HERRAMIENTA PARA EL DISEÑO DE SISTEMAS SOLARES FOTOVOLTAICOS BASADA EN REDES NEURONALES ARTIFICIALES (RNA) PARA DETERMINAR LA CONFIGURACIÓN, SELECCIÓN DE EQUIPOS Y ARREGLOS FOTOVOLTAICOS EN COLOMBIA.

HAROLD OSWALDO OCHOA BUITRAGO

FABIAN YESID RAMIREZ LEON

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA

FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS

PROGRAMA DE INGENIERÍA EN ENERGÍA

BUCARAMANGA

2020

HERRAMIENTA PARA EL DISEÑO DE SISTEMAS SOLARES FOTOVOLTAICOS BASADA EN REDES NEURONALES ARTIFICIALES (RNA) PARA DETERMINAR LA CONFIGURACIÓN, DIMENSIONAMIENTO, SELECCIÓN DE EQUIPOS Y ARREGLOS FOTOVOLTAICOS EN COLOMBIA.

HAROLD OSWALDO OCHOA BUITRAGO

FABIAN YESID RAMÍREZ LEON

Trabajo de grado para optar el título de

Ingeniero en Energía

Director:

PhD. LUIS SEBASTIAN MENDOZA CASTELLANOS

Codirectores:

PhD. ANA LISBETH NOGUERA GALINDO

PhD CARLOS JULIO ARIZMENDI PEREIRA

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA

FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS

PROGRAMA DE INGENIERÍA EN ENERGÍA

BUCARAMANGA

2020

Nota de aceptación

_____

_____

_____

_____

_____

_____

_____

_____

_____

Firma de director

_____

_____

Firma de los calificadores

Bucaramanga, Septiembre de 2020

Contenido

Lista de figuras

Lista de tablas

Lista de símbolos

Símbolo	Parámetro	Símbolo de la unidad
$\boldsymbol{\lambda_1}$	Valor aleatorio	[-]
$\boldsymbol{\lambda}$	Valor de regulación	[-]
$\boldsymbol{\varphi}$	Latitud	[°]
$\boldsymbol{\varepsilon}$	Desempeño de la clasificación	
$\boldsymbol{\theta}$	Valor límite de la función de activación	
$\boldsymbol{\delta}$	Ángulo de declinación	[°]
$\boldsymbol{\Delta_i}$	Número de ejemplos pertenecientes a la clase mayoritaria	[u]
$\boldsymbol{\alpha}$	Largo del paso	[-]
$\boldsymbol{\beta}$	Ángulo del panel	[°]
$\boldsymbol{\sigma}$	Cantidad de vecinos tomados (Ancho del núcleo)	[u]
$\boldsymbol{a}$	Estado de activación	
α_l	Ángulo de elevación	[°]
$\boldsymbol{Azimut}$	Angulo azimut	[°]
$\boldsymbol{D_w}$	Distancia entre dos datos	[u]
$\boldsymbol{D}$	Conjunto de datos	
$\boldsymbol{d_{th}}$	Grado de desbalance máximo	[-]
$\boldsymbol{dc}$	Cantidad de características	[u]
$\boldsymbol{d}$	Grado de desbalance de la clase minoritaria	[-]
$\boldsymbol{r}$	Tasa	[-]
$\boldsymbol{EoT}$	Ecuación del tiempo	[min]
$\boldsymbol{f_{act}}$	Función de activación	
$\boldsymbol{GMT}$	Diferencia del tiempo local con las coordenadas del tiempo universal	[h]
$\boldsymbol{G}$	Cantidad de datos sintéticos que se deben generar.	[u]

Símbolo	Parámetro	Símbolo de la unidad
g	Cantidad de datos sintéticos por clase mayoritaria	[u]
HRA	Ángulo horario	[°]
I_{Inv}^{max}	Corriente máxima del inversor	[A]
$I_{scPanel}$	Corriente de corto circuito del panel	[A]
l	Número de la variable	
$LSTM$	Meridiano del tiempo local estándar	[°]
LT	Tiempo local	[min]
m_l	Clase mayoritaria	[u]
m_s	Clase minoritaria	[u]
$MPPT_{High}$	Voltaje máximo para MPPT	[V]
$MPPT_{Low}$	Voltaje mínimo para MPPT	[V]
net	Función de la red de entrada/Promedio ponderado	
η	Valor de error mínimo	
n	Numero día del año	[u]
N_{PMax}	Número máximo de paneles en paralelo	[u]
$N_{SMinmpp}$	Número mínimo de paneles en serie	[u]
$N_{sMaxmpp}$	Número máximo de paneles en serie	[u]
p	Probabilidad de que un punto seleccione a otro como referencia	[%]
p_{ac}	Potencia máxima de CA para el inversor en condiciones normales	[W]
p_{dc}	Nivel de potencia de CC en el que se alcanza la potencia nominal en Corriente Alterna	[W]
p_{nt}	Potencia consumida por el inversor cuando no hay corriente de entrada.	[W]
p_{so}	Potencia requerida para iniciar el proceso de inversión	[W]
s	Datos sintéticos	[u]

Símbolo	Parámetro	Símbolo de la unidad
$\boldsymbol{t}$	Posición	
$\boldsymbol{Tc}$	Factor corrección del tiempo	[min]
$\boldsymbol{v_{mp}}$	Voltaje de máxima potencia del panel	[V]
$\boldsymbol{v_{oc}}$	Voltaje de circuito abierto del panel.	[V]
$\boldsymbol{v_{ac}}$	Voltaje nominal en CA de la salida del inversor	[V]
$\boldsymbol{V_{MPPT}^{max}}$	Voltaje máximo del MPPT	[V]
$\boldsymbol{V_{MPPT}^{min}}$	Voltaje mínimo del MPPT	[V]
$\boldsymbol{V_{panel}^{max}}$	Voltaje máximo del panel	[V]
$\boldsymbol{w}$	Peso del error	[%]
$\boldsymbol{x}$	Punto	
y	Etiquetas o clases de los puntos	

Lista de acrónimos

ADASYN	Adaptative Synthetic Sampling Approach for Imbalanced Datasets
ANLA	Agencia Nacional de Licencias Ambientales
CA	Corriente Alterna
CC	Corriente Continua
FENOGE	Fondo de Energías No convencionales y Gestión Eficiente de la Energía
FNCER	Fuentes No Convencionales de Energía Renovable
FNCER	Fuentes No Convencionales de Energía Renovable
FSCNCA	Feature Selection for Classification Neighborhood Component Analysis
IDEAM	Instituto de Hidrología, Meteorología y Estudios Ambientales
MSE	Error cuadrático medio
NREL	Laboratorio Nacional de Energías Renovables
SAM	System Advisor Model
SMOTE	Synthetic Minory Oversampling Technique
SNL	Sandia National Laboratories
UPME	Unidad de planeación minero-energética

Título: Herramienta para el diseño de sistemas solares fotovoltaicos basada en redes neuronales artificiales (RNA) para determinar la configuración, dimensionamiento, selección de equipos y arreglos fotovoltaicos en Colombia.

Resumen:

En este proyecto se desarrolló una herramienta que consta de una Red Neuronal Artificial (RNA) que brinda apoyo en el dimensionamiento y proyección de los sistemas fotovoltaicos. Esto fue realizado, mediante una estimación de la configuración eléctrica de algunos proyectos registrados ante la UPME y los datos de ingeniería de detalle que se documentan en la Agencia Nacional de Licencias Ambientales (ANLA). Para la propuesta se implementó una técnica de inteligencia artificial, aprendizaje de máquina supervisado. Para esto, fue implementada una variación del algoritmo K vecinos cercanos, por medio de la función FSCNCA por sus siglas en inglés "*Feature Selection for Classification using Neighborhood Component Analysis".* Se usó la técnica de redes neuronales que permitieron el dimensionamiento, diseño y clasificación de los proyectos solares fotovoltaicos. Adicionalmente se implementó un árbol de decisión, que permitió seleccionar más aproximadas a los requerimientos área, potencia de diseño y presupuesto destinado a paneles e inversores.

Autores: Harold Oswaldo Ochoa Buitrago, Fabian Yesid Ramírez León.

Palabras claves: Redes neuronales, árbol de decisión, grid tie, sistema fotovoltaico conectado a la red.

Title: Tool for the design of photovoltaic solar systems based on artificial neural networks (ANN) to determine the configuration, dimensioning, selection of photovoltaic equipment and arrangements in Colombia.

Abstract:

This project, a tool was developed that consists of an Artificial Neural Network (ANN) that provides support in the dimensioning and projection of photovoltaic systems. This was done by estimating the electrical configuration of some projects registered with the UPME and the detailed engineering data that is documented in the National Environmental Licensing Agency. For the proposal, an artificial intelligence technique, supervised machine learning, was implemented. For this, a variation of the K-nearest neighbors algorithm was implemented, by means of the FSCNCA function for its acronym in English "*Feature Selection for Classification using Neighborhood Component Analysis*". The neural network technique was used that allowed the dimensioning, design and classification of the photovoltaic solar projects. Additionally, a decision tree was implemented, which allowed to select more approximate to the area requirements, design power and budget destined to panels and inverters.

Authors: Harold Oswaldo Ochoa Buitrago, Fabian Yesid Ramírez León.

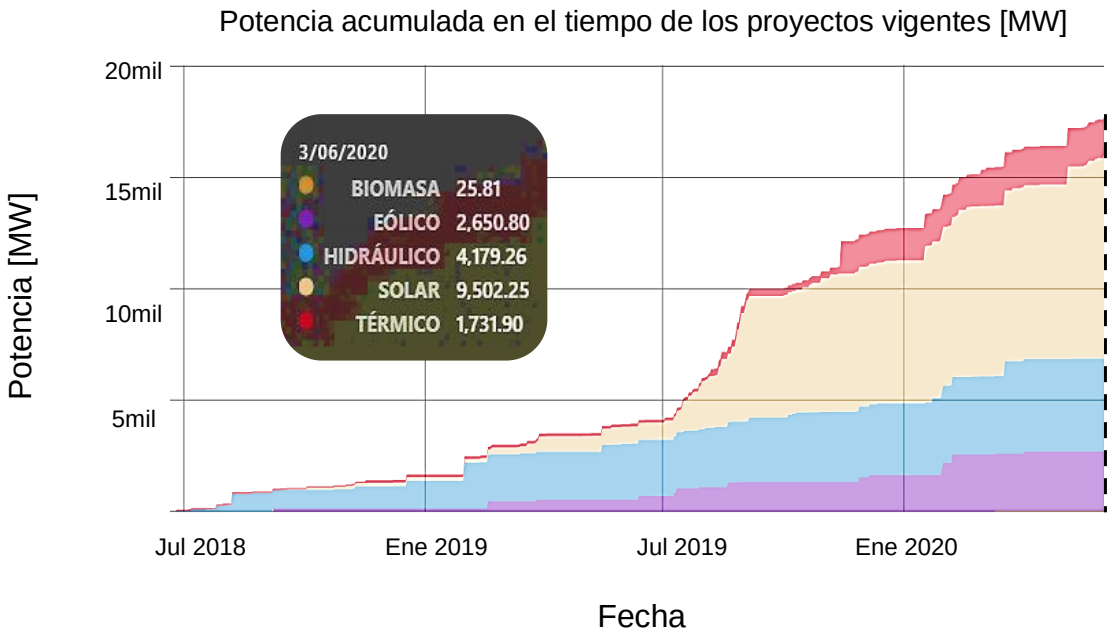
Key words: Neural networks, decision tree, grid tie, grid-connected system.

Introducción

En Colombia la generación de energía fotovoltaica está en continuo crecimiento y esto se evidenció en la subasta del 2019 que adjudicó 1298,9 MW de Fuentes No Convencionales de Energía Renovable (FNCER) de los cuales 17.39% correspondieron a solar fotovoltaica [1]. En mayo de 2020, se registraron 437 proyectos ante la unidad de planeación minero-energética UPME de los cuales 317 corresponden generación con energía solar Fotovoltaica. La Figura 1, presenta los proyectos de generación de energía eléctrica, para potencias de 18.09 [GW], de los cuales 9.50 [GW] son generados con energía solar fotovoltaica, representando un 52.5%. Otros proyectos tales como: centrales hidráulicas aportan 18 [GW] (23.1%), eólica aporta 2.65 [GW] (14.7%), las centrales térmicas aportan 1731.9 [MW] (9.6%) y por ultimo los proyectos de biomasa, siendo los que menos aportan, contribuyen 25.81 [MW] (0.1%).

En el periodo de mayo de 2019 a mayo de 2020, los proyectos solares por capacidad de generación de potencia que fueron registrados ante la UPME, aumentaron 12 veces con respecto al periodo anterior, pasando de 776.43 [MW] a 9329 [MW], se distribuyen en las fases de prefactibilidad (fase 1), factibilidad (fase 2) e ingeniería de detalle (fase 3), tal como se indica en las resoluciones UPME 638 de 2009 y 143 de 2016 para el registro de proyectos de generación de energía eléctrica. Los departamentos con mayor registro de proyectos de generación de energía eléctrica fueron: Valle del Cauca con 43 proyectos, Cesar con 22, Santander con 21 y los departamentos de Meta, Tolima y Huila con 16 proyectos.

Figura 1 Registro de proyectos de generación de electricidad 2020.



Fuente: UPME [2].

Con todos estos datos registrados ante la UPME y la ANLA puede aplicarse técnicas de inteligencia artificial para mejorar los procesos de diseño en proyectos energéticos, las redes neuronales son buena opción debido a que se adapta bien a diferentes conjuntos de datos. En la aplicación a la generación con solar fotovoltaica puede aplicarse para encontrar coeficientes y variables para el proceso dimensionado y clasificación.

En el dimensionamiento de sistemas solares fotovoltaicos es necesario tener en cuentas, parámetros climáticos tales como: radiación solar, velocidad del viento, energéticas (demanda de energía, cargas conectadas, etc.) y financieras (costo de los equipos, ahorro de energía, etc.), adicionalmente tener en cuenta la variedad de equipos para sistemas solares fotovoltaicos disponibles en el mercado, y el desempeño de los mismos a lo largo del tiempo. Para agilizar el dimensionamiento de proyectos solares fotovoltaicos, es necesario el uso de herramientas computacionales que permitan evaluar las características ya mencionadas, algunas de las herramientas computacionales más usadas, son por ejemplo PVSyst, HelioScope y System Advisor Model (SAM), en las cuales el tiempo que se toma para dimensionar puede variar según la experiencia del diseñador con la herramienta.

1. Aspectos generales del proyecto

1.1 Planteamiento del problema

El desarrollo de los proyectos fotovoltaicos en Colombia es de interés nacional, por ello el gobierno da incentivos tributarios mediante el decreto 829 de 2020 y el decreto 030 de la CREG a quien adopte este tipo de proyectos que hacen parte de Fuentes No Convencionales de Energía Renovable (FNCER) mediante la ley 1715 De 2014. Existen los fondos disponibles para la financiación y desarrollo de estos proyectos como el Fondo de Energías No convencionales y Gestión Eficiente de la Energía (FENOGE).

El uso de las herramientas computacionales se ha incrementado, debido a la necesidad de los diseñadores de probar distintos diseños y evaluar rendimientos del sistema junto con los beneficios económicos. Es necesario buscar alternativas de calculo que permitan realizar diseños de sistemas fotovoltaicos. Las herramientas actuales, como HelioScope y PV*SOL dimensionan los sistemas a partir de equipos seleccionados por el diseñador. Siendo necesario realizar comparaciones con las diferentes referencias de cada marca que hay en el mercado para encontrar cual configuración brinda más beneficios y constituye menores costos.

1.2 Objetivo principal

Desarrollar una herramienta computacional que por medio de una red neuronal determine la configuración de los equipos básicos (Panel e inversor) según los parámetros de entrada.

1.3 Objetivos específicos

- Realizar un levantamiento de proyectos fotovoltaicos en Colombia, como fuente de información para clasificar los tipos de instalaciones mediante redes neuronales.
- Identificar las variables relevantes de los proyectos solares fotovoltaicos para su dimensionamiento y clasificación.
- Seleccionar las arquitecturas de las redes neuronales que permita clasificar el tipo de instalación y determinar la configuración del arreglo fotovoltaicos a partir de los datos de entrenamiento.

1.4 Alcances

El presente proyecto tiene como alcance el desarrollo de una herramienta computacional que se base en los datos técnicos y comerciales de proyectos registrados y disponibles al público en la UPME y la Autoridad de Licencias Ambientales ANLA, con la capacidad de seleccionar los equipos básicos (paneles e inversores) y la configuración de paneles, así como la estimación de costos en proyectos solares fotovoltaicos conectados a red para Colombia.

1.5 Limitaciones

Para cumplir el objetivo, el proyecto propuesto se debe tener en cuenta la cantidad y calidad de los datos que van a ser utilizados por el programa, la localización del proyecto, la marca y equipos seleccionados, el tipo de técnica de aprendizaje implementado. A continuación, se detalla cada uno de las limitaciones:

- ✓ Cantidad de datos: El proyecto está limitado por el volumen de datos y en este caso se implementará una cantidad superior a cincuenta y cinco datos reales, y cuarenta y cinco datos sintéticos. Con la cantidad actual de datos reales, no hay un panorama suficientemente amplio, es posible que existan desviaciones en la clasificación de proyectos atípicos a la hora de ejecutar el programa.

- ✓ Localización: El proyecto está delimitado geográficamente para Colombia que comprende 1142 millones [km²] de territorio.

- ✓ Marcas y equipos seleccionados: Los equipos fueron extraídos a partir de la herramienta PVLib [3],siendo posible que algunas referencias y marcas de

equipos no son muy usados o no estén disponibles para el país y además que sus precios no sean iguales, debido a que estos fueron generados a partir de algunos equipos de referencia. Se limita únicamente a paneles solares e inversores.

- ✓ Técnicas usadas: Este proyecto se limita únicamente a trabajar con aprendizaje supervisado y a los algoritmos utilizados, tales como redes neuronales, árbol de decisión y K vecinos cercanos.

- ✓ Geometrías y terrenos: Existe una gama de posibles geometrías y suelos en los que se hacen dichos proyectos. Se considerarán únicamente sitios planos y geometrías cuadradas. Aunque se puede ejecutar el programa con el área total de otro tipo de geometrías, es posible que el resultado del número de paneles no sea el adecuado debido a que no se toman en cuenta las irregularidades del terreno.

1.6 Justificación

La minería de datos es una rama de la estadística y de la ciencia de la computación, que permite extraer información de volúmenes de datos, con dicha información se puede ajustar un modelo que clasifique o pronostique. Este proyecto tiene como finalidad desarrollar una herramienta para el dimensionamiento de sistemas fotovoltaicos conectados a la red, únicamente tomando en cuenta paneles e inversores de distintas marcas, seleccionando la configuración que más se ajuste a las necesidades del usuario, en cuanto a la potencia requerida, el área y el dinero destinado para el proyecto. Se propone hacer un levantamiento de una muestra de datos sobre proyectos solares fotovoltaicos en Colombia, existentes hasta el año 2020, y entrenar dos redes neuronales para el dimensionamiento y clasificación de los proyectos. Posteriormente es necesario implementar un árbol de decisión que permita seleccionar las configuraciones más adecuadas y con menor costo.

2. Marco referencial

2.1 Antecedentes

En el artículo escrito por Mellit et al.[4] titulado como “Dimensionamiento de sistemas fotovoltaicos autónomos utilizando un modelo de red neuronal adaptativa”, se hace uso de una red neuronal de función de base radial adaptativa, con el fin de encontrar un modelo que se ajuste a los 2 coeficientes necesarios para el dimensionamiento de la planta fotovoltaica, capacidad de la matriz fotovoltaica y capacidad de almacenamiento. La arquitectura de la red neuronal usada, tiene 2 neuronas en la capa de entrada, en la capa oculta tiene 8 neuronas y en la capa de salida tiene 2 neuronas. Se usó un grupo de 200 datos, donde 90% de los datos se usaron para el entrenamiento, 5% para validación de la red neuronal y 5% para evaluación de la red neuronal. Al terminar de entrenar la red neuronal, se obtuvo un coeficiente de determinación (R^2) de 0.97.

En el artículo escrito por Hontoria et al. [5] titulado como “Un nuevo enfoque para dimensionar sistemas fotovoltaicos autónomos basado en redes neuronales”, se desarrolló una red neuronal para predecir los valores de la capacidad de la batería, según los valores de la capacidad del generador, la probabilidad de pérdida de carga y el índice de claridad. Para entrenar la red neuronal se recolectó datos de varios sistemas fotovoltaicos autónomos simulados y reales, donde varía la capacidad de la batería y la capacidad del generador, obteniendo la probabilidad de perdida de carga. La arquitectura de la red neuronal seleccionada consta de una capa de entrada con 3 neuronas, donde cada una de las neuronas corresponde a la capacidad del generador, la probabilidad de pérdida de carga y el índice de claridad, es seguida por una capa oculta de 9 neuronas, y una capa de salida de 1 neurona, siendo correspondiente al valor de capacidad de la batería. Se comparó los valores reales y los valores pronosticados por la red neuronal, y se obtuvo un error cuadrático medio (MSE) entre 0.035 y 0.073

En el artículo escrito por Alomari et al [6] titulado como “Pronóstico de energía solar fotovoltaica en Jordania utilizando redes neuronales artificiales”, se relaciona la irradiancia solar y la potencia de salida de la planta fotovoltaica, utilizan una red neuronal para pronosticar la potencia de salida de las siguientes 24 horas en una planta de 264 [kWp], situada en la azotea de la facultad de ingeniería de Applied Science Private University. Utilizando una base de datos de 19800 registros de energía producida por la planta y 20808 registros de estaciones meteorológicas. Se usó una red neuronal con 5 neuronas de entrada, una capa oculta de 22 neuronas y una neurona de salida, para el entrenamiento de la red neuronal, se usó el 80% de los datos, mientras para la validación se usó el 15% de los datos, el 5% de los datos restantes se usó para evaluar el sistema. Al predecir la potencia de los siguientes 10 días, se obtiene una raíz del MSE de 0.0333 y un R2 de 0.9965.

En el artículo escrito por Kumar Amit et [7] titulado como “Predicción de la radiación solar utilizando técnicas de redes neuronales artificiales: una revisión”, los datos de

radiación solar juegan un papel importante en la investigación de la energía solar. Estos datos no están disponibles para la ubicación de interés debido a la ausencia de una estación meteorológica. El objetivo principal del estudio es revisar las técnicas basadas en redes neuronales artificiales (ANN) con el fin de identificar los métodos adecuados disponibles en la literatura para la predicción de la radiación solar e identificar las lagunas de investigación. En el artículo se mostró que las técnicas de redes neuronales artificiales predicen la radiación solar con mayor precisión en comparación con los métodos convencionales. La precisión de predicción de los modelos ANN depende de las combinaciones de parámetros de entrada, el algoritmo de entrenamiento y las configuraciones de la arquitectura.

En el artículo escrito por Fermín rodríguez et al [8] titulado “Predicción de energía solar a través de redes neuronales artificiales utilizando previsiones meteorológicas para el control de microrredes” se creó una herramienta para predicción de parámetros que intervienen en la producción de energía solar para estimar la producción de irradiación futura y optimizar el control de red. Se utilizó una arquitectura con 146 entradas de temporada, tiempo e irradiación, una salida correspondiente a irradiación, una capa oculta con las funciones de activación log-sigmoide y lineal, el algoritmo de aprendizaje Levenberg-Marquardt. La diferencia entre energía producida y la estimada fue de 0,5e-9%.

2.2 Marco teórico.

En esta sección se muestra brevemente los temas que fueron tomados en cuenta en el desarrollo del proyecto, comenzando con el movimiento aparente del sol y finalizando con redes neuronales.

2.2.1 Movimiento aparente del sol

Al hacer dimensionamientos fotovoltaicos, es necesario tener en cuenta las sombras producidas por los paneles solares, debido a que ellas pueden variar según la ubicación geográfica, se tomó en cuenta el movimiento aparente del sol. Según Kalogirou Soteris [9] para calcular el movimiento aparente del sol es necesario computar:

Ecuación del tiempo (EoT): La ecuación (1) y (2) describen el tiempo que varía el planeta tierra a lo largo de los días (n) del año, principalmente debido su excentricidad.

$$EoT = 9.87sen(2B) - 7.53cos(B) - 1.5sen(B) \quad (1)$$

$$B = \frac{360}{365} * (n - 81) \quad (2)$$

Hora local meridiano estándar ($\boldsymbol{LSTM}$): En la ecuación (3) muestra el meridiano de referencia similar al de Greenwich para una zona horaria en particular. Donde (ΔGMT) es la diferencia entre la hora local y hora de Greenwich.

$$LSTM = 15 * (\Delta GMT) \tag{3}$$

Factor de corrección del tiempo ($\boldsymbol{TC}$): En la ecuación (4) se representa la variación de la hora solar local en una zona horaria por variaciones de longitud.

$$TC = 4 * (Longitud - LSTM) + EoT \tag{4}$$

Tiempo Solar Local ($\boldsymbol{LST}$): La ecuación (5) une las correcciones del factor de tiempo y las une al tiempo local (LT).

$$LST = LT + \frac{TC}{60} \tag{5}$$

Ángulo Horario ($\boldsymbol{HRA}$): Número de grados que el sol se mueve a través del cielo, negativo en la mañana y positivo en la tarde, como se representa en la siguiente ecuación.

$$HRA = 15 * (LST - 12) \tag{6}$$

Ángulo de declinación ($\boldsymbol{\delta}$): Es el ángulo entre el ecuador y una línea desde el centro del sol al centro de la tierra.

$$\delta = -23.45 * \cos\left(\frac{360}{365} * (n + 10)\right) \tag{7}$$

Ángulo de elevación ($\boldsymbol{\alpha}$): Altura angular del sol en el cielo medido desde la horizontal.

$$\alpha = Sen^{-1}(sen\delta * sen\varphi + cos\delta * cos\varphi * \cos(HRA)) \tag{8}$$

Donde la latitud (φ) del lugar de interés. para las ubicaciones del hemisferio norte es positivo y negativo para el hemisferio sur.

Ángulo Azimut: Ángulo del cual proviene la luz del sol, el norte con 0° y el sur con 180°.

$$Azimut = cos^{-1} * \left(\frac{sen\delta * cos\varphi - cos\delta * sen\varphi * \cos(HRA)}{cos\alpha_l}\right) \tag{9}$$

2.2.2**Sistemas fotovoltaicos**

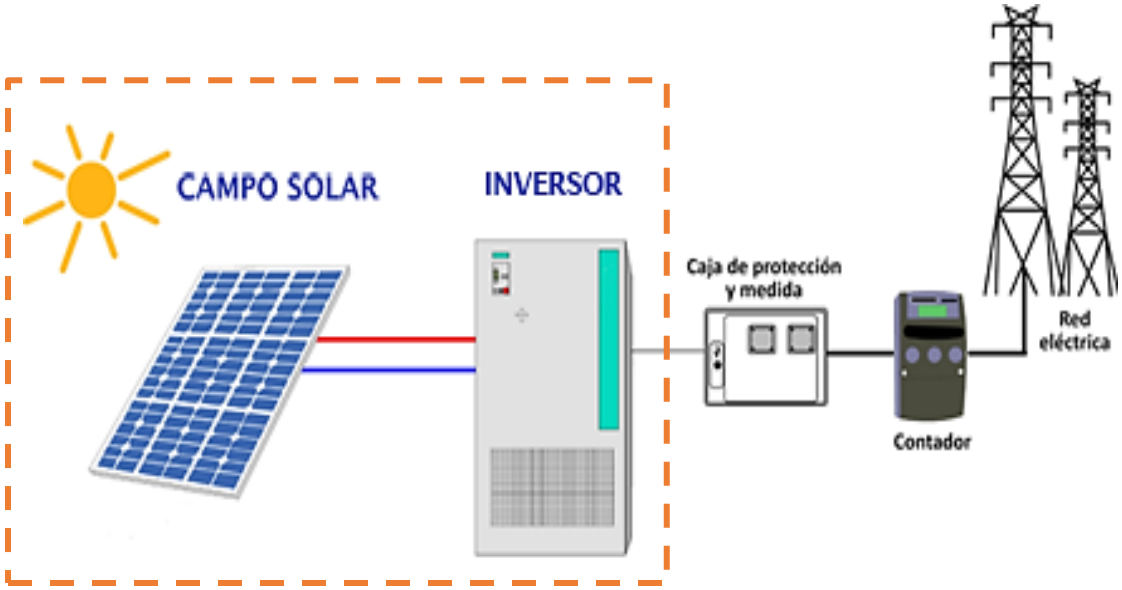
Los sistemas fotovoltaicos se clasifican en dos grupos de acuerdo al objetivo del proyecto: Instalaciones conectadas a la red eléctrica, que tienen el propósito de

suministrar la energía eléctrica al sistema interconectado e instalaciones aisladas de la red eléctrica, que tiene la finalidad de satisfacer total o parcialmente la demanda de energía eléctrica a pequeñas comunidades para el autoconsumo [10]. Este proyecto, se enfoca a los sistemas fotovoltaicos conectados a red ya que permite simplificar el uso de las baterías eléctricas.

2.2.3 **Sistemas fotovoltaicos conectados a la red**

Son los sistemas que, una vez generada la energía eléctrica, la pueden consumir o cederla a la red eléctrica [10] .En la Figura 2 se seleccionan los componentes de los sistemas fotovoltaicos conectados a la red que son tomados en cuenta durante de desarrollo de la tesis.

Figura 2 Esquema sistema fotovoltaico conectado a la red.



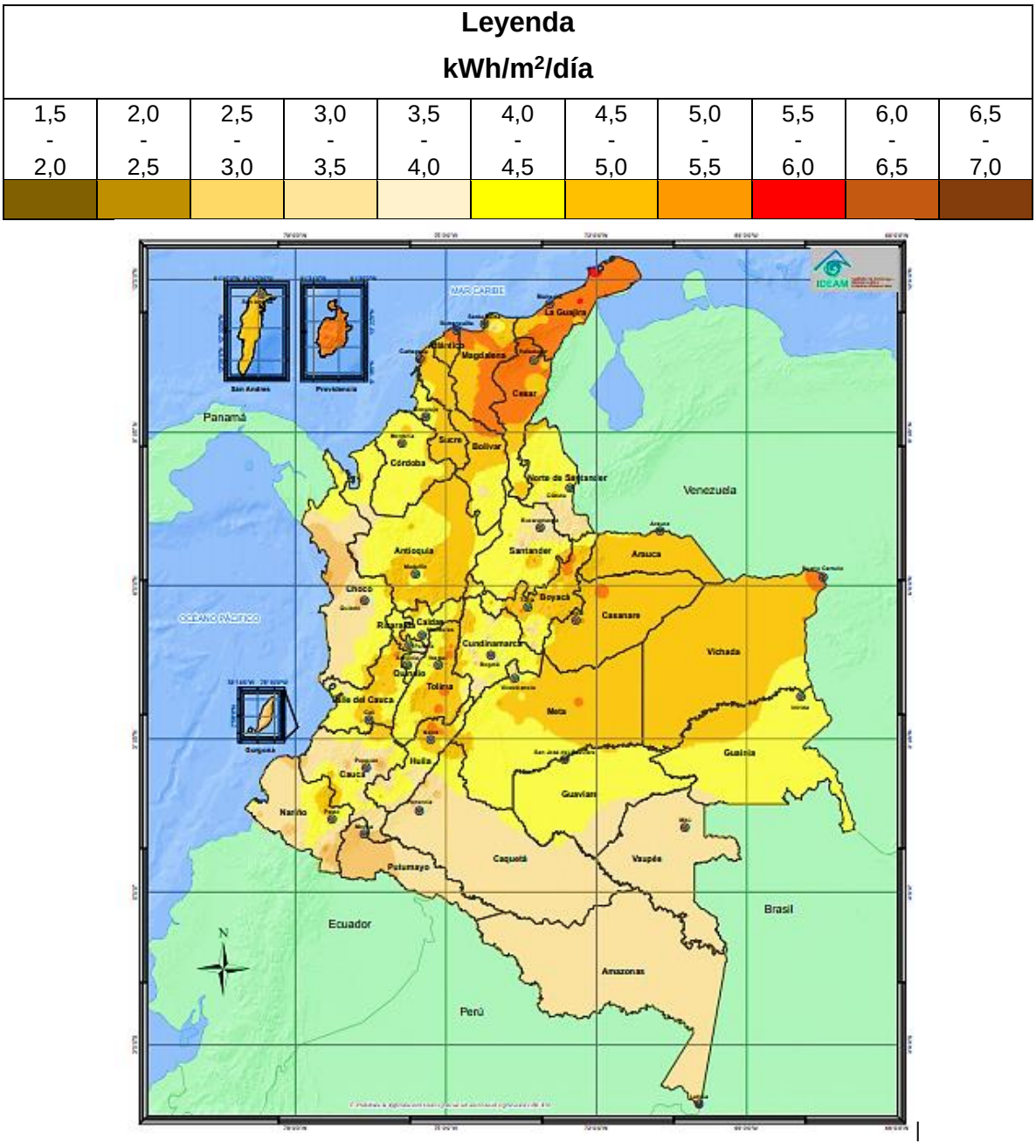
Fuente: Coll M. [11]

2.2.4**Dimensionamiento de sistemas fotovoltaicos conectados a la red**

Es una parte importante a la hora de diseñar proyectos solares fotovoltaicos, ya que permite determinar características importantes del sistema que se piensa instalar como lo son: los tipos de paneles, inversores, cableado, protecciones, pérdidas, cantidad de quipos y potencias en Watts producidas por los mismos en ciertas condiciones. Para caracterizar el sitio donde se piensa construir el proyecto fotovoltaico y para la evaluación del sistema, según los equipos seleccionados, es necesario pasar por una serie de pasos.

Sitio: Principalmente es para determinar las condiciones meteorológicas y la radiación solar del sitio, algunas bases de datos, como la base de datos del IDEAM da una visualización de la radiación solar en distintas partes de Colombia, como se muestra en la Figura 3.

Figura 3 Mapa de Colombia con radiación solar.



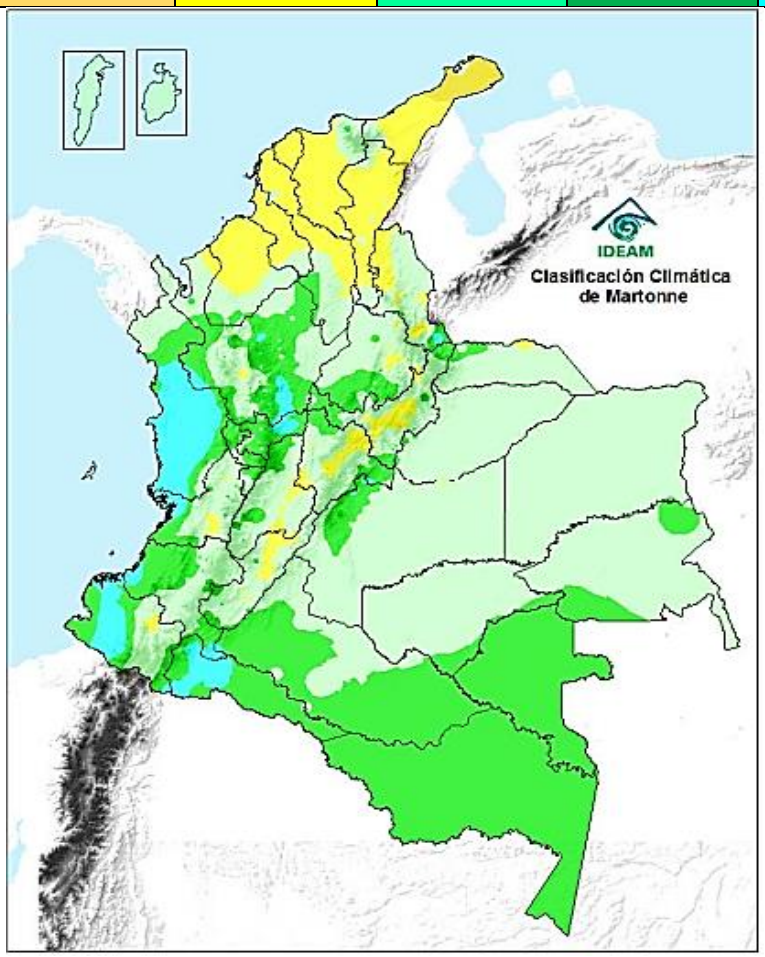
Fuente: IDEAM [12]

Los sitios de mayor radiación solar están principalmente en los departamentos del norte del país, de la Guajira, el Cesar, Magdalena y Atlántico. En la parte central y en lo llanos, la radiación, aunque es de menor intensidad también son sitios de alto

potencial y atractivos para proyectos fotovoltaicos, departamentos como: el Meta, Casanare, Antioquia y Tolima. Por último, en el sur del país y en algunas partes del occidente exceptuando el Valle del Cauca, la radiación solar es relativamente baja y son zonas que presentan un alto grado de humedad y lluvia. En la Figura 4 muestra el mapa de clasificación climática de Martonne donde el valor en la leyenda corresponde a la relación entre la precipitación en mm de agua y la temperatura en [°C].

