



Desarrollo de una Aplicación Distribuida para el Apoyo a la Categorización de Datos Mediante Análisis Exploratorio

Franck Jhonatan Burbano, Leidi Yakeline Pereira

franckj07@gmail.com,

leidipere@gmail.com

Ingeniería Informática

Facultad de Ingeniería

Corporación Universitaria Autónoma de Nariño

Pasto, Nariño

Asesor Metodológico: Betty Magaly Mora Revelo

Asesor Técnico: Héctor Andrés Mora

RESUMEN

En el presente proyecto damos a conocer la importancia y manejo de grandes cantidades o volúmenes de datos. Diariamente son obtenidos por diferentes medios, lugares y métodos, convirtiéndose aquella información en masiva o un Big Data, por lo que se requiere que ésta sea correctamente organizada, procesada y clasificada para obtener un resultado claro. Mediante el apoyo de las TIC (Tecnologías de la Información y las Comunicaciones), la tecnología ha permitido realizar cambios a nivel del manejo de la información, generando una constante búsqueda a la buena administración de grandes volúmenes de datos. Para el apoyo en una correcta toma de decisiones mediante la interpretación de la información, es de gran importancia manejarla mediante un sistema distribuido, el cual genera una serie de características positivas de gran impacto para esta herramienta y el cómo se aplica en la transformación de datos a través de la categorización, donde se requiere tener una base de datos correctamente procesada. En algunas empresas se utilizan diferentes herramientas que ayudan tanto en la calidad de datos, como en la organización que se necesita para manejarla y poder tener a mano la información que sea requerida de forma rápida y confiable, con ello se minimiza el riesgo de error que se tiene por el mal manejo y al tener un correcto procesamiento en su calidad con herramientas adecuadas a su volumen y organización son más confiables y acertadas al momento de la toma de decisiones. Uno de estos procesos organizacionales es categorizar la información, en este caso en específico se han utilizado diferentes formatos que en conjunto nos facilitan una herramienta modular que con ayuda de un gestor de base de datos Postgres, un servicio en la nube o API el cual permite ser más accesible y la conexión a GUI o frontend, que muestra la vista al usuario o consumidor final, generando una interacción y decida la categorización según su criterio, estas características nos ayudan en la toma de decisiones mediante un correcto procesamiento de datos que obtenga información específica y clara, dando mayor confianza, minimizando el riesgo de error en la decisión y generando una serie de novedades que ayudan a la aplicación con un mejor rendimiento, como el sistema distribuido, el cual trabaja de forma modular, es multiplataforma, que permite interactuar con aplicaciones móviles, haciendo que cada una de sus fases sea independiente de la otra, que quiere decir esto, si alguno de los elementos falla, los otros podrán seguir realizando la función correctamente sin depender su trabajo de otra fase.

Palabras clave: Big Data, Categorización, TIC, Postgres, Api, GUI, Toma de decisiones.

I.	Introducción	3
II.	Descripción del Problema	5
II-A.	Formulación del problema	7
III.	Justificación	7
IV.	Objetivos	8
IV-A.	Objetivo General	8
IV-B.	Objetivos Específicos	8
V.	Marco Teórico	9
V-A.	Tratamiento de datos	9
V-B.	Knowledge Discovery in Databases (KDD)	9
V-C.	Categorización de Datos	10
VI.	Marco Legal	11
VII.	Metodología	13
VII-A.	Plan de Acción	14
VIII.	Resultados, Análisis y Discusión	14
VIII-A.	Resultados	14
IX.	Fase 1. Definición de requerimientos	14
IX-A.	Diagramas UML	14
IX-A1.	Diagrama Arquitectónico	14
IX-A2.	Diagrama de Capas	15
IX-B.	Diagrama de Comportamiento	16
IX-B1.	Diagrama de Actividades	16
IX-C.	Diagrama Estructural	19
IX-C1.	Diagrama de clases	19
IX-C2.	Diagrama de paquetes	19
IX-C3.	Consultas en base de datos de tablas numéricas	20
IX-C4.	Clasificación de columnas en hechos y dimensión	20
X.	Fase 2. Clasificación de hechos probables y dimensiones	21
X-A.	Selección de hechos probables	22
X-B.	Creación de categorías	23
X-C.	Generar datos categorizados	24
XI.	Fase 3: Visualización de categorías	25
XI-A.	Conexiones con Api	25
XI-B.	Funcionamiento API	26
XI-C.	Generación de vistas	29
XI-D.	Validación y pruebas	33
XII.	Conclusiones y Recomendaciones	34
XII-A.	Conclusiones	34
XII-B.	Recomendaciones	34

XIII. Marcas Registradas	35
XIII-1. Node.js	35
XIII-2. PostgreSQL	35
XIII-3. Android	35
XIII-4. Kotlin	35
XIII-5. PL/pgSQL	36
XIII-6. Heroku	36
Anexos	37
A. Scripts de funciones medianamente acopladas al SGBD PostgreSQL	37
A1. Visualizar información base de datos	37
A2. Clasificación en hechos probables y dimensiones	37
A3. Función clasificación y creación de tabla categorizada	37
B. Código fuente del API	38
B1. Instalación paquetes	38
B2. Código index	38
B3. Configuración y rutas en categorías	39
B4. Login	39
C. Código fuente de la aplicación móvil y APK	40
C1. Login	40
C2. Copiar ruta	41
C3. Creación categorías	41
C4. Categorizar	42
C5. Tabla categorizada	42
D. Manual de instalación de cada producto anterior(Anexos externos)	42
Referencias	43

I. INTRODUCCIÓN

En la actualidad el volumen de datos que se maneja en todas las empresas o negocios por pequeños que sean, son de vital importancia ya que, al tener la información clara, organizada y de calidad, el resultado a la hora de tomar decisiones es más acertada y eficaz dependiendo de lo que se busque y con qué datos se requiere indagar. Con esto las empresas mueven cantidades gigantes de información y utilizan procesos de manejo que les ayuda a organizarla, extraer datos en maneras de informes y mejorar el rendimiento de la organización con la correcta toma de decisiones. Existen diferentes tipos de procesamiento de datos que ayudan a un rendimiento más eficaz ahorrando tiempo en el análisis si este se lo hiciera de manera manual.

Cuando un sistema se demora en el mejoramiento de la calidad y en la organización de la información, que de cierta manera son largos para cualquier sistema de información básico, tiende a presentarse inconvenientes en la lentitud a la hora de manejar una gran cantidad de datos y al mismo tiempo someterla a procesos largos, al no contar con suficiente espacio y herramientas adecuadas, no existe rapidez, al contrario, se vuelve pesado y tardío para mostrar algún resultado, es importante someter la información a estos procesos para que esta sea más acertada y confiable.

De esto depende que se realice una correcta toma de decisiones, la información debe ser procesada adecuadamente y con ello extraer los datos específicos que se requiera para analizarlos, para ello se ha optado trabajar con los sistemas distribuidos, quienes facilitan el proceso, en el que interactúan dos o más máquinas para cumplir con un mismo objetivo [1] en este caso con el procesamiento de la información desde una base sin tratamiento y de forma modular, recorriendo cada una de las herramientas realizadas por cada uno de los grupos como se observa en la figura 2, manteniendo el orden que cada uno tiene.

En Colombia existen trabajos donde se han utilizado sistemas distribuidos, con el fin de mejorar y facilitar el apoyo a la toma de decisiones, implementándolos como una herramienta que optimiza procesos y mejora la calidad de resultados en las empresas, como las PYMES quienes han utilizado esta tecnología para generar ventaj

competitivas [2], en Nariño como el estudio de desequilibrios regionales en Colombia, con el fin de conseguir información de calidad trabajando con diferentes equipos por un mismo fin [3].

La gran cantidad de información que se genera cotidianamente a consecuencia de herramientas que a diariamente son utilizadas se vuelve una situación que requiere el manejo de tecnologías que ayude a controlar tanto la calidad de datos, como la organización interna para una mayor facilidad en su manejo [4] y así facilitar la correcta toma de decisiones de una manera más eficiente y acertada, es por ello que se requiere de una herramienta que ayude con esta necesidad para que los datos sean claros y limpios, con una organización adecuada que nos ayude a un mejor registro sobre estos, se propone este proyecto, una herramienta modular que con ayuda de un sistema distribuido como se muestra en la figura 2, realice cada uno de los distintos pasos, y en específico la categorización de datos, por medio de un análisis exploratorio de la información previamente procesada.

Los procesos por los cuales se somete a una base de datos para tener información confiable, limpia, de calidad y organizada son diversas, algunas empresas o negocios aun lo hacen de manera manual, por la cantidad de información o desconocimiento del tema, pero en su mayoría adoptan gestores de bases de datos y métodos que ayudan con esta tarea, teniendo como finalidad extraer la información necesaria y poder realizar una correcta toma de decisiones sobre nuevos datos [5], Big Data, nos brinda herramientas que nos ayuda a predecir en cualquier campo facilitándonos la toma de decisiones preventivas [6], es muy importante tener esta información de manera clara y organizada, marcando una necesidad en adoptar herramientas que permitan realizarlo como procesos de identificación y categorización de conceptos claves, técnicas de búsqueda etc. [7], en la cual, los elementos de base como palabras y oraciones se agrupan bajo elementos denominado categoría superior, donde se establecen niveles por los que se ordenan las partes [8], lo que da un punto de vista organizacional y claro con la información.

Herramientas que ayudan a tener mejor comunicación, en el que dos o más máquinas colaboran para obtener un buen resultado, son los sistemas distribuidos [1], que trabaja por separado o en módulos, como en este proyecto, cada equipo desarrollará su proyecto con las mismas herramientas enfocadas cada uno en su funcionalidad.

En el proceso del desarrollo de esta herramienta, se ha desarrollado un plan de acción con las siguientes fases: Primera, se realiza los planos del sistema para representar la arquitectura, estructura y comportamiento del sistema mediante diagramas UML. Segunda, construir scripts para realizar una clasificación de columnas en dimensiones y hechos por medio de la herramienta con la ayuda del usuario. Tercera, implementar interfaces que permitan la conexión a la base de datos, visualizar el contenido y extraer la información para categorizarla.

La importancia de este proyecto de investigación es brindar información clara y precisa de diferentes aplicaciones utilizadas, como sistemas distribuidos, base de datos, Android, la novedad se enfoca en trabajar de manera modular, multiplataforma, interactúa con aplicaciones móviles, haciendo que cada una de sus fases sea independiente de la otra, se desarrolla la aplicación en Android en Kotlin, en donde se aporta al análisis de la información tratándola según las necesidades como la importancia que tiene la organización en clasificación de la información para su correcta manipulación en donde se aporta en la línea de investigación de sistemas.

La base de datos que se maneja es previamente procesada y sometida a herramientas que ayudan a su organización y calidad respecto a datos, en este caso aplicamos el proceso de categorización de información y para ello se maneja una metodología con la cual se va a realizar el desarrollo de la aplicación distribuida para el apoyo a la categorización mediante un análisis exploratorio, se ejecutan diferentes fases y actividades para culminar con éxito la herramienta.

Con los procesos realizados, manteniendo la información con la mayor integridad, se observa que los datos han sido correctamente procesados en la aplicación y ejecutados según lo planeado mediante la interfaz interactuando con el usuario, generando correctamente el proceso de categorización, mediante la clasificación del usuario por medio de la herramienta y opciones suministradas, evidenciando a través de la aplicación que la base discretizada pasada por el grupo anterior carga correctamente los datos para realizar el proceso de categorización, en donde la ejecución de los scripts a través del gestor de base de datos Postgres se ejecutan correctamente y finalizado el proceso de selección por parte del usuario, se obtiene la tabla con la información de la categorización listos para ser entregados al grupo siguiente.

Los procesos descritos anteriormente se obtienen con una serie de procesos los cuales son desarrollados mediante diagramas UML en los que se explica paso por paso el funcionamiento del sistema en la figura 2 miramos el diagrama arquitectónico sobre el cual observamos el trabajo modular por cada grupo procesos a los cuales la información pasará por cada uno de ellos en el orden respectivo de forma independiente y en nuestro caso recibiremos la base del módulo de discretización y nosotros pasarlo al grupo de topología de cubos para que ellos continúen con el proceso.

En la figura 3, se visualiza el diagrama de capas de la herramienta de categorización, mostrando específicamente su estructura, en las figuras 4, 5 y 6 se observan los diagramas de comportamiento de actividades, en los que se muestra el proceso que se realiza al clasificar las columnas en dimensiones o hechos la seleccionan final definiendo su clasificación, es así que el usuario experto interactúa con la aplicación para seleccionar si los datos son hechos o pueden cambiar a dimensión, dando una clasificación definitiva a las columnas.

En las figuras 7 y 8 obtenemos en diagrama de paquetes y clases en las cuales evidenciamos el proceso de la aplicación, de las figuras 9 al 13 se observan las consultas para extraer información desde la base de datos y posteriormente de las figuras 14 al 16 las vistas que tendrá el usuario para categorizar los datos en hecho o dimensión como la creación de la tabla categorizada.

En la fase 3 tenemos las figuras 17 a 21 que nos indican las conexiones del Api para su funcionalidad hacia la base de datos el servidor y su configuración en general y como mediante estos procesos en las figuras 22, 23 y 24 se obtiene la aplicación en su diseño final y funcionalidad.

II. DESCRIPCIÓN DEL PROBLEMA

El manejo de datos en la actualidad no solo representa una gran cantidad, variedad y volumen de información, sino que se ha convertido en el tema de “moda” encontrándose diariamente en los periódicos y revistas; igualmente, los sectores económicos, las empresas y consultoras más importantes intentan mostrar sus posibles aplicaciones y generan informes frecuentes al respecto [4], por ello los directivos procuran tener herramientas adecuadas para la explotación, análisis, una correcta organización y categorización de la información que permitan obtener el conocimiento necesario en el proceso de toma de decisiones estratégicas [9].

En un estudio realizado en marzo del 2007 por SAS (empresa líder en software y servicios de negocios, quien es el vendedor más grande de soluciones de inteligencia de negocios en el mundo), en donde 175 profesionales de negocios dieron sus puntos de vista sobre la inteligencia de negocios de sus organizaciones, representando una diversidad de tamaños y tipos de industrias de todo el mundo. Se encontró que el 80 por ciento de las organizaciones que manejan adecuadamente sus datos y usan informes analíticos avanzados exceden el rendimiento obtenido en años anteriores resulta incluso aún más convincente que las diferencias entre las prácticas de las organizaciones más o menos exitosas [9]. Es el gran impacto que en la sociedad está produciendo el desarrollo de las TICs y aquellos que se van a producir como consecuencia de su amplia difusión en la sociedad [9].

La cantidad de información depende mucho del rendimiento del sistema, a mayor volumen de datos, más pesado y tardío es el proceso interno que maneja y somete al sistema a esforzarse mucho, provocando volcamiento de información, ya que internamente es demasiado para un sistema simple, debido a que en su mayoría las empresas u organizaciones no tienen contemplado la gran cantidad de información que con el tiempo va acumulándose de manera abismal y es entonces cuando no pueden organizarla ni controlarla de manera adecuada, requiriendo y optando por adoptar herramientas que faciliten estos procesos y controlarlos de manera fácil y rápida, pero aun así, hay muchos quienes son reacios a cambios y los datos no contemplan una calidad adecuada ni una limpieza que permita manejarlos ya que existirían muchos datos sucios dentro la información.

