UNIVERSIDAD COMPLUTENSE MADRID

ntic master
revolucionamos la comunicación

Trabajo Final del Master en Big Data y Data Engineering

Prototipo de Sistema de Vigilancia Epidemiológica

Tutores: Ing. Jorge Centeno e Ing. Alberto González

Estudiante: Estrella Adriana Sicardi Segade

Índice

1. Introducción:

1.1. Contexto:

La vigilancia epidemiológica juega un rol fundamental en la protección de la salud pública, especialmente frente a las crecientes amenazas de enfermedades emergentes y reemergentes. En las últimas décadas, el mundo ha sido testigo del surgimiento de nuevas infecciones, como el COVID-19, así como del resurgimiento de enfermedades que parecían controladas, como el dengue, el zika y la chikungunya. Estas enfermedades, muchas de ellas de origen zoonótico y transmitidas por vectores, exigen sistemas de vigilancia robustos y eficientes que permitan detectar rápidamente brotes y aplicar medidas de control oportunas.

Las enfermedades zoonóticas, que se transmiten de animales a humanos, representan un desafío complejo debido a la interacción constante entre la salud humana, la salud animal y el medio ambiente. Ejemplos recientes, como la pandemia de COVID-19 y la gripe aviar, evidencian cómo los patógenos pueden saltar barreras entre especies y propagarse a nivel global. Por otro lado, las enfermedades transmitidas por vectores, como el dengue y el zika, afectan a millones de personas en todo el mundo y su expansión geográfica se ha visto impulsada por factores como el cambio climático, la globalización y la urbanización descontrolada.

En este contexto, el enfoque de Una Salud cobra relevancia al integrar la salud humana, animal y ambiental en una estrategia conjunta para prevenir y controlar enfermedades infecciosas. Esta visión interdisciplinaria permite identificar los vínculos entre la salud de las personas, los animales y su entorno, ofreciendo una respuesta más efectiva ante las amenazas sanitarias globales.

La gestión eficiente de la información es un pilar clave en este proceso. Para que los sistemas de vigilancia epidemiológica funcionen adecuadamente, es esencial contar con datos precisos, actualizados y accesibles que puedan ser rápidamente analizados. La capacidad de procesar grandes volúmenes de datos y detectar patrones de transmisión es crucial para prevenir brotes, asignar recursos de manera efectiva y diseñar políticas de salud pública informadas. En un mundo cada vez más interconectado, donde las enfermedades pueden propagarse a velocidad sin precedentes, la vigilancia epidemiológica efectiva es más relevante que nunca para mitigar sus impactos en la sociedad.

Un sistema de vigilancia epidemiológica de enfermedades transmisibles es una herramienta esencial para la salud pública, cuyo objetivo principal es monitorear de manera continua la ocurrencia y distribución de enfermedades infecciosas en una población.Una solución tecnológica debe basarse en una arquitectura sólida y

escalable, que permita la recolección, análisis y visualización de datos en tiempo real. Al permitir detectar patrones y tendencias, se facilitará la tarea de identificación temprana de brotes epidémicos y la implementación de medidas de control oportunas por parte de las autoridades de salud estatales.

1.2. Objetivos:

Tomando este contexto, el objetivo de este trabajo es elaborar un prototipo de pipeline de datos para procesar notificaciones de sospecha de enfermedades transmisibles de interés en Uruguay. Este prototipo tiene como finalidad proporcionar una herramienta eficiente y automatizada que facilite el manejo de información crítica relacionada con la vigilancia epidemiológica en tiempo real, fortaleciendo la capacidad de respuesta ante posibles brotes (Figura 1).

Figura 1: Ilustración del objetivo del prototipo a desarrollar, donde a partir de los datos obtenidos en todo el territorio, se puede monitorizar la situación de manera digital.

Más concretamente, partiendo de las notificaciones de enfermedades cuya sospecha es de reporte obligatorio que provienen de los prestadores de servicios de salud (Hospitales, Policlínicas, Emergencias, Laboratorios, etc), nos proponemos elaborar un prototipo que permita la ingesta y almacenamiento de esa información, y su posterior procesamiento para la elaboración de análisis que permitan apoyar la toma de decisiones. Por ser un prototipo nos limitaremos a cuatro enfermedades (eventos): dengue, leptospirosis y meningitis tanto viral como bacteriana. Para

simplificar la casuística no se incluye en este trabajo los detalles de la determinación etiológica exacta de las meningitis y la interpretación de los resultados de laboratorio ha sido reducida a 2 tipos de examen (serología o PCR) con solo a 3 posibles resultados (positivo, negativo o indeterminado).

Como un segundo objetivo se plantea la productivización de un modelo predictivo para la favorabilidad del dengue, que permitiría predecir el nivel de riesgo de brote que tienen las distintas zonas geográficas en función de sus características climáticas y biosociales.

Por último, y no menos importante, dada la naturaleza sensible de los datos de salud (que incluyen información de seres humanos), se construyeron datos sintéticos con Faker, siendo un tercer objetivo la obtención de datos con formatos útiles que permitieran simular lo más posible las características de los casos de usos principales. En particular, para las direcciones de los pacientes, se necesita su geolocalización dentro del territorio Uruguayo de forma coherente, por lo que se utilizan direcciones y coordenadas obtenidas al azar a partir de un dataset libre del "Correo Uruguayo". Los datos incluyen notificaciones de 2024 de los cuatro eventos mencionados, resultados de laboratorio correspondientes a algunos de ellos, y un dataset de 5 años anteriores de casos confirmados de leptospirosis, para construir un corredor endémico. Los detalles de la construcción de los datos serán discutidos en el Anexo 1.

El código de este proyecto de TFM se puede acceder online en el repositorio https://github.com/esicardi/TFM

2. Solución:

2.1. Arquitectura:

Se plantea como solución una arquitectura basada en un sistema de cola de notificaciones que se ingente en tiempo real, con datos persistidos y procesados con un refinamiento serial y progresivo, enriqueciendo los datos a partir de datos ya almacenados que pueden ser ingestados en lotes (batch). Además se propone la construcción de una base de datos escalable y flexible que permita trabajar de manera sencilla con datos estructurados de forma variable (ya que los campos de información es diferente para cada enfermedad, y puede incluso tener ciertas variantes de un paciente a otro). Adicionalmente se propone la implementación de tableros de mando (dashboard) para poder visualizar y analizar la situación epidemiológica de manera más agil y eficiente.

En cuanto a la productivización del modelo de Machine Learning, se propone implementar las buenas prácticas de MLOps, registrando los datos y condiciones de entrenamiento de cada versión, así como las predicciones realizadas y sus métricas, de manera de permitir su posterior auditoría.

2.2. Infraestructura

- En cuanto a la infraestructura, para la cola de datos de las notificaciones, la solución elegida para la cola de datos es Confluent Kafka, que naturalmente proporciona una solución eficiente de administración de un flujo de datos continuo e irregular que debe ser ingestado.

- Para la base de datos de notificaciones, se optó por Mongo DB y su servidor Atlas, dada la flexibilidad de esquema de esta base de datos. Mongo DB permite incluir datos estructurados con diferencias en el esquema dentro de la misma colección, y anexar campos anidados a partir del formato JSON original, lo que es ideal para la característica variada de los datos de las notificaciones, que incluyen síntomas, signos y paraclínica diferente para cada evento (sospecha de enfermedad), sin por eso sacrificar las características ACID (atomicidad, consistencia, aislamiento y durabilidad) para el manejo de sus documentos, lo que es crucial en procesos ETL (Extract, Transform, Load) que manejan grandes volúmenes de datos y requieren operaciones transaccionales seguras.

- Los datos estructurados provenientes de otras fuentes fueron modelados a partir de una base de datos SQL con resultados de laboratorio que permite cruzar los datos desde la ingesta para generar una base de datos enriquecida. Podría haber múltiples instancias de este estilo, como pueden ser también los certificados de defuncíón, pero se optó por considerar solo los resultados de laboratorio para ejemplificar este tipo de transacción. La plataforma utilizada fue Azure SQL Database.

- En cuanto al flujo de datos se eligió la arquitectura Medallion de Delta administrada con Spark desde Azure Databricks con Python, montada sobre contenedores ADSL Gen 2 de un servicio de Almacenamiento de Azure Blob Storage, con conección a Confluent Kafka a través de la biblioteca Python de Confluent, un conector PyMongo para MongoDB y un conector JDBC para Azure SQL.

- Azure Databricks también se eligió para conexión con Power BI para la visualización de datos, brindando en su catálogo las vistas de las tablas agregadas de datos necesarias procesadas a partir de las bases de datos mencionadas anteriormente.

- Power BI fue utilizada como herramienta principal de visualización aunque también se implementaron algunas visualizaciones con Matplotlib de Python y

una interfase con ReTool. En particular, ReTool ofrece un conector nativo con MongoDB como una ventaja estratégica. El mapa interactivo generado para dicha API se montó en un servidor web de Netilfy.

- En lo que refiere a Machine Learning se optó también por Azure Databricks, con su integración a MLflow, lo que permite el registro y serialización adecuado de los modelos y sus versiones, persistiendo con Spark la información relevante para auditoría. Flask puede utilizarse para construir API Rest de servicio de predicción del modelo, pero requiere de un servidor aparte dado que Databricks no soporta directamente, su implementación, ofreciendo solamente el API rest asociado a MLflow, por lo que su utilización queda como mejora planteada a futuro. Si bien el modelo original estaba escrito en R, utilizando las bibliotecas FuzzySim y ModEvA, y la idea original era aprovechar el soporte de Databricks para R, finalmente se implementó una versión Python del modelo, por ser más ágil su utilización en este contexto.

- Por último, en lo que refiere a integración se utilizó Azure Synapse Analytics, dada su flexibilidad y sencillez para generar triggers de funcionamiento y su integración natural con Azure DevOps, para lo que refiere al versionado de repositorios, si bien este trabajo se comparte a través de GitHub, que también posee conexión a Synapse, dada la necesidad de compartir el código de forma abierta.

La figura 2 muestra un esquema de la infraestructura propuesta para la implementación de la arquitectura del sistema.

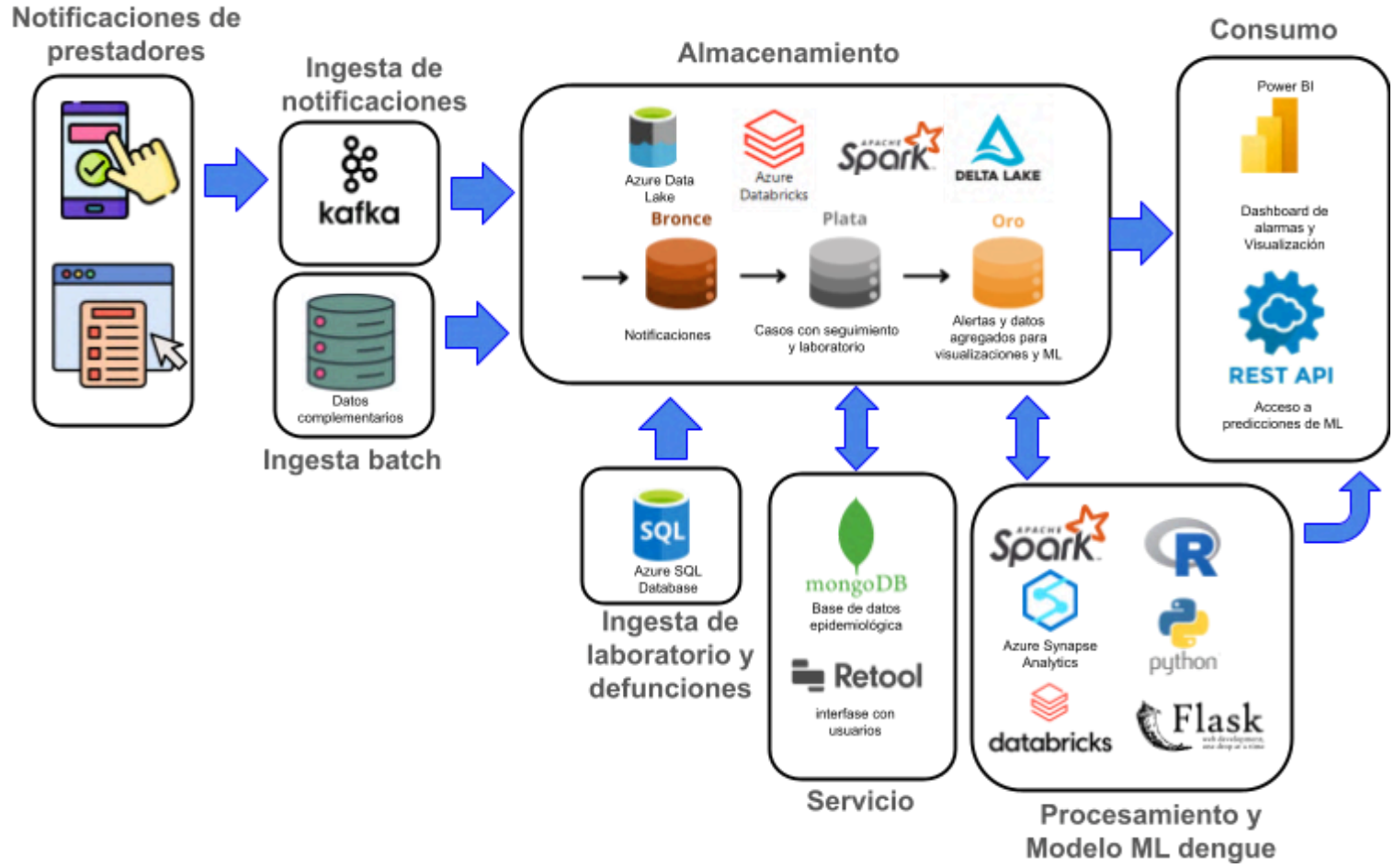


Figura 2: Diagrama de Arquitectura Propuesta

2.3. Flujos de Datos

El diseño de la solución de vigilancia epidemiológica comienza con la captura de notificaciones mediante Kafka, que actúa como el corazón del sistema. Kafka recibe en tiempo real las notificaciones de nuevos casos de dengue desde diversas fuentes, como laboratorios y hospitales. Cada notificación se trata como un evento que se agrega a un tópico específico, lo que permite un manejo eficiente y ordenado de los datos en tiempo real.

