ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

DISEÑO DE SISTEMA DATA WAREHOUSE SOBRE UNA ARQUITECTURA EN LA NUBE APLICADO A CASO DE ESTUDIO

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

MEJÍA QUISHPE JHONN PATRICIO
Jhonn.mejia@epn.edu.ec

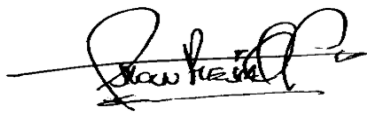
DIRECTOR: Gabriela Lorena Suntaxi Oña
gabriela.suntaxi@epn.educ.ec

Quito, julio 2022

DECLARACIÓN

Yo Mejía Quishpe Jhonn Patricio declaró bajo juramento que el trabajo aquí descrito es de mi autoría; y, que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Mejía Quishpe Jhonn Patricio

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Mejía Quishpe Jhonn Patricio, bajo nuestra supervisión.

PhD. Gabriela Suntaxi.

DIRECTORA DE PROYECTO

AGRADECIMIENTO

Agradezco a mis padres Wilman Enrique Mejía Robalino y Carmen Yolanda Quishpe Minaya por el apoyo, empuje, enseñanzas y amor incondicional.

Agradezco a mis hermanos Enrique, Stalyn, Sarai y Sebastian Mejia Quishpe por su amistad, consejos, amor y lealtad infinita.

Agradezco a mi tutora Gabriela Suntaxi por el voto de confianza, el tiempo y ayuda en todo este proceso de realización de mi tesis.

Agradezco a cada uno de mis amigos a lo largo de mi carrera en la facultad, en especial a Mauricio por ser una persona leal e incondicional.

Agradezco a todas y cada una de las personas que han llegado y pasado por mi vida, porque de cada uno de ellos he aprendido y me he llevado lo mejor que cada uno me pudo ofrecer.

DEDICATORIA

Dedico este trabajo a mis padres Wilman Y Yolanda, a mis hermanos Enrique, Stalyn, Sarai y Sebastian, a mis tíos Laura Magdalena, Laura Enma, Normita, Silvia, Willian, Carlitos, y Edison, a mis abuelitos Mami Pera, Papi Gilo, Mami Eva y Papi Segundo.

Por vos y para vos Damián Ezequiel.06:22

XXVIII-V

TABLA DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

El presente proyecto de titulación tiene por objetivo diseñar un sistema Cloud Data Warehouse sobre una arquitectura en la Nube para una mediana empresa comercial. Este proyecto se efectúa con el propósito de agilizar los procesos de manejo de información, control, monitoreo y actualización de los datos de las sucursales que dispone la empresa Bikes Extreme Ecuador. Asimismo, el presento proyecto servirá para una adecuada lectura de los datos generados por las ventas, gestión de proveedores y control de clientes para la toma de decisiones por parte de la directiva de la empresa.

En relación al método aplicado se planteó, en primer lugar, para la adecuación, manejo y transformación de los datos la metodología CRISP-DM. CRISP-DM establece una serie de pasos para configurar los datos que se van a trabajar. En segundo lugar, se usó la metodología MSF para la implementación de la infraestructura necesaria en la Nube para la construcción del Cloud Data Warehouse. Además, se utilizó una herramienta de visualización de datos para la presentación de resultados de forma fácil y adecuada.

Finalmente, se puede concluir que a través de la presente investigación y con la implementación apropiada de este tipo de infraestructura se logró un control y análisis adecuado de la información. Adicionalmente, se consiguió una acertada forma de presentación de la información por medio de la herramienta de visualización de los datos. Esta herramienta contribuyó de forma eficiente a una correcta interpretación de los datos que son proporcionados por la infraestructura y una acertada toma de decisiones por parte de la directiva de la empresa antes mencionada.

Palabras claves: Cloud Data Warehouse, Arquitectura en la Nube, CRISP-DM, MSF.

ABSTRACT

This degree project aims to design a Cloud Data Warehouse system on a Cloud architecture for a medium-sized commercial company. This project is carried out to streamline the processes of information management, control, monitoring, and updating of the data of the branches that the company Bikes Extreme Ecuador has. Likewise, the present project will serve as an adequate reading of the data generated by sales, supplier management, and customer control for decision-making by the company's management.

Concerning the method applied, the CRISP-DM methodology was first proposed for the data's adequacy, management, and transformation. CRISP-DM establishes a series of steps to configure the data to be worked on. Second, the MSF methodology was used to implement the necessary infrastructure in Cloud the Cloud for the construction of the Cloud Data Warehouse. In addition, a data visualization tool was used to present results easily and adequately.

Finally, it can be concluded that adequate control and analysis of the information were achieved through the present investigation and with the appropriate implementation of this type of infrastructure. Additionally, the data visualization tool achieved a correct way of presenting the information. Therefore, this tool efficiently contributed to a correct interpretation of the data provided by the infrastructure and correct decision-making by the company's directive mentioned above.

Keywords: Cloud Data Warehouse, Cloud architecture, CRISP-DM, MSF.

1. INTRODUCCIÓN

1.1. Planteamiento del problema

El *Cloud Data Warehouse* (CDW) es definido como un conjunto de datos orientados a un dominio integrado y no volátil, que no varía en el tiempo y que ayuda a la toma de decisiones en las organizaciones (Efendi y Krisanty, 2020). Es por ello, que en la actualidad algunas empresas están enfrentando desafíos con respecto al manejo de sus operaciones de negocio a través de la Tecnología de la Información (TI), logrando una mayor reducción de costos y aumentando sus ingresos. Por lo que, las organizaciones han implementado proyectos de CDW para mejorar la capacidad de medir, entender y analizar las operaciones de negocio.

Las actividades diarias de procesamiento de datos crean cantidades masivas de datos, por los que el *Cloud Computing* (computación en la Nube) se ha convertido en un nuevo paradigma para el alojamiento de dichos datos y la prestación de servicios a través de internet. La computación en la Nube es atractiva para los propietarios de negocios, ya que elimina el requisito de que los usuarios planifiquen con anticipación el aprovisionamiento y permite que las empresas comiencen con los recursos fundamentales y los escalen sólo cuando hay un aumento en la demanda de servicios (Efendi y Krisanty, 2020). Por lo tanto, la computación en la Nube es un modelo para permitir un acceso de red conveniente y bajo demanda a un grupo compartido de recursos informáticos configurables como: redes, servidores, almacenamiento, aplicaciones y servicios que se pueden aprovisionar y liberar rápidamente con un mínimo esfuerzo de administración o interacción del proveedor de servicios.

Los CDW están atravesando actualmente dos transformaciones muy importantes que tienen el potencial de impulsar niveles significativos de innovación empresarial. La primera área de transformación es el impulso para aumentar la agilidad general y la gran mayoría de los departamentos de TI están experimentando un rápido aumento de la demanda de datos. Los directivos quieren tener acceso a más datos históricos, mientras que, al mismo tiempo, los científicos de datos y los analistas de negocios están explorando formas de introducir nuevos flujos de datos en el almacén para enriquecer el análisis existente, así como impulsar nuevas áreas de análisis (Assiroj, 2021).

En este sentido, la rápida expansión de los volúmenes y fuentes de datos significa que los equipos de TI necesitan invertir más tiempo y esfuerzo, asegurando que el rendimiento de

las consultas permanezca constante y necesitan proporcionar cada vez más entornos para equipos individuales para validar el valor comercial de los nuevos conjuntos de datos. Con respecto a la segunda área de transformación, esta gira en torno a la necesidad de mejorar el control de costos. Existe una creciente necesidad de hacer más, con cada vez menos recursos, al mismo tiempo que se garantiza que todos los datos sensibles y estratégicos estén completamente asegurados, a lo largo de todo el ciclo de vida, de la manera más rentable (Assiroj, 2021).

La computación en la Nube ha atraído mucha atención en los últimos tiempos. Los medios de comunicación, así como los analistas, son en general muy positivos sobre las oportunidades que la Nube ofrece. Al respecto, según un comunicado de prensa de Gartner indicó que, la Nube será la pieza central de las nuevas experiencias digitales. La actitud positiva hacia la importancia y la influencia de la computación en la Nube dio como resultados pronósticos optimistas del mercado relacionados con la Nube e indicaron que los ingresos globales de la Nube totalizarán $ 474 mil millones en 2022, frente a $ 408 mil millones en 2021 (Gartner, 2021).

La computación en la Nube presenta un nuevo modelo para la prestación de servicios de TI y, por lo general, implica acceso de autoservicio bajo demanda a través de una red, que es dinámicamente escalable y elástica, utilizando grupos de recursos a menudo virtualizados. A través de estas características, la computación en la Nube tiene el potencial de mejorar la forma en que operan las empresas y la TI al ofrecer un inicio rápido, flexibilidad, escalabilidad y rentabilidad (Díaz y Matta, 2020).

Por lo anteriormente expuesto, el objetivo principal de este proyecto es implementar un sistema de CDW sobre una arquitectura en la Nube, para poder almacenar datos consolidados de diversas fuentes y con ellos proporcionar soporte a la toma de decisiones en una mediana empresa comercial. Otro de los objetivos del proyecto es aprovechar todas las ventajas en torno a la implementación de la infraestructura en Nube tales como escalabilidad, seguridad, precios y elasticidad.

1.2. Objetivo general

Diseñar un sistema *Cloud Data Warehouse* sobre una arquitectura en la Nube para una mediana empresa comercial.

1.3. Objetivos específicos

- Realizar un estudio del trabajo relacionados en el área de *Cloud Data Warehouse* y *Cloud Computing*.
- Diseñar la arquitectura de servidores en la Nube para la implementación de un *Data Warehouse.*
- Diseñar la infraestructura necesaria para *Cloud Data Warehouse* en *la Nube.*
- Implementar el sistema de *Cloud Data Warehouse*.
- Implementar un *Cloud dashboard* para visualización de información de datos.
- Probar el funcionamiento de la solución integral.

1.4. Alcance

El presente proyecto estará focalizado en buscar la implementación de un sistema de Data Warehouse en un ambiente en la Nube, para poder almacenar datos consolidados de variadas fuentes y con estas proporcionar soporte sobre la toma de decisiones de una mediana empresa comercial, además de contar con un ambiente fácil de escalar en recursos, elástico y seguro. La propuesta se concentra en la estructura de datos de la empresa Bikes Extreme Ecuador (BEE) de la categoría PYME (Pequeña y Mediana Empresa) considerando la metodología *Cross-Industry Standard Process for Data Mining* o mejor conocida como CRISP-DM. A partir de esta metodología se proyecta la adecuación, minería y análisis de la data, considerando una fase de comprensión del negocio y de los datos asociados a este, la realización del modelado de los datos y la evaluación de estos como base para la implementación de la Inteligencia empresarial o business intelligence.

El estudio contempla el desarrollo de una estructura soportada en la Nube, a partir de la cual, con el apoyo de un *Extract, transform and load* (ETL), se gestionan los datos para establecer un CDW que alimentará un dashboard, con el empleo de la herramienta Power BI de la compañía Microsoft, para ofrecer visualizaciones de la data de manera interactiva y ajustada a las necesidades de la empresa BEE.

1.5. Justificación de la investigación

1.5.1. Justificación teórica

El presente estudio centra su justificación en diferentes contextos, desde lo teórico y tomando en cuenta la definición de Efendi y Krisanty (2020) un *Data Warehouse* es un sistema

informático para archivar y analizar datos históricos de una organización, donde la empresa BEE copia y almacena información de sus sistemas en una base de datos. En tal sentido, la ejecución de este trabajo contribuirá con las derivaciones o resultados que emerjan al estado del arte vinculado al establecimiento de esquemas de inteligencia de negocio y manejo de la data que considera la tecnología de la Nube.

1.5.2. Justificación práctica

Desde lo práctico, con base en el objetivo de diseñar un sistema de almacenamiento de datos para que los líderes de las PYMES que, por su poca experiencia y progresiva consolidación de sus operaciones, no disponen de una forma correcta de procesar los datos. Dichas empresas tampoco saben usar la data como un recurso estratégico que les ayude al momento de la toma de decisiones y estas puedan ser analizadas e implementadas de una manera más rápida y precisa.

Asimismo, los líderes de la empresa BEE tienen mayor necesidad de acceder de forma fácil y rápida a la información del negocio de una manera estructurada y de calidad, para que puedan emplear esta información en la toma de decisiones. La inquietud de implementar un CDW surge de la necesidad de dar respuesta a estos requerimientos y va a permitir reunir y estructurar la información dispersa por los distintos sistemas informáticos de la empresa. Asimismo, el CDW servirá como base para la construcción de otros sistemas de inteligencia de negocio.

Con la implementación de un CDW en una organización, se espera tener una mejora en la facilidad de acceso hacia la información, una mayor flexibilidad y rapidez de respuesta frente a la toma de decisiones. Otro de los aspectos importantes que se desea mejorar con la implementación de un CDW es que mejora la comunicación entre departamentos y personas. Pero no solo se desea obtener beneficios entorno al manejo de la información, sino también en relación a la optimización de la infraestructura para obtener beneficios en costos, seguridad y crecimiento.

1.5.3. Justificación metodológica

La metodología y fases a ser implementada se justifica por cuanto están estructuradas de manera sistemática y coordinada con el fin de alcanzar los objetivos que se delinearon para el estudio. Para la creación del CDW se utilizará la metodología CRISP-DM (*Cross Industry Standard Process for Data Mining*) que es una de las principales metodologías usadas por los analistas en la inteligencia de negocios y que proporciona una descripción

normalizada del ciclo de vida de un proyecto estándar de análisis de datos, de forma análoga a como se hace en la ingeniería del software con los modelos de ciclo de vida de desarrollo de software. Este modelo cubre las fases de un proyecto, sus tareas respectivas, y las relaciones entre estas (Mejías, 2018).

La metodología CRISP-DM contempla el proceso de análisis de datos como un proyecto profesional, estableciendo así un contexto mucho más rico que influye en la elaboración de los modelos. Este contexto tiene en cuenta la existencia de un cliente que no es parte del equipo de desarrollo, así como el hecho de que una vez hallado el modelo adecuado se requerirá de un despliegue y un mantenimiento, y es preciso documentarlo de forma exhaustiva para que otros equipos de desarrollo utilicen el conocimiento adquirido y trabajen a partir del cliente (Mejías, 2018). Adicionalmente, esta metodología será detallada en el capítulo 3 y está representada por seis fases las cuales son: Comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue. Seguidamente en la Figura 1 se presenta las fases del CRISP-DM.

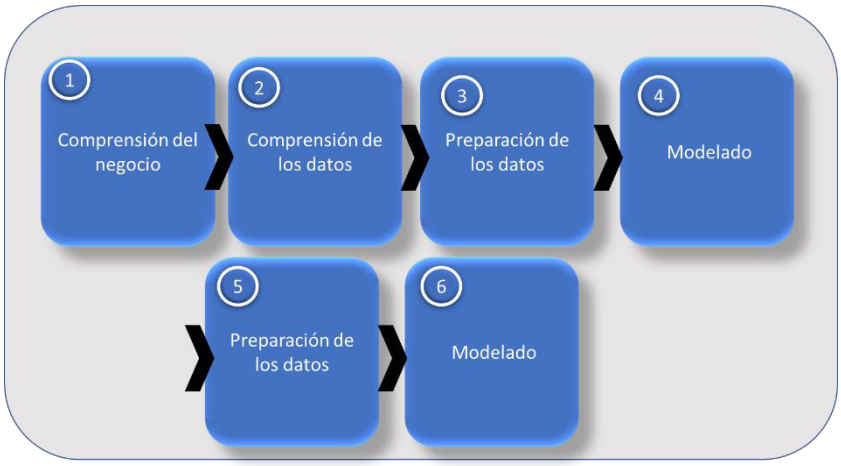


Figura 1. Fases del CRISP-DM

Fuente: Elaboración propia con base en datos de Mejías (2018).

Para la gestión del ambiente en Nube se usará la metodología *Microsoft Solutions Framework* (MSF), el cual es un marco de trabajo de referencia para construir e implantar sistemas empresariales distribuidos basados en herramientas y tecnologías de Microsoft (Alavandhar y Nikiforova, 2017). Este comprende un conjunto de modelos, conceptos y guías que contribuyen a alinear los objetivos de negocio y tecnológicos, reducir los costos de la utilización de nuevas tecnologías, y asegurar el éxito en la implantación de las tecnologías Microsoft. Asimismo, este marco estará detallado en el capítulo 3 y está representado por cinco fases las cuales son: Visión, planificación, desarrollo, estabilización, liberación.

2. MARCO TEÓRICO

A continuación, se presenta la exposición del conjunto de investigaciones, teorías y conceptos en los que se basa la presente investigación y que recopila información científica existente sobre el tema del diseño de CDW. En este capítulo trataremos la implementación de un CDW sobre una arquitectura en la Nube a través del empleo de las herramientas de desarrollo y de visualización de datos, así como el empleo de la Metodología CRISP-DM y MSF respectivamente.

De igual manera, se expondrá la implementación de un CDW con el apoyo de la plataforma de Amazon con el propósito de desarrollar de forma eficiente la contextualización de un almacén de datos tipo CDW, así como definir los elementos básicos vinculados sobre el establecimiento de la arquitectura en la Nube.

2.1. Cloud Data Warehouse

Desde hace unos años es incuestionable que el uso de un CDW avanza a pasos agigantados, ya que es una parte fundamental para las organizaciones a la hora de tomar decisiones estratégicas. Originalmente, los CDW se construyeron para centros de datos locales, pero estos requieren de un hardware costoso, actualizaciones que demandan mucho tiempo, mantenimiento constante y administración de interrupciones y, a medida que aumenta la cantidad de datos recopilados, se incrementan sus costos. Además, los centros de datos locales no permiten aumentar, disminuir o suspender las cargas de trabajo elásticamente según sea necesario. Si pensamos en alojarlos en la Nube, en cambio, es posible recopilar aún más datos de una multitud de fuentes y escalar de forma instantánea y elástica para admitir usuarios y cargas de trabajo prácticamente ilimitados. De esta manera, se necesita mayor agilidad, versatilidad, resiliencia y escalabilidad para adaptarse a la nueva normalidad y la Nube, entre otros factores, les proporciona todos esos elementos (Eklund, 2019). Seguidamente, en la Figura 2 se presenta la contextualización de un CDW en una empresa.

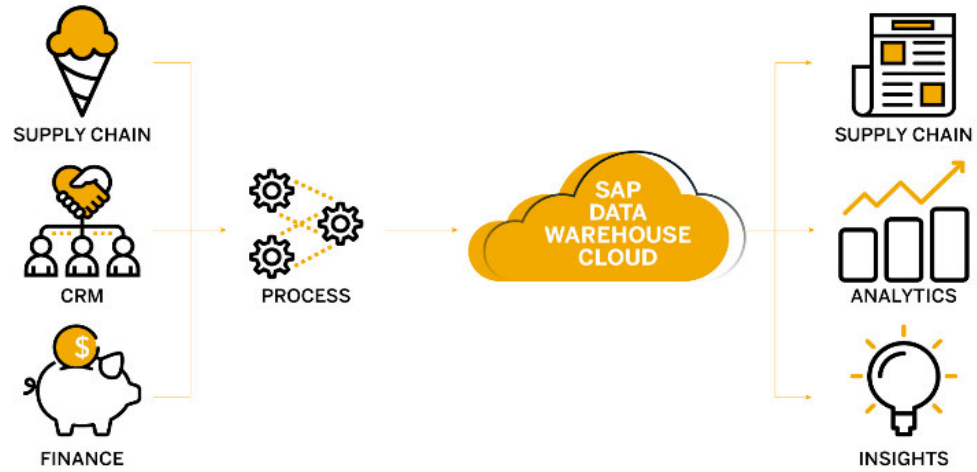


Figura 2. Contextualización de la aplicabilidad de un Cloud Data Warehouse para una empresa.

Fuente: tomado de SAP (2022)

Un CDW funciona como un repositorio central donde llega la información de una o más fuentes de datos. Los datos fluyen hacia un almacén de datos desde el sistema transaccional y otras bases de datos relacionales, los datos pueden ser: estructurados, semiestructurados o no estructurados. Estos datos se procesan, transforman e incorporan para que los usuarios puedan acceder a los datos procesados en el almacén de datos a través de herramientas de Business Intelligence, clientes SQL y hojas de cálculo. Un almacén de datos combina información proveniente de diferentes fuentes en una base de datos completa (SAP, 2022).

Al fusionar toda esta información en un solo lugar, una organización puede analizar a sus clientes de manera más integral. Esto ayuda a garantizar que se ha considerado toda la información disponible. El almacenamiento de datos hace posible la minería de datos, la cual busca patrones en los datos que puedan conducir a mayores ventas y ganancias (SAP, 2022).

2.1.1. Etapas generales del Cloud Data Warehouse

Anteriormente, las organizaciones comenzaron con un uso relativamente simple del almacenamiento de datos (Garani *et al.,* 2019). Sin embargo, con el tiempo, comenzó un empleo más sofisticado del almacenamiento de datos. Seguidamente, se presentan las etapas generales de uso del almacén de datos:

Base de datos operativa fuera de línea: En esta etapa, los datos simplemente se copian de un sistema operativo a otro servidor. De esta forma, la carga, el procesamiento y la generación de informes de los datos copiados no afectan el rendimiento del sistema operativo.

Almacén de datos fuera de línea: Los datos del *Datawarehouse* se actualizan regularmente desde la base de datos operativa. Los datos en CDW se mapean y transforman para cumplir con los objetivos de CDW.

Almacén de datos en tiempo real: En esta fase, los almacenes de datos se actualizan cada vez que se realiza una transacción en la base de datos operativa. Por ejemplo, sistemas de reservas de líneas aéreas o ferroviarias.

Almacén de datos integrado: En esta etapa, los almacenes de datos se actualizan continuamente cuando el sistema operativo realiza una transacción. Luego, el CDW genera transacciones que se devuelven al sistema operativo.

2.1.2. Propiedades del Cloud Data Warehouse

Los CDW, disponen de ciertas características que le han otorgado reconocimiento e importancia en las organizaciones, Según Eklund (2019) entre las más relevantes se presentan las siguientes:

Orientación a la empresa: Los datos serán almacenados y organizados. Un almacén de datos estará organizado a temas importantes tales como clientes, proveedores, productos y ventas realizadas.

Integrado: Estará diseñado para almacenar todos los datos empresariales. Esto conlleva utilizar diferentes fuentes de datos heterogéneas, como bases de datos relacionales, archivos planos y registros transaccionales procedentes de distintos sistemas.

Cambiantes con el tiempo: Los datos se almacenarán para proporcionar información desde una perspectiva histórica, pero a su vez, todos los cambios producidos en los datos deberán ser registrados para poder reflejar todas las variaciones en el tiempo.

2.1.3. Mejores prácticas para implementar un Cloud Data Warehouse

Una buena práctica está representada por los aspectos técnicos y metodológicos, que han permitido obtener buenos resultados en torno a una gestión en TI, en este contexto y en

lo que respecta a un CDW, Baker y Thien (2020) menciona que se deben considerar las siguientes pautas para su implementación:

- Definir un plan para comprobar la coherencia, precisión e integridad de los datos.
- El almacén de datos debe estar bien integrado, definido y con una planificación establecida.
- Al diseñar el almacén de datos, comprobar el empleo de herramienta adecuadas, respete el ciclo de vida, controlar los conflictos de datos y aprender de los errores.
- Nunca sustituir los sistemas e informes operativos
- No invertir demasiado tiempo en extraer, limpiar y cargar datos.
- Asegurarse de involucrar a todas las partes interesadas, incluido el personal de la empresa, en el proceso de implementación del CDW. Establezca que es un proyecto conjunto. No hay que crear un almacén de datos que no sea útil para los usuarios finales.
- Preparar un plan de formación para los usuarios finales.

2.1.4. Beneficios de un Cloud Data Warehouse

Entre los principales beneficios de la implementación de un CDW, Según Eklund (2019) son los siguientes:

Escalabilidad: La gran ventaja de la Nube sobre la solución local es que las posibles ampliaciones pueden realizarse fácilmente y sin esfuerzo. La ampliación de los sistemas locales es una tarea que consume mucho tiempo y recursos, ya que generalmente implica la compra e instalación de nuevo hardware. Por el contrario, los datos que se mantienen en la Nube se pueden ampliar o reducir al instante sin problemas

Coste: El almacenamiento de datos en la Nube ofrece beneficios de costes considerables al eliminar los costes iniciales que suelen ser los más elevados. Se eliminan los costes de hardware, salas de servidores, problemas de personal relacionados con IT o costes operativos para mantener su CDW.

Conectividad: Un almacén de datos en la Nube también facilita la conexión e integración con otros servicios en la Nube para ayudarlo a manipular mejor sus datos.

Confiabilidad: La ventaja de los CDW basados en la Nube es que siempre están disponibles. El tiempo de actividad y la confiabilidad están garantizados a través del SLA de su proveedor.

Rapidez: El almacén de datos permite a los usuarios comerciales acceder rápidamente a datos críticos de algunas fuentes, todo en un solo lugar.

Tiempo: El almacén de datos ayuda a reducir el tiempo de respuesta total para el análisis y la generación de informes.

Datos: El almacén de datos almacena una gran cantidad de datos históricos. Esto ayuda a los usuarios a analizar diferentes períodos de tiempo y tendencias para hacer predicciones futuras.

2.1.5. Almacén de datos

Es un proceso para recopilar y administrar datos de diversas fuentes para proporcionar información comercial significativa. Adicionalmente, se define como un almacén de datos y generalmente se usa para conectar y analizar datos comerciales de fuentes heterogéneas. Asimismo, este puede formar parte del núcleo de sistemas de *business intelligence* (BI) que está diseñado para el análisis y la generación de informes de datos (Garani et al., 2019).

2.1.5.1. Componentes del almacén de datos

Un almacén de datos está conformado por un conjunto de elementos claves o componentes, según Baker y Thien (2020) se tiene como principales:

Administrador de carga: También se denomina componente frontal. Realiza todas las operaciones asociadas a la extracción y carga de datos al almacén. Estas operaciones incluyen transformaciones para preparar los datos para ingresar al CDW.

Gerente de Almacén: Este efectúa operaciones asociadas con la gestión de los datos en el almacén, como el análisis de datos para garantizar la consistencia, creación de índices y vistas, generación de desnormalización y agregaciones, transformación y fusión de datos de origen y archivos, así como la copia de seguridad de los datos.

Administrador de consultas: También se conoce como componente de back-end. Ejecuta todas las operaciones operativas relacionadas con la gestión de las consultas de los

usuarios y las operaciones de estos componentes del almacén de datos son consultas directas a las tablas adecuadas para programar la ejecución de consultas.

Herramientas de acceso del usuario final: Esto se clasifica en cinco grupos diferentes, como: informes de datos, herramientas de consulta, herramientas de desarrollo de aplicaciones, herramientas *Executive Informations System* (EIS), Herramientas *Overview of Online Analytical Processing* (OLAP) y herramientas de minería de datos.

2.1.5.2. Tipos de almacén de datos

Los almacenes de datos son categorizados según diversos criterios, según Garani *et al*. (2019) los tres principales tipos de almacén de datos son:

Almacén de datos empresariales (EDW): *Enterprise Data Warehouse* (EDW) es un almacén centralizado. Proporciona un servicio de soporte de decisiones en toda la empresa. Ofrece un enfoque unificado para organizar y representar datos. También proporciona la capacidad de clasificar los datos según el tema y dar acceso según esas divisiones.

Almacén de datos operativos: El almacén de datos operativos, que también se denomina *Operational Data Store* (ODS), es un almacén de datos necesario cuando ni el almacén de datos ni los sistemas *Online transaction processing* (OLTP) admiten las necesidades de informes de las organizaciones. En ODS, el almacén de datos se actualiza en tiempo real. Por lo tanto, es ampliamente empleado para actividades rutinarias como el almacenamiento de registros de los empleados.

Data mart: Un *data mart* es un subconjunto del almacén de datos. Está especialmente diseñado para una línea de negocio en particular, como ventas o finanzas. En un data mart independiente, los datos se pueden recopilar directamente de las fuentes.

2.1.5.3. Pasos para implementar el almacén de datos

La mejor manera de abordar el riesgo comercial asociado con una implementación de CDW según Harvy *et al.* (2019) es emplear una estrategia de tres puntas como se muestra a continuación:

Estrategia empresarial: Se identifican técnicas, incluida la arquitectura y las herramientas actuales, también se identifican hechos, dimensiones y atributos. Adicionalmente, se pasa el mapeo y la transformación de datos.

Entrega por etapas: La implementación del almacén de datos debe realizarse por etapas en función de las áreas temáticas, por lo que las entidades comerciales relacionadas, como la reserva y la facturación, deben implementarse primero y luego integrarse entre sí.

Creación de prototipos iterativos: Es recomendable que el almacén de datos debe desarrollarse y probarse de forma iterativa.

A continuación, en la Tabla 1 se presentan los pasos clave en la implementación de un CDW junto con sus resultados.

Tabla 1
Pasos para la implementación de un CDW

Paso	Tareas	Entregables
1	Necesidad de definir el alcance del proyecto	Definición del alcance
2	Necesidad de determinar las necesidades del negocio.	Modelo de datos lógicos
3	Definir los requisitos del almacén de datos operativo	Modelo de almacén de datos operativos
4	Adquirir o desarrollar herramientas de extracción	Extraer herramientas y software
5	Definir los requisitos de datos del almacén de datos	Modelo de datos de transición
6	Documentar datos faltantes	Lista de proyectos pendientes
7	Asigna el almacén de datos operativos al almacén de datos	D/W *Data Integration Map*
8	Desarrollar el diseño de la base de datos del almacén de datos	Diseño de base de datos CDW
9	Extraer datos del almacén de datos operativos	Extractos de datos CDW integrados
10	Almacén de datos de carga	Carga de datos inicial
11	Mantener el almacén de datos	Acceso a datos en curso y cargas posteriores

Fuente: tomado de Harvy *et al.* (2019)

2.1.5.4. Ventajas del Almacén de Datos

Según Eklund (2019), entre las principales ventajas se presentan los siguientes:

- El almacén de datos permite a los usuarios comerciales acceder rápidamente a datos críticos de algunas fuentes, todo en un solo lugar.
- El almacén de datos proporciona información consistente sobre varias actividades multifuncionales.

- Ayuda a integrar muchas fuentes de datos para reducir el estrés en el sistema de producción.
- El almacén de datos ayuda a reducir el tiempo de respuesta total para el análisis y la generación de informes.
- La reestructuración y la integración facilitan el uso de informes y análisis por parte del usuario.
- El almacén de datos permite a los usuarios acceder a datos críticos de la cantidad de fuentes en un solo lugar. Por lo tanto, ahorra tiempo al usuario al recuperar datos de múltiples fuentes.
- Estos almacenan una gran cantidad de datos históricos ayudando a los usuarios a analizar diferentes períodos de tiempo y tendencias para hacer predicciones futuras.

2.1.5.5. Desventajas del almacén de datos

Según Eklund (2019), entre las principales desventajas se presentan los siguientes:

- No es una opción ideal para datos no estructurados.
- El almacén de datos puede quedar obsoleto con relativa rapidez
- Es difícil realizar cambios en los tipos y rangos de datos, el esquema de fuente de datos, los índices y las consultas.
- El almacén de datos puede parecer fácil, pero en realidad es demasiado complejo para los usuarios promedio.
- A pesar de los mejores esfuerzos en la gestión de proyectos, el alcance del proyecto de almacenamiento de datos siempre aumentará.
- En algún momento, los usuarios del almacén desarrollarán diferentes reglas comerciales.
- Las organizaciones necesitan gastar muchos de sus recursos para fines de capacitación e implementación.

2.1.5.6. Herramientas de almacenamiento de datos

Entre algunas de las algunas de las herramientas existentes para CDW más empleadas en el marcado para el desarrollo del almacenamiento de datos se presentan las siguientes:

SAP: Esta es una plataforma integrada de gestión de datos, para mapear todos los procesos de negocio de una organización. Es un conjunto de aplicaciones de nivel empresarial para sistemas abiertos de cliente/servidor. Además, es una de las mejores herramientas de almacenamiento de datos que ha establecido nuevos estándares para brindar las mejores soluciones de administración de información empresarial (SAP, 2022).

Características:

- Proporciona soluciones comerciales altamente flexibles y transparentes.
- La aplicación desarrollada con SAP puede integrarse con cualquier sistema.
- Sigue el concepto modular para una fácil instalación y utilización del espacio.
- Se puede crear un sistema de base de datos que combine análisis y transacciones. Al mismo tiempo, estas bases de datos de última generación se pueden implementar en cualquier dispositivo.
- Proporcione soporte para la implementación local o en la Nube.
- Arquitectura de almacenamiento de datos simplificada.
- Integración con aplicaciones SAP y no SAP.
- Soporte por vía de correo electrónico y formulario en línea.
- Prueba sin costo con la versión gratuita básica.

MarkLogic: Representa una solución útil de almacenamiento de datos que hace que la integración de datos sea más fácil y rápida mediante una variedad de funciones empresariales. Esta herramienta ayuda a realizar operaciones de búsqueda muy complejas. Puede consultar diferentes tipos de datos como documentos, relaciones y metadatos (MarkLogic, 2022).

Características:

- La Interfaz de Programa de Aplicación (API) de Optic puede realizar uniones y agregados sobre documentos y filas.
- Permite especificar reglas de seguridad más complejas para todos los elementos dentro de los documentos.
- Escritura, lectura, aplicación de parches y eliminación de documentos en formato JSON, XML, texto o binario.
- Replicación de base de datos para recuperación ante desastres.

- Permite especificar las opciones de salida en la configuración del servidor de aplicaciones.
- Importación y exportación de información de configuración.

Oracle Autonomous Database: El software de almacén de datos de Oracle es una colección de datos que se trata como una unidad. El propósito de esta base de datos es almacenar y recuperar información relacionada. Ayuda al servidor a administrar de manera confiable grandes cantidades de datos para que varios usuarios puedan acceder a los mismos datos (Oracle ADW, 2022).

Características:

- Distribuye los datos de la misma manera entre los discos para ofrecer un rendimiento uniforme.
- Funciona para clústeres de aplicaciones reales y de instancia única.
- Ofrece pruebas de aplicaciones reales.
- Arquitectura común entre cualquier Nube Privada y la Nube pública de Oracle.
- Conexión de alta velocidad para mover grandes datos.
- Funciona de forma eficiente con las plataformas UNIX/Linux y Windows.
- Proporciona soporte para la virtualización.
- Permite conectarse a la base de datos, tabla o vista remota.
- Soporte a través de Formulario en línea.
- Proporciona una prueba gratuita de 30 días.

Amazon RedShift: Es una herramienta de almacenamiento de datos. Es una herramienta simple y rentable para analizar todo tipo de datos utilizando SQL estándar y herramientas de BI existentes. También permite ejecutar consultas complejas contra petabytes de datos estructurados, utilizando la técnica de optimización de consultas (Amazon Redshift, 2022).

Características:

- Sin costes iniciales para su instalación.
- Permite automatizar la mayoría de las tareas administrativas comunes para supervisar, gestionar y escalar el almacén de datos.
- Permite cambiar el número o el tipo de nodos.
- Ayuda a mejorar la fiabilidad del clúster del almacén de datos.
- Cada centro de datos está totalmente equipado con control de climatización.

- Supervisa continuamente la salud del clúster. Adicionalmente, replica automáticamente los datos de las unidades que fallan y sustituye los nodos cuando es necesario.
- Soporte por medio de correo electrónico, formulario en línea.
- Ofrece una prueba sin pago con la capa gratuita de AWS.

2.2. Computación en la Nube

La computación en la Nube se define como el almacenamiento y el acceso a datos y servicios informáticos a través de Internet con la disponibilidad bajo demanda de servicios informáticos como servidores, almacenamiento de datos, redes, bases de datos, entre otros. El objetivo principal de la computación en la Nube es dar acceso a los centros de datos a muchos usuarios y estos pueden acceder a los datos desde un servidor remoto (Rodríguez, 2020).

