

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

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

**UTILIZACIÓN DE LA TÉCNICA DE AGRUPAMIENTO MODIFIED
FOLLOW THE LEADER PARA LA OBTENCIÓN DE CURVAS DE
CARGA REPRESENTATIVAS DE CONSUMIDORES TIPO:
RESIDENCIAL COMERCIAL E INDUSTRIAL**

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

JUAN LEONARDO MALDONADO MOYOLEMA

jleobro@hotmail.com

DIRECTOR: PATRICIA ELIZABETH OTERO VALLADARES

patricia.otero@epn.edu.ec

DMQ, febrero 2022

CERTIFICACIONES

Yo, JUAN LEONARDO MALDONADO MOYOLEMA 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.

JUAN MALDONADO

Certifico que el presente trabajo de integración curricular fue desarrollado por JUAN LEONARDO MALDONADO MOYOLEMA, bajo mi supervisión.

PATRICIA OTERO
DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el producto resultante 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.

JUAN MALDONADO

PATRICIA OTERO

DEDICATORIA

Dedico este trabajo a mi madre María Teresa, mi padre Simón Bolívar y mis hermanos: Alejandra, Richard y Cristian, quienes siempre me han apoyado a lo largo de mi vida, sin duda no hay palabras para expresar el amor que siempre me han hecho sentir. Si pudiera de alguna forma regresar un poco de todo lo que me han dado, lo haría aún a cuesta de mi vida. Los amo.

AGRADECIMIENTO

Agradezco a todos los mentores que he tenido a lo largo de mi vida, ya que cada uno de ellos ha aportado de alguna forma en mi desarrollo como profesional, sin ellos nunca habría descubierto el maravilloso mundo del conocimiento.

ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN	VII
ABSTRACT	VIII
1 INTRODUCCIÓN	1
1.1 OBJETIVO GENERAL	1
1.2 OBJETIVOS ESPECÍFICOS	2
1.3 ALCANCE	2
1.4 MARCO TEÓRICO.....	2
1.4.1 Demanda eléctrica	2
1.4.2 Tipos de consumidores.....	4
1.4.3 Normalizar datos	6
1.4.4 Modified follow the leader.....	6
1.4.5 Índices de adecuación	7
1.4.6 Software Matlab.....	7
2 METODOLOGÍA	8
2.1 Tratamiento de datos	8
2.2 Consideraciones iniciales.....	8
2.3 Desarrollo del algoritmo	9
2.4 Aplicación del algoritmo en matlab.....	13
2.4.1 Importar datos desde excel.....	13
2.4.2 Clasificar base de datos.....	13
2.4.3 Respaldo de datos	14
2.4.4 Corrección de datos	15
2.4.5 Normalizar datos	15
2.4.6 Calcular constante de ponderamiento.....	15
2.4.7 Técnica de agrupamiento MFL.....	15
2.4.8 Calcular índice de adecuación MIA	15
2.4.9 Calcular índice de adecuación CDI	16

2.4.10	Graficar centros.....	16
2.4.11	Graficar grupos.....	17
2.4.12	Detalle de miembros de cada grupo	17
2.4.13	Función MFL_Distancia	18
2.4.14	Función MFL_Asignar	18
2.4.15	Función MFL_Reasignar	18
2.4.16	Función MFL_Eliminar.....	19
2.4.17	Función MFL_DE.....	19
2.4.18	Función MFL_Condensar.....	20
3	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	21
3.1	Resultados	21
3.1.1	Consumidores residenciales de lunes a viernes.....	21
3.1.2	Consumidores residenciales para los días sábado	29
3.1.3	Consumidores residenciales para los días domingo	31
3.1.4	Consumidores comerciales de lunes a viernes	32
3.1.5	Consumidores comerciales para los días sábado.....	38
3.1.6	Consumidores comerciales para los días domingo	40
3.1.7	Consumidores industriales para los días de lunes a viernes.....	41
3.1.8	Consumidores industriales para los días sábado	43
3.1.9	Consumidores industriales para los días domingo.....	44
3.2	Conclusiones.....	46
3.3	Recomendaciones.....	47
4	REFERENCIAS BIBLIOGRÁFICAS	49
5	ANEXOS.....	50
	ANEXO I: BASE DE DATOS.....	50
	ANEXO II: SCRIPT PRINCIPAL DE MATLAB	50
	ANEXO III FUNCIONES CREADAS EN MATLAB E ÍNDICES DE ADECUACIÓN	51

RESUMEN

Este trabajo de integración curricular utilizó las mediciones de consumo de potencia activa de 236 clientes pertenecientes a la empresa eléctrica CNEL-Unidad de Negocios Azogues, para formar una base de datos que contiene las curvas de carga diaria de todos los clientes, cada curva de carga diaria se construyó con la unión de potencias pico promedio en intervalos de una hora, durante un día para un mismo cliente. Se clasificó la base de datos por tipo de consumidor y por días de la semana, este conjunto de curvas de carga diaria fue normalizado con respecto a la máxima potencia activa de cada curva. Las curvas de carga diaria normalizadas ingresaron al algoritmo de agrupamiento “Modified Follow The Leader” que encontró curvas de carga diaria representativas, las mismas que proporcionan diferentes patrones de consumo eléctrico para el subconjunto tratado. Finalmente se calculó para cada subconjunto de datos los índices de adecuación: “The Mean Index Adequacity” y “The Clustering Dispersion Indicator” con lo que se analiza si el algoritmo es óptimo para agrupar iguales formas de curva de carga y separar diferentes formas de curvas de carga representativa, respectivamente.

PALABRAS CLAVE: Modified Follow The Leader, curvas de carga diaria representativas, índices de adecuación.

ABSTRACT

This curricular integration work used the consumption measurements of active power from 236 customers belonging to electrical company CNEL-Azogues Business Unit, to form a database containing the daily load curves of all the clients, each daily load curve was constructed with the union of the average peak powers at intervals of one hour, during one day for the same client. The database was classified by type of consumer and by days of the week, this set of daily load curves was normalized with respect to the maximum active power of each curve. The normalized daily load curves entered the “Modified Follow The Leader” which found representative daily load curves, the provide different patterns of electrical consumption for the treated subset. Finally, the adequacy indices were calculated for each data subset: “The Mean Index Adequacy” and “The Clustering Dispersion Indicator” with which we analyze whether the algorithm is optimal for grouping the same load curve shapes and separating different representative load curve shapes, respectively.

KEYWORDS: Modified Follow the Leader, Load curves representative, Adequacy indices.

1 INTRODUCCIÓN

El presente trabajo de integración curricular abordará la obtención de curvas de carga representativa diaria, utilizando la técnica de agrupamiento no supervisada sigue al líder modifica o MFL (Modified Follow-The-Leader). Esta técnica se aplica al conjunto de curvas de carga diaria producto de las mediciones del consumo de potencia activa de 236 clientes, para obtener curvas de carga diaria representativas, además se calcula dos índices de adecuación: dispersión de agrupamiento o CDI (Clustering Dispersion Indicator) e índice de medios o MIA (Mean Index Adequacy).

El análisis de la curva de carga diaria, permite identificar patrones de consumo responsables de las variaciones en la demanda eléctrica, con la finalidad de realizar la planificación eléctrica a corto, mediano y largo plazo [1], [2].

La técnica de agrupamiento MFL consiste en obtener curvas de carga diaria representativas, que son el centro de cada agrupación realizada por el algoritmo, este algoritmo asigna las curvas de carga diaria a un grupo específico, basándose en el cálculo de la distancia euclidiana ponderada entre la curva de carga centro del grupo y la curva de carga diaria analizada, si la distancia es menor a un umbral previamente establecido, se asignará la curva de carga diaria a dicho grupo y se recalculará el centro del mismo, caso contrario se creará un nuevo grupo en el que la nueva curva de carga diaria también es el centro de dicho grupo [3].

Se parte de las potencias activas consumidas por 236 clientes pertenecientes a la empresa eléctrica CNEL-Unidad de Negocios Azogues para obtener curvas de carga diaria uniendo las potencias pico promedio registradas para cada hora, formando la base de datos a analizar. Antes de ingresar las curvas de carga diaria al algoritmo, se las separa de lunes a viernes, sábados, domingos y por tipo de consumidor. Cada curva de carga diaria se normaliza con respecto a su máxima potencia activa registrada. El algoritmo de la técnica de agrupamiento MFL se desarrolla en Matlab y se incluye el despliegue de las curvas de carga diaria representativas para tres tipos de consumidores: residencial, comercial e industrial. Para comprobar la bondad del algoritmo se analiza los valores de los índices MIA y CDI.

1.1 OBJETIVO GENERAL

Utilizar técnicas de agrupamiento para la obtención de curvas de carga diaria representativas de consumidores tipo: residencial, comercial e industrial.

1.2 OBJETIVOS ESPECÍFICOS

1. Realizar la investigación bibliográfica referente a la obtención de curvas de carga representativas mediante la aplicación de la técnica de agrupamiento MFL.
2. Estructurar las bases de datos con las mediciones de cada uno de los tipos de clientes a tratar.
3. Desarrollar el algoritmo de la técnica de agrupamiento MFL en MatLab, para obtener las curvas de carga representativas, junto con los indicadores de adecuación.
4. Realizar el análisis de las curvas de carga representativas y de los índices de adecuación obtenidos, para emitir conclusiones y recomendaciones.

1.3 ALCANCE

Se parte de la investigación bibliográfica del algoritmo MFL, y su aplicación a la obtención de curvas de carga representativas para consumidores del tipo residencial, comercial e industrial.

Con la base teórica obtenida se pasa al tratamiento de los datos de cada tipo de consumidor y se estructura el algoritmo para su programación en Matlab. El programa en Matlab despliega las curvas representativas del tipo de consumidor analizado y calcula los índices MIA y CDI.

Los resultados obtenidos permiten evaluar la técnica de agrupamiento utilizada y su adecuación para la aplicación en obtención de curvas de carga representativas, con lo que se emite conclusiones y recomendaciones.

1.4 MARCO TEÓRICO

1.4.1 Demanda eléctrica

El estudio de la demanda eléctrica es necesario para realizar la planificación eléctrica tanto a largo, mediano y corto plazo. Por tal motivo es necesario analizar las curvas de carga diaria de cada consumidor, para encontrar diferentes patrones de consumo según el tipo de cliente, ya sea por: la actividad económica que realizan, el estilo de vida, las festividades que celebran y los factores climáticos que influyen a lo largo del año [1].

Dentro del territorio ecuatoriano las tarifas promedio de consumo en 2021 fueron: 0,1031 USD c/kWh para el sector residencial; 0,1044 USD c/kWh para el sector comercial; 0.799 USD c/kWh para el sector industrial.

Como se puede apreciar el precio del kilovatio hora depende del tipo de consumidor, esto debido a que la demanda de energía para una residencia no es la misma que necesita un comercio y peor aún una industria para operar. El objetivo de la planificación eléctrica es proveer de energía a todos sus clientes, por tal motivo se tiene un parque de generación eléctrica para solventar la demanda, este parque contiene centrales: hidroeléctricas, de biogás, fotovoltaicas, termoeléctricas y solares. Cada una de las centrales presenta sus propios inconvenientes y diferentes costos de generación eléctrica. Con el fin de abastecer a toda la población de energía eléctrica se solicita la misma a las diferentes centrales, sin embargo, existen ciertas horas al día en las que todos los clientes necesitan de energía eléctrica, es aquí cuando se hace necesario y obligatorio recurrir a las centrales de energía que más rápido pueden solventar la demanda, generalmente este tipo de generación es la termoeléctrica que también es una de las más caras, valor que se compensa al establecer diferentes tipos de consumidores. Este es el motivo por el cual es necesario estudiar los patrones de consumo de los clientes, con el fin de que a futuro se pueda solventar la demanda utilizando energías limpias y económicas [4].

1.4.1.1 Curva de carga

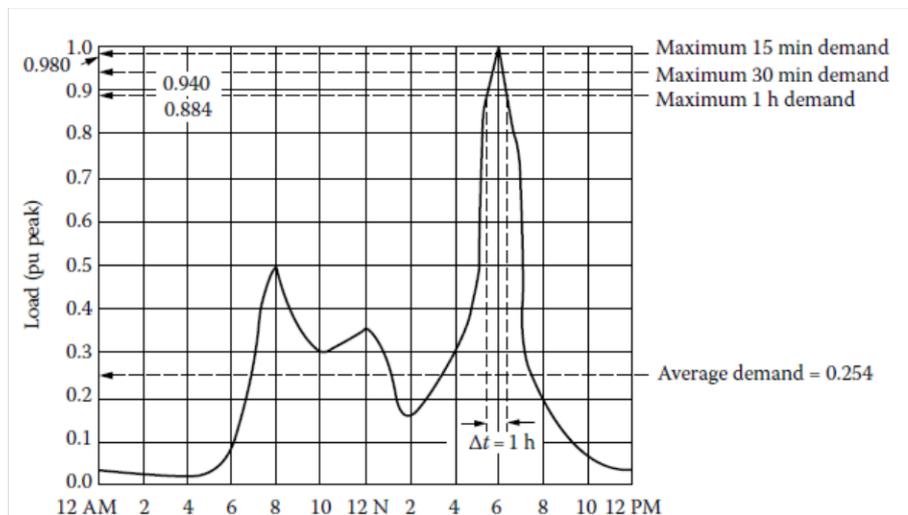


Figura 1.1. Curva de variación de la demanda diaria [2].

La curva característica de carga o curva de carga consiste en graficar el consumo eléctrico promedio que tiene un cliente durante un intervalo de tiempo, colocando en el eje horizontal los intervalos de tiempo de forma ascendente y en el eje vertical el consumo eléctrico para cada intervalo, como se observa en la Fig. 1.1. La curva de carga permite ver exactamente los intervalos en los que se produce mayor o menor demanda eléctrica. Analizando los historiales de las curvas de carga, se realiza la planificación diaria de energía [2].

1.4.1.2 Curva de carga representativa

La curva de carga representativa es una curva de carga que identifica un patrón de consumo de un grupo de clientes específico. En la Fig. 1.2. Las curvas de carga diaria de color: azul, tomate, morado, verde y rosa, se las puede reemplazar por la curva de color rojo, siendo esta la curva de carga representativa del grupo, que identifica un patrón de consumo.

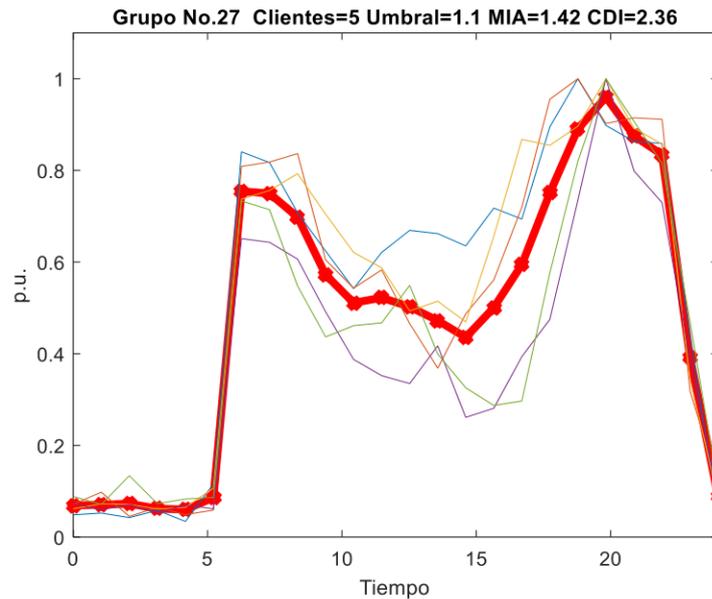


Figura 1.2. Curvas de carga diarias y curva de carga diaria representativa.

1.4.2 Tipos de consumidores

Los tipos de consumidores que se tratarán en el presente trabajo se detallan a continuación.

1.4.2.1 Consumidor residencial

La curva de carga representativa de un consumidor residencial se la puede ver en las Fig. 1.3., la primera pertenece a una carga residencial urbana y la segunda a una carga residencial rural, no existen diferencias significantes, de hecho, las dos curvas son parecidas describiendo que de 05:00 a 15:00 horas se tiene un consumo de energía casi plano, sin embargo, a partir de la 15:00 a 17:00 horas se produce un aumento de la demanda alcanzando un pico máximo a las 17:00 horas, después de esta hora se produce una reducción en la demanda, hasta las 05:00 del día siguiente. El comportamiento residencial se describe de la forma detallada debido a que la mayoría de personas se levanta temprano para alistarse al trabajo, después se ocupa la mínima energía eléctrica debido a que en el día no se necesita mayor número de luces prendidas, sin embargo, en

la noche es necesaria la luz, sumado a que todas las personas en su mayoría regresan a sus hogares.

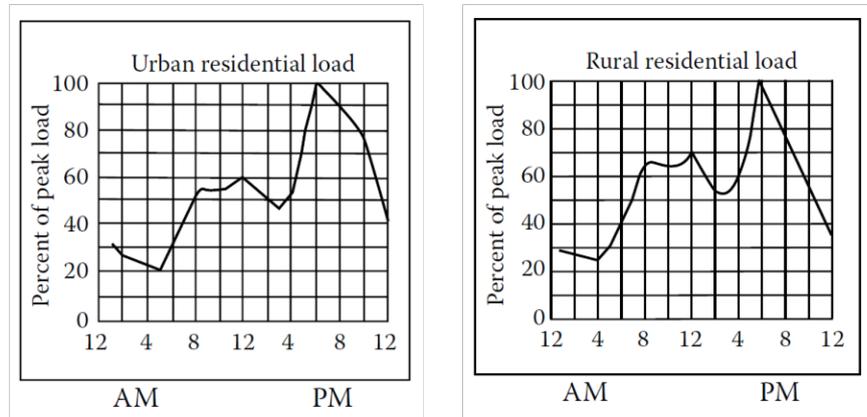


Figura 1.3. Curva de carga representativa residencial: urbana y rural respectivamente [2].

1.4.2.2 Consumidor comercial

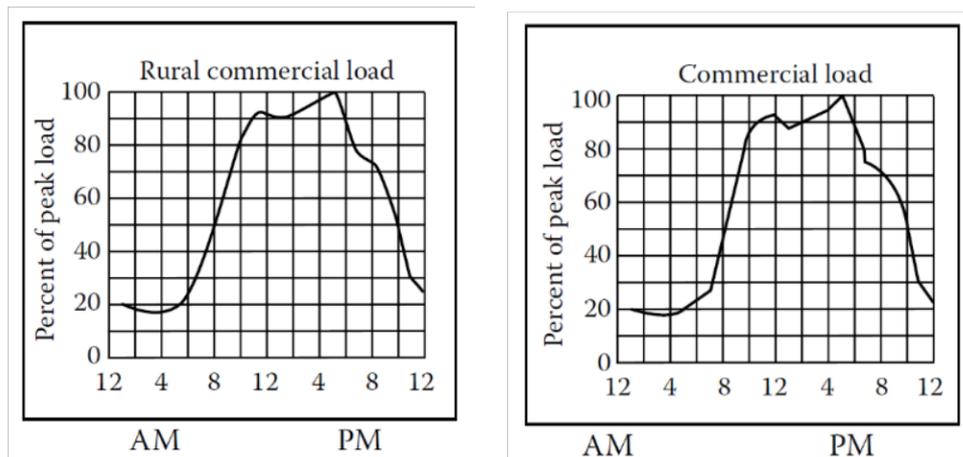


Figura 1.4. Curva de carga representativa comercial: rural y urbana respectivamente [2].

La curva de carga representativa para un cliente tipo comercial, se ve enormemente influenciada por el ritmo de vida de la población, como se puede observar en la Fig. 1.4. tiende a parecerse a la curva de carga de un consumidor residencial, en el sentido que a partir de las 05:00 horas hasta las 17:00 horas se produce una demanda casi continua, esto coincide con los horarios en los que las personas salen de sus hogares, hasta que regresan a los mismos.

1.4.2.3 Consumidor Industrial

La curva de carga representativa para un consumidor tipo industrial, depende mucho del país, ya que en países industrializados la forma de la curva de carga será prácticamente plana y con una demanda casi del cien por ciento con respecto a su demanda máxima. La curva de carga de la Fig. 1.5. representa a una industria de un país en desarrollo, ya que

nuevamente se presenta la mayor demanda de energía de 05:00 a 17:00 horas, coincidiendo con los horarios de trabajo de las personas, sin embargo se debe notar que la mayor parte de tiempo la curva de carga se mantiene en un consumo energético de más de la mitad de la demanda máxima.

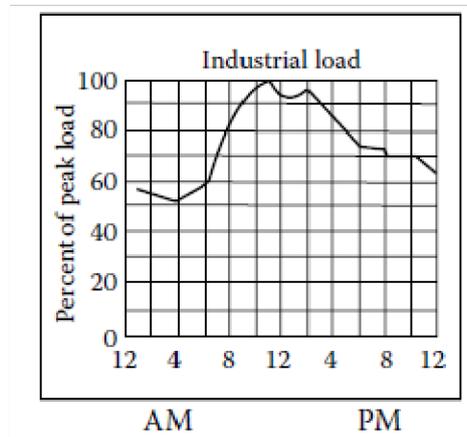


Figura 1.5. Curva de carga representativa industrial [2].

1.4.3 Normalizar datos

Normalizar la base de datos consiste en que cada curva de carga tenga un pico máximo de 1, existen varias formas de normalizar datos, se utiliza la división de la potencia activa de cada hora, para la potencia activa máxima de toda la curva de carga. No se normaliza cada curva para la potencia activa máxima de toda la base de datos, ya que esto ocasiona la pérdida de patrones de consumo de clientes especiales.