Figura 4 Clasificaciones climáticas de Colombia.

Leyenda					
Árido	Semiárido	Subhúmedo	Húmedo	Húmedo lluvioso	Húmedo lluvioso, sin diferencias estacionales, todo el año
0-5	5-10	10-30	30-70	70-150	>150

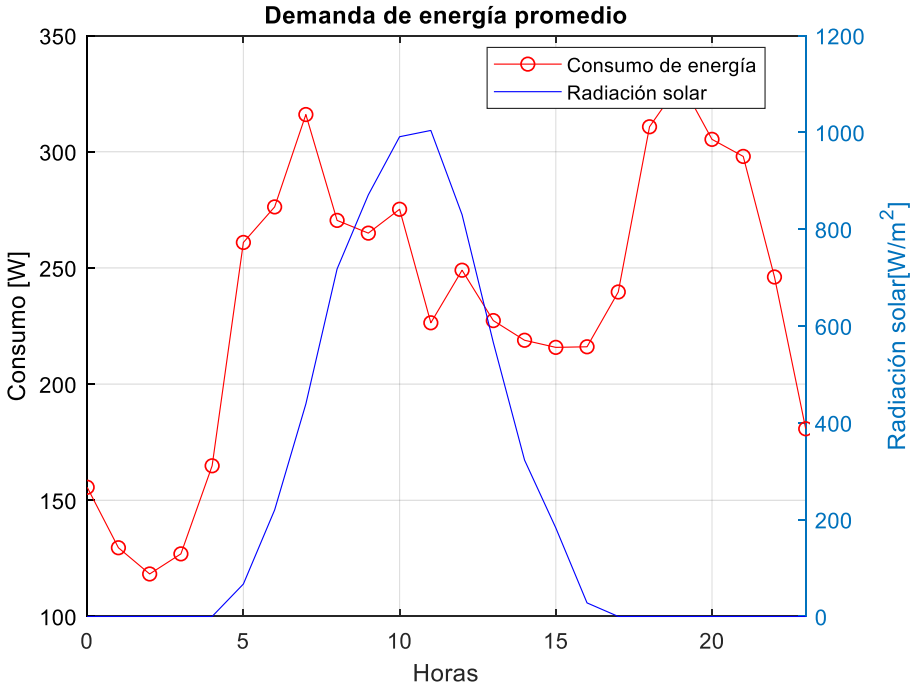


Fuente: IDEAM [13]

Demanda de energía: Para determinar la demanda de energía consumida, es necesario calcular los kWh/día en función del tiempo de operación o solicitar la curva de demanda a la comercializadora de energía. Finalmente, el proyecto debe cumplir los lineamientos de las normas para instalaciones eléctricas que en el caso de Colombia es el RETIE y la norma NTC 2050.

Con los datos de consumo se puede hacer una visualización del comportamiento de la carga, e identificar los picos de demanda durante el día y determinar la potencia a instalar. En la Figura 5 se muestra un ejemplo de la demanda de energía de un hogar promedio de estrato 4 [14] , con el objetivo de mostrar las fluctuaciones de energía que hay durante el día, y se compararó con la radiación solar en un día promedio.

Figura 5 Demanda de energía de un hogar promedio VS Radiación solar.



Fuente: Hernández y Carrillo [14].

Configuración del arreglo fotovoltaico: Es el proceso en el que se detallan características del sistema fotovoltaico.

a) **Posición del panel:** Según Lamigueiro [15] para determinar el ángulo de inclinación del panel (β) , mostrado en la ecuación (10) es relacionada la latitud en la que está el panel con una constante.

$$\beta = 3.7 + 0.69 * |latitud| \quad (10)$$

La ecuación (11) según Stems [16] para determinar la sombra del panel, es necesario tener en cuenta su longitud (lt), su ángulo del panel (β) y su posición

geográfica, ya que el movimiento aparente del sol puede cambiar según la latitud y longitud donde se esté ubicado.

$$d_{sombra} = lt * (\cos(\beta) + \text{sen}(\beta) * \cot(\alpha) * \cos(Azimuth_{Panel} - Azimuth) \quad (11)$$

b) **Cálculos para el arreglo:** Según los equipos disponibles, se identifica los inversores y paneles disponibles, a fin de seleccionar la combinación más conveniente. Siempre debe estar a la mano la ficha técnica de los equipos seleccionados, es una parte esencial para su correcto dimensionamiento.

Según los datos anteriores, se calculan las cantidades máximas y mínimas de paneles solares por cada lazo del inversor. Según Lamigueiro [15] para encontrar la cantidad mínima de paneles solares en serie por cada MPPT ($N_{SMinmpp}$), se utiliza la ecuación (12), donde se hace una relación entre el voltaje mínimo de entrada del inversor (V_{MPPT}^{min}) y el voltaje máximo que puede generar el panel (V_{panel}^{max}).

$$N_{Sm} = \frac{V_{MPPT}^{min}}{V_{panel}^{max}} \quad (12)$$

Para encontrar la cantidad máxima de paneles en serie por MPPT ($N_{sMaxmpp}$) se usó la ecuación (13) , donde se relaciona el voltaje máximo del MPPT (V_{MPPT}^{max}) y el voltaje máximo que puede generar el panel (V_{panel}^{max}).

$$N_{sM} = \frac{V_{MPPT}^{max}}{V_{panel}^{max}} \quad (13)$$

Para encontrar la cantidad máxima de paneles en paralelo por inversor (N_{PMax}), se usó la ecuación (14) , donde se relaciona la corriente máxima soportada por el inversor (I_{Inv}^{max}) y la corriente de corto circuito del panel ($I_{scPanel}$).

$$N_{PM} = \frac{I_{Inv}^{max}}{I_{scPanel}} \quad (14)$$

2.2.5 **Inteligencia artificial**

Esta sección hace una breve introducción a la inteligencia artificial, comenzando con una definición de la inteligencia artificial sus distintas ramas. Knowles [17] define a la inteligencia artificial como el desarrollo de sistemas informáticos que tienen la capacidad de realizar acciones que requieran inteligencia humana. La inteligencia artificial tiene distintos campos de aplicación, como lo es la clasificación, la predicción, la optimización y la toma de decisiones [18]. En la Tabla 1, se describen las ramas de la inteligencia artificial y su campo de acción.

Tabla 1 Ramas de la inteligencia artificial.

Rama de la inteligencia artificial	Concepto	Uso	Ventajas y desventajas
Lógica difusa o borrosa	Parte del concepto de que existen verdades parciales, es decir hay puntos intermedios. [19]	• Sistemas de control. • Sistemas de decisión.	✓ Puede aplicarse con actuadores ✓ No lineal. ✓ Puede garantizar la estabilidad en circuito cerrado ✗ Puede llegar a ser difícil entender sus resultados. ✗ Estabilidad no garantizada. ✗ Requiere atención en sistema de control.[20]
Sistemas expertos	Es un sistema que trata de simular el razonamiento de un experto en cierta área de conocimiento. [21]	• Decisiones financieras. • Planificación.	✓ Proporciona explicaciones. ✓ Puede trabajar continuamente. ✓ Puede adaptase a nuevas condiciones. ✗ No tiene sentido común. ✗ Puede dar soluciones erróneas. ✗ Dificultad en hacer las reglas de inferencia.
Visión de computadora	Dotar a una computadora con la visión, a través de distintas técnicas, como el procesamiento de imágenes y el análisis de imágenes. [22]	• Reconocimiento óptico de caracteres. • Captura de movimiento.	✓ Alta precisión. ✓ Fácil de integrar. ✗ La sensibilidad de detección en algunas aplicaciones puede ser baja.[23] ✗ Tiene bajo desempeño. ✗ Necesita expertos.
Procesamiento del lenguaje natural	Hacer comprender a una computadora el lenguaje humano. [24]	• Traducción automática de textos. • Respuestas automáticas.	✓ Muy flexible. ✓ Representa fácilmente nuevos conceptos. ✓ Libre de expresión. ✗ Poco predecible. ✗ Puede requerir muchas palabras claves. ✗ No es compacto.
Aprendizaje de máquina	Lograr aprendizaje de una computadora mediante cantidades de datos y experiencia. [25]	• Marketing personalizado. • Antivirus.	✓ No necesita intervención humana. ✓ Identifica patrones rápidamente. ✓ Manejo de multidimensionalidad. ✗ Los datos adquiridos necesitan un

Rama de la inteligencia artificial	Concepto	Uso	Ventajas y desventajas
			procesamiento para quitar sesgos. ❌ Necesita recursos computacionales. ❌ Susceptible a errores.

2.2.6 Tratamiento de datos

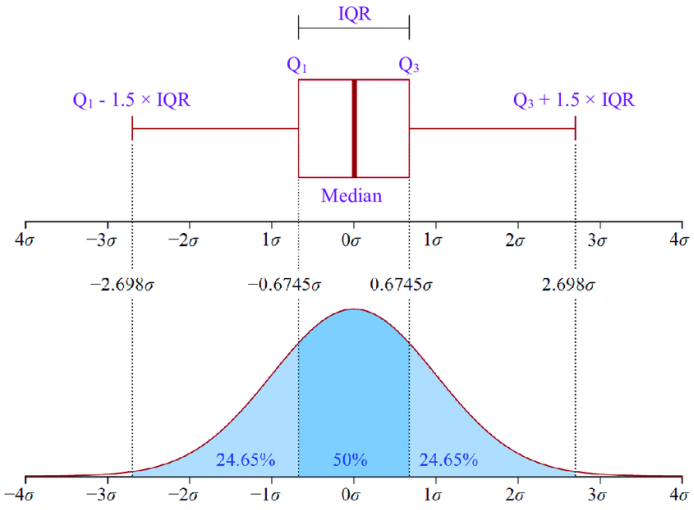
Para resolver problemas relacionados con datos, es necesario extraer de éstos información relevante que revele algún tipo de relación o de patrón. Para lograr esto, es necesario llevar a cabo un proceso que relaciona partes de la estadística con la programación.

Visualización de datos

La visualización de datos principalmente es para transmitir de forma gráfica los datos obtenidos de la ANLA. Dando una descripción de lo que hay en ellos, existen distintas formas de visualizar los datos y su información, se mostrarán las formas que se usaron en el desarrollo de esta tesis.

- Diagrama de cajas y bigotes: Es una manera de visualizar la concentración de los datos y de mostrar valores estadísticos, tales como los cuartiles, la media, el mínimo, el máximo y los valores atípicos. Se observa en la Figura 6 una distribución normal, dando a entender que el diagrama de cajas y bigotes puede pasar a ser una función de densidad ya que presenta una clara relación.

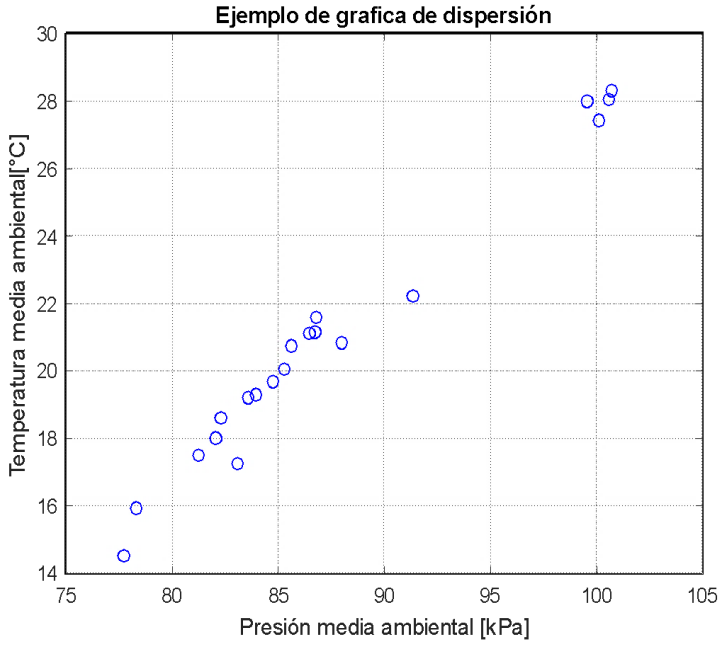
Figura 6 Relación caja y bigotes con función de densidad simétrica.



Fuente: Yang T. et al [26]

- Diagrama de dispersión: Es una manera de visualizar los datos en un plano cartesiano, cada punto representa a un proyecto. La Figura 7 muestra el diagrama de dispersión de los proyectos fotovoltaicos que fueron utilizados contra la temperatura promedio del sitio. Se pueden hacer dos observaciones, la primera es un claro comportamiento lineal y la segunda es que hay un grupo de puntos en la parte superior, relaciona cantidad de los proyectos que son próximos a la zona costera del país.

Figura 7 Muestra de diagrama de dispersión.



Fuente: Autor usando datos de la NASA.

Preprocesamiento de datos

Se encarga de mejorar la calidad de los datos[1], se hace por medio de limpieza de datos, llenado de datos y reducción de dimensionalidad [27]. En el desarrollo de la tesis, se hace este procedimiento con los datos de los proyectos solares fotovoltaicos (área, potencia, cantidad de emisiones de CO_2, etc.).

- Limpieza de datos: La limpieza de datos se hace sobre todo para prevenir errores y reconoce los puntos atípicos y las inconsistencias. Con esto se puede reducir el volumen de los datos, al eliminar datos erróneos y ruido, durante el desarrollo de la tesis, la variable que más valores atípicos obtuvo, fue la cantidad de emisiones de CO_2.
- Imputación de espacios faltantes: Se le conoce como imputación al sustituir los espacios faltantes con valores. Se suele sustituir el espacio faltante con la ayuda de

[1] Calidad de datos: Es una manera de medir la coherencia, confiabilidad y actualización de los datos.

la información cercana. Existen diversos métodos de imputación, observar en la Tabla 2.

Tabla 2 Algunos métodos de imputación.

Método de imputación	Explicación
Llenado manual	Es una opción en las bases de datos pequeñas, en las bases de datos grandes no es muy factible. [28]
Sustitución de la media	Los datos faltantes son reemplazados por la media, es recomendable usarla cuando hay menos del 10% de los datos faltantes. [29]
Imputación por regresión	Se hace regresión con los datos disponibles, los datos faltantes se incluyen al haber hecho la regresión. En este método la variabilidad y la covarianza se conservan bien. [29]
Interpolación lineal	Método simple que toma los 2 puntos finales más cercanos y los usa como estimación para los valores faltantes. [30]

Debido a la cantidad de datos faltantes en la matriz de datos, se optó por usar la interpolación lineal, según artículo de Junninena et al [30], la interpolación lineal comparado con otras técnicas de imputación, es una forma confiable y rápida de llenar brechas de datos.

Validación cruzada

Es un método utilizado en el desarrollo del proyecto, en la selección de características y para el entrenamiento de las redes neuronales. Según Berrar [31] se define como “Un método de remuestreo de datos para evaluar la capacidad de generalización de los modelos predictivos y evitar el sobre ajuste”. El funcionamiento del método se presenta en la Figura 8 , donde se observa que un conjunto de datos se divide en 3 partes iguales, la mayoría de datos es enviado a entrenar el modelo, mientras los demás esperan a validar el modelo y así dar su grado de generalización.

Figura 8 Ejemplo validación cruzada con 3 particiones

	Conjunto de datos		
Iteración 1	Entrenamiento	Entrenamiento	Validación
Iteración 2	Entrenamiento	Validación	Entrenamiento
Iteración 3	Validación	Entrenamiento	Entrenamiento

La validación cruzada suele dividir los conjuntos de datos en 3 parte, según Hastie et al [32] los conjuntos de datos se definen como:

1. Conjunto de entrenamiento: Es el conjunto de datos que el modelo tiene como ejemplo para ajustarse.
2. Conjunto de validación: Este conjunto de datos no ajusta el modelo. Su función es evaluar el grado de ajuste del modelo y evitar el sobre ajuste.
3. Conjunto de evaluación: Es el conjunto que se encarga de evaluar las respuestas del modelo y así seleccionar el modelo final.

2.2.7 **Selección de características**

La reducción de dimensionalidad es una forma de seleccionar las variables que más información aportan al modelo, se hace por medio de la sustracción de las variables que puedan ser redundantes y presenten una mayor dispersión, ayuda a incrementar la velocidad del modelo [28]. El modelo usado para seleccionar las características más relevantes del conjunto de datos de los proyectos solares fotovoltaicos es conocido por sus siglas en inglés Feature Selection for Classification/Regression Neighborhood Component Analysis for High Dimensional Data (FSCNCA). Según Yang et al [33] la distancia ponderada (D_w) mostrada en la ecuación (15), para un número de ejemplos (N) correspondiente a 100 proyectos, una cantidad de características (dc) correspondientes a la cantidad de paneles en serie, en paralelo, entre otras. Se mide la distancia de un punto, en la posición (x_{il}) a otro punto en la posición (x_{jl}), mientras se varía la relevancia de la característica, cambiando el peso (wl).

$$D_w(x_i, x_j) = \sum_{l=1}^{dc} wl^2 * |x_{il} - x_{jl}| \tag{15}$$

En la ecuación (16) se muestra la distribución de probabilidad, que relaciona el valor de entrada (z) con el ancho de kernel (σ), el cual determina el número de vecinos cercanos tomados. En la ecuación (17) muestra la probabilidad del proyecto fotovoltaico x_i seleccione al proyecto fotovoltaico x_j y se le designa con el símbolo p_{ij}.

$$k(z) = e^{\left(-\frac{z}{\sigma}\right)} \tag{16}$$

$$p_{ij} = \begin{cases} \dfrac{k\left(D_w(x_i, x_j)\right)}{\sum_{j \neq i} k\left(D_w(x_i, x_j)\right)}, para\ i \neq j \\ 0 \qquad\qquad\qquad\qquad\quad , para\ i = j \end{cases} \tag{17}$$

La ecuación (18) muestra la probabilidad de que el proyecto fotovoltaico x_i sea clasificado correctamente. El valor de y_{ij} puede variar según la clase de los proyectos fotovoltaicos seleccionados, si los dos son de la misma clase entonces y_{ij} tomará el valor de 1, en caso contrario tomará el valor de 0.

$$p_i = \sum_j y_{ij} * p_{ij} \tag{18}$$

La ecuación (19) muestra el desempeño de clasificación ε, el cual toma en cuenta la ecuación (18) para la correcta clasificación del proyecto fotovoltaico y se le resta el producto del valor del peso de la característica w_l y del valor de regularización λ. Una manera de prevenir el sobre ajuste, es modificando el valor de regularización λ por medio de la validación cruzada, hasta encontrar el valor que menor MSE produzca.

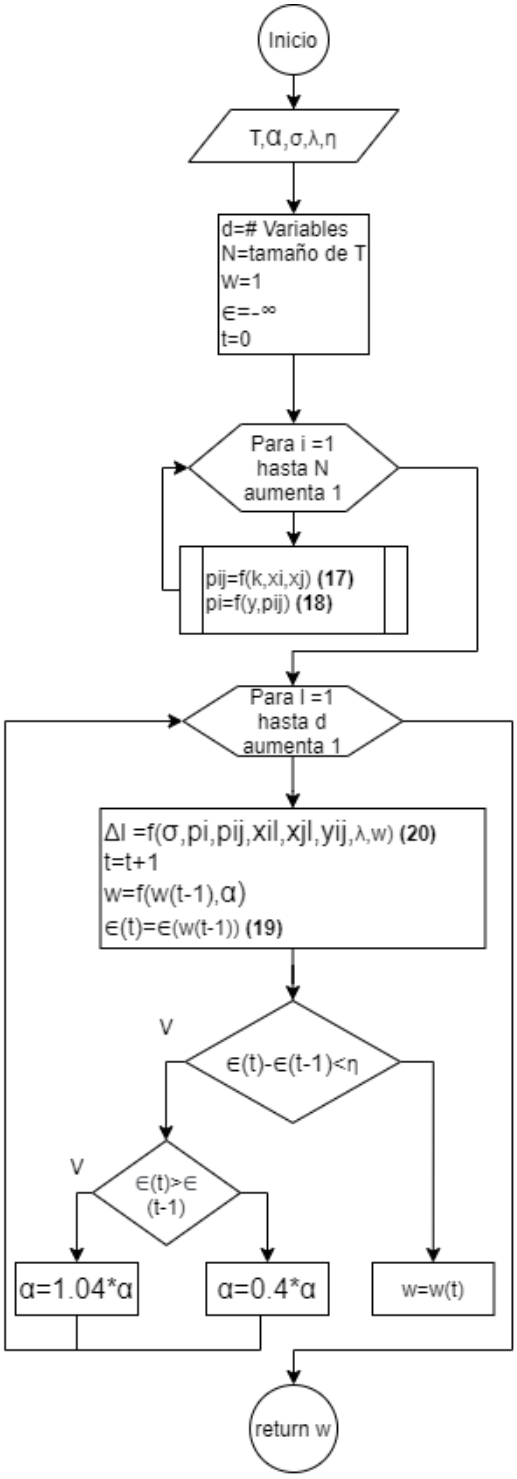
$$\varepsilon(w) = \sum_i \sum_j y_{ij} * p_{ij} - \lambda * \sum_{l=1}^{d} w_l^2 \tag{19}$$

Se usa el gradiente descendente para poder encontrar el valor de mínimo error, entonces se deriva la ecuación (19) y se obtiene la ecuación (20), la cual actualizará el valor de los pesos.

$$\Delta_l = 2\left(\frac{1}{\sigma} * \sum_i \left(\sum_{j \neq i} p_{ij} * |x_{il} - x_{jl}| - \sum_j y_{ij} * p_{ij} * |x_{il} - x_{jl}|\right) - \lambda\right) * w_l \tag{20}$$

En la Figura 9 se muestra el diagrama de flujo del algoritmo usado por Yang et al [33] para el desarrollo de FSCNCA y en la Tabla 3 se muestra el nombre de las variables utilizadas. El algoritmo se encarga encontrar el mínimo error de clasificación o regresión, variando los pesos correspondientes a cada una de las características, los pesos representan la relevancia de la variable, tomando como referencia la característica de respuesta.

Figura 9 Diagrama de flujo FSCNCA.



Fuente: Yang et al [33]

Tabla 3 Seudónimos usados en la selección de características

Variable	Símbolo de la variable
Conjunto de datos de entrenamiento	T
Paso del gradiente	α
Ancho de kernel (Cantidad de vecinos cercanos tomado)	σ
Valor positivo	η
Valor de regularización (Lambda)	λ
Número de características	dc
Número de ejemplos	N
Pesos de las características	$\vec{w}$
Probabilidad de selección de un ejemplo con un ejemplo distinto	p_{ij}
Error	$\in$
Derivada de la función de error	Δl
Contador	t

2.2.8 **Generación de datos**

Según Chang et al [34] un conjunto desbalanceado se caracteriza por tener un número de ejemplos mayor en algunas clases que en otras. En el desarrollo del proyecto, se balanceó la clase minoritaria (generación), con la clase mayoritaria (autoconsumo), para ello se usó la técnica de sobre muestreo de minorías sintéticas, conocido por sus siglas en inglés, Synthetic Minory Oversampling Technique (SMOTE), y enfoque de muestreo sintético adaptativo para conjuntos de datos desequilibrados, conocido por sus siglas en inglés, Adaptative Synthetic Sampling Approach for Imbalanced Datasets (ADASYN).

Según Haibo et al. [35]. Para calcular el grado de desbalance d entre un conjunto de datos se debe relacionar la cantidad de datos de la clase mayoritaria m_l , con la cantidad de datos de la clase minoritaria m_s, como se muestra en la ecuación (21).

$$d = \frac{m_s}{m_l} \tag{21}$$

En la ecuación (22) se muestra la cantidad de datos que se deben generar, se denominan con la letra G para suplir el desbalance, el porcentaje de balance β que se quiere, ese valor puede variar de 0 a 1. En el caso de la tesis, se seleccionó un β de 1, ya que se era necesario tener la clase de generación y autoconsumo completamente balanceados.

$$G = (m_l - m_s) * \beta \tag{22}$$

En la ecuación (23) se muestra la tasa de proyectos de clase mayoritaria cercanos a los de la clase minoritaria, con respecto a la cantidad de vecinos cercanos tomados en cuenta. Para un proyecto de clase minoritaria (generación), toma los vecinos cercanos pertenecientes a la clase mayoritaria (autoconsumo) Δ_i y divide ese valor por la cantidad K de vecinos cercanos tomados en cuenta. En la ecuación (24) se normaliza la tasa calculada anteriormente.

$$r_i = \frac{\Delta_i}{K} \tag{23}$$

$$\widehat{r_i} = \frac{r_i}{\sum_{i=1}^{m_s} r_i} \tag{24}$$

En la ecuación (25) se muestra la cantidad de datos que deben ser generados por cada dato minoritario.

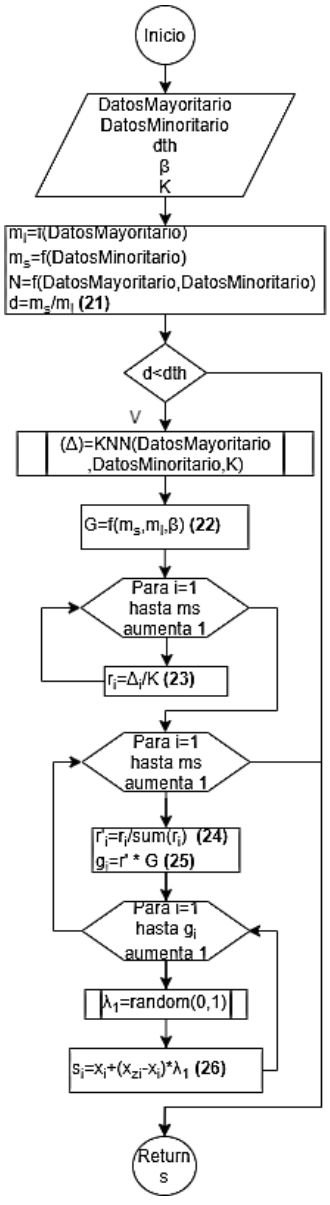
$$g_i = \widehat{r_i} * G \tag{25}$$

La ecuación (26) muestra la fórmula de generar un dato sintético s_i, la cual toma la distancia entre el proyecto x_i y el proyecto de clase minoritaria seleccionado aleatoriamente x_{zi}, a dicha distancia se le multiplica un valor aleatorio λ_1.

$$s_i = x_i + (x_{zi} - x_i) * \lambda_1 \quad (26)$$

Mediante la Figura 10 se muestra una representación del algoritmo de ADASYN, y en la Tabla 4 se muestra los nombres de las variables utilizadas. El algoritmo genera datos sinteticos de una clase minoritaria, a partir de la distancia de dos ejemplos de la misma clase.

Figura 10 Diagrama de ADASYN.



Fuente: Haibo et al. [35]

Tabla 4 Seudónimos usados en la generación de datos

Variable	Simbolo de la variable
Valor máximo de desbalance	dth
Porcentaje de balance	β
Número de vecinos cercanos tomados	K
Número de ejemplos de la clase mayoritaria	m_l
Número de ejemplos de la clase minoritaria	m_s
Número de ejemplos	N
Grado de desbalance	d
Distancia euclidea entre los ejemplos	Δ
Número de ejemplos a generar	G
Tasa de ejemplos de clase mayoritaria cercanos ejemeplos de clase minoritaria	r
Número de ejemplos a generar por cada muestra de clase minoritaria	g
Número aleatorio	λ_1
Ejemplo de referencia	x_i
Ejemplo aleatorio	x_{zi}
Ejemplo sintetico	s_i

2.2.9 **Aprendizaje de maquina**

Según James W. et al [36] el aprendizaje de maquina se puede interpretar como una manera de extraer conocimiento a partir de la información contenida por un conjunto de datos y transmitirla a un modelo. Como se muestra en la Tabla 5, existen 3 tipos de aprendizaje de máquina, el supervisado, no supervisado y el reforzado, para el desarrollo de esta tesis, se usa el aprendizaje supervisado, ya que una de las necesidades es que el modelo aprenda a partir de los ejemplos.

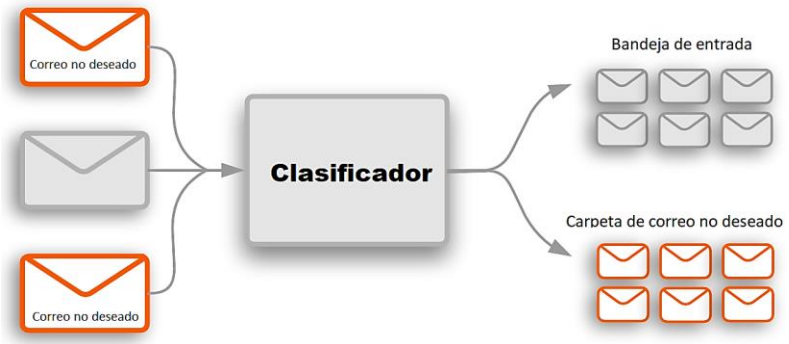
Tabla 5 Tipos de aprendizaje

Tipo de aprendizaje	**Descripción**	**Aplicaciones**
Aprendizaje supervisado	Necesita datos para poder entrenar, los cuales deben estar etiquetados con las respuestas deseadas.	Regresión y clasificación
Aprendizaje no supervisado	Al igual que el anterior caso, se necesitan datos, en este caso, no están etiquetados.	Agrupamiento
Aprendizaje reforzado	Es un tipo de aprendizaje que va cambiando según las recompensas y castigos que tenga a medida que corre el tiempo.	Clasificación y regresión

2.2.10 **Algoritmos de aprendizaje supervisado**

Es un tipo de aprendizaje que necesita ejemplos y respuestas para aprender, funciona tanto para clasificación como para regresión. El ejemplo más común de aprendizaje supervisado es la clasificación de correos electrónicos no deseados (ver Figura 11), donde entra a un algoritmo un conjunto de correos electrónicos ya etiquetados si son deseados o no deseados, el algoritmo al ya estar entrenado está con la capacidad de clasificar los nuevos correos electrónicos.

Figura 11 Ejemplo aprendizaje supervisado.



Fuente: Kumar K. [37]

En la Tabla 6 se muestran los algoritmos que se usaron durante el desarrollo del proyecto, cada uno tomó parte importante, desde la generación de datos sintéticos, hasta el objetivo final del proyecto.

Tabla 6 Algoritmos usados.

Algoritmo	**Aplicación**
K vecinos cercanos	Clasificación
Árboles de decisión	Clasificación
Redes neuronales	Predicción numérica/Clasificación

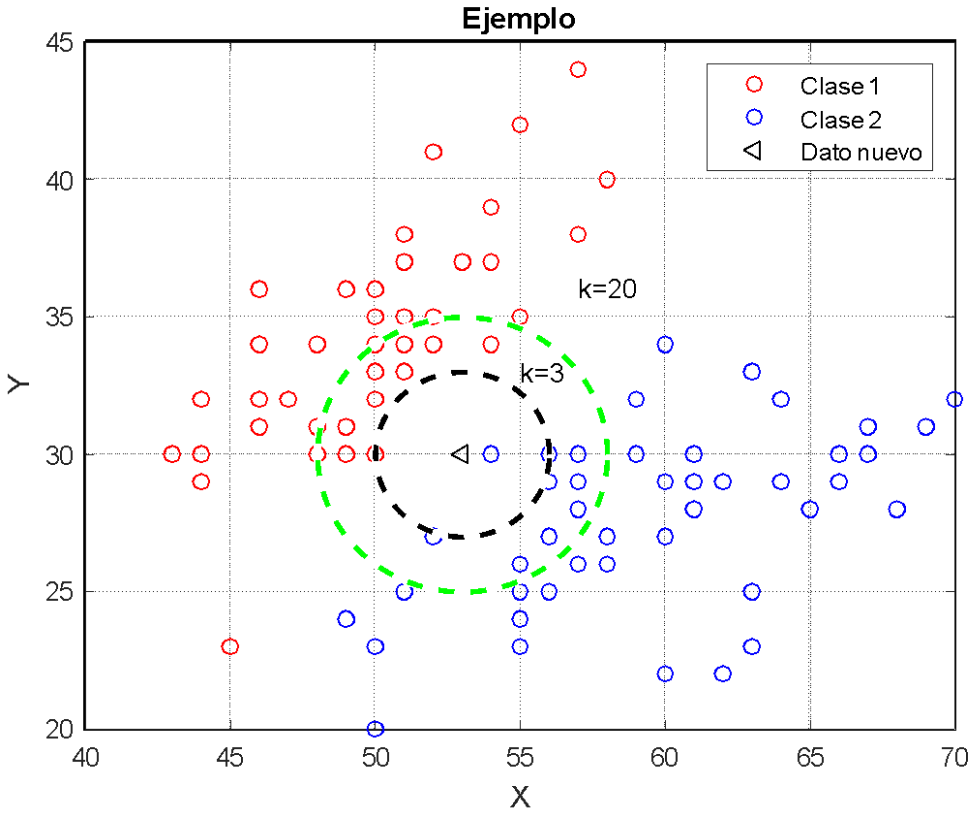
Fuente: Autores

K vecinos cercanos: Es un algoritmo de clasificación, que permite encontrar el punto con mayor probabilidad de pertenecer al conjunto de ejemplos más cercanos [18].En el algoritmo se define que los k números cercanos al punto, son llamados "vecindad" (ver

Figura 12). Se selecciona que el nuevo punto es perteneciente a la clase donde más tiene vecinos, esto puede variar según el k que se utilice. Normalmente se mide la distancia entre el punto nuevo al resto de los puntos, mediante la distancia euclídea (ver ecuación (27)) siendo la longitud de la diferencia de las coordenadas de 2 ejemplos.

$$d_{euclidea} = \sqrt{(x_i - x_j)^T \cdot (x_i - x_j)} \tag{27}$$

Figura 12 K vecinos cercanos.



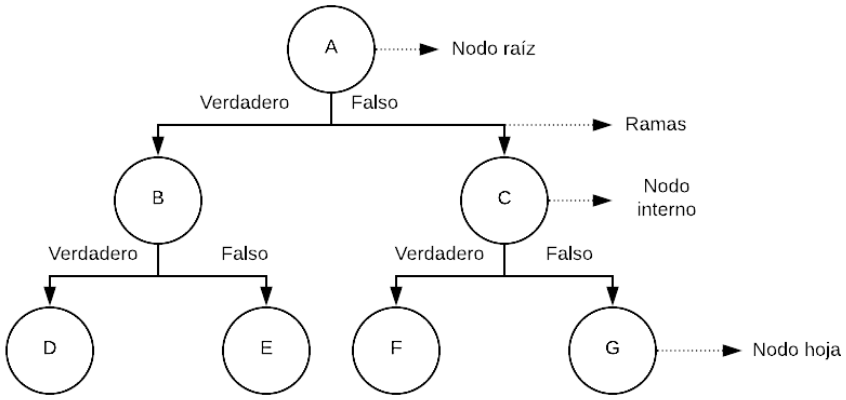
Fuente: Autores con base de datos de MATLAB

La cantidad de vecinos cercanos tomados en cuenta, puede hacer que se tengan errores, por ejemplo en la

Figura 12 se agrega un dato, no se sabe qué tipo de dato es, entonces si se toman los tres vecinos más cercanos, se puede decir que el dato que se agrega, es de la clase 2.Mientras que si se toman los veinte vecinos más cercanos, tendría nueve vecinos más cercanos de la clase 2 y once vecinos más cercanos de la clase 1, convirtiendo el dato agregado en clase 1.

