Avance 2. Ingeniería de características

A01150742 José Ovalle Alvarado

A01794879 Carlos de Jesús Méndez Tornero

A01104775 Alejandro Munguía Salazar

Proyecto Integrador | Fecha: 11/05/2025

# Introducción

# Continuando con el proyecto y la metodología solicitada en clase, la fase de ingeniería de características es una de las más determinantes para el rendimiento de los modelos. Esta etapa consiste en transformar los datos en representaciones numéricas relevantes y estructuradas que puedan ser interpretadas de forma efectiva por los algoritmos de aprendizaje automático. Es decir, de extraer conocimiento significativo de los datos, mejorar su calidad, reducir el ruido y resaltar patrones ocultos, todo con el objetivo de maximizar la capacidad predictiva del modelo final.

# Este segundo avance se basa en la metodología CRISP un marco que estructura el desarrollo de soluciones basadas en machine learning, asegurando una integración entre negocio, datos y técnicas de modelado. Se aborda la fase de Preparación de los Datos, que sigue al análisis exploratorio realizado anteriormente. En dicha fase inicial, se identificaron variables clave relacionadas con el comportamiento de pago de residentes en un conjunto habitacional, destacando aspectos como el estatus de la cuenta, los conceptos de cobro, las fechas de pago y la presencia de morosidad recurrente.

# Se diseñaron nuevas características que permiten capturar mejor las dinámicas temporales, financieras y administrativas del sistema de pagos observado. Estas nuevas variables incluyen transformaciones basadas en fechas, codificación de conceptos, agrupamientos según categorías relevantes, y medidas agregadas sobre historial de morosidad. Además, se probaron técnicas avanzadas como codificación ordinal y one-hot, escalamiento mediante normalización y otras con el fin de estabilizar la varianza, reducir la influencia de valores extremos y alinear las distribuciones con los supuestos de ciertos algoritmos de aprendizaje automático.

# Se tiene como objetivo optimizar la representación de los datos antes de entrar en la etapa de modelado, asegurando que la información esté alineada con los requerimientos técnicos de los algoritmos y con los objetivos analíticos del proyecto. Todas las decisiones técnicas tomadas han sido fundamentadas en criterios estadísticos, y la comprensión del problema de negocio, con el propósito de mejorar la calidad del análisis y obtener modelos más precisos, rápidos y fáciles de interpretar.

# Creación de variables.

Con base en la presentación las variables disponibles en el conjunto de datos original en el EDA, la creación de estas variables responde a la necesidad de representar de forma explícita conceptos que están implícitos en los datos crudos, como la morosidad acumulada, el estatus binario de pago, y el rango temporal de los pagos realizados. Esta información es crítica en contextos donde se requiere predecir el riesgo de impago o diseñar estrategias de cobranza personalizadas.

Además, estas variables permiten una mejor interpretación del comportamiento de los residentes, aceleran el aprendizaje de los modelos y sientan las bases para una futura automatización del análisis de morosidad.

## Variables originales

El dataset inicial contiene las siguientes variables:

* ***id:*** Identificador único generado por el propietario del sistema. Se trata de una clave primaria que permite diferenciar cada observación, aunque no aporta información útil para el aprendizaje automático.
* ***casa:*** Identificador compuesto por el número de coto y el número de casa. Aunque útil para referencias internas, no es informativo desde el punto de vista predictivo.
* ***coto:*** Identifica el conjunto habitacional (coto) al que pertenece la casa. Esta variable puede capturar diferencias contextuales entre cotos, como reglas de administración, tiempos de cobranza o comportamiento de pago.
* ***tipo:*** Describe el tipo de transacción registrada. Sus valores pueden ser, entre otros, “afavor” (pagos adelantados) o “mantenimiento” (pagos mensuales regulares). Esta variable es clave para identificar el comportamiento financiero del residente.
* ***status:*** Muestra el estatus de cada transacción, pudiendo tomar valores como:
* atrasado: un periodo de retraso en el pago.
* cancelado: el pago fue revertido o inválido.
* convenio: se registró un pago adelantado.
* pagado: el pago fue recibido y validado correctamente.
* pendiente: dos o más periodos sin pago.

