

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

DISEÑO APLICATIVO WEB QUE PERMITA LA VISUALIZACIÓN DE INDICADORES EN LA PRODUCCIÓN AGROPECUARIA EN EL PAIS APLICANDO HERRAMIENTAS DE ANALITICA DE DATOS.

JULIE MARCELA GUTIERREZ PACHECO
jmgutierrezp@udistrital.edu.co

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
ESPECIALIZACIÓN INGENIERÍA DE SOFTWARE
BOGOTÁ D.C., COLOMBIA
MAYO 2024

DISEÑO APLICATIVO WEB QUE PERMITA LA VISUALIZACIÓN DE INDICADORES EN LA PRODUCCIÓN AGROPECUARIA EN EL PAIS APLICANDO HERRAMIENTAS DE ANALITICA DE DATOS.

JULIE MARCELA GUTIERREZ PACHECO

Trabajo de grado presentado como requisito parcial para optar al título de:

Especialista de Ingeniería de Software

Director: Ing. Beatriz Elisa Jaramillo Moreno, PhD

Revisor: Ing. Oswaldo Alberto Romero Villalobos, PhD

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD DE INGENIERÍA
ESPECIALIZACIÓN INGENIERÍA DE SOFTWARE
BOGOTÁ D.C., COLOMBIA
MAYO 2024

Resumen

La implementación de un aplicativo web que permita la visualización geográfica y alfanumérica de información del sector agrícola producto del análisis de datos realizado a diferentes fuentes de información empleado las herramientas *big data* para generar información relevante, la cual podrá ser consultada por varios usuarios por medio del aplicativo web desarrollado sin generar costos elevados, con restricciones de ingreso *login*, entorno visual agradable y con opciones de realizar filtros en la información.

Tabla de contenido

Resumen 3
Tabla de contenido 4
Lista de figuras 7
Lista de tablas 9
INTRODUCCIÓN 10
CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN 11
1. Descripción de la Investigación 12
1.1. Planteamiento del problema 12
1.2. Objetivos 12
1.2.1. Objetivo General 12
1.2.2. Objetivo Específicos 12
1.3. Justificación de la investigación 13
1.4. Hipótesis 13
1.5. Marco referencial 14
1.5.1. Marco teórico 14
1.5.2. Marco Metodológico 20
1.5.3. Marco legal 24
1.6. Metodología de la investigación 24
1.6.1 Tipo de estudio 24
1.6.2 Método de investigación 25
1.6.3. Fuentes y técnicas para la recolección de la información 25
1.6.4. Tratamiento de la información 25
1.7. Organización del trabajo de grado 25
1.8. Estudio de sistemas previos 26
2.1 Aplicación de la Metodología CRISP-DM 31
2.1.1 Comprensión del negocio 31
2.1.2 Comprensión de los datos 32
2.1.2 Preparación de los datos 36
2.1.4 Implementación 40
2.2. Arquitectura de software 41
2.2.1 Diagrama de componentes 42

2.2.1 Diagrama de despliegue 43
2.3. Metodología de desarrollo 44
2.3.1 Aplicación de la metodología Kanban 44
2.4. Análisis Funcional 45
2.4.1 Definición de requerimientos 45
2.4.2 Descripción de Requerimientos funcionales 46
2.4.3 Requerimientos no funcionales del proyecto 49
2.4.3 Casos de Uso 50
2.4.4. Diagramas de Actividades 52
2.4.5. Prototipo de la herramienta tecnológica 57
2.5. Modelo de clases 58
2.6. Modelo Relacional 60
3. Implementación del prototipo 60
3.1 Base de datos 61
3.2 Interfaz de usuario - FronEnd 62
3.2.1 Vista ingreso 65
3.3. Microservicio Gestión de usuarios 67
3.3.1. Configurar el Microservicio de Autenticación 67
3.3.2. Configuración de la Base de Datos 69
3.3.3. Creación del Modelo de Usuario 70
3.3.4. Creación del Controlador de Usuarios 71
3.3.4. Creación del Controlador de Productos 72
3.3.5. Definición de las Rutas usuarios, productos y validación de tokes. 73
3.3.6. Configurar modelo Servidor 75
3.3.7. Configurar Docker 76
3.3.8. Desplegar el Microservicio 77
3.3.9. Consumir el Microservicio desde la Aplicación Angular 77
3.3.10. Desplegar en AWS 79
CIERRE DE LA INVESTIGACIÓN 82
CAPÍTULO 4. RESULTADOS Y DISCUSIÓN 83
CAPÍTULO 5. CONCLUSIONES 83
5.1. Verificación, contraste y evaluación de los objetivos. 83
5.2. Síntesis del modelo propuesto 83

5.3. Aportes originales 84
CAPÍTULO 6. PROSPECTIVA DEL TRABAJO DE GRADO 84
6.1. Alcances y limitaciones 84
6.2. Líneas de investigación futuras 84
6.3. Trabajos de Investigación futuros 84
REFERENCIAS WEB 85
BIBLIOGRAFÍA 86

Lista de figuras

Figura 1. Etapas CRISP—DM Fuente: https://www.adictosaltrabajo.com/2021/01/14/metodologia-crisp-dm/ 21
Figura 2. Información datos libres sector Agrícola. Fuente: Elaboración propia. 33
Figura 3. Cargue de información del departamento. (Elaboración propia)..... 37
Figura 4. Cargue de información municipio. (Elaboración propia) 38
Figura 5. Información cargada. (Elaboración propia) 39
Figura 6. Relación de la información. (Elaboración propia) 40
Figura 7. Indicadores sector agrícola. (Elaboración propia) 41
Figura 8. Diagrama de Componentes. (Elaboración propia)..... 43
Figura 9. Diagrama de Despliegue. (Elaboración propia)..... 44
Figura 10. Tablero metodología diseño herramienta tecnológica. (Elaboración propia) ... 45
Figura 11. Caso de uso general. (Elaboración propia) 50
Figura 12. Caso de uso Gestión de Usuario. (Elaboración propia)..... 51
Figura 13. Caso de uso Gestión de información. (Elaboración propia)..... 51
Figura 14. Caso de uso Consulta información. (Elaboración propia)..... 52
Figura 15. Diagrama de actividades Gestión de Usuarios Administrador. (Elaboración propia) 53
Figura 16. Diagrama de actividad Gestión de usuario. (Elaboración propia) 54
Figura 17. Diagrama de actividad Consulta de información. (Elaboración propia)..... 55
Figura 18. Diagrama de actividad Consulta de información. (Elaboración propia)..... 56
Figura 19. Interfaz registro. (Elaboración propia) 57
Figura 20. Interfaz información agrícola. (Elaboración propia) 57
Figura 21. Interfaz flujo de caja. (Elaboración propia) 58
Figura 22. Modelo de clases. (Elaboración propia) 59
Figura 23. Modelo Relacional 60
Figura 24. Diagrama de Arquitectura. (Elaboración propia) 61
Figura 25. Estructura de la Tabla 'Usuario'. (Elaboración propia) 62
Figura 26. Herramientas instaladas. (Elaboración propia) 63
Figura 27. Distribución carpetas Frontend. (Elaboración propia)..... 63
Figura 28.user.service.ts. (Elaboración propia)..... 64
Figura 29.producto.service.ts. (Elaboración propia)..... 64
Figura 30.error.service.ts. (Elaboración propia) 65
Figura 31.Interfaz login. (Elaboración propia) 65
Figura 32. Interfaz registro. (Elaboración propia) 66
Figura 33. Vista información agrícola con mapa. (Elaboración propia)..... 66
Figura 34. Interfaz información agrícola con graficas. (Elaboración propia) 67
Figura 35. Carpetas Backend..... 68
Figura 36. Creación package.json. (Elaboración propia)..... 69
Figura 37.Configuración base de datos. (Elaboración propia)..... 70
Figura 38. Creación Modelo de Usuario. (Elaboración propia)..... 70
Figura 39. models product. (Elaboración propia)..... 71
Figura 40.Controlador user. (Elaboración propia) 72

Figura 41. Controlador producto. (Elaboración propia) 72
Figura 42. Ruta usuarios. (Elaboración propia) 73
Figura 43. Ruta productos. 74
Figura 44. Ruta validación tokes. 75
Figura 45. Configuración servidor. (Elaboración propia) 76
Figura 46. Configuración Docker. (Elaboración propia) 76
Figura 47. Desplegar el Microservicio. (Elaboración propia) 77
Figura 48. Configuración Microservicio. (Elaboración propia) 77
Figura 49. Servicio 'product.serve.ts'. (Elaboración propia) 78
Figura 50.error.server.ts.(Elaboración propia) 79
Figura 51.Desplegar en AWS 80
Figura 52.alizar le Frontend para consumir el Microservicio en AWS 81

Lista de tablas

Tabla 1.Hoja del data set departamento 34
Tabla 2. Hoja Data set Municipio 34
Tabla 3. Hoja Data set Características cultivos 34
Tabla 4. Hoja Data set Insumo 35
Tabla 5. Hoja Data set Tipo de cultivo 35
Tabla 6. Requerimientos 45
Tabla 7. Descripción REQ-q. Crear y actualizar usuarios. 46
Tabla 8.Descripción REQ-2. Administrar información. 47
Tabla 9. Descripción REQ-3. Control de Acceso. 47
Tabla 10. Descripción REQ-4. Persistencia de datos de los usuarios. 47
Tabla 11. Descripción REQ-5. Persistencia de la información. 48
Tabla 12. Descripción REQ-6. Consulta de información. 48
Tabla 13. Descripción REQ-7. Consulta de información flujo de caja. 48
Tabla 14. Requerimientos no funcionales. 49

INTRODUCCIÓN

Hoy hay muchos datos, que son valiosos, pero un dato por separado no nos ofrece más información de la que se ve a simple vista. Cada vez se desarrollan más herramientas para poder almacenarlos y procesarlos. La analítica de datos es el conjunto de acciones que permite recolectar y aprovechar los datos para encontrar tendencias y oportunidades de negocio para las empresas y presentarlos en un formato pictórico o gráfico para que se puedan analizar con facilidad.

De acuerdo con cifras del DANE para el año 2023 el 67,5% de hogares en las cabeceras poseían conexión a internet, lo que es un indicativo que muchos hogares usan y generan nueva información. Esta información es muy útil, es clave su recolección empresarial como gubernamental. La recopilación de esta información permite almacenar y analizar datos importantes, para identificar, extraer y transformarlos.

La extracción de datos consiste en recopilar información relevante desde una variedad de fuentes, como bases de datos, redes sociales, datos abiertos, paginas, entre otros. Una vez los datos han sido recolectados, entra la etapa de transformación. Aquí es donde se aplica una serie de técnicas y algoritmos para convertir los datos en una forma adecuada para el análisis, asegurando que estén limpios, estructurados y listos para su procesamiento.

Cuando los datos se extraen y se transforman, se inicia el análisis exploratorio. Esta etapa implica explorar los datos en busca de patrones, tendencias y relaciones que puedan ayudar a comprender mejor el problema y generar insights. Este análisis exploratorio se realiza con herramientas de visualización, estadística descriptiva, técnicas de minería de datos y aprendizaje automático. Este proceso ayuda a identificar la estructura de los datos y sus características más relevantes.

El proceso de extracción y transformación de datos no estaría completo sin una adecuada visualización de estos. Para este proyecto la visualización de los datos transformados hacia los usuarios se realizará por medio de un aplicativo web. En el cual se implementarán herramientas tecnológicas interrelacionadas como Power BI y Angular. Con la finalidad de que el usuario no tenga que instalar nuevas aplicaciones para ver la interrelación de la información y pronósticos.

PARTE I.

CONTEXTUALIZACIÓN DE LA INVESTIGACIÓN

1. Descripción de la Investigación

Este capítulo está enfocado en la exposición del problema de investigación que se aborda en su totalidad en este documento.

1.1. Planteamiento del problema

En la actualidad existe una gran cantidad de datos abiertos en el sector agropecuario en Colombia disponibles, creado por diferentes entidades con valiosa información, por lo cual se genera la necesidad de aplicar variedad de tecnologías para conectarlos, sacar el mayor provecho y generar predicciones que beneficien a este sector.

Uno de los campos tecnológicos que concentra diversas herramientas y metodologías para obtener información válida para la toma de decisiones a partir de los datos en bruto es la analítica de datos.

A través del análisis de datos, los agricultores pueden recopilar información precisa y detallada sobre diversos aspectos de su actividad, como las condiciones climáticas, la calidad del suelo, el uso de agua y fertilizantes, incrementos de precios en insumos y productos, cultivos y rendimientos por región.

Esta información se puede recopilar en un dashboard para mostrar interactivamente los datos y por medio de un aplicativo web transmitir esta información a los usuarios sin instalar otra herramienta.

1.2. Objetivos

1.2.1. Objetivo General

Construir e implementar un aplicativo web que permita la visualización de indicadores como cultivos por región, edad por cultivo, la calidad del suelo, incrementos de precios en insumos y productos por región en la producción agropecuaria en el país aplicando herramientas de analítica de datos para su integración.

1.2.2. Objetivo Específicos

- Analizar diferentes fuentes de información como documentos en pdf, Word, archivos Excel, entre otros, relacionados con el sector agropecuario en el país.
- Selección de software y elaboración del proceso ETL (Extract, Trasnform y Load).

- Elaborar tablero de mando (*Dashboard*) en *Power BI*.
- Construcción de un aplicativo web en Angular, con su correspondiente arquitectura.
- Publicación de la aplicación web en AWS.

1.3. Justificación de la investigación

Este proyecto busca intercambiar información con diferentes usuarios por medio de un aplicativo web en el cual se visualicen datos geográficos y alfanuméricos, de diferentes fuentes de datos abiertos relacionadas con el sector agropecuario.

Ante la gran cantidad de datos que se genera constantemente es necesario adoptar procesos TI como el Big data, analítica de datos, los cuales obtienen, estructuran, procesan y analizan estos grandes volúmenes de información con herramientas como ETL, Bases de datos, Dashboard, software como Power BI, SQL, lenguajes como Python, Javascript, entre otros.

Se emplea procesos ETL para obtener una integración consolidada de los datos de diferentes fuentes, formatos y su cargue en un repositorio de destino. Una base de datos la cual será usada en el software Power BI para graficar en un Tablero de mando (Dashboard) indicadores en nuestro caso de estudio del sector agrícola del país que nos pronostique incrementos en los valores de materia prima, ventas, costes, tipo de cultivo, vida del cultivo, regiones en las que se da cada tipo de cultivo, entre otros.

La aplicación de teorías y enfoques previos en el diseño e implementación de un aplicativo web, utilizando conjunto de técnicas como Power Query, Power BI, Javascript, Angular permitirá convertir datos sin procesar en información práctica. Implementando una serie de herramientas, tecnologías y procesos para encontrar tendencias, resolver problemas mediante datos y mediante la incrustación en un aplicativo web los usuarios puedan ver lo sin instalar otro software.

1.4. Hipótesis

La implementación de un aplicativo web que permitirá la visualización geográfica y alfanumérica de información del sector agropecuario producto del análisis de datos realizado a diferentes fuentes de información, empleando las herramientas Big Data para generar beneficios como predicciones, determinación de indicadores de mercado, insumos y productos.

1.5. Marco referencial

1.5.1. Marco teórico

1.5.1.1. Conceptos tecnológicos

1.5.1.1.1. Aplicación web

Programa y/o aplicación que se ejecuta en un entorno web y está en un servidor remoto para que cualquier usuario pueda usarla sin tener que realizar procesos de instalación, es decir, que las aplicaciones web sé adaptan a las necesidades para ejecutarse en un servidor con conexión a internet, esto con el fin que los usuarios puedan acceder desde cualquier dispositivo a través de cualquier navegador. También son conocidas por tener una funcionalidad similar a una aplicación de software de escritorio o una aplicación móvil con la diferencia por el entorno lógicamente web.

Una aplicación web es un software de aplicación que se ejecuta en un servidor web, a diferencia de los programas de software basados en computadora que se ejecutan localmente en el sistema operativo del dispositivo. El usuario accede a las aplicaciones web a través de un navegador web con una conexión de red estable. Estas aplicaciones se programan utilizando una estructura modelada cliente-servidor: el usuario recibe servicios a través de un servidor externo alojado por un tercero. Ejemplos de aplicaciones web de uso común incluyen: correo electrónico, ventas en línea, banca en línea, subastas en línea, entre otras.

En el anterior modelo informático de aplicaciones, la carga de procesamiento de la aplicación se comparte entre el código del servidor y el código instalado en cada cliente localmente. En otras palabras, la aplicación tiene su propio programa cliente recopilando que sirve como interfaz de usuario y debe instalarse en la computadora personal de cada usuario por separado. Las actualizaciones de código del lado del servidor de aplicaciones generalmente también requieren que se instalen actualizaciones de código del lado del cliente en la estación de trabajo de cada usuario, lo que aumenta los costos de soporte y reduce la productividad. Además, los componentes de cliente y servidor de una aplicación anteriormente estaban estrechamente relacionados con arquitecturas informáticas y sistemas operativos específicos, y transferirlos a otros componentes suele ser muy costoso para todas las aplicaciones, excepto para las más grandes.

Por el contrario, las aplicaciones web utilizan documentos web escritos en formatos estándar como HTML y JavaScript, que son compatibles con varios navegadores web. Puede pensar en una aplicación web como una variante específica del software cliente-servidor, donde se utiliza un protocolo estándar (HTTP) para descargar el software cliente en la computadora cliente al acceder a la página web correspondiente. Cada vez que visite la página web, se actualizará el software de red del cliente. Durante la sesión, el navegador web interpretará y mostrará la página y actuará como un cliente universal para cualquier aplicación web.

La aplicación a su vez tiene un proceso de diseño y desarrollo que hacen uso de distintas tecnologías que ayudan al correcto funcionamiento, pero estas se dividen en dos campos muy importantes, como lo son:

Backend

El backend es conocido como el área de lógica del negocio de la página web, es la parte interna de la página web, no es visible para el usuario y no tiene interfaz gráfica. Es la parte encargada de todas las funciones y tareas complejas que requieren capacidad de procesamiento. Se encarga de tareas que no son percibidas por el usuario, por ejemplo, conexión a bases de datos, autenticación, entre otros, para desarrollar backend existen distintos lenguajes de programación como lo son: Java, C, Go, Python, PhP, Node Js (JavaScript), entre otros.

Frontend

Frontend es todo el entorno visual, son todas las interfaces e interacciones entre la aplicación y el usuario, es el encargado de brindar una buena la experiencia del usuario, a través del diseño que se implemente con distintos lenguajes de programación y tecnologías.

1.5.1.1.2. BI

Business Intelligence (BI) es un tipo de software que se alimenta de datos de negocios y presenta reportes, paneles, tablas y gráficos de forma amigable para el usuario. Las herramientas de BI permiten a los usuarios de negocio acceder a diferentes tipos de datos: históricos y actuales, de terceros e internos, así como datos semiestructurados y datos no estructurados como las redes sociales. Los usuarios pueden analizar esta información para obtener insights sobre el rendimiento del negocio.

Las plataformas de BI tradicionalmente se basan en los almacenes de datos para obtener su información básica. Un almacén de datos agrega información de múltiples fuentes a un sistema central para respaldar el análisis y la generación de informes comerciales. El software de BI realiza búsquedas en los almacenes y presenta los resultados al usuario en forma de informes, tablas y mapas.

Los almacenes de datos pueden incluir un motor de procesamiento analítico en línea (OLAP) para admitir consultas multidimensionales. Por ejemplo: ¿Cuáles fueron las ventas de nuestra región este y la región oeste en este año, en comparación con el año pasado?

"OLAP proporciona una poderosa tecnología para el descubrimiento de datos, lo que facilita la inteligencia empresarial, los cálculos analíticos complejos y el análisis predictivo", dice el gerente de ofertas de IBM, Doug Dailey, en su blog de almacenamiento de datos. "Uno de los principales beneficios de OLAP es la coherencia de la información y los cálculos que utiliza para impulsar los datos a fin de mejorar la calidad del producto, las interacciones con los clientes y las mejoras de los procesos".

Algunas soluciones de inteligencia empresarial más nuevas pueden extraer e ingerir datos sin procesar directamente utilizando tecnología como Hadoop, pero los almacenes de datos siguen siendo la fuente de datos de elección en muchos casos. (IBM, 2024)

1.5.1.1.3. Big data analytics

El análisis de big data es el uso de técnicas analíticas avanzadas contra conjuntos de big data muy grandes y diversos que incluyen datos estructurados, semiestructurados y no estructurados, de diferentes fuentes y en diferentes tamaños, desde terabytes hasta zettabytes.

Puede definirse como conjuntos de datos cuyo tamaño o tipo está más allá de la capacidad de las bases de datos relacionales tradicionales para capturar, gestionar y procesar datos con baja latencia. Las características de big data incluyen alto volumen, alta velocidad y gran variedad. Las fuentes de datos se están volviendo más complejas que las de los datos tradicionales porque están impulsadas por la inteligencia artificial (IA) , los dispositivos móviles, las redes sociales y el Internet de las cosas (IoT). Por ejemplo, los diferentes tipos de datos se originan a partir de sensores, dispositivos, video/audio, redes, archivos de registro, aplicaciones transaccionales, web y redes sociales; muchos de ellos se generan en tiempo real y a gran escala. (IBM, IBM, 2024)

1.5.1.1.4. CRM (Customer Relationship Management)

Las siglas CRM significa Customer Relationship Management, lo que traducido al español es algo como Gestión de Relaciones con el Cliente. Podríamos decir que el término se refiere a una serie de estrategias comerciales, prácticas y de soluciones tecnológicas que están enfocadas a la relación con los clientes de una empresa.

Una empresa puede tener una única base de datos con la información de los clientes. Y luego, el CRM mantendrá un registro de todas tus interacciones con cada uno de ellos, y así tener centralizada una visión de todos los clientes y sus estadísticas en una única herramienta. (Fernández, 2023)

1.5.1.1.5. Dashboards

Un cuadro de mando o dashboard suele ser una hoja que muestra los datos que tiene una empresa en su base de datos. En la mayoría de los casos un cuadro de mando, se diseña para ser interactivo para que las personas puedan investigar los datos de manera self service, o autoservicio. Los tres programas de BI más comunes para crear cuadros de mando son Power BI, QlikSense, Pentaho y Tableau. (Datamedia, 2024)

1.5.1.1.6. Data

Dato en informática es la expresión general que describe aquellas características de la entidad sobre la que opera.

Los programas y aplicaciones tienen como función el procesamiento de datos, ya que cada lenguaje de programación tiene un conjunto de datos a partir de los cuales trabaja. Toda la información que entra y sale de un ordenador lo hace en forma de datos.

Dentro de los archivos existen datos que son paquetes más pequeños de otros datos llamados registros (reunidos por características iguales o similares). (concepto, s.f.)

Existen diferentes tipos de datos como: numérico, texto y lógico.

1.5.1.1.7. Data No Estructurada

Los datos no estructurados se refieren a datos que no están organizados en una estructura o un modelo diseñados. Los datos no estructurados suelen estar categorizados como cualitativos y podrían estar generados por máquinas o humanos. Los datos no estructurados son el tipo de datos más abundante disponible y, una vez analizados, pueden usarse para guiar decisiones comerciales y lograr objetivos comerciales entre muchos otros casos de uso.

Los datos no estructurados suelen estar almacenados en su formato nativo. Esto contribuye al desafío de convertir estos datos en información procesable. Si bien los datos no estructurados pueden ser más desafiantes que los datos estructurados, con frecuencia suelen contener información más detallada y completa que no está disponible en los datos estructurados. Como resultado, muchas organizaciones están invirtiendo en tecnologías como machine learning (ML) y procesamiento de lenguaje natural (NLP) para analizar mejor y obtener información a partir de los datos no estructurados. (Heeg, s.f.)

1.5.1.1.8. Data Scientist

El principal objetivo de un Data Scientist es analizar e interpretar datos y extraer conocimiento a partir de los mismos para que la empresa haga uso de esa información y la utilice en su beneficio, tanto para la resolución de problemas como para generar nuevas oportunidades de negocio.

Esta «ciencia de los datos», nacida del método científico, es la evolución de lo que hasta ahora se conocía como Analista de datos, pero a diferencia de éste que sólo se dedicaba a analizar fuentes de datos de una única fuente, el Data Scientist debe explorar y analizar datos de múltiples fuentes, a menudo inmensas (conocidas como Big Data), y que pueden tener formatos muy diferentes. Además, debe tener una fuerte visión de negocio para ser

capaz de extraer y transmitir recomendaciones a los responsables de negocio de su empresa.

Estos conjuntos de datos pueden provenir de los datos generados por todo tipo de dispositivos electrónicos (como un móvil, todo tipo de sensores, secuenciadores de genoma, …), redes sociales, datos médicos, páginas web, … y afectan de manera muy significativa la investigación actual en muchos campos como las ciencias biológicas, la informática médica, la salud, las ciencias sociales, por citar sólo algunos. (Obiols, s.f.)

1.5.1.1.9. DataWarehouse

Un Data Warehouse es un almacén electrónico donde generalmente una empresa u organización mantiene una gran cantidad de información. Los datos de un data warehouse deben almacenarse de forma segura, fiable, fácil de recuperar y fácil de administrar.

Un data warehouse es un repositorio unificado para todos los datos que recogen los diversos sistemas de una empresa. El repositorio puede ser físico o lógico y hace hincapié en la captura de datos de diversas fuentes sobre todo para fines analíticos y de acceso.

Data Warehouse es una arquitectura de almacenamiento de datos que permite a los ejecutivos de negocios organizar, comprender y utilizar sus datos para tomar decisiones estratégicas. Un data warehouse es una arquitectura conocida ya en muchas empresas modernas. (PowerData, 2024)

1.5.1.1.10. ETL (Extract, Transform, Load)

ETL: Fase de extracción

La fase de extracción de un proceso ETL consiste en captar datos de varias fuentes. Estas fuentes pueden ser internas como un CRM, ERP o ficheros drive que tenemos almacenados en la nube, pero también suelen existir fuentes de datos externas de gran valor como Web Services de otros colaboradores, proveedores o clientes, ficheros de bases de datos abiertas, e incluso datos extraídos de la web como redes sociales u otras páginas.

Estos datos extraídos son guardados en su forma original en almacenes de datos, normalmente en sistemas cloud, para luego poder ser tratados. (IEBS, 2024)

ETL: Fase de transformación

Esta es la fase esencial del proceso. La fase de transformación consiste en procesar los datos de forma que sean coherentes con el modelo de negocio de la organización. En la fase de extracción, los datos capturados pueden clasificarse como datos estructurados o no estructurados, y todos deben transformarse para obtener información de ellos.

Un dato estructurado será el más fácil de transformar, ya que se encuentra en bases de datos relacionales como tipos de texto, numéricos, etc. Pero los datos no estructurados son mucho más complejos de transformar puesto que no poseen una estructura interna determinada. Nos referimos a datos de tipo PDF, mensajes de texto, vídeos, imágenes, emails, etc. (IEBS, 2024)

ETL: Fase de carga

Por último, la fase de carga de un proceso ETL consiste en almacenar los datos ya transformados en un sistema destino del que se puedan nutrir todas las áreas de la organización. Estos sistemas de almacenamiento reciben el nombre de Data Warehouse y son el origen de datos para distintas herramientas de analítica descriptiva, diagnóstica, predictiva y prescriptiva. (IEBS, 2024)

1.5.1.1.11. Hadoop

Apache Hadoop es un marco de código abierto que se utiliza para almacenar y procesar de manera eficiente conjuntos de datos grandes cuyo tamaño varía entre los gigabytes y los petabytes de datos. En lugar de utilizar una sola computadora grande para procesar y almacenar los datos, Hadoop facilita la creación de clústeres de varias computadoras para analizar conjuntos de datos masivos en paralelo y con mayor rapidez. (AWS, s.f.)

1.5.1.1.12. NoSQL

El término NoSQL, que significa "no solo SQL", hace referencia a tipos de bases de datos no relacionales. Las bases de datos NoSQL usan un formato no tabular para almacenar datos en lugar de tablas relacionales basadas en reglas, como las bases de datos relacionales. Las bases de datos NoSQL utilizan un modelo de esquema flexible que admite una amplia variedad de datos no estructurados, como documentos, pares clave-valor, columnas amplias, gráficos y mucho más. Por lo general, las organizaciones eligen bases de datos NoSQL para sus grandes cantidades de datos que no se ajustan a un modelo relacional, y tienen un esquema dinámico que ofrece más flexibilidad, escalabilidad horizontal y facilidad de desarrollo.

NoSQL funciona más rápido porque las consultas no tienen que acceder a varias tablas para entregar una respuesta. En lugar de usar filas y columnas conocidas en tablas relacionales, las bases de datos NoSQL almacenan los datos en un formato tabular, como documentos JSON. (Cloud, s.f.)

