



Proyecto Atenea: AI para mejorar el acceso a la información de las áreas de Business Intelligence e Information Technology de Comercial Card S.A.S.

Yesika Milena Carvajal Díaz

Modalidad de Prácticas de Semestre de Industria

Asesor Interno

Luis Germán García Morales, Ph. D. Ingeniería Electrónica

Universidad de Antioquia

Facultad de Ingeniería

Ingeniería Electrónica

Medellín

2024

Cita	Carvajal Díaz [1]
Referencia Estilo IEEE (2020)	[1] Y. Carvajal Díaz, “Proyecto Atenea: AI para mejorar el acceso a la informacion de las áreas de Business Intelligence e Information Technology de Comercial Card S.A.S.”, Semestre de industrial, Ingeniería Electrónica, Universidad de Antioquia, Medellín , 2024.



Repositorio Institucional: <http://bibliotecadigital.udea.edu.co>

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda.

Decano/Director: Julio César Saldarriaga.

Jefe departamento: Eduardo Emiro Rodríguez Ramírez.

El contenido de esta obra corresponde al derecho de expresión de los autores y no compromete el pensamiento institucional de la Universidad de Antioquia ni desata su responsabilidad frente a terceros. Los autores asumen la responsabilidad por los derechos de autor y conexos.

RESUMEN

Este informe presenta un enfoque detallado sobre la creación de modelos de AI en las áreas de BI e IT, analizando las clasificaciones de tickets y su correspondiente exactitud. En la sección de BI, se destaca la priorización de clasificaciones y la relevancia de la sensibilidad de la información al considerar la automatización de procesos. Se optó por no automatizar solicitudes que involucraban datos sensibles, mientras que se implementaron tableros de Power BI para facilitar la gestión de información sobre ventas. En la sección de IT, se observó que la mayoría de los tickets ya contaban con automatizaciones preexistentes, con un enfoque en mejorar la comunicación entre las áreas de IT y BI para la correcta gestión de solicitudes. A pesar de las limitaciones en la automatización, se ofrecieron soluciones estadísticas y recomendaciones para mejorar el flujo de trabajo. Este informe no solo identifica áreas de mejora en la gestión de tickets, sino que también establece estrategias que pueden ser adoptadas para fomentar la eficiencia operativa en la organización.

ABSTRACT

This report shows a detailed approach to the creation of AI models in the BI and IT areas, analyzing ticket classifications and their corresponding accuracy. In the BI section, the prioritization of classifications and the relevance of data sensitivity when considering process automation are highlighted. Requests involving sensitive data were not automated, while Power BI dashboards were implemented to facilitate the management of sales information. In the IT section, it was observed that most tickets already had pre-existing automations, with a focus on improving communication between the IT and BI areas for the proper handling of requests. Despite the limitations in automation, statistical solutions and recommendations were provided to improve the workflow. This report not only identifies areas for improvement in ticket management but also establishes strategies that can be adopted to promote operational efficiency within the organization.

CONTENIDO

1. Introducción	4
2. Objetivos	5
3. Marco teórico y trabajos relevantes	6
3.1. Trabajos relevantes	6
3.2. Marco conceptual	11
4. Metodología	13
4.1. Requerimientos funcionales y no funcionales	13
4.2. Evaluación y selección de algoritmos de AI	14
4.3. Desarrollo de modelo de AI para clasificación de tickets	16
4.3.1. Clasificación de tickets	16
4.3.2. Creación del modelo de AI	22
4.4. Automatización de procesos	30
5. Resultados y análisis	36
5.1 Desarrollo de modelo de AI para clasificación de tickets	36
5.1.1 Clasificación de tickets	36
5.1.2 Creación del modelo de AI	37
5.2. Automatización de procesos	41
6. Conclusiones	52
7. Referencias bibliográficas	56

1 INTRODUCCIÓN

En un entorno empresarial cada vez más dinámico y competitivo, la mejora de los procesos mediante la inteligencia artificial (AI) se ha convertido en un factor crucial para mejorar la eficiencia operativa y la toma de decisiones. Este informe se centra en la implementación de un modelo de AI para la clasificación de tickets en las áreas de Business Intelligence (BI) e Information Technology (IT) de la empresa Comercial Card S.A.S.

La necesidad de automatizar el proceso de clasificación de solicitudes surgió a partir de la creciente demanda de atención al cliente y la complejidad de las peticiones recibidas. Los antecedentes del estudio revelan que la manualidad en el procesamiento de tickets genera retrasos y errores, afectando la satisfacción del cliente y la productividad del equipo. A través de la integración de las APIs de Google Cloud Vertex AI, se buscó abordar este problema mediante la creación de un modelo de AI capaz de clasificar solicitudes de manera precisa y eficiente.

Los alcances de este estudio abarcan el diseño, desarrollo y evaluación del modelo de AI. Se establecieron limitaciones, entre ellas la necesidad de ajustar constantemente los parámetros del modelo, para lograr una apropiada precisión. Además, el informe documenta los cambios implementados para mejorar su rendimiento.

La metodología empleada incluye un enfoque iterativo en la creación del modelo, donde se analizaron y ajustaron los parámetros a partir de una matriz de incertidumbre y la exactitud alcanzada en las clasificaciones. Este proceso involucró la recolección de datos de tickets previos, la priorización de clasificaciones y el análisis de resultados obtenidos tras cada modificación.

El significado de este estudio radica en su contribución al avance del campo de la inteligencia artificial aplicada en la gestión de solicitudes. La automatización de la clasificación de tickets no solo mejora el uso de recursos humanos, sino que también mejora la capacidad de respuesta de la empresa ante las necesidades del cliente, estableciendo un modelo escalable que puede adaptarse a futuras demandas del mercado. La aplicación de este modelo en Comercial Card S.A.S. establece un precedente para la implementación de soluciones de AI en otros ámbitos, impulsando así la innovación y la eficiencia operativa en el sector.

2 OBJETIVOS

2.1 *Objetivo general*

Desarrollar un sistema que permita la clasificación y respuesta automática de los tickets recibidos en las áreas de Business Intelligence (BI) e Information Technology (IT), con el propósito de mejorar los tiempos de gestión y respuesta de los empleados. Este sistema empleará algoritmos de Inteligencia Artificial disponibles a través de las APIs de Google Cloud Vertex AI, utilizando el lenguaje de programación Python como principal herramienta de desarrollo.

2.2 *Objetivos Específicos*

- Definir los requerimientos funcionales y no funcionales que faciliten la selección de los algoritmos de AI y la construcción de un modelo para la clasificación de tickets y la generación de respuestas automáticas.
- Evaluar varios algoritmos de AI y seleccionar dos que realicen adecuadamente la clasificación y generación de respuestas automáticas de los tickets recibidos, de acuerdo con los criterios y requerimientos establecidos, para poder obtener la herramienta adecuada para la clasificación de los tickets.
- Desarrollar el modelo para la clasificación de los tickets, utilizando la herramienta de Vertex AI y la descripción dada por los solicitantes en los tickets, teniendo como base los algoritmos de AI seleccionados anteriormente, para lograr una clasificación apropiada de los tickets.
- Desarrollar y evaluar un programa en Python que implemente el modelo propuesto junto con los algoritmos de AI seleccionados, para la clasificación y respuesta automática de tickets.

3 MARCO TEÓRICO Y TRABAJOS RELEVANTES

En esta sección, se describen una serie de conceptos y terminología necesarios para comprender el desarrollo de este proyecto. Además, se introducen algunos trabajos relevantes que se han tomado como base del estado del arte para el desarrollo.

3.1 Trabajos relevantes

3.1.1 Automatización del Proceso de Solicitudes de Activos Tecnológicos en el Banco de Crédito Centroamericano [1]

El objetivo del proyecto es automatizar el proceso de solicitudes de activos tecnológicos en el Banco de Crédito Centroamericano. En este proyecto se pretende estudiar la eficiencia del procedimiento para la realización de las solicitudes de activos tecnológicos de Bancentro antes de realizar el proyecto, y realizar el análisis y diseño de una aplicación Web para una automatización de este procedimiento. Además, se busca desarrollar una base de datos para la aplicación Web con Microsoft SQL Server.

En el desarrollo de la aplicación se utilizaron las siguientes herramientas de software:

- Gestor de base de datos: Microsoft SQL Server 2005 Edición Estándar.
- IDE de desarrollo: Microsoft Visual Studio 2008 (ASP.net).
- Lenguaje de programación: C#, HTML, Java Script.
- Controles de tercero para aplicación: Infragistics 2009.1 para ASP.net

La arquitectura de desarrollo utilizada para la elaboración de la aplicación fue programación orientada a objeto en N-capas, la cual permite que gran parte del código de esta aplicación sea reutilizable y que el sistema sea escalable. Se encapsuló el código de la aplicación en las capas de la figura 1.

Según el estudio realizado para determinar la eficiencia del procedimiento para la realización de las solicitudes, es que el tiempo total promedio de cada solicitud es de 10.3 días, en donde gran parte de este tiempo es consumido por la gestión y autorización de la documentación en forma manual, dado que estas gestiones se realizan trasladando la documentación de la solicitud de manera impresa de oficina en oficina hasta cumplir con todo el ciclo del proceso (figura 2).

Como conclusión, se obtuvo que, por medio del desarrollo del sistema, se logró automatizar el proceso de solicitudes de activos tecnológicos, permitiendo al personal involucrado en la gestión del proceso reducir los tiempos de respuestas, obteniendo información en tiempo real acerca de la etapa en la que se encuentran las solicitudes realizadas y reduciendo los costos de la gestión del proceso.

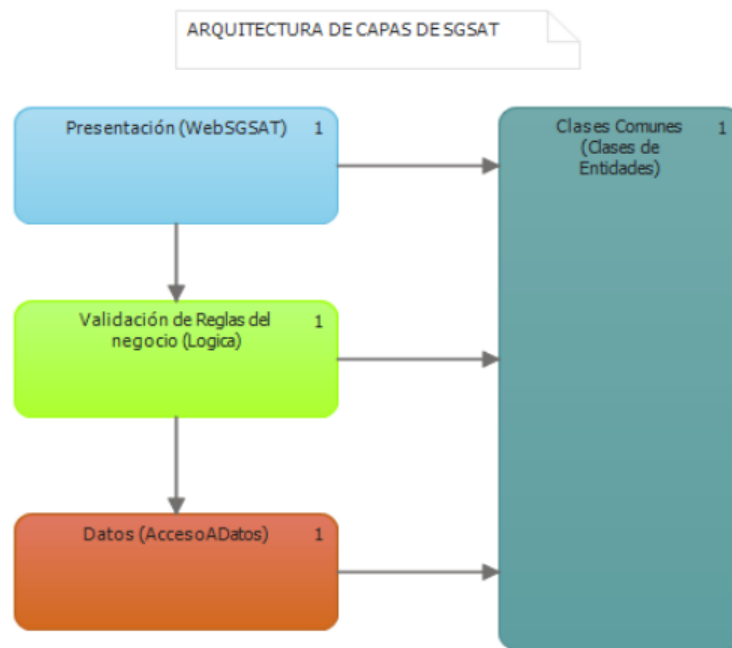


Figura 1. Diagrama de arquitectura de desarrollo del proyecto Automatización del Proceso de Solicitudes de Activos Tecnológicos en el Banco de Crédito Centroamericano [1].

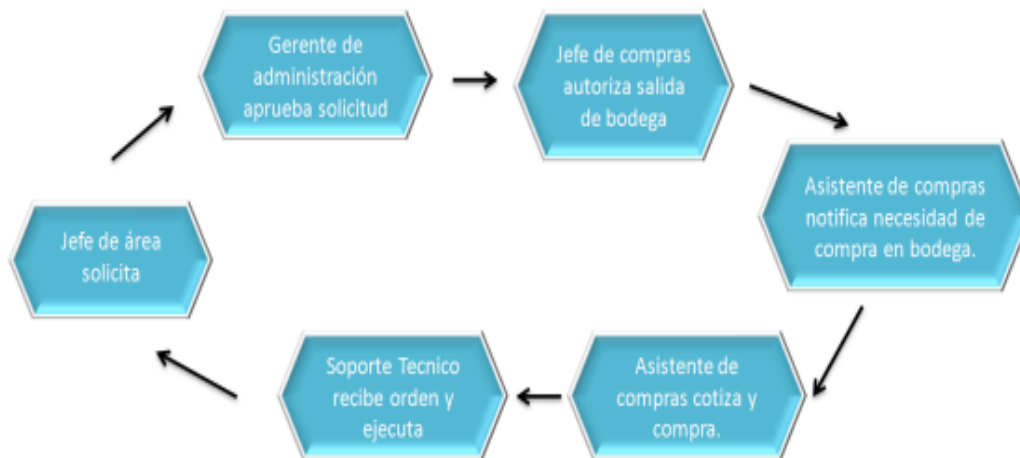


Figura 2. proceso de solicitud de activos informáticos en el Banco de Crédito Centroamericano desde los inicios del área administrativa [1].

Además, se recomienda capacitar a todos los usuarios involucrados en el uso del sistema, proceder con la implementación a nivel nacional una vez finalizado el tiempo de prueba y validación del sistema, posteriormente ampliar el sistema para que gestione todas las solicitudes de activos del banco, e incorporar la aplicación y la base de datos del sistema al plan de contingencia ya existente en la empresa.

3.1.2 Automatización Solicitudes Pedidos y Compras Proyecto Corona [2]

Corona es una multinacional colombiana con 140 años de historia empresarial dedicada a la manufactura y comercialización de productos para el hogar, la construcción, la industria, la agricultura y el sector de energía.

El proyecto busca estudiar las solicitudes que llegan al área de TI, y con este estudio determinar la ruta a seguir para la automatización de las solicitudes.

Antes de la realización del proyecto, se encontraban procesos manuales repetitivos, por lo que se realiza el proyecto pensando en clasificar estos procesos.

En el proyecto, primeramente, se hizo el clustering de las solicitudes (figura 3).

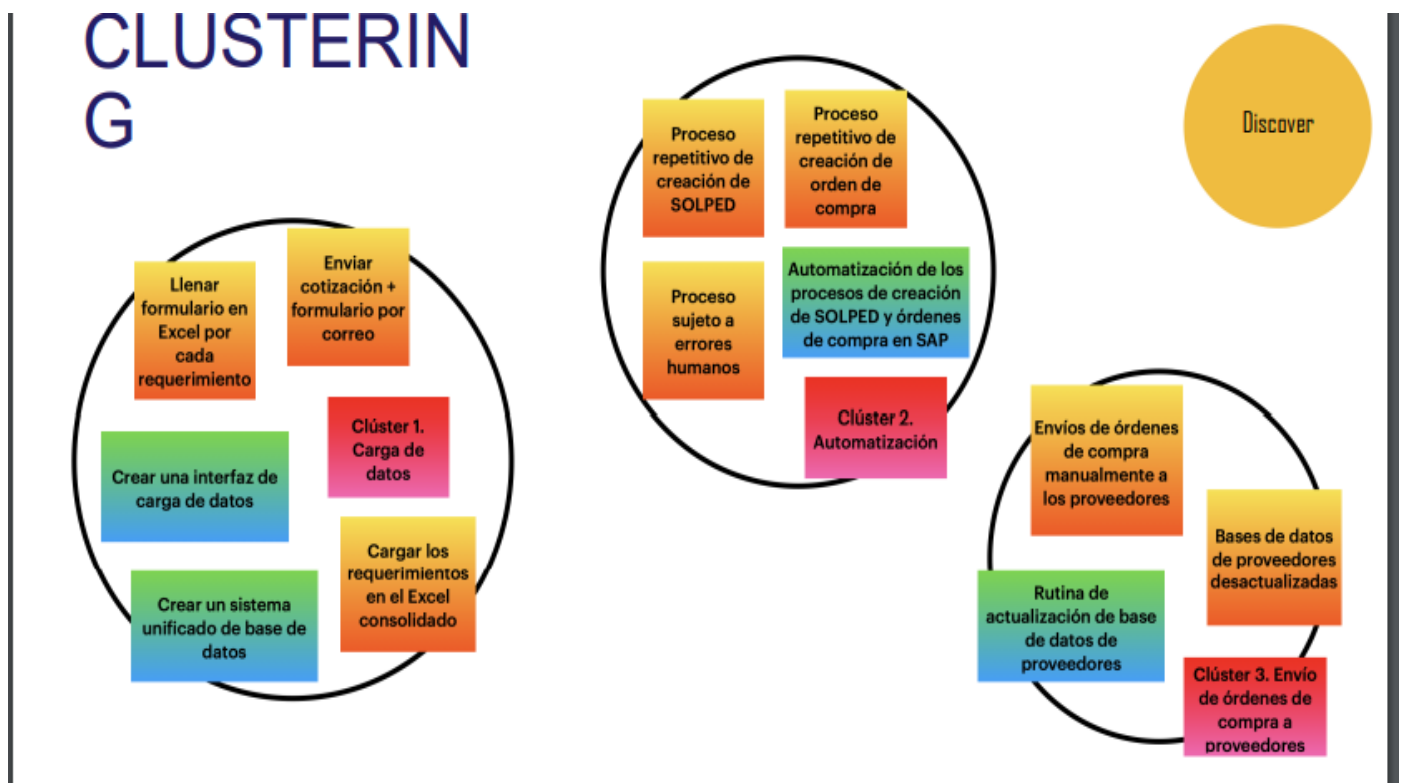


Figura 3. Solución de la realización del clustering de las solicitudes [2].

Además, se realizó un mapa de stakeholder (figura 4) para entender la gestión de las solicitudes con el personal involucrado en la realización de estas en la empresa.

Para la automatización de estas solicitudes, se hizo uso de IRPA's (Intelligent Robotic Process Automation), con las cuales la solución se manda a los solicitantes por correo electrónico.

Después de implementar el proyecto, se hizo la validación con un asesor de tecnología y con un usuario final.

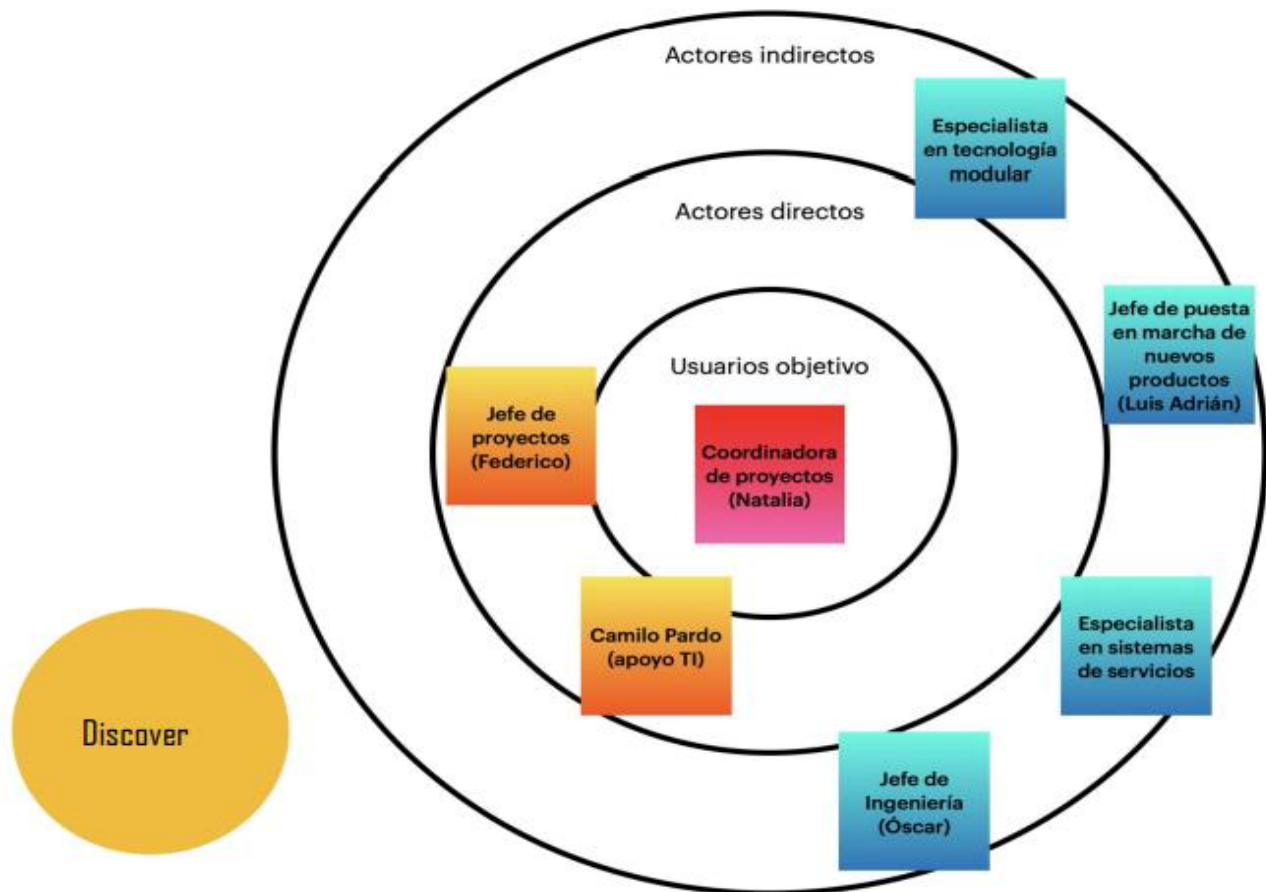


Figura 4. Mapa de Stakeholder para las solicitudes del proyecto B [2].

