Pontificia Universidad Católica del Ecuador

Facultad de Hábitat, Infraestructura y Creatividad

Maestría en Sistemas de Información mención Data Science

Trabajo previo para la obtención del titulo de Magister en Sistemas de Información mención Data Science

Tema:

Análisis de factores que influyen en la retención de usuarios en aplicaciones móviles mediante análisis de datos.

Autor:

Helen Madeleine Correa Sisalema

Tutor:

Damián Aníbal Nicolalde Rodríguez Mg.

Quito – marzo 2026

AGRADECIMIENTO

Agradezco a Dios por todas sus bendiciones y por acompañarme y guiarme en mi camino personal, académico y laboral.

A mis padres, por su paciencia y amor incondicional por ser mi inspiración y motor para salir adelante en cada etapa.

A mis amigos, la familia que uno elije, gracias por compartir conmigo tantas alegrías.

A mis compañeros de estudio, por el apoyo incondicional y la predisposición para compartir conocimientos.

¡Muchas gracias a todos!

Tabla de contenido

Lista de Tablas

Lista de Ilustraciones

Introducción

1.1 Introducción

En la actualidad, las aplicaciones móviles gracias a los procesos de digitalización se han integrado en el día a día de los usuarios llegando a convertirse en un componente fundamental tanto en aspectos personales como en aspectos comerciales en la interacción entre usuarios y empresas. Para inicios de 2025 a nivel mundial existían aproximadamente 8,93 millones de aplicaciones (Turner, 2025), evidenciando la proliferación de apps y la intensa competencia por la atención de los usuarios.

El principal desafío para los desarrolladores y las empresas es la retención de usuarios, es decir, la capacidad de mantener a los usuarios activos y comprometidos con el uso constante en el tiempo de las aplicaciones. Diversas investigaciones mencionan que las apps experimentan una alta tasa de abandono después de la primera interacción, según datos de AppsFlyer (2020), el 45% de las desinstalaciones ocurren en las 24 horas posteriores a la instalación. Para mitigar esta pérdida, el análisis descriptivo es insuficiente; se requiere un modelado predictivo que identifique patrones de riesgo, transformando los datos de mercado y características técnicas en un indicador de alta retención clave de éxito y sostenibilidad de una aplicación móvil, dado que las elevadas tasas de abandono ocasionan pérdidas de recursos e ingresos significativos para las empresas.

Se propone analizar los factores que posicionan a una aplicación en un segmento de alta retención, mediante el uso de datos agregados de características técnicas y comportamiento de mercado, mediante la aplicación de modelos de Machine Learning. Como principal objetivo identificar qué variables tienen mayor impacto en la lealtad del mercado, predecir la probabilidad de que una aplicación móvil se encuentre en el segmento de alta retención y sugerir estrategias y sugerir estrategias para mejorar la fidelización. Este enfoque toma importancia en un contexto competitivo, donde el costo de adquisición de nuevos usuarios es elevado y retener a los existentes es una alternativa más rentable y sostenible (Boozary, Sheykhan, GhorbanTanhaei, & Magazzino, 2025).

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar y validar un modelo de Machine Learning capaz de predecir la probabilidad de éxito en la retención de una aplicación móvil mediante el uso de características-metricas de desempeño agregadas.

1.2.2 Objetivos específicos

- Identificar y caracterizar los patrones de desempeño (valoración, reseñas, instalaciones) asociados con la retención o el abandono (churn).
- Determinar la fuerza y dirección de la correlación entre las variables clave de uso (por ejemplo, numero de instalaciones, volumen de feddback útil para medir la validación social que se tiene de la aplicacion) y la tasa de retención.
- Implementar y evaluar diferentes modelos de Machine Learning (Regresión Logística, Random Forest, Gradient Boosting) seleccionando el algoritmo con mayor capacidad de generalización mediante métricas de AUC (Area Under the Curve), F1-score y Recall, priorizando esta última para minimizar los falsos negativos en la detección del churn temprano.
- Proponer estrategias de intervención basadas en los modelos predictivos que permitan mejorar la retención de usuarios y el Customer Lifetime Value (CLV).

1.3 Justificación

La retención en aplicaciones móviles es el principal desafío para los desarrolladores y las corporaciones en un mercado digital altamente competitivo. Diversos estudios indican que la mayoría de los usuarios abandonan las apps poco después de instalarlas, lo que impacta negativamente en la rentabilidad (AppsFlyer; 2020). Retener usuarios es más rentable que adquirir nuevos, y mejorar este indicador puede aumentar el Customer Lifetime Value y reducir costos operativos (Boozary, Sheykhan, GhorbanTanhaei, & Magazzino, 2025).

La Ciencia de Datos permite transformar grandes volúmenes de datos en conocimiento mediante técnicas como Machine Learning, análisis de cohortes y segmentación (Davenport, Harris, & Morison, 2010). Este proyecto se justifica por su capacidad para:

- Predecir el abandono mediante modelos de clasificación
- Identificar variables críticas que influyen en la fidelización

Los resultados pueden generalizarse a distintos tipos de aplicaciones (financieras, educativas, de salud, etc), y servir de base para el desarrollo de dashboards de monitoreo en tiempo real.

1.4 Preguntas de investigación

1.4.1 Pregunta principal

¿Cómo pueden los modelos de Machine Learning clasificar a las aplicaciones según su potencial de retención, usando datos agregados de desempeño, para permitir intervenciones de retención eficientes?

1.4.2 Preguntas secundarias

- ¿Qué variables técnicas y de mercado (valoración, volumen reseñas, instalaciones) tienen el mayor impacto al calificar una aplicación móvil como aplicación de alta retención?
- ¿Cómo se relación las características propias de una aplicación (tamaño, categoría, costo, etc) en su potencial de retención en un entorno altamente competitivo?
- ¿Qué algoritmos de Machine Learning (regresión logística, árboles de decisión, redes neuronales) proporcionan mejores resultados para predecir el abandono?
- ¿Qué estrategias de intervención personalizadas, sugeridas por los resultados del modelo de Machine Learning, se pueden proponer para optimizar la experiencia del usuario y aumentar el valor de vida en el mercado?

2 Desarrollo

A continuación, se presenta el marco teórico el cual es una revisión de literatura que aborda temas de retención, modelos predictivos y toma de decisiones basas en resultados. Esta revisión literaria tiene como objetivo integrar definiciones, hallazgos empíricos y enfoques metodológicos de diversos autores para brindar un contexto detallado el cual respalda las decisiones de esta investigación.

2.1 Marco Teórico

En las últimas décadas, la proliferación de aplicaciones móviles ha transformado la vida cotidiana, ofreciendo diversos servicios como comunicación, entretenimiento, finanzas y salud, entre otros; generado un mercado altamente competitivo donde la cantidad de descargas ya no garantiza el éxito, lo que realmente determina la sostenibilidad y el valor de una aplicación es

la retención de usuarios, es decir, la capacidad de mantener a los usuarios activos y comprometidos a lo largo del tiempo.

Diversos estudios muestran que la mayoría de las apps enfrentan un abandono significativo poco tiempo después de su instalación, la tasa promedio de retención a nivel global el primer día ronda el 25% de usuarios activos, pero descendiendo hasta cerca del 6% al día 30. Esta pérdida sustancial y acelerada de usuarios compromete severamente la viabilidad y sostenibilidad del negocio.

El principal desafío de esta investigación recae en comprender e identificar qué características técnicas propias de una aplicación, categorías de mercado e información agregada de interacción son factores determinantes para que una aplicación móvil logre situarse en el segmento de alta retención. Dado que el análisis se realiza a partir de métricas agregadas (como rating, volumen de reseñas e instalaciones) el enfoque se centra en el desempeño medible de la aplicación y no en el comportamiento individual de los usuarios. En este contexto, la Ciencia de Datos y las técnicas de Machine Learning representan una herramienta estratégica: mediante el uso de modelos de aprendizaje supervisado basados en métricas de desempeño real en la tienda de Google Play Store, es posible clasificar que variables del producto afectan el potencial de permanencia de una aplicación y detectar patrones que se asocien al éxito en el mercado de las aplicaciones móviles y anticiparse a la pérdida de relevancia en el mercado. De esta manera, se pueden diseñar e implementar estrategias de optimización de la app orientadas a mejorar la propuesta de valor y sostenibilidad en un mercado altamente competitivo.

Los conceptos que se presentan a continuación constituyen la base teórica para comprender el enfoque de esta investigación, orientada al uso de la Ciencia de Datos y las técnicas de Machine Learning

Retención: Es la capacidad de una aplicación móvil o servicio para mantener a sus usuarios activos y comprometidos a largo plazo (IBM, s. f.).

Deserción: El abandono de usuarios, o churn, es el fenómeno en el que un individuo deja de usar o desinstala una aplicación móvil (Sivakani et al., 2025). La literatura suele tratar el fenómeno churn mediante el uso de datos longitudinales y a nivel de individuo el presente estudio adopta un enfoque alternativo macro - analítico debido a la disponibilidad de datos públicos se emplean indicadores agregados de desempeño como variables proxy de retención; de esta forma la deserción no se medirá como un evento puntual de un individuo si no a una disminución en los niveles de fidelización y tracción (desempeño medido con indicadores

agregados capaces de atraer y generar interacción) de la aplicación móvil dentro de la tienda permitiendo modelar la sostenibilidad desde una perspectiva de éxito competitivo.

Machine Learning (ML): Es una rama de la Inteligencia Artificial, proporciona la capacidad de que los sistemas aprendan y se adapten a partir de grandes volúmenes de datos, identificando patrones y automatizando la toma de decisiones (Bishop, 2006). En el contexto de este estudio, el ML se aplicará mediante algoritmos de clasificación, cuya función primordial es clasificar a las aplicaciones según su potencial de retención, permitiendo determinar si una aplicación muestra señales de posicionamiento dentro de su mercado es decir si presenta patrones de aceptación.

Valor de Vida del Cliente- Customer Lifetime Value (CLV): es un concepto fundamental que representa la proyección del beneficio neto total que una empresa espera obtener de toda la relación futura con un cliente. Es una métrica crítica para la gestión de la rentabilidad que abarca la adquisición, retención y monetización (Dogan et al., 2025). En este proyecto el aumento del CLV está vinculado con la capacidad de las apps para permanecer en el grupo de alta retención y engagement. Mantener a los usuarios activos es más rentable que la adquisición de nuevos usuarios. Así, el análisis predictivo es un pilar fundamental, no predice conductas individuales, sino que estima el nivel de vulnerabilidad competitiva del activo digital.

Big Data: Hace referencia al análisis de grandes volúmenes de información (datos) que, por su velocidad, variedad y veracidad, superan la capacidad de las herramientas tradicionales de procesamiento, destaca el valor que aportan al transformar datos en conocimiento útil para la toma de decisiones (Davenport, Harris, & Morison, 2010). En el contexto de aplicaciones móviles, Big Data permite capturar y procesar métricas agregadas de desempeño para identificar casos de éxito o perdida de relevancia en el mercado

Algoritmos Predictivos: Técnicas como regresión logística, árboles de decisión y redes neuronales que permiten estimar la probabilidad de eventos futuros, En el marco de este trabajo, estos algoritmos se emplean para clasificar a las aplicaciones móviles dentro de niveles de retención (alta o baja).

2.1.1 El Modelo de Comportamiento agregado del usuario y el Ciclo de Vida del producto digital

El ciclo de vida de una aplicación móvil es diferente al de los productos físicos tradicionales estos últimos suelen tener fases claras y usualmente estables (introducción, desarrollo, madurez

y caída) por otra parte las apps al estar en un entorno altamente competitivo y volátil su ciclo es mas incierto la instalación y desinstalación pueden ocurrir en instantes la permanencia del producto depende de su capacidad de generar valor para los usuarios.

Desde el marketing digital, el ciclo de vida del cliente se divide en las siguientes etapas de seguimiento: adquisición, activación, retención, monetización y recomendación (YouAppi Marketing, 2025). Dichas etapas no pueden ser observadas ni mediables a nivel individual cuando se utiliza información agregada por esta razón la interpretación se la realiza a través métricas colectivas observables.

El análisis pase del tratamiento individual (user-level) a un tratamiento macro (app-level), donde las decisiones individuales se manifiestan como tendencias estadísticas agregadas. La fase de adquisición se mide en el volumen de instalaciones; la satisfacción es considerada calificación promedio de la app; y la interacción activa es considerada con el número de reseñas y su proporción respecto al total de descargas; estas métricas funcionan como indicadores indirectos del compromiso con el producto.

La fase de retención implica la capacidad para mantener relevancia y utilidad percibida después de la descarga inicial. Diversos estudios manifiestan que la permanencia depende más de la percepción de valor que de factores estáticos (Wu et al., 2022).

Cada acción individual (instalación, calificación y reseña) va generando un rastro digital colectivo el cual conforma indicadores estructurados que permiten medir la salud competitiva de la aplicación. Sivakani et al. (2025) señalan que sin acceder a registros personales se puede modelar el compromiso y la probabilidad de permanencia, al analizar grandes volúmenes de interacción transformando señales dispersas en métricas estratégicas.

