



UNIVERSIDAD  
CATÓLICA  
DE CUENCA

**UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

**UNIDAD ACADÉMICA DE INGENIERÍA, INDUSTRIA  
Y CONSTRUCCIÓN**

**CARRERA DE ELECTRICIDAD**

**PROYECCIÓN DE LA DEMANDA DE LOS VEHÍCULOS  
ELÉCTRICOS DEL CANTÓN CUENCA Y SU IMPACTO EN  
LA RED DE DISTRIBUCIÓN.**

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL  
TÍTULO DE INGENIERO ELÉCTRICO**

**AUTORES: ADRIAN PATRICIO CUMBE TAPIA, ALEX GUILLERMO  
LEMA LUCERO**

**DIRECTOR: ING. CARLOS MAURICIO ORELLANA UGUÑA, MGTR**

**CUENCA - ECUADOR**

**2025**

**DIOS, PATRIA CULTURA Y DESARROLLO**



**UNIVERSIDAD CATÓLICA DE CUENCA**

*Comunidad Educativa al Servicio del Pueblo*

**UNIDAD ACADÉMICA INGENIERÍA, INDUSTRIA Y  
CONSTRUCCIÓN**

**CARRERA DE ELECTRICIDAD**

PROYECCIÓN DE LA DEMANDA DE LOS VEHÍCULOS ELÉCTRICOS DEL CANTÓN  
CUENCA Y SU IMPACTO A LA RED DE DISTRIBUCIÓN

**PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE  
INGENIERO ELÉCTRICO**

**AUTOR: ADRIAN PATRICIO CUMBE TAPIA, ALEX GUILLERMO LEMA LUCERO**

**DIRECTOR: ING. CARLOS MAURICIO ORELLANA UGUÑA, MGTR**

**CUENCA - ECUADOR**

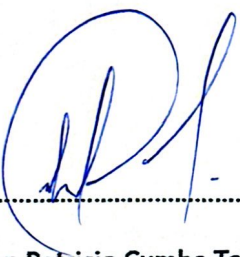
**2025**

**DIOS, PATRIA, CULTURA Y DESARROLLO**

## DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD


Adrian Patricio Cumbe Tapia y Alex Guillermo Lema Lucero portadore(a)s de las cédulas de ciudadanía N° 0150135937 y 0107825192. Declaramos ser autore(a)s de la obra: "Proyección de la demanda de los vehículos eléctricos en el cantón Cuenca y su impacto a la red de distribución", sobre la cual nos hacemos responsables sobre las opiniones, versiones e ideas expresadas. Declaramos que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximimos a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaramos finalmente que nuestra obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también nos responsabilizamos y eximimos a la Universidad Católica de Cuenca de toda reclamación al respecto.

Cuenca, 30 de abril de 2025

F: 

Adrian Patricio Cumbe Tapia

0150135937

F: 

Alex Guillermo Lema Lucero

01078825192

## CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Adrián Patricio Cumbe Tapia y Alex Guillermo Lema Lucero, bajo mi supervisión.



---

Ing. Carlos Mauricio Orellana Uguña, Mgtr  
DIRECTOR

## **DEDICATORIA**

Con todo mi corazón, dedico esta tesis a mis amados padres, Osvaldo Cumbe y Martha Tapia, quienes han sido el pilar más firme y constante en mi vida. Gracias por su amor incondicional, por cada sacrificio silencioso, por cada palabra de aliento y por estar siempre presentes, guiándome con su ejemplo, su fortaleza y sus valores. Han sido mi motor en los días difíciles y mi alegría en los logros alcanzados. Esta meta no sería posible sin su entrega, su esfuerzo y ese amor inmenso que me brindan día tras día.

También dedico este trabajo a mi querido hermano Javier, quien ha estado a mi lado en cada paso de este camino. Gracias por tu apoyo incondicional, por tu preocupación constante y por brindarme siempre una palabra de ánimo cuando más lo necesitaba. Tu presencia ha sido fundamental para llegar hasta aquí.

## DEDICATORIA

A Dios, fuente de mi fortaleza y guía en cada paso de este camino, le entrego este logro con gratitud infinita.

A ti, papito Julio Guillermo Lema Gordillo, por ser mi pilar, mi ejemplo y mi mayor inspiración. Tu esfuerzo y dedicación han sido la base sobre la que construyo mis sueños.

A ti, mamita Rosa Ximena Lucero Juela, por tu amor incondicional, por cada sacrificio silencioso y por enseñarme que la perseverancia y el cariño son las fuerzas más grandes para alcanzar cualquier meta.

A mi abuelita María Hortensia Gordillo, cuya sabiduría y ternura han dejado en mí huellas imborrables. Tu amor es un faro que ilumina mi camino, recordándome siempre mis raíces y valores.

A mi compañero y amigo de batalla, Adrián Patricio Cumbe Tapia, con quien compartí cada desafío, cada desvelo y cada momento de esfuerzo en esta travesía académica. Tu compromiso y apoyo han sido invaluable.

A nuestro tutor, Diego Morales, por su invaluable orientación, paciencia y sabiduría, que nos han guiado con firmeza y claridad en este proceso. Su apoyo incondicional ha sido clave para convertir este trabajo en una realidad.

A mi querida familia, que con su amor, fe y apoyo inquebrantable me han impulsado siempre a dar lo mejor de mí.

## **AGRADECIMIENTOS**

A mis padres y a mi hermano, por ser el cimiento sobre el cual he construido este camino. Su esfuerzo constante, su entrega silenciosa y su presencia firme han sido fundamentales para llegar hasta aquí. Nada de esto habría sido posible sin los sacrificios que han hecho a lo largo del tiempo, muchas veces dejando de lado sus propios anhelos para impulsarme a avanzar.

Más allá del apoyo académico, han sido quienes han moldeado mi carácter, mis principios y la forma en la que hoy veo la vida. Su amor, su guía y su ejemplo han sido la luz en cada paso que he dado. Por todo ello, y por mucho más, les estaré eternamente agradecido.

También quiero agradecer con profundo cariño a mis abuelos, tíos y a mi sobrino, quienes siempre han estado pendientes de mí tanto en mi formación académica como en mi vida personal. Gracias por acompañarme, por sus palabras de aliento, por su amor y por compartir conmigo tantas alegrías a lo largo de este camino. Su presencia ha sido un abrazo constante que me ha dado fuerza y motivación en cada etapa de este proceso.

A mi compañero y amigo Alex, por caminar junto a mí durante esta etapa con lealtad, dedicación y compromiso. Su apoyo ha sido clave en los momentos de mayor exigencia, y su compañerismo ha hecho que este proceso sea más llevadero y enriquecedor. Gracias por compartir el esfuerzo y también la convicción de llegar hasta el final.

Al Ingeniero Diego Morales y al Ingeniero Javier Cabrera, por su disposición, conocimiento y compromiso durante toda esta etapa de formación. Su guía ha sido mucho más que académica: ha estado marcada por la paciencia, la cercanía y una vocación genuina por enseñar. Agradezco el tiempo que dedicaron, las enseñanzas que compartieron y el trato que trascendió lo profesional para convertirse en apoyo personal. Han sido docentes, tutores, y también amigos en el proceso.

## AGRADECIMIENTOS

Primero, quiero agradecer a Dios, por ser mi guía en cada momento de incertidumbre y por darme la fuerza para seguir adelante en este recorrido. Su luz ha iluminado mi camino, y me ha brindado la serenidad y el impulso necesario en cada paso.

A ti, papito Julio Guillermo Lema Gordillo, por ser el hombre que ha definido mi sentido de responsabilidad y dedicación. Tu amor y sacrificio me han enseñado que el esfuerzo y la constancia son la clave para lograr todo lo que uno se propone.

A ti, mamita Rosa Ximena Lucero Juela, por tu ternura, tus enseñanzas y tu constante apoyo, que siempre ha sido el motor para mis éxitos. Gracias por cada sacrificio que has hecho y por estar siempre a mi lado, sin importar la distancia.

A mi abuelita María Hortensia Gordillo, quien siempre me ha brindado su cariño y sabiduría. Gracias por ser una fuente de inspiración constante, por enseñarme la importancia de la humildad y el amor en todo lo que hago.

A Adrián Patricio Cumbe Tapia, mi compañero de lucha, por su apoyo, amistad y dedicación en este proyecto. Hemos superado juntos los retos más grandes, y tu compromiso ha sido fundamental en este logro.

A mi tutor, Ing. Diego Morales, por su apoyo constante, por haber creído en mí desde el primer día y por ser no solo un mentor académico, sino también una gran fuente de inspiración personal. Su sabiduría y paciencia han sido cruciales para el desarrollo de esta tesis.

A todas las personas que han llegado a mi vida a lo largo de este camino, por sus palabras de aliento, consejos y por aportar un poco de su luz en mi proceso. Cada uno de ustedes ha dejado una huella en mi desarrollo académico y personal.

Y, por supuesto, a mi familia y amigos, que siempre han estado a mi lado con su amor, paciencia y confianza. Este logro también es suyo, porque sin su apoyo constante, no hubiera sido posible.

## RESUMEN

Este trabajo presenta una metodología para proyectar la demanda de vehículos eléctricos en el cantón Cuenca y evaluar su impacto en la red de distribución de la Empresa Eléctrica Regional Centro Sur (CENTROSUR). La estimación de la demanda se realiza para el período 2024–2034 mediante la aplicación del modelo de Bass, que permite simular el crecimiento tecnológico y la adopción de los vehículos eléctricos en el entorno urbano. La distribución de los vehículos proyectados se asigna proporcionalmente a la población de cada parroquia, buscando reflejar de forma coherente su presencia en el territorio. El análisis técnico se centra en el alimentador 0324, modelado en el software OpenDSS, considerando cuatro escenarios de evaluación: estado actual, proyección al 2028, al 2033 y un límite máximo de capacidad del sistema. Para cada escenario se analizan parámetros eléctricos relevantes, como perfiles de potencia y tensión, con el fin de identificar posibles restricciones operativas que pudieran surgir debido al incremento de la carga. Los resultados obtenidos permiten establecer umbrales críticos de carga y ofrecen criterios técnicos para una planificación de la red más sólida y anticipada frente al avance de la movilidad eléctrica en Cuenca. Esta investigación contribuye a fortalecer el proceso de transición energética en la ciudad, proponiendo soluciones orientadas a garantizar la calidad y continuidad del servicio eléctrico.

**Palabras clave:** vehículos eléctricos, modelo de Bass, OpenDSS, Python.

## ABSTRACT

This paper presents a methodology for projecting the demand for electric vehicles in the city of Cuenca and evaluating their impact on the distribution network of the Regional Electric Company of the Southern Center (CENTROSUR, by its Spanish acronym). Demand estimation is carried out for the 2024–2034 period using the Bass model, which allows the simulation of technological growth and the adoption of electric vehicles in the urban environment. The projected vehicle distribution is allocated proportionally to the population of each parish, aiming to consistently reflect their territorial presence. The technical analysis focuses on feeder 0324, modeled using OpenDSS software, and considers four evaluation scenarios: the current state, projections for 2028 and 2033, and the system's maximum load capacity. For each scenario, relevant electrical parameters—such as power and voltage profiles—are analyzed to identify potential operational constraints that may arise due to the increase in load. The results obtained allow for establishing critical load thresholds and provide technical criteria for more robust and forward-looking network planning in response to the advancement of electric mobility in Cuenca. This research strengthens the city's energy transition process, proposing solutions to guarantee the quality and continuity of the electric service.

*Keywords:* electric vehicles, Bass model, OpenDSS, Python.

## ÍNDICE DE CONTENIDO

### Tabla de contenido

DECLARATORIA DE AUTORÍA Y RESPONSABILIDAD .....	iii
CERTIFICACIÓN.....	iv
DEDICATORIA .....	v
DEDICATORIA .....	vi
AGRADECIMIENTOS.....	vii
AGRADECIMIENTOS.....	viii
RESUMEN.....	ix
ABSTRACT .....	x
CAPITULO 1 .....	17
1.1 Introducción .....	17
1.2 OBJETIVO General .....	18
1.3 OBJETIVOS específicos .....	18
1.4 ALCANCE.....	18
1.4 JUSTIFICACIÓN.....	18
1.5 METODOLOGÍA.....	19
1.6 ESTADO DEL ARTE .....	19
CAPITULO 2 MARCO TEÓRICO.....	22
2.1 Movilidad eléctrica .....	22
2.2 METODOS DE PROYECCION DE DATOS .....	22
2.3 SECTOR ELÉCTRICO ECUATORIANO.....	24
2.4 Resolución Codificada Nro. Arcernnr 002/20.....	25
2.4 LEY ORGÁNICA DE EFICIENCIA ENÉRGICA .....	26
2.5 LEY ORGÁNICA de transporte terrestre tránsito y seguridad vial.....	26
2.6 Resolución ARCONEL-024/2024 .....	27
2.7 Pliego Tarifario 2025 para los Proveedores del Servicio de Carga de Energía a Vehículos Eléctricos .....	27
2.8 ORDENANZA MUNICIPAL Y PLAN DE ELECTROMOVILIDAD DE LA CIUDAD DE CUENCA .....	28
2.9 CARACTERISTICAS SOCIOECONÓMICAS DE LOS POTENCIALES USUARIOS DE VE. ....	29
2.10 EMPRESA ELÉCTRICA REGIONAL CENTROSUR .....	29
2.11 ARC GIS.....	31
2.12 Python .....	32
2.13 OPEN DSS .....	32

CAPITULO 3 DESARROLLO .....	36
3.1 RECOPIACIÓN DE DATOS .....	36
3.2 MODELO DE BASS EN PYHTON.....	38
3.3 Parroquias rurales con mayor crecimiento de VE .....	53
3.3 selección DEL ALIMENTADOR.....	56
3.4 MODELADO DEL ALIMENTADOR .....	57
3.5 Características de la carga.....	59
3.6 Análisis de resultados.....	59
3.7 CONCLUSIONES .....	62
3.8 RECOMENDACIONES .....	64
REFERENCIAS BIBLIOGRÁFICAS .....	66
Bibliografía .....	66
ANEXOS .....	70
Anexo 1 proyección de ve bellavista .....	70
Anexo 2 creacion de grafo.....	72
Anexo 3 creación de archivos .dss .....	79
Anexo 4 simulación .....	103
AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL .....	114

## INDÍCE DE FIGURAS

Figura 1: Curva sigmoïdal. ....	24
Figura 2: Mapa del área de servicio. ....	30
Figura 3: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Baños. ....	41
Figura 4: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Bellavista. ....	41
Figura 5: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Cañaribamba. ....	41
Figura 6: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Checa. ....	42
Figura 7: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Chiquintad. ....	42
Figura 8: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Cumbe. ....	42
Figura 9: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia El Batan. ....	43
Figura 10: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia El Sagrario. ....	43
Figura 11: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia El Valle. ....	43
Figura 12: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia El Vecino. ....	44
Figura 13: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Hermano Miguel. ....	44
Figura 14: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Huayna Cápac. ....	44
Figura 15: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Llaçao. ....	45
Figura 16: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Machángara. ....	45
Figura 17: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Molleturo. ....	45
Figura 18: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Monay. ....	46
Figura 19: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Nulti. ....	46
Figura 20: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Octavio Cordero Palacios. ....	46
Figura 21: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Paccha. ....	47
Figura 22: Curva sigmoïdal del modelo de Bass aplicada a las adopciones de VE de la parroquia Quingeo. ....	47

Figura 23: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Ramírez Dávalos. ....	47
Figura 24: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Ricaurte. ....	48
Figura 25: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia San Blas. ....	48
Figura 26: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia San Joaquín. ....	48
Figura 27: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia San Sebastián. ....	49
Figura 28: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Santa Ana. ....	49
Figura 29: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Sayausí. ....	49
Figura 30: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Sidcay. ....	50
Figura 31: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Sinincay. ....	50
Figura 32: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Sucre. ....	50
Figura 33: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Tarqui. ....	51
Figura 34: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Totoracocha. ....	51
Figura 35: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Turi. ....	51
Figura 36: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Victoria del Portete. ....	52
Figura 37: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Yanuncay. ....	52
Figura 38: Representación polinómica de tercer grado como aproximación visual del comportamiento proyectado por el modelo de Bass en la parroquia Baños. ....	54
Figura 39: Representación polinómica de tercer grado como aproximación visual del comportamiento proyectado por el modelo de Bass en la parroquia Ricaurte. ....	54
Figura 40: Representación polinómica de tercer grado como aproximación visual del comportamiento proyectado por el modelo de Bass en la parroquia El Valle. ....	55
Figura 41: Representación polinómica de tercer grado como aproximación visual del comportamiento proyectado por el modelo de Bass en la parroquia Sinincay. ....	55
Figura 42: Representación polinómica de tercer grado como aproximación visual del comportamiento proyectado por el modelo de Bass en la parroquia Sayausí. ....	55
Figura 43: Alimentador 0324 Centro sur. ....	57
Figura 44: Creación de la red desde Python. ....	58
Figura 45: Carga activa de los VE. ....	60
Figura 46: Distribución de tensiones por escenarios. ....	60
Figura 47: Factor de potencias s. ....	61

Figura 48: Evolución anual de la carga activa y reactiva..... 62

## TABLA DE CONTENIDOS

Tabla 1: Principales modelos de difusión de tecnologías. ....	23
Tabla 2: Niveles de voltaje Fuente: .....	25
Tabla 3: Alimentadores pertenecientes a CENTROSUR. ....	30
Tabla 4: Parámetros de líneas de distribución. Fuente: (Palacios, 2020) .....	33
Tabla 5: Parámetros de transformadores de distribución. Fuente: (Palacios, 2020) .....	34
Tabla 6: Parámetros de cargas de distribución. ....	34
Tabla 7: Venta de vehículos eléctricos anuales y acumulados. ....	36
Tabla 8: Distribución proporcional de Vehículos eléctricos para el Cantón Cuenca.....	37
Tabla 9: Distribución proporcional de Vehículos eléctricos para las parroquias de Cuenca. }.....	38
Tabla 10: Parámetros $p$ , $q$ , y $m$ del modelo de Bass para la proyección de vehículos eléctricos en Cuenca.....	40
Tabla 11: Resultados de las proyecciones de VE en las parroquias de Cuenca. ....	53
Tabla 12: Resultados de las proyecciones de VE en las parroquias de Cuenca. ....	53
Tabla 13: Proyección de VE en parroquias abastecidas por el alimentador 0104. ....	56
Tabla 14: Parroquias que abarca el alimentador 0324. ....	56

## CAPITULO 1

### 1.1 INTRODUCCIÓN

Durante los últimos años, el calentamiento global y el cambio climático ha cobrado gran relevancia en el debate público, debido a sus efectos perjudiciales tanto sobre los entornos naturales como sobre la sociedad en general (Albuquerque et al., 2020). Dentro de los sectores emisores de gases de efecto invernadero, el transporte ocupa un lugar destacado, al ser responsable de cerca del 20% del dióxido de carbono (CO<sub>2</sub>) liberado globalmente. En este contexto, el transporte terrestre se posiciona como la fuente predominante, especialmente en las ciudades, donde la congestión vehicular reduce la eficiencia energética debido a velocidades irregulares y ciclos frecuentes de aceleración y frenado. (Morejón-Monteros et al., 2024).

La movilidad eléctrica se consolida como una alternativa sostenible para reducir la contaminación generada por el transporte convencional, al tiempo que permite incorporar fuentes de energía renovable al sector y disminuir su consumo energético (Perdomo et al., 2022). En la actualidad, los vehículos eléctricos representan una opción eficiente para promover una movilidad limpia, ya que el uso de tracción eléctrica en los automóviles contribuye significativamente a la reducción de la contaminación ambiental, en especial a la atmosférica y acústica (Martínez Gutiérrez, 2013).

No obstante, la adopción de estos valiosos medios de transporte ha avanzado con mayor rapidez que el desarrollo de la infraestructura pública necesaria para su funcionamiento, lo que ha llevado a que muchos propietarios implementen de manera improvisada sistemas de carga en sus hogares, sin tener pleno conocimiento sobre su comportamiento ni sobre el impacto que pueden generar en el suministro eléctrico (Herrera Basabe et al., 2024). Estos efectos pueden manifestarse en distintos problemas dentro del sistema eléctrico, tales como desviaciones de tensión, sobrecargas térmicas, desequilibrios en el voltaje, fluctuaciones en la demanda y alteraciones en las corrientes de cortocircuito que afectan el sistema de protección. A esto se suma el crecimiento acelerado no solo de los vehículos eléctricos en las redes de distribución, sino también de los sistemas fotovoltaicos residenciales (Karmaker et al., 2024).

La transición hacia un futuro basado en tecnologías de bajas emisiones de carbono tendrá un impacto significativo en las redes de distribución, generando modificaciones en la demanda y en los perfiles de carga, lo que planteará nuevos desafíos para el sistema de distribución eléctrica (Cardenas Cobos y Jaramillo León, 2017). Según proyecciones de la Agencia Internacional de la Energía, para el año 2040 se prevé que la electricidad será la fuente de energía final con mayor crecimiento dentro de la matriz energética. Al mismo tiempo, el sector eléctrico se perfila como el principal responsable de disminuir la presencia de combustibles fósiles en el mix energético mundial (Batalla Bejerano y Jové Llopis, 2019).

## **1.2 OBJETIVO GENERAL**

Modelar el crecimiento de la demanda energética de vehículos eléctricos (VE) en la ciudad de Cuenca hasta el año 2034, mediante simulaciones determinando su impacto en la red de distribución.

## **1.3 OBJETIVOS ESPECÍFICOS**

- Analizar el número de VE actuales en uso en Cuenca y las políticas y programas locales identificando patrones de adopción y evaluando las políticas y estrategias públicas para el impulso de la electromovilidad en la ciudad.
- Desarrollar un modelo de crecimiento basado en la recopilación y análisis de datos del periodo 2014-2024 que refleje la realidad del entorno.
- Evaluar la capacidad actual de la red de distribución en el cantón, identificando puntos críticos en donde la cargabilidad se encuentre sobre niveles de alarma (software OPEN DSS y ARCMAP), y proponiendo medidas y soluciones basadas en los resultados del análisis para adaptar la red a las nuevas demandas.

## **1.4 ALCANCE**

Esta investigación se enfoca en la proyección de la demanda energética asociada a VE vehículos eléctricos privados en la ciudad de Cuenca, utilizando el modelo de Bass para estimar su crecimiento por parroquias hasta el año 2034. Con base en los resultados obtenidos, se selecciona un alimentador de media tensión perteneciente a la red de distribución de la Empresa Eléctrica Regional Centro Sur C.A. (EERCS), correspondiente a una de las zonas con mayor concentración proyectada de vehículos eléctricos. A dicho alimentador se le incorpora la carga adicional estimada, con el fin de analizar su comportamiento ante el incremento de la demanda.

## **1.4 JUSTIFICACIÓN**

Los sistemas eléctricos atraviesan actualmente una profunda transformación impulsada por diversos factores, entre los que destacan la descarbonización, la reducción de costos, la descentralización de la generación, la incorporación de tecnologías de almacenamiento, la necesidad de mayor flexibilidad operativa y el avance de la digitalización (Batalla Bejerano y Jové Llopis, 2019).

En este escenario, la movilidad eléctrica emerge como una estrategia clave para reducir las emisiones contaminantes y promover un modelo de transporte limpio y eficiente. El VE se ha convertido, por tanto, en un componente esencial dentro de las políticas de descarbonización tanto a nivel global como local (Frías y Jaime, 2023). No obstante, en el contexto ecuatoriano, la adopción de esta tecnología aún se encuentra en una fase incipiente, limitada por barreras informativas, percepciones de riesgo e incertidumbre tecnológica (Reyes Campaña et al., 2021).

El crecimiento sostenido que se proyecta para el parque vehicular eléctrico en los próximos años plantea importantes retos para el sistema eléctrico, particularmente en las redes de distribución. La incorporación masiva de estos vehículos podría generar un incremento considerable en la demanda energética, afectando los perfiles de tensión, provocando sobrecargas y aumentando las pérdidas técnicas si no se implementan estrategias de planificación adecuadas. Por, lo que resulta fundamental anticipar estos escenarios mediante modelos de proyección y simulación, que permitan evaluar el comportamiento del sistema

eléctrico frente al crecimiento de la demanda generada por vehículos eléctricos, asegurando así una integración eficiente y sostenible en la infraestructura actual.

## **1.5 METODOLOGÍA**

La investigación se desarrolla en tres fases. En la primera etapa, se realiza un análisis de los datos históricos sobre la venta de VE en Ecuador durante el período 2014-2023, utilizando como fuente principal la información proporcionada por la Asociación de Empresas Automotrices del Ecuador (AEADE). Este estudio se complementa con datos provenientes del Instituto Nacional de Estadística y Censos (INEC), la Agencia Nacional de Tránsito (ANT) y la clasificación de los vehículos. Asimismo, se examinan las políticas y programas implementados a nivel local para fomentar la electromovilidad en la ciudad de Cuenca. Para fomentar la electromovilidad en la ciudad de Cuenca, estableciendo así una base sólida para aplicar el modelo de Bass en la proyección del crecimiento anual por parroquia.

En la segunda fase, se identifican las zonas con mayor concentración estimada de VE. Con base en esta información, se elige un alimentador de la red de distribución correspondiente al sector de mayor demanda proyectada, considerando aspectos técnicos como su ubicación geográfica, nivel de tensión y densidad de vehículos por parroquia. El alimentador seleccionado se somete a un proceso de modelado con el objetivo de evaluar el efecto del incremento de carga de los VE sobre la red eléctrica. Para ello, se emplea la base geoespacial (geodatabase) proporcionada por la Empresa Eléctrica Regional Centro Sur (EERCS), la cual se integra en el entorno ArcGIS, procesándose mediante scripts en Python para generar archivos compatibles con la plataforma de simulación eléctrica OpenDSS. En esta etapa se desarrolla la simulación progresiva de la carga asociada a los VE de acuerdo con los resultados de la proyección por parroquias.

Finalmente, en la tercera fase, se analiza el comportamiento del alimentador modelado bajo diferentes cantidades de VE conectados. El estudio evalúa el impacto en la red de distribución considerando aspectos técnicos como: niveles de tensión (en el lado de baja tensión), potencia, factor de potencia y determinación del número de VE y año en el cual la red de distribución se vea afectado por la incorporación de los VE.

## **1.6 ESTADO DEL ARTE**

En los últimos años, el avance de la electromovilidad en Ecuador ha sido analizado en distintas investigaciones, centradas principalmente en la incorporación de vehículos eléctricos (VE) en las redes de distribución. Estos trabajos han explorado temas como la capacidad de integración, los efectos sobre la infraestructura eléctrica y la ubicación estratégica de las estaciones de carga. Un estudio destacado en este campo es el realizado por (Cárdenas Bravo y Machuca Chabla, 2024) quienes proponen una metodología para estimar la capacidad de alojamiento (HC) de recursos energéticos distribuidos (DER), enfocándose específicamente en vehículos eléctricos y sistemas fotovoltaicos (SFV) dentro de las redes de media y baja tensión de la Empresa Eléctrica Regional Centro Sur C.A. (EERCS). Esta investigación se distingue por su enfoque detallado, que incluye la corrección de errores en la representación del Sistema de Información Geográfica (SIG), la formulación de distintos escenarios de penetración tecnológica y la definición de los límites operativos de los parámetros eléctricos conforme a la normativa técnica de la EERCS. Además, se realizan simulaciones para calcular la HC, cuyos resultados se presentan mediante diagramas de caja, facilitando la visualización de los valores máximos, mínimos, promedios y

atípicos de los parámetros eléctricos evaluados. Dichos hallazgos resultan fundamentales para delimitar niveles de penetración seguros de VE y SFV en la red eléctrica de Cuenca.

Por su parte, (Villegas Aldeán y Trelles Lituma, 2023) desarrollan un estudio similar, en el que plantean una metodología iterativa basada en series temporales y simulaciones mediante el software CYMDIST para analizar la Capacidad de Alojamiento Fotovoltaica (PVHC). Su enfoque resalta la relevancia de establecer criterios técnicos específicos que definan los límites operativos para la incorporación de sistemas fotovoltaicos en la red eléctrica, considerando aspectos como las restricciones de voltaje, la capacidad térmica de los conductores, el comportamiento del flujo de potencia y la generación reactiva en la red. El estudio también contempla distintos escenarios de penetración de generación distribuida, incorporando la variabilidad de la irradiancia solar y de la demanda eléctrica. Adicionalmente, se emplea un modelo probabilístico que complementa los análisis deterministas, mejorando la precisión de los resultados. Esta metodología integral permite examinar el comportamiento de la red ante la inyección fluctuante de energía solar, proporcionando una evaluación más detallada sobre el impacto que puede tener la presencia de SFV en el desempeño y estabilidad de las redes de distribución.

La incorporación progresiva de sistemas de generación distribuida mediante tecnología solar fotovoltaica, conectados en sincronía con las redes de distribución, ha representado un desafío considerable para las empresas distribuidoras, tanto en el ámbito operativo como en el de la planificación del sistema eléctrico. De acuerdo con el estudio de (Patiño Chitacapa, 2023), una elevada integración de generación distribuida puede afectar negativamente el desempeño del sistema eléctrico, lo que hace indispensable estimar la capacidad de alojamiento para definir límites seguros de incorporación. La metodología propuesta en esta investigación analiza distintos escenarios de adopción de generación distribuida por parte de los consumidores, utilizando simulaciones de cortocircuitos y flujos de potencia mediante series temporales. Este enfoque considera tanto las restricciones operativas y regulatorias como el estado actual del sistema y de los equipos involucrados. Además, se emplea información real y de alta resolución sobre los perfiles de carga en los nodos fuente, los niveles de tensión registrados por sensores de calidad, los esquemas de control implementados y las características de la demanda según el tipo de cliente. En este modelo, la capacidad del generador se dimensiona individualmente para cada usuario, tomando como referencia su autoconsumo y conforme a la normativa nacional vigente. Finalmente, la capacidad de alojamiento de un alimentador en redes de media y baja tensión se determina con base en los criterios de impacto establecidos en los estándares técnicos y marcos regulatorios, garantizando que la integración de generación distribuida no represente riesgos para la operación ni comprometa la calidad y confiabilidad del suministro eléctrico.

Los estudios revisados evidencian que la integración de VE y generación distribuida plantea nuevos desafíos técnicos para las redes de distribución. La mayoría de las investigaciones se orientan al análisis del comportamiento eléctrico del sistema bajo diferentes escenarios de carga, tomando en cuenta condiciones operativas reales y las restricciones establecidas por la normativa vigente. Estos antecedentes permiten entender la variación de parámetros como el perfil de voltaje, las pérdidas técnicas, el flujo de potencia y el nivel de utilización de la capacidad

instalada, variables que también son consideradas en el presente estudio enfocado en el sistema de distribución eléctrica de Cuenca.

## **CAPITULO 2 MARCO TEÓRICO**

### **2.1 MOVILIDAD ELÉCTRICA**

Los medios de transporte eléctricos abarcan una amplia gama de vehículos que funcionan mediante energía eléctrica, desde opciones de micro movilidad como Scooter y bicicletas eléctricas con pedaleo asistido, hasta vehículos motorizados de dos y tres ruedas, como calesas y bicicletas de carga eléctricas. También se incluyen vehículos ultralivianos de cuatro ruedas, como automóviles y furgonetas, así como unidades de mayor tamaño como camiones, autobuses, trolebuses, tranvías, trenes, aviones y transbordadores (Movilidad Sostenible para Todos, 2021).

Desde el punto de vista ambiental, el VE se distingue frente a los automóviles de combustión interna por no generar emisiones durante su operación, lo que contribuye a evitar la liberación de gases contaminantes en entornos urbanos. Esta característica favorece una mejora en la calidad del aire en las ciudades y, en consecuencia, en la salud de sus habitantes (Vélez Sanchez, 2017).

La incorporación de esta tecnología conlleva diversas ventajas, entre ellas la reducción de las emisiones de óxidos de nitrógeno, un menor nivel de ruido debido a que el motor del vehículo eléctrico es considerablemente más silencioso que el de un automóvil convencional, y un alto nivel de eficiencia, ya que el rendimiento de los motores eléctricos alcanza aproximadamente el 95% (Ramos Sarango, 2024).

