

**INSTITUTO TECNOLÓGICO DE TOLUCA**

**ANTEPROYECTO DE RESIDENCIA PROFESIONAL**

**“ANÁLISIS DE DATOS CON EL ALGORITMO DE AGRUPAMIENTO  
K-PROTOTYPES”**

**Ingeniería en Sistemas Computacionales**

**PRESENTA:**

**Fabiola Ramírez Guerrero**

**No. CONTROL:**

**C18280980**

**ASESOR INTERNO:**

**Andrea Olimpia Nájera Ávila**

**ASESOR EXTERNO:**

**Dra. Eréndira Rendón Lara**

**METEPEC, ESTADO DE MÉXICO, JUNIO DE 2024**

Agradecimientos

Deben ser concretos, indicando las razones por las cuales se menciona a cada persona o institución.

Resumen

Es una síntesis del proyecto final. Una representación abreviada del proyecto. Resaltando las ideas generales del proyecto y los logros alcanzados, destacando puntos de vista del autor que motiven al lector a leer el documento. No debe exceder de una cuartilla.

Índice

[1 Introducción 1](#_Toc169975387)

[2 Descripción de la empresa u organización y del puesto o área del trabajo del estudiante. 2](#_Toc169975388)

[3 Problema a resolver 3](#_Toc169975389)

[4 Objetivos 5](#_Toc169975390)

[4.1 General 5](#_Toc169975391)

[4.2 Específicos 5](#_Toc169975392)

[5 Justificación 6](#_Toc169975393)

[6 Marco Teórico (fundamentos teóricos) 7](#_Toc169975394)

[6.1 Minería de datos 7](#_Toc169975395)

[6.1.1 Disciplinas relacionadas con minería de datos 8](#_Toc169975396)

[6.2 Clustering 9](#_Toc169975397)

[6.2.1 Objetivos del Clustering 10](#_Toc169975398)

[6.2.2 Importancia del Clustering en el Análisis de Datos 10](#_Toc169975399)

[6.2.3 Aplicaciones del Clustering 11](#_Toc169975400)

[6.3 Tipos de algoritmos de agrupamiento (clustering) 12](#_Toc169975401)

[6.3.1 Clustering jerárquico 13](#_Toc169975402)

[6.3.2 Clustering particional 14](#_Toc169975403)

[6.3.3 Algoritmos de agrupamiento de recorridos simples. 15](#_Toc169975404)

[6.4 Algoritmos de Clustering 16](#_Toc169975405)

[6.4.1 El algoritmo DBSCAN 16](#_Toc169975406)

[6.4.2 El algoritmo BIRCH 16](#_Toc169975407)

[6.4.3 El Algoritmo K-Means 17](#_Toc169975408)

[6.4.4 El algoritmo k-modes 22](#_Toc169975409)

[6.4.5 Algoritmo k-Prototypes 23](#_Toc169975410)

[6.5 Investigaciones y proyectos usados con el algoritmo K-Prototypes 26](#_Toc169975411)

[6.5.1 Mejora de K-Prototypes mediante optimización evolutiva 26](#_Toc169975412)

[6.5.2 Aplicaciones en bioinformática 26](#_Toc169975413)

[6.6 Proyectos prácticos con el algoritmo K-Prototype 26](#_Toc169975414)

[6.6.1 Análisis de clientes en comercio electrónico 26](#_Toc169975415)

[6.6.2 Estudio de patrones de viaje en transporte público 26](#_Toc169975416)

[6.7 Visual Studio Code (VS Code) 27](#_Toc169975417)

[6.8 Python 27](#_Toc169975418)

[6.9 Librerías de Python 28](#_Toc169975419)

[6.9.1 NumPy 28](#_Toc169975420)

[6.9.2 Pandas 28](#_Toc169975421)

[6.9.3 scikit-learn 29](#_Toc169975422)

[6.9.4 Matplotlib y Seaborn 29](#_Toc169975423)

[6.9.5 SciPy 29](#_Toc169975424)

[6.9.6 PyCaret 29](#_Toc169975425)

[6.9.7 Jupyter Notebook 29](#_Toc169975426)

[7 Procedimiento y descripción de las actividades realizadas. 30](#_Toc169975427)

[8 Resultados 30](#_Toc169975428)

[9 Conclusiones de Proyecto 30](#_Toc169975429)

[10 Recomendaciones 30](#_Toc169975430)

[11 Competencias desarrolladas y/o aplicadas 31](#_Toc169975431)

[11.1 Competencias Específicas 31](#_Toc169975432)

[11.2 Competencias genéricas 32](#_Toc169975433)

[12 Fuentes de información 33](#_Toc169975434)

[13 Anexos 36](#_Toc169975435)

Índice de figuras

[Figura 1 Representación de un algoritmo de agrupamiento o clustering 12](#_Toc169954808)

[Figura 2 Agrupamiento jerárquico 13](#_Toc169954809)

[Figura 3 Clustering particional 15](#_Toc169954810)

[Figura 4 Operación de cruza de un modelo evolutivo. 24](#_Toc169954811)

# Introducción

Es el contenido global de lo que va a encontrarse en el documento. Incluye los aspectos relevantes de los antecedentes, del problema, los objetivos, la justificación y la hipótesis (si aplica).

# Descripción de la empresa u organización y del puesto o área del trabajo del estudiante.

El Instituto Tecnológico de Toluca pertenece al sistema de la Secretaría de Educación Pública conocido como Tecnológico Nacional de México. Esta red se encarga de formar profesionales en la Ciencia y la Tecnológica, así como impulsarlos a través de la creación de proyectos y su divulgación.

El ITT se ha consolidado como una de las universidades de calidad en el Estado de México con respecto a esta área. Han tratado de cubrir las áreas industriales y de servicios en varias regiones del país, vinculándose a empresas, organizaciones e instituciones. Promueven entre sus catedráticos y estudiantes valores como la justicia, independencia, democracia y solidaridad. Por eso procuran el deporte y la cultura como estrategia para mentes y cuerpos sanos.

Realizan actividades donde vinculan a sus estudiantes a proyectos con comunidades, así como la movilidad a otras ciudades o países. Y también imparten capacitación continua para sus egresados y personas en general que quieran seguirse preparando.

Misión

Ofrecer servicios de Educación Superior Tecnológica con calidad, equidad y pertinencia, orientados a la formación integral del ser humano mediante la oferta de programas educativos acreditados y que promuevan el desarrollo sustentable para la conformación de una sociedad justa y humana.

Visión

Ser una Institución de Educación Superior Tecnológica basada en valores y reconocida por ofrecer un servicio de calidad, mediante el desarrollo sostenido, sustentable y equitativo de su entorno y a través del alto desempeño de su personal.

# Problema a resolver

Actualmente, las organizaciones se enfrentan a desafíos significativos al intentar analizar grandes conjuntos de datos que contienen tanto variables numéricas como categóricas. Estos datos mixtos son comunes en muchos campos, desde el comercio minorista hasta la investigación científica, y la falta de herramientas adecuadas para procesarlos y agruparlos eficazmente representa un obstáculo importante.

Los métodos tradicionales de análisis de datos, como los algoritmos de agrupamiento estándar, suelen ser efectivos solo con un tipo de variable, ya sea numérica o categórica, pero no ambos. Esta limitación impide la identificación de patrones y tendencias clave en los datos mixtos, lo que puede llevar a interpretaciones incompletas y decisiones menos efectivas. La incapacidad de manejar estos datos de manera eficiente resulta en una visión parcial de las relaciones importantes entre las variables numéricas y categóricas, afectando negativamente la toma de decisiones estratégicas.

