Extracción de datos desde una plataforma de administración de equipos que brindan servicios de internet y su procesamiento para desarrollo de un tablero de indicadores

Data mining from a cloud management dashboard platform of internet wireless access point devices, and its processing for the development of a Key Performance Indicator (KPI) dashboard

DOI: 10.46932/sfjdv3n1-094

Received in: Jan 30st, 2021
Accepted in: Feb 1th, 2022

Cecilia Hurtado Valdez
MS, Economics
Independent Consultant
Arenque 2882, Loma Bonita Residencial, Zapopan, Jalisco, México
E-mail: ceci.hurtadov@gmail.com

Luis Alejandro León Dávila
MT, Learning Technologies
Centro Universitario de los Valles, Red Universitaria de Jalisco, Benemérita Universidad de Guadalajara
Carretera Guadalajara - Ameca Km. 45.5, C.P. 46600, Ameca, Jalisco, México
E-mail: luis.leon@valles.udg.mx

Jesús Alberto Moctezuma Gómez
Dipl, Computing Engineering
Independent Consultant
Castillo de Chapultepec 2883, El Fortín, 45066 Zapopan, Jalisco, México
E-mail: alberto.moctezuma91@gmail.com

Rubén Yáñez Reyna
MPA, Virtual Environments Management
Sistema de Universidad Virtual, Red Universitaria de Jalisco, Benemérita Universidad de Guadalajara
Calle Mezquitán No. 302, Col. Centro Barranquitas, C.P. 44100, Guadalajara, Jalisco, México
E-mail: ruben.yanez@academicos.udg.mx

RESUMEN

El siguiente trabajo expone las experiencias en cuanto al uso de herramientas y desarrollo de métodos para extraer, procesar y mostrar información sobre el uso del servicio de internet prestado por un programa federal en México. La fuente de datos utilizada es la plataforma de administración de equipos de conectividad "Dashboard Cisco-Meraki", por tratarse del fabricante que provee el equipamiento a los sitios de internet que conforman la muestra. Los datos extraídos son referentes a las conexiones, así como al tráfico recibido y enviado desde los dispositivos mediante los cuales los usuarios acceden al servicio de internet. Los datos se almacenaron de forma mensual, a partir de febrero de 2018 y hasta marzo de 2019. Se obtuvieron alrededor de 4 millones de registros de conexiones de usuario diariamente y 120 millones de registros mensuales. Se expondrá cómo se realiza la extracción, almacenamiento y

procesamiento de estos datos y cómo posteriormente se interactúa con otras bases de datos que contienen variables sociodemográficas que describen los sitios de internet analizados. Lo anterior con el objetivo de calcular algunos indicadores relacionados con los beneficios del programa federal, mismos que se presentan mediante tablero de control de inteligencia de negocios.

Palabras clave: Extracción de datos, cloud management dashboard, API, REST, big data, business intelligence.

ABSTRACT

The following work reports the experience regarding the use of certain tools and the development of methods to mining, processing and displaying relevant information on the use of free internet services delivered by a Federal Program in Mexico. The data source is the "Cisco-Meraki Dashboard", provided by the company that offers the wireless access point devices on the free internet sites contained in the analyzed sample. The obtained data refers to user connections, sent and received traffic registered by the usage of the internet service. The data was stored on a monthly basis, from February 2018 to March 2019. We obtained about 4 million registers daily and about 120 million monthly. The aim of this work is to explain the roadmap to mining, storing and processing the data, as well as the interaction of the clean data with other sociodemographic databases of the internet sites; all of this with the objective of calculating and displaying in a business intelligence dashboard, the KPI related to the benefits given by the federal program.

Keywords: Data mining, cloud management dashboard, API, REST, big data, business intelligence

1 CONTEXTO

Con el objetivo de cuantificar los beneficios de un programa federal1 que proporciona internet gratuito en sitios públicos, y para su visualización en un tablero de indicadores, se realizaron los convenios requeridos2 para acceder a los datos generados por las conexiones de usuarios de dichos sitios de internet del programa. Estos datos son almacenados en las diversas plataformas de administración de los fabricantes de los equipos que proporcionan la conectividad en los sitios de internet del programa federal.

Inicialmente se realizó un análisis de las restricciones con las que cuentan las plataformas de los diversos fabricantes para acceder a la información; en algunas de ellas, los datos no se concentran en una nube accesible vía web, sino que se guardan en un servidor, y aunque los datos si son accesibles vía API, se requieren licenciamientos que generan costos asociados. Otras plataformas requieren que los proveedores adquieran hardware para poder realizar las consultas de los datos.

Con base en lo anterior, y tomando en cuenta que para el desarrollo de un tablero de indicadores es relevante que los datos requeridos se encuentren siempre disponibles, se decidió componer la muestra con sitios que cuentan con equipamiento Cisco-Meraki, cuya plataforma de administración permite

[1] Los beneficios se cuantifican de forma parcial, ya que únicamente se utiliza una muestra de datos generados por usuarios del programa.

