

Tecnológico Nacional de México

Instituto Tecnológico Superior de Misantla

Ingeniería en Sistemas Computacionales

Asignatura: Ciencia de datos

Semestre: 8

Nombre de la Actividad:

Proyecto Conectividad Del Tecnológico
Superior De Misantla.

DOCENTE: DR. ROBERTO ÁNGEL MELÉNDEZ
ARMENTA

Integrantes:

JUAN CARLOS TEDE GARCÍA
FERNANDO MARTÍNEZ OROZCO

Fecha de Entrega: 21/06/24

índice

Introducción	4
Marco Teórico	5
Importancia de la Conectividad en Instituciones Educativas	5
Velocidad de Internet: Conceptos Clave.....	5
Técnicas de Análisis de Datos	5
Algoritmo K-means	6
Desarrollo	7
Recolección de Datos.....	7
Preprocesamiento de Datos	7
Explicación Del Código.....	7
Resultados	10
Código Completo.....	19
Croquis de las áreas con internet en el tecnológico superior de Misantla	24
Conclusión	25

Figura 1 Importación de librerías y se carga el dataset.....	7
Figura 2 Filtro de columnas y la escala de características.....	8
Figura 3 Método Del Codo.....	8
Figura 4 selección De Cluster.....	9
Figura 5 Orden del dataframe	9
Figura 6 Grafica dl método del codo	10
Figura 7 Grafica Del Método Del Codo	10
Figura 8 Se calculan las velocidades promedio de carga y descarga para los cluster e identificar el mejor rendimiento	11
Figura 9 Mejores áreas rendimiento de internet.....	11
Figura 10 Se crea un Raking de las mejores áreas de internet	12
Figura 11 Resultados Del Rankin	12
Figura 12 Histograma de los cluster sobre la distribución de las velocidades.....	13
Figura 13 Histograma de carga y descarga.....	13
Figura 14 Creacion de la Grafica de distribución de áreas en función del cluster	14
Figura 15 Grafica De la distribución en cluster.....	14
Figura 16 Distribucion Del Cluster	15
Figura 17 Grafica De La Distribución De Las Áreas Cluster.....	15
Figura 18 Grafica de Promedios de cada cluster	16
Figura 19 Grafica De Las velocidades de cada cluster.....	16
Figura 20 Se muestra los cluster y sus características.....	17
Figura 21 Características del promedio de cada cluster.....	17
Figura 22 Creación de la tabla de contingencia y del cluster asignado.....	18
Figura 23 FIGURA 23 IDENTIFICACION DE CADA AREA CON SU CLUSTER	18
Figura 24 Tabla de contingencia de áreas y cluster.....	18
Figura 25 Croquis Del Tecnológico superior de Misantla	24

Introducción

El Tecnológico Superior de Misantla, comprometido con la mejora continua y la calidad de sus servicios, ha emprendido un proyecto de evaluación de la velocidad de internet en diferentes áreas del campus. Este estudio busca identificar las áreas con el mejor y el peor rendimiento en términos de velocidad de internet, permitiendo a la institución tomar decisiones informadas para optimizar la infraestructura de red y mejorar la experiencia de los usuarios.

En un entorno académico, la conectividad a internet es fundamental para el desarrollo de actividades académicas, administrativas y de investigación. La velocidad de internet influye directamente en la eficiencia y efectividad de los procesos educativos, desde la descarga de materiales de estudio hasta la realización de videoconferencias y el acceso a plataformas en línea. Por lo tanto, es crucial asegurar una conectividad robusta y confiable en todas las áreas del campus.

El objetivo principal de este proyecto es analizar las velocidades de carga y descarga de internet en diversas áreas del Tecnológico Superior de Misantla. Se tomaron mediciones tres veces al día, en diferentes horarios, durante un período de dos meses. Los datos recolectados se analizaron utilizando técnicas de agrupamiento (clustering) para identificar patrones y diferencias en el rendimiento del internet entre las distintas áreas. Este análisis proporcionará una visión clara de las áreas que requieren mejoras en la infraestructura de red y las que ya cuentan con un rendimiento óptimo.

Marco Teórico

Importancia de la Conectividad en Instituciones Educativas

En el contexto de las instituciones educativas, la conectividad a internet juega un papel crucial. La evolución de la educación hacia un modelo más digitalizado ha incrementado la dependencia de los servicios de internet. El acceso a recursos educativos en línea, la participación en clases virtuales, y la colaboración en proyectos de investigación son solo algunas de las actividades que dependen de una conexión a internet rápida y confiable.

Velocidad de Internet: Conceptos Clave

La velocidad de internet se mide comúnmente en megabits por segundo (Mbps) y se divide en dos categorías principales:

- **Velocidad de Descarga (Download Speed):** Es la rapidez con la que se pueden recibir datos desde internet. Es crucial para actividades como la descarga de archivos, la transmisión de videos, y la carga de páginas web.
- **Velocidad de Carga (Upload Speed):** Es la rapidez con la que se pueden enviar datos a internet. Es esencial para subir archivos, realizar videollamadas, y transmitir en vivo.

Técnicas de Análisis de Datos

El análisis de datos para evaluar el rendimiento de la red implica varias técnicas, entre las cuales el clustering es una de las más efectivas para identificar patrones y agrupar datos similares

Algoritmo K-means

El algoritmo K-means es una técnica de clustering que particiona los datos en K clusters predefinidos. Cada dato pertenece al cluster con la media más cercana, lo que minimiza la variación dentro de los clusters. Los pasos básicos del algoritmo son:

- Inicialización: Selección de K puntos iniciales (centroides).
- Asignación de Clusters: Cada punto de datos se asigna al cluster cuyo centroide está más cerca.
- Actualización de Centroides: Los centroides de los clusters se recalculan como la media de todos los puntos asignados a cada cluster.
- Repetición: Los pasos 2 y 3 se repiten hasta que las asignaciones no cambian significativamente.

