ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER EN BIG DATA: TECNOLOGÍA Y ANALÍTICA AVANZADA

Desarrollo de un modelo predictivo de costes y optimización de la cadena de suministro en el sector de la construcción mediante técnicas de Big Data

Autor: Miguel González Lavín

Director: Alfonso de Lucas David

Madrid

Junio 2026

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Desarrollo de un modelo predictivo de costes y optimización de la cadena de suministro en el sector de la construcción mediante técnicas de big data

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2025/26 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Miguel González Lavín Fecha: 07/06/2026

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Alfonso de Lucas David Fecha: 07/06/2026

Vº Bº del Coordinador de Proyectos

Fdo.: Carlos Morrás Ruiz-Falcó Fecha: ……/ ……/ ……

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. Miguel González Lavín DECLARA ser el titular de los derechos de propiedad intelectual de la obra: Desarrollo de un modelo predictivo de costes y optimización de la cadena de suministro en el sector de la construcción mediante técnicas de Big Data, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor **CEDE** a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

a) Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar "marcas de agua" o cualquier otro sistema de seguridad o de protección.
b) Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
c) Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
d) Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
e) Asignar por defecto a estos trabajos una licencia Creative Commons.
f) Asignar por defecto a estos trabajos un HANDLE (URL *persistente)*.

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

a) Que la Universidad identifique claramente su nombre como autor de la misma
b) Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
c) Solicitar la retirada de la obra del repositorio por causa justificada.
d) Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

a) Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
b) Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
c) Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
d) Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción

de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusive del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

Madrid, a 7 de junio de 2026

ACEPTA

Fdo……

Motivos para solicitar el acceso restringido, cerrado o embargado del trabajo en el Repositorio Institucional:

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MÁSTER EN BIG DATA: TECNOLOGÍA Y ANALÍTICA AVANZADA

Desarrollo de un modelo predictivo de costes y optimización de la cadena de suministro en el sector de la construcción mediante técnicas de Big Data

Autor: Miguel González Lavín

Director: Alfonso de Lucas David

Madrid

Junio 2026

Agradecimientos

A mis padres, por su apoyo incondicional y por enseñarme desde pequeño que el esfuerzo siempre tiene recompensa.

A mi hermana, por su ánimo constante, por estar presente en cada momento y por ser un referente para mí.

A mi abuela, por todo su cariño y por haberme cuidado siempre con tanta dedicación.

A mis amigos de la universidad, por los momentos vividos y por hacer más llevadera cada etapa del camino.

A todo el equipo de Arpada, en especial al departamento de tecnología, por la ayuda durante todo el proceso.

A mi director, Alfonso De Lucas David, por su guía, su implicación y su disposición a lo largo de todo el desarrollo de este trabajo.

DESARROLLO DE UN MODELO PREDICTIVO DE COSTES Y OPTIMIZACIÓN DE LA CADENA DE SUMINISTRO EN EL SECTOR DE LA CONSTRUCCIÓN MEDIANTE TÉCNICAS DE BIG DATA

Autor: González Lavín, Miguel.
Director: De Lucas David, Alfonso.
Entidad Colaboradora: Arpada S. A.

RESUMEN DEL PROYECTO

Este trabajo desarrolla un sistema de predicción de curvas de producción por código de compra para obras de construcción residencial. Se implementan y comparan seis modelos, desde una media ponderada histórica hasta redes LSTM, obteniendo con LightGBM y Random Forest una reducción del error del 77% frente al método experto de referencia (RMSE de 3.5 pp frente a 15.26 pp). Los resultados se integran en una interfaz web con Streamlit accesible para perfiles no técnicos.

Palabras clave: Curva S, Predicción de series temporales, Big Data, Construcción

1. Introducción

La gestión del aprovisionamiento de materiales en obras de construcción residencial es uno de los principales focos de ineficiencia operativa. Los departamentos de compras deben anticipar cuándo y cuánto se necesitará de cada partida presupuestaria (estructura, instalaciones, acabados) para evitar tanto los costes de almacenamiento por sobrecompra como los riesgos de parada de obra por falta de suministro. Esta planificación se realiza habitualmente de forma manual, apoyada en la experiencia acumulada de los técnicos, sin un modelo cuantitativo que formalice y explote el histórico de obras anteriores [1].

El presente trabajo aborda este problema aplicando técnicas de big data y machine learning a un catálogo de más de 60 obras residenciales [2]. El objetivo es desarrollar un sistema capaz de predecir la evolución de la curva de producción de cada código de compra a lo largo del tiempo de obra, desde los primeros estadios hasta su finalización, y de mejorar sustancialmente la precisión respecto al método experto actual.

2. Definición del proyecto

El punto de partida del trabajo es un dataset extraído de la base de datos SQL Server corporativa, compuesto por 69 obras residenciales, 156 códigos de compra con cobertura histórica suficiente y 6.973 series temporales de avance acumulado normalizadas al intervalo [0,1] en tiempo y producción. Cada serie representa la evolución mensual del porcentaje de avance de una partida presupuestaria concreta en una obra concreta.

El análisis exploratorio previo al modelado cumple un doble propósito. Por un lado, valida empíricamente la hipótesis de curva S [3] logística que fundamenta el diseño del modelo M6: el ajuste logístico de tres parámetros obtiene un R^2 superior a 0.9 en la mayoría de las series, confirmando con datos reales lo que hasta ahora se asumía como hipótesis teórica del sector [1]. Por otro lado, caracteriza la heterogeneidad del dataset: la cobertura histórica varía enormemente entre códigos, desde partidas con decenas de obras disponibles hasta códigos de uso puntual con apenas tres o cuatro registros, lo que

condiciona directamente qué modelos son viables en cada caso y cómo debe diseñarse el protocolo de evaluación.

El sistema completo abarca la extracción y normalización de los datos, el entrenamiento y evaluación de seis modelos de predicción con complejidad creciente, y la integración de los resultados en una capa de visualización accesible para el usuario final.

3. Descripción del modelo/sistema/herramienta

Se implementan seis modelos de predicción evaluados con leave-one-out consistente: M1, media ponderada histórica que actúa como baseline; M2, similitud por Dynamic Time Warping; M3, LightGBM con formulación punto a punto; M4, Random Forest con la misma formulación; M5, red LSTM encoder-decoder; y M6, ajuste logístico paramétrico.

Todos reciben como entrada las características estáticas de la obra y la porción de curva ya observada, y predicen los puntos futuros hasta el 100% del tiempo de la obra.

