

FACULTAD DE INGENIERÍA CIVIL

**INTELIGENCIA ARTIFICIAL APLICADA EN LA PRIORIZACIÓN DE LA
ELECCIÓN DE FUENTES DE MATERIALES EN EL MEJORAMIENTO DE LA RED
VIAL TERCIARIA. ESTUDIO DE CASO EN EL DEPARTAMENTO DEL CAUCA
(COLOMBIA)**

**PROYECTO DE GRADO PRESENTADO A LA FACULTAD DE INGENIERÍA
CIVIL DE LA UNIVERSIDAD DEL CAUCA PARA LA OBTENCIÓN DEL TÍTULO DE**

INGENIERO CIVIL

PRESENTADO POR:

IVAN FELIPE ERAZO CERON

ASESORES:

PhD. CARLOS ANIBAL CALERO

PhD. FERNEY QUIÑONES SINIESTIERRA

POPAYAN, COLOMBIA 2023

PROYECTO DE TESIS PARA LA OBTENCIÓN DEL TÍTULO DE

INGENIERO CIVIL

PRESENTADO POR:

IVAN FELIPE ERAZO CERON

ASESORES:

PhD. CARLOS ANIBAL CALERO

PhD. FERNEY QUIÑONES SINIESTIERRA

UNIVERSIDAD DEL CAUCA

FACULTAD DE INGENIERÍA CIVIL

DEDICATORIA

A mis padres, quienes me han apoyado en todo momento, a mi familia por su amor incondicional. A mi couple por su paciencia y comprensión durante todo el proceso de investigación. A mi director de tesis, por su guía y enseñanzas que me han permitido crecer como profesional. Y a todas las personas que han sido parte de este camino, gracias por ser parte de esta gran aventura.

AGRADECIMIENTOS

Gratitud a la Universidad del Cauca que me ha permitido recibir una excelente formación, a sus miembros por su calidez humana, en especial a los profesores de la facultada de Ingeniería Civil con quienes he compartido en este transcurrir de enseñanzas. Quiero expresar mis agradecimientos a:

El profesor Carlos Aníbal Calero por su disposición, paciencia y todos sus valiosos comentarios que hicieron posible la culminación de este documento, sus consejos y ayuda en mi formación profesional, sus valiosas enseñanzas e incentivarme a seguir el camino de la investigación.

El profesor Ferney Quiñones Siniesterra por sus aportes y comentarios que han logrado ser parte de la culminación de este proyecto de investigación.

Mis familiares, que siempre estuvieron apoyándome con sus palabras y oraciones en mi caminar por la Universidad.

PROGRAMA DE INGENIERÍA CIVIL

POPAYAN, COLOMBIA

2023

Contenido

Capítulo 1. Introducción.....	3
1.1. Problema	4
1.2. Justificación.....	9
1.3. Objetivos	11
1.3.1. Objetivo General	12
1.3.2. Objetivos Específicos:.....	12
1.4. Beneficios del proyecto.....	12
1.5. Organización	13
Capítulo 2. Revisión de literatura	16
2.1. Lenguaje de programación	16
2.2. Ruta mínima.....	17
2.2.1. Algoritmo de Dijkstra	18
2.3. REDES NEURONALES.....	30
2.3.1. EL PERCEPTRÓN	35
2.3.2. PERCEPTRÓN MULTICAPA	38
Capítulo 3. Metodología	52
3.1. RECOLECCIÓN DE DATOS PARA CONSTRUCCIÓN DE LOS SOFTWARES	52
3.2. CONSTRUCCIÓN DE GRAFO	56
3.2.1. Nodos para el análisis.....	57
3.2.2. Enlaces.....	59
3.2.3. Resultado	63
3.3. RUTA MÍNIMA PARA ACARREO DE MATERIALES.....	71
3.3.1. Desarrollo del algoritmo de ruta mínima.....	74

3.4. Desarrollo de las Redes Neuronales Artificiales	77
3.4.2. Búsquedas de las fuentes de material a partir de los requerimientos ...	87
Capítulo 4. Toma de datos	91
4.1. Registros en campo	91
4.2. Funcionalidades de las aplicaciones: Canteras-Móvil y Canteras-Web.....	94
4.2.1. Canteras-Móvil	94
4.2.2. Canteras-Web	97
4.3. Base de datos inicial para CANTERAS-MÓVIL Y CANTERAS-WEB	101
Capítulo 5. Conclusiones, recomendaciones y trabajo futuro	104
5.1. Conclusiones y recomendaciones.....	104
5.2. Trabajo Futuro.....	106
Capítulo 6. Bibliografía	109
Capítulo 7. Anexos	110

TABLA DE FIGURAS

MAPA DE RED VIAL PARA COLOMBIA.....	4
COMPOSICIÓN DE RED VIAL PARA COLOMBIA.	5
ESTADO DE LA RED VIAL EN EL DEPARTAMENTO DEL CAUCA.	5
TIPO DE SUPERFICIE DE RODADURA EN LAS VÍAS TERCARIAS DEL DEPARTAMENTO DEL CAUCA.	6
COMPARACIÓN DE LA POBLACIÓN RURAL RESPECTO A SU INVERSIÓN NACIONAL.	7
SENSIBILIDAD DE LAS ACTIVIDADES EN LA DURACIÓN DEL PROYECTO.....	8
COEFICIENTE DE CORRELACIÓN JERÁRQUICA DE DURACIÓN DEL PROYECTO.....	8
TOTAL DE FUENTES DE EXTRACCIÓN PARA EL DEPARTAMENTO DEL CAUCA EN LAS ZONAS: NORTE, SUR, CENTRO Y ORIENTE EN SOFTWARE "CANTERAS MÓVIL".....	10
MAPA DE DISTRIBUCIÓN DE FUENTES DE EXTRACCIÓN. EN ROSADO LAS CAPTURADAS POR LA OFICINA DE INFRAESTRUCTURA DE LA GOBERNACIÓN DEL CAUCA Y EN AZUL LAS FUENTES DE EXTRACCIÓN CAPTURADAS POR EL CDT VIAL CON: "CANTERAS MÓVIL".....	10
ORGANIZACIÓN.....	14
GRAFO DE ANÁLISIS.....	18
CAMINO MÍNIMO EN GRAFOS.	18
GRAFO PONDERADO.	19
DIJKSTRA.....	21
ITERACIÓN 1 PARA EL GRAFO.....	22
SEGUNDA ITERACIÓN.....	23
TERCERA ITERACIÓN.....	23
CUARTA ITERACIÓN.....	24
QUINTA ITERACIÓN.....	25
SÉPTIMA ITERACIÓN.....	26
OCTAVA ITERACIÓN.....	27
NOVENA ITERACIÓN.....	27
DECIMO PRIMERA ITERACIÓN.....	28
ANÁLISIS PARA DECIMA SEGUNDA ITERACIÓN.....	29
DECIMA SEGUNDA ITERACIÓN.....	30
NEURONA BIOLÓGICA.....	31
COMPORTAMIENTO DE UNA NEURONA.....	31
ESQUEMA COMPARATIVO DE RED NEURONAL BIOLÓGICA VS NEURONA ARTIFICIAL.....	32
MODELO NEURONAL DE McCULLOCH-PITTS.....	33
ALGUNAS FUNCIONES DE TRANSFERENCIA UTILIZADAS.....	35

PROBLEMAS DE CLASIFICACIÓN LINEALMENTE SEPARABLES.	37
PUERTA LÓGICA XOR, IMPOSIBLE DE RESOLVER CON 1 NEURONA.	38
ARQUITECTURA DE UN PERCEPTRÓN MULTICAPA CON UNA CAPA OCULTA.	39
EXPLICACIÓN SOBRE MLP COMO APROXIMADORES DE FUNCIONES.	39
NEURONA SIN FUNCIÓN DE ACTIVACIÓN.....	40
NEURONA QUE PASA POR LA FUNCIÓN DE ACTIVACIÓN SIGMOIDE.....	40
DESCENSO DEL GRADIENTE EN UNA RED CON ÚNICAMENTE 2 PESOS.	43
EJEMPLO DE ESTRUCTURA DE RED NEURONAL.	44
EJEMPLO DE ESTRUCTURA DE RED NEURONAL.	45
FUNCIÓN ESCALÓN.	49
FUNCIÓN LOGÍSTICA.....	50
FUNCIÓN TANH.	51
FUNCIÓN RELU.	51
DATOS CAPTURADOS CON CANTERAS-MÓVIL Y VISUALIZADOS EN CANTERAS-WEB.	54
VISUALIZACIÓN DE FRANJAS GRANULOMÉTRICAS DE LAS RESPECTIVAS FUENTES DE EXTRACCIÓN EN CANTERAS-WEB.....	55
DATOS DE FUENTES DE EXTRACCIÓN EN POSTGRESQL.	56
INFORMACIÓN PROCESADA PARA CONSTRUCCIÓN DEL GRAFO PARA VISITAS TÉCNICAS A LAS FUENTES.	57
MAPA DE FUENTES DE EXTRACCIÓN EN EL SOFTWARE CANTERAS-MÓVIL, PARA CONSTRUCCIÓN DE GRAFO.....	59
EJES VIALES DE VÍAS RURALES EN EL DEPARTAMENTO DEL CAUCA.	60
GRAFO PARA ANÁLISIS DE RUTA MÍNIMA.	61
ESQUEMA DE GRAFO.....	62
GRAFO RESULTANTE EN EL SOFTWARE CANTERAS-WEB	63
PROCESO PARA ENCONTRAR LA RUTA MÍNIMA CON EL SOFTWARE CANTERAS-WEB	64
RUTA MÍNIMA EN EL SOFTWARE CANTERAS-WEB.....	65
FUENTES DE EXTRACCIÓN EN EL SOFTWARE CANTERAS-MÓVIL.....	66
OPCIÓN PARA DETERMINAR LA RUTA MÁS CORTA ENTRE LA FUENTE Y EL PUNTO GEORREFERENCIADO EN CANTERAS-MÓVIL.	66
RUTA MÍNIMA GENERADA EN GOOGLE MAPS A TRAVÉS DEL SOFTWARE CANTERAS-MÓVIL.....	67
RECORRIDO DE LA RUTA MÍNIMA ENTRE LOS NODOS DE INICIO Y FIN GENERADO POR CANTERAS-WEB.....	70
RUTA MÍNIMA PARA ACARREO DE MATERIALES GENERADO EN CANTERAS-MÓVIL.....	72
ESQUEMA DE GRAFO PARA EL ANÁLISIS DE VISITAS TÉCNICAS A LAS FUENTES DE EXTRACCIÓN.	73
GRAFO NO DIRIGIDO CONSTRUIDO A PARTIR DE LA INVESTIGACIÓN DE LA INFORMACIÓN CAPTURADA Y SUMINISTRADA INICIALMENTE.	75
PORTE DE CÓDIGO DE LA CONSTRUCCIÓN DEL GRAFO Y OBTENER LA RUTA MÍNIMA PARA TIEMPOS DE VISITA EN CANTERAS-MÓVIL	76
CÓDIGO EN PYTHON PARA MOSTRAR EL GRAFO.	77

BASE DE DATOS PARA EL ENTRENAMIENTO DE LAS REDES NEURONALES.....	78
CREACIÓN DEL MODELO DE RED NEURONAL. CAPA DE ENTRADA CON 3 CARACTERÍSTICAS, UNA CAPA OCULTA CON 64 NEURONAS Y UNA CAPA DE SALIDA.	79
ANÁLISIS DE REQUISITOS X1, X2 Y X3 SEGÚN ARTÍCULO 311-13 INVIAS	83
BASE DE DATOS FINAL PARA EL ENTRENAMIENTO DEL MODELO DE RED NEURONAL.	84
ESTRUCTURACIÓN PARA DESARROLLAR EL MODELO DE RED NEURONAL.....	86
PARTE DEL CÓDIGO PARA ENTRENAR EL MODELO DE RED NEURONAL.	86
PARTE DEL CÓDIGO PARA ENTRENAR EL MODELO DE RED NEURONAL.....	87
ENTRADA DE DATOS PARA GENERAR LA PREDICCIÓN.....	88
RESULTADO A PARTIR DE LOS DATOS DE ENTRADA.	88
RESULTADOS DE LA PREDICCIÓN DEL MODELO PARA LAS FUENTES DE EXTRACCIÓN CANDIDATAS A SER EXPLOTADAS.	89
GRÁFICA DE CBR.	90
GRÁFICO DE DUREZA.....	90
GRANULOMETRÍA DE LA FUENTE GENERADA POR EL MODELO DE PREDICCIÓN.	91
FORMULARIO PARA CAPTURA DE INFORMACIÓN DE UNA FUENTE DE EXTRACCIÓN EN CAMPO MEDIANTE CANTERAS-MÓVIL.	92
INTERFAZ INICIAL DEL SOFTWARE CANTERAS-MÓVIL.	93
CAPTURA DE DATOS EN CAMPO MEDIANTE CANTERAS-MÓVIL.	93
VISUALIZACIÓN DEL REPORTE ADJUNTANDO. EN ESTE CASO REPORTE REALIZADO POR CDT VIAL PARA LAS FUENTES DE EXTRACCIÓN EN: EL TAMBO CAUCA	94
OPCIONES DE CAPTURA DE DATOS EN CAMPO EN CANTERAS-MÓVIL.....	95
ESTADÍSTICAS DE LOS DATOS REGISTRADOS. CANTERAS-MÓVIL	96
GRÁFICO: DISTANCIAS DE LA FUENTE DE EXTRACCIÓN A LA CABECERA MUNICIPAL VS TIEMPO DE RECORRIDO A LA CABECERA MUNICIPAL. CANTERAS-MÓVIL	96
ESTADÍSTICA DE ENSAYOS PARA LAS FUENTES DE EXTRACCIÓN.	97
IMÁGENES SATELITALES DE LAS FUENTES DE EXTRACCIÓN.....	98
FRANJAS GRANULOMÉTRICAS DE LAS DIFERENTES FUENTES DE EXTRACCIÓN.	99
IMPLEMENTACIÓN DEL ALGORITMO DE DIJKSTRA.	99
IMPLEMENTACIÓN DE REDES NEURONALES PARA DETERMINAR POSIBLES FUENTES DE EXTRACCIÓN A EXPLOTAR QUE CUMPLA CON E-INVIAS.	100
CONFORMACIÓN DE LA BASE DE DATOS INICIAL. PROCESAMIENTO EN ACCESS.	101
FORMACIÓN DE LA ESTRUCTURA DE DATOS GEOESPACIALES EN ARCMAP.	102
BASE DE DATOS VISTA EN CANTERAS-MÓVIL.	102
BASE DE DATOS INICIAL SE VISUALIZA DATOS GEOESPACIALES EN COLOR ROSA.	103
BASE DE DATOS REALIZADA CON LA INFORMACIÓN CAPTURA POR EL CDT VIAL DE LA UNIVERSIDAD DEL CAUCA.	104

REFERENCIA DE TABLAS

TABLA 1: REQUISITOS DE LOS AGREGADOS PARA AFIRMADOS.....	53
TABLA 2: FRANJAS GRANULOMÉTRICAS DEL MATERIAL DE AFIRMADO.....	53
TABLA 3: DATOS PARA GRAFO	58
TABLA 4: RESULTADOS SOBRE CANTERAS-MÓVIL VS CANTERAS-WEB.	71

RESUMEN

La presente investigación tiene como finalidad desarrollar una herramienta computacional que permita determinar las posibles fuentes de extracción a ser explotadas en la construcción de afirmados en vías terciarias. Esta herramienta no solo busca identificar una o más fuentes de extracción candidatas, sino también optimizar los tiempos de acarreo mediante la generación de rutas cortas y proporcionar información sobre los tiempos de visita a múltiples fuentes de extracción basándose en las rutas mínimas entre un nodo de inicio y un nodo final. En este caso, los nodos corresponden a puntos georreferenciados asociados a las fuentes de extracción que serán evaluadas mediante la herramienta computacional desarrollada.

El objetivo principal de esta investigación es implementar inteligencia artificial utilizando Redes Neuronales Artificiales (Artificial Neural Networks) a partir de la información recolectada en campo utilizando un modelo de software enfocado en la captura de datos de campo y laboratorio. Las Redes Neuronales Artificiales se integrarán dentro del software web desarrollado.

La búsqueda de información acerca del suministro de materiales en la construcción de subbases, bases y afirmados ha permitido evidenciar sobrecostos y mayores tiempos de ejecución debido a la incertidumbre de las fuentes de materiales a ser explotadas, ya sea o por qué no se encuentran fuentes de extracción cercanas o por qué no se tiene confiabilidad del material que cumplan con los estándares de calidad exigidos por INVIAS.

Esta metodología de búsqueda de materiales, basada en el uso de técnicas de inteligencia artificial como las redes neuronales, proporciona una base para identificar una o varias fuentes de extracción potenciales que cumplan con los estándares de calidad exigidos por el Instituto Nacional de Vías para los afirmados. Esta metodología ayudará en la planificación de proyectos viales y permitirá reducir el tiempo y los costos asociados a la incertidumbre en el suministro de materiales desde una fuente de extracción.

En resumen, esta investigación combina la aplicación de técnicas de inteligencia artificial, específicamente las Redes Neuronales Artificiales, con un software web y software-móvil diseñados para la captura y análisis de datos en campo y laboratorio. El enfoque propuesto proporciona una solución eficiente y confiable para la selección de fuentes de extracción en la construcción de afirmados en vías terciarias, mejorando la planificación y optimización de los recursos en proyectos viales.

ABSTRACT

The purpose of this research is to develop a computational tool that allows determining the possible sources of extraction to be exploited in the construction of affirmed roads in tertiary roads. This tool not only seeks to identify one or more candidates pull sources, but also to optimize carry times by generating short paths and providing information on visit times to multiple pull sources based on the minimum paths between a start node. and an end node. In this case, the nodes correspond to georeferenced points associated with the extraction sources that will be evaluated using the developed computational tool. The main objective of this research is to implement artificial intelligence using Artificial Neural Networks (Artificial Neural Networks) from the information collected in the field using a software model focused on the capture of field and laboratory data. The Artificial Neural Networks will be integrated into the developed web software. The search for information about the supply of materials in the construction of subbases, bases and affirmations has allowed us to show cost overruns and longer execution times due to the uncertainty of the sources of materials to be exploited, either or because extraction sources cannot be found. nearby or because there is no reliability of the material that meets the quality standards required by INVIAS. This materials search methodology, based on the use of artificial intelligence techniques such as neural networks, provides a basis for identifying one or several potential extraction sources that meet the quality standards required by the National Highway Institute for the claimed. This methodology will help in the planning of road projects and will reduce the time and costs associated with the uncertainty in the supply of materials from an extraction source. In summary, this research combines the application of artificial intelligence techniques, specifically Artificial Neural Networks, with web and mobile software designed for data capture and analysis in the field and laboratory. The proposed approach provides an efficient and reliable solution for the selection of extraction sources in the construction of paved roads in tertiary roads, improving the planning and optimization of resources in road projects.

Capítulo 1. Introducción

El departamento del Cauca se destaca por su diversidad agrícola, historia y riqueza cultural. Aunque ha desempeñado un papel importante en el pasado, en la actualidad enfrenta desafíos y se encuentra rezagado en diversos aspectos. Este contexto motiva la presente investigación, que busca comprender las características demográficas y económicas del departamento para encontrar soluciones a sus problemáticas.

El Cauca se caracteriza por tener una población significativa en áreas rurales. Según el Departamento Administrativo Nacional de Estadística (DANE, 2021), para el año 2021, la población total ascendía a 1'491'937 habitantes, de los cuales el 60% residía en zonas rurales. Es crucial considerar estas particularidades al buscar mejoras económicas, ya que gran parte del desarrollo económico se concentra en áreas urbanas y no se ha beneficiado plenamente a la población rural.

La pobreza también representa un desafío importante en el departamento. Entre 2014 y 2020, el Cauca se ubicó en el octavo lugar entre los diez departamentos con los mayores índices de pobreza, con un 55.6% de su población en situación de pobreza y un 23.6 en pobreza extrema (Departamento Administrativo Nacional de Estadística, 2021). Ante la diversidad de la producción agrícola existente en el departamento, una alternativa para mejorar la calidad de vida de las comunidades es facilitar la movilidad y la comunicación hacia los centros urbanos y municipales. Esto impulsaría la valorización de los terrenos, fortalecería la producción agrícola y mejoraría la comunicación en general.

La inversión en infraestructura vial, como la construcción y el mejoramiento de carreteras, se ha vuelto fundamental para mejorar la economía de las comunidades rurales, considerando su gran concentración poblacional en el Cauca. Sin embargo, al emprender proyectos de construcción vial, uno de los aspectos cruciales es encontrar materiales adecuados para la capa de rodadura, ya sea para pavimentos rígidos, flexibles, placa huella o afirmados, entre otras alternativas. La investigación de Darío & Orobio (2015), destaca que la incertidumbre en la elección de los materiales, como la base granular y la carpeta asfáltica, puede ocasionar demoras y sobrecostos en los proyectos viales.

En este contexto, la presente investigación tiene como objetivo optimizar los tiempos de planificación y ejecución de proyectos viales mediante el desarrollo de una aplicación web (App Web). Esta aplicación utiliza conceptos de redes neuronales y teoría de grafos para encontrar materiales que cumplan con las normas establecidas por el Instituto Nacional de Vías (INVIAS) para afirmados y nuevas alternativas tecnológicas en la construcción de carreteras y optimizar distancias de acarreo desde la fuente y el punto de intervención. Además, permitirá la consulta de fuentes de extracción y zonas de producción agrícola, con la finalidad de aprovechar los materiales provenientes de la producción agrícola de la región combinándolos con materiales convencionales, con la idea de mejorar la calidad de la capa de rodadura en vías terciarias.

En el futuro, este proyecto busca interactuar con fuentes de materiales y requisitos para infraestructuras de pavimentos flexibles y rígidos, utilizando una base de datos proporcionada por el centro de desarrollo tecnológico vial de la universidad del Cauca. De

esta manera, se pretende brindar soluciones innovadoras y contribuir al desarrollo de la región en términos de infraestructura vial y calidad de vida para sus habitantes.

1.1. Problema

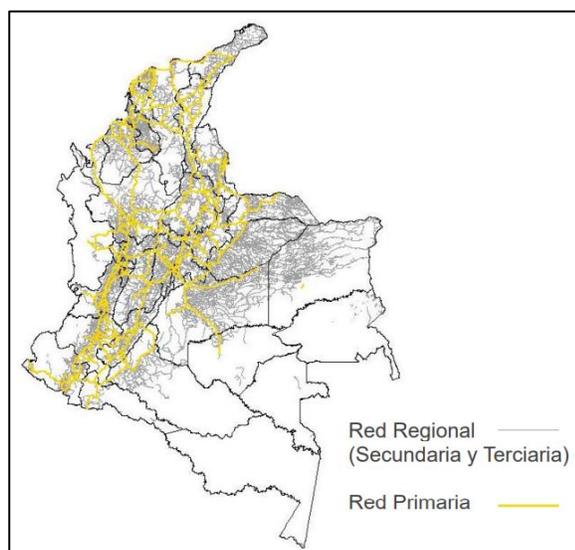
Para comprender el problema, se han identificado tres puntos importantes que deben ser evaluados. En primer lugar, es necesario conocer la cantidad de red vial terciaria en el territorio colombiano, especialmente en el Departamento del Cauca, en relación con la red vial primaria y secundaria, con el fin de determinar en qué sectores se concentra la población de Colombia y, en particular, la del departamento del Cauca. Como segundo parámetro, es fundamental evaluar la situación actual de la red vial terciaria, incluyendo los mejoramientos realizados en los últimos años, la inversión destinada a las carreteras en el departamento, su impacto en la economía y, como tercer punto importante, comprender las problemáticas asociadas a la planificación de proyectos viales en cuanto a la elección de una fuente de extracción para su explotación.

Según la caracterización realizada por el instituto Nacional de Vías (INVIAS, 2008), la Agencia Nacional de Infraestructura (ANI) y Pantoja (2018), en Colombia se han contabilizado actualmente un total de 204.855 km de red vial, clasificados según su funcionalidad de la siguiente manera:

- 5.81% (17.434 Km) corresponde a la red primaria.
- 22.03% (45.137 Km) corresponde a la red secundaria.
- 69.46% (142.284 Km) corresponde a la red terciaria.

Se puede evidenciar que la red terciaria es la de mayor extensión en el territorio colombiano. La Figura 1y Figura 2, representan esta distribución.

Figura 1: Mapa de red vial para Colombia



Pantoja, 2018

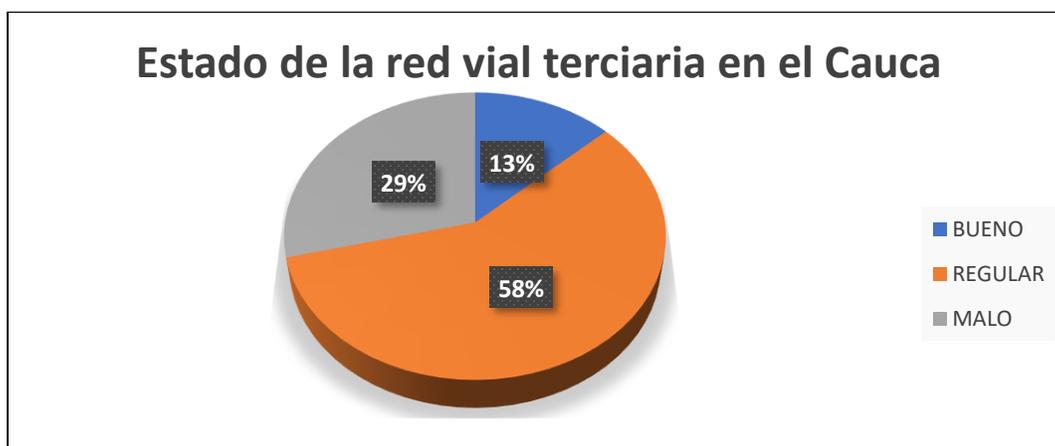
Figura 2: Composición de red vial para Colombia.



Fuente: Adaptado por el Autor

La red terciaria se concentra principalmente en los municipios, abarcando un total de 100748 Km de extensión (Pantoja, 2018) (Figura 2).

Figura 3: Estado de la red vial en el Departamento del Cauca.



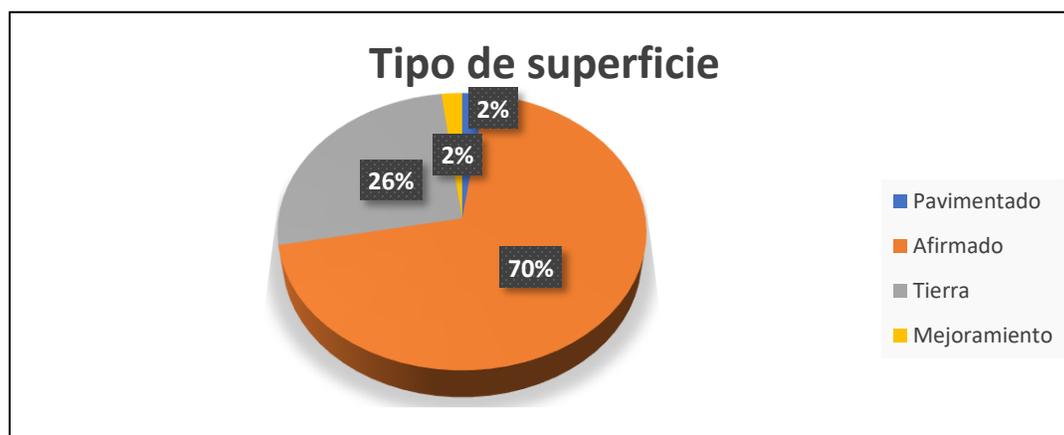
Fuente: Procesamiento de datos realizado por el Autor a partir de datos suministrados por CDT Vial de la Universidad del Cauca

Para conocer el estado de la red vial en el Departamento del Cauca, se ha realizado el análisis de los datos suministrados por el Centro de Desarrollo Tecnológico Vial de la Universidad del Cauca, con el objetivo de determinar el estado actual de la red vial terciaria y buscar alternativas de solución para optimizar tiempos y recursos en la intervención de la infraestructura vial.

Como se observa en la Figura 3, aproximadamente el 87% de la red vial terciaria se encuentra en condiciones regulares y malas, mientras que solo el 13% se encuentra en condiciones “buenas”. Esta situación dificulta la comunicación y el transporte de productos agrícolas, así como el desarrollo del turismo en sectores potenciales, entre otros aspectos que impactan la economía del departamento del Cauca.

Otro aspecto importante a evaluar es el tipo de superficie de rodadura de las vías terciarias del departamento del Cauca, ya que, en su mayoría, están compuestas por afirmados. Extrapolando la información de las condiciones de la red terciaria mencionada anteriormente (Figura 3), se puede concluir que gran parte de la capa de rodadura se encuentra en condiciones malas y regulares

Figura 4: Tipo de superficie de rodadura en las vías terciarias del Departamento del Cauca.

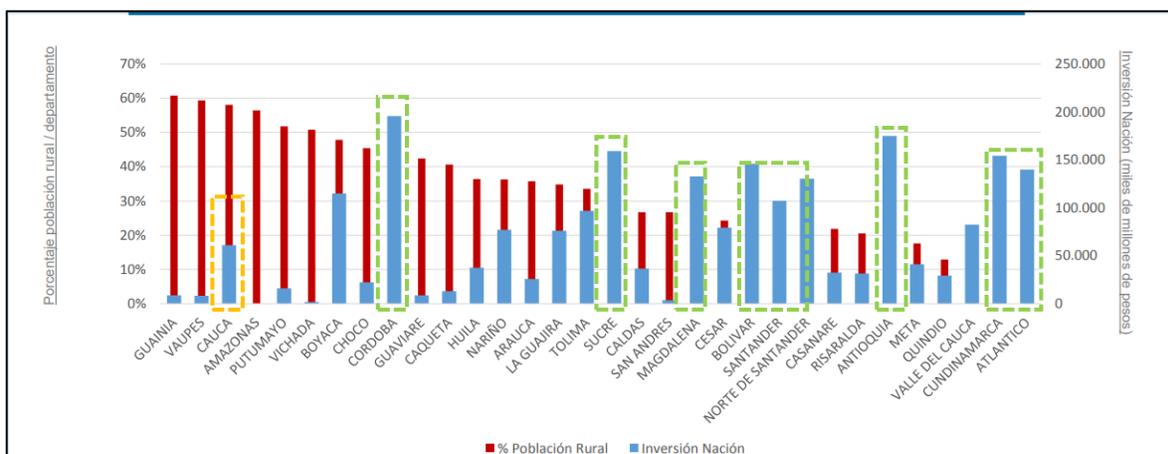


Fuente: Procesamiento realizado por el Autor a partir de datos suministrados por CDT Vial

La Figura 4 muestra el porcentaje de las diferentes superficies de rodadura presentes en las vías terciarias, destacando la predominancia de los afirmados, los cuales, como se mencionó, presentan condiciones deficientes en gran medida. Esta figura (Figura 4) es de suma importancia, ya que permite comprender la situación de la infraestructura vial terciaria y buscar alternativas de solución para mejorar las características de la capa de rodadura en los afirmados, que es el tipo de superficie más utilizado en las carreteras rurales.

Al comparar el departamento del Cauca con otros departamentos como Antioquia, Atlántico, Cundinamarca, entre otros, donde la concentración de zonas rurales es menor, se puede evidenciar que se han invertido considerablemente en infraestructura vial de tercer orden en comparación con el Cauca el cual tiene un gran volumen de población en zonas rurales, cómo lo muestra el gráfico (Figura 5).

Figura 5: Comparación de la población rural respecto a su inversión Nacional.

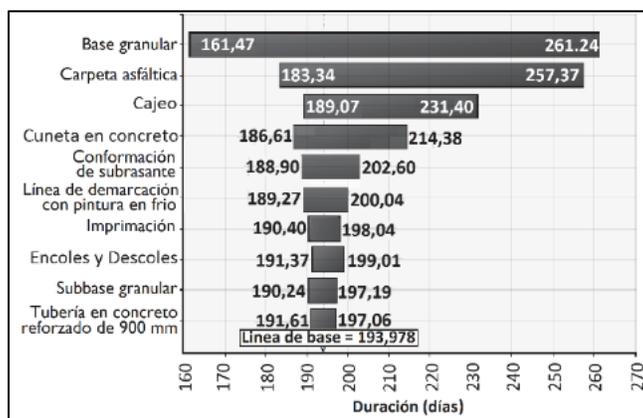


Fuente: Departamento Nacional de Planeación, 2021

En ese sentido es claro comprender que no se ha avanzado en la calidad de movilidad y la comunicación hacia los centros de comercialización de productos agrícolas, lo cual afecta la economía debido a la insuficiente inversión realizada en los últimos años.

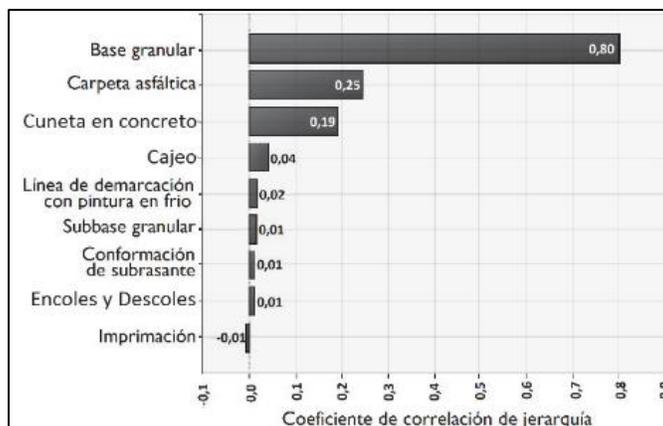
Otro problema investigado es la incertidumbre en la planificación de proyectos viales relacionados con la explotación de fuentes de extracción, debido a la falta de caracterización y estudios de estas fuentes. En muchos casos, no se cuenta con información sobre distancias de recorrido, accesibilidad, entre otros aspectos. Esta falta de certeza ha ocasionado sobrecostos en la ejecución de proyectos viales debido a la poca confiabilidad de la investigación realizada sobre las fuentes. Según lo mencionado por Darío & Orobio (2015), actividades como la construcción de una base granular inciden en la duración del proyecto debido a las grandes cantidades de obra a ejecutar y a la lógica del modelo de la Ruta Crítica (CPM) utilizado durante la planificación. Por lo tanto, es importante prestar atención a esta metodología CPM, que permite identificar las actividades con mayores cantidades y el tiempo previsto para su realización a lo largo del proyecto.

Figura 6: Sensibilidad de las actividades en la duración del proyecto.



Fuente: Gómez Darío y Orobio Armando (2015)

Figura 7: Coeficiente de correlación jerárquica de duración del proyecto.



Fuente: Gómez Darío y Orobio Armando (2015)

La Figura 6 y Figura 7, basadas en la investigación realizada por Darío & Orobio (2015), sobre la rehabilitación de un tramo de la vía Jamundí-Timba, muestran que las actividades con mayor incidencia en la duración del proyecto son la base granular, la carpeta asfáltica, el cajeo y las cunetas de concreto. Estas actividades son de gran magnitud y se analizan mediante la metodología CPM en la planificación de proyectos. Especialmente, la actividad de suministro de materiales para la base granular, incluyendo la búsqueda de materiales, las rutas de acceso y los tiempos de acarreo, es la que tiene mayor impacto debido a sus altos volúmenes. La incertidumbre en estos aspectos puede resultar en sobrecostos. Por lo

tanto, es fundamental buscar alternativas de solución que permitan mejorar estos problemas y así realizar más kilómetros de vías mejoradas a través de una adecuada planificación de la estructura vial.

1.2. Justificación

La red terciaria tiene un alto impacto en el crecimiento de la región y el empleo. Esto se debe a que se facilita la comercialización de productos agrícolas al mejorar el transporte y la interacción con diferentes zonas de la región. Este impacto genera un efecto positivo en el PIB. De acuerdo con los informes del Departamento Nacional de Planeación (DNP, 2020), una inversión de 1 billón en la red vial terciaria puede aumentar el PIB en un 0.082% y generar un total de 10720 empleos. Por lo tanto, la inversión en infraestructura vial terciaria ayuda a mejorar las condiciones económicas de las poblaciones rurales.

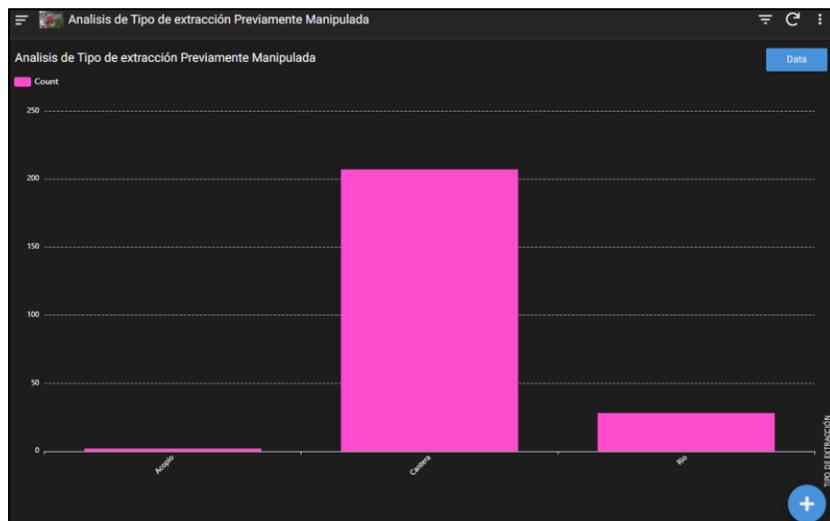
El departamento del Cauca es uno de los que tiene una mayor cantidad de población asentada en zonas rurales (Gamarra-Vergara, 2008; Departamento Nacional de Planeación [DNP], 2020). Además, se ha observado un aumento de la pobreza en las zonas más alejadas del territorio debido a la falta de oportunidades.

El departamento del Cauca cuenta con una gran cantidad de materiales que poseen buenas características mecánicas y físicas, los cuales pueden ser utilizados en la construcción, mejoramiento y nuevas alternativas de implementación en la infraestructura vial debido a su ubicación geográfica. Sin embargo, no existe una metodología que ayude a reducir la incertidumbre en la información sobre las fuentes de extracción a explotar. Esto se debe a la falta de investigaciones relevantes sobre las fuentes, lo cual dificulta tener certeza sobre el material que se extrae y su aprovechamiento.

