

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

**UTILIZACIÓN DE TÉCNICAS DE AGRUPAMIENTO PARA LA
OBTENCIÓN DE CURVAS DE CARGA REPRESENTATIVAS DE
TIPO: RESIDENCIAL, COMERCIAL E INDUSTRIAL**

**UTILIZACIÓN DE LA TÉCNICA DE AGRUPAMIENTO FUZZY K
MEANS PARA LA OBTENCIÓN DE CURVAS REPRESENTATIVAS DE
TIPO: RESIDENCIAL COMERCIAL E INDUSTRIAL**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO ELÉCTRICO**

KEVIN FABRICIO JARAMILLO QUISHPE

kevin.jaramillo@epn.edu.ec

DIRECTOR: PATRICIA ELIZABETH OTERO VALLADARES

patricia.otero@epn.edu.ec

DMQ, agosto 2022

CERTIFICACIONES

Yo, KEVIN FABRICIO JARAMILLO QUISHPE declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



Kevin Fabricio Jaramillo Quishpe

Certifico que el presente trabajo de integración curricular fue desarrollado por KEVIN FABRICIO JARAMILLO QUISHPE, bajo mi supervisión.



Patricia Elizabeth Otero Valladares
DIRECTORA

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

KEVIN FABRICIO JARAMILLO QUISHPE

PATRICIA ELIZABETH OTERO VALLADARES

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
ÍNDICE DE CONTENIDO.....	III
RESUMEN	V
ABSTRACT	VI
1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	2
1.4 Marco teórico	2
1.4.1 Minería de datos.....	2
1.4.2 Normalización de los datos.....	3
1.4.3 Método Fuzzy K-Means.....	4
1.4.4 Características de las cargas eléctricas.....	6
1.4.5 Demanda.....	6
1.4.6 Curvas de carga	6
1.4.7 Software Matlab.....	7
1.4.8 Índices de adecuación	8
2 METODOLOGÍA.....	9
2.1 Tratamiento de la base de datos.....	9
2.2 Importación de la base de datos Excel a Matlab.....	10
2.3 Desarrollo del algoritmo de agrupación Fuzzy K-means en Matlab	12
2.4 Cálculo de los índices de adecuación	19
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	20
3.1 Resultados	20

4	Conclusiones	31
5	Recomendaciones	32
6	REFERENCIAS BIBLIOGRÁFICAS	33
7	ANEXOS.....	35
	ANEXO I: BASE DE DATOS	35
	ANEXO II: SCRIPT DE MATLAB	35
	ANEXO III: FUNCIÓN MIAyCDI	57

RESUMEN

El siguiente trabajo de integración curricular tiene por objetivo principal encontrar una curva representativa de carga a través de la técnica de agrupamiento fuzzy k-means (FKM) para tres tipos de consumidores: industriales, comerciales y residenciales; para ello se utilizará una base de datos en formato Excel que contiene las mediciones de potencia de los tres grupos de consumo en intervalos de una hora a lo largo del día. Esta base de datos debe ser tratada con el fin de obtener resultados más confiables, dado que medidas distorsionadas entorpecen al algoritmo.

Se utiliza el software Matlab para desarrollar el algoritmo FKM, por lo que es necesario importar la base de datos para crear submatrices que contengan los datos de potencia clasificados desde los lunes hasta los domingos para los tres tipos de consumidores; así se obtiene tres curvas de carga por cada cliente, una que represente a la semana laboral (lunes a viernes), otra para los sábados y una última para los domingos.

Con las curvas para cada grupo de consumo se realizará el respectivo análisis y explicar el comportamiento del consumidor a lo largo del día y la semana. Finalmente, con las curvas ya obtenidas se calcula los índices de adecuación Mean Index Adequacy (MIA) y Cluster Dispersion Index (CDI) que permiten conocer el estado de las curvas a través de la medición del rendimiento del algoritmo y que tan eficiente es para agrupar los datos y encontrar un centro entre todos estos.

PALABRAS CLAVE: fuzzy k-means, curvas representativas de carga, demanda, índices de adecuación.

ABSTRACT

The following curricular integration work has as its main objective to find a representative load curve through the fuzzy k-means (FKM) clustering technique for three types of consumers: industrial, commercial, and residential; for this purpose, an Excel database containing the power measurements of the three consumption groups in one-hour intervals throughout the day will be used. This database must be treated to obtain more reliable results, since distorted measurements hinder the algorithm.

Matlab software is used to develop the FKM algorithm, so it is necessary to import the database to create sub-matrices containing the power data classified from Mondays to Sundays for the three types of consumers; thus, three load curves are obtained for each customer, one representing the work week (Monday to Friday), another for Saturdays and a last one for Sundays.

With the curves for each consumption group, the respective analysis will be carried out and the consumer's behavior throughout the day and week will be explained. Finally, with the curves already obtained, the Mean Index Adequacy (MIA) and Cluster Dispersion Index (CDI) are calculated, which allow us to know the state of the curves by measuring the performance of the algorithm and how efficient it is in grouping the data and finding a center among all these curves.

KEYWORDS: fuzzy k-means, representative load curve, demand, adequacy indexes.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Las curvas de carga son una representación del comportamiento de la demanda y sirven para explicar las variaciones de la demanda en un periodo de tiempo que puede ser diario, semanal, mensual o anual [1]. Sin embargo, la gran cantidad de datos obliga a buscar una manera sencilla que permita analizar la información y una de estas es clasificando las curvas, y esto ayuda a entender las variaciones y patrones existentes entre los diferentes consumidores y que puede ser de gran importancia para los planificadores del sistema eléctrico [1].

Por lo tanto, las curvas de carga buscan reducir la complejidad en el análisis de los datos agrupando en clases a los diferentes consumidores, y a esto se lo conoce como curvas de carga representativas (o RLC por sus siglas en inglés), que son un conjunto de curvas de cargas en una sola cuyas variaciones y patrones son similares [1]. Las RLC son usadas por las empresas eléctricas para determinar la contribución de cada grupo en la curva de carga del sistema, en el diseño eléctrico para escoger el equipamiento eléctrico o en el desarrollo de la base de datos para el sistema EMS (energy management system) [2].

Entonces las RLC pueden obtenerse de distintas formas, pero en este trabajo de integración curricular se utilizará el algoritmo fuzzy k-means (también llamado fuzzy c-means) que es una técnica de agrupamiento de datos ampliamente utilizada que permite separar en diferentes grupos (o clusters) los distintos datos basados en una lógica difusa, que utiliza el grado de pertenencia que tiene cada dato a un cluster, haciendo al algoritmo adecuado para determinar las curvas de carga representativas debido a la variación aleatoria del comportamiento de los consumidores [2].

Todos estos datos se tomarán de las mediciones de los equipos de calidad instalados en las cabeceras de los alimentadores donde los clientes se conectan a medio voltaje y de los transformadores de distribución; esta información es remitida por la empresa eléctrica CNEL-Unidad de Negocios Azogues. Por otra parte, antes de ingresar los datos al algoritmo, pasarán por un tratamiento para clasificarlos por tipo de consumidores y una subclasificación en días de la semana y fines de semana. El algoritmo se desarrollará en el software MATLAB y por medio de este se obtendrán las curvas de carga representativas para los tres tipos de consumidores: residencial, comercial e industrial.

Finalmente, para verificar el rendimiento del algoritmo se calcularán índices que adecuación que son MIA (Means Index Adequacy) que señala que tan compactos son los grupos y CDI (Clustering Dispersion Index) que indica que tan separados están los grupos. Estos índices muestran la calidad del algoritmo entre menor sea su valor [3].

1.1 Objetivo general

Utilizar la técnica de agrupamiento fuzzy k-means para obtener las curvas de carga representativas según el tipo de consumidor residencial, comercial e industrial.

1.2 Objetivos específicos

1. Realizar una investigación bibliográfica de la obtención de curvas de carga representativas según el tipo consumidor usando la técnica de agrupamiento fuzzy k-means.
2. Tratar la base de datos de cada tipo de consumidor para usarla como entrada para el algoritmo.
3. Desarrollar el algoritmo de agrupamiento en el software MATLAB y obtener las curvas de carga representativas incluyendo los índices de adecuación.
4. Analizar los resultados y emitir conclusiones.

1.3 Alcance

Partiendo de la investigación bibliográfica del algoritmo fuzzy k-means y su aplicación para obtener las curvas de carga representativas para consumidores del tipo residencial, comercial e industrial. Y con la base teórica obtenida se pasa al tratamiento de los datos para cada tipo de consumidor.

Se desarrollará el algoritmo en el software MATLAB y por medio de este mismo programa se desplegarán las curvas representativas de cada tipo de consumidor analizado y también los índices MIA y CDI.

Los resultados obtenidos permitirán evaluar la técnica de agrupamiento utilizada y su adecuación en la obtención de curvas representativas de carga, para finalmente emitir conclusiones y recomendaciones.

1.4 Marco teórico

1.4.1 Minería de datos

La minería de datos aparece, no por el avance de la tecnología, sino por la necesidad de manejar la gran cantidad de datos almacenados actualmente en los sistemas de información de instituciones, empresas, gobiernos y demás organizaciones [4]. Estos datos son la materia prima que se utiliza para obtener un conocimiento que puede ser muy valioso para las organizaciones y la toma de decisiones [4] pero a veces extraer esta información

puede ser extremadamente retardador [5], ya que las técnicas y herramientas del análisis de datos tradicional no pueden utilizarse por el gran volumen de los datos [5].

Entonces la minería de datos mezcla el análisis de datos tradicional con complejos algoritmos para procesamiento de datos [5]. Pero no son las únicas disciplinas que se integran, es decir, no solamente se requieren habilidades matemáticas e informáticas sino también aquellas especializadas en negocios y organización empresarial y muchas más que ayudan a la extracción completa del conocimiento a través de los datos [6]. Además, la cantidad de algoritmos que se utilizan para minería es muy basta, pero se los divide en dos grupos: métodos supervisados y no supervisados.

Los métodos no supervisados son algoritmos usados especialmente en técnicas de agrupamiento cuyo objetivo es encontrar grupos similares en una base datos que a primera vista no tienen ninguna relación entre ellos, es decir, el agrupamiento (o clustering) descubre patrones de similitud en los datos y los divide en grupos basándose en estas semejanzas entre los datos [6]. Existen dos métodos principales en los algoritmos de agrupamiento: *métodos jerárquicos* y *métodos particionales*, este último es el que se encarga de agrupar a los datos en conjuntos como se puede observar en la Figura 1.1.

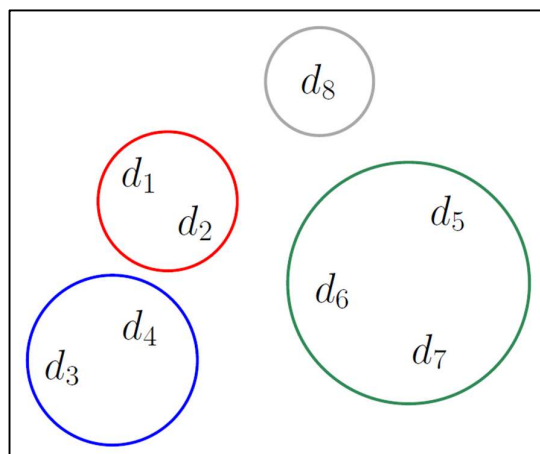


Figura 1.1 Agrupamiento particional [6]

1.4.2 Normalización de los datos

Generalmente las diferentes características de los datos no permiten realizar una comparación entre ellos debido a que no se encuentran en una misma referencia, y por ello es necesario llevarlos a una misma escala que facilite esta comparativa [7]. La normalización es de gran ayuda en varios métodos de minería de datos donde existen valores extremos, ya sean muy altos o bajos pero que pueden llegar a entorpecer al resto

de la base de datos y por lo tanto distorsionar el resultado del modelo [6]. Algunos de los métodos de normalización son [6]:

- **Normalización por el máximo:** se basa en encontrar el valor máximo en un conjunto de datos y dividirlo para el resto de los valores, obteniendo un rango de valores entre [0,1] [6].

$$z_i = \frac{x_i}{x_{max}} \quad (1)$$

Donde z_i es el valor normalizado, x_i representa a los valores del grupo y x_{max} es el valor máximo del conjunto de datos [6].

- **Normalización por diferencia:** este método intenta compensar la diferencia entre los valores del grupo con respecto al máximo y mínimo del conjunto de datos [6].

$$z_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (2)$$

- **Normalización basada en la desviación estándar:** este método también se conoce como estandarización de valores, y se caracteriza por que la media es igual a cero y la desviación estándar es igual a uno, por lo que los valores del resto del conjunto de datos se encontrarán en este rango [6].

$$z_i = \frac{x_i - \mu}{\sigma} \quad (3)$$

Donde μ representa la media y σ la varianza del conjunto de datos.

1.4.3 Método Fuzzy K-Means

Para entender el algoritmo de fuzzy k-means o también conocido como fuzzy c-means es necesario primero entender el algoritmo k-means en el cual se basa; este es un algoritmo iterativo que forma parte de las técnicas de agrupamiento no supervisado que se encarga de agrupar n datos en k clústeres (o grupos), a través de la minimización de la distancia euclidiana entre los datos y el centroide del clúster respectivo [8]. Algunas de las características de este algoritmo son [6]:

- El número de grupos o clústeres debe definirse antes de empezar el algoritmo.
- Cada clúster se define por medio de un punto llamado centroide que sirve para determinar el grupo al que pertenece cada dato.

El funcionamiento del algoritmo se basa en tres pasos que se repiten hasta encontrar el número de grupos, y estos son [6]:

- i. Identificar k puntos como centroides iniciales.
- ii. Asignar a cada centroide los puntos más cercanos de la base de datos y formar los k grupos.
- iii. Recalcular los centroides apoyándose en los datos que forman parte del grupo.

A continuación, se presenta un pseudocódigo que representa la operación del algoritmo [6]:

Entrada: D (conjunto de datos) y k (número de clústeres)

Inicializar los clústeres vacíos $P = \{p_1, p_2, \dots, p_k\}$

Seleccionar los centroides iniciales $\mu_1, \mu_2, \dots, \mu_k$

Mientras (condición de parada no satisfecha) **hacer**

Para todo $d_i \in D$ **hacer**

Asignar la instancia d_i al clúster p_j | $d(d_i, \mu_j)$ es mínima

Fin para

Para todo $p_i \in P$ **hacer**

Recalcular μ_i según los valores $d_i \in p_i$

Fin para

Fin mientras

Devolver conjunto de clústeres P

Una vez revisado el método k-means se puede notar que cada dato pertenece a uno y solamente a un grupo, sin embargo, en muchas ocasiones sucede que un dato se encuentra muy cerca de dos o más clústeres por lo que es difícil clasificarlo en un grupo, y para este inconveniente se desarrolló el algoritmo de fuzzy k-means o también llamado fuzzy c-means; este algoritmo asigna a cada dato un grado de pertenencia a cada clúster construyendo una partición difusa (de ahí el nombre fuzzy) o suave de la base de datos, que matemáticamente se define como [9]:

Sea X un conjunto de datos y x_i un elemento perteneciente a X , se dice que una partición $P = \{C_1, C_2, \dots, C_n\}$ es una partición suave de X si y solo si cumple con las siguientes condiciones [10]:

- i) $\forall x_i \in X, \forall C_j \in P \quad 0 \leq \mu_{C_j}(x_i) \leq 1$
- ii) $\forall x_i \in X \exists C_j \in P \text{ tal que } \mu_{C_j}(x_i) > 0$

Donde $\mu_{C_j}(x_i)$ representa el grado de pertenencia de x_i al clúster C_j .

1.4.4 Características de las cargas eléctricas

Las cargas eléctricas son una parte muy importante en cualquier sistema de distribución de energía eléctrica, ya que en base a estas se realizan algunas actividades, tales como [11]:

- Selección y comprobación de los conductores y transformadores.
- Determinación de la caída de voltaje.
- Establecer los sistemas de protección, entre otros.

Por estas razones la estimación correcta de la carga es crucial, ya que, si se considera una carga muy alta todo el sistema quedaría sobredimensionado lo que se traduce en una mayor inversión, y si es una carga menor de la esperada los equipos se sobrecargarán y tendrán mayores pérdidas e incluso una reducción en su vida útil [11]. Entonces a las cargas dentro de un sistema de distribución se conoce como *demanda* y representa a los usuarios y la cantidad de energía eléctrica que estos necesitan para abastecer sus necesidades.

1.4.5 Demanda

Según [12] la demanda de una instalación o sistema es la carga en los terminales de recepción promediada en un intervalo de tiempo específico; a este intervalo se lo conoce como *intervalo de demanda* y puede ser de 15 min, 30 min, 1 hora e incluso mayor sin embargo el valor de este intervalo dependerá del valor de la demanda que se desee conocer [12]. Además, a la variación de la demanda con respecto al tiempo se conoce como *curva de carga*.

1.4.6 Curvas de carga

Las curvas de carga se obtienen colocando en el eje de las abscisas el tiempo, que sería el intervalo de demanda, y en el otro eje la carga pico correspondiente a ese instante; estas curvas representan el comportamiento de las cargas que pueden ser residenciales, comerciales e industriales. Se utilizan como base para el arranque o interrupción de unidades del sistema, maniobras que modifican la red y también para escoger el número y potencia de los transformadores en las redes de distribución. Según el período que cubran

estas curvas se pueden clasificar en diarias, semanales, mensuales o anuales en la Figura 1.2 se puede observar un ejemplo de curva de carga diaria típica para un cliente residencial, comercial e industrial [11].