2.2.1. Arquitectura en la Nube

La arquitectura en la Nube representa una serie de componentes interconectados, desde herramientas y aplicaciones de software hasta redes y almacenamiento de servidores, los cuales se combinan para formar una Nube de recursos compartidos (Rodríguez, 2020). También, se refiere a la infraestructura completa de hardware y software que las empresas e instituciones utilizan para crear, indexar, almacenar y compartir grandes cantidades de datos de múltiples usuarios y ubicaciones. Asimismo, los bloques de construcción más básicos de la arquitectura de la Nube se representan como entrega de *front-end, back-end* o basada en la Nube.

El *front-end* de la Nube representa el punto en el que un usuario interactúa con los clientes de software, las interfaces de usuario y los dispositivos o redes de los clientes. Esto puede ser tan simple como una aplicación de correo electrónico o tan complejo como herramientas de análisis profundas basadas en Inteligencia Artificial (IA). Al mismo tiempo, el *back-end* de la arquitectura de la Nube puede denominarse simplemente el hardware real detrás de la Nube, desde el almacenamiento de datos hasta los procesadores y los conmutadores de red, también conocido como Infraestructura como Servicio (IaaS) (Rodríguez, 2020).

2.2.1.1. Fundamentos de la arquitectura en la Nube

El *front-end* que está representado por los clientes y dispositivos utilizados para virtualizar, o para acceder y administrar, todos los datos de la Nube (ITU, 2012). Estas herramientas *front-end* pueden variar desde web virtual y aplicaciones móviles hasta herramientas complejas de análisis y automatización, según las necesidades particulares de una organización:

El *back-end* se compone de servidores virtuales, almacenamiento e infraestructura, como CPU y GPU, conmutadores de red y tarjetas aceleradoras que potencian el acceso y las consultas de los usuarios. A diferencia del hardware de red tradicional y los centros de datos internos, la Nube permite a las empresas escalar fácilmente a medida que cambian sus necesidades sin requerir de comprar y mantener su propio equipo (ITU, 2012).

2.2.1.2. Tipos de arquitectura en la Nube

Según ITU (2012), existen tres tipos de arquitectura en la Nube, los cuales se detallan a continuación:

Nube pública: Representa un marco completo de terceros que ofrecen recursos informáticos como redes, memoria, procesamiento y almacenamiento. Este es el tipo de computación en la Nube más común en la actualidad, lo que permite a las empresas escalar sus recursos según sea necesario sin comprar ni mantener su propio hardware o software.

Nube privada: En este modelo, la organización gestiona todo el sistema en la Nube. La decisión de mantener un entorno de Nube privada a menudo se debe a la seguridad y soberanía de los datos, el cumplimiento de la industria o la disponibilidad de recursos de almacenamiento y procesamiento. Una Nube privada puede ser alojada por un tercero o como parte del propio centro de datos de una empresa.

Nube híbrida: Esta ofrece la mejor solución de ambos modelos, en la que una organización mantiene una Nube privada optimizada para sus propios recursos y al mismo tiempo puede aprovechar los vastos recursos de la Nube pública debido al costo y la escalabilidad. Por lo que, una Nube híbrida combina elementos de Nube pública y privada conectados de forma segura a través de una red privada virtual (VPN) o un canal privado.

2.2.1.3. Ventajas de la arquitectura en la Nube

La arquitectura de computación en la Nube permite que las organizaciones reduzcan o eliminen su dependencia de la infraestructura de servidor, almacenamiento y red en las instalaciones. Por lo tanto, las organizaciones que adoptan la arquitectura de la Nube a menudo trasladan los recursos de TI a la Nube pública, eliminando la necesidad de servidores y almacenamiento en las instalaciones, y reduciendo la necesidad de espacio, refrigeración y energía del centro de datos de TI, y reemplazándolos con un gasto mensual de TI. Es por ello, que este cambio de gasto de capital a gasto operativo es una de las principales razones de la popularidad actual de la computación en la Nube (Rodríguez, 2020).

2.2.1.4. Desventajas de la arquitectura en la Nube

La arquitectura de computación en la Nube también puede presentar inconvenientes al de rendimiento variable, ya que proporciona recursos proporciona recursos simultáneamente a otras empresas y cualquier inconveniente del arrendatario y podría afectar el rendimiento de sus recursos compartidos (Rodríguez, 2020).

Por otra parte, se pueden presentar problemas técnicos, debido a que pueden estar propensos a interrupciones y otros problemas técnicos. Incluso, las mejores empresas proveedoras de servicios en la Nube pueden enfrentarse a este tipo de problemas a pesar de mantener altos estándares de mantenimiento (Rodríguez, 2020).

Asimismo, otro inconveniente al trabajar con servicios de computación en la Nube es el riesgo de seguridad, ya que antes de adoptar la tecnología de la Nube, se debe ser consciente del hecho de que se compartirá toda la información confidencial de la empresa con un proveedor de servicios de computación en la Nube de terceros y existe el riesgo de que los piratas informáticos pueden acceder a esta información (Rodríguez, 2020).

2.2.2. Servicios en la Nube IaaS, PaaS y SaaS

La infraestructura como servicio (IaaS) es una forma de computación en la Nube que proporciona recursos informáticos virtualizados a través de Internet. En el modelo IaaS, el proveedor de la Nube administra las infraestructuras de TI, como los recursos de almacenamiento, servidores, redes, y los entrega a las organizaciones suscriptoras a través de máquinas virtuales accesibles a través de Internet. IaaS puede tener muchos beneficios para las organizaciones, como hacer que las cargas de trabajo sean más rápidas, fáciles, flexibles y rentables (Amron et al., 2017).

Por otra parte, la plataforma como servicio (PaaS) es un modelo de computación en la Nube en el que un proveedor externo ofrece herramientas de hardware y software a los usuarios a través de Internet. Por lo general, estas herramientas son necesarias para el desarrollo de aplicaciones. Por lo que, un proveedor de PaaS aloja el hardware y el software en su propia infraestructura. Como resultado, PaaS libera a los desarrolladores de tener que instalar hardware y software internos para desarrollar o ejecutar una nueva aplicación (Amron et al., 2017).

Además, el software como servicio (SaaS) es un modelo de distribución de software en el que un proveedor de la Nube aloja aplicaciones y las pone a disposición de los usuarios finales a través de Internet. En este modelo, un proveedor de software independiente puede contratar a un proveedor de Nube externo para alojar la aplicación o a través de empresas más grandes, como Microsoft (Amron et al., 2017).

2.2.3. Proveedores de la Nube

2.2.3.1. Azure

Microsoft Azure es una plataforma de Nube pública con más de 200 productos y servicios accesibles a través de la Internet pública. Al igual que otros proveedores de Nube pública, Azure administra y mantiene el hardware, la infraestructura y los recursos a los que se puede acceder de forma gratuita o de pago por uso, bajo demanda con soluciones que incluyen infraestructura como servicio (IaaS), plataforma como servicio (PaaS) y software como servicio (SaaS) que se pueden usar para servicios como análisis, informática virtual, almacenamiento, redes, entre otros. y se puede utilizar para reemplazar o complementar servidores locales de empresas (Azure, 2021).

Ventajas de Azure

Según Azure (2021), entre las ventajas y fortalezas de Azure se pueden mencionar las siguientes:

- Azure tiene muchos centros de datos y siguen expandiéndose, lo que significa que los servicios y sus aplicaciones estarán más cerca de los usuarios.
- Debido a que Microsoft ha brindado soporte a clientes locales durante más de 40 años, tienen una amplia oferta de Nube híbrida para llevar a todos sus clientes existentes a la Nube.

- Presenta una muy buena integración con las herramientas y tecnologías existentes, como Visual Studio, Active Directory y File Storage.
- Soporte para Nube híbrida: Permiten aprovechar infraestructura existente y combinarla con la escalabilidad que ofrece Azure. También ofrecen soluciones para llevar la inteligencia de la Nube al borde de los dispositivos IoT.

Desventajas de Azure

Según Azure (2021), entre los contras y debilidades de Azure se pueden mencionar las siguientes:

- Dado que Azure está tratando de abarcar todo el espacio posible de computación en la Nube, a veces algunos servicios simplemente no reciben suficiente atención. Esto puede significar que algunos servicios de análisis de datos que utiliza una determinada característica de Azure pueden retrasarse un poco.
- Azure intentará mantenerse al día con todas las tendencias de la computación en la Nube, por lo que la cantidad de servicios nuevos y servicios existentes puede ser muy extensa, por lo que se debe centrarse solo en los que se necesita para un proyecto.

2.2.3.2. Google Cloud

Google Cloud Platform es conocido por su amplia gama de ofertas de SaaS familiares para los usuarios cotidianos. Además, tienen versiones empresariales de sus productos con seguridad avanzada y personalizaciones para clientes comerciales, junto con otras herramientas de productividad y colaboración. Además, tienen una amplia gama de soluciones analíticas y de Nube pre empaquetadas para diferentes industrias, así como soporte para la transformación digital y la modernización de la infraestructura (Google Cloud, 2022).

Además de los productos SaaS y IaaS este ofrece el motor de cómputo de Google, ofrece máquinas virtuales, también proporciona la Nube de Google tiene App Engine como plataforma para desarrollar aplicaciones. Asimismo, ofrece servicios de funciones en la Nube para que los desarrolladores pueden usar estas funciones para automatizar procesos y configurarlos para que se activen según sea necesario y proporciona un modelo de pago por uso (Google Cloud, 2022).

Ventajas de Google Cloud

Según Google Cloud (2022), entre las ventajas y fortalezas de Google Cloud se pueden mencionar las siguientes:

- Opciones gratuitas, Google Cloud ofrece más de 20 productos gratuitos. Aparte de esto, también ofrecen créditos de 300 USD para probar completamente toda la plataforma Google Cloud.
- Precios detallados y transparentes, ofrecen un modelo de pago por uso. También, brindan herramientas para ayudar a administrar y limitar gastos, así como alertas a tiempo.
- Google Cloud Platform funciona en la misma infraestructura que ofrece todos los productos de consumo de Google. Esto garantiza un alto nivel de confiabilidad y tiempo de actividad, garantizando el 99,5% de tiempo de actividad para la mayoría de sus servicios.

Desventajas de Google Cloud

Según Google Cloud (2022), entre los contras y debilidades de Google Cloud se pueden mencionar las siguientes:

- No tienen la gama más amplia de servicios de los tres principales actores en la computación en la Nube, Google tiene el catálogo más pequeño de productos y servicios.
- No es el más experimentado y una vez más, Google parece ser el menos experimentado en la prestación de servicios empresariales en la Nube. Pero son uno de los líderes en productos SaaS de consumo.

2.2.3.3. Amazon

AWS o Amazon Web Services es un proveedor de servicios en la Nube que ofrece varios servicios informáticos a los que se puede acceder a través de Internet público. Asimismo, administran y mantienen el hardware y la infraestructura, lo que ahorra a las organizaciones y a las personas el costo y la complejidad de comprar y ejecutar recursos en el sitio. Se puede acceder a estos recursos de forma gratuita o mediante pago por uso (AWS, 2022).

Según AWS (2022), esta ópera globalmente en lo que denominan regiones, teniendo 25 en total repartidas en seis continentes. Cada región consta de varias zonas de

disponibilidad. Y estos son los centros de datos físicos donde se alojan las computadoras y están separadas geográficamente para reducir la probabilidad de que un desastre local afecte a toda una región. Además, hay más de 200 ubicaciones de borde repartidas por todo el mundo como parte de la red de entrega de contenido de AWS. Por otra parte, las regiones están disponibles públicamente y hay algunas regiones especiales. Dos regiones están designadas para uso de aquellos que trabajan en y para el gobierno de los EE. UU. denominado AWS GovCloud. Asimismo, hay dos regiones en China, que son operadas por empresas locales calificadas de acuerdo con las leyes y regulaciones chinas. Todas las zonas de disponibilidad y las ubicaciones de borde de estas regiones están unidas a través de la red privada de fibra óptica de alta velocidad propiedad de AWS.

AWS proporciona una variedad tan amplia de servicios que se adapta casi a cualquier caso de uso. Los servicios van desde el almacenamiento básico y la computación hasta los servicios de nicho más especializados, como la transmisión de medios, la robótica e incluso la computación cuántica. También, dispone de un servicio que permite controlar flotas de satélites espaciales. Asimismo, de los casos de uso habituales de centros de datos remotos, las organizaciones aprovechan cada vez más AWS como una inversión en aprendizaje automático y análisis de datos para ayudar a entender los datos de las organizaciones (AWS, 2022).

Ventajas de AWS

Según AWS (2022), entre las ventajas y beneficios de AWS se pueden mencionar las siguientes:

- En cuanto a las fortalezas, debido a que AWS tenía una gran ventaja sobre los competidores actuales, AWS tiene la oferta más sólida y completa, lo que representa que actualmente dispone de más de 200 servicios, dispone de nuevas características, mejoras y servicios que salen al mercado semanalmente y muchas de esas nuevas funciones son el resultado directo de las solicitudes de los clientes.
- Largo historial de reducciones de precios y ofrece muchas herramientas y programas para ayudar a las empresas a optimizar sus gastos.
- Prueba gratuita en muchos servicios, Amazon Web Services ofrece muchos de sus productos en la Nube, incluido EC2 en una prueba gratuita. Para algunos productos, garantizan una prueba gratuita de 12 meses.

- Fácilmente escalable, Amazon es fácil de escalar hacia arriba o hacia abajo, lo que lo ayuda a crear rápidamente aplicaciones sólidas sin grandes inversiones, y las ubicaciones de sus servidores están disponibles en todo el mundo, por lo que puede obtener soporte asegurado del proveedor si decide globalizarse.
- Múltiples modelos de pago, además del modelo de pago por uso, AWS también ofrece un modelo de ahorro cuando se contrata. Por lo tanto, puede optar por uno o dos años de compromiso a cambio de precios más bajos, y para algunos servicios, ofrecen precios escalonados, lo que significa que su costo por uso disminuye a medida que más se utiliza.

Desventajas de AWS

Según AWS (2022), entre los contras y debilidades de AWS se pueden mencionar las siguientes:

- Algunas empresas minoristas ven a Amazon como un competidor directo y no se atreven a contratar sus servicios por considerarlo un rival y prefieren adoptar un enfoque de múltiples Nubes.
- AWS no cobra por subir datos en su Nube, pero se debe pagar un bajo importe para recuperar esos datos. Estos se denominan "*cargos de salida*".

2.2.4. Factores para seleccionar un proveedor de la Nube

Según Amron et al. (2017), indican que, al momento de seleccionar un proveedor de la Nube, se deben considerar ciertos elementos para asegurar una adecuada implementación, así como un correcto funcionamiento y seguridad, entre los más relevantes se pueden mencionar los siguientes:

2.2.4.1. Procesos y salud empresarial

Salud financiera: El proveedor debe tener un historial de estabilidad y estar en una posición financiera saludable con suficiente capital para operar con éxito a largo plazo.

Organización, gobierno, planificación y gestión de riesgos: El proveedor debe tener una estructura de gestión formal, políticas de gestión de riesgos establecidas y un proceso formal para evaluar a los proveedores de servicios y sus terceros.

Confianza: Comprobable reputación del proveedor y de sus socios. Se debe investigar el nivel de experiencia en la Nube y buscar reseñas de sus clientes.

Conocimiento empresarial y técnico: El proveedor debe comprender el negocio de sus clientes y lo que está buscando hacer, así como poder combinarlo con su experiencia técnica.

Auditoría de cumplimiento: El proveedor debe poder validar el cumplimiento de todos los requisitos del cliente a través de una auditoría de terceros.

2.2.4.2. Apoyo a la administración

Acuerdos de nivel de servicio: Los proveedores deben poder comprometerse con un nivel de servicio con el que el cliente se sienta cómodo.

Informes de rendimiento: El proveedor debería poder brindar informes de rendimiento.

Supervisión de recursos y gestión de la configuración: Debe haber suficientes controles para que el proveedor rastree y monitoree los servicios proporcionados a los clientes y cualquier cambio realizado en sus sistemas.

Facturación y contabilidad: Esto debe estar automatizado para que pueda monitorear qué recursos está utilizando y el costo, para que no genere facturas inesperadas. También debe haber soporte para cuestiones relacionadas con la facturación.

2.2.4.3. Capacidades técnicas y procesos.

Facilidad de implementación, administración y actualización: El proveedor debe poseer mecanismos que faciliten implementar, administrar y actualizar software y aplicaciones de sus clientes.

Interfaces estándar: El proveedor debe usar API estándar y transformaciones de datos para que las organizaciones pueda establecer fácilmente conexiones con la Nube.

Gestión de eventos: El proveedor debe tener un sistema formal para la gestión de eventos que esté integrado con su sistema de seguimiento o gestión.

Gestión del cambio: El proveedor debe tener procesos documentados y formales para solicitar, registrar, aprobar, probar y aceptar cambios.

Capacidad híbrida: Incluso si inicialmente no se planea usar una Nube híbrida, el proveedor debe estar en capacidad de admitir este modelo, ya que proporciona ventajas que tal vez se puedan aprovechar posteriormente.

2.2.4.4. Prácticas de seguridad

Infraestructura de seguridad: Debe haber una infraestructura de seguridad integral para todos los niveles y tipos de servicios en la Nube.

Políticas de seguridad: Deben existir políticas y procedimientos de seguridad completos para controlar el acceso a los sistemas del proveedor y del cliente.

Gestión de identidad: Los cambios en cualquier servicio de aplicación o componente de hardware deben autorizarse en función de un rol personal o grupal y se debe requerir autenticación para que cualquier persona cambie una aplicación o datos.

Copia de seguridad y retención de datos: Deben existir políticas y procedimientos para garantizar la integridad de los datos de los clientes y estar en funcionamiento.

Seguridad física: Deben existir controles que garanticen la seguridad física, incluso para el acceso a los equipos informáticos ubicados en el mismo lugar. Además, los centros de datos deben contar con salvaguardias ambientales para proteger los equipos y los datos de eventos disruptivos. Igualmente, debe haber redes y energía redundantes y un plan documentado de recuperación de desastres y continuidad del negocio.

A continuación, se realiza una comparación por medio de los factores antes mencionados de los tres proveedores indicados para establecer cual se ajusta mejor para el proyecto planteado, por ello, se presenta en la Tabla 2 los elementos y las puntuaciones obtenidas en base a la contrastación realizada y las ponderaciones obtenidas, donde la más baja está representada por el numero 1 y la más alta 5.

Tabla 2
Comparación de proveedores AWS, Azure y Google Cloud

Clasificación	Factores	AWS	Azure	Google Cloud
Procesos y salud empresarial	Salud financiera	4	4	4
	Organización, gobierno, planificación y gestión de riesgos.	5	4	5
	Confianza	5	5	4
	Conocimiento empresarial y técnico.	5	5	5
	Auditoría de cumplimiento.	4	4	4
Apoyo a la administración	Acuerdos de nivel de servicio.	4	3	4
	Informes de rendimiento.	5	5	4
	Supervisión de recursos y gestión de la configuración.	4	4	4
	Facturación y contabilidad.	5	4	3
Capacidades técnicas y procesos.	Facilidad de implementación, administración y actualización.	5	4	4
	Interfaces estándar.	4	5	4
	Gestión de eventos.	4	4	4
	Gestión del cambio.	5	4	4
	Capacidad híbrida.	5	5	4
Prácticas de seguridad	Infraestructura de seguridad.	5	5	5
	Políticas de seguridad.	5	5	5
	Gestión de identidad.	4	4	4
	Copia de seguridad y retención de datos.	5	4	4
	Seguridad física.	5	5	5
Totales		**84**	**79**	**76**

Fuente: Elaboración propia basado en Amron et al. (2017)

2.2.4.5. Selección de proveedor

Una vez analizados los proveedores antes mencionados se puede indicar que no parece que haya grandes diferencias en la calidad de los servicios entre los proveedores AWS, Azure y Google Cloud, y esencialmente la elección se reduce a las necesidades. Por ejemplo, si se requiere una Nube híbrida, Azure puede ser la mejor opción, pero AWS ofrece mayores ventajas en lo que respecta a la gama de servicios.

Por otra parte, si la opción seleccionada está enfocada en el análisis, Google Cloud Platform puede ser una mejor opción, y aunque no hay una inversión inicial para comenzar a trabajar, elegir una solución en la Nube puede tener un impacto a largo plazo. Incluso con las opciones más flexibles, es probable que se limite el proveedor y la migración será complicada.

Sin embargo, por medio de la comparación empleando los factores para la selección del proveedor en la Nube, se presentó en la Tabla 2 los elementos y las puntuaciones obtenidas en base ducha contrastación realizada y basado en las ponderaciones obtenidas, se ha establecido para el tema de estudio la empresa Amazon Web Services por presentar mejores beneficios y reunir las condiciones apropiadas para realizar el desarrollo del proyecto.

2.3. Herramientas para el desarrollo del CDW

Seguidamente, se presenta la descripción del conjunto de herramientas que se emplean en la presente investigación para el diseño del CDW sobre una arquitectura en la Nube en la cual se apoya para crear los elementos necesarios para materializar la concepción, tales como: Herramientas tanto de Amazon como de Microsoft, servidores virtuales, bases de datos, canales de comunicación entre instancias, funciones, visualizadores de datos, plataformas de desarrollo y herramientas de transferencia de archivos, entre otros.

2.3.1. Instancia Amazon EC2

Según AWS (2022), una instancia de Amazon EC2 es un servidor virtual en Elastic Compute Cloud (EC2) de Amazon para ejecutar aplicaciones en la infraestructura de Amazon Web Services (AWS). Por lo tanto, esta es una plataforma informática en la Nube integral y en evolución; EC2 es un servicio que permite a los suscriptores comerciales ejecutar programas de aplicación en el entorno informático. Puede servir como un conjunto prácticamente ilimitado de máquinas virtuales (VM). Además, Amazon proporciona varios tipos de instancias con diferentes configuraciones de CPU, memoria, almacenamiento y recursos de red para adaptarse a las necesidades del usuario. Por lo que, cada tipo está disponible en varios tamaños para abordar los requisitos de carga de trabajo específicos.

2.3.1.1. Tipos de instancias EC2

Según AWS (2022), los tipos de instancias se agrupan en familias según los perfiles de aplicación de destino. Estos grupos incluyen lo siguiente:

Propósito general: Una instancia de propósito general es una máquina virtual que está diseñada para manejar una variedad de cargas de trabajo. Por lo tanto, están optimizadas

para tener una gran cantidad de núcleos de CPU, almacenamiento bajo demanda y memoria. Algunos casos de uso comunes para instancias de propósito general incluyen alojamiento de servidor web, desarrollo y prueba de software.

Computación optimizada: Las instancias optimizadas para computación se utilizan para ejecutar aplicaciones de big data que requieren grandes cantidades de potencia de procesamiento y memoria en la Nube de AWS. Por lo que, estas instancias están diseñadas y optimizadas para ejecutar aplicaciones informáticas y de uso intensivo de datos que requieren un rendimiento de red rápido, amplia disponibilidad y operaciones de entrada/salida (I/O) por segundo o *Input/Output Operations Per Second* (IOPS). Los ejemplos de tipos de aplicaciones incluyen modelado y simulación científica y financiera, aprendizaje automático, almacenamiento de datos empresariales e inteligencia comercial.

Unidad de procesamiento de gráficos (GPU): Estas instancias proporcionan una forma de ejecutar aplicaciones con uso intensivo de gráficos más rápido que con las instancias EC2 estándar. Por lo tanto, los sistemas que dependen de Graphics Processing Unit (GPU) incluyen juegos y trabajo de diseño. Por ejemplo, las distribuciones de Linux frecuentemente aprovechan las GPU para representar interfaces gráficas de usuario, mejorar las velocidades de compresión y acelerar las consultas a la base de datos.

Memoria optimizada: Las instancias con memoria optimizada utilizan una unidad de estado sólido de alta velocidad para proporcionar un acceso ultrarrápido a los datos y ofrecer un alto rendimiento. Estas instancias son ideales para aplicaciones que requieren más memoria y menos potencia de CPU, incluidas bases de datos de código abierto, análisis de big data en tiempo real y cachés en memoria.

Almacenamiento optimizado: Las instancias optimizadas para almacenamiento son ideales para aplicaciones que requieren un alto rendimiento de Entrada y Salida (E/S), como las bases de datos *Not Only SQL database* (NoSQL) que almacenan y recuperan datos en tiempo real. También, son ideales para aplicaciones que hacen un uso intensivo de la memoria, como el procesamiento de datos, el almacenamiento de datos, las cargas de trabajo de análisis y el procesamiento de registros.

Micro: Una instancia micro está diseñada para aplicaciones con bajo rendimiento. Por lo tanto, el tipo de instancia micro puede servir como un pequeño servidor de base de datos, como una plataforma para pruebas de software o como un servidor web que no requiere altas tasas de transacciones.

2.3.1.2. Ventajas de Amazon EC2

Según AWS (2022), entre las ventajas de Amazon EC2 son los siguientes:

Fiabilidad: Amazon EC2 ofrece una disponibilidad del 99,9 % en cada región de Amazon EC2. Los servicios son altamente confiables y las instancias se pueden reemplazar fácil y rápidamente.

La seguridad: Amazon colabora con Amazon *Virtual Private Cloud* (VPC) para proporcionar redes seguras y recursos informáticos. Las instancias informáticas se alojan en una Nube privada virtual o VPC y se les asigna un rango de direcciones IP, ayudando al usuario a determinar qué instancias están expuestas a Internet y cuáles se mantienen privadas.

Adaptabilidad: Se Puede elegir entre una variedad de tipos de instancias, paquetes de software, almacenamiento de instancias y sistemas operativos en EC2. Por lo que, este permite especificar la memoria, la CPU y el tamaño de la partición de arranque que es mejor para el sistema operativo y la aplicación.

2.3.1.3. Desventajas de Amazon EC2

Según AWS (2022), entre las desventajas de Amazon EC2 son los siguientes:

Limitaciones: Cuando se migra a la Nube, Amazon Web Services puede encontrar algunos problemas comunes de computación en la Nube, como tiempo de inactividad, control limitado y protección de respaldo.

Restricciones de seguridad: Debido a que la seguridad es una de las características más importantes, AWS restringe algunas de sus características que no se pueden cambiar en absoluto.

2.3.2. Servidor de base de datos Amazon DynamoDB

Amazon DynamoDB, también conocido como Dynamo Database o DDB, es un servicio de base de datos NoSQL completamente administrado proporcionado por Amazon Web Services y es conocido por sus bajas latencias y escalabilidad. Además, Amazon hace que sea simple y rentable almacenar y recuperar cualquier cantidad de datos, así como atender cualquier nivel de tráfico de solicitudes. Todos los elementos de datos se almacenan en unidades de estado sólido, que proporcionan un alto rendimiento de E/S y pueden manejar solicitudes a gran escala de manera más eficiente. Un usuario de AWS interactúa con el

servicio mediante la consola de administración de AWS o una API de DynamoDB (AWS, 2022).

DynamoDB utiliza un modelo de base de datos NoSQL, que no es relacional, lo que permite documentos, gráficos y columnas entre sus modelos de datos. Por lo que, un usuario almacena datos en tablas de DynamoDB, luego interactúa con ellos a través de consultas GET y PUT, que son operaciones de lectura y escritura, respectivamente. DynamoDB admite operaciones CRUD básicas y operaciones condicionales, donde cada consulta de DynamoDB se ejecuta mediante una clave principal identificada por el usuario, que identifica de forma exclusiva cada elemento (AWS, 2022).

2.3.2.1. Ventajas de Amazon DynamoDB

Según AWS (2022), entre las ventajas de Amazon DynamoDB son los siguientes:

Escalable: Almacenamiento ilimitado virtual, los usuarios pueden almacenar una cantidad infinita de datos según sus necesidades

Rentable: Reducción de costos, mientras que una gran parte de los datos puede migrar de SQL a NOSQL. Básicamente, cobra por leer, escribir y almacenar datos junto con cualquier característica opcional que se habilite en DynamoDB.

Replicación de datos: Todos los elementos de datos se almacenan en SSD y la replicación se administra internamente en varias zonas de disponibilidad en una región o puede estar disponible en varias regiones

Sin servidor: DynamoDB se escala horizontalmente al expandir una sola tabla en varios servidores

2.3.2.2. Desventajas de Amazon DynamoDB

Según AWS (2022), entre las desventajas de Amazon DynamoDB son los siguientes:

Uniones de tablas: Las uniones son imposibles, no dispone de Trigger. Con DynamoDB, solo se puede aprovisionar con capacidad en todo el nivel de la tabla.

Regional: Disponible solo para tablas de una sola región.

Restricciones: Limitado a un máximo de 10 elementos o 4 MB de datos. No se puede rediseñar todas las aplicaciones.

2.3.3. Amazon API Gateway

Amazon API Gateway es una característica de AWS que permite a los desarrolladores conectar aplicaciones que no son de AWS a recursos de back-end de AWS, como servidores y código. La puerta de enlace aumenta el acceso de los clientes de AWS a las aplicaciones compatibles y la utilidad general de los otros servicios en la Nube de Amazon. Por ello, una API permite que los programas de software se comuniquen, haciéndolos más funcionales. Además, un usuario de AWS puede crear, administrar y mantener las API dentro de Amazon API Gateway (AWS, 2022).

2.3.3.1. Tipos de API que admite Amazon API Gateway

API RESTful: Estas API se comunican con un servidor mediante métodos Hypertext Transfer Protocol (HTTP) como GET, POST, PUT y DELETE. Estos son los mismos métodos que se utilizan para acceder a páginas web y crear un recurso o realizar una acción. Con Amazon API Gateway, las API RESTful se utilizan para cargas de trabajo sin servidor y backends HTTP que utilizan API HTTP (AWS, 2022).

API de WebSocket: Esta API crea canales de comunicación bidireccionales a través de una única conexión de protocolo de control de transmisión. Facilita la comunicación cliente-servidor en aplicaciones en tiempo real, como juegos en línea, chat web y sistemas de negociación de acciones. A diferencia de las API HTTP tradicionales que dependen del cliente para iniciar la comunicación, la API de WebSocket[1] permite que el servidor envíe mensajes al cliente sin que el cliente los solicite (AWS, 2022).

2.3.3.2. Ventajas de Amazon API Gateway

Según AWS (2022), entre las ventajas de Amazon API Gateway son los siguientes:

Eficiente y escalable: Se puede ejecutar varias versiones de la misma API simultáneamente para que pueda iterar, probar y lanzar nuevas versiones rápidamente. Adicionalmente, se paga por las llamadas API y la transferencia de datos, por lo que no existen tarifas mínimas ni tarifas iniciales.

[1] WebSocket: Es un protocolo de comunicaciones para una conexión TCP dúplex completa, bidireccional y persistente desde el navegador web de un usuario a un servidor.

Monitoreo: Puede supervisar las métricas de rendimiento y la información sobre las llamadas a la API, la latencia de los datos y las tasas de error desde el panel de API Gateway, que le permite controlar visualmente las llamadas a los servicios.

2.3.3.3. Desventajas de Amazon API Gateway

Según AWS (2022), una desventaja de Amazon API Gateway es:

Latencia: El principal inconveniente es la latencia, ya que, en ciertos casos, el servicio puede agregar costosos milisegundos a sus tiempos de respuesta. Este problema se ve agravado por el hecho de que no puede ajustar API Gateway es un servicio administrado y AWS no permite modificar los parámetros de rendimiento.

2.3.4. AWS Glue

AWS Glue es un servicio en la Nube que prepara los datos para el análisis a través de procesos automatizados de extracción, transformación y carga (ETL). El servicio administrado es un método simple y rentable para categorizar y administrar big data en la empresa. Brinda a las organizaciones una herramienta de integración de datos que da formato a la información de fuentes de datos dispares y la organiza en un repositorio central, donde se puede utilizar para informar las decisiones comerciales (AWS, 2022).

2.3.4.1. Características de AWS Glue

Según AWS (2022), entre las características principales de Glue son las siguientes:

Descubrimiento automático de esquemas: Glue permite a los desarrolladores automatizar rastreadores para obtener información relacionada con el esquema y almacenarla en el catálogo de datos, que luego se puede usar para administrar trabajos.

Planificador de trabajos: Los trabajos de Glue pueden establecerse y llamarse según un calendario flexible, ya sea mediante activadores basados en eventos o bajo demanda. Pueden iniciarse varios trabajos en paralelo, y los usuarios pueden especificar las dependencias entre los trabajos.

Puntos finales del desarrollador: Los desarrolladores pueden usarlos para depurar Glue, así como crear lectores, escritores y transformaciones personalizados, que luego se pueden importar a bibliotecas personalizadas.

Generación automática de código: El proceso ETL genera código automáticamente, y la única entrada necesaria es una ubicación o ruta para almacenar los datos. El código está en Scala o Python.

Catálogo de datos integrado: Actúa como un almacén de metadatos singular de datos de una fuente dispar en la canalización de AWS. Una cuenta de AWS tiene un catálogo.

2.3.4.2. Ventajas del uso de Glue

Según AWS (2022), entre las ventajas de AWS Glue son los siguientes:

- **Tolerancia a fallos:** Los trabajos que fallan en Glue son recuperables, y los registros en Glue pueden ser depurados.
- **Filtrado:** Filtra los datos erróneos.
- **Soporte:** Soporta varias fuentes de datos no nativas de Java Database Connectivity (JDBC).
- **Mantenimiento y despliegue:** Mantenimiento y despliegue sencillos, ya que el servicio está completamente gestionado por AWS.

2.3.4.3. Desventajas del uso de Glue

Según AWS (2022), entre las desventajas de AWS Glue se presentan los siguientes:

- **Compatibilidad limitada:** Aunque AWS Glue funciona con una variedad de fuentes de datos de uso común, sólo funciona con servicios que se ejecutan en AWS. Las organizaciones pueden necesitar un servicio ETL de terceros si las fuentes no están basadas en AWS.
- **No hay sincronización de datos incremental:** Todos los datos se organizan primero en S3, por lo que Glue no es la mejor opción para trabajos ETL en tiempo real.
- **Curva de aprendizaje:** Los equipos que utilicen Glue deben tener un buen conocimiento de Apache Spark.
- **Consultas a bases de datos relacionales:** Glue tiene un soporte limitado para consultas de bases de datos relacionales tradicionales, sólo consultas SQL.

2.3.5. Amazon RDS (servicio de base de datos relacional)

Según AWS (2022), Amazon Relational Database Service (RDS) es un servicio de base de datos SQL administrado proporcionado por Amazon Web Services (AWS), el cual

admite una variedad de motores de base de datos para almacenar y organizar datos. También ayuda con las tareas de administración de bases de datos relacionales, como la migración de datos, la copia de seguridad, la recuperación y la aplicación de parches.

Adicionalmente, facilita la implementación y el mantenimiento de bases de datos relacionales en la Nube. Un administrador de la Nube utiliza Amazon RDS para configurar, operar, administrar y escalar una instancia relacional de una base de datos en la Nube. Por otra parte, Amazon RDS no es en sí mismo una base de datos, es un servicio utilizado para gestionar bases de datos relacionales (AWS, 2022).

2.3.5.1. Características de Amazon RDS

Según AWS (2022), entre las características de Amazon RDS incluyen lo siguiente:

Replicación: RDS usa la característica de replicación para crear réplicas de lectura. Estas son copias de solo lectura de las instancias de la base de datos que usan las aplicaciones sin alterar la base de datos de producción original. Los administradores también pueden habilitar la conmutación por error automática en varias zonas de disponibilidad a través de la implementación de RDS Multi-AZ y con la replicación de datos síncrona.