A partir del procesamiento de datos suministrados por el Centro Desarrollo Tecnológico para la Innovación de la Infraestructura Vial Terciaria del Departamento del Cauca de la Universidad del Cauca (CDT), en colaboración con la secretaría de infraestructura de la gobernación del Cauca hasta el año 2020, se pudo determinar que existen 237 fuentes de extracción en las zonas Norte, Centro, Sur y Oriente del departamento. Estas fuentes se han clasificado en tres categorías: Cantera, Río y Acopio, mediante la creación del primer software (Canteras-Móvil) para la gestión de fuentes de materiales. La información captura de nuevas fuentes de extracción, utilizando el software, han sido analizadas en el laboratorio y registradas en el mismo. Las figuras (Figura 8 y Figura 9) muestran el total de fuentes de extracción procesadas y almacenadas en el software Canteras móvil, para su

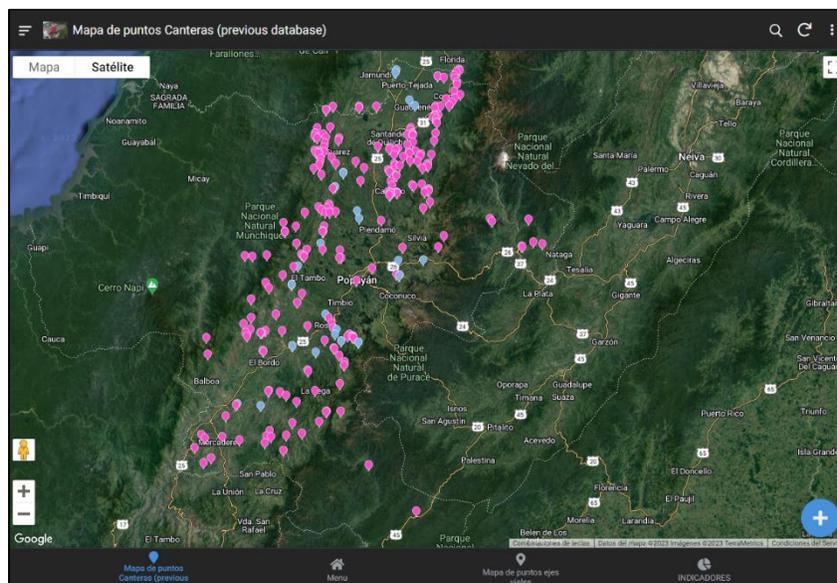
posterior administración, búsqueda y análisis. En color rosa las capturadas por la oficina de Secretaría de infraestructura de la gobernación del Cauca y en azul las fuentes capturadas por el CDT Vial.

Figura 8: Total de Fuentes de extracción para el Departamento del Cauca en las zonas: Norte, Sur, Centro y Oriente en Software "Canteras Móvil"



Fuente: Por el Autor

Figura 9: Mapa de distribución de fuentes de extracción. En rosado las capturadas por la oficina de Infraestructura de la Gobernación del Cauca y en azul las fuentes de extracción capturadas por el CDT Vial con: "Canteras Móvil"



Fuente: Por el Autor

No se ha identificado ninguna alternativa viable para mejorar los tiempos de planificación y ejecución, ni para mitigar las incertidumbres asociadas a la selección de fuentes de extracción. Con el propósito de abordar esta necesidad, se plantea el desarrollo de un software destinado a optimizar la gestión de fuentes de extracción. Se busca con ello la creación de una herramienta que contribuya a agilizar los procesos de elección de fuentes. Para lograrlo, se propone emplear el entrenamiento de redes neuronales artificiales. Este entrenamiento se ha realizado empleando una base de datos que engloba información relativa a las fuentes de extracción en el departamento del Cauca. Dichas fuentes han sido exhaustivamente caracterizadas y evaluadas en laboratorio por el CDT, confrontando sus propiedades con las normativas establecidas por el Instituto Nacional de Vías (INVIAS) para la construcción de afirmados.

Los softwares Canteras móvil y Canteras Web, están diseñados con un objetivo específico. El primero (Canteras-Móvil) permite capturar información en campo sobre una fuente de extracción, comprobar su existencia en la base de datos, rectificar o editar sus características, realizar la captura de datos de laboratorio y generar informes en formato PDF, hojas de cálculo y fotografías. Además, facilita la interacción con toda la información de la captura de datos en campo y laboratorio mediante gráficos y finalmente uno de los objetivos generales es realizar el proceso de ruta mínima para determinar tiempos y distancia de acarreo de materiales entre una fuente candidata a ser explotada y el sitio de construcción.

Estos datos pueden ser procesados en el software Canteras-Web, el cual tiene como objetivo generar proyecciones de rutas para visitar más de una fuente en el menor recorrido posible, con el fin de extraer materiales de forma más eficiente. El software Canteras-Web también es capaz de generar posibles fuentes de extracción para ser explotadas, a partir del procesamiento y desarrollo del entrenamiento de redes neuronales artificiales. En resumen, el objetivo final de estos softwares es administrar, gestionar y procesar fuentes de extracción con el fin de reducir los tiempos de ejecución, evitar sobrecostos y mantener un flujo de trabajo continuo en la planificación de proyectos viales para el suministro de materiales.

1.3. Objetivos

Los objetivos del proyecto se presentan a continuación:

1.3.1. Objetivo General

El objetivo general es aplicar inteligencia artificial para obtener una o más fuentes de extracción a explotar, según las exigencias, mediante el uso de herramientas como el entrenamiento de Redes Neuronales Artificiales (ANN, por sus siglas en inglés) a partir de la creación de un software, teniendo en cuenta las especificaciones del Instituto Nacional de Vías (INVIAS) en el caso de los afirmados.

1.3.2. Objetivos Específicos:

- Verificar y validar las fuentes y tipo de material disponible en las zonas de estudio del Departamento del Cauca.
- Crear herramientas computacionales (Software: Canteras Móvil) para la captura de datos en campo y en laboratorio, con el fin de obtener información técnica de las fuentes de extracción, incluyendo ensayos del material encontrado en una zona determinada.
- Realizar la clasificación de fuentes y rutas de acceso a los materiales mediante la creación del software Canteras Web.
- Crear un modelo (Grafo) que determine la mejor ruta de acceso a las fuentes de extracción, considerando los nodos de interés y sus aristas de acceso, utilizando el software Canteras-Web.

1.4. Beneficios del proyecto

El propósito fundamental de esta investigación es generar una o varias fuentes de extracción que reúnan las cualidades propicias para su explotación, garantizando la conformidad con los requisitos preestablecidos. Este objetivo se concreta mediante la utilización de un formulario en el software Canteras-Web, que permite la entrada de datos precisos. Las fuentes de extracción así obtenidas se consideran dotadas de propiedades físicas y mecánicas apropiadas para su empleo en afirmados (inicialmente), pues su idoneidad se ha respaldado mediante la comparación con los estándares del Instituto Nacional de Vías (INVIAS).

Esta iniciativa tiene la capacidad de rebajar los lapsos de ejecución y elevar la confiabilidad de los materiales extraídos, en consecuencia de la reducción de incertidumbre en el suministro de los mismos. Este logro representaría un ahorro financiero y una ganancia temporal, dado que la búsqueda de fuentes de extracción conformes con criterios de calidad

es un proceso demandante. A su vez, el proyecto puede consolidarse como una herramienta multifuncional para la gestión, investigación y análisis de fuentes de extracción.

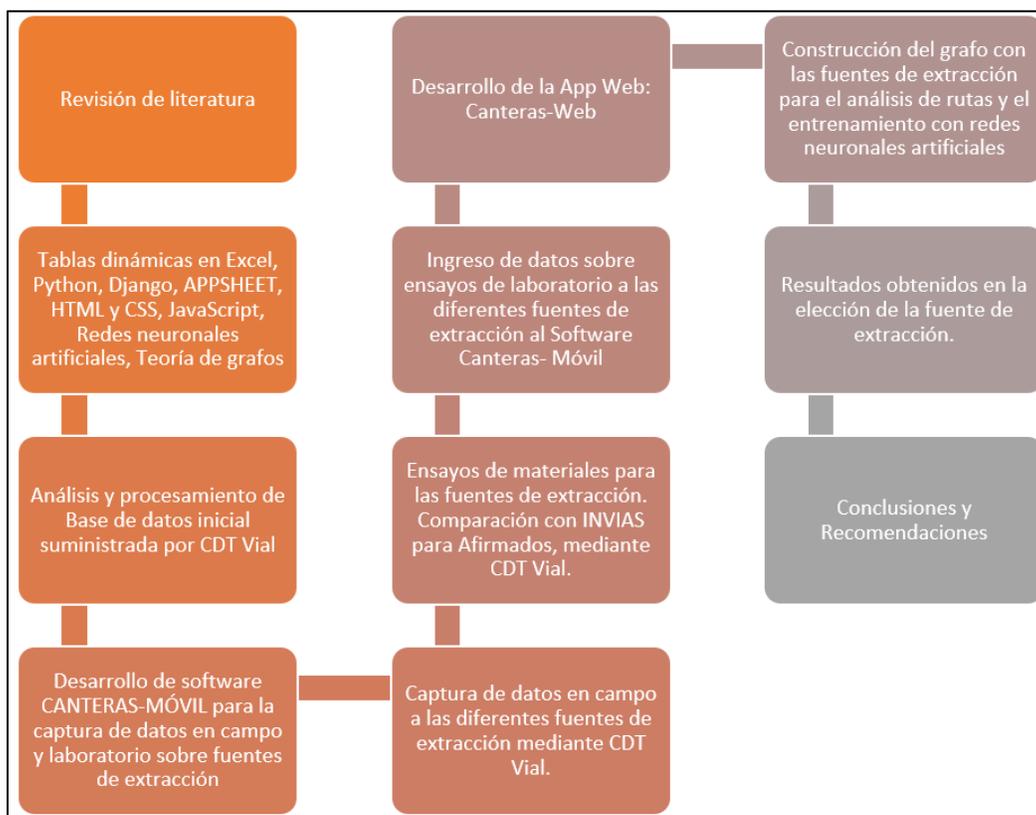
El sistema informático propuesto como trabajo futuro se le añadirá la opción de combinación de materiales a fin de mejorar las características de aquellos que no alcancen los parámetros requeridos. Por ejemplo, sería viable combinar materiales de las fuentes X y Y, generando así una composición apta. Adicionalmente, la herramienta computacional optimizará la búsqueda de rutas de acceso más cortas, evitando excesos en los trayectos de acarreo del material y, por ende, reduciendo los costos asociados.

Este proyecto provee herramientas informáticas para optimizar gastos, agilizar procesos y ampliar las posibilidades de construcción de vías terciarias. Durante el proceso de recopilación de información, se ha constatado que el abastecimiento de materiales representa un componente significativo en el aumento de los costos y los plazos de ejecución. Por lo tanto, esta investigación presenta una metodología innovadora para el suministro de materiales, subsanando la incertidumbre que podría desencadenar consecuencias sustanciales en términos económicos y temporales.

1.5. Organización

El proceso organizativo se ha construido siguiendo una serie de pasos con la finalidad de lograr el objetivo planteado en la presente tesis. El diseño puede describirse a continuación en el siguiente esquema.

Figura 10: Organización.



Fuente: Por el Autor

Como primer paso, se realizó una exhaustiva revisión del estado del arte en relación al tema de investigación. Los estudios previos no proporcionaron una metodología clara para la selección de fuentes de extracción que pudieran mejorar la incertidumbre en el suministro de materiales para proyectos viales, en particular para afirmados. La falta de registros claros en este proceso de planificación afianza la relevancia de nuestro estudio y el logro de nuestro objetivo.

Como segundo paso, se llevó a cabo un estudio de todas las herramientas a utilizar a lo largo de la investigación. Se recopiló información básica que permitiera desarrollar los programas necesarios, así como herramientas computacionales para el procesamiento de los datos iniciales. Esto proporcionó una base sólida de información para iniciar la ejecución.

A continuación, se desarrolló el software Canteras-Móvil, diseñado para la captura de nuevos datos en campo y los resultados de laboratorio de las fuentes de extracción que fueron recopilados por el CDT Vial. Como siguiente paso, se creó el software Canteras-

Web, cuyo objetivo es generar resultados sobre las fuentes de extracción a partir de los datos capturados en campo y analizados en laboratorio. El resultado de este es la selección de una fuente de extracción que cumpla con los requisitos exigidos por el Instituto Nacional de Vías para los afirmados. Es importante destacar que esta selección se realiza mediante el entrenamiento de redes neuronales artificiales utilizando la base de datos mencionada anteriormente.

Otro resultado importante es la determinación de las rutas de acarreo más cortas, lo cual reduce costo y tiempo. Este cálculo es realizado utilizando el algoritmo de Dijkstra con las fuentes de extracción que están incluidas en el modelo de entrenamiento de la red neuronal artificial.

Por último, cabe mencionar la organización investigativa que se ha llevado a cabo en este estudio.

1. Revisión Bibliográfica: Se realiza una revisión de literatura existente sobre la extracción de materiales para afirmados, normas técnicas del Instituto nacional de Vías, problemas con la incertidumbre del suministro de estos materiales en la planeación y ejecución de proyectos viales.
2. Recopilación de datos: Se establece la nueva base de datos con la información de ensayos de laboratorio de diferentes fuentes de extracción de materiales para afirmados, suministradas por el CDT Vial de la Universidad del Cauca.
3. Diseño y desarrollo de redes neuronales artificiales y la construcción del Grafo: Utilización de técnicas de aprendizaje automático y redes neuronales artificiales para desarrollar un modelo que pueda analizar los datos recopilados y predecir la calidad y aptitud de diferentes fuentes de extracción para afirmados. Algoritmos sobre redes neuronales convolucionales, redes neuronales recurrentes, entre otros, Teoría de Grafos, algoritmo de Dijkstra.
4. Validación y evaluación del modelo: Evaluaciones de la precisión y eficacia del modelo de redes neuronales artificiales.
5. Análisis de resultados y conclusiones: Se analiza los resultados obtenidos a partir del modelo de redes neuronales artificiales. Comparar las diferentes fuentes de extracción según los requisitos del Instituto Nacional de Vías. Se evalúa la metodología propuesta.

Capítulo 2. Revisión de literatura

2.1. Lenguaje de programación

Para el desarrollo del software Canteras-Web, se ha hecho uso del framework Django. De acuerdo con Django Software Foundation (2023) está escrito en Python, tiene como filosofía el desarrollo rápido y el diseño limpio y pragmático. Sigue el patrón arquitectónico modelo-téplate-view (MTV) es mantenido por la Django Software Foundation. Es gratuito y de código abierto bajo la licencia BSD. Django es conocido por su capacidad para aplicar en sitios web de alto tráfico con facilidad y sus características de seguridad integradas. También tiene una gran comunidad de desarrolladores que contribuyen a su desarrollo y mantenimiento.

Para trabajar con Django, será de suma importancia tener conceptos de Python, HTML, CSS y JavaScript, estos últimos contribuirán al buen entorno de diseño, dinamismo, estética y todo el entorno visual que se pueda realizar. Gran parte de la funcionalidad de Canteras-Web es desarrollado utilizando Python y toda la parte visual y dinámica es utilizando HTML, CSS y JavaScript.

Python es un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, multiplataforma y orientado a objetos. Un lenguaje interpretado es aquel que se ejecuta utilizando un programa intermedio llamado intérprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora (Lenguajes compilados). La ventaja de los lenguajes compilados es que su ejecución es más rápida. Sin embargo los lenguajes interpretados son más flexibles y portables.

Python tiene muchas de las características de los lenguajes compilados, por lo que se podría decir que es semi interpretado. En Python, Java y muchos lenguajes, el código fuente se traduce a un pseudo código máquina intermedio llamado bytecode la primera vez que se ejecuta, generando archivos .pyc o .pyo (bytecode optimizado), que son los que se ejecutarán en sucesivas ocasiones.

Cuando se dice que es un lenguaje fuertemente tipado se refiere a que no se permite tratar a una variable como si fuera de un tipo distinto al que tiene, es necesario convertir de forma explícita dicha variable al nuevo tipo previamente. Por ejemplo, si se tiene una variable tipo string no se podrá tratarla como un número. Adicionalmente este lenguaje de programación es de Tipado dinámico. Las características de tipado dinámico se refieren a que no es necesario declarar el tipo de dato que va a contener una determinada variable, sino que su

tipo se determinará en tiempo de ejecución según el tipo de valor al que se le asigne (Duque, Raúl, Python Para todos, 2011)

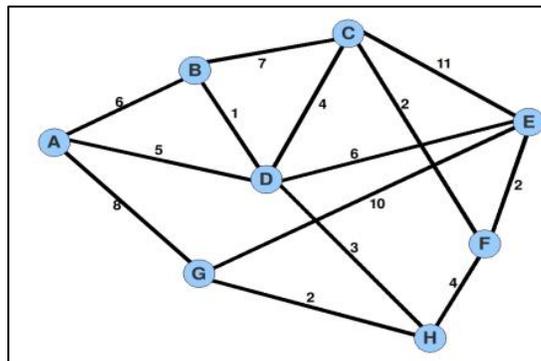
Para utilizar el framework Django es necesario conocer de HTML, CSS y el lenguaje de programación JavaScript. En cuanto a HTML este es un lenguaje de marcado con el que rodeas contenido, para decirle a los navegadores acerca de los encabezados, listas, tablas, etc. CSS es el lenguaje de hojas de estilos con el cual estilas la página, para decirles a los navegadores que cambien el color, fuente, diseño y más (Khanacademy, Introducción a HTML y CSS). En cuanto al lenguaje de programación Java Script, permite implementar funciones complejas en páginas web, también muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas productoras de vídeo, etc. Puedes apostar que JavaScript está ahí. En ese sentido se dice que es un lenguaje de secuencias de comandos que te permite crear contenido de actualización dinámica, controlar multimedia, animar imágenes y prácticamente muchas más cosas (Mdn web docs, 2023).

2.2. Ruta mínima

La determinación de la ruta mínima o más corta entre dos nodos o puntos de referencia en un entorno con múltiples opciones de rutas es un proceso complejo. Por lo tanto, es importante familiarizarse con las herramientas disponibles para abordar el problema de la Ruta mínima. Este tema es de relevancia en la presente investigación, ya que uno de los procesos a desarrollar está relacionado con las distancias de acarreo del material extraído o las visitas a fuentes de extracción donde es posible pasar por otras fuentes cómo resultado del camino más corto. Esto resulta muy útil si se desean visitar varias fuentes de materiales en un solo día, considerando que cada punto de control se convierte en una fuente de extracción. El algoritmo de la Ruta mínima de Dijkstra es una herramienta invaluable para abordar problemas como los planteados en esta investigación.

El algoritmo de Ruta mínima de Dijkstra permite determinar la ruta más corta que conecta dos nodos en un grafo. En la siguiente figura, se muestra una recopilación de nodos y enlaces. En el caso de los nodos, se refiere a las fuentes de extracción, mientras que los enlaces representan los accesos viales para llegar a dichas fuentes. El grafo, por su parte, se refiere al conjunto de nodos y enlaces que permiten acceder a cada nodo, teniendo en cuenta otros aspectos que se detallarán posteriormente.

Figura 11: Grafo de análisis



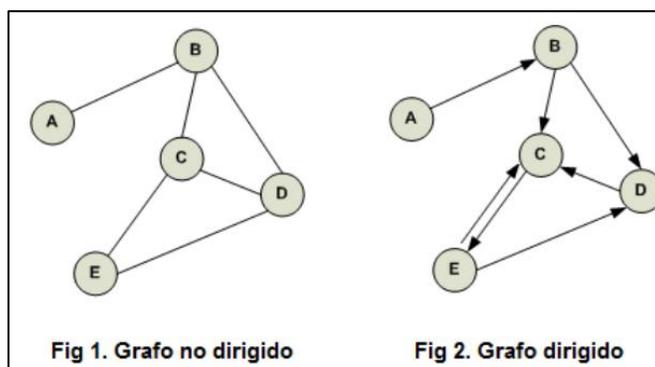
Fuente: Rey Salcedo, Dijkstra, el camino más corto.

2.2.1. Algoritmo de Dijkstra

Para comprender el algoritmo de Dijkstra es importante comprender la definición de grafo. De acuerdo con Sclar (2016) un grafo es un conjunto, no vacío, de objetos llamados vértices (o nodos) y una selección de pares de vértices, llamados aristas (edges en inglés) que pueden ser orientados (dirigidos) o no. “Un grafo es un conjunto de puntos y líneas que unen pares de esos puntos”

En los grafos dirigidos las aristas son doble mano (se puede ir en ambos sentidos). En los dirigidos en cambio, las aristas se recorren en un único sentido: Desde el origen al destino de la flecha. Si se quiere representar que la calle une las esquinas u y v es doble mano se deberá poner dos aristas: una que una vaya de u a v y la otra que vaya de v a u, tal como se visualiza en la siguiente figura.

Figura 12: Camino mínimo en grafos.

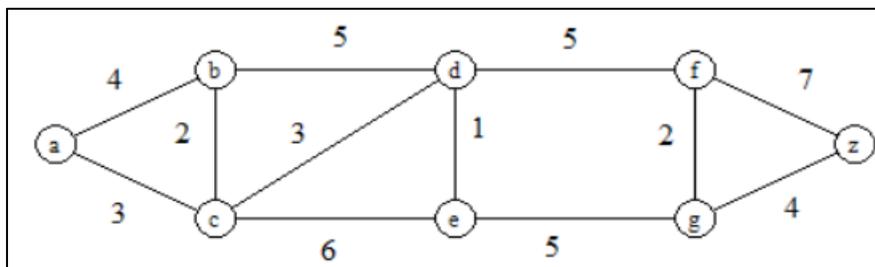


Fuente: Melanie Sclar

A medida que los problemas se dificultan, puede suceder que sea difícil modelar el problema con un grafo, pero que una vez que lo hayamos hecho, el problema se vuelva sencillo. En la estructura de un grafo existe el término “peso” para referirse más precisamente a la distancia de recorrido entre un nodo y otro, en ese sentido modelar un grafo con n cantidad de nodos y n aristas resulta tedioso. Dijkstra modela este hecho tal como describe su procedimiento:

El algoritmo de Dijkstra sirve para cualquier grafo con pesos (dirigido o no) siempre y cuando estos no sean negativos. Este algoritmo calcula las distancias mínimas desde un nodo inicial a todos los demás. Para hacerlo, en cada paso se toma el nodo más cercano al inicial que aún no fue visitado (le diremos v). Este nodo tiene calculada la menor distancia al nodo inicial. Luego, se recalcula todos los caminos mínimos, teniendo en cuenta a v como camino intermedio. En cada paso se tendrá un subconjunto de nodos que ya tienen calculada su mínima distancia y los demás tienen calculada su mínima distancia si solo puedo usar los nodos del conjunto como nodos intermedios. Con cada iteración se agrega un nodo más a nuestro conjunto, hasta resolver el problema en su totalidad (Sclar, 2016). A continuación se realiza una explicación formal de acuerdo con Sclar (2016)

Figura 13: Grafo ponderado.



Fuente: Se, Dijkstra. Caminos mínimos

De acuerdo con lo que se visualiza en la Figura 13. Se desarrolló un algoritmo que permite hallar un camino de longitud mínima entre dos vértices en un grafo ponderado y su longitud. Sean a el vértice de partida y z el vértice final. Al peso de una arista cualquiera $\{u, v\}$ se le denotará por $w(u, v)$; si $\{u, v\}$ no es arista se considera $w(u, v) = \infty$. Se va a seguir un proceso iterativo:

Se construye un conjunto distinguido de vértices al que se añade un vértice en cada iteración. Este conjunto es el conjunto de vértice para los que se ha obtenido un camino de longitud mínima desde el vértice de partida a . También se lleva a cabo un proceso de etiquetado de los vértices. Este etiquetado es la longitud de un camino mínimo de “ a ” al

correspondiente vértice pasando únicamente por vértices en el conjunto distinguido de vértices, junto a los vértices por los que pasa el camino.

Inicio: en el paso 0 se define:

$S_0 = \emptyset$, se comienza con \emptyset como conjunto distinguido de vértices.

$L_0(a) = 0$, $C_0(a) = \emptyset$ es la etiqueta longitud-camino del vértice a en el paso 0

$L_0(v) = \infty$, $C_0(v) = \emptyset \forall v$ vértice $v \neq a$, ∞ es la etiqueta longitud-camino de cualquier vértice distinto de a

- Se supone construidos en el paso $(k-1)$ -ésimo S_{k-1} , el conjunto de vértices distinguidos, y $L_{k-1}(v)$, $C_{k-1}(v)$, la etiqueta de cualquier vértice v .
- Veamos cómo se construyen S_k , $L_k(v)$, $C_k(v)$ en el paso k -ésimo:

Se actualiza el conjunto distinguido de vértices: S_k se construye añadiendo a S_{k-1} un vértice cuya etiqueta $L_{k-1}(v)$ sea mínima entre los vértices v que no están en S_{k-1} . Sea u un tal vértice, esto es, $S_k = S_{k-1} \cup \{u\}$.

Se actualiza las etiquetas de todos los vértices que no están en S_k de la forma siguiente:

$$L_k(v) = \text{mínimo} \{L_{k-1}(v), L_{k-1}(u) + w(u,v)\}$$

Si $L_k(v) = L_{k-1}(v)$, se hace $C_k(v) = C_{k-1}(v)$

Si $L_k(v) = L_{k-1}(u) + w(u,v)$, se hace $C_k(v) = C_{k-1}(u) \cup \{u\}$

La justificación reside en que $L_k(v)$ es la longitud de un camino de longitud mínima entre a y v que pasa sólo por vértices de S_k . La actualización se puede llevar a cabo considerando ó bien el más corto que pasa por vértices de S_{k-1} ó bien es el resultado de añadir la arista $\{u, v\}$ al camino más corto obtenido en el paso $k-1$.

- El proceso finaliza en el momento en el que el vértice z se añade al conjunto distinguido. En este caso la etiqueta $L_m(z)$ es la longitud mínima entre a y z y $C_m(z)$ es un camino de longitud mínima entre a y z . Esto es debido a la observación siguiente:

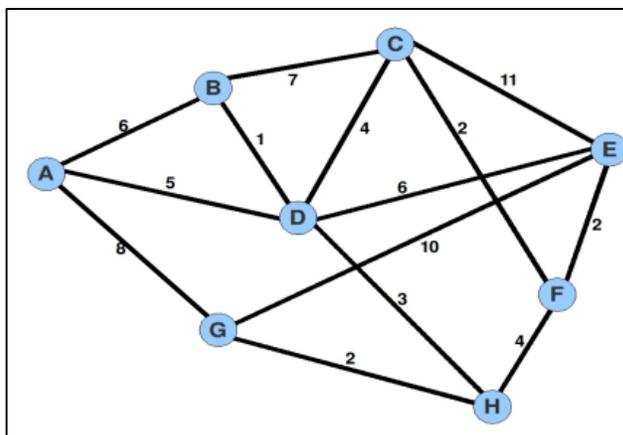
Observación 5.45. En el paso k -ésimo del algoritmo de Dijkstra se verifica:

- a) La etiqueta $L_k(v)$ de cada vértice que está en S_k es la longitud de un camino de longitud mínima entre a y dicho vértice.

- b) La etiqueta $L_k(v)$ de cada vértice que no está en S_k es la longitud de un camino de longitud mínima entre a y dicho vértice que sólo contiene (exceptuando el propio vértice) vértice de S_k .

Para afianzar el concepto anterior se presenta el siguiente ejemplo proporcionado por Salcedo (2016) con la finalidad de comprender de manera más didáctica y gráfica.

Figura 14: Dijkstra.

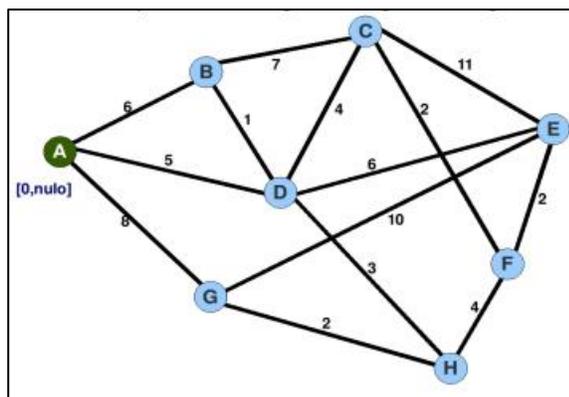


Fuente: Rey Salcedo, 2016

Las propiedades de un nodo son dos: *acumuladoPeso* y otro llamando *nodoAntecesor*, el uso de estas variables se comprenderá en el transcurso de la explicación del ejemplo.

1. **(Primera iteración)** Se selecciona un nodo de inicio, en este caso se usará el nodo A, e inicializa sus variables de la siguiente forma:
 $acumuladoPeso = 0$;
 $nodoAntecesor = \text{nulo}$;
 Al ser el nodo inicio, no tiene pesos acumulados ni nodo antecesor, además lo cambia de color para indicar que ya está en uso, quedando según la siguiente figura.

Figura 15: Iteración 1 para el grafo



Fuente: Adaptado por el Autor

2. **(segunda iteración)** Se marca los nodos adyacentes de A, inicializando sus variables de la siguiente forma:

acumuladoPeso= [acumuladoPeso del nodo antecesor (en este caso el de A)] + peso de la arista que los une,

nodoAntecesor = el nombre del nodo antecesor;

así para B:

acumuladoPeso = $0+6=6$;

nodoAntecesor = A;

Para D:

acumuladoPeso= $0+5=5$;

nodoAntecesor =A;

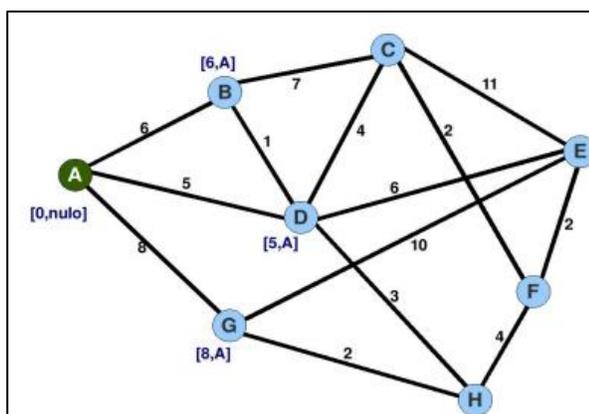
Para G:

acumuladoPeso= $0+8=8$;

nodoAntecesor = A;

Quedando según la siguiente figura:

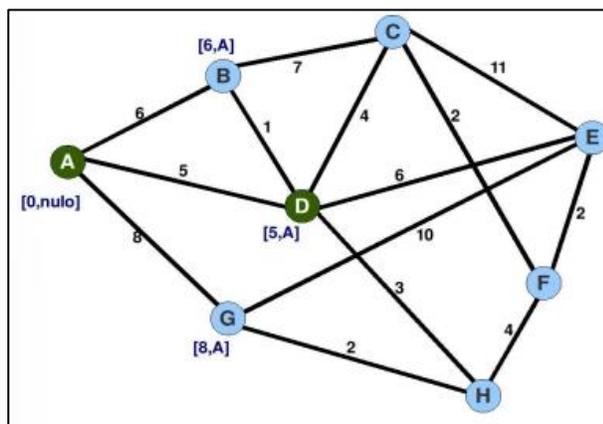
Figura 16: Segunda iteración.



Fuente: adaptado por el Autor

- (Tercera iteración)** Se localiza los nodos visitados y que no esté marcado con el color verde y que tenga menor peso acumulado (acumuladoPeso), en caso de tener más de uno con la característica antes descritas estamos en la libertad de escoger cualquiera de estos, una vez localizado lo marcamos de color verde. Para este caso el nodo a escoger es el D, dado que su peso acumulado es el menor de todos con 5, quedando según la siguiente figura:

Figura 17: Tercera iteración.



Fuente: Adaptado por el Autor

- (Cuarta iteración)** Se ubica en el nodo D, el cual fue seleccionado en el paso anterior y se marca sus nodos adyacentes siempre y cuando no se halla usado, es decir no este de color verde. (ver como este paso es análogo a la iteración 2)

Para este caso los nodos adyacentes de D no usados son: C, E, H, B; se inicializa los valores de los nodos adyacentes así:

Para C:

acumuladoPeso = $5 + 4 = 9$

nodoAntecesor=D;

Para E

acumuladoPeso = $5 + 6 = 11$;

nodoAntecesor = D;

Para H:

acumuladoPeso = $5 + 3 = 8$;

nodoAntecesor = D;

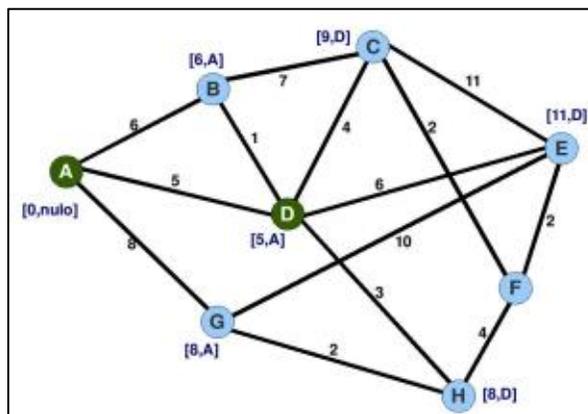
Para B:

Como el nodo B ya este marcado (por el nodo A), primero se calcula el peso acumulado así: acumuladoPeso = $5 + 1 = 6$,

Como el nuevo acumuladoPeso no genera un valor más pequeño del que ya estaba, es decir no lo mejora, el nodo B no se modifica. Es de anotar que el numero 5 corresponde al acumuladoPeso del nodo antecesor.

Quedando según la siguiente figura:

Figura 18: Cuarta iteración.

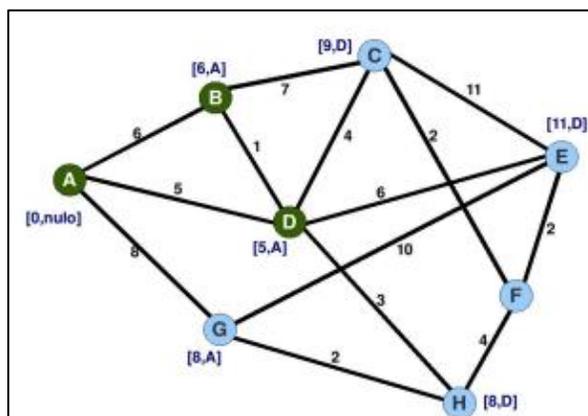


Fuente: Adaptado por el Autor

5. **(Quinta iteración):** se localiza los nodos visitados y que no esté marcado con el color verde y que tenga menor peso acumulado (acumuladoPeso), en caso de que tenga más de uno con las características antes descritas estamos en la libertad de escoger cualquiera de estos.

Una vez localizado lo marcamos de color verde. Para nuestro caso el nodo a escoger es el B, dado que es el de menor peso acumulado con 6. Quedando según la siguiente figura:

Figura 19: Quinta iteración.



Fuente: Adaptado por el Autor

6. **(Sexta iteración):** Posteriormente se ubica en el nodo B, el cual fue seleccionado en el paso anterior y se marca sus nodos adyacentes siempre y cuando no se halla usado, es decir no este de color verde.

Para este caso el único nodo adyacentes de B no usado es C.

Se inicializa los valores de nuestro nodo adyacentes así:

Para C:

Como el nodo C ya está marcado (por el nodo D), primero se calcula el peso acumulado así: $\text{acumuladoPeso} = 6 + 7 = 13$;

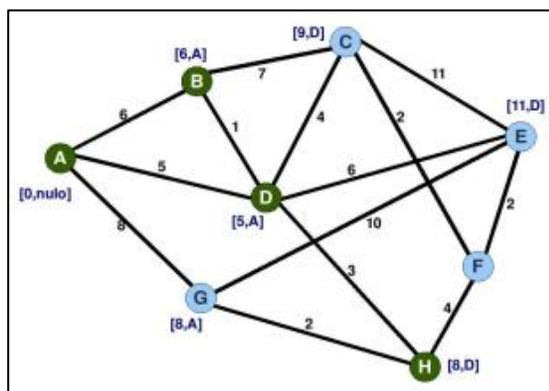
Como el nuevo acumuladoPeso no genera un valor más pequeño del que ya estaba, es decir no lo mejora, el nodo C no lo modificamos. En ese sentido el gráfico no sufre ningún cambio.

7. **(Séptima iteración):** Se localiza los nodos visitados y que no esté marcado con el color verde y que tenga menor peso acumulado (acumuladoPeso), en caso de que tengamos más de uno con la característica antes descrita estamos en la libertad de escoger cualquiera de estos.

Una vez localizado se marca de color verde. (ver como este paso es análogo a la tercera y quinta iteración)

Para este caso es posible escoger entre los nodos G, H ya que el peso acumulado en cada uno de ellos son los menores con 8; se inclina a usar el nodo H (si usamos el nodo G el resultado no varía). Quedando según la siguiente figura:

Figura 20: Séptima iteración.



Fuente: Adaptado por el Autor

8. **(Octava iteración):** Se ubica en el nodo H, el cual fue escogido en el paso anterior y se marcan sus nodos adyacentes siempre y cuando no se halla usado, es decir no este de color verde. Para este caso los nodos adyacentes de H usados son: F, G. Inicializamos los valores de nuestros nodos adyacentes así:

Para F:

acumuladoPeso=8+4=12;

nodoAntecesor = H;

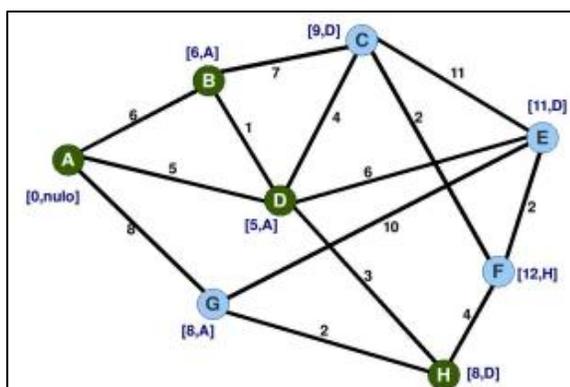
Para G:

Como el nodo G ya está marcado (por el nodo A), primero calculamos el peso acumulado así:

acumuladoPeso=8+2=10;

Como el nuevo acumuladoPeso no genera un valor más pequeño del que ya estaba, es decir no lo mejora, el nodo G no es modificado. Es de anotar que el numero 8 corresponde al acumuladoPeso del nodo antecesor. Quedando según la siguiente figura:

Figura 21: Octava iteración.

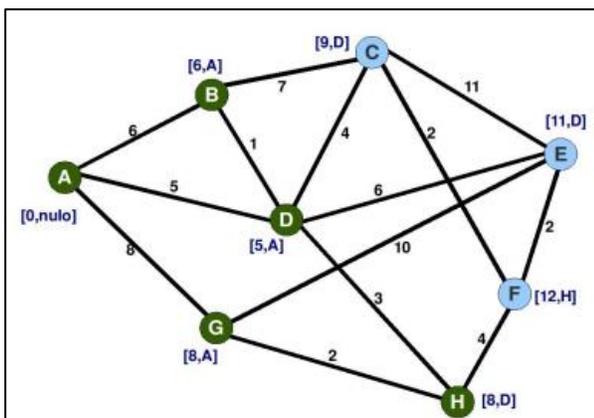


Fuente: Adaptada por el autor

9. **(Novena iteración)** Se localiza los nodos visitados y que no esté marcado con el color verde y que tenga menor peso acumulado (acumuladoPeso), en caso de tener más de uno con la característica antes descritas está en la libertad de escoger cualquiera de estos.

