

# Solución web para la visualización de datos de simulaciones en salud

## Web-based tool for analysis and visualization of healthcare simulation data

Andrés Fabián Jiménez Torres<sup>1</sup> 

<sup>1</sup>Fundación Salutia Centro de Investigaciones en Economía, Gestión y Tecnologías en Salud. Bogotá, Colombia  
[andres.jimenez@salutia.org](mailto:andres.jimenez@salutia.org)

(Recibido: 8 febrero 2019; aceptado: 18 julio 2019)

### Resumen

En este artículo se presenta la construcción de un software que integró los diversos resultados de un simulador de los estados de salud de la población de Risaralda y los transformó en indicadores a través del uso de una bodega de datos, siendo estos representados en una solución web por distintos tipos de gráficos. El resultado obtenido fue un software con la capacidad de responder las consultas de los usuarios finales en un tiempo apropiado por medio de una interfaz gráfica con múltiples visualizaciones de fácil entendimiento que apoyó la toma de decisiones relacionadas con las políticas en salud del departamento. Todo el trabajo realizado en esta investigación mostró que el uso adecuado de las técnicas para el manejo de datos permite ahorrar en costos de infraestructura. Sumado a lo anterior, hace énfasis en la importancia de la correcta presentación de la información.

**Palabras clave:** Visualización, Crecimiento exponencial de datos, Procesamiento de datos, Representación de información, Comprensión del usuario final, Información sobre salud.

### Abstract

This article presents the design of a software that integrated several results from simulations on a population's health status and transformed them into indicators represented by different chart types. The result obtained was a software able to answer final users queries in a proper time and using a graphic interface with multiple understandable charts to support health policies decision making in the department. The investigation made work showed that correct use of data management technics allows to save in infrastructure costs, in addition to emphasizing in right data presentation importance.

**Keywords:** Visualization, Exponential data increase, Data processing, Information representation, Final user comprehension, Health information.

## 1. Introducción

La adecuada presentación de información por medio de visualizaciones tiene la capacidad de expresar grandes volúmenes de información, además de facilitar a los seres humanos la interpretación (Ware, 2013). En las soluciones basadas en internet, la velocidad de las consultas de la información y la concurrencia de usuarios son aspectos problemáticos a tener en cuenta en el diseño, que pueden modificar de forma positiva o negativa la experiencia del usuario final (Unger y Chandler, 2012).

Actualmente, los avances de las tecnologías de la información proveen suficientes herramientas para trabajar con grandes volúmenes de información en tiempos relativamente cortos. No obstante, la velocidad de respuesta de una consulta de información depende de la correcta planeación, conformación y configuración de los diferentes elementos que componen a los sistemas de información. Los componentes físicos como procesadores, memoria y discos deben ser cuidadosamente seleccionados; de igual manera, es



Cite this work as Jiménez Torres A. (2019). Solución web para la visualización de datos de simulaciones en salud. Revista Colombiana de Computación, 20(2), 37-55. <https://doi.org/10.29375/25392115.3720>

necesario realizar pruebas sobre el rendimiento de los contenedores de la información y el funcionamiento de los algoritmos de extracción (Ahmed y Smith, 2017).

Dentro del proceso de planeación de un sistema de información, es necesario hacer una estimación de cuántos usuarios finales van a hacer uso del sistema y qué tanta carga puede ejercer sobre este. Lo anterior se conoce como la concurrencia de usuarios y define la capacidad máxima de consultas en simultáneo que es capaz de responder el sistema de información. La concurrencia también sucede cuando dos o más actividades separadas ocurren de manera simultánea; es decir, cuando un único sistema computacional ejecuta múltiples actividades en paralelo (Williams, 2012).

Los procesos computacionales para extraer información de grandes volúmenes de datos, a pesar de los avances en las tecnologías de la información, tienen un alto costo tanto en tiempo como en uso de recursos físicos. En contextos reales, donde múltiples usuarios consultan la información, se requiere de una infraestructura (una base de datos, entre otras opciones) adecuada que permita cumplir con los estándares internacionales, por ejemplo, en el tiempo de espera de las peticiones de los usuarios (International Organization for Standardization, 2018).

Otra dificultad que presentan los grandes volúmenes de información es su representación y visualización. Las representaciones numéricas son una herramienta útil para la comprensión de un determinado fenómeno, pero, en grandes cantidades, pueden ser confusas. En respuesta a esta dificultad, uno de los más grandes beneficios de la presentación de datos a través de visualizaciones es la gran cantidad de información que puede ser interpretada, siempre y cuando el gráfico sea el adecuado (Ware, 2013).

En el diseño de visualización de datos, básicamente, existen dos tipos de gráficos: para representación y para exploración (Chen, Härdle y Unwin, 2008). Los gráficos para representación de información son estáticos, resumen lo que tratan de expresar y normalmente contienen explicaciones acerca de las variables representadas. Por otra parte, los gráficos para exploración de información se usan para expresar ideas de forma rápida y liviana, en vez de lenta y precisa. Ambos tipos de gráficos son útiles para un software enfocado en las visualizaciones, cuando el público objetivo es variado y cuando se busca presentar la información desde diferentes puntos de vista.

La Gobernación de Risaralda (Colombia), como parte de la estrategia para fortalecer las capacidades de gobernanza de la Secretaría de Salud del departamento (entidad rectora de las políticas públicas del sector salud en el territorio), promueve innovaciones basadas en ciencia y tecnología, aplicadas a la gestión pública y al buen gobierno (Gobernación de Risaralda, 2016). Para implementar dicha estrategia, en el departamento, se desarrolló el proyecto “Desarrollo de capacidades CT+I para investigación y simulación de políticas públicas en salud y seguridad social en el departamento de Risaralda” (abreviado SimuDatSalud), enfocado en el fortalecimiento de la toma de decisiones en el área de la salud y las políticas públicas en esta materia a través de la simulación de las condiciones socioeconómicas, de salubridad y del funcionamiento del sistema de salud en el departamento de Risaralda (Salutia Centro de Investigaciones en Salud, 2016).

El proyecto SimuDatSalud, Risaralda, contempló dos tipos de desarrollos computacionales: software de modelado y simulación y software de inteligencia institucional. Con respecto al software de modelado y simulación, el SimuDatSalud incluye cuatro simuladores en: (i) demografía, para simular la situación sociodemográfica del departamento; (ii) salud, para simular los factores y resultados del estado de salubridad de la población en el territorio; (iii) sistema de salud, para simular los comportamientos y costos del sistema de salud del departamento; y (iv) gestión del riesgo en salud, que es una aplicación particular de los anteriores simuladores a la enfermedad isquémica coronaria. En el SimuDatSalud se usó la técnica de la microsimulación con el fin de poder evaluar eventos de manera muy específica y conectar de manera coherente los cuatro simuladores (Salutia Centro de Investigaciones en Salud, 2016; Fundación Salutia, 2018).

El objetivo de este artículo es dar cuenta del proceso llevado a cabo para la construcción de una solución web para la visualización de los resultados de las microsimulaciones del estado de salud pasado, actual y futuro de la población de Risaralda. Dichos resultados son generados en el proyecto SimuDatSalud y se busca que apoyen la toma de decisiones de los usuarios en los distintos grupos de interés. Este desarrollo computacional requirió de la aplicación de una metodología para la construcción de una bodega de datos para optimizar el manejo y procesamiento, y de técnicas de representación gráfica y dinámica de la información que permiten presentar de manera rápida grandes volúmenes de información de forma clara y comprensible para los usuarios finales (alcaldes, gobernadores, estadísticos, funcionarios públicos del sector salud, representantes de las EPS e IPS, pacientes y ciudadanos en general).

Este artículo está organizado de la siguiente forma: la primera sección describe el método usado para la construcción del sistema de información, la selección de las herramientas informáticas necesarias para

la programación y los algoritmos que fueron usados en el motor de base de datos y en el código fuente del aplicativo de visualización; la segunda sección presenta los resultados obtenidos de la implementación de todas las técnicas descritas en la primera sección en relación con el usuario final, y la última sección discute y concluye los resultados del desarrollo de la herramienta con respecto a los resultados obtenidos por otras soluciones web para la visualización de información.

## 2. Método

Para el desarrollo de este trabajo y por las características que presentaba, se optó por usar la metodología de investigación aplicada descriptiva, que se enfoca en encontrar mecanismos que conlleven al logro de un objetivo definido, haciendo una descripción detallada de una situación, objeto o fenómeno. Respecto a la parte técnica, se optó por usar como base la metodología de construcción de bodegas de datos de Ralph Kimball, debido a su efectividad y aplicabilidad en situaciones como las que se presentaron en esta investigación (Kimball y Ross, 2013).

A continuación, se presenta el contexto general y los desafíos de la investigación, junto con una amplia descripción de los pasos que siguieron para dar solución a los problemas encontrados en el camino y cómo todo el conjunto de estas acciones llevó a obtener el resultado final.