Almacenamiento: RDS proporciona tres tipos de almacenamiento:

- Unidad de estado sólido Solid State Drive (SSD) de propósito general. Amazon recomienda este almacenamiento como la opción predeterminada.
- Operaciones de entrada-salida aprovisionadas por segundo (IOPS). Almacenamiento SSD para cargas de trabajo intensivas de E/S.
- Magnético. Una opción de menor costo.

Vigilancia: El servicio de Amazon CloudWatch permite el monitoreo administrado, lo que proporciona a los usuarios ver la capacidad y las métricas de E/S.

Parcheo: RDS proporciona parches para cualquier motor de base de datos que elija el usuario.

Copias de seguridad: Otra característica es la detección y recuperación de fallas. RDS proporciona copias de seguridad de instancias administradas con registros de transacciones para permitir la recuperación en un momento dado. Los usuarios eligen un período de retención y restauran las bases de datos en cualquier momento durante ese

período. También pueden tomar instantáneas manualmente de las instancias que quedan hasta que se eliminen manualmente.

Facturación incremental: Los usuarios pagan una tarifa mensual por las instancias que lanzan.

Cifrado: RDS utiliza el cifrado de clave pública para proteger las copias de seguridad automatizadas, las réplicas de lectura, las instantáneas de datos y otros datos almacenados en reposo.

2.3.5.2. Ventajas de Amazon RDS

Según AWS (2022), entre las principales ventajas de Amazon RDS es que ayuda a las organizaciones a lidiar con la complejidad de administrar grandes bases de datos relacionales. Otros beneficios incluyen lo siguiente:

Facilidad de uso: Los administradores no necesitan aprender herramientas específicas de administración de bases de datos. También pueden administrar varias instancias de bases de datos mediante la consola de administración. RDS es compatible con los motores de bases de datos con los que los usuarios ya pueden estar familiarizados, como MySQL y Oracle, y automatiza los procesos manuales de copia de seguridad y recuperación.

Rentabilidad: Los clientes solo pagan por lo que usan. Además, se reduce el tiempo dedicado al mantenimiento de las instancias, ya que las tareas de mantenimiento, como las copias de seguridad y la aplicación de parches, están automatizadas.

Réplica: El uso de réplicas de lectura enruta el tráfico pesado de lectura lejos de la instancia de la base de datos principal, lo que reduce la carga de trabajo en esa única instancia.

Separación RDS: Divide el cómputo y el almacenamiento para que los administradores puedan escalarlos de forma independiente.

2.3.5.3. Desventajas de Amazon RDS

Según AWS (2022), entre algunas desventajas de usar Amazon RDS incluyen las siguientes:

Falta de acceso a la raíz: Debido a que es un servicio administrado, los usuarios no tienen acceso raíz al servidor que ejecuta RDS. Por lo tanto, restringe el acceso a ciertos procedimientos a aquellos con privilegios avanzados.

Falta del tiempo Los sistemas deben desconectarse para algunos procedimientos de aplicación de parches y escalado. El tiempo en estos procesos varía. Con el escalado, los recursos informáticos necesitan unos minutos de tiempo de inactividad en promedio.

2.3.6. AWS Lambda (Amazon Web Services Lambda),

Según AWS (2022), AWS Lambda es un servicio de computación en la Nube basado en eventos de AWS que permite a los desarrolladores programar funciones mediante el pago por uso sin tener que proporcionar almacenamiento o recursos informáticos para respaldarlas. Por otra parte, frecuentemente se denomina función como servicio o Function As a Service (FaaS).

2.3.6.1. Ventajas de AWS Lambda

Según AWS (2022), entre algunas de las ventajas se pueden mencionar:

- Uno de los principales beneficios de AWS Lambda es que abstrae la administración del servidor del profesional de TI. Con AWS Lambda, Amazon administra los servidores, lo que permite que un desarrollador se concentre más en escribir el código de la aplicación.
- AWS admite código escrito en una variedad de lenguajes de programación. Los lenguajes AWS Lambda incluyen Node.js, Python, Java y C#. Los desarrolladores también pueden usar herramientas de compilación de código, como Maven o Gradle, y paquetes para crear funciones.

2.3.6.2. Desventajas de AWS Lambda

Según AWS (2022), Lambda limita la cantidad de recursos informáticos y de almacenamiento que puede utilizar para ejecutar y almacenar funciones, estos límites se aplican por región y pueden aumentarse solicitándolo por medio de la consola del centro de soporte.

2.3.7. Data mart

Un data mart es un subconjunto de un CDW. Los data marts permiten a los usuarios recuperar información para departamentos o temas individuales, lo que mejora el tiempo de respuesta del usuario. Debido a que los data marts catalogan datos específicos, a menudo requieren menos espacio que los almacenes de datos empresariales, lo que los hace más fáciles de buscar y más económicos de ejecutar (AWS, 2022).

2.3.7.1. Tipos de Data marts

Según AWS (2022), existen tres tipos básicos de data marts:

Dependiente: Este ofrece centralización y permite obtener los datos de una organización desde un único almacén de datos. Existen dos métodos para crear un data mart dependiente: uno en el que los usuarios pueden acceder tanto al data mart como al data warehouse, y otro en el que el acceso de los usuarios está limitado solo al data mart. Este último método puede producir lo que comúnmente se conoce como depósito de chatarra de datos, ya que todos los datos comienzan con una fuente común, pero generalmente se descartan o desechan.

Independiente: se crea sin usar un almacén de datos central y es ideal para grupos más pequeños dentro de una empresa u organización. Los data marts independientes no tienen relación con el almacén de datos empresarial ni con ningún otro data mart. Los datos se ingresan desde una fuente de datos interna o externa, y sus análisis se realizan de forma autónoma. Debido a que los data marts independientes no funcionan ni interactúan con los almacenes de datos, los usuarios necesitan un almacenamiento coherente y centralizado de datos empresariales, como una base de datos relacional, a la que puedan acceder varios usuarios.

Híbrido: Esta combina la entrada de fuentes de datos que no forman parte del almacén de datos, como los datos operativos, y brinda a los usuarios una integración. Por esto, Los data marts híbridos requieren una limpieza de datos mínima, admiten grandes estructuras de almacenamiento y son flexibles. Los data marts híbridos son adecuados para entornos con múltiples bases de datos y organizaciones que requieren una respuesta rápida.

2.3.7.2. Ventajas de Data mart

Según AWS (2022), entre su desventaja se pueden mencionar:

Rapidez: Un data mart permite un acceso más rápido a los datos.

Facilidad: Es fácil de usar, ya que está diseñado específicamente para las necesidades de sus usuarios. Por lo tanto, un data mart puede acelerar los procesos comerciales.

Históricos: Contiene datos históricos que permiten al analista determinar las tendencias de los datos.

2.3.7.3. Desventajas de Data mart

Según AWS (2022), entre su desventajas se pueden mencionar

Soporte: Muchas veces las empresas crean demasiados data marts dispares y no relacionados sin mucho beneficio. Puede convertirse en un gran obstáculo para mantener.

Capacidad: Estos no puede proporcionar análisis de datos de toda la empresa, ya que su conjunto de datos es limitado.

2.3.8. Amazon Lightsail

Amazon Lightsail es un servicio en la Nube ofrecido por AWS que agrupa la memoria y el poder de cómputo para usuarios de la Nube. Por lo que, AWS empaqueta la memoria, el procesamiento, el almacenamiento y la transferencia en VM para que los clientes adquieran estos servicios, luego libera esa capacidad informática como instancia de Amazon Elastic Compute Cloud (EC2). Amazon Lightsail obtiene su poder de cómputo de una instancia EC2. Además, Amazon EC2 es un servicio web que proporciona capacidad informática segura y configurable en la Nube (AWS, 2022).

2.3.8.1. Ventajas de Amazon Lightsail

Según AWS (2022), entre su desventaja se pueden mencionar:

Costos: El precio es fijo, por lo que sabe de antemano cuánto le costará ejecutar la carga de trabajo que sea necesaria sobre la plataforma.

2.3.8.2. Desventajas de Amazon Lightsail

Según AWS (2022), entre su desventajas se pueden mencionar

Limitaciones: Es un poco más limitante que EC2. Por lo que, a diferencia de cuando se crea una instancia EC2, por ejemplo, hay 36 imágenes diferentes que puede elegir y en Lightsail no es dispone de muchas opciones.

2.3.9. Microsoft Power BI, herramienta de apoyo a la inteligencia empresarial

En la actualidad las organizaciones están optando por la inteligencia de negocios como una manera de emplear los datos en la toma de decisiones, para ello, se usan herramientas no solo para procesar la data sino también para generar representaciones sólidas de tendencias de los datos. Power BI forma parte de este grupo de alternativas que en la actualidad están sustentado el *Business Intelligence* (Microsoft, 2022). Seguidamente, en la Figura 3 se presenta la pantalla inicial de la aplicación Power BI.

Power BI es una aplicación generada por la empresa Microsoft y entre sus funcionalidades está la de facilitar la integración de datos y el análisis de estos. Según Microsoft (2022), este software presenta como características:

- Permite conectar simultáneamente con más de cien orígenes de datos.
- Facilita Integración controles y API.
- Dispone de variedad de objetos visuales para inteligencia artificial.
- Generar informes de las consultas.

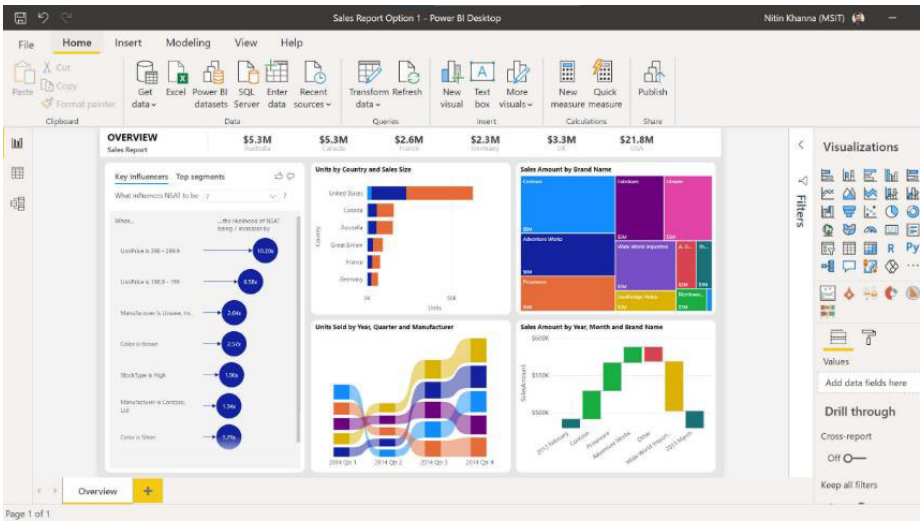


Figura 3. Pantalla de la aplicación Power BI

Fuente: tomado de Microsoft (2022).

2.3.9.1. Ventajas de Microsoft Power BI

Según Microsoft (2022), entre sus ventajas se pueden mencionar:

Publicación segura de informes: Se puede automatizar la actualización de datos de configuración y publicar informes que permitan a todos los usuarios aprovechar la información más reciente.

Fácil de usar: Power BI es fácil de usar y los usuarios pueden encontrarlo fácilmente las opciones disponibles en cada uno de los paneles y reportes.

Innovación constante: El producto Power BI se actualiza todos los meses con nuevas funciones y características.

2.3.9.1. Desventajas de Microsoft Power BI

Según Microsoft (2022), entre sus desventaja se pueden mencionar:

Restricciones: Los paneles e informes solo se comparten con los usuarios que tienen los mismos dominios de correo electrónico.

No comparte: dashboard nunca acepta ni pasa parámetros de usuario, cuenta o cualquier otra entidad.

Limita fuentes: Muy pocas fuentes de datos permiten conexiones en tiempo real a los informes y paneles de Power BI.

2.3.10. Node.Js

Node.js (Node), es una plataforma de desarrollo de código abierto para ejecutar código JavaScript en el lado del servidor y es útil para desarrollar aplicaciones que requieren una conexión persistente desde el navegador al servidor, también se usa para aplicaciones en tiempo real, como chat, fuentes de noticias y notificaciones de alertas que se muestran en el escritorio. Por lo que, este está diseñado para ejecutarse en un servidor HTTP dedicado y emplear un solo hilo con un proceso a la vez. Las aplicaciones de Node.js se basan en eventos y se ejecutan de forma asincrónica. El código creado en la plataforma Node no sigue el modelo tradicional de recibir, procesar, enviar, esperar. En cambio, Node procesa las solicitudes entrantes en una pila de eventos constante y envía pequeñas solicitudes una tras otra sin esperar respuestas (Nodejs, 2022).

2.3.10.1. Ventajas de Node.Js

Según Microsoft (2022), entre sus ventajas se pueden mencionar:

Rendimiento: Alto rendimiento para aplicaciones en tiempo real.

Escalabilidad: Fácil escalabilidad para aplicaciones modernas.

Entendimiento: Fácil de aprender y rápido de adaptar.

Respuesta: Mejora el tiempo de respuesta de la aplicación y aumenta el rendimiento.

2.3.10.2. Desventajas de Node.Js

Según Microsoft (2022), entre sus desventajas se pueden mencionar:

Consume recursos: Reduce el rendimiento cuando se manejan tareas informáticas pesadas.

Soporte: El modelo de programación asincrónica de Node.js dificulta el mantenimiento del código.

2.3.11. FileZilla

FileZilla es una herramienta de software de protocolo de transferencia de archivos o File Transfer Protocol (FTP) de código abierto y gratuita que permite a los usuarios configurar o conectarse a otros servidores FTP para intercambiar archivos. FileZilla admitía tradicionalmente el Protocolo de transferencia de archivos sobre seguridad de la capa de transporte. El software cliente de FileZilla está disponible para todas las plataformas de forma gratuita (FileZilla, 2022).

2.3.11.1. Ventajas de FileZilla

Según FileZilla (2022), entre sus ventajas se pueden mencionar:

Fácil: En general, la facilidad de uso y las opciones adicionales en la transferencia de datos es muy conveniente.

Conexión: La conexión a sitios FTP fácil de operar, así como la capacidad de visualización de los procesos.

2.3.11.2. Desventajas de FileZilla

Según FileZilla (2022), entre sus desventajas se pueden mencionar:

Actualizaciones: FileZilla requiere actualizaciones frecuentes. En algunos momentos puede detener el trabajo si no está actualizado.

Capacidad: En algunos casos el tamaño de los archivos puede representar un problema.

2.3.12. Visual Studio (VS) Code

Visual Studio (VS), Code es un editor de programación de código abierto que se utiliza principalmente para corregir y reparar errores de codificación de aplicaciones web y en la Nube. VS Code está desarrollado por Microsoft y es compatible con los sistemas operativos macOS, Linux y Windows. Por otra parte, las herramientas de VS Code se pueden utilizar para mejorar la funcionalidad de cualquier código escrito y basado en el marco Electron, VS Code emplea el mismo componente de edición que se emplea en Azure DevOps. Al incorporar múltiples extensiones de FTP, los usuarios pueden sincronizar el código entre el servidor y el editor sin tener que descargar software adicional (VSC, 2022).

2.3.12.1. Ventajas de Visual Studio (VS) Code

Según VSC (2022), entre sus ventajas se pueden mencionar:

- Útil para depurar el código de forma rápida y eficiente.
- Liviano y rápido: nunca presenta fallas al cambiar de archivo y escribir el código fuente.
- Admite JavaScript, TypeScript, HTML, CSS, JSON, entre otros.

2.3.12.2. Desventajas de Visual Studio (VS) Code

Según VSC (2022), entre sus desventajas se pueden mencionar:

- A veces arroja errores por razones inapropiadas y puede confundir a los desarrolladores.

- Por estar diseñado principalmente para el desarrollo web, sólo permite abrir archivos con la extensión .java pero no proporciona mucho poder para un proyecto java.

2.3.13. Postman

Postman es una herramienta de desarrollo de API que ayuda a crear, probar y modificar API y cualquier funcionalidad que pueda necesitar desarrollador está encapsulada en esta herramienta (Postman, 2022).

2.3.13.1. Ventajas de Postman

Según Postman (2022), entre sus ventajas se pueden mencionar:

- **Facilidad de uso:** Es una herramienta fácil de manejar y probar soluciones API de forma rápida.
- **Prueba de procesos:** Su interfaz es sencilla de utilizar para probar procesos que son activados por solicitudes http.

2.3.13.2. Desventajas de Postman

Según Postman (2022), entre sus desventajas se pueden mencionar:

- **Dificultad para iniciar:** se debe tener un conjunto específico de habilidades técnicas para trabajar con Postman y no existen mucha información como tutoriales o material didáctico en internet.

2.3.14. pgAdmin

pgAdmin es una plataforma de desarrollo y administración de código abierto con funciones para PostgreSQL y está diseñado para monitorear y administrar múltiples servidores de bases de datos, tanto locales como remotos, a través de una sola interfaz gráfica que permite la fácil creación y administración de objetos de BD, así como una serie de otras herramientas para para facilitar la administración de datos (pgAdmin, 2022).

2.3.14.1. Ventajas de pgAdmin

Según pgAdmin (2022), entre sus ventajas se pueden mencionar:

- PostgreSQL puede ejecutar sitios web dinámicos y aplicaciones web como una opción de pila.
- Al usar pgAdmin, se puede crear, ver y editar todos los objetos comunes de PostgreSQL.
- El tablero de pgAdmin permite a los usuarios monitorear las actividades del servidor, como bloqueos de bases de datos, sesiones conectadas del servidor y transacciones actuales del servidor.

2.3.14.2. Desventajas de pgAdmin

Según pgAdmin (2022), entre sus desventajas se pueden mencionar:

- En comparación con las herramientas GUI de pago, la GUI de pgAdmin es lenta y no es intuitiva.
- Los recursos utilizados por pgAdmin muy elevados.

2.4. Metodología para el desarrollo del CDW

A continuación, se presenta la descripción la Metodología CRISP-DM y MSF respectivamente, con el propósito de desarrollar de forma eficiente la contextualización de un almacén de datos tipo CDW, así como definir los elementos básicos vinculados sobre el establecimiento de la arquitectura en la Nube.

2.4.1. Metodología CRISP-DM en la construcción de un Cloud Data Warehouse.

La metodología CRISP-DM está sustentada en estándares internacionales que reflejan la robustez de sus procesos y que facilitan la unificación de sus fases en una estructura confiable y amigable para el usuario. Además de ello, esta tecnología interrelaciona las diferentes fases del proceso entre sí, de tal manera que se consolida un proceso iterativo y recíproco (Mejías, 2018).

Según Mejías (2018), en el desarrollo de una aplicación cuyo objetivo es ayudar a los líderes de la empresa a la toma de decisiones, es de suma importancia manejar y transformar los datos con los que se va a trabajar, de manera que sean útiles. CRISP-DM es un modelo de referencia que provee un ciclo de vida para un proyecto de CDW. Es así como, el ciclo de vida del proyecto según la metodología CRISP-DM está basado en seis fases que se describen a continuación:

- **Comprensión del negocio:** Esta fase es la más importante porque reúne las tareas de comprensión de los objetivos y requisitos del proyecto desde una perspectiva empresarial o institucional, con el fin de convertirlos en objetivos técnicos y en un plan de proyecto. Es importante comprender con claridad los objetivos y el problema del proyecto.
- **Comprensión de los datos:** Esta fase, junto a las dos próximas, son las que demandan el mayor esfuerzo y tiempo en un proyecto de CDW porque comprende la recolección inicial de datos, con el objetivo de establecer un primer contacto con el problema, familiarizándose con ellos, identificar su calidad y establecer las relaciones más evidentes que permitan definir las primeras hipótesis.
- **Preparación de los datos:** Esta fase comprende tareas como: selección de datos, limpieza de datos, generación de variables adicionales, integración de diferentes orígenes de datos y cambios de formato.
- **Modelado:** Esta fase selecciona las técnicas de modelado más apropiadas para el proyecto de CDW, dicha técnica se elige de acuerdo los siguientes criterios: ser apropiada para el problema, disponer de datos adecuados, cumplir los requisitos del problema, tiempo adecuado para obtener un modelo y conocimiento de la técnica.
- **Evaluación:** En esta fase se evalúa el modelo, teniendo en cuenta el cumplimiento de los criterios de éxito del problema. Debe considerarse, además, que la fiabilidad calculada para el modelo se aplica solamente para los datos sobre los que se realizó el análisis.
- **Despliegue o divulgación:** Esta fase surge una vez que se ha construido y validado el modelo. Generalmente, un proyecto de CDW no concluye en la implantación del modelo, pues se debe presentar los resultados de manera comprensible para los líderes de la empresa, con el objetivo de lograr un incremento del conocimiento y ayudar en la toma de decisiones.

Seguidamente, en la Figura 4 se presenta el modelo de procesos CRISP-DM de minería de datos:

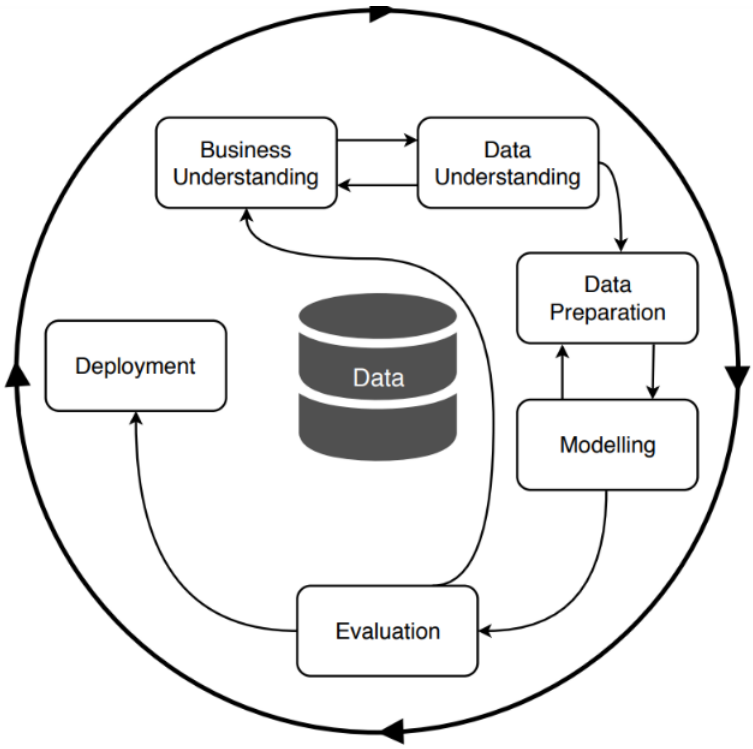


Figura 4. El modelo de proceso CRISP-DM de minería de datos

Fuente: tomado de Martínez et al. (2021)

2.4.2. Metodología MSF para la implementación de la infraestructura necesaria en la Nube para la construcción de un Cloud Data Warehouse.

En relación al modelo de procesos *Microsoft Solution Framework* (MSF), está definido por la combinación de dos modelos de procesos que son muy empleados en proyectos de desarrollo, como lo son cascada y espiral. Es por ello, que se cumplirán con las 5 fases propuestas por esta metodología para la implementación de la infraestructura en la Nube. Las fases a desarrollar se describen a continuación:

Visión: Se definirán los requerimientos de infraestructura de la empresa.

Planeación: Se creará un borrador del plan maestro del proyecto para la implementación de la infraestructura necesaria.

Desarrollo: Se realizarán implementaciones internas de la infraestructura.

Estabilización: Se probará la infraestructura aplicada.

Implantación: Se obtendrá la aprobación del cliente final sobre la infraestructura implementada.

3. DESARROLLO DEL PROYECTO

En esta sección se presenta el desarrollo de la Metodología CRISP-DM y MSF respectivamente. Estas metodologías permitieron la puesta en marcha del proyecto de forma eficiente y la organización de cada fase del diseño del almacén de datos tipo CDW, así como definir los elementos básicos vinculados sobre el establecimiento de la arquitectura en la Nube. A continuación, se presentan cada una de las fases de la metodología CRISP-DM junto con las actividades desarrolladas en cada fase.

3.1. Comprensión del negocio

La empresa Bikes Extreme Ecuador es una comercializadora de bicicletas y repuestos. Además, ofrece servicios a sus clientes en reparación y mantenimiento, su principal negocio lo conforman las ventas de bicicletas de ruta y las ventas de bicicletas de montaña o *Mountain Bike* (MTB). Adicionalmente, esta vende accesorios para el uso de los equipos tanto de montaña como de ruta (Bike Extreme, 2022).

Establecimiento de los objetivos

- Mejorar la eficiencia en las ventas y la calidad de prestación de los servicios.
- Establecer políticas que faciliten la satisfacción de los clientes.
- Integrar sistemas computarizados para la mejora continua de la empresa.
- Establecer un sistema de toma de decisiones entorno a la adquisición de productos nuevos y productos existentes.

3.2. Comprensión de los datos

La empresa Bikes Extreme Ecuador, maneja un conjunto de datos relacionados con su actividad de comercialización, específicamente relacionado con ventas, productos, clientes y proveedores. Para la estructuración del proyecto de conformación de CDW fue necesario tener una compresión de estos datos, para de este modo poder vincularlos y establecer una estructura lógica de estos. Se inició con recopilación de estos y su posterior descripción.

3.2.1. Recopilación de los datos

Se recolectaron los datos de la empresa del último trimestre para el diseño de la estructura de la base de datos. Los datos se recopilaron en el siguiente formato como se presenta en la Figura 5.

Figura 5. Datos de la empresa.

Fuente: Elaboración Propia

3.2.2. Descripción de los datos

Los datos de la empresa Bikes Extreme Ecuador, se presentan en cuatro categorías principales:

- **Products:** Contiene la información de todos los productos que se comercializan dentro de la empresa.
- **Sales:** Contiene el registro de todas las ventas de los productos.
- **Suppliers:** Contiene la información de todos los proveedores que la empresa utiliza.
- **Clients:** Contiene registros con información de todos los clientes que maneja la empresa. Entre estos se pueden mencionar el registro único de contribuyente (RUC), nombre y apellido del cliente, correo, teléfono, entre otros

3.2.2.1. Estructura categoría Products

En esta se registraron los productos que la empresa comercializa.

- **idProduct:** Llave primaria.
- **idSupplier:** Código del proveedor.
- **Name:** Nombre del producto.
- **Price:** Precio del producto.
- **Stock:** Cantidad en existencia.

3.2.2.2. Estructura categoría Sales

En esta se registran las transacciones de las ventas de la empresa.

- **Idsale:** Llave primaria.
- **Idproduct:** Llave foránea.
- **Idclient:** Llave foránea.
- **Quantity:** Cantidad de productos.
- **Unitprice:** Precio por unidad
- **Total:** Total de la transección
- **Date:** Fecha de la venta

3.2.2.3. Estructura categoría Suppliers

En esta se registraron los datos de los proveedores de la empresa.

- **Idsupplier:** Llave primaria.
- **Ruc:** Registro Único de Contribuyentes (RUC)
- **Name:** Nombre del proveedor.
- **Address:** Dirección del proveedor.
- **Email:** Correo electrónico del proveedor.
- **Phone:** Número de teléfono del proveedor.

3.2.2.4. Estructura categoría clientes

En esta se registraron los datos de los clientes de la empresa

- **Idclient:** Llave primaria.
- **Ruc:** Registro Único de Contribuyentes (RUC)
- **Firstname:** Nombre del cliente
- **Lastname:** Apellido del cliente
- **Email:** Correo electrónico del cliente.
- **Phone:** Número de teléfono del cliente.

3.3. Preparación de los datos

Esta fase comprende tareas como la selección de datos de interés para el negocio y para el CDW. Estos datos se relacionan con proveedores, clientes, ventas y productos. El proceso de preparación involucra la actividad de limpieza de la data. Sin embargo, la data de la empresa Bikes Extreme Ecuador no requirió mayores adecuaciones, ya que esta estaba depurada y presentaba una estructura y formato homologado, que facilitó el proceso de

preparación de los datos. Para los datos de fecha se trabajó con un formato normalizado como el siguiente; MM/DD/AAAA.

Por otra parte, los datos de origen de las Sucursales 1 y 2 como los productos están agrupados por factura, luego estos cargados a la base de datos de integración. Estos productos por factura fueron desglosados producto por producto y se insertaron en un registro por cada producto con un identificador de la sucursal de procedencia, lo que permite realizar de forma adecuada los análisis estadísticos a través de Power BI.

3.4. Modelado

Esta fase selecciona las técnicas de modelado más apropiadas para el proyecto de CDW, dicha técnica se elige de acuerdo los siguientes criterios: ser apropiada para el problema, disponer de datos adecuados, cumplir los requisitos del problema, tiempo adecuado para obtener un modelo y conocimiento de la técnica.

Esta fase de modelado se realizó utilizando la metodología MSF explicada en la sección 2.4.2. A continuación, se listan cada una de las fases de esta metodología y se detallan las actividades realizadas.

3.4.1. Visión

La empresa Bikes Extreme Ecuador buscar la implementación de un sistema de CDW en un ambiente en la Nube, para poder almacenar datos consolidados de variadas fuentes, ya que actualmente dispone de dos sucursales con alto volumen de ventas e inventarios. Es necesario. Además, una integración de toda la información que manejan dichas sucursales, de forma oportuna y extraer información que sirva de soporte a la toma de decisiones en el contexto logístico, así como de clientes, proveedores y ventas. Además, la información que surja de los datos facilitará contar con un ambiente fácil de escalar en recursos, elástico y seguro

3.4.2. Planeación

Seguidamente se presenta la estructura de las Nubes:

- Nube 1: DynamoDB, base de datos no relacional de AWS. Con las tablas Clients, Suppliers, Products, Sales1.
- Nube 2: RDS, base de datos relacional de AWS. Con la tabla Sales2.
- Nube 3: en esta se dispone de los siguientes elementos:

- VPS cloud de AWS, con una API REST que contendrá el ETL y las rutas para consultar la base de integración desde Power BI.
- RDS, con las tablas sales_dm y purchases_dm.
- VPS cloud de AWS, con Power BI.

Se utilizan tres Nubes en el proyecto, ya que la Sucursal 1 llamada cld1 (emplea una base de datos no relacional), sucursal 2 llamada cld2 (emplea una base de datos relacional), pero el CDW está en la capacidad de poder interactuar con ambas estructuras. Además, también fueron separadas en tres Nubes para poder utilizar la capa gratuita de Amazon AWS. Sin embargo, el mismo proyecto puedo implementarse utilizando una sola suscripción. A continuación, en la Figura 6 se presenta el esquema de las Nubes que presenta el desarrollo.

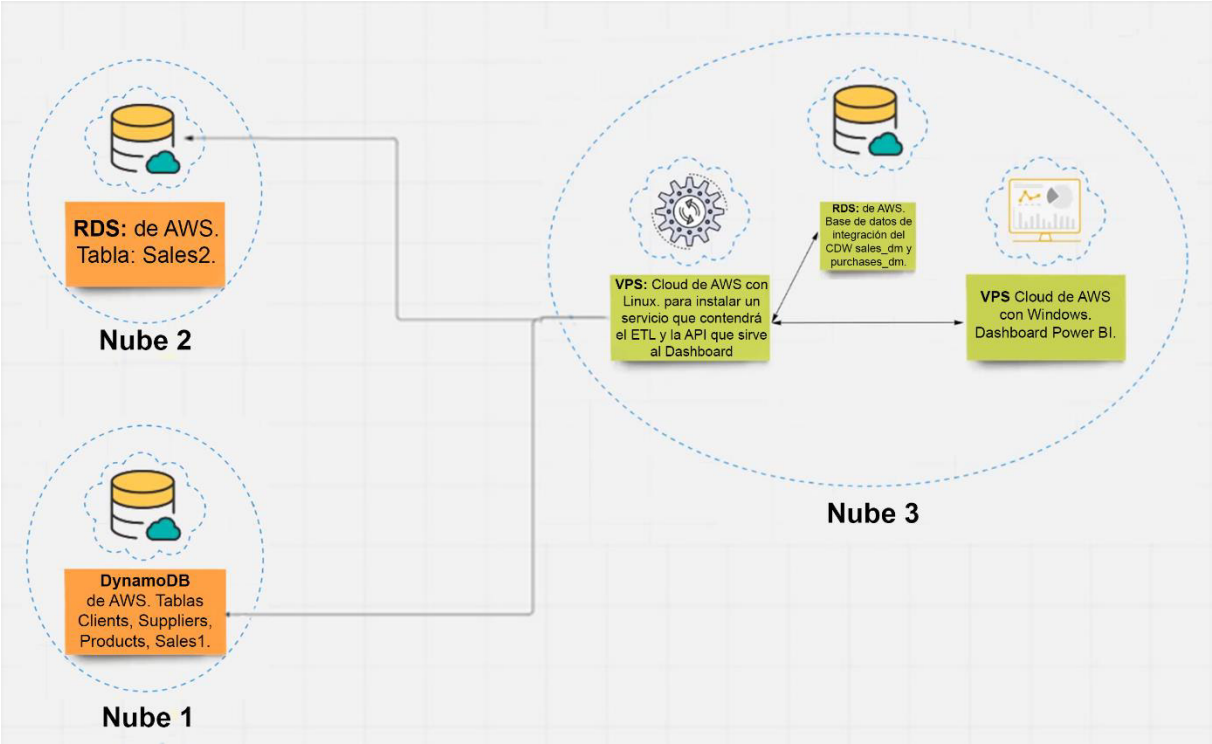


Figura 6. Esquema de las Nubes propuestas.

Fuente: Elaboración propia.

La Nube 1: Lo conforma la base de datos DynamoDB de AWS que contiene cuatro tablas: Clients, suppliers, products y sales_cld1, esta es alimentada con información de la sucursal N°1 de la empresa, también está asociada a una primera cuenta de AWS, en la Figura 7 se presenta el esquema de la Nube indicada anteriormente.

Por otra parte, Amazon DynamoDB, la cual es una base de datos NoSql, para que esta pueda ser manipulada desde el exterior requiere de implementaciones como lo son las funciones

Lambda. Estas están diseñadas con Java Scribd en el framework Node.Js, donde cada función fue creada para cada una de las tablas de DynamoDB con el propósito de interactuar con la base de datos realizado inserciones, lecturas de los datos o filtrado, en la Figura 8 se muestra la representación de los elementos internos de la Nube 1. Adicionalmente, Se creó un API Gateway que funciona como una puerta de entrada desde el exterior, cuyo propósito es poder comunicarse con las funciones Lambda, donde el ETL se comunica por medio de rutas de protocolo HTTP por medio del método GET y POST.

Figura 7. Nube 1.

Fuente: Elaboración propia.

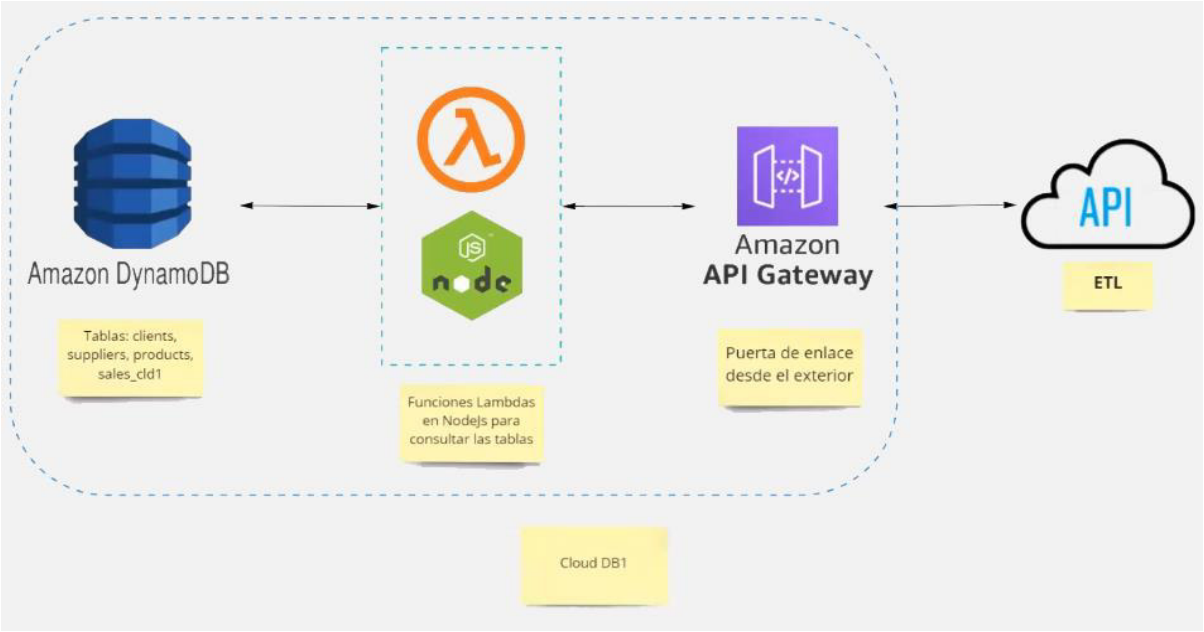


Figura 8. Elementos de la Nube 1.