Desarrollo

Recolección de Datos

Se tomaron mediciones de la velocidad de Internet en diferentes áreas del campus tres veces al día en horarios variables durante dos meses. Cada medición incluyó las velocidades de carga y descarga en Mbps, durante los meses de octubre y noviembre del año pasado.

Preprocesamiento de Datos

Se cargaron los datos desde un archivo CSV y se seleccionaron las características relevantes para el clustering: velocidad de carga, velocidad de descarga y área. Se escalaron las características numéricas para normalizar las diferencias en magnitudes.

Explicación Del Código

Importamos las bibliotecas necesarias: pandas para la manipulación de datos, KMeans de sklearn para el clustering, StandardScaler para la normalización de datos, y matplotlib y seaborn para la visualización.

Cargamos el archivo CSV que contiene los datos de velocidad de internet y lo almacenamos en un DataFrame llamado df.

Mostramos las columnas del DataFrame para entender qué datos tenemos disponibles.

Seleccionamos las características relevantes para el clustering: Carga (Mbps), Descarga (Mbps) y Área. Luego eliminamos las filas con valores nulos utilizando dropna().

```
1 import pandas as pd
2 from sklearn.cluster import KMeans
3 from sklearn.preprocessing import StandardScaler
4 import matplotlib.pyplot as plt
5 import seaborn as sns
6
7 # Cargar los datos con el delimitador correcto
8 df = pd.read_csv('C:/ood/dataset_velocidad_internet.csv', delimiter=';', encoding='latin1')
9
10 # Mostrar las columnas del DataFrame
11 print(f"Columnas del dataset: {df.columns}")
12
13 # Seleccionar las características para clustering (excluyendo 'Promedio (Mbps)')
14 features = ['Carga (Mbps)', 'Descarga (Mbps)', 'Area']
15 X = df[features].dropna()
16
```

FIGURA 1 IMPORTACIÓN DE LIBRERÍAS Y SE CARGA EL DATASET

Filtramos las columnas numéricas (Carga (Mbps) y Descarga (Mbps)) que serán utilizadas para el escalado.

Utilizamos StandardScaler para escalar estas características, lo que normaliza los datos para que tengan media 0 y desviación estándar 1. Esto es importante para que el algoritmo de clustering funcione correctamente.

```
16
17 # Filtrar solo las columnas numéricas para el escalado
18 numero_features = ['Carga (Mbps)', 'Descarga (Mbps)']
19 X_numero = X[numero_features]
20
21 # Escalar las características
22 scaler = StandardScaler()
23 X_scaled = scaler.fit_transform(X_numero)
24
25
```

FIGURA 2 FILTRO DE COLUMNAS Y LA ESCALA DE CARACTERÍSTICAS

Utilizamos el método del codo para determinar el número óptimo de clusters. Esto se hace calculando la suma de los errores al cuadrado (WCSS) para diferentes números de clusters (de 1 a 10).

```
25 # Determinar el número óptimo de clusters usando el método del codo
26 wcss = []
27 for i in range(1, 11):
28     kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=42)
29     kmeans.fit(X_scaled)
30     wcss.append(kmeans.inertia_)
31
```

FIGURA 3 MÉTODO DEL CODO

Basándonos en el método del codo, seleccionamos 4 clusters como el número óptimo.

Aplicamos el algoritmo de KMeans para agrupar los datos en 4 clusters y agregamos una nueva columna al DataFrame (Cluster) que contiene el número de cluster asignado a cada fila.

```
38
39 # Seleccionar el número óptimo de clusters (por ejemplo, 4)
40 n_clusters = 4
41 kmeans = KMeans(n_clusters=n_clusters, init='k-means++', max_iter=300, n_init=10, random_state=42)
42 df['Cluster'] = kmeans.fit_predict(X_scaled)
43
44 # Ordenar el DataFrame por la columna 'Cluster'
45 df = df.sort_values(by='Cluster')
46
```

FIGURA 4 SELECCIÓN DE CLUSTER

Ordenamos el DataFrame por la columna Cluster para facilitar la visualización y el análisis posterior.

```
44 # Ordenar el DataFrame por la columna 'Cluster'
45 df = df.sort_values(by='Cluster')
46
```

FIGURA 5 ORDEN DEL DATAFRAME

Resultados

Se grafica los resultados del método del codo para visualizar en qué punto la disminución de WCSS se vuelve menos pronunciada, indicando el número óptimo de clusters.

```
32 # Graficar el método del codo
33 plt.plot(range(1, 11), wcss)
34 plt.title('Método del Codo')
35 plt.xlabel('Número de Clusters')
36 plt.ylabel('WCSS')
37 plt.show()
```

FIGURA 6 GRAFICA DL MÉTODO DEL CODO

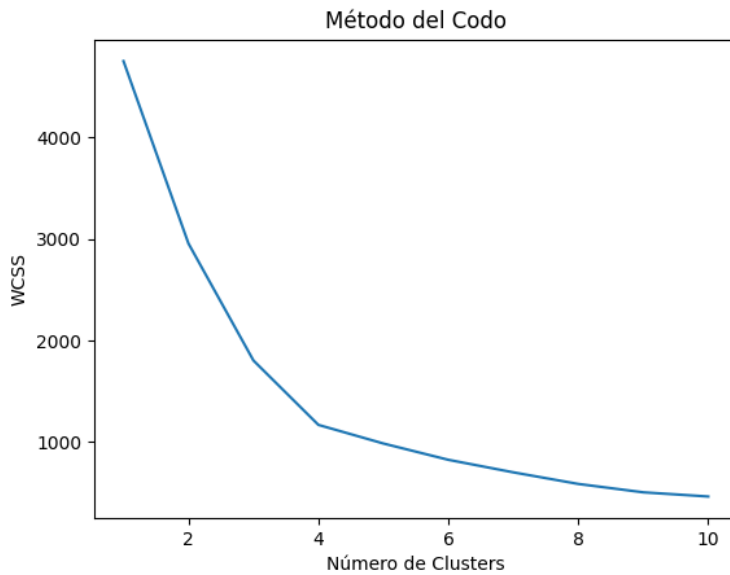


FIGURA 7 GRAFICA DEL MÉTODO DEL CODO

Se calcula las velocidades promedio de carga y descarga para cada cluster.