En la era digital, la validación social (feedback de los usuarios) actúa como un mecanismo que influye en la decisión de nuevos usuarios y en la percepción de calidad de quienes ya forman parte de este mercado; el volumen de reseñas puede interpretarse como señal de mayor compromiso.

Las tiendas digitales utilizan algoritmos de visibilidad una aplicación con muchas descargas y bien valorada obtiene mayor exposición reforzando su ciclo de vida mientras aquellas con bajo engagement pierden rápidamente visibilidad y entran en un declive acelerado.

En este estudio Churn no se interpreta como la acción de un individuo que desinstala la app, sino como la disminución en la proporción de reseñas respecto a instalaciones (señal temprana de pérdida de relevancia competitiva).

2.1.2 Teoría de la Inversión-Satisfacción

La Teoría de la Inversión-Satisfacción sostiene que la satisfacción del usuario con la experiencia y la percepción de su inversión (tiempo, datos ingresados o personalización) constituyen predictores fundamentales de la retención. Un usuario satisfecho que reconoce haber invertido esfuerzo en la plataforma muestra menor propensión a migrar hacia la competencia (Zeithaml et al., 1996).

Usualmente esta teoría se aplica cuando se puede observar el comportamiento individual pero cuando tratamos en entornos con datos agregados, la inversión y la satisfacción deben reinterpretarse a través de indicadores observables a nivel de aplicación.

En este estudio, el indicador de satisfacción se mide con la variable Rating (evaluación cuantitativa del valor percibido por el usuario). Una calificación “alta” indica que la aplicación supera las expectativas del mercado por otra parte el volumen de Reviews (reseñas) se reinterpreta como una medida indirecta de inversión de esfuerzo pues para realizar una reseña el usuario dedica tiempo y formula una opinión participando en la comunidad digital.

Wu et al. (2022) muestran que las reseñas en línea son un mediador clave ya que influyen en la percepción de calidad y en la confianza de otros usuarios, desde la lógica del marketing digital, se convierte en un activo estratégico.

El indicador de Engagement (Reviews / Installs) permite traducir la inversión a términos agregados. Si una aplicación cuenta con muchas instalaciones pero pocas reseñas poco compromiso a lo largo del tiempo mientras una alta tasa de participación indica mayor estabilidad competitiva. La combinación de una valoración positiva y un alto ratio de participación constituye un proxy de lealtad agregada dentro del ecosistema digital (Davenport et al., 2010).

2.1.3 Métricas de Mercado como Proxies de Retención

El acceso libre a registros individuales suele estar limitado por la privacidad y la confidencialidad de los datos ante esta limitación la investigación recurre a variables proxy (medidas indirectas que permiten capturar el efecto que se desea analizar) Como señala Box

(1969), todo modelo es una simplificación de la realidad su valor radica en capturar patrones relevantes, aunque no refleje el fenómeno en su totalidad.

En este estudio, la retención no se mide como la permanencia de un usuario en la aplicación ni como la frecuencia de uso individual; se define como un estado de desempeño competitivo sostenido, medible mediante indicadores públicos de Google Play Store, la retención se interpreta como un fenómeno agregado que refleja aceptación, interacción y posicionamiento dentro del ecosistema digital.

Las variables que permiten medir la retención son:

- Rating: es la valoración promedio que tiene la aplicación una calificación alta indica cumplimiento de expectativas.
- Engagement (Reviews/Installs): Esta variable mide la participación dentro de la app.

Como sostienen Hyrynsalmi, Suominen y Mäntymäki (2015), la salud de una aplicación depende de diversos factores algunos son: la reputación acumulada, críticas de los usuarios, etc. El uso de métricas agregadas como proxies de retención integrando teoría de marketing, ciencia de datos e inteligencia de mercado en el análisis de aplicaciones móviles.

2.1.4 Modelos de Predicción de Churn basados en Machine Learning

El modelado de la retención de aplicaciones móviles se aborda comúnmente como un problema de clasificación binaria en Machine Learning, como principal objetivo se tiene clasificar si una aplicación logrará mantenerse en un segmento alto o si, por el contrario, entrará en una fase de pérdida de tracción competitiva. La literatura especializada señala que los modelos supervisados resultan particularmente efectivos en este tipo de tareas:

- Regresión Logística: Constituye la base de la clasificación binaria y ofrece la ventaja de la interpretabilidad de sus coeficientes, lo que facilita justificar las estrategias de negocio a partir de los resultados (Hosmer et al., 2013).
- Modelos No Lineales (Árboles de Decisión): Permiten capturar relaciones complejas y no lineales entre variables, ofreciendo en muchos casos una mayor precisión predictiva (Hossain, 2025).
- Redes Neuronales: Superan a los modelos tradicionales en escenarios con grandes volúmenes de datos y patrones altamente complejos, aunque presentan la limitación de una menor interpretabilidad (Gupta, Bharti, Pathik & Sharma, 2022).

En conjunto, estos enfoques proporcionan un marco robusto para clasificar si una aplicación logrará mantenerse en un segmento alto.

Diversos estudios han analizado el fenómeno del abandono desde distintas perspectivas y sectores por ejemplo en el ámbito de las telecomunicaciones, Kulkarni, Patil, Patil y Bhoite (2019) aplicaron técnicas de Machine Learning para predecir el churn, el modelo se basó en el uso de variables contractuales, sus resultados demostraron que la Regresión Logística constituye una herramienta robusta para la interpretabilidad, con una exactitud del 80.38 %. Este hallazgo justifica la inclusión de dicho algoritmo en el presente estudio como modelo de referencia inicial, permitiendo comparar su desempeño frente a métodos más complejos y no lineales.

Wu, Zhang, Zhu, y Liu (2022) centran su investigación en identificar los factores que influyen en el uso de aplicaciones móviles de salud evidenciando que la e-satisfacción y las tienen impacto positivo sobre la intención de permanecer en la app. Este aporte respalda la decisión de utilizar Rating y el volumen de Reviews como componentes fundamentales de la variable proxy de retención.

Para seleccionar los algoritmos de análisis consideramos lo expuesto por K Kavitha, Kumar, Kumar y Harish (2020) quienes compararon Regresión Logística, Random Forest y XGBoost para la prediccion de abandono en telecomunicaciones, teniendo como principal resultado que los modelos basados en árboles de decisión ofrecen mejor rendimiento al capturar relaciones no lineales entre variables.

La literatura usualmente se centra en el abandono temprano según AppsFlyer (2020), cerca del 45 % de las desinstalaciones ocurre en las primeras 24 horas. Pese a que este estudio no cuenta con información minuto a minuto, el uso de datos agregados (valoraciones, descargas y reseñas) actúan como filtro de retención.

La evidencia académica respalda tanto la elección de variables como de algoritmos, fortaleciendo la validez de analizar la retención desde una perspectiva agregada basada en el posicionamiento de mercado. La revisión evidencia varios vacíos que este proyecto busca abordar:

La revisión del marco teórico justifica la selección de una metodología cuantitativa basada en Data Science y Machine Learning, orientada a la clasificación y la interpretabilidad. El enfoque será de Modelado Predictivo (Clasificación Binaria).

Tabla 1 *Justificación modelos*

Algoritmo	Justificación
Regresión Logística	Los coeficientes permiten responder la pregunta sobre *qué variables influyen más* en el modelo
Random Forest / GBM	Seleccionados por su eficacia predictiva demostrada en la literatura para problemas de *churn*. Su comparación directa permitirá seleccionar el modelo más robusto (mayor AUC/F1-score).

La combinación de métodos asegura robustez y precisión, mientras que la comparación de métricas como AUC y F1-score permitirá seleccionar el modelo más confiable.

El desarrollo de este marco teórico establece una base epistemológica que vincula la retención con modelos predictivos, entendiendo la retención como un proceso impulsado por la satisfacción del usuario, el valor de uso percibido y la interacción social en entornos digitales; aunque no se puede capturar de forma directa el abandono (churn) debido a la naturaleza de los datos de Google Play Store (datos transversales), se emplean próxies metodológicas (basadas en métricas agregadas) como representaciones válidas del fenómeno que se desea estudiar (retencion-abandono).

La revisión conceptual previamente descrita permite integrar la Teoría de la Inversión-Satisfacción, que explica el "porqué" del compromiso del usuario, con la estructura lógica de este estudio. Dado que no se dispone de datos a nivel individual, se emplean métricas agregadas para representar de forma indirecta el abandono (no se mide como un evento aislado, sino como la incapacidad de una aplicación para mantener estándares competitivos en el mercado). Al emplear estas métricas con modelos de aprendizaje supervisado como Random Forest y Gradient Boosting, capaces de clasificar y anticipar patrones de comportamiento el estudio pasa de ser descriptivo a tener la capacidad de explicar factures de causalidad en la retención. Por lo tanto el estudio convierte los datos en información útil (con fundamentos) para la toma de decisiones estratégicas.

2.2 Metodología

Es importante recalcar que el presente estudio se basa en un dataset de naturaleza transversal y a nivel de información agregada los datos son de un solo periodo de tiempo para múltiples aplicaciones móviles no posee información de individuos, de tal forma el análisis no rastrea el ciclo de vida de un usuario específico, sino que evalúa indicadores de desempeño en la tienda de aplicaciones que funcionan como métricas agregadas de éxito dentro del mercado. Cuando hablamos de retención se hace referencia a la capacidad de una aplicación para mantener un posicionamiento competitivo en el mercado.

La investigación se desarrolla bajo un enfoque cuantitativo con un diseño correlacional-predictivo. Para llevar a cabo el análisis de forma ordenada y eficaz, se utiliza la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining). Esta metodología ofrece una guía paso a paso que ayuda a conectar lo que se quiere entender del comportamiento de los usuarios con la construcción y puesta en marcha de modelos de Machine Learning. En otras palabras, permite pasar de una idea de negocio a soluciones basadas en datos de forma estructurada, revisando y mejorando cada etapa del proceso.

2.2.1 Fases CRISP-DM Aplicado

1. **Comprensión del Negocio:** En esta etapa se traduce la necesidad de retención en un problema de clasificación. Dado que el dataset empleado es el de Google Play Store la retención no puede medirse a nivel individual, el análisis se dirige a identificar aplicaciones con alta capacidad de retención. Se define el éxito de la app (High Retention) como la capacidad de sostener una calificación elevada y un nivel de engagement superior a la mediana del mercado.
2. **Comprensión de los Datos:** El dataset usado contiene información agregada de aplicaciones. Los datos son de tipo transversal (cross-sectional). Se realizó un análisis exploratorio (EDA) para entender la distribución de las variables Rating, Reviews e Installs.
3. **Preparación de los Datos:** Se realiza la limpieza y transformación de los datos para el modelado, asegurando que la base de datos refleje con fidelidad el comportamiento del mercado de aplicaciones móviles. Esta etapa incluye:
    - **Ingeniería de Características:** Creación de la variable objetivo High_Retention mediante una lógica booleana que combina satisfacción y

compromiso (Engagement_Rate), es importante mencionar que en principio para evitar incurrir en data leakage se empleó una separación metodológica donde excluimos del conjunto de entrenamiento a Engagement_Rate (Reviews / Installs) con la finalidad de que el modelo no repita una operación matemática fija si no que aprenda patrones de retención real. Un análisis más profundo evidenció un riesgo de dependencia estructural, dado que los componentes Reviews e Installs permanecían integrados en el modelo.

- **Tratamiento:** Posteriormente a las variables originales Reviews e Installs se les aplicó un proceso de estandarización mediante StandardScaler para evitar sesgos debido a sus magnitudes y que sus escalas sean comparables. Si bien el Factor de Inflación de la Varianza (VIF < 1.7) confirmó la ausencia de multicolinealidad lineal entre predictores, se determinó que esta métrica no es suficiente para descartar el data leakage de tipo no lineal.
- **Estrategia de Robustez:** Debido a que las variables Reviews e Installs participan directamente en la construcción de la variable objetivo (Retención), se consideró necesario evaluar si el desempeño de los modelos podía estar condicionado por esta relación estructural. Para ello se implementó un Escenario de Robustez que consistió en repetir el proceso de entrenamiento y evaluación excluyendo dichas variables del conjunto de predictores. Este procedimiento permitió verificar si el modelo mantenía capacidad predictiva utilizando únicamente atributos intrínsecos de la aplicación (categoría, tamaño y precio), garantizando que el aprendizaje refleje patrones de comportamiento genuinos y no una mera reproducción de la fórmula del objetivo.

4. **Modelado:** Se implementan algoritmos de clasificación supervisada seleccionados por su robustez en la literatura de churn:
   - Regresión Logística: Para garantizar la interpretabilidad de los factores influyentes.
   - Random Forest: Para capturar relaciones complejas y no lineales, buscando mayor precisión.
   - Gradient Boosting: Para capturar relaciones no lineales complejas
5. **Evaluación:** Los modelos se validan utilizando métricas de desempeño objetivas:
   - AUC (Area Under the Curve) y F1-score: Para medir la capacidad global del modelo.