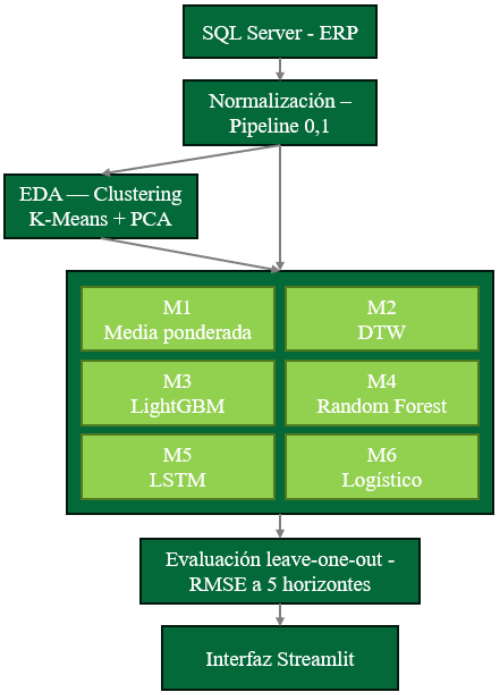


Ilustración 1 - Sistema desarrollado

4. Resultados

La evaluación se realiza a cinco horizontes de observación (10%, 20%, 30%, 50% y 70% del tiempo transcurrido), usando el RMSE en puntos porcentuales de avance acumulado como métrica principal. Los resultados se resumen en la tabla siguiente:

Modelo	*RMSE global*	*10%*	*20%*	*30%*	*50%*	*70%*
M1 Media Ponderada	15.26 pp	15.26	15.26	15.26	15.26	15.26
M2 DTW	8.59 pp	10.30	8.36	7.82	7.98	8.78
M3 LightGBM	3.52 pp	8.23	5.90	3.72	1.82	0.67
M4 Random Forest	3.46 pp	7.91	5.65	3.90	1.76	0.68
M5 LSTM	5.86 pp	10.99	8.39	6.22	3.15	1.47
M6 Logístico	12.72 pp	16.19	13.89	12.24	8.60	6.32

Ilustración 2 - Comparación RMSE modelos utilizados

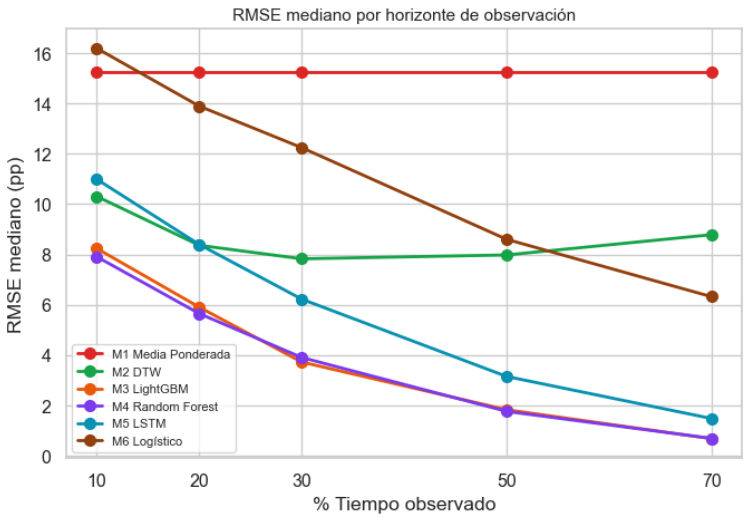


Ilustración 3 - Comparación del RMSE mediano de cada uno de los modelos

5. Conclusiones

El sistema desarrollado convierte el conocimiento tácito acumulado por el departamento de estudios en un modelo cuantitativo reproducible que mejora en un 77% la precisión del método experto de referencia [4]. La información más valiosa para la predicción no son las características estáticas de la obra sino la propia curva parcialmente observada, lo que justifica el diseño de modelos que la incorporen explícitamente. La arquitectura está diseñada para su migración al entorno productivo Databricks con Azure Data Lake Storage y su integración con los dashboards de Power BI existentes, con líneas de trabajo futuras que incluyen alertas automáticas de desviación, reentrenamiento periódico al incorporar nuevas obras finalizadas y la exploración de arquitecturas Transformer para el modelo secuencial.

6. Referencias

[1] Muszynska, K. (2020). The S-curve as a tool for planning and controlling of construction process — Case study. Applied Sciences, 10(6), 2071. https://doi.org/10.3390/app10062071.

[2] Gondia, A., Siam, A., El-Dakhakhni, W., & Nassar, A. H. (2020). Machine learning algorithms for construction projects delay risk prediction. Journal of Construction Engineering and Management, 146(1), 04019085. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001736.

[3] Forbes, T., & Riso, T. (2024). Guide to S-Curve Modeling in Construction. Procore. https://www.procore.com/library/s-curve-modeling-construction.

[4] Alizadeh, E. (2020, octubre 11). An illustrative introduction to dynamic time warping. Towards Data Science. https://towardsdatascience.com/an-illustrative-introduction-to-dynamic-time-warping-36aa98513b98/.

DEVELOPMENT OF A PREDICTIVE COST MODEL AND SUPPLY CHAIN OPTIMISATION IN THE CONSTRUCTION SECTOR USING BIG DATA TECHNIQUES

Author: González Lavín, Miguel.
Supervisor: De Lucas David, Alfonso.
Collaborating Entity: Arpada S. A.

ABSTRACT

This work develops a production curve prediction system by purchase code for residential construction projects. Six models are implemented and compared, ranging from a historical weighted average to LSTM networks, with LightGBM and Random Forest achieving a 77% error reduction over the expert baseline (RMSE of 3.5 pp vs. 15.26 pp). Results are integrated into a Streamlit web interface accessible to non-technical users.

Keywords: S- curve, Time series forecasting, Big Data, Construction

1. Introduction

Materials procurement management in residential construction projects is one of the main sources of operational inefficiency. Purchasing departments must anticipate when and how much of each budget line (structure, installations, finishes) will be needed, in order to avoid both overstocking costs and work stoppages due to supply shortages. This planning is typically carried out manually, relying on accumulated expert knowledge, without a quantitative model that formalizes and leverages historical data from previous projects [1].

This work addresses this problem by applying big data and machine learning techniques to a dataset of over 60 residential construction projects [2]. The goal is to develop a system capable of predicting the evolution of the production curve for each purchase code throughout the project timeline (from early stages to completion) and to substantially improve accuracy compared to the current expert method.

2. Project definition

The starting point is a dataset extracted from the corporate SQL Server database, comprising 69 residential projects, 156 purchase codes with sufficient historical coverage, and 6,973 time series of cumulative progress normalized to the [0,1] interval in both time and production. Each series represents the monthly evolution of the completion percentage for a specific budget item in a specific project.

The exploratory analysis prior to modelling serves a dual purpose. On one hand, it empirically validates the logistic S-curve [3] hypothesis underlying model M6: the three-parameter logistic fit achieves an R^2 above 0.9 for most series, confirming with real data what had previously been assumed as a theoretical industry hypothesis [1]. On the other hand, it characterizes the dataset's heterogeneity: historical coverage varies widely across purchase codes (from items with dozens of available projects to codes used only occasionally with barely three or four records) which directly determines which models are viable in each case and how the evaluation protocol must be designed.

The complete system covers data extraction and normalization, training and evaluation of six prediction models of increasing complexity, and integration of results into a visualization layer accessible to the end user.

3. System description

Six prediction models are implemented and evaluated using consistent leave-one-out cross-validation: M1, a historical weighted average serving as the baseline; M2, similarity-based matching using Dynamic Time Warping; M3, LightGBM with a point-by-point formulation; M4, Random Forest with the same formulation; M5, an LSTM encoder-decoder network; and M6, parametric logistic curve fitting.

All models receive the static project characteristics and the already-observed portion of the curve as input and predict future points up to 100% of the project timeline.