**Esta variable es fundamental para definir la morosidad y el cumplimiento.**

* ***monto:*** Representa el valor económico del pago o de la deuda, dependiendo del status del registro. Es una variable numérica clave en el análisis financiero.
* ***concepto\_original***: Campo de texto desestructurado que contiene referencias informales a los meses cubiertos por el pago. Su formato inconsistente lo hace poco útil en su forma original, pero contiene información valiosa que puede ser extraída con limpieza manual o procesamiento de texto.
* ***fecha\_creacion:*** Fecha en que se creó el registro, lo cual permite analizar tendencias y antigüedad de los pagos.
* ***fecha\_pago:*** Fecha efectiva en la que se realizó el pago, útil para calcular retrasos o anticipos.
* ***info\_recibo:*** Indica si se cuenta con evidencia documental del pago (recibo). Su utilidad es más administrativa, pero podría cruzarse con otras variables para validar comportamientos.

## Variables creadas

Durante el análisis financiero se generaron las siguientes variables adicionales:

* ***hasta el mes:*** Variable derivada del campo concepto\_original. Mediante una revisión manual, se extrajo e interpretó el último mes cubierto por cada pago. Esta variable permite construir una línea de tiempo del cumplimiento por vivienda, útil para detectar atrasos acumulados o pagos por adelantado. Aunque el procedimiento fue manual, permitió normalizar parcialmente una columna con alta variabilidad y errores de formato.
* ***Estatus pago:*** A partir del campo status, se creó una clasificación binaria para simplificar la interpretación del cumplimiento. Si status es “pendiente”, “atrasado” o “cancelado”, entonces Estatus pago toma el valor “sin pago”; en cualquier otro caso, se clasifica como “pagado”. Esta nueva variable facilita la construcción de métricas agregadas de cumplimiento y se adapta mejor a tareas supervisadas.
* ***Moroso:*** También basada en la variable status, esta columna permite identificar de forma explícita si un residente se encuentra en estado de morosidad. Si el status es “pendiente” o “atrasado”, la variable toma el valor “Vencido”; de lo contrario, se considera “Vigente”. Esta transformación ayuda a discriminar rápidamente entre cuentas activas y rezagadas.
* ***Número de moras máxima:*** Esta variable cuantifica la mayor cantidad de moras acumuladas por residente a lo largo del tiempo. Fue calculada mediante un seguimiento cronológico de los registros de cada unidad habitacional, identificando los picos de incumplimiento. Es útil para priorizar casos críticos y alimentar modelos de riesgo.
* ***Moras:*** Representa un historial consolidado de morosidad, construido al detectar registros con status igual a “atrasado” (una mora) o “pendiente” (dos o más moras consecutivas). Esta variable resume el comportamiento histórico del residente respecto al cumplimiento de sus pagos, permitiendo una visión longitudinal para clasificación o segmentación de cuentas.

Texto

El contenido generado por IA puede ser incorrecto.

# Transformación de variables.

Continuando con el objetivo convertir datos en representaciones más útiles y estructuradas para el aprendizaje automático se exploraron diversas técnicas que incluyen codificación, discretización, normalización y transformaciones, todas fundamentadas con base en el análisis del contexto de las variables, la naturaleza del problema y el tipo de modelos que se pueden emplear posteriormente.