Las simulaciones que genera el SimuDatSalud Risaralda se componen de una serie de archivos en formato plano, los cuales representan el comportamiento de la “sociedad artificial”, que se refiere al modelo computacional que simula el comportamiento de la población de Risaralda. Cada única simulación consta de un paquete de ocho archivos de granularidad individual por cada año, por cada uno de los años de simulación (2010 al 2050). Dado el alto volumen de información, que está presente en esos ocho archivos, se optó por la creación de indicadores para mostrar las características y comportamientos de la “sociedad artificial”, los cuales debían ser representados a través de diferentes tipos de visualizaciones estadísticas.

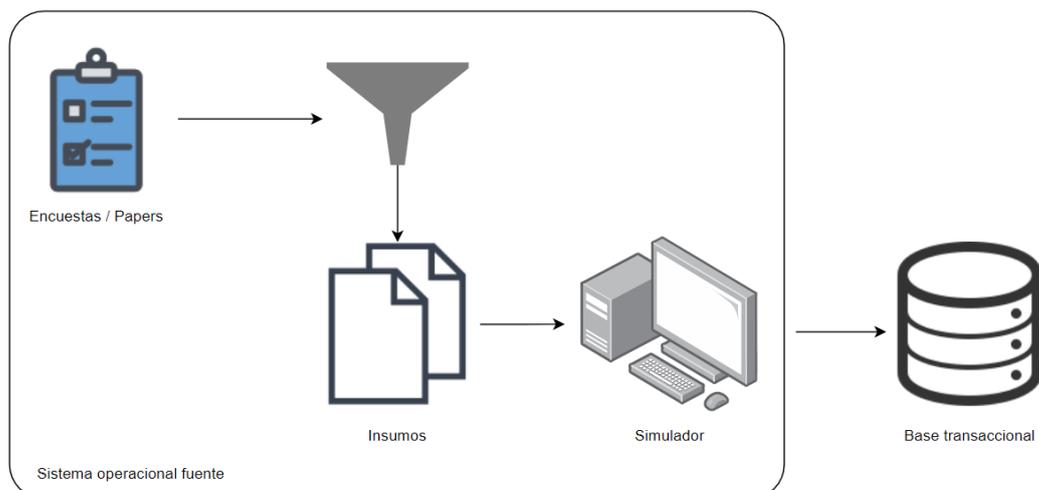
### 2.1 El proceso de transformación de la información

En el SimuDatSalud Risaralda, el manejo eficiente de sus grandes volúmenes de datos se abordó desde la perspectiva de las bases de datos dimensionales y la inteligencia de negocios (Kimball y Ross, 2013). El modelo de Kimball y Ross se enfoca en la construcción estratégica de una bodega de datos con el fin de ejecutar procesos de inteligencia de negocios de manera rápida y eficaz. Según este modelo, la correcta construcción de una bodega de datos está en la capacidad de responder a los siguientes requisitos de diseño: fácil acceso a la información, una presentación consistente de la misma y adaptabilidad a cambios y la temporalidad de esta información. El modelo propone cuatro componentes distintos que conforman el entorno de la bodega de datos: sistemas operacionales fuente (también conocidos como los sistemas transaccionales), proceso ETL (extracción, transformación y carga), y área de presentación de datos y aplicaciones de inteligencia de negocios (Kimball y Ross, 2013).

#### 2.1.1 Sistemas operacionales fuente

Los sistemas operacionales fuente capturan la información de las transacciones del negocio y hacen referencia al mínimo control que se puede llegar a tener sobre el contenido y el formato de esos datos (Kimball y Ross, 2013). En, el SimuDatSalud, los sistemas transaccionales son todos aquellos mecanismos de recolección de información empleados para describir el estado de salud de la población; en particular, la Encuesta Nacional de Situación Nutricional, ENSIN (Ministerio de Salud y Protección Social, 2015a), la Encuesta Nacional de Demografía y Salud (Ministerio de Salud y Protección Social, 2015b) y la Encuesta Nacional de Salud Pública (Ministerio de Salud y Protección Social, 2007).

Los datos de estos sistemas transaccionales no proceden de manera directa del proceso ETL, sino que se filtraron y depuraron en el flujo de información del sistema operacional fuente como insumo para el software de simulación (ver Figura 1).

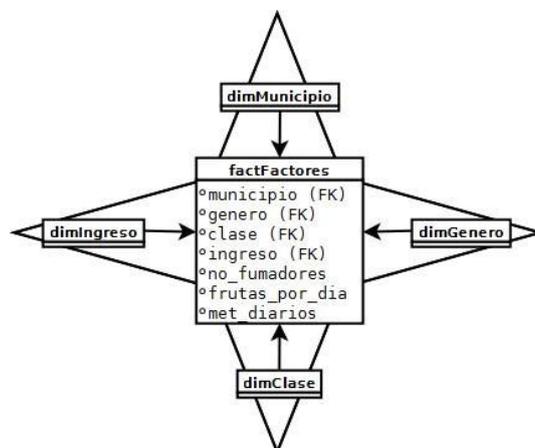


**Figura 1.** Diagrama del sistema operacional fuente

### 2.1.2 Selección de la arquitectura de la bodega de datos

Dependiendo del contexto y el uso que se le va a dar a la bodega de datos, se pueden aplicar diferentes arquitecturas de construcción para las cuales existe software especializado (por ejemplo: Talend, Pentaho, Tableau, entre otros). En la investigación SimuDatSalud Risaralda se utilizó el modelo dimensional, que es una solución informática que apoya la inteligencia de negocios, debido a la relativa simplicidad de su construcción, además de la posibilidad de ser implementado usando un motor de bases de datos transaccional (Rizzi, 2007). El esquema que se usó dentro del modelo dimensional es una organización en estrella (ver Figura 2).

Cada proceso del negocio es representado por un modelo dimensional. Se compone de una o más tablas de hechos, las cuales contienen mediciones de los eventos seleccionados que están descritos y contextualizados por las tablas de dimensión (Kimball y Ross, 2013). Por ejemplo, a partir de la Figura 2, se puede afirmar que el valor de la medida “frutas\_por\_día” está determinado por las dimensiones: municipio, género, ingreso y clase; es decir, la medida probablemente variará dependiendo de la dimensión que use para analizarla. Esta característica resultó útil en el SimuDatSalud Risaralda porque las consultas sobre los archivos resultantes de la simulación están a cargo y bajo criterio de los usuarios finales, quienes pueden elegir distintos filtros o cruces de los indicadores.



**Figura 2.** Diagrama del modelo dimensional en estrella de Kimball  
**Fuente:** Adaptado de Kimball y Ross (2013).

### 2.1.3 Selección del motor de base de datos

Como requisito no funcional, en el SimuDatSalud Risaralda, se debía usar únicamente software libre y sin costo alguno, excepto si era completamente necesario. Dentro del listado de motores de base de datos disponibles (PostgreSQL, MariaDB-MySQL, SQL Server y MongoDB), se eligió el motor PostgreSQL, debido a sus prestaciones con el manejo de los datos, a que no generó costos y por las herramientas que ofrece.

### 2.1.4 Definición de la granularidad

En el caso de una bodega de datos, la granularidad indica el significado de la información que expresa un único registro de la tabla de hechos (Kimball y Ross, 2013). En el diseño se definió que los resultados de las simulaciones fueran representados por medio de indicadores anuales, filtrados por características geográficas, del sistema de salud o del individuo. Este requisito de diseño definió la granularidad de la bodega de datos, según la cual un único registro de la bodega de datos representa una medición del estado o comportamiento de un grupo de individuos en un año concreto, especificado por las características geográficas, del sistema de salud o del individuo. Dicha definición se tradujo en rendimiento al momento de realizar las consultas (Tabla 1).

**Tabla 1.** Ejemplo de la configuración de la bodega de datos correspondiente a un año

Tipo_servicio	Sexo	Subregión	Clase	Frecuencia_uso	Intensidad_uso
1	1	1	1	3,27	1,02
2	1	3	1	4,02	1,51
6	2	3	3	3,95	1,37
22	0	0	0	1,01	0,89
22	1	0	0	1,23	0,93
22	2	0	0	0,94	0,78

### 2.1.5 Definición de las dimensiones y hechos

Las dimensiones son elementos que responden a preguntas como “quién, qué, dónde, cuándo, por qué y cómo” de los eventos del proceso del negocio (Kimball y Ross, 2013). En la investigación SimuDatSalud, el desarrollo de la bodega de datos siguió la metodología de Kimball y Ross (2013), según la cual los elementos de agrupación y filtrado fueron definidos en términos de dimensiones, siendo la dimensión más importante la correspondiente al filtro del tiempo, puesto que el objetivo de una bodega de datos es generar un registro histórico de la información.