Una vez localizado lo marcamos de color verde. Para este caso el nodo a escoger es el G, dado que su peso acumulado es el menor con 8. Quedando según la siguiente figura:

Figura 22: Novena iteración.



Fuente: Adaptada por el Autor

10. **(Decima iteración)** Se ubica el nodo G, el cual fue escogido en el paso anterior y se marca sus nodos adyacentes siempre y cuando no se halla usado, es decir no este de color verde. (Este paso es análogo a la segunda, cuarta, sexta iteración, etc.) Para nuestro caso el único nodo adyacente de G no usado es E.

Inicializamos los valores de nuestro nodo adyacentes así:

Para E:

Como el nodo E ya está marcado (por el nodo D), primero calculamos el peso acumulado así:

AcumuladoPeso=8+10=18;

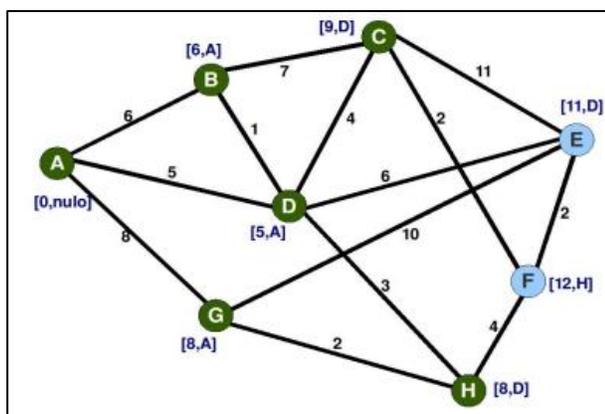
Como el nuevo acumuladoPeso no genera un valor más pequeño del que ya estaba, es decir no lo mejora, el nodo E no se modifica.

Recordemos que el numero 8 corresponde al acumuladoPeso del nodo antecesor.

En ese sentido nuestro gráfico no sufre cambios.

11. **(Décimo primera iteración):** Se localiza los nodos visitados y que no esté marcado con el color verde y que tenga menor peso acumulado (acumuladoPeso), en caso de que tengamos más de uno con la característica antes descritas estamos en la libertad de escoger cualquiera de estos. Una vez localizado se marca de color verde. Para este caso el nodo a escoger es el C ya que su peso acumulado es el menor con 9. Quedando según la siguiente figura:

Figura 23: Décimo primera iteración.



Fuente: Adaptada por el Autor

12. **(Décimo segunda iteración):** Se ubica en el nodo C, el cual fue escogido en el paso anterior y se marca los nodos adyacentes siempre y cuando no lo hallamos usado, es decir no este de color verde.

Para este caso los nodos adyacentes de C son E, F.

Inicializamos los valores de los nodos adyacentes así:

Para E:

Como el nodo E ya está marcado (por el nodo D), primero se calcula el peso acumulado así:

$$\text{acumuladoPeso} = 9 + 11 = 20;$$

como el nuevo acumuladoPeso no genera un valor más pequeño del que ya estaba, es decir no lo mejora, el nodo E no se modifica.

Para F:

Como el nodo F ya está marcado (por el nodo H), primero se calcula el peso acumulado así:

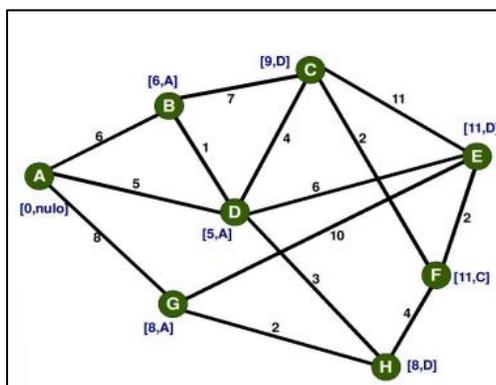
$$\text{acumuladoPeso} = 9 + 2 = 11;$$

como el valor acumulado de F es 12 y el anterior es menor (11) es decir que si lo mejora, se cambia el valor acumulado de F por 11 y el nodo antecesor por C por ser C quien lo mejoro.

Es de recordar que el numero 9 corresponde al acumuladoPeso del nodo antecesor.

Resultado la siguiente figura:

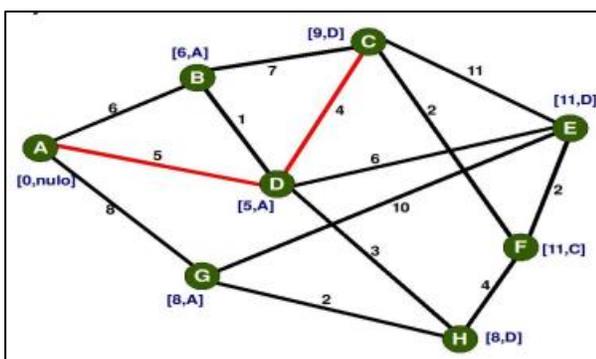
Figura 24: Análisis para decima segunda iteración.



Fuente: Adaptado por el Autor

Con la décima quinta iteración se termina de aplicar el algoritmo de Dijkstra a este grafo, ya que se marca todos nuestros nodos; ahora es posible identificar el camino más corto de un nodo cualquiera al nodo A el cual inicialmente fue seleccionado como nodo inicio. En ese sentido, de manera de ejemplo si se desea ver la ruta más corta del nodo C al nodo A. Primero se ubica el C y se verifica su nodo antecesor, gráficamente se observa que es el nodo D. Ahora se localiza en el nodo D y se define cuál es su nodo antecesor, gráficamente es el A. Como finalmente el nodo A no tiene antecesor significa que se ha llegado al nodo inicio. Fácilmente ver que la ruta es A, D y C.

Figura 25: Decima segunda iteración.



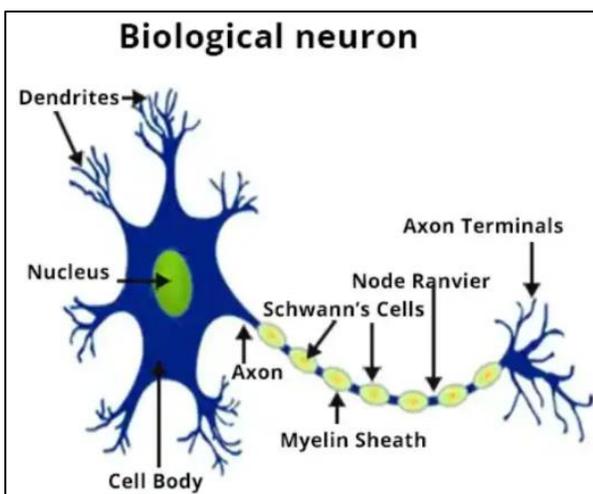
Fuente: Adaptado por el Autor

2.3. REDES NEURONALES

Las redes neuronales artificiales tratan de imitar el funcionamiento del cerebro. De acuerdo con Gallastegui (2022). Los equipos informáticos habituales trabajan exactamente al contrario que el cerebro: son máquinas de grano grueso, con muy escaso procesamiento paralelo. Únicamente tienen un pequeño número de “Cores” trabajando en paralelo, cada uno de ellos con una capacidad de cómputo mayor que una neurona. La cuestión es que 80000 millones de neuronas trabajando en paralelo son capaces de procesar información cotidiana más rápido que los computadores que conocemos. A esto hay que añadir otra cuestión muy importante: los algoritmos. Nuestros algoritmos tienden a resolver los problemas de manera secuencial, al igual que lo hacen los procesadores. Sin embargo, las situaciones cotidianas a las que nos enfrentamos: reconocimiento de imágenes, de sonidos, extracción de significados semánticos, aprendizaje, etc. Se formulan mejor emulando nuestros sistemas biológicos de aprendizaje: es decir, tomando el cerebro como modelo. Esto nos lleva a la necesidad de crear un modelo artificial de las neuronas biológicas.

En ese sentido resulta importante conocer previamente el funcionamiento de una neurona biológica para entender el modelo simplificado de neurona en el que se basan las redes neuronales artificiales. De acuerdo con Bobadilla (2020) en la Figura 26 se aprecia que el elemento central de una neurona es su núcleo; el núcleo se encarga de realizar el procesamiento de la información que le llega a la neurona. Esta información llega desde otras neuronas o, en algunas ocasiones, desde estímulos exteriores: nervio óptico, auditivo, etc. La información procesada se transmite a lo largo del axón hacia otras neuronas o hacia elementos biológicos del cuerpo, que finalmente provocan las reacciones a los estímulos de entrada: movimientos musculares, etc.

Figura 26: Neurona Biológica.



Fuente: Deloitte technology

Los terminales del axón se denominan sinapsis; son los encargados de realizar esta transferencia hacia las siguientes neuronas. El procesamiento simplificado de una neurona biológica es el siguiente:

- Inicialmente, el núcleo de la célula se encuentra en reposo, con una carga potencial de unos -70mV .
- Cuando llegan señales de la sinapsis de otras neuronas, el cuerpo de la célula va perdiendo carga negativa.
- Siguiendo el proceso anterior, el cuerpo de la célula llega a situarse con un potencial de unos $+30\text{mV}$
- Interviene un proceso que descarga esta tensión, devolviendo a la célula a su estado de reposo.

Figura 27: Comportamiento de una neurona.

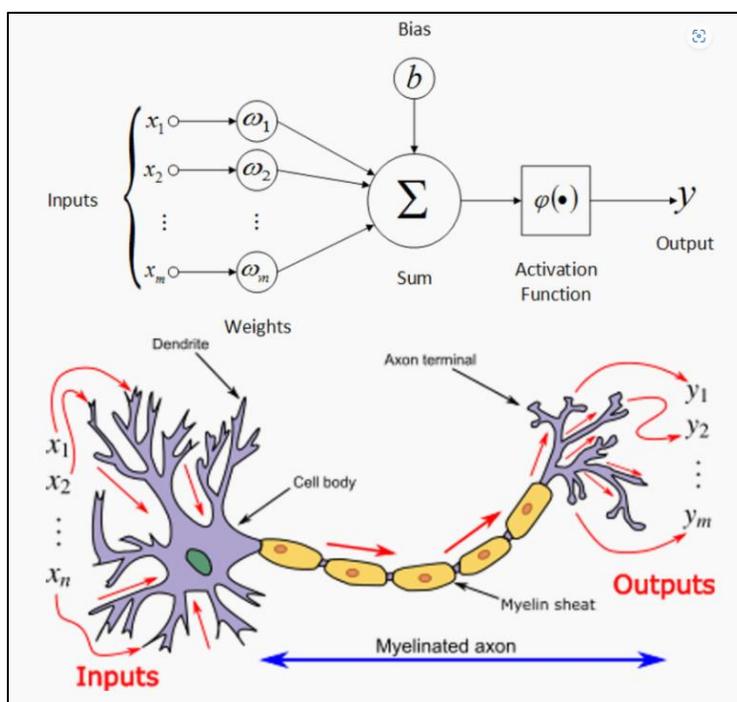


Fuente: Adaptado por el Autor

Los sistemas neuronales biológicos presentan una característica determinante para el aprendizaje y el almacenamiento de la información: los neurotransmisores. Los neurotransmisores controlan actúan químicamente sobre las sinapsis, amplificando o disminuyendo la cantidad de potencial que se transmite de una neurona a otra. De esta manera, existen sinapsis inhibitoras y sinapsis excitadoras. Esta característica permite variar la respuesta de las redes neuronales, soportando la evolución necesaria que requieren los mecanismos de aprendizaje.

Ahora es importante conocer el comportamiento de una neurona artificial, está es un modelo simplista de cómo actúa el cerebro, sin embargo día a día se sigue mejorando la estructuración y el comportamiento de los sistemas de redes neuronales artificiales para que se asimilen más al comportamiento del cerebro. En la Figura 28 se puede detallar una comparación de un modelo de neurona artificial vs el de una neurona biológica.

Figura 28: Esquema comparativo de red neuronal biológica vs neurona artificial.

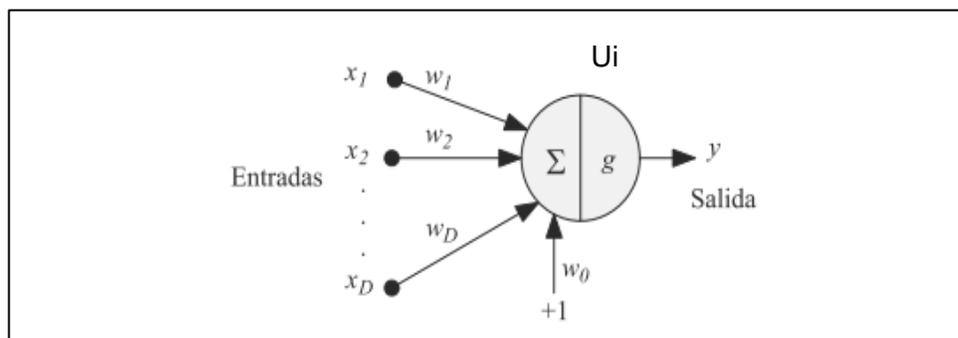


Fuente: Deloitte technology

La Figura 29 muestra el esquema de una neuronal artificial denominada perceptrón. Sus elementos son:

- U_i : Representa la neurona i -ésima
- y_i : Representa el resultado que genera la neurona i , equivalente a la salida que se produce en el axón de la neurona biológica.
- W_{ji} : Representa el valor de inhibición o excitación entre las neuronas u_j y u_i . Es el equivalente al efecto de los neurotransmisores sobre la sinapsis que une las neuronas u_j y u_i .

Figura 29: Modelo Neuronal de McCulloch-Pitts.



Fuente: Gallastegui, 2022

- Cuando $w_{ji} > 0$ se modela una sinapsis excitadora;
- cuando $w_{ji} < 0$ se modela una sinapsis inhibidora;
- cuando $w_{ji} = 0$ se modela ausencia de conexión
- a: Representa la suma de las señales que le llegan a la neurona u_i .

El primer paso para obtener la salida y_i de la neurona es calcular la suma ponderada a de las entradas, llamadas activación de la neurona.

$$a = \sum_{i=1}^D w_i * x_i + w_0$$

Ecuación 1

Donde w_0 es un umbral o sesgo que se utiliza para compensar la diferencia entre el valor medio de entradas, sobre todo el conjunto de entrenamiento, y el correspondiente valor medio de las salidas deseadas. Posteriormente, a partir de este valor a se obtiene la salida y de la neurona mediante la aplicación de una función, llamada función de activación o de transferencia $g(a)$, es decir:

$$y = g(a) = g\left(\sum_{i=1}^D wixi + wo\right) = g\left(\sum_{i=0}^D wixi\right)$$

Ecuación 2

- $g(a)$ Para representar la función de salida o transferencia. Es el equivalente al proceso que realiza el núcleo de la neurona biológica en función de la acumulación de los voltajes que le llegan: $g(a)i$.

Donde, como se observa, es posible tratar al umbral w_o como un peso más si se supone una entrada añadida x_o con un valor fijo de 1 (Figura 29). Finalmente, también es posible reescribir esta ecuación en notación vectorial como:

$$g(a) = g(W^T * X)$$

Ecuación 3

Si se toma W como el vector de pesos y X como el vector de entrada a la red.

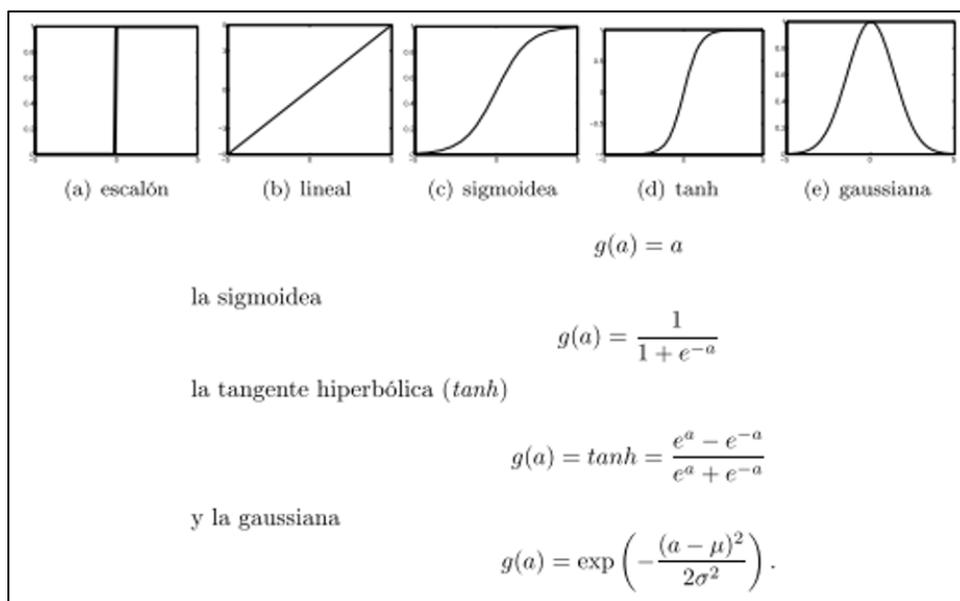
La función de transferencia empleada en esta modelo básica de McCulloch-Pitts es la función escalón definida por la ecuación:

$$f(x) = \begin{cases} 0, & a < 0 \\ 1, & a > 0 \end{cases}$$

Ecuación 4: Derivada de la función RELU. Fuente: Adaptado por el Autor

Y que se puede ver en la Figura (Figura 30). En los modelos actuales se escogen otro tipo de funciones, normalmente monótonas y derivables. Las más comunes, cuyas formas se muestran también en la Figura 30, son la función lineal, la sigmoidea, la tangente hiperbólica y la más utilizada en la mayoría de los casos en la nueva era de inteligencia artificial la función "RELU".

Figura 30: Algunas funciones de transferencia utilizadas.



Fuente: Gallastegui, Luis Miguel Garay, 2022

2.3.1. EL PERCEPTRÓN

Lo que se observó en la Figura 29 es lo que se conoce como Perceptrón el cuál es la red de neuronas artificiales más sencillas. Está compuesta únicamente por una capa de neuronas de entrada y otra capa de neuronas de salida. Su estudio es obligado dado que es la base de los métodos actuales.

El perceptrón es el primer modelo de red de neuronas artificiales (Rosenblatt, 1962.).El perceptrón consistía en una primera capa de j neuronas con funciones ϕ_j que se encargaban de transformar los datos entrada. Estas funciones reciben un subconjunto aleatorio de entradas a través de unos pesos fijos y les aplican una función de activación de tipo escalón. De nuevo como se ha mencionado anteriormente existe un peso especial o sesgo w_0 y de igual modo se definía una entrada artificial x_0 con valor 1 asociado a este peso, una entrada artificial X_0 con valor 1 asociado a este peso, en el caso del perceptrón se considera una función de activación extra $\phi_0=1$. La salida de esta primera capa de funciones ϕ_j se conecta a través de unos pesos a una última neurona para, finalmente, calcular la salida del perceptrón como:

$$y = g(a) = g\left(\sum_{j=0}^M w_j * \phi_j(x)\right) = g(W^T * \phi),$$

Ecuación 5

Donde ϕ es el vector formado por las funciones de activación ϕ_0, \dots, ϕ_M y g es la función escalón definida para los valores $\{-1, 1\}$.

La principal aportación del perceptrón es que la adaptación de sus pesos se realiza teniendo en cuenta el error entre la salida que obtiene la red y la salida que se desearía obtener. Es, por tanto, el primero método de aprendizaje supervisado. Para ilustrar este método, supóngase un problema de clasificación binaria (dos clases) en el que a cada vector de entrada X^n se le asocia un valor objetivo t^n , de modo que la salida deseada para la red es:

- $t^n = 1$, si X^n pertenece a la clase C1, y
- $t^n = -1$, si X^n pertenece a la clase C2

Teniendo en cuenta que cada vector de entrada X^n genera su correspondiente vector de activación ϕ^n y considerando las ecuaciones Ecuación 5 y Ecuación 2 se obtiene que, para que el perceptrón realice una clasificación correcta, deberá cumplir que:

- $w^T * \phi^n > 0$, si $X^n \in C1$
- $w^T * \phi^n < 0$, si $X^n \in C2$,

Lo que es lo mismo: $(w^T * \phi^n)t^n > 0$. Ello sugiere la minimización de la siguiente función de error conocida como el criterio del perceptrón:

$$E^{perc}(W) = - \sum_{\phi^n \in M} (W^T * \phi^n)t^n,$$

Ecuación 6

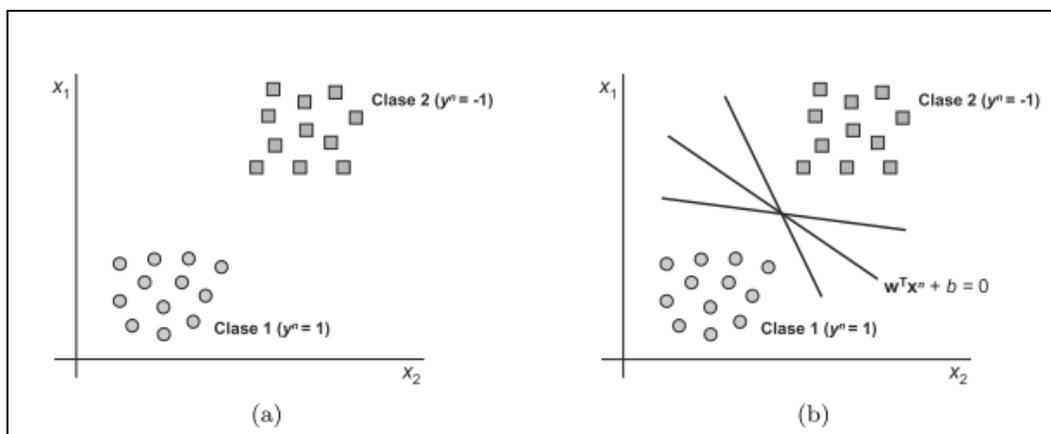
Donde M es el conjunto de vectores ϕ^n que han sido clasificados incorrectamente, de modo que $E^{perc} = 0$ cuando todos los vectores han sido adecuadamente clasificados.

Con el objetivo de determinar los pesos w que minimizan la función de error E se sigue un proceso adaptativo que consiste en comenzar con unos valores iniciales w aleatorios e ir modificándolos iterativamente cuando la salida del perceptrón no coincide con la salida deseada.

Ahora es importante recalcar de acuerdo con lo expresado por Gallastegui (2022): Conviene señalar que los perceptrones tienen varias limitaciones.

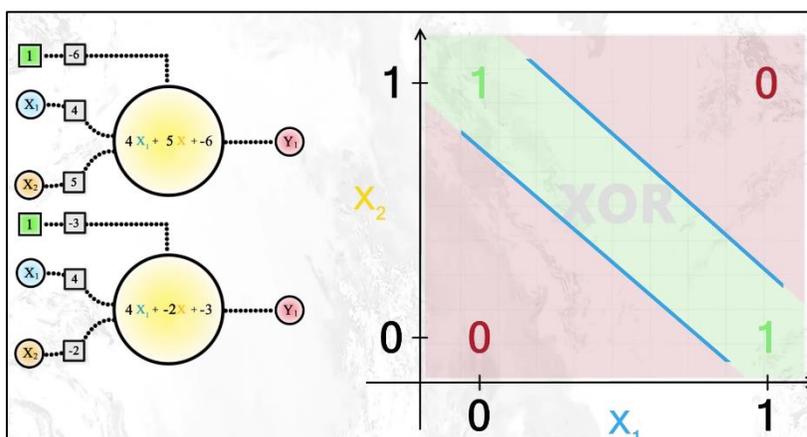
1. Primero, los valores de salida del perceptrón sólo pueden ser binarios, debido a la función de activación empleada (escalón).
2. Segundo, los perceptrones sólo pueden clasificar vectores linealmente separables, es decir, los pesos determinan zonas de clasificación separadas por un hiperplano. Así, en el caso de un perceptrón con 2 entradas, se clasificaría correctamente todos los patrones sólo si se pudiera establecer una línea recta entre las clases tal y como se aprecia en la Figura 31. En ese sentido se puede comprobar fácilmente, por ejemplo, su incapacidad para aprender la operación XOR (O exclusivo) que implica datos no separables linealmente, tal como se evidencia en la siguiente Figura 32. Es importante mencionar que el problema con la puerta XOR se solucionó aumentando a dos neuronas.

Figura 31: Problemas de clasificación linealmente separables.



Fuente: Gallastegui, Luis Miguel Garay, 2022

Figura 32: Puerta lógica XOR, imposible de resolver con 1 neurona.

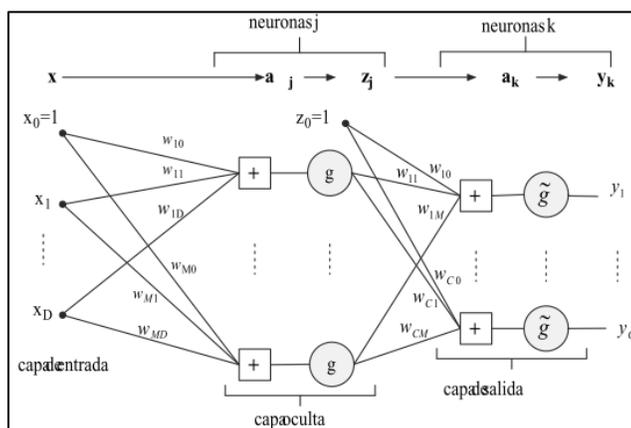


Fuente: Gallastegui, Luis Miguel Garay, 2022

2.3.2. PERCEPTRÓN MULTICAPA

Gallastegui (2022) menciona que el perceptrón simple es capaz de resolver problemas de clasificación e implementar funciones lógicas, por ejemplo la función OR, pero es incapaz de implementar otras funciones simples como la función XOR. Minsky y Papert publicaron un libro titulado Perceptrons (Minsky y Papert, 1969) en el que se exponían estas limitaciones, y que supuso el abandono por parte de muchos científicos de la investigación en redes neuronales, pues no se encontraba un algoritmo de aprendizaje capaz de implementar funciones de cualquier tipo. Las limitaciones de las redes de una sola capa hicieron que se plantease la necesidad de implementar redes en las que se aumentase el número de capas introduciendo capas intermedias entre la capa de entrada y la capa de salida, de manera que se pudiese implementar cualquier función con el grado de precisión deseado. La función que cumple dicha capa intermedia es tratar de realizar una proyección en la que resulten separables linealmente los patrones de entrada de manera que la unidad de salida pueda realizar una clasificación correcta. Surge así el perceptrón Multicapa, en adelante MLP (MultiLayer Perceptrons), cuya arquitectura se muestra en la siguiente figura

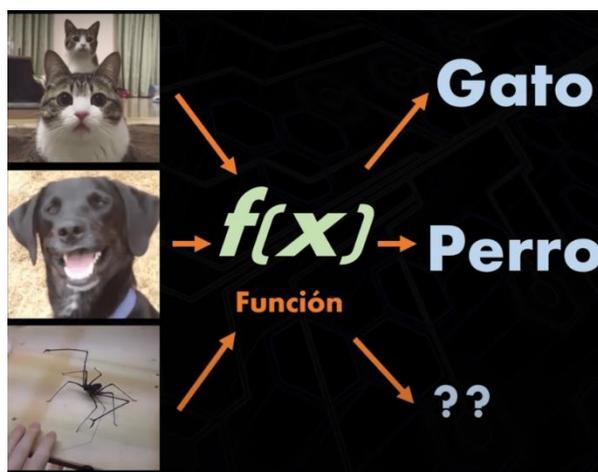
Figura 33: Arquitectura de un perceptrón multicapa con una capa oculta.



Fuente: Gallastegui 2022

Se puede establecer que un perceptrón multicapa son aproximadores de funciones, esto debido a algunos mecanismos que utiliza el cerebro como: Entrar una variable, procesarla y generar una respuesta. El siguiente ejemplo muestra esquemáticamente lo que se trata de explicar:

Figura 34: Explicación sobre MLP como aproximadores de funciones.



Fuente: Ringa Tech

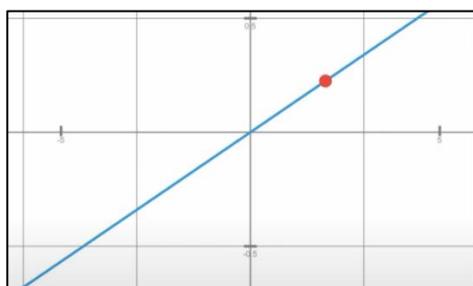
Como se visualizó en la red neuronal básica o perceptrón simple, su función de activación es la función escalón, esta función genera resultados lineales y es esta una de sus limitaciones para enfrentarse a problemas más complejos. El perceptrón simple con su función escalonada podría proporcionar ayuda para solucionar problemas de clasificación binaria, sin embargo es importante mencionar que debido a que se trata de modelar problemas de la vida cotidiana como por ejemplo el procesamiento de imágenes,

reconocimiento de voz, etc. El proceso de aprendizaje con una función de activación que genere resultados lineales no podría generar los resultados próximos a la realidad, esto debido a que, para responder problemas complejos se necesita más de una sola neurona, es decir un sistema de redes neuronales o lo que se conoce como “redes multicapa” o “perceptrón multicapa”. En sus comienzos el problema se modelaba con lo más básico como lo es el perceptrón.

En ese sentido cuando se quiere realizar un sistema de redes neuronales complejos utilizar una función de activación lineal para todas las neuronas el resultado es el mismo que si se utiliza un solo perceptrón ya que al final procesar muchas funciones lineales la respuesta final es una sola función lineal, por ende se caerá en el modelo básico del perceptrón.

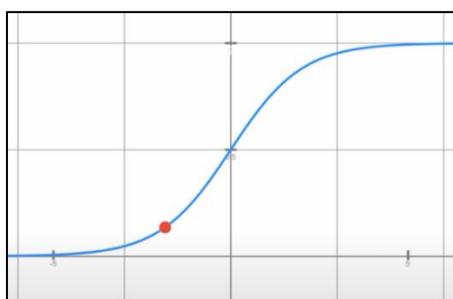
Las funciones de activación ayudan a salir del problema, su trabajo principal es agregar componentes no lineales a la red. Es decir: La neurona además de obtener una suma ponderada y agregar un sesgo, pasará el resultado por una función que llamamos función de activación y en ese punto pasará el resultado a las siguientes capas. Observa el siguiente ejemplo de una neurona sin funciones de activación y otra con una función de activación denominada función sigmoide-logística.

Figura 35: Neurona sin función de activación.



Fuente: Ringa Tech, 2022

Figura 36: Neurona que pasa por la función de activación sigmoide



Fuente: Ringa Tech, 2022

En la anterior función, Figura 35, si se cambia los valores de entrada, la salida también lo hará de manera constante por lo que forma una línea, ahora si se agrega una función de activación, por ejemplo la función logística o sigmoide (Figura 36) y el resultado pasa por dicha función antes de darlo a las siguientes capas, se puede ver que, al cambiar los valores de la entrada, la salida ya no cambia de manera constante formando una línea, en ese sentido es posible afirmar que la forma en la que la salida cambia según las entradas, depende de que función de activación estamos utilizando. Como se ha mencionado las funciones de activación son muy importante para generar redes neuronales, en ese sentido es obligatorio el estudio para determinar en el tipo de función que se utilizó en el presente proyecto, sin embargo antes de comenzar a estudiar las funciones de activación es relevante comprender el aprendizaje automático de una red neuronal actual.

2.3.2.1. APRENDIZAJE AUTOMATICO DE UNA RED NEUORNAL

Cuando se crea una red neuronal se necesita entrenarla antes de hacer predicción, durante el entrenamiento se asignan ejemplos que pasan por todas las capas de la red una por una hacia adelante realizando las operaciones de las neuronas y obtener un resultado al final. Después de varios ejemplos se utiliza la función de costo (Ecuación 7: Binary Crossentropy. Fuente: Ringa Tech, 2022) para definir el grado de acierto en sus predicciones. Según los resultados se debe ajustar los parámetros de la red para mejorar los **pesos w_j y sesgos b_j** . Para esto se calcula la derivada o gradiente de la función de costo respecto a cada uno de los pesos y sesgos de la red y se realiza capa por capa hacia atrás una por una hasta llegar al inicio este proceso es llamado **Backpropagation algorithm o propagación hacia atrás**.

2.3.2.2. Función de pérdida: Binary Crossentropy o entropía cruzada binaria:

Es una función utilizada comúnmente en problemas de clasificación binaria, donde solo hay dos clases posibles. Esta función calcula la diferencia entre las probabilidades predicas por el modelo y las etiquetas reales de los datos. Matemáticamente, la fórmula de la entropía cruzada binaria es la siguiente:

$$\text{Binary Crossentropy} = -\frac{1}{N} \sum_{i=1}^N [y_i * \log(P_i) + (1 - Y_i) * \log(1 - P_i)]$$

Ecuación 7: Binary Crossentropy. Fuente: Ringa Tech, 2022

Donde:

- N: es el número de ejemplos de entrenamiento.
- Y_i : es la etiqueta real del ejemplo i (**0** o **1** en problemas de clasificación binaria)

- P_i : es la probabilidad predicha por el modelo para el ejemplo i de pertenecer a la clase 1.

La función de pérdida Binary Crossentropy penaliza al modelo de manera más significativa cuando sus predicciones están lejos de las etiquetas reales.

- Si $Y_i = 1$ (el ejemplo pertenece a la clase 1), entonces el término $\log(P_i)$ penaliza al modelo si la probabilidad predicha P_i es baja, acercándola a 1.
- Si $Y_i = 0$ (el ejemplo pertenece a la clase 0), entonces el término $\log(1-P_i)$ penaliza al modelo si la probabilidad predicha P_i es alta, acercándola a 0.

En resumen, la función de pérdida Binary Crossentropy se utiliza para evaluar qué tan bien el modelo puede predecir probabilidades para problemas de clasificación binaria, penalizando las predicciones incorrectas y recompensando las predicciones precisas

2.3.2.3. ALGORITMO DE BACKPROPAGATION

Este algoritmo utiliza el descenso del gradiente para ajustar los pesos de la red. Además, dicho ajuste se realiza comenzando por la capa de salida, según el error cometido, y propagando este error a las capas anteriores, hasta llegar a la capa de las unidades de entrada, de ahí que se denomina algoritmo de retro propagación del error. Básicamente consiste en dos fases que se repiten hasta conseguir minimizar el error:

1. En la primera fase, o paso hacia adelante, se aplica un patrón de entradas de la red y su efecto se propaga a través de la misma, capa a capa. Finalmente, la red presenta un conjunto de salidas como respuesta a dicho patrón de entrada. Es decir, este paso consiste en el cálculo de la Ecuación 8 para un conjunto de pesos dado. Normalmente, en la primera iteración del algoritmo se parte de un conjunto de pesos con un valor inicial pequeño aleatorio
2. En la siguiente fase, o paso hacia atrás, los pesos de la red se recalculan de acuerdo con una regla de ajuste del error. Esto es, se calcula el valor de la función de error que compara la respuesta actual de la red y la respuesta deseada, y este error se propaga hacia atrás. A continuación se explica a detalle.

$$E^n = \frac{1}{2} \sum_{k=0}^c (e_k^n)^2,$$

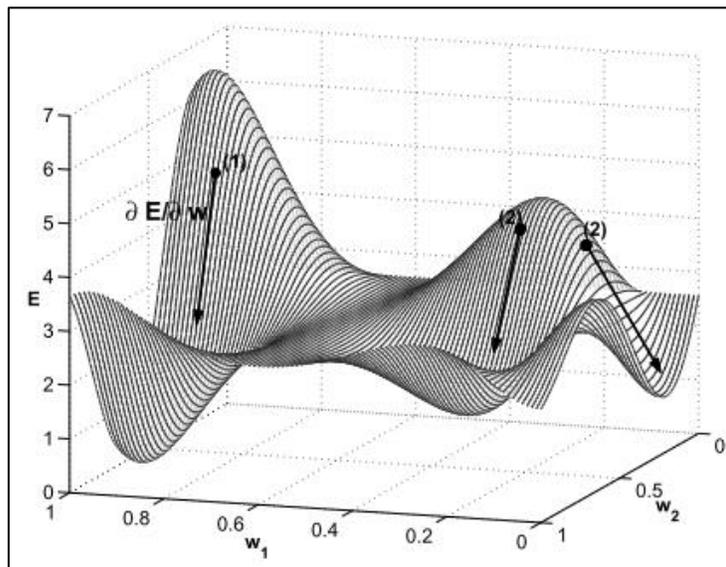
Ecuación 8: Función de Error. Fuente: Gallastegui, Luis Miguel Garay, 2022

Donde el factor $1/2$, que no altera el resultado de la función, se introduce para facilitar los cálculos posteriores, y el error e_k^n se define como:

$$e_k^n = y_k^n - t_k^n$$

Ecuación 9: Definición del error. Fuente: Gallastegui, Luis Miguel Garay, 2022

Figura 37: Descenso del gradiente en una red con únicamente 2 pesos.



Fuente: Gallastegui, 2022

En la Figura 37 se muestra un ejemplo de descenso gradiente en una red con únicamente 2 pesos. Las flechas indican direcciones en el espacio de búsqueda suponiendo distintos valores iniciales para los pesos w_1 y w_2 . El punto (1) conduce a un mínimo global mientras que los puntos (2) conducen a mínimos locales.

Si se supone que en la iteración n del algoritmo de aprendizaje en el que se ha introducido un patrón X^n . Como se ha visto en las reglas de aprendizaje, el algoritmo de retro propagación corrige el peso sináptico w_{kj} añadiéndole un incremento Δw_{kj} . En este caso, el incremento es proporcional al gradiente:

$$\frac{\partial E}{\partial w_{kj}}$$

Ecuación 10: Derivada parcial de la función de Error, respecto al peso de la red neuronal. Fuente: Carlos Santana, 2022

Dado que, si se considera la superficie de error que se forma al presentar E junto el espacio de pesos, este gradiente determina la dirección de la búsqueda en este espacio para

obtener el valor del peso w_{kj} que conduce a un mínimo de E. La Figura 37 muestra lo mencionado.

Observa que siguiendo la dirección indicada por el gradiente se conseguirá minimizar el error aunque no siempre se podrá alcanzar el resultado óptimo.