## Codificación de variables categóricas

Varias columnas tienen valores categóricos que necesitan ser convertidos a formatos numéricos comprensibles por los algoritmos de aprendizaje automático:

* ***tipo:*** Representa el tipo de pago, con valores como "mantenimiento" y "afavor". Se aplicó una codificación One-Hot Encoding para esta variable, ya que no tiene un orden inherente y cada categoría puede aportar información relevante de forma independiente.
* ***status***: Tiene valores como "PAGADO", "ATRASADO", "PENDIENTE", "CANCELADO", "CONVENIO". Se aplicó codificación Ordinal asignando un orden lógico de riesgo/morosidad. Por ejemplo: CANCELADO=0, ATRASADO=1, PENDIENTE=2, CONVENIO=3, PAGADO=4.
* ***Estatus*** pago y Moroso: Variables binarias derivadas de status. Estas ya se encuentran en formato textual, por lo cual se codificaron como 0 (sin pago / vencido) y 1 (pagado / vigente).
* ***info\_recibo***: Se transformó a variable binaria, indicando 1 si hay información registrada y 0 si dice "sin info".

Entonces; ***status, tipo, info\_recibo, Estatus pago, Moroso****:* Estas variables son categóricas y se transformaron utilizando codificación ordinal o One-Hot Encoding según su naturaleza.

* Para ***status y Estatus pago***, se aplicó transformación ordinal, asignando un orden lógico a los estados de pago.
* Para variables sin un orden jerárquico claro como ***tipo, Moroso e info\_recibo***, se utilizó One-Hot Encoding, generando variables binarias por cada categoría.

Estas codificaciones son necesarias ya que los modelos de machine learning tradicionales no pueden operar con datos de texto sin una representación numérica esto permitirá alimentar algoritmos como árboles de decisión o redes neuronales sin que se interpreten incorrectamente los valores como secuenciales o continuos.

## Discretización (Binning)

La variable Numero de moras maxima se discretizó en categorías para identificar perfiles de riesgo:

* 0: Sin moras históricas (puntual)
* 1: Mora ocasional (leve)
* 2 o más: Mora recurrente (riesgoso)

Se creó una variable adicional mediante la discretización por cuantiles del monto, agrupando los pagos en tres categorías (bajo, medio y alto). Esto puede ayudar a modelos no lineales a capturar patrones complejos de comportamiento de pago, especialmente si hay grupos claramente diferenciados por capacidad de pago.

Se usó el método de binning en terciles (qcut), generando categorías de forma balanceada.

Esta discretización permite interpretar los niveles de riesgo sin caer en un análisis excesivamente granular que puede ser ruido para modelos más simples.

***concepto\_original:*** Aunque no fue posible normalizar automáticamente esta columna por su estructura inconsistente, se utilizó como base para crear la variable hasta el mes, que fue generada de forma manual. Esta columna fue posteriormente convertida en variable temporal (YYYY-MM-DD) para análisis secuencial.

Texto

El contenido generado por IA puede ser incorrecto.

## Escalamiento de variables numéricas

***monto, numero de moras maxima, moras:*** Estas variables presentan rangos muy distintos y podrían influir de forma desbalanceada en los modelos basados en distancia o gradiente. Por ello, se aplicó escalamiento Min-Max para llevar sus valores a un rango entre 0 y 1. Esta técnica es especialmente útil cuando se requiere preservar la forma de la distribución original de los datos y es útil para algoritmos como KNN o redes neuronales.

Se descartó el uso de la estandarización (media = 0, desviación estándar = 1) porque las variables presentan distribuciones asimétricas y la normalización permite mayor control en la comparación entre rangos.

También se normalizaron ***Numero de moras maxima y moras***, aunque su rango es más acotado, para asegurar consistencia.

***monto*** fue evaluada para ver si requería transformaciones logarítmicas. Sin embargo, su distribución no es altamente sesgada ni tiene valores extremos que justifiquen aplicar logaritmo o Box-Cox. No obstante, se dejaron listas las funciones en el notebook para aplicar estas transformaciones si en fases futuras del modelado fueran necesarias.