En la figura 5, se observan los pasos a seguir para dar respuesta a las solicitudes. Se encuentra en rojo los pasos que se realizaron en el proyecto.

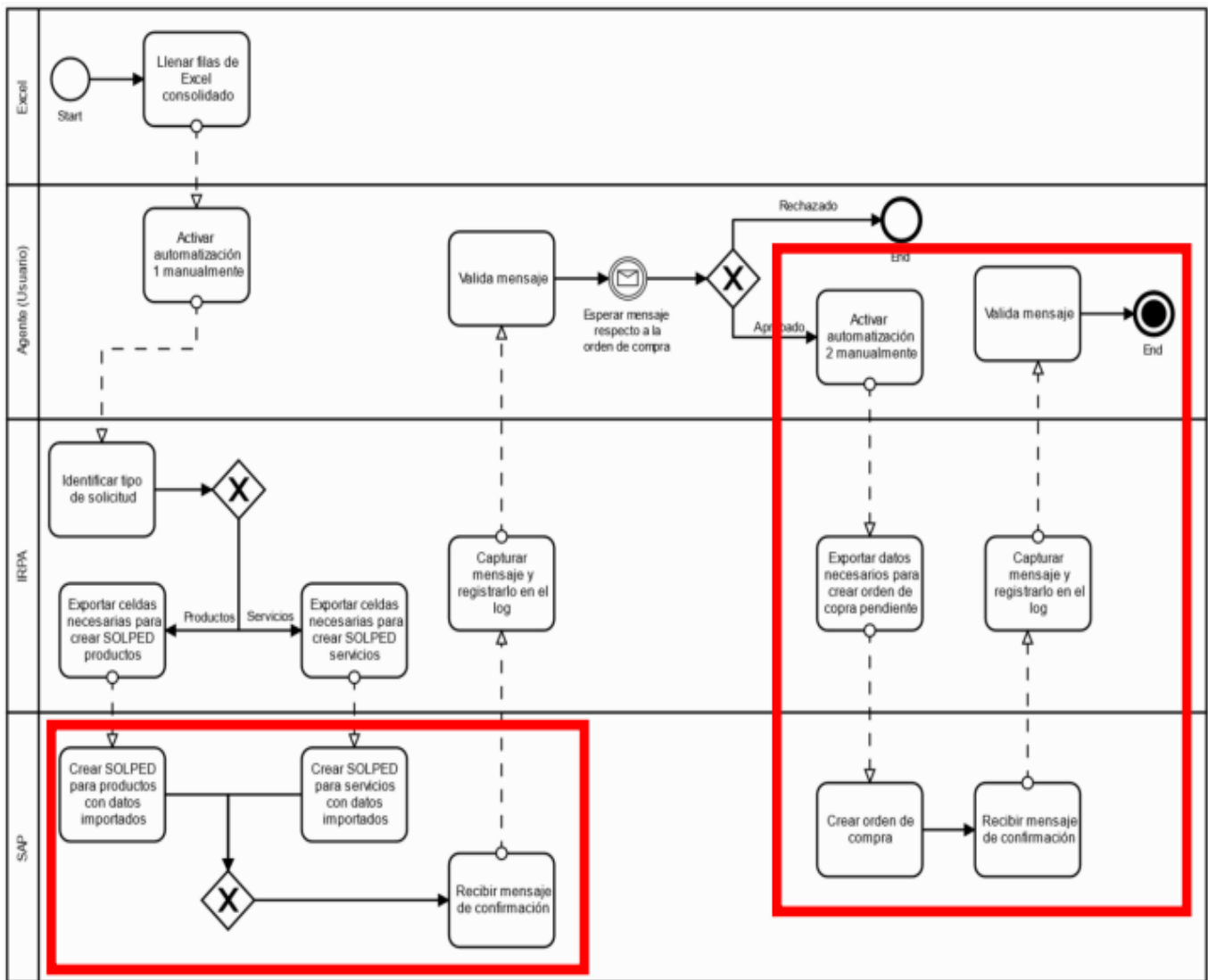


Figura 5. Ruta de ejecución de las solicitudes, con rutas de estudio y realización del proyecto [2].

3.1.3 Automatización de la clasificación de solicitudes que recibe el área de soporte interno de Agestic [3]

El sistema de gestión de solicitudes y tickets de Agestic requiere de asignación manual de diversas etiquetas (grupo, sitio, clasificación, sub- clasificación, tipo de ticket, técnico, etc.) Con base al contenido de la solicitud. El proyecto “Automatización de la clasificación de solicitudes que recibe el área de soporte interno de Agestic” tiene como objetivo automatizar la asignación de dichas etiquetas utilizando modelos de IA que infieran dicha asignación a partir de las descripciones recibidas. Además, debe cerrar el ticket y notificar automáticamente con una plantilla predeterminada. Esto es para mejorar los tiempos y formas de asignación de solicitudes y tickets que llegan a Soporte Agestic, y se utiliza el procesamiento de lenguaje natural (NPL), la predicción de alertas tempranas, el Deep Learning, y el lenguaje de programación Python.

El modelo toma la decisión sin interacción con una persona (a excepción de quien toma el ticket) y luego esta persona puede cambiar la etiqueta.

Como consideraciones finales se encontraron que el sistema contiene un código de integración específico para atender las solicitudes. Para ello, se desarrolló un webhook integrado con la API específica de ServiceDesk, y que se está migrando a MLOPS de RedHat.

3.3 Marco conceptual

Para la realización del proyecto presentado, es necesario definir primeramente los requerimientos funcionales, los cuales se concentran en responder a preguntas como “¿Qué características debe tener el software?” o “¿Qué tareas debe realizar?”. Por otro lado, los requisitos no funcionales se ocupan de aspectos de calidad, rendimiento y experiencia del usuario, la eficiencia, la seguridad, la usabilidad, la escalabilidad y otros factores que determinan la calidad y el rendimiento del software [4].

Una vez definidos los requerimientos funcionales y no funcionales, se hace una evaluación y una selección de los algoritmos de AI para poder obtener la herramienta adecuada para la clasificación de los tickets. Posteriormente, se utiliza el lenguaje Python para cargar el modelo de AI y realizar la iteración por cada ticket para su clasificación, junto con la API Google Cloud Vertex AI.

La AI es una subdisciplina del campo de la informática enfocada en la creación de agentes inteligentes. Estos son sistemas capaces de analizar, aprender y actuar de manera autónoma [5]. La AI puede ser predictiva o generativa. La inteligencia artificial predictiva y la generativa tienen objetivos y enfoques distintos. La AI generativa parte de crear cosas nuevas y originales inspirados en lo que ya existe. Aprende patrones y estructuras de datos, y luego genera nuevo contenido similar. La AI predictiva intenta predecir el futuro con las tendencias. Analiza datos del pasado para identificar patrones, y luego usa esos patrones para hacer predicción [6]. Para la programación de los modelos de IA, se pueden emplear diversos lenguajes. Un lenguaje ampliamente utilizado para el desarrollo de aplicaciones que involucran IA es Python. Python es un lenguaje de programación ampliamente utilizado en las aplicaciones WEB, el desarrollo de software, la ciencia de datos y el machine learning ML (los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes [7]). Una manera de usar modelos de AI en Python es mediante plataformas que se pueden conectar usando código y permisos. Una de estas plataformas que se puede conectar con Python es la de Google Cloud Vertex AI. Vertex AI es una plataforma de aprendizaje automático (AA) que permite entrenar además de implementar modelos de AA y aplicaciones de AI, y personalizar modelos de lenguaje grande (LLM) para usarlos en las aplicaciones con tecnología de AI [8].

Durante el desarrollo de este trabajo, se realizó un diagrama de flujo y un diseño de diagrama de arquitectura de software para que los empleados tengan un entendimiento de la tecnología usada, además de servir como guía de orientación. El diagrama de flujo es definido como aquella

representación gráfica del algoritmo o proceso. A través de los diagramas de flujo se logra representar esquemáticamente las ideas y los conceptos en relación, donde generalmente se emplea a la hora de especificar de una forma gráfica los algoritmos [9]. El diseño de diagrama de arquitectura de software consiste en elaborar un diagrama de la arquitectura, donde se crean representaciones visuales de los componentes de un sistema de software. En un sistema de software, el término arquitectura se refiere a las funciones establecidas, sus implementaciones y las interacciones entre sí. Dado que el software es inherentemente abstracto, los diagramas de arquitectura ilustran visualmente los diversos flujos de datos dentro del sistema [10].

Muchos tickets requieren solo de la búsqueda en las bases de datos, mientras que otros requieren información específica, con lo que se requiere el uso de la plataforma de Google Cloud BigQuery. Google Cloud BigQuery es una plataforma de análisis de datos completamente administrada y lista para la IA que ayuda a maximizar el valor de los datos. Se diseñó para ser multimotor, multiformato y de múltiples nubes [11]. Las búsquedas en esta plataforma se generan por medio de código SQL, el cual se puede ejecutar por medio de lenguaje Python. SQL significa lenguaje de consulta estructurado. Es un lenguaje de programación diseñado para gestionar los datos almacenados en un sistema de gestión de bases de datos relacionales RDBMS [12].

Utilizando el lenguaje de programación de Python también se automatizan las demás labores necesarias para la generación de la solución del ticket, como sería la búsqueda de parámetros específicos, el filtrado de la información o la depuración de los datos, por lo que con este lenguaje se implementa la automatización de los tickets, entendiéndose la automatización como el término que se utiliza para aplicaciones de tecnología en las que se minimiza la interacción humana [13]. Dentro del proceso de automatización tenemos también la depuración de los datos y el filtrado de la información. La depuración de los datos es el proceso por el que se modifican los datos de determinado sistema de almacenamiento para asegurar que se cuenta con la información correcta y precisa [14]. Por otro lado, el filtrado de la información, dentro del procesamiento de los datos, tiene como finalidad seleccionar aquellos fragmentos de la información que son relevantes, descartando lo que se considera ruido o información irrelevante [15]. Los filtros permiten restringir los datos que se ven a elementos de interés [16].

4 METODOLOGÍA

En esta sección se explora la metodología que se utilizó para el desarrollo de los objetivos específicos. La metodología se divide en cuatro sub-secciones, que corresponden a los objetivos específicos sobre requerimientos funcionales y no funcionales, evaluación y selección de algoritmos de AI, desarrollo de modelo de AI para clasificación de tickets y automatización de procesos. La última sub-sección está dividida en dos partes, una para el área de BI y la otra para el área de IT.

4.1 *Requerimientos funcionales y no funcionales*

Para definir los requerimientos funcionales y no funcionales, se debe tener un contexto y conocimiento de las labores que realiza la empresa, y cómo se manejan las áreas de interés, sus herramientas y su funcionamiento. Por esta razón, primeramente, se entendió el funcionamiento de la empresa, la estructura de sus bases de datos, y se definieron los planes de acción para el desarrollo de la práctica. Posteriormente, se definieron los requerimientos funcionales y no funcionales, los cuales se presenta a continuación:

4.1.1 *Requerimientos funcionales*

Con los programas en Python se debe:

- Clasificar con modelo para AI los tickets de las áreas de BI y de TI.
- Crear base de datos con la clasificación de todos los tickets (históricos) de las áreas de BI e IT que se encuentren en el CRM.
- Actualizar diariamente la base de datos de las clasificaciones de los tickets históricos con los tickets más recientes.
- Crear base de datos con la clasificación diaria de los tickets del área de BI que se encuentren en el CRM como activos y que se hayan solicitado desde hace 1 día.
- Actualizar 3 veces al día las bases de datos de las clasificaciones diarias de los tickets del último día.
- Contener como campos para todas las bases de datos anteriores la fecha de ingesta de cada registro, la fecha en la que se solicitó el ticket, el número identificador del ticket, la clasificación que se le dio con AI, y un resumen de descripción del ticket.

4.1.2 *Requerimientos no funcionales*

- Se debe hacer una documentación de los programas en cada función y clase del programa, y en el repositorio se debe decir qué versión se tiene de Python y dar una descripción de lo que hace cada programa.
- Los códigos que se realicen deben tener documentación según las normas PEP8, y para código en Python, el uso de documentación con librería docstrings.

4.2 Evaluación y selección de algoritmos de AI

Para realizar la selección de los algoritmos de AI, se realizó la arquitectura tecnológica del proyecto, teniendo en cuenta los requisitos funcionales y no funcionales. La arquitectura tecnológica se observa en la figura 6.

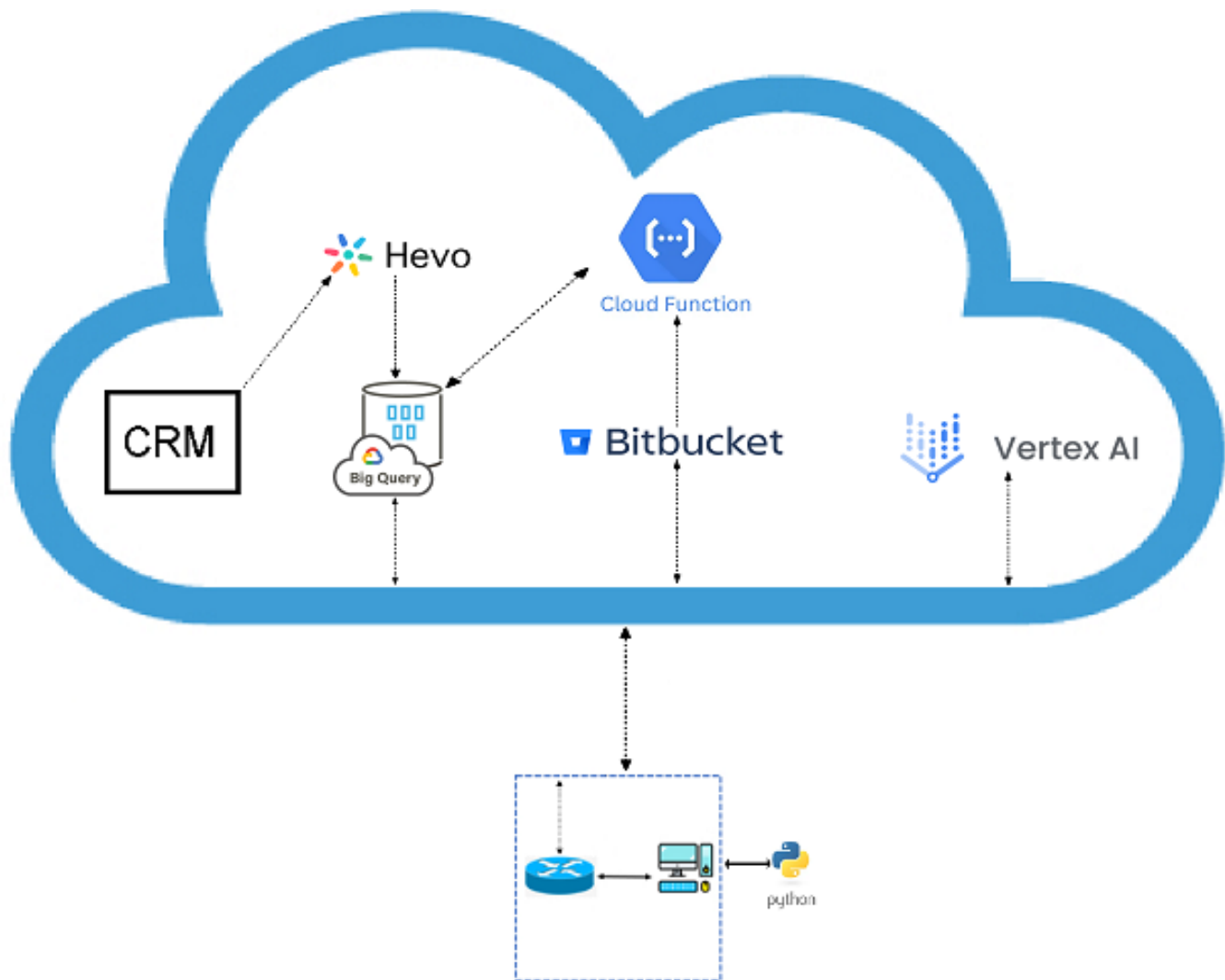


Figura 6. Arquitectura tecnológica planteada para el proyecto Atenea.

En la arquitectura tecnológica de la figura 6, se puede ver que para la red interna se necesita únicamente el programa Python. La red interna se conecta a la nube de forma bidireccional, ya que necesita darle instrucciones y prompts a Vertex AI, y obtener su respuesta, necesita modificar repositorios, y extraer también estos repositorios (pull), y necesita entregar una petición de consulta a BigQuery, obtener su resultado, y además crear nuevas tablas. Debido a esto, todas las conexiones que ingresan a la nube son bidireccionales. Otra conexión bidireccional es la de BigQuery con Function, ya que Function generará peticiones de SQL a las bases de datos solicitadas en los tickets, por lo que manda una petición a BigQuery y recibe su resultado. Los datos que se encuentran en el CRM de la empresa, se mandan a BigQuery por medio de Hevo,

por lo que las dos conexiones que hay entre estos tres servidores es unidireccional, partiendo en el CRM y terminando en BigQuery. La última conexión es unidireccional desde BitBucket hacia Function, ya que los códigos generados por Python se almacenan en BitBucket, y desde Function se da la ubicación de los repositorios en BitBucket para automatizar los códigos, pero Function no da ningún código ni otra información al repositorio ni al servidor en general de BitBucket.

Con la arquitectura tecnológica planteada, se llevó a cabo la evaluación de los algoritmos de AI.

La clasificación de tickets necesita ser una inteligencia que aprende patrones y estructuras de datos con los ejemplos dados para dar como respuesta la clasificación. No necesitamos la AI predictiva, ya que no necesitamos una predicción futura, sino una respuesta actual con datos similares, no con datos que siguen una función respecto al tiempo.

Sabiendo el tipo de AI que se debía implementar, se usaron las APIs de Google Cloud Vertex AI. Estas permiten crear modelos de clasificación con inteligencia artificial generativa que pueden ser orquestados desde Python usando la librería nativa de Google. Se encontraron tres tipos de APIs de AI generativa para la clasificación de los tickets. Estos fueron:

1. Claude 3.5 Sonnet [17].
2. Content classification [18].
3. Chat Vertex AI Studio Modelo gemini-1.5-flash-001 [19].

La primera API consiste en la generación de un código SQL por medio de una petición. En esta API se requiere que se ingrese el nombre de la tabla con su ubicación, los nombres de sus campos, y se haga una explicación de la tabla y de cada uno de sus campos.

En la segunda API se ingresan los tipos de tickets que el modelo tiene, y se dan ejemplos de los tickets ya clasificados. Después, se entrega el ticket a clasificar.

La tercera API es un chat abierto de AI. Esta API no es específica para un tema, sino que es un chat en el que se puede preguntar por cualquier tema en general. Con esta API se puede clasificar explicándole detalladamente a la AI qué es lo que se requiere.

Para utilizar la primera API, es necesario documentar aproximadamente 200 tablas, con sus respectivas columnas, lo cual representa una tarea compleja. Además, si la aplicación no funciona adecuadamente, se habrán invertido muchas horas en este proceso y será necesario redefinir las tablas hasta obtener un resultado con un nivel de precisión aceptable.

Para utilizar la segunda API, se requiere ingresar los ejemplos para la clasificación de los datos, lo cual se puede hacer con relativa facilidad, ya que los tickets se encuentran en una tabla. Esta tabla se puede pasar a formato Excel, se pueden clasificar en este software, y por medio de código de Python se puede almacenar en un .txt los ejemplos de las clasificaciones para entregarlos a la API como un prompt.

Para utilizar la tercera API, se requiere lo mismo que para la segunda API, pero con especificaciones adicionales para que la API comprenda claramente qué información recibe y

qué se solicita. Estas especificaciones se pueden agregar al archivo .txt que se utiliza como prompt para garantizar un funcionamiento adecuado.

