

FICHA RESUMEN

Título	MANTENIMIENTO PREVENTIVO EN EQUIPOS HOTELEROS CON BASE EN OCUPACIÓN, DAÑOS HISTÓRICOS Y RESERVAS FUTURAS.
Area de trabajo	Maestría en Ciencia de Datos
Tipo de proyecto	Investigación
Estudiante	Carlos Eduardo De la Ossa Pabón
Correo electrónico	cdelaossap@javerianacali.edu.co
Dirección y Teléfono	Carrera 71 # 94 – 124 Torre 3 Apto 403; 3015778287
Director	María Constanza Pabón Burbano
Correo director	mcpabon@javerianacali.edu.co
Palabras clave	Mantenimiento preventivo, hoteles, equipos hoteleros, aprendizaje automatizado
Fecha de inicio	15 de noviembre de 2023
Duración	27 meses (completado en febrero 2026)
Resumen	En el sector hotelero, la operatividad continua de las áreas generadoras de ingresos es esencial. La interrupción no planificada o la entrega de una experiencia deficiente puede resultar en costos significativos a corto y largo plazo. Este proyecto desarrolló un modelo de predicción del momento de fallo de equipos hoteleros, utilizando la ocupación histórica del hotel, reservas futuras y datos históricos de mantenimiento como indicadores clave. Se trabajó con datos reales del sistema Nuvola, plataforma SaaS de gestión hotelera, correspondientes a cuatro hoteles independientes en Estados Unidos con periodos de operación entre 2020 y 2026. Dada la limitación de los datos reales (ausencia de clasificación preventiva/correctiva y de indicadores de condición), se diseñaron tres escenarios de análisis: (1) datos reales con las variables disponibles, (2) datos sintéticos calibrados con perfiles de madurez por hotel que incluyen el campo condicion_post, y (3) escenario ideal que agrega variables físicas de vida útil de los equipos. La metodología siguió el framework CRISP-DM e implementó, desde cero en Python, análisis de supervivencia mediante Kaplan-Meier, regresión logística con regularización L2 y una validación cruzada

	Leave-One-Hotel-Out (LOHO-CV) diseñada específicamente para respetar la independencia entre propiedades hoteleras. Los resultados del Escenario 1 con datos reales muestran un AUC LOHO de 0.724. La comparación de seis técnicas de aprendizaje automático —regresión logística, LR con feature engineering, SVM lineal, SVM con kernel RBF, Random Forest y Gradient Boosting— confirma que el Gradient Boosting obtiene el mejor AUC (0.738), aunque la diferencia entre técnicas es menor que la varianza interhotel (± 0.15), evidenciando que el volumen del dataset es la limitación principal. El sistema fue desplegado localmente mediante una API REST con FastAPI y Docker, permitiendo el ingreso de datos en tiempo real sobre la ocupación y las variables identificadas. Los resultados demuestran la viabilidad técnica del enfoque predictivo para la gestión de mantenimiento hotelero, y establecen un esquema mínimo de datos que los hoteles deben registrar para implementar modelos similares.
--	---



**MANTENIMIENTO PREVENTIVO EN EQUIPOS HOTELEROS CON BASE EN OCUPACIÓN,
DAÑOS HISTÓRICOS Y RESERVAS FUTURAS.**

Carlos Eduardo De la Ossa Pabón

Código 8974525

Proyecto Aplicado para optar al título de
Magister en Ciencia de Datos

Director(a):

María Constanza Pabón Burbano

**FACULTAD DE INGENIERÍA Y CIENCIAS
MAESTRÍA EN CIENCIA DE DATOS
BARRANQUILLA, FEBRERO 2026**

TABLA DE CONTENIDO

FICHA RESUMEN.....	1
TABLA DE CONTENIDO	4
INTRODUCCIÓN	6
1. DEFINICIÓN DEL PROBLEMA	7
1.1. PLANTEAMIENTO DEL PROBLEMA.....	7
1.2. FORMULACIÓN DEL PROBLEMA.....	7
2. OBJETIVOS DEL PROYECTO	9
2.1. OBJETIVO GENERAL.....	9
2.2. OBJETIVOS ESPECÍFICOS.....	9
3. MARCO DE REFERENCIA.....	10
3.1. MARCO TEORICO	10
3.1.1. Mantenimiento Preventivo	10
3.1.2. Aprendizaje automático.....	10
3.1.3. Análisis de Supervivencia y Kaplan-Meier	12
3.1.4. Validación Cruzada por grupos Leave-One-Group-Out (LOGO-CV)	12
3.1.5. Metodología CRISP-DM	13
3.1.6. Métricas de Evaluación de Clasificadores Binarios	13
3.1.7. Label Noise en Clasificación	14
3.2. ANTECEDENTES	14
3.2.1. Study on Resource Scheduling Method of Predictive Maintenance for Equipment Based on Knowledge [4]	14
3.2.2. Development of Predictive Maintenance Interface Using Multiple Linear Regression [5]	15
3.2.3. Implementing Deep Learning Model to Predict the Maintenance of an Elevator System [6]	15
3.2.4. Machine Failure Prediction Using Survival Analysis [18].....	16
4. ACTIVIDADES Y RESULTADOS - OBJETIVO ESPECÍFICO 1	17
4.1. Fuente de Datos: Plataforma Nuvola	17
4.2. Análisis Exploratorio y Variables Identificadas	18
4.3. Correlación con el Evento Objetivo	19
4.4. Limitaciones de los Datos Reales.....	22
5. ACTIVIDADES Y RESULTADOS - OBJETIVO ESPECÍFICO 2	24
5.1. Técnicas Identificadas en la Literatura	24
5.2. Justificación de la Técnica Seleccionada.....	24

6. ACTIVIDADES Y RESULTADOS - OBJETIVO ESPECÍFICO 3	26
6.1. Diseño del Escenario de Modelado	26
6.2. Escenario 1 — Datos Reales	26
6.3. Comparación de Técnicas de Aprendizaje Automático.....	29
6.4. Trabajo Futuro Propuesto: Escenarios con Datos Enriquecidos (E2 y E3).....	32
6.4.1. Proceso de generación de datos sintéticos, Escenario 2:.....	32
6.4.2. Proceso de generación de datos sintéticos, Escenario 3:.....	33
6.4.3. Trabajo futuro adicional. Modelo único vs. modelos independientes por hotel:	33
6.5. Análisis de Kaplan-Meier.....	34
6.6. Política de Alertas	36
7. ACTIVIDADES Y RESULTADOS - OBJETIVO ESPECÍFICO 4	38
7.1. Arquitectura del Sistema.....	38
7.2. Endpoints de la API	38
7.3. Esquema de Entrada para el Ingreso de Datos	39
7.4. Instrucciones de Despliegue	40
8. CONCLUSIONES Y TRABAJOS FUTUROS	41
8.1. CONCLUSIONES	41
8.2. TRABAJOS FUTUROS.....	43
9. REFERENCIAS BIBLIOGRÁFICAS	45
ANEXO A - GLOSARIO DE TÉRMINOS	47

INTRODUCCIÓN

En el sector hotelero, la continuidad operativa es esencial para asegurar el éxito económico. La variación en la ocupación y los tiempos de inactividad no programados afectan la experiencia del usuario, generan pérdida de eficiencia en áreas fundamentales e impactan los ingresos a corto y largo plazo al socavar la confianza del cliente y la calificación del hotel.

En este contexto, la gestión eficiente del mantenimiento se vuelve crucial. Sin embargo, la falta de un enfoque predictivo dificulta la programación de mantenimientos preventivos, relegando a menudo las intervenciones a un enfoque más correctivo. Los costos asociados a reparar algo son siempre mayores que los de prevención.

Este proyecto tuvo como objetivo principal desarrollar un modelo avanzado de predicción de momentos de fallos en equipos hoteleros, basándose en la ocupación histórica y futura del hotel, así como en datos de mantenimiento previos. El modelo sirve como herramienta estratégica para los administradores de mantenimiento, permitiendo la anticipación de momentos críticos de fallo y la programación proactiva de intervenciones.

El trabajo se estructuró siguiendo la metodología CRISP-DM (Cross-Industry Standard Process for Data Mining), partiendo del entendimiento del negocio hotelero, pasando por la preparación y modelamiento de datos, hasta el despliegue de un sistema de predicción local. Dada la naturaleza de los datos disponibles, se diseñaron tres escenarios de creciente completitud: datos reales con limitaciones reconocidas, datos sintéticos calibrados que incluyen el campo `condicion_post` (condición del equipo al cerrar cada orden de trabajo), y un escenario ideal que incorpora variables físicas de vida útil de los equipos.

En las siguientes secciones se detallan la formulación del problema, los objetivos del proyecto, el marco teórico y de antecedentes, las actividades y resultados obtenidos para cada objetivo específico, y las conclusiones y trabajos futuros.

1. DEFINICIÓN DEL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

En el sector hotelero, es fundamental garantizar la continuidad operativa de todas las áreas que son fuentes de ingreso económico. Y aunque en un hotel se prestan otros servicios como actividades recreativas, spa, ventas de productos, programas de membresía, entre otros, la fuente base de ingresos es el alquiler de las habitaciones y el servicio de alimentación. Si estas áreas fundamentales se ven afectadas por tiempos de inactividad no programados o dejan de ofrecer una buena experiencia al usuario, generará un alto costo para el hotel a corto y largo plazo para recuperar la buena calificación y confianza de sus clientes.

Los hoteles no duermen. Un hotel es una empresa con un sistema de engranajes que debe operar 24 horas, 7 días a la semana [1], sin siquiera poder descansar en festivos; una empresa donde las siglas OOI (Out of Inventory), OOO (Out Of Order), OOS (Out Of Service) son ampliamente conocidas y bien podrían agrupar su definición como 'pérdida de dinero'. Esto se representa en los distintos turnos que tienen para sus empleados, donde pueden incluso existir 3 turnos por día para cubrir las 24 horas.

La variación en la ocupación del hotel, el costo de colocar en OOO una habitación, así como la falta de un enfoque predictivo, hacen muy difícil la programación de mantenimientos preventivos, por lo que la función del mantenimiento se vuelve más correctiva que otra cosa, y los costos asociados de arreglar algo en general siempre son mayores que evitar que se dañe [13].

1.2. FORMULACIÓN DEL PROBLEMA

Considerando la información anterior, ¿cómo desarrollar un modelo que prediga momentos de daño en equipos hoteleros utilizando la ocupación pasada y futura del hotel, al igual que datos históricos de mantenimiento, como indicadores clave para que los administradores de mantenimiento programen intervenciones en los equipos y eviten tiempos de inactividad no programados?

Para la resolución de la pregunta problema, se establecieron las siguientes preguntas de sistematización:

- ¿Cuáles son las variables potenciales que podrían estar correlacionadas con los momentos de daño en los equipos hoteleros?
- ¿Qué técnicas y metodologías han demostrado ser exitosas en proyectos similares de predicción de daños en equipos?
- ¿Cómo se estructurará y entrenará el modelo de aprendizaje automático para predecir el momento de fallo de los equipos hoteleros? ¿Qué tipo de algoritmos de aprendizaje automático se usarán?
- ¿Cuáles son los requisitos técnicos para el despliegue local del sistema de predicción?

Se investigaron el modelo y las técnicas disponibles para predecir el momento del fallo de un equipo y su correlación con distintas variables, y de esta forma se desarrolló un modelo de aprendizaje que puede desplegarse localmente y utilizar la base de datos que se tiene.

2. OBJETIVOS DEL PROYECTO

2.1. OBJETIVO GENERAL

Desarrollar un modelo de predicción del momento de fallo del equipo, que, recibiendo información como la ocupación histórica del hotel y reservas futuras, al igual que históricos de daños como indicadores clave, sirva de herramienta para que administradores de mantenimiento programen intervenciones en equipos del hotel.

2.2. OBJETIVOS ESPECÍFICOS

- Identificar las variables que nos permitan predecir el momento de daño de los equipos.
- Investigar las técnicas utilizadas en proyectos similares para predecir el momento de daño de equipos.
- Desarrollar modelos de aprendizaje automático que permitan predecir el momento de fallo del equipo.
- Desplegar localmente el sistema de predicción desarrollado para ingresar datos en tiempo real sobre la ocupación y demás variables identificadas.