1.5.1.1.13. SQL

El lenguaje de consulta estructurada (SQL) es un lenguaje de programación para almacenar y procesar información en una base de datos relacional. Una base de datos relacional almacena información en forma de tabla, con filas y columnas que representan diferentes atributos de datos y las diversas relaciones entre los valores de datos. Puede usar las instrucciones SQL para almacenar, actualizar, eliminar, buscar y recuperar información de la base de datos. También puede usar SQL para mantener y optimizar el rendimiento de la base de datos. (AWS, AWS, 2024)

1.5.2. Marco Metodológico

1.5.2.1. CRISP-DM (Cross-Industry Standard Process for Data Mining) Metodología de datos

La metodología CRISP-DM, o “Cross-Industry Standard Process for Data Minind” fue concebido a finales de 1996 por un consorcio de tres empresas, líderes de ciencia de datos, Daimler-Benz (posteriormente DaimlerChrysler), ISL (posteriormente SPSS) y NCR, y financiado por la comisión Europea. Tras sucesivas reuniones con otros profesionales del mismo campo, sesiones de trabajo y diversas implantaciones piloto, el proyecto concluyó en 1999 con un primer borrador del modelo de referencia, que posteriormente fue revisado hasta elaborar su primera versión completa (CRISP-DM 1.0) publicado en 2000. (Niño, 2020)

Esta metodología CRISP-DM integra todas las tareas necesarias en los proyectos de minería de datos, desde la fase de comprensión del problema hasta la puesta en producción de sistemas automatizados analíticos, predictivos y/o prospectivos. Como metodología incluye descripciones de las fases normales de un proyecto, las tareas necesarias en cada fase y una explicación de las relaciones entre las tareas. Como modelo de proceso ofrece un resumen del ciclo vital de minería de datos.

El ciclo vital del modelo contiene seis fases con flechas que indican las dependencias más importantes y frecuentes entre fases. La secuencia de las fases no es estricta. En la aplicación la mayoría de los proyectos avanzan y retroceden entre fases si es necesario. (IBM, IBM, 2021)

Pueden encontrarse interacciones que permiten mejorar la aproximación obtenida en otras fases anteriores. Las fases son las siguientes: (Gil, 2021)

- Comprensión del negocio
- Comprensión de los datos
- Preparación de los datos
- Modelado.
- Evaluación.
- Despliegue.



Figura 1. Etapas CRISP—DM Fuente: https://www.adictosaltrabajo.com/2021/01/14/metodologia-crisp-dm/

1.5.2.1.1 CRISP-DM, Comprensión del negocio

Esta es la primera fase por donde debe comenzar todo proyecto de minería de datos, comprendiendo en profundidad el problema que se quiere resolver y estableciendo los requisitos y objetivos del proyecto desde una perspectiva empresarial para luego trasladarlos a objetivos técnicos y a un plan de proyecto.

En primer lugar, se establece cuáles serán los criterios para medir el éxito en el proyecto, ya sean de tipo cualitativo o cuantitativo. Después, se realiza una evaluación de la situación actual determinando los antecedentes y requisitos del problema, tanto en términos de negocio como de minería de datos. Y por último, se realiza un plan de proyecto donde se tiene en cuenta qué pasos se deben seguir y qué procedimientos se emplearán en cada uno de ellos.

1.5.2.1.2 CRISP-DM, Comprensión de los datos

Es esta fase se lleva a cabo la recolección y exploración inicial de los datos, con el objetivo de establecer un primer contacto con el problema. Esta fase suele ser crítica en el proyecto dado que un mal entendimiento de los datos tiene como consecuencia un aumento en el tiempo global del proyecto y también reduce las garantías de éxito.

Para llevar a cabo esta fase hay que desarrollar una serie de tareas. La primera es recolectar datos iniciales y adaptarlos a las necesidades del proyecto para su posterior procesamiento. Luego se deben describir formalmente los datos obtenidos: número de instancias (filas) y atributos (columnas), el significado de los atributos y una descripción rigurosa del formato de los datos. Después se exploran los datos aplicando técnicas básicas de estadística descriptiva que revelan propiedades de estos. Por último, se lleva a cabo una verificación de los datos para determinar su consistencia, la cantidad y distribución de los valores nulos o valores fuera de rango que puedan provocar ruido en el modelado posterior.

1.5.2.1.3 CRISP-DM, Preparación de los datos

La fase de preparación trata de seleccionar, limpiar y generar conjuntos de datos correctos, organizados y preparados para la fase de modelado. Esta es una fase sumamente crítica en un proyecto de minería de datos. Los errores en los datos que se pasan por alto y que no son resueltos en esta fase se trasladan hasta la fase de modelado, lo que genera una reducción en la exactitud de los modelos o incluso, es posible entregar al cliente resultados basados en datos que aún contienen errores no detectados.

Por esta razón, esta fase es crucial y generalmente demanda siempre el mayor esfuerzo y tiempo del proyecto, aproximadamente un 75% del tiempo total.

1.5.2.1.4 CRISP-DM, Modelado

En esta fase se lleva a cabo la creación de modelos de conocimiento a partir de los datos suministrados desde la fase anterior. Estos modelos de conocimiento pueden ser de distintos tipos, por ejemplo, se pueden crear modelos de clasificación o regresión con el objetivo de estimar o inferir el valor de una determinada variable.

Para afrontar esta fase hay que seguir una serie de pautas que nos ayudarán a obtener mejores resultados. Aunque parezca obvio, el primer paso es seleccionar los algoritmos de modelado más apropiados al problema. Posteriormente se genera un plan de prueba, donde configuramos los valores de los parámetros que se usarán para los algoritmos de

aprendizaje automático (machine learning), ya que muchos de estos pueden ser configurados para determinar las características del modelo que se generará. También, se determinan las métricas de evaluación que se calcularán para evaluar la bondad de los modelos. Luego construimos los modelos, ejecutando los algoritmos seleccionados sobre los datos preparados con el fin de generar uno o más modelos y calculando las métricas. Y acabamos evaluando los resultados, donde se analizan las métricas de evaluación obtenidas con el fin de conocer la bondad de los modelos generados y garantizar que cumplan con los criterios de éxito definidos al inicio del proyecto.

1.5.2.1.5 CRISP-DM, Evaluación

Si los modelos obtenidos cumplen con las expectativas de negocio, se procede a la explotación del modelo. Si no, se evalúa en esta fase si se procede a iterar nuevamente sobre los pasos anteriores con el objetivo de encontrar nuevos resultados.

Para ello, se realiza una evaluación formal de los resultados obtenidos en las fases anteriores del proyecto, teniendo en cuenta los criterios de éxito de negocio y explicando las causas que provocaron los grados de éxitos alcanzados y revisando todo el proceso llevado a cabo con el fin de encontrar errores que puedan afectar al éxito.

1.5.2.1.6 CRISP-DM, Despliegue

En esta última fase es donde se definen las estrategias para su implementación, monitorización y mantenimiento de los modelos. Esto nos ayudará a observar cualquier comportamiento irregular del sistema y corregirlo de forma que no provoque deficiencias en el servicio del cliente.

1.5.2.1 Metodología Kanban

La metodología Kanban representa las tareas pendientes del proyecto usando elementos visuales como tableros. Los equipos ágiles usan este enfoque para visualizar mejor los flujos de trabajo y el progreso de los proyectos. Por lo general, este método se aplica en una herramienta de software que te permite cambiar y arrastrar tableros dentro de los proyectos. El concepto más importante a tener en cuenta es que el objetivo principal del marco Kanban es centrarse en las tareas más importantes del proyecto, manteniendo una estructura simple. Kanban significa firmar o etiquetar en japonés. Este método fue creado por el ingeniero Taiichi Õno para la empresa automovilística japonesa Toyota, que lo utiliza desde la década de 1950. El método Kanban fue recogido más tarde por los desarrolladores de software informático en la década de 2000 y luego adoptado por empresas de todos los sectores. (Asana, 2024)

La metodología Kanban se basa en una serie de principios básicos con el fin de obtener el máximo rendimiento de su flujo de trabajo.

Este método permite visualizar el proceso de realización de una tarea. Toma la forma de una tabla con tres o incluso cuatro columnas:

1.5.3. Marco legal

La normatividad de acceso a la información debe considerarse para analizar datos. A continuación, se describen algunas de estas leyes en materia de acceso:

- Constitución Política de Colombia 1991. Artículo 20 y 74, se consagra el acceso a la información como derecho fundamental.
- Ley 1581 de 17 de octubre de 2012. Por el cual se dictan disposiciones generales para la Protección de Datos Personales.
- Ley 1712 de 06 de Marzo de 2014. Por medio de la cual se crea la Ley de Transparencia y del Derecho de Acceso a la Información Pública Nacional y se dictan otras disposiciones.
- Decreto 103 de 20 de enero de 2015. Por el cual se reglamenta parcialmente la Ley 1712 de 2014 y se dictan otras disposiciones.
- Decreto 1078 de 26 de mayo de 2015. Por medio del cual se expide el Decreto Único Reglamentario del Sector de Tecnologías de la Información y las Comunicaciones.
- Decreto 415 de 07 de marzo de 2016. Por el cual se adiciona el Decreto Único regulador del sector de la Función Pública, Decreto 1083 de 2015, relativo a la definición de los lineamientos para el fortalecimiento institucional en Tecnologías de la Información y las Comunicaciones.
- Conpes 3920 de 17 de abril de 2018. Política Nacional de Explotación de Datos (Big Data).
- Decreto 1008 de 14 de junio de 2018. Por el cual se establecen los lineamientos generales de la Política de Gobierno Digital y se subroga el capítulo 1 del título 9 de la parte 2 del libro 2 del Decreto 1078 de 2015, Decreto Único Reglamentario del sector Tecnologías de la Información y las Comunicaciones.

1.6. Metodología de la investigación

1.6.1 Tipo de estudio

El proyecto de investigación está orientado al tipo de estudio descriptivo, el cual plantea mediante el diseño de un aplicativo web que permita la visualización de los diferentes

indicadores del sector agropecuario obtenidos mediante el uso e implementación de la Analítica de Big Data, la cual involucra grandes volúmenes de información de diferentes fuentes de datos, brindando la oportunidad de crear nuevos modelos, predicción de resultados y toma de decisiones aumentando la fiabilidad de las herramientas empleadas en el manejo masivo de información.

1.6.2 Método de investigación

El método de la presente investigación será deductivo ya que parte de una premisa general y se aplica la lógica para llegar a una conclusión específica. Mediante la aplicación de teorías frecuentes para el análisis de Big Data se llegará a conclusiones y presis generales de la información tratada para posteriormente su visualización.

1.6.3. Fuentes y técnicas para la recolección de la información

Para la presente investigación la fuente y técnica empleada serán revisión de registros, debido a que se realizará la recolección de grandes volúmenes de datos del sector agropecuario a partir de diferentes fuentes de información.

1.6.4. Tratamiento de la información

El tratamiento de la información y su posterior visualización la realizaremos a través de herramientas que ayuden a entender las diferentes etapas del análisis de Big Data a la información, como son recopilación de datos, procesamiento, almacenamiento y visualización, empleando lenguajes especializados como Javascript, Angular y Power BI. Una vez los datos sean tratados, almacenados y visualizados en un dashboard, construiremos un aplicativo web con su correspondiente diseño y desarrollo en lenguaje Javascript, Angular, para su publicación con muchos usuarios. Donde se enseñarán datos integrados en el sector agrícola para la toma de decisiones.

1.7. Organización del trabajo de grado

La presente investigación se encuentra estructurada de la siguiente forma: en el primer capítulo se desarrolla la descripción y contextualización de la investigación, en el segundo capítulo se explica la estructura del análisis de la información y visualización, en el tercer capítulo se encuentran los resultados de la investigación.

1.8. Estudio de sistemas previos

Se realizó una búsqueda de diferentes proyectos, tesis y trabajos de investigación relacionados con el tema de estudio, en los cuales se pueden ver la directriz y estructura de orientación del tema seleccionado.

A continuación, se listan algunos ejemplos de trabajos realizados por diferentes autores:

- Mérida Fonseca, César Manuel, Ríos Alvarado, Richer Paúl, (2024), **Propuesta de la plataforma de big data orientado al sector turístico.** Tipo de investigación utilizada metodología de investigación formulada por Hernández Sampieri. Población las pequeñas y medianas empresas (PYMEs) y Muestra una empresa del sector turístico. Técnicas e instrumentos de recolección de datos utilizados Big data analytics, data analytics, data minning, etc.

  El objetivo general de la investigación es proponer una plataforma de big data para la gestión de la información en tiempo real orientado al sector turístico. Objetivos específicos: a) Recopilar y analizar información de la situación actual de big data. b) Analizar y realizar una comparación de las tres plataformas de big data con mayor presencia en el mercado actual. c) Proponer las fases y niveles para la gestión de big data en el sector turístico. d) Diseñar la plataforma de big data para el sector turístico.

  Las principales conclusiones a las cuales llegaron fueron: a) Se propuso una plataforma de big data orientada al sector turístico para cubrir las deficiencias que presentan plataformas de big data como Oracle e IBM, en cubrir aspectos importantes en un sector que está creciendo a un sector que está creciendo a una velocidad muy rápida debido al consumo masivo de nuevas tecnologías móviles. b)La investigación realizada, en la etapa inicial de diseño, tiene como finalidad sentar las bases para una futura implementación de la plataforma, considerando los requisitos mínimos obtenidos en las fichas técnicas por cada componente para un desarrollo exitoso. c) Es importante resaltar que los nuevos emprendedores en adoptar big data en sus procesos pueden llegar a formar parte fundamental del proceso de maderación de big data. Debido a que, con las nuevas competencias en el mercado, los frameworks y plataformas propuestas pueden ser gran incentivador para que otras empresas del sector tomen la importancia debida a big data.

  Este proyecto se relaciona con la investigación porque se enfoca en la utilización de grandes volúmenes de datos que se encuentran en las redes para facilitar la toma de decisiones en un sector. También en la creación de una plataforma Big Data personalizada y en el uso de nuevas tecnologías. De esta investigación se tomó los aspectos teóricos tecnológicos, los cuales aportaron en la claridad del tratamiento de grandes volúmenes de datos y un punto de partida de las metodologías de procesamiento. También la forma de abordar un problema de esta rama.

- Ferro Rugeles Rubén Darío, Fernández Moncada Nicolás, Cossio Escobar Gonzalo, (2023), **Pronóstico de ventas a través de una aplicación web aplicando machine learning y variables exógenas**, tipo de investigación exploratoria,

Población las pequeñas y medianas empresas (PYMEs) y Muestra una empresa del sector tecnológico. La técnica e instrumento de recolección de datos utilizados.

Objetivo general, desarrollar una solución de analítica predictiva de ventas (en Euros) de una empresa del sector tecnológico que, mediante la utilización de técnicas de machine learning y la consideración de variables exógenas cuantitativas, permiten estimar las ventas de la compañía a corto, mediano y largo plazo. Objetivos específicos: a) Elaborar un modelo de web scrapping que capture la variable macroeconómica más importantes validadas por expertos de la compañía. b) Realizar un análisis exploratorio de datos (EDA) para identificar y caracterizar la evolución de las ventas y las variables exógenas asociadas con la solución. c) Seleccionar modelos de machine learning de series temporales que permita predecir el comportamiento de las ventas teniendo en cuenta múltiples variables internas de la compañía y externas del mercado. d) Desarrollar un código escalable que permita la estimación de ventas de otra compañía a corto, mediano y largo plazo. e) Desarrollar una herramienta de visualización en una aplicación web que permita presentar el histórico y la estimación de ventas de la compañía a corto, mediano y largo plazo.

Las principales conclusiones del autor son: 1) Con respecto al desarrollo de la aplicación web para presentar la predicción de ventas, representa una herramienta significativa en el ámbito de la analítica de datos y la toma de decisiones comerciales. Debido a su enfoque en la propiedad intelectual y la capacidad de adaptación a las necesidades específicas de cada cliente, se ha logrado evitar los costos asociados con las licencias de tableros de control preexistentes. 2) La experiencia del usuario se ha basado en la implementación de una página de inicio de sesión que garantiza un acceso personalizado a la aplicación. La posibilidad de agregar nuevas funcionalidades a futuro ofrece una visión de la escalabilidad de la aplicación y las características futuras que se pueden agregar, manteniéndola en constante evolución. 3) Con respecto a la funcionalidad de visualización de la predicción de ventas a través del tablero de control incluido en la aplicación, permite a los usuarios monitorear indicadores clave y examinar series temporales de ventas en pestañas que abarcan desde el corto hasta el largo plazo, lo cual proporciona una visión profunda de las ventas pasadas y las proyecciones futuras, lo que se traduce en una herramienta valiosa para respaldar la toma de decisiones basada en datos en el contexto empresarial. 4) La aplicación web representa una herramienta estratégica valiosa, ya que se adapta a los cambios comerciales y asegura la no dependencia de terceros.

Este proyecto se relaciona con la investigación porque parte desde el tratamiento de la información por medio de la analítica de datos hasta la visualización de la información por medio de una aplicación web, donde se pueden representar los valores, aplicar filtros, representación rápida y sencilla. De la investigación se tomó aspectos como la aplicación de la metodología Scrum en el desarrollo de todo el trabajo mencionado. Esta es una metodología ágil muy frecuentemente usada que se enfoca en el trabajo en equipo, la colaboración y la entrega de resultados iterativos y continuos, para producir un producto final de la mejor calidad. Esta

metodología se va realizando entregas funcionales del producto por partes sin esperar al final, viendo resultados constantemente y teniendo aprobación del producto en cada entrega.

- Quilindo Celis Luz Amanda y Vega Guarín Juan Sebastián, (2021), **Modelo de Arquitectura de software para el procesamiento de datos en arquitecturas actuales**. Tipo de investigación utilizada es descriptiva. Población las pequeñas y medianas empresas (PYMEs) y muestra grupo financiero que ofrece servicios de tecnologías de la información y canales electrónicos. Técnicas e instrumentos de recolección de datos utilizados fueron fuentes secundarias de información como libros, artículos y documentos que ayudan a comprender los diferentes patrones de software que pueden ser usados al implementar una solución arquitectural.

  El objetivo general de la investigación es definir un modelo de arquitectura de software enfocado en el procesamiento de datos a través de un prototipo, con el fin de entender los elementos y patrones necesarios para su diseño. Objetivos específicos: a) Definir cuáles son los atributos de calidad a tener en cuenta al momento de implementar una arquitectura de software frente al problema del procesamiento de datos. b) Analizar los problemas más comunes en el procesamiento de datos para identificar cuáles son los patrones arquitecturales orientados a solucionar dichos problemas. c) Evaluar el desempeño del modelo implementando mediante un prototipo basado en una herramienta existente, tomando en cuantos parámetros como la lógica de negocio y la infraestructura propuesta.

  Una de las principales conclusiones a las cuales llegó el trabajo fue que en las arquitecturas de procesamiento de datos hay varios sistemas que se deben entender y separar al momento de análisis, estos son: la ingesta de datos, la programación de las tareas, el procesamiento de los datos, el almacenamiento y la visualización de los mismos. La ingesta de datos es el componente encargado de leer datos de diferentes fuentes usando API's y examinando la estructura de los datos para determinar si la información es relevante para el negocio y sí es relevante para los intereses de cada compañía. El lenguaje piloto en la actualidad para el desarrollo de soluciones de esta índole es Python, pero también existen soluciones mucho más grandes en la nube desarrolladas por gigantes como Google, Azure, AWS, ect. Otra conclusión es la importancia de seleccionar el lugar de almacenamiento de los datos según las necesidades de la compañía, debido a la gran variedad de ofertas y el tipo o cantidad de información, por ejemplo en las bases de datos existe NoSQL, BigQuery, Graphos, etc.

  La investigación se relaciona con el proyecto por la implementación de modelos arquitectónicos, procesamiento de datos, patrones de diseño y técnicas de normalización, estandarización y analítica de datos para implementar en el prototipo. En la adaptación de un modelo de procesamiento de datos adaptado a una lógica de negocio particular. Los aspectos que se tomaron de la investigación son la Arquitectura para el Procesamiento de Datos por lotes, la cual tiene los siguientes componentes lógicos: Almacenamiento de datos, procesamiento por lote, análisis e informes y orquestación.

- Duarte Yara Maria paula, Salcedo Montenegro Joaquín Santiago, (2018), **Prototipo de software para la visualización de indicadores de datos abiertos.** Tipo de investigación utilizada explicativa. Población contratación en Colombia y muestra sector público. Técnicas e instrumentos de recolección de datos utilizados datos abiertos en Colombia.

  El objetivo general de la investigación es desarrollar un prototipo de software que administre la visualización de indicadores de gestión de contratos utilizando datos abiertos en Colombia. Objetivos específicos: a) Seleccionar los indicadores de gestión que apoyen a la toma de decisiones en los procesos contractuales. b) Seleccionar la técnica de visualización de datos que exprese las características de los indicadores de gestión. c) Implementar los indicadores de gestión a través de un prototipo de software. d) Realizar las pruebas funcionales al prototipo de software con el fin de asegurar su correcta funcionalidad.

  Principal conclusión es que, a través del proceso de inteligencia empresarial establecido por la extracción de datos, la transformación de datos y el cargue en la fuente de datos, se llegó a la realización de un prototipo de visualización dashboard, el cual permite la realización de diferentes estudios según el enfoque o el caso específico.

  Explicar por qué se relacionan las investigaciones. Luego debe describir y explicar qué aspectos tomó de la investigación que le sirvieron de ayuda para realizar en su trabajo.

  La investigación se relaciona con el proyecto porque su objetivo es desarrollar un prototipo de software que administre la visualización de indicadores de gestión de contratos utilizando datos abiertos en Colombia. Representando un fenómeno en la contratación pública en el tiempo, sus características y posibles comportamientos en el tiempo.

PARTE II.

DESARROLLO DE LA INVESTIGACIÓN

2.1 Aplicación de la Metodología CRISP-DM

2.1.1 Comprensión del negocio

Para el marco de desarrollo en este proyecto se enfocó en el sector agropecuario a nivel departamental y municipal de Colombia, para dar mayor contexto del sector que se está abordando se identifican los actores que están directamente interesados; estos actores son: a) El ministerio de agricultura y desarrollo rural entidad que tiene como objetivo primordial la formulación, coordinación y adopción de las políticas, planes, programas y proyectos del Sector Agropecuario, Pesquero y de Desarrollo Rural. b) DANE, entidad responsable de la planeación, levantamiento, procesamiento, análisis y difusión de las estadísticas oficiales de Colombia. Esta entidad tiene el objetivo de incentivar los análisis y la toma de decisiones con base en los datos. Por último los usuarios que deseen consultar y profundizar en temas del sector agrícola.

Esta información ayudará por ejemplo a ubicar los cultivos, el tipo, su permanencia, entre otras variables que por ejemplo ayudan en la determinación de su valor comercial. La información está obtenida de datos abiertos de entidades públicas como el DANE, las cuales obtiene esta información por encuestas y censos anuales.

2.1.1.1 Establecer los objetivos del negocio

2.1.1.1.1 Objetivos generales del negocio

Obtener un mejor entendimiento del comportamiento de los productos agrícolas de los diferentes departamentos y municipios del país utilizando los datos del DANE, Ministerio de agricultura, FINAGRO, banco agrario, participando desde el campo académico en la gestión productiva del país.

2.1.1.1.2 Objetivos específicos del negocio

- Implementar la metodología CRISP-DM para identificar indicadores y patrones en el sector agrícola.
- Caracterizar los tipos de cultivos por municipios y departamentos.
- Estimar el flujo de caja de los cultivos, ingresos y costos de los cultivos.
- Experimentar, validar y probar en correlaciones entre variables.

2.1.1.2 Evaluar el problema

2.1.1.2.1 Información disponible

Los datos que son objeto del estudio son originados por el DANE, FINAGRO, Ministerio de Agricultura, consultados en información abierta al público. Esta información está agrupada por Departamento, Municipio, tipo de cultivo, año de realización, entre otras variables. El formato de la información es en Excel y pdf.

2.1.1.2.2 Requisitos, supuestos y restricciones

- Requisitos del proyecto: Identificar los cultivos por municipio y departamento, ciclo del cultivo, grupo, estado, insumos, valor por kilo, entre otras. Que permita el entendimiento agrícola por región. Por otra parte, construir información para determinar la renta de cada cultivo.

- Describir los supuestos: El gran número de información agrícola sin relacional sirve para caracterizar y generar indicadores en el sector.

- Comprobar las restricciones: La información consultada y utilizada en el proyecto es de libre consulta.

2.1.1.2.3 Riesgos

- Poco tiempo para desarrollar el proyecto.
- Necesidad de replantear las variables o introducir nuevas.

2.1.1.3 Establecer los objetivos de la minería de datos

- Identificar las variables más relevantes para la generación de la información usando los datos abiertos.
- Implementar un sistema de visualización de resultados de la información que pueda ser utilizados por los usuarios.

2.1.2 Comprensión de los datos

El objetivo de esta fase es familiarizarse con la forma en que se encuentran almacenados los datos y sus características con el fin de establecer variables iniciales aproximadas y

evitar futuros problemas en la siguiente fase de preparación de los datos. Las variables son descriptivas para mejorar el entendimiento de la relación entre estas. Por último, se verifica la calidad de los datos para determinar la consistencia, representación de la realidad, campos innecesarios o vacíos y valores fuera de rango, para lograr la mayor completitud y corrección de los datos.

2.1.2.1 Recolectar datos iniciales

Los datos fueron consolidados y obtenidos de las encuestas del DANE sector agropecuario y FINAGRO; los cuales son de uso abierto y en los formatos tipo (xlsx) y (pdf). El formato (xlsx) contiene información como departamentos, municipios, cultivos por municipio y departamento, clases, insumos, precios, costos, entre otra información.

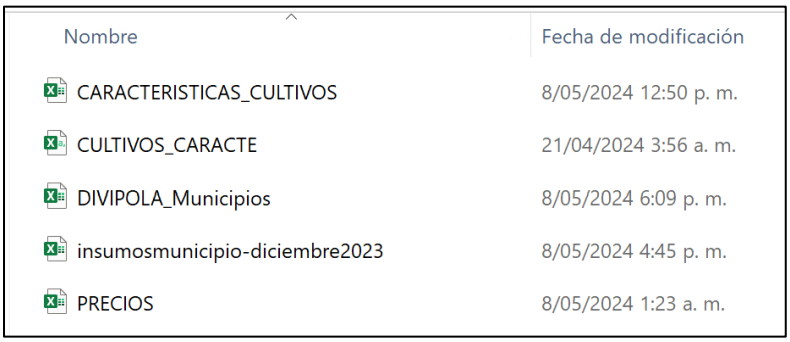


Figura 2. Información datos libres sector Agrícola. Fuente: Elaboración propia.

- Características_cultivos: Esta hoja de la data set contiene los datos de cultivos como zona, distancia de siembra, densidad, duración del ciclo, entre otras. La hoja contiene 22 variables y 100 datos.
- Cultivos_Caracte: Esta hoja de la data set contine los datos de otras variables de los cultivos como grupo, cultivo, estado físico y ciclo del cultivo. La hoja contiene 17 variables y 206.069 datos.
- Divipola_Municipios: Este archivo contiene dos hojas de la data set, una contiene los datos de los departamentos con su identificador, nombre, latitud y longitud. La otra hoja contiene los datos de los municipios con identificador, nombre, latitud y longitud. La primera hoja contiene 04 variables y 99 datos. La segunda hoja contiene 06 variables y 1.123 datos.
- Insumospormunicipio_diciembre2023: Esta hoja de la data set contine los datos de características de insumos usados en la producción agrícola por municipio y departamento. Con su respectiva presentación y precio promedio comercial. La hoja contiene 10 variables y 8.378 datos.
- Precios: Este archivo contiene varias hojas de la data set, cada hoja está dividida por tipo de cultivo, en las cuales está el departamento, municipio, temperatura, altitud, precipitación, tipo de suelo, distancia de siembra, densidad, duración del ciclo, inicio de producción, valor del jornal, precio de venta, productividad. Cada hoja contiene 20 variables y 100 datos.

2.1.2.2 Describir los datos

Tabla 1.Hoja del data set departamento

VARIABLE	TIPO	DESCRIPCIÓN
ID_DEPARTAMENTO	Varchar	Identificación del departamento.
NOMBRE_DEPARTAMENTO	Texto	Nombre del departamento.
LATITUD	Texto	Posición geográfica Latitud.
LONGITUD	Texto	Posición geográfica Longitud.