Una base de datos requiere de información clara, limpia, organizada y de calidad para que sea sometida a diversos procesos de extracción de informes y datos, la base que se maneja en cualquier empresa o negocio por más pequeño que sea requiere someterse a estos procesos como la limpieza e información ordenada y clara ya que existe diversa información que ya sea por error de digitación o cualquier otra razón está mal guardada provocando redundancia de datos, someterla a procesos importantes como categorizarla para tener claro el contenido más destacado con el que se pueda organizar el documento, con una base correctamente procesada con datos claros se obtiene una correcta toma de decisiones ya que se disminuye el error de fallar por mal manejo de guardado de información, la toma de decisiones permite automatizar procesos que si se hicieran con la intervención del ser humano podrían tener un alto costo en tiempo, haciendo que se abran las puertas para su implementación con sistemas incluyentes [5].

La importancia del desarrollo de los Sistemas Distribuidos se debe a la necesidad de los usuarios que requieren de aplicaciones más sofisticadas, así como también de las funciones administrativas y económicas, la opinión de varios personajes que los han investigado a través de la historia han definido un DSS como un sistema interactivo basado en computador que da soporte a los tomadores de decisiones en vez de reemplazarlos, utiliza datos y modelos para

resolver problemas con diferentes grados de estructura: no estructurados, semiestructurados, tareas estructuradas y no estructuradas y se enfoca en la efectividad más que en la eficiencia de los procesos de decisión [4].

Los sistemas distribuidos ayudan en la toma de decisiones, facilitando el rápido proceso para obtener resultados y así facilitar al usuario, a escoger una opción apropiada y justificada mediante datos ya que este sistema de acreditación contrasta con los ya existentes sistemas “centralizados”, por ser un sistema “distribuido”, inspirado en el sistema mediante el cual la comunidad científica internacional válida los artículos de investigación [1].

Dentro de la toma de decisiones existen múltiples subprocesos, uno de ellos es la categorización, objeto de la presente investigación, la cual se define como el manejo de categorías con los datos que previamente se han procesado, esto responde a un enfoque estático, tomando la información desde un punto de vista cualitativo, organizado. Esto es, a considerar la sensibilidad desde la posible repercusión material de tales datos en la esfera de derechos de las personas. En este sentido, Big Data aporta unas posibilidades de modelado de la información capaces de convertir en sensibles datos que no responden objetivamente a esta categorización de acuerdo con la Ley. [6].

El tema de los sistemas distribuidos y su propagación dentro de las nuevas dinámicas sociales relacionadas a la sociedad del conocimiento a nivel internacional han tenido una gran relevancia en medida que los procesos internacionales de aseguramiento de la calidad no han encontrado necesarios modelos establecidos. Por ejemplo, la organización basada en Bruselas mediante sistemas distribuidos válida artículos de investigación [1], en Perú se realizó un proyecto enfocado en la recuperación de información de múltiples bibliotecas basándose en una arquitectura distribuida y asegurar un óptimo nivel de rendimiento en la recuperación de información, obteniendo buenos resultados con el método empleado [10].

En España también se han realizado proyectos en los sistemas de fabricación holónicos tanto en la programación como en su control en donde la arquitectura propuesta, distribuida según órdenes y máquinas, así como la tecnología utilizada para implementar el sistema, lo dotan de las características deseadas de flexibilidad, reconfigurabilidad, adaptabilidad, rapidez de respuesta, robustez y mantenibilidad obteniendo muy buen rendimiento [11] con la aplicación de sistemas distribuidos en diferentes procesos con un fin: mejorar y aportar en el conocimiento dando un énfasis a nivel internacional para futuros proyectos.

En Colombia existen diversos procesos que se han llevado a cabo en diferentes proyectos en aprovechamiento de los sistemas distribuidos en el apoyo a la toma de decisiones es por eso que se han encontrado o identificado obstáculos que en este caso las PYMES (según la Ley para el Fomento de la Micro, Pequeña y Mediana Empresa, Ley 590) tienen para su desarrollo en Colombia, como las dificultades en la identificación y acceso a la tecnología adecuada, la formalización y absorción de nuevas tecnologías, por eso el uso y apropiación de las organizaciones a herramientas tecnológicas dentro de las que se incluyen el software y los sistemas de información, buscando así generar ventajas competitivas [2].

Estableciendo los parámetros de clasificación dentro de los que se destacan: número de trabajadores y activos totales, categorización jurídica que hace referencia a unidades de explotación económica, realizadas por persona natural o jurídica, en actividades empresariales, agropecuarias, industriales, comerciales o de servicios, rural o urbana [2].

En el país también se han realizado proyectos en busca de los desequilibrios regionales en Colombia, el cual mediante un análisis de información mediante sistemas distribuidos evidencia factores diferenciales entre las diversas regiones y sobre el crecimiento económico de los diferentes departamentos de Colombia dichas diferencias son explicadas por los efectos estructurales, regionales y de reasignación económica de cada región. Estos efectos han permitido que algunas regiones experimenten unos mejores niveles de crecimiento a lo largo de este nuevo siglo, y así mismo, han generado un efecto negativo sobre el crecimiento de las regiones más pobres de Colombia [3].

En Nariño se ha desarrollado un proyecto con el cual genera la importancia de la toma de decisiones mediante un procesamiento de información que es denominado: Desequilibrios regionales en Colombia en el que hace un estudio en todas las regiones de Colombia sobre la desigualdad en el crecimiento económico que presentan las regiones mediante un estudio, análisis utilizando y procesando la información, mediante sistemas distribuidos se han logrado obtener resultados como Bogotá en donde se han alejado del ingreso promedio nacional, mientras que regiones como Valle y Antioquia han mostrado una tendencia a aproximarse al promedio nacional, para los demás departamentos se logra identificar que se encuentran por debajo del promedio formando un club de regiones rezagadas con los casos sobresalientes de Chocó, Nariño y algunos departamentos del oriente y Amazonía [3], evidenciando información de calidad validado por un proceso claro.

De igual manera en Nariño se realizó un estudio sobre el tamaño de la gota de aspersión, potencial de deriva y riesgo para los organismos no blanco del glifosato aplicado por vía aérea, para el control de la coca en Colombia específicamente en la región de Nariño y Putumayo en donde se realizó un estudio mediante sistemas distribuidos en las plantas las cuales se las utilizaron para estas pruebas como subrogados para otras plantas en el medio ambiente, un proceso normal para evaluar riesgos en donde se evidencio que el buen manejo en la toma de decisiones efectiva en la aplicación aérea de plaguicidas puede apoyarse en el uso de herramientas adecuadas e información de modelaje y evaluación de riesgo [12].

El manejo de grandes cantidades de información y la necesidad en la toma de decisiones de gestión eficiente en una institución es innegable su importancia, la gestión oportuna y adecuada de los indicadores de internación y la toma de acciones sobre los mismos, es necesario contar con información de calidad [13], organizada que tenga un orden y sea de facil entendimiento que ayude a mitigar y disminuir riesgos con equipos informáticos al presentar información como reportes [14].

Si no se genera un procesamiento confiable con la información los resultados que se brinden se darán como no confiables pues cuestiones como atributos faltantes o con valores no plausibles impactan de manera exponencial en las conclusiones a obtener [15] si no se poseen estas herramientas es clara la incapacidad de las empresas y sus directivos para alcanzar el cien por cien del potencial al igual que para acceder, analizar e interpretar los datos para facilitar la toma de decisiones en las organizaciones [4].

Por todo lo anterior es necesario realizar un análisis detallado sobre los datos que permita apoyar en la toma de decisiones y para ello se propone el desarrollo de una aplicación que lo permita mediante un sistema distribuido específicamente en la categorización de datos que permita tener una organizacion y orden en la informacion manejada todo esto por medio de un proceso de análisis con el cual se pueda explorar la información previamente procesada y discretizada consiguiendo así obtener datos de calidad logrando un análisis de similitud textual sobre la información suministrada para conseguir así finalmente generar como resultado de estos diversos procesos una tabla de resumen categorizada, la cual puede continuar con diferentes transformaciones al manejo de datos.

II-A. Formulación del problema

¿Cómo desarrollar una aplicación distribuida para el apoyo a la Categorización de datos mediante análisis exploratorio?

III. JUSTIFICACIÓN

Las telecomunicaciones y la transmisión de datos de un lado a otro era una actividad que tardaba días por la tecnología que se utilizaba antes, hoy gracias a su evolución solo se tarda segundos, es por esto que se ha dado paso a un nuevo campo con el fin de identificar las diferentes metodologías de análisis de procesamiento de datos. La importancia de este trabajo de investigación es poder brindar una información clara y precisa de las diferentes aplicaciones utilizadas.

A través de este proyecto se pretende realizar un aporte a la investigación, aplicando los conocimientos adquiridos en el CIP (Curso de Investigación Pre gradual) como, sistemas distribuidos, programación, bases de datos, android, entre otros.

La herramienta realizada en este modulo como lo referencia en la figura 2 depende de los resultados de las aplicaciones anteriores y a su vez se dan los resultados para la continuación de los procesos siguientes ya que al manejar grandes cantidades de información, este proceso nos permite en este punto obtener información previamente procesada es decir limpia, organizada y categorizada con la cual se procedería a realizar procesos KDD de extracción de la información útiles y entendibles para el usuario.

Refiriéndonos con Sistema distribuido podemos definir que es aquel en el que dos o más máquinas colaboran para la obtención de un resultado y están basados en las características de transparencia, eficiencia, flexibilidad escalabilidad y fiabilidad. El objetivo principal de los sistemas distribuidos es el mejor desempeño, mayor fiabilidad y disponibilidad, comparación de recursos e información, además del mejoramiento de la comunicación [1].

Los Sistemas Distribuidos tienen una mayor confiabilidad, su beneficio es que, al ser distribuido, si un módulo falla, no afecta a los demás, el sistema sobrevive como un todo. Si los datos están distribuidos de una manera eficiente, las peticiones de los usuarios van a poder ser satisfechas directamente desde los datos locales, por lo que se proporciona una respuesta más rápida. Por otro lado, en sistemas centralizados, todas las solicitudes tienen que pasar a través del ordenador central, lo cual incrementa el tiempo de respuesta [1].

Está más que claro que hoy en día la informática se ha impuesto en la sociedad de forma considerable. Cada vez son más los dispositivos y los avances tecnológicos que requieren de esta ciencia y a la misma vez esta se nutre de todas las novedades que asiduamente se presentan en el mercado. Sin embargo, aunque todo ello se ha asentado, en la actualidad se desconoce de muchos sistemas y técnicas que hacen posible lo que se conoce como informática en su sentido más amplio. Especialmente si se piensa en dedicarse a ello se debe colocar manos a la obra para aumentar estos conocimientos.

La novedad del proyecto se enfoca en que este sistema distribuido trabaja de forma modular, es multiplataforma, interactúa con aplicaciones móviles, haciendo que cada una de sus fases sea independiente de la otra, que quiere decir esto, si alguno de los elementos falla, los otros podrán seguir realizando la función correctamente. Por este motivo los sistemas distribuidos suelen otorgar bastante confianza a la hora de trabajar con ellos, ya que es muy raro que falle el sistema por completo.

En los laboratorios de la institución se dispone del material y los recursos humanos específicos como los docentes ya que cada uno de ellos tiene la experiencia en el manejo de cada uno de los temas del módulo que les corresponde tanto para la investigación en esta área como para dar apoyo a los estudiantes a cargo del proyecto, haciendo que el este sea entendible y llegue a su finalización, el tiempo por parte de ellos y el nuestro ha sido acorde permitiendo tener una mayor interacción en el desarrollo.

Se cuenta con los equipos tecnológicos necesarios, con características que exigen las nuevas plataformas, permitiendo que el proyecto se ejecute sin presentar problemas al momento de procesar los datos. Esta aplicación se desarrolla en una plataforma multilenguaje, trabaja con pgAdmin que es un gestor de base de datos presentando características que lo hacen óptimo para cualquier sistema automatizado como: rapidez, multiplataforma, multiusuario y permite la encriptación de datos lo que asegura que estos puedan viajar por la red de forma segura y no ser interceptados. La aplicación móvil se desarrolla en Android, con un lenguaje de programación nuevo llamado Kotlin haciendo que sea una novedad para nuestro proyecto.

En cuanto a referencias técnicas sobre la aplicación móvil, puede ser ejecutada en cualquier dispositivo que contenga un sistema operativo Android, sin importar sus recursos.

Por medio de este proyecto se pretende aportar al análisis de la información que se maneja con mucha frecuencia hoy en día, sacando provecho de ella, manejándola adecuadamente según las necesidades, es por ello que debe pasar por diversos procesos mediante un pre procesamiento el cual a pesar de manipular los datos en calidad y limpieza no es suficiente, es necesario la importancia que tiene la organización o clasificación superestructural para la lectura y representación documental, es decir clasificar la información para una correcta manipulación.

Un trabajo que conlleva bastante esfuerzo realizarlo si se lo hace manualmente para ello se aporta con una herramienta que funciona independientemente con sistemas distribuidos alrededor de las diferentes etapas que maneja las fases de las manipular información o bases de datos, al categorizar los datos estamos aportando en la línea de investigación de sistemas inteligentes el cual se enriquecería al manejar una base de información previamente discretizada y de manera modular hacer una categorización interactuando con el cliente o usuario final directamente.

IV. OBJETIVOS

IV-A. *Objetivo General*

Desarrollar una aplicación distribuida para el apoyo a la categorización mediante análisis exploratorio.

IV-B. *Objetivos Específicos*

1. Realizar los planos del sistema para representar la arquitectura, estructura y comportamiento del sistema mediante diagramas UML.

2. Construir script para clasificar a las columnas en dimensiones y hechos probables mediante funcionalidades medianamente acopladas en el motor de base de datos.
3. Implementar las interfaces necesarias para replicar la información en un motor de base de datos y categorizar los hechos probables en hechos o dimensiones a criterio de un usuario experto.

V. MARCO TEÓRICO

V-A. *Tratamiento de datos*

El tratamiento de los datos es un proceso muy laborioso y es muy importante definir datos de entrada para obtener resultados satisfactorios. Por otra parte, el propósito del preprocesamiento de datos es principalmente corregir las inconsistencias de los datos que serán la base de análisis en procesos de minería de datos. En el caso de las fuentes de datos estructuradas, el propósito no es distinto y pueden ser aplicadas diversas técnicas estadísticas y de aprendizaje computacional [16]. Con el preprocesamiento de datos se pretende que los datos que van a ser utilizados en tareas de análisis o descubrimiento de conocimiento conserven su coherencia.

Las herramientas de inteligencia de negocio se han centrado en el tratamiento de los datos con el fin de brindar una información óptima, que permita a los analizadores generar una buena toma de decisiones con el fin de mejorar los procesos y, por ende, los niveles competitivos de la organización frente al mercado [17].

V-B. *Knowledge Discovery in Databases (KDD)*

La exploración y calidad de datos permite extraer información relevante, con el fin de mejorar el rendimiento de un sistema que permita su inspección de la información formada por contenido, generación y exploración de datos en las empresas, especialmente en las organizaciones actuales, donde se necesita agilizar procesos entorno a la información que manejan, para que sea reutilizable y automatizada cada día es más fuerte.

Se entiende que la efectividad del Data Mining (DM), depende en gran medida de una buena preparación de los datos para lograr el mejoramiento en la utilización de los mismos. Este proceso de preparación suele incluir la construcción y utilización de complejos Almacenes de Datos o Data Warehouses (DW) [18].

De igual forma se lleva a cabo una etapa de limpieza, la cual consiste en la búsqueda de los datos anómalos o también llamados outliers, son datos que se encuentran a una distancia anormal de los demás datos, un outlier es una observación que parece inconsistente con el resto de los valores de la muestra, siempre teniendo en cuenta el modelo probabilístico que debe seguir la muestra. Existen varias técnicas para la detección de datos anómalos como la desviación estándar y la caja de bigotes, entre otros [19].