Árboles de decisión: Se definen como modelos que tienen una forma de árbol, los cuales tienen como el fin clasificar un conjunto de datos [38]. El árbol de decisiones tiene la estructura de un diagrama de flujo, donde los nodos internos son una variable, cada rama es una salida y cada nodo final (denominado nodo hoja) es una clase, en Figura 13 se puede observar la estructura ya mencionada. La parte superior del árbol de decisiones, es un nodo raíz [28].

Figura 13 Componentes de un árbol de decisión



Fuente: Autores

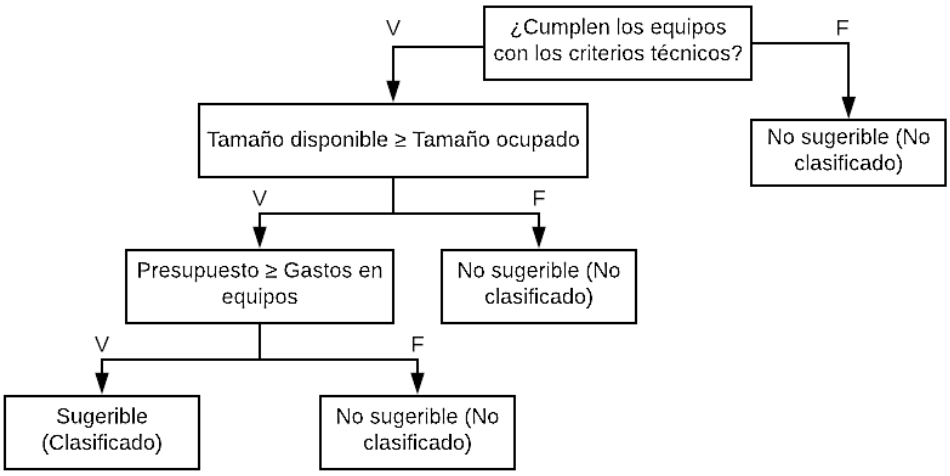
Árbol de decisión utilizado: Para que la herramienta clasificara los paneles solares que se visualizan en la Figura 14, en sugeribles y no sugeribles, se hizo un árbol de decisión que tenía tres criterios principales:

1. Cumplimiento de requerimientos técnicos, donde los equipos que cumplan dichos requisitos pasan al siguiente nodo interno del árbol de decisión, en caso contrario pasan a los nodos hoja.

2. Cumplimiento de los espacios, se toman los equipos que cumplan con el espacio disponible pasan al siguiente nodo interno, en caso contrario pasan a los nodos hoja.

3. Cumplimiento del presupuesto, se toman los resultados del presupuesto y es comparado con la cantidad de dinero disponible, los equipos que cumplan este requisito son guardados.

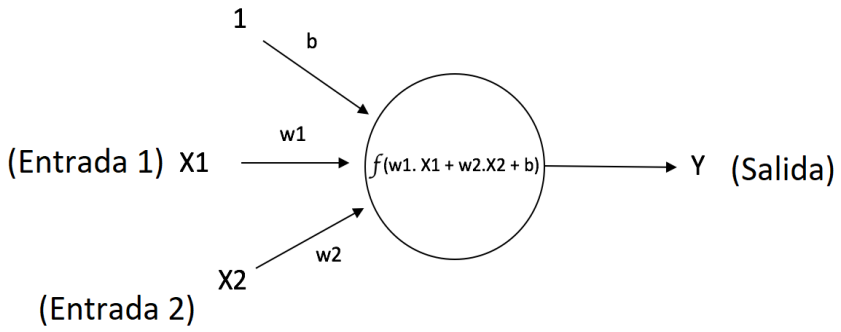
Figura 14 Árbol de decisión usado.



Fuente: Autores

Redes neuronales: Es una simulación computacional de un conjunto de neuronas biológicas, que permite relacionar las señales de entrada con las señales de salida [39]. Cada neurona toma su conjunto de datos de entrada y les asigna un valor de relevancia en Figura 15 se observa la representación de una neurona o también llamada perceptrón, donde se muestran 2 entradas de valores y de 1 constante nombrada como Bias (b).

Figura 15 Representación de una neurona artificial.



Fuente: Ujwlkarn [40]

Los perceptrones tienen pasos internos, cada uno de ellos tiene el fin de manejar los valores de entrada a la neurona. Se detallan sobre los pasos internos de las neuronas en Tabla 7.

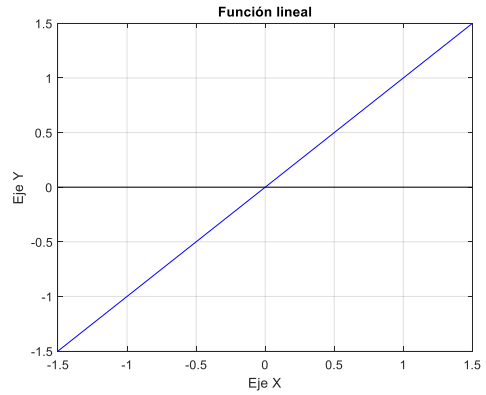
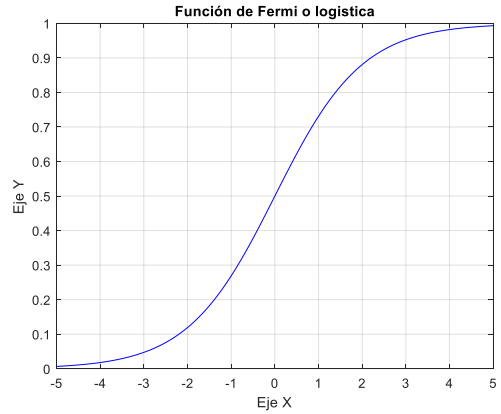
Tabla 7 Detalles funcionamiento del perceptrón.

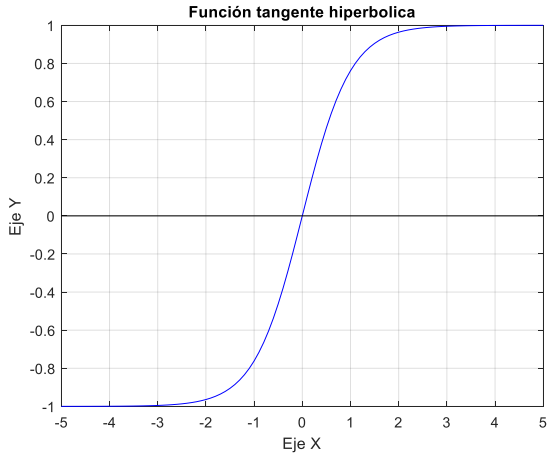
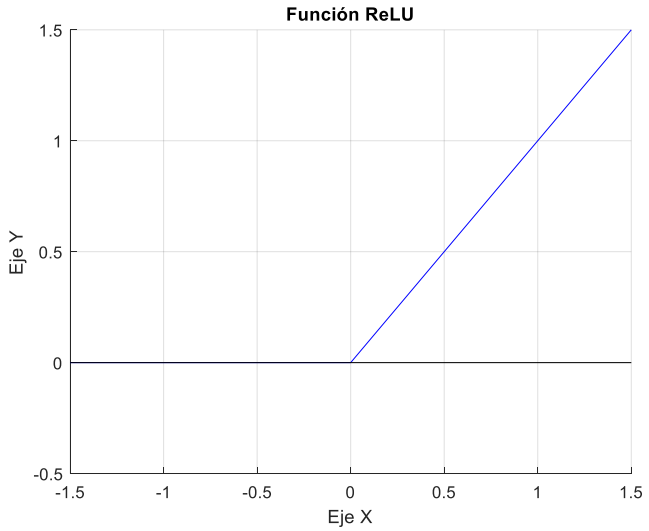
Nombre de paso	**Descripción**
Función de propagación	Esta función se encarga de cambiar los datos de entrada, según los pesos que se tienen para esos datos. Es una función lineal que multiplica los valores de los datos de entrada (x_i) por los pesos (w_i) que se tienen para cada uno de ellos y se le suma el valor de Bias, como se muestra en la ecuación (28). Se le conoce como red de entrada (Network input). $$net = \sum (x_i * w_i) + \mathrm{b} \quad (28)$$
Función de activación	Es el grado de excitación de una neurona, es la relación entre la entrada y la salida, también es conocida como función de transferencia. El nuevo estado de activación ($a_j(t)$) de la neurona (j) está en función de la red de entrada ($net_j(t)$), en función del estado de activación anterior ($a_j(t-1)$) y el valor límite de la función de activación (θ_j). $$a_j(t) = f_{act}(net_j(t), a_j(t-1), \theta_j) \quad (29)$$

Fuente: David Kriesel, et al [41]

Tipos de funciones de activación: Debido a que existen distintas funciones de activación, donde muchas de ellas son variaciones de otras, como se puede inferir a partir del artículo escrito por Bircanoglu et al [42]. Un factor influyente en la cantidad de las funciones de activación tomadas en cuenta en el desarrollo del proyecto, es la limitada capacidad computacional. Se optó por tomar una cantidad reducida y representativa de funciones de activación. En la Tabla 8 se muestran las funciones de activación evaluadas en el desarrollo de la tesis, donde en conjunto con la cantidad de neuronas pueden cambiar el desempeño de la red neuronal.

Tabla 8 Funciones de activación

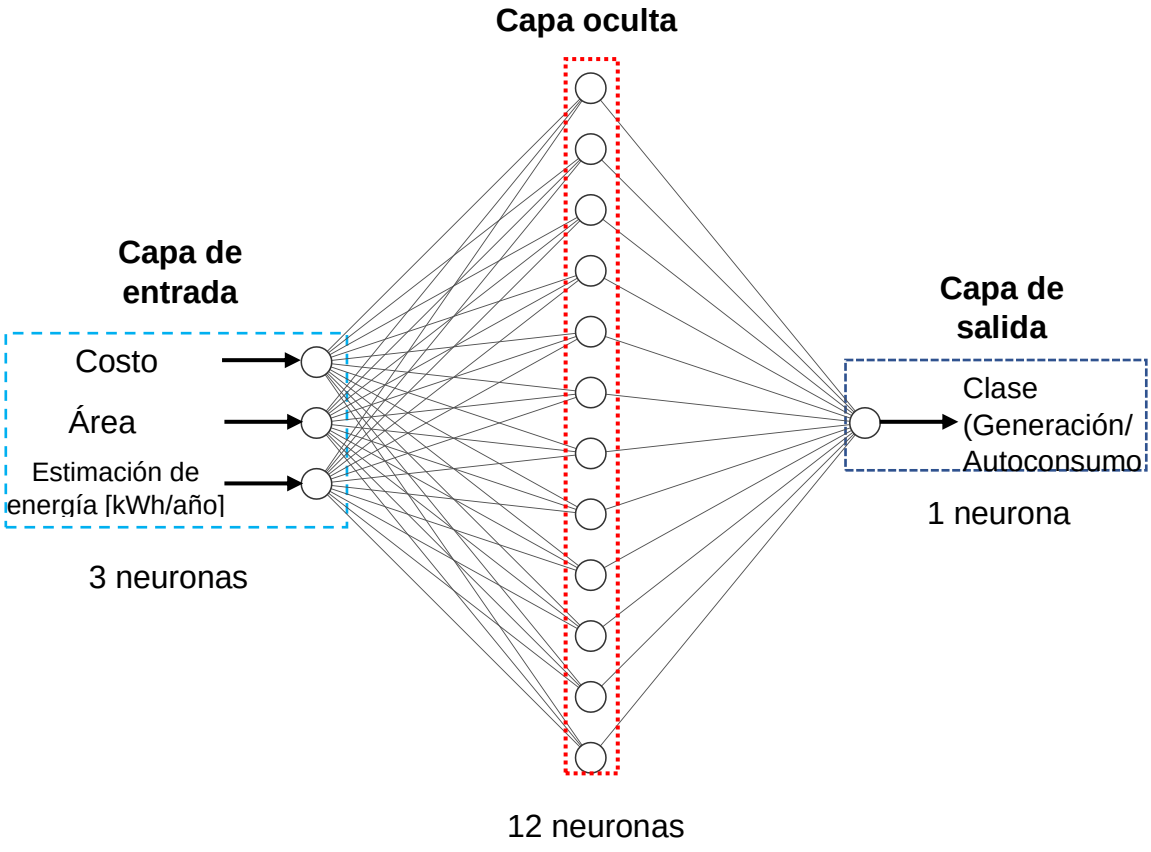
Función de activación	Forma	Propiedades
Función lineal	Figura 16 Función lineal.  Fuente: Autores	• Tiene una salida proporcional a la entrada. • Permite múltiples salidas • No es eficaz con la propagación hacia atrás. • No permite apilar capas de neuronas con esta función de activación. • La función que usa es: $f_{act}(net) = net$ (30)
Función de Fermi o logística	Figura 17 Función logística, de Fermi o sigmoide.  Fuente: Autores	• No lineal. • Permite apilar capas. • Buena para clasificar. • Cuando los valores de la red de entrada (net) son muy grandes o muy pequeños, no pueden hacer cambios en la red neuronal. • Costosa computacionalmente • La función que usa es: $f_{act}(net) = \frac{1}{1 + e^{-net}}$ (31)

Función de activación	Forma	Propiedades
Función tangente hiperbólica	Figura 18 Función tangente hiperbólica.  Fuente: Autores	• No lineal • Permite apilar capas. • Buena para clasificar. • Gradiente más pronunciado que la función logística. • Tiene el mismo problema que la función logística o de Fermi, con los valores grandes y pequeños. • Costosa computacionalmente. • La función que usa es: $$f_{act}(net) = \frac{2}{1 + e^{-2*net}} - 1 \quad (32)$$
Función unidad línea rectificada (ReLU)	Figura 19 Función ReLU.  Fuente: Autores	• No lineal. • Puede apilar capas. • No es costosa computacionalmente. • Puede hacer que una parte de la red neuronal esté inactiva. • La función que usa es: $$f_{act}(net) = \begin{cases} 0\ para\ net < 0 \\ net\ para\ net \geq 0 \end{cases} \quad (33)$$

Red neuronal multicapa

Es el tipo de red neuronal más frecuente, no tiene retroalimentación de las salidas de las neuronas y la mayoría de las veces se alimenta hacia adelante [43]. Consta de una capa de entrada y una de salida, la cantidad de capas ocultas pueden variar según la complejidad del problema a resolver. En la Figura 20 se observa la arquitectura de la red neuronal usada en la red neuronal de clasificación, cada una de las neuronas de entrada se enlaza con las neuronas de la capa oculta, cada una de ellas le asigna un peso a los valores que reciben por parte de ellas. Tiene tres entradas (Costo [$], área [$m^2$] y estimación de energía $\left[\frac{\text{kWh}}{año}\right]$) a continuación tiene una capa oculta, y finaliza con una neurona que da como resultado el tipo de proyecto usado.

Figura 20 Red neuronal multicapa.

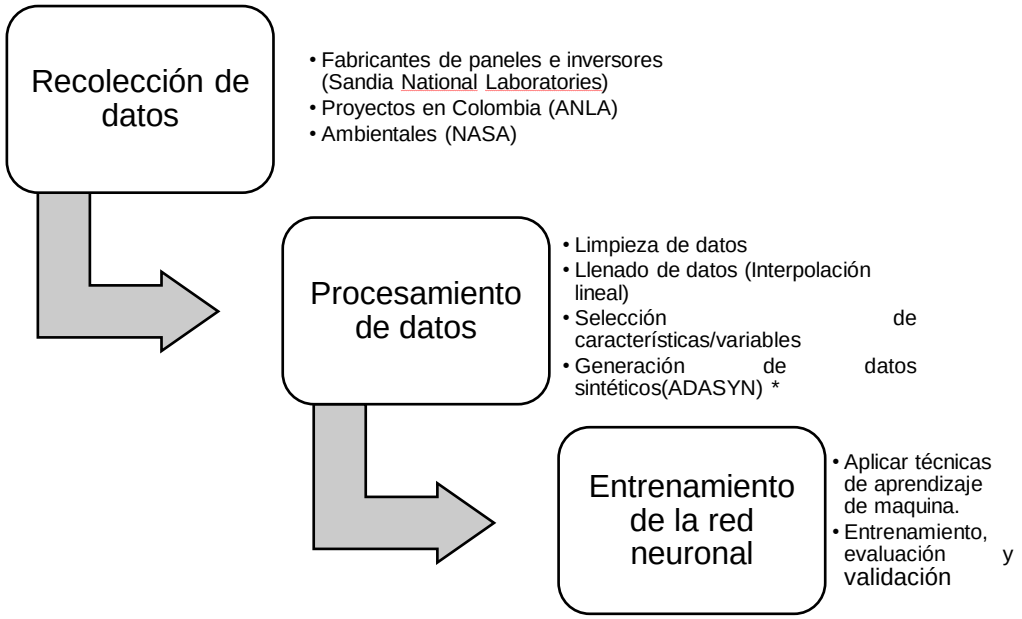


Fuente: Autores

3. Metodología.

En la Figura 21 se muestra la metodología usada en el desarrollo del proyecto.

Figura 21. Metodología



Fuente: Autores

4. Recolección de datos

En esta fase se recopilan los datos de distintas fuentes, provenientes de tres orígenes distintos, la primera es la Autoridad Nacional de Licencias Ambientales (ANLA), la segunda fuente es la herramienta de PVLib, desarrollada por los laboratorios de Sandia National Laboratories (SNL), siendo una herramienta libre código abierto y que permite simular el desempeño de sistemas de energía fotovoltaica. La tercera fuente es la base de datos de la NASA, se tomó los datos ambientales relacionados con la velocidad del viento, la temperatura del sitio y la irradiancia. La Tabla 9 describe las fuentes usadas y lo que se hizo con sus datos.

Tabla 9 Fuentes de datos, tipos de datos y el uso dado.

Fuente	Datos	Uso
ANLA	Proyectos fotovoltaicos conectados a la red. (Ver Anexo A)	Entrenar una red neuronal de clasificación, con el fin de clasificar los nuevos proyectos como generación o autoconsumo. Servir como modelo para una herramienta secundaria que genera datos a partir de ellos.
PVLib / SNL	Fichas técnicas de inversores y paneles fotovoltaicos.	Ser objeto del dimensionamiento.
NASA	Datos del ambiente, como temperatura, velocidad del viento o radiación solar.	Para hacer una simulación del sistema en el punto donde esté el proyecto. Se obtuvo una descripción estadística (Media, mediana, moda, máximo, mínimo) de todos los puntos en Colombia.

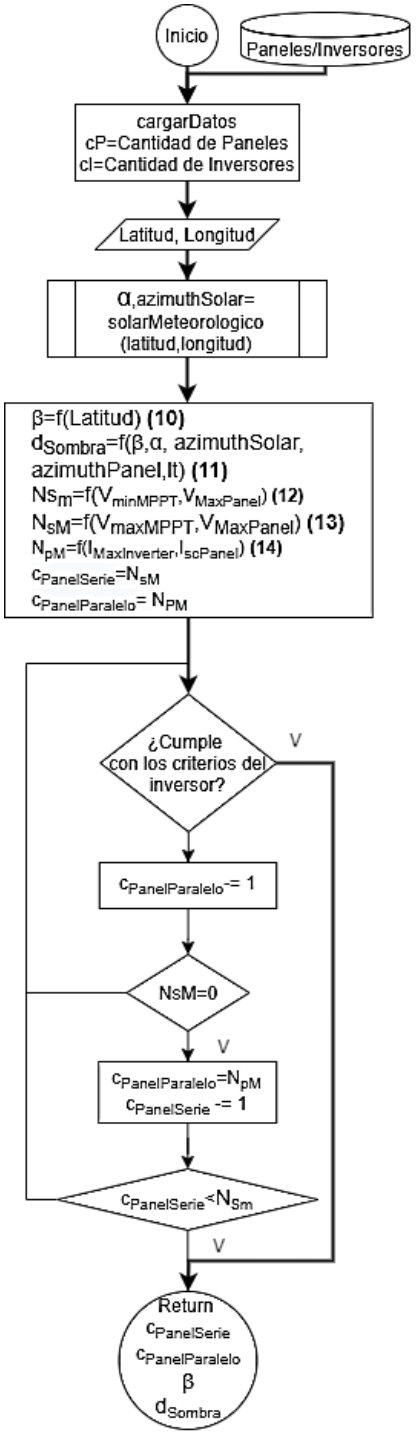
4.1 Desarrollo de la herramienta auxiliar

Durante el levantamiento de datos, los detalles sobre las plantas solares fotovoltaicas (ángulo del panel, número de paneles en serie y en paralelo, ect.) fueron escasos, se optó por desarrollar una herramienta que realiza el dimensionamiento y toma en cuenta los límites del inversor, mostrados a continuación:

1. Potencia máxima del inversor >= Potencia de los paneles.
2. Voltaje máximo del inversor >= Voltaje de corto circuito de los paneles.
3. Potencia de autoconsumo del inversor <= Potencia de los paneles.
4. Voltaje máximo por MPPT <= Voltaje máxima de los paneles en serie.
5. Voltaje mínimo por MPPT >= Voltaje máxima de los paneles en serie.
6. Corriente máxima permitida por el inversor >= La corriente máxima de los paneles en paralelo.

La Figura 22 muestra el algoritmo de la herramienta auxiliar y en la Tabla 10 se muestra los símbolos usados. La herramienta estimó 11594 ejemplos, de los cuales 5794.

Figura 22 Diagrama de flujo de la herramienta auxiliar.



Fuente: Autores

Tabla 10 Seudónimos usados en la herramienta auxiliar

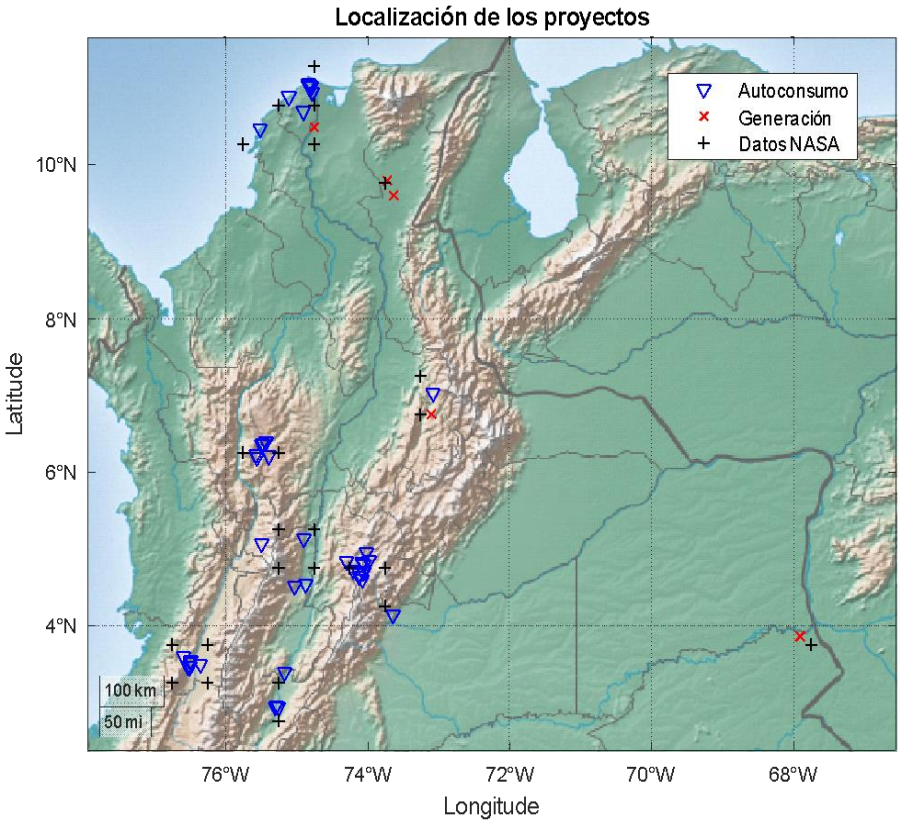
Variable	Símbolo de variable
Cantidad de Paneles	cP
Cantidad de Inversores	cI
Altitud solar	α
Ángulo del panel	β
Distancia de la sombra del panel	d_{Sombra}
Número mínimo de paneles en serie	N_{sm}
Número máximo de paneles en serie	N_{sM}
Número máximo de paneles en paralelo	N_{pM}
Cantidad de paneles en serie	$C_{PanelSerie}$
Cantidad de paneles en paralelo	$C_{PanelParalelo}$

Fuente: Autores

4.2 Tratamiento de datos

El tratamiento y la visualización de los datos, son puntos importantes ya que estos nos permiten conocer más sobre estos, dándonos información detallada, por ejemplo, posibles datos erróneos o si se puede discriminar entre las dos clases que hay (Autoconsumo o generación). En la Figura 23 se muestra la ubicación de los proyectos fotovoltaicos en el territorio nacional, se observa una concentración de proyectos en la parte centro del país, en su mayoría de autoconsumo, existe una mayor concentración plantas de generación en el norte del país.

Figura 23 Localización de los proyectos que fueron registrados.



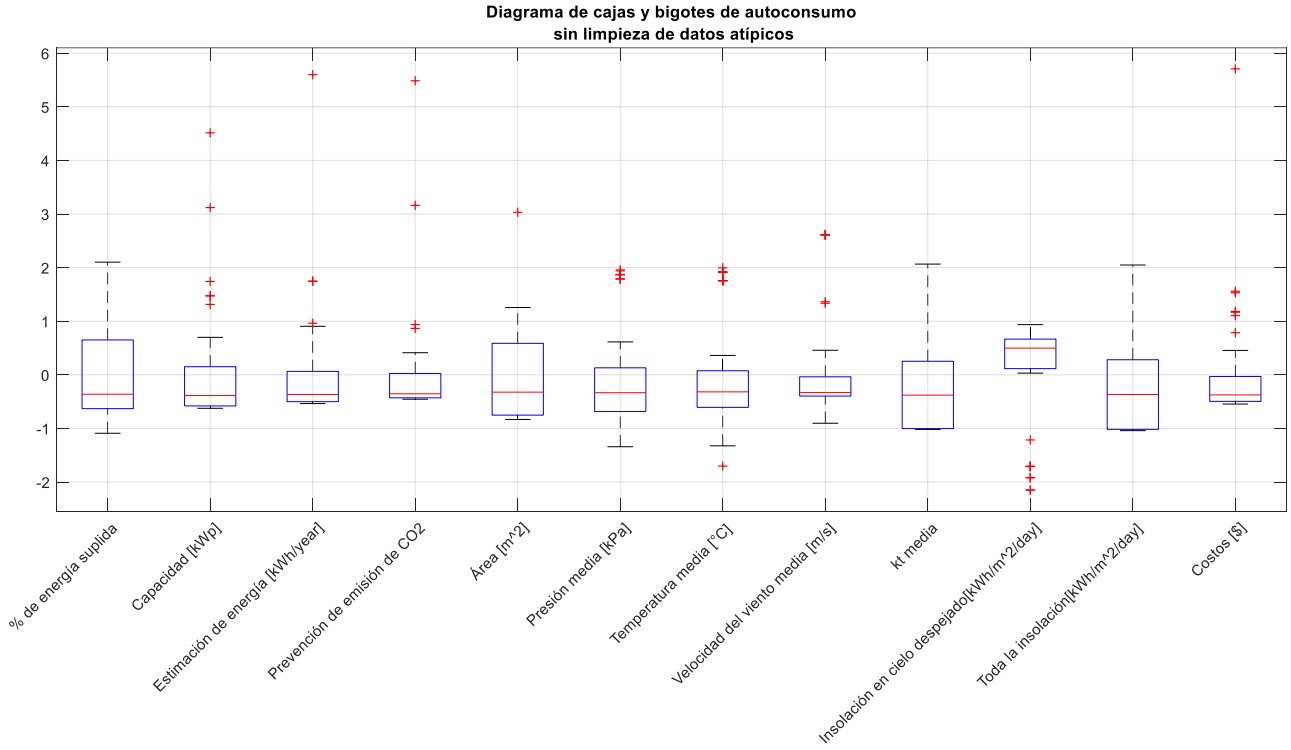
Fuente: Autores

4.2.1 Visualización de datos

Una vez visualizado las ubicaciones de los proyectos, se toman sus datos y se hace un diagrama de cajas y bigotes, para poder obtener información adicional sobre los datos, como la concentración o los datos atípicos. Todos los datos son normalizados, ya que existe una diferencia entre las magnitudes de los datos, hace que no se pueda apreciar correctamente la gráfica. Se puede ver en la Figura 24 una cantidad de datos atípicos muy notoria, en Tabla 11 se cuenta la cantidad de

datos atípicos en cada variable para este caso. En las variables como energía suplida, se observa que no hay ningún dato atípico, debido a que los proyectos de autoconsumo siempre están en el mismo rango (de 10% a 100%). Lo datos atípicos más notorios se encuentran en la variable de estimación de energía y en los costos, estos valores altos son explicados porque es posible que dichos datos tengan errores o sean pertenecientes a otro conjunto. También es posible que debido a la cantidad de datos que se tiene, no abarque el panorama completo de los proyectos de generación.

Figura 24 Diagrama de cajas y bigotes autoconsumo sin limpieza de datos.



Fuente: Autores

Tabla 11 Cantidad de atípicos en autoconsumo

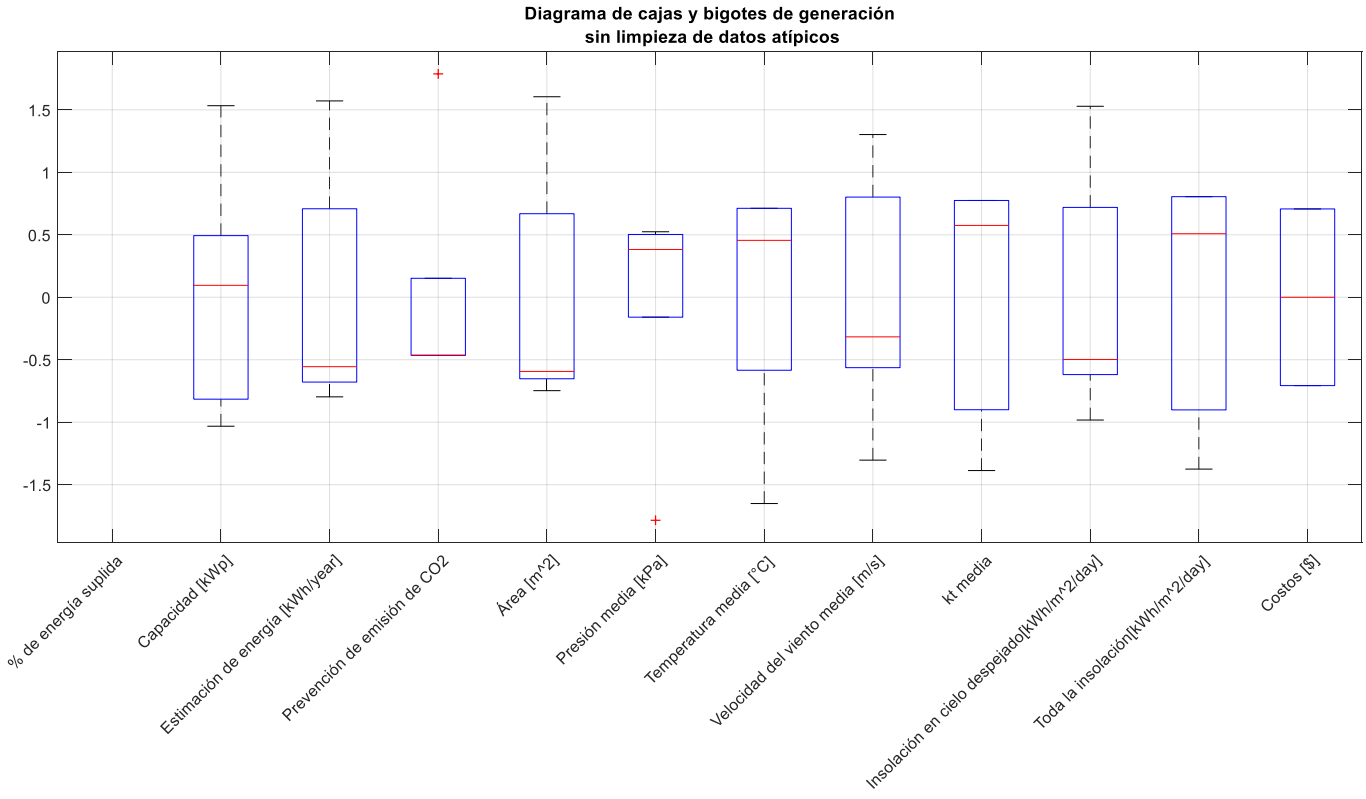
Cantidad de atípicos	Variable
12	Prevención emisión atmosférica [kgCO2/año]
11	Temperatura media [°C]
11	Insolación en el cielo despejado [kWh/m^2/día]
10	Estimación de energía [kWh/año]
10	Presión media [kPa]
10	Costo [$]

Cantidad de atípicos	Variable
7	Capacidad instalada [kWp]
7	Velocidad del viento [m/s]
1	Área [m^2]
0	% Energía suplida
0	Índice de claridad (kt) media
0	Toda la insolación [kWh/m^2/día]

Fuente: Autores

Para los datos de generación se hace un diagrama de cajas y bigotes (ver Figura 25), debido a la poca cantidad de datos que se tienen, es menos probable encontrar un dato atípico. Los dos únicos datos atípicos representarían plantas fotovoltaicas de generación de energía con características diferentes al grupo, en un caso, previniendo una mayor cantidad de emisiones de CO_2 y en el otro caso, estando un lugar con más elevado con respecto a las demás plantas.

Figura 25 Diagrama de cajas y bigotes de generación.

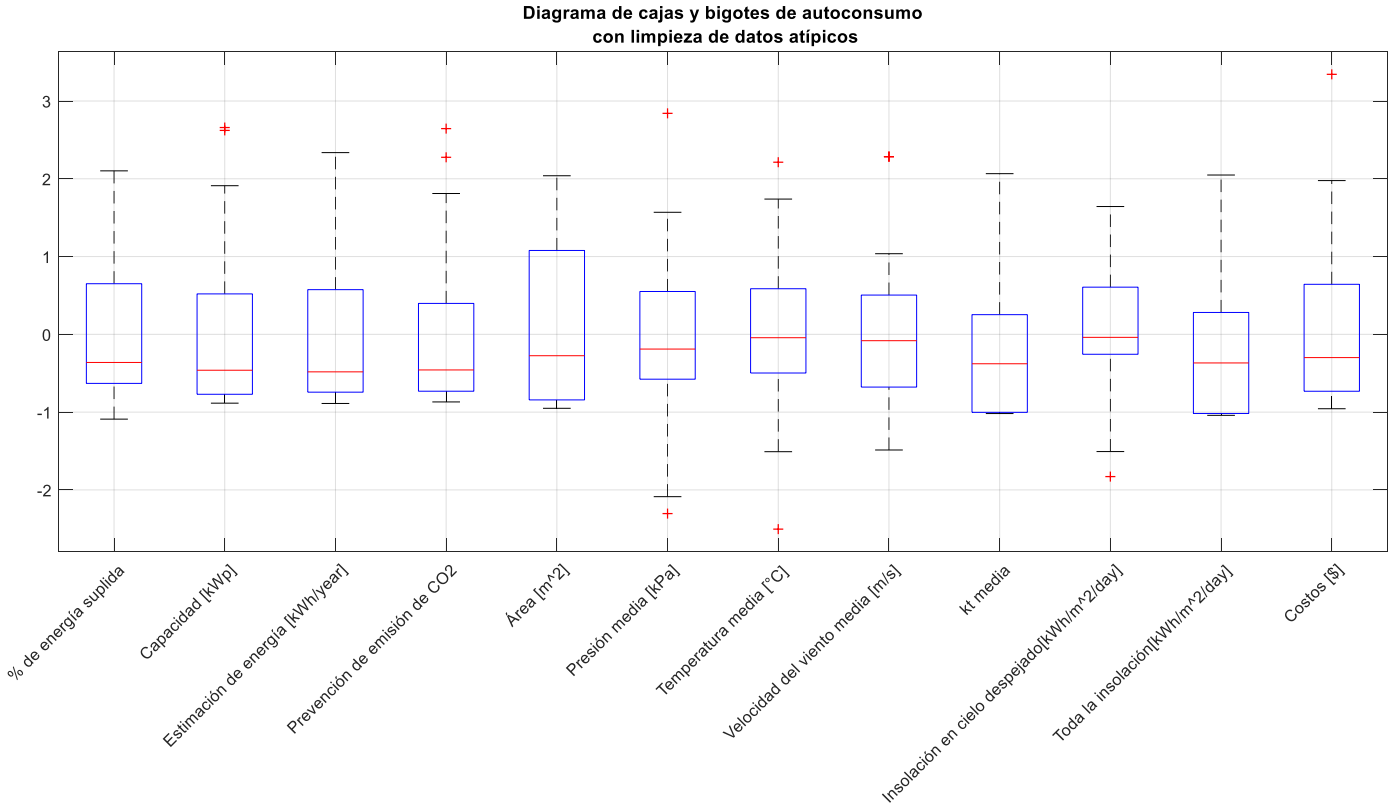


Fuente: Autores

4.2.2**Limpieza de datos**

Se hace un tratamiento de datos para mejorar la calidad de los datos y poder tener un mejor desempeño a la hora de hacer el modelo. Se hizo una limpieza de datos, excluyendo los datos atípicos. Se hizo únicamente para los datos de autoconsumo ya que la cantidad de datos de Generación son muy pocos. La problemática del desbalance de datos se resuelve en la sección Generación de datos sintéticos . Se hace un diagrama de cajas y bigotes mostrado en la Figura 26 para observar los resultados, y se puede analizar que los datos atípicos residuales, son producto de un nuevo cálculo para hacer el diagrama de cajas y bigotes.