Debido a la forma como están planteados los indicadores (medidas o hechos), la granularidad para el tiempo se definió únicamente de forma anual. Con base en esto, se redefinió la dimensión tiempo y se ubicó en el nombre de las tablas de hechos, con esto se obtuvo una mejoría considerable en la velocidad de las consultas. Del mismo modo, la dimensión política (entendida como las medidas que se toman en pro del mejoramiento del estado de salud de la población de Risaralda) también fue ajustada de esa forma. Las tablas se nombran como se muestra a continuación:

- factRiesgoEnfermar\_2010: se guarda la información de los indicadores de riesgo de enfermar del año 2010. No hay política pública aplicada.
- factRiesgoEnfermar\_51100\_2010: se guarda la información de los indicadores de riesgo de enfermar del año 2010 que están siendo afectados por la política 51100 (código que expresa un impuesto a las bebidas azucaradas de un 10%).

La Tabla 1 muestra un ejemplo de la estructura e implementación de una tabla de hechos según lo que se ha descrito anteriormente.

Este método, implícitamente, hace un uso eficiente de las estructuras de datos (grafos) que funcionan al interior del motor de base de datos debido a la forma en que se indexa la información al interior de las tablas de hechos, además de la propia indexación del nombre de dichas tablas.

#### 2.1.6 Proceso de extracción, transformación y carga (ETL)

Esta es un área de trabajo compuesta por estructuras de datos y un conjunto de procesos. Este proceso define una clara división entre los sistemas operacionales fuente y el área de presentación de la información (Kimball y Ross, 2013). La idea principal de este proceso es simplificar los datos a través del uso de medidas predefinidas que se almacenan en tablas de hechos, enlazadas con tablas de dimensiones (filtros), de modo que su posterior consulta, así como su actualización, sea más ágil.

Para la implementación del proceso ETL, se usó el lenguaje de programación R, el mismo en que fue desarrollado el simulador de la investigación SimuDatSalud Risaralda. Este lenguaje se define como un lenguaje de procesamiento estadístico computacional, ideal para la definición de los indicadores (medidas) (The R Foundation, 2019). En el diseño de la solución web, cada indicador estaba acompañado de dos medidas adicionales, correspondientes a los intervalos de confianza, los cuales están basados en desviaciones estándar y en percentiles a lo largo de simulaciones tipo Montecarlo. La definición del proceso ETL se dividió en tres subprocesos: distribuciones, momentos y carga a la bodega de datos.

- Distribuciones: se calcularon todas las medidas con 0,1 y 2 filtros. Este proceso se calculó por cada año. Cada registro generado se identificó con el año, la política y la simulación. Se generaron tantos archivos, en formato RData, como años y simulaciones se definan.
- Momentos: se tomaron los archivos generados en el proceso anterior y se calcularon la media, la desviación estándar y los límites inferior y superior de cada uno de los indicadores. Al finalizar este proceso, se generaron tres archivos en formato RData: uno con 0 filtros, otro con 1 filtro y el último con 2 filtros.
- Carga a la bodega de datos: se tomaron los tres archivos generados en el proceso anterior y se cargaron a las tablas de dimensiones y de hechos. Se realizó la carga de manera paralela, es decir, se carga más de un registro en simultáneo.

En estos subprocesos, las distribuciones corresponden a la extracción y levemente a la transformación parcial; los momentos, a la transformación completa; y la carga, a la carga de los datos procesados.

## 2.2 Selección del lenguaje de programación

Aunque el lenguaje Python ofrece herramientas para el análisis avanzado de datos y se acopla adecuadamente con distintas bibliotecas de gráficos estadísticos y científicos (Python Software Foundation, 2019), se descartó porque carece de soporte nativo y no tiene una trayectoria en el desarrollo de aplicaciones web dirigidas a usuarios no especializados, además porque el lenguaje R (que se usó para la construcción del simulador) cumple las mismas funciones. De manera similar, el uso de la interfaz web que ofrece el lenguaje R, que está dedicado al análisis de datos y no a la interacción con usuarios finales (The R Foundation, 2019), se descartó porque requiere el pago por su uso cuando se trata de aplicaciones web. También se descartó el uso del lenguaje Java, por los problemas de inestabilidad que se presentan en las versiones 8 y 9 del JDK y JRE de Java (Oracle, 2019). Finalmente, se eligió el lenguaje PHP (versión 5.6) porque demanda poco uso de recursos para el despliegue de servicios web y cuenta con un número significativo de frameworks, como Laravel y CodeIgniter (The PHP Group, 2019).

## 2.3 Selección y generación de los gráficos

La selección de la biblioteca sobre la cual se iban a trabajar los gráficos se determinó por dos factores: soporte y costos de uso comercial. Aunque Google Charts ofrece soporte a largo plazo y es gratuita, esta biblioteca se descartó porque ofrece poca variedad de gráficos enfocados al ámbito científico (Google Developers, 2019). Por el contrario, se eligió la biblioteca Plotly.JS (Plotly, 2019) porque ofrece gratuitamente una amplia cantidad de gráficos enfocados al ámbito científico.

## 2.4 Procesos y algoritmos especiales

A medida que se desarrolló el simulador SimuDatSalud, se implementaron pequeñas soluciones que ayudaron con problemas puntuales y resultaron ser eficientes e importantes. A continuación, se describe cada uno de estos aprendizajes.

### 2.4.1 La extensión “pg\_prewarm” de PostgreSQL

Una de las técnicas usadas por la computación moderna es el uso de memorias de acceso rápido (RAM, ROM, caché del procesador) diferentes a las destinadas para el almacenamiento bruto con el fin de poder ejecutar procesos de manera más rápida (Jacob, Ng y Wang, 2008). La extensión “pg\_prewarm” es una funcionalidad del motor de base de datos PostgreSQL, la cual se basa en esta técnica. Se trata de un espacio que se reserva en la memoria de acceso rápido del computador o servidor, la cual se usa para almacenar elementos de uso frecuente. Dicha extensión fuerza la carga de objetos en esta memoria, de modo que se puede decidir qué elementos se almacenarán. De forma nativa, PostgreSQL hace uso de esta memoria, pero depende de la frecuencia de uso que tengan los elementos del sistema a través del tiempo (The PostgreSQL Global Development Group, 2019).

La razón y fuente de inspiración del uso de este recurso está en uno de los principios de los cubos multidimensionales tipo Multidimensional Online Analytical Processing (MOLAP), el cual hace uso de la memoria caché para optimizar el rendimiento de las consultas de información. Respecto de la bodega de datos que se construyó para el desarrollo, la idea ejecutada fue cargar todas las tablas del modelo dimensional a la memoria caché de la base de datos, de modo que, aparte de la optimización que ofrecía en sí el modelo, se podía aún reducir más los tiempos de consulta.

Para hacer uso de esta funcionalidad se procedió de la siguiente forma. En primer lugar, fue necesario crear la extensión en el esquema que están las tablas que se deseen cargar en la memoria por medio de la siguiente instrucción:

```
CREATE EXTENSION pg_prewarm
```

Posterior a este proceso, se ejecuta la instrucción para cargar los objetos deseados a la memoria caché, por medio de la siguiente instrucción:

```
SELECT pg_prewarm(«esquema».«objeto»,«método»)
```

Este método de carga de objetos en la memoria se ve directamente afectado por la configuración del tamaño de la memoria caché destinada en la base de datos. Dado el caso de que se cargue un número de objetos superior al que dicha memoria puede almacenar, esta eliminará los objetos más antiguos con el fin de responder a la petición.

### 2.4.2 Los ceros faltantes

Al trabajar con los datos provenientes de las consultas realizadas en la base de datos, cuando se hacía uso de la instrucción *group by* en conjunto con algunas condiciones (*where* de la consulta), el dominio de la variable llegaba incompleto y eso representaba un problema a la hora de hacer las gráficas estadísticas. Esta dificultad se ilustra en la Tabla 2, donde se suministran unos datos acerca del riesgo de enfermar que tiene un grupo de individuos, cuya variable sexo tiene el siguiente dominio: {1: mujer, 2: hombre}.

**Tabla 2.** Ejemplo de datos para riesgo de enfermar

ID	Sexo	riesgo_enfermar
1	1	0,20
2	1	0,15
3	2	0,40
4	1	0,30

5	2	0,35
6	2	0,10

**Fuente:** consola

En el ejemplo (Tabla 1), se requiere determinar cuántas personas, por cada sexo, tienen un riesgo mayor o igual a 0,35, para lo cual se ejecuta la siguiente instrucción del lenguaje de programación R:

tabla [,sum(riesgo\_enfermar >= 0,35), by = .(sexo)]

El resultado de lo anterior se muestra en la Tabla 3. Como se aprecia en esta Tabla 3, solo hay uno de los dos sexos presentes. Lo que normalmente un programador esperaría en este tipo de consulta es que, si no están presentes en la función de agregación se muestren un cero (0), es decir, ver el sexo 1 con conteo cero (0) tal como se muestra en la Tabla 4.

**Tabla 3.** Resultados del ejemplo

Sexo	Conteo
2	2

**Fuente:** consola

**Tabla 4.** Resultado del ejemplo con inclusión de ceros

Sexo	Conteo
1	0
2	2

**Fuente:** consola

Esta situación, aunque parece simple, puede causar muchas molestias, siendo más notorio el efecto cuando se usa más de una variable en el *group by* de la consulta, lo cual causa que sea más complicado determinar los ceros faltantes. Considere el ejemplo que se muestra en la Tabla 5.