2.3.2.4. FORMULACIÓN MATEMÁTICA PARA EL APRENDIZAJE AUTOMÁTICO

Es importante comprender de manera general que el algoritmo Backpropagation es utilizado para calcular las derivadas parciales de cada uno de los parámetros de la red con respecto a la función de costo (función de error). Ahora el método del descenso del gradiente es utilizado para optimizar a la red neuronal, en pocas palabras para funcionar el método del descenso del gradiente se necesita del vector gradiente el cual está compuesto por las derivadas parciales de los parámetros de una red con respecto al error o función de costo cuando se evalúa la red completa. Explicación a continuación

1. Como primer paso se supone una red de neuronas con varias capas que posee parámetros inicializados de forma aleatoria, en ese sentido cuando la red genere una predicción y se compare con el valor real, se observa que la función de costo le asigna un valor de error elevado, en ese sentido ese error será importante para entrenar a la red.

Figura 38: Ejemplo de estructura de red neuronal.



Fuente: Carlos Santana Vega, 2022

2. Como segundo paso se debe pensar lo siguiente: ¿Cómo varía el costo o error ante un cambio del parámetro w_i ? Esto hace formular la siguiente derivada parcial: Cómo varía el costo respecto a cada uno de los parámetros de la red. En cuanto a los parámetros de una red es importante mencionar que existen 2 tipos: El parámetro

w que representa el peso de cada entrada y el parámetro de b bias o sesgo que es un dato inicial que se posiciona en la red. En ese sentido se debe calcular 2 derivadas parciales.

$$\frac{\partial c}{\partial w} \text{ y } \frac{\partial c}{\partial b}$$

Ecuación 11: Derivadas parcial del coste respecto al peso y respecto al sesgo respectivamente.

Fuente: Por el Autor

- Ahora se va a trabajar hacia atrás es decir implementado el algoritmo de Backpropagation. Se deberá calcular la derivada parcial de la última capa y se irá realizando operaciones hasta llegar a la capa de entrada.

Figura 39: Ejemplo de estructura de red neuronal.



Fuente: Carlos Santana Vega, 2022

Dentro del análisis es importante conocer los parámetros que influyen en la derivada parcial, estos parámetros son:

- Resultado de la suma ponderada: Este resultado es la suma de pesos w para cada entrada x y la suma del sesgo b . Para referirnos a este parámetro se lo hará de la siguiente manera:

$$Z^L = W^L X + b^L$$

Ecuación 12: Suma ponderada antes de entrar a la función de activación. Fuente: Carlos Santana, 2022

- Z sería luego pasada por la función de activación y a la que se llamará como función de activación " a ":

$$a = (Z^L)$$

Ecuación 13: Función de activación sobre la suma ponderada. Fuente: Carlos Santana, 2022

- El resultado de las activaciones de las neuronas en la última capa conformaría el resultado de la red que luego sería evaluado por la función de Costo “**c**”, para determinar así el Error de la red.

$$C(a(Z^L))$$

Ecuación 14: Composición de Funciones. Fuente: Carlos Santana, 2022

La anterior expresión donde el resultado de una función es pasado por otra función y así sucesivamente, es lo que se conoce como composición de funciones. Y para calcular la derivada de una composición de funciones hay que hacer uso de una herramienta del cálculo matemático denominado “**chain rule**” o regla de la cadena. Para comprender la regla de la cadena se hace uso del siguiente ejemplo que proporciona Carlos Santana Vega (2022): “Si deseamos conocer que tan rápido es una liebre en comparación a una tortuga y se conoce que tan rápido es la liebre con respecto al perro y el perro que tan rápido respecto a la tortuga, entonces, la regla de la cadena nos dice”:

$$\frac{\partial \text{liebre}}{\partial \text{Perro}} = 3, \frac{\partial \text{Perro}}{\partial \text{Tortuga}} = 5, \text{Entonces } \frac{\partial \text{liebre}}{\partial \text{Tortuga}} = 3 * 5 = 15$$

En ese sentido lo que indica la regla de la cadena es que para calcular la derivada de una composición de funciones, se debe multiplicar cada una de las derivadas intermedias.

$$\frac{\partial \text{liebre}}{\partial \text{Tortuga}} = \frac{\partial \text{liebre}}{\partial \text{Perro}} * \frac{\partial \text{Perro}}{\partial \text{Tortuga}}$$

Aplicando lo anterior al problema se tiene que: Al determinar las derivadas parciales del costo respecto a los pesos y sesgos, influyen a través de la siguiente composición en lo que se había descrito anteriormente:

$$\frac{\partial c}{\partial w^L} \text{ y } \frac{\partial c}{\partial b^L}$$

$$Z^L = W^L X + b^L, \quad C(a(Z^L))$$

En ese sentido aplicado la regla de la cadena se deberá calcular todas las derivadas intermedias:

$$\frac{\partial c}{\partial w^L} = \frac{\partial c}{\partial a^L} * \frac{\partial a}{\partial z^L} * \frac{\partial z}{\partial w^L} \quad (1)$$

$$\frac{\partial c}{\partial b^L} = \frac{\partial c}{\partial a^L} * \frac{\partial a}{\partial z^L} * \frac{\partial z}{\partial b^L} \quad (2)$$

$$Z^L = W^L a^{L-1} + b^L, \quad C(a^L(Z^L))$$

Resolviendo cada derivada parcial se tendrá que:

$$\frac{\partial z^L}{\partial b^L}:$$

$$\frac{\partial z^L}{\partial w^L}:$$

$$\sum_i a_i^{L-1} * w_i^L + b^L$$

$$\frac{\partial z^L}{\partial w^L} = a_i^{L-1}$$

$$\frac{\partial z^L}{\partial b^L} = 1$$

Las derivadas parciales de: $\frac{\partial c}{\partial a^L}, \frac{\partial a}{\partial z^L}$ son la derivada de la función de costé o la función de error respecto a la variación de la función de activación y la segunda ecuación sería la derivada de la función de activación respecto a la suma ponderada. Por ende estas expresiones son fáciles de obtener si se tiene la función de coste y la función de activación. En este punto se recuerda de lo importante que la función de activación sea derivable.

Con las expresiones previamente resueltas y además teniendo en cuenta lo siguiente:

$$\delta^L = \frac{\partial c}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L}$$

Se reestructura las ecuaciones para la última capa de la neurona:

$$\frac{\partial c}{\partial w^L} = \delta^L * a_i^{L-1} \quad (1)$$

$$\frac{\partial c}{\partial b^L} = \delta^L \quad (2)$$

Finalmente, ahora para retroceder a la capa L-1, se sigue en el mismo procedimiento que la capa L, resultando en las siguientes ecuaciones:

$$\frac{\partial c}{\partial w^{L-1}} = \frac{\partial c}{\partial a^L} * \frac{\partial a}{\partial z^L} * \frac{\partial z}{\partial a^{L-1}} * \frac{\partial a^{L-1}}{\partial z^{L-1}} * \frac{\partial z^{L-1}}{\partial w^{L-1}} \quad (1)$$

$$\frac{\partial c}{\partial b^L} = \frac{\partial c}{\partial a^L} * \frac{\partial a}{\partial z^L} * \frac{\partial z}{\partial a^{L-1}} * \frac{\partial a^{L-1}}{\partial z^{L-1}} * \frac{\partial z^{L-1}}{\partial w^{L-1}} \quad (2)$$

Teniendo en cuenta que:

$$\frac{\partial c}{\partial a^L} * \frac{\partial a}{\partial z^L} = \delta^L: \text{Error de la capa L}$$

$$\frac{\partial z}{\partial a^{L-1}} = W^L: \text{Matriz de parametros } W \text{ que conecta ambas capas}$$

$$\frac{\partial a^{L-1}}{\partial z^{L-1}}: \text{Derivada de la función de activación}$$

Repitiendo el proceso de la capa L para las derivadas parciales en la capa L-1, se obtiene:

$$\frac{\partial z^{L-1}}{\partial w^{L-1}} = a_i^{L-2}$$

$$\frac{\partial z^{L-1}}{\partial b^{L-1}} = 1$$

Teniendo en cuenta que:

$$\frac{\partial c}{\partial a^L} * \frac{\partial a}{\partial z^L} * \frac{\partial z}{\partial a^{L-1}} * \frac{\partial a^{L-1}}{\partial z^{L-1}} = \delta^{L-1}$$

$$\delta^L * W^L * \frac{\partial a^{L-1}}{\partial z^{L-1}} = \delta^{L-1}$$

Se pueden reescribir las ecuaciones 1 y 2 para la capa L-1

$$\frac{\partial c}{\partial w^{L-1}} = \delta^{L-1} * a_i^{L-2} \quad (1)$$

$$\frac{\partial c}{\partial b^L} = \delta^{L-1} \quad (2)$$

Así sucesivamente se puede ir recorriendo todas las capas de la red hasta el final, Con esto se calcula todos los errores y las derivadas parciales de la red haciendo uso de las siguientes cuatro expresiones:

$$\delta^L * W^L * \frac{\partial a^{L-1}}{\partial z^{L-1}} = \delta^{L-1} \quad (1)$$

Ecuación 15: Parámetro para calcular el error en la capa L-1. Fuente: Carlos Santana Vega, 2022

$$\delta^L = \frac{\partial c}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \quad (2)$$

Ecuación 16: Parámetro para calcular el error en la última capa. Fuente: Carlos Santana Vega, 2022

$$\frac{\partial c}{\partial w^{L-1}} = \delta^{L-1} * a_i^{L-2} \quad (3)$$

Ecuación 17: Como cambia la función de coste respecto a los pesos en la capa L-1. Fuente: Adaptado de Carlos Santana Vega, 2022.

$$\frac{\partial c}{\partial b^{L-1}} = \delta^{L-1} \quad (4)$$

Ecuación 18: Como cambia la función de coste respecto a los sesgos en la capa L-1. Fuente: Adaptado de Carlos Santana Vega, 2022

Es de anotar que las funciones de activación son importantes dentro del entrenamiento de una red neuronal y por ende amerita el estudio de algunas funciones más utilizadas

2.3.2.5. Funciones de activación en una red neuronal

De acuerdo con Tech (2022), las funciones de activación ayudan a salir de la linealidad de las respuesta. Algunas funciones que son las siguientes:

1. Función Escalón: Esta función genera el componente no lineal que se necesita, sin embargo no es apta para el aprendizaje automático, ya que, como se ha estudiado en la formulación matemática al utilizar el algoritmo de Backpropagation que calcula las derivadas parciales y que este a su vez hace uso del método del descenso del gradiente para optimizar los parámetros de la red, en una de las derivadas parciales se involucra la derivada de la función de activación y como se visualizó esta función de activación en el valor de cero no tiene derivada.

Figura 40: Función Escalón.



Fuente: Ringa Tech, 2022

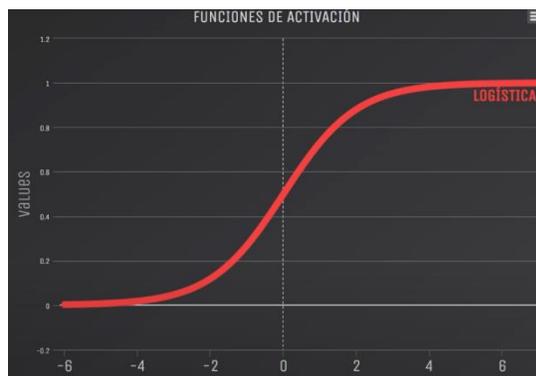
2. Funciones Sigmoides.

- a. **Función Logística:** Esta función acota los valores positivos a un valor de 1 y valores negativos a un valor de 0. Esta función si es derivable sin embargo presenta un problema y es que para valores grandes siempre será 1 y para valores pequeños siempre será 0, esto genera el problema “desvanecimiento de gradiente” es decir la derivada se vuelve pequeña, por ende se obtiene cambios muy pequeños conforme se pasa capa por capa, causando que el entrenamiento será muy pequeño. Esta función es muy utilizada en problemas de clasificación binaria.

$$f(x) = \frac{1}{1 + e^{-x}}$$

Ecuación 19: Función de activación Logística. Fuente: Ringa, Tech, 2022

Figura 41:: Función Logística.



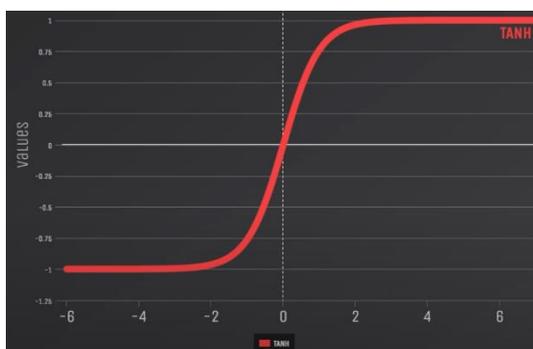
Fuente: Ringa Tech, 2022

- b. **Función Tangente Hiperbólica:** Esta función esta acotada entre valores de 1, -1. En casi todos los casos esta función es mayor que la función logística. La derivada de esta función es mucho mayor que la derivada de la función logística, es decir al tener una derivada mayor el aprendizaje será mayor. Sin embargo esta función también presenta el problema que presenta la función logística, es decir el “Desvanecimiento de gradiente”: Para valores mayores el valor se hace 1 y para valores pequeños se hace -1. Además que el tiempo computacional para esta función es mayor ya que su derivada es más compleja que la función logística.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Ecuación 20: Función Tangente Hiperbólica. Fuente: Ringa Tech, 2022

Figura 42: Función TANH.



Ringa Tech, 2022

3. Función ReLU (Unidad lineal rectificada): Esta función es la más importante hasta el momento, respecto a la mayoría de las funciones de activación previamente vistas. Esta función es muy simple: Regresa 0 para valores pequeños y X para valores mayores a 0. La derivada de esta función es simple:

$$f'(x) = \begin{cases} 0, & x < 0 \\ 1, & x > 0 \end{cases}$$

Ecuación 21: Valores de la derivada de la función RELU. Fuente: Adaptada por el Autor

$$\max(0, x)$$

Figura 43:: Función ReLU.



Fuente: Ringa Tech, 2022

Esta función presenta el problema de “Desvanecimiento de gradiente” para valores menores a 0

Capítulo 3. Metodología

La determinación de la posible fuente de extracción por ser explotada es el resultado del entrenamiento de la base de datos capturada por el CDT Vial de la universidad del Cauca con la creación del software Canteras-Móvil e implementado redes neuronales artificiales a través de la creación del software Canteras-Web.

Este proceso se ha realizado siguiendo una serie de actividades, teniendo claro los pasos de la planeación ejecutada y en general aplicando una metodología que permita cumplir con el objetivo de la presente investigación descrita a continuación.

3.1. RECOLECCIÓN DE DATOS PARA CONSTRUCCIÓN DE LOS SOFTWARES

Para la información que se ha capturado de las fuentes de extracción mediante la creación del software Canteras-Móvil, se ha implementado un formulario para la entrada de datos dentro del software, que posee la siguiente información:

- Tipo de extracción, es decir si la fuente de extracción es de Río, Cantera o Acopio
- Nombre de la Fuente
- Observación de la fuente: Esta es una observación que se realiza por el especialista encargado de capturar la información. Algunas observaciones que se han manejado en la base de datos preliminar sobre las fuentes de extracción son: si el material a extraer es:
 - Arena
 - Balastro
 - Material de Afirmado
 - Material de Rio
 - Roca
 - Roca Muerta
 - Roca Meteorizada, y otras.

La información acerca de fuentes de extracción capturadas por el CDT Vial de la Universidad del Cauca mediante Canteras-Móvil, permite tener una base de datos para el funcionamiento de algunas herramientas del software Canteras-Web, ya que, en se puede navegar a través de la herramienta de búsqueda de fuentes de extracción mediante la visualización de imágenes satelitales de alta resolución. A parte de capturar la información en campo para las fuentes de extracción, también es importante capturar la información de

los ensayos y la comparación de información respecto a requisitos como material de afirmado exigido de acuerdo con el Artículo 311-13 del Instituto Nacional de Vías.

Tabla 1: Requisitos de los agregados para afirmados.

CARACTERÍSTICA	NORMA DE ENSAYO INV	REQUISITO
Dureza (O)		
Desgaste en la máquina de los Ángeles (Gradación A), máximo (%) - 500 revoluciones	E-218	50
Durabilidad (O)		
Pérdidas en ensayo de solidez en sulfatos, máximo (%) - Sulfato de sodio - Sulfato de magnesio	E-220	12 18
Limpieza (F)		
Límite líquido, máximo (%)	E-125	40
Índice de plasticidad (%)	E-125 y E-126	4 - 9
Contenido de terrones de arcilla y partículas deleznales, máximo (%)	E-211	2
Contracción lineal	E-127 o E-129	Tabla 311 - 3
Resistencia del material (F)		
CBR (%): porcentaje asociado al grado de compactación mínimo especificado (numeral 311.5.2.2.2); el CBR se medirá sobre muestras sometidas previamente a cuatro días de inmersión.	E-148	≥ 15

Fuente: Artículo 311-E Invias

Tabla 2: Franjas granulométricas del material de afirmado.

Tabla 311 - 2. Franjas granulométricas del material de afirmado

TIPO DE GRADACIÓN	TAMIZ (mm / U.S. Standard)							
	37.5 1 1/2"	25.0 1"	19.0 3/4"	9.5 3/8"	4.75 No. 4	2.00 No. 10	0.425 No. 40	0.075 No. 200
% PASA								
A-38	100	-	80-100	60-85	40-65	30-50	13-30	9-18
A-25	-	100	90-100	65-90	45-70	35-55	15-35	10-20
Tolerancias en producción sobre la fórmula de trabajo (±)	0%	7%		6%			3%	

Fuente: Artículo 311-E Invias.

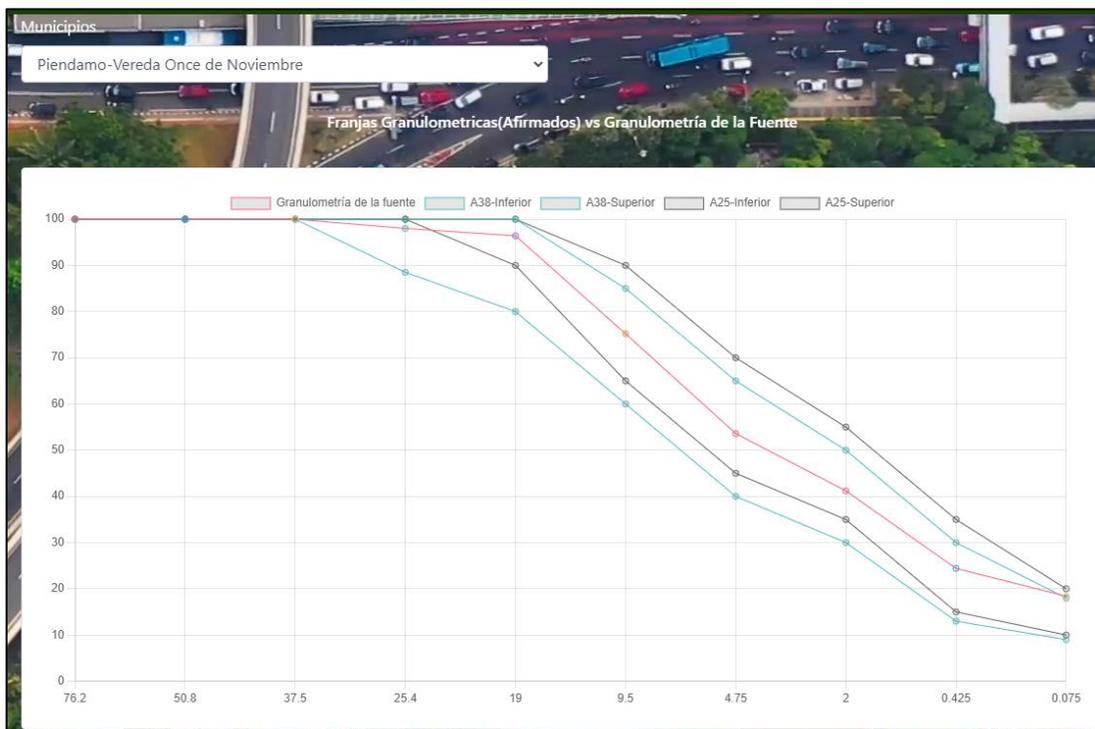
Figura 44: Datos capturados con Canteras-Móvil y visualizados en Canteras-Web.

Municipio	Requisito de CBR	Desgaste en la máquina de los Angeles para NT1	Requisito del Desgaste M.A	Angularidad de Finos sin compactar	Limite Liquidido
El Tambo	15,0	52,0	50,0	49,8	40,8
El Tambo	15,0	56,0	50,0	46,5	26,4
Totoro	15,0	71,0	50,0	62,6	41,6
Totoro	15,0	52,0	50,0	53,8	43,7
Rosas	15,0	60,0	50,0	55,4	32,8
Rosas	15,0	38,0	50,0	54,7	34,5
Bolívar	15,0	41,0	50,0	54,5	41,1
Piendamó	15,0	44,0	50,0	60,3	53,2
Popayán	15,0	60,0	50,0	64,8	43,8
Popayán	15,0	33,0	50,0	48,8	30,8

Municipio	Vereda	Observación de la Fuente	Otras Observaciones	CBR	Requi de CB
El Tambo	Magines	Roca Muerta	Color amarillo	50,7	15,0
El Tambo	Corregimiento cuatro esquinas	Roca Muerta	Color amarillo	25,8	15,0
Totoro	Vereda Sabaleta	Roca	Segunda punto de toma , Grado de meteorización III a IV	65,7	15,0
Totoro	Vereda San Antonio	Material altamente meteorizado	Dearman tipo V a VI	51,9	15,0
Rosas	Vereda las Trincheras	Roca	Roca Meteorizada	40,1	15,0
Rosas	Vereda Ufugu	Roca	Roca Meteorizada	50,2	15,0
Bolívar	Vereda la medina	Macizo Rocoso	Roca parcialmente meteorizada	30,3	15,0
Piendamó	Vereda Once de Noviembre	Macizo Rocoso	Altamente alterado	53,8	15,0
Popayán	Vereda Quintana	Agregados Petreos	Material para mejoramiento de vías	32,7	15,0
Popayán	Vereda el Agrado	Macizo Rocoso	visible de meteorización	26,7	15,0

Fuente: Por el Autor

Figura 45: Visualización de Franjas granulométricas de las respectivas fuentes de extracción en Canteras-Web.



Fuente: Por el Autor

La base de datos con información de campo y ensayos de laboratorio se ha registrado previamente, y está permitiendo continuar al siguiente proceso, el cual es el entrenamiento de redes neuronales artificiales y la creación del grafo necesario para obtener rutas mínimas de acarreo o rutas mínimas de visita a diferentes fuentes. Como primera instancia se obtuvo la siguiente base de datos con la información descrita anteriormente.

Figura 46: Datos de Fuentes de Extracción en PostgreSQL.

public.Dashboard_dasboard/CANTERAS/postgres@PostgreSQL...

Query Query History Scratch Pad

```
1 SELECT * FROM public."Dashboard_dasboard"
2 ORDER BY id ASC
```

Data Output Messages Notifications

	tipoextraccion	nombre_fuente	observacion_fuente	otras_observaciones	codigo	fecha	coordenada_Latitud
	character varying (150)	text	text	text	text	date	double precision
1	Cantera	Magines	Roca Muerta	Color amarillo	nan	2022-07-22	2.4000194
2	Cantera	Chisquio	Roca Muerta	Color amarillo	nan	2022-07-22	2.475211
3	Cantera	Santa Bárbara	Roca meteorizada	Grado de meteorización II a III	nan	2022-08-18	2.7895333
4	Cantera	Zabaletas	Roca	Segunda punto de toma , Grado de meteorización III a ...	nan	2022-09-02	2.499
5	Cantera	San Antonio	Material altamente meteorizado	Dearman tipo V a VI	nan	2022-09-02	2.4967222
6	Cantera	Las Trincheras	Roca	Roca Meteorizada	nan	2022-07-29	2.2825
7	Cantera	Ufugu	Roca	Roca Meteorizada	nan	2022-07-29	2.2216944
8	Rio	Capellania	Material de Rio	aparentemente bien gradado	nan	2022-08-05	1.9249969
9	Cantera	El colerin	Macizo Rocoso	Roca parcialmente meteorizada	nan	2022-08-05	1.9162222
10	Cantera	Once de Noviembre	Macizo Rocoso	Altamente alterado	nan	2022-08-10	2.661955
11	Cantera	Belencito	Agregados Petreos	Material para mejoramiento de vías	nan	2022-08-11	2.4355278
12	Cantera	Los Cerrillos	Macizo Rocoso	visible de meteorización	nan	2022-08-11	2.563538
13	Cantera	La Cuchilla	Agregados Petreos	Meteorización IV a VI	nan	2022-08-18	2.84365
14	Cantera	San Pedro Bajo	Macizo Rocoso	Roca dura y resistente ante la excavación	nan	2022-08-24	2.1736738
15	Cantera	Hierbas buenas	Agregados Petreos	visible de meteorización	nan	2022-08-30	2.1934069
16	Cantera	San Roque	Macizo Rocoso	Roca dura y resistente ante la excavación	nan	2022-08-30	2.1705833
17	Cantera	Vereda La Esmeralda	Macizo Rocoso	Grado de meteorización III a IV	nan	2023-01-27	2.76775

Total rows: 23 of 23 Query complete 00:00:00.303 Ln 1, Col 1

public.Dashboard_dasboard/CANTERAS/postgres@PostgreSQL...

Query Query History Scratch Pad

```
1 SELECT * FROM public."Dashboard_dasboard"
2 ORDER BY id ASC
```

Data Output Messages Notifications

	coordenada_Longitud	municipio	vereda	zona	Distancia_Cabecera	Tiempo_Viaje_Cabecera	Dureza_maquina
	double precision	character varying (150)	character varying (255)	character varying (10)	double precision	double precision	double precision
1	-76.8755808	El Tambo	Magines	Centro	23.7	70	
2	-76.852986	El Tambo	Corregimiento cuatro esquinas	Centro	8.2	25	
3	-76.7049944	Morales	Vereda Santa Barbara	Centro	11.4	31	
4	-76.3509444	Totoro	Vereda Sabaleta	Oriente	7	15	
5	-76.4519722	Totoro	Vereda San Antonio	Oriente	8.7	14	
6	-76.7425555	Rosas	Vereda las Trincheras	Centro	9	20	
7	-76.6988889	Rosas	Vereda Ufugu	Centro	7	15	
8	-77.09344	Bolivar	Vereda Capellania	Sur	45	103	
9	-76.9994444	Bolivar	Vereda la medina	Sur	25.9	48	
10	-76.6081439	Piendamó	Vereda Once de Noviembre	Centro	23.3	49	
11	-76.4492778	Popayán	Vereda Quintana	Centro	20.7	60	
12	-76.762308	Popayán	Vereda el Agrado	Centro	40	85	
13	-76.6720694	Morales	Vereda Santa la Cuchilla	Centro	12.7	33	
14	-76.6782428	La Sierra	San Pedro Bajo	Centro	11.5	33	
15	-76.6338589	Sotará	Vereda Hierbas buenas	Centro	25.2	71	
16	-76.6098333	Sotará	Vereda San Roque	Centro	21.1	59	
17	-76.4555694	Caldono	Vereda la Esmeralda	Norte	10	30	

Total rows: 23 of 23 Query complete 00:00:00.303 Ln 1, Col 1

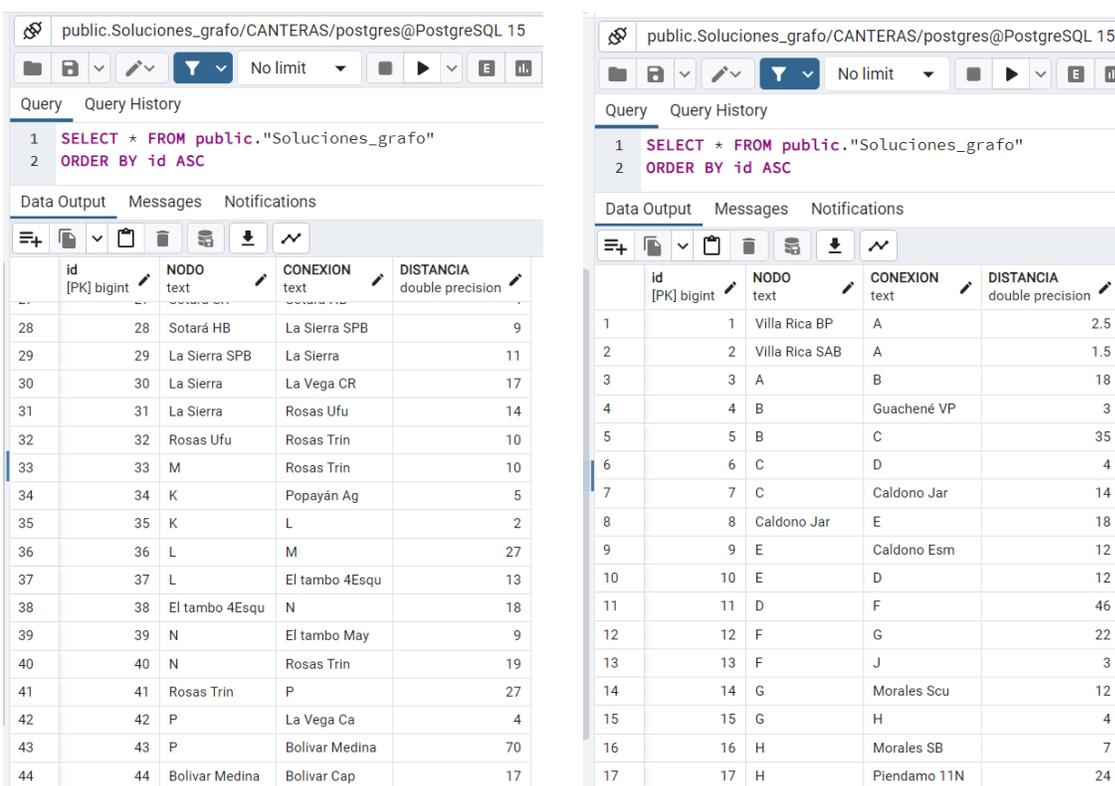
Fuente: Por el Autor

3.2. CONSTRUCCIÓN DE GRAFO

Para obtener el grafo, se ha recolectado información sobre los accesos a las fuentes de extracción de la base de datos. Esta información se ha recolectado con la base de datos preliminar suministrada por la Oficina de Infraestructura de la Gobernación del Cauca, a través del CDT Vial de la Universidad del Cauca, también se ha recopilado información en

la base de datos de Google maps. Es importante mencionar que dentro de la base de datos capturada por el CDT Vial a través del software Canteras-Móvil existe una columna llamada “Distancia” la cual explica la distancia de recorrido desde la fuente de extracción hasta la cabecera municipal a la que pertenece, así como también una columna de Tiempo, que indica el tiempo desde la fuente de extracción hasta la cabecera municipal. Con la información recolectada en la base de datos nueva, la base de datos preliminar y información de Google maps, se ha determinado los diferentes accesos a los nodos que en este caso representan las fuentes de extracción.

Figura 47: Información procesada para construcción del Grafo para visitas técnicas a las Fuentes.



id [PK] bigint	NODO text	CONEXION text	DISTANCIA double precision
28	Sotará HB	La Sierra SPB	9
29	La Sierra SPB	La Sierra	11
30	La Sierra	La Vega CR	17
31	La Sierra	Rosas Ufu	14
32	Rosas Ufu	Rosas Trin	10
33	M	Rosas Trin	10
34	K	Popayán Ag	5
35	K	L	2
36	L	M	27
37	L	El tambo 4Esqu	13
38	El tambo 4Esqu	N	18
39	N	El tambo May	9
40	N	Rosas Trin	19
41	Rosas Trin	P	27
42	P	La Vega Ca	4
43	P	Bolivar Medina	70
44	Bolivar Medina	Bolivar Cap	17

id [PK] bigint	NODO text	CONEXION text	DISTANCIA double precision
1	Villa Rica BP	A	2.5
2	Villa Rica SAB	A	1.5
3	A	B	18
4	B	Guachené VP	3
5	B	C	35
6	C	D	4
7	C	Caldono Jar	14
8	Caldono Jar	E	18
9	E	Caldono Esm	12
10	E	D	12
11	D	F	46
12	F	G	22
13	F	J	3
14	G	Morales Scu	12
15	G	H	4
16	H	Morales SB	7
17	H	Piendamó 11N	24

Fuente: Por el Autor

3.2.1. Nodos para el análisis

Para el análisis de ruta corta o ruta mínima desarrollado mediante el algoritmo de Dijkstra. Se ha tomado cómo nodos del grafo la información de fuentes de extracción que presentan ensayos de laboratorio, ya que, de estas fuentes de extracción se ha recuperado información de distancias, tiempos y demás información, además de que esta información será la que se utilizará para el entrenamiento de la red neuronal artificial con el fin de

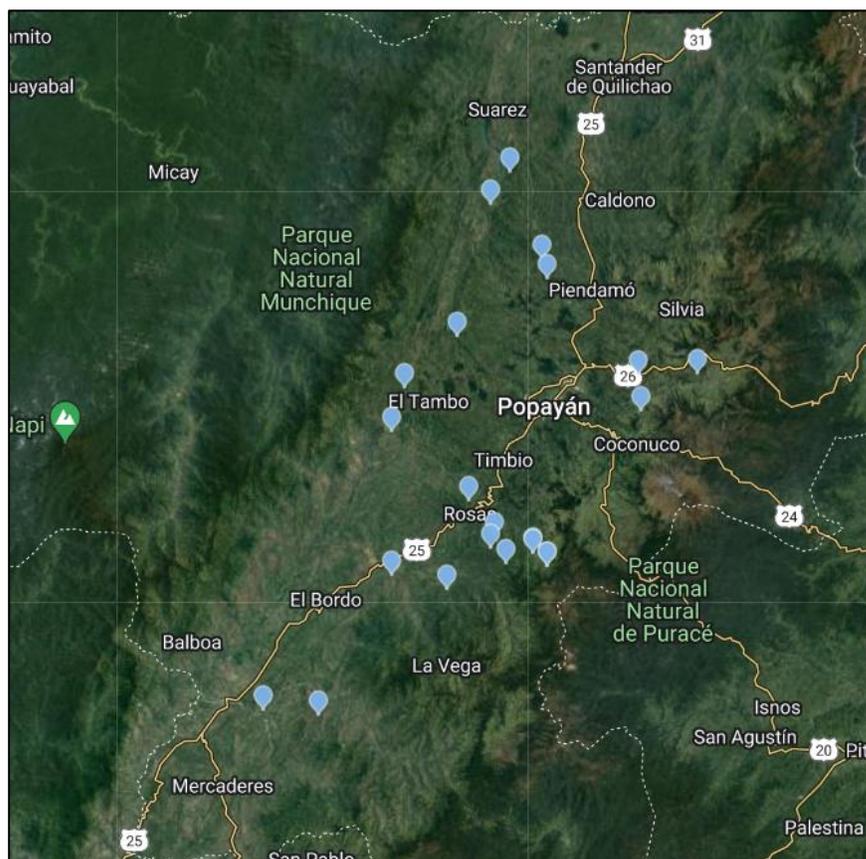
predecir mediante el modelo ARN, la elección de las posibles fuentes de extracción a explotar. En ese sentido la base de datos a utilizar es la que se ha capturado en campo mediante el software Canteras-Móvil por el CDT Vial de la universidad del Cauca y se concluyó en la siguiente base de datos:

Tabla 3: Datos para Grafo.

tipoextraccion	Coord_Latitud	Coord_Longitud	Municipio	Vereda	Distancia_Cabecera	Tiempo_Viaje_Cabecera
Cantera	2.40001940	-76.87558080	El Tambo	Magines	23.70	70.00
Cantera	2.47521100	-76.85298600	El Tambo	Corregimiento cuatro esquinas	8.20	25.00
Cantera	2.78953330	-76.70499440	Morales	Vereda Santa Barbara	11.40	31.00
Cantera	2.49900000	-76.35094440	Totoro	Vereda Sabaleta	7.00	15.00
Cantera	2.49672220	-76.45197220	Totoro	Vereda San Antonio	8.70	14.00
Cantera	2.28250000	-76.74255550	Rosas	Vereda las Trincheras	9.00	20.00
Cantera	2.22169440	-76.69888890	Rosas	Vereda Ufugu	7.00	15.00
Rio	1.92499690	-77.09344000	Bolívar	Vereda Capellanía	45.00	103.00
Cantera	1.91622220	-76.99944440	Bolívar	Vereda la medina	25.90	48.00
Cantera	2.66195500	-76.60814390	Piendamó	Vereda Once de Noviembre	23.30	49.00
Cantera	2.43552780	-76.44927780	Popayán	Vereda Quintana	20.70	60.00
Cantera	2.56353800	-76.76230800	Popayán	Vereda el Agrado	40.00	85.00
Cantera	2.84365000	-76.67206940	Morales	Vereda Santa la Cuchilla	12.70	33.00
Cantera	2.17367380	-76.67824280	La Sierra	San Pedro Bajo	11.50	33.00
Cantera	2.19340690	-76.63385890	Sotará	Vereda Hierbas buenas	25.20	71.00
Cantera	2.17058330	-76.60983330	Sotará	Vereda San Roque	21.10	59.00
Cantera	2.76775000	-76.45556940	Caldono	Vereda la Esmeralda	10.00	30.00
Cantera	2.85075000	-76.48597220	Caldono	Cantera Vereda El Jardín	10.37	23.00
Rio	3.25638890	-76.46088880	Villa Rica	Bocas del Palo (Vereda Cantarito)	12.70	30.00
Rio	3.24444720	-76.46280550	Villa Rica	Santa Barbara Vereda Cantarito	12.10	20.00
Rio	3.11377770	-76.38588800	Guachené	Vereda Pilamo	5.00	10.00
Cantera	2.13152770	-76.77997220	La Vega	Corregimiento San Miguel, Vereda Costa Rica	27.10	71.00
Cantera	2.15499970	-76.87439480	La Vega	Corregimiento Los Uvos, Vereda la Calista	45.40	180.00

Fuente: Del autor

Figura 48: Mapa de fuentes de extracción en el software Canteras-Móvil, para construcción de Grafo.