Además, la integración y comprensión de datos mixtos es crucial para diversas aplicaciones, desde el análisis de comportamiento del cliente hasta la investigación científica y la gestión operativa. Sin herramientas adecuadas, las organizaciones tienen dificultades para integrar datos de diferentes fuentes y obtener una comprensión holística. Esta deficiencia limita la capacidad para desarrollar estrategias efectivas y mejorar procesos operativos.

Existe una necesidad urgente de herramientas que puedan manejar eficientemente datos mixtos, permitiendo la identificación de patrones y tendencias que no son evidentes cuando se analizan tipos de datos por separado. Un sistema que utilice algoritmos avanzados, como el K-Prototypes, puede agrupar datos mixtos en clústeres significativos, mejorando la precisión del análisis y facilitando la toma de decisiones basada en datos.

Sin una solución adecuada, las organizaciones continuarán enfrentando dificultades para extraer valor significativo de sus datos, lo que puede resultar en decisiones subóptimas y limitar la capacidad de innovación. En un entorno empresarial competitivo, la capacidad de analizar y actuar sobre datos de manera efectiva es un diferenciador clave. La falta de herramientas adecuadas puede poner a las organizaciones en desventaja frente a competidores que sí pueden aprovechar completamente sus datos.

# Objetivos

## General

Implementación del algoritmo de agrupamiento K-Prototypes para encontrar la descripción de patrones en un conjunto de datos con datos numéricos y categóricos.

## Específicos

1. Estudio y análisis de algoritmos de agrupamiento (K-Prototypes)
2. Análisis y comprensión de la base de datos que será utilizada
3. Análisis estadístico de la base de datos que se utilizará
4. Diseño del prototipo del Software
5. Implementación en el lenguaje Phyton
6. Realización de pruebas
7. Presentación de resultados

# Justificación

Existen pocos algoritmos de agrupamiento que trabajen con datos de tipo mezclado (Numérico y categóricos) por esta razón, contar con esta implementación del algoritmo de agrupamiento permitirá, realizar estudios de análisis de datos, para cualquier conjunto de datos de este tipo. Muchos conjuntos de datos contienen una combinación de variables numéricas y categóricas, lo que presenta un desafío considerable para los métodos de agrupamiento tradicionales. La falta de herramientas adecuadas para analizar estos datos mixtos puede llevar a interpretaciones incompletas y decisiones menos efectivas.

La implementación del algoritmo de agrupamiento K-Prototypes permitirá analizar conjuntos de datos mixtos de manera más precisa y eficiente. Este proyecto se enfoca en desarrollar un software que utilice dicho algoritmo para identificar patrones y tendencias en los datos, lo que facilitará una comprensión más profunda de la información disponible.

Al permitir un análisis más completo de estos datos, se podrá mejorar la gestión de recursos, optimizar la seguridad y el rendimiento del sistema, y adaptar sus estrategias a las necesidades y comportamientos de los usuarios. La integración del algoritmo K-Prototypes es una herramienta accesible que permita a los usuarios realizar análisis complejos sin necesidad de una formación avanzada en análisis de datos.

El desarrollo de este software no solo solucionará un problema técnico significativo, sino que también abrirá nuevas oportunidades para el análisis de datos. Al ofrecer una solución robusta para el análisis de datos mixtos, el proyecto contribuirá a la toma de decisiones más informadas y estratégicas, mejorando así la eficiencia y efectividad operativa.

# Marco Teórico (fundamentos teóricos)

## Minería de datos

En la actualidad, las empresas que manejan grandes bases de datos buscan la manera de procesarlas, con la finalidad de encontrar aquello que les permita monitorear las ventas para realizar mercadeo dirigido y las tendencias para predecir situaciones futuras. En general, lo que se busca es encontrar los modelos o patrones que le den sentido a la enorme cantidad de datos. Sin duda, eso le compete a la minería de datos (MD), cuya finalidad es buscar información valiosa o conocimiento mediante el examen minucioso de una enorme cantidad de datos auxiliándose de técnicas y metodologías (algoritmos) diseñados para las tareas de clasificación, regresión, agrupamiento y detección de desviaciones entre otras.

El proceso de minería de datos involucra ajustar modelos o determinar patrones a partir de datos observados. Este ajuste normalmente es de tipo estadístico, en el sentido que se permite un cierto ruido o error dentro del modelo. Los algoritmos de minería de datos realizan en general tareas de predicción (de datos desconocidos) y de descripción (de patrones). Los objetivos de la predicción y la descripción son llevados a cabo por el uso de las tareas primarias de la minería de datos: clasificación, regresión, clustering o agrupamiento, sumarización, dependencia del modelo, detección de cambios y desviación. La mayoría de los métodos de minería de datos están basados en conceptos de aprendizaje de máquina, reconocimiento de patrones y estadística. El conjunto de algoritmos diferentes para resolver cada uno de estos problemas puede ser a menudo desconcertante tanto para los analistas de datos expertos, como para los novatos. Los desarrollos actuales de minería de datos se basan en la búsqueda de métodos que manejen de manera eficiente la clasificación de grandes bases de datos, con datos de diferente naturaleza o una mezcla de estos.

### Disciplinas relacionadas con minería de datos

Por definición, se considera el KDD como un campo interdisciplinario en el que se reúnen investigadores de diferentes campos. El término minería de datos ha sido comúnmente usado por estadísticos, analistas de datos y la comunidad MIS (Sistemas para el Manejo de Información), mientras KDD ha sido más usado por inteligencia artificial e investigaciones de aprendizaje de máquina. Los campos que están relacionados con las distintas partes del proceso KDD son:

Estadística

Proporciona una valiosa ayuda en el análisis de datos, pero generalmente no es suficiente y presenta algunos inconvenientes como son la mala adecuación a datos de tipo nominal, sus resultados pueden ser difíciles de interpretar y requiere que el usuario decida dónde y cómo analizar los datos. Sin embargo, juega un papel importante en algunos pasos dentro del proceso de KDD, sobre todo en la selección y muestreo de datos, en la minería de datos y en la evaluación del conocimiento extraído.

Inteligencia Artificial y Aprendizaje de Máquina

Estas disciplinas aportan algoritmos y métodos para el aprendizaje automático, lo que permite que los sistemas identifiquen patrones y realicen predicciones a partir de los datos. Los enfoques de aprendizaje supervisado y no supervisado son esenciales para desarrollar modelos predictivos y de clustering.

Bases de Datos y Sistemas de Información

El almacenamiento eficiente y la recuperación de datos son cruciales en la minería de datos. Las bases de datos relacionales y NoSQL, junto con los sistemas de gestión de bases de datos (DBMS), facilitan la organización y el acceso a grandes conjuntos de datos, soportando las operaciones de minería.

Matemáticas y Algoritmos

La teoría matemática subyace en muchas técnicas de minería de datos. Los algoritmos de optimización, la teoría de grafos y las matemáticas discretas son fundamentales para el desarrollo de métodos eficientes y efectivos para el análisis de datos.

Visualización de Datos

La capacidad de visualizar datos es crucial para interpretar y comunicar los resultados de la minería de datos. Herramientas y técnicas de visualización ayudan a presentar patrones complejos de manera comprensible y a facilitar la toma de decisiones basadas en datos.