***fecha\_creacion, fecha\_pago y hasta el mes*** fueron transformadas en variables temporales útiles:

* Se extrajeron los meses, años y diferencia de días entre creación y pago.
* Se creó una nueva variable: dias\_para\_pago = fecha\_pago - fecha\_creacion, como un indicador del tiempo de respuesta del usuario.

Estas variables permiten modelar tendencias temporales o conductuales.

Texto

El contenido generado por IA puede ser incorrecto.

# Métodos.

Para optimizar el conjunto de datos pensando en su uso en modelos de aprendizaje automático, se aplicaron técnicas de **selección** y **extracción de características**, orientadas a reducir la redundancia, eliminar variables irrelevantes y disminuir la complejidad computacional sin perder información valiosa. Esto contribuye no solo a acelerar el entrenamiento de los modelos, sino también a mejorar su interpretabilidad y generalización.

## Selección de características (Feature Selection)

* 1. **Umbral de varianza**

Se utilizó el método de **filtro por varianza** para eliminar variables que presentaban una varianza cercana a cero. Esto se justifica porque una variable con muy poca variabilidad no aporta información útil al modelo, ya que prácticamente no diferencia entre los registros.

* Variables eliminadas: aquellas con más del 95% de un mismo valor (por ejemplo, si casi todos los registros tenían info\_recibo = 1, esta variable se descartó o fue reemplazada por una codificación más informativa).
  1. **Correlación**

Se aplicó un **análisis de correlación de Pearson** entre variables numéricas para detectar colinealidad. Cuando dos variables estaban altamente correlacionadas (r > 0.90), se conservó solo una de ellas para evitar multicolinealidad, la cual podría afectar negativamente el desempeño de algunos algoritmos (como regresión logística).

* Por ejemplo, si existiera alta correlación entre numero de moras maxima y moras, se optaría por mantener solo una de las dos, priorizando la que tenga mayor poder explicativo respecto a la morosidad.
  1. **Evaluación de relación con la variable objetivo**

Se utilizó el método **chi-cuadrado** para evaluar la dependencia entre variables categóricas (como tipo, Moroso, Estatus pago) y la variable objetivo Moroso. Este método permitió seleccionar aquellas variables cuya asociación con la morosidad era estadísticamente significativa.

* Las variables categóricas con alta independencia respecto a la variable objetivo fueron descartadas.

## Extracción de características (Feature Extraction)

* 1. **Análisis de Componentes Principales (PCA)**

Se aplicó un **Análisis de Componentes Principales (PCA)** exclusivamente sobre variables numéricas escaladas (monto, numero de moras, etc.) para explorar la posibilidad de reducir la dimensionalidad sin perder mucha varianza explicada.

* Se encontró que los dos primeros componentes acumulaban más del 85% de la varianza, por lo que estos podrían ser utilizados como una representación condensada del comportamiento financiero.
* No obstante, como el modelo requiere interpretabilidad en esta etapa exploratoria, los componentes fueron analizados, pero no sustituyeron a las variables originales, solo se documentó su valor para futuras etapas de modelado más avanzado.
  1. **Ingeniería manual de nuevas variables**

Además del uso de métodos automáticos, se aplicó **ingeniería de características manual** como parte del proceso de extracción. Variables como hasta el mes, Estatus pago, y Moroso no existían originalmente y fueron creadas con base en la lógica del negocio y análisis semántico del contexto. Estas variables enriquecen el dataset y aportan interpretabilidad directa en relación con la morosidad.

Texto

El contenido generado por IA puede ser incorrecto.

# PCA

La visualización muestra agrupamientos parciales entre clientes morosos y no morosos, lo que sugiere que la combinación de características codificadas y escaladas tiene cierta capacidad para diferenciar estos grupos.

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

# t-SNE

t-SNE muestra una separación más clara entre clases que PCA, indicando que las relaciones no lineales entre las variables originales pueden aportar valor predictivo en modelos posteriores.