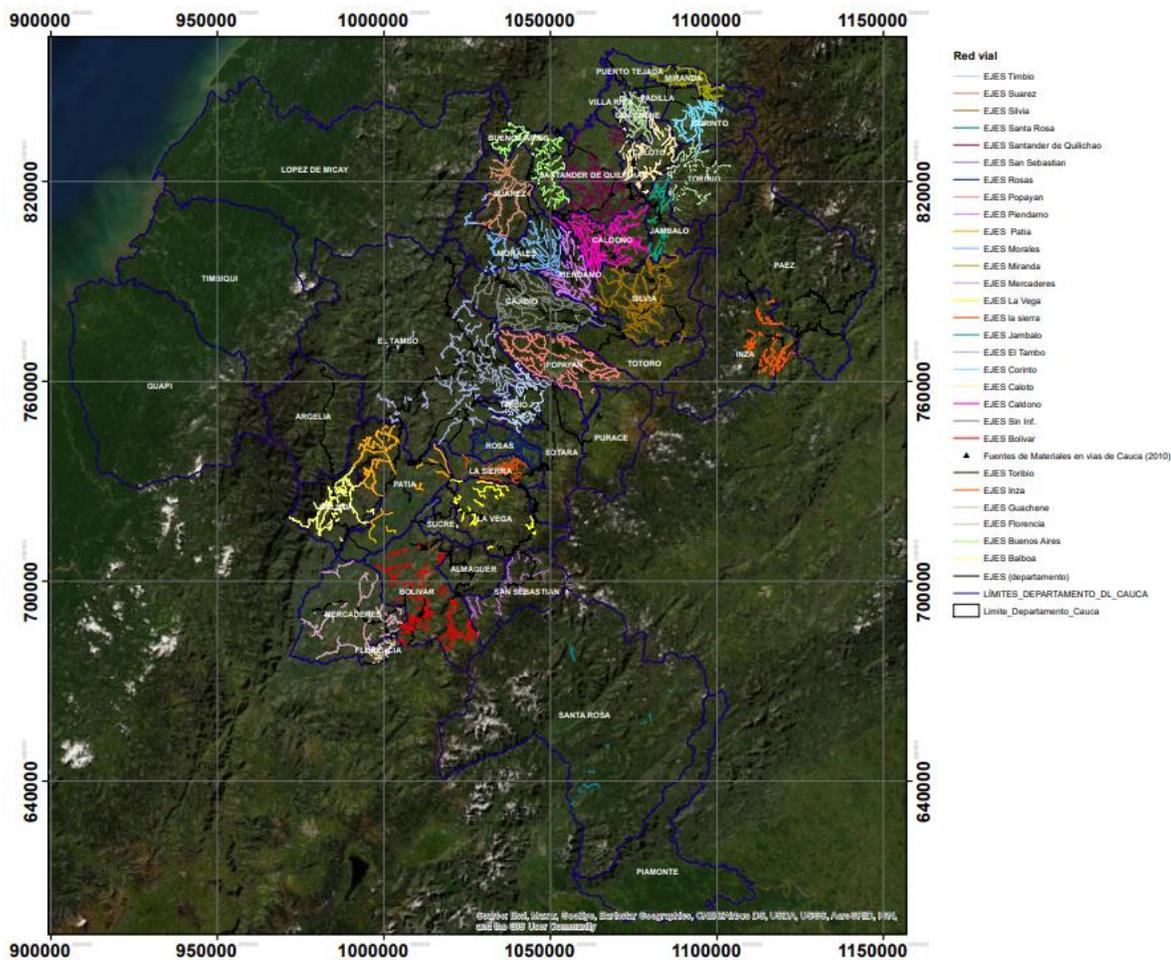


Fuente: Por el Autor

3.2.2. Enlaces

En cuanto a los enlaces, Como primera instancia se ha recurrido a la base de datos inicial suministrada por el CDT Vial, la cual contiene información de accesos viales a las fuentes de extracción, estos enlaces son vías rurales. De acuerdo con la información de la base de datos inicial, se ha extraído información preliminar de distancias, rutas, conexiones y demás información importante para tener en cuenta a la hora de comprobar la información de campo realizada por el CDT Vial a las diferentes fuentes de extracción que serán analizadas en laboratorio y se han capturado mediante el software Canteras-Móvil.

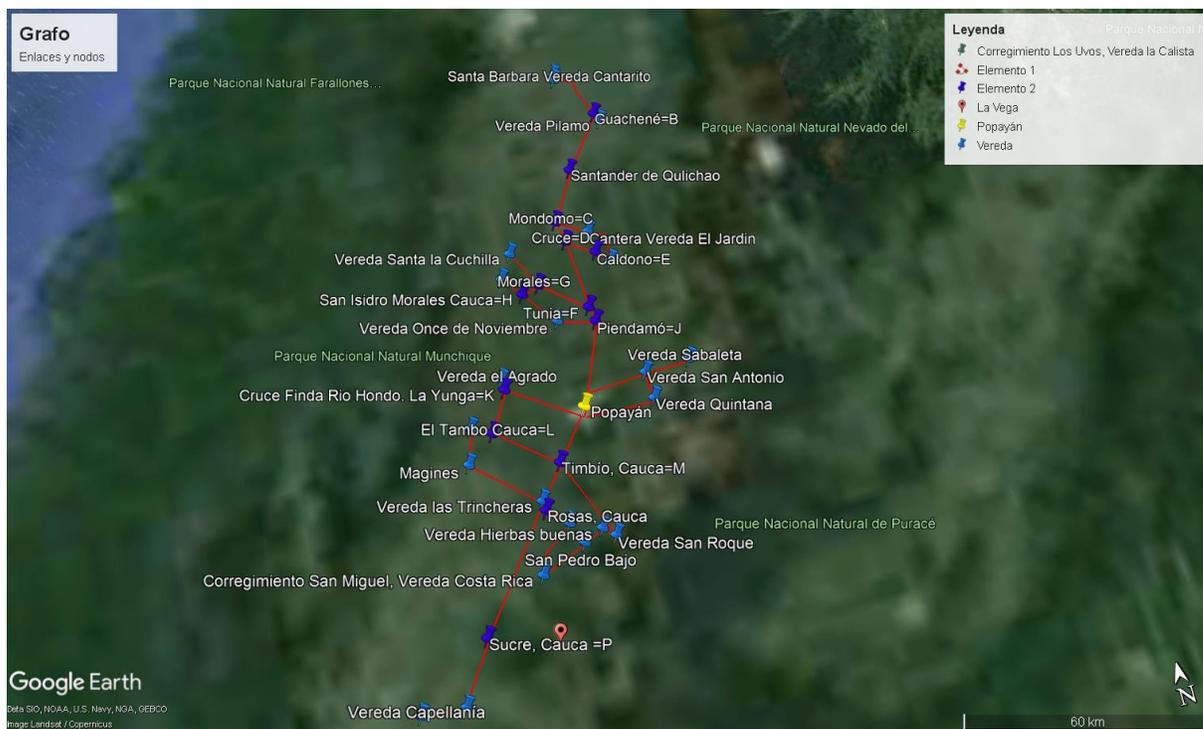
Figura 49: Ejes viales de vías rurales en el Departamento del Cauca.



Fuente: Realizado por el Autor

De acuerdo con la información de enlaces a las fuentes de extracción y además la información capturada por el CDT Vial de la universidad del Cauca a través del software Canteras-Móvil, se ha llegado al siguiente grafo, con el cuál se podrá trabajar el algoritmo de Dijkstra para la ejecución de visitas técnicas.

Figura 50: Grafo para análisis de ruta mínima.



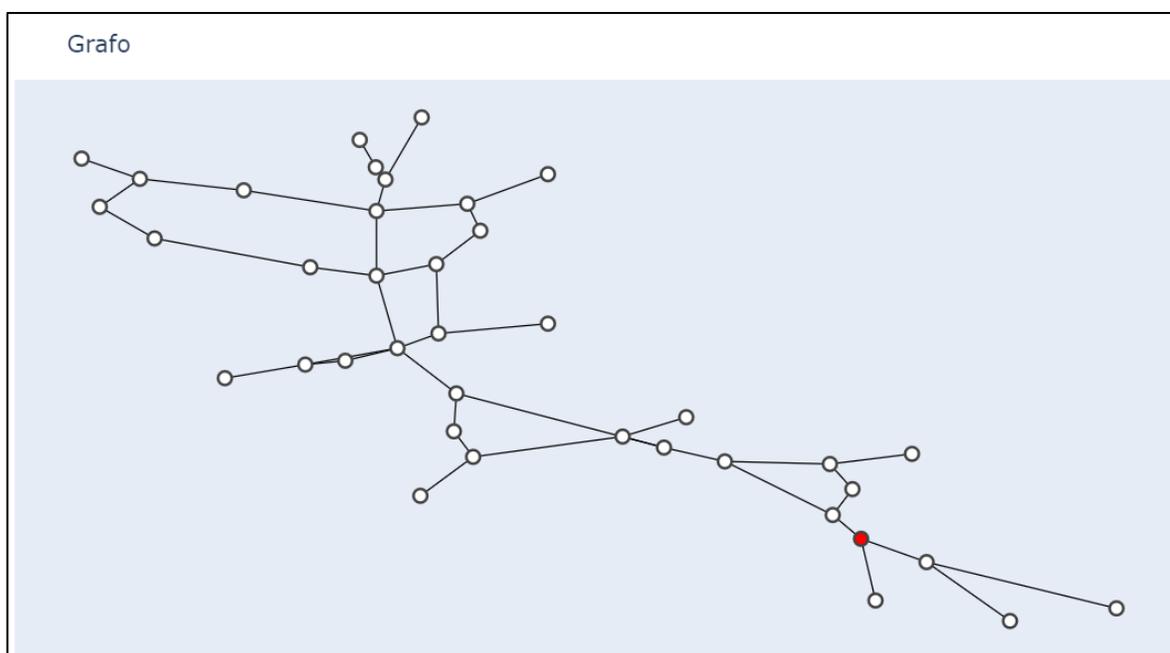
Fuente: Realizado por el Autor

Los nodos en color negro son las fuentes de extracción y los nodos de color azul son puntos de control, los cuales son conexión a diferentes enlaces, estos en realidad son en su mayoría municipios, en la Figura 50 se visualiza cada punto de control con su respectivo municipio o el nombre del sitio.

3.2.3. Resultado

El resultado de aplicar el algoritmo de Dijkstra al grafo que se ha construido previamente, es un esquema donde se visualiza los puntos y las aristas del grafo, sin embargo es importante tener claro los mapas presentados en la Figura 50 y Figura 51, ya que estos determinan de alguna manera una mejor comprensión de la ruta que se deberá seguir cuando se proceda a encontrar la ruta más corta entre un nodo de inicio y un nodo final en el Software. A continuación se visualiza el grafo resultante desarrollado en el software CANTERAS-WEB, en la sección de “Soluciones”.

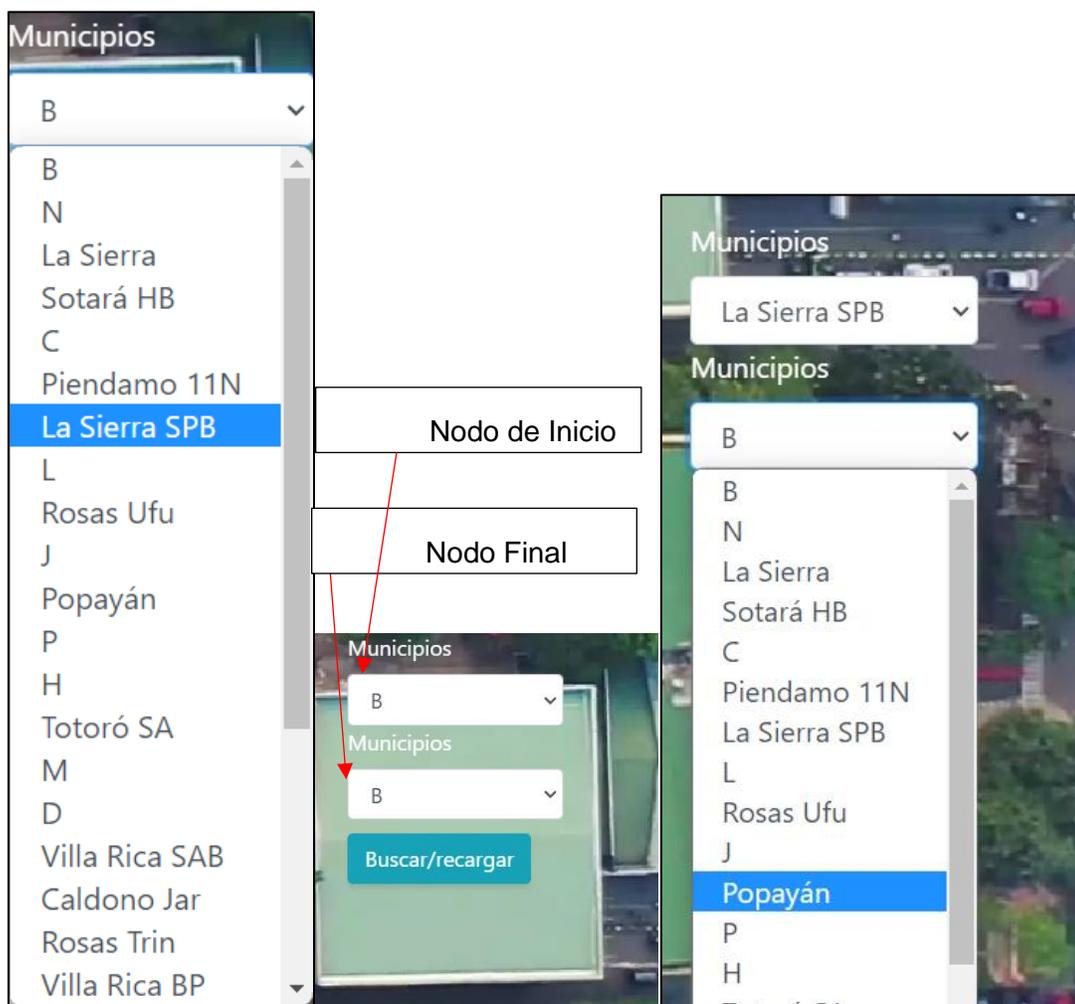
Figura 52: Grafo resultante en el software CANTERAS-WEB.



Fuente: Por el Autor

A continuación se presenta el siguiente ejemplo de ruta mínima entre dos nodos del grafo. Como ejemplo se quiere determinar una visita técnica a las fuentes de la base de datos y determinar la ruta más corta entre los nodos: Nodo de Inicio: Fuente de extracción “La Sierra-San Pedro Bajo”, Nodo Final: “Popayán”.

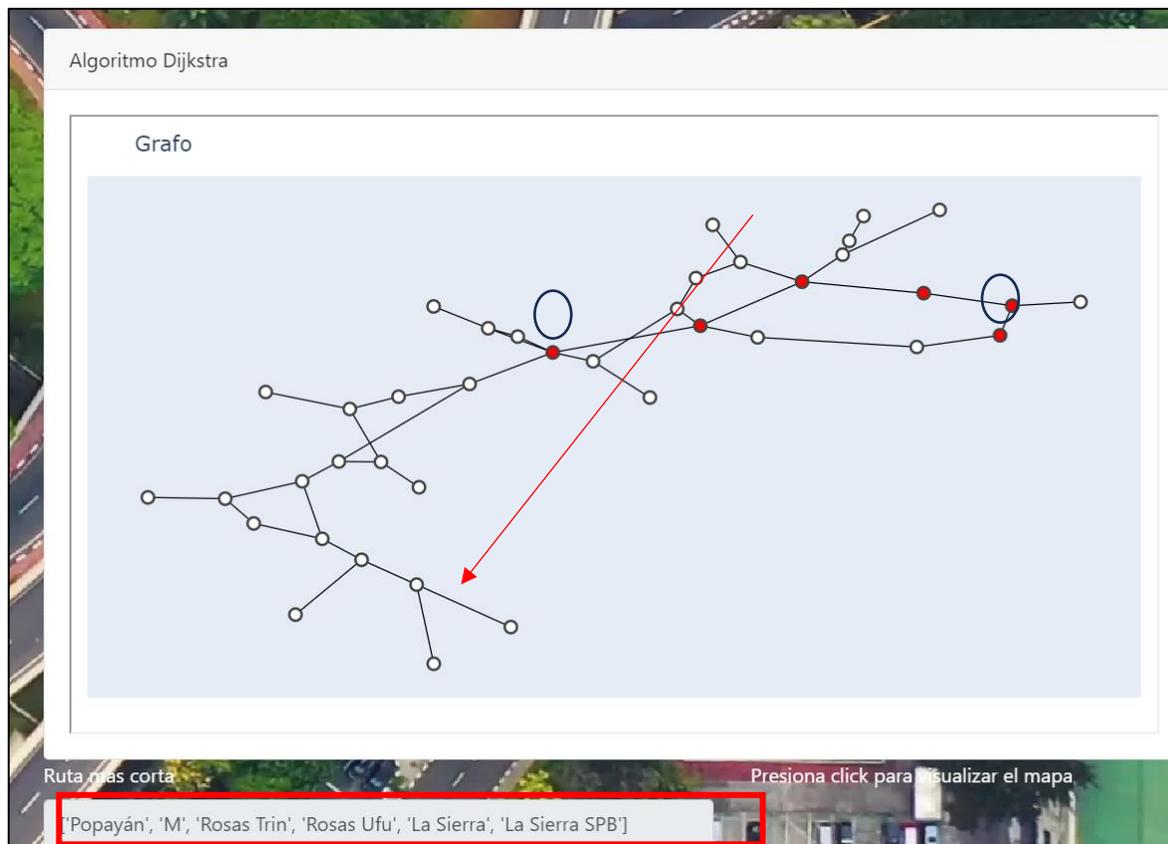
Figura 53: Proceso para encontrar la ruta mínima con el software Canteras-WEB.



Fuente: Por el Autor

El Resultado de la ruta más corta entre los nodos de inicio y fin descritos anteriormente, se muestra en la Figura 54, adicionalmente en la caja de resultado se muestra el orden de la ruta que se deberá seguir de acuerdo con la aplicación del algoritmo de Dijkstra.

Figura 54: Ruta mínima en el Software Canteras-WEB.



Fuente: Por el Autor

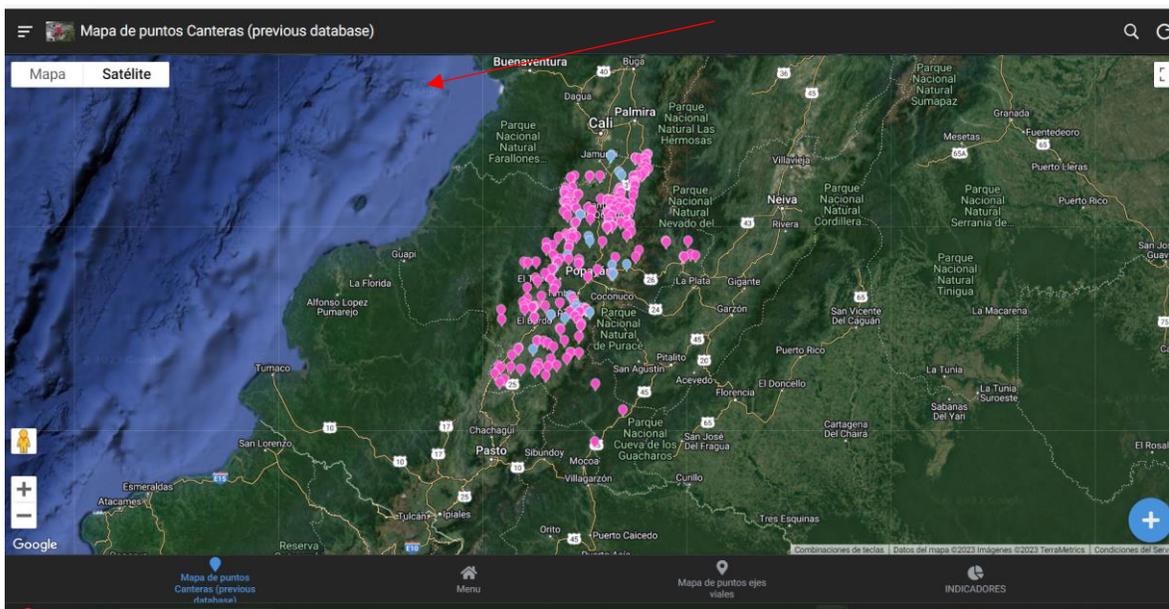
La ruta más corta se muestra en la caja de resultados (Figura 54), es importante aclarar que en la caja de resultados se muestra el orden de visitas determinado por el algoritmo de Dijkstra, tal como se describe a continuación:

1. Fuente de Extracción: La Sierra San Pedro Bajo (Fuente)
2. Municipio de La Sierra Cauca (Punto de Control)
3. Fuente de Extracción: Rosas, Vereda Ufugu (Fuente)
4. Fuente de Extracción: Rosas, Vereda las Trincheras (Fuente)
5. M: Timbío, Cauca (Punto de Control)
6. Municipio de Popayán (Punto de Control).

Esta ruta es muy importante dado que posee información de tiempos de recorrido y distancias, por ende puede ser evaluada también por el segundo software Canteas-Móvil

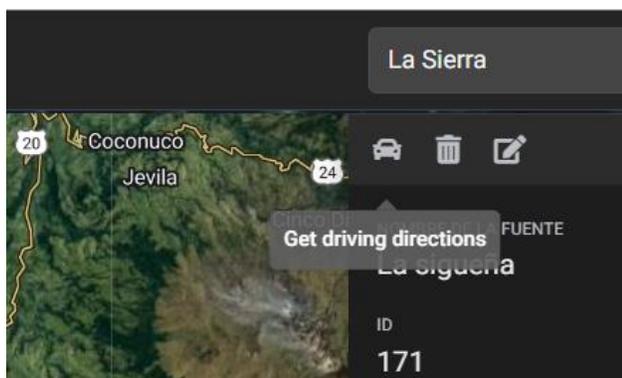
(software para captura de datos en campo) para establecer los tiempos de recorrido entre cada par de nodo de la ruta y comparar los resultados entre lo generado por Canteras-Web y la evaluación entre cada par de nodos por Canteras-Móvil tal cómo se sigue en siguiente análisis:

Figura 55: Fuentes de Extracción en el software Canteras-Móvil.



Fuente: Por el Autor

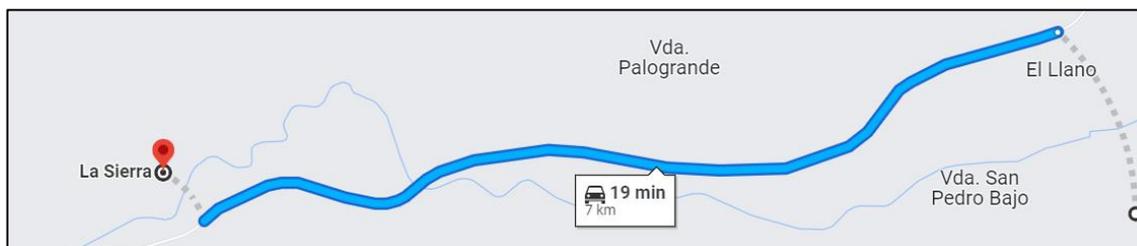
Figura 56: Opción para determinar la ruta más corta entre la fuente y el punto georreferenciado en Canteras-Móvil.



Fuente: Por el Autor

1. Fuente de extracción: La Sierra San Pedro Bajo-Municipio La Sierra, Cauca. (19 minutos)

Figura 57: Ruta mínima generada en Google Maps a través del software Canteras-Móvil



Fuente: Por el Autor

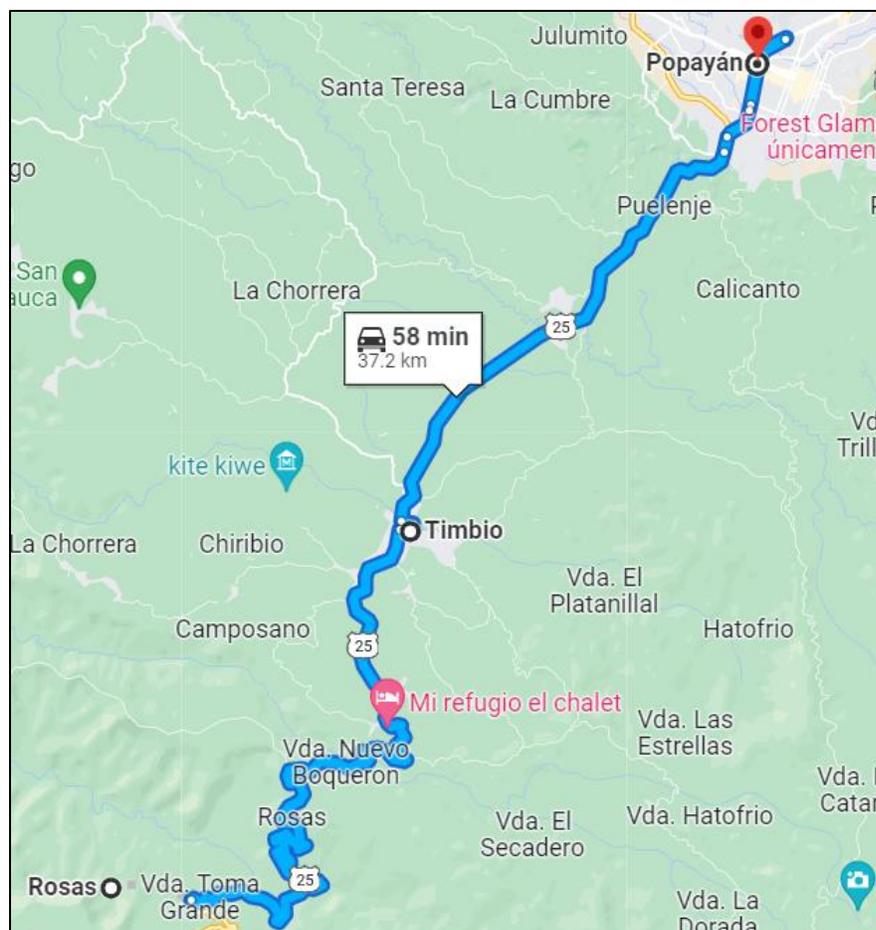
2. Municipio de La Sierra-Fuente de extracción: Rosas, Vereda Ufugu. (47 minutos)



3. Fuente de Extracción: Rosas, Vereda Ufugu- Fuente de Extracción: Rosas, Vereda las Trincheras (20 minutos)



4. Fuente de extracción: Rosas Vereda las Trincheras-Municipio de Timbío-Municipio de Popayán (58 minutos)



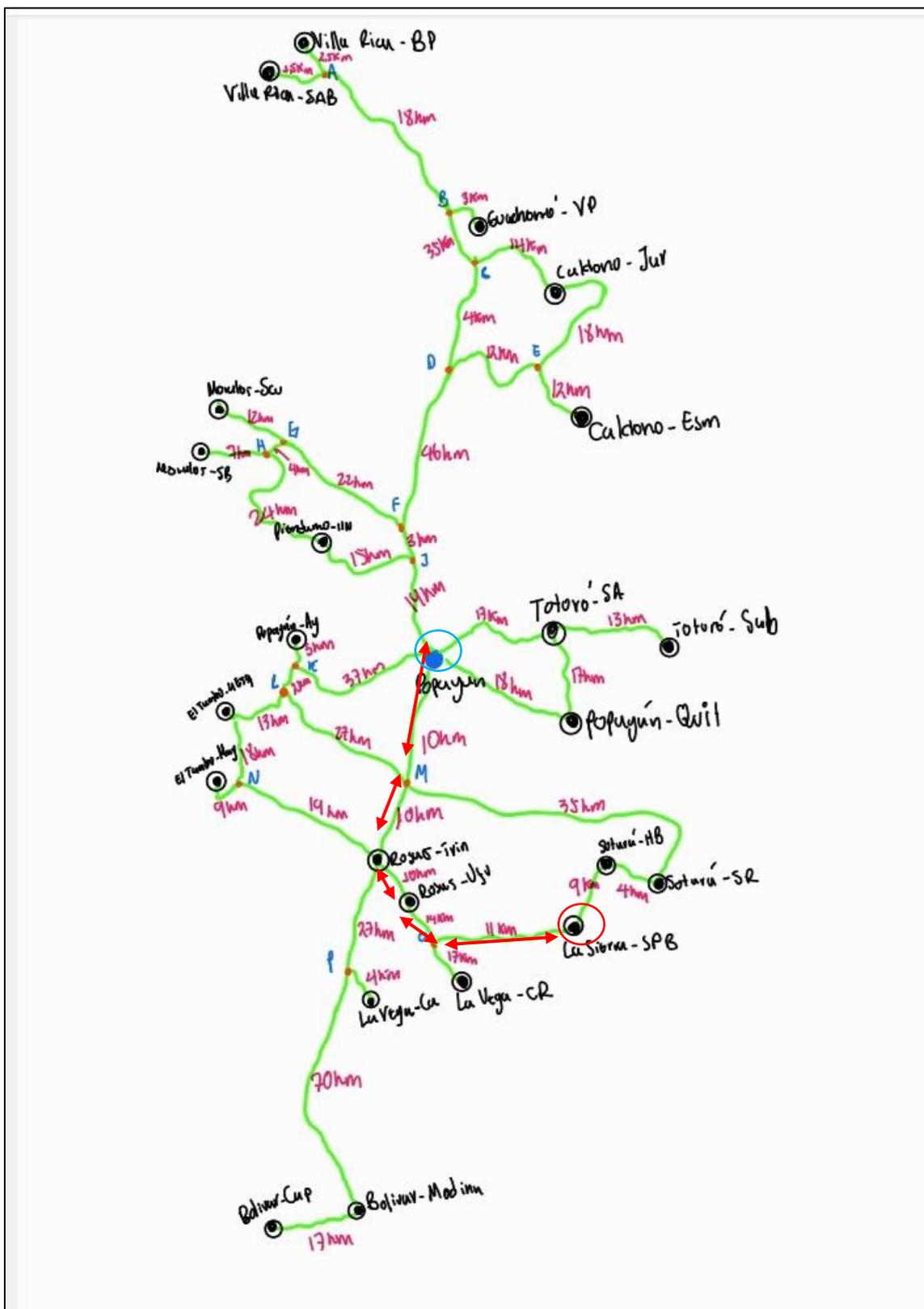
Finalmente se contabiliza los tiempos de recorrido para las visitas a las fuentes de extracción:

$$\textit{Tiempo Total: } 19 + 47 + 20 + 58 = 144 \textit{ minutos}$$

$$\textit{Tiempo Total} = 144 \textit{ minutos} * \frac{1 \textit{ hora}}{60 \textit{ minutos}} = 2.4 \textit{ horas}$$

Adicionalmente se puede visualizar el mapa de la Figura 58 para determinar la distancia de toda la ruta y mediante una velocidad constante se puede lograr obtener un tiempo aproximado de viaje.

Figura 58: Recorrido de la ruta mínima entre los nodos de Inicio y Fin generado por Canteras-Web.



Fuente: Por el Autor

Distancia total de la ruta:

$$\text{Distancia: } 11\text{Km} + 14\text{Km} + 10\text{Km} + 10\text{Km} + 10\text{Km} = 55\text{Km}$$

Suponiendo una velocidad constante de 15Km/h en vías rurales

$$\text{Tiempo} = \frac{55\text{Km}}{15\text{Km/h}} \cong 4 \text{ horas}$$

Comparación entre softwares: Canteras-Móvil y Canteras-WEB

Tabla 4: Resultados sobre Canteras-Móvil Vs Canteras-Web.

Software: Canteras-Móvil	2.4 horas	30 km/h
Software: Canteras-WEB	4 horas	15 km/h

Fuente: Por el Autor

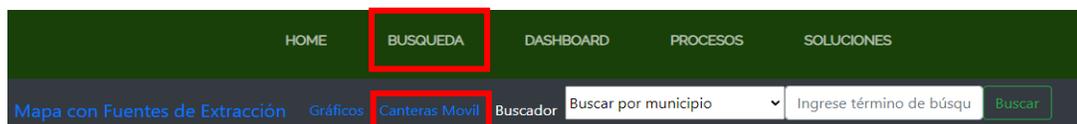
Gráficamente el grafo con sus accesos y la información de distancias se visualiza en la Figura 60.

3.3. RUTA MÍNIMA PARA ACARREO DE MATERIALES

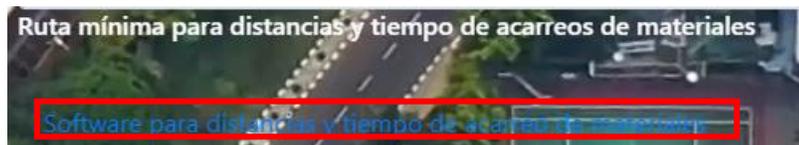
Otra utilidad del software Canteras-Móvil aparte de la captura de datos en campo, es la de encontrar la ruta más corta entre dos puntos: El nodo de ubicación espacial en el momento de análisis (Sitio de Intervención con la infraestructura vial) y el nodo final el cuál es una fuente de extracción, este análisis permite determinar los tiempos de acarreo desde un punto de interés cualquier y un nodo final el cual tiene que ser una fuente de extracción. A partir de esta información también se puede lograr determinar una ruta completa con varias fuentes de extracción, sin embargo demanda mucho más tiempo el proceso de cómputo, ya que, se tendría que manipular el nodo de inicio y nodo final en cada momento entre un par de fuentes de extracción (Ver anexo: Ejemplo de Ruta mínima entre Canteras-Móvil Vs Canteras-Web).

Para ingresar al software Canteras-Móvil desde Canteras – Web se puede realizar de 2 maneras.

1. Desde la opción “Búsqueda-Canteras-Móvil”.

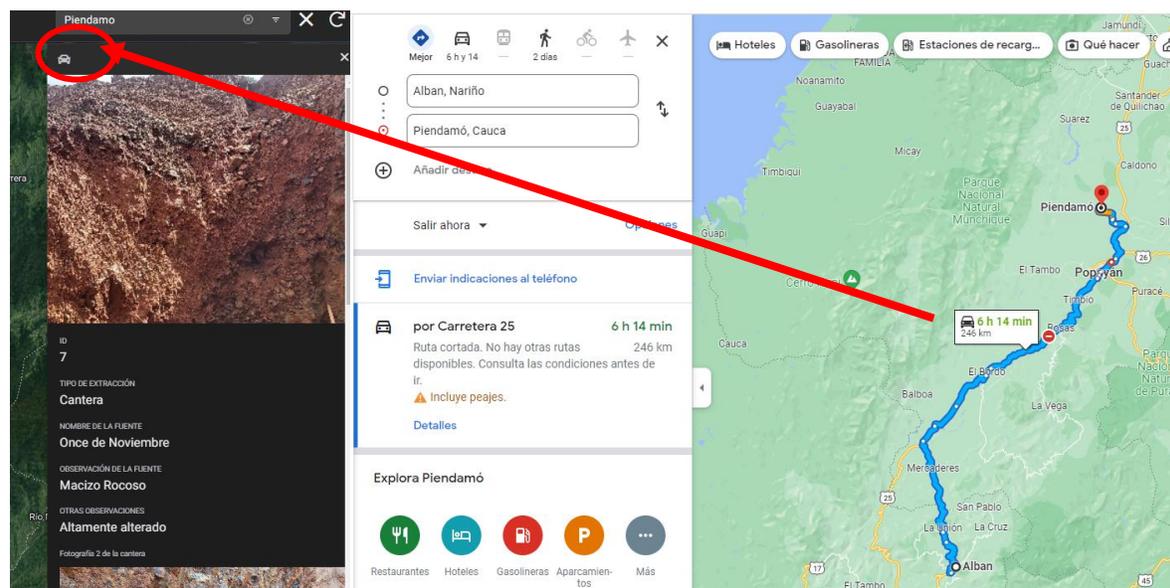


- Desde la opción “Soluciones-Software para distancias y tiempo de acarreo de materiales



Una vez dentro del software Canteras-Móvil se puede realizar el proceso de ruta mínima para establecer los tiempos y distancias de recorrido. En el siguiente ejemplo se muestra un recorrido desde la ubicación geoespacial actual hasta una fuente de extracción cualquiera. Nodo de Inicio: San José de Alban Nariño y Nodo final Fuente de Extracción: Piendamó-Once de Noviembre

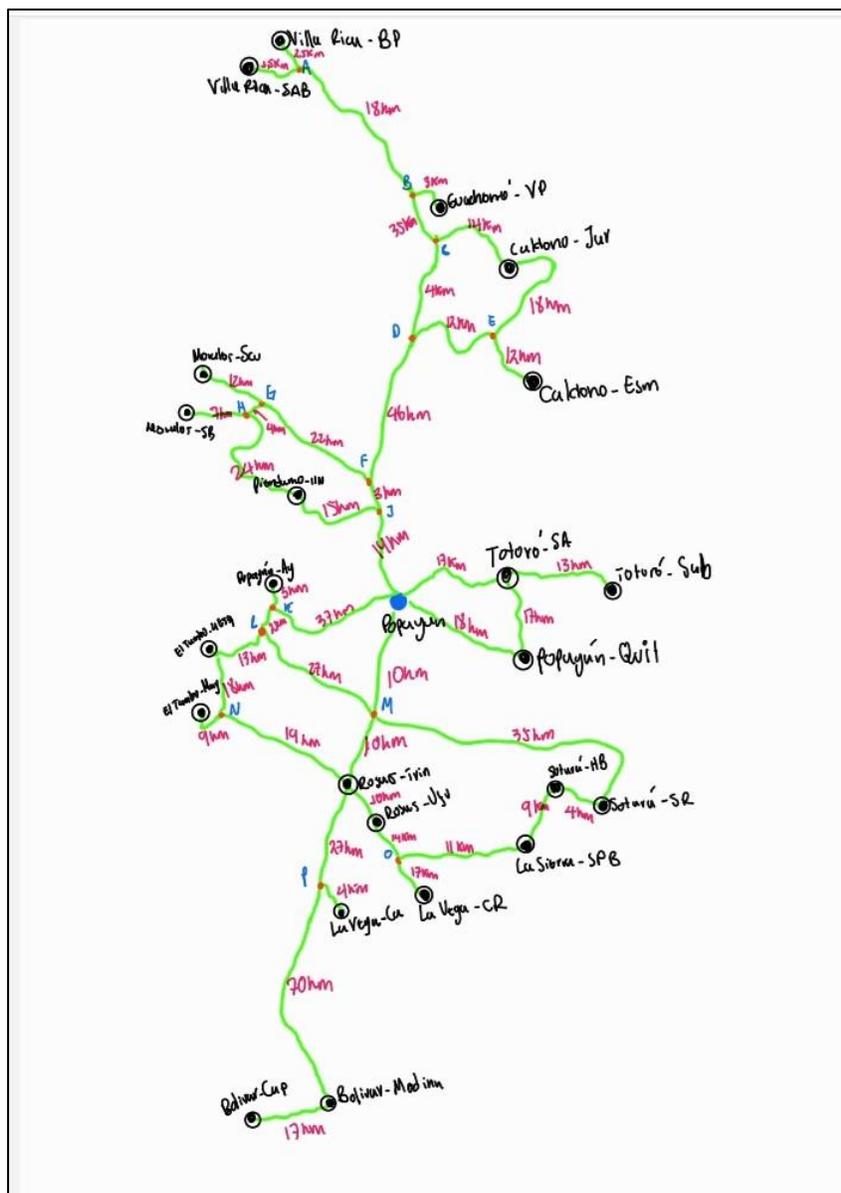
Figura 59: Ruta mínima para Acarreo de materiales generado en Canteras-Móvil.