Una vez capturadas, las notificaciones son ingeridas en Delta Lake, que sirve como el almacén centralizado de los datos. Este lago de datos, construido sobre Apache Spark, permite la gestión de los datos mediante particiones diarias o semanales, lo que facilita la escalabilidad y el manejo del tiempo. Los datos de Kafka son procesados por Spark Streaming y escritos en tablas Delta, lo que asegura una estructura eficiente y accesible para posteriores análisis.

El enriquecimiento de los datos es un paso crucial en esta arquitectura. Los datos almacenados en Delta Lake se cruzan con información adicional proveniente de bases de datos SQL que contienen resultados de laboratorio y registros de fallecimientos. Este cruce de información se realiza mediante Spark SQL, aprovechando la capacidad de Delta Lake para ejecutar consultas SQL sobre datos tanto estructurados como semiestructurados, lo que permite un análisis más profundo y contextualizado de los casos de dengue.

Para el almacenamiento detallado de cada caso, se utiliza MongoDB. Este sistema de bases de datos permite almacenar los registros de manera detallada y flexible, gracias a su estructura de documentos, que modela la información de manera más cercana a como se presenta en el mundo real. Además, ReTool, una herramienta de desarrollo low-code, facilita la creación de interfaces de usuario personalizadas para interactuar con MongoDB. Esto incluye la posibilidad de diseñar dashboards y formularios que permitan visualizar y gestionar los datos de los casos de manera eficiente. Si bien ReTool permite también la edición de los registros de MongoDB, esta herramienta no fue implementada aún, dado que se requiere una puesta a punto de la sincronización con Delta, de manera de mantener todos los registros actualizados, y se plantea como mejora a futuro.

La visualización de los datos se realiza mediante Power BI, que se conecta tanto a Delta Lake como a MongoDB para crear visualizaciones interactivas y dashboards personalizados. Power BI ofrece una amplia gama de herramientas y capacidades de análisis que facilitan la comprensión de la situación epidemiológica, permitiendo a los usuarios explorar los datos en detalle y obtener insights valiosos que informen la toma de decisiones.

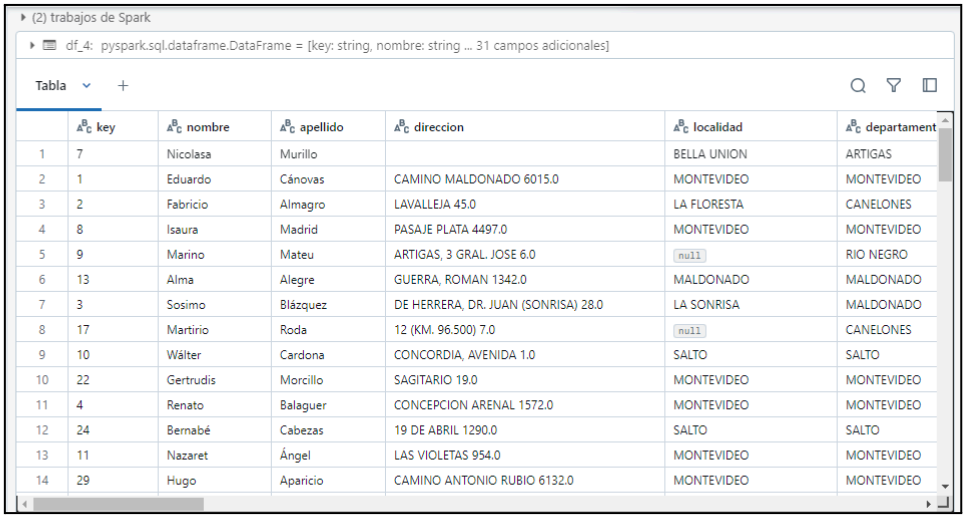
En el ámbito del modelado predictivo, Databricks se utiliza para entrenar y ejecutar un modelo de Machine Learning que predice la probabilidad de un brote de dengue

en función de variables geográficas y otros factores relevantes. Este modelo se entrena utilizando los datos históricos almacenados en Delta Lake, lo que permite anticipar futuros brotes y tomar medidas preventivas. Las predicciones generadas por el modelo se pueden exponer mediante una API REST desarrollada con Flask, que permitiría a otras aplicaciones consumir estas predicciones y tomar decisiones informadas en tiempo real. Sin embargo, dado que Databricks no soporta directamente esta implementación, este desarrollo queda como mejora a futuro. Además, siendo un modelo que predice mapas de riesgo y no valores puntuales, su utilidad dentro del flujo interno, es independiente al servicio externo que permitiría una API REST de estas características, siendo suficiente la implementación provista por el registro de MLFlow.

2.4. Ingestas y Transformaciones

Los datos de las notificaciones son ingestados desde el streaming de datos Kafka hacia Azure Databricks donde son perduradas en tablas Delta del nivel bronce de la arquitectura Medallion propuesta. La arquitectura Medallion divide la arquitectura en tres capas: Bronze (Bronce), Silver (Plata) y Gold (Oro) que se describen a continuación.

1. Capa Bronce: Corresponde a los datos crudos o sin procesar. En nuestro caso, se ingestan los datos de las notificaciones provenientes de topics de Kafka. Dado que el formato de cada notificación es diferente (en lo que refiere a los signos, síntomas y paraclínica relevantes), las notificaciones se encuentran en topics Kafka por separado según a qué enfermedad corresponde. Luego desde Databricks los datos son perdurados en tablas delta con Spark agregando la fecha en que las notificaciones fueron agregadas a la tabla (Ver figura 3). Adicionalmente, todas las notificaciones, para todos los eventos se agregan a la colección notificaciones_eventos en la base de datos vigilancia de MongoDB en Atlas. El multiformato de Mongo permite que las notificaciones de diversos eventos convivan dentro de la misma colección (Ver figura 4).


(2) trabajos de Spark

df_4: pyspark.sql.dataframe.DataFrame = [key: string, nombre: string ... 31 campos adicionales]

Tabla

	key	nombre	apellido	direccion	localidad	departament
1	7	Nicolasa	Murillo		BELLA UNION	ARTIGAS
2	1	Eduardo	Cánovas	CAMINO MALDONADO 6015.0	MONTEVIDEO	MONTEVIDEO
3	2	Fabricio	Almagro	LAVALLEJA 45.0	LA FLORESTA	CANELONES
4	8	Isaura	Madrid	PASAJE PLATA 4497.0	MONTEVIDEO	MONTEVIDEO
5	9	Marino	Mateu	ARTIGAS, 3 GRAL. JOSE 6.0	null	RIO NEGRO
6	13	Alma	Alegre	GUERRA, ROMAN 1342.0	MALDONADO	MALDONADO
7	3	Sosimo	Blázquez	DE HERRERA, DR. JUAN (SONRISA) 28.0	LA SONRISA	MALDONADO
8	17	Martirio	Roda	12 (KM. 96.500) 7.0	null	CANELONES
9	10	Wálter	Cardona	CONCORDIA, AVENIDA 1.0	SALTO	SALTO
10	22	Gertrudis	Morcillo	SAGITARIO 19.0	MONTEVIDEO	MONTEVIDEO
11	4	Renato	Balaguer	CONCEPCION ARENAL 1572.0	MONTEVIDEO	MONTEVIDEO
12	24	Bernabé	Cabezas	19 DE ABRIL 1290.0	SALTO	SALTO
13	11	Nazaret	Ángel	LAS VIOLETAS 954.0	MONTEVIDEO	MONTEVIDEO
14	29	Hugo	Aparicio	CAMINO ANTONIO RUBIO 6132.0	MONTEVIDEO	MONTEVIDEO

Figura 3. Visualización de tabla Spark de Notificaciones de Eventos en Databricks

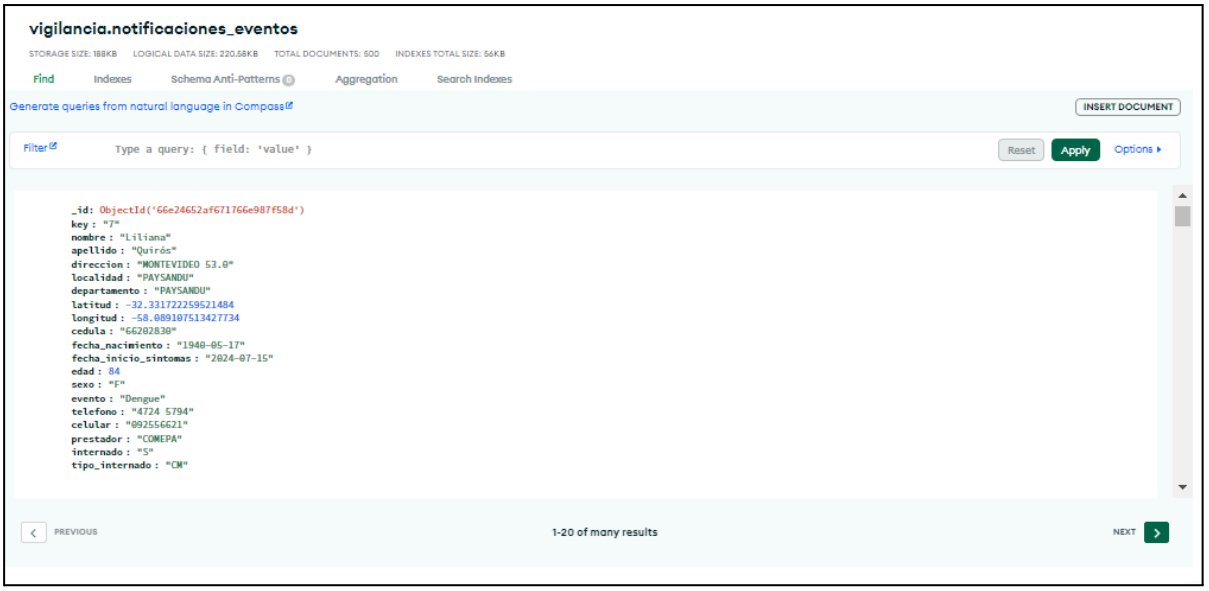


Figura 4: Colección notificaciones_eventos en la base de datos vigilancia de Mongo DB en Atlas.