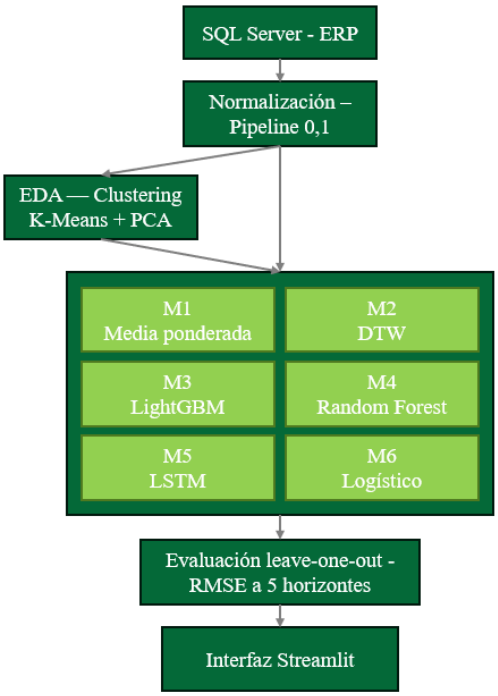


Ilustración 4 - System structure

4. Results

Evaluation is carried out at five observation horizons (10%, 20%, 30%, 50%, and 70% of elapsed time), using RMSE in percentage points of cumulative progress as the main metric. Results are summarized in the figures below.

Modelo	*RMSE global*	*10%*	*20%*	*30%*	*50%*	*70%*
M1 Media Ponderada	15.26 pp	15.26	15.26	15.26	15.26	15.26
M2 DTW	8.59 pp	10.30	8.36	7.82	7.98	8.78
M3 LightGBM	3.52 pp	8.23	5.90	3.72	1.82	0.67
M4 Random Forest	3.46 pp	7.91	5.65	3.90	1.76	0.68
M5 LSTM	5.86 pp	10.99	8.39	6.22	3.15	1.47
M6 Logístico	12.72 pp	16.19	13.89	12.24	8.60	6.32

Ilustración 5 - RMSE Comparison Across Models

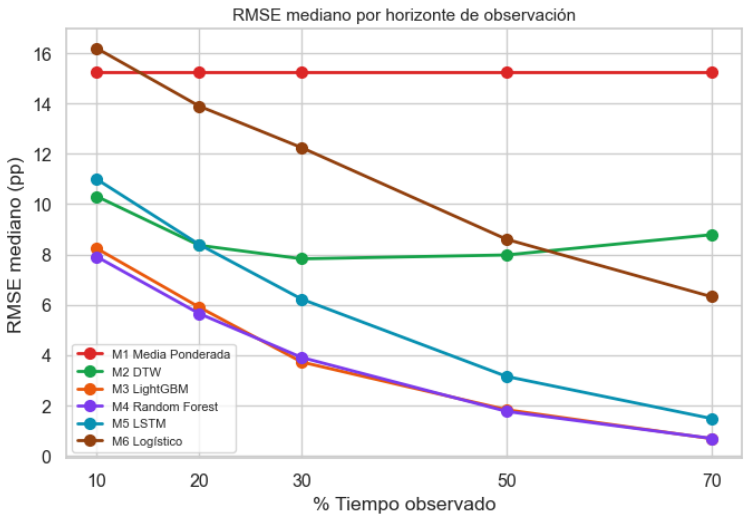


Ilustración 6 - Median RMSE Comparison Across Models

5. Conclusions

The developed system converts the tacit knowledge accumulated by the studies department into a reproducible quantitative model that improves prediction accuracy by 77% compared to the expert reference method [4]. The most valuable information for prediction is not the static project characteristics but rather the partially observed curve itself, which justifies the design of models that incorporate it explicitly. The architecture is designed for migration to a production environment on Databricks with Azure Data Lake Storage and integration with existing Power BI dashboards. Future work includes automatic deviation alerts, periodic retraining as new completed projects are incorporated, and the exploration of Transformer architectures for the sequential model.

6. References

[1] Muszynska, K. (2020). The S-curve as a tool for planning and controlling of construction process — Case study. Applied Sciences, 10(6), 2071. https://doi.org/10.3390/app10062071.

[2] Gondia, A., Siam, A., El-Dakhakhni, W., & Nassar, A. H. (2020). Machine learning algorithms for construction projects delay risk prediction. Journal of Construction Engineering and Management, 146(1), 04019085. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001736.

[3] Forbes, T., & Riso, T. (2024). Guide to S-Curve Modeling in Construction. Procore. https://www.procore.com/library/s-curve-modeling-construction.

[4] Alizadeh, E. (2020, octubre 11). An illustrative introduction to dynamic time warping. Towards Data Science. https://towardsdatascience.com/an-illustrative-introduction-to-dynamic-time-warping-36aa98513b98/.

Índice de la memoria

Índice de figuras

Índice de tablas

Índice de abreviaturas

Abreviatura	*Significado*
ADLS	Azure Data Lake Storage
ANOVA	Analysis of Variance (Análisis de la Varianza)
API	Application Programming Interface (Interfaz de Programación de Aplicaciones)
BI	Business Intelligence (Inteligencia de Negocio)
BIM	Building Information Modeling
CPU	Central Processing Unit (Unidad Central de Procesamiento)
CRISP-DM	Cross-Industry Standard Process for Data Mining
CSS	Cascading Style Sheets
DOI	Digital Object Identifier
DTW	Dynamic Time Warping (Deformación Dinámica del Tiempo)
EDA	Exploratory Data Analysis (Análisis Exploratorio de Datos)
ERP	Enterprise Resource Planning (Sistema de Planificación de Recursos Empresariales)
ETS	Error, Trend, Seasonality (modelo de suavizado exponencial)

Abreviatura	*Significado*
GA	Genetic Algorithm (Algoritmo Genético)
GPU	Graphics Processing Unit (Unidad de Procesamiento Gráfico)
ID	Identificador único
JDBC	Java Database Connectivity
KDE	Kernel Density Estimation (Estimación de Densidad por Núcleo)
KPI	Key Performance Indicator (Indicador Clave de Rendimiento)
LSTM	Long Short-Term Memory (Red Neuronal Recurrente de Memoria a Largo-Corto Plazo)
MDPI	Multidisciplinary Digital Publishing Institute
ML	Machine Learning (Aprendizaje Automático)
ODS	Objetivos de Desarrollo Sostenible
RMSE	Root Mean Squared Error (Raíz del Error Cuadrático Medio)
SQL	Structured Query Language (Lenguaje de Consulta Estructurado)
UML	Unified Modeling Language (Lenguaje Unificado de Modelado)
URL	Uniform Resource Locator (Localizador Uniforme de Recursos)

Capítulo 1. INTRODUCCIÓN

El sector de la construcción se caracteriza por la gestión simultánea de múltiples proyectos de alta complejidad, donde la correcta planificación del aprovisionamiento de materiales y la previsión del avance de las distintas obras son factores determinantes tanto para la rentabilidad económica como para el cumplimiento de los plazos comprometidos con el cliente. En este contexto, la capacidad de anticipar cómo va a evolucionar una obra a lo largo de su ciclo de vida, y de traducir esa evolución en necesidades concretas de los diferentes materiales y mano de obra especializada en cada momento, representa una ventaja competitiva de primer orden para cualquier empresa constructora.

El problema central que aborda este Trabajo de Fin de Máster nace de una tensión operativa bien conocida en el sector: los operarios en obra no pueden permanecer parados esperando materiales, pero tampoco es viable ni económicamente eficiente acumular stock en exceso en la propia obra. Adicionalmente a esto, hay que respetar la lógica constructiva que lleva a una secuencia de diferentes oficios en obra con sus solapes correspondientes. La solución a esta tensión es la previsión precisa de cuándo y en qué cantidad se va a necesitar cada tipo de material o trabajo a lo largo del ciclo de ejecución, lo que en la terminología del sector se traduce en conocer la curva de producción de cada código de compra: la evolución temporal del porcentaje de avance acumulado de cada partida presupuestaria desde el inicio hasta la finalización de la obra.

