Pontificia Universidad Católica del Ecuador

Facultad de hábitat, infraestructura y creatividad

TEMA:

Aplicación de las ciencias de datos para segmentación de estudiantes en una IES

AUTOR:

PABLO GIOVANNY CAIZA IZA

TRABAJO PREVIA A LA OBTENCIÓN DEL TÍTULO DE MAGISTER EN SISTEMAS DE

INFORMACIÓN MENCIÓN DATA SCIENCE

TUTOR:

MSC. MIGUEL DIMITRI ORTIZ NAVARRETE

Quito, Marzo – 2025

AGRADECIMIENTO

Es importante expresar mi gratitud a mi tutor de tesis, ya que su conocimiento y guía en este trabajo fue crucial para la culminación de este proceso, agradecer también a la Pontificia Universidad Católica del Ecuador por la experiencia académica y por crear un ambiente propicio que apoya la realización profesional y personal, además agradecer al Instituto Cordillera por el apoyo incondicional de sus autoridades su compromiso con la formación de sus colaboradores y finalmente a mis docentes que supieron guiar su experiencia y experticia elementos que son indispensables para lograr este objetivo.

DEDICATORIA

Dedico este trabajo enteramente a mi familia por su compromiso, sacrificio y apoyo incondicional, sin el cual no se podría alcanzar este objetivo, Agradecer a todos las personas que directa o indirectamente me brindaron su ayuda y apoyo en este proceso.

RESUMEN

En este estudio de segmentación de estudiantes para una Institución de Educación Superior se utilizó técnicas avanzadas de ciencia de datos para determinar patrones en la información de matrículas de los estudiantes de los últimos 5 años, se utilizaron algoritmos de agrupación como K-Means, K-Modes, K-Prototypes, DBSCAN y un algoritmo Hibrido Neuronal Autoencoder + Kmodes para determinar cuál es el mejor modelo para definir patrones en el contexto educativo, el algoritmo K-Prototypes fue el mejor modelo seleccionado para las variables académicas y demográficas que se utilizaron con un rendimiento del 60% evaluado con el coeficiente de silueta y la distancia de hamming.

Se utilizó la metodología CRISPDM para guiar el proceso de ciencia de datos, sus fases permitieron el análisis de toda la data y permitió encontrar segmentos representativos dentro de la información, del modelo elegido se identificaron insights muy valiosos que se presentó en el análisis de resultados y que permitió desarrollar e implementación de estrategias educativas derivadas del análisis de las variables involucradas, lo que representó a la institución una oportunidad de mejora y personalización el servicio educativo.

ABSTRACT

In this student segmentation study for a Higher Education Institution, advanced data science techniques were used to determine patterns in student enrollment information from the past 5 years. Clustering algorithms such as K-Means, K-Modes, K-Prototypes, DBSCAN, and a Hybrid Neural Autoencoder + Kmodes algorithm were used to determine the best model for defining patterns in the educational context. The K-Prototypes algorithm was the best selected model for the academic and demographic variables used, with a performance of 60%, evaluated with the silhouette coefficient and Hamming distance.

The CRISPDM methodology was used to guide the data science process. Its phases allowed for the analysis of all the data and allowed for the identification of representative segments within the information. From the chosen model, very valuable insights were identified that were presented in the analysis of results and that allowed for the development and implementation of educational strategies derived from the analysis of the variables involved, which represented an opportunity for the institution to improve and personalize the educational service.

Contenido

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. CAPÍTULO I: INTRODUCCIÓN

"La segmentación de clientes es el proceso de clasificar a los clientes en diferentes grupos en función de características que tengan en común, como preferencias, comportamientos o datos demográficos" (Contentsquare, 2024)

Las IES gestionan masivamente información de sus estudiantes, esto abre una oportunidad para identificar información importante para la gestión educativa.

En el caso del Tecnológico Superior Cordillera una institución con 30 años de trayectoria en servicios educativos la ausencia de herramientas y personal capacitado Figura 1 *Situación Actual de la IES* ha limitado la capacidad de explorar la información para identificar segmentos significativos de la información demográfica y académica disponible.

Este trabajo de titulación pretende identificar segmentos significativos de estudiantes utilizando variables demográficas y académicas, haciendo uso de la ciencia de datos, identificando información importante que no se puede determinar a simple vista, esto requiere del uso de una estructura metodológica robusta que guie el análisis en grandes volúmenes de datos, para explotar estas variables.

Dicha estructura metodológica permitirá abordar el problema desde una perspectiva integral, desde la preparación hasta la interpretación pasando por todas sus fases como la recolección y limpieza de datos, el análisis exploratorio, la aplicación de técnicas de segmentación , la validación, evaluación de los segmentos y la interpretación, generación de insights, esta estructura no solo organizará el proceso de segmentación si no que maximizará el valor de la información disponible de la IES, mejorando la retroalimentación para mostrar posibles acciones

frente a los segmentos evidenciándolos y que otros responsables puedan tomar acciones frente a ellos.

1.1.Planteamiento del Problema

La IES de orden técnico y tecnológico dado su naturaleza periódica de captación de estudiantes generan y almacenan volúmenes de información considerables, datos demográficos como edad sexo, lugar de residencia y modalidad de estudio entro otras, suelen ser utilizados y actualizados mayormente para cumplir con informes solicitados por los entes regulares (Caces, 2024), pero no para algún tipo de análisis profundo.

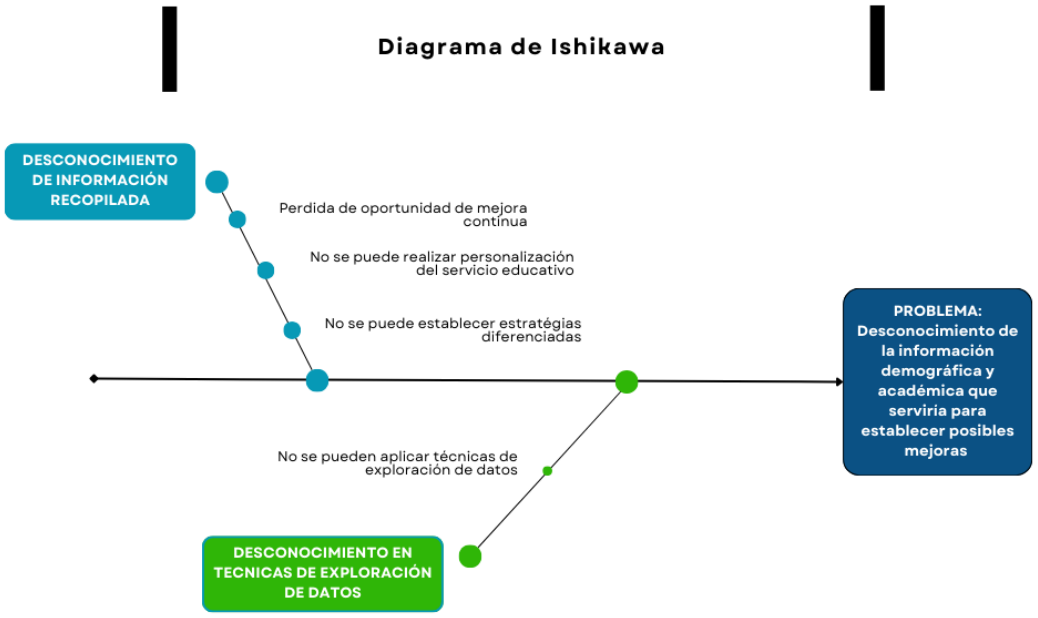
Esta cantidad de información no está siendo utilizada como una herramienta para identificar características comunes importantes, este factor es una oportunidad para aplicar técnicas de segmentación que permitan descubrir características comunes y posteriormente analizar oportunidades de mejora y personalización del servicio educativo, actualmente no se ha realizado estudio alguno con ciencia de datos

Como se describe en la Figura 1 *Situación Actual de la IES.* El desconocimiento de la información demográfica y académica disponible, junto con la falta de conocimiento en técnicas de exploración de datos, es una limitante para descubrir patrones y relaciones significativas entre variables, esta es una oportunidad valiosa para utilizar algoritmos como K-Modes, K-Means o DBSCAN (Ramirez, 2023).

La aplicación de estas técnicas de agrupamientos de datos permitiría a la IES conocer y apoyar a la personalización del servicio educativo estableciendo posibles estrategias basadas en los hallazgos conociendo de mejor manera las necesidades de los estudiantes.

Figura 1

Situación Actual de la IES



Nota. *Fuente: desarrollo del autor.

1.2. Justificación

"La segmentación de mercado en las instituciones educativas es clave para identificar grupos con características comunes" (Hemsley-Brown, 2020) , variables como la edad, sexo, parroquia de residencia, jornada, étnica, raza, estado civil, carrera, modalidad, nivel, jornada, colegio entre otras que se puedan identificar nos permiten comprender las características de la población estudiantil que permitan identificar segmentos que a simple vista no podría ser visibilizados como la distribución de preferencias por variables, patrones de comportamiento, comunidades pequeñas o atípicas, necesidades específicas, información cuya relación no es obvia sin un análisis profundo.

El presente trabajo de titulación busca aplicar técnicas de segmentación que permitan identificar grupos de estudiantes con características comunes, basándose principalmente en estas variables demográficas, el objetivo es utilizar los datos disponibles en un periodo de tiempo de 5 años de información histórica de la IES y generar segmentos relevantes que puedan ser utilizados para posterior planificación y toma de decisiones (Bowen, 2024).

1.3. Objetivos

1.3.1. Objetivo General

Aplicar técnicas, métodos y metodologías de ciencia de datos para la segmentación de mercados en una IES utilizando variables demográficas y académicas

1.3.2. Objetivos Específicos

- Aplicar técnicas de extracción y limpieza de datos para obtener información requerida de la IES.
- Analizar metodologías para la segmentación de datos que sustenten el proyecto.
- Aplicar técnicas de segmentación en de datos de una IES, para identificar grupos significativos.
- Aplicar técnicas de visualización para presentar datos e identificar segmentos importantes para el mercado

1.4. Alcance

El presente proyecto de titulación engloba el análisis y segmentación de los datos de estudiantes dentro de una IES en la ciudad de Quito, utilizando técnicas de segmentación y

agrupamiento, haciendo uso de variables demográficas y académicas como edad, sexo, parroquia de residencia, jornada, étnica, raza, estado civil, carrera, modalidad, nivel, jornada, colegio entre otras, para identificar grupos homogéneos que compartan características comunes.

El proyecto abarcará las siguientes actividades principales:

Extracción y preparación *de los datos a analizar, s*e va a acceder a la base de datos de la IES institucional vía SQL, se extraerán los datos de los últimos 5 años de los periodos académicos disponibles, se realizará la limpieza valores faltantes, valores duplicados e inconsistencias, transformación de los datos, esto garantizará que sean de calidad para el análisis posterior.

Análisis y exploración, *s*e realiza en esta fase una inspección de las variables para identificar posibles relaciones patrones y valores atípicos u outliers.

Aplicación de algoritmos de agrupamiento, el proyecto analizará algoritmos de agrupamiento jerárquico, K-Means K-Modes, DBSCAN K-Prototypes, Autoencoder + K-Modes, tomando en cuenta sus ventajas y limitaciones, según los datos disponibles y los objetivos planteados, K-Means requiere definición previa de clústeres utilizando datos numéricos, mientras que K-Modes es eficiente para volúmenes de datos, por el contrario DBSCAN identifica clústeres de forma arbitraria sin definir un número, siendo útil para datos con ruido, la elección del modelo idóneo dependerá de las particularidades de los datos y los objetivos del análisis, decidiendo por el enfoque más adecuado, K-Prototypes nos permitirá explotar las variables categóricas y numéricas mientras que Autoencoder (neuronal) y K-Modes nos dará un enfoque hibrido con reducción a una representación más compacta para obtener segmentos más homogéneos

Validación de resultados, en este apartado se evaluará la calidad de los grupos encontrados, utilizando métricas como el índice de silueta, el método del codo y la distancia de Hamming para

Visualización y reportes, se generarán gráficos que representen la distribución de los segmentos y la interpretación de sus características, gráficos de distribución para explicar los segmentos.

Delimitaciones del proyecto, el análisis se enfoca de forma exclusiva en los datos proporcionados por la institución, no se incluyen en el estudio factores externos como los socioeconómicos, las sugerencias derivadas de los resultados se orientarán a mejorar la toma de las decisiones internas sin implicar estrategias operativas detalladas, el estudio no incluye la etapa de implementación o despliegue.

Resultados esperados, identificación de segmentos homogéneos de estudiantes, herramientas visuales que permitan la comprensión y comunicación de los hallazgos, recomendaciones basadas en los datos encontrados para apoyar la planificación estratégica institucional.

1.5. Unidad de Análisis

El Tecnológico Universitario Cordillera es una institución educativa residente en la ciudad de Quito Ecuador con más de 30 años de trayectoria brindando servicios educativos, oferta carreras en varias modalidades y áreas como la salud, finanzas, tecnología, educación inicial, diseño, producción, recursos humanos (Tecnológico Universitario Cordillera, 2024), la institución ha enfrentado varios cambios generacionales de administración y gestión, actualmente hay una gran necesidad institucional de tomar decisiones informadas desde el punto de vista del análisis de datos, cuenta con un promedio de 4000 estudiantes por periodo académico luego de la pandemia y anterior a la pandemia 6000, maneja una base de datos de información estructurada,

El análisis de la información de los estudiantes para evidenciar patrones en los datos demográficos y académicos se va a centrar en el Tecnológico Universitario Cordillera, se firmó un acuerdo de confidencialidad que nos permite el acceso a los datos para el desarrollo del estudio que se encuentra en el ANEXO 3

2. CAPÍTULO II: MARCO TEÓRICO Y CONCEPTUAL

2.1. Segmentación de clientes

La segmentación de datos es un enfoque que se originó en el marketing, según Cristóbal Fernández Revista Colombiana de Marketing se define como “el proceso de dividir un mercado en segmentos o grupos identificables, más o menos similares y significativos” (Fernandez, 2001)

Con base a esta afirmación se puede definir que el proceso de la segmentación se basa en dividir un conjunto variado (heterogéneo) en subconjuntos con características comunes (homogéneo), en este proceso de toman en cuenta características comunes compartidas, la segmentación se ha llevado a diferentes áreas entre las cuales el área educativa

Tipos de segmentación:

- Segmentación geográfica, se basa en la ubicación física de clientes utilizando variables como país, ciudad, región.
- Segmentación demográfica, que ocupa características como edad, sexo, profesión entre otras.
- Segmentación psicográfica, considera características como personalidad, estilo de vida y valores.
- Segmentación conductual se basa en analizar las variables del comportamiento de consumo.
- Segmentación por industria, está enfocada en sectores de mercado específicos.
- Segmentación por productos, se basa en las características propias del producto

Las empresas tienen varias formas de segmentar o agrupar a sus clientes, utilizando las variables de las cuales se disponen, si se dispone de datos de ubicación como país, ciudad, sector utilizaran la segmentación geográfica, si disponen de datos como intereses, valores o estilo de vida utilizaran segmentación psicográfica, en fin como se menciona es a partir de la data y los interés que se buscan encontrar, todo esto con objeto de conocer y entender bien a los clientes para atraerlos y mantenerlos informados, obviamente sin invertir muchos recursos en este proceso.

En el caso de estudio de la IES, la segmentación permitirá dividir grupos de estudiantes con características similares, comunes, utilizando para este propósito variables demográficas y académicas se espera recomendar estrategias para cada grupo encontrado y de esta forma mejorar el servicio educativo.

2.2.Algoritmos de Agrupamiento

Los algoritmos de agrupamiento, como K-Modes, K-Means y DBSCAN, K-Prototypes, son métodos de aprendizaje no supervisado que ayudan a identificar patrones en los datos (Chambi, 2022).

“El algoritmo K-Means realiza agrupación de datos basados en la proximidad de características, buscando minimizar la variabilidad dentro de los grupos, por otro lado DBSCAN identifica grupos basados en la densidad de puntos, lo que lo hace particularmente útil en la identificación de clústeres de diferentes formas o tamaños” (Sevilla, 2024).

2.2.1. K-Modes

“El algoritmo K-Modes es un algoritmo que utiliza las diferencias desajustes totales entre los puntos de datos” (Bonthu, 2021), este algoritmo ha sido utilizado mayormente en la segmentación de mercados y se aplicaría en la agrupación en la gestión académica en la IES

principalmente para mejorar el análisis de grandes volúmenes de datos (como en el tamaño de nuestra muestra data).

Es una variante del algoritmo K-Means diseñada específicamente para datos categóricos, en lugar de utilizar la distancia euclidiana K-Modes mide la similitud entre puntos de datos utilizando una distancia basada en la coincidencia de valores categóricos.

Los centroides de los clusters se definen como la moda (el valor más frecuente) de las categorías, al disponer de datos categóricos en el proyecto para la IES el algoritmo se puede usar para describir grupos de segmentación.

- El algoritmo consta de los siguientes pasos:
- Generar aleatoriamente (k) centroides en espacios del conjunto de datos.
- Obtener las diferencias (dissimilarities) de cada uno de los datos a todos los centroides y asignar al grupo cuya distancia sea menor.
- Estimar la moda en cada uno de los grupos para actualizar los centroides.
- Comprobar si los centroides se han desplazado por debajo de un valor límite, si es así esta es la solución, en caso contrario volver al punto 2.

2.2.2. K-Means

"Es un algoritmo de clustering que agrupa los datos en K clusters en función de la proximidad entre los puntos de datos" (Lloyd, 1982), aunque se utiliza principalmente para datos numéricos, en este proyecto se emplea para comprender la comparación con K-Modes en su capacidad de identificar patrones en datos numéricos al disponer de estos o realizar

transformaciones en la data para obtenerlos también se los puede utilizar para el análisis de segmentos

El algoritmo consta de los siguientes pasos:

- Inicialización, una vez escogido el número de grupos, k, se establecen k centroides en el espacio de los datos, por ejemplo, escogiéndolos aleatoriamente.
- Asignación objetos a los centroides, cada objeto de los datos es asignado a su centroide más cercano.
- Actualización centroides, se actualiza la posición del centroide de cada grupo tomando como nuevo centroide la posición del promedio de los objetos pertenecientes a dicho grupo.

2.2.3. DBSCAN

"Es un algoritmo de clustering que agrupa los datos en función de su densidad, detectando grupos de puntos que están cercanos entre sí en un espacio de datos, a diferencia de K-Means o K-Modes, DBSCAN no requiere especificar el número de clusters y es capaz de detectar outliers o puntos de ruido que no pertenecen a ningún clúster" (Ester, 1996)

DBSCAN se puede adaptar para trabajar con datos categóricos al utilizar distancias personalizadas en el caso de estudio podría ayudar a detectar grupos de estudiantes con características inusuales, o detectar datos atípicos en las variables demográficas, lo cual puede ser valioso en el análisis de necesidades especiales.

DBSCAN se basa en lo siguiente:

- Calcula la matriz de distancias entre los distintos puntos generalmente se utiliza la distancia Euclídea, aunque se pueden usar otras.

- Teniendo en cuenta los parámetros del modelo, clasifica a cada punto entre punto core, punto frontera y ruido, en este sentido, puede que salgan diferentes puntos core ya que puede haber varias zonas de densidad, cada uno de esos puntos core pertenecerá a un clúster.

2.2.4. K-Prototypes

El Algoritmo K-Prototypes es un algoritmo de clustering que combina la idea de K-Means (para datos numéricos) y K-Modes (para datos categóricos), es usado para datos que tienen estas dos mezclas de variables en un formato mixto, utiliza la distancia euclidiana para las variables numéricas y comparación de valores para las variables categóricas.

K-Prototypes funciona de la siguiente forma

- Se elige los clusters iniciales en k grupos con promedios para números y valores más cercanos para las categorías

- Se asigna cada dato a un cluster, según la distancia numérica y las diferencias categóricas

- Se actualizan los clusters, re recalculan los promedios y las modas de cada grupo con los datos que se asignan.

- Se repite el proceso hasta que los datos ya no cambien de grupo de forma significativa

2.2.5. Algoritmo Mixto: Autoencoder + K-Modes

Este algoritmo mixto combina redes neuronales para reducir la dimensionalidad con las ventajas de K-Modes (clustering para datos categóricos) con el objetivo de mejorar la segmentación de los datos categóricos, ya que por sí solo K-Modes podría no ser mejor, se usa cuando las categorías tienen muchas dimensiones y relaciones no triviales (Sanz Angulo, 2019).

Este enfoque mixto mejora la calidad del agrupamiento y nos ayudará a capturar las relaciones complejas entre las categorías seleccionadas al reducir su dimensionalidad y el ruido de los datos categóricos, generaliza mejor en problemas con muchas categorías.

Este algoritmo mixto funciona así:

- El Autoencoder es una red neuronal que aprende a representar los datos en un espacio más compacto.
- Se aplica K-Modes sobre los datos ya transformados
- Cada dato original se asigna a un cluster basado en los que ya están creados

2.3.Metodología y Técnicas

La metodología que se seguirá en este trabajo es CRISP-DM (Cross-Industry Standard Process for Data Mining), una metodología estándar probada que se utilizada en proyectos de análisis de datos principalmente por características como el enfoque iterativo y cíclico que garantiza la mejora continua al poder volver a una etapa anterior para mejorarla.

Esta metodología guía el proceso del proyecto través de fases que están claramente definidas, desde la comprensión del problema hasta la evaluación de los resultados, en este estudio, se utilizará CRISP-DM para segmentar a los estudiantes de una IES utilizando variables demográficas mediante el uso de técnicas de agrupamiento (K-Modes- K-Means, DBSCAN, K-Prototypes y algoritmos Mixtos)

Las fases de CRISP-DM que se aplicarán en el estudio son las siguientes:

2.3.1. Comprensión del Negocio

En esta fase iniciaremos el proyecto definiendo el problema y los objetivos de la IES, se identifican las preguntas que el estudio debe responder, se determinan aquí los recursos como los datos, herramientas los beneficios esperados, las restricciones y las limitaciones.

2.3.2. Comprensión de los Datos

En esta fase se recopilan, analizan y exploran los datos disponibles de los estudiantes de la IES, se ejecutará una revisión inicial de la calidad y estructura de los datos para garantizar que son adecuados para el estudio.

Se detallan los datos que van a ser obtenidos de la base de datos de la institución y contendrán variables como la edad, el sexo, la parroquia de residencia entre otras que pueden apoyar a la identificación de los segmentos también se identificará si los datos contienen valores atípicos o nulos que puedan afectar el análisis y se tomarán decisiones sobre cómo tratarlos si eliminarlos mantenerlos o procesarlos.

2.3.3. Preparación de los Datos

En esta etapa con los datos seleccionados y comprendidos se proceden a limpiar y transformarlos para la fase del modelado, se realiza además de la limpieza, eliminación de registros duplicados o erróneos y normalización de las variables, la transformación de los datos en un formato adecuado para los algoritmos de segmentación, aquí se asegura que los datos sean consistentes y estén listos para todo el análisis utilizando los algoritmos de clusterización que se definieron.

Es importante señalar que antes de utilizar algoritmos de clustering es necesario que los datos numéricos se encuentren dentro de una escala común, hay que identificar límites razonables, por ejemplo una edad de 120 años en el conjunto de datos de la IES probablemente deba ser considerado como un outlier (dato atípico), dependiendo del análisis se debe tomar una decisión reemplazar por la media o eliminar el dato.

Método del rango intercuartílico, esta técnica estadística se utiliza para detectar valores atípicos dentro del conjunto de la IES, se revisará la dispersión de los datos mediante diagramas de bigotes o bloxplot por ejemplo dentro del contexto de nuestros data no podrían existir estudiantes de más de 100 años

Manejo de valores nulos, para tratar los valores faltantes o no asignados se puede utilizar la técnica de imputación por moda (valor más frecuente) en la variable que se analice, en nuestro conjunto de datos también se puede categorizar el valor faltante si no afecta significativamente al análisis (como “No especificada”).

Normalización, esta técnica de preprocesamiento de datos ajusta los valores de las variables a un rango específico (por lo general entre 0 y 1) de esta forma se garantiza que las variables

numéricas tengan una escala comparable evitando rangos que dominen el análisis, pero esto dependerá del algoritmo que se vaya a utilizar

Estandarización esta técnica se usa para convertir los datos numéricos a una distribución estándar con una media de 0 y una desviación estándar de 1, permitiendo que las variables tengan escalas comparables y facilite su análisis principalmente en algoritmos que dependen de la distancia entre datos.

Transformación y codificación de variables, es necesario transformar las variables categóricas previo su uso cuando se utilizan algoritmos de clusterización como K-Means entre otros que trabajan con datos numéricos se detallan dos principales técnicas:

- One-Hot Enconding este método crea columnas binarias separadas para cada categoría de una variable por ejemplo si se tiene la variable modalidad valores como “presencial”, “virtual” e “híbrida” este método crear tres columnas una por categoría donde 1 representa si tiene ese valor o 0 en caso contrario.

- Label Enconding este método asigna un número único a cada categoría, por ejemplo, para la variable “estado civil” se puede codificar las categorías como soltero = 0, casado = 1, divorciado = 2, hay que tener cuidado con este método ya que algunos algoritmos interpretan estos valores como ordinales cuando no lo son

2.3.4. Modelado

En esta fase se aplicarán los algoritmos de segmentación a los datos preparados, los algoritmos utilizados serán K-Modes, K-Means, DBSCAN, K-Prototypes y Autoencoder + K-Modes, se evaluarán los resultados obtenidos por cada algoritmo para determinar cuál de ellos produce clústeres más significativos y útiles para la IES.

2.3.5. Evaluación

Una vez que los modelos han sido aplicados, se evaluarán los resultados para verificar si cumplen con los objetivos establecidos en la fase de comprensión del negocio, en esta fase se analizarán los segmentos generados por los algoritmos y se evaluará su utilidad, se realizarán ajustes en los algoritmos o en los datos si se considera necesario para mejorar los resultados.

Se organizan las técnicas en función de su propósito, facilita su aplicación desde las necesidades del análisis.

Métricas de calidad *de clústeres* estas métricas evalúan la cohesión interna y la separación entre los clústeres:

- Índice de Silueta esta métrica permitirá evaluar si los estudiantes están bien agrupados en función de las variables utilizadas como edad carrera, modalidad, los valores cercanos a +1 datos bien agrupados, y 0 sugiere que los puntos están en el límite de dos clusters, y valores a -1 reflejan una mala clasificación, esta métrica es muy útil para medir la cohesión interna de los grupos y su separación, en definitiva encuentra el número de clústeres que maximiza la puntuación media de silueta evalúa la compactación y separación de los clústeres comparando la distancia entre clústeres y también combina distancias euclidianas y de haming para algoritmos que utilizan combinación de variables categóricas y numéricas

- Distancia de Hamming, esta métrica mide las diferencias entre dos valores categóricos contando los elementos que son distintos, se aplica principalmente a los datos categóricos, es importante considerar que no considera magnitud de la diferencia, solo compara si los elementos son iguales o diferentes

Determinación del número óptimo de clústeres, estas técnicas ayudan a seleccionar el número más adecuado de clústeres:

- Método del codo K-Means, se utiliza para determinar el número óptimo de clusters, ya que se selecciona el número de clusters en el punto donde la curva de la suma de errores al cuadrado WCSS (inercia)se aplana, formando un codo.
- Método del codo K-Prototypes opera calculando el costo total para cada k, sumando la distancia euclidiana para atributos numéricos y la disimilitud para los atributos categóricos y se identifica el codo donde la disminución del costo se vuelve menos significativa.

Evaluación visual permite explorar y validar visualmente la distribución de los clústeres:

- PCA: Simplifica los datos a dos o tres dimensiones para su visualización.
- t-SNE: Representa datos complejos de alta dimensionalidad en espacios más simples.

Evaluación contextual, más allá de las métricas técnicas, verifica si los clústeres tienen sentido práctico y aportan valor

- Interpretabilidad: Los clústeres deben ser coherentes con el contexto del problema.
- Relevancia práctica: Los resultados deben facilitar la toma de decisiones o generar insights útiles.

2.3.6. Despliegue

En este estudio no se considerará esta etapa porque está por fuera del alcance del mismo, sin embargo en el análisis de resultados se pretende mostrar el análisis de los segmentos identificados que permitan una comprensión clara de los grupos de estudiantes, se proporcionarán recomendaciones y sugerencias iniciales con hallazgos obtenidos, hay que considerar que estos resultados pueden servir como base para futuras implementaciones en caso de que se desee ampliar al despliegue del modelo de una forma automatizada.

2.4.Fundamentos de programación en Python

Según el sitio web oficial de Python software foundation:

“Python es un lenguaje de programación potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un simple pero efectivo sistema de programación orientado a objetos. La elegante sintaxis de Python y su tipado dinámico, junto a su naturaleza interpretada lo convierten en un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en muchas áreas, para la mayoría de plataformas” (Python Software Foundation, 2024)

Python es un lenguaje interpretado, lo que significa que se ejecuta línea por línea y no requiere ser compilado antes de ejecutarse (Amazon, s.f.)

Gracias a su facilidad de uso y flexibilidad se ha convertido en una herramienta muy útil para tareas de desarrollo rápido y experimentación en varias actividades estas características permiten que se puedan utilizar en proyectos de casi todo tipo.

2.4.1. Características principales

Se detallan a continuación características que hacen de este lenguaje una opción muy viable.

- Sintaxis simple y legible, “La claridad del código de Python es una de sus mayores fortalezas, a diferencia de otros lenguajes que requieren delimitadores como llaves o punto y coma, usa la indentación para definir bloques de código” (Llamas, 2024) esta característica mejora indudablemente la lectura del código ya que típicamente las llaves pueden complicar la lectura a primera vista esto minimiza errores mejora el tiempo de depuración.

- Multiplataforma, esta es una característica que lo ha hecho muy utilizado ya que funciona en varios sistemas operativos sin necesidad de realizar modificación alguna en el código, con esto se garantiza proyecto en cualquier entorno (SISE, 2024) , varios lenguajes como Java utilizan este esquema escriba una sola vez y corra en cualquier parte y esta característica hace de este lenguaje una opción ideal

- Versatilidad, “Python es un lenguaje de programación potente y versátil que desempeña un papel fundamental en una amplia variedad de soluciones tecnológicas. Desde aplicaciones web, motores de búsqueda y juegos hasta software de animación e incluso otros lenguajes de programación, Python está en el corazón de la innovación” (DataCamp, 2024).
  Python se puede utilizar desde el aprendizaje básico hasta proyectos reales de implementación, además tiene una de las ventajas fuertes y es el uso de bibliotecas de su comunidad adapta para de varios entornos esto le permite resolver casi cualquier problema.

- Comunidad, al ser open source tiene una comunidad activa que incluye desarrolladores, investigadores e interesados del lenguaje crean actualizaciones y

soporte tan extenso que se torna una opción muy viable para la ciencia de datos, “Python dispone de una de las comunidades más activas, lo que es importante a la hora de encontrar soluciones a los problemas comunes y soporte, así como módulos en los que basarse a la hora de desarrollar programas” (Arsys, 2024)

- Bibliotecas integradas

  “Bibliotecas como TensorFlow, PyTorch, y Keras permiten a los desarrolladores construir y entrenar modelos de aprendizaje automático e IA de forma eficiente. Estas herramientas ofrecen excelentes características para el procesamiento de datos, la implementación de redes neuronales, y la realización de operaciones matemáticas complejas, lo que facilita en gran medida el desarrollo de sistemas de IA avanzados” (Godaddy, 2023).

  Existen bibliotecas que implementan recursos que no se encuentran en otros lenguajes lo que permitirá realizar las actividades del estudio.

Se han detallado las características más importantes del lenguaje Python, como lenguaje tiene la suficiente confianza y madurez para cumplir con los objetivos del trabajo de titulación, su sintaxis simple y legible facilitará la codificación de sus etapas y mejorará la interpretación de cada implementación, su característica de multiplataforma permitirá que el trabajo se realice dentro de ambientes locales y en la nube.

La comunidad activa de este lenguaje permitirá el acceso a documentación en foros y repositorios en los que se puede apoyar para encontrar soluciones a desafíos comunes dentro de la implementación de la segmentación de la IES.

Python como se detalla más adelante es muy sobresaliente en ciencia de datos gracias a la versatilidad de sus bibliotecas especializadas que van desde la preparación de los datos disponibles de la IES, la creación de los modelos de clustering y la evaluación del desempeño de estos modelos.

Analizándolo desde varias características Python es una elección útil para cumplir con el objetivo general del trabajo de titulación.

2.4.2. Aplicación de Python en la ciencia de datos

Python ha ganado mucho terreno en la ciencia de datos, este ámbito que combina métodos científicos y algoritmos avanzados para extraer conocimiento e insights de datos estructurados y no estructurados, su flexibilidad y su vasta cantidad de bibliotecas especializadas y su capacidad para manejar grandes volúmenes de información lo posicionan como una de las opciones más destacadas en este campo (Nodd3r, s.f.).

Algunas de sus principales aplicaciones dentro de ciencia de datos incluyen:

- Manipulación y limpieza de datos, Python facilita el procesamiento y limpieza de datos especialmente cuando se trabaja con data poco estructurada o incompleta, librerías como pandas permiten realizar tareas como eliminación de valores nulos, transformación de tipos de datos, filtrado y agregación de registros (Torres, 2021).

- Análisis estadístico, gracias a la compatibilidad con herramientas como SciPy y statsmodels Python soporta análisis estadístico avanzado, incluyendo pruebas de hipótesis, análisis de regresión y modelado probabilístico facilitando la exploración inicial de los datos, una solución muy práctica para aplicar la estadística descriptiva (Verne Academy, 2019).

- Visualización, Python ofrece variedad en herramientas de visualización de datos para crear gráficos personalizados y visualizaciones complejas, bibliotecas como Matplotlib, Seaborn y Ploty proporcionan funciones para gráficos estáticos y dinámicos, útiles al momento de comunicar hallazgos de forma clara y efectiva (Carmateck, 2023)

- Modelado predictivo y machine learning con bibliotecas como scikit-learn, Python facilita la implementación de algoritmos de clasificación, regresión y clustering, para proyectos más avanzados, como redes neuronales tensorFlow y PyTorch son importantes en aprendizaje profundo (Scikit-learn, 2022)

- Automatización de tareas repetitivas, Python permite construir scripts para automatizar procesos tareas como la extracción de datos, la generación de reportes y la actualización de bases de datos esto permite reducir el tiempo y los errores asociados con tareas manuales (Nuclio Digital School, 2023)

En el contexto del trabajo de titulación aplicando las características de Python que dispone para la ciencia de datos, este lenguaje se convierte en una herramienta importante para la implementación de la segmentación de estudiantes, la manipulación y limpieza da datos es crucial para preparar los datos históricos de 5 años disponibles se deben tratar valores nulos, faltantes, transformar tipos de datos como fechas de nacimiento, edades.

El análisis estadístico es una tarea inherente para explorar las características demográficas y académicas de los estudiantes, se pueden analizar correlaciones y análisis estadístico para comprender la relación entre las variables antes de aplicar las técnicas de segmentación para la IES.

Para la visualización de datos, Python permite la creación de gráficos, que muestren patrones de comportamiento en las variables de los estudiantes como la distribución por carreras, modalidades, jornada o edad, con librerías gráficas se pueden realizar histogramas, gráficos de dispersión y mapas de calor que permitan visualizar las relaciones que alimentarán nuestro modelo de segmentación

Para el aprendizaje automático (machine learning), scikit-learn permiten implementar algoritmos de clustering como K-Means o DBSCAN, K-Modes, K-Prototypes, con el objetivo de identificar grupos homogéneos de estudiantes, estas técnicas nos permitirán descubrir perfiles de los estudiantes con características comunes.

El enfoque de Python a la ciencia de datos se puede aplicar para garantizar la gestión eficiente de los datos y optimizar la implementación del modelo de segmentación de estudiantes para la IES.

2.4.3. Librerías más relevantes en Python para ciencia de datos

Este apartado destaca las herramientas más importantes de Python que han revolucionado el procesamiento y análisis de datos, convirtiéndose en estándares dentro del campo y permitirá realizar

- NumPy, ofrece soporte para la manipulación de arreglos multidimensionales y operaciones matemáticas avanzadas, actúa como el núcleo de otras bibliotecas, por a su capacidad para ofrecer una base eficiente para cálculos numéricos (NumPy, 2020).

- *Pandas* es la biblioteca más importante y conocida para manipulación de datos ya que facilita tareas como la agrupación el filtrado y la transformación, apoyado en estructuras denominadas dataframes (McKinney, 2010)

- *Matplotlib y seaborn*, son bibliotecas que se utilizan para generar visualizaciones, pero seaborn está orientado y muy optimizado para gráficos estadísticos, facilitando la interpretación de datos (Hunter, 2007).

- *Scikit-learn*, provee implementaciones de algoritmos de machine learning y también proporciona herramientas para el preprocesamiento de datos y la validación de modelos (Pedregosa, 2011).

En el desarrollo del trabajo de titulación las bibliotecas de Python son herramientas indispensables para lograr los objetivos del estudio.

Pandas se convierte en la herramienta esencial para la manipulación de los datos que se dispone, esta librería permite crear y trabajar con dataframes, facilitando tareas como la limpieza de datos la creación de variables derivadas, la agrupación de estudiantes por variables como jornada, carrera, modalidad, etc. así como patrones por parroquia o sectores.

Para la visualización de los resultados, Matplotlib y Seaborn nos permitirán crear gráficos descriptivos para visualizar la distribución de las características de las estudiantes reflejadas en las variables, con seaborn se pueden generar gráficos de densidad y mapas de calor, para identificar posibles grupos homogéneos de estudiantes, utilizando matplotlib se puede generar gráficos para los informes finales.

Estas librerías nos permitirán abordar cada una de las fases del modelo de la segmentación, comenzando por la importación de datos, la limpieza, la visualización la predicción la automatización lo que garantizará la eficiencia del modelo propuesto.

2.4.4. Potencial de crecimiento de Python

Python está en constante evolución como lenguaje de aplicación en varias áreas y campos tecnológicos y más en la ciencia de datos, dada su comunidad activa en el desarrollo y la facilidad que tiene a la adaptación a nuevas necesidades, su crecimiento en la industria es muy importante especialmente en áreas donde otros lenguajes eran estándares.

- Crecimiento en tecnologías emergentes, Python está ganando terreno en tecnologías como el Internet de las cosas (IoT) donde Java era el lenguaje por excelencia, se utiliza actualmente para desarrollar sistemas integrados y analizar datos generados por sensores (IoT Consulting, 2017).

- Mejoras continuas, la comunidad de desarrolladores y la Python Software Foundation trabajan continuamente en nuevas versiones que mejoran la eficiencia y añaden funcionalidades, como gestión de memoria y nuevas estructuras de datos (Python Software Foundation, 2020).

- Adopción empresarial, grandes empresas como las reconocidas Google, Facebook, Netflix y Spotify utilizan Python para tareas de análisis de datos, gestión de contenido y procesamiento del lenguaje natural, lo que apoya su importancia en el mercado laboral (Academia 4.0, s.f.)

- *Soporte educativo*, Python se ha consolidado como el estándar para aprender programación debido a su simplicidad y poder, formando una generación de estudiantes y profesionales que adoptarán Python como su herramienta principal (DataCamp Used, 2023)

- *Integración con nuevas tecnologías*, Python se adapta fácilmente a servicios que funcionen en la nube, a hardware especializado y herramientas de inteligencia artificial, ampliando su aplicación en el mundo digital (Unite.AI., 2023).

La mejora continua del lenguaje Python y su relevancia aseguran eficiencia en memoria y velocidad de procesamiento lo que se traduce en la capacidad de procesar datos históricos de estudiantes como los de la IES de forma rápida y eficiente.

2.5. Métricas de desempeño

"La agrupación es una técnica poderosa para encontrar patrones y estructuras en los datos, pero ¿cómo sabemos si las agrupaciones que obtenemos son significativas y confiables? La evaluación del agrupamiento es el proceso de medir la calidad y validez de los resultados del agrupamiento, que se puede realizar desde diferentes perspectivas y utilizando diferentes criterios" (FasterCapital, s.f.).

2.5.1. Coeficiente de silueta

"La Silhouette es una métrica que permite evaluar la calidad de los clústeres generados mediante algoritmos de clustering basados en la distancia euclídea. Como es el caso de k-means. Cuantificando la relación que existe entre la separación de los diferentes clústeres y la similitud entre los puntos de un mismo clúster en un valor que varía entre -1 y 1. Los valores cercanos a 1

indican la mejor separación de los clústeres y los cercanos a -1 la peor. Información que se puede utilizar para seleccionar el número óptimo de clústeres" (Rodriguez, Daniel, 2023).

El coeficiente de silueta es una métrica que permitiría medir la calidad de la segmentación de los estudiantes de la IES, esta métrica evalúa dos aspectos de los clústeres de la segmentación la cohesión y la separación.

Una vez que se determinen los clústeres de los estudiantes o los grupos, este coeficiente permite verificar si los grupos dentro de cada clúster en realidad se parecen entre si (cohesión) y si están diferenciados (separación),

Por ejemplo, podemos decir que si en un cluster se agrupan estudiantes de una misma jornada entonces la cohesión debería ser alta y este índice mostrará un valor cercano a 1, por otro lado, si en este cluster están mezclados con estudiantes que deberían estar en otros clústeres, al a separación va a ser menor y el valor de la silueta se acercará a 0 o valores negativos, este coeficiente se utilizara para decidir cuantos grupos hay que crear para la segmentación, si al usar por ejemplo 2 clústeres silueta es 0.6 y al usar 4 clústeres se obtiene 0.8 la opción optima será elegir 4 clústeres

2.5.2. Método del codo

"El método del codo es una técnica que se emplea para determinar el número óptimo de clústeres para el algoritmo de K-means. La idea detrás de este método es bastante sencilla. Identificar el número de clústeres para el que se observa un cambio significativo en la tasa de disminución de la varianza intra-cluster (también conocido como suma total de las distancias al cuadrado) para ello se debe ejecutar el algoritmo de k-means con diferentes números de clústeres

(k) y calcular la suma de las distancias al cuadrado de cada punto respecto a su centroide." (Rodriguez, Daniel, 2023).