[2] La información utilizada es propiedad de la Secretaría de Comunicaciones y Transportes del Gobierno de México y se consulta y utiliza bajo convenio de confidencialidad.

obtener de manera ágil y estándar, información robusta para los análisis requeridos. El principal factor facilitador de esta plataforma, es que se encuentra disponible en la nube y se puede acceder vía interfaz web, con credenciales de acceso, o vía API mediante interfaces tipo REST hacia un URL en Internet, para esto solo se requiere contar con un API-Key, que es una cadena de texto que se obtiene desde la interfaz web para poder realizar solicitudes hacia la nube del fabricante Cisco-Meraki. Adicionalmente, los datos se actualizan en tiempo real desde los dispositivos que brindan la conectividad, ubicados en los sitios de internet del programa. Finalmente, la interfaz web permite asignar etiquetas a dichos dispositivos y grupos de dispositivos, lo que agrega elementos de identificación que resultan útiles para el procesamiento.

Se considera importante la facilidad de acceso a los datos a través de interfaces estándares con fundamento en el creciente uso y aplicación del concepto de "Redes definidas por software" (SDN, por sus siglas en inglés). La coordinación que se puede dar entre los dispositivos que forman parte de una configuración centralizada permite que dichos dispositivos puedan ajustar sus parámetros para no afectar el funcionamiento de otros dispositivos cercanos y es posible proveer información de los dispositivos que conforman la red en la que se encuentran.

El único aspecto limitante en el proceso de extracción de datos desde la fuente, es decir, desde la nube de Meraki, fue el hecho de que sólo es posible realizar 5 peticiones por segundo por cada organización de dispositivos. Posteriormente explicaremos con mayor detalle cómo se subsanó esta limitante.

Con la selección de esta plataforma, la muestra de sitios con la que se cuenta para el análisis oscila entre 14,318 y 23,847 sitios de internet mensuales, mismos que generan datos sobre conexiones de usuarios, alrededor de 4 millones de registros diarios y 120 millones de registros mensuales. Los datos recabados para este trabajo integran información a partir de febrero de 2018 y hasta marzo de 2019.

Específicamente, los datos consumidos desde la plataforma de administración Cisco-Meraki son:

Tabla 1. Variables obtenidas vía API de Meraki para cada conexión de dispositivos de usuario. Elaboración propia.

Variable	Significado
Sent	Tráfico enviado por el consumo del dispositivo de usuario
Recv	Tráfico recibido por consumo del dispositivo de usuario
Id	Identificador que tiene asignado el dispositivo por su usuario
description	Descripción del dispositivo de usuario, tomado del propio dispositivo
mdnsname	Nombre de dominio del dispositivo de usuario, tomado del propio dispositivo
dhcphostname	Identificador del dispositivo de usuario que se registra en el equipo que le asigna su dirección IP para establecer conexión
Mac	Identificador del dispositivo del usuario que está ligado a su interfaz de red
Ip	Dirección IP del dispositivo del usuario
vlan	Identificador del tráfico de red del que el dispositivo de usuario participa. Útil para identificar tráfico de PAI-I o PAI-E
switchport	Identificador del puerto del switch al que está conectado el dispositivo

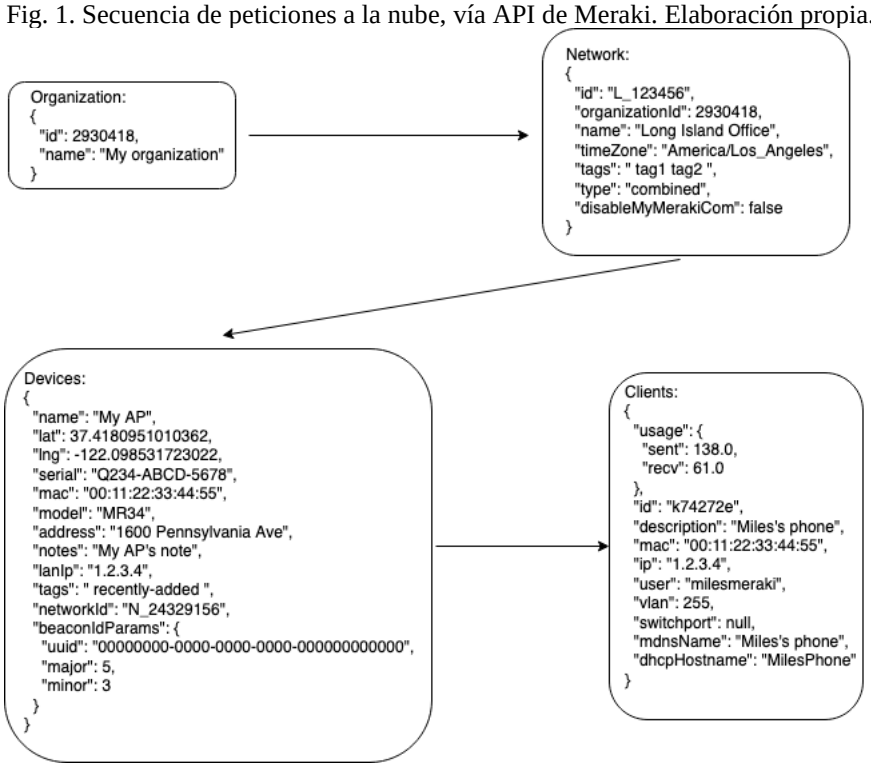
2 DESCRIPCIÓN DE LA SOLUCIÓN TECNOLÓGICA

La obtención de los datos descritos anteriormente se realiza a través de peticiones WEB al portal especificado por Cisco-Meraki, que concentra la información enviada por cada dispositivo que brinda la conectividad en los sitios respectivos. La invocación de URLs a este portal se realizó utilizando reglas específicas que consisten en el envío de parámetros que conforman el URL que se invoca. Las consultas se realizaron cada hora, de 9:00 a 21:00 horas, todos los días de la semana, para obtener la mayor cantidad de datos posibles durante los diferentes horarios de servicio de los sitios de internet del programa.