- Recall: Métrica prioritaria para minimizar los falsos negativos, asegurando que se detecte a la mayor cantidad posible de usuarios en riesgo de abandono.

6. **Despliegue:** El modelo se concibe como una herramienta analítica de apoyo para la toma de decisiones estratégicas en desarrollo y marketing móvil. dado que trabaja con datos agregados, debe interpretarse como un indicador de salud de mercado.

2.2.2 Técnicas de Recolección de Datos

La recolección de datos no rastrea la actividad individual en tiempo real (logs internos), sino a través de la extracción y procesamiento de indicadores de desempeño agregados del mercado. Las técnicas aplicadas son:

Extracción de Indicadores de Mercado: consiste en obtener registros técnicos de Google Play Store, menciona variables agregadas como instalaciones, reseñas, calificación y características técnicas (tamaño, categoría, precio); ofrece una visión macro del desempeño de la aplicación.

Dado que no se cuenta con datos individuales, se realiza una aproximación basada en la proporción de interacción (relación entre el número de descargas y la cantidad de reseñas) si una app acumula muchas instalaciones, pero recibe pocas reseñas, se interpreta como una señal de pérdida de interés o de tracción en el mercado.

2.2.3 Justificación de Métodos

La combinación del método cuantitativo y CRISP-DM se justifica por la necesidad de transformar datos de mercado en conocimiento estratégico. Esta combinación se justifica por los siguientes pilares:

- Eficiencia analítica: el enfoque cuantitativo permite procesar grandes volúmenes de información agregada para identificar patrones de éxito que no son visibles a simple vista.
- Validación de Indicadores Proxy: Ante la ausencia de datos longitudinales el método cuantitativo justifica la creación de una variable proxy que permite aproximar con rigor estadístico el concepto de "fidelidad de mercado".
- Optimización del Retorno de Inversión (ROI): el uso de ciencia de datos permite pasar de una gestión intuitiva a una basada en evidencia.

- Interpretabilidad: La selección de modelos dentro del flujo CRISP-DM permite a los equipos de negocio entender el "porqué" del abandono, cerrando la brecha entre el análisis técnico y la toma de decisiones.

2.3 Resultados y Análisis

A continuación, se muestran los resultados de analizar el datset de Google Play Store para identificar los factores que influyen en la retención.

2.3.1 Resultados

Los resultados permiten tener una visión clara sobre el abandono de clientes y evaluar los algoritmos de machine learning implementados para su predicción.

2.3.1.1 Descripción de la Base de Datos

Se analiza el dataset “Google Play Store” (Gupta, 2018), el cual contiene información sobre aplicaciones disponibles en la tienda oficial de Android. Originalmente la base contenía datos de más de 10,000 aplicaciones. Tras un proceso de depuración y procesamiento de valores nulos en variables como Rating, la muestra final está conformada por 8,196 registros únicos.

Para Turner (2025) la expansión masiva de aplicaciones móviles necesita un tratamiento riguroso de datos que garantice la validez de los resultados.

2.3.1.1.1 Clasificación de variables

Las variables se clasifican en:

- **Variable objetivo:** Dado que el dataset no cuenta con registros de actividad individual (logs de uso diario), se construyó la variable objetivo High_Retention[1] como una variable proxy que aproxima el concepto de retención diseñada para capturar el éxito de una aplicación; surge de la combinación de la satisfacción percibida (Rating > 4.2) con el compromiso activo (Engagement (Reviews / Installs) > mediana). La lógica es que una aplicación con un alto nivel de retención no solo debe mantener calificaciones superiores al promedio del mercado, reflejando valoración positiva, sino también incentivar una participación de los usuarios, evidenciada en una mayor proporción de reseñas respecto a sus instalaciones. Aunque esta medida es un proxy indirecto

[1] Teoría de la Señalización de Mercado los usuarios dependen de indicadores (reseñas, descargas) como señales de confianza (Davenport et al., 2010)

constituye la mejor aproximación disponible con datos públicos, permitiendo identificar qué características técnicas o de mercado se asocian con un mejor posicionamiento competitivo.

Esta variable presenta desequilibrio de clase aproximadamente 1 de cada 3 aplicaciones logra superar los estándares de alta retención definidos previamente, 5,322 apps (64.9% de la muestra) representa el churn, mientras que el 35.09% (2,877 apps) representan alta retención (ver ANEXO B).

- **Variables numéricas:**
  - Rating: indicador principal de satisfacción del usuario.
  - Reviews: Volumen total de opiniones.
  - Installs: Cantidad de instalaciones.
  - Price: Costo de la app en dólares.
  - Size_MB: Tamaño de la aplicación.
- **Variables categóricas:**
  - Category: Segmento de mercado.
  - Type: Modelo de monetización.
  - Content Rating: Público objetivo.
  - Genres: Sub-categorías específicas de la aplicación.

La Clasificación de variables cualitativas y cuantitativas se puede apreciar en ANEXO A)

2.3.1.2 Análisis descriptivo

Al observar la Tabla 2 de métricas podemos observar el comportamiento de estas variables señalando patrones sobre cómo los usuarios interactúan con las aplicaciones:

- **Reviews**: Reseñas promedio de 255,251 confirman que el volumen de interacción es un determinante crítico, funciona como un indicador de validación social genera confianza en el usuario y fomenta retención (Wu et al., 2022).
- **Installs**: La brecha entre el percentil 25 y el percentil 75 evidencia un mercado altamente concentrado, una minoría de aplicaciones concentra la mayor parte de la cuota de mercado.
- **Size_MB**: El peso promedio que tienen las aplicaciones (21.75 MB) refleja una tendencia hacia la optimización un tamaño pequeño sirve como estrategia de accesibilidad permite mitigar las limitaciones de almacenamiento en dispositivos de gama media y baja.

Se observa que las variables Reviews e Installs son componentes fundamentales en el cálculo del Engagement_Rate, base para la construcción de la variable objetivo (Retención). Por tanto, se identifican como variables con riesgo de dependencia estructural, lo que motiva un análisis diferenciado en la etapa de evaluación.

Tabla 2 *Estadísticas*

Métrica	Reviews	Size_MB	Installs
Media (mean)	255.251,47	21,75	9.165.089,72
Desv. Estándar (std)	1.985.593,85	21,04	58.250.865,45
Mínimo (min)	1	0,01	1
25% (Q1)	126	5,8	10.000,00
50% (Mediana)	3.004,00	18	100.000,00
75% (Q3)	43.813,00	27	1.000.000,00
Máximo (max)	78.158.306,00	100	1.000.000.000,00

2.3.1.3 Análisis de los factores de retención

En la Tabla 3 podemos apreciar el análisis comparativo entre aplicaciones con alta y baja retención. Se observa que las aplicaciones con alta retención alcanzan un rating promedio de 4.56, frente al 3.98 de aquellas con baja retención. Esto sugiere que la satisfacción del usuario, reflejada en el Rating, es un indicador clave de la permanencia del usuario en la plataforma.

Respecto a las variables de volumen, las aplicaciones con alta retención registran promedios significativamente superiores en Reviews (544,186) e Installs (10.7M) en comparación con el grupo de baja retención. Es importante notar que, dado que la métrica de retención se construye a partir del ratio de estas variables, esta diferencia es esperada y describe la estructura de los datos analizados. Asimismo, se observa que el tamaño de la aplicación (Size_MB) es ligeramente superior en las apps de alta retención (24.71 MB vs 20.27 MB), lo que podría estar asociado a una mayor complejidad o cantidad de funciones en apps más exitosas.

La mediana de las Reviews muestra una diferencia marcada entre los grupos: en aplicaciones de alta retención alcanza 19,800, mientras que en las de baja retención apenas llega a 1,529. Esta brecha tan amplia evidencia un contraste abismal en la distribución de reseñas, lo que refuerza la hipótesis de que el modelo podría ‘aprender’ una separación demasiado sencilla si se incluyen estas variables como predictores. En otras palabras, el riesgo es que el modelo se

apoye en un patrón matemático directo derivado del volumen de reseñas, en lugar de captar relaciones más profundas entre los atributos intrínsecos de las aplicaciones.

Tabla 3 *Análisis de los factores de retención*

High_Retention	Variables	mean	median	std
CHURN	**Installs**	8.347.994,97	100.000,00	61.608.708,42
	Rating	3,98	4,10	0,55
	Reviews	110.148,53	1.529,50	1.419.346,69
	Size_MB	20,27	16,00	19,90
RETENCION	**Installs**	10.792.122,19	500.000,00	50.880.884,49
	Rating	4,56	4,50	0,20
	Reviews	544.186,37	19.800,00	2.767.391,11
	Size_MB	24,71	21,75	22,87

2.3.1.4 Evaluación de los Modelos de Machine Learning

Para medir el impacto que tienen las características técnicas y de mercado de las aplicaciones móviles en la retención se emplearon modelos de Machine Learning , en primer lugar, se aplicó una partición de datos 80% para entrenamiento y 20% para prueba. Esto permite identificar patrones de deserción y comprobar la capacidad de los modelos para generalizar más allá de los datos utilizados en su ajuste inicial. Con la finalidad de garantizar que el desempeño no dependa de una división sesgada se implementó un muestreo estratificado en la partición train-test, preservando la proporción original de la variable objetivo en ambos subconjuntos, paso importante cuando trabajamos con clases desbalanceadas (en nuestro caso 64.9% representa el churn, 35.09% representan alta retención). De esta manera se garantiza que la clase minoritaria no quede subrepresentada en el conjunto de datos de prueba, evitando sesgos que podrían generar resultados distorsionados y poco fiables. Así, se confirma que los hallazgos de esta investigación reflejan tendencias estructurales presentes en el dataset de Google Play Store y no son simples ruidos estadísticos derivados de la partición de la muestra.

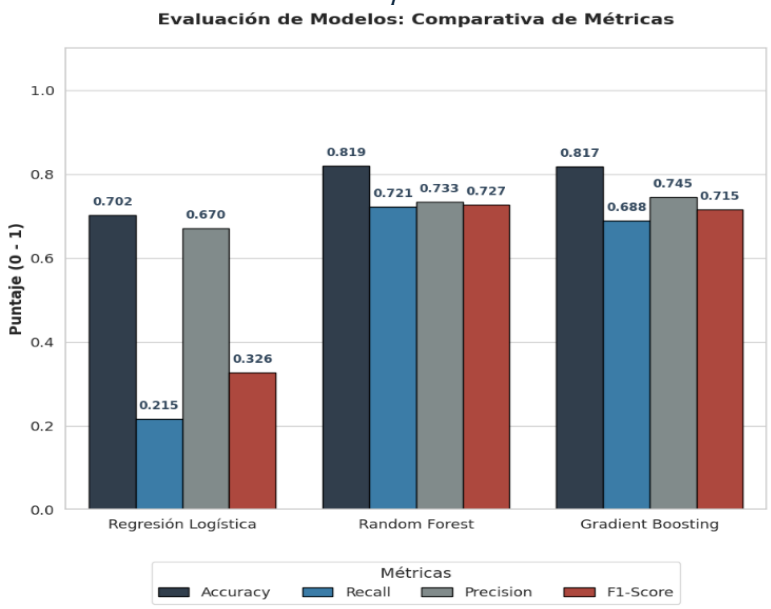
A continuación, se evaluaron tres algoritmos de aprendizaje supervisado cuyos resultados se presentan en la Tabla 4:

- Regresión Logística.
- Random Forest.
- Gradient Boosting,

Tabla 4 *Evaluación de modelos*

	Logistic Regression	Random Forest	Gradient Boosting
Accuracy	0,70	0,82	0,82
Recall	0,22	0,72	0,69
Precision	0,67	0,73	0,75
F1-Score	0,33	0,73	0,72
AUC Test	0,71	0,90	0,91

Ilustración 1 *Comparación métricas*



La deserción de clientes representa uno de los desafíos más críticos para las empresas. El costo de adquirir un nuevo cliente es superior al costo de mantener uno existente; la capacidad de predecir qué usuarios abandonarán el servicio es una necesidad fundamental para las empresas.

- Gradient Boosting Classifier (GBC): Este modelo construye árboles de decisión de forma secuencial, cada nuevo árbol corrige los errores residuales de los anteriores. En este estudio, alcanzó un AUC de 0.9129, indica una probabilidad del 91.2% de que el modelo clasifique correctamente a una aplicación con alta retención frente a una de baja. Aunque su Recall es del 68.8%, su equilibrio con una precisión del 74.5% lo convierte en el modelo robusto para identificar patrones complejos. La superioridad del Gradient Boosting Classifier en este análisis se explica por su capacidad para capturar relaciones no lineales y dependencias entre variables, logrando identificar patrones sutiles de deserción que otros algoritmos no pudieron detectar.
- Logistic Regression: La Regresión Logística obtuvo un Accuracy del 70.2%, sin embargo, su principal debilidad se observa en el Recall (21.5%), lo que indica una

limitada capacidad para identificar aplicaciones con alta retención. A pesar de su menor desempeño predictivo frente a los modelos de ensamble, este modelo proporciona un marco interpretable para analizar la relación entre las variables independientes y la probabilidad de retención. En este contexto se evaluó la presencia de multicolinealidad mediante el análisis del Variance Inflation Factor (VIF), donde todas las variables presentaron valores inferiores a 1.7. Estos resultados sugieren una baja correlación lineal entre los predictores. No obstante, es importante señalar que el VIF únicamente permite detectar multicolinealidad entre variables independientes y no permite evaluar posibles dependencias estructurales entre los predictores y la variable objetivo.