Así mismo, la discretización corresponde a una operación que permite recorrer variables con valores continuos y pasarlos a valores discretos. Este manejo es utilizado ampliamente para comprimir datos y así facilitar a las máquinas digitales en lo que corresponde a términos de espacio y tiempo. La discretización es frecuentemente aplicada a las variables que serán usadas en clasificación o análisis de asociación. En general, la discretización en cuanto a resultados, se considera que para obtener mejores resultados en los algoritmos de minería de datos, en el momento del análisis de la información, será más viable la ejecución tanto en tiempo, espacio y reducción de visión.

El proceso de discretización puede ser dividido en dos tareas principales: La primera es encontrar el número de grupos discretos en que se mapean los valores continuos y la segunda es definir el rango o límites de cada intervalo

en el dominio continuo o bien llamado número de clases [20].

De igual manera, el manejo de datos no solamente requiere de información limpia es necesario que se encuentre organizada, agrupada y categorizada mediante un análisis, el cual es previamente realizado por medio de scripts en el sistema gestor de base de datos (SGBD) PostgreSQL; se crea una tabla en la que se inserta la información, como el nombre de las columnas y su clasificación, también se crean datos adicionales como el orden, nombre de la dimensión a la que pertenece la columna, todo bajo la revisión del usuario, quien va a complementar cada uno de los campos que contiene la categorización y por último poder observar la tabla final, es decir, el resultado obtenido, facilitando la toma de decisiones, permitiendo automatizar procesos que se realizan con la intervención del cliente [5].

Ademas, la topología en estrella, es creada a partir de un Data Warehouse, como subconjuntos de datos, los cuales son llamados Data Marts; este tipo de conjunto consiste en que, exista una única tabla central, la cual es llamada: tabla de hechos, que contiene todas las medidas, también una tabla adicional por cada una de las perspectivas que se desea analizar dicha información, es decir, por cada una de las dimensiones. Todo esto con el fin de responder las necesidades específicas según el tema, mejorando el rendimiento, lo cual permite obtener resultados más tempranos al aplicar buenas prácticas. El tiempo de implementación es rápido y la utilización de cada Data Mart permite que la solución no presente un alto costo [21].

Por consiguiente, el valor de una herramienta de procesamiento analítico en línea (OLAP) se deriva de la capacidad de analizar eficientemente los datos desde múltiples puntos de vista, por lo que las herramientas OLAP están diseñadas para calcular previamente las agregaciones y almacenarlas directamente en las bases de datos. Aunque este diseño permite un acceso rápido, significa que debe haber una cantidad significativa de preparación de los datos para la presentación OLAP, mostrando al usuario una visión multidimensional de los datos (esquema multidimensional) para cada actividad que es objeto de análisis [22].

Finalmente, la visualización multidimensional de cubos OLAP permite acceder a un conjunto de datos compuesto de medidas detalladas por dimensiones, este proceso presenta la facilidad para acceder a la información desde diversas perspectivas, ofreciendo al analista la posibilidad de crear nuevo conocimiento de la información que reposa en las bodegas de datos y así utilizarla como apoyo en los procesos de la toma de decisiones. Las bases de datos multidimensionales están desarrolladas en una estructura que agiliza el acceso a las diferentes consultas que se realizan en la base de datos y de este modo optimizar el tiempo que se necesita para la generación de reportes y visualización de datos [23].

V-C. Categorización de Datos

La Categorización de datos se refiere a manejar la información de manera organizada con el fin de clasificar y ordenarla por medio de un usuario experto y conocedor de la base de datos para que quien tenga acceso a esta información pueda entenderla y extraer datos con mayor facilidad de la que lo haría si no lo estará.

Al tener información clasificada y organizada facilita la obtención de información y con ello el análisis que se requiera según la necesidad para la cual se requiera y con ello generar y soportar la toma de decisiones necesarias para generar a largo plazo ventajas competitivas en organizaciones y empresas.

Es por ello que se ha implementado este punto tan importante para este proyecto con el fin de optimizar procesos en el mejoramiento y la calidad de datos al momento de su clasificación y la organización de la información con el fin de generar mayor comprensión es por ello que en empresas que requieren de estos sistemas los directivos procuran tener herramientas adecuadas para la explotación, análisis, una correcta organización y categorización de la información que permitan obtener el conocimiento necesario en el proceso de toma de decisiones estratégicas [9].



VI. MARCO LEGAL

Las distintas leyes mencionadas a continuación, permiten la protección de proyectos de investigación:

Según el artículo 1 de la ley 23 de 1982 (28 de enero de 1982). Los autores de obras literarias, científicas y artísticas gozarán de protección para sus obras en la forma prescrita por la presente ley y, en cuanto fuere compatible con ella, por el derecho común. También protege esta ley a los intérpretes o ejecutantes, a los productores de fonogramas y a los organismos de radiodifusión, en sus derechos conexos a los del autor. por la presente ley se da seguridad y protección al trabajo realizado por los autores de este proyecto.

Otro estatuto que aporta a este proyecto es la LEY 1341 DE 2009 (Julio 30), por la cual se definen Principios y conceptos sobre la sociedad de la información y la organización de las Tecnologías de la Información y las Comunicaciones -TIC-, se crea la Agencia Nacional del Espectro y se dictan otras disposiciones.

Acorde con el Decreto número 1317 de 2013, Mediante la Ley 1581 de 2012 se expidió el Régimen General de Protección de Datos Personales, el cual, de conformidad con su artículo 1, tiene por objeto de desarrollar el derecho constitucional que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos, y los demás derechos, libertades y garantías constitucionales a que se refiere el artículo 15 de la Constitución Política; así como el derecho a la información consagrado en el artículo 20 de la misma. Que la Ley 1581 de 2012 constituye el marco general de la protección de los datos personales en Colombia. Que mediante sentencia C-748 del 6 de octubre de 2011 la Corte Constitucional declaró exequible el Proyecto de Ley Estatutaria No. 184 de 2010 Senado, 046 de 2010 Cámara. Que con el fin de facilitar la implementación y cumplimiento de la Ley 1581 de 2012 se deben reglamentar aspectos relacionados con la autorización del Titular de información para el Tratamiento de sus datos personales, las políticas de Tratamiento de los Responsables y Encargados, el ejercicio de los derechos de los Titulares de información, las transferencias de datos personales y la responsabilidad demostrada frente al Tratamiento de datos personales, este último tema referido a la rendición de cuentas.

Para este proyecto se empleó herramientas de código abierto por medio de la fundación para el software libre que establece la Licencia Pública General de GNU: Es una licencia libre de copyleft para software y otros tipos de obras.

Las licencias para la mayoría del software y otros trabajos prácticos están diseñados para quitarle la libertad de compartir y cambiar los trabajos. Por el contrario, la Licencia Pública General de GNU está destinada a garantizar su libertad para compartir y cambiar todas las versiones de un programa, para asegurarse de que siga siendo software libre para todos sus usuarios. Nosotros, la Free Software Foundation, utilizamos la Licencia Pública General de GNU para la mayoría de nuestro software; se aplica también a cualquier otro trabajo publicado de esta manera por sus autores. También puede aplicarlo a sus programas.

Cuando hablamos de software libre, nos referimos a la libertad, no al precio. Nuestras licencias públicas generales están diseñadas para garantizar que tenga la libertad de distribuir copias de software gratuito (y cobrar por ellas si lo desea), que reciba el código fuente o que pueda obtenerlo si lo desea, que puede cambiar el software o use partes de él en nuevos programas gratuitos, y que sepa que puede hacer estas cosas.

Para proteger sus derechos, debemos evitar que otros le nieguen estos derechos o le pidan que renuncie a ellos. Por lo tanto, tiene ciertas responsabilidades si distribuye copias del software o si lo modifica: responsabilidades para respetar la libertad de los demás.

Por ejemplo, si distribuye copias de dicho programa, ya sea gratis o por una tarifa, debe transmitir a los destinatarios las mismas libertades que recibió. Debe asegurarse de que ellos también reciban o puedan obtener el código fuente. Y debe mostrarles estos términos para que conozcan sus derechos.

Los desarrolladores que usan la GNU GPL protegen sus derechos con dos pasos: (1) hacer valer los derechos de autor sobre el software y (2) ofrecerle esta Licencia que le otorga permiso legal para copiarlo, distribuirlo y /

o modificarlo.

Para la protección de los desarrolladores y autores, la GPL explica claramente que no hay garantía para este software gratuito. Por el bien de los usuarios y de los autores, la GPL requiere que las versiones modificadas se marquen como modificadas, de modo que sus problemas no se atribuyan erróneamente a los autores de versiones anteriores.

Algunos dispositivos están diseñados para negar a los usuarios el acceso para instalar o ejecutar versiones modificadas del software dentro de ellos, aunque el fabricante puede hacerlo. Esto es fundamentalmente incompatible con el objetivo de proteger la libertad de los usuarios para cambiar el software. El patrón sistemático de tal abuso ocurre en el área de productos para uso individual, que es precisamente donde es más inaceptable. Por lo tanto, hemos diseñado esta versión de la GPL para prohibir la práctica de esos productos. Si tales problemas surgen sustancialmente en otros dominios, estamos listos para extender esta disposición a esos dominios en futuras versiones de la GPL, según sea necesario para proteger la libertad de los usuarios.

Por último, cada programa está constantemente amenazado por las patentes de software. Los estados no deberían permitir que las patentes restrinjan el desarrollo y el uso de software en computadoras de uso general, pero en las que lo hacen, deseamos evitar el peligro especial de que las patentes aplicadas a un programa gratuito puedan convertirlo en propiedad efectiva. Para evitar esto, la GPL asegura que las patentes no se pueden utilizar para hacer que el programa no sea libre.

A continuación se nombran los términos y condiciones precisos para la copia, distribución y modificación del software libre en el presente trabajo:

1. Código fuente.
2. Permisos básicos.
3. Protección de los derechos legales de los usuarios de la ley contra la elusión.
4. Transporte de copias literales.
5. Transporte de versiones de origen modificadas.
6. Transporte de formularios no fuente.
7. Términos adicionales.
8. Terminación.
9. No se requiere aceptación para tener copias.
10. Licencias automáticas de destinatarios intermedios.
11. Patentes.
12. No rendirse a la libertad de los demás.
13. Utilícelo con la Licencia pública general GNU Affero.
14. Versiones revisadas de esta licencia.
15. Descargo de responsabilidad de la garantía.
16. Limitación de responsabilidad.



VII. METODOLOGÍA

La metodología con la cual se va a realizar el desarrollo de la aplicación distribuida para el apoyo a la categorización mediante un análisis exploratorio, se efectuará en varias fases y con ello actividades, con el fin de cumplir el desarrollo total de la aplicación.

Fase 1. Definición de requerimientos.

Actividad 1.

Se realizan los diagramas UML en los que se mostrará de manera clara su funcionamiento: el diagrama arquitectónico, donde se observa de manera general el proyecto y la aplicación, los diagramas de actividades, en los que se detalla las actividades principales del funcionamiento de la aplicación y el diagrama estructural, que detalla los paquetes y clases que se manejan.

Actividad 2.

Con la información previamente obtenida, se realizarán una serie de consultas sobre la base de datos, con el fin de identificar si los datos son hechos (datos tipo numéricos) o dimensiones (datos tipo texto), según se muestre la información obtenida en la base discretizada.

Actividad 3.

Una vez hechas las consultas, se obtiene la clasificación de columnas en hechos y dimensiones.

Fase 2. Clasificación de hechos probables y dimensiones.

Actividad 1 y 2.

Se toma únicamente los hechos, los cuales se enviarán a una interfaz, donde el usuario con conocimientos previos del tema, tendrá que seleccionar si pertenecen o no a la dimensión de hechos, con esta elección estarán determinándolo por criterio propio, dando paso a una selección de espacio de las columnas con esto quien esté manejando la herramienta deberá ubicar la columna en una dimensión o espacio dado previamente por la aplicación.

Actividad 3.

Se genera una tabla de resultado que visualiza las columnas con los datos ya categorizados mediante conocimientos previos del usuario.

Fase 3. Visualización de categorías

Actividad 1.

Se crea un API (interfaz de programación de aplicaciones) con el fin de conectar la base de datos pgadmin a la herramienta Android y se utiliza heroku como servidor.

Actividad 2.

Se desarrolla las vistas de la Aplicación mediante la herramienta Android, con su lenguaje de programación kotlin.

Actividad 3.

Se realiza una prueba general con la cual se ejecuta todo el funcionamiento de la aplicación y se procede a corregir errores (si estos se generan) hasta que la herramienta funcione previos esquemas y diseños.

VII-A. *Plan de Acción*

Fases	Actividades	Entregables	Recursos
Fase 1: Definición de requerimientos	Diseño de diagramas UML	Diagramas	Tecnológicos, Tiempo personal
	Consultas en base de datos de tablas numéricas	Consultas generadas	Tecnológicos, Tiempo personal
	Clasificación de columnas en hechos y dimensiones	Tabla de resultados generados	Tecnológicos, Tiempo personal
Fase 2: Clasificación de hechos probables y dimensiones	Selección de hechos probables	Selección en aplicación	Tecnológicos, Tiempo personal
	Selección de columnas en categoría	Selección en aplicación	Tecnológicos, Tiempo personal
	Generar datos categorizados	Resultados generados	Tecnológicos, Tiempo personal
Fase 3: Visualización de categorías	Conexiones con Api	Scripts e imágenes de conexión	Tecnológicos, Tiempo personal
	Generación de vistas	Imágenes de la aplicación	Tecnológicos, Tiempo personal
	Validación y pruebas	Imágenes de aplicación funcionando	Tecnológicos, Tiempo personal

Figura 1: Diagrama de fases y actividades para realizar el proyecto.

VIII. **RESULTADOS, ANÁLISIS Y DISCUSIÓN**

VIII-A. *Resultados*

Una vez procesados los datos, se obtienen los resultados manteniendo la información con la mayor integridad posible realizada a lo largo de este proyecto, se observa que los datos han sido correctamente ejecutados, mediante las decisiones del usuario experto con conocimiento del tema de categorización, quien aprovechó la herramienta suministrada, generando correctamente el proceso de categorización con el criterio del cliente, quien al final realiza el análisis con el fin de organizar la información con las opciones brindadas en la aplicación.

A través de la Aplicación, se cargará la base de datos previamente discretizada, para realizar el proceso de categorización, se ejecutarán los scripts a través del gestor de base de datos postgres, luego el usuario con previos conocimientos, utilizará sus criterios personales para elegir la forma en la que se categorizarán los datos permitiendo la interacción con la aplicación, una vez finalizado el proceso, se obtendrá como resultado una tabla con la información y datos específicos categorizados para ser entregado al siguiente grupo.

En este proceso se obtienen una serie de procesos los cuales se han desarrollado en base a diagramas UML los cuales contienen los procesos que se han realizado a lo largo de este proyecto:

IX. FASE 1. DEFINICIÓN DE REQUERIMIENTOS

IX-A. *Diagramas UML*

IX-A1. *Diagrama Arquitectónico* : En la figura 2 podemos observar un diagrama general que evidencia los procesos en los cuales la información será procesada, cada uno de estos pasos es un proyecto individual en el cual la información será manejada cuidadosamente con una aplicación de formato modular, es decir no dependerá la una de la otra, tan solo para el traspaso de la información que se manejará dependiendo del fin de cada uno de los ellos. Como se ha reiterado a lo largo del documento, el módulo que se desarrollará es el de categorización, quien recibe los datos discretizados del anterior módulo, realiza el procedimiento de categorización de datos y luego se trasfiere los resultados obtenidos al siguiente módulo de topología de cubos, para continuar con el proceso.