Figura 26 Diagrama de cajas y bigotes, con limpieza de datos.



Fuente: Autores

4.2.3**Imputación de datos utilizando interpolación lineal**

Las variables donde existan puntos que no fueron llenados. Esto se debe a que, en los informes de la ANLA, algunas veces se especificaba cierto tipo de datos, pero algunas otras veces esos datos no eran informados. En la Tabla 12 se observan la cantidad de datos faltantes por característica.

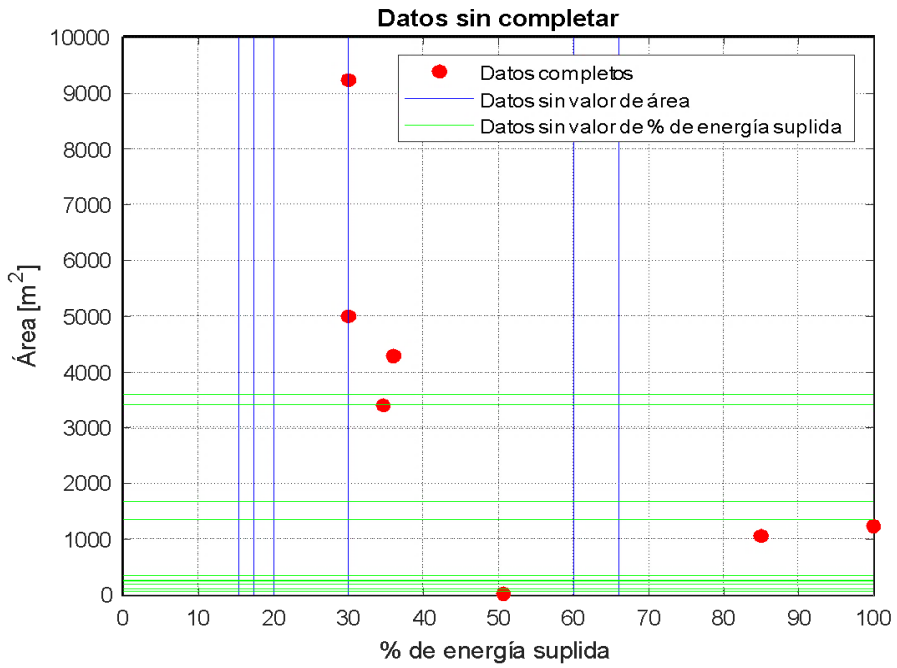
Tabla 12 Espacios faltantes para cada variable.

Cantidad de valores faltantes	Característica o variable
37	% Energía suplida
31	Área [m^2]
2	Estimación de energía [kWh/año]
2	Prevención emisión atmosférica [kgCO2/año]

Fuente: Autores

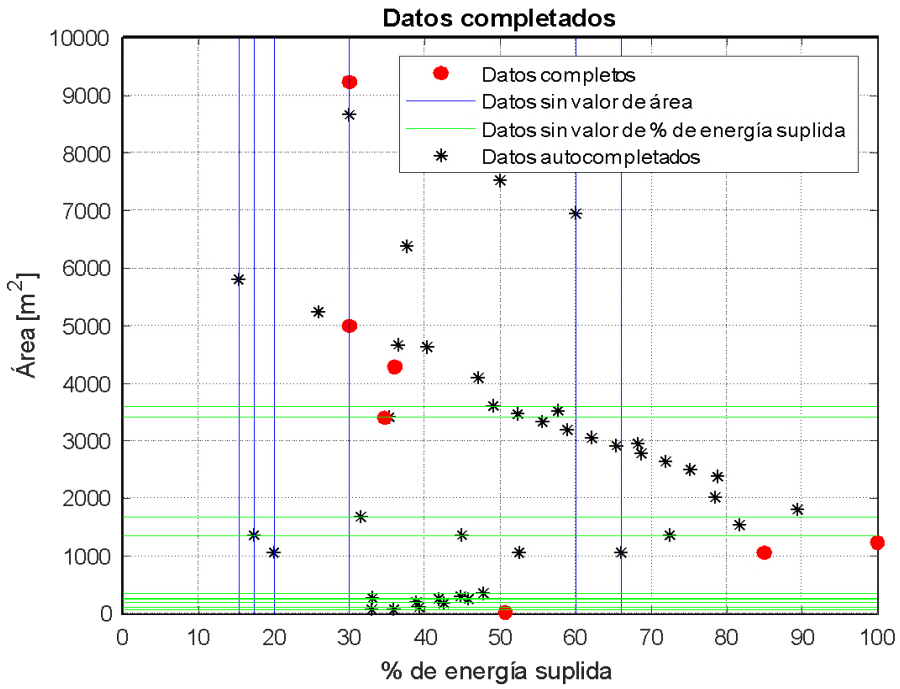
Para la imputación de estos datos, se hizo una interpolación tomando los datos más cercanos para encontrar el número que se relaciona con la pareja del valor faltante. Según Noor et al [44], al comparar la interpolación lineal con otros modelos, se obtiene un buen ajuste de los datos. En la Figura 27 se muestra los datos que están completos y las variables que tienen datos incompletos. Al realizar interpolación lineal se completan los datos, como se observa en la Figura 28,mostrando un comportamiento lineal que no se sale de los límites establecidos.

Figura 27 Grafica de dispersión con datos faltantes



Fuente: Autores

Figura 28 Grafica de dispersión con los datos completados

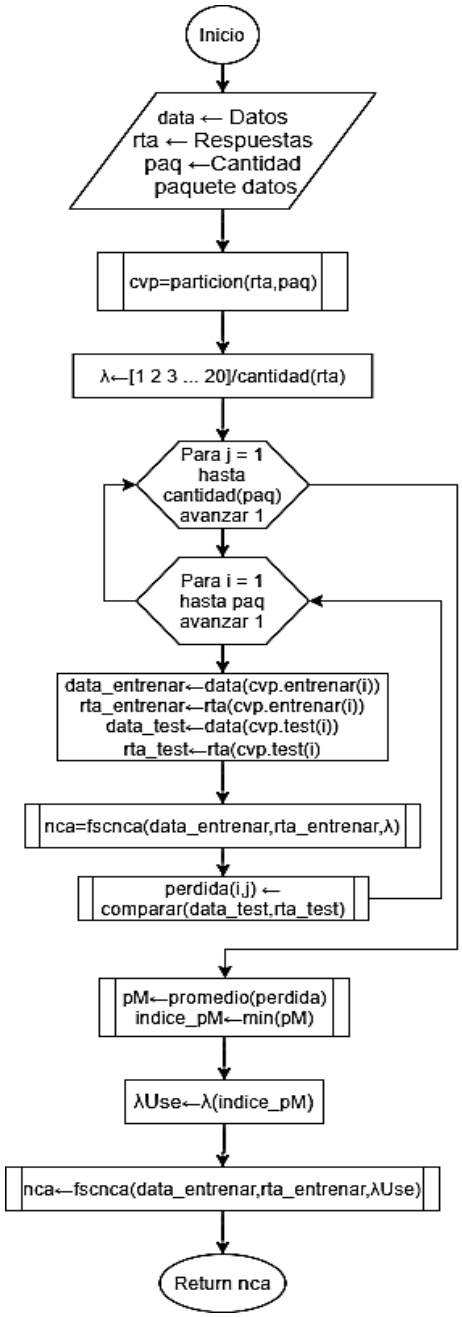


Fuente: Autores

5. Selección de características

La selección de características se hace para encontrar las variables que mejor describan el comportamiento que se quiere estudiar. En este caso se quiere saber cuáles variables pueden discriminar mejor los proyectos y cuales variables son las mejores para describir el dimensionamiento de los proyectos fotovoltaicos. En el diagrama de flujo (ver Figura 29) se observa la serie de pasos que se tomaron para encontrar el mejor valor de regularización que se necesita en la selección de características. Se dividen en los partes 10 partes iguales, y se hace validación cruzada para cada uno de los Lamba, seleccionando el Lambda que menor error produzca.

Figura 29 Diagrama de flujo selección valor de regularización (Lambda)



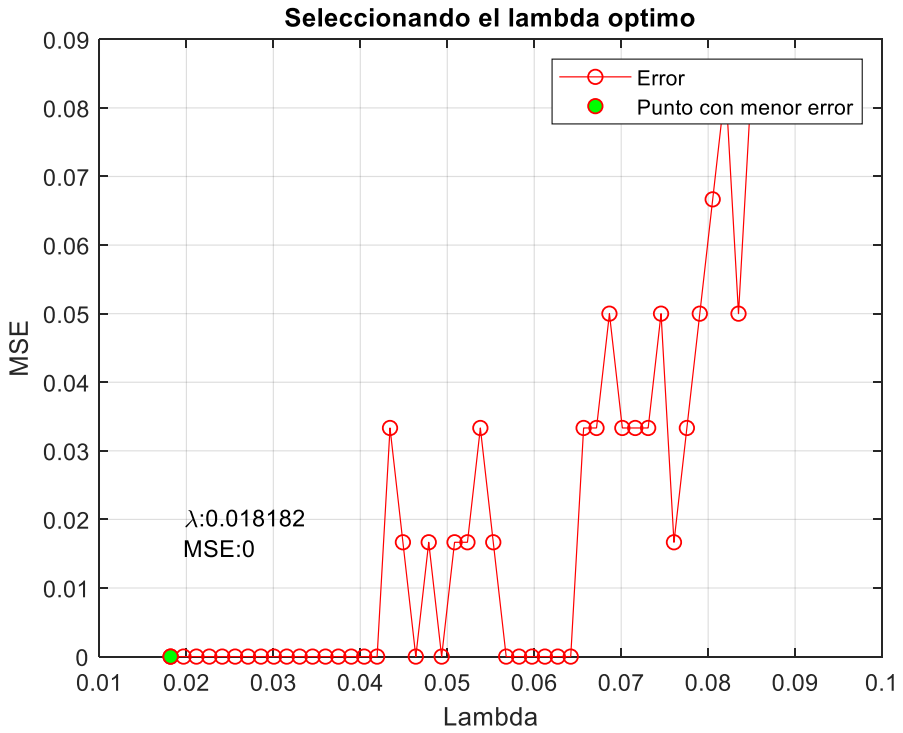
Fuente: Autores

5.1.1 **Selección de características para clasificación**

La selección de variables o también llamado selección de características, se hace con el fin de mejorar el rendimiento de la red neuronal, debido a que se le incluyen

únicamente las variables más importantes. Se usó Selección de características para datos de alta dimensión (Neighborhood Component Feature Selection) para poder seleccionar dichas variables relevantes. Para el proceso de selección de características, se necesita cambiar el parámetro de regulación lambda (λ), en la Figura 30 se muestra el MSE entre los puntos que predice el algoritmo y los puntos reales, se selecciona la lambda de menor MSE.

Figura 30 Valor del parámetro de regularización vs MSE. Para clasificación.



Fuente: Autores

Al hacer una selección de características de la base de datos, se obtiene Tabla 13 donde se observa los pesos de cada una de variables. Las variables con mayor peso, están relacionadas con dar un aporte más significativo a la clasificación de las clases de las variables (Generación o autoconsumo).

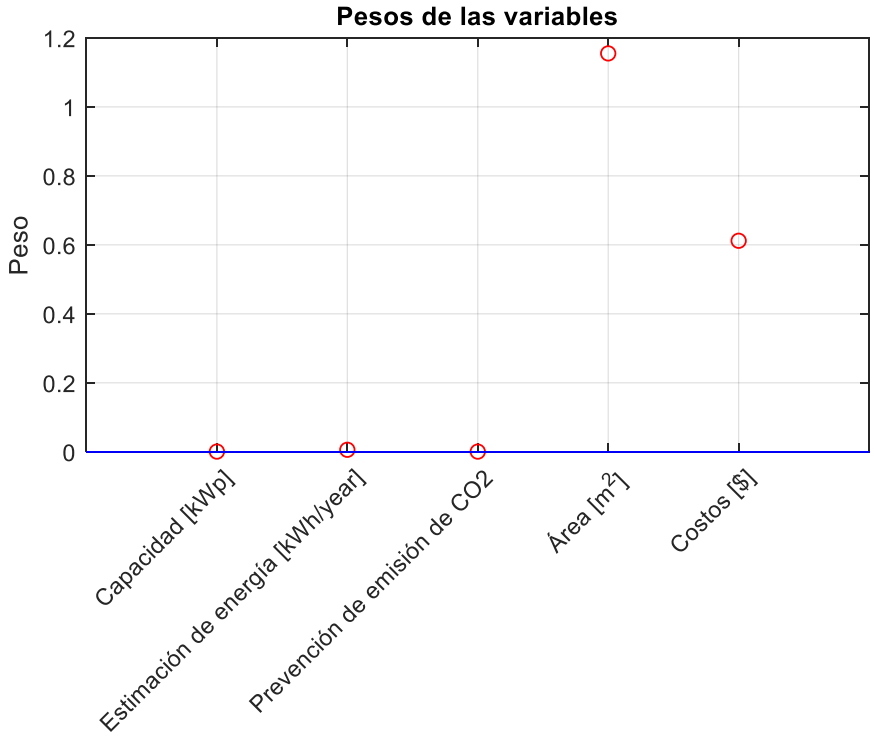
Tabla 13 Pesos de las características o variables usando FSCNCA

Variable	Peso
Área [m^2]	1.155
Costos [$]	0.61182
Estimación de energía [kWh/año]	0.0058211
Capacidad [kWp]	0.00058162
Prevención de emisión de CO2	0.00051239

Fuente: Autores

La Figura 31 muestra el peso de cada una de las variables, se observa que la variable más influyente es el área, seguida por el costo, la línea azul representa la última variable que es tomada en cuenta, en este caso es la estimación de energía [kWh/año]

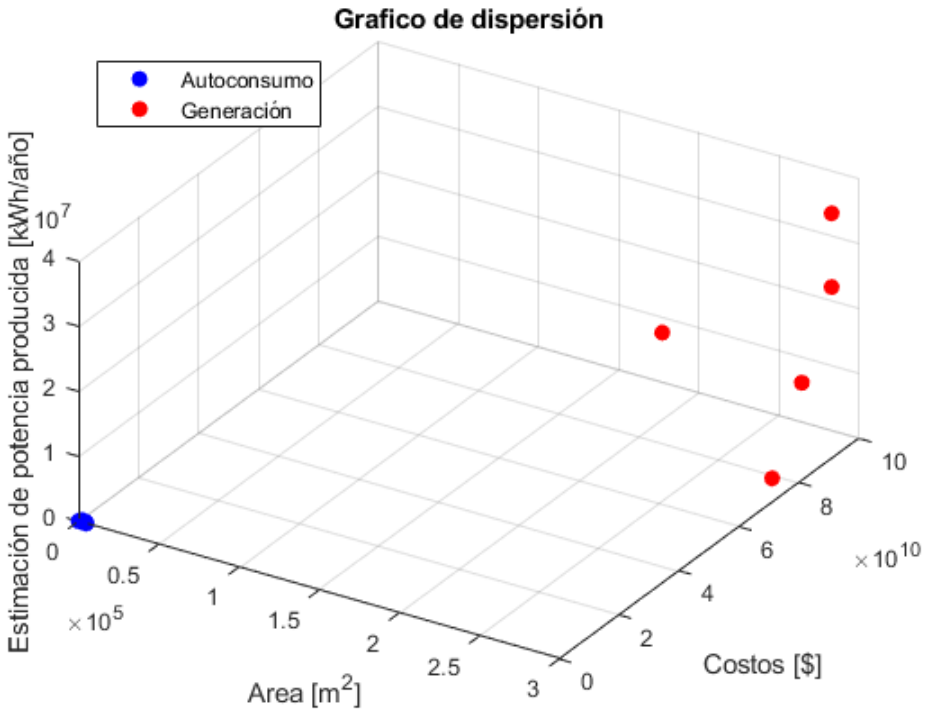
Figura 31 Peso de las variables para la clasificación.



Fuente: Autores

Al hacer el diagrama de dispersión de las tres variables como se muestra en la Figura 32, se observa que es sencillo distinguir entre las clases, debido a la escasa cantidad de datos, no se pueda conocer que tan cerca pueden estar las dos clases evaluadas.

Figura 32 Grafica de dispersión de las 3 variables más relevantes

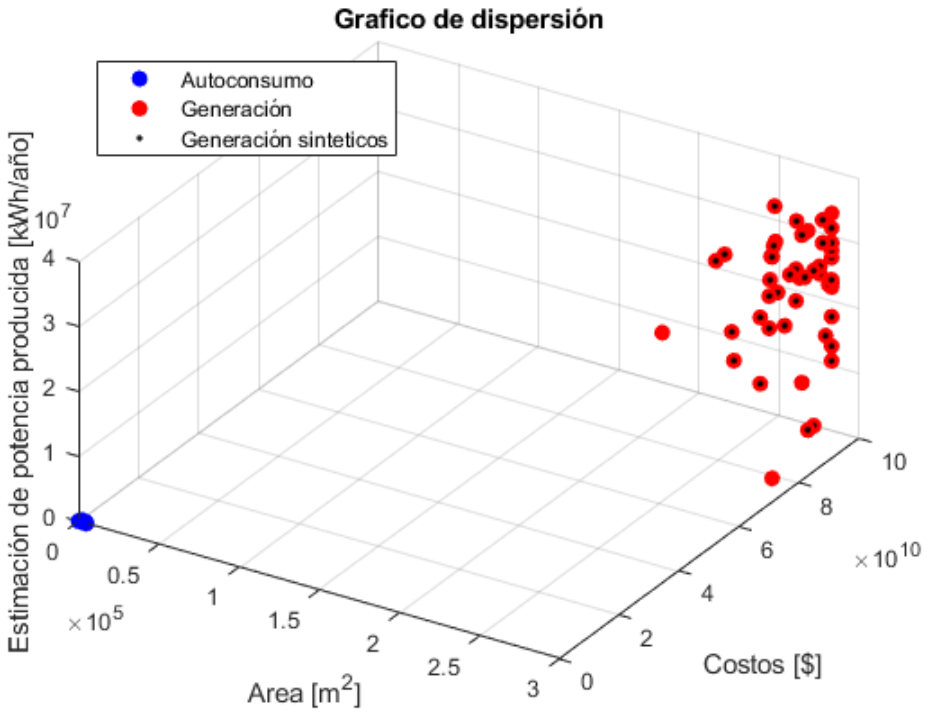


Fuente: Autores

5.1.2 **Generación de datos sintéticos**

Para la correcta clasificación de los datos, se necesita que las dos clases tengan la misma cantidad de datos, de lo contrario se puede obtener un resultado de clasificación sesgado, dándole mayor importancia a los datos de autoconsumo. Se conoce que la base de datos tiene mayor cantidad de datos referentes al autoconsumo, abarcando cerca del 90% de los datos en general, siendo así, la base de datos tiene un desbalance en sus clases. Por ello, se generan datos sintéticos, mediante la técnica de ADASYN (Adaptative Synthetic Sampling Approach for Imbalanced Learning), es una técnica basada en SMOTE (Synthetic Minory Oversampling Technique). Generando aproximadamente cuarenta y cinco datos sintéticos, se hace un balance entre las clases de autoconsumo y generación, expandiendo la base de datos a un total de cien. En la Figura 33 se observan los datos generados por medio de la técnica, los datos están en el mismo rango en el que estaban los cinco iniciales.

Figura 33 Diagrama de dispersión con datos sintéticos

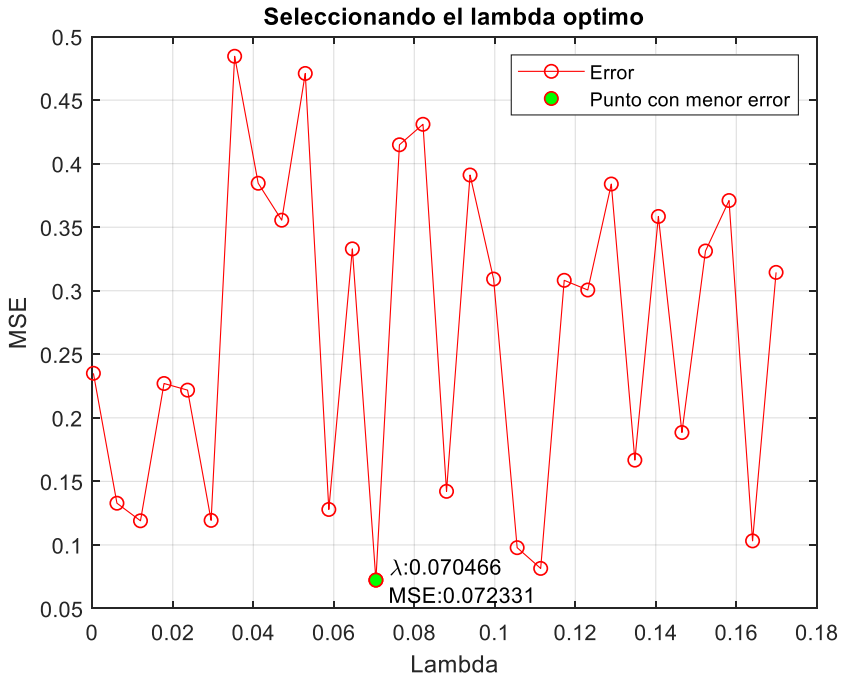
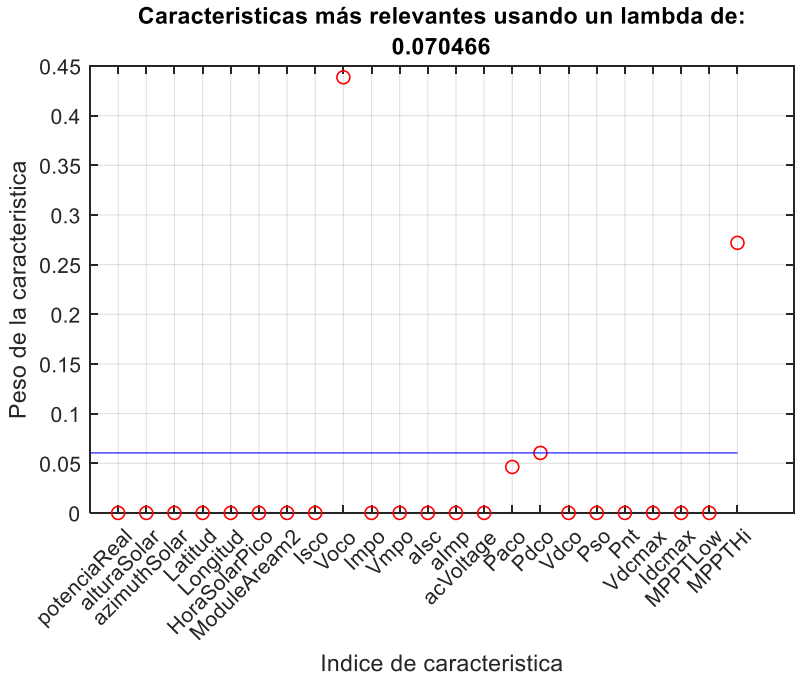


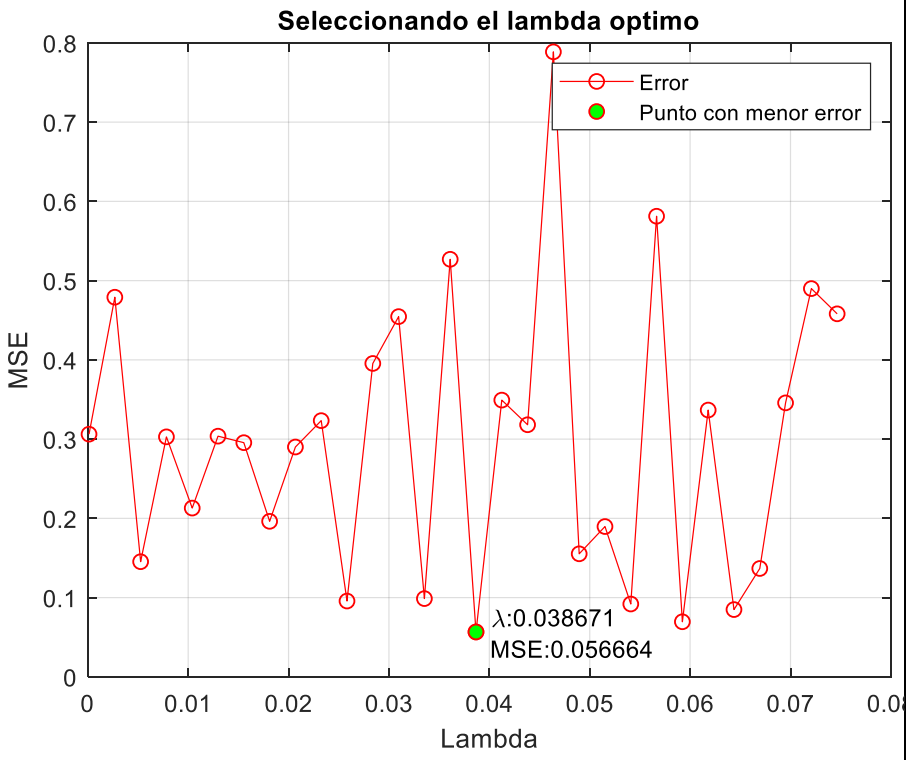
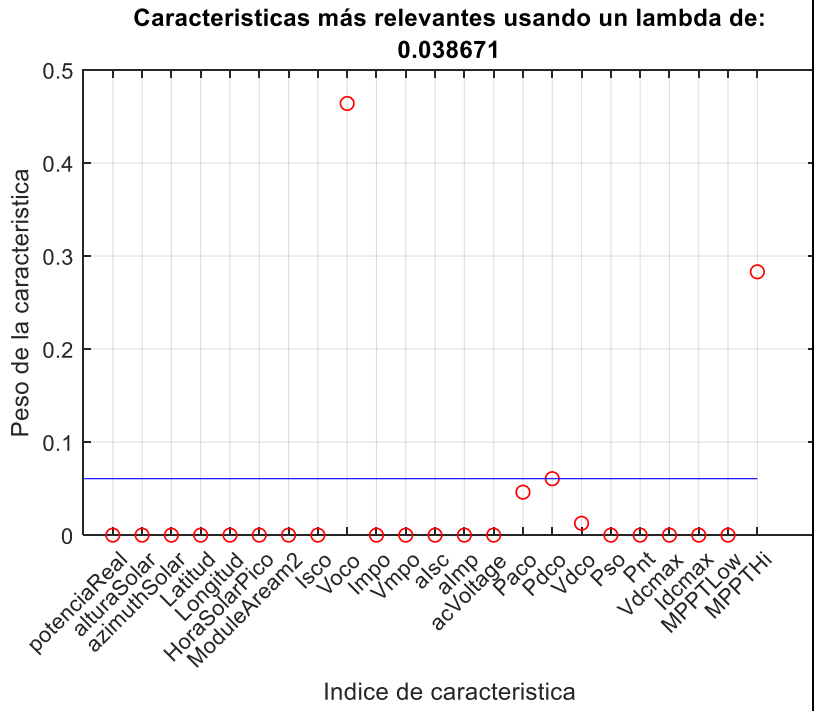
Fuente: Autores

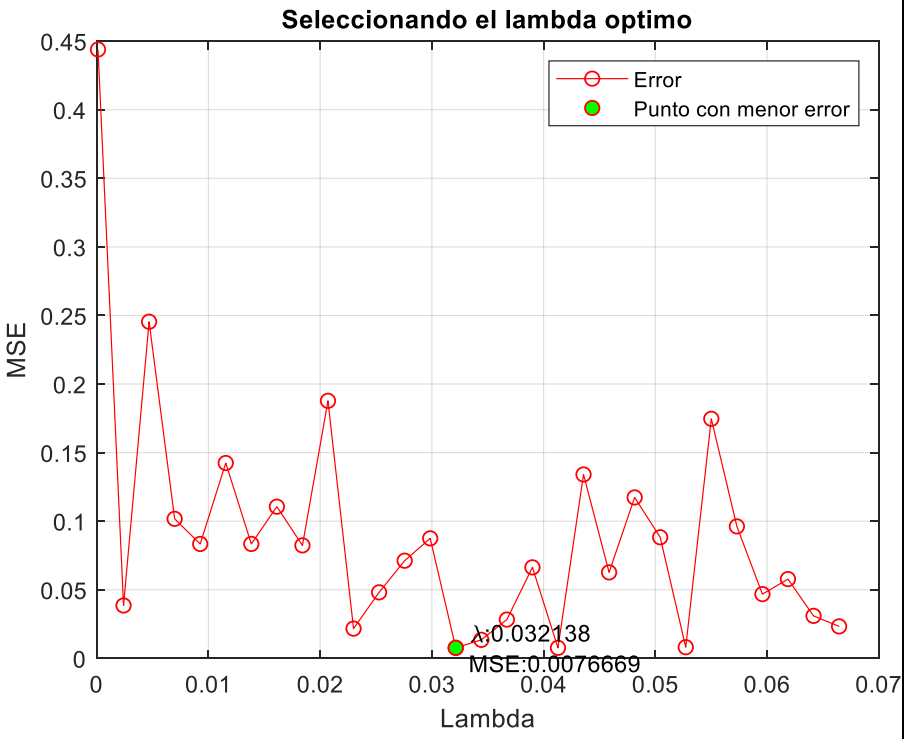
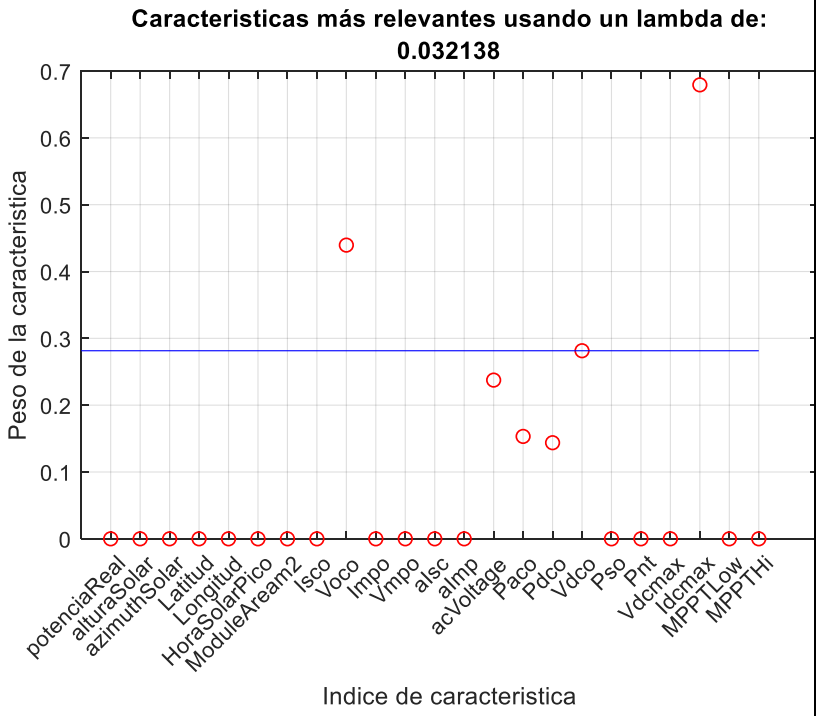
5.1.3 **Selección de características para la regresión**

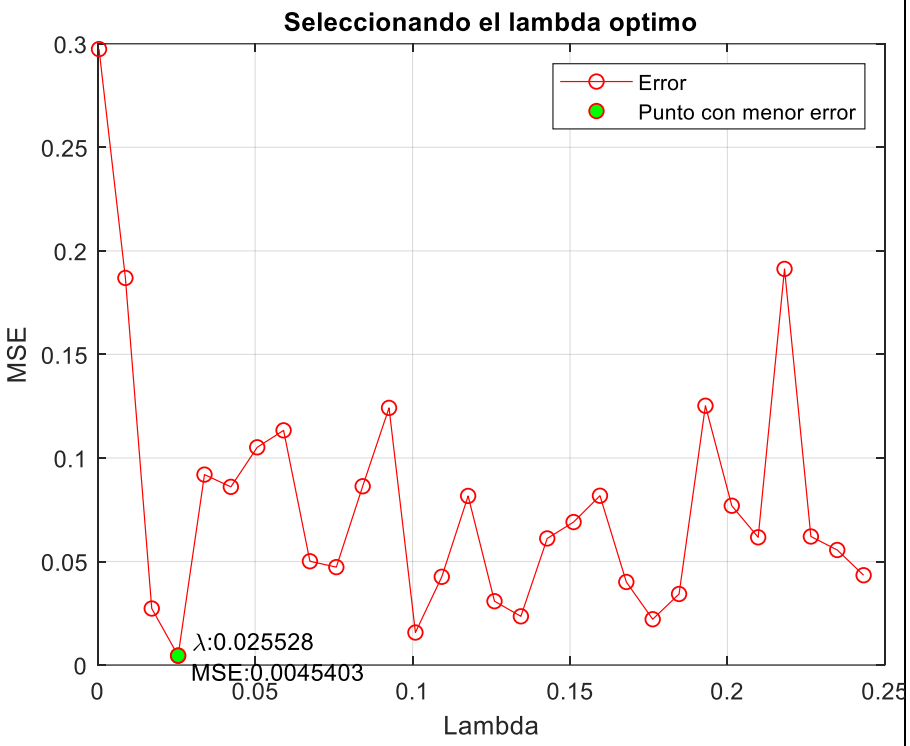
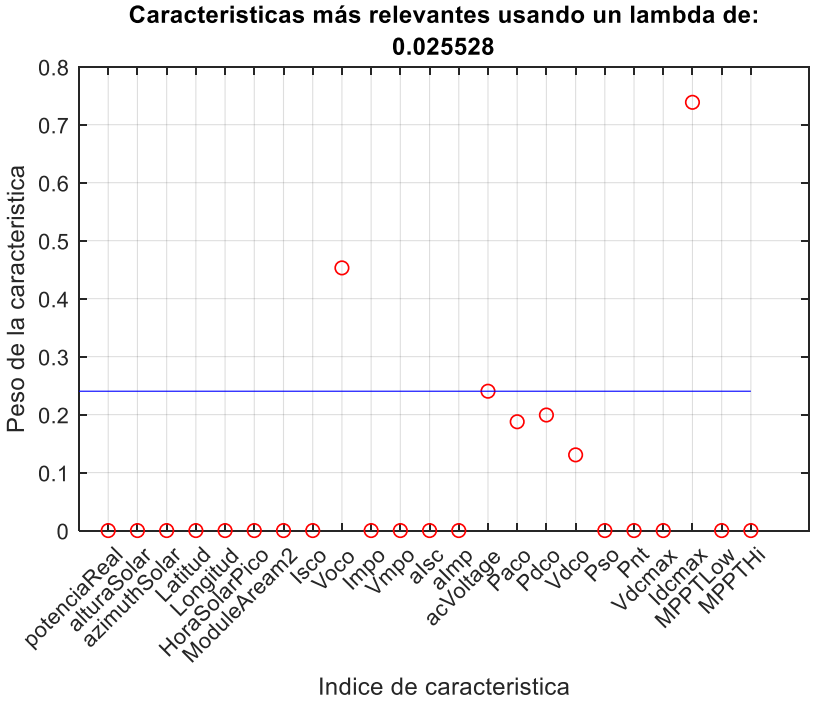
Se deben seleccionar las variables que mejor describan el comportamiento de una respuesta, se hace mediante la función Feature Selection Regression Neighboorhood Component Analysis por sus siglas en inglés (FSRNCA), para cada grupo de datos, se dividió en 2 conjuntos de datos, uno de entrenamiento y uno de validación, se hizo validación cruzada para poder determinar el valor de regularización (λ). En la Tabla 14 se muestran las gráficas donde se observa el error cuadradito medio (MSE) contra el valor del Lamba, o valor de regulación.