Este método es una técnica que permite identificar la cantidad óptima de clústeres que se necesita para segmentar estudiantes de la IES, se basa en la observación de la inercia midiendo la distancia promedio de los estudiantes a los centroides de sus clústeres, la idea es buscar un punto de inflexión o codo donde la gráfica mejora la segmentación y debe ser significativa, evitando grupos muy pequeños y poco relevantes, facilitando la interpretación de los grupos por ejemplo estudiantes jóvenes estudiantes de la modalidad hibrida, se analiza el costo entendido como la distancia promedio.

2.5.3. Matriz de correlación de Cramér

"Es un coeficiente creado por el estadístico sueco Herald Cramer. Funciona como una medida de relación estadística basada en Ji cuadrado. Es decir, para hacer una corrección del coeficiente Ji Cuadrado donde se pueda precisar la fuerza de asociación entre dos o más variables. En este sentido, el resultado del coeficiente varía entre cero y uno (siendo cero un valor nulo de asociación)" (Isea, 2018).

Esta tabla nos permitirá medir que tan relacionadas están entre si las variables categóricas, nos permitirá por ejemplo medir la relación entre la jornada y la modalidad, con el fin de descartarla o no del estudio o realizar algún tipo de análisis.

3. CAPÍTULO III: MARCO METODOLÓGICO

3.1.Marco Metodológico de investigación

3.1.1. Tipo de Investigación

El desarrollo del proyecto corresponde a una investigación aplicada, dado que se busca transformar los conocimientos teóricos sobre segmentación en herramientas prácticas que permitan agrupar estudiantes de la IES con base en variables demográficas y académicas, este enfoque aplicado permitirá identificar patrones y características clave que puedan ser utilizados posteriormente para mejora en la toma de decisiones a la interna

3.1.2. Enfoque Metodológico

Se va a utilizar un enfoque cuantitativo, centrado en la recopilación y análisis de datos numéricos, categóricos, históricos de los estudiantes matriculados, esto permitirá identificar relaciones, patrones y grupos homogéneos aplicando técnicas estadísticas y de aprendizaje automático

3.1.3. Población y Muestra

La población incluye a todos los estudiantes matriculados en la IES en los últimos 5 años, la muestra se compone por todos los registros disponibles en este periodo de tiempo, asegurando la inclusión de datos relevantes para el análisis de segmentación, este conjunto de datos permite el análisis de las características de los estudiantes y su evolución a lo largo del periodo de tiempo seleccionado.

3.1.4. Recolección de Datos

La información que se va a utilizar como fuente principal de la data es la información de la base de datos del ERP institucional, que permite la operación de la IES aquí se puede encontrar la información académica y demográfica que se requiere para el estudio.

3.2.Marco Metodológico de ciencia de datos

Para la metodología en ciencia de datos, emplearé CRISP-DM, que es ampliamente aceptada ya que ofrece una guía estructurada y eficiente para llevar a cabo un proyecto de análisis de datos.

El primer paso es el entendimiento del negocio y aquí se buscará adquirir un conocimiento detallado sobre los objetivos y necesidades del estudio de segmentación de la IESS, aquí se van a establecer las preguntas que se desean responder y definir las guías para las siguientes fases de análisis.

En la etapa de comprensión de los datos, recolectaré y me familiarizaré con ellos utilizando la herramienta de acceso a la base de datos empresarial, este proceso permite identificar la calidad y la validez de los datos disponibles que van a utilizarse dentro del estudio ya que es vasta y variada y muy relevante para la segmentación de clientes

La extracción de los datos se realizará vía el lenguaje de consulta utilizando Toad MySQL que es un IDE (Entorno de Desarrollo Integrado) que se utiliza para interactuar con la base de datos MariaDB vía lenguaje SQL, permitiendo realizar consultas con bastante detalle y exportarlos al formato requerido para el estudio y encuentren listos para exportarlos a un archivo de tipo csv o xlsx.

Finalmente se va a importar los datos a la herramienta Google Colab con Python utilizando librerías estadísticas y de análisis para la exploración detallada realizando un examen de las variables a dentro del estudio, esto nos permitirá obtener una visión de la información.

Luego de esto se realizará la etapa de preparación de los datos, en esta fase realizan diversas tareas como la limpieza de datos y si es necesario la integración de información de otras fuentes, es importante para garantizar que los datos se encuentren en un formato correcto y apto para llevar a cabo la siguiente etapa.

En la etapa del Modelado a continuación se aplicará las técnicas de ciencia de datos y los algoritmos para explorar modelos y seleccionar el más idóneo, este proceso implica experimentación de técnicas y ajustes para optimizar el rendimiento del modelo.

En la etapa de evaluación que sigue al modelado se examina su desempeño para asegurar que cumple con los objetivos del entendimiento del negocio, aplicaré métricas de desempeño para medir la calidad de los segmentos para definir su ajuste o mejora.

Para la etapa del despliegue del modelo solo ejecutaremos un informe de resultados con la descripción de los hallazgos, sugerencias y recomendaciones de los mismos, ya que no se encuentran dentro de los alcances del estudio.

Utilizando el enfoque de CRISP-DM, planeo realizar un análisis estructurado y efectivo que ofrezca información importante y clave para la segmentación de los estudiantes en la IES.

4. CAPÍTULO IV: RESULTADOS

4.1.Aplicación de las técnicas de Minería de datos para la segmentación de clientes

Comprensión del negocio

4.1.1. Objetivo general:

Aplicar técnicas métodos y metodologías de ciencia de datos para identificar grupos homogéneos de la información de estudiantes basados en variables demográficas y académicas.

4.1.2. Objetivos específicos:

- Identificar grupos de estudiantes con características similares basándonos en sus datos académicos y demográficos
- Analizar patrones en los datos históricos de los últimos 5 años, utilizando algoritmos de agrupamiento.
- Detectar factores que diferencian los segmentos estudiantiles según la parroquia/sector de residencia carrera, modalidad, sexo.
- Analizar la distribución de estudiantes por jornadas, edad, modalidad, carreras para mejorar la planificación de recursos académicos.
- Proveer información para posterior toma de decisiones estratégicas en la institución educativa de acuerdo a los hallazgos encontrados

4.1.3. Conocimientos del dominio:

De acuerdo al dataset elegido “información de estudiantes de una IES” el contexto se centra en los datos históricos de registro de los estudiantes, datos académicos como la carrera, el nivel, la jornada, la modalidad y datos demográficos como la edad, sector y parroquia de residencia.

4.1.4. Restricciones:

- El estudio se limita a 5 años desde el periodo académico actual 2024-2025 hacia atrás.
- El estudio se limita a las carreras de pregrado ofrecidas por la IES.
- El estudio no incluye la fase de despliegue de CRISP-DM pero se detalla en el apartado un informe de resultados que puede servir como base para futuras implementaciones.

4.2.Comprensión de los datos

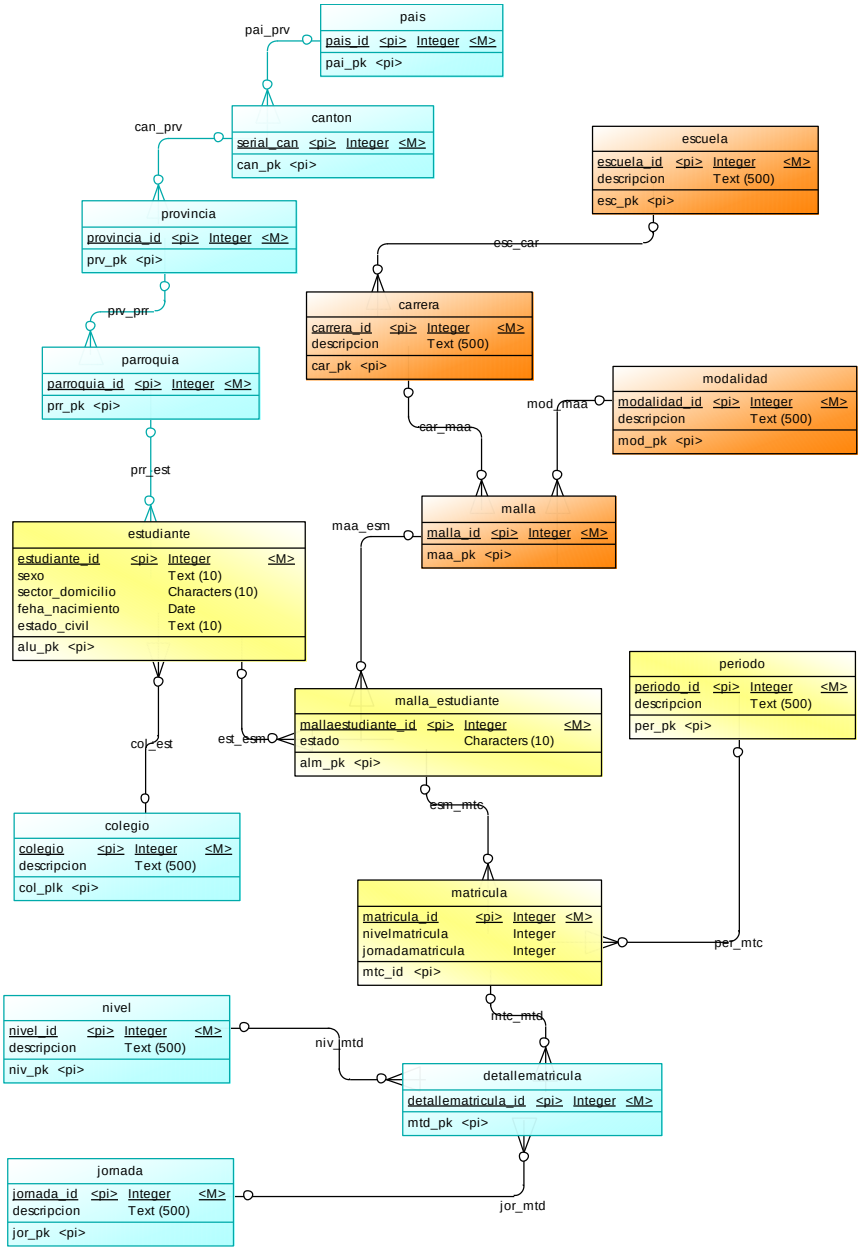
Como se mencionó previamente la IES dispone de una amplia variedad de información relacionada con los datos para el estudio del cual se detallan las variables a utilizarse para la segmentación

4.2.1. Estructura de Datos

Se detallan en la Figura *2* Modelo Conceptual Académico Origen , el modelo conceptual de la base de datos de las entidades que van a participar en el estudio, con los campos a utilizarse, se pretende abordar una visión del origen y contexto de los datos

Figura 2

Modelo Conceptual Académico Origen



Nota. *Fuente: desarrollo del autor

Se detalla la lista de entidades de la base de datos con el detalle del contexto y los datos a utilizar:

- **País:** En esta estructura se almacenan los países de domicilio de los estudiantes → a utilizar nombre del país.
- **Provincia:** En esta entidad se almacenan las provincias de domicilio de los estudiantes → a utilizar nombre de la provincia.
- **Cantón:** En esta entidad se almacenan los cantones de domicilio de los estudiantes → a utilizar nombre del cantón.
- **Parroquia:** En esta entidad se almacenan las parroquias de domicilio de los estudiantes → a utilizar nombre de la parroquia.
- **Colegio:** En el momento del ingreso de un estudiante se solicita el registro de este dato, se lo incluye para el análisis → a utilizar nombre del colegio.
- **Escuela:** Esta es la categorización general de las escuelas que se disponen en la IES → a utilizar nombre de la escuela.
- **Carrera:** Representa las carreras disponibles de la institución → a utilizar nombre de la carrera.
- **Modalidad:** En esta entidad se categoriza la modalidad de estudio asociada a una malla que va a tener un estudiante en el momento de iniciar sus estudios → a utilizar nombre de la modalidad.
- **Malla:** En esta entidad se almacenan las mallas académicas de las carreras y su nombre está asociado a una resolución en el momento de su aprobación → a utilizar nombre de la malla.

- **Periodo:** Esta entidad almacena los periodos académicos que se aperturaron para proceso de matrícula, por lo general se abren dos periodos por año que inician en abril y en octubre, contiene toda la información de inicio, fin de actividades y campos de estado para diferentes procesos → a utilizar nombre del periodo académico.

- **Nivel:** En esta entidad se categorizan los niveles de una malla y también se asocian a un horario que al final se encuentra atado al horario para poder ingresar una materia al detalle matrícula y también al maestro de la materia → a utilizar nombre del nivel.

- **Jornada:** En esta entidad se almacenan las jornadas que están asociadas a los horarios y se seleccionan en una matrícula → a utilizar nombre de la jornada.

- **Estudiante:** Es la entidad que representa al estudiante, la tabla que centraliza la información personal, demográfica, datos de residencia y en fin, la data que inicialmente se llena para permitir el ingreso a la institución de un estudiante. Se va a utilizar: sexo, fecha de nacimiento con la que se puede calcular la edad en el periodo académico, estado civil general del estudiante, sector de domicilio general del estudiante, el colegio que se hereda por relación y asociado a esto los datos de residencia, cuyo enlace principal es la parroquia que por cascada llega al país → a utilizar sexo, fecha de nacimiento, edad (calculada a partir de la fecha de nacimiento y la fecha periodo académico de estudio), estado civil, sector de domicilio, parroquia asociada (datos de estructura asociados a la parroquia).

- **Matrícula:** Esta entidad representa el maestro registro de matrícula de un estudiante en un periodo académico determinado. Aquí se relacionan la malla del estudiante activa en ese momento, el periodo académico y por herencia en cascada asocia a toda la estructura que se detalla en el gráfico. Se utilizarán dos campos: el nivel de la matrícula y la jornada de la matrícula, que se guardan en este maestro y también en el detalle → a utilizar nivel de la matrícula, jornada de la matrícula.

- **Detalle de matrícula:** Esta entidad almacena el detalle de la matrícula que está relacionada con el horario para registrar materias definidas por su malla y parámetros académicos como arrastres y máximos de materias a tomar. Si bien es cierto esta tabla se la incluye en el gráfico, pero es más bien para propósitos de conocimiento de estructura y para denotar que el nivel y la jornada están asociados con este detalle, no se utilizará ningún campo de esta entidad porque no es relevante para el estudio → a utilizar ningún campo.

- **Malla estudiante:** Esta entidad guarda las mallas por las que puede pasar un estudiante. En el mejor de los casos, el estudiante puede tener un solo registro, pero por temas de cambios de carrera, homologaciones entre mallas y continuación de estudios en otras mallas siempre tienen varios registros, pero siempre hay un registro único de esta malla que está asociado en el periodo. No se van a utilizar ningún campo de esta tabla, tan solo se usará para la conexión con el estudiante y la matrícula en un periodo determinado → a utilizar ningún campo.

4.3.Recopilación de datos

- Fuente: Datos históricos ERP institucional del módulo académico
- Variables incluidas
    - Demográficas, Sector y parroquia de residencia, sexo, estado civil, fecha de nacimiento, edad
    - Académicas, Periodo, escuela, carrera, modalidad, jornada, colegio de procedencia.

Los datos para el estudio se obtienen de la base de datos empresarial que es la base del ERP institucional guardando información relacionada con los procesos académicos, financieros, administrativos, hay que destacar que la información es vasta por lo que se enfoca en extraer datos académicos de 5 años es decir desde el año 2020 al 2024 correspondientes a 10 periodos académicos, desde el periodo académico actual hacia atrás, en la data mostrada se han elegido variables que aporten al proceso de segmentación enriqueciéndolo con información útil para el estudio.

4.3.1. Proceso De Extracción

En esta primera etapa del proceso ETL se llevó a cabo utilizando la base de datos empresarial y un DBMS (Sistema Gestor de Base de Datos) para el acceso y la definición del conjunto de datos se detallan las actividades realizadas para la extracción

Acceso a la base de datos:

- Los datos históricos para el estudio se almacenan en una base de datos MariaDB en un servidor Alma Linux.

- Se utilizó una conexión con el servidor MariaDB desde Toad MySQL con las credenciales disponibles para el acceso a la estructura de datos.
- La estructura de la base de datos contiene datos relacionados que almacenan datos históricos sobre estudiantes, matriculas, periodos académicos carreras, modalidades

Identificación de los datos de estudio

- Se analizó las tablas y las relaciones expuestas anteriormente para lograr el conjunto de datos que se va a utilizar en el estudio
- Se consideraron tablas relacionadas con los datos demográficos como parroquia de residencia, edad.
- Se consideraron tablas relacionadas con datos académicos, estudiante, matriculas, periodos académicos, carreras, modalidades.

Consulta SQL para extraer los datos

- Se redactó una consulta utilizando el lenguaje SQL ver ANEXO 2 para extraer únicamente las columnas y registros relevantes de la estructura planteada en el entendimiento del negocio, la consulta tiene los campos solicitados y como se comentaba en el detalle de datos utilizar los campos de nivel y jornada al ser calculados en algún punto pueden llegar a tener valores como no determinado mientras se procesa la matrícula por eso se utiliza un left join para descartar si existe algún registro que se encuentre en esta forma,

- Además, se utilizó un filtro con el nivel de estudios por que actualmente la IES oferta postgrado tecnológico y el estudio se limita solo a la oferta tecnológica de siempre, además se incluye un ordenamiento inicial por estudiante

- Se definió una columna edad en el periodo de estudio, calculada en función de la fecha de nacimiento y la fecha de inicio del periodo académico de la matrícula.

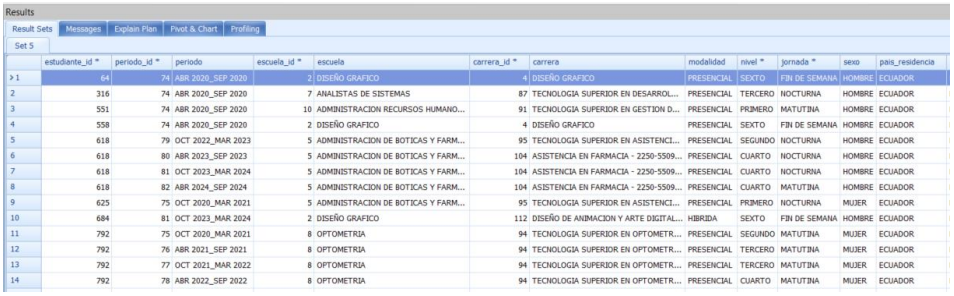
- Se filtró en la consulta para limitar el conjunto de datos del registro de matrícula de los últimos 5 años dado que en cada año existen dos periodos se obtienen datos desde el año 2020 hasta el año 2024.

Exportación de los datos

Se muestra en la Figura 3 *Exportación de datos* se muestra la exportación directamente desde Toad MySQL en formato CSV común para el análisis posterior.

Figura 3

Exportación de datos



Results

Result Sets | Messages | Explain Plan | Pivot & Chart | Profiling

Set 5

	estudiante_id	periodo_id	periodo	escuela_id	escuela	carrera_id	carrera	modalidad	nivel	jornada	sexo	pais_residencia
> 1	64	74	ABR 2020_SEP 2020	2	DISEÑO GRAFICO	4	DISEÑO GRAFICO	PRESENCIAL	SEXTO	FIN DE SEMANA	HOMBRE	ECUADOR
2	316	74	ABR 2020_SEP 2020	7	ANALISTAS DE SISTEMAS	87	TECNOLOGIA SUPERIOR EN DESARROL...	PRESENCIAL	TERCERO	NOCTURNA	HOMBRE	ECUADOR
3	551	74	ABR 2020_SEP 2020	10	ADMINISTRACION RECURSOS HUMANO...	91	TECNOLOGIA SUPERIOR EN GESTION D...	PRESENCIAL	PRIMERO	MATUTINA	HOMBRE	ECUADOR
4	558	74	ABR 2020_SEP 2020	2	DISEÑO GRAFICO	4	DISEÑO GRAFICO	PRESENCIAL	SEXTO	FIN DE SEMANA	HOMBRE	ECUADOR
5	618	79	OCT 2022_MAR 2023	5	ADMINISTRACION DE BOTICAS Y FARM...	95	TECNOLOGIA SUPERIOR EN ASISTENCI...	PRESENCIAL	SEGUNDO	NOCTURNA	HOMBRE	ECUADOR
6	618	80	ABR 2023_SEP 2023	5	ADMINISTRACION DE BOTICAS Y FARM...	104	ASISTENCIA EN FARMACIA - 2250-5509...	PRESENCIAL	CUARTO	NOCTURNA	HOMBRE	ECUADOR
7	618	81	OCT 2023_MAR 2024	5	ADMINISTRACION DE BOTICAS Y FARM...	104	ASISTENCIA EN FARMACIA - 2250-5509...	PRESENCIAL	CUARTO	NOCTURNA	HOMBRE	ECUADOR
8	618	82	ABR 2024_SEP 2024	5	ADMINISTRACION DE BOTICAS Y FARM...	104	ASISTENCIA EN FARMACIA - 2250-5509...	PRESENCIAL	CUARTO	MATUTINA	HOMBRE	ECUADOR
9	625	75	OCT 2020_MAR 2021	5	ADMINISTRACION DE BOTICAS Y FARM...	95	TECNOLOGIA SUPERIOR EN ASISTENCI...	PRESENCIAL	PRIMERO	NOCTURNA	MUJER	ECUADOR
10	684	81	OCT 2023_MAR 2024	2	DISEÑO GRAFICO	112	DISEÑO DE ANIMACION Y ARTE DIGITAL...	HIBRIDA	SEXTO	FIN DE SEMANA	HOMBRE	ECUADOR
11	792	75	OCT 2020_MAR 2021	8	OPTOMETRIA	94	TECNOLOGIA SUPERIOR EN OPTOMETR...	PRESENCIAL	SEGUNDO	MATUTINA	MUJER	ECUADOR
12	792	76	ABR 2021_SEP 2021	8	OPTOMETRIA	94	TECNOLOGIA SUPERIOR EN OPTOMETR...	PRESENCIAL	TERCERO	MATUTINA	MUJER	ECUADOR
13	792	77	OCT 2021_MAR 2022	8	OPTOMETRIA	94	TECNOLOGIA SUPERIOR EN OPTOMETR...	PRESENCIAL	TERCERO	MATUTINA	MUJER	ECUADOR
14	792	78	ABR 2022_SEP 2022	8	OPTOMETRIA	94	TECNOLOGIA SUPERIOR EN OPTOMETR...	PRESENCIAL	CUARTO	MATUTINA	MUJER	ECUADOR

Nota. *Fuente: desarrollo del autor

En la exportación como se muestra en la Figura 4 *Exportación codificación* se revisó la codificación de caracteres UTF-8 para evitar problemas con caracteres especiales y el separador de columnas “,”.

Figura 4

Exportación codificación

Nota. *Fuente: desarrollo del autor

Resultado

Un archivo CSV estructurado con los datos académicos y demográficos, está listo para el análisis en Python utilizando Google Colab, se detallan cada campo con su descripción.

- **estudiante_id**: Esta variable contiene el código único del estudiante en la base de datos. Es de tipo numérico (int64).
- **periodo_id**: Esta variable contiene el identificador del periodo académico o clave primaria del periodo, en el que el estudiante realizó su registro de matrícula, es de tipo numérico (int64).
- **periodo**: Esta variable categórica muestra el nombre o descripción del periodo académico, en el que el estudiante de la IES tuvo registro de matrícula es de tipo varchar.

- **escuela_id**: Esta variable numérica contiene el identificador o clave primaria de la escuela a la que pertenece un estudiante en algún periodo académico, es de tipo numérico (int64).

- **escuela**: Esta variable categórica contiene la descripción de la escuela que se asigna a un estudiante dentro de la matrícula en un periodo determinado es de tipo varchar.

- **carrera_id**: Variable que contiene el identificador único o clave primaria de la carrera del estudiante una carrera pertenece a una escuela su tipo de dato es numérico (int64).

- **carrera**: Esta variable categórica almacena el nombre o descripción de la carrera, representa el nombre de la carrera del estudiante en un registro de matrícula determinado es de tipo varchar.

- **modalidad**: Esta variable categórica almacena la modalidad de estudio (presencial, virtual, etc.) de la carrera su tipo de dato es varchar.

- **nivel**: Esta variable categórica contiene el nivel de estudios del registro en ese periodo académico es de tipo varchar.

- **jornada**: Esta variable categórica identifica la jornada de estudio (matutina, nocturna, etc.) del registro su tipo de dato es varchar.

- **sexo**: Esta variable categórica almacena el sexo del estudiante su tipo de dato es varchar.

- **pais_residencia**: Esta variable categórica almacena el país de residencia del estudiante, es de tipo varchar.

- **provincia residencia**: Variable categórica que identifica la provincia de residencia, su tipo de dato es varchar.
- **canton_residencia**: Esta variable categórica contiene el cantón de residencia del estudiante, su tipo de dato es varchar.
- **parroquia residencia**: Esta variable categórica contiene la parroquia de residencia del estudiante es de tipo varchar.
- **sector domicilio**: Esta variable categórica contiene el sector o zona de domicilio en la ciudad del estudiante es de tipo varchar.
- **fecha_nacimiento**: Esta variable contiene la fecha de nacimiento del estudiante, es de tipo date.
- **edad**: Esta variable discreta que contiene la edad del estudiante en años que tuvo con respecto al periodo académico de estudio del estudiante, su tipo de dato es numérico (int64).
- **colegio_id**: Esta variable numérica contiene el identificador único o clave primaria del colegio de procedencia del estudiante su tipo de dato es numérico (int64).
- **colegio**: Esta variable categórica contiene el nombre o descripción del colegio de procedencia del estudiante es de tipo varchar.
- **estado_civil:** Esta variable categórica contiene el estado civil del estudiante es de tipo varchar

4.4.Exploración de los datos

En esta etapa, se ha empleado las herramientas de Python con dos de sus librerías pandas y Ploty para el formateo y visualización de las estadísticas de las variables del estudio se detalla la exploración de los datos

- **estudiante_id**: El tipo de dato es entero, el recuento total es 48304, no hay valores nulos, hay 16074 valores distintos que es el número de estudiantes en el conjunto de datos de análisis.

- **periodo_id**: La columna tiene datos de tipo entero y contiene un total de 48304 registros, todos los registros completos no tienen valores nulos ni vacíos, existen 10 valores distintos que son el número total de periodos seleccionados para el estudio.

periodo: La columna es de tipo texto, contiene un total de 48304 registros, todos los registros completos no tienen valores nulos ni vacíos, existen 10 valores distintos como se muestra en la

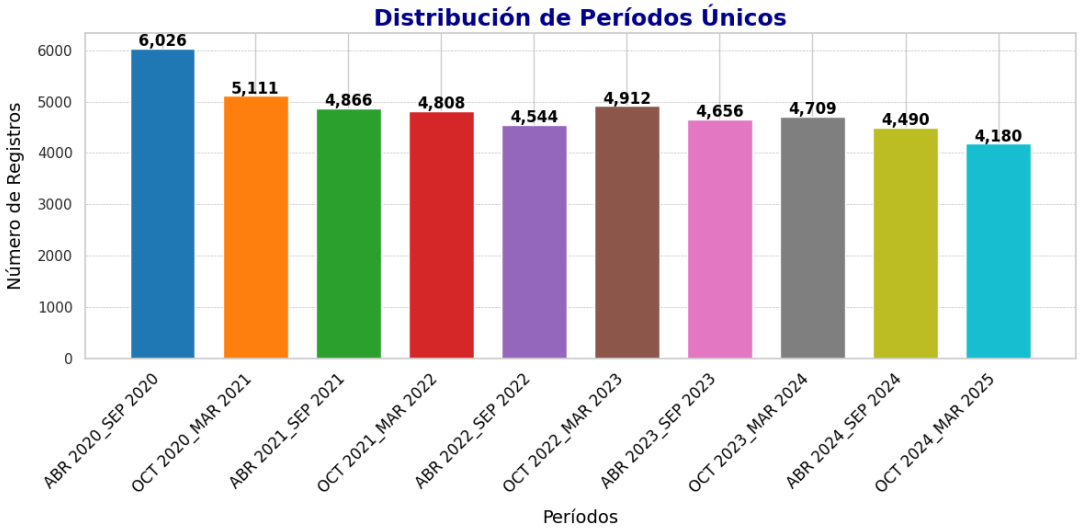
- Figura 5

  Periodos Académicos en el estudio ,

  la moda es ABR 2020_SEP 2020, que es la categoría más frecuente con 6026 apariciones, mientras que la categoría menos frecuente es “OCT 2024_MAR 2025” con una frecuencia de 4182.

Figura 5

Periodos Académicos en el estudio



*Nota. * Fuente: desarrollo del autor*

Análisis: Existen 10 periodos académicos como se muestra en la Tabla 1 *Estadísticos Periodo* en estudio que comienzan en el año 2020 y terminan en el año 2025, aunque hay una disminución gradual de registros de matrículas por periodo se puede apreciar que hay una ligera consistencia, hay factores post pandemia que deben ser considerados para esta baja:

La situación económica del país, la competencia agresiva por cuota de mercado,

El cambio de mallas de 6 niveles a 4 niveles que ahora es la tendencia en todos los institutos

como estrategia de captación y que al parecer no está funcionando, de ahí que se mantiene en observación el crecimiento fortaleciendo la retención y las estrategias de captación.

Tabla 1

Estadísticos Periodo

Estadística	Valor
Tipo de dato	object
Recuento	48304
Valores únicos	10
Valores nulos (NaN), Cadenas Vacías	0
Moda	ABR 2020_SEP 2020
Frecuencia máxima	6026
Categoría más frecuente	ABR 2020_SEP 2020
Frecuencia mínima	4182
Categoría menos frecuente	OCT 2024_MAR 2025

*Nota. * Fuente: desarrollo del autor*

- **escuela_id**: La columna tiene datos de tipo entero, contiene un total de 48304 registros sin valores nulos ni vacíos existen 15 valores distintos y únicos

- **escuela**: La columna tiene datos de tipo texto sin valores nulos ni vacíos, hay 15 valores, la moda es **DISEÑO GRAFICO**, que es la categoría más frecuente con 7487 apariciones, mientras que la categoría menos frecuente es **VENTAS** con una frecuencia de 93 como se muestra en la Tabla 2 *Estadísticos columna escuela*.

Tabla 2

Estadísticos columna escuela

Estadística	Valor
Tipo de dato	object
Valores únicos	15
Valores nulos (NaN)	0
Vacíos (cadenas vacías)	0
Moda	DISEÑO GRAFICO
Frecuencia máxima	7487
Categoría más frecuente	DISEÑO GRAFICO
Frecuencia mínima	93
Categoría menos frecuente	VENTAS

*Nota. * Fuente: desarrollo del autor*

Análisis: Para propósito de reportes existe esta categoría escuela que es la categorización general existiendo 15, según el estadístico tenemos más frecuentemente registros de la escuela de diseño gráfico que es una de las escuelas más importantes de la IES y a su vez un proyecto con la escuela de **VENTAS** que estaba dirigido para profesionales que necesitaban obtener un título en solo dos semestres.

- **carrera_id**: La columna tiene datos de tipo entero, contiene un total de 48304 registros, sin valores nulos ni vacíos existen 47 valores distintos y únicos
- **carrera**: La columna tiene datos de tipo texto no contiene valores nulos ni vacíos, existen 47 valores únicos, la moda es "TECNOLOGIA SUPERIOR EN ADMINISTRACION FINANCIERA - 2250-550412B-P-01 – NVHRT", que es la categoría más frecuente con 5220 apariciones, mientras que la categoría menos frecuente es "GESTION ESTRATEGICA DE REDES SOCIALES - 2250-55041301-L-1701" con una frecuencia de 2 como se muestra en la Tabla 3 *Estadísticos columna carrera*.

Tabla 3

Estadísticos columna carrera

Estadística	Valor
Tipo de dato	object
Valores nulos (NaN)	0
Moda	TECNOLOGIA SUPERIOR EN ADMINISTRACION FINANCIERA - 2250-550412B-P-01 - NVHRT
Frecuencia máxima	5220
Categoría más frecuente	TECNOLOGIA SUPERIOR EN ADMINISTRACION FINANCIERA - 2250-550412B-P-01 - NVHRT
Frecuencia mínima	2

Categoría menos frecuente	GESTION ESTRATEGICA DE REDES SOCIALES - 2250-550414301-L-1701

*Nota. * Fuente: desarrollo del autor*

Análisis: esta columna representa las carreras que cuenta el instituto y muestra que administración financiera es la carrera más importante debido a que es una de las más antiguas en el instituto y la que menos se aparece es redes sociales que fue un proyecto de carrera en pandemia que no tuvo el éxito deseado y se terminó por cerrar.

- **modalidad**: Esta columna es de tipo texto, contiene un total de 48304 registros, no tiene valores nulos ni vacíos, la moda es PRESENCIAL, que es la categoría más frecuente con 46248 apariciones.

  mientras que la categoría menos frecuente es EN LINEA con 940 apariciones.

- La modalidad con más registros es la **PRESENCIAL** con 46248, seguida de la modalidad **HÍBRIDA** con 1116, mientras que la modalidad con menos registros es **EN LÍNEA** con 940

Análisis: contamos con tres modalidades en la IES, presencial, hibrida y en línea como se muestra en la

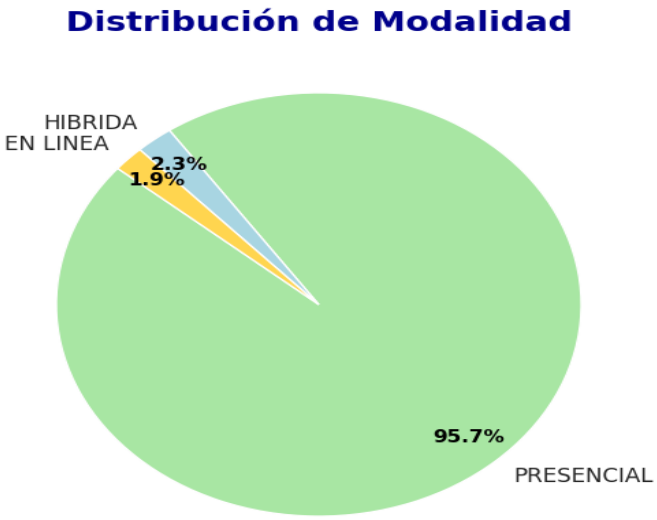
Figura 6

Distribución de modalidades total de acuerdo a los datos mostrados la modalidad Presencial es sin duda la predominante seguida por las carreras Hibridas y las carreras

En Línea que se crearon en pandemia para dar alternativas virtuales a los estudiantes y que al momento está en fase de promoción.

Figura 6

Distribución de modalidades total



Modalidad	Registros	Porcentaje
PRESENCIAL	46248	95.7%
HIBRIDA	1116	2.3%
EN LINEA	940	1.9%

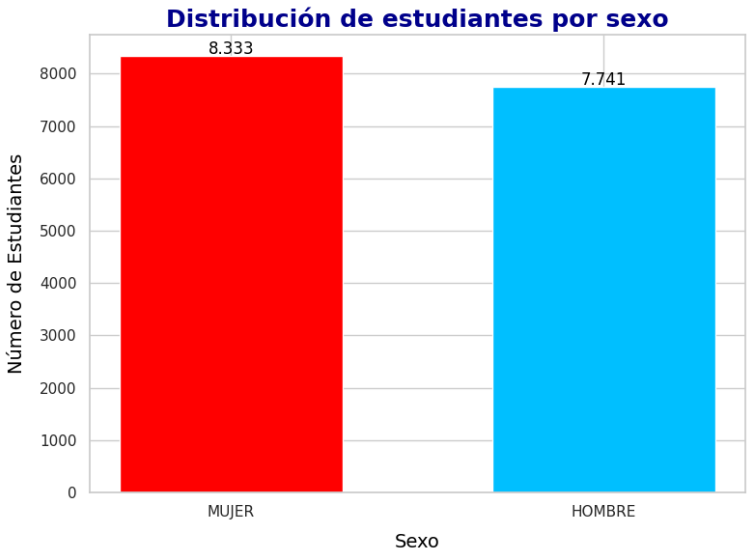
*Nota. * Fuente: desarrollo del autor*

- **nivel:** dependiendo de la malla se disponen de carreras de 2, 4 y 6 niveles
- **jornada**: se dispone de tres jornadas matutina nocturna y fin de semana que son utilizadas para propósitos de arrastre y nuevas carreras (recientes)

sexo: La columna de tipo texto, no tiene valores nulos ni vacíos, se presenta el gráfico de la distribución por sexo como se muestra en la

Figura 7

Distribución de estudiantes por sexo



*Nota. * Fuente: desarrollo del autor*

Análisis: El 51.85% de los estudiantes de la IES son mujeres y el 48.15% son hombres esto evidencia una población estudiantil relativamente balanceada.

- **pais_residencia**: La columna tiene datos de tipo texto, registros, todos completos, sin valores nulos ni vacíos, hay 3 valores únicos, la moda es ECUADOR, que es la categoría más frecuente con 48298 apariciones, mientras que la categoría menos frecuente es VENEZUELA con una frecuencia de 3., considerando que el país es de residencia en la etapa de depuración se tienen que revisar los datos

- **provincia residencia**: La columna tiene datos de tipo texto, registros, todos completos, sin valores nulos ni vacíos, hay 21 valores únicos, la moda es PICHINCHA, que es la categoría más frecuente con 48117 apariciones, mientras que la categoría menos frecuente es SANTA ELENA con una frecuencia de 1.

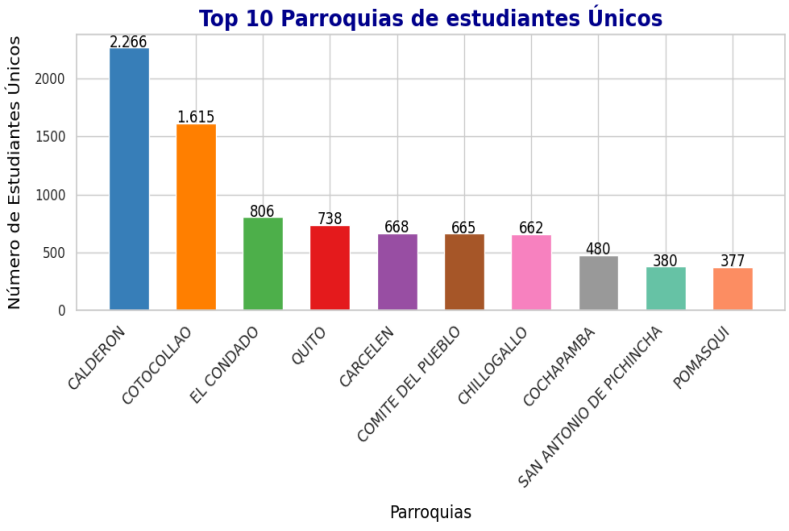
- **canton_residencia**: La columna tiene datos de tipo texto, no tiene valores nulos ni vacíos, hay 41 valores únicos, la moda es QUITO, que es la categoría más frecuente con 46654 apariciones, mientras que la categoría menos frecuente es LA LIBERTAD con una frecuencia de 1.

- **parroquia residencia**: La columna es de tipo texto, no tiene sin valores nulos ni vacíos, hay 151 valores únicos, la moda es **CALDERON**, que es la categoría más frecuente con 7142 apariciones seguida de COTOCOLLAO con 1615 apariciones.

Análisis: de las parroquias de residencia de los estudiantes, se puede evidenciar que la primera y segunda parroquia con más estudiantes es Calderón y Cotocollao relativamente, se deduce porque está ubicado como única matriz en el sector de la prensa y Logroño y es el Instituto más grande más cercano al norte como se indica en la Figura 8 *Top de parroquias de residencia de estudiantes*.

Figura 8

Top de parroquias de residencia de estudiantes



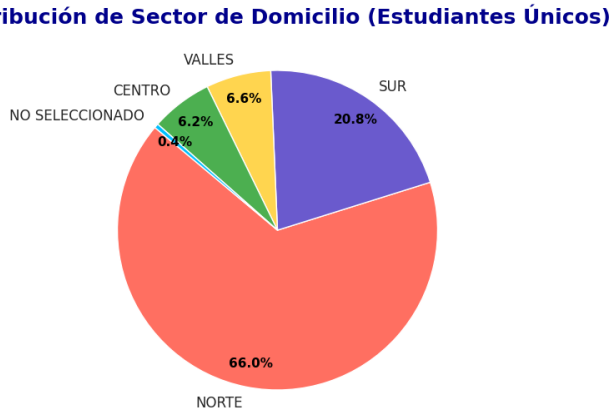
*Nota. * Fuente: desarrollo del autor*

- **sector_domicilio**: La columna de tipo texto, no tiene valores nulos ni vacíos, hay 5 valores únicos, la moda es NORTE, que es la categoría más frecuente con 31528 apariciones, mientras que la categoría menos frecuente es NO SELECCIONADO con una frecuencia de 183.

Análisis: Como se puede evidenciar en la **Figura 9 Distribución de sector de domicilio** por estudiante Hay un predominio del sector norte con el 66% lo que representa una mayoría abrumadora comparada con los demás sectores, esto refuerza las conclusiones con el top parroquia en donde Calderón y Cotocollao eran las parroquias con más concentración, hay que considerar que antes de la pandemia el sector sur era de donde venían más estudiantes pero por la competencia instalada por este sector y por la seguridad del trayecto hacia el norte ha bajado al 20% el sector de residencia, hay poca representatividad para los valles y el centro.