Figura 2: Diagrama Arquitectónico.

IX-A2. Diagrama de Capas: En la figura 3, se especifica el diagrama de capas que conforma el módulo de categorización, se observa una capa de acceso a datos o base de datos, que se conecta mediante una salida visual de información con Localhost, esta se acopla a la capa lógica de negocios que es un servidor, teniendo una conexión HTTP, al ser una aplicación web podrá ser accesible siempre que se cuente con una conexión a internet desde cualquier parte, donde se enviará una solicitud de entrada y se obtendrá una salida o respuesta, a la vez esta capa se conecta con a la de visualización o llamada FRONTEND, siendo la forma en la que el usuario interactúa con la herramienta.

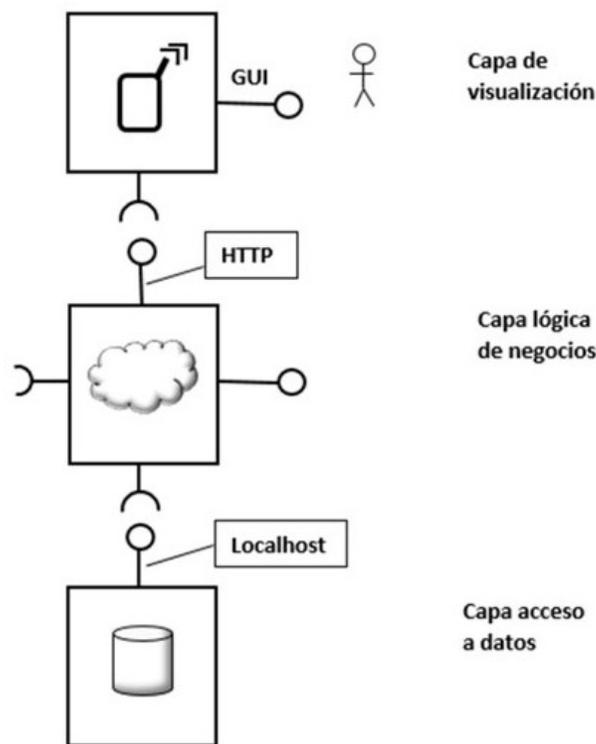


Figura 3: Diagrama de capas.

IX-B. Diagrama de Comportamiento

IX-B1. *Diagrama de Actividades:* En la figura 4, se muestra el primer diagrama de actividades, dando a conocer de manera clara el proceso de selección y clasificación de la información en hechos o dimensiones, se hace una consulta hacia las columnas de la tabla, luego se clasifica, si el tipo de dato es numérico se lo cataloga como hecho probable y si es un tipo de dato texto se lo denomina dimensión, haciendo un proceso de separación.

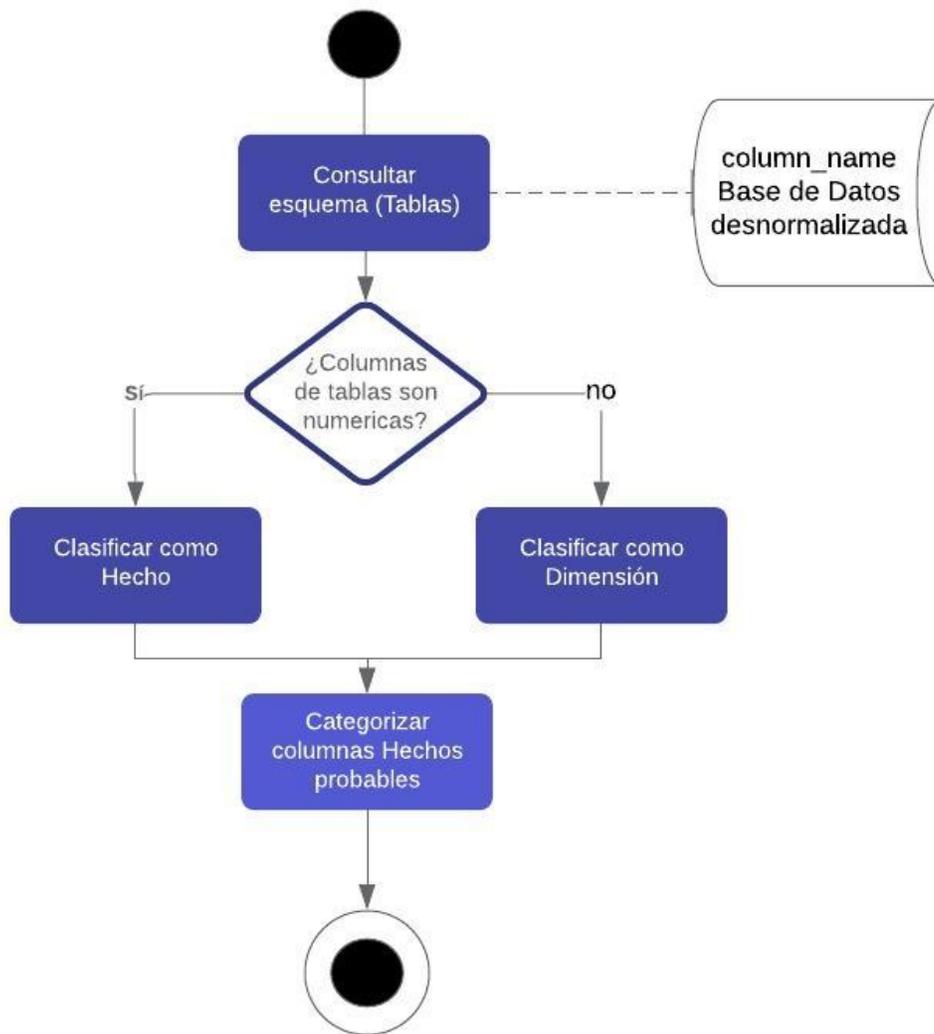


Figura 4: Diagrama de actividades de clasificación de columnas.

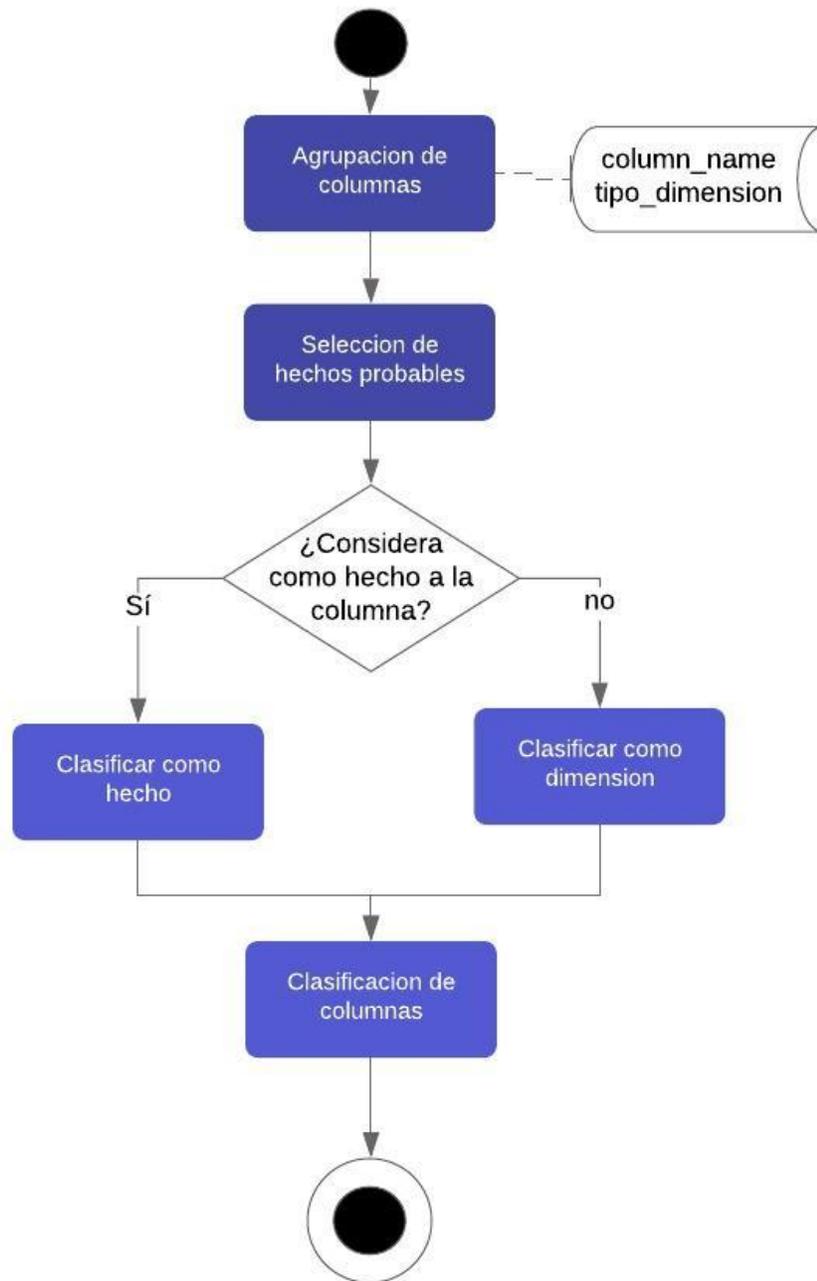


Figura 5: Diagrama de actividades clasificación de hechos y dimensiones.

En la figura 5, se muestra la actividad 2, que describe paso a paso el cómo al finalizar la actividad descrita en la figura 4, obtiene una clasificación de dimensiones y hechos probables por medio de una consulta, se da paso a la actividad 2, que es la interacción de la aplicación con el usuario, donde este debe seleccionar, si los hechos probables previamente adquiridos con anterioridad, se denominan en este grupo o se los cambia, este proceso será de análisis del usuario dependiendo mucho de sus conocimientos sobre el tema.

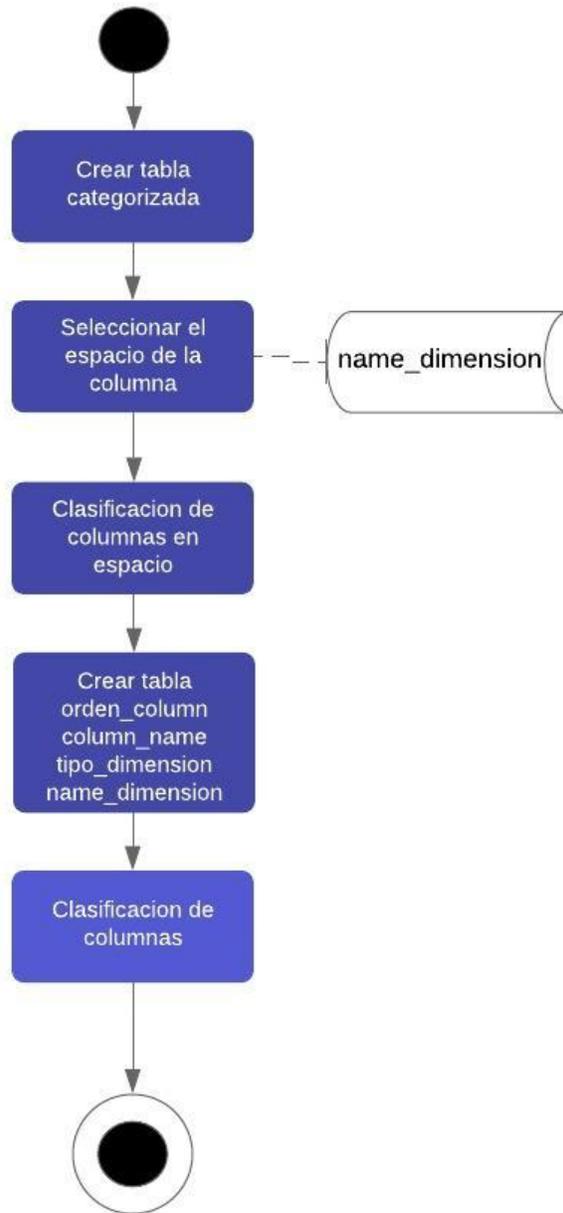


Figura 6: Diagrama de actividades de cambios según criterio del usuario.

En la figura 6, se muestra el proceso que continúa en la actividad 2 de la figura 5, donde el usuario realiza cambios según su criterio en la clasificación, ahora estos serán sujetos al mismo criterio del usuario donde seleccionará su dimensión, para lo cual se confiará en su criterio, análisis y conocimiento previo, esta selección se hará en todas las columnas de la tabla, una vez finalice este proceso, se dará paso como se indica en el diagrama, a generar como producto final una nueva tabla, donde indique el orden, los nombres de las columnas, la clasificación de las columnas (si son hechos o dimensiones) y la clasificación de la dimensión es decir el nombre del espacio que fue designado por el usuario.

IX-C. Diagrama Estructural

IX-C1. Diagrama de clases : En la figura 7, se visualiza el diagrama de clases, describiendo la estructura de la aplicación, mostrando las clases, sus atributos, métodos y las relaciones entre los objetos.

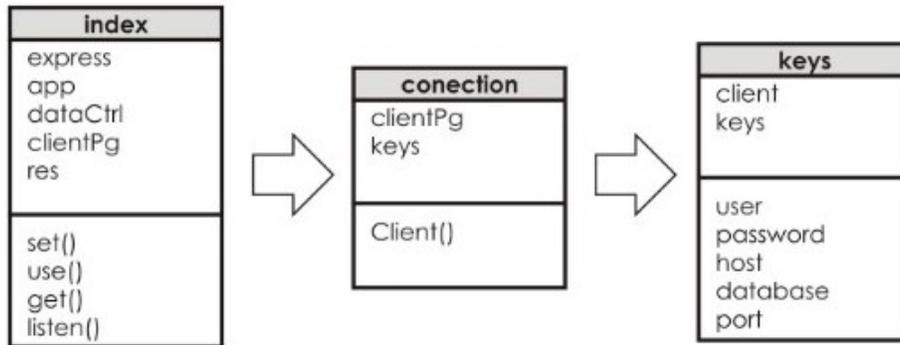


Figura 7: Diagrama de Clases de la aplicación.

IX-C2. Diagrama de paquetes : En la figura 8, se muestra el diagrama de paquetes, proporcionando por medio del gestor de paquetes Node, el acceso hacia los paquetes que fueron necesarios en la creación de la herramienta, este paquete se encuentra en el desarrollo de la Api y nos muestra la estructura de cada elemento.

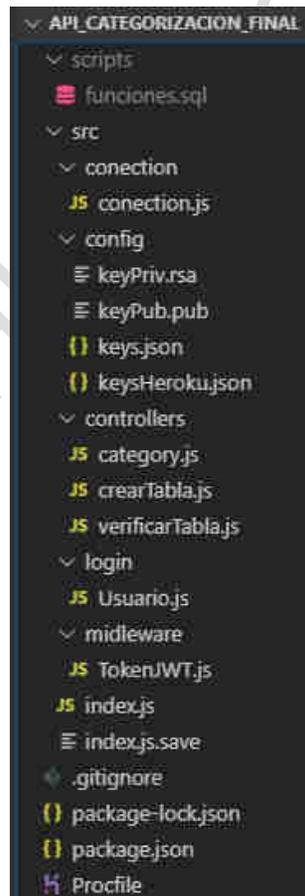


Figura 8: Diagrama de paquetes.