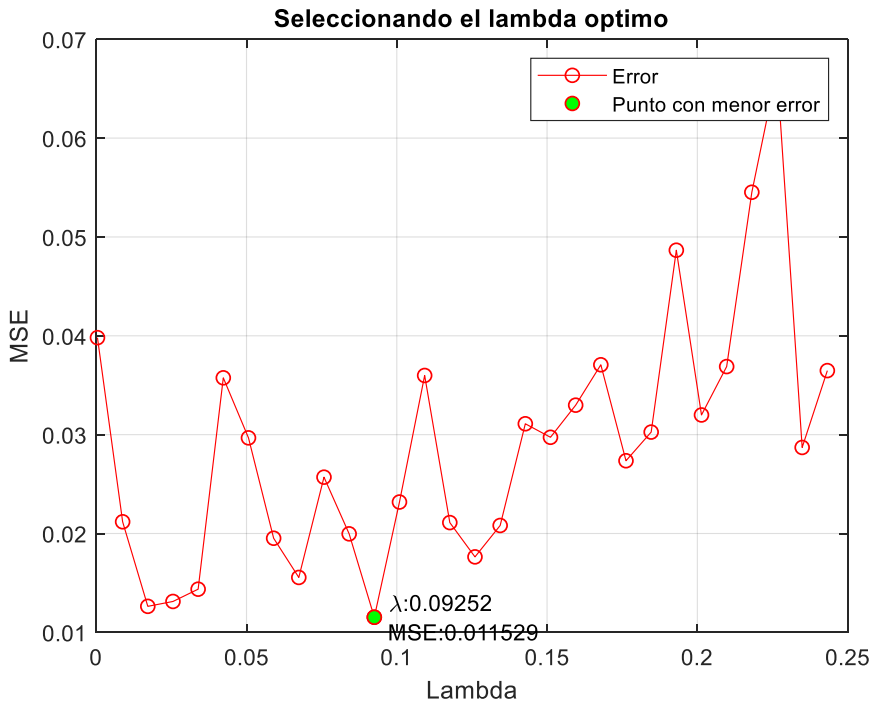
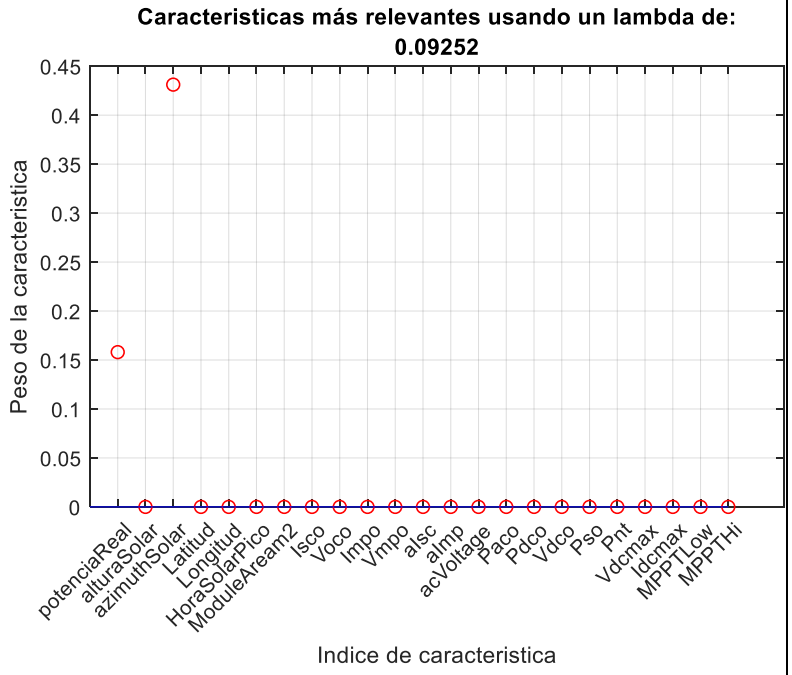
Tabla 14 Selección de características para las respuestas

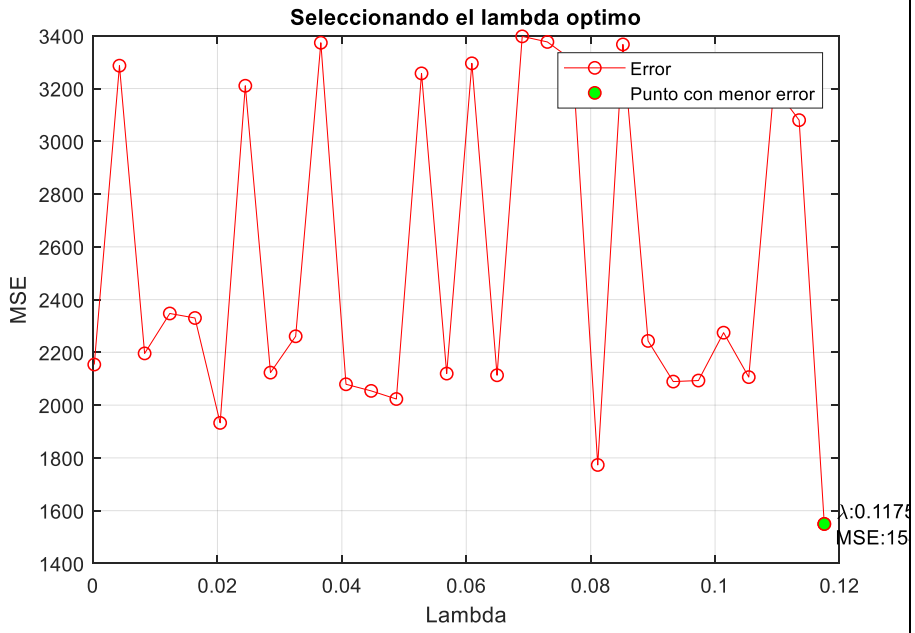
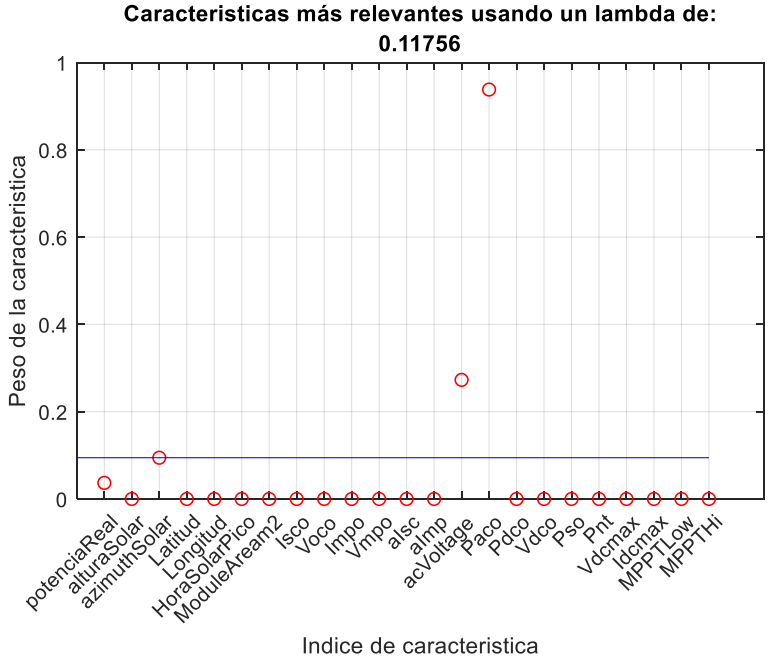
Variable	Grafica para seleccionar el parámetro de regulación Lambda VS MSE	Características seleccionadas
Cantidad de paneles en serie		 Características seleccionadas: • Voltaje de circuito abierto del panel (V_{oc}). • Voltaje máximo por MPPT ($MPPT_{Hi}$). • Potencia en DC en el que alcanza la potencia nominal (P_{dc}).

Variable	Grafica para seleccionar el parámetro de regulación Lambda VS MSE	Características seleccionadas
Cantidad máxima de paneles en serie		 Características seleccionadas: • Voltaje de circuito abierto del panel (V_{oc}). • Voltaje máximo por MPPT ($MPPT_{Hi}$). • Potencia en DC en el que alcanza la potencia nominal (P_{dc}).

Variable	Grafica para seleccionar el parámetro de regulación Lambda VS MSE	Características seleccionadas
Cantidad de paneles en paralelo		 Características seleccionadas: • Máxima DC permitida en el inversor ($I_{dc\ Máx}$). • Voltaje de circuito abierto del panel (V_{oc}). • Voltaje de DC al que alcanza la potencia nominal (V_{dc}).

Variable	Grafica para seleccionar el parámetro de regulación Lambda VS MSE	Características seleccionadas
Cantidad máxima de paneles en paralelo		 Características seleccionadas: 1. Máxima DC permitida en el inversor ($I_{dc\ Máx}$). 2. Voltaje de circuito abierto del panel (V_{oc}). 3. Voltaje AC nominal para la salida del inversor (V_{ac})

Variable	Grafica para seleccionar el parámetro de regulación Lambda VS MSE	Características seleccionadas
Ángulo del panel		 Características seleccionadas: 1. Azimut Solar 2. Potencia real 3. Latitud

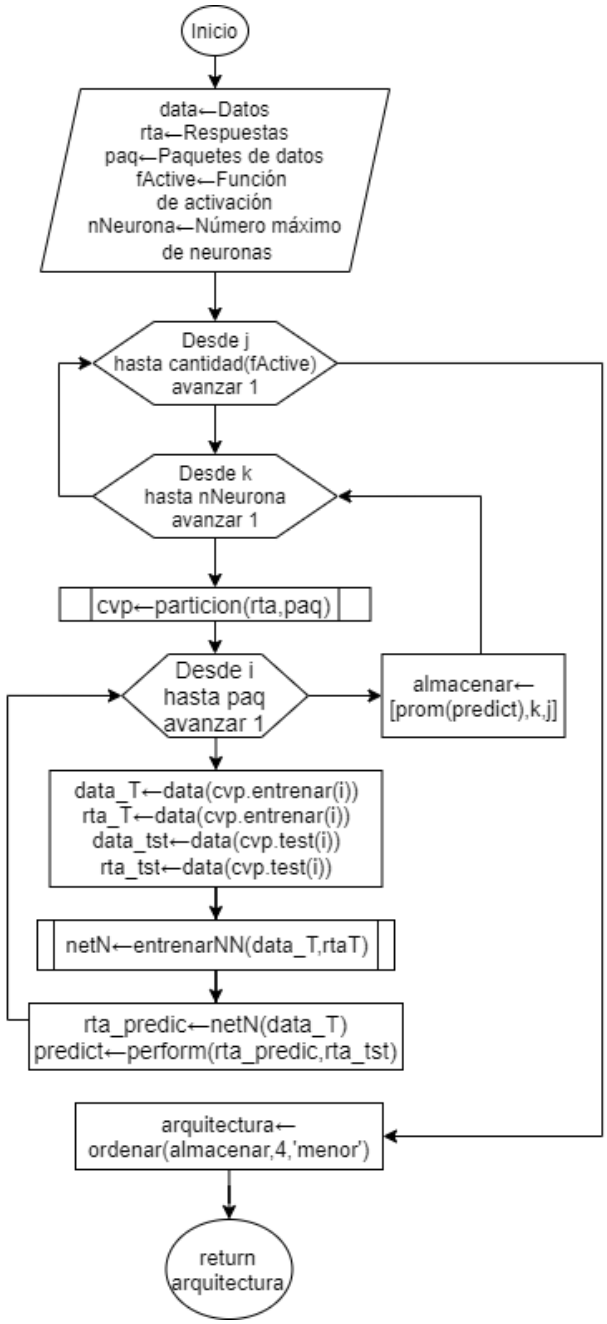
Variable	Grafica para seleccionar el parámetro de regulación Lambda VS MSE	Características seleccionadas
Cantidad de inversores		 Características seleccionadas: 1. Máxima potencia en CA para el funcionamiento nominal (P_{ac}). 2. Voltaje CA nominal para la salida del inversor (V_{ac}) 3. Azimut Solar

Fuente: Autores

6. Redes neuronales

En esta sección se presenta el algoritmo usado para entrenar distintas arquitecturas de redes neuronales. Para la clasificación de los proyectos fotovoltaicos, se usaron los datos extraídos de la ANLA, mientras para pronóstico se usó los datos generados a partir de la herramienta auxiliar. En la Figura 34 se observa un diagrama de flujo que contiene los pasos realizados para entrenar la red neuronal,

Figura 34 Diagrama de flujo del entrenamineto de la red neuronal



Fuente: Autores

6.1.1 **Redes neuronales para clasificación**

Para seleccionar la red neuronal propuesta para clasificar las clases de proyectos fotovoltaicos, se ejecuta el algoritmo del diagrama de flujo en la Figura 34. Al finalizar el proceso, se vuelve a pasar por el algoritmo aquel modelo que haya ocupado mejores resultados en cuanto a tiempo y MSE. En la Tabla 15 se muestran los detalles de entrada al algoritmo

Tabla 15 Entradas algoritmo para selección de arquitectura de la red neuronal de clasificación

Datos usados clasificación	
Base de datos	ANLA (ver Anexo A)
Cantidad de datos	100
Cantidad máxima de neuronas	30
Particiones de datos	20
Cantidad de datos que no se entrenan en la red neuronal	5
Cantidad de datos que se entrenan en la red neuronal	95
Porcentaje de datos de entrenamiento	70%
Porcentaje de datos de evaluación	15%
Porcentaje de datos de validación	15%
Número de funciones de activación usadas	4
Número máximo de iteraciones	1000
Desempeño mínimo	0
Gradiente mínimo	1e-6
Número máximo de validaciones con un error mayor al anterior	6
Algoritmo de entrenamiento	Retro propagación gradiente conjugada escalada

Fuente: Autores

Selección de arquitectura de la red neuronal: Se tomaron las cantidades de neuronas que provocaran el menor MSE, para las distintas combinaciones de funciones de activación.

La Tabla 16 representa con color verde aquellos valores que tengan un menor MSE, en caso contrario se representa con color rojo, los valores intermedios se representan con color amarillo. Se observa que la mayoría de los datos tiene un buen desempeño, es decir tiene un bajo MSE, el tiempo de ejecución en la mayoría es menor a un segundo.

Al comparar los valores de MSE de los modelos, se revela que el modelo perteneciente a la primera fila, tiene un MSE menor que el resto, también sobresale el bajo tiempo que tomó el modelo al ser entrenado. Por las razones indicadas anteriormente, se selecciona dicho modelo para ejecutarse en la herramienta final.

Tabla 16 Mejores arquitecturas de redes neuronales para clasificación.

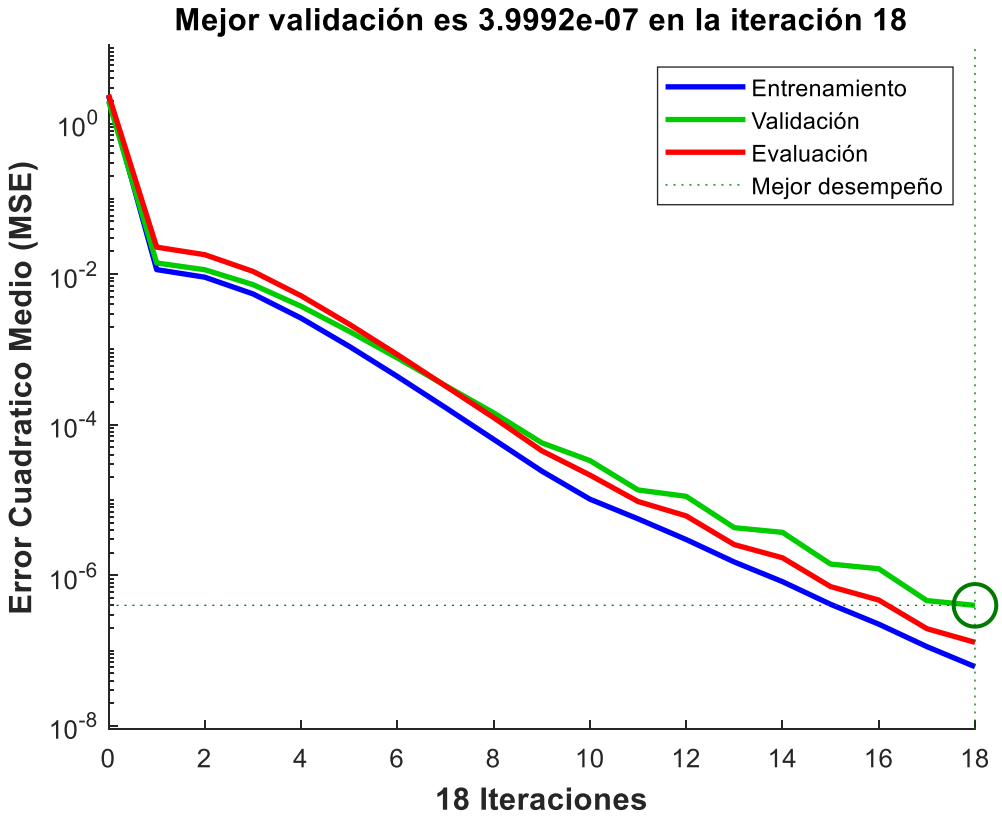
Tiempo	Desempeño (MSE) general	Desempeño entrenamiento	Desempeño validación	Desempeño test	Desempeño conjunto no entrar NN	Cantidad Neuronas en la capa oculta	Función de activación de la capa de salida	Función de activación de la capa oculta
0.187±0.024	1.28E-15	1.26E-15	1.24E-15	1.39E-15	1.23E-15	12	Tangente hiperbólica	Tangente hiperbólica
0.302±0.052	3.07E-15	3.03E-15	3.30E-15	3.01E-15	6.14E-15	2	Lineal	Sigmoidal
0.255±0.036	2.86E-14	5.97E-15	1.08E-14	1.52E-13	2.22E-11	8	Lineal	Tangente hiperbólica
0.2±0.029	2.38E-10	2.22E-10	3.60E-10	1.92E-10	2.34E-10	18	Tangente hiperbólica	ReLU
0.3±0.061	2.03E-09	1.85E-09	2.46E-09	2.40E-09	2.30E-09	26	Tangente hiperbólica	Sigmoidal
0.245±0.057	1.88E-09	7.22E-10	7.98E-09	1.15E-09	9.76E-10	10	Tangente hiperbólica	Lineal
0.475±0.26	1.38E-06	6.00E-10	9.17E-06	1.01E-09	3.59E-04	6	Lineal	ReLU
0.19±0.019	1.29E-03	8.67E-04	3.12E-03	1.44E-03	1.50E-03	28	Lineal	Lineal
0.347±0.068	1.25E-01	1.21E-01	1.04E-01	1.67E-01	1.25E-01	18	ReLU	Sigmoidal
0.866±0.317	1.25E-01	1.21E-01	1.25E-01	1.46E-01	1.25E-01	16	ReLU	Tangente hiperbólica
0.287±0.054	1.25E-01	1.21E-01	1.46E-01	1.25E-01	1.25E-01	28	Sigmoidal	Sigmoidal
0.218±0.036	1.25E-01	1.16E-01	1.67E-01	1.25E-01	1.25E-01	28	ReLU	ReLU
0.414±0.12	1.25E-01	1.16E-01	1.67E-01	1.25E-01	1.25E-01	28	ReLU	Lineal
0.216±0.029	1.25E-01	1.21E-01	1.88E-01	8.33E-02	1.25E-01	30	Sigmoidal	Tangente hiperbólica
0.221±0.023	1.25E-01	1.12E-01	2.29E-01	8.33E-02	1.25E-01	26	Sigmoidal	Lineal
0.206±0.022	1.25E-01	1.12E-01	2.29E-01	8.33E-02	1.25E-01	26	Sigmoidal	ReLU

Desempeño							
	Alto			Medio			Bajo

Fuente: Autores

Entrenamiento de la red neuronal: En la Figura 35 se muestra el entrenamiento de la red neuronal seleccionada, se observa que el error cuadrático medio disminuye a medida que las iteraciones aumentan, el conjunto de datos entrenamiento (Se representan en color azul), el conjunto de datos de evaluación (Se representa de color rojo) y el conjunto de datos de validación (Se representan de color verde), mantienen valores del MSE similares, lo que significa que el modelo alcanzó la generalización.

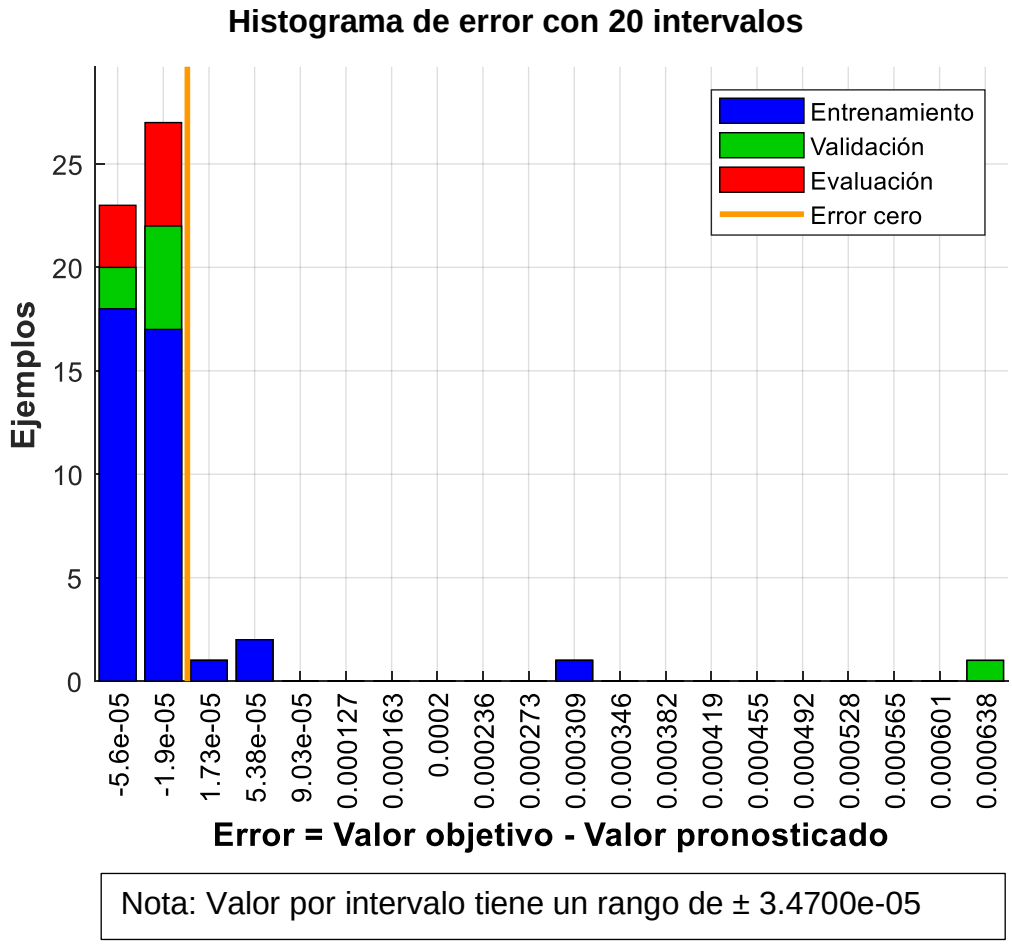
Figura 35 Desempeño de la red neuronal de clasificación



Fuente: Autores

Para observar la diferencia entre los resultados y los valores objetivos, se hace un histograma para ver la concentración de los datos, como se observa en la Figura 36 todos los valores de error son muy cercanos a cero, lo que significa que los valores pronosticados son muy cercanos a los valores que se tienen como objetivo, mostrando que la red neuronal tiene una alta eficiencia a la hora clasificar los proyectos.

Figura 36 Histograma de error con 20 intervalos. Para clasificación



Fuente: Autores

6.1.2 Redes neuronales para regresión

En la Tabla 17 se observan los detalles de la entrada al algoritmo de entrenamiento de la red neuronal, en este caso se seleccionaron los ejemplos de la clase autoconsumo, y siendo 5794 ejemplos, de los cuales se extrae un conjunto de datos de 580, para poder evaluar el modelo después de ser entrenado. Durante el entrenamiento de la red neuronal, se divide el conjunto de datos en 70% de los datos para su entrenamiento, 15% para la validación y 15% para la evaluación del modelo.

Tabla 17 Entradas algoritmo para selección de arquitectura de la red neuronal de regresión

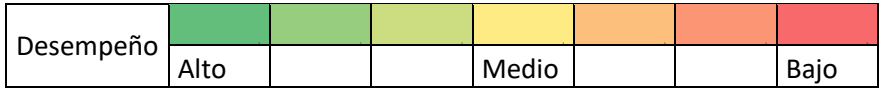
Datos usados	
Base de datos	Herramienta auxiliar
Cantidad de datos	5794
Cantidad máxima de neuronas	55
Paquetes de datos	10
Cantidad de datos que no entran al sistema	580
Cantidad de datos que entran al sistema	5211
Porcentaje de datos de entrenamiento	70%
Porcentaje de datos de evaluación	15%
Porcentaje de datos de validación	15%
Número de funciones de activación usadas	4
Número máximo de iteraciones	1000
Desempeño mínimo	0
Gradiente mínimo	1e-6
Número máximo de validaciones con un error mayor al anterior	6
Algoritmo de entrenamiento	Retro propagación gradiente conjugada escalada

Fuentes: Autores

Selección de arquitectura de la red neuronal de regresión: Se observa en la Tabla 18 que el desempeño es menor que en la red neuronal de clasificación, y se observa que el tiempo de ejecución promedio también es mayor. Se selecciona la arquitectura de la red neuronal que menor error de validación presente, siendo la fila que está resaltada.

Tabla 18 Mejores arquitecturas para la red neuronal de regresión.

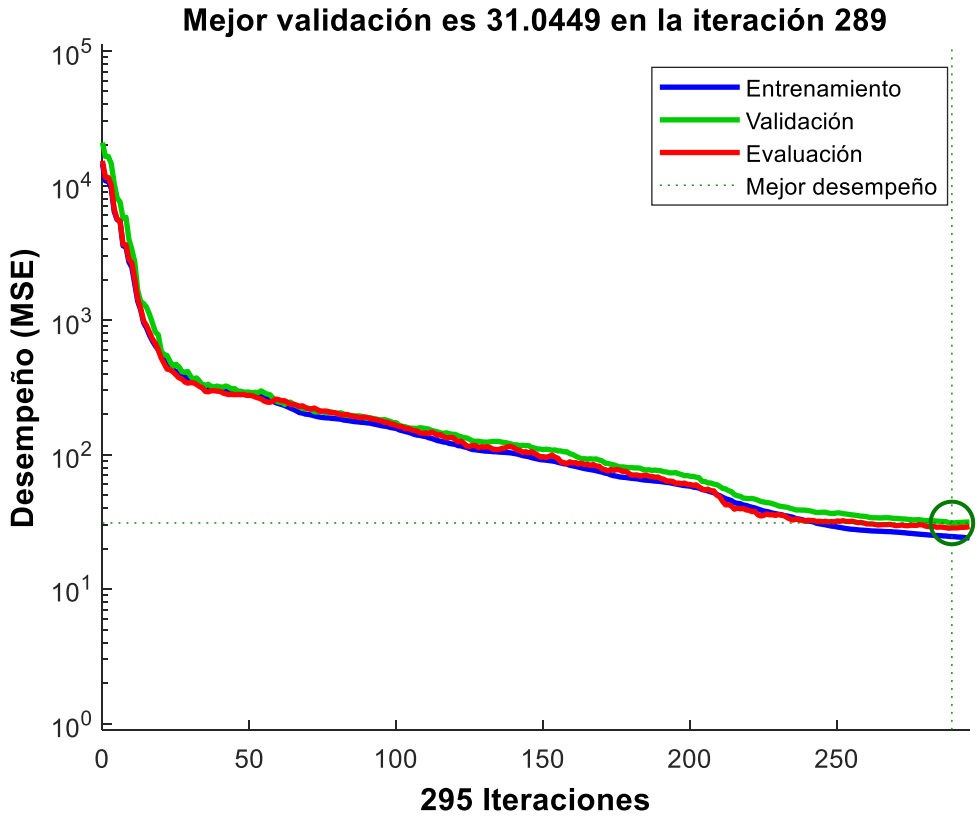
Tiempo	Desempeño (MSE) general	Desempeño entrenamiento	Desempeño validación	Desempeño test	Desempeño conjunto no entrar NN	Cantidad Neuronas en la capa oculta	Función de activación de la capa de salida	Función de activación de la capa oculta
2.85±1.32	147.53	145.28	159.87	145.68	174.60	50	Lineal	ReLU
2.58±1.66	156.57	157.85	149.57	157.64	193.11	30	ReLU	ReLU
4.68±1.78	184.97	158.11	267.29	228.06	275.57	55	ReLU	Sigmoide
3.19±0.86	250.04	202.23	352.81	370.44	341.57	50	Lineal	Sigmoide
7.01±5.34	339.12	300.17	334.04	526.05	442.26	50	Lineal	Tangente Hiperbólica
5.04±1.75	418.88	385.26	410.96	583.74	502.60	55	ReLU	Tangente Hiperbólica
1.04±0.89	2,328.07	2,412.11	2,272.69	1,991.09	2,342.18	5	ReLU	Lineal
0.64±0.13	5,585.91	5,771.04	5,637.46	4,670.08	5,608.60	5	Lineal	Lineal
1.14±0.45	15,435.57	15,731.41	17,045.31	12,444.88	15,436.69	5	Sigmoide	Lineal
3.61±3.76	15,436.10	15,731.94	17,045.82	12,445.42	15,437.22	5	Tangente Hiperbólica	Lineal
1.45±1.21	15,435.55	15,681.14	17,080.65	12,644.06	15,435.45	10	Sigmoide	ReLU
2.73±1.83	15,435.58	15,830.63	16,077.63	12,949.39	15,436.81	5	Tangente Hiperbólica	ReLU
2.56±2.13	15,435.54	16,207.81	14,047.82	13,218.05	15,434.35	15	Tangente Hiperbólica	Tangente Hiperbólica
1.7±1.4	15,435.54	16,207.81	14,047.82	13,218.05	15,434.35	15	Sigmoide	Tangente Hiperbólica
1.14±0.16	15,435.54	16,080.21	13,861.39	14,000.17	15,434.59	15	Sigmoide	Sigmoide
5.71±6.48	15,435.56	16,080.22	13,861.41	14,000.19	15,434.60	15	Tangente Hiperbólica	Sigmoide



Fuente: Autores

Entrenamiento de la red neuronal de regresión: Al igual que en la clasificación, se usó el 70% de los datos para el entrenamiento, el 15% para la evaluación y el 15% para la validación. En la Figura 37 se observa que en la iteración 289, la red neuronal obtiene su mejor desempeño, a partir de ahí su conjunto de validación comienza a tener un error mayor, por esto se detiene el entrenamiento de la red neuronal.

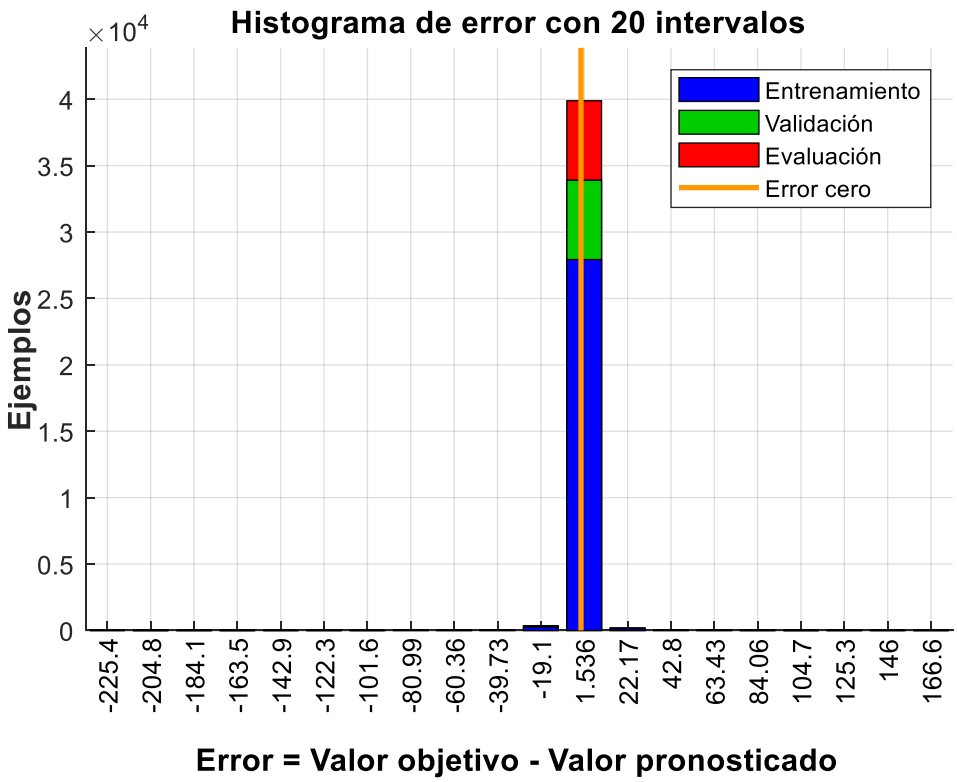
Figura 37 Desempeño de la red neuronal. Para regresión.



Fuente: Autores

En la Figura 38 se muestra un histograma con 20 intervalos que representa la calidad de las respuestas y su concentración, lo hace comparando los valores de entrada (llamados valores objetivo) y los valores pronosticados por la red neuronal, los intervalos van desde -246.036 hasta 187.236, cada uno de ellos de los valores mostrados abarca ± 20.636. La mayoría de las respuestas tiene un error que está en el rango de -39.736 y 42.806. Se observa una mayor concentración de datos en el intervalo de -0.5146± 1.5385, representando respuestas acertadas por parte de la red neuronal.

Figura 38 Histograma de error de la red neuronal de regresión.



Nota: Valor por intervalo tiene un rango de ± 20.636

Fuente: Autores

En la Tabla 19 presenta los valores de R^2. La red neuronal de regresión tiene un R^2 cercano a 1 en todos sus conjuntos, lo que significa que la red neuronal sí está generalizando con el conjunto de datos que se le suministró.

Tabla 19 R^2 de la red neuronal de regresión.