La estructura con la que se guardan los datos en la nube de Meraki es sencilla, se tiene un elemento inicial que se llama Organization. Un proveedor de conectividad puede tener una o varias "organizaciones" en las que gestiona sus redes y dispositivos. La Organization contiene "Networks", que son los espacios físicos en los que se instalan los dispositivos (Devices) que ofrecen la conectividad a los clientes ("Clients") o para nuestros propósitos, dispositivos de usuarios del programa.

Todos los días a las 23:00 horas se realizan consultas a la nube de Meraki para verificar las organizaciones, redes y dispositivos disponibles, mismos a los que se acotará la consulta del día siguiente. Es necesario realizar esta acción puesto que nuestra muestra es dinámica y se modifica constantemente con base en la vigencia y disponibilidad de los servicios de los sitios de internet del programa. La figura 1 muestra la secuencia de peticiones en los niveles antes descritos, hasta llegar a los datos requeridos sobre las conexiones de los dispositivos de usuarios del programa.

Fig. 1. Secuencia de peticiones a la nube, vía API de Meraki. Elaboración propia.



Las consultas se realizan cada hora con el fin de obtener los datos sobre las conexiones de la hora anterior, con lo que se registran datos en un periodo de tiempo diario desde las 8:00 a las 20:00 horas. Esto es así, para alcanzar a obtener los registros de conexiones y tráfico para el total dispositivos que deban ser consultados en ese día, recordando que solamente se pueden consultar 5 sitios por cada organización, por segundo. Por ello, se distribuyen el total de dispositivos a consultar a lo largo de una hora, y así sucesivamente durante 12 horas diariamente.

Se desarrollaron una serie de scripts en lenguaje Python que se ejecutan de forma automática y con capacidad en multiprocesamiento, para que se realice cada hora la consulta de la información generada por los dispositivos de usuario, esto implica un proceso distinto por cada organización, que ejecuta a su vez 5 hilos o procesos adicionales cada segundo de forma paralela.

Las peticiones que se pueden realizar a la nube de Meraki para obtener los datos necesarios para la visualización de los beneficios del programa de internet gratuito, en el tablero de indicadores desarrollado, son muy simples, enseguida se muestra un ejemplo de solicitud:

curl -L -H 'X-Cisco-Meraki-API-Key: <key>' -X GET -H 'Content-Type: application/json' 'https://api.meraki.com/api/v0/devices/[serial]/clients?timespan=86400'

La petición “http request” es de tipo GET:

GET /devices/[serial]/cliens

Los encabezados que se deben enviar en esta petición son:

X-Cisco-Meraki-API-Key {{X-Cisco-Meraki-API-Key}}

Ejemplo de respuesta:

Successful HTTP Status: 200

```
[
  {
    "description": "Hayg's Nexus 5",
    "mdnsName": "Hayg's Nexus 5",
    "dhcpHostname": "HaygNexus5",
    "usage": {"sent": 1337.0, "recv": 7331.0},
    "mac": "00:18:D3:AD:B3:3F",
    "ip": "1.2.3.4",
    "id": "lk12uq",
    "switchport": null
  },
  ...
]
```

El periodo de tiempo en el que se requiere obtener los datos de la consulta, considerado desde el instante en que se ejecuta la consulta hacia atrás, se debe especificar mediante el parámetro “timespan”,

en segundos, considerando 86,400 segundos para un día completo y con un máximo de 2,592,000 segundos de tiempo de consulta hacia atrás.

Se observa que lo que se obtiene de dicha consulta es un listado en formato JSON con la respuesta obtenida, en donde se muestran las conexiones realizadas por los dispositivos de usuarios del sitio donde se encuentra el dispositivo que brinda la conectividad.

Los datos obtenidos desde la plataforma de administración Cisco-Meraki, son complementados para su almacenamiento con las siguientes variables:

Tabla 2. Variables que se agregan a la base de datos. Elaboración propia.

Variable	Significado
Fecha	Día, mes, año del que se obtiene la información del dispositivo que brinda la conectividad
hora_inicio	Hora de inicio que cubre el periodo de poleo
hora_fin	Hora de fin que cubre el periodo de poleo
GID	Identificador global del sitio
NID	Identificador interno del sitio (no requerido)

Esta respuesta recibida en formato JSON, se transforma de manera que pueda ser almacenada en una base de datos para consultas, procesamiento y cruces de información que se explicarán a continuación.

3 ASPECTOS CRÍTICOS Y RELEVANTES

Para gestionar los datos obtenidos mediante los métodos descritos, resultó crítico contar con un sistema de almacenamiento de datos idóneo, que cuente con un esquema que responda a los requerimientos del proyecto, pero sobre todo, que se adapte y ajuste al tipo de datos con los que se opera. En el caso de los datos descritos en este trabajo, por sus características, podrían ser almacenados en un esquema tradicional, es decir, tablas relacionadas con sus correspondientes llaves, en una base de datos que cumpla con el estándar SQL, debido a que realmente no se están almacenando datos fuera de los tipos estándares, sino que incluso en una sola tabla sería posible guardar los datos obtenidos.

