**Proyecto Final: Esperanza de Vida al Nacer**





Somos una consultora especializada en análisis y toma de decisiones basada en datos. Nuestro equipo está conformado por profesionales altamente capacitados y con amplia experiencia en proyectos de consultoría de datos, lo que nos permite abordar de manera integral proyectos de diversa complejidad y magnitud.

Nuestra misión es ayudar a nuestros clientes a obtener información valiosa a partir de sus datos para mejorar su toma de decisiones y optimizar sus procesos. Nos destacamos por brindar soluciones personalizadas, adaptadas a las necesidades y objetivos específicos de cada cliente. Trabajamos de manera colaborativa, estableciendo una estrecha relación con nuestros clientes para comprender a fondo su problemática y ofrecerles las mejores soluciones.

📜 **Semana 1: Propuesta de proyecto**

**Entendimiento de la situación actual:**

La empresa farmacéutica Pfizer ha delineado su objetivo de desarrollar un nuevo medicamento. Para dar inicio a la planificación de este proyecto, la empresa requiere recopilar datos relevantes que influyan en la esperanza de vida, tasas de fecundidad promedio y otros factores demográficos clave. Esta información será fundamental para determinar la ubicación demográfica más adecuada en la cual invertir y llevar a cabo el desarrollo del medicamento.

Para lograr este objetivo, Pfizer ha contratado a nuestra consultora, con el fin de realizar un estudio sobre la esperanza de vida y otros indicadores relevantes para comprender los factores que influyen en la vitalidad de las personas.

Con el modelo de ML trataremos de resolver estos interrogantes

¿Qué hace que un país tenga mayor esperanza de vida?

¿Cuáles son las características predominantes para que un país tenga mayor esperanza de vida?

**Objetivos:**

1. Analizar las tendencias actuales de la esperanza de vida en diferentes países y regiones.

2. Identificar los factores determinantes de la esperanza de vida, como las tasas de mortalidad infantil, la calidad de los servicios de atención médica, el acceso a servicios básicos de salud y otros indicadores relevantes.

3. Proporcionar recomendaciones para el desarrollo del nuevo medicamento que promueve la fertilidad y la vitalidad en hombres y mujeres.

**Alcance:**

Recopilar y analizar datos demográficos, como la esperanza de vida al nacer, la edad media de la población, el índice de envejecimiento y otros, de diferentes fuentes confiables.

El estudio se centrará en países de América del Sur, América del Norte, Centroamérica y Oceanía, utilizando datos recopilados durante los últimos 30 años.

Como posibilidad de continuidad del proyecto, se plantea la expansión a otros países de los otros continentes o la inclusión de factores adicionales.

**KPIs:**

La metodología de determinación de KPI sigue el concepto de los reconocidos objetivos SMART. El acrónimo S (specific) específico, M (Mesurable) medible, A (Attainable) alcanzable, R (Relevant) relevante, T (Time-bound) con limite de tiempo.

* Esperanza de vida saludable - KPI aumento 0.1% comparando dos últimos años con datos. Es una medida que combina la esperanza de vida al nacer con la proporción de años vividos en buena salud. El índice de esperanza de vida saludable proporciona una visión más completa de la salud de una población al considerar tanto la duración como la calidad de vida.
* Mortalidad Infantil - KPI disminución 0.1% tasa de mortalidad infantil comparando dos últimos años con datos. Es la cantidad de niños que mueren antes de cumplir un año de edad por cada 1,000 nacidos vivos. Es un KPI crítico para evaluar la calidad de los servicios de atención médica materno-infantil y la salud general de la población.
* Mortalidad materna
* Esperanza de vida al nacer - KPI aumento 0.1% comparando dos últimos años con datos: Es el indicador principal que mide la esperanza de vida promedio de una población al nacer. Es un KPI fundamental para evaluar la salud y el bienestar general de una población.
* KPI: Disminución del 0,1% de la tasa de fecundidad en mujeres de edad 15 a 19 años.(verificar la data)
* KPI: Aumento del 0,1% de la tasa de fecundidad en mujeres mayores a 30 años(verificar la data)

Nota: El porcentaje de aumento o disminución está sujeto a modificación, que será probablemente definido la semana entrante.

**Repositorio GitHub:**

Se creó un repositorio público en GitHub para permitir la colaboración en el desarrollo del proyecto. Se utilizarán diferentes ramas (branches) y se mantendrá un control de versiones del trabajo .[ <https://github.com/datainsi/Esperanza_de_vida>]

**Solución propuesta:**

* **Stack tecnológico:**

**Lenguaje de programación: Python**

Hemos seleccionado Python como el lenguaje de programación principal para nuestros proyectos. Python es una elección excepcional debido a su versatilidad y popularidad en diversos campos, como el análisis de datos, la inteligencia artificial

Las librerías seleccionadas de Python para nuestro proyecto son:

* Pandas: es una librería fundamental para el análisis y manipulación de datos en Python. Proporciona estructuras de datos flexibles, como DataFrames, que facilitan la limpieza, transformación y exploración de datos. Con Pandas, podemos realizar operaciones avanzadas en datos tabulares de manera sencilla y eficiente.
* Scikit-learn: es una librería de aprendizaje automático que ofrece una amplia gama de algoritmos y herramientas para la construcción de modelos predictivos. Con Scikit-learn, podemos realizar tareas como clasificación, regresión, agrupación y selección de características. Además, cuenta con funciones para evaluar y optimizar modelos de aprendizaje automático.
* NumPy: es una librería esencial para la computación numérica en Python. Proporciona una estructura de matriz multidimensional eficiente y funciones matemáticas de alto rendimiento. Con NumPy, podemos realizar operaciones numéricas complejas de manera rápida y eficiente.
* Seaborn: es una librería de visualización de datos basada en Matplotlib. Ofrece una interfaz de alto nivel para crear gráficos estadísticos atractivos y claros. Con Seaborn, podemos visualizar datos de manera efectiva, lo que nos permite obtener insights valiosos y comunicar resultados de manera impactante.
* PySpark: es una librería de Python para el procesamiento distribuido de datos utilizando Apache Spark. Permite el procesamiento de grandes volúmenes de datos de manera escalable y eficiente. Con PySpark, podemos aprovechar la potencia del procesamiento paralelo para realizar análisis y transformaciones complejas en grandes conjuntos de datos.
* SQLAlchemy: es una librería de Python para trabajar con bases de datos relacionales. Proporciona una interfaz de alto nivel para interactuar con bases de datos de manera intuitiva y flexible. Con SQLAlchemy, podemos realizar consultas, modificar y administrar bases de datos de forma programática y segura.

La selección de estas librerías nos brinda una poderosa combinación de herramientas para el análisis de datos, el aprendizaje automático y la manipulación de datos, lo que nos permitirá desarrollar proyectos sólidos y efectivos."

Además de las librerías de Python mencionadas anteriormente, hemos seleccionado dos herramientas de visualización muy efectivas para potenciar nuestros proyectos:

* Power BI: es una herramienta líder en la visualización de datos empresariales. Con Power BI, podemos crear paneles interactivos y atractivos que nos permiten analizar y explorar nuestros datos de manera intuitiva. Esta herramienta nos brinda la capacidad de conectar y transformar datos provenientes de diversas fuentes, realizar análisis avanzados y generar informes y paneles personalizados.
* Streamlit: es una librería de Python que nos permite crear aplicaciones web interactivas para la presentación y compartición de nuestros modelos de aprendizaje automático. Con Streamlit, podemos desarrollar interfaces de usuario sencillas y elegantes, que permiten a los usuarios interactuar con nuestros modelos y visualizar los resultados en tiempo real. Streamlit facilita la implementación para demostrar el funcionamiento de nuestros modelos de machine learning de manera eficiente y efectiva.
* **Herramientas de trabajo :** Visual Studio Code **,** JupyterLab

Se ha elegido GitHub como nuestro repositorio de código principal. GitHub es una plataforma líder y ampliamente utilizada para el control de versiones y la colaboración en el desarrollo de software. Al aprovechar GitHub, obtendremos una serie de beneficios:

* Control de versiones: GitHub utiliza Git, un sistema de control de versiones distribuido, que nos permite mantener un historial completo de todos los cambios realizados en nuestro código. Podremos rastrear las modificaciones, revertir a versiones anteriores si es necesario y colaborar de manera efectiva en proyectos compartidos.
* Colaboración: GitHub facilita la colaboración en proyectos de desarrollo de software. Podemos invitar a otros desarrolladores a contribuir en nuestro repositorio, gestionar tareas y problemas, y revisar y aprobar cambios mediante solicitudes de extracción. La plataforma proporciona herramientas para la comunicación y el seguimiento del progreso del proyecto, lo que mejora la eficiencia y la coordinación en equipo.
* **Metodología de trabajo:**

Utilizaremos la metodología ágil Scrum para gestionar el desarrollo del proyecto. Realizaremos reuniones periódicas para revisar el progreso, definir tareas y asignar roles. Cada miembro del equipo tendrá responsabilidades específicas y trabajará en colaboración para lograr los objetivos.

* **Diseño detallado - Entregables:**

1. Análisis exploratorio de datos: Documento que detalla el proceso de limpieza y exploración de los datos.

2. Modelos estadísticos y de aprendizaje automático: Informe que describe los modelos desarrollados y los resultados obtenidos.

3. Dashboard interactivo: Presentación visual de los resultados del análisis, incluyendo gráficos y visualizaciones interactivas.

4. Informe final y recomendaciones: Documento que resume los hallazgos del análisis y proporciona recomendaciones basadas en los resultados.

* **Equipo de trabajo - Roles y responsabilidades:**

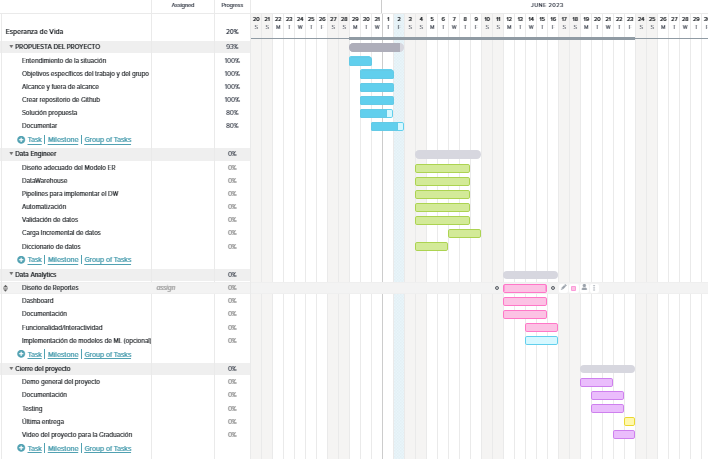
- Data Engineer: Encargado de diseñar e implementar los pipelines de ETL y configurar la infraestructura de datos.

- Data Analyst: Responsable del análisis exploratorio de datos, modelado estadístico y desarrollo de modelos predictivos.

- Functional Analyst: Asegurar de que el proyecto progrese de manera efectiva y se alcancen los objetivos establecidos al asumir diferentes roles.

* **Cronograma general - Gantt:**

Se creará un diagrama de Gantt que muestre las tareas y los plazos estimados para cada una de las etapas del proyecto. Esto permitirá un seguimiento claro del progreso y asegurará el cumplimiento de los hitos establecidos.[https://app.teamgantt.com/projects/gantt?ids=3553277]



* **Análisis preliminar de calidad de datos:**

Realizaremos un análisis detallado de los datos con los que vamos a trabajar. Analizaremos la descripción de cada columna, los tipos de datos, el método de adquisición y las fechas de adquisición y actualización.

Para la recopilación y el análisis de la data, decidimos basarnos en la base de datos brindada por el Banco Mundial y la Organización Mundial de la Salud (OMS) dado que la confianza y transparencia de su información.

**Proyecto Final: Esperanza de Vida al Nacer**





Somos una consultora especializada en análisis de datos y toma de decisiones basada en datos. Nuestro equipo está conformado por profesionales altamente capacitados y con amplia experiencia en proyectos de consultoría de datos, lo que nos permite abordar de manera integral proyectos de diversa complejidad y magnitud.

Nuestra misión es ayudar a nuestros clientes a obtener información valiosa a partir de sus datos para mejorar su toma de decisiones y optimizar sus procesos. Nos destacamos por brindar soluciones personalizadas, adaptadas a las necesidades y objetivos específicos de cada cliente. Trabajamos de manera colaborativa, estableciendo una estrecha relación con nuestros clientes para comprender a fondo su problemática y ofrecerles las mejores soluciones.

👨‍💻 **Semana 2: Data Engineering**

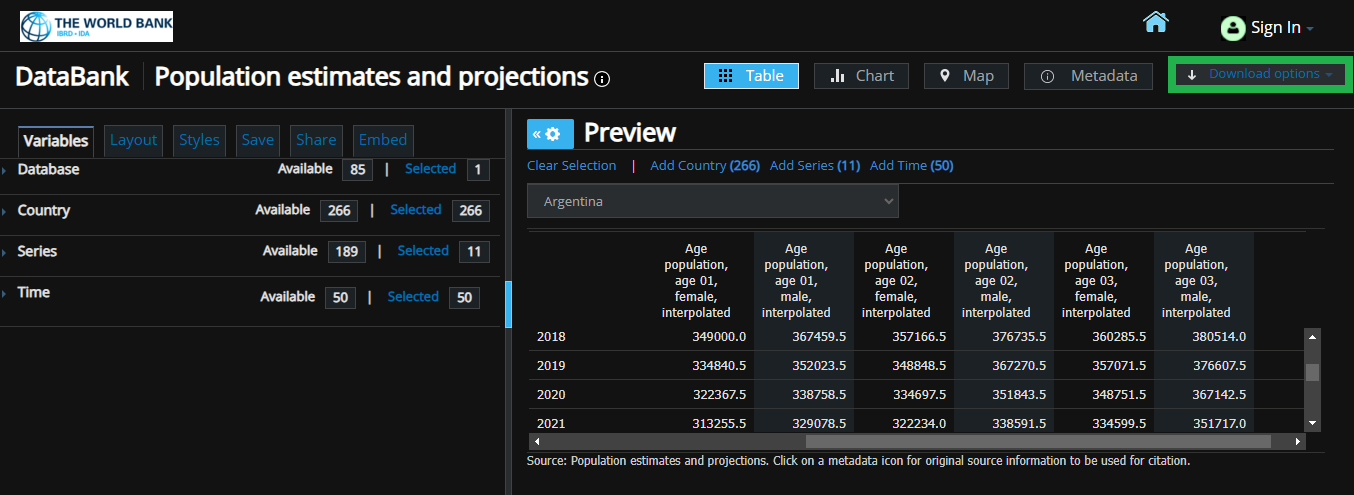
**Variedad de fuentes de datos:**

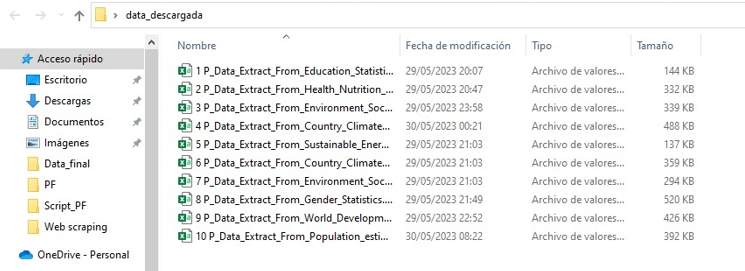
Al utilizar tanto la descarga de archivos CSV del Banco Mundial como la API de la OMS, se ha aprovechado la variedad de fuentes de datos disponibles. Esto permite obtener información diversa y enriquecedora de diferentes organismos y fuentes confiables, lo que a su vez mejora la calidad y la relevancia de los datos extraídos.