	Entrenamiento	Validación	Evaluación	General
R^2	0.99911	0.99922	0.99906	0.99912

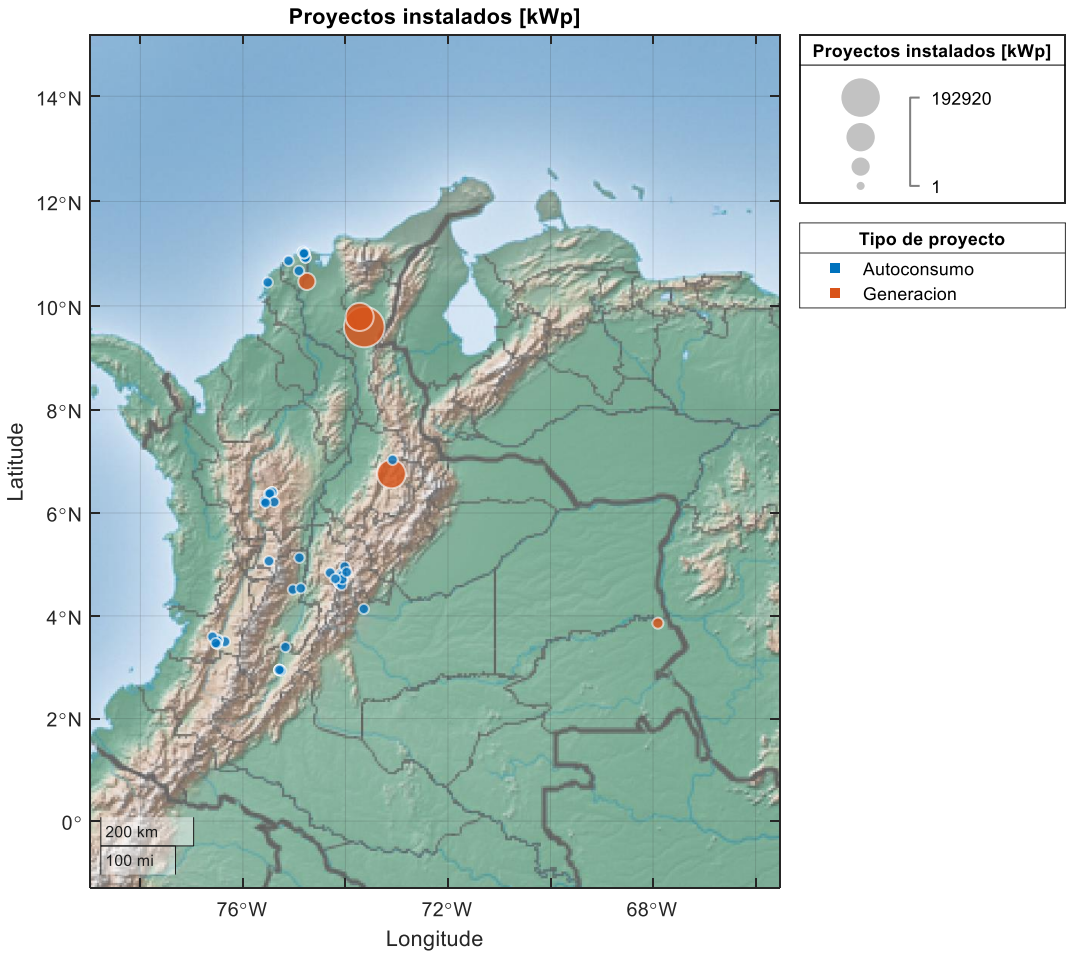
7. Resultados

7.1 Base de datos de los proyectos

Se registraron los proyectos fotovoltaicos de la base de datos de la ANLA, siendo un total de 55, donde 5 de ellos siendo de generación, y los otros 50 siendo de autoconsumo, después se relacionaron los datos ambientales (como velocidad del viento, temperatura y radiación del sol) de la base de datos de la NASA, para sus respectivas ubicaciones. Se muestra en la Figura 39 la mayoría de los proyectos son registrados en la parte central y norte del país, ninguno de ellos se registró en la parte sur, se muestra la totalidad de los datos recolectados de la ANLA, se

observa que en cuanto a las plantas de generación, la de mayor capacidad está en el Paso, Cesar, mientras la de menor capacidad está en Vichada.

Figura 39 Proyectos fotovoltaicos y su capacidad.



Fuente: Autores

7.2 Variables más relevantes para la clasificación

La selección de características se hizo por medio del Análisis de componentes de vecindario (FSCNCA) donde se seleccionan las 3 variables que obtuvieran un puntaje mayor en su relevancia, en la Tabla 20 se muestran las variables más relevantes para clasificar las 2 clases existentes en los proyectos solares fotovoltaicos (generación y autoconsumo), su relevancia se ve reflejada con el valor de su peso.

Tabla 20 Variables seleccionadas por respuesta. Clasificación.

Variable	Peso
Área [m^2]	1.155
Costos [$]	0.61182
Estimación de energía [kWh/año]	0.0058211

Fuente: Autores

7.3 Variables más relevantes para la regresión

Las variables seleccionadas para el pronóstico de las configuraciones fotovoltaicas, fueron seleccionadas a partir de las características que mejor describían su comportamiento. En la Tabla 21 se observan tres columnas, una columna con las variables que se busca pronosticar (nombrada como Respuestas), otra columna con las variables que fueron seleccionadas debido a que representaban un comportamiento más claro de las respuestas (nombrada como Variables seleccionadas), y por último una columna que contiene el peso o relevancia de las variables (nombrada como Pesos de las variables).

Tabla 21 Variables seleccionadas por respuesta. Regresión.

Respuestas	Variables seleccionadas	Pesos de las variables
Cantidad de paneles en serie	Voltaje de circuito abierto del panel (V_{oc}).	0.43855
	Voltaje máximo por MPPT ($MPPT_{Hi}$).	0.27188
	Potencia máxima con CC del inversor (P_{dc}).	0.060408
Cantidad máxima de paneles en serie	Voltaje de circuito abierto del panel (V_{oc}).	0.46395
	Voltaje máximo por MPPT ($MPPT_{Hi}$).	0.28302
	Potencia máxima con CC del inversor (P_{dc}).	0.060669
Cantidad de paneles en paralelo	Máxima CC permitida en el inversor ($I_{dc\ Máx}$).	0.67933
	Voltaje de circuito abierto del panel (V_{oc}).	0.43949
	Voltaje de CC al que alcanza la potencia nominal (V_{dc}).	0.28143

Respuestas	Variables seleccionadas	Pesos de las variables
Cantidad máxima de paneles en paralelo	Máxima CC permitida en el inversor ($I_{dc\ Máx}$).	0.7386
	Voltaje de circuito abierto del panel (V_{oc}).	0.45302
	Voltaje CA nominal para la salida del inversor (V_{ac})	0.24035
Ángulo del panel	Azimut Solar	0.93815
	Potencia requerida	0.27293
	Longitud	0.094506
Cantidad de inversores	Máxima potencia en CA para el funcionamiento nominal (P_{ac}).	0.93815
	Voltaje CA nominal para la salida del inversor (V_{ac})	0.27293
	Azimut Solar	0.094506

Fuente: Autores

7.4 Arquitecturas seleccionadas

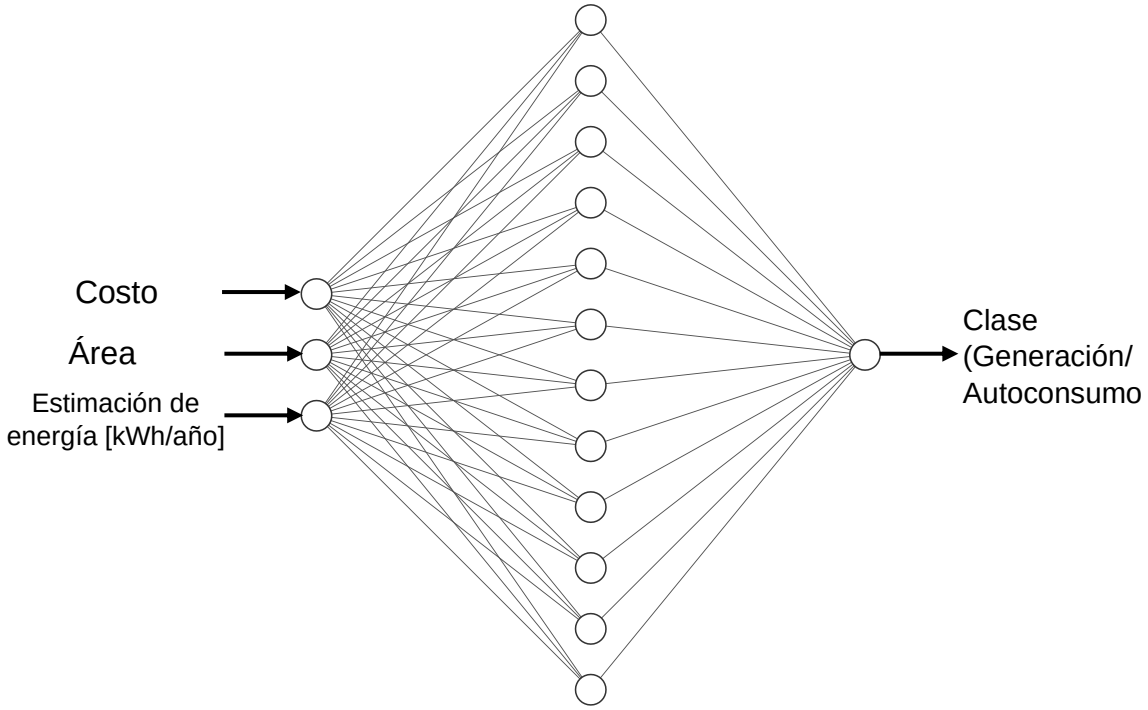
Se hicieron 2 redes neuronales, cada una con un objetivo distinto. Se representa

1. **Red neuronal de clasificación:** Se encarga de clasificar los proyectos, pueden ser de autoconsumo o de generación, tiene 3 variables de entrada, Costo [$], Estimación de energía producida al año [kWh/año], Área [m^2]. La arquitectura que mejores resultados presentó se muestra en Tabla 22 y su forma gráfica está representada en la Figura 40. Su error promedio es 1.25e-15, lo que quiere decir que tiene una clasificación muy precisa.

Tabla 22 Arquitectura seleccionada para clasifciacion.

Tiempo	Desempeño (MSE) general	Desempeño entrenamiento	Desempeño validación	Desempeño test	Desempeño conjunto no entrar NN	Cantidad Neuronas	Función de activación de la capa de salida	Función de activación de la capa oculta
0.187±0.024	1.28E-15	1.26E-15	1.24E-15	1.39E-15	1.23E-15	12	Tangente hiperbólica	Tangente hiperbólica

Figura 40 Representación red neuronal de clasificación.



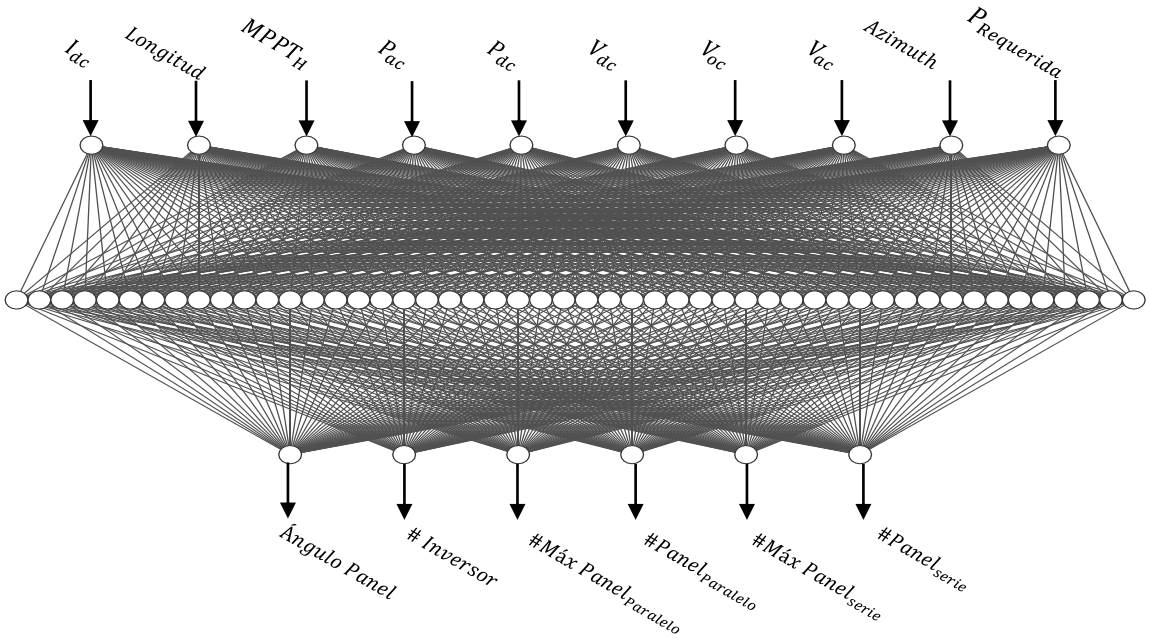
Fuente: Autores

2. **Red neuronal para predicción:** Se encarga del esbozo de la configuración de los equipos que van en serie, los equipos que van en paralelo y su cantidad máxima, adicionalmente selecciona la cantidad de inversores que debe tener. La Tabla 23 muestra el MSE de la red neuronal y detalles sobre las neuronas, como la cantidad de neuronas en la capa oculta y sus funciones de activación. Al evaluar el modelo en un conjunto de datos ajeno al conjunto de datos de entrenamiento, se observa que tiene un MSE 174.60, lo que se traduce en una desviación alta a la hora de dar la respuesta

Tabla 23 Arquitectura de la Red Neuronal para regresión.

Tiempo	Desempeño (MSE) general	Desempeño entrenamiento	Desempeño validación	Desempeño test	Desempeño conjunto no entrar NN	Cantidad Neuronas en la capa oculta	Función de activación de la capa de salida	Función de activación de la capa oculta
2.85±1.32	147.53	145.28	159.87	145.68	174.60	50	Lineal	ReLU

Figura 41 Representación para la Red Neuronal para regresión

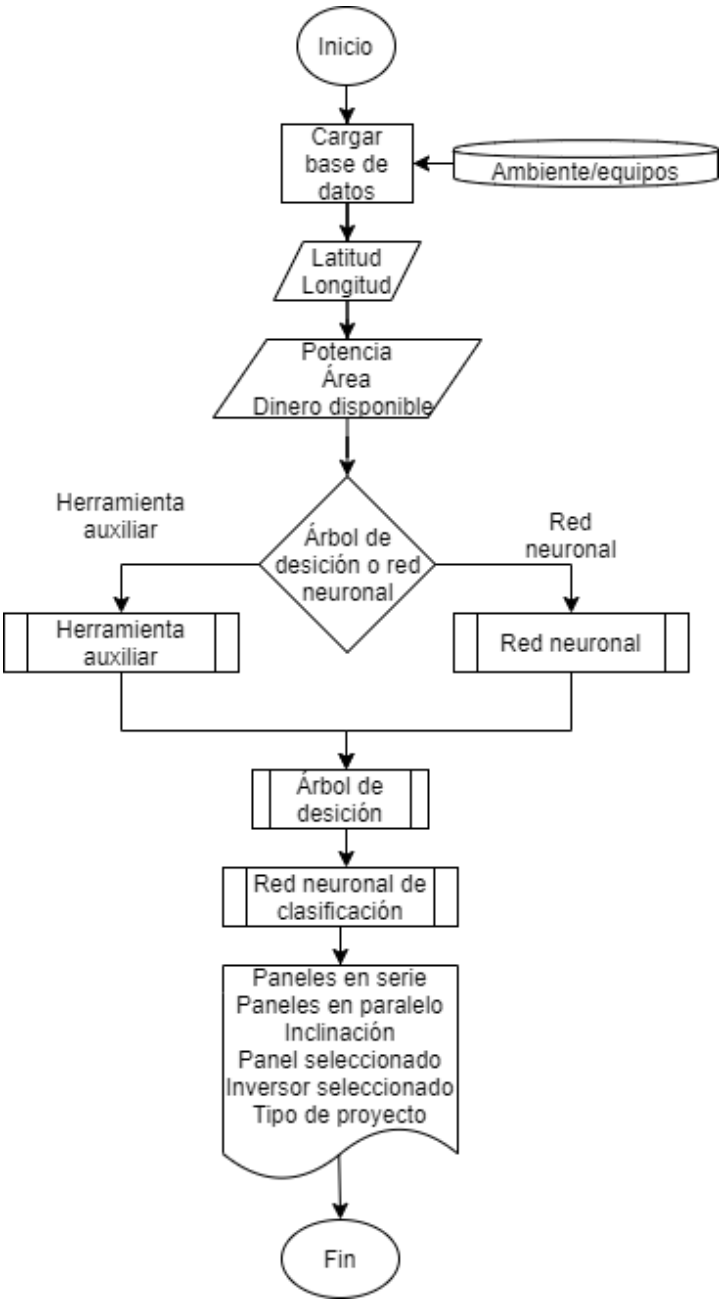


Fuente: Autores

7.5 Herramienta

En la Figura 42 se muestran los pasos seguidos por la herramienta, da inicio cargando la base de datos y recibiendo la ubicación geográfica del proyecto, a continuación, debe recibir datos como el área, la potencia y el presupuesto, para posteriormente dar a elegir entre la herramienta que se desee ejecutar, herramienta auxiliar o redes neuronales. Los resultados son enviados a un árbol de decisión encargado de seleccionar la configuración que más se ajuste a las necesidades del usuario y finalmente los resultados ingresan a la red neuronal de clasificación, la cual estimará si el proyecto es de generación o autoconsumo.

Figura 42 Diagrama de flujo de la herramienta final.



Fuente: Autores

7.6 Ensayo de la herramienta

Se seleccionaron 2 ejemplos aleatoriamente, para evaluar la herramienta desarrollada. En la Tabla 24 se observan los resultados producidos por sus 2 distintas herramientas, donde ninguna de las 2 herramientas logró encontrar la cantidad de dinero necesaria para suplir las necesidades del usuario. Se observa

que las redes neuronales tienen valores más altos, exceptuando el precio del proyecto, el cual es menor con respecto al de la herramienta auxiliar, también se detalla que, para los 2 casos, las redes neuronales logran clasificar correctamente el proyecto. La herramienta auxiliar logra ajustarse a las necesidades del usuario, pero superando el precio disponible por el usuario, y su tiempo de ejecución es mayor en los ejemplos presentados.

Tabla 24 Ejemplos ejecutados en la herramienta

	Panel Para	# Inversor	Ángulo Panel	Área Terreno	Potencia Generada	Precio Proyecto	Clase	Herramienta	Área Disponible	Potencia Necesaria	Dinero Disponible	Tiempo ejecución	# Ejemplo
2	1	5264	11	1.6. E+08	8.2. E+06	9.3. E+12	Generación	Redes neuronales	3.1. E+03	6.6. E+05	3.6. E+08	9.487	1
1	1	944	12	1.3. E+03	6.6. E+05	5.2. E+15	Generación	Herramienta auxiliar				9.559	
1	2	1990	2	1.9. E+05	3.1. E+06	6.3. E+12	Generación	Redes neuronales	3.2. E+02	1.2. E+06	4.1. E+08	2.067	2
1	1	1733	7	2.3. E+03	1.2. E+06	9.5. E+15	Generación	Herramienta auxiliar				6.474	

Opción	
Menor	Mayor

Fuente: Autores

8. Conclusiones

En esta tesis se desarrolló una herramienta computacional, que por medio de redes neuronales y una herramienta auxiliar permitió estimar las configuraciones fotovoltaicas según sus parámetros de entrada. Mediante un árbol de decisión se logró la selección de los equipos que más se aproximan a los requerimientos del usuario.

Durante el levantamiento de datos de distribución de proyectos solares en Colombia, se muestra una gran concentración en la región Andina con el 72.72 % del total, en la región Caribe se representó un 23.65 %, en la región Amazónica y la Orinoquía se encontró un 3.63% de los datos, siendo las regiones con menores proyectos fotovoltaicos registrados en la ANLA, hasta el momento de la escritura de este documento (noviembre de 2020).

Se identificaron y seleccionaron las características más relevantes para la clasificación de los proyectos y para el pronóstico de las configuraciones fotovoltaicas, donde se observó que las respuestas pueden describirse con las 3 variables más importantes.

Se seleccionó las arquitecturas que mejor se adaptan a los datos de clasificación y de regresión, se hizo comparando cada uno de los modelos y seleccionando el modelo que tuviese un menor MSE y un R^2 más cercano a 1.

En el ensayo de la herramienta se pudo observar que la red neuronal aproxima las cantidades de paneles en serie, en paralelo, sus ángulos y las cantidades de inversores, esto se debe al grado de generalización que tiene. Adicionalmente se observa la correcta clasificación por parte de la red neuronal.

La red neuronal tiene un menor desempeño en algunas características como el área y la potencia, pero es capaz de encontrar configuraciones que pueden dar un menor precio al que genera la herramienta auxiliar.

9. Recomendaciones

Se recomienda adicionar un sistema de gestión de base de datos conectado a la web, en el cual pueda almacenar los datos de los proyectos que se realicen usando la herramienta, esto para poder expandir la base de datos.

También se recomienda ampliar la base de datos, usando un conjunto de datos con los diseños óptimos, esto para que el modelo predictivo pueda dar mejores resultados en cuanto al diseño básico de los sistemas fotovoltaicos. Adicionalmente lograr que se interpreten las geometrías de los sitios donde se quiere realizar el proyecto, esto para tener en cuenta posibles lugares donde no se puedan ubicar los paneles solares.

La interface puede mejorarse y hacer más amigable la interacción con el usuario, permitiendo adicionalmente visualizar el proyecto, y permitirle más interacción con los equipos que se quieren usar.

10. Biografía

[1] UPME, “Circular Externa No 046-2019,” 2019.

[2] “Microsoft Power BI.” [Online]. Available: https://app.powerbi.com/view?r=eyJrIjoiNzBhN2Q4YmMtN2IxMy00Mjg2LWJhZTctMjRkNWE2NDdlMzI0IiwidCI6IjgxNTAwZjZkLWJjZTktNDgzNC1iNDQ2LTc0YjVmYjljZjEwZSIsImMiOjh9. [Accessed: 15-Jun-2020].

[3] “PV Performance Modeling Collaborative | PV_LIB Toolbox,” *PV Lib*, 2018. [Online]. Available: https://pvpmc.sandia.gov/applications/pv_lib-toolbox/. [Accessed: 02-Jun-2020].

[4] A. Mellita and M. Benghanem, “Sizing of stand-alone photovoltaic systems using neural network adaptive model,” *Desalination*, vol. 209, no. 1-3 SPEC. ISS., pp. 64–72, 2007, doi: 10.1016/j.desal.2007.04.010.

[5] L. Hontoria, J. Aguilera, and P. Zufiria, “A new approach for sizing stand alone photovoltaic systems based in neural networks,” *Sol. Energy*, vol. 78, no. 2, pp. 313–319, 2005, doi: 10.1016/j.solener.2004.08.018.

[6] M. H. Alomari, J. Adeeb, and O. Younis, “Solar photovoltaic power forecasting in Jordan using artificial neural networks,” *Int. J. Electr. Comput. Eng.*, vol. 8, no. 1, pp. 497–504, 2018, doi: 10.11591/ijece.v8i1.pp497-504.

[7] A. K. Yadav and S. S. Chandel, “Solar radiation prediction using Artificial Neural Network techniques: A review,” *Renew. Sustain. Energy Rev.*, vol. 33, pp. 772–781, 2014, doi: 10.1016/j.rser.2013.08.055.

[8] F. Rodríguez, A. Fleetwood, A. Galarza, and L. Fontán, “Predicting solar energy generation through artificial neural networks using weather forecasts for microgrid control,” *Renew. Energy*, vol. 126, pp. 855–864, 2018, doi: 10.1016/j.renene.2018.03.070.

[9] X. Luo *et al.*, *Solar water heating system*. 2018.

[10] R. D. Prasad and R. C. Bansal, “Photovoltaic systems,” *Handb. Renew. Energy Technol.*, pp. 205–224, 2011, doi: 10.1142/9789814289078_0009.

[11] M. C. MERINO, “Escuela Técnica Superior de Ingeniería del Diseño Grado en Ingeniería Electrónica Industrial y Automática,” 2019.

[12] meteorología y estudios ambientales (IDEAM) Instituto de hidrología, “IRRADIACIÓN GLOBAL HORIZONTAL MEDIO DIARIO ANUAL,” 2014. [Online]. Available: http://atlas.ideam.gov.co/basefiles/RadiacionSolar13.pdf. [Accessed: 03-Jun-2020].

[13] Subdirección de Meteorología and IDEAM, “Clasificaciones Climaticas Colombia,” 05-Jun-2020. [Online]. Available: http://www.ideam.gov.co/documents/21021/418894/Clasificaciones+Climáticas.pdf/15d8c5c8-a850-43f9-a954-2ce98d927f98. [Accessed: 04-Jul-2020].

[14] K. A. HERNÁNDEZ and J. S. CARRILLO CRUZ, “ANÁLISIS DE LA CURVA DE DEMANDA ELÉCTRICA PARA USUARIOS RESIDENCIALES ESTRATO 4 EN LA CIUDAD DE BOGOTÁ ANTE DIFERENTES ESCENARIOS DE LOS HÁBITOS DE CONSUMO,” *Univ. Dist. Fr. JOSÉ CALDAS*, vol. 01, pp. 1–163, 2017.

[15] O. Perpiñan Lamigueiro, *E S Fotovoltaica*. 2012.

[16] A. Smets, K. Jäger, O. Isabella, R. A. C. M. M. Van Swaaij, and M. Zeman, *Solar Energy - The physics and engineering of photovoltaic conversion, technologies and systems*. 2016.

[17] E. M. Knowles, *Oxford Dictionary of Phrase and Fable*. Oxford University Press, 2005.

[18] S. Rusell and P. Norvig, *Inteligencia Artificial. Un Enfoque Moderno*. 2004.

[19] V. Novák, I. Perfilieva, and J. Močkoř, *Mathematical Principles of Fuzzy Logic*, no. January. 1999.

[20] F. Behrooz, N. Mariun, M. H. Marhaban, M. A. M. Radzi, and A. R. Ramli, “Review of control techniques for HVAC systems-nonlinearity approaches based on fuzzy cognitive maps,” *Energies*, vol. 11, no. 3, 2018, doi: 10.3390/en11030495.

[21] B. G. BUCHANAN, “Expert systems: working systems and the research literature,” *Expert Syst.*, vol. 3, no. 1, pp. 32–50, 1986, doi: 10.1111/j.1468-0394.1986.tb00192.x.

[22] D. Huttenlocher, “Computer vision,” *Comput. Sci. Handbook, Second Ed.*, pp. 43-1-43–23, 2004, doi: 10.4324/9780429042522-10.

[23] E. Gharibkhani and R. C. Akdeniz, “Advantages and Disadvantages of Machine Vision Applications for Automatic Sorting of Aflatoxin Contaminated Dried Figs,” vol. 10, pp. 7–10, 2014.

[24] A. Cortez, H. Vega, and J. Pariona, “Procesamiento de lenguaje natural robusto,” *Prim. encuentro Grup. Investig. sobre Proces. del Leng.*, vol. 2013, no. 3, p. 147, 2011.

[25] A. Geron, *Hands-On Machine Learing With Scikit-Learn & Tensor Flow*. O’Reilly Media, 2017.

[26] T. Yang, Q. Zhao, X. Wang, and Q. Zhou, “Sub-pixel chessboard corner localization for camera calibration and pose estimation,” *Appl. Sci.*, vol. 8, no. 11, 2018, doi: 10.3390/app8112118.

[27] F. Herrera, “Big Data: Preprocesamiento y calidad de datos,” *Novática*, no. 237, p. 17, 2016.

[28] R. Goyena and A. . Fallis, *Data mining concepts and techniques*, vol. 53, no. 9. 2019.

[29] A. Olinsky, S. Chen, and L. Harlow, “The comparative efficacy of imputation methods for missing data in structural equation modeling,” *Eur. J. Oper. Res.*, vol. 151, no. 1, pp. 53–79, 2003, doi: 10.1016/S0377-2217(02)00578-7.

[30] H. Junninen, H. Niska, K. Tuppurainen, J. Ruuskanen, and M. Kolehmainen, “Methods for imputation of missing values in air quality data sets,” *Atmos. Environ.*, vol. 38, no. 18, pp. 2895–2907, 2004, doi: 10.1016/j.atmosenv.2004.02.026.

[31] D. Berrar, “Cross-validation,” *Encycl. Bioinforma. Comput. Biol. ABC Bioinforma.*, vol. 1–3, no. April, pp. 542–545, 2018, doi: 10.1016/B978-0-12-809633-8.20349-X.

[32] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2009.

[33] W. Yang, K. Wang, and W. Zuo, “Neighborhood component feature selection for high-dimensional data,” *J. Comput.*, vol. 7, no. 1, pp. 162–168, 2012, doi: 10.4304/jcp.7.1.161-168.

[34] H. H. Hsu, C. Y. Chang, and C. H. Hsu, *Big Data Analytics for Sensor-Network Collected Intelligence*. 2017.

[35] H. He, Y. Bai, E. A. Garcia, and S. Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” *Proc. Int. Jt. Conf. Neural Networks*, no. 3, pp. 1322–1328, 2008, doi: 10.1109/IJCNN.2008.4633969.

[36] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu, “Definitions, methods, and applications in interpretable machine learning,” *Proc. Natl. Acad. Sci. U. S. A.*, vol. 116, no. 44, pp. 22071–22080, 2019, doi: 10.1073/pnas.1900654116.

[37] N. K. K, “Naive Bayes : Text Classifier for Spam Detection. - Naveen Kumar K - Medium,” 2019. [Online]. Available: https://medium.com/@naveeen.kumar.k/naive-bayes-spam-detection-7d087cc96d9d. [Accessed: 03-Jun-2020].

[38] I. H. W. and E. Fran, “Introducción a Data Mining,” *SAS Train. courses*, vol. (2thEd.), no. Dm, p. 34, 2007.

[39] K. A. Gaurav and L. Patel, *Machine Learning With R*. 2020.

[40] “How to Bulid a Neural Network in Python | Blog | Dimensionless.” [Online]. Available: https://dimensionless.in/building-a-neural-network-in-python-language-modeling-task/. [Accessed: 03-Jun-2020].

[41] David Kriesel, “A Brief Intoduction to Neural Networks,” *Nonlinear Syst. Identif.*, pp. 239–297, 2001, doi: 10.1007/978-3-662-04323-3_10.

[42] C. Bircano and T. Çizge, “Yapay Sinir A ˘ glarında Aktivasyon Fonksiyonlarının Kar ¸ sıla ¸ stırılması A Comparison of Activation Functions in Artificial Neural Networks,” *2018 26th Signal Process. Commun. Appl. Conf.*, pp. 1–4.

[43] P. Marius-Constantin, V. E. Balas, L. Perescu-Popescu, and N. Mastorakis, “Multilayer perceptron and neural networks,” *WSEAS Trans. Circuits Syst.*, vol. 8, no. 7, pp. 579–588, 2009.

[44] M. N. Noor, A. S. Yahaya, N. A. Ramli, and A. M. M. Al Bakri, “Filling missing data using interpolation methods: Study on the effect of fitting distribution,” *Key Eng. Mater.*, vol. 594–595, pp. 889–895, 2014, doi: 10.4028/www.scientific.net/KEM.594-595.889.

Anexo A : **Datos recolectados de la ANLA**

#	Lugar	Latitud	Longitud	Tipo de superficie	Porcentaje de energía eléctrica suplida [%]	Capacidad [kWP]	Estimación de potencia producida anualmente [kWh/año]	Prevención emisión atmosférica [kgCO2/año]	Área [m^2]	Tipo de proyecto	Link
1	Vía Bitaco la Cumbre Valle del Cauca	3.593	-76.593	Techo inclinado	66	28	31100	11413.13		Autoconsumo	ANLA01
2	Medellín	6.197	-75.559	Plano	20	99.36	144400	53000		Autoconsumo	ANLA02
3	Urabá apartado Antioquia	6.364	-75.493	Plano		440	548900	220109		Autoconsumo	ANLA03
4	Soledad Atlántico	10.926	-74.778	Plano	85	165	275556	106900	1056.68	Autoconsumo	ANLA04
5	Piedras Tolima	4.537	-74.871	Plano		97.8	136541	57105		Autoconsumo	ANLA05
6	Cali Valle del cauca	3.462	-76.512	Plano		162	250630	91981		Autoconsumo	ANLA06
7	Bogotá Cundinamarca	4.641	-74.120	Plano		351	485270	194539	2500	Autoconsumo	ANLA07
8	Girardota, Antioquía	6.402	-75.433	Plano		801.36	1064000	390488		Autoconsumo	ANLA08
9	Barranquilla	11.019	-74.835	Plano		345.6	522000	191574		Autoconsumo	ANLA09
10	Cartagena, Bolívar	10.452	-75.515	Plano		50.35	75755	27802		Autoconsumo	ANLA10
11	Alvarado Tolima	4.513	-75.025	No especifica		148.2	230,599	149889.35		Autoconsumo	ANLA11
12	Neiva, Huila	2.941	-75.296	No especifica		54	81000	29,727		Autoconsumo	ANLA12
13	Soledad, Atlántico	10.926	-74.779	No especifica		85	140970	51737		Autoconsumo	ANLA13
14	Ceuta Funza, Cundinamarca	4.721	-74.196	No especifica		3.2	3986	1598.4		Autoconsumo	ANLA14
15	Puerto Inírida, Guainía	3.859	-67.905	Plano	100	2494.8	37816000	2943000	15000	Generación	ANLA15
16	El paso, Cesar	9.788	-73.723	Plano	100	86200	175573689	64435540	2060000	Generación	ANLA16
17	Palmira, Valle del Cauca	3.498	-76.353	Techo inclinado		902.4	1274	1,756,610	3,607.07	Autoconsumo	ANLA17
18	Bogotá	4.698	-74.055	Plano		37.2	47802	17543	250	Autoconsumo	ANLA18
19	Suba, Bogotá, Cundinamarca	4.801	-74.052	Plano		30.2	38,325	14,065		Autoconsumo	ANLA19
20	Soacha, Cundinamarca	4.601	-74.075	Plano		12.96	16,835	6,178.45	105	Autoconsumo	ANLA20
21	Facatativá, Cundinamarca	4.839	-74.298	Techo inclinado	36	97.9	135000	52380	4,287.94	Autoconsumo	ANLA21
22	El paso, Cesar	9.595	-73.632	Plano	100	192920	338,015,185	219710	4371700	Generación	ANLA22
23	Salamina, Magdalena	10.470	-74.753	Plano	100	23916	3762000	13,878	300000	Generación	ANLA23
24	Los Santos, Santander	6.756	-73.103	Plano	100	90000	2.60E+07	2000000000	250000	Generación	ANLA24
25	Sabanalarga, Barranquilla.	10.673	-74.907	Plano		327.6	496000	182032	3,409.34	Autoconsumo	ANLA25
26	Medellín, Antioquia	6.230	-75.570	Plano	34.64042685	505.3	671955	252391	3400	Autoconsumo	ANLA26
27	Montevideo en la ciudad de Bogotá D.C	4.642	-74.120	Plano		30	38,063	13969	270	Autoconsumo	ANLA27

#	Lugar	Latitud	Longitud	Tipo de superficie	Porcentaje de energía eléctrica suplida [%]	Capacidad [kWP]	Estimación de potencia producida anualmente [kWh/año]	Prevención emisión atmosférica [kgCO2/año]	Área [m^2]	Tipo de proyecto	Link
28	Cali, Valle del Cauca	3.463	-76.502	Techo inclinado		240.84	365.54	134153	1680	Autoconsumo	ANLA28
29	Neiva	2.949	-75.288	Plano	30	10.8	16200	3200	5000	Autoconsumo	ANLA29
30	Bogotá, Cundinamarca	4.713	-74.063	Plano		10.88	13890	5010	70	Autoconsumo	ANLA30
31	Caldas, Manizales	5.066	-75.491	Plano		9.72	12120	4448	69.69	Autoconsumo	ANLA31
32	Bogotá, Cundinamarca	4.642	-74.121	Plano		30	38063	7574.537	200	Autoconsumo	ANLA32
33	Punta Cangrejo, Juan de Acosta, Atlántico	10.864	-75.109	Techo inclinado		10.62	16,947	6,219.50		Autoconsumo	ANLA33
34	Cajicá – Zipaquirá, Cundinamarca	4.959	-74.015	Plano		15	22700	9		Autoconsumo	ANLA34
35	Barranquilla, Atlántico	11.008	-74.803	Plano		47.25	72,300	26,534	347.77	Autoconsumo	ANLA35
36	Villavicencio, Meta	4.135	-73.642	Techo inclinado	50.67620963	14.16	3,719.35	740.15	20	Autoconsumo	ANLA36
37	Suba, Bogotá, Cundinamarca	4.825	-74.074	Techo inclinado		1,427.80	10,220	3,750.74		Autoconsumo	ANLA37
38	Girardota, Antioquia	6.403	-75.432	Techo inclinado	30	800	1,064,000	412,000	9236.11	Autoconsumo	ANLA38
39	Mariquita, Tolima	5.131	-74.898	Techo inclinado	30	444.6	698,800	256460		Autoconsumo	ANLA39
40	Girardota, Antioquia	6.378	-75.474	Plano		196	258,500	100298		Autoconsumo	ANLA40
41	Cali, Valle del Cauca	3.471	-76.528	Techo plano		5.67	7,670	2814		Autoconsumo	ANLA41
42	Sopó, Cundinamarca	4.851	-73.977	Techo plano	60	8.32	11900	7695		Autoconsumo	ANLA42
43	Neiva, Huila	2.952	-75.288	Techo plano		59.36	81442	29889		Autoconsumo	ANLA43
44	Vereda Potosí, Villa vieja, HUILA	3.393	-75.171	Plano	15.31	28.06	44649.07	17904		Autoconsumo	ANLA44
45	Barranquilla atlántico	11.034	-74.827	Plano		1959	2859400	1069378		Autoconsumo	ANLA45
46	Piedecuesta Santander	7.030	-73.081	Techo inclinado		1	1542	611.18		Autoconsumo	ANLA46
47	Yumbo valle del cauca	3.521	-76.497	Techo inclinado		207				Autoconsumo	ANLA47
48	Yumbo valle del cauca	3.521	-76.497	Techo inclinado		739.2				Autoconsumo	ANLA48
49	Neiva Huila	2.948	-75.288	Techo inclinado		54	81000	29729		Autoconsumo	ANLA49
50	Barranquilla Atlántico	11.006	-74.810	Techo plano		17.7	29830	10947		Autoconsumo	ANLA50
51	Neiva Huila	2.932	-75.254	Techo plano		16.2	24300	8918		Autoconsumo	ANLA51
52	Guarne, Antioquia	6.213	-75.390	Techo inclinado	100	120	175,900	64555	1,235.07	Autoconsumo	ANLA52
53	Barranquilla Atlántico	11.003	-74.789	No especifica		180.8	285000	114300	1360	Autoconsumo	ANLA53
54	Bogotá, Cundinamarca	4.679	-74.122	No especifica		296.4	438,376	160884		Autoconsumo	ANLA54
55	Yumbo valle del cauca	3.541	-76.492	No especifica	17.39	216	300500	106940		Autoconsumo	ANLA55