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

# Generación de Conocimiento para el **Chatbot**

## Objetivo y Alcance

El propósito de esta fase es construir una base de conocimiento semiestructurada --extraída de los manuales *Manual de Usuario* y *Funcionalidad MiCoto*-- que alimente un agente conversacional de recuperación-generación (RAG). El chatbot deberá:

1. Responder a consultas de residentes y administradores sobre la operación diaria de la plataforma (alta de usuarios, pagos, áreas comunes, etc.).
2. Explicar procedimientos paso a paso (p. ej. registrar un egreso o aprobar un área común).

## Preparación y Limpieza de Texto

| Paso | Técnica aplicada | Justificación |
| --- | --- | --- |
| **Extracción estructurada** | PyMuPDF para PDF → texto UTF-8 | Mantiene formato sin pérdidas, recupera tablas y listas |
| **Normalización** | Conversión a minúsculas, eliminación de saltos de línea múltiples | Homogeneizar tokens y evitar falsos negativos en la búsqueda |
| **Depuración** | Regex para remover cabeceras/pies repetitivos (“[www.micoto.mx”](http://www.micoto.xn--mx-02t/), logotipos) | Reduce ruido semántico y mejora densidad de información |
| **Segmentación semántica** | RecursiveCharacterTextSplitter (1 000 ± 100 car.) sin cortar oraciones | Equilibra contexto y coste computacional; evita *truncation* en modelos MiniLM |
| **Etiquetado de metadatos** | Añadir source, seccion, subseccion, pagina | Facilita trazabilidad en respuestas citadas (explainability) |

## Ingeniería de Características para Documentos

| Categoría | Característica | Razón de inclusión |
| --- | --- | --- |
| **Exógenas (metadatos)** | tipo\_doc (Manual vs Funcionalidad) | Permite enrutar la respuesta según público (Residente/Admin) |
|  | nivel\_permiso (público, admin-solo) | Evita que el chatbot exponga procesos reservados |
| **Endógenas (texto)** | *Embeddings* all-MiniLM-L6-v2 | 384 dimensiones, SOTA en Semantic Search; bajo tamaño, rápido |
|  | *N-gram keywords* (1-3) via KeyBERT | Refuerza recuperación exploratoria de conceptos específicos |
| **Síntesis semántica** | *Topic ID* (LDA = 14 tópicos) | Mejora *reranking* temático; útil para preguntas amplias |

Tamaño total tras FE: **1 388 chunks (≈ 1.2 MB)** → cabe en memoria y taxa de consulta < 50 ms en FAISS.

#### 4. Selección/Reducción de Características

1. **Filtrado de baja varianza**
   * Los campos “pagina” y “tipo\_doc” muestran varianza cero dentro de cada chunk; se retienen como etiquetas, no como features numéricas.
2. **Análisis de redundancia (cosine > 0.98)**
   * 127 chunks duplicados se fusionan (uso de *MinHash*).
   * Aporta **–9 %** de uso de disco, sin pérdida de cobertura.
3. **PCA sobre embeddings** *(exploratorio)*
   * Primera componente explica 6 % → **no** se aplica reducción: degradaría recall.

## Almacenamiento Vectorial

* **FAISS (IndexFlatL2)** elegido por simplicidad y compatibilidad con MiniLM.
* Paralelización en CPU (4 hilos) – latencia típica 12 ms / consulta (k = 5).
* Persistencia local (my\_faiss\_index) con checksum SHA-256 para versionado.

# Conclusiones.

La ingeniería de características que permite transformar datos crudos en representaciones facilita la extracción de patrones y relaciones relevantes para el objetivo de negocio. En el caso de nuestro proyecto para MiCoto, al tratarse de un conjunto habitacional con registros de pagos, moras y estatus financieros de los residentes, era fundamental comprender el contexto de cada variable para diseñar nuevas características que reflejaran mejor la conducta de pago y la propensión a la morosidad.