1.4.4 Modified follow the leader

Dentro de la teoría de minería de datos se tiene algunos métodos para la clasificación de datos que permiten el reconocimiento de patrones, se utilizará el método de clustering o agrupamiento modified follow-the-leader aplicado a curvas de carga diaria.

Este método de agrupamiento necesita: definir un umbral que sirve para agregar un dato a un grupo o crear uno nuevo, si el valor del umbral es muy alto se tendrá pocos grupos y viceversa; encontrar el factor de peso para cada hora de la curva de carga diaria que será aplicado en el cálculo de la distancia euclidiana ponderada.

El algoritmo consiste en calcular la distancia euclidiana ponderada entre un dato y el centro de todos los grupo que existan, después se selecciona la distancia menor calculada y se la compara con el umbral, resultando dos alternativas: la primera es si la distancia es menor o igual al umbral, el dato es agregado a dicho grupo y se procede a recalculer el centro del grupo; la segunda es que la distancia sea mayor al umbral, en este caso el dato es

agregado a un nuevo grupo y también es el centro de dicho grupo. Una vez que se termina de analizar todos los datos, se vuelve a repetir el algoritmo, pudiendo ocurrir que se creen o eliminen grupos al comparar la distancia de cada dato con el umbral. El algoritmo se debe repetir hasta que no exista ningún cambio de datos entre grupos, finalizando así el agrupamiento, donde cada uno de los centros es un patrón de consumo distinto [5].

La ventaja de este algoritmo, radica en que al necesitar solo del umbral para clasificar un dato en un grupo, se puede reducir el umbral hasta conseguir que todas las curvas de carga diaria que pertenecen a un mismo grupo sean muy parecidas, esto sumado al factor de peso con el que se calcula la distancia euclidiana ponderada, resulta en un método de agrupamiento que no necesita de demasiadas iteraciones hasta obtener grupos fijos, aparte el resultado final de los agrupamientos me arrojará grupos en los que solo existe un dato, lo que me dice que dicho dato es atípico y si se desea se podría eliminar de la base de datos y volver a correr el algoritmo. Es por esto que aparte de ser un buen algoritmo para el agrupamiento, también lo es para detectar datos atípicos que hayan pasado el filtro inicial antes de usar cualquier técnica de minería de datos.

1.4.5 Índices de adecuación

Los índices de adecuación: the mean index adequacy y the clustering dispersion indicator permiten calcular el rendimiento de la técnica de agrupamiento, estableciendo que mientras menor valor tengan los índices, proporcionan mayor calidad en su desempeño. El índice MIA proporciona información de que tan parecidas en forma son las curvas de carga diaria que pertenecen a un grupo. El índice CDI indica que tan bien el algoritmo crea grupos, con diferentes formas de curva de carga diaria representativa [5].

1.4.6 Software Matlab

MatLab es un lenguaje de cálculo técnico optimizado para trabajar con matrices, este programa consta de un gran número de funciones previamente definidas que serán utilizadas en el script que se desarrollará, además al estar optimizado para trabajar con matrices, permite operar de una forma directa entre filas y columnas lo que facilita el tratamiento de la base de datos y las operaciones entre datos.

2 METODOLOGÍA

2.1 Tratamiento de datos

Previo al ingreso de los datos en el algoritmo, se desarrolla un script en MatLab que permite obtener los datos desde Excel y clasificar los mismos por días de la semana y por tipo de consumidor, obteniendo un conjunto de curvas de carga diaria, además debido a que algunas curvas de carga tienen varios datos con potencias de cero y potencias negativas, se considera que estas potencias activas tuvieron un error al momento de ser tomadas, por lo tanto, este script inicial elimina curvas de carga con potencias de cero y corrige estas potencias negativas a positivas.

Se crea nueve conjuntos de curvas de carga diaria, tres conjuntos por cada tipo de consumidor (residencial, comercial e industrial), donde el primer conjunto contiene los datos de los días de la semana, el segundo contiene los datos de los días sábado y el tercero contiene los datos de los días domingo.

2.2 Consideraciones iniciales

Previo a iniciar el algoritmo se define que un dato x_i se compone por toda una curva de carga diaria, donde al decir que un dato tiene diferentes características se refiere a las 24 potencias que componen la curva de carga diaria, i representa el número de dato del conjunto a analizar.

$$x_i = \{x_{i1}, \dots, x_{i24}\} \quad (2.2.1)$$

Para todos los conjuntos primero se normaliza sus datos, en (2.2.2) se tiene que x_{ij} es la potencia activa de la curva de carga i , en la hora o característica j y $\text{máx}(x_i)$ representa la potencia activa máxima de la curva i .

$$\text{Potencia activa normalizada de } (x_{ij}) = \frac{x_{ij}}{\text{máx}(x_i)} \quad (2.2.2)$$

Se fija el valor del umbral en 1.5 para encontrar las primeras curvas de carga representativa y se irá modificando este valor según la necesidad de grupos que se requiera. También es necesario calcular el factor de peso que utiliza la distancia euclidiana ponderada, para esto se calcula la varianza (σ^2_j) de la misma característica usando todos los datos con (2.2.3) donde j es una de las 24 horas de la curva de carga diaria, \bar{x}_j es el promedio de todas las potencias activas normalizadas del conjunto a la hora j y n es el número total de datos del conjunto; después se calcula la varianza promedio ($\bar{\sigma}^2$) con (2.2.4) de las características

previamente encontradas, finalmente se encuentra el factor de peso k_j para cada característica dividiendo la varianza de dicha característica para la varianza promedio (2.2.5), obteniéndose 24 constantes de peso una para cada característica.

$$\sigma^2_j = \frac{\sum_{i=1}^n (x_{ij} - \bar{x}_j)^2}{n} \quad (2.2.3)$$

$$\overline{\sigma^2} = \frac{\sum_{j=1}^{24} (\sigma^2_j)}{24} \quad (2.2.4)$$

$$k_j = \frac{\sigma^2_j}{\overline{\sigma^2}} \quad (2.2.5)$$

2.3 Desarrollo del algoritmo

El algoritmo inicia tomando el primer dato (x_1) y debido a que no existen grupos al inicio, crea un grupo uno (G_1) y asigna el primer dato a este grupo, además establece como centro de G_1 a x_1 . Se toma el siguiente dato (x_2) y se calcula la distancia euclidiana ponderada con (2.3.1) entre x_2 y el centro de G_1 (c_1), si esta distancia es menor o igual al valor del umbral, x_2 es asignado a G_1 y se procederá a calcular el nuevo c_1 con (2.3.2), donde m_i es el número de miembros del grupo i ; si la distancia es mayor al umbral se crea un nuevo grupo G_2 donde el c_2 es x_2 .

$$dep_{ic} = \sqrt{\sum_{j=1}^{24} [k_j (x_{ij} - c_{ij})^2]} \quad (2.3.1)$$

$$\text{Nuevo } c_i = \frac{m_i}{m_i + 1} (c_i) + \frac{1}{m_i + 1} (x_i) \quad (2.3.2)$$

El procedimiento se repite con los siguientes datos creándose nuevos grupos, en este caso se calculará la distancia euclidiana ponderada entre todos los grupos existentes y el dato tratado, se escoge la distancia más pequeña para compararla con el umbral y decidir si se crea un nuevo grupo o el dato es asignado a uno ya existente, hasta finalizar con el miembro n de la base de datos, para nuevamente empezar el algoritmo desde el dato uno. En esta ocasión pueden ocurrir tres casos: el primero es que el dato x_i se mantenga en el mismo grupo; el segundo caso es que x_i se asigne a un nuevo grupo G_j , para esto es

necesario retirar el dato x_i del G_i al que pertenecía y recalculer el c_i ahora con (2.3.3) para proceder a añadir el dato x_i a un grupo G_j y recalculer el centro del grupo c_j al que se agregó con (2.3.2); el tercer caso es cuando x_i es el único miembro de un grupo, esto ocasiona que al calcular la distancia euclidiana ponderada entre el dato x_i y todos los centros existentes, se obtiene la mínima distancia igual a cero cuando se calcula con el centro c_i del grupo al que pertenece el dato el x_i , impidiendo que este dato se asigne a otro grupo, por tal motivo se vuelve a calcular la distancia euclidiana ponderada esta vez sin considerar el centro c_i en el que se encuentra el dato, si esta nueva distancia es menor o igual al umbral, el dato se asigna al grupo G_j que corresponda dicha distancia y se calcula el nuevo centro de c_j con (2.3.2), finalmente se elimina el grupo al que pertenecía el datos x_i , si la nueva distancia es mayor al umbral se dejará el grupo con un solo miembro y el centro será el mismo dato.

$$\text{Antiguo } C_i = \frac{m_i}{m_i - 1} (C_i) - \frac{1}{m_i - 1} (x_i) \quad (2.3.3)$$

Una vez que el algoritmo no crea grupos y tampoco existe intercambio de datos entre ellos, se da por finalizado el algoritmo, por tal motivo se debe fijar un número de iteraciones lo suficientemente alto para que el algoritmo logre alcanzar un agrupamiento fijo.

Para calcular el índice de adecuación MIA, primero se define la distancia euclidiana entre dos datos con (2.3.4) donde i y k es el número de dato, j es la hora en la curva de carga a la que se hace referencia, ahora se define la distancia euclidiana entre una curva de carga representativa c_k y todos los miembros x_m del grupo X_k al que pertenece con (2.3.5) donde m_k es el número de miembros del grupo k , finalmente se define el índice MIA con (2.3.6) donde C es el número total de grupos formador por el algoritmo.

$$d(x, y) = \sqrt{\frac{1}{24} \sum_{j=1}^{24} (x_{ij} - y_{kj})^2} \quad (2.3.4)$$

$$de(c_k, X_k) = \sqrt{\frac{1}{m_k} \sum_{m=1}^{m_k} d(c_k, x_m)^2} \quad (2.3.5)$$

$$MIA = \sqrt{\frac{1}{C} \sum_{k=1}^C de(c_k, X_k)^2} \quad (2.3.6)$$

El índice de adecuación CDI se calcula con (2.3.7) donde $de(x_m, X_k)^2$ representa el cuadrado de la distancia euclidiana (2.3.5) entre uno de los miembros x_m y cada uno de los miembros del conjunto X_k , para el grupo k ; de la misma forma $de(c_k, C)^2$ representa el cuadrado de la distancia euclidiana (2.3.5) entre el centro c_k del grupo k , y todo el conjunto de centros C .

$$CDI = \frac{\sqrt{\frac{1}{C} \sum_{k=1}^C \left[\frac{1}{2 m_k} \sum_{m=1}^{m_k} de(x_m, X_k)^2 \right]}}{\sqrt{\frac{1}{2 C} \sum_{k=1}^C de(c_k, C)^2}} \quad (2.3.7)$$

En la Fig. 2.1. se realiza un diagrama de flujo que explica cómo funciona el algoritmo desarrollado.

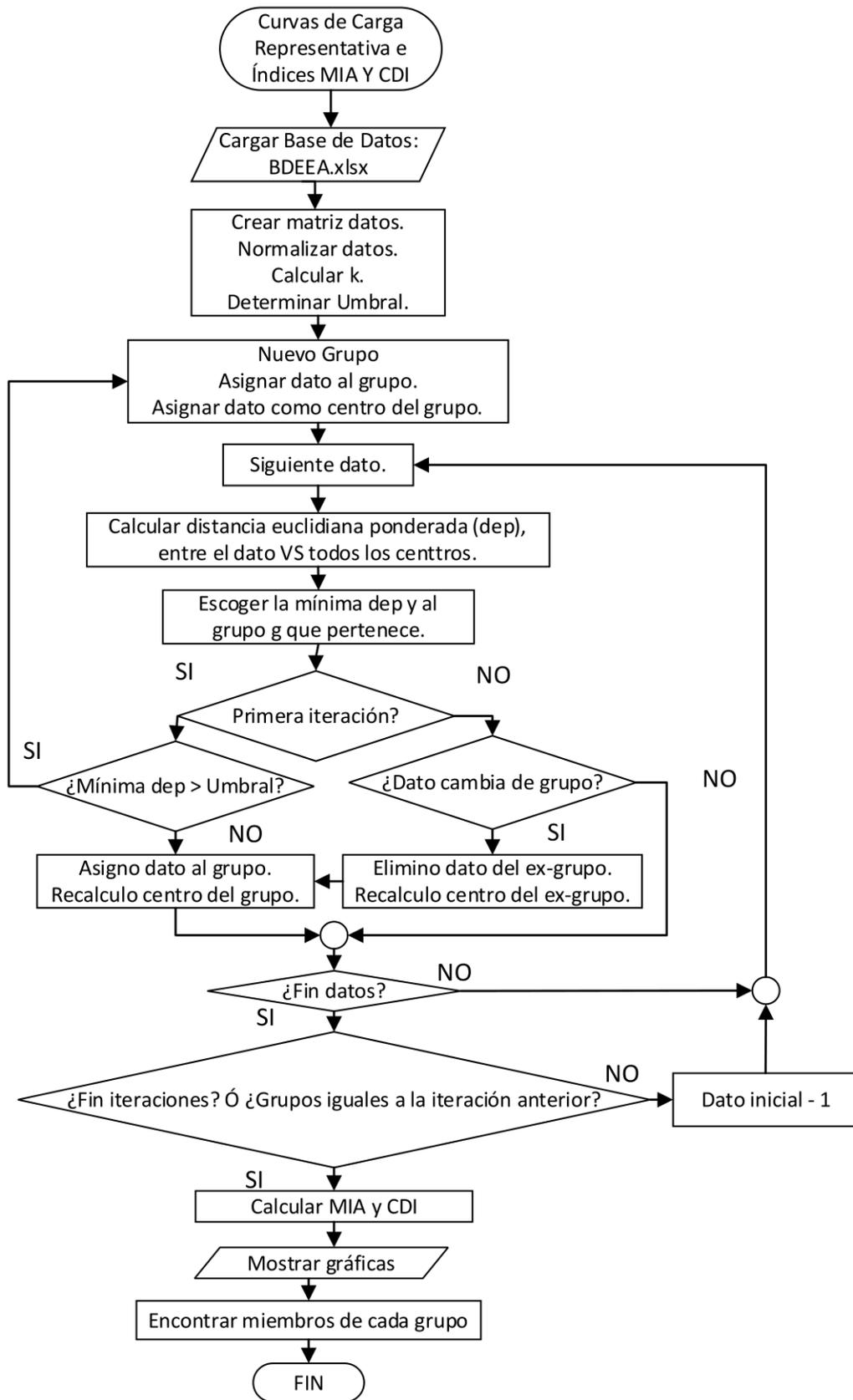


Figura 2.1. Diagrama de flujo del algoritmo desarrollado.

2.4 Aplicación del algoritmo en matlab

2.4.1 Importar datos desde excel

Se crea la función MFL_Importar, dentro de la función se utiliza inputdlg, función que despliega un cuadro de diálogo que permite interactuar con el usuario al pedir información por pantalla, también se puede escribir texto predefinido para que aparezca en el cuadro de diálogo, el texto predefinido que se elige es el nombre del archivo de la base de datos BDEEA.xlsx. Se utiliza otra función llamada readtable, y se la programa para que lea toda la tabla que se encuentre en la primera hoja de un archivo Excel, donde es necesario que tanto el archivo Excel como el script principal estén ubicados en la misma carpeta. La tabla importada desde Excel se guarda en la variable T. En la Fig. 2.2. se muestra el resultado de la función MFL_Importar.

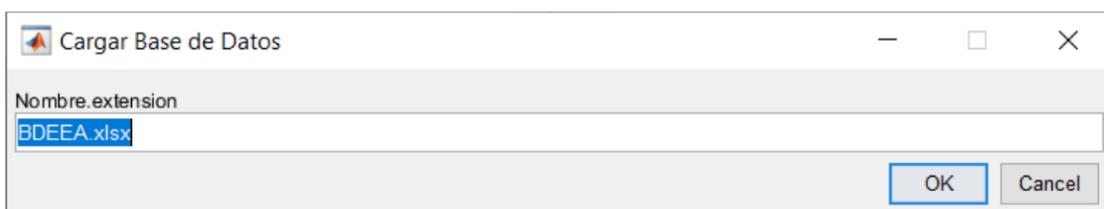


Figura 2.2. Función MFL_Importar.

2.4.2 Clasificar base de datos

La función MFL_Organizar, tiene como parámetro de entrada la tabla T que se cargó anteriormente, aquí se especificó por código que la curva de carga diaria constaría de un dato de potencia por hora, es decir de 24 datos para cada curva de carga diaria, en el caso que se tenga más valores de potencia por día, necesariamente debe cambiarse por programación en esta función, cambiando la asignación directa " $vt = 24;$ " por " $vt = \text{Número de potencias por día};$ ". Se utiliza la función listdlg la misma que permite desplegar una lista de nombres seleccionable, por tal motivo se pide con esta función seleccionar la columna en donde se encuentren: las fechas (FechaSola), los clientes (Medidor), las potencias de cada hora (Promedio P), las horas (HoraSola), estos datos son necesarios para realizar la clasificación. Después se despliega cómo se desea hacer la clasificación, aquí se puede seleccionar cualquier columna y la función desplegará todos los valores diferentes que se encuentren en dicha columna, después la función pregunta si se desea hacer otra clasificación, pudiendo elegir entre si o no, esta parte se programa con un bucle que solo saldrá al seleccionar que no se desea más clasificaciones. La función entregará

dos matrices, una con las curvas de carga diaria organizadas en columnas “C_1;” y otra con los números de las filas de donde se tomaron los datos de la tabla T “c_1;”, esta última matriz sirve para identificar en qué grupo se encuentra cada curva de carga y a quién pertenece dicha curva de carga. Si se elige alguna columna que no corresponda a los datos esenciales para la clasificación, se debe volver a ejecutar esta función ya que no se programa una comprobación de datos. En la Fig. 2.3. se visualiza la función MFL_Organizar.

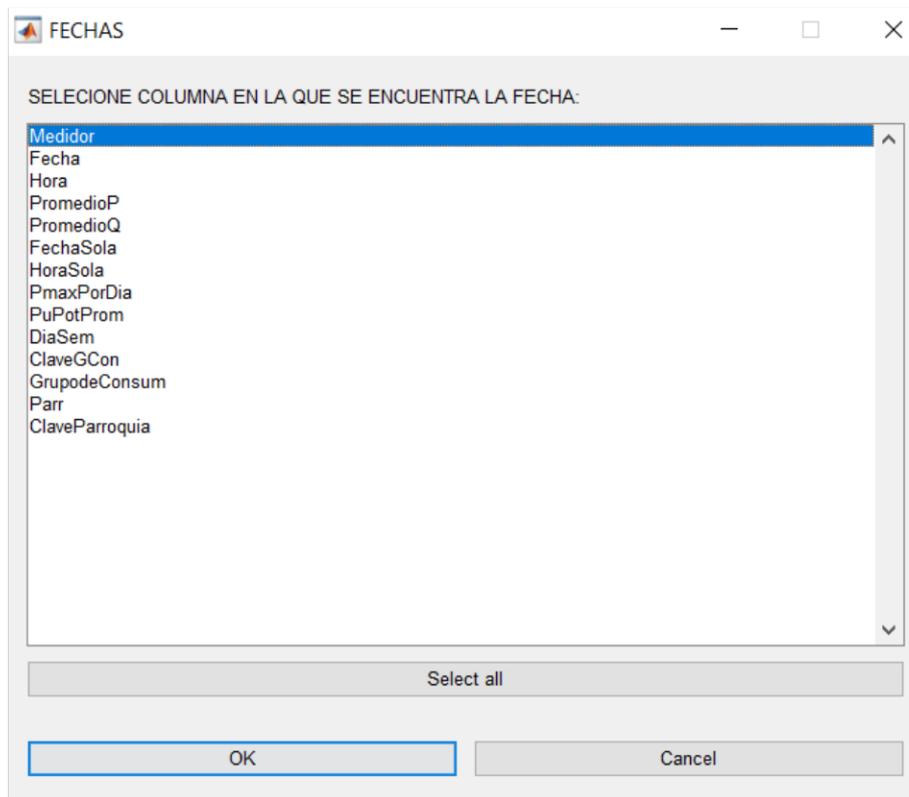


Figura 2.3. Cuadro de diálogo, función MFL_Organizar.

2.4.3 Respaldo de datos

Estas líneas de código se agregan con el fin de realizar varias clasificaciones, para esto se debe cambiar las variables “C_1” y “c_1”, por los nombres que se desee, sin embargo, estos mismos nombres deben asignarse a las variables “D” y “d” de la siguiente forma: “D = C_1;”y “d = c_1;”, caso contrario el programa no reconocerá la nueva clasificación que se haya hecho.

2.4.4 Corrección de datos

En esta parte del script principal, se eliminan las curvas de carga con ceros, además se asegura que todas las potencias sean positivas, estas líneas de código cambian el tamaño de la variable “*D*,” al eliminar las curvas de carga con falta de datos, también cambia el tamaño de la variable “*d*,” ya que la misma contiene las direcciones de los datos.

2.4.5 Normalizar datos

En esta sección del código principal se normaliza los datos de 0 a 1 con respecto a la máxima potencia de cada curva de carga. Si se desea realizar otro tipo de normalización se debe cambiar estas líneas de código, se puede utilizar la función “*max()*” la misma que permite calcular el valor máximo de toda una matriz, o el valor máximo de una columna como en nuestro caso e inclusive el valor máximo de una fila.