**Tabla 5.** Ejemplo con resultado de dos variables de agrupación

Sexo	Edad	Conteo
1	5 a 9 años	2
1	40 a 49 años	2
2	25 a 29 años	5

**Fuente:** consola

Para dar solución a este inconveniente, se optó por dos soluciones. La primera consistió en usar condicionales en el código fuente, el cual mediante operaciones matemáticas determinaba dónde debía agregar o no el valor cero (0). Durante un periodo considerable de tiempo este fue el método empleado para subsanar dicha dificultad, pero, era una solución poco escalable y muy difícil de mantener.

La segunda solución se propuso pensando en la escalabilidad y en la simplicidad: matrices. La idea se formó a partir del análisis de los datos que provenían de la consulta y se determinó que estos presentaban un comportamiento de coordenada fila-columna.

A continuación, se presenta una consulta que se le hace a una tabla:

SELECT calculo FROM tabla WHERE condiciones GROUP BY a,b

Se va a decir que “a” tiene un dominio de tres (3) elementos y “b” tiene un dominio de dos (2). El resultado de esta consulta podría ser como el que se presenta en la Tabla 6.

**Tabla 6.** Resultado de la consulta con agrupación por variables a y b

A	b	Proceso
1	2	23,50
3	1	11,34
3	2	15,97

**Fuente:** consola

Para detectar de manera simple los ceros (0) que hacen falta, se procede construyendo una matriz de “a” filas y “b” columnas y, por cada fila de la Tabla 6, se va a posicionar el campo “proceso” en la posición [“a”, “b”] como se muestra en la Tabla 7.

**Tabla 7.** Matriz resultante con datos faltantes

-	1	2
1	0	23,5
2	0	0
3	11,34	15,97

**Fuente:** consola

De esta forma, sumado a que se tiene una organización más limpia de los datos, se pueden identificar los datos en cero (0) que la consulta no mostró: (a = 1, b = 1), (a = 2, b = 1) y (a = 2, b = 2). Para el caso de variables como la edad, presente en la Tabla 5, que no son de tipo numérico, es necesario hacer una serie numérica, es decir, asignar un número entero consecutivo a cada uno de los elementos del dominio.

### 3. Resultados

El resultado del diseño de la solución web, con todos los parámetros y características establecidas en SimuDatSalud Risaralda, correspondió a una aplicación web sobre PHP 5.6, alimentada por una bodega de datos construida sobre PostgreSQL que destaca en velocidad y concurrencia de usuarios, además de permitir visualizar y analizar rápidamente una enorme cantidad de datos con unos simples pasos y, lo más importante, dirigida a un amplio rango de tipos de usuario. A continuación, se explica cada una de estas características.

#### 3.1 La bodega de datos

Como se describió en el método, se hicieron algunas modificaciones a la metodología de Ralph Kimball en cuanto a la construcción de las tablas de hechos conforme a las necesidades del proyecto. El total de tablas generadas es de 862. Corresponden la gran mayoría a tablas de hechos y son generadas automáticamente por un procedimiento almacenado en PostgreSQL, como se muestra en la Figura 3.

```

FOR i IN ano_ini..ano_fin LOOP
/* Tablas de factores */
IF(politica != '+') THEN
  finalQuery = 'drop table if exists "Investigacion2DWH".factFactores_'||politica||'_'||i||'';
  llave = 'CONSTRAINT factFactores_'||politica||'_'||i||'_pkey PRIMARY KEY (pre_getar, pre_municipio, pre_subregion, pre_sexo, pre_zona, pre_neduca, pre_quintil)';
  finalQuery = finalQuery || 'create table "Investigacion2DWH".factFactores_'||politica||'_'||i||''('||camposFactores||llave||)';
ELSE
  finalQuery = 'drop table if exists "Investigacion2DWH".factFactores_'||i||'';
  llave = 'CONSTRAINT factFactores_'||i||'_pkey PRIMARY KEY (pre_getar, pre_municipio, pre_subregion, pre_sexo, pre_zona, pre_neduca, pre_quintil)';
  finalQuery = finalQuery || 'create table "Investigacion2DWH".factFactores_'||i||''('||camposFactores||llave||)';
END IF;

execute finalQuery;

```

**Figura 3.** Código de muestra del procedimiento almacenado que genera las tablas de hechos

**Fuente:** Salutia Centro de Investigaciones en Salud (2016).

Se crearon 4 tipos de tablas de hechos base:

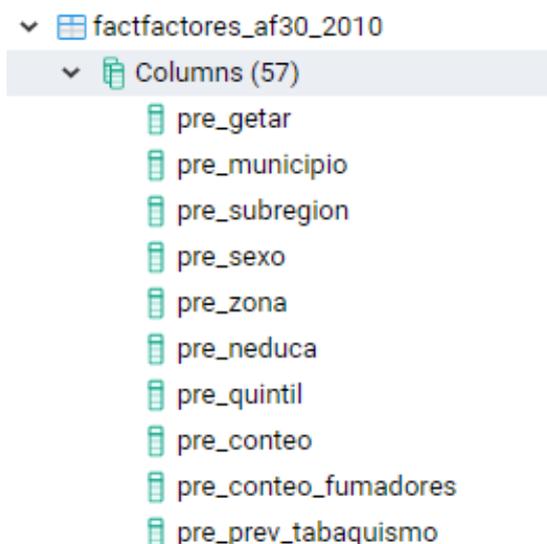
- factFactores: tablas que contienen toda la información de los factores que causan las enfermedades en la población.
- factRiMorbilidadPrio: tablas que contienen las probabilidades de estar en riesgo de enfermar de la población.
- factMorbilidadPrio: tablas que contienen la probabilidad de enfermar de la población.
- factMortalidadPrio: tablas que contienen la probabilidad de fallecer de la población.

Como se puede evidenciar, la construcción de dichas tablas corresponde a la historia natural de la enfermedad, un esquema de organización que permite una fácil comprensión de la distribución de la información en relación con la naturaleza de la investigación.

Conforme a estos tipos de tablas de hechos, se crearon tantos como años y políticas en salud se habían definido y de ahí se explica el enorme número de tablas. Una tabla final luce de la siguiente forma:

*factFactores\_af30\_2019*

Donde “*factFactores*” corresponde al tipo de tabla de hechos base con la que se relaciona, “*af30*” corresponde a un código de una política de salud y “*2019*” corresponde al año en el cual se genera la información almacenada en su interior. Cada tabla estaba indexada por cada una de sus dimensiones y contenían relativamente pocos registros, por lo cual la extracción de información se hace de forma más limpia y rápida. La Figura 4 muestra un ejemplo de la implementación de estas tablas.



**Figura 4.** Ejemplo de una tabla de hechos implementada

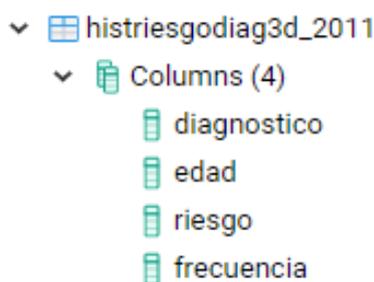
**Fuente:** Salutia Centro de Investigaciones en Salud (2016).

Para la construcción del proceso ETL de la bodega de datos se decidió utilizar el lenguaje de programación R por su versatilidad respecto al manejo de grandes cantidades de datos. Conforme a la

descripción del contexto, los orígenes de datos eran grandes archivos producidos por el mismo lenguaje R, correspondientes a los resultados de las simulaciones por año y política de salud (un archivo particular contenía aproximadamente 35 columnas y un millón de registros).

El ETL construido en el lenguaje de programación R tomaba los archivos fuente de las simulaciones, hacía los cálculos de los indicadores con todas las desagregaciones posibles y de acuerdo con la temática de los indicadores, los cargaba en alguna de las 4 tablas de hechos base de la bodega de datos teniendo en cuenta el año y la política de salud presente. El proceso se realizaba usando técnicas de computación paralela provistas por la biblioteca “*rparallel*”. Lo anterior significó hacer un manejo adecuado de los recursos disponibles del servidor para evitar su colapso.

Junto con las tablas correspondientes a la bodega de datos, se incluyeron otras que no respetan estrictamente la metodología de Ralph Kimball, pero que eran necesarias para poder producir de manera rápida determinadas visualizaciones especiales. Un ejemplo de este tipo de tablas son las que producen un histograma en tres dimensiones de las enfermedades priorizadas en la investigación. Un ejemplo de este tipo de tablas se presenta en la Figura 5.