- Random Forest: Utilizando la técnica de Bagging este modelo genera múltiples árboles de decisión independientes y promedia sus predicciones presentó el rendimiento más alto en términos de Accuracy (81.8%) y mejor Recall (72.0%) de cada 100 aplicaciones que logran una alta retención, el modelo identifica correctamente a 72. Su desempeño confirma que variables como el volumen de "Reviews" (52.7% de importancia) y el número de "Installs" son predictores determinantes. Sin embargo, esta elevada importancia debe interpretarse con cautela, ya que ambas variables participan directamente en la construcción de la variable objetivo definida como el ratio entre reseñas e instalaciones. En consecuencia, parte del poder predictivo observado podría estar asociado a esta relación estructural entre las variables predictoras y el target.

Al comparar los resultados (métricas AUC y Recall) podemos recalcar la superioridad que tiene el modelo Gradient Boosting evidenciando la capacidad del algoritmo para capturar relaciones no lineales y patrones complejos que modelos tradicionales, como la Regresión Logística, no logran representar con la misma precisión.

Aunque los modelos Random Forest y Gradient Boosting presentan accuracy superior al 80%, no es suficiente para evaluar el desempeño en la clase minoritaria de Alta Retención. Para complementar los resultados es útil analizar métricas como recall, precisión y AUC, para entender cómo cada modelo maneja el desequilibrio y qué tan bien logra discernir entre aplicaciones con alto y bajo potencial de retencion.

Existe un trade-off entre precisión y recall. Random Forest tiene un recall de aproximadamente 72 %, identifica una proporción más alta de las aplicaciones que verdaderamente tienen éxito sin omitir tantos casos positivos; Gradient Boosting logra una precisión más alta de 75 %, reduciendo los falsos positivos logrando predicciones positivas sean más fiables además sugiere

una mayor estabilidad ante variaciones del dataset. De la misma forma tiene un AUC alto (0.91) combinadas estas métricas sugieren que el modelo posee una buena capacidad para separar clases a distintos umbrales de decisión, especialmente en contextos donde las clases se encuentran desbalanceadas como en el presente estudio.

Los modelos de ensamble (GBC y RF) favorece la detección de casos relevantes aun aceptando cierto margen de error. Además, al evaluar la estabilidad de los modelos en distintas particiones del dataset, el F1-score se mantiene robusto frente a cambios en la distribución de los datos. Por lo tanto, los modelos no memorizan el ruido, sino que ha aprendido patrones que se pueden generalizar.

Con el objetivo de evaluar la posible influencia de la relación estructural entre algunas variables predictoras y la variable objetivo, se realizó adicionalmente una prueba de robustez. Esta consistió en repetir el proceso de entrenamiento y evaluación de los modelos excluyendo las variables Reviews e Installs, debido a que ambas intervienen directamente en la construcción del indicador de retención. Este procedimiento permite verificar si el desempeño de los modelos se mantiene al utilizar únicamente variables independientes que no participan en la definición del target.

2.3.1.5 Prueba de robustez del modelo

Como medida de validación técnica, se ejecutó un escenario de robustez excluyendo las variables Reviews e Installs del conjunto de predictores. Este procedimiento es esencial para verificar la capacidad de generalización del modelo utilizando únicamente variables que no participaron en la construcción de la variable objetivo, como la categoría, el tamaño y el precio. Los resultados de este escenario permiten determinar si el modelo mantiene una capacidad predictiva real basada en las características del producto o si su desempeño inicial estaba condicionado principalmente por la relación matemática con el volumen de interacción.

A partir de los resultados presentados en la Tabla 5, se observa un cambio significativo en la jerarquía de desempeño de los algoritmos tras la exclusión de las variables estructurales (Reviews e Installs). Este experimento de control permite extraer las siguientes deducciones:

- Impacto de las Variables Estructurales: La disminución del AUC (de niveles de 0.91 a un rango entre 0.58 y 0.61) confirma que el volumen de instalaciones y reseñas constituía el núcleo de la capacidad predictiva original. Esto valida la sospecha inicial sobre la relación intrínseca entre estos predictores y la variable objetivo, evidenciando

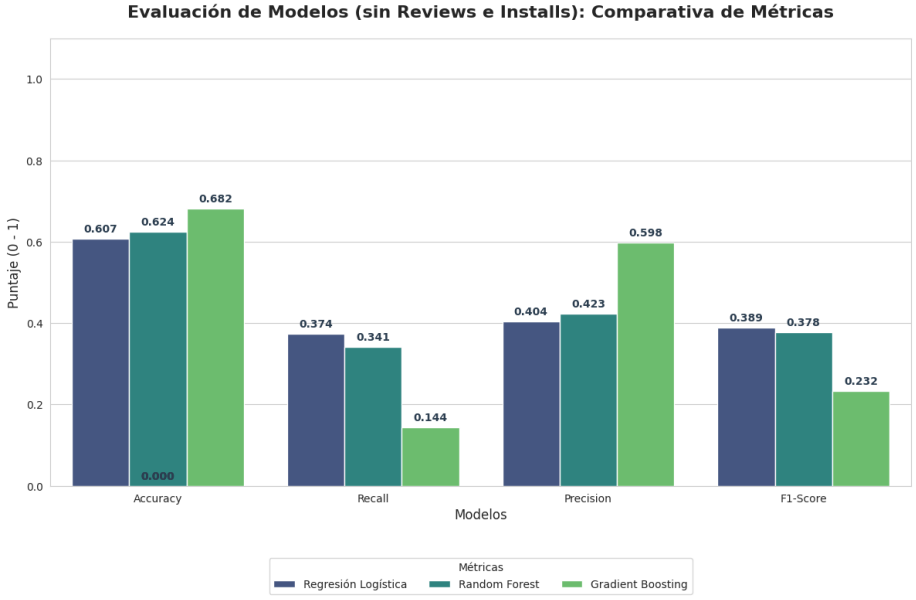
que gran parte del éxito de los modelos de ensamble en la primera fase se debía a la correlación directa entre el uso de la aplicación y su tasa de retención.

- Cambio en la Capacidad de Generalización: Ante la ausencia de los predictores dominantes, la Regresión Logística muestra una resiliencia notable, alcanzando el F1-Score más alto (0.39). A diferencia de los modelos de ensamble como Random Forest, que sufrieron una caída en su capacidad de identificar la clase positiva (cayendo a un Recall de 0.34), el modelo logístico logró un equilibrio superior entre precisión (0.40) y sensibilidad (0.37). Esto sugiere que, en entornos con menor densidad informativa, los modelos lineales simples pueden ser menos propensos a perder señales débiles en comparación con los modelos de ensamble.
- Desempeño del Gradient Boosting (GBC): Aunque el GBC conserva el AUC más alto (0.61), su Recall cayó drásticamente a 0.14. Esto indica que, si bien el modelo aún posee una capacidad de discriminación superior al azar para separar las clases en términos generales, tiene dificultades extremas para detectar correctamente las aplicaciones de "Alta Retención" basándose únicamente en atributos secundarios (como categoría, tamaño, precio o clasificación de contenido).

Tabla 5 *Evaluación de modelos (sin Reviews e Installs)*

	Logistic Regression	**Random Forest**	**Gradient Boosting**
Accuracy	0.61	0.62	0.68
Recall	0.37	0.34	0.14
Precision	0.40	0.42	0.60
F1-Score	0.39	0.38	0.23
AUC Test	0.59	0.58	0.61

Ilustración 2 *Comparación métricas (sin Reviews e Installs)*



La comparación entre la evaluación inicial de los modelos y la prueba de robustez revela un aspecto crítico en la predicción de la retención. En la evaluación original, los modelos de ensamble (Gradient Boosting y Random Forest) mostraron un desempeño sobresaliente, con valores de AUC≈0.91, lo que sugiere una alta capacidad para distinguir entre aplicaciones con alta y baja retención. No obstante, la prueba de robustez evidenció que una parte significativa de este rendimiento estaba asociada a la dependencia estructural de las variables Reviews e Installs, ya que ambas participan directamente en la construcción de la variable objetivo.

Al excluir estos predictores del conjunto de entrenamiento, el desempeño global de los modelos se redujo y convergió hacia valores más moderados (AUC≈0.61). Este resultado sugiere que, si bien las características intrínsecas de la aplicación contienen información relevante para explicar la retención, el comportamiento histórico de adopción reflejado en métricas de interacción del usuario constituye un factor determinante en la capacidad predictiva del modelo.

En conjunto, los resultados confirman que Gradient Boosting sigue siendo la arquitectura más adecuada para capturar patrones complejos dentro del dataset. Su capacidad para modelar relaciones no lineales y dependencias entre variables lo convierte en una alternativa particularmente eficaz cuando se dispone de un ecosistema de datos más completo que combine métricas de desempeño en la tienda de aplicaciones con atributos técnicos y de mercado del producto.

2.3.1.6 Evaluación de las curva ROC

La evaluación de los modelos de Machine Learning revela una dicotomía fundamental entre la simplicidad estadística y la potencia de los modelos de ensamble. En el análisis inicial, el Gradient Boosting Classifier (GBC) y el Random Forest (RF) se posicionaron como herramientas de alta fidelidad, logrando un AUC superior a 0.90. Esta proximidad al valor ideal de 1 indica que los modelos de ensamble no clasifican al azar, sino que capturan patrones complejos de comportamiento. En contraste, la Regresión Logística mostró una curvatura más modesta (AUC de 0.71), confirmando su debilidad estructural ante la naturaleza no lineal de los datos de la Google Play Store.

Sin embargo, al someter a los modelos a una prueba de robustez (excluyendo las variables Reviews e Installs), la morfología de las curvas ROC experimenta una transformación significativa, como se observa en la Ilustración 3.

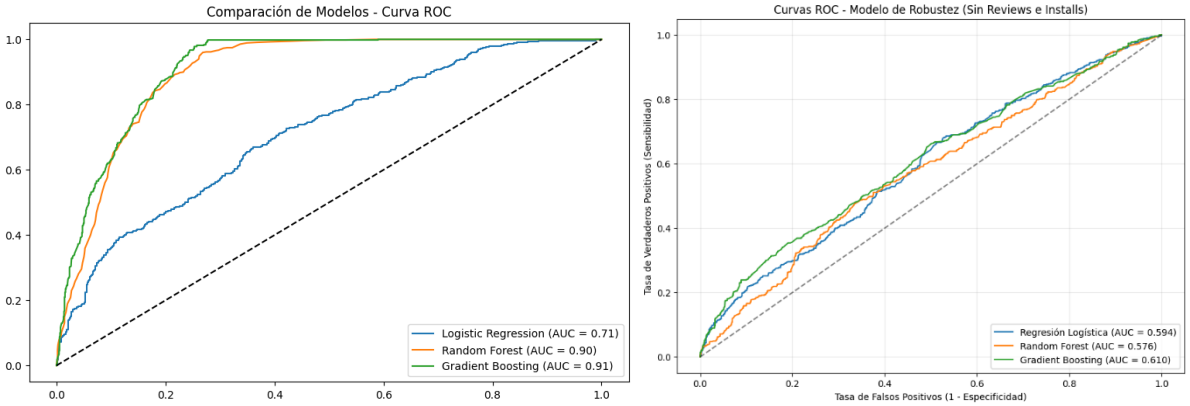
Al eliminar del conjunto de predictores las variables que participan directamente en la construcción de la variable objetivo, las curvas ROC de los tres modelos muestran un desplazamiento hacia la línea de identidad, el desempeño de los modelos se reduce de forma notable: el AUC del modelo Gradient Boosting disminuye de 0.91 a 0.610, mientras que la Regresión Logística alcanza un AUC de 0.594 y el Random Forest un valor de 0.576. Este resultado evidencia que una parte significativa de la capacidad predictiva observada en la evaluación inicial estaba asociada a la relación estructural entre las variables Reviews, Installs y la variable objetivo.

La comparación entre modelos muestra que, al eliminar las variables de volumen la ventaja del Gradient Boosting frente al modelo lineal y otros algoritmos se reduce de forma significativa. Esto indica que los atributos intrínsecos de las aplicaciones (categoría, precio, tamaño) aportan una señal predictiva real pero limitada, reflejada en valores de AUC cercanos a 0.6, lo que corresponde a una capacidad de discriminación modesta, apenas superior al azar.