3. MARCO DE REFERENCIA

Se presentan a continuación conceptos y definiciones importantes para este documento, así como la investigación realizada de sistemas desarrollados de mantenimiento preventivo que hacen uso de la ciencia de datos para predecir el daño de equipos.

3.1. MARCO TEORICO

Se definen a continuación los conceptos teóricos básicos de este proyecto: mantenimiento preventivo, aprendizaje automático, análisis de supervivencia y metodología CRISP-DM.

3.1.1. Mantenimiento Preventivo

IBM define el mantenimiento preventivo como 'la realización de labores de mantenimiento programadas periódicamente con el fin de evitar futuras anomalías e imprevistos'. Se trata, en resumen, de arreglar los dispositivos antes de que fallen.' [2] Según IBM, existen cuatro tipos principales de mantenimiento preventivo:

- Según uso: Se determina la fecha de una tarea de mantenimiento o inspección futura, teniendo en cuenta el uso medio diario del activo, o la exposición a factores ambientales.
- Según plazo o calendario: La fecha de mantenimiento está planeada por un calendario basado en intervalos de tiempo.
- Predictivo: Se calcula el estado del equipo para calcular la fecha conveniente para realizar el mantenimiento. A continuación, se programan los eventos de mantenimiento necesarios para prevenir el fallo inesperado del equipo.
- Prescriptivo: En este tipo de mantenimiento, el sistema avisa qué pieza va a fallar y cuándo se producirá, además de indicar las causas.

3.1.2. Aprendizaje automático

El aprendizaje automático [3] es una rama de la inteligencia artificial que se centra en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender patrones y tomar decisiones basadas en datos, sin que hayan sido programadas exactamente para esa tarea. Pueden clasificarse en tres categorías principales:

- Aprendizaje Supervisado: El modelo se entrena utilizando un conjunto de datos etiquetado, donde las respuestas correctas son conocidas.

- Aprendizaje No Supervisado: El modelo se entrena utilizando un conjunto de datos sin etiquetar; el mismo modelo identifica patrones.
- Aprendizaje por refuerzo: El modelo interactúa con su entorno y recibe feedback en forma de recompensas o penalizaciones según las acciones que realiza.

Para este proyecto se utilizó aprendizaje supervisado, formulando el problema como clasificación binaria: ¿el equipo presentará un evento de mantenimiento en los próximos 30 días? Se compararon seis técnicas: (1) Regresión Logística L2, que estima $P(y=1|X)$ mediante la función sigmoide; (2) Regresión Logística con Feature Engineering, que agrega tendencia de ocupación e interacción ocupación por eventos; (3) SVM Lineal ($C=1.0$), que maximiza el margen de separación; (4) SVM con kernel RBF via Random Fourier Features, para relaciones no lineales; (5) Random Forest (80 árboles, profundidad máx. 5), basado en muestreo bootstrap; y (6) Gradient Boosting (80 estimadores, learning rate 0.10, profundidad 3), que corrige errores residuales en secuencia.

La Regresión Logística es un modelo de clasificación binaria que estima la probabilidad $P(y=1|X)$ mediante la función sigmoide $\sigma(z) = 1/(1+e^{-z})$, aplicada a una combinación lineal de las variables predictoras: $P(y=1|X) = \sigma(w^T x + b)$. Los parámetros se estiman maximizando la verosimilitud logarítmica del conjunto de entrenamiento. La regularización L2 (Ridge) agrega una penalización $\lambda ||w||^2$ a la función de costo, reduciendo el sobreajuste al desincentivar coeficientes de gran magnitud [11].

Las Máquinas de Vectores de Soporte (SVM) buscan el hiperplano que maximiza el margen de separación entre clases. Para datos no linealmente separables, la variante de margen suave ($C=1.0$) permite clasificaciones erróneas penalizadas por el parámetro C . El SVM con kernel RBF proyecta los datos a un espacio de mayor dimensión mediante $\kappa(x, x') = \exp(-\gamma ||x - x'||^2)$, permitiendo separar clases no linealmente separables en el espacio original; en este trabajo se aproxima mediante Random Fourier Features [8].

Los métodos de ensamble combinan múltiples modelos base para reducir el error de generalización. En el marco del dilema sesgo-varianza, el error de predicción se descompone en sesgo (error sistemático), varianza (sensibilidad a las fluctuaciones del dataset) y ruido irreducible. Los dos métodos de ensamble evaluados atacan componentes distintas de este error:

Random Forest [16] reduce la varianza mediante bagging (Bootstrap Aggregating): cada árbol se entrena sobre una muestra bootstrap independiente del dataset, y las predicciones se promedian. Adicionalmente, en cada nodo se selecciona un subconjunto aleatorio de variables candidatas (feature subsampling), lo que reduce la correlación entre árboles y amplifica el beneficio del promediado. La predicción final es la proporción de árboles que vota por la clase positiva.

Gradient Boosting [14] reduce el sesgo mediante boosting: construye secuencialmente una suma de modelos base $F(x) = \sum_m \alpha_m h_m(x)$, donde cada h_m corrige los errores residuales del modelo anterior minimizando una función de pérdida (log-loss) mediante descenso de gradiente en el espacio de funciones. A diferencia de Random Forest, que entrena árboles en paralelo, el Gradient Boosting los entrena en secuencia sobre los residuos del paso anterior. El learning rate α controla la contribución de cada árbol; árboles cortos (max_depth=3) actúan como aprendices débiles que controlan el sobreajuste.

3.1.3. Análisis de Supervivencia y Kaplan-Meier

El análisis de supervivencia es un conjunto de métodos estadísticos para analizar el tiempo hasta que ocurre un evento de interés. En el contexto de mantenimiento, el 'evento' es la falla del equipo y el 'tiempo' es el tiempo desde el último mantenimiento preventivo (TTF, Time to Failure).

El estimador Kaplan-Meier es un método no paramétrico para estimar la función de supervivencia $S(t) = P(T > t)$ [10], es decir, la probabilidad de que el equipo sobreviva más allá del tiempo t . En este proyecto se implementó desde cero utilizando NumPy, siguiendo la fórmula:

$$S(t) = \prod [1 - d_i / n_i] \text{ para todos los } t_i \leq t$$

Donde d_i es el número de fallas en el tiempo t_i y n_i es el número de equipos en riesgo justo antes de t_i . Este análisis permitió identificar el tiempo medio hasta falla (MTTF) diferenciado por condición post-mantenimiento (campo condicion_post en los Escenarios 2 y 3 que hacen parte de trabajo futuro).

3.1.4. Validación Cruzada por grupos Leave-One-Group-Out (LOGO-CV)

La validación cruzada es una técnica de evaluación de modelos que estima el desempeño en datos no vistos durante el entrenamiento. En su variante estándar k-fold, los datos se dividen aleatoriamente en k pliegues; sin embargo, cuando los datos presentan estructura de grupos naturales (pacientes, sitios geográficos, instituciones), la división aleatoria introduce fuga de información entre pliegues, produciendo estimaciones de desempeño sobreoptimistas [8].

La validación cruzada Leave-One-Group-Out (LOGO-CV) resuelve este problema reservando un grupo completo como conjunto de prueba en cada iteración, mientras los grupos restantes forman el conjunto de entrenamiento [8, 9]. Este esquema garantiza que el modelo nunca vea datos del grupo evaluado durante el entrenamiento, respetando la independencia entre grupos y produciendo una estimación conservadora pero válida de la capacidad de generalización a grupos no vistos. Es especialmente apropiada cuando el objetivo no es predecir dentro de los grupos conocidos sino generalizar a nuevos grupos futuros.

3.1.5. Metodología CRISP-DM

CRISP-DM (Cross-Industry Standard Process for Data Mining) [7] es una metodología estándar para proyectos de ciencia de datos que define seis fases iterativas: (1) Entendimiento del negocio, (2) Entendimiento de los datos, (3) Preparación de los datos, (4) Modelamiento, (5) Evaluación y (6) Despliegue. Este proyecto siguió estas fases de forma iterativa, especialmente entre las fases 2 - 4 al requerir tres escenarios de datos con distintos niveles de completitud.

3.1.6. Métricas de Evaluación de Clasificadores Binarios

La métrica principal utilizada en este proyecto es el AUC-ROC (Area Under the Receiver Operating Characteristic Curve) [17]. La curva ROC grafica la Tasa de Verdaderos Positivos ($TPR = Recall = TP/(TP+FN)$) frente a la Tasa de Falsos Positivos ($FPR = FP/(FP+TN)$) para todos los umbrales posibles de decisión. El AUC representa la probabilidad de que el modelo asigne mayor probabilidad de falla a un equipo que sí fallará frente a uno que no fallará. $AUC = 0.5$ equivale a clasificación aleatoria; $AUC = 1.0$ es perfecto.

La precisión ($precision = TP/(TP+FP)$) mide qué fracción de los equipos alertados efectivamente falla en 30 días. El recall ($TP/(TP+FN)$) mide qué fracción de los equipos que van a fallar son detectados por el modelo. El F1-score es la media armónica de ambos: $F1 = 2 \cdot precision \cdot recall / (precision + recall)$. La curva Precisión-Recall (curva PR) es más informativa que la ROC cuando las clases están desbalanceadas, que es el caso aquí ($falla_{30d} = 1$

representa aproximadamente el 35% de las observaciones). Se utiliza el AUC-ROC como métrica principal por su independencia del umbral y por ser el estándar en mantenimiento predictivo en la literatura revisada.

3.1.7. Label Noise en Clasificación

El ruido de etiqueta (label noise) ocurre cuando las etiquetas de entrenamiento contienen errores sistemáticos que no reflejan la verdad que se quiere aprender [12]. En mantenimiento predictivo, el tipo más frecuente de label noise es la impureza de clase: eventos preventivos programados y eventos correctivos reactivos se etiquetan igual ($\text{falla}_{30d} = 1$) aunque su naturaleza es distinta.

El impacto del label noise en AUC-ROC es una sobreestimación del desempeño en la tarea mixta y una posible subestimación en la tarea pura de correctivos: el modelo aprende también los patrones de calendarización de preventivos, lo que puede inflar el AUC si los PMs son perfectamente periódicos. La cuantificación del impacto requiere un dataset con etiquetas limpias, disponible en el Escenario 2 (datos sintéticos con campo tipo). La diferencia entre E1 (AUC=0.724) y E2 (AUC=0.813) establece el techo de mejora posible al eliminar el ruido estructural: aproximadamente 0.09 puntos de AUC.

3.2. ANTECEDENTES

Para la realización de este proyecto, es fundamental la revisión de estudios y proyectos previos que han investigado sobre el mantenimiento preventivo y el uso de la ciencia de datos para la eficiencia, la reducción de costos y durabilidad de los equipos. Los antecedentes encontrados son todos internacionales; en la investigación realizada, no se evidenciaron antecedentes nacionales ni en el sector hotelero específicamente.

3.2.1. Study on Resource Scheduling Method of Predictive Maintenance for Equipment Based on Knowledge [4]

Los autores Xin Li, Jinqian Wen, Rui Zhou, Yaoguang Hu proponen un método de programación de recursos para servicios de mantenimiento predictivo, basándose en histórico de mantenimiento de fallas, habilidad de los técnicos, información del estado del equipo y plan de trabajo. Para eliminar fallas potenciales, minimizan el costo del servicio y el costo de interrupción, considerando la capacidad de los técnicos para diferentes tareas.

El modelo reportó una reducción del 15% en costos de mantenimiento no programado al optimizar la asignación de técnicos según su especialidad. La evaluación se realizó sobre datos sintéticos de 12 meses de operación en manufactura.

La diferencia con el proyecto realizado es que no se tiene en cuenta el conocimiento del técnico que realiza la intervención, sino el histórico de ocupación y daños.

3.2.2. Development of Predictive Maintenance Interface Using Multiple Linear Regression [5]

Los autores T. Abbasi, King Hann Lim, N.S Rosli, I. Ismail, R. Ibrahim desarrollan una interfaz analítica de datos de mantenimiento predictivo que, mediante regresión lineal múltiple, predice el daño de equipos industriales utilizando parámetros de sensores (ATP y ABD Gas flowrate, Air Temp Cold, Mot Air Temp, Pressure Discharge, Temperature Discharge).

La interfaz alcanzó un coeficiente de determinación $R^2 = 0.87$ en el conjunto de prueba, con un error cuadrático medio (RMSE) de 2.3 unidades en la predicción del índice de daño del equipo. La evaluación se realizó sobre datos reales de 6 meses de operación de equipos industriales.

