Universidad San Francisco de Quito
Maestría en Inteligencia Artificial
Módulo: Inteligencia artificial
Taller 3

Edwin Montenegro, Alejandra Pinto, Gustavo Recalde, Galo Travez

13 de marzo de 2024

Índice

1. USO DE APRENDIZAJE NO SUPERVISADO

El aprendizaje no supervisado es una rama fundamental en el campo de la inteligencia artificial y el análisis de datos. Se enfoca en identificar patrones y estructuras ocultas en conjuntos de datos sin la necesidad de etiquetas o supervisión externa. Esto lo hace invaluable en problemas de identificación de patrones, donde se busca comprender la estructura subyacente de los datos y descubrir relaciones inherentes entre las variables.

En el repositorio de talleres `https://github.com/Borreguin/WorkShop1-USFQ` se ha colocado un set de datos que contiene 4 variables:

Cuadro 1: Condiciones iniciales

Tagname	**Alias**	**Descripción**
V005_vent01_CO2	CO2 Ventilation NE	Cantidad de CO2 en ppm - Salida NE
V022_vent02_CO2	CO2 Ventilation SW	Cantidad de CO2 en ppm - Salida SO
V006_vent01_temp_out	Temp. Vent. NE Out	Temperatura en °C - NE
V023_vent02_temp_out	Temp. Vent. SW Out	Temperatura en °C - SO

Cuadro 2: Variables en el set de datos.

El set de datos contiene las mediciones de concentración de CO2 y de temperatura del sistema de ventilación de un edificio inteligente. Se considerará como patrón diario para este ejercicio, aquel que pertenece al día completo, es decir, desde 00:00 hasta 23:59 del día.

1.1. Actividades:

Plotear las variables: Presentar un gráfico por cada variable que muestre sus valores superpuestos por cada día. Para propósito de explicación, la gráfica anterior muestra un ejemplo de una variable de presencia solar que no corresponde a este set de datos, pero sirve de ilustración.

Encontrar patrones – análisis univariable: Utilizando cualquier técnica de aprendizaje no supervisado, encontrar los patrones diarios que existen en el data set, para cada variable individual. Utilizar al menos dos técnicas para verificar su consistencia entre las dos técnicas.

Encontrar anomalías – análisis univariable: Es posible que ciertos perfiles diarios en el set de datos no pertenezcan a los patrones diarios descubiertos en el literal B. ¿Cómo detectarlos?

Encontrar patrones – análisis multivariable: De manera similar al literal B, encontrar los patrones diarios que existen en el data set, para cada par de variables, es decir, las dos de la parte Norte Este y las dos de la parte Sur Oeste. Utilizar al menos dos técnicas para verificar su consistencia entre las dos técnicas.

Encontrar anomalías – análisis multivariable: De manera similar al literal C, encontrar anomalías, pero de los dos pares de variables.

1.2. Desarrollo

Carga y Preparación de Datos Se carga un conjunto de datos desde una URL específica, convirtiendo la columna `'timestamp'` a formato de fecha y hora, y extrayendo el día y el nombre del día de la semana de esta columna.

Heatmaps Se define una función `mostrar_heatmap` que, dependiendo del día de la semana seleccionado, filtra los datos y muestra heatmaps de la variable elegida.

Boxplots Se crea una función para generar gráficos de caja que muestran la distribución diaria de las variables seleccionadas, facilitando la identificación de patrones y outliers.

Selección Interactiva Se implementa un menú interactivo que permite al usuario seleccionar un día de la semana y la variable de interés para la visualización.

```
Selecciona el día de la semana o 'Todos' los días:
1: Lunes
 2: Martes
 3: Miércoles
 4: Jueves
 5: Viernes
 6: Sábado
 7: Domingo
 8: Todos
Introduce el número de tu opción: 1
Introduce el número correspondiente a la variable que deseas visualizar:
1: V005_vent01_CO2
2: V022_vent02_CO2
3: V006_vent01_temp_out
4: V023_vent02_temp_out
5: V005_vent01_CO2', 'V022_vent02_CO2
6: V006_vent01_temp_out', 'V023_vent02_temp_out
```

Figura 1: Sección del día de la semana según variable solicitada

Limpieza y Normalización de Datos Se realiza la limpieza de datos reemplazando los valores faltantes con la media de cada columna. Posteriormente, se normalizan los datos para prepararlos para análisis de clustering utilizando `StandardScaler`.

Clustering con K-Means Se aplica el algoritmo `K-Means` para identificar grupos dentro de los datos, utilizando el método del codo para determinar el número óptimo de clusters.

Análisis de Componentes Principales (PCA) Aunque se importa PCA de `scikit-learn`, el código no muestra su aplicación directa en el fragmento proporcionado.

1.3. Resultados

Heatmaps: Se presentan resultados para una sola variable, así como para todas las variables, atraves de heatmaps que muestran en colores el comportamiento durante las 24 horas del día, durantes 31 días del mes.