Fuente: Elaboración propia

Tabla 2. Hoja Data set Municipio

VARIABLE	TIPO	DESCRIPCIÓN
ID_DEPARTAMENTO	Varchar	Identificación del departamento.
NOMBRE_DEPARTAMENTO	Texto	Nombre del departamento.
ID_MUNICIPIO	Texto	Identificación del municipio.
NOMBRE_MUNICIPIO	Texto	Nombre del municipio.
LATITUD	Texto	Posición geográfica Latitud.
LONGITUD	Texto	Posición geográfica Longitud.

Fuente: Elaboración propia

Tabla 3. Hoja Data set Características cultivos

VARIABLE	TIPO	DESCRIPCIÓN
ID_DEPARTAMENTO	Varchar	Identificación del departamento.
NOMBRE_DEPARTAMENTO	Texto	Nombre del departamento.
ID_MUNICIPIO	Texto	Identificación del municipio.
NOMBRE_MUNICIPIO	Texto	Nombre del municipio.
GRUPO DE CULTIVO	Texto	Clasificación entre frutales, cereales, fibras, flores, hongos, hortalizas, leguminosas, oleaginosas, plantas aromáticas, tubérculos, otros.
SUBGRUPO DE CULTIVO	Texto	Nombre del cultivo
CULTIVO	Texto	Nombre del cultivo
DESAGREGACIÓN REGIONAL Y/O SISTEMA PRODUCTIVO	Texto	Descripción degradación y sistema de producción.
ÁREA SEMBRADA (HA)	Texto	Área en cada municipio sembrada.
ÁREA COSECHADA (HA)	Texto	Área en cada municipio cosechada.
ESTADO FISICO PRODUCCIÓN	Texto	Descripción del estado físico del cultivo
NOMBRE CIENTIFICO	Texto	Nombre científico del cultivo.
CICLO DE CULTIVO	Texto	Clasificación entre transitorio y permanente,

Fuente: Elaboración propia

Tabla 4. Hoja Data set Insumo

VARIABLE	TIPO	DESCRIPCIÓN
ID_DEPARTAMENTO	Varchar	Identificación del departamento.
NOMBRE_DEPARTAMENTO	Texto	Nombre del departamento.
ID_MUNICIPIO	Texto	Identificación del municipio.
NOMBRE_MUNICIPIO	Texto	Nombre del municipio.
CÓDIGO CPC 2.1 A.C.	Texto	Código en el mercado.
NOMBRE DEL PRODUCTO	Texto	Nombre del producto.
PRESENTACIÓN DEL PRODUCTO	Texto	Tipo de presentación del producto en el mercado.
PRECIO PROMEDIO DIC 2023	Texto	Precio en el mercado
VARIACIÓN PORCENTUAL	Texto	Variación porcentual
ID INSUMO	Texto	Identificador
NOMBRE INSUMO	Texto	Nombre del insumo

Fuente: Elaboración propia

Tabla 5. Hoja Data set Tipo de cultivo

VARIABLE	TIPO	DESCRIPCIÓN
ID_DEPARTAMENTO	Varchar	Identificación del departamento.
NOMBRE_DEPARTAMENTO	Texto	Nombre del departamento.
ID_MUNICIPIO	Texto	Identificación del municipio.
NOMBRE_MUNICIPIO	Texto	Nombre del municipio.
ACTUALIZACIÓN	Texto	Fecha del estudio.
TEMPERATURA MEDIA ANUAL	Texto	Temperatura promedio del municipio.
ALTITUD	Texto	Altitud promedio del municipio.
PRECIPITACIÓN	Texto	Precipitación promedio del municipio.
TIPO DE SUELO	Texto	Características generales del suelo.
MATERIAL PROPAGACIÓN	Texto	Tipo de material de propagación por árbol o plántula.
DISTANCIA DE SIEMBRA	Texto	Medida de distancia.
DENSIDAD	Texto	La densidad de siembra
DURACIÓN DEL CICLO	Texto	Tiempo de duración del cultivo comercialmente.
INICIO DE PRODUCCIÓN DESDE LA SIEMBRE	Texto	Edad de inicio de la producción.
VALOR JORNAL	Numérico	Precio del Jornal
PRECIO PONDERADO DE VENTA	Numérico	Precio promedio de venta del producto
PRODUCTIVIDAD	Numérico	Productividad.

VARIABLE	TIPO	DESCRIPCIÓN
PRECIO DE VENTA	Numérico	

Fuente: Elaboración propia

2.1.2.2 Explorar los datos y calidad de los datos.

En el numeral anterior se presenta una descripción de los datos, ahora se pasa a explorar mediante Excel los datos para tener una idea inicial de la información y su contenido, de sus características. Para la exploración inicial de los datos que se encuentran en los archivos en formato Excel, se revisará cuantas hojas tienen, para su selección y eliminación de información no necesaria. La mayoría de información es obtenida de datos abiertos del DANE, esta información no solo contiene del sector agrícola, sino también del sector pecuario por lo cual se elimina la del sector pecuario ya que el estudio no incorpora esta información. La información de FINAGRO está en formato pdf, por lo cual se transforma a Excel y se complementa los campos faltantes.

2.1.2 Preparación de los datos

2.1.3.1 Integrar los datos

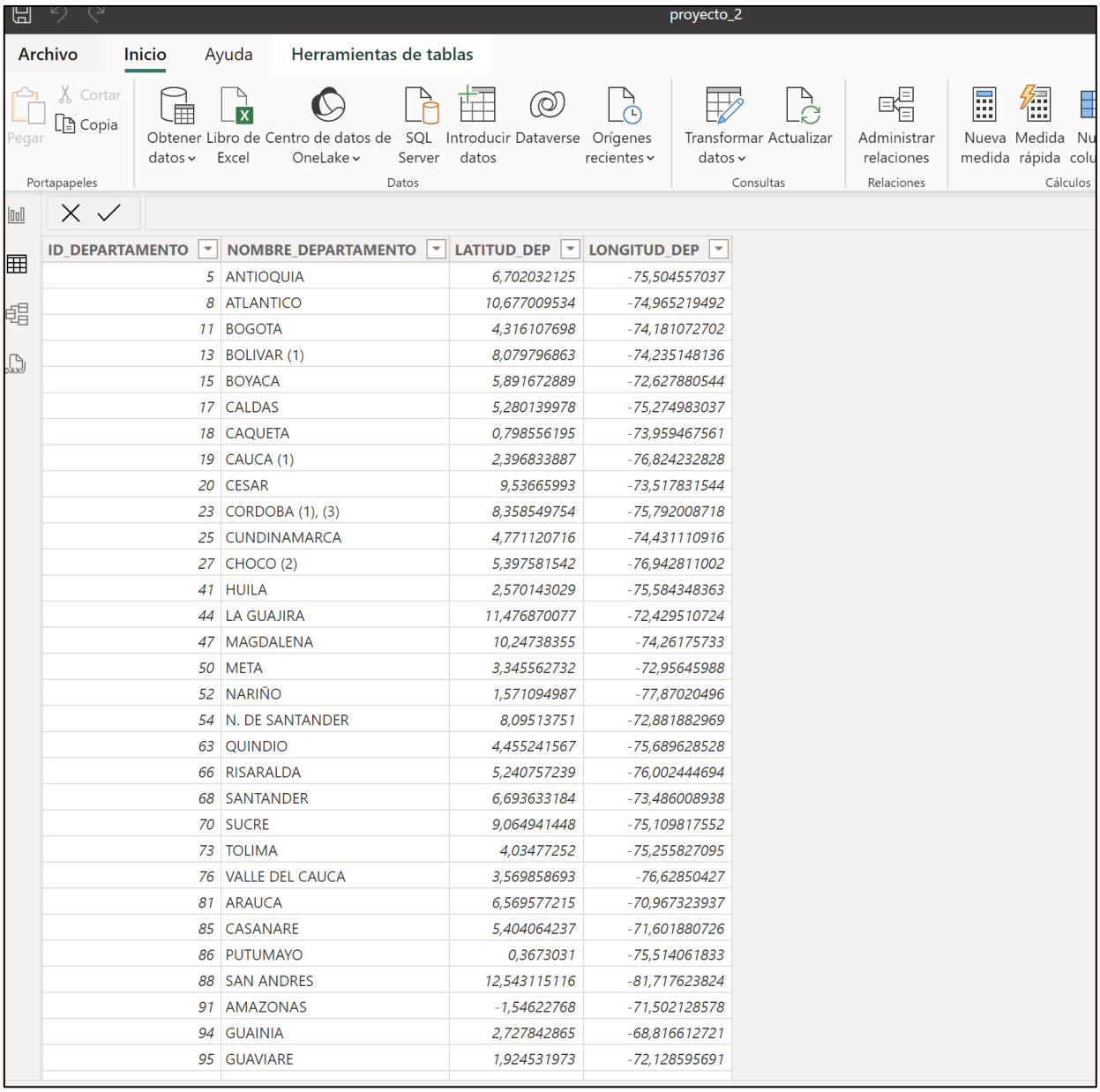
Es necesario disponer de todas las variables relacionadas previamente para la construcción de la información representativa, y se identifican otras variables producto de la transformación e integración de los datos, estos datos tienen como fin enriquecer la precisión del análisis.

2.1.3.2 Limpiar los datos y Construcción de los datos

Inicialmente se debe revisar la integridad de los datos, en primer lugar, como la mayoría está en el formato Excel, se utilizó la herramienta PowerBI para su depuración y revisión, dentro de su examen se consideró lo siguiente:

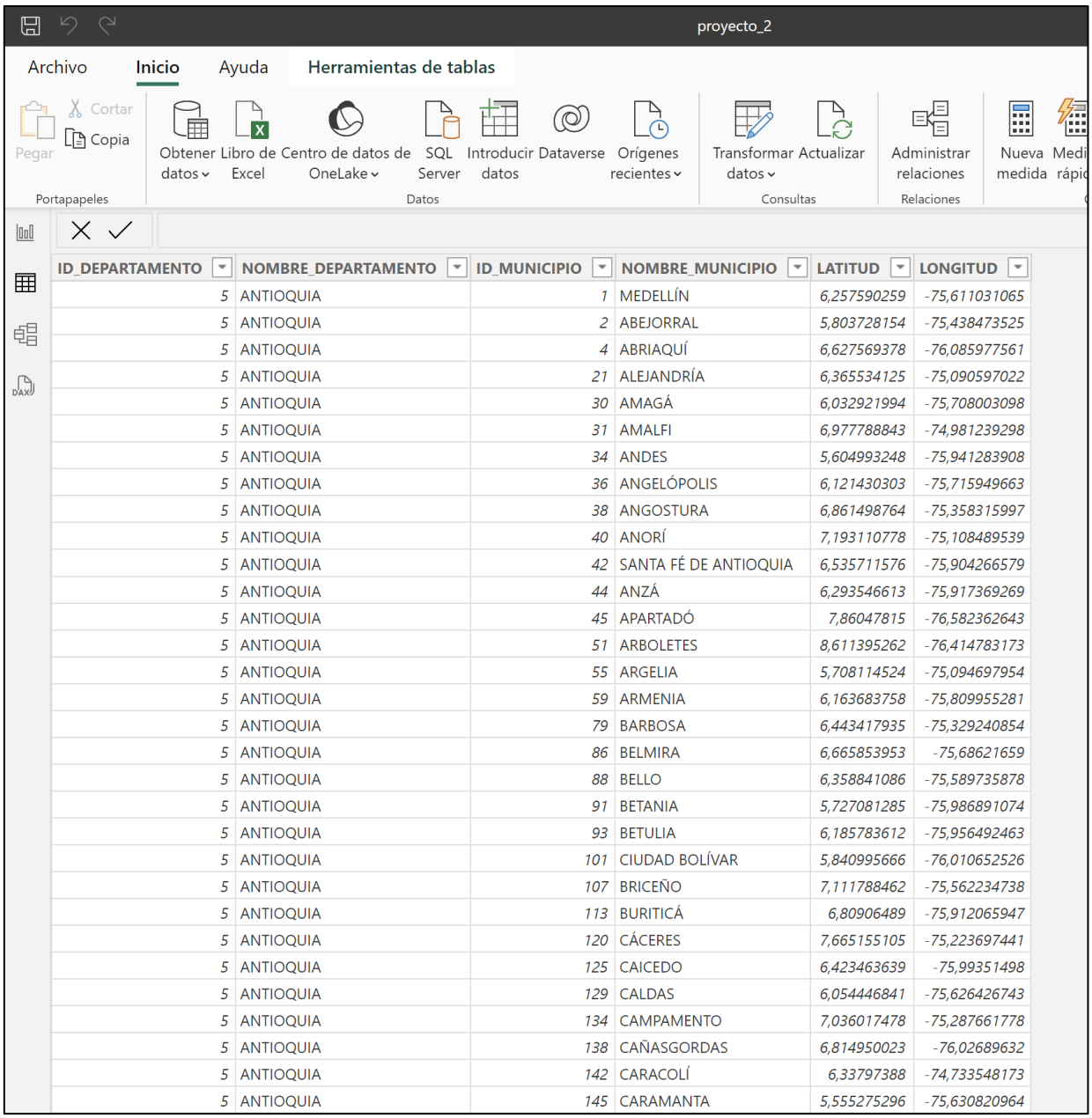
- Datos faltantes
- Numeración de variables
- Tipo de datos
- Relación entre las variables.

Se cargaron los datos inicialmente departamento y municipio, en PowerBI Desktop. Los cuales se editaron para que su variable identificadora no estuviera nula o repetida. Estas tablas tienen latitud y longitud para posteriormente poderlas relacionar con la información geográfica.



ID_DEPARTAMENTO	NOMBRE_DEPARTAMENTO	LATITUD_DEP	LONGITUD_DEP
5	ANTIOQUIA	6,702032125	-75,504557037
8	ATLANTICO	10,677009534	-74,965219492
11	BOGOTA	4,316107698	-74,181072702
13	BOLIVAR (1)	8,079796863	-74,235148136
15	BOYACA	5,891672889	-72,627880544
17	CALDAS	5,280139978	-75,274983037
18	CAQUETA	0,798556195	-73,959467561
19	CAUCA (1)	2,396833887	-76,824232828
20	CESAR	9,53665993	-73,517831544
23	CORDOBA (1), (3)	8,358549754	-75,792008718
25	CUNDINAMARCA	4,771120716	-74,431110916
27	CHOCO (2)	5,397581542	-76,942811002
41	HUILA	2,570143029	-75,584348363
44	LA GUAJIRA	11,476870077	-72,429510724
47	MAGDALENA	10,24738355	-74,26175733
50	META	3,345562732	-72,95645988
52	NARIÑO	1,571094987	-77,87020496
54	N. DE SANTANDER	8,09513751	-72,881882969
63	QUINDIO	4,455241567	-75,689628528
66	RISARALDA	5,240757239	-76,002444694
68	SANTANDER	6,693633184	-73,486008938
70	SUCRE	9,064941448	-75,109817552
73	TOLIMA	4,03477252	-75,255827095
76	VALLE DEL CAUCA	3,569858693	-76,62850427
81	ARAUCA	6,569577215	-70,967323937
85	CASANARE	5,404064237	-71,601880726
86	PUTUMAYO	0,3673031	-75,514061833
88	SAN ANDRES	12,543115116	-81,717623824
91	AMAZONAS	-1,54622768	-71,502128578
94	GUAINIA	2,727842865	-68,816612721
95	GUAVIARE	1,924531973	-72,128595691

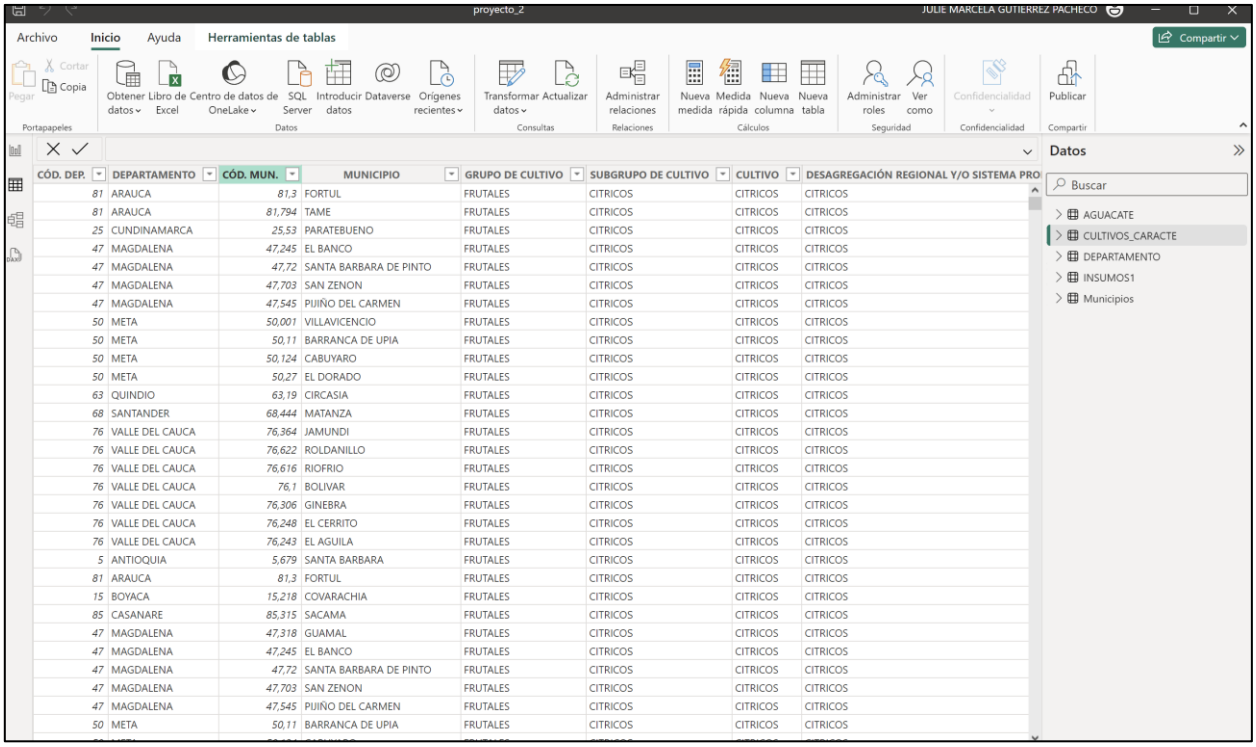
Figura 3. Cargue de información del departamento. (Elaboración propia)



ID_DEPARTAMENTO	NOMBRE_DEPARTAMENTO	ID_MUNICIPIO	NOMBRE_MUNICIPIO	LATITUD	LONGITUD
5	ANTIOQUIA	1	MEDELLÍN	6,257590259	-75,611031065
5	ANTIOQUIA	2	ABEJORRAL	5,803728154	-75,438473525
5	ANTIOQUIA	4	ABRIAQUÍ	6,627569378	-76,085977561
5	ANTIOQUIA	21	ALEJANDRÍA	6,365534125	-75,090597022
5	ANTIOQUIA	30	AMAGÁ	6,032921994	-75,708003098
5	ANTIOQUIA	31	AMALFI	6,977788843	-74,981239298
5	ANTIOQUIA	34	ANDES	5,604993248	-75,941283908
5	ANTIOQUIA	36	ANGELÓPOLIS	6,121430303	-75,715949663
5	ANTIOQUIA	38	ANGOSTURA	6,861498764	-75,358315997
5	ANTIOQUIA	40	ANORÍ	7,193110778	-75,108489539
5	ANTIOQUIA	42	SANTA FÉ DE ANTIOQUIA	6,535711576	-75,904266579
5	ANTIOQUIA	44	ANZÁ	6,293546613	-75,917369269
5	ANTIOQUIA	45	APARTADÓ	7,86047815	-76,582362643
5	ANTIOQUIA	51	ARBOLETES	8,611395262	-76,414783173
5	ANTIOQUIA	55	ARGELIA	5,708114524	-75,094697954
5	ANTIOQUIA	59	ARMENIA	6,163683758	-75,809955281
5	ANTIOQUIA	79	BARBOSA	6,443417935	-75,329240854
5	ANTIOQUIA	86	BELMIRA	6,665853953	-75,68621659
5	ANTIOQUIA	88	BELLO	6,358841086	-75,589735878
5	ANTIOQUIA	91	BETANIA	5,727081285	-75,986891074
5	ANTIOQUIA	93	BETULIA	6,185783612	-75,956492463
5	ANTIOQUIA	101	CIUDAD BOLÍVAR	5,840995666	-76,010652526
5	ANTIOQUIA	107	BRICEÑO	7,111788462	-75,562234738
5	ANTIOQUIA	113	BURITICÁ	6,80906489	-75,912065947
5	ANTIOQUIA	120	CÁCERES	7,665155105	-75,223697441
5	ANTIOQUIA	125	CAICEDO	6,423463639	-75,99351498
5	ANTIOQUIA	129	CALDAS	6,054446841	-75,626426743
5	ANTIOQUIA	134	CAMPAMENTO	7,036017478	-75,287661778
5	ANTIOQUIA	138	CAÑASGORDAS	6,814950023	-76,02689632
5	ANTIOQUIA	142	CARACOLÍ	6,33797388	-74,733548173
5	ANTIOQUIA	145	CARAMANTA	5,555275296	-75,630820964

Figura 4. Cargue de información municipio. (Elaboración propia)

Los otros datos se cargaron, se transformaron, se editaron que no trajera datos nulos en el identificador municipio y en el nombre.



CÓD. DEP.	DEPARTAMENTO	CÓD. MUN.	MUNICIPIO	GRUPO DE CULTIVO	SUBGRUPO DE CULTIVO	CULTIVO	DESAGREGACIÓN REGIONAL Y/O SISTEMA PRO
81	ARAUCA	81,3	FORTUL	FRUTALES	CITRICOS	CITRICOS	CITRICOS
81	ARAUCA	81,794	TAME	FRUTALES	CITRICOS	CITRICOS	CITRICOS
25	CUNDINAMARCA	25,53	PARATEBUENO	FRUTALES	CITRICOS	CITRICOS	CITRICOS
47	MAGDALENA	47,245	EL BANCO	FRUTALES	CITRICOS	CITRICOS	CITRICOS
47	MAGDALENA	47,72	SANTA BARBARA DE PINTO	FRUTALES	CITRICOS	CITRICOS	CITRICOS
47	MAGDALENA	47,703	SAN ZENON	FRUTALES	CITRICOS	CITRICOS	CITRICOS
47	MAGDALENA	47,545	PIJIÑO DEL CARMEN	FRUTALES	CITRICOS	CITRICOS	CITRICOS
50	META	50,001	VILLAVICENCIO	FRUTALES	CITRICOS	CITRICOS	CITRICOS
50	META	50,11	BARRANCA DE UPIA	FRUTALES	CITRICOS	CITRICOS	CITRICOS
50	META	50,124	CABUYARO	FRUTALES	CITRICOS	CITRICOS	CITRICOS
50	META	50,27	EL DORADO	FRUTALES	CITRICOS	CITRICOS	CITRICOS
63	QUINDIO	63,19	CIRCASIA	FRUTALES	CITRICOS	CITRICOS	CITRICOS
68	SANTANDER	68,444	MATANZA	FRUTALES	CITRICOS	CITRICOS	CITRICOS
76	VALLE DEL CAUCA	76,364	JAMUNDI	FRUTALES	CITRICOS	CITRICOS	CITRICOS
76	VALLE DEL CAUCA	76,622	ROLDANILLO	FRUTALES	CITRICOS	CITRICOS	CITRICOS
76	VALLE DEL CAUCA	76,616	RIOFRIO	FRUTALES	CITRICOS	CITRICOS	CITRICOS
76	VALLE DEL CAUCA	76,1	BOLIVAR	FRUTALES	CITRICOS	CITRICOS	CITRICOS
76	VALLE DEL CAUCA	76,306	GINEBRA	FRUTALES	CITRICOS	CITRICOS	CITRICOS
76	VALLE DEL CAUCA	76,248	EL CERRITO	FRUTALES	CITRICOS	CITRICOS	CITRICOS
76	VALLE DEL CAUCA	76,243	EL AGUILA	FRUTALES	CITRICOS	CITRICOS	CITRICOS
5	ANTIOQUIA	5,679	SANTA BARBARA	FRUTALES	CITRICOS	CITRICOS	CITRICOS
81	ARAUCA	81,3	FORTUL	FRUTALES	CITRICOS	CITRICOS	CITRICOS
15	BOYACA	15,218	COVARACHIA	FRUTALES	CITRICOS	CITRICOS	CITRICOS
85	CASANARE	85,315	SACAMA	FRUTALES	CITRICOS	CITRICOS	CITRICOS
47	MAGDALENA	47,318	GUAMAL	FRUTALES	CITRICOS	CITRICOS	CITRICOS
47	MAGDALENA	47,245	EL BANCO	FRUTALES	CITRICOS	CITRICOS	CITRICOS
47	MAGDALENA	47,72	SANTA BARBARA DE PINTO	FRUTALES	CITRICOS	CITRICOS	CITRICOS
47	MAGDALENA	47,703	SAN ZENON	FRUTALES	CITRICOS	CITRICOS	CITRICOS
47	MAGDALENA	47,545	PIJIÑO DEL CARMEN	FRUTALES	CITRICOS	CITRICOS	CITRICOS
50	META	50,11	BARRANCA DE UPIA	FRUTALES	CITRICOS	CITRICOS	CITRICOS

Figura 5. Información cargada. (Elaboración propia)

2.1.3.3 Modelo de relaciones de la información

Este modelo lo genera automáticamente PowerBI, donde se muestran los identificadores de cada tabla y sus atributos. Los cuales se van a utilizar para realizar filtros y selecciones en el Dashboard.

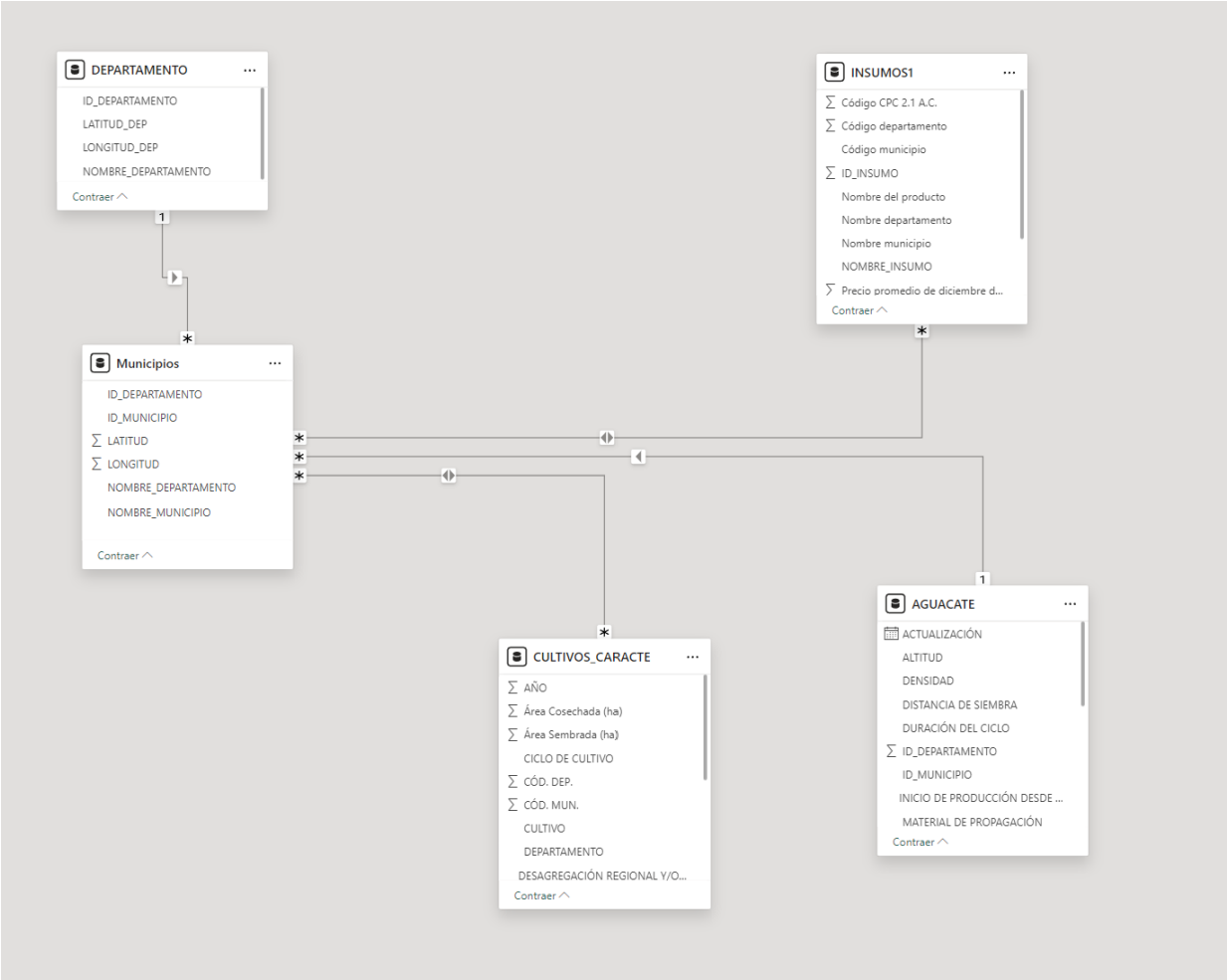


Figura 6. Relación de la información. (Elaboración propia)

2.1.4 Implementación

2.1.4.1 Dashboard Power BI web Indicadores sector agrícola

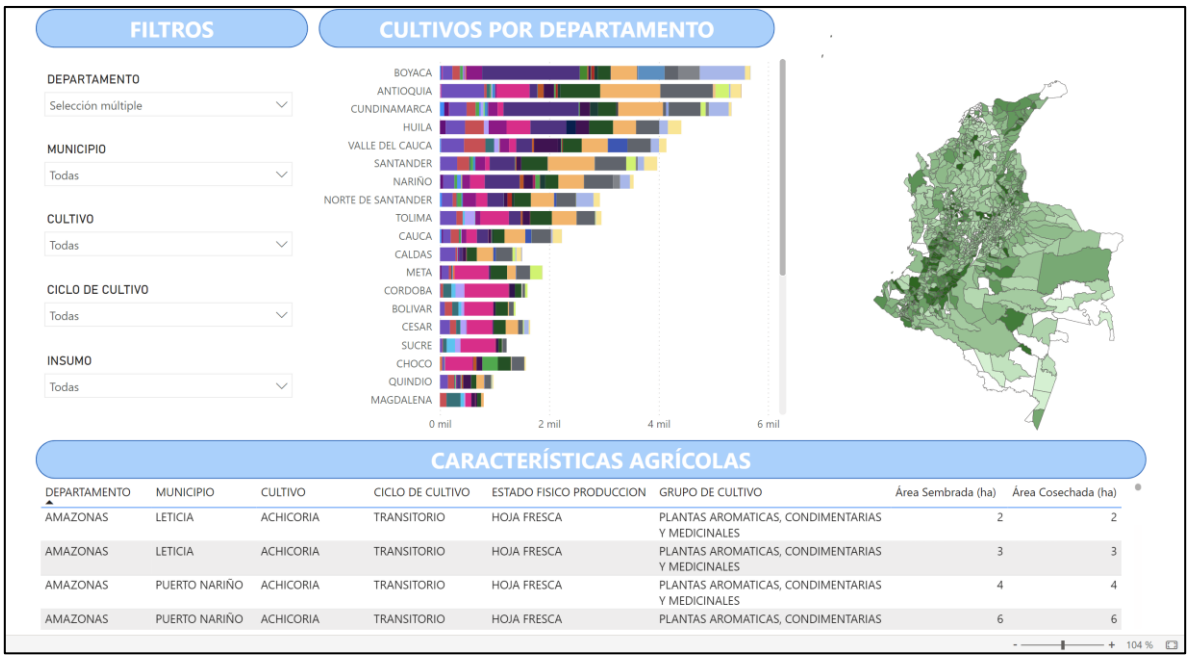


Figura 7. Indicadores sector agrícola. (Elaboración propia)

2.2. Arquitectura de software

La arquitectura de software es la estructura y diseño de un sistema de software. Es la que define como los componentes del software interactúan entre sí, cómo se organizan y cómo cumplen con los requisitos funcionales y no funcionales del sistema. Son las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos.

