOTAL_VACUNACIONES', 'PERSONAS_VACUNADAS_1_DOSIS',
'TOTAL_VACUNACIONES_PER100', 'PERSONAS_VACUNADAS_1MAS_DOSIS_PER100',
'PERSONAS_TOTALMENTE_VACUNADAS', 'PERSONAS_TOTALMENTE_VACUNADAS_PER_100', 'FECHA_PRIMERAS_VACUNAS', 'NUMERO_TIPOS_VACU',
'PERSONAS_REFUERZO_O_DOSIS', 'PERSONAS_REFUERZO_O_DOSIS_PER100']
```

Con el módulo **pycountry** obtenemos datos adicionales para el dataframe y se renombran las columnas con la función **.columns**

```
In [32]: import pycountry_convert as pc

def obtener_continente_ISO3(codigo_ISO3):
    try:
        continente_code = pc.country_alpha3_to_country_alpha2(codigo_ISO3)
        continente = pc.country_alpha2_to_continent_code(continente_code)
        continente_nombre = pc.convert_continent_code_to_continent_name(continente)
        return continente_nombre, continente
    except:
        return None, None

df_vacunacion[['Continente', 'ISO_continente']] = df_vacunacion['ISO3'].apply(obtener_continente_ISO3).apply(pd.Series)

indice_ISO3 = df_vacunacion.columns.get_loc("ISO3")

df_vacunacion.insert(indice_ISO3 + 1, "Continente", df_vacunacion.pop("Continente"))
df_vacunacion.insert(indice_ISO3 + 2, "ISO_continente", df_vacunacion.pop("ISO_continente"))
```

```
In [33]: df_vacunacion
```

```
Out[33]:
```

	PAIS	ISO3	Continente	ISO_continente	OMS_REGION	FUENTE_DE_DATOS	FECHA_ACTUALIZADA	TOTAL_VACUNACIONES	PE
0	Albania	ALB	Europe	EU	EURO	REPORTING	14-05-2023	3080679.0	
1	Chile	CHL	South America	SA	AMRO	REPORTING	02-06-2023	66273384.0	
2	Congo	COG	Africa	AF	AFRO	REPORTING	31-07-2022	833210.0	
3	Côte d'Ivoire	CIV	Africa	AF	AFRO	REPORTING	19-02-2023	25263932.0	

```
In [36]: df_vacunacion.isnull().sum()
```

```
Out[36]: PAIS          0
ISO3            0
Continente       7
ISO_continente   7
OMS_REGION      0
FUENTE_DE_DATOS  0
FECHA_ACTUALIZADA 0
TOTAL_VACUNACIONES 0
PERSONAS_VACUNADAS_1_DOSIS 0
TOTAL_VACUNACIONES_PER100 0
PERSONAS_VACUNADAS_1MAS_DOSIS_PER100 0
PERSONAS_TOTALMENTE_VACUNADAS 0
PERSONAS_TOTALMENTE_VACUNADAS_PER_100 0
FECHA_PRIMERAS_VACUNAS 0
NUMERO_TIPOS_VACUNAS_USADAS 0
PERSONAS_REFUERZO_O_DOSIS 0
PERSONAS_REFUERZO_O_DOSIS_PER100 0
dtype: int64
```

Observo que aparecen 7 registros con el Campo de **Continente y ISO\_continente** nulos

```
In [37]: df_vacunacion[df_vacunacion['Continente'].isnull()]
```

```
Out[37]:
```

	PAIS	ISO3	Continente	ISO_continente	OMS_REGION	FUENTE_DE_DATOS	FECHA_ACTUALIZADA	TOTAL_VACUNACIONES	PERS
44	Kosovo	XKX	None	None	EURO	REPORTING	15-01-2023	1836901.0	
69	Sint Maarten	SXM	None	None	AMRO	REPORTING	02-06-2023	66829.0	
143	Saba	XCA	None	None	AMRO	REPORTING	02-06-2023	4979.0	
149	Bonaire	XAA	None	None	AMRO	REPORTING	02-06-2023	43070.0	
217	Pitcairn	PCN	None	None	WPRO	REPORTING	29-08-2022	117.0	

Sustituyo los valores nulos por los valores **Sin Determinar** y **Z9**

```
In [38]: df_vacuation['Continente'].fillna('Sin Determinar', inplace=True)
df_vacuation['ISO_continente'].fillna('Z9', inplace=True)
```

Verifico los cambios

```
In [39]: df_vacuation[df_vacuation.ISO_continente == 'Z9']
```

```
Out[39]:
```

	PAIS	ISO3	Continente	ISO_continente	OMS_REGION	FUENTE_DE_DATOS	FECHA_ACTUALIZADA	TOTAL_VACUNACIONES	PERS
44	Kosovo	XKX	Sin Determinar	Z9	EURO	REPORTING	15-01-2023	1836901.0	
69	Sint Maarten	SXM	Sin Determinar	Z9	AMRO	REPORTING	02-06-2023	66829.0	
143	Saba	XCA	Sin Determinar	Z9	AMRO	REPORTING	02-06-2023	4979.0	
149	Bonaire	XAA	Sin Determinar	Z9	AMRO	REPORTING	02-06-2023	43070.0	
217	Pitcairn Islands	PCN	Sin Determinar	Z9	WPRO	REPORTING	29-08-2022	117.0	
---	Sint	---	Sin	---	---	REPORTING	02-06-2023	117.0	

```
In [ ]: df_vacuation.columns = ['PAIS', 'PAIS_ISO3', 'CONTINENTE', 'ISO_CONTINENTE', 'OMS_REGION', 'FUENTE_DATOS', 'FECHA_ULT_ACTUALIZA',
'TOTAL_VACUNACION_ACUM', 'NPER_VACUNADAS_1DOSIS',
'TOTAL_VACUNACION_PER100', 'NPER_VACUNADAS_1DOSIS_PER100',
'NPER_VACUNADAS_DOSIS_FULL', 'NPER_VACUNADAS_DOSIS_FULL_PER100', 'FECHA_PRIMERA_VACUNA', 'N_TIPOS_VACUNAS_USADAS',
'NPER_CON_DOSIS_ADICIONAL', 'NPER_CON_DOSIS_ADICIONAL_PER100']
```

Exporto el fichero depurado

```
In [41]: df_vacuation.to_csv('/content/TFM/Ficheros_Depurados/df_vacuation.csv', index=False)
```

## Fuente 4: Tipos de Vacunas

El fichero se puede encontrar en el siguiente enlace:

[https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data\\_Frame\\_vacunas\\_tipo oms.ipynb](https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data_Frame_vacunas_tipo oms.ipynb)

 Open in Colab

Clonación del Repositorio GitHub de TFM

```
In [4]: ! git clone https://github.com/patriciaapenat/TFM.git
```

fatal: destination path 'TFM' already exists and is not an empty directory.

```
In [5]: import pandas as pd
import numpy as np
```

Instalo nuevas versiones del Módulo pycountry

```
In [32]: #!pip install pycountry==20.7.3
#!pip install pycountry-convert==0.7.2
```

## Acceso al dataset de la OMS que a partir de ahora se llamará **df\_vacuation\_Meta**

```
In [7]: df_vacuation_Meta = pd.read_csv ('https://covid19.who.int/who-data/vaccination-metadata.csv')
df_vacuation_Meta.head()
```

```
Out[7]:
```

	ISO3	VACCINE_NAME	PRODUCT_NAME	COMPANY_NAME	AUTHORIZATION_DATE	START_DATE	END_DATE	COMMENT	DATA_SOURCE
0	SHN	AstraZeneca - AZD1222	AZD1222	AstraZeneca	NaN	NaN	NaN	NaN	OV
1	GRL	Moderna - mRNA-1273	mRNA-1273	Moderna	NaN	NaN	NaN	NaN	OV
2	FRO	Moderna - mRNA-1273	mRNA-1273	Moderna	NaN	NaN	NaN	NaN	OV

```
In [8]: df_vacuation_Meta.shape
```

```
Out[8]: (1100, 9)
```

```
In [9]: df_vacuation_Meta.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1100 entries, 0 to 1099
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   ISO3                   1100 non-null  object
1   VACCINE_NAME           1100 non-null  object
2   PRODUCT_NAME           1099 non-null  object
3   COMPANY_NAME           1053 non-null  object
4   AUTHORIZATION_DATE     490 non-null   object
5   START_DATE             814 non-null   object
6   END_DATE               0 non-null     float64
7   COMMENT               0 non-null     float64
8   DATA_SOURCE           1100 non-null  object
dtypes: float64(2), object(7)
memory usage: 77.5+ KB
```

Transformamos los **campos de fecha** que aparecen con tipo de datos cadena de caracteres a formato fecha

- 1.- AUTHORIZATION\_DATE
- 2.- START\_DATE
- 3.- END\_DATE

```
In [10]: df_vacuation_Meta['AUTHORIZATION_DATE'] = pd.to_datetime(df_vacuation_Meta['AUTHORIZATION_DATE']).dt.strftime('%d-%m-%Y')
df_vacuation_Meta['START_DATE'] = pd.to_datetime(df_vacuation_Meta['START_DATE']).dt.strftime('%d-%m-%Y')
df_vacuation_Meta['END_DATE'] = pd.to_datetime(df_vacuation_Meta['END_DATE']).dt.strftime('%d-%m-%Y')
```

Observamos que no hay **ningún valor duplicado**

```
In [14]: df_vacuation_Meta.duplicated().values.any()
```

Out[14]: False

Elimino las variables **END\_DATE** y **COMMENT** por estar vacías en su totalidad

```
In [15]: df_vacuation = df_vacuation_Meta.drop(['END_DATE', 'COMMENT'], axis=1, inplace=True)
```

Observamos que hay **944 valores nulos**

```
In [16]: df_vacuation_Meta.isnull().values.sum()
```

Out[16]: 944

Identificamos en que **Variable** y la **suma de valores nulos**

```
In [17]: df_vacuation_Meta.isnull().sum()
```

```
Out[17]: ISO3                0
VACCINE_NAME              0
PRODUCT_NAME              1
COMPANY_NAME              47
AUTHORIZATION_DATE       610
START_DATE                286
DATA_SOURCE               0
dtype: int64
```

Identificamos **el/los registros** donde se encuentran los **valores nulos**



Identificamos **el/los registros** donde se encuentran los **valores nulos**

- 1.-PRODUCT\_NAME
- 2.-COMPANY\_NAME
- 3.-AUTHORIZATION\_DATE
- 4.-START\_DATE

```
In [18]: df_vacuation_Meta[df_vacuation_Meta['PRODUCT_NAME'].isnull()]
```

```
Out[18]:
```

	ISO3	VACCINE_NAME	PRODUCT_NAME	COMPANY_NAME	AUTHORIZATION_DATE	START_DATE	DATA_SOURCE
17	TUR	Turkovac	NaN	NaN	NaN	NaN	OWID

```
In [19]: df_vacuation_Meta[df_vacuation_Meta['COMPANY_NAME'].isnull()]
```

```
Out[19]:
```

	ISO3	VACCINE_NAME	PRODUCT_NAME	COMPANY_NAME	AUTHORIZATION_DATE	START_DATE	DATA_SOURCE
17	TUR	Turkovac	NaN	NaN	NaN	NaN	OWID
1054	SYC	Julphar - Hayat-Vax	Hayat-Vax	NaN	NaN	NaN	REPORTING
1055	PRY	Julphar - Hayat-Vax	Hayat-Vax	NaN	30-12-2020	24-05-2021	REPORTING
1056	PHL	Julphar - Hayat-Vax	Hayat-Vax	NaN	11-08-2021	25-08-2021	REPORTING
1057	IRN	Shifa - COVIran Barakat	COVIran Barakat	NaN	NaN	NaN	REPORTING
1058	MCO	Novavax - Covavax	Covavax	NaN	NaN	30-03-2022	REPORTING
1059	THA	SII - Covovax	Covovax	NaN	27-04-2022	NaN	REPORTING
1060	IND	SII - Covovax	Covovax	NaN	NaN	NaN	REPORTING
		Moderna - Spikevax					

Sustituimos los valores NaN de las variables **PRODUCT\_NAME** y **COMPANY\_NAME** en "Sin Determinar"

```
In [21]: df_vacuation_Meta['PRODUCT_NAME'].fillna('Sin Determinar', inplace=True)
df_vacuation_Meta['COMPANY_NAME'].fillna('Sin Determinar', inplace=True)
```

Sustituimos los valores **NaN** por valores **0**

```
In [22]: df_vacuation_Meta=df_vacuation_Meta.fillna(0)
df_vacuation_Meta.head()
```

```
Out[22]:
```

	ISO3	VACCINE_NAME	PRODUCT_NAME	COMPANY_NAME	AUTHORIZATION_DATE	START_DATE	DATA_SOURCE
0	SHN	AstraZeneca - AZD1222	AZD1222	AstraZeneca	0	0	OWID
1	GRL	Moderna - mRNA-1273	mRNA-1273	Moderna	0	0	OWID
2	FRO	Moderna - mRNA-1273	mRNA-1273	Moderna	0	0	OWID
3	FRO	Pfizer BioNTech - Comirnaty	Comirnaty	Pfizer BioNTech	0	0	OWID
4	BIH	AstraZeneca - AZD1222	AZD1222	AstraZeneca	0	0	OWID

Verificamos la instrucción y observamos que **no existen valores nulos**

```
In [23]: df_vacuation_Meta.isnull().sum()
```

```
Out[23]:
```

ISO3	0
VACCINE_NAME	0
PRODUCT_NAME	0
COMPANY_NAME	0
AUTHORIZATION_DATE	0
START_DATE	0
DATA_SOURCE	0
dtype:	int64

Mediante la función `.columns` obtenemos el orden y el nombre de las columnas del `df_vacunacion_Meta`

```
In [24]: df_vacunacion_Meta.columns
```

```
Out[24]: Index(['ISO3', 'VACCINE_NAME', 'PRODUCT_NAME', 'COMPANY_NAME',  
              'AUTHORIZATION_DATE', 'START_DATE', 'DATA_SOURCE'],  
              dtype='object')
```

```
In [25]: df_vacunacion_Meta.shape
```

```
Out[25]: (1100, 7)
```

Renombramos el nombre de las Columnas del `df_vacunacion_Meta`

```
In [26]: df_vacunacion_Meta.columns=['PAIS_ISO3', 'NOMBRE_VACUNA', 'NOMBRE_TIPO_VACUNA', 'NOMBRE_COMPañIA',  
                                     'FECHA_AUTORIZACION', 'FECHA_INICIO_VACUNACION', 'FUENTE_DATOS']
```

```
In [27]: df_vacunacion_Meta.columns  
df_vacunacion_Meta.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1100 entries, 0 to 1099  
Data columns (total 7 columns):  
#   Column                Non-Null Count  Dtype  
---  ----  
0   PAIS_ISO3              1100 non-null  object  
1   NOMBRE_VACUNA          1100 non-null  object  
2   NOMBRE_TIPO_VACUNA     1100 non-null  object  
3   NOMBRE_COMPañIA        1100 non-null  object  
4   FECHA_AUTORIZACION     1100 non-null  object  
5   FECHA_INICIO_VACUNACION 1100 non-null  object  
6   FUENTE_DATOS           1100 non-null  object  
dtypes: object(7)  
memory usage: 60.3+ KB
```

```
In [28]: df_vacunacion_Meta['PAIS_ISO3'].unique()
```

Hacemos una última revisión

```
In [33]: df_vacunacion_Meta.duplicated().values.any()
```

```
Out[33]: False
```

```
In [34]: df_vacunacion_Meta.isnull().sum()
```

```
Out[34]: PAIS_ISO3          0  
Continente          10  
ISO_continente       10  
NOMBRE_VACUNA        0  
NOMBRE_TIPO_VACUNA   0  
NOMBRE_COMPañIA      0  
FECHA_AUTORIZACION   0  
FECHA_INICIO_VACUNACION 0  
FUENTE_DATOS         0  
dtype: int64
```

Observo que aparecen 10 registros con el Campo de **Continente** y **ISO\_continente** nulos

```
In [35]: df_vacunacion_Meta[df_vacunacion_Meta['Continente'].isnull()]
```

```
Out[35]:
```

	PAIS_ISO3	Continente	ISO_continente	NOMBRE_VACUNA	NOMBRE_TIPO_VACUNA	NOMBRE_COMPañIA	FECHA_AUTORIZACION
227	SXM	None	None	AstraZeneca - Vaxzevria	Vaxzevria	AstraZeneca	
236	XKX	None	None	AstraZeneca - Vaxzevria	Vaxzevria	AstraZeneca	
335	TLS	None	None	AstraZeneca - Vaxzevria	Vaxzevria	AstraZeneca	23-02-20

Con el módulo **pycountry** obtenemos datos adicionales para el dataframe y se renombran las columnas con la función **.columns**

```
In [30]: import pycountry_convert as pc

def obtener_continente_ISO3(codigo_ISO3):
    try:
        continente_code = pc.country_alpha3_to_country_alpha2(codigo_ISO3)
        continente = pc.country_alpha2_to_continent_code(continente_code)
        continente_nombre = pc.convert_continent_code_to_continent_name(continente)
        return continente_nombre, continente
    except:
        return None, None

df_vacunacion_Meta[['Continente', 'ISO_continente']] = df_vacunacion_Meta['PAIS_ISO3'].apply(obtener_continente_ISO3).apply(
    lambda x: (x[0], x[1]), axis=0)

indice_ISO3 = df_vacunacion_Meta.columns.get_loc("PAIS_ISO3")

df_vacunacion_Meta.insert(indice_ISO3 + 1, "Continente", df_vacunacion_Meta.pop("Continente"))
df_vacunacion_Meta.insert(indice_ISO3 + 2, "ISO_continente", df_vacunacion_Meta.pop("ISO_continente"))
```

```
In [31]: df_vacunacion_Meta
```

```
Out[31]:
```

	PAIS_ISO3	Continente	ISO_continente	NOMBRE_VACUNA	NOMBRE_TIPO_VACUNA	NOMBRE_COMPANÍA	FECHA_AUTORIZACION
0	SHN	Africa	AF	AstraZeneca - AZD1222	AZD1222	AstraZeneca	
1	GRL	North America	NA	Moderna - mRNA-1273	mRNA-1273	Moderna	
2	FRO	Europe	EU	Moderna - mRNA-1273	mRNA-1273	Moderna	
3	FRO	Europe	EU	Pfizer BioNTech - Comirnaty	Comirnaty	Pfizer BioNTech	

```
In [34]: df_vacunacion_Meta.isnull().sum()
```

```
Out[34]: PAIS_ISO3      0
Continente      10
ISO_continente   10
NOMBRE_VACUNA    0
NOMBRE_TIPO_VACUNA  0
NOMBRE_COMPANÍA  0
FECHA_AUTORIZACION  0
FECHA_INICIO_VACUNACION  0
FUENTE_DATOS     0
dtype: int64
```

Observo que aparecen 10 registros con el Campo de **Continente y ISO\_continente** nulos

```
In [35]: df_vacunacion_Meta[df_vacunacion_Meta['Continente'].isnull()]
```

```
Out[35]:
```

	PAIS_ISO3	Continente	ISO_continente	NOMBRE_VACUNA	NOMBRE_TIPO_VACUNA	NOMBRE_COMPANÍA	FECHA_AUTORIZACION
227	SXM	None	None	AstraZeneca - Vaxzevria	Vaxzevria	AstraZeneca	
236	XKX	None	None	AstraZeneca - Vaxzevria	Vaxzevria	AstraZeneca	
335	TLS	None	None	AstraZeneca - Vaxzevria	Vaxzevria	AstraZeneca	23-02-20
343	PCN	None	None	AstraZeneca - Vaxzevria	Vaxzevria	AstraZeneca	17-05-20
404	SXM	None	None	Pfizer BioNTech - Comirnaty	Comirnaty	Pfizer BioNTech	

Sustituyo los valores nulos por los valores Sin Determinar y Z9

```
In [36]: df_vacunacion_Meta['Continente'].fillna('Sin Determinar', inplace=True)
df_vacunacion_Meta['ISO_continente'].fillna('Z9', inplace=True)
```

Verifico los cambios