La comparación entre escenarios confirma que la retención de una aplicación está fuertemente asociada a indicadores de validación social y alcance —como reseñas e instalaciones—, reflejo del comportamiento histórico de adopción. Sin embargo, también se identifica un componente informativo en atributos técnicos y de mercado (categoría, precio, tamaño), capturado con mayor eficacia por modelos no lineales como Gradient Boosting. El análisis de robustez demuestra que el modelo no depende exclusivamente de la relación matemática con las variables de volumen, aún bajo condiciones más restrictivas, mantiene una capacidad de

discriminación positiva pero moderada (AUC ≈ 0.6), lo que refuerza la interpretación de Gradient Boosting como el algoritmo más adecuado para representar patrones complejos en aplicaciones móviles.

Ilustración 3 *Curvas ROC*



2.3.1.7 Análisis matriz de confusión

El análisis de las matrices de confusión en las dos etapas del estudio (Modelo Original y Prueba de Robustez) permite evidenciar cambios sustanciales en el comportamiento predictivo de los algoritmos cuando se eliminan variables altamente informativas del modelo. La matriz de confusión constituye una herramienta fundamental para la evaluación de modelos de clasificación, ya que compara las predicciones realizadas por el algoritmo con los valores reales del conjunto de datos y permite identificar distintos tipos de resultados: verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos.

En la etapa correspondiente al Modelo Original, los algoritmos de ensamble, Gradient Boosting y Random Forest, mostraron un desempeño superior en la identificación de aplicaciones con Alta Retención. La inclusión de Reviews e Installs permitió capturar con mayor precisión los patrones asociados al éxito de las aplicaciones dentro de la tienda digital. En particular, el modelo de Gradient Boosting logró reducir significativamente la presencia de falsos negativos, alcanzando niveles de precisión cercanos al 75% en la detección de aplicaciones con alto potencial de fidelización.

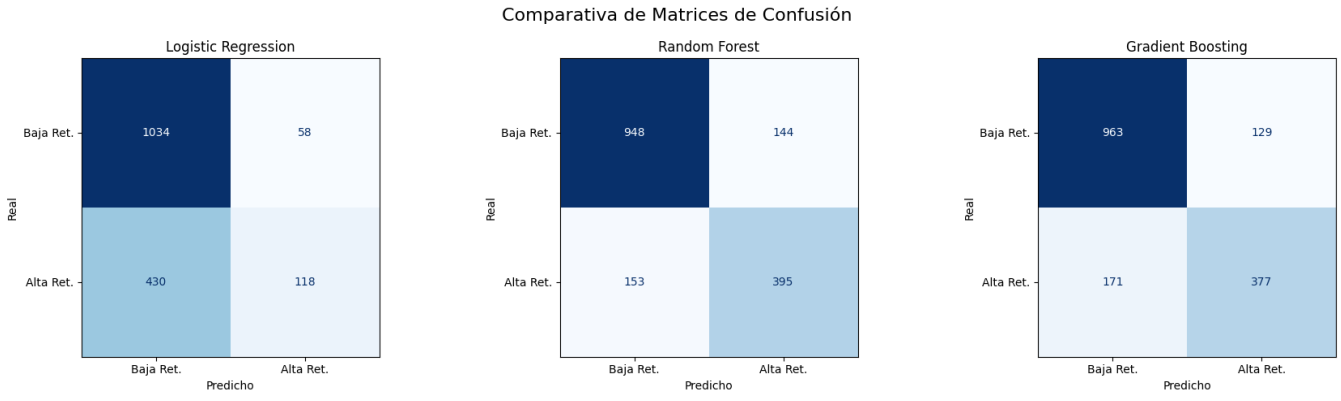
Sin embargo, al realizar la Prueba de Robustez, en la cual se excluyeron las variables estructurales que podían introducir sesgos predictivos, se observa una modificación significativa en el comportamiento de los modelos. Las matrices de confusión muestran que el modelo de Gradient Boosting, previamente el de mejor desempeño, adopta un comportamiento marcadamente conservador. En esta nueva configuración, el modelo logra identificar

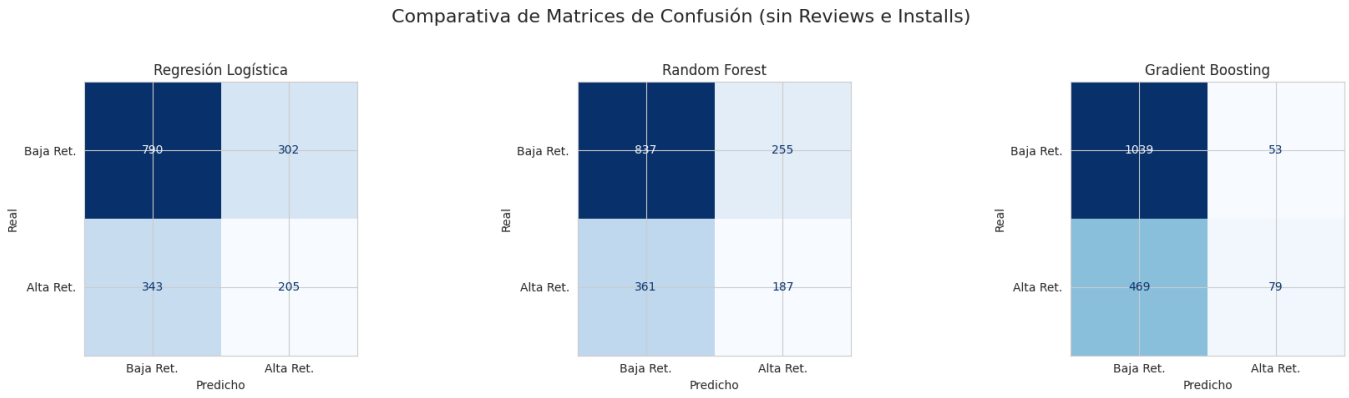
únicamente 79 casos de Alta Retención, mientras que clasifica erróneamente 469 aplicaciones exitosas como de Baja Retención, lo que implica un incremento considerable en el número de falsos negativos. Este comportamiento sugiere que, ante la ausencia de señales predictivas fuertes, el algoritmo tiende a privilegiar la clase mayoritaria, reduciendo el número de falsos positivos y aumentando su especificidad, pero sacrificando capacidad de detección de la clase minoritaria.

Por otro lado, la Regresión Logística presenta un comportamiento diferente frente a la eliminación de estas variables. Demuestra una mayor capacidad para identificar casos de Alta Retención en el escenario de robustez, logrando capturar 205 aplicaciones exitosas y superando al modelo de Gradient Boosting en términos de sensibilidad. No obstante, esta mayor predisposición a clasificar observaciones como positivas implica un aumento en los falsos positivos, alcanzando 302 casos, lo que evidencia un trade-off entre capacidad de detección y precisión en las predicciones.

En el caso del Random Forest, el modelo se sitúa en una posición intermedia entre ambos enfoques. Con 187 aciertos en la clase positiva, su arquitectura basada en múltiples árboles de decisión entrenados mediante técnicas de bagging permite mantener un equilibrio relativo entre sensibilidad y control de errores. Esto se traduce en un desempeño más estable que el de Gradient Boosting en el escenario de robustez, pero con un comportamiento más conservador que el observado en la Regresión Logística..

Ilustración 4 *Comparativa de Matrices de Confusión*





2.3.1.8 Análisis Comparativo: Curvas Precision-Recall

A diferencia de la curva ROC, que muestra la tasa de verdaderos positivos y la tasa de falsos positivos, la curva Precision-Recall muestran cómo se comporta cada modelo al clasificar correctamente al grupo objetivo (positivos) frente a los falsos positivos, la precisión indica qué porcentaje de predicciones positivas son verdaderas y el recall mide qué proporción de todos los positivos reales fueron capturados por el modelo. Una curva más alta y extendida hacia la derecha en el gráfico nos dice que el modelo mantiene buena precisión incluso al aumentar el número de positivos identificados, esto es importante dado que nuestros datos se encuentran desbalanceados.

El área bajo la curva Precision-Recall muestra cómo se equilibra la precisión frente al recall refleja directamente la capacidad del modelo para priorizar correctamente los positivos frente a los falsos positivos, esta métrica toma relevancia para datos con pocos casos positivos.

En el escenario donde se omiten las variables de volumen (Installs y Reviews), la curva PR (ver Ilustración 5) revela la verdadera capacidad de los algoritmos para extraer valor de los metadatos técnicos.

Gradient Boosting Classifier (GBC): Se consolida como el modelo más robusto con un Average Precision (AP) de 0.457. Su curva se mantiene consistentemente por encima de los demás modelos, lo que confirma que logra detectar casos de retención real con una tasa significativamente menor de falsos positivos.

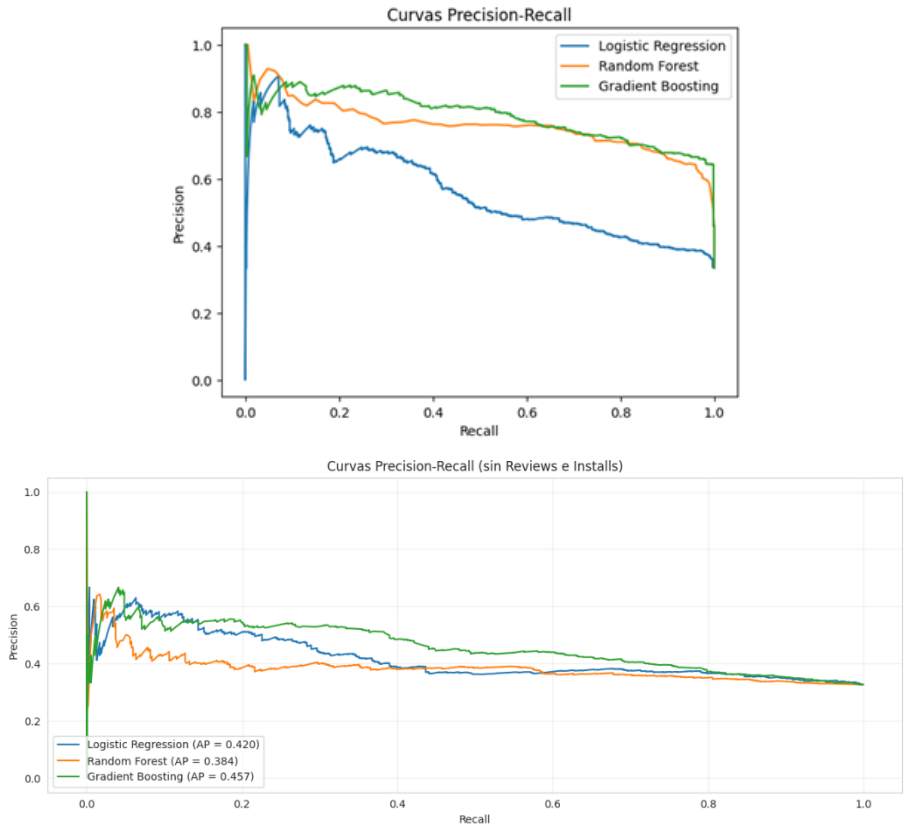
Regresión Logística: Presenta una resiliencia notable con un AP de 0.420. Aunque es un modelo lineal, supera al Random Forest en este contexto de señal débil, lo que sugiere que, ante la falta de variables dominantes, una estructura de coeficientes ponderados es más estable para evitar el ruido.

Random Forest: Muestra el desempeño más limitado (AP de 0.384). Su curva cae de forma prematura, indicando que, para capturar más casos positivos, el modelo incurre en un error desproporcionado de falsos positivos.

Al comparar estos resultados con el modelo inicial, la diferencia es drástica. En el modelo original, el área bajo la curva PR es considerablemente mayor, impulsada por la alta correlación entre la validación social y la permanencia del usuario. Sin embargo, la Prueba de Robustez actúa como un validador de rigor científico: demuestra que el Gradient Boosting no solo depende de variables obvias, sino que posee la arquitectura jerárquica más eficiente para identificar aplicaciones con potencial de éxito basándose únicamente en atributos de diseño y mercado.

Las curvas Precision-Recall (PR) son un respaldo visual porque muestran cómo el modelo equilibra la Precisión y el Recall en distintos umbrales de decisión; la curva PR muestra la capacidad real del modelo para priorizar los casos relevantes y ajustar su comportamiento según las necesidades del negocio, ofreciendo una comparación más rigurosa y útil entre modelos en contextos de desbalance de clases.

Ilustración 5 *Curva Precision-Recall*



2.3.1.9 Identificación de multicolinealidad

Con el fin de asegurar la estabilidad de los modelos y evaluar la relación entre los predictores, se aplicó el Factor de Inflación de la Varianza (VIF). Este análisis es fundamental para detectar la multicolinealidad, es decir, si una variable independiente puede ser explicada linealmente por otras, lo que podría distorsionar la estimación de los coeficientes. Generalmente, valores de VIF superiores a 5 indican una multicolinealidad significativa que requiere atención.