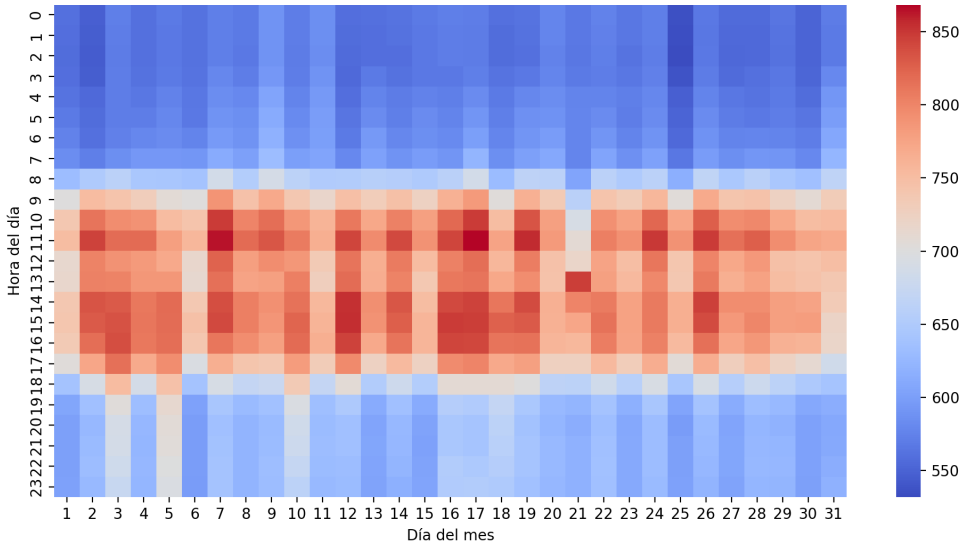


Figura 2: Heatmap de la variable CO2 en NE, durante 31 días, 24 horas

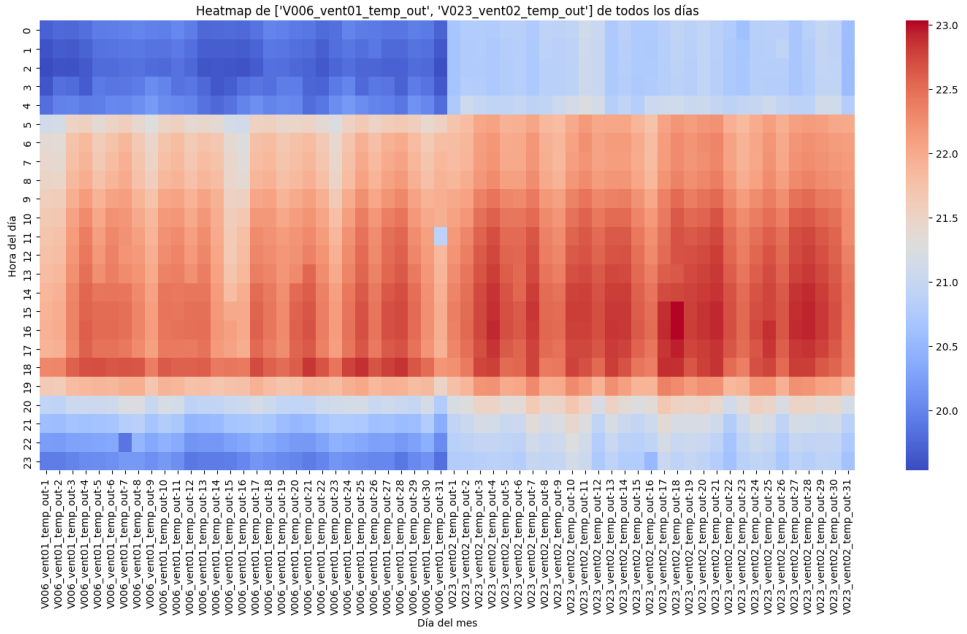


Figura 3: Heatmap de las variables Temperatura en NE y SO, en °C.

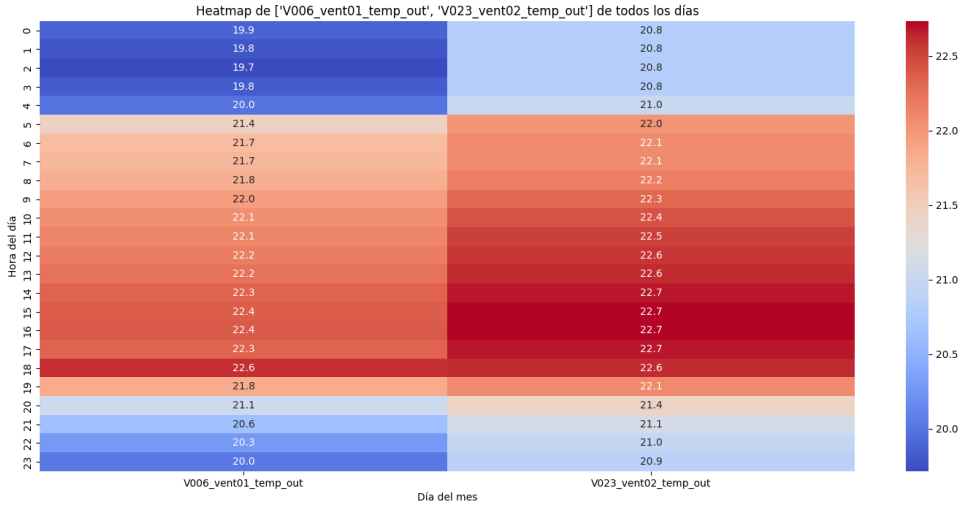


Figura 4: Heatmap de la variable Temperatura promedio en NE y SO las 24 todos los días

Boxplots: Descripción de los boxplots dado por la distribución diaria por cada variable durante las 24 horas del día.

- En la parte superior izquierda, tenemos un gráfico de Distribución diaria de V005_vent01_CO2, que muestra una variabilidad considerable en las mediciones con algunos valores atípicos representados por círculos fuera del rango intercuartílico.
- En la parte superior derecha, el Distribución diaria de V022_vent02_CO2 presenta una distribución similar de CO2, aunque con rangos intercuartílicos ligeramente diferentes y también con valores atípicos.
- En la parte inferior izquierda, el gráfico Distribución diaria de V006_vent01_temp_out"muestra las mediciones de temperatura, con una distribución más uniforme a lo largo del día y menos valores atípicos en comparación con las mediciones de CO2.

- En la parte inferior derecha, el Distribución diaria de V023_vent02_temp_out"muestra mediciones de temperatura para otro conjunto de datos, también con menos valores atípicos y una distribución relativamente uniforme a lo largo del día.

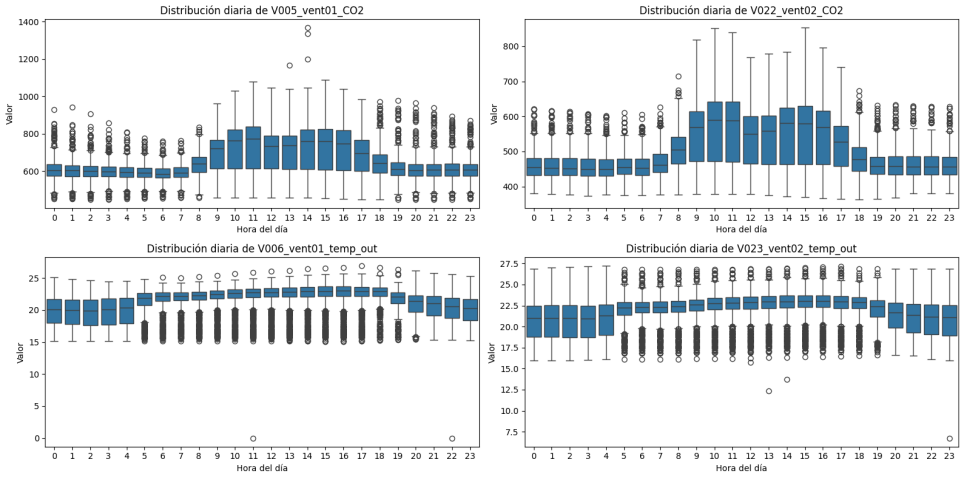


Figura 5: Distribución diaria de la variable Y.