2.4.6 Calcular constante de ponderamiento

Estas líneas de código son simplificadas por el uso de las funciones “*var()*” y “*mean*”, las mismas son programadas de forma que la primera obtiene la varianza de todos los datos de una fila y la segunda el promedio de los datos de toda una columna.

2.4.7 Técnica de agrupamiento MFL

Aquí se desarrolla la estructura principal del algoritmo MFL, antes de ejecutar esta parte del script se debe fijar el valor del umbral en la variable “*u = 1.5;*”, después de este punto no se debe realizar ningún cambio en la programación. La estructura inicia con el cálculo de la distancia euclidiana ponderada, después se divide en dos partes: la primera consiste en la asignación de datos a un determinado grupo por primera vez, por lo tanto solo se ejecuta cuando el contador de iteraciones es igual a uno; la segunda parte es la reasignación de datos a un grupo determinado, la misma se ejecuta desde la iteración dos hasta que no exista un cambio de datos entre los grupos, la variable que guarda el movimiento de cada dato a un determinado grupo es “*gd*”. Se diseña el algoritmo para que se ejecute cien veces antes de salir del bucle o cuando no exista ningún cambio en la variable “*gd*”.

2.4.8 Calcular índice de adecuación MIA

Se crea la función MFL_MIA, la misma necesita las variables “*centros*” y “*grupos*” como parámetros de entrada y devuelve el valor del índice de adecuación MIA. Para calcular la distancia euclidiana normal que necesita el índice de adecuación MIA se crea la función

MFL_DE, la misma permite calcular la distancia euclidiana entre un dato y varios datos pertenecientes a un grupo, lo que facilita enormemente el cálculo del índice MIA.

2.4.9 Calcular índice de adecuación CDI

Se crea la función MFL_CDI, la misma que utiliza las variables “centros” y “grupos” como parámetros de entrada y devuelve el valor del índice de adecuación CDI. Esta función reduce enormemente el cálculo del índice de adecuación utilizando la función MFL_DE dos veces, una para calcular el numerador y la otra para el denominador, haciendo que sean más entendibles las líneas de código.

2.4.10 Graficar centros

Se crea la función MFL_Graficar_C exclusiva para graficar los centros, ya que se desea asignar a cada color de curva de carga representativa una identificación, el parámetro de entrada para esta función es el conjunto de todos los centros encontrados “centros”, el resultado de la función se lo visualiza en la Fig. 2.4.

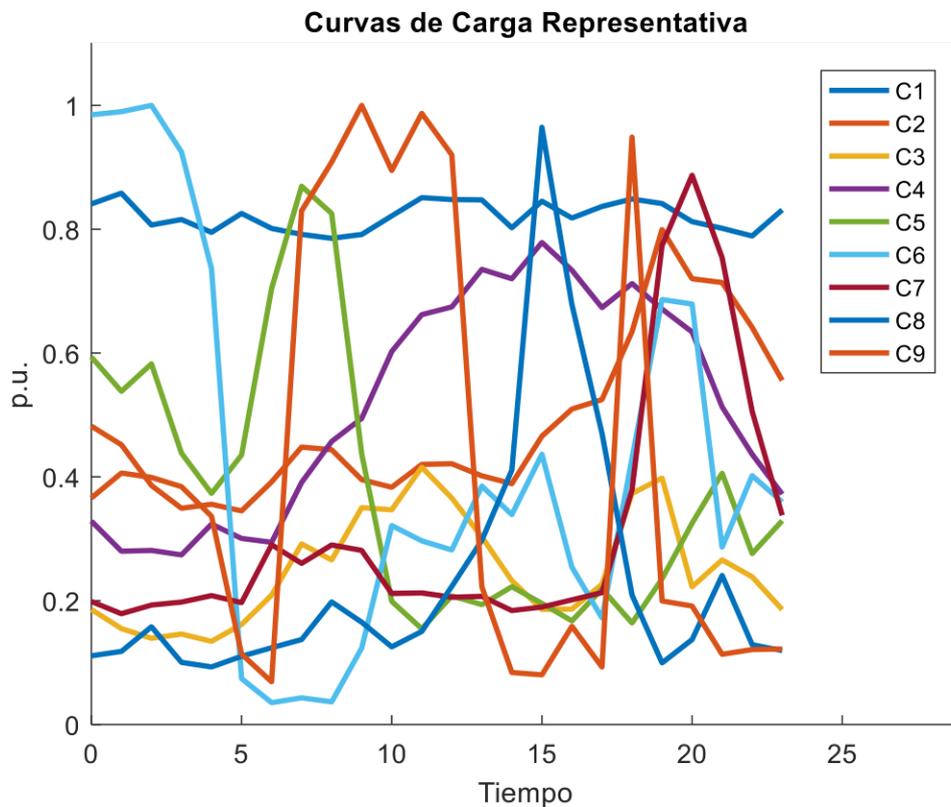


Figura 2.4. Resultado de la función MFL_Graficar_C.

2.4.11 Graficar grupos

La función MFL_Graficar_G, permite graficar cada curva de carga representativa con sus respectivos miembros, además se despliega en cada gráfica el número de miembros de cada grupo junto al valor del umbral y los índices de adecuación. Los parámetros de entrada de esta función son: la matriz que contiene todos los centros, la matriz que contiene todos los miembros de los grupos, el umbral y los índices de adecuación, la curva de carga representativa se la grafica resaltada y con color negro en Fig. 2.5.

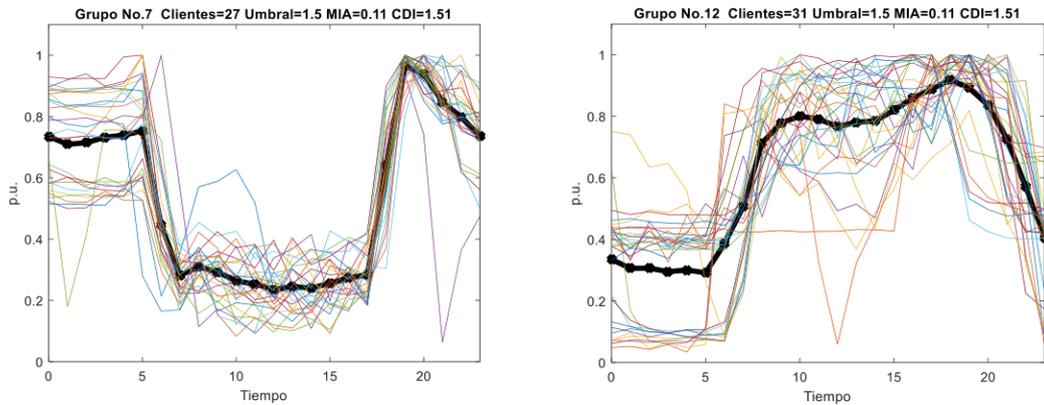


Figura 2.5. Resultado de la función MFL_Graficar_G.

2.4.12 Detalle de miembros de cada grupo

1	2	3	4	5	6	7	8	9	10	11	12	13	14
Medidor	Fecha	Hora	PromedioP	PromedioQ	FechaSola	HoraSola	PmaxPorDia	PuPotProm	DiaSem	ClaveGCon	GrupodeConsum	Parr	ClaveParroquia
91	'300695'	'43476.08...	NaN	1.1582e+03	1.4633e+03	'2019-01-11'	2	4.0962e+03	0.2827	5	20 'Comercial'	'JAVIER L.	22
92	'300695'	'43479.08...	NaN	496.7950	1.0591e+03	'2019-01-14'	2	4.0685e+03	0.1221	1	20 'Comercial'	'JAVIER L.	22
93	'300695'	'43480.08...	NaN	1.1323e+03	1.4303e+03	'2019-01-15'	2	4.4767e+03	0.2529	2	20 'Comercial'	'JAVIER L.	22
94	'300695'	'43481.08...	NaN	488.0180	1.0761e+03	'2019-01-16'	2	4.2225e+03	0.1156	3	20 'Comercial'	'JAVIER L.	22
95	'300695'	'43482.08...	NaN	924.5420	1.4514e+03	'2019-01-17'	2	4.1162e+03	0.2246	4	20 'Comercial'	'JAVIER L.	22
96	'300695'	'43483.08...	NaN	946.5740	1.4413e+03	'2019-01-18'	2	4.2958e+03	0.2203	5	20 'Comercial'	'JAVIER L.	22
97	'300803'	'43186.08...	NaN	96.4256	442.0310	'2018-03-27'	2	1.3563e+03	0.0711	2	20 'Comercial'	'AZOGUES'	11
98	'300803'	'43187.08...	NaN	158.8970	597.2040	'2018-03-28'	2	1.1870e+03	0.1339	3	20 'Comercial'	'AZOGUES'	11
99	'400042'	'43452'	0.0833	2.9501e+03	-4.9365e+03	'2018-12-18'	2	6.7710e+03	0.4357	2	20 'Comercial'	'AZOGUES'	11
100	'400042'	'43453'	0.0833	2.8603e+03	-4.8303e+03	'2018-12-19'	2	7.3950e+03	0.3868	3	20 'Comercial'	'AZOGUES'	11
101	'3'	[]	0	0	0 []	0	0	0	0	0	0 []	[]	0
102	'11212319'	'43445.08...	NaN	9.3637	51.1662	'2018-12-11'	2	1.2250e+04	7.6440e-04	2	20 'Comercial'	'JAVIER L.	22
103	'11212319'	'43446.08...	NaN	9.6880	52.3396	'2018-12-12'	2	1.2155e+04	7.9704e-04	3	20 'Comercial'	'JAVIER L.	22
104	'11212319'	'43447.08...	NaN	10.4386	51.9310	'2018-12-13'	2	1.2079	8.6419e-04	4	20 'Comercial'	'JAVIER L.	22
105	'11212319'	'43448.08...	NaN	8.9316	46.7191	'2018-12-14'	2	1.2294e+04	7.2649e-04	5	20 'Comercial'	'JAVIER L.	22
106	'11212319'	'43451.08...	NaN	12.2344	47.0682	'2018-12-17'	2	1.2234e+04	1.0000e-03	1	20 'Comercial'	'JAVIER L.	22
107	'11212319'	'43452.08...	NaN	11.3922	48.5300	'2018-12-18'	2	1.2155e+04	9.3727e-04	2	20 'Comercial'	'JAVIER L.	22
108	'11212319'	'43453.08...	NaN	11.2900	48.1582	'2018-12-19'	2	1.2143e+04	9.2976e-04	3	20 'Comercial'	'JAVIER L.	22
109	'11212319'	'43454.08...	NaN	10.1286	52.0367	'2018-12-20'	2	1.2396e+04	8.1711e-04	4	20 'Comercial'	'JAVIER L.	22
110	'4'	[]	0	0	0 []	0	0	0	0	0	0 []	[]	0
111	'11212322'	'43264.08...	NaN	2.4541e+03	3.5246e+03	'2018-06-13'	2	1.4046e+04	0.1747	3	20 'Comercial'	'AZOGUES'	11
112	'11212322'	'43265.08...	NaN	1.8930e+03	3.0843e+03	'2018-06-14'	2	1.1445e+04	0.1654	4	20 'Comercial'	'AZOGUES'	11
113	'11212322'	'43266.08...	NaN	1.7371e+03	2.9162e+03	'2018-06-15'	2	1.1086e+04	0.1567	5	20 'Comercial'	'AZOGUES'	11
114	'11212322'	'43269.08...	NaN	1.5157e+03	2.4711e+03	'2018-06-18'	2	1.1588e+04	0.1308	1	20 'Comercial'	'AZOGUES'	11
115	'11212322'	'43270.08...	NaN	1.7652e+03	2.6254e+03	'2018-06-19'	2	1.2350	0.1429	2	20 'Comercial'	'AZOGUES'	11
116	'11212322'	'43271.08...	NaN	1.9058e+03	2.8149e+03	'2018-06-20'	2	1.2688e+04	0.1502	3	20 'Comercial'	'AZOGUES'	11
117	'11212322'	'43272.08...	NaN	1.7398e+03	2.5662e+03	'2018-06-21'	2	1.4299e+04	0.1217	4	20 'Comercial'	'AZOGUES'	11

Figura 2.6. Resultado de la función MFL_Miembros.

Esta función se crea con la finalidad de ver cuáles son los miembros de cada grupo. La función tiene como parámetros de entrada, la matriz con todos los centros, las direcciones de cada potencia activa de la curva de carga diaria, la tabla inicial de la base de datos y el vector en donde se encuentran las asignaciones de un dato a un determinado grupo. En la Fig. 2.6. se puede observar el resultado de esta función, donde cada fila que inicia con: un número y todas las demás celdas son cero o vacío indican el inicio de un nuevo grupo; además cada fila representa una curva de carga diaria.

2.4.13 Función MFL_Distancia

Esta función utiliza los parámetros de entrada: curva de carga diaria " $D(:, i)$ ", la matriz de centros "*centros*" y la constante de ponderamiento para cada característica de un dato " k ". Como resultado la función devuelve el valor de la distancia euclidiana más pequeña y el grupo con el que se obtuvo dicho valor. En el caso que no se tenga ningún centro, la función devuelve el valor de " $DE = 0$ " y " $g = 1$ ", lo que significa que es la primera vez que se ejecuta esta función en el algoritmo, para todos los demás casos se ejecuta normalmente la función. Esta función se ejecuta en el script principal y es parte esencial de toda la técnica de agrupamiento MFL.

2.4.14 Función MFL_Asignar

La función MFL_Asignar tiene como parámetros de entrada: el umbral " u ", la matriz de grupos "*grupos*", la matriz de centros "*centros*", la curva de carga diaria que se asignará a un determinado grupo " $D(:, i)$ ", la distancia euclidiana ponderada " DE " y el grupo con el que se obtuvo esa distancia " g ". La función devuelve las variables: " $gd(i)$ ", que es el vector que almacena a que grupo fue asignada la curva de carga diaria, donde si en la primera celda se encuentra el valor de " n ", significará que la curva de carga diaria número 1 se asignó al grupo " n "; la matriz grupos que contiene todos los grupos con sus respectivos datos y la matriz centros. La programación interna de la función se encarga de asignar la curva de carga analizada a un grupo y calcular el nuevo centro de cada grupo después de comparar la distancia euclidiana ponderada con el umbral. También se encarga de crear un nuevo grupo y asignar la curva tratada como centro y miembro del nuevo grupo, si la mínima distancia es mayor al umbral.

2.4.15 Función MFL_Reasignar

La función MFL_Reasignar utiliza los mismos parámetros de entrada de la función MFL_Asignar, aumentando las variables: " gd " que es el vector que contiene las

asignaciones de los datos a un grupo; y la constante de ponderamiento “ k ”. Esta función se encarga de reasignar las curvas de carga diaria a diferentes grupos. La parte interna de esta función se divide en tres casos: el primero cuando la distancia euclidiana ponderada es menor o igual que el umbral y el grupo de la curva de carga al que se asignó no coincide con el grupo al que se le quiere asignar, en este caso se debe retirar la curva de carga del grupo anterior, para esto se utiliza la función `MFL_Eliminar`, la misma que retira una curva de carga de un grupo determinado, después se actualiza el centro del grupo del cual se retira la curva, para finalmente agregar el dato al nuevo grupo y recalculando el nuevo centro de dicho grupo; el segundo caso se ejecuta cuando la mínima distancia euclidiana ponderada es mayor al umbral y el grupo al que se desea asignar la curva de carga no coincide con el grupo al que fue asignado previamente, en esta parte se debe retirar la curva de carga del grupo al que fue asignado anteriormente, recalculando el centro de dicho grupo, después se crea un nuevo grupo y se asigna la curva de carga como primer miembro del grupo y centro del mismo; el tercer caso se da cuando la distancia euclidiana ponderada es igual a cero y el grupo al que se quiere agregar la curva de carga coincide con el grupo asignado previamente, esto quiere decir que en el grupo solo existe esta curva de carga por tal motivo se vuelve a calcular la distancia euclidiana ponderada sin considerar el centro de este grupo, ya que al existir solo un dato en el grupo y buscar la distancia euclidiana ponderada entre si mismo, siempre me dará cero, por tal motivo no se considera dicho centro y se evalúa con respecto a los demás centros, después se cambia a un grupo ya creado solo si la distancia euclidiana ponderada es menor o igual al umbral, caso contrario no tiene sentido mover la curva de carga diaria a otro grupo.

2.4.16 Función `MFL_Eliminar`

Esta función se crea para eliminar un dato de un grupo específico, los parámetros de entrada son: la matriz de grupos “*grupos*”, el dato que se va a eliminar del grupo “*dato*” y el grupo en el que se encuentra el dato, información que es proporcionada por el vector “*gd*”, devolviendo la matriz grupos después de eliminar el dato.

2.4.17 Función `MFL_DE`

La función `MFL_DE` ayuda a calcular la distancia euclidiana entre un dato versus un conjunto de datos, lo que facilita el cálculo de los índices MIA y CDI. La función necesita los parámetros de entrada: “*d*” que es el dato fijo con el que se calculará la distancia euclidiana, la matriz grupos “*grupos*” y “*g*” que es una variable de control, es decir puede contener el grupo que se desea del conjunto total de grupos o puede ser vacío, cuando ocurre el primer caso significa que la función calculará la distancia euclidiana del dato vs

todos los datos del conjunto especificado; en el segundo caso quiere decir que la función recibirá como conjunto de datos la matriz centros, esta matriz no necesita ser condensada con la función MFL_Condensar, por tal motivo el algoritmo en esta ocasión calculará la distancia euclidiana entre una curva de carga representativa vs todas las curvas de carga representativa o matriz de centros.

2.4.18 Función MFL_Condensar

Debido a que la matriz grupos es creada con un número de columnas igual al número de datos total en análisis, y un número de filas igual al producto de 24 por el número total de datos en análisis, considerando que la matriz contendrá las curvas de carga asignadas a cada grupo, por ende puede darse el caso que el grupo uno contenga todos los datos, por tal motivo el número de columnas de la matriz grupos tiene dicha longitud; también puede darse el caso en el que cada dato formó su propio grupo, por tal motivo se crea la matriz grupos con tal número de filas; esto debido a que para MatLab cuando trabaja con bucles es más rápido asignar un espacio de memoria que después se puede ir editando, que aumentar la dimensión de una matriz, ya que este último caso conlleva a que se busque un espacio de memoria en donde copiar todos los datos nuevos y después recorrer todos los espacios de memoria para ocupar el vacío que se crea al aumentar la dimensión de la matriz, con el fin de que cada variable ocupe una posición continua en los espacios de memoria. Por tal motivo la función MFL_Condensar reduce a la matriz grupos a una matriz con un número de columnas igual al número de datos que contiene y con un número de filas igual a 24 ya que se tiene una potencia por cada hora del día. Los parámetros de entrada son: la matriz grupos "*grupos*" y el grupo que se desea condensar "*g*", la variable de salida es "*g*" en donde se almacena la matriz condensada, una vez más MatLab sugiere que se reutilice la variable "*g*", ya que no se encuentra en un bucle y fácilmente se le puede buscar un espacio de memoria.

3 RESULTADOS, RECOMENDACIONES

CONCLUSIONES

Y

3.1 Resultados

3.1.1 Consumidores residenciales de lunes a viernes

Para los consumidores residenciales de lunes a viernes con un umbral de 1,5; se obtiene los índices MIA de 0.17 y CDI de 3,19; para un total de 11 grupos, como se observa en la Fig. 3.1. Un resumen de los grupos representativos se presenta en la Fig. 3.2.

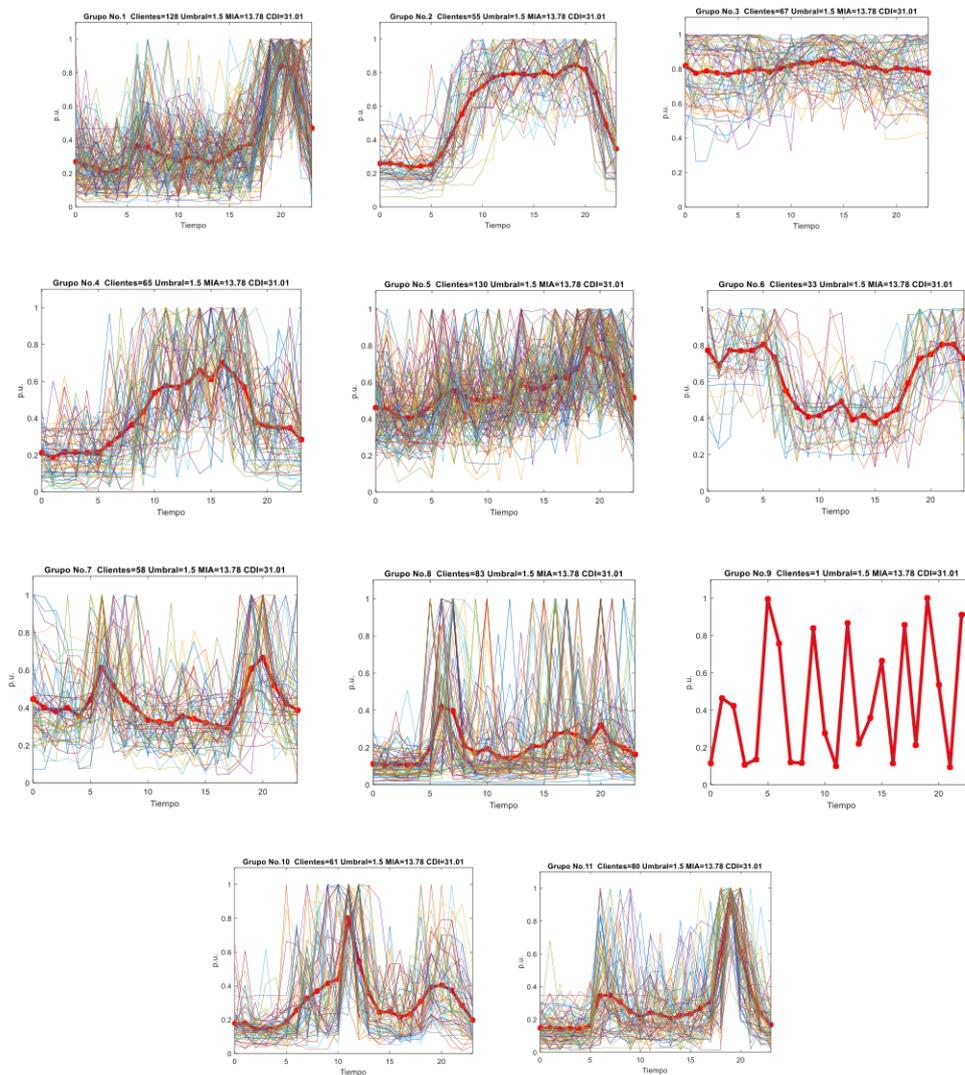


Figura 3.1. Curvas de carga representativa junto a sus curvas de carga diaria, para cada grupo pertenecientes a consumidores tipo residencial de lunes a viernes.