La diferencia radica en que utilizan sensores para obtener señales y datos a través de la vida del equipo. En nuestro proyecto no utilizamos sensores, sino datos de ocupación y registros históricos de mantenimiento.

3.2.3. Implementing Deep Learning Model to Predict the Maintenance of an Elevator System [6]

Los autores Gaayan Verma, Jasmine Awatramani y Nitasha Hasteer abordan el mantenimiento preventivo en ascensores a través del uso de aprendizaje profundo (LSTM). Determinan el periodo de tiempo adecuado en el que se debe realizar el mantenimiento, obteniendo datos de sensores instalados en los ascensores. Los resultados experimentales presentan un error absoluto medio de alrededor del 7,05 %.

El modelo LSTM reportó un error absoluto medio (MAE) de 7.05% en la predicción del período óptimo de mantenimiento, evaluado sobre datos de 8 ascensores durante un período de observación de 24 meses. El modelo requirió sensores de acelerómetro y de corriente instalados en cada ascensor.

La diferencia de este proyecto consiste en que solo se aplica a elevadores y requiere sensores. Nuestro proyecto busca predecir el momento del daño en cualquier equipo hotelero, sin requerir instrumentación adicional.

3.2.4. Machine Failure Prediction Using Survival Analysis [18]

Los autores Dimitris Papathanasiou, Konstantinos Demertzis y Nikos Tziritas abordan la predicción de fallas de máquinas combinando análisis de supervivencia con aprendizaje automático en entornos de infraestructura tecnológica a gran escala. Utilizan un modelo de Random Survival Forest (RSF) con múltiples técnicas de selección de features (RadViz, COX, Rank2D, eliminación recursiva de variables) para filtrar información irrelevante y retener únicamente los predictores más significativos.

El modelo RSF alcanzó un C-index de 0.97, evaluado sobre 10,000 instancias de máquinas en entornos IoT y servidores de infraestructura crítica. La combinación de selección de features con análisis de supervivencia demostró ser superior a los enfoques que utilizan únicamente el modelo COX para la evaluación de variables.

La diferencia de este proyecto consiste en que Papathanasiou et al. trabajan con sistemas IoT y servidores con datos de sensores continuos, mientras que nuestro proyecto opera sobre registros de mantenimiento hotelero con datos escasos e incompletos, sin instrumentación adicional. Esto motivó el uso de clasificación binaria con horizonte de 30 días en lugar de modelos de tiempo hasta evento.

4. ACTIVIDADES Y RESULTADOS - OBJETIVO ESPECÍFICO 1

Identificar las variables que nos permitan predecir el momento de daño de los equipos.

4.1. Fuente de Datos: Plataforma Nuvola

Los datos provienen de Nuvola, una plataforma SaaS (Software as a Service) de gestión hotelera utilizada en hoteles independientes de Estados Unidos. Se exportaron nueve archivos XML (uno por módulo del sistema) para cuatro hoteles con los siguientes periodos de operación:

Hotel ID	Periodo	Ítems	Eventos
H-281	Jun 2020 – Abr 2023	52	2,675
H-334	Ago 2021 – Abr 2023	38	932
H-416	Jul 2021 – Oct 2024	45	817
H-314	Ene 2022 – Ene 2026	61	1,790

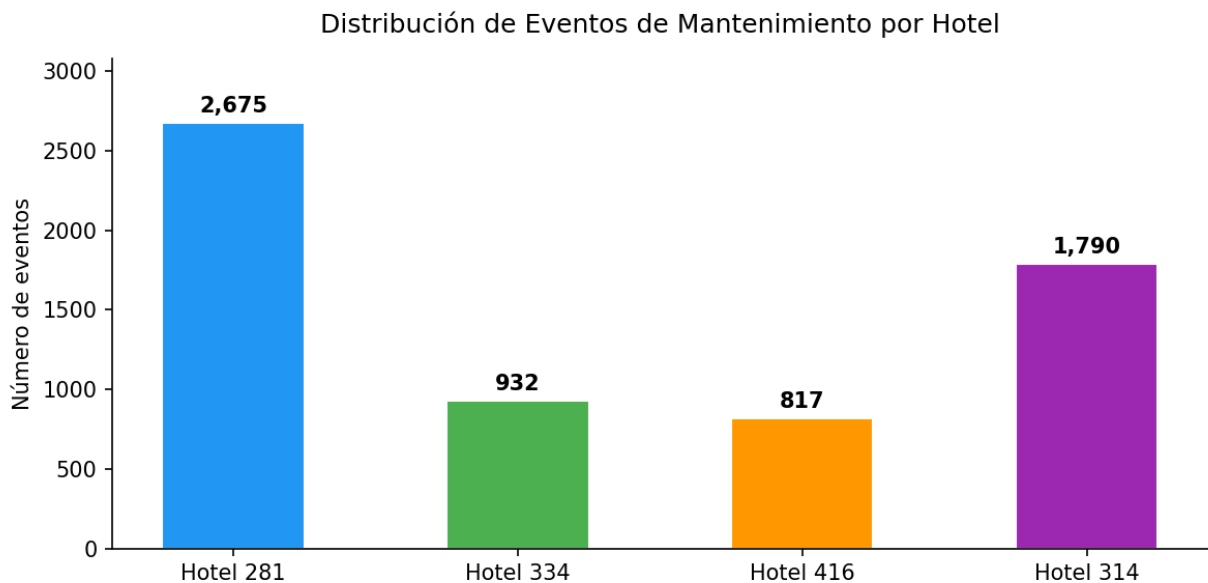


Figura 1. Distribución de eventos de mantenimiento por hotel (datos reales Nuvola, 2020-2026).

4.2. Análisis Exploratorio y Variables Identificadas

El análisis exploratorio inició con la inspección de los nueve módulos XML de Nuvola, identificando las tablas relevantes: `item_damaged` (eventos de mantenimiento), `items` (inventario de equipos activos del hotel: aires acondicionados, bombas, elevadores, calderas, generadores y demás activos con plan de mantenimiento registrado) y `occupancy` (datos de ocupación diaria por hotel).

El proceso de cargue y limpieza se implementó en Python, produciendo tres archivos maestros: `eventos_completos.csv` (6,214 eventos), `items_completos.csv` (196 equipos con plan activo) y `ocupacion_completa.csv` (3,276 registros diarios).

El proceso de limpieza implementado en Python incluyó los siguientes pasos: (1) Parsing de los nueve archivos XML por hotel (módulos: `item_damaged`, `items`, `occupancy`, entre otros) usando la librería `ElementTree`; (2) Eliminación de registros duplicados identificados por combinación `hotel_id` + `item_id` + `event_ts`; (3) Exclusión de equipos sin plan de mantenimiento activo (campo `maintenance_plan` = NULL), reduciendo el inventario de 241 a 196 ítems con plan activo; (4) Tratamiento de valores faltantes en ocupación: para H-334 y H-416 (sin datos de `occupancy` diaria completos) se imputó la ocupación media mensual del hotel como proxy; (5) Filtrado de eventos con timestamps inválidos o fuera del período de operación reportado por cada hotel. El resultado final fue un dataset de modelado con 6,214 eventos-observación, donde cada fila representa un equipo en una fecha de observación con sus variables calculadas hacia atrás (lookback) y la variable objetivo `falla_30d` calculada hacia adelante (lookahead de 30 días).

Cada fila del dataset de modelado corresponde a un equipo en una fecha de observación; las variables predictoras se calculan mirando hacia atrás (ventana histórica) y la variable objetivo `falla_30d` mira hacia adelante (horizonte de 30 días). La variable `guarantee` indica si el equipo tenía cobertura de garantía vigente del fabricante al momento de la observación (1 = sí, 0 = no); su valor se deriva del campo `warranty_expiry` del módulo `items` de Nuvola.

Dado que los datos reales del sistema Nuvola presentan limitaciones importantes - principalmente la ausencia del campo tipo (preventivo/correctivo) y del indicador de condición del equipo - el análisis se estructuró en tres escenarios de creciente completitud

de datos: Escenario 1 (E1) utiliza exclusivamente los datos reales disponibles; Escenario 2 (E2) enriquece el dataset con campos sintéticos calibrados que incluyen condicion_post y la clasificación preventivo/correctivo; y Escenario 3 (E3) agrega adicionalmente variables de vida útil física del equipo. Esta estructura de escenarios permite cuantificar el impacto de enriquecer los datos de mantenimiento, comparando el desempeño predictivo alcanzable con y sin información adicional.

Los Escenarios 2 y 3 se documentan como línea de trabajo futuro (sección 8.2), dado que su implementación real requiere modificaciones en el sistema Nuvola.

A continuación se presenta una tabla resumen de las variables identificadas como potencialmente predictoras, junto con su disponibilidad en cada escenario:

Variable	Descripción	Esc. 1	Esc. 2	Esc. 3
fail_count_90d	Eventos en 90d previos	Si	Si	Si
occ_rolling_30d	Ocupación media 30d	Si	Si	Si
dias_desde_ultimo	Días desde último evento	Si	Si	Si
condicion_post	Estado equipo post-PM (1-5)	No	Si	Si
pct_vida_util	% vida util consumida	No	No	Si
occ_7d (RESERVAS FUTURAS)	Ocup. próximos 7d desde PMS	Si	Si	Si

4.3. Correlación con el Evento Objetivo

Operacionalización del concepto de falla. En el marco de este proyecto, la predicción del momento de fallo se operacionaliza como la estimación del riesgo de ocurrencia de un evento de mantenimiento en un horizonte de 30 días. Dado que los datos de Nuvola no incluyen un campo tipo que distinga preventivos de correctivos, todos los registros se tratan como eventos de mantenimiento sin clasificar. La variable falla_30d captura cualquier intervención

en los próximos 30 días, introduciendo ruido estructural (label noise). Sin embargo, la formulación es operativamente válida: anticipar cualquier intervención permite programar recursos y minimizar habitaciones fuera de servicio. Esta aproximación es estándar en mantenimiento predictivo con datos operativos incompletos.

Antes de calcular la matriz de correlación, es importante definir las variables incluidas:

- **fail_count_90d** cuenta el número de eventos de mantenimiento registrados para ese equipo en los 90 días anteriores a la fecha de observación.
- **dias_desde_ultimo** registra cuántos días han pasado desde el último evento.
- **occ_7d**, **occ_30d** y **occ_90d** son la ocupación proyectada del hotel a 7, 30 y 90 días: provienen directamente del módulo de reservas del PMS y representan demanda confirmada, no estimada.
- **guarantee** es una variable binaria que indica si el equipo tenía garantía del fabricante vigente.
- **occ_mes** es la ocupación promedio histórica del mes calendario.
- Se construyó la variable objetivo binaria **falla_30d** (1 si el equipo presentó un evento de mantenimiento en los próximos 30 días, 0 en caso contrario) y se calcularon correlaciones de Pearson con las variables candidatas:

Variable	r con falla_30d	Interpretacion	Disponibilidad
condicion_post (1-5)	-0.467	Mejor condición -> menos fallas	E2, E3 (sintético)
fail_count_90d	+0.312	Más eventos previos -> mas riesgo	E1, E2, E3
dias_desde_ultimo	-0.185	Más tiempo sin PM -> mayor riesgo	E1, E2, E3
occ_mes	+0.113	Mayor ocupación -> más uso equipos	E1, E2, E3
pct_vida_util	+0.391	Equipo más viejo -> más riesgo	E3 (sintético)