Time Series: Las imágenes proporcionadas son gráficos de líneas que representan valores diarios superpuestos para dos variables de medición de CO2 y dos de temperatura en un sistema de ventilación, probablemente a lo largo de varios días y a diferentes horas del día.

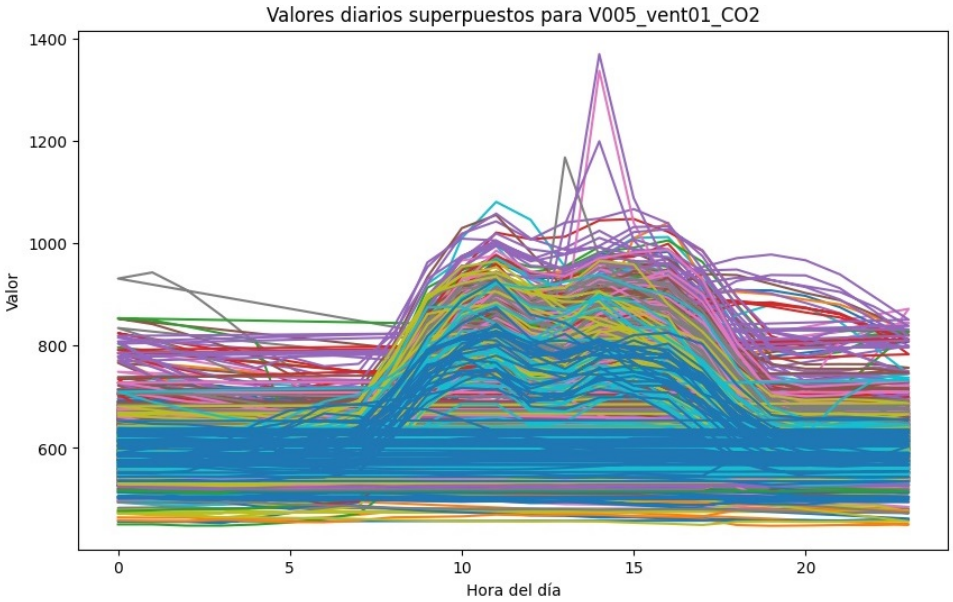


Figura 6: Valores diarios superpuestos, univariable

La gráfica de V005_vent01_CO2 muestra la variabilidad en las concentraciones de CO2 medidas por el sensor V005 a lo largo del día. Los picos y valles indican fluctuaciones en los niveles de CO2, lo que podría sugerir patrones de uso del sistema de ventilación o cambios ambientales.

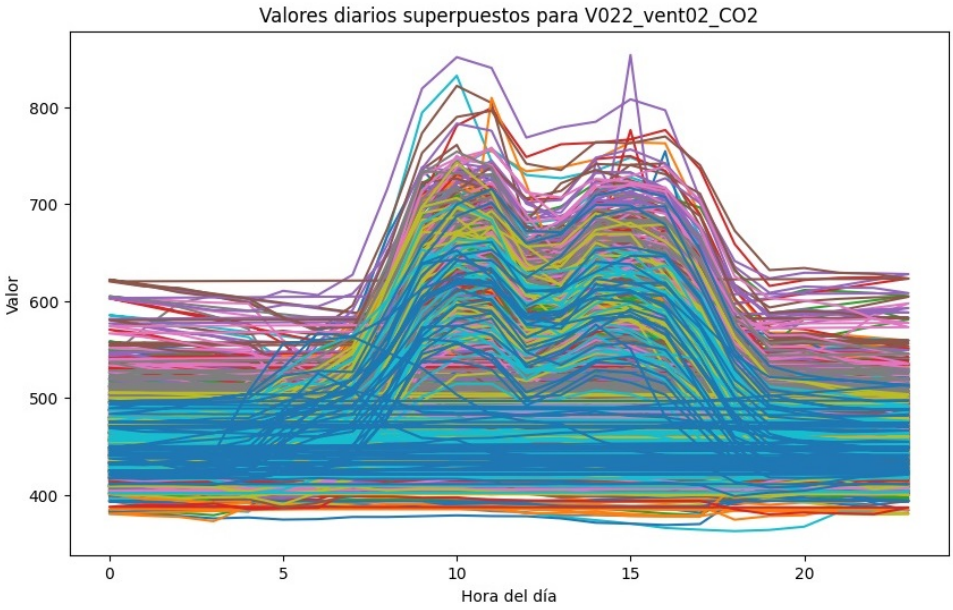
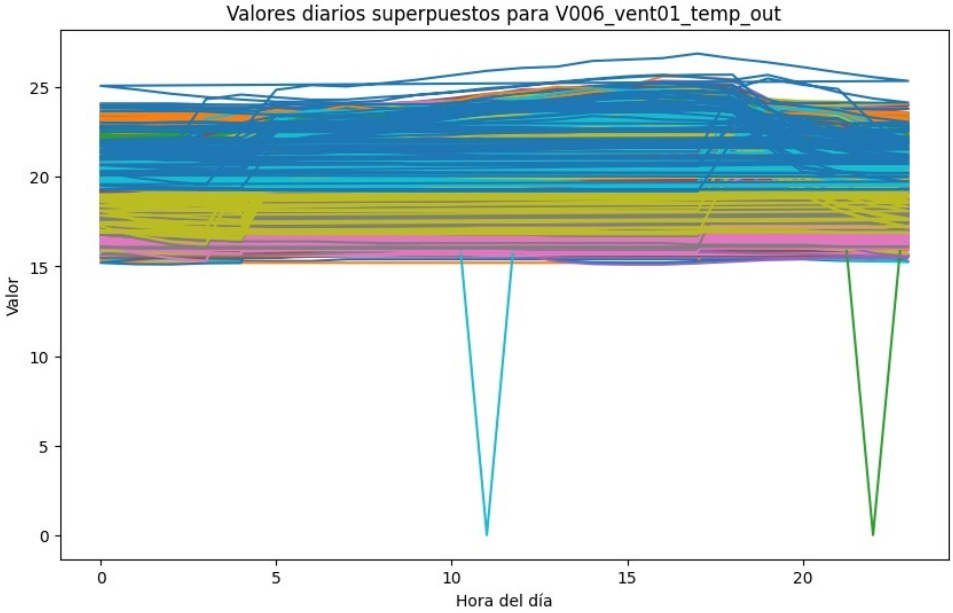