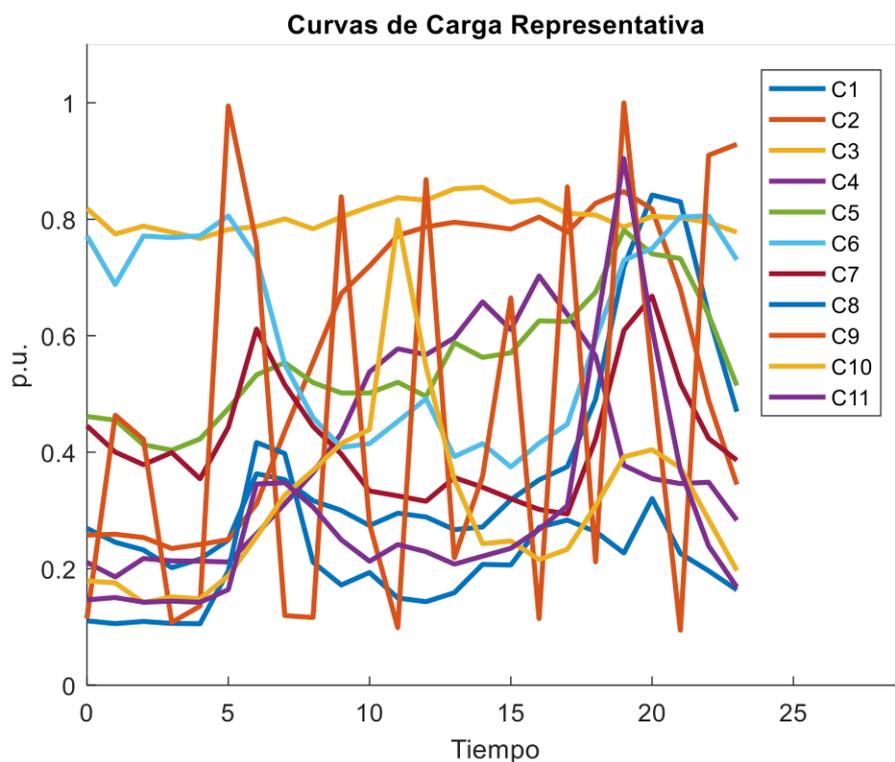


Figura 3.2. Curvas de carga representativa para cada grupo, pertenecientes a consumidores tipo residencial de lunes a viernes.

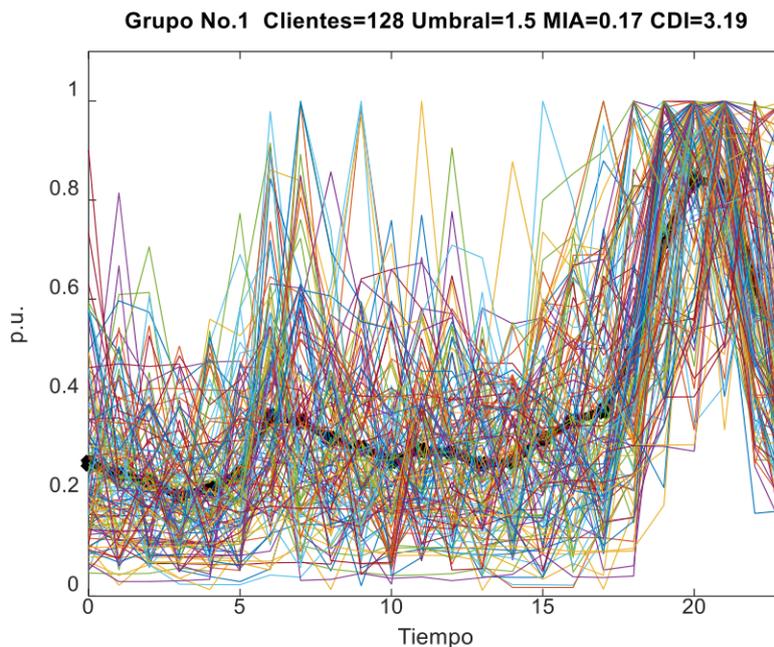


Figura 3.3. Curva de carga representativa del grupo uno junto a sus curvas de carga diaria, consumidores tipo residencial de lunes a viernes.

A los consumidores del grupo uno Fig. 3.3., para identificar la forma de la curva de carga representativa, se los denomina tipo A, donde su curva es casi plana de 00:00 a 18:00 horas, donde empieza a aumentar la demanda hasta llegar a las 20:00 horas, cuando alcanza el pico máximo y empieza a bajar el consumo hasta las 00:00. Se puede decir que estos consumidores pasan la mayor parte del día fuera del hogar, por lo tanto se puede clasificar el tipo A como consumidores que pertenecen al grupo productivo de oficinas.

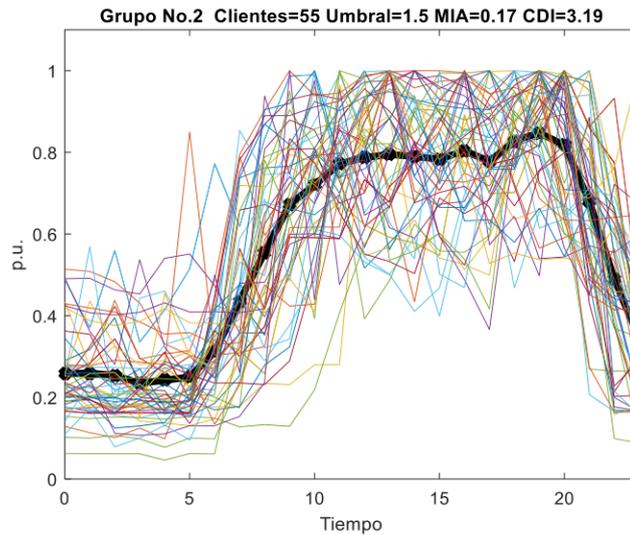


Figura 3.4. Curva de carga representativa del grupo dos junto a sus curvas de carga diaria, consumidores tipo residencial de lunes a viernes.

En la Fig. 3.4. se observa al grupo dos y se los denomina como consumidores tipo B, los cuales permanecen la mayor parte del tiempo en sus hogares.

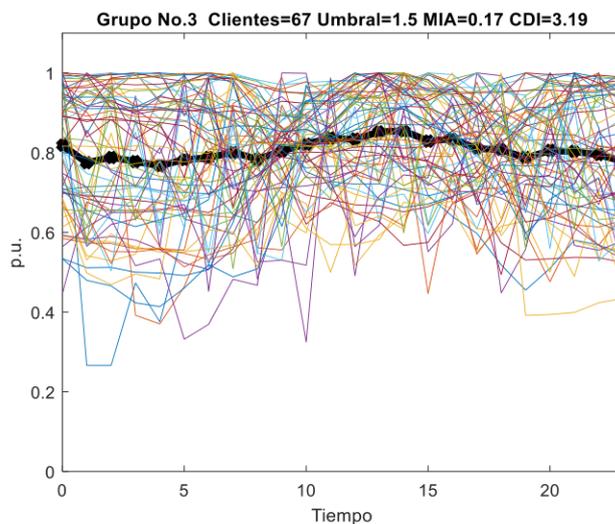


Figura 3.5. Curva de carga representativa del grupo tres junto a sus curvas de carga diaria, consumidores tipo residencial de lunes a viernes.

En Fig. 3.5. se observa al grupo tres y se los denomina como consumidores tipo C, debido a la forma de la curva de carga representativa estos consumidores tienen un consumo similar al de un país industrializado, donde todo el día presentan la misma demanda de energía eléctrica.

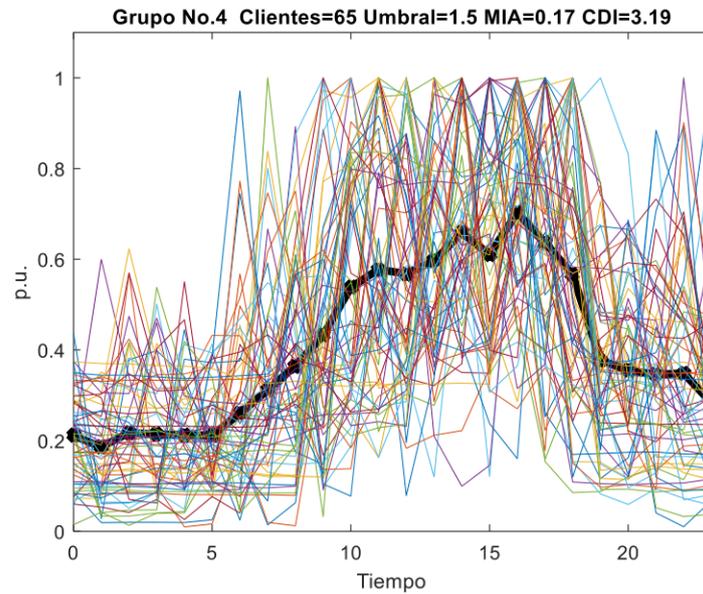


Figura 3.6. Curva de carga representativa del grupo cuatro junto a sus curvas de carga diaria, consumidores tipo residencial de lunes a viernes.

La Fig. 3.6. perteneciente al grupo cuatro, se la denomina tipo B.1. debido a su similitud con el tipo B, sin embargo, se hace énfasis que el grupo B.1. presenta un menor consumo de energía.

La Fig. 3.7. es la curva de carga representativa para el grupo cinco, su forma se asemeja a los consumidores del tipo A, por lo tanto, se la denomina como tipo A.1. teniendo en cuenta que este nuevo grupo de consumidores al tener mayor demanda eléctrica, se podría deducir que se trata de trabajadores de oficina con mayor remuneración económica.

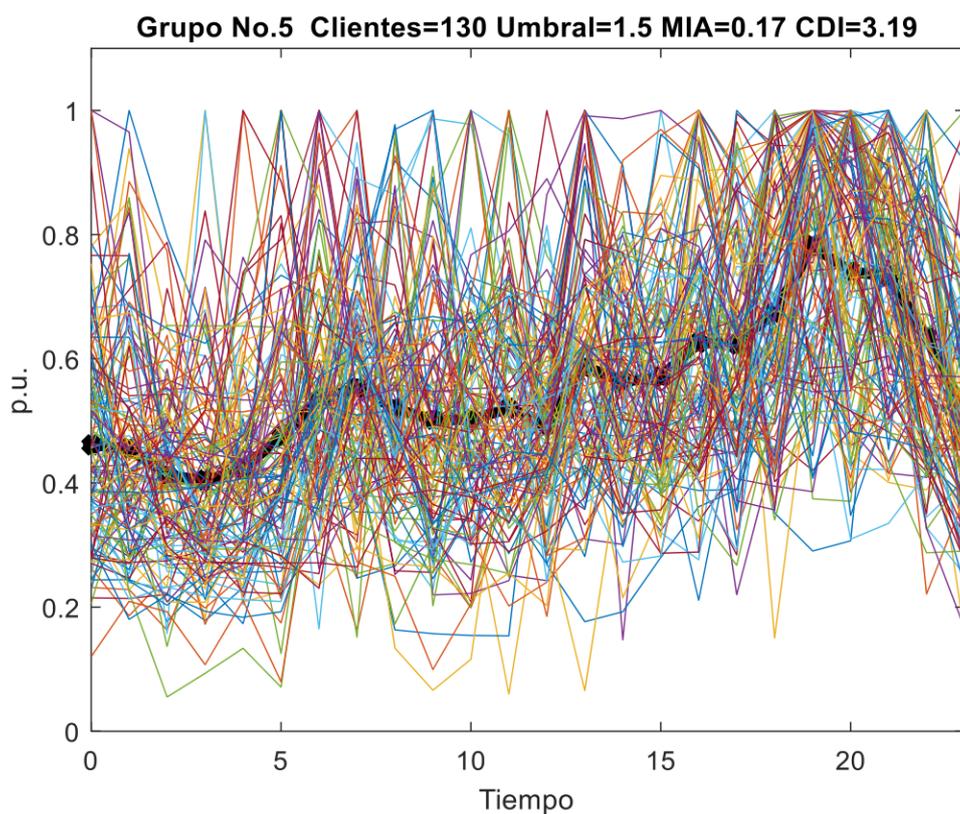


Figura 3.7. Curva de carga representativa del grupo cinco junto a sus curvas de carga diaria, consumidores tipo residencial de lunes a viernes.

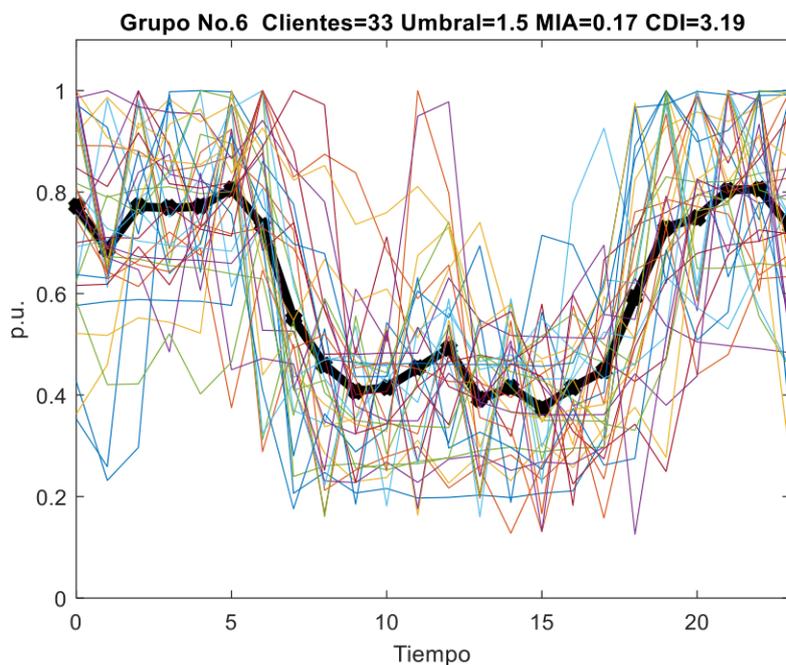


Figura 3.8. Curva de carga representativa del grupo seis junto a sus curvas de carga diaria, consumidores tipo residencial de lunes a viernes.

En la Fig. 3.8. se puede apreciar la curva de carga característica del grupo seis, la misma que se la denomina tipo C.1., debido a que la forma de la curva de carga fuera de tipo C si no se presentara el valle en horas laborables.

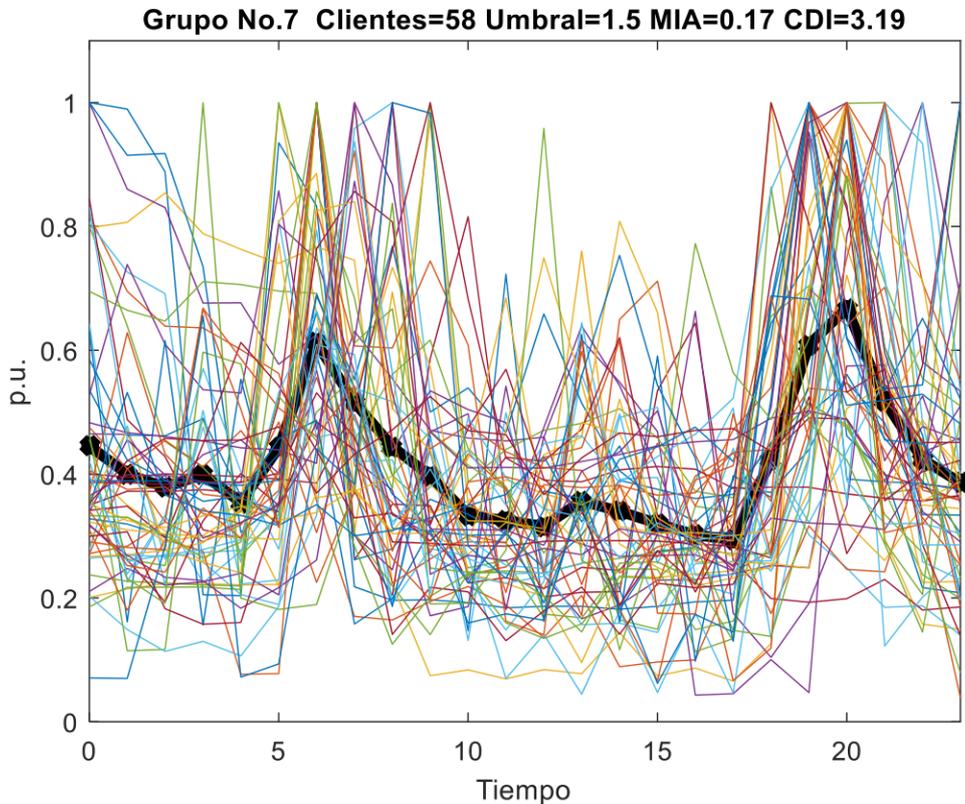


Figura 3.9. Curva de carga representativa del grupo siete junto a sus curvas de carga diaria, consumidores tipo residencial de lunes a viernes.

La Fig. 3.9. representa al grupo siete, se la denomina tipo A.2. debido a que su forma da información de consumidores que pasan la mayor parte del día fuera de casa como trabajadores del comercio y solo se reúnen a la hora de desayuno y merienda, dando información que posiblemente se refiere a trabajadores que viven solos o en pareja sin hijos.

La Fig. 3.10. representa al grupo ocho, se la denomina tipo A.3. debido a la similitud con A.2., se puede considerar que este sector de consumidores tiene un menor número de equipos eléctricos, debido a su menor demanda.

La Fig. 3.11. representa al grupo nueve, esta curva al tener un único miembro, se la considera un dato atípico.

La Fig. 3.12. representa al grupo diez, se le denomina tipo A.4. ya que su forma es similar al tipo A.3, sin embargo, presenta un pico a las 11:00 horas y otro a las 20:00 horas.

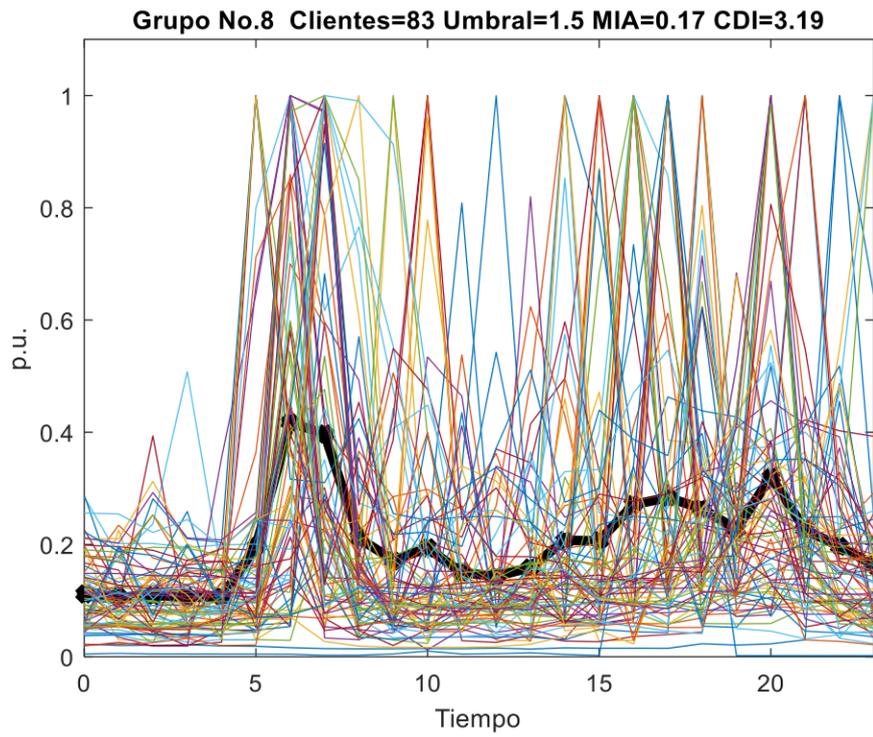


Figura 3.10. Curva de carga representativa del grupo ocho junto a sus curvas de carga diaria, consumidores tipo residencial de lunes a viernes.