```
In [37]: df_vacunacion_Meta[df_vacunacion_Meta.ISO_continente == 'Z9']
```

```
Out[37]:
```

	PAIS_ISO3	Continente	ISO_continente	NOMBRE_VACUNA	NOMBRE_TIPO_VACUNA	NOMBRE_COMPAÑIA	FECHA_AUTORIZACION
227	SXM	Sin Determinar	Z9	AstraZeneca - Vaxzevria	Vaxzevria	AstraZeneca	
236	KXX	Sin Determinar	Z9	AstraZeneca - Vaxzevria	Vaxzevria	AstraZeneca	
335	TLS	Sin Determinar	Z9	AstraZeneca - Vaxzevria	Vaxzevria	AstraZeneca	23-02-20
343	PCN	Sin Determinar	Z9	AstraZeneca - Vaxzevria	Vaxzevria	AstraZeneca	17-05-20
404	SXM	Sin Determinar	Z9	Pfizer BioNTech - Comirnaty	Comirnaty	Pfizer BioNTech	
411	KXX	Sin Determinar	Z9	Pfizer BioNTech - Comirnaty	Comirnaty	Pfizer BioNTech	
---	---	Sin	--	Pfizer BioNTech -	-	--	-----

```
In [38]: df_vacunacion_Meta.columns=['PAIS_ISO3', 'CONTINENTE', 'ISO_continente', 'NOMBRE_VACUNA', 'NOMBRE_TIPO_VACUNA', 'NOMBRE_COMPAÑIA', 'FECHA_AUTORIZACION', 'FECHA_INICIO_VACUNACION', 'FUENTE_DATOS']
```

Exporto el fichero depurado

```
In [ ]: df_vacunacion_Meta.to_csv('df_vacunacion_tipo.csv', index=False)
```

## Fuente 5: Data on testing for COVID-19 by week and country

El fichero se puede encontrar en el siguiente enlace:

[https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data\\_Frame\\_Testing\\_eu.ipynb](https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data_Frame_Testing_eu.ipynb)

Primero se crea un diccionario con toda la información sobre las variables, hacer esto hará que sea más sencillo llevar a cabo la depuración

```
data_dict = {
    "country": "String",
    "country_code": "2-letter ISO country code",
    "year_week": "yyyy-Www",
    "level": "National (archived dataset with national subnational data to week 36, 2022 is available on ECDC's website)",
    "region": "2-letter ISO country code where level is national",
    "region_name": "Country name where level is national",
    "new_cases": "Number of new confirmed cases",
    "tests_done": "Number of tests done",
    "population": "Numeric",
    "testing_rate": "Testing rate per 100,000 population",
    "positivity_rate": "Weekly test positivity (%): 100 x Number of new confirmed cases/number of tests done per week",
    "testing_data_source": [
        "Country API",
        "Country GitHub",
        "Country website",
        "Manual webscraping",
        "Other",
        "Survey",
    ]
}
```

```

    "TESSy: data provided directly by Member States to ECDC via TESSy"
]
}

```

## Ahora cargamos el dataset y configuramos nuestro entorno de trabajo

```

# importar paquetes
import pandas as pd
import numpy as np
import datetime as dt
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# configurar los gráficos
sns.set_style('whitegrid')
sns.set_palette('flare')

```

## Leemos el archivo

```

# Leer el archivo
df_datos4 = pd.read_csv('https://opendata.ecdc.europa.eu/covid19/testing/csv/data.csv')
# cargamos los datos
df_datos4.head()

```

## Empezamos revisando la información

```

df_datos4.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5460 entries, 0 to 5459
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   country                5460 non-null   object
1   country_code           5460 non-null   object
2   year_week              5460 non-null   object
3   level                  5460 non-null   object
4   region                 5460 non-null   object
5   region_name            5460 non-null   object
6   new_cases              5265 non-null   float64
7   tests_done             4406 non-null   float64
8   population             5460 non-null   int64
9   testing_rate           4406 non-null   float64
10  positivity_rate        4384 non-null   float64
11  testing_data_source    4406 non-null   object
dtypes: float64(4), int64(1), object(7)
memory usage: 512.0+ KB

# Revisamos si hay duplicados
df_datos4.duplicated().sum().any()
False

# Revisamos si hay nulos
df_datos4.isna().sum().any()
True

# Revisamos donde están los nulos
df_datos4.isna().sum()
country                0
country_code           0
year_week              0
level                  0
region                 0
region_name            0
new_cases              195
tests_done             1054

```

```

population          0
testing_rate        1054
positivity_rate      1076
testing_data_source  1054
dtype: int64

```

Una vez revisado el estatus inicial podemos empezar a hacer modificaciones

Este código convierte una columna en un DataFrame en una categoría y verifica que la conversión se realice correctamente.

```

# Modificar categoría
df_datos4['testing_data_source'] = df_datos4['testing_data_source'].astype('category'); assert
df_datos4['testing_data_source'].dtype == 'category'

```

Este código convierte la columna 'year\_week' en un DataFrame en un tipo de datos de fecha y hora utilizando la función 'pd.to\_datetime'. Luego, se extraen el número de semana y el año de la columna 'year\_week' y se almacenan en la misma columna, pero en formato de cadena de texto utilizando el método 'dt.strftime'.

```

# Convertir la columna 'year_week' a tipo datetime
df_datos4['year_week'] = pd.to_datetime(df_datos4['year_week'] + '-1', format='%Y-W%-w')
# Extraer el número de semana y el año
df_datos4['year_week'] = df_datos4['year_week'].dt.strftime('%Y-%W')

```

Ahora, tenemos que obtener el código ISO3

```

# Sabemos que disponemos del nombre del país en inglés
df_datos4['country']:
# Sabemos que disponemos del nombre del país en inglés
df_datos4['country']
0          Austria
1          Austria
2          Austria
3          Austria
4          Austria
...
5455        Sweden
5456        Sweden
5457        Sweden
5458        Sweden
5459        Sweden
Name: country, Length: 5460, dtype: object

```

Podemos hacerlo utilizando una función que implemente el módulo pycountry, definimos obtener\_iso3

La función toma un parámetro country, que representa el nombre del país para el cual se desea obtener el código ISO 3. A continuación, utiliza la función pycountry.countries.get(name=country) para buscar el objeto Country correspondiente al nombre del país en la biblioteca pycountry.

Si se encuentra un objeto Country válido para el nombre del país, se devuelve su código ISO 3 utilizando el atributo alpha\_3. En caso de que no se encuentre un objeto Country válido, la función captura la excepción LookupError y no realiza ninguna acción adicional.

Finalmente, si se devuelve un código ISO 3 válido, este se asigna a la columna 'iso3' en el DataFrame 'df\_datos4' utilizando el método apply en la columna 'country'.

```

import pycountry
def obtener_iso3(country):

```

```

try:
    pais = pycountry.countries.get(name=country)
    if pais is not None:
        return pais.alpha_3
except LookupError:
    pass
return None

# Obtener el código ISO 3 correspondiente a los nombres de país en la columna 'country'
df_datos4.insert(1, 'iso3', df_datos4['country'].apply(obtener_iso3))

```

Y ahora sólo verificamos que haya funcionado correctamente

```

df_datos4['iso3']

0      AUT
1      AUT
2      AUT
3      AUT
4      AUT
...
5455   SWE
5456   SWE
5457   SWE
5458   SWE
5459   SWE
Name: iso3, Length: 5460, dtype: object

```

Después de revisar los valores nulos vemos que sería mejor eliminarlos así procedemos a ello

```

# Eliminar nulos
df_datos4.dropna(subset=['new_cases', 'tests_done', 'testing_rate', 'positivity_rate',
'testing_data_source'], inplace=True)

```

Y eliminamos las columnas que no utilizaremos

```

df_datos4.drop(['country_code', 'region_name', 'region', 'country'], axis=1, inplace=True)
### Verificamos
df_datos4.info()
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4384 entries, 40 to 5459
Data columns (total 9 columns):
#   Column                Non-Null Count  Dtype
---  -
0   iso3                  4384 non-null  object
1   year_week             4384 non-null  object
2   level                 4384 non-null  object
3   new_cases             4384 non-null  float64
4   tests_done            4384 non-null  float64
5   population            4384 non-null  int64
6   testing_rate          4384 non-null  float64
7   positivity_rate       4384 non-null  float64
8   testing_data_source   4384 non-null  category
dtypes: category(1), float64(4), int64(1), object(3)
memory usage: 312.6+ KB

```

Y por último exportamos a CSV

```

import os

def guardar_como_csv(df, nombre_archivo):
    # Obtener la ruta completa del directorio actual
    current_directory = os.getcwd()
    # Definir la ubicación y el nombre del archivo CSV
    file_path = os.path.join(current_directory, nombre_archivo)

```

```
# Guardar el DataFrame como un archivo CSV en la ubicación especificada
df.to_csv(file_path, index=False)

# Llamar a la función para guardar el DataFrame df_datos4 como un archivo CSV
guardar_como_csv(df_datos4, "DEFMODTestingCovid19.csv")
```

### Propuesta para la base de datos SQL

- iso3: VARCHAR o CHAR (cadena de caracteres de longitud fija o variable que representa un código de país de 3 letras).
- year\_week: VARCHAR o CHAR (cadena de caracteres de longitud fija o variable que representa una semana en formato "año-semana").
- level: VARCHAR o CHAR (cadena de caracteres de longitud fija o variable que representa el nivel geográfico).
- new\_cases: FLOAT o DECIMAL (número decimal que representa la cantidad de nuevos casos).
- tests\_done: FLOAT o DECIMAL (número decimal que representa la cantidad de pruebas realizadas).
- population: INTEGER o BIGINT (número entero que representa la población).
- testing\_rate: FLOAT o DECIMAL (número decimal que representa la tasa de pruebas).
- positivity\_rate: FLOAT o DECIMAL (número decimal que representa la tasa de positividad).
- testing\_data\_source: VARCHAR o CHAR (cadena de caracteres de longitud fija o variable que representa la fuente de datos de pruebas).

### Fuente 6: Data on hospital and ICU admission rates and current occupancy for COVID-19

El fichero se puede encontrar en el siguiente enlace:

[https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data\\_Frame\\_Pacientes\\_Hospitalizados%20%26%20UCI\\_eu.ipynb](https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data_Frame_Pacientes_Hospitalizados%20%26%20UCI_eu.ipynb)

### Fuente de Datos - European Centre for Disease Prevention and Control

<https://www.ecdc.europa.eu/en/publications-data/download-data-hospital-and-icu-admission-rates-and-current-occupancy-covid-19>

### Descarga de datos

Ingresos en hospital UCI fueron registrados: <https://www.ecdc.europa.eu/en/publications-data/download-data-hospital-and-icu-admission-rates-and-current-occupancy-covid-19>

### Información del Dataset

Los usuarios deben tener en cuenta que los archivos de datos descargables contienen información sobre las tasas de admisión de hospitalización y Unidad de Cuidados Intensivos (UCI) y la ocupación actual para COVID-19 por fecha y país. Cada fila contiene los datos correspondientes a una determinada fecha (día o semana) y por país. El archivo se actualiza semanalmente. Puede utilizar los datos de acuerdo con la política de derechos de autor del ECDC.



El dataset que estaremos trabajando contiene Información acerca de Ocupación diaria del hospital, Ocupación diaria de la UCI, Nuevos ingresos hospitalarios semanales por 100k, Nuevas admisiones semanales en UCI por 100k (Daily hospital occupancy, Daily ICU occupancy, Weekly new hospital admissions per 100k, Weekly new ICU admissions per 100k) en países europeos.

Columnas: country, indicator, date, year\_week, value, source, url

```
In [2]: import pandas as pd
import numpy as np
import datetime
```

## Acceso al dataset de hospitalizaciones en UCI

Exploración e información del DataFrame df\_hosp\_UCI:

```
In [4]: df_hosp_UCI = pd.read_csv ('https://opendata.ecdc.europa.eu/covid19/hospitalicuadmissionrates/csv/data.csv')
df_hosp_UCI
```

```
Out[4]:
```

	country	indicator	date	year_week	value	source	url
0	Austria	Daily hospital occupancy	2020-04-01	2020-W14	856.000000	Country_Website	NaN
1	Austria	Daily hospital occupancy	2020-04-02	2020-W14	823.000000	Country_Website	NaN
2	Austria	Daily hospital occupancy	2020-04-03	2020-W14	829.000000	Country_Website	NaN
3	Austria	Daily hospital occupancy	2020-04-04	2020-W14	826.000000	Country_Website	NaN
4	Austria	Daily hospital occupancy	2020-04-05	2020-W14	712.000000	Country_Website	NaN
...	...	...	...	...	...	...	...
27628	Sweden	Weekly new ICU admissions per 100k	2023-05-28	2023-W21	0.095672	TESSy COVID-19 combined sources	NaN
27629	Sweden	Weekly new ICU admissions per 100k	2023-06-04	2023-W22	0.019134	TESSy COVID-19 combined sources	NaN
27630	Sweden	Weekly new ICU admissions per 100k	2023-06-11	2023-W23	0.047836	TESSy COVID-19 combined sources	NaN
27631	Sweden	Weekly new ICU admissions per 100k	2023-06-18	2023-W24	0.028702	TESSy COVID-19 combined sources	NaN
27632	Sweden	Weekly new ICU admissions per 100k	2023-06-25	2023-W25	0.028702	TESSy COVID-19 combined sources	NaN

27633 rows × 7 columns

```
In [5]: df_hosp_UCI['country'].unique()
```

```
Out[5]: array(['Austria', 'Belgium', 'Bulgaria', 'Cyprus', 'Czechia', 'Estonia',
'France', 'Germany', 'Greece', 'Hungary', 'Iceland', 'Ireland',
'Italy', 'Latvia', 'Liechtenstein', 'Lithuania', 'Luxembourg',
'Malta', 'Netherlands', 'Portugal', 'Romania', 'Slovakia',
'Slovenia', 'Spain', 'Sweden'], dtype=object)
```

Carga del df países para realizar el “join”:

```
In [6]: df_paises = pd.read_csv ('/content/TFM/Ficheros_Depurados/paises_region_oms_v2.csv')
df_paises
```

```
Out[6]:
```

	PAIS_ISO3	PAIS_ISO2	PAIS_NOM	COD_CONTINENTE	CONTINENTE	OMS_REGION	DESC_OMS_REGION	PAIS_NOM_2
0	AGO	AO	Angola	AF	Africa	AFRO	Africa	Angola
1	BDI	BI	Burundi	AF	Africa	AFRO	Africa	Burundi
2	BEN	BJ	Benin	AF	Africa	AFRO	Africa	Benin
3	BFA	BF	Burkina Faso	AF	Africa	AFRO	Africa	Burkina Faso
4	BWA	BW	Botswana	AF	Africa	AFRO	Africa	Botswana
...	...	...	...	...	...	...	...	...
244	VCT	VC	Saint Vincent and the Grenadines	NaN	North America	AMRO	América	Saint Vincent and the Grenadines
245	VGB	VG	British Virgin Islands	NaN	North America	AMRO	América	British Virgin Islands
246	BLM	BL	San Bartolomé	NaN	North America	AMRO	América	Saint Barthélemy
247	MAF	MF	San Martín	NaN	North America	AMRO	América	Saint Martin
248	SXM	SX	San Martín	NaN	North America	AMRO	América	San Martín

249 rows × 8 columns

```
In [7]: df_hosp_UCI_NEW = pd.merge(left=df_hosp_UCI, right=df_paises,
how='left', left_on='country', right_on='PAIS_NOM_2')
```

```
In [9]: df_hosp_UCI_NEW.head()
```

```
Out[9]:
```

	country	indicator	date	year_week	value	source	url	PAIS_ISO3	PAIS_ISO2	PAIS_NOM	COD_CONTINENTE	CONTINEI
0	Austria	Daily hospital occupancy	2020-04-01	2020-W14	856.0	Country_Website	NaN	AUT	AT	Austria	EU	Eur
1	Austria	Daily hospital occupancy	2020-04-02	2020-W14	823.0	Country_Website	NaN	AUT	AT	Austria	EU	Eur
2	Austria	Daily hospital occupancy	2020-04-03	2020-W14	829.0	Country_Website	NaN	AUT	AT	Austria	EU	Eur
3	Austria	Daily hospital occupancy	2020-04-04	2020-W14	826.0	Country_Website	NaN	AUT	AT	Austria	EU	Eur
4	Austria	Daily hospital occupancy	2020-04-05	2020-W14	712.0	Country_Website	NaN	AUT	AT	Austria	EU	Eur

Depuración del nuevo df:

```
In [10]: df_hosp_UCI_NEW.isnull().sum()
```

```
Out[10]: country          0
indicator          0
date              0
year_week         0
value            0
source           0
url             27633
PAIS_ISO3         0
PAIS_ISO2         0
PAIS_NOM          0
COD_CONTINENTE    0
CONTINENTE        0
OMS_REGION        0
DESC_OMS_REGION   0
PAIS_NOM_2        0
dtype: int64
```

```
In [11]: df_hosp_UCI_NEW.columns
```

```
Out[11]: Index(['country', 'indicator', 'date', 'year_week', 'value', 'source', 'url',
               'PAIS_ISO3', 'PAIS_ISO2', 'PAIS_NOM', 'COD_CONTINENTE', 'CONTINENTE',
               'OMS_REGION', 'DESC_OMS_REGION', 'PAIS_NOM_2'],
              dtype='object')
```

```
In [12]: df_hosp_UCI_NEW = df_hosp_UCI_NEW[['PAIS_ISO3', 'PAIS_NOM', 'indicator', 'date', 'year_week', 'value', 'source', 'url']]
```

```
In [13]: df_hosp_UCI_NEW.head()
```

```
Out[13]:
```

	PAIS_ISO3	PAIS_NOM	indicator	date	year_week	value	source	url
0	AUT	Austria	Daily hospital occupancy	2020-04-01	2020-W14	856.0	Country_Website	NaN
1	AUT	Austria	Daily hospital occupancy	2020-04-02	2020-W14	823.0	Country_Website	NaN
2	AUT	Austria	Daily hospital occupancy	2020-04-03	2020-W14	829.0	Country_Website	NaN
3	AUT	Austria	Daily hospital occupancy	2020-04-04	2020-W14	826.0	Country_Website	NaN
4	AUT	Austria	Daily hospital occupancy	2020-04-05	2020-W14	712.0	Country_Website	NaN

```
In [14]: df_hosp_UCI_NEW[df_hosp_UCI_NEW['PAIS_ISO3'].isnull()].count()
```

```
Out[14]: PAIS_ISO3      0
PAIS_NOM      0
indicator      0
date          0
year_week     0
value         0
source        0
url           0
dtype: int64
```

Transformar la variable **date** y **year\_week** en formato **datetime**

```
In [15]: df_hosp_UCI_NEW['date'] = pd.to_datetime(df_hosp_UCI_NEW['date']).dt.strftime('%d-%m-%Y')
```

Asignación de formato fecha a columna “year\_week”:

```
In [16]: # Convertir la columna 'year_week' a tipo datetime
df_hosp_UCI_NEW['year_week'] = pd.to_datetime(df_hosp_UCI_NEW['year_week'] + '-1', format='%Y-W%-1')
# Extraer el número de semana y el año
df_hosp_UCI_NEW['year_week'] = df_hosp_UCI_NEW['year_week'].dt.strftime('%Y-%W')
```

Eliminación de la columna “url”:

In [17]:

```
df_hosp_UCI_NEW
```

Out[17]:

	PAIS_ISO3	PAIS_NOM	indicator	date	year_week	value	source	url
0	AUT	Austria	Daily hospital occupancy	01-04-2020	2020-14	856.000000	Country_Website	NaN
1	AUT	Austria	Daily hospital occupancy	02-04-2020	2020-14	823.000000	Country_Website	NaN
2	AUT	Austria	Daily hospital occupancy	03-04-2020	2020-14	829.000000	Country_Website	NaN
3	AUT	Austria	Daily hospital occupancy	04-04-2020	2020-14	826.000000	Country_Website	NaN
4	AUT	Austria	Daily hospital occupancy	05-04-2020	2020-14	712.000000	Country_Website	NaN
...	...	...	...	...	...	...	...	...
27628	SWE	Sweden	Weekly new ICU admissions per 100k	28-05-2023	2023-21	0.095672	TESSy COVID-19 combined sources	NaN
27629	SWE	Sweden	Weekly new ICU admissions per 100k	04-06-2023	2023-22	0.019134	TESSy COVID-19 combined sources	NaN
27630	SWE	Sweden	Weekly new ICU admissions per 100k	11-06-2023	2023-23	0.047836	TESSy COVID-19 combined sources	NaN
27631	SWE	Sweden	Weekly new ICU admissions per 100k	18-06-2023	2023-24	0.028702	TESSy COVID-19 combined sources	NaN
27632	SWE	Sweden	Weekly new ICU admissions per 100k	25-06-2023	2023-25	0.028702	TESSy COVID-19 combined sources	NaN

27633 rows × 8 columns

Elimino la columna **url** del **df\_hosp\_UCI\_NEW** mediante la función **.drop**

In [18]:

```
df_hosp_UCI_NEW.drop(['url'], axis=1, inplace=True)
```

Validación de nulos:

```
In [19]: df_hosp_UCI_NEW[df_hosp_UCI_NEW['PAIS_ISO3'].isnull()]
```

```
Out[19]: PAIS_ISO3 PAIS_NOM indicator date year_week value source
```

```
In [20]: df_hosp_UCI_NEW
df_hosp_UCI_NEW.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 27633 entries, 0 to 27632
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   PAIS_ISO3    27633 non-null  object
1   PAIS_NOM     27633 non-null  object
2   indicator    27633 non-null  object
3   date         27633 non-null  object
4   year_week    27633 non-null  object
5   value        27633 non-null  float64
6   source       27633 non-null  object
dtypes: float64(1), object(6)
memory usage: 1.7+ MB
```

```
In [21]: df_hosp_UCI_NEW.duplicated().values.any()
```

```
Out[21]: False
```

```
In [22]: df_hosp_UCI_NEW.isnull().sum()
```

```
Out[22]: PAIS_ISO3    0
PAIS_NOM    0
indicator    0
date         0
year_week    0
value        0
source       0
dtype: int64
```

```
In [23]: df_hosp_UCI_NEW.columns = ['PAIS_ISO3', 'PAIS_NOM', 'INDICADOR', 'FECHA', 'ANY_SEMANA', 'VALOR', 'FUENTE_ORIGEN']
```

Exportación del df final a formato CSV:

```
In [24]: df_hosp_UCI_NEW.to_csv('/content/TFM/Ficheros_Depurados/df_hosp_UCI_NEW.csv', index= False)
```

## Fuente 7: Datos sobre casos diarios registrados por estado en USA

El fichero se puede encontrar en el siguiente enlace:

[https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data\\_Frame\\_csse\\_covid\\_daily\\_repo  
rt\\_US.ipynb](https://github.com/patriciaapenat/TFM/blob/main/Notebooks/Data_Frame_csse_covid_daily_repo_rt_US.ipynb)

 Open in Colab

In [1]: `! git clone https://github.com/patriciaapenat/TFM.git`

```
Cloning into 'TFM'...
remote: Enumerating objects: 195, done.
remote: Counting objects: 100% (91/91), done.
remote: Compressing objects: 100% (46/46), done.
remote: Total 195 (delta 50), reused 79 (delta 44), pack-reused 104
Receiving objects: 100% (195/195), 10.21 MiB | 9.21 MiB/s, done.
Resolving deltas: 100% (95/95), done.
```

USA daily state reports (csse\_covid\_19\_daily\_reports\_us) This table contains an aggregation of each USA State level data.

File naming convention MM-DD-YYYY.csv in UTC.

Field description

Province\_State - The name of the State within the USA.

Country\_Region - The name of the Country (US).

Last\_Update - The most recent date the file was pushed.

Lat - Latitude.

Long\_ - Longitude.

Long\_ - Longitude.

Confirmed - Aggregated case count for the state.

Deaths - Aggregated death toll for the state.

Recovered - Aggregated Recovered case count for the state.

Active - Aggregated confirmed cases that have not been resolved (Active cases = total cases - total recovered - total deaths).

FIPS - Federal Information Processing Standards code that uniquely identifies counties within the USA.

Incident\_Rate - cases per 100,000 persons.

Total\_Test\_Results - Total number of people who have been tested.

People\_Hospitalized - Total number of people hospitalized. (Nullified on Aug 31, see Issue #3083)

Case\_Fatality\_Ratio - Number recorded deaths \* 100/ Number confirmed cases.

UID - Unique Identifier for each row entry.

ISO3 - Officially assigned country code identifiers.

Testing\_Rate - Total test results per 100,000 persons. The "total test results" are equal to "Total test results (Positive + Negative)" from COVID Tracking Project.

Hospitalization\_Rate - US Hospitalization Rate (%): = Total number hospitalized / Number cases. The "Total number hospitalized" is the "Hospitalized – Cumulative" count from COVID Tracking Project. The "hospitalization rate" and "Total number hospitalized" is only presented for those states which provide cumulative hospital data. (Nullified on Aug 31, see Issue #3083)

Update frequency Once per day between 04:45 and 05:15 UTC.

update frequency once per day between 07:45 and 08:15 UTC.

```
In [2]: from datetime import datetime, timedelta
import pandas as pd
import numpy as np
from datetime import datetime
```

Carga del archivo datos\_combinados\_csse\_covid\_19\_daily\_reports\_us.csv en el dataframe df\_daily\_reports\_us

```
In [3]: df_daily = pd.read_csv('/content/TFM/Ficheros_Depurados/datos_combinados_csse_covid_19_daily_reports_us.csv')
```

```
In [4]: df_daily.tail()
```

```
Out[4]:
```

	Province_State	Country_Region	Last_Update	Lat	Long_	Confirmed	Deaths	Recovered	Active	FIPS	...	Total_Test_Res
61609	Virginia	US	2023-01-01 04:31:23	37.7693	-78.1700	2199302	22670	NaN	NaN	51.0	...	↑
61610	Washington	US	2023-01-01 04:31:23	47.4009	-121.4905	1883676	15038	NaN	NaN	53.0	...	↑
61611	West Virginia	US	2023-01-01 04:31:23	38.4912	-80.9545	624721	7672	NaN	NaN	54.0	...	↑
61612	Wisconsin	US	2023-01-01 04:31:23	44.2685	-89.6165	1960878	15802	NaN	NaN	55.0	...	↑
61613	Wyoming	US	2023-01-01 04:31:23	42.7560	-107.3025	182847	1958	NaN	NaN	56.0	...	↑

5 rows × 21 columns

Transponemos el dataframe para visualizar las variables más comodamente

```
In [5]: df_daily.T
```

```
Out[5]:
```

	0	1	2	3	4	5	6	7
Province_State	Alabama	Alaska	American Samoa	Arizona	Arkansas	California	Colorado	Connecticut
Country_Region	US	US	US	US	US	US	US	US
Last_Update	2021-01-02 05:30:44	2021-01-02 05:30:44	2021-01-02 05:30:44	2021-01-02 05:30:44	2021-01-02 05:30:44	2021-01-02 05:30:44	2021-01-02 05:30:44	2021-01-02 05:30:44
Lat	32.3182	61.3707	-14.271	33.7298	34.9697	36.1162	39.0598	41.5978
Long_	-86.9023	-152.4044	-170.132	-111.4312	-92.3731	-119.6816	-105.3111	-72.7554
Confirmed	365747	47019	0	530267	229442	2434971	362438	185708
Deaths	4872	206	0	9015	3711	26298	5435	5995
Recovered	202137.0	7165.0	NaN	76934.0	199247.0	NaN	18102.0	9800.0
Active	158738.0	39648.0	NaN	444318.0	26484.0	NaN	314186.0	169913.0
FIPS	1.0	2.0	60.0	4.0	5.0	6.0	8.0	9.0
Incident_Rate	7459.375895	6427.355802	0.0	7285.171274	7602.945718	6164.469663	5854.774381	5208.781229
Total_Test_Results	NaN	1275750.0	2140.0	5155330.0	2051488.0	33058311.0	4444206.0	4320693.0

21 rows x 61614 columns

```
In [6]: df_daily.shape
```

```
Out[6]: (61614, 21)
```

```
In [7]: df_daily.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 61614 entries, 0 to 61613
Data columns (total 21 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Province_State      61614 non-null  object
1   Country_Region      61614 non-null  object
2   Last_Update         61595 non-null  object
3   Lat                 59472 non-null  float64
4   Long_              59472 non-null  float64
5   Confirmed           61614 non-null  int64
6   Deaths             61614 non-null  int64
7   Recovered           15122 non-null  float64
8   Active              15122 non-null  float64
9   FIPS                61595 non-null  float64
10  Incident_Rate       59472 non-null  float64
11  Total_Test_Results  36637 non-null  float64
12  People_Hospitalized 5129 non-null   float64
13  Case_Fatality_Ratio 49027 non-null  float64
14  UID                 61614 non-null  float64
15  ISO3                61614 non-null  object
16  Testing_Rate        45921 non-null  float64
17  Hospitalization_Rate 5129 non-null   float64
18  Date                51754 non-null  object
19  People_Testing      11816 non-null  float64
20  Mortality_Rate      12027 non-null  float64
dtypes: float64(14), int64(2), object(5)
memory usage: 9.9+ MB
```

Transformamos los campos Date a tipo fecha

```
In [8]: df_daily['Date'] = pd.to_datetime(df_daily['Date']).dt.strftime('%d.%m.%Y')
```

Corroboramos la conversión de los campos

```
In [9]: df_daily.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 61614 entries, 0 to 61613
Data columns (total 21 columns):
#   Column              Non-Null Count  Dtype
---  -
0   Province_State      61614 non-null  object
1   Country_Region      61614 non-null  object
2   Last_Update         61595 non-null  object
3   Lat                 59472 non-null  float64
4   Long_              59472 non-null  float64
5   Confirmed           61614 non-null  int64
6   Deaths             61614 non-null  int64
7   Recovered           15122 non-null  float64
8   Active              15122 non-null  float64
9   FIPS                61595 non-null  float64
10  Incident_Rate       59472 non-null  float64
11  Total_Test_Results  36637 non-null  float64
12  People_Hospitalized 5129 non-null   float64
13  Case_Fatality_Ratio 49027 non-null  float64
14  UID                 61614 non-null  float64
15  ISO3                61614 non-null  object
16  Testing_Rate        45921 non-null  float64
17  Hospitalization_Rate 5129 non-null   float64
18  Date                51754 non-null  object
19  People_Testing      11816 non-null  float64
20  Mortality_Rate      12027 non-null  float64
dtypes: float64(14), int64(2), object(5)
memory usage: 9.9+ MB
```



Contamos las filas duplicadas si existen

```
In [10]: df_daily.duplicated().sum()
```

Out[10]: 54

Eliminamos las filas duplicadas

```
In [11]: df_daily = df_daily.drop_duplicates()
```

Contamos los valores nulos si es que existen

```
In [12]: df_daily.isnull().values.sum()
```

Out[12]: 374428

Exploramos en que variables existen nulos y que cantidad

```
In [13]: df_daily.isnull().sum()
```

```
Out[13]: Province_State      0
Country_Region             0
Last_Update                19
Lat                       2140
Long_                     2140
Confirmed                  0
Deaths                    0
Recovered                 46438
Active                   46438
FIPS                      19
Incident_Rate             2140
Total_Test_Results        24923
```

```
In [14]: df_daily.describe().T
```

```
Out[14]:
```

	count	mean	std	min	25%	50%	75%	max
Lat	59420.0	3.684030e+01	1.078922e+01	-1.427100e+01	3.484050e+01	3.906185e+01	4.236165e+01	6.137070e+01
Long_	59420.0	-8.520742e+01	4.930721e+01	-1.701322e+02	-1.011658e+02	-8.794420e+01	-7.702680e+01	1.456739e+02
Confirmed	61560.0	8.727221e+05	1.456052e+06	0.000000e+00	6.490025e+04	3.395635e+05	1.021065e+06	1.212970e+07
Deaths	61560.0	1.157355e+04	1.714695e+04	0.000000e+00	1.102000e+03	4.892000e+03	1.469050e+04	1.011590e+05
Recovered	15122.0	8.578908e+04	1.933303e+05	0.000000e+00	3.872000e+03	1.754850e+04	8.425575e+04	2.470308e+06
Active	15122.0	5.936916e+04	1.190129e+05	0.000000e+00	3.166000e+03	1.475250e+04	6.111500e+04	1.408516e+06
FIPS	61541.0	3.284888e+03	1.724951e+04	1.000000e+00	1.800000e+01	3.300000e+01	4.800000e+01	9.999900e+04
Incident_Rate	59420.0	1.489822e+04	1.130987e+04	0.000000e+00	4.116387e+03	1.231563e+04	2.547638e+04	5.892799e+04
Total_Test_Results	36637.0	1.144235e+07	2.018320e+07	1.768000e+03	1.771434e+06	4.784927e+06	1.296786e+07	1.844461e+08
People_Hospitalized	5129.0	6.151227e+03	1.379109e+04	2.000000e+00	5.580000e+02	2.014000e+03	6.001000e+03	8.999500e+04
Case_Fatality_Ratio	48973.0	1.330293e+00	5.956818e-01	0.000000e+00	1.007556e+00	1.287507e+00	1.591811e+00	8.803612e+00
UID	61560.0	7.676455e+07	2.357348e+07	1.600000e+01	8.400001e+07	8.400003e+07	8.400004e+07	8.410000e+07
Testing_Rate	45921.0	1.161119e+05	1.307875e+05	5.973514e-08	7.422100e+03	7.692253e+04	1.787269e+05	7.901759e+05

```
In [15]: df_daily.sort_values(by=['Province_State', 'Date'], na_position='first').T
```

```
Out[15]:
```

	116	290	464	638	812	986	1160	1334
Province_State	Alabama	Alabama	Alabama	Alabama	Alabama	Alabama	Alabama	Alabama
Country_Region	US	US	US	US	US	US	US	US
Last_Update	2023-01-02 04:31:21	2023-01-03 04:31:34	2023-01-04 04:31:12	2023-01-05 04:31:16	2023-01-06 04:31:58	2023-01-07 04:31:23	2023-01-08 04:32:04	2023-01-09 04:31:38
Lat	32.3182	32.3182	32.3182	32.3182	32.3182	32.3182	32.3182	32.3182
Long_	-86.9023	-86.9023	-86.9023	-86.9023	-86.9023	-86.9023	-86.9023	-86.9023
Confirmed	1568934	1568934	1568934	1587224	1587224	1587224	1587224	1587224
Deaths	20737	20737	20737	20776	20776	20776	20776	20776
Recovered	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Active	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
FIPS	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Incident_Rate	31998.262354	31998.262354	31998.262354	32371.285195	32371.285195	32371.285195	32371.285195	32371.285195
Total_Test_Results	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
People_Hospitalized	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Case_Fatality_Ratio	1.321725	1.321725	1.321725	1.308952	1.308952	1.308952	1.308952	1.308952

```

In [16]: df_daily['Date'].isnull().sum()

Out[16]: 9806

Nuevo dataframe daily_act con las columnas necesarias solo

In [17]: df_daily_act = df_daily[['Province_State', 'Confirmed', 'Deaths', 'Recovered', 'Active', 'ISO3', 'Date']]

In [18]: df_daily_act

Out[18]:

```

	Province_State	Confirmed	Deaths	Recovered	Active	ISO3	Date
0	Alabama	365747	4872	202137.0	158738.0	USA	01.01.2021
1	Alaska	47019	206	7165.0	39648.0	USA	01.01.2021
2	American Samoa	0	0	NaN	NaN	ASM	01.01.2021
3	Arizona	530267	9015	76934.0	444318.0	USA	01.01.2021
4	Arkansas	229442	3711	199247.0	26484.0	USA	01.01.2021
...	...	...	...	...	...	...	...
61609	Virginia	2199302	22670	NaN	NaN	USA	NaN
61610	Washington	1883676	15038	NaN	NaN	USA	NaN

61560 rows × 7 columns

```

In [19]: df_daily_act.columns = ['PROVINCIA_ESTADO', 'CASOS_CONFIRMADOS', 'DEFUNCIONES', 'CASOS_RECUPERADOS', 'CASOS_ACTIVOS', 'PAIS_ISO3', 'FECHA']

In [20]: df_daily_act

Out[20]:

```

	PROVINCIA_ESTADO	CASOS_CONFIRMADOS	DEFUNCIONES	CASOS_RECUPERADOS	CASOS_ACTIVOS	PAIS_ISO3	FECHA
0	Alabama	365747	4872	202137.0	158738.0	USA	01.01.2021
1	Alaska	47019	206	7165.0	39648.0	USA	01.01.2021
2	American Samoa	0	0	NaN	NaN	ASM	01.01.2021
3	Arizona	530267	9015	76934.0	444318.0	USA	01.01.2021
4	Arkansas	229442	3711	199247.0	26484.0	USA	01.01.2021
...	...	...	...	...	...	...	...
61609	Virginia	2199302	22670	NaN	NaN	USA	NaN
61610	Washington	1883676	15038	NaN	NaN	USA	NaN
61611	West Virginia	624721	7672	NaN	NaN	USA	NaN
61612	Wisconsin	1960878	15802	NaN	NaN	USA	NaN
61613	Wyoming	182847	1958	NaN	NaN	USA	NaN

61560 rows × 7 columns

```

In [21]: df_daily_act.to_csv("/content/TFM/Ficheros_Depurados/df_daily_report_us_final.csv", index = False)

```

 Open in Colab

```
In [1]: ! git clone https://github.com/patriciaapenat/TFM.git
```

```
Cloning into 'TFM'...
remote: Enumerating objects: 195, done.
remote: Counting objects: 100% (91/91), done.
remote: Compressing objects: 100% (46/46), done.
remote: Total 195 (delta 50), reused 79 (delta 44), pack-reused 104
Receiving objects: 100% (195/195), 10.21 MiB | 9.21 MiB/s, done.
Resolving deltas: 100% (95/95), done.
```

USA daily state reports (csse\_covid\_19\_daily\_reports\_us) This table contains an aggregation of each USA State level data.

File naming convention MM-DD-YYYY.csv in UTC.

Field description

Province\_State - The name of the State within the USA.

Country\_Region - The name of the Country (US).

Last\_Update - The most recent date the file was pushed.

Lat - Latitude.

Long\_ - Longitude.

Long\_ - Longitude.

Confirmed - Aggregated case count for the state.

Deaths - Aggregated death toll for the state.

Recovered - Aggregated Recovered case count for the state.

Active - Aggregated confirmed cases that have not been resolved (Active cases = total cases - total recovered - total deaths).

FIPS - Federal Information Processing Standards code that uniquely identifies counties within the USA.

Incident\_Rate - cases per 100,000 persons.

Total\_Test\_Results - Total number of people who have been tested.

People\_Hospitalized - Total number of people hospitalized. (Nullified on Aug 31, see Issue #3083)

Case\_Fatality\_Ratio - Number recorded deaths \* 100/ Number confirmed cases.

UID - Unique Identifier for each row entry.

ISO3 - Officially assigned country code identifiers.

Testing\_Rate - Total test results per 100,000 persons. The "total test results" are equal to "Total test results (Positive + Negative)" from COVID Tracking Project.

Hospitalization\_Rate - US Hospitalization Rate (%): = Total number hospitalized / Number cases. The "Total number hospitalized" is the "Hospitalized - Cumulative" count from COVID Tracking Project. The "hospitalization rate" and "Total number hospitalized" is only presented for those states which provide cumulative hospital data. (Nullified on Aug 31, see Issue #3083)

Update frequency Once per day between 04:45 and 05:15 UTC.

Report inspecting some per day between 2020 and 2022 data.