Fuente: Por el Autor

De acuerdo con lo generado por la base de datos de Google Maps, el tiempo de recorrido y la distancia hasta la fuente de extracción: Piendamó-Once de Noviembre desde el nodo de inicio: San José de Albán Nariño es de: 6 horas y 14 minutos con 24km. Es importante mencionar que este es un solo ejemplo para aclarar la funcionalidad del software.

Figura 60: Esquema de Grafo para el análisis de visitas técnicas a las Fuentes de Extracción.



Fuente: Por el Autor

Teniendo el grafo construido se puede aplicar el algoritmo de Dijkstra el cuál determina la ruta más corta entre dos nodos utilizando la teoría de grafos. En este caso se aplicó la librería NetworkX en Python dado que puede utilizar una gran cantidad de nodos y enlaces sin tener problemas de rendimiento en el software-web. La librería NetworkX es de uso libre y de código abierto. Se distribuye bajo la licencia BSD, lo que significa que puede ser utilizada, modificada y distribuida de forma gratuita, tanto para proyectos personales como comerciales.

En ese sentido si se desea ir desde un nodo de inicio, un ejemplo la fuente de extracción el “Tambo-Magines”, hacía la fuente de extracción “Esmeralda-Caldono”, el resultado de aplicar el algoritmo de Dijkstra al Grafo previamente construido, será una ruta mínima que al menos pase por una fuente de extracción y al menos un punto de control, que en este caso los puntos de control son en su mayoría cabeceras municipales, la información de estos puntos de control se encuentran en el mapa de ayuda incluido en el software Canteras-Web (Ver ítem: Enlaces del Grafo. Ejemplo de ruta mínima con Canteras-Web).

3.3.1. Desarrollo del algoritmo de ruta mínima

Para el desarrollo de ruta mínima con la base de datos descrita en los anteriores ítem, se ha utilizado NetworkX de Python, una librería de uso libre, de código abierto. Se distribuye bajo la licencia BSD, lo que significa que se puede utilizarla, modificarla y distribuirla de forma gratuita. No es necesario pagar ninguna tarifa para utilizar NetworkX. Es importante revisar los detalles de la licencia y asegurarse de cumplir con sus términos al utilizar la biblioteca. Para ser uso de la librería es importante tener claro los conceptos de sus funciones, objetos, clases y en generar la programación orientada a objetos para hacer uso correctamente de las librerías a utilizar, además se tiene que seguir la documentación de la librería para lograr el correcto funcionamiento.

Cómo primer paso se construye el grafo con los nodos y enlaces para lograr aplicar el algoritmo de Dijkstra. La base de datos de nodos es la información capturada por el CDT Vial de la Universidad del Cauca, esta posee dos columnas importantes las cuales son el tiempo y la distancia desde la fuente de extracción a la cabecera municipal que se han registrado en el transcurso de la captura de datos en campo, sin embargo mediante la búsqueda de información con la base de datos preliminar se ha encontrado más enlaces con sus distancias que conectan a una determinada fuente de extracción, a partir de la recolección de esta información y con la que se tiene en el software Canteras-Móvil el grafo resultante es el que se muestra en la Figura 61:

Para construir el grafo en el software Canteras-Web se ha hecho uso de una librería que permita dibujar el grafo con los nodos y enlaces, esta librería al igual que la librería NetworkX de Python es de código abierto, la librería para el dibujo del grafo es Plotly. En la siguiente sección se muestra parte del código utilizado para el dibujo en pantalla del grafo y la construcción de este. Este código completo se muestra en ítem de Anexos:

Figura 62: Parte de código de la construcción del grafo y obtener la ruta mínima para tiempos de visita en Canteras-Móvil

```
#!/*****GRAFOS*****/
#función para el grafo y el algoritmo de disjktra
def generar_grafica(request):
    # Obtener los datos del modelo "Grafo"
    datos = grafo.objects.all()

    # Crear el grafo vacío
    G = nx.Graph()

    # Agregar los nodos y las aristas al grafo
    for dato in datos:
        G.add_edge(dato.NODO, dato.CONEXION, weight=dato.DISTANCIA)

    #ruta mínima
    ruta_mas_corta = ''
    if request.method == 'POST':
        origen = request.POST.get('origen')
        destino = request.POST.get('destino')
        # Calcular la ruta más corta utilizando el algoritmo de Dijkstra
        ruta_mas_corta = nx.dijkstra_path(G, origen, destino)
```

Fuente: Por el Autor

Figura 63: Código en Python para mostrar el grafo.

```
# Generar la visualización del grafo utilizando Plotly
pos = nx.spring_layout(G)

edge_x = []
edge_y = []
for edge in G.edges():
    x0, y0 = pos[edge[0]]
    x1, y1 = pos[edge[1]]
    edge_x.extend([x0, x1, None])
    edge_y.extend([y0, y1, None])

edge_trace = go.Scatter(
    x=edge_x, y=edge_y,
    line=dict(width=1, color='black'),
    hoverinfo='none',
    mode='lines')

node_x = []
node_y = []
for node in G.nodes():
    x, y = pos[node]
    node_x.append(x)
    node_y.append(y)
```

Fuente: Por el Autor

3.4. Desarrollo de las Redes Neuronales Artificiales

REDES NEURONALES ARITIFICIALES

Posterior a la construcción del grafo, es importante la construcción de las redes neuronales con la base de datos que se ha descrito anteriormente, la base de datos preliminar que ha servido para la construcción del grafo será la misma para el entrenamiento de las redes neuronales con la finalidad de cumplir con el objetivo de la presente investigación

Para la construcción del modelo de red neuronal artificial, entrenar el modelo y generar predicciones, de manera preliminar se ha hecho uso de la misma base de datos que se utilizó en la construcción del grafo, tal como se evidencia a continuación:

Figura 64: Base de datos para el entrenamiento de las redes neuronales.

Query Query History Scratch Pad x

```

1 SELECT * FROM public."Dashboard_dashboard"
2 ORDER BY id ASC

```

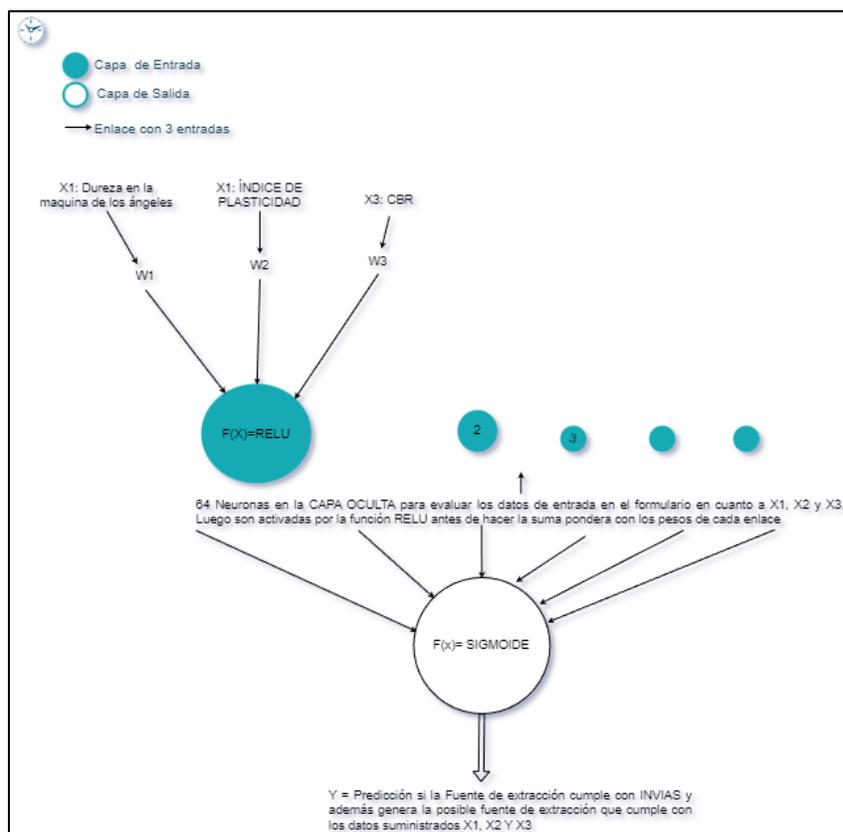
Data Output Messages Notifications

	id [PK] bigint	tipoextraccion character varying (150)	nombre_fuente text	observacion_fuente text	otras_observaciones text	codigo text	fecha date	co do
1	1	Cantera	Magines	Roca Muerta	Color amarillo	nan	2022-07-22	
2	2	Cantera	Chisquio	Roca Muerta	Color amarillo	nan	2022-07-22	
3	3	Cantera	Santa Bárbara	Roca meteorizada	Grado de meteorización II a III	nan	2022-08-18	
4	4	Cantera	Zabaletas	Roca	Segunda punto de toma , Grado de meteorización III a ...	nan	2022-09-02	
5	5	Cantera	San Antonio	Material altamente meteorizado	Dearman tipo V a VI	nan	2022-09-02	
6	6	Cantera	Las Trincheras	Roca	Roca Meteorizada	nan	2022-07-29	
7	7	Cantera	Ufugu	Roca	Roca Meteorizada	nan	2022-07-29	
8	8	Rio	Capellania	Material de Rio	aparentemente bien gradado	nan	2022-08-05	
9	9	Cantera	El colerín	Macizo Rocoso	Roca parcialmente meteorizada	nan	2022-08-05	
10	10	Cantera	Once de Noviembre	Macizo Rocoso	Altamente alterado	nan	2022-08-10	
11	11	Cantera	Belencito	Agregados Petreos	Material para mejoramiento de vías	nan	2022-08-11	
12	12	Cantera	Los Cerrillos	Macizo Rocoso	visible de meteorización	nan	2022-08-11	
13	13	Cantera	La Cuchilla	Agregados Petreos	Meteorización IV a VI	nan	2022-08-18	
14	14	Cantera	San Pedro Bajo	Macizo Rocoso	Roca dura y resistente ante la excavación	nan	2022-08-24	
15	15	Cantera	Hierbas buenas	Agregados Petreos	visible de meteorización	nan	2022-08-30	
16	16	Cantera	San Roque	Macizo Rocoso	Roca dura y resistente ante la excavación	nan	2022-08-30	
17	17	Cantera	Vereda La Esmeralda	Macizo Rocoso	Grado de meteorización III a IV	nan	2023-01-27	

Fuente: Realizado por el Autor

Continuando con la construcción de la estructura de red neuronal se ha creado el siguiente modelo de red Neuronal:

Figura 65: Creación del modelo de red Neuronal. Capa de entrada con 3 características, una capa oculta con 64 neuronas y una capa de salida.



Fuente: Realizado por el Autor

Como se evidencia en la Figura 65 la estructura de red neuronal está compuesta por una entrada con 3 enlaces las cuales son las 3 características que se envían desde el formulario del software Canteras-Web en el apartado de “Soluciones” estos tres enlaces asociados a las características (**X1:Dureza**=Ensayo en la Maquina de los Ángeles, **X2:IP**=Índice de Plasticidad, **X3:CBR**=Prueba de penetración para comprobar las características mecánicas del suelo) tienen sus tres pesos asociados $w1$, $w2$ y $w3$ respectivamente y son transmitidos a cada una de las 64 neuronas. El procedimiento que se realiza en cada neurona de las 64 existentes en la capa oculta de la estructura de red neuronal se ejecuta siguiendo el siguiente procedimiento.

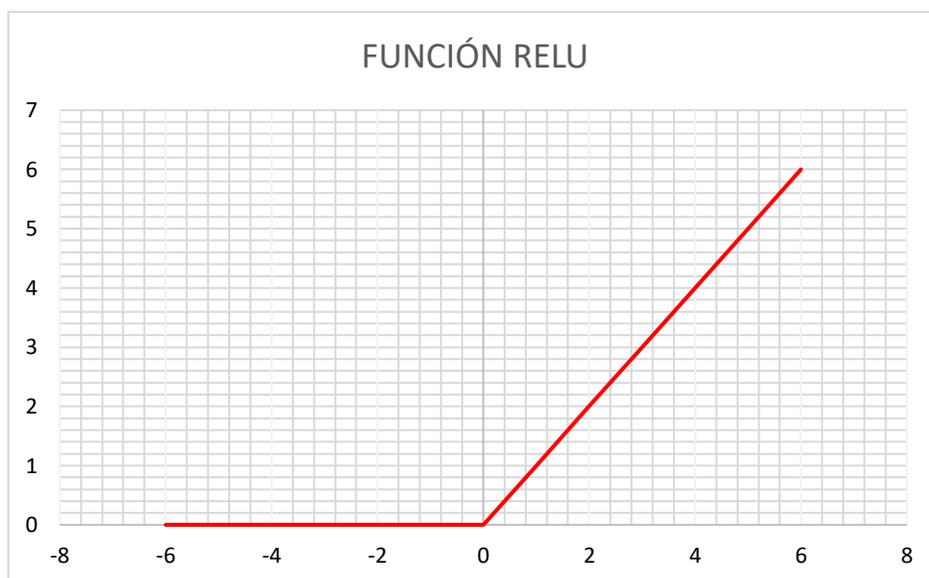
1. A cada neurona de la capa oculta se le realiza la suma ponderada de $w1,x1$; $w2,x2$; $w3,x3$ tal como siguen:

$$\left(\sum_{i=1}^3 (w_i * x_i) \right) + b = (x_1 * w_1 + x_2 * w_2 + x_3 * w_3) + b$$

Ecuación 22: Suma ponderada del modelo de red neuronal. Fuente: Por el Autor

2. Seguidamente se activa cada neurona de la capa oculta utilizando la función de activación RELU. Como se ha mencionado esta función comprende valores:

$$\max(0, x)$$



En ese sentido la derivada de esta función es muy sencilla y por tanto un costo computacional muy bajo.

$$\frac{d}{dx}(f(RELU)) = \begin{cases} 0, & \text{para } X < 0 \\ 1, & \text{Para } X > 0 \end{cases}$$

Ecuación 23: Derivada de la función de activación utilizada. Fuente: Adaptada por el Autor

3. RELU de acuerdo con (Tech, 2022) es mucho más efectiva que las funciones sigmoide y Tangente Hiperbólica en las capas ocultas.
4. Seguidamente de activar las 64 neuronas con la función RELU, se vuelve a realizar la suma ponderada de la llegada de pesos de las 64 neuronas de la capa oculta a la única neurona de salida y se activa la neurona de salida aplicando la función LOGÍSTICA, la cuál es la indicada para una capa de salida en problemas de clasificación binaria (Tech, 2022).

$$\left(\sum_{i=1}^{n=64} (w_i * x_i) \right) + b = (x_1 * w_1 + x_2 * w_2 + x_3 * w_3 + \dots + x_{64} * w_{64}) + b$$

Ecuación 24: Suma ponderada después de salir de la capa oculta y llegar a la capa de salida.

Fuente: Por el Autor

$$f_{Logistica} \left(\left(\sum_{i=1}^{n=64} (w_i * x_i) \right) + b \right)$$

Ecuación 25: Función logística para activar la capa de salida. Fuente: Por el Autor

5. Finalmente se ha realizado el entrenamiento del modelo para realizar predicciones con valores de entrada del formulario x_1 , x_2 y x_3 . Para el entrenamiento se utilizó la función de costo: Binary Crossentropy

$$Binary\ Crossentropy = -\frac{1}{N} \sum_{i=1}^N [y_i * \log(P_i) + (1 - Y_i) * \log(1 - P_i)]$$

Ecuación 26: Función de Perdida o de Costo utilizada para el entrenamiento del modelo. Fuente:

Adaptada por el Autor

6. Se ha utilizado la función Binary Crossentropy debido a que cómo se ha estudiado, esta función es muy utilizada en problemas de clasificación binaria, donde se busca asignar una de dos clases posibles, también es conocida como “log los”. Esta función mide la discrepancia entre las probabilidades predichas y las etiquetas reales.

Antes de describir el proceso de entrenamiento, es importante conocer cómo se realiza la ejecución del entrenamiento:

Una vez creada la red neuronal le pasamos los ejemplos para que comience el aprendizaje automático, es decir le damos unos datos de entrada y sus respectivas salidas. Seguidamente se realiza las predicciones haciendo uso de la función de costo. Posterior a ello se evalúa el error de la función de costo para conocer ca calidad de la predicción. Haciendo uso del algoritmo de Backpropagation se realiza la “retro propagación de error” en dirección hacia atrás es decir evaluado capa por capa encontrando las derivadas parciales de la función de costo respecto a los parámetros de la red en este caso los pesos y sesgos, cómo se tiene una sola capa oculta es menor el tiempo de cómputo para realizar la evaluación de la “retro propagación de error”. Al momento de encontrar las derivadas parciales de la función de costo respecto a los parámetros de la red se genera un vector de

gradiente, este vector gradiente es utilizado por el método del “Descenso del gradiente” para encontrar mínimos locales o globales y poder optimizar los pesos y sesgos para que se genere los resultados esperados en las predicciones.

7. El proceso de entrenamiento se ha realizado de acuerdo a los parámetros exigidos por el artículo 311-13 de INVIAS. Es decir por cada fuente de extracción se evalúan las 3 entradas del formulario
 - X1: Ensayo en la Máquina de los Ángeles
 - X2: Ensayo de CBR
 - X3: Índice de Plasticidad

Estos tres parámetros son evaluados de acuerdo con lo exigido por INVIAS para afirmados. Posteriormente se realiza una capa oculta con la granulometría de cada fuente de materiales. Es necesario recalcar que las fuentes de extracción de la base de datos no cumplen al 100% con las franjas granulométricas exigidas por INVIAS y por ende como trabajo futuro se desea añadir una herramienta dentro del software Canteras-Web que permita hacer una combinación de una fuente X con una fuente Y, con esto se buscaría mejorar el resultado de la granulometría y que logre ajustarse a alguna franja granulométrica exigida por INVIAS, además realizar esta combinación de fuentes puede generar resultados favorables en cuanto a los ensayos de Índice de Plasticidad, Dureza y CBR, sin embargo para tener certeza de este aspecto se tendría que evaluar varias combinaciones de materiales, un ejemplo sería realizar los ensayos de Dureza, Índice de Plasticidad y CBR para una combinación de una fuente X con una Fuente Y, analizar sus resultados, entrenar esta base de datos con la creación de un modelo predictivo utilizando A.N.N para que en el momento de combinar otro par de Fuentes en el software genere resultados favorable de los parámetros CBR, Índice de Plasticidad, Dureza, entre otros. Recordemos que los parámetros X1, X2 y X3 son los datos de entrada del formulario descritos previamente.

Antes de entrenar al modelo con lo expuesto se ha realizado un análisis sencillo de las fuentes que cumplen y no cumplen con los requisitos exigidos por INVIAS para comprender que tan dispersos son los datos respecto a considerar dos clases en el entrenamiento, es decir:

- Clase 1: Fuentes que cumplen con los tres parámetros X1, X2 y X3
- Clase 0: Fuentes que no cumplen con alguno de los tres parámetros X1, X2 y X3

De acuerdo con lo expuesto por la teoría de redes neuronales, el entrenamiento de una estructura de red neuronal depende de los datos. Los datos a entrenar tienden a estar más presente para una clase, es decir no hay un balance de clases, esto podría generar problemas con la predicción, ya que, cuando se pasen datos de entrada de X1, X2 y X3 y estos estén en el límite del requisito y se desea predecir si estos datos cumplen o no cumplen con INVIAS, la tendencia de la predicción podría estar más a que no lo hacen, así los parámetros de entrada sean correctos, eso debido al desbalance de clases que se ha evidenciado en la base de datos.

La Figura 66 muestra el análisis realizado para la mayoría de las fuentes de extracción que posee los 3 ensayos (X1, X2 y X3) evaluando si cumplen o no. Como se observa la mayoría de las Fuentes de extracción no cumplen con los tres ensayos, se evidencia que todas cumplen con el CBR (X3), pero en su mayoría no poseen los requerimientos de Dureza (X1) o con el índice de plasticidad (X2), en ese sentido encontrar una fuente de extracción que cumplan los tres ensayos es complejo y a la hora de realizar el entrenamiento tiende a estar más presente la “clase 0” donde en su mayoría no cumple. Es por esta la razón que se vuelve importante la creación de una herramienta dentro del software que permita hacer la combinación de un material, hacerle los respectivos ensayos de esta combinación, entrenar el modelo y luego realizar la predicción con estos resultados debido a que si se combina una fuente con otra, existe más posibilidad de que se logre cumplir con lo exigido por INVIAS.

Figura 66: Análisis de requisitos X1, X2 y X3 según Artículo 311-13 INVIAS.

Extracción	Municipio	Zona	Vereda	Nombre Fuente	Observacion	Observación 2	Dure	Requisito Dureza	Índice de plasticidad	Requisito IP	CBR	Requisito CBR
Cantera	El Tambo	Centro	Magines	Magines	Roca Muerta	Color amarillo	52	no cumple	17.7	no cumple	51	cumple
Cantera	El Tambo	Centro	Corregimiento	Chisquico	Roca Muerta	Color amarillo	56	no cumple	10.8	no cumple	26	cumple
Cantera	Morales	Centro	Vereda Santa	Santa Bárbara	Roca meteorizada	Grado de meteorización II a III	32	cumple	0	no cumple	77	cumple
Cantera	Totoro	Oriente	Vereda Sabal	Zabaletas	Roca	Segunda punto de toma , Grado de meteorización III a IV	71	no cumple	8.8	cumple	66	cumple
Cantera	Totoro	Oriente	Vereda San A	San Antonio	Material altamente meteorizado	Dearman tipo V a VI	52	no cumple	10.9	no cumple	52	cumple
Cantera	Rosas	Centro	Vereda las Tri	Las Trincheras	Roca	Roca Meteorizada	60	no cumple	10.5	no cumple	40	cumple
Cantera	Rosas	Centro	Vereda Ufugu	Ufugu	Roca	Roca Meteorizada	38	cumple	11.2	no cumple	50	cumple
Rio	Bolívar	Sur	Vereda Capell	Capellania	Material de Rio	aparentemente bien gradado	21	cumple	0	no cumple	79	cumple
Cantera	Bolívar	Sur	Vereda la mei	El colerin	Macizo Rocoso	Roca parcialmente meteorizada	41	cumple	17.2	no cumple	30	cumple
Cantera	Piendamó	Centro	Vereda Once	Once de Noviembre	Macizo Rocoso	Altamente alterado	44	cumple	20.3	no cumple	54	cumple
Cantera	Popayán	Centro	Vereda Quinti	Belencito	Agregados Petreos	Material para mejoramiento de vías	60	no cumple	11.6	no cumple	33	cumple
Cantera	Popayán	Centro	Vereda el Agra	Los Cerrillos	Macizo Rocoso	visible de meteorización	33	cumple	14.6	no cumple	27	cumple
Cantera	Morales	Centro	Vereda Santa	La Cuchilla	Agregados Petreos	Meteorización IV a VI	37	cumple	11.6	no cumple	59	cumple
Cantera	La Sierra	Centro	San Pedro Baj	San Pedro Bajo	Macizo Rocoso	Roca dura y resistente ante la excavación	60	no cumple	9.3	no cumple	56	cumple
Cantera	Sotará	Centro	Vereda Hierbi	Hierbas buenas	Agregados Petreos	visible de meteorización	63	no cumple	9.2	no cumple	37	cumple
Cantera	Sotará	Centro	Vereda San R	San Roque	Macizo Rocoso	Roca dura y resistente ante la excavación	44	cumple	11.7	no cumple	33	cumple
Cantera	Caldono	Norte	Vereda la Espr	Vereda La Esmeralda	Macizo Rocoso	Grado de meteorización III a IV	69	no cumple	14.4	no cumple	37	cumple
Cantera	Caldono	Norte	Cantera Vered	Cantera Verena El Jardín	Macizo Rocoso	Grado de meteorización II a III	65	no cumple	25	no cumple	33	cumple

Fuente: Por el Autor

Entrenamiento de las Redes Neuronales Artificiales

Para el entrenamiento de la base de datos se ha hecho uso de datos adicionales para compensar la falta de datos que cumplen con los tres requisitos de entrada X1, X2 y X3. La Base de datos establecida es la siguiente:

Figura 67: Base de datos final para el entrenamiento del modelo de red neuronal.

	Nombre_Fuente text	Observacion text	Observacion_2 text	Dureza double precision	IP double precision	CBR double precision
17	Vereda La Esmeralda	Macizo Rocoso	Grado de meteorización III a IV	69	14.4	37
18	Cantera Verena El Jardín	Macizo Rocoso	Grado de meteorización II a III	65	25	33.3
19	to) Bocas del Palo	Balastro , Material de Rio	Material para mejoramiento de vías , Arena de Rio	42	0	21.9
20	Santa Barbara	Balastro , Material de Rio	Material para mejoramiento de vías , Arena de Rio	41	0	21.9
21	Vereda Pilamo	Balastro , Material de Rio	Arena de Rio , Material utilizado para relleno	24	0	43.8
22	da Costa Rica	Costa rica	Roca no meteorizada	63	15.9	65.3
23	la Calista	Los uvos	Depósitos Aluviales	63	20.9	56.4
24		Entrenamiento 1	Entrenamiento 1	16	4.1	16
25		Entrenamiento 2	Entrenamiento 2	17	4.5	18
26		Entrenamiento 3	Entrenamiento 3	20	4.3	22
27		Entrenamiento 4	Entrenamiento 4	25	5.2	25
28		Entrenamiento 5	Entrenamiento 5	18	5.3	26
29		Entrenamiento 6	Entrenamiento 6	19	5.5	28
30		Entrenamiento 7	Entrenamiento 7	26	5.8	31
31		Entrenamiento 8	Entrenamiento 8	30	5.9	32
32		Entrenamiento 9	Entrenamiento 9	31	6	35
33		Entrenamiento 10	Entrenamiento 10	35	6.2	38

Total rows: 69 of 69 Query complete 00:00:00.401 Ln 1, Col 1

Fuente: Por el Autor

Para realizar el entrenamiento del modelo es importante conocer parte del código estructurado con DJANGO. El entrenamiento se ha hecho con la base de datos (Figura 67), la cual contiene información de todas las fuentes de extracción capturadas por el CDT Vial de la Universidad del Cauca, esta posee ensayos de las fuentes capturadas y adicionalmente se han añadido datos ficticios que debería cumplir una fuente de extracción para que a la hora de entrenar el modelo no haya un desbalance de clases como se ha mencionado previamente. En la estructuración de la red neuronal se ha hecho uso de tres entradas o tres enlaces que llegan a cada neurona de las 64 neuronas de la capa oculta, estas pasan por la función de activación RELU. Seguidamente se vuelven a pasar estos resultados a una segunda capa, la cuál es la capa de salida, esta posee una sola neurona y como se menciona todos los resultados de las salidas de todas las neuronas de la capa oculta llegan a la única neurona de la capa de salida y vuelve a realizarse el proceso de activación utilizando la función LÓGISTICA. Este proceso se repite iterativamente para que la red neuronal aprenda mediante el algoritmo de Backpropagation y el método del descenso del gradiente optimice o ajuste los pesos y sesgos de cada capa hacia atrás, en este caso se devuelve desde la capa de salida, luego pasa por la capa oculta analizando los errores y finalmente llega a la capa de entrada para ajustar completamente los parámetros de la red.

3.4.1.1. Estructuración del entrenamiento

La estructuración del entrenamiento se ha hecho siguiendo el siguiente esquema:

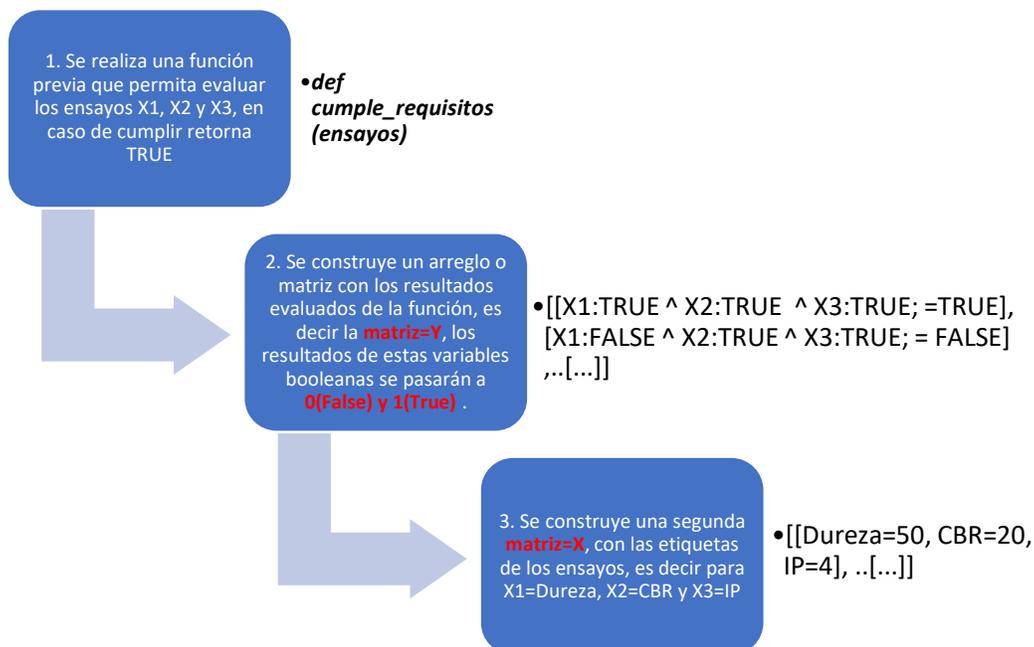
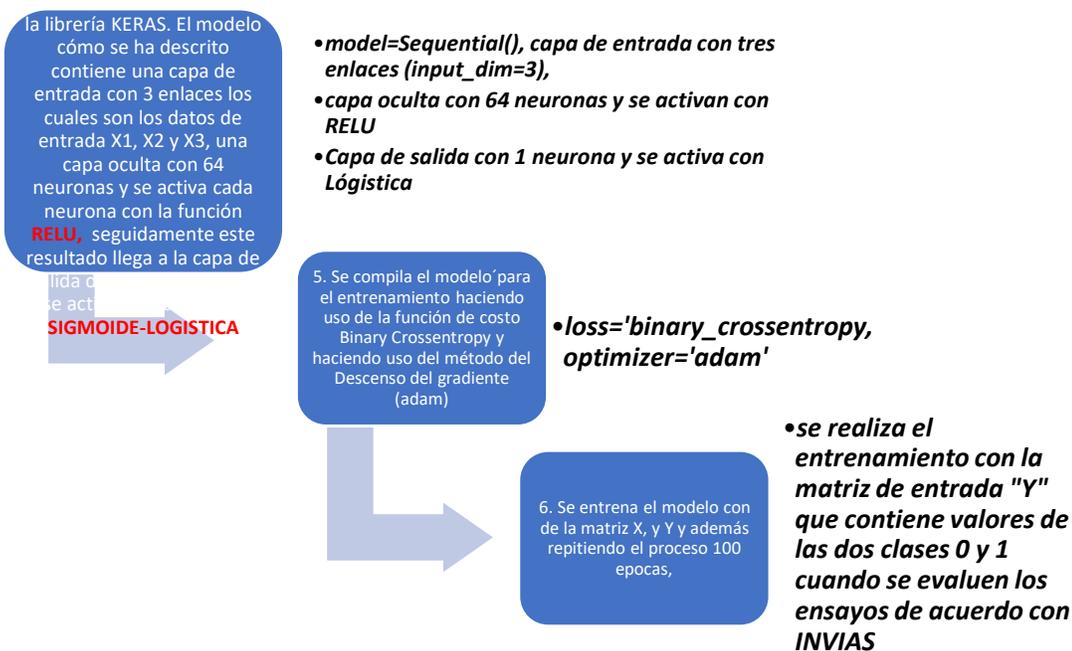


Figura 68: Estructuración para desarrollar el modelo de Red Neuronal.



Fuente: Realizado por el Autor.

Parte del código para crear y entrenar el modelo.

Figura 69: Parte del código para entrenar el modelo de red neuronal.

```
#Función para evaluar los requisitos de un ensayo
def cumple_requisitos(ensayo):
    if (ensayo['Dureza_maquina_angeles']>50 and
        4<ensayo['IP']<=9 and
        ensayo['CBR']>=15):
        return True
    else:
        return False

# Cargar los datos desde la base de datos Django
datos_fuente = dashboard.objects.values('Dureza_maquina_angeles', 'IP', 'CBR')

# Etiquetar los datos según si cumplen o no con los requisitos
y = np.array([cumple_requisitos(datos) for datos in datos_fuente])

# Convertir los datos a un arreglo de numpy
X = np.array([[datos_fuentes['Dureza_maquina_angeles'], datos_fuentes['IP'], datos_fuentes['CBR']] for datos_fuentes in datos_fuente])

# Construir el modelo de red neuronal
model = Sequential()
model.add(Dense(64, input_dim=3, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

Fuente: Por el Autor

Figura 70: Parte del código para entrenar el modelo de red neuronal.