Por la complejidad de la primera API, se descarta su uso. La utilización de la segunda y la tercera API no variaba mucho, lo importante era especificar bien para que no se presente un porcentaje de error alto. Se planea utilizar la tercera API, ya que los prompts son más baratos, y se tiene más libertad en las peticiones.

4.3 Desarrollo de modelo de AI para clasificación de tickets

Para desarrollar el modelo de AI primero se necesitó estudiar y clasificar manualmente todos los tickets de ambas áreas (clasificación histórica). De los tickets históricos, se eligieron algunos tickets los cuales se entregaron al modelo como ejemplo de lo que debía realizar. Después de realizar la clasificación, se desarrolló el modelo de AI, y posteriormente se usaron los tickets históricos para determinar la exactitud del modelo.

4.3.1 Clasificación de tickets

Para la realización de la clasificación de los tickets, se requiere información de los tickets anteriores. Esta información se obtiene de las bases de datos de la empresa. Para obtenerla, se deben realizar flujos de datos desde el CRM de la empresa y BigQuery por medio de Hevo. Esto se logró buscando los campos del código HTML de la página y programando el envío de la información de estos campos por medio de una tabla cruda en Hevo. Para facilitar la lectura de esta información, se realiza una vista de la tabla cruda, para que tenga los nombres de los campos de forma más entendible, y para que los datos estén mejor estandarizados. La ubicación de esta vista se observa en la figura 7, nombrada como tickets_rino, y en el apartado de bi_visualization. Los campos de la vista se observan en la figura 8.

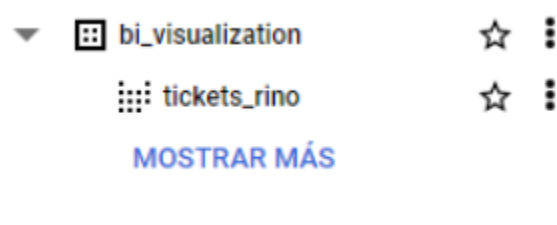


Figura 7. Ubicación de vista de tabla con información de los tickets.

Una vez se tuvo esta vista de la información de los tickets, se pudo acceder a la información de la vista de BigQuery por medio de consultas con código SQL, conectando el servidor de Google Cloud de BigQuery a Python con código y credenciales, y con esto consultar toda la información del histórico de los tickets. Teniendo la información de los tickets, se estudiaron y se buscó cómo clasificarlos. Después, se realizó la clasificación manual del histórico de los tickets (esta clasificación manual de los tickets históricos se usó más adelante para encontrar la exactitud del

modelo y la matriz de incertidumbre). Posteriormente, se escogieron los tickets más relevantes para ingresarlos como ejemplos en el modelo de AI.

tickets_rino CONSULTA COMPARTIR COPIAR

ESQUEMA DETALLES EXPLORADOR DE TABLAS **VISTA PREVIA** ESTA

Filtro Ingresar el nombre o el valor de la propiedad

<input type="checkbox"/>	Nombre del campo	Tipo	Modo	Clave	Interc
<input type="checkbox"/>	ticket_id	STRING	NULLABLE	-	-
<input type="checkbox"/>	criticidad	STRING	NULLABLE	-	-
<input type="checkbox"/>	justificacion_criticidad	STRING	NULLABLE	-	-
<input type="checkbox"/>	empresa	STRING	NULLABLE	-	-
<input type="checkbox"/>	tipo_servicio	STRING	NULLABLE	-	-
<input type="checkbox"/>	categoria	STRING	NULLABLE	-	-
<input type="checkbox"/>	subcategoria	STRING	NULLABLE	-	-
<input type="checkbox"/>	estado_solicitud	STRING	NULLABLE	-	-
<input type="checkbox"/>	fecha_solicitud	DATETIME	NULLABLE	-	-
<input type="checkbox"/>	solicitado por	STRING	NULLABLE	-	-

EDITAR ESQUEMA

Figura 8. Campos de vista de tabla con información de los tickets.

4.3.1.1 Área de BI

Al estudiar los tickets para el área de BI, se encontró que estos se podían clasificar por dos tipos: por tipo de servicio (plataforma/Bancolombia) o por tipo de petición (información que solicitaban).

Por el tipo de servicio, se clasifica en plataforma o en Bancolombia, ya que el servicio de Bancolombia tiene un trato diferente a los demás servicios. Las diferencias de los correspondientes bancarios de Bancolombia incluyen:

- Establecimientos exclusivos para los servicios de Bancolombia.
- Almacenamiento en bases de datos separadas.
- Campos diferentes en las bases de datos respecto a los demás servicios.

- Asignación de número de corresponsal bancario (CB) para los empleados que laboran en los establecimientos (además del id de detallista)
- Cálculo de comisiones con respecto a la cantidad de transacciones, no del valor de estas.
- No suelen pedir información de los corresponsales (o detallistas), ya que los funcionarios de la sede central de Comercial Card S.A.S. ya tienen toda la información al ser menos empleados y ser información delicada.
- La información del tipo de transacción y el id de la transacción es más relevante que para el caso de los demás servicios.

4.3.1.1.1 Por tipo de servicio

Las primeras clasificaciones que se determinaron para el tipo de servicio al estudiar el histórico de los tickets fueron:

- bancolombia
- corresponsales
- plataforma
- todos

La clasificación de "bancolombia" se refiere a las solicitudes correspondientes a los corresponsales bancarios de Bancolombia, la clasificación de "corresponsales" son para las solicitudes tanto de corresponsal de Bancolombia como de plataforma, pero que en las de plataforma se refieran a otro(s) corresponsal(es), esto porque la información que se solicita suele ser respecto a cantidad de transacciones y no al monto; las clasificaciones de plataforma se refiere a cuando no se tienen ningún corresponsal de Bancolombia, y "todos" es cuando se tienen tanto corresponsales bancarios Bancolombia, como plataforma sin corresponsales bancarios.

No se agregaron las clasificaciones de "otros" ni de "area equivocada", ya que, aunque había tickets que no correspondían al área de BI o que sean complejos de determinar, igual se pueden clasificar en alguna de estas tres clasificaciones.

Al realizar las pruebas del modelo de AI y observar su exactitud, se encontró que las clasificaciones del modelo influían en la exactitud del modelo, por lo que las clasificaciones se modificaron posteriormente según los resultados del modelo de AI.

Se encontró un aumento en los porcentajes de exactitud, realizando los siguientes cambios al modelo:

- Priorizando clasificaciones
- Probando diferentes nombres para las clasificaciones.

- Eliminando la clasificación de "corresponsales", clasificándola como "plataforma y bancolombia" (la clasificación de "bancolombia" tenía descripción con frases como "corresponsal Bancolombia", "corresponsalía", corresponsales bancarios", por lo que se piensa que esto hacía que el modelo clasificara estos tickets como "corresponsales").

Para la clasificación por servicio, se dividieron las clasificaciones para que fueran más específicas (porque con esto el modelo por tipo de petición aumentó considerablemente); sin embargo, no se encontró una división que aumentara la exactitud. Para las clasificaciones "plataforma y bancolombia" y "plataforma sin bancolombia", se dividió por tipos de servicios que se ofrecen (apuestas, telefonía móvil, etcétera), y también se dividió por productos (Nequi, Essa y otras entidades), pero el porcentaje disminuyó en todas las pruebas

Se creó una nueva clasificación llamada "corresponsales", la cual eran otros corresponsales bancarios que no fueran Bancolombia, pero el porcentaje de exactitud también disminuyó considerablemente.

4.3.1.1.2 Por tipo de petición

Las primeras clasificaciones que se determinaron para el tipo de petición al estudiar el histórico de los tickets fueron:

- ventas
- fecha máxima inactividad
- información detallistas
- información detallistas directos
- información detallistas indirectos
- nequi fiscalia
- otros

Se determinaron estas clasificaciones, porque para los tickets importantes a automatizar se encuentran en los tickets que solicitan ventas e información de detallistas. Los demás tickets, se podían clasificar como "otros". Estos tickets incluyen a los tickets poco comunes, a los que van dirigidos a otras áreas, y a dos tipos de tickets (atc y comisiones) cuyos datos se manejan manualmente.

Al realizar las pruebas del modelo de AI y observar su exactitud, se encontró que las clasificaciones del modelo influían en la exactitud del modelo, por lo que las clasificaciones se modificaron posteriormente según los resultados del modelo de AI.

Se encontró un aumento en los porcentajes de exactitud, realizando los siguientes cambios al modelo:

- Priorizando clasificaciones.
- Dividiendo de las clasificaciones "informacion detallistas", "informacion detallistas directos" e "informacion detallistas indirectos" la clasificación de "puntos", ya que se solía pedir georreferenciación, direcciones, ciudades y departamentos, y estas son palabras claves y específicas. Además, estos tickets tenían baja exactitud en su clasificación (se tendía a clasificar como ventas o comisiones).
- División de algunas clasificaciones de "otros" por la clasificación "comisiones" y "atc".
- Poniendo la clasificación "atc" y "comisiones" como las más prioritarias (ya que eran tickets más específicos y que en el mismo ticket tendían a decir la palabra "ATC" y "comisión").
- Eliminando las clasificaciones de "nequi fiscalia" y de "fecha maxima inactividad" (estos tickets son muy frecuentes, así que los empleados no suelen dar detalles de lo que se solicita, sino que usan las frases "nequi de fiscalía", "nequi para fiscalía", "fecha de inactividad", "fecha máxima de inactividad" para hacer la solicitud. Debido a esto, esta clasificación se encontró con código de Python y palabras claves, y se obtuvo un 100% de éxito encontrándose de esta forma).

4.3.1.2 Área de IT

El área de IT no tenía trato diferente respecto al tipo de servicio del que se hablaba, por lo que se clasificó respecto a las peticiones que se solicitaban. Se encontraron gran diversidad de tipos de peticiones, y a esta área le llegaban muchos tickets diarios, por lo que, después de realizar la clasificación manual, se hizo un conteo de la cantidad de clasificaciones. Con este conteo, se observó los que menos cantidad tenían y se revisó si se podían analizar con otra clasificación, y los tickets a los que no se les podía incluir en otra clasificación con mayor cantidad de tickets, se les agregaba a la clasificación de "otros". Así, las clasificaciones anteriores son las más solicitadas y relevantes a analizar y, posiblemente, automatizar.

Las primeras clasificaciones que se determinaron para el área de TI al estudiar el histórico de los tickets fueron:

- asignación a equipos
- automatización y envíos
- comisiones
- etiquetas
- modificación de informes en modulo
- modificación de módulos
- otros

- perfiles de rino
- permisos
- preguntas
- reparación funcionalidad rino
- solicitud informes
- tableros
- verificación de informes en módulos
- viáticos

Al realizar las pruebas del modelo de AI, se encontró que las clasificaciones del modelo influían en la exactitud del modelo, por lo que las clasificaciones se modificaron posteriormente según los resultados del modelo de AI.

Se encontró un aumento en los porcentajes de exactitud, realizando los siguientes cambios al modelo:

- Priorizando clasificaciones.
- Poniendo las clasificaciones más específicas como más prioritarias (que no tuvieran tantos productos y que en el mismo ticket tendiera a decir palabras claves).
- Agrupando las clasificaciones "modificación de informes en modulo", "verificación de informes en modulos", "tableros" a la clasificación de "modificación de modulos", ya que todas las clasificaciones siguen siendo peticiones de modificaciones en módulos del CRM que se encuentran con problemas de información. Además, se observó que para todas las clasificaciones primero se debía hacer una verificación de esta información, aunque fuera solicitada como orden o como urgencia, sin mostrar incertidumbre en las fallas.
- Agrupando la clasificación "permisos" a la clasificación de "reparación funcionalidad rino", ya que "permisos" se refiere a problemas en cuanto a permisos de usuario para ingresar a una pestaña de la página del CRM, y esto es problema del funcionamiento del CRM de RINO. Además, si se solicita permisos por primera vez, igual se debe comprobar si la razón de no tener permisos es por un problema entre los códigos que observan las bases de datos para generar los permisos.
- Separando las clasificaciones de "otros" entre las clasificaciones de "modificación de modulos" y "reparación funcionalidad rino". Aunque en "otros" habían peticiones particulares, estas se podían ver como preguntas particulares (clasificación "preguntas", como mal funcionamiento del CRM (clasificación "reparación funcionalidad rino") o como sospecha o petición de corrección en elementos o data de un módulo (clasificación "modificación de modulos").

4.3.2 Creación del modelo de AI

Se estudiaron las APIs en Vertex AI. Se encontró que las APIs tienen “bloqueos de contenido dañino” (incitación al odio o a la violencia, contenido peligroso, contenido sexual explícito y contenido de acoso). Entre más se reducían estos bloqueos, mayor cantidad de tickets clasificaba el modelo, por lo que la exactitud aumentaba. Las configuraciones para este bloqueos se observan en la figura 9.

Configuración de seguridad

Puedes ajustar la probabilidad de recibir una respuesta del modelo que podría incluir contenido dañino. El contenido se bloquea en función de la probabilidad de que sea dañino. [Más información](#)

Incitación al odio o a la violencia	Bloquear poco	▼	?
Contenido peligroso	Bloquear poco	▼	?
Contenido sexual explícito	Bloquear poco	▼	?
Contenido de acoso	Bloquear poco	▼	?

[Restablecer configuración predeterminada](#)

[Guardar](#)

[Cerrar](#)

Figura 9. Configuración de bloqueos de contenido dañino en APIs de AI generativa en Vertex AI [19].

Las APIs también tienen “grados de libertad” o “temperatura” (que sirve para dar respuestas menos objetivas y más intuitivas). La configuración de esta temperatura se observa en la figura 10. Disminuyendo estas a 0% se obtiene un aumento en la exactitud.

Temperatura ?

0 2

Figura 10. Configuración de temperatura en APIs de AI generativa en Vertex AI [19].

El bloqueo de contenido dañino y la temperatura se modifican por medio de código de Python. En el código de Python se crean variables con estas especificaciones, y estas se entregan a la API al generar un prompt.

Al modificar las dos configuraciones (bloqueo de contenido dañino y temperatura), la API genera un código en Python. Este código incluye variables con las especificaciones ajustadas, las cuales se envían a la API cuando se genera un prompt.

La creación del contexto para el modelo de AI lo dividiremos en tres partes: contexto inicial, contexto de ejemplos y contexto aclaratorio. El contexto inicial constaba de un resumen de lo que era la empresa, sus funciones, su organización, las palabras claves que se debían entender o se podían malinterpretar con otros significados, explicación de las funciones del área (BI o TI); después se le especificó a la AI el papel que hacía (por ejemplo, "eres un experto en la clasificación de solicitudes de bases de datos de la área de BI de la empresa de PTM"), y qué debía hacer con el prompt que se le iba a entregar (por ejemplo, "recibirás una descripción del ticket y debes clasificarlo con las clasificaciones y ejemplos que se te darán"). En el contexto inicial, también se decían qué clasificaciones existían. En el contexto de ejemplos se toman los tickets previamente clasificados y seleccionados para el modelo, se hace un recorrido por cada uno de los tickets, en donde se extrae la descripción y la clasificación del ticket, y estos campos se muestran como ejemplo del prompt que se le ingresa a la AI (campo de descripción), y del resultado que esta debe arrojar (campo de clasificación). Por último, el contexto aclaratorio, al igual que el contexto inicial, es un escrito plano. En este, se le dieron especificaciones y detalles de cuándo se escogía cierta clasificación de un ticket, o en qué se basaba para las clasificaciones específicas más complicadas. Además, se le hacían especificaciones en general para que la AI tuviera en cuenta. El diagrama de flujo que muestra las partes del código es el mostrado en la figura 11, y los bloques de "creación de modelo AI" y "Creación de archivo de prueba" se observan en las figuras 12 y 13, respectivamente.

Se realizaron diferentes modelos de AI, y se realizó un código para obtener la exactitud y la matriz de incertidumbre del modelo. Con la matriz de incertidumbre, se iba observando qué clasificaciones tenían menos aciertos, y con esto se iba modificando el modelo de AI. El ideal para la exactitud en un modelo de clasificación de AI es del 85%, ya que, entre menos exactitud, hay menos acierto en las clasificaciones (underfitting), y al tener mucha exactitud se puede tener un sobreajuste (overfitting), lo cual significa que ese modelo funciona tan bien para los tickets de prueba, que es un modelo personalizado para ellos, por lo que si se cambian los datos (tickets) por nuevos datos diferentes, la exactitud baja considerablemente. Por lo tanto, se determinó un ideal de exactitud del 85%, con una desviación del 5%, con lo que se buscaba tener un modelo con exactitud de entre el 80% y el 90%, siendo el 85% la mejor exactitud.



Figura 11. Diagrama de flujo para el código de creación de modelo de AI.

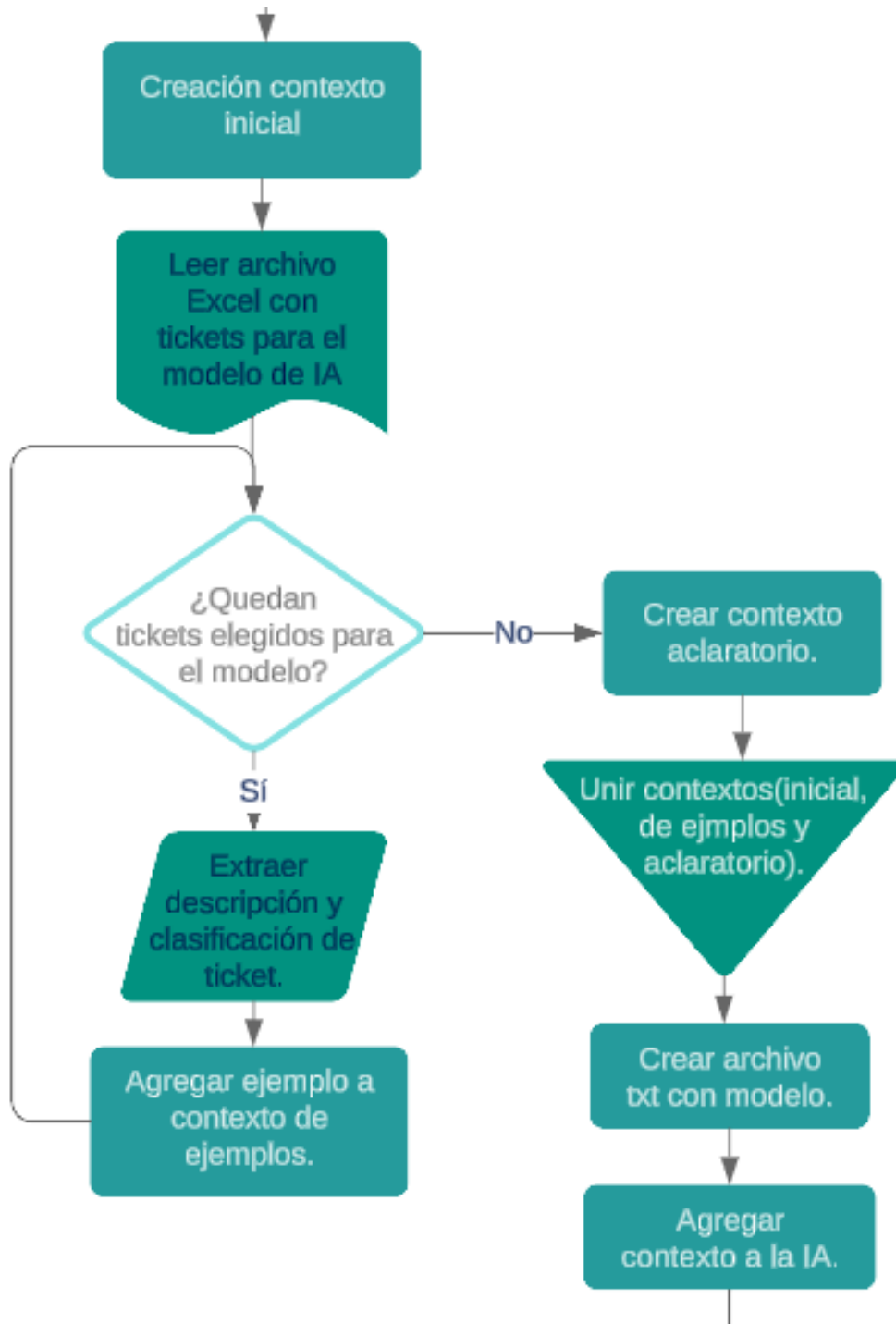


Figura 12. Diagrama de flujo para bloque de creación de modelo AI en código de creación de modelo de la figura 11.



Figura 13. Diagrama de flujo para bloque de creación de archivo de prueba de modelo AI en código de creación de modelo de la figura 11.