Figura 9

Distribución de sector de domicilio por estudiante



Sector	Registros	Porcentaje
NORTE	10602	66.0%
SUR	3348	20.8%
VALLES	1055	6.6%
CENTRO	999	6.2%
NO SELECCIONADO	70	0.4%

*Nota. * Fuente: desarrollo del autor*

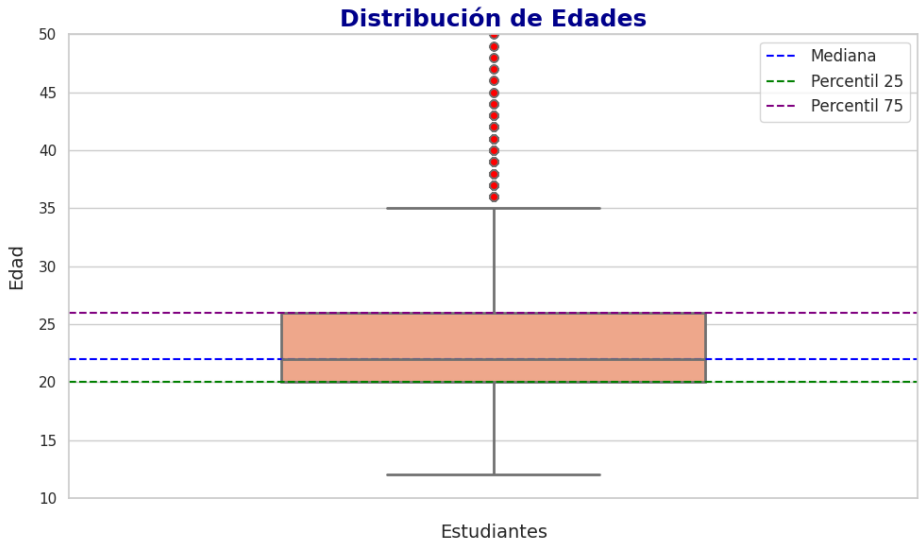
- **fecha_nacimiento**: La columna tiene datos de tipo fecha, contiene un total de 48304 registros, todos completos, sin valores nulos ni vacíos, hay 6455 valores distintos y únicos, la fecha más antigua es 1900-01-01, la más reciente es 2022-10-12, y el rango entre las fechas es de 122 años, hay que realizar un proceso de limpieza y depuración de este campo.

- **edad**: La columna tiene datos de tipo entero, con un total de 48304 registros, todos completos, sin valores nulos ni vacíos, contiene 56 valores distintos y únicos, el valor mínimo es -2 y el máximo es 120, el promedio es 23.64, en este apartado hay que considerar que aplicar depuración de datos.

Cabe indicar que existen outliers que se puede visualizar en el gráfico de cajas y bigotes en la Figura 10 *Distribución de Edades*, valores hay que revisar y depurar ciertos registros, la mayoría de estudiantes se encuentran entre 20 y 30 años, la mediana es aproximadamente 25 años.

Figura 10

Distribución de Edades



*Nota. * Fuente: desarrollo del autor*

Análisis: la media se concentra alrededor de los 25 años, entre los 22 y los 28 años se encuentra concentrado el 50% de los estudiantes, hay rangos entre los 15 años que se tendrían que revisar, pero valores alrededor de los 35 indican que hay estudiantes dispuestos a estudiar a pesar de la diferencia de edad del promedio, existen outliers que deben ser revisados.

- **colegio_id**: La columna tiene datos de tipo entero, todos completos, sin valores nulos ni vacíos, tiene 1461 valores distintos y únicos

- **colegio**: La columna tiene datos de tipo texto, contiene, todos completos, sin valores nulos ni vacíos, hay 1446 valores únicos, la moda es NO ASIGNADO, que es la categoría más frecuente con 1644 apariciones, mientras que la categoría menos frecuente es POLICIA NACIONAL con una frecuencia de 1.

Análisis: hay una cantidad considerable de registros no asignados por lo que se debería considerar eliminar esta columna

Debido a que no aportaría mucho esto se debe a que hay una cantidad importante de registros que ahora la realiza el personal de admisión y al no ser obligatorio no lo ingresan como se puede evidenciar en la

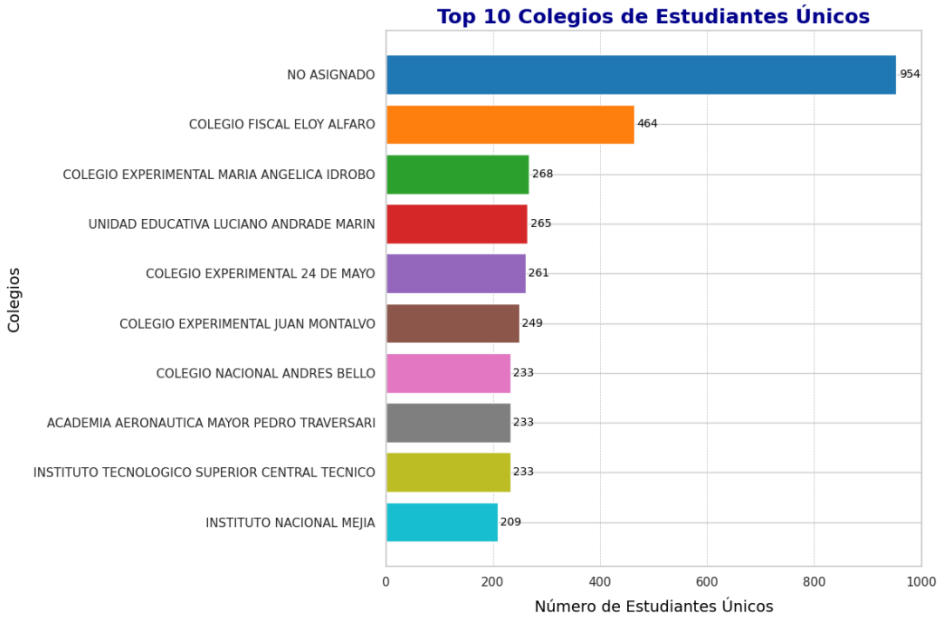
Figura 11

Top de colegios de estudiantes,

Sin embargo, se destaca que el colegio Eloy Alfaro que esta por el sector es el que más estudiantes contribuye los demás colegios son variables.

Figura 11

Top de colegios de estudiantes



*Nota. * Fuente: desarrollo del autor*

- **estado_civil:** La columna tiene datos de tipo texto, contiene un total de 48302 registros, todos completos, sin valores nulos ni vacíos, hay 5 valores únicos, la moda es SOLTER@, que es la categoría más frecuente con 43781 apariciones, mientras que la categoría menos frecuente es VIUD@ con una frecuencia de 46.

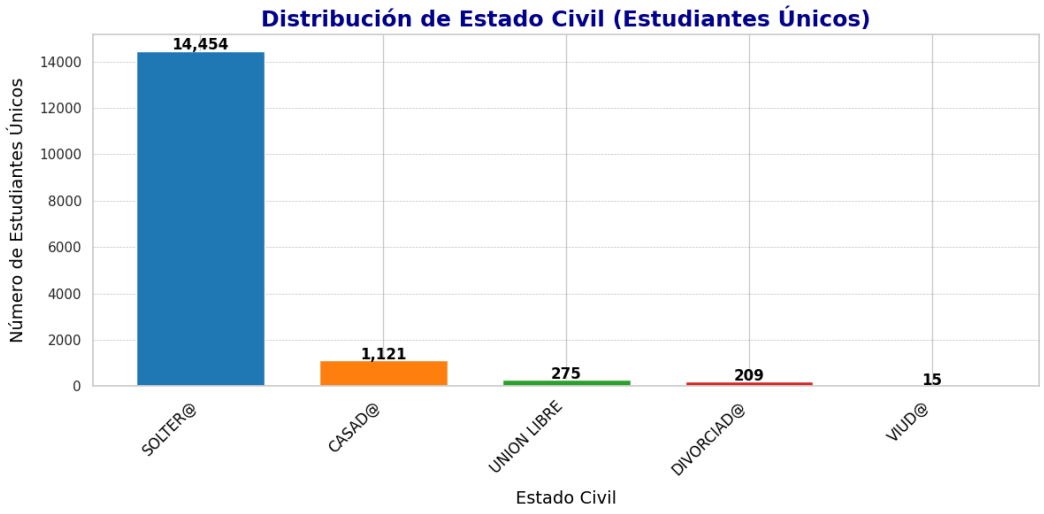
Análisis: Cómo se puede evidenciar en la

Figura 12

Distribución de estado civil de estudiantes la mayoría son solteros es grupo dominante en la distribución.

Figura 12

Distribución de estado civil de estudiantes



*Nota. * Fuente: desarrollo del autor*

4.5.Verificación de la calidad de los datos

De la revisión inicial de la data set de estudiantes se concluye que no existen mayores problemas, existen datos que deben ser tratados en el siguiente capítulo, podemos afirmar que los datos son de calidad y se puede utilizarlos para cumplir el propósito del estudio.

Preparación de la data

4.5.1. Manejo de valores nulos

En este punto no es necesario la imputación o tratamiento de valores nulos ya que no hay datos faltantes como se puede evidenciar en la Figura 13 *Datos Nulos*.

Figura 13

Datos Nulos

Manejo de Valores Nulos en el Dataset

Columnas	Valores Nulos Antes	Valores Nulos Después
estudiante_id	0	0
periodo_id	0	0
periodo	0	0
escuela_id	0	0
escuela	0	0
carrera_id	0	0
carrera	0	0
modalidad	0	0
nivel	0	0
jornada	0	0
sexo	0	0
pais_residencia	0	0
provincia_residencia	0	0
canton_residencia	0	0
parroquia_residencia	0	0
sector_domicilio	0	0
fecha_nacimiento	0	0
edad	0	0
colegio_id	0	0
colegio	0	0
estado_civil	0	0

*Nota. * Fuente: desarrollo del autor*

4.5.2. Corrección de datos inconsistente

Homogenizar textos

El campo estado_civil tiene caracteres como el carácter “@” que se utilizan en la IES para garantizar neutralidad en reportes, pero por generalización se va a homogenizarlos como se evidencia en la

Figura 14

Homogenizar estado civil

Figura 14

Homogenizar estado civil

Visualización de 'estado_civil': Antes y Después

Valores Antes de Normalización	Valores Después de Normalización
SOLTER@	Soltero
DIVORCIAD@	Divorciado
CASAD@	Casado
UNION LIBRE	Union Libre
VIUD@	Viudo

*Nota. * Fuente: desarrollo del autor*

Para mantener la consistencia, claridad y uniformidad para las variables categóricas como escuela, nivel, jornada, sector_domicilio, modalidad, sexo, pais_residencia, provincia_residencia, canton_residencia, parroquia_residencia, carrera, colegio, estado_civil, periodo se transformaron a minúsculas como se muestra en la Figura 15

Homogenizar variables categóricas

Figura 15

Homogenizar variables categóricas

Homogenización: Conversión a Minúsculas

Columna	Valores Antes	Valores Después
sexo	HOMBRE	hombre
sexo	MUJER	mujer
modalidad	EN LINEA	en linea
sector_domicilio	CENTRO	centro
sector_domicilio	NO SELECCIONADO	no seleccionado
nivel	QUINTO	quinto
escuela	GESTIÓN EMPRESARIAL	gestión empresarial
escuela	MARKETING INTERNO Y EXTERNO	marketing interno y externo
escuela	ADMINISTRACION PARA EL DESARROLLO DEL TALENTO INFANTIL	administracion para el desarrollo del talento infantil
escuela	ENFERMERÍA	enfermería
escuela	ADMINISTRACION TURISTICA HOTELERA	administracion turistica hotelera
escuela	ADMINISTRACION INDUSTRIAL Y DE LA PRODUCCION	administracion industrial y de la produccion
escuela	VENTAS	ventas
escuela	SEGURIDAD Y PREVENCIÓN DE RIESGOS LABORALES	seguridad y prevención de riesgos laborales

*Nota. * Fuente: desarrollo del autor*

4.5.3. Manejo de Valores atípicos

Variable fecha de nacimiento y edad en el periodo académico

Se listan los valores encontrados como outliers, basados en un programa que evalúa la edad entre el rango menores o igual que 15 y mayores o iguales que 80. Y se listan 17 casos

Los registros con problemas son 17 de 48302 registros totales y representan el 0.035% de los datos de edad se procede a eliminarlos, estos registros con observaciones se pueden ver en la Figura 16

Variable edad atípico

Figura 16

Variable edad atípico

Valores Atípicos en Edad

Índice	ID Estudiante	Fecha de Nacimiento	Edad
1	600015014	2014-07-21	9
2	600015014	2014-07-21	9
3	600048770	1900-01-01	120
4	600048812	1900-01-01	120
5	600053802	2022-10-12	-2
6	600053802	2022-10-12	-1
7	600053802	2022-10-12	-1
8	600053802	2022-10-12	0
9	600053802	2022-10-12	0
10	600053802	2022-10-12	0
11	600053802	2022-10-12	1
12	600053802	2022-10-12	1
13	600053802	2022-10-12	-2
14	600059866	2011-01-27	12
15	600059866	2011-01-27	12
16	600059866	2011-01-27	13
17	600059866	2011-01-27	13

*Nota. * Fuente: desarrollo del autor*

Columna colegio

De 48285 registros del dataset el campo colegio tiene 1643 registros con la categoría “no asignado” que representa el 3.40% de la información, muchos de estos campos pertenecen al mismo estudiante siendo de esta forma vamos a dejar los datos faltantes como la categoría “no asignado” para el estudio, como se puede evidenciar *en la* Figura 17 *Colegio Análisis.*

Figura 17

Colegio Análisis

Colegio	Cantidad de Estudiantes
no asignado	1643
colegio fiscal eloy alfaro	1338
colegio experimental maria angelica idrobo	834
colegio experimental juan montalvo	822
unidad educativa luciano andrade marin	800
colegio experimental 24 de mayo	779
instituto tecnologico superior central tecnico	723
colegio nacional andres bello	696
instituto nacional mejia	656

*Nota. * Fuente: desarrollo del autor*

4.5.4. Creación, eliminación de variables

Los atributos identificados con _id son claves primarias de las tablas no aportan con información relevante al estudio, por esta razón serán eliminados como se muestra en la Figura 18

Variables de secuencia

Figura 18

Variables de secuencia

Columnas con '_id' en df_cleaned

Columnas _id	Tipo de Dato
estudiante_id	int64
periodo_id	int64
escuela_id	int64
carrera_id	int64
colegio_id	int64

*Nota. * Fuente: desarrollo del autor*

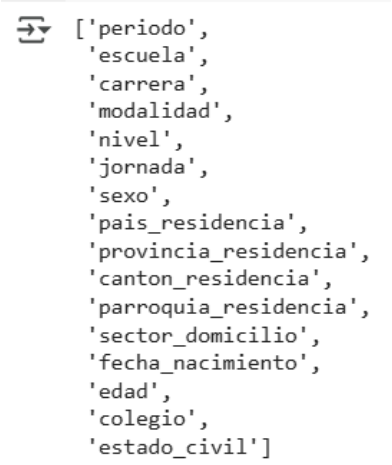
4.5.5. Data final

De esta forma el dataset limpio tiene 16 columnas y 48285 registros como se muestra en la Figura 19

Lista columnas limpias.

Figura 19

Lista columnas limpias



```
['periodo',
 'escuela',
 'carrera',
 'modalidad',
 'nivel',
 'jornada',
 'sexo',
 'pais_residencia',
 'provincia_residencia',
 'canton_residencia',
 'parroquia_residencia',
 'sector_domicilio',
 'fecha_nacimiento',
 'edad',
 'colegio',
 'estado_civil']
```

*Nota. * Fuente: desarrollo del autor*

4.6.Modelamiento

4.6.1. K-Means

Desarrollaremos en este apartado el algoritmo K-Means un algoritmo para identificar patrones y segmentar conjuntos de estudiantes en grupos homogéneos, este método nos permitirá clasificarlos en diferentes categorías en base a las variables seleccionadas en el capítulo anterior.

Para determinar el número óptimo de clusters (k) se va a emplear el método del codo como se definió en las métricas del marco metodológico, este evalúa la variabilidad dentro de los grupos y ayuda a identificar el punto en el que crear más clusters ya no genera una mejora significativa en la segmentación.

Selección de Variables:

De las variables preprocesadas anteriormente hay que considerar que las variables se encuentren en un formato adecuado ya que K-Means trabaja solo con variables numéricas, porque

se utiliza la distancia euclidiana, en la Tabla 4 *Selección de Variables para K-Means* se detalla la selección y las transformaciones realizadas a las variables

Tabla 4

Selección de Variables para K-Means

Antes	Tratamiento Aplicado	Después
periodo, escuela, carrera, pais_residencia, provincia_residencia, canton_residencia, colegio, fecha_nacimiento	No consideradas por irrelevancia en el clustering con K- Means	No consideradas
modalidad (categórica: presencial, híbrida, virtual)	One-Hot Encoding (columnas separadas para cada categoría)	Columna por categoría entre 1 y 0
jornada (categórica: matutina, nocturna, fin de semana)	One-Hot Encoding	Columna por categoría entre 1 y 0
parroquia_residencia (categórica con muchos valores)	One-Hot Encoding	Columna por categoría entre 1 y 0
sector_domicilio (categórica: norte, sur, valles)	One-Hot Encoding	Columna por categoría entre 1 y 0
estado civil (categórica: soltero, casado, unión libre, etc.)	One-Hot Encoding	Columna por categoría entre 1 y 0
nivel (ordinal: primero a décimo)	Codificación ordinal y escalado con StandardScaler	Valores transformados con media 0 y desviación estándar 1
sexo (categórica: hombre/mujer)	Mapeo binario (hombre=0, mujer=1)	Valores 1 y 0
edad (numérica)	Escalado con StandardScaler	Valores transformados con media 0 y desviación estándar 1

*Nota. * Fuente: desarrollo del autor*

Método del codo

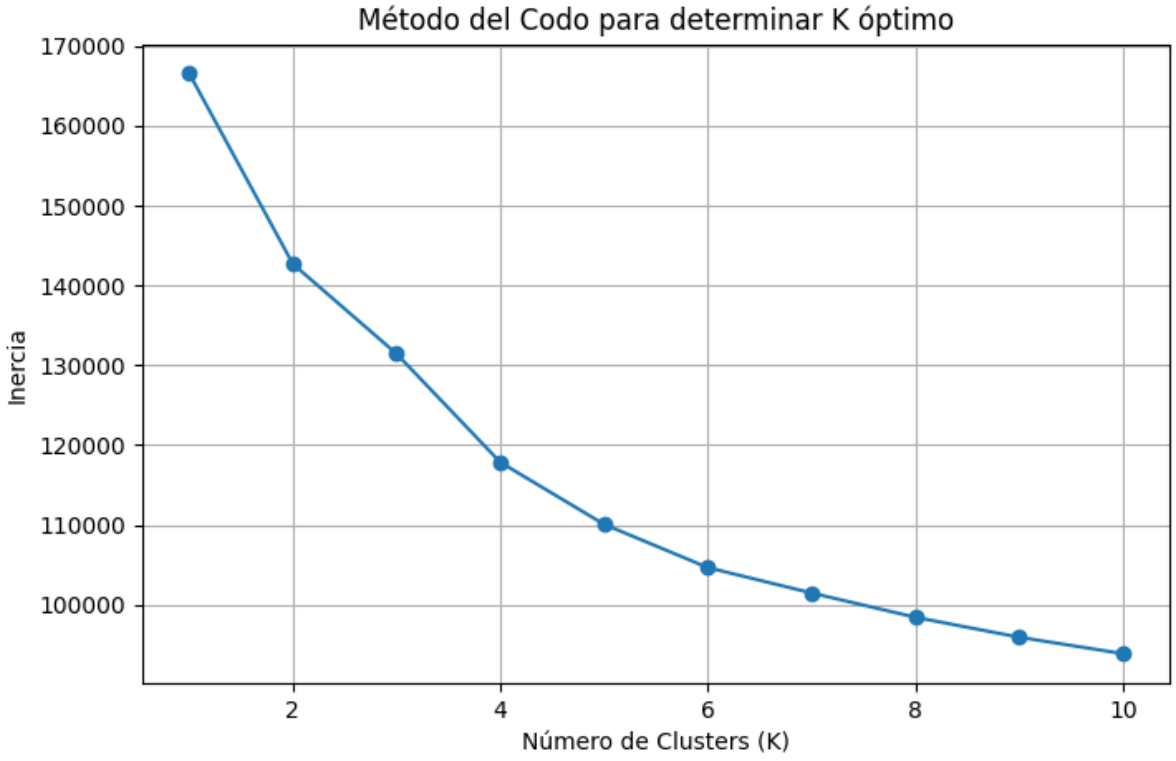
EL método del codo es una técnica que se utiliza en algoritmo como K-Means, para determinar el número óptimo de clusters, basado en la relación entre el número de clusters y la suma de los errores cuadráticos dentro del grupo (IBM, 2024), en este punto se realizó un programa

en Python que analiza un rango de valores entre 1 y 10 en un bucle iterativo de esta forma se define la cantidad ideal de los clusters sin necesidad de realizar una asignación arbitraria, en el eje x se grafican el número de clusters y el eje representa el cálculo de la inercia, esta mide la variabilidad de los clusters los valores más bajos indican grupos o clusters más compactos.

Esta gráfica Figura 20 *Método del codo k óptimo K-Means* nos ayuda a identificar el punto donde la reducción de la inercia comienza a estabilizarse forman un codo que sugiere el número óptimo de clusters

Figura 20

Método del codo k óptimo K-Means



*Nota. * Fuente: desarrollo del autor*

- Aquí podemos observar que la disminución de la inercia es pronunciada al inicio de 3 o 4 clusters, luego la reducción comienza a desacelerarse

- El codo más marcado esta entre 3 o 4 lo que sugiere que este valor es óptimo para la segmentación de la data de estudiantes

- Un número de clústeres inferior a 3 podría ser insuficiente para captura la variabilidad de los datos, ya que la inercia disminuye significativamente.

- Un número mayor a 4 puede no ser necesario ya que la reducción de la inercia no es muy significativa, indica que los grupos adicionales que no aportan una mejora a la segmentación

- Se puede concluir que los grupos de la IES se agrupan de manera más eficiente en 4 segmentos.

Modelo K-Means

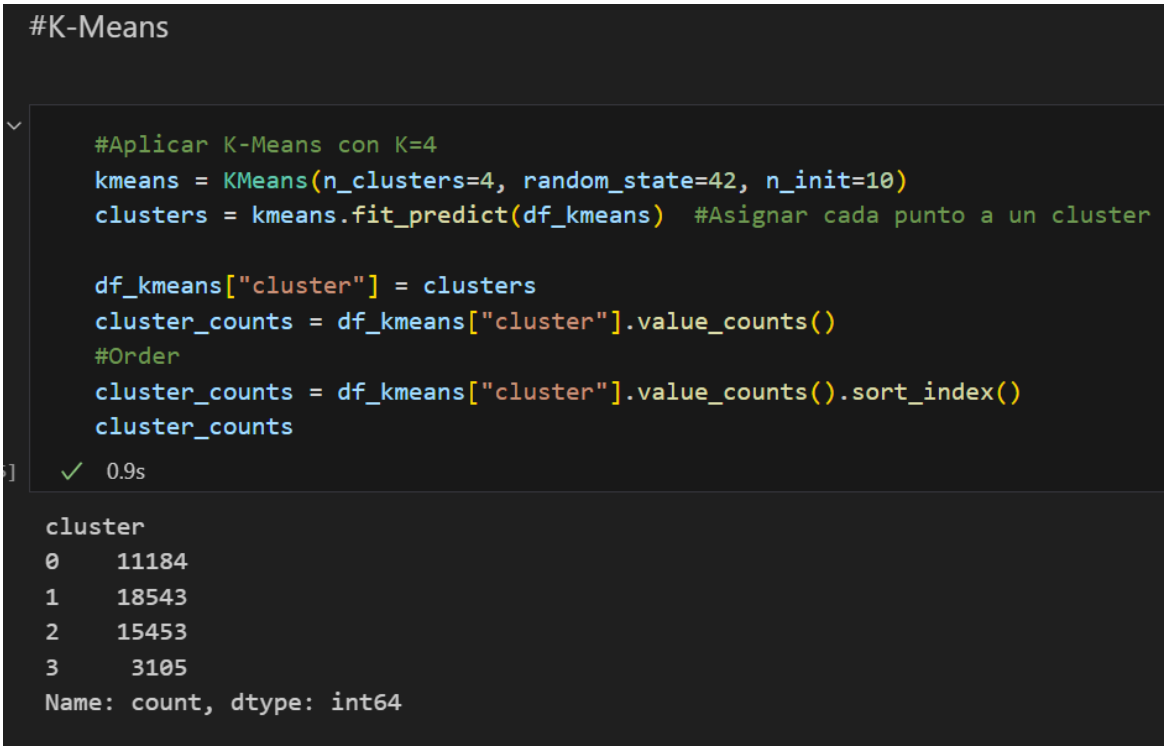
Se ejecutó el algoritmo de K-Means con los 4 clústeres obtenidos, es decir segmentar 4 grupos según similitudes en sus características, asignando cada estudiante a un cluster, agregando los clústeres al dataset y contando la cantidad como se puede apreciar en la

Figura 21

Modelo K-Means

Figura 21

Modelo K-Means



*Nota. * Fuente: desarrollo del autor, el código fuente completo se encuentra en el ANEXO 1.*

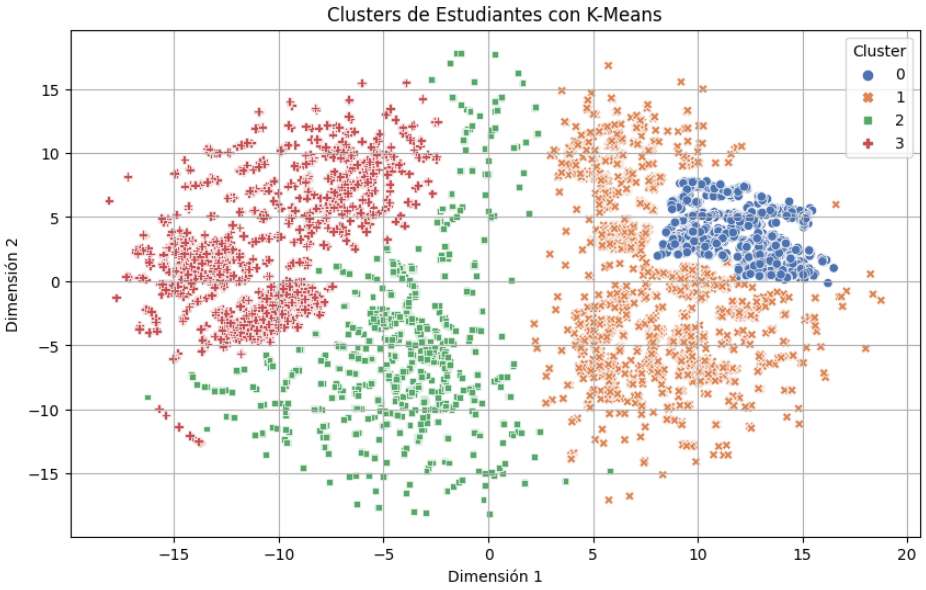
Se evidencia gráficamente en la

Figura 22

Gráfico de Clústeres K-Means los 4 clústeres que representan un grupo de estudiantes con características similares, el cluster con menos estudiantes es el 3 con 3105 estudiantes mientras que el más grande es el cluster 1 con 18543 estudiantes.

Figura 22

Gráfico de Clústeres K-Means



*Nota. * Fuente: desarrollo del autor*

Del grafico de los clusters de K-Means se detalla a continuación en las conclusiones por cada cluster.

CLÚSTER 0:

De color rojo, de forma cruces ubicado en la parte izquierda del gráfico, grupo con 11184 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 22.21 años con un rango de 17 a 34 años, el sexo más frecuente es mujer, el nivel académico más frecuente es primero, la parroquia de residencia más común es Chillogallo, la mayoría reside en el sector sur, el estado civil predominante es soltero.

CLÚSTER 1:

De color verde de forma cuadrados: ubicado en la parte media del gráfico, grupo con 18543 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más

común es matutina, la edad promedio es de 21.66 años con un rango de 16 a 34 años, el sexo más frecuente es mujer, el nivel académico más frecuente es primero, la parroquia de residencia más común es calderón, la mayoría reside en el sector norte, el estado civil predominante es soltero.

CLÚSTER 2:

De color naranja, de forma cruces en x ubicado en la parte derecha centro del gráfico, grupo con 15453 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es nocturna, la edad promedio es de 24.22 años con un rango de 16 a 34 años, el sexo más frecuente es mujer, el nivel académico más frecuente es primero, la parroquia de residencia más común es calderón, la mayoría reside en el sector norte, el estado civil predominante es soltero.

CLÚSTER 3:

De color azul, de forma círculos ubicado en la parte derecha del gráfico, grupo con 3,105 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es nocturna, la edad promedio es de 37.85 años con un rango de 30 a 68 años, el sexo más frecuente es mujer, el nivel académico más frecuente es primero, la parroquia de residencia más común es calderón, la mayoría reside en el sector norte, el estado civil predominante es casado.

Estadísticos por Cluster

Se detallan a continuación los estadísticos con variables importantes dentro de los clústeres encontrados por K-Means

Variable Sexo:

El gráfico muestra la distribución de género en los cuatro clústeres de K-Means, en todos los clústeres, las mujeres son mayoría, destacando el clúster 2 con 9706 mujeres y 8837 hombres, el clúster 3 presenta la mayor diferencia de género (6,055 mujeres y 5129 hombres), mientras que en los clústeres 0 y 1 la diferencia es menor como se puede evidenciar en la Figura 23 *Distribución por sexo por clusters K-Means*

Figura 23

Distribución por sexo por clusters K-Means

*Nota. * Fuente: desarrollo del autor*

Variable Modalidad:

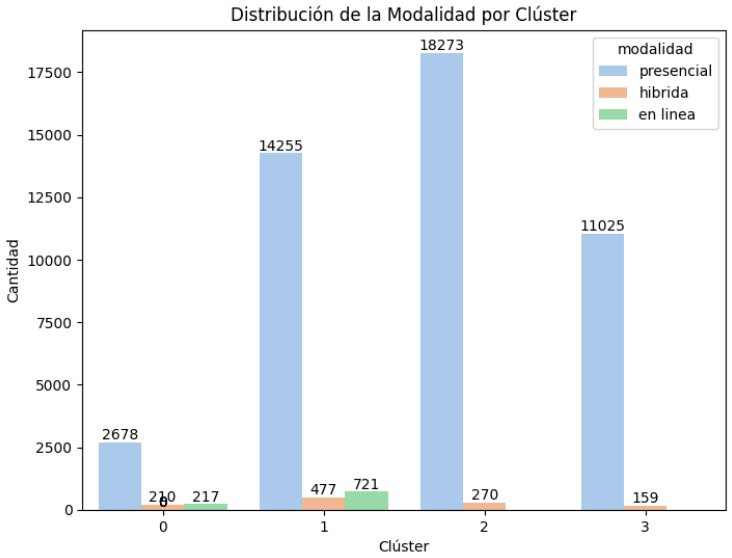
Se muestra la distribución de la modalidad de estudio en los cuatro clústeres de K-Means, la modalidad presencial predomina en todos los clústeres, especialmente en el clúster 2 con 18273 estudiantes, la modalidad híbrida y en línea tienen una presencia mucho menor en todos los grupos, siendo más destacada en el clúster 1 con 477 estudiantes híbridos y 721 en línea como se puede

apreciar en la Figura 24

Distribución por Modalidad por Clusters K-Means

Figura 24

Distribución por Modalidad por Clusters K-Means



*Nota. * Fuente: desarrollo del autor*

Variable Jornada:

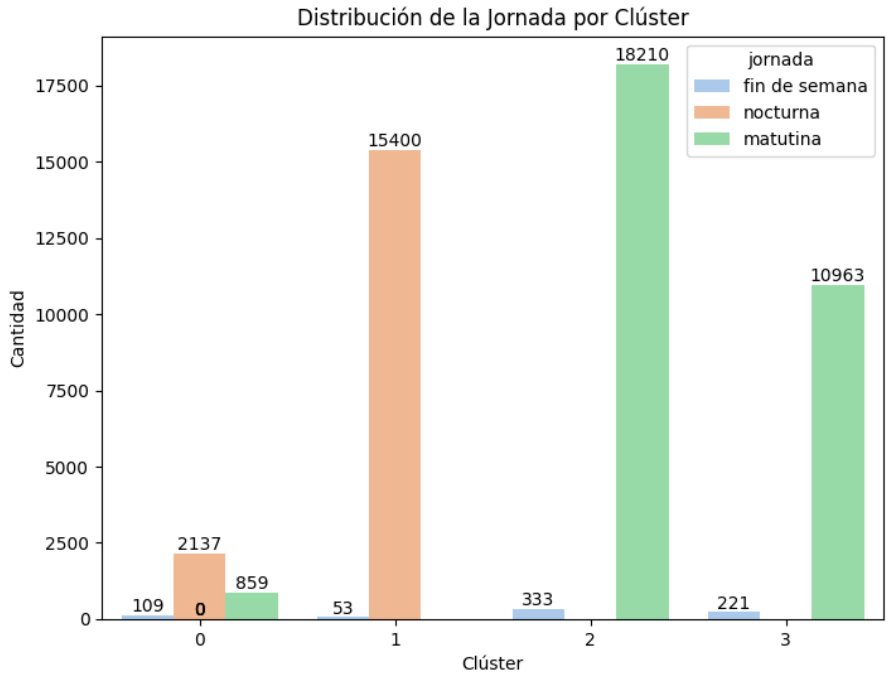
Se muestra la distribución de la jornada de estudio en los cuatro clústeres de K-Means, la jornada matutina predomina en los clústeres 2 y 3, con 18210 y 10963 estudiantes, respectivamente, la jornada nocturna destaca en el clúster 1, con 15400 estudiantes, la jornada de fin de semana tiene una menor representación, solo es mejor en el clúster 0 con 2137 estudiantes como se evidencia en la

Figura 25

Distribución por jornada por Clusters K-Means

Figura 25

Distribución por jornada por Clusters K-Means



*Nota. * Fuente: desarrollo del autor*

Variable edad y nivel:

Se observa en el gráfico de dispersión

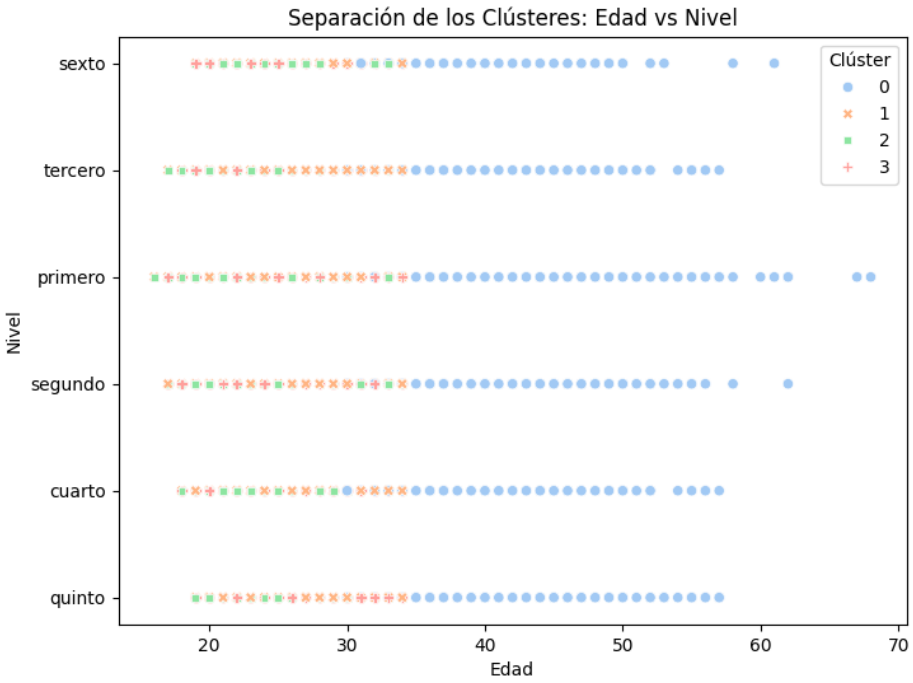
Figura 26

Separación de los clústeres edad vs nivel K-Means la relación entre la edad y el nivel académico para los cuatro clústeres identificados mediante K-Means, se observa que los estudiantes de los niveles más bajos (de primero a tercero) presentan una mayor dispersión en las edades, especialmente en los clústeres 1 y 2, en los niveles superiores (de cuarto a sexto),

Los estudiantes se agrupan principalmente en los clústeres 0 y 3 con edades que van entre los 20 y 60 años, la distribución por clúster refleja la variabilidad de la edad en combinación con el nivel.

Figura 26

Separación de los clústeres edad vs nivel K-Means



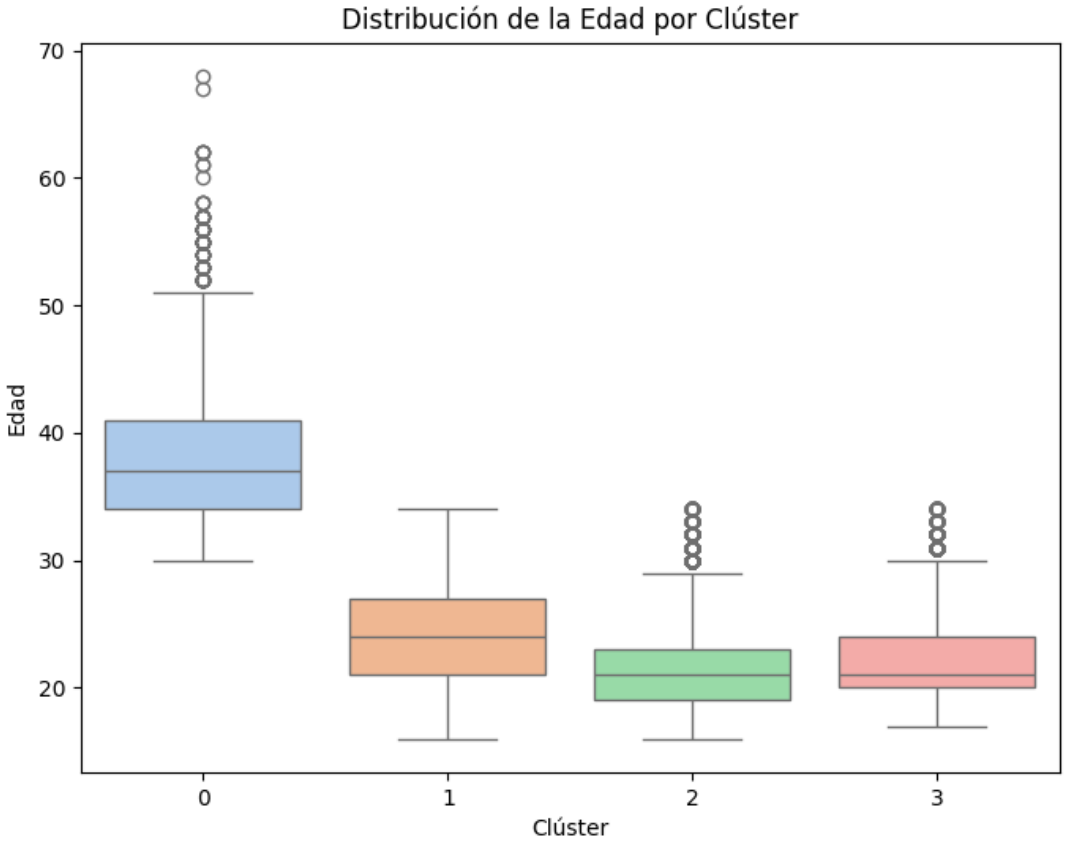
*Nota. * Fuente: desarrollo del autor*

Variable Edad:

El gráfico de cajas y bigotes (bloxplot) Figura 27 *Distribución de la Edad por Clusters* muestra la distribución de la edad en cada uno de los cuatro clústeres, el clúster 0 presenta la mayor dispersión, con edades que van desde los 30 hasta los 68 años, aunque se aprecian edades por encima de los 50 años, los clústeres 1, 2 y 3 tienen distribuciones menos variadas (compactas), que van principalmente entre los 17 y 34 años, el clúster 1 presenta una media de 25 años, mientras que los clústeres 2 y 3 tienen medianas alrededor de los 20 años, este gráfico evidencia que el clúster 0 agrupa a los estudiantes de mayor edad, mientras que los otros clústeres tienen a los estudiantes más jóvenes.

Figura 27

Distribución de la Edad por Clusters



*Nota. * Fuente: desarrollo del autor*

4.6.2. DBSCAN

En este apartado se desarrollará el algoritmo DBSCAN (Density-Based Spatial Clustering of Aplications with Noise), es una técnica de segmentación que identifica grupos de datos basados en la densidad de los puntos (Ester, 1996), este algoritmo nos permitirá clasificar a los estudiantes en diferentes segmentos según las variables seleccionadas en el capítulo anterior, adaptando las transformaciones necesarias para su aplicación.

A diferencia de K-Means este algoritmo de agrupamiento no requiere especificar el número de clústeres previamente, es capaz de identificar registros que no pertenecen a ningún cluster, categorizándolos como ruido.

Para determinar los parámetros adecuados para la segmentación se utilizará el método de k-distancias que nos permite identificar el valor óptimo de EPS (Epsilon), de esta forma obtendremos clústeres resultantes que reflejen patrones reales presentes en los datos.

Selección de variables

En el caso de DBSCAN a diferencia de K-Means, las variables categóricas como modalidad, jornada, parroquia de residencia, sector_domicilio y estado civil, se transformaron mediante Label Encoding asignando un número único por categoría y no con One- Hot Encoding, debido a que DBSCAN no depende de la distancia euclidiana.