Fuente: Elaboración propia.

La Nube 2: Está constituida por una base de datos RDS conformada por una tabla relacional llamada Sales_cld2. Es de relevancia destacar que se persigue que el CDW pueda leer datos de diferentes fuentes, así como datos relacionados o no relacionados. Esta Nube representa la sucursal número 2 de la empresa y contiene datos asociados con las ventas. En la Figura 9 se presenta la representación de la Nube 2.

Figura 9. Nube 2.

Fuente: Elaboración propia.

La Nube 3: Lo conforma la Nube VPS donde se encuentra el ETL que fue diseñado de manera que realice un barrido diario en las bases de datos de origen y actualice con los datos nuevos que se han generado dentro de las tablas de ventas. Este realiza toda la integración de las bases de datos de origen a la de integración del CDW y también alberga la Nube del Power BI. Por otra parte, la base de datos que se crea dentro de esta Nube es una base RDS llamada Integración del CDW. En la Figura 10 se presenta gráficamente la Nube 3.

El ETL presente en esta nube verifica las ventas de las dos sucursales del día en curso para actualizar la base de datos integración. También realiza un proceso de chequeo por medio de dos Data mart, donde para la Nube 2 de ventas las integra dentro de la tabla integración y para la Nube 1 en la tabla productos. Adicionalmente, verifica el stock mínimo de cada producto y genera una tabla resumen con esa información. En la Figura 11, se presenta el código empleado en el Data mart.

Esta Nube, también, contiene la base de datos integración, que es donde se centraliza toda la información que se genera por medio de los Data mart y el ETL. En la Figura 10 se puede observar dicha Nube.

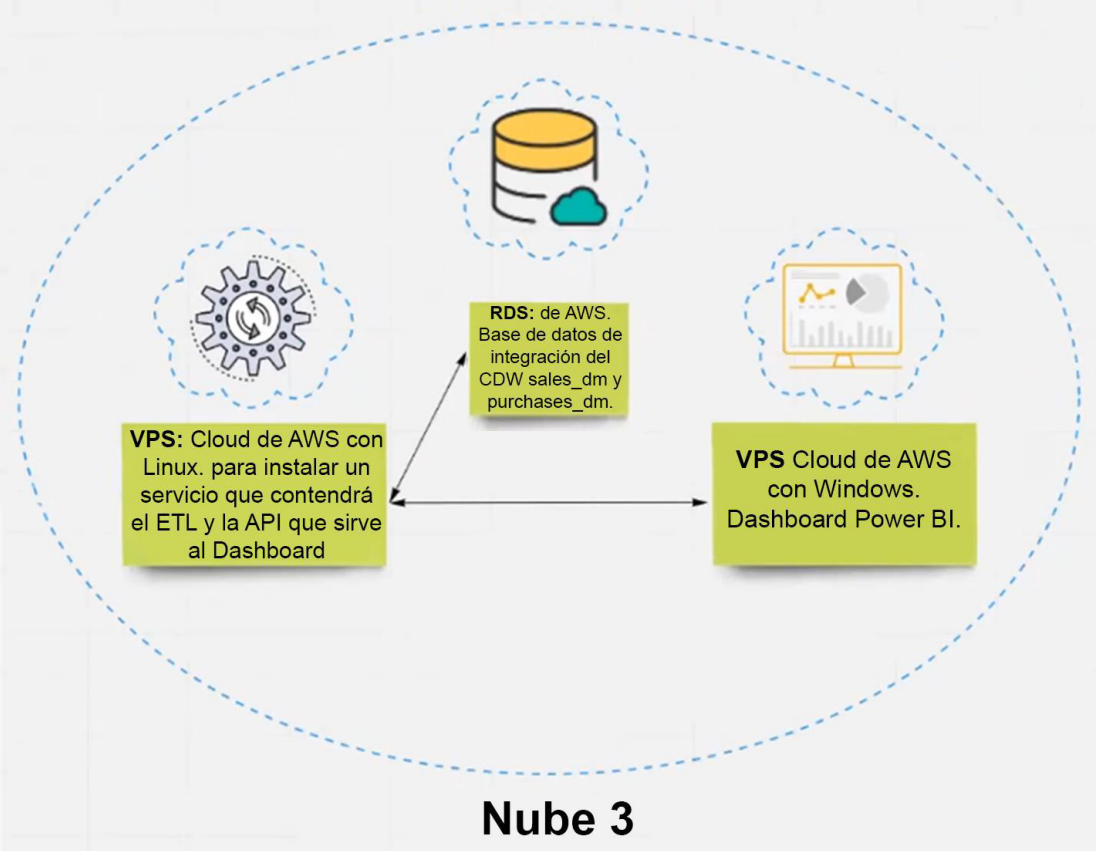


Figura 10. Nube 3.

Fuente: Elaboración propia.

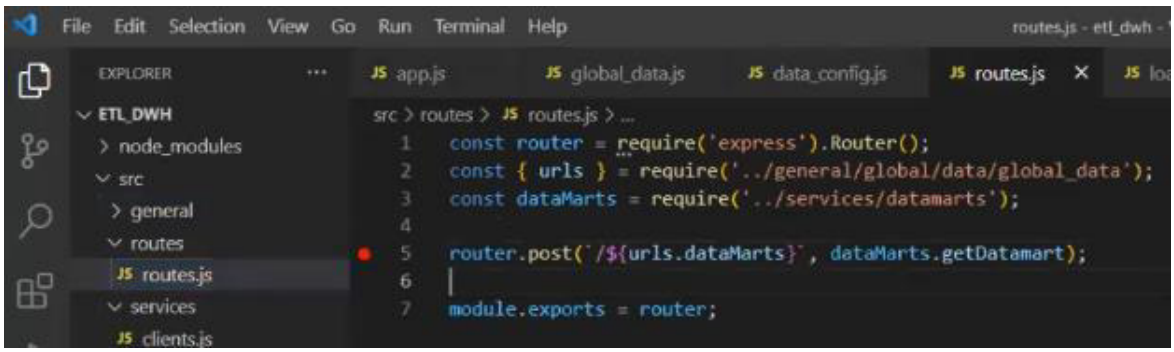


Figura 11. Data mart.

Fuente: Elaboración propia.

3.4.3. Desarrollo

A continuación, se explica la construcción de las 3 Nubes indicadas en la Figura 6.

3.4.3.1. Construcción de la Nube 1

Creación de las tablas

En este paso se desarrolló la creación de las tablas en "*DynamoDB*" de AWS, donde estuvieron las "*Tablas*" (Clients, Suppliers, Products). La tabla de ventas se encuentra en una DB relacional (RDS) también de AWS. Por otra parte, las "*Funciones Lambdas*" de cada tabla tendrán el código para realizar las consultas sobre las mismas. El "*API Gateway*" permitirá consultar desde el exterior las tablas a través de "*Funciones Lambdas*". El "*ETL*" se conectó por medio del "*API Gateway*". A continuación, en la Figura 12 se presenta el diagrama de las tablas en AWS DynamoDB, sus Funciones Lambdas y la API de consulta.

Figura 12. Diagrama de las tablas en AWS DynamoDB, sus Funciones Lambdas y la API de consulta.

Fuente: Elaboración propia.

A continuación, se presentan los pasos para la creación de las tablas.

1. Iniciar sesión en la consola de AWS como se presenta en la Figura 13.
2. Ingresar a DynamoDB como se observa en la Figura 14 para crear las tablas.

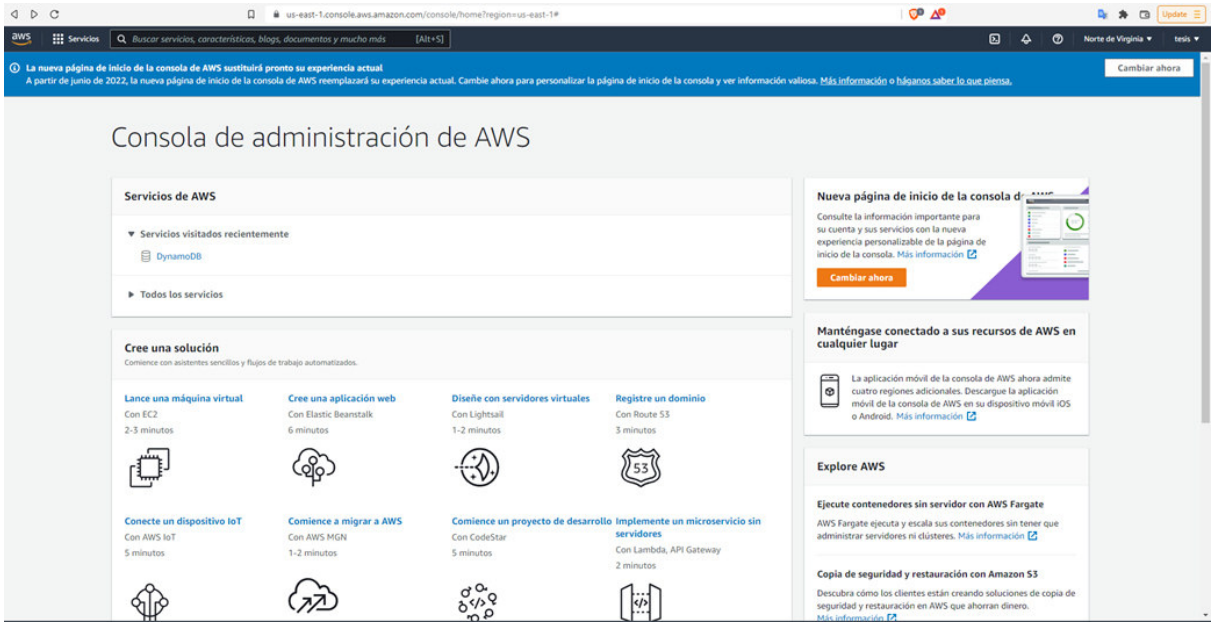


Figura 13. Inicio de sesión en la consola de AWS.

Fuente: Elaboración propia.

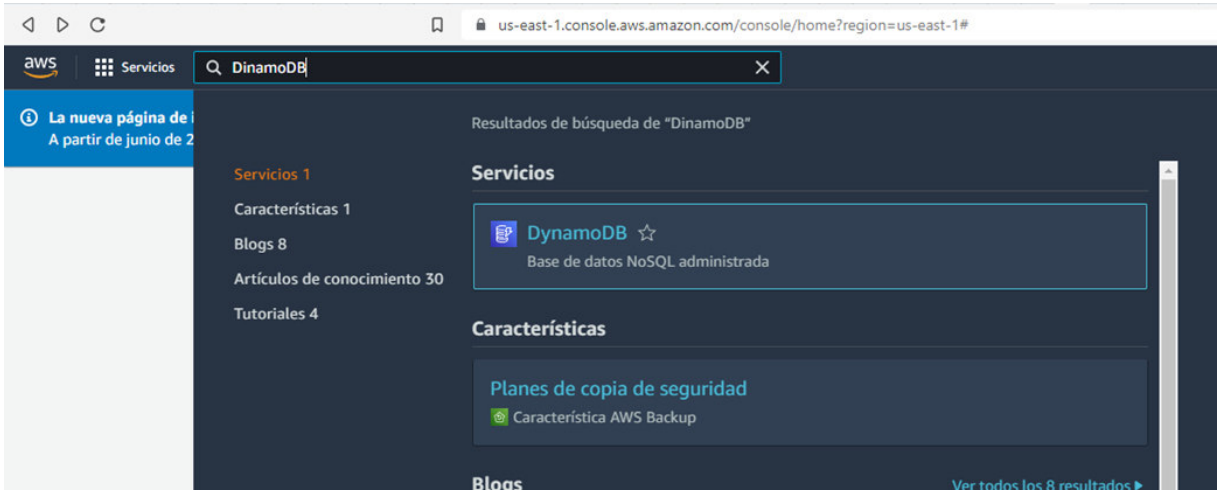


Figura 14. Ingreso al servicio DynamoDB.

Fuente: Elaboración propia.

3. Una vez ingresado dentro de DynamoDB se inició con la creación de cada una de las tablas. Como se observa en la Figura 15, se presenta la pantalla de bienvenida del servicio DynamoDB y luego el ingreso a la pantalla para crear las tablas como se

describe en la Figura 16. En la Figura 17 se presenta la consola con las tablas generadas dentro del servicio.

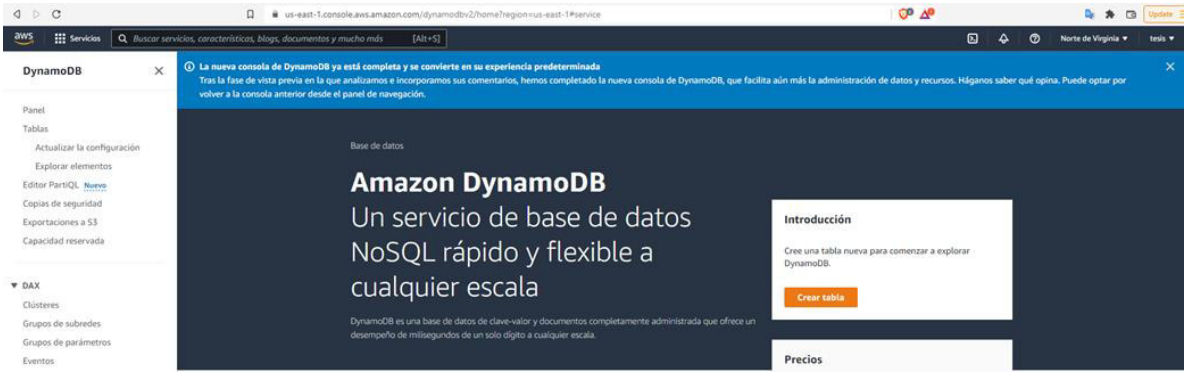


Figura 15. Servicio de base de datos de Amazon DynamoDB.

Fuente: Elaboración propia.

Figura 16. Servicio crear tabla de DynamoDB.

Fuente: Elaboración propia.

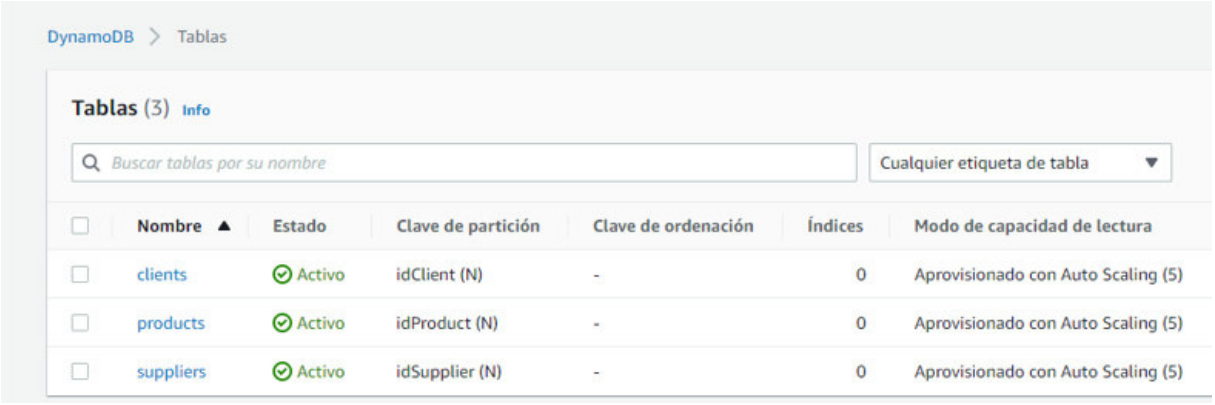


Figura 17. Consola de las tablas de DynamoDB.

Fuente: Elaboración propia.

Creación de las Funciones Lambdas

Dentro del proyecto se crearon 2 lambdas para cada una de las tablas. Lo anterior se debe a que este tipo de funciones se utilizan para un rol específico, debido a que solo presenta dominio dentro de las variables y de su rango, Amazon presta la función como servicio en la cual se puede colocar una función Lambda y esta debe realizar un solo proceso. Es por ello por lo que, se tienen dos funciones, una para leer y otra para escribir. Además, se crean dos funciones Lambda adicionales, ya que se requiere leer de una tabla filtrando las ventas por producto por fecha y la otra función trae todos los datos con bajo stock.

A continuación, se muestra la creación de las funciones Lambdas para poder realizar las consultas de las tablas.

1. Ingresar al panel de Lambdas como se presenta en la Figura 18.
2. Crear la función como se presenta en la Figura 19.

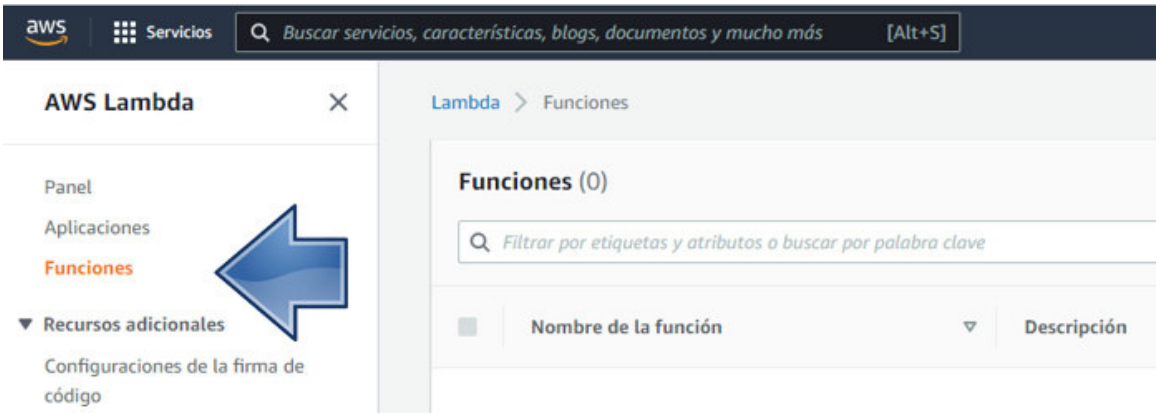


Figura 18. Panel Lambdas de DynamoDB.

Fuente: Elaboración propia.

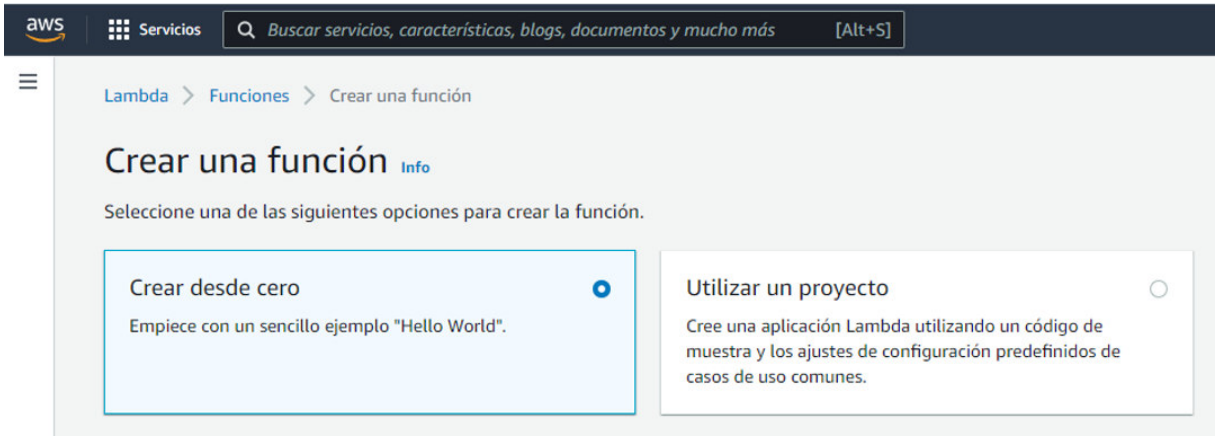


Figura 19. Crear función en Lambdas de DynamoDB.

Fuente: Elaboración propia.

3. En esta pantalla se debe seleccionar la opción "*Crear desde cero*" para crear la función como se presenta en la Figura 20.
4. Ingresar el fichero "*index.js*" con la codificación necesaria para generar la función de código de consulta como se presenta en la Figura 21.

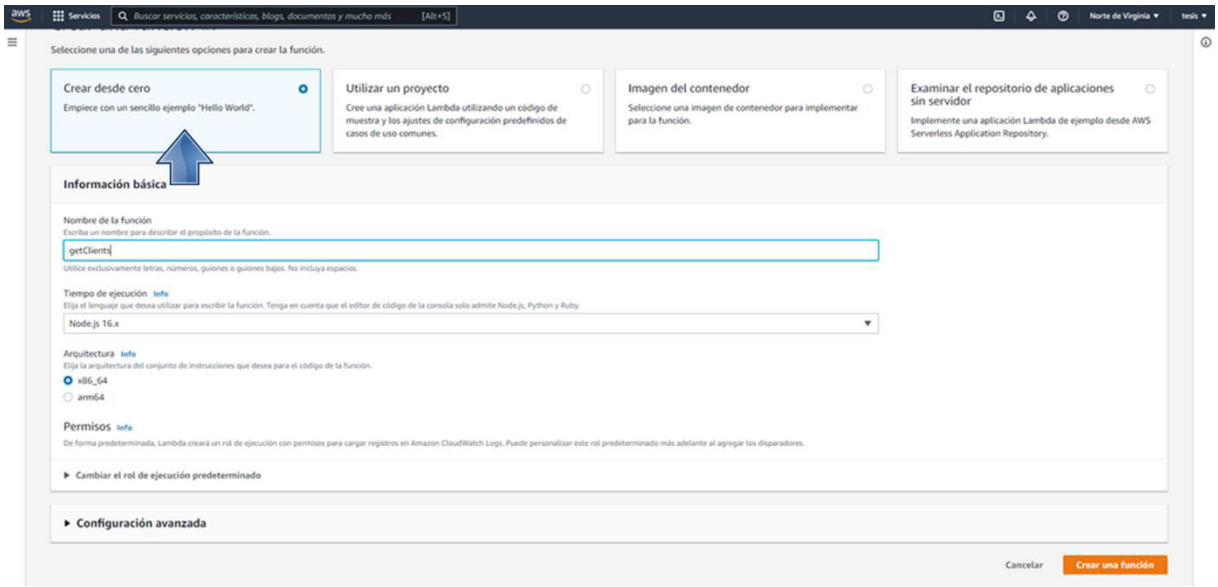


Figura 20. Crear desde cero en Lambdas de DynamoDB.

Fuente: Elaboración propia.

```
exports.handler = async (event) => {
    let statusCode = 400;
    let correct = false;
    let data = [];
    let info;
    let body;

    body = JSON.stringify({
        correct,
        data,
        info
    });

    return {
        statusCode,
        body,
    };
};
```

Figura 21. Código de la consulta.

Fuente: Elaboración propia.

5. Una vez diseñado el código y copiado en el archivo, este se carga en Lambdas de "*DynamoDB*" como se observa en la Figura 22.
6. Dar permisos a la Lambda en el panel IAM como se observa en la Figura 23.

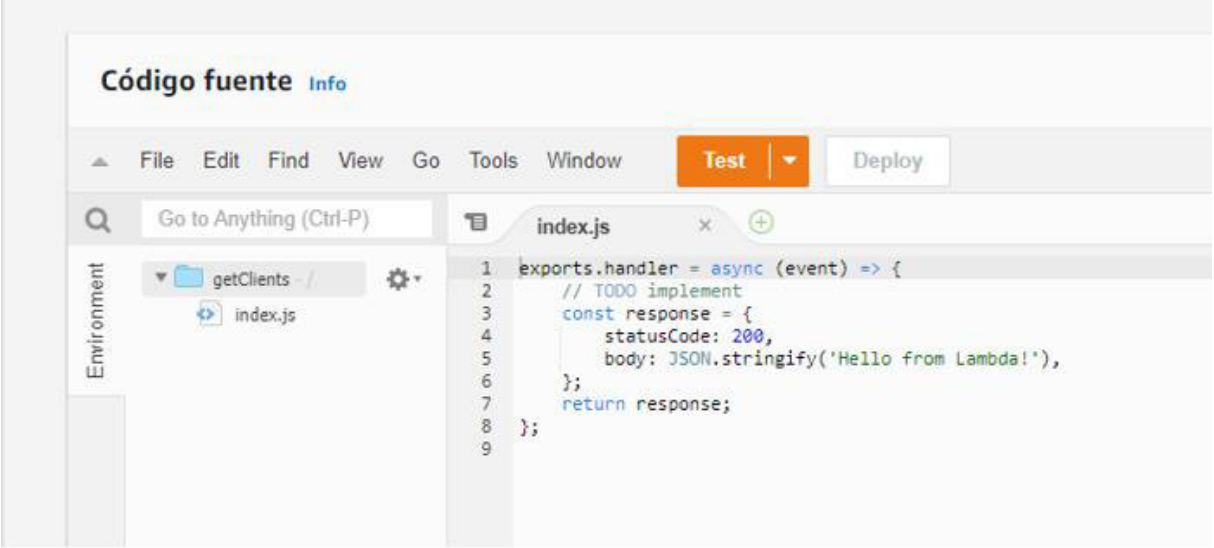


Figura 22. Código fuente en Lambdas de DynamoDB.

Fuente: Elaboración propia.

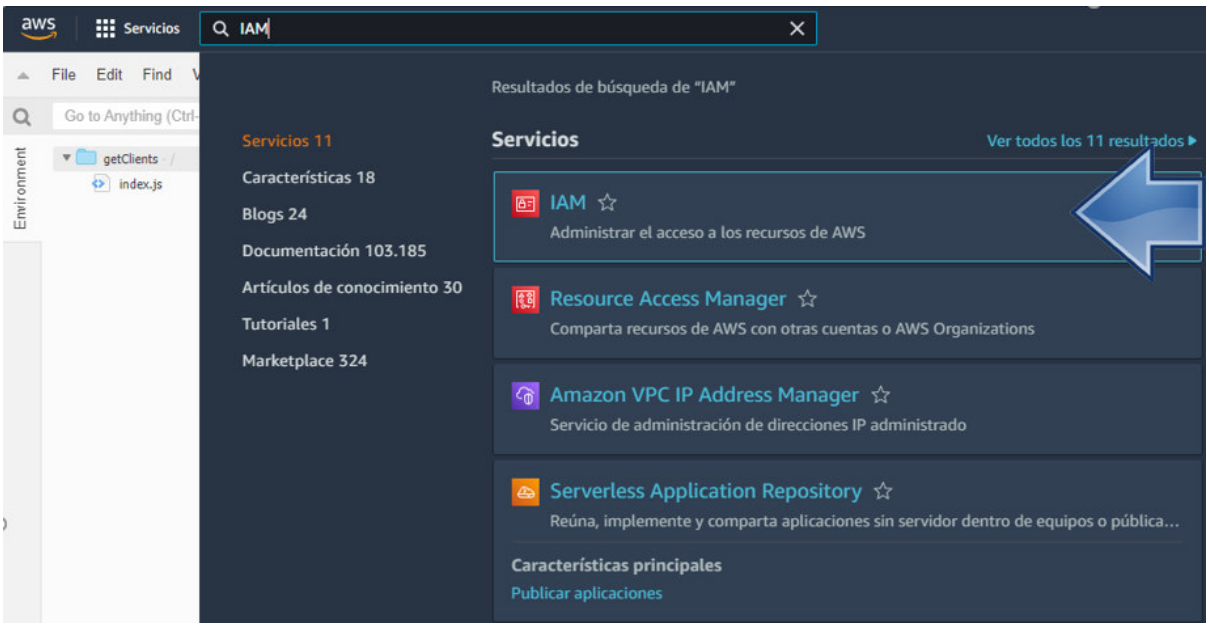


Figura 23. Panel IAM de Lambdas de DynamoDB.

Fuente: Elaboración propia.

7. Crear los "*Roles*" que representan una identidad con los permisos específicos y sus credenciales de acceso, como se observa en la Figura 24.
8. Para asignar los roles, se debe seleccionar el rol de la función dentro de la pantalla "*Roles*" y se presiona sobre "C*rear rol*" como se observa en la Figura 25.

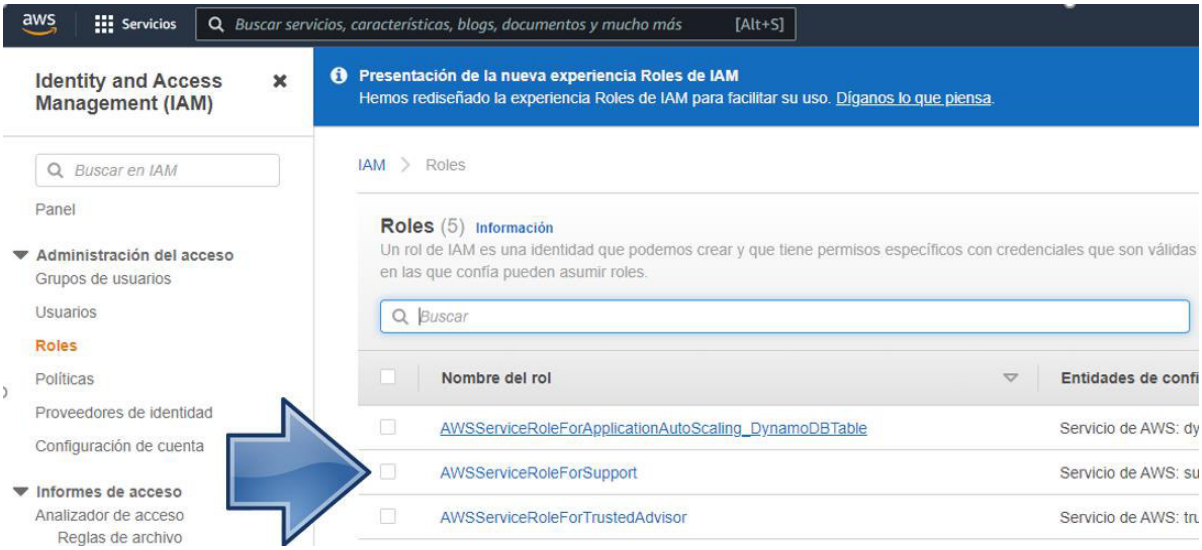


Figura 24. Roles de Lambdas de DynamoDB.

Fuente: Elaboración propia.

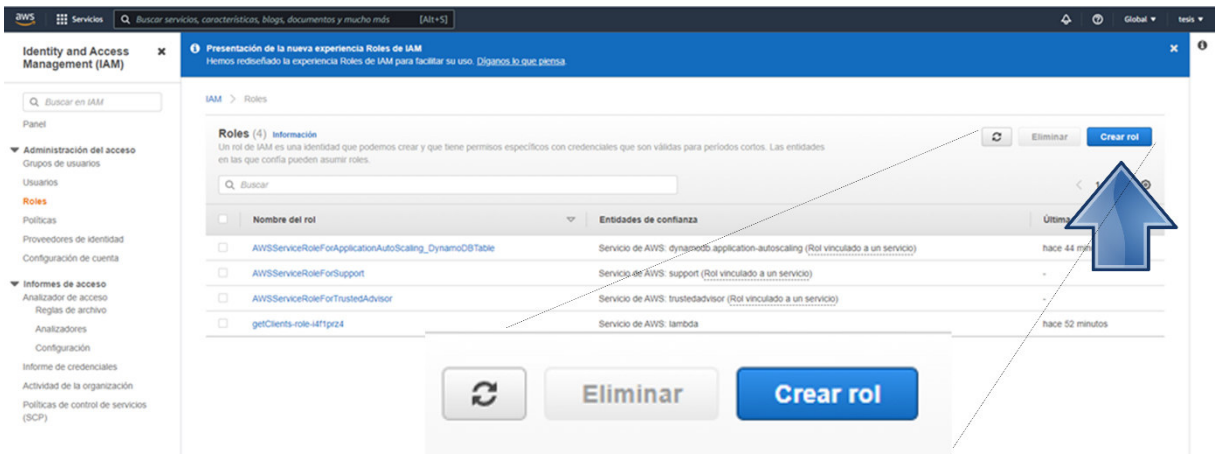


Figura 25. Selección de los roles de Lambdas de DynamoDB.

Fuente: Elaboración propia.

9. A continuación, se deben asociar las políticas de permisos al rol seleccionado como se puede observar en la Figura 26. En esta pantalla se debe seleccionar la opción "*Dynamo*" y el nombre de la política que en este caso es: "*AmazonDynmoDBFullAccess*", como se presenta en la Figura 27.
10. Además, se debe a agregar el permiso en las opciones "*Añadir Permiso*" y "*Crear política insertada*" que se presenta en la Figura 28.

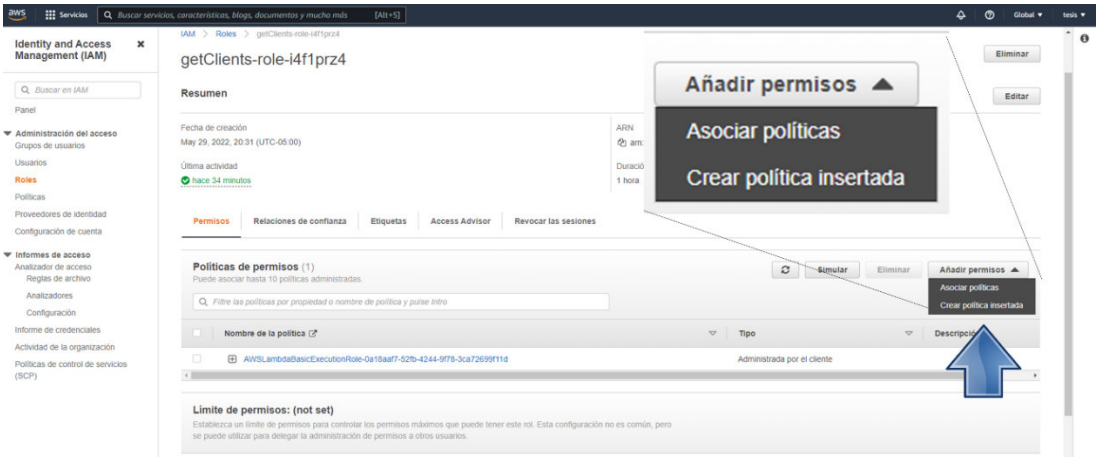


Figura 26. Añadir permisos en Lambdas de DynamoDB.

Fuente: Elaboración propia.