4.3.2.1 Área de BI

Para el área de BI se crearon y analizaron dos tipos de clasificación: por tipo de servicio, y por tipo de petición. En este apartado se ven los cambios que se realizaron a los modelos y lo que se obtuvo de estos cambios.

4.3.2.1.1 Por tipo de servicio

Se encontró un aumento en los porcentajes de exactitud, realizando los siguientes cambios al modelo:

- Priorizando clasificaciones.

- Escribiendo en el contexto aclaratorio que las clasificaciones estaban por orden de prioridad.
- Explicando en el contexto aclaratorio que "prioridad" significaba que, si se encontraba que podía corresponder en esa clasificación, se clasificara como en ella sin evaluar las clasificaciones siguientes.
- Poniendo las clasificaciones más específicas como más prioritarias (que no tuvieran tantos productos y que en el mismo ticket tendiera a decir palabras claves).
- Dando más contexto de los roles de clientes, mayoristas, distribuidores y detallistas.
- Probando diferentes nombres para las clasificaciones.
- Eliminando la clasificación de "corresponsales", clasificándola como "plataforma y bancolombia" (la clasificación de "bancolombia" tenía descripción con frases como "corresponsal Bancolombia", "corresponsalía", "corresponsales bancarios", por lo que se piensa que esto hacía que el modelo clasificara estos tickets como "corresponsales").

La priorización de las clasificaciones, el cambio en los nombres de las clasificaciones, y la eliminación de la clasificación de "corresponsales" fueron las más influyentes en el porcentaje de exactitud. Cualquier pequeño cambio en el orden de la priorización de las clasificaciones generalmente aumentaba o disminuía considerablemente el porcentaje de exactitud, por lo que se hicieron varias pruebas modificando su orden.

Para la clasificación por servicio, se dividieron las clasificaciones para que fueran más específicas (porque con esto el modelo por tipo de petición aumentó considerablemente); sin embargo, no se encontró una división que aumentara la exactitud. Para las clasificaciones "plataforma y bancolombia" y "plataforma sin bancolombia", se dividió por tipos de servicios que se ofrecen (apuestas, telefonía móvil, etcétera), y también se dividió por productos (Nequi, Essa y otras entidades), pero el porcentaje disminuyó en todas las pruebas.

Se creó una nueva clasificación llamada "corresponsales", la cual eran otros corresponsales bancarios que no fueran Bancolombia, pero el porcentaje de exactitud también disminuyó considerablemente.

Se tomaron los tickets en los que el modelo fallaba en las clasificaciones y se añadieron a los ejemplos para el modelo. La exactitud del modelo tuvo variaciones pequeñas (máximo 0.02% y mínimo -0.01%).

4.3.2.1.2 Por tipo de petición

Se encontró un aumento en los porcentajes de exactitud, realizando los siguientes cambios al modelo:

- Priorizando clasificaciones.
- Escribiendo en el contexto aclaratorio que las clasificaciones estaban por orden de prioridad.

- Explicando en el contexto aclaratorio que "prioridad" significaba que, si se encontraba que podía corresponder en esa clasificación, se clasificara como en ella sin evaluar las clasificaciones siguientes.
- Dividiendo de las clasificaciones "informacion detallistas", "informacion detallistas directos" e "informacion detallistas indirectos" la clasificación de "puntos", ya que se solía pedir georreferenciación, direcciones, ciudades y departamentos, y estas son palabras claves y específicas. Además, estos tickets tenían baja exactitud en su clasificación (se tendía a clasificar como ventas o comisiones).
- División de algunas clasificaciones de "otros" por la clasificación "comisiones" y "atc".
- Poniendo la clasificación "atc" y "comisiones" como las más prioritarias (ya que eran tickets más específicos y que en el mismo ticket tendían a decir la palabra "ATC" y "comisión").
- Dando más contexto de los roles de clientes, mayoristas, distribuidores y detallistas.
- Eliminando las clasificaciones de "nequi fiscalia" y de "fecha maxima inactividad" (estos tickets son muy frecuentes, así que los empleados no suelen dar detalles de lo que se solicita, sino que usan las frases "nequi de fiscalía", "nequi para fiscalía", "fecha de inactividad", "fecha máxima de inactividad" para hacer la solicitud. Debido a esto, esta clasificación se encontró con código de Python y palabras claves, y se obtuvo un 100% de éxito encontrándose de esta forma).

Se tomaron los tickets en los que el modelo fallaba en las clasificaciones y se añadieron a los ejemplos para el modelo. La exactitud del modelo tuvo variaciones pequeñas, y estas variaciones fueron negativas hasta de 0.33%.

La priorización de las clasificaciones, la división de estas, y la eliminación de las clasificaciones de "nequi fiscalia" y de "fecha maxima inactividad" fueron las más influyentes en el porcentaje de exactitud. Cualquier pequeño cambio en el orden de la priorización de las clasificaciones generalmente aumentaba o disminuía considerablemente el porcentaje de exactitud, por lo que se hicieron muchas pruebas modificando su orden.

4.3.2.2 Área de IT

El área de IT no tenía trato diferente respecto al tipo de servicio del que se hablaba, por lo que se clasificó respecto a las peticiones que se solicitaban. Se encontraron gran diversidad de tipos de peticiones, y a esta área le llegaban muchos tickets diarios, por lo que, después de realizar la clasificación manual, se hizo un conteo de la cantidad de clasificaciones. Con este conteo, se observó los que menos cantidad tenían y se revisó si se podían analizar con otra clasificación, y los tickets a los que no se les podía incluir en otra clasificación con mayor cantidad de tickets, se les agregaba a la clasificación de "otros". Así, las clasificaciones anteriores son las más solicitadas y relevantes a analizar y, posiblemente, automatizar.

Se encontró un aumento en los porcentajes de exactitud, realizando los siguientes cambios al modelo:

- Profundizando las funciones de las que se encargaba el área de IT en el contexto inicial de la AI.
- Priorizando clasificaciones.
- Escribiendo en el contexto aclaratorio que las clasificaciones estaban por orden de prioridad.
- Explicando en el contexto aclaratorio que "prioridad" significaba que, si se encontraba que podía corresponder en esa clasificación, se clasificara como en ella sin evaluar las clasificaciones siguientes.
- Poniendo las clasificaciones más específicas como más prioritarias (que no tuvieran tantos productos y que en el mismo ticket tendiera a decir palabras claves).
- Agrupando las clasificaciones "modificación de informes en modulo", "verificación de informes en modulos", "tableros" a la clasificación de "modificación de modulos", ya que todas las clasificaciones siguen siendo peticiones de modificaciones en módulos del CRM que se encuentran con problemas de información. Además, se observó que para todas las clasificaciones primero se debía hacer una verificación de esta información, aunque fuera solicitada como orden o como urgencia, sin mostrar incertidumbre en las fallas.
- Agrupando la clasificación "permisos" a la clasificación de "reparación funcionalidad rino", ya que "permisos" se refiere a problemas en cuanto a permisos de usuario para ingresar a una pestaña de la página del CRM, y esto es problema del funcionamiento del CRM de RINO. Además, si se solicita permisos por primera vez, igual se debe comprobar si la razón de no tener permisos es por un problema entre los códigos que observan las bases de datos para generar los permisos.
- Separando las clasificaciones de "otros" entre las clasificaciones de "modificación de modulos" y "reparacion funcionalidad rino". Aunque en "otros" habían peticiones particulares, estas se podían ver como preguntas particulares (clasificación "preguntas", como mal funcionamiento del CRM (clasificación "reparacion funcionalidad rino") o como sospecha o petición de corrección en elementos o data de un módulo (clasificación "modificación de modulos").

Las modificaciones más influyentes en el porcentaje de exactitud fueron la profundización en las funciones del área de IT, la priorización de las clasificaciones, la modificación de las clasificaciones "verificación de informes en modulos", "modificación de informes en modulos" y "permisos".

En este modelo no se necesitaba dar contexto de los roles de clientes, mayoristas, distribuidores y detallistas, porque no se tienen división de estos en las solicitudes ni el área de IT, más que en los tableros o módulos que tienen esta área específica. Estos módulos con nombres que separaban los tipos de clientes se detallaron en las especificaciones que se hicieron más adelante de las

clasificaciones, pero para el modelo no se usaron. Este contexto de roles se eliminó del modelo de AI, y no se tuvieron cambios en la exactitud.

Se tomaron los tickets en los que el modelo fallaba en las clasificaciones y se añadieron a los ejemplos para el modelo. La exactitud del modelo tuvo variaciones negativas (hasta de 2.85%).

4.4 Automatización de procesos

En esta sección se abordan las soluciones de los tickets de ambas áreas (BI e IT), la capacidad de reducir o eliminar los tiempos de resolución, así como los procesos llevados a cabo para el estudio y las bases de datos que permiten la clasificación automática de los tickets.

4.4.1 Área de BI

Se trabajó en la generación automática de la clasificación de los tickets, usando el modelo de AI, la API de Vertex, código de Python, y la conexión con BigQuery, en donde se guardó en una tabla los tickets clasificados (tickets históricos), y cada día se añadían los tickets más actuales. Además, se creó otra tabla con los tickets que se han recibido desde hace un día (tickets diarios), y esta tabla se actualiza tres veces al día. La ingesta histórica de tickets se basó en el diagrama de flujos de la figura 14, la actualización de los tickets históricos en el diagrama de flujo de la figura 15, y la actualización de los tickets diarios en el diagrama de flujo de la figura 16.

Estos códigos se agregaron a dos repositorios de BitBucket. Un repositorio (daily-classification-tickets) para el código de actualización de los tickets diarios, y el otro repositorio (historical-classification-tickets) para el código de la ingesta histórica de tickets y para el código de la actualización de los tickets históricos. En el repositorio para los tickets históricos se automatizó el código para la actualización (diagrama de flujo 16), llamándolo desde el main del repositorio; el código de ingesta histórica sólo se almacenó en el repositorio como un archivo .pynb para que se descargara y fuera ejecutado cuando todo el historial tuviera que ser reemplazado. En el repositorio historical-classification-tickets también se almacenaron los archivos para la creación de los modelos de AI para la clasificación de estos.

Estos repositorios se automatizaron por medio de Google Cloud Function, en donde se determinaron parámetros, credenciales, horario en que se ejecutaban los códigos, y se dio la URL de la ubicación del repositorio. Al ejecutar Function, se actualizaban las tablas en BigQuery o se creaban en caso de que no existieran. Las tablas generadas por los códigos se observan en la figura 17, en donde la tabla `classified_tickets` del apartado de `automated`, es la tabla para los tickets diarios; y `historical_classified_tickets`, también del apartado de `automated`, es la tabla para los tickets históricos; y `tickets_rino` es la vista generada a partir de la información del CRM de la empresa, en donde se realizan las consultas SQL para generar las tablas de los tickets diarios y de los tickets históricos.

Los campos de las tablas se observan en las figuras 18 y 19 y las vistas previas de las tablas se observan en las figuras 20 y 21.

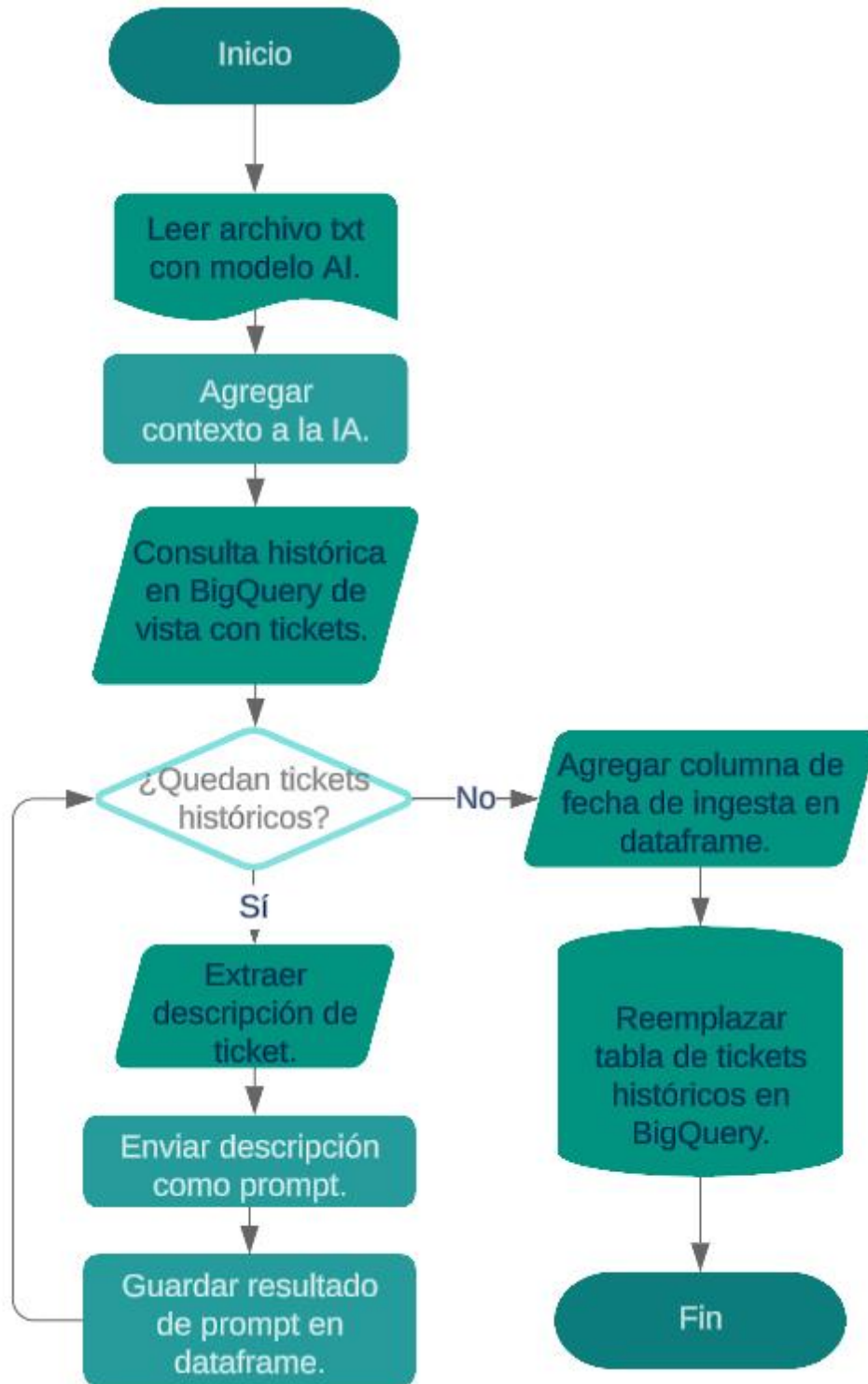


Figura 14. Diagrama de flujo para ingesta histórica de tickets como tabla a BigQuery.

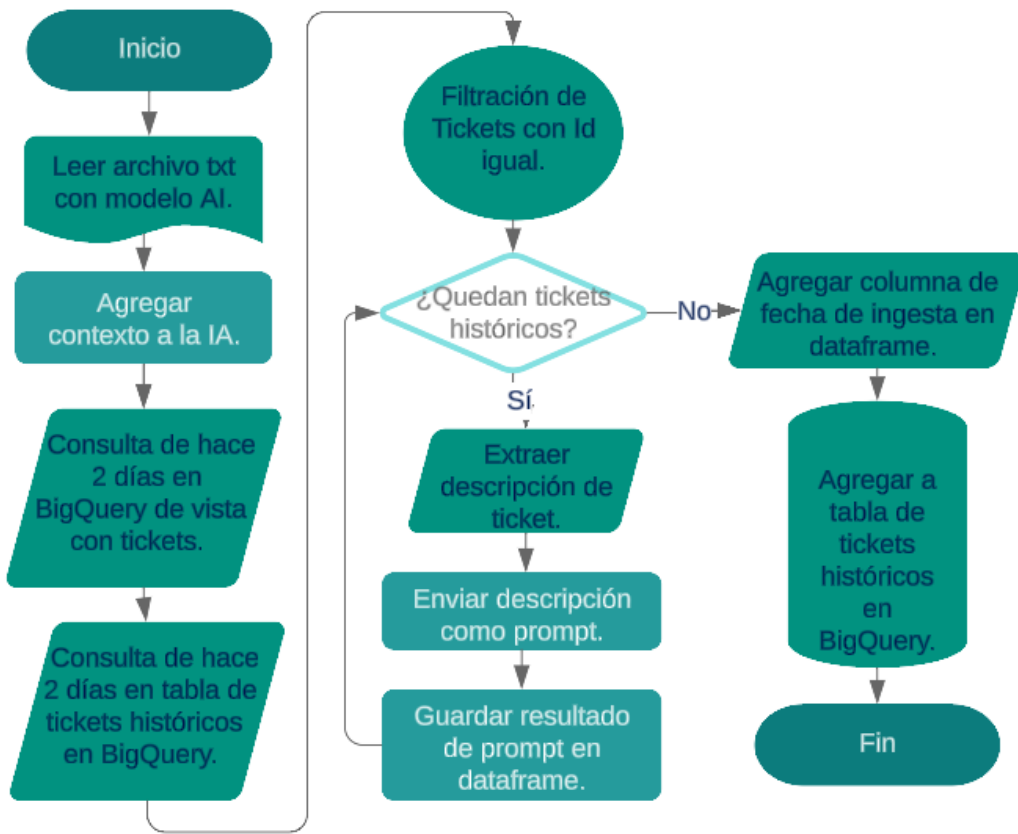


Figura 15. Diagrama de flujo para actualización de tickets históricos en tabla de BigQuery.

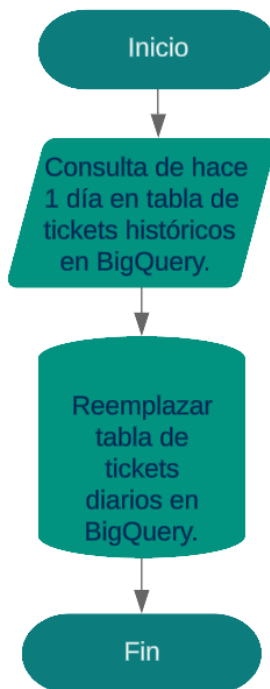


Figura 16. Diagrama de flujo para actualización de tickets diarios en tabla de BigQuery.

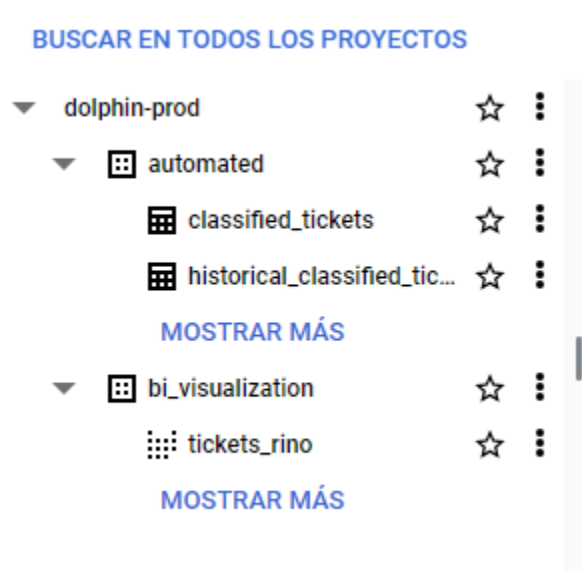


Figura 17. Tablas de BigQuery resultantes de la automatización de los códigos del repositorio en BitBucket automatizados en Google Cloud Function.

Se buscó qué tipos de tickets podían ser automatizados. Se encontraron 3 clasificaciones de tickets que cumplían para ser automatizados, teniendo en cuenta que no eran tickets muy particulares (poco estándar) que hubiera que analizar a profundidad.

Para analizar mejor las clasificaciones, se realizaron dos modelos de AI: uno para determinar qué campos se solicitaban mayormente respecto a la clasificación, y el otro para dar información general de lo que la AI pensara que era importante.

Se analizaron las clasificaciones, se separaron los campos más solicitados que no tuvieran información sensible, y se hizo una evaluación de qué era lo que más se solicitaba y con respecto a qué campos.

Se estudió la mejor opción para realizar cada una de las 3 clasificaciones, se comparó lo que ya se tenía en el área de BI, códigos que pudiera resolver estas solicitudes de tickets y tableros, pero se determinó que los códigos y tableros no tenían información suficiente para suplir la información solicitada en los tickets, ni en cuando a margen de fechas ni cantidad de filtros necesarios.

Se optó por hacer un código en una clasificación ("nequi fiscalía") y dos tableros para las otras dos (fecha máxima de inactividad y ventas).