Los resultados obtenidos en la Tabla 6 muestran que todas las variables presentan valores inferiores a 1.7, lo que garantiza la ausencia de multicolinealidad lineal. Las variables Reviews (1.67) e Installs (1.67), a pesar de estar lógicamente vinculadas en el ecosistema de aplicaciones, aportan información estadísticamente diferenciada desde una perspectiva de regresión lineal.

Size_MB (1.03) y Price (1.00) indican correlación nula con las demás variables. El precio y el peso de la aplicación actúan como factores independientes en el comportamiento del usuario.

Es importante precisar que, si bien un VIF bajo descarta la multicolinealidad, no es una herramienta suficiente para descartar el data leakage en este contexto. Dado que la variable objetivo (Engagement_Rate) se construye mediante el cociente Reviews/Installs, existe una dependencia estructural no lineal entre el target y estos predictores. El VIF, al ser una métrica de asociación lineal, no detecta esta circularidad matemática.

Debido a que la importancia relativa de Reviews e Installs supera el 90% en los modelos iniciales (ver Ilustración 6), se reconoce que el modelo podría estar aproximando la fórmula del target en lugar de identificar patrones de comportamiento independientes.

Tabla 6 *VIF*

Feature	VIF
Reviews	1,67
Size_MB	1,03
Installs	1,67
Price	1,00

2.3.2 Análisis

En esta sección se discuten los hallazgos en función de las preguntas de investigación y el marco teórico establecido, contrastando los datos empíricos con la literatura académica vigente.

2.3.2.1 Determinantes de retención

2.3.2.1.1 Análisis general

La Tabla 7 presenta un análisis comparativo de la relevancia de los predictores en dos escenarios: el Modelo Original y el Modelo sin Fuga (prueba de robustez). Esta distinción es fundamental para identificar si el modelo aprende patrones reales o si su capacidad predictiva depende de una relación estructural con la variable objetivo.

En el escenario original, estos modelos de Ensamble presentan una concentración extrema en las métricas de interacción:

En Gradient Boosting, el 92.47% de la importancia recae en Reviews e Installs. En Random Forest, esta cifra es del 64.8%. Como se analizó previamente, esto confirma que los árboles de decisión detectan con facilidad la operación no lineal que define al target.

Al excluir las variables sospechosas de generar data leakage, se observa un cambio drástico. El Size_MB emerge como el predictor principal (42.17% en GB y 65.58% en RF), seguido por el Price y la Category. Esto sugiere que, una vez eliminada la señal del comportamiento previo, el tamaño de la aplicación es el factor técnico que más influye en la decisión de retención del usuario.

El modelo lineal muestra un comportamiento atípico en comparación con los ensambles; en el modelo original, domina la variable Category (80.87%). Es notable que las variables Reviews e Installs tengan un peso casi nulo ($\leq 0.01\%$), lo que ratifica la incapacidad de la regresión logística para procesar la relación matemática no lineal que vincula a estos predictores con el éxito de retención.

Al eliminar los predictores de interacción, la importancia se redistribuye hacia el Type (46.09%) y Size_MB (27.55%), alineándose más con los factores estructurales de la aplicación.

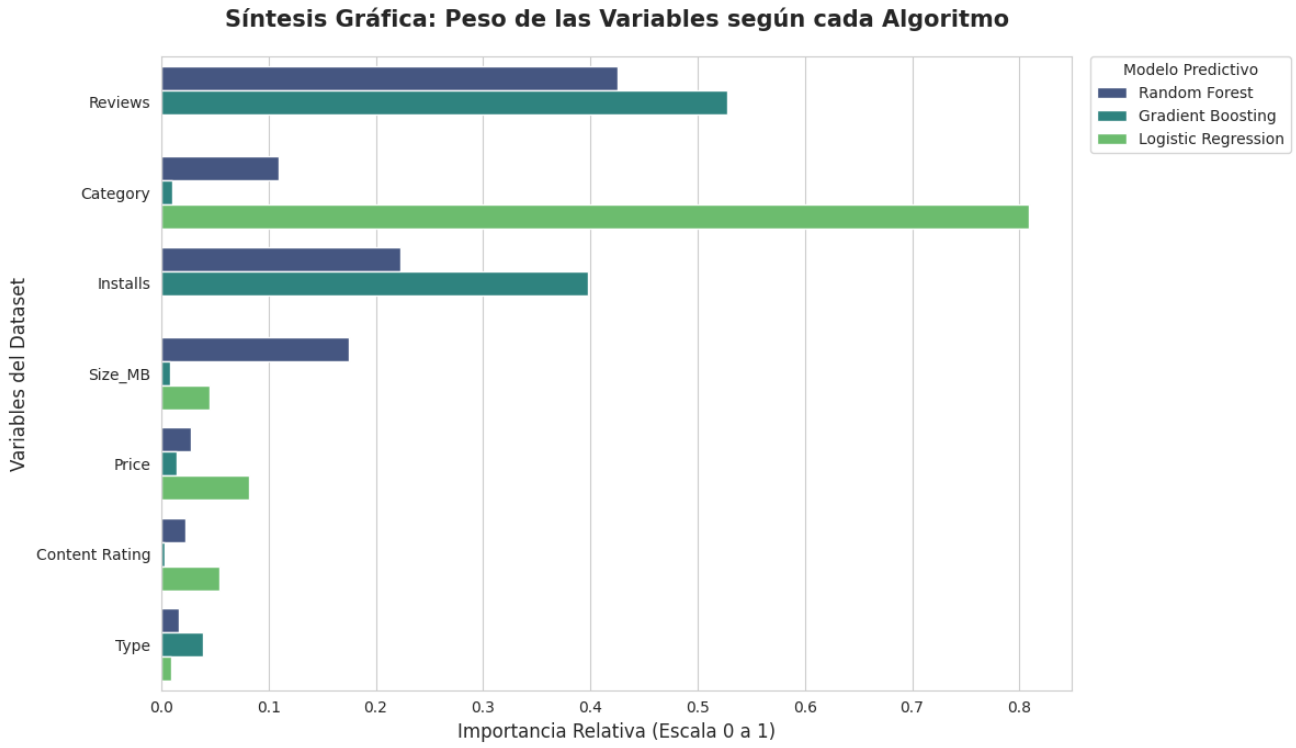
La drástica caída en la importancia de variables al pasar al "Modelo sin fuga" confirma que el alto desempeño inicial estaba impulsado por la circularidad entre el target y sus componentes (Reviews/Installs).

Una vez aislada la dependencia estructural, el Tamaño (Size_MB) y el Precio se consolidan como los determinantes genuinos de la retención, validando la hipótesis de que aplicaciones muy pesadas o con costos elevados enfrentan mayores barreras de permanencia.

Tabla 7 *Comparación determinantes de retención*

Variable	Gradient Boosting		Random Forest		Logistic Regression	
	Modelo Original	**Modelo sin fuga**	**Modelo Original**	**Modelo sin fuga**	**Modelo Original**	**Modelo sin fuga**
Reviews	52.73%		42.54%		0.01%	
Installs	39.74%		22.26%		0.0%	
Type	3.94%	3.89%	1.68%	1.95%	0.97%	46.09%
Price	1.46%	30.18%	2.78%	4.89%	8.17%	9.21%
Category	1.02%	16.48%	11.0%	24.56%	80.87%	1.19%
Size_MB	0.84%	42.17%	17.48%	65.58%	4.51%	27.55%
Content Rating	0.28%	7.28%	2.25%	3.02%	5.47%	15.95%

Ilustración 6 *Pesos de las variables según cada modelo (original)*



2.3.2.1.2 Análisis Gradient Boosting,

El análisis de importancia de variables para el algoritmo Gradient Boosting (ver Anexo C) permite identificar un cambio de paradigma en la capacidad predictiva del modelo cuando se controla la dependencia estructural.

En el modelo inicial, las variables Reviews (52.73%) e Installs (39.74%) concentran casi la totalidad del peso predictivo. Bajo este enfoque, la retención se interpreta como un fenómeno de tracción social: el éxito pasado (instalaciones) y la inversión de tiempo del usuario (reseñas) son los mejores predictores del éxito futuro. Desde la Teoría de la Inversión-Satisfacción, un alto número de reseñas indica que el usuario ha dedicado esfuerzo, lo que eleva su barrera de salida y lo posiciona en el grupo de "Alta Retención".

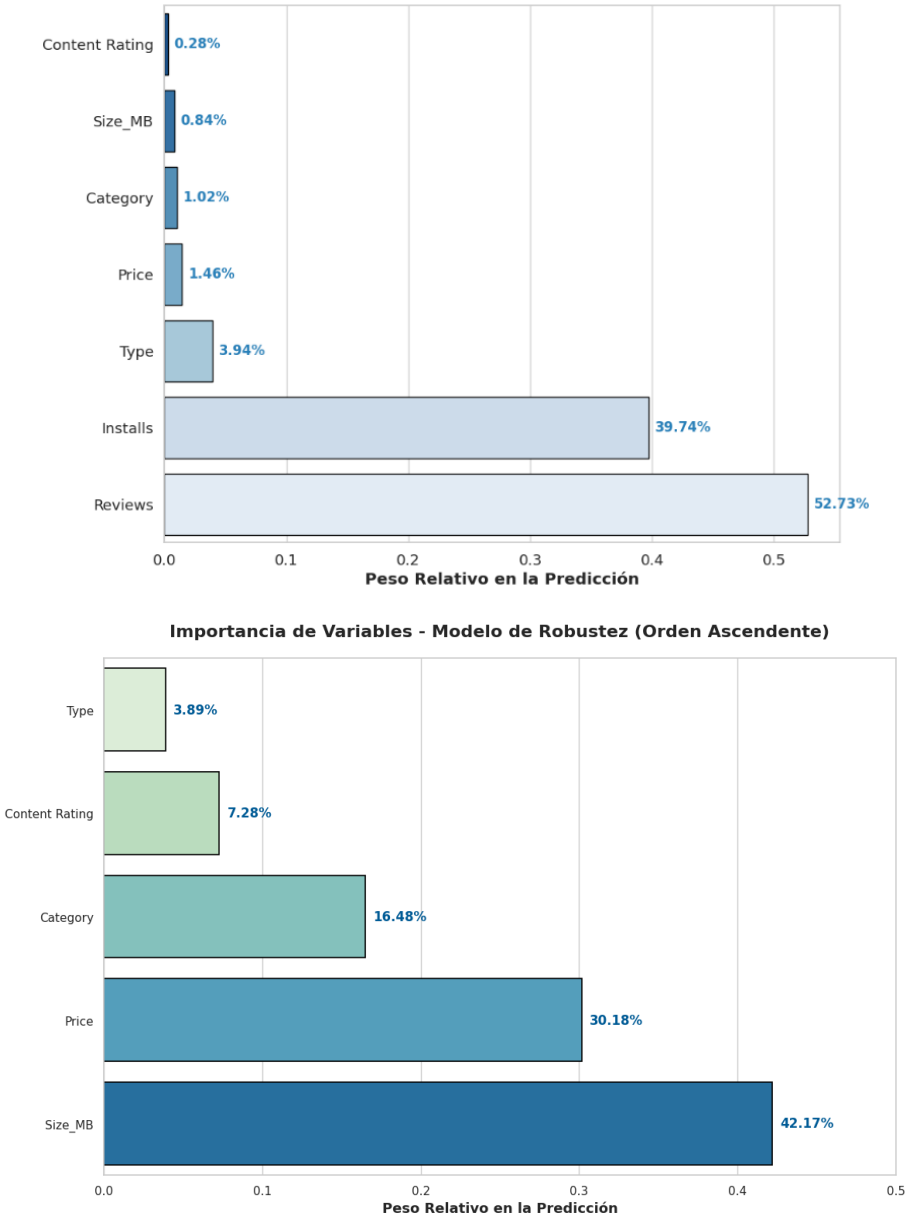
Al excluir Reviews e Installs para mitigar el riesgo de data leakage (dado que estas componen el ratio de la variable objetivo), la jerarquía de importancia se reconfigura drásticamente, revelando los determinantes intrínsecos de la aplicación:

Size_MB (42.17%): Se convierte en el predictor dominante, el peso del archivo es la principal barrera técnica para la retención. Aplicaciones excesivamente pesadas pueden ser desinstaladas con mayor frecuencia en favor de la optimización del almacenamiento del dispositivo.

Price (30.18%): Adquiere una relevancia significativa que no era visible en el modelo original. Esto indica que el costo es un factor determinante cuando no existe una base de usuarios consolidada que valide la calidad de la app.

Category (16.48%): El contexto o propósito de la aplicación influye en la retención, sugiriendo que ciertos nichos de mercado poseen una lealtad de usuario inherente más alta.

***Ilustración** 7 Importancia de las variables predictoras Gradient Boosting,*



2.3.2.2 Sensibilidad precio - lealtad

El análisis de la variable Precio ofrece una perspectiva sobre la psicología del consumidor en la Google Play Store. En el modelo predictivo inicial, el precio mostró una influencia mínima (1.46%), lo que sugería que la retención estaba ligada exclusivamente a la utilidad percibida. Sin embargo, al realizar la prueba de robustez (modelo sin fuga), la importancia del precio ascendió al 30.18%, consolidándose como el segundo factor más influyente.