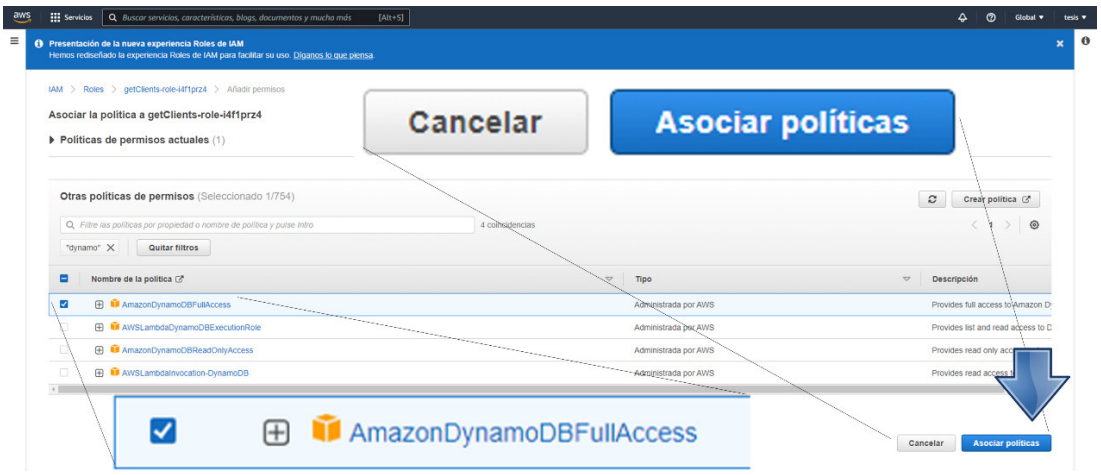


Figura 27. Selección de la política en Lambdas de DynamoDB.

Fuente: Elaboración propia.

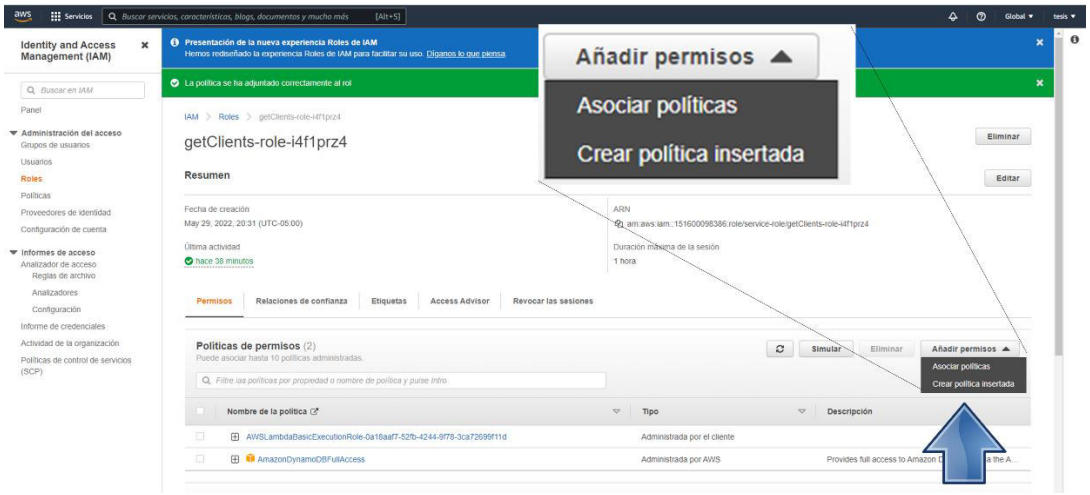


Figura 28. Pantalla añadir permiso en Lambdas de DynamoDB.

Fuente: Elaboración propia.

11. Elegir el servicio como se presenta en la Figura 28, buscar y seleccionar la opción "*Dynamo*"; luego seleccionar la opción "*revisar política*" como se presenta en la Figura 30.
12. Finalmente, se debe proporcionar permisos de lectura y escritura, y presionar sobre el botón "*revisar la política*" como se presenta en la Figura 31.

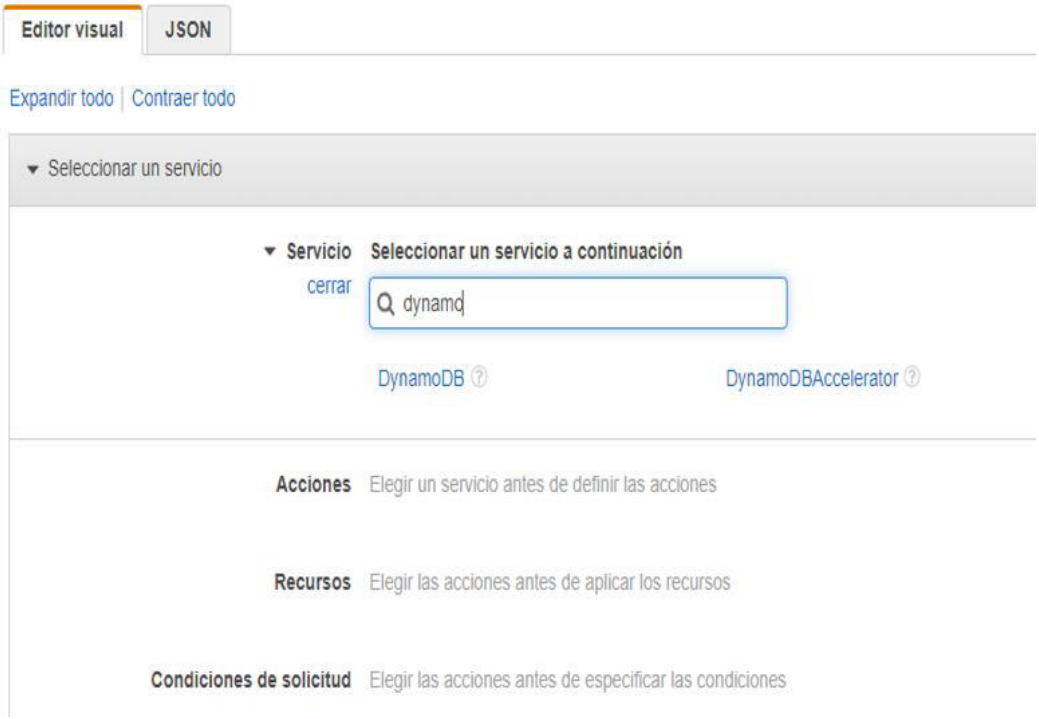


Figura 29. Elegir el servicio en Lambdas de DynamoDB.

Fuente: Elaboración propia.

Figura 30. Elegir revisar política en Lambdas de DynamoDB.

Fuente: Elaboración propia.

Figura 31. Crear políticas en Lambdas de DynamoDB.

Fuente: Elaboración propia.

13. Dentro de la pantalla antes mencionada se debe ingresar en la sección de "*recursos*" en todos los apartados y hay que seleccionar la opción "*Agregar ARN para restringir el acceso*". También se debe agregar la región y el nombre de la tabla como se presenta en la Figura 32.

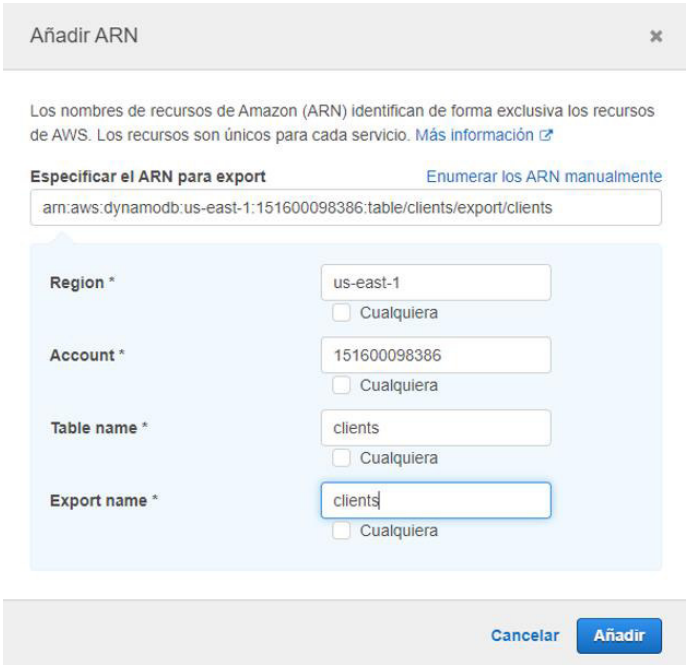


Figura 32. Añadir ARN en Lambdas de DynamoDB.

Fuente: Elaboración propia.

14. Una vez se ingresa a la pantalla "*ARN*" en la sección "*Index*" se debe colocar la clave primaria de la tabla correspondiente, y en la opción "*stream*", en el apartado "*Stream label* *" se debe colocar el signo "*" y habilitar el check como se observa en la Figura 33.
15. Presionar sobre la opción de "*Revisar la política*" como se observa en la Figura 34.

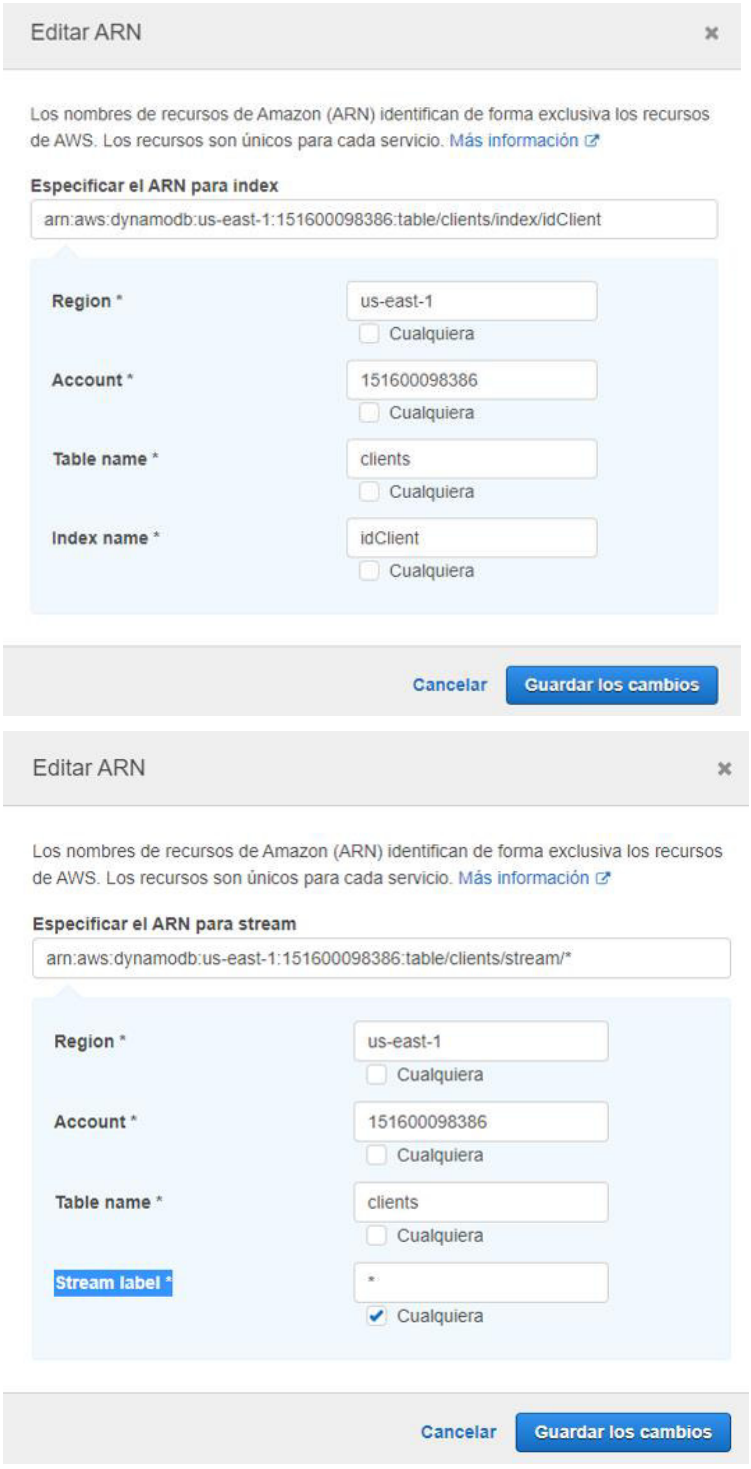


Figura 33. Editar ARN en Lambdas de DynamoDB.

Fuente: Elaboración propia.

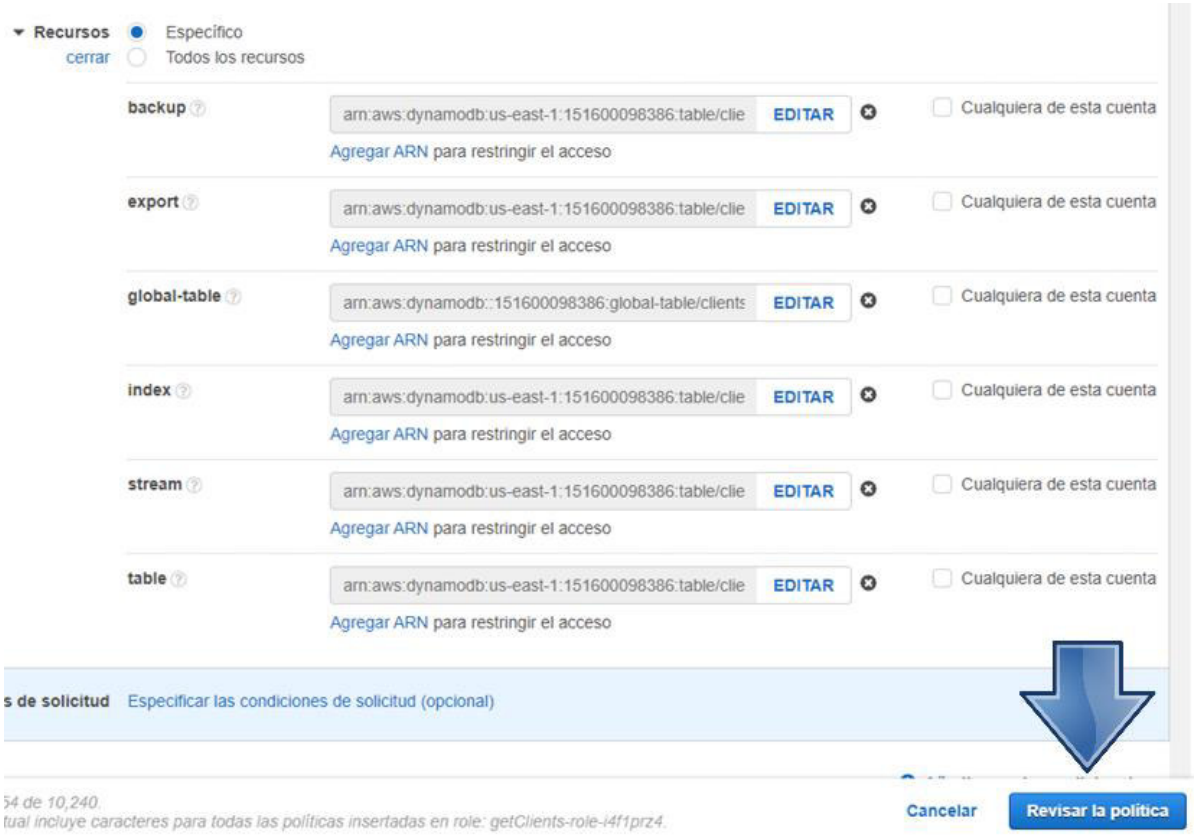


Figura 34. Recursos de ARN en Lambdas de DynamoDB.

Fuente: Elaboración propia.

Terminada la generación de Lambda para la creación de las funciones, ya es posible interactuar con las bases de datos. Seguidamente en la sección **3.4.3.4.** se presenta alguno de los códigos empleados para la interacción entre las bases de datos y la integración del ambiente para la creación del CDW.

Creación de la API Gateway

1. Generar la API Gateway en la opción "*Amazon API Gateway*" y seleccionar el apartado "*API REST*" como se presenta en la Figura 35

Figura 35. Pantalla de Amazon API Gateway.

Fuente: Elaboración propia.

2. Asignar inicialmente un nombre en "*Revisar la política*" como se presenta en la Figura 36.

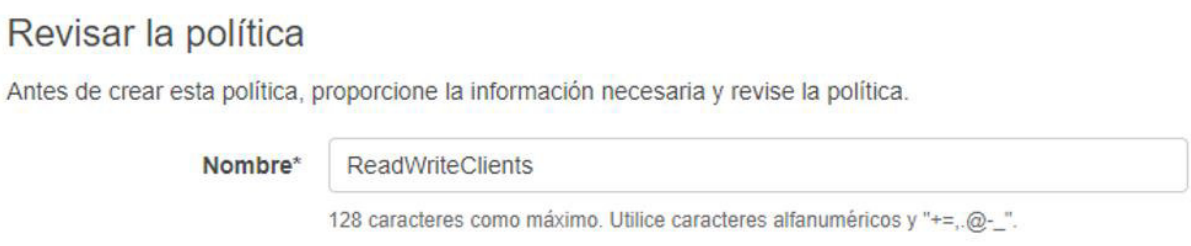


Figura 36. Pantalla crear política de Amazon API Gateway.

Fuente: Elaboración propia.

3. Crear la API Gateway, con la opción “*Crear API nueva*”.
4. Crear las rutas de acceso por medio de la opción "*Acciones*" y "*Crear recurso*".
5. Se coloca el nombre de la ruta y se presiona en “*Crear recurso*”, como se observa en la Figura 37 y Figura 38.

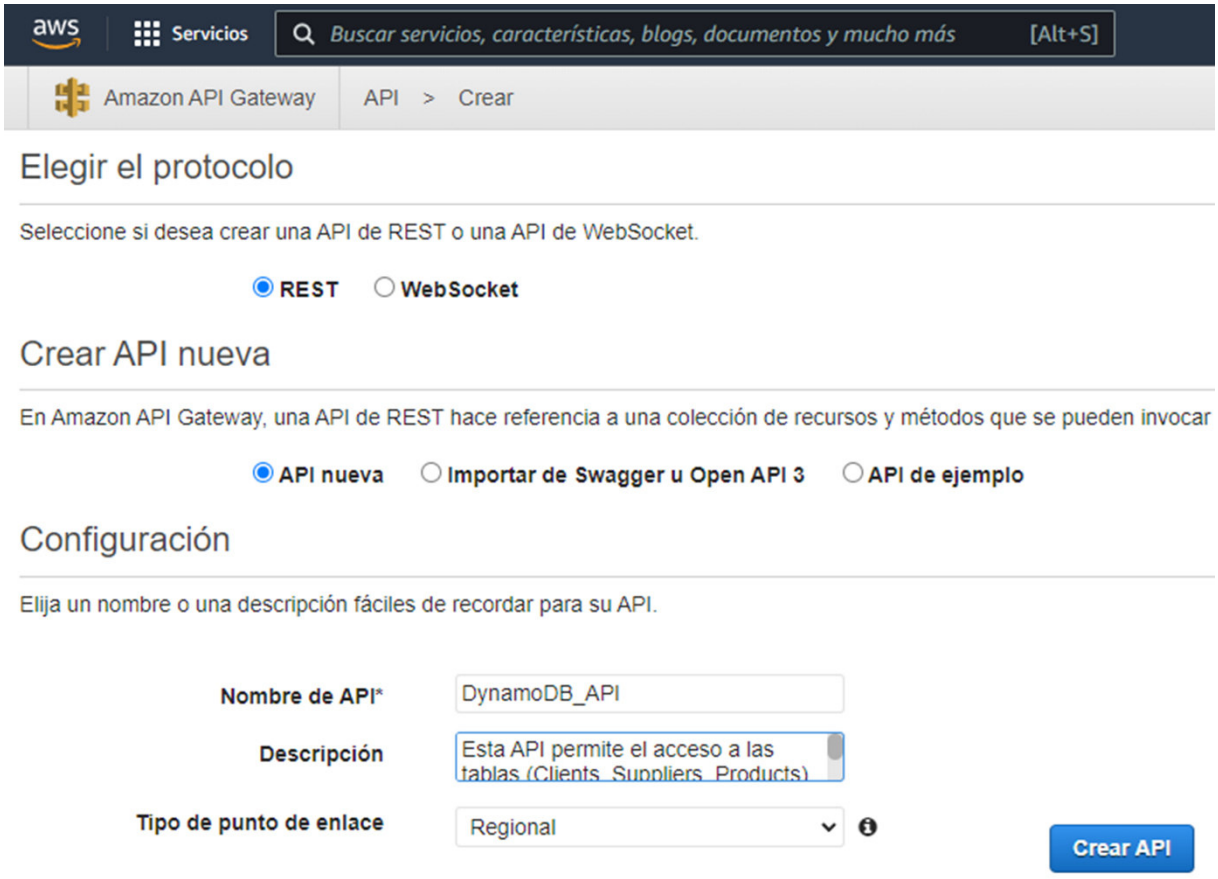


Figura 37. Pantalla crea API nueva.

Fuente: Elaboración propia.

Figura 38. Pantalla crea rutas de acceso.

Fuente: Elaboración propia.

6. Para crear el método Get, se selecciona la opción "*Crear un método*" y se selecciona "*GET*" como se presentan en las Figura 39 y Figura 40.

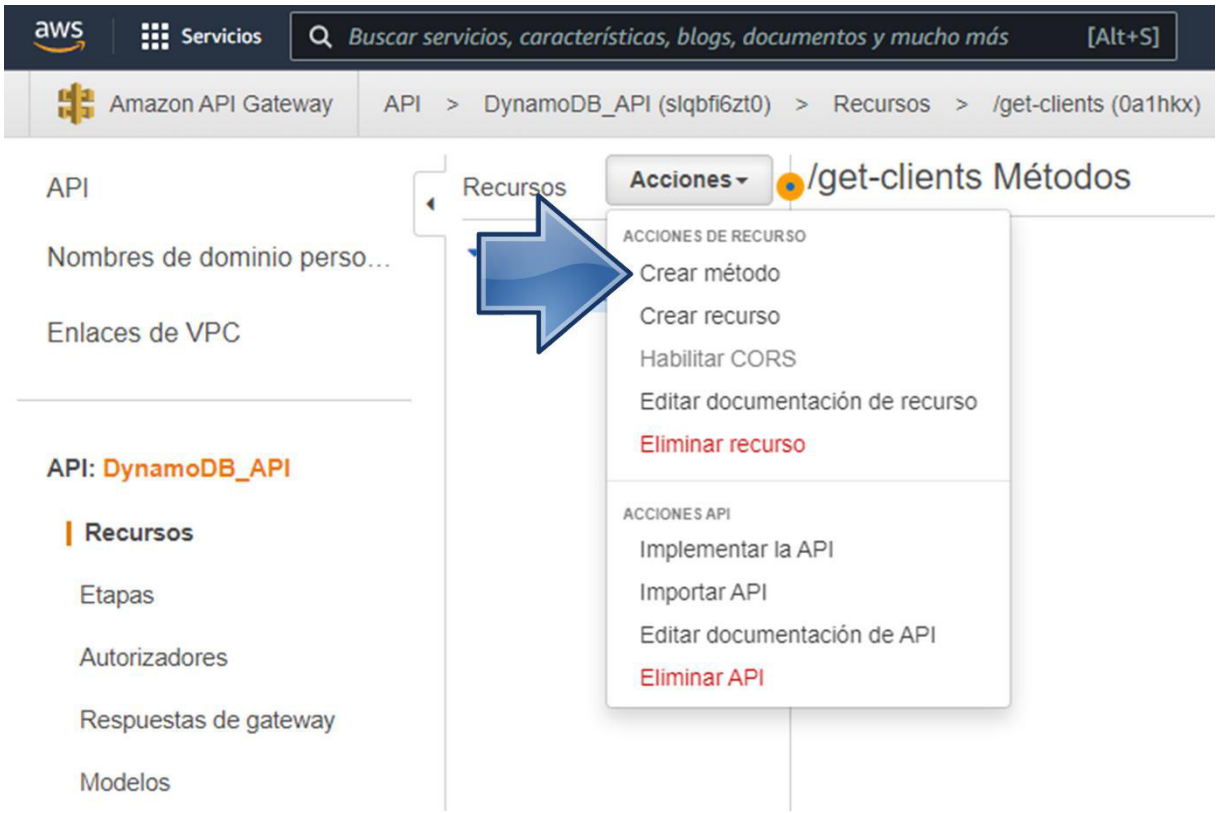


Figura 39. Opción crear un método.

Fuente: Elaboración propia.

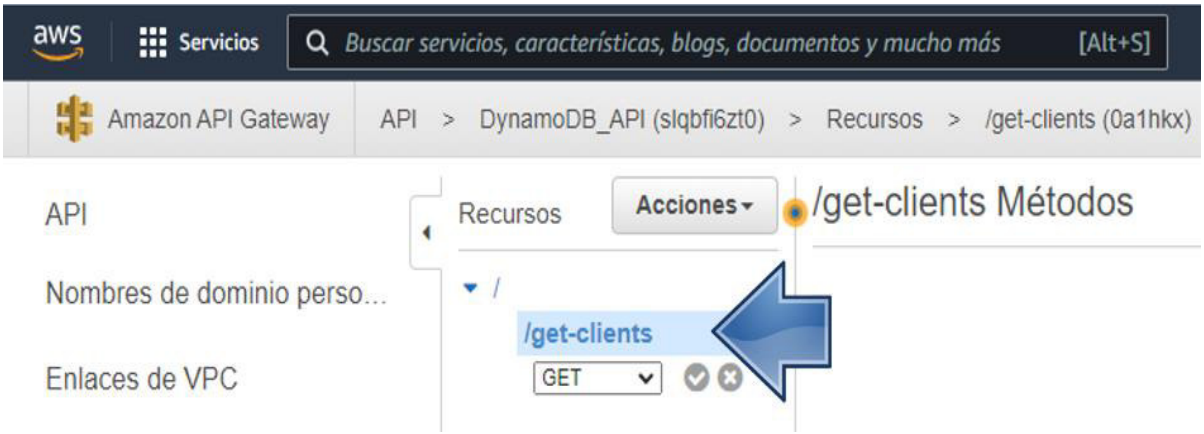


Figura 40. Opción GET.

Fuente: Elaboración propia.

7. Seleccionar la opción "*Función Lambda*" y "*Usar la integración de proxy Lambda*" y se asigna el nombre de la función que debe responder a la petición como se presenta en la Figura 41.

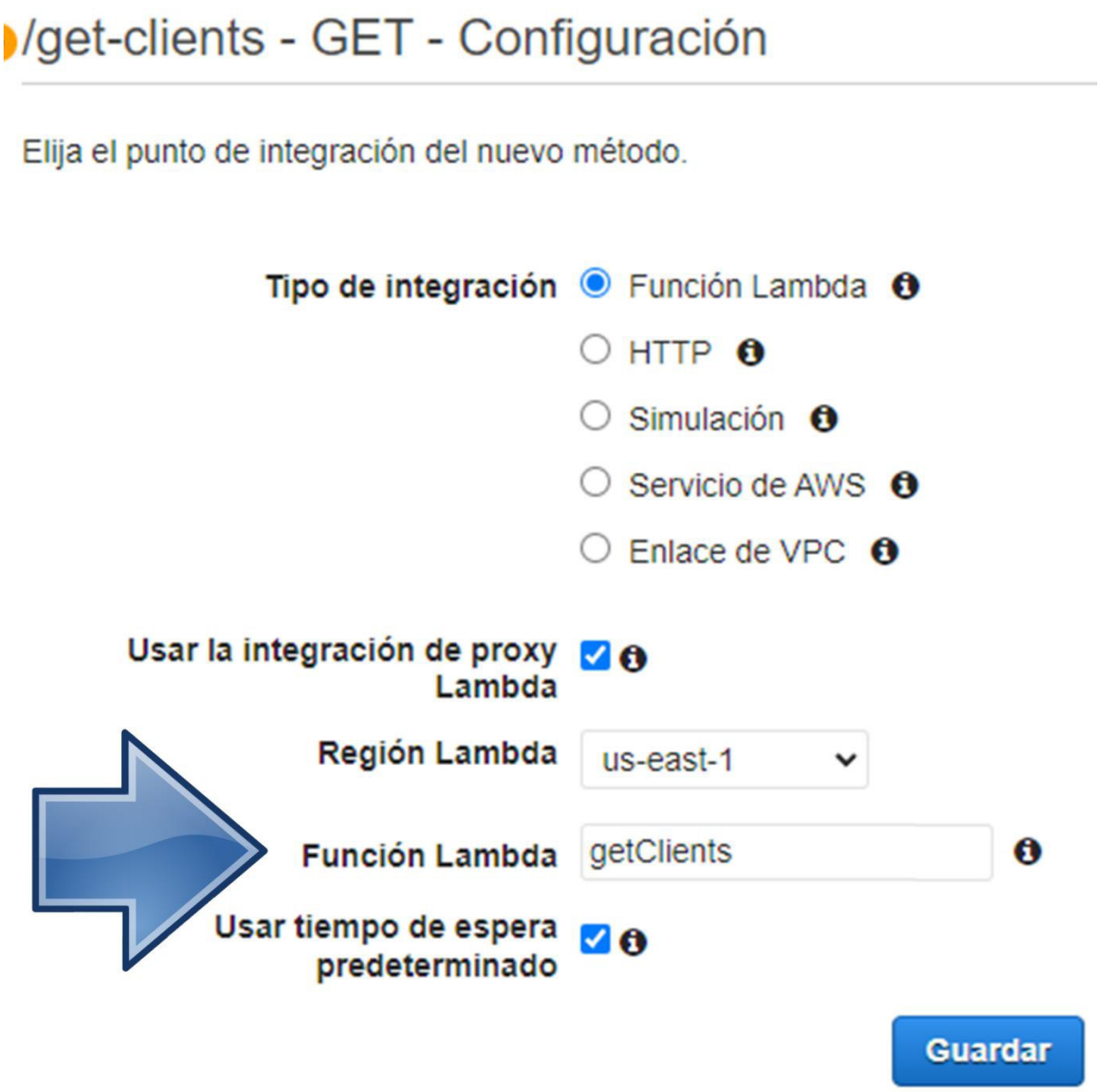


Figura 41. Pantalla configuración GET.

Fuente: Elaboración propia.

8. Utilizar la base de datos Postgres para el manejo de la base de datos relacional creada en la Nube 2.
9. Crear el CDW empleando los diferentes tipos y fuentes de datos, para permitir el manejo de las tablas de la Nube 2.
10. Para el manejo de las tablas se emplea DynamoDB que es un servicio de base de datos NoSQL que facilita el manejo de las tablas de la Nube 1.

Terminada la creación del API Gateway ya es posible interactuar con esta. Seguidamente, en la sección **3.4.3.4.** se presenta la integración del ambiente para la creación del CDW.

3.4.3.2. Construcción de la Nube 2

A continuación, se presenta la creación de la instancia de origen "*rds-origin*", donde por medio de la "*Cuenta 2*" se genera la instancia de base de datos que contiene la tabla de ventas "*sales_cld2*".

1. Buscar "*RDS"* como se presenta en la Figura 42.
2. Buscar la opción "*Crear base de datos*" como se presenta en la Figura 43.

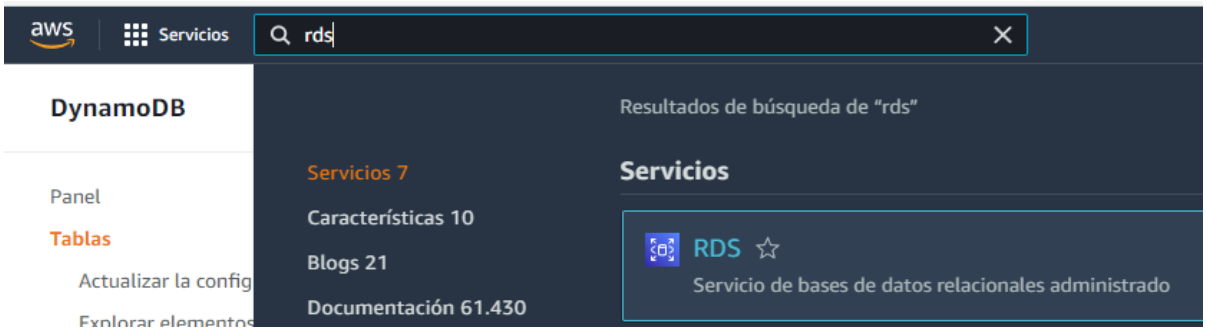


Figura 42. RDS en DynamoDB.

Fuente: Elaboración propia.

Figura 43. Opción crear base de datos en DynamoDB.

Fuente: Elaboración propia.

3. Seleccionar el motor de base de datos a utilizar y se elige "*PostgreSQL*" y la "*versión (14-1)*" como se presenta en la Figura 44.
4. En "*Plantilla*", se selecciona la "*Capa gratuita*" como se muestra en la Figura 45.

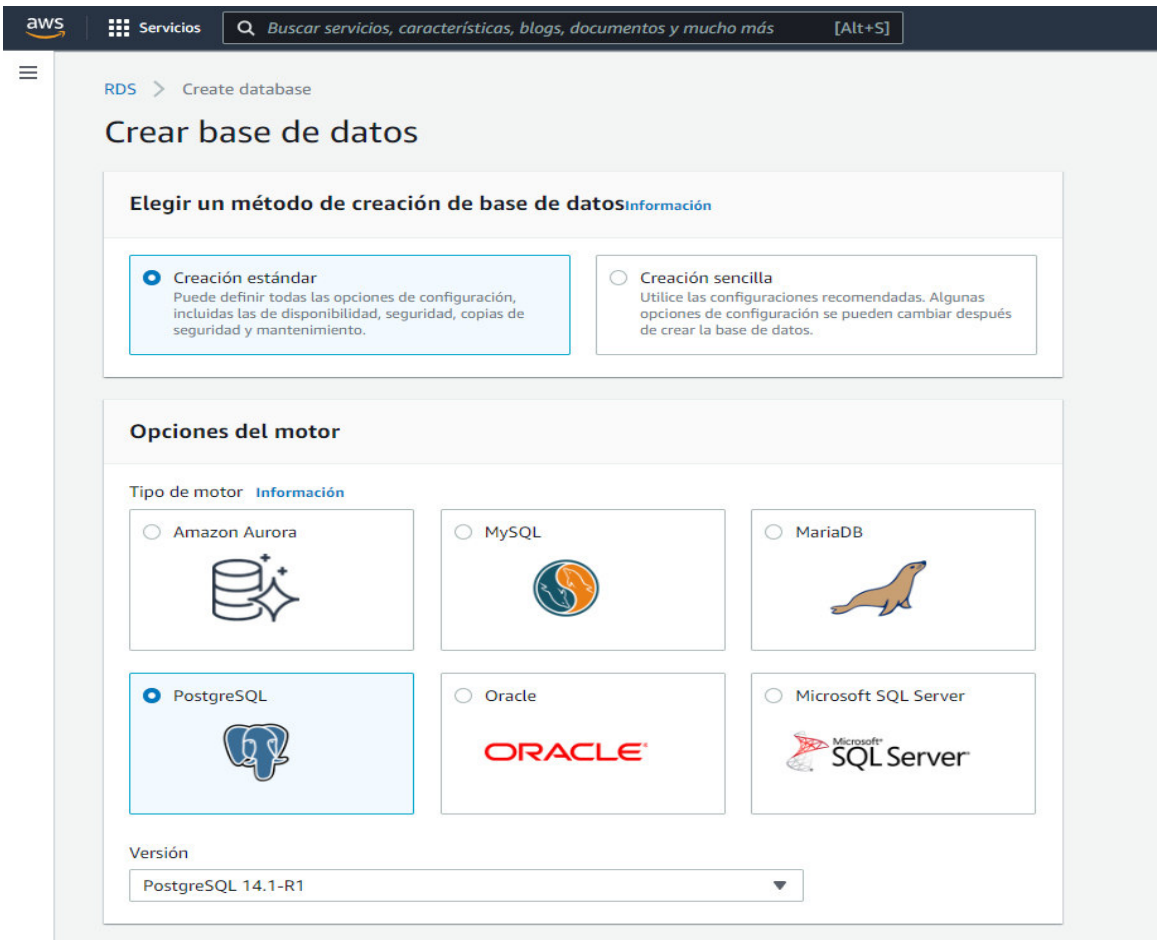


Figura 44. Opción base de datos PostgreSQL en DynamoDB.