```
In [2]: from datetime import datetime, timedelta
import pandas as pd
import numpy as np
from datetime import datetime
```

Carga del archivo datos\_combinados\_csse\_covid\_19\_daily\_reports\_us.csv en el dataframe df\_daily\_reports\_us

```
In [3]: df_daily = pd.read_csv('/content/TFM/Ficheros_Depurados/datos_combinados_csse_covid_19_daily_reports_us.csv')
```

```
In [4]: df_daily.tail()
```

```
Out[4]:
```

	Province_State	Country_Region	Last_Update	Lat	Long_	Confirmed	Deaths	Recovered	Active	FIPS	—	Total_Test_Res
61609	Virginia	US	2023-01-01 04:31:23	37.7693	-78.1700	2199302	22670	NaN	NaN	51.0	—	f
61610	Washington	US	2023-01-01 04:31:23	47.4009	-121.4905	1883676	15038	NaN	NaN	53.0	—	f
61611	West Virginia	US	2023-01-01 04:31:23	38.4912	-80.9545	624721	7672	NaN	NaN	54.0	—	f
61612	Wisconsin	US	2023-01-01 04:31:23	44.2685	-89.6165	1960878	15802	NaN	NaN	55.0	—	f
61613	Wyoming	US	2023-01-01 04:31:23	42.7560	-107.3025	182847	1958	NaN	NaN	56.0	—	f

5 rows × 13 columns

Transponemos el dataframe para visualizar las variables más comodamente

```
In [5]: df_daily.T
```

```
Out[5]:
```

	0	1	2	3	4	5	6	7
Province_State	Alabama	Alaska	American Samoa	Arizona	Arkansas	California	Colorado	Connecticut
Country_Region	US	US	US	US	US	US	US	US
Last_Update	2021-01-02 05:30:44	2021-01-02 05:30:44	2021-01-02 05:30:44	2021-01-02 05:30:44	2021-01-02 05:30:44	2021-01-02 05:30:44	2021-01-02 05:30:44	2021-01-02 05:30:44
Lat	32.3182	61.3707	-14.271	33.7298	34.9697	36.1162	39.0598	41.5978
Long_	-86.9023	-152.4044	-170.132	-111.4312	-92.3731	-119.6816	-105.3111	-72.7554
Confirmed	365747	47019	0	530267	229442	2434971	362438	185708
Deaths	4872	206	0	9015	3711	26298	5435	5995
Recovered	202137.0	7165.0	NaN	76934.0	199247.0	NaN	18102.0	9800.0
Active	158738.0	39648.0	NaN	444318.0	26484.0	NaN	314186.0	169913.0
FIPS	1.0	2.0	60.0	4.0	5.0	6.0	8.0	9.0
Incident_Rate	7459.275895	6427.355802	0.0	7285.171274	7602.945718	6164.469663	5854.774381	5208.781229
Total_Test_Results	NaN	1275750.0	2140.0	5155330.0	2051488.0	33058311.0	4444206.0	4320693.0

21 rows x 61614 columns

```
In [6]: df_daily.shape
```

```
Out[6]: (61614, 21)
```

```
In [7]: df_daily.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 61614 entries, 0 to 61613
Data columns (total 21 columns):
#   Column              Non-Null Count  Dtype  
---  --
0   Province_State      61614 non-null  object 
1   Country_Region      61614 non-null  object 
2   Last_Update         61595 non-null  object 
3   Lat                 59472 non-null  float64
4   Long_              59472 non-null  float64
5   Confirmed           61614 non-null  int64  
6   Deaths             61614 non-null  int64  
7   Recovered           15122 non-null  float64
8   Active              15122 non-null  float64
9   FIPS                61595 non-null  float64
10  Incident_Rate       59472 non-null  float64
11  Total_Test_Results  36637 non-null  float64
12  People_Hospitalized 5129 non-null   float64
13  Case_Fatality_Ratio 49027 non-null  float64
14  UID                 61614 non-null  float64
15  ISO3                61614 non-null  object 
16  Testing_Rate        45921 non-null  float64
17  Hospitalization_Rate 5129 non-null   float64
18  Date                51754 non-null  object 
19  People_Testes       11816 non-null  float64
20  Mortality_Rate      12027 non-null  float64
dtypes: float64(14), int64(2), object(5)
memory usage: 9.9+ MB
```

Transformamos los campos Date a tipo fecha

```
In [8]: df_daily['Date'] = pd.to_datetime(df_daily['Date']).dt.strftime('%d.%m.%Y')
```

Corroboramos la conversión de los campos

```
In [9]: df_daily.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 61614 entries, 0 to 61613
Data columns (total 21 columns):
#   Column              Non-Null Count  Dtype  
---  --
0   Province_State      61614 non-null  object 
1   Country_Region      61614 non-null  object 
2   Last_Update         61595 non-null  object 
3   Lat                 59472 non-null  float64
4   Long_              59472 non-null  float64
5   Confirmed           61614 non-null  int64  
6   Deaths             61614 non-null  int64  
7   Recovered           15122 non-null  float64
8   Active              15122 non-null  float64
9   FIPS                61595 non-null  float64
10  Incident_Rate       59472 non-null  float64
11  Total_Test_Results  36637 non-null  float64
12  People_Hospitalized 5129 non-null   float64
13  Case_Fatality_Ratio 49027 non-null  float64
14  UID                 61614 non-null  float64
15  ISO3                61614 non-null  object 
16  Testing_Rate        45921 non-null  float64
17  Hospitalization_Rate 5129 non-null   float64
18  Date                51754 non-null  object 
19  People_Testes       11816 non-null  float64
20  Mortality_Rate      12027 non-null  float64
dtypes: float64(14), int64(2), object(5)
memory usage: 9.9+ MB
```

Contamos las filas duplicadas si existen

```
In [10]: df_daily.duplicated().sum()
```

Out[10]: 54

Eliminamos las filas duplicadas

```
In [11]: df_daily = df_daily.drop_duplicates()
```

Contamos los valores nulos si es que existen

```
In [12]: df_daily.isnull().values.sum()
```

Out[12]: 374428

Exploramos en que variables existen nulos y que cantidad

```
In [13]: df_daily.isnull().sum()
```

```
Out[13]: Province_State      0
Country_Region              0
Last_Update                 19
Lat                         2140
Long_                       2140
Confirmed                   0
Deaths                     46438
Recovered                   46438
Active                     19
FIPS                        2140
Incident_Rate               2140
Total_Test_Results          24923
```

```
In [14]: df_daily.describe().T
```

```
Out[14]:
```

	count	mean	std	min	25%	50%	75%	max
Lat	59420.0	3.684030e+01	1.078922e+01	-1.427100e+01	3.484050e+01	3.906185e+01	4.236165e+01	6.137070e+01
Long_	59420.0	-8.520742e+01	4.930721e+01	-1.701322e+02	-1.011658e+02	-8.794420e+01	-7.702680e+01	1.456739e+02
Confirmed	61560.0	8.727221e+05	1.456052e+06	0.000000e+00	6.49025e+04	3.395635e+05	1.021065e+06	1.212970e+07
Deaths	61560.0	1.157355e+04	1.714695e+04	0.000000e+00	1.102000e+03	4.892000e+03	1.469050e+04	1.011590e+05
Recovered	15122.0	8.578908e+04	1.933303e+05	0.000000e+00	3.872000e+03	1.754850e+04	8.425575e+04	2.470308e+06
Active	15122.0	5.936916e+04	1.190129e+05	0.000000e+00	3.166000e+03	1.475250e+04	6.111500e+04	1.408516e+06
FIPS	61541.0	3.284088e+03	1.724951e+04	1.000000e+00	1.800000e+01	3.300000e+01	4.800000e+01	9.999900e+04
Incident_Rate	59420.0	1.489822e+04	1.130987e+04	0.000000e+00	4.116387e+03	1.231563e+04	2.547638e+04	5.892799e+04
Total_Test_Results	36637.0	1.144235e+07	2.018320e+07	1.768000e+03	1.771434e+06	4.784927e+06	1.296786e+07	1.844461e+08
People_Hospitalized	5129.0	6.151227e+03	1.379109e+04	2.000000e+00	5.580000e+02	2.014000e+03	6.001000e+03	8.999500e+04
Case_Fatality_Ratio	48973.0	1.330295e+00	5.956818e-01	0.000000e+00	1.007556e+00	1.287507e+00	1.391811e+00	8.03612e+00
UID	61560.0	7.676455e+07	2.357348e+07	1.600000e+01	8.400001e+07	8.400003e+07	8.400004e+07	8.410000e+07
Testing_Rate	45921.0	1.161119e+05	1.307875e+05	5.973514e-08	7.422100e+03	7.692252e+04	1.787269e+05	7.901759e+05

```
In [15]: df_daily.sort_values(by=['Province_State','Date'], na_position='first').T
```

```
Out[15]:
```

	116	290	464	638	812	986	1160	1334
Province_State	Alabama	Alabama	Alabama	Alabama	Alabama	Alabama	Alabama	Alabama
Country_Region	US	US	US	US	US	US	US	US
Last_Update	2023-01-02 04:31:21	2023-01-03 04:31:34	2023-01-04 04:31:12	2023-01-05 04:31:16	2023-01-06 04:31:58	2023-01-07 04:31:23	2023-01-08 04:32:04	2023-01-09 04:31:38
Lat	32.3182	32.3182	32.3182	32.3182	32.3182	32.3182	32.3182	32.3182
Long_	-86.9023	-86.9023	-86.9023	-86.9023	-86.9023	-86.9023	-86.9023	-86.9023
Confirmed	1568934	1568934	1568934	1587224	1587224	1587224	1587224	1587224
Deaths	20737	20737	20737	20776	20776	20776	20776	20776
Recovered	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Active	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
FIPS	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
Incident_Rate	31998.262354	31998.262354	31998.262354	32371.285195	32371.285195	32371.285195	32371.285195	32371.285195
Total_Test_Results	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
People_Hospitalized	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Case_Fatality_Ratio	1.321725	1.321725	1.321725	1.308952	1.308952	1.308952	1.308952	1.308952

```
In [16]: df_daily['Date'].isnull().sum()
```

```
Out[16]: 9886
```

Nuevo dataframe daily\_act con las columnas necesarias solo

```
In [17]: df_daily_act = df_daily[['Province_State', 'Confirmed', 'Deaths', 'Recovered', 'Active', 'ISO3', 'Date']]
```

```
In [18]: df_daily_act
```

```
Out[18]:
```

	Province_State	Confirmed	Deaths	Recovered	Active	ISO3	Date
0	Alabama	365747	4872	202137.0	158738.0	USA	01.01.2021
1	Alaska	47019	206	7165.0	39648.0	USA	01.01.2021
2	American Samoa	0	0	NaN	NaN	ASM	01.01.2021
3	Arizona	530267	9015	76934.0	444318.0	USA	01.01.2021
4	Arkansas	229442	3711	199247.0	26484.0	USA	01.01.2021
...	...	...	...	...	...	...	...
61609	Virginia	2199302	22670	NaN	NaN	USA	NaN
61610	Washington	1883676	15038	NaN	NaN	USA	NaN

61560 rows x 7 columns

```
In [19]: df_daily_act.columns = ['PROVINCIA_ESTADO', 'CASOS_CONFIRMADOS', 'DEFUNCIONES', 'CASOS_RECUPERADOS', 'CASOS_ACTIVOS', 'PAIS_ISO3',
```

```
In [20]: df_daily_act
```

```
Out[20]:
```

	PROVINCIA_ESTADO	CASOS_CONFIRMADOS	DEFUNCIONES	CASOS_RECUPERADOS	CASOS_ACTIVOS	PAIS_ISO3	FECHA
0	Alabama	365747	4872	202137.0	158738.0	USA	01.01.2021
1	Alaska	47019	206	7165.0	39648.0	USA	01.01.2021
2	American Samoa	0	0	NaN	NaN	ASM	01.01.2021
3	Arizona	530267	9015	76934.0	444318.0	USA	01.01.2021
4	Arkansas	229442	3711	199247.0	26484.0	USA	01.01.2021
...	...	...	...	...	...	...	...
61609	Virginia	2199302	22670	NaN	NaN	USA	NaN
61610	Washington	1883676	15038	NaN	NaN	USA	NaN
61611	West Virginia	624721	7672	NaN	NaN	USA	NaN
61612	Wisconsin	1960878	15802	NaN	NaN	USA	NaN
61613	Wyoming	182847	1958	NaN	NaN	USA	NaN

61560 rows x 7 columns

```
In [21]: df_daily_act.to_csv("/content/TFM/Ficheros_Depurados/df_daily_report_us_final.csv", index = False)
```

## BASE DE DATOS Y PROCEDIMIENTO DE CARGA

Creamos una BBDD MySQL donde almacenaremos todos los ficheros depurados, normalizados y transformados. El Script de la BBDD es el siguiente y podemos observar los campos que contiene cada tabla, así como la estructura y tipología de datos de la misma.

### Script para creación de BBDD

```
CREATE SCHEMA PROVA_TFM;
```

```
CREATE TABLE PROVA_TFM.PAISES (  
PAIS_ISO3 VARCHAR(3) NOT NULL,  
PAIS_ISO2 VARCHAR(2) NOT NULL ,  
PAIS_NOM VARCHAR(100) NOT NULL,  
COD_CONTINENTE VARCHAR(2) NOT NULL,  
CONTINENTE VARCHAR(100) NOT NULL,  
OMS_REGION VARCHAR(5) NOT NULL,  
DESC_OMS_REGION VARCHAR(100) NOT NULL,  
PAIS_NOM_2 VARCHAR(100) NOT NULL,  
PRIMARY KEY (PAIS_ISO3));
```

```
CREATE TABLE PROVA_TFM.VACUNACIONES (  
PAIS VARCHAR(50) NOT NULL,  
PAIS_ISO3 VARCHAR(3) NOT NULL,  
CONTINENTE VARCHAR(50) NOT NULL,  
ISO_CONTINENTE VARCHAR(2) NOT NULL,  
OMS_REGION VARCHAR(50) NOT NULL ,  
FUENTE_DATOS VARCHAR(50) NULL ,  
FECHA_ULT_ACTUALIZACION VARCHAR(10) NULL ,  
TOTAL_VACUNACION_ACUM DOUBLE NULL,  
NPER_VACUNADAS_1DOSIS INT NULL,  
TOTAL_VACUNACION_PER100 DOUBLE NULL,  
NPER_VACUNADAS_1DOSIS_PER100 DOUBLE NULL,
```



```

NPER_VACUNADAS_DOSIS_FULL INT NULL,
NPER_VACUNADAS_DOSIS_FULL_PER100 DOUBLE NULL,
FECHA_PRIMERA_VACUNA VARCHAR(10) NULL,
N_TIPOS_VACUNAS_USADAS DOUBLE NULL,
NPER_CON_DOSIS_ADICIONAL DOUBLE NULL,
NPER_CON_DOSIS_ADIDICIONAL_PER100 DOUBLE NULL,
PRIMARY KEY (PAIS_ISO3));

```

```

CREATE TABLE PROVA_TFM.VACUNAS_TIPOS (
PAIS_ISO3 VARCHAR(3) NOT NULL,
CONTINENTE VARCHAR(50) NOT NULL,
ISO_CONTINENTE VARCHAR(2) NOT NULL,
NOMBRE_VACUNA VARCHAR(100) NULL ,
NOMBRE_TIPO_VACUNA VARCHAR(90) NULL ,
NOMBRE_COMPAÑIA VARCHAR(50) NULL ,
FECHA_AUTORIZACION VARCHAR(10) NULL,
FECHA_INICIO_VACUNACION VARCHAR(10) NULL,
FUENTE_DATOS VARCHAR(50) NULL);

```

```

CREATE TABLE PROVA_TFM.CASOS_CONFIRMADOS_DEFUNCIONES_US (
PROVINCIA_ESTADO VARCHAR(50) NOT NULL,
CASOS_CONFIRMADOS INT NULL,
DEFUNCIONES INT NULL,
CASOS_RECUPERADOS INT NULL,
CASOS_ACTIVOS INT NULL,
PAIS_ISO3 VARCHAR(3) NOT NULL,
FECHA VARCHAR(10) NULL);

```

```

CREATE TABLE PROVA_TFM.CASOS_HOSPITALIZADOS_UCI_EU (

```

```
PAIS_ISO3 VARCHAR(3) NOT NULL,  
PAIS_NOM VARCHAR(100) NOT NULL,  
INDICADOR VARCHAR(100) NULL,  
FECHA VARCHAR(10) NULL,  
ANY_SEMANA VARCHAR(10) NOT NULL,  
VALOR DOUBLE NULL,  
FUENTE_ORIGEN VARCHAR(100) NULL);
```

```
CREATE TABLE PROVA_TFM.TESTING_COVID_EU (  
PAIS_ISO3 VARCHAR(3) NOT NULL,  
ANY_SEMANA VARCHAR(10) NOT NULL,  
NIVEL VARCHAR(50) NOT NULL,  
CASOS_NUEVOS INT NULL,  
N_TESTS_REALIZADOS INT NULL,  
POBLACION INT NULL,  
RATIO_TESTS DOUBLE NULL,  
RATIO_POSITIVO DOUBLE NULL,  
FUENTES_TESTS VARCHAR(50) NULL);
```

```
CREATE TABLE PROVA_TFM.COVID_DAILY (  
PAIS VARCHAR(50) NOT NULL,  
PAIS_ISO3 VARCHAR(3) NOT NULL,  
PAIS_ISO2 VARCHAR(2) NOT NULL,  
FECHA_INFORMADA VARCHAR(10) NOT NULL,  
OMS_REGION VARCHAR(50) NOT NULL ,  
CASOS_NUEVOS INT NULL,  
CASOS_ACUM INT NULL,  
MUERTES_NUEVAS INT NULL,  
MUERTES_ACUM INT NULL );
```

```

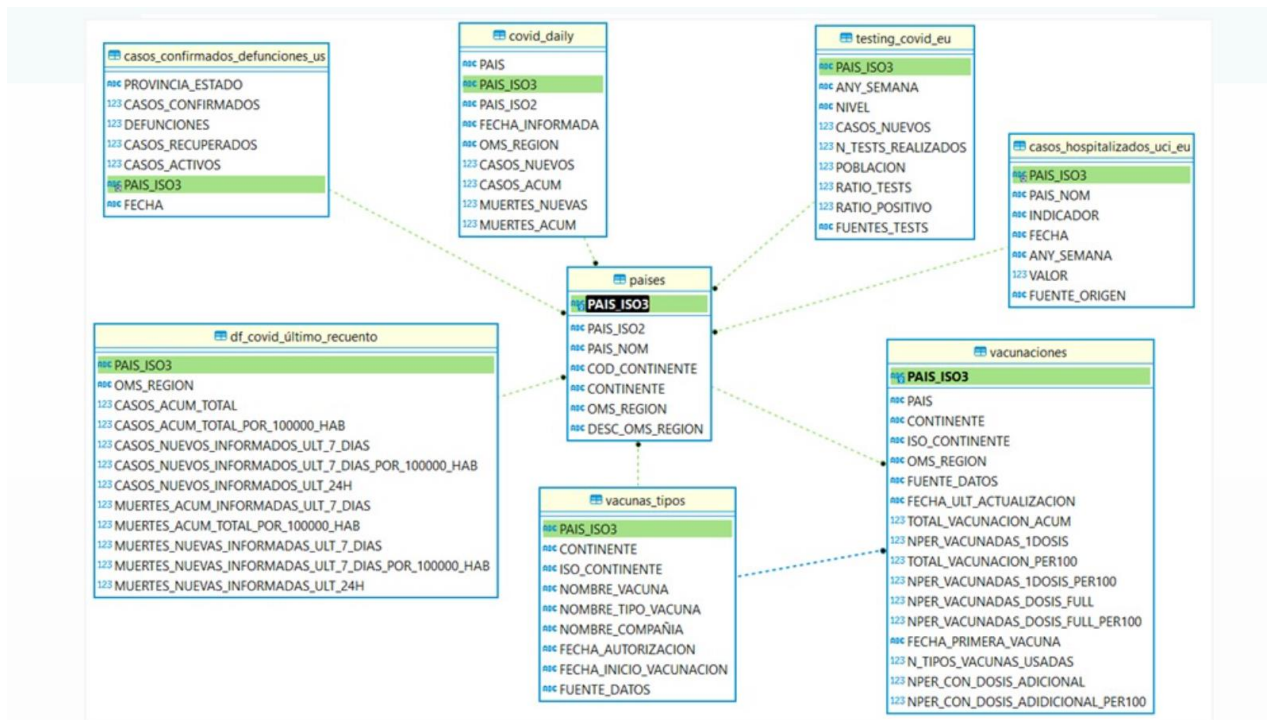
CREATE TABLE PROVA_TFM.CASOS_COVID_ULTIMO_RECUESTO (
PAIS_ISO3 VARCHAR(3) NOT NULL,
OMS_REGION VARCHAR(50) NOT NULL ,
CASOS_ACUM_TOTAL INT NULL,
CASOS_ACUM_TOTAL_POR_100000_HAB DOUBLE NULL,
CASOS_NUEVOS_INFORMADOS_ULT_7_DIAS INT NULL,
CASOS_NUEVOS_INFORMADOS_ULT_7_DIAS_POR_100000_HAB DOUBLE NULL,
CASOS_NUEVOS_INFORMADOS_ULT_24H INT NULL,
MUERTES_ACUM_INFORMADAS_ULT_7_DIAS INT NULL,
MUERTES_ACUM_TOTAL_POR_100000_HAB DOUBLE NULL,
MUERTES_NUEVAS_INFORMADAS_ULT_7_DIAS INT NULL,
MUERTES_NUEVAS_INFORMADAS_ULT_7_DIAS_POR_100000_HAB DOUBLE NULL,
MUERTES_NUEVAS_INFORMADAS_ULT_24H INT NULL );

```

### Modelo de datos de la BBDD – TFM COVID-19

Hemos construido una BBDD en Modo estrella con la tabla “**Países**” como tabla principal o maestra en la que se pueden observar las diferentes relaciones con el resto de tablas de dimensiones.

El Campo **PAIS\_ISO3** es la clave principal de la Tabla de Países y se relaciona con el resto de tablas de dimensiones a través del mismo campo **PAIS\_ISO3** que pasa a ser la clave secundaria.



## Notebook Jupyter de Conexión y Carga de Ficheros a la BBDD – TFM COVID-19

En el siguiente notebook se realiza la conexión a la BBDD, interacción con las tablas que forman la BBDD, así como la carga de ficheros depurados y actualizados.

Dicho notebook se encuentra en el repositorio de GitHub // TFM/Notebooks

[Carga ficheros a BBDD.ipynb](#)

# Conexión BBDD y Carga de Ficheros - TFM COVID -19

Instalar librerías de **MySQL** para establecer la Conexión con la **BBDD**

