Development of a data processing system for epidemiological surveillance in Chubut: experience and lessons learned

Ignacio Guzmán Hernández[1], Lucía Pablovich[1], Luciano Perdomo[1,2], Leo Ordinez[1,2], and Julieta D'Andrea[3]

[1] Departamento de Informática Trelew, FI - UNPSJB, Trelew, Argentina
[2] Laboratorio de Investigación en Informática, FI - UNPSJB, Puerto Madryn, Argentina leo.ordinez@gmail.com
https://linvi.unp.edu.ar/
[3] Dirección de Epidemiología, Secretaría de Salud, Rawson, Argentina
{abc,lncs}@uni-heidelberg.de

Abstract. Epidemiological surveillance involves timely decision-making for disease prevention and control. In Argentina, information on Notifiable Events (NOE) is centrally managed through the National Health Surveillance System (SNVS 2.0). This platform fulfills its function of registering and systematizing the more than 150 types of NOE across the country. Through an incremental and evolutionary delivery process of functional software prototypes, an alternative is presented for expanding epidemiological surveillance capabilities using data from SNVS 2.0 for the Chubut province. These capabilities include processing, storage, visualization, and report generation, along with a scalable architecture that allows for the incorporation of new, more advanced analytical methods. Experience shows that a gradual and continuous adoption of technology yields consistent results.

Keywords: epidemiological surveillance, software engineering, endemic corridor, evolutionary prototype, agile development

Desarrollo de un sistema de procesamiento de datos para vigilancia epidemiológica en Chubut: Experiencias y Aprendizajes

Abstract. La vigilancia epidemiológica implica la toma de decisiones oportunas para la prevención y control de enfermedades. En Argentina, la información sobre Eventos de Notificación Obligatoria (ENO) se realiza de forma centralizada en el Sistema Nacional de Vigilancia de la Salud (SNVS 2.0). Esta plataforma cumple con su función de registro y

sistematización de los más de 150 tipos ENO de todo el país. A través de un proceso de entrega incremental y evolutivo de prototipos de software funcionales, se presenta una alternativa para la ampliación de capacidades de vigilancia epidemiológica a partir de los datos provenientes del SNVS 2.0, para la jurisdicción Chubut. Esas capacidades son el procesamiento, almacenamiento, visualización y generación de reportes, junto con una arquitectura escalable que permite incorporar nuevos métodos de análisis más avanzados. La experiencia muestra que una adopción gradual y continua de tecnología genera resultados consistentes.

Keywords: vigilancia epidemiológica, ingeniería de software, corredor endémico, prototipado evolutivo, desarrollo ágil

1 Motivación

La vigilancia epidemiólogica constituye un instrumento central en la Salud Pública, permitiendo recolectar, analizar e interpretar datos sobre eventos de interés sanitario, generando información valiosa para la toma de decisiones y generación de políticas públicas dentro de un escenario cambiante y urgente.

En este contexto, la disponibilidad de datos oportunos, consistentes y correctamente procesados se vuelve fundamental para fortalecer la capacidad de respuesta de los sistemas de salud. Sin embargo, la variabilidad en la forma de registro y la necesidad de adaptación a nuevas estrategias de análisis presentan desafíos constantes para las áreas de epidemiología, donde se dedica mucho tiempo a tareas repetitivas, incrementando la carga operativa y dificultando la reproducibilidad de los reportes debido al error humano y la poca trazabilidad y auditoría en los procesos.

El Sistema Nacional de Vigilancia de la Salud (SNVS 2.0) (Ministerio de Salud de la Nación Argentina, 2026b) es el sistema donde actualmente se recopila información sobre los Eventos de Notificación Obligatoria (ENO) para ponerlos a disposición de quienes deban tomar decisiones de salud pública, en particular, para este trabajo, la Dirección de Epidemiología de Chubut (DEC) (Dirección Provincial de Patologías Prevalentes y Epidemiología, 2026). Actualmente, la forma en la que allí se estaba trabajando consistía en importar los datos desde el SNVS 2.0 como planillas .csv, y luego realizar búsqueda, cálculos simples y gráficos con Hojas de cálculo. Esto significa que todas las tareas de procesamiento de datos se realizaban manualmente con muy poco soporte tecnológico.

Es importante mencionar la sensibilidad de los datos con los que se trabaja, ya que incluyen información personal e historia clínica de los ciudadanos, lo que implica confidencialidad a la hora de utilizarlos para pruebas, y el cumplimiento de requerimientos de seguridad. A los fines del trabajo aquí presentado, se establecieron mecanismos de anonimización y cláusulas de confidencialidad.