Anexo B : **Tabla de dimensionamiento de la herramienta auxiliar**

#Paneles serie	#Máx paneles serie	#Paneles Paralelo	#Máx Paneles Paralelo	#Inversor	Área sombra panel	Área del terreno	Potencia generada	Precio proyecto	Código Panel	Código Inversor	Latitud	Longitud	Radiación	Área disponible	Potencia necesaria	Dinero disponible
1	1	1	2	166	1.34. E+00	2.22. E+02	1.16. E+05	9.1E+14	328	399	3.54	- 76.49	4.14	1.06. E+03	1.16. E+05	7.16. E+07
1	1	1	2	708	1.33. E+00	9.40. E+02	4.96. E+05	3.9E+15	328	399	4.68	- 74.12	4.99	1.06. E+03	4.96. E+05	1.80. E+08
1	1	1	2	942	1.42. E+00	1.34. E+03	6.59. E+05	5.2E+15	328	399	11.00	- 74.79	4.99	1.06. E+03	6.60. E+05	2.68. E+08
1	1	1	2	1370	1.37. E+00	1.87. E+03	9.59. E+05	7.5E+15	328	399	6.21	- 75.39	5.81	1.06. E+03	9.59. E+05	3.56. E+08
1	1	1	2	634	1.33. E+00	8.41. E+02	4.44. E+05	3.5E+15	328	399	2.93	- 75.25	4.54	1.54. E+03	4.44. E+05	3.64. E+08
1	1	1	2	958	1.42. E+00	1.36. E+03	6.71. E+05	5.3E+15	328	399	11.01	- 74.81	4.14	2.02. E+03	6.71. E+05	3.00. E+08
1	1	1	2	870	1.33. E+00	1.15. E+03	6.09. E+05	4.8E+15	328	399	2.95	- 75.29	4.54	2.50. E+03	6.09. E+05	2.40. E+08
1	1	1	2	757	1.34. E+00	1.01. E+03	5.30. E+05	4.2E+15	328	399	3.52	- 76.50	4.99	2.64. E+03	5.30. E+05	1.81. E+08
1	1	1	2	692	1.34. E+00	9.27. E+02	4.84. E+05	3.8E+15	328	399	3.52	- 76.50	6.19	2.78. E+03	4.84. E+05	1.21. E+08
1	1	1	2	418	1.36. E+00	5.69. E+02	2.93. E+05	2.3E+15	328	399	7.03	- 73.08	5.81	2.92. E+03	2.93. E+05	6.08. E+07
1	1	1	2	944	1.42. E+00	1.34. E+03	6.61. E+05	5.2E+15	328	399	11.03	- 74.83	4.46	3.05. E+03	6.61. E+05	3.60. E+08
1	1	1	2	319	1.34. E+00	4.28. E+02	2.23. E+05	1.8E+15	328	399	3.39	- 75.17	4.14	3.19. E+03	2.24. E+05	1.28. E+08
1	1	1	2	705	1.33. E+00	9.35. E+02	4.94. E+05	3.9E+15	328	399	2.95	- 75.29	5.81	3.33. E+03	4.94. E+05	1.68. E+08
1	1	1	2	21	1.34. E+00	2.81. E+01	1.47. E+04	1.2E+14	328	399	4.85	- 73.98	4.54	3.47. E+03	1.45. E+04	1.55. E+07
1	1	1	2	150	1.34. E+00	2.01. E+02	1.05. E+05	8.2E+14	328	399	3.47	- 76.53	5.21	3.61. E+03	1.05. E+05	8.47. E+07
1	1	1	2	311	1.38. E+00	4.29. E+02	2.18. E+05	1.7E+15	328	399	6.38	- 75.47	5.85	2.50. E+02	2.18. E+05	1.54. E+08
1	1	1	2	179	1.34. E+00	2.40. E+02	1.25. E+05	9.8E+14	328	399	5.13	- 74.90	4.14	1.78. E+02	1.25. E+05	9.63. E+07
1	1	1	2	84	1.38. E+00	1.16. E+02	5.88. E+04	4.6E+14	328	399	6.40	- 75.43	4.54	1.05. E+02	5.88. E+04	5.61. E+07
1	1	1	2	635	1.34. E+00	8.50. E+02	4.45. E+05	3.5E+15	328	399	4.82	- 74.07	4.54	4.29. E+03	4.44. E+05	1.78. E+08
1	1	1	2	488	1.33. E+00	6.47. E+02	3.42. E+05	2.7E+15	328	399	4.14	- 73.64	4.54	3.41. E+03	3.42. E+05	1.47. E+08
1	1	1	2	341	1.42. E+00	4.83. E+02	2.39. E+05	1.9E+15	328	399	11.01	- 74.80	4.54	3.40. E+03	2.39. E+05	1.17. E+08
1	1	1	2	251	1.34. E+00	3.36. E+02	1.76. E+05	1.4E+15	328	399	4.96	- 74.01	5.85	2.70. E+02	1.76. E+05	8.68. E+07
1	1	1	2	1999	1.45. E+00	2.91. E+03	1.40. E+06	1.1E+16	328	399	10.86	- 75.11	5.81	1.68. E+03	1.40. E+06	2.46. E+08
1	1	1	2	81	1.33. E+00	1.08. E+02	5.67. E+04	4.5E+14	328	399	4.64	- 74.12	5.28	5.00. E+03	5.70. E+04	3.03. E+07
1	1	1	2	90	1.36. E+00	1.22. E+02	6.30. E+04	4.9E+14	328	399	5.07	- 75.49	5.81	7.00. E+01	6.32. E+04	3.12. E+07
1	1	1	2	69	1.33. E+00	9.17. E+01	4.83. E+04	3.8E+14	328	399	4.71	- 74.06	4.99	6.97. E+01	4.85. E+04	2.74. E+07
1	1	1	2	195	1.33. E+00	2.59. E+02	1.37. E+05	1.1E+15	328	399	2.95	- 75.29	4.54	2.00. E+02	1.36. E+05	8.68. E+07
1	1	1	2	63	1.34. E+00	8.44. E+01	4.41. E+04	3.5E+14	328	399	3.46	- 76.50	4.14	2.49. E+02	4.40. E+04	2.29. E+07

#Paneles serie	#Máx paneles serie	#Paneles Paralelo	#Máx Paneles Paralelo	#Inversor	Área sombra panel	Área del terreno	Potencia generada	Precio proyecto	Código Panel	Código Inversor	Latitud	Longitud	Radiación	Área disponible	Potencia necesaria	Dinero disponible
1	1	1	2	89	1.33. E+00	1.18. E+02	6.23. E+04	4.9E+14	328	399	4.64	- 74.12	4.14	2.99. E+02	6.21. E+04	5.53. E+07
1	1	1	2	306	1.37. E+00	4.18. E+02	2.14. E+05	1.7E+15	328	399	6.23	- 75.57	4.54	3.48. E+02	2.15. E+05	1.42. E+08
1	1	1	2	97	1.41. E+00	1.37. E+02	6.79. E+04	5.3E+14	328	399	10.67	- 74.91	4.78	2.00. E+01	6.77. E+04	3.06. E+07
1	1	1	2	387	1.34. E+00	5.18. E+02	2.71. E+05	2.1E+15	328	399	4.84	- 74.30	4.54	9.59. E+01	2.71. E+05	2.83. E+07
1	1	1	2	872	1.33. E+00	1.16. E+03	6.10. E+05	4.8E+15	328	399	4.60	- 74.08	5.81	1.72. E+02	6.11. E+05	1.55. E+08
1	1	1	2	976	1.34. E+00	1.31. E+03	6.83. E+05	5.4E+15	328	399	4.80	- 74.05	4.54	2.48. E+02	6.83. E+05	2.82. E+08
1	1	1	2	1733	1.33. E+00	2.30. E+03	1.21. E+06	9.5E+15	328	399	4.70	- 74.05	6.19	3.24. E+02	1.21. E+06	4.09. E+08
1	1	1	2	35	1.34. E+00	4.69. E+01	2.45. E+04	1.9E+14	328	399	3.50	- 76.35	4.26	4.00. E+02	2.42. E+04	2.29. E+07
1	1	1	2	54	1.33. E+00	7.17. E+01	3.78. E+04	3E+14	328	399	4.72	- 74.20	4.54	4.76. E+02	3.78. E+04	5.20. E+07
1	1	1	2	423	1.42. E+00	5.99. E+02	2.96. E+05	2.3E+15	328	399	10.93	- 74.78	4.99	5.52. E+02	2.96. E+05	1.54. E+08
1	1	1	2	210	1.33. E+00	2.79. E+02	1.47. E+05	1.2E+15	328	399	2.94	- 75.30	5.24	6.28. E+02	1.47. E+05	6.31. E+07
1	1	1	2	104	1.35. E+00	1.40. E+02	7.28. E+04	5.7E+14	328	399	4.51	- 75.03	4.99	7.03. E+02	7.25. E+04	1.05. E+08
10	10	9	9	0	4.30.E-01	0.00. E+00	0.00. E+00	0	110	312	10.45	- 75.51	4.14	7.79. E+02	4.14. E+03	1.48. E+08
1	1	1	2	1260	1.42. E+00	1.79. E+03	8.82. E+05	6.9E+15	328	399	11.02	- 74.84	4.26	8.55. E+02	8.82. E+05	4.34. E+08
1	1	1	2	772	1.38. E+00	1.07. E+03	5.40. E+05	4.2E+15	328	399	6.40	- 75.43	4.14	9.31. E+02	5.40. E+05	2.81. E+08
1	1	1	2	357	1.33. E+00	4.74. E+02	2.50. E+05	2E+15	328	399	4.64	- 74.12	4.63	1.01. E+03	2.50. E+05	1.28. E+08
1	1	1	2	157	1.34. E+00	2.10. E+02	1.10. E+05	8.6E+14	328	399	3.46	- 76.51	6.19	1.08. E+03	1.10. E+05	3.82. E+07
1	1	1	2	125	1.33. E+00	1.66. E+02	8.75. E+04	6.9E+14	328	399	4.54	- 74.87	5.4	1.16. E+03	8.75. E+04	5.14. E+07
1	1	1	2	710	1.42. E+00	1.01. E+03	4.97. E+05	3.9E+15	328	399	10.93	- 74.78	4.14	1.24. E+03	4.97. E+05	1.15. E+08
1	1	1	2	1069	1.38. E+00	1.48. E+03	7.48. E+05	5.9E+15	328	399	6.36	- 75.49	4.14	1.36. E+03	7.49. E+05	3.10. E+08
1	1	1	2	1173	1.37. E+00	1.60. E+03	8.21. E+05	6.4E+15	328	399	6.20	- 75.56	4.14	1.36. E+03	8.21. E+05	3.46. E+08
1	1	1	2	1910	1.34. E+00	2.56. E+03	1.34. E+06	1.1E+16	328	399	3.59	- 76.59	6.19	1.36. E+03	1.34. E+06	3.82. E+08
1	1	1	2	14755	1.36. E+00	2.01. E+04	1.03. E+07	8.1E+16	328	399	6.76	- 73.10	4.14	3.00. E+05	1.03. E+07	9.07. E+10
1	1	1	2	614483	1.40. E+00	8.61. E+05	4.30. E+08	3.4E+18	328	399	10.47	- 74.75	4.99	3.00. E+05	4.30. E+08	9.07. E+10
1	1	1	2	1705964	1.40. E+00	2.38. E+06	1.19. E+09	9.4E+18	328	399	9.59	- 73.63	6.19	3.00. E+05	1.19. E+09	8.08. E+10
1	1	1	2	155112	1.40. E+00	2.17. E+05	1.09. E+08	8.5E+17	328	399	9.79	- 73.72	4.54	3.00. E+05	1.09. E+08	7.09. E+10
1	1	1	2	532286	1.33. E+00	7.09. E+05	3.73. E+08	2.9E+18	328	399	3.86	- 67.91	4.14	2.50. E+05	3.73. E+08	6.10. E+10
1	1	1	2	271368	1.36. E+00	3.69. E+05	1.90. E+08	1.5E+18	328	399	6.76	- 73.10	5.28	3.00. E+05	1.90. E+08	9.07. E+10
1	1	1	2	91825	1.40. E+00	1.29. E+05	6.43. E+07	5.1E+17	328	399	10.47	- 74.75	5.81	3.00. E+05	6.43. E+07	9.07. E+10
1	1	1	2	339132	1.40. E+00	4.74. E+05	2.37. E+08	1.9E+18	328	399	9.59	- 73.63	5.85	2.95. E+05	2.37. E+08	7.29. E+10

#Paneles serie	#Máx paneles serie	#Paneles Paralelo	#Máx Paneles Paralelo	#Inversor	Área sombra panel	Área del terreno	Potencia generada	Precio proyecto	Código Panel	Código Inversor	Latitud	Longitud	Radiación	Área disponible	Potencia necesaria	Dinero disponible
1	1	1	2	532755	1.40. E+00	7.45. E+05	3.73. E+08	2.9E+18	328	399	9.79	- 73.72	5.85	3.00. E+05	3.73. E+08	9.07. E+10
1	1	1	2	585491	1.33. E+00	7.80. E+05	4.10. E+08	3.2E+18	328	399	3.86	- 67.91	5.21	3.00. E+05	4.10. E+08	8.67. E+10
1	1	1	2	18818	1.36. E+00	2.56. E+04	1.32. E+07	1E+17	328	399	6.76	- 73.10	5.28	3.00. E+05	1.32. E+07	6.69. E+10
1	1	1	2	507034	1.40. E+00	7.11. E+05	3.55. E+08	2.8E+18	328	399	10.47	- 74.75	5.81	3.00. E+05	3.55. E+08	8.97. E+10
1	1	1	2	752143	1.40. E+00	1.05. E+06	5.27. E+08	4.1E+18	328	399	9.59	- 73.63	5.85	3.00. E+05	5.27. E+08	7.88. E+10
1	1	1	2	339132	1.40. E+00	4.74. E+05	2.37. E+08	1.9E+18	328	399	9.79	- 73.72	5.85	3.00. E+05	2.37. E+08	8.08. E+10
1	1	1	2	50455	1.33. E+00	6.72. E+04	3.53. E+07	2.8E+17	328	399	3.86	- 67.91	5.21	2.80. E+05	3.53. E+07	9.07. E+10
1	1	1	2	348838	1.36. E+00	4.74. E+05	2.44. E+08	1.9E+18	328	399	6.76	- 73.10	5.28	3.00. E+05	2.44. E+08	7.68. E+10
1	1	1	2	529112	1.40. E+00	7.42. E+05	3.70. E+08	2.9E+18	328	399	10.47	- 74.75	5.81	3.00. E+05	3.70. E+08	8.47. E+10
1	1	1	2	440571	1.40. E+00	6.16. E+05	3.08. E+08	2.4E+18	328	399	9.59	- 73.63	5.85	3.00. E+05	3.08. E+08	6.69. E+10
1	1	1	2	339132	1.40. E+00	4.74. E+05	2.37. E+08	1.9E+18	328	399	9.79	- 73.72	5.85	2.70. E+05	2.37. E+08	8.87. E+10
1	1	1	2	727222	1.33. E+00	9.68. E+05	5.09. E+08	4E+18	328	399	3.86	- 67.91	5.21	2.80. E+05	5.09. E+08	8.57. E+10
1	1	1	2	150826	1.36. E+00	2.05. E+05	1.06. E+08	8.3E+17	328	399	6.76	- 73.10	5.28	3.00. E+05	1.06. E+08	8.77. E+10
1	1	1	2	298608	1.40. E+00	4.19. E+05	2.09. E+08	1.6E+18	328	399	10.47	- 74.75	5.81	2.50. E+05	2.09. E+08	8.18. E+10
1	1	1	2	110359	1.40. E+00	1.54. E+05	7.73. E+07	6.1E+17	328	399	9.59	- 73.63	5.85	3.00. E+05	7.73. E+07	7.09. E+10
1	1	1	2	38751	1.40. E+00	5.42. E+04	2.71. E+07	2.1E+17	328	399	9.79	- 73.72	5.85	2.75. E+05	2.71. E+07	8.47. E+10
1	1	1	2	50455	1.33. E+00	6.72. E+04	3.53. E+07	2.8E+17	328	399	3.86	- 67.91	5.21	3.00. E+05	3.53. E+07	6.99. E+10
1	1	1	2	593358	1.36. E+00	8.07. E+05	4.15. E+08	3.3E+18	328	399	6.76	- 73.10	5.28	3.00. E+05	4.15. E+08	6.99. E+10
1	1	1	2	198503	1.40. E+00	2.78. E+05	1.39. E+08	1.1E+18	328	399	10.47	- 74.75	5.81	2.50. E+05	1.39. E+08	7.88. E+10
1	1	1	2	128261	1.40. E+00	1.79. E+05	8.98. E+07	7.1E+17	328	399	9.59	- 73.63	5.85	3.00. E+05	8.98. E+07	7.88. E+10
1	1	1	2	56653	1.40. E+00	7.93. E+04	3.97. E+07	3.1E+17	328	399	9.79	- 73.72	5.85	3.00. E+05	3.97. E+07	9.07. E+10
1	1	1	2	474471	1.33. E+00	6.32. E+05	3.32. E+08	2.6E+18	328	399	3.86	- 67.91	5.21	2.65. E+05	3.32. E+08	9.07. E+10
1	1	1	2	523919	1.36. E+00	7.12. E+05	3.67. E+08	2.9E+18	328	399	6.76	- 73.10	5.28	2.80. E+05	3.67. E+08	8.97. E+10
1	1	1	2	383853	1.40. E+00	5.38. E+05	2.69. E+08	2.1E+18	328	399	10.47	- 74.75	5.81	3.00. E+05	2.69. E+08	8.87. E+10
1	1	1	2	56653	1.40. E+00	7.92. E+04	3.97. E+07	3.1E+17	328	399	9.59	- 73.63	5.85	3.00. E+05	3.97. E+07	9.07. E+10
1	1	1	2	20849	1.40. E+00	2.92. E+04	1.46. E+07	1.1E+17	328	399	9.79	- 73.72	5.85	3.00. E+05	1.46. E+07	8.28. E+10
1	1	1	2	82342	1.33. E+00	1.10. E+05	5.76. E+07	4.5E+17	328	399	3.86	- 67.91	5.21	3.00. E+05	5.76. E+07	8.67. E+10
1	1	1	2	736993	1.36. E+00	1.00. E+06	5.16. E+08	4.1E+18	328	399	6.76	- 73.10	5.28	2.65. E+05	5.16. E+08	7.68. E+10
1	1	1	2	162944	1.40. E+00	2.28. E+05	1.14. E+08	9E+17	328	399	10.47	- 74.75	5.81	3.00. E+05	1.14. E+08	9.07. E+10
1	1	1	2	90803	1.40. E+00	1.27. E+05	6.36. E+07	5E+17	328	399	9.59	- 73.63	5.85	3.00. E+05	6.36. E+07	9.07. E+10

#Paneles serie	#Máx paneles serie	#Paneles Paralelo	#Máx Paneles Paralelo	#Inversor	Área sombra panel	Área del terreno	Potencia generada	Precio proyecto	Código Panel	Código Inversor	Latitud	Longitud	Radiación	Área disponible	Potencia necesaria	Dinero disponible
1	1	1	2	167108	1.40. E+00	2.34. E+05	1.17. E+08	9.2E+17	328	399	9.79	- 73.72	5.85	3.00. E+05	1.17. E+08	8.47. E+10
1	1	1	2	18568	1.33. E+00	2.47. E+04	1.30. E+07	1E+17	328	399	3.86	- 67.91	5.21	3.00. E+05	1.30. E+07	9.07. E+10
1	1	1	2	99606	1.36. E+00	1.35. E+05	6.97. E+07	5.5E+17	328	399	6.76	- 73.10	5.28	3.00. E+05	6.97. E+07	9.07. E+10
1	1	1	2	93336	1.40. E+00	1.31. E+05	6.53. E+07	5.1E+17	328	399	10.47	- 74.75	5.81	3.00. E+05	6.53. E+07	8.18. E+10
1	1	1	2	199869	1.40. E+00	2.79. E+05	1.40. E+08	1.1E+18	328	399	9.59	- 73.63	5.85	3.00. E+05	1.40. E+08	8.77. E+10
1	1	1	2	605884	1.40. E+00	8.48. E+05	4.24. E+08	3.3E+18	328	399	9.79	- 73.72	5.85	3.00. E+05	4.24. E+08	9.07. E+10
1	1	1	2	409342	1.33. E+00	5.45. E+05	2.87. E+08	2.3E+18	328	399	3.86	- 67.91	5.21	2.70. E+05	2.87. E+08	8.77. E+10
1	1	1	2	180395	1.36. E+00	2.45. E+05	1.26. E+08	9.9E+17	328	399	6.76	- 73.10	5.28	2.90. E+05	1.26. E+08	7.58. E+10
1	1	1	2	109605	1.40. E+00	1.54. E+05	7.67. E+07	6E+17	328	399	10.47	- 74.75	5.81	3.00. E+05	7.67. E+07	8.28. E+10
1	1	1	2	1453119	1.40. E+00	2.03. E+06	1.02. E+09	8E+18	328	399	9.59	- 73.63	5.85	2.60. E+05	1.02. E+09	7.88. E+10
1	1	1	2	498273	1.40. E+00	6.97. E+05	3.49. E+08	2.7E+18	328	399	9.79	- 73.72	5.85	3.00. E+05	3.49. E+08	9.07. E+10
1	1	1	2	409342	1.33. E+00	5.45. E+05	2.87. E+08	2.3E+18	328	399	3.86	- 67.91	5.21	3.00. E+05	2.87. E+08	8.18. E+10

Anexo C : **Explicación del script del proceso**

En este script se muestran los pasos que se siguieron para obtener los resultados, comenzando procesamiento de los datos, pasando por la selección de los datos, terminando en el proceso que se hizo para seleccionar el modelo.

Procesamiento de los datos

Visualización de sitios de los proyectos

Según la longitud y latitud que tengan los proyectos de las bases de datos, se van a graficar con un símbolo, el cual puede variar la forma según su clase.

```
rng("default")
hoob=1;
if isfile('codigoProcesoF.mlx')==0
    if hoob==0
        patch="d:\tesis\AprendizajePorRefuerzo";
    else
        helpdlg('Seleccione la ruta donde guardó el archio','¿Donde guardaste el trabajo?');
        [file,patch]=uigetfile('*.mlx');
    end
    cd ..\..\..\..
    cd (patch)
end

% sitioActual="d:\tesis\AprendizajePorRefuerzo";
% if pwd ~=sitioActual
%     cd
%     cd d:\tesis\AprendizajePorRefuerzo
% end
close all
clear
clc
%Base del algortimo
%Carga, clasificacion, y señalizacón de los datos
%La columna discriminante
cCl=19;
%Dejó al azar la actualizacion de los datos Actualizar=1 No actualizar=0
r1='Sí';
r2='No';
titleMessage='Bases de datos';
actualizar=questdlg('¿Desea actualizar los datos?',titleMessage,r1,r2,r1);
switch actualizar
    case "Sí"
        actualizar=1;
    case 'No'
        actualizar=0;
end
```

```
%El llamado de las bases de datos
if exist('bd','var')==0 && exist('radiacion','var')==0 &&...
        exist('bdT','var')==0  && exist('costos','var')==0||actualizar==1
    [bd,radiacion,bdT,paneles,eEquipos,inverter,costos]=baseDeDatos(actualizar);
end
%_____________________________________
%Completando los datos de la base de datos

if (string(bdT.Properties.VariableNames(end))=="Costos")~=1
    [xT,~]=size(bdT);
    [xcT,~]=size(costos);
    %Se completa la tabla para que no exista problemas de compatibilidad
    costos=table([costos;zeros((xT-xcT),1)],'VariableNames',"Costos");
    bdT=[bdT,costos];

    %_____________________________________
    %Genera precios aleatorios
        %Paneles

rpP=table(randomPrices(paneles.Material,eEquipos(:,2:3)),'VariableNames',"PreciosGeneradosPan
el");
    paneles=[paneles,rpP];
        %Inversores
        %Genera los precios de los invertores

rpI=table(randomPrices(inverter.Source,[eEquipos(:,4),eEquipos(:,4)]),'VariableNames',"Precio
sGeneradosInverter");
            %Genera la cantidad de mppt de los inversores
    [filasI,columnasI]=size(inverter.Source);
            %Genera la cantidad de mppt
    mpptGenerado=table(randi([1,4],filasI,columnasI),'VariableNames',"CantidadMppt");
            %Une todos los datos
    inverter=[inverter,rpI,mpptGenerado];
    if exist("pricesGeneratedModulex.xlsx","file")~=0 &&...
            exist("pricesGeneratedInverter.xlsx","file")~=0
    delete("pricesGeneratedModule.xlsx");
    delete("pricesGeneratedInverter.xlsx");

    end
    writetable(rpP,'pricesGeneratedModule.xlsx')
    writetable(rpI,'pricesGeneratedInverter.xlsx')

    %_____________________________________

end

%Los codigos de color
codeColor='ymcrgbk';
[~,y0]=size(codeColor);
```

```
 %Los codigos de formas
 marcador='oxsdvph^><.';
 [~,y1]=size(marcador);
 %El llamado a la funcion clasificador
 [conjunto,sector]=clasificador(bdT,cCl);
 [filSector,~]=size(sector);
 %Se pone estos nombres temporales, para reservar un espacio en la memoria
 nombres=categorical({'a','b'});
 %Datos usados para crear la matriz
 [~,coord]=table2matrix(bdT);
 dUsed=num2str(coord);
 nU=bdT.Properties.VariableNames(coord);
 %El creador de conjuntos y subconjuntos
 for k =1:filSector
    v = genvarname('subConjunto', who);
    z = genvarname('estadisticas', who);
    y = genvarname('subConjuntoFull',who);
    w = genvarname('matrix',who);
        if k~=1
            eval([v ' = conjunto(sum(sector(1:k-1))+1:sum(sector(1:k)),:);']);
            eval([z '= estatisticsData(subConjunto',int2str(k-1),');']);
            eval([y '=autocompletar(subConjunto',int2str(k-1),',estadisticas',int2str(k-
1),');']);

            eval(['geoplot(subConjunto',int2str(k-1),'.Latitud,subConjunto',int2str(k-
1),'.Longitud,"',(marcador(randi([1,y1]))),...
                ,'","Color","',(codeColor(randi([1,y0]))),'","LineWidth",',int2str(randi([5
10])/10),');']);
            eval ([w '=table2array(subConjuntoFull',int2str(k-1),'(:,[',dUsed,']));']);
            nombres(k)=eval(['table2array(subConjunto',int2str(k-1),'(1,',int2str(cCl-
1),'));']);
        else
            eval([v ' = conjunto(1:sector(k),:);']);
            eval([z '= estatisticsData(subConjunto);']);
            eval([y '=autocompletar(subConjunto,estadisticas);']);

            tituloUse=eval(['subConjunto.Properties.VariableNames(',int2str(cCl-1),');']);
            figure('Name',string(tituloUse))

eval(['geoplot(subConjunto.Latitud,subConjunto.Longitud,"',(marcador(randi([1,y1]))),'","Colo
r","',(codeColor(randi([1,y0]))),'","LineWidth",',int2str(randi([5 10])/10),');']);
            eval ([w '=table2array(subConjuntoFull(:,[',dUsed,']));']);
            nombres=eval(['table2array(subConjunto(1,',int2str(cCl-1),'));']);
            hold on
        end
 end
 %Genera nombres a los productos
 legend(string(nombres));
 title (tituloUse)
```

```
 geoplot(bdT.latitud,bdT.longitud,'k+','LineWidth',1,'DisplayName','Puntos de datos
ambientales tomados')
 legend();
 geobasemap colorterrain
```

En el grafico anterior, se observa la localización, esto sirve para saber más adelante para estimar cual es la cantidad real de potencia producida según los módulos/paneles seleccionados.

Visualización de datos

Para poder visualizar los datos primero se cargan los datos numéricos de la base de datos.

```
 %Para la eliminación de outliers
 memoria=[12:15,17,22:27,59];
 %memoria=[12:15,17,22:59];
 namesUsed=["% de energía suplida","Capacidad [kWp]","Estimación de energía
[kWh/year]","Prevención de emisión de CO2","Área [m^2]",...)
 "Presión media [kPa]","Temperatura media [°C]","Velocidad del viento media [m/s]",...
 "kt media","Insolación en cielo despejado[kWh/m^2/day]","Toda la
insolación[kWh/m^2/day]","Costos [$]"];
 dataSC=flipud(table2array(subConjunto(:,memoria)));
 dataGen=flipud(table2array(subConjunto1(:,memoria)));
 for i=1:2
     switch i
         case 1
             tComplemento=" sin limpieza de datos atípicos";
         case 2
             tComplemento=" con limpieza de datos atípicos";
     end
     figure
     boxAutoConsumo=boxplot(normalize(dataSC));
     xticklabels(namesUsed);
     xtickangle(45)
     grid
     title(["Diagrama de cajas y bigotes de autoconsumo",tComplemento])
     dataSC(isoutlier(dataSC))=NaN;
     figure
     boxGen=boxplot(normalize(dataGen));
     xticklabels(namesUsed);
     xtickangle(45)
     grid
     dataGen(isoutlier(dataGen))=[NaN];
     title(["Diagrama de cajas y bigotes de generación",tComplemento])
 end
```

Siendo el primer gráfico, el de los datos sin ningún tipo de tratamiento, el seguido por una limpieza de datos atípicos.

Imputación de datos

Se hizo un Imputación de datos usando interpolación lineal entre los puntos más cercanos al dato faltante.

```
dataSC=fillmissing(dataSC,'linear',1,'EndValues','nearest');
dataGen=fillmissing(dataGen,'linear',1,'EndValues','nearest');
```

Clasificación

Selección de características/variables

Se seleccionaron las 3 características/variables más relevantes para clasificación, se usó el método de fscnca. Se lee la cantidad de filas en autoconsumo y en generación, después se fichan las clases que hay, a continuación, se seleccionan las características más relevantes mediante el método de FSCNA. Al hacerse la prueba del cambio del Lambda se observó que no existió cambio alguno en el error medio cuadrado.

```
tituloCalculoLambda="Seleccionando el lambda optimo";
scythe=5; %<---------------------------------------
rng("default")
%Reconoce la cnatidad de datos
[rowSC,~]=size(dataSC);
[rowGen,colGen]=size(dataGen);
%Concatena los datos para clasificar
dataComplete=[dataSC;dataGen];
dataComplete1=dataComplete;
dataComplete=dataComplete(:,[2:5,end]);
%Nombres
namesUsed1=namesUsed;
namesUsed=namesUsed(:,[2:5,end]);
%Concatena los datos de respuesta
fichasU=[zeros(rowSC,1);ones(rowGen,1)];
fichasU1=fichasU;
%Selecciona las variables más relevantes
cvp=cvpartition(fichasU,'KFold',10);
numValidTest=cvp.NumTestSets;
n=length(fichasU);
lambdaVal=linspace(1,5,50)/n;
lossvals=zeros(length(lambdaVal),numValidTest);
for i=1:length(lambdaVal)
    for j=1:numValidTest
        dataTrain=dataComplete(cvp.training(j),:);
        fichaTrain=fichasU(cvp.training(j),:);
        dataTest=dataComplete(cvp.test(j),:);
        fichasTest=fichasU(cvp.test(j),:);
        mdl1=fscnca(dataTrain,fichaTrain,'FitMethod',"exact",'Solver',"sgd",'Lambda',...
            lambdaVal(i),'IterationLimit',30,'GradientTolerance',1e-4,'Standardize',true);
        lossvals(i,j)=loss(mdl1,dataTest,fichasTest,'LossFunction','classiferror');
    end
end
lambdaMemoria=[lambdaVal',mean(lossvals,2)];
figure
plot(lambdaMemoria(:,1),lambdaMemoria(:,2),'ro-')
```

```
 [~,idx]=min(lambdaMemoria(:,2));
 lambdaUsing=lambdaMemoria(idx,1);
 xlabel("Lambda");
 ylabel("MSE");
 title(tituloCalculoLambda);
 hold on
 plot(lambdaMemoria(idx,1),lambdaMemoria(idx,2),'ro','MarkerFaceColor','g');
 grid
 text(lambdaMemoria(idx,1),lambdaMemoria(idx),['
\lambda:',num2str(lambdaMemoria(idx,1)),newline,...
         '  MSE:',num2str(lambdaMemoria(idx,2))]);
 legend (["Error","Punto con menor error"])
 %Ordena las variables más relevantes
 mdl = fscnca(dataComplete,fichasU,'FitMethod','exact','Solver','sgd',...
     'Lambda',lambdaUsing,'Standardize',true,'Verbose',1);
 %mdl=fscnca(dataComplete,fichasU,'Solver','sgd','Verbose',1);
 selectorCP=ordenar([mdl.FeatureWeights,(1:length(mdl.FeatureWeights))'],1,'mayor');
 %Muestra las variables más relevantes
 figure()
 plot(mdl.FeatureWeights,'ro')
 grid
 hold on
 plot(linspace(0,colGen,colGen),repmat(mdl.FeatureWeights(selectorCP(scythe,2)),colGen),'b-
');
 xticklabels(namesUsed(1:end));
 xtickangle(45)
 xticks(1:12)
 %ylim([0 mdl.FeatureWeights(selectorCP(2,2))])
 title('Pesos de las variables')
 ylabel('Peso')
 T=table(namesUsed(selectorCP(:,2))',selectorCP(:,1),'VariableNames',{'Variable','Peso'});
```

La grafica se acerca a las dos siguientes variables después del costo, ya que el peso del costo es un valor muy grande con respecto a las demás variables. Las características que tienen poco peso, se deben a que aportan poco a la clasificación de las dos clases existentes. Las características/variables más relevantes fueron:

1. Costo [$].
2. Estimación de energía [kWh/año].
3. Área [m^2].

Diagrama de dispersión de las características seleccionadas

Para ver si los datos se pueden clasificar, se observa mediante un diagrama de dispersión.

```
 columnaUsing=2;
 figure
 plot3(dataSC(:,selectorCP(1,columnaUsing)),dataSC(:,selectorCP(2,columnaUsing)),dataSC(:,sel
ectorCP(3,columnaUsing)),'bo','MarkerFaceColor','b','DisplayName','Autoconsumo');
 hold on
```

```
 grid
 plot3(dataGen(:,selectorCP(1,columnaUsing)),dataGen(:,selectorCP(2,columnaUsing)),dataGen(:,
selectorCP(3,columnaUsing)),'ro','MarkerFaceColor','r','DisplayName','Generación');
 xlabel(namesUsed(selectorCP(1,columnaUsing)));
 ylabel(namesUsed(selectorCP(2,columnaUsing)));
 zlabel(namesUsed(selectorCP(3,columnaUsing)));
 title("Grafico de dispersión");
 legend();
 legend('Location','northwest');
 view([32 45])
```

El resultado es una gráfica de dispersión, en el cual se distinguen las clases de Autoconsumo (Azul) y la clase de Generación (Rojo), se observa que, de la clase de Autoconsumo, se concentran en los puntos más cercanos a cero, mostrando que la los costos, la estimación de energía anual y del área del autoconsumo son considerablemente menores a la generación.