### **2.2 METODOS DE PROYECCION DE DATOS**

En la actualidad, se destinan esfuerzos constantes al desarrollo de productos y servicios innovadores. El proceso mediante el cual una tecnología novedosa se expande dentro del mercado se conoce como difusión. Este fenómeno, que abarca desde la creación de una idea hasta su adopción por los usuarios finales, constituye un campo de estudio relevante en el entorno económico contemporáneo. Diversos modelos han sido propuestos en la literatura para representar este proceso (ver Tabla 8); sin embargo, muchos de ellos no consideran un elemento crucial en la difusión de nuevas tecnologías: los efectos de red. Estos efectos se refieren al incremento del valor percibido de un producto o servicio en función del número de personas que lo utilizan, lo que resulta determinante en la adopción de innovaciones tecnológicas (Sánchez Loor, 2022).

Tabla 1: Principales modelos de difusión de tecnologías.

Fuente: (Sánchez Loor, 2022)

Modelo	Ecuación	
Gompertz	$\frac{dx(t)}{dt} = \beta * x(t) * \ln \frac{1}{x(t)}$	$x(t) = e^{-e^{(-\beta * t + k)}}$
Gaussiano	$\frac{dx(t)}{dt} = \frac{1}{\sqrt{2\pi\sigma^2}} * e^{-\frac{(t-\mu)^2}{2\sigma^2}}$	
Weibull	$\frac{dx(t)}{dt} = \frac{\beta}{\alpha} * \left(\frac{t}{\alpha}\right)^{\beta-1} e^{-\left(\frac{t}{\alpha}\right)^\beta}$	$x(t) = 1 - e^{-\left(\frac{t}{\alpha}\right)^\beta}$
Logístico	$\frac{dx(t)}{dt} = \beta * x(t) * (1 - x(t))$	$x(t) = \frac{1}{[1 + e^{(-\beta * (t - t_0))}]}$
Bass	$\frac{dx(t)}{dt} = q * x(t) \left(1 - \frac{x(t)}{M}\right) + p(M - x(t))$	$x(t) = M \frac{1 - e^{-(p+q)*t}}{1 + \left(\frac{q}{p}\right) * e^{-(p+q)*t}}$

El modelo de Gompertz a pesar de su utilidad, el modelo presenta ciertas limitaciones inherentes a su naturaleza determinista, lo que puede reducir su precisión al representar fenómenos con alta variabilidad estocástica. Además, la captura de correlaciones en datos secuenciales exige modificaciones estructurales específicas, aumentando no solo la complejidad del modelo, sino también los recursos computacionales requeridos para su implementación y validación (Batún et al., 2023) (Batún et al., 2023). Mientras que en el modelo Weibull, lo que requiere experiencia técnica avanzada. Su análisis puede ser limitado si los datos son insuficientes o no representativos del sistema evaluado. Por último, el modelo logístico puede ser menos flexible para modelar fenómenos con múltiples fases de crecimiento o cambios abruptos en la tasa de adopción, requiere parámetros bien definidos lo que puede ser difícil con datos limitados (Weissmann, 2008).

Para la estimación se utiliza el método de Bass que se basa en mínimos cuadrados no lineales, que busca minimizar la suma de los errores al cuadrado entre los valores reales y los valores generados por el modelo. Según (Marumo, Okuno, & Akiko, 2023), este algoritmo permite alcanzar una solución con menor error en un menor número de iteraciones en comparación con métodos tradicionales, logrando una reducción del error de ajuste que puede superar el 30% al 50% en escenarios prácticos.

Matemáticamente el modelo de Bass se define de la siguiente manera:

$$S(t) = \left[ p + \left( \frac{q}{m} \right) N(t - 1) \right] [m - N(t - 1)]$$

donde:  $N(t)$ , es el número de consumidores que adoptaron el producto en el momento  $t$ ,  $S(t)$ , es el número de nuevos consumidores que adoptaron el producto en el periodo  $t$ ,  $m$  representa el número máximo de consumidores potenciales que podrían adquirir el producto,  $p$  coeficiente de innovación, que indica la probabilidad de que un innovador adquiera o adopte el producto durante el período  $t$  y  $q$  coeficiente de imitación, representa la probabilidad de que un imitador adopte el nuevo producto. Este parámetro refleja el efecto del boca a boca, es decir, la influencia

que ejercen los innovadores sobre los imitadores a través de la comunicación e interacción social, fenómeno también conocido como efecto contagio (Weissmann, 2008).

A partir de los parámetros  $m$ ,  $p$  y  $q$ , para el trabajo de investigación se define los siguientes parámetros:

- Clientes potenciales ( $p$ ): se refiere a aquellas personas que pertenecen al mercado objetivo del producto, aunque aún no han realizado su consumo.
- Clientes ( $q$ ): son los individuos que ya han adoptado y consumen el producto ofrecido por la empresa.
- Tamaño del mercado ( $m$ ): hace referencia al total de personas que conforman el mercado, incluyendo tanto a los clientes de la empresa en estudio como a los de sus competidores, así como a aquellos que aún no adquieren productos de ninguna empresa. Este parámetro se considera constante durante todo el periodo de análisis del modelo. A medida que el tamaño del mercado aumenta, la tasa de adopción de la empresa tiende a disminuir, ya que una mayor dimensión del mercado implica una menor participación relativa y, por ende, una mayor dificultad para alcanzar a los clientes potenciales.
- $N(t-1)$ : representa la cantidad de VE que pueden ser incorporados en el sistema en el año  $t - 1$ .

Gráficamente la curva del modelo Bass, se indica en la Figura 1 (Weissmann, 2008).

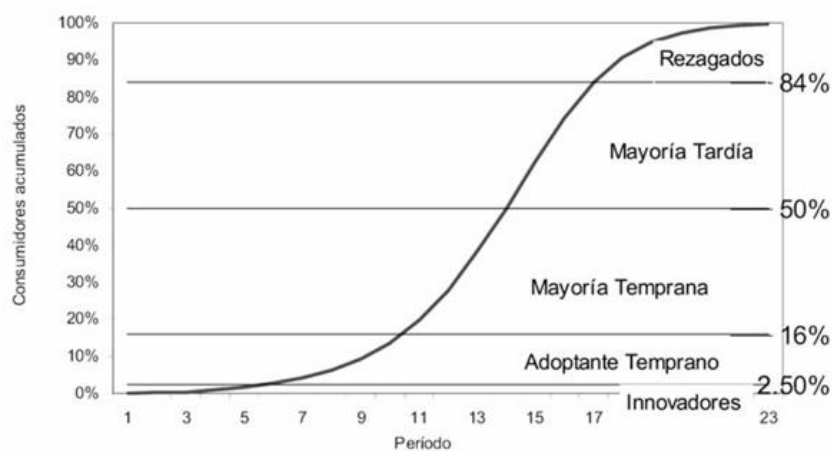


Figura 1: Curva sigmoideal.

Fuente: (Weissmann, 2008).

### 2.3 SECTOR ELÉCTRICO ECUATORIANO

La consolidación de la movilidad eléctrica en Ecuador está estrechamente vinculada con la evolución del sistema energético nacional. El Atlas del Sector Eléctrico Ecuatoriano 2023, publicado por la Agencia de Regulación y Control de Energía y Recursos Naturales No Renovables (ARCERNNR), proporciona un diagnóstico técnico y geográfico de la infraestructura eléctrica del país, el cual resulta clave para entender la viabilidad y sostenibilidad del transporte eléctrico (Agencia Nacional de Regulación y Control de Electricidad (ARCONEL), 2023).

Un elemento sobresaliente es la estructura de generación eléctrica, caracterizada por una creciente participación de fuentes renovables. En el año 2023, más del 61 % de la potencia instalada correspondió a energías limpias, con un claro predominio de la energía hidroeléctrica, que representó el 95 % de esa participación. Este dato adquiere particular relevancia al evidenciar que Ecuador posee las condiciones estructurales necesarias para que el servicio de carga de vehículos eléctricos se sustente principalmente en energía limpia, minimizando la huella de carbono asociada al transporte (Agencia Nacional de Regulación y Control de Electricidad (ARCONEL), 2023).

Las provincias de Azuay y Napo concentran las principales centrales hidroeléctricas del país, como Molino, Sopladora y Coca Codo Sinclair, las cuales no solo abastecen de manera eficiente la demanda energética nacional, sino que también aportan robustez y estabilidad al sistema interconectado, favoreciendo la expansión territorial de puntos de carga eléctrica. Otro aspecto relevante evidenciado en el Atlas es la existencia de sistemas aislados, que brindan energía a regiones remotas no integradas al sistema interconectado nacional (Agencia Nacional de Regulación y Control de Electricidad (ARCONEL), 2023).

No obstante, el documento también revela una dependencia considerable del consumo de combustibles fósiles en ciertas provincias, como Guayas, Orellana y Sucumbíos. En estas regiones, la generación térmica continúa siendo significativa, lo que reafirma la urgencia de acelerar la implementación del transporte eléctrico como una estrategia efectiva para disminuir el uso de derivados del petróleo y reducir las emisiones contaminantes (Agencia Nacional de Regulación y Control de Electricidad (ARCONEL), 2023).

## 2.4 RESOLUCIÓN CODIFICADA NRO. ARCERNNR 002/20

El control del nivel de voltaje es un aspecto clave dentro de la evaluación de la calidad del servicio eléctrico en las redes de distribución. De acuerdo con la normativa vigente emitida por la Agencia de Regulación y Control de Energía y Recursos Naturales No Renovables (ARCERNNR), se establecen lineamientos técnicos que permiten verificar si los valores de voltaje entregados a los usuarios se mantienen dentro de márgenes aceptables. La variación del voltaje se calcula como el porcentaje de diferencia entre el voltaje real medido en el punto de entrega y el voltaje nominal esperado. Para ello, se toman promedios de mediciones realizadas cada 3 segundos durante intervalos de 10 minutos, aplicando la siguiente fórmula (Agencia de Regulación y Control de Energía y Recursos Naturales No Renovables (ARCERNNR), 2023):

$$\Delta V_k = \frac{V_k - V_n}{V_n} \cdot 100$$

Donde  $V_k$  representa el voltaje promedio medido y  $V_n$  el valor nominal correspondiente.

Los valores permitidos de desviación del voltaje varían según el nivel de tensión:

*Tabla 2: Niveles de voltaje Fuente:*

Nivel de tensión	Rango Admisible
Alto Voltaje (AV)	±5.0%

Medio Voltaje (MV)	±6.0%
Bajo Voltaje (BV)	±8.0%

Para que una distribuidora cumpla con los estándares establecidos, al menos el 95% de los datos registrados durante un período mínimo de siete días consecutivos deben ubicarse dentro de los rangos admisibles en todas las fases del suministro eléctrico.

## 2.4 LEY ORGÁNICA DE EFICIENCIA ENÉRGICA

En 2019, la Asamblea Nacional aprobó un marco legal orientado a definir la normativa y el funcionamiento del Sistema Nacional de Eficiencia Energética (SNEE), con el objetivo de impulsar una matriz energética segura y sostenible en el país (Asamblea Nacional del Ecuador, 2019).

Del mismo modo, el artículo 14 contempla la implementación de un programa para el retiro gradual de vehículos convencionales, incluidos los destinados al transporte público, los cuales deberán ser sustituidos por unidades eléctricas a partir del año 2025. Además, se establece que, a partir del 2030, solo podrán integrarse al sistema de transporte urbano aquellos vehículos que operen con tecnología eléctrica (Asamblea Nacional del Ecuador, 2019).

Por otra parte, el artículo 22 destaca la incorporación de incentivos destinados a promover la eficiencia energética a escala nacional. En este contexto, se establece que los proyectos orientados a este objetivo podrán acceder a mecanismos de financiamiento con condiciones preferenciales. Asimismo, en el sector del transporte público, se prevén beneficios arancelarios para los componentes asociados a los vehículos eléctricos (Asamblea Nacional del Ecuador, 2019).

## 2.5 LEY ORGÁNICA DE TRANSPORTE TERRESTRE TRÁNSITO Y SEGURIDAD VIAL

La Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial ha sido reformada para incluir disposiciones que fomentan la adopción de vehículos eléctricos en Ecuador. Estas reformas, vigentes desde 2008, establecen incentivos normativos clave que favorecen la circulación, visibilidad e infraestructura para este tipo de vehículos. Entre estos cambios se incluyen distintos incentivos, entre los que se destacan los siguientes (Asamblea Nacional del Ecuador, 2014):

- Se dispone que los VE quedarán exentos de las restricciones aplicadas a la circulación vehicular, sin importar el tipo de regulación vigente, ya sea de carácter local o nacional (Asamblea Nacional del Ecuador, 2014).
- Asimismo, se establece que los centros comerciales y las entidades públicas que proporcionen servicio de estacionamiento deberán asignar, como mínimo, el 2 % del total de espacios disponibles para el uso exclusivo de vehículos eléctricos, además de aplicar una tarifa preferencial destinada a estos usuarios (Asamblea Nacional del Ecuador, 2014).
- Se estipula que las placas de identificación vehicular deberán incorporar un distintivo especial que facilite la identificación de los VE (Asamblea Nacional del Ecuador, 2014).

## **2.6 RESOLUCIÓN ARCONEL-024/2024**

En el marco de la transición energética impulsada por el Estado ecuatoriano, la movilidad eléctrica se ha consolidado como una prioridad estratégica dentro de la política pública energética. La Resolución Nro. ARCONEL-024/2024, aprobada por el Directorio de la Agencia de Regulación y Control de Electricidad (ARCONEL), representa un instrumento normativo fundamental que establece el régimen tarifario aplicable al servicio de carga de vehículos eléctricos para el período 2025 (Agencia de Regulación y Control de Electricidad (ARCONEL), 2024).

Este régimen se sustenta en un enfoque técnico-económico desarrollado por la Dirección Técnica de Regulación Económica y Tarifas del Sector Eléctrico, específicamente en el Informe INF.DTRET.2024.092, que proporciona las bases metodológicas para la fijación de los límites máximos de costo por kWh de energía cargada. Dichos valores tarifarios han sido calculados con criterios de eficiencia, sostenibilidad y equidad tarifaria, procurando mantener una accesibilidad económica adecuada para los usuarios sin comprometer la viabilidad del servicio (Agencia de Regulación y Control de Electricidad (ARCONEL), 2024).

La Resolución establece que tanto empresas eléctricas distribuidoras como prestadores privados deben sujetarse estrictamente a los valores tarifarios definidos, diferenciados por tipo de vehículo (liviano o pesado), tipo de carga (semi-rápida, rápida, ultra-rápida) y tipo de corriente (AC o DC). Esta estructura tarifaria permite una mayor adaptación a las necesidades del mercado, ofreciendo incentivos diferenciados según el perfil del usuario y el tipo de infraestructura instalada (Agencia de Regulación y Control de Electricidad (ARCONEL), 2024). Esta normativa contribuye significativamente al desarrollo ordenado del ecosistema de carga en el país, consolidando un entorno regulatorio claro que estimula la inversión privada y facilita el crecimiento sostenido del parque vehicular eléctrico.

## **2.7 PLIEGO TARIFARIO 2025 PARA LOS PROVEEDORES DEL SERVICIO DE CARGA DE ENERGÍA A VEHÍCULOS ELÉCTRICOS**

Como parte de la consolidación del marco normativo para el desarrollo del transporte eléctrico en Ecuador, el Pliego Tarifario 2025, aprobado mediante Resolución Nro. ARCONEL-024/2024, representa un paso técnico clave para garantizar una transición energética eficiente y sostenible. Este documento establece los límites máximos del costo del servicio de carga de energía para vehículos eléctricos, diferenciando criterios tarifarios según tipo de vehículo, nivel de carga y tecnología empleada (Agencia de Regulación y Control de Electricidad (ARCONEL), 2024).

La estructura tarifaria definida se sustenta en estándares internacionales como la norma IEC 61851-1 y SAE J1772, lo cual asegura una alineación técnica con las mejores prácticas globales. Asimismo, se adopta una clasificación técnica que excluye los modos y niveles de carga considerados obsoletos o inseguros, priorizando modalidades como la carga rápida y ultra-rápida mediante corriente alterna (AC) y continua (DC), respectivamente (Agencia de Regulación y Control de Electricidad (ARCONEL), 2024).

El modelo tarifario se configura de manera diferenciada para vehículos livianos y pesados, de modo que se ajusta a la capacidad técnica y al tipo de infraestructura utilizada. Para el año 2025,

se establecen techos tarifarios que garantizan la accesibilidad económica del servicio sin desincentivar la inversión privada. Por ejemplo, el costo máximo permitido para carga ultra-rápida DC alcanza los 28,51 ctvs/kWh en el caso de vehículos livianos y 17,51 ctvs/kWh para vehículos pesados, evidenciando una estructura flexible pero controlada (Agencia de Regulación y Control de Electricidad (ARCONEL), 2024). La regulación también establece parámetros claros de facturación y control, obligando a los operadores a detallar en cada factura el consumo exacto de energía, el precio aplicado y el límite regulatorio correspondiente (Agencia de Regulación y Control de Electricidad (ARCONEL), 2024).

## **2.8 ORDENANZA MUNICIPAL Y PLAN DE ELECTROMOVILIDAD DE LA CIUDAD DE CUENCA**

La Ordenanza para la Promoción y Fortalecimiento de la Movilidad Activa en el cantón Cuenca (Concejo Cantonal de Cuenca, 2021), tiene como objetivo impulsar el uso de medios de transporte sostenibles y no motorizados, como el desplazamiento a pie y el uso de la bicicleta, mediante la aplicación de políticas públicas locales. Este instrumento normativo establece lineamientos para el desarrollo de infraestructura segura y accesible, la adecuación del espacio público y la ejecución de acciones orientadas a mejorar la seguridad vial y la calidad de vida de la ciudadanía. Asimismo, promueve la incorporación de estos modos de transporte en los procesos de planificación urbana, contribuyendo a una movilidad más equitativa, eficiente y ambientalmente responsable. Esta ordenanza constituye un componente fundamental para la transición hacia una ciudad más saludable y con menor huella ecológica.

El Plan de Electromovilidad de Cuenca (Municipio de Cuenca – Dirección General de Movilidad (DGM) y EMOV EP, 2023), fue presentado en abril de 2023 como una estrategia integral orientada a promover la descarbonización del transporte en el cantón, con una proyección hasta el año 2030. Entre los objetivos principales del plan se plantea que, para dicho año, al menos el 10 % de la flota de buses urbanos y de los vehículos pertenecientes a empresas privadas sea eléctrica. Asimismo, se contempla la implementación de 25 puntos de recarga pública y la habilitación de 8 estaciones exclusivas destinadas a flotas comerciales y de transporte público. Adicionalmente, se busca disminuir en un 5 % las emisiones de gases de efecto invernadero provenientes del parque automotor y alcanzar una participación del 5 % de vehículos eléctricos dentro del total del parque vehicular (Municipio de Cuenca – Dirección General de Movilidad (DGM) y EMOV EP, 2023).

El plan incluye un total de 44 medidas estratégicas organizadas en diferentes ejes, tales como gobernanza, desarrollo de infraestructura, incentivos económicos, regulación local y educación ciudadana. Además, contempla la capacitación de al menos 600 personas en temas relacionados con la electromovilidad hasta el año 2030. También se prevé la incorporación progresiva de bicicletas eléctricas al sistema BICICUENCA, con el objetivo de que representen el 30 % del total. De igual forma, se plantea el reemplazo paulatino de las flotas municipales y del servicio de taxis por unidades eléctricas, con una meta de 20 taxis eléctricos para el año 2025 y 80 unidades para 2030 (Municipio de Cuenca – Dirección General de Movilidad (DGM) y EMOV EP, 2023).

## **2.9 CARACTERÍSTICAS SOCIOECONÓMICAS DE LOS POTENCIALES USUARIOS DE VE.**

Diversos estudios han identificado que la adopción de VE está fuertemente influenciada por características socioeconómicas específicas de los consumidores. En el caso colombiano, los resultados de encuestas aplicadas a una muestra de 200 personas revelan que quienes muestran mayor disposición a adquirir un VE pertenecen, en su mayoría, a estratos socioeconómicos medios y altos, con edades entre los 30 y 60 años. Este grupo suele contar con un nivel educativo medio o alto y reside principalmente en áreas urbanas de ciudades principales, donde existe mayor conocimiento sobre la tecnología y mejores condiciones de infraestructura de carga. Uno de los factores más relevantes es el poder adquisitivo. El costo de los VE continúa siendo una barrera significativa, por lo que su adquisición está generalmente limitada a personas con ingresos superiores al promedio. Asimismo, el nivel de información y conciencia ambiental influye en la decisión de compra: quienes están más familiarizados con los problemas ambientales y reconocen al vehículo eléctrico como una solución sostenible tienden a mostrar mayor interés en su adopción. Además, se identifica una brecha informativa considerable, ya que una parte significativa de la población desconoce tanto los beneficios como los incentivos gubernamentales disponibles, lo que restringe su interés y acceso a este tipo de tecnología. Las personas que cuentan con mejor acceso a la información, a medios digitales, y que poseen formación técnica o profesional, son quienes más fácilmente consideran a los VE como una opción viable (Reinemer, 2018).

## **2.10 EMPRESA ELÉCTRICA REGIONAL CENTROSUR**

El área de cobertura de la Empresa Eléctrica Regional Centro Sur (CENTROSUR) abarca un total de 30.273,40 km<sup>2</sup>, lo que equivale al 11,71 % del territorio nacional. De esta extensión, el 64,96 % corresponde a la provincia de Morona Santiago, con 19.666 km<sup>2</sup>, excluyendo las zonas de Huamboya, Pablo Sexto y Gualaquiza, donde el servicio se presta de forma parcial. Por su parte, la provincia del Azuay representa el 26,19 % del total, equivalente a 7.928 km<sup>2</sup>, mientras que el 8,12 % restante corresponde a la provincia de Cañar, con excepción de los cantones Azogues y Déleg. A pesar de que Morona Santiago constituye la mayor proporción del área de servicio, la mayor concentración de usuarios se encuentra en la provincia del Azuay, que abarca el 26,19 % del territorio atendido (Empresa Eléctrica Regional CENTROSUR C.A, 2022).

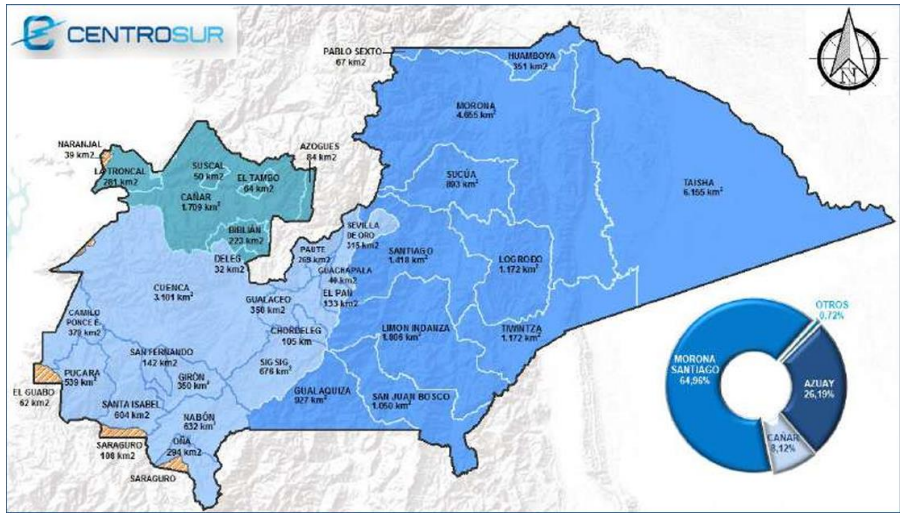


Figura 2: Mapa del área de servicio.

Fuente: (Empresa Eléctrica Regional CENTROSUR C.A., 2022).

El sistema eléctrico de la Empresa Eléctrica Regional Centro Sur (EERCS) está compuesto por subestaciones de reducción y seccionamiento, transformadores de potencia y distribución, líneas de distribución de baja tensión, alimentadores primarios, líneas de subtransmisión, luminarias destinadas al alumbrado público y la infraestructura correspondiente al sistema de medición (Empresa Eléctrica Regional Centro Sur C.A., 2025).

La configuración de los alimentadores se establece según los dispositivos de protección y maniobra que los componen, permitiendo su redireccionamiento en situaciones de emergencia para aislar posibles fallas. En la Empresa Eléctrica, los alimentadores se identifican mediante un código de cuatro dígitos: los dos primeros corresponden a la subestación a la que están conectados, el tercer dígito indica el nivel de tensión (0 para 6,3 kV, 1 para 13,8 kV y 2 para 22 kV), y el cuarto dígito representa el número de salida de la subestación correspondiente (Empresa Eléctrica Regional Centro Sur C.A., 2025). Ver tabla 1.

Tabla 3: Alimentadores pertenecientes a CENTROSUR.

Fuente: (Empresa Eléctrica Regional Centro Sur C.A., 2025)

S/E	DENOMINACION	TIPO	VOLTAJE	ALIMENTADOR
01	Luis Cordero	Reducción	22/6,3	0101-0102-0103-0104
02	Centenario	Reducción	22/6,3	0201- 0202- 0203-0204-0205
03	Monay	Reducción	69/22	0321- 0322- 0323-0324-0325
04	Parque industrial	Reducción	69/22	0421- 0422- 0423- 0424-0425- 0426-0427-0428

05	El Arenal	Reducción	69/22	0521- 0522- 0523- 0524- 0525- 0526- 0527- 0528- 0529- 05210
07	Ricaurte	Reducción	69/22	0721- 0722- 0723
08	Turi	Reducción	69/22	0821- 0822- 0823- 0824
09	Huablincay	Reducción	69/22	0921
12	El Descanso	Reducción	69/22	1221- 1222- 1223
13	Chaulayacu	Reducción	69/22	1321- 1322- 1323
14	Lentag	Reducción	69/22	1421- 1422- 1423- 1424-1425
15	Gualaceo	Reducción	69/22	1521- 1522- 1523- 1524
17	Los Cerezos	Reducción	69/22	1721- 1722- 1723- 1724
18	Cañar	Reducción	69/22	1821- 1822- 1823- 1824-1825
21	Macas	Reducción	69/13,8	2121- 2122- 2123- 2125-2127-2128
22	Méndez	Reducción	69/13,8	2211- 2212- 2213- 2223
23	Limón	Reducción	69/13,8	2311- 2312
50	La Troncal	Reducción	69/13,8	5011- 5012- 5013- 5014-5015

## 2.11 ARC GIS

ArcGIS constituye una herramienta clave en esta investigación, ya que permite integrar, visualizar y analizar la información geoespacial relacionada con la red de distribución eléctrica del cantón Cuenca. Es un conjunto de programas de Sistemas de Información Geográfica (SIG) desarrollado por ESRI (Environmental Systems Research Institute). Este software es escalable y se presenta en tres niveles de licencia: ArcView, Arc Editor y ArcInfo, los cuales se diferencian por la cantidad de funciones que ofrecen, siendo Arc Editor más avanzado que ArcView, y ArcInfo el más completo de los tres. Cada uno de estos niveles incluye dos aplicaciones principales: ArcMap y ArcCatalog. ArcMap permite visualizar, analizar información geográfica y elaborar mapas, mientras que ArcCatalog se utiliza para la administración y organización de los datos. Adicionalmente, se dispone de ArcToolBox, una herramienta accesible tanto desde

ArcMap como desde ArcCatalog, que reúne diversas funciones para la conversión, manipulación y análisis de datos geoespaciales (Collado Latorre y Navarro Jover, 2013).

En ArcMap, el mapa representa el nivel más alto de organización dentro del programa. Este puede estar compuesto por marcos de datos que contienen información geográfica y actúan como mapas interactivos, permitiendo trabajar directamente con los datos espaciales. Además, incluye documentos de presentación que reúnen los resultados obtenidos, facilitando su visualización de forma estructurada y atractiva (Cárdenas Bravo y Machuca Chabla, 2024).

## **2.12 PYTHON**

Python ofrece múltiples ventajas que lo convierten en una herramienta altamente atractiva, tanto en entornos profesionales como en el ámbito educativo para el aprendizaje de la programación. Desde una perspectiva didáctica, destacan especialmente las siguientes características:

- Python se caracteriza por ser un lenguaje altamente expresivo, lo que permite escribir programas más concisos y compactos, generalmente con una extensión menor en comparación con su equivalente en lenguajes como C.
- Python destaca por su legibilidad, gracias a una sintaxis clara y estructurada que facilita la escritura de programas más comprensibles en comparación con otros lenguajes de programación.
- Python proporciona un entorno interactivo que permite realizar pruebas de manera sencilla, lo cual resulta útil para explorar y aclarar aspectos específicos del lenguaje.

Los lenguajes de programación evolucionan con el tiempo, adaptándose a nuevas necesidades y mejoras, y Python no es la excepción. La experiencia acumulada con cada versión del lenguaje, junto con la influencia de otros lenguajes utilizados por los desarrolladores, genera una demanda constante por incorporar nuevas funcionalidades o simplificar la forma en que se expresan determinados procesos. Python fue creado originalmente por Guido van Rossum, quien se basó en su experiencia previa participando en el desarrollo del lenguaje experimental ABC (Marzal Varó et al., 2014).

Python puede ejecutarse en editores de código fuente livianos pero potentes, como Visual Studio Code, el cual ofrece múltiples funcionalidades que lo convierten en una herramienta altamente eficiente. Además, este entorno cuenta con un amplio ecosistema de extensiones compatibles con diversos lenguajes de programación y proporciona un sólido soporte para tareas de depuración (Cárdenas Bravo y Machuca Chabla, 2024).

## **2.13 OPEN DSS**

OpenDSS es una herramienta ampliamente utilizada para la simulación de redes de distribución eléctrica. Desarrollado por el EPRI, este software permite realizar análisis en distintos intervalos de tiempo, como ciclos anuales, semanales o diarios. Su estructura, basada en comandos de texto, ofrece una gran flexibilidad en la configuración del modelo y facilita la identificación de posibles errores en la codificación. Al estar diseñado para representar redes eléctricas reales, es fundamental incorporar medidores y monitores dentro del sistema a analizar. Además, se debe ajustar la frecuencia a 60 Hz para asegurar la validez de los resultados. También es necesario liberar la memoria de configuraciones previas y establecer correctamente la ruta de trabajo,

especialmente cuando se trabaja con esquemas de modelado modular (Vera Silva y Marulanda Guerra, 2018).

### 2.13.1 LÍNEAS DE DISTRIBUCIÓN

En OpenDSS, las líneas pueden ser definidas directamente mediante el objeto *Line*. Estas se caracterizan por sus impedancias, las cuales pueden especificarse a través de sus componentes simétricas de secuencia o mediante matrices de impedancia, que deben ser previamente calculadas para ser incorporadas como parámetros del objeto *Line* (Palacios, 2020). Los parámetros para las líneas se presentan en la Tabla 2.

Tabla 4: Parámetros de líneas de distribución. Fuente: (Palacios, 2020)

Parámetro	Descripción
Phases	Número de fases. Por defecto es 3.
Bus1	Nombre de la barra a la cual está conectado el primer terminal.
Bus2	Nombre de la barra a la cual está conectado el segundo terminal.
LineCode	Nombre del objeto Line Code que contiene la definición de impedancias, en caso de existir.
Length	Factor por el cual se multiplica los datos de impedancia.
RO, XO, R1, X1	Resistencia y reactancia de secuencia cero y positiva en ohmios por unidad de longitud.
C0, C1	Capacitancias de secuencia cero y positiva en micro siemens por unidad de longitud.
B0, B1	Susceptancias de secuencia cero y positiva en micro siemens por unidad de longitud. Alternativa C0 y C1
Rmatrix	Matriz de resistencias en ohmios por unidad de longitud.
Xmatrix	Matriz de reactancias en ohmios por unidad de longitud.
Cmatrix	Matriz de capacitancias en nano faradios por unidad de longitud.
Geometry	Nombre del objeto Line Geometry que contiene la geometría la línea, en caso de existir.
BaseFreq	Frecuencia de impedancias descritas. Por defecto es 60 Hz

### 2.13.2 TRANSFORMADORES

OpenDSS permite modelar transformadores de distintos tipos, siendo la conexión por defecto del tipo estrella-delta. En este contexto, el número de conductores por terminal corresponde a uno más que el número de fases, ya que se incluye un conductor adicional destinado al neutro. En configuraciones delta, dicho terminal de neutro permanece sin conexión (Palacios, 2020). Los

principales parámetros que se utilizan para modelar un transformador se presentan en la Tabla 3.