Después de crear los tableros, se hizo la consulta de las personas que requirieron solicitudes de tickets en los últimos 6 meses y las cantidades de tickets que solicitaba, y se ordenaron. De esta lista, se descartaron algunas personas. Se citó a las personas elegidas a una reunión de Microsoft Teams y en esta reunión se explicaron las tablas, y se tomaron anotaciones de cambios sugeridos para información que ellos desearan. Algunas sugerencias no se tomaron en cuenta porque se referían a información sensible, pero con las sugerencias que se tomaron en cuenta se cambiaron los tableros.

historical_clas... 🔍 + 👤 📄 ⋮ 🔄

< **ESQUEMA** DETALLES VISTA PREVIA EXP >

Filtro Ingresar el nombre o el valor de la propiedad

<input type="checkbox"/>	Nombre del campo	Tipo	Modo	Clave
<input type="checkbox"/>	fecha_ingesta	STRING	NULLABLE	-
<input type="checkbox"/>	fecha_solicitud	TIMESTAMP	NULLABLE	-
<input type="checkbox"/>	ticket_id	STRING	NULLABLE	-
<input type="checkbox"/>	clasificacion	STRING	NULLABLE	-
<input type="checkbox"/>	resumen_descripcion	STRING	NULLABLE	-

EDITAR ESQUEMA VER POLÍTICAS DE ACCESO DE FILA

Figura 18. Campos de la tabla de tickets históricos de la figura 17.

e... ets ✕ + ▼

classified_tick... 🔍 + 👤 📄 ⋮ 🔄

< **ESQUEMA** DETALLES VISTA PREVIA EXP >

Filtro Ingresar el nombre o el valor de la propiedad

<input type="checkbox"/>	Nombre del campo	Tipo	Modo	Clave
<input type="checkbox"/>	fecha_ingesta	STRING	NULLABLE	-
<input type="checkbox"/>	fecha_solicitud	TIMESTAMP	NULLABLE	-
<input type="checkbox"/>	ticket_id	STRING	NULLABLE	-
<input type="checkbox"/>	clasificacion	STRING	NULLABLE	-
<input type="checkbox"/>	resumen_descripcion	STRING	NULLABLE	-

EDITAR ESQUEMA VER POLÍTICAS DE ACCESO DE FILA

Figura 19. Campos de la tabla de tickets diarios de la figura 17.

ESQUEMA	DETALLES	VISTA PREVIA	EXPLORADOR DE TABLAS	VISTA PREVIA	ESTADÍSTICAS	LINAJE	PERFIL DE DATOS	CALIDA
Fila	fecha_ingesta	fecha_solicitud	ticket_id	clasificacion	resumen_descripcion			
1	2024-07-04 16:53:24	2023-09-21 07:57:32 UTC	S04718	informacion detallistas indirect...	Se solicita el consolidado de la base de datos de QR para el mes de agosto.			
2	2024-07-04 16:53:24	2024-05-31 14:40:06 UTC	S07368	informacion detallistas, informacion detallistas directos, informacion detallistas indirectos, informacion	Se solicita la base de datos actualizada para el mes de junio de los siguientes tipos de clientes de Bancolombia:			
3	2024-07-04 16:53:24	2024-05-27 09:27:17 UTC	S07318	area equivocada este ticket no se relaciona con las es de tickets que se han definido. el ticket	Se solicita separar los nombres y apellidos de una base de datos donde ambos se encuentran en una sola celda.			
4	2024-07-15 15:00:23	2024-07-15 07:44:57 UTC	S07880	informacion detallistas,	Se solicita la base de datos			

Figura 20. Vista previa de la tabla de tickets históricos de la figura 17.

ESQUEMA	DETALLES	VISTA PREVIA	EXPLORADOR DE TABLAS	VISTA PREVIA	ESTADÍSTICAS	LINAJE	PERFIL DE DATOS	CA
Fila	fecha_ingesta	fecha_solicitud	ticket_id	clasificacion	resumen_descripcion			
1	2024-09-02 15:00:23	2024-09-02 11:19:51 UTC	S08487	comisiones	Asunto: Pago de ADQ no efectuado en la plataforma PTM			

Figura 21. Vista previa de la tabla de tickets diarios de la figura 17.

4.4.2 Área de IT

Para el caso de IT, había gran cantidad de tickets, por lo que antes de hacer la clasificación se hizo un prompt con muestras de los tickets, separadas de a 200 tickets, para que la AI diera un contexto sobre qué trataban los tickets y la cantidad de solicitudes de cada ticket que se había. Este contexto ayudó a decidir las clasificaciones para los tickets.

Se realizó la clasificación de los tickets de IT. Se estudiaron los tickets de IT, y de cada clasificación se hicieron sub-clasificaciones, y detallados de los tickets para estudiarlos a detalle. De esto se obtuvieron estadísticas y un análisis de los tickets, el cual se les entregó al área de IT.

5 RESULTADOS Y ANÁLISIS

En esta sección se muestran los resultados obtenidos en el proyecto y sus análisis. Las subsecciones son los resultados y análisis del desarrollo de modelo de AI para la clasificación de los tickets (donde se encuentran los resultados de la clasificación de tickets y de la evaluación de la creación el modelo de AI) y de la automatización de los procesos elegidos a gestionar.

5.1 Desarrollo de modelo de AI para clasificación de tickets

A continuación, se muestra el resultado del estudio de los tickets, las clasificaciones resultantes para las áreas de BI y de IT, y el análisis del modelo de AI con mejor resultado.

5.1.1 Clasificación de tickets

A continuación, se muestran las clasificaciones resultantes para los modelos de las áreas de BI y de IT después de los estudios y de la comprobación en exactitud de los modelos.

5.1.1.1 Área de BI

Al estudiar los tickets para el área de BI, se encontró que estos se podían clasificar por dos tipos: por tipo de servicio (plataforma/Bancolombia) o por tipo de petición (información que solicitaban).

5.1.1.1.1 Por tipo de servicio

Las clasificaciones resultantes para el tipo de servicio en el modelo de AI fueron:

- Bancolombia
- Plataforma y Bancolombia
- Plataforma sin Bancolombia

La lista anterior está ordenada en el mismo orden en que se determinó en el código de contexto para la priorización de clasificaciones.

La clasificación de "bancolombia" se refiere a las solicitudes correspondientes a los corresponsales bancarios de Bancolombia, la clasificación de "corresponsales" son para las solicitudes tanto de corresponsal de Bancolombia como de plataforma, pero que en las de plataforma se refieran a otro(s) corresponsal(es), esto porque la información que se solicita suele ser respecto a cantidad de transacciones y no al monto; las clasificaciones de plataforma se refiere a cuando no se tienen ningún corresponsal de Bancolombia, y "todos" es cuando se tienen tanto corresponsales bancarios Bancolombia, como plataforma sin corresponsales bancarios.

5.1.1.1.2 Por tipo de petición

Las clasificaciones resultantes para el tipo de petición en el modelo de AI fueron:

- atc

- comisiones
- area equivocada
- ventas
- puntos
- informacion detallistas
- informacion detallistas directos
- informacion detallistas indirectos
- otros

La lista anterior está ordenada en el mismo orden en que se determinó en el código de contexto para la priorización de clasificaciones.

5.1.1.2 Área de IT

Las clasificaciones resultantes para el tipo de servicio en el modelo de AI fueron:

- asignacion a equipos
- automatizacion de envios
- viaticos
- comisiones
- perfiles de rino
- permisos
- etiquetas
- solicitud informes
- reparacion funcionalidad rino
- modificacion de modulos
- preguntas

La lista anterior está ordenada en el mismo orden en que se determinó en el código de contexto para la priorización de clasificaciones.

5.1.2 Creación del modelo de AI

A continuación, se muestran los resultados de mayor exactitud y mejor matriz de incertidumbre para los modelos de AI desarrollados en las áreas de BI y de IT.

5.1.2.1 Área de BI

El área de BI se ha dividido en dos tipos de clasificación: por tipo de servicio y por tipo de petición. A continuación, se presentan los mejores resultados del modelo de IA para cada una de estas clasificaciones.

5.1.2.1.1 Por tipo de servicio

El resultado con mejor matriz de incertidumbre y mayor exactitud para las clasificaciones por tipo de servicio fueron los de las figuras 22 y 23, respectivamente:

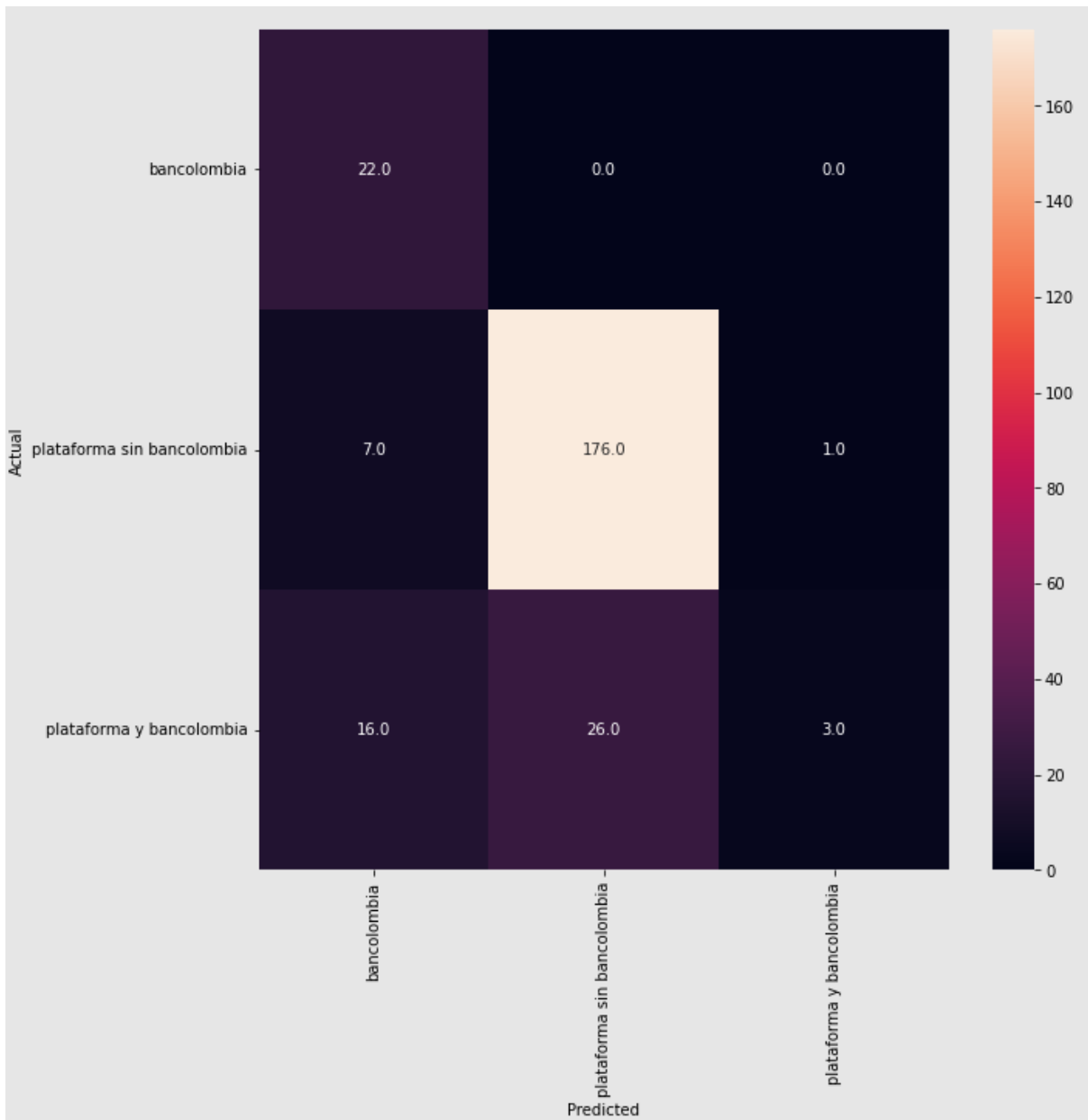


Figura 22. Matriz de incertidumbre para la mayor exactitud lograda en la clasificación por tipo de servicio de los tickets de BI.

```
Tickets con igual clasificación: 201
Tickets totales: 251
Promedio exactitud en clasificación de tickets: 0.8007968127490039
```

Figura 23. Resultado para la mayor exactitud lograda en la clasificación por tipo de servicio de los tickets de BI.

5.1.2.1.2 Por tipo de petición

El resultado con mejor matriz de incertidumbre y mayor exactitud para las clasificaciones por tipo de petición fueron los de las figuras 24 y 25, respectivamente:

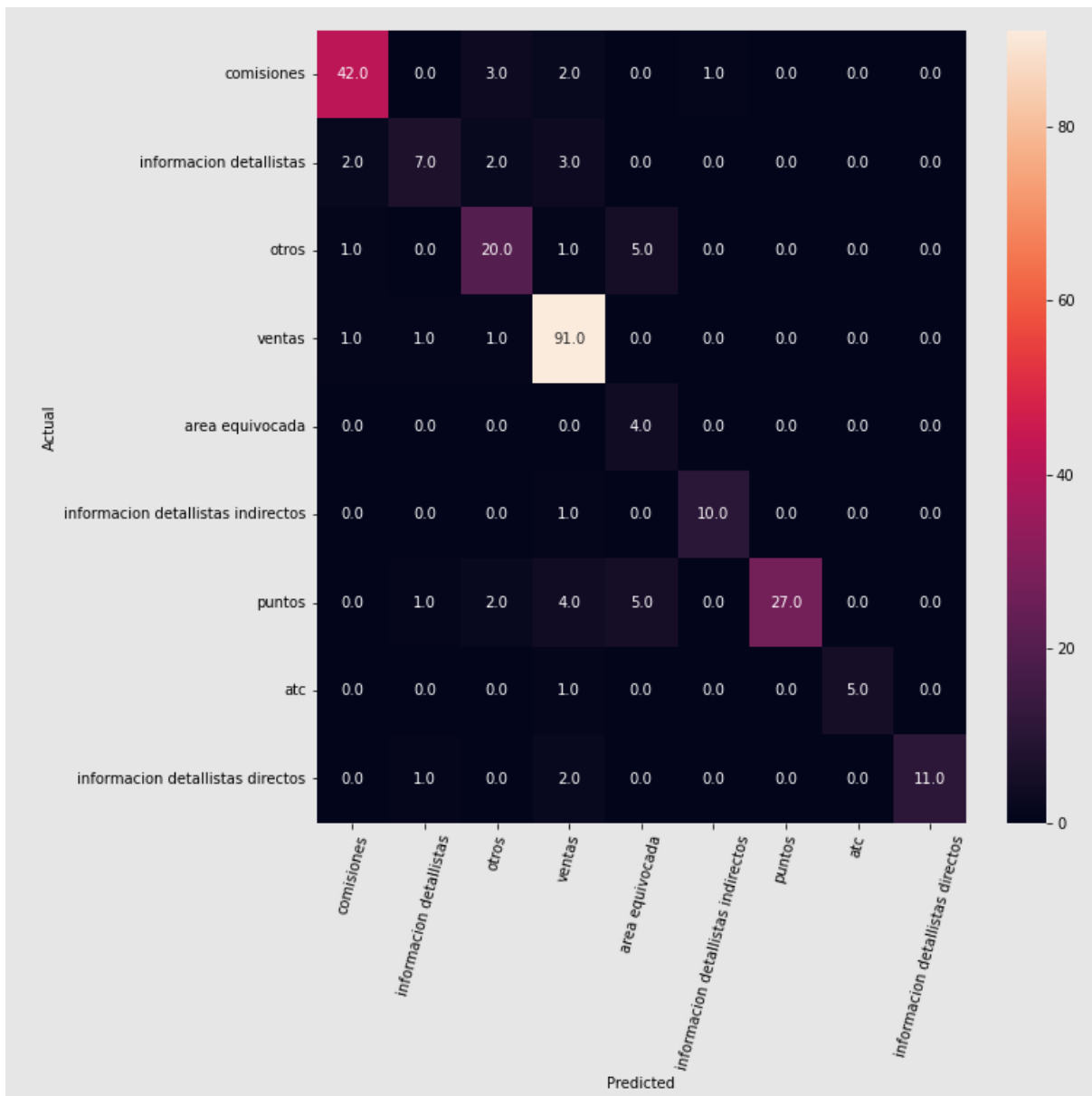


Figura 24. Matriz de incertidumbre para la mayor exactitud lograda en la clasificación por tipo de petición de los tickets de BI.


```
Tickets con igual clasificación: 217
Tickets totales: 257
Promedio exactitud en clasificación de tickets: 0.8443579766536965
```

Figura 25. Resultado para la mayor exactitud lograda en la clasificación por tipo de petición de los tickets de BI.

5.1.2.2 Área de IT

El resultado con mejor matriz de incertidumbre y mayor exactitud para las clasificaciones para el área de IT fueron los de las figuras 26 y 27, respectivamente:

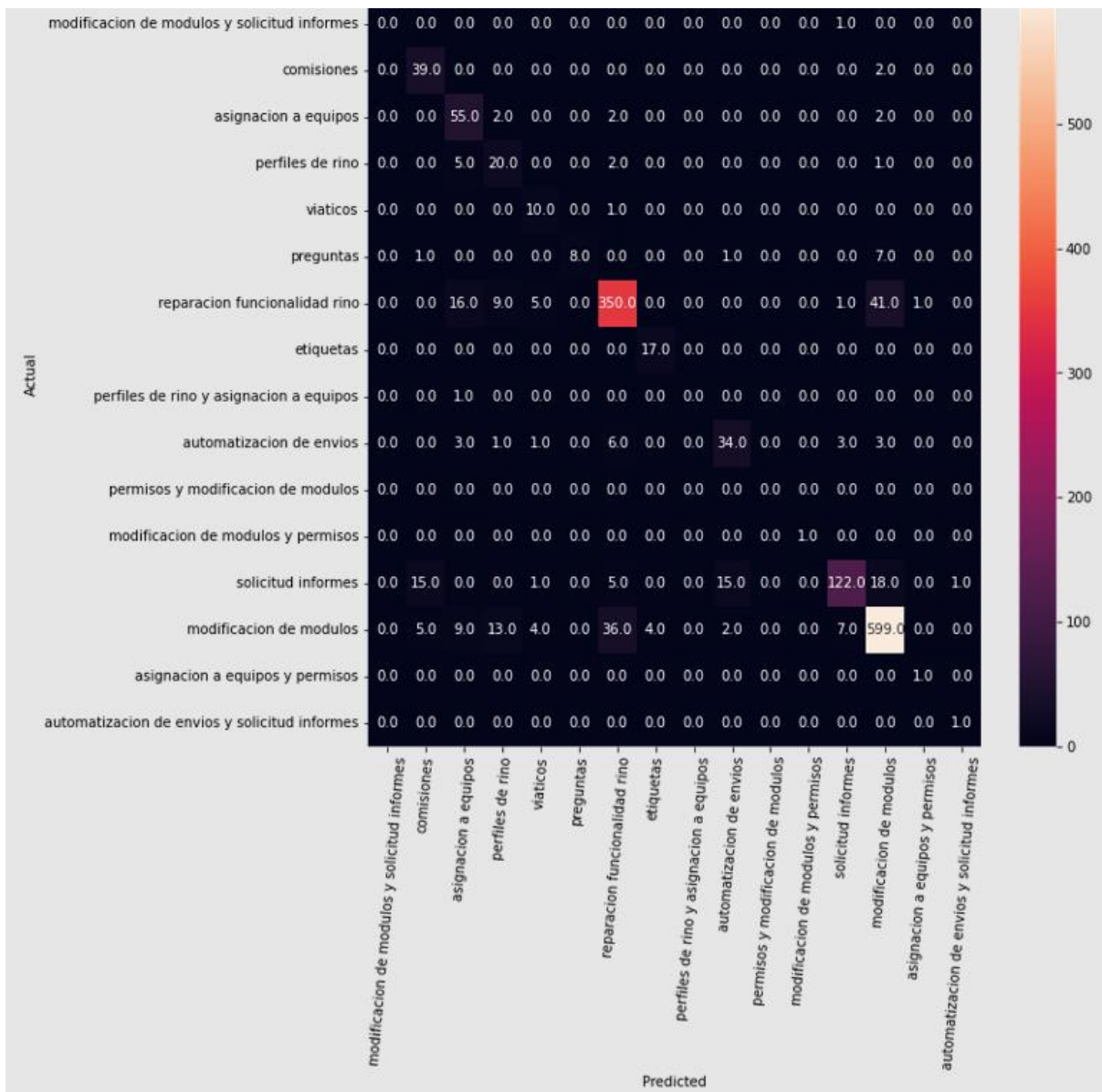


Figura 26. Matriz de incertidumbre para la mayor exactitud lograda en la clasificación del área de IT.

```
Tickets con igual clasificación: 1257
Tickets totales: 1521
Promedio exactitud en clasificación de tickets: 0.8264299802761341
```

Figura 27. Resultado para la mayor exactitud lograda en la clasificación del área de IT.