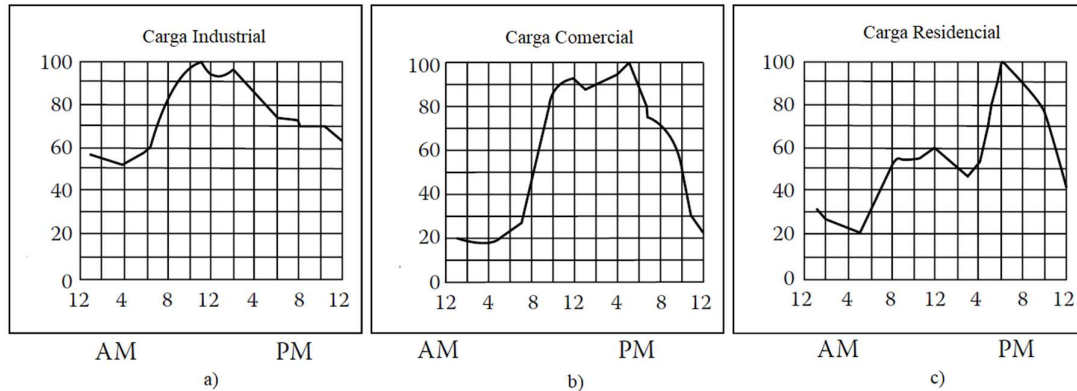


Figura 1.2 Curvas de carga: a) industrial, b) comercial y c) residencial [12]

Estas curvas típicas tienen distinto comportamiento que está en coherencia con las actividades que se realizan, por ejemplo la curva residencial comienza su consumo alrededor de las 6 am donde la demanda comienza a crecer porque las personas despiertan y se preparan para ir a sus trabajos, luego la curva se mantiene relativamente estable hasta las 5 pm que nuevamente sube y esto es debido a que las personas llegan de su trabajo a sus casas y consumen más energía por los diferentes dispositivos como: focos, televisiones, computadoras entre otros aparatos hasta llegar a un pico máximo que en este caso está alrededor de las 7 pm y a partir de este punto la demanda disminuye, ya que las personas van a dormir apagando todos sus dispositivos [11], [12].

Por otra parte, la curva para carga comercial comienza su actividad a las 8 am que coincide con el inicio de la jornada laboral y se mantiene hasta las 6 pm que nuevamente concuerda con la salida de los trabajos. Finalmente, la curva para carga industrial es un caso que depende mucho de la actividad que se esté realizando, y para este caso se puede observar que se trata de una industria que produce las 24 horas del día, ya que su consumo nunca decae notablemente, pero si la aumenta en el horario de trabajo por lo que puede ser una empresa que trabaje en base a turnos para no parar su producción [11], [12].

1.4.7 Software Matlab

Matlab pertenece a la empresa MathWorks que se especializa en software de computación matemática y que incluye a toda una familia de productos como se puede observar en la Figura 1.3. El nombre de MATLAB proviene de la abreviatura de **Matrix Laboratory** (laboratorio matricial) y es una herramienta computacional usada para resolver problemas

matemáticos, especialmente aquellos relacionados con matrices y gráficas, además cuenta con una interfaz amigable para la programación, así como la visualización de los resultados. Generalmente es usado para tareas como: desarrollo de algoritmos, adquisición de datos, análisis, exploración y visualización de datos, así como también modelación, simulación y gráficas científicas [13].

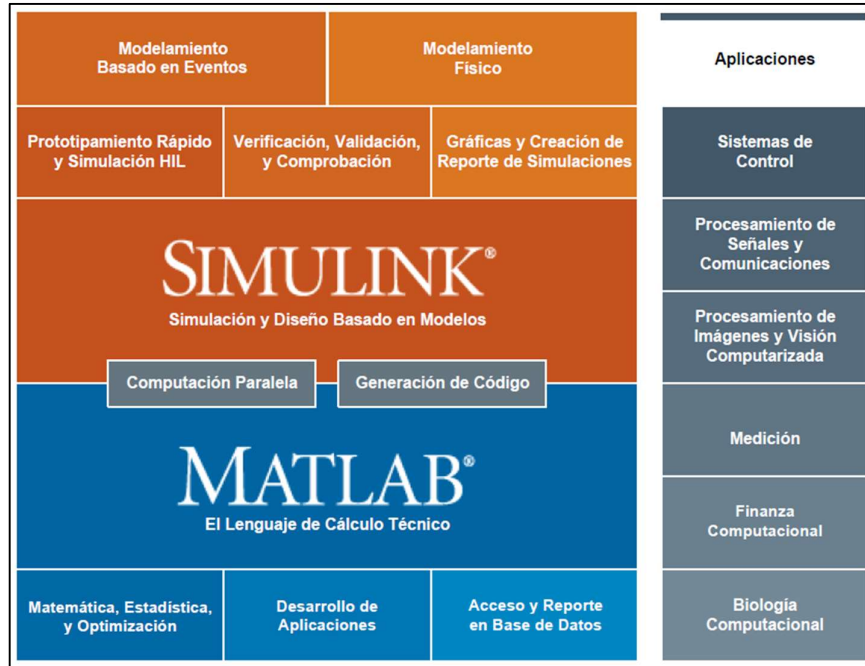


Figura 1.3 Familia de productos MathWorks

1.4.8 Índices de adecuación

Para determinar que los clústeres obtenidos a través del algoritmo de agrupación son aceptables se utiliza índices de adecuación, y existen varios de ellos, pero en este trabajo se utilizarán solo dos; uno de ellos llamado MIA (Mean Index Adequacy) que mide la dispersión de los datos y consiste en calcular la distancia promedio de cada miembro de un clúster al centroide de este por medio de la siguiente ecuación [14]:

$$MIA = \sqrt{\frac{1}{K} \sum_{i=1}^K d^2(C_i)} \quad (4)$$

Donde C_i es el centro calculado del clúster i y $d^2(C_i)$ es la distancia euclidiana promedio al cuadrado; entonces grandes distancias reflejan un mal ajuste de los datos es decir que los datos se encuentran muy dispersos del centro. Y el otro índice se conoce como CDI (Cluster Dispersion Index) que mide la calidad del agrupamiento y entre menor sea el valor mejor es la agrupación y se calcula con la siguiente expresión [14]:

$$CDI = \frac{1}{d(C)} \sqrt{\sum_{i=1}^K d^2(C_i)} \quad (5)$$

Donde $d(C)$ es la distancia promedio de los clústeres entre dos cualesquiera de ellos [14].

2 METODOLOGÍA

2.1 Tratamiento de la base de datos

Antes de utilizar la información contenida en la base de datos es importante realizar un tratamiento de estos con el objetivo de que los resultados sean más confiables. La base de datos fue proporcionada por la Empresa Eléctrica Azogues y es un archivo Excel con nombre "BDEEA.xlsx" proveniente de la ciudad de Azogues que contiene las mediciones de potencia de los tres grupos de consumo de interés, que son: industrial, comercial y residencial; además incluye la hora y día de la semana en la que se tomó la medición. Así pues, el archivo tiene 24 valores de potencia por día, es decir, una medición por hora sin embargo algunas celdas se encontraban en blanco y fueron eliminadas de la base de datos mientras que otras contenían valores de potencia cero o negativos, pero en este caso se las mantuvo para no reducir más el tamaño de la base de datos. Por otra parte, algunas fechas de las mediciones no coincidían con el formato del archivo, pero este error se arregló modificándolas para que tengan coherencia con el resto de la base de datos.

Por último, para la normalización de datos se tomó el valor de potencia máxima medida de un día y dividiéndolo para cada uno de los datos correspondientes al mismo día como se puede observar en la siguiente expresión:

$$P_{ij,norm} = \frac{P_{ij}}{P_{i,max}} \quad (6)$$

Donde P_{ij} es el valor de potencia medido en el día i y a una hora j , mientras que $P_{i,max}$ es la potencia máxima medida del día i y al dividir estos dos valores se obtiene $P_{ij,norm}$ que representa la potencia normalizada.

Con el tratamiento realizado en la base de datos se trató de asegurar la mayor calidad en la información ya que esta servirá como entrada para el software Matlab donde se realizará el análisis de agrupamiento a través de la técnica fuzzy k-means.

2.2 Importación de la base de datos Excel a Matlab

Como se explicó anteriormente la base de datos se encuentra en formato Excel y se realizó un tratamiento de los datos con el objetivo de tener una mayor calidad de estos. Para importar la base de datos hacía el software Matlab en las nuevas versiones se puede utilizar tres funciones que permiten llevar a cabo la misma acción, estas son: “*readtable*, *readcell* y *readmatrix*” [15]. Cada una de estas funciones tiene por objetivo transferir toda la información contenida en el archivo Excel hacía Matlab con la diferencia de que las tres funciones manejan un distinto tipo de dato.

Tabla 2.1. Funciones y su tipo de dato

Función	Tipo de dato	Sintaxis
readtable	Tabla	$T = \text{readtable}(\text{filename}, \text{opts})$
readcell	Celda	$T = \text{readcell}(\text{filename}, \text{opts})$
readmatrix	Matriz	$T = \text{readmatrix}(\text{filename}, \text{opts})$

En la Tabla 2.1 se puede observar el tipo de dato que crean las funciones y su sintaxis que en los tres casos es similar. Para este trabajo se decidió utilizar la función *readmatrix* debido a que tiene la característica de poder escoger el formato de los datos que contendrá la matriz, ya sea double, string, duration entre otros porque como se puede observar en la Figura 2.1 el archivo Excel tiene varias columnas y cada una abarca diferente información por lo que esta propiedad será de gran ayuda para el proceso de importación.

A	B	C	D	E	F	G	H	I	J	K	L	M	N
Medidor	Fecha	Hora	Promedio	Promedio	FechaSola	HoraSola	PmaxPorD	PuPotPron	DiaSem	ClaveGCo	GrupodeConsum	Parr	ClaveParroquia
400096	martes, noviembre 06, 2018	3:00:00 PM	3181.935	915.241667	06-11-martes	15	3181.935	1	2	30 Industrial	SAN MIGUEL	22	
400096	martes, noviembre 06, 2018	4:00:00 PM	3092.08167	880.815	06-11-martes	16	3181.935	0.97176542	2	30 Industrial	SAN MIGUEL	22	
400096	martes, noviembre 06, 2018	5:00:00 PM	2969.395	825.063333	06-11-martes	17	3181.935	0.93320417	2	30 Industrial	SAN MIGUEL	22	
400096	martes, noviembre 06, 2018	6:00:00 PM	2994.27667	827.126667	06-11-martes	18	3181.935	0.94102383	2	30 Industrial	SAN MIGUEL	22	
400096	martes, noviembre 06, 2018	7:00:00 PM	2844.78167	764.751667	06-11-martes	19	3181.935	0.89404141	2	30 Industrial	SAN MIGUEL	22	
400096	martes, noviembre 06, 2018	8:00:00 PM	2844.71833	777.455	06-11-martes	20	3181.935	0.89402151	2	30 Industrial	SAN MIGUEL	22	
400096	martes, noviembre 06, 2018	9:00:00 PM	2760.63167	777.685	06-11-martes	21	3181.935	0.86759524	2	30 Industrial	SAN MIGUEL	22	
400096	martes, noviembre 06, 2018	10:00:00 PM	2780.085	693.416667	06-11-martes	22	3181.935	0.87370892	2	30 Industrial	SAN MIGUEL	22	
400096	martes, noviembre 06, 2018	11:00:00 PM	2736.58667	663.463333	06-11-martes	23	3181.935	0.86003852	2	30 Industrial	SAN MIGUEL	22	
400096	miércoles, noviembre 07, 2018	12:00:00 AM	2676.89667	504.296667	07-11-miérco	0	3137.53	0.853186	3	30 Industrial	SAN MIGUEL	22	
400096	miércoles, noviembre 07, 2018	1:00:00 AM	2554.19667	436.41	07-11-miérco	1	3137.53	0.8340788	3	30 Industrial	SAN MIGUEL	22	
400096	miércoles, noviembre 07, 2018	2:00:00 AM	2577.23667	440.91	07-11-miérco	2	3137.53	0.82142216	3	30 Industrial	SAN MIGUEL	22	
400096	miércoles, noviembre 07, 2018	3:00:00 AM	2596.155	438.431667	07-11-miérco	3	3137.53	0.82745185	3	30 Industrial	SAN MIGUEL	22	
400096	miércoles, noviembre 07, 2018	4:00:00 AM	2594.54	439.383333	07-11-miérco	4	3137.53	0.82693711	3	30 Industrial	SAN MIGUEL	22	
400096	miércoles, noviembre 07, 2018	5:00:00 AM	2568.195	447.171667	07-11-miérco	5	3137.53	0.81854038	3	30 Industrial	SAN MIGUEL	22	
400096	miércoles, noviembre 07, 2018	6:00:00 AM	2587.27167	503.696667	07-11-miérco	6	3137.53	0.82462053	3	30 Industrial	SAN MIGUEL	22	
400096	miércoles, noviembre 07, 2018	7:00:00 AM	2584.37167	492.941667	07-11-miérco	7	3137.53	0.82369624	3	30 Industrial	SAN MIGUEL	22	
400096	miércoles, noviembre 07, 2018	8:00:00 AM	2586.78167	465.428333	07-11-miérco	8	3137.53	0.82446436	3	30 Industrial	SAN MIGUEL	22	
400096	miércoles, noviembre 07, 2018	9:00:00 AM	2559.79333	451.76	07-11-miérco	9	3137.53	0.81586258	3	30 Industrial	SAN MIGUEL	22	
400096	miércoles, noviembre 07, 2018	10:00:00 AM	2522.11667	442.495	07-11-miérco	10	3137.53	0.8038542	3	30 Industrial	SAN MIGUEL	22	
400096	miércoles, noviembre 07, 2018	11:00:00 AM	2658.53333	501.393333	07-11-miérco	11	3137.53	0.8473332	3	30 Industrial	SAN MIGUEL	22	

Figura 2.1. Archivo Excel “BDEEA.xlsx”

A continuación, se presentará el pseudocódigo que detalla el proceso que se siguió para importar la base de datos en Excel hacía Matlab:

leer el archivo “BDEEA.xlsx” en formato duration

Guardar en una variable la columna correspondiente a la hora

Leer el archivo "BDEEA.xlsx" en formato double

Guardar en una variable la columna correspondiente a las mediciones de potencia normalizadas y en otra variable la columna correspondiente a los días de la semana

Leer el archivo "BDEEA.xlsx" en formato string

Guardar en una variable la columna correspondiente al grupo de consumo

Unir las cuatro variables para formar una sola matriz que contenga únicamente los datos necesarios para realizar el análisis de agrupamiento

Todos estos pasos se ven materializados en la Figura 2.2 que muestra el código implementado en el software Matlab para realizar la importación de la base de datos, que comienza leyendo los datos del archivo BDEEA.xlsx. Con el comando "OutputType" se escoge el formato de los datos, que para el caso de tiempos el formato que se utiliza es "duration", y de esta manera se almacena la columna correspondiente a las horas de las mediciones de potencia en la variable "tiempo". Aunque una cuestión que se presentó con los datos de tiempo fue que la hora 24:00 se representaba con cero por lo que se decidió realizar un barrido de toda la matriz para sustituir los valores.

```
datos_aux=readmatrix("BDEEA.xlsx","OutputType","duration");
tiempo=hours(datos_aux(:,3));%tiempo en horas
[f,c]=size(tiempo);
for i=1:f%para cambiar el valor de 0 horas a 24 horas.
    if tiempo(i,1)==0
        tiempo(i,1)=24;
    end
end
datos_aux2=readmatrix("BDEEA.xlsx","OutputType","double");
pot_prom_pu=datos_aux2(:,9);
dia_sem=datos_aux2(:,10);
datos_aux3=readmatrix("BDEEA.xlsx","OutputType","string");
grupo=datos_aux3(:,12);
datos_val=[tiempo pot_prom_pu dia_sem grupo];%formacion de la matriz que se va
%utilizar para el analisis
```

Figura 2.2 Código para la importación de los datos del archivo Excel

Después de obtener los tiempos, nuevamente se lee el archivo Excel, pero en esta ocasión el formato que se escogió para los datos es "double" porque lo que se busca es almacenar los valores de potencia normalizada y los días de la semana, ya que los días en el archivo Excel están representados a través de números como se puede observar en la Tabla 2.1,

así pues las mediciones de potencia se guardan en la variable “*pot_prom_pu*” mientras que los días en la variable “*día_sem*”.