```
In [1]: import pandas as pd
import numpy as np

#!pip install mysql-connector-python
#!pip install sqlalchemy
#!pip install PyMySQL
#!pip install ipython-sql
```

Instalar la librería **xldr** para importar **ficheros xls**

```
In [2]: !pip install xldr
```

Establezco conexión con la **BBDD SQL** con Pandas

```
In [3]: import mysql.connector
```

```
In [36]: try:
    connection=mysql.connector.connect(
        host='localhost',
        port= 3306,
        user= 'root',
        password='Juanky_123',
        db= 'prova_tfm'
    )
    if connection.is_connected():
        print('Conexión establecida correctamente')
        info_server=connection.get_server_info()
        print(info_server)

except Exception as ex:
    print(ex)
finally:
    if connection.is_connected():
        connection.close()
        print('Conexión finalizada')
```

```
Conexión establecida correctamente
8.0.31
Conexión finalizada
```

Cargo el **Pluggin para sql**

```
In [5]: %load_ext sql
```

Conexión a **MySQL BBDD local** mediante sql

```
In [6]: %sql mysql+pymysql://root:Juanky_123@localhost/prova_tfm
```

```
In [7]: %sql show tables
```

```
* mysql+pymysql://root:***@localhost/prova_tfm
9 rows affected.
```

```
Out[7]: Tables in prova_tfm
casos_confirmados_defunciones_us
casos_hospitalizados_uci_eu
ciudades
covid_daily
df_covid_ultimo_reuento
países
testing_covid_eu
vacunaciones
vacunas_tipos
```

# Interacción con la BBDD

Instrucciones SQL para describir, mostrar, cargar ficheros a la **BBDD**

Verificar estructura de la Tabla Maestra **países**

In [26]: `%sql DESC países`

`* mysql+pymysql://root:***@localhost/prova_tfm`  
`8 rows affected.`

Out[26]:

	Field	Type	Null	Key	Default	Extra
	PAIS_ISO3	varchar(3)	NO	PRI	None	
	PAIS_ISO2	varchar(2)	NO		None	
	PAIS_NOM	varchar(100)	NO		None	
	COD_CONTINENTE	varchar(2)	NO		None	
	CONTINENTE	varchar(100)	NO		None	
	OMS_REGION	varchar(5)	NO		None	
	DESC_OMS_REGION	varchar(100)	NO		None	
	PAIS_NOM_2	varchar(100)	NO		None	

Acceso a la **tabla países de la BBDD** e importación al **df\_paises**

In [39]: `from sqlalchemy import create_engine, text`  
`engine = create_engine("mysql+pymysql://root:Juanky_123@localhost/prova_tfm")`  
`query = 'SELECT * FROM países'`  
`with engine.begin() as conn:`  
`df_paises = pd.read_sql_query(sql=text(query), con=conn)`  
`df_paises.head()`

Out[39]:

	PAIS_ISO3	PAIS_ISO2	PAIS_NOM	COD_CONTINENTE	CONTINENTE	OMS_REGION	DESC_OMS_REGION	PAIS_NOM_2
0	ABW	AW	Aruba	NA	North America	AMRO	América	Aruba
1	AFG	AF	Afghanistan	AS	Asia	EMRO	Mediterráneo Oriental	Afghanistan
2	AGO	AO	Angola	AF	Africa	AFRO	Africa	Angola
3	AIA	AI	Anguilla	NA	North America	AMRO	América	Anguilla
4	ALB	AL	Albania	EU	Europe	EURO	Europa	Albania

## Borrado y Carga de ficheros depurados a la BBDD local

1.- Carga del **df\_daily\_report\_us\_final.csv** depurado y actualizado

In [19]: `df_daily_us = pd.read_csv('df_daily_report_us_final.csv')`  
`df_daily_us.head()`

Out[19]:

	PROVINCIA_ESTADO	CASOS_CONFIRMADOS	DEFUNCIONES	CASOS_RECUPERADOS	CASOS_ACTIVOS	PAIS_ISO3	FECHA
0	Alabama	365747	4872	202137.0	158738.0	USA	01.01.2021
1	Alaska	47019	206	7165.0	39648.0	USA	01.01.2021
2	American Samoa	0	0	NaN	NaN	ASM	01.01.2021
3	Arizona	530267	9015	76934.0	444318.0	USA	01.01.2021
4	Arkansas	229442	3711	199247.0	26484.0	USA	01.01.2021

Borrado de la Tabla **casos\_confirmados\_defunciones\_us** y carga del dataframe depurado **df\_daily\_us** a la BBDD

In [20]: `%sql delete from casos_confirmados_defunciones_us`  
`* mysql+pymysql://root:***@localhost/prova_tfm`  
`61560 rows affected.`

Out[20]: `[]`

In [51]: `df_daily_us.to_sql('casos_confirmados_defunciones_us', con = conn, if_exists = 'append', index=False)`

Out[51]: `61560`

## CONCLUSIONES

En el presente entregable de nuestro Trabajo de Fin de Máster (TFM) se ha llevado a cabo la depuración, transformación y carga de diversas fuentes de datos relacionadas con el COVID-19, como los datos de pruebas realizadas por semana y país en Europa, informes diarios de casos en los Estados Unidos, datos de hospitalización y ocupación de unidades de cuidados intensivos, metadatos y datos de vacunación, entre otros.

La limpieza y transformación de los datos se ha realizado con el objetivo de obtener una base de datos consistente y lista para su análisis posterior. Se ha utilizado el lenguaje SQL para montar y gestionar las bases de datos, así como los dataframes generados.

En cuanto a las fuentes de información utilizadas, se ha trabajado con datos proporcionados por la Organización Mundial de la Salud (OMS), informes diarios de casos y muertes, datos de vacunación, así como datos de pruebas de COVID-19 por semana y país. También se han utilizado datos específicos de los Estados Unidos, como los informes diarios de casos por estado y datos de hospitalización, asimismo, datos sobre las pruebas de Covid-19 en el continente europeo.

Durante el proceso de depuración, transformación y enriquecimiento de los datos, se ha llevado a cabo la integración de diferentes fuentes para obtener una visión más completa de la situación de la pandemia. Además, se han desarrollado tablas adicionales, como la tabla de países con información relevante, en los anexos encontrarán también información sobre nuevas tablas constituidas como son de indicadores de desarrollo del banco mundial seleccionados por nuestro equipo y la tabla de datos climáticos globales.

En cuanto a las hipótesis formuladas para la próxima entrega, estas representan suposiciones basadas en las relaciones esperadas entre variables relevantes y la incidencia, propagación y consecuencias del COVID-19. Las hipótesis nulas, por otro lado, representan la ausencia de dichas relaciones. En el próximo proceso de análisis de datos, pruebas estadísticas, se evaluará la evidencia en apoyo o en contra de estas hipótesis, lo que contribuirá a la comprensión y conocimiento sobre la pandemia de COVID-19, así mismo estos datos se llevarán a dashboards elaborados en Power BI que serán acompañados de su respectivo storytelling.

Es importante destacar que el análisis de las hipótesis requerirá un enfoque riguroso, utilizando métodos adecuados de análisis estadístico y teniendo en cuenta otros factores que puedan influir en los resultados. Además, se respaldarán las afirmaciones con bibliografía académica especializada para garantizar la validez y la robustez de los resultados obtenidos.

En resumen, a pesar de las dificultades encontradas durante el proceso, se ha logrado cumplir con las tareas establecidas en esta segunda entrega del TFM. Se ha realizado la depuración, transformación y carga de los datos, y se ha preparado el terreno para el análisis y la respuesta a las hipótesis planteadas. El uso de Python se ha destacado como herramientas eficaces para llevar a cabo estos procesos. A medida que avancemos en la investigación, se espera obtener resultados que contribuyan al entendimiento de la pandemia de COVID-19.

## ANEXOS

En cuanto a tema de análisis de datos hemos preparado dos tablas adicionales a la tabla países para poder realizar integraciones: WDI y temp.

Entonces, además de las tablas de covid tenemos

- países, que contiene información sobre cada país, región OMS, continente, ISO2, ISO3, etc.
- WID, que contiene información de 2018 a 2022 de algunos indicadores de desarrollo seleccionados por el equipo
- temp, que contiene datos sobre el clima global en 2013, si bien sería mejor tener datos más cercanos ha sido imposible encontrar un dataset similar con estas características puesto que muchos suelen tener registros por segundos (lo que nos resulta imposible de manejar por cuestiones informáticas) o por estaciones de control climático sin divisiones territoriales

### Preparación del entorno de trabajo

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import IPython as display
import datetime as dt
```

```
# configurar los gráficos
sns.set_style('whitegrid')
sns.set_palette('mako')
```

### Tabla países

Debido a que hay código como NN y NA siempre hay que revisar los NaN o Z9

```
print(países.loc[países['COD_CONTINENTE'] == 'Z9', 'CONTINENTE'])
```

Se agrega el código faltante

```
países.loc[(países['COD_CONTINENTE'] == 'Z9') & (países['CONTINENTE'] == 'North America'),
           'COD_CONTINENTE'] = 'NA'
países.loc[(países['COD_CONTINENTE'] == 'Z9') & (países['CONTINENTE'] == 'Europe')]
```

Y agregamos las coordenadas

```
coord = pd.read_csv("C:/Users/Patricia/Desktop/Github/TFM/DF
Covid/world_country_and_usa_states_latitu
# Realiza la fusión utilizando el código ISO2 como clave
países = países.merge(coord[['country_code', 'longitude', 'latitude']],
                      left_on='PAIS_ISO2', right_on='country_code', how='left')
países.loc[pd.isna(países['longitude'])]
# Define las coordenadas de longitud y latitud correspondientes
longitudes = [17.0, -177.0, -68.98, np.nan, -62.83, -63.06, np.nan]
latitudes = [-22.0, 0.0, 12.17, np.nan, 17.9, 18.07, np.nan]

# Itera sobre los países faltantes y actualiza los valores correspondientes en la tabla
for i, pais in enumerate(['Namibia', 'United States Minor Outlying Islands', 'Curaçao', 'Sudán
del Sur', 'San Bartolomé', 'San Martín', 'San Martín']):
    index = países[países['PAIS_NOM'] == pais].index[0]
    países.at[index, 'longitude'] = longitudes[i]
    países.at[index, 'latitude'] = latitudes[i]
países.sort_values('PAIS_NOM')
```



F	Afghanistan	AS	Asia	EMRO	Mediterráneo Oriental	Afghanist
L	Albania	EU	Europe	EURO	Europa	Alba
Z	Algeria	AF	Africa	AFRO	Africa	Alge
S	American Samoa	OC	Oceania	WPRO	Pacífico Occidental	Americ Sam
O	Andorra	EU	Europe	EURO	Europa	Ando
...	...	...	...	...	...	...
F	Wallis and Futuna	OC	Oceania	WPRO	Pacífico Occidental	Wallis a Futu
d	Western Sahara	AF	Africa	Z999	Sin determinar	West Sah.
E	Yemen	AS	Asia	EMRO	Mediterráneo Oriental	Yem
l	Zambia	AF	Africa	AFRO	Africa	Zam
v	Zimbabwe	AF	Africa	AFRO	Africa	Zimbab

## Indicadores de Desarrollo

```
WDI = pd.read_csv("C:/Users/Patricia/Desktop/Github/TFM/DF Covid/WDIData.csv")
```

Nos quedamos únicamente con los años que nos interesa revisar

```
WDI = WDI[['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code', '2018', '2019',
```

Como vemos que hay regiones también procedemos a eliminarlas, para esto revisamos en la tabla países y todos aquellos registro que el WDI['Country Code'] que no estén en países['PAIS\_ISO3'] los eliminamos

```
WDI = WDI[WDI['Country Code'].isin(países['PAIS_ISO3'].unique())]
```

```
WDI['Indicator Code'].unique()
```

```
indicators = WDI['Indicator Name'].unique()
```

```
'''
    'Access to clean fuels and technologies for cooking, rural (% of rural populat
ion)',
    'Access to clean fuels and technologies for cooking, urban (% of urban populat
ion)',
    ...,
    'Women who were first married by age 18 (% of women ages 20-24)',
    "Women's share of population ages 15+ living with HIV (%)",
    'Young people (ages 15-24) newly infected with HIV'], dtype=object)
```

```
np.savetxt(f"C:/Users/Patricia/Desktop/Github/TFM/DF Covid/WDI_indicators.txt", indicators
```

Después de exportarlos a txt y realizar una lectura cuidadosa declaramos los que queremos

mantener, a continuación, una explicación de cada indicador cortesía de ChatGPT pero revisado manualmente

- Access to clean fuels and technologies for cooking (% of population) : Este indicador muestra el porcentaje de la población que tiene acceso a combustibles y tecnologías limpias para cocinar, como gas natural, electricidad o cocinas eficientes. Proporciona información sobre el nivel de acceso a fuentes de energía seguras y menos contaminantes para las actividades culinarias.
- Access to electricity (% of population) : Este indicador muestra el porcentaje de la población que tiene acceso a la electricidad. Sirve para evaluar el nivel de electrificación de un país y su capacidad para brindar servicios básicos a la población.
- Adequacy of social insurance programs (% of total welfare of beneficiary households) Este indicador representa el porcentaje del bienestar total de los hogares beneficiarios que es

cubierto por los programas de seguro social. Mide la efectividad y la cobertura de los programas de seguro social en la protección del bienestar de los hogares.

- Adequacy of social safety net programs (% of total welfare of beneficiary households): Este indicador muestra el porcentaje del bienestar total de los hogares beneficiarios que es cubierto por los programas de redes de seguridad social. Evalúa la efectividad y la cobertura de los programas de protección social en la prevención y mitigación de la pobreza.
- Adjusted net national income (annual % growth) : Este indicador representa el crecimiento anual porcentual del ingreso nacional neto ajustado. El ingreso nacional neto ajustado tiene en cuenta los factores externos, como el agotamiento de recursos naturales y las emisiones de carbono, para proporcionar una medida más precisa del crecimiento económico sostenible.
- Adjusted net national income per capita (annual % growth) : Este indicador muestra el crecimiento anual porcentual del ingreso nacional neto ajustado per cápita. Proporciona información sobre el crecimiento económico ajustado a la población y puede indicar cambios en el nivel de vida de los ciudadanos.
- Air transport, passengers carried : Este indicador muestra la cantidad de pasajeros transportados por vía aérea en un determinado período de tiempo. Sirve como una medida del volumen y la importancia del transporte aéreo en un país o región.
- Annual freshwater withdrawals, domestic (% of total freshwater withdrawal) : Este indicador muestra el porcentaje del total de extracciones de agua dulce que se utiliza para uso doméstico en un año determinado. Ayuda a evaluar la disponibilidad y la gestión del agua potable para uso residencial en relación con el uso total de agua.
- Bank capital to assets ratio (%) : Este indicador muestra el porcentaje de los activos totales de un banco que está respaldado por capital. Proporciona una medida de la solidez financiera de un banco y su capacidad para absorber pérdidas.
- Bank liquid reserves to bank assets ratio (%) : Este indicador muestra el porcentaje de los activos totales de un banco que se mantiene como reservas líquidas, como efectivo o depósitos en bancos centrales. Sirve como indicador de la capacidad del banco para hacer frente a retiros de fondos y afrontar situaciones de estrés financiero.
- Birth rate, crude (per 1,000 people) : Este indicador muestra el número promedio de nacimientos por cada 1,000 personas en un año determinado. Proporciona información sobre la tasa de natalidad en una población y puede indicar tendencias demográficas y cambios en la estructura poblacional.
- Births attended by skilled health staff (% of total) : Este indicador muestra el porcentaje de nacimientos en los que el parto es atendido por personal de salud capacitado. Sirve como indicador de la calidad de la atención médica durante el parto y puede reflejar el acceso a servicios de salud materna adecuados.
- Cause of death, by communicable diseases and maternal, prenatal and nutrition conditions (% of total) : Este indicador muestra el porcentaje de muertes causadas por enfermedades transmisibles y condiciones relacionadas con la salud materna, prenatal y nutrición. Ayuda a evaluar la carga de enfermedades prevenibles y las condiciones de salud en una población.
- Cause of death, by injury (% of total) : Este indicador muestra el porcentaje de muertes causadas por lesiones. Proporciona información sobre la incidencia y la gravedad de diferentes tipos de lesiones y puede ayudar a orientar los esfuerzos de prevención y respuesta.
- Cause of death, by non-communicable diseases (% of total) : Este indicador muestra el porcentaje de muertes causadas por enfermedades no transmisibles, como enfermedades

- cardiovasculares, cáncer, diabetes y enfermedades respiratorias crónicas. Ayuda a evaluar la carga de enfermedades crónicas y la efectividad de las estrategias de prevención y control.
- Community health workers (per 1,000 people) : Este indicador muestra la cantidad de trabajadores de salud comunitarios por cada 1,000 personas. Los trabajadores de salud comunitarios son profesionales de la salud capacitados que brindan atención básica y promueven la salud en comunidades locales.
  - Control of Corruption: Estimate : Este indicador representa una estimación del control de la corrupción en un país. Evalúa la percepción y la efectividad de las medidas anticorrupción, así como la transparencia y la integridad en la administración pública.
  - Control of Corruption: Percentile Rank : Este indicador muestra el rango percentil en el que se encuentra un país en términos de control de la corrupción. Proporciona una comparación relativa de la situación de un país en relación con otros países en cuanto a la lucha contra la corrupción.
  - Coverage of social insurance programs (% of population) : Este indicador muestra el porcentaje de la población que está cubierta por programas de seguro social. Ayuda a evaluar la efectividad y la amplitud de los programas de protección social en un país.
  - Coverage of social protection and labor programs (% of population) : Este indicador muestra el porcentaje de la población que está cubierta por programas de protección social y laboral. Incluye medidas como la seguridad social, el desempleo, la capacitación laboral y otros programas destinados a proteger y apoyar a los trabajadores y sus familias.
  - Coverage of social safety net programs (% of population) : Este indicador muestra el porcentaje de la población que está cubierta por programas de redes de seguridad social. Los programas de redes de seguridad social proporcionan apoyo y asistencia a las personas y familias en situaciones de vulnerabilidad económica o social.
  - Current education expenditure, total (% of total expenditure in public institutions) : Este indicador muestra el porcentaje del gasto total en educación en relación con el gasto total en instituciones públicas. Proporciona información sobre la inversión en educación en relación con otros sectores y puede indicar el compromiso y la prioridad de un país con la educación.
  - Current health expenditure (% of GDP) : Este indicador muestra el porcentaje del Producto Interno Bruto (PIB) de un país que se destina al gasto en salud. Ayuda a evaluar la inversión en salud en relación con el tamaño de la economía y puede reflejar el nivel de compromiso de un país con la salud de su población.
  - Current health expenditure per capita, PPP (current international dls) : Este indicador muestra el gasto en salud por persona en dólares internacionales de paridad de poder adquisitivo (PPP). Proporciona una medida del gasto promedio en salud ajustado a las diferencias de poder adquisitivo entre países.
  - Death rate, crude (per 1,000 people) : Este indicador muestra la tasa de mortalidad general por cada 1,000 personas en un año determinado. Proporciona información sobre el nivel de mortalidad en una población y puede indicar cambios en la salud y la calidad de vida.
  - Cause of death, by communicable diseases and maternal, prenatal and nutrition conditions (% of total) : Este indicador muestra el porcentaje de muertes causadas por enfermedades transmisibles y condiciones relacionadas con la salud materna, prenatal y nutrición. Ayuda a evaluar la carga de enfermedades prevenibles y las condiciones de salud en una población.
  - Cause of death, by injury (% of total) : Este indicador muestra el porcentaje de muertes causadas por lesiones. Proporciona información sobre la incidencia y la gravedad de

diferentes tipos de lesiones y puede ayudar a orientar los esfuerzos de prevención y respuesta.