Figura 7: Valores diarios superpuestos, univariable

La gráfica de V022_vent02_CO2 muestra las concentraciones de CO2 medidas por otro sensor, V022. Los patrones de datos pueden ser útiles para comparar el rendimiento de diferentes áreas o sistemas de ventilación.



La gráfica de V006_vent01_temp_out muestra las temperaturas medidas por el sensor V006. A diferencia de los gráficos de CO2, las líneas parecen más agrupadas y consistentes, lo que indica menos variabilidad en las mediciones de temperatura a lo largo del día.

La gráfica de V023_vent02_temp_out representa las mediciones del sensor V023, deonde los patrones de temperatura muestran consistencia con variaciones menos pronunciadas que las mediciones de CO2.

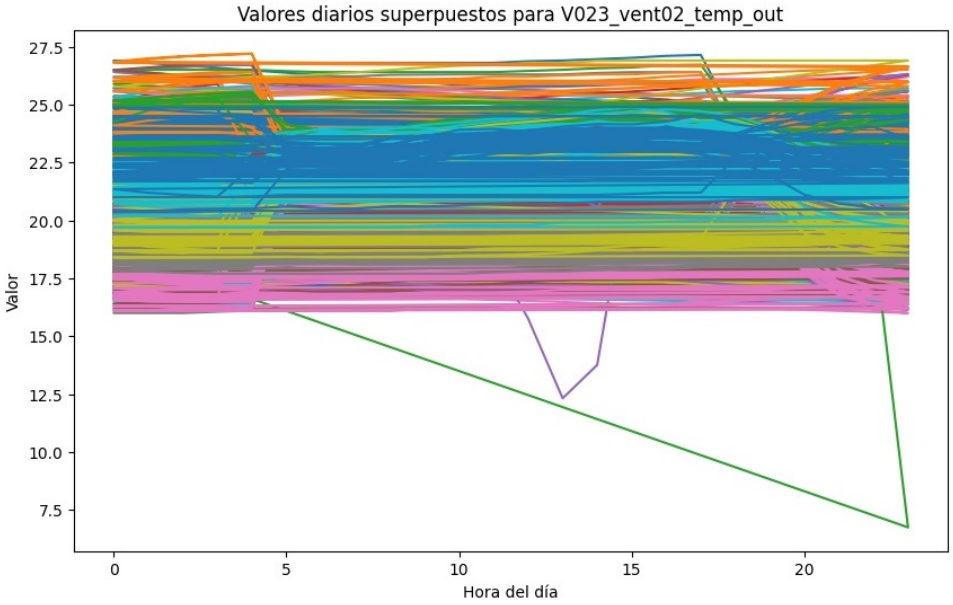


Figura 8: Valores diarios superpuestos, univariable

Clustering: Descripción de los resultados del clustering. Se uso el método del codo para encontrar el número de clusters adecuado.

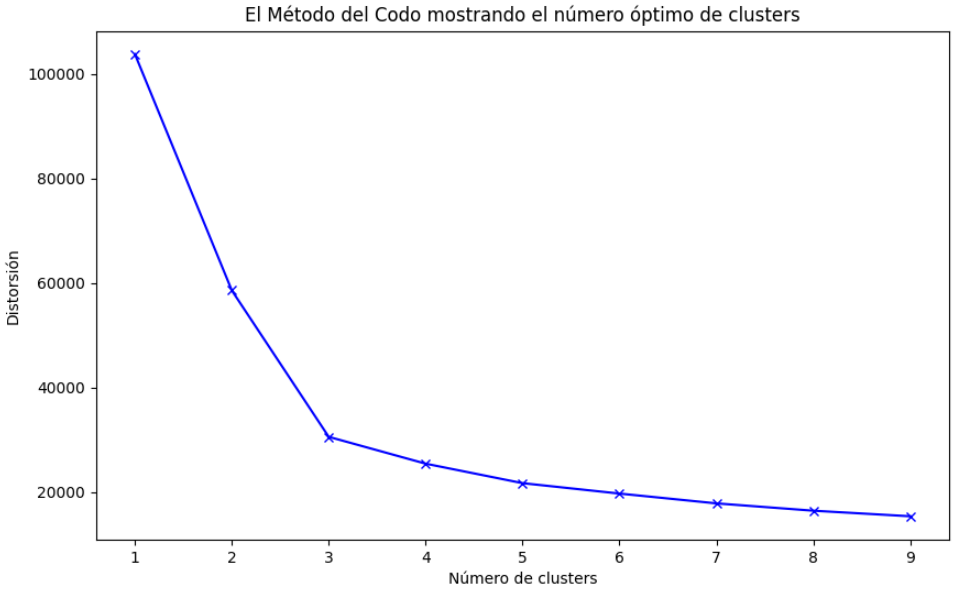


Figura 9: Clusters diarios.

En la imagen se puede observar los clusters diarios para la variable temperatura, donde se observa pequeñas variaciones entre las 5 am y las 19 pm. Además, la mayor marte de las horas se mantiene en una temperatura entre los 22°C a los 27°C.