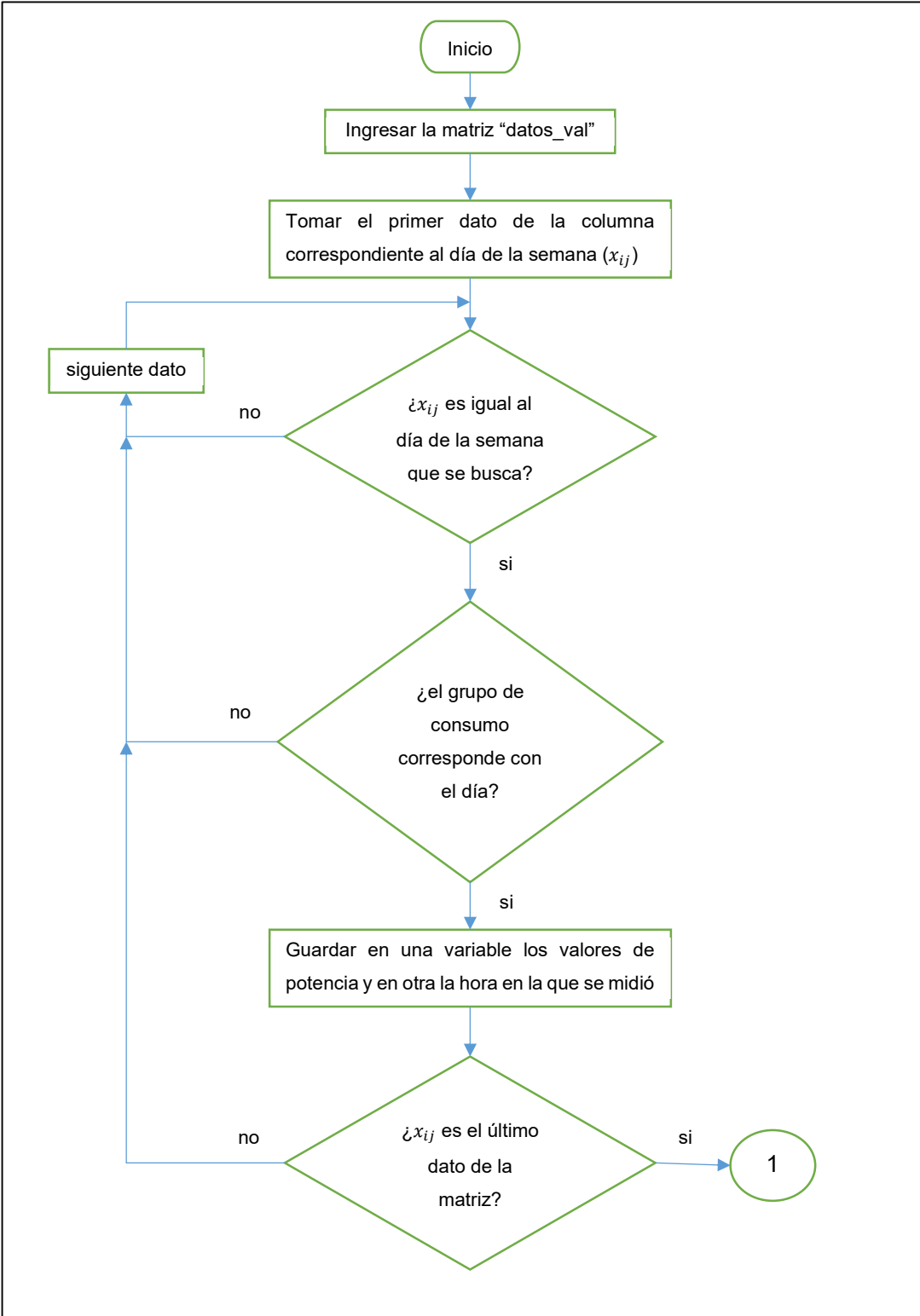
Tabla 2.2 Representación de los días de la semana

Día de la semana	Número
lunes	1
martes	2
miércoles	3
jueves	4
viernes	5
sábado	6
domingo	7

Por último, se lee una vez más el archivo Excel y el formato para los datos en esta ocasión es “*string*” porque en este caso lo que se busca almacenar es el grupo de consumo y estos están contenidos en la base de datos en formato de caracteres por lo que es preferible importarlos en el mismo formato y guardarlos en una variable llamada “*grupo*”. Entonces la finalidad de este código de importación era crear cuatro submatrices que contengan los datos de horas, valores de potencia, día de la semana y grupo de consumo para posteriormente unirlos en una sola matriz llamada “*datos_val*” que condensa toda la información necesaria para realizar el algoritmo de agrupamiento en el software Matlab.

2.3 Desarrollo del algoritmo de agrupación Fuzzy K-means en Matlab

En esta sección se explicará el desarrollo del algoritmo FKM para el análisis de la información guardada en la matriz creada al final del literal anterior (“*datos_val*”), pero esta exposición se la realizará únicamente para el caso de los consumidores industriales, debido a que el procedimiento será el mismo tanto para los comerciales como residenciales.



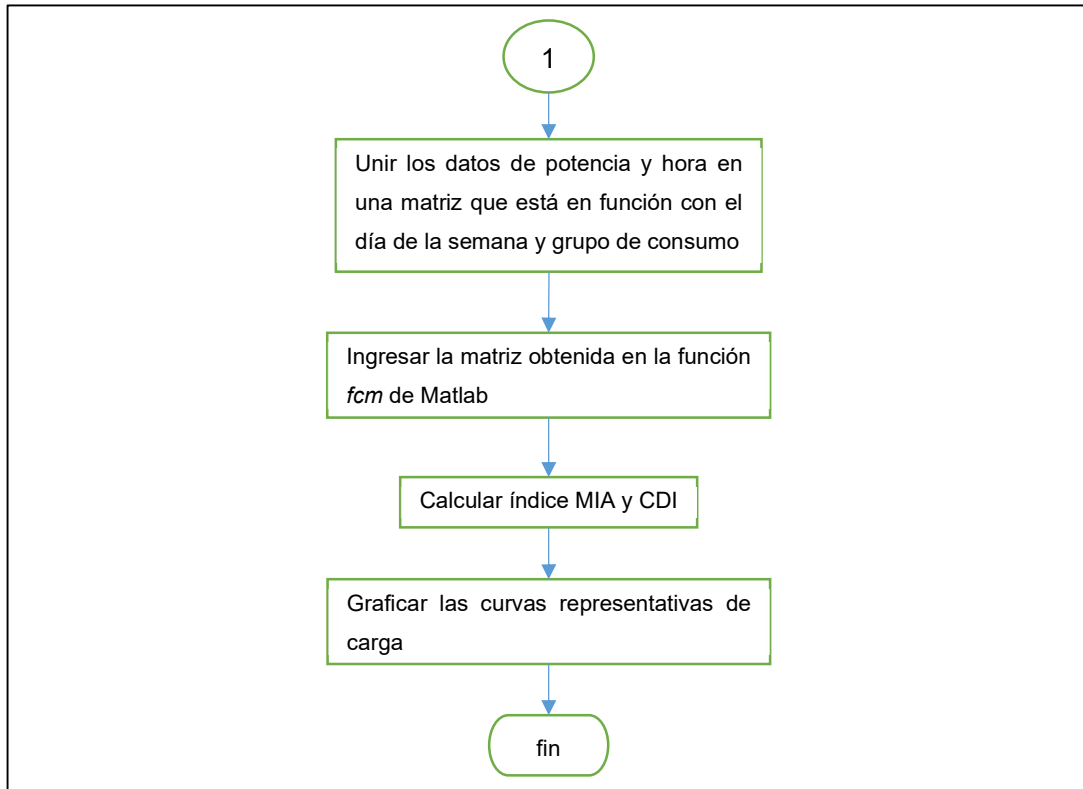


Figura 2.3 Diagrama de flujo del algoritmo desarrollado

La Figura 2.3 muestra el diagrama de flujo que se estructuró para clasificar la base de datos en función del día de la semana y grupo de consumo, realizar el análisis de agrupamiento y obtener las curvas de carga con sus respectivos índices de adecuación. Con esto y junto a porciones del código ya implementado en el software Matlab lo que se busca es enseñar de la forma más sencilla posible cómo funciona el programa desarrollado y los resultados obtenidos por este.

Así pues, la Figura 2.4 presenta la porción de código utilizado para clasificar la base datos que comienza obteniendo el tamaño (filas y columnas) de la matriz “*datos_val*” para poder realizar el barrido de toda la matriz por medio de un ciclo *for*, y dentro de este una condición *if* que expresa que: “*si el dato del día de la semana es igual a uno (o lunes) y el grupo de consumo corresponde a industrial*” entonces se guarda en la variable “*hora_L_ind*” el valor de la hora y en “*pt_L_ind*” la potencia respectiva a ese tiempo.

Sin embargo, es importante notar que la condición se hace por medio del comando “*strcmp*” que sirve para comparar cadenas de caracteres [16] y esto es debido a que la matriz principal (“*datos_val*”) es del tipo *string* por lo que es necesario transformar los valores de potencia y tiempo al formato “*double*” porque no es posible realizar cálculos ni utilizar algunas funciones con caracteres, por lo tanto para el proceso de cambio se utilizó el

comando “*str2double*” [17] y una vez realizado, el último paso que falta es unir en una sola variable los datos de potencia y tiempo obteniendo una matriz llamada “*lunes_ind*” que contiene todos los datos correspondientes al día lunes del grupo industrial, y este es el procedimiento que se utilizará para el resto de los días de la semana y el grupo de consumo respectivo.

```
[filas,columnas]=size(datos_val);
aux=1;
for i=1:filas%segmento para el dia lunes, la hora con su valor de potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'1') && strcmp(datos_val(i,j+1),'Industrial')
            hora_L_ind(aux,1)=datos_val(i,1);
            pt_L_ind(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
end
A=str2double(hora_L_ind);
B=str2double(pt_L_ind);
lunes_ind=[A B];%forma la matriz que corresponde al dia lunes para
%el grupo industrial
```

Figura 2.4 Código para clasificar la base de datos en función del día de la semana y grupo de consumo

Por otra parte, para el algoritmo de agrupación fuzzy k-means (FKM) el software Matlab cuenta con una función específica para realizar esta técnica de agrupamiento llamada “*fcm*”, cuya sintaxis se muestra a continuación:

$$[centers, U] = fcm(data, N_c)$$

$$[centers, U] = fcm(data, N_c, options)$$

Donde **data** corresponde a la base de datos que se desea agrupar, **centers** contiene el número de centros calculados en base a **N_c**, **U** representa la matriz de partición difusa calculada que contiene los grados de pertenencia de cada punto dentro de cada clúster y **options** es un vector que permite controlar algunas opciones para el proceso de agrupamiento [18]. En la Tabla 2.3 se detallan las diferentes opciones que se pueden controlar, pero si cualquiera de estas en el vector se coloca como NaN se utilizará el valor por defecto.

Tabla 2.3 Opciones de agrupamiento de la función fcm [18]

Opción	Descripción	Valor por defecto
Opción (1)	Exponente de la matriz de partición, que se representa como un escalar mayor a 1. Controla la superposición aproximada entre los clústeres donde valores grandes expresan un mayor grado de superposición.	2
Opción (2)	Máximo número de iteraciones	100
Opción (3)	Error mínimo de la función objetivo entre dos iteraciones consecutivas	$1 * 10^{-5}$
Opción (4)	Indicador de visualización de información que muestra el valor de la función de objetivo después de cada iteración: True (1) – Muestra el valor de la función objetivo False (0) – No muestra el valor de la función objetivo	true

Ahora para poder utilizar esta función es necesario instalar una herramienta (o *toolbox*) en el software Matlab llamada “*Fuzzy Logic Toolbox*” que provee a Matlab de funciones, aplicaciones entre otras para realizar análisis, diseños y simulaciones basadas en lógica difusa [19]. La instalación de esta herramienta se la hace a través del software Matlab en la pestaña “*APPS*” en la opción “*Get More Apps*” como se muestra en la Figura 2.5.

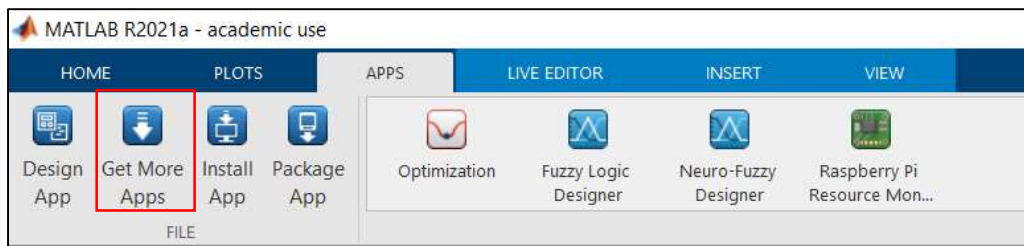


Figura 2.5 Pestaña “*APPS*” del software Matlab

Al dar clic sobre este botón se abrirá una nueva ventana que corresponde al explorador de extensiones (o Add-on explorer) y en esta se escribirá el nombre de la herramienta en la barra de búsqueda como se enseña en la Figura 2.6

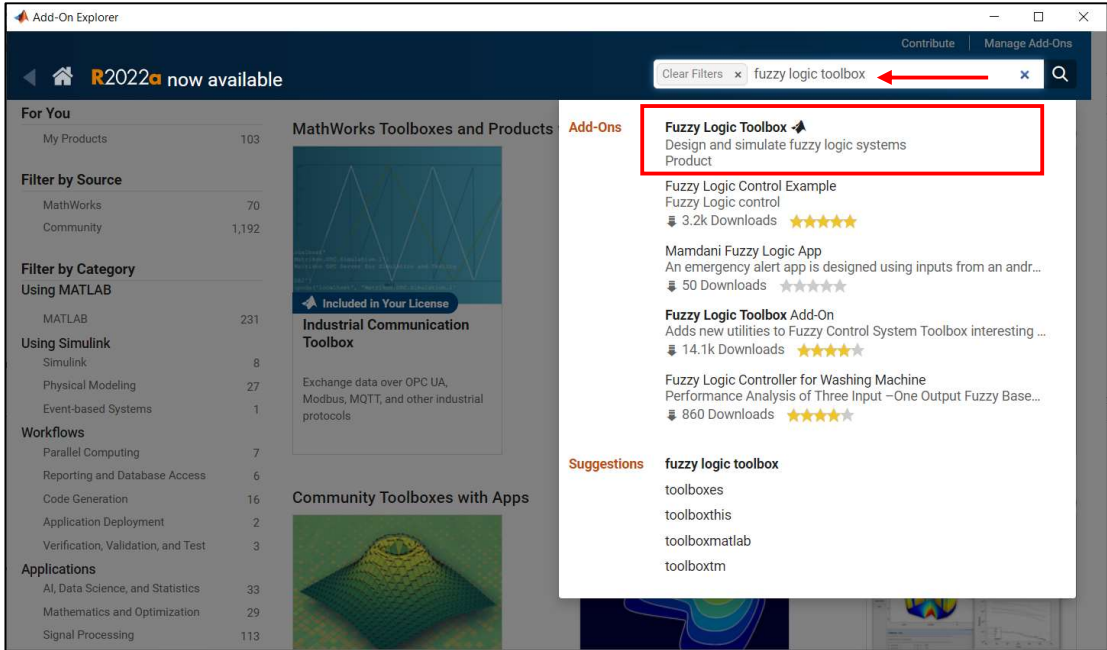


Figura 2.6 Add-on Explorer

Por último, se desplegará la interfaz de la herramienta como se muestra en la Figura 2.7 y se debe dar clic sobre el botón “ADD” y seguir los pasos de instalación, y una vez terminado el proceso poder hacer uso de las nuevas habilidades de Matlab.

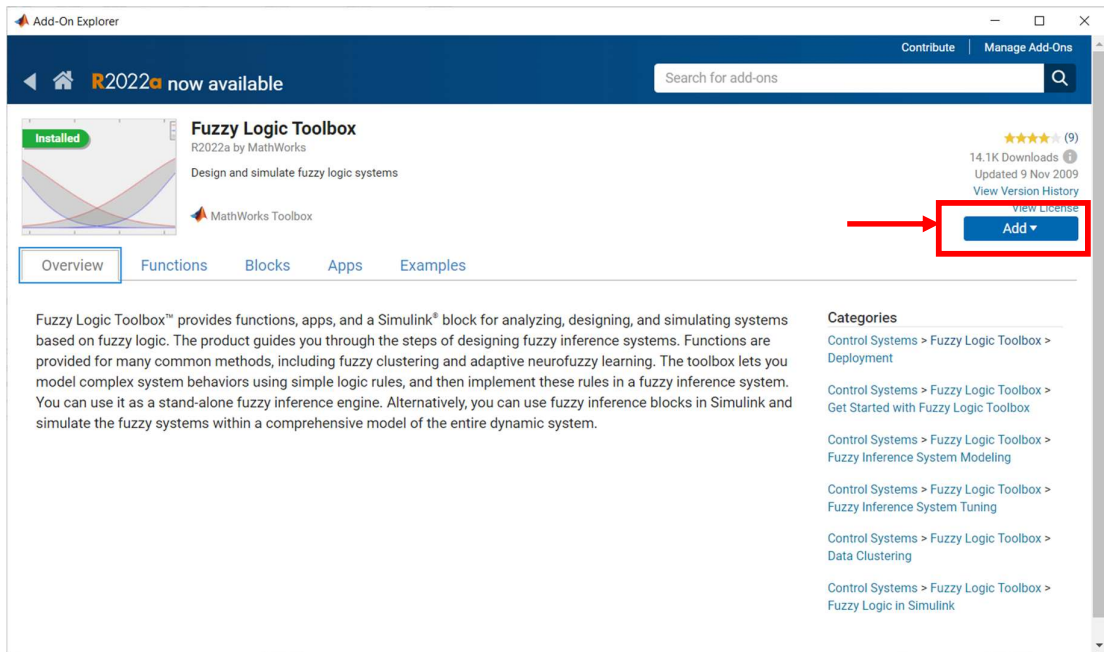


Figura 2.7 Fuzzy Logic Toolbox

Luego de la explicación de la función “*fcm*” y sus requisitos en la Figura 2.8 se puede observar el código implementado para realizar el algoritmo FKM que comienza definiendo el número de centros o centroides (N) que se desea calcular, luego se establece un vector “*options*” que contiene las opciones con las que se quiere realizar el agrupamiento y donde se especifica que no se muestren los valores de la función objetivo en cada iteración y para el resto que se trabaje con sus valores por defecto (NaN); estos datos en conjunto con la matriz “*lunes_ind*” ingresan dentro de la función “*fcm*” que entrega dos matrices, la primera llamada “*C_ind_L*” que contiene los centros calculados (para los datos del día lunes del grupo industrial) y otra matriz “*U_ind_L*” que corresponde a la matriz de partición difusa, la cual almacena los grados de pertenencia de cada dato dentro de cada clúster.