La codificación numérica no distorsiona los resultados, variables numéricas y ordinales como edad, nivel y sexo se procesan similar en ambos algoritmos utilizando StandarScaler y un mapeo binario en el caso de sexo, se detalla las transformaciones realizadas por campo en la

Tabla 5

Selección de Variables DBSCAN

Tabla 5

Selección de Variables DBSCAN

Antes	Tratamiento Aplicado	Después
periodo, escuela, carrera, pais_residencia, provincia_residencia, canton_residencia, colegio, fecha_nacimiento	No consideradas por irrelevancia en el clustering con DBSCAN	No consideradas
modalidad (categórica: presencial, híbrida, virtual)	Codificación categórica asignando un número único a cada categoría	Valores numéricos (0, 1, 2, etc.)
jornada (categórica: matutina, nocturna, fin de semana)	Codificación categórica asignando un número único a cada categoría	Valores numéricos (0, 1, 2, etc.)
parroquia_residencia (categórica con muchos valores)	Codificación categórica asignando un número único a cada categoría	Valores numéricos (0, 1, 2, etc.)
sector_domicilio (categórica: norte, sur, valles)	Codificación categórica asignando un número único a cada categoría	Valores numéricos (0, 1, 2)
estado_civil (categórica: soltero, casado, unión libre, etc.)	Codificación categórica asignando un número único a cada categoría	Valores numéricos (0, 1, 2, etc.)
nivel (ordinal: primero a décimo)	Codificación ordinal y escalado con StandardScaler	Valores transformados con media 0 y desviación estándar 1
sexo (categórica: hombre/mujer)	Mapeo binario (hombre=0, mujer=1)	Valores 1 y 0
edad (numérica)	Escalado con StandardScaler	Valores transformados con media 0 y desviación estándar 1

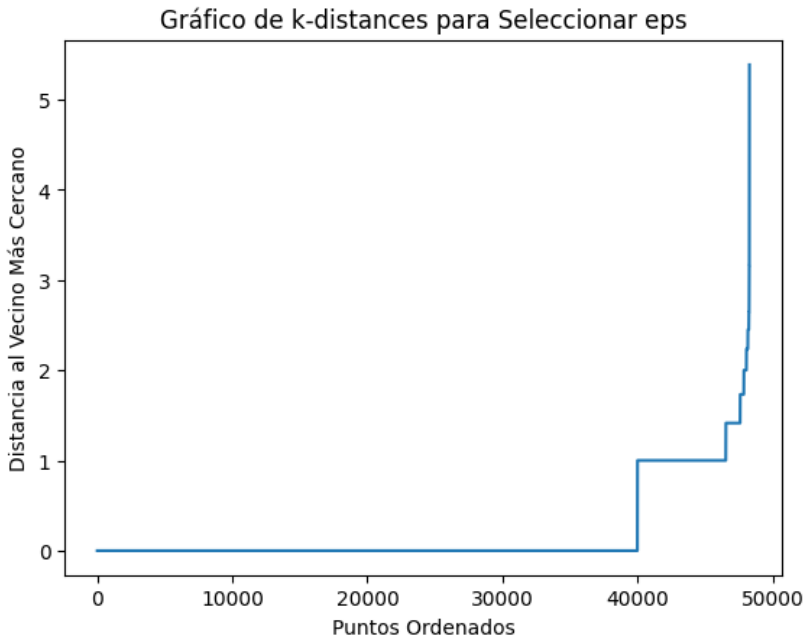
*Nota. * Fuente: desarrollo del autor*

Método de k-distances

Para seleccionar los parámetros adecuados, se utilizará el método de k-distancias vecinos como se puede evidenciar en la Figura 28 *K Distances parámetro DBSCAN* más cercanos, que permite identificar el valor óptimo de EPS analizando el punto donde la curva de distancias muestra una inflexión significativa.

Figura 28

K Distances parámetro DBSCAN



*Nota. * Fuente: desarrollo del autor*

- En este gráfico se puede observar donde la curva empieza a subir de forma abrupta el punto se llama punto de codo, este valor es el recomendado para EPS (Epsilon), porque antes de este punto los valores se agrupan de manera compacta y luego son más dispersos.

- El codo se encuentra aproximadamente entre 1.5 y 2.0 eligiendo menores a 1.5 se obtendrían Clusters pequeños y mayores a 2.0 se corre el riesgo de juntar puntos que no deberían estar en el mismo grupo.

- Para el valor de la variable min_samples se considerará un valor alto como 600 (determinado en las pruebas iterativas) debido a que valores menores generan clústeres muy pequeños.

Modelo DBSCAN

Se ejecutó el algoritmo DBSCAN Figura 29 *Modelo DBSCAN* utilizando los parámetros seleccionados EPS en 2.0 y min_samples 600 logrando identificar clústeres basados en la densidad de los datos, cada estudiante fue asignado a un clúster según la proximidad y cantidad de puntos vecinos, incluyendo aquellos clasificados como ruido.

Figura 29

Modelo DBSCAN

```
    dbscan = DBSCAN(eps=2.0, min_samples=600)  # Variables determinadas
    df_dbscan['cluster_dbscan'] = dbscan.fit_predict(df_dbscan)

    cluster_counts = df_dbscan['cluster_dbscan'].value_counts().sort_index()
    #cluster_counts = df_dbscan['cluster_dbscan'].value_counts()
    print("Número de registros por clúster:")
    print(cluster_counts)
[27] ✓ 5.5s
...  Número de registros por clúster:
cluster_dbscan
-1      9297
 0     10173
 1      2813
 2      2598
 3     12222
 4      2771
 5      2816
 6      1889
 7      1132
 8      1084
 9       740
 10      750
Name: count, dtype: int64
```

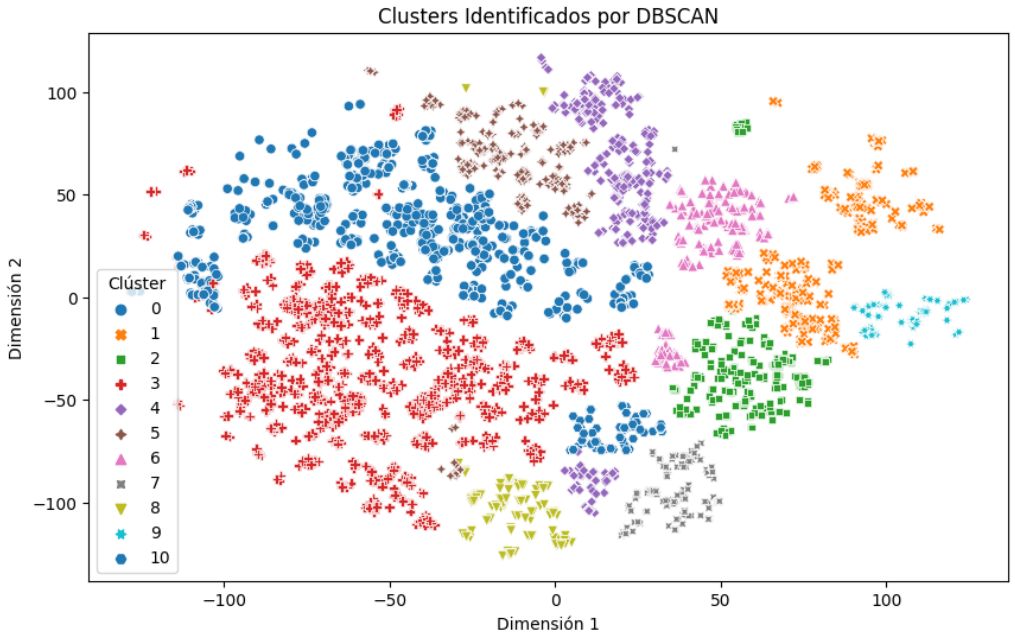
*Nota. * Fuente: desarrollo del autor, el código fuente completo se encuentra en el ANEXO 1.*

Como se puede apreciar en el grafico Figura 30

Clusters DBSCAN se muestran los clusters generados por este algoritmo

Figura 30

Clusters DBSCAN



*Nota. * Fuente: desarrollo del autor*

A continuación, la descripción de los clústeres obtenidos cabe indicar que el cluster de ruido fue omitido porque no representa un grupo homogéneo con características comunes.

CLÚSTER 0:

De color azul, de forma círculos ubicado en la parte superior izquierda del gráfico, grupo con 10,173 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 23.08 años con un rango de 16 a 46 años, el sexo más frecuente es mujer, el nivel académico más frecuente es primero, la parroquia de residencia más común es calderón, la mayoría reside en el sector norte, el estado civil predominante es soltero.

CLÚSTER 1:

De color naranja, con forma cruces en x, ubicado en la parte superior derecha del gráfico, grupo con 2,813 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 23.38 años con un rango de 17 a 42 años, el sexo más frecuente es mujer, el nivel académico más frecuente es quinto, la parroquia de residencia más común es san antonio de pichincha, la mayoría reside en el sector norte, el estado civil predominante es soltero.

CLÚSTER 2:

De color verde, de forma cuadrados, ubicado en la parte derecha del gráfico, grupo con 2,598 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 23.29 años con un rango de 17 a 44 años, el sexo más frecuente es hombre, el nivel académico más frecuente es primero, la parroquia de residencia más común es quito, la mayoría reside en el sector sur, el estado civil predominante es soltero.

CLÚSTER 3:

De color rojo, de forma cruces +, ubicado en la parte central del gráfico, grupo con 12,222 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 23.23 años con un rango de 16 a 50 años, el sexo más frecuente es mujer, el nivel académico más frecuente es primero, la parroquia de residencia más común es cotocollao, la mayoría reside en el sector norte, el estado civil predominante es soltero.

CLÚSTER 4:

de color morado, de forma rombos ubicado en la parte superior del gráfico, grupo con 2,771 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 23.74 años con un rango de 17 a 45 años, el sexo

más frecuente es mujer, el nivel académico más frecuente es primero, la parroquia de residencia más común es kennedy, la mayoría reside en el sector norte, el estado civil predominante es soltero.

CLÚSTER 5:

De color marrón, de forma diamantes pequeños ubicado en la parte superior central del gráfico, grupo con 2,816 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 22.54 años con un rango de 17 a 43 años, el sexo más frecuente es mujer, el nivel académico más frecuente es primero, la parroquia de residencia más común es el condado, la mayoría reside en el sector norte, el estado civil predominante es soltero.

CLÚSTER 6:

de color rosado, de forma triángulos ubicado en la parte superior derecha del gráfico, grupo con 1,889 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 22.88 años con un rango de 17 a 45 años, el sexo más frecuente es mujer, el nivel académico más frecuente es primero, la parroquia de residencia más común es pomasqui, la mayoría reside en el sector norte, el estado civil predominante es soltero.

CLÚSTER 7:

De color gris, asteriscos, ubicado en la parte inferior derecha del gráfico, grupo con 1,132 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 23.32 años con un rango de 17 a 40 años, el sexo más frecuente es hombre, el nivel académico más frecuente es quinto, la parroquia de residencia más común es la magdalena, la mayoría reside en el sector sur, el estado civil predominante es soltero.

CLÚSTER 8:

De color amarillo de forma triángulos invertidos ubicado en la parte inferior central del gráfico, grupo con 1,084 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 23.15 años con un rango de 17 a 39 años, el sexo más frecuente es mujer, el nivel académico más frecuente es primero, la parroquia de residencia más común es Guamaní, la mayoría reside en el sector sur, el estado civil predominante es soltero.

CLÚSTER 9:

De color celeste, de forma estrellas ubicado en la parte derecha del gráfico, grupo con 740 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 23.12 años con un rango de 17 a 39 años, el sexo más frecuente es hombre, el nivel académico más frecuente es primero, la parroquia de residencia más común es Solanda, la mayoría reside en el sector sur, el estado civil predominante es soltero.

CLÚSTER 10:

De color azul oscuro, de forma círculos ubicado en la parte inferior derecha del gráfico, grupo con 750 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 22.72 años con un rango de 17 a 34 años, el sexo más frecuente es mujer, el nivel académico más frecuente es primero, la parroquia de residencia más común es la ferroviaria, la mayoría reside en el sector sur, el estado civil predominante es soltero.

4.6.3. Modelo K-Prototypes

Se desarrollará el algoritmo K-Prototypes, que es una extensión de K-Means algoritmo ya ejecutado, diseñado para manejar variables categóricas como variables numéricas, en mi caso tengo variables de los dos tipos por lo que este algoritmo permitirá segmentar a los estudiantes en grupos considerando la combinación de características cuantitativas y cualitativas.

K-Prototypes opera asignando cada estudiante a un grupo basado en la similitud de sus características, para ello se seleccionan K centroides, que representan los centros de cada grupo, Luego los estudiantes se asignan de forma iterativa al cluster más cercano usando la distancia euclidiana para la variable numérica y la disimilitud para las variables categóricas, basándonos en la frecuencia de coincidencias, este proceso se realiza hasta que los clústeres se estabilizan

Para determinar el número óptimo de clústeres se aplicará el método del codo que evalúa la variabilidad dentro de los grupos y nos ayuda a mostrar el punto donde añadir más clústeres no aporta a la segmentación, se añade en este punto el costo de inercia que mide la disimilitud total dentro de los clusters.

Selección de variables

Para definir el uso de las variables debido a la naturaleza de mis características en su mayoría categóricas se utilizó un análisis de correlación de Cramer V, no se utilizó en este algoritmo la correlación de Pearson que también podría usar al convertir las características a variables numéricas ya que esto no reflejaría una relación real entre las variables.

Se ha encontrado de la tabla creada por Cramer en la

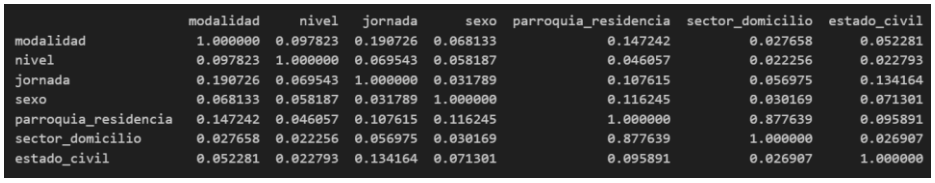
Figura 31

Correlación Cramer K-Prototypes:

- Hay una relación positiva alta entre el sector de domicilio y la parroquia de residencia 0.87 lo que indica que estas variables están fuertemente asociadas y podrían contener información redundante esto es natural ya que el sector generaliza a la parroquia
- Existe una correlación entre estado civil y parroquia de residencia 0.096, que si bien no es alta se puede explorar si ciertos sectores mayor prevalencia de estados civiles específicos.
- Se ha encontrado una relación moderada entre la jornada y la modalidad 0.191 lo que sugiere que ciertas jornadas podrían estar más asociadas a un tipo de modalidad
- Se ha encontrado una relación moderada entre parroquia de residencia y Jornada 0.107 indicando que la parroquia de residencia podría influir en la elección de la jornada, sería interesante explotar esta relación
- Se ha encontrado relación débil entre el sexo del estudiante 0.058 y el nivel académico
- Se muestra una relación débil entre el estado civil y el nivel académico 0.022 no influye en la progresión académica de niveles.

Figura 31

Correlación Cramer K-Prototypes



	modalidad	nivel	jornada	sexo	parroquia_residencia	sector_domicilio	estado_civil
modalidad	1.000000	0.097823	0.190726	0.068133	0.147242	0.027658	0.052281
nivel	0.097823	1.000000	0.069543	0.058187	0.046057	0.022256	0.022793
jornada	0.190726	0.069543	1.000000	0.031789	0.107615	0.056975	0.134164
sexo	0.068133	0.058187	0.031789	1.000000	0.116245	0.030169	0.071301
parroquia_residencia	0.147242	0.046057	0.107615	0.116245	1.000000	0.877639	0.095891
sector_domicilio	0.027658	0.022256	0.056975	0.030169	0.877639	1.000000	0.026907
estado_civil	0.052281	0.022793	0.134164	0.071301	0.095891	0.026907	1.000000

*Nota. * Fuente: desarrollo del autor*

El Tratamiento de las variables para este algoritmo se detallan en la Tabla 6 *Análisis Variables K-Prototypes*

Tabla 6

Análisis Variables K-Prototypes

Antes	T**ransformación**	Después
periodo, escuela, carrera, país_residencia, provincia_residencia, cantón_residencia, colegio, fecha_nacimiento	No consideradas en K-Prototypes	No consideradas
modalidad (presencial, híbrida, virtual)	Se mantiene como categórica (sin transformación)	Se usa directamente en K-Prototypes
jornada (matutina, nocturna, fin de semana)	Se mantiene como categórica	Se usa directamente en K-Prototypes
parroquia_residencia (categoría con muchos valores)	Se mantiene como categórica	Se usa directamente en K-Prototypes
sector_domicilio (eliminado)	Eliminado del análisis	No se usa
estado civil (soltero, casado, unión libre, etc.)	Se mantiene como categórica	Se usa directamente en K-Prototypes
nivel (ordinal: primero a décimo), sexo(hombre/mujer)	Se mantiene como categórica (sin escalado)	Se usa directamente en K-Prototypes
edad (numérica)	Se mantiene como numérica (sin transformación)	Se usa directamente en K-Prototypes

*Nota. * Fuente: desarrollo del autor*

Método del Codo

Esta técnica aplicada para determinar el número óptimo de clusters en K-Prototypes, para agrupar los datos mixtos de nuestro dataset, que combina una variable numérica y varias categóricas.

El procedimiento consistió en evaluar diferentes valores para k y calcular el costo de inercia que representa la distancia interna entre los puntos dentro de un cluster, a medida que (k) aumenta la inercia disminuye, ya que los clústeres son más pequeños y homogéneos formando un codo en la gráfica (IBM, 2024),

En este análisis se probaron valores entre K entre 3 y 6 (el costo computacional es alto para muchas más iteraciones) y se graficó el costo de inercia para visualizar el punto óptimo como se puede apreciar en la .

Figura 32

Método del Codo K-Prototypes.

Figura 32

Método del Codo K-Prototypes

*Nota. * Fuente: desarrollo del autor*

Método del Codo K-Prototypes

- Se puede observar que la disminución de la inercia es pronunciada al inicio, entre 3 y 4 clústeres, posteriormente la reducción comienza a desacelerarse.
- El codo más marcado se encuentra entre 3 y 4, lo que sugiere que este valor es óptimo para la segmentación de la data de estudiantes.
- Un número de clústeres inferior a 3 podría no capturaría la variabilidad de los datos, ya que la inercia disminuye significativamente en esta región.
- Un número mayor a 4 puede no ser necesario, dado que la reducción en la inercia no es muy significativa.

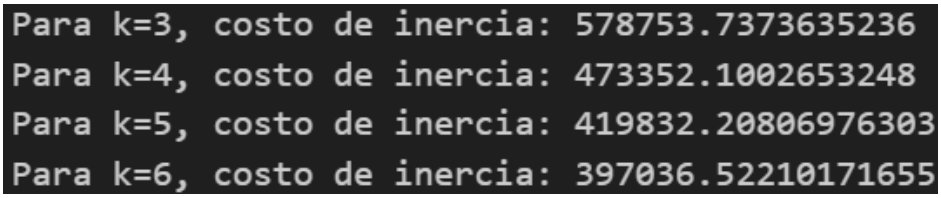
Se puede concluir que los estudiantes de la IES se agrupan de manera más eficiente en cuatro segmentos, permitiendo una segmentación balanceada y significativa.

Luego de calcular el método del codo con los cálculos de inercia se obtiene los siguientes valores mostrados en la Figura 33

Costo Inercia para K en K-Prototypes

Figura 33

Costo Inercia para K en K-Prototypes



*Nota. * Fuente: desarrollo del autor*

Existe una tendencia decreciente a medida que el costo de la inercia disminuye en cada iteración que se realizó para k, esto es esperado debido a que mientras más clusters se generan implica que los puntos están más cerca de sus centroides, pero hay que notar que de 3 a 4 la reducción es más notoria de 4 a 5 baja un poco y de 5 a 6 no es tanto como al inicio.

Concluimos que 4 puede ser el mejor número de clústeres, ya que después de este punto la reducción en la inercia es mucho menos pronunciada.

Modelo K-Prototypes

Se ejecutó el algoritmo de **K-Prototypes** con los **4 clústeres** sugeridos en el método del codo, segmentando a los estudiantes en **4 grupos** según similitudes de sus características, considerando tanto variables categóricas como numéricas cada estudiante fue asignado a un clúster, agregando esta clasificación al dataset y contando la cantidad de registros en cada grupo, como se puede apreciar en la Figura 34 *Modelo K-Prototypes 1* y en la Figura 35 Modelo K-Prototypes 2

Figura 34

Modelo K-Prototypes 1

```python
# Definir el número de clusters (ajustar según necesidad)
k = 4  # Definido por el metodo del codo

# Aplicar K-Prototypes
kproto = KPrototypes(n_clusters=k, init='Huang', verbose=2, random_state=42)
clusters = kproto.fit_predict(data_matrix, categorical=[df_kproto_encoded.columns.get_loc(col) for col in categorical_cols])

# Agregar la asignación de clusters al dataframe
df_kproto['cluster'] = clusters

# Mostrar los primeros resultados
print(df_kproto.head())
```

*Nota. * Fuente: desarrollo del autor, el código fuente completo se encuentra en el ANEXO 1.*

Figura 35

Modelo K-Prototypes 2

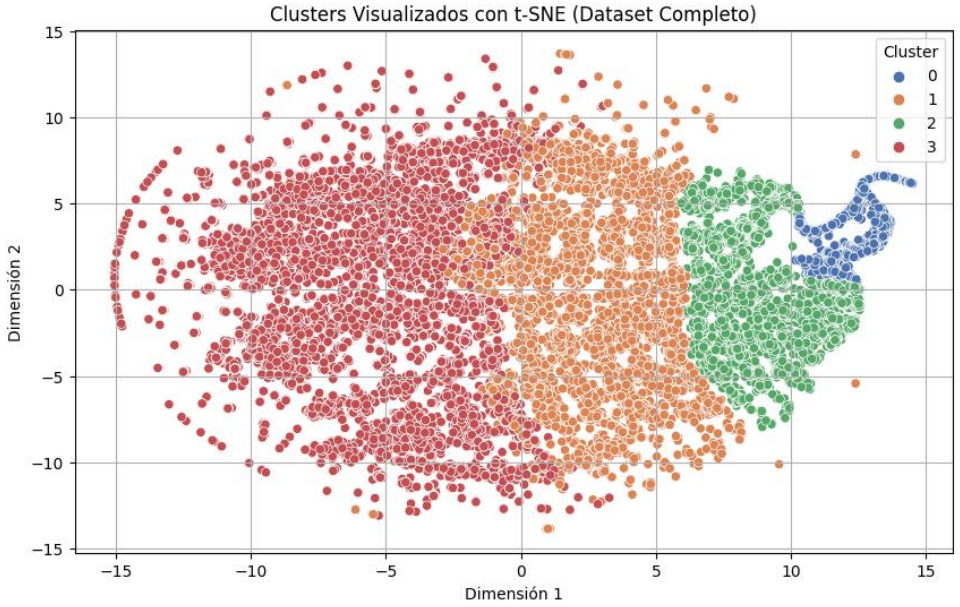
```
Número de registros por clúster:
cluster
0     1941
1    15495
2     6644
3    24205
Name: count, dtype: int64
```

*Nota. * Fuente: desarrollo del autor*

El gráfico muestra la segmentación de los estudiantes con el algoritmo **K-Prototypes**, visualizada mediante **t-SNE** para reducir la dimensionalidad a dos dimensiones, cada punto representa un registro de estudiante y los colores indican los **4 clústeres** que se formaron según similitudes en sus características, esta visualización permite evaluar la separación y estructura de los grupos en el dataset como se aprecia en la Figura 36 *Clusters K-Prototypes*

Figura 36

Clusters K-Prototypes



*Nota. * Fuente: desarrollo del autor*

La descripción de cada clusters se muestra a continuación

CLUSTER 0:

De color azul ubicado en la parte superior derecha del gráfico con 1941 estudiantes, la mayoría de los ellos pertenecen a la modalidad presencial, la jornada nocturna es la más común, la edad promedio es de 40.87 años, con un rango de 36 a 68 años, el sexo más frecuente, es mujer, el nivel académico más frecuente es primero, la parroquia de residencia más común es calderón, la mayoría reside en el sector norte, el estado civil predominante es casado, la carrera más frecuente es tecnología superior en optometría, el periodo más frecuente es abril 2024 - septiembre 2024.

CLÚSTER 1:

De color Naranja ubicado en la parte central izquierda del gráfico, con 15495 estudiantes, la mayoría de los ellos pertenecen a la modalidad presencial, la jornada matutina es la más común la edad promedio es de 24.30 años, con un rango de 22 a 27 años, el sexo más frecuente es hombre,

el nivel académico más frecuente es quinto, la parroquia de residencia más común es Calderón, la mayoría reside en el sector Norte, el estado civil predominante es soltero, la carrera más frecuente es tecnología superior en administración financiera, el periodo más frecuente es abril 2020 - septiembre 2020.

CLÚSTER 2:

De color verde ubicado en la parte central derecha del gráfico con 6644 estudiantes la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada nocturna es la más común, la edad promedio es de 30.56 años, con un rango de 28 a 35 años, el sexo más frecuente es mujer, el nivel académico más frecuente es quinto, la parroquia de residencia más común es calderón, la mayoría reside en el sector norte, el estado civil predominante es soltero, la carrera más frecuente es tecnología superior en administración financiera, el periodo más frecuente es abril 2020 - septiembre 2020.

CLUSTER 3:

De color rojo ubicado en la parte izquierda del gráfico con 24205 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, La jornada matutina es la más común, la edad promedio es de 19.95 años, con un rango de 16 a 22 años, el sexo más frecuente es mujer, el nivel académico más frecuente es primero, la parroquia de residencia más común es calderón, la mayoría reside en el sector norte, el estado civil predominante es soltero, la carrera más frecuente es diseño gráfico, el periodo más frecuente es abril 2020 - septiembre 2020.

Estadísticos por Cluster

Se detallan a continuación los estadísticos principales de los clústeres encontrados por K-Prototypes

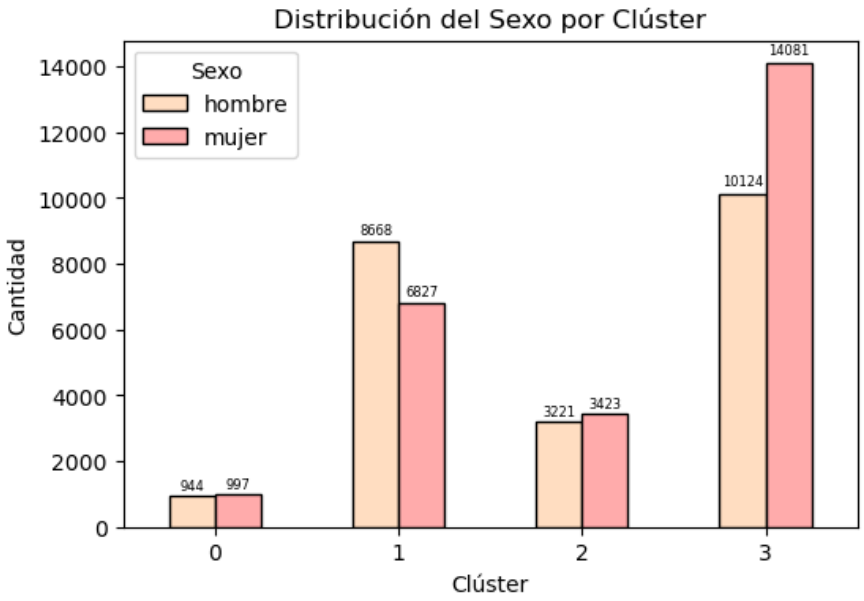
Variable Sexo

En la Figura 37 *Distribución por sexo cluster K-Prototypes* se muestra la distribución de género en los cuatro clústeres de K-Prototypes, los hombres son mayoría en el clúster 1, mientras que en el clúster 3 ocurre lo contrario.

- En el clúster 0, la distribución es casi igual con 944 hombres y 997 mujeres.
- En el clúster 1, los hombres son mayoría con 8,668, mientras que las mujeres alcanzan 6,827.
- En el clúster 2, la distribución es más equitativa con 3,221 hombres y 3,423 mujeres.
- En el clúster 3, las mujeres son mayoría, con 14,081 mujeres y 10,124 hombres, siendo este el clúster con mayor diferencia de género.

Figura 37

Distribución por sexo cluster K-Prototypes



*Nota. * Fuente: desarrollo del autor*

Variable Modalidad

El gráfico muestra la distribución de la modalidad de estudio en los cuatro clústeres de K-Prototypes, hay que notar que la modalidad presencial es ampliamente predominante en todos los clústeres, mientras que las modalidades híbridas y en línea tienen una presencia mucho menor, esto es esperable pues el fuerte de la institución es la modalidad presencial.

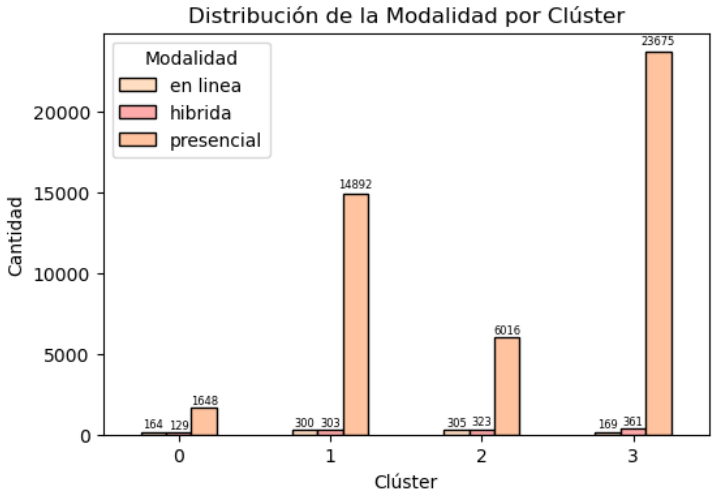
- En el clúster 0, 1,648 estudiantes pertenecen a la modalidad presencial, con una cantidad reducida en la modalidad en línea (164) e híbrida (129).
- En el clúster 1, la modalidad presencial es la dominante con 14,892 estudiantes, mientras que la modalidad híbrida (303) y en línea (300) tienen valores muy bajos.
- En el clúster 2, se observa una tendencia similar, con 6,016 estudiantes en modalidad presencial, mientras que la modalidad en línea (305) e híbrida (323) representan una minoría.

- En el clúster 3, 23,675 estudiantes están en la modalidad presencial, lo que representa el grupo con mayor cantidad de estudiantes presenciales, las modalidades en línea (169) e híbrida (361) tienen la menor representación.

Como se evidencia en la Figura 38 *Distribución de cluster por Modalidad K-Prototypes,* la preferencia por la modalidad presencial es muy clara, con más de 90% de los estudiantes en esta categoría, esto sugiere que la mayoría de los estudiantes prefieren asistir físicamente a clases, lo que puede estar relacionado con la oferta académica y la confianza en esta modalidad.

Figura 38

Distribución de cluster por Modalidad K-Prototypes



*Nota. * Fuente: desarrollo del autor*

Variable Jornada

La

Figura 39

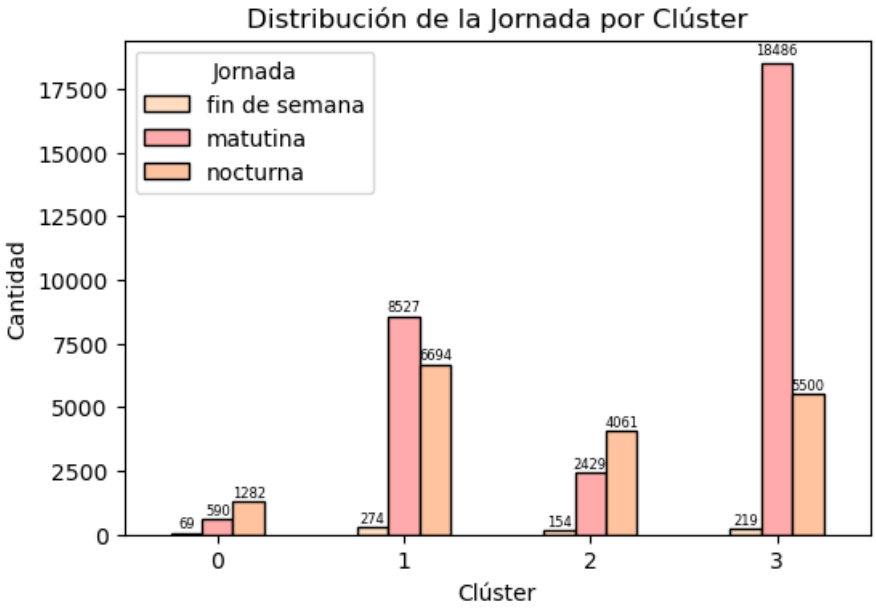
Distribución de cluster por jornada K-Prototypes muestra la distribución de los estudiantes según la jornada en la que están inscritos. Se observa que las jornadas matutina y nocturna predominan, mientras que la jornada de fin de semana tiene una representación mínima.

- Clúster 0: La jornada nocturna es la más frecuente con 1,282 estudiantes, seguida de la jornada matutina con 590 estudiantes y un pequeño grupo en fin de semana (69 estudiantes).
- Clúster 1: La jornada matutina es la predominante, con 8,527 estudiantes, mientras que la nocturna cuenta con 6,694 y fin de semana con 274.
- Clúster 2: Se observa una mayor distribución entre jornadas, con 4,061 estudiantes en nocturna, 2,429 en matutina y 154 en fin de semana.
- Clúster 3: Es el grupo con más estudiantes en jornada matutina (18,486), seguido por nocturna (5,500) y fin de semana con 219.

La jornada matutina es la más elegida en los clústeres 1 y 3, mientras que la jornada nocturna tiene una presencia importante en los clústeres 0 y 2. La jornada de fin de semana es minoritaria en todos los grupos, lo que indica que pocos estudiantes optan por esta modalidad, esto es esperable pues en periodos actuales se está abriendo esta opción de horario.

Figura 39

Distribución de cluster por jornada K-Prototypes



*Nota. * Fuente: desarrollo del autor*

Variable Edad

La Figura 40 *Boxplot edad cluster K-Prototypes* se visualiza la dispersión de las edades de los estudiantes en cada clúster mediante un diagrama de caja y bigotes boxplot se puede interpretar que cada clúster agrupa a estudiantes con diferentes rangos de edad, destacando la diversidad de edad en el clúster 0 y la edad marcada en el clúster 3.

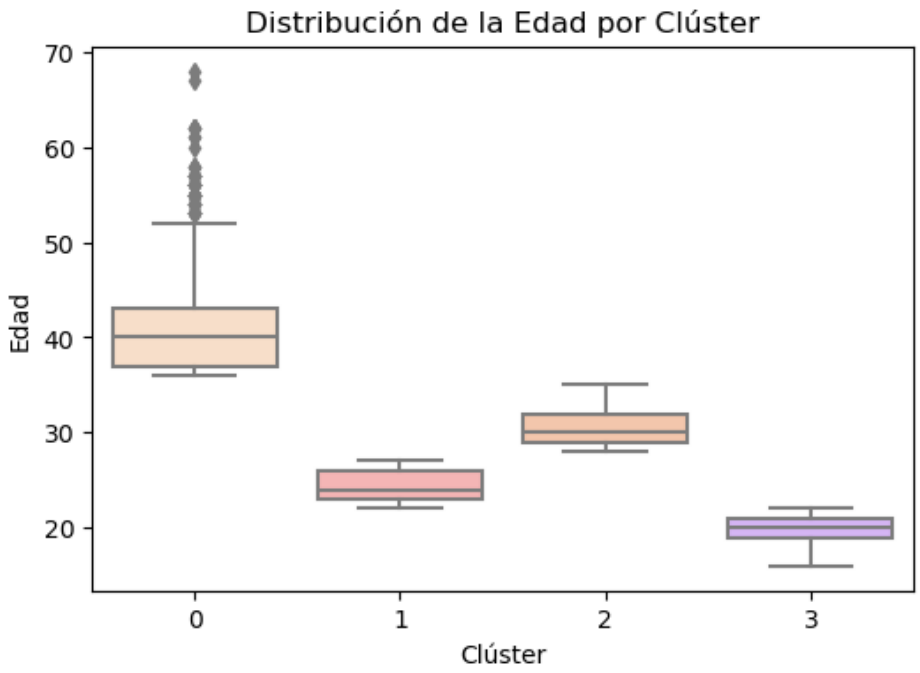
- Clúster 0: Es el grupo con la mayor edad promedio (40.87 años) y con una distribución más amplia, variando entre 36 y 68 años, se visualizan valores atípicos (outliers) hay presencia de estudiantes con edades significativamente mayores que la mayoría, de alguna forma esto se espera pues son personas que tienden a

continuar sus estudios luego de abandonarlos o son trabajadores que necesitan completar su formación.

- Clúster 1: Se compone de estudiantes más jóvenes, con una edad promedio de 24.29 años y un rango entre 22 y 27 años, lo que sugiere una menor variabilidad en este grupo.
- Clúster 2: Tiene una edad promedio de 30.56 años, con valores comprendidos entre 28 y 35 años, lo que indica que está conformado principalmente por adultos jóvenes.
- Clúster 3: Es el grupo con los estudiantes más jóvenes, con una edad promedio de 19.95 años y un rango entre 16 y 22 años, mostrando una menor dispersión en comparación con los otros clústeres.

Figura 40

Boxplot edad cluster K-Prototypes



*Nota. * Fuente: desarrollo del autor*

4.6.4. Modelo K-Modes

En este apartado se implementará el algoritmo K-Modes, una variante de K-Means, que usa específicamente variables categóricas, a diferencia de K-Means, que usa la distancia euclidiana para agrupar datos numéricos K-Modes opera con disimilitudes entre categorías, utilizando la frecuencia de coincidencias como criterio de similitud.

En este caso, dado que todas las variables del dataset son categóricas, K-Modes permitirá segmentar a los estudiantes en grupos basados en características cualitativas como modalidad, nivel, jornada, parroquia de residencia, estado civil y cuantitativa como la edad.

Para determinar el número óptimo de clusters, se aplicará el método del codo en un rango de pruebas, analizando la variabilidad dentro de los grupos y ayudando a identificar el punto en el que añadir más clusters deja de aportar significativamente a la segmentación

Este análisis se pretende permita generar una segmentación adecuada de los estudiantes, facilitando la identificación de patrones en sus características.

Selección de variables

Para la selección de variables en el modelo K-Modes, se utilizó Cramer V como se muestra en la

Tabla Kramer K-Modes no se aplicó la correlación de Pearson, ya que convertir las variables a valores numéricos podría distorsionar las relaciones reales en los datos.

Figura 41

Tabla Kramer K-Modes

	modalidad	nivel	jornada	sexo	parroquia_residencia	sector_domicilio	estado_civil	edad_categoria
modalidad	1.000000	0.097823	0.190726	0.068133	0.147242	0.027658	0.052281	0.115610
nivel	0.097823	1.000000	0.069543	0.058187	0.046057	0.022256	0.022793	0.146317
jornada	0.190726	0.069543	1.000000	0.031789	0.107615	0.056975	0.134164	0.215664
sexo	0.068133	0.058187	0.031789	1.000000	0.116245	0.030169	0.071301	0.021961
parroquia_residencia	0.147242	0.046057	0.107615	0.116245	1.000000	0.877639	0.095891	0.120557
sector_domicilio	0.027658	0.022256	0.056975	0.030169	0.877639	1.000000	0.026907	0.024210
estado_civil	0.052281	0.022793	0.134164	0.071301	0.095891	0.026907	1.000000	0.258624
edad_categoria	0.115610	0.146317	0.215664	0.021961	0.120557	0.024210	0.258624	1.000000

*Nota. * Fuente: desarrollo del autor.*

Se muestran las conclusiones en la tabla generado por Cramer

- Hay una relación positiva alta entre el sector de domicilio y la parroquia de residencia (0.87), lo que indica que estas variables están fuertemente asociadas y podrían contener información redundante tal cual sucedió en el algoritmo de K-Prototypes.

- Existe una correlación entre estado civil y parroquia de residencia (0.095), que, si bien no es alta, podría explorarse si en ciertos sectores hay mayor prevalencia de estados civiles específicos.

- Se ha encontrado una relación moderada entre la jornada y la modalidad (0.190), lo que sugiere que ciertas jornadas podrían estar más asociadas a un tipo de modalidad, podríamos sugerir que por ejemplo la jornada matutina está bastante relacionada con la modalidad presencial.

- Se ha encontrado una relación moderada entre parroquia de residencia y jornada (0.107), indicando que la parroquia de residencia podría influir en la elección de la jornada, lo que sería interesante explorar más a fondo.

- Se ha encontrado relación débil entre el sexo del estudiante (0.058) y el nivel académico, lo que sugiere que el género no tiene un impacto significativo en la progresión académica en niveles.