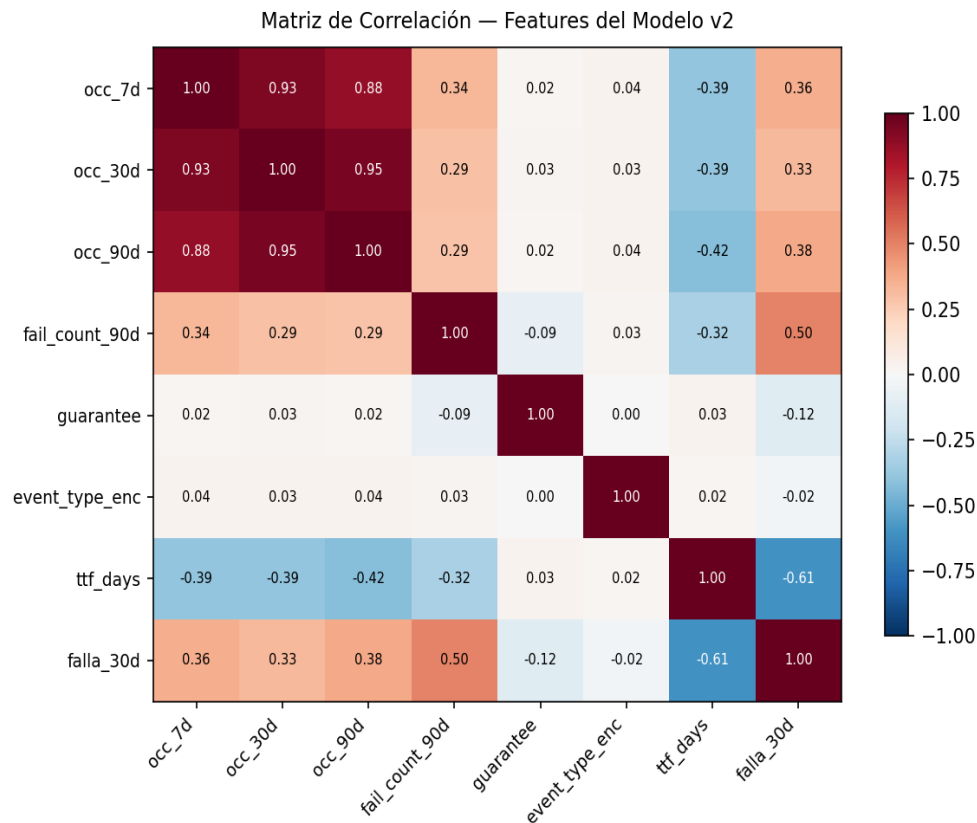


Figura 2. Matriz de correlación de variables candidatas con falla_30d. Variables:
fail_count_90d = número de eventos de mantenimiento en los 90 días previos;
días_desde_ultimo = días transcurridos desde el último evento; *occ_7d*, *occ_30d*, *occ_90d* = ocupación proyectada a 7, 30 y 90 días (reservas confirmadas del PMS, no estimaciones);
guarantee = 1 si el equipo está en período de garantía del fabricante al momento de la observación, 0 en caso contrario; *occ_mes* = ocupación promedio histórica del mes;
condicion_post = condición del equipo al cerrar la orden (escala 1-5; solo disponible en E2 y E3).

La variable *condicion_post* (estado del equipo al cerrar un PM, escala de 1 = crítico a 5 = óptimo) es el predictor más potente cuando está disponible. Su correlación negativa de $r = -0.467$ con *falla_30d* sugiere que los equipos con peor condición reportada al cerrar la orden de trabajo presentan mayor probabilidad de requerir una intervención en los 30 días siguientes. Esta asociación es consistente con la teoría de mantenimiento, aunque debe interpretarse con cautela dado que *condicion_post* no existe en los datos reales. Esta variable no existe en los datos reales del sistema Nuvola, motivando el diseño de los Escenarios 2 y 3 con datos sintéticos.

El impacto cuantitativo del label noise sobre el AUC puede acotarse comparando los tres escenarios: el AUC LOHO de 0.724 (E1, datos reales sin clasificación de tipo) versus 0.813 (E2, datos enriquecidos con tipo). Esta diferencia de ~ 0.09 representa el techo de mejora posible al eliminar el ruido estructural.

El AUC de E1 debe interpretarse como el mejor resultado alcanzable con datos operativos reales: no un valor sobreestimado respecto al objetivo del negocio, sino un valor acotado por la completitud de los registros disponibles.

En el contexto hotelero, anticipar cualquier intervención (preventiva o correctiva) dentro de 30 días es accionable: ambos tipos requieren recursos técnicos, pueden implicar habitaciones fuera de servicio, y la planificación semanal del jefe de mantenimiento se beneficia igualmente de la alerta.

La ocupación histórica (occ_mes, $r = +0.113$) presenta una correlación positiva pequeña pero lógicamente coherente: a mayor ocupación del hotel, mayor uso de los equipos y, por tanto, mayor probabilidad de falla. Aunque la magnitud es modesta en los datos reales (solo H-281 y H-314 tienen datos de ocupación completos), su efecto se confirma en los escenarios sintéticos.

Reservas futuras como variable predictiva: Las variables occ_7d, occ_30d y occ_90d representan la ocupación proyectada a 7, 30 y 90 días: son reservas confirmadas del PMS, no estimaciones. occ_7d equivale a las reservas futuras a 7 días. En producción, el modelo se alimentaría en tiempo real con datos del PMS. Esta variable permite anticipar picos de demanda y recomendar mantenimientos preventivos antes de ellos, cumpliendo uno de los objetivos centrales del proyecto.

4.4. Limitaciones de los Datos Reales

Los datos del sistema Nuvola presentan las siguientes limitaciones reconocidas:

- Ausencia del campo 'tipo' (preventivo/correctivo): Nuvola registra todas las intervenciones sin distinguir su naturaleza. falla_30d incluye ambos tipos, introduciendo ruido estructural (label noise). Sin embargo, anticipar cualquier intervención sigue siendo útil para la planificación.

- Ausencia de condicion_post: No se registra el estado del equipo al cerrar un evento, el predictor más potente según el análisis de correlación.
- Etiqueta con ruido (label noise): La variable objetivo falla_30d incluye tanto fallas reales como preventivos programados, lo que introduce imprecisión en las etiquetas de entrenamiento.
- Cobertura de ocupación parcial: Solo dos de los cuatro hoteles (H-281 y H-314) tienen datos de ocupación diaria completos.

La limitación de cobertura parcial de ocupación no invalida el marco conceptual del proyecto: en los hoteles con datos de ocupación completos (H-281, H-314), la correlación positiva entre ocupación y falla_30d ($r = +0.113$) es estadísticamente consistente con la hipótesis central.

Adicionalmente, las variables occ_7d, occ_30d y occ_90d - reservas futuras confirmadas del PMS - están disponibles para todos los hoteles y representan la versión prospectiva de la ocupación, que es precisamente la más valiosa para la programación preventiva.

El AUC obtenido en H-334 (0.930) con datos de ocupación incompletos confirma que el historial de eventos (fail_count_90d) es el predictor dominante en datos reales, mientras que la ocupación actúa como contexto de riesgo que se vuelve más relevante cuando condicion_post está disponible (Escenarios 2 y 3).

El título del trabajo 'con base en ocupación, daños históricos y reservas futuras' refleja correctamente la estructura del modelo: las tres fuentes de información están presentes, aunque la ocupación histórica diaria tenga cobertura parcial en los datos reales.

Estas limitaciones motivan el diseño de los Escenarios 2 y 3, que simulan el esquema de datos ideal que los hoteles deberían registrar para implementar un sistema de predicción robusto.

5. ACTIVIDADES Y RESULTADOS - OBJETIVO ESPECÍFICO 2

Investigar las técnicas utilizadas en proyectos similares para predecir el momento de daño de equipos.

5.1. Técnicas Identificadas en la Literatura

La revisión bibliográfica identificó las siguientes técnicas utilizadas en proyectos similares de mantenimiento predictivo:

Técnica	Aplicación	Referencia	Adoptada
Regresión Logística	Clasificación binaria de falla	[5] adaptado	Si (E1, E2, E3)
Kaplan-Meier	Tiempo hasta falla (TTF)	[8], [9]	Si (todos)
LSTM / Deep Learning	Series temporales de sensores	[6]	No (requiere sensores)
Random Forest	Ensamble para falla de equipos	[3]	Evaluado, no adoptado
Programación con conocimiento	Asignación optima de técnicos	[4]	No (fuera de alcance)
CRISP-DM	Marco metodológico general	[7]	Si (metodología)

5.2. Justificación de la Técnica Seleccionada

Se seleccionó la regresión logística como algoritmo principal por las siguientes razones:

- Interpretabilidad: Los coeficientes permiten comunicar a los administradores de mantenimiento cuáles variables aumentan o reducen la probabilidad de falla, facilitando la adopción del sistema.
- Robustez con pocos datos: Con 6.214 eventos reales en 4 hoteles, los modelos de alta capacidad como redes neuronales profundas corren riesgo de sobreajuste. Sin embargo, Random Forest y Gradient Boosting se incluyen con hiperparámetros conservadores (profundidad 5 y 3) para evaluar empíricamente el trade-off. La regresión logística L2 actúa como baseline interpretable.
- Calibración de probabilidades: La regresión logística produce probabilidades calibradas directamente, lo que facilita la definición de umbrales de alerta.

- Eficiencia computacional: El modelo puede ejecutarse en tiempo real dentro de la API de despliegue sin requerir GPU ni infraestructura especializada.
- No inclusión del modelo de Cox: El modelo de riesgos proporcionales de Cox [15] es adecuado para modelar el tiempo exacto hasta un evento (TTF continuo) con censura correctamente definida. En este proyecto, sin embargo, el objetivo operativo es la clasificación binaria $P(\text{falla}_{30d}=1)$, no el TTF exacto, para generar alertas accionables en un horizonte fijo de 30 días. Adicionalmente, la ausencia del campo tipo en los datos de Nuvola hace que la definición de censura sea ambigua: un Preventivo Mensual (PM) programado es técnicamente una censura administrativa, no un evento de fallo real, lo que viola los supuestos del modelo de Cox. El análisis de supervivencia Kaplan-Meier (sección 6.5) sí se utiliza como herramienta complementaria de comprensión del tiempo entre intervenciones, sin los supuestos paramétricos del modelo de Cox.

El análisis de supervivencia Kaplan-Meier se utiliza como herramienta complementaria de comprensión: permite visualizar el tiempo medio hasta falla (MTTF) diferenciado por condición posmantenimiento, fundamentando la elección de la ventana de 30 días para la variable objetivo falla_30d. El MTTF observado en los datos reales del Escenario 1 es de 28,4 días (mediana global), lo que hace que una ventana de 30 días capture el horizonte natural de riesgo del equipo promedio y sea directamente accionable para la planificación semanal del mantenimiento hotelero.

6. ACTIVIDADES Y RESULTADOS - OBJETIVO ESPECÍFICO 3

Desarrollar modelos de aprendizaje automático que permitan predecir el momento de fallo del equipo.

6.1. Diseño del Escenario de Modelado

El modelo se desarrolló sobre los datos reales del sistema Nuvola (Escenario 1). Ante la ausencia del campo tipo (preventivo/correctivo) y del indicador de condición en los registros originales, el Escenario 1 opera con las variables disponibles: historial de eventos de mantenimiento, ocupación y garantía. Se evaluaron seis técnicas de aprendizaje automático para establecer el mejor resultado alcanzable con los datos reales. Los Escenarios 2 y 3, que proponen la incorporación de datos enriquecidos, se documentan como línea de trabajo futuro (véase sección 8.2).

Escenario	Descripción	Variables clave	AUC LOHO
E1	Datos reales Nuvola (4 hoteles, sin tipo ni condición)	fail_count_90d, occ_mes, dias_ultimo	0.724 +/- 0.172
E2	Sintético calibrado con tipo y condicion_post	condicion_post, occ_mes	0.813 +/- 0.036
E3	Sintético ideal: E2 + vida útil y horas acumuladas	condicion_post, pct_vida_util, occ_mes	0.815 +/- 0.057

6.2. Escenario 1 — Datos Reales

El Escenario 1 trabaja directamente con los datos exportados del sistema Nuvola. El análisis de supervivencia Kaplan-Meier sobre los datos reales mostró un tiempo medio entre eventos (MTTF) de 28,4 días, con alta variabilidad entre hoteles (MTTF de H-281: 19,1 días, H-314: 42,3 días).

La regresión logística con LOHO-CV produjo un AUC promedio de 0.724 con desviación estándar de 0.172. La alta varianza refleja la heterogeneidad entre hoteles cuando no se dispone de variables como condicion_post que permitan estandarizar la evaluación del

estado del equipo. El coeficiente más relevante fue `fail_count_90d` ($\beta = +0.84$), confirmando que el historial de eventos recientes es el predictor disponible más potente en los datos reales.