**Figura 5.** Ejemplo de tabla para producir gráfico en 3 dimensiones

**Fuente:** Salutia Centro de Investigaciones en Salud (2016).

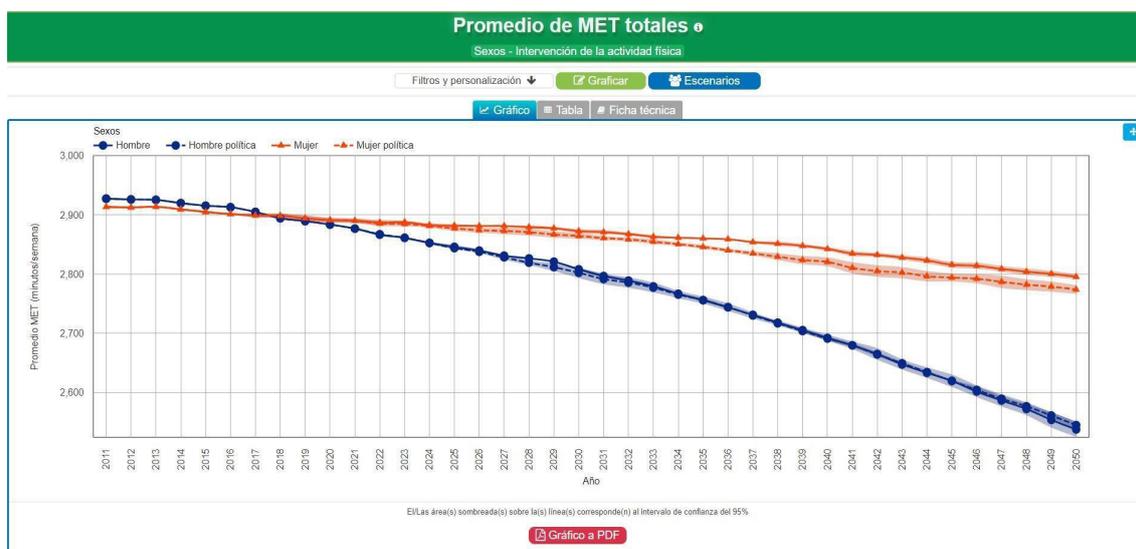
### 3.2 Concurrencia y velocidad

Independientemente del navegador o el hardware del computador donde se ejecute, si se exige al aplicativo realizar una consulta de alta complejidad (por ejemplo, un indicador desde el año 2011 hasta el 2050, desagregado por género del individuo, que compare los resultados de la implementación de una política pública de actividad física), con una concurrencia de 70 usuarios simultáneos, realizando la prueba desde un equipo remoto ajeno a la red donde está el servidor y con un servicio de internet con un ancho de banda de 5 MB se obtienen los resultados que se muestran en la Tabla 8.

**Tabla 8.** Línea de tiempo de una consulta compleja de información

Tiempo (segundos)	Proceso
0''	El usuario selecciona los filtros y envía la petición.
0,3''	El servidor, mediante PHP, comienza la interpretación de la petición del usuario.
0,5''	Comienza la consulta a la bodega de datos.
2,5''	La base de datos entrega el resultado de la consulta.
2,7''	PHP termina de procesar los datos de la consulta y hace el llamado a las vistas para mostrar el resultado.
3,7''	El usuario observa el resultado de la consulta en su terminal.

Se puede apreciar en los resultados que los tiempos de respuesta al usuario no son los más cortos, por ejemplo, comparado con GapMinder (Gapminder, 2018), que es un conocido software diseñado para visualizar información en economía, salud, medio ambiente, educación, demografía y salud, a nivel mundial, de manera gráfica y comprensible para los usuarios (Rosling y Zhang, 2011). Sin embargo, si se



**Figura 6.** Vista del indicador promedio Met (unidad de medida del índice metabólico) total desagregado por sexo y con implementación de una política pública

**Fuente:** Salutia Centro de Investigaciones en Salud (2016).

tienen en cuenta las limitaciones de hardware y software que se tuvieron en este proyecto, los resultados obtenidos son aceptables, además de cumplir con las expectativas que tenían los usuarios objetivo.

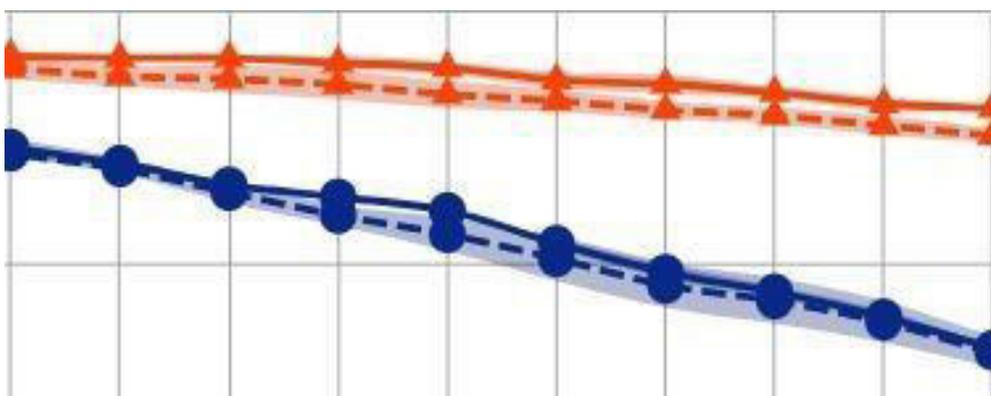
### 3.3 Interfaz gráfica del usuario

La simplicidad y el tamaño del gráfico estadístico fueron los principales criterios para la construcción de la interfaz gráfica. Del mismo modo que ocurre con los teléfonos móviles en la actualidad, la relación del contenido con respecto al cuerpo del dispositivo sobrepasa el 70%. Conforme a esto, el tamaño del gráfico estadístico respecto a la porción visible del cuerpo del sitio web está cercano al 75% (ver Figura 6).

Los componentes más importantes para destacar son los siguientes (Figura 6):

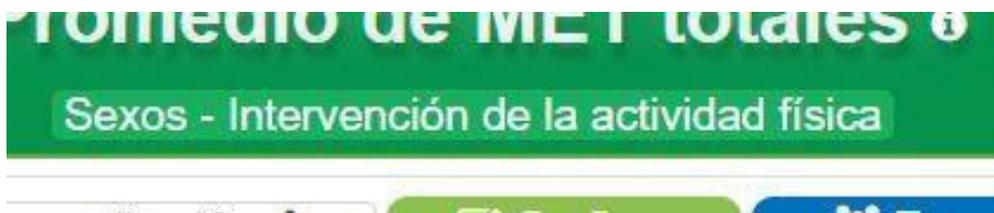
- Tipo de gráfico presentado: esta visualización es un gráfico que representa de manera clara y sencilla la tendencia de la información a lo largo del tiempo.
- Presentación de la política pública: de manera paralela, se presentan los resultados de la implementación de la política, en el mismo color de la serie, con una diferenciación por medio de una línea punteada (ver Figura 7).
- Fácil visualización de los datos de la consulta: al usar muchos filtros a la hora de hacer una consulta, puede resultar difícil recordarlos posteriormente; para facilitar esto, en la parte superior del gráfico, una sección muestra los datos consultados (ver Figura 8).
- Posibilidad de enfocar el gráfico en áreas específicas: cuando el comportamiento de dos series se sobrepone, es difícil visualizar lo que realmente está pasando. Para superar esta dificultad se implementó la posibilidad de hacer acercamientos en puntos específicos (ver Figura 9).

En esta solución web, también es posible realizar consultas de información con mayor detalle, que se denominan microvisualizaciones, las cuales son la representación más pura de los resultados generados en los procesos de microsimulación a través de gráficos estadísticos. En esta solución web, se optó por el uso de histogramas porque permiten mostrar de manera muy detallada el comportamiento y tendencia de una variable en un momento específico y a través del tiempo. Esta visualización se presentó en dos versiones: la primera, un histograma tradicional en dos dimensiones (ver Figura 10) y, la segunda, un histograma en tres dimensiones (ver Figura 11).



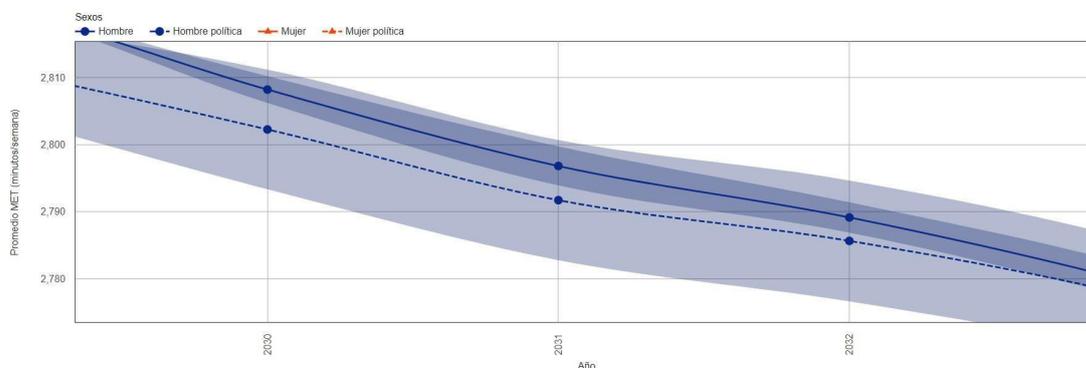
**Figura 7.** Línea punteada y continua en representación de la política

Fuente: Salutia Centro de Investigaciones en Salud (2016).