Ciencias de la Computación

Las ciencias de la computación proporcionan los fundamentos técnicos y teóricos para el desarrollo de software y herramientas de minería de datos. Esto incluye desde el diseño de algoritmos hasta la implementación de sistemas escalables y eficientes.

## Clustering

Clustering es una técnica de aprendizaje no supervisado que se utiliza para agrupar un conjunto de objetos en subgrupos o clusters, de manera que los objetos en un mismo cluster sean más similares entre sí que con los objetos de otros clusters. La similitud entre objetos se define a menudo mediante una medida de distancia, como la distancia Euclidiana, aunque puede variar según el tipo de datos y el algoritmo utilizado.

El Clustering es un enfoque o método popular usado para segmentar grandes conjuntos de datos heterogéneos, dentro de pequeños subconjuntos homogéneos que puedan ser fácilmente manejados, analizados y modelados por separado. Los métodos de agrupamiento reparten un conjunto de objetos dentro de grupos, de tal forma que los objetos en el mismo grupo son más similares mutuamente que objetos en diferentes grupos conforme a algún criterio definido. El Clustering ha sido estudiado en diversas áreas, cada una tiene sus métodos para organizar los datos, en las áreas de análisis de datos y estadística se requiere a los datos en forma de números (cuantitativos). En aprendizaje de máquina, en forma de conceptos y simbolos (cualitativos). En minería de datos frecuentemente se trata con todos los tipos de datos o con la mezcla de estos: cuantitativos (numéricos) y cualitativos (categóricos o simplificación de simbólicos).

### Objetivos del Clustering

Descubrimiento de estructuras: Identificar patrones ocultos en los datos sin etiquetas predefinidas.

Reducción de dimensionalidad: Simplificar grandes conjuntos de datos agrupándolos en clusters representativos.

Detección de anomalías: Identificar puntos de datos que no pertenecen a ningún cluster, lo que puede ser útil para detectar fraudes o fallos.

Segmentación de mercados: Dividir un mercado en segmentos distintos para identificar grupos de clientes con comportamientos similares.

### Importancia del Clustering en el Análisis de Datos

El clustering es una herramienta fundamental en el análisis de datos y tiene aplicaciones en diversas disciplinas. En biología, por ejemplo, se usa para clasificar especies similares basadas en características genéticas. En el comercio electrónico, ayuda a segmentar a los clientes en grupos basados en su comportamiento de compra, permitiendo la personalización de campañas de marketing. En seguridad informática, se emplea para detectar patrones anómalos que podrían indicar ataques cibernéticos.

### Aplicaciones del Clustering

Biología y Medicina: Agrupamiento de genes con funciones similares, clasificación de enfermedades.

Marketing y Ventas: Segmentación de clientes, análisis de preferencias de consumidores.

Redes Sociales: Detección de comunidades en redes sociales, análisis de opiniones y sentimientos.

Seguridad y Fraude: Detección de transacciones fraudulentas, identificación de patrones inusuales en datos de seguridad.

Procesamiento del Lenguaje Natural: Agrupamiento de documentos o términos, análisis de sentimientos.

El clustering no solo organiza los datos en grupos manejables, sino que también facilita el descubrimiento de relaciones complejas y patrones en los datos que no son fácilmente visibles. Esto permite una comprensión más profunda y una toma de decisiones más informada en una variedad de campos.

## Tipos de algoritmos de agrupamiento (clustering)

Los algoritmos de agrupamiento reciben como entrada al conjunto de datos o la misma base de datos y sin supervisión de ningún tipo y de forma autónoma, agrupan los datos dentro de k grupos o clusters. Clustering es útil en ciertas aplicaciones de biologia, medicina, sociologia y otras donde no se encuentran bien definidas las clases. La figura 1 muestra la representación de un algoritmo de agrupación, la entrada del algoritmo se representa por el conjunto de datos u objetos y dependiendo del algoritmo de agrupamiento se solicita o no el parámetro k; la salida, es una etiqueta para cada uno de los elementos del conjunto de datos, la cual indica que dicho dato pertenece a cierta clase, grupo o cluster.

Tabla

Descripción generada automáticamente con confianza baja

Figura 1 Representación de un algoritmo de agrupamiento o clustering

Los algoritmos de agrupamiento conceptual desarrollados en aprendizaje de máquina agrupan datos con valores categóricos y también producen descripciones conceptuales de los clusters. La descripción conceptual de los grupos es importante en minería de datos, porque ayuda a la interpretación de los resultados del agrupamiento.

Los algoritmos de agrupamiento estadístico cuentan con algoritmos basados en la búsqueda de objetos que tengan conceptos iguales o similares y su eficiencia depende de buenas estrategias de búsqueda. Para los problemas de minería de datos que frecuentemente involucran muchos conceptos y espacios muy grandes de objetos, los métodos basados en conceptos pueden ser de potencial desventaja.

### Clustering jerárquico

Organiza los datos en una secuencia anidada de grupos. Una de sus características es el impacto visual del dendograma que habilita al analista de datos a visualizar los objetos mezclados en grupos o divisiones sucesivas de niveles de proximidad, los métodos de agrupamiento generalmente requieren solo el uso de la matriz de proximidad, estas técnicas son populares en las ciencias biológicas, sociales y de comportamiento por que se necesita construir taxonomías. Los algoritmos de clustering jerárquico tienen como objetivo unir grupos (cada grupo de objetos es manejado como si fuera un sólo objeto) para formar uno nuevo, tendiendo hacia un nivel de generalización mayor (métodos aglomerativos) o bien separando grupos formando nuevos sub-agrupamientos de ellos (métodos divisivos), tendiendo hacia un nivel mayor de particularización. Los algoritmos de clustering jerárquico descomponen una base de datos, en varios niveles de clustering. La descomposición jerárquica es representada por un dendograma o árbol, que divide iterativamente las bases de datos en subconjuntos más pequeños hasta que cada subconjunto consta de un objeto. En tal jerarquía, cada nodo del árbol representa un cluster de las bases de datos.

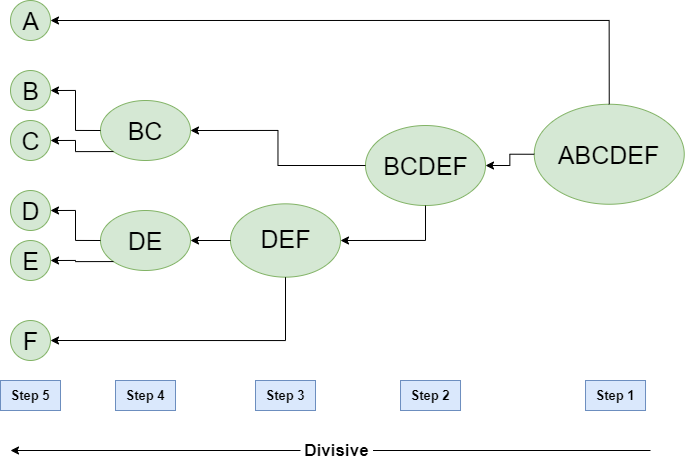


Figura 2 Agrupamiento jerárquico

### Clustering particional

Por su parte el clustering particional genera una simple partición de los datos, en un intento de recuperar los grupos naturales presentes en éstos, para ello cuentan con los datos en forma de matriz de patrones, donde las características han sido medidas en la escala ratio. Las técnicas de partición son usadas para aplicaciones de ingeniería y son apropiadas para una eficiente representación y comprensión de grandes bases de datos. El problema de agrupamiento puede formalizarse de la siguiente manera: Dados patrones en un espacio métrico de d-dimensiones, determinar una partición de los patrones en k cluster o grupos, tal que los patrones en un grupo sean más similares entre si que con otros patrones de diferentes grupos. El valor de k puede o no ser especificado.