Históricamente, esta previsión ha recaído inicialmente sobre los departamentos de estudios de las empresas, cuyos técnicos elaboraban las estimaciones de avance apoyándose en su experiencia acumulada en proyectos anteriores y en el conocimiento profundo del comportamiento habitual de cada tipo de obra, aunque posteriormente podían ser modificados por los jefes de obra, que enfrentan a la realidad del día a día de la obra y deben hacer ajustes derivados de la participación de contingencias o de nuevas peticiones del cliente durante la ejecución de los trabajos. Este enfoque, basado en el criterio experto, ha funcionado de forma razonablemente efectiva mientras el volumen de proyectos simultáneos

era manejable y mientras los técnicos con más experiencia podían dedicar tiempo suficiente a la planificación. Sin embargo, presenta limitaciones estructurales que se agravan a medida que la empresa crece y gestiona un mayor número de obras en paralelo.

La primera limitación es la dependencia del conocimiento tácito individual. La experiencia que permite a un técnico estimar con acierto cuándo se va a necesitar la carpintería exterior o cuándo va a arrancar la instalación de climatización está almacenada en la memoria de las personas, no en los sistemas de información de la empresa. Cuando ese técnico no está disponible, cambia de proyecto o abandona la organización, ese conocimiento se pierde o se transfiere de forma incompleta. La segunda limitación es la escasa escalabilidad del proceso manual: a medida que aumenta el número de obras activas simultáneamente, el tiempo disponible por obra para hacer una previsión detallada se reduce, lo que obliga a simplificar las estimaciones o a asumirlas directamente de proyectos anteriores sin ajustarlas a las particularidades de este nuevo proyecto. La tercera limitación es la ausencia de retroalimentación sistemática: sin un mecanismo que compare las previsiones históricas con los avances reales registrados, es difícil mejorar la precisión de las estimaciones de forma objetiva y acumulativa a lo largo del tiempo. Las obras se terminan, y pocas veces se destinan recursos y tiempo en analizar qué se podía haber hecho mejor para haber reducido sobrecostes o plazos, ya que las siguientes obras en cartera consumen todos los recursos disponibles.

Las empresas disponen, sin embargo, de un activo de información valioso que hasta ahora no ha sido explotado de forma sistemática para este propósito: el histórico de curvas de producción reales de todas sus obras anteriores, desagregado por código de compra y registrado con periodicidad mensual en su sistema de gestión. Este histórico contiene la huella cuantitativa del conocimiento experto que los técnicos de estudio y los jefes de obra han aplicado durante años: en él están implícitos los patrones de comportamiento de cada tipo de partida presupuestaria en cada tipo de obra, las estacionalidades del sector, los efectos del tamaño y la complejidad del proyecto sobre el ritmo de ejecución, y las similitudes y diferencias entre proyectos de distinta naturaleza.

El presente Trabajo de Fin de Máster propone explotar ese histórico mediante técnicas de Big Data y aprendizaje automático para construir un sistema de predicción de curvas de producción por código de compra. El objetivo es que, dado un nuevo proyecto con sus características conocidas en el momento de inicio, el sistema sea capaz de predecir cómo va a evolucionar el avance acumulado de cada partida presupuestaria a lo largo del tiempo, proporcionando al departamento de estudios, en primer lugar, a la hora de presentar ofertas, y a los jefes de obra posteriormente una herramienta de apoyo cuantitativa que complemente y refuerce el criterio experto existente.

El trabajo que aquí se expone se estructura de la siguiente manera. En el Capítulo 2 se describen las tecnologías empleadas en el desarrollo del proyecto, incluyendo los lenguajes de programación, las arquitecturas de datos y los modelos utilizados. En el Capítulo 3 se analiza el estado de la cuestión, abordando la evolución del sector, la curva de producción S y las soluciones existentes en el ámbito del cálculo de esta curva. Posteriormente, en el Capítulo 4 se define el trabajo desarrollado, junto con la justificación, objetivos, metodología y planificación económica.

A continuación, los Capítulos 5 a 9 detallan el proceso de obtención y tratamiento de los datos, el análisis exploratorio, los distintos modelos utilizados y la interfaz. En el Capítulo 10 se analizan los resultados obtenidos y se evalúa el impacto funcional y organizativo del sistema. Finalmente, en el Capítulo 11 se recogen las conclusiones del proyecto y se plantean las líneas de trabajo futuro.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

En este capítulo se describen las principales tecnologías, herramientas y enfoques empleados en el desarrollo de este Trabajo de Fin de Máster. El objetivo es facilitar la lectura y comprensión de los capítulos posteriores, proporcionando el contexto técnico necesario. La arquitectura del proyecto combina herramientas de extracción y almacenamiento de datos en la nube, procesamiento y modelado en Python, control de versiones mediante Bitbucket y visualización de resultados mediante una interfaz web.

2.1 LENGUAJES DE PROGRAMACIÓN

El desarrollo del sistema se ha realizado utilizando Python [1] como lenguaje de programación principal. Su elección se fundamenta en su versatilidad, claridad sintáctica y amplio uso en proyectos relacionados con el tratamiento de datos, la automatización de procesos y la integración con tecnologías de Inteligencia Artificial.

Python dispone de un ecosistema muy amplio de librerías especializadas que facilitan la manipulación de datos, la conexión con servicios externos y la generación automatizada de documentos. Estas características permiten desarrollar soluciones complejas de forma eficiente y mantenible, adaptándose a los requisitos técnicos del proyecto. El listado completo de librerías utilizadas, junto con una breve descripción de su función, se recoge en el ANEXO B: Librerías utilizadas.

Por otro lado, se ha utilizado el lenguaje SQL [2] para la consulta y explotación de los datos almacenados en las distintas tablas y estructuras del sistema. Permite realizar operaciones de selección, filtrado, ordenación y agregación de datos de manera eficiente.

En el proyecto ha sido fundamental para la extracción de los datos relevantes desde el datalake y otros sistemas de almacenamiento, proporcionando los conjuntos de datos necesarios para su posterior transformación y análisis.

2.2 ENTORNO DE DESARROLLO Y CONTROL DE VERSIONES

Visual Studio Code [3] es el entorno de desarrollo integrado utilizado a lo largo de todo el proyecto. Su elección se justifica por la combinación de ligereza, extensibilidad y soporte nativo para Python y Jupyter Notebooks, que permite trabajar tanto con scripts modulares como con cuadernos de experimentación dentro del mismo entorno. Las extensiones de Python, Pylance [4] y Jupyter [5] proporcionan autocompletado, análisis estático de tipos y ejecución interactiva de celdas, lo que agiliza el ciclo de desarrollo y depuración del código.

El control de versiones del proyecto se gestiona mediante Bitbucket [6], plataforma de alojamiento de repositorios Git. El uso de Git [7] como sistema de control de versiones garantiza la trazabilidad completa de todos los cambios realizados en el código a lo largo del desarrollo, permitiendo revertir modificaciones, comparar versiones y mantener ramas separadas para la experimentación y la versión estable del sistema. Bitbucket se integra de forma nativa con el ecosistema Atlassian, lo que facilita su adopción en entornos corporativos donde otras herramientas del mismo ecosistema ya están implantadas.

La estructura del repositorio sigue una organización modular donde cada fase del pipeline tiene su propio módulo: extracción, preprocesado, feature engineering, modelos y evaluación. Esta separación facilita la reutilización de componentes y la incorporación futura del repositorio a un pipeline de integración continua en el entorno productivo Databricks [8].