Te presentamos el proceso de obtención y extracción de datos, donde utilizamos tanto fuentes estáticas como dinámicas.

* **Data estática (.CSV)**

Desde la página del Banco Mundial, realizamos una selección de los indicadores relevantes para nuestro análisis. Aplicamos filtros según los países, años y series de datos que nos interesan, para obtener una información más específica y precisa. Una vez configurados los parámetros de búsqueda, procedemos a descargar los datos en formato CSV.



Para completar el proceso de descarga de datos, se siguió el mismo enfoque que se muestra en la imagen para los otros .CSV, obteniendo los 10 datasets:

* **Data dinámica (API)**

Paso a detallar descarga descarga de la API OMS:

Import request: para realizar solicitudes HTTP

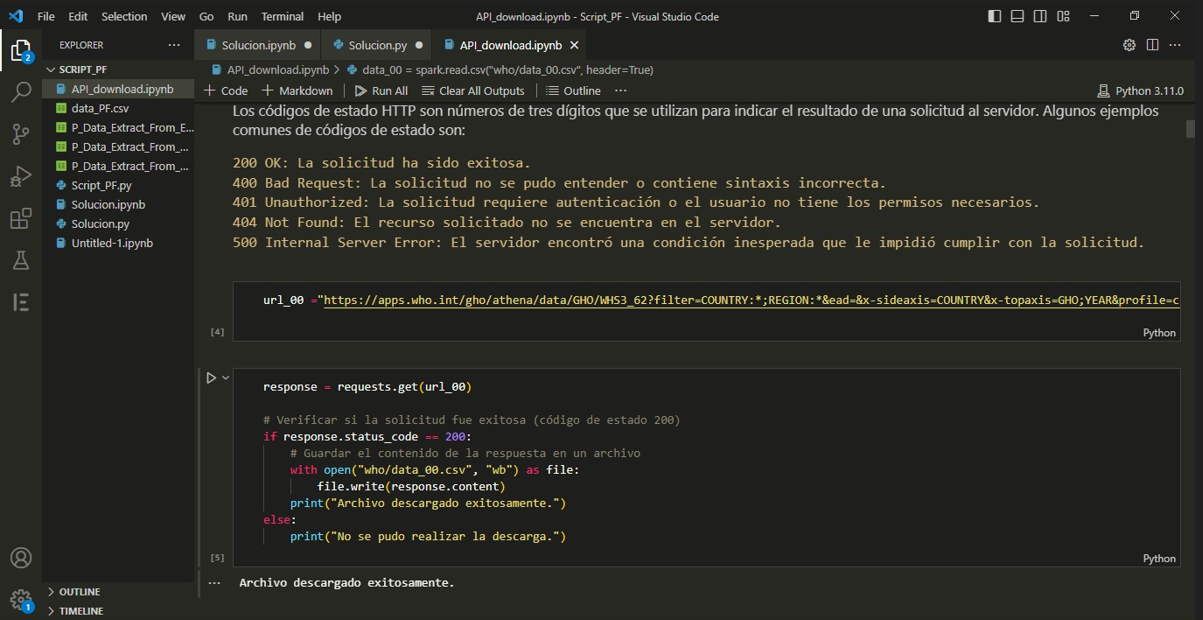
Import io: tipo E/S de texto, binario o sin formato

import csv: manipulación de archivos csv.

Importación de la librería pyspark.

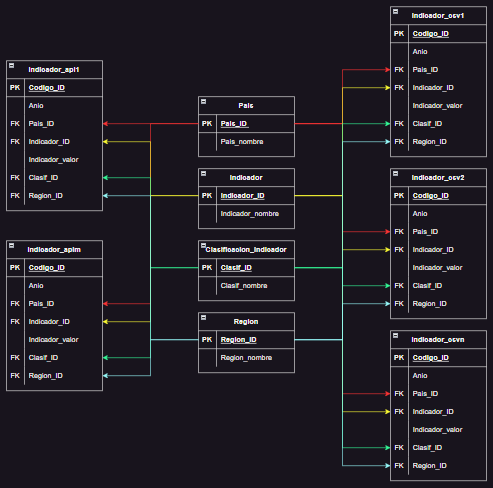
Luego, se pide respuesta a la url donde se encuentra el archivo con el indicador pedido. Se realiza la descarga. Se lee el archivo y se elimina la primera fila donde tiene anexos que no servirán al estudio.

En algunos casos se procede a transponer los datos del dataframe, teniendo como columnas los años y convirtiéndolas en dos columnas siendo una el año y a otra columna el valor del indicador.



**Diseño adecuado del Modelo ER:**

Para las estructuras de almacenamiento de datos, adoptaremos un modelo entidad-relación (ER). Diseñaremos y especificaremos las tablas, relaciones y tipos de datos necesarios para representar de manera adecuada los datos relacionados con la esperanza de vida al nacer.



**Pipelines para alimentar el Data Lake:**

Desarrollaremos pipelines de extracción, transformación y carga (ETL) para alimentar el Data Lake. Estos pipelines automatizan el proceso de integración y limpieza de los datos, asegurando la calidad de los mismos.

El primer paso será establecer la conexión con nuestro Data Lake en Azure Storage Blob utilizando las bibliotecas de Azure Storage Blobs Client. Esta conexión nos permitirá acceder y cargar los datos almacenados en el Data Lake.

A continuación, importamos los datos en formato “CSV”.

Luego aplicaremos las transformaciones necesarias para adaptarlos a nuestros requisitos. Esto incluirá procesos como limpieza de datos, filtrado, agregación y normalización. Utilizamos la biblioteca pandas para llevar a cabo estas transformaciones de manera eficiente.

Posterior a las transformaciones, procederemos a cargar los datos en nuestro Data Lake utilizando las bibliotecas de Azure Storage Blobs Client. Aseguraremos una estructura adecuada de carpetas y archivos que se ajuste a nuestras necesidades y al esquema de datos definido para nuestro Data Lake.

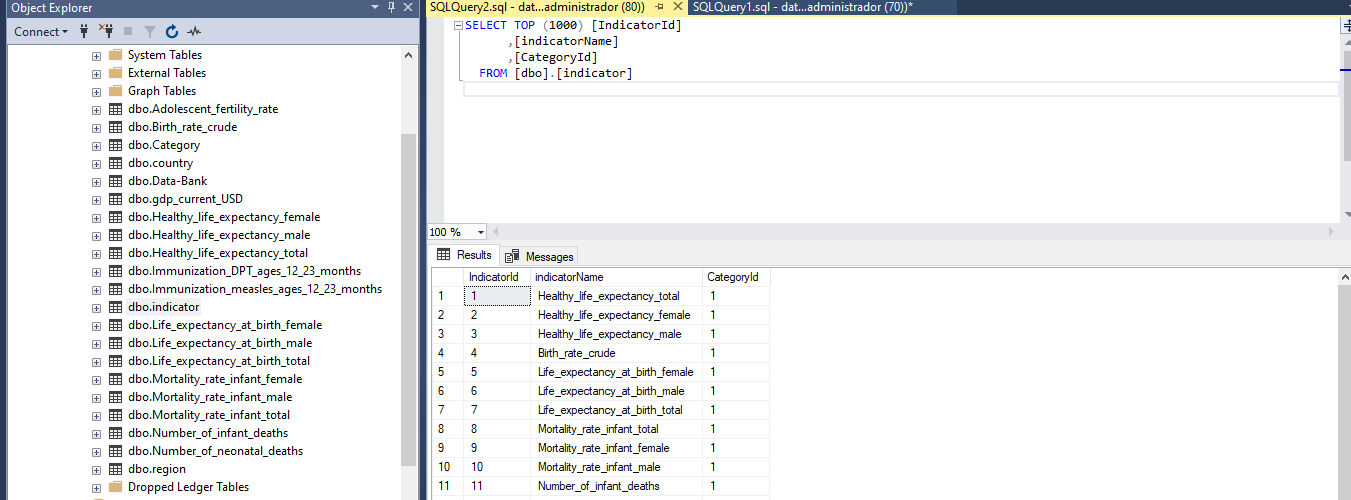
**Data Warehouse:**

Implementaremos una estructura de Data Warehouse para almacenar los datos preparados y listos para su análisis. Para automatizar el proceso de carga y creación de una tablas base de datos en Azure SQL DataBase, utilizaremos Azure Data Factory.

Azure Data Factory es una herramienta de orquestación de datos en la nube, para crear un flujo de trabajo que nos permita extraer los datos de Azure Storage Blob y cargarlos en una base de datos en Azure SQL DataBase.

Azure Data Factory nos brinda la capacidad de definir tareas de extracción, transformación y carga de datos (ETL) dentro de nuestro flujo de trabajo.

Una vez configurado el flujo de trabajo en Azure Data Factory, los datos se enviarán desde Azure Storage Blob a Azure SQL DataBase.



Posteriormente a enviar los datos a Azure SQL Database, llega el momento de mejorar la estructura de la base de datos normalizándola. Para lograr esto, creamos cuatro tablas dimensionales: "Country", "Region", "Indicadores" y "Category". Además, creamos 15 tablas de datos , una por cada indicador, con los siguientes campos: "id\_country", "id\_region", "id\_indicador", "id\_category" y el valor del indicador.

Es importante destacar que hemos implementado una clave surrogada utilizando el código del país y los dos últimos dígitos del año correspondiente. Esta clave nos proporciona un identificador único para cada registro. Las tablas dimensionales y las tablas de datos se relacionan mediante claves primarias y foráneas.

Gracias a este enfoque, hemos logrado una base de datos completamente normalizada y relacionada, lista para realizar análisis posteriormente mediante una conexión con Power BI.

**Automatización:**

La automatización de los procesos de ETL es una estrategia valiosa cuando se trata de manejar grandes volúmenes de datos que requieren actualizaciones frecuentes. Sin embargo, en el caso específico donde los datos se cargan con una frecuencia muy baja, no es necesario implementar una automatización completa por las siguientes razones:

* **Frecuencia de carga:** Si los datos se cargan solo una vez al año o con una frecuencia muy baja, el esfuerzo y el tiempo necesario para desarrollar y mantener un sistema automatizado de ETL pueden superar los beneficios obtenidos. La automatización se vuelve más relevante cuando hay actualizaciones periódicas y regulares que justifican la inversión en tiempo y recursos para configurar y mantener los procesos automatizados.
* **Complejidad de los procesos:** Los procesos de carga simples no involucran transformaciones complejas o una gran cantidad de datos, realizarlos localmente puede ser más eficiente y práctico.
* **Costo y recursos:** La implementación de una automatización completa de los procesos de ETL implica invertir en herramientas, infraestructura y recursos humanos. Si los datos se cargan con baja frecuencia, destinar recursos significativos para desarrollar y mantener la automatización puede no ser justificado desde una perspectiva de costos.

En cambio, en escenarios donde los datos se cargan con poca frecuencia, es más práctico realizar el proceso de ETL de forma manual cuando sea necesario. Esto implica dedicar el tiempo y los recursos adecuados para garantizar la calidad y la integridad de los datos durante la carga.

Es importante tener en cuenta que, aunque no se requiera una automatización completa, seguir buenas prácticas en la ejecución manual de los procesos de ETL sigue siendo esencial para garantizar la consistencia de los datos y mantener la calidad de los mismos en el Data Warehouse.

**Validación de datos:**

En el caso del proyecto, donde se utilizan múltiples fuentes de datos y se aplican transformaciones durante el proceso de extracción, transformación y carga (ETL), la validación de datos juega un papel crucial para asegurar la confiabilidad de la información analizada.

A continuación, se presentan algunos argumentos para respaldar la importancia de la validación de datos en este proyecto:

* **Consistencia de los datos:** Validar la coherencia de los valores entre diferentes fuentes y asegurar que estén dentro de los rangos esperados.
* **Integridad de los datos:** Verificar que las transformaciones aplicadas durante el proceso de ETL no introduzcan errores y que los datos resultantes sean consistentes.
* **Completitud de los datos:** Asegurarse de que no falten valores importantes y que los conjuntos de datos estén completos.
* **Detección de errores y anomalías:** Identificar valores atípicos, inconsistentes o perdidos, y aplicar reglas de negocio para garantizar la coherencia de los datos.
* **Confiabilidad del análisis:** Mejorar la calidad de los resultados y reducir el riesgo de tomar decisiones incorrectas al asegurar la precisión, completitud y consistencia de los datos utilizados.
* **Diccionario de datos:**

Elaboramos un diccionario de datos que describa el significado y la definición de cada columna en las tablas del Data Warehouse. Esto facilitará la comprensión y el análisis de los datos para todo el equipo.