Generación de datos a partir de ADASYN/SMOTE

Para la correcta discriminación de clases, se hizo un balance de datos, entre la clase minoritaria es decir Generación y la clase mayoritaria, es decir Autoconsumo. Para ello se hizo la siguiente sección que envía los datos a la función de ADASYN.

Donde las variables:

1. Dth = Umbral máximo de desbalance. Puede estar de cero a uno.
2. Beta = Balance de clases. Puede estar de cero a uno, donde cero es no generar datos y uno es balance total de datos.
3. Kn = Cantidad de vecinos cercanos tomados para generar datos. Puede ser la cantidad que se desee, es recomendable hacer pruebas.

```
 datosGenerados=ones(rowSC-rowGen,scythe);
 dth=1;
 beta=1;
 kn=10;
 for i=1:scythe
     switch namesUsed(selectorCP(i,2))
         case namesUsed(1)
             pIn=2;
         otherwise
             pIn=1;

     end

chest=ADASYNPropio(dataGen(:,[pIn,selectorCP(i,columnaUsing)]),dataSC(:,[pIn,selectorCP(i,col
umnaUsing)]),dth,beta,kn);
     if i==1
         datosG(:,[1,i])=chest;
     else
         datosG(:,i)=chest(:,2);
     end
```

```
 end

 dataGenNew=[dataGen(:,selectorCP(1:scythe,columnaUsing));datosG];
 dataCompleteUsing=[dataSC(:,selectorCP(1:scythe,columnaUsing));dataGenNew];
 fichasCompleteUsing=[ones(rowSC,1);-ones(rowSC,1)];

 figure
 plot3(dataSC(:,selectorCP(1,columnaUsing)),dataSC(:,selectorCP(2,columnaUsing)),dataSC(:,sel
ectorCP(3,columnaUsing)),'bo','MarkerFaceColor','b','DisplayName','Autoconsumo');
 hold on
 grid
 plot3(dataGenNew(:,1),dataGenNew(:,2),dataGenNew(:,3),'ro','MarkerFaceColor','r','DisplayNam
e','Generación');
 plot3(datosG(:,1),datosG(:,2),datosG(:,3),'k.','MarkerFaceColor','k','DisplayName','Generaci
ón sinteticos');
 xlabel(namesUsed(selectorCP(1,columnaUsing)));
 ylabel(namesUsed(selectorCP(2,columnaUsing)));
 zlabel(namesUsed(selectorCP(3,columnaUsing)));
 title("Grafico de dispersión");
 legend('Location','northwest');
 view([32 45])
 dataComplete=dataComplete1;
 fichasU=fichasU1;
 namesUsed=namesUsed1;
```

Donde los puntos rojos con un punto negro en el centro, son datos sintéticos y válidos.

Predicción

En esta sección se hizo una herramienta auxiliar que haga diseños fotovoltaicos según los datos de los proyectos fotovoltaicos (se incluyó los datos sintéticos). Después se hizo una selección de variables que mejor describían el comportamiento de los datos, esto para poder disminuir el coste computacional en el momento de hacer la red neuronal.

Herramienta auxiliar

Se hizo esta herramienta para simular los datos hechos por alguien con el conocimiento para poder hacer un dimensionamiento de una planta fotovoltaica, seleccionando únicamente los paneles/módulos y los inversores.

Se toman las variables que van a ser leídas por la función principal y se cerciora que no exista ningún archivo que contenga los dimensionamientos. Dando como resultado:

1. Costo del proyecto usando los módulos sugeridos.
2. Cantidad de módulos en serie.
3. Cantidad de módulos máxima en serie.
4. Cantidad de módulos en paralelo.
5. Cantidad de módulos máxima en paralelo.
6. Ángulo del panel.
7. Potencia generada.
8. Código del panel.
9. Cantidad de inversores.

10. Código del inversor.
11. Número del proyecto.

```
latitudes=[subConjunto.Latitud;subConjunto1.Latitud;repmat(subConjunto1.Latitud,9,1)];
longitudes=[subConjunto.Longitud;subConjunto1.Longitud;repmat(subConjunto1.Longitud,9,1)];
costo=dataCompleteUsing(:,namesUsed(selectorCP(:,2))'=="Costos [$]");
potencia=dataCompleteUsing(:,namesUsed(selectorCP(:,2))'=="Capacidad [kWp]");
area=dataCompleteUsing(:,namesUsed(selectorCP(:,2))'=="Área [m^2]");
porcentajeSuplir=dataCompleteUsing(:,namesUsed(selectorCP(:,2))'=="% de energía suplida");
radiaciones=[bdT.radAllMediana;repmat(subConjunto1.radAllMediana,9,1)];
%Para el diseño
if isfile("DesFinal.xlsx")==0

    [AdesAcept,AtableDesAcept]=autodesign(area,potencia,costo,latitudes,longitudes);
    writetable(AtableDesAcept,"DesFinal.xlsx")
%     namesUsed=["% de energía suplida","Capacidad [kWp]","Estimación de energía
[kWh/year]","Prevención de emisión de CO2","Área [m^2]",...)
% "Presión media [kPa]","Temperatura media [°C]","Velocidad del viento media [m/s]",...
% "kt media","Insolación en cielo despejado[kWh/m^2/day]","Toda la
insolación[kWh/m^2/day]","Costos [$]"];
else
    if exist("AtableDesAcept","var")==0 || exist("AdesAcept","var")==0
        AtableDesAcept=readtable("DesFinal.xlsx");
        AdesAcept=table2array(AtableDesAcept);
    end
end
```

Selección de características/variables para predicción

Se seleccionan las variables que mejor describan el comportamiento con respecto a los valores de interés, se usó FSRNCA para encontrarlas. Para

Se hizo una prueba con el valor de regulación lambda (λ), ella influye en el desempeño del modelo de regresión que tiene la función, para seleccionar la lambda optimo, se evaluó el modelo en distintos puntos de lambda.

Se dividen los datos en 2 grupos, uno de validación y uno de entrenamiento, el grupo de datos de validación se encarga de decir que tanto se ajusta el modelo a una serie de datos con la que no fue entrenado.

```
rng("default")
%Para los proyectos
%Selecciona los proyectos que son validos
puntos=(AtableDesAcept.CodigoDelpanel~=0);
%Hace una tabla con cada uno de los componentes
%Para los proyectos
design_proyectos=AtableDesAcept(puntos,1:end-1); %Completos
respuestas_proyectos=design_proyectos(:,[2,3,4,5,6,12]); %Repuestas
data_proyectos=design_proyectos(:,[1,7,8,9]);%Los datos de los proyectos
%Los espacios de los datos
%[cantidad_respuestas,filas_respuestas]=size(respuestas_proyectos); %
%[cantidad_datos,columna_datos]=height(data_proyectos); %
```

```
 %Para el sitio
 sitio_proyectos=table(latitudes(puntos),longitudes(puntos),radiaciones(puntos),'VariableName
s',{'Latitud','Longitud','HoraSolarPico'});
 %Para los paneles
 paneles_proyectos=paneles(AtableDesAcept.CodigoDelpanel(puntos),[3,7:12,44]);
 %Para los inversores
 inversor_proyectos=inverter(AtableDesAcept.CodigoDelInversor(puntos),[4,6:9,14:18]);
 %Datos concatenados
 dataConcat_0=[design_proyectos,sitio_proyectos,paneles_proyectos,inversor_proyectos];
%Todo + respuestas
 %Datos concatenados sin respuestas
 dataConcat_1=[data_proyectos,sitio_proyectos,paneles_proyectos,inversor_proyectos];
%Todo - respuestas
 %Coeficiente de correlacion
 corelacion=corrcoef(table2array(dataConcat_0));
 naq0=dataConcat_0.Properties.VariableNames;
 naq1=dataConcat_1.Properties.VariableNames;
 %_____________________________________________________________________________________
_________%
 %Para seleccionar las variables relevantes para la regresión
 %Se selecciona la cantida de datos
 cData=length(respuestas_proyectos.CantidadModulosSerie);
 %30% de los datos son para el trabajo
 porcentajePaquetes=randi([20,30])/100;
 cvp=cvpartition(respuestas_proyectos.CantidadModulosSerie,'holdout',porcentajePaquetes);
 %Entrenamiento
 respuestas_train=table2array(respuestas_proyectos(cvp.training,:));
 datos_train=table2array(dataConcat_1(cvp.training,:));
 %Test
 respuestas_test=table2array(respuestas_proyectos(cvp.test,:));
 datos_test=table2array(dataConcat_1(cvp.test,:));
 %La cantidad de datos seleecionados
 sc=3; %<--------------------------------------
 %Mide cuantos datos hay
 [~,columnasDatos]=size(respuestas_test);
 [~,caracteristicasDatos]=size(datos_test);
 %Hace una cantidad de datos en blanco, esto para ahorrar tiempo de
 %procesamiento
 caracteristicasS=zeros(sc,2,columnasDatos);
 bestLL=zeros(columnasDatos,2);
 %Se guarda espacio para los datos importantes
 dataImportant_proyectos=zeros(cData,sc,columnasDatos);
 dataImportant_names=cell(columnasDatos,sc);
 for posicion=1:columnasDatos
     n=length(respuestas_train(:,posicion));
     cvp=cvpartition(length(respuestas_train(:,posicion)),'kfold',10);
     validSet=cvp.NumTestSets;
     lambdaVal=linspace(0,60,60)*std(respuestas_train(:,posicion))/n;
     lossvals=zeros(length(lambdaVal),validSet);
```

```
    for i=1:length(lambdaVal)
        for k=1:validSet
            newDatos_train=datos_train(cvp.training(k),:);
            newRespuestas_train=respuestas_train(cvp.training(k),posicion);

            newDatos_Val=datos_train(cvp.test(k),:);
            newRespuestas_Val=respuestas_train(cvp.test(k),posicion);

            nca=fsrnca(newDatos_train,newRespuestas_train,'FitMethod','exact',...
                'Solver','minibatch-lbfgs','lambda',lambdaVal(i),'GradientTolerance',1e-
4,'IterationLimit',30);
            lossvals(i,k)=loss(nca,newDatos_Val,newRespuestas_Val);
        end
    end
    meanloss=mean(lossvals,2);
    figure ('Name',['Caracteristicas para:
',char(respuestas_proyectos.Properties.VariableNames(posicion))])
    %subplot(2,1,1);
    plot(lambdaVal,meanloss,'r-o')
    title(tituloCalculoLambda)
    xlabel("Lambda")
    ylabel("MSE")
    hold on

    [~,idx]=min(meanloss);
    bestLL(posicion,:)=[lambdaVal(idx),meanloss(idx)];

    plot(lambdaVal(idx),meanloss(idx),'ro','MarkerFaceColor','g')

    text(lambdaVal(idx),meanloss(idx),['  \lambda:',num2str(lambdaVal(idx)),newline,...
        '  MSE:',num2str(meanloss(idx))]);
    grid on
    legend (["Error","Punto con menor error"])
    figure ('Name',['Caracteristicas para:
',char(respuestas_proyectos.Properties.VariableNames(posicion))])


ncaUse=fsrnca(datos_train,respuestas_train(:,posicion),'FitMethod','exact','Solver','lbfgs','
Lambda',lambdaVal(idx));
    %subplot(2,1,2);
    plot(ncaUse.FeatureWeights,'ro')
    title(["Caracteristicas más relevantes usando un lambda de:",lambdaVal(idx)])
    xlabel("Indice de caracteristica")
    ylabel("Peso de la caracteristica")
    xticks(1:caracteristicasDatos)
    xticklabels(naq1(1:end))
    xtickangle(45)

    grid on
```

```
    kT=ordenar([ncaUse.FeatureWeights,(1:caracteristicasDatos)'],1,'mayor');
    pesosRegresion(posicion).a=kT;
    hold on

    plot((0:caracteristicasDatos)',repmat(kT(sc,1),caracteristicasDatos+1,1),'b-')

    caracteristicasS(:,:,posicion)=kT(1:sc,:);

dataImportant_proyectos(:,:,posicion)=table2array(dataConcat_1(:,caracteristicasS(:,2,posicio
n)));
    dataImportant_names(posicion,:)=naq1(caracteristicasS(:,2,posicion));
 end
 dataCompleteUsingNN=table2array(dataConcat_1(:,unique(dataImportant_names)));
 fichasCompleteUsing=table2array(respuestas_proyectos);
```

Código herramienta auxiliar

```
 function enviarConfiguracion=modoAuto(potencia,latitud,longitud)
 %%
 %Ecuaciones a usar
 %El angulo del panel
 betaAngulo=@(latitud) (ceil(3.7+0.69.*abs(latitud)));
 %Maximo paneles en serie
 cMppt=@(vmppt,vmax)(ceil(vmppt./vmax));
 %Distancia de sombras
 distanceShadow=@(l,thetaMod,altitudeSolar,azimuthModule,azimuthSolar)...
    (l.*(cosd(thetaMod)+sind(thetaMod)./tand(altitudeSolar).*cosd(azimuthModule-
azimuthSolar)));
 %Precio proyectos
 priceProyect=@(cantPanel,pricePanel,cantInverter,priceInverter)...
    (cantPanel.*pricePanel.*cantInverter.*priceInverter+cantInverter.*priceInverter);
 %Area total
 areaOcupada=@(areaIndividuo,cantPanel,cantInverter)...
    (areaIndividuo.*cantPanel.*cantInverter);
 %%
 %Carga base de datos
 [panel,inverter]=loadEquip();
 [fP,~]=size(panel);
 [fI,~]=size(inverter);
 %Carga la azimuth
 [~,~,cutterAzimuth]=solarMeteorologico(latitud,longitud,10,14,'no');
 alturaLitte=min(cutterAzimuth(:,2));
 azimuthSolar=max(cutterAzimuth(:,1));
 %%
 %Comienza el dimensionamiento
 config=combvec(1:fP,1:fI);
 cConfig=length(config);
```

```
%Valores por defecto
angulo=betaAngulo(latitud);
if latitud>0
    azimuthPanel=180;
else
    azimuthPanel=0;
end
posPanel=config(1,:);
posInverter=config(2,:);
%Potencia máxima del panel
potMaxPanel=panel.Impo(posPanel).*panel.Vmpo(posPanel);
%Precio panel
pricePanel=panel.PreciosGeneradosPanel(posPanel);
%Precio del inversor
priceInverter=inverter.PrecioGeneradosInverter(posInverter);
%Cantidades de paneles
cMaxS=cMppt(inverter.MPPTHi(posInverter),potMaxPanel);
cMinS=cMppt(inverter.MPPTHi(posInverter),potMaxPanel);
cMaxP=cMppt(inverter.Idcmax(posInverter),panel.Isco(posPanel));
cPanelPot=cMppt(potencia,potMaxPanel);
%El numero de mppts actuales
mpptUse=round(inverter.Vdcmax(posInverter)./inverter.MPPTHi(posInverter));
%Cantidad paneles iniciales
cSerie=cMaxS;
cParalelo=cMaxP;
profe=1;
permiso=0;
%cVer=[];
while true
    cPinicial=cSerie.*cParalelo.*mpptUse;
    %La cantidad de inversores
    %cInv=round(potenciaD./inverter.Paco(posInverter));%
    cInv=round(potencia./(potMaxPanel.*cPinicial));
    cInv(cInv<1)=1;
    %Condiciones
    c1=inverter.Vdcmax(posInverter).*inverter.Idcmax(posInverter)>=cPinicial.*potMaxPanel;
    c2=inverter.Vdcmax(posInverter)>=(panel.Voco(posPanel).*cSerie);
    c3=inverter.Pso(posInverter)<=(potMaxPanel.*cPinicial);
    c4=inverter.MPPTLow(posInverter)<=panel.Vmpo(posPanel).*cSerie;
    c5=inverter.MPPTHi(posInverter)>=panel.Vmpo(posPanel).*cSerie;
    c6=inverter.Idcmax(posInverter)>=panel.Impo(posPanel).*cParalelo;
    c7=cMinS<=cSerie;
    cT=(cPinicial.*cInv>=cPanelPot-permiso).*(cPinicial.*cInv<=cPanelPot+permiso);
    c8=~logical(c1.*c2.*c3.*c4.*c5.*c6.*c7.*cT);
    %Disminuye uno en paralelo
    cParalelo(c8)=cParalelo(c8)-1;
    %La posicion de los que estan en cero%
    selectParalelo=logical(cParalelo==0);
    %Los de paralelos son restaurados
```

```
    cParalelo(selectParalelo)=cMaxP(selectParalelo);
    %Disminuye uno en paralelo
    cSerie(selectParalelo)=cSerie(selectParalelo)-1;
    %Selecciona los de en serie negativos
    selecSerie=logical((cSerie==0));
    %Selecciona y los pone en el maximo
    cSerie(selecSerie)=cMaxS(selecSerie);
    if profe==1
        minimo=sum(selecSerie);
        strike=0;
        mat=0;
    elseif sum(selecSerie)==minimo
        strike=strike+1;
        if strike==10
            resMppt=logical((mpptUse>1).*selecSerie);
            mpptUse(resMppt)=mpptUse(resMppt)-1;
            mat=mat+1;
            strike=0;
            if mat==6
                break;
            end
        end
    elseif minimo>sum(selecSerie)
        minimo=sum(selecSerie);
        strike=0;
    end
    profe=profe+1;
end
%Angulo
anguloExport=repmat(angulo,cConfig,1);
%Relacion
relacion=randi([50,60],cConfig,1)./100;
%Logitudes del panel
longitudPanel=panel.ModuleAream2(posPanel)./relacion;
%Areas del panel
areaU=distanceShadow(longitudPanel,angulo,alturaLitte,azimuthPanel,azimuthSolar).*relacion;
%Area terreno
areaTotal=areaOcupada(areaU,cPinicial,cInv);
%areaTotal=areaTotal+areaTotal.*randi([200,350],cConfig,1)./1000;
%Valor total del proyecto
priceGeneral=priceProyect(cPinicial,pricePanel,cInv,priceInverter);
%Valor total de paneles%
cTotal=cPinicial.*cInv;
%Potencia generada
potGenerada=cTotal.*potMaxPanel;
c9=potGenerada>0;
%variables={'PanelSerie','MaxPanelSerie','PanelPara','MaxPanelPara','numInverter','areaPanel
pShadow','areaTerreno','PotenciaGenerada'...
%    ,'PrecioProyecto','codePanel','codeInverter'};
```

```
 latitudInput=repmat(latitud,length(areaTotal),1);
 longitudInput=repmat(longitud,length(areaTotal),1);
 %Adiciono el azimuth solar usado y la altitud solar usada
 solUsado=repmat([alturaLitte,azimuthSolar],length(areaTotal),1); %<---- Usado

 configuracionSelected=[cSerie,cMaxS,cParalelo,cMaxP,cInv,cTotal,anguloExport,areaU,areaTotal
,potGenerada,priceGeneral,posPanel',posInverter',latitudInput,longitudInput,solUsado];
 enviarConfiguracion=configuracionSelected(logical(~c8.*c9),:);
 %Mirar posible error
 pI=posInverter;
 pP=posPanel;
 %xxxxxxxxx
 load('nnFuller.mat','net');
 answereUse=[cSerie,cMaxS,cParalelo,cMaxP,cInv,anguloExport];
 answereUse=answereUse(logical(~c8.*c9),:);
 input=[inverter.Idcmax(pI),longitudInput,inverter.MPPTHi(pI),inverter.Paco(pI),inverter.Pdco
(pI),...

inverter.Vdcmax(pI),panel.Voco(pP),inverter.acVoltage(pI),repmat([azimuthPanel,potencia],leng
th(cSerie),1)];
 input=input(logical(~c8.*c9),:);

 adapt(net,input',answereUse');
 end
```

Redes neuronales

En esta sección se explora todo el procedimiento que se conllevó seleccionar la arquitectura de las redes neuronales que se trabajan en el proyecto.

Se da un máximo de 30 neuronas para clasificación y 50 para la predicción, debido a que un mayor número de neuronas, implica un mayor coste computacional. Para ver el desempeño de la red neuronal de clasificación, se dividen los datos en 2 conjuntos de datos (5 veces), uno de validación y otro de entrenamiento. El conjunto de entrenamiento se divide nuevamente en 3 conjuntos de datos, uno que será exclusivamente para el entrenamiento, otro que será para la validación y el último para comprobar la calidad de las respuestas,

```
 if exist ('netUsingTesis.mat','file') ==0
     for cant=1:2
     switch cant
         case cant==1
             dataCompleteUsingNN=datos1;
             fichasCompleteUsing=rtas1;
             ck_paq=5;
             cNN_Max=30;
             cNN_init=2;
             cNN_step=2;
             cRun=150;
         case cant==2
```

```
            dataCompleteUsingNN=datos2;
            fichasCompleteUsing=rtas2;
            ck_paq=10;
            cNN_max=42;
            cNN_init=2;
            cNN_step=4;
            cRun=150;
     end

usarEsteT=fastTrainerNN(dataCompleteUsingNN,fichasCompleteUsing,ck_paq,cNN_Max,cNN_step,cNN_i
nit,cRun);
     end
 end
```

Funciones usadas

En esta sección están las funciones que son internas del ejecutable.

Generador de precios aleatorios

Son datos de paneles e inversores que son generados a partir de los datos que se tiene en la ANLA, dichos paneles e inversores o se usan directamente aquí, se usan exclusivamente para que generen datos a sus semejantes.

```
function price=randomPrices(baseDeDatos,estadisticas)
%BaseDeDatos = Vector con los discriminantes
%Estadisticas = Estadistica descriptiva...
%Las estadisticas importadas deben tener la siguiente estructura
%Promedio
%Desviacion estandar
%Media
%Moda
%Minimo
%Maximo
%Esto se hace para poder tener un orden a la hora de generar los datos
%aleatorios de los precios o de lo que se quiera generar
%Extrae las estadisticas
promedio=estadisticas(1,:);
desviacion=estadisticas(2,:);
minimo=estadisticas(5,:);
maximo=estadisticas(6,:);
%Determina el tamaño
[filasB,~]=size(baseDeDatos);
%Guarda una cantidad de memoria para los datos que serán usados
price=zeros(filasB,1);
%Busca los elementos discriminantes
discriminante=unique(baseDeDatos);
%Saca la cantidad de discriminantes
```

```
[filasD,~]=size(discriminante);
%Contador de filas del precio
for j=1:filasD

    for i=1:filasB

        if baseDeDatos(i)==discriminante(j)
            desvT=desviacion(j)*randi([1,3]);
            if round(promedio(j)-desvT)>minimo(j) && round(promedio(j)+desvT)<maximo(j)
              price(i)=randi([round(promedio(j)-desvT),round(promedio(j)+desvT)]);
            else
                price(i)=randi([round(minimo(j)),round(maximo(j))]);
            end
        end
    end

end

end
```

Estadísticas de la base de datos

En esta parte se sacan las estadísticas de las bases de datos. Estadísticas básicas como:

- Promedio o media
- Desviación estándar
- Mediana
- Moda
- Mínimo
- Máximo

```
function tablaEstadistica=estatisticsData(baseDeDatos)
%La formula usada para las estadisticas
report=@(data)([mean(data); std(data);median(data);mode(data);min(data);max(data)]);
[~,cBd]=size(baseDeDatos);
tablaEstadistica=zeros(6,cBd);
for j=1:cBd
    if istable(baseDeDatos(:,j))==1
        temporal=table2array(baseDeDatos(:,j));
    else
        temporal=baseDeDatos(:,j);
    end
    if isnumeric(temporal)==1
    %Parte agregada, quita los outliers
    temporal(isoutlier(temporal))=nan;
    temporal(isnan(temporal))=[];
        if isempty(temporal)==0
            tablaEstadistica(:,j)=report(temporal);
        end
```

```
    end
end
end
function bdComplete=autocompletar(baseDeDatos,estadisticas)
%Para autocompletar los datos faltantes con las estadisticas
%Primero se miran los tamaños de las bases de datos
[f,c]=size(baseDeDatos);
%Se igualan las bases de datos, esto para evitar alguna alteración futura
bdComplete=baseDeDatos;
%Se recorren todas las columnas
for j=1:c
    %Se asigna un contador para encontrar las variables con NaN
    contador=1;
    %Se resume un segmento de codigo en una variable, esto se hace para no
    %saturar la pantalla de codigo
    acortador=table2array(baseDeDatos(:,j));
    %Se pregunta ¿Alguna de éstas condiciones se cumple?)
    if isstring(acortador)==0 &&...
        isdatetime(acortador)==0 &&...
        iscategorical(acortador)==0
        %En caso que no lo sea, entonces se busca NaN
        rowNan=find(isnan(table2array(baseDeDatos(:,j))));
        %Se pregunta ¿Hay filas con NaN?
        if isempty(rowNan)==0
            %En caso que sí existan las filas con NaN
            %Se ordenan las filas aunque...
            %Esta parte no es realmente necesaria
            %Sin embargo se puso por si llegase a lanzar las filas en desorden
            rowNan=ordenar(rowNan,1,'menor');
            [cRN,~]=size(rowNan);
            %Se recorren las filas
            for i=1:f

                if contador<=cRN
                     %Se pregunta ¿La fila actual es alguna fila señalada de
                    %tener NaN?
                    if i==rowNan(contador)
                        %En caso de ser verdad, se llama las estadisticas para
                        %que generen algún dato aleatorio que esté en su rango.
                        bdComplete(i,j)=table(randomPrices(1,estadisticas(:,j)));
                        %Aumenta el contador, esto para seguir buscando
                        contador=contador+1;
                    end
                end
            end
        end
    end
end
end
```

Selección de módulos/paneles solares y de los inversores

En esta sección se iteran los paneles para cada módulo, dando como resultado una matriz de potenciales arreglos para cada inversor, según sea su caso.

Datos de entrada

1. Número máximo de arreglos en serie.
2. Número máximo de arreglos en paralelo.
3. Potencia necesaria [W].
4. Área necesaria [m^2].
5. Potencia máxima del módulo/panel [W].
6. Área del módulo/panel [m^2].
7. Ángulo sugerido según la formula [°].
8. Altura solar máxima del peor mes del año [°].
9. Azimut en un rango de horas del sol en el peor mes del año [°].

Datos de salida

1. Número de módulos/paneles en serie.
2. Número de módulos/paneles en paralelo.
3. Potencia generada [W].
4. Área ocupada [m^2]

```
 function
bases=maxPot(nMSerie,nMParalelo,potNeed,areaNeed,potMax,areaMod,betaM,solarM,azimuthM)
 a=randi([1,100]);
 rng (a)
 %Formulas
 generado=@(nSerie,nParalelo,potAr)(nSerie*nParalelo*potAr);
 distanceShadow=@(l,thetaMod,altitudeSolar,azimuthModule,azimuthSolar)...
     (l.*(cosd(thetaMod)+sind(thetaMod)/tand(altitudeSolar)*cosd(azimuthModule-
azimuthSolar)));

 cantMS=nMSerie;
 cantMP=nMParalelo;
 [rowCant,~]=size(nMSerie);
 bases=zeros(rowCant,7);
 alturaLitte=min(azimuthM(:,2));
 azimuthUsed=max(azimuthM(:,1));
 acum=[];
 for i=1:rowCant
     mpptUsed=1;
     cumplir=0;
     while cumplir==0
         relacion=randi([50,59]);

areaCom=distanceShadow(sqrt(areaMod(i)/(relacion)/100),betaM,alturaLitte,solarM,azimuthUsed)*
(areaMod(i)*(relacion));
         potGen(mpptUsed)=generado(cantMS(i),cantMP(i),potMax(i));
```

```
        areaAct(mpptUsed)=generado(cantMS(i),cantMP(i),areaCom);
        if sum(potGen)<=potNeed && sum(areaAct)<=areaNeed|| mpptUsed>3
            moduloSAcum(mpptUsed)=cantMS(i);
            moduloPAcum(mpptUsed)=cantMP(i);
            if sum(potGen)/potNeed>0.8 && sum(potGen)/potNeed<1 || mpptUsed>3

bases(i,:)=[sum(moduloSAcum),nMSerie(i),sum(moduloPAcum),nMParalelo(i),sum(potGen),sum(areaAc
t),mpptUsed];
                cumplir=1;
            else
                mpptUsed=mpptUsed+1;
            end
        else
            cantMS(i)=cantMS(i)-1;
            if cantMS(i)==0
                cantMS(i)=nMSerie(i);
                cantMP(i)=cantMP(i)-1;
                if cantMP(i)==0
                    break;
                end
            end
        end
    end
end
bases=[bases,(1:rowCant)'];
end
```

Entrenamiento de las redes neuronales

El script usado para entrenar a las redes neuronales, tiene como entrada dos distintos tipos de datos, los datos que serán exclusivamente para la red neuronal, los cuales se dividen en 70% para el entrenamiento, y 30% para entrenamiento y test.

```
function performGen=procesoNN1(x,t,data_validation,fichas_validation,fActiv1,fActiv2,neural)
%rng('default')
tic
trainFcn = 'trainlm';  % Levenberg-Marquardt backpropagation.
% Create a Fitting Network
hiddenLayerSize = neural;
net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivision
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
```

```
 net.divideParam.trainRatio = 70/100;
 net.divideParam.valRatio = 15/100;
 net.divideParam.testRatio = 15/100;

 % Choose a Performance Function
 % For a list of all performance functions type: help nnperformance
 net.performFcn = 'mse';  % Mean Squared Error

 % Choose Plot Functions
 % For a list of all plot functions type: help nnplot
 net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
     'plotregression', 'plotfit'};

 % Train the Network
 net.layers{1}.transferFcn=char(fActiv1);
 net.layers{2}.transferFcn=char(fActiv2);
 [net,tr] = train(net,x,t);

 % Test the Network
 y = net(x);
 e = gsubtract(t,y);
 performance = perform(net,t,y);

 % Recalculate Training, Validation and Test Performance
 trainTargets = t .* tr.trainMask{1};
 valTargets = t .* tr.valMask{1};
 testTargets = t .* tr.testMask{1};
 trainPerformance = perform(net,trainTargets,y);
 valPerformance = perform(net,valTargets,y);
 testPerformance = perform(net,testTargets,y);
 performaneDataUnknown=immse(sim(net,data_validation),fichas_validation);
 %El desempeño
 performGen=[toc,performance,trainPerformance,valPerformance,testPerformance,performaneDataUn
known];
 init(net);
 end
```

Se hace un scrpt que haga más facil el proceso de llamar la funcion de entrenamiento de la red neuronal

```
 function
usarEsteTable=fastTrainerNN(dataCompleteUsingNN,fichasCompleteUsing,kPaquete,cNeurona,pcNeuro
na,icNeurona,cantidadR)
 %Esta función hace todo el proceso del entrenamiento de la red neuronal
 %Los datos que piden son:
 %dataCompleteUsingNN            -> Los datos que se tienen
 %fichasCompleteUsing            -> Las respuestas de los datos
 %kPaquete                       -> La cantidad de particiones de paquetes
 %cNeurona                       -> La cantidad máxima de neuronas
```

```
 %pcNeurona                      -> La cantidad de neuronas que se salta hasta llegar a la
otra neurona
 %icNeurona                      -> La cantidad inicial de neuronas
 %cantidadR                      -> La cantidad de corridas
 %EJEMPLO
 %Datos                          -> data=rand(100,3)
 %Respuesta                      -> rta=rand(100,1)
 %Cantidad paquete               -> pq=10
 %Cantidad neuronas              -> cN=20
 %Cantidad de salto de neuronas  -> cS=2
 %Neurona inicial                -> cini=2
 %Cantidad runs                  -> runing=150
 %uTable=fastTrainerNN(data,rta,pq,cN,cs,cini,runing)
 %Las funciones de activación
 activationFunction={'poslin','logsig','tansig','purelin'};

 %Columna donde está validación
 colValidacion_sitio=4;
 %Hablando de la neurona
 %Hablando de las cantidades
 cantidadesColumnasUsing=4;
 %La cantidad inicial de columnas
 inicial=6;
 resF0=zeros(length(activationFunction),inicial*2+3);
 resFTotal=[];
 %Función de activación 1
 for fActiv0=1:length(activationFunction)
     resF1=zeros(length(activationFunction),inicial*2+2);
     %Función de activación 2
     for fActiv1=1:length(activationFunction)
         resNeural=zeros(length(icNeurona:pcNeurona:cNeurona),inicial*2+1);
         %Cantidad de neuronas
         cNUse=1;
         for cN=icNeurona:pcNeurona:cNeurona
             %Para dividirlo en una cantiad de campos
             cvp=cvpartition(fichasCompleteUsing(:,1),'kfold',kPaquete);
             resRun=zeros(cantidadR*cvp.NumTestSets,inicial);
             %Cantidad de corridas
             for runner=1:cantidadR
                 reSum=zeros(cvp.NumTestSets,inicial);
                 %Evaluando en distintos paquetes de datos
                 for kF=1:cvp.NumTestSets
 %                     if activationFunction(fActiv1)=="tansig"
 %                         fichasCompleteUsing(fichasCompleteUsing==0)=-1;
 %                     else
 %                         fichasCompleteUsing(fichasCompleteUsing==-1)=0;
 %                     end
                     %Entrenamiento
                     data_train=dataCompleteUsingNN(cvp.training(kF),:);
```

```
                    fichas_train=fichasCompleteUsing(cvp.training(kF),:);
                    %Validacion
                    data_validation=dataCompleteUsingNN(cvp.test(kF),:);
                    fichas_validation=fichasCompleteUsing(cvp.test(kF),:);
                    %Datos que serán entrenados por la red neuronal
                    x = data_train';
                    t = fichas_train';
                    %Da el desempeño
                    performGen=procesoNN1(x,t,data_validation',fichas_validation',...
                        activationFunction(fActiv0),activationFunction(fActiv1),cN);
                    reSum(kF,:)=performGen;%<-Se recopila 5    filas con 6 columnas
                %Finaliza la evaluación en distintos paquetes de datos
                end
            col_in=(runner-1)*cvp.NumTestSets+1;
            col_out=cvp.NumTestSets*runner;
            resRun(col_in:col_out,:)=reSum;%<-Se recopila 150  filas con 12 columnas
            %Finaliza los run
            end
        resNeural(cNUse,:)=[mean(resRun),std(resRun),cN];%<-Se recopila 150   filas con 13
columnas
        cNUse=cNUse+1;
        %Finaliza las cantidades de neuronas
        end

seNBest_F1=[ordenar(resNeural,colValidacion_sitio,'menor'),repmat(fActiv1,length(icNeurona:pc
Neurona:cNeurona),1)]; %*
     resF1(fActiv1,:)=seNBest_F1(1,:);%<-Se recopila 4    filas con 14 columnas
     %Finaliza la función de activación 1
     end
 seNBest_F0=ordenar(resF1,colValidacion_sitio,'menor');
 resF0(fActiv0,:)=[seNBest_F0(1,:),fActiv0];%<-Se recopila 4    filas con 15 columnas
 resFTotal=[resFTotal;seNBest_F0,repmat(fActiv0,length(activationFunction),1)];         %<-Se
recopila 16   filas con 15 columnas
 %Finaliza la función de activación 0
 end
 usarEste=ordenar(resFTotal,colValidacion_sitio,'menor');
 usarEste_names={'TiempoMean','PerformanceMean',...
     'PerformanceTrainMean','ValidationPerformanceMean','TestPerformanceMean',...
     'Set2PerformanceMean',...

'TiempoStd','PerformanceStd','PerformanceTrainStd','PerformanceValidationStd','PerformanceTes
tStd',...
     'Set2PerformanceStd','CantidadNeuronas','FActivOut','FActivIn'};
 usarEsteTable=array2table(usarEste,'VariableNames',usarEste_names);
 end
```