2.3 ALMACENAMIENTO Y ARQUITECTURA DE DATOS

La fuente primaria de datos del proyecto es una base de datos relacional alojada en Azure SQL Database [9], el servicio de base de datos como servicio de Microsoft Azure basado en el motor SQL Server [10]. Esta base de datos centraliza el histórico de seguimiento económico de las obras de la empresa, incluyendo tanto las características de cada proyecto como el registro mensual de avance por código de compra.

La conexión desde Python se establece mediante el protocolo JDBC [11] usando el driver oficial de Microsoft para SQL Server, gestionado a través de las librerías JPype [12] y JayDeBeAPI [13]. Esta aproximación permite ejecutar consultas SQL arbitrarias desde el entorno Python sin necesidad de exportaciones manuales intermedias, garantizando que los datos utilizados en cada ejecución del pipeline corresponden siempre a la versión más actualizada disponible en el sistema. Las credenciales de acceso se almacenan en un fichero de configuración externo al repositorio para evitar su exposición en el control de versiones.

2.4 ENTRENAMIENTO DE LOS MODELOS

El entrenamiento de los modelos de machine learning, incluyendo LightGBM [14] y Random Forest [15], se realiza en el entorno local dado que sus requisitos computacionales son manejables en un equipo convencional. El tiempo de entrenamiento de estos modelos sobre el dataset disponible es del orden de minutos, lo que hace innecesario el uso de infraestructura de cómputo adicional para esta fase.

El entrenamiento del modelo de deep learning basado en redes LSTM [16] requiere un volumen de cómputo significativamente mayor que los modelos de machine learning clásicos, especialmente durante las fases de búsqueda de hiperparámetros donde se entrena el modelo múltiples veces con distintas configuraciones. Para este propósito se utiliza Google Colaboratory con aceleración por GPU [17], servicio que proporciona acceso gratuito a unidades de procesamiento gráfico NVIDIA en la nube a través de una interfaz de cuadernos Jupyter compatible con el entorno de desarrollo local.

La GPU acelera el entrenamiento de redes neuronales de forma sustancial porque las operaciones matriciales que dominan el cómputo de una red LSTM son paralelizables de forma natural sobre la arquitectura masivamente paralela de una GPU. El código de entrenamiento está implementado en PyTorch [18] con soporte para ejecución tanto en CPU como en GPU mediante la abstracción de dispositivo de PyTorch, de forma que el mismo script se ejecuta sin modificaciones en el entorno local durante el desarrollo y en Google Colaboratory durante el entrenamiento intensivo.

2.5 VISUALIZACIÓN Y PRESENTACIÓN DE RESULTADOS

Para facilitar la interacción con el sistema y la visualización de los resultados del modelo, se ha desarrollado una interfaz web mediante Streamlit [19], un framework de Python orientado a la construcción rápida de aplicaciones de datos. Su elección se fundamenta en la coherencia con el resto del stack tecnológico del proyecto, ya que permite desarrollar interfaces interactivas en Python de manera sencilla y sin necesidad de conocimientos de desarrollo web frontend.

La aplicación permite visualizar las distintas obras, tanto finalizadas como en ejecución. A su vez, se puede analizar la evolución de los distintos códigos de compra junto con la predicción de los modelos desarrollados. En el Capítulo 9: Interfaz Web se explican detalladamente las funcionalidades y componentes de la página web.

La aplicación se despliega como un servicio web accesible desde el navegador, siendo compatible con un despliegue futuro en Azure [20] o en Streamlit Cloud [21] para su integración en el entorno productivo.

Capítulo 3. ESTADO DE LA CUESTIÓN

En este capítulo se analiza el contexto actual del problema abordado en el trabajo, examinando la evolución histórica del uso de las curvas S en la gestión de proyectos de construcción, las soluciones tecnológicas y metodológicas existentes para su predicción, y las herramientas comerciales disponibles en el sector. Este análisis permite identificar las limitaciones de los enfoques actuales y justificar la contribución original del presente trabajo.

3.1 LA CURVA S EN LA GESTIÓN DE PROYECTOS DE CONSTRUCCIÓN

La representación gráfica del avance acumulado de un proyecto a lo largo del tiempo como una curva con forma de S tiene sus raíces en los primeros estudios sistemáticos sobre productividad industrial de principios del siglo XX. Sin embargo, su adopción generalizada en el sector de la construcción se produjo a partir de los años sesenta y setenta, impulsada por la necesidad de las grandes organizaciones contratistas y agencias gubernamentales de controlar el avance de proyectos de infraestructura de alta complejidad y larga duración.

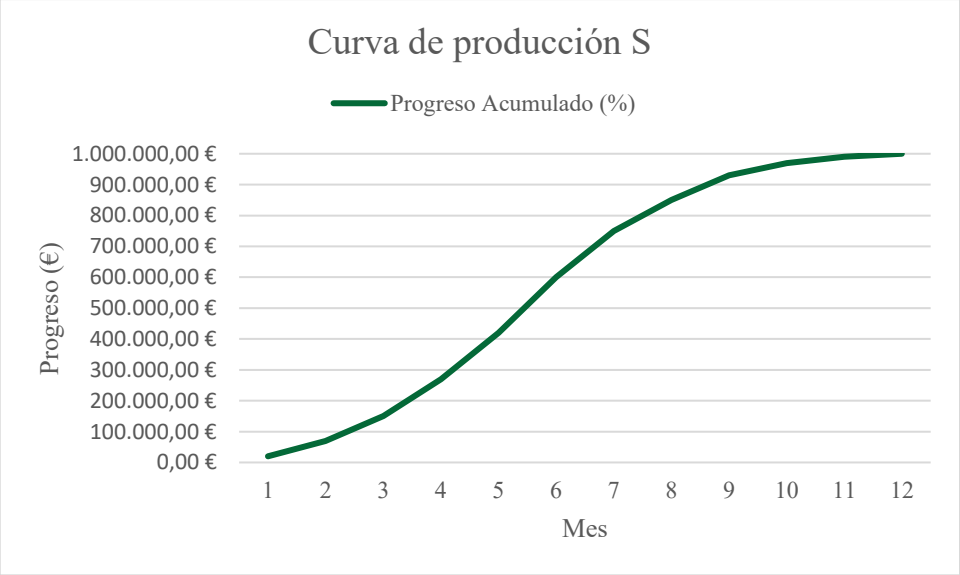


Figura 1 - Curva de producción S en una obra. Elaboración propia

La forma característica de la curva S responde a la dinámica natural de cualquier proyecto de construcción. En las fases iniciales el avance es lento porque la obra está en proceso de movilización y se utilizan de forma intensiva materiales de coste global más bajo (hormigón, acero…): se contratan subcontratas, se realizan los trabajos previos de acondicionamiento del terreno, excavación, estructura… y se ponen en marcha los procedimientos administrativos y logísticos. A medida que la obra alcanza su plena actividad, el ritmo de producción se acelera, entran materiales de mayor coste y oficios más especializados y la curva presenta su tramo más empinado. En las fases finales, los trabajos de acabado y los remates tienen un rendimiento menor por unidad de tiempo, el ritmo de certificación se ralentiza y la curva vuelve a aplanarse hasta alcanzar el 100% de ejecución. Este patrón, que matemáticamente se aproxima bien a una función logística o a una curva de Gompertz, es robusto y se reproduce de forma consistente en proyectos de muy distinta naturaleza y escala.