Los patrones de arquitectura son soluciones probadas para problemas comunes en el diseño y desarrollo de aplicaciones para aprovecharlas en otras creaciones. Es de especial importancia ya que la manera en que se estructura un sistema tiene un impacto directo sobre la capacidad de éste para satisfacer lo que se conoce como los atributos de calidad del sistema.

Dentro de las arquitecturas de software se encuentra la de microservicios, está divide una aplicación en una serie de servicios implementables de forma independiente que se comunican a través de API. Este enfoque permite implementar y escalar cada servicio individual de forma independiente, así como la entrega rápida y frecuente de aplicaciones grandes y complejas.

Dentro de los modelos de representación de software centrados en los componentes físicos se encuentran el diagrama de despliegue el cual muestra la arquitectura de ejecución de un sistema y el diagrama de componente que se utiliza en el desarrollo para describir sistemas con arquitectura orientada a servicios. A continuación, se describen estos dos diagramas con base en la implementación de la aplicación realizada.

2.2.1 Diagrama de componentes

Los diagramas de componentes están clasificados como diagrama de estructura, representa de forma estática el sistema de información. Este diagrama proporciona una vista de alto nivel de los componentes dentro de un sistema. Los componentes pueden ser de software una base de datos o interfaz de usuario, de hardware un circuito, microchip o dispositivo; o una unidad de negocio como un proveedor, nómina o envío.

El diagrama muestra las dependencias que tiene cada uno de los componentes de la aplicación. Centrándose en cada proceso y función que realiza dentro del contexto del software, identificando la estructura general y su comportamiento de servicio.

El siguiente diagrama de componentes para el proyecto señala las distribuciones, implementando tres capas principales que soportará el aplicativo y una capa paralela a todas la cual tendrá objetos de transferencia de datos y utilitarios necesarios para el correcto funcionamiento del sistema.

2.2.1.1 Capa de Presentación: Para este proyecto se diseñó una interfaz de usuario tipo web, la cual está construida con tecnología como Angular, Framework de JavaScript de código abierto. Con el cual se implementarán todos los formularios y estilos necesarios en el proyecto.

2.2.1.2 Capa de Lógica de negocio: La capa de lógica de negocios es un intermediario para el intercambio de datos entre la capa de presentación y la capa de acceso a datos. Maneja reglas, cálculos y lógica dentro de una aplicación que dictan cómo se comportan. El objetivo de este elemento en la arquitectura es garantizar que la capa de presentación contenga la menor lógica de negocio posible con el fin de tener la posibilidad a futuro de reutilizar el componente para una capa de interfaz por ejemplo una aplicación de escritorio.

2.2.1.3 Capa de acceso a datos: Es una capa que proporciona acceso simplificado a los datos almacenados en el almacenamiento persistente de algún tipo, tal como una entidad relación de base de datos. Para el proyecto se utilizará PostgreSQL, como gestor de datos, AWS web para crear un servicio en la nube.

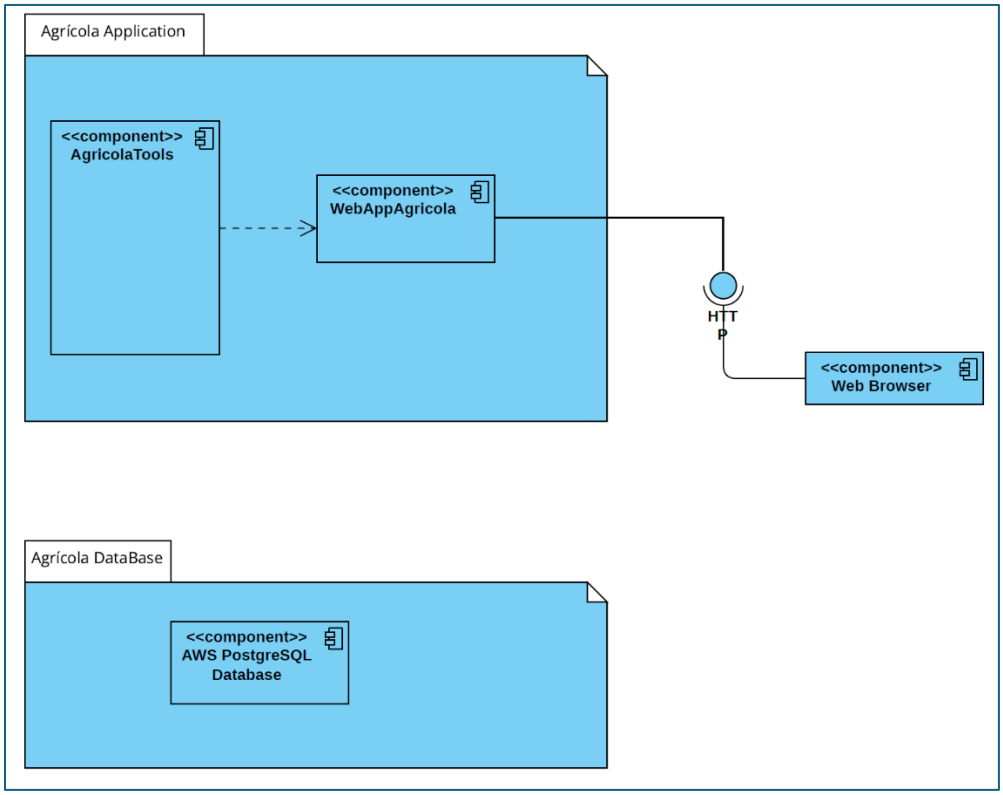


Figura 8. Diagrama de Componentes. (Elaboración propia)

2.2.1 Diagrama de despliegue

El diagrama de despliegue se utiliza para representar el despliegue físico de los componentes de un sistema. Muestra cómo se asignan los artefactos de software a los componentes de hardware, conocidos como nodos, e ilustra las configuraciones físicas de software y hardware que son críticas para la ejecución, operación y escalabilidad de un sistema.

En el proyecto de grado, la aplicación tendrá tres nodos Cliente, Servidor de Aplicaciones y Servidor de Bases de datos, y tres componentes Web Browser, Web Appp Agrícola y AWS MySQL Database. El usuario por medio del Internet utilizando un navegador web, buscará la app, ingresará a través del protocolo HTTP, realizará una consulta, se desplegarán las opciones del servidor del sitio web, una vez se reciben las peticiones se comunicará con el motor de base de datos.

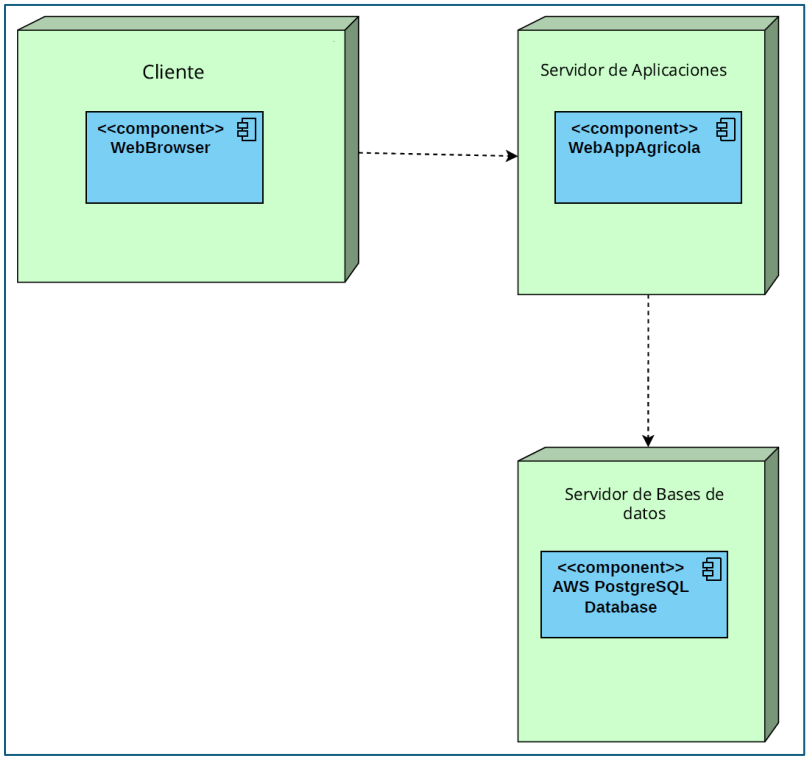


Figura 9. Diagrama de Despliegue. (Elaboración propia)

2.3. Metodología de desarrollo

2.3.1 Aplicación de la metodología Kanban

El desarrollo de un aplicativo web estará enmarcado dentro de la metodología Kanban en la cual se creará un tablero en la herramienta Trello con las tareas para la realización de proyecto y se visualizará el estado de cada una.

Al ser un método visual permite que con un simple vistazo se conozca el estado del proyecto y se pueda asignar nuevas tareas de manera efectiva.

2.3.1.1 Implementación del Tablero Kanban

Fase del ciclo de producción

Este método se basa en el desarrollo incremental, en la división del trabajo en diferentes partes. Por lo tanto, se dividirá el trabajo en diferentes pasos, a continuación de muestra las listas o lotes de tarjetas que representan hitos, conjunto de ideas u objetivos del equipo:

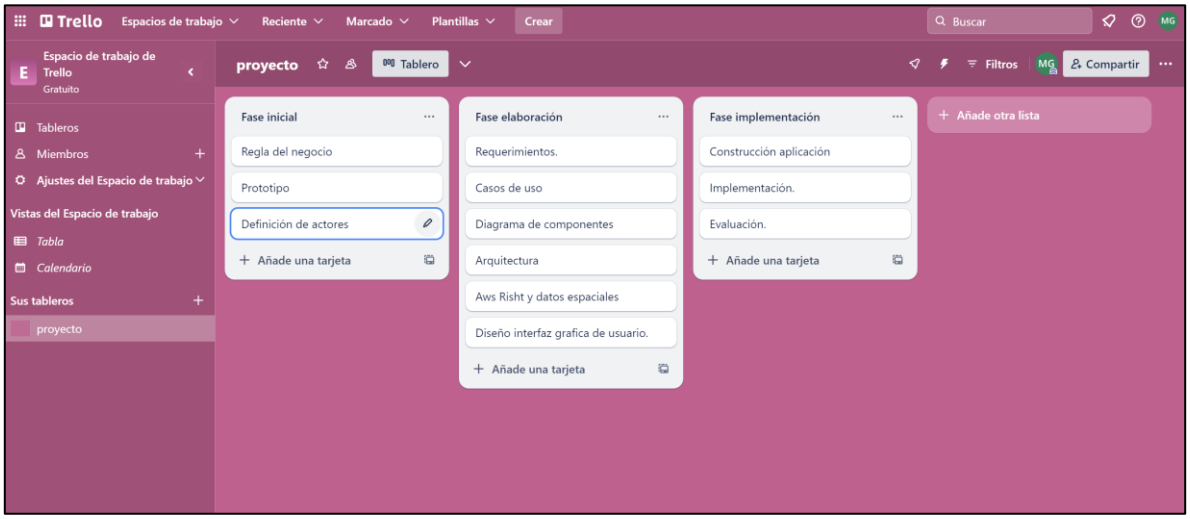


Figura 10. Tablero metodología diseño herramienta tecnológica. (Elaboración propia)

2.4. Análisis Funcional

2.4.1 Definición de requerimientos

Tabla 6. Requerimientos

ID	Subsistema	Nombre	Descripción	Importancia	Estimación / Días	Tipo
1	Gestión de usuarios	Administrar usuarios	Interfaz que permite realizar toda la gestión de usuarios de ingreso a la aplicación	1	2	RF
2	Gestión de información	Administrador información	Interfaz que permite ingresar nueva información del sector agrícola	1	1	RF
3	Gestión de usuarios	Control de acceso	Se requiere validar de existencia del usuario que desee entrar al aplicativo	1	1	RF
4	Gestión de usuarios	Persistencia datos de usuario	Los datos de usuario deben ser	1	2	RF

ID	Subsistema	Nombre	Descripción	Importancia	Estimación / Días	Tipo
			almacenados para su respectiva consulta y/o actualización			
5	Gestión de actividades	Selección de información	Interfaz que permite al usuario seleccionar la información agrícola, allí se debe generar las clasificaciones por departamento y municipio.	2	4	RF
6	Gestión de actividades	Consulta de información seleccionada	Todos los datos seleccionados deben poder se generar en un reporte.	1	3	RF

Fuente: Elaboración propia.

2.4.2 Descripción de Requerimientos funcionales

Los requisitos generales del proyecto se refieren a los aspectos clave que debe cumplir la herramienta tecnológica que permita la visualización de indicadores en la producción agropecuaria en el país. Los requisitos funcionales de la aplicación son los siguientes:

Tabla 7. Descripción REQ-q. Crear y actualizar usuarios.

REQ -1	DESCRIPCIÓN
Nombre	Administrar usuarios
Clase	Requerimiento Funcional
Explicación breve del requerimiento	El Usuario podrá ingresar al sistema con su rol de administrador donde podrá realizar la gestión de usuarios.
Objetivo	Encargado de validar la creación y/o actualización de usuarios para permitir el ingreso a la aplicación.
Entrada	Login y Password
Proceso	El usuario administrador ingresa con su login y password, predeterminado para su rol, luego podrá realizar las actividades de gestión de usuarios.
Salida	Validación, aceptación o mensaje de error al realizar una creación y/o actualización.

Fuente: Elaboración propia.

Tabla 8.Descripción REQ-2. Administrar información.

REQ -2	DESCRIPCIÓN
Nombre	Administrar información
Clase	Requerimiento Funcional
Explicación breve del requerimiento	El Administrador podrá ingresar con su rol al aplicativo donde podrá ingresar nueva información.
Objetivo	Encargado de ingresar nueva información al aplicativo.
Entrada	Login y Password.
Proceso	El usuario Administrador ingresa con su login y password, luego ingresa a la pestaña ingresar nueva información, donde podrá incorporar nuevos datos.
Salida	Validación, aceptación o mensaje de advertencia al intentar crear y/o actualizar los datos.

Fuente: Elaboración propia.

Tabla 9. Descripción REQ-3. Control de Acceso.

REQ -3	DESCRIPCIÓN
Nombre	Control de acceso
Clase	Requerimiento Funcional
Explicación breve del requerimiento	El sistema realiza la verificación de los usuarios que están tratando de ingresar a la aplicación.
Objetivo	Controlar el acceso de usuarios que no tengan login y password.
Entrada	Login y Pasword
Proceso	El usuario ingresa su respectivo login y password, el sistema identifica si es un usuario válido dentro del sistema, de no ser así no permitirá el ingreso.
Salida	Validación, aceptación o mensaje de advertencia al intentar realizar la autenticación.

Fuente: Elaboración propia.

Tabla 10. Descripción REQ-4. Persistencia de datos de los usuarios.

REQ -4	DESCRIPCIÓN
Nombre	Persistencia de datos de los usuarios
Clase	Requerimiento Funcional
Explicación breve del requerimiento	El sistema debe almacenar la información de los usuarios.
Objetivo	Permite realizar la verificación de datos en el proceso de autenticación. También mantiene la integridad de la información para realizar consultas o actualizaciones.
Entrada	Data del usuario.
Proceso	El sistema registra y consulta información de los usuarios en la base de datos MySQL.
Salida	Validación, aceptación o mensaje de advertencia al intentar crear y/o actualizar los datos.

Tabla 11. Descripción REQ-5. Persistencia de la información.

REQ -5	DESCRIPCIÓN
Nombre	Persistencia de la información
Clase	Requerimiento Funcional
Explicación breve del requerimiento	El sistema debe almacenar la información actualizada o nueva.
Objetivo	Permite ingresar nuevos datos al aplicativo. También mantiene la integridad de la información para realizar consultas o actualizaciones.
Entrada	Data agrícola.
Proceso	El sistema registra y consulta información nueva en la base de datos MySQL.
Salida	Validación, aceptación o mensaje de advertencia al intentar crear y/o actualizar los datos.

Fuente: Elaboración propia.

Tabla 12. Descripción REQ-6. Consulta de información.

REQ -6	DESCRIPCIÓN
Nombre	Consulta de información
Clase	Requerimiento Funcional
Explicación breve del requerimiento	El sistema debe proveer una interfaz que permita a los usuarios consultar la información del sector agrícola.
Objetivo	Permite realizar la consulta de la información, visualizando en el mapa donde está ubicada esta, por medio de una gráfica comparativa con otros lugares que tenga la misma clase de producto agrícola.
Entrada	Elemento visual que permita realizar la selección de la información de interés.
Proceso	El usuario después de autenticarse ingresa al aplicativo y visualiza las diferentes opciones que tiene para seleccionar la información. A medida que selecciona información las otras casillas van cambiando acorde con la selección.
Salida	Mapa con el departamento seleccionado, graficas relacionada con la información seleccionada y reporte de la información.

Fuente: Elaboración propia.

Tabla 13. Descripción REQ-7. Consulta de información flujo de caja.

REQ -7	DESCRIPCIÓN
Nombre	Consulta de información flujo de caja
Clase	Requerimiento Funcional
Explicación breve del requerimiento	El sistema debe proveer una interfaz que permita a los usuarios consultar la información del flujo de caja del cultivo seleccionado.
Objetivo	Permite realizar la consulta de la información, visualizando en el mapa donde está ubicada esta, muestra una tabla con la

	información de flujo de caja del cultivo seleccionado y muestra un comparativo de insumos empleados.
Entrada	Elemento visual que permita realizar la selección de la información de interés.
Proceso	El usuario selecciona la pestaña flujo de caja y visualiza las diferentes opciones que tiene de información. A medida que selecciona información las otras casillas van cambiando la información.
Salida	Mapa con el departamento seleccionado, tabla de flujo de caja del cultivo seleccionado, gráficas relacionada con la información seleccionada y la opción imprimir reporte.

Fuente: Elaboración propia.

2.4.3 Requerimientos no funcionales del proyecto

En este ítem se exponen las características que restringen la operación o funcionamiento de la herramienta tecnológica, en la tabla 14 se listan los requerimientos no funcionales:

Tabla 14. Requerimientos no funcionales.

ÍTEM	DESCRIPCIPÓN
RNF - 01	La herramienta tecnológica se debe desplegar en cualquier navegador web sin la necesidad de instalar complementos adicionales.
RNF - 02	El tiempo máximo de respuesta ante peticiones cortas no debe superar los 8 segundos.
RNF - 03	El tiempo máximo de respuesta ante peticiones de procedimientos largos no debe superar los 4 minutos.
RNF - 04	La herramienta tecnológica debe soportar un máximo de 50 usuarios simultáneos y esto no afectará el rendimiento de respuesta de las peticiones.
RNF - 05	La herramienta tecnológica debe estar documentada adecuadamente (manuales técnicos, manuales funcionales, manual de instalación, diccionario de taos, entre otro).
RNF - 06	Indicar el tipo de licenciamiento y la infraestructura tecnológica adicional requerida para soportar y ejecutar la herramienta tecnológica.
RNF - 07	La herramienta tecnológica debe cumplir con los estándares bajo una arquitectura de microservicios.
RNF - 08	Debe ser adaptativo, quiere decir que se visualizará independiente del tamaño de pantalla.
RNF - 09	La navegación y funcionalidad de las herramientas debe ser moderno, intuitivo y de fácil acceso facilitando el uso del aplicativo de usuarios.
RNF - 10	La herramienta tecnológica debe proveer una ayuda en línea en la que se muestran las funcionalidades y prácticas de uso.
RNF - 11	El idioma debe ser español.
RNF - 12	Debe tener una ayuda.

Fuente: Elaboración propia.

2.4.3 Casos de Uso

En base a los requerimientos que se han identificado y con el fin de visualizar más claramente las interacciones de los diferentes actores involucrados con los requerimientos levantados presentamos los siguientes diagramas de casos de uso de la aplicación.

2.4.3.1 Caso de Uso General

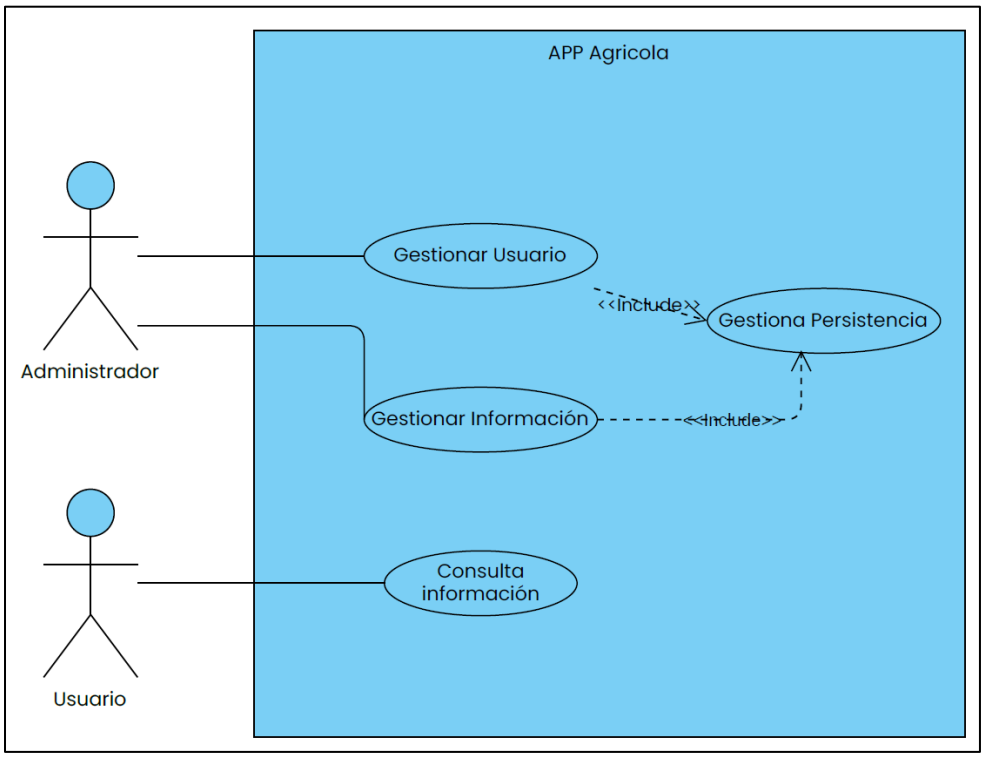


Figura 11. Caso de uso general. (Elaboración propia)

2.4.3.2 Gestión de Usuarios

Este caso de uso ilustra la gestión de usuarios, desde su creación, edición y hasta actualización de contraseñas.

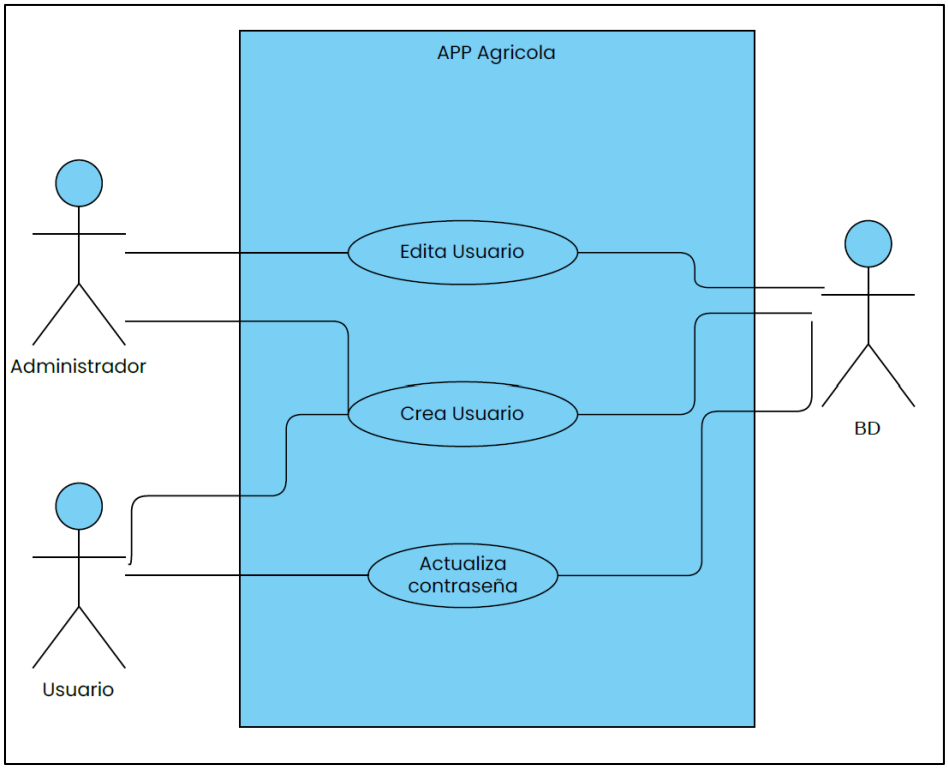


Figura 12. Caso de uso Gestión de Usuario. (Elaboración propia)

2.4.3.3 Gestión de Información

Este caso de uso ilustra la gestión de información, desde su creación, edición y actualización.