- Cause of death, by non-communicable diseases (% of total) : Este indicador muestra el porcentaje de muertes causadas por enfermedades no transmisibles, como enfermedades cardiovasculares, cáncer, diabetes y enfermedades respiratorias crónicas. Ayuda a evaluar la carga de enfermedades crónicas y la efectividad de las estrategias de prevención y control.
- Community health workers (per 1,000 people) : Este indicador muestra la cantidad de trabajadores de salud comunitarios por cada 1,000 personas. Los trabajadores de salud comunitarios son profesionales de la salud capacitados que brindan atención básica y promueven la salud en comunidades locales.
- Control of Corruption: Estimate : Este indicador representa una estimación del control de la corrupción en un país. Evalúa la percepción y la efectividad de las medidas anticorrupción, así como la transparencia y la integridad en la administración pública.
- Control of Corruption: Percentile Rank: Este indicador muestra el rango percentil en el que se encuentra un país en términos de control de la corrupción. Proporciona una comparación relativa de la situación de un país en relación con otros países en cuanto a la lucha contra la corrupción.
- Coverage of social insurance programs (% of population) : Este indicador muestra el porcentaje de la población que está cubierta por programas de seguro social. Ayuda a evaluar la efectividad y la amplitud de los programas de protección social en un país.
- Coverage of social protection and labor programs (% of population) : Este indicador muestra el porcentaje de la población que está cubierta por programas de protección social y laboral. Incluye medidas como la seguridad social, el desempleo, la capacitación laboral y otros programas destinados a proteger y apoyar a los trabajadores y sus familias.
- Coverage of social safety net programs (% of population) : Este indicador muestra el porcentaje de la población que está cubierta por programas de redes de seguridad social. Los programas de redes de seguridad social proporcionan apoyo y asistencia a las personas y familias en situaciones de vulnerabilidad económica o social.
- Current education expenditure, total (% of total expenditure in public institutions) : Este indicador muestra el porcentaje del gasto total en educación en relación con el gasto total en instituciones públicas. Proporciona información sobre la inversión en educación en relación con otros sectores y puede indicar el compromiso y la prioridad de un país con la educación.
- Current health expenditure (% of GDP) : Este indicador muestra el porcentaje del Producto Interno Bruto (PIB) de un país que se destina al gasto en salud. Ayuda a evaluar la inversión en salud en relación con el tamaño de la economía y puede reflejar el nivel de compromiso de un país con la salud de su población.
- Current health expenditure per capita, PPP (current international dls) : Este indicador muestra el gasto en salud por persona en dólares internacionales de paridad de poder adquisitivo (PPP). Proporciona una medida del gasto promedio en salud ajustado a las diferencias de poder adquisitivo entre países.
- Death rate, crude (per 1,000 people) : Este indicador muestra la tasa de mortalidad general por cada 1,000 personas en un año determinado. Proporciona información sobre el nivel de mortalidad en una población y puede indicar cambios en la salud y la calidad de vida.
- GNI growth (annual %) : Este indicador muestra el crecimiento anual del Ingreso Nacional Bruto (INB) de un país. El INB es la suma del PIB y los ingresos netos del exterior. El

crecimiento del INB refleja la evolución de la economía y puede indicar cambios en la capacidad productiva y el nivel de ingresos de un país.

- GNI per capita growth (annual %) : Este indicador muestra el crecimiento anual del Ingreso Nacional Bruto (INB) per cápita de un país. El INB per cápita es el INB dividido por la población. El crecimiento del INB per cápita refleja el cambio porcentual en el ingreso promedio de los habitantes de un país.
- Government Effectiveness: Estimate : Este indicador representa una estimación de la eficacia del gobierno de un país. Evalúa la calidad de los servicios y las políticas gubernamentales, incluyendo la capacidad para implementar y hacer cumplir las leyes, la transparencia en la gestión y la eficiencia en la prestación de servicios públicos.
- Gross national expenditure (% of GDP) : Este indicador muestra el porcentaje del gasto nacional bruto en relación con el Producto Interno Bruto (PIB) de un país. El gasto nacional bruto incluye el consumo final, la inversión y las exportaciones netas. Proporciona información sobre la demanda interna y el nivel de actividad económica.
- Human capital index (HCI) (scale 0-1) : Este indicador es un índice que mide el capital humano en un país en una escala de 0 a 1, donde 1 representa el máximo nivel de capital humano. El capital humano se refiere al conocimiento, las habilidades y la salud de la población, y el HCI evalúa el nivel de desarrollo humano en términos de capital humano.
- Immunization, DPT (% of children ages 12-23 months) : Este indicador muestra el porcentaje de niños de 12 a 23 meses que han sido inmunizados contra la difteria, el tétanos y la tos ferina. Proporciona información sobre la cobertura de vacunación y la protección de los niños contra estas enfermedades.
- Immunization, HepB3 (% of one-year-old children) : Este indicador muestra el porcentaje de niños de un año de edad que han sido inmunizados contra la hepatitis B. Proporciona información sobre la cobertura de vacunación y la protección de los niños contra esta enfermedad.
- Immunization, measles (% of children ages 12-23 months) : Este indicador muestra el porcentaje de niños de 12 a 23 meses que han sido inmunizados contra el sarampión. Proporciona información sobre la cobertura de vacunación y la protección de los niños contra esta enfermedad.
- Incidence of HIV, all (per 1,000 uninfected population) : Este indicador muestra la incidencia del VIH en una población, medida como el número de nuevos casos de infección por VIH por cada 1,000 personas no infectadas en un año determinado. Proporciona información sobre la propagación de la infección por VIH y el impacto de la epidemia en la salud de la población.
- Incidence of malaria (per 1,000 population at risk) : Este indicador muestra la incidencia de la malaria en una población, medida como el número de nuevos casos de malaria por cada 1,000 personas en riesgo de contraer la enfermedad en un año determinado. Proporciona información sobre la carga de la malaria y la efectividad de las medidas de prevención y control.
- Incidence of tuberculosis (per 100,000 people) : Este indicador muestra la incidencia de la tuberculosis en una población, medida como el número de nuevos casos de tuberculosis por cada 100,000 personas en un año determinado. Proporciona información sobre la carga de la tuberculosis y la efectividad de los programas de control de la enfermedad.
- Increase in poverty gap at dls 1.90 ( dls 2011 PPP) poverty line due to out-of-pocket health care expenditure (% of poverty line) : Este indicador muestra el incremento en la brecha de pobreza en la línea de pobreza de dls 1.90 (en paridad de poder adquisitivo de 2011) debido

a los gastos de atención médica pagados directamente por las personas. Indica el impacto de los gastos de salud en la situación de pobreza de la población.

- Increase in poverty gap at 3.20 dls (dls 2011 PPP) poverty line due to out-of-pocket health care expenditure (% of poverty line) : Este indicador muestra el incremento en la brecha de pobreza en la línea de pobreza de 3.20 dls (en paridad de poder adquisitivo de 2011) debido a los gastos de atención médica pagados directamente por las personas. Indica el impacto de los gastos de salud en la situación de pobreza de la población.
- Individuals using the Internet (% of population) : Este indicador muestra el porcentaje de la población que utiliza Internet. Proporciona información sobre la penetración de Internet en una población y refleja el acceso a las tecnologías de la información y la comunicación.
- Inflation, consumer prices (annual %) : Este indicador muestra la tasa de inflación anual, medida como el cambio porcentual en los precios de los bienes y servicios de consumo en un año determinado. Indica la variación de los precios en la economía y puede tener impacto en el poder adquisitivo de la población.
- Intentional homicides (per 100,000 people) : Este indicador muestra la tasa de homicidios intencionales por cada 100,000 personas en una población. Proporciona información sobre la seguridad y la violencia en un país y puede reflejar la situación de la delincuencia y el orden público.
- Land area (sq. km): Este indicador muestra el área total de tierra de un país medida en kilómetros cuadrados. Proporciona información sobre el tamaño geográfico de un país y puede ser relevante para evaluar su capacidad de recursos naturales y desarrollo territorial.
- Level of water stress: freshwater withdrawal as a proportion of available freshwater resources: Este indicador muestra el nivel de estrés hídrico en relación con la disponibilidad de recursos de agua dulce. Mide la proporción de agua dulce que se retira con respecto a la cantidad total disponible en una región o país. Proporciona información sobre el uso sostenible del agua y los desafíos relacionados con su escasez.
- Life expectancy at birth, female (years): Este indicador muestra la esperanza de vida al nacer para las mujeres, es decir, la cantidad promedio de años que se espera que viva una mujer al nacer en un determinado país. Proporciona una medida del estado de salud y calidad de vida de las mujeres en una población.
- Life expectancy at birth, male (years): Este indicador muestra la esperanza de vida al nacer para los hombres, es decir, la cantidad promedio de años que se espera que viva un hombre al nacer en un determinado país. Proporciona una medida del estado de salud y calidad de vida de los hombres en una población.
- Life expectancy at birth, total (years): Este indicador muestra la esperanza de vida al nacer para la población en general, es decir, la cantidad promedio de años que se espera que viva una persona al nacer en un determinado país. Proporciona una medida general del estado de salud y calidad de vida en una población.
- Literacy rate, adult total (% of people ages 15 and above): Este indicador muestra el porcentaje de la población adulta de 15 años en adelante que puede leer y escribir. Proporciona información sobre el nivel de alfabetización en una población y puede ser un indicador del nivel educativo y el acceso a oportunidades.
- Mobile cellular subscriptions (per 100 people): Este indicador muestra el número de suscripciones de telefonía móvil por cada 100 personas en un país. Proporciona información sobre la penetración de los servicios de telefonía móvil y la disponibilidad de comunicación en una población.

- Mortality rate attributed to unsafe water, unsafe sanitation and lack of hygiene (per 100,000 population): Este indicador muestra la tasa de mortalidad atribuida al agua insalubre, saneamiento deficiente y falta de higiene. Mide el número de muertes relacionadas con la falta de acceso a agua potable segura, saneamiento adecuado e higiene. Proporciona información sobre los riesgos para la salud asociados a estas condiciones y la necesidad de mejorar la infraestructura y las prácticas sanitarias.
- Multidimensional poverty headcount ratio (% of total population): Este indicador muestra el porcentaje de la población que se encuentra en situación de pobreza multidimensional. La pobreza multidimensional considera varios aspectos, como el acceso a servicios básicos, la educación, la salud y el nivel de vida. Proporciona una medida más completa de la pobreza que solo considerar el ingreso.
- Multidimensional poverty index (scale 0-1): Este indicador muestra el índice de pobreza multidimensional en una escala del 0 al 1, donde 0 representa la ausencia total de pobreza multidimensional y 1 representa la pobreza multidimensional más extrema. El índice combina diferentes dimensiones de la pobreza para proporcionar una medida compuesta de la pobreza en un país.
- Multidimensional poverty intensity (average share of deprivations experienced by the poor): Este indicador muestra la intensidad de la pobreza multidimensional, es decir, la cantidad promedio de privaciones experimentadas por las personas en situación de pobreza. Proporciona información sobre la gravedad y la profundidad de la pobreza multidimensional en un país.
- Number of people pushed below the 3.65 dls (dls 2017 PPP) poverty line by out-of-pocket health care expenditure: Este indicador muestra el número de personas que han caído por debajo de la línea de pobreza de dls 3.65 (en paridad de poder adquisitivo de 2017) debido a los gastos de atención médica pagados directamente por ellas. Indica el impacto de los gastos de salud en la situación económica de la población.
- Number of people spending more than 10% of household consumption or income on out-of-pocket health care expenditure: Este indicador muestra el número de personas que gastan más del 10% del consumo o ingreso familiar en gastos de atención médica pagados directamente por ellas. Proporciona información sobre la carga económica de los gastos de salud en los hogares y la capacidad de afrontar dichos gastos.
- Number of people spending more than 25% of household consumption or income on out-of-pocket health care expenditure: Este indicador muestra el número de personas que gastan más del 25% del consumo o ingreso familiar en gastos de atención médica pagados directamente por ellas. Proporciona información sobre la carga económica significativa de los gastos de salud en los hogares y su impacto en el bienestar financiero de las familias.
- Number of surgical procedures (per 100,000 population): Este indicador muestra el número de procedimientos quirúrgicos realizados por cada 100,000 personas en una población. Proporciona información sobre la disponibilidad y acceso a servicios quirúrgicos en un país, lo cual puede ser indicativo de la infraestructura de salud y la capacidad de brindar atención médica.
- Nurses and midwives (per 1,000 people): Este indicador muestra el número de enfermeras y parteras por cada 1,000 personas en una población. Proporciona información sobre la disponibilidad de profesionales de enfermería y partería, que desempeñan un papel crucial en la prestación de servicios de salud y atención materna e infantil.
- People practicing open defecation (% of population): Este indicador muestra el porcentaje de la población que practica la defecación al aire libre, es decir, la falta de acceso a servicios

de saneamiento adecuados. Proporciona información sobre las condiciones de higiene y la falta de infraestructura sanitaria en una población.

- People using at least basic drinking water services (% of population): Este indicador muestra el porcentaje de la población que utiliza al menos servicios básicos de agua potable. Mide la proporción de personas que tienen acceso a una fuente mejorada de agua potable, lo cual es esencial para la salud y el bienestar de la población.
- People using at least basic sanitation services (% of population): Este indicador muestra el porcentaje de la población que utiliza al menos servicios básicos de saneamiento. Mide la proporción de personas que tienen acceso a instalaciones de saneamiento mejoradas, lo cual es fundamental para la salud pública y la prevención de enfermedades.
- People using safely managed drinking water services (% of population): Este indicador muestra el porcentaje de la población que utiliza servicios de agua potable gestionados de manera segura. Mide la proporción de personas que tienen acceso a una fuente de agua potable segura, lo cual implica que la fuente está protegida contra la contaminación y garantiza una calidad adecuada del agua.
- People using safely managed sanitation services (% of population): Este indicador muestra el porcentaje de la población que utiliza servicios de saneamiento gestionados de manera segura. Mide la proporción de personas que tienen acceso a instalaciones de saneamiento que están diseñadas de manera segura, protegidas contra la contaminación y aseguran una disposición adecuada de los desechos.
- People with basic handwashing facilities including soap and water (% of population): Este indicador muestra el porcentaje de la población que cuenta con instalaciones básicas para el lavado de manos, incluyendo jabón y agua. Proporciona información sobre la disponibilidad de recursos para la higiene personal, lo cual es fundamental para prevenir enfermedades y promover la salud.
- Political Stability and Absence of Violence/Terrorism: Estimate: Este indicador muestra una estimación de la estabilidad política y la ausencia de violencia/terrorismo en un país. Evalúa la percepción y el nivel de seguridad en términos de estabilidad política y la presencia o ausencia de actos violentos o terroristas.
- Political Stability and Absence of Violence/Terrorism: Percentile Rank: Este indicador muestra el ranking de un país en términos de estabilidad política y la ausencia de violencia/terrorismo en comparación con otros países. Proporciona una medida relativa de la estabilidad política y la seguridad en un país.
- Population ages 65 and above, total: Este indicador muestra el número total de personas en una población que tienen 65 años o más. Proporciona información sobre la estructura demográfica y la distribución de edades en una población.
- Population density (people per sq. km of land area): Este indicador muestra la densidad de población, es decir, el número de personas por kilómetro cuadrado de área terrestre. Proporciona información sobre la concentración y distribución de la población en un área geográfica determinada.
- Population growth (annual %): Este indicador muestra la tasa de crecimiento anual de la población. Indica el cambio porcentual en el tamaño de la población durante un año específico y refleja la dinámica de crecimiento demográfico de un país.
- Population in largest city: Este indicador muestra el número de personas que viven en la ciudad más grande de un país. Proporciona información sobre la concentración urbana y la importancia relativa de una ciudad en términos de población.



- Population in the largest city (% of urban population): Este indicador muestra el porcentaje de la población urbana que vive en la ciudad más grande de un país. Proporciona información sobre la proporción de la población urbana que se concentra en la ciudad más grande en comparación con otras áreas urbanas.
- Population in urban agglomerations of more than 1 million (% of total population): Este indicador muestra el porcentaje de la población total que vive en aglomeraciones urbanas con más de 1 millón de habitantes. Proporciona información sobre la urbanización y la concentración de la población en áreas urbanas densamente pobladas.
- Population living in slums (% of urban population): Este indicador muestra el porcentaje de la población urbana que vive en asentamientos informales (slums). Proporciona información sobre las condiciones de vivienda precarias y la falta de servicios básicos en los asentamientos urbanos más pobres.
- Population, female: Este indicador muestra el número total de mujeres en una población. Proporciona información sobre la estructura demográfica y la distribución de género en una población.
- Population, male: Este indicador muestra el número total de hombres en una población. Proporciona información sobre la estructura demográfica y la distribución de género en una población.
- Population, total: Este indicador muestra el número total de personas en una población. Es una medida fundamental para comprender el tamaño de una población y su evolución a lo largo del tiempo.
- Poverty headcount ratio at national poverty lines (% of population): Este indicador muestra el porcentaje de la población que vive por debajo de la línea de pobreza nacional. Proporciona información sobre la incidencia de la pobreza en un país y la proporción de personas que viven en condiciones de pobreza.
- Prevalence of current tobacco use (% of adults): Este indicador muestra el porcentaje de adultos que son consumidores actuales de tabaco. Proporciona información sobre la prevalencia del consumo de tabaco en una población y su impacto en la salud pública.
- Prevalence of moderate or severe food insecurity in the population (%): Este indicador muestra el porcentaje de la población que experimenta inseguridad alimentaria moderada o grave. Mide la falta de acceso regular y adecuado a alimentos suficientes y nutricionalmente adecuados en una población.
- Prevalence of severe food insecurity in the population (%): Este indicador muestra el porcentaje de la población que experimenta inseguridad alimentaria grave. Mide la falta de acceso regular y adecuado a alimentos suficientes y nutricionalmente adecuados en una población, en su forma más severa.
- Prevalence of undernourishment (% of population): Este indicador muestra el porcentaje de la población que padece desnutrición. Mide la proporción de personas que no reciben suficientes nutrientes para satisfacer sus necesidades alimentarias diarias.
- Proportion of population pushed below the dls 3.65 (dls 2017 PPP) poverty line by out-of-pocket health care expenditure (%): Este indicador muestra el porcentaje de la población que cae por debajo de la línea de pobreza de dls 3.65 debido a los gastos de atención médica pagados de su propio bolsillo. Proporciona información sobre el impacto económico de los gastos de salud en la población.
- Proportion of population spending more than 10% of household consumption or income on out-of-pocket health care expenditure (%): Este indicador muestra el porcentaje de la población que gasta más del 10% de su consumo o ingreso familiar en gastos de atención

médica pagados de su propio bolsillo. Mide el impacto económico de los gastos de salud en la población.