La curva S es ampliamente reconocida en la literatura como una herramienta clave para la planificación y el control de la ejecución de proyectos de construcción, permitiendo comparar el avance planificado con el avance real y detectar desviaciones sobre el calendario previsto. Su integración con la metodología del Valor Ganado, conocida en inglés como Earned Value Management, ha consolidado su uso como instrumento de control de costes y plazos en proyectos de todo tipo [22].

En la práctica profesional, la curva S se ha utilizado históricamente de dos formas complementarias. La primera es como herramienta de control: una vez iniciada la obra, se compara periódicamente el avance real acumulado con la curva planificada para detectar retrasos o adelantos. La segunda, y más relevante para este trabajo, es como herramienta de previsión: antes del inicio de la obra o en sus primeras fases, se estima cómo va a evolucionar el avance a lo largo del tiempo para planificar los recursos necesarios, los flujos de caja y las necesidades de aprovisionamiento. Esta segunda función es la que históricamente se ha realizado de forma manual, apoyándose en la experiencia de los técnicos y en la analogía con proyectos anteriores.

En la práctica, se distinguen múltiples tipos de curvas S según la variable que representan: curvas de coste acumulado, de flujo de caja, de horas de mano de obra, de entrega de materiales o de valor ganado, cada una con aplicaciones específicas en la gestión del proyecto [23]. Esta diversidad refleja que la curva S no es un único indicador sino una familia de representaciones del progreso acumulado que pueden aplicarse a cualquier variable medible en el tiempo.

3.2 MÉTODOS TRADICIONALES DE ESTIMACIÓN DE CURVAS S

Durante décadas, la estimación de la curva S de un proyecto se realizó mediante métodos puramente empíricos basados en el juicio experto. El técnico responsable de la planificación, apoyándose en su experiencia acumulada en proyectos similares, trazaba manualmente una curva que describía cómo esperaba que evolucionara el avance. Esta curva obedecía a una secuencia lógica del proceso constructivo que luego se cuantificaba en base a los importes de los capítulos y partidas de la obra. Este método, aunque efectivo cuando el técnico tenía experiencia relevante y el proyecto era similar a los anteriores, presentaba limitaciones estructurales evidentes: dependía de la disponibilidad del experto, no era sistemático ni reproducible, y no incorporaba de forma objetiva la información estadística de todos los proyectos históricos de la empresa.

El primer intento sistemático de formalizar matemáticamente la forma de la curva S llevó al uso de funciones polinómicas de tercer grado, que son capaces de reproducir la forma característica con un número reducido de parámetros. Estas funciones se ajustaban a los datos históricos de proyectos completados y se usaban para estimar la curva de nuevos proyectos. Sin embargo, los polinómicos tienen limitaciones importantes: pueden producir valores negativos o superiores al 100% fuera del rango de entrenamiento, y su interpretación en términos del dominio constructivo es limitada.

La función logística ofrece ventajas sobre el polinómico porque está acotada naturalmente entre 0 y un valor máximo L, sus tres parámetros tienen interpretación directa en términos del proceso constructivo, y su forma reproduce fielmente la dinámica de arranque lento,

aceleración central y desaceleración final. Chao y Chen propusieron modelar específicamente la posición del punto de inflexión y la pendiente en ese punto como las variables geométricas clave de la curva S, utilizando seis factores relacionados con los atributos y condiciones del proyecto como entradas a un modelo de redes neuronales, demostrando que este enfoque superaba a los modelos anteriores en precisión de predicción [24].

3.3 APLICACIÓN DE MACHINE LEARNING A LA PREDICCIÓN DE CURVAS S

La irrupción del aprendizaje automático en la gestión de proyectos de construcción durante la última década ha abierto nuevas posibilidades para la predicción de curvas S que superan las limitaciones de los métodos tradicionales. El enfoque general consiste en aprender de forma automática la relación entre las características del proyecto y la forma de su curva de producción a partir de los datos históricos disponibles, sin necesidad de que un experto codifique esa relación de forma explícita.

Chao y Chien desarrollaron en 2009 un modelo de redes neuronales para estimar la curva S de proyectos de construcción representada mediante parámetros polinómicos, demostrando su utilidad para la gestión contractual de proyectos [25] [26]. Este trabajo pionero estableció el precedente de usar datos históricos de proyectos para entrenar modelos predictivos de curvas S, aunque se limitaba a la predicción de la curva global del proyecto sin desagregarla por partidas presupuestarias.

$$y = ax^3 + bx^2 + (1 - a - b)x \qquad E.\ 1$$

Posteriormente se propusieron modelos que combinaban el razonamiento basado en casos con el ajuste de curvas logísticas [27]: el concepto de case-based reasoning permite recuperar proyectos históricos similares al proyecto en curso y usar sus curvas como estimación inicial, que se refina progresivamente a medida que se dispone de datos reales de avance del nuevo proyecto. Este enfoque híbrido, que combina similitud entre proyectos con ajuste

paramétrico, es conceptualmente muy próximo a la arquitectura de modelos propuesta en el presente trabajo.

En los últimos años, los modelos de gradient boosting han ganado protagonismo como herramienta de predicción en el sector de la construcción por su capacidad de manejar datos tabulares heterogéneos con alta eficiencia computacional. Estudios recientes han propuesto frameworks que combinan LightGBM con optimización metaheurística para predecir desviaciones de coste y plazo en proyectos de construcción, utilizando features extraídas de registros BIM, históricos de planificación y datos de liquidación de costes. La principal ventaja de LightGBM sobre otros modelos de gradient boosting como XGBoost es su velocidad de entrenamiento y su eficiencia en el uso de memoria, lo que lo hace especialmente adecuado cuando el volumen de datos es elevado o cuando se necesita reentrenar el modelo con frecuencia al incorporar nuevos proyectos [28].

Las redes neuronales recurrentes, y en particular la arquitectura LSTM, han demostrado su utilidad para la predicción de series temporales en el ámbito de la construcción. Se ha propuesto el uso de modelos LSTM para la predicción de índices de costes de construcción, demostrando ventajas sobre otros métodos avanzados como las máquinas de vectores de soporte gracias a su capacidad para capturar dependencias de largo alcance en la secuencia temporal y procesar vectores de características de alta dimensionalidad [29]. En el contexto de la predicción de curvas S, la arquitectura LSTM es especialmente relevante porque permite que el modelo aprenda tanto del histórico de avance ya observado de la obra en curso como de las características estáticas del proyecto, combinando información secuencial y tabular en una única representación.

3.4 SIMILITUD ENTRE PROYECTOS Y RAZONAMIENTO BASADO EN CASOS

Un enfoque alternativo y complementario al aprendizaje supervisado es el razonamiento basado en casos, que formaliza el proceso intuitivo que sigue un técnico experto cuando dice que este proyecto se parece a aquel otro de hace tres años. En lugar de aprender una función

global que mapee características a curvas, estos métodos buscan en el histórico los proyectos más similares al nuevo y usan sus curvas como estimación.

La medida de similitud entre proyectos es el componente crítico de este enfoque. Para proyectos descritos por variables numéricas y categóricas, la similitud se puede calcular mediante distancias ponderadas que asignan más peso a las variables más discriminativas. Para las curvas de producción en sí mismas, la medida de similitud más adecuada no es la distancia euclídea punto a punto, que requiere que las series tengan la misma longitud y estén alineadas en el tiempo, sino la distancia de tiempo dinámico o DTW por sus siglas en inglés.