Se muestra a continuación el tratamiento de variables en la Tabla 7

Tratamiento Variables K-Modes

Tabla 7

Tratamiento Variables K-Modes

Antes	Transformación K-Modes	Después
modalidad (presencial, híbrida, virtual)	Se mantiene como categórica (sin transformación)	Se usa directamente en K-Modes
jornada (matutina, nocturna, fin de semana)	Se mantiene como categórica	Se usa directamente en K-Modes
parroquia_residencia (categoría con muchos valores)	Se mantiene como categórica	Se usa directamente en K-Modes
sector_domicilio (eliminado)	Eliminado del análisis	No se usa
estado civil (soltero, casado, unión libre, etc.)	Se mantiene como categórica	Se usa directamente en K-Modes
nivel (ordinal: primero a décimo), sexo(hombre/mujer)	Se mantiene como categórica (sin escalado)	Se usa directamente en K-Modes
edad (numérica)	Convertida a categórica (rangos)	Se usa directamente en K-Modes
estudiante_id, periodo id, escuela_id, carrera_id, colegio_id, periodo, escuela, carrera, país_residencia, provincia_residencia, cantón_residencia, fecha_nacimiento, colegio	Eliminado del análisis	No se usa

*Nota. * Fuente: desarrollo del autor*

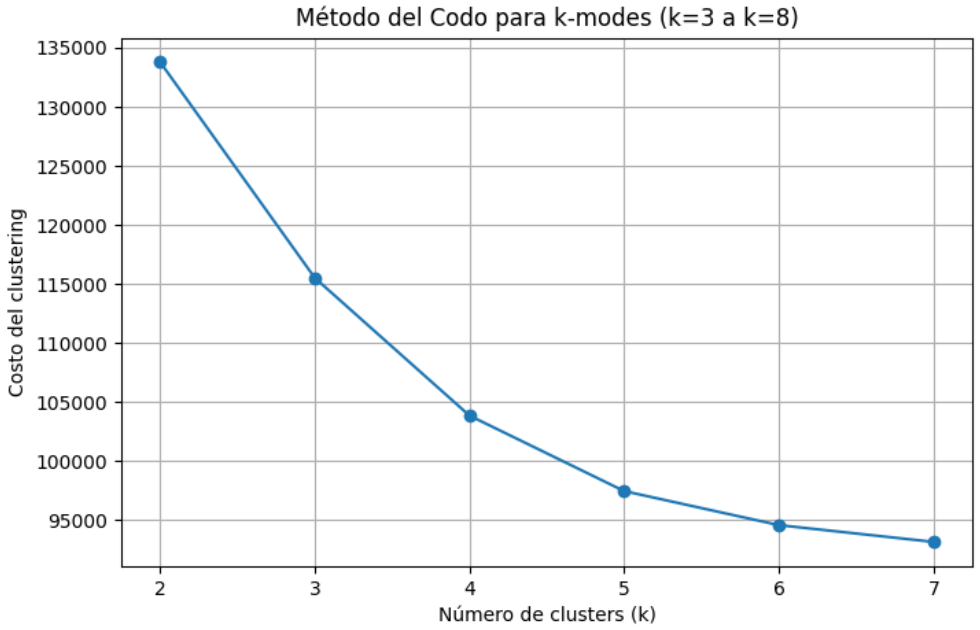
Método del Codo

Esta técnica se aplicó para determinar el número óptimo de clusters en K-Modes, diseñado específicamente para agrupar datos categóricos.

En este análisis, se probaron valores de k entre 2 y 7, y se graficó la reducción del costo para visualizar el punto en el que añadir más clusters deja de aportar significativamente a la segmentación como se aprecia en la Figura 42 *Método del codo K-Modes*

Figura 42

Método del codo K-Modes



*Nota. * Fuente: desarrollo del autor*

- Se puede observar que la disminución del costo de disimilitud es pronunciada al inicio, entre 3 y 5 clusters, posteriormente la reducción comienza a desacelerarse.

- El codo más marcado se encuentra entre 5 y 6, lo que sugiere que este valor es óptimo para la segmentación de la data de estudiantes.
- Valores de 3 podría ser insuficiente para capturar la variabilidad de los datos y valores mayores a 6 puede no ser necesario, dado que la reducción en el costo no es muy significativa, lo que indica que los grupos adicionales no aportan una mejora sustancial a la segmentación.
- Se puede concluir que los estudiantes de la IES se agrupan de manera más eficiente en cinco, permitiendo una segmentación balanceada y significativa.

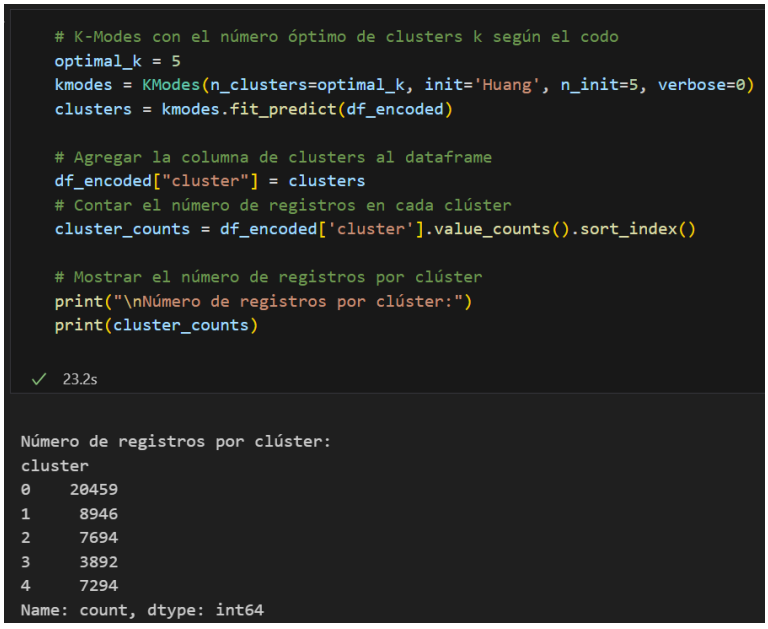
Modelo K-Modes

Se ejecutó el algoritmo de K-Modes con los 5 clústeres sugeridos en el método del codo, segmentando a los estudiantes en grupos según similitudes en sus características categóricas cada estudiante fue asignado a un clúster

Agregando esta clasificación al dataset y contando la cantidad de registros en cada grupo, permitiendo así una segmentación más estructurada y significativa de la población estudiantil como se aprecia en la

Figura 43

Modelo K-Modes



*Nota. * Fuente: desarrollo del autor, el código fuente completo se encuentra en el ANEXO 1.*

Clusters

La

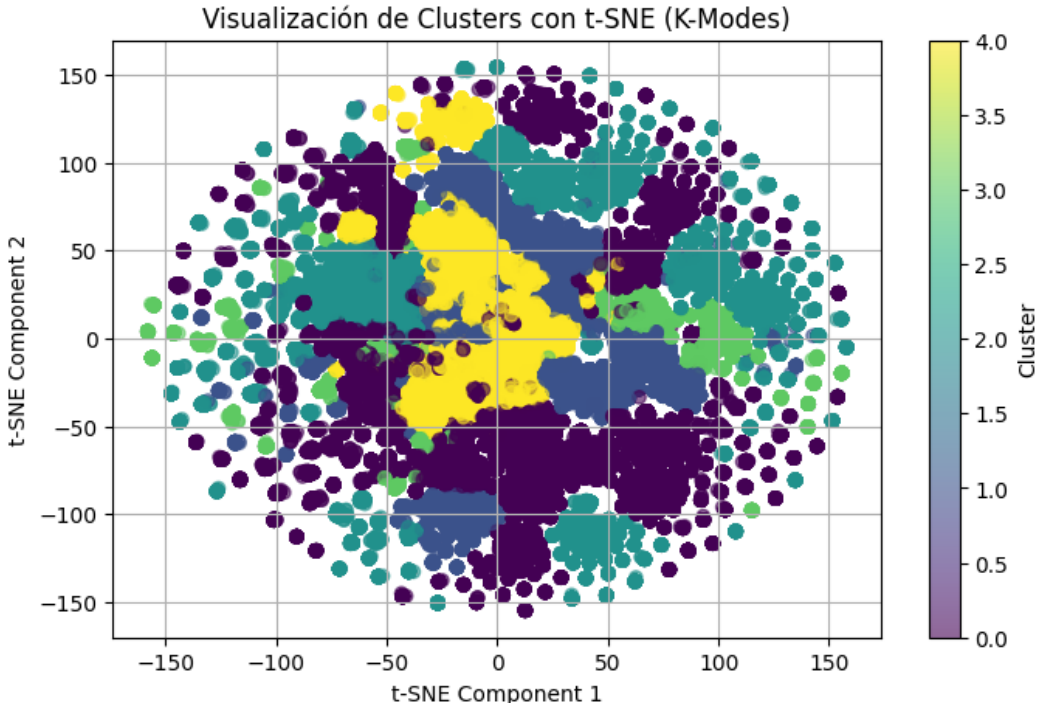
Figura 44

Clusters K-Modes muestra la segmentación de los estudiantes con el algoritmo K-Modes, visualizada mediante t-SNE para reducir la dimensionalidad

Se observa que algunos grupos presentan una separación más definida, mientras que otros tienen solapamiento, lo que sugiere que puede existir similitud entre ciertos segmentos de estudiantes.

Figura 44

Clusters K-Modes



*Nota. * Fuente: desarrollo del autor*

Se muestra la descripción a continuación la descripción de los clusters

CLUSTER 0:

Ubicado en la parte superior izquierda del gráfico, 20459 estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada matutina es la más común, la edad promedio es de 22.14 años con un rango de 16 a 68 años, el sexo más frecuente es hombre, el nivel académico más frecuente es primero, la parroquia de residencia más común es calderón, la mayoría reside en el sector norte, el estado civil predominante es soltero

CLUSTER 1:

Ubicado en la parte central izquierda del gráfico, con 8,946 estudiantes

la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada matutina es la más común, la edad promedio es de 22.31 años, con un rango de 17 a 58 años, el sexo más frecuente es mujer, el nivel académico más frecuente es segundo, la parroquia de residencia más común es calderón, la mayoría reside en el sector norte, el estado civil predominante es soltero.

CLUSTER 2:

Ubicado en la parte derecha del gráfico, con 7,694. Estudiantes, la mayoría de los estudiantes pertenecen a la modalidad presencial, la jornada nocturna es la más común, la edad promedio es de 28.94 años, con un rango de 17 a 62 años, el sexo más frecuente es hombre, el nivel académico más frecuente es tercero, la parroquia de residencia más común es cotocollao, la mayoría reside en el sector norte, el estado civil predominante es soltero.

CLUSTER 3:

Ubicado en la parte inferior izquierda del gráfico con 3892 estudiantes, la mayoría de ellos pertenecen a la modalidad presencial, la jornada matutina es la más común, la edad promedio es de 23.60 años, con un rango de 18 a 54 años, el sexo más frecuente es mujer, el nivel académico más frecuente es quinto, la parroquia de residencia más común es cotocollao. la mayoría reside en el sector norte, el estado civil predominante es soltero.

CLUSTER 4:

Ubicado en la parte inferior derecha del gráfico con 7294 estudiantes, la mayoría de ellos pertenecen a la modalidad presencial, la jornada nocturna es la más común, la edad promedio es de 23.93 años, con un rango de 16 a 58 años, el sexo más frecuente es mujer, el nivel académico más frecuente es quinto, la parroquia de residencia más común es calderón, la mayoría reside en el sector norte, el estado civil predominante es soltero.

4.6.5. Modelo Autoencoder + K-Modes

En este apartado se implementará un modelo basado en Autoencoder y K-Modes para la segmentación de estudiantes, combinando una red neuronal Autoencoder con un algoritmo de clustering categórico como K- Modes, permitiendo representar de manera más eficiente la estructura de los datos antes de realizar la agrupación.

La red está compuesta por un encoder, que transforma los datos originales en una representación de menor dimensión, y un decoder que reconstruye los datos a partir de esta representación.

Una vez obtenida la representación latente, se aplica el algoritmo K-Modes para la agrupación, en este modelo no es necesario aplicar el método del codo para determinar el número óptimo de clusters, porque no se minimiza una función de inercia, sino que asigna cada observación al cluster con mayor similitud en función de las variables categóricas disponibles

Selección de variables

La descripción de la selección de las variables se muestra en la

Tabla 8

Selección de Variables Autoencoder

Antes	Transformación Autoencoder + K-Modes	Después
modalidad (presencial, híbrida, virtual)	Se convierte en One-Hot Encoding para el Autoencoder	Se usa la representación latente en K-Modes
jornada (matutina, nocturna, fin de semana)	Se convierte en One-Hot Encoding para el Autoencoder	Se usa la representación latente en K-Modes
parroquia_residencia (categoría con muchos valores)	Se convierte en One-Hot Encoding para el Autoencoder	Se usa la representación latente en K-Modes
sector_domicilio (eliminado)	Eliminada del análisis, dado que anteriormente se determinó que tiene una alta relación con la parroquia de domicilio	No se usa
estado_civil (soltero, casado, unión libre, etc.)	Se convierte en One-Hot Encoding para el Autoencoder	Se usa la representación latente en K-Modes
nivel (ordinal: primero a décimo)	Se convierte en One-Hot Encoding para el Autoencoder (sin escalado)	Se usa la representación latente en K-Modes
sexo (hombre/mujer)	Se convierte en One-Hot Encoding para el Autoencoder	Se usa la representación latente en K-Modes
edad (numérica)	Se convierte en categorías (rangos de edad) y luego en One-Hot-Encoding para el Autoencoder	Se usa la representación latente

*Nota. * Fuente: desarrollo del autor*

Modelo Autoencoder + Kmodes

Se ejecutó un Autoencoder para reducir la dimensionalidad de los datos categóricos, generando una representación latente optimizada luego se aplicó el algoritmo K- Modes con un número predefinido de clústeres, permitiendo así una segmentación más estructurada y

significativa de la población estudiantil se puede evidenciar el modelo en la Figura 45

Modelo Autoencoder 1

Figura 45

Modelo Autoencoder 1

```
# Aplicar K-Modes en la representación latente
optimal_k = 3  # Ajusta según el método del codo
kmodes = KModes(n_clusters=optimal_k, init="Huang", n_init=5, verbose=0)
clusters = kmodes.fit_predict(df_cleaned)
```

*Nota. * Fuente: desarrollo del autor*

Los parámetros elegidos para el entrenamiento del autoencoder variacional (vae) buscan equilibrio entre precisión y eficiencia computacional sin tardar demasiado, se establecieron 50 épocas para garantizar un aprendizaje suficiente sin riesgo de sobreajuste, mientras que un "batch size" de 32 optimiza el uso de memoria y la estabilidad en la actualización de los pesos, además, el uso de shuffle=True evita que el modelo dependa del orden de los datos mejorando su capacidad de generalización, la elección de estos valores permite obtener una representación latente efectiva para la posterior aplicación del algoritmo K-Modes como se puede evidenciar en la Figura 46

Modelo Autoencoder 2

Figura 46

Modelo Autoencoder 2

```
# Entrenar el Categorical VAE
df_numpy = df_onehot.values
vae.fit(df_numpy, df_numpy, epochs=50, batch_size=32, shuffle=True, verbose=1)

# Extraer la representación latente
encoder = keras.Model(inputs, z_mean)
latent_representation = encoder.predict(df_numpy)
```

*Nota. * Fuente: desarrollo del autor, , el código fuente completo se encuentra en el ANEXO 1.*

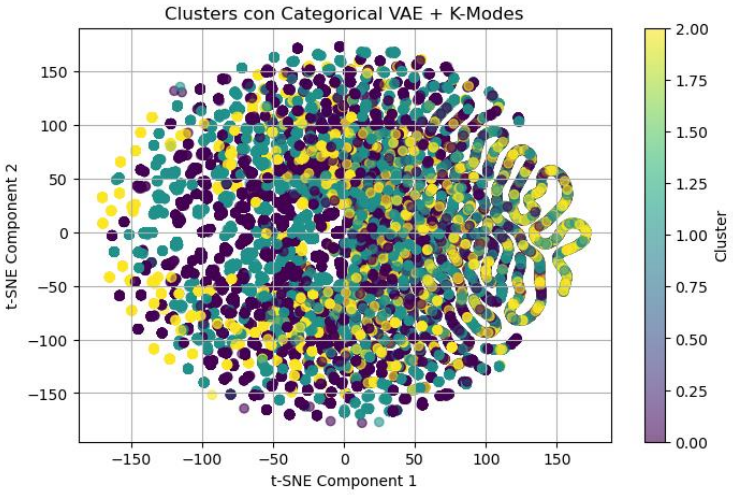
Clusters

La gráfica muestra la distribución de los estudiantes en el espacio de dos dimensiones generado por T-Sne después de aplicar el modelo de autoencoder variacional (vae) y la agrupación con K-Modes, cada punto representa un estudiante y los colores indican los diferentes clústeres asignados, hay puntos dispersos lo que indica solapamiento, hay áreas donde los puntos de un mismo color están más concentrados, lo que sugiere que el modelo logró identificar ciertos patrones en los datos, este análisis permite visualizar la estructura de los grupos.

Concluimos que Autoencoder variacional no logro separar de manera efectiva los datos antes de aplicar K-Modes como se evidencia en la Figura 47 *Clusters Autoencoder*

Figura 47

Clusters Autoencoder



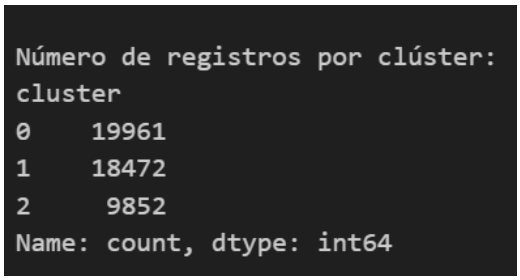
*Nota. * Fuente: desarrollo del autor*

La siguiente tabla muestra la distribución de 3 clusters que este algoritmo definió como se puede apreciar en la Figura 48

Distribución de Clusters Autoencoder

Figura 48

Distribución de Clusters Autoencoder



*Nota. * Fuente: desarrollo del autor*

Se muestra a continuación la descripción de los clusters generados por el algoritmo

CLÚSTER 0:

Grupo con 27,751 estudiantes, la mayoría de ellos pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 23.05 años con un rango de 16 a 68 años, el sexo más frecuente es hombre, el nivel académico más frecuente es primero, la parroquia de residencia más común es calderón, la mayoría reside en el sector norte, el estado civil predominante es soltero.

CLÚSTER 1:

Grupo con 8,394 estudiantes, la mayoría de ellos pertenecen a la modalidad presencial, la jornada más común es matutina, la edad promedio es de 23.72 años con un rango de 17 a 61 años, el sexo más frecuente es hombre, el nivel académico más frecuente es segundo, la parroquia de

residencia más común es Chillogallo, la mayoría reside en el sector sur, el estado civil predominante es soltero.

CLÚSTER 2:

Grupo con 12,140 estudiantes, la mayoría de los ellos pertenecen a la modalidad presencial, la jornada más común es nocturna, la edad promedio es de 24.95 años con un rango de 16 a 62 años, el sexo más frecuente es mujer, el nivel académico más frecuente es segundo, la parroquia de residencia más común es Cotocollao, la mayoría reside en el sector norte, el estado civil predominante es soltero.

4.7.Etapa de Evaluación

En esta etapa se realizó la evaluación de los modelos para verificar su desempeño y determinar si cumplen con los objetivos establecidos al inicio del proyecto, la evaluación se llevó a cabo mediante métricas cuantitativas y análisis cualitativos, considerando las características del problema y las necesidades de la IES.

4.8.Evaluación del desempeño

4.8.1. Modelo K-Means:

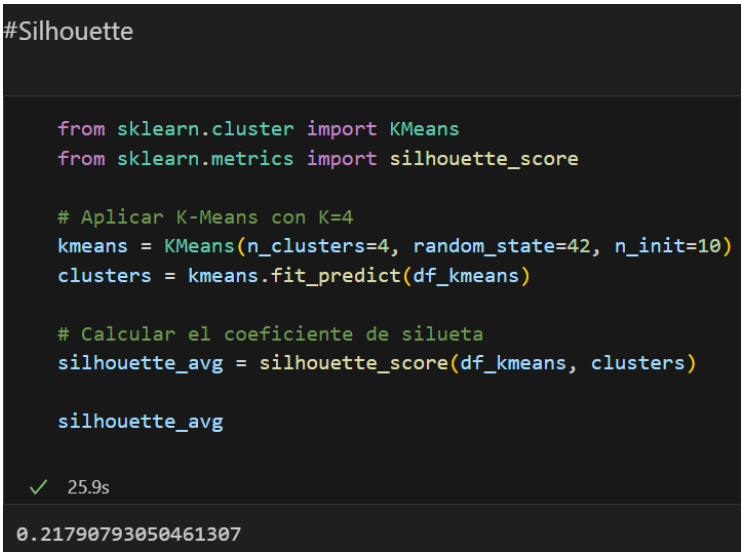
Se utilizó el Coeficiente de Silueta

Figura 49

Coeficiente de silueta K-Mean para evaluar la cohesión y separación de los clústeres, el valor obtenido se indicó una segmentación baja para el contexto del estudio.

Figura 49

Coeficiente de silueta K-Mean



*Nota. * Fuente: desarrollo del autor*

El coeficiente de silueta que se obtuvo es de 0.218, indicando una calidad de agrupación baja , esto muestra que los clústeres formados por el algoritmo K-Means presentan una separación no tan diferenciada entre ellos, se presenta solapamiento, el modelo no es tan funcional como para identificar patrones,

Pese que se definió un número apropiado de clústeres con el método del codo, se realizaron pruebas con reducción de dimensionalidad con PCA pero los resultados tampoco fueron muy buenos se obtuvo un 0.1885 de silueta como se puede apreciar en la

Figura 50

Coeficiente de silueta con PCA

Figura 50

Coeficiente de silueta con PCA

```
# Escalar las variables
scaler = StandardScaler()
df_pca[df_pca.select_dtypes(include=['number']).columns] = scaler.fit_transform(df_pca.select_dtypes(include=['number']))

# Aplicar PCA
pca = PCA(n_components=0.95)  # Retener el 95% de la varianza
pca_result = pca.fit_transform(df_pca)

# Aplicar K-Means con un número óptimo de clusters (ajustar si es necesario)
kmeans = KMeans(n_clusters=4, random_state=42, n_init=10)
labels = kmeans.fit_predict(pca_result)

# Calcular el coeficiente de silueta
silhouette_avg = silhouette_score(pca_result, labels)
print(f"Coeficiente de Silueta: {silhouette_avg}")
```

✓ 24.6s

```
Coeficiente de Silueta: 0.18851413262543523
```

*Nota. * Fuente: desarrollo del autor*

4.8.2. Modelo DBSCAN:

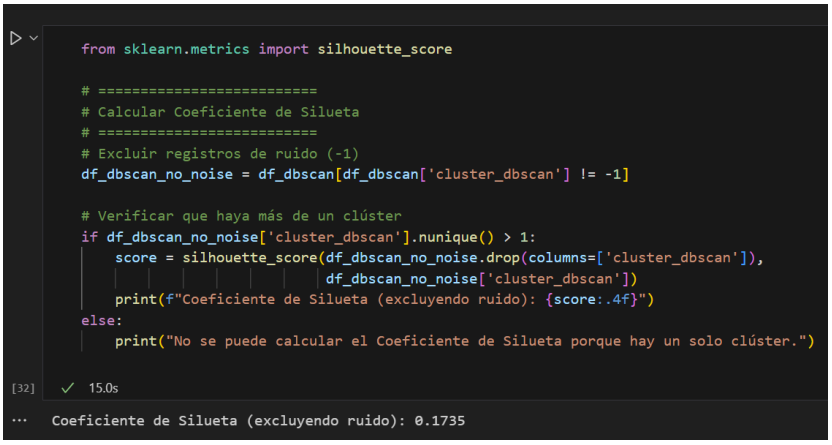
Se ajustaron los parámetros de densidad utilizando el gráfico de k-distances y pese a que se seleccionó un valor adecuado de EPS = 2.0 y un min_samples = 600, el modelo generó 11 clústeres después de excluir el ruido, con un Coeficiente de Silueta de 0.1735 como se muestra en la

Figura 51

Coeficiente de silueta DBSCAN ***¡Error! No se encuentra el origen de la referencia.***, indicando que la separación entre clústeres no es tan clara como con K-Means

Figura 51

Coeficiente de silueta DBSCAN



*Nota. * Fuente: desarrollo del autor*

Del valor obtenido por el coeficiente para DBSCAN de 0.1735 indica una calidad de agrupación muy baja, los clústeres no están diferenciados dentro de cada grupo, este valor más bien existe un solapamiento, el modelo identificó un total de 11 clústeres sin el de ruido, se detectaron 9297 registros de como ruido, esto representa el 15.50% del total de registros, este porcentaje puede afectar la cohesión y separación de los clústeres.

4.8.3. Modelo K-Prototypes

Se utilizó K-Prototypes debido a la naturaleza mixta de las variables categóricas y numéricas que se dispone, se logró una segmentación más representativa, los resultados obtenidos indican un índice de silueta de 0.5991 para las variables numéricas y una distancia de Hamming promedio de 0.5338 para las variables categóricas como se aprecia en la Figura 52 *K-Prototypes Evaluación*, lo que refleja una mejora radical y mejor diferenciación de los grupos.

El algoritmo identificó 4 clústeres principales, con diferencias significativas en modalidad, jornada y parroquia de residencia, la visualización a dos dimensiones de t-SNE mostró que los

clústeres están bien separados, existe muy baja superposición en algunos segmentos, este modelo permitió capturar de manera más precisa las relaciones entre las variables numéricas y categóricas, ofreciendo una segmentación útil para la interpretación de los segmentos.

Figura 52

K-Prototypes Evaluación

*Nota. * Fuente: desarrollo del autor*

4.8.4. Modelo K-Modes

Se utilizó el algoritmo K-Modes debido a que casi todas las variables del dataset son categóricas, transformando la edad, en rangos, el número óptimo de clústeres determinado fue 5, según el análisis del método del codo con el costo de inercia

La segmentación mostró cierta estructura en la agrupación de los datos, con factores como la modalidad, jornada y parroquia de residencia como las variables más influyentes en la formación de grupos, sin embargo, la visualización con T-Sne evidencia solapamiento entre los clústeres,

La diferenciación entre grupos no es muy clara, algunas categorías comparten muchas similitudes, lo que dificulta la correcta separación, la matriz de Cramer mostró que las asociaciones entre variables dentro de los clústeres no son lo suficientemente fuertes para generar grupos bien definidos, la distancia promedio de hamming obtenida fue de 0.5047, la disimilitud entre los valores categóricos dentro de los grupos no es lo suficientemente alta para asegurar una

segmentación bien diferenciada como se muestra en la Figura 53 *Evaluación K-Modes 1* y en la Figura 54 Evaluación K-Modes 2

Figura 53

Evaluación K-Modes 1

```
# Validar el modelo con la distancia de Hamming
hamming_distance = pairwise_distances(df_encoded.iloc[:, :-1], metric="hamming").mean()

# Mostrar la validación
print(f"Distancia promedio de Hamming: {hamming_distance}")
```

*Nota. * Fuente: desarrollo del autor*

Figura 54

Evaluación K-Modes 2

```
Distancia promedio de Hamming: 0.5004763941362437
```

*Nota. * Fuente: desarrollo del autor*

4.8.5. Modelo Autoencoder + Kmodes

Se implementó el algoritmo mixto Autoencoder variacional Vae combinado con k-modes para evaluar si un enfoque basado en redes neuronales mejoraba la segmentación de los estudiantes, aunque la representación latente optimizó la estructura de los datos, la distancia promedio de hamming de 0.5033 Figura 55 *Evaluación Autoencoder* indica que la separación entre clústeres no mejoró significativamente.

La visualización con T-Sne muestra una distribución más organizada en comparación con k-modes puro, pero aún con solapamiento, aunque el Autoencoder logró reducir la dimensionalidad, la segmentación no es completamente definida.

Figura 55

Evaluación Autoencoder

```
# Aplicar Label Encoding a todas las variables categóricas
df_encoded = df_cleaned.copy()

for col in df_cleaned.columns:
    le = LabelEncoder()
    df_encoded[col] = le.fit_transform(df_cleaned[col])

# Calcular la distancia promedio de Hamming
hamming_distance = pairwise_distances(df_encoded.iloc[:, :-1], metric="hamming").mean()

print(f"Distancia promedio de Hamming: {hamming_distance}")

✓ 42.8s

Distancia promedio de Hamming: 0.5033291579423935
```

Figura 56 Evaluación Autoencoder
***Fuente**: Desarrollo propio Pablo Caiza*

4.8.6. Comparación con los Objetivos del Negocio

El **modelo de K-Means** demostró no ser tan eficaz para la segmentación de los estudiantes debido a la transformación de todas las variables categóricas a numéricas, lo que hizo que la distancia euclidiana utilizada por el algoritmo no midiera distancias reales entre los datos, esto afectó su capacidad para formar grupos homogéneos y bien diferenciados, dificultando la interpretación de los resultados, aunque k-means es útil para identificar perfiles en datos puramente numéricos, en este caso la conversión de las variables categóricas generó una representación que no reflejaba con precisión la similitud entre los estudiantes, lo que limitó su efectividad en la segmentación y en la identificación de patrones claros.

El **modelo DBSCAN**, aunque es eficaz para detectar patrones en áreas de alta densidad y manejar datos atípicos presentó ciertas limitaciones en este contexto, Si bien logró identificar

grupos específicos, generó una gran cantidad de clústeres pequeños y dispersos dificultando la interpretación y aplicación práctica de los resultados de la segmentación.

el porcentaje significativo de registros clasificados como ruido complicó aún más la obtención de grupos representativos. estas características hacen que, para el propósito de segmentar a los estudiantes de manera clara y útil

El **modelo de K-Prototypes** demostró ser más adecuado para la segmentación de los estudiantes debido a su capacidad para manejar tanto variables categóricas como numéricas, logrando una segmentación más representativa con un índice de silueta de 0.5991 para las variables numéricas y una distancia promedio de hamming de 0.5338 para las categóricas, la visualización con T-Sne muestra que aunque existe baja superposición, la diferenciación entre los 4 clústeres es más clara en comparación con los otros algoritmos, facilita la interpretación de los resultados y permite identificar perfiles bien definidos de estudiantes

El **modelo de K-Modes,** aunque está diseñado específicamente para datos categóricos, presentó limitaciones en este estudio, a pesar de segmentar los datos en 5 clústeres, la visualización con T-Sne mostró un alto grado de solapamiento entre los grupos, dificultando la diferenciación de perfiles en la data de la IES, además, la distancia promedio de hamming indica que la disimilitud entre los valores categóricos dentro de los clústeres no fue lo suficientemente alta para generar una segmentación bastante clara, esto sugiere que K-Modes no es la mejor opción para segmentar a los estudiantes en este caso

El **modelo Autoencoder** combinado con K-Modes se implementó con la expectativa de que un enfoque basado en redes neuronales mejoraría la segmentación al generar una representación latente optimizada antes del clustering, sin embargo, la distancia promedio de

hamming indica que la separación entre los clústeres no mejoró significativamente respecto a los otros algoritmos, aunque el Autoencoder logró reducir la dimensionalidad de los datos y estructurar mejor la distribución de los clústeres, la visualización con T-Sne mostró que aún persiste el solapamiento, este enfoque no logró mejorar la segmentación.

4.8.7. Interpretación y Validación

Se realizaron análisis descriptivos de cada clúster para comprender las características predominantes en términos de edad, nivel académico, sexo, modalidad, jornada y sector de residencia, proporcionando perfiles claros de los diferentes grupos de estudiantes.

La visualización de los clústeres mediante gráficos de barras, boxplots y su validación por las métricas seleccionadas permitió segmentación y su alineación con las expectativas del proyecto.

4.8.8. Decisión de Modelo

Tras la evaluación de los cinco modelos abordados, se determinó en métricas y visualización que el modelo de K-Prototypes con 4 clústeres es el más adecuado para el contexto del estudio de segmentación en la IES, debido a su capacidad para manejar tanto variables categóricas como numéricas, logró una segmentación clara y diferenciada, su desempeño en la separación de los grupos, identificó perfiles más claros de los estudiantes que si bien no es completamente perfecto pero captura la naturaleza de la data y aporta a los objetivos del proyecto, se recomienda utilizar los resultados de este modelo para diseñar estrategias educativas personalizadas y mejorar los servicios ofrecidos a cada segmento de estudiantes en función de los clústeres obtenidos.

4.8.9. Análisis de Resultados

Descripción general de los Clústeres

Partiendo del análisis de clústeres realizado con el algoritmo K-Prototypes, se identificaron cuatro grupos bien definidos de estudiantes, cada uno presenta características específicas en cuanto a modalidad, jornada, edad, sexo y otras variables relevantes que permiten OBTENER perfiles claros dentro de la población estudiantil de la IES, se definen en este apartado los patrones y hallazgos encontrados así como los insights y sugerencias para cada grupo (cluster).

1. **Clúster 0 (Azul): Estudiantes Adultos con Preferencia Nocturna**

Patrones hallados del cluster

- Son estudiantes adultos, con una edad promedio notablemente más alta 40.87 años.
- Prefieren claramente la jornada nocturna.
- Hay una alta concentración en modalidades presenciales.
- La carrera que predomina es claramente Optometría.
- Se destaca un alto porcentaje de mujeres y estudiantes con estado civil casado.

Insights y sugerencias:

- Hay un perfil típico de estudiantes de edad adulta muy probablemente estudiantes que ya laboran o realizan algún tipo de actividad económica y que optan por estudiar en la noche en los horarios que no interfieren con actividades laborales.
- Se sugiere generar estrategias específicas como servicios de apoyo en horarios extendidos y programas dirigidos a adultos, quizá tutorías nocturnas o recomendar al estudiante modalidades híbridas que podría adaptarse a su perfil.

- Se sugiere considerar el tema de la seguridad principalmente en los días próximos al fin de semana ya que en horario nocturno se complica la seguridad en el sector.

2. Clúster 1 (Naranja): Jóvenes Adultos con Orientación Profesional Temprana

Patrones encontrados:

- Edad promedio de 24.30 años.
- Son mayormente hombres.
- Predomina claramente la jornada matutina y modalidad presencial.
- Carrera destacada: Administración Financiera.
- El estado civil predominante es soltero.

Insights y sugerencias:

- Este grupo representa los jóvenes adultos iniciando o finalizando su etapa de estudios, probablemente enfocados en integrarse rápidamente al mercado laboral dada su edad y al tiempo que van a estar en el instituto.
- Se sugiere fortalecer el proceso de titulación desde los niveles medios incentivando las prácticas profesionales y pasantías a tiempo.
- Se sugiere fortalecer los procesos de vinculación laboral temprana, como pasantías y buscar convenios estratégicos con empresas para la inserción, si bien es cierto se están realizando esfuerzos pero siendo un cluster bien definido se recomienda mejorar estos procesos.

3. Clúster 2 (Verde): Adultos Jóvenes con perfil nocturno con equilibrio Hombre/Mujer

Patrones hallados:

- Este grupo tiene una edad promedio intermedia 30.56 años
- Distribución de género equilibrada.
- Preferencia de estudios marcada por la jornada nocturna.
- Carrera predominante: Administración Financiera.
- Estado civil mayoritariamente soltero.

Insights y recomendaciones:

- Segmento que podría representar adultos jóvenes en una transición laboral o mejora profesional dada su edad.
- Se recomienda ofrecer programas flexibles con opciones híbridas y fortalecer habilidades específicas en administración financiera enfatizando en competencias laborales prácticas dentro de la carrera.

4. Clúster 3 (Rojo): Jóvenes en Etapa Inicial de estudios

Patrones encontrados:

- Edad promedio más baja que se sitúa en 19.95 años.
- Predominio notable de mujeres.
- Preferencia absoluta por la jornada matutina, modalidad presencial.

- Carrera predominante: Diseño Gráfico.
- Mayoritariamente solteros.

Insights y sugerencias:

- Representa claramente estudiantes recién ingresados o en etapas iniciales de estudios tecnológicos en la IES.
- Al ser el grupo más grande de la segmentación, se sugiere orientar esfuerzos hacia la adaptabilidad, hacia la prevención del abandono estudiantil, ofreciendo acompañamiento académico y psicológico, además de reforzar la orientación académica para mantener la retención en las etapas iniciales.
- Debido a que la carrera predominante es Diseño Gráfico se sugiere promover iniciativas estudiantiles y emprendimientos creativos o proyectos para la IES que aprovechen esta tendencia para beneficio del estudiante y la institución, el estudiante podría sentirse muy identificado con la IES al ser parte estos proyectos se han hecho pilotos aleatorios en este sentido pero dado la compactación de este grupo se deben mejorar los esfuerzos
- Los proyectos deben ser enfocados en fortalecer la equidad de género pero dada la predominancia de mujeres en este segmento se debe poner énfasis en sus necesidades académicas y psicológicas únicas.

Tendencias generales

- Se determinó con la segmentación un predominio de modalidad presencial, la mayoría absoluta de los estudiantes, más del 90% en todos los clústeres está en

modalidad presencial, sugiriendo que aún existe fuerte preferencia o necesidad de interacción presencial en el aprendizaje, se han abierto nuevas opciones pero el esfuerzo principal debe ser para esta modalidad.

- Sector geográfico común: Calderón se mantiene constante como la parroquia predominante en todos los clústeres, es una zona de relevancia estratégica geográfica y esto indica la posible necesidad de ampliar o fortalecer la oferta educativa en esta localidad realizando estrategias de promoción y captación.

- Variabilidad significativa en la edad y jornada: Existen diferencias claras en preferencias de horarios en función del rango de edad.

- La jornada nocturna prevalece claramente en adultos mayores (Cluster 0) y adultos jóvenes intermedios (Cluster 2)

- Mientras que matutina prevalece en jóvenes (Clusters 1 y 3).

Conclusiones Generales y sugerencias del resultado

- Segmentación Estratégica: se sugiere utilizar estos perfiles para segmentar campañas de comunicación y estrategias académicas adaptadas a cada segmento marcado de la IESS.

- Flexibilidad y adaptabilidad: se sugiere promover modalidades híbridas o en línea para clústeres adultos que requieren flexibilidad en sus horarios y garanticen su seguridad en la jornada nocturna.

- Servicios y soporte académico dirigido: se sugiere establecer servicios diferenciados según necesidades específicas, como horarios nocturnos de atención

en áreas como secretaría, gestión académica y bienestar estudiantil así como generar programas de retención específicos para los estudiantes jóvenes iniciales principalmente

- Fortalecer alianzas laborales: para los grupos profesionales tempranos, se sugiere desarrollar vínculos efectivos con empresas y redes profesionales, fortalecer proyectos como bolsa de empleo para su pronta inserción en el mercado laboral.

Contraste con los objetivos del estudio

Identificar grupos con características generales

La segmentación permitió identificar claramente los cuatro grupos homogéneos descritos, con perfiles específicos basados en modalidades jornadas, edad sexo, carrera y nivel académico y otras variables, cada grupo muestra particularidades demográficas y académicas que nos ayudaron en su caracterización y análisis.

Análisis de Patrones Históricos

Al analizar la información histórica de los últimos cinco años en los periodos descritos, se encontró estabilidad en ciertos patrones, como la prevalencia constante de la parroquia Calderón como lugar principal de residencia, lo que indica una demanda educativa sostenida en esta área, la selección de modalidades y jornadas fue un patrón importante.

Factores Diferenciadores por Parroquia/Sector, Carrera, Modalidad y Sexo

Los análisis revelaron factores diferenciadores en función de estas variables

- Parroquia/Sector: Calderón es la más común en todos los grupos, sugiriendo de forma concluyente una influencia fuerte de esta área en la demanda educativa.
- Carreras: Diferencias marcadas por grupo, desde Optometría en adultos mayores hasta Diseño Gráfico en estudiantes más jóvenes.
- Modalidad: La presencialidad es muy dominante, aunque existe una necesidad emergente en modalidades flexibles para ciertos grupos.
- Sexo: Diferencias significativas, con presencia importante femenina en grupos jóvenes y adulta, y masculina en jóvenes adultos próximos a la inserción laboral.

Distribución para la Mejora en Planificación de Recursos

El análisis detallado de la distribución por jornada, edad, modalidad y carreras permitirá a la IES asignar recursos para:

- Implementación de posibles estrategias de retención promoción, movilización de personal a laboral en jornadas de estudiantes nocturnos, proyectos para las carreras más representativas.
- Reforzar o replantear horarios nocturnos para estudiantes adultos.
- Intensificar servicios de orientación académica para estudiantes jóvenes en etapa inicial.
- Fortalecer la infraestructura física, tecnológica, académica en las modalidades presencial e híbrida para potenciarla y promocionarla.

Provisión de Información Estratégica

Los resultados brindan información muy valiosa que sugiere decisiones como:

- Diseñar campañas segmentadas de comunicación y promoción dirigidas específicamente a cada perfil que se identificó en los segmentos en términos de ubicación geográfica, modalidad, jornada y sexo.
- Implementar estrategias académicas adaptados a las características detectadas en cada clúster.
- Optimizar la oferta educativa según demandas geográficas y demográficas específicas.

El análisis realizado mediante técnicas de ciencia de datos utilizando la metodología CRISPDM iterativa y aplicando un algoritmo de clusterización como K-Prototypes, cumple exitosamente los objetivos del estudio planteados en estudio, y proporciona la IES insights valiosos para una toma de decisiones basada en evidencia técnica y sólida permitiendo una planificación eficiente y estratégica en búsqueda de mejora continua.

5. CAPITULO V: CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presenta las conclusiones y recomendaciones obtenidas del análisis realizado en el marco de la investigación planteada y el objetivo general, en el cual se quiso abordar la segmentación de estudiantes desde diferentes enfoques utilizando algoritmos de clustering muy conocidos, como K-Means, DBSCAN, K-Modes, K-Prototypes y una combinación de Autoencoder con K-Modes, este enfoque desde varias aristas permitió evaluar la eficacia y la aplicabilidad de cada algoritmo, analizando los resultados de cada uno en términos de claridad, diferenciación y utilidad para la toma decisiones dentro de la IES.

5.1.Conclusiones

- El modelo más adecuado para segmentar a los estudiantes de la IES según los resultados de la evaluación es el algoritmo K-Prototypes, es el algoritmo más efectivo dada su capacidad de combinación de variables mixtas entre categóricas y numéricas de forma natural, esto permitió una segmentación clara y significativa con un índice de silueta satisfactorio y una distancia de haming aceptable

- Aunque el Algoritmo K-Means mostro resultados moderados bajos, la transformación obligatoria de variables categóricas a numéricas ocasiono limitaciones al tratar de medir distancias reales entre los datos de los estudiantes, afectando su capacidad de crear clústeres homogéneos y diferenciados

- El algoritmo DBSCAN tuvo limitaciones muy significativas con un coeficiente de silueta bajo 0.067 indicando que los clusters tienden a solaparse de forma considerable haciendo que la interpretación de los resultados sea muy compleja.

- K-Modes y la combinación de Autoencoder + K-Modes tampoco lograron una mejor segmentación ya que mostraron un grado alto de solapamiento y los clústeres no están bien definidos con una distancia de haming cercana a 0.5 limitando su utilidad para identificar perfiles claros.

- El análisis de la segmentación permitió extraer información importante contenida en los datos institucionales, se halló más que patrones, permitiendo una compresión más profunda de los estudiantes desde muchas dimensiones, en preferencias académicas, características demográficas y necesidades.