| **Columna** | **Definición** |
| --- | --- |
| - `Time` | : El año o período al que corresponden los datos. |
| - `Country Name` | : El nombre del país. |
| - `Country Code` | : El código del país. |
| - `Birth rate, crude (per 1,000 people)` | : Tasa de natalidad bruta, es decir, el número de nacimientos por cada 1,000 personas. |
| - `Death rate, crude (per 1,000 people)` | : Tasa de mortalidad bruta, es decir, el número de muertes por cada 1,000 personas. |
| - `Fertility rate, total (births per woman)` | : Tasa de fertilidad total, es decir, el número promedio de hijos que una mujer tendría a lo largo de su vida. |
| - `Life expectancy at birth, female (years)` | : Esperanza de vida al nacer para mujeres en años. |
| - `Life expectancy at birth, male (years)` | : Esperanza de vida al nacer para hombres en años. |
| - `Life expectancy at birth, total (years)` | : Esperanza de vida al nacer total en años. |
| - `Mortality rate, adult, female (per 1,000 female adults)` | : Tasa de mortalidad de adultos femeninos, es decir, el número de muertes de mujeres adultas por cada 1,000 mujeres adultas. |
| - `Mortality rate, adult, male (per 1,000 male adults)` | : Tasa de mortalidad de adultos masculinos, es decir, el número de muertes de hombres adultos por cada 1,000 hombres adultos. |
| - `Mortality rate, infant (per 1,000 live births)` | : Tasa de mortalidad infantil, es decir, el número de muertes de niños menores de un año por cada 1,000 nacidos vivos. |
| - `Mortality rate, infant, female (per 1,000 live births)` | : Tasa de mortalidad infantil femenina, es decir, el número de muertes de niñas menores de un año por cada 1,000 nacidos vivos. |
| - `Mortality rate, infant, male (per 1,000 live births)` | : Tasa de mortalidad infantil masculina, es decir, el número de muertes de niños menores de un año por cada 1,000 nacidos vivos. |
| - `Mortality rate, neonatal (per 1,000 live births)` | : Tasa de mortalidad neonatal, es decir, el número de muertes de niños menores de 28 días por cada 1,000 nacidos vivos. |
| - `Population growth (annual %)` | : Tasa de crecimiento de la población anual en porcentaje. |
| - `Number of infant deaths` | : Número de muertes de niños menores de un año. |
| - `Number of neonatal deaths` | : Número de muertes de niños menores de 28 días. |
| - `Net migration` | : Migración neta, es la diferencia entre el número de inmigrantes y el número de emigrantes. |
| - `Population, female` | : Población femenina. |
| - `Population, female (% of total population)` | : Porcentaje de la población total que es femenina. |
| - `Population, male` | : Población masculina. |
| - `Population, male (% of total population)` | : Porcentaje de la población total que es masculina. |
| - `Population, total` | : Población total. |
| - `Rural population` | : Población rural. |
| - `Rural population (% of total population)` | : Porcentaje de la población total que vive en áreas rurales. |
| - `Rural population growth (annual %)` | : Tasa de crecimiento de la población rural anual en porcentaje. |
| - `Urban population` | : Población urbana. |
| - `Urban population (% of total population)` | : Porcentaje de la población total que vive en áreas urbanas. |
| - `Urban population growth (annual %)` | : Tasa de crecimiento de la población urbana anual en porcentaje. |
| - `Sex ratio at birth (male births per female births)` | : Relación de sexos al nacer, es decir, la cantidad de nacimientos masculinos por cada nacimiento femenino. |
| - `Adolescent fertility rate (births per 1,000 women ages 15-19)` | : Tasa de fertilidad adolescente, es decir, el número de nacimientos por cada 1,000 mujeres de edades entre 15 y 19 años. |
| - `GDP (current US$)` | : Producto Interno Bruto (PIB) actual en dólares estadounidenses. |
| - `GDP growth (annual %)` | : Tasa de crecimiento del Producto Interno Bruto (PIB) anual en porcentaje. |
| - `Immunization, DPT (% of children ages 12-23 months)` | : Porcentaje de niños de 12 a 23 meses vacunados contra difteria, tos ferina y tétanos. |
| - `Immunization, measles (% of children ages 12-23 months)` | : Porcentaje de niños de 12 a 23 meses vacunados contra el sarampión. |

* **Workflow de tecnologías:**

Una descripción ampliada del flujo de trabajo detallando las tecnologías utilizadas en cada etapa:

1. **Extracción de datos:**

* **Utilización de descargas de archivos CSV del Banco Mundial:** Se accede a la página del Banco Mundial y se seleccionan los indicadores relevantes. Se aplican filtros según países, años y series de datos de interés. Luego, se descargan los datos en formato CSV. Para esta etapa, se pueden utilizar tecnologías como Python (con bibliotecas como requests) para realizar solicitudes HTTP y descargar los archivos CSV.
* **Utilización de la API de la OMS:** Se utiliza la API de la OMS para obtener datos dinámicos. Se envían solicitudes HTTP a la API para obtener los indicadores deseados. La respuesta se recibe en formato CSV y se procesa para extraer los datos relevantes. Para esta etapa, se pueden utilizar tecnologías como Python (con bibliotecas como requests) para realizar solicitudes HTTP y procesar los datos CSV recibidos.

1. **Transformación de datos:**

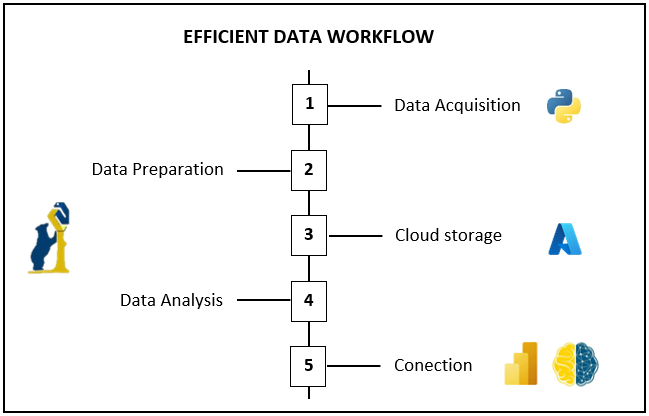
* **Limpieza y procesamiento de datos estáticos:** Una vez que se tienen los archivos CSV descargados, se procede a leerlos y realizar las transformaciones necesarias. Esto puede incluir la eliminación de filas o columnas no relevantes, la corrección de errores, la normalización de formatos, etc. Para esta etapa, se pueden utilizar tecnologías como Python (con bibliotecas como pandas) para manipular y transformar los datos en formato CSV.
* **Procesamiento de datos dinámicos:** Los datos obtenidos de la API de la OMS, en formato CSV, también requieren transformación. Se extraen los datos relevantes del CSV y se realiza cualquier procesamiento adicional necesario, como la transposición de los datos para tener años como columnas y valores de indicadores en filas. Para esta etapa, se pueden utilizar tecnologías como Python (con bibliotecas como pandas) para manipular y transformar los datos.

1. **Almacenamiento de datos:**

* **Data Lake**: Se utiliza Azure Storage Blob como Data Lake para almacenar los datos extraídos y transformados. Se puede acceder a Azure Storage Blob mediante bibliotecas como Azure Storage Blobs Client en Python para cargar los datos en formato CSV.
* **Data Warehouse:** Se implementa un Data Warehouse utilizando Azure SQL DataBase para almacenar los datos preparados y listos para su análisis. Se utiliza Azure Data Factory para automatizar el proceso de carga y creación de tablas en Azure SQL DataBase. Azure Data Factory permite definir tareas de extracción, transformación y carga de datos (ETL) y transferir los datos desde Azure Storage Blob a Azure SQL DataBase.

1. **Análisis de datos:**

* **Exploración de datos:** Una vez que los datos se encuentran en el Data Warehouse, se pueden realizar análisis exploratorios utilizando herramientas como Azure Synapse Analytics, que proporciona capacidades de análisis y consulta avanzadas.



👨‍💻 **Semana 2: Data Engineering**

Como equipo, hemos decidido utilizar Visual Studio Code y Jupyter Lab como nuestras herramientas de desarrollo integrado (IDE, por sus siglas en inglés) para trabajar en nuestros proyectos.

* Visual Studio Code es un IDE altamente versátil y ampliamente utilizado, con una amplia gama de extensiones y complementos disponibles.
* Jupyter Lab, por otro lado, es una poderosa herramienta para el análisis interactivo de datos y la creación de cuadernos (notebooks). Proporciona un entorno flexible y colaborativo que combina código, visualizaciones y documentación en un solo lugar.

**Variedad de fuentes de datos:**

Al utilizar tanto la descarga de archivos CSV de la OPS y como la librería wbgapi que nos brinda la data API del Banco Mundial, se ha aprovechado la variedad de fuentes de datos disponibles. Esto permite obtener información diversa y enriquecedora de diferentes organismos y fuentes confiables, lo que a su vez mejora la calidad y la relevancia de los datos extraídos.

Te presentamos el proceso de obtención y extracción de datos, donde utilizamos tanto fuentes estáticas como dinámicas.

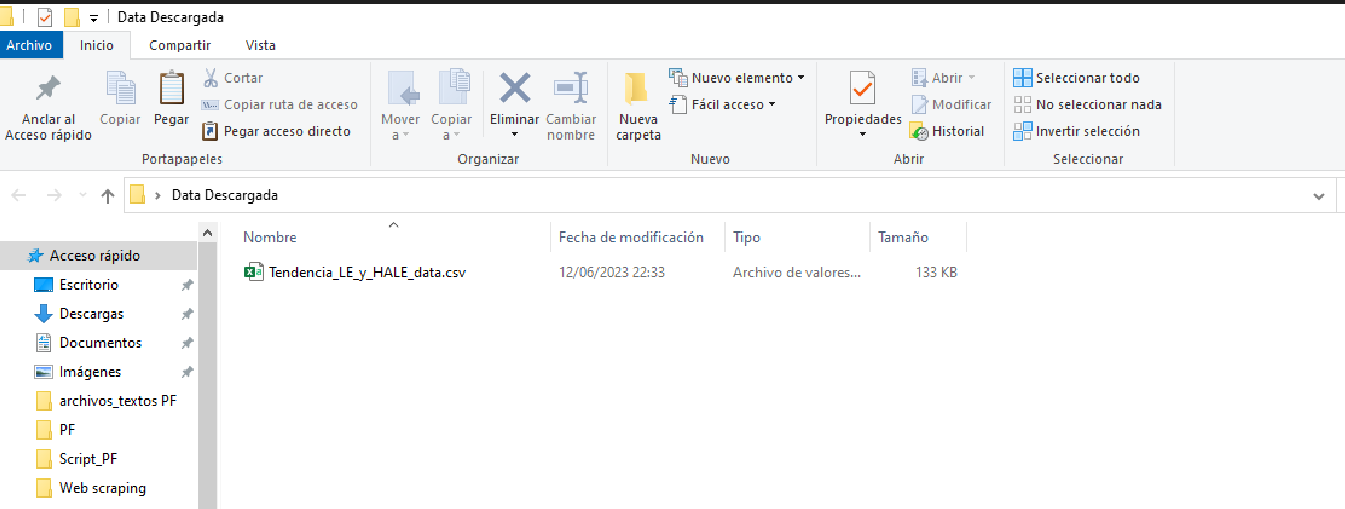
* **Data estática (.CSV)**

Tomando en consideración que ésta data es una data complementaria.

Desde la página de la OPS, realizamos una selección de los indicadores relevantes para nuestro análisis. Aplicamos filtros según los países, años y series de datos que nos interesan, para obtener una información más específica y precisa. Una vez configurados los parámetros de búsqueda, procedemos a descargar los datos en formato CSV.



Para completar el proceso de descarga de datos, se siguió el mismo enfoque que se muestra en la imagen, obteniendo 1 dataset con varios indicadores que son de nuestro interés.



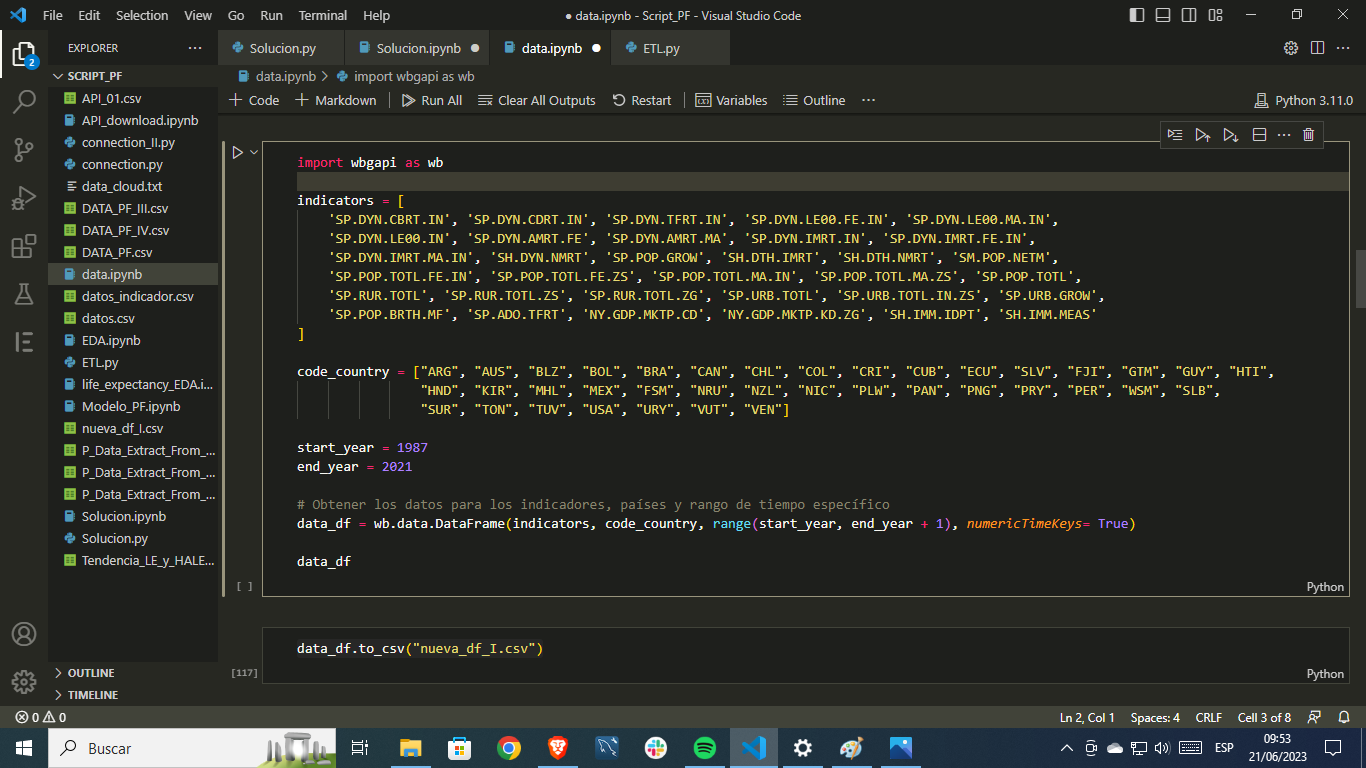
* **Data dinámica (API)**

Paso a detallar descarga descarga de la API del Banco mundial mediante la librería wbgapi:

Import wbgapi: librería que contiene la data de la API del Banco Mundial

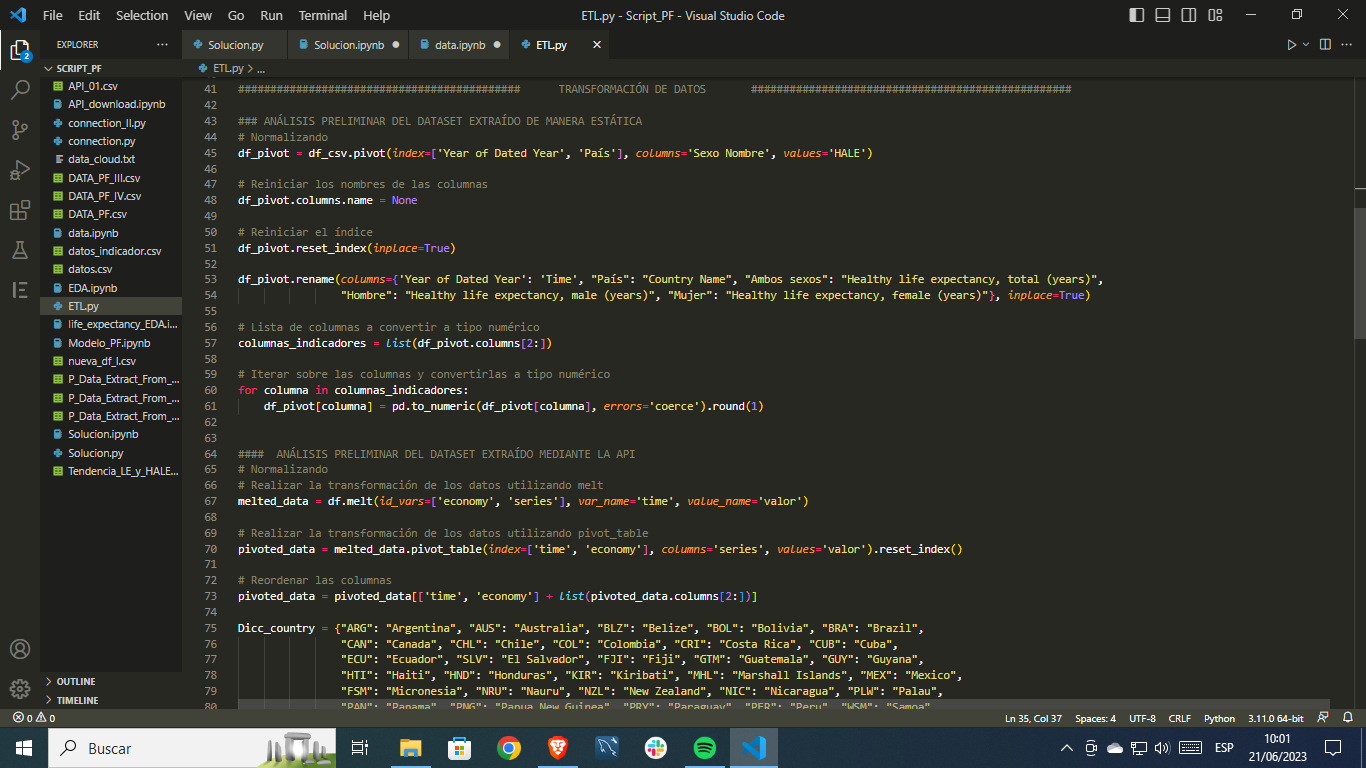
Luego, se crea una lista con los códigos de los indicadores que se va a trabajar, del mismo modo se hace con los países y finalmente marcamos el rango de los años que nos interesa.

Por último guardamos en un csv la data extraída de la API.

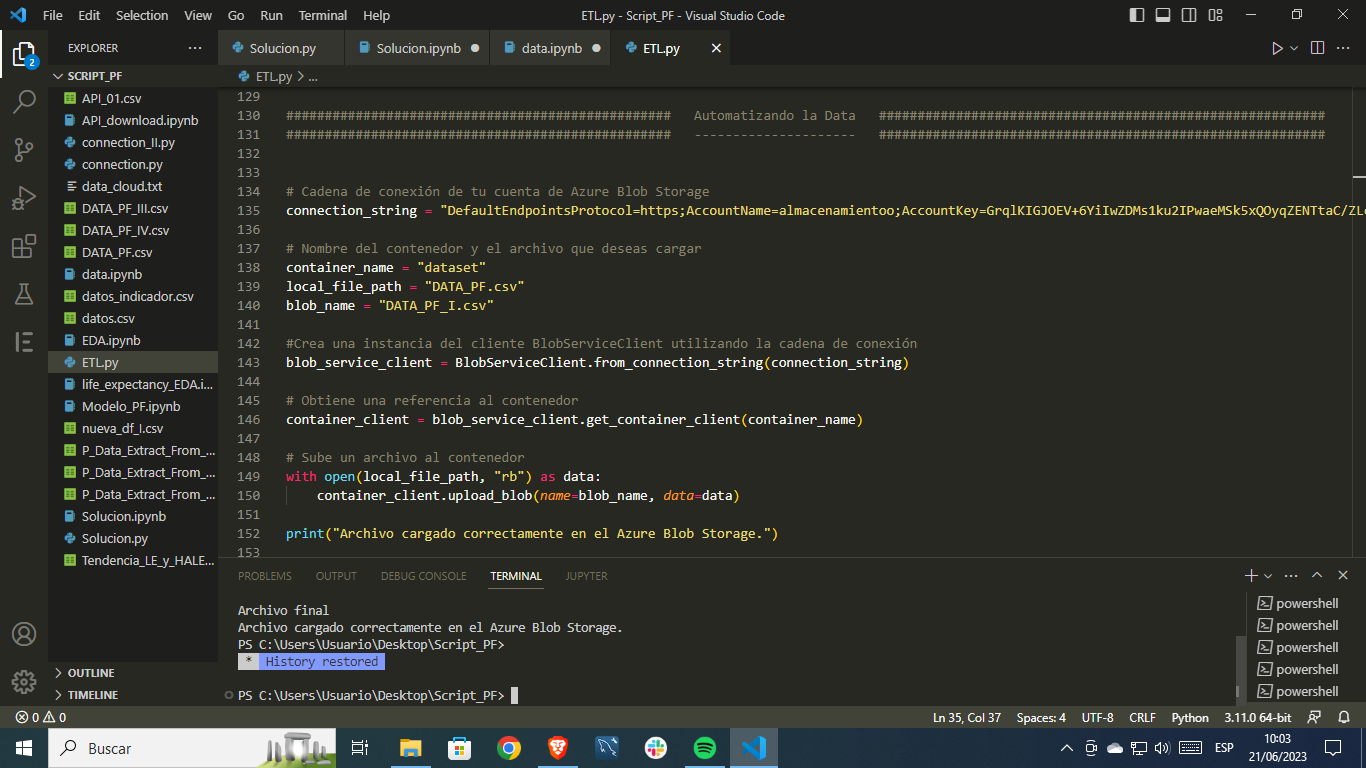


* **ETL**

Una vez extraída la data tanto de manera estática como dinámica, realizamos el proceso de transformación de los datos de cada dataset extraída de tal manera que logramos consolidar los datos en un solo dataset final.



Y de manera final lo que se hace es enviar el csv consolidado al servidor de Azure, lo cuál de esta manera generamos una automatización de la data en un sistema de almacenamiento.



**Pipelines para alimentar el Data Lake:**

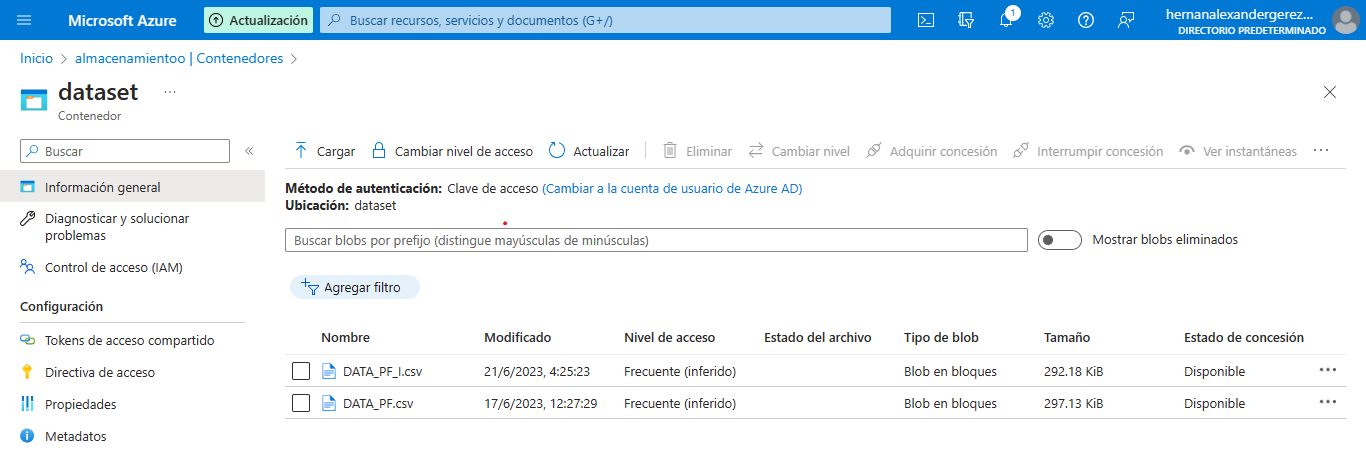
Desarrollaremos pipelines de extracción, transformación y carga (ETL) para alimentar el Data Lake. Estos pipelines automatizan el proceso de integración y limpieza de los datos, asegurando la calidad de los mismos.

El primer paso será establecer la conexión con nuestro Data Lake en Azure Storage Blob utilizando las bibliotecas de Azure Storage Blobs Client. Esta conexión nos permitirá acceder y cargar los datos almacenados en el Data Lake.

A continuación, importamos los datos en formato “CSV”.

Luego aplicaremos las transformaciones necesarias para adaptarlos a nuestros requisitos. Esto incluirá procesos como limpieza de datos, filtrado, agregación y normalización. Utilizamos la biblioteca pandas para llevar a cabo estas transformaciones de manera eficiente.

Posterior a las transformaciones, procederemos a cargar los datos en nuestro Data Lake utilizando las bibliotecas de Azure Storage Blobs Client. Aseguraremos una estructura adecuada de carpetas y archivos que se ajuste a nuestras necesidades y al esquema de datos definido para nuestro Data Lake.



**Data Warehouse:**

Implementaremos una estructura de Data Warehouse para almacenar los datos preparados y listos para su análisis. Para automatizar el proceso de carga y creación de una tablas base de datos en Azure SQL DataBase, utilizaremos Azure Data Factory.

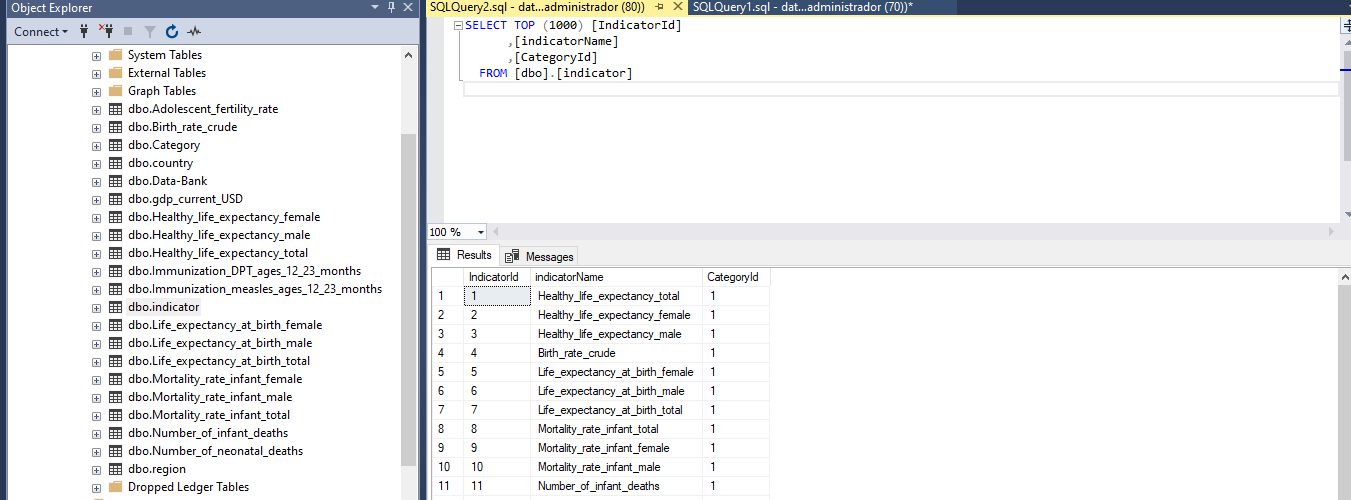
Azure Data Factory es una herramienta de orquestación de datos en la nube, para crear un flujo de trabajo que nos permita extraer los datos de Azure Storage Blob y cargarlos en una base de datos en Azure SQL DataBase.

Azure Data Factory nos brinda la capacidad de definir tareas de extracción, transformación y carga de datos (ETL) dentro de nuestro flujo de trabajo.

Una vez configurado el flujo de trabajo en Azure Data Factory, los datos se enviarán desde Azure Storage Blob a Azure SQL DataBase.

Azure SQL Database es un servicio de base de datos relacional totalmente administrado ofrecido por Microsoft Azure. Proporciona una plataforma segura y escalable para almacenar, administrar y acceder a datos estructurados en la nube.

* Administración simplificada: Azure SQL Database se encarga de las tareas de administración y mantenimiento de la base de datos.
* Escalabilidad y elasticidad: Azure SQL Database permite escalar vertical y horizontalmente para satisfacer las necesidades de rendimiento
* Integración con servicios de Azure: Azure SQL Database se integra con otros servicios de Azure



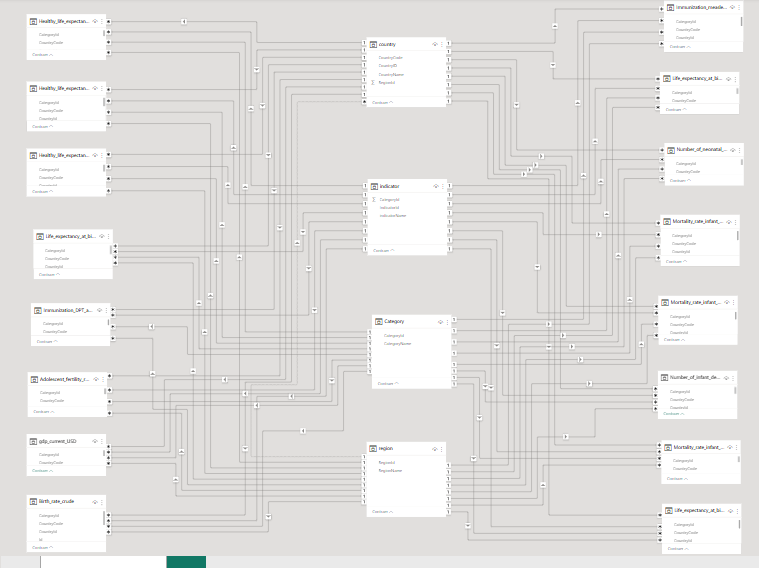
**Diseño adecuado del Modelo ER:**

Para las estructuras de almacenamiento de datos, adoptaremos un modelo entidad-relación (ER). Diseñaremos y especificaremos las tablas, relaciones y tipos de datos necesarios para representar de manera adecuada los datos relacionados con la esperanza de vida al nacer

Posteriormente a enviar los datos a Azure SQL Database, llega el momento de mejorar la estructura de la base de datos normalizándola. Para lograr esto, creamos cuatro tablas dimensionales: "Country", "Region", "Indicator" y "Category". Además, creamos 16 tablas de datos , una por cada indicador, con los siguientes campos: "Id\_Country", "Id\_Region", "Id\_Indicator", "Id\_Category" y el valor del indicador.