El DTW es un algoritmo para medir la similitud entre dos secuencias temporales que pueden variar en velocidad, permitiendo detectar similitudes aunque una serie esté adelantada o atrasada respecto a la otra o aunque ambas tengan longitudes distintas [30]. Esta propiedad es especialmente valiosa en el contexto de las curvas S de construcción, donde dos obras pueden tener el mismo patrón de producción pero ejecutado en duraciones muy distintas, y donde la comparación en tiempo absoluto sin normalización llevaría a considerar disimilares curvas que son en realidad muy parecidas en su forma [31].

Investigaciones recientes han propuesto el uso de DTW para la clasificación de tendencias a largo plazo en series temporales de monitorización estructural, demostrando que bajo condiciones de pérdida de datos donde las métricas de distancia euclídea y coseno producen resultados incorrectos, el DTW mantiene una precisión de localización elevada gracias a su robustez ante desalineamientos temporales [32] [33].

3.5 HERRAMIENTAS COMERCIALES PARA LA GESTIÓN Y PREVISIÓN DE CURVAS S

El mercado de software de gestión de proyectos de construcción ofrece herramientas que incorporan la curva S como elemento de seguimiento y control, aunque con capacidades de predicción muy limitadas en comparación con lo que propone este trabajo.

Las plataformas más extendidas en el sector, como Primavera P6 [34], Microsoft Project [35] o Presto [36], permiten generar curvas S de avance planificado a partir del programa de obra y compararlas con el avance real registrado. Sin embargo, estas herramientas generan la curva S a partir de la planificación introducida manualmente por el técnico, no la predicen de forma automática a partir de datos históricos. El técnico sigue siendo el responsable de estimar cuándo y con qué ritmo se va a ejecutar cada partida.

Plataformas más avanzadas orientadas a la analítica de datos en construcción, como ProNovos [37], han incorporado capacidades de visualización y seguimiento de curvas S con integración automática de datos de los sistemas ERP y de campo. Estas plataformas permiten conectar los sistemas de campo con herramientas de gestión para actualizar automáticamente las curvas S con los datos reales de avance, facilitando la comparación entre lo planificado y lo ejecutado y apoyando las decisiones de planificación financiera y de recursos. Sin embargo, su funcionalidad se centra en el seguimiento y la visualización del avance real, no en la predicción de la curva futura a partir de características del proyecto y datos históricos.

En el ámbito de las patentes tecnológicas, se han registrado sistemas que combinan clustering no supervisado de proyectos históricos con modelos de machine learning para predecir qué proyectos pasados son más similares a uno nuevo, y usar esa similitud para generar predicciones de parámetros del proyecto. Este tipo de soluciones, orientadas al mercado de grandes contratistas internacionales, operan a nivel de proyecto global y no desagregan la predicción por partidas o códigos de compra, que es precisamente la granularidad que requiere el caso de uso que motiva este trabajo.

La principal brecha identificada en el análisis del mercado es la ausencia de herramientas que predigan la evolución de la producción a nivel de código de compra individual, es decir, que respondan a la pregunta de cuándo y en qué cantidad se va a necesitar cada tipo de material o trabajo específico a lo largo de la obra. Las soluciones existentes trabajan con la curva global del proyecto o con grandes agrupaciones de actividades, pero no con la granularidad necesaria para apoyar decisiones de aprovisionamiento detallado. Esta brecha es la que justifica la originalidad y la relevancia práctica del presente trabajo.

3.6 LIMITACIONES DE LOS ENFOQUES EXISTENTES Y CONTRIBUCIÓN DE ESTE TRABAJO

Del análisis anterior se desprenden tres limitaciones principales de los enfoques existentes que el presente trabajo aborda de forma explícita.

La primera es la granularidad. La literatura académica y las herramientas comerciales trabajan mayoritariamente con la curva S global del proyecto o con grandes agrupaciones de actividades. La predicción a nivel de código de compra individual, que es la granularidad relevante para la planificación del aprovisionamiento de materiales, no ha sido abordada de forma sistemática en la literatura revisada.

La segunda es la comparación de modelos. La mayoría de los trabajos académicos proponen y evalúan un único enfoque metodológico sin compararlo sistemáticamente con alternativas de distinta complejidad. Este trabajo propone la comparación explícita de seis familias de modelos bajo el mismo protocolo de evaluación leave-one-out, lo que permite identificar en qué condiciones cada enfoque es más adecuado y qué nivel de complejidad es necesario para obtener mejoras significativas sobre el baseline heurístico.

La tercera es la aplicabilidad directa a datos reales de empresa. La mayor parte de la investigación académica trabaja con datasets públicos o con colecciones reducidas de proyectos de características homogéneas. Este trabajo utiliza datos reales operacionales de una empresa constructora española, con toda la heterogeneidad, irregularidad y problemas de calidad que eso implica, lo que hace que los resultados sean directamente transferibles al entorno productivo.

Capítulo 4. DEFINICIÓN DEL TRABAJO

En este capítulo se define el alcance del proyecto desarrollado, estableciendo su justificación técnica y de negocio a partir del análisis del estado de la cuestión presentado en el capítulo anterior. Se detallan los objetivos generales y específicos que guían el desarrollo del sistema, la metodología empleada para su consecución y la planificación temporal y económica asociada.

4.1 JUSTIFICACIÓN

El análisis del estado de la cuestión ha demostrado que, a pesar de la existencia de herramientas comerciales de seguimiento de obra y de una línea de investigación académica activa en predicción de curvas S, ninguna de las soluciones existentes aborda el problema a la granularidad que requiere la planificación operativa del aprovisionamiento de materiales: la predicción del avance acumulado por código de compra individual a lo largo del ciclo de vida de cada obra.

Las herramientas de gestión de proyectos ampliamente utilizadas en el sector generan curvas S a partir de la planificación introducida manualmente por el técnico, sin capacidad de predicción automática basada en el histórico de proyectos anteriores. Las plataformas analíticas más avanzadas, como ProNovos, ofrecen visualización y seguimiento del avance real pero no predicción prospectiva. Y la literatura académica, aunque ha avanzado significativamente en la predicción de la curva S global del proyecto mediante redes neuronales y modelos paramétricos, no ha abordado la desagregación por partidas presupuestarias individuales.

Esta brecha tiene consecuencias operativas directas y costosas para cualquier empresa constructora. Cuando no se dispone de una previsión precisa de cuándo y en qué cantidad se va a necesitar cada tipo de material o trabajo, el departamento de compras y los jefes de obra se ven obligados a tomar decisiones de aprovisionamiento basadas en la experiencia

individual de los técnicos, sin respaldo cuantitativo sistemático. El resultado son dos tipos de errores con impacto económico significativo: el retraso en el suministro que obliga a parar trabajos y dejar operarios sin actividad, y el exceso de stock en obra que incrementa los costes de almacenamiento y aumenta el riesgo de deterioro o robo de material.

Como se ha comentado anteriormente, con la evolución de las tecnologías, el histórico de curvas de producción toma una gran importancia. El saber cómo explotar estos datos es algo crítico y diferencial en el sector de la construcción.

El presente Trabajo de Fin de Máster se justifica en la oportunidad de convertir ese activo de datos en un sistema de predicción automática que proporcione al departamento de estudios y a los jefes de obra una herramienta de apoyo cuantitativa, reproducible y escalable, que complemente y refuerce el criterio experto existente sin pretender sustituirlo.