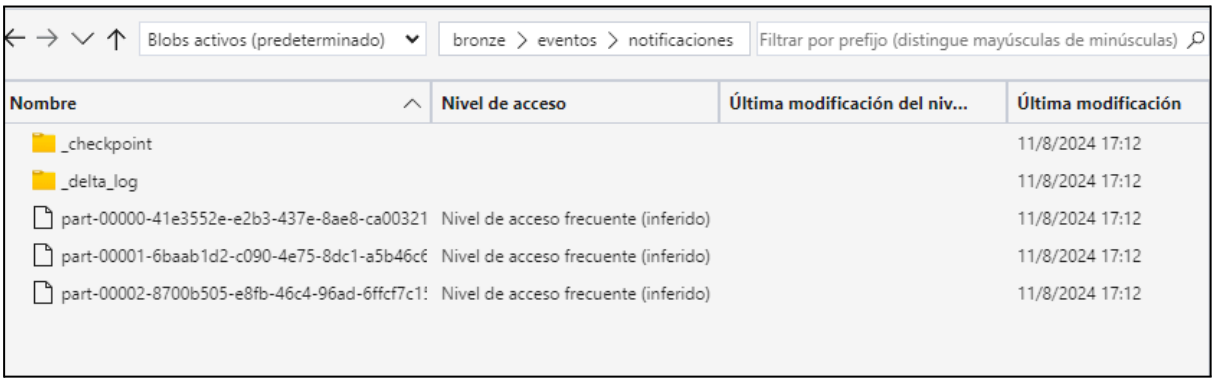


Figura 5: Captura del contenedor “bronze” del Delta Lake de Azure visto desde Microsoft Azure Storage Explorer.

La capa de bronce aloja las tablas en formato detla dentro del contenedor bronze de Azure Blob Storage, que admite el formato ADSL Gen 2 (ver Figura 5).

2. Capa Plata: En la capa plata, se transforma y limpia la información obtenida de la capa bronce. En nuestro caso, se cruzan los datos de las notificaciones con los resultados de laboratorio, que son leídos desde una base de datos Azure SQL permitiendo generar alarmas y se agrega la información sobre el año epidemiológico y la semana epidemiológica (ver definición en glosario). Ver figura 6. Para eso, desde Databricks nos conectamos a Azure SQL con un conector de JDBC. Los datos sintéticos de la base de datos se simulan a partir de los datos sintéticos de las notificaciones, tomando un subconjunto de las mismas, para que las identidades de los casos (persona y evento) sean coherentes. Por simplicidad, las pruebas se dividen entre serológica y PCR

(ver definición en glosario), y los resultados posibles son positivo, negativo e indeterminado, si bien la casuística de la interpretación del resultado de estos estudios es un poco más compleja en la vida real. Además, ese resultado de las pruebas solo está disponible cuando la prueba se ha finalizado, lo que se refleja en la variable "estado" de la tabla, cuyo resultado puede ser "finalizado", "en proceso" o "enviado". Solamente aquellas muestras con estado finalizado tienen valor no nulo en la variable "resultado". Adicionalmente, se muestra la variable "fecha_muestra" que en este modelado de datos falsos se elige unos dias después del inicio de síntomas, pero que en la vida real puede ser relevante para determinar la validez de la muestra y elegir el test a realizar que sea más apropiado.

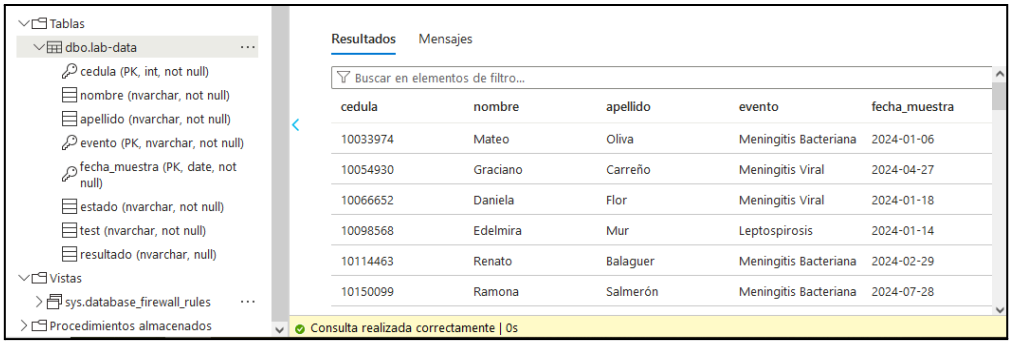


Figura 6. Base de datos SQL con los resultados de laboratorio.

Los resultados de laboratorio, se asocian junto con los datos de las notificaciones, en un campo estructurado para generar una nueva tabla de casos, que además agrega tres nuevos campos, AE (año epidemiológico), SE (semana epidemiológica) y estado. El estado puede ser "confirmado", si la prueba de laboratorio realizada es positiva; "descartado", si la prueba de laboratorio es negativo, o "seguimiento" si no hay prueba de laboratorio disponible, el resultado es indeterminado, o la prueba de laboratorio no tiene aún resultado. Estos registros de los casos se perduran en la capa plata del Deltalake y también conforman la colección "casos_seguimiento" dentro de la base de datos "vigilancia" de MongoDB a la que nos conectamos con pymongo.
Adicionalmente, se genera la tabla de alarmas, para casos en que se requiere acciones. Si una notificación no tiene muestra asociada, se genera la alarma "no tiene muestra en el laboratorio", para poder solicitar que la muestra sea enviada. Los casos confirmados generan una alarma de "investigación de campo" (ver glosario) y aquellos casos en "seguimiento" con un resultado "indeterminado" generan una alarma de "se solicita segunda muestra". Las alarmas se archivan en la colección "alarmas_epidemiologicas" de la base de datos vigilancia de MongoDB (figura 7).

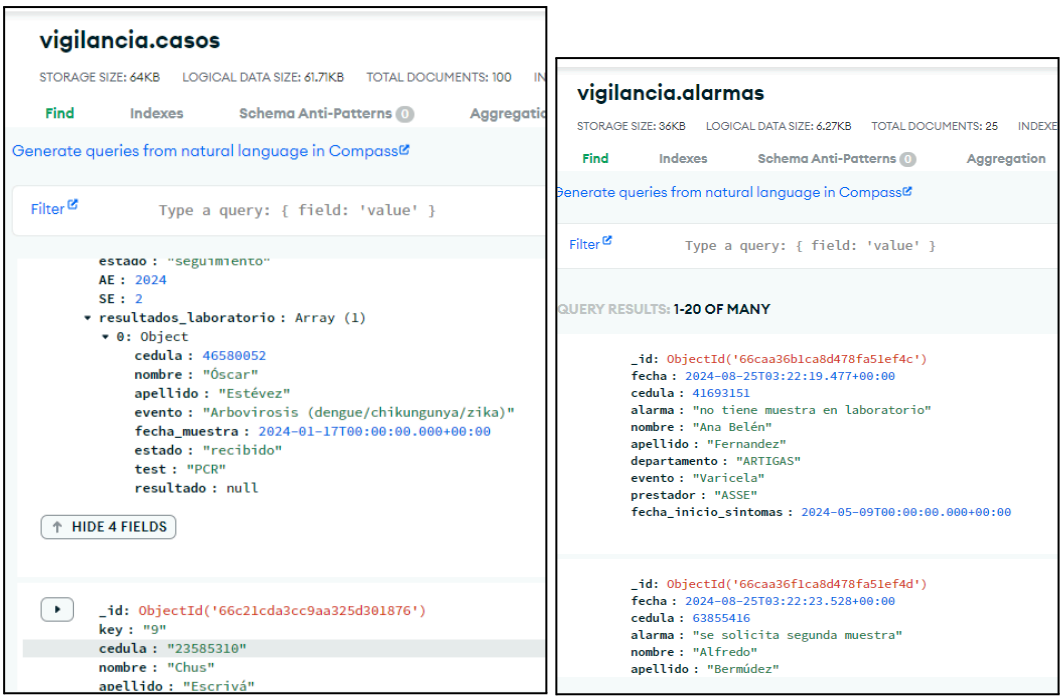


Figura 7: Captura de las colecciones de “casos” y de “alarmas” en la base de datos de “vigilancia”, vista desde Compass, en el servidor de MongoDB dentro de Atlas.

Adicionalmente, se genera una tabla que registra, para el año epidemiológico en curso (AE=2024) para cada valor la variable evento (que en este caso puede tomar los valores “Leptospirosis”, “Dengue”, “Meningitis Viral” o “Meningitis Bacteriana”) y cada semana epidemiológica (en este caso la SE para el año en curso llega hasta 27, porque los datos simulados suponen que el año no está cerrado, en cuyo caso el valor de SE va de 1 a 52 o 53, dependiendo del año), la cantidad de casos confirmados . Esta tabla (curva_epi) también se perdura con spark en formato Delta en la Capa Plata, en el contenedor “silver” de la cuenta de almacenamiento de Azure Blob Storage que también contiene el contenedor “bronze” (ver parte anterior) y el contenedor “gold” (ver siguiente parte). Ver figura 8.

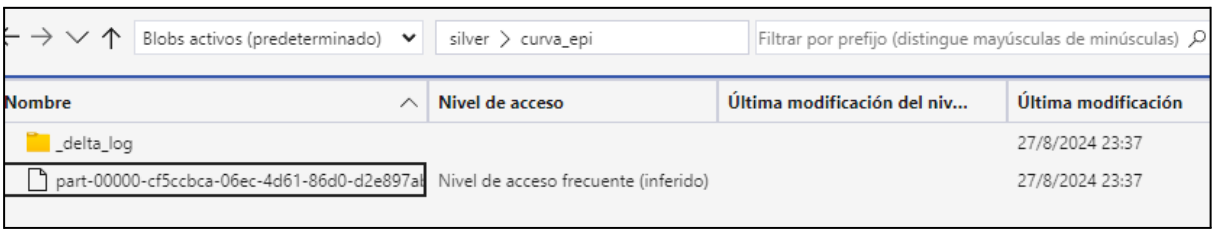


Figura 8: Captura del contenedor “silver” del Delta Lake de Azure visto desde Microsoft Azure Storage Explorer.

3. Capa Oro: Datos optimizados para análisis. La capa oro contiene los datos totalmente refinados y optimizados para el análisis y visualización. En nuestro

caso en esta capa se procesan las tablas necesarias para la construcción de un corredor endémico (ver Anexo 3) y tablas necesarias para medir la performance del modelo de Machine Learning.
Las transformaciones asociadas a la construcción del corredor endémico se ejemplifican en el evento Leptospirosis. Se utilizan los casos confirmados del AE 2024 y se necesitan los datos de los 5 años anteriores. Para modelar esto se simularon con Faker datos históricos de casos confirmados de "Leptospirosis", los cuales fueron guardados en formato JSON y anexados directamente a la colección de casos_seguimiento de la base de datos de vigilancia de MongoDB (en este caso los valores de AE van de 2019 a 2023, con fecha de inicio de síntomas acorde, y los valores de SE van de 1 a 52 o 53 según corresponda a cada AE). A partir de estos datos para el evento Leptospirosis, extraidos a Databricks con pymongo desde MongoDB, para cada valor de AE y SE se calcula la incidencia acumulada. Esto es, para cada año, para la semana 1 se cuenta los casos confirmados con SE =1, para la semana 2, se suman los casos confirmados con SE=2 a los de la semana 1, y así sucesivamente. Para cada año (AE) se reinicia el conteo en la SE=1. Posteriormente, para la variable incidencia acumulada, para cada valor de SE, se calcula la media geométrica (a partir de los valores de esta variable en los años 2019, 2020, 2021, 2022 y 2023) y los extremos del intervalo de confianza (logarítmico) al 95%. Esto se guarda en la tabla spark corredor_2019_2024. Adicionalmente, para los datos de casos confirmados de 2024 se genera la tabla acumulados_2024, que simplemente, da, para ese año, para el evento Leptospirosis, la incidencia acumulada de casos confirmados hasta la última semana que haya datos. Eso permite construir una curva de incidencia acumulada que se puede comparar con los valores del corredor endémico, de manera de identificar visualmente cuando hay cambios significativos en la misma. Las tablas son perduradas con Spark en formato Delta en el contenedor "gold" de la Azure Blob Storage Account y serán utilizadas en la etapa de visualización (Figura 9).

Por otro lado, en esta capa se generan tablas útiles para las predicciones del modelo de Machine Learning. Concretamente, se utilizan datos climáticos promediados, datos geográficos y datos de presencia o no de dengue[1], para lo que divide el mapa de Latinoamérica en "polígonos" (en realidad cuadrados de 0,5x0,5 de longitud y latitud) que son enumerados con un número de identificación (FID). Ver más detalles en Anexo 2.

[1] Aquí cuando se habla de presencia o no de dengue en una zona geográfica, se refiere a un concepto de ecología que indica si históricamente han existido casos de dengue en esa región, y toma valor 1 (presencia) o 0 (ausencia). Entonces, una vez que se ha detectado la presencia en una zona, ya ese valor 1 no cambia más (porque se demuestra que es posible la aparición de esa especie en esa zona). Solo aquellos sitios con valor cero son susceptibles de cambiar.