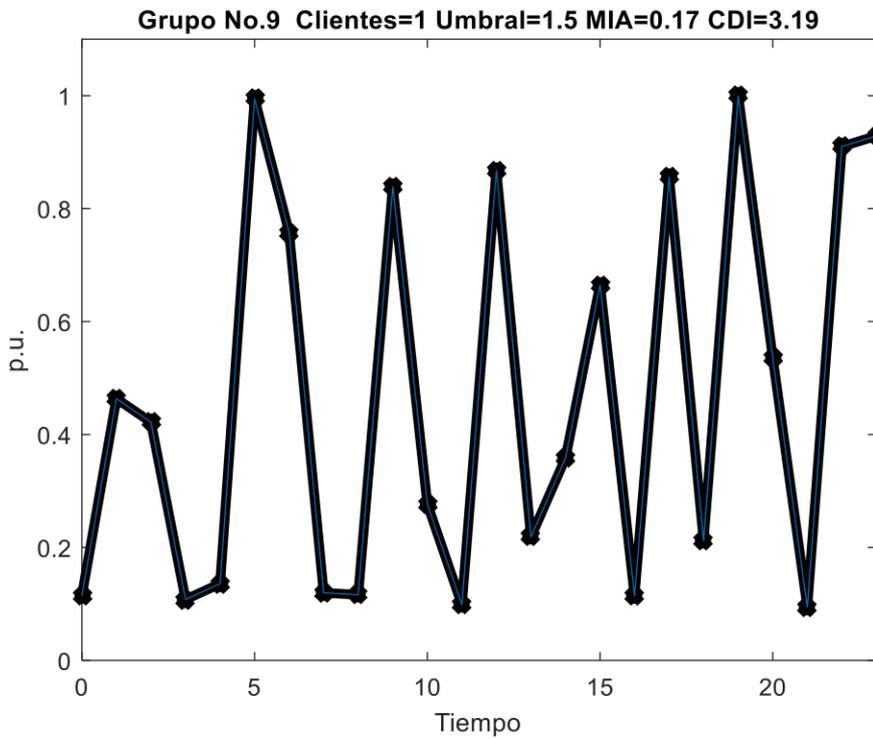


Figura 3.11. Curva de carga representativa del grupo nueve junto a sus curvas de carga diaria, consumidores tipo residencial de lunes a viernes.

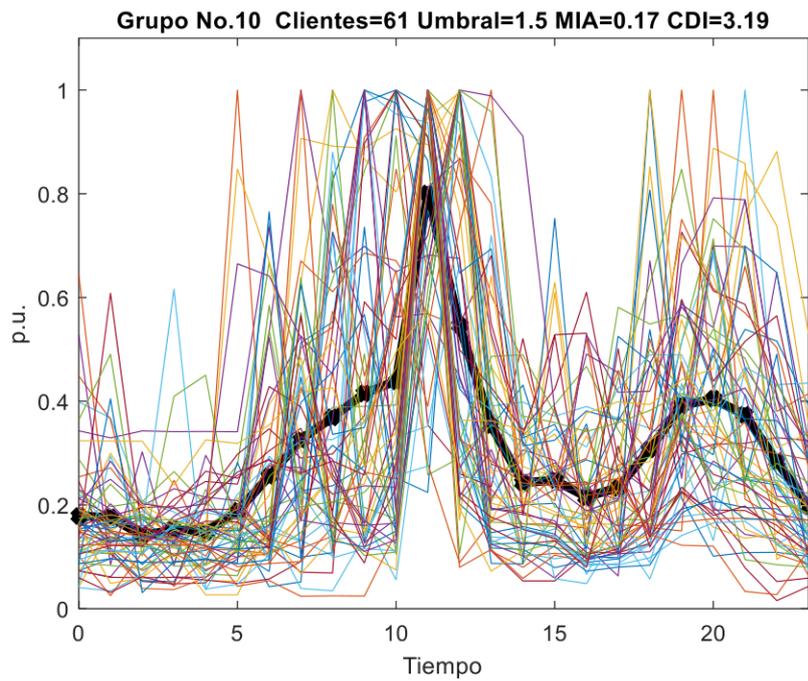


Figura 3.12. Curva de carga representativa del grupo diez junto a sus curvas de carga diaria, consumidores tipo residencial de lunes a viernes.

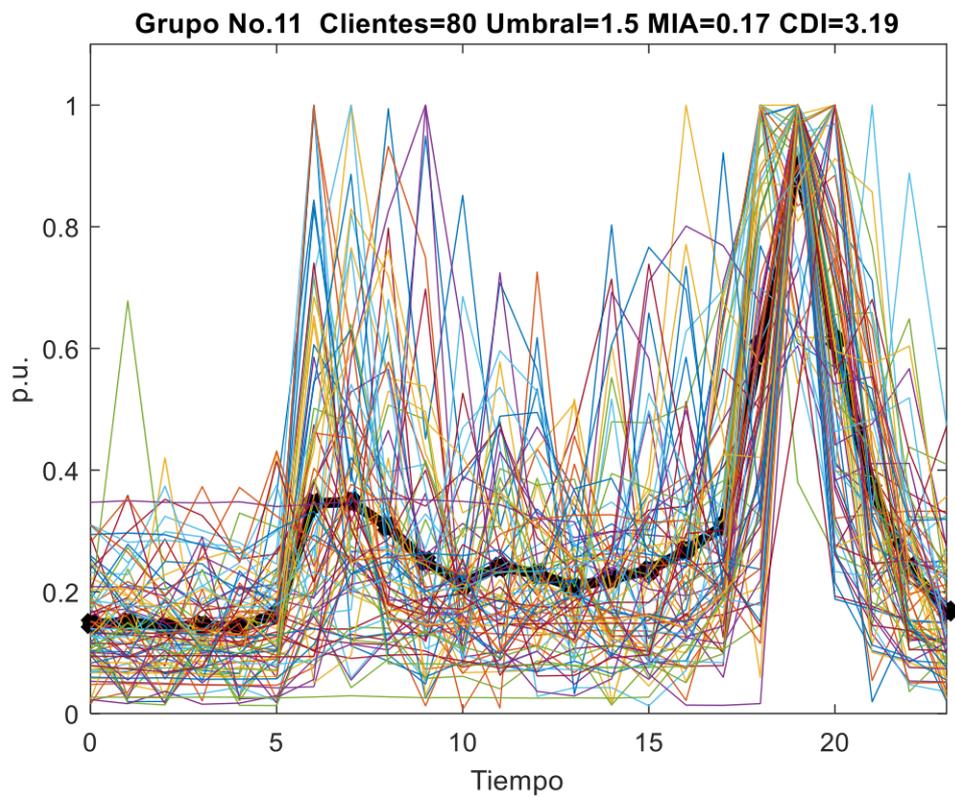


Figura 3.13. Curva de carga representativa del grupo diez junto a sus curvas de carga diaria, consumidores tipo residencial de lunes a viernes.

La Fig. 3.13. representa al grupo once y se la denomina tipo A.5., en este caso particular de trabajadores de oficina, se puede concluir que son hogares en los que viven más de dos personas, sin embargo, un grupo sale a trabajar en horarios de oficina y otro permanece en casa.

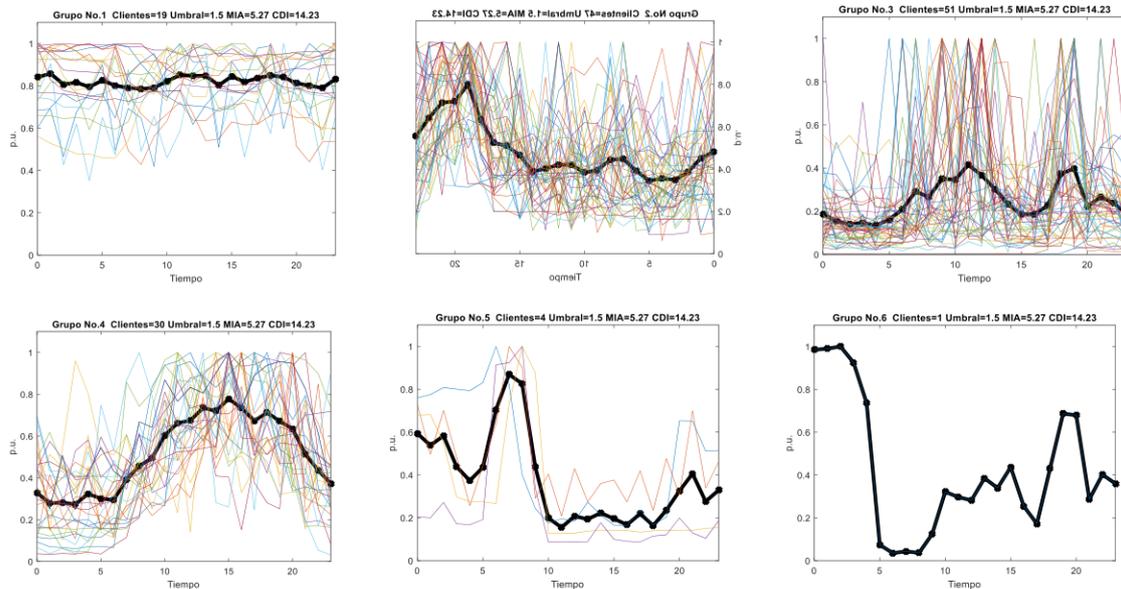
En la Tabla 3.1. se resumen la asignación de cada grupo a un tipo específico de consumidor.

Tabla 3.1. Grupos obtenidos junto a su tipo.

Grupo No.	Tipo
1	A
2	B
3	C
4	B.1.
5	A.1.
6	C.1.
7	A.2.
8	A.3.
9	Atípico
10	A.4.
11	A.5.

3.1.2 Consumidores residenciales para los días sábado

Para los consumidores residenciales días sábados Fig. 3.14. con un umbral de 1,5; se obtiene los índices MIA de 0,16 y CDI de 2,95; para un total de 9 grupos, como se observa a continuación.



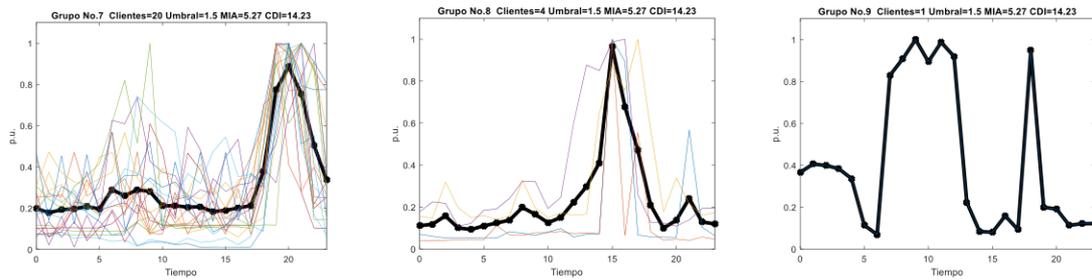


Figura 3.14. Curva de carga representativa junto a sus curvas de carga diaria, para cada grupo pertenecientes a consumidores tipo residencial días sábado.

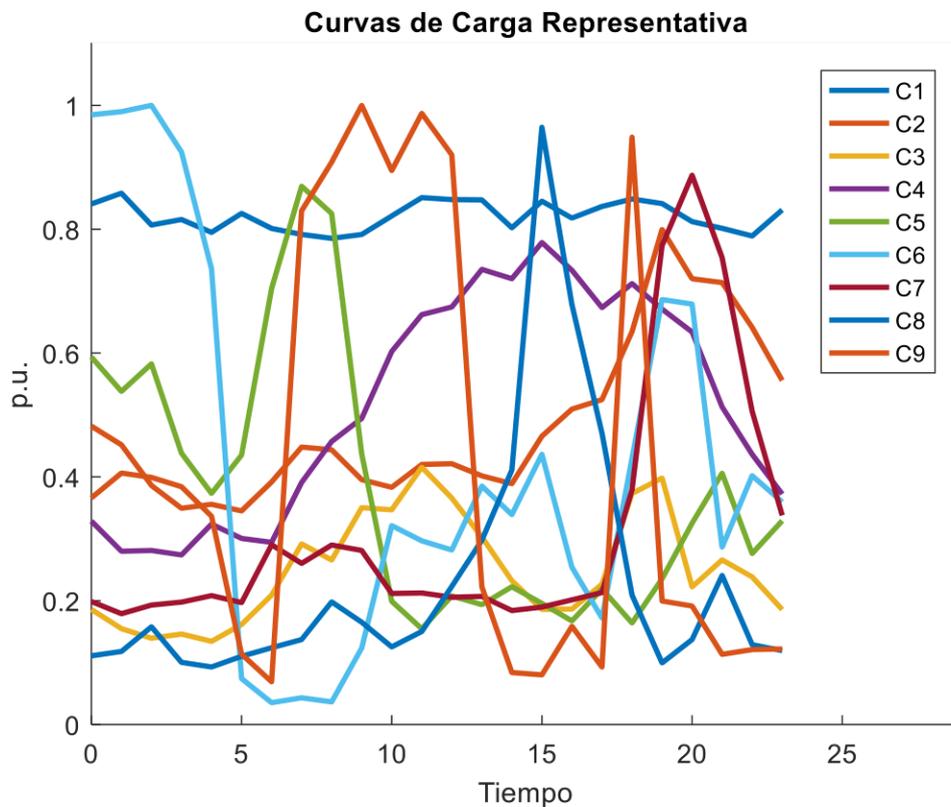


Figura 3.15. Curvas de carga representativa para cada grupo, pertenecientes a consumidores tipo residencial días sábado.

Como se puede observar en las Fig. 3.14. y Fig. 3.15. se tiene en total nueve grupos, dentro de los cuales el algoritmo a encontrado dos datos atípicos, y dos grupos con cuatro miembros cada uno, revisando la matriz MIEMBROS, se puede ver que el grupo cuatro, tiene cuatro clientes diferentes y el grupo ocho tiene tres clientes diferentes, razón por la cual no se puede considerar estos grupos como datos atípicos. Se encuentran los siguientes tipos de consumidores resumidos en la Tabla 3.2.

Tabla 3.2. Grupos obtenidos junto a su tipo.

Grupo No.	Tipo
1	C
2	A.3.
3	A.2.
4	B
5	A.4.
6	Atípico
7	A.1.
8	A
9	Atípico

3.1.3 Consumidores residenciales para los días domingo

Para los consumidores residenciales días domingo con un umbral de 1,5; se obtiene los índices MIA de 0,14 y CDI de 3,32; para un total de 9 grupos, como se observa a continuación.

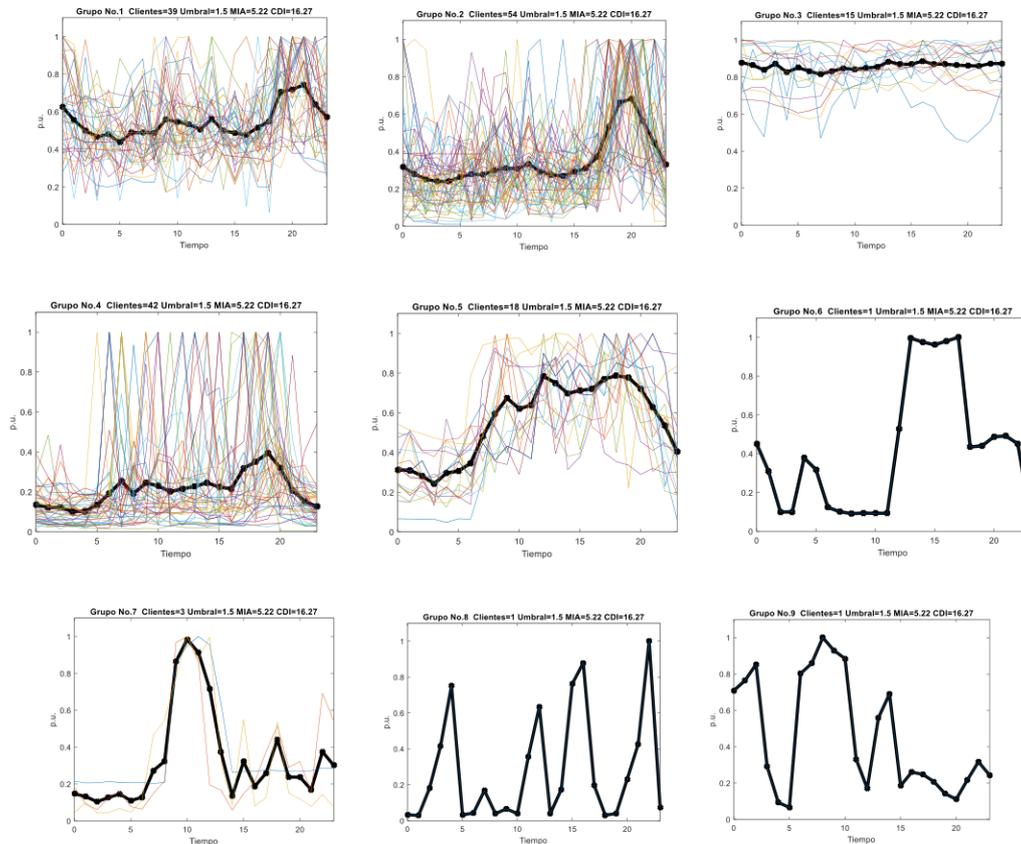


Figura 3.16. Curva de carga representativa junto a sus curvas de carga diaria, para cada grupo pertenecientes a consumidores tipo residencial días domingo.

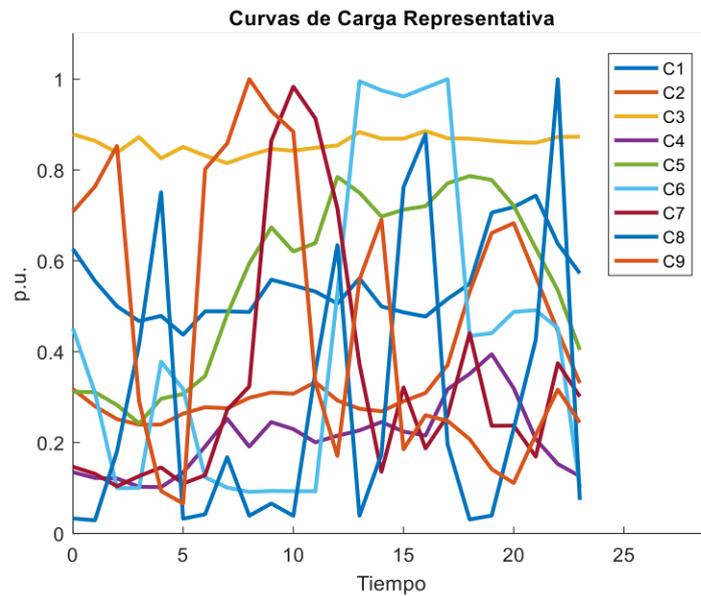


Figura 3.17. Curvas de carga representativa para cada grupo, pertenecientes a consumidores tipo residencial días domingo.

En las Fig. 3.16 y 3.17 se puede observar que el algoritmo a encontrado tres datos atípicos pertenecientes a los grupos seis, ocho y nueve, además presenta un grupo con tres datos, revisando la matriz MIEMBROS, se obtiene que los tres clientes son diferentes, razón por la cual no se los considera datos atípicos. Además se encuentran los siguientes tipos de consumidores resumidos en la Tabla 3.3.

Tabla 3.3. Grupos obtenidos junto a su tipo.

Grupo No.	Tipo
1	A.3.
2	A.1.
3	C
4	A.2.
5	B
6	Atípico
7	A.4.
8	Atípico
9	Atípico

3.1.4 Consumidores comerciales de lunes a viernes

Para los consumidores comerciales de lunes a viernes con un umbral de 1,5; se obtiene los índices MIA de 0,11 y CDI de 1,51; para un total de 13 grupos, como se observa a continuación.