La peculiaridad de nuestro proyecto se trata de la gran cantidad de datos que se obtienen de las consultas puesto que guardamos información sobre cada usuario que se conecta a los sitios de internet del programa, en la muestra de sitios utilizada. Con esta consideración se buscó una opción con características que ofrecieran velocidad de escritura; se realizaron análisis comparativos entre las bases de datos NoSQL más utilizadas, entre ellas MongoDB y Cassandra.

Los casos de uso de la base de datos Cassandra permiten ver que se ha utilizado en diversos proyectos relacionados con datos recabados vía sensores y dispositivos IoT, mensajes y datos extraídos de redes sociales, detección de fraudes y datos en series de tiempo. Adicionalmente, encontramos que su

arquitectura es escalable y tolerante a fallas, su modelo de datos flexible y sus procesos de escritura y lectura son muy eficientes y permiten que aplicaciones críticas de Big Data estén disponibles de forma continua y se puede escalar a millones de transacciones por segundo. Otra característica de Cassandra que difiere de los esquemas relacionales, es que para el modelado de datos en Cassandra se deben considerar las solicitudes (queries) que el sistema hará a la base de datos y no existen agregaciones (joins) de datos. En este esquema eficiente de consultas complejas, debe considerarse también obtener respuestas de una única tabla; esto hace que sea común que se tengan varias tablas, con datos duplicados para dar respuestas a diferentes solicitudes.

La base de datos Cassandra se instaló y configuró con esquema de único nodo con un equipo con 90 GB de RAM y 20 procesadores, utilizando infraestructura de nube privada3.

El esquema más común de trabajo de la base de datos Cassandra contempla la configuración de un clúster de nodos en los que se puedan distribuir los procesos de almacenamiento de datos. El propio concepto de llave primaria que usa Cassandra en su modelo de datos permite ubicar cada registro en el nodo o nodos correspondientes para su consulta. En este proyecto se trabajó aprovechando los recursos disponibles en el servidor que se asignó, pero en otros casos, esta base de datos permite trabajar en un esquema de clúster para el crecimiento que se pueda requerir. La figura 2 muestra los recursos asignados al equipo en el que se ejecutó la base de datos Cassandra.

Fig. 2. Recursos asignados en el equipo utilizado para la base de datos Cassandra.

```
top - 16:05:42 up 215 days,  3:50,  1 user,  load average: 3.96, 3.45, 2.74
Tasks: 507 total,   1 running, 377 sleeping,   0 stopped, 129 zombie
%Cpu0  :  9.2 us,  3.9 sy,  0.0 ni, 83.9 id,  1.0 wa,  0.0 hi,  2.0 si,  0.0 st
%Cpu1  :  9.0 us,  2.7 sy,  0.0 ni, 87.3 id,  0.7 wa,  0.0 hi,  0.3 si,  0.0 st
%Cpu2  :  7.9 us,  3.0 sy,  0.0 ni, 87.5 id,  0.7 wa,  0.0 hi,  1.0 si,  0.0 st
%Cpu3  :  6.6 us,  6.3 sy,  0.0 ni, 85.7 id,  1.3 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu4  :  6.6 us,  3.6 sy,  0.0 ni, 89.4 id,  0.3 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu5  : 12.8 us,  2.0 sy,  0.0 ni, 84.9 id,  0.3 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu6  : 15.3 us,  5.0 sy,  0.0 ni, 78.4 id,  1.0 wa,  0.0 hi,  0.3 si,  0.0 st
%Cpu7  : 19.5 us,  4.4 sy,  0.0 ni, 75.8 id,  0.3 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu8  :  9.3 us,  2.6 sy,  0.0 ni, 87.4 id,  0.3 wa,  0.0 hi,  0.3 si,  0.0 st
%Cpu9  : 12.7 us,  1.3 sy,  0.0 ni, 85.7 id,  0.3 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu10 :  9.7 us,  0.7 sy,  0.0 ni, 89.3 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
%Cpu11 : 12.4 us,  2.7 sy,  0.0 ni, 84.3 id,  0.3 wa,  0.0 hi,  0.3 si,  0.0 st
%Cpu12 :  6.3 us,  3.3 sy,  0.0 ni, 89.7 id,  0.7 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu13 :  8.9 us,  3.0 sy,  0.0 ni, 87.4 id,  0.7 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu14 :  8.9 us,  2.3 sy,  0.0 ni, 88.2 id,  0.3 wa,  0.0 hi,  0.3 si,  0.0 st
%Cpu15 : 32.9 us,  6.0 sy,  0.0 ni, 60.8 id,  0.0 wa,  0.0 hi,  0.3 si,  0.0 st
%Cpu16 : 20.3 us,  3.0 sy,  0.0 ni, 76.7 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu17 :  6.7 us,  2.0 sy,  0.0 ni, 91.3 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu18 : 12.5 us,  2.0 sy,  0.0 ni, 85.1 id,  0.3 wa,  0.0 hi,  0.0 si,  0.0 st
%Cpu19 :  4.7 us,  7.0 sy,  0.0 ni, 87.3 id,  0.7 wa,  0.0 hi,  0.3 si,  0.0 st
KiB Mem : 93828760 total,  2664948 free, 39510772 used, 51653040 buff/cache
KiB Swap:  4194300 total,  3813716 free,   380584 used. 48327944 avail Mem
```

[3] La infraestructura de nube privada fue proporcionada por la Universidad de Guadalajara.