IX-C3. Consultas en base de datos de tablas numéricas: En la figura 9 se muestra el resultado de las consultas realizadas en un Script SQL mostrando la información de la base de datos que ha sido suministrada para su análisis, en la que se especifica la información de la tabla que se pretende visualizar y sobre qué tabla se va a trabajar. Con el fin de obtener y consultar el tipo de dato en categorización que es una parte del análisis que se maneja durante el proceso de ejecución del proyecto.

	table_name character varying	column_name character varying	data_type character varying
1	categories	fresh	numeric
2	categories	milk	numeric
3	categories	grocery	text
4	categories	frozen	text
5	categories	detergents_paper	text
6	categories	delicatessen	text
7	categories	channel	text
8	categories	region	text
9	categories	anio	numeric
10	categories	trimestre	numeric

Figura 9: Resultado de consulta SQL para visualizar la información de la base de datos a través de PgAdmin. (Anexo código fuente A1)

IX-C4. Clasificación de columnas en hechos y dimensión: En las figuras 10 muestra el resultado de la consulta realizada sobre la base de datos postgres separando hechos probables (H) y dimensiones (D). En donde un hecho puede ser de tipo numérico y una dimensión generalmente son de tipo texto.

	column_name character varying	isdimension text
1	fresh	H
2	milk	H
3	grocery	D
4	frozen	D
5	detergents_paper	D
6	delicatessen	D
7	channel	D
8	region	D
9	anio	H
10	trimestre	H
11	mes	H
12	departamento	D
13	municipio	D

Figura 10: Resultados de clasificación en hechos probables y dimensión mediante una función en la base de datos. (Anexo código fuente A2)

X. FASE 2. CLASIFICACIÓN DE HECHOS PROBABLES Y DIMENSIONES

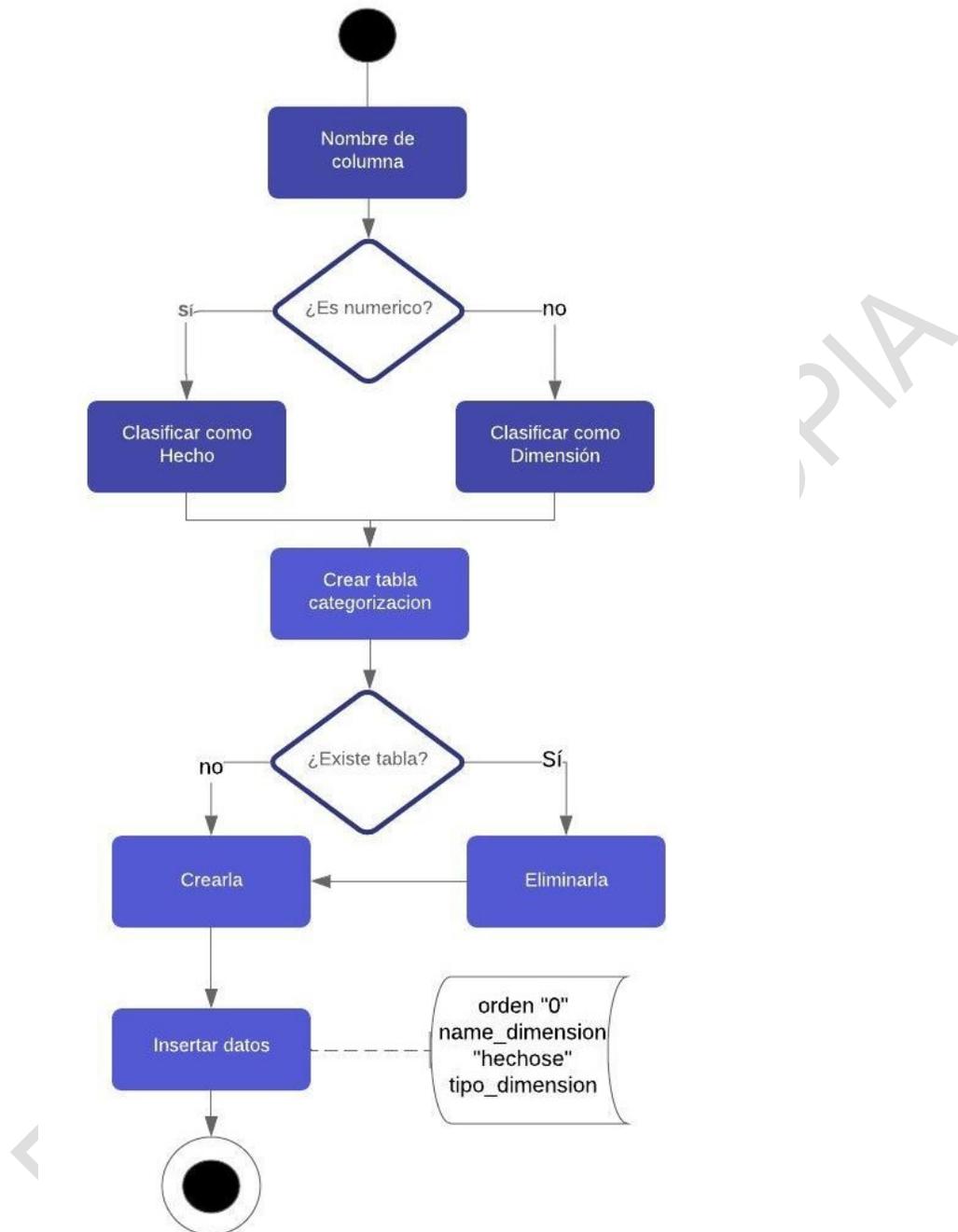


Figura 11: Clasificación y creación de tabla categorización contenedora de columnas en hechos probables y dimensión mediante una función.

(Anexo código fuente A3)

En la figura 11, se observa la función con la cual se clasifica y crea una tabla nueva contenedora de la consulta, mediante un cursor, que mediante unos parámetros internos permite crear la tabla y almacenar los datos de la consulta, denominando a dimensión y hechos, el orden, nombre de la columna y nombre de la dimensión.

	ordern character varying (2)	column_name character varying (50)	tipo_dimension character varying (3)	name_dimension character varying (50)
1	0	grocery	d	hechose
2	0	frozen	d	hechose
3	0	detergents_paper	d	hechose
4	0	delicatessen	d	hechose
5	0	channel	d	hechose
6	0	region	d	hechose
7	0	anio	h	hechose
8	0	trimestre	h	hechose
9	0	mes	h	hechose
10	0	departamento	d	hechose
11	0	municipio	d	hechose

Figura 12: Visualización de tabla categorizada generada.

En la figura 12, se observa la tabla que se genera después del proceso anterior completado, se crea la tabla con un orden inicializado en "0"(cero), nombre de la columna correspondiente, el tipo de dimensión (si es hecho o dimensión en función al script) y el nombre de la columna el cual se genera como "hechose".

Esta tabla será posteriormente editada por el usuario experto, quien le dará el orden correspondiente mediante la interacción con la aplicación, el cambio en el tipo de dimensión (hecho o dimensión) y nombre de dimensión por aquel que le corresponda según su criterio.

X-A. Selección de hechos probables



Figura 13: Selección de hechos probables y el orden mediante la aplicación.
(Anexo código fuente C4)

Con la clasificación realizada en el proceso anterior de la figura 12, se procede, como se muestra en la figura 13, a realizar la selección sobre los hechos probables, el usuario experto es quien según su criterio dará una clasificación final sobre las columnas y elegirá si es un hecho o se cambia por una dimensión.

En este paso también selecciona el orden de la columna según el criterio del usuario para ello deberá analizar la información que se le suministre.

X-B. Creación de categorías



Figura 14: Creación, modificación y eliminación de categorías.
(Anexo código fuente C3)

En la figura 14, se encuentra el listado de las categorías que se crearon por defecto, seguido de las opciones de insertar, modificar y eliminar categoría, que serán utilizadas para realizar el agrupamiento de categorías.

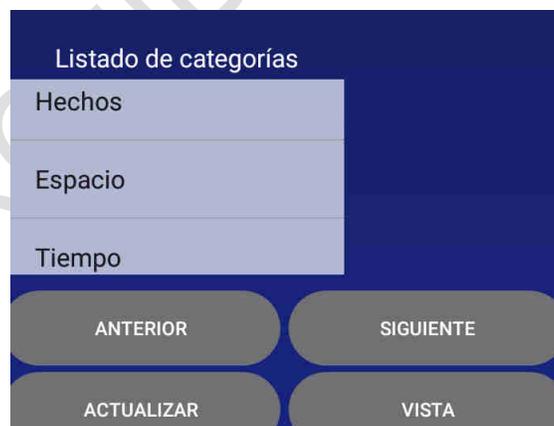
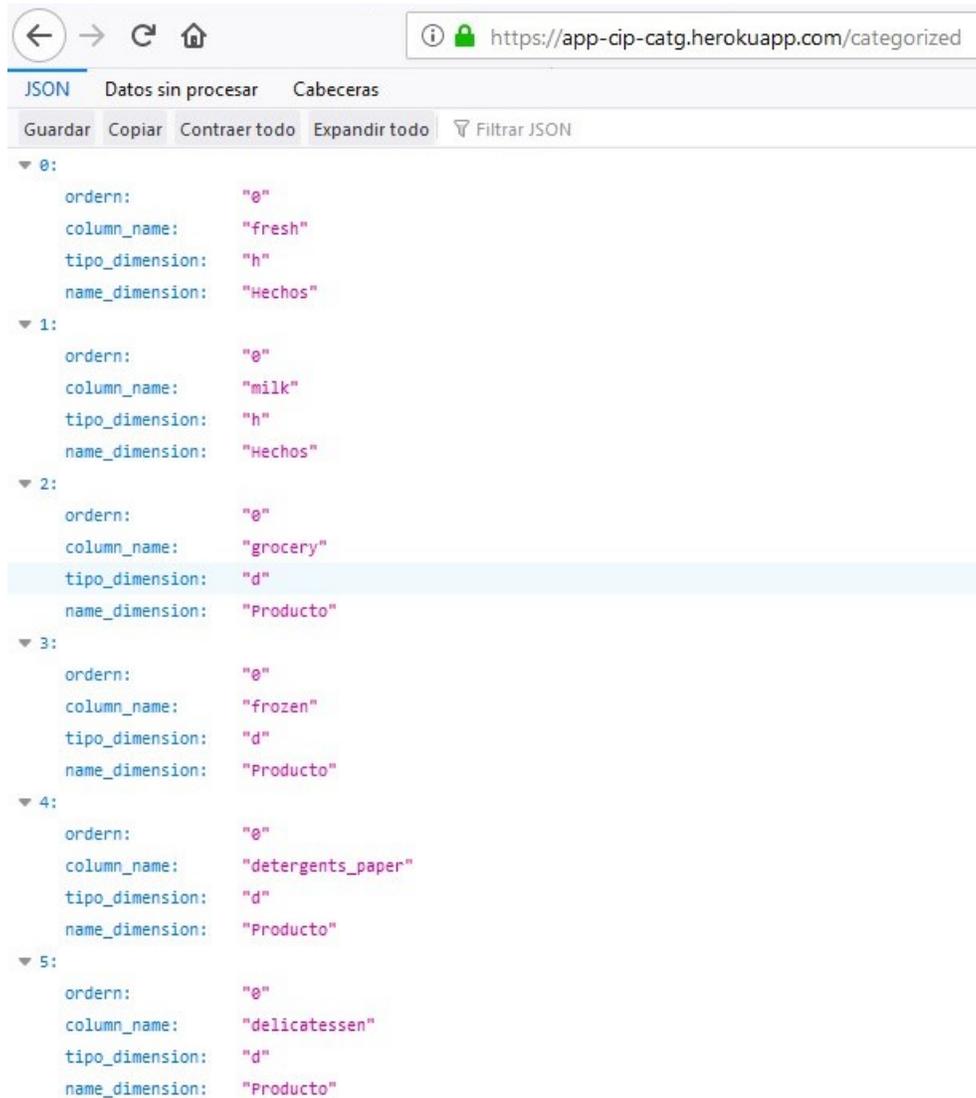


Figura 15: Selección en agrupación del espacio por cada columna.
(Anexo código fuente C4)

En la figura 15, se muestra el proceso de agrupación del espacio, donde el usuario experto bajo su criterio clasifica a qué categoría pertenece cada una de las columnas, es así que da como finalizado el proceso de selección y clasificación de la información.

X-C. Generar datos categorizados



```
JSON Datos sin procesar Cabeceras
Guardar Copiar Contraer todo Expandir todo Filtrar JSON
0:
  orden: "0"
  column_name: "fresh"
  tipo_dimension: "h"
  name_dimension: "Hechos"
1:
  orden: "0"
  column_name: "milk"
  tipo_dimension: "h"
  name_dimension: "Hechos"
2:
  orden: "0"
  column_name: "grocery"
  tipo_dimension: "d"
  name_dimension: "Producto"
3:
  orden: "0"
  column_name: "frozen"
  tipo_dimension: "d"
  name_dimension: "Producto"
4:
  orden: "0"
  column_name: "detergents_paper"
  tipo_dimension: "d"
  name_dimension: "Producto"
5:
  orden: "0"
  column_name: "delicatessen"
  tipo_dimension: "d"
  name_dimension: "Producto"
```

Figura 16: Datos categorizados en formato Json.

En la figura 16, muestra el último paso a realizarse, el resultado que se genera bajo los procesos anteriores culminados detallados en la figura 13 y 14 de selección y clasificación de datos, con el desarrollo completo satisfactoriamente, se generan datos en formato Json donde se detalla su orden, nombre de columna, tipo de dimensión (si fueron catalogadas como Hecho o Dimensión) y el nombre de la dimensión (categoría). Datos que fueron previamente seleccionados y analizados por el usuario experto.

Estos datos se entregan como resultado de la categorización al siguiente módulo de topología de cubos, para continuar con su proceso.

XI. FASE 3: VISUALIZACIÓN DE CATEGORÍAS

XI-A. Conexiones con Api

```
JS conection.js ×
src > conection > JS conection.js > ...
1  const {Client} = require('pg');
2  //const keys = require('../config/keys.json'); // servidor local
3  const keys = require('../config/keysHeroku.json'); // servidor heroku
4
5  class Connection {
6    constructor(){
7      return new Client(keys);
8    }
9  }
10
11  module.exports = Connection;
```

Figura 17: Código configuración conexión API Categorización de Datos.

En la figura 17, se observa la configuración de la conexión que se ha realizado de la Api a la base de datos PgAdmin con el servidor Heroku.

```
{} keysHeroku.json ×
src > config > {} keysHeroku.json > ...
1  {
2    "user": "rdpyljgcqkuiwf",
3    "password": "aae71da605626632b88ed6cca1ab1e07c831260c3c1047adcd051178e882b0f0",
4    "host": "ec2-174-129-242-183.compute-1.amazonaws.com",
5    "database": "d9m6kh6bqcdpte",
6    "port": 5432
7  }
```

Figura 18: Código Credenciales de Conexión API Categorización de Datos.

En la figura 18, se detalla la configuración de la conexión con el servidor, es decir, las credenciales de autenticación utilizadas de la Api a la plataforma Heroku .