Se identifica el cluster con el mejor rendimiento (el que tiene las velocidades promedio más altas).

Y se muestra las áreas que pertenecen al cluster con el mejor rendimiento.

```
47 # Evaluar el cluster con mejor performance basado en las velocidades de carga y descarga
48 cluster_performance = df.groupby('Cluster')[['numero_features']].mean()
49 best_cluster = cluster_performance[['Carga (Mbps)', 'Descarga (Mbps)']].mean(axis=1).idxmax()
50
51 print("Cluster con mejor rendimiento:")
52 print(f"Cluster {best_cluster}")
53 print(cluster_performance.loc[best_cluster])
54
55 # Mostrar las áreas asociadas con el cluster de mejor rendimiento
56 areas_mejor_rendimiento = df[df['Cluster'] == best_cluster]['Area'].unique()
57 print("Áreas con mejor rendimiento de internet:")
58 print(areas_mejor_rendimiento)
```

FIGURA 8 SE CALCULAN LAS VELOCIDADES PROMEDIO DE CARGA Y DESCARGA PARA LOS CLUSTER E IDENTIFICAR EL MEJOR RENDIMIENTO

```
Áreas con mejor rendimiento de internet:
['AULA 21' 'AULA 11' 'AULA 16' 'AULA 9' 'AULA 7' 'CB' 'AULA 19' 'AULA 4'
 'Enfermeria' 'AULA 3' 'CI' 'VINCULACION' 'AULA 36' 'RECURSOS FINANCIEROS'
 'AULA 31' 'AULA 34' 'AULA 35' 'AULA 32' 'AULA 39' 'AULA 2' 'INGLES'
 'AULA 33' 'AULA 13' 'AULA 14' 'AULA 6' 'PEGA' 'DIRECCION GENERAL'
 'Sala A' 'Sala B' 'Recepcion' 'Sala E' 'AULA 23' 'NEWTON'
 'Sistematizacion' 'LE' 'CIIDE' 'AULA 30' 'Sala D' 'AULA 15' 'Sala C'
 'DIRECCION ACADEMICA' 'AULA 38' 'AULA 20' 'AULA 5' 'Sala de maestros'
 'AULA 18' 'JEFATURA' 'ASISTENTES' 'AULA 8' 'AULA 10' 'AULA 1' 'AULA 12'
 'AULA 22' 'AULA 17']
```

FIGURA 9 MEJORES ÁREAS RENDIMIENTO DE INTERNET

Se crea un ranking de las áreas basado en la velocidad de descarga promedio, ordenándolas de mayor a menor.

```
60 # Crear un ranking de las áreas basadas en la velocidad de descarga de internet
61 area_ranking = df.groupby('Área')['Descarga (Mbps)'].mean().sort_values(ascending=False)
62 print("Ranking de áreas basado en la velocidad de descarga de internet:")
63 print(area_ranking)
```

FIGURA 10 SE CREA UN RAKING DE LAS MEJORES ÁREAS DE INTERNET

```
Ranking de áreas basado en la velocidad de descarga de internet:
Área
Sala B      7.923864
AULA 1      7.687727
DIRECCION ACADEMICA  7.612273
AULA 18     7.597500
ASISTENTES  7.585227
AULA 9      7.571591
Sala de maestros  7.558409
AULA 11     7.552500
AULA 19     7.539091
NEWTON      7.538182
AULA 4      7.507500
AULA 13     7.474545
Recepcion   7.472500
INGLES      7.462955
AULA 8      7.447500
AULA 2      7.443636
AULA 30     7.440682
AULA 36     7.436364
Sistematizacion  7.413409
DIRECCION GENERAL  7.397273
AULA 14     7.382045
AULA 15     7.374318
AULA 32     7.341591
AULA 6      7.332273
Sala D      7.318636
AULA 33     7.284773
PEGA       7.275682
AULA 23     7.273182
LE          7.272500
AULA 21     7.272273
AULA 31     7.259773
AULA 7      7.246136
RECURSOS FINANCIEROS  7.240909
CB          7.239318
CI          7.237500
AULA 16     7.232500
Enfermeria  7.190227
AULA 3      7.185227
AULA 10     7.104773
AULA 39     7.101136
VINCULACION 7.077955
```

FIGURA 11 RESULTADOS DEL RANKIN

Se hace histogramas para visualizar la distribución de las velocidades de carga y descarga en cada cluster.

```
65 # Graficar los clusters
66 fig, axes = plt.subplots(1, 2, figsize=(15, 5))
67
68 for i, feature in enumerate(numero_features):
69     for cluster in range(n_clusters):
70         cluster_data = df[df['Cluster'] == cluster]
71         axes[i].hist(cluster_data[feature], alpha=0.5, label=f'Cluster {cluster}')
72         axes[i].set_title(f'Histograma de {feature}')
73         axes[i].set_xlabel(feature)
74         axes[i].set_ylabel('Frecuencia')
75         axes[i].legend()
76
77 plt.tight_layout()
78 plt.show()
```