Tabla 5: Parámetros de transformadores de distribución. Fuente: (Palacios, 2020)

Parámetro	Descripción
Phases	Número de fases, por defecto es 3.
Windings	Número de devanados, por defecto es 2.
XHL	Porcentaje de reactancia alto a bajo devanado 1 a 2.
Wdg	Devanado que tendrá los parámetros especificados.
Bus	Nombre de la barra a la cual se conecta el terminal del devanado.
Conn	Tipo de conexión del devanado, estrella o delta.
KV	Voltaje nominal del devanado, línea a línea para trifásico.
Tap	Tap por unidad del devanado.
%R	Porcentaje de resistencia del devanado respecto a los KVA base.
%loadloss	Porcentaje de pérdidas a carga nominal.
%noloadloss	Porcentaje de perdidas en vacío con base a voltaje nominal.

### 2.13.1 CARGAS

En OpenDSS, las cargas se modelan utilizando el objeto *Load*. Cuando se definen cargas trifásicas, estas se consideran balanceadas por defecto; sin embargo, si se desea representar cargas desbalanceadas, es necesario definir tres cargas monofásicas de manera individual. Las cargas pueden especificarse de tres formas distintas: mediante potencia activa y factor de potencia, potencia activa y reactiva, o a través de potencia aparente y factor de potencia (Palacios, 2020). Los parámetros de una carga se presentan en la Tabla 4.

Tabla 6: Parámetros de cargas de distribución.

Fuente: (Palacios, 2020)

Parámetro	Descripción
Bus1	Nombre de la barra a cuál está conectada la carga.
Phases	Número de fases
KV	Voltaje base para la carga.
KVA	Potencia aparente base total.
KW	Potencia activa nominal total.

KVAR	Potencia reactiva total.
FP	Factor de potencia nominal de la carga. Valor positivo es atraso.
Model	Número entero que indica el modelo de carga que utiliza.
Conn	Tipo de conexión delta o triangulo. Por defecto triangulo.

## CAPITULO 3 DESARROLLO

### 3.1 RECOPIACIÓN DE DATOS

La recopilación de información representa una fase esencial en el desarrollo del presente estudio, ya que permite establecer una base sólida para proyectar el crecimiento de la demanda de vehículos eléctricos en el cantón Cuenca. Para garantizar la confiabilidad de los datos, se recurre a las estadísticas publicadas por la Asociación de Empresas Automotrices del Ecuador (AEADE), entidad que registra y difunde periódicamente la comercialización de vehículos en el país, clasificados por tipo de tecnología y por provincia (Asociación de Empresas Automotrices del Ecuador (AEADE), 2023).

En esta investigación se toma como referencia el periodo comprendido entre los años 2016 y 2023, con un enfoque específico en los vehículos eléctricos. La información recopilada permite identificar el comportamiento del mercado en la provincia del Azuay, donde se registran un total acumulado de 161 unidades comercializadas durante dicho intervalo (Asociación de Empresas Automotrices del Ecuador (AEADE), 2023).

A continuación, se presentan una tabla que resume el número de vehículos eléctricos vendidos anualmente en la provincia del Azuay, así como el total acumulado correspondiente al periodo de análisis. Ver tabla 7.

*Tabla 7: Venta de vehículos eléctricos anuales y acumulados.*

*Fuente: (Asociación de Empresas Automotrices del Ecuador (AEADE), 2023)*

PROVINCIA	AÑO	VENTAS DE VE	VENTAS ACUMULADAS DE ELECTRICOS
AZUAY	2016	0	0
AZUAY	2017	1	1
AZUAY	2018	1	2
AZUAY	2019	1	3
AZUAY	2020	42	45
AZUAY	2021	10	55
AZUAY	2022	17	72
AZUAY	2023	89	161

Para estimar la cantidad de vehículos eléctricos por Cantón, se adopta una distribución proporcional basada en la población. Esta decisión se justifica en estudios como (Ceballos Delgado et al., 2016), donde se resalta la importancia de caracterizar la carga total del sistema considerando factores demográficos. Dado que la densidad poblacional está estrechamente

relacionada con la demanda energética y la presencia de infraestructura urbana, distribuir los vehículos eléctricos según el tamaño poblacional permite establecer un punto de partida realista y técnicamente válido para evaluar el impacto sobre la red de distribución, especialmente en contextos con información limitada sobre hábitos de carga o infraestructura instalada. Para ello, se toman como referencia los datos de los padrones electorales más recientes, los cuales reflejan de manera actualizada la cantidad de habitantes en la zona (Consejo Nacional Electoral del Ecuador, 2024).

La estimación se realiza mediante la siguiente fórmula, que permite distribuir proporcionalmente el total de vehículos eléctricos registrados en la provincia del Azuay con base en la población de Cuenca:

$$\text{Vehículos eléctricos Cuenca} = \left( \frac{\text{Población Cuenca}}{\text{Población Azuay}} \right) * \text{Ventas Acumuladas de VE}$$

Este procedimiento garantiza un cálculo más preciso del parque vehicular eléctrico en la ciudad, asegurando que la estimación esté sustentada en una base demográfica representativa y alineada con la realidad territorial.

*Tabla 8: Distribución proporcional de Vehículos eléctricos para el Cantón Cuenca.*

*Fuente: (Asociación de Empresas Automotrices del Ecuador (AEADE), 2023), (Consejo Nacional Electoral del Ecuador, 2024)*

AÑO	POBLACIÓN AZUAY	POBLACIÓN CUENCA	VENTAS ACUMULADAS DE VE	VE CUENCA
2017	646455	453040	1	1
2018	646455	453040	2	1
2019	646455	453040	3	2
2020	646455	453040	45	32
2021	618406	435963	55	39
2022	618406	435963	72	51
2023	635051	448552	161	114

De manera similar, este proceso se aplica a cada una de las parroquias urbanas y rurales del cantón Cuenca, con el objetivo de obtener una distribución detallada de los vehículos eléctricos en cada una de ellas. Para ello, se adopta el mismo criterio de asignación proporcional, considerando la población de cada parroquia en relación con la población total del cantón.

$$\text{Vehículos eléctricos parroquias} = \left( \frac{\text{Población Parroquias}}{\text{Población Cuenca}} \right) * \text{Ventas Acumuladas de VE}$$

Los valores obtenidos mediante esta metodología se consideran como datos históricos, ya que reflejan la distribución territorial de los vehículos eléctricos en función de la población registrada en los últimos años.

Tabla 9: Distribución proporcional de Vehículos eléctricos para las parroquias de Cuenca. }

Fuente: (Asociación de Empresas Automotrices del Ecuador (AEADE), 2023), (Consejo Nacional Electoral del Ecuador, 2024).

PARROQUIAS	POBLACION POR PARROQUIA			PROPORCIONES DE VEHICULOS ELECTRICOS						
	PADRON 2017	PADRON 2021	PADRON 2023	2017	2018	2019	2020	2021	2022	2023
BAÑOS	22824	22198	23078	0	0	0	2	2	3	6
CHAUCHA	1260	1255	1286	0	0	0	0	0	0	0
CUMBE	6157	5440	5564	0	0	0	0	0	1	1
ICTORIA DEL PORTETI	4620	4278	4403	0	0	0	0	0	1	1
QUINGEO	5669	5906	6115	0	0	0	0	1	1	2
TARQUI	9624	8987	9198	0	0	0	1	1	1	2
EL VALLE	15426	15810	16528	0	0	0	1	1	2	4
SANTA ANA	5201	5344	5476	0	0	0	0	0	1	1
TURI	6046	6191	6519	0	0	0	0	1	1	2
PACCHA	4349	4280	4499	0	0	0	0	0	1	1
NULTI	4467	4706	5067	0	0	0	0	0	1	1
LLACAO	3790	3735	3885	0	0	0	0	0	0	1
RICAUARTE	16326	16957	17771	0	0	0	1	2	2	5
SIDCAY	4278	3617	3651	0	0	0	0	0	0	1
OCTAVIO CORDERO	2960	2175	2163	0	0	0	0	0	0	1
CHECA	4119	2934	2917	0	0	0	0	0	0	1
CHIQUINTAD	5125	4197	4258	0	0	0	0	0	0	1
SININCAY	11431	9854	10211	0	0	0	1	1	1	3
SAYAUSÍ	10945	11100	11640	0	0	0	1	1	1	3
MOLLETURO	6102	6284	6299	0	0	0	0	1	1	2
SAN JOAQUIN	8313	8466	8853	0	0	0	1	1	1	2
YANUNCAY	22120	24435	26705	0	0	0	2	2	3	7
EL BATAN	12738	13453	14227	0	0	0	1	1	2	4
SAN SEBASTIAN	34973	32648	33569	0	0	1	2	3	4	9
BELLAVISTA	16119	16349	16996	0	0	0	1	1	2	4
HERMANO MIGUEL	7988	9475	10019	0	0	0	1	1	1	3
EL VECINO	43512	39859	39984	1	1	1	3	4	5	10
MACHANGARA	8195	9273	10294	0	0	0	1	1	1	3
TOTORACOCCHA	21778	21587	22264	0	0	0	2	2	3	6
MONAY	12381	12927	13556	0	0	0	1	1	2	3
HUAYNACAPAC	17614	16662	16846	0	0	0	1	1	2	4
CAÑARIBAMBA	9461	9362	9536	0	0	0	1	1	1	2
SUCRE	23477	22064	22072	0	0	0	2	2	3	6
RAMIREZ DAVALOS	13229	11412	11141	0	0	0	1	1	1	3
SAN BLAS	23947	20883	20478	0	0	0	2	2	2	5
SAGRARIO	26476	21860	21484	0	0	0	2	2	3	5
TOTAL	453040	435963	448552	1	1	2	32	39	51	114

### 3.2 MODELO DE BASS EN PYHTON

Para implementar y ajustar el modelo de Bass, es necesario importar varias bibliotecas de Python que ayudan a realizar tareas de ajustes de curvas, cálculo de métricas y visualización de resultados.

- NumPy: es una biblioteca utilizada principalmente para el cálculo numérico y el procesamiento de grandes volúmenes de datos. facilita el trabajo con arreglos multidimensionales de manera eficiente y ofrece una variedad de funciones matemáticas y lógicas que permiten realizar operaciones sobre estos datos con gran rapidez (NumPy, 2025).
- Matplotlib.pyplot: Se usa para generar gráficos y visualizaciones de los resultados obtenidos (Matplotlib, 2025).
- Scipy.optimize: Contiene la función `curve_fit` que se utiliza para ajustar los parámetros del modelo de Bass a los datos históricos (SciPy, 2025).

- Pandas: Se utiliza para organizar los datos en formato tabular y exportarlos a un archivo Excel (Pandas, 2025).

Para determinar los parámetros del modelo de Bass ( $p$ ,  $q$ , y  $m$ ) en el contexto de la adopción de vehículos eléctricos en Cuenca, se llevó a cabo un proceso de ajuste basado en datos históricos de adopción. El modelo de Bass requiere la estimación precisa de tres parámetros fundamentales:  $p$ ,  $q$  y  $m$ . Cada uno de estos valores representa un componente distinto del proceso de adopción tecnológica. Este procedimiento se ejecuta computacionalmente a través del algoritmo Levenberg–Marquardt, incorporado en la función `curve_fit` de la biblioteca SciPy de Python. El uso del método de mínimos cuadrados no lineales se justifica principalmente por la estructura matemática del modelo de Bass, que es inherentemente no lineal y no puede ajustarse correctamente mediante técnicas de regresión lineal convencional. Aunque existen aproximaciones lineales del modelo, estas presentan serias limitaciones, entre ellas la incapacidad de estimar errores estándar para los parámetros y la pérdida de precisión al proyectar escenarios de largo plazo (Lazo Paz, 2021).

La regresión no lineal mediante mínimos cuadrados permite mantener la integridad estructural del modelo, asegurando una estimación simultánea y coherente de los parámetros  $p$ ,  $q$  y  $m$ . Asimismo, este método ofrece mayor robustez cuando se dispone de serie de datos cortas, evita sesgos inter temporales y permite incorporar restricciones en los valores de los parámetros (Lazo Paz, 2021).

El parámetro  $m$  no puede deducirse directamente de los datos observados se inicia su estimación utilizando un valor inicial aproximado, definido de manera empírica. En este caso, se adopta como valor inicial una multiplicación del valor máximo observado en el conjunto de datos históricos. Por ejemplo, en la parroquia Baños el valor máximo de adopciones acumuladas hasta el último año observado es 6, se puede establecer como valor inicial  $m$ , el doble de dicho valor, es decir, 12. Permitiendo que el algoritmo ajuste este valor hacia arriba o hacia abajo en función de la dinámica observada en la serie de datos.

El valor  $p$  muestra un inicio de lento proceso de adopción (en este caso la parroquia de Baños no tiene registros de adopción de 2017 hasta 2019), se parte de un valor inicial conservador de  $p = 0.01$ . Este valor permite reflejar un bajo nivel de adopción espontánea. El algoritmo ajusta posteriormente este valor de acuerdo con la pendiente observada en los primeros años.

En los datos considerados, se observa un crecimiento más notorio después del año 2020 en la parroquia Baños, lo cual sugiere que la imitación  $q$  comienza a jugar un papel más relevante. Por este motivo, se adopta como valor inicial  $q = 0.3$ , dentro de los rangos típicos reportados en la literatura. El algoritmo ajusta dinámicamente este valor en función de la tasa de crecimiento acumulado en los años posteriores, buscando el punto óptimo que reproduzca con mayor precisión el comportamiento observado.

En la siguiente tabla se presentan los valores de los parámetros  $p$ ,  $q$  y  $m$  obtenidos por el algoritmo para cada una de las parroquias del cantón Cuenca. Además, se incluyen los indicadores de ajuste del modelo, como el error cuadrático medio (MSE), el coeficiente de determinación ( $R^2$ ) y el porcentaje de inserción estimado en cada parroquia. Ver tabla 9.

Tabla 10: Parámetros p, q, y m del modelo de Bass para la proyección de vehículos eléctricos en Cuenca.

Fuente: (Asociación de Empresas Automotrices del Ecuador (AEADE), 2023)

PARROQUIAS	VALORES DEL MODELO DE BASS			% DE ERROR		%
	P	Q	M	MSE	R2	INSERCCION
BAÑOS	0,002519	0,523149	60	0,2857	93%	0,00251322
CHAUCHA	0	0	0	0	0%	0
CUMBE	0,004108	1	2	0	100%	0,00035945
VICTORIA DEL PORTETI	0,004108	1	2	0	100%	0,00045424
QUINGEO	0,005487	0,887148	3	0	100%	0,0004906
TARQUI	0,005829	0,309672	20	0	100%	0,00173951
EL VALLE	0,001473	0,648363	40	0	100%	0,00242014
SANTA ANA	0,004108	1	2	0	100%	0,00036523
TURI	0,005487	0,887148	3	0	100%	0,00046019
PACCHA	0,004108	1	2	0	100%	0,00044454
MULTI	0,004108	1	2	0	100%	0,00039471
LLACAO	0,000229	1	10	0	100%	0,002574
RICAUARTE	0,001677	0,613497	50	0,285714	90%	0,0027573
SIDCAY	0,000229	1	10	0	100%	0,00273898
OCTAVIO CORDERO	0,000229	1	10	0	100%	0,00462321
CHECA	0,000229	1	10	0	100%	0,00342818
CHIQUINTAD	0,000229	1	10	0	100%	0,00234852
SININCAY	0,001593	0,618677	30	0,285714	71%	0,00293801
SAYAUSÍ	0,001593	0,618677	30	0,285714	71%	0,00257732
MOLLETURO	0,005487	0,887148	3	0	100%	0,00047627
SAN JOAQUIN	0,005829	0,309672	20	0	100%	0,0018073
YANUNCAY	0,001514	0,637132	70	0,285714	95%	0,00258379
EL BATAN	0,001473	0,648363	40	0	100%	0,00281156
SAN SEBASTIAN	0,001956	0,578933	90	0,285714	97%	0,00265126
BELLAVISTA	0,001473	0,648363	40	0	100%	0,00235349
HERMANO MIGUEL	0,001593	0,618677	30	0,285714	71%	0,00299431
EL VECINO	0,003706	0,426568	100	0,571429	94%	0,00232593
MACHANGARA	0,001593	0,618677	30	0,285714	71%	0,00291432
TOTORACOCHA	0,00252	0,523149	60	0,285714	93%	0,0026051
MONAY	0,011348	0,655821	6	0	100%	0,00044261
HUAYNACAPAC	0,001473	0,648363	40	0	100%	0,00237445
CAÑARIBAMBA	0,005829	0,309672	20	0	100%	0,00167785
SUCRE	0,00252	0,523149	60	0,285714	93%	0,00262776
RAMIREZ DAVALOS	0,001593	0,618677	30	0,285714	71%	0,00269276
SAN BLAS	0,003069	0,466919	50	0,428571	85%	0,00234398
SAGRARIO	0,004212	0,404269	50	0,285714	91%	0,00214113
TOTAL			1035			0,00223163

El código implementado no solo permite estimar los parámetros del modelo de Bass, sino que también genera de forma automática la representación gráfica de la curva sigmoideal característica de este modelo. Esta curva S refleja el comportamiento acumulativo de las adopciones de vehículos eléctricos a lo largo del tiempo, partiendo de una fase inicial lenta, seguida de un crecimiento acelerado por efecto de la imitación, hasta alcanzar una etapa de saturación del mercado. En dicha gráfica se visualizan tanto los datos históricos como las proyecciones obtenidas, así como los indicadores de ajuste que validan la precisión del modelo aplicado. Ver siguientes figuras de ejemplo.

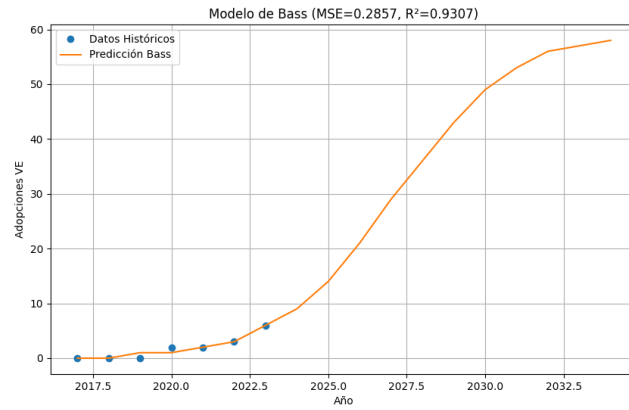


Figura 3: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Baños.

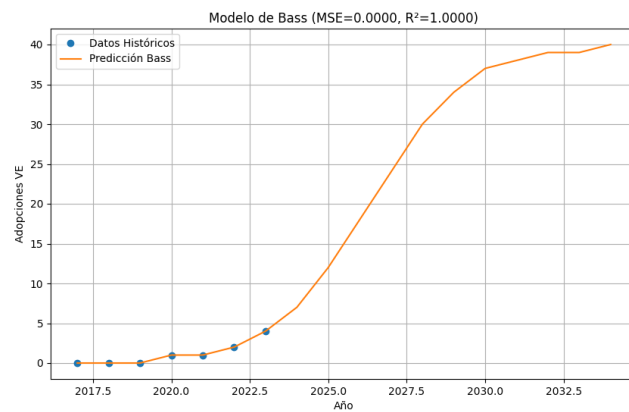


Figura 4: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Bellavista.

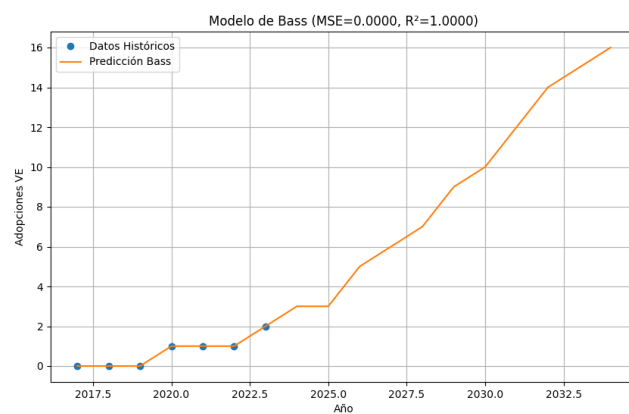


Figura 5: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Cañaribamba.

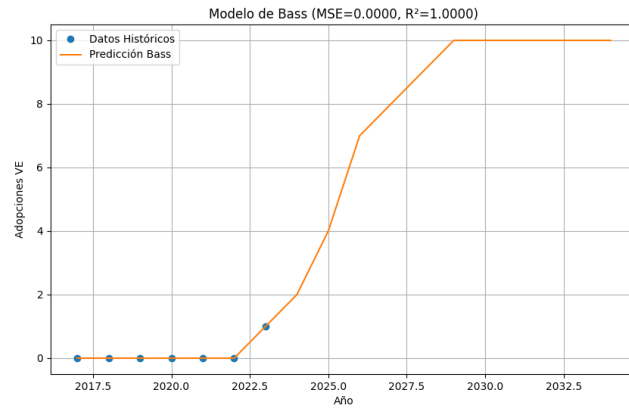


Figura 6: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Checa.

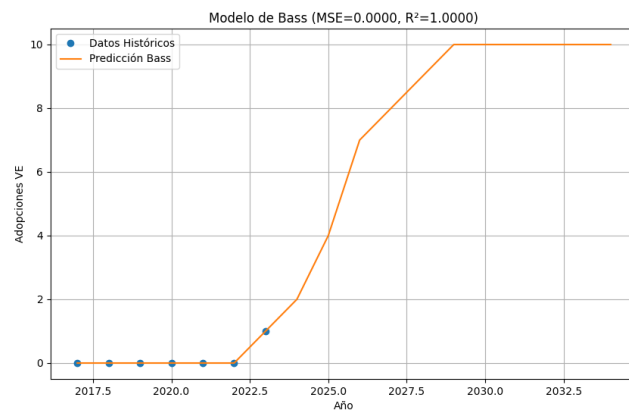


Figura 7: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Chiquintad.

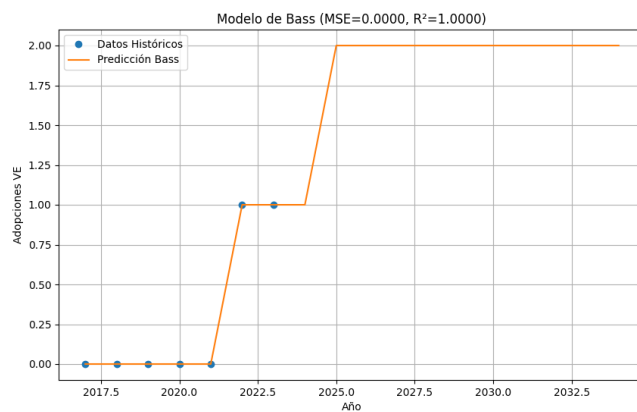


Figura 8: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Cumbe.

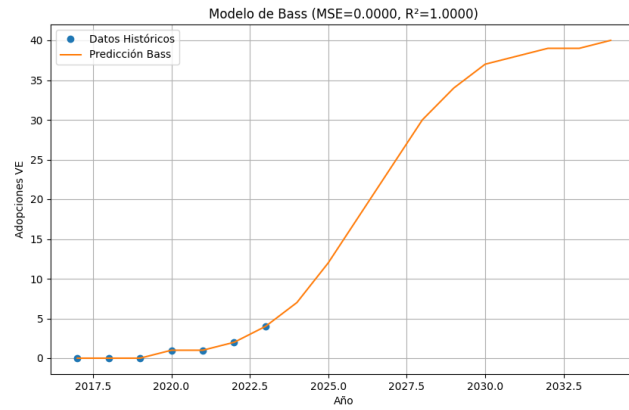


Figura 9: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia El Batán.

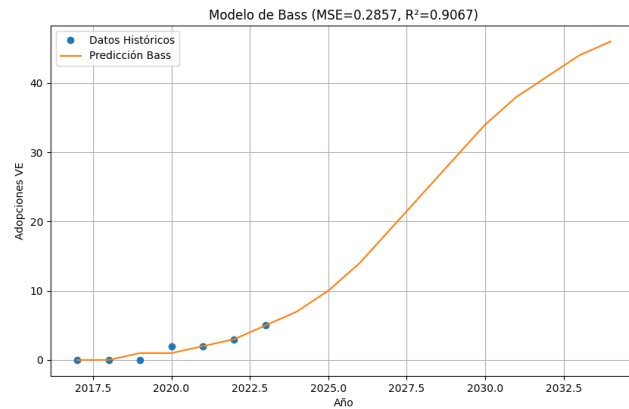


Figura 10: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia El Sagrario.

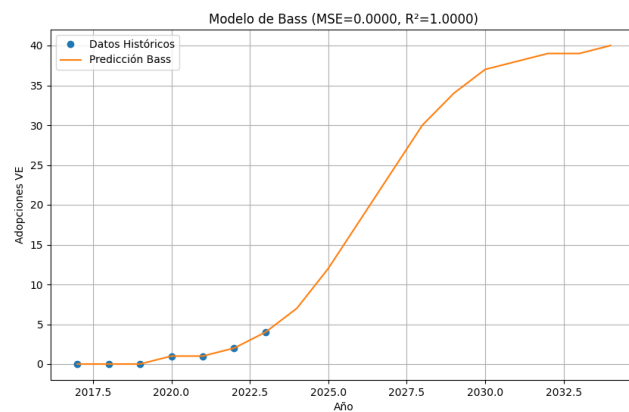


Figura 11: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia El Valle.

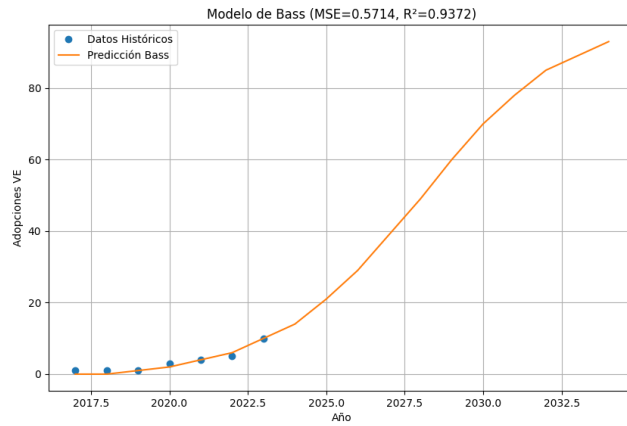


Figura 12: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia El Vecino.

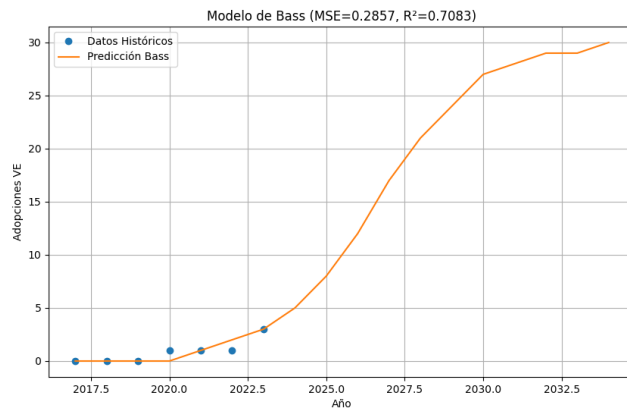


Figura 13: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Hermano Miguel.

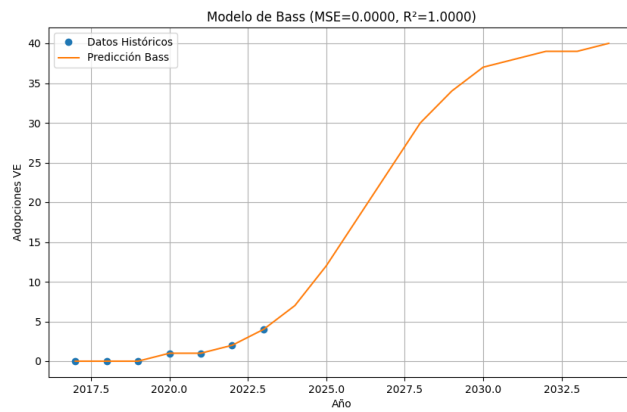


Figura 14: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Huayna Cápac.

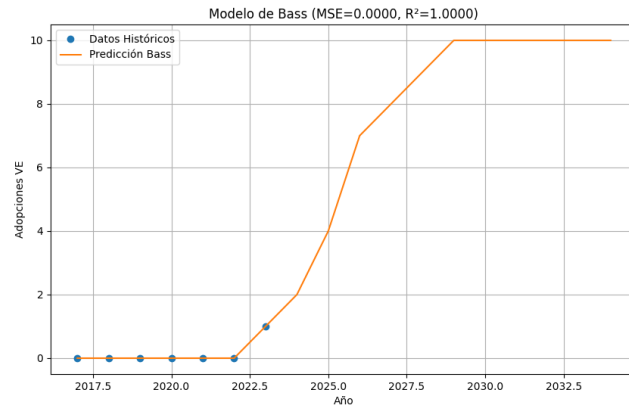


Figura 15: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Llaoca.

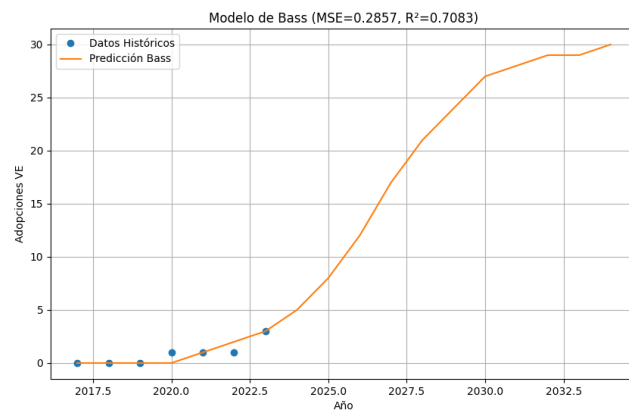


Figura 16: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Machángara.

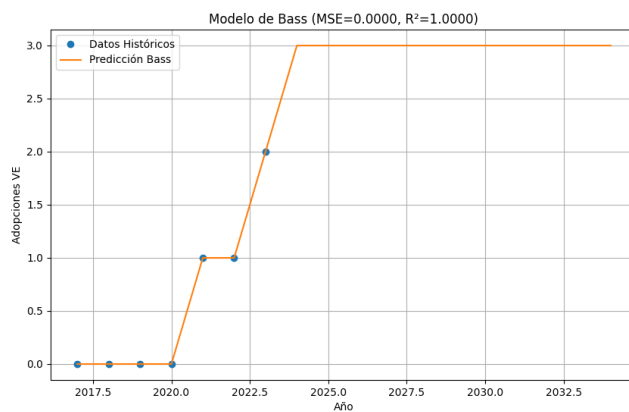


Figura 17: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Molleturo.

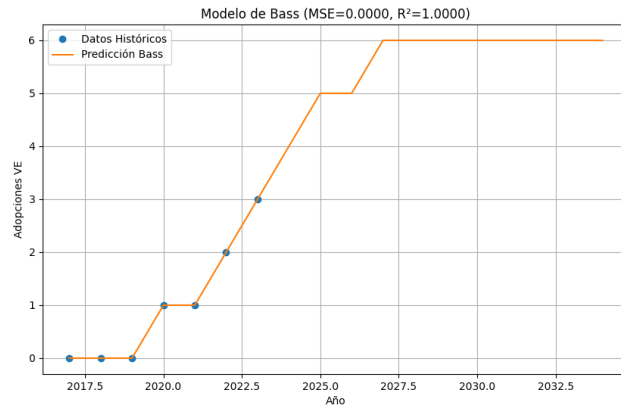


Figura 18: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Monay.