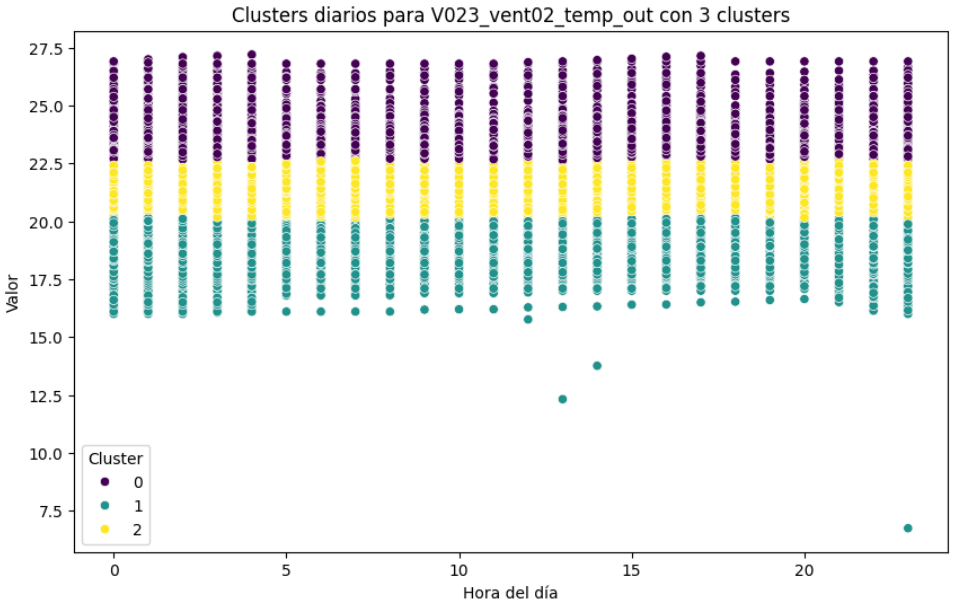


Figura 10: Clusters diarios para la variable W.

La imagen 10 muestra un diagrama de dispersión, usado para visualizar la relación entre dos variables cuantitativas. En este caso, las variables son CO2 Ventilation NE (ppm) en el eje x y Temp. Vent. NE Out (°C) en el eje y.

El gráfico también presenta una clasificación en clusters, que son agrupaciones de puntos basadas en la similitud de sus características, lo que sugiere que se ha realizado un análisis de clustering, probablemente utilizando el algoritmo K-Means. Hay tres clusters identificados por colores diferentes: azul (Cluster 0), verde (Cluster 1) y naranja (Cluster 2). Además, se marcan los centroides de cada cluster con una ẍr̈oja, que representan el centro medio de los puntos en cada cluster.

El propósito de este gráfico es identificar patrones dentro del conjunto de datos y entender cómo la concentración de CO2 se relaciona con la temperatura de salida en un sistema de ventilación NE. Los clusters pueden indicar condiciones operativas distintas o tipos de comportamiento dentro del sistema monitoreado.

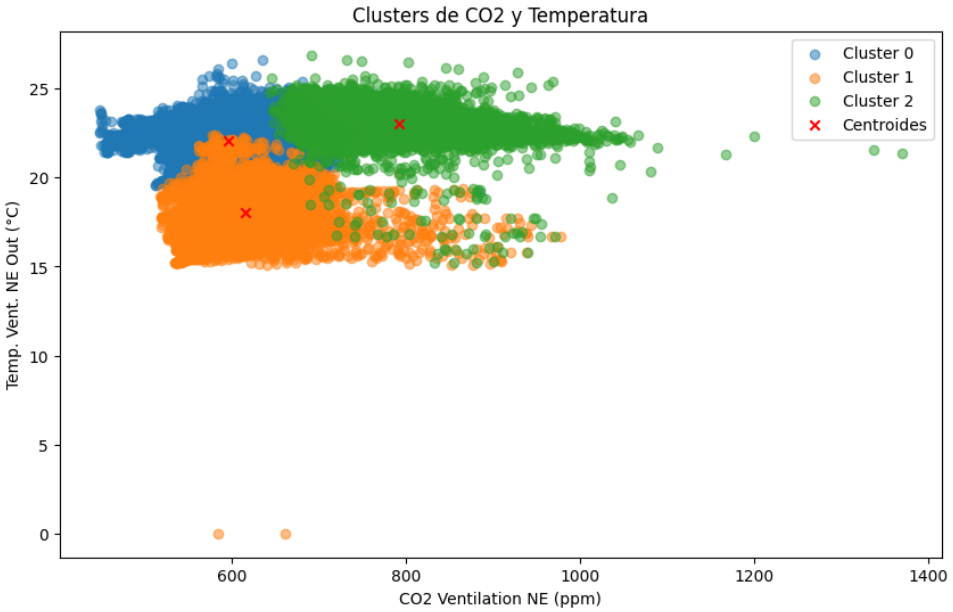


Figura 11: Clusters diarios para la variable W.

1.4. Conclusiones Sección 1

- En los gráficos boxplot presentados, se observan las distribuciones de los niveles de CO2 y de temperatura a lo largo de los días de la semana para dos sensores distintos, cada uno midiendo CO2 y temperatura respectivamente. Aquí te detallo los patrones que se pueden visualizar:

- Para los niveles de CO2 (V005_vent01_CO2 y V022_vent02_CO2):

  La mediana de los niveles de CO2 tiende a ser más alta durante los días laborales que durante el fin de semana, lo que refleja una mayor actividad en días laborales.

  Hay una cantidad considerable de valores atípicos para los días laborales, particularmente en el sensor V005_vent01_CO2, con picos o eventos de alta emisión de CO2 que no siguen el patrón general.

  El sensor V022_vent02_CO2 muestra menor variabilidad y menos valores atípicos en comparación con V005_vent01_CO2, lo que podría sugerir que la fuente de CO2 medida por V022 es más constante o que el ambiente es más controlado.

- Para las temperaturas (V006_vent01_temp_out y V023_vent02_temp_out):

  La temperatura muestra una variabilidad más constante a lo largo de la semana con medias y medianas que no varían drásticamente entre días laborales y fines de semana.

  No se observan patrones claros que diferencien los días laborales de los fines de semana en términos de temperatura, lo que sugiere que la temperatura no está tan influenciada por la actividad en ele edificio como los niveles de CO2.