Es importante destacar que hemos implementado una clave subrogada utilizando el código del país y los dos últimos dígitos del año correspondiente. Esta clave nos proporciona un identificador único para cada registro. Las tablas dimensionales y las tablas de datos se relacionan mediante claves primarias y foráneas.

Gracias a este enfoque, hemos logrado una base de datos completamente normalizada y relacionada, lista para realizar análisis posteriormente mediante una conexión con Power BI.

****

**Automatización:**

La automatización de los procesos de ETL es una estrategia valiosa cuando se trata de manejar grandes volúmenes de datos que requieren actualizaciones frecuentes. Sin embargo, en el caso específico donde los datos se cargan con una frecuencia muy baja, no es necesario implementar una automatización completa por las siguientes razones:

* **Frecuencia de carga:** Si los datos se cargan solo una vez al año o con una frecuencia muy baja, el esfuerzo y el tiempo necesario para desarrollar y mantener un sistema automatizado de ETL pueden superar los beneficios obtenidos. La automatización se vuelve más relevante cuando hay actualizaciones periódicas y regulares que justifican la inversión en tiempo y recursos para configurar y mantener los procesos automatizados.
* **Complejidad de los procesos:** Los procesos de carga simples no involucran transformaciones complejas o una gran cantidad de datos, realizarlos localmente puede ser más eficiente y práctico.
* **Costo y recursos:** La implementación de una automatización completa de los procesos de ETL implica invertir en herramientas, infraestructura y recursos humanos. Si los datos se cargan con baja frecuencia, destinar recursos significativos para desarrollar y mantener la automatización puede no ser justificado desde una perspectiva de costos.

En cambio, en escenarios donde los datos se cargan con poca frecuencia, es más práctico realizar el proceso de ETL de forma manual cuando sea necesario. Esto implica dedicar el tiempo y los recursos adecuados para garantizar la calidad y la integridad de los datos durante la carga.

Es importante tener en cuenta que, aunque no se requiera una automatización completa, seguir buenas prácticas en la ejecución manual de los procesos de ETL sigue siendo esencial para garantizar la consistencia de los datos y mantener la calidad de los mismos en el Data Warehouse.

**Validación de datos:**

En el caso del proyecto, donde se utilizan múltiples fuentes de datos y se aplican transformaciones durante el proceso de extracción, transformación y carga (ETL), la validación de datos juega un papel crucial para asegurar la confiabilidad de la información analizada.

A continuación, se presentan algunos argumentos para respaldar la importancia de la validación de datos en este proyecto:

* **Consistencia de los datos:** Validar la coherencia de los valores entre diferentes fuentes y asegurar que estén dentro de los rangos esperados.
* **Integridad de los datos:** Verificar que las transformaciones aplicadas durante el proceso de ETL no introduzcan errores y que los datos resultantes sean consistentes.
* **Completitud de los datos:** Asegurarse de que no falten valores importantes y que los conjuntos de datos estén completos.
* **Detección de errores y anomalías:** Identificar valores atípicos, inconsistentes o perdidos, y aplicar reglas de negocio para garantizar la coherencia de los datos.
* **Confiabilidad del análisis:** Mejorar la calidad de los resultados y reducir el riesgo de tomar decisiones incorrectas al asegurar la precisión, completitud y consistencia de los datos utilizados.
* **Diccionario de datos:**

Elaboramos un diccionario de datos que describa el significado y la definición de cada columna en las tablas del Data Warehouse. Esto facilitará la comprensión y el análisis de los datos para todo el equipo.

| **Columna** | **Definición** |
| --- | --- |
| - `Time` | : El año o período al que corresponden los datos. |
| - `Country Name` | : El nombre del país. |
| - `Country Code` | : El código del país. |
| - `Birth rate, crude (per 1,000 people)` | : Tasa de natalidad bruta, es decir, el número de nacimientos por cada 1,000 personas. |
| - `Death rate, crude (per 1,000 people)` | : Tasa de mortalidad bruta, es decir, el número de muertes por cada 1,000 personas. |
| - `Fertility rate, total (births per woman)` | : Tasa de fertilidad total, es decir, el número promedio de hijos que una mujer tendría a lo largo de su vida. |
| - `Life expectancy at birth, female (years)` | : Esperanza de vida al nacer para mujeres en años. |
| - `Life expectancy at birth, male (years)` | : Esperanza de vida al nacer para hombres en años. |
| - `Life expectancy at birth, total (years)` | : Esperanza de vida al nacer total en años. |
| - `Mortality rate, adult, female (per 1,000 female adults)` | : Tasa de mortalidad de adultos femeninos, es decir, el número de muertes de mujeres adultas por cada 1,000 mujeres adultas. |
| - `Mortality rate, adult, male (per 1,000 male adults)` | : Tasa de mortalidad de adultos masculinos, es decir, el número de muertes de hombres adultos por cada 1,000 hombres adultos. |
| - `Mortality rate, infant (per 1,000 live births)` | : Tasa de mortalidad infantil, es decir, el número de muertes de niños menores de un año por cada 1,000 nacidos vivos. |
| - `Mortality rate, infant, female (per 1,000 live births)` | : Tasa de mortalidad infantil femenina, es decir, el número de muertes de niñas menores de un año por cada 1,000 nacidos vivos. |
| - `Mortality rate, infant, male (per 1,000 live births)` | : Tasa de mortalidad infantil masculina, es decir, el número de muertes de niños menores de un año por cada 1,000 nacidos vivos. |
| - `Mortality rate, neonatal (per 1,000 live births)` | : Tasa de mortalidad neonatal, es decir, el número de muertes de niños menores de 28 días por cada 1,000 nacidos vivos. |
| - `Population growth (annual %)` | : Tasa de crecimiento de la población anual en porcentaje. |
| - `Number of infant deaths` | : Número de muertes de niños menores de un año. |
| - `Number of neonatal deaths` | : Número de muertes de niños menores de 28 días. |
| - `Net migration` | : Migración neta, es la diferencia entre el número de inmigrantes y el número de emigrantes. |
| - `Population, female` | : Población femenina. |
| - `Population, female (% of total population)` | : Porcentaje de la población total que es femenina. |
| - `Population, male` | : Población masculina. |
| - `Population, male (% of total population)` | : Porcentaje de la población total que es masculina. |
| - `Population, total` | : Población total. |
| - `Rural population` | : Población rural. |
| - `Rural population (% of total population)` | : Porcentaje de la población total que vive en áreas rurales. |
| - `Rural population growth (annual %)` | : Tasa de crecimiento de la población rural anual en porcentaje. |
| - `Urban population` | : Población urbana. |
| - `Urban population (% of total population)` | : Porcentaje de la población total que vive en áreas urbanas. |
| - `Urban population growth (annual %)` | : Tasa de crecimiento de la población urbana anual en porcentaje. |
| - `Sex ratio at birth (male births per female births)` | : Relación de sexos al nacer, es decir, la cantidad de nacimientos masculinos por cada nacimiento femenino. |
| - `Adolescent fertility rate (births per 1,000 women ages 15-19)` | : Tasa de fertilidad adolescente, es decir, el número de nacimientos por cada 1,000 mujeres de edades entre 15 y 19 años. |
| - `GDP (current US$)` | : Producto Interno Bruto (PIB) actual en dólares estadounidenses. |
| - `GDP growth (annual %)` | : Tasa de crecimiento del Producto Interno Bruto (PIB) anual en porcentaje. |
| - `Immunization, DPT (% of children ages 12-23 months)` | : Porcentaje de niños de 12 a 23 meses vacunados contra difteria, tos ferina y tétanos. |
| - `Immunization, measles (% of children ages 12-23 months)` | : Porcentaje de niños de 12 a 23 meses vacunados contra el sarampión. |

* **Workflow de tecnologías:**

Una descripción ampliada del flujo de trabajo detallando las tecnologías utilizadas en cada etapa:

**Extracción de datos:**

* **Utilización de descargas de archivos CSV del Banco Mundial:** Se accede a la página del Banco Mundial y se seleccionan los indicadores relevantes. Se aplican filtros según países, años y series de datos de interés. Luego, se descargan los datos en formato CSV. Para esta etapa, se pueden utilizar tecnologías como Python (con bibliotecas como requests) para realizar solicitudes HTTP y descargar los archivos CSV.
* **Utilización de la API de la OMS:** Se utiliza la API de la OMS para obtener datos dinámicos. Se envían solicitudes HTTP a la API para obtener los indicadores deseados. La respuesta se recibe en formato CSV y se procesa para extraer los datos relevantes. Para esta etapa, se pueden utilizar tecnologías como Python (con bibliotecas como requests) para realizar solicitudes HTTP y procesar los datos CSV recibidos.

**Transformación de datos:**

* **Limpieza y procesamiento de datos estáticos:** Una vez que se tienen los archivos CSV descargados, se procede a leerlos y realizar las transformaciones necesarias. Esto puede incluir la eliminación de filas o columnas no relevantes, la corrección de errores, la normalización de formatos, etc. Para esta etapa, se pueden utilizar tecnologías como Python (con bibliotecas como pandas) para manipular y transformar los datos en formato CSV.
* **Procesamiento de datos dinámicos:** Los datos obtenidos de la API de la OMS, en formato CSV, también requieren transformación. Se extraen los datos relevantes del CSV y se realiza cualquier procesamiento adicional necesario, como la transposición de los datos para tener años como columnas y valores de indicadores en filas. Para esta etapa, se pueden utilizar tecnologías como Python (con bibliotecas como pandas) para manipular y transformar los datos.

**Almacenamiento de datos:**

Implementamos nuestro Data Lake y Data Warehouse en la nube de Azure debido a las numerosas facilidades y características que ofrece. Azure nos proporciona una potencia de cómputo y almacenamiento suficiente, sin depender de una máquina local.

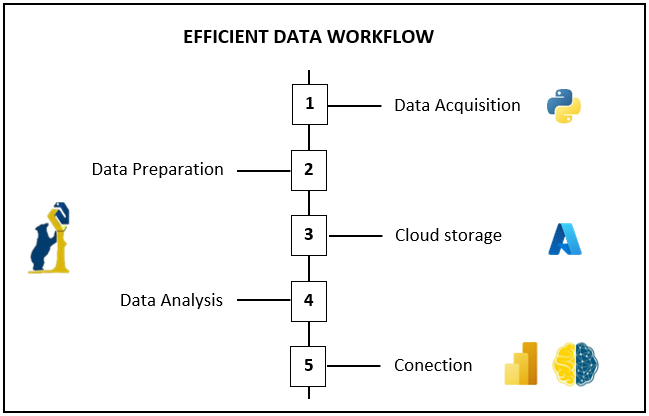
Además, hemos aprovechado el bono gratuito que se otorga al registrarnos.

Esta elección nos brinda múltiples beneficios. En primer lugar, Azure cuenta con una amplia gama de servicios y funcionalidades que facilitan el almacenamiento y procesamiento de datos. En segundo lugar, la capa gratuita de Azure nos permite comenzar sin incurrir en costos adicionales.

* **Data Lake**: Se utiliza Azure Storage Blob como Data Lake para almacenar los datos extraídos y transformados. Se puede acceder a Azure Storage Blob mediante bibliotecas como Azure Storage Blobs Client en Python para cargar los datos en formato CSV.
* **Data Warehouse:** Se implementa un Data Warehouse utilizando Azure SQL DataBase para almacenar los datos preparados y listos para su análisis. Se utiliza Azure Data Factory para automatizar el proceso de carga y creación de tablas en Azure SQL DataBase. Azure Data Factory permite definir tareas de extracción, transformación y carga de datos (ETL) y transferir los datos desde Azure Storage Blob a Azure SQL DataBase.

**Análisis de datos:**

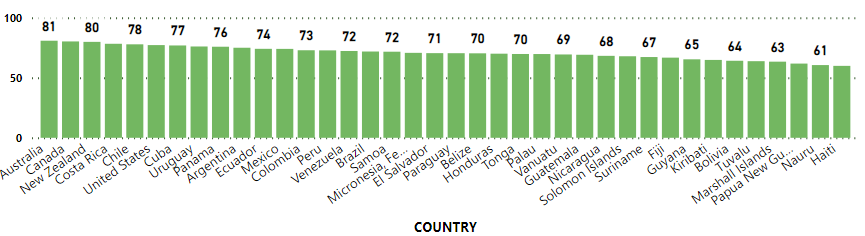
* **Exploración de datos:** Una vez que los datos se encuentran en el Data Warehouse, se pueden realizar análisis exploratorios utilizando herramientas como Azure SQL DataBase que proporciona capacidades de análisis y consulta avanzadas.



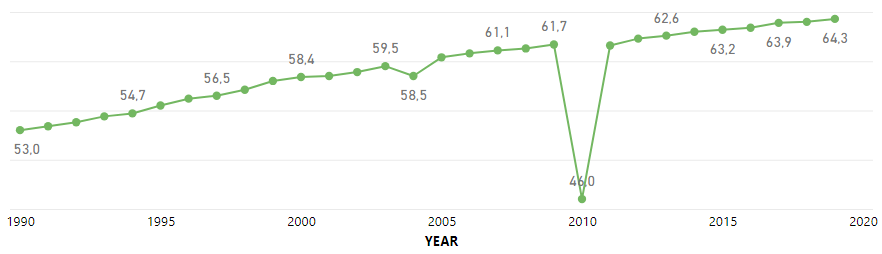
**📈 Semana 3: Data Analytics + ML**

**Diseño de Reportes/Dashboards:**

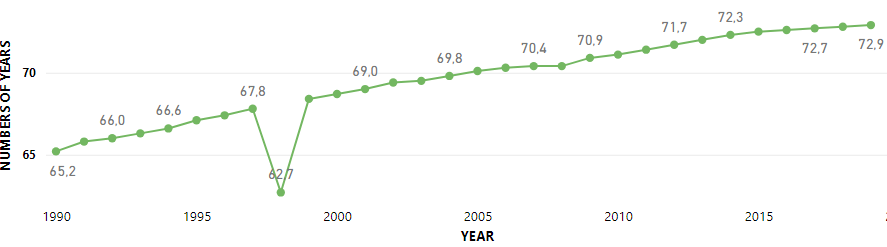
Siguiendo con el análisis de datos de nuestro proyecto, al momento de visualizar y tomar las mejores decisiones, y plasmar toda la información recopilada, como consultora decidimos, en primera instancia, analizar el comportamiento de la esperanza de vida en países de América y Oceanía. Optamos por este alcance debido al notable contraste entre los países de Centroamérica y las islas de Oceanía en comparación con Norteamérica y países como Australia. En el caso de Norteamérica (Canadá y Estados Unidos), la esperanza de vida ha aumentado gradualmente en los últimos 30 años. Es importante destacar que, en general, las mujeres tienen una mayor esperanza de vida promedio que los hombres, debido a diversas razones. Canadá lidera el ranking en este sentido, con una esperanza de vida general de 80 años, siendo las mujeres las que tienen una mayor esperanza de vida al nacer (84 años).