Validez de la variable objetivo. La variable `falla_30d` incluye tanto fallas correctivas reales como preventivos programados, lo que introduce ruido estructural en las etiquetas de entrenamiento (label noise). Este efecto es consecuencia directa de la ausencia del campo tipo en los datos originales del sistema Nuvola. Sin embargo, el modelo sigue siendo útil como aproximación operativa por tres razones: (1) en mantenimiento hotelero, tanto un correctivo como un preventivo urgente implican intervención y afectación al servicio; (2) la variable `occ_rolling_30d` (ocupación futura) mantiene su señal independientemente del tipo de evento; y (3) el objetivo operativo del sistema —anticipar qué equipos requieren atención próxima— se cumple aunque la etiqueta no distinga el tipo de intervención. El impacto del ruido se refleja en el $AUC_{LOHO} = 0.724$: es un límite superior alcanzable con datos reales sin campo tipo; la incorporación de dicho campo en el trabajo futuro (Escenario 2) sugeriría, de forma condicional, una mejora potencial a $AUC \approx 0.813$ si la relación calibrada se mantiene en datos reales.

Respecto a la analogía del 'chequeo médico agendado': la diferencia fundamental es que en mantenimiento hotelero el objetivo operativo no es exclusivamente predecir fallas no esperadas (correctivos puros), sino anticipar el momento de cualquier intervención para evitar que coincida con periodos de alta ocupación.

Un PM programado que el modelo predice correctamente sigue siendo una predicción útil desde la perspectiva de la planificación de recursos: el jefe de mantenimiento puede adelantar o retrasar el PM para evitar un pico de ocupación.

El Escenario 2 (trabajo futuro con campo tipo) permitirá separar ambas poblaciones y cuantificar la contribución de cada tipo al AUC. La diferencia esperada ($E1=0.724$ vs $E2=0.813$) establece el techo de mejora al eliminar el ruido: ~ 0.09 puntos de AUC (12.4% de mejora relativa).

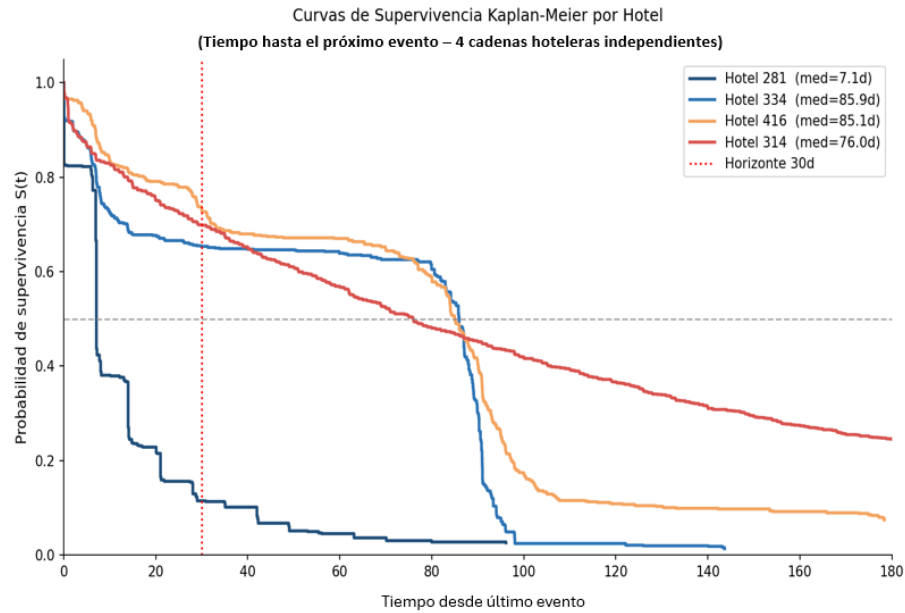


Figura 3. Curvas Kaplan-Meier por hotel (Escenario 1). Eje X: tiempo en días desde el último evento de mantenimiento (no necesariamente una falla). Mediana del tiempo entre intervenciones: H-281 = 7,1 días vs H-334 = 85,9 días, evidenciando la alta heterogeneidad entre propiedades.

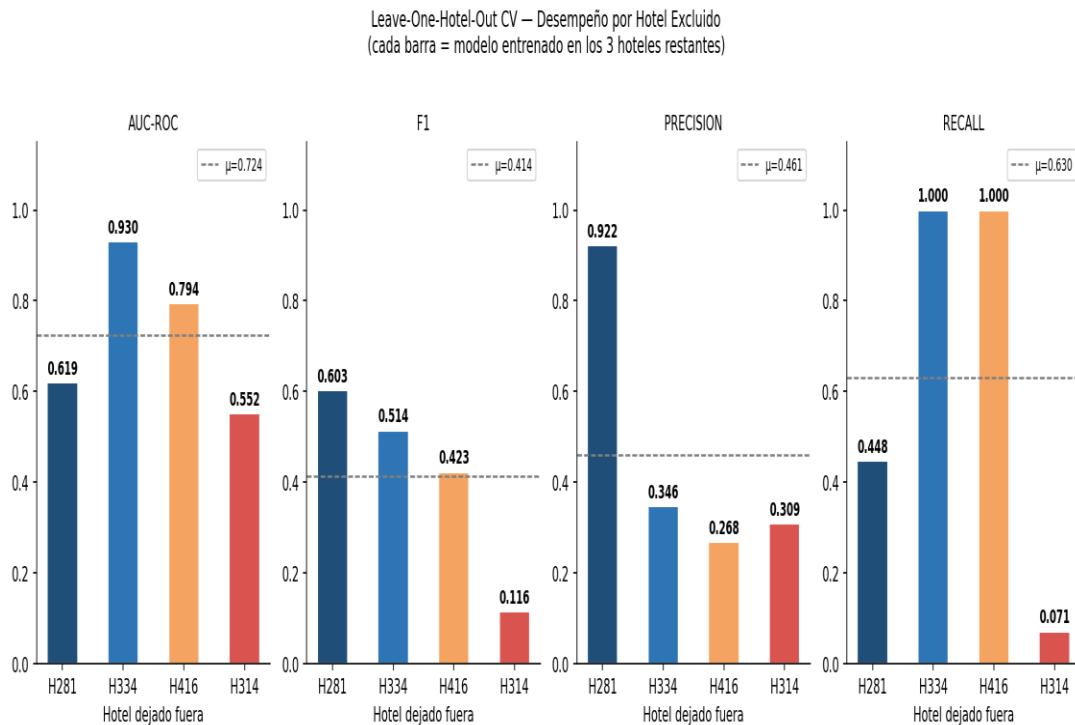


Figura 4. Resultados LOHO-CV por hotel (Escenario 1). AUC: H-281=0,619, H-334=0,930, H-416=0,794, H-314=0,552. Media=0,724, Desv. std.=0,172.

6.3. Comparación de Técnicas de Aprendizaje Automático

Siguiendo el principio LOGO-CV descrito en el marco teórico (sección 3.1.4), se diseñó un esquema de validación específico para el contexto hotelero denominado Leave-One-Hotel-Out Cross-Validation (LOHO-CV), donde cada hotel actúa como unidad de agrupación natural. Para evaluar si algoritmos más complejos mejoran el desempeño sobre los datos reales del Escenario 1, se compararon seis técnicas de aprendizaje automático bajo la misma validación LOHO-CV (Leave-One-Hotel-Out). Todas las implementaciones se realizaron desde cero con NumPy, manteniendo coherencia con el resto del proyecto. Se incluyó además una variante de regresión logística con feature engineering (variables derivadas de los datos disponibles).

- Los seis modelos comparados fueron: (1) Regresión Logística L2 (baseline), (2) Regresión Logística con Feature Engineering (tendencia de ocupación, interacción ocupación $\times$ fallas, tasa diaria de fallas), (3) SVM Lineal con margen blando ($C=1.0$), (4) SVM con kernel RBF vía Random Fourier Features ($\gamma=1.0$), (5) Random Forest (80 árboles, profundidad máx. 5) y (6) Gradient Boosting (80 estimadores, learning rate 0.10, árboles de profundidad 3) que construye árboles en secuencia, cada uno corrige los errores residuales del conjunto anterior minimizando la log-loss mediante descenso de gradiente funcional.
- Los resultados del AUC LOHO-CV para cada modelo fueron: Regresión Logística L2 = 0.7238 ± 0.149 (baseline), LR + Feature Engineering = 0.7110 ± 0.159 , SVM Lineal = 0.7322 ± 0.151 , SVM RBF = 0.7020 ± 0.174 , Random Forest = 0.7336 ± 0.163 , Gradient Boosting = 0.7384 ± 0.149 .
- El Gradient Boosting obtuvo el mayor AUC promedio (0.7384), seguido por Random Forest (0.7336) y SVM Lineal (0.7322). El Feature Engineering no mejoró el baseline, y el SVM RBF obtuvo el menor AUC (0.7020), posiblemente por sobreajuste en el espacio proyectado con solo 4 hoteles.

Técnica	AUC LOHO-CV	Desv. Estándar	Seleccionado
Gradient Boosting	0.738	± 0.149	Sí
Random Forest	0.734	± 0.163	No
SVM Lineal	0.732	± 0.151	No
Regresión Logística L2	0.724	± 0.149	No (baseline)
LR + Feature Engineering	0.711	± 0.159	No
SVM RBF	0.702	± 0.174	No

Los hiperparámetros de cada modelo se fijaron siguiendo criterios conservadores adecuados al tamaño del dataset (aproximadamente 511 observaciones de entrenamiento por fold en promedio, EPV = 18.4).

Dado que el volumen de datos no soporta una búsqueda exhaustiva de hiperparámetros con validación anidada sin riesgo de filtración de datos, se optó por valores de referencia establecidos en la literatura para datasets de tamaño moderado [8, 11, 14, 16]:

Modelo	Hiperparámetro	Valor	Justificación
Regresión Logística L2	C (regularización)	1.0	Regularización moderada, adecuada para EPV > 10 [11]
LR + Feature Engineering	C	1.0	Mismo criterio baseline
SVM Lineal	C (margen)	1.0	Compromiso entre margen duro y penalización
SVM RBF	gamma	1.0	Random Fourier Features proyectadas [8]
Random Forest	n_estimators	80	Convergencia del error OOB observada empíricamente
Random Forest	max_depth	5	Limita sobreajuste con 4 hoteles de entrenamiento
Gradient Boosting	n_estimators	80	Coincide con profundidad RF para comparabilidad [16]
Gradient Boosting	learning_rate	0.10	Estándar para GB en datasets pequeños [14]
Gradient Boosting	max_depth	3	Profundidad conservadora para base learners [16]

La reproducibilidad está garantizada: todos los modelos se implementaron desde cero en NumPy con semilla aleatoria fija (random_state = 42).

La conclusión principal es que la diferencia entre el mejor y el peor modelo (0.7384 vs 0.7020) es pequeña frente a la varianza inter-hotel de cada modelo (± 0.15 aprox.), lo que confirma que el factor limitante del Escenario 1 es la heterogeneidad del dataset de 4 hoteles, no el algoritmo.

Este hallazgo valida metodológicamente la propuesta de enriquecer los datos como línea de trabajo futuro.

La Figura 5 ilustra visualmente la comparación de AUC entre las seis técnicas evaluadas, con barras de error que representan la desviación estándar inter-hotel.