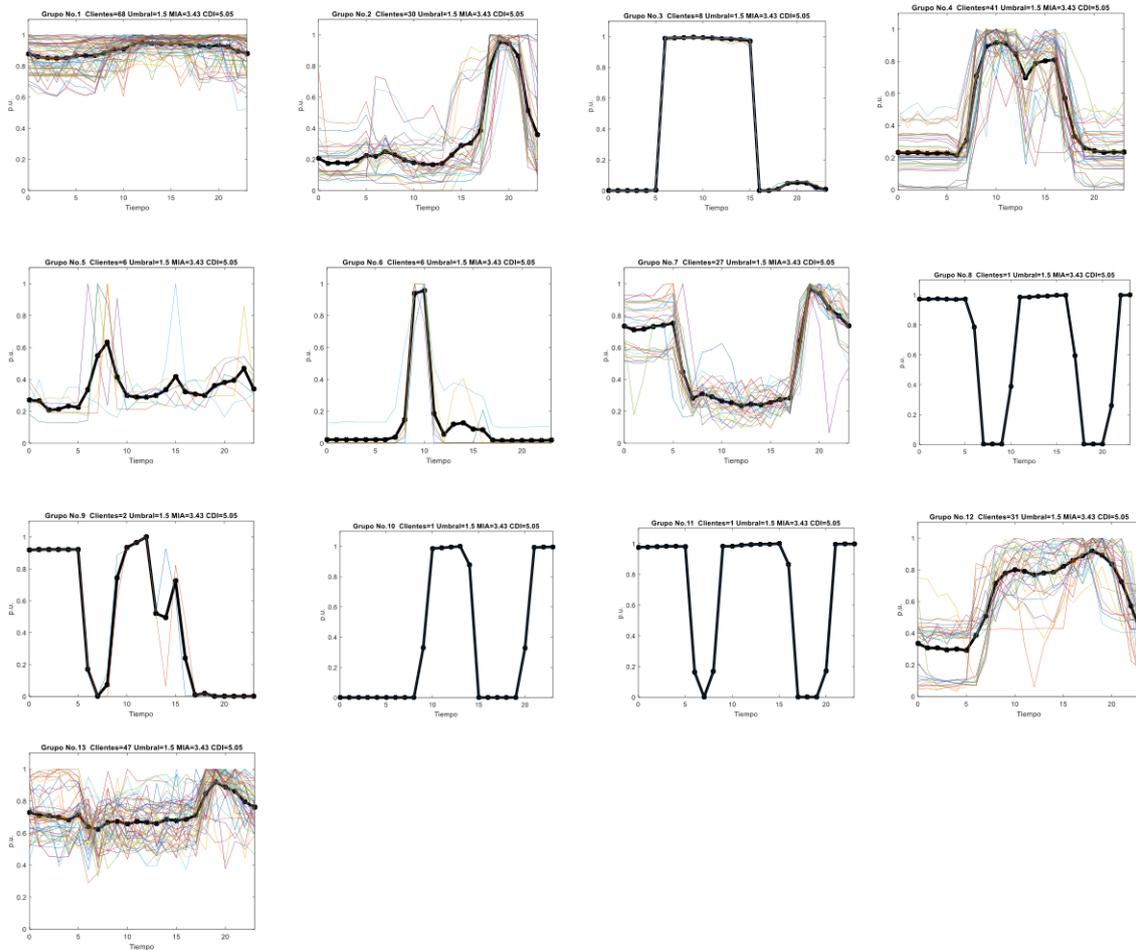


Figura 3.18. Curva de carga representativa junto a sus curvas de carga diaria, para cada grupo pertenecientes a consumidores tipo comercial de lunes a viernes.

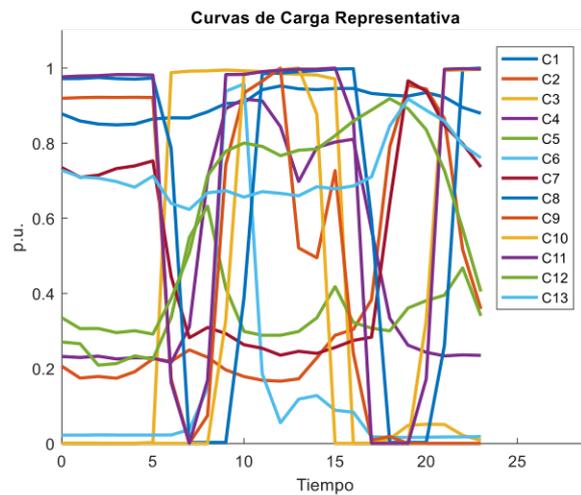


Figura 3.19. Curvas de carga representativa para cada grupo, pertenecientes a consumidores tipo comercial de lunes a viernes.

En las Fig. 3.18 y 3.19, se obtienen 13 grupos después de aplicar el algoritmo MFL, aquí se pueden encontrar diferentes tipos de comercios, razón por la cual se asignará un tipo de comercio a cada grupo.

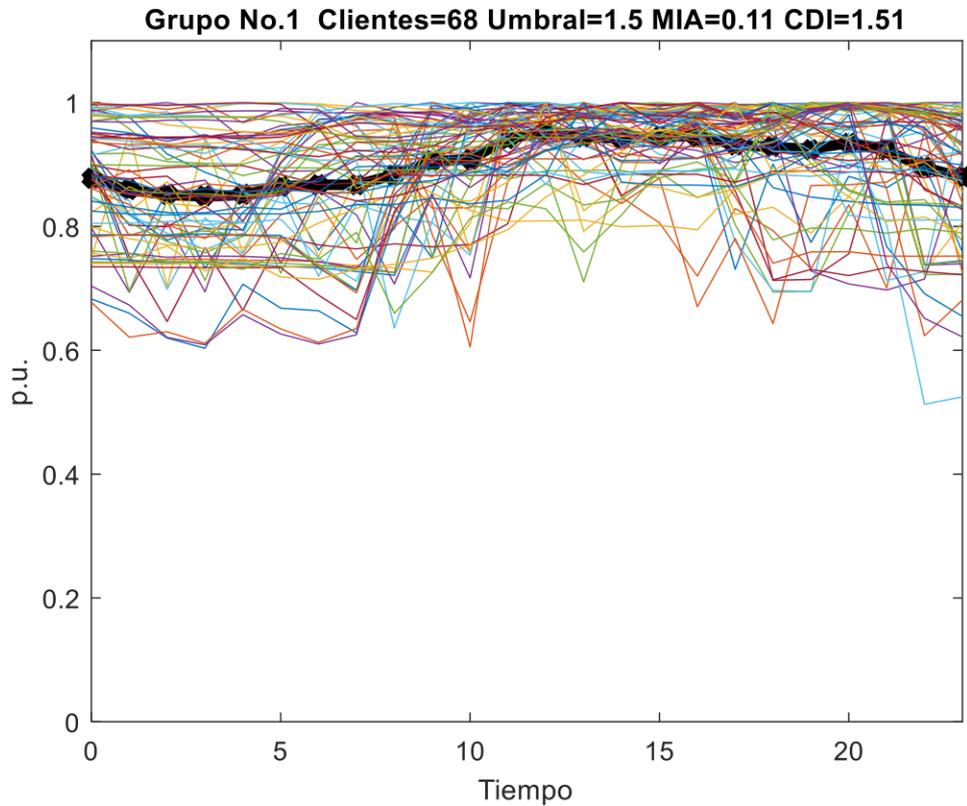


Figura 3.20. Curva de carga representativa del grupo uno junto a sus curvas de carga diaria, consumidores tipo comercial de lunes a viernes.

La Fig. 3.20. representa al grupo uno, a este tipo de consumidores para tener una similitud con los tipos asignados anteriormente, se le denomina tipo C, considerando que ahora los consumidores son comercios y no residencias.

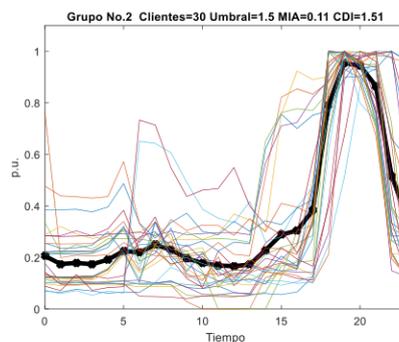


Figura 3.21. Curva de carga representativa del grupo dos junto a sus curvas de carga diaria, consumidores tipo comercial de lunes a viernes.

La Fig. 3.21. representa al grupo dos y se le asigna el tipo A, previamente explicado, haciendo referencia a comercios y no residencias.

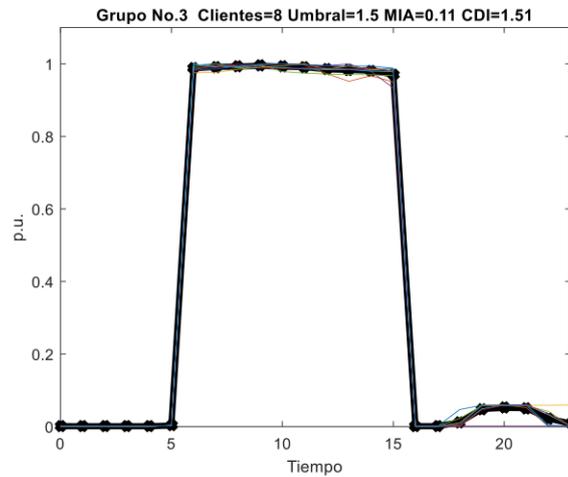


Figura 3.22. Curva de carga representativa del grupo tres junto a sus curvas de carga diaria, consumidores tipo comercial de lunes a viernes.

La Fig. 3.22. representa al grupo tres, sin embargo, sus ocho miembros pertenecen al mismo cliente, aún cuando no es un dato atípico, produce una nueva clasificación en la que se puede observar que se trata de un local comercial en el cual se tiene horas fijas de apertura y cierre, se la denomina tipo B.2. debido a su semejanza con la tipo B.

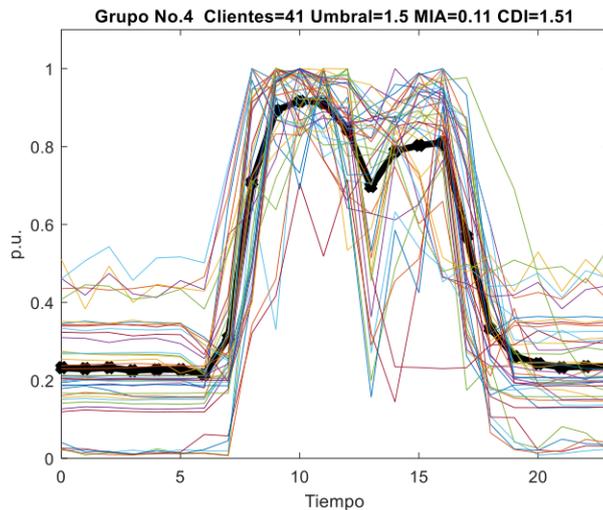


Figura 3.23. Curva de carga representativa del grupo cuatro junto a sus curvas de carga diaria, consumidores tipo comercial de lunes a viernes.

La Fig. 3.23. contiene la curva de carga representativa del grupo cuatro, que se asemeja al tipo B con dos picos a las 10:00 y 15:00 horas, a este tipo de comercios se denomina tipo B.3.

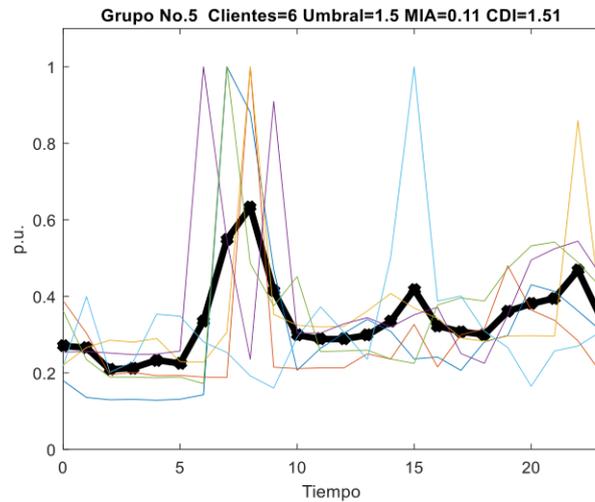


Figura 3.24. Curva de carga representativa del grupo cinco junto a sus curvas de carga diaria, consumidores tipo comercial de lunes a viernes.

La Fig. 3.24. representa al grupo cinco y se asemeja al tipo A.3., con la diferencia que los consumidores son comercios.

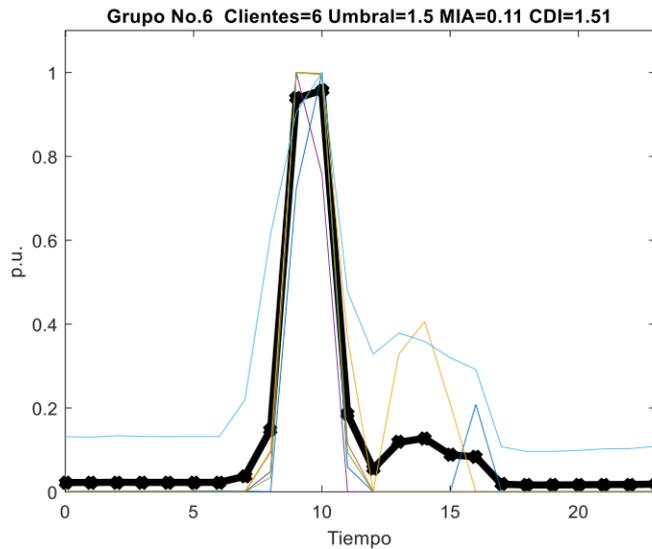


Figura 3.25. Curva de carga representativa del grupo seis junto a sus curvas de carga diaria, consumidores tipo comercial de lunes a viernes.

La Fig. 3.25. representa al grupo seis y contiene 6 miembros, de los cuales 5 son del mismo comercio, por lo tanto, se tiene dos miembros diferentes, razón por la cual se le asigna un nuevo tipo y debido a su similitud con el tipo B, se denomina tipo B.4.

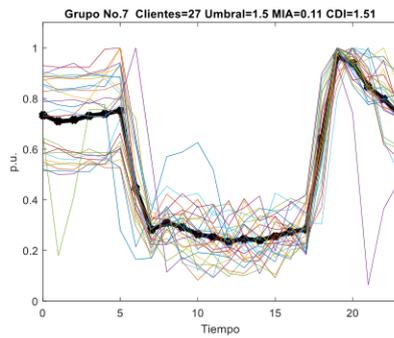


Figura 3.26. Curva de carga representativa del grupo siete junto a sus curvas de carga diaria, consumidores tipo comercial de lunes a viernes.

La Fig. 3.26. representa al grupo siete y es del tipo C.1.

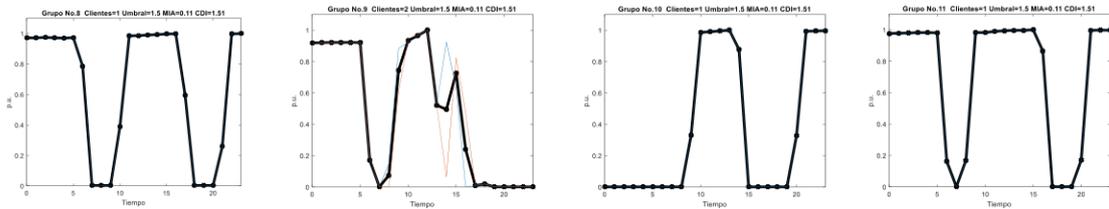


Figura 3.27. Curvas de carga representativa de los grupos ocho, nueve, diez y once junto a sus curvas de carga diaria, consumidores tipo comercial de lunes a viernes.

La Fig. 3.27. representa las curvas de un mismo consumidor asignadas a diferentes grupos, como se puede observar existen tres curvas muy parecidas y solo un grupo muy poco parecido a los demás, no se asigna un nuevo tipo de consumo a estas curvas debido a que debido a su forma pueden ser incluidas a los consumidores de tipo C, debido a que estas curvas se asignarían a dicho tipo aumentando el umbral.

La Fig. 3.28. representa al grupo doce y se le asigna el tipo B. Mientras que la curva de carga representativa en la Fig. 3.29. representa al grupo trece y se le asigna el tipo A.1.

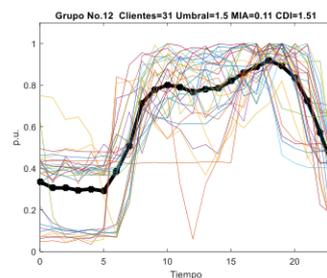


Figura 3.28. Curva de carga representativa del grupo doce junto a sus curvas de carga diaria, consumidores tipo comercial de lunes a viernes.

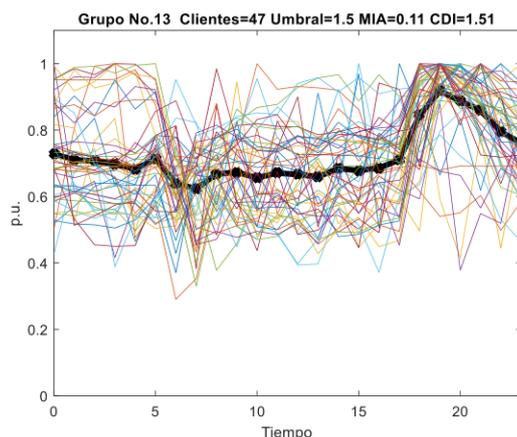


Figura 3.29. Curva de carga representativa del grupo trece junto a sus curvas de carga diaria, consumidores tipo comercial de lunes a viernes.

La Tabla 3.4. resume la asignación de tipos a cada uno de los grupos.

Tabla 3.4. Grupos obtenidos junto a su tipo.

Grupo No.	Tipo
1	C
2	A
3	B.2.
4	B.3.
5	A.3.
6	B.4.
7	C.1.
8,9,10,11	Atípico
12	B
13	A.1.

3.1.5 Consumidores comerciales para los días sábado

Para los consumidores comerciales días sábado con un umbral de 1,5; se obtiene los índices MIA de 0,12 y CDI de 1,05; para un total de 9 grupos, como se observa a continuación.

En el conjunto de grupos que se muestran en las Fig. 3.30. y Fig. 3.31. se tienen los grupos cuatro y seis con un solo miembro cada uno, razón por la cual son considerados datos atípicos, también los grupos cinco y nueve tiene dos miembros cada uno, sin embargo, en ambos grupos los miembros son diferentes comercios, razón por la cual se les asigna un tipo a cada grupo, la asignación de tipos se detalla en la Tabla 3.5.

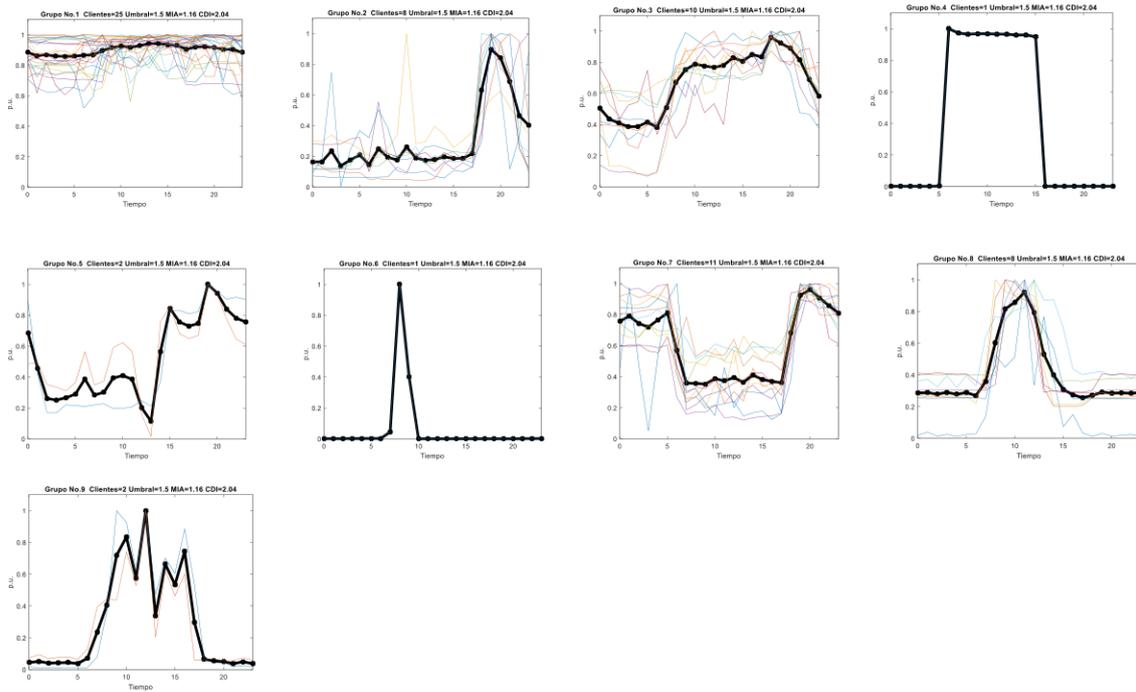


Figura 3.30. Curva de carga representativa junto a sus curvas de carga diaria, para cada grupo pertenecientes a consumidores tipo comercial días sábado.

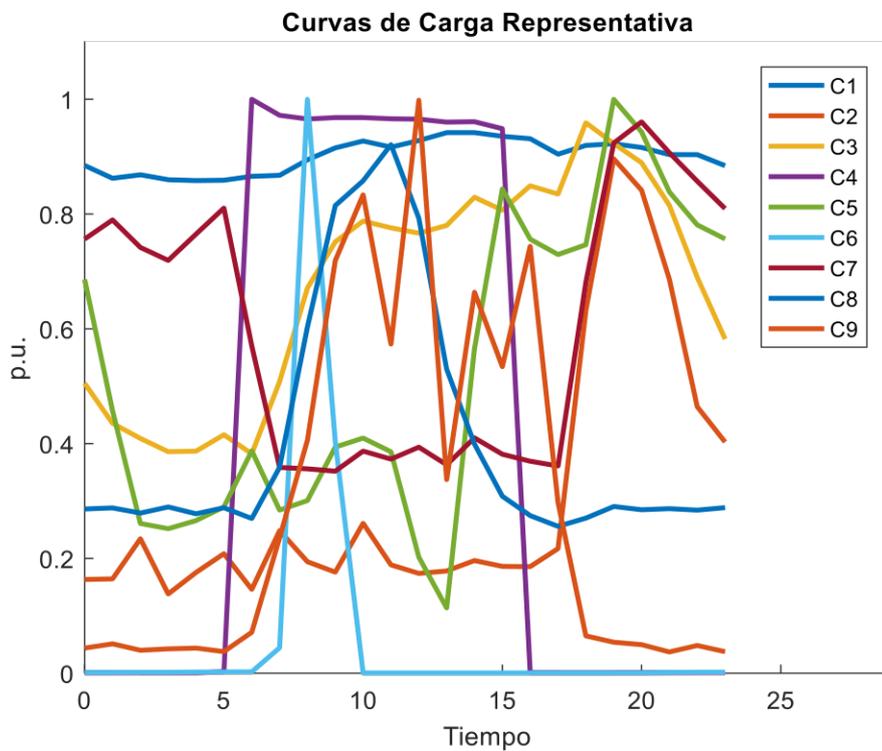


Figura 3.31. Curvas de carga representativa para cada grupo, pertenecientes a consumidores tipo comercial días sábado.