```
# Compilar el modelo
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Entrenar el modelo
model.fit(X, y, epochs=100, batch_size=32)

# Ruta de la carpeta para guardar el modelo (reemplaza 'ruta_absoluta' con la ruta real)
ruta_carpeta_modelo = Path('C:/Users/IVAN ERAZO/OneDrive - unicauca.edu.co/TRABAJO DE GRADUACION')
# Crear la carpeta si no existe
ruta_carpeta_modelo.mkdir(parents=True, exist_ok=True)

# Guardar el modelo en la nueva carpeta
ruta_archivo_modelo = ruta_carpeta_modelo / 'modelo.keras'
model.save(str(ruta_archivo_modelo))
```

Fuente: Por el Autor

3.4.2. Búsquedas de las fuentes de material a partir de los requerimientos

Una vez entrenada la red neuronal, se ha realizado la predicción para analizar si los datos de entrada en el formulario X1, X2 y X3 cumplen de acuerdo con el artículo 311-E13 INVIAS para Afirmados

- X1: Ensayo en la máquina de los Ángeles (Desgaste)
- X2: Índice de Plasticidad
- X3: Ensayo de CBR.

Además de la predicción si estos datos de entrada cumplen con los requisitos de entrada, el modelo predictivo encuentra a las fuentes de extracción con buenas características candidatas a ser explotadas. A continuación se realiza el modelamiento para analizar los resultados:

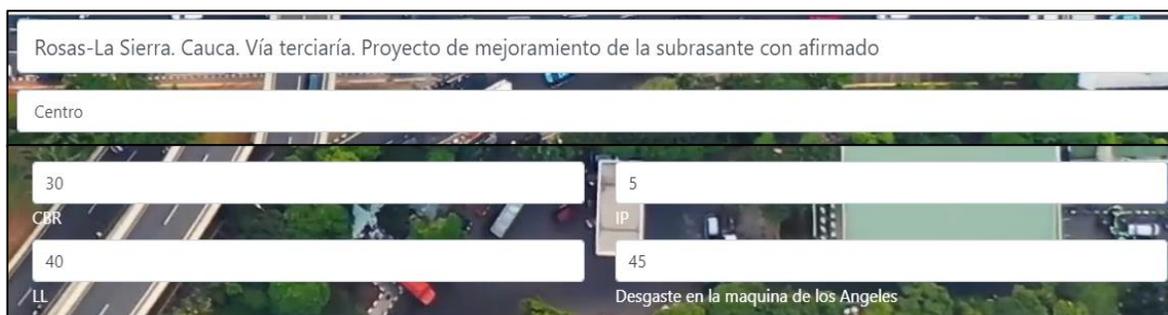
En la sección de “Procesos” se diligencia el formulario de acuerdo con las exigencias que se tenga para un proyecto, al final el modelo predice si estos datos cumplen o no con los requisitos con INVIAS e indica que fuentes de extracción sirven para el proyecto dependiendo de las exigencias.

Es importante aclarar que debido a que el número de fuentes en la base de datos es pequeño, con un valor de 23 Fuentes de extracción capturadas, el modelo predictivo encuentra rápidamente aquellas que estén acorde a las exigencias técnicas, este resultado lo genera así los datos de entrada en el formulario sean datos que no cumplan o en este

caso, resultados de ensayos no favorables, al final el modelo predictivo siempre busca las fuentes de extracción candidatas a ser explotadas por los valores en sus ensayos debido a la poca cantidad de canteras y que resultaría poco práctico hacer un modelo predictivo filtrando lugares o no incluyendo fuentes en la predicción. A continuación se realiza el ejemplo:

- Vía terciaria: Rosas-La Sierra. Proyecto de Mejoramiento de la subrasante con Afirmado con las siguientes características.
 - Dureza:45
 - IP:5
 - CBR:30

Figura 71: Entrada de datos para generar la predicción.



Fuente: Por el Autor

Resultado de la predicción del modelo en cuanto a si estos datos de entrada cumplen o no cumplen con INVIAS.

Figura 72: Resultado a partir de los datos de entrada.



Fuente: Por el Autor

Como se puede observar, la entrada de datos para X1, X2 y X3 cumplen con los establecido por INVIAS.

Resultados de las fuentes de extracción candidatas a ser explotadas (generadas por el modelo predictivo)

Figura 73: Resultados de la predicción del modelo para las Fuentes de Extracción candidatas a ser explotadas.

DUREZA	IP	CBR
41,0	0,0	21,9
42,0	0,0	21,9
21,0	0,0	78,6
32,0	0,0	76,8
24,0	0,0	43,8

Nombre de la Fuente
Santa Barbara
Bocas del Palo
Capellanía
Santa Bárbara
Vereda Pilamo

Fuente: Por el Autor

El modelo predictivo, busca en su base de datos las fuentes de extracción que cumplen con la mayoría de los parámetros con INVIAS, el resultado obtenido se muestra en la Figura 73. Estas fuentes de extracción cumplen con X1 y X3, sin embargo se observa que el índice de plasticidad es 0. Esta explicación puede ser que la fuente de extracción al ser un material de río, el material fino se lavó y por ende estos ensayos no se reportan o sencillamente son 0 para poder asignarles una variable numérica y no una cadena “string”, la cual se haría complejo manipularla en el entrenamiento de la red neuronal. Estas fuentes se pueden analizar en la sección “Dashboard” para analizar sus gráficas en cuanto a los ensayos.

Vista general de las gráficas X1:Dureza, X2:índice de Plasticidad, X3:Ensayo de CBR

Figura 74: Gráfica de CBR.



Fuente: Por el Autor

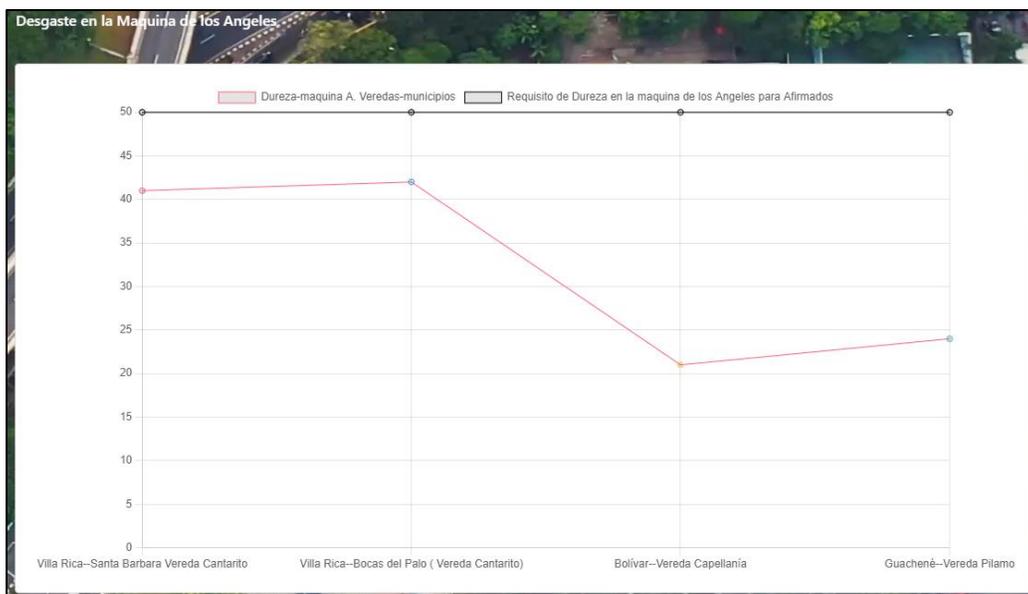
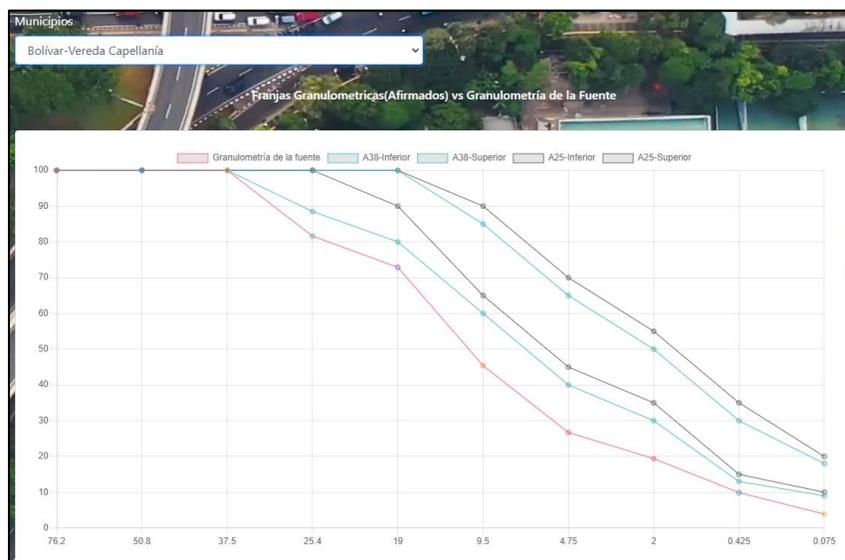


Figura 75: Gráfico de Dureza. Fuente: Por el Autor

En la sección de “Soluciones” se puede visualizar la granulometría y si se adapta a alguna de las franjas por INVIAS, sin embargo como se había mencionado anteriormente, debido a que muchas de estas fuentes de extracción de la base de datos no se ajustan a las franjas granulométricas se ve necesario como trabajo futuro crear una herramienta para lograr combinar dos fuentes de extracción, mirar el comportamiento en cuanto a los ensayos X1, X2 y X3, posterior realizar el entrenamiento de esta base de datos para generar predicciones de combinaciones y como se comportaría en cuanto a los ensayos de Dureza,

Índice de Plasticidad y CBR. En la imagen que se presenta a continuación se observa la granulometría para la fuente de extracción “Capellania-Bolivar Cauca”.

Figura 76: Granulometría de la fuente generada por el modelo de predicción.



Fuente: Por el Autor

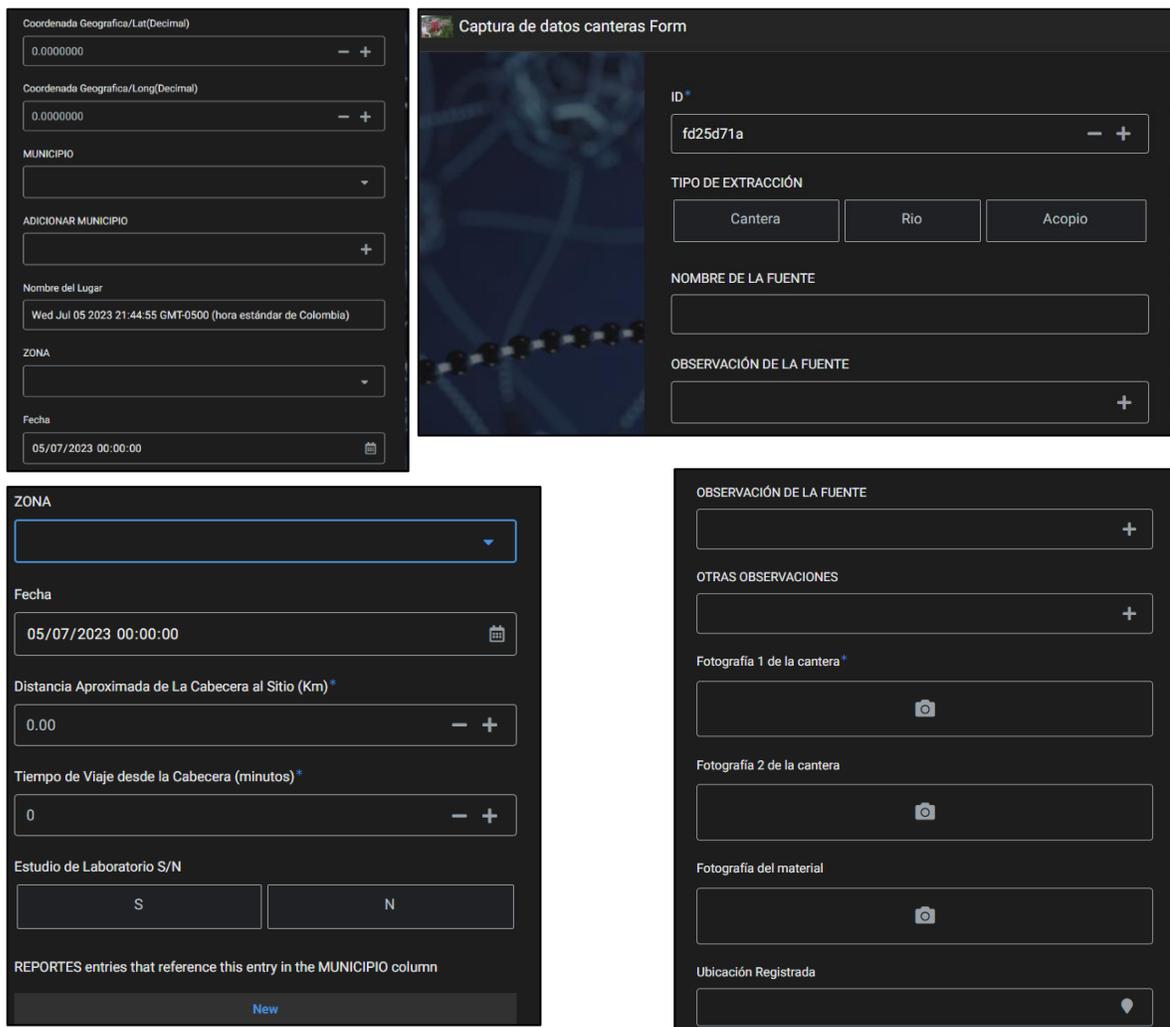
Estas herramientas proporcionan información para lograr comprender la respuesta de la predicción del modelo.

Capítulo 4. Toma de datos

4.1. Registros en campo

La información registrada en campo para las fuentes de extracción visitadas y luego analizadas en laboratorio por parte del CDT Vial de la universidad del Cauca se ha realizado a través de la creación del software Canteras-Móvil. Para el diseño de entrada de datos se ha buscado seguir una metodología primando el minimalismo, esto con la finalidad de no generar confusiones y pueda ser fácil y sencillo la captura de datos. El formulario que se deberá seguir tiene las siguientes características.

Figura 77: Formulario para captura de información de una Fuente de extracción en Campo mediante Canteras-Móvil.



Coordenada Geografica/Lat(Decimal)
0.0000000 - +

Coordenada Geografica/Long(Decimal)
0.0000000 - +

MUNICIPIO
[dropdown]

ADICIONAR MUNICIPIO
+

Nombre del Lugar
Wed Jul 05 2023 21:44:55 GMT-0500 (hora estándar de Colombia)

ZONA
[dropdown]

Fecha
05/07/2023 00:00:00

ZONA
[dropdown]

Fecha
05/07/2023 00:00:00

Distancia Aproximada de La Cabecera al Sitio (Km)*
0.00 - +

Tiempo de Viaje desde la Cabecera (minutos)*
0 - +

Estudio de Laboratorio S/N
S N

REPORTES entries that reference this entry in the MUNICIPIO column
New

Captura de datos canteras Form

ID*
fd25d71a - +

TIPO DE EXTRACCIÓN
Cantera Rio Acopio

NOMBRE DE LA FUENTE
[text field]

OBSERVACIÓN DE LA FUENTE
[text field] +

OBSERVACIÓN DE LA FUENTE
[text field] +

OTRAS OBSERVACIONES
[text field] +

Fotografía 1 de la cantera*
[camera icon]

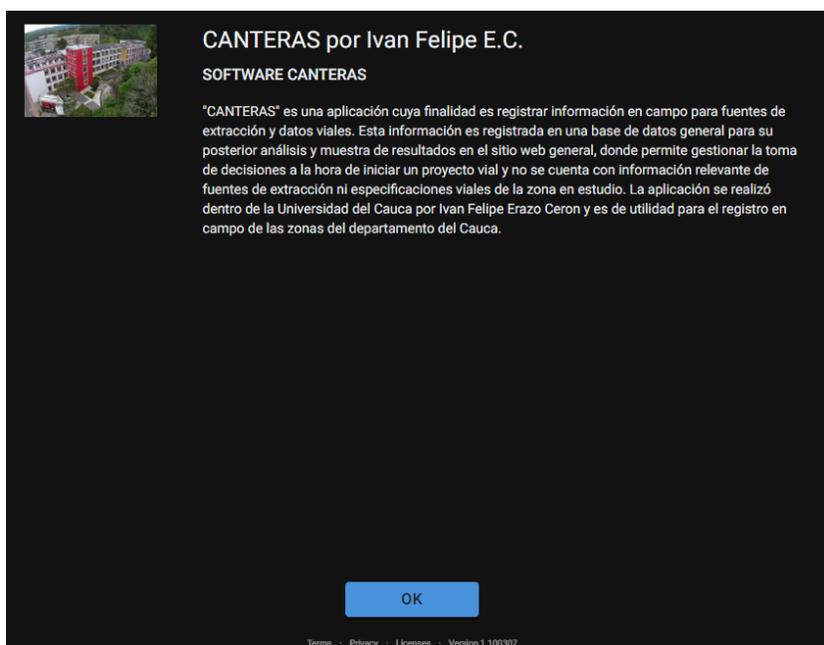
Fotografía 2 de la cantera
[camera icon]

Fotografía del material
[camera icon]

Ubicación Registrada
[location pin icon]

Fuente: Por el Autor

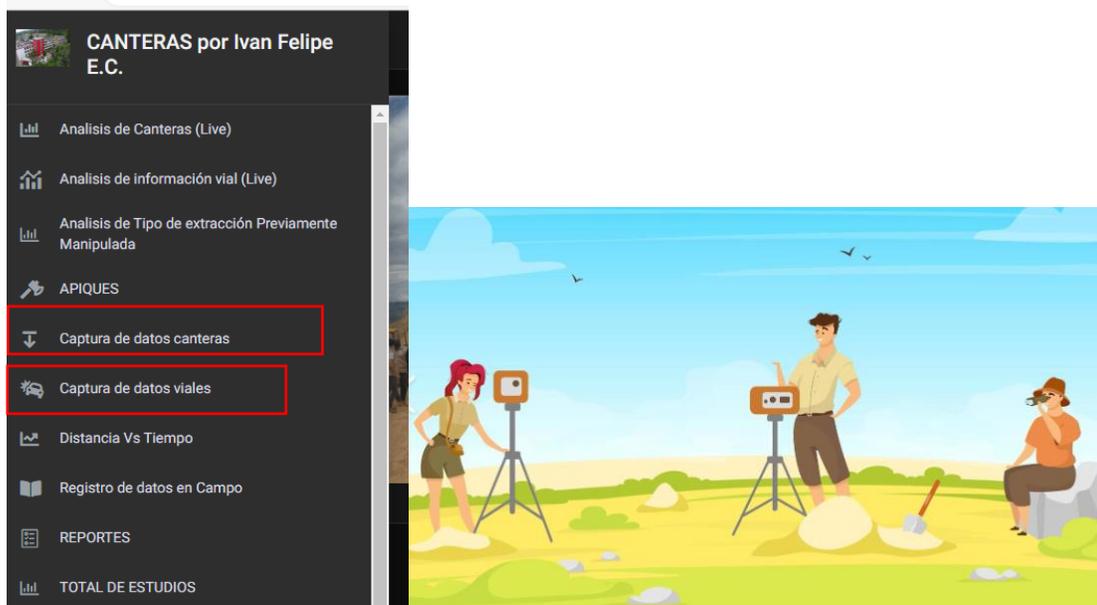
Figura 78: Interfaz inicial del Software Canteras-Móvil.



Fuente: Por el Autor

Seguir la información del formulario resulta sencilla, al final se guarda la información capturada en campo o en laboratorio, dependiendo la sección donde se entre a registrar la información:

Figura 79: Captura de datos en campo mediante Canteras-Móvil.



Fuente: Por el Autor.

La sección de Reportes indica la información capturada en laboratorio para una determinada fuente de extracción. Este formato puede ser adjuntando en archivo .xlsx, .txt y .pdf. Las siguientes imágenes muestran un ejemplo de reporte realizado por el CDT Vial y adjunto al software Canteras-Móvil en formato PDF.



Figura 80: Visualización del reporte adjuntando. En este caso reporte realizado por CDT Vial para las fuentes de extracción en: El Tambo Cauca

RESUMEN DE RESULTADOS DE LA CARACTERIZACIÓN FÍSICA EFECTUADA A LOS AGREGADOS DE LAS CANTERAS MUNICIPIO DE EL TAMBO, CAUCA									
Característica	Norma de ensayo INV	Requisito	Cantera Magines	Cantera Chisquío					
Dureza (O)									
Desgaste en la máquina de los ángeles a quinientas (500) revoluciones (Granulometría A), máximo (%).	E-218	50	52	56					
-NT1		30							
-NT2		30							
-NT3		30							
Limpieza (F)									
Limite líquido, máximo (%).	E-125	35	40.8	26.4					
Índice de plasticidad (%).	E-125 y E-126	4 - 9	17.7	10.8					
Resistencia del material (F)									
CBR (%): Porcentaje asociado al grado de compactación mínimo especificado (numeral 311.5.2.2.2); el CBR se debe medir sobre muestras sometidas previamente a cuatro días (4 días) de inmersión.	E-148	≥ 15	50.7	25.8					
Tipo de Gradación	Tamiz (mm/U.S Standard)								
	1½ Pulgadas	1 Pulgada	¾ Pulgada	½ Pulgada	Nro. 4	Nro. 10	Nro. 40	Nro. 200	
	Pasa Tamiz (%)								
A-38	100	-	80 - 100	60 -85	40 - 65	30 - 50	13 -30	9 -18	
A-25	-	100	90 - 100	65 -90	45 - 70	35 - 55	15 -35	10 -20	

Fuente: CDT Vial

4.2. Funcionalidades de las aplicaciones: Canteras-Móvil y Canteras-Web.

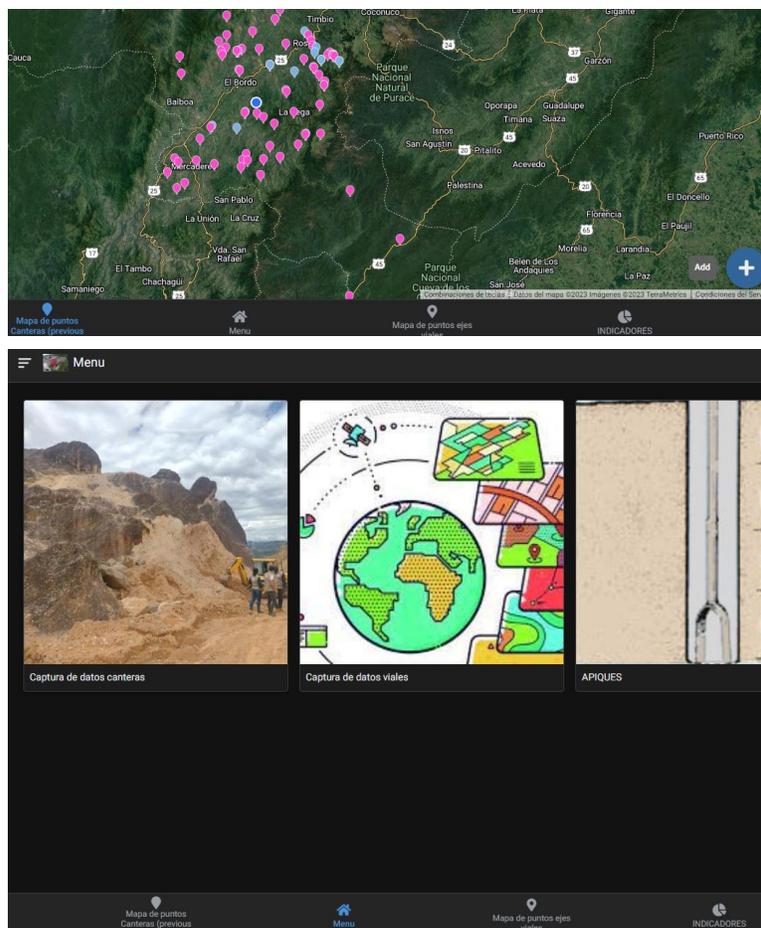
4.2.1. Canteras-Móvil

Canteras-Móvil es una aplicación cuya finalidad es el registro de datos en campo y de laboratorio, además de ser utilizada como herramienta de captura de datos, también permite observar estadísticas sobre el registro de datos, observar la ubicación en tiempo real, distancias de recorrido entre el punto de ubicación geográfica cualquiera y la fuente de extracción registrada o que se encuentre en la base de datos del software, adicionalmente permite gestionar, anexar, descargar información en formato PDF, xlsx y

Word, sin embargo lo recomendable es que se anexe información en formato PDF, esta información será de mucha utilidad cuando se tengan reportes de laboratorio sobre una fuente de extracción que se haya capturado o se encuentre en la base de datos del software. A continuación se muestra algunas de sus funciones más relevantes:

1. Captura de datos en Campo para Fuentes de Extracción:

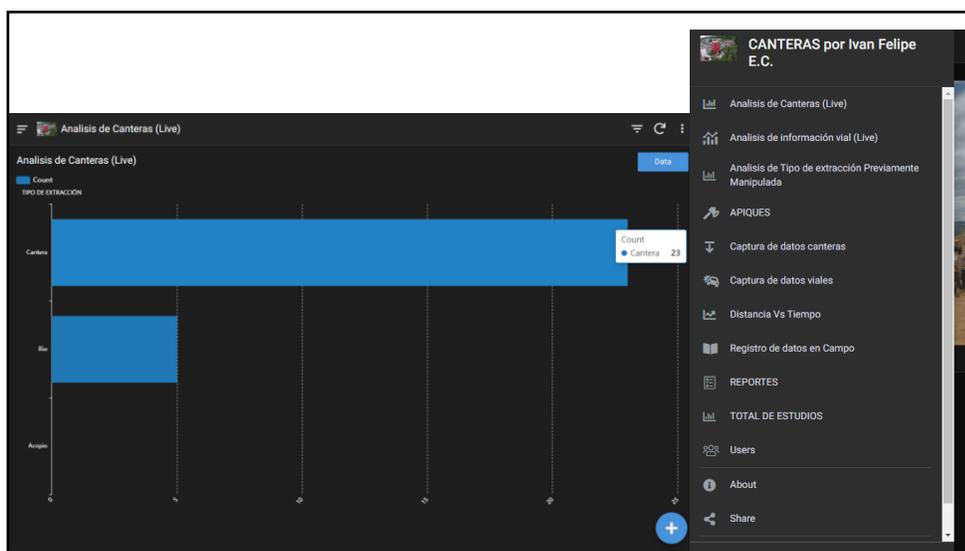
Figura 81: Opciones de Captura de datos en Campo en Canteras-Móvil.



Fuente: Por el Autor

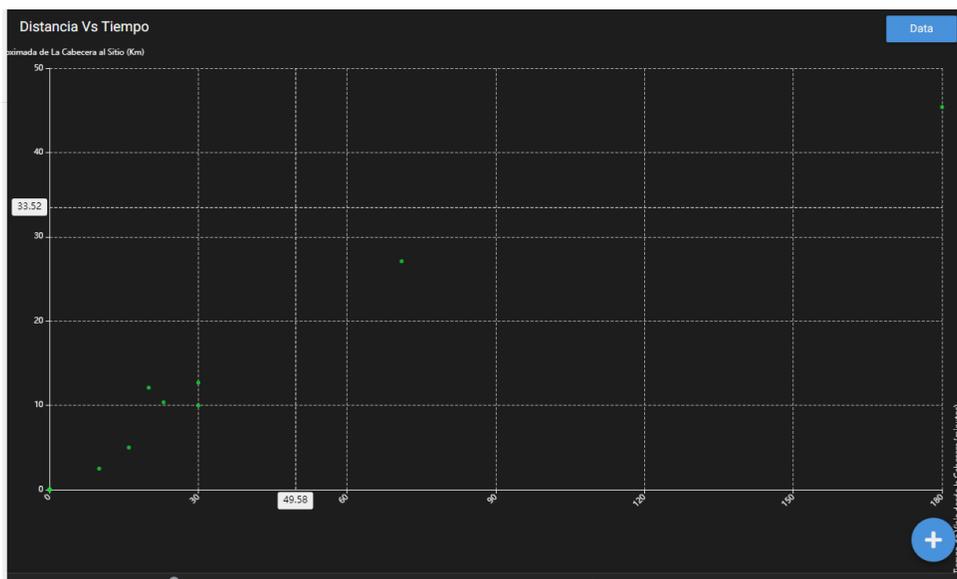
2. Información sobre Estadísticas de Registros.

Figura 82: Estadísticas de los datos registrados. Canteras-Móvil



Fuente: Por el Autor

Figura 83: Gráfico: Distancias de la fuente de extracción a la cabecera municipal Vs Tiempo de recorrido a la cabecera municipal. Canteras-Móvil



Fuente: Por el Autor

El software Canteras-Móvil tiene varias funcionalidades, sin embargo se puede decir que su objetivo principal es la entrada de datos sobre fuentes de extracción y datos viales, visualización de estadísticas, tiempos y distancias mínima de acarreo y en general la administración de datos de entrada.

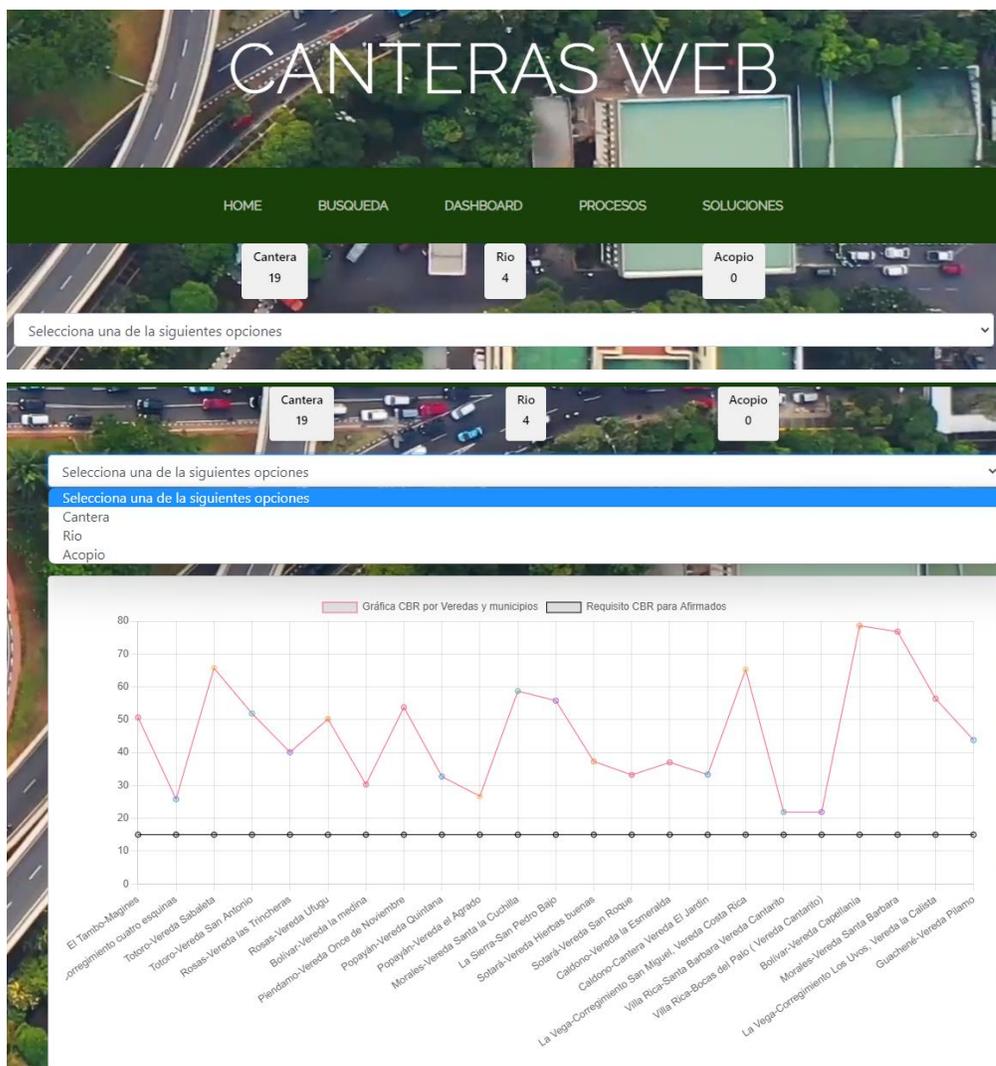
4.2.2. Canteras-Web

El software Canteras-Web tiene varias funcionalidades, sin embargo su principal objetivo es cumplir con el tema de la presente investigación. Lograr determinar rutas mínimas a partir de las fuentes de extracción para sus posibles visitas técnicas de campo y deducir la fuente potencialmente a ser explotada, ya que a partir del entrenamiento de la red neuronal artificial se logra determinar el lugar y la fuente de extracción que cumpla con los requisitos para Afirmados de acuerdo con la normatividad vigente del Instituto Nacional de Vías.

Algunas de sus funciones más relevantes son las siguientes:

1. Estadísticas sobre las fuentes de extracción que se tienen registros

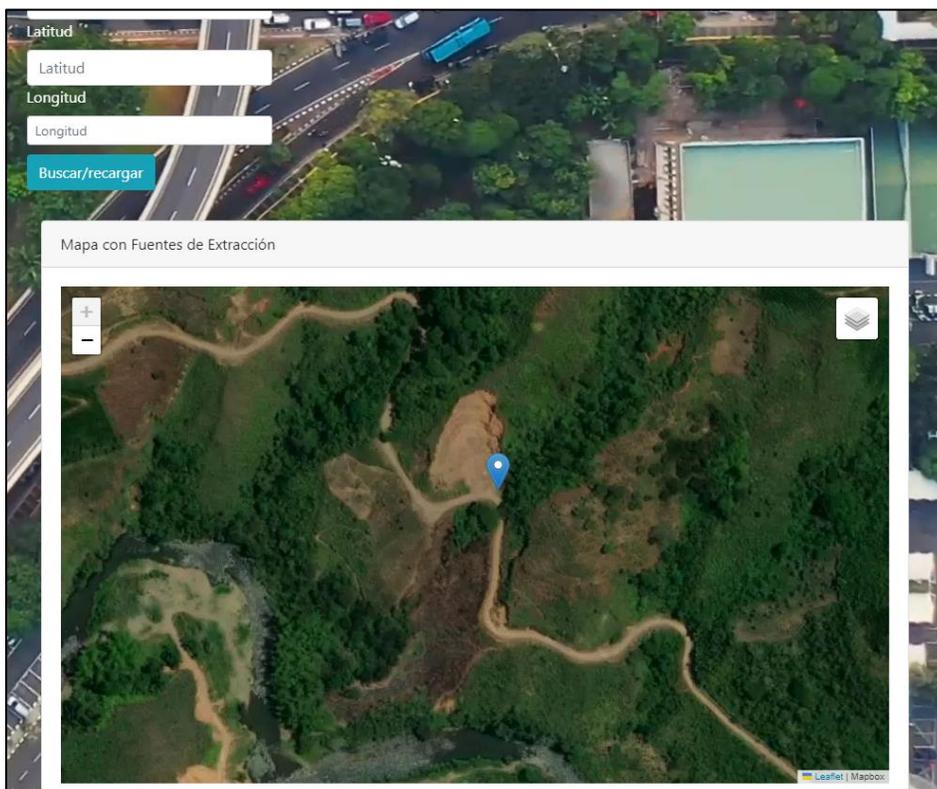
Figura 84: Estadística de ensayos para las fuentes de Extracción.



Fuente: Por el Autor

2. Búsqueda de fuentes de extracción con alta resolución de imágenes satelitales.

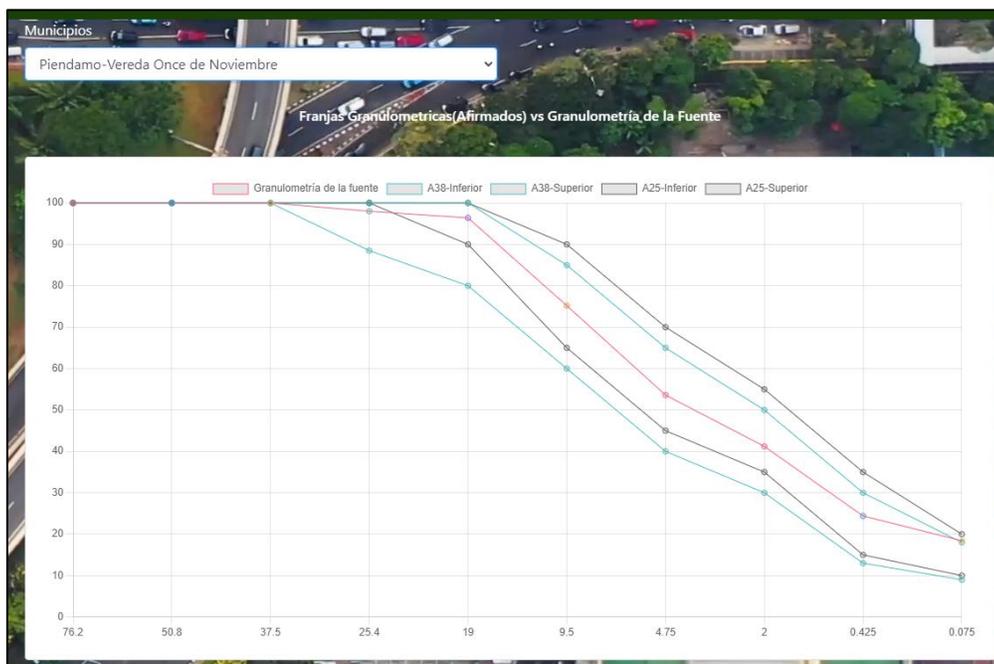
Figura 85: Imágenes satelitales de las Fuentes de Extracción.



Fuente por el Autor

3. Visualización de granulometrías de las diferentes fuentes de extracción en la base de datos que cumplan con los requisitos de las franjas granulométricas exigidas por el Instituto Nacional de Vías para el caso de Afirmados.

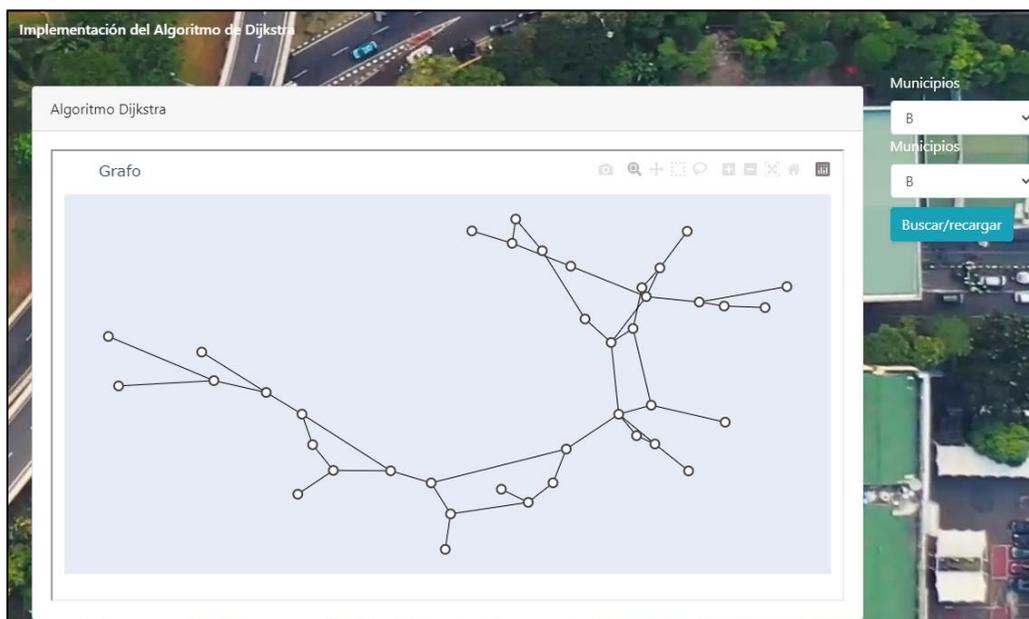
Figura 86: Franjas Granulométricas de las diferentes Fuentes de Extracción.



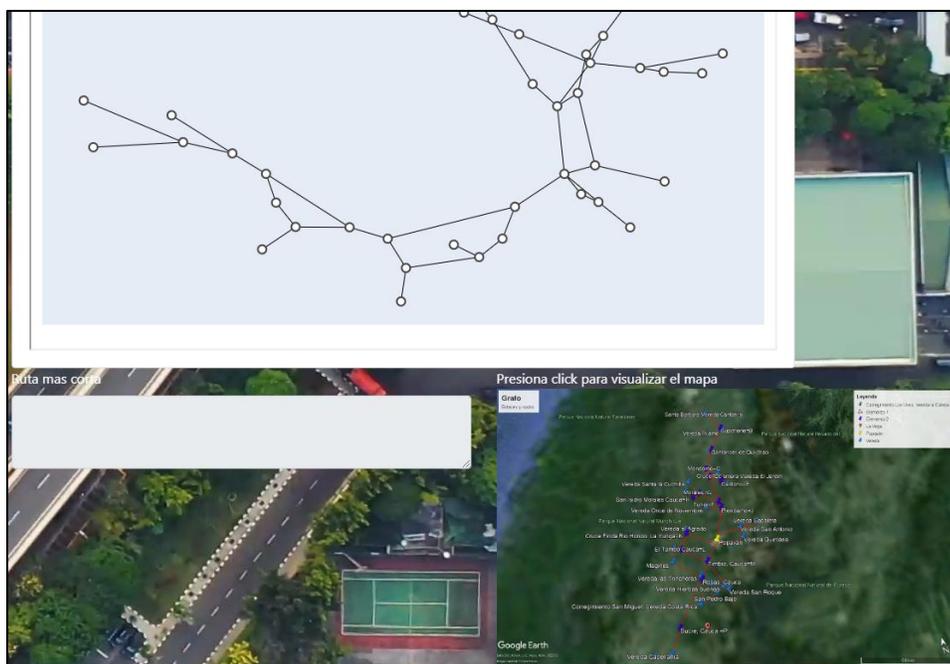
Fuente: Por el Autor

- Implementación del Algoritmo de Dijkstra para determinar rutas más cortas de visitas a diferentes fuentes de extracción y conocer los tiempos o distancias del recorrido a estas fuentes.

Figura 87: Implementación del Algoritmo de Dijkstra.

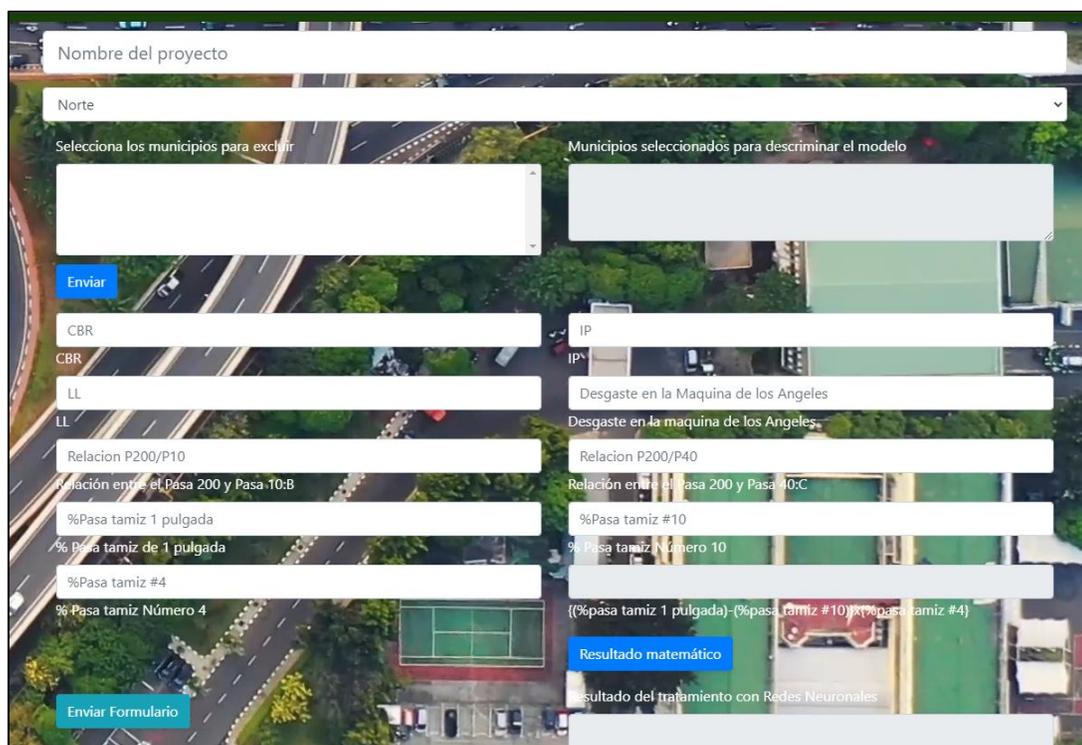


Fuente: Por el Autor



5. Determinación de la posible fuente de extracción a explotar a través de la implementación de Artificial Neuronal Networks.

Figura 88: Implementación de Redes Neuronales para determinar posibles fuentes de extracción a Explotar que cumpla con E-INVIAS. Fuente:



Nombre del proyecto

Norte

Selecciona los municipios para excluir

Municipios seleccionados para discriminar el modelo

Enviar

CBR

IP

Desgaste en la Maquina de los Angeles

Relacion P200/P10

Relacion P200/P40

Relación entre el Pasa 200 y Pasa 10:B

Relacion entre el Pasa 200 y Pasa 40:C

%Pasa tamiz 1 pulgada

%Pasa tamiz #10

%Pasa tamiz de 1 pulgada

%Pasa tamiz Número 10

%Pasa tamiz #4

((%pasa tamiz 1 pulgada) - (%pasa tamiz #10) x (%pasa tamiz #4))

%Pasa tamiz Número 4

Resultado matemático

Enviar Formulario

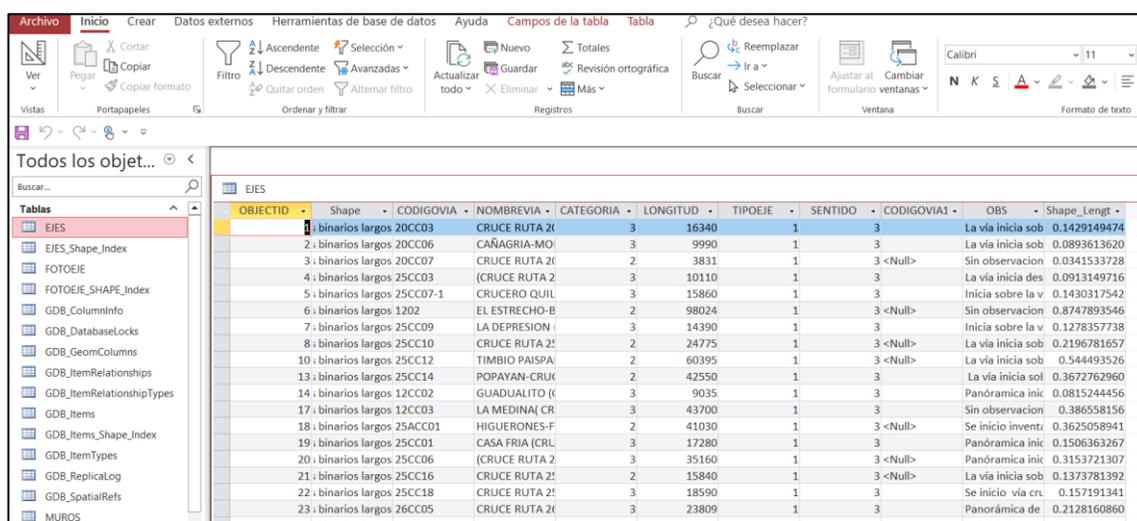
Resultado del tratamiento con Redes Neuronales

Fuente: Por el Autor

4.3. Base de datos inicial para CANTERAS-MÓVIL Y CANTERAS-WEB

Para la formación de la base de datos, es importante mencionar que se ha trabajado con información preliminar para su posible comprobación en campo de nuevos datos, esta base de datos inicial fue suministrada por la oficina de Infraestructura de la gobernación del Cauca a través del CDT Vial de la Universidad del Cauca. La información que se ha suministrado ha pasado por una serie de softwares que han permitido discriminar los datos que no entrarán para la conformación de una base de datos preliminar que entrará hacer parte del software Canteras-Móvil y Canteras-Web. Esta información ha sido procesada por Access, ArcMap y Excel.

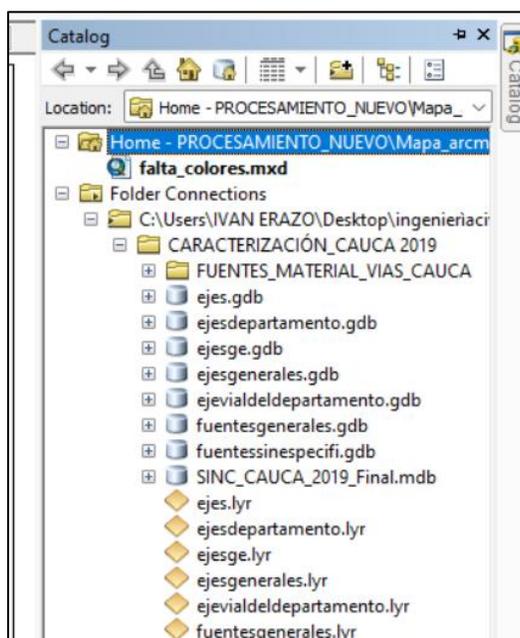
Figura 89: Conformación de la base de datos inicial. Procesamiento en Access.



OBJECTID	Shape	CODIGOVIA	NOMBRE VIA	CATEGORIA	LONGITUD	TIPOEJE	SENTIDO	CODIGOVIA1	OBS	Shape_Length
1	binarios largos 20CC03	CRUCE RUTA 2	3	16340	1	3	La vía inicia sob	0.1429149474		
2	binarios largos 20CC06	CAÑAGRIA-MO	3	9990	1	3	La vía inicia sob	0.0893613620		
3	binarios largos 20CC07	CRUCE RUTA 2	2	3831	1	3 <Null>	Sin observacion	0.0341533728		
4	binarios largos 25CC03	CRUCE RUTA 2	3	10110	1	3	La vía inicia des	0.0913149716		
5	binarios largos 25CC07-1	CRUCERO QUIL	3	15860	1	3	Inicia sobre la v	0.1430317542		
6	binarios largos 1202	EL ESTRECHO-B	2	98024	1	3 <Null>	Sin observacion	0.8747893546		
7	binarios largos 25CC09	LA DEPRESION	3	14390	1	3	Inicia sobre la v	0.1278357738		
8	binarios largos 25CC10	CRUCE RUTA 2	2	24775	1	3 <Null>	La vía inicia sob	0.2196781657		
10	binarios largos 25CC12	TIMBIO PAISPA	2	60395	1	3 <Null>	La vía inicia sob	0.544493526		
13	binarios largos 25CC14	POPAYAN-CRUC	2	42550	1	3	La vía inicia sol	0.3672762960		
14	binarios largos 12CC02	GUADUALITO (3	9035	1	3	Panorámica inic	0.0815244456		
17	binarios largos 12CC03	LA MEDINA(CR	3	43700	1	3	Sin observacion	0.386558156		
18	binarios largos 25ACCO1	HIGUERONES-F	2	41030	1	3 <Null>	Se inicio invent	0.3625058941		
19	binarios largos 25CC01	CASA FRIA (CR	3	17280	1	3	Panorámica inic	0.1506363267		
20	binarios largos 25CC06	(CRUCE RUTA 2	3	35160	1	3 <Null>	Panorámica inic	0.3153721307		
21	binarios largos 25CC16	CRUCE RUTA 2	2	15840	1	3 <Null>	La vía inicia sob	0.1373781392		
22	binarios largos 25CC18	CRUCE RUTA 2	3	18590	1	3	Se inicio vía cru	0.157191341		
23	binarios largos 26CC05	CRUCE RUTA 2	3	23800	1	3	Panorámica de	0.2128160860		
24	binarios largos 26CC06	CRUCE RUTA 2	3	16870	1	3	Se inicio invent	0.120376038		

Fuente: Por el Autor

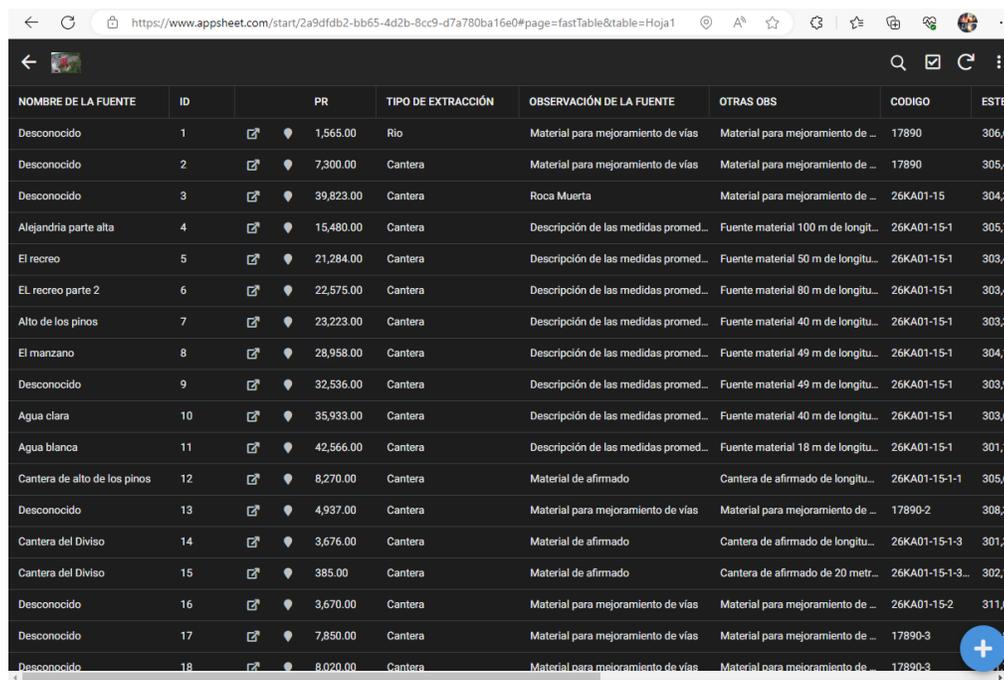
Figura 90: Formación de la estructura de datos geospaciales en ArcMap.



Fuente: Por el Autor

Finalmente se obtiene la base de datos preliminar que será parte del software Canteras-Móvil y Canteras-Web

Figura 91: Base de datos vista en Canteras-Móvil.

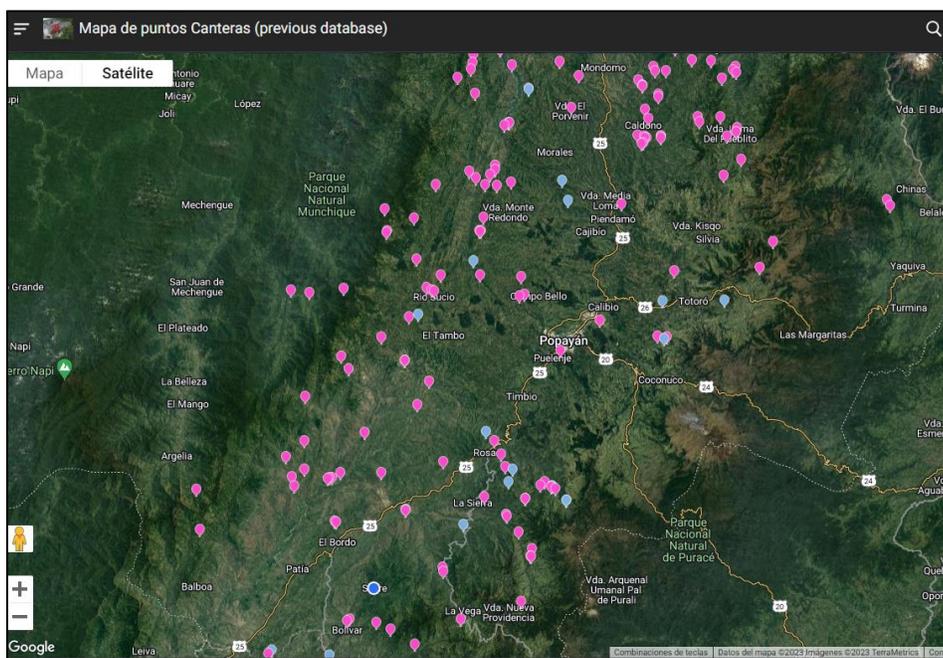


NOMBRE DE LA FUENTE	ID	PR	TIPO DE EXTRACCIÓN	OBSERVACIÓN DE LA FUENTE	OTRAS OBS	CODIGO	ESTE
Desconocido	1	1,565.00	Rio	Material para mejoramiento de vías	Material para mejoramiento de ...	17890	306,6
Desconocido	2	7,300.00	Cantera	Material para mejoramiento de vías	Material para mejoramiento de ...	17890	305,4
Desconocido	3	39,823.00	Cantera	Roca Muerta	Material para mejoramiento de ...	26KA01-15	304,3
Alejandro parte alta	4	15,480.00	Cantera	Descripción de las medidas promed...	Fuente material 100 m de longitu...	26KA01-15-1	305,7
El recreo	5	21,284.00	Cantera	Descripción de las medidas promed...	Fuente material 50 m de longitu...	26KA01-15-1	303,4
EL recreo parte 2	6	22,575.00	Cantera	Descripción de las medidas promed...	Fuente material 80 m de longitu...	26KA01-15-1	303,4
Alto de los pinos	7	23,223.00	Cantera	Descripción de las medidas promed...	Fuente material 40 m de longitu...	26KA01-15-1	303,3
El manzano	8	28,958.00	Cantera	Descripción de las medidas promed...	Fuente material 49 m de longitu...	26KA01-15-1	304,7
Desconocido	9	32,536.00	Cantera	Descripción de las medidas promed...	Fuente material 49 m de longitu...	26KA01-15-1	303,9
Agua clara	10	35,933.00	Cantera	Descripción de las medidas promed...	Fuente material 40 m de longitu...	26KA01-15-1	303,6
Agua blanca	11	42,566.00	Cantera	Descripción de las medidas promed...	Fuente material 18 m de longitu...	26KA01-15-1	301,7
Cantera de alto de los pinos	12	8,270.00	Cantera	Material de afirmado	Cantera de afirmado de longitu...	26KA01-15-1-1	305,6
Desconocido	13	4,937.00	Cantera	Material para mejoramiento de vías	Material para mejoramiento de ...	17890-2	308,3
Cantera del Divisio	14	3,676.00	Cantera	Material de afirmado	Cantera de afirmado de longitu...	26KA01-15-1-3	301,3
Cantera del Divisio	15	385.00	Cantera	Material de afirmado	Cantera de afirmado de 20 metr...	26KA01-15-1-3...	302,1
Desconocido	16	3,670.00	Cantera	Material para mejoramiento de vías	Material para mejoramiento de ...	26KA01-15-2	311,0
Desconocido	17	7,850.00	Cantera	Material para mejoramiento de vías	Material para mejoramiento de ...	17890-3	
Desconocido	18	8,020.00	Cantera	Material para mejoramiento de vías	Material para mejoramiento de ...	17890-3	

Fuente: Por el Autor

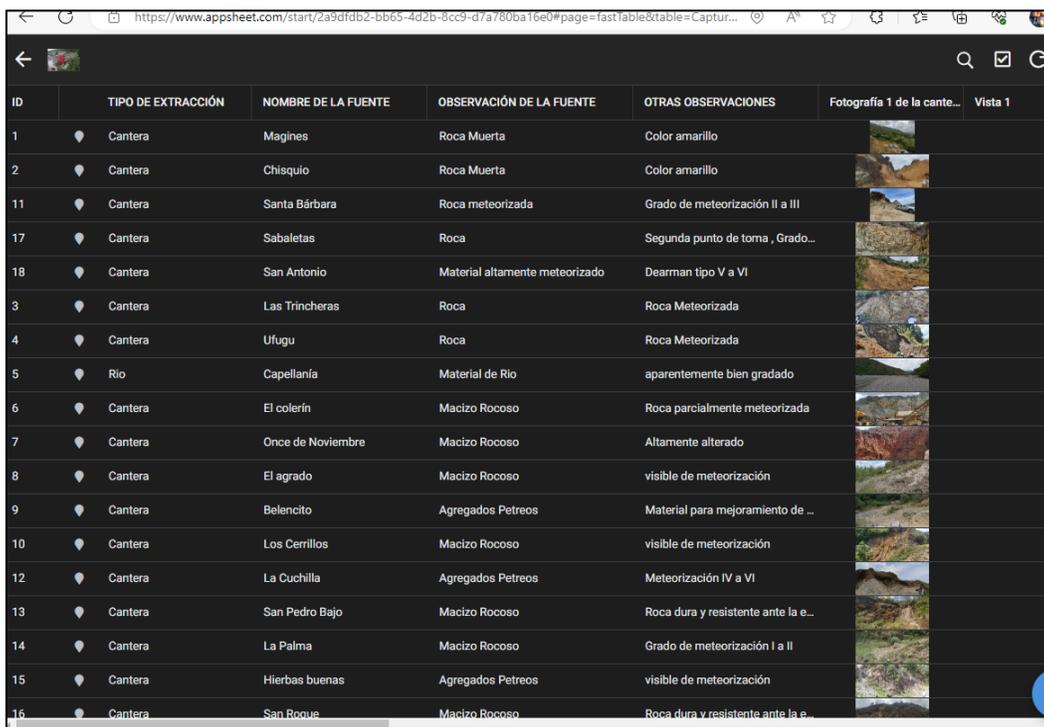
Seguidamente a la base de datos preliminar, se ha construido la base de datos final para ser entrenada con las redes neuronales artificiales, esta base de datos es construida a partir de la información capturada por el CDT Vial de la universidad del Cauca en campo y en laboratorio de las fuentes de extracción que se han tomado.

Figura 92: Base de datos inicial se visualiza datos geoespaciales en color rosa.



Fuente: Por el Autor

Figura 93: Base de datos realizada con la información captura por el CDT Vial de la Universidad del Cauca.



ID	TIPO DE EXTRACCIÓN	NOMBRE DE LA FUENTE	OBSERVACIÓN DE LA FUENTE	OTRAS OBSERVACIONES	Fotografía 1 de la cantera...	Vista 1
1	Cantera	Magines	Roca Muerta	Color amarillo		
2	Cantera	Chisquío	Roca Muerta	Color amarillo		
11	Cantera	Santa Bárbara	Roca meteorizada	Grado de meteorización II a III		
17	Cantera	Sabaletas	Roca	Segunda punto de toma , Grado...		
18	Cantera	San Antonio	Material altamente meteorizado	Dearman tipo V a VI		
3	Cantera	Las Trincheras	Roca	Roca Meteorizada		
4	Cantera	Ufugu	Roca	Roca Meteorizada		
5	Rio	Capellanía	Material de Rio	aparentemente bien gradado		
6	Cantera	El colerín	Macizo Rocoso	Roca parcialmente meteorizada		
7	Cantera	Once de Noviembre	Macizo Rocoso	Altamente alterado		
8	Cantera	El agrado	Macizo Rocoso	visible de meteorización		
9	Cantera	Belencito	Agregados Petreos	Material para mejoramiento de ...		
10	Cantera	Los Cerrillos	Macizo Rocoso	visible de meteorización		
12	Cantera	La Cuchilla	Agregados Petreos	Meteorización IV a VI		
13	Cantera	San Pedro Bajo	Macizo Rocoso	Roca dura y resistente ante la e...		
14	Cantera	La Palma	Macizo Rocoso	Grado de meteorización I a II		
15	Cantera	Hierbas buenas	Agregados Petreos	visible de meteorización		
16	Cantera	San Roque	Macizo Rocoso	Roca dura y resistente ante la e...		

Fuente: Por el Autor

Capítulo 5. Conclusiones, recomendaciones y trabajo futuro

A continuación se presentan las conclusiones y recomendaciones definidas a partir de proyecto realizado:

5.1. Conclusiones y recomendaciones

En cuanto a los datos capturados en campo, se ha determinado que parametrizar más las variables respecto a las observaciones realizadas por el especialista resulta mucho más fácil para generar modelos predictivos y obtener mucha más información respecto al comportamiento de una fuente de extracción, además el hecho de parametrizar variables y no generar observaciones generales puede mejorar el tiempo de procesamiento de la base de datos para desarrollar reportes, implementar modelos predictivos haciendo uso de redes neuronales, generación de gráficos, etc. En ese sentido se ha logrado comprender que entre más objetivo sea la caracterización de una fuente de extracción, mejor resulta el tiempo de cómputo para realizar el postproceso de la base de datos y el desarrollo de un

software. También es importante mencionar acerca de la funcionalidad de los softwares desarrollados.

La funcionalidad de encontrar rutas mínimas en Canteras-Móvil se ha podido comprender que es esencial para optimizar el proceso de acarreo de materiales desde las fuentes de extracción hasta el punto de intervención. En este proceso se ha utilizado técnicas de algoritmos de grafos para calcular las rutas que minimizan la distancia y el tiempo de acarreo. El software, mediante la ayuda de la base de datos de Google Maps proporciona a los usuarios la capacidad de ingresar ubicaciones de origen y destino, y luego calcula automáticamente la ruta más eficiente, teniendo en cuenta las restricciones del terreno y las condiciones de la carretera, etc. Esto puede tener un impacto significativo en la eficiencia operativa de las canteras, reduciendo costos y mejorando la logística de transporte.

Canteras-Web ha implementado un sistema de grafos para determinar las rutas mínimas para realizar visitas técnicas a varias fuentes de extracción. Esta funcionalidad es crucial para la planificación de visitas y evaluaciones de campo en múltiples ubicaciones. Mediante la creación de un grafo que conecta las fuentes de extracción y los puntos de interés, el software calcula las rutas más cortas para cada visita técnica, minimizando la distancia recorrida y optimizando el tiempo de viaje. Esto puede aumentar la eficiencia de las actividades de inspección y evaluación, así como reducir los costos asociados con el transporte y el tiempo de desplazamiento.

En cuanto a la utilización de Redes Neuronales Artificiales y generar un modelo predictivo a través de la creación del software Canteras-Web genera un espectro de trabajo grande para futuros aspectos que se pueda incorporar en el software, ya que, la implementación de un modelo predictivo basado en redes neuronales es un enfoque innovador para predecir si una fuente de extracción cumple con los requisitos de INVIAS. Sin embargo, la disponibilidad limitada de datos de ensayos de laboratorio de la universidad del Cauca planteó desafíos en la formación del modelo. El desbalance de clases, donde hay más datos de una clase que de la otra, puede afectar la capacidad del modelo para generalizar.

Para abordar este problema, se realizó un aumento de datos ficticios y se ajustó la forma en que el modelo interpreta y clasifica los datos. Esto ayudó a mitigar el sobreajuste y mejoró la capacidad del modelo para predecir con precisión. Además, la normalización de los datos y la elección adecuada de funciones de activación y función de pérdida son esenciales para un rendimiento óptimo.

Es importante destacar que los resultados de la predicción deben ser interpretados en función del umbral de decisión elegido. Los modelos basados en redes neuronales, como cualquier modelo predictivo, tienen limitaciones y se deben evaluar en un contexto más amplio.

El análisis de la funcionalidad de encontrar rutas mínimas y la determinación de rutas cortas para visitas técnica en Canteras-Móvil y Canteras-Web demuestra el valor de la optimización y la planificación que se ha venido buscando. Además, el enfoque de aprendizaje automático en el modelo predictivo proporciona una herramienta innovadora para evaluar el cumplimiento con los requisitos de INVIAS en fuentes de extracción.

Sin embargo, se debe considerar que estos sistemas dependen en gran medida de la calidad y cantidad de datos de entrada. En el caso del modelo predictivo, la disponibilidad limitada de datos generó desafíos que requirieron ajustes para obtener resultados satisfactorios. Para futuras investigaciones, es recomendable continuar recolectando y ampliando la base de datos con datos de ensayos de laboratorio reales, lo que podría mejorar aún más la precisión del modelo.

En resumen, el software Canteras-Móvil y Canteras-Web ofrece soluciones prácticas y avanzadas para la industria, y el modelo predictivo basado en redes neuronales muestra el potencial de la tecnología de aprendizaje automático para abordar desafíos específicos. Sin embargo, la calidad y representatividad de los datos siguen siendo un factor clave en el éxito de estas soluciones.

El desarrollo de una estrategia para combinar múltiples fuentes de extracción y evaluar la granulometría resultante, junto con la incorporación de predicciones mejoradas de propiedades de materiales, podría aportar significativamente a la optimización de la operación de canteras.

5.2. Trabajo Futuro

El trabajo a realizar en otros proyectos se dividió en los títulos a continuación:

Combinación de Fuentes y Evaluación de Granulometría:

Definición de Criterios de Combinación: Establecer criterios claros para combinar las fuentes de extracción. Esto puede incluir propiedades como ubicación geográfica, tipo de material, disponibilidad, entre otros. También se pueden definir objetivos de granulometría específicos que deben cumplirse según las franjas granulométricas exigidas por INVIAS.

Modelado de Combinación: Utilizar técnicas de mezcla de materiales para determinar la proporción adecuada de cada fuente que cumpla con los objetivos de granulometría. Esto puede involucrar modelos matemáticos o incluso algoritmos de optimización que consideren las restricciones y objetivos definidos.

Realización de Ensayos: Realizar ensayos de laboratorio en la mezcla resultante para evaluar su comportamiento en términos de DUREZA, Índice de Plasticidad y CBR. Estos ensayos proporcionarán datos cruciales para comprender cómo la combinación de fuentes afecta las propiedades de los materiales.

Mejora del Modelo Predictivo:

Recopilación de Datos: Continuar recopilando datos de ensayos de laboratorio reales de las mezclas resultantes. Cuanto más variados y representativos sean los datos, más sólida será la base para el entrenamiento del modelo predictivo.

Entrenamiento y Ajuste del Modelo: Utilizar los datos recopilados para entrenar y ajustar el modelo predictivo. Es posible que necesite ajustes en la arquitectura de la red neuronal, funciones de activación, función de pérdida, entre otros, para reflejar las propiedades de las mezclas combinadas.

Abordar el Desbalance de Clases: Dado que es probable que las mezclas cumplan con los requisitos de INVIAS con más frecuencia que las fuentes individuales, el desbalance de clases podría persistir. Puede considerar técnicas como sobremuestreo, submuestreo o pesos de clase para manejar este desafío.

Implementación de Predicciones Mejoradas:

Integración en la Plataforma: Una vez que el modelo predictivo haya sido mejorado y validado, intégrealo en la plataforma Canteras-Móvil o Canteras-Web. Esto permitirá a los usuarios predecir el comportamiento de las mezclas resultantes en términos de DUREZA, Índice de Plasticidad y CBR antes de implementarlas en la operación real.

Evaluación y Validación Continua:

Monitoreo de Predicciones: Realice un monitoreo continuo de las predicciones del modelo en relación con los resultados reales de los ensayos de laboratorio. Esto permitirá ajustar y mejorar el modelo a medida que se acumulen más datos.

Retroalimentación de los Usuarios: Obtenga retroalimentación de los usuarios de la plataforma sobre la precisión de las predicciones. Esto puede ayudar a identificar posibles áreas de mejora y optimización adicional.

Impacto y Beneficios:

La integración de predicciones mejoradas en la plataforma proporcionará a los usuarios información valiosa para la toma de decisiones, permitiendo una planificación más precisa y eficiente de las operaciones de extracción y acarreo.

La evaluación de la granulometría resultante y su comportamiento en relación con los requisitos de INVIAS contribuirá a la calidad y estandarización de los materiales extraídos.

La combinación de fuentes puede permitir la optimización de recursos y la reducción de costos de transporte y operación.

En resumen, esta propuesta de trabajo futuro aborda la integración de datos, modelado y predicciones para optimizar aún más la operación de canteras. La combinación de fuentes y la evaluación de granulometría junto con la mejora del modelo predictivo

Capítulo 6. Bibliografía

- Agencia Nacional de Infraestructura(ANI). (n.d.). *2.3 Colombia Red Carretera* (pp. 1–5).
- Bobadilla, J. (2020). *Machine Learning y Deep Learning usando Python, Scikit y Keras* (R. Editorial (ed.); Ediciones).
- Darío, H., & Orobio, A. (2015). *Effects of uncertainty on scheduling of highway construction projects Efectos de la incertidumbre en la programación de proyectos de construcción de carreteras*.
- Departamento Administrativo Nacional de Estadística. (2021). Pobreza monetaria en Colombia. *Boletín Técnico*, 1–85.
- Departamento Nacional de Planeación (DNP). (2020). Mejoramiento de vías terciarias -vías de tercer orden. *Proyectos TIPO. Soluciones Ágiles Para Un Nuevo País*, 2, 79.
- Django Software Foundation. (n.d.). *Django*. <https://www.djangoproject.com>
- Duque, R. G. (n.d.). *Python PARA TODOS*.
- Gallastegui, L. M. G. (2022). Inteligencia Artificial: In *Miradas sobre el emprendimiento ante la crisis del coronavirus*. <https://doi.org/10.2307/j.ctv2gz3w9c.97>
- Gamarra-Vergara, J. (2008). Economía del Departamento del Cauca : concentración de tierras y pobreza. *Aragón*, 95.
- Khanacademy. (n.d.). *Introducción a HTML y CSS*. <https://es.khanacademy.org/computing/computer-programming/html-css>
- Mdn web docs. (2023). *JavaScript*. Marzo, 2023. https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/What_is_JavaScript
- Pantoja, N. (2018). *Vías Terciarias En Colombia* : 39. http://sogeocol.edu.co/documentos/mp_20180104.pdf
- Papert, M. (1969). *Perceptrons*.
- Rosenblatt, 1962. (n.d.). *El Perceptrón*.
- Salcedo, R. (2016). *Teoría de Grafos*.
- Sclar, M. (2016). *Camino mínimo en grafos Contenidos*.
- Tech, R. (2022). *Funciones de Activación*.
- Vías, I. nacional. (2008). *INVIAS*.

Capítulo 7. Anexos

CÓDIGO PARA LA EJECUCIÓN DEL ALGORITMO DE DIJKSTRA