Posteriormente se utiliza el comando “*max*” para obtener los valores más altos de la matriz “*U_ind_L*” y guardarlos en otra llamada “*maxU_ind_L*”, además se crea un vector de celdas (“*index_ind_L*”) con tantas celdas como centroides se definan, y luego estos se emplean en un ciclo “*for*” junto el comando “*find*” para clasificar cada uno de los datos en el clúster para el que tiene el mayor grado pertenencia [20], [21].

```

N=12; %# de centros
options=[NaN NaN NaN 0];
[C_ind_L,U_ind_L]=fcm(lunes_ind,N,options);%(fin)
%Para clasificar cada punto en el cluster para el que tiene
%el mayor grado de pertenencia.
maxU_ind_L = max(U_ind_L);
index_ind_L = cell(N,1);
for k=1:N
    index_ind_L{k} = find(U_ind_L(k,:) == maxU_ind_L);
end%(fin de la clasificacion)
%Para graficar los resultados de la agrupacion
figure
hold on
for k=1:N
    plot(lunes_ind(index_ind_L{k},1),lunes_ind(index_ind_L{k},2),'o');
    plot(C_ind_L(k,1),C_ind_L(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
hold off%(fin de la graficacion)

```

Figura 2.8 Algoritmo FKM para la base de datos “*lunes_ind*”

Finalmente, para visualizar los datos se debe graficar la matriz con los grados de pertenencia y los centroides calculados, pero debido a que son muchos datos es necesario utilizar un ciclo “*for*” que en cada iteración dibuje en una misma gráfica cada uno de los valores obteniendo la Figura 2.9 donde se puede observar que los centroides son

representados por una X mientras que a los datos de los clústeres como pequeños círculos (o) de diferentes colores. Además, si se observa los centroides se puede notar que estos forman una curva que sería la curva representativa de carga para el día lunes de los industriales, sin embargo, eso se explicará con más detalle en la sección de los resultados ya que las gráficas mostradas en este momento solo corresponden a una muestra de cómo trabaja el algoritmo de agrupación FKM.

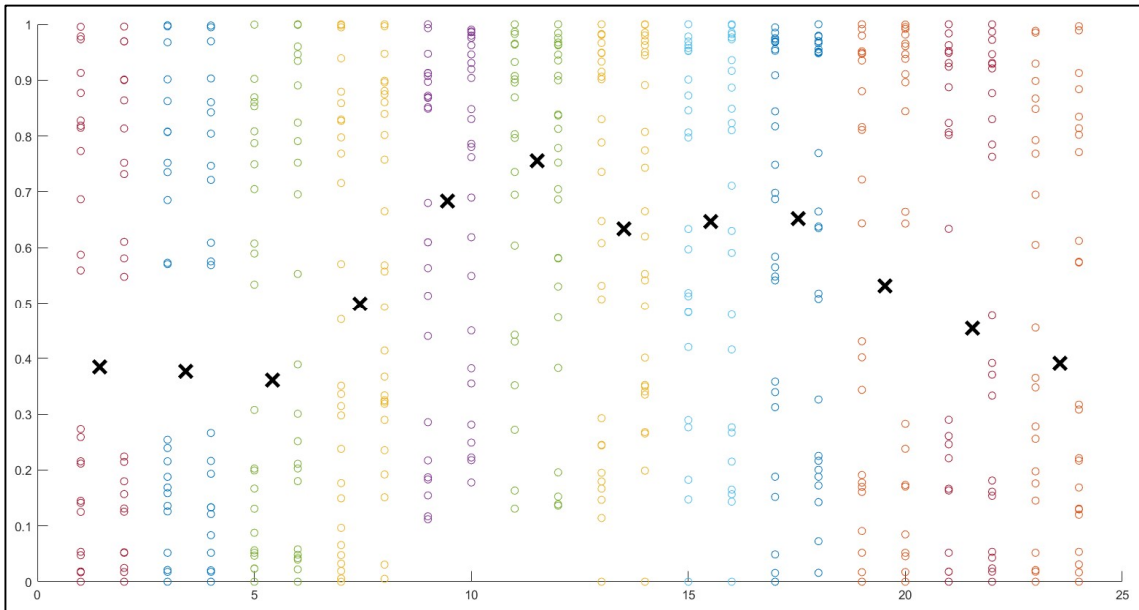


Figura 2.9 Gráfica de FKM para los datos del lunes de industriales

2.4 Cálculo de los índices de adecuación

Una parte importante luego de haber realizado el algoritmo de agrupación es evaluar el desempeño de este, y para ello existen varios índices que permiten conocer el rendimiento del algoritmo, sin embargo, para este trabajo únicamente se utilizarán dos indicadores conocidos como MIA y CDI. El índice MIA evalúa la dispersión de los datos, es decir como estos se acumulan alrededor de los centros, como se puede observar en la Figura 2.9 mientras que el índice CDI evalúa la posición de los centroides, y para ambos casos entre menor sea el valor obtenido mejor percepción se tiene del funcionamiento del algoritmo.

Para calcular estos índices se utilizan las ecuaciones (4) y (5) presentadas al inicio de este trabajo en la sección del marco teórico, donde se explicó que para el índice MIA era necesario calcular la distancia promedio de los todos los puntos de datos de un clúster al centroide de este mientras que para el índice CDI se calcula la distancia entre dos centros cualesquiera, y en este caso se ha seleccionado a los dos primeros centroides debido a

que su posición en la gráfica es la más estable. Así pues, para realizar este proceso se creó una función llamada “MIAyCDI” y se la incluyó dentro del código desarrollado; esta función necesita como datos de entrada la matriz que contenga los datos correspondientes a un día de la semana con su grupo de consumo (“lunes_ind”), también la matriz de partición difusa entregada por la función “fcm” que contiene los grados de pertenencia de los datos (“U_ind_L”), el número de centroides (N) y la matriz que contiene a todos los centroides (“C_ind_L”). Con todos estos datos la función realiza las operaciones necesarias para entregar como resultados los valores correspondientes a los índices MIA y CDI. A continuación, se presenta un pseudocódigo que muestra los pasos que sigue la función para obtener una respuesta.

Ingresar a la función la matriz que contiene los datos respecto de un día de la semana y un tipo de consumidor, la matriz que contiene los grados de pertenencia, el número de centroides y la matriz que contiene los centroides calculados.

Calcular la distancia promedio de todos los datos de un clúster al centroide de este.

Reemplazar la distancia promedio en la ecuación (1) y obtener el índice MIA.

Calcular la distancia entre dos centroides.

Reemplazar la distancia en la ecuación (2) y obtener el índice CDI.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Resultados

En esta sección se presentarán las gráficas obtenidas a través del algoritmo FKM y un análisis del comportamiento de estas para conocer si el método es viable y logra reflejar el consumo de potencia dependiendo del día y el tipo de consumidor. Se mostrarán tres gráficas por cada tipo de consumidor, una que abarcará la semana laboral, es decir, el consumo de potencia de lunes a viernes, otra enfocada para los sábados y una última específica para los domingos, sin embargo, antes de continuar es necesario aclarar que la gráfica para la semana laboral se obtiene a través de unir todos los datos de lunes hasta viernes y con ello obtener una curva representativa de todos esos días.

Empezando por los consumidores industriales, la Figura 3.1 muestra la curva representativa de carga para la semana laboral donde se puede observar que el consumo de potencia empieza a partir de las 05:00 y esto puede ser debido a que comienzan los preparativos de las instalaciones hasta alrededor de las 08:00 que coincide con la hora de entrada de los trabajadores y comienza la producción representada por una cresta cuyo

pico de potencia es de 0.74, luego hay un valle que concuerda con la hora de almuerzo (alrededor de las 13:00) y donde el consumo baja un poco para luego retomar la producción con una nueva cresta cuyo pico también es de 0.74 y a partir de las 17:00 el consumo de potencia comienza a decaer debido a que termina el horario laboral y los trabajadores retornan a sus hogares a descansar.

Los valores de los índices MIA y CDI son de 1.57 y 0.92 respectivamente. Se puede notar que el comportamiento de la curva coincide con el ejemplo presentado en la Figura 1.2 para consumidores industriales que caracterizaba a la curva con dos crestas y un valle entre ellas, similar a lo obtenido en esta curva para la semana laboral.

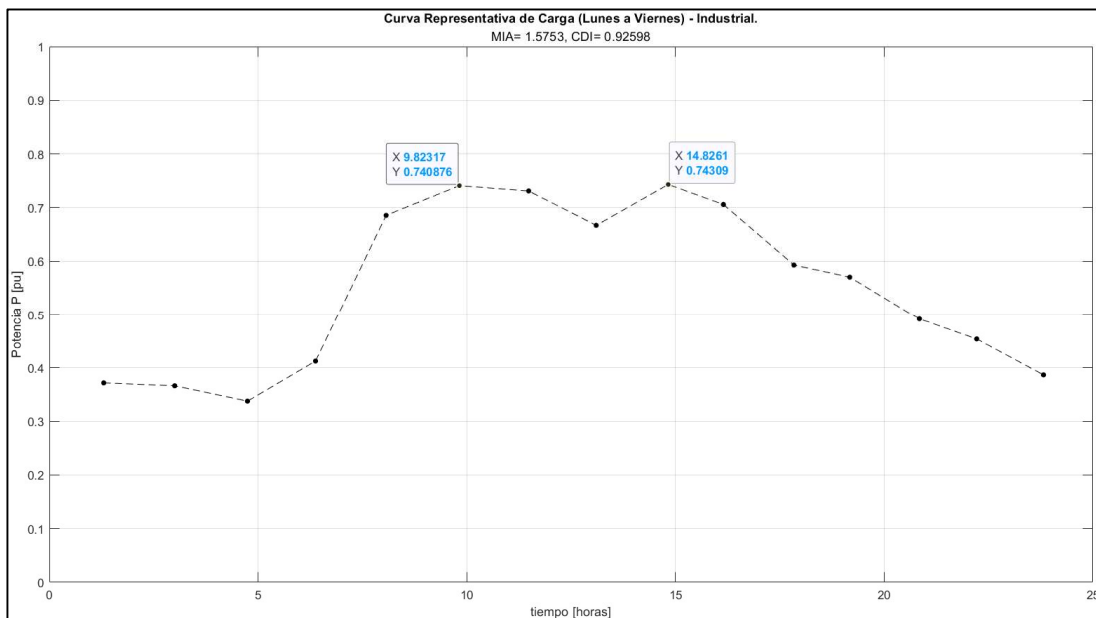


Figura 3.1 Curva de carga representativa de lunes a viernes para industriales

Siguiendo con los consumidores industriales, la Figura 3.2 muestra la curva representativa de carga para los sábados donde se puede observar que el consumo de potencia aumenta igualmente alrededor de las 05:00 pero a diferencia de la Figura 3.1 esta únicamente cuenta con una sola cresta cuyo pico de potencia es de 0.77, por lo que se puede interpretar que estos días (sábados) se trabaja únicamente medio tiempo, ya que, a partir de las 14:00 la curva decae, o en otras palabras la potencia disminuye a lo largo del día hasta empezar una nueva jornada. Por otra parte, el índice MIA tiene un valor de 1.03 mientras que el CDI es igual a 0.62.

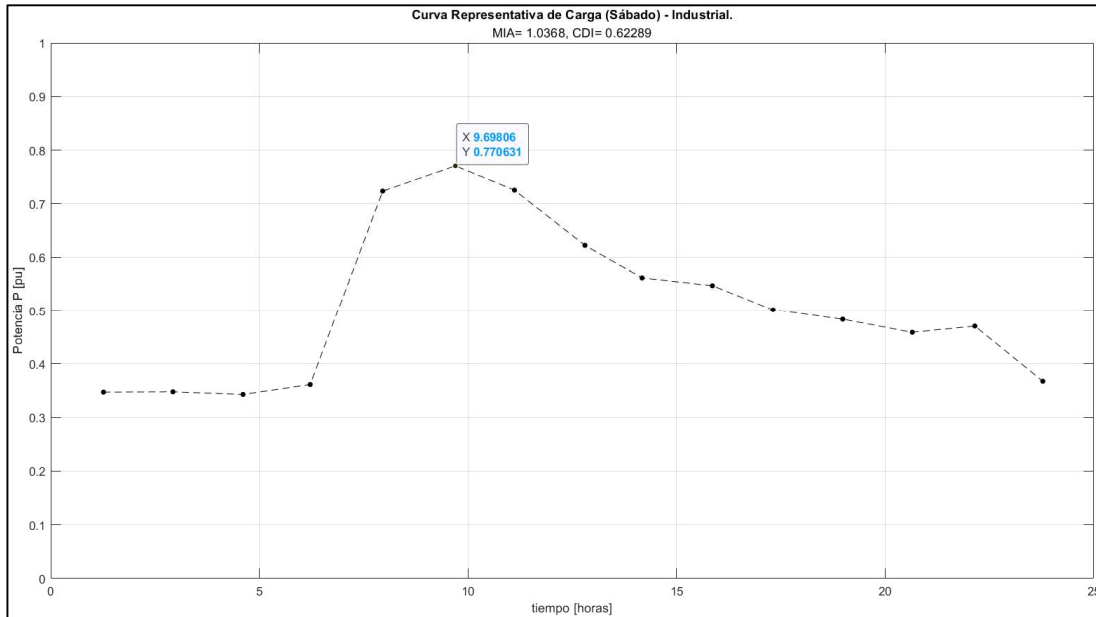


Figura 3.2 Curva de carga representativa para sábado de industriales

La Figura 3.3 muestra la curva representativa de carga para los domingos y al igual que las dos curvas anteriores esta también empieza su consumo de potencia a partir de las 05:00 además se puede observar que tiene el mismo patrón que la Figura 3.1, es decir, con dos crestas y un valle, sin embargo en esta curva hay otra característica y es con respecto a los picos, ya que el primero es de 0.69 mientras que el segundo es de 0.67 y esto tiene más coherencia con la Figura 1.2 donde se puede observar que la primera cresta (o montaña) es más alta que la segunda, lo cual tiene sentido ya que en la mañana la producción es mucho mayor que en la tarde porque luego del almuerzo la productividad de los trabajadores decae. Los valores para los índices MIA y CDI son de 1.13 y 0.59 respectivamente.

Otro detalle de la Figura 3.2 es que abarca una mayor extensión de tiempo, por lo que se puede interpretar que en el periodo que se tomaron las mediciones el grupo industrial estuvo haciendo horas extras los domingos porque la segunda cresta comienza a decaer a partir de las 20:00.

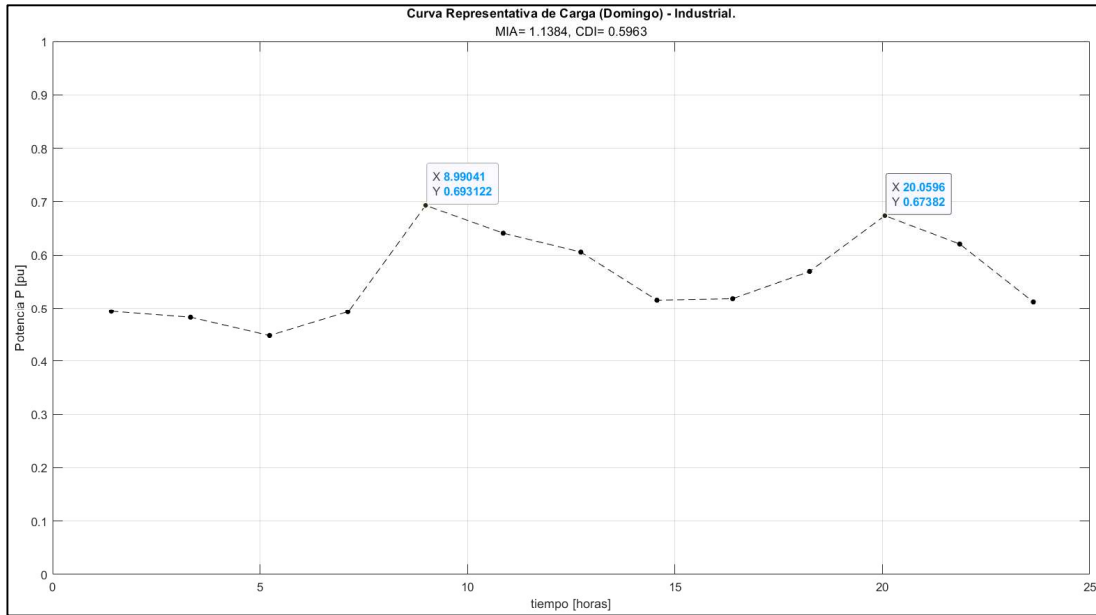


Figura 3.3 Curva de carga representativa para domingo de industriales

Ahora se presentarán las curvas correspondientes a los consumidores comerciales, empezando por la Figura 3.4 que muestra a la curva representativa de carga para la semana laboral (lunes a viernes) donde se puede observar que el consumo de potencia empieza a partir de las 05:00 y comparando nuevamente con la Figura 1.2 se puede notar que también existen dos crestas y un valle similar al caso de los industriales con la diferencia de que la primera cresta es menor que la segunda, y en este caso el primer pico es de 0.75 mientras que el segundo es de 0.79, esto puede explicarse debido a que las personas generalmente realizan sus compras en horas de la tarde luego de salir del trabajo o almorzar por ello se puede notar el incremento en el consumo de potencia.