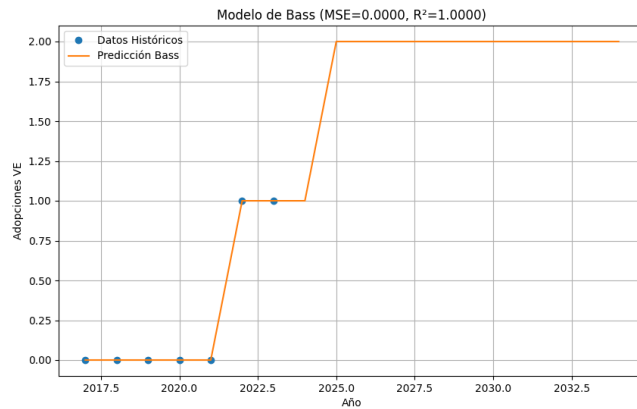


Figura 19: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Nulti.

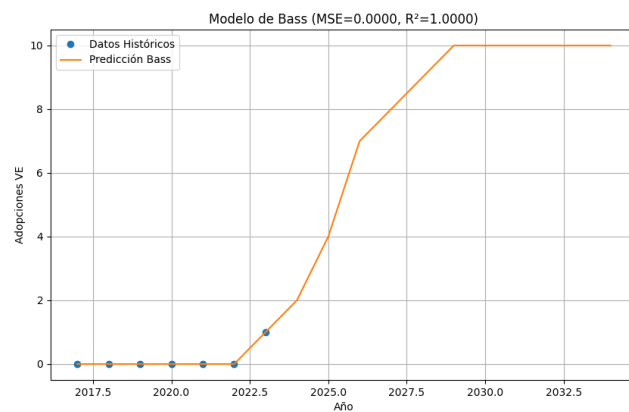


Figura 20: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Octavio Cordero Palacios.

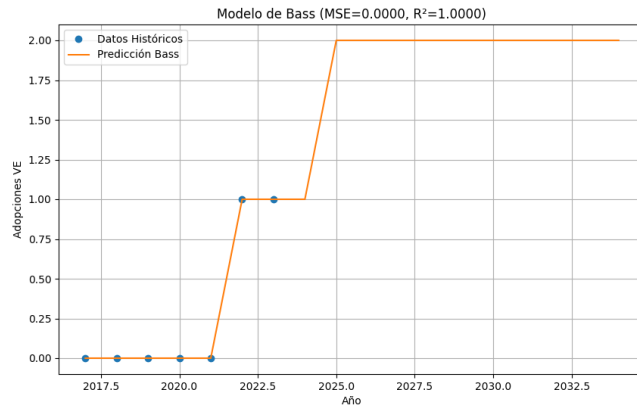


Figura 21: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Paccha.

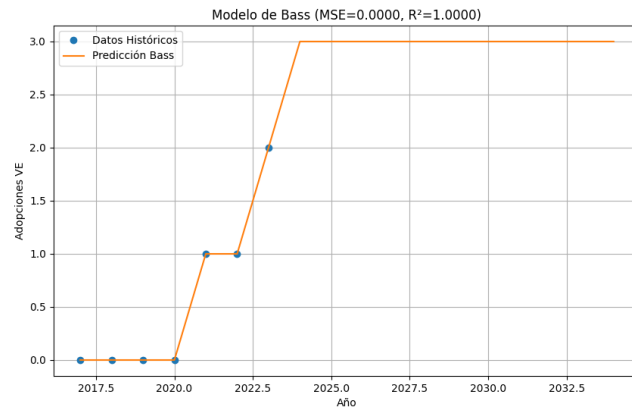


Figura 22: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Quingeo.

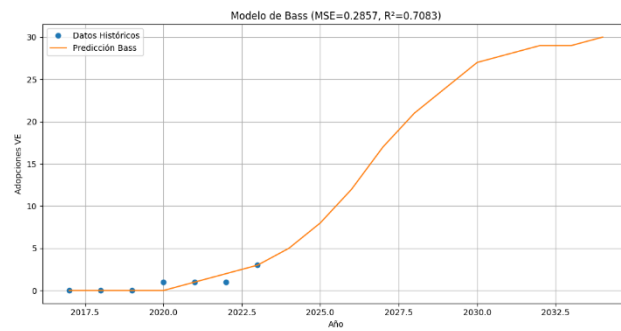


Figura 23: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Ramírez Dávalos.

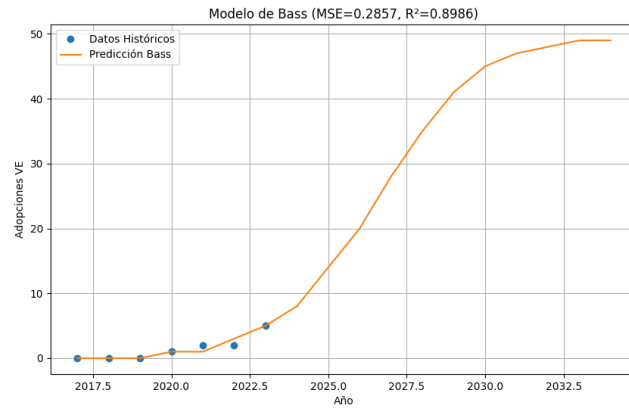


Figura 24: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Ricaurte.

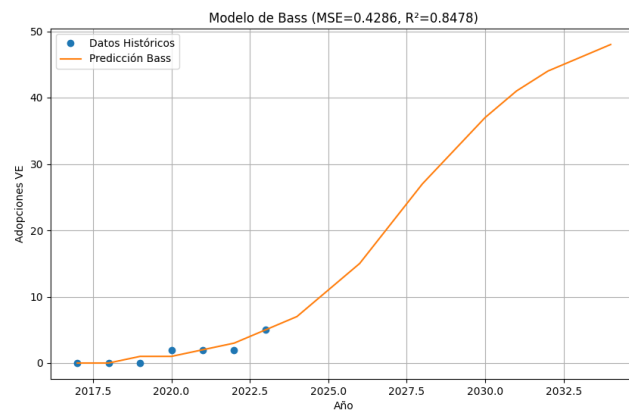


Figura 25: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia San Blas.

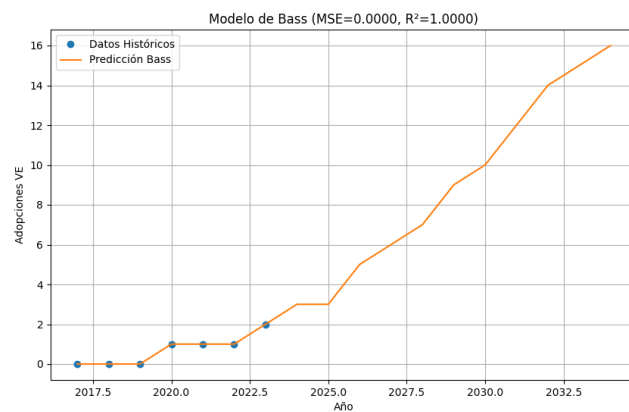


Figura 26: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia San Joaquín.

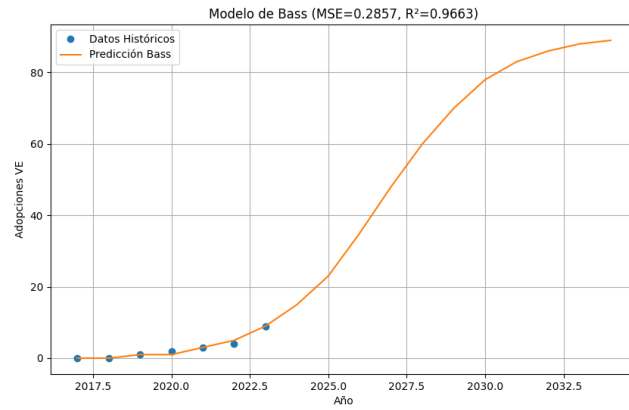


Figura 27: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia San Sebastián.

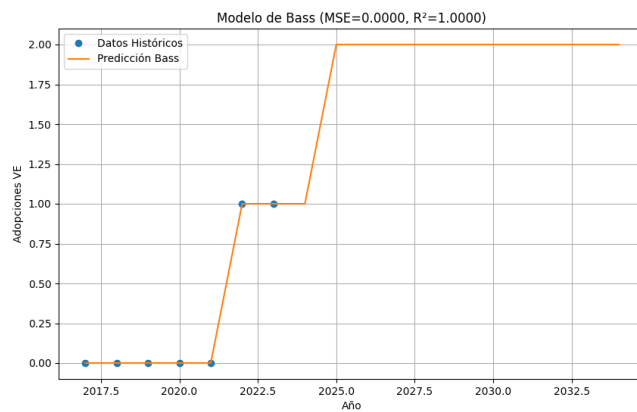


Figura 28: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Santa Ana.

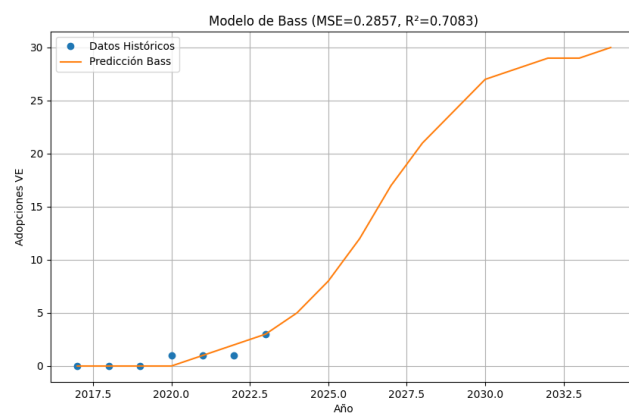


Figura 29: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Sayausí.

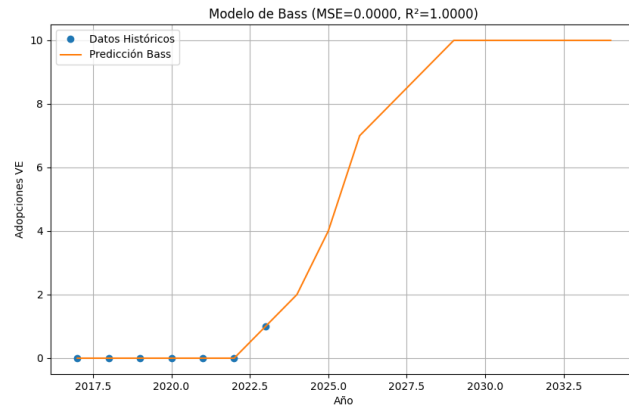


Figura 30: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Sidcay.

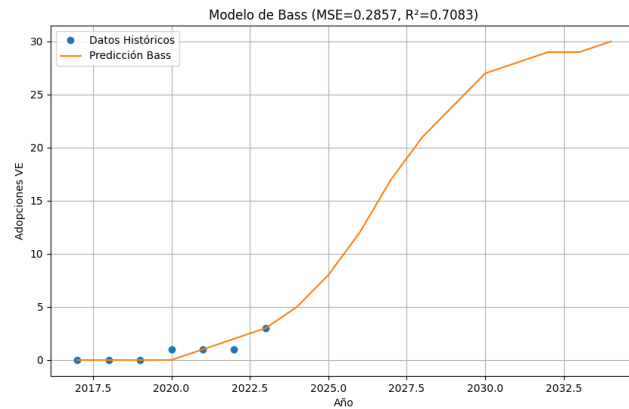


Figura 31: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Sinincay.

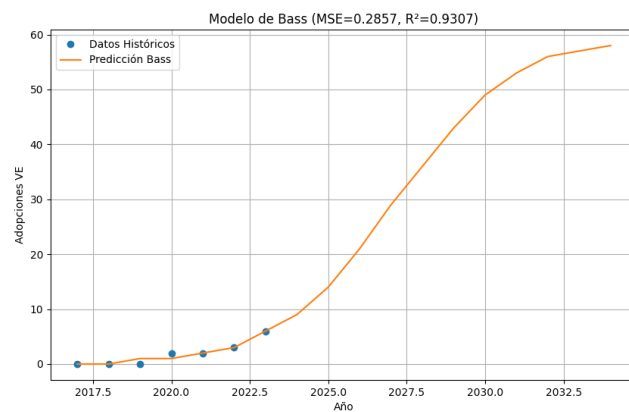


Figura 32: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Sucre.

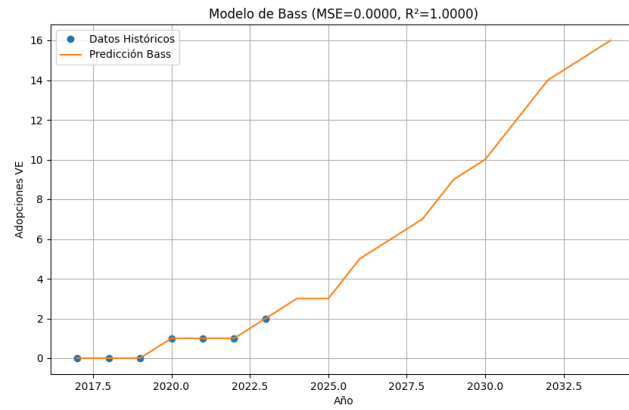


Figura 33: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Tarqui.

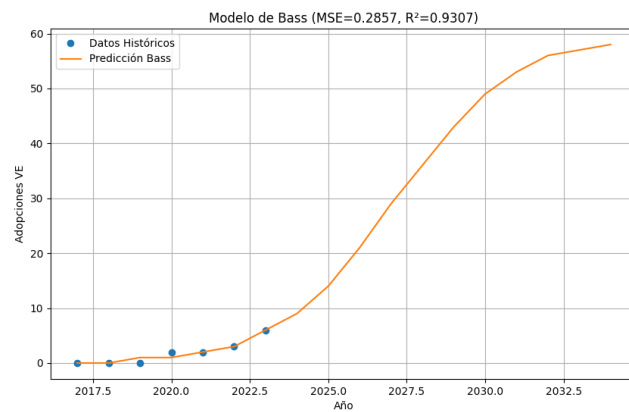


Figura 34: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Totoracocha.

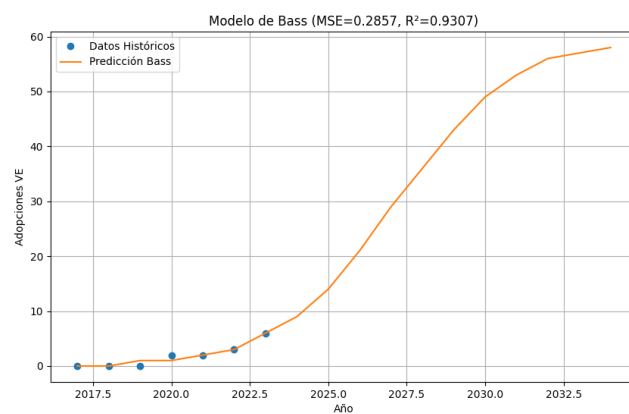


Figura 35: Curva sigmoideal del modelo de Bass aplicada a las adopciones de VE de la parroquia Turi.

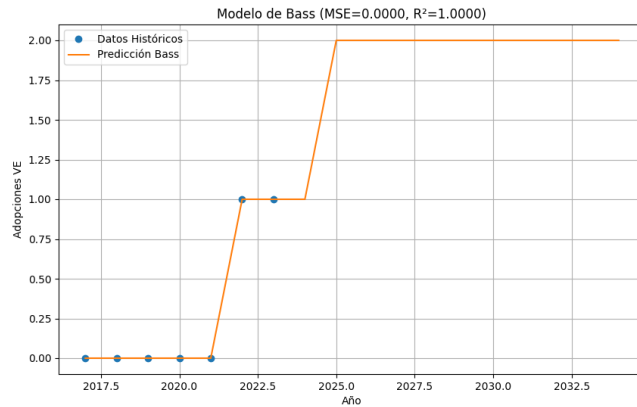


Figura 36: Curva sigmoide del modelo de Bass aplicada a las adopciones de VE de la parroquia Victoria del Portete.

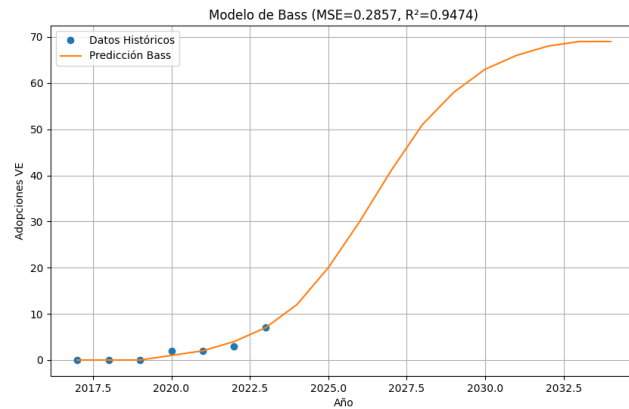


Figura 37: Curva sigmoide del modelo de Bass aplicada a las adopciones de VE de la parroquia Yanuncay.

A continuación, se presentan los resultados obtenidos de las proyecciones de adopción de vehículos eléctricos en cada una de las parroquias del cantón Cuenca. Estas proyecciones han sido generadas a partir de los parámetros ajustados del modelo de Bass, permitiendo estimar el comportamiento futuro de adopción tecnológica en el territorio, de acuerdo con la dinámica particular de cada parroquia y su potencial de crecimiento. Ver tabla 10.

Tabla 11: Resultados de las proyecciones de VE en las parroquias de Cuenca.

PARROQUIAS	PROYECCIONES DE VEHICULOS ELECTRICOS										
	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034
BAÑOS	9	14	21	29	36	43	49	53	56	57	58
CHAUCHA	0	0	0	0	0	0	0	0	0	0	0
CUMBE	1	2	2	2	2	2	2	2	2	2	2
VICTORIA DEL PORTETI	1	2	2	2	2	2	2	2	2	2	2
QUINGEO	3	3	3	3	3	3	3	3	3	3	3
TARQUI	3	3	5	6	7	9	10	12	14	15	16
EL VALLE	7	12	18	24	30	34	37	38	39	39	40
SANTA ANA	1	2	2	2	2	2	2	2	2	2	2
TURI	3	3	3	3	3	3	3	3	3	3	3
PACCHA	1	2	2	2	2	2	2	2	2	2	2
NULTI	1	2	2	2	2	2	2	2	2	2	2
LLACAO	2	4	7	8	9	10	10	10	10	10	10
RICAUARTE	8	14	20	28	35	41	45	47	48	49	49
SIDCAY	2	4	7	8	9	10	10	10	10	10	10
OCTAVIO CORDERO	2	4	7	8	9	10	10	10	10	10	10
CHECA	2	4	7	8	9	10	10	10	10	10	10
CHIQUINTAD	2	4	7	8	9	10	10	10	10	10	10
SININCAY	5	8	12	17	21	24	27	28	29	29	30
SAYAUSÍ	5	8	12	17	21	24	27	28	29	29	30
MOLLETURO	2	3	3	3	3	3	3	3	3	3	3
SAN JOAQUIN	3	3	5	6	7	9	10	12	14	15	16
YANUNCAY	12	20	30	41	51	58	63	66	68	69	69
EL BATAN	7	12	18	24	30	34	37	38	39	39	40
SAN SEBASTIAN	15	23	35	48	60	70	78	83	86	88	89
BELLAVISTA	7	12	18	24	30	34	37	38	39	39	40
HERMANO MIGUEL	5	8	12	17	21	24	27	28	29	29	30
EL VECINO	14	21	29	39	49	60	70	78	85	89	93
MACHANGARA	5	8	12	17	21	24	27	28	29	29	30
TOTORACOCHA	9	14	21	29	36	43	49	53	56	57	58
MONAY	4	5	5	6	6	6	6	6	6	6	6
HUAYNACAPAC	7	12	18	24	30	34	37	38	39	39	40
CAÑARIBAMBA	3	3	5	6	7	9	10	12	14	15	16
SUCRE	9	14	21	29	36	43	49	53	56	57	58
RAMIREZ DAVALOS	5	8	12	17	21	24	27	28	29	29	30
SAN BLAS	7	11	15	21	27	32	37	41	44	46	48
SAGRARIO	7	10	14	19	24	29	34	38	41	44	46
TOTAL	179	282	412	547	670	777	862	915	958	977	1001

### 3.3 PARROQUIAS RURALES CON MAYOR CRECIMIENTO DE VE

En el análisis de las proyecciones de crecimiento de la movilidad eléctrica en las zonas rurales del cantón Cuenca, se identifican cinco parroquias que concentran la mayor cantidad de vehículos eléctricos estimados para el período 2024–2034: Baños, Ricaurte, El Valle, Sinincay y Sayausí. Ver tabla 11.

Tabla 12: Resultados de las proyecciones de VE en las parroquias de Cuenca.

PARROQUIAS	PROYECCIONES DE VEHICULOS ELECTRICOS										
	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034
BAÑOS	9	14	21	29	36	43	49	53	56	57	58
EL VALLE	7	12	18	24	30	34	37	38	39	39	40
RICAUARTE	8	14	20	28	35	41	45	47	48	49	49
SININCAY	5	8	12	17	21	24	27	28	29	29	30
SAYAUSÍ	5	8	12	17	21	24	27	28	29	29	30

Como complemento al modelo de Bass, se ajustó una función polinómica de grado tres que permite visualizar de forma aproximada la dinámica de crecimiento acumulado, siguiendo el patrón sigmoïdal característico de la adopción tecnológica.

Entre ellas, la parroquia de Baños presenta el mayor incremento previsto, pasando de 9 vehículos eléctricos en 2024 a un total de 58 unidades en 2034.

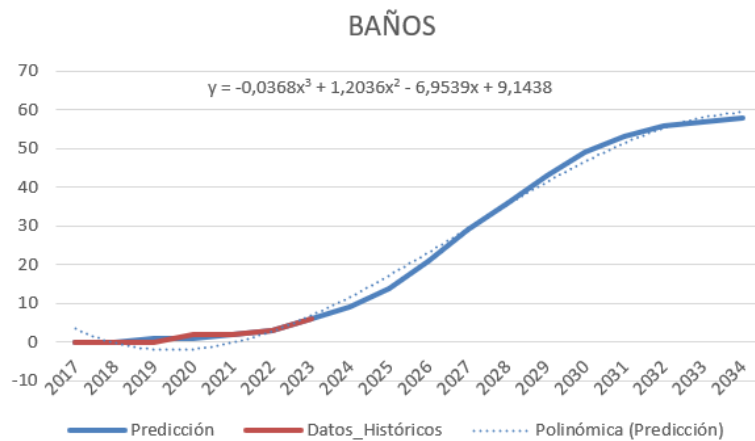


Figura 38: Representación polinómica de tercer grado como aproximación visual del comportamiento proyectado por el modelo de Bass en la parroquia Baños.

De manera similar, Ricaurte muestra un notable aumento, alcanzando 49 vehículos eléctricos. Ver figura 35. Por su parte, El Valle también evidencia un crecimiento significativo, incrementando de 7 a 40 vehículos eléctricos en el mismo intervalo. Ver figura 36. En tanto, las parroquias de Sinincay y Sayausí comparten un patrón de desarrollo paralelo, con un aumento desde 5 unidades en 2024 hasta llegar a 30 en 2034. Ver figura 37 y 38.

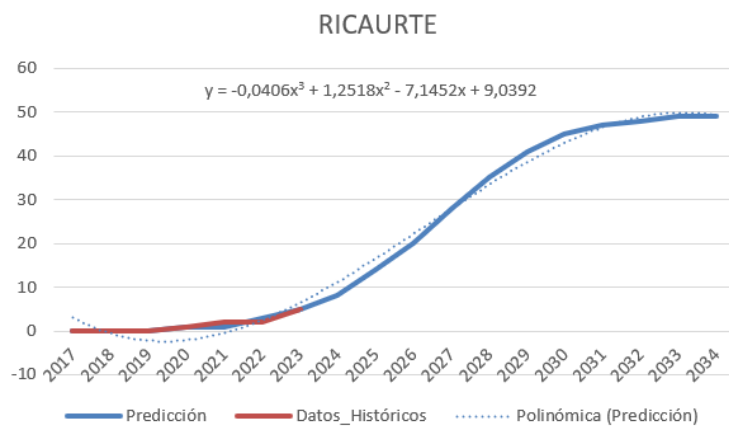


Figura 39: Representación polinómica de tercer grado como aproximación visual del comportamiento proyectado por el modelo de Bass en la parroquia Ricaurte.

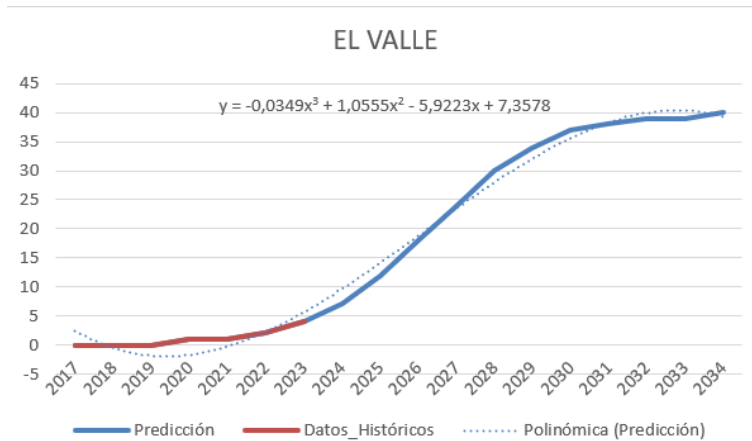


Figura 40: Representación polinómica de tercer grado como aproximación visual del comportamiento proyectado por el modelo de Bass en la parroquia El Valle.

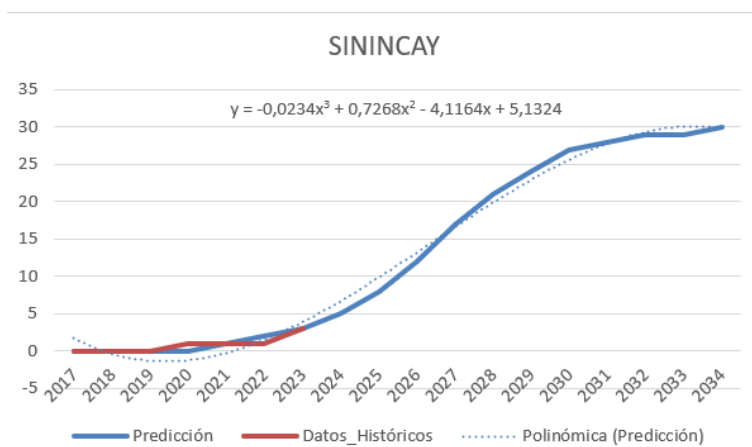


Figura 41: Representación polinómica de tercer grado como aproximación visual del comportamiento proyectado por el modelo de Bass en la parroquia Sinincay.

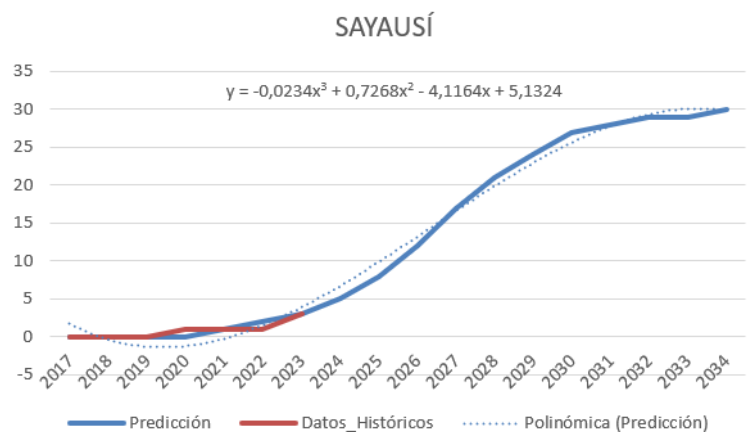


Figura 42: Representación polinómica de tercer grado como aproximación visual del comportamiento proyectado por el modelo de Bass en la parroquia Sayausí.

Estas parroquias forman parte de los sectores que han registrado un notable crecimiento poblacional, infraestructura vial y aumento de servicios básicos, siendo catalogadas dentro de las zonas de mayor jerarquía de asentamientos humanos en el área rural del cantón (GAD Municipal del Cantón Cuenca, 2024). Además, se resalta que parroquias como Baños, Ricaurte,

El Valle y Sinincay pertenecen al segundo nivel de jerarquía por su densidad poblacional, lo cual es un indicador clave de su dinamismo y desarrollo urbano sostenido (GAD Municipal del Cantón Cuenca, 2024).

El Valle y Ricaurte, por ejemplo, figuran como zonas con una alta concentración de población y urbanización creciente, mientras que Sinincay y Sayausí, aunque conservan características rurales, han mostrado un proceso de transformación con mejoras significativas en la dotación de servicios, infraestructura y conectividad (GAD Municipal del Cantón Cuenca, 2024).

### 3.3 SELECCIÓN DEL ALIMENTADOR

Para la selección del alimentador a analizar, se consideró inicialmente la proyección de incorporación de vehículos eléctricos en las diferentes parroquias del cantón Cuenca. Según la información analizada (ver Tabla 12), el alimentador 0104 se identifica como la zona con la mayor proyección de crecimiento en la adopción de vehículos eléctricos.

Tabla 13: Proyección de VE en parroquias abastecidas por el alimentador 0104.

PARROQUIAS	PROYECCIONES DE VEHICULOS ELECTRICOS										
	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034
SAN SEBASTIAN	15	23	35	48	60	70	78	83	86	88	89
BELLAVISTA	7	12	18	24	30	34	37	38	39	39	40
EL VECINO	14	21	29	39	49	60	70	78	85	89	93
SUCRE	9	14	21	29	36	43	49	53	56	57	58
RAMIREZ DAVALOS	5	8	12	17	21	24	27	28	29	29	30

Sin embargo, no se consideró en el presente estudio debido a limitaciones en el acceso a información técnica detallada y actualizada (Falta de información de las características de las líneas y transformadores). En su lugar, se seleccionó el alimentador 0324, (ver tabla 13) ya que su cobertura abarca tanto zonas urbanas como rurales del cantón, otorgándole un carácter representativo dentro del sistema de distribución. De acuerdo con las proyecciones realizadas, este alimentador concentrará un total estimado de 208 vehículos eléctricos para el año 2034, lo que equivale al 20,78% del total proyectado en el cantón. Además, su área de influencia incluye parroquias como San Blas, Huayna Cápac, Cañaribamba, El Valle, Totoracocha y Monay, donde se observa un crecimiento sostenido en la adopción de movilidad eléctrica. Esta combinación de cobertura territorial y concentración de demanda proyectada convierte al alimentador 0324 en un caso de estudio idóneo para evaluar el impacto técnico de la electromovilidad sobre la red de distribución.

Tabla 14: Parroquias que abarca el alimentador 0324.

PARROQUIAS	PROYECCIONES DE VEHICULOS ELECTRICOS										
	2024	2025	2026	2027	2028	2029	2030	2031	2032	2033	2034
EL VALLE	7	12	18	24	30	34	37	38	39	39	40
TOTORACOCHA	9	14	21	29	36	43	49	53	56	57	58
MONAY	4	5	5	6	6	6	6	6	6	6	6
HUAYNACAPAC	7	12	18	24	30	34	37	38	39	39	40
CAÑARIBAMBA	3	3	5	6	7	9	10	12	14	15	16
SAN BLAS	7	11	15	21	27	32	37	41	44	46	48

El alimentador 0324 es un sistema trifásico que opera a un nivel de tensión de 22 kV. Actualmente, atiende 3.968 puntos de carga, entre usuarios residenciales, comerciales e

industriales. Cuenta con 186 transformadores de media a baja tensión (MT/BT), con una potencia instalada total de 13,525 kVA. Su infraestructura eléctrica comprende aproximadamente 77,21 km de red aérea y 16,80 km de red subterránea en baja tensión, 17,82 km de red aérea y 2,79 km de red subterránea en media tensión, lo que permite una cobertura amplia y diversa dentro del área de influencia del alimentador (Empresa Eléctrica Regional Centro Sur C.A., 2025).