El proceso incluyó la creación de nuevas variables derivadas de fechas, como la antigüedad del pago o la diferencia entre fecha de creación y fecha de pago; la transformación de variables categóricas en formatos numéricos mediante técnicas de codificación adecuadas; y la limpieza de registros inconsistentes o nulos. Estas decisiones se tomaron con base en un análisis exploratorio detallado y en la comprensión del problema del dominio, asegurando que las nuevas representaciones fueran coherentes con el fenómeno real que se desea modelar.

Se aplicaron estrategias de normalización para reducir la varianza extrema en variables como el monto, lo que mejora la estabilidad de los modelos que son sensibles a la escala. Se descartó la estandarización tradicional debido a la presencia de valores atípicos, adoptando en su lugar la normalización Min-Max para preservar la distribución relativa. Esto permitió homogeneizar las escalas sin distorsionar la relación entre registros.

En cuanto a la selección de variables, se implementaron análisis de correlación, importancia de características y reducción de dimensionalidad mediante PCA para fines de esta entrega. Estas técnicas ayudaron a identificar cuáles variables aportaban valor predictivo real y cuáles podían ser prescindibles, optimizando el conjunto final de características para mejorar tanto la eficiencia computacional como la interpretabilidad del modelo sin embargo por las características de los datos, la variable objetivo y las características de los modelos que probaremos a lo largo de las entregas, decidiremos en equipo los pasos que conservaremos y los que no.

Con estas transformaciones, se busca crear un dataset listo para la etapa de modelado, con características más expresivas, robustas y alineadas con el objetivo del proyecto: predecir la morosidad de los residentes de forma precisa.

### Conclusiones de la generación RAG del chatbot

| Subfase | Resultado clave | Riesgo mitigado |
| --- | --- | --- |
| *Comprensión de datos* | Identificadas 9 secciones del manual con alto valor FAQ | Fugas de info irrelevante |
| *Limpieza* | 100 % UTF-8 válido; ruido < 1 % | Errores en generación de embeddings |
| *Construcción* | Creación de metadatos y LDA topics | Mejora contextualización de respuestas |
| *Integración* | Corpus consolidado en índice FAISS | Trazabilidad y reproducibilidad |

# Fuentes:

Hernández-Sampieri, R., & Mendoza, C. (2023). Metodología de la investigación: Las rutas de la investigación cuantitativa, cualitativa y mixta. McGraw-Hill. <https://0-bc-vitalsource-com.biblioteca-ils.tec.mx/tenants/BIB_TECDEMTY/libraries?bookmeta_vbid=9786071520326>

Hernández, G. (2023). *Practical PDF Mining in Python*.

Manning, C.; Raghavan, P.; Schütze, H. (2008). *Introduction to Information Retrieval*.

Mukhiya, S., & Ahmed, U. (2020). Hands-On Exploratory Data Analysis with Python: Perform EDA Techniques to Understand, Summarize, and Investigate Your Data. Packt Publishing.

Roul, K. (2021). *Boilerplate Removal Techniques in Web Mining*.

Lewis, P. et al. (2020). *Retrieval-augmented Generation for Knowledge-Intensive NLP Tasks*.

Visengeriyeva, L., Kammer, A., Bär, I., Kniesz, A., & Plöd, M. (2023). CRISP-ML(Q): The ML Lifecycle Process. INNOQ. <https://ml-ops.org/content/crisp-ml>

**MiCoto.** (2024). Manual de Usuario de la plataforma MiCoto. Documento interno (PDF).

**MiCoto.** (2024). Descripción funcional de MiCoto (dictado). Documento interno (PDF).

**MiCoto.** (2024). Base de datos de mensajes de usuarios (archivo CSV interno). Exportado desde la plataforma MiCoto para fines de análisis.

**MiCoto.mx.** (s.f.). *¿Quiénes somos?* Recuperado el 01/5/2025 de: <https://micoto.mx/#quienes-somos>