Otra particularidad es que parece que el sector comercial trabaja más horas ya que el consumo de potencia empieza a decaer a partir de las 20:00 lo que se puede interpretar como que se trabajó en horas extras o que en el periodo de medición hubo una gran carga de trabajo que obligó a las empresas a extender su horario laboral para poder cumplir con la demanda que se presentaba. Los índices MIA y CDI para esta curva de consumidores comerciales es de 1.58 y 0.96 respectivamente.

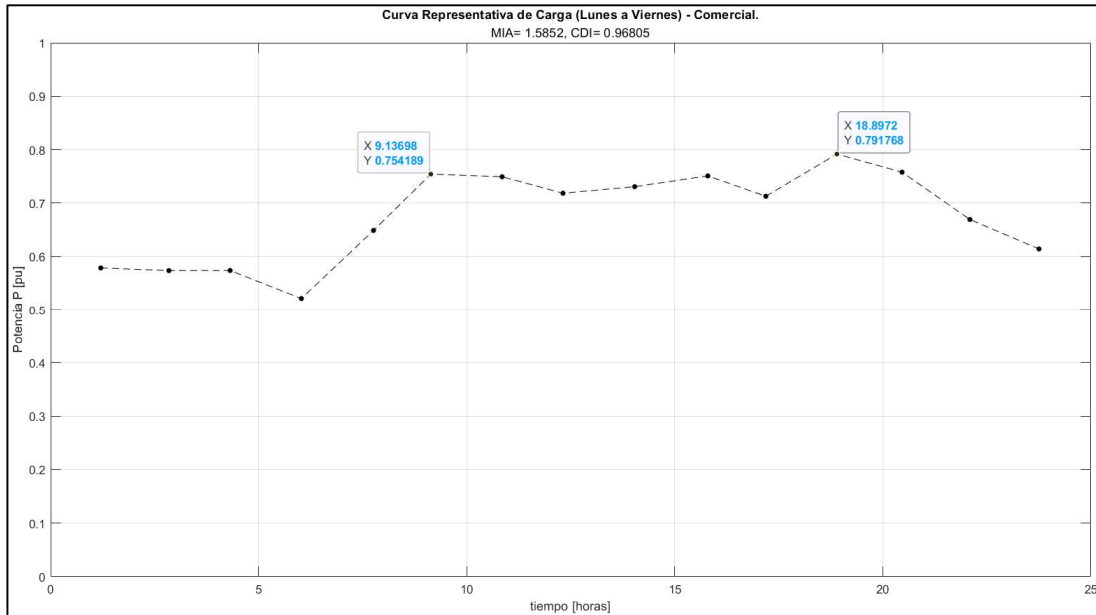


Figura 3.4 Curva de carga representativa de lunes a viernes para comerciales

Continuando con los consumidores comerciales, la Figura 3.5 muestra a la curva representativa de carga para los sábados y se puede observar que el comportamiento es similar a la Figura 3.3 porque el consumo de potencia comienza a partir de las 05:00, también se mantiene la forma de dos crestas cuyos picos son de 0.77 y 0.81 respectivamente, aunque en esta ocasión se puede notar más la diferencia entre las dos y el valle que se forma entre ellas, también de igual manera el primer pico es mayor que el segundo manteniendo esa conducta de consumo y además la curva sigue abarcando un mayor periodo de tiempo, ya que nuevamente la potencia comienza a decaer a partir de las 20:00 por lo que parece que el fin de la jornada laboral se encuentra alrededor de esta hora. Y los valores para los índice MIA y CDI son de 1.24 y 0.62 respectivamente.

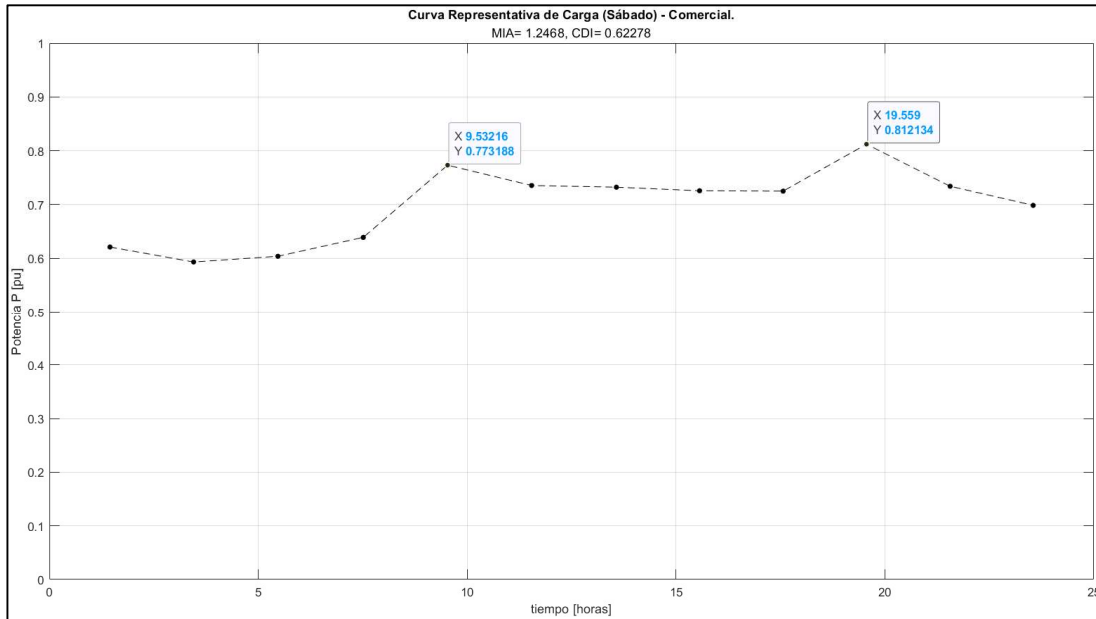


Figura 3.5 Curva de carga representativa para sábado de comerciales

Y para finalizar con los consumidores comerciales la Figura 3.6 muestra la curva representativa de carga para los domingos donde se puede observar que en este caso el consumo de potencia comienza alrededor de las 07:00 y al igual que en las anteriores dos figuras esta también cuenta con dos crestas cuyos picos son de 0.73 y 0.81 manteniendo así el patrón de que el primer pico es mayor que el segundo, pero la característica que llama más la atención es que luego de las 20:00 el consumo no decae mayormente sino que es leve; además en las tres curvas de los comerciales se puede notar que el consumo de potencia en horas de la madrugada es muy alto, incluso mayor que para los industriales, lo cual es atípico porque debería ser al revés, sin embargo, puede haber una explicación a este fenómeno.

Hay que recordar que la base de datos proviene de Azogues que es una ciudad que se encuentra situada en la provincia de Cañar, y al investigar datos económicos a través del portal del INEC y que se puede observar en la Figura 3.7 se encontró que el comercio para esta ciudad representa el 62.63% de toda la actividad económica de la ciudad mientras que la parte industrial únicamente representa un 6.638%, es decir casi 10 veces menos por lo que tendría sentido que el comercio tenga un mayor consumo de potencia [22].

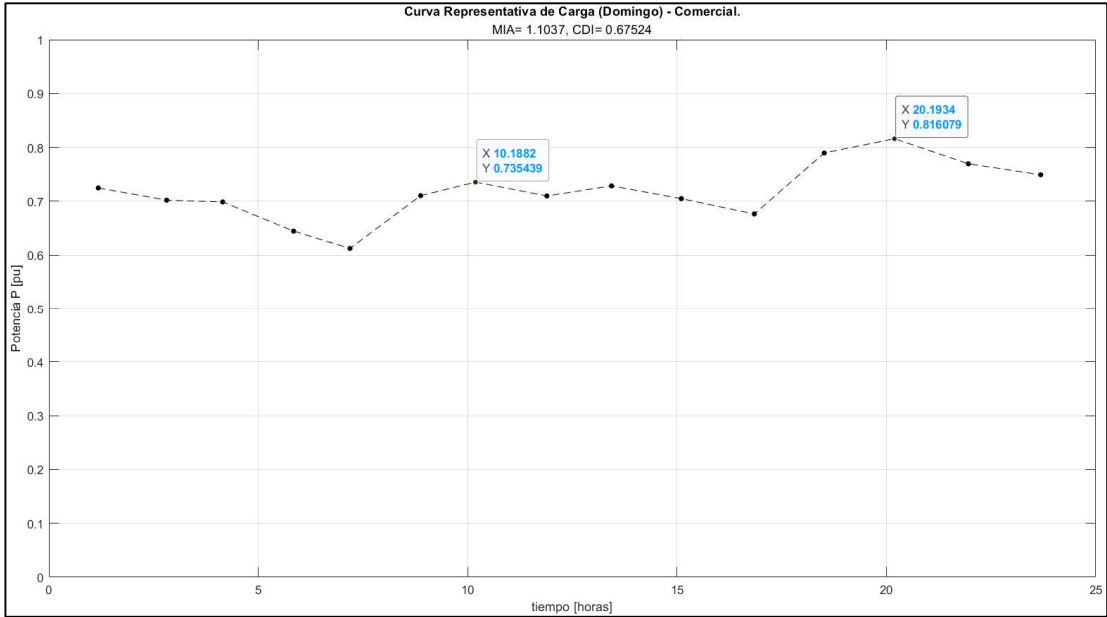


Figura 3.6 Curva de carga representativa para domingo de comerciales

Además en este 62.63% de comercio el mayor porcentaje se lo lleva el comercio al por menor con un 75.38% y a su vez esta sección tiene sus principales actividades en venta al por menor de productos farmacéuticos y medicinales, cosméticos y artículos con un 26.91% seguido de la venta al por menor en comercios no especializados con predominio en la venta de alimentos con 20.35% [22]; así pues el comercio tiene sus bases en medicamentos y alimentos, que no solamente se venden dentro de la ciudad sino que se exporta a las demás provincias por lo que necesitan trabajar con un horario más amplio, como se puede observar en las tres curvas y en base a turnos que sería la única manera de explicar el consumo de potencia en las horas de madrugada.

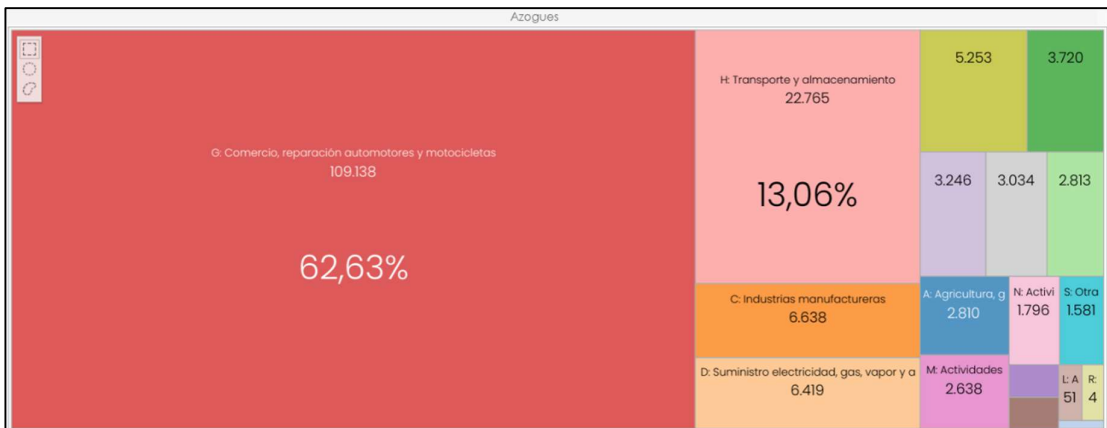


Figura 3.7 Actividad económica de la ciudad Azogues segmentada [22]

En otras palabras, el comercio al ser la base de la economía de la ciudad no varía en todo el año y se puede decir que las curvas industriales no siempre deben ser mayores que la de los comerciales, sino que su comportamiento depende de la ubicación y la economía de la ciudad.

Finalmente se muestran las curvas para los consumidores residenciales, comenzando por la Figura 3.8 que pertenece a la semana laboral y donde se puede observar que el consumo de potencia se eleva a partir de las 05:00, cuando las personas se despiertan para alistarse para ir a sus trabajos o lugares de estudio (escuela, colegio o universidad), existiendo un pequeño pico alrededor de las 07:00, que corresponden al uso de varios aparatos electrodomésticos como microondas, sanducheras, televisión entre otros. A continuación, el consumo de potencia decae por el inicio de la jornada laboral pero después la potencia empieza a crecer nuevamente a partir de las 15:00 que es cuando las personas comienzan a llegar de sus escuelas, trabajo y ya encienden los aparatos de la casa como la computadora para hacer tareas o simplemente entretenimiento. Esta tendencia se mantiene y sigue aumentando hasta tener un pico alrededor de las 19:00 que es cuando ya todas las luces de las casas están encendidas y la familia se sienta a cenar. Posterior a esta hora el consumo desciende porque ya se disponen para dormir.

Los valores de los índices para la curva son de 2.14 y 1.30 respectivamente, rompiendo con la tendencia de las anteriores gráficas en las que no se obtuvieron valores tan altos especialmente en lo que respecta al índice CDI. Esto puede deberse a que para obtener la curva de la semana laboral se combinan todos los datos de los cinco días por lo que la densidad de puntos combinado con que muchos de ellos tienen valor cero o negativo hace que la dispersión no sea de la mejor calidad y esto se refleja en los valores obtenidos en los índices de adecuación.

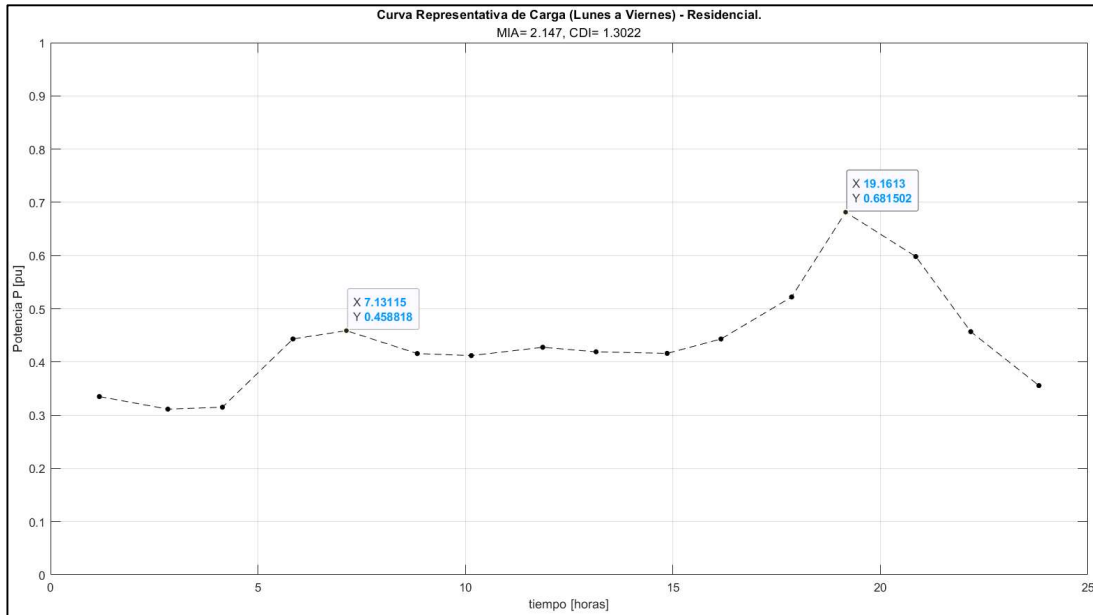


Figura 3.8 Curva de carga representativa de lunes a viernes para residenciales

Continuando con los consumidores residenciales, la Figura 3.9 muestra a la curva representativa de carga para los sábados donde puede observar que el consumo de potencia se eleva nuevamente a partir de las 05:00 debido a que podría haber personas que trabajan hasta estos días y como no son muchas esto se refleja como un pico de potencia no muy pronunciado pero luego de las 10:00 el consumo vuelve a aumentar y existe un nuevo pico más evidente y que muestra que ya se está usando los aparatos domésticos para desayunar. Posteriormente la potencia se mantiene relativamente en el mismo consumo hasta que nuevamente a las 15:00 el consumo comienza a subir porque las personas están regresando de sus trabajos y encienden sus aparatos. Este crecimiento se mantiene hasta las 19:00 cuando se presenta el pico máximo del día y que corresponde a la cena, pero luego de esta hora la potencia disminuye porque ya se preparan para dormir y empezar un nuevo día. Los valores obtenidos para los índices MIA y CDI para esta curva son de 1.48 y 0.9 respectivamente, por lo que los índices vuelven a estar en la tendencia que se tenía desde que se presentaron las gráficas.