5.2.Recomendaciones

- Implementar el modelo K-Prototypes como el algoritmo principal para futuras segmentaciones, dada su capacidad para encontrar patrones en la data mixta entre variables categóricas y numéricas.

- Realizar un seguimiento periódico a los segmentos que generó K-Prototypes para garantizar la estabilidad y relevancia de los perfiles esto de cara a los cambios académicos naturales dentro de la IES y a los cambios demográficos de la población estudiantil.

- Para la estructura actual de la data disponible se recomienda evitar algoritmos como K-Means, DBSCAN y K-Modes en futuras investigaciones donde predominen las variables categóricas o mixtas, debido a las limitaciones en términos del clustering y la calidad de los grupos que se forman con estos modelos.

- Se deben explorar técnicas avanzadas como modelos híbridos o redes neuronales especificas en futuras investigaciones para profundizar precisión y claridad dado

que se espera una mejora continua en la segmentación adaptándola periódicamente a la realidad académica, demográfica.

- Se evidencia en los clústeres la parroquia con más estudiantes son del sector norte y específicamente en la parroquia de calderón por lo que se recomienda prestar atención a este hallazgo ya que este sector aporta presencia en la composición demográfica estudiantil, pese a que este sector tiene varias instituciones tenemos preferencia en este sector y se evidencia en la estructura de los clusters

- Se recomienda diseñar estrategias de apoyo académico y fortalecimiento de la seguridad a los estudiantes del cluster de 1941 registros que son de la modalidad presencial en la jornada nocturna principalmente en los primeros niveles y de estado civil casados que representan el segmento más pequeño pero engloba a estudiantes maduros que posiblemente trabajan y pueden ser un grupo vulnerable por la situación de seguridad que atraviesa el país.

- Se recomienda fortalecer programas de inducción, adaptación y retención enfocados a los estudiantes jóvenes de modalidad presencial en la jornada matutina con mayor representatividad del sexo femenino y con estado civil soltero que representa el cluster más numeroso con 24205 registros este segmento es muy importante para la institución.

- Utilizar los resultados obtenidos con la segmentación para desarrollar estrategias educativas y administrativas personalizadas y efectivas, aprovechando el conocimiento obtenido por la segmentación hallado desde una perspectiva multidimensional en los datos de los estudiantes.

- En cuanto a la gestión de la data se pudo evidencia que se deben realizar procesos de actualización un poco más frecuentes para que datos ubicación sean más precisos, dada la naturaleza de los cambios de domicilio, este proceso si bien es cierto esta implementado en la gestión del portal estudiantil pero se puede mejorar en términos de frecuencia, buscando métodos que no sean vistos como invasivos para el estudiante sino más bien enfocándolos en propuestas de valor permitiendo ofertar servicios diferenciados.

- Se recomienda a futuro incluir variables de tipo socioeconómico y de análisis de deserción para complementar y mejorar el modelo actual

- Se recomienda estudiar y diseñar estrategias en función del análisis de resultados de la segmentación para mejorar el servicio educativo llevándolo a un nivel diferenciado

BIBLIOGRAFÍA

Academia 4.0. (s.f.). *Academia 4.0*. Obtenido de https://www.a4.cr/post/8-grandes-empresas-que-usan-python

Amazon. (s.f.). *Amazon Python*. Obtenido de https://aws.amazon.com/es/what-is/python/

Arsys. (2024). *Arsys*. Obtenido de https://www.arsys.es/blog/python-que-es-y-para-que-sirve

Bonthu, H. (29 de Mayo de 2021). *harikabonthu*. Obtenido de harikabonthu: https://harikabonthu96.medium.com/kmodes-clustering-2286a9bfdcfb

Bowen, B. (2024). https://revistacodigocientifico.itslosandes.net/index.php/1/article/view/406/893. *Código Científico*, 748-749.

Caces. (Mayo de 2024). *Modelo de Evaluación Externa 2024 fon fines de acreditación para los insitutos superiores ténicos y tecnológicos*. Obtenido de https://www.caces.gob.ec/: https://www.caces.gob.ec/

Carmateck. (2023). Obtenido de https://www.carmatec.com/es_mx/blog/20-mejores-bibliotecas-de-python-para-aprendizaje-automatico/

Chambi, P. (14 de Septiembre de 2022). *Revistas de investigación Universidad Nacional de San Marcos Peru.* Obtenido de Revistas de investigación Universidad Nacional de San Marcos Peru: https://revistasinvestigacion.unmsm.edu.pe/index.php/idata/article/view/23623/20103

Contentsquare. (19 de Agosto de 2024). *Contentsquare*. Obtenido de https://contentsquare.com/es-es/guias/segmentacion-de-clientes/

DataCamp. (2024). *DataCamp*. Obtenido de https://www.datacamp.com/es/blog/all-about-python-the-most-versatile-programming-language

DataCamp Used. (2023). *DataCamp*. Obtenido de https://www.datacamp.com/es/blog/what-is-python-used-for

Ester, M. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)* (págs. 226-231). Portland, Oregon: AAAI Press.

FasterCapital. (s.f.). *FasterCapital*. Obtenido de FasterCapital: https://fastercapital.com/es/tema/%C2%BFc%C3%B3mo-medir-la-calidad-y-validez-de-los-resultados-del-clustering.html

Fernandez, R. (2001). Segmentación de mercados, buyscando la correlación entre variables sicológicas y demográficas. *redalyc.org*, 3.

Godaddy. (2023). *Godaddy*. Obtenido de https://www.godaddy.com/resources/latam/desarrollo/python-que-es

Hemsley-Brown. (2020). Higher Education Market Segmentation. En Hemsley-Brown, *Higher Education Market Segmentation* (págs. 711-713). Dordrecht, Países Bajos: Springer.

Huang, Z. (1998). Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Mining and Knowledge Discovery*, 283-304.

Hunter, J. D. (2007). *Ieeexplore*. Obtenido de https://ieeexplore.ieee.org/document/4160265

IBM. (2024). *IBM*. Obtenido de https://www.ibm.com/mx-es/topics/k-means-clustering

IoT Consulting. (2017). Obtenido de https://iotconsulting.tech/por-que-python-gana-popularidad-en-proyectos-de-iot/

Isea. (2018). Obtenido de https://mariafatimadossantosestadistica1.wordpress.com/wp-content/uploads/2018/06/coeficientes-v-de-cramer-y-c-de-pearson.pdf

Llamas, L. (16 de Julio de 2024). *2024*. Obtenido de Luis Llamas: https://www.luisllamas.es/python-indentacion-comentarios/

Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory.*, 129-137.

McKinney, W. (2010). *Proceedings Scipy*. Obtenido de https://proceedings.scipy.org/articles/Majora-92bf1922-00a

Nodd3r. (s.f.). *Nodd3r*. Obtenido de https://nodd3r.com/blog/por-que-se-utiliza-python-en-la-ciencia-de-datos

Nuclio Digital School. (2023). *Nuclio Digital School*. Obtenido de https://nuclio.school/blog/automatizar-tareas-con-python/

NumPy. (2020). *NumPy*. Obtenido de https://numpy.org/

Pedregosa, F. (2011). Obtenido de https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

Python Software Foundation. (2020). *Python Software Foundation*. Obtenido de https://docs.python.org/3.9/whatsnew/

Python Software Foundation. (2024). *Python Documentation*. Obtenido de https://docs.python.org/es/3.13/tutorial/index.html

Ramirez, L. (05 de Enero de 2023). *IEBS*. Obtenido de Iebs School: https://www.iebschool.com/blog/algoritmo-k-means-que-es-y-como-funciona-big-data/

Rodriguez, Daniel. (23 de Junio de 2023). *Analytics Lane*. Obtenido de Analytics Lane: https://www.analyticslane.com/2023/06/23/numero-optimo-de-clusteres-con-silhouette-e-implementacion-en-python/

Rodriguez, Daniel. (9 de Junio de 2023). *Analytics Lane*. Obtenido de Analytics Lane: https://www.analyticslane.com/2023/06/09/metodo-del-codo-elbow-method-para-seleccionar-el-numero-optimo-de-clusteres-en-k-means/

Rodriguez, Daniel Davies-Bouldinen. (30 de Junio de 2023). *Analytics Lane*. Obtenido de Analytics Lane.

Sanz Angulo, P. (2019). Obtenido de http://uvadoc.uva.es/handle/10324/37908

Scikit-learn. (2022). *Scikit-learn*. Obtenido de https://scikit-learn.org/stable/

Sevilla, U. d. (Enero de 2024). *Departamento de ciencias de la computación e inteligencia artificial*. Obtenido de Departamento de ciencias de la computación e inteligencia artificial: https://www.cs.us.es/~fsancho/Blog/posts/Clustering/

SISE. (2024). *SISE*. Obtenido de https://www.sise.edu.pe/blog/caracteristicas-principales-python

Tecnológico Universitario Cordillera. (Febrero de 2024). Obtenido de https://www.cordillera.edu.ec/wp-content/uploads/2024/02/Registro-Institucional-comprimido.pdf

Torres, A. (9 de diciembre de 2021). *freeCodeCamp*. Obtenido de https://www.freecodecamp.org/espanol/news/limpieza-de-datos-en-pandas-explicado-con-ejemplos/

Unite.AI. (2023). Obtenido de https://www.unite.ai/es/10-mejores-software-de-aprendizaje-autom%C3%A1tico/

Verne Academy. (2019). *Verne Academy*. Obtenido de https://verneacademy.com/blog/articulos-ia/10-librerias-python-data-science-machine-learning/

ANEXO 1

Detalle de código de Modelo

dbscan

March 19, 2025

[23]:
```
import pandas as pd
import plotly.graph_objects as go

# Cargar el archivo estudiantes_cleaned.xlsx
df_cleaned = pd.read_excel('estudiantes_cleaned.xlsx', engine='openpyxl')
df_cleaned.head()
```

[23]:

	periodo	escuela	carrera	modalidad	nivel	jornada	sexo	pais_residencia	provincia_residencia	canton_residencia	parroquia_residencia	sector_domicilio	fecha_nacimiento	edad	colegio	estado_civil
0	abr 2020_sep 2020	diseño grafico	diseño grafico	presencial	sexto	fin de semana	hombre	ecuador	pichincha	quito	carcelen	norte	1980-12-28	39	colegio internacional rudolf steiner	soltero
1	abr 2020_sep 2020	analistas de sistemas	tecnologia superior en desarrollo de software …	presencial	tercero	nocturna	hombre	ecuador	pichincha	quito	san antonio de pichincha	valles				
2	abr 2020_sep 2020	administracion recursos humanos - personal	tecnologia superior en gestion de talento huma…	presencial	primero	matutina	hombre	ecuador	pichincha	quito	carcelen	norte				
3	abr 2020_sep 2020	diseño grafico	diseño grafico	presencial	sexto	fin de semana	hombre	ecuador	pichincha	quito	calderon	norte				
4	oct 2022_mar 2023	administracion de boticas y farmacias	tecnologia superior en asistencia en farmacia …	presencial	segundo	nocturna	hombre	ecuador	pichincha	quito	carcelen	norte				

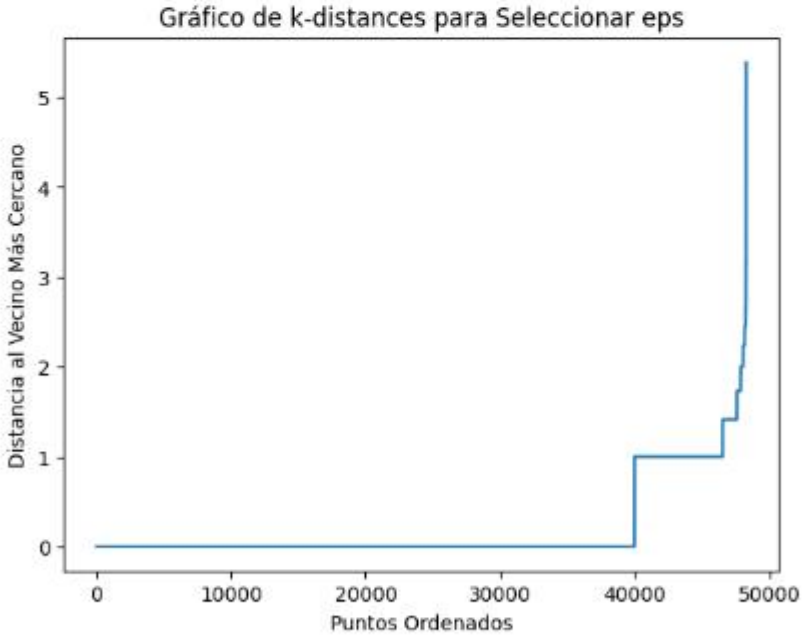
1	1973-11-05	46	colegio particular italia	divorciado
2	1989-04-21	30	no asignado	soltero
3	1989-01-20	31	colegio miguel angel asturias	soltero
4	1979-11-13	42	theodore w anderson	divorciado

0.1 Transformaciones

[25]:
```
from sklearn.preprocessing import StandardScaler
df_dbscan = df_cleaned.copy();
# Eliminar columnas irrelevantes
columns_to_drop = ["periodo", "escuela", "carrera", "pais_residencia",
                   "provincia_residencia", "canton_residencia",
                   "colegio", "fecha_nacimiento"]
df_dbscan.drop(columns=columns_to_drop, inplace=True)

# Codificación ordinal para 'nivel'
nivel_order = ["primero", "segundo", "tercero", "cuarto", "quinto", "sexto",
               "septimo", "octavo", "noveno", "decimo"]
df_dbscan["nivel"] = pd.Categorical(df_cleaned["nivel"],␣
 ↪categories=nivel_order, ordered=True).codes + 1

# Convertir 'sexo' a valores numéricos
df_dbscan["sexo"] = df_dbscan["sexo"].map({"hombre": 0, "mujer": 1})

# Escalar solo las columnas numéricas
scaler = StandardScaler()
df_dbscan[["edad", "nivel"]] = scaler.fit_transform(df_dbscan[["edad",␣
 ↪"nivel"]])

# Codificación para variables categóricas
categorical_cols = ["modalidad", "jornada", "parroquia_residencia",␣
 ↪"sector_domicilio", "estado_civil"]
for col in categorical_cols:
    df_dbscan[col] = df_dbscan[col].astype("category").cat.codes

# Convertir todos los valores booleanos/categóricos a enteros
df_dbscan = df_dbscan.applymap(lambda x: int(x) if isinstance(x, (bool, int,␣
 ↪float)) else x)
```

```
C:\Users\Usuario\AppData\Local\Temp\ipykernel_25936\1444387864.py:27:
FutureWarning: DataFrame.applymap has been deprecated. Use DataFrame.map
instead.
  df_dbscan = df_dbscan.applymap(lambda x: int(x) if isinstance(x, (bool, int,
float)) else x)
```

[26]:
```python
from sklearn.neighbors import NearestNeighbors
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN

# ==========================
# Encontrar el Mejor eps (Método k-distances)
# ==========================
nearest_neighbors = NearestNeighbors(n_neighbors=5)
nearest_neighbors.fit(df_dbscan)
distances, indices = nearest_neighbors.kneighbors(df_dbscan)

# Ordenar distancias y graficar
distances = np.sort(distances[:, -1])
plt.plot(distances)
plt.title('Gráfico de k-distances para Seleccionar eps')
plt.xlabel('Puntos Ordenados')
plt.ylabel('Distancia al Vecino Más Cercano')
plt.show()
```



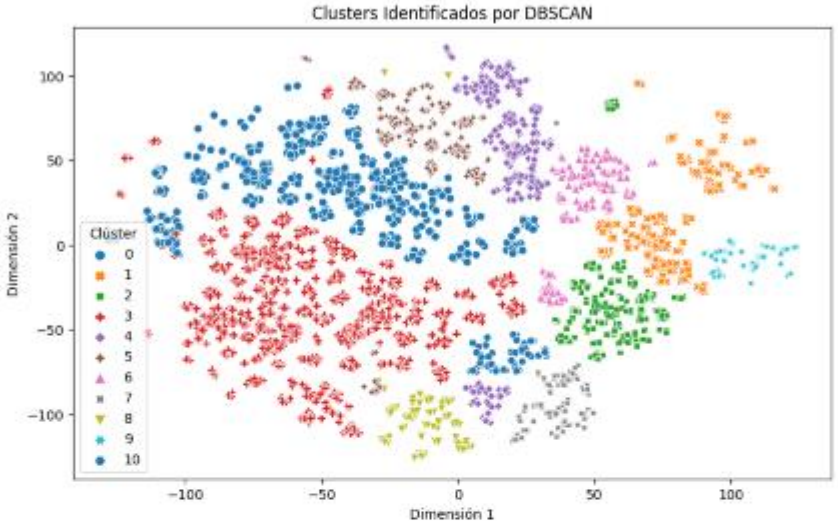
#Modelo DBSCAN

[27]:
```
dbscan = DBSCAN(eps=2.0, min_samples=600)  # Variables determinadas
df_dbscan['cluster_dbscan'] = dbscan.fit_predict(df_dbscan)

cluster_counts = df_dbscan['cluster_dbscan'].value_counts().sort_index()
#cluster_counts = df_dbscan['cluster_dbscan'].value_counts()
print("Número de registros por clúster:")
print(cluster_counts)
```

```
Número de registros por clúster:
cluster_dbscan
-1      9297
 0     10173
 1      2813
 2      2598
 3     12222
 4      2771
 5      2816
 6      1889
 7      1132
 8      1084
 9       740
 10      750
Name: count, dtype: int64
```

[28]:
```
from sklearn.manifold import TSNE
import seaborn as sns
import matplotlib.pyplot as plt

# ==========================
# Reducción de Dimensionalidad con t-SNE
# ==========================
tsne = TSNE(n_components=2, random_state=42)
df_dbscan[['tsne1', 'tsne2']] = tsne.fit_transform(df_dbscan.
  ↪drop(columns=['cluster_dbscan']))

# ==========================
# Filtrar Solo los 10 Clústeres (Excluir Ruido -1)
# ==========================
df_clusters = df_dbscan[df_dbscan['cluster_dbscan'].isin(range(0, 11))]

# ==========================
# Gráfico de los Clústeres
# ==========================
plt.figure(figsize=(10, 6))
sns.scatterplot(data=df_clusters, x='tsne1', y='tsne2', hue='cluster_dbscan',␣
  ↪palette='tab10', style='cluster_dbscan')
```

```
plt.title('Clusters Identificados por DBSCAN')
plt.xlabel('Dimensión 1')
plt.ylabel('Dimensión 2')
plt.legend(title='Clúster')
plt.show()
```



#Reversar para interpretar

[29]:
```
# Crear una copia del dataset original
df_original = df_cleaned.copy()

# Asignar la columna de clúster del dataset transformado al dataset original
df_original["cluster"] = df_dbscan['cluster_dbscan']

# Excluir el cluster de ruido (-1)
df_original_sin_ruido = df_original[df_original['cluster'] != -1]

# Guardar el dataset sin ruido
df_original_sin_ruido.to_excel("df_dbscan_sin_ruido.xlsx", index=False,␣
  ↪engine="openpyxl")

# Verificar las primeras filas del nuevo DataFrame
df_original_sin_ruido.head()
```

[29]:

```
            periodo                                    escuela  \
0  abr 2020_sep 2020                          diseño grafico
2  abr 2020_sep 2020  administracion recursos humanos - personal
3  abr 2020_sep 2020                          diseño grafico
4  oct 2022_mar 2023       administracion de boticas y farmacias
5  abr 2023_sep 2023       administracion de boticas y farmacias

                                        carrera   modalidad    nivel  \
0                                diseño grafico  presencial    sexto
2  tecnologia superior en gestion de talento huma…  presencial  primero
3                                diseño grafico  presencial    sexto
4  tecnologia superior en asistencia en farmacia …  presencial  segundo
5     asistencia en farmacia - 2250-550916a01-p-1701  presencial   cuarto

         jornada    sexo pais_residencia provincia_residencia  \
0  fin de semana  hombre         ecuador            pichincha
2       matutina  hombre         ecuador            pichincha
3  fin de semana  hombre         ecuador            pichincha
4       nocturna  hombre         ecuador            pichincha
5       nocturna  hombre         ecuador            pichincha

  canton_residencia parroquia_residencia sector_domicilio fecha_nacimiento  \
0             quito             carcelen            norte       1980-12-28
2             quito             carcelen            norte       1989-04-21
3             quito             calderon            norte       1989-01-20
4             quito             carcelen            norte       1979-11-13
5             quito             carcelen            norte       1979-11-13

   edad                              colegio estado_civil  cluster
0    39  colegio internacional rudolf steiner      soltero        0
2    30                          no asignado      soltero        0
3    31         colegio miguel angel asturias      soltero        0
4    42                  theodore w  anderson   divorciado        0
5    43                  theodore w  anderson   divorciado        0
```

#resumen de clusters

[30]:

```
# Crear un resumen detallado para cada clúster usando df_original_sin_ruido
cluster_details = []

# Excluir el cluster de ruido (-1)
df_original_sin_ruido = df_original[df_original['cluster'] != -1]

for cluster_id in df_original_sin_ruido['cluster'].unique():
    cluster_data = df_original_sin_ruido[df_original_sin_ruido['cluster'] ==␣
 ↪cluster_id]
    detail = {
```

```
        'Cluster': cluster_id,
        'Edad promedio': cluster_data['edad'].mean(),
        'Edad mínima': cluster_data['edad'].min(),
        'Edad máxima': cluster_data['edad'].max(),
        'Nivel más frecuente': cluster_data['nivel'].mode().iloc[0],
        'Sexo más frecuente': cluster_data['sexo'].mode().iloc[0],
        'Modalidad más frecuente': cluster_data['modalidad'].mode().iloc[0],
        'Jornada más frecuente': cluster_data['jornada'].mode().iloc[0],
        'Parroquia más frecuente': cluster_data['parroquia_residencia'].mode().
 ↪iloc[0],
        'Sector de domicilio más frecuente': cluster_data['sector_domicilio'].
 ↪mode().iloc[0],
        'Estado civil más frecuente': cluster_data['estado_civil'].mode().
 ↪iloc[0],
        'Cantidad de registros': len(cluster_data),
        'Porcentaje del total': len(cluster_data) / len(df_original_sin_ruido)␣
 ↪* 100
    }
    cluster_details.append(detail)

# Convertir la lista de detalles en un DataFrame
df_cluster_details = pd.DataFrame(cluster_details)

# Mostrar el DataFrame final
df_cluster_details

# Guardar el DataFrame en un archivo Excel
#df_cluster_details.to_excel("df_cluster_details_dbscan.xlsx", index=False,␣
 ↪engine="openpyxl")
```

[30]:

	Cluster	Edad promedio	Edad mínima	Edad máxima	Nivel más frecuente
0	0	23.080900	16	46	primero
1	1	23.376822	17	42	quinto
2	4	23.736918	17	45	primero
3	7	23.318021	17	40	quinto
4	6	22.881948	17	45	primero
5	3	23.227213	16	50	primero
6	5	22.535511	17	43	primero
7	2	23.291763	17	44	primero
8	9	23.121622	17	39	primero
9	8	23.146679	17	39	primero
10	10	22.717333	17	34	primero

	Sexo más frecuente	Modalidad más frecuente	Jornada más frecuente
0	mujer	presencial	matutina
1	mujer	presencial	matutina
2	mujer	presencial	matutina

3	hombre	presencial	matutina
4	mujer	presencial	matutina
5	mujer	presencial	matutina
6	mujer	presencial	matutina
7	hombre	presencial	matutina
8	hombre	presencial	matutina
9	mujer	presencial	matutina
10	mujer	presencial	matutina

	Parroquia más frecuente	Sector de domicilio más frecuente
0	calderon	norte
1	san antonio de pichincha	norte
2	kennedy	norte
3	la magdalena	sur
4	pomasqui	norte
5	cotocollao	norte
6	el condado	norte
7	quito	sur
8	solanda	sur
9	guamani	sur
10	la ferroviaria	sur

	Estado civil más frecuente	Cantidad de registros	Porcentaje del total
0	soltero	10173	26.092644
1	soltero	2813	7.215041
2	soltero	2771	7.107315
3	soltero	1132	2.903457
4	soltero	1889	4.845081
5	soltero	12222	31.348107
6	soltero	2816	7.222735
7	soltero	2598	6.663589
8	soltero	740	1.898020
9	soltero	1084	2.780343
10	soltero	750	1.923669

#Evaluación de Silouetthe Dbscan

[32]:
```
from sklearn.metrics import silhouette_score

# ==========================
# Calcular Coeficiente de Silueta
# ==========================
# Excluir registros de ruido (-1)
df_dbscan_no_noise = df_dbscan[df_dbscan['cluster_dbscan'] != -1]

# Verificar que haya más de un clúster
if df_dbscan_no_noise['cluster_dbscan'].nunique() > 1:
```

```
    score = silhouette_score(df_dbscan_no_noise.
 ↪drop(columns=['cluster_dbscan']),
                             df_dbscan_no_noise['cluster_dbscan'])
    print(f"Coeficiente de Silueta (excluyendo ruido): {score:.4f}")
else:
    print("No se puede calcular el Coeficiente de Silueta porque hay un solo␣
 ↪clúster.")
```

```
Coeficiente de Silueta (excluyendo ruido): 0.1735
```

kprototypes

March 19, 2025

#kprototype

[1]:
```
import os
print(os.getcwd())
```

```
d:\Pablo\Maestria\Proyecto de
Titulación\DatosDatasetDefinitivo\localdatakprototype
```

[2]:
```
# Importar librerías necesarias
import pandas as pd
from sklearn.preprocessing import StandardScaler
# Cargar el dataset proporcionado
file_path = "estudiantes_cleaned.xlsx"
df_cleaned = pd.read_excel(file_path)
df_cleaned.head()
```

[2]:

	estudiante_id	periodo_id	periodo	escuela_id	escuela	carrera_id	carrera	modalidad	nivel	jornada	…	pais_residencia	provincia_residencia	canton_residencia
0	64	74	abr 2020_sep 2020	2	diseño grafico	4	diseño grafico	presencial	sexto	matutina	…	ECUADOR	PICHINCHA	QUITO
1	316	74	abr 2020_sep 2020	7	analistas de sistemas	87	tecnologia superior en desarrollo de software …	presencial	tercero	matutina	…	ECUADOR	PICHINCHA	QUITO
2	551	74	abr 2020_sep 2020	10	administracion recursos humanos - personal	91	tecnologia superior en gestion de talento huma…	presencial	primero	matutina	…	ECUADOR	PICHINCHA	QUITO
3	558	74	abr 2020_sep 2020	2	diseño grafico	4	diseño grafico	presencial	sexto	matutina	…	ECUADOR	PICHINCHA	QUITO
4	618	79	oct 2022_mar 2023	5	administracion de boticas y farmacias	95	tecnologia superior en asistencia en farmacia …	presencial	segundo	matutina	…	ECUADOR	PICHINCHA	QUITO

```
0  fin de semana  …         ecuador            pichincha             quito
1       nocturna  …         ecuador            pichincha             quito
2       matutina  …         ecuador            pichincha             quito
3  fin de semana  …         ecuador            pichincha             quito
4       nocturna  …         ecuador            pichincha             quito

       parroquia_residencia sector_domicilio fecha_nacimiento edad  \
0                  carcelen            norte       1980-12-28   39
1  san antonio de pichincha           valles       1973-11-05   46
2                  carcelen            norte       1989-04-21   30
3                  calderon            norte       1989-01-20   31
4                  carcelen            norte       1979-11-13   42

   colegio_id                              colegio estado_civil
0        1891  colegio internacional rudolf steiner      soltero
1         909             colegio particular italia   divorciado
2          -1                          no asignado      soltero
3         440        colegio miguel angel asturias      soltero
4        2383                 theodore w  anderson   divorciado

[5 rows x 21 columns]
```

[3]:
```
df_cleaned.shape
```

[3]:
```
(48285, 21)
```

#Transformaciones

[4]:
```
# Importar librerías necesarias
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Copiar el dataset original para trabajar sobre él
df_kproto = df_cleaned.copy()

# Eliminar columnas irrelevantes
columns_to_drop =␣
 ↪["estudiante_id","periodo_id","escuela_id","carrera_id","colegio_id","periodo",␣
 ↪"escuela", "carrera", "pais_residencia",
                "provincia_residencia", "canton_residencia",␣
 ↪"fecha_nacimiento", "colegio"]
df_kproto.drop(columns=columns_to_drop, inplace=True)
df_kproto.to_excel("df_kproto.xlsx", index=False)
```

[5]:
```
df_kproto.head()
df_kproto.shape
```

[5]: (48285, 8)

#Análisis de variables

[8]:
```
import pandas as pd
import numpy as np
import scipy.stats as ss

# Configurar pandas para mostrar todas las columnas y filas sin cortar
pd.set_option("display.max_rows", None)
pd.set_option("display.max_columns", None)
pd.set_option("display.width", 1000)

# Utilizar df_kproto en lugar de leer un archivo
categorical_cols_kproto = df_kproto.select_dtypes(include=['object']).columns

# Función para calcular Cramér V entre dos variables categóricas
def cramers_v(x, y):
    """Calcula la correlación de Cramér V entre dos variables categóricas."""
    confusion_matrix = pd.crosstab(x, y)
    chi2 = ss.chi2_contingency(confusion_matrix)[0]
    n = confusion_matrix.sum().sum()
    phi2 = chi2 / n
    r, k = confusion_matrix.shape
    phi2corr = max(0, phi2 - ((k-1)*(r-1)) / (n-1))
    r_corr = r - ((r-1)**2) / (n-1)
    k_corr = k - ((k-1)**2) / (n-1)
    return np.sqrt(phi2corr / min(r_corr-1, k_corr-1))

# Crear la matriz de Cramér V
num_categorical_kproto = len(categorical_cols_kproto)
cramers_v_matrix_kproto = pd.DataFrame(np.zeros((num_categorical_kproto,␣
↪num_categorical_kproto)),
                                       index=categorical_cols_kproto,␣
↪columns=categorical_cols_kproto)

for col1 in categorical_cols_kproto:
    for col2 in categorical_cols_kproto:
        if col1 != col2:
            cramers_v_matrix_kproto.loc[col1, col2] =␣
↪cramers_v(df_kproto[col1], df_kproto[col2])
        else:
            cramers_v_matrix_kproto.loc[col1, col2] = 1.0  # La correlación␣
↪consigo misma es 1

# Imprimir la matriz completa sin que se divida
print(cramers_v_matrix_kproto)
```

```
                     modalidad     nivel   jornada      sexo
parroquia_residencia  sector_domicilio  estado_civil
modalidad              1.000000  0.097823  0.190726  0.068133
0.147242          0.027658      0.052281
nivel                  0.097823  1.000000  0.069543  0.058187
0.046057          0.022256      0.022793
jornada                0.190726  0.069543  1.000000  0.031789
0.107615          0.056975      0.134164
sexo                   0.068133  0.058187  0.031789  1.000000
0.116245          0.030169      0.071301
parroquia_residencia   0.147242  0.046057  0.107615  0.116245
1.000000          0.877639      0.095891
sector_domicilio       0.027658  0.022256  0.056975  0.030169
0.877639          1.000000      0.026907
estado_civil           0.052281  0.022793  0.134164  0.071301
0.095891          0.026907      1.000000
```

Mapa de calor correlación de cramer

#Eliminar variable altamente correlacionada

[7]:
```
df_kproto = df_kproto.drop(columns=['sector_domicilio'], errors='ignore')
```

[8]:
```
df_kproto.shape
```

[8]:
```
(48285, 7)
```

[9]:
```
df_kproto.to_excel("df_kproto.xlsx", index=False)
```

[11]:
```
df_kproto.head()
```

[11]:

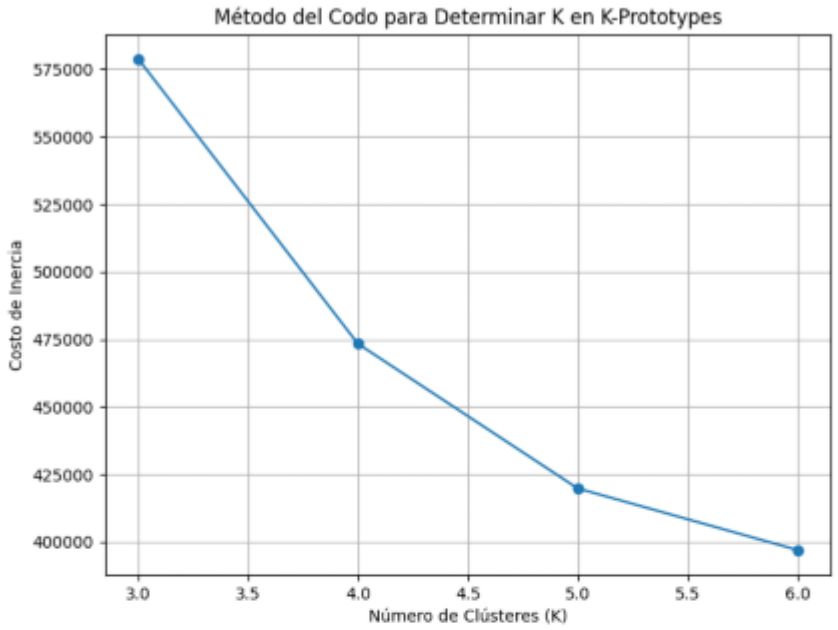
	modalidad	nivel	jornada	sexo	parroquia_residencia	edad	estado_civil
0	presencial	sexto	fin de semana	hombre	carcelen	39	soltero
1	presencial	tercero	nocturna	hombre	san antonio de pichincha	46	divorciado
2	presencial	primero	matutina	hombre	carcelen	30	soltero
3	presencial	sexto	fin de semana	hombre	calderon	31	soltero
4	presencial	segundo	nocturna	hombre	carcelen	42	divorciado

#Metodo del Codo costo

[12]:
```
from kmodes.kprototypes import KPrototypes
import matplotlib.pyplot as plt
```

```
# Asegurar que las columnas categóricas están en formato string
categorical_cols = ["modalidad", "nivel", "jornada", "parroquia_residencia",␣
 ↪"estado_civil", "sexo"]
df_kproto[categorical_cols] = df_kproto[categorical_cols].astype(str)

# Definir los índices de las columnas categóricas
categorical_indices = [df_kproto.columns.get_loc(col) for col in␣
 ↪categorical_cols]

# Asegurar que no hay valores NaN
df_kproto.dropna(inplace=True)

# Definir el rango de K a probar
k_range = range(3, 7)
cost_values = []

# Evaluar K-Prototypes con diferentes valores de K
for k in k_range:
    kproto = KPrototypes(n_clusters=k, random_state=42)

    # Ajustar el modelo con las variables categóricas correctamente definidas
    clusters = kproto.fit_predict(df_kproto, categorical=categorical_indices)

    # Almacenar el costo de inercia
    cost_values.append(kproto.cost_)
    print(f"Para k={k}, costo de inercia: {kproto.cost_}")

# Graficar el Método del Codo
plt.figure(figsize=(8, 6))
plt.plot(k_range, cost_values, marker='o', linestyle='-')
plt.xlabel('Número de Clústeres (K)')
plt.ylabel('Costo de Inercia')
plt.title('Método del Codo para Determinar K en K-Prototypes')
plt.grid(True)
plt.show()
```

```
Para k=3, costo de inercia: 578753.7373635236
Para k=4, costo de inercia: 473352.1002653248
Para k=5, costo de inercia: 419832.20806976303
Para k=6, costo de inercia: 397036.52210171655
```



#Modelo Kprototype nuevo

[13]:
```
pip install kmodes
```

```
Requirement already satisfied: kmodes in c:\users\usuario\anaconda3\lib\site-
packages (0.12.2)
Requirement already satisfied: numpy>=1.10.4 in
c:\users\usuario\anaconda3\lib\site-packages (from kmodes) (1.26.0)
Requirement already satisfied: scikit-learn>=0.22.0 in
c:\users\usuario\anaconda3\lib\site-packages (from kmodes) (1.4.2)
Requirement already satisfied: scipy>=0.13.3 in
c:\users\usuario\anaconda3\lib\site-packages (from kmodes) (1.13.0)
Requirement already satisfied: joblib>=0.11 in
c:\users\usuario\anaconda3\lib\site-packages (from kmodes) (1.4.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in
c:\users\usuario\anaconda3\lib\site-packages (from scikit-learn>=0.22.0->kmodes)
(2.2.0)
Note: you may need to restart the kernel to use updated packages.
```

[14]:
```
#!pip install kmodes
import pandas as pd
```

```python
import numpy as np
from kmodes.kprototypes import KPrototypes
from sklearn.preprocessing import LabelEncoder

# Identificar variables categóricas y numéricas
categorical_cols = ['modalidad', 'nivel', 'jornada', 'sexo',␣
 ↪'parroquia_residencia', 'estado_civil']
numerical_cols = ['edad']

# Copiar el dataset para modificarlo
df_kproto_encoded = df_kproto.copy()

# Codificar variables categóricas usando LabelEncoder
label_encoders = {}
for col in categorical_cols:
    le = LabelEncoder()
    df_kproto_encoded[col] = le.fit_transform(df_kproto_encoded[col])
    label_encoders[col] = le  # Guardar el codificador para futuras conversiones

# Convertir el dataframe a numpy array para K-Prototypes
data_matrix = df_kproto_encoded.values

# Definir el número de clusters (ajustar según necesidad)
k = 4  # Definido por el metodo del codo

# Aplicar K-Prototypes
kproto = KPrototypes(n_clusters=k, init='Huang', verbose=2, random_state=42)
clusters = kproto.fit_predict(data_matrix, categorical=[df_kproto_encoded.
 ↪columns.get_loc(col) for col in categorical_cols])

# Agregar la asignación de clusters al dataframe
df_kproto['cluster'] = clusters
df_kproto.to_excel("df_kproto_with_clusters.xlsx", index=False)
# Mostrar los primeros resultados
print(df_kproto.head())
```

```
Init: initializing centroids
Init: initializing clusters
Starting iterations…
Run: 1, iteration: 1/100, moves: 10678, ncost: 507001.5561649023
Run: 1, iteration: 2/100, moves: 5683, ncost: 489172.8652108286
Run: 1, iteration: 3/100, moves: 1697, ncost: 482329.13063954154
Run: 1, iteration: 4/100, moves: 1266, ncost: 478410.210208063
Run: 1, iteration: 5/100, moves: 178, ncost: 477977.13154731004
Run: 1, iteration: 6/100, moves: 21, ncost: 477970.7487661594
Run: 1, iteration: 7/100, moves: 0, ncost: 477970.7487661594
Init: initializing centroids
```

```
Init: initializing clusters
Starting iterations…
Run: 2, iteration: 1/100, moves: 10135, ncost: 585714.7045277342
Run: 2, iteration: 2/100, moves: 7957, ncost: 551073.0248899859
Run: 2, iteration: 3/100, moves: 6932, ncost: 524079.02866628167
Run: 2, iteration: 4/100, moves: 6253, ncost: 505298.53677929344
Run: 2, iteration: 5/100, moves: 3057, ncost: 495178.91062189837
Run: 2, iteration: 6/100, moves: 1432, ncost: 491365.7795876129
Run: 2, iteration: 7/100, moves: 379, ncost: 489593.7722660632
Run: 2, iteration: 8/100, moves: 1029, ncost: 487716.2440337063
Run: 2, iteration: 9/100, moves: 3847, ncost: 482415.5586590227
Run: 2, iteration: 10/100, moves: 2678, ncost: 477985.6841630108
Run: 2, iteration: 11/100, moves: 63, ncost: 477970.7487661594
Run: 2, iteration: 12/100, moves: 0, ncost: 477970.7487661594
Init: initializing centroids
Init: initializing clusters
Starting iterations…
Run: 3, iteration: 1/100, moves: 7572, ncost: 561319.74044049
Run: 3, iteration: 2/100, moves: 9294, ncost: 532958.5132989006
Run: 3, iteration: 3/100, moves: 6907, ncost: 511021.03084149427
Run: 3, iteration: 4/100, moves: 5032, ncost: 497700.4910364839
Run: 3, iteration: 5/100, moves: 1605, ncost: 492753.8013555735
Run: 3, iteration: 6/100, moves: 853, ncost: 489674.5154581295
Run: 3, iteration: 7/100, moves: 772, ncost: 488229.14700772596
Run: 3, iteration: 8/100, moves: 3932, ncost: 482775.4284497055
Run: 3, iteration: 9/100, moves: 2180, ncost: 478334.35924193653
Run: 3, iteration: 10/100, moves: 792, ncost: 477971.4638761409
Run: 3, iteration: 11/100, moves: 7, ncost: 477970.7487661594
Run: 3, iteration: 12/100, moves: 0, ncost: 477970.7487661594
Init: initializing centroids
Init: initializing clusters
Starting iterations…
Run: 4, iteration: 1/100, moves: 15525, ncost: 501735.60474455036
Run: 4, iteration: 2/100, moves: 4767, ncost: 489544.3993662736
Run: 4, iteration: 3/100, moves: 2497, ncost: 482830.6546796538
Run: 4, iteration: 4/100, moves: 976, ncost: 481693.6046071502
Run: 4, iteration: 5/100, moves: 271, ncost: 480591.5437535654
Run: 4, iteration: 6/100, moves: 1928, ncost: 477138.46592454315
Run: 4, iteration: 7/100, moves: 907, ncost: 476780.96369589237
Run: 4, iteration: 8/100, moves: 0, ncost: 476780.96369589237
Init: initializing centroids
Init: initializing clusters
Starting iterations…
Run: 5, iteration: 1/100, moves: 8902, ncost: 489403.1818150144
Run: 5, iteration: 2/100, moves: 2899, ncost: 482232.35863847594
Run: 5, iteration: 3/100, moves: 511, ncost: 479509.2513451258
Run: 5, iteration: 4/100, moves: 1110, ncost: 477209.68673764315
Run: 5, iteration: 5/100, moves: 41, ncost: 477185.38117118704
```

```
Run: 5, iteration: 6/100, moves: 0, ncost: 477185.38117118704
Init: initializing centroids
Init: initializing clusters
Starting iterations…
Run: 6, iteration: 1/100, moves: 10192, ncost: 545795.2105959433
Run: 6, iteration: 2/100, moves: 4291, ncost: 510124.3271864149
Run: 6, iteration: 3/100, moves: 4005, ncost: 493845.1632214623
Run: 6, iteration: 4/100, moves: 2069, ncost: 486979.42548261466
Run: 6, iteration: 5/100, moves: 994, ncost: 483958.8307020463
Run: 6, iteration: 6/100, moves: 2074, ncost: 478646.58150222996
Run: 6, iteration: 7/100, moves: 896, ncost: 477975.446465544
Run: 6, iteration: 8/100, moves: 18, ncost: 477970.7487661594
Run: 6, iteration: 9/100, moves: 0, ncost: 477970.7487661594
Init: initializing centroids
Init: initializing clusters
Starting iterations…
Run: 7, iteration: 1/100, moves: 6996, ncost: 515979.6223927374
Run: 7, iteration: 2/100, moves: 2387, ncost: 500546.36177248595
Run: 7, iteration: 3/100, moves: 1593, ncost: 493378.7292995226
Run: 7, iteration: 4/100, moves: 937, ncost: 489874.8305892902
Run: 7, iteration: 5/100, moves: 406, ncost: 489131.31126838375
Run: 7, iteration: 6/100, moves: 2898, ncost: 483915.0644481645
Run: 7, iteration: 7/100, moves: 2047, ncost: 481207.8833205701
Run: 7, iteration: 8/100, moves: 2368, ncost: 477979.84282377595
Run: 7, iteration: 9/100, moves: 35, ncost: 477970.7487661594
Run: 7, iteration: 10/100, moves: 0, ncost: 477970.7487661594
Init: initializing centroids
Init: initializing clusters
Starting iterations…
Run: 8, iteration: 1/100, moves: 9656, ncost: 544945.7221462906
Run: 8, iteration: 2/100, moves: 7101, ncost: 515576.8183835224
Run: 8, iteration: 3/100, moves: 4631, ncost: 500558.85470918426
Run: 8, iteration: 4/100, moves: 2717, ncost: 490717.71681471576
Run: 8, iteration: 5/100, moves: 2877, ncost: 483914.50436044653
Run: 8, iteration: 6/100, moves: 1264, ncost: 481890.021294015
Run: 8, iteration: 7/100, moves: 277, ncost: 480787.1025065694
Run: 8, iteration: 8/100, moves: 643, ncost: 478925.8321146819
Run: 8, iteration: 9/100, moves: 2208, ncost: 476781.4335769198
Run: 8, iteration: 10/100, moves: 32, ncost: 476780.96369589237
Run: 8, iteration: 11/100, moves: 0, ncost: 476780.96369589237
Init: initializing centroids
Init: initializing clusters
Starting iterations…
Run: 9, iteration: 1/100, moves: 7915, ncost: 535183.8605521352
Run: 9, iteration: 2/100, moves: 5494, ncost: 504087.6345353204
Run: 9, iteration: 3/100, moves: 3965, ncost: 492597.5789890182
Run: 9, iteration: 4/100, moves: 1007, ncost: 487771.4334260538
Run: 9, iteration: 5/100, moves: 1180, ncost: 485272.1400260537
```

```
Run: 9, iteration: 6/100, moves: 243, ncost: 484496.4138737518
Run: 9, iteration: 7/100, moves: 144, ncost: 484185.54186647
Run: 9, iteration: 8/100, moves: 2, ncost: 484185.47012699314
Run: 9, iteration: 9/100, moves: 0, ncost: 484185.47012699314
Init: initializing centroids
Init: initializing clusters
Init: initializing centroids
Init: initializing clusters
Starting iterations…
Run: 10, iteration: 1/100, moves: 7799, ncost: 491419.09911050356
Run: 10, iteration: 2/100, moves: 3693, ncost: 478705.2928153649
Run: 10, iteration: 3/100, moves: 1811, ncost: 474507.76990268595
Run: 10, iteration: 4/100, moves: 1114, ncost: 473828.2627311069
Run: 10, iteration: 5/100, moves: 0, ncost: 473828.2627311069
Best run was number 10
    modalidad    nivel        jornada    sexo      parroquia_residencia  edad  \
0  presencial    sexto  fin de semana  hombre                  carcelen    39
1  presencial  tercero       nocturna  hombre  san antonio de pichincha    46
2  presencial  primero       matutina  hombre                  carcelen    30
3  presencial    sexto  fin de semana  hombre                  calderon    31
4  presencial  segundo       nocturna  hombre                  carcelen    42

  estado_civil  cluster
0      soltero        0
1   divorciado        0
2      soltero        2
3      soltero        2
4   divorciado        0
```