5.2 Automatización de procesos

En esta sección se muestra el estudio de cada clasificación de tickets respecto al área (BI e IT), su posibilidad de automatización, y cómo se dio solución a los tickets seleccionados.

5.2.1 Área de BI

Dado que el tipo de clasificación de servicio arrojó valores de exactitud menores y que eran clasificaciones que no daban tanta información como en la clasificación de petición, se tomó la clasificación de petición para realizar las clasificaciones.

En las clasificaciones de petición, se estudiaron los campos de datos que se solicitaban por tipo de clasificación, y se encontró que para los tickets de detallistas (informacion detallistas, informacion detallistas directos e informacion detallistas indirectos), la información que se solicitaba era sensible, por lo que se debía evaluar si dar esta información, o qué cantidad de información sensible de un cliente era apto dar. Por esta razón, se optó por no automatizar las soluciones de este tipo de tickets.

Para el caso de las clasificaciones de "nequi fiscalia", este también tiene información sensible, pero es un ticket en el que se deben buscar información y depurar muchos datos para dar solución, así que se realizó un código de Python que dependía de un empleado. El empleado debe descargar archivos de Excel (en donde se encuentra la información a depurar) y ejecutar el código. Este código genera un archivo Excel con la información solicitada, y un archivo Excel sin la depuración de los datos, para que se pueda hacer una comparación en caso de que se produzca una mala depuración. Se optó por no enviar automáticamente el código por correo electrónico, por tener información sensible que es mejor evaluar para saber si dar solución. En la figura 28 se observa el repositorio que se creó y al que se le subió el código para la solución de los tickets con clasificación "nequi fiscalia".







 rasp-ai-vision	74.9 KB	4 days ago	
 tickets-nequi-fiscalia	286.2 KB	10 hours ago	
 unified-base-etl-call	36.6 MB	4 days ago	

Figura 28. Repositorio para la solución de tickets "nequi fiscalia".

Para el caso de las clasificaciones de "otros", es información que se debe estudiar individualmente o son preguntas específicas de conocimientos del área, por lo que no se automatizan.

Para el caso de las clasificaciones de "area equivocada", esta es información que va dirigida a otras áreas diferentes a BI (generalmente para IT), por lo que se contactó con las personas que realizaban este tipo de tickets y se les informó en qué casos realizar los tickets en BI y en IT, y cómo realizar las solicitudes al área de IT.

Para los tickets con clasificación "fecha maxima inactividad", se les puede dar una solución automatizable, ya que el ticket consiste en llamar a un query a BigQuery, y no realizar ni particiones ni depuraciones en el resultado arrojado por BigQuery. Se planteó el uso de un código que se ejecutara cada 2 días, aproximadamente; esto teniendo en cuenta que el ticket no requiere operaciones ni depuraciones, sólo el llamado a un query sin modificaciones, y también teniendo en cuenta que se sacaron los datos de estos tickets en específico, y se encontró que se requerían aproximadamente cada 7 días, variando entre 6 y 9 días. Y como los tickets son urgentes, se espera dar respuesta al ticket en máximo 2 días después de recibirlo. Sin embargo, se cambió esta idea por el uso un tablero. Con este tablero, la solicitante podía abrirlo y renovarlo cuando lo necesitara, sin necesidad de esperar a que el área de BI dé respuesta. Después de la reunión que se citó en Microsoft Teams con las personas elegidas para dar explicación de los tableros y para recibir retroalimentación de estos, se agregó un filtro, en la que se pueda seleccionar un rango de fecha de inactividad. El resultado del tablero es el que se observa en la figura 29.

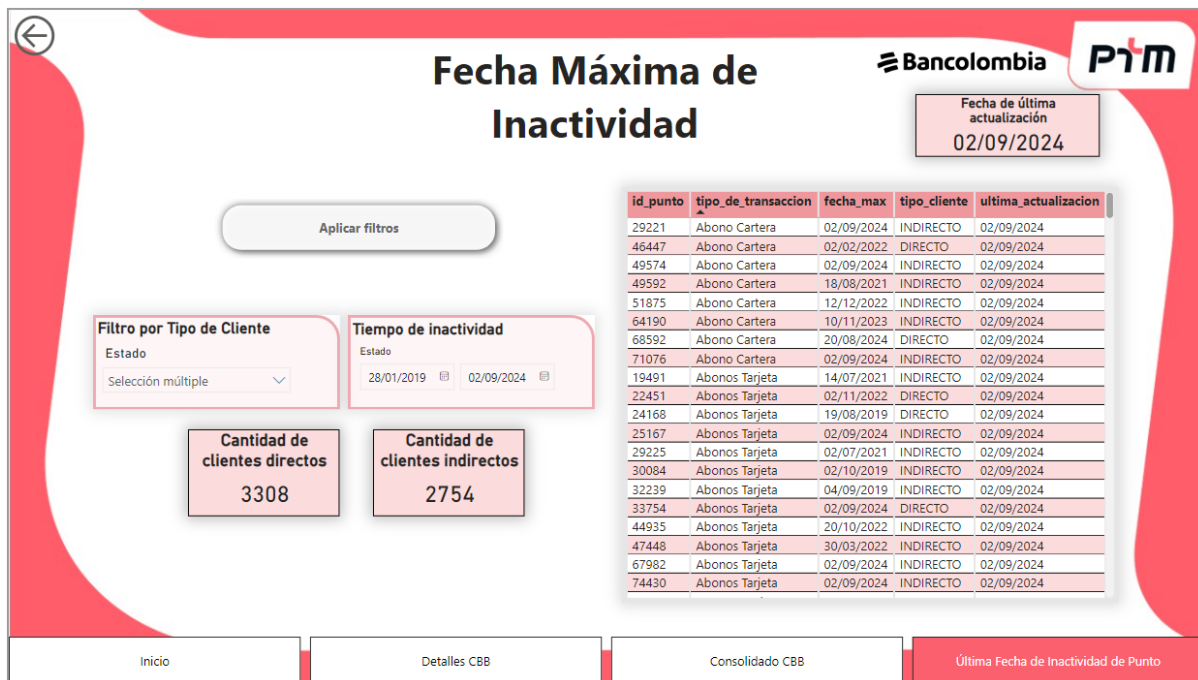


Figura 29. Tablero para solución del ticket con clasificación "fecha maxima inactividad".

La clasificación de "ventas" es automatizable, ya que va dirigida al área de BI, y mucha información que solicita no es información sensible, ya que respecta a información de ventas realizadas respecto a clientes, áreas geográficas, productos o estadísticas de ventas. Para esta área, se optó por realizar un tablero de Power BI con la información de ventas tanto para el caso de plataforma como para el caso de los corresponsales bancarios de Bancolombia. Con los tickets de ventas, se realizó un modelo de AI en el que se consultaban los campos más requeridos, y otro modelo en el que se preguntaba específicamente y de forma coloquial lo que el empleado de la empresa requería, y para qué.

Con la información que arrojaron los modelos para los casos de los tickets de clasificación de "ventas", se encontraron tres tipos de ventas fundamentales que generalizaban los tickets de ventas: detalle de ventas, con solidado de ventas por meses y consolidado de ventas por días. También se encontró que las ventas se consultaban respecto a los códigos o id de detallistas, distribuidores, mayoristas, respecto al departamento, municipio, productos y clasificaciones de productos, además se encontró que en todas las solicitudes de ventas se especificaban rangos de fechas.

Además, en los campos que se encontraron, se halló que era viable usar todos los campos solicitados salvo por tres campos (destino_venta, dirección y georreferenciación), ya que estos también eran información sensible del usuario.

Se pensaron en posibles soluciones, y se llegó a la conclusión de que la mejor opción era la realización de tableros de Power BI para que los empleados que solicitaran esta información realizaran una auto-gestión. Los tableros serían automatizados cada vez que el empleado de la empresa realizara una consulta por fecha. Para lograr esto, se hizo una partición de las tablas de ventas y de las tablas de corresponsales.

Al realizar la partición, se hizo un chequeo y reorganización de las tablas y los códigos automatizados que ya poseía la empresa, ya que al no tener filtro por fecha se hacía una consulta de toda la tabla, y se debía cambiar el código cuando se tomaba en cuenta el día actual en los query que se realizaban en BigQuery con la función "current".

Se planteó el uso de 4 hojas, los cuales se dividían en 2 tableros. Un tablero era respecto a plataforma, y el otro tablero para corresponsales de Bancolombia. En estos, la primera hoja hacía referencia al detalle de las ventas desde hace 3 meses (se eligió 3 meses porque con este rango de tiempo se determinaban los detallistas que estaban vigentes, al realizar una venta en este lapso de tiempo). Desde esta misma hoja se podía consultar el consolidado de ventas con días específicos, ya que se agregó estadísticas (una de líneas de barras horizontales para las ciudades y departamentos, otra línea de barras horizontal para productos y clasificaciones de productos, y un diagrama de sectores, en el que se consultaban por detallistas). En todos los diagramas, se puede exportar la información como archivo de Excel, y en el diagrama de sectores, al exportar la información se exportaban además los campos de cod_detallsita, id_detallista, cod_distribuidor, id_distribuidor, cod_mayorista e id_mayorista. Si se requerían más campos, se podía hacer un cruce sencillo entre estas estadísticas.

Con los tableros realizados, la nueva arquitectura tecnológica es la observada en la figura 30.

Los tableros resultantes para las ventas en plataforma se observan en las figuras 30, 31 y 32, y en las figuras 33, 34 y 35 se observan los tableros para el caso de las ventas por correspondencia bancaria de Bancolombia.

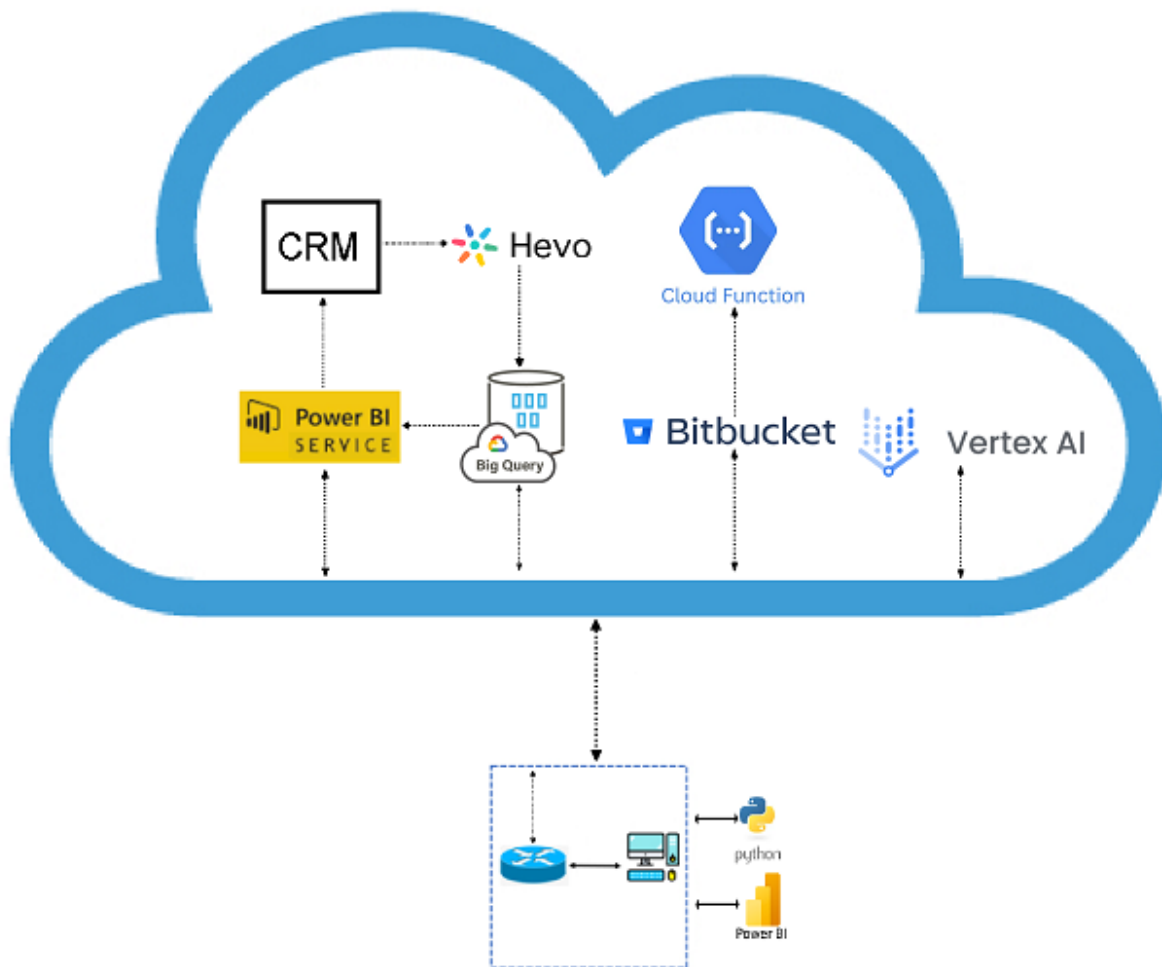


Figura 30. Nueva arquitectura tecnológica después de la automatización de tablas de Power BI para los tickets de área de BI.

5.2.1.1 Ventas de plataforma

Se presentan los tableros resultantes para las ventas en plataforma, correspondiendo al área de BI de la empresa Comercial Card S.A.S. Los tableros se observan en las figuras 31, 32 y 33.

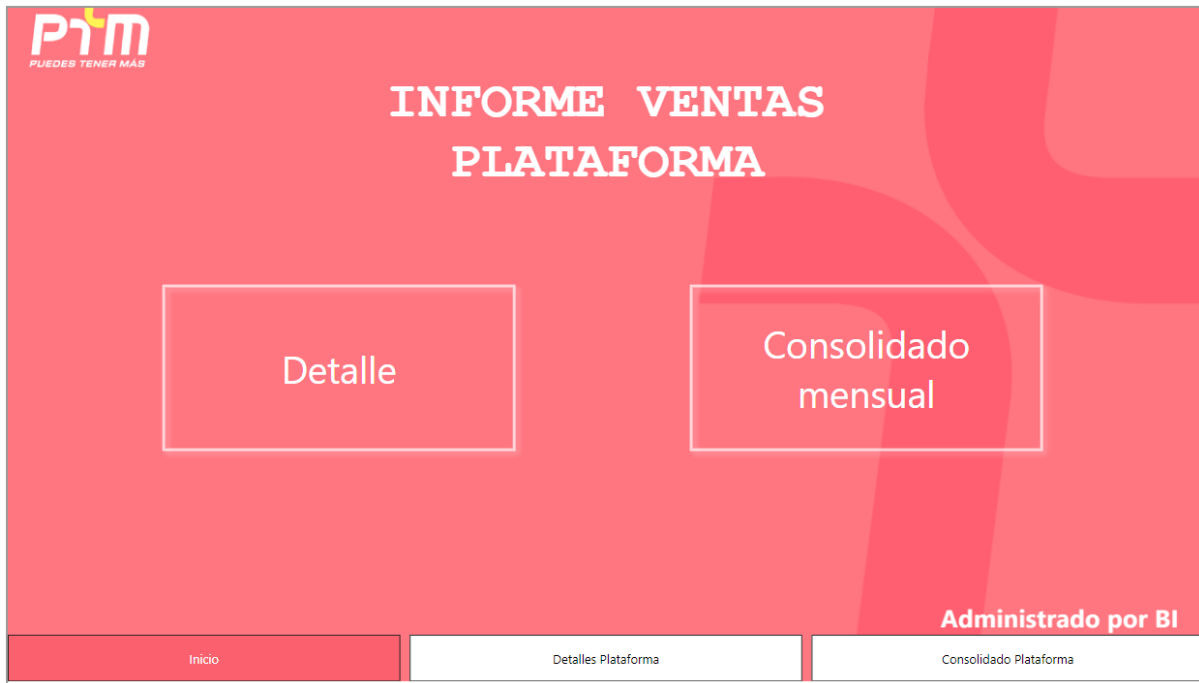


Figura 31. Inicio de tablero para ventas de plataforma.

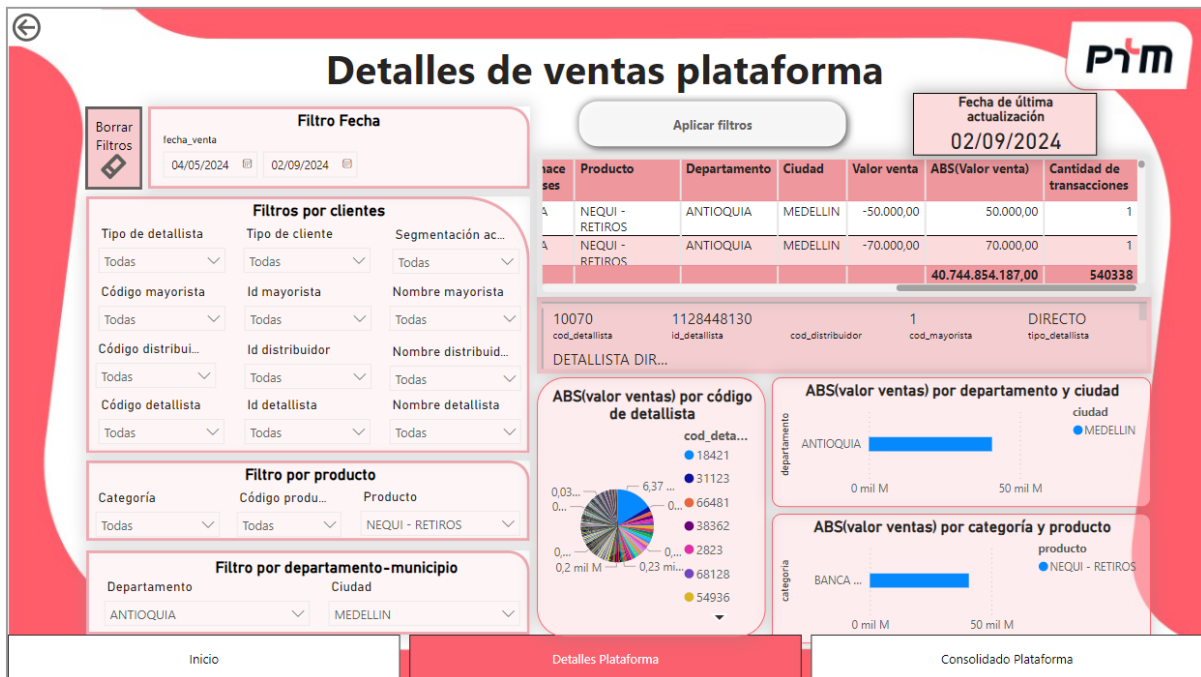


Figura 32. Hoja de detalles de las ventas de plataforma.

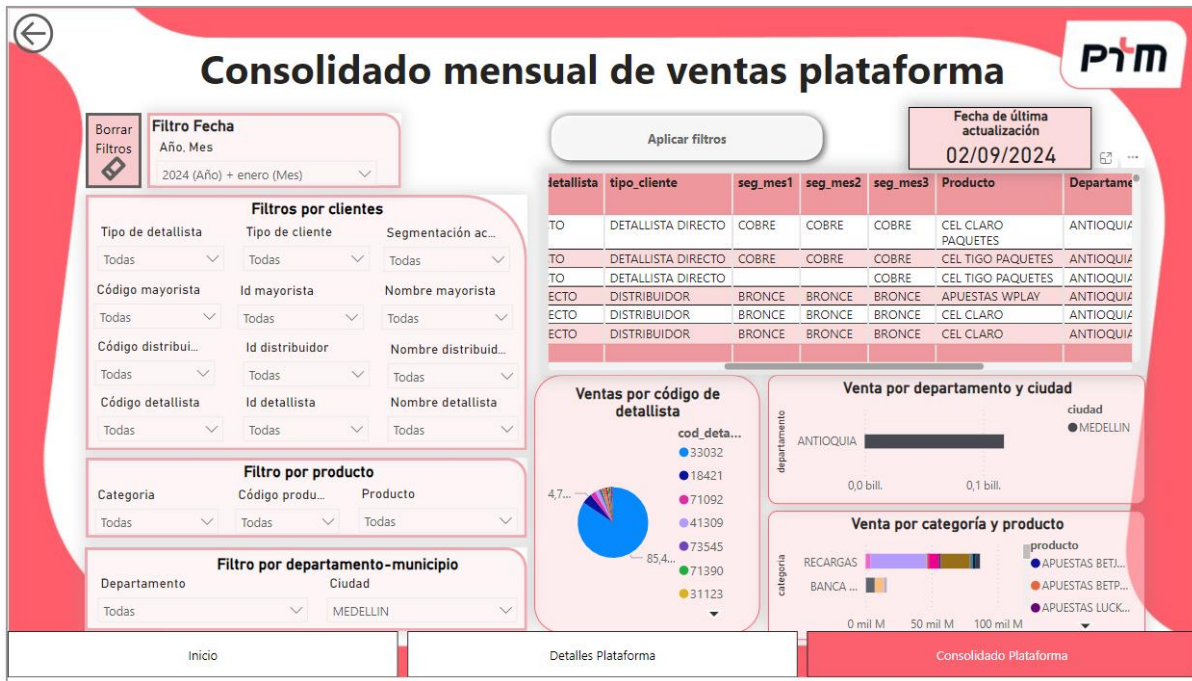


Figura 33. Hoja de consolidado de las ventas de plataforma.