FIGURA 12 HISTOGRAMA DE LOS CLUSTER SOBRE LA DISTRIBUCIÓN DE LAS VELOCIDADES

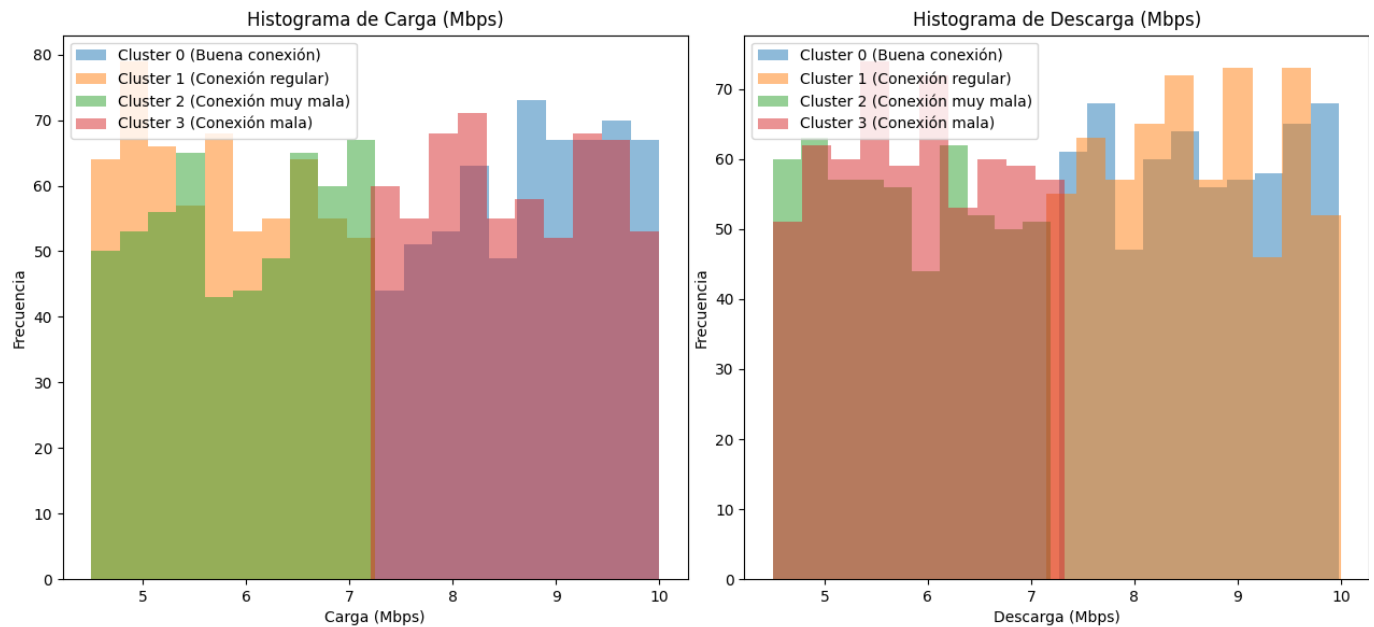


FIGURA 13 HISTOGRAMA DE CARGA Y DESCARGA

Se utiliza countplot de Seaborn para graficar la distribución de las áreas en función de los clusters. Esto nos ayuda a visualizar cuántas áreas pertenecen a cada cluster.

```
# Graficar la distribución de áreas en función de los clusters
plt.figure(figsize=(10, 6))
sns.countplot(x='Cluster', hue='Area', data=df, palette='viridis')
plt.title('Distribución de Áreas en Clusters')
plt.xlabel('Cluster')
plt.ylabel('Cantidad de Áreas')
plt.legend(title='Área', loc='upper right')
plt.grid(True)
plt.show()
```

FIGURA 14 CREACION DE LA GRAFICA DE DISTRIBUCIÓN DE ÁREAS EN FUNCIÓN DEL CLUSTER

Obtuvimos 4 clusters y se diferencian por nombre:

- Buena: Estas áreas generalmente cuentan con mejor velocidad de descarga y carga
- Regular: Estas áreas generalmente cuentan con una velocidad de descarga buena y una velocidad de carga mala.
- Mala: Estas áreas generalmente cuentan con una velocidad de carga buena y una velocidad de descarga mala
- Muy Mala: Estas áreas generalmente cuentan con mala velocidad de descarga y carga.