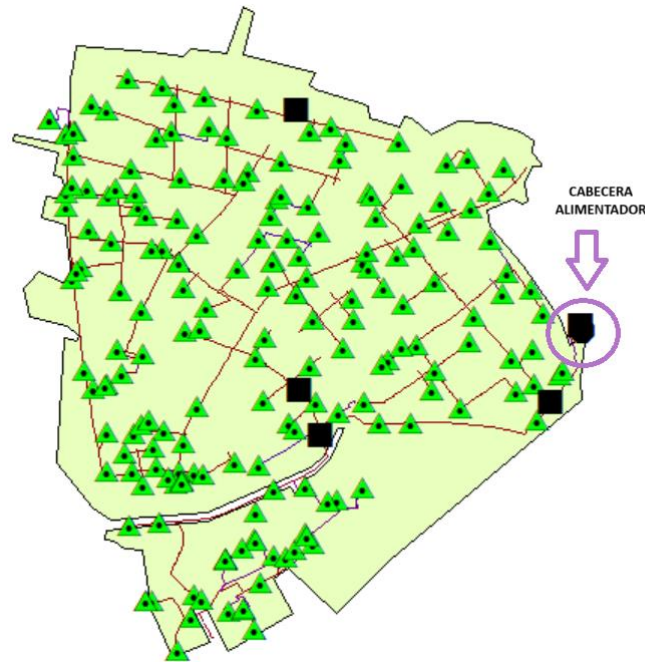


Figura 43: Alimentador 0324 Centro sur.

Fuente: (Empresa Eléctrica Regional Centro Sur C.A., 2025)

### 3.4 MODELADO DEL ALIMENTADOR

Como complemento al análisis eléctrico convencional, se adopta una representación topológica de la red de distribución mediante teoría de grafos. Esta metodología permite evaluar la estructura del sistema enfocándose en aspectos como conectividad, radialidad y robustez, lo cual resulta especialmente relevante frente a la incorporación creciente de nuevas cargas distribuidas, como los vehículos eléctricos (Pagani y Aiello, 2013).

La construcción de esta estructura topológica se realiza utilizando un script en Python, el cual interpreta el archivo maestro de OpenDSS (Master.dss). En este archivo se encuentran definidos todos los elementos eléctricos del sistema. El script identifica las líneas eléctricas mediante las instrucciones New.Line y extrae los nodos de conexión especificados en los atributos bus1 y bus2. Para garantizar la consistencia en la identificación de nodos, se eliminan los sufijos de fase incluidos en los nombres de barra (por ejemplo: .1, .2, .3). Con la información extraída, se genera un grafo no dirigido a través de la biblioteca networkx. En este grafo:

- Cada nodo representa una barra de la red.
- Cada enlace o arista corresponde a una conexión física entre dos nodos, representando una línea eléctrica.

Esta abstracción permite obtener una visión estructural del sistema sin necesidad de incorporar parámetros eléctricos detallados como impedancias, potencias o flujos. El grafo resultante posibilita:

- Verificar la conectividad general de la red.
- Identificar barras terminales o nodos desconectados.
- Detectar componentes aislados.

Adicionalmente, se genera una visualización gráfica del grafo mediante matplotlib, lo que facilita la validación visual de la estructura modelada. Para permitir su uso en plataformas externas de análisis de redes, se exporta el grafo en formato (grapgmil) compatible con herramientas como Gephi o Cytoscape.

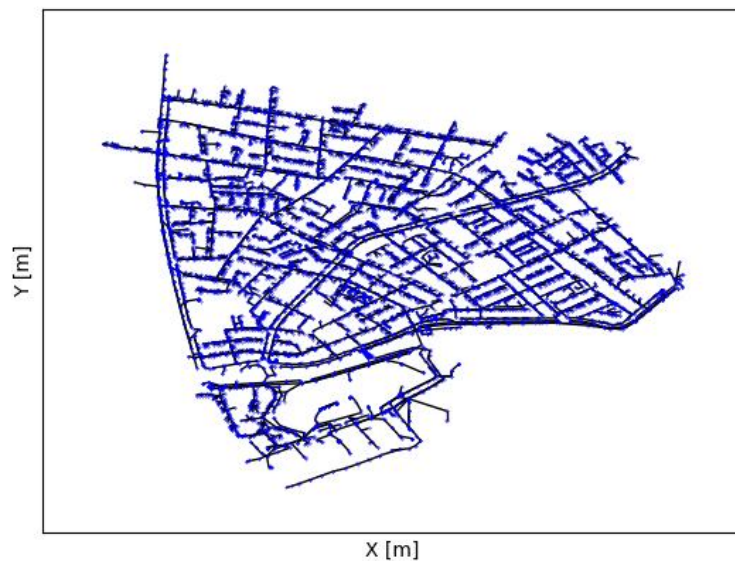


Figura 44: Creación de la red desde Python.

Fuente: (Empresa Eléctrica Regional Centro Sur C.A., 2025)

Aunque este enfoque no considera variables eléctricas propiamente dichas, sí aporta una base sólida para realizar estudios complementarios de tipo estructural. Entre ellos se destacan la identificación de nodos críticos, el análisis de resiliencia ante fallos, y la evaluación de posibles estrategias de expansión, especialmente en el contexto de transformación de la red debido a la movilidad eléctrica.

Una vez verificada la estructura topológica de la red, es necesario complementar esta representación con un modelo eléctrico detallado que permita simular el comportamiento del sistema bajo distintos escenarios operativos. Para ello, se automatiza la generación de los archivos de entrada del entorno OpenDSS, utilizando un segundo script en Python, diseñado para construir el modelo eléctrico a partir de datos estructurados. Este proceso tiene como propósito asegurar la coherencia y eficiencia en la creación de los archivos (.dss) especialmente cuando se trabaja con sistemas de gran tamaño o cuando se requiere evaluar múltiples configuraciones. La automatización permite traducir información técnica como: impedancias, potencias, niveles de tensión, ubicación de cargas y transformadores en archivos compatibles con el motor de simulación. Se generan los siguientes archivos:

- Line.dss: que define las líneas del sistema, incluyendo los buses de conexión, longitudes, parámetros eléctricos y disposición de fases.
- Transformer.dss: que especifica los transformadores, sus niveles de tensión y potencias nominales.
- Load.dss: que contiene las cargas distribuidas, detallando potencias activas y reactivas, factor de potencia y ubicación en la red.
- BusCoords.dss: que incorpora las coordenadas geográficas de los nodos, necesarias para la visualización gráfica del sistema en OpenDSS.
- Master.dss: que actúa como archivo principal y agrupa los demás archivos mediante instrucciones Redirect, facilitando la ejecución completa del modelo.

Una vez generado y estructurado el modelo eléctrico del sistema, se procede a la simulación de distintos escenarios operativos mediante un script desarrollado en Python. Este script, que emplea la biblioteca opendssdirect, permite ejecutar automáticamente el modelo en OpenDSS.

Insertar dinámicamente cargas correspondientes a (VE) y evaluar el comportamiento de la red bajo diferentes condiciones de penetración de esta nueva demanda. La función principal del script permite simular distintos escenarios, comenzando con una red inicial, un crecimiento anual del parque vehicular eléctrico entre los años 2025 y 2032 con una tasa de incremento del 5%, y finalmente un escenario límite en el que la carga total no supera los 5 MW en el sistema. Durante la ejecución de cada escenario, el script recopila información clave del sistema, entre ellas:

- Niveles de tensión en todas las barras, diferenciando entre media tensión (MT) y baja tensión (BT).
- Potencias activas y reactivas de todas las cargas, incluyendo las asociadas a VE.
- Corrientes máximas en líneas.
- Porcentaje de cargabilidad en transformadores.
- Factores de potencia de las cargas conectadas.

El modelo identifica además los extremos de tensión (mínimos y máximos) en cada nivel de tensión y el nodo correspondiente en el que se presentan. Estas métricas permiten detectar posibles problemas técnicos como caídas de tensión fuera del rango normativo, sobrecarga de transformadores o degradación en el factor de potencia.

### **3.5 CARACTERÍSTICAS DE LA CARGA**

La carga asociada al vehículo eléctrico se considera como una carga trifásica, con una demanda aparente de aproximadamente 1.6 kVA. Esta corresponde a una potencia activa de 1.52 kW y una potencia reactiva cercana a 0.50 kvar, lo que refleja un factor de potencia en el orden de 0.95. Estas características representan un perfil de consumo eficiente y estable, adecuado para estudios de impacto en redes de distribución eléctrica

### **3.6 ANÁLISIS DE RESULTADOS**

La Figura 45 presenta la variación de la demanda activa total en función del nivel de penetración de VE en la red de distribución modelada. Se analizan cinco escenarios: uno inicial (sin vehículos eléctricos), un escenario base con 110 VE (2027) (Sánchez Llor, 2022), proyecciones para los años 2028 y 2032 basadas en una tasa de crecimiento anual del 5 %, y una capacidad límite del

alimentador permitido de 5 MW. En condiciones sin carga de los VE, la demanda del sistema se encuentra en 4 MW. La incorporación de 110 VE en el escenario original se incrementa a 4,152 MW, lo cual representa un aumento del 3.8 % con respecto al estado inicial. A medida que se proyecta el crecimiento anual, la carga alcanza 4,174 MW en el año 2028 y 4,212 MW en 2032, evidenciando una tendencia sostenida al alza. El escenario límite año 2072 con 658 VE, simula el punto de saturación del sistema, donde la incorporación de nuevos VE deja de ser viable sin exceder el umbral máximo de 5 MW establecido para este estudio. En dicho escenario, se alcanza una carga total de 5000.00 kW, lo que implica un incremento del 25 % en la potencia activa con respecto a las condiciones actuales.

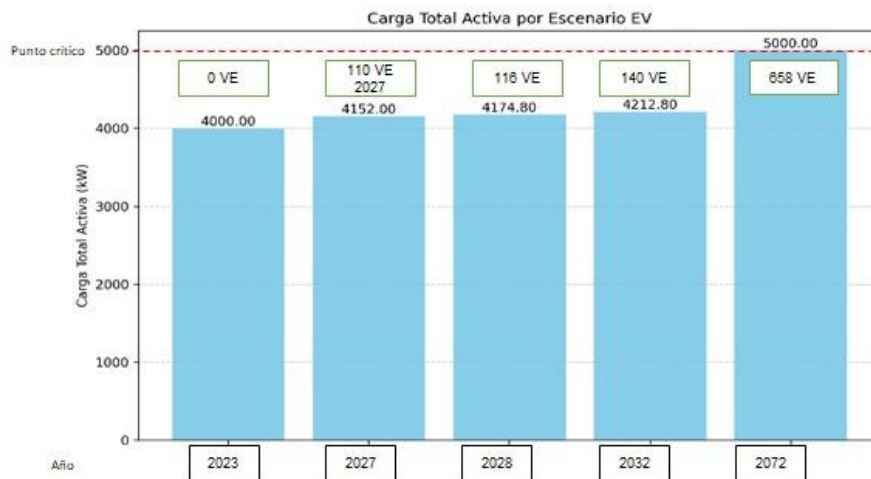


Figura 45: Carga activa de los VE.

La Figura 46 presenta la distribución de los niveles de tensión por barra del sistema de distribución, expresados en valores por unidad (p.u.), para cinco escenarios progresivos de electrificación vehicular: estado inicial (sin VE), escenario base con 100 vehículos eléctricos (2027), proyecciones para los años 2028 y 2032, y un escenario límite correspondiente a la carga máxima simulada de 5 MW.

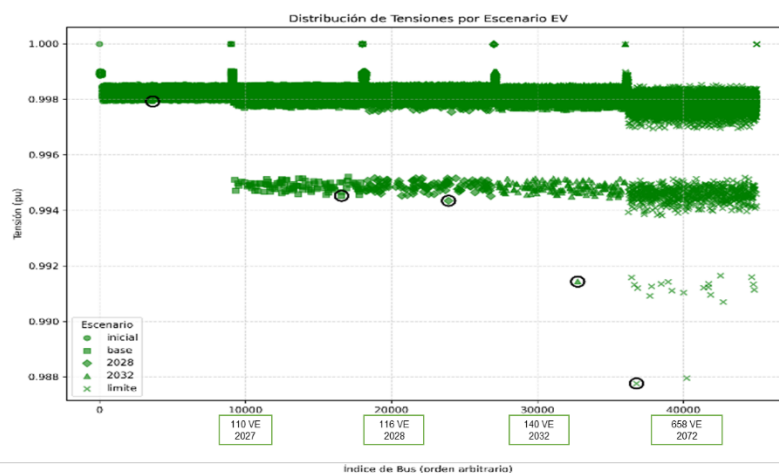


Figura 46: Distribución de tensiones por escenarios.

Esta representación gráfica permite visualizar cómo se modifican los perfiles de tensión en la red conforme se incrementa la carga total del sistema debido a la penetración de VE. Se distinguen claramente dos grupos de nodos: los de baja tensión (BT), donde se observan las principales variaciones. En el escenario base con 110 VE (2027), la tensión mínima registrada es de 0.995 p.u., lo cual refleja un comportamiento estable en la red frente a una incorporación moderada de carga adicional. En 2028 con 116 VE, el valor alcanza los 0.994. El año 2032 con 140 VE, la tensión mínima en dicho año alcanza los 0.992 p.u., confirmando que, bajo las condiciones modeladas, la red aún conserva márgenes adecuados de operación sin manifestar problemas de regulación. El escenario límite en 2072 con 658 VE, correspondiente a la máxima incorporación de VE antes de alcanzar los 5 MW de carga total activa, muestra un descenso adicional en el perfil de tensión. En este caso, la tensión mínima registrada en barras de baja tensión es de 0.988 p.u. Si bien este valor continúa dentro del rango permitido por la Regulación ARCERNR-002/20 (que establece  $\pm 8\%$  sobre el valor nominal para redes de BT, equivalente a un rango entre 0.92 y 1.08 p.u.) conforme a la Regulación Nro. ARCERNR 002/20 (Codificada) (Agencia de Regulación y Control de Energía y Recursos Naturales No Renovables (ARCERNR), 2023).

La Figura 47 muestra la distribución del factor de potencia (PF) en cada una de las cargas del sistema para distintos niveles de penetración de VE: escenario base (110 VE), proyecciones para los años 2028 y 2032, y el escenario límite correspondiente a una carga total activa de 5 MW. Esta variable es crítica para el análisis del desempeño eléctrico del sistema, ya que un PF bajo implica mayor circulación de potencia reactiva, afectando negativamente la eficiencia de la red y generando sobrecargas en transformadores y líneas.

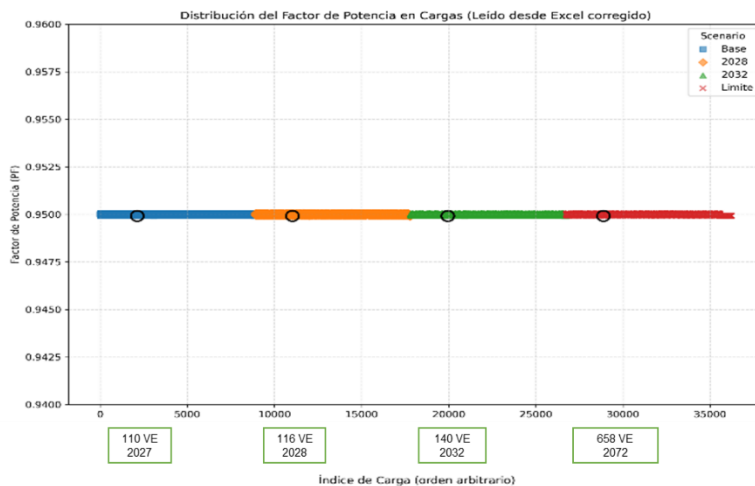


Figura 47: Factor de potencias s.

Como se puede observar, el factor de potencia en todos los escenarios se mantiene estable alrededor de 0.95, valor típico de operación para cargas residenciales o comerciales sin corrección de reactivos. No se registran variaciones significativas entre escenarios, ya que los vehículos eléctricos modelados se conectan con un perfil de carga específico que incorpora un PF de 0.95, lo cual se mantiene constante a lo largo del crecimiento proyectado.

La Figura 48 muestra la evolución de la carga activa (kW) y reactiva (kvar) del sistema de distribución modelado, para el período comprendido entre los años 2025 y 2032. Esta representación se basa en la proyección del crecimiento del parque vehicular eléctrico, considerando una tasa de incremento anual del 5 %, aplicada sobre una carga base de 110 VE iniciales.

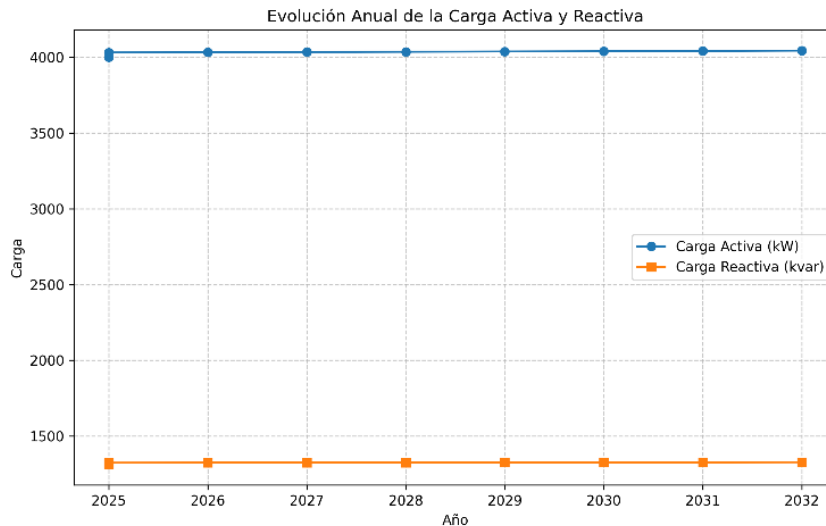


Figura 48: Evolución anual de la carga activa y reactiva.

Los resultados de la carga activa, la cual pasa de 4 MW en 2025 a aproximadamente 4,212 MW en 2032, reflejando un aumento del orden del 5.3 % en ocho años. Por su parte, la carga reactiva también muestra una ligera variación, con un porcentaje de crecimiento de 5.75% aumentando de 1330.00 kvar a 1406.40 kvar en el mismo período. Este comportamiento se debe a la incorporación de VE, cada uno modelado con una potencia reactiva de aproximadamente 0.50 kvar, lo cual se mantiene constante por unidad instalada. La relación entre ambas curvas evidencia que el perfil del factor de potencia agregado no se ve significativamente alterado a lo largo del tiempo. La información presentada en esta figura permite identificar que, bajo un crecimiento gradual de la movilidad eléctrica y manteniendo las condiciones técnicas de conexión actuales, la red mantiene un comportamiento eléctrico estable. No obstante, la pendiente ascendente de ambas curvas indica que el sistema podría requerir ajustes futuros, en especial si se aceleran los planes de adopción de VE o si se incorporan nuevas cargas sensibles de manera simultánea.

### 3.7 CONCLUSIONES

El análisis desarrollado permite comprender con mayor claridad el estado actual y las proyecciones futuras de la movilidad eléctrica en el cantón Cuenca. A partir de una estimación proporcional basada en la flota vehicular provincial, se determina que, para el año 2024 circulan 114 vehículos eléctricos en el cantón, distribuidos de manera coherente con la densidad poblacional de las parroquias urbanas y rurales. Este diagnóstico refleja una concentración inicial de la electromovilidad en zonas urbanas, donde las condiciones de infraestructura y acceso energético son más favorables. Desde el ámbito normativo y estratégico, se identifica un ecosistema institucional propicio para el desarrollo de esta tecnología. A nivel nacional,

instrumentos como la Ley Orgánica de Eficiencia Energética, la Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial, y las resoluciones tarifarias emitidas por el regulador ARCONEL, establecen incentivos arancelarios, tarifas diferenciadas para el servicio de carga y condiciones técnicas que favorecen la adopción progresiva de vehículos eléctricos. En el plano local, destaca la implementación del Plan de Electromovilidad de Cuenca, que propone metas claras como alcanzar una participación del 5 % de vehículos eléctricos en el parque automotor cantonal, junto con la habilitación de infraestructura pública y privada de recarga, promoviendo así un enfoque multisectorial y sostenible. La proyección del crecimiento vehicular se realizó mediante el modelo de difusión de Bass, el cual permite capturar el comportamiento de adopción tecnológica en función de factores de innovación e imitación. Con base en esta metodología, se estima que para el año 2034 la flota de vehículos eléctricos en Cuenca superará las 1.000 unidades, lo que evidencia una transición estructural hacia un modelo de transporte más limpio. Esta dinámica plantea retos concretos para la planificación urbana, vial y energética, que requieren una articulación efectiva entre instituciones, actores del sector eléctrico y la ciudadanía.

Desde el punto de vista eléctrico, el análisis técnico realizado con el software OpenDSS permite evaluar el comportamiento de la red de distribución ante distintos niveles de penetración de carga asociada a vehículos eléctricos. Se simularon cinco escenarios: un caso inicial sin carga adicional, un escenario base con 110 VE, proyecciones al 2028 y 2032 con crecimiento del 5 % anual, y un escenario límite condicionado a una carga total activa de 5 MW. En todos los casos, los resultados muestran que los principales parámetros eléctricos: como la carga total, el perfil de tensiones y el factor de potencia se mantienen dentro de los márgenes normativos establecidos por la Regulación ARCERNR-002/20 para redes de baja tensión. La tensión mínima registrada en los escenarios simulados no desciende por debajo de 0.9877 p.u., lo cual indica un comportamiento estable, sin indicios de afectación crítica en la calidad del suministro. El factor de potencia global permanece constante en aproximadamente 0.95, reflejando que la incorporación progresiva de vehículos eléctricos, bajo condiciones técnicas bien definidas, no genera una carga reactiva excesiva ni compromete la eficiencia del sistema.

Adicionalmente, la evolución anual de la carga activa y reactiva evidencia una tendencia ascendente moderada y técnicamente predecible, sin exceder los límites establecidos para la operación segura del sistema modelado. En conjunto, estos hallazgos técnicos permiten concluir que la red eléctrica de distribución del cantón Cuenca, bajo el modelo y supuestos considerados, es capaz de soportar un proceso gradual de electrificación vehicular hasta el horizonte 2032 sin requerir reforzamientos inmediatos. No obstante, al aproximarse a los niveles máximos de carga, el sistema comienza a mostrar señales que podrían comprometer su estabilidad si no adoptan medidas de refuerzo. En este contexto, se identifica la necesidad de implementar acciones como el recalcular y sustitución de calibres de conductores en tramos sobrecargados, ampliación de la capacidad de los transformadores, instalación de sistemas de regulación de voltaje, compensación de potencia reactiva, y el desarrollo de estrategias de gestión activa de la demanda.

Desde la perspectiva de la planificación energética, estos resultados deben ser integrados en los planes de expansión y modernización de la red de distribución del cantón. Esto implica vincular directamente los escenarios de crecimiento de la electromovilidad con la programación de

inversiones, priorización de obras, y el establecimiento de políticas tarifarias, técnicas y regulatorias que anticipen la evolución de la demanda. Además, es crucial que se articule una coordinación efectiva entre CENTROSUR, el gobierno local, responsables de la movilidad y los usuarios finales, para asegurar que la transición hacia una movilidad sostenible esté acompañada por un sistema eléctrico preparado, resiliente y tecnológicamente actualizado.

### **3.8 RECOMENDACIONES**

El análisis desarrollado confirma que Cuenca se encuentra en una etapa temprana, pero con un panorama alentador en su camino hacia la movilidad eléctrica. Con una estimación actual de 114 vehículos eléctricos y una proyección que supera las 1000 unidades para el año 2034, se vislumbra una evolución constante respaldada por un marco normativo favorable y una progresiva aceptación social. La aplicación del modelo de Bass permite anticipar un cambio importante en las dinámicas de movilidad urbana, lo que plantea la necesidad de fortalecer la planificación de la infraestructura eléctrica y de transporte. El entorno regulatorio actual ofrece condiciones propicias para el desarrollo de esta transición, aunque su éxito dependerá del impulso continuo de políticas públicas, del acceso justo a tecnologías sostenibles y de una ciudadanía cada vez más consciente de los beneficios de este nuevo modelo de transporte. En este escenario, Cuenca cuenta con las condiciones necesarias para avanzar hacia un sistema de movilidad más limpio, eficiente y alineado con los principios de desarrollo sostenible.

Desde una perspectiva técnica, se considera fundamental fortalecer la planificación eléctrica del cantón mediante un enfoque prospectivo que anticipe los efectos del crecimiento progresivo de vehículos eléctricos sobre la red de distribución. La incorporación de esta nueva carga requiere una evaluación permanente de los perfiles de tensión, el comportamiento de los transformadores, las pérdidas técnicas y la calidad del suministro, considerando distintos escenarios de crecimiento. Aunque los resultados obtenidos mediante simulación muestran que la red mantiene un desempeño aceptable en todos los casos analizados, es necesario mantener un monitoreo técnico constante, en especial en zonas con alta densidad poblacional o mayor concentración de carga proyectada.

La implementación de estrategias de gestión de demanda, tales como la programación horaria de carga, sistemas de respuesta ante demanda y esquemas tarifarios diferenciados, permitirá mitigar picos de consumo y evitar sobrecargas localizadas. Asimismo, se recomienda considerar soluciones de infraestructura adaptativa, como refuerzos en tramos críticos de la red, incorporación de dispositivos de compensación de potencia reactiva, almacenamiento energético y, en el mediano plazo, generación distribuida en sectores estratégicos.

Desde el plano institucional, se sugiere continuar consolidando el marco regulatorio relacionado con la conexión técnica de cargadores, la estandarización de equipos y la interoperabilidad entre actores del sistema. A nivel local, la ejecución efectiva del Plan de Electromovilidad de Cuenca dependerá de la articulación interinstitucional entre el municipio, la Empresa Eléctrica Regional Centro Sur, organismos de planificación urbana y operadores privados. Este trabajo conjunto debe permitir el seguimiento de indicadores de avance, la priorización de inversiones y la incorporación de nuevos instrumentos de gestión. Para que esta transición sea inclusiva y

sostenible, es fundamental garantizar el acceso equitativo a tecnologías limpias mediante incentivos económicos, financiamiento preferencial y programas de educación ciudadana. La vigilancia tecnológica, el análisis de tendencias globales y la constante revisión de experiencias internacionales exitosas permitirán asegurar que Cuenca mantenga una trayectoria firme y planificada hacia una movilidad sostenible, resiliente y energéticamente eficiente.

## REFERENCIAS BIBLIOGRÁFICAS

### BIBLIOGRAFÍA

- Agencia de Regulación y Control de Electricidad (ARCONEL). (2024). *Pliego Tarifario para los Proveedores del Servicio de Carga de Energía a Vehículos Eléctricos*. Quito.
- Agencia de Regulación y Control de Electricidad (ARCONEL). (2024). *Resolución Nro. ARCONEL-024/2024*. Agencia de Regulación y Control de Electricidad (ARCONEL), Quito.
- Agencia de Regulación y Control de Energía y Recursos Naturales No Renovables (ARCERNNR). (2023). *Regulación Nro. ARCERNNR 002/20: Calidad del servicio de distribución y comercialización de energía eléctrica*. Quito.
- Agencia Nacional de Regulación y Control de Electricidad (ARCONEL). (2023). *Atlas del Sector Eléctrico Ecuatoriano 2023*. Quito.
- Albuquerque, F. D., Maraga, M. A., Chowdhury, R., Mauga, T., & Alzard, M. (2020). Greenhouse gas emissions associated with road transport projects: current status, benchmarking, and assessment tools. (E. B.V., Ed.) *Transportation Research Procedia*, 2018–2030. <https://doi.org/https://doi.org/10.1016/j.trpro.2020.08.261>
- Asamblea Nacional del Ecuador. (2014). *Ley Orgánica de Transporte Terrestre, Tránsito y Seguridad Vial*. Quito: Registro Oficial del Ecuador. <https://www.turismo.gob.ec/wp-content/uploads/2016/04/LEY-ORGANICA-DE-TRANSPORTE-TERRESTRE-TRANSITO-Y-SEGURIDAD-VIAL.pdf>
- Asamblea Nacional del Ecuador. (2019). *Ley Orgánica de Eficiencia Energética*. Quito: Registro Oficial del Ecuador. [https://www.recursoyenergia.gob.ec/wp-content/uploads/2022/12/20190319-S\\_R\\_O\\_449\\_19\\_MARZO\\_LEY-ORGANICA-DE-EFICIENCIA-ENERGETICA.pdf](https://www.recursoyenergia.gob.ec/wp-content/uploads/2022/12/20190319-S_R_O_449_19_MARZO_LEY-ORGANICA-DE-EFICIENCIA-ENERGETICA.pdf)
- Asociación de Empresas Automotrices del Ecuador (AEADE). (2023). *Informe sobre el sector automotor en Ecuador*. Asociación de Empresas Automotrices del Ecuador. Quito: AEADE.
- Batalla Bejerano, J., & Jové Llopis, E. (2019). Mercados eléctricos ante la transición energética: diseño y retos de futuro. (I. d. Fiscales, Ed.) *Presupuesto y Gasto Público*, 97, 177–192.
- Batún, J., Cool, R., Guerrero, A., & Pantí, H. (2023). Estimación de la función de Gompertz considerando la correlación entre los errores. Dos aplicaciones: COVID-19 y Biogás. *Medical Journal of The Islamic Republic of Iran*, 21(2).
- Cárdenas Bravo, J., & Machuca Chabla, P. (2024). *Capacidad de alojamiento de vehículos eléctricos y generación fotovoltaica para autoabastecimiento en la red de distribución de la Empresa Eléctrica Regional Centro Sur C.A.* Universidad de Cuenca, Cuenca. <https://dspace.ucuenca.edu.ec/bitstreams/dfb37e8f-6c3e-4d73-9db2-0381f5530eb5/download>

- Cardenas Cobos, J. G., & Jaramillo León, B. D. (2017). *Modelación y simulación de las redes de baja tensión de la Empresa Eléctrica Regional Centro Sur C.A.* Universidad de Cuenca, Cuenca.
- Ceballos Delgado, J. E., Caicedo Bravo, E., & Ospina Arango, S. (2016). Una Propuesta Metodológica para Dimensionar el Impacto de los Vehículos Eléctricos sobre la Red Eléctrica. *Revista Ingeniería*.
- Collado Latorre, J. C., & Navarro Jover, J. M. (2013). *ArcGIS 10: Prácticas paso a paso*. Valencia: Universitat Politècnica de València.  
[https://riunet.upv.es/bitstream/handle/10251/70699/TOC\\_6098\\_01\\_01.pdf?sequence=5&isAllowed=y](https://riunet.upv.es/bitstream/handle/10251/70699/TOC_6098_01_01.pdf?sequence=5&isAllowed=y)
- Concejo Cantonal de Cuenca. (2021). *Ordenanza para la Promoción y Fortalecimiento de la Movilidad Activa en el Cantón Cuenca*. Cuenca: Gaceta Oficial del Cantón Cuenca.
- Consejo Nacional Electoral del Ecuador. (2024, 11 15). *Consejo Nacional Electoral del Ecuador*. Consejo Nacional Electoral del Ecuador: <https://www.cne.gob.ec/>
- Empresa Eléctrica Regional Centro Sur C.A. (2025, 3 14). *Empresa Eléctrica Regional Centro Sur C.A.* Empresa Eléctrica Regional Centro Sur C.A.: <https://www.centrosur.gob.ec>
- Empresa Electrica Regional CENTROSUR C.A. (2022). *PLAN ESTRATÉGICO 2022-2025*. Cuenca: Empresa Electrica Regional CENTROSUR C.A. <https://www.centrosur.gob.ec>
- Empresa Electrica Regional CENTROSUR C.A. (2022). *PLAN ESTRATÉGICO 2022-2025*. Cuenca: Empresa Electrica Regional CENTROSUR C.A. <https://www.centrosur.gob.ec/wp-content/uploads/pdfsvarios/5.1-Plan-Estrategico-2022-2025.pdf>
- Frías, P., & Jaime, R. (2023). *VEHÍCULO ELÉCTRICO: SITUACIÓN ACTUAL Y PERSPECTIVAS FUTURAS*. Instituto de Investigación Tecnológica IIT ICAI, Universidad Pontificia Comillas, Madrid.
- GAD Municipal del Cantón Cuenca. (2024). *Diagnóstico PDOT-act. 2023 - 2027*. GAD Municipal del Cantón Cuenca, Cuenca.  
<https://www.cuenca.gob.ec/sites/default/files/planificacion/Diagn%C3%B3stico%20PDOT-act.%202023%20-%202027.pdf>
- González López, V. (2017). *Análisis del Modelo de Bass. Ampliación mediante el reemplazo, la innovación, la inversión y el atractivo del producto*. Universidad de A Coruña.
- Herrera Basabe, P. T., Hernández Areu, O., García del Sol, D., & Díaz Alfonso, E. (2024). Medición de parámetros energéticos durante el proceso de carga de vehículos eléctricos. (C. Universidad Tecnológica de La Habana "José Antonio Echeverría", Ed.) *Ingeniería Energética*, 45(3).  
<https://doi.org/https://rie.cujae.edu.cu/index.php/RIE/index>
- Karmaker, A. K., Prakash, K., Siddique, M. N., Hossain, M. A., & Pota, H. (2024). Electric vehicle hosting capacity analysis: Challenges and solutions. (E. Ltd., Ed.) *Renewable and*