Tabla 3.5. Grupos obtenidos junto a su tipo.

Grupo No.	Tipo
1	C
2	A
3	B
4	Atípico
5	A.5.
6	Atípico
7	C.1.
8	B.1.
9	B.2.

3.1.6 Consumidores comerciales para los días domingo

Para los consumidores comerciales días domingo con un umbral de 1,5; se obtiene los índices MIA de 0,12 y CDI de 0,72; para un total de 8 grupos, como se observa a continuación.

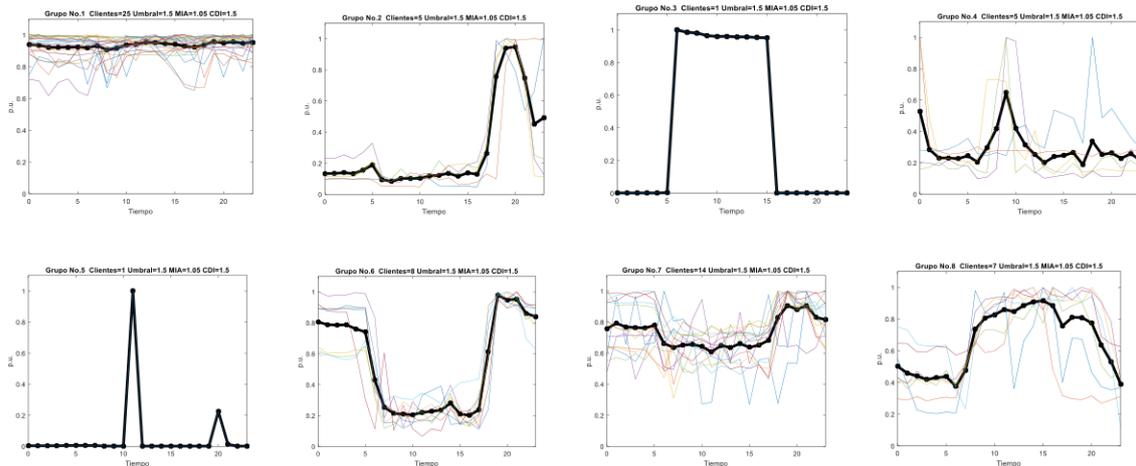


Figura 3.32. Curva de carga representativa junto a sus curvas de carga diaria, para cada grupo pertenecientes a consumidores tipo comercial días domingo.

Las Fig.3.32. y Fig. 3.33. contienen los grupos resultado de aplicar el algoritmo MFL al tipo de consumidores comerciales para los días domingo. En estas figuras se puede apreciar que los grupos tres y seis son datos atípicos ya que contiene un solo miembro cada uno. Para los otros grupos se asigna un tipo como se detalla en la Tabla 3.6.

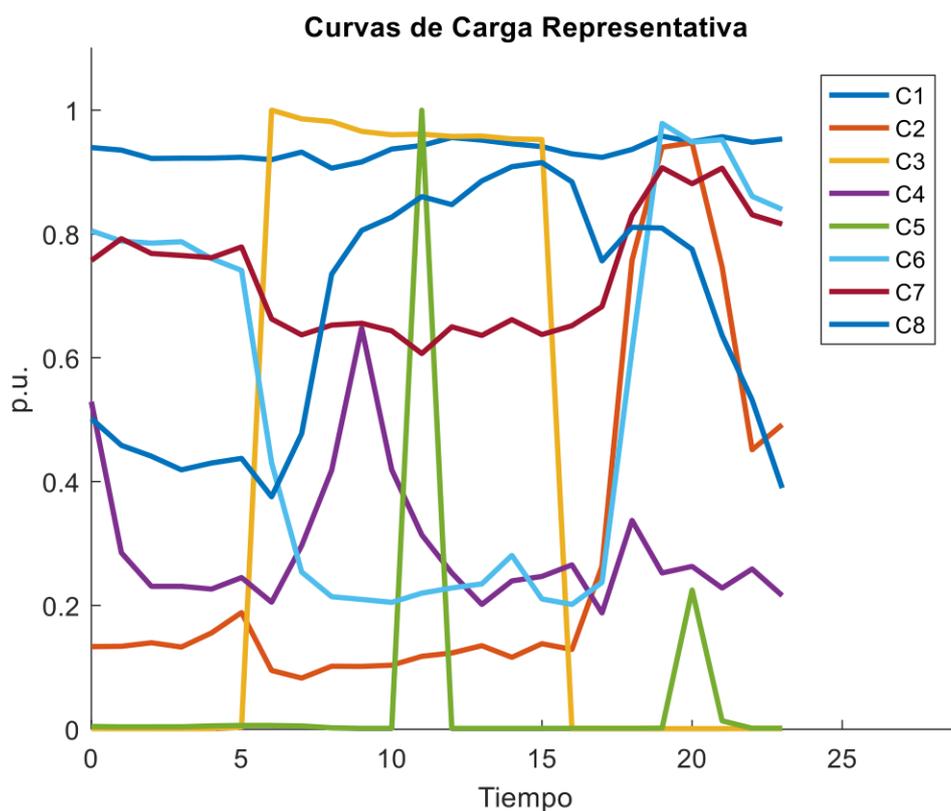


Figura 3.33. Curvas de carga representativa para cada grupo, pertenecientes a consumidores tipo comercial días domingo.

Tabla 3.6. Grupos obtenidos junto a su tipo.

Grupo No.	Tipo
1	C
2	A
3	Atípico
4	A.3.
5	Atípico
6	C.1.
7	A.2.
8	B

3.1.7 Consumidores industriales para los días de lunes a viernes

Para los consumidores industriales de lunes a viernes con un umbral de 1,5; se obtiene los índices MIA de 0,15 y CDI de 2,36; para un total de 7 grupos, como se observa a continuación.

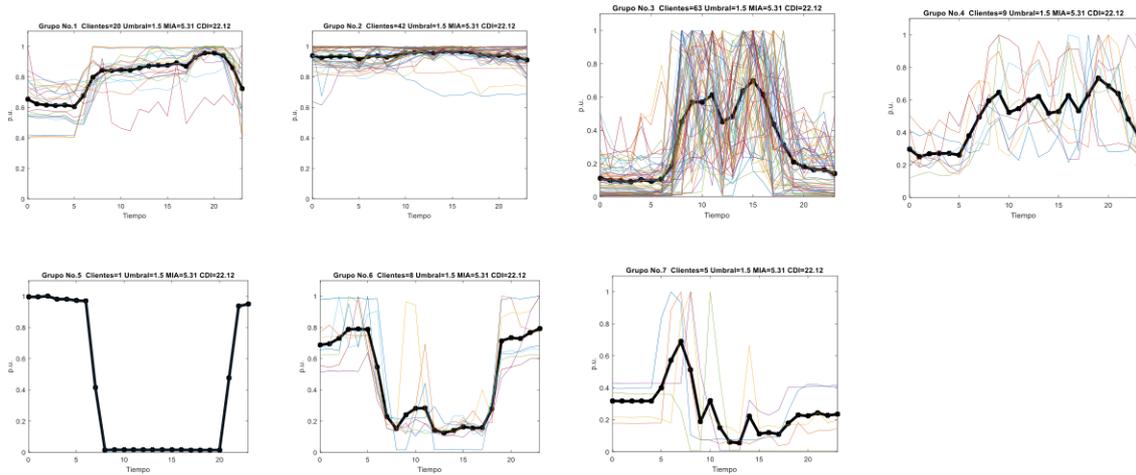


Figura 3.34. Curva de carga representativa junto a sus curvas de carga diaria, para cada grupo pertenecientes a consumidores tipo industrial de lunes a viernes.

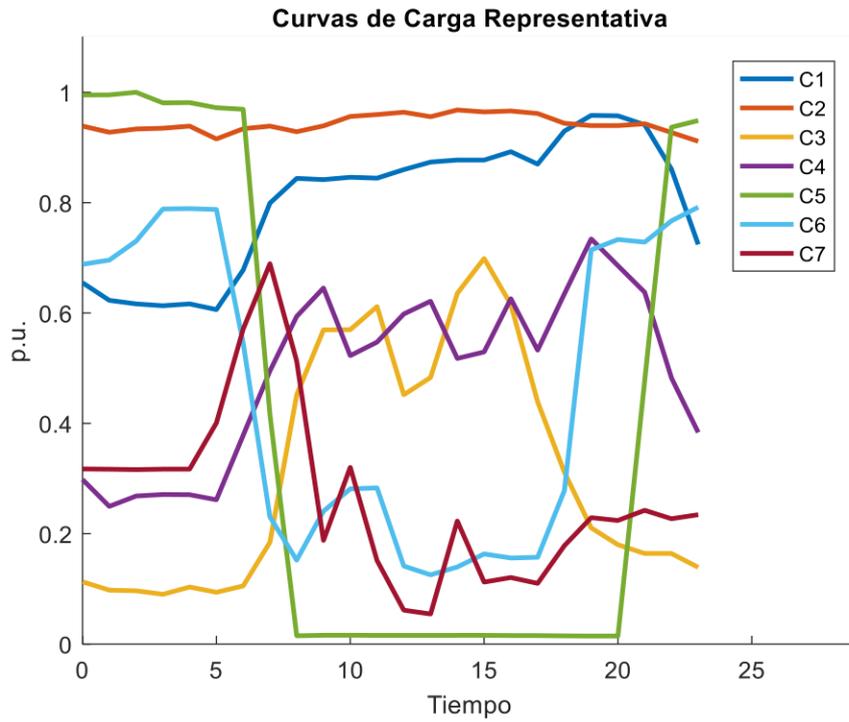


Figura 3.35. Curvas de carga representativa para cada grupo, pertenecientes a consumidores tipo industrial de lunes a viernes.

Las Fig. 3.34. y Fig. 3.35. resumen los grupos obtenidos tras ejecutar el algoritmo a la base de datos correspondiente a consumidores tipo industrial para los días de lunes a viernes. Se observa que el grupo cinco es un dato atípico debido a que solamente se compone por un miembro. Para los demás grupos se asigna un tipo de consumidor industrial, basándose

en las formas de curvas de carga representativas previamente descritas, resumiéndolas en la Tabla 3.7.

Tabla 3.7. Grupos obtenidos junto a su tipo.

Grupo No.	Tipo
1	B
2	C
3	B.3.
4	B.1.
5	Atípico
6	C.1.
7	A.3.

3.1.8 Consumidores industriales para los días sábado

Para los consumidores industriales días sábado con un umbral de 1,5; se obtiene los índices MIA de 0,13 y CDI de 0,58; para un total de 7 grupos, como se observa a continuación.

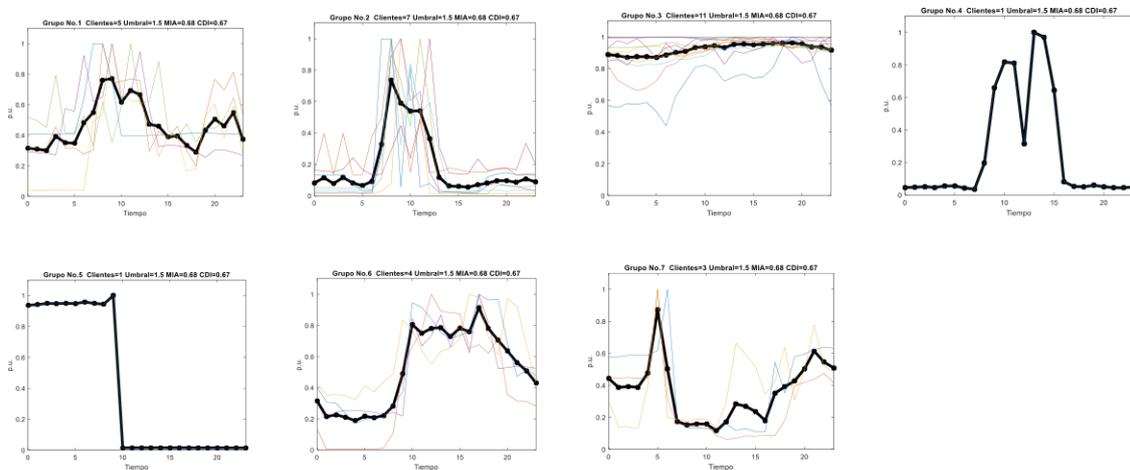


Figura 3.36. Curva de carga representativa junto a sus curvas de carga diaria, para cada grupo pertenecientes a consumidores tipo industrial días sábado.

Las Fig. 3.36. y Fig. 3.37. contienen las curvas de carga representativa en las cuales se puede observar que los grupos cuatro y cinco contienen un solo miembro, por tal motivo son considerados datos atípicos. Para todos los demás grupos se asigna un tipo específico como se detalla en la Tabla 3.8.

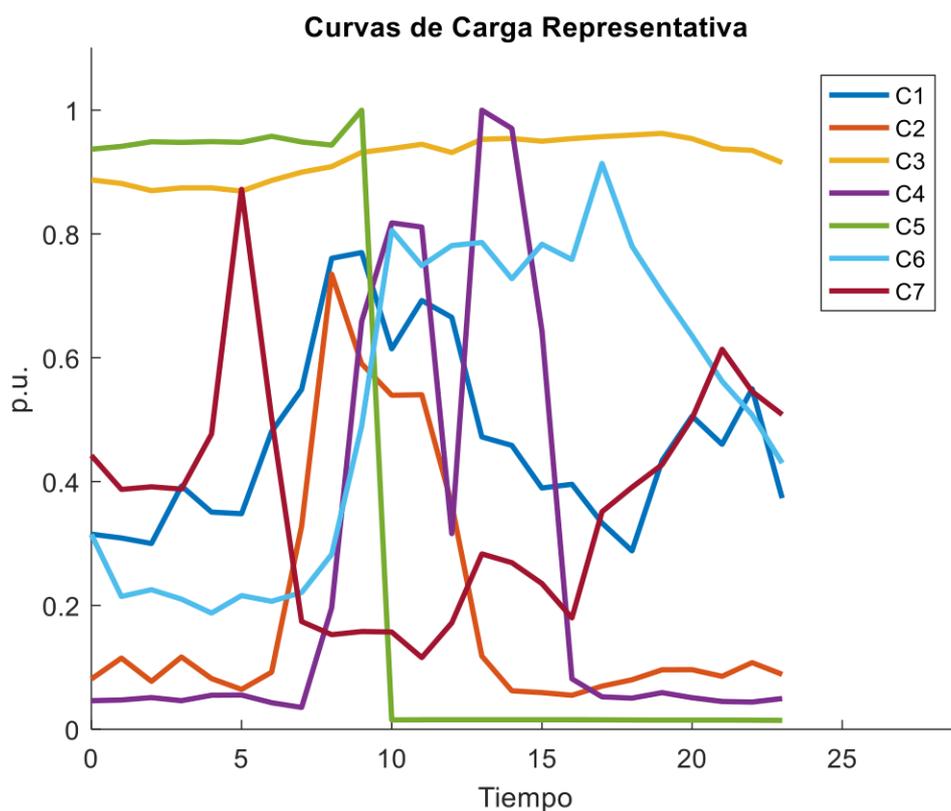


Figura 3.37. Curvas de carga representativa para cada grupo, pertenecientes a consumidores tipo industrial días sábado.

Tabla 3.8. Grupos obtenidos junto a su tipo.

Grupo No.	Tipo
1	A.2.
2	B.4.
3	C
4	Atípico
5	Atípico
6	B.1.
7	A.3.

3.1.9 Consumidores industriales para los días domingo

Para los consumidores industriales días domingo con un umbral de 1,5; se obtiene los índices MIA de 0,12 y CDI de 0,96; para un total de 7 grupos, como se observa a continuación.

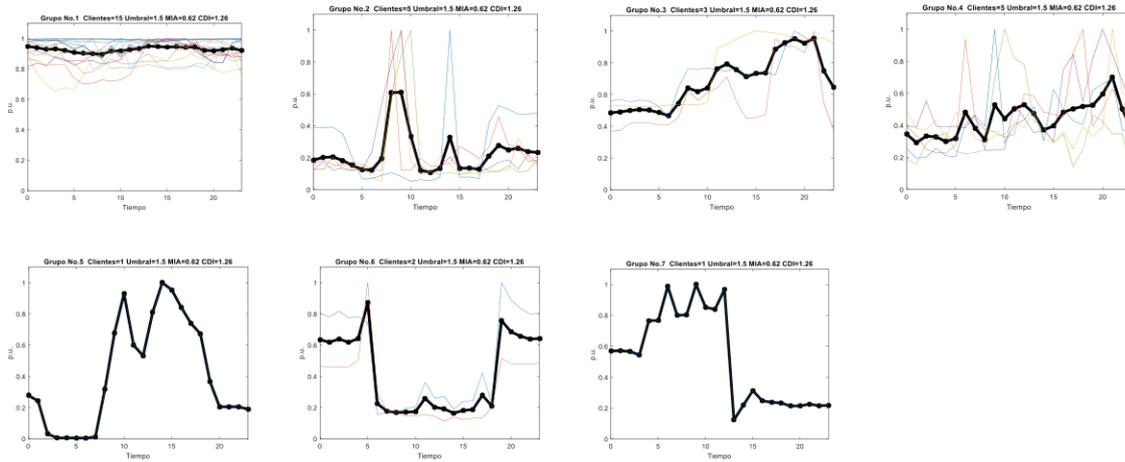


Figura 3.38. Curva de carga representativa junto a sus curvas de carga diaria, para cada grupo pertenecientes a consumidores tipo industrial días domingo.

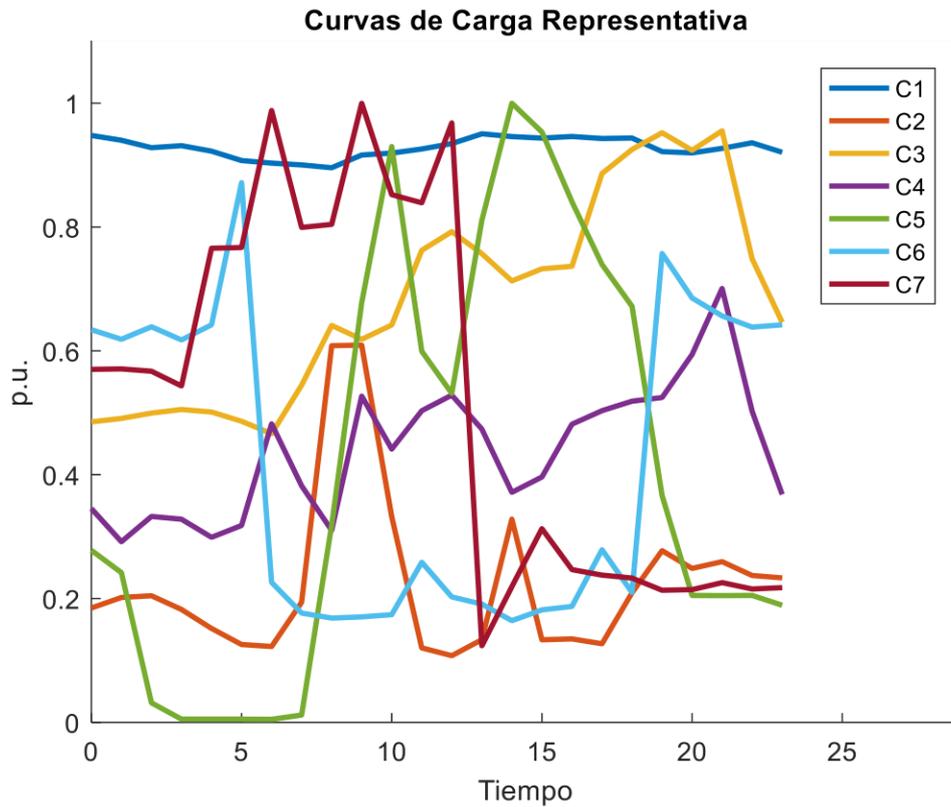


Figura 3.39. Curvas de carga representativa para cada grupo, pertenecientes a consumidores tipo industrial días domingo.

Las Fig. 3.38. y Fig. 3.39. contienen las curvas de carga representativa para el tipo de consumidor industrial días domingo. Para el resto de grupos se asigna un tipo de consumo, tal como se detalla en la Tabla 3.9.

Tabla 3.9. Grupos obtenidos junto a su tipo.