Dado esta problemática, se propuso desarrollar un sistema de soporte a la toma de deciciones que pudiera automatizar y optimizar el procesamiento, análisis y visualización de datos epidemiológicos. Este sistema se posiciona entonces como un instrumento de soporte consistente, trazable, extensible y seguro.

El objetivo de este trabajo radica en presentar la construcción de una herramienta operativa a la misma vez que se documenta la experiencia de desarrollo, los criterios adoptados y los aprendizajes que surgieron al trabajar en conjunto con la DEC.

2 Materiales y métodos

En virtud del Convenio entre la DEC y la FI - UNPSJB, que enmarca este trabajo y las características de ambas partes, se diseñó una estrategia de implementación basada en prototipos funcionales, que sirvieran al doble fin de colaborar en la definición de requerimientos y a la vez facilitaran las tareas de procesamiento cotidianas. Se pueden distinguir tres (Bortman, 1999) fases de trabajo: A) Prototipo inicial, basado en Google Colab(Google LLC, 2026), que sirvió para definir una primera versión del pipeline de procesamiento de datos y explorar el dataset, al tiempo de manipular los primeros ENO. B) Prototipo refinado, también basado en Google Colab, que permitió estructurar mejor el pipeline, modularizar e incorporar elementos interactivos. C) Prototipo de sistema, basado en una arquitectura web cliente-servidor, que utilizó las lógicas de procesamiento de los prototipos e incorpora roles de usuario, configurabilidad en las visualizaciones y una herramienta integrada de reporte.

2.1 Metodología de trabajo

Para la organización del proyecto de desarrollo se emplearon metodologías ágiles (Martin, 2013), dado que los requerimientos iniciales no se encontraban completamente definidos y fueron refinándose a partir de entregas funcionales incrementales y la retroalimentación de los interesados. Esto permitió adaptar el sistema de manera continua frente a cambios en las necesidades y en el entendimiento del dominio.

El convenio de trabajo no establecía plazos estrictos, y existía cierto grado de incertidumbre, asociado tanto a la aparición de situaciones sanitarias urgentes como a modificaciones en los criterios de notificación y clasificación de casos. En este contexto, se mantuvieron reuniones periódicas de seguimiento y una comunicación continua con los interesados.

El uso de metodologías ágiles favoreció el proceso de aprendizaje para ambas partes. El equipo de desarrollo incorporó conocimientos propios del dominio epidemiológico y el equipo de epidemiología profundizó su entendimiento sobre las características y capacidades que provee un sistema de información.

Como herramientas de soporte para la gestión del proyecto se utilizaron inicialmente GitHub Projects(GitHub, Inc., 2026b) y posteriormente Trello(Atlassian, 2026).

2.2 Estructuras de datos

Los conjuntos de datos utilizados como entrada se presentan en dos formatos principales: nominales y agrupados. Los datasets nominales consisten en registros de casos individuales, identificados mediante un código propio del SNVS 2.0. Estos registros contienen múltiples variables demográficas, clínicas y temporales, tales como edad, síntomas, comorbilidades, fechas relevantes (inicio de síntomas, notificación, toma de muestra, entre otras) y resultados de laboratorio, alcanzando más de 100 columnas.

Una característica de estos datasets es que la evolución de un mismo caso puede representarse mediante múltiples filas que comparten el mismo identificador, reflejando actualizaciones del mismo caso a lo largo del tiempo.

Por su parte, los datasets agrupados consisten en conteos semanales de casos, organizados según variables como establecimiento de salud, tipo de evento, sexo y grupo etario. Estos conjuntos no siempre son disjuntos respecto de los nominales, sino que los valores del conteo agregados pueden corresponder a los registrados individualmente.

Debido a la mayor profundidad de información, el análisis se basó principalmente en los datasets nominales, aunque se requirió la integración de ambos formatos.

2.3 Corredor endémico

Dentro de las herramientas estadísticas y de visualización que se utilizaron, el Corredor Endémico (CE) es la principal para el análisis del comportamiento temporal cíclico de los eventos sanitarios. Se genera a partir de datos históricos anuales para definir bandas de comportamiento esperado por semana epidemiológica y zonas de seguridad y alerta, permitiendo detectar desviaciones significativas respecto al patrón habitual.