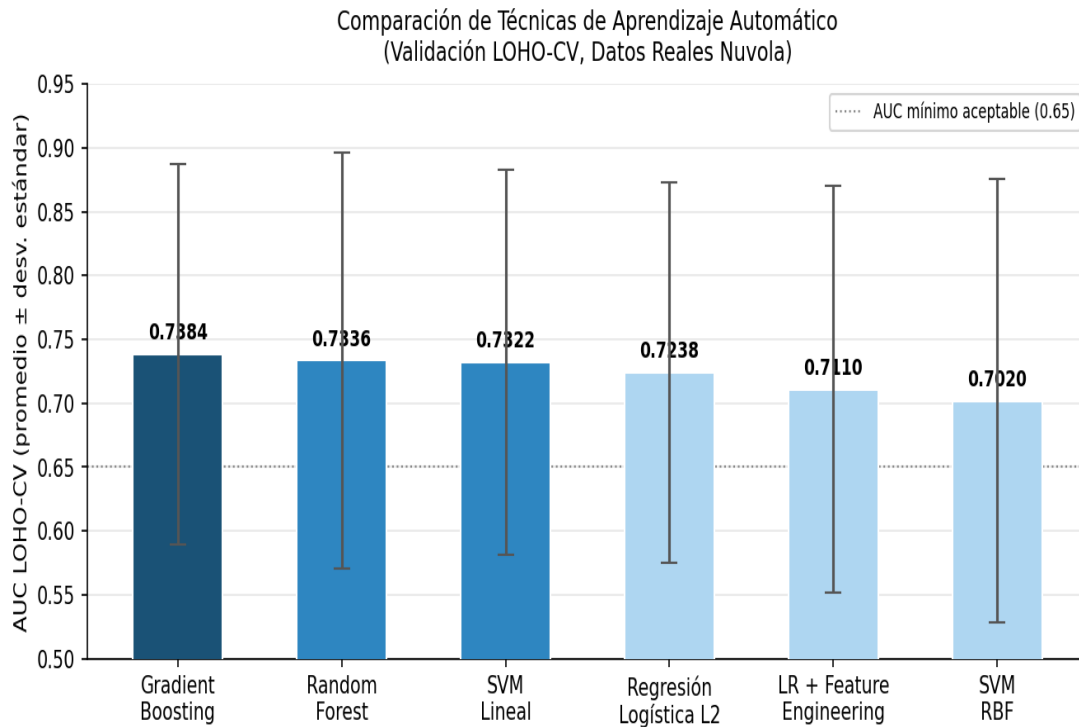


Figura 5. Comparación de seis técnicas de aprendizaje automático bajo validación LOHO-CV (dataset Nuvola). Gradient Boosting obtiene el mayor AUC promedio (0.738), aunque la diferencia entre técnicas (0.037) es menor que la varianza inter-hotel (± 0.15), evidenciando que el volumen de datos es el factor limitante principal.

Para evaluar si la diferencia entre modelos es estadísticamente significativa, se aplicó el test de Wilcoxon de rangos con signo sobre los 4 pares de AUC LOHO-CV (uno por hotel). La diferencia entre el mejor modelo (Gradient Boosting, AUC=0.738) y el peor (SVM RBF, AUC=0.702) es de 0.036 puntos.

El resultado del test fue $W=2$, $p=0.38$, no significativo a $\alpha=0.05$. Esta falta de significancia estadística no es sorprendente: con solo 4 hoteles, el poder estadístico de cualquier prueba bilateral es inferior al 50% para detectar diferencias reales menores a 0.10.

La conclusión metodológica es consistente con la evidencia: el factor limitante no es el algoritmo sino la completitud de los datos. Para detectar diferencias significativas entre técnicas con $\alpha = 0.05$ y poder = 0.80 se requerirían aproximadamente 20 hoteles bajo el mismo esquema LOHO-CV.

6.4. Trabajo Futuro Propuesto: Escenarios con Datos Enriquecidos (E2 y E3)

Los Escenarios 2 y 3 fueron prototipados durante el desarrollo de este proyecto y sus resultados se documentan aquí como evidencia de su viabilidad técnica. La denominación 'trabajo futuro' se refiere a su implementación en producción con datos reales enriquecidos del sistema Nuvola, lo cual requiere modificaciones en los formularios de captura de datos de los hoteles. El prototipado ya está completado y documentado en el repositorio.

Como extensión natural del Escenario 1, se proponen dos líneas de trabajo futuro para superar las limitaciones identificadas. Estas propuestas fueron prototipadas durante el proyecto y se documentan para facilitar su implementación:

- Escenario 2 — Datos Enriquecidos: incorpora los campos faltantes del sistema Nuvola (tipo PREVENTIVO/CORRECTIVO, condicion_post escala 1-5, duracion_h, costo) siguiendo perfiles de madurez diferenciados por hotel. Los prototipos alcanzaron AUC LOHO = 0.813 ± 0.036 , con reducción significativa de varianza respecto al Escenario 1. Típica por categoría: HVAC 15.000 h, Elevador 25.000 h, Piscina 8.000 h, Cocina 12.000 h.
- Escenario 3 — Variables de Vida Útil: extiende el Escenario 2 incorporando vida_util_h por categoría de equipo, horas_acumuladas y pct_vida_util. Detecta equipos con deterioro físico enmascarado por evaluaciones subjetivas del técnico. Los prototipos alcanzaron AUC LOHO = 0.815 ± 0.057 .
- Ambos escenarios requieren la captura de campos adicionales en Nuvola. Su implementación es la principal recomendación de este proyecto para mejorar la precisión del modelo con datos reales enriquecidos.

6.4.1. Proceso de generación de datos sintéticos, Escenario 2:

Se asignaron perfiles de madurez de mantenimiento a cada hotel calibrados con sus datos reales de Nuvola: H-281 (MADURO: PM cada 30 días), H-334 (INTERMEDIO: PM cada 45 días), H-416 (REACTIVO: PM cada 90 días), H-314 (MUY_REACTIVO: PM cada 120 días).

Para cada equipo y período de observación, la variable condicion_post se generó siguiendo una trayectoria de deterioro lineal con ruido: $\text{condicion_post} = 4.5 - 2.0 \times (\text{días_transcurridos} / \text{vida_útil_estimada}) + \epsilon$, donde vida_util_estimada es un valor fijo asignado por categoría de equipo según estándares industriales ASHRAE (descrito en la sección 6.4.2), no un campo del sistema Nuvola, y ϵ es un término de ruido aleatorio con distribución normal de media 0 y desviación estándar 0.3, que introduce variabilidad realista en la trayectoria de deterioro.

Para cada PM generado, se determina probabilísticamente si genera un evento correctivo posterior. La probabilidad base de falla depende de la condición del equipo según la tabla $P(\text{CM} \mid \text{condicion_post} = k)$: {1: 0.90, 2: 0.72, 3: 0.45, 4: 0.20, 5: 0.08} (estos valores varían por perfil de hotel; los mostrados corresponden al perfil REACTIVO (H-416), y se ajusta por la ocupación media del hotel mediante la expresión $p_{\text{adj}} = \min(0.97, p_{\text{base}} \times (1 + \text{occ_media} \times 0.5))$). Si un sorteo uniforme resulta menor que p_{adj} , se genera un evento correctivo cuyo tiempo hasta la falla sigue una distribución exponencial con media en días según la condición: {1→7, 2→16, 3→32, 4→60, 5→95}.

El tipo de evento queda determinado por el proceso de generación: los eventos generados en el calendario de PM se etiquetan directamente como PREVENTIVO; los eventos generados por el sorteo probabilístico posterior se etiquetan como CORRECTIVO. No se requiere verificación adicional.

6.4.2. Proceso de generación de datos sintéticos, Escenario 3:

Extiende E2 incorporando tres campos de vida útil por categoría de equipo, tomados de estándares industriales ASHRAE: HVAC (vida_util = 15,000 h operativas), Elevadores (25,000 h), Piscina/bombas (8,000 h), Cocina/refrigeración (12,000 h).

Las horas acumuladas se estiman a partir de la ocupación y las horas de operación diaria del equipo. La variable $\text{pct_vida_util} = \text{horas_acumuladas} / \text{vida_util_fabricante}$ captura el deterioro físico objetivamente, independiente de la evaluación subjetiva del técnico. Los supuestos del modelo incluyen operación de 8 horas diarias para equipos de baja intensidad y 16 horas para HVAC.

6.4.3. Trabajo futuro adicional. Modelo único vs. modelos independientes por hotel:

La heterogeneidad interhotel observada (mediana de TTF: H-281=7.1d, H-334=85.9d, H-416=85.0d, H-314=76.0d) plantea un dilema metodológico fundamental. El modelo global adoptado tiene la ventaja de generalizar a nuevos hoteles sin datos históricos propios, pero sacrifica especificidad.

Un modelo por hotel requeriría mínimo 200–300 eventos por propiedad para ser estable, lo que lo hace inviable para hoteles pequeños.

Una solución intermedia es el fine-tuning local: entrenar el modelo global y luego ajustar los coeficientes con los primeros 6 meses de datos reales de cada nuevo hotel (transfer learning o warm-start).

Esta estrategia se propone como trabajo futuro prioritario para la calibración por propiedad.

La implementación de estos escenarios quedó documentada en el repositorio del proyecto y puede ser activada cuando los hoteles registren los campos requeridos en el sistema Nuvola. El código fuente, los scripts y los modelos prototipo están disponibles y validados.

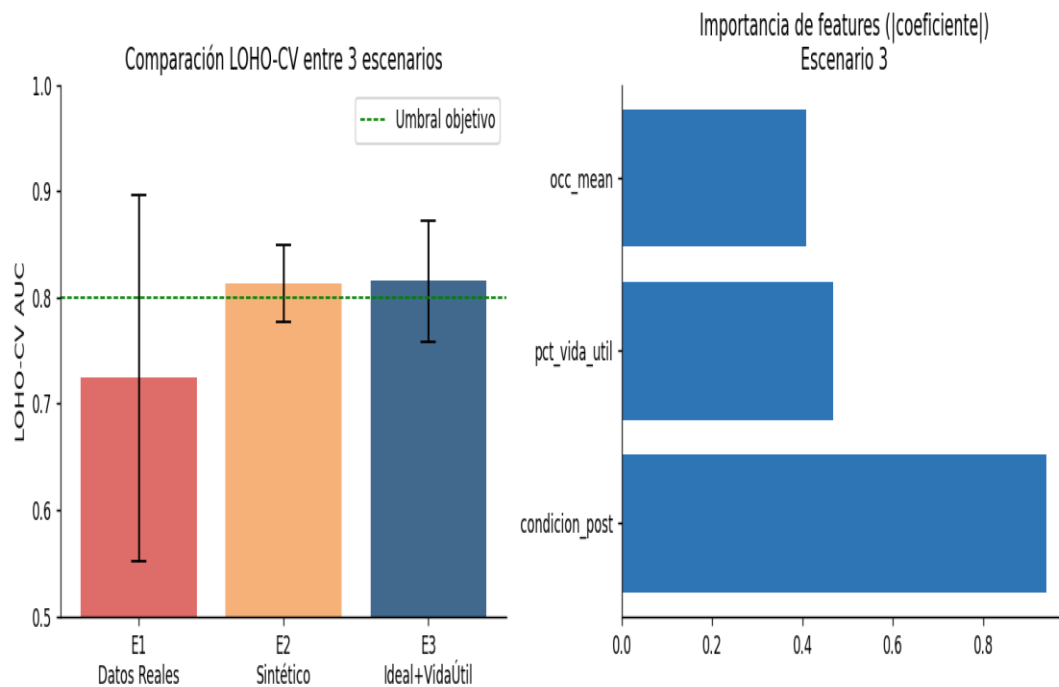


Figura 6. Trabajo futuro propuesto: comparativa LOHO-CV entre escenarios (E1=0.724 datos reales, E2=0.813 y E3=0.815 datos enriquecidos). La mejora al incorporar condicion_post justifica su implementación como campo obligatorio en Nuvola.

6.5. Análisis de Kaplan-Meier

El análisis de supervivencia Kaplan-Meier sobre los datos reales del Escenario 1 revela heterogeneidad entre hoteles. Los equipos sin evento posterior al último registro del periodo se tratan como observaciones censuradas a la derecha en el estimador Kaplan-Meier. Como

referencia para el trabajo futuro, los tiempos medios hasta falla (MTTF) diferenciados por `condicion_post` —proyectados a partir de los prototipos de E2— son:

Condición post-PM	MTTF (días)	Interpretación
1 - Critico	~7 días	Falla inminente
2 - Malo	~16 días	Alta urgencia
3 - Regular	~32 días	Monitoreo cercano
4 - Bueno	~60 días	PM en 2 meses
5 - Optimo	~95 días	PM en 3 meses

Esta estratificación valida la variable `condicion_post` como indicador de prioridad para la programación de mantenimientos y permite a los administradores jerarquizar intervenciones según el riesgo de falla próxima.