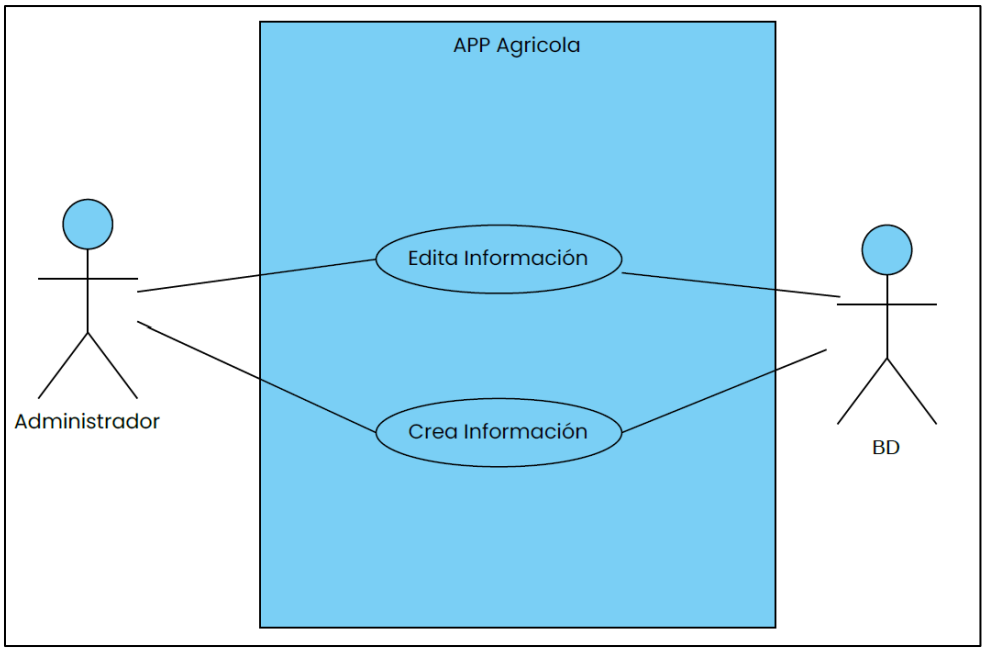


Figura 13. Caso de uso Gestión de información. (Elaboración propia)

2.4.3.4 Consulta información

Este caso de uso permite ver que el usuario consulta y genera reporte.

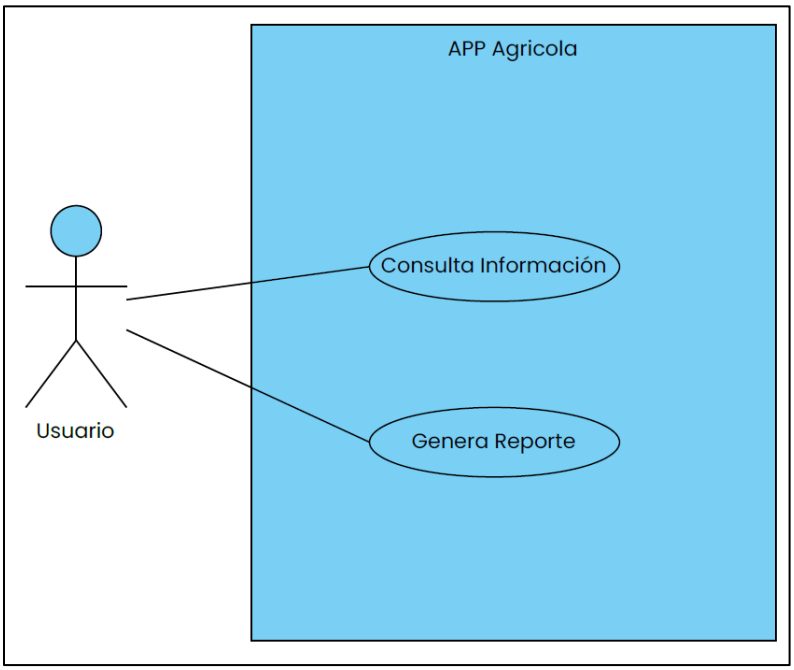


Figura 14. Caso de uso Consulta información. (Elaboración propia)

2.4.4. Diagramas de Actividades

El diagrama de actividad muestra las actividades ejecutadas por la aplicación. A continuación, se ilustran el flujo de las principales actividades generadas en las actividades de la aplicación.

2.4.4.1 Gestión de Usuarios Administrador

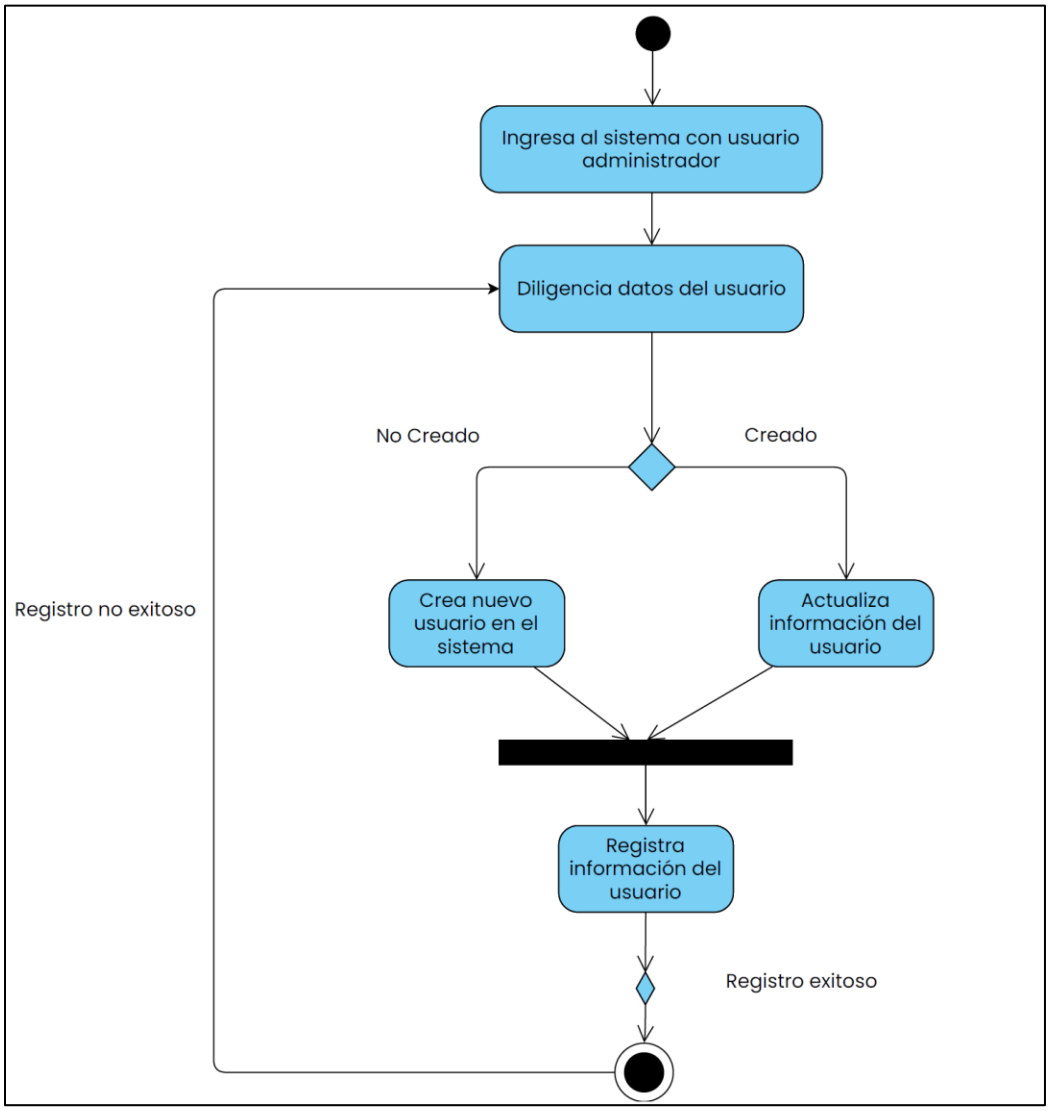


Figura 15. Diagrama de actividades Gestión de Usuarios Administrador. (Elaboración propia)

2.4.4.2 Gestión de Usuarios

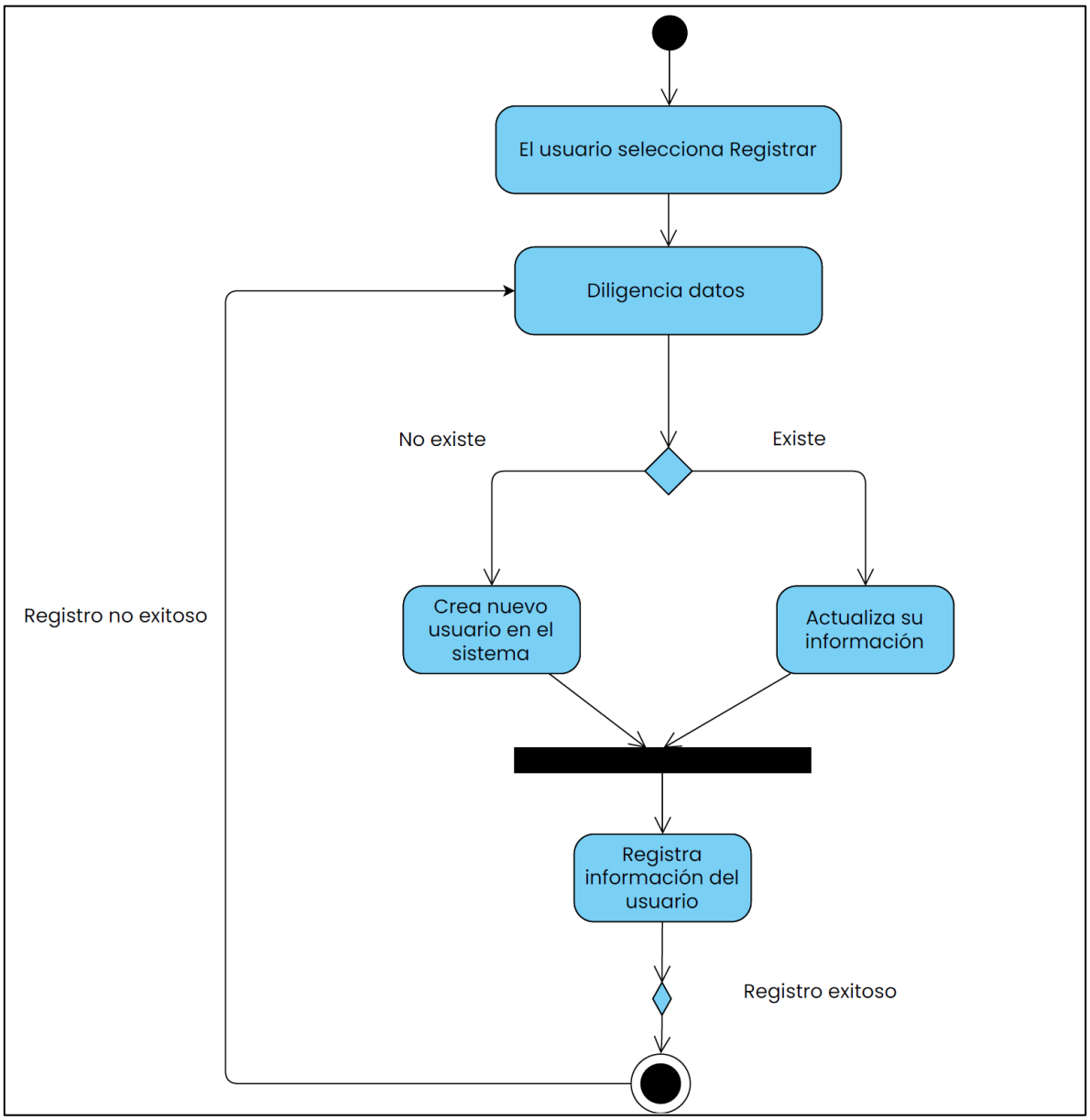


Figura 16. Diagrama de actividad Gestión de usuario. (Elaboración propia)

2.4.4.3 Consulta de información

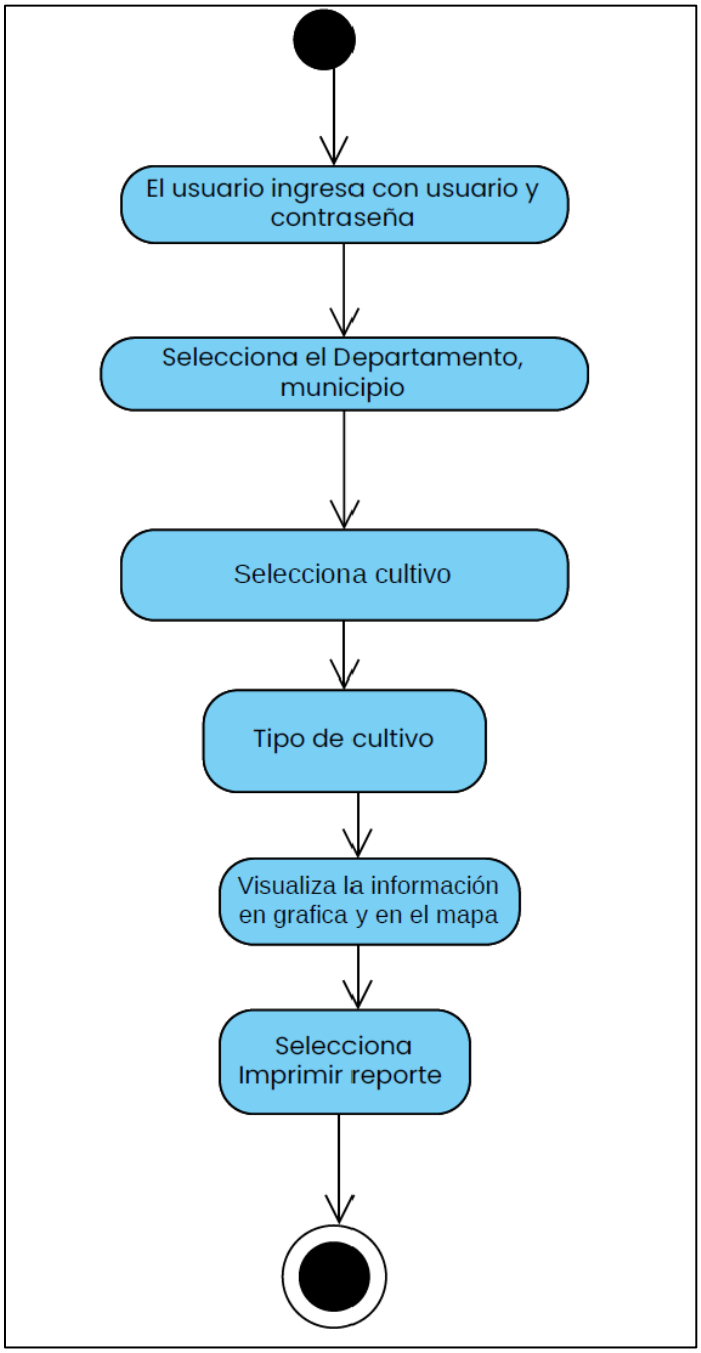


Figura 17. Diagrama de actividad Consulta de información. (Elaboración propia)

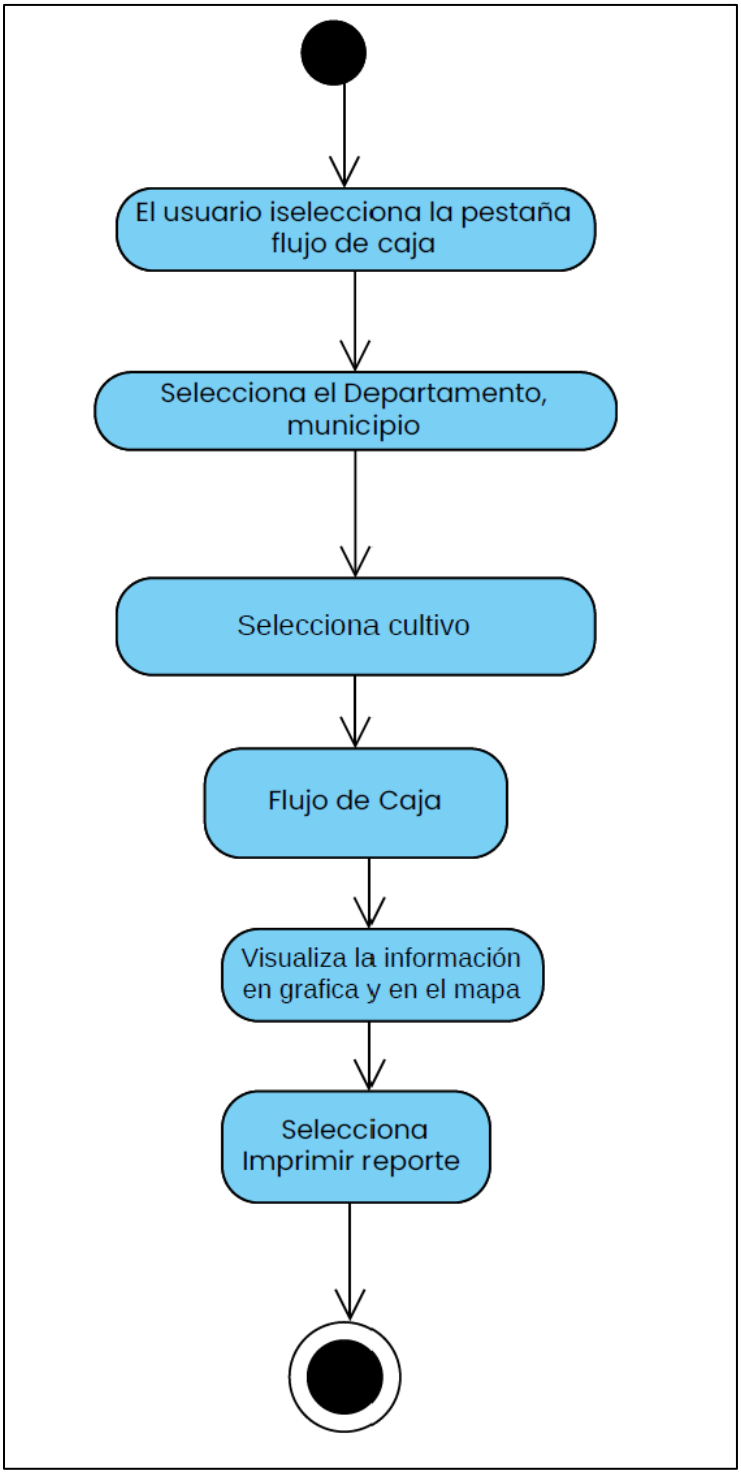


Figura 18. Diagrama de actividad Consulta de información. (Elaboración propia)

2.4.5. Prototipo de la herramienta tecnológica

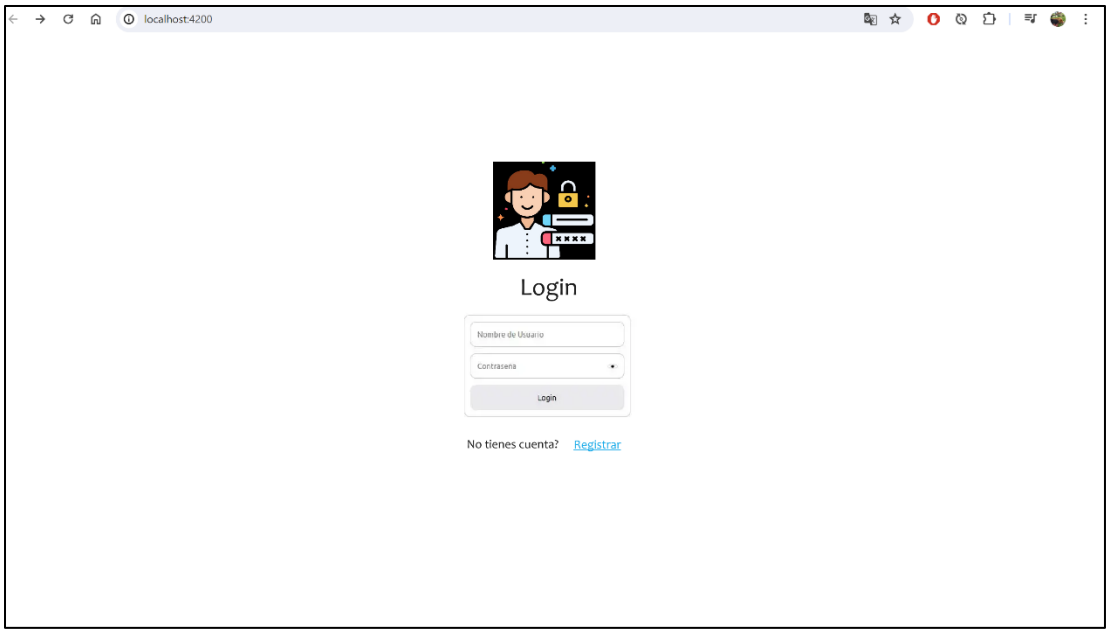


Figura 19. Interfaz registro. (Elaboración propia)

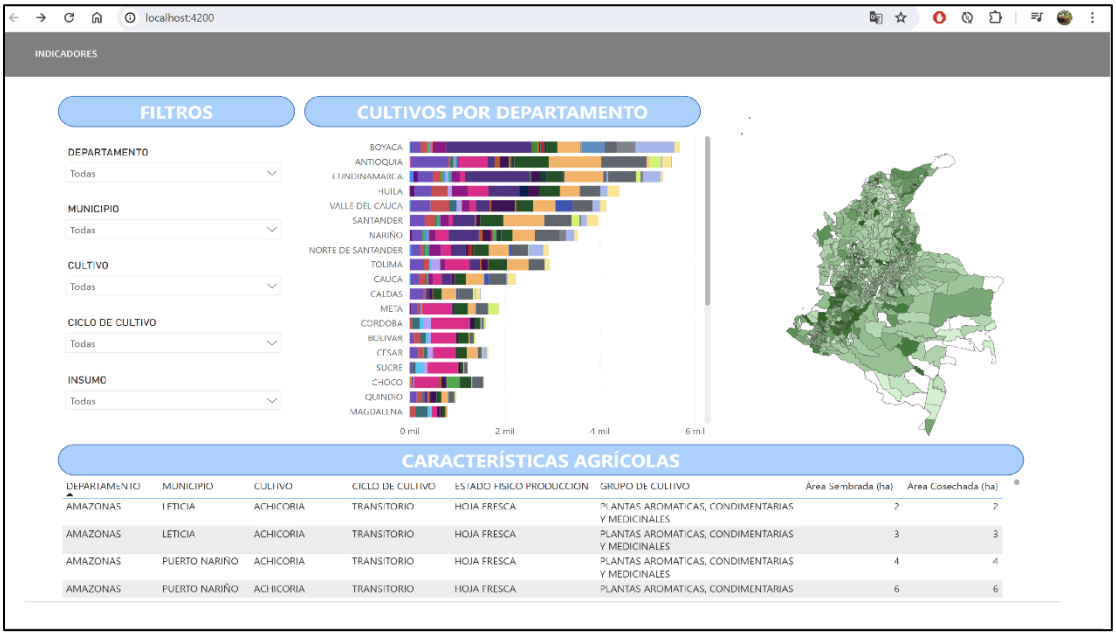
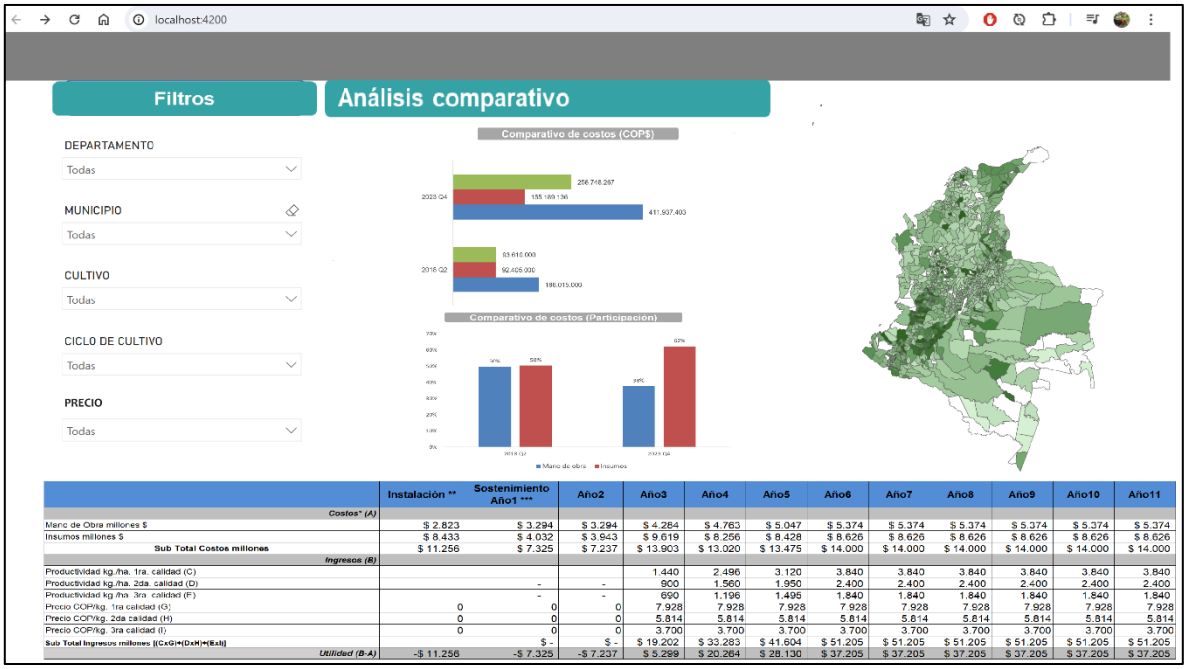


Figura 20. Interfaz información agrícola. (Elaboración propia)



	Instalación **	Sostenimiento Año1 ***	Año2	Año3	Año4	Año5	Año6	Año7	Año8	Año9	Año10	Año11
Costos (A)*												
Mano de Obra millones $	$ 2.823	$ 3.294	$ 3.294	$ 4.284	$ 4.763	$ 5.047	$ 5.374	$ 5.374	$ 5.374	$ 5.374	$ 5.374	$ 5.374
Insumos millones $	$ 8.433	$ 4.032	$ 3.943	$ 9.619	$ 8.256	$ 8.428	$ 8.626	$ 8.626	$ 8.626	$ 8.626	$ 8.626	$ 8.626
Sub Total Costos millones	$ 11.256	$ 7.325	$ 7.237	$ 13.903	$ 13.020	$ 13.475	$ 14.000	$ 14.000	$ 14.000	$ 14.000	$ 14.000	$ 14.000
Ingresos (B)												
Productividad kg./ha. 1ra. calidad (C)				1.440	2.496	3.120	3.840	3.840	3.840	3.840	3.840	3.840
Productividad kg./ha. 2da. calidad (D)		-	-	900	1.560	1.950	2.400	2.400	2.400	2.400	2.400	2.400
Productividad kg./ha. 3ra. calidad (F)		-	-	690	1.196	1.495	1.840	1.840	1.840	1.840	1.840	1.840
Precio COP/kg. 1ra calidad (G)	0	0	0	7.928	7.928	7.928	7.928	7.928	7.928	7.928	7.928	7.928
Precio COP/kg. 2da calidad (H)	0	0	0	5.814	5.814	5.814	5.814	5.814	5.814	5.814	5.814	5.814
Precio COP/kg. 3ra calidad (I)	0	0	0	3.700	3.700	3.700	3.700	3.700	3.700	3.700	3.700	3.700
Sub Total Ingresos millones [(CxG)+(DxH)+(ExI)]	$ -	$ -	$ -	$ 19.202	$ 33.283	$ 41.604	$ 51.205	$ 51.205	$ 51.205	$ 51.205	$ 51.205	$ 51.205
Utilidad (B-A)	-$ 11.256	-$ 7.325	-$ 7.237	$ 5.299	$ 20.264	$ 28.130	$ 37.205	$ 37.205	$ 37.205	$ 37.205	$ 37.205	$ 37.205

Figura 21. Interfaz flujo de caja. (Elaboración propia)

2.5. Modelo de clases

A continuación, se muestra el respectivo modelo de clases generado durante el desarrollo de la aplicación.