Se implementó el método propuesto por Bortman Bortman, 1999, que construye el CE a partir de la media geométrica de las tasas de incidencia históricas y su intervalo de confianza. El procedimiento es el siguiente:

1. Para cada semana epidemiológica j ($j = 1, \ldots, 52$) se calculan las tasas de incidencia semanales de los n años históricos disponibles:
$$r_{ij} = \frac{c_{ij}}{P_i} \times 100\,000 \tag{1}$$
donde c_{ij} es el número de casos en la semana j del año i y P_i es la población del año i.
2. Se aplica una transformación logarítmica para aproximar la distribución a la normalidad:
$$y_{ij} = \ln(r_{ij} + 1) \tag{2}$$
3. Para cada semana j se calculan la media y el desvío estándar de los valores transformados:
$$\bar{y}_j = \frac{1}{n} \sum_{i=1}^{n} y_{ij}, \qquad s_j = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (y_{ij} - \bar{y}_j)^2} \tag{3}$$

4. Se construye el intervalo de confianza al 95% mediante la distribución t de Student con $n-1$ grados de libertad:

$$\mathrm{IC}_j = \bar{y}_j \pm t_{\alpha/2,\, n-1} \cdot \frac{s_j}{\sqrt{n}} \tag{4}$$

5. Se aplica la transformación inversa para obtener los límites en la escala original de tasas:

$$L_{\mathrm{inf},j} = e^{\bar{y}_j - t \cdot s_j/\sqrt{n}} - 1, \qquad M_j = e^{\bar{y}_j} - 1, \qquad L_{\mathrm{sup},j} = e^{\bar{y}_j + t \cdot s_j/\sqrt{n}} - 1 \tag{5}$$

Los valores resultantes definen tres zonas: la *zona de éxito*, por debajo del límite inferior del intervalo de confianza (L_{inf}); la *zona de seguridad*, entre el límite inferior y la media geométrica (M); y la *zona de alerta*, entre la media y el límite superior (L_{sup}). Los valores que superan el límite superior indican una situación epidémica. Se requiere un mínimo de tres períodos históricos y un máximo de siete para que el cálculo sea válido, ya que "las condiciones que mantienen la endemia como los criterios diagnósticos y los mecanismos de notificación y registro hayan cambiado" (Bortman, 1999, p. 3). Se excluyen los años 2020 y 2021 por el efecto distorsivo de la pandemia de COVID-19 sobre los patrones habituales de los eventos.

2.4 Requerimientos

En una visión retrospectiva de todo el proceso, considerando los tres prototipos descritos, las actividades generales que realiza el sistema son:

- Cargar los archivos extraídos del SNVS 2.0 o reportados a nivel nacional.
- Procesar los datos, realizando limpieza, normalización, consistencia y homogeinización.
- Consolidación de los datos en la base de datos propia, junto con cargas previas.
- Procesamiento individual de ENO, ya sea provenientes de dataset nominales o agrupados.
- Visualización de estadísticas descriptivas de ENO.
- Comparación de ENO
- Generación y visualización de CE asociado a un ENO.
- Previsualización, edición y generación de reporte publicable.

3 Desarrollo

El desarrollo del sistema se organizó en tres fases, en correspondencia con las etapas descritas en la sección 2: un prototipo inicial en Google Colab que sirvió para explorar los datos y validar los primeros cálculos, un prototipo refinado que modularizó la lógica en un paquete Python (Python Software Foundation, 2026) reutilizable, y el sistema web con arquitectura cliente-servidor. Cada fase trajo consigo aprendizajes que alimentaron las decisiones de la siguiente. A lo largo de todas ellas se trabajó en contacto directo con la DEC, lo que permitió que cada iteración incorporara retroalimentación concreta del equipo usuario.

3.1 Prototipado en Google Colab

El punto de partida fue un notebook monolítico en Google Colab que se implementó como base de trabajo para conocer el dominio y organizar los primeros procesos. Este notebook contenía todo en un solo archivo, desde la carga de datos nominales .csv exportados del SNVS 2.0 y el preprocesamiento con pandas (The pandas development team, 2026) hasta el cálculo de CE y la generación de gráficos con plotly (Plotly Technologies Inc., 2026).

Si bien cumplía su función como herramienta exploratoria, presentaba limitaciones importantes a la hora de escalar. Cualquier modificación requería navegar miles de líneas de código, no había separación entre lógica de negocio y presentación, y la reproducibilidad dependía enteramente de ejecutar las celdas en el orden correcto. En cuanto a la experiencia de usuario, tampoco resultaba ideal: era necesario correr celdas en orden, no desconectarse de la sesión, y los datos nuevos no se integraban con los anteriores a menos que se realizara una unión previa en un archivo csv.

Esta primera etapa requirió además comprender la estructura de la información proveniente del SNVS 2.0 (solo nominal en esta instancia), el manejo de datos tabulares mediante pandas y los criterios epidemiológicos detrás de los cálculos, como el CE, que requiere definir años históricos, calcular cuartiles y determinar zonas de éxito, seguridad, alerta y brote.