**Figura 8.** Filtrados usados en una consulta

Fuente: Salutia Centro de Investigaciones en Salud (2016).



**Figura 9.** Uso del acercamiento en un gráfico

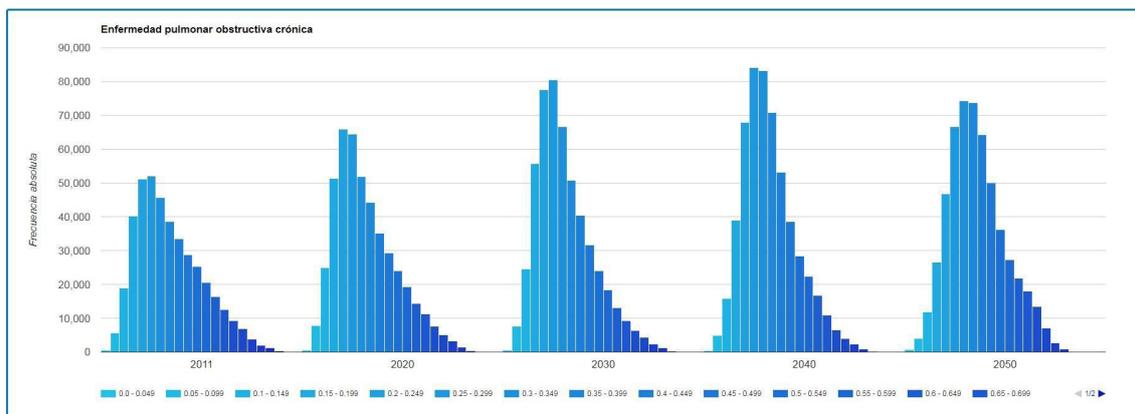
Fuente: Salutia Centro de Investigaciones en Salud (2016).

El histograma en dos dimensiones (Figura 10) muestra la distribución de alguna de las enfermedades priorizadas, en SimuDatSalud Risaralda, que se modelaron desde el año 2010 hasta el año 2050, en decenios. Esta visualización está pensada para usuarios con conocimientos básicos en estadística.

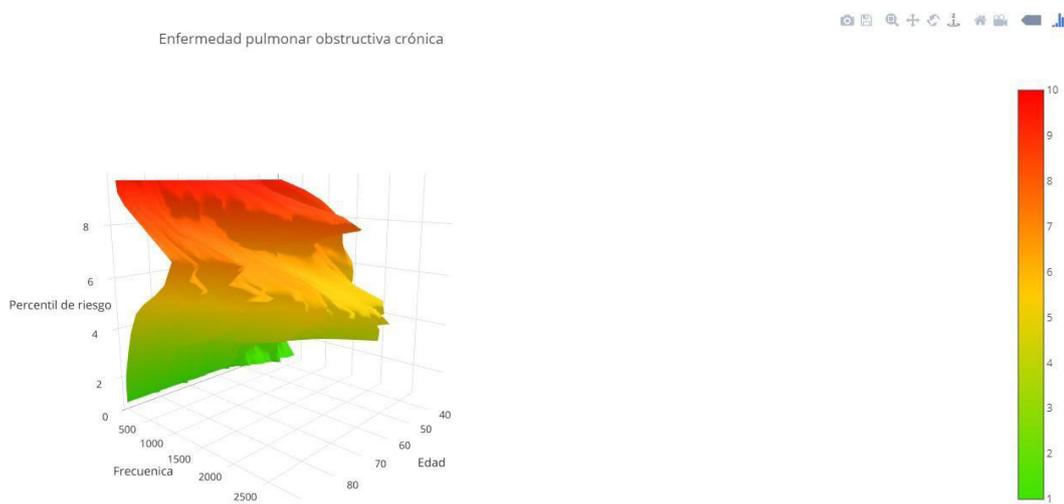
El histograma en tres dimensiones (Figura 11) muestra la distribución de alguna de las enfermedades priorizadas que se modelaron en un año específico e incluye las dimensiones de percentil de riesgo y edad. Esta visualización está pensada para usuarios con conocimientos básicos en estadística y un buen nivel en interpretación de gráficos multidimensionales.

### 3.4 Validación de los resultados

La bodega de datos diseñada para esta investigación fue puesta a prueba tanto en su rendimiento como en su calidad de construcción mediante dos tipos de pruebas: la primera, enfocada al rendimiento, fue una prueba de estrés al sistema en sus diferentes capas y la segunda, enfocada a la calidad, fue una prueba de escalabilidad.



**Figura 10.** Ejemplo de histograma en dos dimensiones  
**Fuente:** Salutia Centro de Investigaciones en Salud (2016).



**Figura 11.** Ejemplo de histograma en tres dimensiones  
**Fuente:** Salutia Centro de Investigaciones en Salud (2016).

Las pruebas de estrés se realizaron usando la herramienta JMeter y se plantearon en dos momentos: la respuesta desde la bodega de datos y la respuesta desde la solución web. Los resultados de este ejercicio mostraron que la bodega de datos estaba en capacidad de responder 500 peticiones realizadas simultáneamente y 900 peticiones distribuidas en intervalos de 3 segundos, y la solución web estaba en capacidad de responder 250 peticiones simultáneamente y 450 peticiones en intervalos de 3 segundos en el caso de la consulta más compleja para ambos casos y con la configuración del servidor usado.

Los anteriores resultados se ajustaron satisfactoriamente a las necesidades y limitaciones de presupuesto que se tenían estipuladas, además de superar en velocidad en ciertas ocasiones a software como Health Metrics (Institute for Health Metrics and Evaluation, 2019). En el caso de la pérdida de rendimiento de la solución web con respecto a la bodega de datos, se trata de falta de capacidad del servidor que responde las peticiones HTTP, es decir, para igualar los valores es necesario aumentar los recursos de procesador y memoria RAM de este servidor.

Las pruebas de escalabilidad consistieron en la adición de dimensiones y medidas (indicadores) a las tablas ya existentes, conforme a los nuevos requerimientos que se presentaron en el desarrollo de la investigación. Lo esperado era que dichos ajustes no significaran un esfuerzo mayor, al menos para la bodega de datos. Las pruebas mostraron que la estructura propuesta para la bodega de datos es escalable, pero dependiendo de la complejidad del indicador, el proceso ETL en R puede requerir de más o menos trabajo.

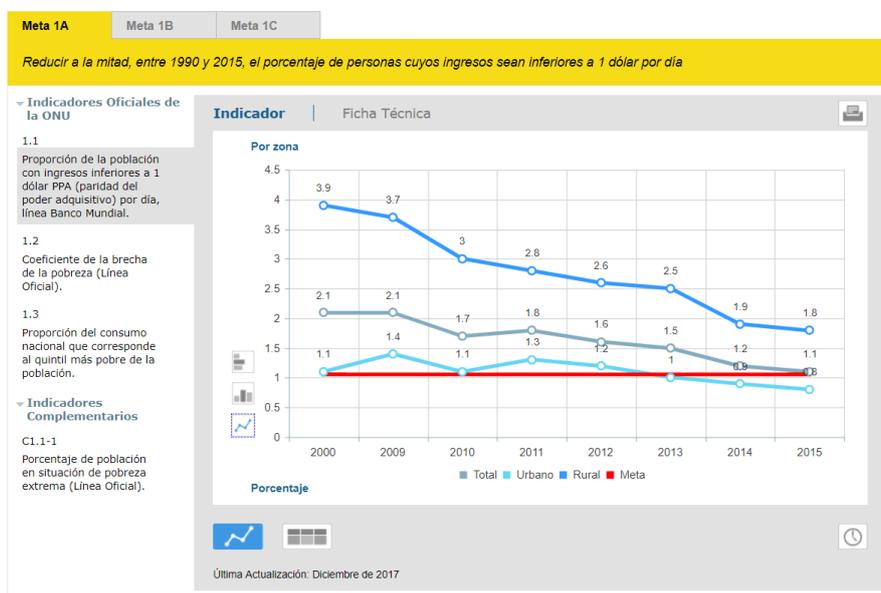
La dificultad del proceso ETL es causada por la naturaleza y procesos que se hacen sobre los datos fuente, pero debido a que no se espera que cambie drásticamente en el futuro el funcionamiento de la solución web respecto a los datos, la flexibilidad y escalabilidad fue aprobada por los miembros que hacían parte de la investigación.

#### 4. Discusión

Las técnicas computacionales usadas en este desarrollo se basaron principalmente en la guía para el desarrollo de bodegas de datos de Ralph Kimball (Kimball y Ross, 2013), lo cual garantiza en gran medida su correcto funcionamiento, al ser una metodología que ha sido ampliamente probada. Fue complicado encontrar desarrollos afines al nuestro, que especificaran las técnicas usadas para su construcción, pero se pueden comparar las características y funciones que cada uno ofrece respecto al producto final.