Un criterio de agrupamiento como el error cuadrático puede adoptarse, el criterio puede clasificarse como global o local. El criterio global, representa cada grupo por un prototipo y asigna los patrones al grupo de acuerdo con el prototipo más similar. El criterio local, forma grupos utilizando una estructura local en los datos. Por ejemplo, los grupos pueden formarse por la identificación de las regiones de alta densidad en el espacio de patrones o por la asignación de un patrón y los i-ésimos k vecinos más cercanos del mismo grupo. Cada grupo es representado por su centro de gravedad y los objetos restantes son asignados al grupo cuyo objeto representativo sea más cercano.

Estos algoritmos inician con una partición inicial de las bases de datos y entonces usan una estrategia de control iterativa para mover los objetos de un grupo a otro, tratando de optimizar la calidad de los clusters durante el agrupamiento. Pero estos algoritmos presentan las siguientes limitaciones: Pueden aplicarse en problemas en los cuales existen objetos descritos por atributos numéricos exclusivamente, no manejan ausencia de información y muchas veces es necesario especificar el número de grupos que formarán el agrupamiento. Estos algoritmos de partición consideran el agrupamiento como un problema de optimización y pueden sufrir de minimos locales.

Los algoritmos de partición construyen separan una base de datos en conjuntos de k grupos, donde k puede o no ser un parámetro de entrada. Cada grupo puede ser representado por un centro de gravedad k-means, o por uno de los objetos del grupo localizado cerca de su centro k-medoid, el resto de los objetos son asignados al grupo más cercano. La figura de todos los grupos encontrados por un algoritmo de partición es convexa como en la figura 2.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Figura 3 Clustering particional

### Algoritmos de agrupamiento de recorridos simples.

Estos algoritmos agrupan los objetos cercanos del conjunto de objetos utilizando como base una condición local de agrupamiento, recorriendo una sola vez todo el conjunto de objetos. Dentro de esta familia cabe destacar los algoritmos BIRCH y DBSCAN los cuales agrupan conjuntos de datos utilizando criterios locales.

## Algoritmos de Clustering

### El algoritmo DBSCAN

Este algoritmo trabaja sobre una noción basada en la densidad de los grupos y está diseñado para descubrir grupos de formas arbitrarias e irregulares en bases de datos espaciales con ruido, para ello utiliza estructuras de indices como árboles R\*-Tree y sus variantes. La idea es que cada rama del árbol no contenga más puntos de los preestablecidos en el parámetro "MinPts". Una de las limitaciones que presenta este algoritmo es el manejo de datos espaciales exclusivamente. El agrupamiento generado por el algoritmo puede variar si se cambia el orden de entrada de los objetos del conjunto de datos.

* Principio: Agrupa puntos que están densamente conectados.
* Ventajas: Detecta clusters de formas arbitrarias y maneja ruido (outliers).
* Limitaciones: Depende del parámetro de densidad y puede variar con el orden de los datos.

### El algoritmo BIRCH

Este algoritmo utiliza una estructura de árbol llamada CF-tree (cluster features tree), el cual almacena en sus nodos descripciones compactas de subgrupos (CF's), donde un CF se define como: número de objetos, suma lineal y suma de los cuadrados de todos los objetos en ese grupo. La metodologia que utiliza este algoritmo es la de generar un árbol de CF's con el conjunto de datos, para obtener las descripciones de los subgrupos y aplicar posteriormente un algoritmo de clasificación no supervisada para agrupar estos subgrupos. Asi, BIRCH transforma el problema de clasificar el conjunto de datos original en clasificar subgrupos de este conjunto de datos. BIRCH fue diseñado para agrupar grandes bases de datos con atributos numéricos, resumiendo la base de datos original hasta ajustarla al tamaño de la memoria principal disponible en el sistema en que se ejecuta, asi minimiza los costos de lectura y escritura de los datos. Las limitaciones que presenta este algoritmo son las siguientes: puede aplicarse en problemas donde se manejan datos numéricos exclusivamente, y no todas las funciones de distancia pueden aplicarse al utilizar este algoritmo, ya que la información manejada por los CF's permite manipular solamente una familia de medidas que puedan expresarse en términos del centroide, radio y diámetro, dependiendo del orden de entrada de los datos.

* Principio: Utiliza un árbol CF (Cluster Features) para agrupar datos.
* Ventajas: Eficiente en memoria y adecuado para grandes bases de datos numéricas.
* Limitaciones: Solo funciona con datos numéricos y tiene restricciones en las medidas de distancia.

### El Algoritmo K-Means

El algoritmo k-Means fue propuesto hace poco más de tres décadas y es uno de los algoritmos de agrupamiento más usados en una amplia variedad de áreas. k-Means es un algoritmo de agrupamiento restringido, por lo que recibe como parámetros el número de agrupamientos a formar y se encuentra definido sobre datos continuos, es decir, únicamente permite trabajar con objetos descritos por medio de un conjunto de atributos numéricos. Este algoritmo calcula iterativamente los centros de los agrupamientos mientras que al mismo tiempo minimiza una función objetivo. k-Means usa la distancia Euclidiana para comparar objetos y promedios para calcular los centros de los agrupamientos, lo que no le permite trabajar con atributos no numéricos. Frecuentemente es usado para inicializar otros algoritmos de mayor costo computacional (por ejemplo, el algoritmo EM) y básicamente consiste en los siguientes pasos:

1. Seleccionar aleatoriamente los centros iniciales.
2. Asignar cada objeto al agrupamiento cuya distancia con su centro sea mínima.
3. Recalcular los centros.
4. Repetir los pasos 2 y 3 hasta que no haya cambios en los centros para dos iteraciones consecutivas

Este algoritmo es simple, fácil de programar, y fácil de entender, pero tiene ciertas desventajas entre las que se encuentran: Únicamente puede ser aplicado a conjuntos de datos numéricos.

No garantiza una solución única, puesto que depende en gran medida de las condiciones iniciales. Por lo que solamente puede obtener soluciones locales. En la literatura se han propuesto algoritmos que resuelven las desventajas del k-Means. Algunos de estos algoritmos se enfocan en solucionar la dependencia de las condiciones iniciales, pero no permiten trabajar con datos mezclados; mientras que los restantes resuelven el problema de la aplicabilidad sobre conjuntos de datos mezclados, aunque siguen dependiendo de las condiciones iniciales.

* Principio: Divide datos en k clusters minimizando la suma de distancias cuadradas entre los datos y sus centros.
* Ventajas: Simple y fácil de implementar.
* Limitaciones: Solo trabaja con datos numéricos, depende de la inicialización y puede encontrar mínimos locales.

Estrategias para Mejorar la Robustez de k-Means ante Condiciones Iniciales:

Para resolver la dependencia de las condiciones iniciales del algoritmo k-Means se han propuesto soluciones que siguen distintas estrategias. Algunos algoritmos buscan semillas iniciales que permitan encontrar un mejor mínimo local, mientras que otros realizan una búsqueda global. A continuación, se presentan algunos de los trabajos que se encuentran en la literatura que atacan el problema de la dependencia de las condiciones iniciales del algoritmo k-Means.