- Railways, passengers carried (million passenger-km): Este indicador muestra el número total de pasajeros transportados por ferrocarril en millones de pasajeros-kilómetro. Proporciona información sobre la utilización y la importancia relativa del transporte ferroviario en un país.
- Real interest rate (%): Este indicador muestra la tasa de interés real, es decir, la tasa de interés ajustada por inflación. Refleja el rendimiento real de las inversiones y el costo real del endeudamiento, teniendo en cuenta el efecto de la inflación.
- Risk of catastrophic expenditure for surgical care (% of people at risk): Este indicador muestra el riesgo de gastos catastróficos debido a la atención quirúrgica, como porcentaje de las personas en riesgo. Mide la proporción de personas que enfrentan un alto nivel de gastos en atención quirúrgica en relación con sus ingresos o capacidad de pago.
- Risk of impoverishing expenditure for surgical care (% of people at risk): Este indicador muestra el riesgo de empobrecimiento debido a los gastos en atención quirúrgica, como porcentaje de las personas en riesgo. Mide la proporción de personas cuyos gastos en atención quirúrgica los empujan por debajo de la línea de pobreza o reducen su capacidad adquisitiva.
- Rural population (% of total population): Este indicador muestra el porcentaje de la población que vive en áreas rurales en relación con la población total. Proporciona información sobre la distribución de la población entre áreas rurales y urbanas en un país.
- Survival to age 65, female (% of cohort): Este indicador muestra el porcentaje de mujeres de una cohorte que sobreviven hasta los 65 años de edad. Proporciona información sobre la esperanza de vida y la salud de las mujeres en un país.
- Survival to age 65, male (% of cohort): Este indicador muestra el porcentaje de hombres de una cohorte que sobreviven hasta los 65 años de edad. Proporciona información sobre la esperanza de vida y la salud de los hombres en un país.
- Total alcohol consumption per capita (liters of pure alcohol, projected estimates, 15+ years of age): Este indicador muestra el consumo total de alcohol per cápita en litros de alcohol puro, proyectado para personas de 15 años en adelante. Proporciona información sobre los niveles de consumo de alcohol en una población.
- Tuberculosis case detection rate (% of all forms): Este indicador muestra el porcentaje de casos de tuberculosis detectados en relación con todos los casos de tuberculosis, incluyendo todas las formas de la enfermedad. Mide la efectividad de los sistemas de detección y diagnóstico de la tuberculosis en un país.
- Tuberculosis treatment success rate (% of new cases): Este indicador muestra el porcentaje de casos nuevos de tuberculosis que se tratan exitosamente. Mide la eficacia del tratamiento de la tuberculosis en un país y su capacidad para curar la enfermedad.
- UHC service coverage index: Este indicador muestra el índice de cobertura de servicios de salud universal (UHC, por sus siglas en inglés). Mide el grado de cobertura de servicios de salud esenciales y accesibles para toda la población, sin importar su capacidad de pago.
- Urban population (% of total population): Este indicador muestra el porcentaje de la población que vive en áreas urbanas en relación con la población total. Proporciona información sobre la distribución de la población entre áreas urbanas y rurales en un país.

```
indicators_to_keep = [
    'EG.CFT.ACCS.ZS', 'EG.ELC.ACCS.ZS', 'per_si_allsi.adq_pop_tot',
    'per_sa_allsa.adq_pop_tot', 'NY.ADJ.NNTY.KD.ZG', 'NY.ADJ.NNTY.PC.KD.ZG',
    'IS.AIR.PSGR', 'ER.H2O.FWDM.ZS', 'FB.BNK.CAPA.ZS',
```

```

'FD.RES.LIQU.AS.ZS', 'SP.DYN.CBRT.IN', 'SH.STA.BRTC.ZS',
'SH.DTH.COMM.ZS', 'SH.DTH.INJR.ZS', 'SH.DTH.NCOM.ZS',
'SH.MED.CMHW.P3', 'CC.EST', 'CC.PER.RNK',
'per_si_allsi.cov_pop_tot', 'per_allsp.cov_pop_tot', 'per_sa_allsa.cov_pop_tot',
'SE.XPD.CTOT.ZS', 'SH.XPD.CHEX.GD.ZS', 'SH.XPD.CHEX.PP.CD',
'SP.DYN.CDRT.IN', 'SH.STA.DIAB.ZS', 'EN.CLC.DRSK.XQ',
'SH.XPD.GHED.CH.ZS', 'SH.XPD.GHED.GD.ZS', 'SH.XPD.GHED.GE.ZS',
'SH.XPD.PVTD.CH.ZS', 'SE.PRM.CUAT.ZS', 'SH.XPD.EHEX.CH.ZS',
'IT.NET.BBND.P2', 'IT.MLT.MAIN.P2', 'AG.LND.FRST.ZS',
'NY.GDP.MKTP.CD', 'NY.GDP.DEFL.ZS', 'NY.GDP.DEFL.ZS.AD',
'NY.GDP.MKTP.KD.ZG', 'NY.GDP.PCAP.CD', 'NE.CON.GOVT.ZS',
'SI.POV.GINI', 'NY.GNP.MKTP.KD.ZG', 'NY.GNP.PCAP.KD.ZG',
'NE.DAB.TOTL.ZS', 'HD.HCI.OVRL', 'SH.IMM.IDPT',
'SH.IMM.HEPB', 'SH.IMM.MEAS', 'SH.HIV.INCD.TL.P3',
'SH.MLR.INCD.P3', 'SH.TBS.INCD', 'SH.UHC.NOP1.ZG',
'SH.UHC.NOP2.ZG', 'IT.NET.USER.ZS', 'FP.CPI.TOTL.ZG',
'VC.IHR.PSRC.P5', 'AG.LND.TOTL.K2', 'ER.H2O.FWST.ZS',
'SP.DYN.LE00.MA.IN', 'SP.DYN.LE00.FE.IN', 'SP.DYN.LE00.IN',
'SE.ADT.LITR.ZS', 'IT.CEL.SETS.P2', 'SH.STA.WASH.P5',
'I.POV.MDIM', 'SI.POV.MDIM.XQ', 'SI.POV.MDIM.IT',
'SH.UHC.NOP2.TO', 'SH.UHC.OOPC.10.TO', 'SH.UHC.OOPC.25.TO',
'SH.SGR.PROC.P5', 'SH.MED.NUMW.P3', 'SH.STA.ODFC.ZS',
'SH.H2O.BASW.ZS', 'SH.STA.BASS.ZS', 'SH.H2O.SMDW.ZS',
'SH.STA.SMSS.ZS', 'SH.STA.HYGN.ZS', 'PV.EST',
'PV.PER.RNK', 'SP.POP.65UP.TO', 'EN.POP.DNST',
'SP.POP.GROW', 'EN.URB.LCTY', 'EN.URB.LCTY.UR.ZS',
'EN.URB.MCTY.TL.ZS', 'EN.POP.SLUM.UR.ZS', 'SP.POP.TOTL.FE.IN',
'SP.POP.TOTL.MA.IN', 'SP.POP.TOTL', 'SI.POV.NAHC.SH.PR.V.SMOK',
'SN.ITK.MSFI.ZS', 'SN.ITK.SVFI.ZS', 'SN.ITK.DEFC.ZS',
'SH.UHC.NOP2.ZS', 'SH.UHC.OOPC.10.ZS', 'IS.RRS.PASG.KM',
'FR.INR.RINR', 'SH.SGR.CRSK.ZS', 'SH.SGR.IRSK.ZS',
'SP.RUR.TOTL.ZS', 'SP.DYN.TO65.FE.ZS', 'SP.DYN.TO65.MA.ZS',
'SH.ALC.PCAP.LI', 'SH.TBS.DTEC.ZS', 'SH.TBS.CURE.ZS',
'SH.UHC.SRVS.CV.XD', 'SP.URB.TOTL.IN.ZS'
]

WDI = WDI[['Country Name', 'Country Code', 'Indicator Name', 'Indicator Code', '2018', '2019',
'2020', '2021', '2022']]
WDI = WDI[WDI['Indicator Code'].isin(indicators_to_keep)]

WDI['Indicator Code'].nunique()

```

Por tanto, habríamos podido pasar de 1478 a 108, es importante tener en cuenta que estos indicadores son una preselección de cuestiones que podrían relacionarse con nuestro foco de investigación

Al final habremos obtenido una tabla con los datos de 2018 a 2022 de Los Indicadores de Desarrollo Mundial (WDI) del Worl Bank por países según su ISO3.

Obtenido de: World Bank. (Revisado en julio de 2023). World Development Indicators. Recuperado de: <https://datacatalog.worldbank.org/search/dataset/0037712/World-Development-Indicators>

Sólo nos queda obtener el ISO2

```

# Agregar la columna 'PAIS_ISO2' a 'WDI' utilizando el mapeo
WDI['PAIS_ISO2'] = WDI['Country Code'].map(países.set_index('PAIS_ISO3')['PAIS_ISO2'].to_dict())
WDI

```

Out[43]:

	Country Name	Country Code	Indicator Name	Indicator Code	2018	2019	2020	2021	2022	PAIS_ISO2
72422	Afghanistan	AFG	Access to clean fuels and technologies for coo...	EG.CFT.ACCS.ZS	31.100000	32.45	33.800	35.400	NaN	AF
72425	Afghanistan	AFG	Access to electricity (% of population)	EG.ELC.ACCS.ZS	93.430878	97.70	97.700	97.700	NaN	AF
72437	Afghanistan	AFG	Adequacy of social insurance programs (% of to...	per_si_allsi.adq_pop_tot	NaN	NaN	NaN	NaN	NaN	AF
72439	Afghanistan	AFG	Adequacy of social safety net programs (% of t...	per_sa_allsa.adq_pop_tot	NaN	NaN	NaN	NaN	NaN	AF

Blame 5316 lines (5316 loc) · 1.43 MB

growth)

...	...	...	...	...	...	...	...	...	...	...
393043	Zimbabwe	ZWE	Total alcohol consumption per capita (liters o...	SH.ALC.PCAP.LI	4.670000	NaN	NaN	NaN	NaN	ZW
393084	Zimbabwe	ZWE	Tuberculosis case detection rate (% all forms)	SH.TBS.DTEC.ZS	80.000000	69.00	55.000	54.000	NaN	ZW
393085	Zimbabwe	ZWE	Tuberculosis treatment success rate (% of new ...	SH.TBS.CURE.ZS	84.000000	84.00	88.000	NaN	NaN	ZW
393086	Zimbabwe	ZWE	UHC service coverage index	SH.UHC.SRVS.CV.XD	NaN	55.00	NaN	55.000	NaN	ZW
393113	Zimbabwe	ZWE	Urban population (% of total population)	SP.URB.TOTL.IN.ZS	32.209000	32.21	32.242	32.303	32.395	ZW

23220 rows × 10 columns

Dataset obtenido de: <https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data>

```
nRowsRead = None
temp = pd.read_csv('C:/Users/Patricia/Desktop/Github/TFM/DF Covid/Temperatura/GlobalLandTemperaturesByCountry.csv', delimiter=',', nrows = nRowsRead)
temp.dataframeName = 'GlobalLandTemperaturesByCountry.csv'
nRow, nCol = temp.shape
print(f'There are {nRow} rows and {nCol} columns')
```

577461 2013-09-01 NaN NaN Zimbabwe

Debido a que es un dataset enorme nos quedamos sólo con el 2000 en adelante

```
temp = temp[temp['dt'].str.startswith('20')]
temp.loc[:, 'dt'] = pd.to_datetime(temp['dt'], format='%Y-%m-%d')
temp.loc[:, 'DIA_DEL_AÑO'] = temp['dt'].dt.dayofyear
temp.loc[:, 'MES'] = temp['dt'].dt.month
temp.loc[:, 'AÑO'] = temp['dt'].dt.year
```

```
temp.sort_values(['AÑO', 'DIA_DEL_AÑO'], inplace=True)
temp.drop('dt', axis=1, inplace=True)
```

Sólo nos faltaría obtener el ISO2

```
# Agregar la columna 'PAIS_ISO2' a 'WDI' utilizando el mapeo
temp['PAIS_ISO2'] = temp['Country'].map(paises.set_index('PAIS_NOM')['PAIS_ISO2'].to_dict())
```

Cómo vemos hay distintos registros para un mismo país y mes, tenemos que hacer la media entre ellos para que tengamos el registro medio de cada país en cada mes

```
temperatura = temp.groupby(['PAIS_ISO2', 'MES'])['AverageTemperature'].mean().reset_index()
```

y habremos obtenido un dataset con la temperatura media mensual de los países según su ISO2

Es importante tener en cuenta que estos datos no entran lo suficiente en detalle para tener una idea real del clima en un país pues nos da una media para todo el territorio sin tomar en cuenta fenómenos o variaciones geográficas, pero para el propósito de saber si en invierno o verano nos funcionarán

## ANÁLISIS INDIVIDUAL POR DATASET

### DATA FRAME COVID DAILY OMS GLOBAL

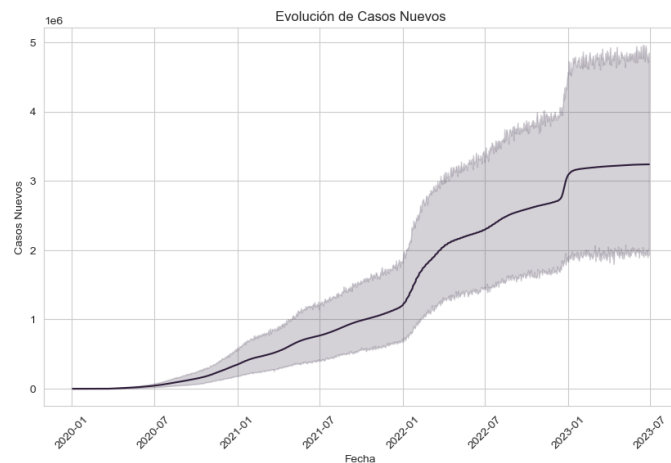
```
coviddaily = coviddaily.drop("Unnamed: 0", axis=1) # Se elimina la fila unnamed, esta fila correspond
coviddaily['FECHA_INFORMADA'] = pd.to_datetime(coviddaily['FECHA_INFORMADA'], format='%d-%m-%Y')
coviddaily['DIA_DEL_AÑO'] = coviddaily['FECHA_INFORMADA'].dt.dayofyear
coviddaily['MES'] = coviddaily['FECHA_INFORMADA'].dt.month
coviddaily['AÑO'] = coviddaily['FECHA_INFORMADA'].dt.year

coviddaily.sort_values(['AÑO', 'DIA_DEL_AÑO'], inplace=True)
```

#### Graficamos los casos y muertes nuevas

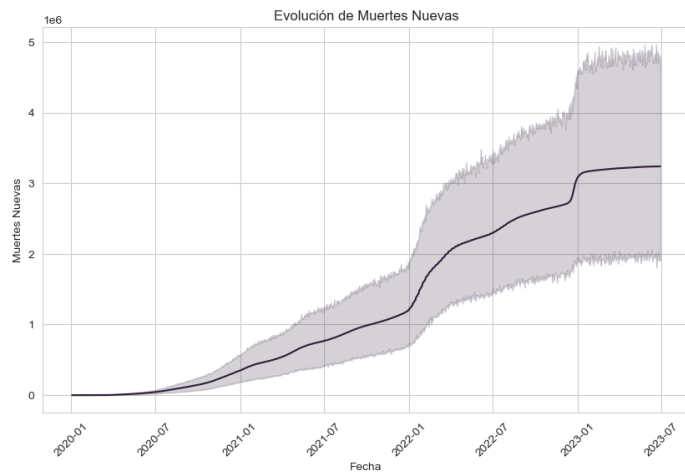
```
plt.figure(figsize=(10, 6)) # Tamaño de la figura
```

```
# Gráfico de casos nuevos
sns.lineplot(x='FECHA_INFORMADA', y='CASOS_ACUM', data=coviddaily)
plt.xlabel('Fecha') # Etiqueta del eje x
plt.ylabel('Casos Nuevos') # Etiqueta del eje y
plt.title('Evolución de Casos Nuevos') # Título del gráfico
plt.xticks(rotation=45) # Rota las etiquetas del eje x para una mejor legibilidad
plt.grid(True) # Muestra la cuadrícula
plt.show() # Muestra el gráfico
```



```
plt.figure(figsize=(10, 6)) # Tamaño de la figura
```

```
# Gráfico de muertes nuevas
sns.lineplot(x='FECHA_INFORMADA', y='CASOS_ACUM', data=coviddaily)
plt.xlabel('Fecha') # Etiqueta del eje x
plt.ylabel('Muertes Nuevas') # Etiqueta del eje y
plt.title('Evolución de Muertes Nuevas') # Título del gráfico
plt.xticks(rotation=45) # Rota las etiquetas del eje x para una mejor legibilidad
plt.grid(True) # Muestra la cuadrícula
plt.show() # Muestra el gráfico
```



```
# Agrupar los datos por país y calcular la suma de casos y muertes acumulados
grouped = coviddaily.groupby('PAIS_ISO3').agg({'CASOS_ACUM': 'sum', 'MUERTES_ACUM': 'sum'})

# Obtener los nombres de los países y los valores de casos y muertes acumulados
países = grouped.index
cases = grouped['CASOS_ACUM']
deaths = grouped['MUERTES_ACUM']

# Configurar el tamaño de la figura y los ejes
fig, ax = plt.subplots(figsize=(10, 30))

# Crear el gráfico de barras horizontales para casos acumulados
ax.barh(países, cases, label='Cases - Cumulative Total')

# Crear el gráfico de barras horizontales para muertes acumuladas
ax.barh(países, deaths, label='Deaths - Cumulative Total')

# Configurar las etiquetas de los ejes y el título del gráfico
ax.set_xlabel('Count')
ax.set_ylabel('PAIS_ISO3')
ax.set_title('Cases and Deaths - Cumulative Total by Country')

# Mostrar una leyenda
ax.legend()

# Ajustar el espacio entre las barras
plt.tight_layout()

# Mostrar el gráfico de barras
plt.show()
```



```
# Imprimir el coeficiente de correlación y el valor p
print('Coeficiente de correlación de Spearman:', correlacion_spearman)
print('Valor p:', p_valor)
```

```
Coeficiente de correlación de Spearman: -0.05202680413632658
Valor p: 7.49643620172126e-180
```

La correlación de Spearman mide la relación monotónica entre dos variables. En este caso, el coeficiente de correlación de -0.052 indica una correlación débil y negativa entre la variable categórica (PAIS) y la variable numérica (CASOS\_ACUM). Esto sugiere que no hay una relación lineal clara entre el país y el número acumulado de casos de COVID-19.