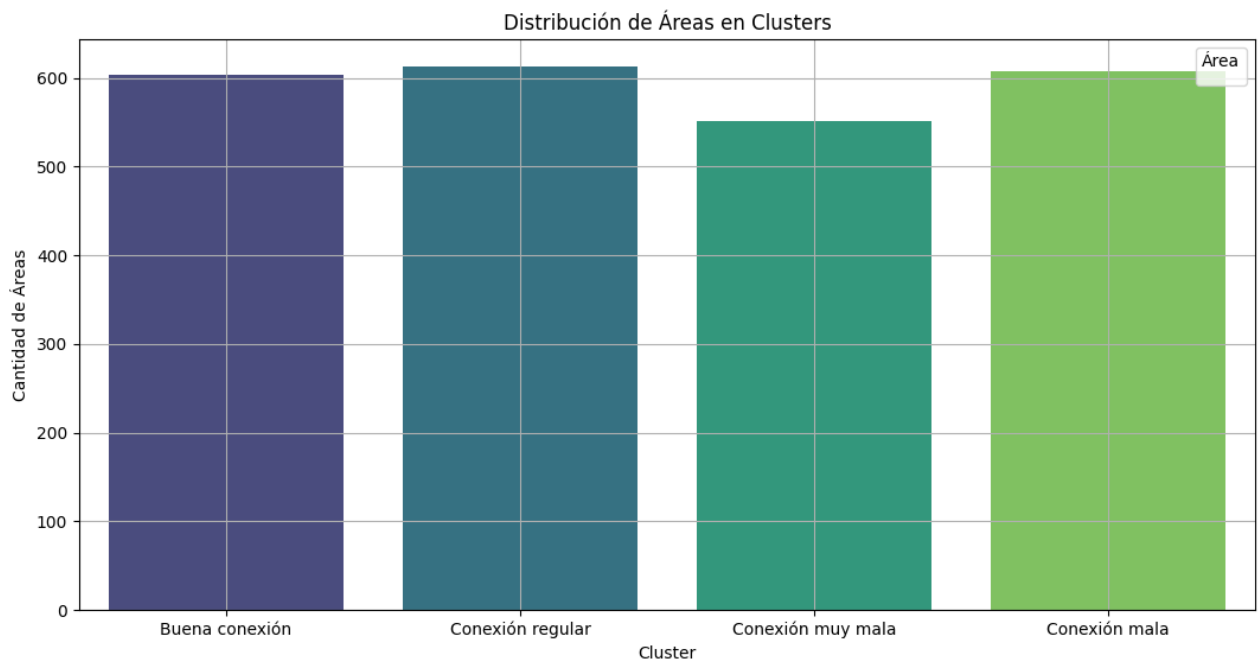


FIGURA 15 GRAFICA DE LA DISTRIBUCIÓN EN CLUSTER

Se utiliza scatterplot de Seaborn para graficar la distribución de los clusters en función de las velocidades de carga y descarga. Esto nos permite visualizar cómo se agrupan las áreas según sus características de velocidad.

```
90 # Graficar los clusters en función del area
91 plt.figure(figsize=(10,6))
92 sns.scatterplot(x='Descarga (Mbps)', y='Carga (Mbps)', hue='Cluster', data=df, palette='viridis', alpha=0.7)
93 plt.title('Distribución de Clusters en Áreas')
94 plt.xlabel('Velocidad de Descarga (Mbps)')
95 plt.ylabel('Velocidad de Carga (Mbps)')
96 plt.legend(title='Cluster')
97 plt.grid(True)
98 plt.show()
99
```

FIGURA 16 DISTRIBUCION DEL CLUSTER

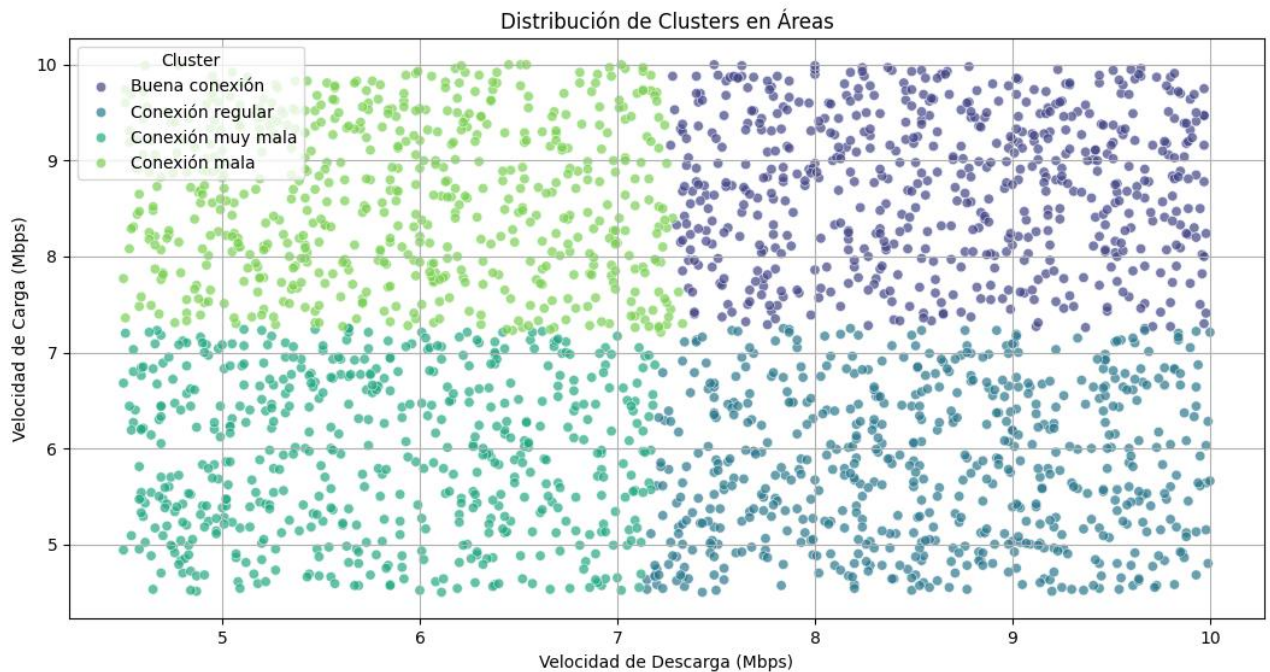


FIGURA 17 GRAFICA DE LA DISTRIBUCIÓN DE LAS ÁREAS CLUSTER

Se crea una gráfica de barras para mostrar los promedios de velocidades de carga y descarga de cada cluster. Esto proporciona una comparación clara entre los diferentes clusters.

```

100 # Graficar los promedios de cada cluster en una gráfica de barras
101 cluster_performance.plot(kind='bar', figsize=(12,8))
102 plt.title("Promedio de Velocidades por Cluster")
103 plt.xlabel("Cluster")
104 plt.ylabel("Velocidad (Mbps)")
105 plt.grid(True)
106 plt.show()
107