Algoritmos de Búsqueda de Semillas Iniciales:

Se propone un algoritmo que permite refinar los centros iniciales para algoritmos de agrupamiento. Este algoritmo se basa en una técnica de estimación de modas de una distribución y opera sobre un pequeño subconjunto de objetos de la muestra original, por lo que únicamente requiere una pequeña porción del total de la memoria necesaria para almacenar todo el conjunto de datos, lo que hace a este algoritmo atractivo para conjuntos de datos muy grandes. Sostienen que al inicializar los centros de forma que estos queden cerca de la moda se pueden encontrar agrupamientos correctos más frecuentemente y el algoritmo itera menos veces, por lo que converge más rápido hacia un mejor mínimo local.

El algoritmo CCIA (Cluster Center Initialization Algorithm) permite calcular los centros iniciales para el algoritmo k-Means. Este algoritmo está basado en los hechos experimentales de que objetos muy similares forman el núcleo de los agrupamientos, por lo que estos objetos pueden ayudar en la búsqueda de los centros iniciales. El algoritmo genera un número mayor de agrupamientos que el deseado, por lo que agrupamientos similares son unidos usando un método de condensación de datos multiescala. Los centros de estos agrupamientos son usados como los centros iniciales con los que se ejecuta el algoritmo k-Means, lo que permite obtener mejores mínimos locales que los que se pudieran obtener si se ejecuta dicho algoritmo con centros iniciales aleatorios.

Se realiza un estudio comparativo de cuatro métodos de inicialización del algoritmo k-Means. Los métodos de inicialización comparados son: Inicialización Aleatoria, Forgy, MacQueen y Kaufman.

1. Inicialización Aleatoria: Aleatoriamente divide el conjunto de datos en una partición de k grupos y a partir de esta genera las semillas con las que se inicializa el algoritmo k-Means.
2. Forgy: Aleatoriamente elige k objetos del conjunto de datos (semillas) y asigna el resto de los objetos al grupo representado con la semilla más cercana.
3. MacQueen: Elige aleatoriamente k objetos del conjunto de datos. Asigna, siguiendo el orden de los objetos, el resto de los objetos al grupo con el centroide más cercano. Después de cada asignación realiza un recálculo de los centroides.
4. Kaufman: El agrupamiento inicial se obtiene mediante la selección sucesiva de objetos representativos hasta que se seleccionen k objetos. El primer objeto representativo es el que se encuentra más cerca del centro del conjunto de datos. El resto de los objetos representativos se selecciona de acuerdo a una regla heurística que elige las instancias que prometen tener el mayor número de objetos a su alrededor.

Existen algunas diferencias interesantes entre los cuatro métodos de inicialización. Kaufman es el único determinístico, el método de Inicialización Aleatoria genera una partición inicial independiente del orden de los objetos, y MacQueen genera una partición inicial dependiente del orden de los objetos. Estas diferencias implican distintos costos computacionales entre los cuatro algoritmos. En el estudio comparativo, concluyen que Kaufman obtuvo los mejores resultados, haciendo que el algoritmo k-Means sea más efectivo y un poco más independiente de las condiciones iniciales y del orden de entrada de los objetos. Kaufman mejora la velocidad de convergencia del k-Means en comparación con la Inicialización Aleatoria, aunque aún con este método, el algoritmo k-Means sigue obteniendo una solución local.

Algoritmos de Búsqueda Global

El algoritmo Global k-Means es un algoritmo de agrupamiento determinístico global que minimiza una función objetivo y es completamente independiente de las condiciones iniciales con las que se ejecute. Este algoritmo proporciona mejores resultados que el algoritmo k-Means en términos de una función objetivo, debido principalmente a que realiza una búsqueda global mientras que el algoritmo k-Means realiza una búsqueda local.

Otra ventaja de este algoritmo es que, para resolver el problema de k agrupamientos, resuelve todos los problemas intermedios con k agrupamientos, lo cual puede ser útil en aplicaciones donde se busca el número óptimo de grupos a formar, probando con distintos valores de k.

El algoritmo Global k-Means procede de forma incremental y utiliza el algoritmo k-Means como algoritmo de búsqueda local. Esto significa que, para resolver el problema con k agrupamientos, resuelve secuencialmente todos los problemas intermedios con k agrupamientos.

El algoritmo Global k-Means es un algoritmo que resuelve la dependencia de las condiciones iniciales del algoritmo k-Means, aunque tiene un alto costo computacional. Para mitigar este inconveniente, se han propuesto varias versiones rápidas de este algoritmo. Estas versiones rápidas reducen significativamente el tiempo de ejecución sin afectar notablemente la calidad de los resultados. La principal diferencia es que no consideran todos los objetos del conjunto de datos para cada nuevo centro, sino que utilizan un pequeño subconjunto de objetos apropiados, obtenido mediante heurísticas como la partición del espacio de datos usando árboles k-dimensionales. Esto permite reducir el número de objetos examinados para un nuevo centro.

Los algoritmos Global k-Means y sus versiones rápidas solucionan la dependencia de las condiciones iniciales del algoritmo k-Means y permiten obtener una solución global. Sin embargo, su aplicabilidad se limita a conjuntos de datos numéricos, lo que restringe su uso en problemas con datos mezclados.

Extensiones del algoritmo k-Means que permiten trabajar con datos mezclados:

El algoritmo k-Means ha demostrado su efectividad en la clasificación no supervisada restringida, bajo la suposición de que los objetos deben estar descritos por atributos que permitan el establecimiento de alguna métrica entre ellos. Sin embargo, esta suposición no siempre se cumple en ciencias denominadas suaves (Soft Sciences) como Medicina, Sociología y Política. En estas disciplinas, los objetos se describen mediante atributos numéricos, no numéricos y/o con ausencia de información. En tales casos, no siempre es posible definir una métrica para comparar objetos, y solo se puede determinar el grado de similaridad.

En este contexto, se han propuesto los algoritmos k-Means con Funciones de Similaridad y k-Prototypes. Ambos algoritmos realizan agrupamiento restringido y se describen con mayor detalle a continuación.

Algoritmo k-Means con Funciones de Similaridad:

El algoritmo k-Means con Funciones de Similaridad es un algoritmo de agrupamiento restringido diseñado para problemas donde no se pueden usar métricas tradicionales para comparar objetos, sino que solo se puede calcular la similaridad entre ellos. Este algoritmo es una extensión del k-Means y es muy útil en problemas de minería de datos y extracción de conocimiento.

El objetivo del algoritmo es formar agrupamientos donde los objetos de cada grupo sean lo más similares posible entre sí, y a la vez, cada grupo sea lo menos similar posible a los otros. Para lograrlo, usa una función de similaridad en lugar de una métrica como la distancia Euclidiana, integrando funciones de comparación entre atributos, lo que le da mayor flexibilidad para modelar el problema adecuadamente.

A diferencia del k-Means, este algoritmo usa objetos del mismo conjunto de datos como centros de los agrupamientos, llamados objetos representativos. Un objeto representativo es aquel que, en promedio, se parece más a los objetos de su agrupamiento y menos a los de otros agrupamientos. El proceso comienza seleccionando aleatoriamente k objetos representativos. Luego, el resto de los objetos se asignan al agrupamiento cuyo objeto representativo sea más similar a ellos. Después de agrupar todos los objetos, se actualizan los objetos representativos y el proceso se repite hasta obtener el mismo conjunto de objetos representativos en dos iteraciones consecutivas o alcanzar un número máximo de iteraciones.