- Sustainable Energy Reviews*, 189, 1-15.  
<https://doi.org/https://doi.org/10.1016/j.rser.2023.113916>
- Lazo Paz, D. A. (2021). *Estimación del Modelo de Difusión de Bass por Mínimos Cuadrados No Lineales*. Universidad Nacional de San Agustín de Arequipa, Arequipa.
- Martínez Gutiérrez, J. A. (2013). Vehículo eléctrico: análisis y prospectiva de factores tecnológicos y económicos. *Vehículo eléctrico: análisis y prospectiva de factores tecnológicos y económicos*. Universidad de Valladolid.
- Marumo, N., Okuno, T., & Akiko, T. (2023). Majorization-minimization-based Levenberg–Marquardt method for constrained nonlinear least squares. *Computational Optimization and Applications* .
- Marzal Varó, A., Gracia Luengo, I., & García Sevilla, P. (2014). *Introducción a la programación con Python 3*. Publicacions de la Universitat Jaume I.  
<https://doi.org/http://dx.doi.org/10.6035/Sapientia93>
- Matplotlib. (2025, 3 18). *Pyplot tutoria*. Pyplot tutoria:  
<https://matplotlib.org/stable/tutorials/pyplot.html>
- Monatnía Portillo, C. V. (2015). *Aplicación de Estacionalidad al Modelo Discreto de Bass propuesto por D. Satoh para la predicción de ventas de nuevos productos tecnológicos*. Facultad Politécnica, Universidad San Lorenzo, San Lorenzo.
- Morejón-Monteros, P., Banegas-Arias, D., & Ochoa-Correa, D. (2024, Junio 30). Impacto de la carga lenta de vehículos eléctricos en la calidad de energía de la red de distribución: Una prospección literaria. (U. E. Amazónica, Ed.) *CEDAMAZ*, 14(1), 69-79.  
<https://doi.org/https://doi.org/10.54753/cedamaz.v14i1.2220>
- Movilidad Sostenible para Todos. (2021). *Movilidad eléctrica sostenible: Componentes esenciales y recomendaciones de políticas*. Washington D.C.: Movilidad Sostenible para Todos (SuM4All). <https://doi.org/https://www.sum4all.org>
- Municipio de Cuenca – Dirección General de Movilidad (DGM) y EMOV EP. (2023). *Plan de Electromovilidad de Cuenca*. Cuenca: Municipio de Cuenca / EMOV EP.
- NumPy. (2025, 3 18). *NumPy*. NumPy: <https://numpy.org/doc/>
- Pagani, G., & Aiello, M. (2013). The Power Grid as a Complex Network: a Survey. 2688-2700.
- Palacios, R. (2020). Reconfiguración óptima de redes de distribución que contienen generación distribuida hidroeléctrica, aplicando un algoritmo de optimización basado en colonia de hormigas. *RECONFIGURACIÓN ÓPTIMA DE REDES DE DISTRIBUCIÓN QUE CONTIENEN GENERACIÓN DISTRIBUIDA HIDROELÉCTRICA, APLICANDO UN ALGORITMO DE OPTIMIZACIÓN BASADO EN COLONIA DE HORMIGAS*. ESCUELA POLITÉCNICA NACIONAL, Quito.
- Pandas. (2025, 3 18). *Pandas*. Pandas: <https://pandas.pydata.org/>

- Patiño Chitacapa, A. (2023). *Estimación de la capacidad de alojamiento de generación solar fotovoltaica a través de una penetración masiva a nivel de consumidores en redes de distribución mediante simulación de flujos de carga con series de tiempo*. Universidad Católica de Cuenca, Cuenca. <https://dspace.ucuenca.edu.ec/>
- Perdomo, M. M., Manassero, U., & Vega, J. R. (2022, Junio). Impacto de los vehículos eléctricos sobre la red de distribución: Análisis bajo distintos modos de operación. (U. d. (Uruguay), Ed.) *Memoria Investigaciones en Ingeniería*, 22, 40-50. <https://doi.org/https://doi.org/10.36561/ING.22.5>
- Ramos Sarango, J. F. (2024). *Electromovilidad: Oportunidades y condiciones para su desarrollo en el Ecuador. Análisis de la normativa y regulaciones necesarias para la incorporación de la electromovilidad*. Escuela Politécnica Nacional, Quito. <https://bibdigital.epn.edu.ec/>
- Reinemer, J. M. (2018). *Evaluación de factores socioeconómicos y técnicos que afectan la aplicación del vehículo eléctrico*. Universidad de la Sabana, Chía, Colombia.
- Reyes Campaña, G. G., Guanuche Larco, D. J., Pulles Tinoco, S. A., & Aguirre Stoica, M. S. (2021). Estudio de la percepción de vehículos eléctricos en la ciudad de Quito. *Dominios de las Ciencias*, 7(5), 937-958.
- Sánchez Loor, L. I. (2022). *Modelo para la Proyección de Vehículos Eléctricos Particulares en el Ecuador y sus Requerimientos de Demanda*. Escuela Politécnica Nacional, Quito.
- SciPy. (2025, 3 18). *SciPy optimize module*. SciPy optimize module: <https://docs.scipy.org/doc/scipy/reference/optimize.html>
- Vélez Sanchez, J. G. (2017). *Análisis y estimación de la demanda eléctrica con la implementación de vehículos eléctricos conectados a una red de distribución en Cuenca y el Ecuador*. Universidad de Cuenca, Cuenca. <https://dspace.ucuenca.edu.ec/>
- Vera Silva, C. D., & Marulanda Guerra, A. R. (2018). OpenDSS aplicado al modelamiento y simulación dinámica de microrredes. *Visión Electrónica*, 12(2), 149-161. <https://doi.org/https://doi.org/10.14483/22484728.13994>
- Villegas Aldeán, M., & Trelles Lituma, G. (2023). *Capacidad de alojamiento con generación distribuida en alimentadores de MT de la Empresa Eléctrica CENTROSUR*. Universidad de Cuenca, Cuenca. <https://rest-dspace.ucuenca.edu.ec/server/api/core/bitstreams/a5b00571-5a32-446d-a879-422ec14149fe/content>
- Weissmann, V. (2008). Difusión de nuevas tecnologías y estimación de la demanda de nuevos productos: un análisis comparativo entre Argentina y EE.UU. *Palermo Business Review*, 1, 5–18. <https://www.palermo.edu/economicas/publicaciones.html> si

## ANEXOS

### ANEXO 1 PROYECCIÓN DE VE BELLAVISTA

```
import numpy as np
from scipy.optimize import curve_fit
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error, r2_score
import pandas as pd

# Datos históricos
años = np.array([2017, 2018, 2019, 2020, 2021, 2022, 2023])
datos_historicos = np.array([0, 0, 0, 1, 1, 2, 4])

# Definir el modelo de Bass
def modelo_bass(t, p, q, m):
    return m * (1 - np.exp(-(p + q) * t)) / (1 + (q / p) * np.exp(-(p + q) * t))

# Ajustar los parámetros p, q, m usando curve_fit
def ajustar_parametros_bass(años, datos_historicos):
    tiempo = np.arange(0, len(años)) # Crear un vector de tiempo basado
    en los años
    p0 = [0.01, 0.3, max(datos_historicos)*2] # Valores iniciales para
    p, q y m
    popt, pcov = curve_fit(modelo_bass, tiempo, datos_historicos, p0=p0,
    bounds=(0, [1, 1, max(datos_historicos)*10]))
    return popt # Retorna p, q, m ajustados

# Llamar a la función para ajustar los parámetros
p, q, m = ajustar_parametros_bass(años, datos_historicos)
print(p,q,m)

##### ESPECIFICAR VALORES DE P,Q Y M MANUALMENTE
# p= 0.003
# q=0.005
# m=500

# Calcular predicciones para el período histórico
tiempo_historico = np.arange(0, len(años))
predicciones_historicas = np.round(modelo_bass(tiempo_historico, p, q,
m))

# Calcular MSE y R²
mse = mean_squared_error(datos_historicos, predicciones_historicas)
r2 = r2_score(datos_historicos, predicciones_historicas)
```

```

# Mostrar los valores ajustados y métricas
print(f"Parámetros del modelo:")
print(f"p (coeficiente de innovación) = {p}")
print(f"q (coeficiente de imitación) = {q}")
print(f"m (mercado potencial) = {m}")
print("\nMétricas de ajuste:")
print(f"MSE = {mse:.4f}")
print(f"R² = {r2:.4f}")

# Predicciones usando los valores ajustados
tiempo_futuro = np.arange(0, len(años) + 11) # Predecir para los
próximos años
predicciones = np.round(modelo_bass(tiempo_futuro, p, q, m))
print(predicciones)

# Graficar los resultados
plt.figure(figsize=(10, 6))
plt.plot(años, datos_historicos, 'o', label='Datos Históricos')
plt.plot(np.arange(años[0], años[0] + len(tiempo_futuro)), predicciones,
label='Predicción Bass')
plt.xlabel('Año')
plt.ylabel('Adopciones VE')
plt.title(f'Modelo de Bass (MSE={mse:.4f}, R²={r2:.4f})')
plt.legend()
plt.grid(True)
plt.show()

# Crear DataFrame con los resultados
años_prediccion = np.arange(años[0], años[0] + len(tiempo_futuro))
df_resultados = pd.DataFrame({
    'Año': años_prediccion,
    'Predicción': predicciones.astype(int)
})

# Agregar datos históricos
df_resultados['Datos_Históricos'] = np.nan # Inicializar con NaN
df_resultados.loc[df_resultados['Año'].isin(años), 'Datos_Históricos'] =
datos_historicos

# Agregar parámetros y métricas en hojas separadas
with pd.ExcelWriter('resultados_modelo_bass_BellaVista.xlsx') as writer:
    # Hoja 1: Predicciones y datos históricos
    df_resultados.to_excel(writer, sheet_name='Predicciones',
index=False)

    # Hoja 2: Parámetros y métricas

```

```

df_parametros = pd.DataFrame({
    'Parámetro': ['p (coeficiente de innovación)',
                  'q (coeficiente de imitación)',
                  'm (mercado potencial)',
                  'MSE',
                  'R2'],
    'Valor': [p, q, m, mse, r2]
})
df_parametros.to_excel(writer, sheet_name='Parámetros', index=False)

print("\nResultados guardados en 'resultados_modelo_bass.xlsx'")

```

## ANEXO 2 CREACION DE GRAFO

```

from osgeo import ogr
import networkx as nx
import pandas as pd
import matplotlib.pyplot as plt

# import pyproj
# from scipy.spatial import distance

from graph_functions import *
# Variables
ruta_gdb = r"C:\Users\Usuario\Desktop\UNIVERSIDAD CATOLICA\NOVENO
CICLO\TESIS\MAPAS POR PARROQUIAS\TESIS\EERCS_06012025.gdb"

# feeder = '040021'
# Ntrafo = '80303'
feeder = '0500030V04'
# feeder = '0500030V03'

# Se debe copiar los OBJECTID EN MIOID el las capas Tramo,
PuestoTransfDistribucion y Luminarias
# Read geodatabase and feature class
db = ogr.Open(ruta_gdb,0)
OHLine_MV = db.GetLayer("TramoDistribucionAereo") # Overhead Line
UGLine_MV = db.GetLayer("TramoDistribucionSubterraneo") # Undergorund
Line
OHLine_LV = db.GetLayer("TramoBajaTensionAereo") # Overhead Line
UGLine_LV = db.GetLayer("TramoBajaTensionSubterraneo") # Undergorund
Line
Luminary = db.GetLayer("Luminaria")
Structure = db.GetLayer("CATALOGOESTRUCTURA")
DistTransf = db.GetLayer("PuestoTransfDistribucion") # Distribution
Transformers
LoadPoint = db.GetLayer("PuntoCarga")
Conexion = db.GetLayer("CONEXIONCONSUMIDOR")

```

```

Attributes = db.GetLayer("ATRIBUTOSCONSUMIDOR")
Protect = db.GetLayer("PuestoProteccionDinamico")
Layer = DistTransf

# print_layer_names(db)
# Llamar a la función para imprimir los nombres de los campos de una capa
específica
print_layer_fields(Layer)
# Imprimir elementos
# print_layer_elements(Layer)
G = nx.Graph() # Create an empty graph structure
fields = ['MIOID', 'ALIMENTADORID', 'LONGITUDSISTEMA',
"CODIGOCONDUCTORFASE", "CODIGOCONDUCTORNEUTRO", "SUBTIPO", 'FASECONEXION',
'VOLTAJE', 'CONFIGURACIONCONDUCTORES']
lyr_lines(G, OHLine_MV, 'MV_OH', 'ALIMENTADORID', feeder, fields)
lyr_lines(G, UGLine_MV, 'MV_UG', "ALIMENTADORID", feeder, fields)
print('Edges MV created')

fields = ['MIOID', 'ALIMENTADORID', 'PUESTOTRANSFDISTOBJECTID',
"LONGITUDSISTEMA", "CODIGOCONDUCTORFASE", "CODIGOCONDUCTORNEUTRO",
"SUBTIPO", 'FASECONEXION', 'VOLTAJE', 'SECUENCIAFASE',
'CONFIGURACIONCONDUCTORES']
lyr_lines(G, OHLine_LV, 'LV_OH', 'ALIMENTADORID', feeder, fields)
lyr_lines(G, UGLine_LV, 'LV_UG', 'ALIMENTADORID', feeder, fields)
print('Edges LV created')

# Cambiar el nombre del atributo en las aristas
for u, v, attributes in G.edges(data=True):
    if 'MIOID' in attributes:
        attributes['OBJECTID'] = attributes.pop('MIOID')

# Acometidas:
for edge in G.edges():
    if (G.edges[edge]['TYPE'] == 'LV_OH') and (G.edges[edge]['SUBTIPO'] in
[7, 8, 9]):
        G.edges[edge]['TYPE'] = 'LV_OH_A'
# fields = ['ALIMENTADORID', "POTENCIAKVA", "VOLTAJE", "CODIGOESTRUCTURA",
"VOLTAJESECUNDARIO", 'FASECONEXION']
# lyr_trf(G, DistTransf, 'TD', "MIOID", Ntrafo, fields, 'MIOID',
stop_on_match=False)
# print('DistTransf Nodes created')

fields = ["MIOID", 'ALIMENTADORID', "POTENCIAKVA", "VOLTAJE",
"CODIGOESTRUCTURA", "VOLTAJESECUNDARIO", 'FASECONEXION', 'SECUENCIAFASE',
'CODIGOPUESTO', 'TOTALCLIENTES', 'SUMACONSUMO']
lyr_trf(G, DistTransf, 'TD', 'ALIMENTADORID', feeder, fields,
stop_on_match=False)
print('DistTransf Nodes created')

```

```

# Cambiar el nombre del atributo en los nodos
for node, attributes in G.nodes(data=True):
    if 'MIOID' in attributes:
        attributes['OBJECTID'] = attributes.pop('MIOID')
fields = ['ALIMENTADORID', 'GLOBALID', 'NUMEROCLIENTES',
"PUESTOTRANSFDISTOBJECTID", "FASECONEXION"]
lyr_load_points(G, LoadPoint, 'ALIMENTADORID', feeder, fields)
print('Load points information added to the graph.')

# fields = ['ALIMENTADORID', 'GLOBALID', 'NUMEROCLIENTES',
"PUESTOTRANSFDISTOBJECTID", "FASECONEXION"]
# lyr_load_points(G, LoadPoint, "PUESTOTRANSFDISTOBJECTID", Ntrafo,
fields)
# print('Load points information added to the graph.')
# Cambiar el nombre del atributo en los nodos ORIGINALES
for node, attributes in G.nodes(data=True):
    if 'FASECONEXION' in attributes:
        attributes['FASECONEXION_0'] = attributes.pop('FASECONEXION')
    if 'SECUENCIAFASE' in attributes:
        attributes['SECUENCIAFASE_0'] = attributes.pop('SECUENCIAFASE')
# Cambiar el nombre del atributo en las aristas
for u, v, attributes in G.edges(data=True):
    if 'SECUENCIAFASE' in attributes:
        attributes['SECUENCIAFASE_0'] = attributes.pop('SECUENCIAFASE')
def lyr2DataFrame(lyr, fields):
    # Convertir la capa lyr a DataFrame
    lyr_data = []
    for feature in lyr:
        feature_data = {}
        for field in fields:
            feature_data[field] = feature.GetField(field)
        lyr_data.append(feature_data)
    df = pd.DataFrame(lyr_data)
    return df

# CREAMOS DATAFRAME con los nodos que tengan 'GLOBALID' (que tengan
PuntoCarga) -----
data_list = []
# recorro todos los nodos (en busca de nodos que tengan GLOBALID)
for node, attributes in G.nodes(data=True):
    # si tiene el parámetro GLOBALID
    if 'GLOBALID' in attributes:
        data_row = {
            'GLOBALID': attributes['GLOBALID'],
            'NUMEROCLIENTES': attributes['NUMEROCLIENTES']
        }
        data_list.append(data_row)
# df with filtered nodes (with 'GLOBALID', 'NUMEROCLIENTES')
df_g = pd.DataFrame(data_list)

```

```

# CREAMOS DATAFRAME de la capa df_Conexion -----
-----
fields = ['PUNTOCARGAGLOBALID', 'CODIGOUNICO', "SECUENCIAFASE"]
df_Conexion = lyr2DataFrame(Conexion, fields)

# Filtrar las filas en df_Conexion donde 'PUNTOCARGAGLOBALID' coincide con
GLOBALID de df_g
df_merged = pd.merge(df_g, df_Conexion, left_on='GLOBALID',
right_on='PUNTOCARGAGLOBALID', how='left')
del df_merged['PUNTOCARGAGLOBALID']

# CREAMOS DATAFRAME de la capa df_Attributes -----
-----
fields = ['CODIGOUNICO', "POTENCIAACTIVA", "TIPOTARIFA",
"CONSUMOPROMEDIO"]
df_Attributes = lyr2DataFrame(Attributes, fields)

df_merged = pd.merge(df_merged, df_Attributes, left_on='CODIGOUNICO',
right_on='CODIGOUNICO', how='left')

df_merged.rename(columns={'SECUENCIAFASE': 'SECUENCIAFASE_CLIENTE'},
inplace=True)
# Se agrega la información del Dataframe al grafo

for index, row in df_merged.iterrows():
    global_id = row['GLOBALID']
    matching_nodes = [node for node, attrs in G.nodes(data=True) if
attrs.get('GLOBALID') == global_id]
    for node in matching_nodes:
        # Agregar la información del DataFrame al nodo del grafo
        for column in df_merged.columns:
            if column != 'GLOBALID' and column != 'NUMEROCLIENTES':
                # G.nodes[node][column] = row[column]
                if column not in G.nodes[node]:
                    G.nodes[node][column] = [row[column]] # Si la columna
no existe, se crea una lista con el valor
                else:
                    # if not isinstance(G.nodes[node][column], list):
                    #     G.nodes[node][column] = [G.nodes[node][column]]
# Convertir a lista si no lo es
                    G.nodes[node][column].append(row[column]) # Si la
columna ya existe, se agrega el valor a la lista
fields = ['ALIMENTADORID', 'MIOID', "POTENCIA", "CODIGOESTRUCTURA",
"PUESTOTRANSFDISTOBJECTID", 'SECUENCIAFASE']
lyr_load_points(G, Luminary, 'ALIMENTADORID', feeder, fields)

# fields = ['ALIMENTADORID', 'MIOID', "POTENCIA", "CODIGOESTRUCTURA",
"PUESTOTRANSFDISTOBJECTID", 'FASECONEXION']

```

```

# lyr_load_points(G, Luminary, "PUESTOTRANSFDISTOBJECTID", Ntrafo, fields)

# Cambiar el nombre del atributo en los nodos
for node, attributes in G.nodes(data=True):
    if 'FASECONEXION' in attributes:
        attributes['FASECONEXION_LUMINARY'] =
attributes.pop('FASECONEXION')
    if 'MIOID' in attributes:
        attributes['OBJECTID_LUMINARY'] = attributes.pop('MIOID')
    if 'SECUENCIAFASE' in attributes:
        attributes['SECUENCIAFASE_LUMINARY'] =
attributes.pop('SECUENCIAFASE')

# CREAMOS DATAFRAME con los nodos que tengan "CODIGOESTRUCTURA" (que tengan
Luminaria) -----
data_list = []
# recorro todos los nodos (en busca de nodos que tengan "CODIGOESTRUCTURA")
for node, attributes in G.nodes(data=True):
    # si tiene el parámetro "CODIGOESTRUCTURA"
    if "CODIGOESTRUCTURA" in attributes:
        data_row = {
            "CODIGOESTRUCTURA": attributes["CODIGOESTRUCTURA"],
        }
        data_list.append(data_row)
# df with filtered nodes (with "CODIGOESTRUCTURA")
df_g = pd.DataFrame(data_list)

# CREAMOS DATAFRAME de la capa df_Structure -----
-----
fields = ["CODIGOESTRUCTURA", "TIPO"]
df_Structure = lyr2DataFrame(Structure, fields)
# print('df_Structure -----')
# df_Structure.info()
# print(df_Structure)
# print('-----')

# Filtrar las filas en df_Conexion donde 'PUNTOCARGAGLOBALID' coincide con
GLOBALID de df_g
df_merged = pd.merge(df_g, df_Structure, left_on="CODIGOESTRUCTURA",
right_on="CODIGOESTRUCTURA", how='left')

print('df_merged -----')
df_merged.info()
print(df_merged)
print('-----')

# Se agrega la información del Dataframe al grafo
for index, row in df_merged.iterrows():
    global_id = row["CODIGOESTRUCTURA"]

```

```

    matching_nodes = [node for node, attrs in G.nodes(data=True) if
attrs.get("CODIGOESTRUCTURA") == global_id]
    for node in matching_nodes:
        # Agregar la información del DataFrame al nodo del grafo
        for column in df_merged.columns:
            if column != "CODIGOESTRUCTURA":
                G.nodes[node][column] = row[column]

print('Luminary points information added to the graph.')
fields = ['GLOBALID', 'FASECONEXION', 'VOLTAJE']
lyr_disyuntor(G, Protect, 'ALIMENTADORID', feeder, fields, 'SUBTIPO', 1)
print('Protect points information added to the graph.')
# REGRESO ORIGINALES
# Cambiar el nombre del atributo en los nodos
for node, attributes in G.nodes(data=True):
    if 'FASECONEXION_0' in attributes:
        attributes['FASECONEXION'] = attributes.pop('FASECONEXION_0')
    if 'SECUENCIAFASE_0' in attributes:
        attributes['SECUENCIAFASE'] = attributes.pop('SECUENCIAFASE_0')
# Cambiar el nombre del atributo en las aristas
for u, v, attributes in G.edges(data=True):
    if 'FASECONEXION_0' in attributes:
        attributes['FASECONEXION'] = attributes.pop('FASECONEXION_0')
    if 'SECUENCIAFASE_0' in attributes:
        attributes['SECUENCIAFASE'] = attributes.pop('SECUENCIAFASE_0')
see_graph(G)
see_graph_image(G, 0.1)
isolates = list(nx.isolates(G)) # find all isolated nodes

if len(isolates) > 0:
    print('The number of isolated nodes is: ' +
str(nx.number_of_isolates(G)))
    print(isolates) # Return list of isolates in the graph. Isolates
are nodes with no neighbors (degree zero)

    print('The ID of the isolated nodes is')
    for i in isolates:
        print('OBJECTID: '+str(G.nodes[i]['OBJECTID'])+' TYPE:
'+str(G.nodes[i]['TYPE']))
else:
    print('There arent isolated nodes ')
print(nx.is_connected(G))
# iterate through nodes
bus_name = 0
for n in G:
    if 'OBJECTID' not in G.nodes[n]:
        bus_name += 1
        G.nodes[n]['OBJECTID'] = str(bus_name)
    if G.nodes[n]['TYPE'] == 'TD':

```

```

        bus_name += 1
        G.nodos[n]['BUSP'] = str(bus_name) # Nodo
ficticio trafo
for node in G.nodos():
    # Obtener las coordenadas 'x' e 'y' directamente del nombre del nodo
    (que es una tupla)
    x, y = node

    # Asignar las coordenadas como atributos 'pos_x' y 'pos_y' del nodo
    G.nodos[node]['pos_x'] = x
    G.nodos[node]['pos_y'] = y
    lat, lon = convert_coordinates(x, y)
    G.nodos[node]['latitude'] = lat
    G.nodos[node]['longitude'] = lon
# Convertir los valores de los atributos de lista a cadenas
for nodo in G.nodos():
    attributes = G.nodos[nodo]
    for key, value in attributes.items():
        if value is None:
            G.nodos[nodo][key] = "nan" # Convertir None a "nan" como cadena
de texto
        if isinstance(value, list):
            G.nodos[nodo][key] = ','.join(map(str, value)) # Convertir la
lista a una cadena separada por comas

for edge in G.edges(data=True):
    attributes = edge[2] # El tercer elemento es el diccionario de
atributos
    for key, value in attributes.items():
        if value is None:
            attributes[key] = 'nan'
## Guardar el grafo para su lectura en formato graphml
nx.write_graphml(G, f"graph_{feeder}_n.graphml")

print(f"Trafo guardado con nombre graph_{feeder}.graphml")
print("\nAtributos de las aristas SECUENCIAFASE: -----")

graph = G
for edge in graph.edges(data=True):
    attributes = edge[2] # El tercer elemento es el diccionario de
atributos
    if 'SECUENCIAFASE' in attributes and attributes['SECUENCIAFASE'] not
in ['a', 'b', 'c', 'ab', 'bc', 'ac', 'abc']:
        print("Arista: \t", edge[:2]) # Los primeros dos elementos son los
nodos que conecta la arista
        for key, value in attributes.items():
            if value is None:
                print("El atributo '{}' de la arista {} tiene valor
None.".format(key, edge[:2]))

```

```

        else:
            print("\t '{}': {}".format(key, value))
see_graph(G)

```

### ANEXO 3 CREACIÓN DE ARCHIVOS .DSS

```

import networkx as nx
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import math

from graph_functions import *
# G = nx.read_graphml("graph_0500040V07_MV.graphml")
# G = nx.read_graphml("graph_0500040V07_80303.graphml")
# G = nx.read_graphml("graph_0500040V07.graphml")
# G = nx.read_graphml("graph_0500040V07_n.graphml")
G = nx.read_graphml("graph_0500030V04_n.graphml")

posiciones = {nodo: (float(G.nodes[nodo]['pos_x']),
                    float(G.nodes[nodo]['pos_y']))
              for nodo in G.nodes()}

# Dibujar el grafo
# nx.draw(grafo, pos=posiciones, with_labels=True, node_size=200,
node_color='skyblue', font_size=8)

# Agregar etiquetas a las aristas
# etiquetas_aristas = {(nodo1, nodo2): G[nodo1][nodo2]['TYPE'] for (nodo1,
nodo2) in G.edges()}
ax = nx.draw(G, pos=posiciones, with_labels=False, node_size=0.1,
node_color='b', edge_color='gray')

# Mostrar el gráfico
plt.xlabel('Eje X[m]')
plt.ylabel('Eje Y[m]')
plt.title("Red Eléctrica")
plt.show()
isolates = list(nx.isolates(G)) # find all isolated nodes

if len(isolates) > 0:
    print('The number of isolated nodes is: ' +
str(nx.number_of_isolates(G)))
    print(isolates) # Return list of isolates in the graph. Isolates
are nodes with no neighbors (degree zero)

    print('The ID of the isolated nodes is')
    for i in isolates:

```

```

        print('OBJECTID:      '+str(G.nodos[i]['OBJECTID'])+'      TYPE:
'+str(G.nodos[i]['TYPE']))
else:
    print('There arent isolated nodes ')
print(nx.is_connected(G))

if nx.is_connected(G) is False:
    print('*** Guardo Subgrafo Principal ***')
    # Si existen subgrafos sólo guardamos el principal
    # Se muestran las desconexiones
    G2 = save_principal_graph(G)
print("Número de nodos:", nx.number_of_nodes(G))
print("Número de aristas:", nx.number_of_edges(G))
if nx.is_connected(G) is False:
    print("Número de nodos:", nx.number_of_nodes(G2))
    print("Número de aristas:", nx.number_of_edges(G2))
print(nx.is_connected(G))

if nx.is_connected(G) is False:
    print('*** Uniendo Subgrafos ***')
    # Si existen subgrafos sólo guardamos el principal
    G = fix_graph(G)
print("Número de nodos:", nx.number_of_nodes(G))
print("Número de aristas:", nx.number_of_edges(G))
print(nx.is_connected(G))
see_graph(G)
for edge in G.edges(data = True):
    attributes = edge[2]
    if 'LV' in attributes['TYPE'] and attributes['SECUENCIAFASE'] not in
['a', 'b', 'c', 'ab', 'ac', 'bc', 'abc']:
        nodei = edge[0] # Nodo inicial del borde
        nodef = edge[1] # Nodo final del borde
        if G.nodos[nodei]['TYPE'] == 'TD':
            attributes['SECUENCIAFASE'] = G.nodos[nodei]['SECUENCIAFASE']
            continue
        elif G.nodos[nodef]['TYPE'] == 'TD':
            attributes['SECUENCIAFASE'] = G.nodos[nodef]['SECUENCIAFASE']
            continue
        else:
            # buscar aristas vecinas y tomar el atributo 'SECUENCIAFASE'
            que contenga mayor número de caracteres
            vecinos = list(G.neighbors(nodei)) + list(G.neighbors(nodef))
# nodos vecinos
            secuencias = []
            for vecino in vecinos:
                if G.has_edge(nodei, vecino) and 'SECUENCIAFASE' in
G.edges[nodei, vecino]:
                    secuencias.append(G.edges[nodei,
vecino]['SECUENCIAFASE'])

```

```

        elif G.has_edge(vecino, nodei) and 'SECUENCIAFASE' in
G.edges[vecino, nodei]:
            secuencias.append(G.edges[vecino,
nodei]['SECUENCIAFASE'])
            if G.has_edge(nodef, vecino) and 'SECUENCIAFASE' in
G.edges[nodef, vecino]:
                secuencias.append(G.edges[nodef,
vecino]['SECUENCIAFASE'])
            elif G.has_edge(vecino, nodef) and 'SECUENCIAFASE' in
G.edges[vecino, nodef]:
                secuencias.append(G.edges[vecino,
nodef]['SECUENCIAFASE'])

# Eliminar valores 'nan' de secuencias
secuencias = [s for s in secuencias if s != 'nan']

if secuencias:
    print(f'secuencias: {secuencias}')
    print(f'secuencia max: {max(secuencias, key=len)}')
    attributes['SECUENCIAFASE'] = max(secuencias, key=len)
else:
    print(f'TYPE:          {attributes['TYPE']},          OBJECTID:
{attributes['OBJECTID']}')

# arreglo una línea mal codificada encontrada:
# if attributes['OBJECTID'] == 117220:
#     attributes['SECUENCIAFASE'] = 'bc'

# if attributes['OBJECTID'] == 1381901:
#     attributes['VOLTAJE'] = 240

# # Específico para el trafo encontrado
# if (attributes['TYPE'] == 'LV_UG') and (attributes['OBJECTID'] in
[28006, 262830, 262829]):
#     # print('puta')
#     attributes['CODIGOCONDUCTORFASE'] = 'CO00071'
#     attributes['CODIGOCONDUCTORNEUTRO'] = 'CO00071'

for node, attributes in G.nodes(data=True):
    if 'CODIGOUNICO' in attributes:
        secuencias_clientes = []
        secuencias_clientes =
attributes['SECUENCIAFASE_CLIENTE'].split(',')
        secuencia_max = [secuencia for secuencia in secuencias_clientes if
secuencia != 'nan']
        if len(secuencia_max) > 0:
            secuencia_max = max(secuencia_max, key=len)