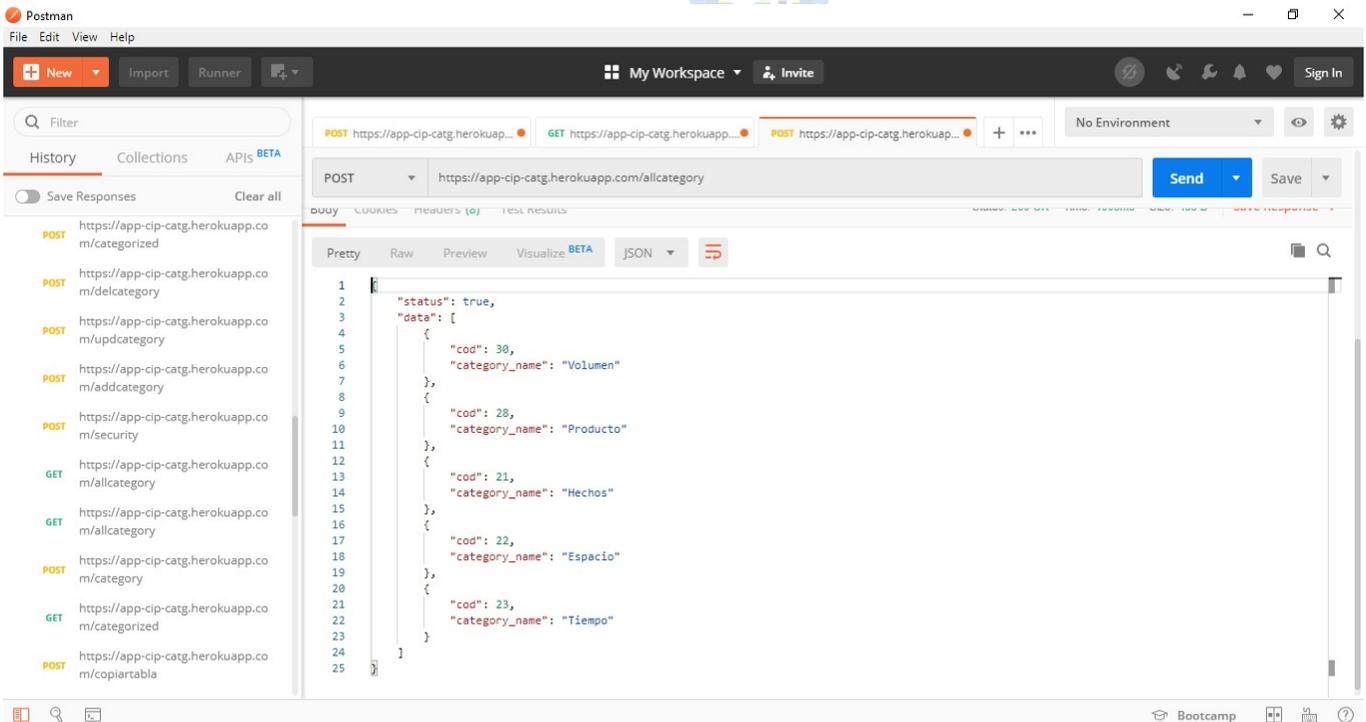


Figura 21: Lista de categorías
(Anexo código fuente Api B3)

En la Figura 21, a través de un método POST se visualiza las categorías que se encuentran creadas.

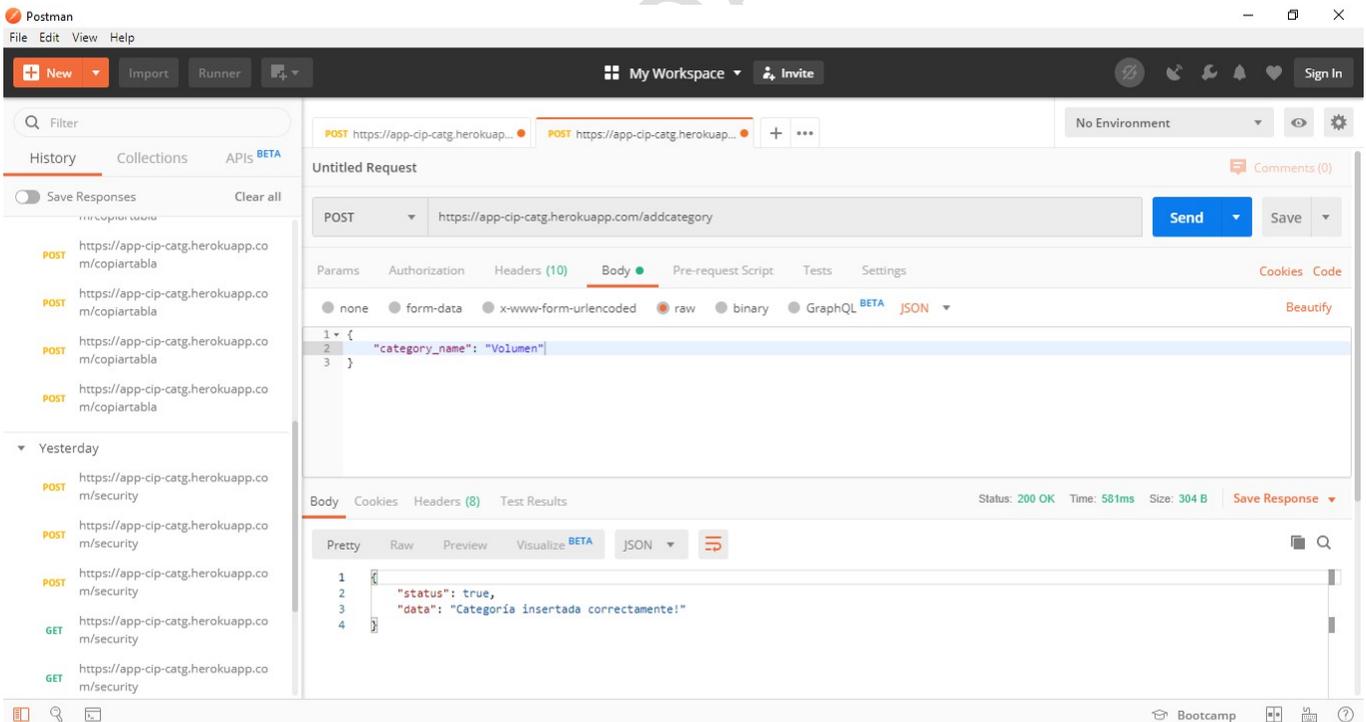


Figura 22: Insertar una nueva categoría
(Anexo código fuente Api B3)

En la Figura 22, a través de un método POST se ingresa una nueva categoría.

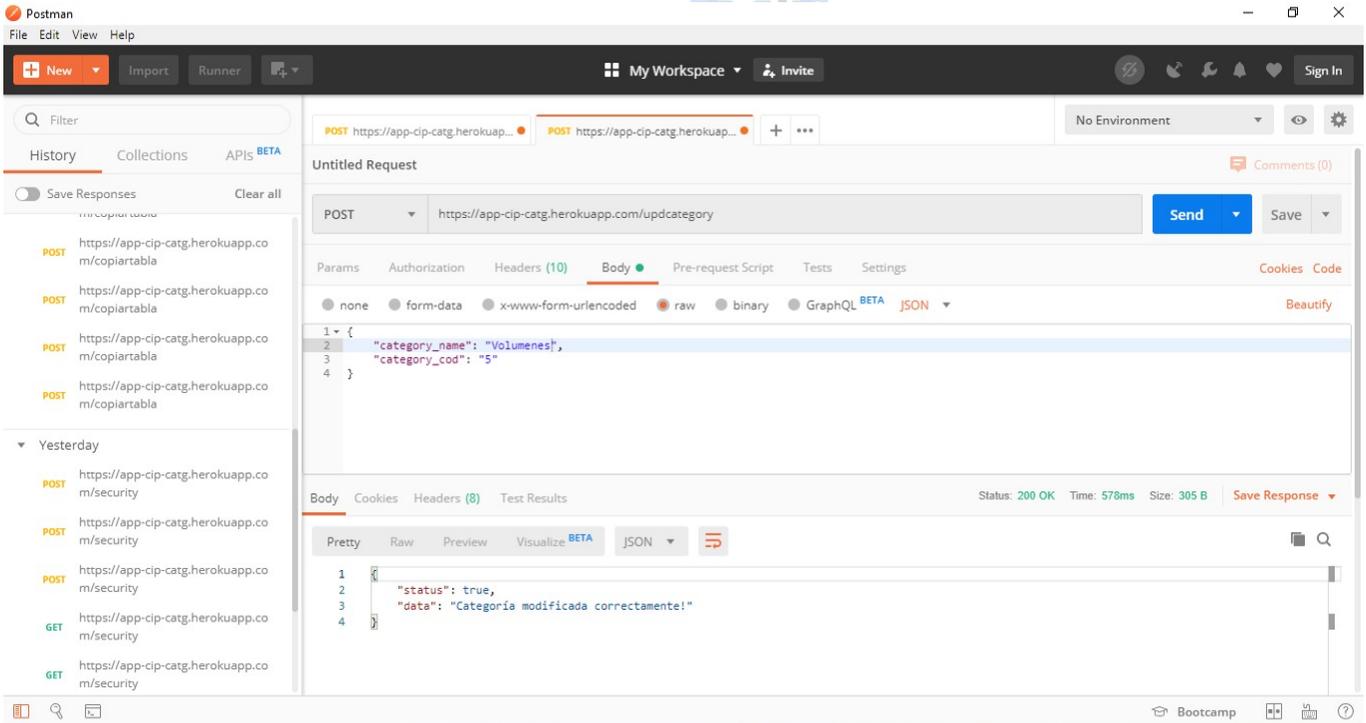


Figura 23: Modificar una nueva categoria
(Anexo codigo fuente Api B3)

En la Figura 23, a traves de un metodo POST se edita o modifica una categoria.

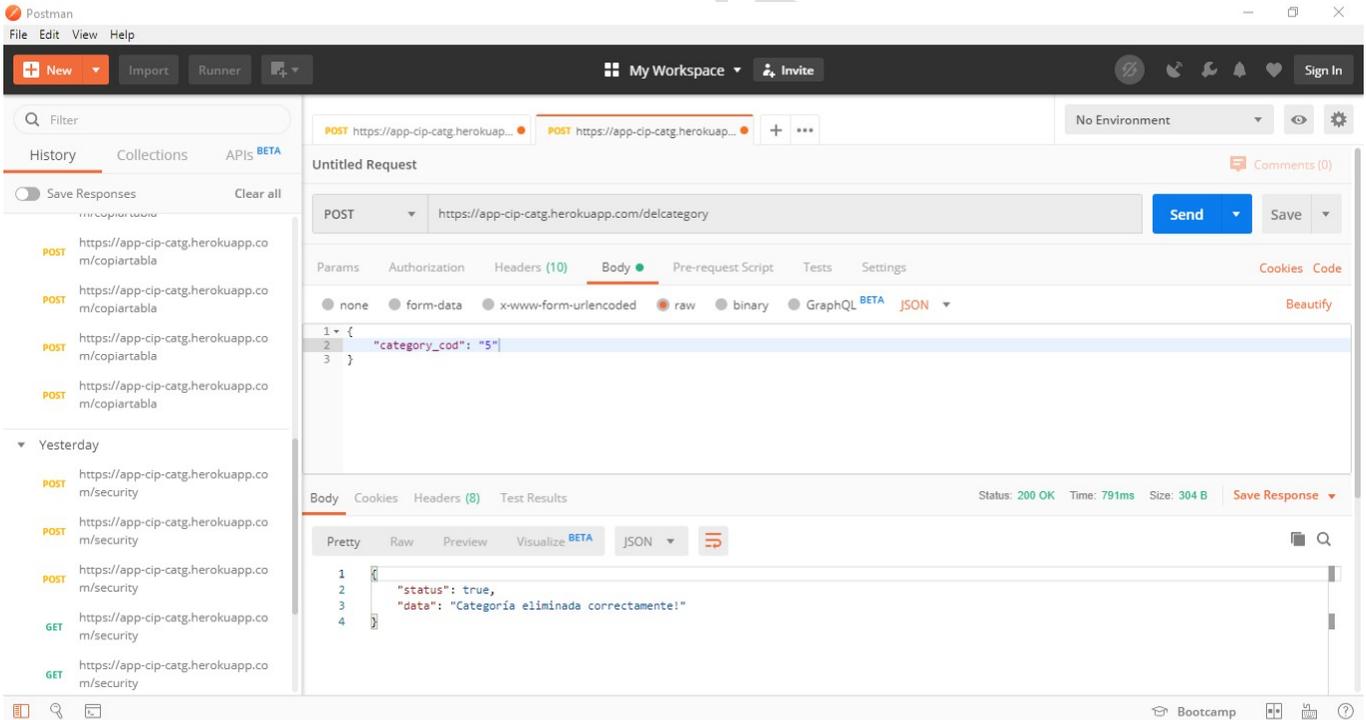


Figura 24: Eliminar una nueva categoria
(Anexo codigo fuente Api B3)

En la Figura 24, a traves de un metodo POST se elimina una categoria.

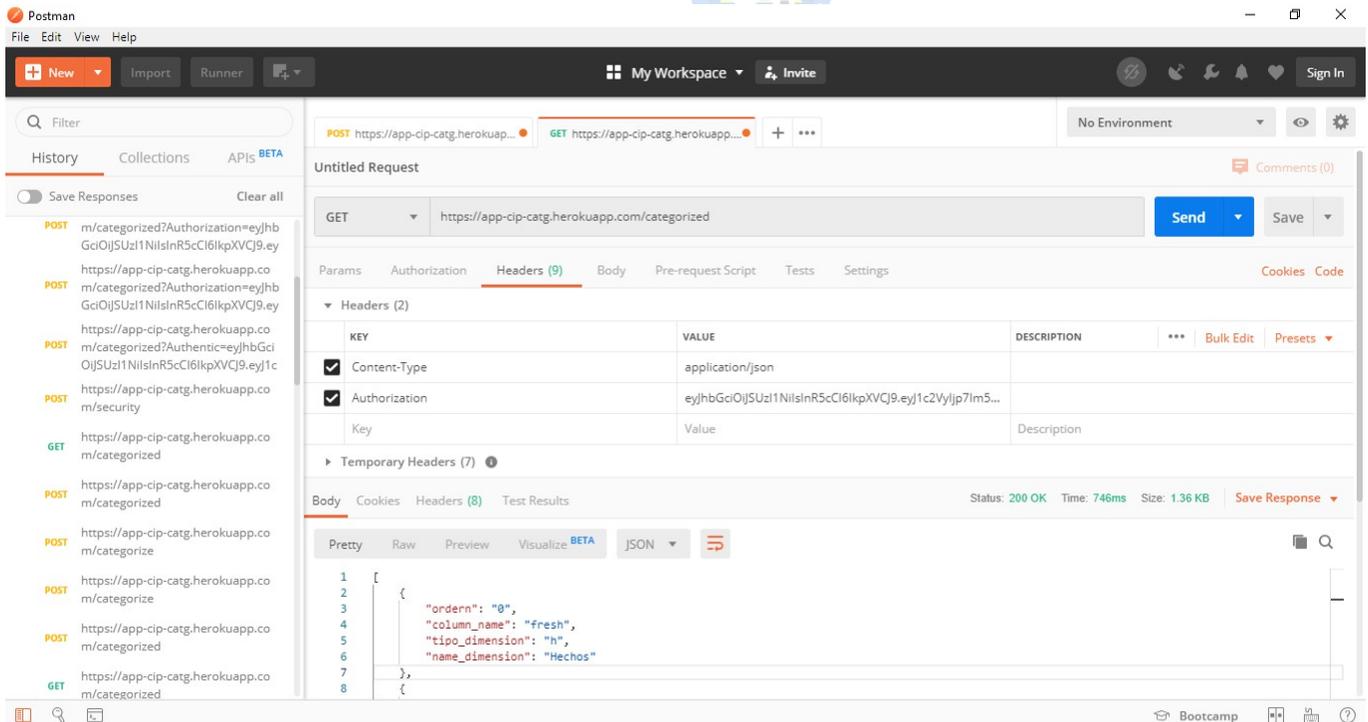


Figura 25: Categorizar la tabla discretizada

En la Figura 25, a través de un método POST el usuario experto en el tema se encarga de categorizar los datos de la tabla, colocando su orden, el tipo y nombre de la dimensión.

XI-C. Generación de vistas



Figura 26: Ingreso de credenciales para Logueo (Anexo código fuente C1)

En la Figura 26, se visualiza los campos para ingresar las credenciales y tener acceso a las funciones de la App.