Este algoritmo hereda uno de los principales problemas del k-Means: la fuerte dependencia de las condiciones iniciales con las que se ejecuta.

### El algoritmo k-modes

Permite extender k-means para ser utilizado con variables categóricas, a partir del cálculo de una medida de disimilitud que permita comparar observaciones categóricas, y la utilización de modas en lugar de medias para calcular los clusters. Por motivos de eficiencia computacional, el algoritmo k-modes sigue estos pasos para un número prestablecido de k clusters:

1. Selecciona k modas, una para cada cluster.
2. Asigna cada observación al cluster con la moda más cercana según la medida de disimilitud.
3. Actualiza la moda de cada cluster después de cada asignación.
4. Recalcula la medida de disimilitud después de que todas las observaciones han sido asignadas, y procede a reubicar aquellas más cercanas a otro cluster que al propio.
5. Recalcula las modas para los clusters afectados.
6. Repite los dos últimos pasos hasta que ninguna observación cambie de cluster después de un ciclo completo para todo el conjunto de datos.

Las principales características del algoritmo K-Modes son:

1. Utiliza una medida de disimilitud para comparar objetos categóricos.
2. Reemplaza el cálculo de promedios por el de modas.
3. Usa un método basado en frecuencias para actualizar las modas de los grupos.

El algoritmo K-Modes sigue una estructura similar al K-Means, pero está adaptado para manejar datos categóricos, lo que lo hace adecuado para aplicaciones donde los datos no son numéricos. La integración de K-Means y K-Modes en el algoritmo K-Prototypes permite trabajar con conjuntos de datos mixtos, combinando la efectividad de ambos métodos.

* Principio: Extiende K-Means para datos categóricos usando una medida de disimilitud basada en frecuencias.
* Ventajas: Agrupa datos categóricos de manera eficiente.
* Limitaciones: Solo trabaja con datos categóricos.

### Algoritmo k-Prototypes

El algoritmo k-Prototypes es un algoritmo de agrupamiento restringido que permite agrupar grandes conjuntos de datos mixtos. Constituye una integración de los algoritmos k-Modes y k-Means. Este algoritmo se basa en el paradigma del algoritmo k-means, preservando su eficiencia, pero eliminando la limitación de trabajar con únicamente datos de tipo numérico.

El algoritmo k-Modes fue la primera extensión del algoritmo k-Means orientada al agrupamiento de datos categóricos. Sigue la misma idea que el k-Means, pero con la principal diferencia en la medida de similitud utilizada para comparar objetos. Sus características principales incluyen el uso de una medida de disimilaridad para comparar objetos, el reemplazo del uso de promedios por el de modas, y un método basado en frecuencias para actualizar las modas. Este algoritmo fue diseñado exclusivamente para agrupar grandes conjuntos de datos categóricos.

El algoritmo k-Prototypes integra tanto al k-Means como al k-Modes para eliminar la limitación de trabajar únicamente con un solo tipo de datos. Asume que la medida de disimilaridad entre atributos numéricos se define por el cuadrado de la distancia Euclidiana y la medida de disimilaridad entre atributos categóricos se define por el número de incoincidencias de categorías entre objetos. La disimilaridad entre dos objetos se define como:

Donde es un peso utilizado para equilibrar las dos partes, evitando favoritismos entre los dos tipos de atributos. Un pequeño valor de indica que el agrupamiento está dominado por los atributos numéricos, mientras que un valor grande implica que los atributos categóricos dominan el agrupamiento.

El algoritmo k-prototypes agrupa objetos con atributos de tipo numérico y categórico de una manera muy similar a como lo hace el algoritmo k-means. Este método actualiza dinámicamente los k prototipos de los datos con el objetivo de maximizar la similitud dentro de los clústeres. La medida de similitud se deriva de ambos tipos de atributos: numéricos y categóricos. Cuando se introducen conjuntos de datos exclusivamente numéricos, el algoritmo funciona de manera idéntica al algoritmo k-means. A continuación, se muestra el algoritmo k-prototypes en el algoritmo 6.

**Funcionamiento**

Integración de K-Means y K-Modes:

* K-Means: Utiliza la distancia euclidiana para medir la similitud entre objetos numéricos.
* K-Modes: Utiliza la disimilitud de coincidencia para datos categóricos, basándose en la frecuencia de las categorías.

​**Proceso de Clustering:**

* Inicialización: Selección de 𝑘 prototipos iniciales.
* Asignación: Cada objeto se asigna al clúster cuyo prototipo es el más similar.
* Actualización: Recalcula los prototipos basándose en las modas (para datos categóricos) y los promedios (para datos numéricos).
* Iteración: Repite la asignación y actualización hasta la convergencia.

**Parámetro 𝛾:**

* Controla el balance entre atributos numéricos y categóricos.
* Un valor bajo de 𝛾 γ da más peso a los atributos numéricos, mientras que un valor alto prioriza los categóricos.

**Aplicaciones**

El K-Prototypes es útil en áreas donde los conjuntos de datos contienen tanto atributos numéricos como categóricos, como en estudios de mercado, análisis de clientes, y ciencias sociales. Permite identificar patrones y agrupamientos en datos heterogéneos, ofreciendo una visión más completa que los algoritmos que solo manejan un tipo de dato.

**Algoritmo k-prototypes**

1. Entrada: X // conjunto de datos
2. Entrada: γ // peso en datos categóricos
3. Salida: k // clústeres
4. Begin
5. selecciona k prototipos iniciales del conjunto de datos X, uno para cada clúster.
6. asigna cada objeto en X al clúster cuya evaluación de similitud sea la más cercana.
7. Evalúa la función de costo como la suma de medidas de similitud de los atributos numéricos y categóricos.
8. Al asignar todos los datos a los clústeres, se reevalúa la similitud de los datos respecto al dato prototipo del clúster. Si el dato tiene un dato prototipo más cercano respecto al que fue asignado previamente, entonces el dato es relocalizado en el otro clúster y se actualizan ambos clústeres.
9. se repite el punto 7 hasta que los datos no cambien de localización durante el ciclo completo de clustering.
10. End

## Investigaciones y proyectos usados con el algoritmo K-Prototypes

### Mejora de K-Prototypes mediante optimización evolutiva

Investigadores han propuesto mejoras al algoritmo K-Prototypes usando técnicas de optimización evolutiva como los algoritmos genéticos. Estos métodos buscan optimizar la selección de prototipos iniciales y el parámetro 𝛾 γ, mejorando la precisión y estabilidad del clustering.

### Aplicaciones en bioinformática

En el campo de la bioinformática, el K-Prototypes se ha utilizado para agrupar datos genéticos y clínicos, donde los atributos incluyen información tanto numérica (como la expresión génica) como categórica (como el tipo de enfermedad).

## Proyectos prácticos con el algoritmo K-Prototype

### Análisis de clientes en comercio electrónico

Un proyecto de una gran empresa de comercio electrónico utilizó K-Prototypes para segmentar a sus clientes en función de datos mixtos, como el historial de compras (numérico) y preferencias de producto (categórico). Este análisis permitió personalizar las ofertas y mejorar la experiencia del cliente.

Resultados: Se lograron aumentos significativos en la satisfacción del cliente y en las tasas de conversión de ventas.