```

FIGURA 18 GRAFICA DE PROMEDIOS DE CADA CLUSTER

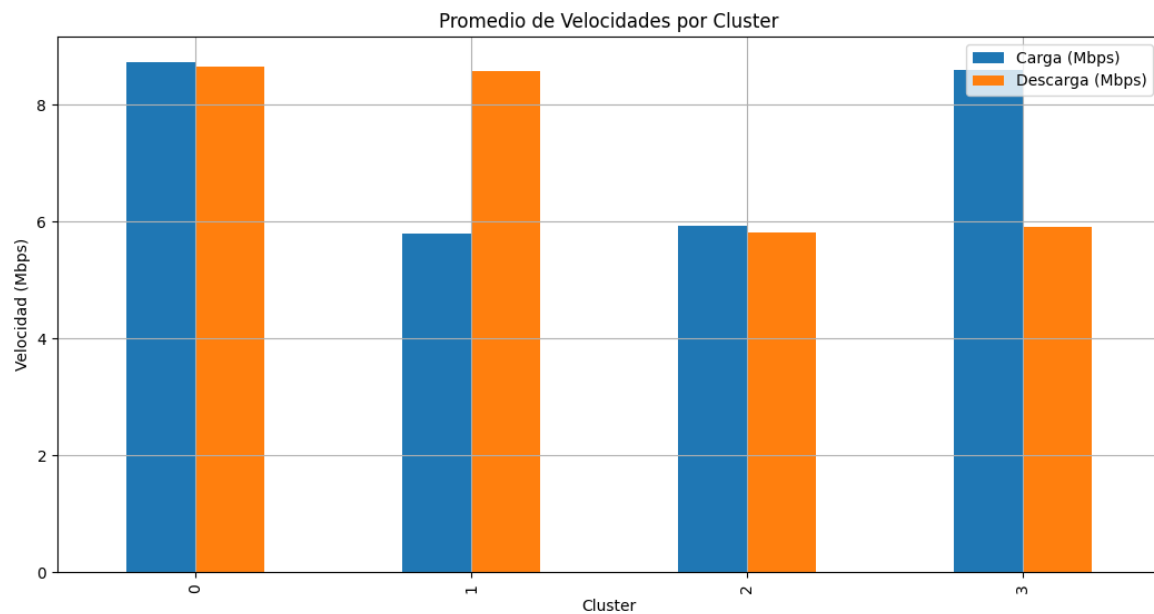


FIGURA 19 GRAFICA DE LAS VELOCIDADES DE CADA CLUSTER

Se Imprime las características promedio de cada cluster para obtener una visión general de cómo se comportan las velocidades de carga y descarga en cada grupo.

Creamos una tabla de contingencia para contar cuántas áreas hay en cada cluster.

Entonces se determina el cluster predominante para cada área basándonos en esta tabla de contingencia.

Y se ordena los resultados por cluster para una mejor presentación.

```
108 # Mostrar los clusters y sus características promedio
109 print("Características promedio de cada cluster:")
110 print(df.groupby('Cluster')[numero_features].mean())
111
112 # Crear una tabla de contingencia para contar las áreas en cada cluster
113 contingency_table = pd.crosstab(df['Area'], df['Cluster'])
114
115 # Determinar el cluster al que pertenece cada área
116 area_to_cluster = contingency_table.idxmax(axis=1)
117
118 # Ordenar el resultado por cluster
119 area_to_cluster = area_to_cluster.sort_values()
120
```

FIGURA 20 SE MUESTRA LOS CLUSTER Y SUS CARACTERÍSTICAS

```
Características promedio de cada cluster:
Carga (Mbps) Descarga (Mbps)
Cluster
0      8.726540      8.647384
1      5.801533      8.573573
2      5.919457      5.802736
3      8.594300      5.912817

Fecha Hora Router ... Promedio (Mbps) Cluster Area_Cluster
716 2023-10-19 13:00 7 ...      8.080 0      3
1881 2023-11-17 16:00 18 ...      8.845 0      0
754 2023-10-19 16:00 23 ...      8.495 0      3
1879 2023-11-17 13:00 17 ...      8.590 0      1
1877 2023-11-17 14:00 17 ...      8.490 0      0
```

FIGURA 21 CARACTERÍSTICAS DEL PROMEDIO DE CADA CLUSTER

Mas adelante se agrega una nueva columna al DataFrame original con el cluster asignado a cada área.

Se guarda el DataFrame actualizado en un archivo CSV.

Y se imprime las primeras filas del DataFrame actualizado, la tabla de contingencia y el cluster asignado a cada área para verificar los resultados.

```
122 # Agregar una nueva columna con el cluster asignado a cada área en el DataFrame original
123 df['Area_Cluster'] = df['Area'].map(area_to_cluster)
124
125 # Guardar el DataFrame actualizado en un archivo CSV
126 df.to_csv('dataset_velocidad_internet_actualizado.csv', index=False)
127
128 print(df.head())
129
130 print("Tabla de contingencia de áreas y clusters:")
131 print(contingency_table)
132
133 print("Cluster asignado a cada área basado en el conteo:")
134 print(area_to_cluster)
```

FIGURA 22 CREACIÓN DE LA TABLA DE CONTINGENCIA Y DEL CLUSTER ASIGNADO

Tabla de contingencia de áreas y clusters:

Cluster	0	1	2	3
Área				
ASISTENTES	10	17	4	13
AULA 1	14	15	7	8
AULA 10	10	9	8	17
AULA 11	15	10	7	12
AULA 12	7	13	9	15
AULA 13	13	11	11	9
AULA 14	6	20	11	7
AULA 15	10	12	6	16
AULA 16	11	10	9	14
AULA 17	5	10	14	15
AULA 18	13	12	11	8
AULA 19	15	11	5	13
AULA 2	13	11	7	13
AULA 20	8	12	10	14
AULA 21	10	12	7	15
AULA 22	7	8	19	10
AULA 23	10	14	12	8
AULA 3	13	9	12	10
AULA 30	14	14	9	7
AULA 31	11	14	10	9
AULA 32	9	14	11	10
AULA 33	15	9	7	13
AULA 34	14	5	14	11
AULA 35	6	13	12	13
AULA 36	9	18	11	6
AULA 38	10	9	9	16
AULA 39	7	11	17	9
AULA 4	12	10	10	12
AULA 5	8	7	18	11
AULA 6	16	13	8	7
AULA 7	13	9	13	9
AULA 8	12	12	9	11
AULA 9	13	15	9	7
CB	12	9	11	12
CI	12	9	11	12
CIIDE	10	2	13	19
DIRECCION ACADEMICA	17	10	9	8
DIRECCION GENERAL	12	12	9	11
Enfermeria	12	10	11	11
INGLES	14	14	11	5

FIGURA 24 TABLA DE CONTINGENCIA DE ÁREAS Y CLUSTER

Cluster asignado a cada área basado en el conteo:

Área	
AULA 30	Buena conexión
Enfermeria	Buena conexión
CI	Buena conexión
CB	Buena conexión
AULA 8	Buena conexión
AULA 7	Buena conexión
AULA 6	Buena conexión
INGLES	Buena conexión
AULA 4	Buena conexión
AULA 34	Buena conexión
AULA 33	Buena conexión
DIRECCION ACADEMICA	Buena conexión
AULA 3	Buena conexión
NEWTON	Buena conexión
DIRECCION GENERAL	Buena conexión
AULA 2	Buena conexión
AULA 19	Buena conexión
AULA 11	Buena conexión
AULA 18	Buena conexión
Sala A	Buena conexión
AULA 13	Buena conexión
Sala B	Buena conexión
RECURSOS FINANCIEROS	Conexión regular
AULA 1	Conexión regular
Sala de maestros	Conexión regular
AULA 9	Conexión regular
AULA 14	Conexión regular
Sala D	Conexión regular
Sistematización	Conexión regular
AULA 36	Conexión regular
AULA 35	Conexión regular
AULA 32	Conexión regular
AULA 31	Conexión regular
AULA 23	Conexión regular
ASISTENTES	Conexión regular
LE	Conexión muy mala
Sala E	Conexión muy mala
PEGA	Conexión muy mala
AULA 39	Conexión muy mala
AULA 5	Conexión muy mala
AULA 22	Conexión muy mala
JEFATURA	Conexión mala
AULA 38	Conexión mala

FIGURA 23 IDENTIFICACIÓN DE CADA ÁREA CON SU CLUSTER

Código Completo

```
import pandas as pd

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

import seaborn as sns

# Cargar los datos con el delimitador correcto

df = pd.read_csv('C:/ccd/dataset_velocidad_internet.csv', delimiter=',', encoding='latin1')

# Mostrar las columnas del DataFrame

print(f"Columnas del dataset: {df.columns}")

# Seleccionar las características para clustering (excluyendo 'Promedio (Mbps)')

features = ['Carga (Mbps)', 'Descarga (Mbps)', 'Area']

X = df[features].dropna()

# Filtrar solo las columnas numéricas para el escalado

numeric_features = ['Carga (Mbps)', 'Descarga (Mbps)']

X_numeric = X[numeric_features]

# Escalar las características

scaler = StandardScaler()

X_scaled = scaler.fit_transform(X_numeric)

# Determinar el número óptimo de clusters usando el método del codo

wcss = []

for i in range(1, 11):

    kmeans = KMeans(n_clusters=i, init='k-means++', max_iter=300, n_init=10, random_state=42)
```

```
kmeans.fit(X_scaled)

wcss.append(kmeans.inertia_)

# Graficar el método del codo
plt.plot(range(1, 11), wcss)
plt.title('Método del Codo')
plt.xlabel('Número de Clusters')
plt.ylabel('WCSS')
plt.show()

# Seleccionar el número óptimo de clusters (por ejemplo, 4)
n_clusters = 4

kmeans = KMeans(n_clusters=n_clusters, init='k-means++', max_iter=300, n_init=10,
random_state=42)

df['Cluster'] = kmeans.fit_predict(X_scaled)

# Asignar etiquetas descriptivas a los clusters
cluster_labels = {0: 'Buena conexión', 1: 'Conexión regular', 2: 'Conexión muy mala', 3: 'Conexión
mala'}

df['Cluster_Label'] = df['Cluster'].map(cluster_labels)

# Ordenar el DataFrame por la columna 'Cluster'
df = df.sort_values(by='Cluster')

# Evaluar el cluster con mejor performance basado en las velocidades de carga y descarga
cluster_performance = df.groupby('Cluster')[numeric_features].mean()
best_cluster = cluster_performance[['Carga (Mbps)', 'Descarga (Mbps)']].mean(axis=1).idxmax()

print("Cluster con mejor rendimiento:")
print(f"Cluster {best_cluster} ({cluster_labels[best_cluster]})")
print(cluster_performance.loc[best_cluster])
```

```
# Mostrar las áreas asociadas con el cluster de mejor rendimiento
areas_mejor_rendimiento = df[df['Cluster'] == best_cluster]['Area'].unique()
print("Áreas con mejor rendimiento de internet:")
print(areas_mejor_rendimiento)

# Crear un ranking de las áreas basadas en la velocidad de descarga de internet
area_ranking = df.groupby('Area')['Descarga (Mbps)'].mean().sort_values(ascending=False)
print("Ranking de áreas basado en la velocidad de descarga de internet:")
print(area_ranking)

# Graficar los clusters
fig, axes = plt.subplots(1, 2, figsize=(15, 5))

for i, feature in enumerate(numeric_features):
    for cluster in range(n_clusters):
        cluster_data = df[df['Cluster'] == cluster]
        axes[i].hist(cluster_data[feature], alpha=0.5, label=f'Cluster {cluster}')
    ({cluster_labels[cluster]}')
    axes[i].set_title(f'Histograma de {feature}')
    axes[i].set_xlabel(feature)
    axes[i].set_ylabel('Frecuencia')
    axes[i].legend()

plt.tight_layout()
plt.show()

# Graficar la distribución de áreas en función de los clusters
plt.figure(figsize=(10, 6))
sns.countplot(x='Cluster_Label', data=df, palette='viridis')
```

```
plt.title('Distribución de Áreas en Clusters')
```

```
plt.xlabel('Cluster')
```

```
plt.ylabel('Cantidad de Áreas')
```

```
plt.legend(title='Área', loc='upper right')
```

```
plt.grid(True)
```

```
plt.show()
```

```
# Graficar los clusters en función del área
```

```
plt.figure(figsize=(10, 6))
```

```
sns.scatterplot(x='Descarga (Mbps)', y='Carga (Mbps)', hue='Cluster_Label', data=df,  
palette='viridis', alpha=0.7)
```

```
plt.title('Distribución de Clusters en Áreas')
```

```
plt.xlabel('Velocidad de Descarga (Mbps)')
```

```
plt.ylabel('Velocidad de Carga (Mbps)')
```

```
plt.legend(title='Cluster')
```

```
plt.grid(True)
```

```
plt.show()
```

```
# Graficar los promedios de cada cluster en una gráfica de barras
```

```
cluster_performance.plot(kind='bar', figsize=(12, 8))
```

```
plt.title('Promedio de Velocidades por Cluster')
```

```
plt.xlabel('Cluster')
```

```
plt.ylabel('Velocidad (Mbps)')
```

```
plt.grid(True)
```

```
plt.show()
```

```
# Mostrar los clusters y sus características promedio
```

```
print("Características promedio de cada cluster:")
```

```
print(df.groupby('Cluster')[numeric_features].mean())
```