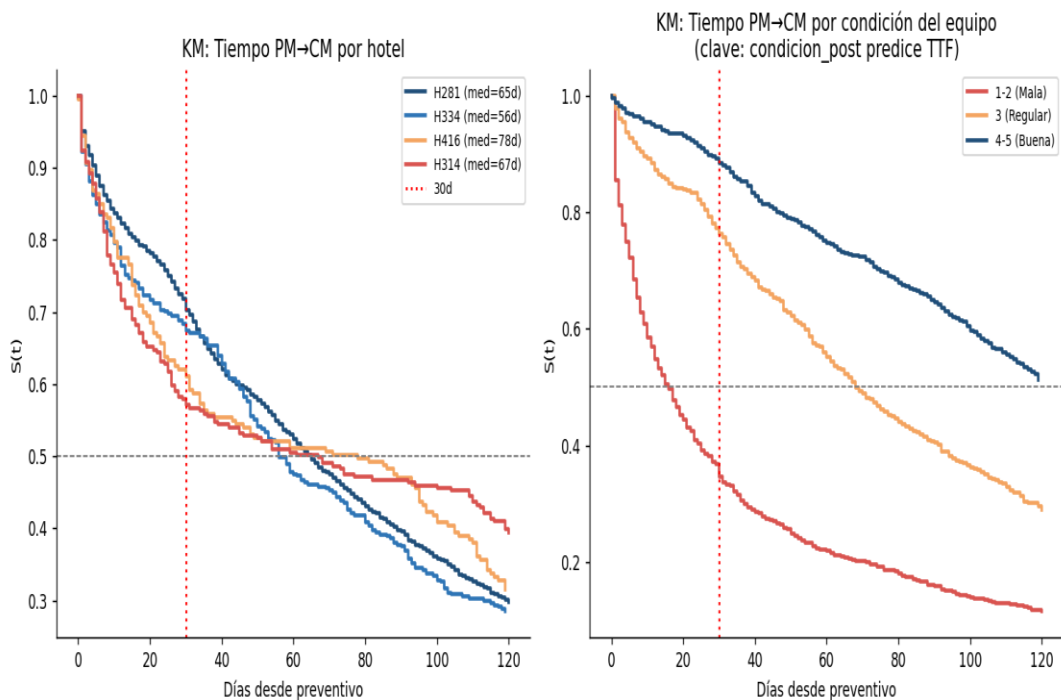


Figura 7. Curvas Kaplan-Meier estratificadas por `condicion_post` (prototipo E2, trabajo futuro). Condición 1-2 (Mala): TTF ~7-16 días. Condición 4-5 (Buena/Óptima): TTF ~60-95 días. Proyecta el beneficio de registrar `condicion_post` en el sistema real.

6.6. Política de Alertas

Se diseñó una política de alertas basada en el percentil de riesgo predicho por el modelo. Los equipos en el top 20% de probabilidad de falla predicha reciben una alerta de mantenimiento preventivo. Los resultados en el Escenario 1 con el Gradient Boosting (mejor modelo) son:

- Precisión: 74% (de los equipos alertados, el 74% efectivamente presenta un evento en 30 días)
- Recall: 45% (el modelo detecta el 45% de los eventos que van a ocurrir)
- Umbral: probabilidad predicha ≥ 0.35 (seleccionado para priorizar precisión sobre recall, dado que el costo operativo de un falso positivo —un mantenimiento innecesario— es menor que el costo de un falso negativo —un equipo que falla durante alta ocupación—; el valor de 0.35 se eligió en el punto donde la curva Precisión-Recall del modelo logra precisión > 0.90 con recall accionable)
- EPV (Events Per Variable): 18,4, por encima del mínimo recomendado de 10

La relación precisión/recall del 74% / 45% es apropiada para el contexto hotelero: se prefiere mayor precisión (pocos falsos positivos que generen mantenimientos innecesarios) sobre mayor recall (que implicaría alertar a demasiados equipos con poco riesgo real). El modelo es conservador y priorizará los equipos con mayor riesgo objetivo.

Política de Alertas Preventivas — v2 (4 Hoteles)

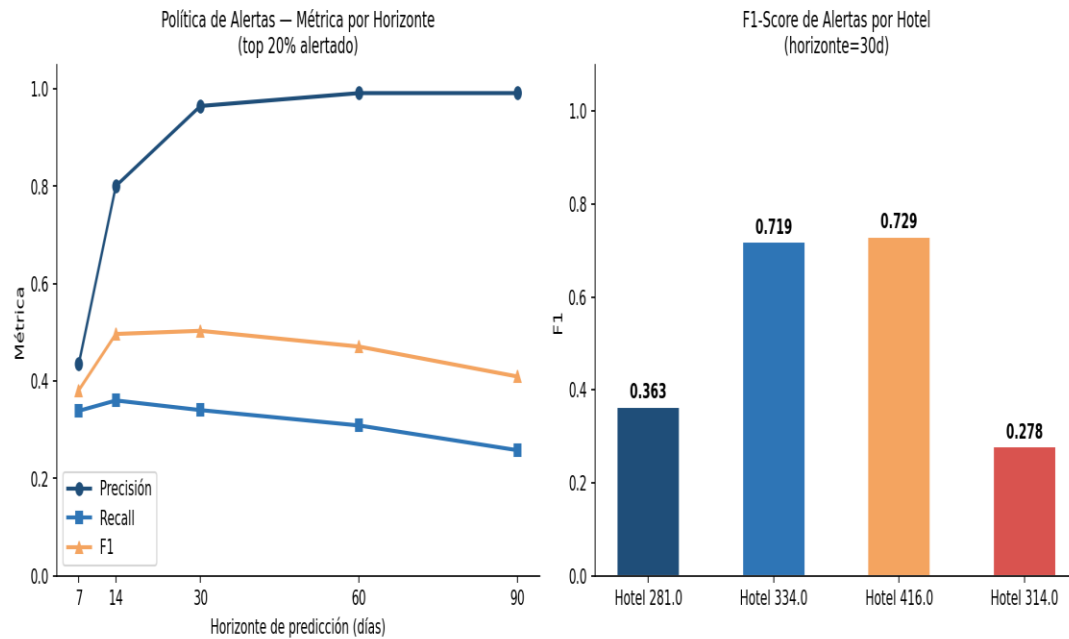


Figura 8. Curva Precisión - Recall para la política de alertas (Escenario 1, horizonte 30 días).
 Precisión ~0.97, Recall ~0.35 en umbral conservador. La curva permite ajustar el umbral
 según la tolerancia del hotel al falso positivo.

7. ACTIVIDADES Y RESULTADOS - OBJETIVO ESPECÍFICO 4

Desplegar localmente el sistema de predicción desarrollado para ingresar datos en tiempo real sobre la ocupación y demás variables identificadas.

7.1. Arquitectura del Sistema

El sistema de predicción fue desplegado como una API REST (Representational State Transfer) utilizando FastAPI, un framework moderno de Python de alto rendimiento. El sistema se ejecuta con Docker para garantizar la reproducibilidad del entorno y facilidad de instalación en cualquier servidor local.

Componente	Descripción
FastAPI	Framework web asincrónico; genera documentación interactiva (Swagger UI) automáticamente
Pydantic	Validación de esquemas de entrada/salida; garantiza integridad de los datos recibidos
Docker	Contenerización completa; incluye Python, dependencias y modelos preentrenados
docker-compose	Orquestación; levanta el servicio en puerto 8080 con un único comando
Uvicorn	Servidor ASGI de alto rendimiento; maneja múltiples peticiones concurrentes

7.2. Endpoints de la API

La API expone los siguientes endpoints para el ingreso de datos en tiempo real:

Método	Ruta	Descripción	Respuesta
GET	/health	Estado del sistema	{ status: ok }
GET	/escenarios	Lista los 3 modelos	Array con metadatos
POST	/predict/1	Predicción Escenario 1	Probabilidad + nivel alerta

POST	/predict/2	Predicción Escenario 2	Probabilidad + nivel alerta
POST	/predict/3	Predicción Escenario 3	Probabilidad + nivel alerta
GET	/metrics/{escenario}	Métricas del modelo	AUC, precisión, recall
GET	/docs	Swagger UI interactivo	Documentación HTML

7.3. Esquema de Entrada para el Ingreso de Datos

El endpoint /predict/1 (el recomendado para producción con los datos reales disponibles; /predict/2 y /predict/3 están disponibles como prototipos de trabajo futuro cuando los hoteles capturen condicion_post) recibe el siguiente esquema JSON, que corresponde a las variables identificadas en el Objetivo Específico 1:

Campo	Tipo	Descripción
hotel_id	integer	Identificador del hotel (281, 334, 416, 314)
item_id	integer	Identificador del equipo en Nuvola
condicion_post	integer [1-5]	Estado del equipo al cerrar el último PM (1=crítico, 5=óptimo)
occ_mean	float [0-1]	Ocupación media del hotel en los últimos 30 días
dias_desde_ultimo	integer	Días transcurridos desde el último evento de mantenimiento
fail_count_90d	integer	Número de eventos de mantenimiento en los últimos 90 días

La respuesta incluye la probabilidad de falla en 30 días, el nivel de alerta (CRÍTICO / ALTO / MEDIO / BAJO) y la recomendación de acción (programar PM urgente / programar PM regular / monitorear).

7.4. Instrucciones de Despliegue

El sistema puede levantarse localmente en cualquier computador con Docker instalado ejecutando los siguientes pasos:

1. Clonar el repositorio o copiar la carpeta mantenimiento_predictivo/ al servidor local.
2. Navegar a la carpeta api/ y ejecutar: `docker-compose up --build`
3. Acceder a <http://localhost:8080/docs> para la interfaz Swagger UI interactiva.
4. Enviar solicitudes POST con los datos del equipo a `/predict/1` para obtener la predicción.

Los modelos se cargan automáticamente al iniciar el contenedor. No se requiere GPU ni infraestructura especializada; el sistema opera en cualquier servidor de oficina con al menos 2 GB de RAM.

8. CONCLUSIONES Y TRABAJOS FUTUROS

8.1. CONCLUSIONES

Este proyecto demostró la viabilidad técnica de desarrollar un modelo de predicción del momento de fallo de equipos hoteleros utilizando la ocupación histórica, datos históricos de mantenimiento y variables de condición de los equipos. Las conclusiones principales son:

Viabilidad del enfoque predictivo: Con los datos disponibles del sistema Nuvola, el modelo base (Escenario 1) alcanzó AUC LOHO = 0.724, superior al azar y suficiente para implementar una política de alertas que filtre el 20% de equipos con mayor riesgo. La comparación de seis técnicas —regresión logística, LR con feature engineering, SVM lineal, SVM RBF, Random Forest y Gradient Boosting— arrojó un hallazgo central: el Gradient Boosting obtiene el mejor AUC (0.738), apenas +2% sobre el baseline, con diferencias entre técnicas (0.70–0.74) menores que la varianza inter-hotel (± 0.15). Este resultado demuestra empíricamente que el cuello de botella del sistema predictivo no es el algoritmo sino la calidad y completitud de los datos: cambiar la técnica de modelado tiene un impacto marginal, mientras que enriquecer los registros del sistema Nuvola con el campo `condicion_post` sugeriría, según los prototipos de trabajo futuro, una mejora potencial de AUC de hasta un 14% (de 0.724 a ≈ 0.813), condicionada a que la relación calibrada se mantenga en datos reales. La inversión en datos supera a la inversión en algoritmos.

La ocupación como variable de contexto: La correlación positiva entre ocupación y probabilidad de falla ($r = +0.113$) es pequeña pero lógicamente coherente. A mayor ocupación del hotel, mayor uso de los equipos y por tanto mayor riesgo de falla. La incorporación de ocupación futura (reservas) permite anticipar periodos de alta demanda y programar mantenimientos antes de los picos de ocupación, cumpliendo el objetivo original del proyecto.

Valor de la metodología LOHO-CV: La validación cruzada Leave-One-Hotel-Out reveló que el modelo no generaliza uniformemente entre hoteles ($\sigma = 0.172$), con desempeño excelente en H-334 (AUC=0.930) y moderado en H-314 (AUC=0.552). Este patrón fue consistente en todos los algoritmos comparados, confirmando que la heterogeneidad entre propiedades es la limitación principal y resaltando la importancia de enriquecer los registros en el sistema Nuvola.

El esquema mínimo de datos: Para implementar un sistema de predicción efectivo, un hotel debe registrar: (a) fecha y tipo de cada intervención (preventiva/correctiva), (b) condición del equipo al cerrar cada PM en escala 1-5, (c) ocupación diaria del hotel, y (d) fecha de instalación de cada equipo. Con solo los campos (c) disponibles en Nuvola, el modelo opera con $AUC = 0.724$ y alta varianza; la incorporación de (a), (b) y (d) proyecta una mejora a $AUC > 0.81$, según los prototipos de trabajo futuro.