Figura 27: Vista de inicio

En la Figura 27, se observa la vista principal de la App, una pequeña descripción de Categorización y sus desarrolladores.



Figura 28: Vista del menú

En la Figura 28, se encuentran las diferentes opciones que permiten llevar a cabo el proceso de categorización.



Figura 29: Copiar ruta de la tabla
(Anexo código fuente C2)

En la Figura 29, permite agregar la ruta de la base de datos discretizada del módulo anterior y hace una copia.



Figura 30: Creación de categorías
(Anexo código fuente C3)

En la Figura 30, se muestra el listado de las categorías o dimensiones disponibles, además de las opciones de insertar, modificar y eliminar categorías.



Figura 31: Categorización y orden de columnas (Anexo código fuente C4)

En la Figura 31, se muestra la vista de categorización, dividiéndose en 4 secciones, en la primera, se permite seleccionar el orden, en la segunda se visualiza el nombre de la columna, en la tercera, se hace una selección de hecho o dimensión, en la cuarta, está el listado de las dimensiones o categorías creadas, en la quinta, hay unos botones que realizan la acción de ir una columna antes (anterior), una columna después (siguiente), la actualización o modificación de los datos (actualizar) y esta vista, que permite cargar la tabla que se va a categorizar.

ORDERN	COLUMN_NAME	TIPO_DIMENSION	NAM
0	fresh	h	Hech
0	milk	h	Prod
0	grocery	d	Prod
0	frozen	d	Volu
0	detergents_paper	d	Prod
0	delicatessen	d	Prod
0	channel	d	Hech
0	region	d	Espa
0	anio	h	Tierr
1	trimestre	d	Tierr
2	mes	d	Tierr
0	departamento	d	Espa
1	municipio	d	Espa

Figura 32: Tabla Categorizada (Anexo código fuente C5)

En la Figura 32, se detalla a través de una tabla el resultado final de los datos categorizados, con los encabezados orden, nombre de columna, tipo de dimensión y nombre de dimensión.

XI-D. Validación y pruebas

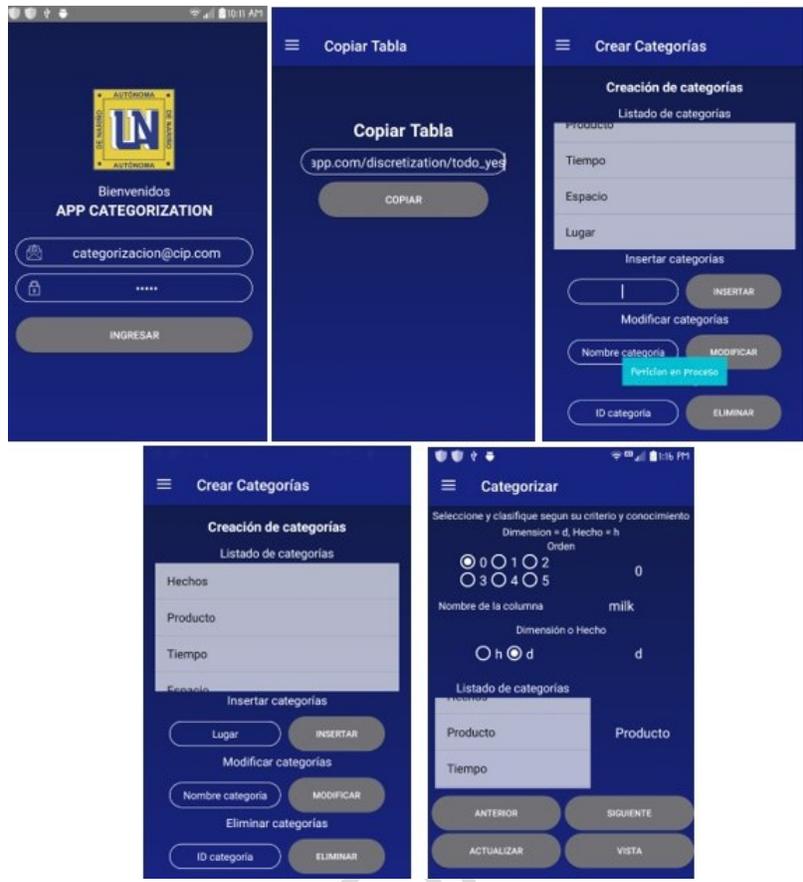


Figura 33: Diseño de vistas en pruebas

En la figura 33, se detalla la aplicación en función mediante pruebas realizadas para detectar errores antes de implementarlo y presentarlo.

ORDERN	COLUMN_NAME	TIPO_DIMENSION	NAM
0	fresh	h	Hecl
0	milk	h	Prod
0	grocery	d	Prod
0	frozen	d	Volu
0	detergents_paper	d	Prod
0	delicatessen	d	Prod
0	channel	d	Hecl
0	region	d	Espa
0	anio	h	Tierr
1	trimestre	d	Tierr
2	mes	d	Tierr
0	departamento	d	Espa
1	municipio	d	Espa

Figura 34: Resultado tabla categorizada final

En la figura 34, se muestra la tabla categorizada en función de los procesos de prueba y el resultado final que muestra.

XII. CONCLUSIONES Y RECOMENDACIONES

XII-A. Conclusiones

En el proceso de ejecución de este proyecto, sobre el desarrollo de una aplicación distribuida para el apoyo a la categorización mediante análisis exploratorio, hemos realizado procesos, los cuales han ayudado a que la herramienta tenga un buen proceso y funcionamiento con la metodología implementada, lo que conlleva a destacar aspectos como:

Los diagramas UML los cuales se realizaron como guía y a la vez de manera explicativa, con el fin de detallar procesos y actividades en el funcionamiento del sistema, sobre los que de manera interactiva son un apoyo al momento de implementar y hacerlos posibles mediante las herramientas utilizadas.

El manejo de herramientas con implementación de código abierto, como Kotlin, que permite obtener la visualización de la categorización y la implementación de Scripts en funciones con la base de datos PgAdmin y Postgres en el manejo de Scripts SQL.

La implementación de herramientas como Android, permite la visualización de cada pre-procesamiento y el replicar la información mediante formatos Json, permitiendo así que cada una de las aplicaciones se realice de forma modular mediante la implementación de sistemas distribuidos.

Una buena orientación por parte de los asesores, brindándonos información oportuna, clara y herramientas de trabajo específicas que aportarían de gran manera al crecimiento y avance del proyecto que a pesar del tiempo se ha logrado culminar satisfactoriamente.

XII-B. Recomendaciones

Al finalizar el proceso de desarrollo de la aplicación se hacen evidentes aspectos futuros que ayuden al mejoramiento de la herramienta y se tome como referencia en trabajos relacionados, en lineamientos metodológicos como en técnicos sobre los cuales se evidenciaron tanto aspectos positivos como también mejoras que aporten en el proceso y desarrollo de aplicaciones.

Tomar como referencia la aplicación en el desarrollo de más proyectos, que apliquen de forma modular herramientas de completos y lentos procesos, dentro de empresas que requieran de procesos más ágiles sin necesidad de arriesgar información ni someterse a largas esperas.

El manejo de bases de datos normalizadas ya que en el proyecto se trabajó con bases no normalizadas, con el fin de trabajar y hacer pruebas a lo largo de la creación del proyecto.

XIII. MARCAS REGISTRADAS

XIII-1. Node.js: Es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google.

Autor original: Ryan Dahl - Licencia: Licencia MIT - Última versión estable: 12.9.1 - Desarrollador(es): Node.js Foundation - Programado en: C++, JavaScript - Escrito en: C, C++, JavaScript.



Figura 35: Node.js.

XIII-2. PostgreSQL: Es un sistema de gestión de bases de datos relacional orientado a objetos y de código abierto, publicado bajo la licencia PostgreSQL, similar a la BSD o la MIT.

Escrito en: C - Fecha del lanzamiento inicial: 8 de julio de 1996 - Última versión estable: 11.5 - Licencia: PostgreSQL License - Desarrollador(es): PostgreSQL Global Development Group - Lanzamiento inicial: 1996.



Figura 36: PostgreSQL

XIII-3. Android: Es un sistema operativo móvil desarrollado por Google, basado en el Kernel de Linux y otros software de código abierto. Fue diseñado para dispositivos móviles con pantalla táctil, como teléfonos inteligentes, tabletas, relojes inteligentes, automóviles y televisores.

Sistema de gestión de paquetes: Google Play, APK - Lanzamiento inicial: 23 de septiembre de 2008 - Tipo de núcleo: Monolítico - Modelo de desarrollo: Código abierto - Licencia: Apache 2.0 y GNU GPL 2 - Última versión estable: Android 10.



Figura 37: Android

XIII-4. Kotlin: Kotlin es un lenguaje de programación de tipado estático que corre sobre la máquina virtual de Java y que también puede ser compilado a código fuente de JavaScript. Es desarrollado principalmente por

JetBrains en sus oficinas de San Petersburgo.

Sistema de tipos: estático - Extensiones comunes: kt - Diseñador: JetBrains - Influído por: Java, Scala, Groovy, C, Gosu - Última versión estable: 1.3.31 - Última versión en pruebas: 1.3 RC.



Figura 38: Kotlin

XIII-5. PL/pgSQL: Es un lenguaje imperativo provisto por el gestor de base de datos PostgreSQL. Permite ejecutar comandos SQL mediante un lenguaje de sentencias imperativas y uso de funciones, dando mucho más control automático que las sentencias SQL básicas.

Influenciado por: PL/SQL, Ada.



Figura 39: PL/pgSQL

XIII-6. Heroku: Es una plataforma como servicio de computación en la Nube que soporta distintos lenguajes de programación. Heroku es propiedad de Salesforce.com.

Organización principal: Salesforce.com - Fundación: 2007 - Oficinas centrales: San Francisco, California, Estados Unidos - Tipo: subsidiary - Persona clave: Margaret Francis - Fundadores: James Lindenbaum, Adam Wiggins, Orion Henry.



Figura 40: Heroku

ANEXOS

A. Scripts de funciones medianamente acopladas al SGBD PostgreSQL

A1. *Visualizar información base de datos:* Se muestra el script ejecutado desde la base de datos para buscar la información de las columnas en donde se especifica el tipo de dato que tiene la columna, en el cual se precisa el nombre de la tabla.

```
select table_name, column_name, data_type from information_schema.columns
where table_schema = 'public'
and table_name = 'categories'
```

Figura 41: Consulta SQL para visualizar la información de la base de datos a través de PgAdmin.

A2. *Clasificación en hechos probables y dimensiones:* Se genera un script por medio del cual separamos las columnas en Hechos(H) Y Dimensiones(D) dependiendo de la información obtenida con la cual genera el nombre de la columna y su correspondiente clasificación, en el cual se precisa el nombre de la tabla.

```
SELECT column_name, CASE WHEN numeric_precision is null THEN 'D' ELSE 'H' END as isDimension
FROM information_schema.columns
WHERE table_schema = 'public'
AND table_name = 'categories';
```

Figura 42: Script de clasificación en hechos probables y dimensión mediante una función en la base de datos.

A3. *Función clasificación y creación de tabla categorizada:* En la figura – se muestra una función por medio de la cual se clasifica si la columna es Hecho(H) o Dimensión(D) y posteriormente se crea una tabla en la cual tiene un Orden que inicia en “0” (cero), nombre de la columna correspondiente, el tipo de dimension (si es hecho o dimension en funcion al script) y el nombre de la columna el cual se genera como ”hechose”.

La tabla que es generada sera posteriormente editada por el usuario experto, quien le dara el orden correspondiente mediante la interacción con la aplicación, el cambio en el tipo de dimension(hecho o dimension) y nombre de dimensión por aquel que le corresponda segun su criterio.

```
2 CREATE OR REPLACE FUNCTION classFact(nameTableNew text, nameTableRows text) RETURNS TEXT AS
3 $BODY$
4 DECLARE
5     query1_sql text:= '';
6     sql_insert text:= '';
7     reg RECORD;
8     cur_isDim CURSOR FOR
9         SELECT column_name, CASE WHEN numeric_precision is null THEN 'd' ELSE 'h' END as isDimension
10        FROM information_schema.columns
11        WHERE table_schema = 'public' AND table_name = '||nameTableRows||';
12
13 BEGIN
14     EXECUTE 'DROP TABLE IF EXISTS public.||nameTableNew';
15     EXECUTE 'CREATE TABLE IF NOT EXISTS public.||nameTableNew||'(ordern character varying(2),column_name character varying(50),
16     tipo_dimension character varying(3), name_dimension character varying(50))';
17
18     OPEN cur_isDim;
19     FETCH cur_isDim INTO reg;
20     WHILE(FOUND) LOOP
21         sql_insert:= 'INSERT INTO public.||nameTableNew||'(ordern,column_name,tipo_dimension,name_dimension) VALUES
22         |('0','||reg.column_name||',' ||reg.isDimension||',' 'hechose' ');
23         EXECUTE sql_insert;
24         --RAISE NOTICE '%', sql_insert;
25         FETCH cur_isDim INTO reg;
26     END LOOP;
27     CLOSE cur_isDim;
28     RETURN 'OK';
29 END;
30 $BODY$
31 LANGUAGE 'plpgsql';
32 SELECT classFact('','');
33
34
35 SELECT * FROM public.category;
36 SELECT * FROM public.categorized;
37 UPDATE categorized SET ordern='0', tipo_dimension='h', name_dimension='Producto' WHERE column_name='fresh';
38 SELECT ordern, column_name, tipo_dimension, name_dimension FROM public.categorized WHERE column_name='fresh';
```

Figura 43: Función clasificación y creación de tabla categorizada

B. Código fuente del API

B1. Instalación paquetes: En este espacio se encuentra la configuración de paquetes y módulos que son necesarios para el funcionamiento de la API.

```
{} package.json ×
{} package.json > ...
1  {}
2  "name": "api_categ",
3  "version": "1.0.0",
4  "description": "",
5  "main": "index.js",
6  "dependencies": {
7    "body-parser": "^1.19.0",
8    "express": "^4.17.1",
9    "fs": "0.0.2",
10   "jsonwebtoken": "^8.5.1",
11   "nodemon": "^1.19.2",
12   "path": "^0.12.7",
13   "pg": "^7.12.1",
14   "request": "^2.88.0",
15   "sha256": "^0.2.0",
16   "util": "^0.12.1"
17 },
18 "devDependencies": {},
19 "scripts": {
20   "start": "nodemon src/index.js",
21   "test": "echo \"Error: no test specified\" && exit 1",
22   "dev": "nodemon src/index.js"
23 },
24 "author": "Franck Burbano - Leidi Pereira",
25 "license": "ISC"
26
27
```

Figura 44: Instalación de paquetes en Api.