El valor p extremadamente bajo (7.496e-180) indica una fuerte evidencia en contra de la hipótesis nula de no correlación. En otras palabras, es altamente improbable que la correlación observada se deba al azar. Sin embargo, debido a que la correlación es muy cercana a cero, su relevancia práctica puede ser limitada

### COVID DAILY US

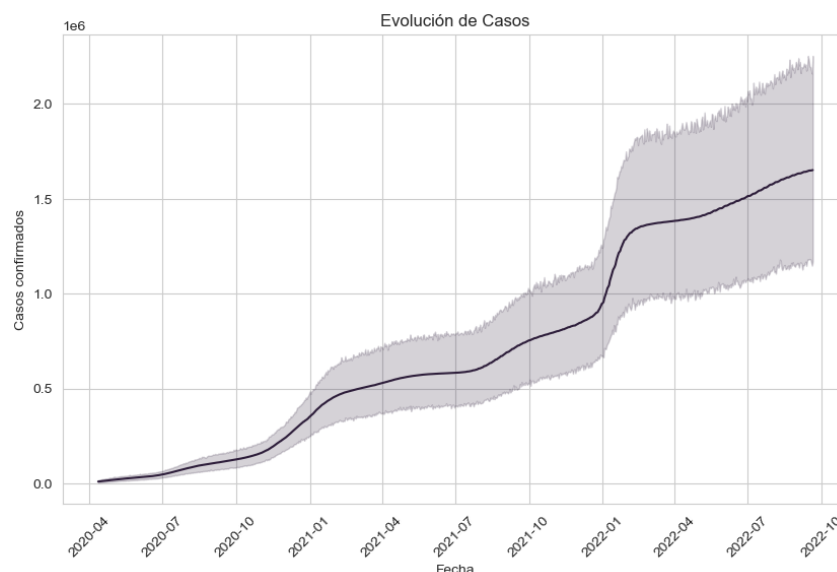
```
coviddaily_us = pd.read_csv("C:/Users/Patricia/Desktop/Github/TFM/DF/
Covid/df_daily_report_us_final.csv")
```

```
coviddaily_us['Date'] = pd.to_datetime(coviddaily_us['Date'], format='%d.%m.%Y')
coviddaily_us['DIA_DEL_AÑO'] = coviddaily_us['Date'].dt.dayofyear
coviddaily_us['MES'] = coviddaily_us['Date'].dt.month
coviddaily_us['AÑO'] = coviddaily_us['Date'].dt.year
```

```
coviddaily_us.sort_values(['AÑO', 'DIA_DEL_AÑO'], inplace=True)
```

```
plt.figure(figsize=(10, 6)) # Tamaño de la figura
```

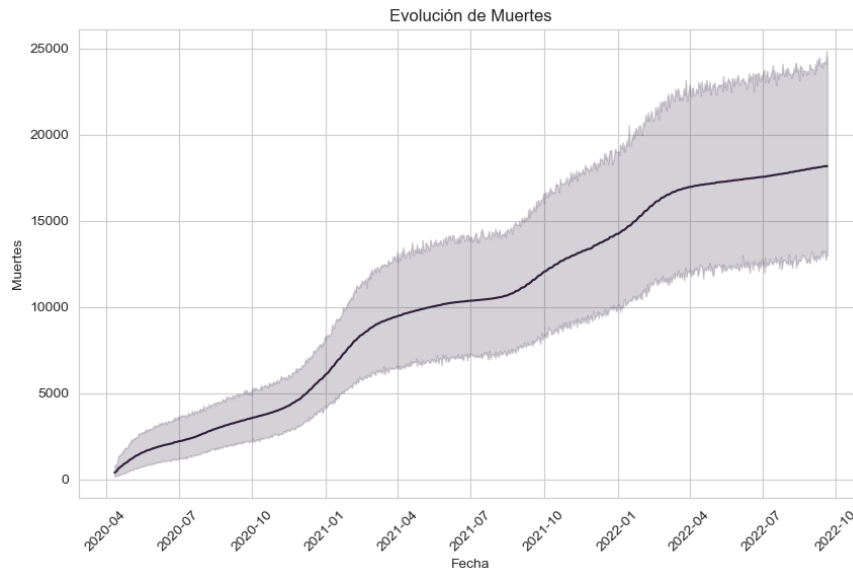
```
# Gráfico de casos nuevos
sns.lineplot(x='Date', y='Confirmed', data=coviddaily_us)
plt.xlabel('Fecha') # Etiqueta del eje x
plt.ylabel('Casos confirmados') # Etiqueta del eje y
plt.title('Evolución de Casos') # Título del gráfico
plt.xticks(rotation=45) # Rota las etiquetas del eje x para una mejor legibilidad
plt.grid(True) # Muestra la cuadrícula
plt.show() # Muestra el gráfico
```



```
plt.figure(figsize=(10, 6)) # Tamaño de la figura
```



```
# Gráfico de muertes nuevas
sns.lineplot(x='Date', y='Deaths', data=coviddaily_us)
plt.xlabel('Fecha') # Etiqueta del eje x
plt.ylabel('Muertes') # Etiqueta del eje y
plt.title('Evolución de Muertes') # Título del gráfico
plt.xticks(rotation=45) # Rota las etiquetas del eje x para una mejor legibilidad
plt.grid(True) # Muestra la cuadrícula
plt.show() # Muestra el gráfico
```



## DATOS UCI

```
uci_hosp
pd.read_csv("C:/Users/Patricia/Desktop/Github/TFM/DFCovid/df_daily_report_us_final.csv")
uci_hosp.isna().sum()
Unnamed: 0      0
Province_State  0
Confirmed       0
Deaths         0
Recovered      46492
Active         46492
ISO3           0
Date          9860
dtype: int64
```

No contamos con la columna Recovered pero deberían ser aquellos casos confirmados que no han fallecido

```
# Reemplazar los NaN en "Recovered" con los valores de "Confirmed" que no figuran como "Deaths"
uci_hosp['Recovered'] = uci_hosp['Recovered'].fillna(uci_hosp['Confirmed'])
uci_hosp['Deaths']
# Convertir la columna "Date" a tipo de datos datetime
uci_hosp['Date'] = pd.to_datetime(uci_hosp['Date'], format='%d.%m.%Y')
```

Eliminamos aquellos sin fecha pues no son registros que podamos situar

```
uci_hosp.dropna(subset=['Date'], inplace=True)
```

Organizamos las fechas

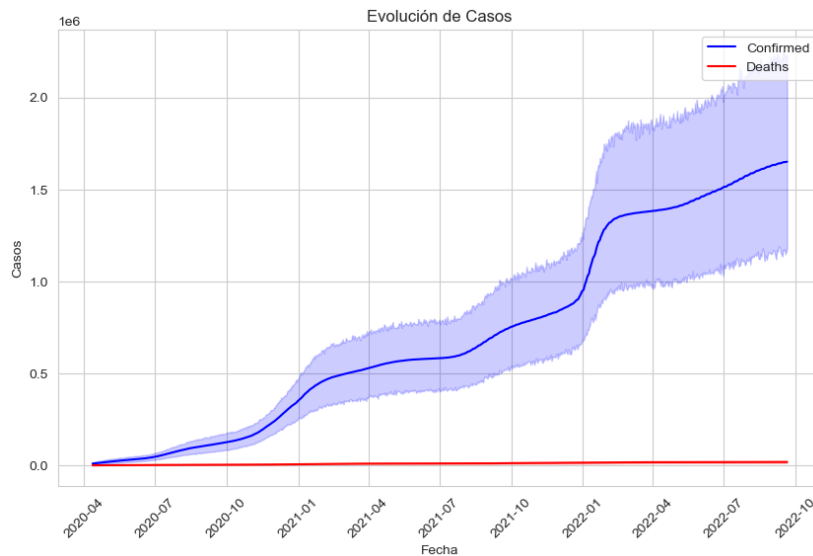
```
uci_hosp['DIA_DEL_AÑO'] = uci_hosp['Date'].dt.dayofyear
uci_hosp['MES'] = uci_hosp['Date'].dt.month
uci_hosp['AÑO'] = uci_hosp['Date'].dt.year
uci_hosp.sort_values(['AÑO', 'DIA_DEL_AÑO'], inplace=True)
```

Graficamos los casos y muertes confirmadas

```
plt.figure(figsize=(10, 6)) # Tamaño de la figura

# Gráfico de casos
sns.lineplot(x='Date', y='Confirmed', data=uci_hosp, color='blue', label='Confirmed')
sns.lineplot(x='Date', y='Deaths', data=uci_hosp, color='red', label='Deaths')

plt.xlabel('Fecha') # Etiqueta del eje x
plt.ylabel('Casos') # Etiqueta del eje y
plt.title('Evolución de Casos') # Título del gráfico
plt.xticks(rotation=45) # Rota las etiquetas del eje x para una mejor legibilidad
plt.grid(True) # Muestra la cuadrícula
plt.legend() # Muestra la leyenda con los nombres de las líneas
plt.show() # Muestra el gráfico
```



## COVID TESTING AT US

```
testing = pd.read_csv("C:/Users/Patricia/Desktop/Github/TFM/DF Covid/df_testing_covid_eu.csv")
```

```
# Corr por defecto será Pearson que es el más utilizado
print("Coeficiente de correlación de Pearson:",
testing['RATIO_TESTS'].corr(testing['RATIO_POSITIVO']))
```

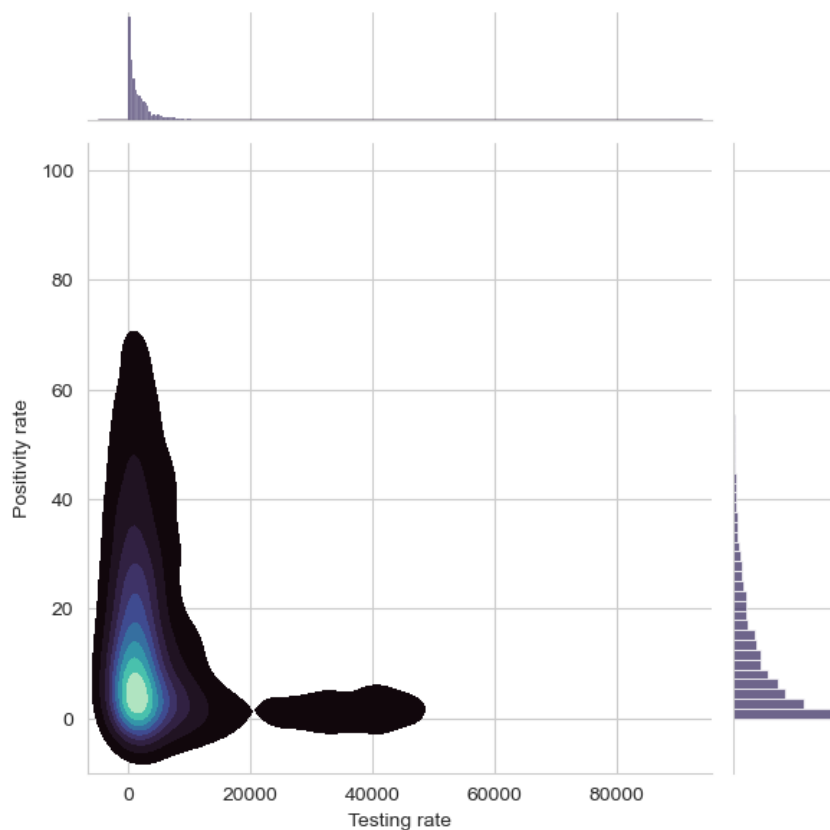
Coeficiente de correlación de Pearson: -0.18767002912753783

**Por tanto hablaríamos de una correlación negativa debil entre testing\_rate y positivity\_rate**

```
# Jointgrid con histograma marginal Testing_rate x positivity_rate
g = sns.jointplot(data=testing, x='RATIO_TESTS', y='RATIO_POSITIVO', kind="kde", cmap="flare",
fill=True)
g.plot_joint(sns.kdeplot, cmap="mako", fill=True)
g.plot_marginals(sns.histplot, color='#3D3164')

# Configurar etiquetas i títol
g.set_axis_labels("Testing rate", "Positivity rate")

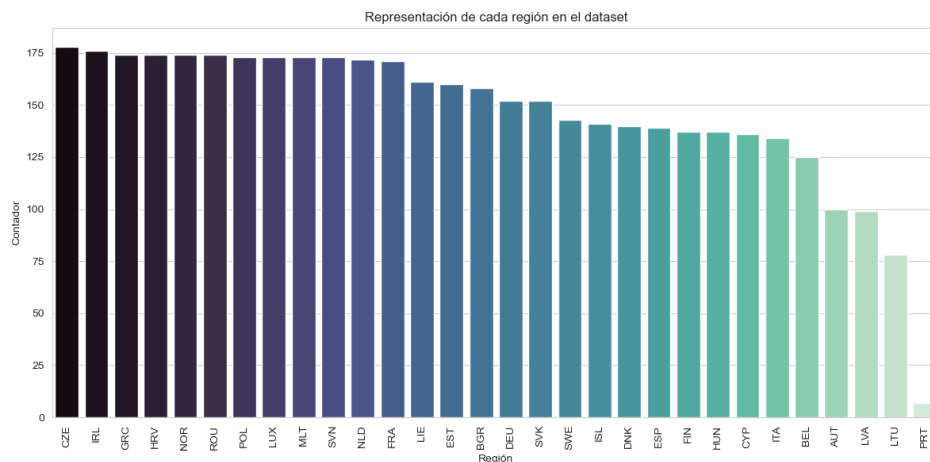
plt.show()
```



### Conteo de representación de regiones

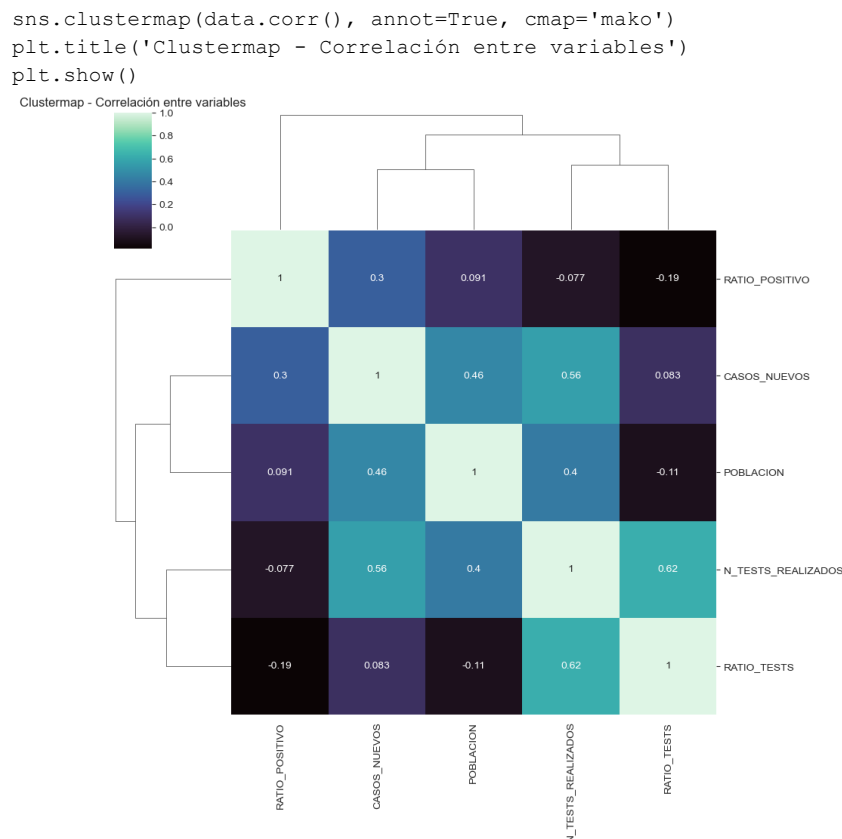
```
# Calcular el conteo de registros por región
region_counts = testing['PAIS_ISO3'].value_counts()

# Crear la visualización del conteo de registros por región
plt.figure(figsize=(12, 6)) # Ajustar el tamaño de la figura
sns.countplot(data=testing, x='PAIS_ISO3', palette='mako', order=region_counts.index)
plt.xlabel('Región')
plt.ylabel('Contador')
plt.title('Representación de cada región en el dataset')
plt.xticks(rotation=90) # Rotar las etiquetas de las regiones
plt.tight_layout() # Ajustar el espaciado
plt.show()
```



### Cluster maps de correlaciones

```
data = testing[['CASOS_NUEVOS', 'N_TESTS_REALIZADOS', 'POBLACION', 'RATIO_TESTS', 'RATIO_POSITIVO']]
```



## Cases and deaths

```
casesanddeath = pd.read_csv("C:/Users/Patricia/Desktop/Github/TFM/DF Covid/df_Latest reported counts of cases and deaths.csv")
```

En este no hemos realizado gráficos pues creemos que su potencial estaría en poder realizar un mapa coroplético.

## Vaccination global

```
vacc = pd.read_csv("C:/Users/Patricia/Desktop/Github/TFM/DF Covid/df_vaccination.csv")
```

En este también lo idóneo sería hacer cruce con cases and deaths pero a realizar en la próxima entrega

## Vaccines type

```
vacctype = pd.read_csv("C:/Users/Patricia/Desktop/Github/TFM/DF Covid/df_vaccination_tipo.csv")
vacctype['NOMBRE_VACUNA'].nunique()
38
Vemos que el dataset recoge 38 vacunas diferentes
# Contar el número de vacunas autorizadas y reordenar el dataset
vacunas_por_pais = vacctype.groupby('PAIS_ISO3')['NOMBRE_VACUNA'].nunique().reset_index().sort_values(by='NOMBRE_VACUNA', ascending=False)

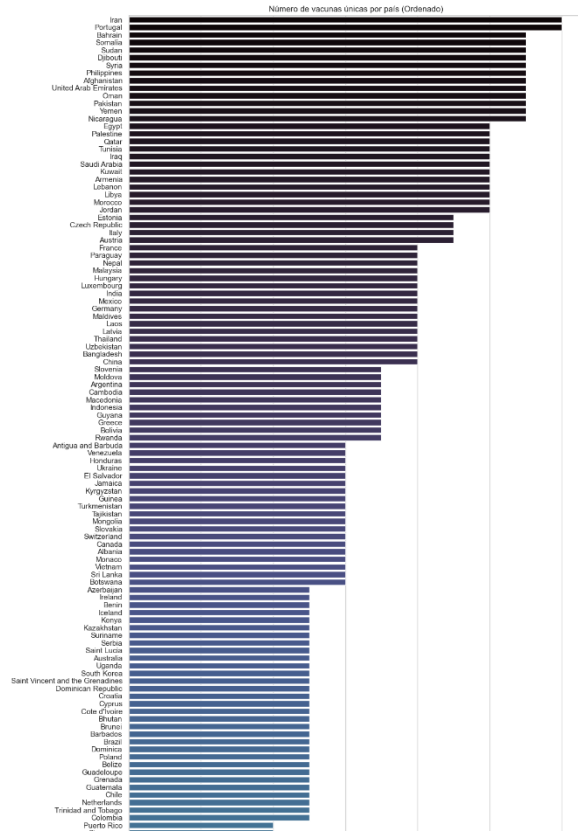
# Realizar merge con el dataframe de países para obtener el nombre completo del país
vacunas_por_pais = vacunas_por_pais.merge(países[['PAIS_ISO3', 'PAIS_NOM']],
left_on='PAIS_ISO3', right_on='PAIS_ISO3', how='left')

# Configurar estilo de Seaborn
sns.set(style="whitegrid")

# Crear el gráfico de barras horizontales con paleta 'mako'
plt.figure(figsize=(12, 45))
```

```
sns.barplot(data=vacunas_por_pais, x='NOMBRE_VACUNA', y='PAIS_NOM', palette='mako')
plt.xlabel('Número de vacunas aprobadas')
plt.ylabel('País')
plt.title('Número de vacunas aprobadas por país')
plt.xticks(rotation=0)
```

```
# Mostrar el gráfico
plt.show()
```





- Hipótesis nula (H04): No hay diferencia en la incidencia de Covid-19 entre zonas urbanas y zonas no urbanas.
- Hipótesis de investigación (H5): El acceso a las medidas higiénicas afecta directamente en la transmisión de Covid-19.
  - Hipótesis nula (H05): No hay relación entre el acceso a las medidas higiénicas y la transmisión de Covid-19.
- Hipótesis de investigación (H6): La vacunación ha reducido la mortalidad del Covid-19.
  - Hipótesis nula (H06): La vacunación no ha reducido la mortalidad del Covid-19.

Estas hipótesis nos permiten formular suposiciones basadas en las relaciones esperadas entre variables relevantes y la incidencia, propagación y consecuencias del Covid-19. Las hipótesis nulas representan la ausencia de dichas relaciones. A través de análisis de datos y pruebas estadísticas, podremos evaluar la evidencia en apoyo o en contra de estas hipótesis, lo que contribuirá a nuestra comprensión y conocimiento sobre la pandemia de Covid-19.

Es importante destacar que las hipótesis deben ser probadas y evaluadas de manera rigurosa, utilizando métodos adecuados de análisis estadístico y teniendo en cuenta otros factores que puedan influir en los resultados. Asimismo, se respaldarán las afirmaciones con bibliografía académica especializada.