Si bien las reseñas puede eclipsar el factor económico, el costo constituye una barrera crítica de permanencia cuando la aplicación no tiene una validación social masiva. Este hallazgo se alinea parcialmente con lo expuesto por Boozary et al. (2025), quienes manifiestan que, aunque la

experiencia del usuario y el valor agregado son ejes de fidelización, el costo económico actúa como un filtro de selección inicial.

En aplicaciones con muchas reseñas, los usuarios son menos sensibles al precio, priorizando la satisfacción y el ecosistema de la app. En ausencia de métricas de popularidad, el Precio (30.18%) se vuelve un determinante clave; los usuarios son más propensos a abandonar o no instalar aplicaciones con costos directos si no perciben un beneficio inmediato superior al de opciones gratuitas.

Por lo tanto, la lealtad en el entorno digital no es inmune al precio, sino que el precio es evaluado en función del riesgo percibido y el soporte de la comunidad de usuarios.

2.3.2.3 Modelos de Boosting

El modelo Gradient Boosting demostró ser el algoritmo con mayor capacidad de discriminación dentro del conjunto de modelos evaluados; aunque el AUC de 0.91 en el modelo inicial sugiere una alta capacidad discriminatoria, la caída a 0.61 en la prueba de robustez indica que la mayor parte de su utilidad práctica reside en la validación social previa, y no únicamente en los atributos intrínsecos de la app.

Es importante señalar que este alto rendimiento está estrechamente asociado con la capacidad del algoritmo para modelar relaciones complejas y no lineales entre variables, particularmente entre indicadores de popularidad como Reviews e Installs. Los modelos de Gradient Boosting se caracterizan por construir un conjunto de modelos débiles —generalmente árboles de decisión— que se entrenan de forma secuencial, donde cada nuevo árbol busca corregir los errores del anterior con el fin de minimizar la función de pérdida y mejorar progresivamente la precisión del modelo. Esto permite capturar patrones complejos en los datos que modelos más simples difícilmente podrían identificar.

No obstante, al aplicar la prueba de robustez mediante la eliminación de los predictores potencialmente asociados a fuga de información (Reviews e Installs), el modelo mantuvo una capacidad predictiva aceptable. En consecuencia, el modelo demuestra cierta capacidad de generalización, lo que refuerza la validez de los resultados obtenidos.

La eficacia del Gradient Boosting también se explica por su arquitectura de aprendizaje secuencial, en la cual cada nuevo árbol se entrena para corregir los errores residuales del modelo previo, optimizando progresivamente las predicciones del conjunto. Bajo este enfoque, la evaluación del modelo prioriza la métrica Recall, ya que uno de los objetivos específicos del

estudio consiste en minimizar los falsos negativos, es decir, evitar que aplicaciones con verdadero potencial de retención sean clasificadas erróneamente como casos de bajo desempeño.

En la gestión de productos digitales, este enfoque permite identificar con mayor precisión las aplicaciones con baja probabilidad de retención, facilitando la detección temprana de problemas de engagement. Además, optimiza el uso de recursos al dirigir estrategias de fidelización y mejoras técnicas hacia las apps con mayor riesgo de abandono. Finalmente, la capacidad del modelo para anticipar riesgos incluso considerando solo atributos intrínsecos —como tamaño o precio— ofrece a los desarrolladores una herramienta analítica para detectar problemas estructurales antes de que afecten la tracción social en el mercado.

2.3.2.4 Impacto de la Calidad Técnica y la Validación Social

La ingeniería de características es esencial para comprender y predecir la retención de usuarios en productos digitales. En aplicaciones móviles, métricas como reseñas e instalaciones suelen interpretarse como señales de aceptación y participación, influyendo en la percepción de calidad y en la decisión de descarga. Sin embargo, al aplicar pruebas de robustez que eliminan la dependencia estructural entre estas métricas sociales y la variable de retención, se observa que los atributos técnicos adquieren un papel central. El tamaño de la aplicación (Size_MB) alcanza una importancia destacada en el modelo de Gradient Boosting, lo que sugiere que la eficiencia técnica condiciona la permanencia inicial del usuario.

Este hallazgo reinterpreta la lógica de la Teoría de Inversión-Satisfacción: la participación activa del usuario debe entenderse más como consecuencia del uso prolongado que como causa primaria de la retención. Antes de consolidar un compromiso sostenido, deben existir condiciones técnicas mínimas que faciliten la adopción y eviten el abandono temprano.

La retención de usuarios depende de la interacción entre factores técnicos y sociales. La validación social refuerza la fidelidad en el largo plazo, mientras que la calidad técnica del software asegura la permanencia inicial. Por ello, las estrategias de gestión deben priorizar la optimización del rendimiento y la reducción de fricciones técnicas, complementándolas con mecanismos de interacción y retroalimentación que fortalezcan el compromiso sostenido del usuario.

3 Conclusiones y Recomendaciones

3.1 Conclusiones

Los resultados obtenidos en la fase de modelado nos permiten concluir:

Se validó que la retención de usuarios puede estimarse de manera confiable mediante una proxy metodológica que integra el Rating y el Engagement, cuando los datos no son de carácter longitudinal. El uso de esta proxy permitió transformar datos estáticos de Google Play Store en un indicador de lealtad de mercado, ofreciendo una visión más del comportamiento de los usuarios. Se validó que la retención puede estimarse mediante una proxy (Rating y Engagement), pero se determinó que su eficacia predictiva real debe medirse excluyendo los componentes directos del ratio (Reviews e Installs) para evitar conclusiones circulares basadas en dependencias estructurales. Además, la comparación de distintos modelos predictivos evidenció que los algoritmos de ensamble son los más adecuados para capturar la complejidad y multidimensionalidad del comportamiento digital.

Aunque inicialmente Reviews e Installs aparecen como los predictores dominantes (92.4%), el análisis de robustez demostró que esto se debe a una dependencia estructural. Al controlar este efecto, emergen factores intrínsecos como el tamaño de la app (Size_MB) y el precio como los verdaderos motores predictivos previos a la consolidación de la app en el mercado

Asimismo, se identificó que la sensibilidad precio-lealtad varía según el contexto de validación de la app. Si bien en el modelo inicial el precio mostró una relevancia baja, el análisis de robustez (excluyendo métricas de volumen) reveló que el precio se posiciona como el segundo factor más influyente (30.18% de importancia).

El modelo Gradient Boosting obtuvo el desempeño más alto en el escenario inicial (AUC = 0.91); sin embargo, al aplicar la prueba de robustez para mitigar la dependencia estructural con la variable objetivo, el rendimiento se ajustó a un AUC de 0.61. Este resultado es metodológicamente más preciso, ya que indica que, aunque los algoritmos de ensamble capturan eficazmente la complejidad del fenómeno, una parte significativa de su precisión inicial se debía a la relación aritmética entre los predictores de volumen y el ratio de retención definido.

Finalmente, el análisis de robustez realizado (excluir las variables que componen el ratio de la variable objetivo) resultó determinante para validar la integridad del estudio. Si bien el modelo inicial presentaba una precisión excepcional (AUC = 0.91) impulsada por métricas de volumen, la prueba de robustez permitió identificar que, en ausencia de una base consolidada de usuarios, factores como el tamaño de la aplicación (Size_MB) y el precio emergen como los predictores reales de éxito, con una importancia relativa del 31.96% y 30.18% respectivamente. Este hallazgo es crucial para la toma de decisiones estratégicas, ya que demuestra que la retención no depende únicamente de la inercia del mercado, sino de la optimización técnica y económica del producto en sus etapas tempranas, garantizando así un modelo libre de dependencias estructurales.

La investigación evidencia que la optimización técnica tiene un efecto directo y medible en la permanencia de los usuarios. En particular, el tamaño de la aplicación emerge como la variable más influyente en el escenario de robustez, con un peso del 42,17%, lo que representa un hallazgo clave para los desarrolladores. Este resultado indica que, más allá de las estrategias de marketing orientadas a incrementar las descargas, factores técnicos como la eficiencia en el peso de la aplicación y una adecuada optimización del rendimiento desempeñan un papel fundamental para reducir el abandono temprano. En consecuencia, una aplicación ligera y técnicamente optimizada presenta una mayor probabilidad de retener a sus usuarios, incluso independientemente del número inicial de reseñas o del nivel de visibilidad en la tienda.

3.2 Limitaciones

Pese a que los resultados del estudio son sólidos se presentan las siguientes limitaciones:

La principal limitación recae en la fuente de información dado que el dataset de Google Play Store contiene información de métricas agregadas a nivel de aplicación; al no contar con datos individuales el estudio no tiene la capacidad de modelar el comportamiento de los usuarios ni predecir el momento exacto en el que ocurrirá la desinstalación de la aplicación por parte del usuario. Es necesario recalcar que los resultados no miden la retención real de usuarios individuales, sino el desempeño relativo de las aplicaciones bajo métricas de éxito agregadas, es decir no es un seguimiento micro-analítico del ciclo de vida del cliente, los resultados son a nivel macro - analítico deben entenderse como una herramienta de análisis de mercado

Otra limitación que está presente en este estudio es el uso de una variable proxy que captura la retención, pese a que su construcción tiene validez al considerar métricas de éxito estándar en las tiendas de aplicaciones; podría no representar con precisión el comportamiento real del

usuario, lo que limita la solidez de las inferencias sobre retención al intentar extenderlas a un carácter individual.

La capacidad de generalizar los hallazgos obtenidos a otros contextos tanto de mercado como de tiempo (validez externa) está restringida por el uso de datos de carácter transversal y la dependencia de un indicador indirecto. Si se dispusieran de datos longitudinales que capturen información a nivel de individuo se podría medir directamente la variable churn y aplicar análisis de tiempo hasta el evento, permitiendo estimar no solo si un usuario abandona sino cuándo y en qué condiciones específicas. En ese escenario, algunos elementos del pipeline (como la limpieza de datos, extracción de características relevantes) serían transferibles, pero debería realizarse una recalibración del umbral de abandono para ajustarse a logs de actividad diaria.

Una limitación metodológica identificada es la dependencia estructural entre la variable objetivo (High_Retention) y los predictores Reviews e Installs. Dado que el target es un cociente de ambos, existe un riesgo de circularidad matemática que el indicador VIF no detecta por su naturaleza no lineal. Para mitigar esta limitación, el estudio incluyó un escenario de robustez que excluye dichos predictores, permitiendo aislar el efecto real de las características intrínsecas de la aplicación.

Si bien los valores de VIF (<1.7) confirman la ausencia de multicolinealidad lineal entre predictores, esta métrica no garantiza la ausencia de data leakage, lo que justifica el análisis comparativo de modelos con y sin variables de volumen.

Es importante mencionar que intentar extrapolar estos hallazgos podria generar algunos problemas dado que se usó solo datos de Google Play, los resultados reflejan el comportamiento del mercado Android, el cual podría diferir del iOS en perfiles de usuario y patrones de retención. Nuestros modelos deben considerarse como un marco de referencia que necesitaría ajustes antes de aplicarse a otros ecosistemas digitales.

Otra limitación del estudio es el tipo de datos usados dado que estos corresponden a un corte estático, esto impide un análisis dinámico ya que no se puede monitorear constantemente.

El análisis no contempla factores externos como la saturación publicitaria que podrían inflar artificialmente el número de instalaciones y desinstalaciones.

3.3 Recomendaciones

A partir de los hallazgos de esta investigación, se proponen las siguientes líneas de estudio:

Para futuras investigaciones se recomienda el uso de un dataset especifico de una aplicación para acceder a datos de eventos individuales. Esto permitiría abandonar el uso de variables proxy y modelar el churn real, mediante análisis de supervivencia (Time-to-Event). Este enfoque eliminaría la dependencia estructural identificada en este estudio, permitiendo observar cómo las acciones del usuario preceden temporalmente al abandono.

Dado que el volumen de Reviews resultó ser un predictor dominante pero con riesgo de circularidad matemática, se sugiere emplear análisis de sentimientos y modelado de temas (LDA). Esto permitiría transformar el dato cuantitativo en información cualitativa, identificando si la retención está ligada a la estabilidad técnica, la usabilidad de la interfaz o la satisfacción con las funcionalidades, factores que los modelos de ensamble actuales solo detectan de forma indirecta a través del volumen de reseñas.

Otra sugerencia es la capacidad de integrar variables externas que capturen el contexto competitivo, tales como la inversión en User Acquisition (UA), la saturación de la categoría o la estacionalidad del mercado. Al aplicar un enfoque de series temporales, se podría analizar cómo fluctuaciones externas impactan en la retención, permitiendo separar el éxito orgánico del éxito impulsado por campañas publicitarias, una distinción que los datos transversales actuales no permiten aislar.

Considerando que este estudio se centró en Google Play Store, para futuras investigaciones se sugiere comparar modelosen el ecosistema de iOS. Esto permitiría verificar si la importancia de variables como el precio y el tamaño de la app (identificadas como críticas en nuestro modelo de robustez) se mantiene constante entre diferentes perfiles socioeconómicos y técnicos de usuarios.