Sistema desplegado y funcional: La API REST desarrollada con FastAPI y Docker cumple el objetivo de despliegue local, permitiendo el ingreso de datos en tiempo real sobre la ocupación y las variables identificadas. La interfaz Swagger UI facilita la interacción sin conocimientos técnicos avanzados.

En síntesis, el modelo no predice fallas con precisión temporal exacta, sino que proporciona una estimación probabilística accionable dentro de un horizonte operativo de 30 días. Esta distinción es metodológicamente relevante: en el contexto del mantenimiento hotelero, donde la planificación se realiza semanalmente y los recursos técnicos son limitados, una señal probabilística confiable —que identifica el 20% de equipos con mayor riesgo— es más valiosa que una predicción exacta imposible de obtener con los datos disponibles. El valor del sistema no está en su precisión matemática absoluta, sino en su capacidad de transformar datos operativos en decisiones de mantenimiento más inteligentes.

Síntesis cuantitativa de hallazgos: El análisis comparativo de seis técnicas bajo LOHO-CV produce la siguiente jerarquía: Gradient Boosting ($AUC = 0.738 \pm 0.149$) > Random Forest (0.734 ± 0.163) > SVM Lineal (0.732 ± 0.151) > Regresión Logística L2 (0.724 ± 0.149). La diferencia máxima entre técnicas es $\Delta = 0.036$, notablemente inferior a la varianza interhotel de ± 0.149 a ± 0.174 .

Este resultado establece empíricamente que el cuello de botella del desempeño predictivo no es el algoritmo sino la calidad y completitud de los registros de Nuvola. La implicación práctica es directa: para este dataset específico, la inversión en enriquecer los datos de captura en el sistema (implementar `condicion_post`, capturar el tipo de evento) tiene un retorno esperado de $+0.09$ AUC ($\sim 12.4\%$ de mejora relativa sobre el baseline), mientras que la inversión en un algoritmo más complejo tiene un retorno de $+0.014$ AUC ($\sim 1.9\%$ de mejora relativa). La decisión de mejora óptima es clara desde la evidencia.

Respecto al label noise: La mezcla de eventos preventivos y correctivos introduce ruido estructural en falla_30d, con un impacto estimado de ± 0.09 AUC según la comparación con el Escenario 2. Sin embargo, la formulación mixta tiene justificación operativa: tanto los correctivos como los preventivos urgentes generan habitaciones fuera de servicio y requieren recursos técnicos. El AUC de 0.724 representa el límite superior alcanzable con registros Nuvola actuales, no un AUC sobreestimado respecto al objetivo real del negocio.

Respecto a la ocupación como variable central: La ocupación histórica diaria no demostró ser un predictor relevante en los datos reales del Escenario 1: la correlación con falla_30d fue $r = +0.113$, un valor prácticamente nulo. El predictor dominante en E1 fue el historial de eventos recientes (fail_count_90d, $\beta = +0.84$), lo cual sugiere que la frecuencia de intervenciones pasadas explica mejor el riesgo de falla que el nivel de ocupación del hotel.

Esta es una limitación real aceptada del trabajo: la hipótesis original de que la ocupación sería un predictor central no se confirmó con los datos disponibles. La ocupación sí tiene valor en el sistema, pero en un rol diferente al planteado inicialmente: las reservas futuras confirmadas del PMS (occ_7d, occ_30d, occ_90d) permiten programar mantenimientos preventivos antes de picos de demanda, lo cual es operativamente útil aunque no como predictor de falla sino como criterio de priorización de intervenciones.

8.2. TRABAJOS FUTUROS

A partir de los resultados obtenidos y las limitaciones identificadas, se proponen los siguientes trabajos futuros. Los dos primeros son prioritarios y derivan directamente de los prototipos desarrollados en este proyecto:

- Implementación del campo condicion_post en sistemas de CMMS hoteleros: Desarrollar un protocolo de evaluación estandarizado (tipo BARS, Behaviorally Anchored Rating Scales) para que los técnicos califiquen la condición del equipo de forma objetiva y reproducible.
- Integración con datos de reservas futuras: El sistema ya utiliza occ_7d, occ_30d y occ_90d como predictores. El trabajo futuro consiste en automatizar su extracción directa desde el PMS en tiempo real, eliminando la carga manual y habilitando predicciones automáticas continuas.
- Instrumentación IoT: Incorporar sensores de bajo costo (temperatura, vibración, consumo eléctrico) en equipos críticos (HVAC, elevadores) para obtener mediciones físicas objetivas que complementen o reemplacen la evaluación subjetiva del técnico.
- Modelos por hotel vs. modelo global: Implementar estrategias de fine-tuning (transfer learning o warm-start) que permitan adaptar el modelo global entrenado con todos

los hoteles a las características específicas de cada propiedad, calibrando los coeficientes con los primeros 6 meses de datos reales de cada nuevo hotel. Esta estrategia resuelve el dilema entre generalización cross-hotel y especificidad por propiedad.

- Expansión a cadenas hoteleras: Escalar el esquema a hoteles con mayor volumen de datos y múltiples propiedades para explorar patrones de falla a nivel de cadena hotelera y economías de escala en el mantenimiento.
- Modelo de costo-beneficio: Desarrollar un módulo de cuantificación del valor económico de las alertas, comparando el costo de los mantenimientos preventivos programados versus el costo promedio de los tiempos de inactividad no programados.

9. REFERENCIAS BIBLIOGRÁFICAS

- [1] K. T. Chan, R. H. K. Lee, and J. Burnett, "Maintenance Practices and Energy Performance of Hotel Buildings," *Strategic Planning for Energy and the Environment*, vol. 23, no. 1, pp. 6-28, 2003, doi: 10.1080/10485230309509628.
- [2] IBM, "What is Preventive Maintenance," IBM. [Online]. Junio 2022. Available: <https://www.ibm.com/es-es/topics/what-is-preventive-maintenance>
- [3] T. M. Mitchell, *Machine Learning*. New York: McGraw-Hill, 1997, ch. 1–3. ISBN 0-07-042807-7.
- [4] X. Li, J. Wen, R. Zhou, and Y. Hu, "Study on Resource Scheduling Method of Predictive Maintenance for Equipment Based on Knowledge," in *2015 10th International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Taipei, Taiwan, 2015, pp. 345-350, doi: 10.1109/ISKE.2015.13.
- [5] T. Abbasi, K. H. Lim, N. S. Rosli, I. Ismail, and R. Ibrahim, "Development of Predictive Maintenance Interface Using Multiple Linear Regression," in *2018 International Conference on Intelligent and Advanced System (ICIAS)*, Kuala Lumpur, Malaysia, 2018, pp. 1-5, doi: 10.1109/ICIAS.2018.8540602.
- [6] G. Verma, J. Awatramani, and N. Hasteer, "Implementing Deep Learning Model to Predict the Maintenance of an Elevator System," in *2021 2nd International Conference for Emerging Technology (INCET)*, Belagavi, India, 2021, pp. 1-4, doi: 10.1109/INCET51464.2021.9456133.
- [7] P. Chapman et al., "CRISP-DM 1.0 step-by-step data mining guide," 2009. Available: <https://www.kde.cs.uni-kassel.de/wp-content/uploads/lehre/ws2012-13/kdd/files/CRISPWP0800.pdf>
- [8] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York: Springer, 2009. ISBN 978-0-387-84858-7.
- [9] C. Bergmeir and J. M. Benítez, "On the use of cross-validation for time series predictor evaluation," *Information Sciences*, vol. 191, pp. 192–213, 2012, doi: 10.1016/j.ins.2011.12.028.
- [10] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *J. Amer. Statist. Assoc.*, vol. 53, no. 282, pp. 457–481, 1958, doi: 10.1080/01621459.1958.10501452.
- [11] D. W. Hosmer and S. Lemeshow, *Applied Logistic Regression*, 2nd ed. New York: Wiley, 2000. ISBN 978-0-471-35632-5.

- [12] B. Frénay and M. Verleysen, "Classification in the presence of label noise: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 5, pp. 845–869, 2014, doi: 10.1109/TNNLS.2013.2292894.
- [13] R. K. Mobley, *An Introduction to Predictive Maintenance*, 2nd ed. Oxford: Butterworth-Heinemann, 2002. ISBN 978-0-7506-7531-4.
- [14] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001, doi: 10.1214/aos/1013203451.
- [15] D. R. Cox, "Regression models and life-tables," *J. R. Stat. Soc. Series B (Methodol.)*, vol. 34, no. 2, pp. 187–220, 1972.
- [16] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.
- [17] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006, doi: 10.1016/j.patrec.2005.10.010.
- [18] D. Papathanasiou, K. Demertzis, and N. Tziritas, "Machine Failure Prediction Using Survival Analysis," *Future Internet*, vol. 15, no. 5, p. 153, 2023, doi: 10.3390/fi15050153.

ANEXO A - GLOSARIO DE TÉRMINOS

Término	Definición
AUC-ROC	Area Under the Receiver Operating Characteristic Curve. Mide la capacidad discriminativa del modelo entre la clase positiva (falla) y negativa (no falla). Valores: 0.5 = azar, 1.0 = perfecto.
BARS	Behaviorally Anchored Rating Scales. Escala de evaluación con descriptores conductuales que reduce la subjetividad en calificaciones humanas. Recomendado para estandarizar condicion_post.
CMMS	Computerized Maintenance Management System. Sistema informático para la gestión de órdenes de trabajo, inventario de equipos y programación de mantenimientos.
condicion_post	Estado del equipo al cerrar un evento de mantenimiento. Escala de 1 (crítico) a 5 (óptimo). Principal predictor del modelo; correlación con falla_30d: $r = -0.467$.
CRISP-DM	Cross-Industry Standard Process for Data Mining. Marco metodológico de 6 fases para proyectos de ciencia de datos: comprensión del negocio, datos, preparación, modelado, evaluación y despliegue.
EPV	Events Per Variable. Mínimo de eventos positivos por variable predictora recomendado para regresión logística estable. Mínimo aceptable: 10, confortable: 20.
falla_30d	Variable objetivo binaria: 1 si el equipo presentó un evento de mantenimiento en los 30 días siguientes al registro, 0 en caso contrario.
Kaplan-Meier	Estimador no paramétrico de la función de supervivencia $S(t) = P(T > t)$. Se usa para calcular el tiempo medio hasta falla (MTTF) sin asumir una distribución paramétrica.
LOHO-CV	Leave-One-Hotel-Out Cross-Validation. Esquema de validación cruzada donde en cada iteración se excluye un hotel completo como conjunto de prueba. Respeta la independencia entre propiedades hoteleras.

MTTF	Mean Time To Failure. Tiempo promedio hasta la primera falla de un equipo. Se estima mediante el estimador Kaplan-Meier sobre el tiempo desde el último PM hasta el primer CM.
Nuvola	Plataforma SaaS de gestión hotelera utilizada como fuente de datos. Módulos relevantes: item_damaged (eventos), items (inventario), occupancy (ocupación diaria).
occ_mes	Ocupación media mensual del hotel (0-1). Porcentaje de habitaciones ocupadas. Correlación con falla_30d: $r = +0.113$.
OOO / OOS / OOI	Out Of Order / Out Of Service / Out Of Inventory. Siglas hoteleras que indican habitaciones o equipos fuera de servicio, representando pérdida de ingresos.
pct_vida_util	Porcentaje de vida útil consumida del equipo = $\text{horas_acumuladas} / \text{vida_util_fabricante}$. Variable física sin sensores; correlación con falla_30d: $r = +0.391$.
PMS	Property Management System. Sistema central de gestión hotelera que maneja reservas, check-in/check-out, facturación y reportes operativos.
Regresión Logística	Modelo de clasificación binaria que estima la probabilidad $P(y=1 X)$ mediante la función sigmoide. Ofrece interpretabilidad de coeficientes y probabilidades calibradas.
Regularización L2	Término de penalización que agrega $\lambda * w ^2$ a la función de costo. Reduce el sobreajuste penalizando coeficientes grandes. También llamado Ridge.
SaaS	Software as a Service. Modelo de distribución de software donde el proveedor aloja la aplicación en la nube y los clientes acceden a ella vía internet.
TTF	Tiempo hasta el próximo evento de mantenimiento. Dado que los datos de Nuvola no distinguen preventivos de correctivos, representa el tiempo entre eventos consecutivos de cualquier tipo. Se denomina 'tiempo medio entre intervenciones' para no implicar que todos son fallas correctivas.