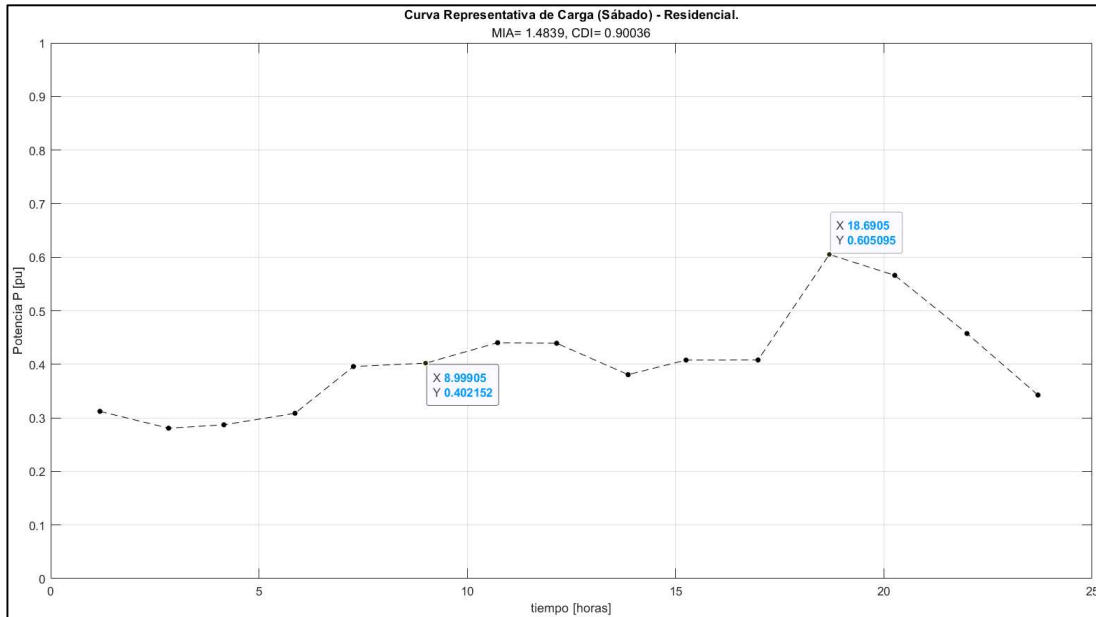


Figura 3.9 Curva de carga representativa para sábado de residenciales

Y por último la Figura 3.10 muestra la curva representativa de carga para los domingos de los consumidores residenciales, y en ella se puede observar que el consumo otra vez empieza a partir de las 05:00 que puede explicarse porque algunas personas tienen trabajo y otras generalmente usan el domingo para realizar limpieza de los hogares, como sea el pico de potencia de la mañana se encuentra alrededor de las 09:00 que es el desayuno de las personas para después mantener la potencia constante, esto parece ser exclusivo de los domingos y tendría sentido ya que al estar todos en casa el consumo luego del desayuno no debería bajar porque los aparatos como televisión, computadoras y demás dispositivos se mantienen prendidos. A las 15:00 el consumo empieza a aumentar hasta llegar al pico máximo del día que esta alrededor de las 19:00, y con la cena se marca el momento en que la potencia decae porque al igual que los otros días de la semana las persona ya se preparan para dormir y empezar nuevamente la semana.

Además, para una mejor visualización de los índices de adecuación, la Tabla 3.1 contiene todos los valores por tipo de curva y el período que abarcan. Con esto se puede notar que tanto el índice MIA como CDI son similares exceptuando los consumidores residenciales que escapan de la tendencia y puede ser debido a que la base de datos es mayor y por lo tanto más contaminada por datos atípicos que provocan que los índices sean más altos.

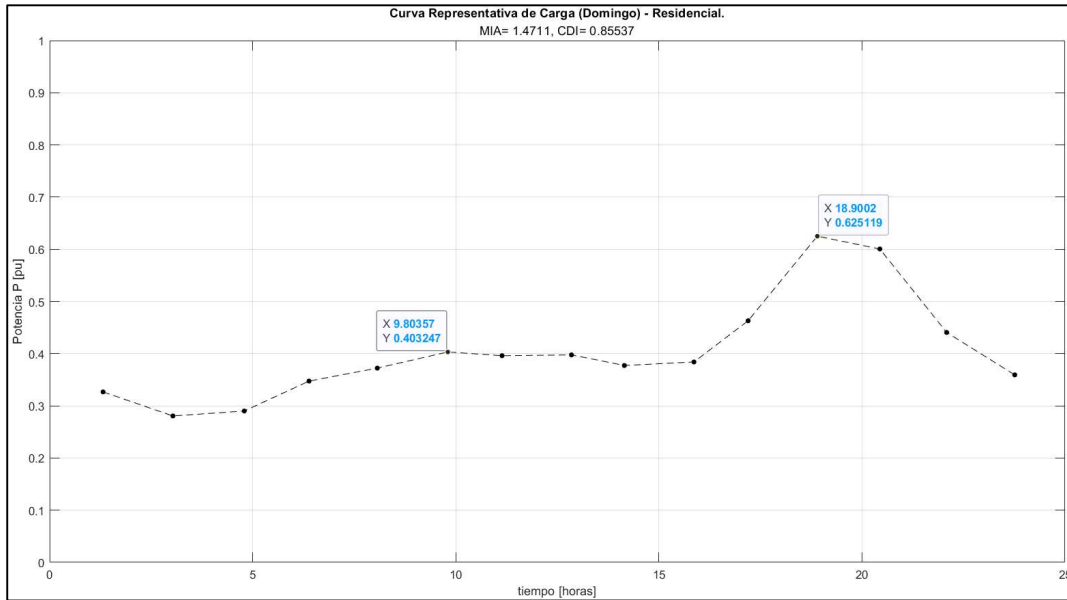


Figura 3.10 Curva de carga representativa para domingo de residenciales

Tabla 3.1 Índices de adecuación por tipo de curva

Tipo de curva	Período [días]	MIA	CDI
Industrial	Lunes a viernes	1.5753	0.9260
Comercial	Lunes a viernes	1.5852	0.9681
Residencial	Lunes a viernes	2.147	1.3022
Industrial	Sábado	1.0368	0.6229
Comercial	Sábado	1.2468	0.6228
Residencial	Sábado	1.4839	0.9004
Industrial	Domingo	1.1384	0.5963
Comercial	Domingo	1.1037	0.6752
Residencial	Domingo	1.4711	0.8554

4 CONCLUSIONES

- El algoritmo fuzzy k-means usa lógica difusa permitiendo que los agrupamientos sean menos restrictivos porque los datos ya no pertenecen a un solo clúster, sino que pueden pertenecer a varios por lo que los centroides tienden a tener una mejor posición, lo cual se refleja en las gráficas obtenidas en este trabajo.
- Se tiene que realizar un tratamiento de la base de datos porque esto ayuda a conseguir resultados más confiables, ya que, al eliminar los datos atípicos como celdas vacías, valores de potencia cero o negativos, y en general datos que no están en coherencia con el resto de la base de datos se asegura que el algoritmo pueda trabajar únicamente con datos limpios y válidos que facilita el cálculo de los centroides para formar las curvas representativas de carga.
- Generalmente se espera que el consumo de potencia del grupo industrial sea mayor al del grupo comercial, sin embargo, en este trabajo se pudo visualizar que esto no necesariamente se cumple, sino que más bien responde a la ubicación y actividad económica de la misma. En otras palabras, se sugiere que la actividad económica rige el consumo de potencia porque se pudo evidenciar que la locación en donde se tomó las mediciones el comercio era mucho mayor que la parte industrial, y este comportamiento se vio reflejado en las curvas de carga.
- En las curvas de los consumidores residenciales existe un patrón en el pico de la noche que se da todos los días de la semana (lunes a domingo) a las 7 pm y esto podría ser utilizado para encontrar tendencias de consumo que servirían en el momento del despacho de la energía, ya que si se conoce los horarios en que las personas consumen la potencia se podría realizar una mejor organización para evitar posibles fallas al momento de alimentar las cargas.
- Los valores obtenidos para los índices MIA y CDI son similares, exceptuando el caso de la curva de consumidores residenciales de la semana laboral, sin embargo, para el resto de las curvas todos reflejan que el algoritmo funciona y entrega buenos resultados.
- El método de cálculo para el índice CDI a través de tomar los dos primeros centroides fue porque al intentar otras técnicas el valor obtenido era extremadamente bajo, y aunque esto debería ser positivo la realidad no lo es porque la base de datos aun contenía varios datos erróneos por lo que un índice muy bajo no reflejaba los resultados que se estaban consiguiendo, por ello se

decidió presentar los valores del índice CDI que parecen estar más en concordancia con las curvas, y el índice MIA.

- Existe mucha información bibliográfica sobre el algoritmo FKM enfocado en la obtención de curvas representativas ya que parece que es muy popular por su “facilidad” de incorporarla en un algoritmo computacional y obtener resultados coherentes con la realidad.
- El software MATLAB es una herramienta computacional usada en varios campos porque permite desarrollar diferentes tipos de algoritmos como la técnica de agrupamiento FKM e incluso por medio de la instalación de toolbox utilizar funciones prediseñadas que realizan un trabajo igual de eficiente. Y además, fórmulas matemáticas complejas como las que describen los índices de adecuación pueden ser resueltas rápidamente gracias al enfoque matricial que permite resolver ecuaciones de manera más óptima.
- Las curvas obtenidas muestran comportamientos similares a la realidad por lo que se puede decir que la técnica de agrupamiento FKM es capaz de asociar los datos de forma correcta para entregar curvas que representen el consumo de potencia de diferentes tipos de clientes.

5 RECOMENDACIONES

- Tratar de asegurarse que en la base de datos las mediciones no tengan valores distorsionados y que sea de la mejor calidad posible con el fin de que los resultados que se vayan a obtener sean coherentes con lo que se está buscando.
- Comparar las curvas obtenidas a través del algoritmo FKM con otras obtenidas por las empresas eléctricas que sirvan como referencia para observar que tan bien se acopla el comportamiento de las curvas con las que se utiliza en la industria.
- Utilizar una base de datos más grande con el fin de probar el alcance del programa y observar si los resultados que entrega tienen un comportamiento coherente con la realidad.
- Si el programa tiene buenos resultados representando los comportamientos de los distintos consumidores, tratar de trasladarlo a otro lenguaje de programación que tenga una mayor capacidad de aplicación en la industria.

- Revisar los valores obtenidos para los índices de adecuación (MIA y CDI) e interpretarlos desde una perspectiva más técnica, así como también corroborar y mejorar las técnicas de cálculo utilizadas.

6 REFERENCIAS BIBLIOGRÁFICAS

- [1] S. Balasubramanian and P. Balachandra, "Characterising electricity demand through load curve clustering: A case of Karnataka electricity system in India," *Comput. Chem. Eng.*, vol. 150, p. 107316, Jul. 2021, doi: 10.1016/J.COMPCHEMENG.2021.107316.
- [2] P. T. T. Binh, N. H. Ha, T. C. Tuan, and L. D. Khoa, "Determination of representative load curve based on Fuzzy K-Means," *PEOCO 2010 - 4th Int. Power Eng. Optim. Conf. Progr. Abstr.*, pp. 281–286, 2010, doi: 10.1109/PEOCO.2010.5559257.
- [3] G. Chicco, R. Napoli, F. Piglione, P. Postolache, M. Scutariu, and C. Toader, "Load pattern-based classification of electricity customers," *IEEE Trans. Power Syst.*, vol. 19, no. 2, pp. 1232–1239, 2004, doi: 10.1109/TPWRS.2004.826810.
- [4] J. Hernández Orallo, M. J. Ramírez Quintana, and C. Ferri Ramírez, "Introducción a la minería de datos.," p. 676, 2004.
- [5] P. Tan, M. Steinbach, and V. Kumar, "Introduction to Data Mining Tan Steinbach Kumar First Edition," 2014, Accessed: May 12, 2022. [Online]. Available: www.pearsoned.co.uk.
- [6] J. Gironés, J. Casas, J. Minguillón, and R. Caihuelas, "Minería de datos: Modelos y Algoritmos.," *Editor. UOC*, no. 1, pp. 1–271, 2017, Accessed: May 12, 2022. [Online]. Available: https://www.cambridge.org/core/product/identifier/CBO9781139058452A007/type/book_part.
- [7] C. C. Aggarwal, "Data Mining: The Textbook," *Springer Int. Publ.*, pp. 285–344, 2015, doi: 10.1007/978-3-319-14142-8_10.
- [8] P. Bhatia, "Data mining and data warehousing : principles and practical techniques," *Cambridge Univ. Press*, pp. 17–20, 2019.
- [9] V. K. Dehariya, S. K. Shrivastava, and R. C. Jain, "Clustering of image data set using k-means and fuzzy k-means algorithms," *Proc. - 2010 Int. Conf. Comput. Intell. Commun. Networks, CICN 2010*, pp. 386–391, 2010, doi: 10.1109/CICN.2010.80.

- [10] J. ROJAS DIAZ Ing Electrónico and R. MORENO LAVERDE en Administración Económica Financiera Profesor Asistente, "Técnicas de lógica difusa aplicadas a la minería de datos," *Sci. Tech.*, vol. 3, no. 40, 2008, doi: 10.22517/23447214.3095.
- [11] J. D. Juárez Cervantes, *Sistemas de Distribución de Energía Eléctrica*, vol. 52, no. 1. 2002.
- [12] T. Gönen and T. Gönen, "Electric power distribution engineering," p. 1021, 2014.
- [13] H. Moore, "MATLAB for engineers," p. 656, 2012.
- [14] A. ; Tureczek, S. ; Nielsen, H. Madsen, A. Tureczek, and S. Nielsen, "General rights Electricity consumption clustering using smart meter data Electricity Consumption Clustering Using Smart Meter Data," vol. 11, no. 4, doi: 10.3390/en11040859.
- [15] "(Not recommended) Read Microsoft Excel spreadsheet file - MATLAB xlsread - MathWorks América Latina." <https://la.mathworks.com/help/matlab/ref/xlsread.html> (accessed Jun. 09, 2022).
- [16] "Compare strings - MATLAB strcmp - MathWorks América Latina." <https://la.mathworks.com/help/matlab/ref/strcmp.html> (accessed Jun. 11, 2022).
- [17] "Convert strings to double precision values - MATLAB str2double - MathWorks América Latina." <https://la.mathworks.com/help/matlab/ref/str2double.html> (accessed Jun. 11, 2022).
- [18] "Fuzzy c-means clustering - MATLAB fcm - MathWorks América Latina." <https://la.mathworks.com/help/fuzzy/fcm.html> (accessed Jun. 12, 2022).
- [19] "Fuzzy Logic Toolbox Documentation - MathWorks América Latina." https://la.mathworks.com/help/fuzzy/index.html?s_tid=CRUX_lftnav (accessed Jun. 12, 2022).
- [20] "Maximum elements of an array - MATLAB max - MathWorks América Latina." <https://la.mathworks.com/help/matlab/ref/max.html> (accessed Jun. 14, 2022).
- [21] "Encontrar índices y valores de elementos distintos a cero - MATLAB find - MathWorks América Latina." <https://la.mathworks.com/help/matlab/ref/find.html> (accessed Jun. 14, 2022).
- [22] "Visualizador de Estadísticas Empresariales2020 | Tableau Public." <https://public.tableau.com/app/profile/instituto.nacional.de.estad.stica.y.censos.inec./viz/VisualizadordeEstadsticasEmpresariales2020/Dportada> (accessed Jun. 20,

2022).