En el caso de Centroamérica, ocurre algo similar a las pequeñas islas de Oceanía, y se destaca Haití como el país con el promedio más bajo. Esto se debe al terremoto ocurrido en 2010 en la capital haitiana de Puerto Príncipe, que dejó un saldo de más de 300,000 víctimas y 350,000 heridos, según datos de la ONU. Este desastre impacta notablemente en nuestro indicador a analizar.



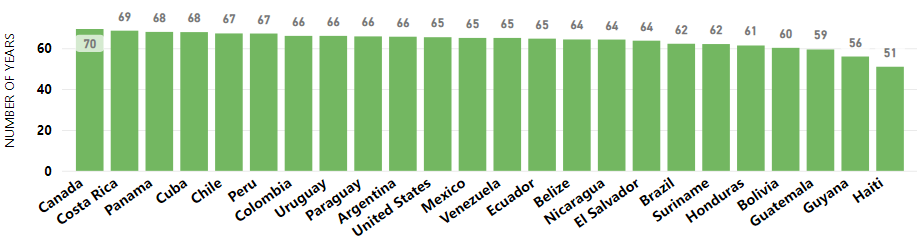
Otro punto destacado que encontramos en nuestro análisis y consideramos importante fue en 1998, en el país de Honduras, donde se registró un pico de 5.1 puntos. Esto se produjo debido al huracán Mitch, que causó aproximadamente 19,000 víctimas y dejó daños devastadores en la sociedad hondureña.



En el caso de Sudamérica y otros países, no encontramos variaciones significativas a lo largo del tiempo. Cabe señalar que estos países no tienen el mismo nivel de esperanza de vida al nacer que países como Canadá o Australia, pero en los últimos años se ha observado un crecimiento gradual.

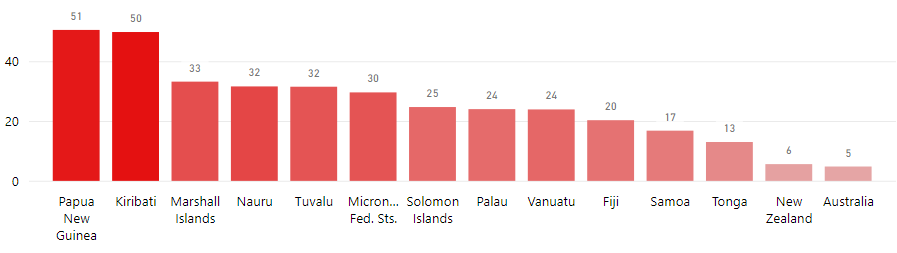
El objetivo de estudiar el comportamiento de este indicador a lo largo de los años es lograr un crecimiento significativo en los próximos años.

En cuanto a la esperanza de vida saludable, obtuvimos datos completos y concisos de los países del continente americano. Si bien no notamos cambios significativos a lo largo del tiempo, es importante destacar que Canadá representa una esperanza de vida saludable de 69 años, lo cual lo posiciona como el país con la mejor esperanza de vida saludable. Por otro lado, Haití ocupa el último puesto en el ranking de esperanza de vida saludable. Como objetivo, planteamos incrementar la cantidad de años de esperanza de vida saludable en un 1% en relación al último año.

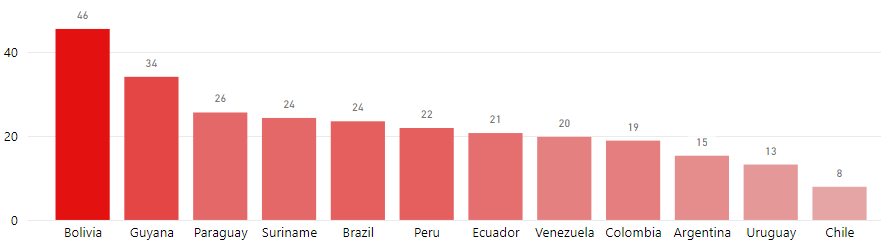


El tercer objetivo se centra en la tasa de mortalidad infantil, para observar y analizar su evolución en los últimos 30 años. En este análisis, hemos optado por mencionar los países con la mayor y menor tasa, y observar su evolución a lo largo del tiempo.

En el caso de Oceanía, se observa un gran contraste entre países como Papúa Nueva Guinea y Australia. La primera presenta una media de 50.50, mientras que la segunda tiene una tasa de 4.80. Aunque en los últimos años se registra una disminución en este valor, sigue siendo sorprendente la diferencia entre ambos países en estudio.



Continuando con el estudio, en Sudamérica, Bolivia lidera este análisis. Se estima que enfermedades como la neumonía, enfermedades diarreicas y distintas dificultades al nacer son las principales causas de esta tasa, junto con factores socioeconómicos y culturales.



En Centroamérica, Haití ocupa el primer puesto con una media de 68.20 muertes por cada mil niños, debido a los factores mencionados anteriormente.

Finalmente, no podemos dejar de mencionar a dos potencias como Canadá y Estados Unidos, que tienen la menor tasa de mortalidad infantil (5.30 en Canadá y 6.80 en Estados Unidos). Este dato es muy relevante para quienes consideran tener hijos.

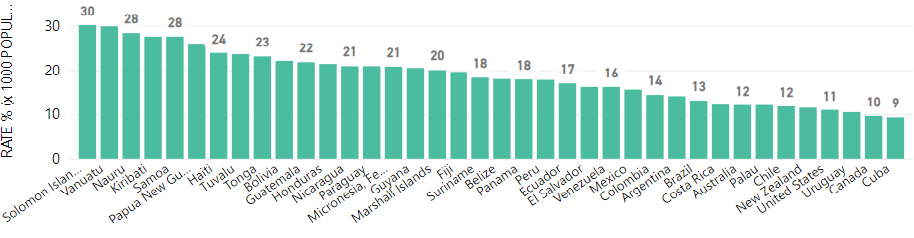
En resumen, el objetivo planteado es reducir en un 1% la tasa de mortalidad en aquellos países con una alta tasa, lo que implica una disminución de la tasa analizada con respecto al último año.

Por otro lado, se decidió analizar los países con la menor tasa de natalidad en los últimos años. En América del Sur, no se observan grandes diferencias entre los países en este aspecto.

En el caso de Centro y Norte América, es importante destacar países como Cuba y Canadá. Aunque las circunstancias son diversas, se observa un gran contraste en relación a la calidad de vida entre ambos. La tasa de natalidad en Cuba es de 11.80 por cada mil habitantes, mientras que en Canadá es de 11.00 por cada mil habitantes. En el caso de Cuba, esto se debe a factores socioeconómicos y a la constante migración.

Finalmente, en Oceanía, Australia lidera este ranking como el país con la menor tasa de natalidad. Australia es un país altamente desarrollado y con una alta calidad de esperanza de vida.

El objetivo de este estudio es incrementar la tasa de fecundidad en aquellos países que tienen una tasa muy baja en un 1% con respecto al año 2019.



**KPIs:**

Identificaremos los indicadores clave de rendimiento (KPIs) que sean relevantes para evaluar el cumplimiento de los objetivos planteados en la propuesta del proyecto. Estos KPIs ayudarán a medir y monitorear el progreso y los resultados del análisis de datos.

* **Incremento esperanza de vida 1% =**

**[**VAR MinValue = MIN('Life\_expectancy\_at\_birth\_total'[Value])

VAR MinCountry = CALCULATE(FIRSTNONBLANK('country'[CountryName], 1), 'Life\_expectancy\_at\_birth\_total'[Value] = MinValue)

RETURN MinValue \* 1.01]

* **Incremento 1% vida saludable =**

[VAR MinValue = MIN('Healthy\_life\_expectancy\_total'[Value])

VAR MinCountry = CALCULATE(FIRSTNONBLANK(country[CountryName], 1), 'Healthy\_life\_expectancy\_total'[Value] = MinValue)

RETURN MinValue \* 1.01]

* **Disminución 1% País Mayor =**

[VAR MaxValue = MAX('Mortality\_rate\_infant\_total'[Value])

VAR MaxCountry = CALCULATE(FIRSTNONBLANK('country'[CountryName], 1), 'Mortality\_rate\_infant\_total'[Value] = MaxValue)

RETURN MaxValue \* 0.99]

* **Aumento 1% País Menor =**

VAR MinValue = MIN('Birth\_rate\_crude'[Value])

VAR MinCountry = CALCULATE(FIRSTNONBLANK('country'[CountryName], 1), 'Birth\_rate\_crude'[Value] = MinValue)

RETURN MinValue \* 1.01

**Modelo ML:**

**Modelo KNN**

La idea inicial fue buscar los indicadores más influyentes para el aumento de la esperanza de vida. Y entregarlos al segundo modelo para que haga una predicción de la esperanza de vida en un futuro cercano.

El modelo KNN (K-Nearest Neighbors) es un algoritmo de aprendizaje supervisado que se utiliza para clasificar o predecir valores numéricos. Se basa en encontrar los K ejemplos más cercanos a un nuevo dato y tomar la clase mayoritaria o promediar sus valores para hacer una predicción. Es simple y fácil de entender, pero puede tener dificultades con conjuntos de datos grandes y dimensionales.

El código del Modelo KNN, realiza los siguientes pasos:

* Definición de características: Se seleccionan las características que se utilizarán en el modelo KNN para encontrar los indicadores más influyentes en la esperanza de vida al nacer.
* Imputación de valores faltantes: Se aplica la imputación para manejar los valores faltantes en los datos, utilizando la estrategia de la media.
* Creación del modelo KNN: Se crea un modelo KNN utilizando la clase NearestNeighbors de sklearn. El modelo se ajusta a los datos imputados.
* Función para obtener recomendaciones: Se define una función llamada "get\_recommendations" que toma un índice de país y devuelve los índices de los indicadores más cercanos recomendados por el modelo KNN.
* Mostrar registros de datos: El usuario tiene la opción de mostrar los primeros 5 o los últimos 5 registros del conjunto de datos.
* Subtítulo - Modelo KNN: Se muestra un subtítulo en la interfaz para indicar que se va a generar los indicadores más influyentes utilizando el modelo KNN.
* Generación de indicadores más influyentes: Al hacer clic en el botón correspondiente, se genera una lista de todos los indicadores más influyentes para cada país en el conjunto de datos utilizando el modelo KNN. Se cuentan las frecuencias de los indicadores recomendados.
* Obtener los indicadores más influyentes: Se seleccionan los siete indicadores más comunes y se muestran en la interfaz.

En resumen, el código carga los datos, ajusta un modelo KNN, permite al usuario explorar los registros de datos y muestra los siete indicadores más influyentes en la esperanza de vida al nacer utilizando el modelo KNN.

**Modelo de Series de Tiempo:**

Es una técnica que se usa para predecir el comportamiento futuro de una variable a lo largo del tiempo. Se basa en patrones y tendencias pasadas para hacer predicciones precisas. Ayuda a comprender la evolución temporal de los datos y a tomar decisiones informadas.

**I.1. Análisis por País:**

En esta sección, se realiza un análisis de series de tiempo para un país seleccionado. El código realiza lo siguiente:

* Filtra los datos hasta el año seleccionado.
* Muestra una lista desplegable para seleccionar el país de interés.
* Filtra los datos solo para el país seleccionado.
* Muestra los datos históricos del indicador "Esperanza de Vida al Nacer".
* Solicita el número de años para el pronóstico.
* Realiza una búsqueda en cuadrícula para encontrar los mejores parámetros (p, d, q) del modelo ARIMA.
* Ajusta el modelo ARIMA con los mejores parámetros encontrados.
* Genera pronósticos para los años siguientes.
* Visualiza los datos históricos y los pronósticos en un gráfico.

**I.2. Crecimiento/Decrecimiento por Región:**

En esta sección, se realiza un análisis de crecimiento o decrecimiento por región. El código realiza lo siguiente:

* Muestra una lista desplegable para seleccionar el tipo de análisis (crecimiento o decrecimiento).
* Selecciona las regiones de interés.
* Itera sobre las regiones seleccionadas y realiza los siguientes pasos para cada región:
  + Filtra los datos hasta el año seleccionado.
  + Define los posibles valores de los parámetros (p, d, q) para el modelo ARIMA.
  + Itera sobre cada país de la región y realiza una búsqueda en cuadrícula para encontrar los mejores parámetros del modelo ARIMA.
  + Ajusta el modelo ARIMA con los mejores parámetros encontrados.
  + Genera pronósticos para los años siguientes.
  + Calcula el crecimiento o decrecimiento esperado según el tipo de análisis.
  + Ordena los países según el crecimiento o decrecimiento esperado.
  + Muestra los resultados en una tabla.

En resumen, el código realiza análisis de series de tiempo para países específicos y análisis de crecimiento o decrecimiento por región utilizando el modelo ARIMA y muestra los resultados en gráficos y tablas.

**Modelo de Regresión Lineal:**

Es un algoritmo de aprendizaje supervisado que busca establecer una relación lineal entre variables predictoras y una variable de salida continua. Se ajusta una línea recta a los datos observados y se utiliza para predecir nuevos valores y analizar el impacto de las variables predictoras. Es una herramienta fundamental en la predicción y comprensión de relaciones entre variables continuas.

El código de regresión lineal realiza los siguientes pasos:

* Permite al usuario seleccionar uno o varios países.
* Para cada país seleccionado:
  + a. Filtra los datos correspondientes al país.
  + b. Crea un conjunto de variables independientes y una variable dependiente.
  + c. Divide los datos en conjuntos de entrenamiento y prueba.
  + d. Escala las características de los conjuntos de entrenamiento y prueba.
  + e. Crea y entrena un modelo de regresión lineal utilizando los conjuntos de entrenamiento escalados.
  + f. Calcula el coeficiente de determinación (R²) del modelo utilizando los conjuntos de prueba escalados.
  + g. Almacena el modelo si el R² obtenido es mejor que el anterior.
  + h. Calcula los residuos del mejor modelo utilizando los conjuntos de prueba escalados.
  + i. Muestra gráficos que representan las relaciones entre las variables independientes y la variable dependiente, así como las predicciones y los valores reales.
* Si no se selecciona ningún país, muestra un mensaje indicando que no se han seleccionado países.

En resumen, el código realiza un análisis de regresión lineal para cada país seleccionado, encontrando las mejores variables independientes y mostrando visualizaciones de las relaciones y predicciones del modelo.