  Los valores atípicos en la temperatura son menos frecuentes que en los niveles de CO2, y donde aparecen, no muestran un patrón consistente a lo largo de los días.

  La consistencia en la temperatura podría deberse a la regulación climática o a que la temperatura exterior tiene menos variación a lo largo de la semana.

2. INVESTIGACIÓN OPERATIVA: TRAVELLING SALEMAN PROBLEM (TSP)

Dentro de la categoría de problemas NP-duro en teoría de la complejidad computacional, el problema del viajante de comercio (TSP) es conocido por ser uno de los problemas clásicos para explicar la complejidad que puede llegar a tener un problema dependiendo del número de instancias a resolver. El término NP-duro significa que no existe un algoritmo conocido que pueda resolver todas las instancias del problema en tiempo polinomial. Sin embargo, existen varios enfoques que pueden llegar a obtener soluciones subóptimas en un tiempo razonable.

El algoritmo de vecinos cercanos es una buena opción para obtener una respuesta en tiempo razonable. Por ejemplo, para el problema planteado en el taller 1, el valor mínimo encontrado por el Grupo 3 fue aproximadamente 1850 km. ¿Será que usando esta heurística y Linear Programming (LP) se puede alcanzar un recorrido menor?

- Para la ejecución de este problema, instalar el paquete GLPK de acuerdo con lo indicado en el readme.md.

- Se coloca la imagen de la solución del vecino más cercano como referencia:

2.1. Analizar el código propuesto

En la carpeta Taller3/P2_TSP se coloca el código del modelado del TSP usando LP, correr el caso 1, con una tolerancia de 0.20 y tiempo límite de 120 segundos, marcar los tiempos que se demora para 10, 20, 30, 40 y 50 ciudades. Subjetivamente, ¿qué tal te parece las soluciones que ha arrojado el modelo sin aplicar todavía una heurística que ayude al modelo?

Resumen del código: En la primera parte del código se generan ciudades aleatorias calculando los caminos y sus respectivas distancias euclidianas entre ellas para simular el problema del TSP. La clase TSP encapsula la lógica necesaria para modelarlo y resolverlo. Se inicializa con ciudades, distancias entre ellas, y un conjunto de heurísticas opcionales para mejorar la búsqueda de soluciones.

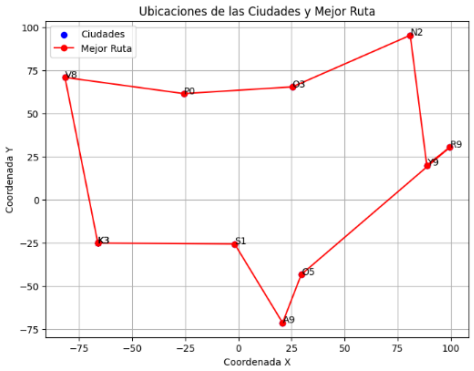
cal_min_max_distances:: Calcula estimaciones de distancias mínimas y máximas posibles basadas en la media de distancias y el número de ciudades. encontrar_la_ruta_mas_corta: Utiliza Pyomo para definir y resolver el modelo matemático del TSP. Incluye la formulación de la función objetivo (minimizar la distancia total del viaje), restricciones para asegurar que cada ciudad sea visitada exactamente una vez, y restricciones opcionales basadas en las heurísticas seleccionadas. Resuelve el modelo usando GLPK y procesa los resultados para extraer la ruta óptima y su distancia total.

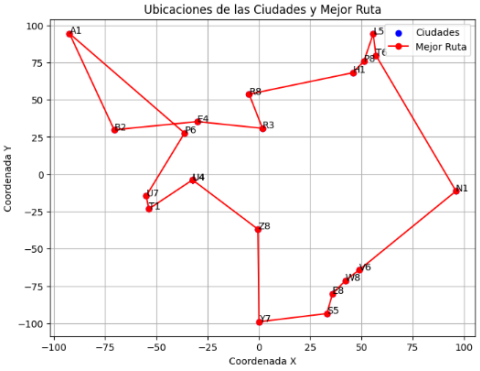
limitar_funcion_objetivo: Establece límites superiores e inferiores en la función objetivo para restringir el espacio de búsqueda a soluciones más realistas.

vecino_cercano: Añade restricciones para favorecer la selección de ciudades cercanas, basándose en la distancia promedio.

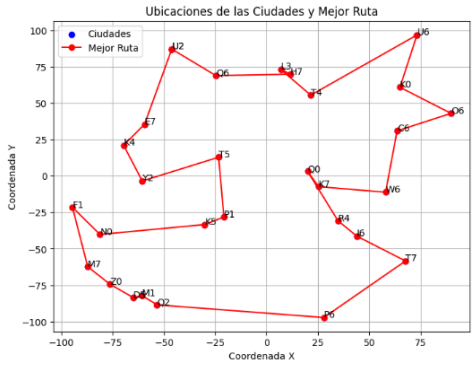
Caso 1. con tolerancia 0,20 y limite 120 s.

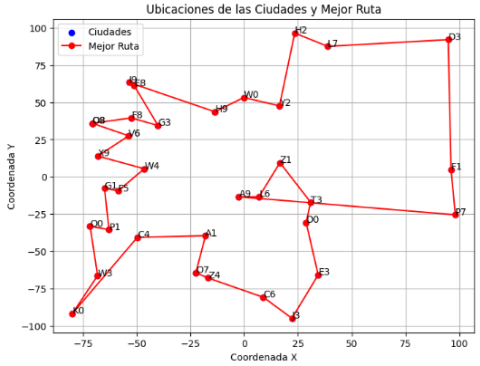
Plot para 10 y 20 ciudades:



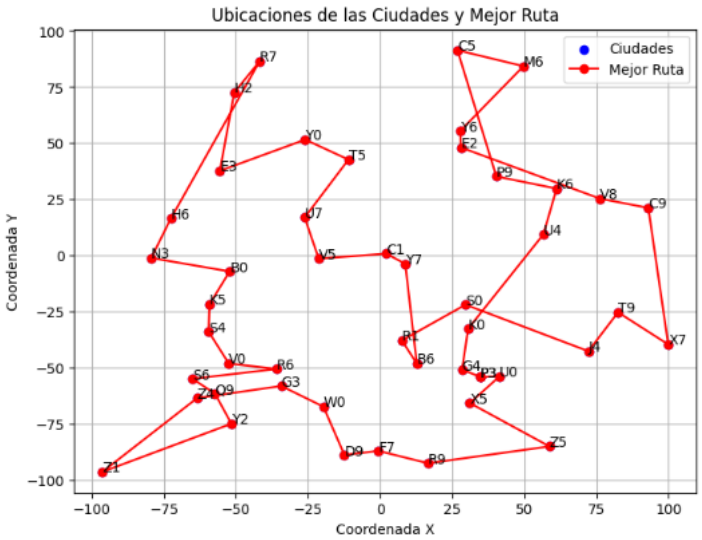


Plot para 30 y 40 ciudades:





Plot para 50 ciudades:



2.2. Analizar el parámetro tee

El parámetro tee se encuentra deshabilitado, activarlo para cualquier interacción, investigar su funcionalidad y comprender su salida en pantalla.

El parámetro tee sirve para controlar la visualización de la salida del solver en la el terminal durante la ejecución del modelo. Cuando tee está establecido en True, la salida generada por el solver se muestra en tiempo real, lo que nos permite seguir el proceso de solución, incluidos los mensajes de diagnóstico, errores, advertencias, y el progreso hacia la solución óptima. Cuando tee está en False, la ejecución del solver es silenciosa y no se muestra ninguna salida en la consola.

El parámetro Tee puede ser útil para:

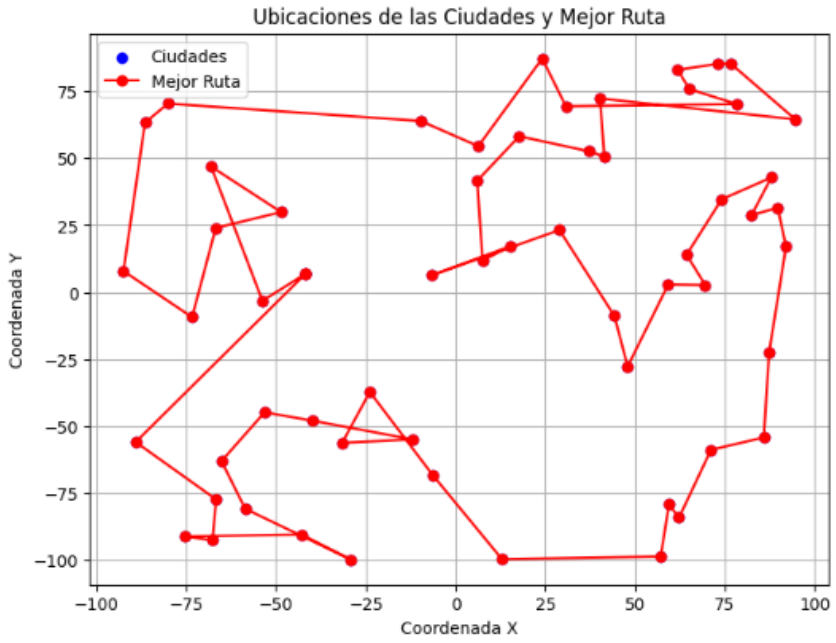
- Diagnóstico: Permite identificar y resolver problemas en el modelo de optimización, como errores en las restricciones o en la función objetivo.
- Monitoreo: Ofrece la posibilidad de monitorear el progreso de la solución, útil especialmente para modelos que requieren tiempos de ejecución largos.
- Análisis de rendimiento: Ayuda a entender el comportamiento del solver, incluyendo la efectividad de las heurísticas, el tiempo de ejecución, y las iteraciones necesarias para alcanzar la solución óptima o una solución factible cuando se aplican límites de tiempo o condiciones de parada.

2.3. Aplicar heurística de límites a la función objetivo

Ejecutar el caso 2, con 70 ciudades. Se ha colocado una heurística que trata de predecir cuanto será el recorrido total. Realizar la corrida de este caso: a) con heurística y b) sin heurística. Explicar los resultados y responder.

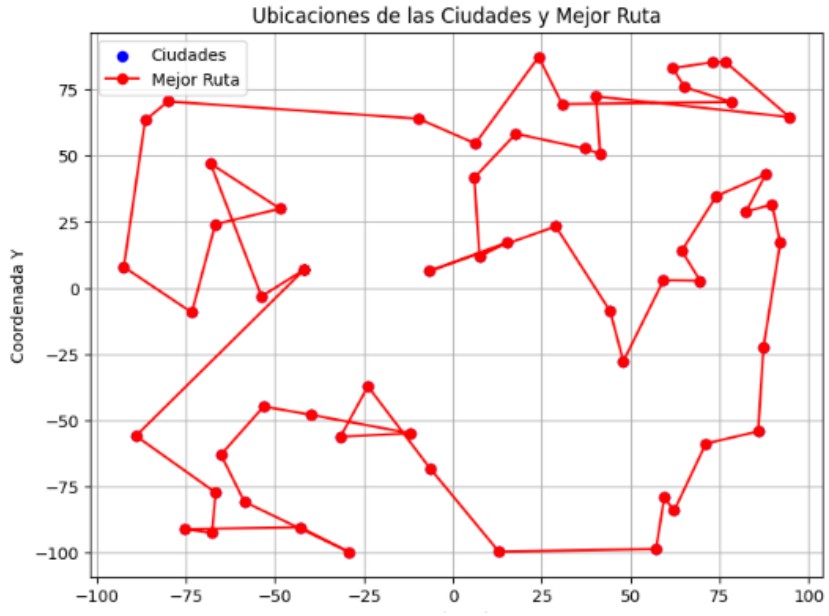
Resultados con 70 ciudades con Heuristica de límites a la función objetivo:

```
Tiempo de ejecución: 01:00
Distancia mínima entre nodos: 3.601388621073815
Distancia máxima entre nodos: 232.78868099630614
Distancia promedio entre nodos: 106.67822216080347
Distancia Total mínima posible: 1138.6369813228828
Distancia Total máxima posible: 1626.62425903269
Heurísticas aplicadas: ['limitar_funcion_objetivo']
No se encontró una solución óptima, la siguiente es la mejor solución encontrada:
Distancia total recorrida: 1502.540639439649
Resultados con heurísticas: Distancia total recorrida = 1502.540639439649
```



Resultados con 70 ciudades sin Heuristica:

```
Tiempo de ejecución: 01:00
Distancia mínima entre nodos: 3.601388621073815
Distancia máxima entre nodos: 232.78868099630614
Distancia promedio entre nodos: 106.67822216080347
Distancia Total mínima posible: 1138.6369813228828
Distancia Total máxima posible: 1626.62425903269
Heurísticas aplicadas: []
No se encontró una solución óptima, la siguiente es la mejor solución encontrada:
Distancia total recorrida: 1437.0024299932688
Resultados sin heurísticas: Distancia total recorrida = 1437.0024299932688
```



1. ¿Cuál es la diferencia entre los dos casos?
Con Heurística: La distancia total recorrida es de 1502 unidades.