7 ANEXOS

ANEXO I: BASE DE DATOS

Archivo Excel con nombre: BDEEA.xlsx

ANEXO II: SCRIPT DE MATLAB

Trabajo de Integración Curricular

Kevin Jaramillo

Limpieza

```
clc;
clear;
close all;
```

Tratamiento de los datos

```
datos_aux=readmatrix("BDEEA.xlsx","OutputType","duration");
tiempo=hours(datos_aux(:,3));%tiempo en horas
[f,c]=size(tiempo);
for i=1:f%para cambiar el valor de 0 horas a 24 horas.
    if tiempo(i,1)==0
        tiempo(i,1)=24;
    end
end
datos_aux2=readmatrix("BDEEA.xlsx","OutputType","double");
pot_prom_pu=datos_aux2(:,9);
dia_sem=datos_aux2(:,10);
datos_aux3=readmatrix("BDEEA.xlsx","OutputType","string");
grupo=datos_aux3(:,12);
datos_val=[tiempo pot_prom_pu dia_sem grupo];%formacion de la matriz que
se va
%utilizar para el analisis
```

Datos Lunes Industrial

```
[filas,columnas]=size(datos_val);
aux=1;
for i=1:filas%segmento para el dia lunes, la hora con su valor de potencia
    for j=1:columnas
```

```

        if strcmp(datos_val(i,j),'1') &&
strcmp(datos_val(i,j+1),'Industrial')
            hora_L_ind(aux,1)=datos_val(i,1);
            pt_L_ind(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_L_ind);
B=str2double(pt_L_ind);
lunes_ind=[A B];%forma la matriz que corresponde al dia lunes para
%el grupo industrial
%Algoritmo fuzzy k means
%C contiene los centros calculados para cada cluster mientras que U
contiene
%la matriz de partición difusa calculada, que indica el grado de
%pertenencia de cada punto dentro de cada cluster
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_ind_L,U_ind_L]=fcm(lunes_ind,N,options);%(fin)
%Para clasificar cada punto en el cluster para el que tiene
%el mayor grado de pertenencia.
maxU_ind_L = max(U_ind_L);
index_ind_L = cell(N,1);
for k=1:N
    index_ind_L{k} = find(U_ind_L(k,:) == maxU_ind_L);
end%(fin de la clasificacion)
%Para graficar los resultados de la agrupacion
figure
hold on
for k=1:N
    plot(lunes_ind(index_ind_L{k},1),lunes_ind(index_ind_L{k},2),'o');
    plot(C_ind_L(k,1),C_ind_L(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
hold off%(fin de la graficacion)

```

Datos Martes Industrial

```

aux=1;
for i=1:filas%segmento para el dia martes, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'2') &&
strcmp(datos_val(i,j+1),'Industrial')
            hora_M_ind(aux,1)=datos_val(i,1);
            pt_M_ind(aux,1)=datos_val(i,2);

```

```

        aux=aux+1;
    end
end
end
A=str2double(hora_M_ind);
B=str2double(pt_M_ind);
martes_ind=[A B];%forma la matriz que corresponde al dia martes para
%el grupo industrial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_ind_M,U_ind_M]=fcm(martes_ind,N,options);
maxU_ind_M = max(U_ind_M);
index_ind_M = cell(N,1);
for k=1:N
    index_ind_M{k} = find(U_ind_M(k,:) == maxU_ind_M);
end
figure
hold on
for k=1:N
    plot(martes_ind(index_ind_M{k},1),martes_ind(index_ind_M{k},2),'o');
    plot(C_ind_M(k,1),C_ind_M(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
hold off

```

Datos Miercoles Industrial

```

aux=1;
for i=1:filas%segmento para el dia miercoles, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'3') &&
strcmp(datos_val(i,j+1),'Industrial')
            hora_W_ind(aux,1)=datos_val(i,1);
            pt_W_ind(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_W_ind);
B=str2double(pt_W_ind);
miercoles_ind=[A B];%forma la matriz que corresponde al dia miercoles para
%el grupo industrial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_ind_W,U_ind_W]=fcm(miercoles_ind,N,options);
maxU_ind_W = max(U_ind_W);

```

```

index_ind_W = cell(N,1);
for k=1:N
    index_ind_W{k} = find(U_ind_W(k,:) == maxU_ind_W);
end
figure
hold on
for k=1:N

plot(miercoles_ind(index_ind_W{k},1),miercoles_ind(index_ind_W{k},2),'o');
    plot(C_ind_W(k,1),C_ind_W(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
hold off

```

Datos Jueves Industrial

```

aux=1;
for i=1:filas%segmento para el dia jueves, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'4') &&
strcmp(datos_val(i,j+1),'Industrial')
            hora_J_ind(aux,1)=datos_val(i,1);
            pt_J_ind(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_J_ind);
B=str2double(pt_J_ind);
jueves_ind=[A B];%forma la matriz que corresponde al dia jueves para
%el grupo industrial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_ind_J,U_ind_J]=fcm(jueves_ind,N,options);
maxU_ind_J = max(U_ind_J);
index_ind_J = cell(N,1);
for k=1:N
    index_ind_J{k} = find(U_ind_J(k,:) == maxU_ind_J);
end
figure
hold on
for k=1:N
    plot(jueves_ind(index_ind_J{k},1),jueves_ind(index_ind_J{k},2),'o');
    plot(C_ind_J(k,1),C_ind_J(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
hold off

```

Datos Viernes Industrial

```

aux=1;
for i=1:filas%segmento para el dia viernes, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'5') &&
strcmp(datos_val(i,j+1),'Industrial')
            hora_V_ind(aux,1)=datos_val(i,1);
            pt_V_ind(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_V_ind);
B=str2double(pt_V_ind);
viernes_ind=[A B];%forma la matriz que corresponde al dia viernes para
%el grupo industrial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_ind_V,U_ind_V]=fcm(viernes_ind,N,options);
maxU_ind_V = max(U_ind_V);
index_ind_V = cell(N,1);
for k=1:N
    index_ind_V{k} = find(U_ind_V(k,:) == maxU_ind_V);
end
figure
hold on
for k=1:N
    plot(viernes_ind(index_ind_V{k},1),viernes_ind(index_ind_V{k},2),'o');
    plot(C_ind_V(k,1),C_ind_V(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
hold off

```

Datos de lunes a viernes de Industrial

```

semana_ind=[lunes_ind;martes_ind;
    miercoles_ind;jueves_ind;viernes_ind];
%Algoritmo fuzzy k means
N_ind_sem=15;%# de centros
options=[NaN NaN NaN 0];
[C_ind_sem,U_ind_sem]=fcm(semana_ind,N_ind_sem,options);
maxU_ind_sem = max(U_ind_sem);
index_ind_sem = cell(N_ind_sem,1);
for k=1:N_ind_sem
    index_ind_sem{k} = find(U_ind_sem(k,:) == maxU_ind_sem);
end
figure
hold on

```

```

for k=1:N_ind_sem

plot(semana_ind(index_ind_sem{k},1),semana_ind(index_ind_sem{k},2), 'o');

plot(C_ind_sem(k,1),C_ind_sem(k,2), 'xk', 'MarkerSize',15, 'LineWidth',3);
end
hold off
aux2=1;
M_ind_P=zeros(N_ind_sem,1);%inicializo el vector para que se pueda
actualizar el tamaño
matriz_C=C_ind_sem;%tomo todos los centros de L a V de los industriales.
M_ind_t=sort(matriz_C(:,1));%ordeno de menor a mayor el tiempo
for i=1:N_ind_sem
    for j=1:N_ind_sem
        if M_ind_t(i,1)==matriz_C(j,1)
            M_ind_P(aux2,1)=matriz_C(j,2);%emparejo cada tiempo con su
respectiva potencia
            aux2=aux2+1;
        end
    end
end
grafica_ind_sem=[M_ind_t M_ind_P];%formo la matriz para graficar la curva
de carga
figure
plot(grafica_ind_sem(:,1),grafica_ind_sem(:,2), '--. k', 'MarkerSize',15);
%Funcion para calcular los indices MIA y CDI
[MIA_ind_sem,CDI_ind_sem]=MIAyCDI(semana_ind,index_ind_sem,N_ind_sem,C_ind
_sem,grafica_ind_sem);
title('Curva Representativa de Carga (Lunes a Viernes) - Industrial.', ...
    ['MIA= ', num2str(MIA_ind_sem), ', CDI= ', num2str(CDI_ind_sem)]);
xlabel('tiempo [horas]');
ylabel('Potencia P [pu]');
grid on
ylim([0 1]);

```

Datos Sabado Industrial

```

aux=1;
for i=1:filas%segmento para el dia sabado, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j), '6') &&
strcmp(datos_val(i,j+1), 'Industrial')
            hora_S_ind(aux,1)=datos_val(i,1);
            pt_S_ind(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end

```

```

end
A=str2double(hora_S_ind);
B=str2double(pt_S_ind);
sabado_ind=[A B];%forma la matriz que corresponde al dia sabado para
%el grupo industrial
%Algoritmo fuzzy k means
N_ind_S=15;%# de centros
options=[NaN NaN NaN 0];
[C_ind_S,U_ind_S]=fcm(sabado_ind,N_ind_S,options);
maxU_ind_S = max(U_ind_S);
index_ind_S = cell(N_ind_S,1);
for k=1:N_ind_S
    index_ind_S{k} = find(U_ind_S(k,:) == maxU_ind_S);
end
figure
hold on
for k=1:N_ind_S
    plot(sabado_ind(index_ind_S{k},1),sabado_ind(index_ind_S{k},2), 'o');
    plot(C_ind_S(k,1),C_ind_S(k,2), 'xk', 'MarkerSize',15, 'LineWidth',3);
end
hold off
aux2=1;
M_ind_P=zeros(N_ind_S,1);%inicializo el vector para que se pueda
actualizar el tamaño
matriz_C=C_ind_S;%tomo todos los centros del sabado de los industriales.
M_ind_t=sort(matriz_C(:,1));%ordeno de menor a mayor el tiempo
for i=1:N_ind_S
    for j=1:N_ind_S
        if M_ind_t(i,1)==matriz_C(j,1)
            M_ind_P(aux2,1)=matriz_C(j,2);%emparejo cada tiempo con su
respectiva potencia
            aux2=aux2+1;
        end
    end
end
grafica_ind_S=[M_ind_t M_ind_P];%formo la matriz para graficar la curva de
carga
figure
plot(grafica_ind_S(:,1),grafica_ind_S(:,2),'--. k', 'MarkerSize',15);
%Funcion para calcular los indices MIA y CDI
[MIA_ind_S,CDI_ind_S]=MIAyCDI(sabado_ind,index_ind_S,N_ind_S,C_ind_S,grafi
ca_ind_S);
title('Curva Representativa de Carga (Sábado) - Industrial.', ...
    ['MIA= ',num2str(MIA_ind_S),', CDI= ',num2str(CDI_ind_S)]);
xlabel('tiempo [horas]');
ylabel('Potencia P [pu]');
grid on;
ylim([0 1]);

```


Datos Domingo Industrial

```
aux=1;
for i=1:filas%segmento para el dia domingo, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'7') &&
strcmp(datos_val(i,j+1),'Industrial')
            hora_D_ind(aux,1)=datos_val(i,1);
            pt_D_ind(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_D_ind);
B=str2double(pt_D_ind);
domingo_ind=[A B];%forma la matriz que corresponde al dia domingo para
%el grupo industrial
%Algoritmo fuzzy k means
N_ind_D=13;%# de centros
options=[NaN NaN NaN 0];
[C_ind_D,U_ind_D]=fcm(domingo_ind,N_ind_D,options);
maxU_ind_D = max(U_ind_D);
index_ind_D = cell(N_ind_D,1);
for k=1:N_ind_D
    index_ind_D{k} = find(U_ind_D(k,:) == maxU_ind_D);
end
figure
hold on
for k=1:N_ind_D
    plot(domingo_ind(index_ind_D{k},1),domingo_ind(index_ind_D{k},2),'o');
    plot(C_ind_D(k,1),C_ind_D(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
hold off
aux2=1;
M_ind_P=zeros(N_ind_D,1);%inicializo el vector para que se pueda
actualizar el tamaño
matriz_C=C_ind_D;%tomo todos los centros del domingo de los industriales.
M_ind_t=sort(matriz_C(:,1));%ordeno de menor a mayor el tiempo
for i=1:N_ind_D
    for j=1:N_ind_D
        if M_ind_t(i,1)==matriz_C(j,1)
            M_ind_P(aux2,1)=matriz_C(j,2);%emparejo cada tiempo con su
respectiva potencia
            aux2=aux2+1;
        end
    end
end
end
```

```

grafica_ind_D=[M_ind_t M_ind_P];%formo la matriz para graficar la curva de
carga
figure
plot(grafica_ind_D(:,1),grafica_ind_D(:,2),'--. k','MarkerSize',15);
%Funcion para calcular los indices MIA y CDI
[MIA_ind_D,CDI_ind_D]=MIAyCDI(domingo_ind,index_ind_D,N_ind_D,C_ind_D,graf
ica_ind_D);
title('Curva Representativa de Carga (Domingo) - Industrial.', ...
      ['MIA= ',num2str(MIA_ind_D),', CDI= ',num2str(CDI_ind_D)]);
xlabel('tiempo [horas]');
ylabel('Potencia P [pu]');
grid on;
ylim([0 1]);

```

Datos Lunes Comercial

```

aux=1;
for i=1:filas%segmento para el dia lunes, la hora con su valor de potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'1') &&
strcmp(datos_val(i,j+1),'Comercial')
            hora_L_com(aux,1)=datos_val(i,1);
            pt_L_com(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_L_com);
B=str2double(pt_L_com);
lunes_com=[A B];%forma la matriz que corresponde al dia lunes para
%el grupo comercial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_com_L,U_com_L]=fcm(lunes_com,N,options);%(fin)
%Para clasificar cada punto en el cluster para el que tiene
%el mayor grado de pertenencia.
maxU_com_L = max(U_com_L);
index_com_L = cell(N,1);
for k=1:N
    index_com_L{k} = find(U_com_L(k,:) == maxU_com_L);
end
figure
hold on
for k=1:N
    plot(lunes_com(index_com_L{k},1),lunes_com(index_com_L{k},2),'o');
    plot(C_com_L(k,1),C_com_L(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
end

```

```
hold off
```

Datos martes Comercial

```
aux=1;
for i=1:filas%segmento para el dia martes, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'2') &&
strcmp(datos_val(i,j+1),'Comercial')
            hora_M_com(aux,1)=datos_val(i,1);
            pt_M_com(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_M_com);
B=str2double(pt_M_com);
martes_com=[A B];%forma la matriz que corresponde al dia martes para
%el grupo comercial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_com_M,U_com_M]=fcm(martes_com,N,options);
maxU_com_M = max(U_com_M);
index_com_M = cell(N,1);
for k=1:N
    index_com_M{k} = find(U_com_M(k,:) == maxU_com_M);
end
figure
hold on
for k=1:N
    plot(martes_com(index_com_M{k},1),martes_com(index_com_M{k},2), 'o');
    plot(C_com_M(k,1),C_com_M(k,2), 'xk', 'MarkerSize',15, 'LineWidth',3);
end
hold off
```

Datos miercoles Comercial

```
aux=1;
for i=1:filas%segmento para el dia miercoles, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'3') &&
strcmp(datos_val(i,j+1),'Comercial')
            hora_W_com(aux,1)=datos_val(i,1);
            pt_W_com(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
```

```

        end
    end
end
A=str2double(hora_W_com);
B=str2double(pt_W_com);
miercoles_com=[A B];%forma la matriz que corresponde al dia miercoles para
%el grupo comercial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_com_W,U_com_W]=fcm(miercoles_com,N,options);
maxU_com_W = max(U_com_W);
index_com_W = cell(N,1);
for k=1:N
    index_com_W{k} = find(U_com_W(k,:) == maxU_com_W);
end
figure
hold on
for k=1:N
plot(miercoles_com(index_com_W{k},1),miercoles_com(index_com_W{k},2),'o');
    plot(C_com_W(k,1),C_com_W(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
hold off

```

Datos jueves Comercial

```

aux=1;
for i=1:filas%segmento para el dia jueves, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'4') &&
strcmp(datos_val(i,j+1),'Comercial')
            hora_J_com(aux,1)=datos_val(i,1);
            pt_J_com(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_J_com);
B=str2double(pt_J_com);
jueves_com=[A B];%forma la matriz que corresponde al dia jueves para
%el grupo comercial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_com_J,U_com_J]=fcm(jueves_com,N,options);
maxU_com_J = max(U_com_J);

```

```

index_com_J = cell(N,1);
for k=1:N
    index_com_J{k} = find(U_com_J(k,:) == maxU_com_J);
end
figure
hold on
for k=1:N
    plot(jueves_com(index_com_J{k},1),jueves_com(index_com_J{k},2), 'o');
    plot(C_com_J(k,1),C_com_J(k,2), 'xk', 'MarkerSize',15, 'LineWidth',3);
end
hold off

```

Datos viernes Comercial

```

aux=1;
for i=1:filas%segmento para el dia viernes, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j), '5') &&
strcmp(datos_val(i,j+1), 'Comercial')
            hora_V_com(aux,1)=datos_val(i,1);
            pt_V_com(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_V_com);
B=str2double(pt_V_com);
viernes_com=[A B];%forma la matriz que corresponde al dia viernes para
%el grupo comercial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_com_V,U_com_V]=fcm(viernes_com,N,options);
maxU_com_V = max(U_com_V);
index_com_V = cell(N,1);
for k=1:N
    index_com_V{k} = find(U_com_V(k,:) == maxU_com_V);
end
figure
hold on
for k=1:N
    plot(viernes_com(index_com_V{k},1),viernes_com(index_com_V{k},2), 'o');
    plot(C_com_V(k,1),C_com_V(k,2), 'xk', 'MarkerSize',15, 'LineWidth',3);
end
hold off

```

Datos lunes a viernes Comercial

```

semana_com=[lunes_com;martes_com;
            miercoles_com;jueves_com;viernes_com];
%Algoritmo fuzzy k means
N_com_sem=15;%# de centros
options=[NaN NaN NaN 0];
[C_com_sem,U_com_sem]=fcm(semana_com,N_com_sem,options);
maxU_com_sem = max(U_com_sem);
index_com_sem = cell(N_com_sem,1);
for k=1:N_com_sem
    index_com_sem{k} = find(U_com_sem(k,:) == maxU_com_sem);
end
figure
hold on
for k=1:N_com_sem

plot(semana_com(index_com_sem{k},1),semana_com(index_com_sem{k},2), 'o');

plot(C_com_sem(k,1),C_com_sem(k,2), 'xk', 'MarkerSize',15, 'LineWidth',3);
end
hold off
aux2=1;
M_com_P=zeros(N_com_sem,1);%inicializo el vector para que se pueda
actualizar el tamaño
matriz_C=C_com_sem;%tomo todos los centros de L a V de los comerciales.
M_com_t=sort(matriz_C(:,1));%ordeno de menor a mayor el tiempo
for i=1:N_com_sem
    for j=1:N_com_sem
        if M_com_t(i,1)==matriz_C(j,1)
            M_com_P(aux2,1)=matriz_C(j,2);%emparejo cada tiempo con su
respectiva potencia
            aux2=aux2+1;
        end
    end
end
grafica_com_sem=[M_com_t M_com_P];%formo la matriz para graficar la curva
de carga
figure
plot(grafica_com_sem(:,1),grafica_com_sem(:,2), '--. k', 'MarkerSize',15);
%Funcion para calcular los indices MIA y CDI
[MIA_com_sem,CDI_com_sem]=MIAyCDI(semana_com,index_com_sem,N_com_sem,C_com
_sem,grafica_com_sem);
title('Curva Representativa de Carga (Lunes a Viernes) - Comercial.', ...
      ['MIA= ',num2str(MIA_com_sem), ', CDI= ',num2str(CDI_com_sem)]);
xlabel('tiempo [horas]');
ylabel('Potencia P [pu]');
grid on
ylim([0 1]);