5.2.1.2 Ventas de corresponsalía bancaria de Bancolombia (CBB)

Se presentan los tableros resultantes para las ventas por corresponsalía bancaria de Bancolombia, correspondiendo al área de BI de la empresa Comercial Card S.A.S. Los tableros se observan en las figuras 34, 35 y 36.

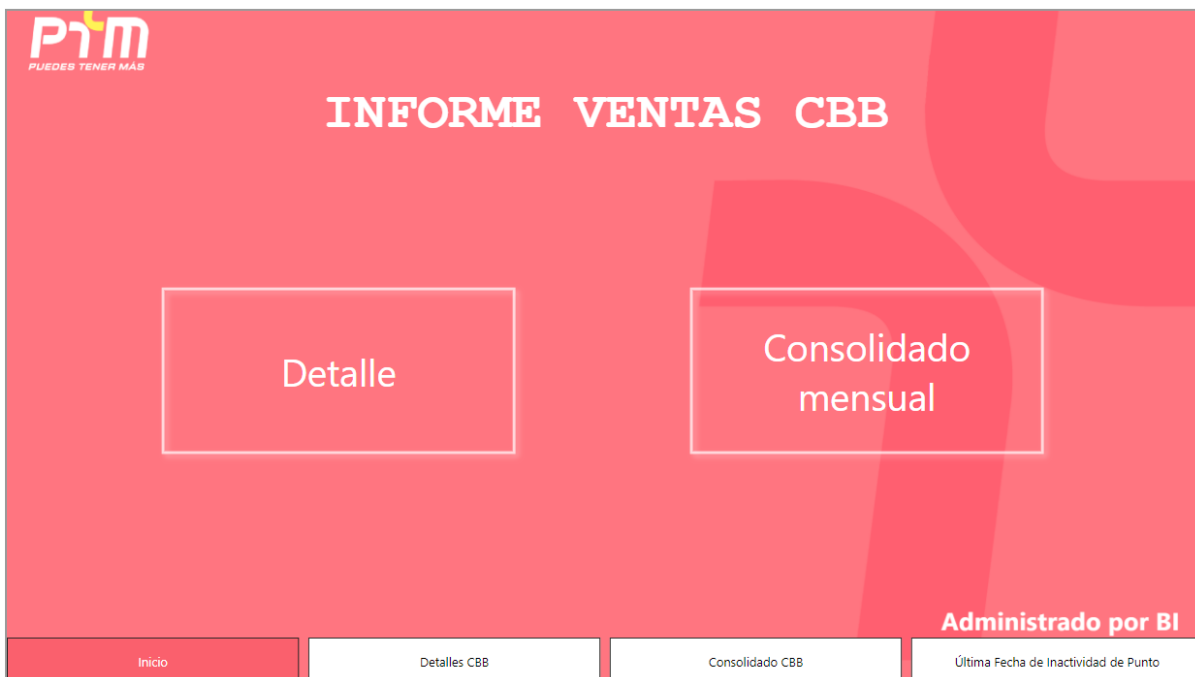


Figura 34. Inicio de tablero para ventas de corresponsales bancarios Bancolombia (CBB).

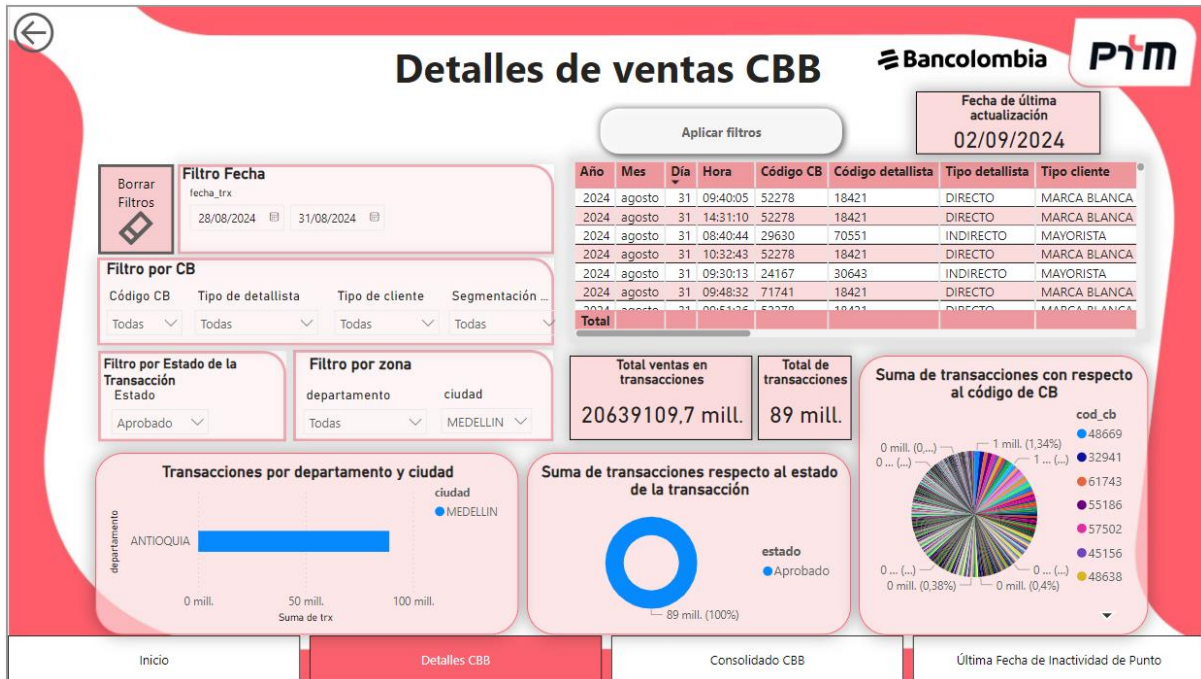


Figura 35. Hoja de detalles de las ventas de CBB.

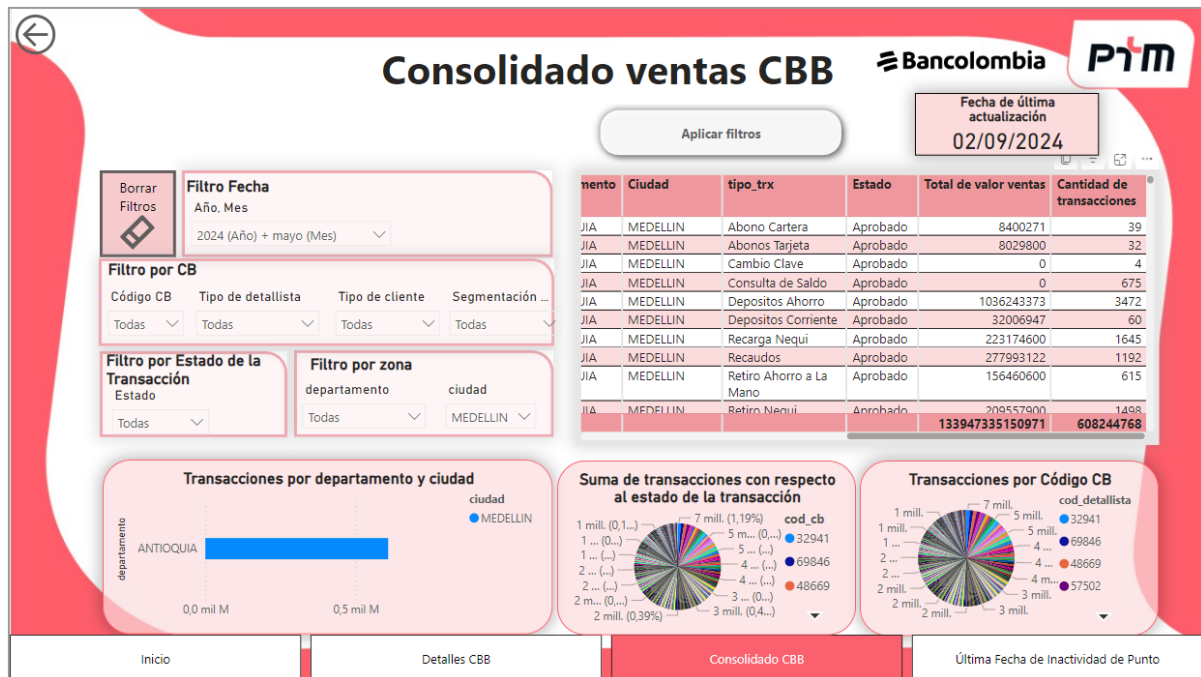


Figura 36. Hoja de consolidado de las ventas de CBB.

La hoja para la fecha máxima de inactividad se agregó como última hoja al tablero de CBB. Esto se observa en la figura 29.

5.2.2 Área de IT

Para el área de IT se hizo un detallado de los tickets, especificando características, ya que eran muchos tickets y se quería tener información de esto y tener gráficas para que el área de IT (que es separada al área de las prácticas en BI) las estudiara y tomaran sus decisiones.

Observando las clasificaciones primarias de estos tickets, no se encontraron tickets a automatizar, ya que para los tickets solicitados ya existían automatizaciones de envíos, y el resto de los tickets debían ser verificados para saber si dar solución o no.

Se encontró, sin embargo, que las clasificaciones "solicitud informes" y "comisiones llegaban al área de IT, debiendo haber sido solicitados para BI, por lo que se hizo esa aclaración al área de IT. Se mostró información de que el 14,4% de los tickets totales son de estas dos clasificaciones. Junto con la información proporcionada de las estadísticas, se observan los solicitantes de estas clasificaciones para que el área tome las medidas que desee, proponiendo que les den información de que esos tickets son para el área de BI y cómo solicitarlos.

Se hicieron estadísticas de los tickets con clasificación y sub-clasificaciones y se presentaron al área de IT. Las estadísticas que se presentaron de los archivos son los que se consideraron los más relevantes y son los mostrados en las figuras 37, 38, 39, 40 y 41:

CANTIDAD DE TICKETS POR CLASIFICACIÓN EN ÁREA TI.

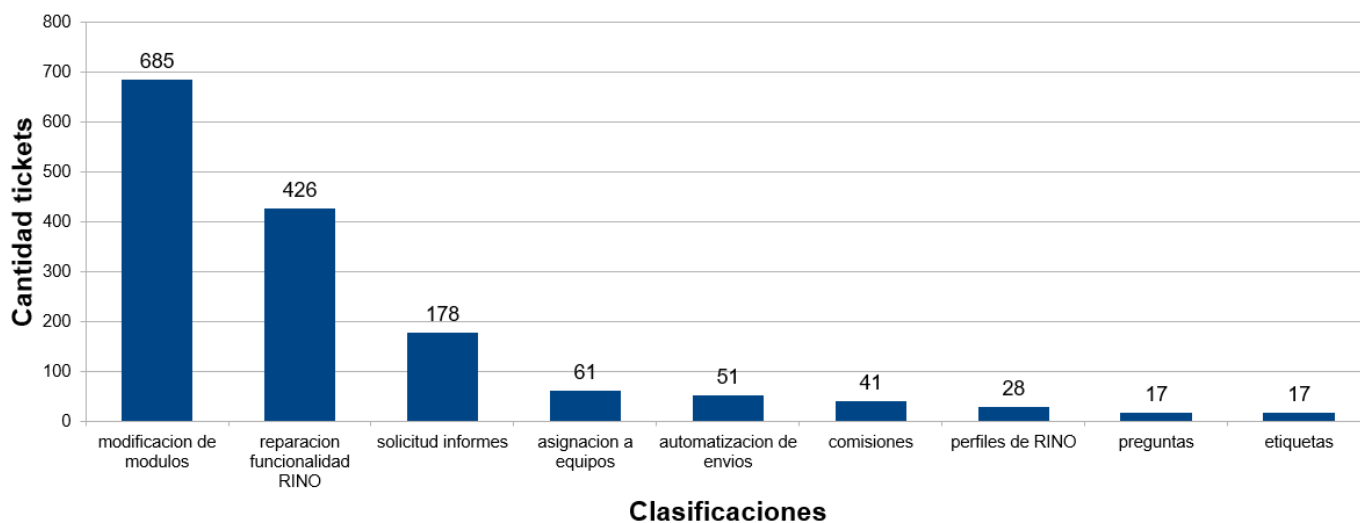


Figura 37. Cantidad de tickets por clasificación principal en el área de IT.

Al obtener las estadísticas de la figura 37, se buscó las mayores solicitudes de las dos clasificaciones más solicitadas, observando las sub-clasificaciones.

TICKETS DE CLASIFICACIÓN "MODIFICACION DE MODULOS" CON MAYOR CANTIDAD.

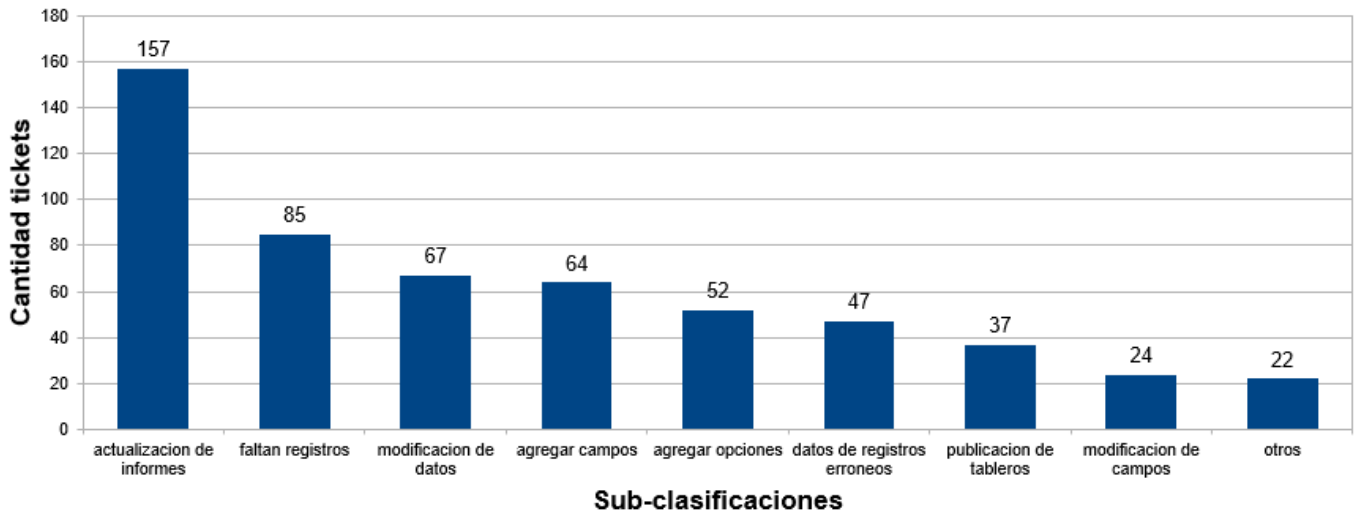


Figura 38. Cantidad de tickets por sub-clasificación de la clasificación más solicitada (modificacion de modulos) en el área de IT.

TICKETS DE CLASIFICACIÓN "REPARACION FUNCIONALIDAD RINO" CON MAYOR CANTIDAD.

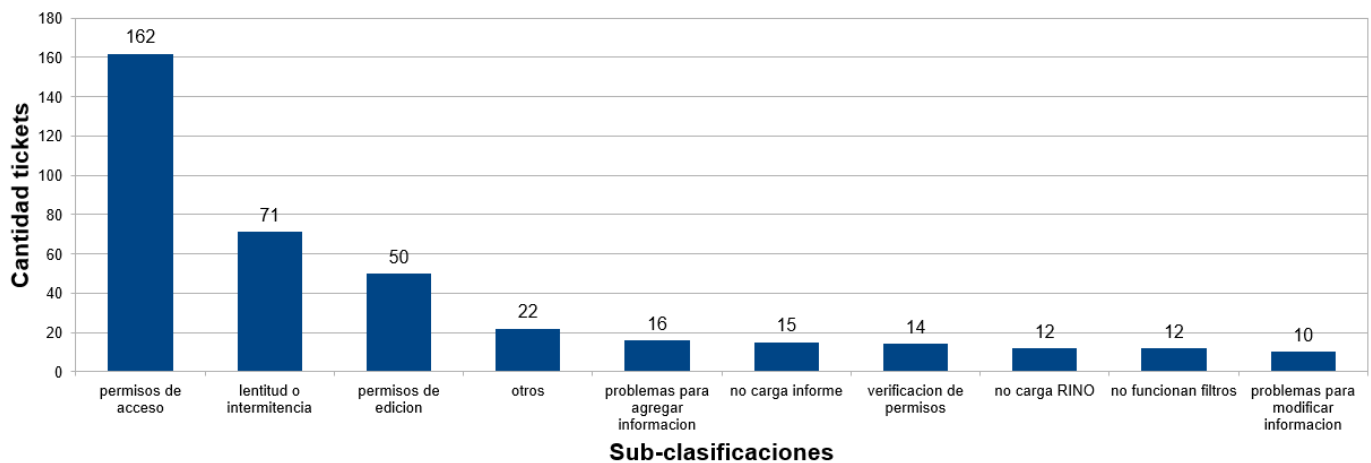


Figura 39. Cantidad de tickets por sub-clasificación de la segunda clasificación más solicitada (reparacion funcionalidad rino) en el área de IT.

Además, de observar las clasificaciones más solicitadas, se realizaron estadísticas de la cantidad de tickets respecto al tiempo que fueron recibidos en el área de IT y que debieron ser dirigidos al área de BI. Estas se observan en las figuras 40 y 41.

CANTIDAD DE TICKETS CON CLASIFICACIÓN "SOLICITUD INFORMES" POR MES.

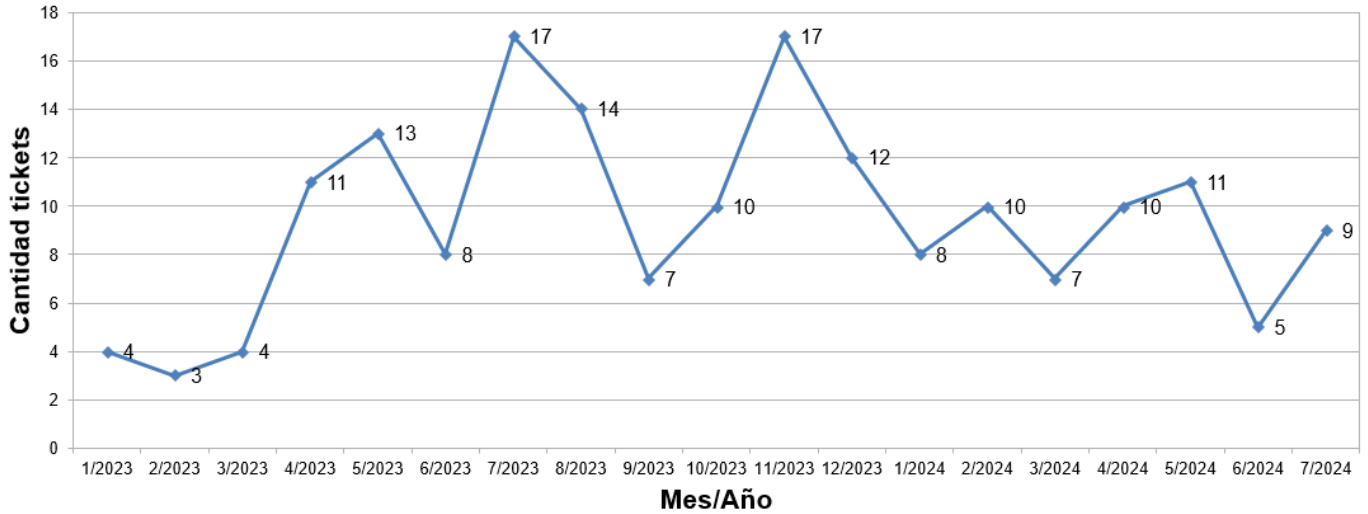


Figura 40. Cantidad de tickets respecto al tiempo de la clasificación "solicitud informes".