```

```

if len(secuencia_max) > 0:
    # Recorrer las aristas conectadas al nodo
    for edge in G.edges(node, data=True):
        edge_data = edge[2]
        if 'SECUENCIAFASE' in edge_data:
            # Verificar si la longitud de la secuencia de la arista
            es menor que la secuencia máxima encontrada
            if (len(edge_data['SECUENCIAFASE']) <
len(secuencia_max)):
                print(f"OBJECTID: {edge_data['OBJECTID']}
'SECUENCIAFASE': {edge_data['SECUENCIAFASE']}\tsecuencia_max:
{secuencia_max}")
                edge_data['SECUENCIAFASE'] = secuencia_max
                if edge_data['VOLTAJE'] == 120:
                    edge_data['VOLTAJE'] = 240
# Diccionarios de mapeo de conductores por tipo de fase
codigo_conductor_mono = {
    'C000046': 'C01', 'C000045': 'C01', 'C000079': 'C01', 'C000081': 'C01',
    'C000082': 'C01'
}

codigo_conductor_bi = {
    'C000051': 'C02', 'C000050': 'C02', 'C000049': 'C02', 'C000052': 'C02',
    'C000088': 'C02', 'C000087': 'C02', 'C000089': 'C02', 'C000085': 'C02'
}

codigo_conductor_tri = {
    'C000055': 'C03', 'C000054': 'C03', 'C000053': 'C03', 'C000091': 'C03'
}

# Iterar sobre los bordes del grafo y actualizar los códigos de conductores
fase según el diccionario de mapeo
for edge in G.edges(data=True):
    edge_data = edge[2]
    if ('LV_OH' in edge_data['TYPE']) or ('LV_UG' in edge_data['TYPE']):
# Ahora también incluye LV_UG
        codigo_actual = edge_data['CODIGOCONDUCTORFASE']
        if (len(edge_data['SECUENCIAFASE']) == 1):
            if (codigo_actual not in codigo_conductor_mono) and
(codigo_actual in codigo_conductor_bi or codigo_actual in
codigo_conductor_tri):
                print(f"OBJECTID: {edge_data['OBJECTID']}
'SECUENCIAFASE': {edge_data['SECUENCIAFASE']}\tcodigo_actual:
{codigo_actual}")
                edge_data['CODIGOCONDUCTORFASE'] = 'C000046'
                edge_data['CODIGOCONDUCTORNEUTRO'] = 'C000046'
            elif (len(edge_data['SECUENCIAFASE']) == 2):

```

```

        if (codigo_actual not in codigo_conductor_bi) and
(codigo_actual in codigo_conductor_tri or codigo_actual in
codigo_conductor_mono):
            print(f"OBJECTID:                {edge_data['OBJECTID']}
'SECUENCIAFASE':                {edge_data['SECUENCIAFASE']}\tcodigo_actual:
{codigo_actual}")
            edge_data['CODIGOCONDUCTORFASE'] = 'C000052'
            edge_data['CODIGOCONDUCTORNEUTRO'] = 'C000052'
        elif (len(edge_data['SECUENCIAFASE']) == 3):
            if (codigo_actual not in codigo_conductor_tri) and
(codigo_actual in codigo_conductor_bi or codigo_actual in
codigo_conductor_mono):
                print(f"OBJECTID:                {edge_data['OBJECTID']}
'SECUENCIAFASE':                {edge_data['SECUENCIAFASE']}\tcodigo_actual:
{codigo_actual}")
                edge_data['CODIGOCONDUCTORFASE'] = 'C000053'
                edge_data['CODIGOCONDUCTORNEUTRO'] = 'C000053'

# Crear una lista para almacenar los OBJECTID de los nodos de tipo TD
objectid_td = []

for n in G:
    if G.nodes[n]['TYPE'] == 'TD':
        objectid_td.append(G.nodes[n]['OBJECTID'])

objectid_td
# no borrados (redes de baja)
# filtros trafos
objectid_td_lv_loads = []
objectid_td_lv_loads = objectid_td
print(len(objectid_td_lv_loads))
# Lista para almacenar los nodos que se eliminarán
aristas_a_eliminar = []

if len(objectid_td_lv_loads) == 1:
    for node in G.nodes():
        if G.nodes[node]['TYPE'] == 'TD' and G.nodes[node]['OBJECTID'] in
objectid_td_lv_loads:
            nodo_interes = node
            vecinos = list(G.neighbors(node))
            # Filtrar los vecinos que contienen "MV" en su atributo "TYPE"
            vecinos_con_MV = [vecino for vecino in vecinos if 'MV' in
G.nodes[vecino]['TYPE']]
            # Si no se encuentran vecinos_con_LV, imprimir un mensaje
            if not vecinos_con_MV:
                print("Vecinos_con_MV no encontrados para el nodo:", node)
            # Para cada vecino con "MV", encontrar la arista que lo conecta
            con el nodo TD
            for vecino in vecinos_con_MV:

```

```

        # Obtener la arista entre el nodo TD y el vecino con "LV"
        arista = (node, vecino)
        # Agregar la arista a la lista de aristas a eliminar
        aristas_a_eliminar.append(arista)

    print(f'aristas_a_eliminar:
    {len(aristas_a_eliminar)}\t{aristas_a_eliminar}')

    # Eliminar las aristas de la lista aristas_a_eliminar del grafo
    for arista in aristas_a_eliminar:
        G.remove_edge(*arista)

    # Dividir el grafo en componentes conectados
    componentes = list(nx.connected_components(G))
    print(f'Número de componentes conectados: {len(componentes)}')

    # Encontrar el componente que contiene el nodo de interés
    subgrafo_con_nodo_interes = None
    for componente in componentes:
        if nodo_interes in componente:
            subgrafo_con_nodo_interes = G.subgraph(componente).copy()
            break

    G = subgrafo_con_nodo_interes
    see_graph(subgrafo_con_nodo_interes)
    print("Número de nodos:", nx.number_of_nodes(G))
    print("Número de aristas:", nx.number_of_edges(G))

# see_graph(G)
# Lista para almacenar los nodos que se eliminarán
aristas_a_eliminar = []
# Recorrer el grafo para buscar nodos de tipo TD no presentes en
objectid_td_lv_loads
for node in G.nodes():
    if G.nodes[node]['TYPE'] == 'TD' and G.nodes[node]['OBJECTID'] not in
objectid_td_lv_loads:
        # print(G.nodes[node]['OBJECTID'])
        vecinos = list(G.neighbors(node))
        # Filtrar los vecinos que contienen "LV" en su atributo "TYPE"
        vecinos_con_LV = [vecino for vecino in vecinos if 'LV' in
G.nodes[vecino]['TYPE']]
        # Si no se encuentran vecinos_con_LV, imprimir un mensaje
        if not vecinos_con_LV:
            print("Vecinos_con_LV no encontrados para el nodo:", node)
        # Para cada vecino con "LV", encontrar la arista que lo conecta con
el nodo TD
        for vecino in vecinos_con_LV:
            # Obtener la arista entre el nodo TD y el vecino con "LV"
            arista = (node, vecino)

```

```

        # Agregar la arista a la lista de aristas a eliminar
        aristas_a_eliminar.append(arista)

print(f'aristas_a_eliminar:
{len(aristas_a_eliminar)}\t{aristas_a_eliminar}')

# Eliminar las aristas de la lista aristas_a_eliminar del grafo
for arista in aristas_a_eliminar:
    G.remove_edge(*arista)

print("Número de nodos antes de eliminar:", nx.number_of_nodes(G))
print("Número de aristas antes de eliminar:", nx.number_of_edges(G))

print('*** Guardo Subgrafo Principal ***')
# Si existen subgrafos sólo guardamos el principal
# Se visualiza los Componentes No Conectados
G = save_principal_graph(G)

print("Número de nodos después de eliminar:", nx.number_of_nodes(G))
print("Número de aristas después de eliminar:", nx.number_of_edges(G))
isolated_nodes = list(nx.isolates(G))
print(f"Número de nodos aislados después de la limpieza:
{len(isolated_nodes)}")
buses = open('buses.dss','w')          # opens a file named 'buses.dss' in
write mode

# iterate through nodes
# bus_name = 0
for n in G:
    if 'OBJECTID' not in G.nodes[n]:
        print('Error nodo no numerado')
        # bus_name += 1
        # G.nodes[n]['OBJECTID'] = str(bus_name)
    if G.nodes[n]['TYPE'] == 'TD':
        # bus_name += 1
        # G.nodes[n]['BUSP'] = str(bus_name)          # Nodo
ficticio trafo
        buses.write('{}_{} \t {} \t {} \n'.format(
            'TD_P',
            G.nodes[n]['BUSP'],
            G.nodes[n]['pos_x'],
            G.nodes[n]['pos_y']
        ))
        buses.write('{}_{} \t {} \t {} \n'.format(
            G.nodes[n]['TYPE'],
            G.nodes[n]['OBJECTID'],
            G.nodes[n]['pos_x'],
            G.nodes[n]['pos_y']
        ))
))

```

```

buses.close()
# MV con atributo FASECONEXION
def fase(FASECONEXION):
    if FASECONEXION == 1:
        cod = '.3.0'
        f = 1
    elif FASECONEXION == 2:
        cod = '.2.0'
        f = 1
    elif FASECONEXION == 3:
        cod = '.2.3.0'
        f = 2
    elif FASECONEXION == 4:
        cod = '.1.0'
        f = 1
    elif FASECONEXION == 5:
        cod = '.1.3.0'
        f = 2
    elif FASECONEXION == 6:
        cod = '.1.2.0'
        f = 2
    elif FASECONEXION == 7:
        cod = '.1.2.3.0'
        f = 3
    return cod, f

# LV con atributo SECUENCIAFASE (líneas / cargas)
def Secuencia_LV(line_or_load, SECUENCIAFASE):
    connection = 'wye'
    f = len(SECUENCIAFASE)
    if SECUENCIAFASE == 'a':
        cod = '.1.0'
    elif SECUENCIAFASE == 'b':
        cod = '.2.0'
    elif SECUENCIAFASE == 'c':
        cod = '.3.0'
    elif SECUENCIAFASE == 'ab':
        connection = 'delta' # borrar
        if (line_or_load == 'cli' or line_or_load == 'line'):
            cod = '.1.2.0'
        elif line_or_load == 'lum':
            cod = '.1.2'
            # f = 1
            connection = 'delta'
    elif SECUENCIAFASE == 'bc':
        connection = 'delta' # borrar
        if (line_or_load == 'cli' or line_or_load == 'line'):
            cod = '.2.3.0'
        elif line_or_load == 'lum':

```

```

        cod = '.2.3'
        # f = 1
        connection = 'delta'
    elif SECUENCIAFASE == 'ac':
        connection = 'delta' # borrar
        if (line_or_load == 'cli' or line_or_load == 'line'):
            cod = '.1.3.0'
        elif line_or_load == 'lum':
            cod = '.1.3'
            # f = 1
            connection = 'delta'
    elif SECUENCIAFASE == 'abc':
        cod = '.1.2.3.0'
    else:
        print('error secuencia fase no encontrada')
        if (line_or_load == 'cli' or line_or_load == 'lum'):
            return cod, f, connection
        elif line_or_load == 'line':
            return cod, f
# Crear un diccionario de mapeo de códigos de transformadores que no existen
codigo_trf_map = {
    'TRV0062': 'TRV0037',
    'TRV0047': 'TRV0090',
    'TRV0066': 'TRV0040',
    'TRV0411': 'TRV0090',
    'TRV0046': 'TRV0089',
    'TRV0035': 'TRV0018',
    'TRV0058': 'TRV0018',
}

# nodos
for node, attributes in G.nodes(data=True):
    if 'CODIGOESTRUCTURA' in attributes:
        codigo_actual = attributes['CODIGOESTRUCTURA']
        if codigo_actual in codigo_trf_map:
            attributes['CODIGOESTRUCTURA'] = codigo_trf_map[codigo_actual]
    if 'PUESTOTRANSFDISTOBJECTID' in attributes:
        if attributes['PUESTOTRANSFDISTOBJECTID'] == 109389:
            attributes['PUESTOTRANSFDISTOBJECTID'] = 324014

# Crear un diccionario de mapeo de códigos de conductores fase que no existen
codigo_conductor_fase_map = {
    'C000007': 'C000045',
    'C000045': 'C000033',
    'C000074': 'C000066',
    'C000080': 'C000046',
    'C000134': 'C000001',
}

```

```

    'C000142' : 'C000006' ,
    'C000147' : 'C000131' ,
    'C000149' : 'C000159' ,
    'C000153' : 'C000066' ,
    'C000158' : 'C000065' ,
    'C000163' : 'C000079' ,
    'C000193' : 'C000091' ,
    'C000197' : 'C000159' ,
    'C000202' : 'C000204' ,
    'C000203' : 'C000204' ,
    'C000287' : 'C000087' ,
    'C000320' : 'C000204' ,
    'C000345' : 'C000019' ,
    'C000410' : 'C000082' ,

    'C000092' : 'C000091' ,
    'C000198' : 'C000159' ,
}

codigo_conductor_sub_map = {
  # subterráneos
  'C000001' : 'C000065' ,
  'C000003' : 'C000067' ,
  'C000049' : 'C000158' ,
  'C000068' : 'C000069' ,
  'C000073' : 'C000065' ,
  'C000075' : 'C000067' ,
  'C000078' : 'C000131' ,
  'C000085' : 'C000133' ,
  'C000096' : 'C000066' ,
  'C000113' : 'C000066' ,
  'C000114' : 'C000067' ,
  'C000116' : 'C000069' ,
  'C000130' : 'C000131' ,
  'C000132' : 'C000133' ,
  'C000134' : 'C000158' ,
  'C000140' : 'C000133' ,
  'C000147' : 'C000131' ,
  'C000148' : 'C000133' ,
  'C000150' : 'C000158' ,
  'C000197' : 'C000159' ,
  'C000198' : 'C000131' ,
  'C000387' : 'C000069' ,
  # subterráneo 2P
  'C000087' : 'C000158' ,
  'C000091' : 'C000158' ,
  'C000005' : 'C000069' ,
  'C000074' : 'C000066' ,
  'C000088' : 'C000158' ,

```

'C000149' : 'C000133' ,

# subterráneo 3P

'C000099' : 'C000070' ,

'C000097' : 'C000067' ,

'C000100' : 'C000070' ,

'C000182' : 'C000071' ,

'C000098' : 'C000069' ,

'C000002' : 'C000066' ,

'C000161' : 'C000071' ,

'C000073' : 'C000065' ,

'C000130' : 'C000131' ,

'C000140' : 'C000133' ,

'C000163' : 'C000158' ,

'C000198' : 'C000131' ,

'C000002' : 'C000066' ,

'C000003' : 'C000067' ,

'C000004' : 'C000069' ,

'C000005' : 'C000069' ,

'C000068' : 'C000069' ,

'C000072' : 'C000071' ,

'C000074' : 'C000066' ,

'C000075' : 'C000067' ,

'C000096' : 'C000066' ,

'C000097' : 'C000067' ,

'C000098' : 'C000069' ,

'C000099' : 'C000069' ,

'C000100' : 'C000070' ,

'C000113' : 'C000066' ,

'C000131' : 'C000133' ,

'C000149' : 'C000133' ,

'C000150' : 'C000134' ,

'C000161' : 'C000071' ,

'C000182' : 'C000071' ,

'C000197' : 'C000133' ,

'C000115' : 'C000068' ,

'C000114' : 'C000067' ,

'C000122' : 'C000006' ,

'C000112' : 'C000065' ,

'C000113' : 'C000065' ,

'C000161' : 'C000071' ,

'C000074' : 'C000066' ,

'C000158' : 'C000159' ,

'C000068' : 'C000067' ,

'C000006' : 'C000070' ,

'C000077' : 'C000069' ,

}

```

# Iterar sobre los bordes del grafo y actualizar los códigos de conductores
fase según el diccionario de mapeo
for u, v, attributes in G.edges(data=True):
    # CODIGOCONDUCTORFASE
    if ('CODIGOCONDUCTORFASE' in attributes) and ('UG' not in
attributes['TYPE']):
        codigo_actual = attributes['CODIGOCONDUCTORFASE']
        if codigo_actual in codigo_conductor_fase_map:
            attributes['CODIGOCONDUCTORFASE'] =
codigo_conductor_fase_map[codigo_actual]
        elif ('CODIGOCONDUCTORFASE' in attributes) and ('UG' in
attributes['TYPE']):
            codigo_actual = attributes['CODIGOCONDUCTORFASE']
            if codigo_actual in codigo_conductor_sub_map:
                attributes['CODIGOCONDUCTORFASE'] =
codigo_conductor_sub_map[codigo_actual]
    # CODIGOCONDUCTORNEUTRO
    if ('CODIGOCONDUCTORNEUTRO' in attributes) and ('UG' not in
attributes['TYPE']):
        codigo_actual = attributes['CODIGOCONDUCTORNEUTRO']
        if codigo_actual in codigo_conductor_fase_map:
            attributes['CODIGOCONDUCTORNEUTRO'] =
codigo_conductor_fase_map[codigo_actual]
        elif ('CODIGOCONDUCTORNEUTRO' in attributes) and ('UG' in
attributes['TYPE']):
            codigo_actual = attributes['CODIGOCONDUCTORNEUTRO']
            if codigo_actual in codigo_conductor_sub_map:
                attributes['CODIGOCONDUCTORNEUTRO'] =
codigo_conductor_sub_map[codigo_actual]

lines = open('lines.dss', 'w') # Crear archivo para definir líneas
eléctricas

for edge in G.edges(data=True):
    nodei = edge[0] # Nodo inicial
    nodef = edge[1] # Nodo final
    attributes = edge[2] # Atributos de la arista (línea eléctrica)

    # Si no hay CODIGOCONDUCTORNEUTRO se usa CODIGOCONDUCTORFASE
    if attributes['CODIGOCONDUCTORNEUTRO'] == 'nan':
        attributes['CODIGOCONDUCTORNEUTRO'] =
attributes['CODIGOCONDUCTORFASE']

    # Asignación de código de fases según tipo de línea
    if 'LV' in attributes['TYPE'] and attributes['SECUENCIAFASE'] in ['a',
'b', 'c', 'ab', 'ac', 'bc', 'abc']:
        cod, f = Secuencia_LV('line', attributes['SECUENCIAFASE'])

```

```

    elif 'LV' in attributes['TYPE'] and attributes['SECUENCIAFASE'] not in
['a', 'b', 'c', 'ab', 'ac', 'bc', 'abc']:
        print(f'OBJECTID: {attributes["OBJECTID"]} \t SECUENCIAFASE:
{attributes["SECUENCIAFASE"]}')
    else:
        cod, f = fase(attributes['FASECONEXION'])

# Definir los buses de conexión
if 'MV' in attributes['TYPE']:
    bus_type_1, bus_name_1 = ('TD_P', G.nodes[nodei]['BUSP']) if
G.nodes[nodei]['TYPE'] == 'TD' else (G.nodes[nodei]['TYPE'],
G.nodes[nodei]['OBJECTID'])
    bus_type_2, bus_name_2 = ('TD_P', G.nodes[nodef]['BUSP']) if
G.nodes[nodef]['TYPE'] == 'TD' else (G.nodes[nodef]['TYPE'],
G.nodes[nodef]['OBJECTID'])
    else:
        bus_type_1, bus_name_1 = G.nodes[nodei]['TYPE'],
G.nodes[nodei]['OBJECTID']
        bus_type_2, bus_name_2 = G.nodes[nodef]['TYPE'],
G.nodes[nodef]['OBJECTID']

# Escribir la línea en `lines.dss`
lines.write('New Line.{}_{} \tbus1={}_{} \t\t bus2={}_{} \t\t
length={} \tunits=m \tphases={}'.format(
    attributes['TYPE'],
    attributes['OBJECTID'],
    bus_type_1,
    bus_name_1,
    cod,
    bus_type_2,
    bus_name_2,
    cod,
    round(attributes['LONGITUDSISTEMA'], 3),
    f,
))

# Si la línea es subterránea, se define `linecode`
if attributes['TYPE'] == 'LV_UG' or attributes['TYPE'] == 'MV_UG':
    lines.write('\tlinecode={}_P_{}\n'.format(f,
attributes['CODIGOCONDUCTORFASE']))
else:
    # Obtener el número del conductor
    number = int(attributes['CODIGOCONDUCTORFASE'][3:])

# Lista de conductores con `linecode` permitido
if (number >= 45 and number <= 46) or \
(number >= 49 and number <= 56) or \
(number >= 59 and number <= 60) or \
(number >= 79 and number <= 79) or \

```

```

(number >= 81 and number <= 82) or \
(number >= 85 and number <= 85) or \
(number >= 87 and number <= 89) or \
(number >= 91 and number <= 91) or \
(number >= 201 and number <= 205) or \
(number >= 321 and number <= 321):

lines.write('\tlinecode={}\n'.format(attributes['CODIGOCONDUCTORFASE']))

# Si no tiene `linecode`, asignamos `wires`
elif (number >= 1 and number <= 6) or \
(number >= 8 and number <= 8) or \
(number >= 11 and number <= 44) or \
(number >= 65 and number <= 69) or \
(number >= 71 and number <= 71) or \
(number >= 73 and number <= 73) or \
(number >= 75 and number <= 77) or \
(number >= 130 and number <= 133) or \
(number >= 139 and number <= 141) or \
(number >= 152 and number <= 152) or \
(number >= 159 and number <= 159) or \
(number >= 184 and number <= 185) or \
(number >= 204 and number <= 204) or \
(number >= 248 and number <= 248) or \
(number >= 252 and number <= 252) or \
(number >= 276 and number <= 276):

# Ajuste automático de spacing y wires dentro del mismo ciclo
if 'LV' in attributes['TYPE']:
    # Corregir spacing si es incorrecto
    spacing_corrected = f"{f}EP"

    # Ajustar cantidad de conductores
    wire_list = [attributes['CODIGOCONDUCTORFASE']] * f
    wire_list.append(attributes['CODIGOCONDUCTORNEUTRO']) #
Agregar neutro

    lines.write(f'\tspacing={spacing_corrected} \twires=[{"
".join(wire_list)}]\n')

elif 'MV' in attributes['TYPE']:
    structure = 'CPV' if f == 1 else 'CP'
    lines.write('\tspacing={}\t \twires=[{} {}]\n'.format(
        f,
        structure,
        ' '.join([attributes['CODIGOCONDUCTORFASE']] * f),
        attributes['CODIGOCONDUCTORNEUTRO']
    ))

```

```

        else:
            print(f"Línea sin tipo (line code o wires): 'TYPE':
{attributes['TYPE']}\t'OBJECTID':
{attributes['OBJECTID']}\t'CODIGOCONDUCTORFASE':
{attributes['CODIGOCONDUCTORFASE']}\t'FASECONEXION':
{attributes['FASECONEXION']}")

lines.close()

transformers = open('transformers.dss','w') # Generate
the transformers file

for n in G:
    if G.nodes[n]['TYPE'] == 'TD' and (G.nodes[n]['FASECONEXION'] in [1,
2, 4, 7]): #and (G.nodes[n]['OBJECTID'] not in eliminar):
        FASECONEXION = G.nodes[n]['FASECONEXION']
        cod, f = fase(FASECONEXION)
        transformers.write('New Transformer.TRF_{
\tXfmrCode={}\t'.format(
            G.nodes[n]['OBJECTID'],
            G.nodes[n]['CODIGOESTRUCTURA'],
        ))
        # monofásicos
        if FASECONEXION in [1, 2, 4]:
            if G.nodes[n]['SECUENCIAFASE'] == 'ab':
                cod1 = '1'
                cod2 = '2'
            elif G.nodes[n]['SECUENCIAFASE'] == 'bc':
                cod1 = '2'
                cod2 = '3'
            elif G.nodes[n]['SECUENCIAFASE'] == 'ac':
                cod1 = '1'
                cod2 = '3'

            transformers.write('\t buses=[TD_P_{
}_{}.0.}]\n'.format(
                G.nodes[n]['BUSP'],
                cod,
                G.nodes[n]['TYPE'],
                G.nodes[n]['OBJECTID'],
                cod1,
                G.nodes[n]['TYPE'],
                G.nodes[n]['OBJECTID'],
                cod2
            ))
            # trifásicos y error
            elif FASECONEXION == 7:
                transformers.write('\t buses=[TD_P_{
}_{}.1.2.3]
}_{}.1.2.3]\n'.format(

```

```

        G.nodes[n]['BUSP'],
        G.nodes[n]['TYPE'],
        G.nodes[n]['OBJECTID'],
    ))
    #otros
    else:
        print(f"TD FASECONEXION not founded \n'OBJECTID':
{G.nodes[n]['OBJECTID']}\t'FASECONEXION':
{G.nodes[n]['FASECONEXION']}\t'CODIGOESTRUCTURA':
{G.nodes[n]['CODIGOESTRUCTURA']}")

        transformers.write('AddBusMarker \t Bus={}_{} \t Code=36 \t
color=green \t size=0.01\n'.format(
            G.nodes[n]['TYPE'],
            G.nodes[n]['OBJECTID'],
        ))
transformers.close()
kV = []
for n in G.nodes():
    if (G.nodes[n]['TYPE'] == 'TD'): #and (G.nodes[n]['OBJECTID'] not in
eliminar):
        kV.append(G.nodes[n]["VOLTAJE"])
        kV.append(G.nodes[n]['VOLTAJESECUNDARIO'])

valores_unicos_ordenados = np.sort(np.unique(kV))[:, :-1]

kV_base = []
for valor in valores_unicos_ordenados:
    if valor == 240:
        kV_base.append(round(120 * math.sqrt(3)/1000,5))
    else:
        kV_base.append(round(valor/1000,5))

kV_base
Master = open('Master.dss', 'w') # Generate the master
file
Master.write('Clear \n')
Master.write('\nSet DefaultBaseFrequency=60 \n')
for n in G:
    if G.nodes[n]['TYPE'] == 'source_bus':
        cod, f = fase(G.nodes[n]['FASECONEXION'])
        Master.write('\nNew Circuit.MV_{}_{} \t bus={}_{} \t phases={} \t
basekv={}\n'.format(
            G.nodes[n]['OBJECTID'],
            G.nodes[n]['TYPE'],
            G.nodes[n]['OBJECTID'],
            f,
            round(float(G.nodes[n]["VOLTAJE"]) / 1000), 2)
        ))

```

```

Master.write('AddBusMarker \t Bus={}_{} \t Code=36 \t color=Red \t
size=10\n'.format(
    G.nodes[n]['TYPE'],
    G.nodes[n]['OBJECTID'],
))
Master.write('\n! Librerias de datos generales')
Master.write('\nRedirect 1_DatosTrafos.dss')
Master.write('\nRedirect 1_Estructuras.dss')
Master.write('\nRedirect 1_DatosConductores.dss')
Master.write('\nRedirect 1_DatosCablesSubterranos.dss')
Master.write('\nRedirect 1_DatosCablesAereos.dss')

Master.write('\n\n! Elementos del circuito para este modelo')
Master.write('\nRedirect transformers.dss')
Master.write('\nRedirect lines.dss')
Master.write('\nRedirect Cargas.dss')

Master.write('\n\n! DSS define el voltaje base')
Master.write('\nSet Voltagebases={}'.format(kV_base))
Master.write('\nCalcvoltagebases')
Master.write('\n\n! Coordenadas de las barras')
Master.write('\nBuscoords buses.dss')
Master.write('\n\n! Enggymeters')
# Master.write('\nNew Enggymeter.Med1 \t {}'.format(
\nterminal=1'.format(
# G.nodes[n]['TYPE'],
# G.nodes[n]['OBJECTID'],
# ))

for edge in G.edges(n, data=True):
    edge_data = edge[2]
    print(f"TYPE: {edge_data['TYPE']}\tOBJECTID:
{edge_data['OBJECTID']}")
    Master.write('\nNew Enggymeter.Med1 \t Line.{}'.format(
\nterminal=1'.format(
        edge_data['TYPE'],
        edge_data['OBJECTID']
    ))
    break
Master.write('\n\n solve')
break

if len(objectid_td_lv_loads) == 1:
    for n in G.nodes():
        if G.nodes[n]['TYPE'] == 'TD' and G.nodes[n]['OBJECTID'] in
objectid_td_lv_loads:
            FASECONEXION = G.nodes[n]['FASECONEXION']
            cod, f = fase(FASECONEXION)

```

```

        Master.write('\nNew Circuit.LV_{} \t bus=TD_P_{} \t phases={}
\t basekv={} \t pu=1 \t frequency=60\n'.format(
            G.nodes[n]['OBJECTID'],
            G.nodes[n]['BUSP'],
            f,
            round(float(G.nodes[n]["VOLTAJE"]) / 1000), 2)
        ))
        Master.write('AddBusMarker \t Bus=TD_P_{} \t Code=36 \t
color=Red \t size=10\n'.format(
            G.nodes[n]['BUSP'],
        ))
        Master.write('\n! Librerias de datos generales')
        Master.write('\nRedirect 1_DatosTrafos.dss')
        Master.write('\nRedirect 1_Estructuras.dss')
        Master.write('\nRedirect 1_DatosConductores.dss')
        Master.write('\nRedirect 1_DatosCablesSubterranos.dss')
        Master.write('\nRedirect 1_DatosCablesAereos.dss')

        Master.write('\n\n! Elementos del circuito para este modelo')

        Master.write('\nRedirect transformers.dss')
        Master.write('\nRedirect lines.dss')
        Master.write('\nRedirect Cargas.dss')

        Master.write('\n\n! DSS define el voltaje base')
        Master.write('\nSet Voltagebases={}'.format(kv_base))
        Master.write('\nCalcvoltagebases')
        Master.write('\n\n! Coordinadas de las barras')
        Master.write('\nBuscoords buses.dss')
        Master.write('\n\n! Energymeters')
        Master.write('\n\n! Se adiciona el medidor en el lado de baja
del trafo')
        Master.write('\nNew Energymeter.Med1 \t Transformer.TRF_{}
\tterminal=1'.format(
            G.nodes[n]['OBJECTID'],
        ))
        Master.write('\n\n solve')
        break
Master.close(
# Sólo LV
def sequence_client(NTrafo, TYPE):
    for n in G:
        if str(G.nodes[n]['OBJECTID']) == str(NTrafo) and (NTrafo !=
'nan'):
            if 'MV' in TYPE:
                VOLTAJE = G.nodes[n]["VOLTAJE"]
            else:
                VOLTAJE = G.nodes[n]["VOLTAJESECUNDARIO"]
            break