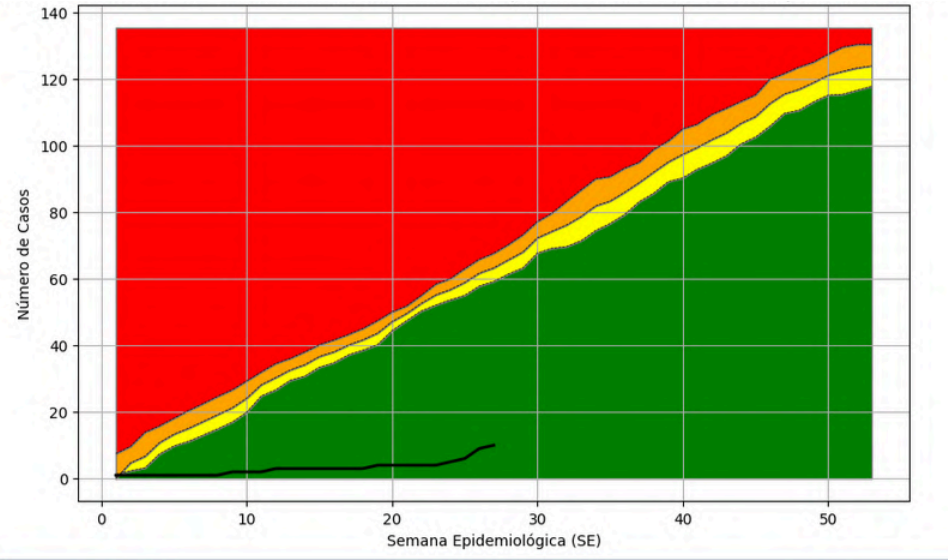


Figura 9: Apariencia del corredor endémico graficado desde Databricks, con la curva epidémica superpuesta (curva negra). Los colores de verde a rojo indican si el nivel de incidencia es bajo, medio, alto, o fuera de lo normal.

Como los únicos datos novedosos (por ser los demás datos promedios históricos o geográficos fijos) son las nuevas incidencias de casos de dengue, se toma la tabla de datos de entrada del modelo, y se modifica la columna “dengue_SA” a partir de los datos de casos confirmados de dengue, en función de la longitud y latitud de la ubicación geográfica del caso, que son extraídos desde MongoDB. Si la latitud y longitud cae dentro de uno de los cuadrados antes mencionados y el valor de dengue_SA era nulo para esa región, se cambia el valor a 1. Como los datos que tenemos son solo de Uruguay (y es la región de interés de las predicciones para nosotros) solo recorremos los valores de FID que caen en su territorio (ver Figura 10).

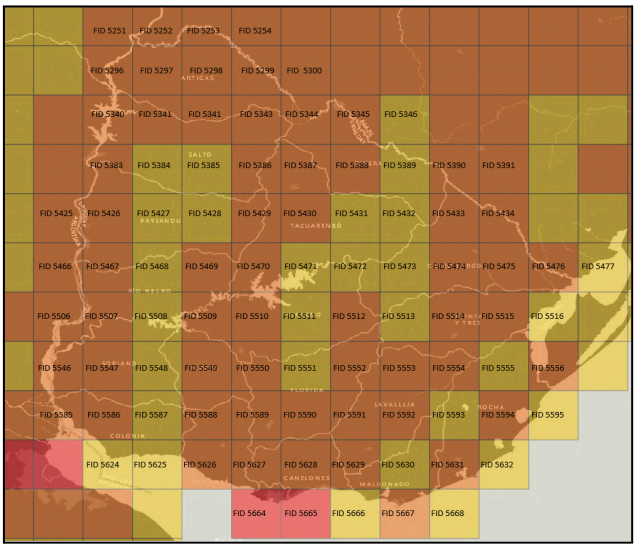


Figura 10: FIDs de los cuadrados (polígonos) correspondientes al territorio uruguayo visualizados desde el programa QGIS.

2.5 Servicio y Consumo

En lo que respecta al servicio y consumo de los datos, este trabajo implementa la base de datos creada en MongoDB, incluyendo las colecciones “notificaciones_eventos”, “casos_seguimiento” y “alarmas_epidemiologicas”; la visualización de los datos con PowerBI y la productivización del Modelo de Favorabilidad para Dengue. Adicionalmente se agrega como interfase tipo “front-end” una API de ReTool, donde visualizamos datos filtrables de las últimos casos procesados así como de las alarmas, así como visualizaciones gráficas, incluyendo un mapa de los resultados del modelo de Machine Learning. A continuación se detallan sus características.

2.5.1. Visualización

En lo que respecta a la base de datos de MongoDB, como ya se dijo, es óptima para los datos epidemiológicos dada su flexibilidad de esquema. Los patrones de infección pueden cambiar rápidamente, con surgimiento de nuevas enfermedades, resurgimiento de viejas enfermedades o aparición de presentaciones de enfermedad con características nuevas, por lo que el poder cambiar rápidamente el esquema sin por eso necesitar crear una colección nueva es una habilidad muy valiosa cuando se trata de datos epidemiológicos. Además MongoDB posee alta eficiencia en lo que refiere a su rendimiento y escalabilidad horizontal, siendo competitivo con las bases de datos SQL en lo que refiere a las características ACID de la operaciones ETL.
Además, MongoDB es muy versátil en cuanto a las búsquedas, incorporando la posibilidad de búsquedas ad hoc, es decir que no necesariamente estén prediseñadas de origen; también posee una buena conexión con Python a través de pymongo, y aporta una interfase intuitiva de navegación (Compass).

En nuestro caso, hemos optado por utilizar la herramienta ReTool para realizar una interfase para la visualización de datos por el usuario final, debido a que la misma posee de forma nativa una conexión, que permite fácilmente desarrollar APIs web que incluyan “queries” interactivas a las colecciones de Mongo. La figura 11 muestra la visualización de los datos de casos_seguimiento y alarmas epidemiológicas en una API de Retool. La misma posee un cuadro de texto que permite filtrar los resultados por palabra clave.

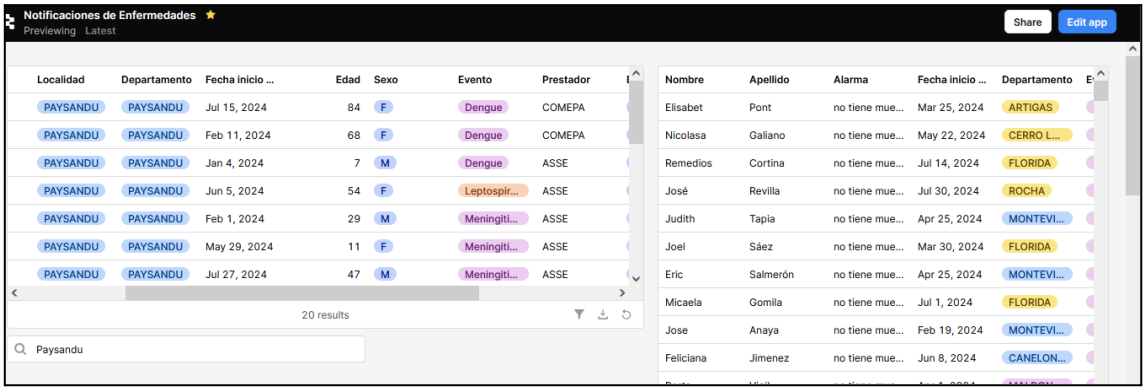


Figura 11: Detalle de la interfase de ReTool buscando datos de la colección de casos_seguimiento a la izquierda y alertas_epidemiologicas a la derecha. en el cuadro de texto de la izquierda se han filtrado los casos de la colección

casos_seguimiento por departamento[2].

ReTool posee además habilidades de editar los registros de MongoDB, pudiendo tanto insertar datos y borrarlos, por lo que inicialmente el proyecto incluía implementar esta posibilidad. Sin embargo esto hubiera supuesto la necesidad de un trigger que permitiera la actualización de las tablas Spark cada vez que uno de estos cambios era realizado, de manera de poder sincronizar los cambios en ambos lados, por lo que, dados los exiguos tiempos disponibles y lo extenso de las aspiraciones del presente proyecto esta característica queda como sugerencia de mejora a futuro.

En lo que respecta a la visualización gráfica de los datos con PowerBI se eligió como ejemplificación la curva epidémica (cantidad de casos confirmados por semana epidemiológica) para dengue y el corredor endémico para leptospirosis (Figura 12).

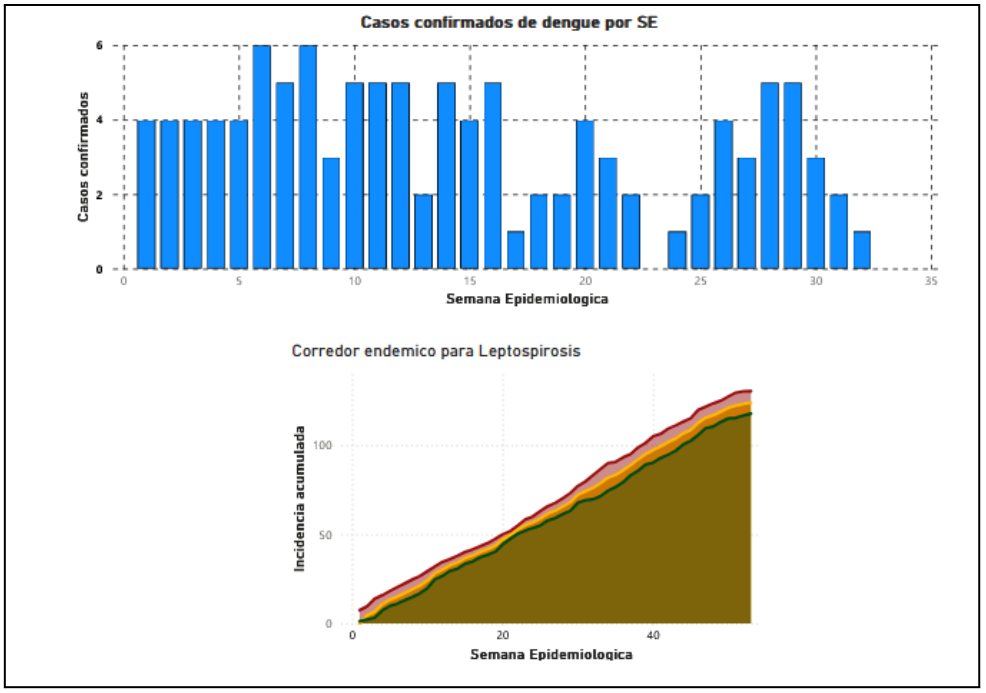


Figura 12: Visualización de la curva epidémica y el corredor endémico desde Power BI.

Para lograr la integración de Power BI con las tablas Delta, se conectó el mismo con Databricks a través de un Token (ver figura 13), de manera de disponibilizar las tablas almacenadas.

[2] El territorio uruguayo se organiza en 19 regiones administrativas llamadas departamentos, Artigas, Canelones, Cerro Largo, Colonia, Durazno, Flores, Florida, Lavalleja, Maldonado, Montevideo, Paysandú, Río Negro, Rivera, Rocha, Salto, San José, Soriano, Tacuarembó y Treinta y Tres.

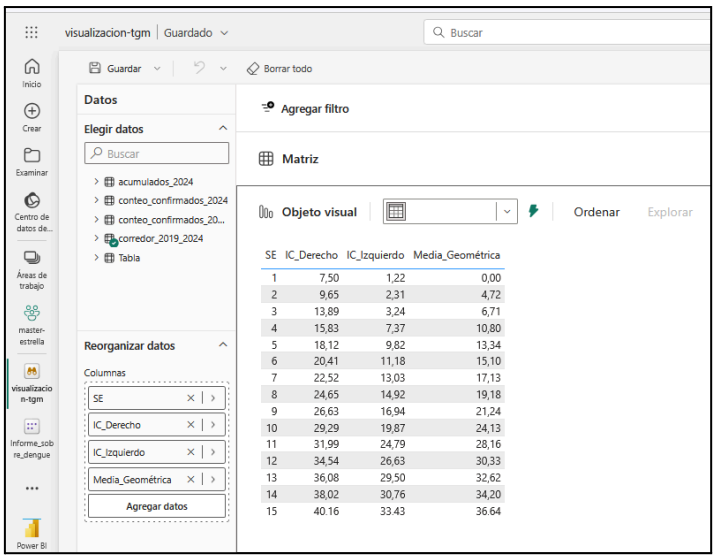


Figura 13: Datos de la tabla corredor_endémico_2019_2024 explorados desde PowerBI, a través del conector con Databricks.