#Resumen de clusters

[15]:
```
# Contar el número de registros en cada clúster
cluster_counts = df_kproto['cluster'].value_counts().sort_index()

# Mostrar el número de registros por clúster
print("\nNúmero de registros por clúster:")
print(cluster_counts)
```

```
Número de registros por clúster:
cluster
0     1941
1    15495
2     6644
3    24205
Name: count, dtype: int64
```

#Validacion del modelo Silueta y hamming

[ ]:

[ ]:
```
# Importar librerías
import pandas as pd
import numpy as np
from sklearn.metrics import silhouette_score
from kmodes.kprototypes import KPrototypes

# Cargar el dataset con clusters asignados
# file_path_clusters = "/mnt/data/df_original_con_clusters (1).xlsx"
# df_kproto = pd.read_excel(file_path_clusters)

# Definir columnas numéricas y categóricas
numeric_cols = ["edad"]  # Ajustar si hay más variables numéricas en el dataset
categorical_cols = ['modalidad', 'nivel', 'jornada', 'sexo',
 ↪'parroquia_residencia', 'estado_civil']

# Validar que existan columnas numéricas antes de calcular el Índice de Silueta
if numeric_cols and df_kproto[numeric_cols].shape[1] > 0 and
 ↪df_kproto["cluster"].nunique() > 1:
    df_numeric = df_kproto[numeric_cols]
    sil_score = silhouette_score(df_numeric, df_kproto["cluster"])
else:
    sil_score = np.nan  # No se puede calcular silueta con un solo cluster o
 ↪sin columnas numéricas

# Validar que existan columnas categóricas antes de calcular la Distancia de
 ↪Hamming
if categorical_cols:
    df_categorical = df_kproto[categorical_cols].astype(str)  # Asegurar que
 ↪sean cadenas antes de codificar
    df_categorical_encoded = df_categorical.apply(lambda x: x.
 ↪astype("category").cat.codes)  # Codificar categorías

    # Calcular la distancia de Hamming promedio dentro de cada cluster
    hamming_distances = []
    for cluster in df_kproto["cluster"].unique():
        cluster_data = df_categorical_encoded[df_kproto["cluster"] == cluster]
        if len(cluster_data) > 1:
            distance_matrix = np.mean(np.mean(cluster_data.values[:, None] !=
 ↪cluster_data.values, axis=2))
            hamming_distances.append(distance_matrix)

    # Promedio de la distancia de Hamming entre todos los clusters
    hamming_avg = np.mean(hamming_distances) if hamming_distances else np.nan
else:
```

```
    hamming_avg = np.nan  # No se puede calcular si no hay columnas categóricas

# Mostrar los resultados
validation_results = {
    "Índice de Silueta (Variables Numéricas)": sil_score,
    "Distancia de Hamming Promedio (Variables Categóricas)": hamming_avg
}

# Imprimir resultados
print(validation_results)
```

```
{'Índice de Silueta (Variables Numéricas)': 0.5991733503729278, 'Distancia de
Hamming Promedio (Variables Categóricas)': 0.5338092662187528}
```

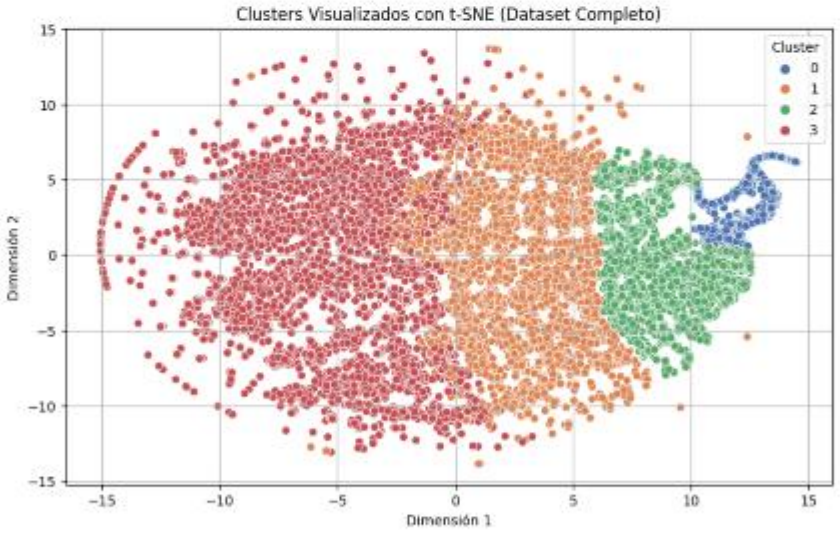
#Modelo k-prototype Unir con el dataset original con 4 k clusters y mandarlo a un archivo para resumenes

[37]:
```
df_original = df_cleaned.copy()
# Asegurar que los índices coincidan para la fusión
#df_original = df_original.reset_index(drop=True)
#df_kproto = df_kproto.reset_index(drop=True)

# Agregar la columna de clusters al dataset original
df_original["cluster"] = df_kproto["cluster"]

# Guardar el dataset original con los clusters asignados
df_original.to_excel("df_prototype_con_clusters.xlsx", index=False)
```

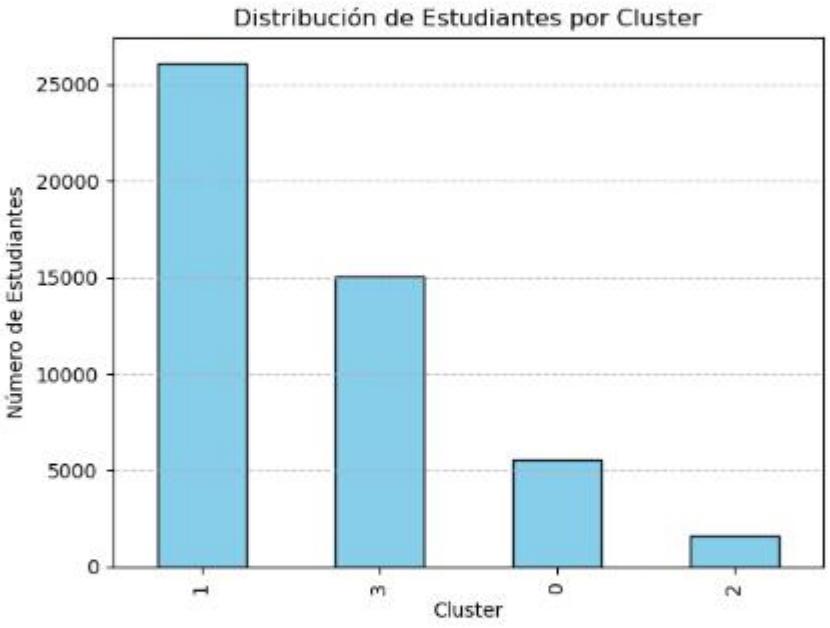
#Grafico de la distribución

[17]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.manifold import TSNE
from sklearn.preprocessing import OneHotEncoder, LabelEncoder
# Crear una copia del dataset transformado con los clusters
df_tSNE_full = df_kproto.copy()

# Transformar variables categóricas para t-SNE
categorical_cols = ["modalidad", "nivel", "jornada", "parroquia_residencia",␣
 ↪"estado_civil","sexo"]

# Aplicar One-Hot Encoding a las variables categóricas
df_tSNE_full = pd.get_dummies(df_tSNE_full, columns=categorical_cols,␣
 ↪drop_first=True)

# Asegurar que 'edad' sigue siendo numérica
df_tSNE_full["edad"] = pd.to_numeric(df_tSNE_full["edad"], errors="coerce")
```

```
# Aplicar t-SNE a todo el dataset sin muestreo
tsne_full = TSNE(n_components=2, perplexity=30, n_iter=300, random_state=42)
df_tsne_full = tsne_full.fit_transform(df_tSNE_full.drop(columns=["cluster"],␣
 ↪errors="ignore"))

# Crear un DataFrame para la visualización
df_plot_full = pd.DataFrame(df_tsne_full, columns=["Dimensión 1", "Dimensión␣
 ↪2"])
df_plot_full["cluster"] = df_tSNE_full["cluster"].values

# Graficar los clusters en 2D con t-SNE sin muestreo
plt.figure(figsize=(10, 6))
sns.scatterplot(x="Dimensión 1", y="Dimensión 2", hue="cluster",␣
 ↪palette="deep", data=df_plot_full)
plt.title("Clusters Visualizados con t-SNE (Dataset Completo)")
plt.xlabel("Dimensión 1")
plt.ylabel("Dimensión 2")
plt.legend(title="Cluster")
plt.grid(True)
plt.show()
```



[30]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Contar cuántos registros hay en cada cluster
df_kproto["cluster"].value_counts().plot(kind="bar", color="skyblue",␣
 ↪edgecolor="black")
plt.xlabel("Cluster")
plt.ylabel("Número de Estudiantes")
plt.title("Distribución de Estudiantes por Cluster")
plt.grid(axis="y", linestyle="--", alpha=0.7)
plt.show()
```



#Estadisticos clusters

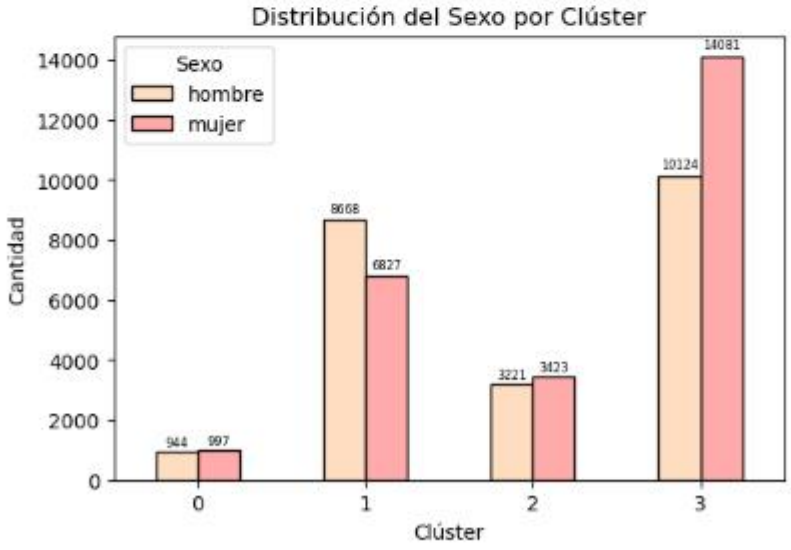
[41]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import matplotlib.colors as mcolors
```

```
# Cargar el archivo de Excel
file_path = "df_prototype_con_clusters.xlsx"
xls = pd.ExcelFile(file_path)

# Verificar las hojas disponibles
xls.sheet_names
# Cargar los datos de la primera hoja
df = pd.read_excel(xls, sheet_name='Sheet1')

# Mostrar las primeras filas para entender la estructura
df.head()

# Agrupar por clúster y calcular estadísticas relevantes
cluster_summary = df.groupby("cluster").agg(
    cantidad_estudiantes=("estudiante_id", "count"),
    edad_promedio=("edad", "mean"),
    edad_minima=("edad", "min"),
    edad_maxima=("edad", "max"),
    sexo_mas_frecuente=("sexo", lambda x: x.mode()[0] if not x.mode().empty␣
 ↪else "No disponible"),
    modalidad_mas_frecuente=("modalidad", lambda x: x.mode()[0] if not x.mode().
 ↪empty else "No disponible"),
    jornada_mas_frecuente=("jornada", lambda x: x.mode()[0] if not x.mode().
 ↪empty else "No disponible"),
    nivel_academico_mas_frecuente=("nivel", lambda x: x.mode()[0] if not x.
 ↪mode().empty else "No disponible"),
    parroquia_mas_comun=("parroquia_residencia", lambda x: x.mode()[0] if not x.
 ↪mode().empty else "No disponible"),
    sector_mas_comun=("sector_domicilio", lambda x: x.mode()[0] if not x.mode().
 ↪empty else "No disponible"),
    estado_civil_mas_frecuente=("estado_civil", lambda x: x.mode()[0] if not x.
 ↪mode().empty else "No disponible")
).reset_index()

# Redondear la edad promedio
cluster_summary["edad_promedio"] = cluster_summary["edad_promedio"].round(2)


# Cargar los datos (asegúrate de que 'df' sea tu DataFrame con los datos␣
 ↪correctos)
# df = pd.read_csv("ruta_del_archivo.csv")  # O carga tu archivo de datos

# Agrupar datos por clúster para cada variable relevante
sexo_por_cluster = df.groupby(["cluster", "sexo"])["sexo"].count().unstack()
modalidad_por_cluster = df.groupby(["cluster", "modalidad"])["modalidad"].
 ↪count().unstack()
```

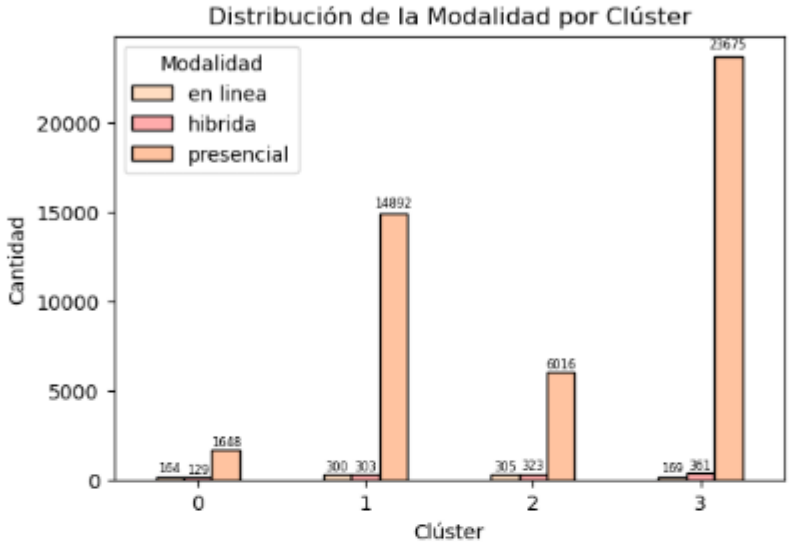
```
jornada_por_cluster = df.groupby(["cluster", "jornada"])["jornada"].count().
 ↪unstack()

# Función para agregar etiquetas con mejor espaciado y tamaño reducido
def add_labels(ax):
    for p in ax.patches:
        if p.get_height() > 0:  # Evitar etiquetas en barras de altura cero
            ax.annotate(f'{int(p.get_height())}',
                        (p.get_x() + p.get_width() / 2., p.get_height() + max(p.
 ↪get_height() * 0.015, 75)),
                        ha='center', va='bottom', fontsize=6, color='black',␣
 ↪rotation=0)

# Paleta de colores pasteles suaves
soft_pastel_colors = ["#FFDDC1", "#FFABAB", "#FFC3A0", "#D5AAFF", "#A0D8B3",␣
 ↪"#C3E0E5"]

# Gráfico 1: Distribución del Sexo por Clúster
plt.figure(figsize=(6,4))
ax = sexo_por_cluster.plot(kind="bar", stacked=False, color=soft_pastel_colors[:
 ↪2], figsize=(6,4), edgecolor="black")
add_labels(ax)
plt.title("Distribución del Sexo por Clúster")
plt.xlabel("Clúster")
plt.ylabel("Cantidad")
plt.legend(title="Sexo")
plt.xticks(rotation=0)
plt.show()

# Gráfico 2: Distribución de la Modalidad por Clúster
plt.figure(figsize=(6,4))
ax = modalidad_por_cluster.plot(kind="bar", stacked=False,␣
 ↪color=soft_pastel_colors[:3], figsize=(6,4), edgecolor="black")
add_labels(ax)
plt.title("Distribución de la Modalidad por Clúster")
plt.xlabel("Clúster")
plt.ylabel("Cantidad")
plt.legend(title="Modalidad")
plt.xticks(rotation=0)
plt.show()

# Gráfico 3: Distribución de la Jornada por Clúster
plt.figure(figsize=(6,4))
ax = jornada_por_cluster.plot(kind="bar", stacked=False,␣
 ↪color=soft_pastel_colors[:3], figsize=(6,4), edgecolor="black")
add_labels(ax)
```

```
plt.title("Distribución de la Jornada por Clúster")
plt.xlabel("Clúster")
plt.ylabel("Cantidad")
plt.legend(title="Jornada")
plt.xticks(rotation=0)
plt.show()

# Gráfico 4: Separación de los Clústeres: Edad vs Nivel
plt.figure(figsize=(6,4))
sns.scatterplot(data=df, x="edad", y="nivel", hue="cluster",␣
  ↪palette=soft_pastel_colors[:4], s=12, edgecolor="black")
plt.title("Separación de los Clústeres: Edad vs Nivel")
plt.xlabel("Edad")
plt.ylabel("Nivel")
plt.legend(title="Clúster")
plt.show()

# Gráfico 5: Distribución de la Edad por Clúster (Boxplot)
plt.figure(figsize=(6,4))
sns.boxplot(data=df, x="cluster", y="edad", palette=soft_pastel_colors[:4])
plt.title("Distribución de la Edad por Clúster")
plt.xlabel("Clúster")
plt.ylabel("Edad")
plt.show()
```

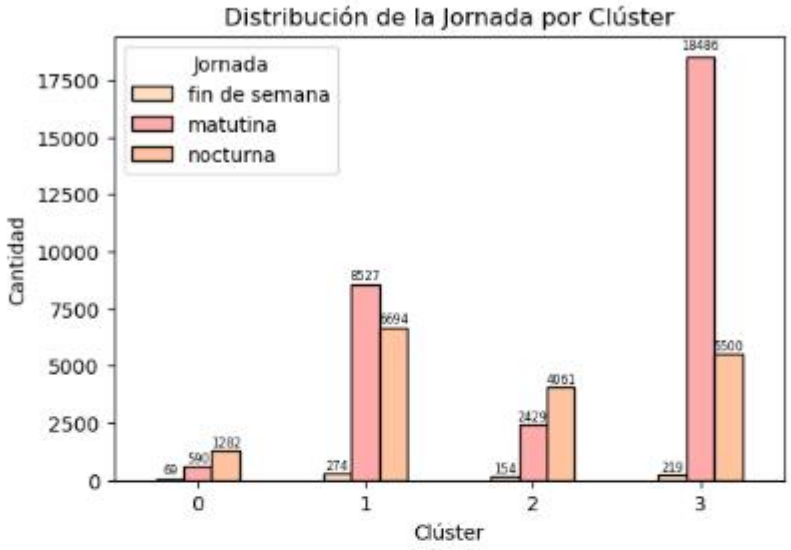
```
<Figure size 600x400 with 0 Axes>
```

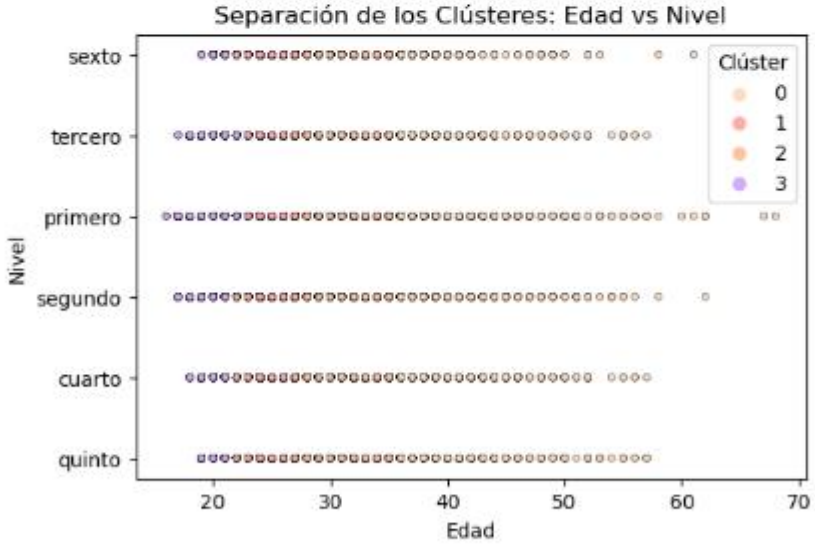


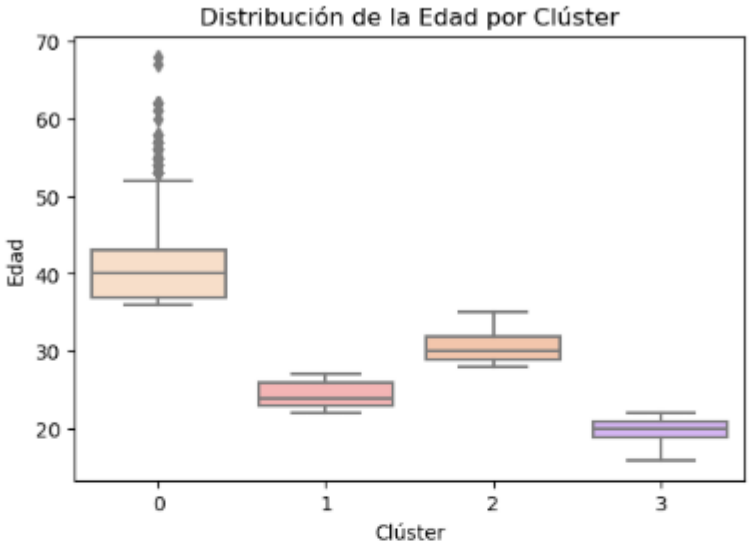
```
<Figure size 600x400 with 0 Axes>
```




```
<Figure size 600x400 with 0 Axes>
```


Distribución de la Jornada por Clúster
Jornada
fin de semana
matutina
nocturna
Cantidad
Clúster
0
1
2
3
69
590
1282
274
8527
6694
154
2429
4061
219
18486
5500


Separación de los Clústeres: Edad vs Nivel
Clúster
0
1
2
3
Nivel
sexto
tercero
primero
segundo
cuarto
quinto
Edad
20
30
40
50
60
70


Distribución de la Edad por Clúster
Edad
70
60
50
40
30
20
0
1
2
3
Clúster

[ ]:

kmodes

March 19, 2025

#Cargando la data

[ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
from kmodes.kmodes import KModes
from sklearn.manifold import TSNE
from sklearn.metrics import pairwise_distances
from sklearn.preprocessing import LabelEncoder

# 1. Cargar el archivo Excel
file_path = "estudiantes_cleaned.xlsx"  # Asegúrate de que el archivo está en␣
↪el mismo directorio
df = pd.read_excel(file_path)
df.head()
```

[ ]:

	estudiante_id	periodo_id	periodo	escuela_id	escuela	carrera_id	carrera	modalidad	nivel	jornada	…	pais_residencia	provincia_residencia	canton_residencia
0	64	74	abr 2020_sep 2020	2	diseño grafico	4	diseño grafico	presencial	sexto	fin de semana	…	ecuador	pichincha	quito
1	316	74	abr 2020_sep 2020	7	analistas de sistemas	87	tecnologia superior en desarrollo de software …	presencial	tercero					
2	551	74	abr 2020_sep 2020	10	administracion recursos humanos - personal	91	tecnologia superior en gestion de talento huma…	presencial	primero					
3	558	74	abr 2020_sep 2020	2	diseño grafico	4	diseño grafico	presencial	sexto					
4	618	79	oct 2022_mar 2023	5	administracion de boticas y farmacias	95	tecnologia superior en asistencia en farmacia …	presencial	segundo					

```
1       nocturna  …         ecuador            pichincha             quito
2       matutina  …         ecuador            pichincha             quito
3  fin de semana  …         ecuador            pichincha             quito
4       nocturna  …         ecuador            pichincha             quito

       parroquia_residencia sector_domicilio fecha_nacimiento edad  \
0                  carcelen            norte       1980-12-28   39
1  san antonio de pichincha           valles       1973-11-05   46
2                  carcelen            norte       1989-04-21   30
3                  calderon            norte       1989-01-20   31
4                  carcelen            norte       1979-11-13   42

   colegio_id                                colegio estado_civil
0        1891  colegio internacional rudolf steiner      soltero
1         909             colegio particular italia   divorciado
2          -1                          no asignado      soltero
3         440        colegio miguel angel asturias      soltero
4        2383                 theodore w  anderson   divorciado

[5 rows x 21 columns]
```

#Preprocesamiento

[ ]:
```
# 2. Eliminar columnas irrelevantes para la segmentación
columns_to_remove =␣
 ↪["estudiante_id","periodo_id","escuela_id","carrera_id","colegio_id","periodo",␣
 ↪"escuela", "carrera", "pais_residencia",
                   "provincia_residencia", "canton_residencia",␣
 ↪"fecha_nacimiento", "colegio"]
df_cleaned = df.drop(columns=columns_to_remove)

# 3. Convertir la columna edad a una variable categórica
df_cleaned['edad_categoria'] = pd.cut(df['edad'].astype(int),
                                      bins=[15, 18, 25, 35, 45, 55, 65, 70],
                                      labels=["15-18", "18-25", "26-35",␣
 ↪"36-45", "46-55", "56-65", "65+"],
                                      include_lowest=True)

# Eliminar la columna original de edad
df_cleaned = df_cleaned.drop(columns=["edad"])

# 4. Convertir todas las variables categóricas a tipo string
df_cleaned = df_cleaned.astype(str)
df_kmodes = df_cleaned.copy()
```

[8]:
```
df_kmodes.shape
```

[8]:
```
(48285, 8)
```

[9]:
```
df_kmodes.head()
```

[9]:

	modalidad	nivel	jornada	sexo	parroquia_residencia	sector_domicilio	estado_civil	edad_categoria
0	presencial	sexto	fin de semana	hombre	carcelen	norte	soltero	36-45
1	presencial	tercero	nocturna	hombre	san antonio de pichincha	valles	divorciado	46-55
2	presencial	primero	matutina	hombre	carcelen	norte	soltero	26-35
3	presencial	sexto	fin de semana	hombre	calderon	norte	soltero	26-35
4	presencial	segundo	nocturna	hombre	carcelen	norte	divorciado	36-45

#Análisis de Variables

[10]:
```
import pandas as pd
import numpy as np
import scipy.stats as ss

# Configurar pandas para mostrar todas las columnas y filas sin cortar
pd.set_option("display.max_rows", None)
pd.set_option("display.max_columns", None)
pd.set_option("display.width", 1000)

# Utilizar df_kmodes en lugar de df_kproto
categorical_cols_kmodes = df_kmodes.select_dtypes(include=['object']).columns

# Función para calcular Cramér V entre dos variables categóricas
def cramers_v(x, y):
    """Calcula la correlación de Cramér V entre dos variables categóricas."""
    confusion_matrix = pd.crosstab(x, y)
    chi2 = ss.chi2_contingency(confusion_matrix)[0]
    n = confusion_matrix.sum().sum()
    phi2 = chi2 / n
    r, k = confusion_matrix.shape
    phi2corr = max(0, phi2 - ((k-1)*(r-1)) / (n-1))
    r_corr = r - ((r-1)**2) / (n-1)
    k_corr = k - ((k-1)**2) / (n-1)
    return np.sqrt(phi2corr / min(r_corr-1, k_corr-1))

# Crear la matriz de Cramér V
num_categorical_kmodes = len(categorical_cols_kmodes)
```

```
cramers_v_matrix_kmodes = pd.DataFrame(np.zeros((num_categorical_kmodes,␣
↪num_categorical_kmodes)),
                                       index=categorical_cols_kmodes,␣
↪columns=categorical_cols_kmodes)

for col1 in categorical_cols_kmodes:
    for col2 in categorical_cols_kmodes:
        if col1 != col2:
            cramers_v_matrix_kmodes.loc[col1, col2] =␣
↪cramers_v(df_kmodes[col1], df_kmodes[col2])
        else:
            cramers_v_matrix_kmodes.loc[col1, col2] = 1.0  # La correlación␣
↪consigo misma es 1

# Imprimir la matriz completa sin que se divida
print(cramers_v_matrix_kmodes)
```

```
                     modalidad     nivel   jornada      sexo
parroquia_residencia  sector_domicilio  estado_civil  edad_categoria
modalidad              1.000000  0.097823  0.190726  0.068133
0.147242          0.027658      0.052281        0.115610
nivel                  0.097823  1.000000  0.069543  0.058187
0.046057          0.022256      0.022793        0.146317
jornada                0.190726  0.069543  1.000000  0.031789
0.107615          0.056975      0.134164        0.215664
sexo                   0.068133  0.058187  0.031789  1.000000
0.116245          0.030169      0.071301        0.021961
parroquia_residencia   0.147242  0.046057  0.107615  0.116245
1.000000          0.877639      0.095891        0.120557
sector_domicilio       0.027658  0.022256  0.056975  0.030169
0.877639          1.000000      0.026907        0.024210
estado_civil           0.052281  0.022793  0.134164  0.071301
0.095891          0.026907      1.000000        0.258624
edad_categoria         0.115610  0.146317  0.215664  0.021961
0.120557          0.024210      0.258624        1.000000
```

[11]:
```
df_kmodes = df_kmodes.drop(columns=['sector_domicilio'], errors='ignore')
df_kmodes.head()
```

[11]:
```
    modalidad    nivel        jornada    sexo      parroquia_residencia
estado_civil edad_categoria
0  presencial    sexto  fin de semana  hombre                  carcelen
soltero          36-45
1  presencial  tercero       nocturna  hombre  san antonio de pichincha
divorciado         46-55
2  presencial  primero       matutina  hombre                  carcelen
soltero          26-35
```

```
3  presencial    sexto  fin de semana  hombre                  calderon
soltero          26-35
4  presencial  segundo       nocturna  hombre                  carcelen
divorciado          36-45
```

[ ]:
```
# 5. Aplicar Label Encoding a todas las variables categóricas para que kmodes␣
↪trabaje bien
label_encoders = {}
df_encoded = df_kmodes.copy()

for col in df_kmodes.columns:
    le = LabelEncoder()
    df_encoded[col] = le.fit_transform(df_kmodes[col])
    label_encoders[col] = le  # Guardar el encoder para futuras conversiones
```

[14]:
```
df_encoded.head()
```

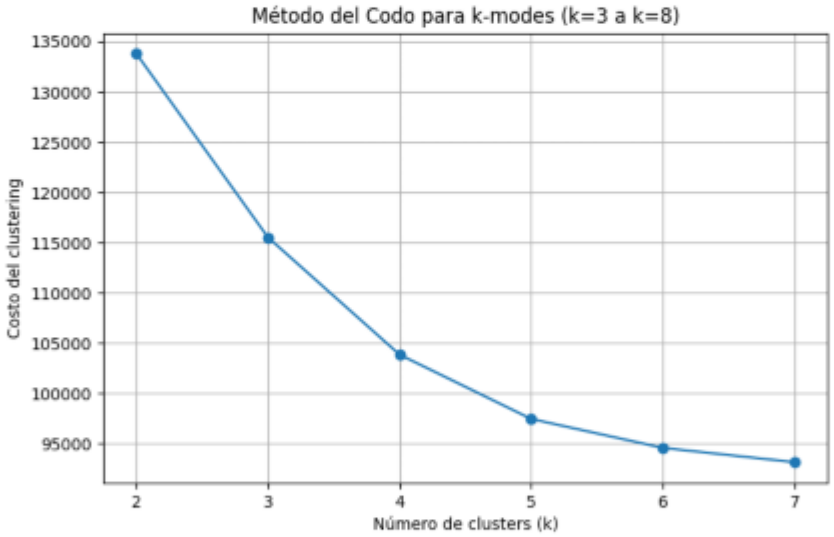
[14]:
```
   modalidad  nivel  jornada  sexo  parroquia_residencia  estado_civil
edad_categoria
0          2      4        0     0                    21             2
3
1          2      5        2     0                   110             1
4
2          2      1        1     0                    21             2
2
3          2      4        0     0                    19             2
2
4          2      3        2     0                    21             1
3
```

#Método del codo con costo

[24]:
```
import matplotlib.pyplot as plt
from kmodes.kmodes import KModes
import numpy as np

# Fijar la semilla aleatoria para resultados consistentes
#np.random.seed(42)

costs = []
K = range(2, 8)  # Probar con valores de k desde 3 hasta 8

for k in K:
    km = KModes(n_clusters=k, init='Huang', n_init=5, verbose=0,␣
↪random_state=42)
    km.fit_predict(df_encoded)
    cost = km.cost_
```

```
    costs.append(cost)  # Guardar el costo para cada k
    print(f"Costo para k={k}: {cost}")  # Imprimir costo por cada k

# Graficar el método del codo
plt.figure(figsize=(8,5))
plt.plot(K, costs, marker='o', linestyle='-')
plt.xlabel('Número de clusters (k)')
plt.ylabel('Costo del clustering')
plt.title('Método del Codo para k-modes (k=3 a k=8)')
plt.grid(True)
plt.show()
```

```
Costo para k=2: 133794.0
Costo para k=3: 115525.0
Costo para k=4: 103861.0
Costo para k=5: 97470.0
Costo para k=6: 94592.0
Costo para k=7: 93169.0
```



#Modelo Kmodes

[32]:
```
# K-Modes con el número óptimo de clusters k según el codo
optimal_k = 5
kmodes = KModes(n_clusters=optimal_k, init='Huang', n_init=5, verbose=0)
```

```
clusters = kmodes.fit_predict(df_encoded)

# Agregar la columna de clusters al dataframe
df_encoded["cluster"] = clusters
# Contar el número de registros en cada clúster
cluster_counts = df_encoded['cluster'].value_counts().sort_index()

# Mostrar el número de registros por clúster
print("\nNúmero de registros por clúster:")
print(cluster_counts)
```

```
Número de registros por clúster:
cluster
0    20459
1     8946
2     7694
3     3892
4     7294
Name: count, dtype: int64
```

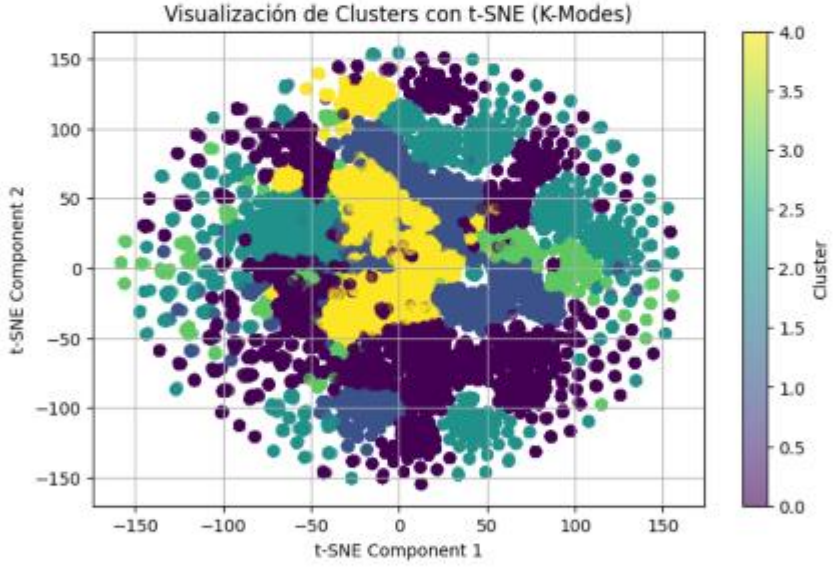
#Registros por clusters

[29]:
```
# Contar el número de registros en cada clúster
cluster_counts = df_encoded['cluster'].value_counts().sort_index()

# Mostrar el número de registros por clúster
print("\nNúmero de registros por clúster:")
print(cluster_counts)
```

```
Número de registros por clúster:
cluster
0    17743
1     7339
2    12521
3     3330
4     7352
Name: count, dtype: int64
```

[ ]:
```
# 9. Aplicar t-SNE para visualización
tsne = TSNE(n_components=2, random_state=42, metric="hamming")
#tsne = TSNE(n_components=2, perplexity=30, random_state=42)

tsne_result = tsne.fit_transform(df_encoded.iloc[:, :-1])  # Ahora son valores␣
↪numéricos

# 10. Graficar t-SNE con los clusters
```

```
plt.figure(figsize=(8, 5))
scatter = plt.scatter(tsne_result[:, 0], tsne_result[:, 1], c=clusters,␣
 ↪cmap='viridis', alpha=0.6)
plt.colorbar(scatter, label="Cluster")
plt.xlabel("t-SNE Component 1")
plt.ylabel("t-SNE Component 2")
plt.title("Visualización de Clusters con t-SNE (K-Modes)")
plt.grid(True)
plt.show()

# Validar el modelo con la distancia de Hamming
hamming_distance = pairwise_distances(df_encoded.iloc[:, :-1],␣
 ↪metric="hamming").mean()

# Mostrar la validación
print(f"Distancia promedio de Hamming: {hamming_distance}")
```




```
Distancia promedio de Hamming: 0.5004763941362437
```

1 aplicando al dataset original para interpretación

[36]:
```
#df_original = df_cleaned.copy()
# Asegurar que los índices coincidan para la fusión
#df_original = df_original.reset_index(drop=True)
#df_kproto = df_kproto.reset_index(drop=True)

# Agregar la columna de clusters al dataset original
df["cluster"] = df_encoded["cluster"]

# Guardar el dataset original con los clusters asignados
df.to_excel("df_kmodes_con_clusters.xlsx", index=False)
```

kmeans

March 19, 2025

[22]:
```
import pandas as pd
import plotly.graph_objects as go
# Cargar el archivo estudiantes_cleaned.xlsx
df_cleaned = pd.read_excel('estudiantes_cleaned.xlsx', engine='openpyxl')
df_cleaned.head()
```