Sin Heurística: La distancia total recorrida es de aproximadamente 1437 unidades.

La distancia total recorrida es menor sin la heurística, lo que sugiere que la heurística de limitar la función objetivo no puede ayudar a guiar al solver hacia soluciones de menor distancia total.
Por otro lado, las imágenes muestran que las rutas son uy similares, las diferencias son mínimas. Probablemente con una heurística bien diseñada podria mejorarse la calidad de las soluciones encontradas por el solver de optimización en un tiempo de ejecución dado.

La heurística de limitar la función objetivo parece no ser efectiva para este caso particular del TSP con 70 ciudades, ya que ayudó a obtener una solución de mayor distancia total (o casi igual) en comparación con la ejecución sin heurística, en el mismo límite de tiempo.

2.4. ¿Sirve esta heurística para cualquier caso? ¿Cuál pudiera ser una razón?

La heurística de limitar la función objetivo puede no ser efectiva en todos los casos. Su efectividad depende de la precisión de las estimaciones utilizadas para definir los límites. Si las estimaciones de la distancia mínima y máxima son representativas del conjunto de soluciones factibles, entonces la heurística puede ser muy útil. Pero si las estimaciones son inexactas, podría resultar en excluir soluciones óptimas o guiar al solver hacia áreas menos prometedoras del espacio de soluciones.

En este caso por ejemplo la heurística no redujo la distancia total recorrida. Esto puede deberse a que las estimaciones utilizadas para los límites fueron buenas aproximaciones para el conjunto de soluciones factibles del TSP.
La utilidad de la heurística puede variar dependiendo de:

- La distribución geográfica de las ciudades.
- La estructura específica del TSP (por ejemplo, la presencia de clusters de ciudades).
- El tamaño del problema (el número de ciudades involucradas).
- Si las ciudades están distribuidas de manera que las rutas más cortas entre ellas tienden a ser similares en longitud y no hay una variabilidad significativa en las distancias, una heurística basada en promedios y extremos puede no ser tan útil.

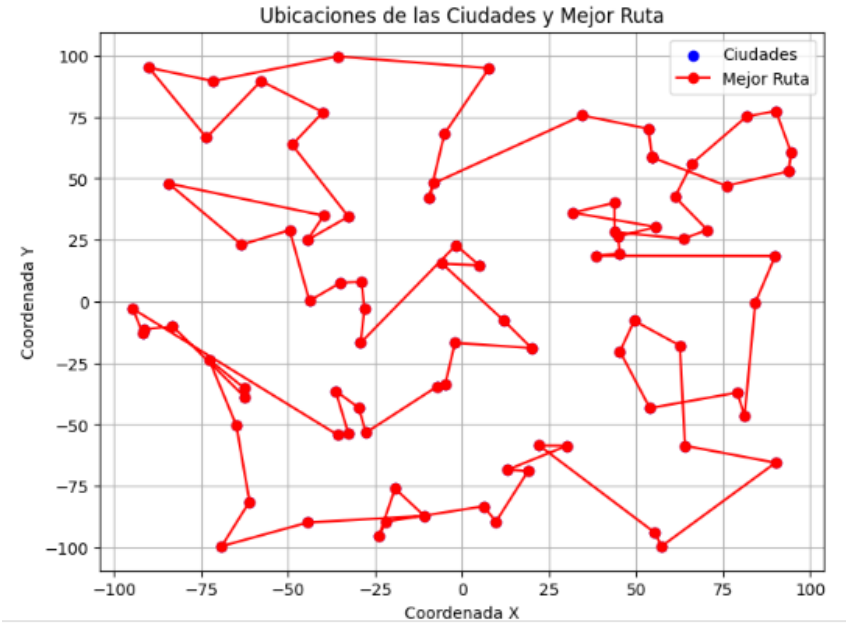
Como se ve en las gráficas, hay varios cruces entre caminos lo cual indica que aún esta lejos de ser la solucion óptima.

2.5. Aplicar heurística de vecinos cercanos

Ejecutar el caso 3, con 100 ciudades. Se ha colocado una heurística de vecinos cercanos. Realizar la corrida de este caso: a) con heurística y b) sin heurística. Explicar los resultados y responder.

Resultados con 100 ciudades con Heuristica del vecino mas cecano:

```
Tiempo de ejecución: 01:01
Distancia mínima entre nodos: 1.6492422502470616
Distancia máxima entre nodos: 243.90295201165566
Distancia promedio entre nodos: 99.15533568800747
Distancia Total mínima posible: 1534.7496991099251
Distancia Total máxima posible: 2192.499570157036
Heurísticas aplicadas: []
No se encontró una solución óptima, la siguiente es la mejor solución encontrada:
Distancia total recorrida: 1756.4471264365548
Resultados sin heurísticas: Distancia total recorrida = 1756.4471264365548
```



Resultados con 100 ciudades sin Heuristica:

```
Tiempo de ejecución: 01:01
Distancia mínima entre nodos: 1.6492422502470616
Distancia máxima entre nodos: 243.90295201165566
Distancia promedio entre nodos: 99.15533568800747
Distancia Total mínima posible: 1534.7496991099251
Distancia Total máxima posible: 2192.499570157036
Heurísticas aplicadas: ['vecino_cercano']
No se encontró una solución óptima, la siguiente es la mejor solución encontrada:
Distancia total recorrida: 1824.3277622713233
```