3.2 Refinamiento y modularización

A partir de estas limitaciones se realizó un esfuerzo de modularización sin abandonar Colab. Se organizó el código en un paquete Python con módulos separados para preprocesamiento, filtros de clasificación por evento, graficación, generación de informes y utilidades como el cálculo de semanas epidemiológicas Argentina.gob.ar, 2018. El paquete se versionaba en GitHub (GitHub, Inc., 2026a) y se instalaba en Colab mediante una celda que clonaba el repositorio y lo montaba como dependencia. Así, distintos notebooks podían consumir las mismas funciones y los cambios se propagaban sin duplicar código.

Esta etapa incluyó además la construcción de una base de datos local propia, consolidando las sucesivas cargas de archivos del SNVS en un repositorio unificado que permitiera comparar datos históricos sin depender de los archivos originales.

La modularización también permitió validar conceptos de diseño como la separación de responsabilidades y la configuración centralizada, dentro de un entorno que el equipo ya conocía y sin realizar demasiados cambios a la lógica de negocio, permitiendo luego una migración más controlada hacia la tercera fase.

3.3 Prototipo de sistema

Con requerimientos más claros, y habiendo validado tanto la lógica de procesamiento como los criterios epidemiológicos en las dos fases de prototipado,

se decidió pasar a un sistema con arquitectura cliente-servidor. Los prototipos habían cumplido su objetivo de iterar sobre los criterios con la DEC y entender qué necesitaba realmente el equipo.

Tanto los prototipos realizados en Google Colab como el sistema propiamente dicho son de uso interno por la DEC y protegidos por el Convenio de colaboración en que se enmarca este trabajo, con lo cual su apertura está limitada.

Arquitectura El sistema se compone de un backend en Python con FastAPI (FastAPI, 2026), una base de datos PostgreSQL (PostgreSQL Global Development Group, 2026) y un frontend en TypeScript con Next.js (Vercel, 2026). La comunicación entre ambos se realiza mediante una API REST cuyo contrato se genera automáticamente a partir de la especificación OpenAPI (OpenAPI Initiative, 2026) del backend, lo que garantiza consistencia de tipos entre cliente y servidor. Para operaciones que pueden demorar (como la carga masiva de archivos o la generación de reportes) se utiliza un esquema de tareas asíncronas con Celery (Celery Project, 2026) y Redis (Redis Ltd., 2026). De este modo la interfaz no se bloquea y el usuario puede seguir el progreso de la operación.

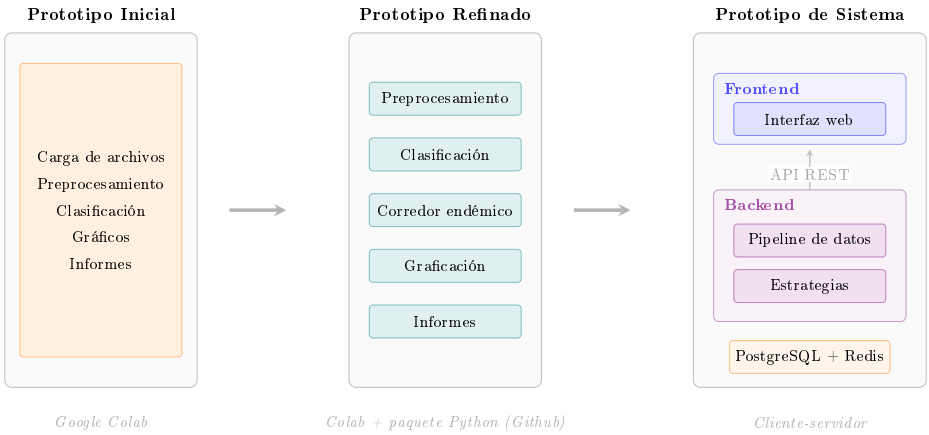

Fig. 1. Evolución de la arquitectura a lo largo de las tres fases de desarrollo.

La decisión de separar backend y frontend responde a necesidades concretas del dominio. El procesamiento de datos epidemiológicos requiere trazabilidad y consistencia, es decir, que un mismo archivo produzca siempre el mismo resultado bajo los mismos criterios. La interfaz, en cambio, debe priorizar la accesibilidad para usuarios que no son necesariamente técnicos. En el prototipo ambos componentes coexistían generando un acoplamiento excesivo y baja cohesión, comprometiendo así la robustez y mantenibilidad del sistema ante cualquier modificación.