### Estudio de patrones de viaje en transporte público

Un estudio en una ciudad metropolitana utilizó K-Prototypes para agrupar patrones de viaje basados en datos de tarjetas de transporte (numéricos) y tipos de usuarios (categóricos, como estudiante o trabajador). Este análisis ayudó a optimizar las rutas de autobuses y horarios.

Resultados: Mejoras en la eficiencia del servicio y una mejor distribución de los recursos del transporte público.

## Visual Studio Code (VS Code)

Visual Studio Code es un editor de código fuente desarrollado por Microsoft. Es ligero pero potente, proporcionando soporte para depuración, control de Git incorporado, resaltado de sintaxis, IntelliSense (completado de código inteligente), snippets y refactorización de código. Su extensibilidad permite añadir funcionalidades adicionales a través de extensiones, lo que lo hace muy versátil para diversos lenguajes y tareas de desarrollo. Es de vital importancia destacar que cuenta con una terminal propia en el editor, es de gran utilidad para ejecutar diferentes comandos al momento de desarrollar un sistema. No está limitado por el tipo de aplicativos que se pueden desarrollar Visual Studio Code va más allá adaptándose a las necesidades del proyecto. Todo lo ya mencionado convierte a Visual en un entorno que es tomado en cuenta para la inclusión de lenguajes y herramientas para análisis de datos multivariantes con coordenadas

paralelas.

## Python

Python es un lenguaje de programación interpretado, de alto nivel y propósito general. Su filosofía de diseño enfatiza la legibilidad del código con el uso de una sintaxis significativa. Python es conocido por su facilidad de aprendizaje y su amplia comunidad de desarrolladores, lo que proporciona una gran cantidad de recursos y bibliotecas para diversas aplicaciones, incluyendo análisis de datos, desarrollo web, inteligencia artificial, y más.

Guido Van Rossum, un programador de computación de los Países Bajos, creó Python. Python comenzó en 1989 en el Centrum Wiskunde & Informatica (CWI), en principio como un proyecto de afición para mantenerse ocupado durante las vacaciones de Navidad. El nombre del lenguaje se inspiró en el programa de televisión de la BBC “Monty Python’s Flying Circus” debido a que Guido Van Rossum era un gran aficionado del programa.

Historial de lanzamientos de Python

Guido Van Rossum publicó la primera versión del código Python (versión 0.9.0) en 1991. Dicha versión ya incluía buenas características, como algunos tipos de datos y funciones para la gestión de errores.

Python 1.0 se lanzó en 1994 con nuevas funciones para procesar fácilmente una lista de datos, como la asignación, el filtrado y la reducción.

Python 2.0 se lanzó el 16 de octubre de 2000, con nuevas características útiles para los programadores, como la compatibilidad con los caracteres Unicode y una forma más corta de recorrer una lista.

El 3 de diciembre de 2008, se lanzó Python 3.0. Incluía características como la función de impresión y más soporte para la división de números y la gestión de errores.

## Librerías de Python

Para el sistema se necesitará varias bibliotecas de Python que ayudarán en diferentes aspectos del desarrollo.

### NumPy

NumPy es fundamental para la computación científica en Python. Proporciona soporte para arrays y matrices multidimensionales y una colección de funciones matemáticas para operar con estos datos. Será utilizado para manejar y manipular datos numéricos de manera eficiente.

### Pandas

Pandas es esencial para la manipulación y análisis de datos. Ofrece estructuras de datos como DataFrames, que permiten trabajar con datos tabulares de manera flexible y potente. Utilizado para la carga, limpieza, y manipulación de datos. Facilita el análisis exploratorio de los datos.

### scikit-learn

Scikit-learn es una biblioteca robusta para el aprendizaje automático en Python. Ofrece una variedad de algoritmos de clasificación, regresión y agrupamiento, así como herramientas para preprocesamiento de datos, selección de modelos y evaluación. Se utilizará para implementar el algoritmo K-Prototypes y posiblemente otros métodos de agrupamiento para comparar resultados.

### Matplotlib y Seaborn

Descripción: Matplotlib es una biblioteca para la creación de gráficos en Python, mientras que Seaborn se basa en Matplotlib y proporciona una interfaz de alto nivel para dibujar gráficos estadísticos. Ambas bibliotecas serán usadas para visualizar los datos y los resultados del agrupamiento, ayudando a interpretar y presentar los patrones encontrados en los datos.

### SciPy

Descripción: SciPy es una biblioteca que utiliza NumPy para proporcionar muchas rutinas numéricas de alto nivel para resolver problemas de álgebra lineal, cálculo diferencial y ecuaciones diferenciales. Se puede utilizar para cualquier cálculo matemático avanzado necesario durante el análisis de datos.

### PyCaret

PyCaret es una biblioteca de aprendizaje automático de código abierto y de bajo código en Python, que permite a los usuarios experimentar y desplegar modelos con poco esfuerzo. Puede ser utilizada para simplificar y acelerar el proceso de desarrollo y validación de modelos de agrupamiento.

### Jupyter Notebook

Jupyter Notebook es una aplicación web que permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Facilita la documentación del proceso de desarrollo, análisis exploratorio de datos y la presentación de resultados de manera interactiva.

# Procedimiento y descripción de las actividades realizadas.

Se describen detalladamente las tareas generadas de cada una de las actividades definidas previamente en el cronograma. Incluye figuras, tablas, capturas de pantallas y todos aquellos elementos que den soporte a la realización del proyecto. Respetando la metodología definida para el desarrollo.

# Resultados

Considera manuales de usuario, reportes del sistema, manual técnico, estadísticas.

# Conclusiones de Proyecto

Las Conclusiones deben iniciar con los objetivos y los alcances planteados en el anteproyecto para conocer lo que se logró. Se desprenden directamente de los Resultados. Describen las limitaciones encontradas durante el desarrollo del proyecto.

# Recomendaciones

Incluye todos los aspectos de mejora, alcances futuros para seguir desarrollando sobre lo que se ha detectado como necesario. Las acciones que se deben hacer con los resultados.

# Competencias desarrolladas y/o aplicadas

Son los conocimientos y habilidades adquiridas y aplicadas durante el desarrollo del proyecto (Trabajo en equipo, comunicación, aplicación de una metodología y/o plataformas).

**COMPETENCIAS DESARROLLADAS Y/O APLICADAS**

En esta sección se deberán mencionar, tanto las competencias genéricas como las específicas que el estudiante ha desarrollado durante la realización de la residencia profesional, así como las competencias adquiridas durante su estancia en el Instituto Tecnológico de Toluca que le permitieron llevar a cabo su proyecto de residencia profesional.

## Competencias Específicas

Son aquellas que en su desarrollo definen, una cualificación profesional concreta, al sujeto en formación; es decir: saberes, quehaceres y manejo de tecnologías propias de un campo profesional específico.

El dominio de estas competencias específicas aportaría, a los sujetos en formación, los conocimientos teóricos y las aplicaciones tecnológicas, propios de cada profesión ya que están vinculados a lo que se denomina “el saber hacer profesional”. (Tobón, 2008)

Manejo de Conocimientos relativos a la ciencia, la tecnología y las humanidades en un campo profesional específico.

Conocimiento de las prácticas profesionales que se desarrollan y la evolución y trascendencia de la profesión.

Utilización con precisión terminología, simbología e instrumentos; así como el uso de lenguajes, algoritmos, heurísticos.

Creación y diseño de aparatos, máquinas y de objetos tecnológicos en general.

Planificación, establecimiento, organización y desarrollo de procesos de distinta índole.