CANTIDAD DE TICKETS CON CLASIFICACIÓN "COMISIONES" POR MES.

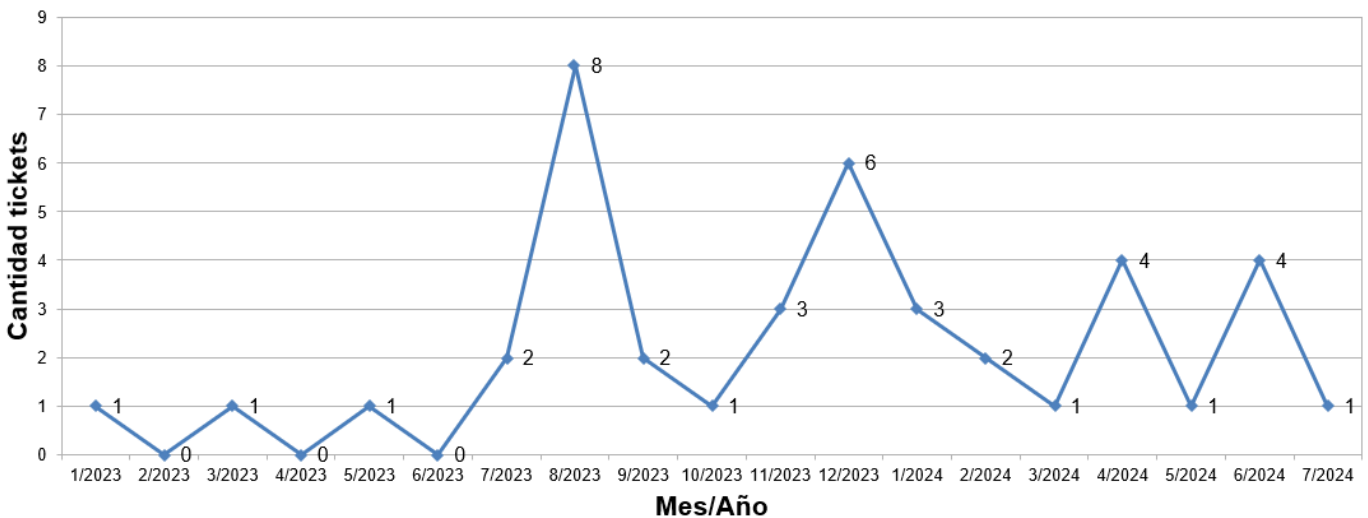


Figura 41. Cantidad de tickets respecto al tiempo de la clasificación "comisiones".

Se me informó que en la empresa se había echo una aclaración de las funciones de las áreas de IT y de BI en agosto del 2023, y que esta podría ser la razón por la que en ambas clasificaciones de tickets se tenía un descenso después de agosto de dicho año.

Se hizo una entrega de un Excel con las clasificaciones, sub-clasificaciones, y detalles de los tickets (con módulo a corregir, o producto requerido, o tipo de filtro a aplicar. Con este archivo de Excel se puede encontrar mucha información de los tickets que llegan al área de IT, con la que esta área puede consultar medidas a gusto.

6 CONCLUSIONES

Después de realizado el proyecto anterior, se encontraron las siguientes conclusiones:

6.1 *Requerimientos funcionales y no funcionales*

- La metodología adoptada permitió una clara definición de los requerimientos funcionales y no funcionales para el desarrollo del proyecto.
- En cuanto a los requerimientos funcionales, se estableció un enfoque en la clasificación automática de tickets mediante un modelo de inteligencia artificial (AI), con un fuerte énfasis en la actualización diaria de las bases de datos y la segmentación de los tickets por área (BI e IT). Este proceso garantiza una mayor eficiencia y organización de los datos históricos y recientes, asegurando una base sólida para la toma de decisiones.
- Desde la perspectiva no funcional, el proyecto cuenta con limitaciones claras, como la ausencia de una interfaz gráfica, la falta de autorización de usuarios y la inexistencia de manuales de usuario. A pesar de esto, se destaca la importancia de seguir buenas prácticas de documentación interna en el código, siguiendo las normas PEP8 y docstrings para Python, lo que facilita la comprensión y mantenibilidad futura del sistema.

6.2 *Evaluación y selección de algoritmos de AI*

- Se concluye que la inteligencia artificial adecuada para la clasificación de tickets es una que se basa en patrones y estructuras de datos, en lugar de enfoques predictivos, lo que se alinea con las necesidades del proyecto.
- La elección de la API (Chat Vertex AI Studio) recibe instrucciones detalladas, lo que la hace flexible para la clasificación de tickets, y esta flexibilidad fue importante en el modelo de AI. Con una API estructurada no se podían obtener variaciones grandes en la exactitud de la API. Por esto, se concluye que, si se usa una API específica para una labor, se revisen los resultados de esta, ya que después los resultados no podrán aumentar en exactitud.

6.3 *Desarrollo de modelo de AI para clasificación de tickets*

- La arquitectura tecnológica definida para la conexión entre los diferentes componentes (Python, Vertex AI, BigQuery, CRM y BitBucket) asegura un flujo de información bidireccional fluido y controlado. Este diseño permite que las consultas y resultados fluyan de manera eficiente entre los sistemas, asegurando que la AI tenga acceso a los datos necesarios para realizar clasificaciones precisas. Además, se evita una sobrecarga innecesaria en el sistema, manteniendo las conexiones unidireccionales cuando es posible, lo que mejora el uso de recursos.
- La clasificación manual de los tickets históricos fue esencial para proporcionar ejemplos concretos al modelo de AI, lo que permitió calibrar y ajustar el sistema con base a la precisión y adecuación a las necesidades de la empresa.

- En el área de IT, se gestionó un alto volumen de tickets mediante la creación de otros modelos de AI usando prompts con ejemplos de 200 tickets. Esto ayudó a contextualizar las solicitudes y a definir clasificaciones adecuadas.
- No se puede esperar que un modelo de AI arroje valores de exactitud altos, ya que puede haber un overfitting, y no funcionar para prompts diferentes a los de prueba.
- Se encontraron especificaciones de la API que reducen significativamente la exactitud del sistema, o que bloquean su funcionamiento, impidiendo que el sistema arroje resultados, por lo que es importante estudiar estas especificaciones de la API, y el código que se usa para ejecutar la API.
- La inclusión de un contexto aclaratorio, que detalle las directrices sobre cómo se debían realizar las clasificaciones más complejas, es un factor importante para guiar a la AI en situaciones donde la ambigüedad puede surgir, y además aumenta significativamente la exactitud de los modelos de AI.
- La priorización de las clasificaciones y el cambio en estas priorizaciones fue el factor que más modificó la exactitud del modelo de AI en todos los modelos estudiados.
- La separación de clasificaciones o la consolidación de estas generó un alto cambio en la exactitud de los modelos de AI.
- El uso de la matriz de incertidumbre fue fundamental para evaluar las clasificaciones que presentan más errores.

6.4 Automatización de procesos

- No todos los procesos pueden ser automatizados, especialmente aquellos que involucran información sensible, o aquellos que requieren intervención presencial. En estos casos, la intervención humana sigue siendo esencial para garantizar la confidencialidad y complemento de supervisión humana.
- La creación de tablas separadas para tickets históricos y tickets diarios es una estrategia para gestionar y analizar datos de diferentes periodos, facilitando la comparación y el análisis a lo largo del tiempo.
- La clasificación de los tickets de IT en sub-clasificaciones y detalles específicos muestra un enfoque metódico que permite una categorización más precisa y útil. Esto es esencial para una gestión efectiva de los tickets.
- El análisis estadístico de los tickets clasificados proporciona información que puede ser utilizada por el área de IT para identificar tendencias, áreas problemáticas y oportunidades de mejora en el servicio.
- La implementación de tableros de Power BI para las solicitudes de ventas demuestra que es posible automatizar con éxito procesos que no implican información delicada. Este enfoque no solo reduce la carga de trabajo del área de BI, sino que también es útil para

los empleados, al permitirles auto-gestionar sus consultas y acceder a datos actualizados en tiempo real, sin necesidad de montar un ticket y esperar su respuesta. Este tipo de automatización destaca cómo la tecnología puede optimizar operaciones repetitivas y de bajo riesgo.

- La clasificación errónea de tickets entre áreas (BI y IT) revela la necesidad de una mejor comunicación y definición de roles. Aunque se lograron avances en la reducción de estos errores tras la aclaración de responsabilidades en agosto de 2023, la existencia de un 14,4% de tickets mal clasificados antes de esa fecha destaca la importancia de asegurar que los procesos sean claros para todos los empleados. Esto también muestra que las automatizaciones deben ir acompañadas de esfuerzos para educar a los usuarios.

Departamento de Ingeniería Electrónica y de Telecomunicaciones

AI PARA MEJORAR EL ACCESO A LA INFORMACIÓN DE LAS ÁREAS DE BI E IT DE COMERCIAL CARD S.A.S.



UNIVERSIDAD DE ANTIOQUIA

Facultad de Ingeniería

PRACTICANTE: Yesika Milena Carvajal Díaz

ASESOR: Luis Germán García Morales

PROGRAMA: Ingeniería Electrónica

Semestre de la práctica: 2024-1

Introducción

Comercial Card S.A.S es una empresa enfocada en ofrecer servicios transaccionales de telefonía móvil, tecnología y servicios públicos, a través de su Plataforma Tecnológica Multiservicios, PTM. En la empresa se tienen las áreas de BI (Business Intelligence) e IT (Information Technology).

Las áreas BI e IT tienen un mecanismo de generación de "tickets" o solicitudes, que permite el orden y manejo de las solicitudes de otras áreas. Se tomaron muestras de la cantidad de tickets que se reciben, y se obtuvo que el promedio mensual de tickets es de 41.9 para el área de BI, y 184.043 para TI, siendo una cantidad significativa de tickets,

Problema y Solución Implementada

Existe la necesidad de automatizar el proceso de clasificación de solicitudes a otras áreas debido a la complejidad de las peticiones y a la creciente demanda. Además, informes previos mostraron que el procesamiento manual de tickets genera retrasos y errores, afectando la satisfacción del cliente y la productividad del equipo. Mediante la integración de las APIs de Google Cloud Vertex AI, se abordó el problema de selección de tickets mediante la creación de un modelo de AI capaz de clasificar solicitudes de manera precisa y eficiente.

La automatización de la clasificación de tickets no solo mejoró el uso de recursos humanos, sino que también mejoró la capacidad de respuesta de la empresa ante las necesidades del cliente, estableciendo un modelo escalable que puede adaptarse a futuras demandas del mercado.



Metodología

Involucra la recolección de datos de tickets previos y la iteración en análisis y ajuste del modelo AI a partir de matriz de incertidumbre y exactitud.

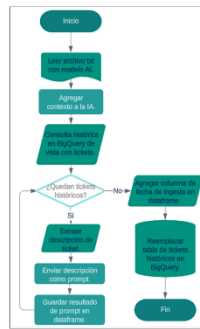
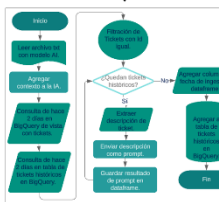


Diagrama de flujo para ingesta histórica de tickets a BigQuery.

Diagrama de flujo para actualización de tickets.



Resultados

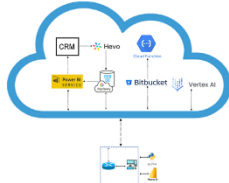
Área BI

Por tipo de servicio

Exactitud: 84,44%



Arquitectura tecnológica del proyecto



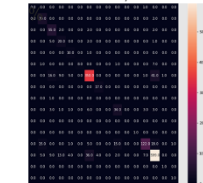
Por tipo de petición

Exactitud: 80,08%



Área IT

Exactitud: 82,64



Tableros de Power BI para solución de tickets



Objetivos

Objetivo general

Desarrollar un sistema que permita la clasificación y respuesta automática de los tickets recibidos en las áreas de Business Intelligence (BI) e Information Technology (IT).

Objetivos específicos

- ✓ Definir los requerimientos funcionales y no funcionales.
- ✓ Evaluar varios algoritmos de AI y seleccionar dos que realicen adecuadamente la clasificación y generación de respuestas automáticas de los tickets recibidos
- ✓ Desarrollar el modelo para la clasificación de los tickets, utilizando la herramienta de Vertex AI y la descripción dada por los solicitantes en los tickets,
- ✓ Desarrollar y evaluar un programa en Python que implemente el modelo propuesto junto con los algoritmos de AI seleccionados, para la clasificación y respuesta automática de tickets.

Conclusiones

- ✓ La metodología adoptada permitió una clara definición de los requerimientos funcionales y no funcionales para el desarrollo del proyecto.
- ✓ La API elegida (Chat Vertex AI Studio) recibe instrucciones detalladas, lo que la hace flexible para la clasificación de tickets. Esto fue importante para obtener variaciones en resultados y exactitud.
- ✓ La matriz de incertidumbre fue fundamental para evaluar y mejorar las clasificaciones.
- ✓ La implementación de tableros de Power BI para las solicitudes de ventas demuestra que es posible automatizar con éxito procesos que no implican información delicada.
- ✓ El análisis estadístico de los tickets clasificados en el área de IT proporciona información que puede ser utilizada para identificar tendencias, tickets problemáticos y oportunidades de mejora.

DATOS DE CONTACTO DEL AUTOR:



+57 3195298298



yesika.carvajal@udea.edu.co



http://ca.linkedin.com/in/yesika-milena-Carvajal-d%C3%A1z.

Referencias Bibliográficas

- [1] "AUTOMATIZACIÓN DEL PROCESO DE SOLICITUDES DE ACTIVOS TECNOLÓGICOS EN EL BANCO DE CRÉDITO CENTROAMERICANO". Formato PDF. Plataforma ribuni. Fecha de publicación: 27 de Julio de 2012. Fecha de citación: 28 de Julio de 2024. Anexo: <https://ribuni.uni.edu.ni/1080/1/38096.pdf%20>.
- [2] "AUTOMATIZACIÓN SOLICITUDES PEDIDOS Y COMPRAS PROYECTOS CORONA". Formato PDF. Plataforma sistemas uniandes. Fecha de publicación: 10 de Marzo de 2022. Fecha de citación: 28 de Julio de 2024. Anexo: <https://sistemas.uniandes.edu.co/images/NOTICIAS/2022/03/grupo-corona.pdf>.
- [3] "Ficha 20: Automatización de la clasificación de solicitudes que recibe el área de soporte interno de Agesic". Informe electrónico. Plataforma Agencia de Gobierno Electrónico y Sociedad de la Información y del Conocimiento. Fecha de publicación: 15 de Julio de 2024. Fecha de citación: 29 de Julio de 2024. Anexo: <https://www.gub.uy/agencia-gobierno-electronico-sociedad-informacion-conocimiento/comunicacion/publicaciones/ficha-20-automatizacion-clasificacion-solicitudes-recibe-area-soporte>.
- [4] "Requerimientos funcionales y no funcionales en Scrum". Plataforma Metodologías Ágiles. Potcast en línea. Fecha de citación: 19 de Abril de 2024. Anexo: <https://metodologiasagiles.org/requerimientos-funcionales-y-no-funcionales-en-scrum/>.
- [5] "Qué es Inteligencia Artificial Generativa?". Video en página WEB. Plataforma YouTube. Fecha de publicación: 16 de Octubre de 2023. Fecha de citación: 19 de Abril de 2024. Anexo: <https://www.youtube.com/watch?v=srDIV4o9tKU>.
- [6] "INTELIGENCIA ARTIFICIAL PREDICTIVA VS. GENERATIVA | IA Básico | AprendelA". Video en página WEB. Plataforma YouTube. Fecha de publicación: 16 de Agosto de 2023. Fecha de citación: 01 de Julio de 2024. Anexo: <https://www.youtube.com/watch?v=hR37QFhIN3w>.
- [7] "¿Qué es Python? – Explicación del lenguaje Python". Potcast en línea. Plataforma AWS. Fecha de citación: 22 de Abril de 2024. Anexo: <https://aws.amazon.com/es/what-is/python/#:~:text=Python%20es%20un%20lenguaje%20de,ejecutar%20en%20muchas%20plataformas%20diferentes>.
- [8] "Introduccion a Vertex AI | Vertex AI". Potcast en línea. Plataforma Google Cloud. Fecha de citación: 30 de Julio de 2024. Anexo: <https://cloud.google.com/vertex-ai/docs/start/introduction-unified-platform?hl=es-419>.
- [9] "Diagrama de flujo". Potcast en línea. Plataforma Diagrama de flujo. Fecha de citación: 22 de Abril de 2024. Anexo: <https://www.diagramasdeflujo.com/>.
- [10] "¿Qué son los diagramas de arquitectura? - Explicación de los diagramas de arquitectura de software y sistemas". Potcast en línea. Plataforma AWS. Fecha de citación: 22 de Abril de 2024. Anexo: <https://aws.amazon.com/es/what-is/architecture-diagramming/>.
- [11] "BigQuery: Almacén de datos empresarial". Potcast en línea. Plataforma Google Cloud. Fecha de citación: 22 de Abril de 2024. Anexo: [https://cloud.google.com/bigquery/?utm_source=bing&utm_medium=cpc&utm_campaign=latam-CO-all-es-dr-BKWS-all-7 all-trial-b-dr-1707800-LUAC0020563&utm_content=text-ad-none-any-DEV_c-CRE_-ADGP_HYBRID+%7C+BKWS+-+BRO+%7C+Txt_Data+Analytics-BIGQUERYKWID_43700079471062399-kwd-78546796644961:loc43&utm_term=KW_BigQuery+GoogleST_BigQuery+Google&gclid=bac9049e34b9119938bf62b85c2e1dd2&gclid=3p.ds&hl=es_419](https://cloud.google.com/bigquery/?utm_source=bing&utm_medium=cpc&utm_campaign=latam-CO-all-es-dr-BKWS-all-7%20all-trial-b-dr-1707800-LUAC0020563&utm_content=text-ad-none-any-DEV_c-CRE_-ADGP_HYBRID+%7C+BKWS+-+BRO+%7C+Txt_Data+Analytics-BIGQUERYKWID_43700079471062399-kwd-78546796644961:loc43&utm_term=KW_BigQuery+GoogleST_BigQuery+Google&gclid=bac9049e34b9119938bf62b85c2e1dd2&gclid=3p.ds&hl=es_419).

-
- [12] "Qué es SQL". Potcast en línea. Plataforma Aprende SQL en Español. Fecha de citación: 14 de Abril de 2024. Anexo: [https://www.sqleasy.com/es/tutorial/que-es-sql/#:~:text=SQL%20es%20un%20lenguaje%20de,%20datos%20relacionales%20\(RDBMS\)](https://www.sqleasy.com/es/tutorial/que-es-sql/#:~:text=SQL%20es%20un%20lenguaje%20de,%20datos%20relacionales%20(RDBMS)).
- [13] "¿Qué es la automatización?". Potcast en línea. Plataforma IBM. Fecha de citación: 14 de Abril de 2024. Anexo: <https://www.ibm.com/eses/topics/automation>.
- [14] "Beneficios de la depuración de datos | DEYDE Calidad de Datos". Potcast en línea. Plataforma deyde datacentric. Fecha de citación: 22 de Abril de 2024. Anexo: <https://deyde.com/blog/calidad-datos/depuracion-dedatos-que-beneficios-aporta/>.
- [15] "Filtro en Informática: Definición, Tipos y Aplicaciones Clave". Potcast en línea. Plataforma Academia Rubicon. Fecha de citación: 22 de Abril de 2024. <https://academiarubicon.es/filtro-definicion-informatica/>.
- [16] "Filtra y limita datos | Looker". Potcast en línea. Plataforma Google Cloud. Fecha de citación: 22 de Abril de 2024. Anexo: <https://cloud.google.com/looker/docs/filtering-and-limiting?hl=es-419>.
- [17] "Claude 3.5 Sonnet – Vertex – AI – My First Project – Consola de Google Cloud". Plataforma Google Cloud. Fecha de citación: 14 de Septiembre de 2024. Anexo: <https://console.cloud.google.com/vertex-ai/publishers/anthropic/model-garden/claude-3-5-sonnet?project=plenary-glass-389023>.
- [18] "Content classification – Vertex AI – Consola de Google Cloud". Plataforma Google Cloud. Fecha de citación: 14 de Septiembre de 2024. Anexo: <https://console.cloud.google.com/vertex-ai/publishers/google/model-garden/language-v1-classify-text-v1>.
- [19] "Vertex AI – My First Project- Consola de Google Cloud". Plataforma Google Cloud. Fecha de citación: 14 de Septiembre de 2024. Anexo: <https://console.cloud.google.com/vertex-ai/studio/chat?project=plenary-glass-389023>.