Fuente: Elaboración propia.

Figura 45. Opción plantillas capa gratuita en DynamoDB.

Fuente: Elaboración propia.

5. Nombrar la base de datos en "*Configuración*" y escribir las credenciales del administrador como se observa en la Figura 46.

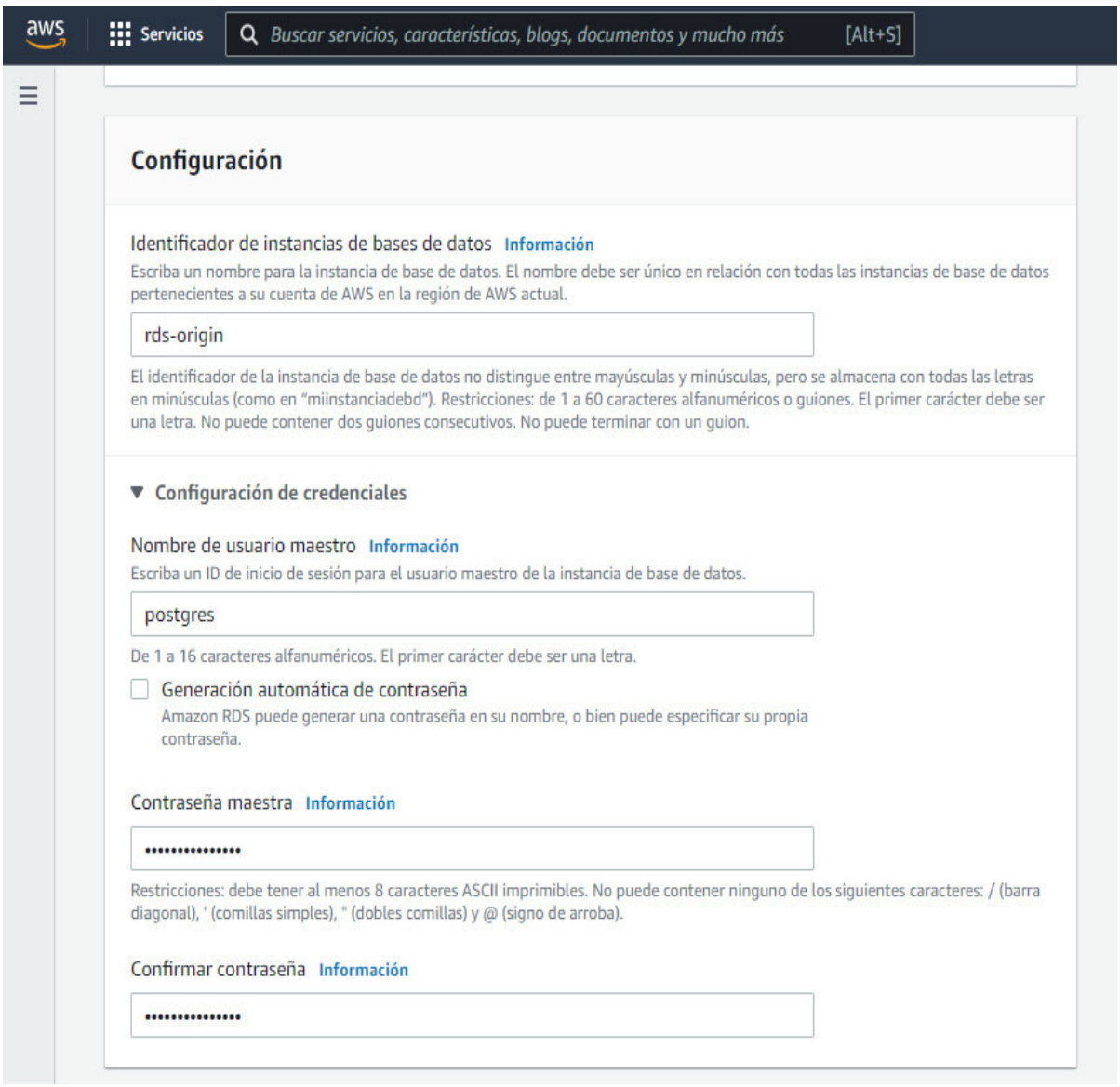


Figura 46. Configuración de la base de datos en DynamoDB.

Fuente: Elaboración propia.

6. Dejar por defecto la opción "*Configuración de la instancia*" y el "*Almacenamiento*" y se deja activada la opción "*Habilitar escalado automático de almacenamiento*", que permite el crecimiento de la capacidad a medida de la exigencia como se presenta en la Figura 47.

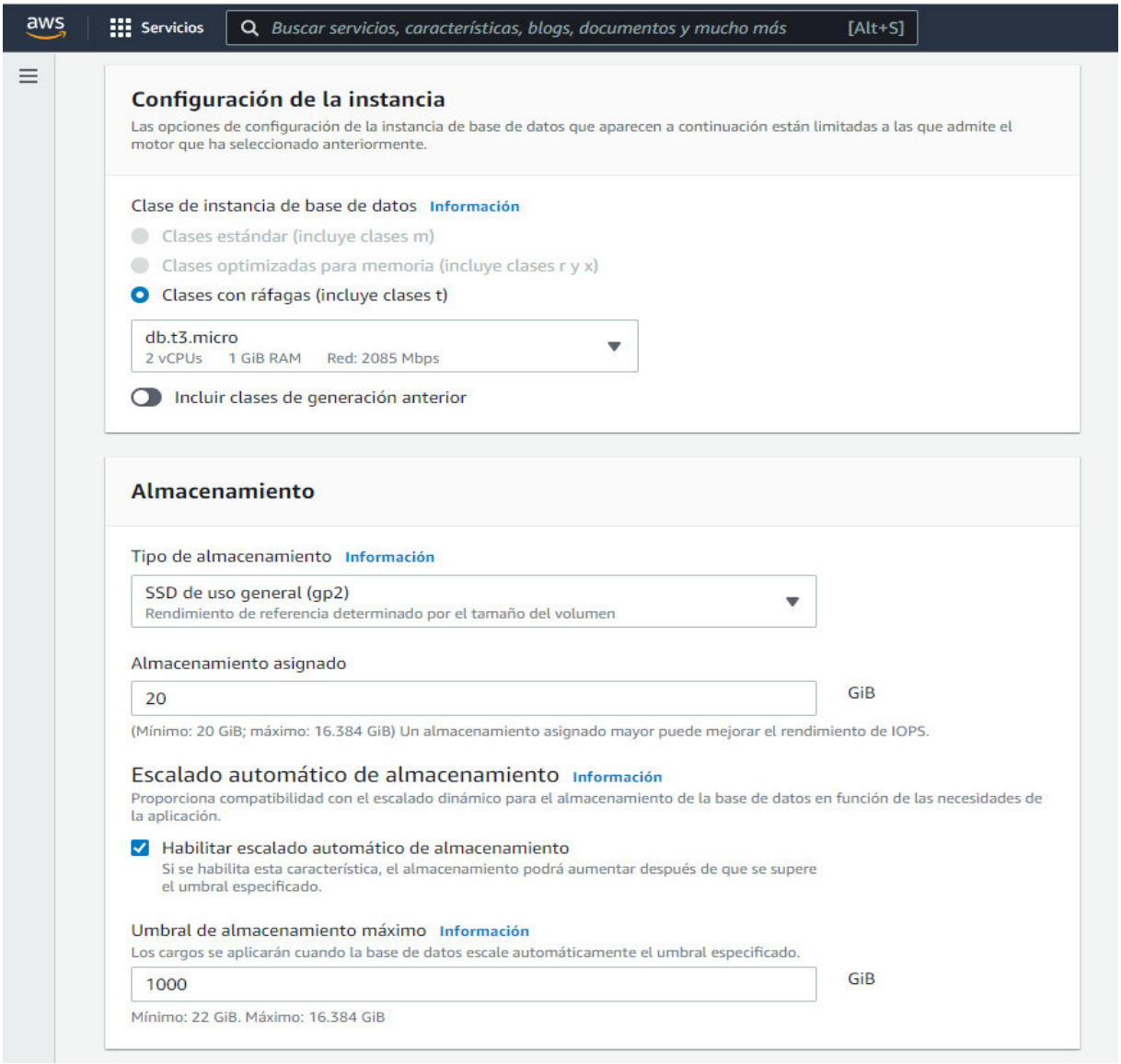


Figura 47. Configuración de la instancia de datos en DynamoDB.

Fuente: Elaboración propia.

7. En "*Conectividad*", se selecciona en "*Tipo de red*" la opción "*IPv4*", y en "*Acceso público*" se elige la opción "*Si*" como se presenta en la Figura 48.
8. En "*Autenticación de base de datos*" se selecciona la "*Autenticación con contraseña*", y se selecciona la opción "*Crear base de datos*" como se muestra en la Figura 49.

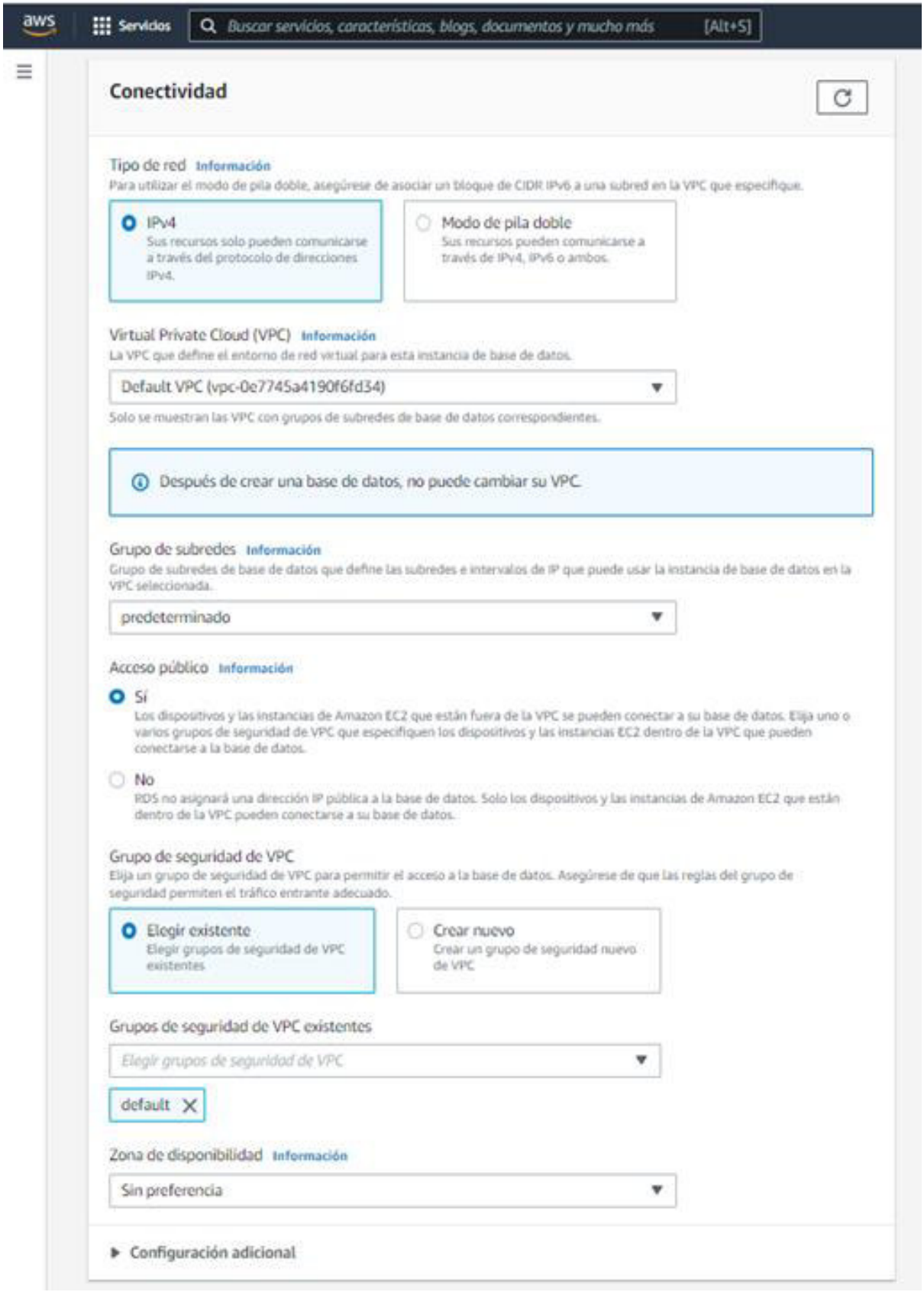


Figura 48. Configuración de la conectividad de datos en DynamoDB.

Fuente: Elaboración propia.

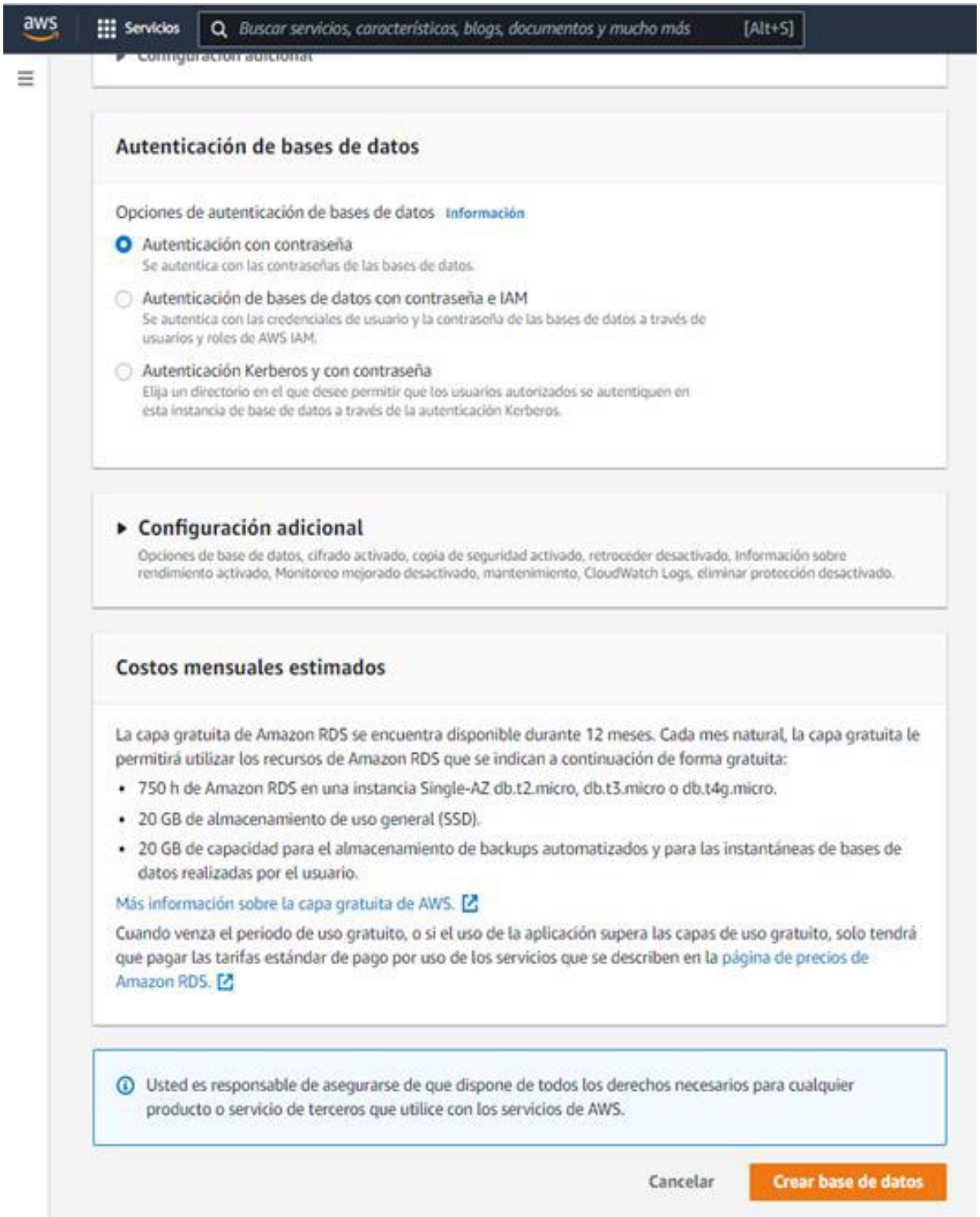


Figura 49. Autenticación de base de datos en DynamoDB.

Fuente: Elaboración propia.

9. Terminado el proceso queda creada la base de datos como se observa en la Figura 50.

Figura 50. Base de datos en DynamoDB.

Fuente: Elaboración propia.

10. En "*Seguridad*", se selecciona "*Grupos de seguridad de la VPC"* como se presenta en la Figura 51.

Figura 51. Grupos de seguridad de la VPC en DynamoDB.

Fuente: Elaboración propia.

11. Presionar "*Crear grupo de seguridad*" como se muestra en la Figura 52.

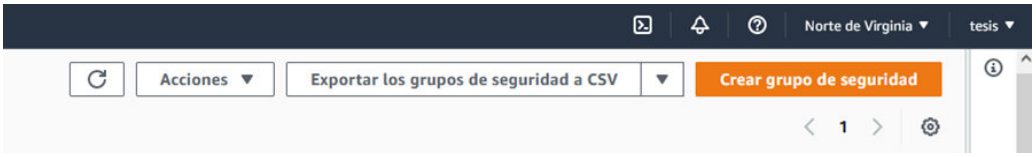


Figura 52. Crear grupo de seguridad en DynamoDB.

Fuente: Elaboración propia.

12. Colocar el nombre del grupo, y las reglas de entrada y salida y seleccionar todas las IP permitidas 0.0.0.0/0.
13. Colocar la dirección IP solo la del servidor del ETL como se presenta en la Figura 53.

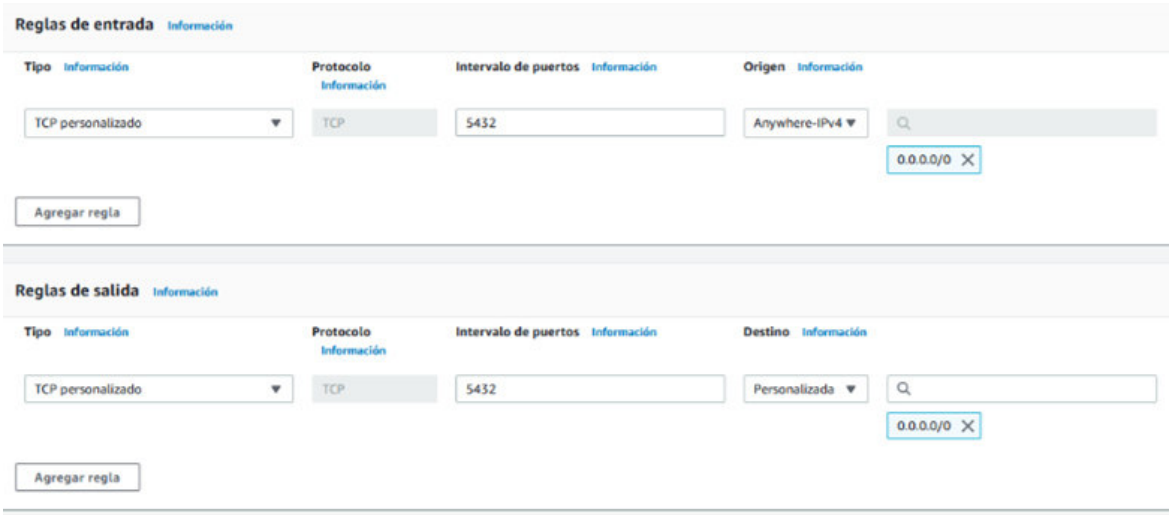


Figura 53. Configuración del grupo de seguridad en DynamoDB.

Fuente: Elaboración propia.

14. Seleccionar "*Modificar*" como se muestra en la Figura 54.

Figura 54. Modificar rds-origin en DynamoDB.

Fuente: Elaboración propia.

15. En "*Conectividad*" se selecciona el grupo de seguridad como se presenta en la Figura 55.

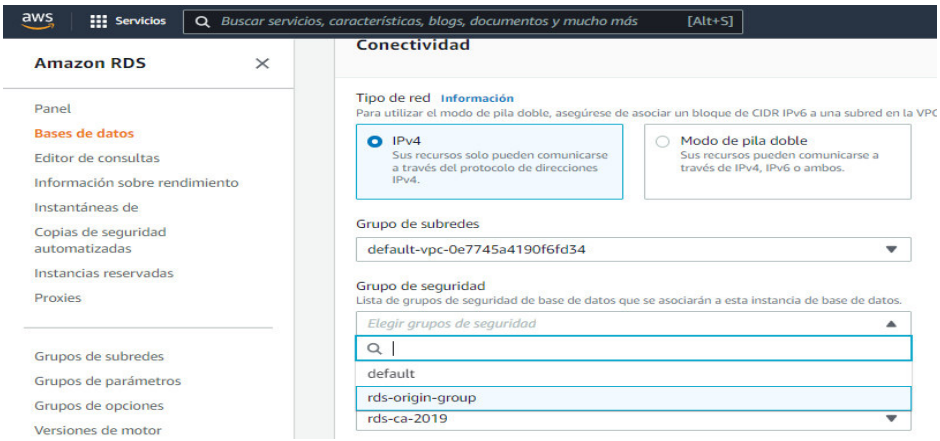


Figura 55. Conectividad RDS en DynamoDB.

Fuente: Elaboración propia.

Una vez concluido con los pasos anteriores, es posible conectarse a DynamoDB desde cualquier administrador de bases de datos y crear las tablas necesarias. En este caso se emplea la herramienta pgAdmin 4. Por otra parte, algunos de los códigos empleados para interactuar con las bases de datos en esta herramienta se describen en la sección **3.8.4.** para realizar la integración del ambiente del CDW.

Creación de la conexión a la instancia AWS

Para realizar la creación de la conexión a la instancia AWS de "*rds-origin*" (Nube 2), la base de datos "*rds-origin*" y la tabla "*sales_cld2*" se deben realizar los siguientes pasos:

1. Abrir la herramienta "*pgAdmin*", se elige "*nueva conexión*" y configurar el nuevo servidor con los datos de "*Punto de enlace y puerto*" del detalle de la base de datos como se presenta en la Figura 56 y Figura 57

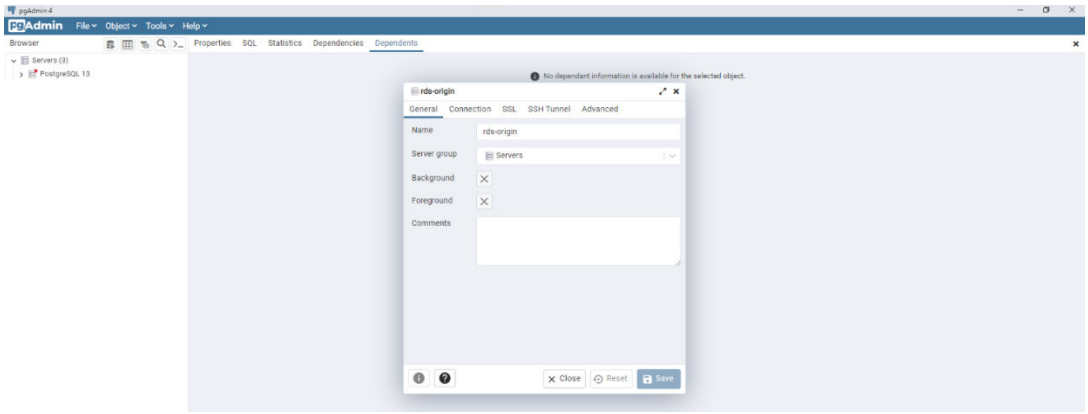


Figura 56. Configuración de la base de datos en pgAdmin.

Fuente: Elaboración propia.

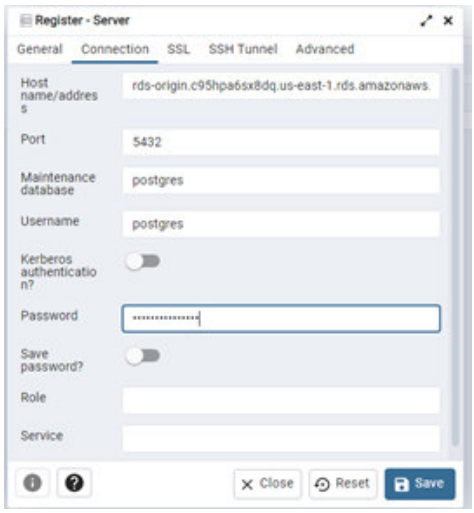


Figura 57. Registrar Servidor en pgAdmin.

Fuente: Elaboración propia.

2. Crear la tabla "*sales_cld2*" como se presenta en la Figura 58

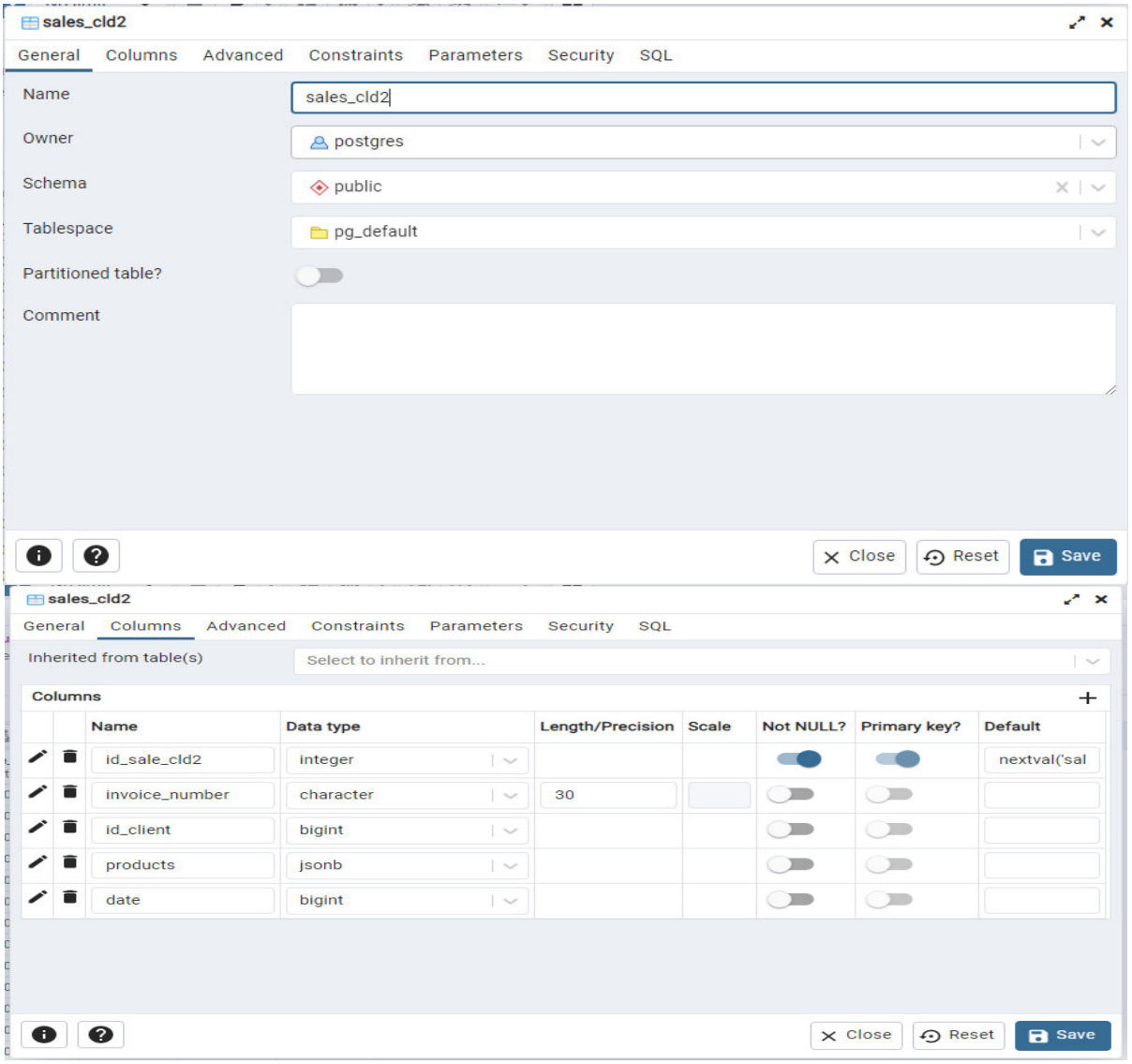


Figura 58. Creación de tablas en pgAdmin.

Fuente: Elaboración propia.

3. Realizados los pasos descritos anteriormente se consigue la conexión y es posible cargar los datos como se muestra en la Figura 59.

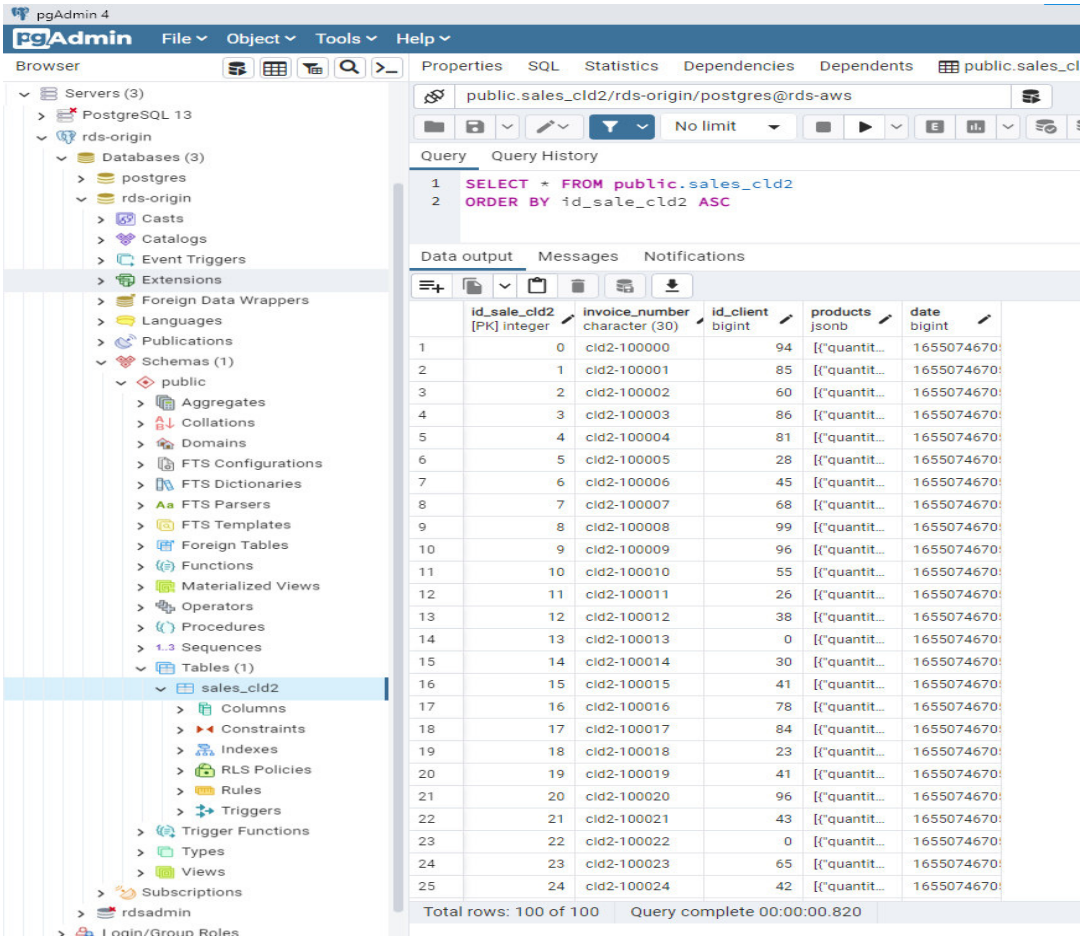


Figura 59. Carga de datos en pgAdmin.

Fuente: Elaboración propia.

Terminada la creación de la conexión es posible interactuar con las bases de datos. Seguidamente, en la sección **3.4.3.4.** se presenta la integración del ambiente para la creación del CDW.

3.4.3.3. Construcción de la Nube 3

Creación de la conexión a la instancia AWS

Para la creación de la conexión a la instancia AWS de integración "*rds-dwh-thesis*", la base de datos "*rds-dwh*" y las tabla "*sales_datamar*" y "*purch_datamart*" en la "*Cuenta 3*", se crea la base de datos de integración, que contiene las tablas de los Data Marts de Ventas "*sales_datamar*" y de Compras "purch_datamart". Esta conexión se crea de la misma forma que se creó anteriormente la conexión a la instancia AWS de "*rds-origin*" (Nube 2) en el punto **3.8.2.1** del presente documento

1. En la Figura 60 se presenta la conexión a la instancia rds-dwh-thesis.

Figura 60. RDS rds-dwh-thesis.

Fuente: Elaboración propia.

2. En la Figura 61 se presenta la creación de la tabla sales_datamar y en la Figura 62 se presenta la creación de la tabla purch_datamart.

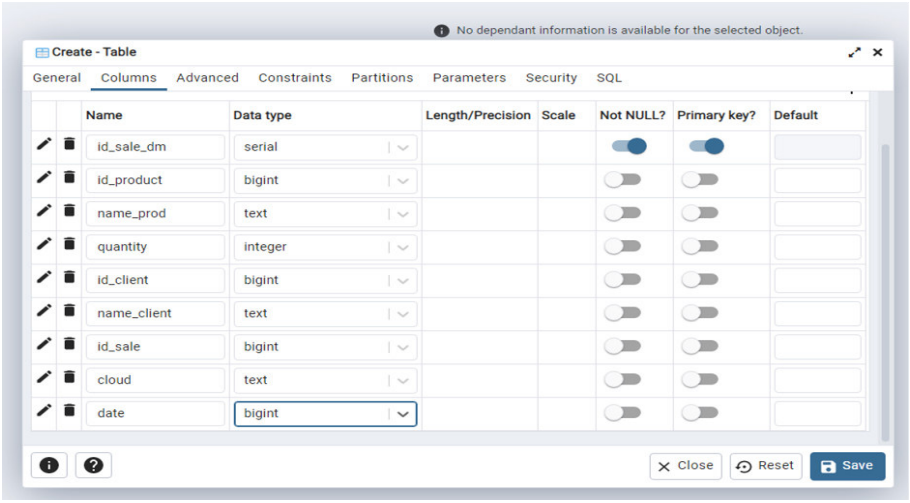


Figura 61. Tabla sales_datamart en pgAdmin.

Fuente: Elaboración propia.

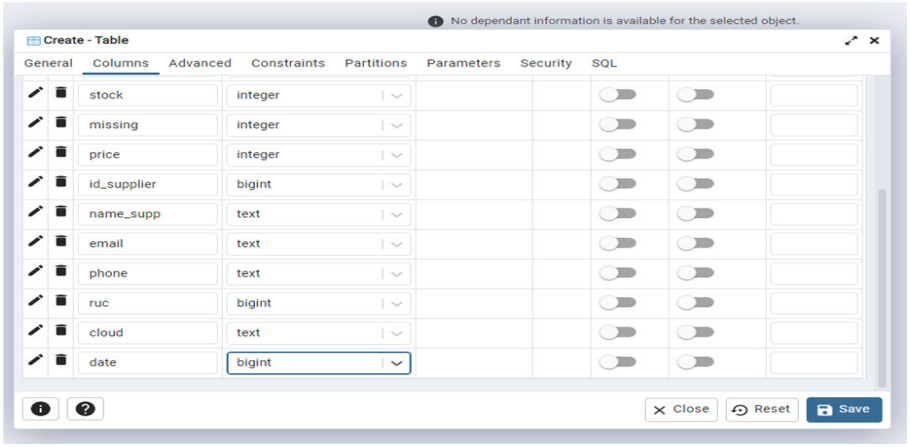


Figura 62. Tabla purch_datamart en pgAdmin.