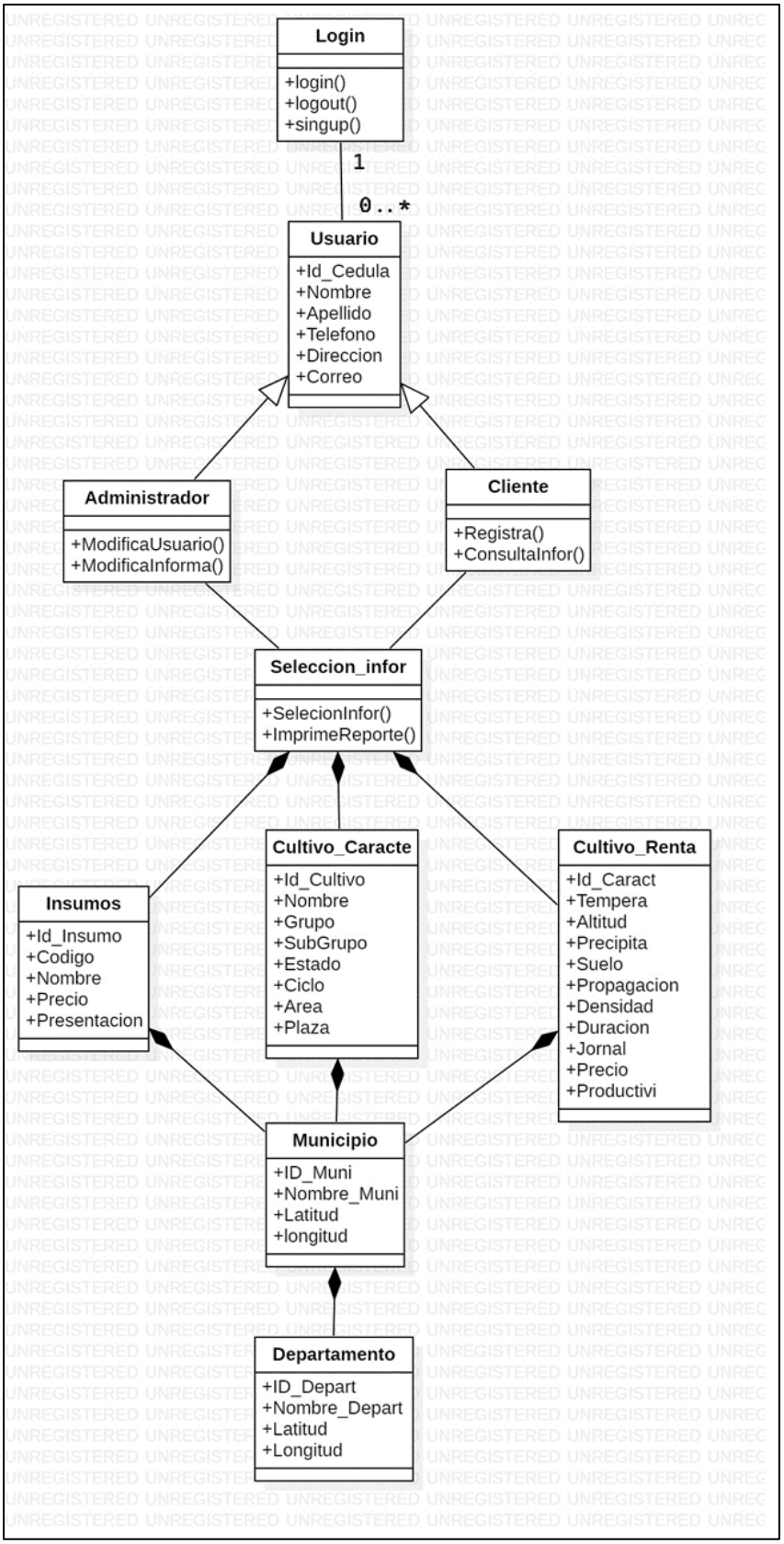


Figura 22. Modelo de clases. (Elaboración propia)

2.6. Modelo Relacional

A continuación, se muestra el modelo de base de datos implementado para el funcionamiento de la aplicación.

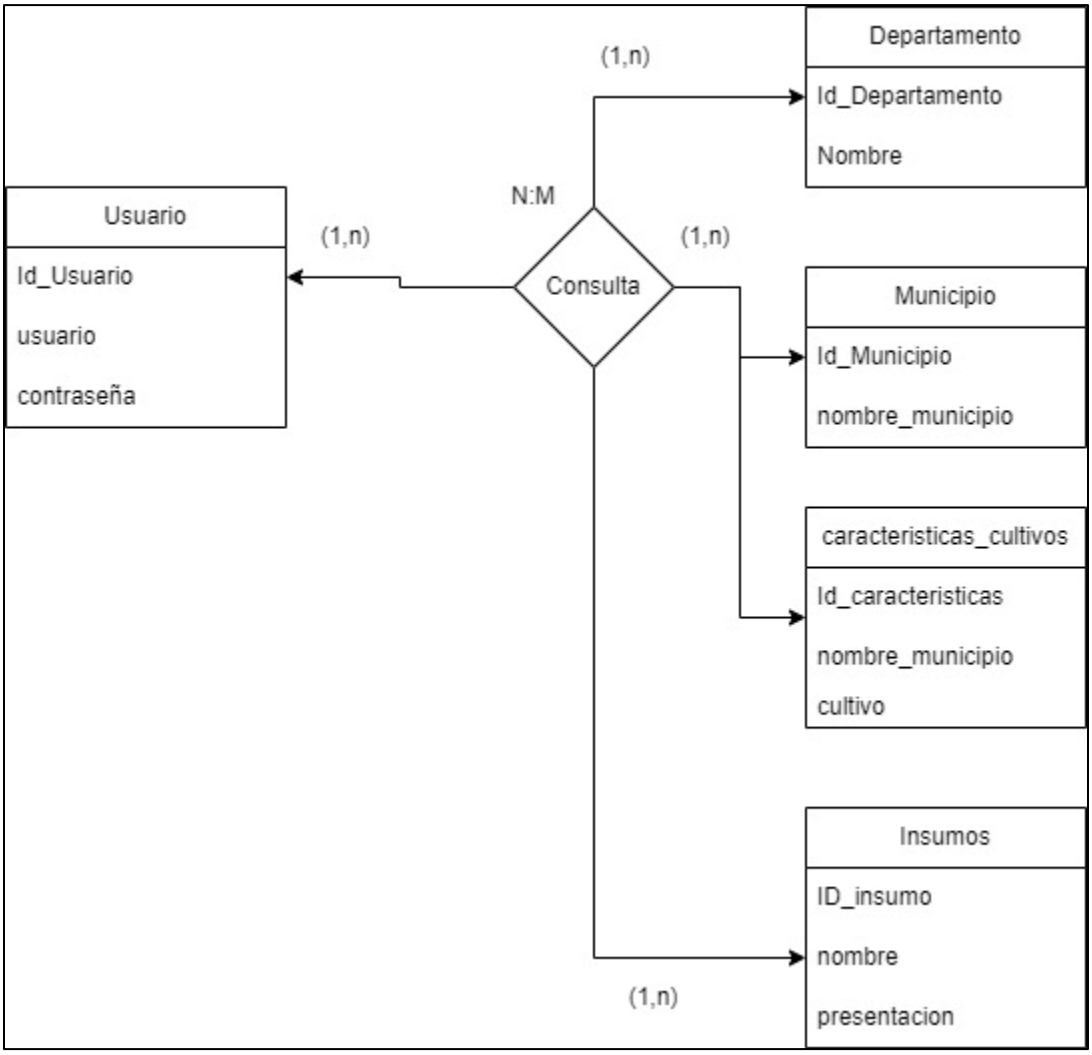


Figura 23. Modelo Relacional

3. Implementación del prototipo

Para el desarrollo de este prototipo se tuvieron en cuenta diferentes frameworks y lenguajes de programación, para la base de datos se usó el lenguaje de SQL bajo el motor MySQL, para el backend se usó el lenguaje de programación JavaScript con el framework NodeJs, para la capa de Frontend se usó el lenguaje de Typescript bajo el framework de Angular

17.3.7. Se usó Amazon RDS para la base de datos MySQL, AWS Elastic Beanstalk para el backend para desplegar la aplicación Node.js y AWS S3 para alojar la aplicación (y opcionalmente CloudFront) para el frontend Angular 17.3.7.

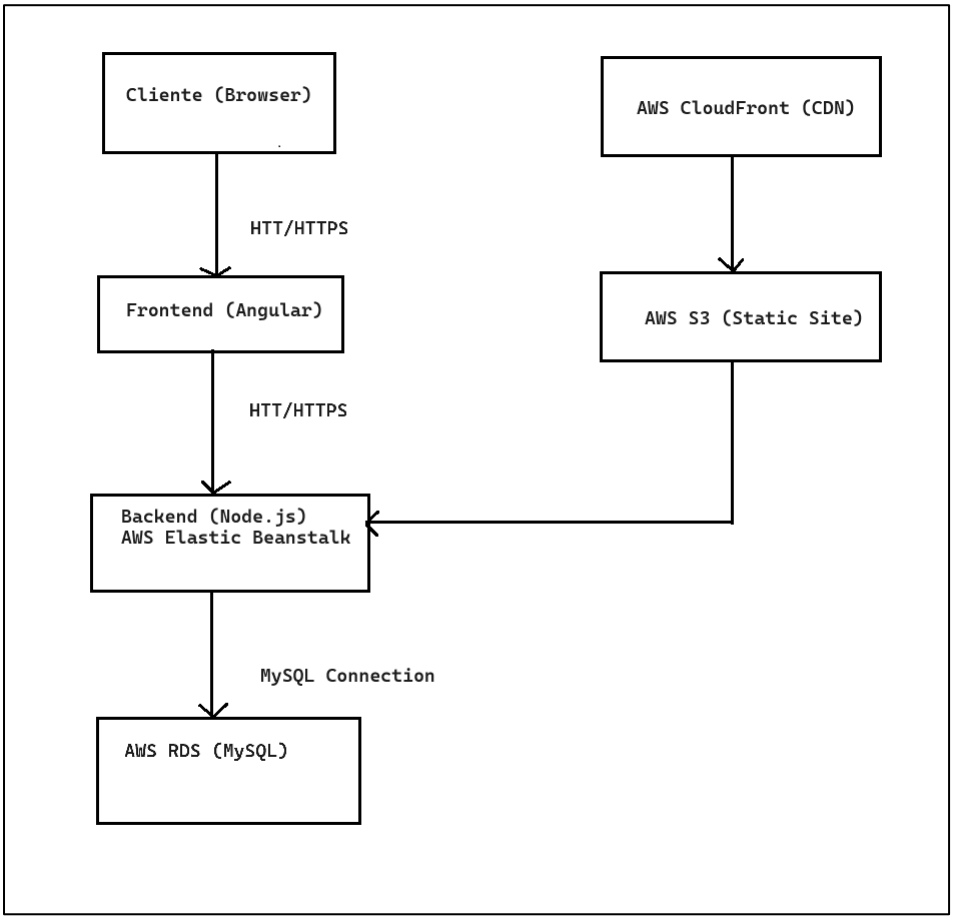


Figura 24. Diagrama de Arquitectura. (Elaboración propia)

3.1 Base de datos

Para el proyecto se elaboró un base de datos, con dos tablas de almacenamiento, la primera con características del municipio y la segunda con información de registro de usuarios y contraseñas. Las cuales se crearon en MySQL por medio de MySQL Workbench y posteriormente se realizó un RDS (servicio de bases de datos relacionales), una instancia de base de datos MySQL la cual usa volúmenes de Amazon Elastic Block Store (Amazon EBS) para el almacenamiento de base de datos y archivos de registro.

Se creó una tabla denominada ‘usuarios’ que almacena información de usuario, incluyendo el nombre de usuario y contraseña. También se creó una table denominada ‘product’ que contiene información de los municipios como temperatura, precipitación, altitud, cultivo, entre otros. A continuación, se muestra el modelo de la base de datos.

Modelo de Base de Datos
Estructura de la Tabla ‘Usuarios’

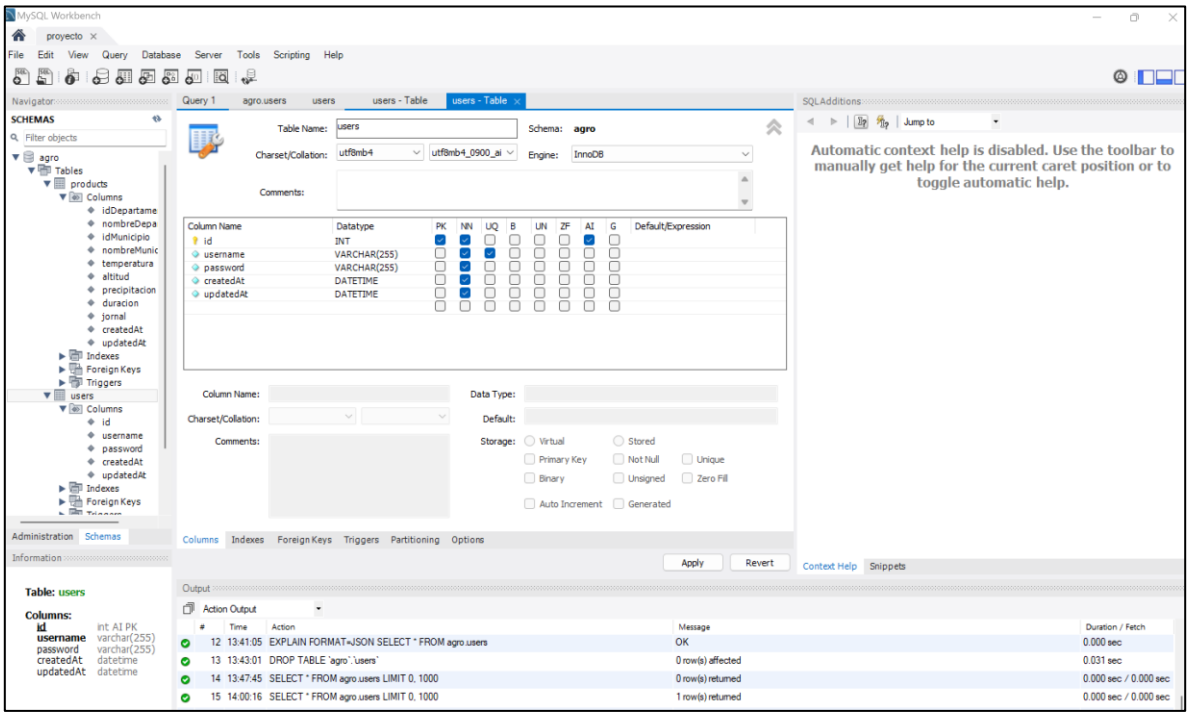


Figura 25. Estructura de la Tabla ‘Usuario’. (Elaboración propia)

3.2 Interfaz de usuario - FronEnd

Para el desarrollo del FrontEnd se utilizó Angular, typescript, json web token, bootstrap, la estructura de carpetas son cinco componentes (carácter, dashboard, login, navbar y sign-in), dos interfaces (producto.ts y user.ts), tres archivos de servicio (error.service.ts, producto.service.ts y user.service.ts) dos de estos para hacer peticiones http y otro servicio para poder compartir código entre componentes. Se creó un spinner cuando se hacen las peticiones http e interceptores.

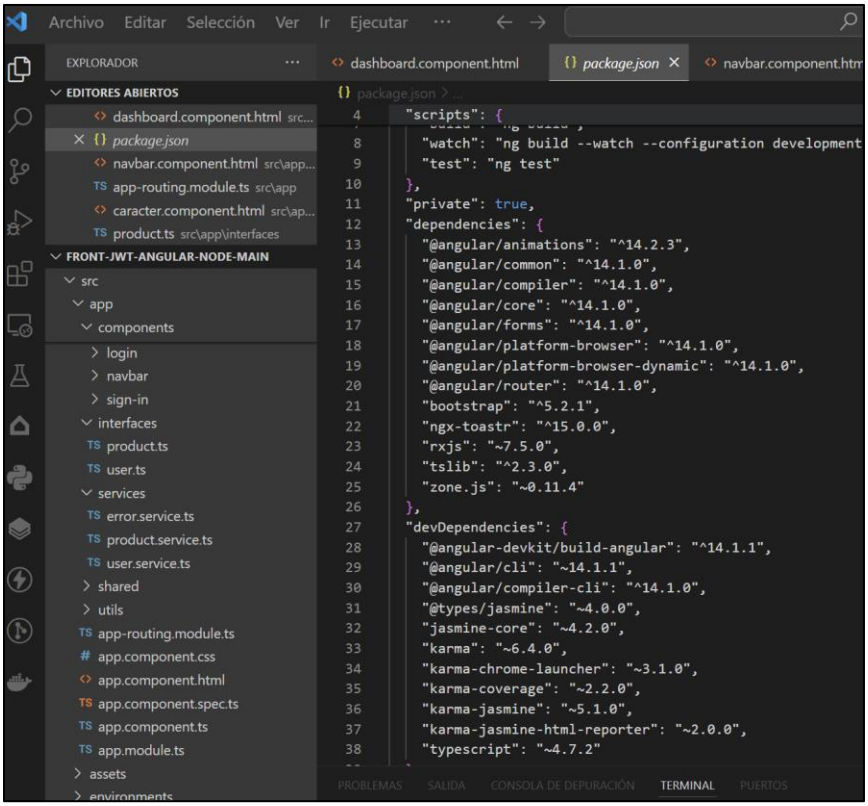


Figura 26. Herramientas instaladas. (Elaboración propia)

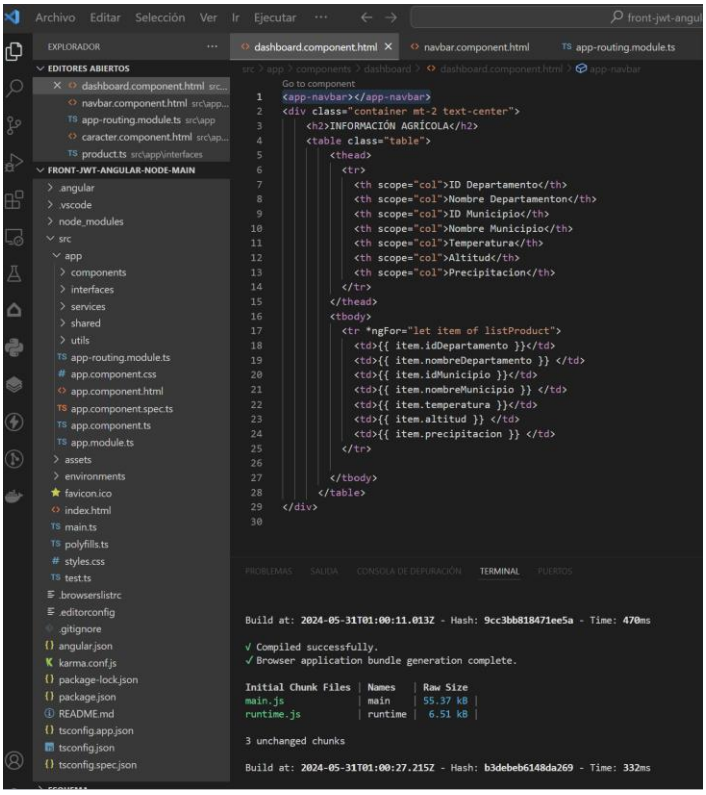


Figura 27. Distribución carpetas Frontend. (Elaboración propia)

Implementación de un servicio para interactuar con el microservicio de autenticación, compuesto por tres archivos (error.service.ts, producto.service.ts y user.service.ts).

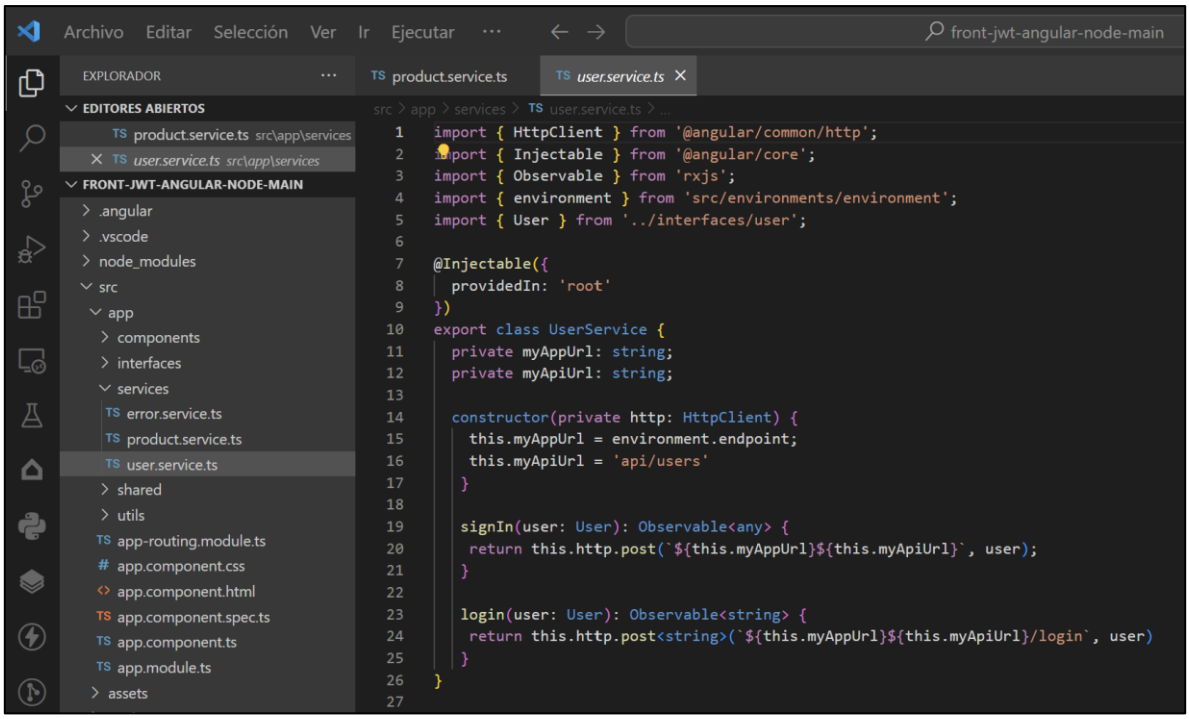


Figura 28.user.service.ts. (Elaboración propia)

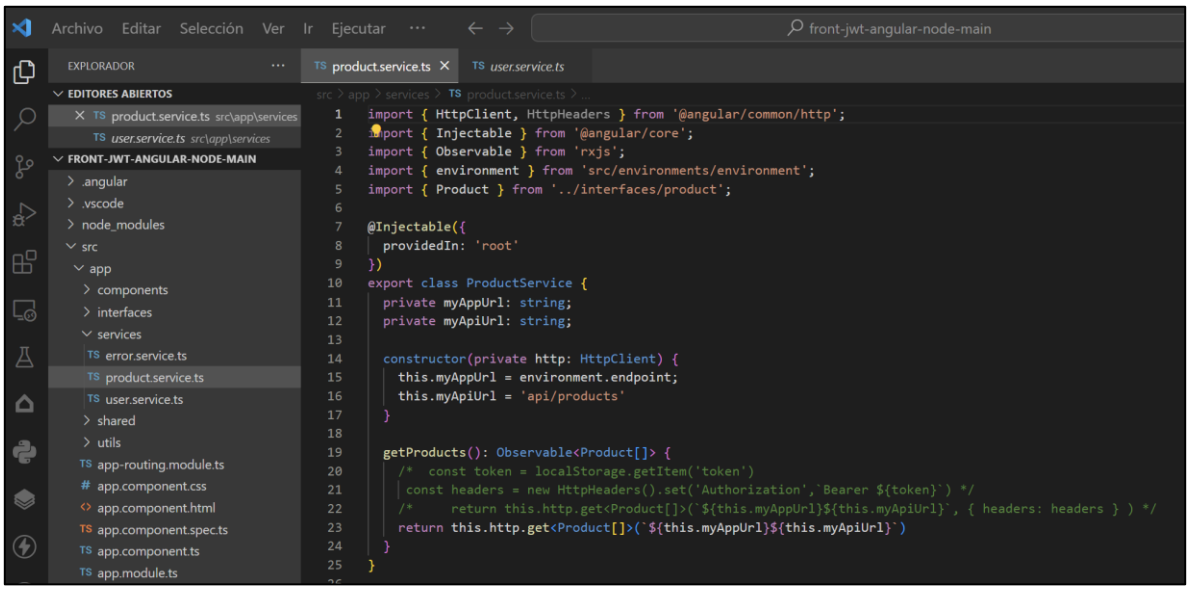


Figura 29.producto.service.ts. (Elaboración propia)

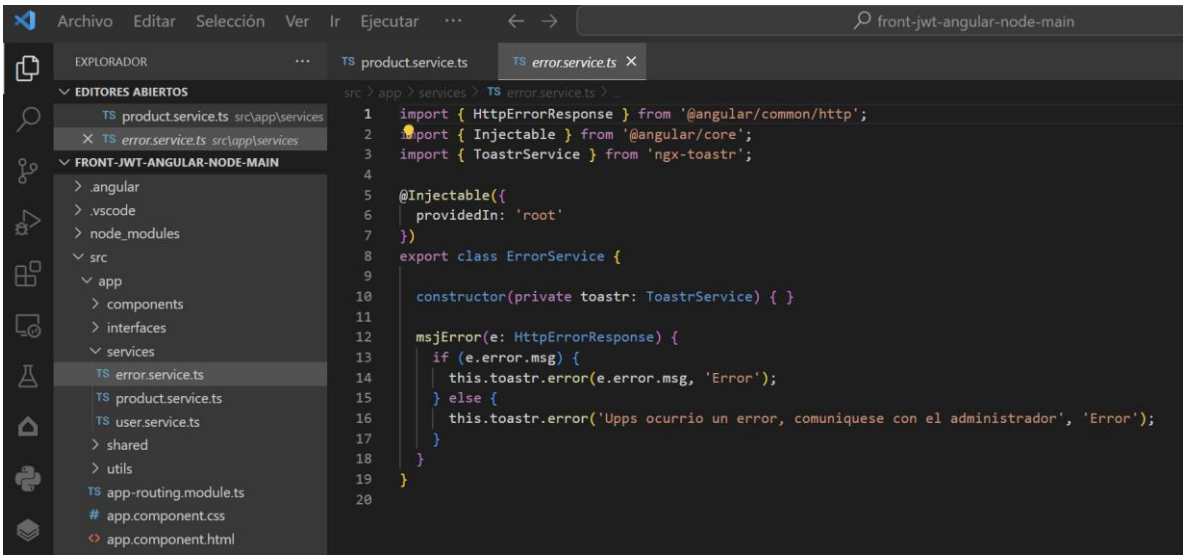


Figura 30.error.service.ts. (Elaboración propia)

3.2.1 Vista ingreso

Para el ingreso al aplicativo, se creó esta vista que cuenta con dos campos de tipo INPUT, donde ingresa nombre usuario y contraseña. Tiene la opción registrarse donde ingresara nombre y clave.