Los valores obtenidos del esquema establecido de único nodo para escritura, han sido de alrededor de 30 mil registros por segundo. Lo anterior aplicado a una base de 10 organizaciones de dispositivos Meraki, con alrededor de 50,000 dispositivos de conectividad4.

Otro factor relevante en el desarrollo del proyecto es la fase de procesamiento de los datos. Las decisiones relativas a las herramientas a utilizar para el procesamiento de los datos y la realización de cálculos para obtener indicadores, se basaron en la información que se decidió mostrar como producto final en el tablero de indicadores, es decir, total de conexiones de usuarios únicos de los sitios de internet del programa, tráfico enviado y recibido en cada sitio; información desagregada tal como dispositivos conectados por sitio, por día, entre otros, así como la generación de gráficos que permiten visualizar la información en relación a clasificaciones sociodemográficas, sitios rurales o urbanos, por mencionar un ejemplo; estos datos se muestran de manera mensual.

Al final de cada mes se realizó el procesamiento de los datos almacenados con el fin de obtener los indicadores mencionados. Para lograr la automatización de estos procesos se utiliza Apache Spark, herramienta que permite realizar consultas a la gran cantidad de registros que se encuentran en la base de datos Cassandra y que a su vez se requieren cruzar con los datos almacenados en otras bases de datos disponibles que proporcionan información sobre características sociodemográficas de los sitios de internet del programa5, la mayoría de estas bases de datos es de tipo relacional. Apache Spark cuenta con características que hacen que su uso e integración sea muy sencillo ya que utiliza un API unificado y resulta muy eficiente para combinar tareas de procesamiento; funciona como una especie de unificador de varios procesos sobre los mismos datos, en memoria. Los lenguajes de programación que se pueden utilizar con Spark son diversos: Java, Scala, Python. En este proyecto se seleccionó Python debido al conocimiento previo del mismo por parte del equipo, así como por la gran cantidad de bibliotecas de procesamiento de datos disponibles para lograr el desarrollo específico de los scripts requeridos para el procesamiento.

Con la interacción de Python con Spark, se desarrollaron varios programas para llevar a cabo dos principales tareas: realizar los cáculos correspondientes al conteo y discriminación de dispositivos únicos de usuarios que se conectan diariamente a los sitios de internet del programa; y ejecutar los cálculos de forma mensual para obtener los indicadores que se presentan en tablero.

El flujo de actividades que se realiza consiste en, como primer paso, descartar las conexiones a dispositivos de conectividad que no se encuentran debidamente identificados como sitios de internet del

[4] El número de organizaciones y dispositivos de nuestra muestra es dinámico se modifica con base en la vigencia y disponibilidad de los servicios.
[5] Las bases de datos utilizadas son propiedad de la Secretaría de Comunicaciones y Transportes del Gobierno de México y se consultan y utilizan bajo convenio de confidencialidad.

programa. Recordemos que anteriormente mencionamos que la plataforma de administración Cisco-Meraki permite identificar los dispositivos a través de etiquetas; cuando un proveedor no etiqueta adecuadamente sus dispositivos, no es posible realizar el cruce con las bases de datos con información sociodemográfica, por lo tanto esos registros no resultan de utilidad para el cálculo de indicadores. Una vez que se identifican los registros válidos, se procesan sus datos con Spark, a través de dataframes, agrupadores y funciones de agregación, y se generan dos tablas de resultados con la estructura requerida para su posterior tratamiento en la herramienta de business intelligence utilizada para la presentación del tablero de indicadores.

La primera tabla de resultados refiere entonces a los resultados por identificador de sitio de internet. Esta es la tabla principal utilizada para calcular la mayoría de indicadores que se presentan en el tablero. Como llaves se utilizan el año, mes, identificador, entidad o región, proveedor y sector, ya que estos son los filtros disponibles para consultas desde el tablero de indicadores. En esta tabla se calcula el número de conexiones, los datos enviados, datos recibidos, número de dispositios de usuarios del servicio de internet únicos; y se distribuye el tráfico mediante una variable con 5 categorías que permiten categorizar si los beneficios del servicio de internet se están dando hacia usuarios del PAI interno o hacia el PAI externo. A esta tabla se anexan además datos de las bases de datos sociodemográficos. Como resultados se obtiene una estructura que se muestra en la figura 3.