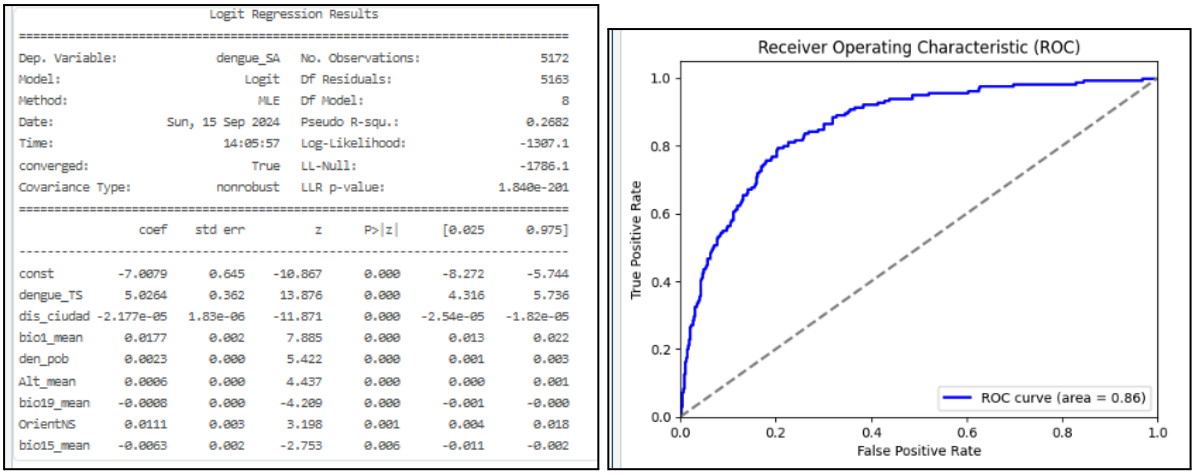
2.5.2. Productivización de Modelo Predictivo

Con respecto al Machine Learning, se implementó la productivización de un modelo de Regresión Logística para predecir la Favorabilidad de dengue para las diferentes zonas geográficas. Los detalles del modelo se explican en el Anexo 2.

El modelo original[3] es una regresión logística escrita en R. En un principio se planteó la idea de usar las capacidades de Databricks para ejecutar comandos R, de manera de entrenar con el código original. Sin embargo, debido a dificultades de dependencias finalmente se optó por reconstruir el modelo en Python. La utilización de Azure Databricks además, dada su integración con MLFlow facilita la serialización del modelo, a partir del registro del mismo en la aplicación, incluyendo su versión y timestamp.. Los datos originales del artículo fueron utilizados como entrada, dividendo los polígonos al azar en una proporción 80-20 entre datos de entrenamiento (train) y los de validación (test). Los datos de cada entrenamiento, los coeficientes resultantes y la fecha de entrenamiento, se perduran en formato Delta en el directorio Modelo_Dengue dentro del contenedor “gold” de Azure Blob Storage. Estos datos incluyen el valor de performance del modelo, que puede ser ilustrado a través de una curva ROC (ver figura 14).

A su vez, a partir de los datos de ocurrencias de dengue, se puede comparar las predicciones del modelo para los cuadrados correspondiente a Uruguay (calculada en la capa “gold” como detallamos anteriormente), con la presencia de la enfermedad en los últimos años en el país. De nuevo, la performance, puede ser medida con una curva ROC.

[3] Applying fuzzy logic to assess the biogeographical risk of dengue in South America. Romero et al. Parasites Vectors (2019) 12:428 https://doi.org/10.1186/s13071-019-3691-5



Logit Regression Results

Dep. Variable:	dengue_SA	No. Observations:	5172
Model:	Logit	Df Residuals:	5163
Method:	MLE	Df Model:	8
Date:	Sun, 15 Sep 2024	Pseudo R-squ.:	0.2682
Time:	14:05:57	Log-Likelihood:	-1307.1
converged:	True	LL-Null:	-1786.1
Covariance Type:	nonrobust	LLR p-value:	1.840e-201

	coef	std err	z	P>|z|	[0.025	0.975]
const	-7.0079	0.645	-10.867	0.000	-8.272	-5.744
dengue_TS	5.0264	0.362	13.876	0.000	4.316	5.736
dis_ciudad	-2.177e-05	1.83e-06	-11.871	0.000	-2.54e-05	-1.82e-05
bio1_mean	0.0177	0.002	7.885	0.000	0.013	0.022
den_pob	0.0023	0.000	5.422	0.000	0.001	0.003
Alt_mean	0.0006	0.000	4.437	0.000	0.000	0.001
bio19_mean	-0.0008	0.000	-4.209	0.000	-0.001	-0.000
OrientNS	0.0111	0.003	3.198	0.001	0.004	0.018
bio15_mean	-0.0063	0.002	-2.753	0.006	-0.011	-0.002

Figura 14: Resultado del entrenamiento del Modelo de Favorabilidad y Curva ROC ilustrando su performance.

Adicionalmente, para poder visualizar las predicciones del modelo se implementó una visualización de las predicciones de favorabilidad en el mapa de Latinoamérica, donde cada polígono (cuadrado) se colorea según una codificación de niveles de favorabilidad, correspondiendo el color verde a la baja favorabilidad (≤ 0,2) el amarillo a la favorabilidad media (entre 0,2 y 0,8) y el rojo a la favorabilidad alta (≥0,8). Recordar que los cuadrados son identificados por un número de FID, y que la región actualizada es solo la correspondiente al territorio uruguayo. Como herramienta de visualización se optó por un script Python utilizando las bibliotecas de Geopandas y Folium. Si bien PowerBI permite la incorporación de scripts Python y R en sus visualizaciones, su implementación solo funciona en la aplicación Desktop de Power BI, y no exportable en la publicación de resultados. Por lo tanto se optó por generar un archivo html a partir de un cuaderno Jupyter, y disponibilizarla (ver página https://tubular-lokum-b7cf2f.netlify.app/) desde el servidor web Netlify. Esto permite visualizar el resultado con nuestra API de Retool (figura 15).

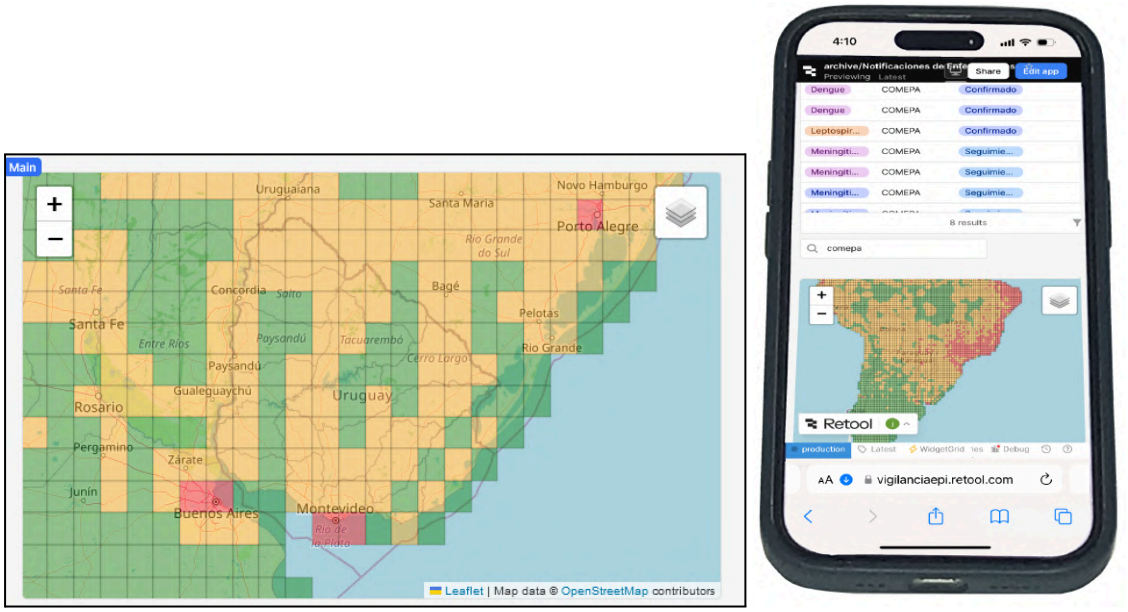


Figura 15: Visualización de la favorabilidad para dengue predecida por el modelo

desde Retool (izquierda), y visión de la API ReTool desde un celular.

Retool además genera APIs responsivas, por lo que se puede diseñar su formato según si el visitante lo hace desde una PC o un celular.

Adicionalmente, Retool también permitiría compartir los resultados de PowerBI como una incrustación, pero dados los permisos por defecto para una cuenta no empresarial de PowerBI, esto no es posible, por lo que se optó por ejemplificarlo incrustando la versión PDF del mismo.

Con vista a las posibles mejoras de este sistema de vigilancia epidemiológica, el desarrollo de una integración de todas las visualizaciones y servicios en una plataforma como Retool como front end presenta buenas potencialidades para su utilización amigable por parte de los técnicos con formación en Medicina, que no son expertos en informática y necesitan ver la evolución de la situación con la mayor inmediatez posible.

3. Orquestación

Para lograr la sincronización y el orden de los procesos, se utilizó Azure Synapse Analytics (ver figura 16), donde se conectó Databricks a Synapse con un Token, de manera de poder importar los cuadernos Databricks, encadenar su ejecución en un “Pipeline” y planificar el momento de ejecución a partir de los triggers. Estos disparadores de actividad permiten sincronizar la logística de actualizaciones del sistema y debe ser afinado finamente tomando en cuenta los horarios de las distintas actividades de los usuarios (previendo posibles sobrecargas) y los tiempos característicos de llegada de los datos, de manera de elegir que las actualizaciones se hagan en tiempo pero no entorpezcan otras tareas. Azure Synapse permite agendar (schedule) las actualizaciones con cierta periodicidad y horario concreto, o puede supeditarse los disparadores a otros parámetros como el volumen de archivos acumulados en una carpeta, o la detección de alguna acción por parte de un usuario u otro programa interactuante.

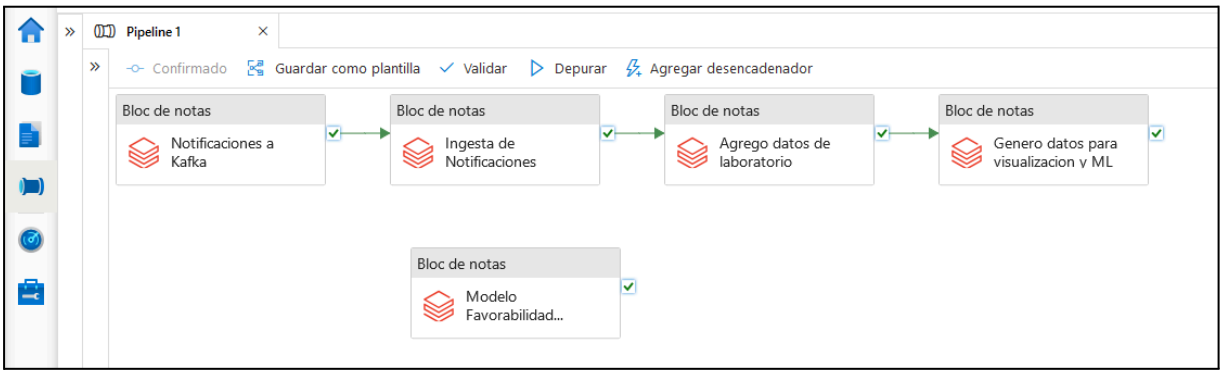


Figura 16: El “pipeline” de la figura muestra la orquestación de los diferentes procesos asociados al flujo de datos del sistema.

4. DevOps

Desde Azure Synapse Analytics es fácil exportar tanto el código de los cuadernos Databricks como la plantilla JSON con los datos de orquestación a Azure DevOps

(figura 17), lo que permite el versionado Git del desarrollo del sistema, con sus correspondientes “ramificaciones” (branches) y “actualizaciones” (commit), de manera sencilla, facilitando la trazabilidad de los cambios realizados y la coordinación del trabajo en equipo.