Fuente: Elaboración propia.

3. Configurar el entorno para la API de producción como se presenta en Figura 63 y Figura 64.

Figura 63. Implementación de la API.

Fuente: Elaboración propia.

Figura 64. Pantalla Etapas.

Fuente: Elaboración propia.

Terminada la creación de la conexión ya es posible interactuar con las bases de datos. Seguidamente en la sección **3.4.3.4.** se describe la integración del ambiente para la creación del CDW.

Instalación y configuración de Power BI

Esta fase se inicia con la etapa de instalación de Power BI por medio de la cuenta 3, en la instancia del VPS Lightsail (con Windows Server 2019) de AWS.

1. Instalar y configurar Power BI como se presenta en la Figura 65.

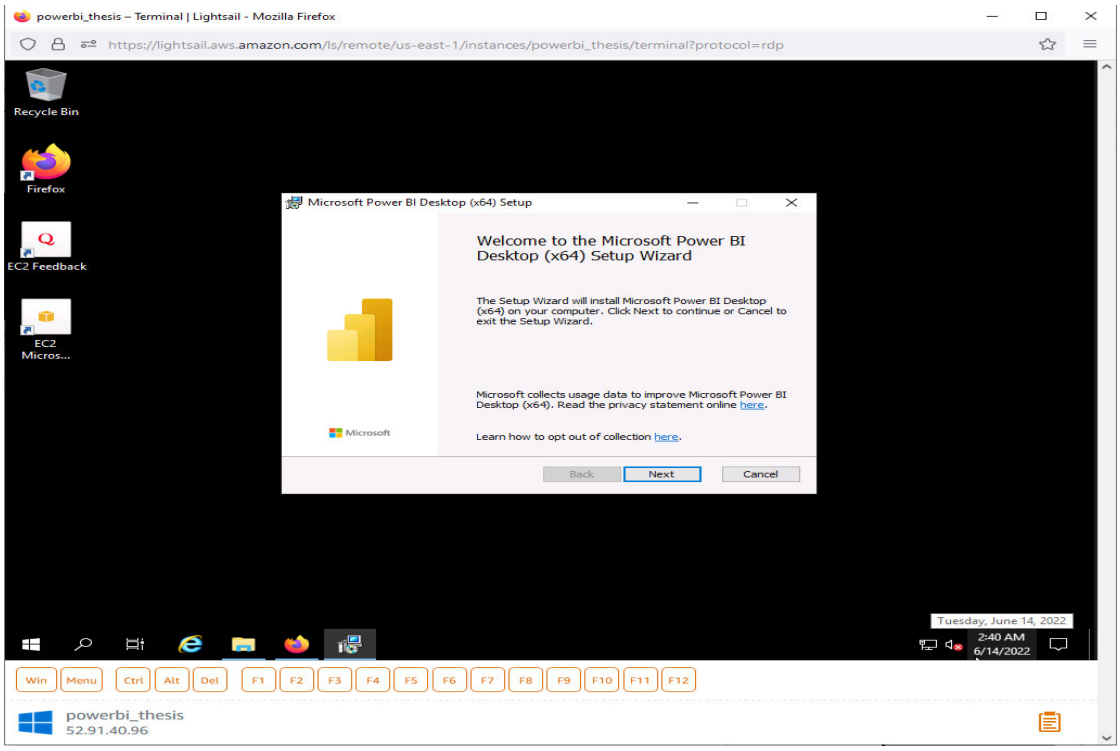


Figura 65. Instalador de Power BI.

Fuente: Elaboración propia.

2. Iniciar la herramienta para configurar y realizar la conexión con el ETL como se presenta en la Figura 66.

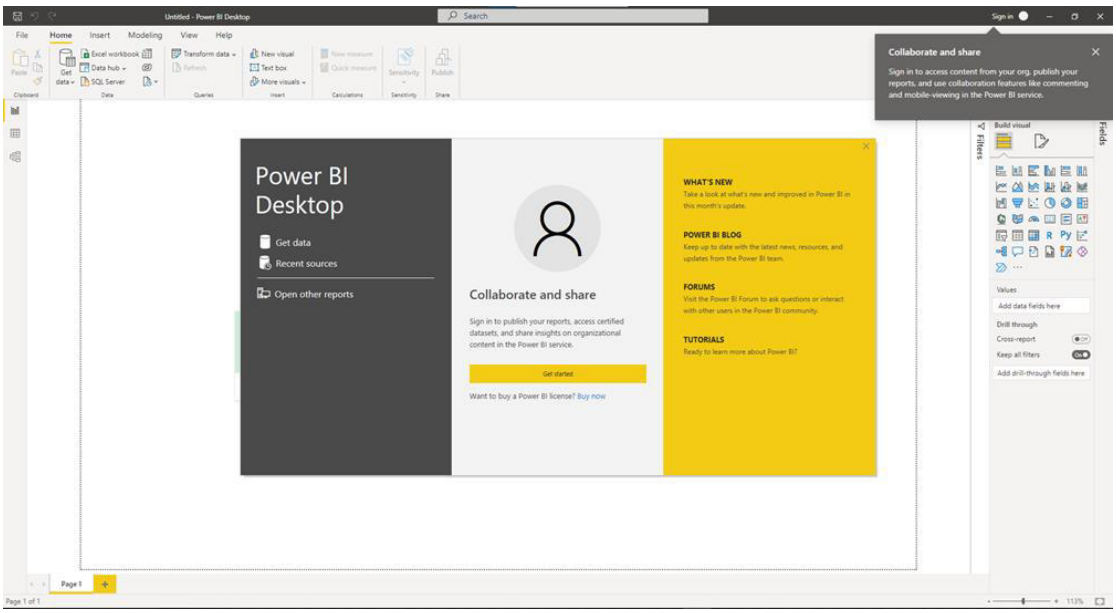


Figura 66. Pantalla inicial de Power BI.

Fuente: Elaboración propia.

3. Presionar sobre el botón "*GET*", luego la opción "*Other*", y seleccionar la opción "*Black Query*", y por último se presiona "*Connect*" como se observa en la Figura 67.

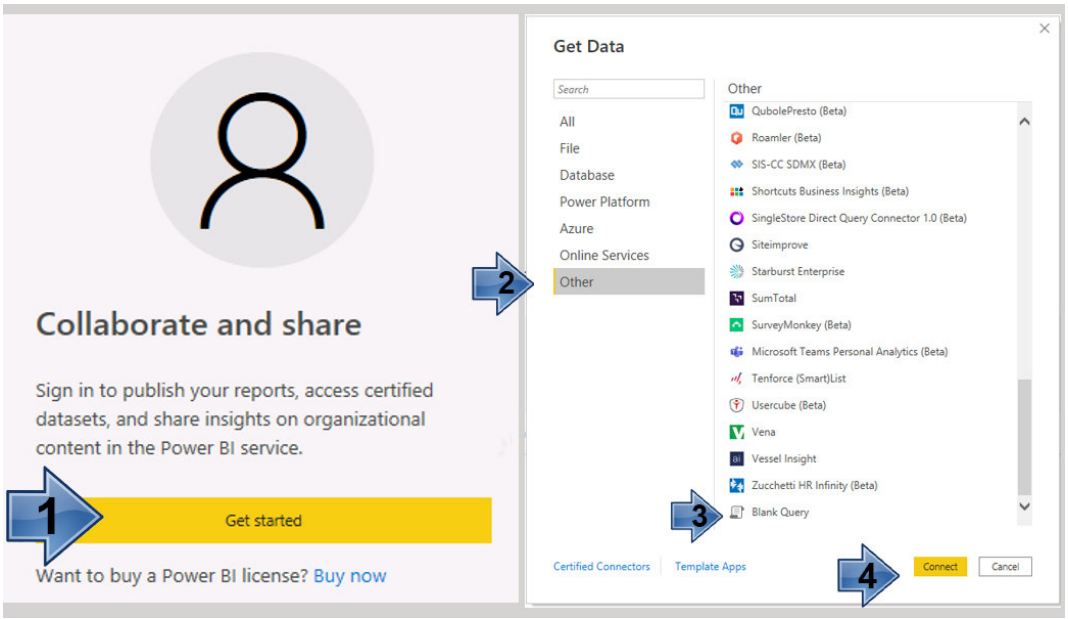


Figura 67. Configuración inicial de Power BI.

Fuente: Elaboración propia.

4. Presionar sobre la opción "*Advanced Editor*" como se observa en la Figura 68.

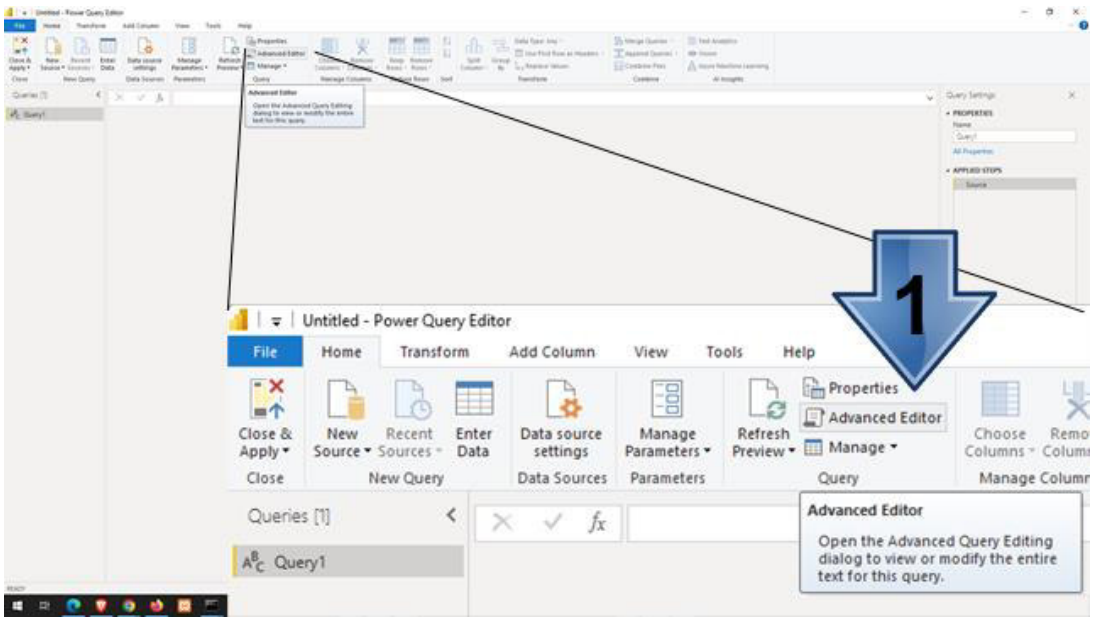


Figura 68. Configuración Advanced editor de Power BI.

Fuente: Elaboración propia.

5. En "*Query1*" configurar el ETL por medio del "*Query*" que se presenta en la Figura 69.
6. Presionar sobre "*Done*".

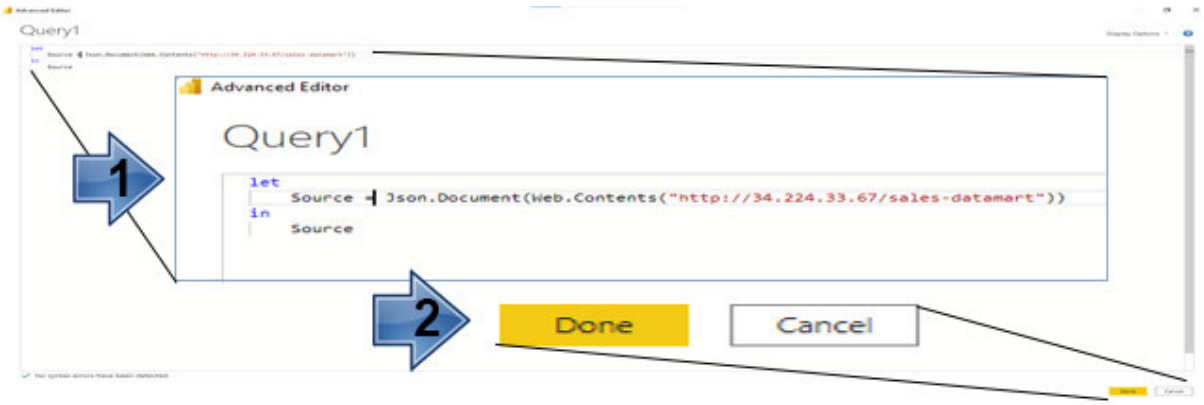


Figura 69. Configuración de Power BI.

Fuente: Elaboración propia.

7. Seleccionar la opción "*LIST*" como se presenta en la Figura 70.
8. Presionar sobre la opción "*TO TABLE*".
9. Presionar "*OK*", como se observa en la Figura 71.

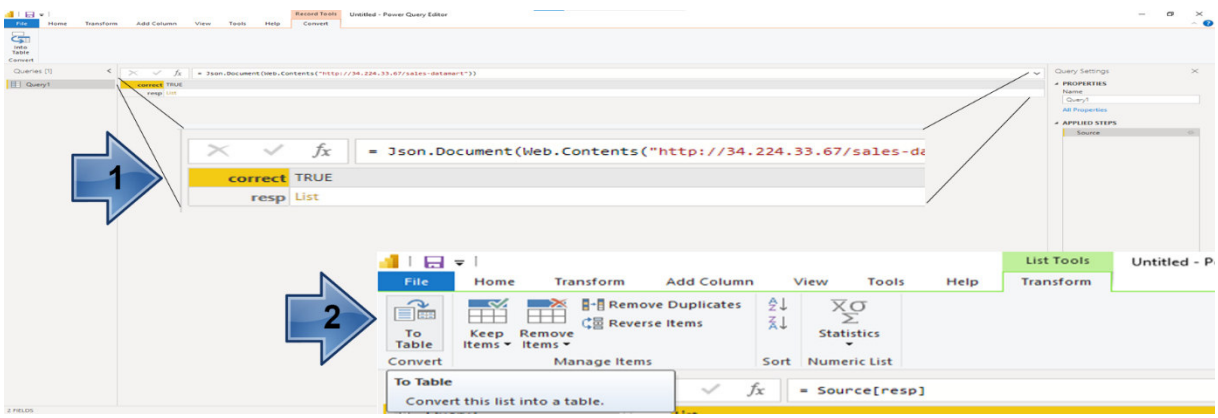


Figura 70. Configuración ETL de Power BI.

Fuente: Elaboración propia.

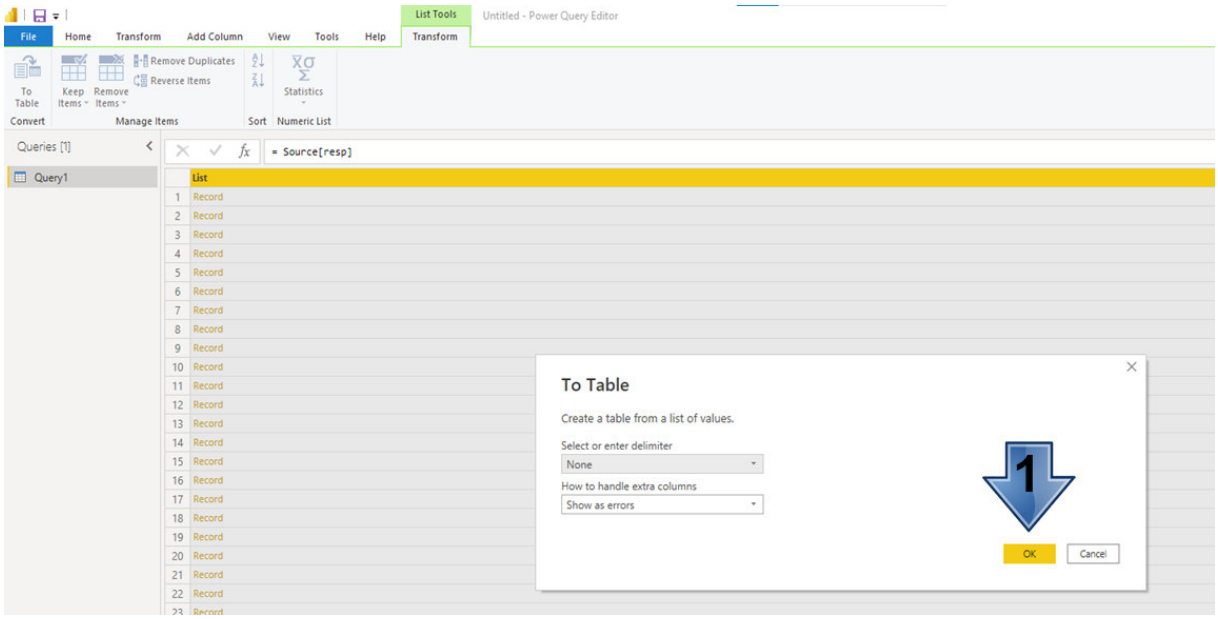


Figura 71. Procesamiento del Query en Power BI.

Fuente: Elaboración propia.

10. Una vez realizada la configuración, la herramienta comienza a presentar los datos y se puede iniciar la elaboración de gráficos a partir de la información, como se presenta en la Figura 72.

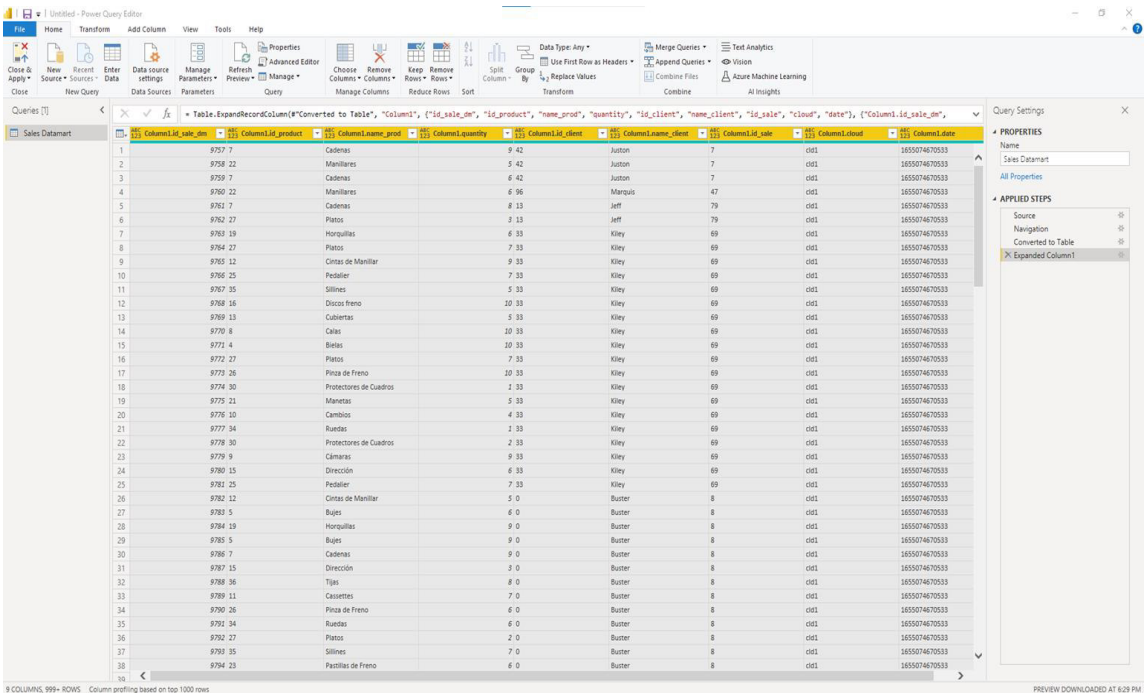


Figura 72. Datos en Power BI.

Fuente: Elaboración propia.

11. Una vez configurado Power BI, ya se pueden hacer los dashboard en la plataforma como se presenta en la Figura 73.

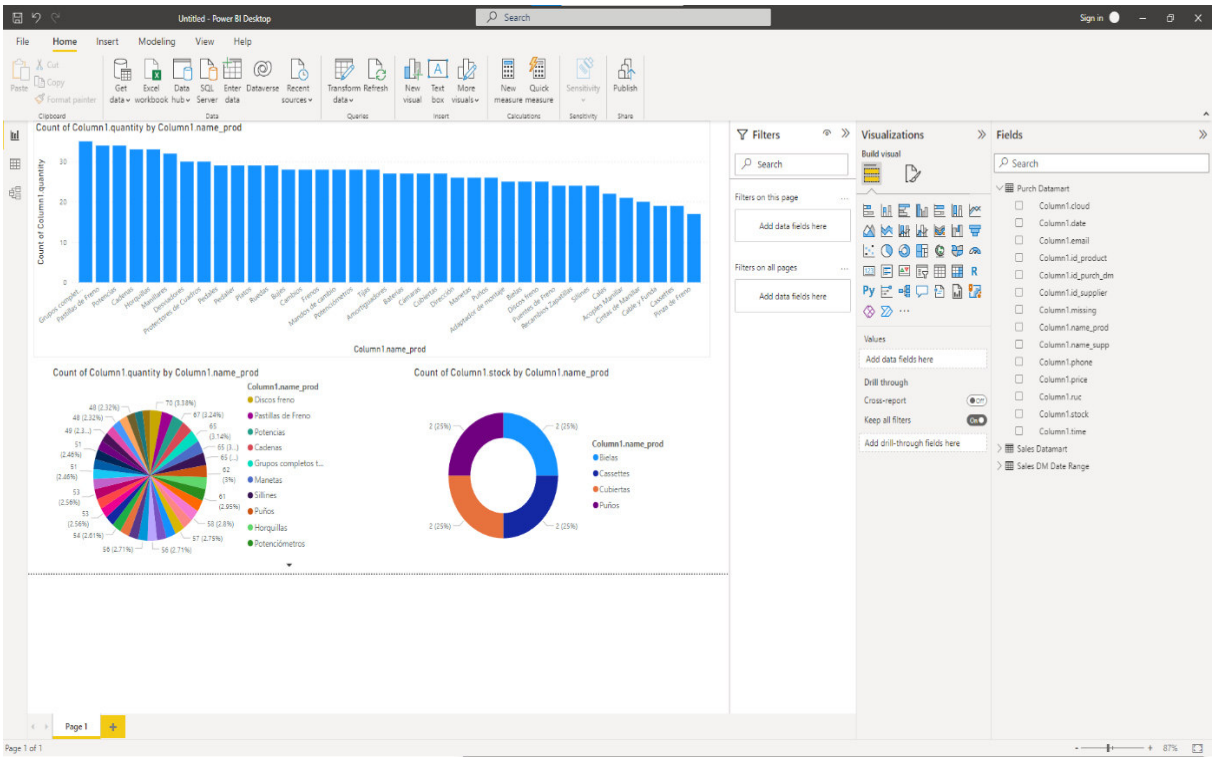


Figura 73. Gráficos y datos en Power BI.

Fuente: Elaboración propia.

3.4.3.4. Integración del ambiente para la creación del CDW

En esta etapa, se describe el desarrollo donde se tienen las Nubes 1, 2 y 3, que contienen tablas, estas usan otros servicios de AWS: "*Funciones Lambdas*" donde va el código para consultar y escribir en ellas. Adicionalmente, se tiene una API Gateway para poder exponerlas, acceder desde afuera a las "*Funciones Lambdas*", y sus tablas. Por otra parte, el ETL es una API REST que consta de dos partes.

- Parte 1: Un script que se ejecuta autónomamente desde que arranca el servicio cada cierta hora parametrizada lee las tablas de origen de datos, las procesa y llena las tablas de los "*Data marts*" en la base de integración como se puede ver Figura 84.
- Parte 2: Las rutas de acceso para poder consultar las tablas de los "*Data marts*" de ventas y compra.

a) Los VPS Cloud de AWS son los del plan "*Lightsail*". De esta forma, todo el código está escrito en JavaScript usando Node.Js y el VPS del ETL es sobre Linux, así como el de Power BI, con Windows. A continuación, en la Figura 74 se presenta la interfaz de Amazon DynamoDB donde se crearon las tablas de la fase de planeación.

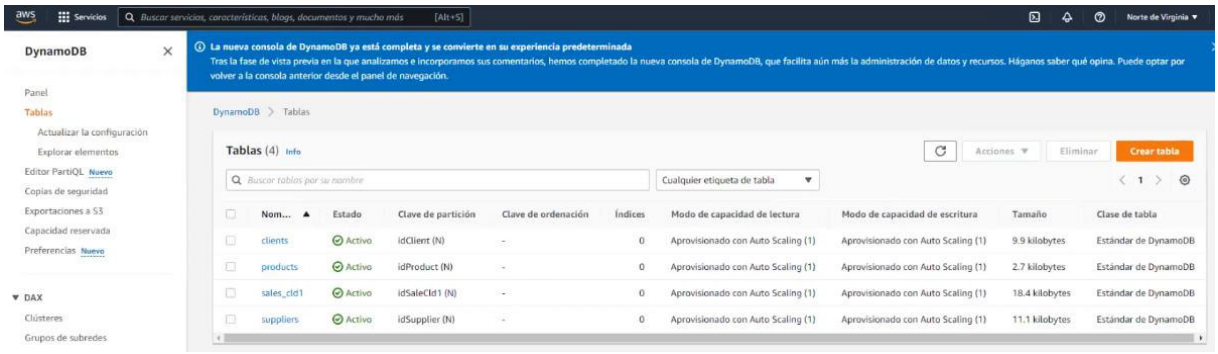


Figura 74. Tablas en DynamoDB.

Fuente: Elaboración propia.

b) Una vez creado Lambda y las Nubes correspondientes se procede a la creación de las funciones, las cuales se componen de dos por cada tabla. A continuación, se presenta una de esas funciones:

1. Función llamada getProductsMissing, la cual se trae todos los productos que presentan un stock en mínimo como se muestra en la Figura 75.

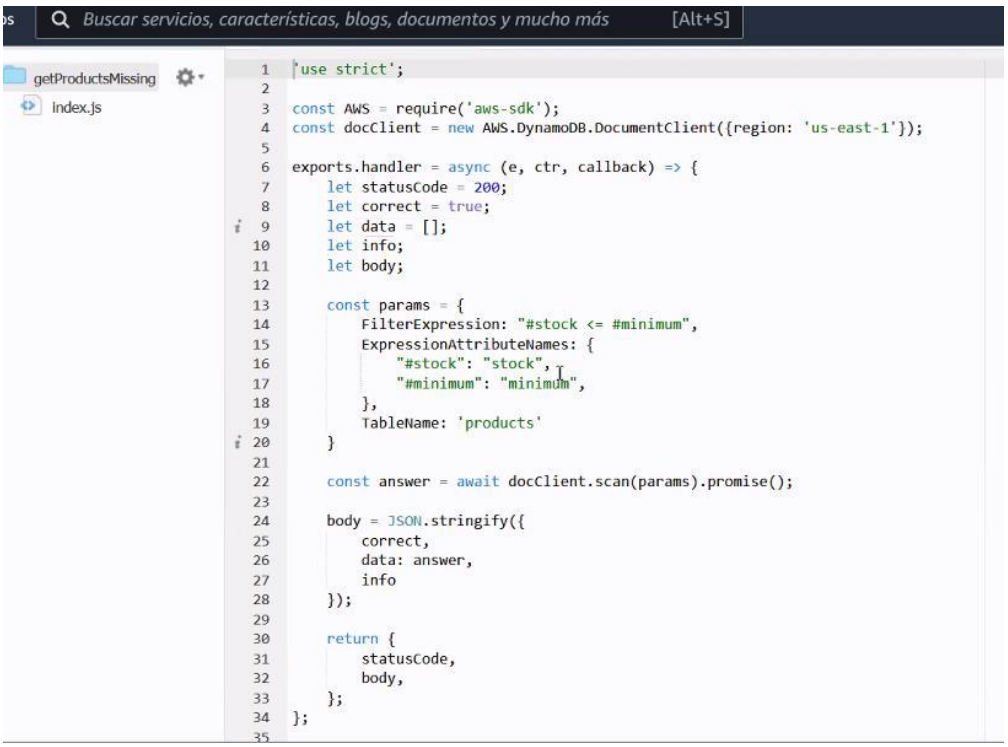


Figura 75. Función Lambda getProductsMissing.

Fuente: Elaboración propia.

c) Códigos de la función creada dentro de Amazon Lambda que sirve para consultar clientes como se presenta en la Figura 76.

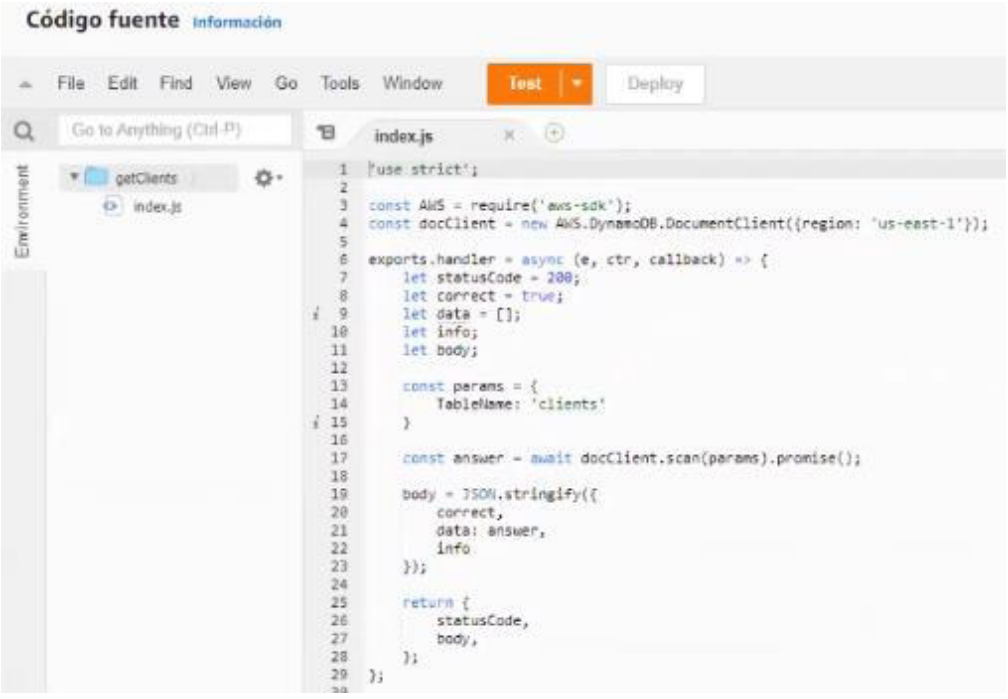


Figura 76. Código de funciones en Amazon Lambda.

Fuente: Elaboración propia.

d) Funciones GET y POST creadas dentro de Amazon Lambda como se presenta en la Figura 77.

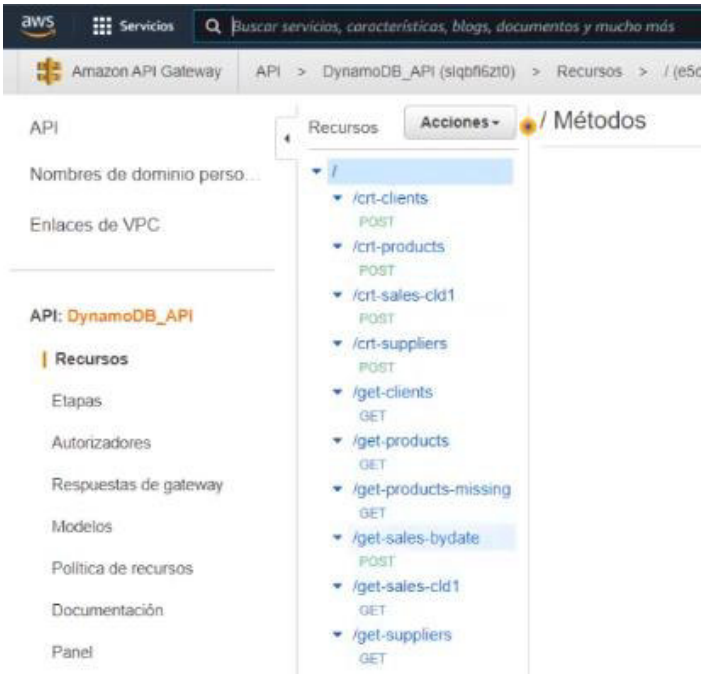


Figura 77. API de Amazon API Gateway.

Fuente: Elaboración propia.

e) Los Data mart fueron diseñados para el proyecto. A continuación, se presentan algunos de los códigos creados en este:

1. Código para presentar las compras realizadas como se presenta en la Figura 78

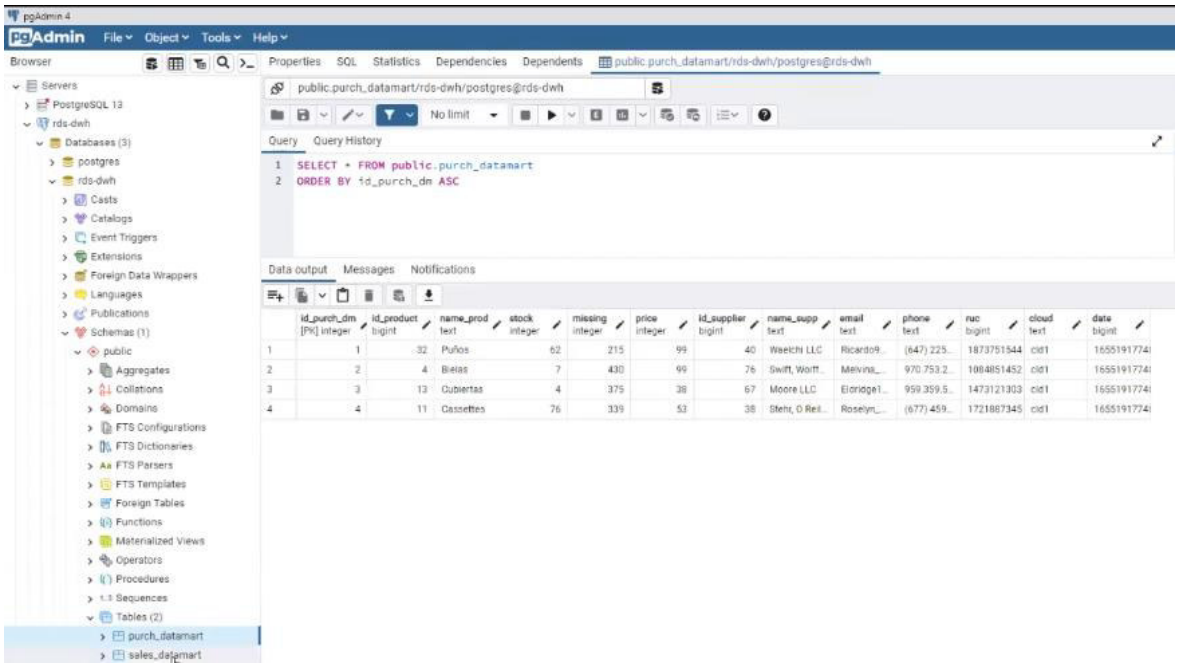


Figura 78. Data mart de compras.

Fuente: Elaboración propia.