```

20 #/*****GRAFOS*****/
21 #función para el grafo y el algoritmo de dijkstra
22 def generar_grafica(request):
23     # Obtener los datos del modelo "Grafo"
24     datos = grafo.objects.all()
25
26     # Crear el grafo vacío
27     G = nx.Graph()
28     #Crear un diccionario para almacenar las etiquetas de los NODOS
29     etiquetas={}
30
31     # Agregar los nodos y las aristas al grafo y adicional las etiquetas de los nodos
32     for dato in datos:
33         G.add_edge(dato.NODO, dato.CONEXION, weight=int(dato.DISTANCIA))
34         # Agregar etiquetas a los nodos desde la base de datos
35         etiquetas[dato.NODO] = dato.NODO # Aquí se asume que el nombre de nodo es la etiqueta
36         # Asegúrate de que el atributo "weight" esté configurado correctamente en las aristas
37         G[dato.NODO][dato.CONEXION]["weight"] = dato.DISTANCIA
38
39     # Asignar las etiquetas a los nodos
40     nx.set_node_attributes(G, etiquetas, 'label')

```

```

#ruta mínima
ruta_mas_corta = ''
distancias_entre_nodos=''
if request.method == 'POST':
    origen = request.POST.get('origen')
    destino = request.POST.get('destino')
    # Calcular la ruta más corta utilizando el algoritmo de Dijkstra
    ruta_mas_corta = nx.dijkstra_path(G, origen, destino)

    # Obtener las distancias de la ruta mínima
    # distancias_ruta = []
    # for i in range(len(ruta_mas_corta) - 1):
    #     u = ruta_mas_corta[i]
    #     v = ruta_mas_corta[i + 1]
    #     distancia = G[u][v]["weight"]
    #     distancias_ruta.append(distancia)

    # Obtener las distancias entre cada par de nodos en la ruta mínima
    distancias_entre_nodos = {}
    for i in range(len(ruta_mas_corta) - 1):
        u = ruta_mas_corta[i]
        v = ruta_mas_corta[i + 1]
        distancia = G[u][v]["weight"]
        distancias_entre_nodos[f'Distancia entre {u} y {v}'] = distancia

    # Ahora distancias_entre_nodos contiene las distancias entre cada par de nodos
    print(distancias_entre_nodos)

```

```

# Generar la visualización del grafo utilizando Plotly
pos = nx.spring_layout(G)
edge_x = []
edge_y = []
for edge in G.edges():
    x0, y0 = pos[edge[0]]
    x1, y1 = pos[edge[1]]
    edge_x.extend([x0, x1, None])
    edge_y.extend([y0, y1, None])

edge_trace = go.Scatter(
    x=edge_x, y=edge_y,
    line=dict(width=1, color='black'),
    hoverinfo='none',
    mode='lines')

node_x = []
node_y = []
for node in G.nodes():
    x, y = pos[node]
    node_x.append(x)
    node_y.append(y)

```

```

# En la generación de la visualización del grafo
node_trace = go.Scatter(
    x=node_x, y=node_y,
    mode='markers+text', # Agregar texto a los markers
    hoverinfo='text',
    text=list(G.nodes), # Usar los nombres de los nodos como texto
    textposition='bottom center', # Posición del texto
    marker=dict(
        showscale=False,
        colorscale='YlGnBu',
        reversescale=True,
        color=[],
        size=10,
        colorbar=dict(
            thickness=15,
            title='Node Connections',
            xanchor='left',
            titleside='right'
        ),
        line_width=2))

# Crear Scatter para las aristas
edge_trace = go.Scatter(
    x=edge_x, y=edge_y,
    line=dict(width=1, color='black'),
    hoverinfo='text', # Mostrar información al pasar el mouse

```

```
# Mostrar las distancias en el texto
text=[G[u][v]["weight"] for u, v in G.edges()],
mode='lines')

#pintar nodos que contenga la ruta mínima
# Crear una lista vacía para los colores de los nodos
node_colors = []

# Recorrer los nodos y asignarles un color
for node in G.nodes():
    x, y = pos[node]
    node_x.append(x)
    node_y.append(y)
    if node in ruta_mas_corta:
        node_colors.append('#FF0000') # Asignar color rojo a los nodos de la ruta mínima
    else:
        node_colors.append('#FFFFFF') # Asignar color blanco por defecto

# Asignar la lista de colores a la propiedad 'color' del 'node_trace'
node_trace['marker']['color'] = node_colors
```

```
# Crear Figure que contiene ambos Scatter
fig = go.Figure(data=[edge_trace, node_trace],
                layout=go.Layout(
                    title='Grafo',
                    showlegend=False,
                    hovermode='closest',
                    margin=dict(b=20, l=5, r=5, t=40),
                    xaxis=dict(showgrid=False, zeroline=False, showticklabels=False),
                    yaxis=dict(showgrid=False, zeroline=False, showticklabels=False)))

# Guardar la figura en un archivo HTML temporal
fig_path = "ProyectoWebApp/static/temp_graph.html"
fig.write_html(fig_path)
return {"fig_path": fig_path, "ruta_mas_corta":ruta_mas_corta,
        "distancias_entre_nodos":distancias_entre_nodos}
```

CODIGO PARA LA IMPLEMENTACIÓN DEL MODELO PREDICTIVO A PARTIR DE LA CONSTRUCCIÓN DE UN MODELO DE RED NEURONAL.

```
#!/-----REDES NEURONALES ARTIFICIALES-----/  
def predecir_fuente_extraccion(request):  
    # Inicializar las variables  
    resultado = None  
    etiqueta = None  
    fuentes_dict = []  
    nombres_fuentes = []  
    if request.method=='POST':  
        # Obtener los valores del formulario  
        dureza = request.POST.get('maquina')  
        indice_plasticidad = request.POST.get('ip')  
        cbr=request.POST.get('cbr')  
        print("Valores obtenidos del formulario:")  
        print("Dureza:", dureza)  
        print("Indice de plasticidad:", indice_plasticidad)  
        print("CBR:", cbr)  
  
        # Verificar que los campos no sean None antes de convertirlos a float  
        if dureza is not None and indice_plasticidad is not None and cbr is not None:  
            dureza = float(dureza)  
            indice_plasticidad = float(indice_plasticidad)  
            cbr = float(cbr)
```

```
# Cargar el modelo de red neuronal previamente entrenado  
ruta_modelo = Path(__file__).resolve().parent.parent / 'modelo_entrenado' / 'modelofinalf.keras'  
modelo = load_model(ruta_modelo)  
  
if modelo is None:  
    # Manejar el caso en que el modelo no se cargó correctamente  
    return {  
        'resultado_prediccion': None,  
        'etiqueta': "Error en carga del modelo",  
        'fuentes_cercanas': [],  
        'nombres_fuentes_cercanas': []  
    }  
  
# Preparar los datos para la predicción  
datos_prediccion = [[dureza, indice_plasticidad, cbr]]  
datos_prediccion = np.array(datos_prediccion)  
  
# Hacer la predicción  
prediccion = modelo.predict(datos_prediccion)  
  
# La predicción es un arreglo de un solo elemento, obtener el valor de la predicción  
resultado = prediccion[0][0]  
probabilidad = prediccion[0][0]  
  
# Establecer el umbral de probabilidad (0.0025 en este caso)  
umbral = 0.6
```

```
# Asignar etiqueta de clasificación binaria
if probabilidad < umbral:
    etiqueta = "cumple requisitos"
else:
    etiqueta = "no cumple requisitos"

# Obtener las fuentes de extracción que cumplen con los requisitos de entrenamiento
fuentes_cumplen_requisitos = dashboard.objects.filter(Dureza_maquina_angeles__lte=50,
                                                    IP__lte=9,
                                                    CBR__gte=15)

# Puedes convertir las fuentes obtenidas en un diccionario si lo deseas
fuentes_dict = [{'Dureza_maquina_angeles': fuente.Dureza_maquina_angeles,
                'IP': fuente.IP,
                'CBR': fuente.CBR} for fuente in fuentes_cumplen_requisitos]

# Obtener los nombres de las fuentes cercanas
nombres_fuentes = [fuente.nombre_fuente for fuente in fuentes_cumplen_requisitos]

# Retornar los valores como un diccionario
return {'resultado_prediccion': resultado,
        'etiqueta': etiqueta,
        'fuentes_cercanas': fuentes_dict,
        'nombres_fuentes_cercanas': nombres_fuentes,}
```

```
88         else:
89             # Manejar el caso en que los valores del formulario no son válidos
90             return {
91                 'resultado_prediccion': None,
92                 'etiqueta': "Valores del formulario no válidos",
93                 'fuentes_cercanas': [],
94                 'nombres_fuentes_cercanas': []
95             }
96     else:
97         # Retornar un diccionario con valores predeterminados o un mensaje de error
98         return {
99             'resultado_prediccion': None,
100            'etiqueta': "No se enviaron datos del formulario",
101            'fuentes_cercanas': [],
102            'nombres_fuentes_cercanas': [],
103        }
104 #/-----/
```