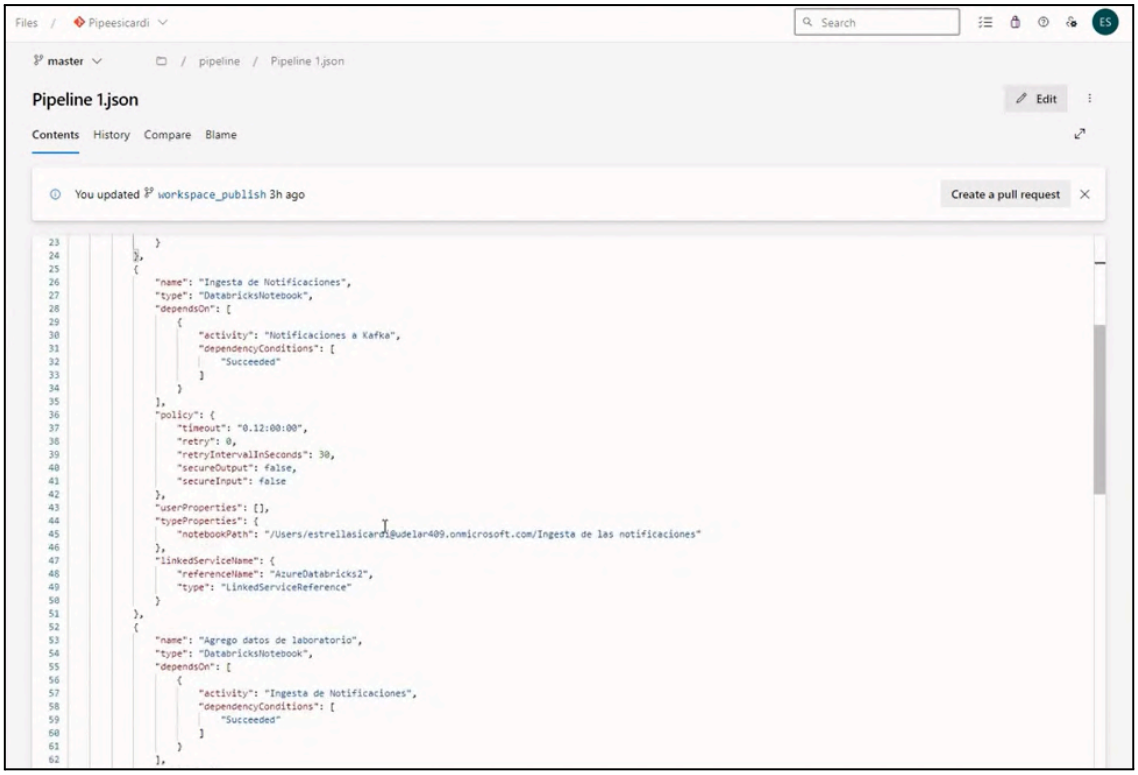


Figura 17: Planilla JSON exportada a DevOps desde Synapse con el Pipeline de Orquestación.

5. Resultados y Conclusiones

Con respecto a los resultados de este trabajo, a pesar del corto tiempo de desarrollo disponible y lo ambicioso del planteamiento original, se logró implementar un esqueleto básico que logra modelar la resolución de la mayoría de las casuísticas más comunes en el área de datos asociados a la vigilancia epidemiológica. Así por ejemplo, si bien se modelan solo 4 enfermedades, es relativamente directo extrapolar esta solución simplemente adicionando nuevas definiciones, y, por ejemplo, el algoritmo del cruzado de datos SQL de laboratorio con la base de datos Mongo de las notificaciones, puede ser fácilmente adaptado para realizar otros cruces de datos como pueden ser las defunciones.

El planteo de una solución para este sistema demandó una extensa investigación partiendo desde la simulación de datos sintéticos que permitieran testear de manera coherente las habilidades deseadas para el funcionamiento óptimo del mismo; pasando por la aplicación de casi todos los conceptos aprendidos en el Master incluyendo bases de datos SQL y NO SQL, Spark, Kafka, Pipelines de datos, Ingesta, Arquitectura de Sistemas, Productivización de Machine Learning etc; y terminando con la exploración de herramientas adicionales, como el caso de ReTool para el desarrollo del Front End, y herramientas como Geopandas y Folium para el manejo y visualización de información geográfica. También se requirió una

profundización en los fundamentos matemáticos de ecología por detrás del modelo de Machine Learning productivizado; entender el funcionamiento de las bibliotecas FuzzySim y modEva de R utilizadas por los autores para permitir la adaptación del modelo a Python; y manejar la aplicación QGIS para acceder a la información de las simulaciones originales, que fue gentilmente proporcionada por dos de los autores, el Dr. David Romero (Universidad de Málaga, España) y el Dr.José Carlos Guerrero Antúnez (Universidad de la República, Uruguay).

Uno de los desafíos que me encontré en este trabajo, es el que respecta a la naturaleza sensible de los datos, el cual pude solucionar con la construcción de datos ficticios, lo que requirió un tiempo considerable para lograr que los mismos tuviesen sentido para el abordaje de las casuísticas que se plantearon resolver con el prototipo implementado.

Dentro de las posibles mejoras a futuro una de las más relevantes, es incluir aspectos de seguridad de los datos, ya sea tanto con la implementación de secretos como con la utilización de firewalls. Este aspecto no fue abordado por no ser el foco principal de la casuística en cuestión, pero con plena conciencia que a la hora de llevar este modelo a producción el pasar por esta implementación es un compromiso ineludible.
Por otro lado, si bien los datos sintéticos permiten visualizar el funcionamiento correcto del sistema a nivel primario, se deberían desarrollar pruebas adicionales para detectar posibles bugs, como pueden ser test unitarios o pruebas de robustez de su desempeño.
Además, como ya se mencionó anteriormente, la casuística de los algoritmos de interpretación de los exámenes de laboratorio, está simplificada, por lo que es necesario agregar más detalles para afinar su performance.
Otra característica importante en lo que respecta al software asociado al sistema de salud, es la interoperabilidad semántica, por lo que, sería necesario implementarla, por medio de herramientas como SNOMED.
Finalmente, a nivel de las visualizaciones hay otras posibles tablas y curvas relevantes, como puede ser la comparación de incidencias acumuladas en función de las regiones, o visualizar la geolocalización de casos de enfermedad para identificar posibles vínculos de contagio entre diferentes pacientes. Además claramente, la incorporación de otras fuentes de datos permitiría el desarrollo de nuevos modelos predictivos, como por ejemplo la vigilancia de enfermedades a nivel de animales domésticos (por parte del Ministerio de Ganadería, Agricultura y Pesca), y el comportamiento de la enfermedad de especies salvajes (por parte del Ministerio de Ambiente) podría permitir modelar las enfermedades zoonóticas de forma integral.

En conclusión si bien el planteo original fue más ambicioso de lo realizable en el tiempo establecido, se logró razonablemente el objetivo principal de este trabajo, desarrollando un prototipo de pipeline de datos para procesar notificaciones de sospecha de enfermedades transmisibles de interés en Uruguay.

En lo personal, siento que he logrado incorporar nuevos conocimientos a nivel informático de manera bastante sólida, a pesar de que mi perfil no es de Ingeniería en Computación sino que mi formación anterior en lo que respecta a programación era más bien asociada a simulaciones en Fortran, R y Matlab.

Anexo 1: Generación de datos sintéticos:

Dado que la información de los pacientes reales es delicada, se optó por utilizar datos sintéticos para el desarrollo y pruebas del sistema de gestión de datos epidemiológicos propuesto. El uso de datos sintéticos permite evaluar y validar el sistema sin poner en riesgo la privacidad de individuos reales.

Para la generación de datos sintéticos se implementó utilizando la biblioteca Faker de Python. Faker es una herramienta altamente eficiente al momento de generar datos ficticios que reproducen con precisión la estructura de los datos reales, brindando así una simulación precisa del sistema. Por simplicidad se incluyeron sólo algunos de la lista total de eventos de notificación obligatoria en Uruguay.
Los campos de datos de las notificaciones son los siguientes:

- Nombre y apellido del paciente, creado para representar nombres típicos en español.
- Cédula de Identidad: Número de documento de identificación del paciente, consistente en un número entero de 7 cifras más un dígito verificador.
- Evento, es la enfermedad sospechada. Algunas enfermedades con características y/o síntomas similares pueden agruparse en un único evento, como es el caso de arbovirosis (que agrupa las sospechas de dengue, chikungunya y zika, siendo todas enfermedades virales transmitidas por el mosquito Aedes aegypti, que tienen varios síntomas parecidos, incluyendo fiebre, erupción cutánea, artralgia/mialgia, cefalea, fatiga/debilidad, y síntomas digestivos (náuseas y vómitos).
- Fecha de inicio de síntomas: generada al azar dentro de cierto rango.
- Fecha de nacimiento y edad: Son creadas de forma consistente para que la edad (en la fecha de inicio de síntomas) concuerde con la fecha de nacimiento.
- Fecha de Consulta: Día en que el paciente acudió al proveedor de servicios. Debe ser igual o mayor (pero cercana) al inicio de síntomas.
- Internación y fecha de internación: Si el paciente ha sido ingresado y la fecha de la misma, consistente con lo anterior.
- Institución de Internación: En el caso de haber internación, se elige de una lista.
- Sector de Internación: CTI o Cuidados moderados
- Cobertura (Prestador de Salud): Institución médica a la que está afiliado el paciente, a elegir de una lista
- Departamento, localidad y dirección: Son elegidas de una lista de direcciones existentes del correo uruguayo (datos libres) de forma de generar información geolocalizable de forma coherente.
- Teléfono de Contacto: Teléfono de contacto del paciente, generado al azar.
- Datos clínicos y paraclínicos: Información sobre signos y síntomas presentes en el paciente y datos de laboratorio y/o estudios imagenológicos, según el evento considerado.

- Datos de laboratorio: Por separado de las notificaciones se genera (al azar) los estudios diagnósticos de laboratorio para algunos de los eventos notificados para pacientes, de forma consistente (se eligirá al azar cierto porcentaje de los casos notificados). Esto conforma la base de datos del laboratorio, donde los estudios tienen un tipo (serología o PCR), un estado (enviado, recibido, en proceso o finalizado) y un resultado (positivo, negativo o indeterminado). La fecha de extracción de muestras es coherente con lo anterior.
- Datos de defunciones: De forma análoga se selecciona la defunción al azar de algunos pacientes, elegidos de las notificaciones originales, y con características coherentes a lo notificado.

Las notificaciones, generadas con Faker (ver cuaderno Jupyter Notebook de Python, “Datos_sinteticos_notificaciones.ipynb”), se salvan en formato json, que luego es leído por Azure Databricks, generando los mensajes que son enviados al topic “notificaciones” de un servidor de Confluent Kafka, con un intervalo de tiempo entre mensaje y mensaje. El código correspondiente a la generación del flujo de mensajes desde Databricks se proporciona en el archivo “Fuente de notificaciones.dbc”. Adicionalmente, desde el mismo cuaderno Jupyter Notebook también se generó un archivo csv con resultados de laboratorio consistentes con las notificaciones, a partir del cuál se creó la base de datos Azure SQL (Ver figura A1).

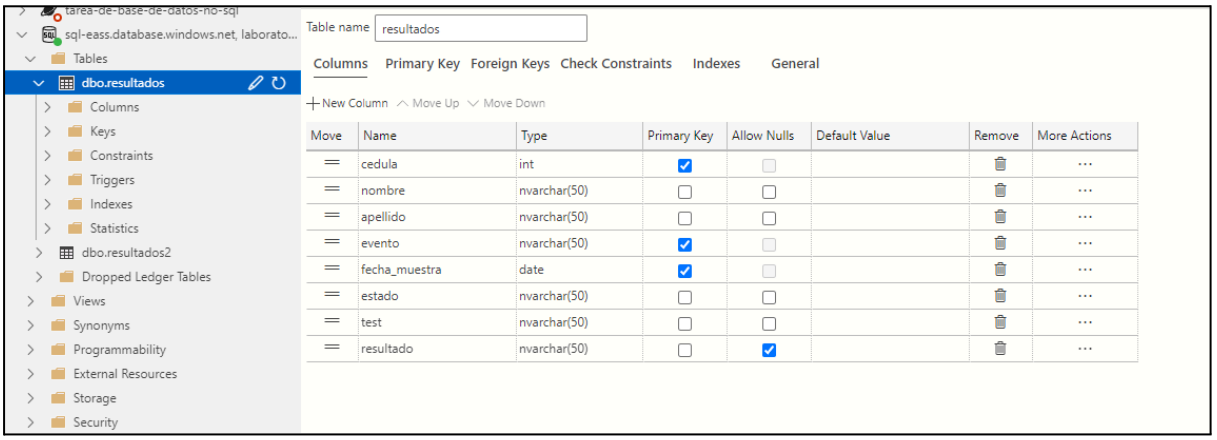


Figura A1: Captura de Azure Data Studio, mostrando el esquema de la tabla de resultados en la base de datos SQL del laboratorio con datos generados con Faker.