SimuDatSalud se puede comparar, en términos de la experiencia del usuario y presentación de la información, con la manera de representar los indicadores del monitoreo de los Objetivos del Milenio (ODM). Por ejemplo, la solución web desarrollada en República Dominicana con el fin de evaluar el progreso de los ODM permite visualizar diversos indicadores a través de 3 tipos gráficos estadísticos: líneas de tendencia, barras y columnas. Adicionalmente, se presenta cada indicador de manera tabular y con la respectiva ficha técnica del cálculo (ODM República Dominicana, 2013).

La información presentada en los ODM República Dominicana para cada indicador corresponde al periodo 1991-2015, aunque este rango de años puede cambiar de acuerdo con el indicador. De manera similar, las desagregaciones varían según el indicador, siendo estas muy reducidas, como se aprecia en la Figura 12. Se desconoce su arquitectura interna o principios de funcionamiento, pero al comparar ODM-RD con SimuDatSalud Risaralda, se aprecia una velocidad de respuesta al usuario muy similar, y se debe tener en cuenta que el segundo cuenta con muchos más indicadores y mucha más flexibilidad en las desagregaciones.



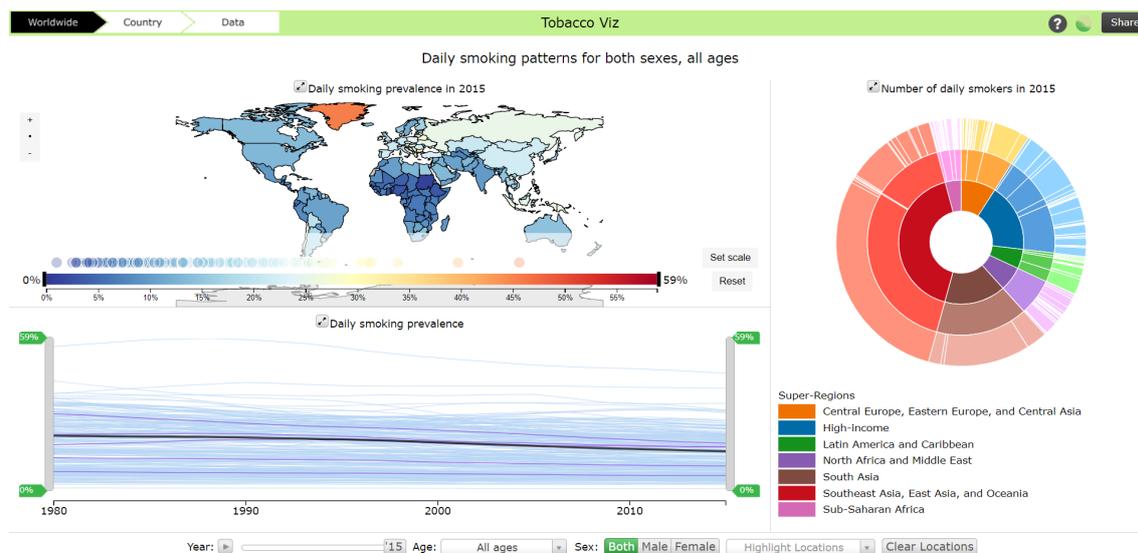
**Figura 12.** Indicador ODM en República Dominicana

**Fuente:** ODM República Dominicana (2013).

Desde la perspectiva del usuario final, los tiempos de respuesta al usuario final de ODM-RD están por debajo de los 3 segundos, de acuerdo con unas mediciones que se hicieron en la plataforma, los cuales son adecuados. SimuDatSalud tiene un tiempo de respuesta que no supera los 5 segundos, que para la funcionalidad que ofrece es más que respetable. Lo anterior es una muestra del correcto funcionamiento de la bodega de datos bajo la modificación de la metodología de Ralph Kimball, tanto en lo teórico como en una aplicación en la vida real (Kimball y Ross, 2013).

En términos de representación de la información, la solución web Health Metrics del Institute for the Health Metrics Evaluation de la Universidad de Washington (Institute for Health Metrics and Evaluation, 2019), se posiciona como referente más que acertado para contrastar con SimuDatSalud. Health Metrics presenta la información de la salud a través de indicadores representados por diferentes tipos de gráficos estadísticos, como se muestra en la Figura 13.

Este software presenta información a nivel mundial y específica de Estados Unidos. La información está disponible desde el año 1980 hasta la actualidad. Ambos desarrollos presentan muchas similitudes respecto a su funcionamiento y propósito general.



**Figura 13.** Ejemplo de visualización de indicador en Health Metrics  
**Fuente:** Institute for Health Metrics and Evaluation (2019).

La velocidad de las consultas en Health Metrics, desde la perspectiva del usuario final, es similar a las generadas en SimuDatSalud Risaralda. Ambos softwares están en un promedio de 5 segundos. Es importante resaltar que el apoyo financiero de Health Metrics es mucho mayor que el que recibió SimuDatSalud Risaralda; mientras el primero para su creación recibió US\$105.000.000 de la Fundación Bill y Mellinda Gates (Mankowski, 2007), el segundo recibió US\$5.384.128 (según la tasa de cambio del dólar en 2007) del fondo de ciencia, tecnología e innovación del Sistema General de Regalías; por tanto, se puede afirmar que la infraestructura tecnológica de Health Metrics es superior a la que se adquirió en SimuDatSalud Risaralda.

Una vez más, se puede apreciar que el funcionamiento de la bodega de datos y toda la arquitectura tecnológica que la rodea está a la altura de soluciones como Health Metrics. Un punto importante por mostrar es que las funcionalidades ofrecidas en ambas soluciones web son muy similares. Es superior Health Metrics, pero SimuDatSalud está en la capacidad de escalarse para generar aún más tipos de visualización, sin comprometer su rendimiento.

## 5. Conclusiones

El desarrollo de la solución web interactiva para análisis y visualización de los resultados de las simulaciones de SimuDatSalud Risaralda muestra de forma simple y comprensible los indicadores en salud de la población. La gran cantidad de información proveniente de las simulaciones se muestra a través de gráficos estadísticos simples (barras, áreas y tendencias) y avanzados (histogramas en dos y tres dimensiones) orientados a los distintos tipos de usuarios (alcaldes, gobernadores, estadísticos, funcionarios públicos del sector salud, representantes de las EPS e IPS, pacientes y ciudadanos en general).

Todo lo descrito anteriormente fue posible por la metodología de Ralph Kimball para la construcción de bodegas de datos (Kimball y Ross, 2013) con una breve modificación planteada a las dimensiones. Una de las más grandes ventajas del uso de esta metodología fue la posibilidad de poder ser implementada en un motor de base de datos relacionales tradicional, lo cual permitió ahorrar costos de implementación. Tal y como se mostró a lo largo de este escrito, esta metodología funciona tanto en lo teórico como en escenarios de la vida real. De nuestro aprendizaje quedaron los siguientes puntos importantes:

- Transformar la dimensión tiempo y ubicarla en el nombre de la tabla permite optimizar en gran manera el tiempo de ejecución de las consultas.
- Todos los filtros (dimensiones) de las tablas de hechos deben estar indexados, sin excepción alguna.
- Depende mucho del motor de base de datos que se use, pero es una ventaja cargar toda la bodega de datos a la memoria RAM o a la memoria caché de la base de datos.
- Las dimensiones de la bodega de datos, de acuerdo con la naturaleza del proyecto, deben ser lo más descriptivas posibles, porque estas serán las que se presentarán junto con los resultados al usuario.

Un aspecto importante por tratar es el uso de la herramienta R como lenguaje de programación para construir el ETL de la bodega de datos. Efectivamente, en la actualidad existen paquetes de software que hacen este proceso, pero, al tratarse de procesos estadísticos de un considerable nivel de complejidad, R fue la mejor opción por su propia naturaleza, además de adaptarse de manera adecuada a las necesidades de esta investigación. Algo negativo de nuestra implementación es que se hizo sobre R de escritorio y no en su versión de servidor, por lo cual fue complejo generar una interfaz amigable de administración del ETL de la bodega de datos.

Otro factor muy importante que fue tratado a lo largo de esta investigación fue la adecuada selección y manejo de los gráficos mediante los cuales se iba a presentar la información. El desafío siempre estuvo en la transformación de la enorme cantidad de datos de simulaciones a visualizaciones que cualquier tipo de usuario final implicado estuviera en la capacidad de entender y aprovechar. De nuestro aprendizaje quedaron los siguientes puntos importantes:

- Si se desea presentar el comportamiento de un fenómeno a lo largo de tiempo la mejor opción de visualización son los gráficos de líneas o de tendencias, por encima de gráficos de barras o columnas.
- Si se desea presentar la comparación entre dos variables sin mostrar trazabilidad en el tiempo, la mejor opción son los gráficos de barras o de columnas.
- A la hora de usar un gráfico de dispersión, se debe tener presente que para hacer el uso correcto de este se deben usar variables continuas (preferiblemente números con punto decimal).
- El uso de los histogramas para usuarios finales no expertos puede generar confusión porque tienen un enfoque puramente estadístico. Para amenizar estos gráficos, se sugiere anexar una descripción detallada del contenido.
- El tamaño del gráfico debe predominar en la interfaz que se le presenta al usuario debido a que es la información que realmente importa y no debe ser opacada a favor de la estética.