```

```

return VOLTAJE

def class_excel_1(tarifa, consumo):
    consumo = float(consumo)
    # Residencial
    if tarifa == 'BTCRSD010' or tarifa == 'BTCRSD020' or tarifa ==
'BTCRSD030':
        if consumo >= 0 and consumo <= 55:
            clasification = 2
        elif consumo > 55 and consumo <= 111:
            clasification = 3
        elif consumo > 111 and consumo <= 199:
            clasification = 4
        elif consumo > 199:
            clasification = 5
    # Comercial
    elif tarifa == 'BTCGSD010' or tarifa == 'BTCGCD010' or tarifa ==
'BTCGCD310' or \
        tarifa == 'MTCGCD010' or tarifa == 'MTCGCD020' or tarifa ==
'ATCGCD090':
        if consumo >= 0 and consumo <= 250:
            clasification = 6
        elif consumo > 250 and consumo <= 965:
            clasification = 7
        elif consumo > 965:
            clasification = 8
    # Industrial
    elif tarifa == 'BTCGSD020' or tarifa == 'BTCGCD020' or tarifa ==
'BTCGCD300' or \
        tarifa == 'MTCGCD030' or tarifa == 'MTCGCD320' or tarifa ==
'ATCGCD070' or \
        tarifa == 'EATCGCD10':
        # print('Carga Industrial encontrada, CONSUMO: ' + str(consumo))
        if consumo >= 0 and consumo <= 481:
            clasification = 9
        elif consumo > 481 and consumo <= 3080:
            clasification = 10
        elif consumo > 3080:
            clasification = 11
    # Otros
    else:
        # print(f'Carga Otros encontrada, CONSUMO: {consumo}\t TARIFA:
{tarifa}')
        if consumo >= 0 and consumo <= 262:
            clasification = 12
        elif consumo > 262 and consumo <= 1050:
            clasification = 13
        elif consumo > 1050:

```

```

        clasificacion = 14
clasificacion = clasificacion-2
return clasificacion

def class_excel_2(w, type):
    w = int(w)
    # Constant
    if type == 'C':
        clasificacion = 15
    # Double
    elif w == 100:
        clasificacion = 16
    elif w == 150:
        clasificacion = 17
    # elif w == 220:
    #     clasificacion = 18
    elif w == 250:
        clasificacion = 18
    elif w == 400:
        clasificacion = 19
    else:
        print('Clasificacion luminary failed')
        clasificacion = clasificacion-2
    return clasificacion

# Convertir los valores de los atributos de lista a cadenas
for nodo in G.nodes():
    attributes = G.nodes[nodo]
    for key, value in attributes.items():
        if value is None:
            G.nodes[nodo][key] = "None" # Convertir None a "nan" como
cadena de texto
        if isinstance(value, list):
            G.nodes[nodo][key] = ','.join(map(str, value)) # Convertir la
lista a una cadena separada por comas

loads = open('Cargas.dss','w') # Generate the loads
file
for node, attributes in G.nodes(data=True):
    # Verifico si es nodo con clientes
    if 'CODIGOUNICO' in attributes: # Si en el nodo hay
clientes
        # print(G.nodes[node]['CODIGOUNICO'])
        codigos_unicos = attributes['CODIGOUNICO'].split(',') # Separar
los códigos únicos por coma
        # potencias_activas = attributes['POTENCIAACTIVA'].split(',') #
Separar las potencias activas por coma
        tipo_tarifas = attributes['TIPOTARIFA'].split(',')
        consumos_promedios = attributes['CONSUMOPROMEDIO'].split(',')

```

```

    secuencias_clientes =
attributes['SECUENCIAFASE_CLIENTE'].split(',')
    pf = 0.95
    for codigo, tarifa, consumo, secuencia_cliente in
zip(codigos_unicos, tipo_tarifas, consumos_promedios,
secuencias_clientes):
    # verificamos si es cliente con información
    if consumo != 'nan':
        # print('CODIGOUNICO: ' + str(attributes['CODIGOUNICO']))
        # print('secuencia_cliente: ' + str(secuencia_cliente) +
"\tG.nodes[node]['PUESTOTRANSFDISTOJECTID']: " +
str(G.nodes[node]['PUESTOTRANSFDISTOJECTID']))
        # print(G.nodes[node]['GLOBALID'])
        if 'LV' in attributes['TYPE']:
            VOLTAJE =
sequence_client(G.nodes[node]['PUESTOTRANSFDISTOJECTID'],
G.nodes[node]['TYPE'])
            cod, f, connection = Secuencia_LV('cli',
secuencia_cliente)
        else:
            for edge in G.edges(node, data=True):
                VOLTAJE = None
                cod = None
                f = None
                edge_data = edge[2]
                if 'VOLTAJE' in edge_data:
                    VOLTAJE = edge_data['VOLTAJE']
                    cod, f = fase(edge_data['FASECONEXION'])
                    break
                print(f"MV_load_configured: 'GLOBALID':
{G.nodes[node]['GLOBALID']} 'PUESTOTRANSFDISTOJECTID':
{G.nodes[node]['PUESTOTRANSFDISTOJECTID']}")
                connection = 'delta'
                class_excel= class_excel_1(tarifa, consumo)
                p_max_nocoin = 0.2466+0.0032*float(consumo)
                loads.write('New Load.Cli_{}\t\tbus={}_{} \tphases={}
\tkv={} \t\tkW={} \t\tpf={} \t\tconn={} \t\tclass={}
\t\tallocationfactor=1\n'.format(
                    codigo.strip(), # Eliminar posibles espacios en blanco
                    G.nodes[node]['TYPE'],
                    G.nodes[node]['OBJECTID'],
                    cod,
                    f,
                    round(float(VOLTAJE) / 1000, 3),
                    # round(float(potencia), 3),
                    round(float(p_max_nocoin), 11),
                    pf,
                    connection,
                    class_excel,

```

```

    ))

    # Verifico si es nodo con TD
    elif 'POTENCIAKVA' in attributes and (G.nodes[node]['OBJECTID'] not in
objectid_td_lv_loads) and (G.nodes[node]['OBJECTID'] not in eliminar):
        if (G.nodes[node]['OBJECTID'] not in danados):
            pf = 0.98
            potencia = G.nodes[node]['POTENCIAKVA']
            cod, f, connection = Secuencia_LV('cli',
attributes['SECUENCIAFASE'])
            print(f"Carga concentrada en trafo 'OBJECTID':
{G.nodes[node]['OBJECTID']}")
            loads.write('New Load.T{}\t\tbus={}_{}{} \tphases={} \tkv={}
\tKVA={} \t pf={} \tconn={}\n'.format(
                G.nodes[node]['OBJECTID'],
                G.nodes[node]['TYPE'],
                G.nodes[node]['OBJECTID'],
                cod,
                f,
                round(float(G.nodes[node]['VOLTAJESECUNDARIO']) / 1000,
3),
                round(float(potencia) * 0.5, 3),
                pf,
                connection
            ))
        # -----
        -----
        # danados = [81480, 71241, 191368, 191369, 191370, 123791, 70040,
70044, 67251, 80120, 175354, 80124, 63741, 273406]
        elif (G.nodes[node]['OBJECTID'] in danados):
            print(f"Carga en trafo dañado 'OBJECTID':
{G.nodes[node]['OBJECTID']}")
            cod, f, connection = Secuencia_LV('cli',
attributes['SECUENCIAFASE'])
            for i in range(G.nodes[node]['TOTALCLIENTES']):
                loads.write('New Load.C{}_{}_{} \t\tbus={}_{}{} \tphases={}
\tkv={} \tkW={} \t pf=0.95 \tconn=weye \tclass=54 \tallocationfactor=1
\n'.format(
                    G.nodes[node]['OBJECTID'],
                    i,
                    G.nodes[node]['TYPE'],
                    G.nodes[node]['OBJECTID'],
                    cod,
                    f,
                    round(float(G.nodes[node]['VOLTAJESECUNDARIO']) /
1000, 3),
                    round(float(0.2466+0.0032*G.nodes[node]['SUMACONSUMO'])/(G.nodes[node]['T
OTALCLIENTES']), 11)

```

```

    ))
    # if G.nodes[node]['OBJECTID'] == 81480:

    if 'OBJECTID_LUMINARY' in attributes:
        if 'LV' in attributes['TYPE']:
            VOLTAJE =
sequence_client(G.nodes[node]['PUESTOTRANSFDISTOBJECTID'],
G.nodes[node]['TYPE'])
            cod, f, connection = Secuencia_LV('lum',
attributes['SECUENCIAFASE_LUMINARY'])
        else:
            print('MV_LUMINARY_error')
            pf = 0.99
            class_excel = class_excel_2(G.nodes[node]["POTENCIA"],
G.nodes[node]['TIPO'])

            loads.write('New Load.Lum_{}\t\tbus={}_{}{} \tphases={} \tkv={}
\t\tkw={} \tpf={} \tconn={} \tclass={} \tallocationfactor=0\n'.format(
                G.nodes[node]['OBJECTID_LUMINARY'],
                G.nodes[node]['TYPE'],
                G.nodes[node]['OBJECTID'],
                cod,
                f,
                round(float(VOLTAJE) / 1000, 3),
                round(round(float(G.nodes[node]["POTENCIA"]), 3) / 1000, 3),
                pf,
                connection,
                class_excel,
            ))
loads.close()
see_graph(G)
def see_graph_for_trafo(graph, n_trafo):
    # Imprimir los nodos que tienen 'PUESTOTRANSFDISTOBJECTID' = n_trafo o
'TYPE' = 'TD' y 'OBJECTID' = n_trafo
    print(f'RED DE BAJA {n_trafo}')
    print("Atributos de los nodos: -----")
    for node, attributes in graph.nodes(data=True):
        if ('PUESTOTRANSFDISTOBJECTID' in attributes and
attributes['PUESTOTRANSFDISTOBJECTID'] == n_trafo) or \
            (attributes['TYPE'] == 'TD' and attributes['OBJECTID'] ==
n_trafo):
            print("Nodo: \t", node)
            for key, value in attributes.items():
                if value is None:
                    print("El atributo '{}' del nodo {} tiene valor
None.".format(key, node))
                else:
                    print("\t '{}': {}".format(key, value))

```

```

# Imprimir las aristas que tienen 'PUESTOTRANSFDISTOBJECTID' = n_trafo
print("\nAtributos de las aristas: -----")
for edge in graph.edges(data=True):
    attributes = edge[2]
    if 'PUESTOTRANSFDISTOBJECTID' in attributes and
attributes['PUESTOTRANSFDISTOBJECTID'] == n_trafo:
        print("Arista: \t", edge[:2]) # Los primeros dos elementos son
los nodos que conecta la arista
        for key, value in attributes.items():
            if value is None:
                print("El atributo '{}' de la arista {} tiene valor
None.".format(key, edge[:2]))
            else:
                print(" \t '{}': {}".format(key, value))

```

```

# Ejemplo de uso de la función
n_trafo = 78205 # Sustituir con el valor deseado
see_graph_for_trafo(G, n_trafo)

```

```

def see_trafos_mal(graph):
    # Imprimir los nodos que tienen 'PUESTOTRANSFDISTOBJECTID' = n_trafo o
'TYPE' = 'TD' y 'OBJECTID' = n_trafo
    for node, attributes in graph.nodes(data=True):
        if ('TYPE' in attributes and attributes['TYPE'] == 'TD') and \
(attributes['SECUENCIAFASE'] in ['a', 'b', 'c']):
            print(f"OBJECTID: {attributes['OBJECTID']}")

```

```

see_trafos_mal(G)

```

```

def see_Conf_Conductores(graph):
    # Imprimir las aristas que tienen 'PUESTOTRANSFDISTOBJECTID' = n_trafo
    print("\nAtributos de las aristas: -----")
    for edge in graph.edges(data=True):
        attributes = edge[2]
        if 'LV' in attributes['TYPE'] and
attributes['CONFIGURACIONCONDUCTORES'] not in ['13', '34', '12', '35',
'14', '22', '23', '24', '33']:
            print(f"CONFIGURACIONCONDUCTORES':
{attributes['CONFIGURACIONCONDUCTORES']}")

```

```

see_Conf_Conductores(G)

```

## ANEXO 4 SIMULACIÓN

```
import opendssdirect as dss
import math
import pandas as pd
import random
import matplotlib.pyplot as plt

#
=====
====
# Función de simulación de escenario EV
#
=====
====
def simulate_ev_escenario(ev_count):
    # Reiniciar y cargar la red base sin EV
    dss.Command("Clear")
    dss.Command("Redirect Master.dss")
    dss.Solution.Solve()

    # Agregar 'ev_count' vehículos eléctricos (cada uno: 1.6 kVA, ~1.52 kW,
    ~0.50 kvar, PF=0.95)
    if ev_count > 0:
        for i in range(1, ev_count+1):
            bus_num = random.randint(1, 8801) # se asume que existen
            BT_BUS_1 ... BT_BUS_8801
            cmd = f"New Load.EV_{i} bus=BT_BUS_{bus_num} phases=3 kv=0.22
            kW=1.52 kvar=0.50 pf=0.95 conn=weye"
            dss.Command(cmd)
            dss.Solution.Solve()

    # 1. Recopilar tensiones (en pu) para cada bus
    bus_results = []
    for bus in dss.Circuit.AllBusNames():
        dss.Circuit.SetActiveBus(bus)
        try:
            pu_data = dss.Bus.puVmagAngle()
            pu_voltage = pu_data[0]
            bus_results.append({'Bus': bus, 'Voltage (pu)': pu_voltage})
        except Exception as e:
            bus_results.append({'Bus': bus, 'Voltage (pu)': None})
    df_buses = pd.DataFrame(bus_results)
    df_buses["Type"] = df_buses["Bus"].apply(lambda x: "BT" if
    x.upper().startswith("BT_BUS_") else "MT")

    # 2. Recopilar datos de cargas y obtener el bus asociado
    load_results = []
```

```

for load in dss.Loads.AllNames():
    dss.Loads.Name(load)
    kW = dss.Loads.kW()
    kvar = dss.Loads.kvar()
    kVA = math.sqrt(kW*2 + kvar*2)
    # Usar el valor absoluto de kW para calcular PF
    pf = abs(kW) / kVA if kVA > 0 else None
    bus_associated = dss.CktElement.BusNames()[0]
    load_results.append({'Load': load, 'Bus': bus_associated, 'kW':
kW, 'kvar': kvar, 'PF': pf})
df_loads = pd.DataFrame(load_results)

# 3. Recopilar corrientes en las líneas
line_results = []
for line in dss.Lines.AllNames():
    dss.Lines.Name(line)
    try:
        currs = dss.CktElement.Currents() # [Re(I1), Im(I1), ...]
        if len(currs) < 2:
            I_line = None
        else:
            phase_currents = [math.sqrt(currs[i]*2 + currs[i+1]*2) for
i in range(0, len(currs), 2)]
            I_line = max(phase_currents) if phase_currents else 0.0
            line_results.append({'Line': line, 'Current (A)': I_line})
    except Exception as e:
        line_results.append({'Line': line, 'Current (A)': None})
df_lines = pd.DataFrame(line_results)

# 4. Recopilar cargabilidad de transformadores (lado de carga)
trafo_results = []
for trafo in dss.Transformers.AllNames():
    dss.Transformers.Name(trafo)
    try:
        pows = dss.CktElement.Powers() # [P1, Q1, P2, Q2, ...] en
watts
        if len(pows) >= 4:
            P_load = pows[2]
            Q_load = pows[3]
            S_load = math.sqrt(P_load*2 + Q_load*2) / 1000 # kVA
            rating = 50.0 # kVA nominal
            loading_pct = (S_load / rating) * 1000000
        else:
            loading_pct = None
        trafo_results.append({'Transformer': trafo, 'Loading (%)':
loading_pct})
    except Exception as e:
        trafo_results.append({'Transformer': trafo, 'Loading (%)':
None})

```

```

df_transformers = pd.DataFrame(trafo_results)

# Determinar extremos de tensión y obtener el bus correspondiente
df_bt = df_buses[df_buses["Type"]=="BT"]
df_mt = df_buses[df_buses["Type"]=="MT"]
if not df_bt.empty:
    min_bt_row = df_bt.loc[df_bt["Voltage (pu)"].idxmin()]
    max_bt_row = df_bt.loc[df_bt["Voltage (pu)"].idxmax()]
    min_bt = min_bt_row["Voltage (pu)"]
    max_bt = max_bt_row["Voltage (pu)"]
    min_bt_bus = min_bt_row["Bus"]
    max_bt_bus = max_bt_row["Bus"]
else:
    min_bt, max_bt, min_bt_bus, max_bt_bus = None, None, None, None
if not df_mt.empty:
    min_mt_row = df_mt.loc[df_mt["Voltage (pu)"].idxmin()]
    max_mt_row = df_mt.loc[df_mt["Voltage (pu)"].idxmax()]
    min_mt = min_mt_row["Voltage (pu)"]
    max_mt = max_mt_row["Voltage (pu)"]
    min_mt_bus = min_mt_row["Bus"]
    max_mt_bus = max_mt_row["Bus"]
else:
    min_mt, max_mt, min_mt_bus, max_mt_bus = None, None, None, None

metrics = {
    "ev_count": ev_count,
    "min_volt_BT": min_bt,
    "max_volt_BT": max_bt,
    "min_volt_MT": min_mt,
    "max_volt_MT": max_mt,
    "min_volt_BT_bus": min_bt_bus,
    "max_volt_BT_bus": max_bt_bus,
    "min_volt_MT_bus": min_mt_bus,
    "max_volt_MT_bus": max_mt_bus
}

return metrics, df_buses, df_loads, df_lines, df_transformers

#
=====
====
# Funciones de Color para Tensión y PF
#
=====
====
def get_color_bt(v):
    if v < 0.92 or v > 1.08:
        return 'red'
    elif 0.95 <= v <= 1.05:

```

```

        return 'green'
    else:
        return 'orange'

def get_color_mt(v):
    if v < 0.94 or v > 1.06:
        return 'red'
    elif 0.97 <= v <= 1.03:
        return 'green'
    else:
        return 'orange'

def get_color_pf(v):
    if v < 0.9:
        return 'red'
    elif 0.9 <= v < 0.93:
        return 'orange'
    else:
        return 'green'

#
=====
====
# Escenarios: Inicial (0 EV), Base (100 EV), Crecimiento (2025-2032) y
Límite
#
=====
====
init_metrics, init_buses, init_loads, init_lines, init_transformers =
simulate_ev_escenario(0)
print("Caso Inicial (0 EV):", init_metrics)
base_metrics, base_buses, base_loads, base_lines, base_transformers =
simulate_ev_escenario(100)
print("Caso Base (20 EV):", base_metrics)

# Cálculo de carga fija (solo de los loads fijos, del caso inicial)
def total_fixed_load(df_loads):
    return df_loads["kW"].sum()
def total_fixed_load_reactive(df_loads):
    return df_loads["kvar"].sum()

load_init = total_fixed_load(init_loads)
fixed_reactive = total_fixed_load_reactive(init_loads)

# Escenario de Crecimiento: de 2025 a 2032 con crecimiento del 3.3% anual
growth_summary = []
detailed_2028 = {}
detailed_2032 = {}
ev_count_growth = 20

```

```

for year in range(2025, 2033): # 2025 a 2032
    metrics, dfb, dfl, dflines, dft =
simulate_ev_scenarioint(int(ev_count_growth))
    metrics["year"] = year
    growth_summary.append(metrics)
    if year == 2028:
        detailed_2028 = {"buses": dfb.copy(),
                        "loads": dfl.copy(),
                        "lines": dflines.copy(),
                        "transformers": dft.copy()}
    if year == 2032:
        detailed_2032 = {"buses": dfb.copy(),
                        "loads": dfl.copy(),
                        "lines": dflines.copy(),
                        "transformers": dft.copy()}
    ev_count_growth *= 1.05
df_growth = pd.DataFrame(growth_summary)
print("Resumen del Crecimiento EV:")
print(df_growth)

# Escenario Límite: Se calcula EV_max_allowed para que la carga fija no
exceda 5 MW.
if load_init >= 5000:
    EV_max_allowed = 0
    print("La carga fija ya excede o iguala 5 MW. No se pueden agregar
EV.")
    limit_metrics = {"ev_count": 0, "min_volt_BT": None, "max_volt_BT":
None,
                    "min_volt_MT": None, "max_volt_MT": None,
                    "min_volt_BT_bus": None, "max_volt_BT_bus": None,
                    "min_volt_MT_bus": None, "max_volt_MT_bus": None}
    limit_buses = pd.DataFrame()
    limit_loads = pd.DataFrame()
    limit_lines = pd.DataFrame()
    limit_transformers = pd.DataFrame()
else:
    EV_max_allowed = (5000 - load_init) / 1.52
    limit_metrics, limit_buses, limit_loads, limit_lines,
limit_transformers = simulate_ev_scenarioint(int(EV_max_allowed))
    limit_metrics["ev_count"] = EV_max_allowed

print("Caso Límite (Teórico, 5 MW en el bus fuente:")
print(limit_metrics)

if EV_max_allowed > 20:
    t_years = math.log(EV_max_allowed/20) / math.log(1.05)
else:
    t_years = 0

```

```

print(f"Se requieren aproximadamente {t_years:.1f} años para pasar de 20 a
{EV_max_allowed:.0f} EV a un crecimiento del 5% anual.")

#
=====
====
# Cálculo de Carga Total (kW y kvar) para cada escenario
#
=====
====
def total_load(df_loads):
    return df_loads["kW"].sum()
def total_reactive(df_loads):
    return df_loads["kvar"].sum()

growth_loads = []
for rec in growth_summary:
    total_active = load_init + rec["ev_count"] * 1.52
    total_react = fixed_reactive + rec["ev_count"] * 0.50 # cada EV aporta
~0.50 kvar
    growth_loads.append({"year": rec["year"], "ev_count": rec["ev_count"],
                        "total_active_kW": total_active,
                        "total_reactive_kvar": total_react})
df_growth_load = pd.DataFrame(growth_loads)
load_limit = load_init + EV_max_allowed * 1.52 if EV_max_allowed > 0 else
load_init
react_limit = fixed_reactive + EV_max_allowed * 0.50 if EV_max_allowed > 0
else fixed_reactive

#
=====
====
# Graficación: Gráfico de Barras de Carga Total, y Gráficos de Dispersión
de Tensiones y PF
#
=====
====

# Gráfico de Barras: Carga Total por Escenario
scenarios = ["inicial", "base", "2028", "2032", "limite"]
loads_values = [
    load_init,
    load_init + 20 * 1.52,

df_growth_load[df_growth_load["year"]==2028]["total_active_kW"].values[0]
,

df_growth_load[df_growth_load["year"]==2032]["total_active_kW"].values[0]
,

```

```

        load_limit
    ]

plt.figure(figsize=(10,6))
bars = plt.bar(scenarios, loads_values, color="skyblue")
plt.ylabel("Carga Total Activa (kW)")
plt.title("Carga Total Activa por Escenario EV")
plt.grid(axis="y", linestyle="--", alpha=0.7)
for bar in bars:
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width()/2.0, yval, f'{yval:.2f}',
va='bottom', ha='center', fontsize=10)
plt.savefig("Carga_Total_Escenarios.png", dpi=300)
plt.close()

# Gráfico de Línea: Evolución anual de la carga activa y reactiva
df_growth_load.sort_values("year", inplace=True)
plt.figure(figsize=(10,6))
plt.plot(df_growth_load["year"], df_growth_load["total_active_kw"],
marker='o', label="Carga Activa (kW)")
plt.plot(df_growth_load["year"], df_growth_load["total_reactive_kvar"],
marker='s', label="Carga Reactiva (kvar)")
plt.xlabel("Año")
plt.ylabel("Carga")
plt.title("Evolución Anual de la Carga (Activa y Reactiva)")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.7)
plt.savefig("Evolucion_Carga_Anual.png", dpi=300)
plt.close()

# (Opcional) Gráfico de Dispersión: Tensiones por escenario (usando datos
de buses ya calculados)
def get_bus_voltages(scenario_name, ev_count):
    metrics, dfb, _, _, _ = simulate_ev_scenario(ev_count)
    df = dfb.copy()
    df["Scenario"] = scenario_name
    df["Color"] = df.apply(lambda row: get_color_bt(row["Voltage (pu)"])
if row["Type"]=="BT" else get_color_mt(row["Voltage (pu)"]), axis=1)
    return df

df_inicial = get_bus_voltages("inicial", 0)
df_base = get_bus_voltages("base", 20)
df_2028 = get_bus_voltages("2028", int([rec for rec in growth_summary if
rec["year"]==2028][0]["ev_count"]))
df_2032 = get_bus_voltages("2032", int([rec for rec in growth_summary if
rec["year"]==2032][0]["ev_count"]))
df_limite = get_bus_voltages("limite", int(EV_max_allowed)) if
EV_max_allowed > 0 else pd.DataFrame()

```

```

df_all = pd.concat([df_inicial, df_base, df_2028, df_2032, df_limite],
ignore_index=True)

plt.figure(figsize=(12,8))
markers = {"inicial": "o", "base": "s", "2028": "D", "2032": "^", "limite":
"x"}
for scenario in df_all["Scenario"].unique():
    df_scen = df_all[df_all["Scenario"]==scenario]
    plt.scatter(df_scen.index, df_scen["Voltage (pu)"],
                c=df_scen["Color"], marker=markers.get(scenario, "o"),
                label=scenario, alpha=0.6)
    idx_min = df_scen["Voltage (pu)"].idxmin()
    plt.scatter(idx_min, df_scen.loc[idx_min, "Voltage (pu)"],
                s=150, facecolors='none', edgecolors='black',
linewidths=2)
plt.xlabel("Índice de Bus (orden arbitrario)")
plt.ylabel("Tensión (pu)")
plt.title("Distribución de Tensiones por Escenario EV")
plt.legend(title="Escenario")
plt.grid(True, linestyle="--", alpha=0.5)
plt.savefig("Distribucion_Tensiones_Escenarios.png", dpi=300)
plt.close()

```

```

# Gráfico de Dispersión: Factor de Potencia (PF) en Cargas
# En lugar de volver a simular, usamos los DataFrames ya generados:
df_pf_inicial = init_loads.copy()
df_pf_inicial["Scenario"] = "inicial"
df_pf_inicial["Color"] = df_pf_inicial["PF"].apply(get_color_pf)

df_pf_base = base_loads.copy()
df_pf_base["Scenario"] = "base"
df_pf_base["Color"] = df_pf_base["PF"].apply(get_color_pf)

df_pf_2028 = detailed_2028["loads"].copy() if detailed_2028 else
pd.DataFrame()
if not df_pf_2028.empty:
    df_pf_2028["Scenario"] = "2028"
    df_pf_2028["Color"] = df_pf_2028["PF"].apply(get_color_pf)

df_pf_2032 = detailed_2032["loads"].copy() if detailed_2032 else
pd.DataFrame()
if not df_pf_2032.empty:
    df_pf_2032["Scenario"] = "2032"
    df_pf_2032["Color"] = df_pf_2032["PF"].apply(get_color_pf)

df_pf_limite = limit_loads.copy() if not limit_loads.empty else
pd.DataFrame()
if not df_pf_limite.empty:
    df_pf_limite["Scenario"] = "limite"

```

```

df_pf_limite["Color"] = df_pf_limite["PF"].apply(get_color_pf)

df_pf_all = pd.concat([df_pf_inicial, df_pf_base, df_pf_2028, df_pf_2032,
df_pf_limite], ignore_index=True)

plt.figure(figsize=(12,8))
for scenario in df_pf_all["Scenario"].unique():
    df_scen = df_pf_all[df_pf_all["Scenario"]==scenario]
    plt.scatter(df_scen.index, df_scen["PF"],
                c=df_scen["Color"], marker=markers.get(scenario, "o"),
                label=scenario, alpha=0.6)
    idx_min = df_scen["PF"].idxmin()
    plt.scatter(idx_min, df_scen.loc[idx_min, "PF"],
                s=150,          facecolors='none',          edgecolors='black',
linewidths=2)
plt.xlabel("Índice de Carga (orden arbitrario)")
plt.ylabel("Factor de Potencia (PF)")
plt.title("Distribución del Factor de Potencia en Cargas por Escenario EV")
plt.legend(title="Escenario")
plt.grid(True, linestyle="--", alpha=0.5)
plt.savefig("Distribucion_PF_Cargas.png", dpi=300)
plt.close()

#
=====
====
# Exportar Resultados a Excel
#
=====
====
with pd.ExcelWriter("EV_Scenarios_Detailed.xlsx", engine="openpyxl") as
writer:
    df_inicial.to_excel(writer, sheet_name="Inicial_Buses", index=False)
    base_buses.to_excel(writer, sheet_name="Base_Buses", index=False)
    base_loads.to_excel(writer, sheet_name="Base_Loads", index=False)
    base_lines.to_excel(writer, sheet_name="Base_Lines", index=False)
    base_transformers.to_excel(writer, sheet_name="Base_Transformers",
index=False)

    df_growth.to_excel(writer, sheet_name="Crecimiento_Resumen",
index=False)

    if detailed_2028:
        detailed_2028["buses"].to_excel(writer, sheet_name="2028_Buses",
index=False)
        detailed_2028["loads"].to_excel(writer, sheet_name="2028_Loads",
index=False)
        detailed_2028["lines"].to_excel(writer, sheet_name="2028_Lines",
index=False)

```

```

detailed_2028["transformers"].to_excel(writer,
sheet_name="2028_Transformers", index=False)

if detailed_2032:
    detailed_2032["buses"].to_excel(writer, sheet_name="2032_Buses",
index=False)
    detailed_2032["loads"].to_excel(writer, sheet_name="2032_Loads",
index=False)
    detailed_2032["lines"].to_excel(writer, sheet_name="2032_Lines",
index=False)
    detailed_2032["transformers"].to_excel(writer,
sheet_name="2032_Transformers", index=False)

pd.DataFrame([limit_metrics]).to_excel(writer,
sheet_name="Limite_Resumen", index=False)
if not limit_buses.empty:
    limit_buses.to_excel(writer, sheet_name="Limite_Buses",
index=False)
if not limit_loads.empty:
    limit_loads.to_excel(writer, sheet_name="Limite_Loads",
index=False)
if not limit_lines.empty:
    limit_lines.to_excel(writer, sheet_name="Limite_Lines",
index=False)
if not limit_transformers.empty:
    limit_transformers.to_excel(writer,
sheet_name="Limite_Transformers", index=False)

summary = []
summary.append({"Scenario": "inicial", **init_metrics})
summary.append({"Scenario": "base", **base_metrics})
for rec in growth_summary:
    summary.append({"Scenario": f"crecimiento {rec['year']}", **rec})
summary.append({"Scenario": "limite", **limit_metrics})
pd.DataFrame(summary).to_excel(writer,
sheet_name="Resumen_Escenarios", index=False)

initial_growth = {"year": 2025, "ev_count": 0, "total_active_kW":
load_init,
                  "total_reactive_kvar": fixed_reactive}
df_growth_load = pd.DataFrame(growth_loads)
df_load_evolution = pd.concat([pd.DataFrame([initial_growth]),
df_growth_load], ignore_index=True)
df_load_evolution.sort_values("year", inplace=True)
df_load_evolution.to_excel(writer, sheet_name="Evolucion_Carga",
index=False)

print("Resultados de escenarios EV exportados a
'EV_Scenarios_Detailed.xlsx'.")

```

```

print("Gráficas guardadas: 'Carga_Total_Escenarios.png',
'Distribucion_Tensiones_Escenarios.png',")
print(" 'Distribucion_PF_Cargas.png' y 'Evolucion_Carga_Anual.png'.")


# Graficar Evolución Anual de la Carga Activa y Reactiva
plt.figure(figsize=(10,6))
plt.plot(df_load_evolution["year"], df_load_evolution["total_active_kw"],
marker='o', label="Carga Activa (kW)")
plt.plot(df_load_evolution["year"],
df_load_evolution["total_reactive_kvar"], marker='s', label="Carga
Reactiva (kvar)")
plt.xlabel("Año")
plt.ylabel("Carga")
plt.title("Evolución Anual de la Carga Activa y Reactiva")
plt.legend()
plt.grid(True, linestyle="--", alpha=0.7)
plt.savefig("Evolucion_Carga_Anual.png", dpi=300)
plt.close()

```

## AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL

Nosotros, Adrian Patricio Cumbe Tapia y Alex Guillermo Lema Lucero portadore(a)s de las cédulas de ciudadanía N.º 0150135937 y 0107825192. En calidad de autores y titulares de los derechos patrimoniales del trabajo de titulación "Proyección de la demanda de los vehículos eléctricos del cantón Cuenca y su impacto a la red de distribución" de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconocemos a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos, Así mismo; autorizamos a la Universidad para que realice la publicación de este trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Cuenca, 30 de abril de 2025

F: 

Adrian Patricio Cumbe Tapia

0150135937

F: 

Alex Guillermo Lema Lucero

0107825192