Anexo 2: Detalles del modelo de Favorabilidad para el Dengue.

Como se definió en el glosario, la favorabilidad en ecología busca modelar que tan apta es una región para que prospere un organismo en ella, más allá del nivel de presencia (prevalencia en el caso del dengue) del mismo en la región. Para este modelo de regresión logística los autores identificaron como características relevantes (features) para la regresión: la temperatura media anual, la distancia media a la ciudad más cercana, la altitud media del terreno, la densidad de

población, la cantidad de precipitación acumulada en el trimestre más frío del año, la orientación Norte-Sur del terreno, la variabilidad máxima de precipitación acumulada estacional, y una variable polinómica calculada a partir de latitud y la longitud, llamada dengue_TS. Estos valores son tomados promediando los valores ubicados el espacio dentro de polígonos (en este caso cuadrados de 0,5°x 0,5° de latitud por longitud) en que se divide el mapa de Latinoamérica, los cuales son numerados con un valor FID para su identificación (figura A2.1 y A2.2). La variable dependiente es la presencia (1) o ausencia (0) de dengue en cada uno de esos polígonos a nivel histórico. La regresión por lo tanto ajusta el Logit(p) donde p es la probabilidad de presencia de dengue en un cuadrado dado.

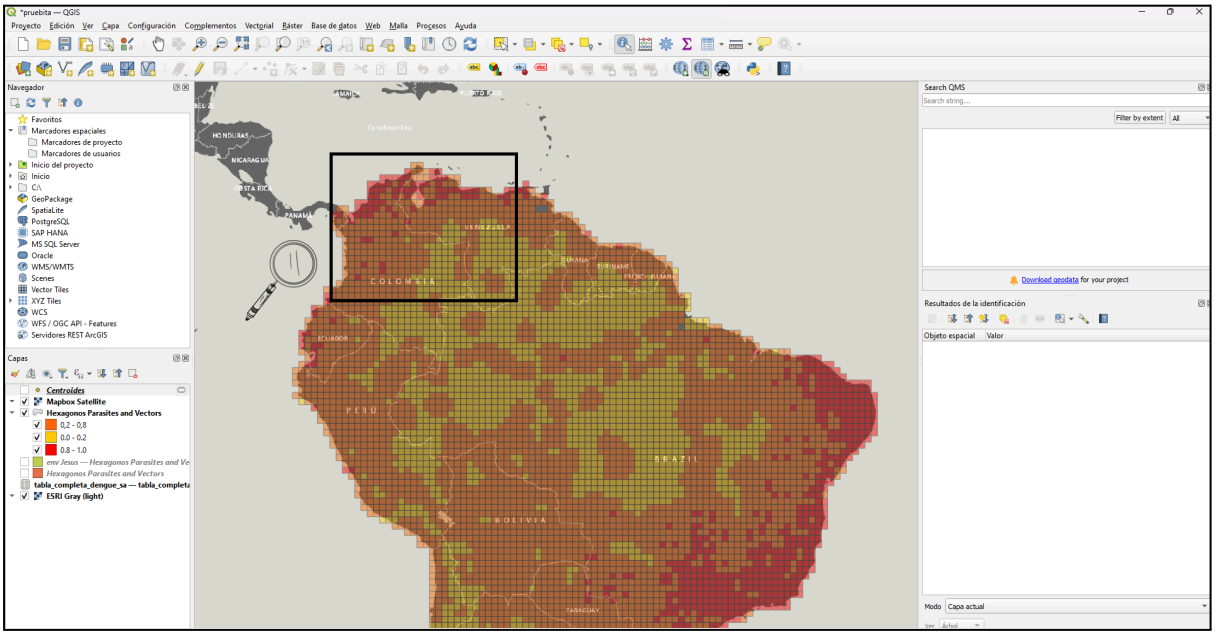


Figura A2.1: Fragmento ampliado en A2.2 que muestra la división en cuadrados de Latinoamérica para la construcción del Modelo de Favorabilidad.

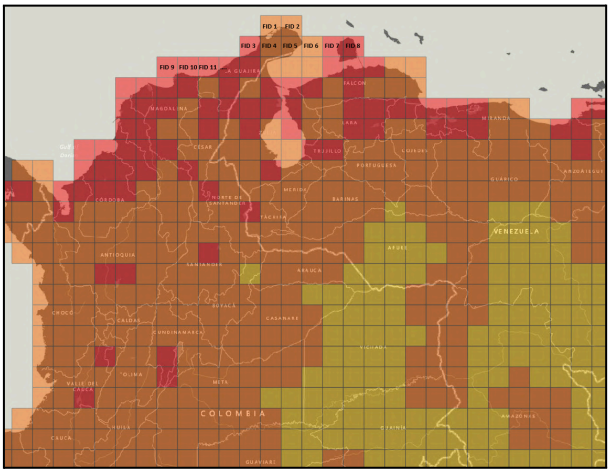


Figura A2.2: La zona ampliada muestra el comienzo de la numeración, que va recorriendo todos los cuadrados que contengan territorio latinoamericano, con

números comenzando del 1 en adelante, de izquierda a derecha y de arriba a abajo, asociando un FID a cada zona.

Con la intención de actualizar los datos para la región de Uruguay, en función de los casos recientes de dengue, como se dijo anteriormente, se evalúa, para cada cuadrado que contiene territorio uruguayo que no esté ocupado por un 1, si en el presente período hubo presencia (1) o ausencia (0) de dengue. Para saber a que cuadrado corresponde un caso, calculo el centro del mismo (centroide del polígono) y comparo las coordenadas de latitud y longitud de la ocurrencia del caso con las de dicho centroide (Figura A2.3). Si tanto longitud como latitud del caso difieren al mismo tiempo en no más ±0,25° con respecto a las coordenadas del centroide para un cuadrado con un FID determinado, podemos decir que el caso se halla dentro de ese cuadrado, y entonces, si su valor de presencia era cero, podemos actualizarlo a al valor 1.

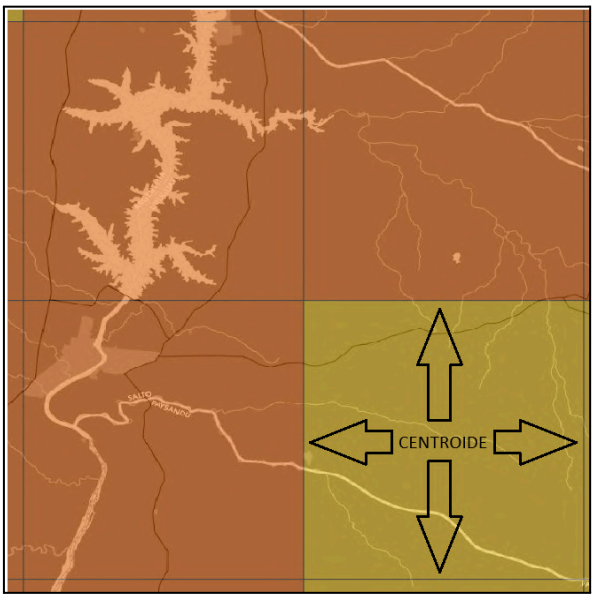


Figura A2.3: Detalle de la ubicación del centroide para uno de los cuadrados en que se divide el mapa.

Anexo 3: Detalles sobre el corredor endémico y la curva epidémica

La curva epidémica, simplemente gráfica, para un cierto año, la incidencia de casos nuevos confirmados en función de la semana epidémica en que estos casos nuevos comenzaron los síntomas. Este tipo de curva permite identificar la variabilidad del surgimiento de casos nuevos en función de la estacionalidad, y detectar patrones de periodicidad o cambios abruptos.
En cuanto al corredor endémico, este se puede realizar, tanto para la incidencia (casos nuevos) como para la incidencia acumulada (suma de casos nuevos acumulados desde la semana epidemiológica 1 del año epidemiológico en curso y la semana en curso en función de la semana epidemiológica. Este último es de elección, por razones estadísticas, cuando el número de casos semanal es un valor

pequeño. Entonces, a partir de los datos de la incidencia acumulada (o la incidencia) en función del número de semana epidemiológica, para varios años consecutivos ya cerrados (entre 5 o 10 años atrás), para cada semana epidemiológica se calcula la media (por algún medio) y un intervalo de confianza al 95%. La metodología concreta para calcular ese valor central y los extremos de ese intervalo de confianza variable en el tiempo, depende del comportamiento de los datos, pudiendo utilizarse la media aritmética, la mediana o la media geométrica, y sus respectivos intervalos de confianza. Adicionalmente se superpone la curva de incidencia acumulada (o incidencia) para compararla con el comportamiento previo y evaluar si es o no anómalo. Para ilustrar esto se pinta la zona inferior al extremo menor del intervalo de confianza como verde (zona segura, la incidencia es aun menor que el mínimo esperable, obviamente esta es la zona deseable), amarillo para la zona entre el extremo inferior y el valor medio, naranja para la zona entre el valor medio y el extremo superior del intervalo, y rojo para valores por encima del intervalo, indicando niveles de alarma creciente, siendo la zona roja, la correspondiente a un brote propiamente dicho, es decir un valor completamente por encima de lo esperado para esa semana epidemiológica. La figura A3.1 muestra un ejemplo de corredor endémico para incidencia acumulada de un evento y la figura A3.2 muestra un corredor endémico para la incidencia (casos nuevos) de un evento. Por lo general los corredores se construyen para las 52 semanas del año, pero a veces se restringen a períodos más cortos, asociados a la presencia de casos de la enfermedad

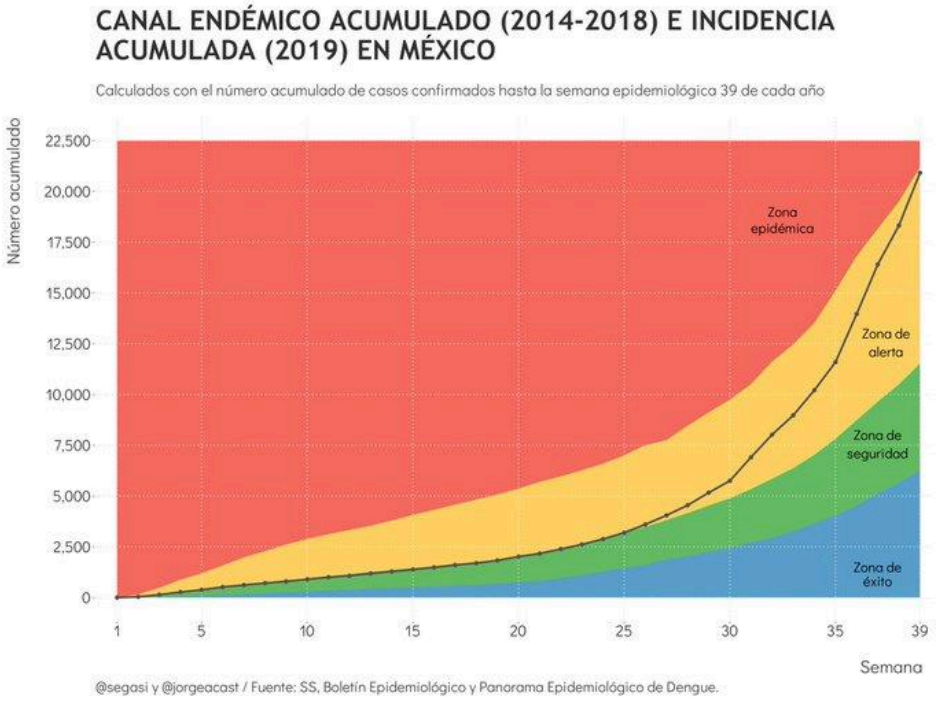
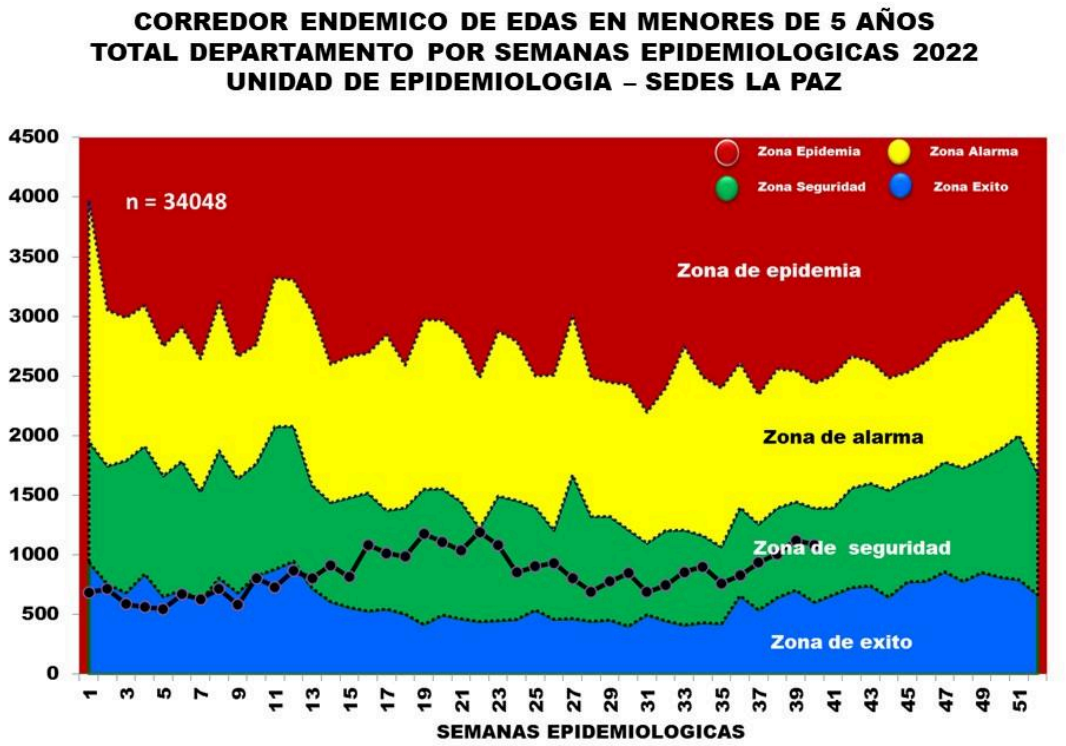