Crear una tabla de contingencia para contar las áreas en cada cluster

```
contingency_table = pd.crosstab(df['Area'], df['Cluster'])
```

Determinar el cluster al que pertenece cada área

```
area_to_cluster = contingency_table.idxmax(axis=1)
```

Ordenar el resultado por cluster

```
area_to_cluster = area_to_cluster.sort_values()
```

Asignar etiquetas descriptivas a cada área basado en el conteo

```
area_to_cluster_labels = area_to_cluster.map(cluster_labels)
```

Agregar una nueva columna con el cluster asignado a cada área en el DataFrame original

```
df['Area_Cluster'] = df['Area'].map(area_to_cluster_labels)
```

Guardar el DataFrame actualizado en un archivo CSV

```
df.to_csv('dataset_velocidad_internet_actualizado.csv', index=False)
```

```
print(df.head())
```

```
print("Tabla de contingencia de áreas y clusters:")
```

```
print(contingency_table)
```

```
print("Cluster asignado a cada área basado en el conteo:")
```

```
print(area_to_cluster_labels)
```


Croquis de las áreas con internet en el tecnológico superior de Misantla

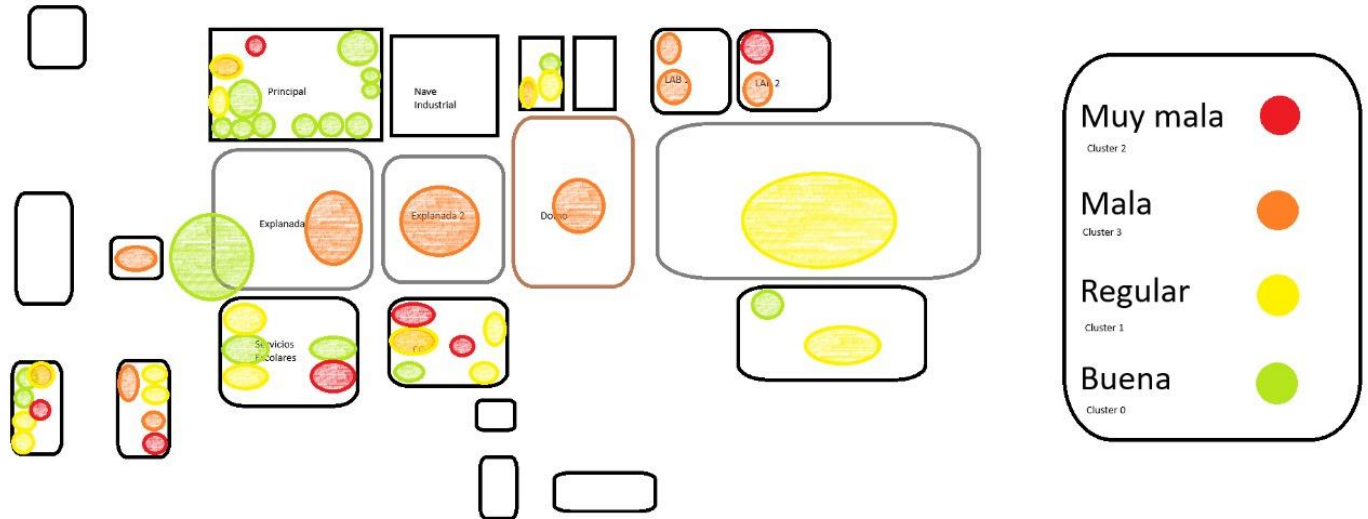


FIGURA 25 CROQUIS DEL TECNOLÓGICO SUPERIOR DE MISANTLA

Conclusión

El análisis exhaustivo de las velocidades de internet en el Tecnológico Superior de Misantla ha proporcionado información valiosa y detallada sobre el rendimiento de la red en distintas áreas del campus. A lo largo de dos meses, se llevaron a cabo mediciones rigurosas tres veces al día en diferentes horarios, lo que permitió capturar una amplia gama de variaciones en el desempeño de la red. Estos datos fueron sometidos a un análisis meticuloso mediante técnicas de clustering, específicamente utilizando el algoritmo de K-means, lo que permitió agrupar las áreas en clusters basados en sus características de velocidad de carga y descarga.

La identificación de áreas con un rendimiento superior y aquellas con un rendimiento inferior ofrece una perspectiva clara sobre los puntos fuertes y débiles de la infraestructura de red del campus. Las áreas con mejor rendimiento han demostrado ser capaces de manejar altas velocidades de carga y descarga de manera consistente, lo cual es crucial para el funcionamiento eficiente de actividades académicas y administrativas que dependen del acceso a internet. Por otro lado, las áreas con menor rendimiento presentan desafíos que deben ser abordados para garantizar una experiencia de usuario satisfactoria en toda la institución.

Los hallazgos de este estudio tienen importantes implicaciones para la planificación y mejora de la infraestructura de red del Tecnológico Superior de Misantla. Es esencial realizar intervenciones en las áreas identificadas con baja velocidad de internet para elevar su rendimiento al nivel de las áreas mejor calificadas. Esto podría incluir la actualización de equipos, la optimización de la configuración de la red y la implementación de nuevas tecnologías que mejoren la capacidad y la cobertura de la red.