2. Código para presentar las ventas como se presenta en la Figura 79.

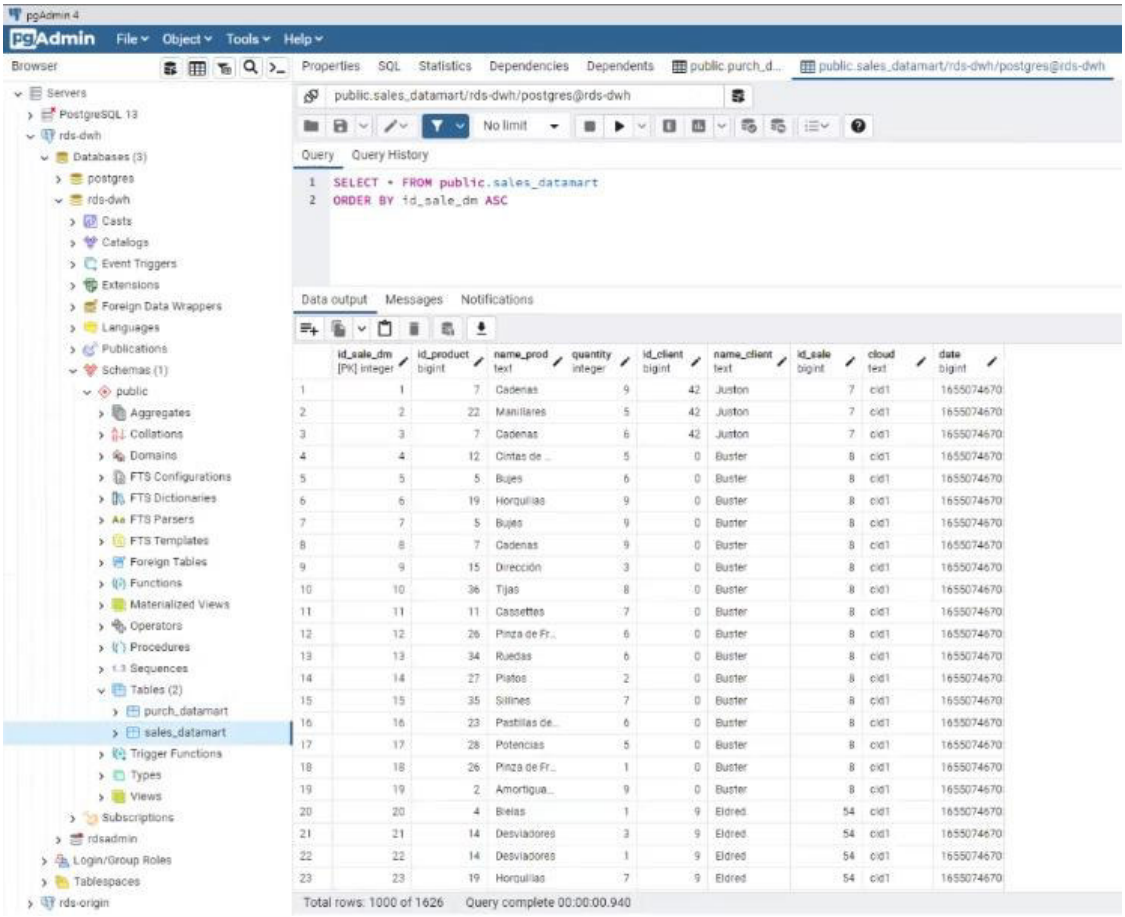


Figura 79. Data mart de ventas.

Fuente: Elaboración propia.

f) Concluida todas las construcciones de las Nubes y las configuraciones, ya se puede comenzar a realizar codificaciones para trabajar sobre ellas. A continuación, se presentan algunas de las funciones empleadas dentro de la implementación del CDW:

a) Codificación para realizar la conexión con la base de datos como se presenta en la Figura 80.

```
const { Pool } = require('pg');

const pool = (configDB) => {
    const poolConnect = new Pool(configDB);
    return poolConnect;
};
```

Figura 80. Función de conexión.

Fuente: Elaboración propia.

b) Configuración de la base de conexión que se utiliza en la función de conexión para indicar con cuáles datos se debe conectar como se presenta en la Figura 81.

```
const dataConfig = {
    server: {
        host: 'localhost',
        port: 3000,
        ssl: false
    },
    dynamoDB: {
        host: 'slqbfi6zt0.execute-api.us-east-1.amazonaws.com/prod',
        port: '',
        ssl: true
    },
    rdsOrigin: {
        host: 'rds-origin.c95hpa6sx8dq.us-east-1.rds.amazonaws.com',
        port: 5432,
        user: 'postgres',
        password: `jhonn_mejia2022`,
        database:'rds-origin'
    },
    rdsDwh: {
        host: 'rds-dwh-thesis.cgs44rw0tj1a.us-east-1.rds.amazonaws.com',
        port: 5432,
        user: 'postgres',
        password: `jhonn_mejia2022`,
        database:'rds-dwh'
    },
    protocol: 'http',
    processConfig: {
        cycleTime: 86400000
    }
};

module.exports = dataConfig;
```

Figura 81. Configuración de la base de conexión.

Fuente: Elaboración propia.

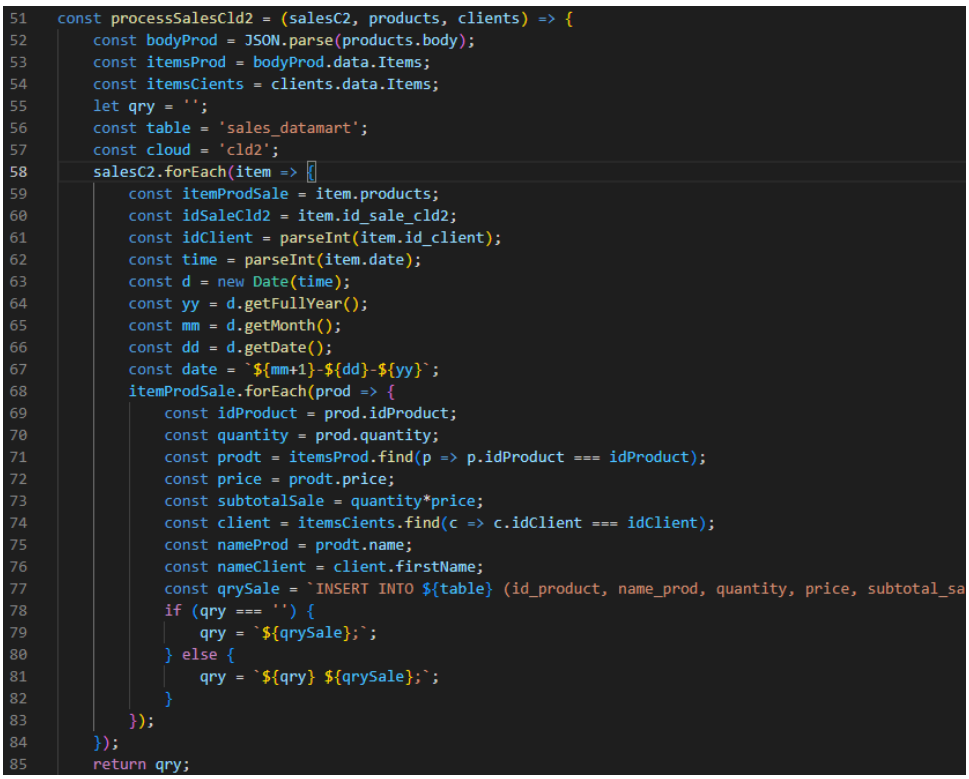
c) Función de procesamiento de la Nube 1 como se observa en la Figura 82, donde toma los datos de la Nube, como: ventas, productos y clientes, luego son integrados por medio de otra función como un único registro en la base de datos integración.

```
const processSalesCld1 = (salesC1, products, clients) => {
    const bodyC1 = JSON.parse(salesC1.body);
    const itemsC1 = bodyC1.data.Items;
    const bodyProd = JSON.parse(products.body);
    const itemsProd = bodyProd.data.Items;
    const itemsCients = clients.data.Items;
    let qry = '';
    const table = 'sales_datamart';
    const cloud = 'cld1';
    itemsC1.forEach(item => {
        const itemProdSale = item.products;
        const idSaleCld1 = item.idSaleCld1;
        const idClient = item.idClient;

        const time = item.date;
        const d = new Date(time);
        const yy = d.getFullYear();
        const mm = d.getMonth();
        const dd = d.getDate();
        const date = `${mm+1}-${dd}-${yy}`;

        itemProdSale.forEach(prod => {
            const idProduct = prod.idProduct;
            const quantity = prod.quantity;
            const prodt = itemsProd.find(p => p.idProduct === idProduct);
            const price = prodt.price;
            const subtotalSale = quantity*price;
            const client = itemsCients.find(c => c.idClient === idClient);
            const nameProd = prodt.name;
            const nameClient = client.firstName;
            const qrySale = `INSERT INTO ${table} (id_product, name_prod, quantity, price, subtotal_sale, id_
            if (qry === '') {
                qry = `${qrySale};`;
            } else {
                qry = `${qry} ${qrySale};`;
            }
        });
    });
    return qry;
}
```

Figura 82. Función de procesamiento de la Nube1.

Fuente: Elaboración propia.

d) Función de procesamiento de la Nube 2 como se presenta en la Figura 83. donde toma los datos de la Nube, como ventas, productos y clientes, los cuales luego son integrados por medio de otra función como un único registro en la base de datos integración.

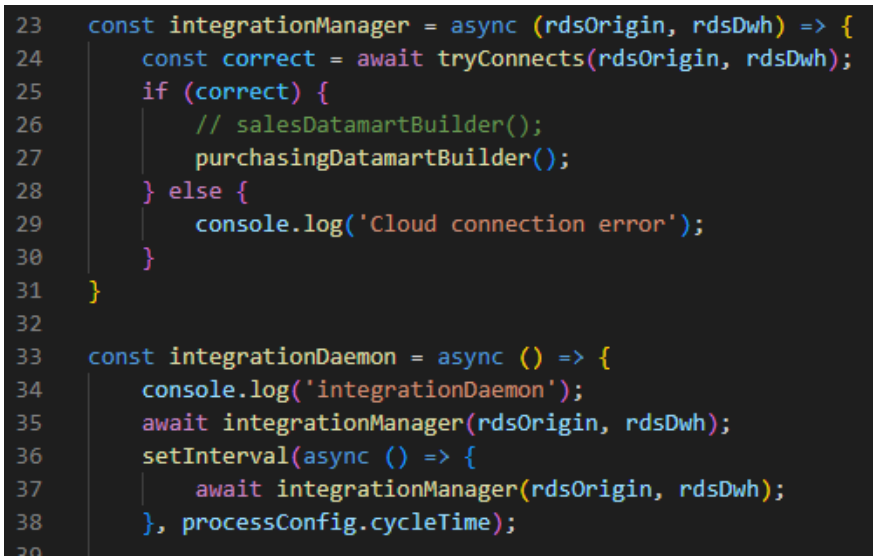


```
const processSalesCld2 = (salesC2, products, clients) => {
    const bodyProd = JSON.parse(products.body);
    const itemsProd = bodyProd.data.Items;
    const itemsClients = clients.data.Items;
    let qry = '';
    const table = 'sales_datamart';
    const cloud = 'cld2';
    salesC2.forEach(item => {
        const itemProdSale = item.products;
        const idSaleCld2 = item.id_sale_cld2;
        const idClient = parseInt(item.id_client);
        const time = parseInt(item.date);
        const d = new Date(time);
        const yy = d.getFullYear();
        const mm = d.getMonth();
        const dd = d.getDate();
        const date = `${mm+1}-${dd}-${yy}`;
        itemProdSale.forEach(prod => {
            const idProduct = prod.idProduct;
            const quantity = prod.quantity;
            const prodt = itemsProd.find(p => p.idProduct === idProduct);
            const price = prodt.price;
            const subtotalSale = quantity*price;
            const client = itemsClients.find(c => c.idClient === idClient);
            const nameProd = prodt.name;
            const nameClient = client.firstName;
            const qrySale = `INSERT INTO ${table} (id_product, name_prod, quantity, price, subtotal_sa
            if (qry === '') {
                qry = `${qrySale};`;
            } else {
                qry = `${qry} ${qrySale};`;
            }
        });
    });
    return qry;
```

Figura 83. Función de procesamiento de la Nube2.

Fuente: Elaboración propia.

e) Función llamada "*integración*" que utiliza el ETL para realizar la composición de los datos de las Nubes 1 y 2 y los carga en la Nube 3. Esta función se ejecuta de forma automática con un parámetro de temporización para cargar la nueva información que contengan las tablas de las dos sucursales como se presenta en la Figura 84.



```
const integrationManager = async (rdsOrigin, rdsDwh) => {
    const correct = await tryConnects(rdsOrigin, rdsDwh);
    if (correct) {
        // salesDatamartBuilder();
        purchasingDatamartBuilder();
    } else {
        console.log('Cloud connection error');
    }
}

const integrationDaemon = async () => {
    console.log('integrationDaemon');
    await integrationManager(rdsOrigin, rdsDwh);
    setInterval(async () => {
        await integrationManager(rdsOrigin, rdsDwh);
    }, processConfig.cycleTime);
```

Figura 84. Función integración.

Fuente: Elaboración propia.

3.4.4. Estabilización

Una vez concluida la fase anterior se procede a realizar las pruebas de la infraestructura, evidenciando la correcta ejecución de todos los procesos y se continua con la fase de implantación del proyecto.

3.4.5. Implantación

En esta fase se procedió a presentar el desarrollo a la empresa que solicitó la elaboración del CDW, por medio de una reunión con la directiva. En esta, se realizó la demostración por medio de los informes de las consultas y los paneles desarrollados en Power BI para la presentación de los datos relevantes de la compañía.

3.5. Evaluación y despliegue

En el capítulo 4 se ejecutan estas fases de la metodología en la cual se analizan los resultados obtenidos.

4. RESULTADOS Y DISCUSIÓN

A continuación, se presentan los resultados obtenidos a través del diseño, planificación e implementación del CDW para la empresa Bikes Extreme Ecuador.

A través de la implementación del CDW para la empresa Bikes Extreme Ecuador del presente proyecto, se logró la integración de los datos de las dos sucursales que maneja la empresa, facilitando el manejo de la información de forma efectiva y segura. Esto permitió un control más exhaustivo de los inventarios, ventas de productos, organización de los proveedores, así como la obtención de datos de los clientes. También facilitó por medio de los datos la toma de decisiones y la planificación estratégica en posibles planes de expansión. Seguidamente, se presentan los objetivos establecidos para la puesta en marcha del CDW:

4.1.1. Mejorar la eficiencia en las ventas y la calidad de prestación de los servicios

Por medio del CDW se logró obtener un gráfico de porcentajes de ventas de productos, el cual proporciona valiosa información a los directivos de la empresa. Esta muestra de forma intuitiva la información, permitiendo realizar filtros dinámicos que presenta los datos de los productos más buscados por los compradores y que permiten establecer relaciones en base a eventos, actividades deportivas, maratones y otros acontecimientos de índole deportiva que pudieran ser aprovechados y que evidencian los datos presentados por el reporte de Power BI que se presenta en la Figura 85.

Nombre Prod.	Cantidad Venta	Precio	Sub Total Venta	Nombre Client.	Fecha
Amortiguadores	8	70	560	Nina	6-12-2022
Amortiguadores	8	70	560	Winston	6-12-2022
Amortiguadores	9	70	630	Buster	6-12-2022
Amortiguadores	9	70	630	Caleb	6-12-2022
Amortiguadores	9	70	630	Hector	6-12-2022
Amortiguadores	9	70	630	Madelyn	6-12-2022
Amortiguadores	9	70	630	Mafalda	6-12-2022
Amortiguadores	9	70	630	Rigoberto	6-12-2022
Baterías	1	16	16	Erick	6-12-2022
Baterías	1	16	16	Lee	6-12-2022
Baterías	10	16	160	Alyson	6-12-2022
Baterías	10	16	160	Marquis	6-12-2022
Baterías	10	16	160	Mellie	6-12-2022
Baterías	10	16	160	Royce	6-12-2022
Baterías	10	16	160	Sigmund	6-12-2022

Figura 85. Reporte porcentajes de ventas de productos en Power BI.

Fuente: Elaboración propia.

4.1.2. Establecer políticas que faciliten la satisfacción de los clientes

Para el cumplimiento de este objetivo, se presenta a la directiva de la empresa un reporte de las ventas por clientes de forma gráfica. Sobre la base de esta información, es posible implementar políticas de bonificaciones especiales por compras en determinados productos o emplear políticas de comunicación, que permitan informar al cliente de productos nuevos. Esta también facilita proporcionar recomendaciones sobre la existencia de ofertas especiales y fomentar la fidelidad de los clientes, todo por medio de los datos obtenidos y que se presentan en la Figura 86.

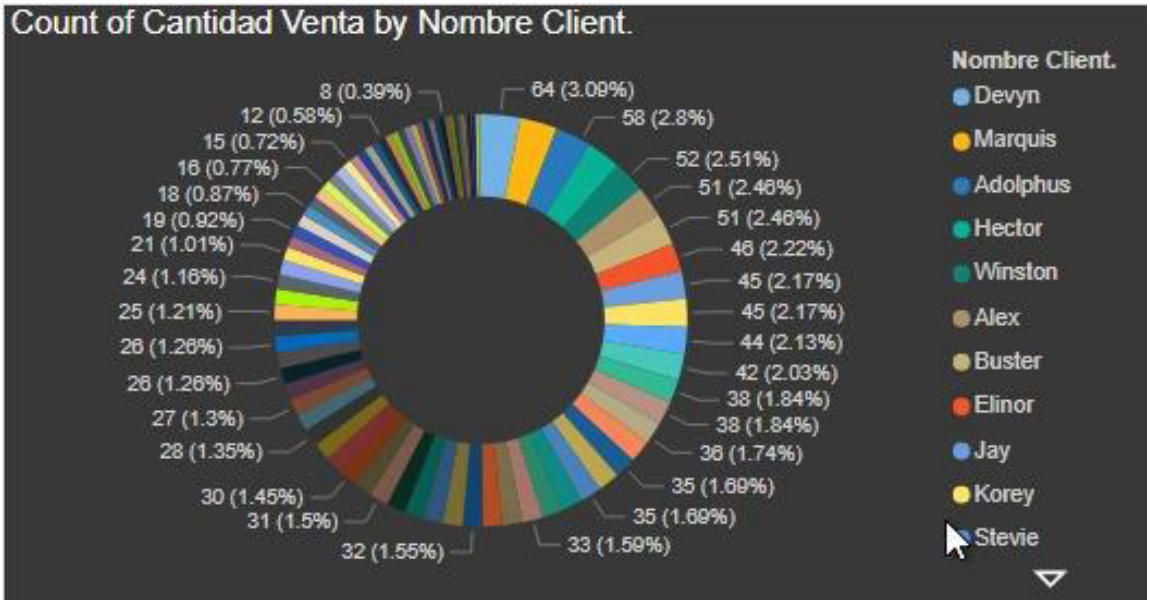


Figura 86. Reporte gráfico de ventas por clientes en Power BI.

Fuente: Elaboración propia.

4.1.3. Integrar sistemas computarizados para la mejora continua de la empresa

Los datos de la empresa se presentan de forma gráfica, intuitiva y con controles de filtrado de las ventas totales y porcentaje por sucursal. Esta funcionalidad gráfica va a permitir a la directiva de la empresa poder manipular la información para su fácil entendimiento, proporcionando gran poder de visualización y entendimiento del negocio. Además, favorecerá una adecuada planificación y visión del negocio, también contribuirán en los planes de mejora continua de la empresa. En la Figura 87 se presenta el reporte antes mencionado, tomado en cuenta que la empresa consta de dos sucursales, Sucursal 1 llamada cld1 (color celeste), sucursal 2 llamada cld2 (color amarillo).

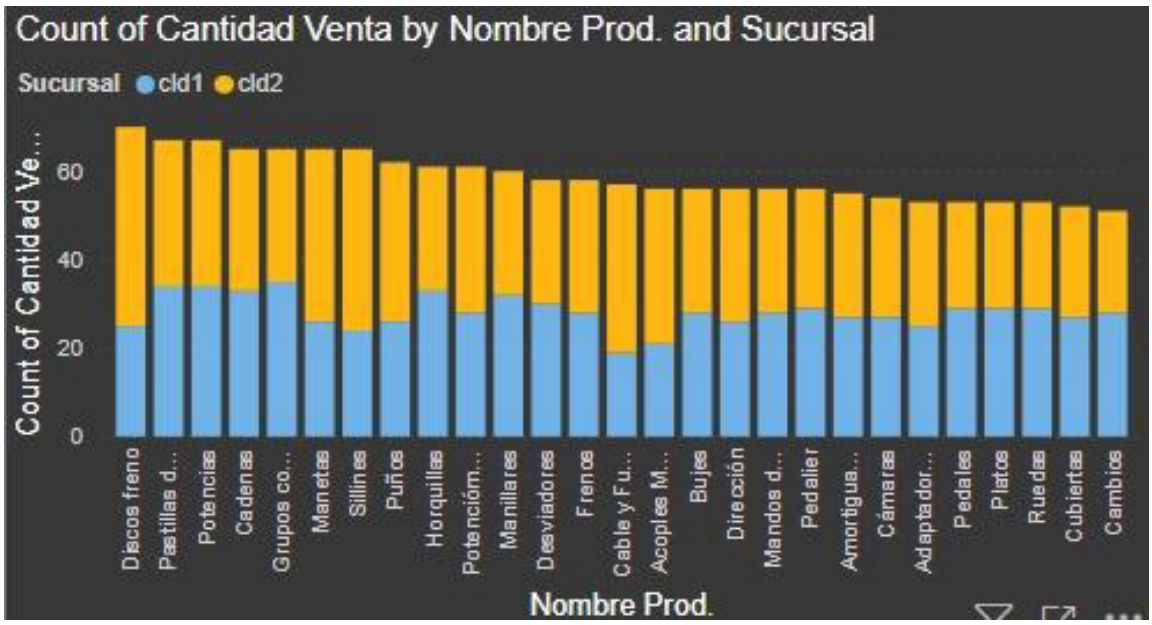


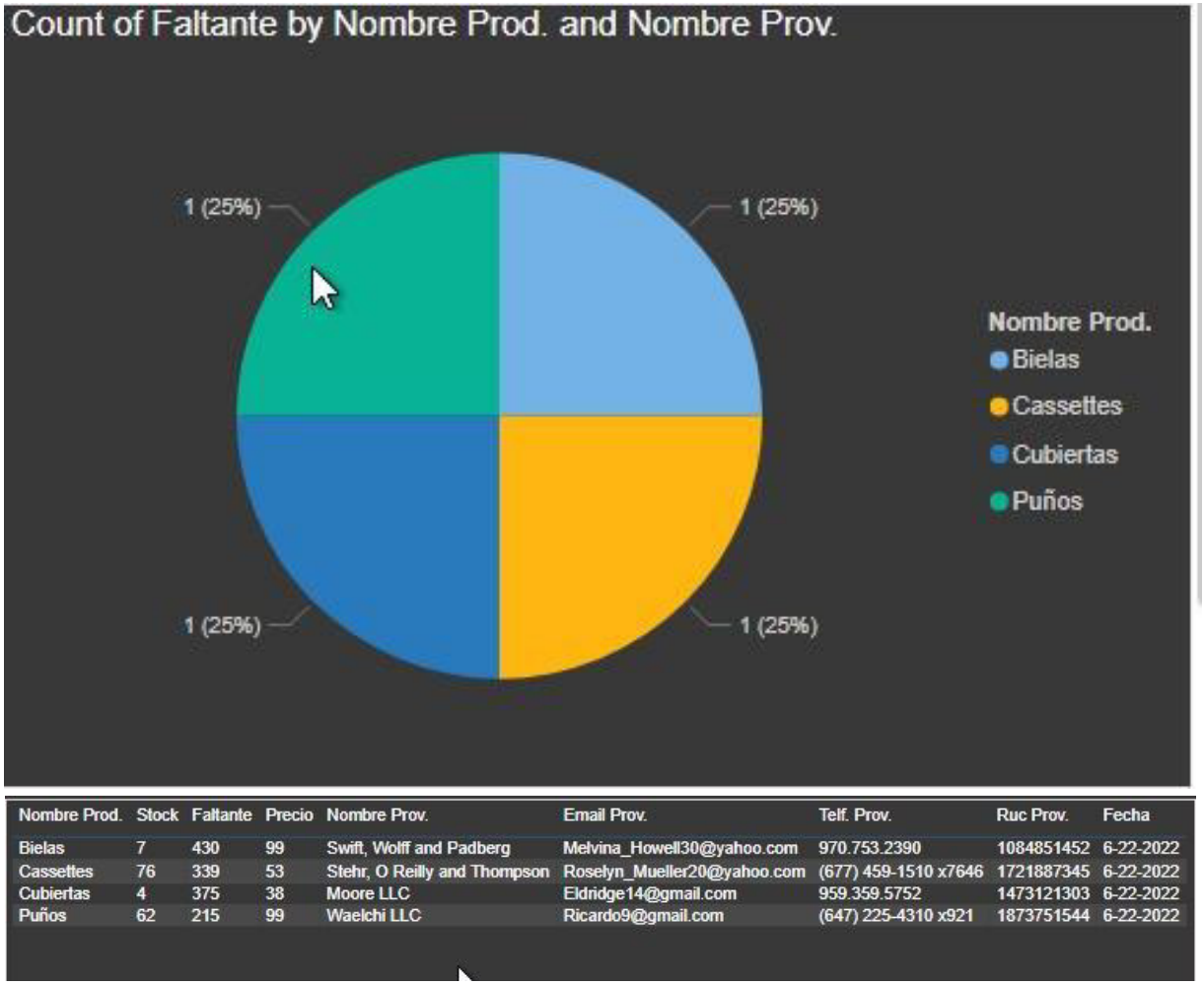
Figura 87. Ventas totales y porcentaje por sucursal en Power BI.

Fuente: Elaboración propia.

4.1.4. Establecer un sistema de toma de decisiones entorno a la adquisición de productos nuevos y productos existentes

Entre los objetivos del estudio, se buscó plasmar a través de los gráficos información relevante para la toma de decisiones, entorno a productos nuevos o existentes. En el reporte mostrado en la Figura 88 es posible visualizar el stock de la empresa y establecer cuáles son los nuevos productos en los cuales la directiva puede invertir. La información que emerge de la interpretación de los gráficos les permite proyectar estrategias con respecto al mercado, como es el caso de la comercialización de Bicicletas Bicycle Motocross (BMX)[2] u otro tipo de proyecciones de mercado que los ejecutivos deseen tomar dependiendo de los datos y el estudio de estos.

[2] Bicicletas BMX: Son bicicletas deportivas todoterreno que se utiliza para carreras y acrobacias



Nombre Prod.	Stock	Faltante	Precio	Nombre Prov.	Email Prov.	Telf. Prov.	Ruc Prov.	Fecha
Bielas	7	430	99	Swift, Wolff and Padberg	Melvina_Howell30@yahoo.com	970.753.2390	1084851452	6-22-2022
Cassettes	76	339	53	Stehr, O Reilly and Thompson	Roselyn_Mueller20@yahoo.com	(677) 459-1510 x7646	1721887345	6-22-2022
Cubiertas	4	375	38	Moore LLC	Eldridge14@gmail.com	959.359.5752	1473121303	6-22-2022
Puños	62	215	99	Waelchi LLC	Ricardo9@gmail.com	(647) 225-4310 x921	1873751544	6-22-2022

Figura 88. Stock por productos en Power BI.

Fuente: Elaboración propia.

5. CONCLUSIONES Y RECOMENDACIONES

En esta sección se muestran las conclusiones que se obtuvieron de las fases, tales como: planificación, comprensión, preparación, modelado, evaluación e implantación para la puesta en funcionamiento de CDW de este proyecto.

5.1. Conclusiones

El proyecto logró realizar un estudio del área de CDW que pudo proporcionar información relevante y que contribuyó significativamente con el diseño de la arquitectura empleada por el desarrollo del proyecto, así como su puesta en funcionamiento y pruebas del mismo.

Para el éxito de un proyecto de CDW es indispensable identificar las necesidades del negocio para de esta forma proponer un diseño de arquitectura de servidores en la Nube que se acople a estas necesidades.

Para asegurar una adecuada funcionalidad del CDW es necesario identificar los elementos fundamentales en el diseño de la infraestructura requerida en base a las necesidades de la empresa en estudio.

Una adecuada herramienta de visualización permite representar los datos recolectados en un formato entendible y fácil de comprender para los usuarios y directivos de las empresas.

La fase de prueba de los elementos de CDW representan un factor importante, ya que permiten la comprobación del buen funcionamiento de todos los elementos que interactúan entre sí y que una vez integrados pueden proporcionar datos interesantes y adecuados dependiendo de las necesidades de análisis y puntos de interés a ser evaluados por los expertos en datos de la empresa.

Los reportes generados a través de Power BI representan una efectiva herramienta que permite presentar la información de forma gráfica y con un fácil entendimiento para los gerentes y directivos de empresas, los cuales pueden obtener mucho beneficio de estos datos por ser una fuente fundamental de información de la empresa.

5.2. Recomendaciones

En esta sección se muestran las recomendaciones derivadas de las fases, tales como: planificación, comprensión, preparación, modelado, evaluación e implantación para la puesta en funcionamiento de CDW de este proyecto.

Se recomienda:

- Realizar un buen análisis de datos para el desarrollo de un CDW, esto con el propósito de identificar de forma precisa las dimensiones, así como la fuente de los datos, ya que estos van a proporcionar información importante para la toma de decisiones una vez culminada la implantación.
- Establecer con claridad cuál será el propósito del CDW para seleccionar de forma correcta el proveedor de servicios donde desplegará la infraestructura necesaria para el proyecto.
- Disponer de un apoyo técnico con el objeto de identificar de forma adecuada los requerimientos y que estos estén alineados con lo que posteriormente estará plasmado en el diseño del CDW.
- Definir con claridad los datos que se almacenaran dentro de las tablas para facilitar así la segmentación y filtrado al momento de trabajar con la herramienta Power BI, ya que, de no definir objetivamente estos datos, la etapa de diseño en la herramienta de visualización puede ser un poco confusa.
- Crear los reportes en Power BI lo más sencillos posibles y con los datos relevantes para ser comprendido de forma sencilla por los directivos, debido a que si se recarga con muchos datos estos pueden llegar a confundir a la persona que interpreta la información en el reporte.
- Disponer de una buena conexión a internet para poder interactuar con las Nubes en la herramienta AWS, ya que es fundamental durante la etapa de diseño, para evitar que no exista demasiada latencia y pueda afectar el buen desarrollo del proyecto.
- Adquirir servicios más robustos de pago en la fase de pruebas del CDW en algunas ocasiones es necesario, ya que puede no ser suficiente la capacidad gratuita que proporciona la plataforma Amazon AWS.

REFERENCIAS BIBLIOGRÁFICAS

Alavandhar, J., & Nikiforova, O. (2017). Several Ideas on Integration of SCRUM Practices within Microsoft Solutions Framework. *Applied Computer Systems*, *21*(5), 71–79. https://sciendo.com/pdf/10.1515/acss-2017-0010

Amazon Redshift. (2022). *AWS | Solución de almacenamiento y análisis de datos en la Nube*. https://aws.amazon.com/es/redshift/?nc2=h_m1

Amron, M. T., Ibrahim, R., & Chuprat, S. (2017). A Review on Cloud Computing Acceptance Factors. *Procedia Computer Science*, *124*, 639–646. https://doi.org/10.1016/j.procs.2017.12.200

Assiroj, priati. (2021). *Data Warehouse & Data Mining* (1st ed., Vol. 1). Politeknik Imigrasi. https://www.researchgate.net/publication/360822489_DATA_WAREHOUSE_DATA_MINING#pf5

AWS. (2022). *Capa gratuita de AWS | Cloud computing*. https://acortar.link/85qVQ9

Azure. (2021). *¿Qué es IaaS? Infraestructura como servicio | Microsoft Azure*. https://azure.microsoft.com/es-es/overview/what-is-iaas/#overview

Baker, O., & Thien, C. (2020). A New Approach to Use Big Data Tools to Substitute Unstructured Data Warehouse. *2020 IEEE Conference on Big Data and Analytics, ICBDA 2020*, 26–31. https://doi.org/10.1109/ICBDA50157.2020.9289757

Diaz, J., & Matta, M. (2020). *Sistema de recomendación automático de servicios Multi-cloud* [Universidad Icesi]. https://doi.org/10.6084/m9.figshare.13661018.v2

Efendi, T. F., & Krisanty, M. (2020). Warehouse Data System Analysis PT. Kanaan Global Indonesia. *International Journal of Computer and Information System (IJCIS)*, *1*(3), 70–73. https://doi.org/10.29040/IJCIS.V1I3.26

Eklund, M. (2019). *Marcus Eklund Data Warehousing in the Cloud – Analysis of an Implementation Project Title: Data Warehousing in the Cloud-Analysis of an Implementation Project* [Åbo Akademi University (Tesis de maestría)]. https://www.doria.fi/bitstream/handle/10024/177841/eklund_marcus.pdf?sequence=2&isAllowed=y

FileZilla. (2022). *FileZilla - The free FTP solution*. https://filezilla-project.org/

Garani, G., Chernov, A. V., Savvas, I. K., & Butakova, M. A. (2019). A Data Warehouse Approach for Business Intelligence. *Proceedings - 2019 IEEE 28th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2019*, 70–75. https://doi.org/10.1109/WETICE.2019.00022

Gartner. (2021). *Gartner Says Cloud Will Be the Centerpiece of New Digital Experiences*. Newsroom. https://acortar.link/lWuBuE

Google Cloud. (2022). *Servicios de computación en la Nube*. https://acortar.link/0h2ROz

Harvy, I., Matitaputty, G., Girsang, A., Michael, S., & Isa, S. (2019). The Use of Book Store GIS Data Warehouse in Implementing the Analysis of Most Book Selling. *2019 7th International Conference on Cyber and IT Service Management, CITSM 2019*. https://doi.org/10.1109/CITSM47753.2019.8965404

ITU. (2012). ITU-T. FG Cloud TR Version 1.0 Part 1: Introduction to the cloud ecosystem: definitions, taxonomies, use cases and highlevel. *International Telecommunication Union*. https://acortar.link/HERspX

MarkLogic. (2022). *Get Started with a Fully Managed Cloud Data Hub - MarkLogic*. https://www.marklogic.com/product/getting-started/

Martinez, F., Contreras, L., Ferri, C., Hernandez, J., Kull, M., Lachiche, N., Ramirez, M., & Flach, P. (2021). CRISP-DM Twenty Years Later: From Data Mining Processes to Data Science Trajectories. *IEEE Transactions on Knowledge and Data Engineering*, *33*(8), 3048–3061. https://doi.org/10.1109/TKDE.2019.2962680

Mejías, H. (2018). *Modelo de minería de datos para la identificación de patrones que influyen en la mora de la Cooperativa de Ahorro y Crédito San José S.J.* [Pontificia Universidad Católica del Ecuador (Tesis de Maestría))]. https://repositorio.pucesa.edu.ec/handle/123456789/2435

Microsoft. (2022). *Visualización de datos | Microsoft Power BI*. https://Power BI.microsoft.com/es-es/

Nodejs. (2022). *Node.js*. https://nodejs.org/es/

Oracle ADW. (2022). *Autonomous Data Warehouse | Oracle*. https://www.oracle.com/autonomous-database/autonomous-data-warehouse/

pgAdmin. (2022). *pgAdmin - Herramientas PostgreSQL*. https://www.pgadmin.org/

Postman. (2022). *Postman API Platform*. https://www.postman.com/

Rodríguez de la Cruz, A. (2020). *Herramienta para despliegue y gestión de plataformas en la Nube* [Universidad de Sevilla]. https://idus.us.es/handle/11441/108955

SAP. (2022). *SAP Data Warehouse Cloud | Integración y analíticas de datos*. https://www.sap.com/latinamerica/products/data-warehouse-cloud.html

VSC. (2022). *Visual Studio Code - Code Editing. Redefined*. https://code.visualstudio.com/