Referencias

AppsFlyer. (2020, 11 de febrero). *Las desinstalaciones de las apps móviles generan pérdidas promedio de más de 47 000 euros al mes*. https://www.appsflyer.com/es/company/newsroom/pr/uninstalls-mobile-apps-generate-losses/

Apptentive & Alchemer. (2022). 2022 Mobile Customer Engagement Benchmark Report [Archivo PDF]. Apptentive y Alchemer. https://www.alchemer.com/wp-content/uploads/2023/02/Apptentive_Alchemer-2022-Mobile-Consumer-Engagement-Report.pdf

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 738). New York: springer.

Boozary, P., Sheykhan, S., GhorbanTanhaei, H., & Magazzino, C. (2025). *Enhancing customer retention with machine learning: A comparative analysis of ensemble models for accurate churn prediction*. *International Journal of Information Management Data Insights, 5*(1), 100331. https://doi.org/10.1016/j.jjimei.2025.100331 (sciencedirect.com)

Box, G. E., & Draper, N. R. (1969). Evolutionary operation: a statistical method for process improvement.

Crudu, A., & MoldStud Research Team. (2024, 29 de marzo). Leveraging machine learning for predictive analytics in native mobile apps. MoldStud. https://moldstud.com/articles/p-leveraging-machine-learning-for-predictive-analytics-in-native-mobile-apps

Davenport, T. H., Harris, J. G., & Morison, R. (2010). *Analytics at work: Smarter decisions, better results*. Harvard Business Press.

Dogan, O., Hiziroglu, A., Pisirgen, A., & Seymen, O. F. (2025). Business analytics in customer lifetime value: An overview analysis. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *15*(1), e1571.

Eda Kavlakoglu, K. O'Brien & A. Downie. (s.f.). ¿Qué es la retención de clientes? IBM. https://www.ibm.com/mx-es/think/topics/customer-retention

Gupta, L. (2018). Google Play Store Apps [Conjunto de datos]. Kaggle. https://www.kaggle.com/datasets/lava18/google-play-store-apps

Gupta, R. K., Bharti, S., Pathik, N., & Sharma, A. (2022). Predicting Churn of Credit Card Customers Using Machine Learning and AutoML. *International Journal of Information Technology Project Management (IJITPM)*, *13*(3), 1-19.

Hossain, M. R. (2025). Predicting Customer Churn in Telecommunications with Machine Learning Models. *Asian Journal of Research in Computer Science*, *18*(1), 53-66.

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression*. John Wiley & Sons.

Hyrynsalmi, S., Suominen, A., & Mäntymäki, M. (2015, agosto). The role of developer multi-homing and keystone developers in mobile ecosystem competition. Ponencia en la 2015 Annual Meeting of the Academy of Management, Vancouver, Canadá. https://www.researchgate.net/publication/282670619_The_role_of_developer_multi-homing_and_keystone_developers_in_mobile_ecosystem_competition

Kavitha, V., Kumar, G. H., Kumar, S. V. M., & Harish, M. (2020). Churn Prediction of Customer in Telecom Industry using Machine Learning Algorithms. *International Journal of Engineering Research & Technology (IJERT)*, *9*(05), 181–184

Kulkarni, A., Patil, A., Bhoite, S., & Patil, M. (2019). Customer Churn Analysis and Prediction. *International Journal of Computer Applications Technology and Research*, *8*(09), 363-366. ISSN: 2319-8656.

Naim, I., Rajuddin, W. O. N., & Ansyori, A. (2024). Customer relationship management in the digital era to enhance customer experience through technology. *Transforma Jurnal Manajemen*, *2*(2), 65-71. https://doi.org/10.56457/tjm.v2i2.131

Sigg, S., Lagerspetz, E., Peltonen, E., Nurmi, P., & Tarkoma, S. (2016). Sovereignty of the apps: there's more to relevance than downloads. arXiv preprint arXiv:1611.10161.

Sivakani, R., Rajasekaran, G., Manimaran, S., Ali, M. M. S., Rajest, S. S., & Regin, R. (2025). Predicting User Engagement in Mobile Applications Using Machine Learning: Insights for Optimizing App Design and Retention. *Central Asian Journal of Mathematical Theory and Computer Sciences*, *6*(3), 472-489. https://cajmtcs.centralasianstudies.org/index.php/CAJMTCS

Song, P. (2025, 23 de noviembre). ML models for user retention prediction in mobile apps. ML Journey. https://mljourney.com/ml-models-for-user-retention-prediction-in-mobile-apps/

Turner, A. (2025, January 5). *How many apps are there in the world (2025)*. BankMyCell. https://www.bankmycell.com/blog/number-of-mobile-apps-worldwide

Universidad de Chile. (2023). Caracterización y predicción de conducta de usuarios de aplicación móvil enfocado a proceso On-Boarding. Tesis de grado.

Wu, P., Zhang, R., Zhu, X., & Liu, M. (2022). Factors Influencing Continued Usage Behavior on Mobile Health Applications. *Healthcare*, *10*(2), 208. https://doi.org/10.3390/healthcare10020208

YouAppi Marketing. (2025, 2 de julio). Retention vs. acquisition: Striking the right balance. YouAppi. https://youappi.com/blog/retention-vs.-acquisition

Zeithaml, V. A., Berry, L. L., & Parasuraman, A. (1996). The behavioral consequences of service quality. *Journal of marketing*, *60*(2), 31-46.

Anexos

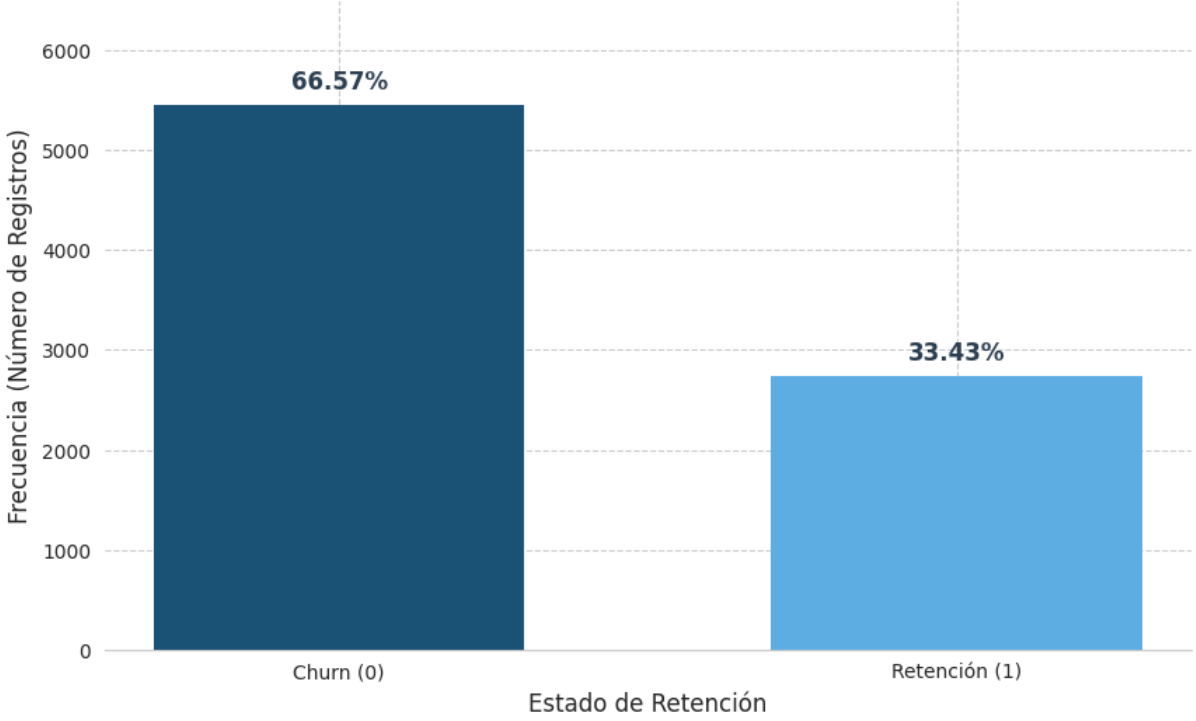
Anexo A

Tabla A *Clasificación de variables cualitativas y cuantitativas del modelo*

Variable	Tipo de Dato	Clasificación
App	object	Cualitativa
Category	object	Cualitativa
Rating	float64	Cuantitativa
Reviews	int64	Cuantitativa
Size	object	Cualitativa
Installs	int64	Cuantitativa
Type	object	Cualitativa
Price	float64	Cuantitativa
Content Rating	object	Cualitativa
Genres	object	Cualitativa
Last Updated	object	Cualitativa
Current Ver	object	Cualitativa
Android Ver	object	Cualitativa
Size_MB	float64	Cuantitativa
Engagement	float64	Cuantitativa
High_Rated	int64	Cualitativa

Anexo B

Ilustración 8 *Distribución variable objetivo*



Anexo C

Tabla 8 *Importancia*

Variable	Gradient Boosting	
	Modelo Original	**Modelo sin fuga**
Reviews	52.73%	
Installs	39.74%	
Type	3.94%	3.89%
Price	1.46%	30.18%
Category	1.02%	16.48%
Size_MB	0.84%	42.17%
Content Rating	0.28%	7.28%

Anexo C- Pipeline en Python

a) Librerías

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.stats.outliers_influence import variance_inflation_factor
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.metrics import (roc_auc_score, roc_curve, accuracy_score, recall_score,
                             precision_score, f1_score, auc, ConfusionMatrixDisplay)
```

b) Procesamiento y limpieza

```
# Carga y limpieza inicial
df = pd.read_csv('googleplaystore.csv')
df.drop_duplicates(subset='App', inplace=True)
df = df[df['Rating'].notna() & (df['Installs'] != 'Free')]

# Normalización de métricas
df['Installs'] = df['Installs'].apply(lambda x: int(str(x).replace(',', '').replace('+', '')))
df['Price'] = df['Price'].apply(lambda x: float(str(x).replace('$', '')) if '$' in str(x) else 0.0)

def clean_size(val):
    val = str(val)
    if 'M' in val: return float(val.replace('M', ''))
    if 'k' in val: return float(val.replace('k', '')) / 1024
    return np.nan

df['Size_MB'] = df['Size'].apply(clean_size).fillna(df['Size_MB'].mean())
```

c) Definición variable objetivo

```
df['Engagement_Rate'] = df['Reviews'] / (df['Installs'] + 1)
df['High_Retention'] = ((df['Rating'] > 4.2) &
                        (df['Engagement_Rate'] > df['Engagement_Rate'].median())).astype(int)
```

d) VIF

```
X_vif_input = df[['Reviews', 'Size_MB', 'Installs', 'Price']]
vif_data = [variance_inflation_factor(X_vif_input.values, i) for i in
range(len(X_vif_input.columns))]
```

e) Modelado y evaluación

```
# Partición del Dataset
# Usamos un 20% para test y 80% para entrenamiento.
# 'stratify' asegura que la proporción de éxito/fracaso sea igual en ambos grupos.
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.20, random_state=42, stratify=y
)

# Escalado de Características
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Entrenamiento de Modelos
models = {
    "Regresión Logística": LogisticRegression(max_iter=1000, class_weight='balanced'),
    "Random Forest": RandomForestClassifier(n_estimators=100, random_state=42),
    "Gradient Boosting": GradientBoostingClassifier(learning_rate=0.1, n_estimators=100,
random_state=42)
}

results = {}

print("--- Evaluación de Modelos ---")
for name, model in models.items():
    # Ajuste del modelo
    model.fit(X_train_scaled, y_train)

    # Predicciones
    y_pred = model.predict(X_test_scaled)
    y_proba = model.predict_proba(X_test_scaled)[:, 1]

    # Métricas de desempeño
    auc_score = roc_auc_score(y_test, y_proba)
    f1 = f1_score(y_test, y_pred)
    acc = accuracy_score(y_test, y_pred)
    results[name] = {"AUC": auc_score, "F1": f1, "Accuracy": acc}
    print(f"{name}: AUC-ROC = {auc_score:.4f} | F1-Score = {f1:.4f} | Acc = {acc:.4f}")

# CURVA ROC
plt.figure(figsize=(10, 6))
for name, model in models.items():
    y_proba = model.predict_proba(X_test_scaled)[:, 1]
    fpr, tpr, _ = roc_curve(y_test, y_proba)
    plt.plot(fpr, tpr, label=f'{name} (AUC = {auc_score:.2f})')

plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel('Tasa de Falsos Positivos')
plt.ylabel('Tasa de Verdaderos Positivos')
```

```
plt.title('Comparación de Curvas ROC')
plt.legend()
plt.show()
```

f) Interpretabilidad del Modelo

```
df_importancia = pd.DataFrame({
    'Variable': features,
    'Importancia': models["Gradient Boosting"].feature_importances_
}).sort_values(by='Importancia', ascending=False)
```