B2. Código index: Se encuentra la configuración que se realiza con los puertos para el traspaso de información mediante los métodos Get y Post en los cuales como primer paso se realiza un logueo mediante un token y con ello la interacción mediante Post y Get para tener la interacción a la base de datos y cada una de sus vistas

```
src > JS indexjs > ...
5  const dataCtrl = require('./controllers/category');
6  const authlog = require('./middleware/TokenJWT');
7  const copyTable = require('./controllers/verificarTabla');
8
9  // Config
10 app.set('port', process.env.PORT || 3002);
11 app.use(bodyParser.urlencoded({ extended: false }));
12 app.use(bodyParser.json());
13
14 // Security
15 app.post('/security', new authlog().Login);
16
17 // Routers
18 app.get('/categories', /*new authlog().ensureToken,*/ dataCtrl.getCategories);
19 app.get('/categorized', /*new authlog().ensureToken,*/ dataCtrl.getCategorized);
20 app.post('/categorize', new authlog().ensureToken, dataCtrl.postCategorize);
21 app.post('/addcategory', new authlog().ensureToken, dataCtrl.postAddCategory);
22 app.post('/updcategory', new authlog().ensureToken, dataCtrl.postUpdCategory);
23 app.post('/delcategory', new authlog().ensureToken, dataCtrl.postDelCategory);
24 app.post('/allcategory', new authlog().ensureToken, dataCtrl.postAllCategory);
25 app.post('/copiantabla', new authlog().ensureToken, new copyTable().SetConsumerTable);
26 app.get('*', (req, res) => {
27   res.json({ msj: 'API Postgres No tienes acceso'})
28   res.end();
29 });
30
31 // Listen Server
32 app.listen(app.get('port'), function(){
33   console.log('Servidor Iniciado...',app.get('port'))
34 });
35
```

Figura 45: Código Configuración y Rutas API Index Categorización de Datos.

B3. Configuración y rutas en categorías: En esta parte se encuentra el código de las rutas en donde se detalla el paquete utilizado para realizar las conexiones hacia las funciones de interacción con la base de datos y el funcionamiento de cada una de las actividades que se realiza en la app.

Se muestra el pseudocódigo de rutas como el eliminar, agregar, listar categorías y la tabla categorizada.

```
src \ controllers > JS category.js : _
1  const Connection = require('../conexion/connection');
2
3  // Routers
4  const getCategorias = (req, res) => {
5
6      let con = new Connection();
7      con.connect();
8
9      con.query('SELECT * FROM public.categories')
10     .then( r => {
11         //res.json({status : true, data : r.rows})
12         res.json(r.rows)
13         res.end()
14         con.end();
15     })
16     .catch (err => {
17         console.log(err);
18         res.json({status : false, data : err})
19         res.end()
20         con.end();
21     });
22 };
23
24 > const getCategorized = (req, res) => { ...
42 };
43
44 > const postCategorize = (req, res) => { ...
62 };
63
64 > const postAddCategory = (req, res) => { ...
83 };
84
85 > const postUpdCategory = (req, res) => { ...
104 };
105
106 > const postDelCategory = (req, res) => { ...
125 };
126
127 > const postAllCategory = (req, res) => { ...
146 };
147
148 module.exports = {
149     getCategorias,
150     getCategorized,
151     postCategorize,
152     postAddCategory,
153     postUpdCategory,
154     postDelCategory,
155     postAllCategory
```

Figura 46: Configuración rutas en Api

B4. Login: A continuación, se indica el proceso de validación de credenciales al ser enviado el correo electrónico y la contraseña insertados en la aplicación para ser posteriormente validados por la API y dependiendo del resultado enviar a la vista de inicio o imprimir un mensaje de datos inválidos.

```
2 const Connection = require('../conexion/conexion');
3 const sha256 = require('sha256');
4
5 // # class #####
6
7 class Usuario{
8
9   async login(identificacion, password){
10
11     let con = new Connection();
12     con.connect();
13
14     const res = await con.query(
15       'SELECT nombre, correo FROM usuarios WHERE correo = $1 AND contrasena = $2',
16       [identificacion, sha256(password)]
17     );
18
19     if(res.rows.length > 0) {
20       await con.end();
21       return res.rows[0];
22     }
23     else{
24       await con.end();
25       return null;
26     }
27
28   }
29
30 }
31
32 module.exports = Usuario;
```

Figura 47: Login de usuario en Api

C. Código fuente de la aplicación móvil y APK

CI. *Login*: En este espacio se muestra la validación de credenciales con usuario y contraseña para ser validada en la Api si los datos son correctos se enviará a la vista de inicio de lo contrario mostrará un mensaje de la petición fallo.

```
val client = AsyncHttpClient()
val params = RequestParams()

params.put("email", txt_username?.text.toString())
params.put("password", txt_password?.text.toString())

client.post( url: "https://app-cip-catg.herokuapp.com/security",
  params,
  object : AsyncHttpResponseHandler() {
    override fun onStart() {...}
    @SuppressWarnings("CommitPrefEdits")
    override fun onSuccess(
      statusCode: Int,
      headers: Array<Header>,
      response: ByteArray
    ) {...}
    override fun onFailure(
      statusCode: Int,
      headers: Array<Header>,
      errorResponse: ByteArray,
      e: Throwable
    ) {
      val data = String(errorResponse, kotlin.text.charset( charsetName: "UTF-8"))
      val objJSON = JSONObject(data)
      val msj = objJSON.getString( name: "data")

      btn_login?.isEnabled = true
      Toast.makeText(
        applicationContext,
        text: "La petición fallo (" + msj + ")",
        Toast.LENGTH_SHORT
      ).show()
    }
  }
```

Figura 48: Login de usuario en Api

C2. Copiar ruta: Código por el cual se copia la ruta formato Json en donde recibe los datos de la base discretizada en donde si el proceso se realiza de forma correcta nos muestra un mensaje de que la tabla fue creada correctamente.

```
val client = AsyncHttpClient()
val params = RequestParams()
params.put("ruta_tabla", txt ruta api?.text.toString())
txt ruta api?.setText("")
client.addHeader( header: "Authorization", MyPreferences(requireContext()).getToken().toString())
client.post( url: "https://app-cip-catg.herokuapp.com/copiartabla",
    params,
    object : AsyncHttpResponseHandler() {
        override fun onStart() {
            Toast.makeText(context, text: "Petición en proceso", Toast.LENGTH_SHORT)
                .show()
        }
        @SuppressWarnings("CommitPrefEdits")
        override fun onSuccess(
            statusCode: Int,
            headers: Array<Header>,
            response: ByteArray
        ) {
            val data = String(response, kotlin.text.charset( charsetName: "UTF-8"))
            Log.i( tag: "CADENA", data)
            val objJSON = JSONObject(data)
            val status = objJSON.getString( name: "status")
            if (status == "true") {
                Toast.makeText(
                    context,
                    text: "Tabla creada correctamente",
                    Toast.LENGTH_SHORT
                ).show()
                Log.i( tag: "Datos:", objJSON.getString( name: "data"))
                onFinish()
            } else {
                btn copiar?.isEnabled = true
                Log.i( tag: "ERROR", data)
            }
        }
    })
```

Figura 49: Copiar ruta de tabla

C3. Creación categorías: Código de la vista en la aplicación de las categorías por la cual el usuario podrá observar el listado de categorías existentes, donde podrá eliminarlas y crearlas según criterio personal.

```
private fun removerListView(): Boolean {...}
private fun cargar_categorias(): Boolean {
    btn upd categorias?.isEnabled = false
    btn del categorias?.isEnabled = false
    val client = AsyncHttpClient()
    client.addHeader( header: "Authorization", MyPreferences(requireContext()).getToken().toString())
    client.post( url: "https://app-cip-catg.herokuapp.com/alicategory",
        object : AsyncHttpResponseHandler() {
            override fun onStart() {...}
            @SuppressWarnings("CommitPrefEdits")
            override fun onSuccess(
                statusCode: Int,
                headers: Array<Header>,
                response: ByteArray
            ) {
                val data = String(response, kotlin.text.charset( charsetName: "UTF-8"))
                //Log.i(" datos ", data)
                val objJSON = JSONObject(data)
                val status = objJSON.getString( name: "status")
                if (status == "true") {
                    Toast.makeText(
                        context,
                        text: "Listado de categorías",
                        Toast.LENGTH_SHORT
                    ).show()
                    val datal = JSONArray(objJSON.getString( name: "data"))
                    Log.i( tag: " Nombres categorías", datal.toString())
                    removerListView()
                    if (datal.length() != 0) {
                        for (i in 0 until datal.length()) {
                            datal.getJSONObject(i)
                            lista_categorias.add(i, datal.getJSONObject(i))
                        }
                    }
                }
            }
        })
}
```

Figura 50: Creación de categorías

C4. *Categorizar*: Parte del código con el cual se categoriza los datos insertados sobre el cual nos da opciones en el orden, elección entre hecho y dimensión como la elección en la categoría a la cual pertenecería.

```
val client = AsyncHttpClient()
client.addHeader( header: "Authorization", MyPreferences(requireContext()).getToken().toString())
client.get( url: "https://app-cip-catg.herokuapp.com/categorized",
object : AsyncHttpResponseHandler() {
    override fun onStart() {
        Toast.makeText(context, text: "Petición en proceso", Toast.LENGTH_SHORT)
            .show()
    }
}
@SuppressLint( ...value: "CommitPrefEdits")
override fun onSuccess(
    statusCode: Int,
    headers: Array<Header>,
    response: ByteArray
) {
    val data = String(response, kotlin.text.charset( charsetName: "UTF-8"))
    //Log.i(" datos ", data)
    Toast.makeText(
        context,
        text: "Listado de columnas para categorizar",
        Toast.LENGTH_SHORT
    ).show()
    val datal = JSONArray(data)
    Log.i( tag: " Nombres de columnas", datal.toString())
    //removerListView()
    if (datal.length() != 0) {
        btn Back?.isEnabled = true
        btn Next?.isEnabled = true
        btn Update?.isEnabled = true
        mvid = 0
        num categorias = 0
        borrarListas()
        for (i in 0 until datal.length()) {
            datal.getJSONObject(i)
        }
    }
}
```

Figura 51: Categorización de columnas

C5. *Tabla categorizada*: Código con el cual se obtiene la tabla categorizada en donde se reemplaza los datos en orden, tipo dimensión (si es Hecho o Dimensión) y el nombre de la categoría, datos fueron cambiados por el usuario experto en referencia a la columna.

```
val client = AsyncHttpClient()
client.addHeader( header: "Authorization", MyPreferences(requireContext()).getToken().toString())
client.get( url: "https://app-cip-catg.herokuapp.com/categorized",
object : AsyncHttpResponseHandler() {
    override fun onStart() {
        Toast.makeText(context, text: "Petición en proceso", Toast.LENGTH_SHORT)
            .show()
    }
}
@SuppressLint( ...value: "CommitPrefEdits")
override fun onSuccess(
    statusCode: Int,
    headers: Array<Header>,
    response: ByteArray
) {
    val data = String(response, kotlin.text.charset( charsetName: "UTF-8"))
    Toast.makeText(
        context,
        text: "Tabla categorizada",
        Toast.LENGTH_SHORT
    ).show()
    //Log.i(" Json completo col:", objJSON.getString("data"))
    val datal = JSONArray(data)
    val data2 = datal.getJSONObject( index: 0)
    Log.i( tag: " Nombres de columnas", data2.toString())
    createHeaders( datal.getJSONObject( index: 0))
    for (i in 0..(datal.length()-1)) {
        setDataRow( datal.getJSONObject(i))
    }
    onFinish()
}
fun setDataRow(data: JSONObject) {
    var tableRow = TableRow(context)
}
```

Figura 52: Creación tabla categorizada

D. *Manual de instalación de cada producto anterior (Anexos externos)*

REFERENCIAS

- [1] Pablo Razeto-Barry, Tomás Veloz Gonzalez, and Christian Blanco. Acreditación en red: un sistema de acreditación distribuida para la educación continua. *Innoeduca. International Journal of Technology and Educational Innovation*, 3(2):146–158, 2017.
- [2] Amaury Cabarcas, Plinio Puello, and Raúl J Martelo. Sistema de información soportado en recuperación xml para pequeñas y medianas empresas (pyme) de cartagena de indias, colombia. *Información tecnológica*, 26(2):135–144, 2015.
- [3] Danny Fernando Sandoval Yate. *Desequilibrios regionales en Colombia: un análisis mediante la descomposición dinámica Shift Share*. PhD thesis, 2017.
- [4] Marsy Dayanna Ortiz Morales, Luis Joyanes Aguilar, and Lillyana María Giraldo Marín. Los desafíos del marketing en la era del big data. *e-Ciencias de la Información*, pages 1–31, 2016.
- [5] Lisset Alexandra Neyra Romero. Categorización automática de respuestas aplicando algoritmos de clasificación supervisada al análisis de las contestaciones de estudiantes a una serie de preguntas tipo test. 2016.
- [6] Ricard Martínez Martínez. Cuestiones de ética jurídica al abordar proyectos de big data. el contexto del reglamento general de protección de datos. *Dilemata*, (24):151–164, 2017.
- [7] Wanda C Rodríguez. Tecnologías de la información y la comunicación: nuevas configuraciones mentales y sus implicaciones para la educación. *Revista de psicología (Santiago)*, 27(1):199–210, 2018.
- [8] José A Moreiro González, Jorge Morato Lara, Sonia Sánchez Cuadrado, and Beatriz Ainhize Rodríguez Barquín. Categorización de los conceptos en el análisis de contenido: su señalamiento desde la retórica clásica hasta los topic maps. *Investigación bibliotecológica*, 20(40):13–31, 2006.
- [9] Leticia Calzada and José Luis Abreu. El impacto de las herramientas de inteligencia de negocios en la toma de decisiones de los ejecutivos. *Revista Daena (International Journal of Good Conscience)*, 4(2), 2009.
- [10] Carlos Alfredo Osorio Pérez. Sistema basado en una arquitectura distribuida con base de datos nosql para la recuperación de información de múltiples bibliotecas con óptimos niveles de rendimiento. 2016.
- [11] JA Araúzo, Ricardo del Olmo Martínez, JJ Laviós, and Juan José de Benito Martín. Programación y control de sistemas de fabricación flexibles: un enfoque holónico. *Revista iberoamericana de automática e informática industrial*, 12(1):58–68, 2015.
- [12] Andrew J Hewitt, Keith R Solomon, and EJP Marshall. Tamaño de la gota de aspersión, potencial de deriva y riesgo para los organismos no blanco del glifosato aplicado por vía aérea, para el control de la coca en colombia.
- [13] Inger S Padilla, Josefina Centeno, Carlos D Engwald, Alejandro N Gonzalez, Matías E Manzotti, Pablo Pascazzi, Diego A Waksman, and Martín M Díaz Maffini. Desarrollo e implementación de una herramienta para revisión y recodificación de datos de pacientes internados. In *VI Congreso Argentino de Informática y Salud (CAIS)-JAIIO 44 (Rosario, 2015)*, 2015.
- [14] Cynthia Lorena Corso, Alejandro García, Leonardo Ciceri, and Fernando Romero. Minería de datos aplicada a la detección de factores para la prevención de incidentes informáticos. In *XVI Workshop de Investigadores en Ciencias de la Computación*, 2014.
- [15] I Es I Ing. Población de ontologías con datos no estructurados utilizando herramientas de minería de datos.
- [16] Claudia Hernández and Jorge Enrique Rodríguez Rodríguez. Preprocesamiento de datos estructurados. *Revista vínculos*, 4(2):27–48, 2008.
- [17] María Ximena Dueñas Reyes. Minería de datos espaciales en búsqueda de la verdadera información. *Ingeniería y universidad*, 13(1):137–156, 2009.
- [18] Chema Maroto. Big data y su impacto en el sector público. *Harvard Deusto Business Review*, pages 16–25, 2017.
- [19] Serguio Manuel Molina. Ovejas negras. 2016.
- [20] Daniel Alejandro García López. Algoritmo de discretización de series tiempo basado en entropía y su aplicación en datos colposcópicas. 2007.
- [21] Gustavo Rivaderra. La metodología de kimball para el diseño de almacenes de datos (data warehouses). *Universidad Catolica de Salta*, pages 56–71, 2010.
- [22] David Loshin. *Business intelligence: the savvy manager's guide*. Newnes, 2012.
- [23] Sebastián Agustín Cortes Sierra and Julián David Rodríguez. Prototipo funcional usando cubos olap para generar análisis de indicadores de gestión. 2017.