[22]:

	periodo	escuela	carrera	modalidad	nivel	jornada	sexo	pais_residencia	provincia_residencia	canton_residencia	parroquia_residencia	sector_domicilio	fecha_nacimiento	edad	colegio	estado_civil
0	abr 2020_sep 2020	diseño grafico	diseño grafico	presencial	sexto	fin de semana	hombre	ecuador	pichincha	quito	carcelen	norte	1980-12-28	39	colegio internacional rudolf steiner	soltero
1	abr 2020_sep 2020	analistas de sistemas	tecnologia superior en desarrollo de software …	presencial	tercero	nocturna	hombre	ecuador	pichincha	quito	san antonio de pichincha	valles	1973-11-05	46	colegio particular italia	divorciado
2	abr 2020_sep 2020	administracion recursos humanos - personal	tecnologia superior en gestion de talento huma…	presencial	primero	matutina	hombre	ecuador	pichincha	quito	carcelen	norte				
3	abr 2020_sep 2020	diseño grafico	diseño grafico	presencial	sexto	fin de semana	hombre	ecuador	pichincha	quito	calderon	norte				
4	oct 2022_mar 2023	administracion de boticas y farmacias	tecnologia superior en asistencia en farmacia …	presencial	segundo	nocturna	hombre	ecuador	pichincha	quito	carcelen	norte				

```
2       1989-04-21    30                          no asignado      soltero
3       1989-01-20    31          colegio miguel angel asturias      soltero
4       1979-11-13    42                   theodore w  anderson   divorciado
```

#Transformaciones Ejecutar transformaciones y guardar en una copia del original

[23]:
```
# Importar librerías necesarias
import pandas as pd
from sklearn.preprocessing import StandardScaler

# Copiar el dataframe original para modificarlo
df_kmeans = df_cleaned.copy()

# Eliminar columnas irrelevantes
columns_to_drop = ["periodo", "escuela", "carrera", "pais_residencia",
                   "provincia_residencia", "canton_residencia",
                   "colegio", "fecha_nacimiento"]
df_kmeans.drop(columns=columns_to_drop, errors='ignore', inplace=True)  #␣
 ↪'ignore' evita errores si una columna no existe

# Asegurar que todas las columnas categóricas tengan el mismo conjunto de␣
 ↪categorías antes de One-Hot Encoding
categorical_cols = ["modalidad", "jornada", "parroquia_residencia",␣
 ↪"sector_domicilio", "estado_civil"]
df_kmeans[categorical_cols] = df_kmeans[categorical_cols].fillna("Desconocido").
 ↪astype("category")

# Aplicar One-Hot Encoding sin eliminar categorías
df_kmeans = pd.get_dummies(df_kmeans, columns=categorical_cols,␣
 ↪drop_first=False)

# Verificar valores únicos en 'nivel' antes de la codificación ordinal
nivel_order = ["primero", "segundo", "tercero", "cuarto", "quinto", "sexto",
               "septimo", "octavo", "noveno", "decimo"]

niveles_unicos = df_cleaned["nivel"].unique()
for nivel in niveles_unicos:
    if nivel not in nivel_order:
        print(f"Advertencia: El valor '{nivel}' en la columna 'nivel' no está␣
 ↪en la lista predefinida.")

# Codificación ordinal para 'nivel'
df_kmeans["nivel"] = pd.Categorical(df_cleaned["nivel"],␣
 ↪categories=nivel_order, ordered=True).codes + 1

# Reemplazar valores fuera de rango en 'nivel'
df_kmeans["nivel"] = df_kmeans["nivel"].replace(-1, df_kmeans["nivel"].median())
```

```
# Convertir 'sexo' en valores numéricos, manejando valores inesperados
df_kmeans["sexo"] = df_kmeans["sexo"].map({"hombre": 0, "mujer": 1}).fillna(-1).
 ↪astype(int)

# Manejo de valores nulos en 'edad' y 'nivel' con asignación directa
df_kmeans["edad"] = df_kmeans["edad"].fillna(df_kmeans["edad"].median())
df_kmeans["nivel"] = df_kmeans["nivel"].fillna(df_kmeans["nivel"].median())

# Escalar 'edad' y 'nivel'
scaler = StandardScaler()
df_kmeans[["edad", "nivel"]] = scaler.fit_transform(df_kmeans[["edad",␣
 ↪"nivel"]])

# Convertir todas las columnas booleanas y categóricas a enteros de forma␣
 ↪eficiente
for col in df_kmeans.select_dtypes(include=['bool', 'float', 'int']).columns:
    df_kmeans[col] = df_kmeans[col].astype(int)

# Guardar el dataset limpio con los cambios aplicados (descomentar si es␣
 ↪necesario)
# file_path_kmeans = "df_kmeans_transformado.xlsx"
# df_kmeans.to_excel(file_path_kmeans, index=False)

# Mostrar las primeras filas para verificar que no se hayan perdido variables
df_kmeans.head()
```

[23]:

	nivel	sexo	edad	modalidad_en linea	modalidad_hibrida	\
0	1	0	2	0	0	
1	0	0	4	0	0	
2	-1	0	1	0	0	
3	1	0	1	0	0	
4	0	0	3	0	0	

	modalidad_presencial	jornada_fin de semana	jornada_matutina	\
0	1	1	0	
1	1	0	0	
2	1	0	1	
3	1	1	0	
4	1	0	0	

	jornada_nocturna	parroquia_residencia_18 de octubre	_	\
0	0	0	_	
1	1	0	_	
2	0	0	_	
3	0	0	_	
4	1	0	_	

```
   sector_domicilio_centro  sector_domicilio_no seleccionado  \
0                        0                                 0
1                        0                                 0
2                        0                                 0
3                        0                                 0
4                        0                                 0

   sector_domicilio_norte  sector_domicilio_sur  sector_domicilio_valles  \
0                       1                     0                        0
1                       0                     0                        1
2                       1                     0                        0
3                       1                     0                        0
4                       1                     0                        0

   estado_civil_casado  estado_civil_divorciado  estado_civil_soltero  \
0                    0                        0                     1
1                    0                        1                     0
2                    0                        0                     1
3                    0                        0                     1
4                    0                        1                     0

   estado_civil_union libre  estado_civil_viudo
0                         0                   0
1                         0                   0
2                         0                   0
3                         0                   0
4                         0                   0

[5 rows x 170 columns]
```

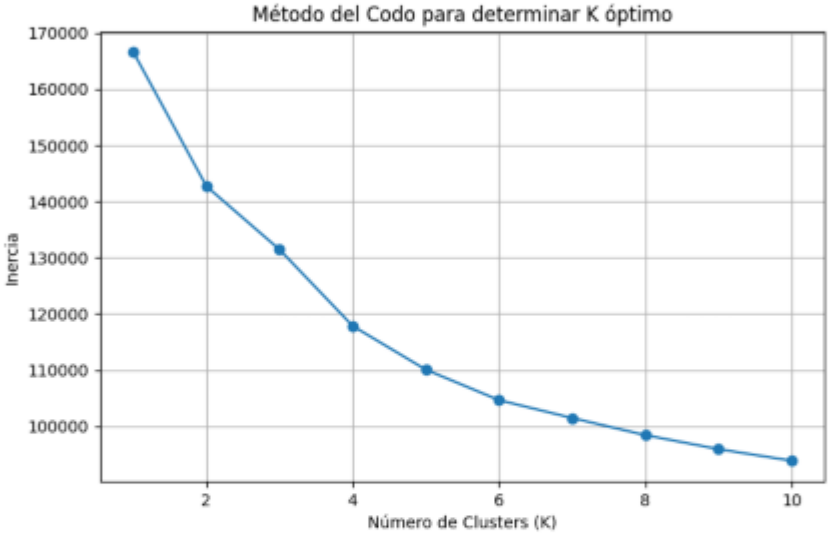
[24]:
```
df_kmeans.shape
```

[24]:
```
(48285, 170)
```

#Método del Codo

[25]:
```
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

# Determinar el número óptimo de clusters usando el método del codo
inertia = []
K_range = range(1, 11)  # Probar de 1 a 10 clusters

for k in K_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    kmeans.fit(df_kmeans)
```

```
    inertia.append(kmeans.inertia_)  # Guardar la inercia para cada valor de k

# Graficar el método del codo
plt.figure(figsize=(8, 5))
plt.plot(K_range, inertia, marker='o', linestyle='-')
plt.xlabel('Número de Clusters (K)')
plt.ylabel('Inercia')
plt.title('Método del Codo para determinar K óptimo')
plt.grid(True)
plt.show()
```

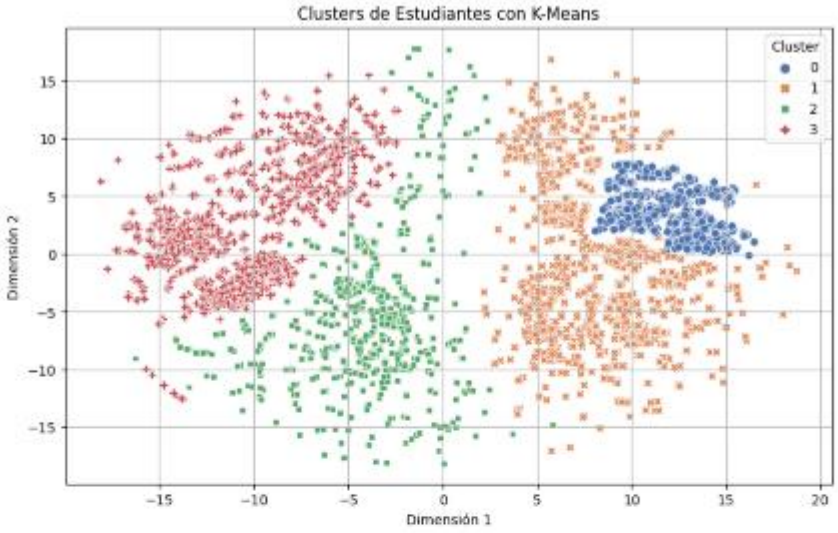


#K-Means

[35]:
```
#Aplicar K-Means con K=4
kmeans = KMeans(n_clusters=4, random_state=42, n_init=10)
clusters = kmeans.fit_predict(df_kmeans)  #Asignar cada punto a un cluster

df_kmeans["cluster"] = clusters
cluster_counts = df_kmeans["cluster"].value_counts()
#Order
cluster_counts = df_kmeans["cluster"].value_counts().sort_index()
cluster_counts
```

[35]:
```
cluster
0    11184
1    18543
2    15453
3     3105
Name: count, dtype: int64
```

[27]:
```
from sklearn.manifold import TSNE
import seaborn as sns
import matplotlib.pyplot as plt

# Aplicar t-SNE para reducir a 2 dimensiones
tsne = TSNE(n_components=2, random_state=42, perplexity=30, n_iter=300)
df_tsne = tsne.fit_transform(df_kmeans)

# Crear un DataFrame para la visualización
df_clusters = pd.DataFrame(df_tsne, columns=["Dimensión 1", "Dimensión 2"])
df_clusters["Cluster"] = clusters

# Graficar los clústeres usando Seaborn
plt.figure(figsize=(10, 6))
sns.scatterplot(
    x="Dimensión 1", y="Dimensión 2", hue="Cluster", palette="deep",
    data=df_clusters, style="Cluster", markers=True
)
plt.title("Clusters de Estudiantes con K-Means")
plt.xlabel("Dimensión 1")
plt.ylabel("Dimensión 2")
plt.legend(title="Cluster")
plt.grid(True)
plt.show()
```



#Revertir para obtener resultados

[28]:
```
# Crear una copia del dataset original
df_original = df_cleaned.copy()
# Eliminar columnas irrelevantes
columns_to_drop = ["periodo", "escuela", "carrera", "pais_residencia",
                   "provincia_residencia", "canton_residencia",
                   "colegio", "fecha_nacimiento"]
df_original.drop(columns=columns_to_drop, inplace=True)
# Asignar la columna de clúster del dataset transformado al dataset original
df_original["cluster"] = df_kmeans["cluster"]
df_original.to_excel("df_original_con_cluster.xlsx", index=False,␣
 ↪engine="openpyxl")
# Verificar las primeras filas del nuevo DataFrame
df_original.head()
```

[28]:

	modalidad	nivel	jornada	sexo	parroquia_residencia	\
0	presencial	sexto	fin de semana	hombre	carcelen	
1	presencial	tercero	nocturna	hombre	san antonio de pichincha	
2	presencial	primero	matutina	hombre	carcelen	
3	presencial	sexto	fin de semana	hombre	calderon	
4	presencial	segundo	nocturna	hombre	carcelen	

	sector_domicilio	edad	estado_civil	cluster

```
0          norte    39      soltero        0
1         valles    46   divorciado        0
2          norte    30      soltero        2
3          norte    31      soltero        2
4          norte    42   divorciado        0
```

[29]:
```python
# Crear un resumen detallado para cada clúster usando df_original
cluster_details = []

for cluster_id in df_original['cluster'].unique():
    cluster_data = df_original[df_original['cluster'] == cluster_id]
    detail = {
        'Cluster': cluster_id,
        'Edad promedio': cluster_data['edad'].mean(),
        'Edad mínima': cluster_data['edad'].min(),
        'Edad máxima': cluster_data['edad'].max(),
        'Nivel más frecuente': cluster_data['nivel'].mode().iloc[0],
        'Sexo más frecuente': cluster_data['sexo'].mode().iloc[0],
        'Modalidad más frecuente': cluster_data['modalidad'].mode().iloc[0],
        'Jornada más frecuente': cluster_data['jornada'].mode().iloc[0],
        'Parroquia más frecuente': cluster_data['parroquia_residencia'].mode().
 ↪iloc[0],
        'Sector de domicilio más frecuente': cluster_data['sector_domicilio'].
 ↪mode().iloc[0],
        'Estado civil más frecuente': cluster_data['estado_civil'].mode().
 ↪iloc[0],
        'Cantidad de registros': len(cluster_data),
        'Porcentaje del total': len(cluster_data) / len(df_original) * 100
    }
    cluster_details.append(detail)

# Convertir la lista de detalles en un DataFrame
df_cluster_details = pd.DataFrame(cluster_details)

# Mostrar el DataFrame final
df_cluster_details
```

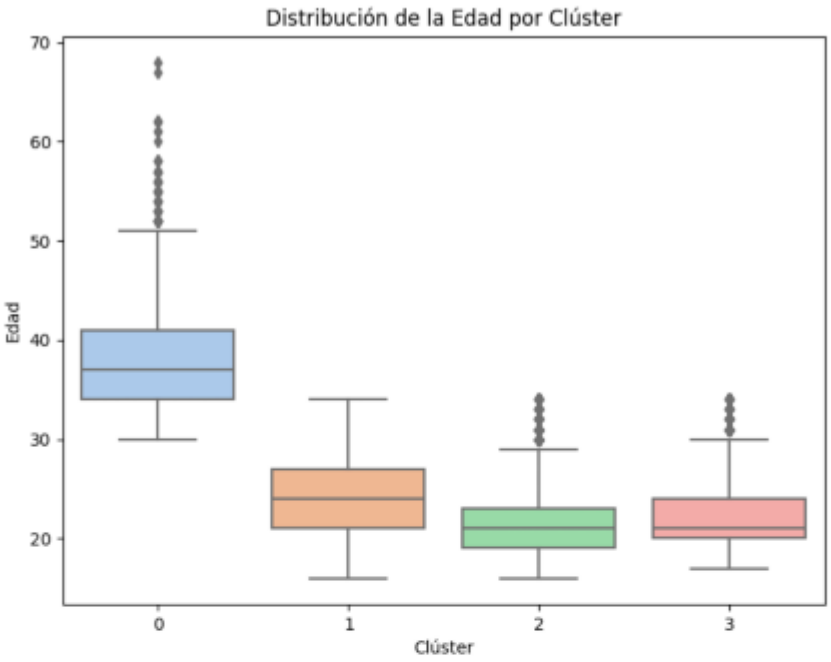
[29]:
```
   Cluster  Edad promedio  Edad mínima  Edad máxima Nivel más frecuente  \
0        0      37.854428           30           68             primero
1        2      21.656258           16           34             primero
2        1      24.218210           16           34             primero
3        3      22.207976           17           34             primero

  Sexo más frecuente Modalidad más frecuente Jornada más frecuente  \
0              mujer              presencial              nocturna
1              mujer              presencial              matutina
2              mujer              presencial              nocturna
```

```
3             mujer             presencial              matutina

  Parroquia más frecuente Sector de domicilio más frecuente  \
0               calderon                              norte
1               calderon                              norte
2               calderon                              norte
3            chillogallo                                sur

  Estado civil más frecuente  Cantidad de registros  Porcentaje del total
0                      casado                   3105              6.430568
1                     soltero                  18543             38.403231
2                     soltero                  15453             32.003728
3                     soltero                  11184             23.162473
```

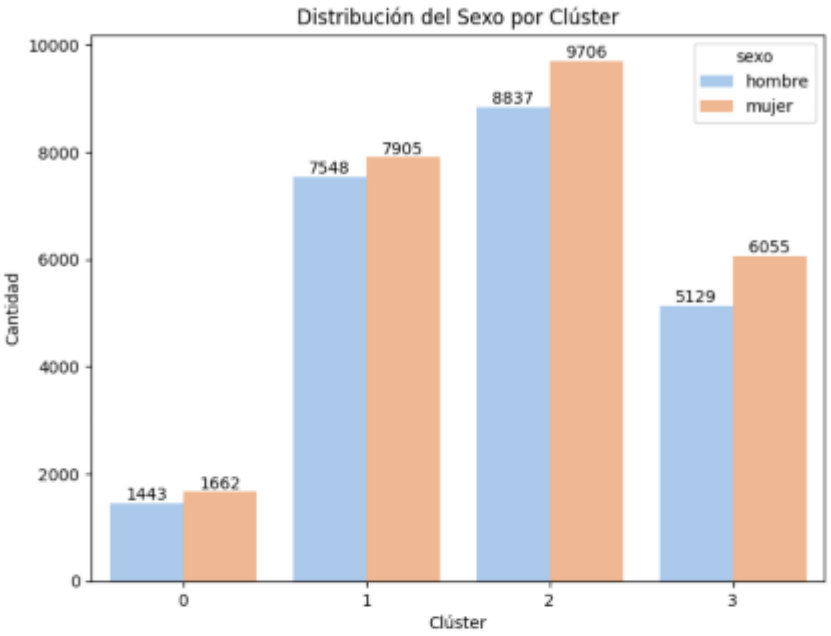
#Estadisticos Cluster

[30]:
```
import matplotlib.pyplot as plt
import seaborn as sns

# Boxplot de la edad por clúster
plt.figure(figsize=(8, 6))
sns.boxplot(x='cluster', y='edad', data=df_original, palette="pastel")
plt.title('Distribución de la Edad por Clúster')
plt.xlabel('Clúster')
plt.ylabel('Edad')
plt.show()

# Gráfico de barras para la distribución de sexo por clúster
plt.figure(figsize=(8, 6))
ax = sns.countplot(x='cluster', hue='sexo', data=df_original, palette="pastel")
for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + p.get_width() / 2., p.
 ↪get_height()), ha='center', va='center', xytext=(0, 5), textcoords='offset␣
 ↪points')
plt.title('Distribución del Sexo por Clúster')
plt.xlabel('Clúster')
plt.ylabel('Cantidad')
plt.show()

# Gráfico de barras para la distribución de modalidad por clúster
plt.figure(figsize=(8, 6))
ax = sns.countplot(x='cluster', hue='modalidad', data=df_original,␣
 ↪palette="pastel")
for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + p.get_width() / 2., p.
 ↪get_height()), ha='center', va='center', xytext=(0, 5), textcoords='offset␣
 ↪points')
```

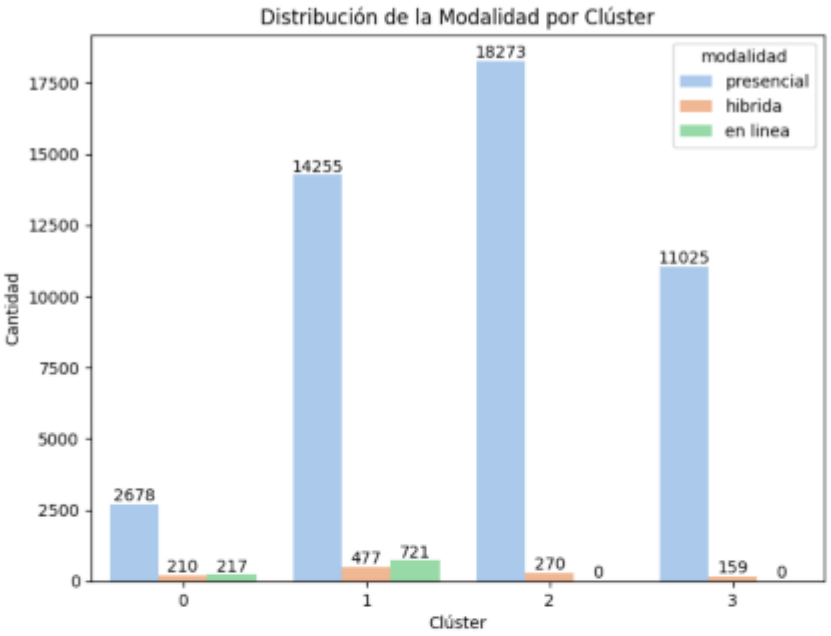
```
plt.title('Distribución de la Modalidad por Clúster')
plt.xlabel('Clúster')
plt.ylabel('Cantidad')
plt.show()

# Gráfico de barras para la distribución de jornada por clúster
plt.figure(figsize=(8, 6))
ax = sns.countplot(x='cluster', hue='jornada', data=df_original,␣
 ↪palette="pastel")
for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + p.get_width() / 2., p.
 ↪get_height()), ha='center', va='center', xytext=(0, 5), textcoords='offset␣
 ↪points')
plt.title('Distribución de la Jornada por Clúster')
plt.xlabel('Clúster')
plt.ylabel('Cantidad')
plt.show()

# Gráfico de barras para la distribución de estado civil por clúster
plt.figure(figsize=(8, 6))
ax = sns.countplot(x='cluster', hue='estado_civil', data=df_original,␣
 ↪palette="pastel")
for p in ax.patches:
    ax.annotate(int(p.get_height()), (p.get_x() + p.get_width() / 2., p.
 ↪get_height()), ha='center', va='center', xytext=(0, 5), textcoords='offset␣
 ↪points')
plt.title('Distribución del Estado Civil por Clúster')
plt.xlabel('Clúster')
plt.ylabel('Cantidad')
plt.show()

# Scatterplot de Edad vs Nivel por Clúster
plt.figure(figsize=(8, 6))
sns.scatterplot(x='edad', y='nivel', hue='cluster', data=df_original,␣
 ↪palette="pastel", style='cluster')
plt.title('Separación de los Clústeres: Edad vs Nivel')
plt.xlabel('Edad')
plt.ylabel('Nivel')
plt.legend(title='Clúster')
plt.show()
#df_pca = df_kmeans.copy()
```


Distribución de la Edad por Clúster
Edad
70
60
50
40
30
20
0
1
2
3
Clúster

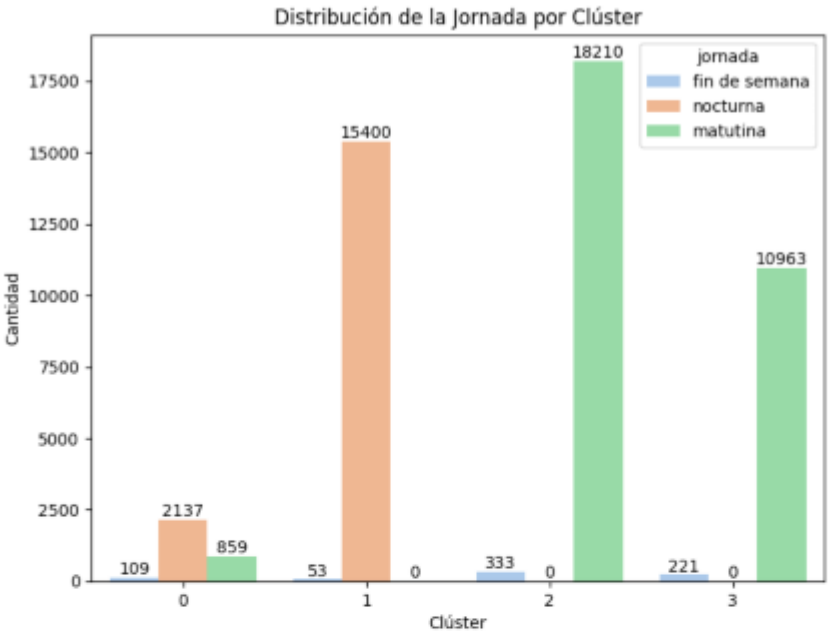
11

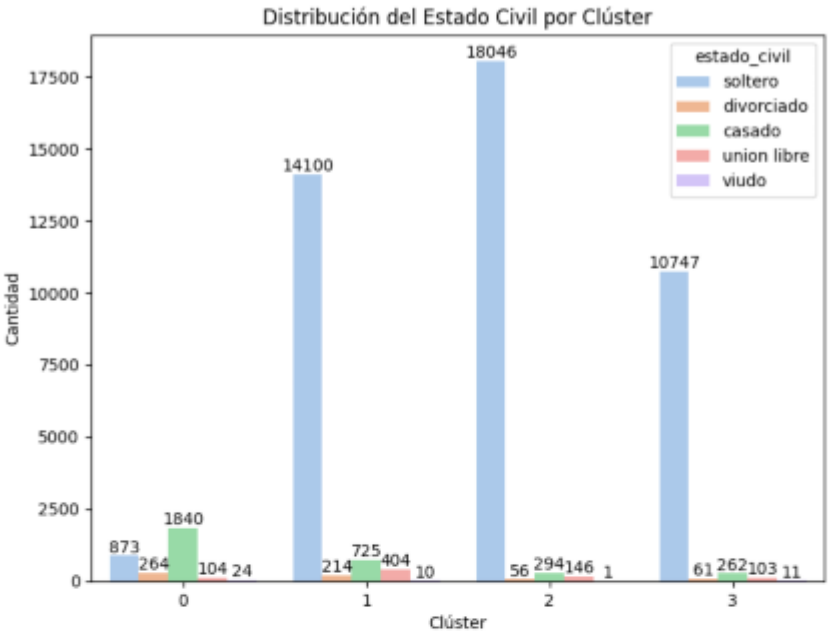

Distribución del Sexo por Clúster
sexo
hombre
mujer
Cantidad
Clúster
0
1
2
3
1443
1662
7548
7905
8837
9706
5129
6055
0
2000
4000
6000
8000
10000

12

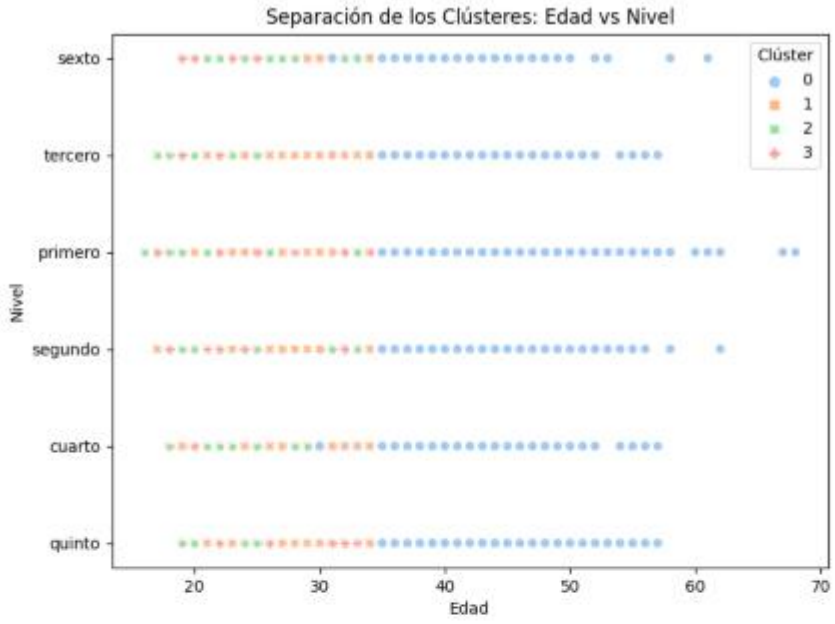

Distribución de la Modalidad por Clúster
modalidad
presencial
hibrida
en linea
Cantidad
Clúster
0
1
2
3
2678
210
217
14255
477
721
18273
270
0
11025
159
0

13


Distribución de la Jornada por Clúster
jornada
fin de semana
nocturna
matutina
Cantidad
Clúster
0
2500
5000
7500
10000
12500
15000
17500
109
2137
859
53
15400
0
333
0
18210
221
0
10963
0
1
2
3


Distribución del Estado Civil por Clúster
estado_civil
soltero
divorciado
casado
union libre
viudo
Cantidad
Clúster
0
1
2
3
873
264
1840
104
24
14100
214
725
404
10
18046
56
294
146
1
10747
61
262
103
11

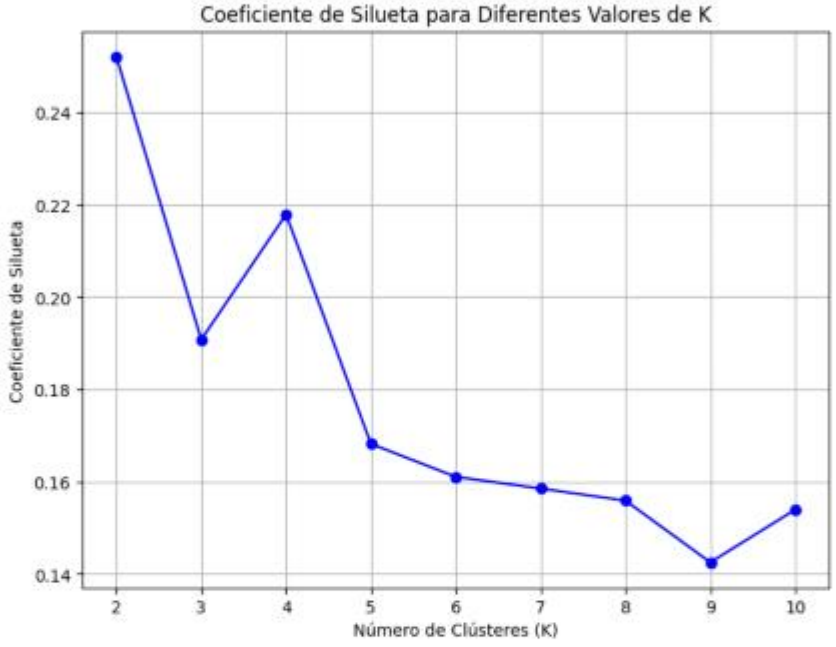
15



#Evaluación sin PCA K con este gráfico se confirman los ideals 4 es el ideal

[34]:
```
from sklearn.metrics import silhouette_score
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

# Lista de valores de K a probar
k_values = range(2, 11)
silhouette_scores = []

# Calcular el coeficiente de silueta para cada K usando df_kmeans
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    clusters = kmeans.fit_predict(df_kmeans)  # Ajustar el modelo y predecir␣
 ↪los clústeres
    score = silhouette_score(df_kmeans, clusters)  # Calcular el coeficiente de␣
 ↪silueta
    silhouette_scores.append(score)

# Graficar el resultado
plt.figure(figsize=(8, 6))
```

```
plt.plot(k_values, silhouette_scores, marker='o', linestyle='-', color='b')
plt.title('Coeficiente de Silueta para Diferentes Valores de K')
plt.xlabel('Número de Clústeres (K)')
plt.ylabel('Coeficiente de Silueta')
plt.xticks(k_values)
plt.grid(True)
plt.show()
```



#Silhouette

[31]:
```
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

# Aplicar K-Means con K=4
kmeans = KMeans(n_clusters=4, random_state=42, n_init=10)
clusters = kmeans.fit_predict(df_kmeans)

# Calcular el coeficiente de silueta
silhouette_avg = silhouette_score(df_kmeans, clusters)
```

```
silhouette_avg
```

[31]: 0.21790793050461307

Evaluar con PCA

[33]:
```
from sklearn.decomposition import PCA
# Crear una copia del dataset original
df_pca = df_cleaned.copy()

# Eliminar columnas irrelevantes
columns_to_drop = ["periodo", "escuela", "carrera", "pais_residencia",
                   "provincia_residencia", "canton_residencia",
                   "colegio", "fecha_nacimiento"]
df_pca.drop(columns=columns_to_drop, errors='ignore', inplace=True)

# Asegurar que todas las columnas categóricas tengan el mismo conjunto de␣
↪categorías antes de One-Hot Encoding
categorical_cols = ["modalidad", "jornada", "parroquia_residencia",␣
↪"sector_domicilio", "estado_civil"]
df_pca[categorical_cols] = df_pca[categorical_cols].fillna("Desconocido").
↪astype("category")

# Aplicar One-Hot Encoding sin eliminar categorías
df_pca = pd.get_dummies(df_pca, columns=categorical_cols, drop_first=False)

# Codificación ordinal para 'nivel'
nivel_order = ["primero", "segundo", "tercero", "cuarto", "quinto", "sexto",
               "septimo", "octavo", "noveno", "decimo"]

df_pca["nivel"] = pd.Categorical(df_cleaned["nivel"], categories=nivel_order,␣
↪ordered=True).codes + 1
df_pca["nivel"] = df_pca["nivel"].replace(-1, df_pca["nivel"].median())

# Convertir 'sexo' en valores numéricos
df_pca["sexo"] = df_pca["sexo"].map({"hombre": 0, "mujer": 1}).fillna(-1).
↪astype(int)

# Manejo de valores nulos en 'edad' y 'nivel' sin usar inplace
df_pca["edad"] = df_pca["edad"].fillna(df_pca["edad"].median())
df_pca["nivel"] = df_pca["nivel"].fillna(df_pca["nivel"].median())

# Escalar las variables
scaler = StandardScaler()
df_pca[df_pca.select_dtypes(include=['number']).columns] = scaler.
↪fit_transform(df_pca.select_dtypes(include=['number']))
```

```python
# Aplicar PCA
pca = PCA(n_components=0.95)  # Retener el 95% de la varianza
pca_result = pca.fit_transform(df_pca)

# Aplicar K-Means con un número óptimo de clusters (ajustar si es necesario)
kmeans = KMeans(n_clusters=4, random_state=42, n_init=10)
labels = kmeans.fit_predict(pca_result)

# Calcular el coeficiente de silueta
silhouette_avg = silhouette_score(pca_result, labels)
print(f"Coeficiente de Silueta: {silhouette_avg}")
```

```
Coeficiente de Silueta: 0.18851413262543523
```

ANEXO 2

Se anexa el acta de confidencialidad

ACUERDO DE CONFIDENCIALIDAD

Este Acuerdo de Confidencialidad (en adelante, el "Acuerdo") es celebrado el 13 de marzo del 2025 entre:

INSTITUTO TECNOLÓGICO UNIVERSITARIO CORDILLERA, con domicilio en Avenida la Prensa y Logroño N45-268, representada por **Ms. David Andres Flores Torres**, en calidad de Rector, en adelante "**LA INSTITUCIÓN**";

Y

Pablo Giovanny Caiza Iza, identificado con número de cedula **1717356586**, con domicilio en Galo Plaza e Ignacio de Veintimilla N15-423, en adelante "**EL INVESTIGADOR**".

Ambas partes acuerdan lo siguiente:

1.OBJETO DEL ACUERDO

LA INSTITUCIÓN proporcionará acceso a ciertos datos e información (en adelante, "**INFORMACIÓN CONFIDENCIAL**") a **EL INVESTIGADOR** con el propósito exclusivo de realizar estudios para el trabajo de titulación sobre segmentación de estudiantes en función de variables demográficas y académicas, para la obtención del título de Magister en Sistemas de Información, mención en Data Science

2. DEFINICIÓN DE INFORMACIÓN CONFIDENCIAL

Se considerará **INFORMACIÓN CONFIDENCIAL** cualquier dato, documento, código, modelo, metodología, resultado de análisis o cualquier otra información relacionada con los registros académicos y demográficos de los estudiantes de **LA INSTITUCIÓN** a la que **EL INVESTIGADOR** tenga acceso en virtud de su investigación, salvo aquella que sea de dominio público.

Los datos especificos a los que **EL INVESTIGADOR** tendrá acceso incluyen registros de matrículas Normales de cinco años que incluyen los siguientes períodos académicos: *Abr 2020_Sep 2020, Abr 2021_Sep 2021, Abr 2022_Sep 2022, Abr 2023_Sep 2023, Abr 2024_Sep 2024, Oct 2020_Mar 2021, Oct 2021_Mar 2022, Oct 2022_Mar 2023, Oct 2023_Mar 2024, Oct 2024_Mar 2025* y de los cuales se extrae la siguiente información:

id del estudiante (solo código secuencial), periodo académico, carrera, especialidad, malla, modalidad de estudio, jornada, sexo, país de procedencia, provincia de procedencia, cantón de procedencia, parroquia de residencia, sector de residencia, fecha de nacimiento, edad en el periodo del registro de matrícula, colegio de procedencia, estado civil.

3. OBLIGACIONES DE CONFIDENCIALIDAD

EL INVESTIGADOR se compromete a:

- No divulgar, compartir, copiar, reproducir ni utilizar la **INFORMACIÓN CONFIDENCIAL** para fines distintos a los autorizados por **LA INSTITUCIÓN**.
- Mantener la **INFORMACIÓN CONFIDENCIAL** en un entorno seguro y protegido contra accesos no autorizados.
- No transferir la **INFORMACIÓN CONFIDENCIAL** a terceros sin autorización expresa y por escrito de **LA INSTITUCIÓN**.

4. DURACIÓN

Este Acuerdo tendrá una duración indefinida y permanecerá en vigor incluso después de la finalización del trabajo de titulación de **EL INVESTIGADOR**.

5. CONSECUENCIAS DEL INCUMPLIMIENTO

En caso de incumplimiento de este Acuerdo, **EL INVESTIGADOR** será responsable por los daños y perjuicios ocasionados y podrá estar sujeto a acciones legales y/o disciplinarias por parte de **LA INSTITUCIÓN**.

6. LEGISLACIÓN APLICABLE

Este Acuerdo se regirá por las leyes del Ecuador y cualquier controversia será resuelta en los tribunales de Quito - Ecuador.

En señal de conformidad, las partes firman este Acuerdo en dos ejemplares el 13 de marzo del 2025.

INSTITUTO TECNOLÓGICO UNIVERSITARIO CORDILLERA

DAVID ANDRES FLORES TORRES

PABLO GIOVANNY CAIZA IZA

```sql
SELECT alumno_acad.serial_alu
         AS estudiante_id,
       periodo_acad.serial_per
         AS periodo_id,
       periodo_acad.descripcion_per
         AS periodo,
       escuela_acad.serial_esc
         AS escuela_id,
       escuela_acad.descripcion_esc
         AS escuela,
       carrera_acad.serial_car
         AS carrera_id,
       carrera_acad.descripcion_car
         AS carrera,
       modalidad_acad.descripcion_mod
         AS modalidad,
       ifnull(nivel_acad.descripcionportal_niv, 'NO DEFINIDO')
         AS nivel,
       ifnull(jornada_acad.descripcion_jor, 'NO DEFINIDO')
         AS jornada,
       alumno_acad.sexo_alu
         AS sexo,
       pais_acad.descripcion_pai
         AS pais_residencia,
       provincia_acad.descripcion_prv
         AS provincia_residencia,
       canton_acad.descripcion_can
         AS canton_residencia,
       parroquia_acad.descripcion_prr
         AS parroquia_residencia,
       alumno_acad.sectordom_alu
         AS sector_domicilio,

       date_format(alumno_acad.fechanacimiento_alu, '%Y-%m-%d')
         AS fecha_nacimiento,
         YEAR(periodo_acad.fechainicio_per)
       - YEAR(alumno_acad.fechanacimiento_alu)
       - IF(
           DATE_FORMAT(periodo_acad.fechainicio_per, '%m-%d') <
           DATE_FORMAT(alumno_acad.fechanacimiento_alu, '%m-%d'),
           1,
           0)
         AS edad,
       colegio_acad.serial_col
         AS colegio_id,
       colegio_acad.descripcion_col
         AS colegio,
       alumno_acad.estadocivil_alu as estado_civil
FROM periodo_acad,
     mallaalumno_acad,
     alumno_acad,
     malla_acad,
     especialidad_acad,
     carrera_acad,
     escuela_acad,
     modalidad_acad,
     pais_acad,
     provincia_acad,
     canton_acad,
     parroquia_acad,
     colegio_acad,
     matricula_acad
     LEFT JOIN nivel_acad ON matricula_acad.nivel_mtc = nivel_acad.serial_niv
     LEFT JOIN jornada_acad
        ON matricula_acad.jornada_mtc = jornada_acad.serial_jor
     LEFT JOIN paralelo_acad
        ON matricula_acad.paralelo_mtc = paralelo_acad.serial_par
WHERE    matricula_acad.serialper = periodo_acad.serial_per
      AND matricula_acad.serial_alm = mallaalumno_acad.serial_alm
      AND mallaalumno_acad.serial_alu = alumno_acad.serial_alu
      AND pais_acad.serial_pai = provincia_acad.serial_pai
      AND provincia_acad.serial_prv = canton_acad.serial_prv
      AND carrera_acad.serial_esc = escuela_acad.serial_esc
      AND colegio_acad.serial_col = alumno_acad.serial_col
      AND canton_acad.serial_can = parroquia_acad.serial_can
      AND parroquia_acad.serial_prr = alumno_acad.prr_residencia
      AND modalidad_acad.serial_mod = malla_acad.serial_mod
      AND mallaalumno_acad.serial_maa = malla_acad.serial_maa
      AND malla_acad.serial_esp = especialidad_acad.serial_esp
      AND especialidad_acad.serial_car = carrera_acad.serial_car
      AND year(fechainicio_per) IN (2020,
                                    2021,
                                    2022,
                                    2023,
                                    2024)
      AND periodo_acad.serial_nve = 1
ORDER BY alumno_acad.serial_alu
```