```

Datos sabado Comercial

```
aux=1;
for i=1:filas%segmento para el dia sabado, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'6') &&
strcmp(datos_val(i,j+1),'Comercial')
            hora_S_com(aux,1)=datos_val(i,1);
            pt_S_com(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_S_com);
B=str2double(pt_S_com);
sabado_com=[A B];%forma la matriz que corresponde al dia sabado para
%el grupo comercial
%Algoritmo fuzzy k means
N_com_S=12;%# de centros
options=[NaN NaN NaN 0];
[C_com_S,U_com_S]=fcm(sabado_com,N_com_S,options);
maxU_com_S = max(U_com_S);
index_com_S = cell(N_com_S,1);
for k=1:N_com_S
    index_com_S{k} = find(U_com_S(k,:) == maxU_com_S);
end
figure
hold on
for k=1:N_com_S
    plot(sabado_com(index_com_S{k},1),sabado_com(index_com_S{k},2),'o');
    plot(C_com_S(k,1),C_com_S(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
hold off
aux2=1;
M_com_P=zeros(N_com_S,1);%inicializo el vector para que se pueda
actualizar el tamaño
matriz_C=C_com_S;%tomo todos los centros del sabado de los comerciales.
M_com_t=sort(matriz_C(:,1));%ordeno de menor a mayor el tiempo
for i=1:N_com_S
    for j=1:N_com_S
        if M_com_t(i,1)==matriz_C(j,1)
            M_com_P(aux2,1)=matriz_C(j,2);%emparejo cada tiempo con su
respectiva potencia
            aux2=aux2+1;
        end
    end
end
end
```

```

grafica_com_S=[M_com_t M_com_P];%formo la matriz para graficar la curva de
carga
figure
plot(grafica_com_S(:,1),grafica_com_S(:,2),'--. k','MarkerSize',15);
%Funcion para calcular los indices MIA y CDI
[MIA_com_S,CDI_com_S]=MIAyCDI(sabado_com,index_com_S,N_com_S,C_com_S,grafi
ca_com_S);
title('Curva Representativa de Carga (Sábado) - Comercial.', ...
      ['MIA= ',num2str(MIA_com_S),', CDI= ',num2str(CDI_com_S)]);
xlabel('tiempo [horas]');
ylabel('Potencia P [pu]');
grid on;
ylim([0 1]);

```

Datos domingo Comercial

```

aux=1;
for i=1:filas%segmento para el dia domingo, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'7') &&
strcmp(datos_val(i,j+1),'Comercial')
            hora_D_com(aux,1)=datos_val(i,1);
            pt_D_com(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_D_com);
B=str2double(pt_D_com);
domingo_com=[A B];%forma la matriz que corresponde al dia domingo para
%el grupo comercial
%Algoritmo fuzzy k means
N_com_D=15;%# de centros
options=[NaN NaN NaN 0];
[C_com_D,U_com_D]=fcm(domingo_com,N_com_D,options);
maxU_com_D = max(U_com_D);
index_com_D = cell(N_com_D,1);
for k=1:N_com_D
    index_com_D{k} = find(U_com_D(k,:) == maxU_com_D);
end
figure
hold on
for k=1:N_com_D
    plot(domingo_com(index_com_D{k},1),domingo_com(index_com_D{k},2),'o');
    plot(C_com_D(k,1),C_com_D(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
hold off

```



```

aux2=1;
M_com_P=zeros(N_com_D,1);%inicializo el vector para que se pueda
actualizar el tamaño
matriz_C=C_com_D;%tomo todos los centros del domingo de los comerciales.
M_com_t=sort(matriz_C(:,1));%orden de menor a mayor el tiempo
for i=1:N_com_D
    for j=1:N_com_D
        if M_com_t(i,1)==matriz_C(j,1)
            M_com_P(aux2,1)=matriz_C(j,2);%emparejo cada tiempo con su
respectiva potencia
            aux2=aux2+1;
        end
    end
end
grafica_com_D=[M_com_t M_com_P];%formo la matriz para graficar la curva de
carga
figure
plot(grafica_com_D(:,1),grafica_com_D(:,2),'--. k','MarkerSize',15);
%Funcion para calcular los indices MIA y CDI
[MIA_com_D,CDI_com_D]=MIAyCDI(domingo_com,index_com_D,N_com_D,C_com_D,graf
ica_com_D);
title('Curva Representativa de Carga (Domingo) - Comercial.', ...
[MIA= ',num2str(MIA_com_D),', CDI= ',num2str(CDI_com_D)]);
xlabel('tiempo [horas]');
ylabel('Potencia P [pu]');
grid on;
ylim([0 1]);

```

Datos lunes residencial

```

aux=1;
for i=1:filas%segmento para el dia lunes, la hora con su valor de potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'1') &&
strcmp(datos_val(i,j+1),'Residencial')
            hora_L_res(aux,1)=datos_val(i,1);
            pt_L_res(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_L_res);
B=str2double(pt_L_res);
lunes_res=[A B];%forma la matriz que corresponde al dia lunes para
%el grupo residencial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];

```

```

[C_res_L,U_res_L]=fcm(lunes_res,N,options);
maxU_res_L = max(U_res_L);
index_res_L = cell(N,1);
for k=1:N
    index_res_L{k} = find(U_res_L(k,:) == maxU_res_L);
end
figure
hold on
for k=1:N
    plot(lunes_res(index_res_L{k},1),lunes_res(index_res_L{k},2), 'o');
    plot(C_res_L(k,1),C_res_L(k,2), 'xk', 'MarkerSize',15, 'LineWidth',3);
end
hold off

```

Datos martes Residencial

```

aux=1;
for i=1:filas%segmento para el dia martes, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j), '2') &&
strcmp(datos_val(i,j+1), 'Residencial')
            hora_M_res(aux,1)=datos_val(i,1);
            pt_M_res(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_M_res);
B=str2double(pt_M_res);
martes_res=[A B];%forma la matriz que corresponde al dia martes para
%el grupo residencial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_res_M,U_res_M]=fcm(martes_res,N,options);
maxU_res_M = max(U_res_M);
index_res_M = cell(N,1);
for k=1:N
    index_res_M{k} = find(U_res_M(k,:) == maxU_res_M);
end
figure
hold on
for k=1:N
    plot(martes_res(index_res_M{k},1),martes_res(index_res_M{k},2), 'o');
    plot(C_res_M(k,1),C_res_M(k,2), 'xk', 'MarkerSize',15, 'LineWidth',3);
end
hold off

```

Datos miercoles Residencial

```
aux=1;
for i=1:filas%segmento para el dia miercoles, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'3') &&
strcmp(datos_val(i,j+1),'Residencial')
            hora_W_res(aux,1)=datos_val(i,1);
            pt_W_res(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_W_res);
B=str2double(pt_W_res);
miercoles_res=[A B];%forma la matriz que corresponde al dia miercoles para
%el grupo residencial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_res_W,U_res_W]=fcm(miercoles_res,N,options);
maxU_res_W = max(U_res_W);
index_res_W = cell(N,1);
for k=1:N
    index_res_W{k} = find(U_res_W(k,:) == maxU_res_W);
end
figure
hold on
for k=1:N
    plot(miercoles_res(index_res_W{k},1),miercoles_res(index_res_W{k},2),'o');
        plot(C_res_W(k,1),C_res_W(k,2),'xk','MarkerSize',15,'LineWidth',3);
    end
hold off
```

Datos jueves Residencial

```
aux=1;
for i=1:filas%segmento para el dia jueves, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'4') &&
strcmp(datos_val(i,j+1),'Residencial')
            hora_J_res(aux,1)=datos_val(i,1);
            pt_J_res(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
```

```

    end
end
A=str2double(hora_J_res);
B=str2double(pt_J_res);
jueves_res=[A B];%forma la matriz que corresponde al dia jueves para
%el grupo residencial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_res_J,U_res_J]=fcm(jueves_res,N,options);
maxU_res_J = max(U_res_J);
index_res_J = cell(N,1);
for k=1:N
    index_res_J{k} = find(U_res_J(k,:) == maxU_res_J);
end
figure
hold on
for k=1:N
    plot(jueves_res(index_res_J{k},1),jueves_res(index_res_J{k},2),'o');
    plot(C_res_J(k,1),C_res_J(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
hold off

```

Datos viernes Residencial

```

aux=1;
for i=1:filas%segmento para el dia viernes, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'5') &&
strcmp(datos_val(i,j+1),'Residencial')
            hora_V_res(aux,1)=datos_val(i,1);
            pt_V_res(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_V_res);
B=str2double(pt_V_res);
viernes_res=[A B];%forma la matriz que corresponde al dia viernes para
%el grupo residencial
%Algoritmo fuzzy k means
N=12;%# de centros
options=[NaN NaN NaN 0];
[C_res_V,U_res_V]=fcm(viernes_res,N,options);
maxU_res_V = max(U_res_V);
index_res_V = cell(N,1);
for k=1:N

```

```

        index_res_V{k} = find(U_res_V(k,:) == maxU_res_V);
    end
    figure
    hold on
    for k=1:N
        plot(viernes_res(index_res_V{k},1),viernes_res(index_res_V{k},2), 'o');
        plot(C_res_V(k,1),C_res_V(k,2), 'xk', 'MarkerSize',15, 'LineWidth',3);
    end
    hold off

```

Datos lunes a viernes Residencial

```

semana_res=[lunes_res;martes_res;
            miercoles_res;jueves_res;viernes_res];
%Algoritmo fuzzy k means
N_res_sem=16;%# de centros
options=[NaN NaN NaN 0];
[C_res_sem,U_res_sem]=fcm(semana_res,N_res_sem,options);
maxU_res_sem = max(U_res_sem);
index_res_sem = cell(N_res_sem,1);
for k=1:N_res_sem
    index_res_sem{k} = find(U_res_sem(k,:) == maxU_res_sem);
end
figure
hold on
for k=1:N_res_sem

plot(semana_res(index_res_sem{k},1),semana_res(index_res_sem{k},2), 'o');

plot(C_res_sem(k,1),C_res_sem(k,2), 'xk', 'MarkerSize',15, 'LineWidth',3);
end
hold off
aux2=1;
M_res_P=zeros(N_res_sem,1);%inicializo el vector para que se pueda
actualizar el tamaño
matriz_C=C_res_sem;%tomo todos los centros de L a V de los residenciales.
M_res_t=sort(matriz_C(:,1));%ordeno de menor a mayor el tiempo
for i=1:N_res_sem
    for j=1:N_res_sem
        if M_res_t(i,1)==matriz_C(j,1)
            M_res_P(aux2,1)=matriz_C(j,2);%emparejo cada tiempo con su
respectiva potencia
            aux2=aux2+1;
        end
    end
end
grafica_res_sem=[M_res_t M_res_P];%formo la matriz para graficar la curva
de carga

```

```

figure
plot(grafica_res_sem(:,1),grafica_res_sem(:,2),'--. k','MarkerSize',15);
%Funcion para calcular los indices MIA y CDI
[MIA_res_sem,CDI_res_sem]=MIAyCDI(semana_res,index_res_sem,N_res_sem,C_res
_sem,grafica_res_sem);
title('Curva Representativa de Carga (Lunes a Viernes) - Residencial.',
...
['MIA= ',num2str(MIA_res_sem),', CDI= ',num2str(CDI_res_sem)]);
xlabel('tiempo [horas]');
ylabel('Potencia P [pu]');
grid on
ylim([0 1]);

```

Datos sabado Residencial

```

aux=1;
for i=1:filas%segmento para el dia sabado, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'6') &&
strcmp(datos_val(i,j+1),'Residencial')
            hora_S_res(aux,1)=datos_val(i,1);
            pt_S_res(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_S_res);
B=str2double(pt_S_res);
sabado_res=[A B];%forma la matriz que corresponde al dia sabado para
%el grupo residencial
%Algoritmo fuzzy k means
N_res_S=15;%# de centros
options=[NaN NaN NaN 0];
[C_res_S,U_res_S]=fcm(sabado_res,N_res_S,options);
maxU_res_S = max(U_res_S);
index_res_S = cell(N_res_S,1);
for k=1:N_res_S
    index_res_S{k} = find(U_res_S(k,:) == maxU_res_S);
end
figure
hold on
for k=1:N_res_S
    plot(sabado_res(index_res_S{k},1),sabado_res(index_res_S{k},2),'o');
    plot(C_res_S(k,1),C_res_S(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
hold off
aux2=1;

```

```

M_res_P=zeros(N_res_S,1);%inicializo el vector para que se pueda
actualizar el tamaño
matriz_C=C_res_S;%tomo todos los centros del sabado de los residenciales.
M_res_t=sort(matriz_C(:,1));%ordeno de menor a mayor el tiempo
for i=1:N_res_S
    for j=1:N_res_S
        if M_res_t(i,1)==matriz_C(j,1)
            M_res_P(aux2,1)=matriz_C(j,2);%emparejo cada tiempo con su
respectiva potencia
            aux2=aux2+1;
        end
    end
end
grafica_res_S=[M_res_t M_res_P];%formo la matriz para graficar la curva de
carga
figure
plot(grafica_res_S(:,1),grafica_res_S(:,2),'--. k','MarkerSize',15);
%Funcion para calcular los indices MIA y CDI
[MIA_res_S,CDI_res_S]=MIAyCDI(sabado_res,index_res_S,N_res_S,C_res_S,grafi
ca_res_S);
title('Curva Representativa de Carga (Sábado) - Residencial.', ...
    ['MIA= ',num2str(MIA_res_S),', CDI= ',num2str(CDI_res_S)]);
xlabel('tiempo [horas]');
ylabel('Potencia P [pu]');
grid on;
ylim([0 1]);

```

Datos domingo Residencial

```

aux=1;
for i=1:filas%segmento para el dia domingo, la hora con su valor de
potencia
    for j=1:columnas
        if strcmp(datos_val(i,j),'7') &&
strcmp(datos_val(i,j+1),'Residencial')
            hora_D_res(aux,1)=datos_val(i,1);
            pt_D_res(aux,1)=datos_val(i,2);
            aux=aux+1;
        end
    end
end
A=str2double(hora_D_res);
B=str2double(pt_D_res);
domingo_res=[A B];%forma la matriz que corresponde al dia domingo para
%el grupo residencial
%Algoritmo fuzzy k means
N_res_D=15;%# de centros
options=[NaN NaN NaN 0];

```

```

[C_res_D,U_res_D]=fcm(domingo_res,N_res_D,options);
maxU_res_D = max(U_res_D);
index_res_D = cell(N_res_D,1);
for k=1:N_res_D
    index_res_D{k} = find(U_res_D(k,:) == maxU_res_D);
end
figure
hold on
for k=1:N_res_D
    plot(domingo_res(index_res_D{k},1),domingo_res(index_res_D{k},2),'o');
    plot(C_res_D(k,1),C_res_D(k,2),'xk','MarkerSize',15,'LineWidth',3);
end
hold off
aux2=1;
M_res_P=zeros(N_res_D,1);%inicializo el vector para que se pueda
actualizar el tamaño
matriz_C=C_res_D;%tomo todos los centros del domingo de los residenciales.
M_res_t=sort(matriz_C(:,1));%ordeno de menor a mayor el tiempo
for i=1:N_res_D
    for j=1:N_res_D
        if M_res_t(i,1)==matriz_C(j,1)
            M_res_P(aux2,1)=matriz_C(j,2);%emparejo cada tiempo con su
respectiva potencia
            aux2=aux2+1;
        end
    end
end
grafica_res_D=[M_res_t M_res_P];%formo la matriz para graficar la curva de
carga
figure
plot(grafica_res_D(:,1),grafica_res_D(:,2),'--. k','MarkerSize',15);
%Funcion para calcular los indices MIA y CDI
[MIA_res_D,CDI_res_D]=MIAyCDI(domingo_res,index_res_D,N_res_D,C_res_D,graf
ica_res_D);
title('Curva Representativa de Carga (Domingo) - Residencial.', ...
    ['MIA= ',num2str(MIA_res_D),', CDI= ',num2str(CDI_res_D)]);
xlabel('tiempo [horas]');
ylabel('Potencia P [pu]');
grid on;
ylim([0 1]);

```

ANEXO III: FUNCIÓN MIAyCDI

```

function [MIA,CDI] = MIAyCDI(datos,grados,Nc,Matriz_C,C_ord)
total_mia=0;%para hacer las sumas
for i=1:Nc
matriz_MIA=[datos(grados{i},1),datos(grados{i},2)];%(x1,y1)
    for j=1:size(matriz_MIA)

```



```

        dist_1=(matriz_MIA(j,1)-Matriz_C(i,1))^2+(matriz_MIA(j,2)-
Matriz_C(i,2))^2;%distancia euclidiana
        suma_mia=dist_1;
        total_mia=total_mia+suma_mia;%sumatorio
    end
end
MIA=sqrt(sqrt(total_mia)/Nc);%indicador mia
%la distancia sera siempre entre los dos primeros centroides
dist_cdi=((C_ord(1,1)-C_ord(2,1))^2+(C_ord(1,2)-
C_ord(2,2))^2)^(1/2);%distancia euclidiana
CDI=MIA/dist_cdi;%indicador CDI
end

```