Pipeline de procesamiento de datos El núcleo del backend es el pipeline que transforma los archivos exportados del SNVS en información estructurada y consultable. Cuando un usuario carga un archivo .csv o .xlsx a través de la interfaz, el sistema detecta automáticamente de qué tipo se trata inspeccionando las columnas presentes. Es importante notar que se soportan distintos tipos de archivos, estos son los de vigilancia nominal, agrupados clínicos, agrupados de laboratorio y de ocupación de camas por infecciones respiratorias agudas. A partir de esa detección se dispara el procesamiento correspondiente.

El archivo de datos nominal es el más detallado y, a la vez, el más complejo. Los datos vienen desnormalizados, con columnas de baja completitud que aportan poco valor analítico y otras cuyo contenido mezcla distintas responsabilidades. La estructura desnormalizada del archivo, se ve donde un mismo caso identificado por un ID puede generar decenas de filas, cada una con información diferente de muestras, síntomas y vacunas, por mencionar algunos ejemplos. Esto significa que los datos llegan en un formato plano donde relaciones uno a muchos se expresan mediante repetición de filas, lo que requiere un procesamiento cuidadoso para evitar duplicados y reconstruir la estructura real de cada caso. A la vez, las columnas que componen los datos presentan inconsistencias. Por ejemplo, los datos de *localidad* y *departamento*, frecuentemente intercambian valores, utilizan nombres diferentes (*e.g.*, "Puerto Madryn", "Pto. Madryn", "Madryn", etc.), usan abreviaturas y códigos, o juntan los dos datos en uno, entre otras.

Para los datos nominales, el pipeline valida el esquema, limpia y normaliza campos (fechas, textos, valores nulos) y luego aplica las estrategias de clasificación del evento epidemiológico. Un registro nominal puede contener más de cien campos que abarcan datos del ciudadano, información clínica, resultados de laboratorio, investigación epidemiológica, muestras, vacunación y tratamiento. El resultado se persiste en un modelo relacional de siete tablas vinculadas que permite consultas desagregadas por múltiples dimensiones.

Para los datos agrupados el procesamiento es diferente. Se valida la estructura según el formato detectado y se insertan los conteos semanales por evento, grupo de edad, sexo y establecimiento. Estos datos alimentan el motor de métricas para el análisis de tendencias poblacionales.

Estrategias de clasificación de casos Uno de los aspectos más relevantes del sistema desde el punto de vista epidemiológico es la clasificación automática de casos nominales. Como se describió en la sección anterior, un registro nominal puede contener más de cien campos con datos desnormalizados y de calidad variable. Analizar esa información manualmente o mediante planillas de cálculo para determinar la clasificación de cada caso resulta un proceso extremadamente laborioso y propenso a errores, sobre todo cuando cada evento de notificación obligatoria tiene criterios propios para determinar si un caso es confirmado, sospechoso, probable o descartado. Estos criterios dependen de combinaciones específicas de campos clínicos, resultados de laboratorio y datos epidemiológicos que varían según el tipo de evento, lo que hace imprescindible contar con un enfoque programático y ordenado.

En el prototipo de Google Colab, esta lógica estaba implementada como filtros individuales por evento. Cada filtro era una función que recibía un DataFrame y retornaba los casos clasificados según las reglas del evento correspondiente (dengue, tuberculosis, sífilis, VIH, síndrome urémico hemolítico, hantavirus, hidatidosis, hepatitis, coqueluche, entre otros). Esto era funcional y correcto, pero agregar un evento nuevo o modificar un criterio existente requería intervenir el código fuente y entender la estructura interna del filtro.

En el sistema actual esta lógica se generalizó mediante estrategias de clasificación almacenadas en base de datos. Cada estrategia define una lista ordenada de reglas que el sistema evalúa sobre los campos del registro. Las reglas pueden comparar valores exactos, verificar pertenencia a listas, buscar patrones en texto o aplicar funciones personalizadas para casos más complejos. También se incluyen detectores especializados como el de tipo de sujeto (humano, animal, indeterminado), necesario para eventos zoonóticos como rabia o mordedura.

Este diseño también responde a un hallazgo confirmado durante el desarrollo: las reglas de análisis en epidemiología son dinámicas y dependen de la complejidad y variabilidad del contexto. Los criterios de clasificación son distintos para cada enfermedad y están sujetos a modificación por parte de las autoridades sanitarias. Ante esta realidad, se consideró lo más adecuado aplicar un patrón Strategy en varios módulos del sistema, externalizando la lógica de procesamiento como reglas configurables almacenadas en base de datos. De este modo, la DEC puede agregar o modificar estrategias de clasificación sin necesidad de modificar el código fuente, lo que responde directamente al requerimiento de adaptabilidad ante nuevos criterios o nuevos eventos que puedan incorporarse al listado de ENO.