Selección y uso de materiales, máquinas y herramientas.

Montaje de aparatos e instalaciones; así como, dar solución a problemas productivos y tecnológicos.

Actuación profesional, trabajar en equipo, apreciar el conocimiento y los hábitos del trabajo, cultivar la precisión y la curiosidad; así como, el entusiasmo y el interés en su quehacer profesional.

## Competencias genéricas

Son aquellas que se pueden aplicar en un amplio campo de ocupaciones, condiciones y situaciones profesionales dado que aportan las herramientas intelectuales y procedimentales básicas que necesitan los sujetos para analizar los problemas, evaluar las estrategias, aplicar conocimientos a casos distintos y aportar soluciones adecuadas. (Tobón 2008)

|  |  |  |
| --- | --- | --- |
| **Instrumentales** | **Interpersonales** | **Sistémicas** |
| Capacidad de análisis y síntesis.  Capacidad de organizar y planificar.  Conocimientos generales básicos.  Conocimientos básicos de la carrera.  Comunicación oral y escrita en su propia lengua.  Conocimiento de una segunda lengua.  Habilidades básicas de manejo de la computadora.  Habilidades de gestión de información (habilidad para buscar y analizar información proveniente de fuentes diversas.  Solución de problemas.  Toma de decisiones. | Capacidad crítica y autocrítica.  Trabajo en equipo.  Habilidades interpersonales.  Capacidad de trabajar en equipo interdisciplinario  Capacidad de comunicarse con profesionales de otras áreas.  Apreciación de la diversidad y multiculturalidad.  Habilidad para trabajar en un ambiente laboral.  Compromiso ético. | Capacidad de aplicar los conocimientos en la práctica.  Habilidades de investigación.  Capacidad de aprender.  Capacidad de adaptarse a nuevas situaciones.  Capacidad de generar nuevas ideas (creatividad).  Liderazgo.  Conocimiento de culturas y  costumbres de otros países.  Habilidad para trabajar en forma autónoma.  Capacidad para diseñar y gestionar proyectos.  Iniciativa y espíritu emprendedor.  Preocupación por la calidad.  Búsqueda del logro. |

# Fuentes de información

1. López, E. S. (2006). Algoritmos de Agrupamiento en la Minería de Datos. Instituto Nacional de Astrofísica, Óptica y Electrónica. Recuperado de <https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/628/1/LopezES.pdf>
2. Hernández, E. (2006). Tesis de Algoritmos de Clustering. Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. Recuperado de <https://www.cs.cinvestav.mx/TesisGraduados/2006/tesisEdnaHernandez.pdf>
3. Amazon Web Services. (n.d.). What is Python? Recuperado de <https://aws.amazon.com/es/what-is/python/>
4. Jain, A. K. (2010). Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 31(8), 651-666.
5. Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. IEEE Transactions on Neural Networks, 16(3), 645-678.
6. Everitt, B. S., Landau, S., Leese, M., & Stahl, D. (2011). Cluster Analysis. John Wiley & Sons.
7. Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. Nature, 585(7825), 357-362. <https://numpy.org/>
8. McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference, 51-56. <https://pandas.pydata.org/>
9. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830. <https://scikit-learn.org/>
10. Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in Science & Engineering, 9(3), 90-95. <https://matplotlib.org/>
11. Waskom, M. (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021. <https://seaborn.pydata.org/>
12. Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., ... & van der Walt, S. J. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods, 17(3), 261-272. <https://scipy.org/>
13. Ali, M., Jankowski, A., & Roy, A. (2020). PyCaret: An open source, low-code machine learning library in Python. arXiv preprint arXiv:2005.13547. <https://www.pycaret.org/>
14. López Esquivel, S. (2015). Algoritmos de Agrupamiento para Datos Mixtos. Instituto Nacional de Astrofísica, Óptica y Electrónica. Recuperado de https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/628/1/LopezES.pdf
15. Hernández, E. (2006). Análisis de Algoritmos de Clustering para Datos Categóricos. Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional. Recuperado de https://www.cs.cinvestav.mx/TesisGraduados/2006/tesisEdnaHernandez.pdf
16. Amazon Web Services. (s.f.). ¿Qué es Python? Recuperado de https://aws.amazon.com/es/what-is/python/
17. Huang, Z. (1998). Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. Data Mining and Knowledge Discovery, 2(3), 283-304.
18. Huang, Z. (1997). Clustering Large Data Sets with Mixed Numeric and Categorical Values. Proceedings of the First Pacific-Asia Conference on Knowledge Discovery and Data Mining, 21-34.
19. López, E. S. (2006). Algoritmo K-Prototypes para Clustering con Datos Mixtos. Tesis de Maestría, Instituto Nacional de Astrofísica, Óptica y Electrónica. Retrieved from https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/628/1/LopezES.pdf
20. Hernández, E. (2006). Análisis Comparativo de Algoritmos de Clustering. Tesis de Doctorado, Centro de Investigación y de Estudios Avanzados del IPN. Retrieved from https://www.cs.cinvestav.mx/TesisGraduados/2006/tesisEdnaHernandez.pdf
21. Huang, Z. (1998). Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values. Data Mining and Knowledge Discovery, 2(3), 283-304.
22. Huang, Z. (1997). Clustering Large Data Sets with Mixed Numeric and Categorical Values. Proceedings of the First Pacific-Asia Conference on Knowledge Discovery and Data Mining, 21-34.
23. Siers, M. J., & Soria, E. M. (2017). Optimizing the K-Prototypes Clustering Algorithm Using Genetic Algorithms. Applied Soft Computing, 55, 196-207.
24. Wang, H., & Luan, Y. (2015). Clustering Mixed-Type Data Using K-Prototypes Algorithm. Journal of Bioinformatics and Computational Biology, 13(1), 1550004.
25. López, E. S. (2006). Algoritmo K-Prototypes para Clustering con Datos Mixtos. Tesis de Maestría, Instituto Nacional de Astrofísica, Óptica y Electrónica. Retrieved from <https://inaoe.repositorioinstitucional.mx/jspui/bitstream/1009/628/1/LopezES.pdf>
26. Hernández, E. (2006). Análisis Comparativo de Algoritmos de Clustering. Tesis de Doctorado, Centro de Investigación y de Estudios Avanzados del IPN. Retrieved from <https://www.cs.cinvestav.mx/TesisGraduados/2006/tesisEdnaHernandez.pdf>

# Anexos

Carta de autorización por parte de la empresa u organización para la titulación y otros si son necesarios.

**FORMATO GENERAL**

1. Tipo de fuentes: Arial
2. Tamaño: 12 texto, 13 subtítulos y 14 títulos.
3. Alineación: Justificada (Referencias alineadas a la izquierda)
4. Márgenes: superior 2.5 cm. , inferior 2.5 cm., izq. 3.0 cm y der. 2.5 cm.
5. Paginación: abajo a la derecha a partir de la Introducción
6. Interlineado: 1.5
7. Las citas textuales si tienen cinco o más renglones, utilizar sangría
8. Títulos, subtítulos y encabezados no llevan puntuación
9. Utilizar esquema de numerado decimal, máximo tres niveles. Ejem.

**3 Marco Teórico**

**3.1 Bases de datos**

**3.1.1 Oracle**

1. Los Anexos se ordenan con letras mayúsculas de la A a la Z.
2. Encabezados: Los títulos y subtítulos en negritas.
3. Los títulos inician en una nueva hoja