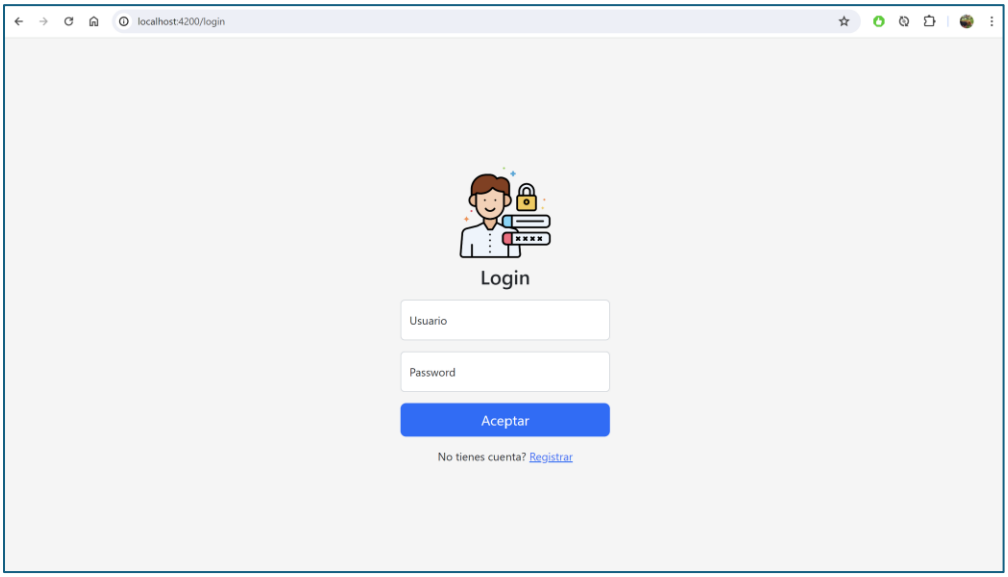


Figura 31.Interfaz login. (Elaboración propia)

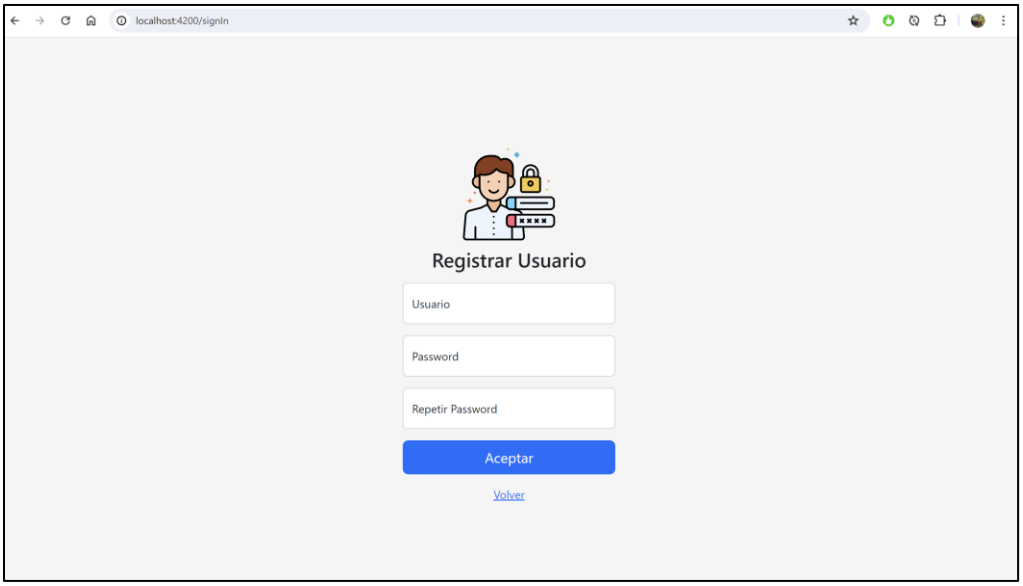


Figura 32. Interfaz registro. (Elaboración propia)

3.2.2 Vista información agrícola con mapa

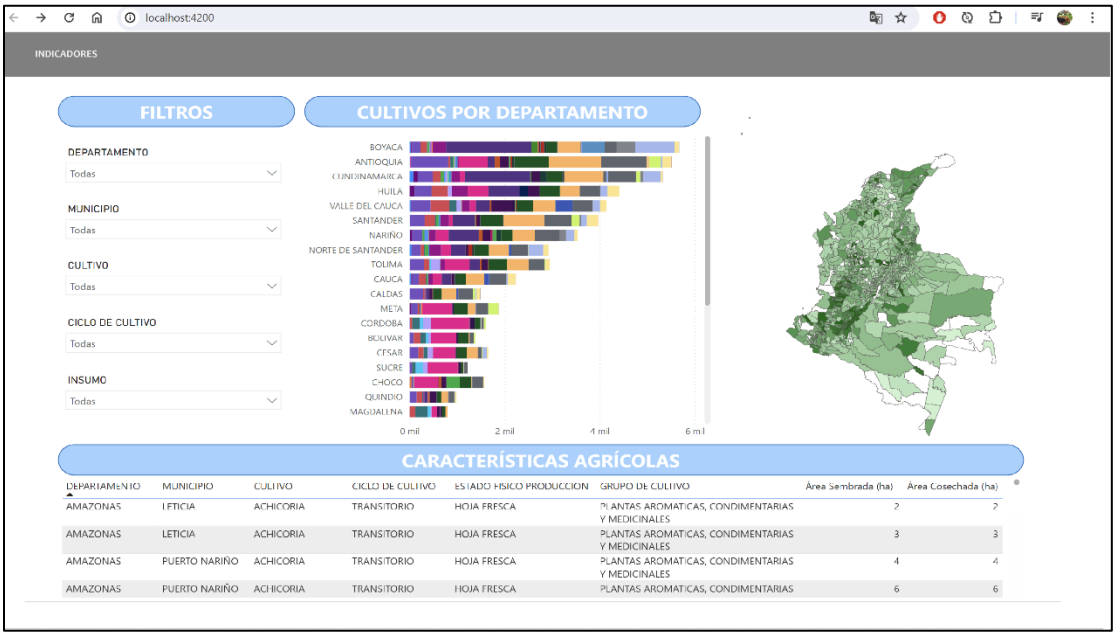


Figura 33. Vista información agrícola con mapa. (Elaboración propia)

3.2.3 Vista información agrícola almacenada en la base de datos.



Figura 34. Interfaz información agrícola con graficas. (Elaboración propia)

3.3. Microservicio Gestión de usuarios

Para la creación de un microservicio de autenticación para la aplicación realizada en Angular, se realizó un microservicio de autenticación independiente en Node.js, que interactúa con la aplicación frontend en Angular a través de una API REST. Para realizar esto se siguieron los pasos: 1. Configurar el Microservicio de Autenticación, 2. Configurar la Base de Datos, 3. Crear el Modelo de Usuario, 4. Crear el Controlador de Autenticación, 5. Definir las Rutas de Autenticación, 6. Configurar el Servidor, 7. Configurar Docker, 8. Desplegar el Microservicio, 9. Consumir el Microservicio desde la Aplicación Angular y 10. Desplegar en AWS.

3.3.1. Configurar el Microservicio de Autenticación

Para crear un microservicio de autenticación para la aplicación realizada en Angular, se realizó un microservicio independiente en Node.js, que interactúa con la aplicación frontend en Angular o través de una API REST.

Para lograr esto se creó una API REST, comando npm init -y `npm init -y`, creación archivo packege.json. Instalación de paquetes (npm install express, bcrypt 8encriptación contraseña), cors (determinado puerto), dotenv (variables de entorno), jsonwebtoken (permite crear el token), sequelize, mysql2) `npm install express bcrypt cors dotenv jsonwebtoken sequelize mysql2`. Instalación typescript,

localmente `npm install typescript --save-dev`. Instalación npm `npm i nodemon --save-dev`. Inicialización `npx tsc --init`.

Estructura del proyecto

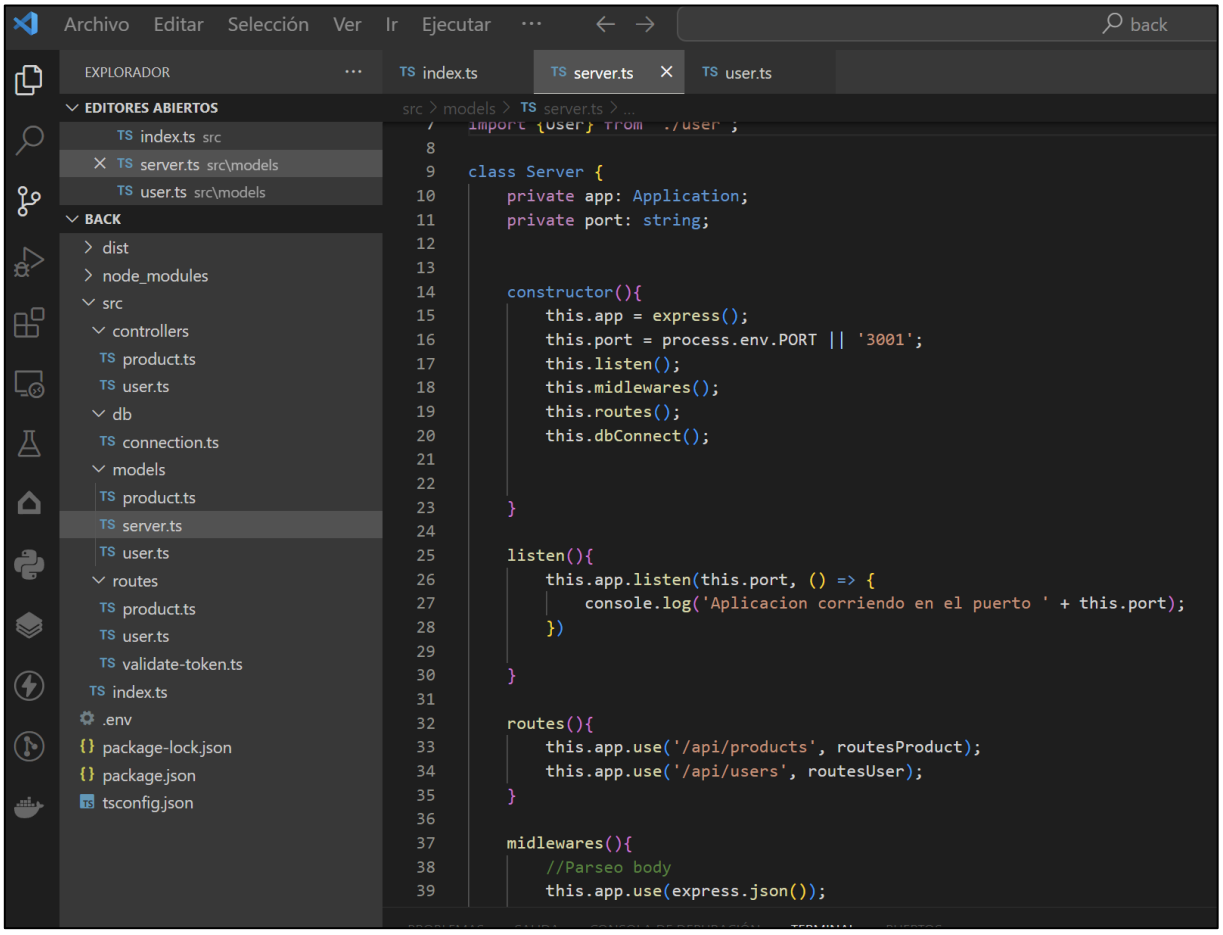


Figura 35. Carpetas Backend.

Creación el archivo ‘package.json’

Se debe instalar dependencias, comando npm install.

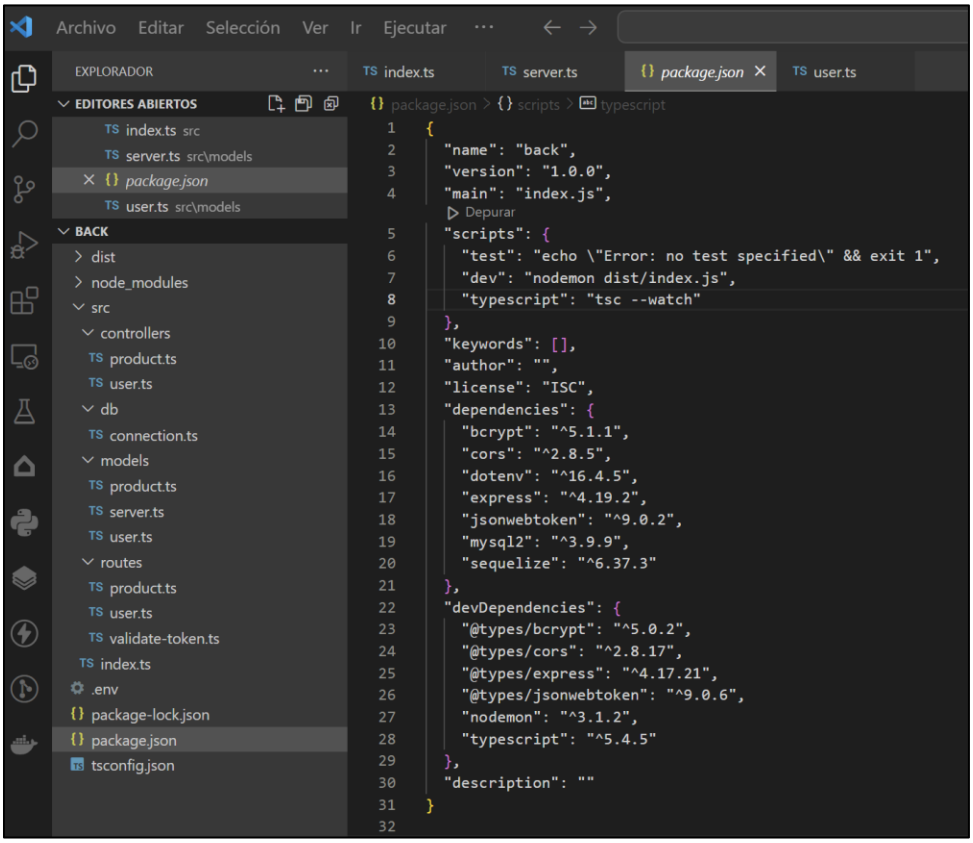


Figura 36. Creación package.json. (Elaboración propia)

3.3.2. Configuración de la Base de Datos

Se creó una carpeta para la conexión a la base de datos denominada db, la cual tiene un archivo connection.ts. Este archivo contiene los parámetros host, dialect, nombre de la base, usuario y clave.

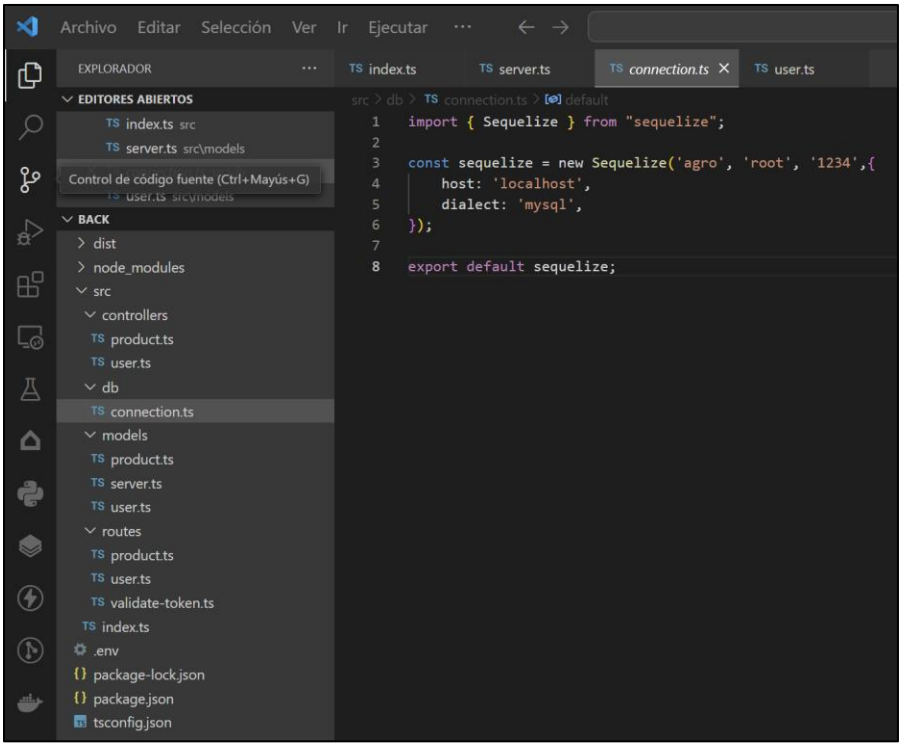


Figura 37.Configuración base de datos. (Elaboración propia)

3.3.3. Creación del Modelo de Usuario

Se creó un archivo ‘src/models/user.ts’, ‘src/models/product.ts’ y ‘src/models/server.ts’

El archivo user.ts, contiene los campos que se construirán en la tabla, su tipo y la conexión con el servidor.

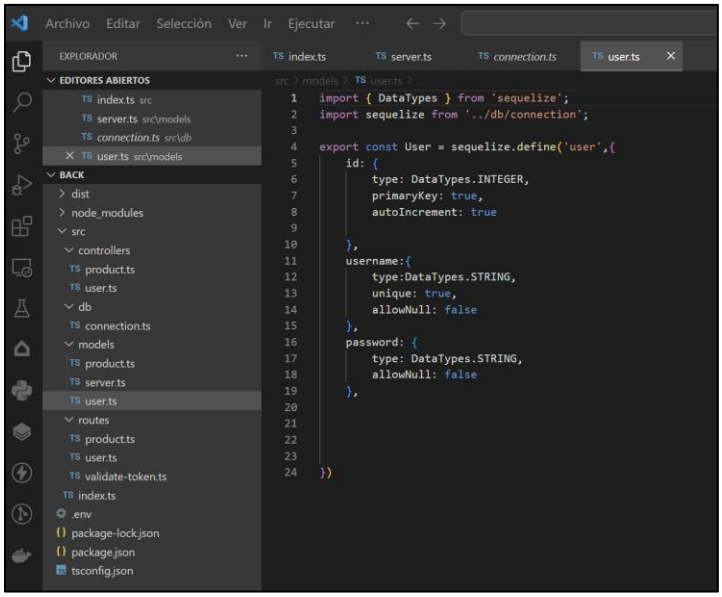


Figura 38. Creación Modelo de Usuario. (Elaboración propia)

El archivo product.ts, contiene los campos que se construirán en la tabla product, su tipo y la conexión con el servidor.

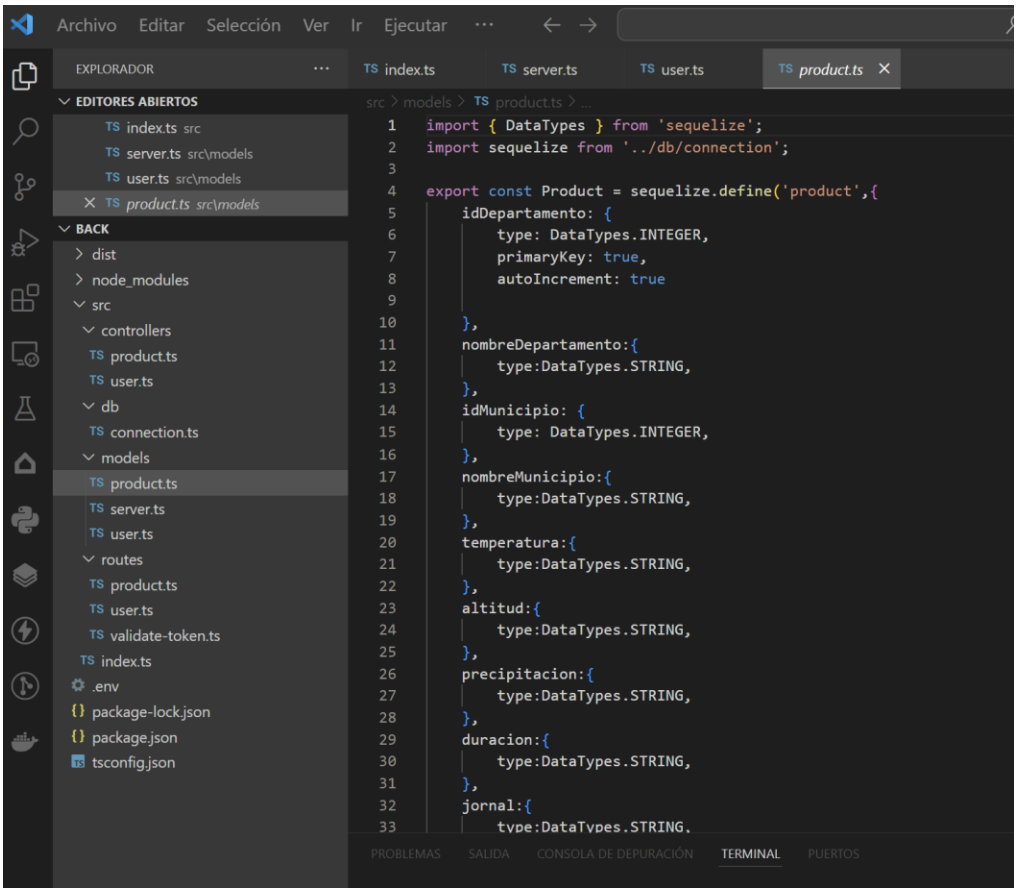


Figura 39. models product. (Elaboración propia).

3.3.4. Creación del Controlador de Usuarios

Se creó un archivo ‘src/controllers/user.ts’. Para el manejo de las solicitudes relacionadas con el usuario, como el registro, inicio de sesión, cerrar sesión, actualización de perfiles, entre otras. Nos proporciona un punto central para gestionar todas las operaciones relacionadas con los usuarios, lo que facilita la implementación de nuevas funcionalidades y el mantenimiento del código.

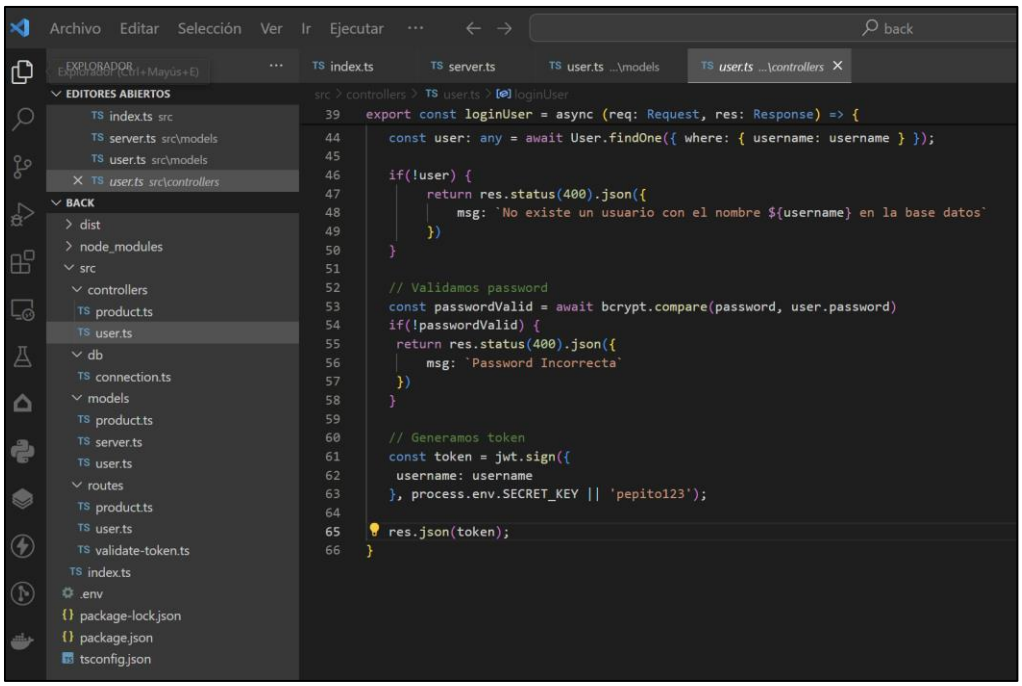


Figura 40.Controlador user. (Elaboración propia)

3.3.4. Creación del Controlador de Productos

Se creó un archivo 'src/controllers/product.ts'. Un controlador de productos permite gestionar todas las operaciones relacionadas con los productos. Un controlador de productos maneja las solicitudes HTTP relacionadas con la creación, lectura, actualización y eliminación de productos en una tienda en línea o cualquier otra aplicación que gestione un catálogo de productos.

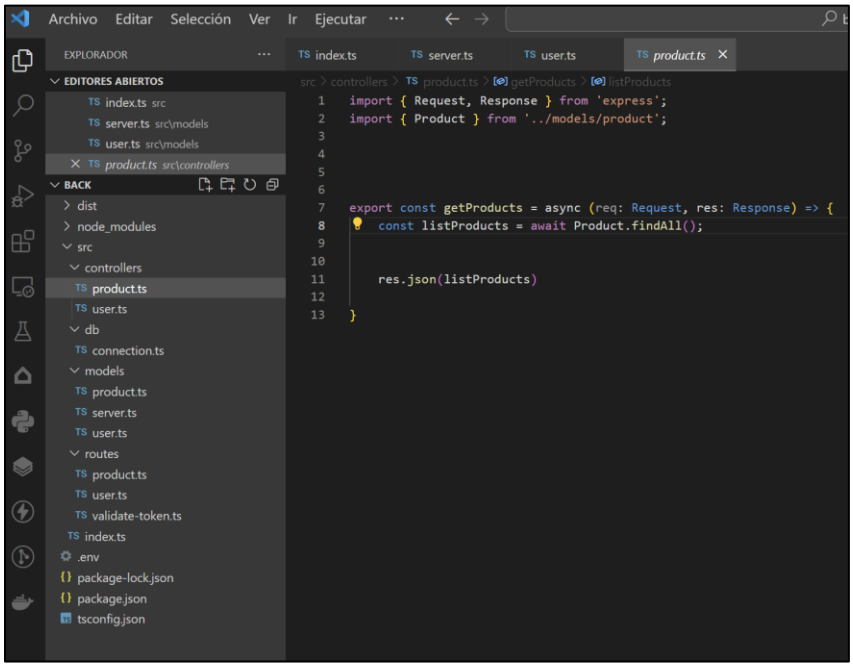


Figura 41. Controlador producto. (Elaboración propia)

3.3.5. Definición de las Rutas usuarios, productos y validación de tokes.

La creación de rutas para usuarios, productos y validación de tokens en una aplicación web se realizó para estructurar y gestionar adecuadamente las diferentes funcionalidades de la aplicación. A continuación, se explica el propósito de cada uno y su implementación en el proyecto.

Se creó un archivo ‘src/routes/user.ts’. Manejan operaciones relacionadas con el usuario como registro, inicio de sesión y perfil.

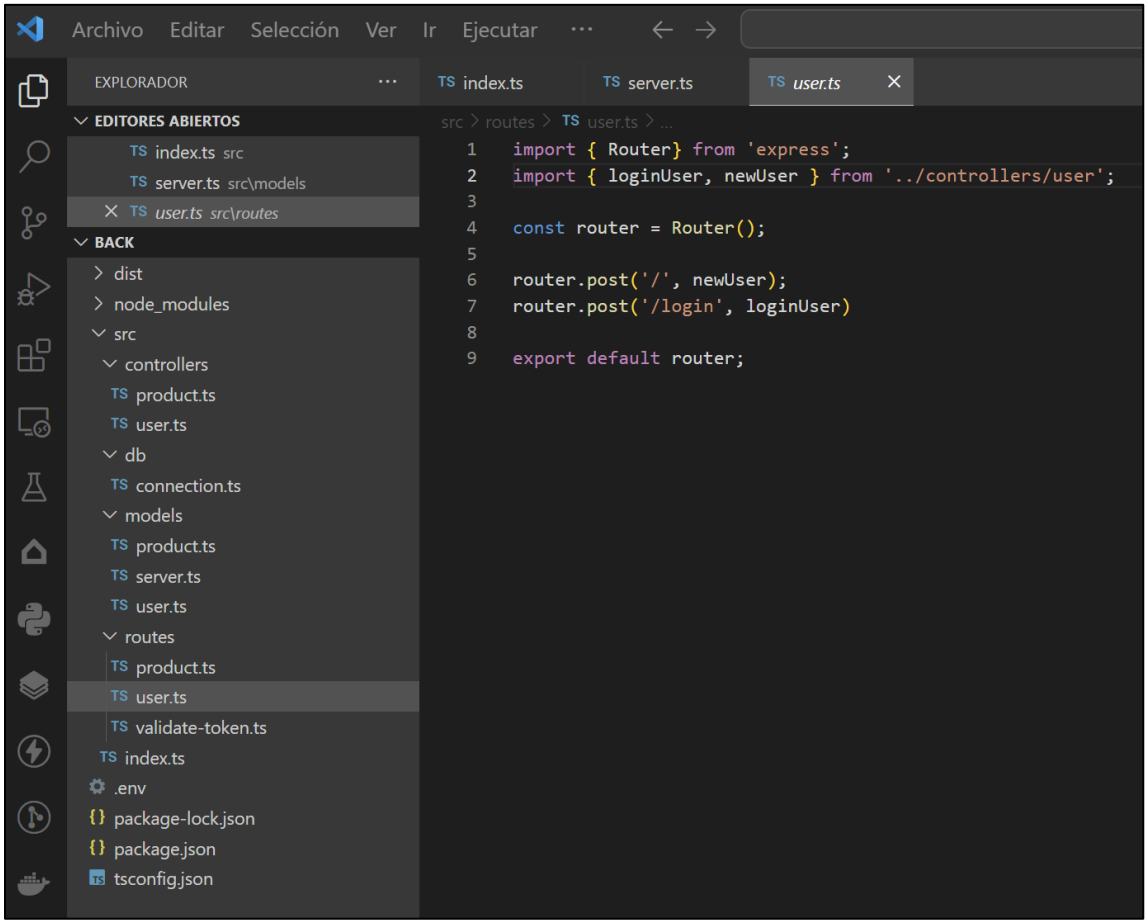


Figura 42. Ruta usuarios. (Elaboración propia)

Se creó un archivo ‘src/routes/product.ts’. Está gestionando operaciones CRUD para productos.