Motor de métricas y corredor endémico El sistema incluye un motor de métricas que ofrece una interfaz unificada para consultar indicadores epidemiológicos sobre las distintas fuentes de datos. Cada métrica se define declarativamente con sus dimensiones, fuente y tipo de cálculo. Al ejecutar una consulta, el motor selecciona el constructor apropiado según la fuente y compone los criterios de filtrado de forma combinable, permitiendo cruzar dimensiones temporales, geográficas y por tipo de evento.

Sobre este motor se implementa la generación del CE con los distintos métodos de cálculo.

En todos los casos se excluyen automáticamente los años 2020 y 2021 del cálculo histórico para evitar que la anomalía de la pandemia distorsione los corredores. El sistema valida que exista un mínimo de años históricos suficientes según el método elegido y señala cuando el CE no puede calcularse de forma confiable.

Generación de boletines epidemiológicos Los boletines epidemiológicos (Ministerio de Salud de la Nación Argentina, 2026a) son el principal producto de comunicación de la DEC, cada semana se publica un informe que resume la situación de los ENO bajo vigilancia. Antes del sistema, armar un boletín era un

trabajo artesanal. Había que generar tablas y gráficos, copiarlos a un procesador de texto, redactar el análisis ENO por ENO y revisar que los números coincidieran entre las distintas secciones. Cada semana se repetía el mismo esfuerzo, y cualquier error de transcripción podía pasar inadvertido hasta que alguien lo detectara en el documento ya publicado.

Para abordar esto, el sistema implementa un editor de boletines basado en bloques. Cada sección puede combinar texto enriquecido, tablas de datos, gráficos generados directamente desde las métricas del sistema e imágenes. El editor trabaja con plantillas en dos niveles: una plantilla de contenido estático (portada, autoridades, metodología) que se mantiene entre ediciones, y una plantilla de sección por ENO que se repite para cada enfermedad incluida en el boletín. Esta última soporta sustitución de variables como nombre del ENO, semana epidemiológica y período, de modo que al iniciar un nuevo boletín el sistema genera un borrador con la estructura ya armada y los datos precargados. El equipo parte de ahí para revisar, ajustar y redactar lo que haga falta.

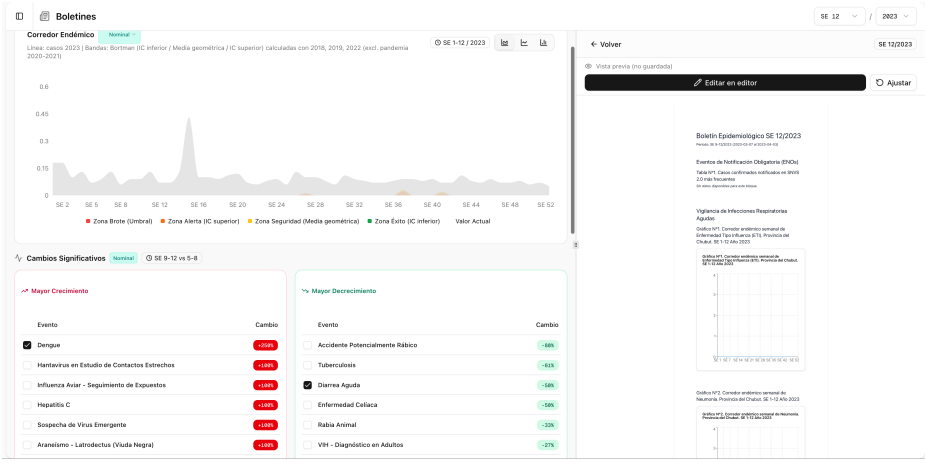

Fig. 2. Vista del corredor endémico con las zonas de Bortman (izquierda), análisis de cambios significativos por evento (izquierda) y editor de boletines con vista previa del documento (derecha).

Con esta aproximación se buscó un equilibrio entre automatización y control editorial. Automatizar completamente la generación de informes era tentador, pero durante el prototipado se observó que cada boletín necesita una contextualización narrativa que varía semana a semana, por ejemplo un brote inusual, un cambio de criterio, una aclaración metodológica. Esas decisiones editoriales son del equipo, por lo tanto se decidió preservar esa libertad editorial. Lo que sí puede hacer el sistema es ahorrar el trabajo mecánico de armar la estructura, insertar los gráficos y verificar que los datos sean consistentes.