**Modelo de ML en producción:**

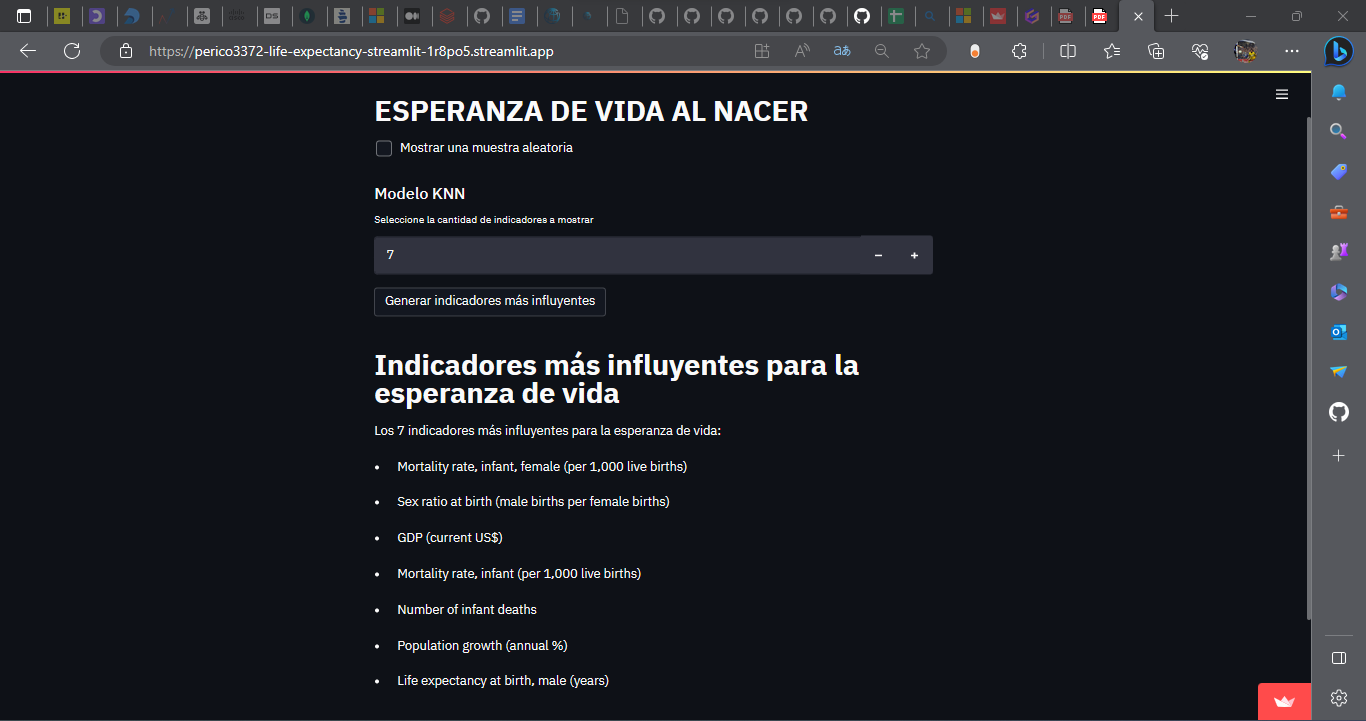
En caso de haber desarrollado un modelo de Machine Learning, lo llevaremos a producción, integrándose en un entorno operativo donde pueda ser utilizado de forma continua para generar predicciones o clasificaciones.

**Modelo KNN**

El objetivo de este código es implementar un modelo de Machine Learning basado en el algoritmo KNN (K-Nearest Neighbors) para generar recomendaciones de indicadores influyentes en la esperanza de vida al nacer. El flujo de trabajo del código se puede describir de la siguiente manera:

* Carga de datos: Se cargan los datos desde un archivo CSV utilizando pandas. Los datos contienen información sobre diversos indicadores relacionados con la esperanza de vida al nacer.
* Selección de características: Se seleccionan las columnas relevantes del conjunto de datos como características para el modelo KNN.
* Imputación de valores faltantes: Se utiliza la estrategia de imputación "mean" (media) para manejar los valores faltantes en las características seleccionadas.
* Creación y ajuste del modelo KNN: Se crea un modelo KNN utilizando la clase NearestNeighbors de scikit-learn. El modelo se ajusta a los datos imputados.
* Generación de recomendaciones: Se define una función get\_recommendations que recibe el índice de un país y devuelve los índices de los indicadores más similares utilizando el modelo KNN.
* Visualización de los datos: Se brinda la opción al usuario de mostrar los primeros o últimos 5 registros del conjunto de datos.
* Generación de indicadores más influyentes: Al hacer clic en un botón, se generan los indicadores más influyentes para la esperanza de vida al nacer. Se recorren todos los países en el conjunto de datos y se obtienen las recomendaciones de indicadores utilizando la función get\_recommendations. Luego, se cuentan las frecuencias de los indicadores recomendados y se muestran los siete indicadores más influyentes.

El código utiliza la biblioteca Streamlit para crear una interfaz de usuario interactiva, donde el usuario puede seleccionar opciones y obtener recomendaciones basadas en el modelo KNN.



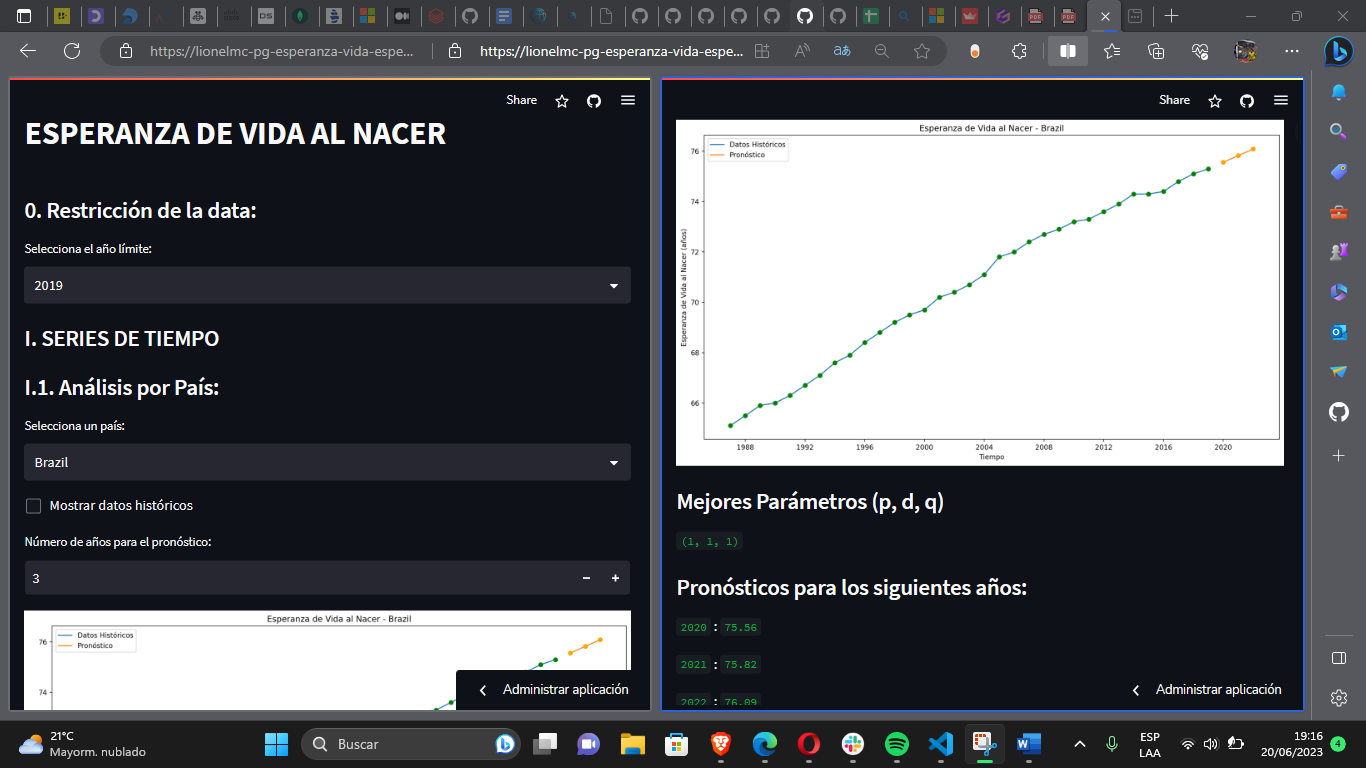
**Modelo de Series de Tiempo:**

[**Streamlit (lionelmc-pg-esperanza-vida-esperanza-vida-v2-1avwj4.streamlit.app)**](https://lionelmc-pg-esperanza-vida-esperanza-vida-v2-1avwj4.streamlit.app/)

El código proporcionado tiene como objetivo realizar un análisis de series de tiempo y pronóstico de la esperanza de vida al nacer para diferentes países. A continuación, se presenta una descripción del flujo de trabajo del código:

**I.1. Análisis por País:**

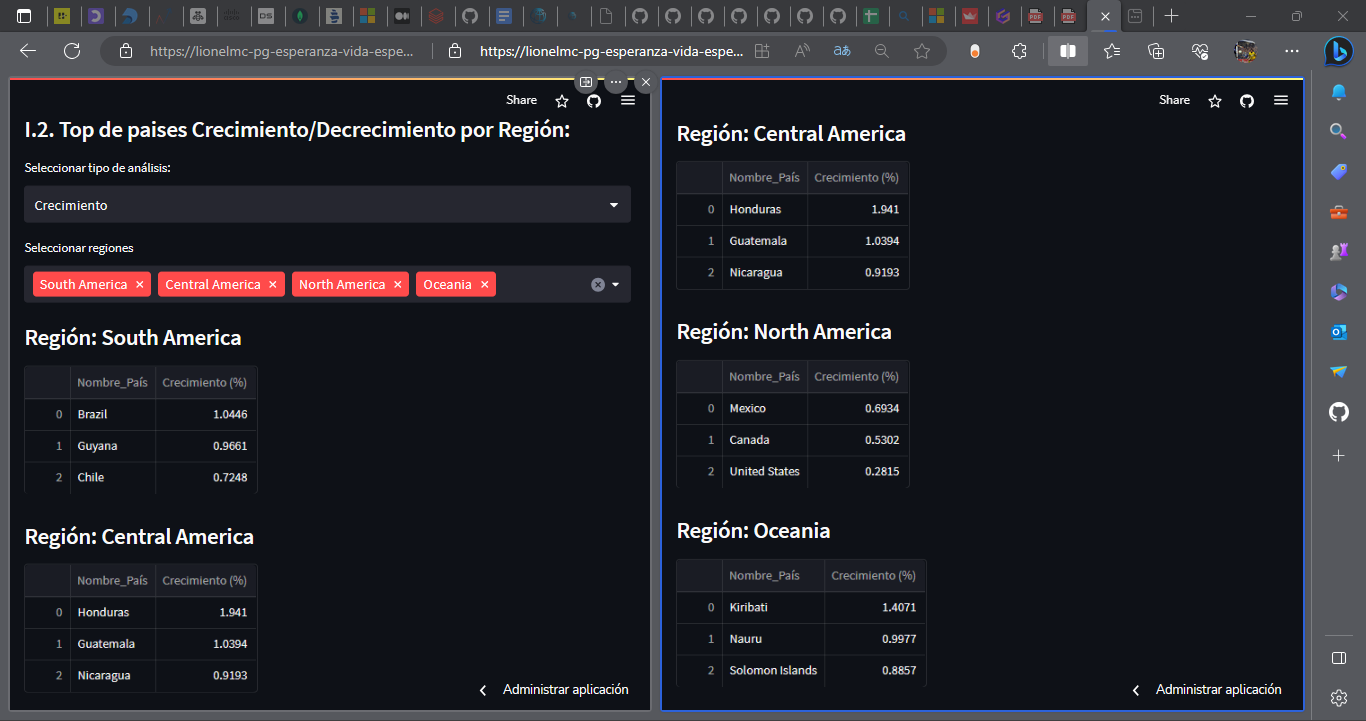
* Importación de bibliotecas: Se importan las bibliotecas necesarias, como Streamlit, pandas, matplotlib, ARIMA de statsmodels, combinations de itertools, train\_test\_split de sklearn.model\_selection, LinearRegression de sklearn.linear\_model, StandardScaler de sklearn.preprocessing y SimpleImputer de sklearn.impute. También se importa la biblioteca warnings.
* Desactivar las advertencias: Se desactivan las advertencias utilizando la función filterwarnings() de la biblioteca warnings.
* Título de la aplicación: Se muestra el título de la aplicación utilizando la función title() de Streamlit.
* Carga de datos: Se carga el archivo CSV que contiene los datos históricos desde 'data.csv' utilizando la función read\_csv() de pandas.
* Preprocesamiento de datos: Se convierte la columna 'Time' en formato de fecha utilizando la función to\_datetime() de pandas. Luego, se establece la columna 'Time' como el índice del DataFrame utilizando la función set\_index().
* Restricción de la data: Se muestra un subencabezado para la restricción de la data y se permite al usuario seleccionar un año límite utilizando la función selectbox() de Streamlit.
* Análisis de series de tiempo: Se muestra un subencabezado para el análisis de series de tiempo y se permite al usuario seleccionar un país de interés utilizando la función selectbox() de Streamlit.
* Filtrado de datos: Los datos se filtran para incluir solo los años anteriores o iguales al año límite seleccionado por el usuario.
* Datos históricos: Se muestra una opción para mostrar los datos históricos seleccionados utilizando la función checkbox() de Streamlit.
* Pronóstico de series de tiempo: Se permite al usuario seleccionar el número de años para el pronóstico utilizando la función number\_input() de Streamlit.
* Búsqueda de mejores parámetros: Se realiza una búsqueda en cuadrícula para encontrar los mejores parámetros (p, d, q) para el modelo ARIMA utilizando bucles for anidados.
* Ajuste del modelo: Se ajusta el modelo ARIMA con los mejores parámetros encontrados utilizando la función ARIMA() de statsmodels y la función fit().
* Generación de pronósticos: Se generan pronósticos para los años siguientes utilizando la función get\_forecast() y la función predicted\_mean() del modelo ajustado.
* Visualización de resultados: Se muestra un gráfico que representa los datos históricos y los pronósticos utilizando la función subplots() de matplotlib. Se utiliza la función plot() para trazar las series de tiempo y la función scatter() para agregar marcadores a los puntos de datos y a los pronósticos.
* Mejores parámetros: Se muestran los mejores parámetros (p, d, q) encontrados utilizando la función write() de Streamlit.
* Pronósticos para los siguientes años: Se muestran los pronósticos para los años siguientes utilizando la función write() de Streamlit.
* Implementación en Streamlit Cloud: El código ha sido adaptado y optimizado para su implementación en Streamlit Cloud, una plataforma en la nube que permite desplegar aplicaciones web interactivas de manera sencilla.