Figura A.3.1: Corredor endémico acumulado de Dengue en México correspondiente al período 2014-2018, superpuesto con la curva de incidencia acumulada para 2019.



A.3.2. Corredor endémico de enfermedad diarreica aguda para la incidencia en menores de 5 años en Bolivia, comparado con la curva epidémica de 2022.

GLOSARIO EPIDEMIOLÓGICO

A

Año epidemiológico: Período de tiempo utilizado para hacer seguimiento de los datos epidemiológicos, formado por entre 52 y 53 semanas epidemiológicas. Las semanas epidemiológicas constan de 7 días, comenzando siempre en Domingo y terminando en Sábado. El año epidemiológico debe comenzar el primer Domingo del año corriente, pudiendo comenzar antes o después del 1 de enero, siempre y cuando los días de desfasaje no sean más de 3. Por lo general los años epidemiológicos tienen 52 semanas, pero los años bisiestos tienen 53.

B

Biosociales (características): Factores relacionados con la interacción entre biología y aspectos sociales que influyen en la salud.

Brote (epidémico): Aumento súbito de casos de una enfermedad en una población específica. Para poder determinar cuando se da un brote, se necesita tener una estimación de cual es la cantidad “habitual” de casos de una enfermedad. Debido a que la mayoría de las enfermedades presenta comportamiento cíclico estacional, es habitual comparar la situación de una cierta semana epidemiológica (numerada del 1 al 52) con semanas de igual numeración en años anteriores.

C

Cambio climático: Es la alteración del clima, causada por la actividad humana. El fenómeno del calentamiento global, unido a la urbanización acelerada del hábitat natural de las especies salvajes, viene causando cambios en el comportamiento animal y aumentando su interacción con los humanos, lo que se ha identificado como una de las causas de enfermedades nuevas a partir de mutaciones de patógenos que afectan a los animales, como es el reciente caso del COVID-19 que se atribuye, dada su estructura genética, a una mutación de un coronavirus originario en los pangolines.

Caso confirmado: Se refiere a una persona que cumple con una definición operativa asociada a una evidencia razonable de la etiología de su enfermedad. La definición epidemiológica puede diferir del diagnóstico exacto con objetivos clínicos ya que en el primero se busca el lograr prevenir nuevos casos de forma efectiva y a tiempo, aunque sea a costa de tener algunos falsos positivos, por lo que a veces no se espera a la confirmación clínica total para tomar medidas preventivas, sino que se actúa ante una sospecha con alta probabilidad. En epidemiologia la definición de caso confirmado depende además de la contingencia local y temporal, debiendo adaptarse ante situaciones especiales.

Caso probable: persona con hallazgos típicos (signos y síntomas) compatibles con la enfermedad, sin confirmación por laboratorio

Chikungunya: Enfermedad viral transmitida por mosquitos aedes aegypty, que provoca fiebre y fuertes dolores articulares.

Contención (medidas de): Acciones tomadas para limitar la propagación de una enfermedad.

D

Defunción: Muerte de una persona, por cualquier causa. En particular en epidemiología es relevante fijar criterios para determinar la causa de muerte, ya que

no es lo mismo morir, por ejemplo, atropellado por un auto, teniendo un examen positivo para COVID-19, que morir efectivamente por una neumonía causada por este virus, si bien, en el caso de pulmonía es también es también relevante confirmar o descartar coinfecciones (es decir infecciones simultáneas) de otros microorganismos.

Dengue: Enfermedad viral transmitida por mosquitos del género *Aedes*.

Demográficos (datos): Información estadística relacionada con la población, como edad, sexo y ubicación geográfica.

E

Emergentes (enfermedades): Enfermedades que aparecen por primera vez o que están aumentando rápidamente en una población.
Epidemiología: Ciencia que estudia la incidencia, distribución y control de las enfermedades.

Eventos (de enfermedad): Casos individuales o agrupados de enfermedades reportadas. En el caso de un sistema de vigilancia epidemiológica, existen un conjunto de enfermedades, que pueden variar de país a país, cuya sospecha es de reporte obligatorio, en un cierto plazo después de detectada. Esta espera mínima puede ser de 24 horas o 1 semana, dependiendo de la gravedad de la enfermedad y la celeridad que requieran las medidas de prevención de contagios.

Etiología (de una enfermedad): Estudio de las causas y orígenes de una enfermedad. En lo que refiere a este trabajo, estamos evaluando la causa de enfermedades transmisibles, por lo que la etiología puede referir a una bacteria, un virus, un hongo o un parásito.

F

Favorabilidad: Término que describe las condiciones que favorecen la presencia de una especie en una cierta región geográfica, más allá de su prevalencia (es decir, un sitio puede ser favorable para la potencial presencia de una especie, pero esto es una cualidad potencial, que puede o no realizarse. Refiere a las condiciones que permiten la proliferación de cierto organismo en un medio de manera eficiente.

G

Geolocalización: Técnica de localizar un objeto o individuo en un espacio geográfico mediante tecnología.

Globalización: Interconexión mundial que facilita la rápida propagación de enfermedades.

Gripe aviar: Enfermedad respiratoria causada en humanos por una variante del virus de la influenza con origen en el virus que suele estar presente como causa de enfermedad en las aves.

I

Incidencia acumulada: Número total de casos nuevos de una enfermedad sumados en un período de tiempo. Se puede calcular de forma absoluta o en forma de relativa a la población susceptible de adquirir la enfermedad. Por lo general se multiplica por un número como 100000 para obtener valores manejables de forma intuitiva. Es más fácil entender que cada 100000 habitantes hay 5 casos en un año que entender que la tasa de casos es de 0,00005 casos/habitante al año,

Infección: Invasión y multiplicación de microorganismos en el cuerpo humano que provocan enfermedad.

Investigación (de campo): Esta actividad incluye la recolección de datos en el lugar donde se está produciendo un brote de enfermedad, tanto para buscar la causa de la misma (si se desconociera) como para buscar de forma activa posibles casos nuevos no reportados, e implementar de forma oportuna medidas preventivas, que pueden ir desde cosas sencillas como recomendaciones, hasta medidas más sofisticadas como la toma de medicamentos de forma preventiva o la vacunación.

L

Letalidad (de las enfermedades): Proporción de casos de una enfermedad que resultan en la muerte.

Leptospirosis: Enfermedad bacteriana transmitida por la orina de animales infectados, especialmente roedores.

M

Medidas preventivas: Estrategias de salud pública diseñadas para prevenir enfermedades antes de que ocurran.

Meningitis bacteriana: Inflamación de las membranas que rodean el cerebro y la médula espinal, causada por bacterias.

Meningitis viral: Inflamación de las membranas que rodean el cerebro y la médula espinal, causada por virus.

N

Notificaciones (de sospecha de enfermedades): Informes obligatorios a las autoridades de salud sobre casos sospechosos de enfermedades.

O

Ovitrampa: Dispositivo utilizado para monitorear y controlar poblaciones de mosquitos mediante la captura de sus huevos.

P

Pandemia: Brote de una enfermedad que se extiende a nivel mundial.

Patógenos: Microorganismos que causan enfermedades (virus, bacterias, hongos, parásitos).

PCR: Prueba de reacción en cadena de la polimerasa utilizada para detectar la presencia de material genético de patógenos. Se está detectando la presencia del microorganismo de manera directa.

Presentación clínica de una enfermedad: Síntomas característicos de los pacientes que presentan cierta enfermedad. Cuando estos síntomas son variables, se dice que la enfermedad tiene múltiples presentaciones.

Prevalencia: Cantidad de casos de una enfermedad que coexisten en un instante dado, independientemente de si son casos nuevos (con síntomas recientes) o no. La diferencia entre prevalencia e incidencia (referente a los casos nuevos que aparecen en un intervalo de tiempo) es especialmente notoria cuando se trata con enfermedades que tienen un desarrollo largo.

Profilaxis (medidas de): Tratamientos preventivos utilizados para evitar la aparición de enfermedades.

R

Reporte obligatorio: Obligatoriedad de informar ciertos casos de enfermedades a las autoridades de salud pública.

Resultado de PCR (positivo, negativo o indeterminado): Interpretación del análisis PCR que indica si una infección está presente o no.

S

Salto entre especies: Fenómeno en el que un patógeno se adapta para infectar una nueva especie.

Salud pública: Referente a las medidas políticas tomadas por el Estado para prevenir enfermedades o paliar sus consecuencias. Estas decisiones en lo posible deben estar basadas en evidencia empírica, que los epidemiólogos aportan.

Semana epidemiológica: Unidad de tiempo utilizada en los reportes de vigilancia epidemiológica. Cada semana epidemiológica comienza en Domingo y termina en Sábado. La primer semana de un año epidemiológico se numera como 1, y las subsiguientes van aumentando de 1 en 1 hasta llegar a 52 (o 53 los años bisiestos). De esta manera, se compara la presencia de enfermedad siempre en una semana epidemiológica actual con algún estimador del valor medio de los años anteriores. Por lo general, no se toman más de 5-10 años, debido al comportamiento variable a largo plazo, y muchas veces se omiten años pandémicos por ser de comportamiento anómalo.

Sensibilidad (de datos en Salud): Importancia de proteger los datos personales en el ámbito de la salud pública.

Serología: Estudio del suero de la sangre para identificar anticuerpos y diagnosticar infecciones. Los anticuerpos son parte de la respuesta inmune del cuerpo ante los patógenos, por lo que la detección de la enfermedad es indirecta, se observa la reacción de las células al microorganismo, pero no el microorganismo directamente. Esto es importante porque la utilización de este tipo de métodos requiere que los tiempos de toma de muestra sean suficientes para que el sistema inmune actúe, y su respuesta será muy diferente en una persona inmunodeprimida (es decir cuyas defensas son defectuosas) que en una persona inmunocompetente (cuyas defensas

actúan de forma normal).

Sistemas de vigilancia epidemiológica: Redes y procesos utilizados para recolectar y analizar datos sobre la salud pública.

U

Urbanización: Crecimiento de las áreas urbanas que puede influir en la aparición de enfermedades.

V

Vectores: Organismos vivos que transmiten enfermedades infecciosas (como los mosquitos).

Z

Zika: Enfermedad viral transmitida por mosquitos, que puede causar malformaciones congénitas.

Zoonóticas (enfermedades): Enfermedades que se transmiten de animales a humanos.