Interfaz y visualización La interfaz se organiza alrededor de las tareas cotidianas del departamento. Un panel de métricas permite construir consultas combinando evento, período, dimensiones geográficas y tipo de indicador. La sección de vigilancia ofrece búsqueda de personas para localizar casos individuales y acceso a los registros nominales y agrupados. Un mapa presenta la distribución geográfica de eventos utilizando datos de georreferenciación de los establecimientos de salud.

La carga de archivos ilustra bien el cambio respecto al prototipo. En Colab había que ejecutar celdas en orden, esperar sin saber qué estaba pasando y confiar en que todo hubiera salido bien. Ahora el proceso es un flujo guiado que muestra el progreso del procesamiento, reporta errores de validación en un lenguaje comprensible y permite verificar los resultados antes de confirmar la incorporación de los datos. Que el usuario pueda entender qué salió mal y corregirlo sin asistencia técnica fue una de las motivaciones principales para construir una interfaz dedicada.

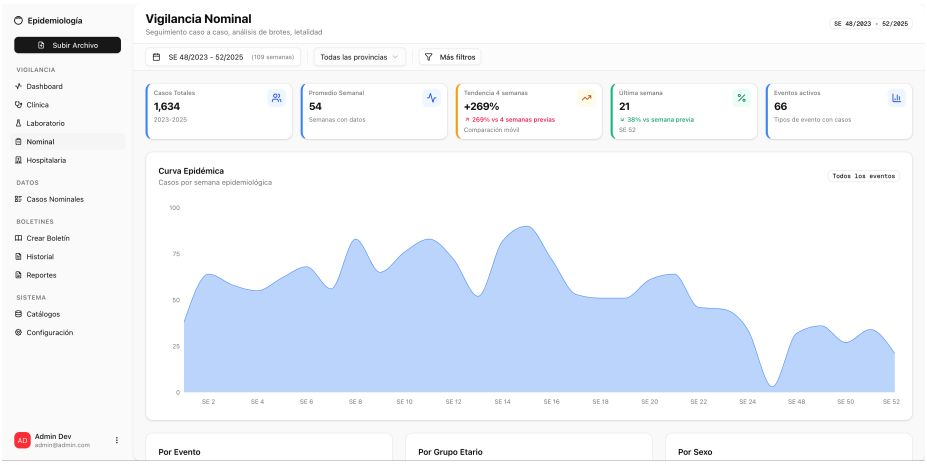

Fig. 3. Panel principal de vigilancia nominal. Se observan indicadores clave (casos totales, promedio semanal, tendencia, eventos activos), la curva epidémica por semana epidemiológica y el menú de navegación del sistema.

4 Discusión

El primer aporte de valor fue la limpieza de los datos de entrada, que luego se convirtió en un proceso ETL (extracción, transformación y carga) completo Amazon Web Services, Inc., n.d., lo que además de transformar los datos a un formato más útil genera cierta independencia del SNVS 2.0 y nuevas capacidades al cargarlos en un almacenamiento propio. En contraste con el flujo previo basado en planillas Excel, donde cada procesamiento era manual y no reproducible, el

pipeline garantiza que un mismo archivo produzca siempre el mismo resultado, aportando la trazabilidad que se identificó como necesidad en la motivación de este trabajo.

La implementación del CE de Bortman, con soporte para análisis semanal y acumulado, constituye un ejemplo de herramienta de análisis incorporada al sistema, y a la vez evidencia la capacidad de la propuesta para escalar y ampliar sus funcionalidades hacia nuevos instrumentos estadísticos.

El uso de metodologías ágiles, prototipado y diseño orientado a un sistema flexible permitió desarrollar prósperamente un sistema para un escenario cambiante que trata eventos urgentes con frecuencia, y que al inicio no tenía requerimientos claros. En particular, la organización en tres fases de prototipado permitió validar la logica de procesamiento con bajo costo previo a dedicar un esfuerzo mayor en una infraestructura más compleja.

El proceso de desarrollo siguió un esquema de aprendizaje interdisciplinario, en el cual se intercambiaron e integraron conocimientos del ámbito informático y epidemiológico. Esta construcción conjunta de conocimiento ayudó a ajustar el alcance del sistema a las necesidades del dominio.