Fig. 3. Estructura de la tabla principal de los sitios con su identificador y sus parámetros descriptores obtenidos de otras bases de datos. Elaboración propia.

```
                    Table "public.grouped_results_by_gid"
        Column          |       Type        | Collation | Nullable | Default
------------------------+-------------------+-----------+----------+---------
 year                   | bigint            |           | not null |
 month                  | bigint            |           | not null |
 gid                    | character varying |           | not null |
 estado                 | character varying |           |          |
 proveedor              | character varying |           |          |
 sector                 | character varying |           |          |
 tam_localidad          | numeric           |           |          |
 marginacion            | character varying |           |          |
 tipo_poblacion         | character varying |           |          |
 edomunloc              | bigint            |           |          |
 latitud                | double precision  |           |          |
 longitud               | double precision  |           |          |
 conexiones             | bigint            |           |          |
 datos_enviados         | bigint            |           |          |
 datos_recibidos        | bigint            |           |          |
 dispositivos           | bigint            |           |          |
 ancho_banda_bajada     | double precision  |           |          |
 ancho_banda_subida     | double precision  |           |          |
 tipo_red               | character varying |           |          |
 tipo_espacial          | character varying |           |          |
 beneficio_pai_externo  | numeric           |           |          |
 beneficio_pai_interno  | numeric           |           |          |
Indexes:
    "grouped_results_by_gid_pkey" PRIMARY KEY, btree (year, month, gid)
    "idx_grouped_results_by_gid_year_month" btree (year, month)
    "idx_grouped_results_by_gid_year_month_estado" btree (year, month, estado)
    "idx_grouped_results_by_gid_year_month_estado_proveedor" btree (year, month, estado, proveedor)
    "idx_grouped_results_by_gid_year_month_estado_proveedor_sector" btree (year, month, estado, proveedor, sector)
    "idx_grouped_results_by_gid_year_month_estado_sector" btree (year, month, estado, sector)
    "idx_grouped_results_by_gid_year_month_proveedor" btree (year, month, proveedor)
    "idx_grouped_results_by_gid_year_month_proveedor_sector" btree (year, month, proveedor, sector)
    "idx_grouped_results_by_gid_year_month_sector" btree (year, month, sector)
```

Finalmente se ejecutan los cálculos correspondientes para obtener los beneficios del programa6.

La segunda tabla agrupa las conexiones por año, mes, entidad o región, sector, proveedor. Contando los dispositivos de usuarios únicos del servicio de internet, el tráfico enviado y recibido, se genera la estructura que se muestra en la figura 4.

Fig. 4. Estructura de la tabla que integra los datos de las posibles combinaciones de filtros aplicables para mostrar resultados de usuarios y datos enviados y recibidos en un mes. Elaboración propia.

```
                    Table "public.users_by_month"
     Column      |       Type        | Collation | Nullable | Default
-----------------+-------------------+-----------+----------+---------
 year            | bigint            |           | not null |
 month           | bigint            |           | not null |
 estado          | character varying |           | not null |
 sector          | character varying |           | not null |
 proveedor       | character varying |           | not null |
 usuarios        | bigint            |           |          |
 datos_enviados  | bigint            |           |          |
 datos_recibidos | bigint            |           |          |
Indexes:
    "users_by_month_pkey" PRIMARY KEY, btree (year, month, estado, sector, proveedor)
    "idx_users_by_month_year_month_estado" btree (year, month, estado)
    "idx_users_by_month_year_month_estado_proveedor" btree (year, month, estado, proveedor)
    "idx_users_by_month_year_month_estado_proveedor_sector" btree (year, month, estado, proveedor, sector)
    "idx_users_by_month_year_month_estado_sector" btree (year, month, estado, sector)
    "idx_users_by_month_year_month_proveedor" btree (year, month, proveedor)
    "idx_users_by_month_year_month_proveedor_sector" btree (year, month, proveedor, sector)
    "idx_users_by_month_year_month_sector" btree (year, month, sector)
```

4 RESULTADOS OBTENIDOS

Derivado de los procesos descritos en las secciones anteriores, se obtiene como resultado la visualización de la información procesada a través de un tablero de indicadores útiles para la toma de decisiones. En este caso se optó por utilizar la herramienta de business intelligence Pentaho, a través de su módulo de Dashboard Designer. En esta interfaz se desarrollaron las vistas necesarias para mostrar los indicadores y gráficos correspondientes. El desarrollo del tablero se apoyó con algunos elementos como Java (Backend), Javasript y Jquery, así como Leaflet para mostrar algunos datos en formato de mapas de calor. El tablero cuenta con la capacidad de configuración de vistas y despliegue de datos diferenciados por rol de consulta. Las figuras 57 y 6 muestran las vistas de algunos indicadores del tablero y de los mapas de calor de tráfico generados, se observan también los filtros que se pueden aplicar a las consultas.

[6] Para realizar estos cálculos se programaron las fórmulas definidas para calcular los elementos de medición de beneficios del programa, que se derivan de los modelos realizados para este fin y que no son objeto de este trabajo.

[7] Para resguardar la confidencialidad de la información, la figura muestra resultados con valor cero, con el objetivo único de ejemplificar la vista del tablero.

Fig. 5. Ejemplo de indicadores obtenidos en un mes a partir del procesamiento previo de los datos integrados. Elaboración propia.