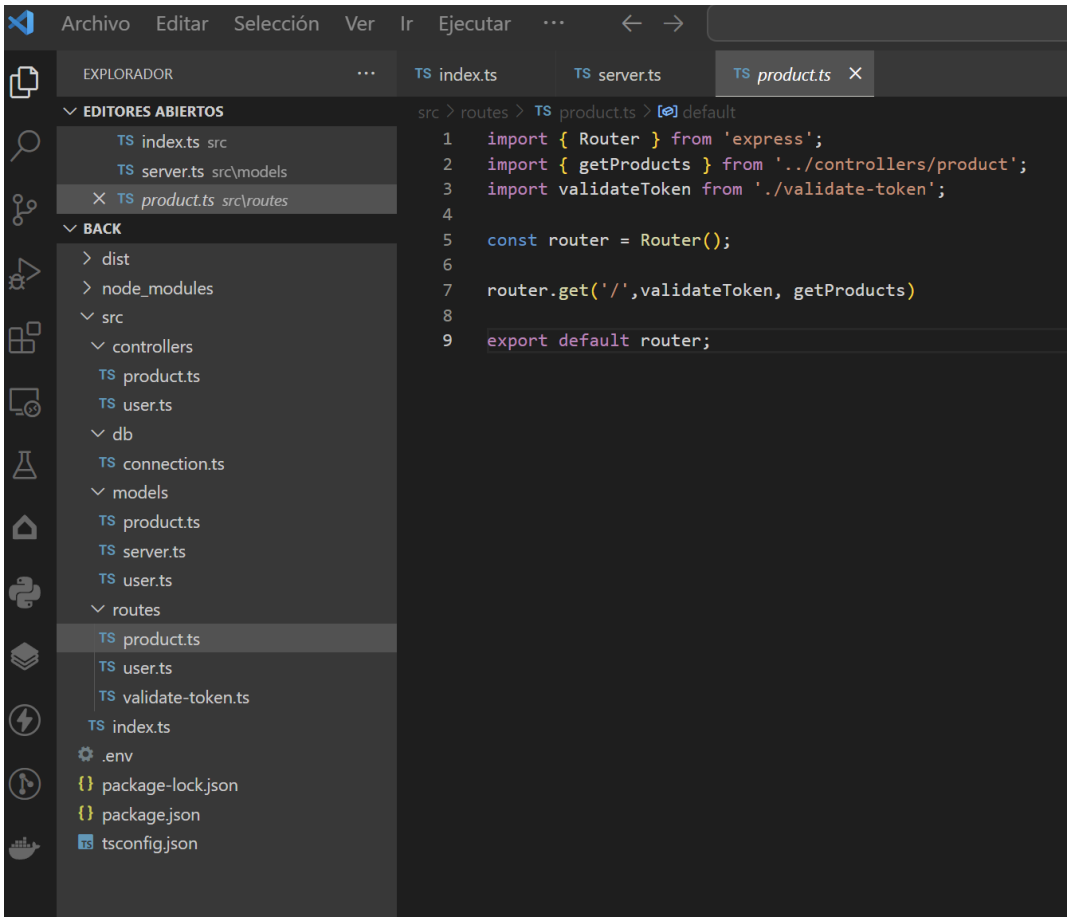


Figura 43. Ruta productos.

Esta ruta protege rutas críticas asegurando que solo los usuarios autenticados puedan acceder a ellas.

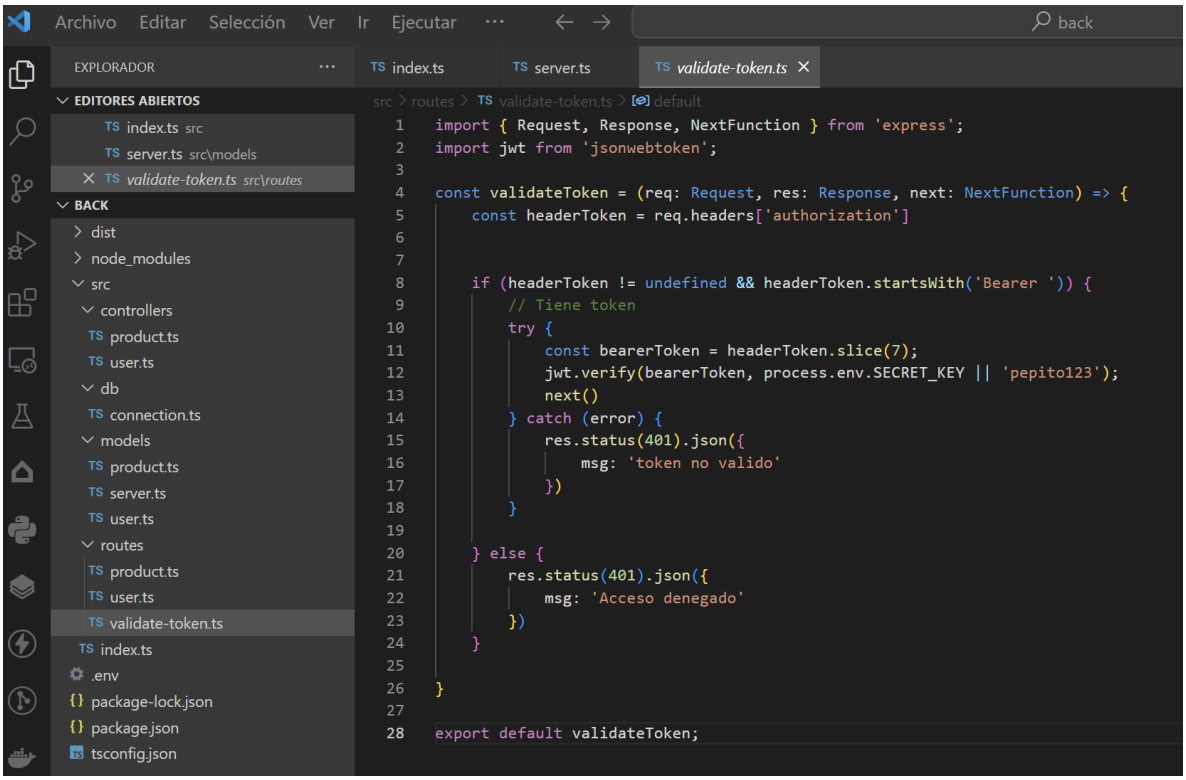


Figura 44. Ruta validación tokes.

3.3.6. Configurar modelo Servidor

El modelo de servidor es crucial para la gestión eficiente y segura de los datos en una aplicación. Proporciona una estructura clara para definir y validar datos, gestiona la lógica de negocio y las relaciones entre datos, y facilita la interacción con la base de datos a través de operaciones CRUD. Utilizando ORMs como Sequelize en Node.js, los desarrolladores pueden trabajar con datos de manera más intuitiva y menos propensa a errores, mejorando la mantenibilidad y escalabilidad de la aplicación.

Se creó un archivo ‘src/models/server.ts’

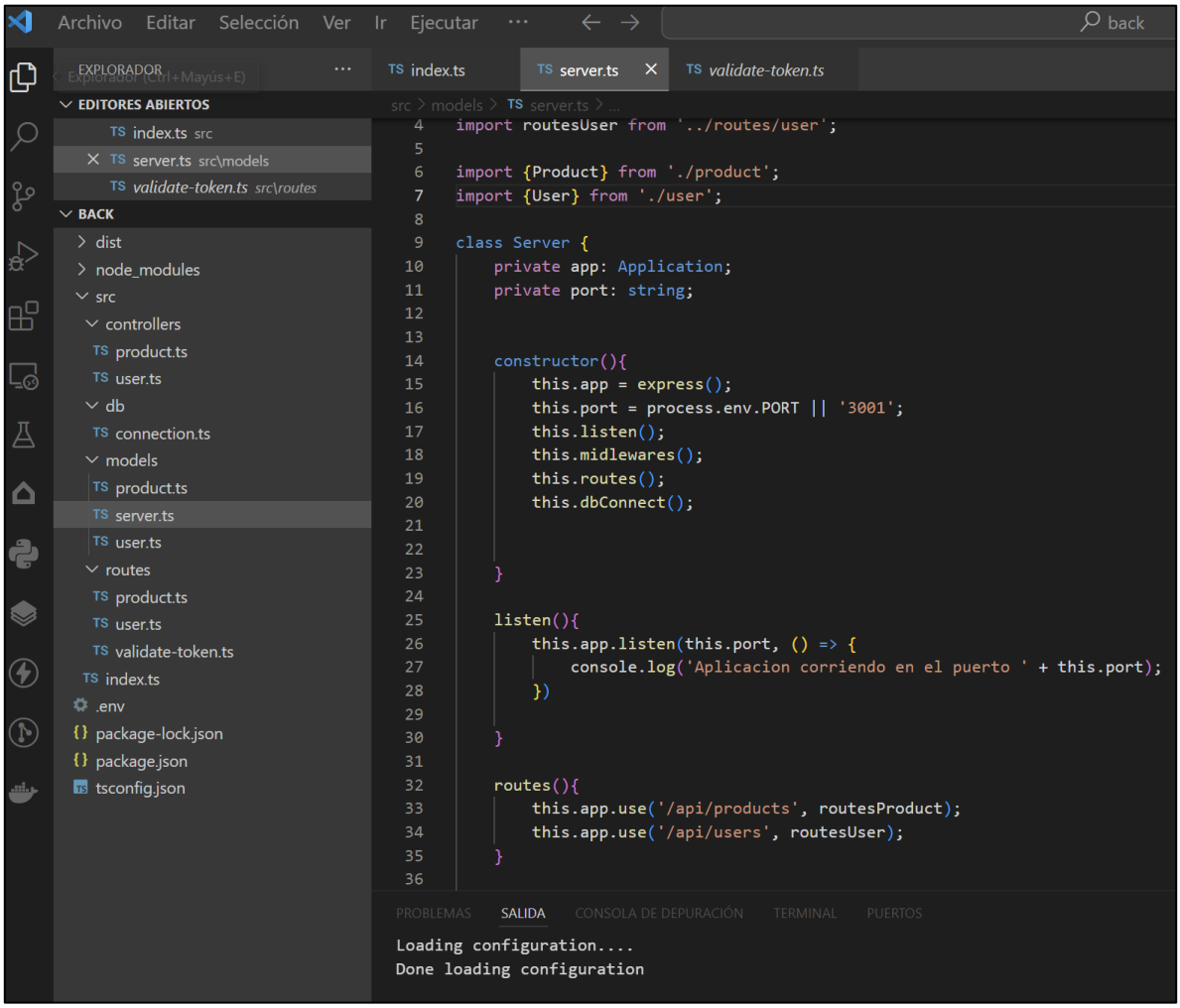


Figura 45. Configuración servidor. (Elaboración propia)

3.3.7. Configurar Docker

Se creó un archivo ‘Dockerfile’

```
FROM node:14

WORKDIR /app

COPY package*.json ./
RUN npm install

COPY . .

EXPOSE 3000
CMD ["node", "src/server.js"]
```

Figura 46. Configuración Docker. (Elaboración propia)

3.3.8. Desplegar el Microservicio

Con los siguientes comandos se construyó y ejecutó la imagen Docker.

```
docker build -t auth-service .
docker run -p 3000:3000 auth-service
```

Figura 47. Desplegar el Microservicio. (Elaboración propia)

3.3.9. Consumir el Microservicio desde la Aplicación Angular

Se creó un servicio en la aplicación Angular para interactuar con el microservicio de autenticación y productos.

Comando: ng generate service services/user

```
ng g s services/user
```

Posteriormente se implementó el servicio ‘user.service.ts’

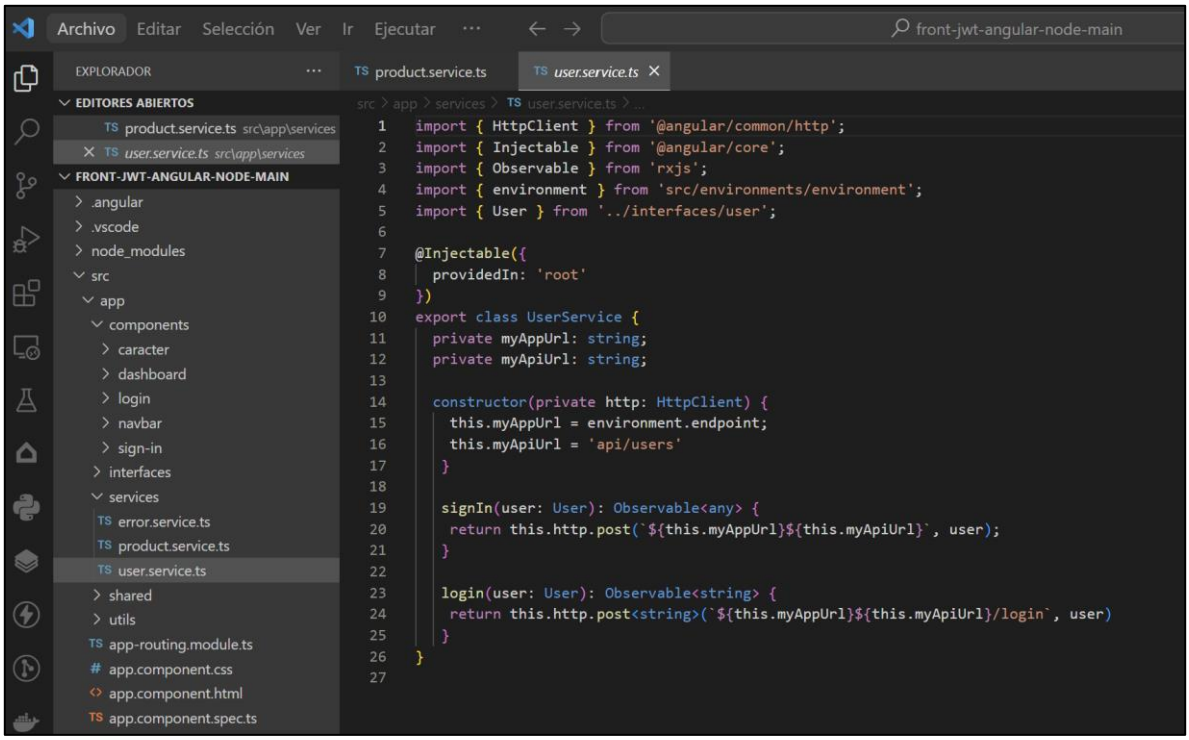


Figura 48. Configuración Microservicio. (Elaboración propia)

El archivo product.service.ts en Angular actúa como una capa de servicio para manejar todas las operaciones relacionadas con los productos, proporcionando una interfaz limpia y coherente para que los componentes interactúen con los datos. Esto no solo mejora la organización del código, sino que también facilita la reutilización y el mantenimiento de la lógica de negocio.

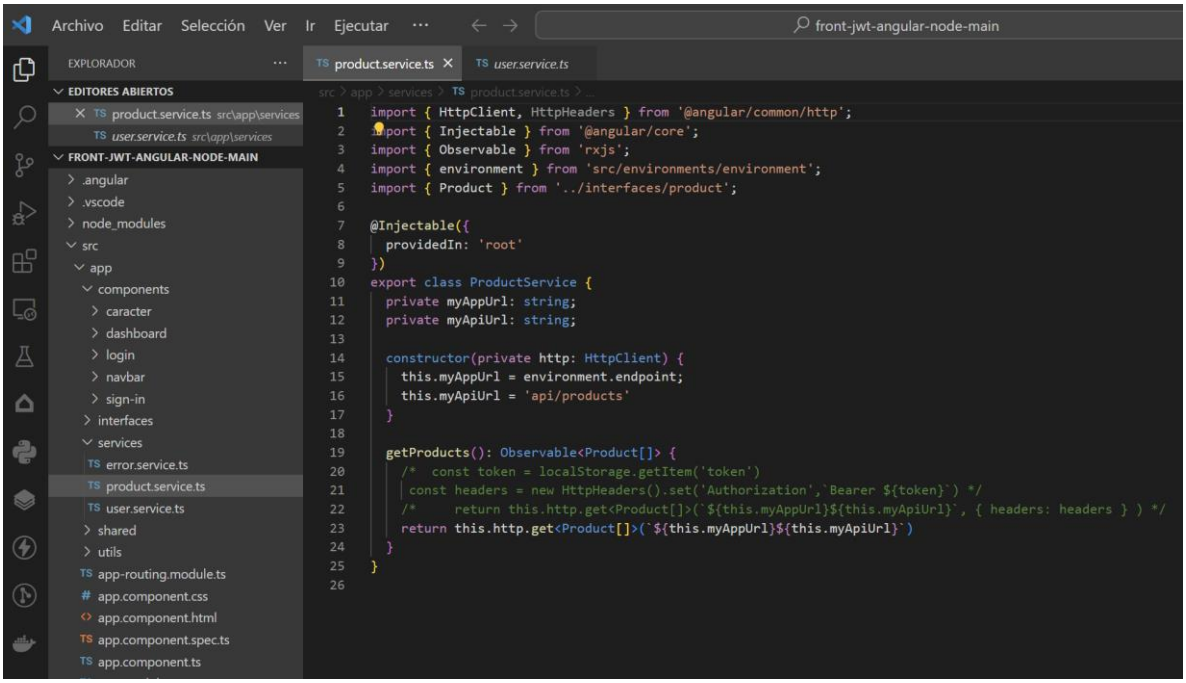


Figura 49. Servicio ‘product.serve.ts’. (Elaboración propia)

Este archivo se usa para representar los errores HTTP, para manejar errores de forma, como un servicio que puede ser inyectado en otro, manejo de los errores del lado del cliente, muestra una alerta con el mensaje de error del usuario y puede agregar lógica adicional para registrar el error en un servicio de registro.

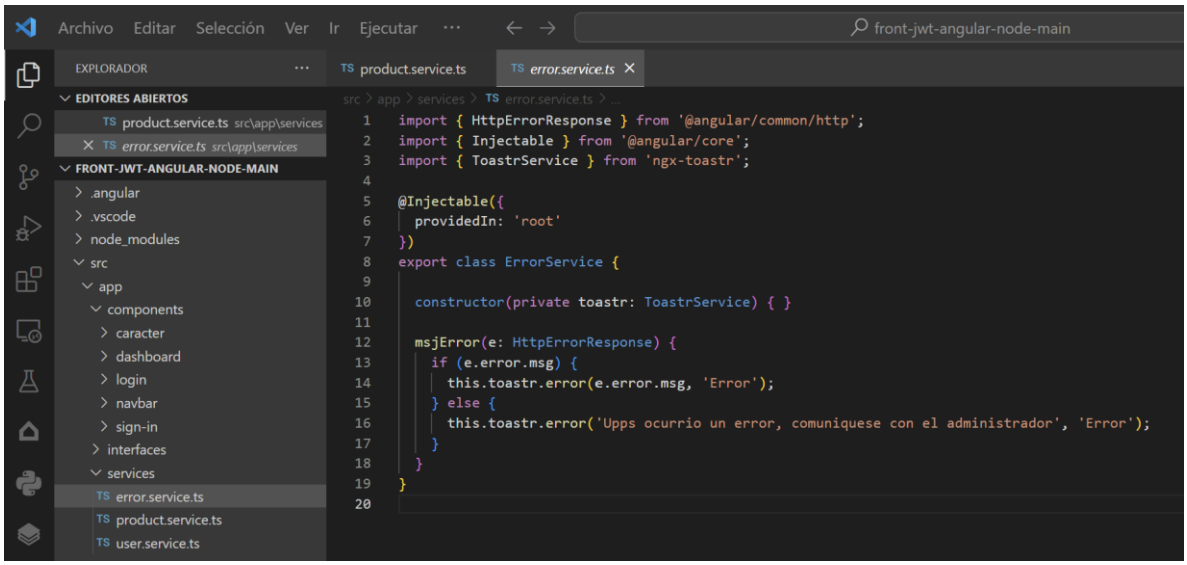


Figura 50.error.server.ts.(Elaboración propia)

3.3.10. Desplegar en AWS

Para desplegar un microservicio de autenticación Node.js junto con una aplicación frontend en Angular en AWS se relacionan varios servicios de AWS como Elastic Beanstalk, ECS (Elastic Container Service), RDS (Relational Database Service), y S3. A continuación describimos el proceso.

3.3.10.1 Desplegar el Microservicio de Autenticación en AWS Elastic Beanstalk

1. Configuración AWS Elastic Beanstalk CLI

Instalar CLI de Elastic Beasntalk, comando pip install awsebcli –upgrade

2. Inicializa el entorno de Elastic Beanstalk

Para esta actividad se usa el comando eb init, se debe seleccionar la región de AWS, la aplicación y Node.js como plataforma.

3. Se creó un entorno de Elastic Beanstalk y se desplegó la aplicación.

Se ejecuta los comandos: eb créate auth-service-env y eb desploy

4. Se crea un archivo ‘.ebextensions’

Se crea una carpeta .ebextensions en la raíz de tu proyecto y añade archivos de configuración como 01_environment.config para definir variables de entorno, ajustes del servidor, etc.

```
option_settings:
  aws:elasticbeanstalk:container:nodejs:
    NodeCommand: "node src/server.js"
  aws:elasticbeanstalk:application:environment:
    DATABASE_HOST: "tu-endpoint-rds"
    DATABASE_USER: "tu-usuario"
    DATABASE_PASSWORD: "tu-contraseña"
    DATABASE_NAME: "mi_base_de_datos"
    JWT_SECRET: "tu_jwt_secret_key"
```

Figura 51.Desplegar en AWS

3.3.10.2 Desplegar la aplicación Angular en AWS S3 y CloudFront

Primero se construye la aplicación en Angular para producción, con el comando: ng build –prod.

Luego se configura AWS S3. Se crea un bucket en S3 accediendo a la consola de S3 y crea se crea el bucket. Posteriormente se configura el bucket para que sea público y habilita el alojamiento estático de sitios web.

Después se suben los archivos de construcción a S3.

```
aws s3 sync ./dist/tu-aplicacion-angular s3://nombre-de-tu-bucket
```

Se crea una distribución de CloudFront. Configurando loudront para apuntar al bucket de S3 como origen y se configuran las políticas de caché según sea necesario.

3.3.10.3 Configuración la Base de Datos en AWS RDS

1.Crear una instancia de MySQL en RDS:

- Accede a la consola de RDS y crea una nueva instancia de MySQL.
- Configura la seguridad y los grupos de seguridad para permitir conexiones desde Elastic Beanstalk.

2.Configurar la conexión a la base de datos en el microservicio:

- Asegúrate de que las variables de entorno para la conexión a la base de datos estén correctamente configuradas en Elastic Beanstalk.

3.3.10.4 Actualizar le Frontend para consumir el Microservicio en AWS

Se debe actualizar ‘auth.service.ts’ en Angular para usar la URL del Microservicio

```
import { Injectable } from '@angular/core';
import { HttpClient, HttpHeaders } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class AuthService {
  private apiUrl = 'http://tu-endpoint-elasticbeanstalk/auth'; // URL del
microservicio desplegado en Elastic Beanstalk

  constructor(private http: HttpClient) { }

  register(user: { nombre_usuario: string; contrasena: string }):
Observable<any> {
    return this.http.post(`${this.apiUrl}/registrar`, user);
  }

  login(user: { nombre_usuario: string; contrasena: string }):
Observable<any> {
    return this.http.post(`${this.apiUrl}/login`, user);
  }

  verifyToken(token: string): Observable<any> {
    const headers = new HttpHeaders().set('Authorization', token);
    return this.http.get(`${this.apiUrl}/verificar`, { headers });
  }
}
```

Figura 52.alizar le Frontend para consumir el Microservicio en AWS

PARTE III.

CIERRE DE LA INVESTIGACIÓN

CAPÍTULO 4. RESULTADOS Y DISCUSIÓN

Este proyecto estuvo enfocado en determinar cómo se puede usar grandes volúmenes de datos y con procesos de analíticas de datos transformarlos, cargarlos con herramientas BI, para generar indicadores y resultados de la información.

Por medio de un aplicativo web utilizar estos reportes para caracterizar un sector en este caso el sector agrícola, y usarlos para generar conocimiento. El resultado más relevante para este proyecto fue mostrar el desarrollo de un aplicativo web con un dashboard incorporado y desarrollado en Power BI con grandes cantidades de información obtenida de datos abiertos.

CAPÍTULO 5. CONCLUSIONES

5.1. Verificación, contraste y evaluación de los objetivos.

Desde el objetivo “Analizar diferentes fuentes de información como documentos en pdf, Word, archivos Excel, entre otros, relacionados con el sector agropecuario en el país” se evidencio que utilizando la herramienta Power BI se está integrando las fuentes de información para realizar la integración y utilización de esta gráficamente.

El objetivo “Selección de software y elaboración del proceso ETL (Extract, Trasnform y Load)”, con la utilización de Power BI y la transformación y cargue, se cumplió con el objetivo.

Se elaboró un tablero de mando (Dashboard) en Power BI, el cual muestra la información por departamento, municipio, tipo de cultivo, insumos y características de cada uno.

Por último para la publicación se realizó un aplicativo web con su fronend, backend, microservicios y publicación en AWS.

5.2. Síntesis del modelo propuesto

La implementación de un aplicativo web que permita la visualización geográfica y alfanumérica de información del sector agrícola producto del análisis de datos realizado a diferentes fuentes de información empleado las herramientas big data para generar información relevante, la cual podrá ser consultada por varios usuarios por medio del aplicativo web desarrollado sin generar costos elevados, con restricciones de ingreso login, entorno visual agradable y con opciones de realizar filtros en la información.

5.3. Aportes originales

Por medio de un aplicativo web desarrollado a bajo costo mostrar a los usuarios indicadores del sector agrícola construidos con grandes volúmenes de información con herramientas de análisis de datos, sin instalación de software y visualizaciones interactivas.

CAPÍTULO 6. PROSPECTIVA DEL TRABAJO DE GRADO

6.1. Alcances y limitaciones

Por medio del desarrollo de un aplicativo en el cual se visualice información determinante de un sector en el caso puntual la Agricultura es valioso para ser utilizado en caracterización y clasificación de precios y entorno. Una de las limitantes en este proyecto fue el tiempo para la realización del proyecto por ser un tema muy amplio de abarcar.

6.2. Líneas de investigación futuras

- Integración de módulos para recoger datos en campo de cultivos.
- Responsive para celular.

6.3. Trabajos de Investigación futuros

Incorporar en la aplicación la opción de ingresar información como tipo de cultivo, coordenadas latitud y longitud, clase, área, entre otras variables para crear nueva información con la existente y poder generar nuevos reportes.

Integración dinámicamente de cultivos con información relevante para la estimación de cuánto puede costar un cultivo comercialmente. Interesados entidades del estado, reforma agraria.

REFERENCIAS WEB

Analítica de datos con Power BI

https://app.powerbi.com/view?r=eyJrIjoiNjlkMjA5MDktMTg2MC00MjI1LTg4ZWItZDZjMmRlNzlkYzVlIiwidCI6ImQ1MTM4OGVmLTZhYjAtNDM2My05Zjk0LWQ1NjY0NGE0NTk3MCIsImMiOjR9

BIBLIOGRAFÍA

Asana, T. (2024). *Team Asana*. Obtenido de https://asana.com/es/resources/project-management-methodologies

AWS. (2024). *AWS*. Obtenido de https://aws.amazon.com/es/what-is/sql/#:~:text=es%20importante%20SQL%3F-,El%20lenguaje%20de%20consulta%20estructurada%20(SQL)%20es%20un%20lenguaje%20de,los%20diferentes%20lenguajes%20de%20programaci%C3%B3n.

AWS. (s.f.). *AWS*. Obtenido de https://aws.amazon.com/es/what-is/hadoop/

Cloud, G. (s.f.). *Google Cloud*. Obtenido de https://cloud.google.com/discover/what-is-nosql?hl=es-419#:~:text=El%20t%C3%A9rmino%20NoSQL%2C%20que%20significa,las%20bases%20de%20datos%20relacionales.

concepto. (s.f.). *concepto*. Obtenido de https://concepto.de/dato/#:~:text=Dato%20en%20inform%C3%A1tica%20es%20la,partir%20de%20los%20cuales%20trabaja.

Datamedia. (2024). *Datamedia*. Obtenido de https://datademia.es/blog/que-es-un-dashboard-o-cuadro-de-mando

Fernández, Y. (18 de Enero de 2023). *Xataka Basics*. Obtenido de https://www.xataka.com/basics/que-crm-sirve-este-tipo-programas

Gil, D. A. (2021). *Adictos al trabajo* . Obtenido de https://www.adictosaltrabajo.com/2021/01/14/metodologia-crisp-dm/

Heeg, R. (s.f.). *elatic*. Obtenido de https://www.elastic.co/es/what-is/unstructured-data

IBM. (2021). *IBM*. Obtenido de https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview

IBM. (2024). *IBM*. Obtenido de https://www.ibm.com/es-es/topics/business-intelligence

IBM. (2024). *IBM*. Obtenido de https://www.ibm.com/analytics/big-data-analytics

IEBS. (2024). *IEBS*. Obtenido de https://www.iebschool.com/blog/que-son-los-procesos-etl-big-data/

Niño, M. (2020). *Blog de Mikel Niño, Industria 4.0, Biga Data Analytics, emprendimiento digital y nuevos negocios*. Obtenido de https://www.mikelnino.com/2015/09/crisp-dm-metodologia-proyectos-data-mining.html

Obiols, A. (s.f.). *inLabFIB*. Obtenido de https://inlab.fib.upc.edu/es/articulos/que-es-un-data-scientist-2/2015/

PowerData. (2024). *PowerData*. Obtenido de https://www.powerdata.es/data-warehouse#:~:text=Un%20Data%20Warehouse%20es%20un,recuperar%20y%20f%C3%A1cil%20de%20administrar.

SAS. (2023). *SAS*. Obtenido de https://www.sas.com/es_co/insights/data-management/what-is-etl.html