Grupo No.	Tipo
1	C
2	A.3.
3	A.1.
4	A.2.
5	Atípico
6	C.1.
7	Atípico

3.2 Conclusiones

- La técnica de agrupamiento MFL es muy útil para encontrar patrones de consumo e identificar datos atípicos. Se debe ejecutar el algoritmo tantas veces se necesite con el fin de encontrar grupos que no contengan demasiadas curvas de carga diaria diferentes a la curva de carga diaria representativa, esto se logra disminuyendo el valor del umbral.
- Existe un gran número de documentos que proporcionan información con respecto al algoritmo MFL, sin embargo, la mayoría no proporciona datos para realizar una comparación del algoritmo, razón por la cual una vez entendido el funcionamiento del algoritmo se encontraron diferentes casos que no contempla la teoría y que se aplicó en el script realizado en MatLab.
- La base de datos proporcionada sirvió para obtener los grupos de consumo específicos que se necesitaba para este trabajo, sin embargo, se debe ampliar la base de datos con el fin de obtener una descripción de completa de un grupo de consumidores durante todo un año, con lo que se entendería mejor a cada consumidor y por ende se podría realizar una buena planificación de la generación eléctrica.
- MatLab se convierte en una herramienta útil debido a sus funciones previamente establecidas que facilitan el trabajo, además al permitir crear funciones se reduce las líneas de código para tareas repetitivas, su estructura especializada para trabajar con matrices ayuda a operar varios datos a la vez, también permite obtener los datos directamente desde Excel, razón por la cual es una herramienta muy útil al trabajar con bases de datos relativamente pequeñas.
- MatLab permite trabajar con números y letras, sin embargo, al tener una matriz con los dos tipos de variables, se vuelve engorroso manipular estos datos; además cuando se desea trabajar con una gran cantidad de datos, el conjunto de instrucciones que ofrece

MatLab aún no se ha desarrollado de forma tal que sea sencilla su manipulación, razón por la cual no se recomienda MatLab R2016a para minería de datos.

- El índice de adecuación MIA, proporciona gran información con respecto al valor que debería adoptar el umbral, ya que, al obtener un índice MIA muy alto, se entiende que el valor del umbral es demasiado grande, razón por la cual el algoritmo está colocando en un mismo grupo, curvas de carga diaria con formas muy diferentes, lo que ocasionará que el índice MIA aumente su valor. En este caso se debería reducir el valor del umbral para lograr que se asignen curvas de carga diaria con la mayor similitud a un mismo grupo, sin embargo, al hacer esto también puede crearse un gran número de grupos, si esto ocurre se puede ir variando el umbral con décimas y milésimas, a fin de obtener un valor de MIA bajo, así como el número de grupos.
- El índice de adecuación CDI, no se lo puede relacionar con el valor del umbral, debido a que aun cuando exista un gran número de grupos, si los mismos tiene formas de curvas de carga representativa parecidas, el valor de CDI aumentará. Por tal motivo este índice califica directamente a la técnica de agrupamiento, razón por la cual mientras mejor sea la técnica, menor valor tendrá este índice.

3.3 Recomendaciones

- Para futuros trabajos se puede utilizar una base de datos muchos más amplia, la mayoría de trabajos que utilizan minería de datos relacionada a curvas de carga diaria, utilizan datos de por lo menos un año. Por tal motivo se recomienda que se realice directamente estos tipos de trabajos con las empresas eléctricas, con el fin de solventar todos los problemas que existan para la toma de datos.
- La mayoría de técnicas de agrupamiento complementan su trabajo con el uso de técnicas de aprendizaje no supervisado, con el fin de predecir futuros valores de los agrupamientos, por tal motivo se recomienda que se verifique si se puede predecir curvas de carga representativa aplicando cualquier técnica de aprendizaje no supervisado.
- Se recomienda recopilar todos los scripts realizados en MatLab y que son relacionados a minería de datos en una misma base de datos, con el fin de reducir los tiempos de obtención de resultados, ya que existen pasos que se repiten para todas las técnicas, tal es el caso de la importación de datos desde Excel a MatLab, así se podría ir puliendo

los algoritmos. Además, debido a que la minería de datos está en su apogeo, el recopilar todos los scripts permitiría aplicarlos con ligeros cambios a distintas ramas de la ingeniería resultando útil este y todos los trabajos venideros relacionados a minería de datos.

- Una vez que se han obtenido los distintos patrones de consumo, se recomienda realizar un estudio que relaciona la obtención de curvas de carga diaria directamente con el parque de generación eléctrica del Ecuador, con el fin de reducir costos de generación. Además una vez hecho este estudio se puede realizar el estudio de tarifas para distintos patrones de consumo.
- Actualmente existen una gran variedad de lenguajes de programación con licencia libre, por tal motivo se recomienda migrar a un lenguaje de programación gratuito con el fin de ir puliendo las librerías y poder finalmente obtener un programa propio, completo y que sea de uso nacional.

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] S. Balasubramanian and P. Balachandra, "Characterising electricity demand through load curve clustering: A case of Karnataka electricity system in India," *Computers & Chemical Engineering*, vol. 150, p. 107316, 2021, doi: <https://doi.org/10.1016/j.compchemeng.2021.107316>.
- [2] Turan Gonen, *Electric power distribution system engineering*, 3rd Edition., vol. 1. New York: CRC Press, 2014.
- [3] N. Mahmoudi-Kohan, M. P. Moghaddam, and S. M. Bidaki, "Evaluating performance of WFA K-means and Modified Follow the leader methods for clustering load curves," *2009 IEEE/PES Power Systems Conference and Exposition*, pp. 1–5, 2009.
- [4] G. Chicco, R. Napoli, P. Postolache, M. Scutariu, and C. Toader, "Customer characterization options for improving the tariff offer," *IEEE Transactions on Power Systems*, vol. 18, no. 1, pp. 381–387, 2003, doi: 10.1109/TPWRS.2002.807085.
- [5] G. Chicco, R. Napoli, F. Piglione, P. Postolache, M. Scutariu, and C. Toader, "Load pattern-based classification of electricity customers," *IEEE Transactions on Power Systems*, vol. 19, no. 2, pp. 1232–1239, 2004, doi: 10.1109/TPWRS.2004.826810.

5 ANEXOS

ANEXO I: BASE DE DATOS

Archivo Excel con nombre: BDEEA.xlsx

ANEXO II: SCRIPT PRINCIPAL DE MATLAB

```
%% Limpieza Total
clear all
close all
clc
%% IMPORTAR DATOS DESDE EXCEL
T=MFL_Importar;
%% CLASIFICAR BASE DE DATOS
[C_1,c_1]=MFL_Organizar(T);
%% RESPALDO DE DATOS
D=C_1;
d=c_1;
%% CORRECCIÓN DE DATOS
D=abs(D);
loc=zeros(1,size(D,2));
k=1;
for i=1:size(D,2)
    if size(find(D(:,i)==0),1)==0 && size(find(D(:,i)>0),1)==24
        loc(k)=i;
        k=k+1;
    end
end
loc(:,loc==0)=[];
D=D(:,loc);
d=d(:,loc);
%% NORMALIZAR DATOS
for i=1:size(D,2)
    D(:,i)=D(:,i)/max(D(:,i));
end
%% CALCULAR CONSTANTE DE PONDERAMIENTO
k=zeros(1,size(D,1));
for i=1:size(D,1)
    k(i)=var(D(i,:),1);
end
k=k/mean(k);
%% TÉCNICA DE AGRUPAMIENTO MFL
u=1.5; %1.3; Ng=35 %1.4 Ng=20
it=1;
grupos=ones(size(D,2)*(size(D,1)),size(D,2))*-1;
centros=ones(size(D))*-1;
gd=zeros(1,size(D,2));
loc=1;
while it<100
for i=1:size(D,2)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```

[DE,g]=MFL_Distancia(D(:,i),centros,k);
if it==1
    [gd(i),grupos,centros]=MFL_Asignar(u,grupos,centros,...
    D(:,i),DE,g);
else
    [gd(i),grupos,centros]=MFL_Reasignar(u,grupos,centros,...
    D(:,i),DE,g,gd(i),k);
end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
if loc==gd
    break
end
loc=gd;
it=it+1;
end
%% CALCULAR ÍNDICE DE ADECUACIÓN MIA
MIA=MFL_MIA(centros,grupos);
%% CALCULAR ÍNDICE DE ADECUACIÓN CDI
CDI=MFL_CDI(centros,grupos);
%% GRAFICAR CENTROS
MFL_Graficar_C(centros);
%% GRAFICAR GRUPOS
MFL_Graficar_G(grupos,centros,u,MIA,CDI);
%% DETALLE DE MIEMBROS DE CADA GRUPO
MIEMBROS=MFL_Miembros(centros,d,T,gd);
%% CERRAR GRÁFICAS
close all
clc

```

ANEXO III FUNCIONES CREADAS EN MATLAB E ÍNDICES DE ADECUACIÓN

FUNCIÓN MFL_Asignar

```

function
[gd,grupos,centros]=MFL_Asignar(u,grupos,centros,datos,DE,g)
loc=find(centros(1,:)==-1,1,'first');
locu=find(grupos(g,:)==-1,1,'first');
    if DE<=u
        gd=g;
        for i=1:size(datos,1)
            grupos(g,locu)=datos(i);
            g=g+size(grupos,2);
        end
        if loc==1
            centros(:,gd)=datos;
        else
            centros(:,gd)=( (locu-
1)/locu)*(centros(:,gd))+(1/locu)*(datos);
        end
    else

```

```

        gd=loc;
        g=gd;
        for i=1:size(datos,1)
            grupos(g,1)=datos(i);
            g=g+size(grupos,2);
        end
        centros(:,gd)=datos;
    end
end

```

FUNCIÓN MFL_Condensar

```

function [g]=MFL_Condensar(grupos,ng)
loc=size(grupos,1)/size(grupos,2);
locu=size(grupos,2);
g=zeros(loc,size(grupos,2));
for i=1:loc
    g(i,:)=grupos(ng,:);
    ng=ng+locu;
end
loc=find(g(1,:)==-1,1,'first');
g(:,loc:size(grupos,2))=[];
end

```

FUNCIÓN MFL_DE

```

function DE=MFL_DE(d,grupos,g)
if isempty(g)
    g=grupos(:,grupos(1,:)~= -1);
else
    g=MFL_Condensar(grupos,g);
end
DE=zeros(size(d,1),size(g,2));
for i=1:size(g,2)
    DE(:,i)=(d-g(:,i)).^2;
end

DE=((1/(size(g,2)))*(sum((((1/(size(d,1))).*sum(DE,1))).^(1/2)).^2)).^(1/2);
end

```

FUNCIÓN MFL_Distancia

```

function [DE,g]=MFL_Distancia(datos,centros,k)
loc=find(centros(1,:)~= -1);
if isempty(loc)==true
    DE=0;
    g=1;

```

```

else
    DE=zeros(size(datos,1),size(loc,2));
    for i=1:size(loc,2)
        DE(:,i)=(datos-centros(:,loc(i))).^2;
    end
    for i=1:size(k,2)
        DE(i,:)=k(i)*DE(i,:);
    end
    [DE,g]=min((sum(DE,1)).^(1/2));
    g=loc(g);
end
end

```

FUNCIÓN MFL_Eliminar

```

function [grupos]=MFL_Eliminar(grupos,datos,gd)
g=ones(size(datos,1),size(grupos,2))*-1;
gdd=gd;
for i=1:size(datos,1)
    g(i,:)=grupos(gdd,:);
    gdd=gdd+size(grupos,2);
end
loc=find(g(1,:)==-1,1,'first');
g(:,loc:size(grupos,2))=[];
for i=1:size(g,2)
    if all(g(:,i)==datos)
        ge=i;
        break;
    end
end
for i=1:size(datos,1)
    loc=grupos(gd,ge:size(grupos,2));
    loc1=size(loc,2);
    loc(:,1)=[];
    loc(1,loc1)=-1;
    grupos(gd,ge:size(grupos,2))=loc;
    gd=gd+size(grupos,2);
end
end

```

FUNCIÓN MFL_Graficar_C

```

function MFL_Graficar_C(centros)
t=size(centros,1)-1;
x=linspace(0,t,size(centros,1));
loc=centros(size(centros,1),:);
centros(2:size(centros,1),:)=centros(1:(size(centros,1)-1),:);
centros(1,:)=loc;
loc=find(centros(1,:)~-1);
figure;

```

```

l{size(loc,2),1}=[];
hold on
for i=1:size(loc,2)
    plot(x,centros(:,loc(i)));
    l{i}=strcat('C',num2str(i));
end
title('Curvas de Carga Representativas');
xlabel('Tiempo'); ylabel('p.u. ');
legend(l, 'Location', 'NorthEast');
axis([0 29 0 1.1]);
hold off
end

```

FUNCIÓN MFL_Graficar_G

```

function MFL_Graficar_G(grupos,centros,u,MIA,CDI)
MIA=round(MIA,2);
CDI=round(CDI,2);
t=size(centros,1)-1;
x=linspace(0,t,size(centros,1));
loc=centros(size(centros,1),:);
centros(2:size(centros,1),:)=centros(1:(size(centros,1)-1),:);
centros(1,:)=loc;
locu=find(centros(1,:)~= -1);
for j=1:size(locu,2)
figure;
plot(x,centros(:,locu(j)),'-k','LineWidth',4);
hold on
g=MFL_Condensar(grupos,locu(j));
loc=g(size(g,1),:);
g(2:size(g,1),:)=g(1:(size(g,1)-1),:);
g(1,:)=loc;
for i=1:size(g,2)
    plot(x',g(:,i));
end
title(strcat('Grupo No.',num2str(j),'
Clientes=',num2str(size(g,2)),...
' Umbral=',num2str(u),' MIA=',num2str(MIA),'
CDI=',num2str(CDI)));
xlabel('Tiempo'); ylabel('p.u. ');
axis([0 23 0 1.1]);
hold off
end
end

```

FUNCIÓN MFL_Importar

```

function T=MFL_Importar
loc=inputdlg({'Nombre.extension'},...
'Cargar Base de Datos',...

```

```
[1 100],...
{'BDEEA.xlsx'});
T=readtable(char(loc));
end
```

FUNCIÓN MFL_Miembros

```
function dirg=MFL_Miembros(centros,d,T,gd)
loc=find(centros(1,:)~-1);
dir=zeros(1,size(d,2));
for j=1:size(d,2)
    dir(1,j)=d(find(d(:,j)~=0,1,'first'),j);
end
n=1;
dirg(1,:)=T(1,:);
for j=1:size(loc,2)
    dirg{n,1}={num2str(j)};
    if loc(j)==2
        m=n;
        end
        n=n+1;
        dirg(n:n+(size(find(gd(1,:)==loc(j)),2)-
1),:)=T(dirg(1,:)==loc(j),:);
        n=n+size(find(gd(1,:)==loc(j)),2);
    end
    dirg(1,:)=dirg(m,:);
    dirg{1,1}={num2str(1)};
end
```

FUNCIÓN MFL_Organizar

```
function [T,d]=MFL_Organizar(T)
vt=24;

f=listdlg('Name','FECHAS',...
'PromptString','SELECCIONE COLUMNA EN LA QUE SE ENCUENTRA LA
FECHA:',...
'ListSize',[500 300],...
'ListString',T.Properties.VariableNames);

cl=listdlg('Name','CLIENTE',...
'PromptString','SELECCIONE COLUMNA EN LA QUE SE ENCUENTRA EL
CLIENTE:',...
'ListSize',[500 300],...
'ListString',T.Properties.VariableNames);

p=listdlg('Name','POTENCIA',...
'PromptString','SELECCIONE COLUMNA EN LA QUE SE ENCUENTRA LA
POTENCIA:',...
'ListSize',[500 300],...
'ListString',T.Properties.VariableNames);
```

```

t=listdlg('Name','TIEMPO',...
'PromptString','SELECCIONE COLUMNA EN LA QUE SE ENCUENTRAN LOS
INTERVALOS DE TIEMPO:',...
'ListSize',[500 300],...
'ListString',T.Properties.VariableNames);

locT=T;
num=size(locT,2)+1;
locT(:,num)=(2:1:(size(locT,1)+1))';

on='SI';
while strcmpi(on,'SI')
c=listdlg('Name','CLASIFICACION',...
'PromptString','SELECCIONE COLUMNA:',...
'ListSize',[500 300],...
'ListString',T.Properties.VariableNames);

if iscell(T{1,c})
    locm=table2cell(unique(T(:,c)));
elseif isnumeric(T{1,c})
    locm=num2str(unique(T(:,c)));
end

loc=unique(T(:,c));

sc=listdlg('Name','SUBCLASIFICACION',...
'PromptString','SELECCIONE UNA O MAS OPCIONES:',...
'ListSize',[500 300],...
'ListString',locm);

loc2=[];
if iscell(T{1,c})
    for i=1:size(sc,2)
        loc1=find(strcmpi(T(:,c),loc{sc(i),1}));
        loc2=[loc2;loc1];
    end
    T=T(loc2,:);
    locT=locT(loc2,:);
elseif isnumeric(T{1,c})
    for i=1:size(sc,2)
        loc1=find(T(:,c)==loc{sc(i),1});
        loc2=[loc2;loc1];
    end
    T=T(loc2,:);
    locT=locT(loc2,:);
end
on=questdlg('Ingresar otra clasificación?',...
'SELECCIONAR',...
'SI','NO','NO');
end

T=locT;
clear loc loc1 loc2 on
n=1;

```

```

D=ones(vt,size(T,1))*-2;
d=ones(vt,size(T,1))*-2;
loc=unique(T{: ,cl});
for tcl=1:size(loc,1)
    loc1=T(strcmpi(T{: ,cl},loc{tcl}),:);
    loc2=unique(loc1{: ,f});
    for tf=1:size(loc2,1)
        loc3=loc1(strcmpi(loc1{: ,f},loc2{tf}),:);
        for tt=1:vt-1
            if ~isempty(find(loc3{: ,t}==(tt),1))
                D(tt,n)=loc3{find(loc3{: ,t}==(tt),1),p};
                d(tt,n)=loc3{find(loc3{: ,t}==(tt),1),num};
            else
                D(tt,n)=0;
                d(tt,n)=0;
            end
        end
    end
    if vt==24
        if ~isempty(find(loc3{: ,t}==0,1))
            D(vt,n)=loc3{find(loc3{: ,t}==0,1),p};
            d(vt,n)=loc3{find(loc3{: ,t}==0,1),num};
        else
            D(vt,n)=0;
            d(vt,n)=0;
        end
    end
    end
    n=n+1;
end
end
loc=find(D(1,:)==-2,1,'first');
D(:,loc:size(D,2))=[];
d(:,loc:size(d,2))=[];
T=D;
end

```

FUNCIÓN MFL_Reasignar

```

function
[gd,grupos,centros]=MFL_Reasignar(u,grupos,centros,datos,DE,g,gd,k
)
    if DE<=u && g~=gd
        [grupos]=MFL_Eliminar(grupos,datos,gd);
        locu=find(grupos(gd,:)==-1,1,'first');
        if locu==1
            centros(:,gd)=ones(size(datos))*-1;
        else
            centros(:,gd)=(locu/(locu-1))*(centros(:,gd))-
(1/(locu-1))*(datos);
        end
        gd=g;
        loc=find(centros(1,:)==-1,1,'first');
        locu=find(grupos(gd,:)==-1,1,'first');
    end

```

```

for i=1:size(datos,1)
    grupos(g,locu)=datos(i);
    g=g+size(grupos,2);
end
if loc==1
    centros(:,gd)=datos;
else
    centros(:,gd)=( (locu-
1)/locu)*(centros(:,gd))+(1/locu)*(datos);
end
elseif DE>u && g~=gd
    [grupos]=MFL_Eliminar(grupos,datos,gd);
    locu=find(grupos(gd,:)==-1,1,'first');
    if locu==1
        centros(:,gd)=ones(size(datos))*-1;
    else
        centros(:,gd)=( (locu)/(locu-1))*(centros(:,gd))-
(1/(locu-1))*(datos);
    end
    loc=find(centros(1,:)==-1,1,'first');
    gd=loc;
    g=gd;
    for i=1:size(datos,1)
        grupos(g,1)=datos(i);
        g=g+size(grupos,2);
    end
    centros(:,gd)=datos;
elseif DE==0 && g==gd
    loc=centros;
    loc(:,gd)=ones(size(centros,1),1)*10;
    [DE,g]=MFL_Distancia(datos,loc,k);
    if DE<=u
        [grupos]=MFL_Eliminar(grupos,datos,gd);
        locu=find(grupos(gd,:)==-1,1,'first');
        if locu==1
            centros(:,gd)=ones(size(datos))*-1;
        else
            centros(:,gd)=( (locu)/(locu-1))*(centros(:,gd))-
(1/(locu-1))*(datos);
        end
        gd=g;
        locu=find(grupos(gd,:)==-1,1,'first');
        for i=1:size(datos,1)
            grupos(g,locu)=datos(i);
            g=g+size(grupos,2);
        end
        centros(:,gd)=( (locu-
1)/locu)*(centros(:,gd))+(1/locu)*(datos);
    end
end
end
end

```

FUNCIÓN MFL_CDI

```

function CDI=MFL_CDI(centros,grupos)
ng=find(centros(1,:) ~= -1);
DE=zeros(size(grupos,2),1);
loc=0;
for i=1:size(ng,2)
    g=MFL_Condensar(grupos,ng(i));
    for j=1:size(g,2)
        DE(j,1)=MFL_DE(g(:,j),grupos,ng(i));
    end
    loc=((1/(2*size(g,2)))*sum(DE.^2,1))+loc;
end
loc=((1/size(ng,2))*loc)^(1/2);
DE=zeros(size(ng,2),1);
for i=1:size(ng,2)
    DE(i,1)=MFL_DE(centros(:,ng(i)),centros,[]);
end
DE=((1/(2*size(ng,2)))*sum(DE.^2,1))^(1/2);
CDI=loc/DE;
end

```

FUNCIÓN MFL_MIA

```

function MIA=MFL_MIA(centros,grupos)
ng=find(centros(1,:) ~= -1);
MIA=zeros(size(ng,2),1);
    for i=1:size(ng,2)
        MIA(i,1)=MFL_DE(centros(:,ng(i)),grupos,ng(i));
    end
MIA=((1/(size(ng,2)))*sum(MIA.^2,1))^(1/2);
end

```