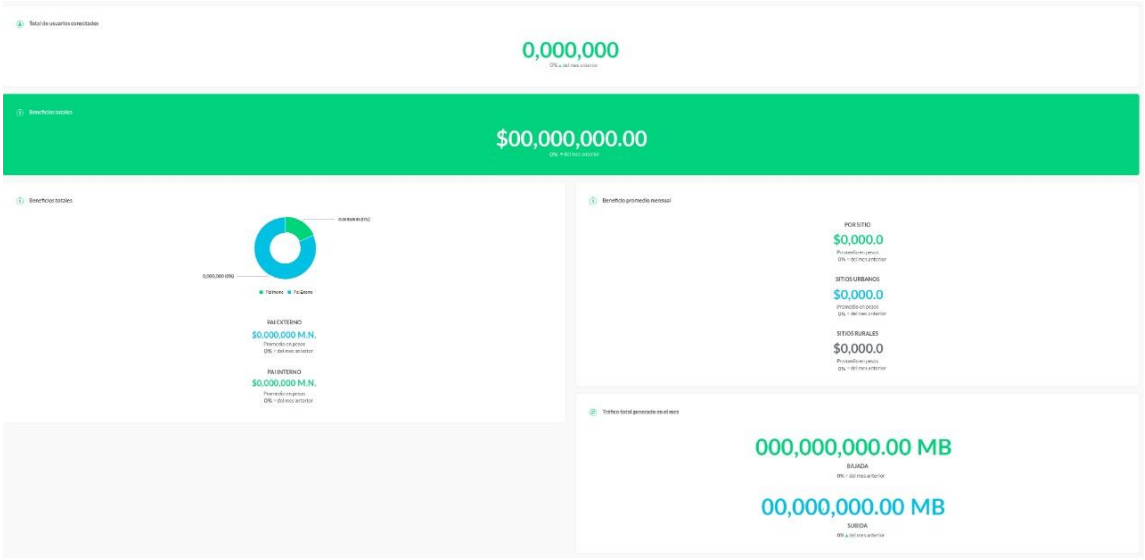
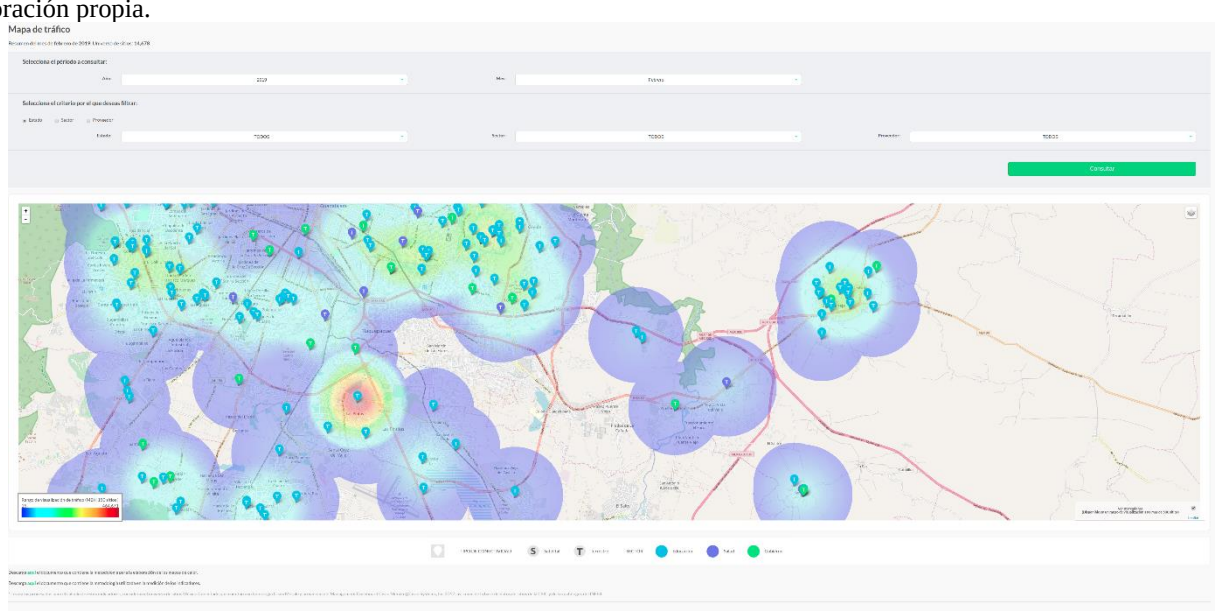


Fig. 6. Representación de la distribución de tráfico por sitio en mapas de calor, a partir del procesamiento de los datos. Elaboración propia.



La ruta de desarrollo de este producto establece un modelo de negocios factible para recabar, analizar, procesar y presentar datos que pueden ser útiles para el seguimiento, toma de decisiones y presentación de resultados, de diversos programas públicos o privados.

5 CONCLUSIONES

El proceso de exploración y análisis de las diversas herramientas disponibles para realizar este tipo de proyectos, así como el establecimiento de los flujos de trabajo y modelos de negocios requeridos durante su desarrollo, deja un precedente documentado para replicar proyectos de esta índole, cuando

buscamos extraer una gran cantidad de datos desde alguna fuente en la nube y requerimos un procesamiento eficiente, expedito y con la capacidad de generar información disponible en todo momento.

Aprendemos además que cualquier herramienta utilizada tiene el potencial de ser flexible y adaptarse a las necesidades específicas de un proyecto en la medida de la creatividad de los participantes del equipo involucrados. Hacemos énfasis en que todos los componentes utilizados son de software libre, lo que elimina costos de licenciamiento y aumenta la posibilidad de integración de componentes y de expandir funcionalidades según se requiera.

Adicionalmente, este proyecto nos deja la experiencia de conformar equipos multidisciplinarios, con diversos perfiles, tales como, administradores de bases de datos, ingenieros de datos, gestores de producto, para concretar una óptima ejecución del proyecto.