Este trabajo resultó en una herramienta flexible, con capacidad de incorporar nuevos datos, y la visualización de diversas dimensiones y relaciones entre los ENO, favoreciendo la toma de decisiones sanitarias. Sin embargo, aún persiste la dependencia fuerte de la calidad y estructuración de los datos sistematizados en el SNVS 2.0, así como de la metodología de recolección de los mismos. Esta restricción limita la capacidad de expansión de funcionalidades a la vez que somete los criterios de manejo de datos a los establecidos por ese sistema. Por otro lado, la gobernanza, sostenibilidad y mantenimiento del sistema también presentan una dependencia marcada, en este caso de la articulación entre las instituciones, mediante el Convenio mencionado. En este caso, dicha dependencia podría sortearse promoviendo la generación de capacidades en la DEC que puedan darle sostenibilidad técnica al proyecto.

La herramienta de generación reportes dentro del sistema aporta una mejora significativa en la elaboración de informes periódicos, y más aún en contextos de brotes o situaciones emergentes, donde la disponibilidad de información oportuna es fundamental para dar una respuesta. Los boletines incorporan directamente gráficos y tablas del CE como bloques editables, junto con curvas epidemiológicas, distribuciones por edad y tablas de resumen, lo que elimina el paso manual de generar y copiar estas visualizaciones. Cabe notar que se optó por automatizar la estructura y la inserción de datos pero preservando el control editorial del equipo, dado que cada boletín requiere contextualización narrativa que varía según el contexto sanitario.

5 Conclusiones y Trabajos Futuros

En este trabajo se presentó un diseño de sistema que complementa al SNVS 2.0, permitiendo el procesamiento, análisis y visualización de datos epidemiológicos, así como la generación de reportes. El sistema provee la integración de datos

provenientes del SNVS 2.0 y mecanismos flexibles para realizar comparaciones y mediciones de diferentes ENO. En particular, el sistema integra de forma natural la herramienta denominada Corredor Endémico, en diferentes variantes.

Como trabajo futuro, se plantea la posibilidad de incorporar herramientas más avanzadas de análisis, específicamente modelos predictivos basados en aprendizaje automático, para detectar tendencias emergentes y generar alertas tempranas que fortalezcan la capacidad de respuesta en la vigilancia epidemiológica. Por otra parte, resulta de interés explorar la integración con fuentes de datos complementarias, como noticias, reportes de investigación de brotes e información climática, así como el cruce con determinantes sociales de la salud, lo que permitiría enriquecer el análisis y aportar una visión más integral del comportamiento de los eventos sanitarios.

References

Amazon Web Services, Inc. (n.d.). qué es extracción, transformación y carga (etl)? https://aws.amazon.com/es/what-is/etl/

Argentina.gob.ar. (2018). Epidemiología argentina. https://www.argentina.gob.ar/salud/epidemiologia

Atlassian. (2026). Trello [Accedido: 2026-04-01]. https://trello.com/

Bortman, M. (1999). Elaboración de corredores o canales endémicos mediante planillas de cálculo. *Revista Panamericana de Salud Pública*, *5*(1).

Celery Project. (2026). Celery distributed task queue [Accedido: 2026-04-01]. https://docs.celeryq.dev/

Dirección Provincial de Patologías Prevalentes y Epidemiología. (2026). Dirección de epidemiología [Ministerio de Salud de Chubut].

FastAPI. (2026). Fastapi framework [Accedido: 2026-04-01]. https://fastapi.tiangolo.com/

GitHub, Inc. (2026a). Github [Accedido: 2026-04-01]. https://github.com/

GitHub, Inc. (2026b). Github projects [Accedido: 2026-04-01]. https://github.com/features/issues

Google LLC. (2026). Google colaboratory [Accedido: 2026-04-01]. https://colab.research.google.com/

Martin, R. C. (2013). *Agile software development, principles, patterns, and practices*. BoD–Books on Demand.

Ministerio de Salud de la Nación Argentina. (2026a). Boletines epidemiológicos [Material de referencia].

Ministerio de Salud de la Nación Argentina. (2026b). Sistema integrado de información sanitaria argentino (sisa) [Accedido: 2026-04-01]. https://sisa.msal.gov.ar/sisa/

OpenAPI Initiative. (2026). Openapi specification [Accedido: 2026-04-01]. https://www.openapis.org/

Plotly Technologies Inc. (2026). Plotly [Accedido: 2026-04-01]. https://plotly.com/

PostgreSQL Global Development Group. (2026). Postgresql [Accedido: 2026-04-01]. https://www.postgresql.org/

Python Software Foundation. (2026). Python programming language [Accedido: 2026-04-01]. https://www.python.org/

Redis Ltd. (2026). Redis [Accedido: 2026-04-01]. https://redis.io/

The pandas development team. (2026). Pandas: Python data analysis library [Accedido: 2026-04-01]. https://pandas.pydata.org/

Vercel. (2026). Next.js [Accedido: 2026-04-01]. https://nextjs.org/