Para el departamento de Risaralda, la adecuada combinación de los tres factores mencionados anteriormente (simulación, manejo de datos y visualizaciones) significó el poder interactuar con la información generada en los procesos de microsimulación de una manera simple, didáctica y rápida, lo cual permite recrear escenarios futuros posibles y realizar experimentos sin arriesgar los recursos o la salud de la población. Es importante enfatizar que esta herramienta apoya la toma de decisiones al ofrecer información (económica, resultados en salud, etc.) sobre el posible efecto de la aplicación de políticas públicas en salud.

Fuera del contexto de esta investigación, la implementación de la metodología propuesta en este escrito es aplicable a cualquier otra investigación o proyecto que necesite presentar información agregada de manera rápida en soluciones web y la presentación de esta información no está limitada a lo presentado, esta se deberá ajustar al contexto.

## 6. Recomendaciones

El desarrollo de esta investigación generó mucho aprendizaje para aquellos que estuvieron implicados directamente en el desarrollo de la bodega de datos junto con el ETL escrito en R y en la solución web junto con todas sus visualizaciones. Los resultados finales fueron muy satisfactorios, pero hubo varios detalles que pueden ser mejorados en una futura implementación.

En cuanto al desarrollo del proceso ETL, que fue desarrollado sobre el lenguaje R, se recomienda reemplazar todos los ciclos *for* tradicionales por ciclos paralelos *foreach*, lo que significará un aumento en la velocidad de los procesos en general. La razón de no haberse implementado en esta investigación fue la limitación del tamaño de la memoria RAM del servidor. No obstante, e independientemente de la capacidad de la memoria, se recomienda hacer un seguimiento detallado a la apertura y cierre de los *clusters* que se usan en R para ejecutar las funciones paralelas; de otro modo, es posible causar un colapso del sistema.

La construcción de la bodega de datos se hizo sobre PostgreSQL 9.6, última versión estable disponible en el momento. Si se desea replicar la metodología usada en esta investigación, se recomienda usar una versión más actual de PostgreSQL y usar características como el paralelismo en las consultas presente en versiones superiores a la 10.0.

La implementación de la solución web fue hecha en CodeIgniter, una herramienta de trabajo para PHP bajo el patrón de diseño MVC que está en la capacidad de gestionar tanto *backend* como *frontend*. En su categoría, no es la única herramienta de trabajo que está en la capacidad de hacer esto; por ejemplo, Laravel tiene el mismo comportamiento. Sin embargo, se recomienda hacer una completa separación de ambas capas de la aplicación haciendo uso de herramientas de trabajo como Angular (Typescript), las cuales se enfocan solo en el *frontend*, permitiendo más orden y flexibilidad en el desarrollo. Para hacer la conexión con PHP (*backend*), se recomienda hacer uso de API REST.

Respecto a la biblioteca principal usada para la visualización (Google Charts), si se desea hacer una implementación compatible con dispositivos móviles, es preferible buscar otra biblioteca gráfica, debido a que el comportamiento de Google Charts no es el más apropiado, además de no ser automático, como sí lo es con otras bibliotecas diseñadas para este propósito. Una recomendación es hacer uso de bibliotecas como AmCharts o PlotlyJS.

## Declaración de conflicto de intereses

Los autores declaran no tener conflicto de intereses con respecto a la investigación, autoría y/o publicación de este artículo.

## Referencias

- Ahmed, I., y Smith, G. (2017). *PostgreSQL 9.6 High Performance*. Brimingham: Packt Publishing Ltd.
- Chen, C., Härdle, W., y Unwin, A. (2008). *Handbook of Data Visualization*. Berlin, Heidelberg: Springer Berlin Heidelberg. Recuperado de <https://doi.org/10.1007/978-3-540-33037-0>
- Gapminder. (2018). Tools. Recuperado en abril 14 de 2019 de [https://www.gapminder.org/tools/#\\$chart-type=bubbles](https://www.gapminder.org/tools/#$chart-type=bubbles)
- Gobernación de Risaralda. (2016). *Plan de desarrollo 2016-2019. Risaralda: verde y emprendedora*. Recuperado de <https://www.crcrisaralda.org/wp-content/uploads/2018/04/Plan-de-Desarrollo-Risaralda.-2016-2019.pdf>
- Google Developers. (2019). About Google chart tools. Recuperado en abril 14 de 2019 de <https://developers.google.com/chart/>
- Institute for Health Metrics and Evaluation. (2019). Data visualizations. Recuperado en abril 14 de 2019 de <http://www.healthdata.org/results/data-visualizations>
- International Organization for Standardization. (2018). ISO 9241-11:2018: Ergonomics of human-system interaction - Part 11: Usability: Definitions and concepts. Recuperado en abril 14 de 2019 de <https://www.iso.org/obp/ui/#iso:std:iso:9241:-11:ed-2:v1:en>
- Jacob, B., Ng, S. W., y Wang, D. T. (2008). *Memory Systems. cache, DRAM, Disk*. Burlington, New Jersey: Morgan Kaufmann Publishers.
- Kimball, R., y Ross, M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling*. Indianapolis, Indiana: John Wiley y Sons, Inc.
- Mankowski, T. (2007). Institute for Health Metrics and Evaluation Launched. Recuperado en abril 14 de 2019 de <http://www.washington.edu/news/2007/07/05/institute-for-health-metrics-and-evaluation-launched/>
- Ministerio de Salud y Protección Social. (2007). Encuesta Nacional de Salud Pública. Recuperado en abril 14 de 2019 de <https://www.minsalud.gov.co/salud/Paginas/EncuestaNacionaldeSaludPublica.aspx>
- Ministerio de Salud y Protección Social. (2015a). Encuesta Nacional de Situación Nutricional (ENSIN). Recuperado en abril 14 de 2019 de <https://www.minsalud.gov.co/salud/publica/epidemiologia/Paginas/encuesta-nacional-de-situacion-nutricional-ensin.aspx>
- Ministerio de Salud y Protección Social. (2015b). *Resumen Ejecutivo. Encuesta Nacional de Demografía y Salud*. Recuperado de <https://profamilia.org.co/wp-content/uploads/2019/06/Resumen-Ejecutivo-Encuesta-Nacional-De-Demografia-Y-Salud-ends-2015.pdf>

- ODM República Dominicana. (2013). Objetivos de Desarrollo del Milenio (ODM). Recuperado en abril 14 de 2019 de <http://odm.gob.do/>
- Oracle. (2019). Known Issues for JDK 8. Recuperado en abril 14 de 2019 de <https://www.oracle.com/technetwork/java/javase/8-known-issues-2157115.html>
- Plotly. (2019). Plotly JavaScript Open Source Graphing Library. Recuperado en abril 14 de 2019 de <https://plot.ly/javascript/>
- Python Software Foundation. (2019). About Python. Recuperado en abril 14 de 2019 de <https://www.python.org/about/>
- Rizzi, S. (2007). Conceptual Modeling Solutions for the Data Warehouse. In *Data Warehouses and OLAP* (pp. 1-26). IGI Global. Recuperado de <https://doi.org/10.4018/987-1-59904-364-7.ch001>
- Rosling, H., y Zhang, Z. (2011). Health advocacy with Gapminder animated statistics. *Journal of Epidemiology and Global Health*, 1(1), 11-14. Recuperado de <https://doi.org/10.1016/j.jegh.2011.07.001>
- Salutia Centro de Investigaciones en Salud. (2016). SimuDatSalud Risaralda. Recuperado en abril 14 de 2019 de <http://simudatsalud-risaralda.co/>
- The PHP Group. (2019). What is PHP? Recuperado en abril 14 de 2019 de <https://php.net/manual/en/intro-what-is.php>
- The PostgreSQL Global Development Group. (2019). F.27. pg\_prewarm. Recuperado en abril 14 de 2019 de <https://www.postgresql.org/docs/9.4/pgprewarm.html>
- The R Foundation. (2019). What is R? Recuperado en abril 14 de 2019 de <https://www.r-project.org/about.html>
- Unger, R., y Chandler, C. (2012). *A Project Guide to UX Design: For user experience designers in the field or in the making*. Berkeley: New Riders.
- Ware, C. (2013). *Information visualization: perception for design* (3<sup>rd</sup> ed.). US: Morgan.
- Williams, A. (2012). *C++ concurrency in action: practical multithreading*. Shelter Island, NY: Manning Publications.

## Sobre los autores

### Andrés Fabián Jiménez Torres

Tecnólogo en desarrollo de software de la Escuela Tecnológica Instituto Técnico Central; estudiante de Ingeniería de Sistemas en la Escuela Tecnológica Instituto Técnico Central; experto en programación y bases de datos e investigador en sistemas de información en la Fundación Salutia Centro de Investigaciones en Economía, Gestión y Tecnologías en Salud, Bogotá, Colombia.