El principal impacto no esperado de este proyecto está relacionado con el tiempo de desarrollo del mismo. Las mayores desviaciones de tiempo se debieron a que las primeras extracciones de datos experimentadas, se realizaron sin el conocimiento suficiente de la situación de los datos y de las capacidades de la plataforma de administración fuente, para ordenar y categorizar los datos de forma que los registros que almacenamos sean lo suficientemente válidos para que los procesos posteriores de análisis y presentación resulten eficaces. Aún después de conocer y definir los flujos y reglas para ordenar y categorizar de mejor manera la información, a fin de realizar extracciones de datos útiles, esta tarea de gestión corrió, en este caso, por parte de los proveedores de los servicios.

REFERENCIAS

Cisco Systems, Inc. The Cisco Meraki Dashboard API - Cisco Meraki. Recuperado de https://documentation.meraki.com/zGeneral_Administration/Other_Topics/The_Cisco_Meraki_Dashboard_API (2018).
Cisco DevNet. Meraki Dashboard API - Cisco DevNet. Recuperado de https://developer.cisco.com/meraki/api/#/rest/getting-started (2018).

Hart-Davis, G. Implementación de iPads en el aula: Planificación, instalación y administración de iPads en escuelas y universidades (Deploying iPads in the Classroom: Planning, Installing, and Managing iPads in Schools and Colleges9). Barnard Castle, Durham, United Kingdom: Apres (2017).

Armbrust, et al. Spark SQL: Procesamiento de datos relacionales en Spark (Spark SQL: Relational Data Processing in Spark). Recuperado de https://amplab.cs.berkeley.edu/wp-content/uploads/2015/03/SparkSQLSigmod2015.pdf (2015).

Chacko. Basheer. Madhu Kumar. Capturando la procedencia de Big Data Analytics utilizando la interfaz SQL (Capturing Provenance for Big Data Analytics done Using SQL Interface). Allahabad, India: IEEE (2015).

Ashish Patro, Suman Banerjee. COAP: un enfoque definido por software para Home WLAN Management de un API abierta (COAP: A Software-Defined Approach for Home WLAN Management through an Open API). Recuperado de http://pages.cs.wisc.edu/~patro/papers/coap_mobiarch2014.pdf (2014).

Manoj R, Patil. Feris, Thia. Pentaho para análisis de Big Data (Pentaho for Big Data Analytics). Birmingham, UK: Packt Publishing Ltd. (2013).

Rhea, Sean. Wang, Eric. Wong, Edmund. Atkins, Ethan. Storer, Nat. LittleTable: una base de datos de series temporales y sus usos (LittleTable: A Time- Series Database and Its Uses). Recuperado de https://meraki.cisco.com/lib/pdf/trust/lt-paper.pdf (2017).

Zaharia, et al. Apache Spark: un motor unificado para el procesamiento de big data (Apache Spark: a unified engine for big data processing). New York, NY, USA: ACM DL (2016).

GLOSARIO

API	Interfaz de programación de aplicaciones, se refiere a rutinas que proveen acceso a funciones de un determinado software.
API-Key	Cadena de texto generada desde la interfaz web del dashboard Meraki, que se usa para identificar a un usuario de las consultas vía API. Se liga a una cuenta de usuario y se utiliza en el proceso de autenticación.
Backend	Es la parte que se conecta con la base de datos y el servidor que utiliza dicho sitio web.
Business intelligence	Inteligencia empresarial, inteligencia de negocios o BI. Conjunto de estrategias y herramientas enfocadas al análisis de datos.
Cisco-Meraki	Empresa de TI administrada en la nube.
Dataframes	Hoja de datos o marco de datos.
Fabricante	Nombre de la empresa o marca tecnológica que crea los equipos o soluciones que brindan la conectividad, a los que se hace referencia.
GB	Gigabyte, unidad de medida de capacidad equivalente a 1000 bytes.
IoT	Internet de las cosas (en inglés, Internet of Things).
JSON	JavaScript Object Notation, (notación de objeto de JavaScript) es un formato de texto para el intercambio de datos.
KPI	Key Performance Indicator.
Lenguaje python	Lenguaje de programación orientado a objetos.
NoSQL	Modelo de gestión de consultas en el que no solo se utiliza el lenguaje SQL.
Organización	Nomenclatura utilizada por la plataforma de administración Cisco-Meraki que se utiliza para describir el conjunto de redes que forman parte de una sola entidad organizativa, tales como una empresa o distrito escolar.
PAI externo	Servicio de conectividad que brinda el dispositivo a la comunidad.
PAI interno	Servicio de conectividad que brinda el dispositivo a la institución.
Proveedores	Empresas responsables de proveer el servicio de conectividad a Internet en los sitios del programa federal analizado.
RAM	Random Access Memory, elemento de hardware para capacidad de un sistema.
REST	La transferencia de estado representacional (en inglés representational state transfer) o REST, describe cualquier interfaz que utilice el protocolo HTTP para obtener datos o ejecutar procesos sobre datos, en formatos como XML, JSON, etc.
Scripts	Conjunto de comandos de algún lenguaje de programación que realizan una tarea en específico.
SQL	Lenguaje específico del dominio utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales.
TICs	Tecnologías de la información y la comunicación. Todos aquellos recursos, herramientas y programas que se utilizan para procesar, administrar y compartir la información mediante diversos soportes tecnológico.
URL	Uniform Resource Locator (Localizador Uniforme de Recursos).
WEB	Sistema de gestión de información más utilizado para la trasmisión de datos a través de internet.