**I.2. Crecimiento/Decrecimiento por Región:**

El código proporcionado tiene como objetivo mostrar el crecimiento o decrecimiento de la esperanza de vida al nacer para diferentes países de diferentes regiones. A continuación, se presenta una descripción del flujo de trabajo del código:

* Importación de bibliotecas: Se importan las bibliotecas necesarias, como Streamlit, pandas y ARIMA de statsmodels.
* Subtítulo y selección de la región: Se muestra un subtítulo indicando el análisis de crecimiento o decrecimiento por región y se permite al usuario seleccionar una región de interés utilizando la función selectbox() de Streamlit.
* Definición de las regiones y países: Se definen las regiones de interés y los países correspondientes a cada región en un diccionario.
* Selección de regiones: Se permite al usuario seleccionar una o varias regiones utilizando la función multiselect() de Streamlit.
* Iteración sobre las regiones seleccionadas: Se itera sobre cada región seleccionada y se realiza el análisis de series de tiempo para los países correspondientes.
* Búsqueda de los mejores parámetros: Se realiza una búsqueda en cuadrícula para encontrar los mejores parámetros (p, d, q) para el modelo ARIMA. Se iteran sobre los posibles valores de p, d y q, y se ajusta el modelo ARIMA con esos parámetros. Se utiliza el criterio de información de Akaike (AIC) para evaluar la calidad del modelo y se actualizan los mejores parámetros si se encuentra un valor de AIC menor.
* Ajuste del modelo y generación de pronósticos: Se ajusta el modelo ARIMA con los mejores parámetros encontrados para cada país y se generan pronósticos para los años siguientes utilizando la función get\_forecast() y la función predicted\_mean() del modelo ajustado.
* Análisis de crecimiento o decrecimiento: Se verifica si se seleccionó el análisis de crecimiento o decrecimiento. Para el crecimiento, se calcula el crecimiento esperado en porcentaje para los próximos años si el crecimiento ha sido positivo en cada año del pronóstico. Para el decrecimiento, se calcula el decrecimiento esperado en porcentaje para los próximos años si el crecimiento ha sido negativo en cada año del pronóstico.
* Ordenamiento de los países: Los países se ordenan según el crecimiento o decrecimiento esperado, de mayor a menor. Los resultados se almacenan en un diccionario de resultados.
* Visualización de los resultados: Se muestra el resultado en una tabla que muestra los tres países con el mayor crecimiento o decrecimiento esperado, junto con el porcentaje correspondiente.
* Implementación en Streamlit Cloud: El código ha sido adaptado y optimizado para su implementación en Streamlit Cloud, una plataforma en la nube que permite desplegar aplicaciones web interactivas de manera sencilla.

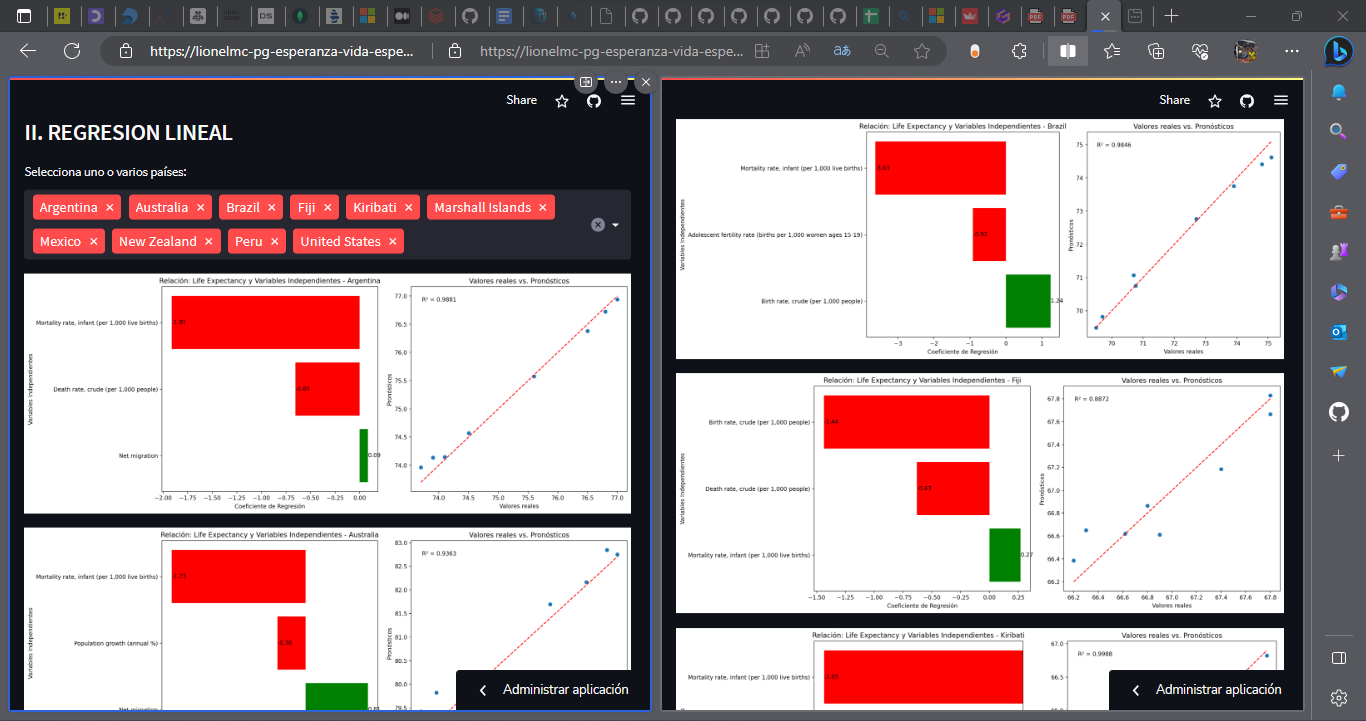


**Modelo de Regresión Lineal:**

[**Streamlit (lionelmc-pg-esperanza-vida-esperanza-vida-v2-1avwj4.streamlit.app)**](https://lionelmc-pg-esperanza-vida-esperanza-vida-v2-1avwj4.streamlit.app/)

El objetivo de este código es implementar un modelo de regresión lineal para predecir la expectativa de vida al nacer utilizando variables independientes relacionadas. El flujo de trabajo del código se puede describir de la siguiente manera:

* Selección de países: El usuario puede seleccionar uno o varios países de una lista proporcionada en la interfaz de Streamlit.
* Verificación de selección de países: Se verifica si se han seleccionado países antes de continuar. Si no se han seleccionado países, se muestra un mensaje indicando que no hay países seleccionados.
* Procesamiento de datos para cada país seleccionado:
* Se filtran los datos para el país seleccionado.
* Se crea un nuevo DataFrame (model\_data) que contiene las variables deseadas para el modelo de regresión.
* Los valores faltantes en model\_data se imputan utilizando la media de cada columna.
* Búsqueda exhaustiva de variables: Se realiza una búsqueda exhaustiva para encontrar las mejores combinaciones de variables independientes que maximicen el coeficiente de determinación (R²) del modelo de regresión. Se prueban todas las combinaciones posibles de 2 y 3 variables independientes.
* División de datos y escalamiento: Se dividen los datos en conjuntos de entrenamiento y prueba. Las características (variables independientes) se escalan utilizando el escalador StandardScaler.
* Creación y entrenamiento del modelo de regresión lineal: Se crea y entrena el modelo de regresión lineal utilizando las variables seleccionadas.
* Evaluación del modelo: Se calcula el coeficiente de determinación (R²) del modelo utilizando el conjunto de pruebas. Un valor más cercano a 1 indica un mejor ajuste del modelo.
* Cálculo de residuos: Se calculan los residuos del modelo para evaluar su desempeño. Los residuos representan la diferencia entre los valores reales y los valores pronosticados por el modelo.
* Análisis de variables seleccionadas: Se crea un DataFrame (relaciones\_data) que muestra las variables seleccionadas y sus relaciones (coeficientes) con la expectativa de vida al nacer. Las variables se ordenan de manera descendente según su relación con la expectativa de vida.
* Visualización de resultados: Se grafica la relación entre las variables independientes y la expectativa de vida, utilizando un gráfico de barras para mostrar los coeficientes de regresión y un gráfico de dispersión para comparar los valores reales y pronosticados. Se muestra el valor de R² en el gráfico.
* Iteración para cada país seleccionado: El flujo de trabajo se repite para cada país seleccionado, generando gráficos y resultados individuales para cada país.
* Visualización de resultados finales: Se muestran los gráficos y los resultados para cada país seleccionado.
* Implementación en Streamlit Cloud: El código ha sido adaptado y optimizado para su implementación en Streamlit Cloud, una plataforma en la nube que permite desplegar aplicaciones web interactivas de manera sencilla. Esto permite a los usuarios acceder a la aplicación, seleccionar países y obtener predicciones de expectativa de vida utilizando el modelo de regresión lineal.



**Recursos y Documentación**

| **Perfil** | **Tipo** | **Link del código / detalle / archivo** |
| --- | --- | --- |
| DA | Diseño de Reportes / Dashboards | https://github.com/datainsi/Esperanza\_de\_vida/blob/main/Dashboard\_final.pbix |
| DA | KPIs | https://github.com/datainsi/Esperanza\_de\_vida/blob/main/Dashboard\_final.pbix |
| ML | Modelo KNN | [**https://github.com/datainsi/Esperanza\_de\_vida/blob/main/Streamlit/ML1/ml\_01.py**](https://github.com/datainsi/Esperanza_de_vida/blob/main/Streamlit/ML1/ml_01.py) |
| ML | Modelo de Series de Tiempo | [**https://github.com/datainsi/Esperanza\_de\_vida/blob/main/Streamlit/ML2/Esperanza\_Vida\_v2.py**](https://github.com/datainsi/Esperanza_de_vida/blob/main/Streamlit/ML2/Esperanza_Vida_v2.py) |
| ML | Modelo de Regresión Lineal | [**https://github.com/datainsi/Esperanza\_de\_vida/blob/main/Streamlit/ML2/Esperanza\_Vida\_v2.py**](https://github.com/datainsi/Esperanza_de_vida/blob/main/Streamlit/ML2/Esperanza_Vida_v2.py) |

**Informe de análisis:**

Elaboramos un informe final que resume los resultados del análisis de datos realizado en el proyecto. Este informe incluirá los hallazgos más relevantes y las conclusiones obtenidas.

| **DA** | **Diseño de Reportes/Dashboards** | |
| --- | --- | --- |
| Se enlista los principales indicadores que se utilizaron en los paneles para el desarrollo de Dashboards y Reportes | |
| Esperanza de vida | Life\_expectency\_total(in years) |
| Esperanza de vida saludable | Healthy\_life\_expectency(in years) |
| Tasa de mortalidad infantil | Mortality\_rate\_infant( % per 1000 infant) |
| Tasa de natalidad | Births\_rate ( ratio per 1000 inhabitants) |
| **KPIs** | |
| En base a un análisis sobre cada indicador, se propuso lograr los siguientes KPI´S | |
| Incremento del 1% de la esperanza de vida. | A fin de lograr una mejora en calidad de la salud , se propuso incrementar en un 1% la esperanza de vida en aquellos países(según su continente) ocupaban el último lugar en esta medición en relación al año 2019. |
| Incremento del 1% de la esperanza de vida saludable. | En el caso de la esperanza de vida saludable, debido a la disponibilidad de los datos, se optó por estudiar el continente americano. Y en base a los resultados, propusimos incrementar en un 1% la esperanza de vida saludable |
| Disminución de 1% de la tasa de mortalidad | En el caso de la mortalidad infantil, consideramos relevante trabajar en aquellos países con una tasa alta como el caso de Haiti, y proyectamos disminuir en 1% para el año 2020, basándose en los datos del año 2019 |
| Incremento del 1% de la tasa de natalidad | Para el análisis de la tasa de natalidad, no se consideró los aspectos socio-económicos y culturales, y se propuso trabajar sobre aquellos países con una baja tasa de natalidad, como el caso de Canadá. Considerando el dato del año 2019, propusimos incrementar en 1% está medición |

| **ML** | **MODELO KNN:** | |
| --- | --- | --- |
| Se lista los indicadores más influyentes para la esperanza de vida, según el modelo:   | Mortality rate, infant, female (per 1,000 live births) | Population growth (annual %) | | --- | --- | | Sex ratio at birth (male births per female births) | Population growth (annual %) | | GDP (current US$) | Mortality rate, adult, female (per 1,000 female adults) | | Mortality rate, infant (per 1,000 live births) | Adolescent fertility rate (births per 1,000 women ages 15-19) | | Number of infant deaths | GDP growth (annual %) | | |
|
| **MODELO DE SERIES DE TIEMPO:** | |
| Con el fin de mitigar la influencia de los datos atípicos generados por la pandemia de COVID-19, el análisis se llevará a cabo hasta el año 2019. En este análisis, identificamos los países líderes en cada región en términos de crecimiento y decrecimiento pronosticados, en base a sus cambios porcentuales. | |
| **Crecimiento** | **Decrecimiento** |
| | **Región** | **País** | **(%)** | | --- | --- | --- | | AS | Brazil | 1.0446 | | AS | Guyana | 0.9661 | | AS | Chile | 0.7248 | | AC | Honduras | 1.9409 | | AC | Guatemala | 1.0394 | | AC | Nicaragua | 0.9193 | | AN | Mexico | 0.6934 | | AN | Canada | 0.5301 | | AN | United States | 0.2814 | | OC | Kiribati | 1.4071 | | OC | Nauru | 0.9976 | | OC | Solomon Islands | 0.8857 | | | **Región** | **País** | **(%)** | | --- | --- | --- | | AS | Suriname | -0.8753 | | AS | Uruguay | -0.2148 | | AS | Paraguay | -0.1310 | | AC | Panama | -0.0933 | | OC | Micronesia, Fed. Sts. | -0.0698 | |
| **MODELO DE REGRESIÓN LINEAL:** | |
| Este análisis utiliza regresión lineal para explorar la relación entre diferentes variables y la esperanza de vida al nacer en varios países, permitiendo seleccionar los países de interés y visualizar las relaciones encontradas. | |
| **Países con Decrecimiento - América del Sur**        **País con Decrecimiento - América Central**    **País con Decrecimiento - Oceanía**    A través del análisis de los países con decrecimiento por región , podemos identificar las variables más significativas y de mayor influencia en la esperanza de vida al nacer. Entre estas variables clave se encuentran: Mortality rate, infant, female (per 1,000 live births), Death rate, crude (per 1,000 people), Fertility rate, total (births per woman),Adolescent fertility rate (births per 1,000 women ages 15-19). | |

**Conclusiones:**

Estos hallazgos nos permiten comprender mejor los factores que afectan la esperanza de vida en los países con mayor decrecimiento en las regiones. La tasa de mortalidad infantil femenina refleja la calidad de atención médica y la salud materna, lo cual tiene un impacto directo en la supervivencia de los recién nacidos y, en última instancia, en la esperanza de vida general. La tasa de mortalidad general nos brinda una visión más amplia de la salud y calidad de vida en la región, ya que abarca a todas las edades.

Además, la tasa de fertilidad total y la tasa de fertilidad adolescente revelan aspectos demográficos importantes. Una alta tasa de fertilidad total puede implicar una mayor presión sobre los recursos y servicios de salud, mientras que una alta tasa de fertilidad adolescente indica posibles desafíos en la educación, el acceso a métodos anticonceptivos y la planificación familiar.