4.2 OBJETIVOS

El objetivo general de este Trabajo de Fin de Máster es desarrollar un sistema de predicción de curvas de producción por código de compra, capaz de estimar cómo va a evolucionar el avance acumulado de cada partida presupuestaria a lo largo del ciclo de vida de un proyecto, a partir de las características conocidas de la obra y del histórico de proyectos anteriores almacenado en los sistemas corporativos.

Para la consecución de este objetivo general es necesario alcanzar los siguientes objetivos específicos:

- Analizar y comprender la estructura de los datos disponibles. Esto implica estudiar el modelo de datos existente en los sistemas corporativos, identificar las tablas y variables relevantes para la predicción de curvas de producción, y comprender la naturaleza, calidad y limitaciones de la información almacenada. Este análisis es la base sobre la que se sustentan todas las decisiones metodológicas posteriores.
- Diseñar y ejecutar un pipeline de extracción, limpieza y normalización de datos. Se pretende construir un flujo técnico reproducible que permita acceder a los datos

desde la base de datos corporativa en Azure, aplicar las transformaciones de limpieza necesarias para garantizar la calidad de los datos, y normalizarlos en un formato homogéneo adecuado para el modelado. Este pipeline incluye la gestión de valores faltantes y semánticos vacíos, la corrección de curvas no monótonas y la normalización temporal al intervalo [0, 1].

- Realizar un análisis exploratorio de datos exhaustivo. El EDA tiene como objetivo caracterizar el dataset en profundidad, validar empíricamente la hipótesis de que las curvas de producción siguen una forma S logística, identificar patrones, anomalías y relaciones entre variables, y extraer conclusiones que guíen el diseño de los modelos. Este análisis incluye la validación del ajuste logístico, el análisis de clustering de arquetipos de curva y el estudio de la relación entre las características de la obra y la forma de su curva de producción.
- Diseñar e implementar el feature engineering necesario para el modelado. Esto consiste en transformar las variables brutas disponibles en representaciones adecuadas para cada familia de modelos, incluyendo la codificación de variables categóricas, la construcción de métricas de similitud entre obras, la extracción de características de forma de curva y la generación de variables temporales y estacionales. Este proceso se describe de forma integrada en el capítulo 8, dentro de la descripción de cada modelo.
- Implementar y comparar seis familias de modelos de predicción. Se desarrollarán y evaluarán los siguientes enfoques: modelo heurístico de media ponderada, modelo heurístico basado en distancia temporal dinámica DTW, modelo de machine learning con LightGBM, modelo de machine learning con Random Forest, modelo de deep learning con red LSTM y modelo paramétrico logístico.
- Evaluar los modelos mediante un protocolo de evaluación riguroso. La comparación entre modelos se realizará mediante un leave-one-out que simula el escenario real de predicción, evaluando cada modelo a distintos porcentajes de avance temporal observado para medir su utilidad en diferentes momentos del ciclo de obra. La métrica principal será el RMSE sobre la curva predicha.

- Integrar el modelo seleccionado en una capa de visualización accesible. El modelo con mejor rendimiento se integrará en un sistema de presentación de resultados realizado en Streamlit, que permita a los usuarios de la empresa consultar las predicciones de curvas de producción para nuevas obras de forma visual e interactiva, sin necesidad de interactuar directamente con el código.

4.3 METODOLOGÍA

Para el desarrollo del proyecto se ha seguido una metodología Agile [38]. De esta manera se han agrupado las tareas por bloques y se han adaptado las fechas en función del progreso y los imprevistos.

En primer lugar se ha hablado con el departamento de estudios para conocer la forma de operar y entender cómo funciona la curva de producción.

Una vez comprendido se ha realizado un análisis de los datos existentes para encontrar posibles patrones o información relevante para el proyecto. A partir de estos descubrimientos se han preparado los datos de manera que se pudieran empezar a usar en los distintos modelos.

Por último se ha diseñado una interfaz que permite a un usuario no muy técnico interactuar con los modelos de una forma muy sencilla y poder visualizar fácilmente los datos, tanto de obras históricas como de obras en ejecución.

4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

A continuación, se muestra la organización y el tiempo que han ocupado las principales tareas realizadas durante el desarrollo del proyecto. Para ello se ha utilizado un diagrama de Gantt orientativo (Figura 2) [39], ya que en función de los imprevistos que surgían durante las semanas se han ido adaptado los plazos.

TAREAS	AÑO	2025			2026			
	MES	OCT	NOV	DIC	ENE	FEB	MAR	ABR
01. FASE PREVIA								
Reunión con los departamentos	Toma de contacto con los equipos implicados para comprender los procesos actuales y las necesidades del sistema.							
Definición del alcance funcional del sistema	Delimitación de qué outputs generará el modelo (curva en S, desglose por códigos, previsión mensual, etc.).							
Análisis de las tecnologías	Evaluación de las tecnologías disponibles para asegurar su adecuación al entorno y a los objetivos del proyecto.							
Plan de actuación	Planificación de las fases, tareas y tiempos necesarios para el desarrollo ordenado del proyecto.							
02. INGENIERÍA DE DATOS								
Identificación de fuentes de datos	Localización de históricos de obras (ERP, control de costes, compras, certificaciones).							
Extracción de datos históricos	Obtención y consolidación de información económica y productiva.							
Análisis exploratorio de datos (EDA)	Identificación de patrones, outliers, valores faltantes y comportamiento temporal.							
Normalización y homogeneización de estructuras	Unificación de nomenclaturas, códigos de compra y formatos temporales.							
Tratamiento de datos incompletos o inconsistentes	Imputación, exclusión o ajuste de registros inválidos.							
Validación de calidad de datos	Comprobación de coherencia, integridad y consistencia antes del modelado.							
03. DESARROLLO DE LOS MODELOS								
Definición del enfoque de forecasting	Selección de los modelos a probar.							
Ajuste de parámetros y entrenamiento	Optimización de los modelos mediante validación cruzada y ajuste de hiperparámetros.							
Evaluación de métricas	Análisis de MAE, RMSE u otras métricas relevantes.							
Validación temporal de coherencia	Verificación de que la trayectoria prevista mantiene comportamiento realista.							
04. VISUALIZACIÓN Y CUADROS DE MANDO								
Diseño de la interfaz web	Creación de una app con Streamlit para visualizar los resultados							
Construcción de dashboard de curva en S	Visualización interactiva de evolución temporal.							
Dashboard de desglose por códigos de compra	Análisis comparativo entre obras y distribución mensual.							
Implementación de los modelos en la interfaz	Incorporación de los modelos diseñados							
05. VALIDACIÓN Y CIERRE								
Pruebas con proyectos reales activos	Aplicación del modelo en obras en ejecución.							
Evaluación de resultados	Análisis del grado de cumplimiento de los objetivos definidos.							
Feedback de los profesionales	Recopilación de la opinión de los usuarios finales para validar la utilidad del sistema.							
Propuestas de mejora futura	Definición de posibles ampliaciones y mejoras del sistema.							
Elaboración de documentación técnica y de usuario	Redacción de manuales y guías para el uso y mantenimiento de la herramienta.							
Formación a los profesionales	Capacitación de los usuarios para el uso correcto de la herramienta.							

Figura 2 - Diagrama de Gantt del proyecto

En cuanto a la estimación económica, al tratarse de un desarrollo tecnológico interno para la empresa, no se contempla un modelo de ingresos ni una cuenta de explotación, sino el coste en el que ha incurrido la organización para llevar a cabo el sistema de predicción de curvas de producción. Se desglosan a continuación los principales conceptos de coste: personal, hardware, software e indirectos.

El coste de personal constituye la partida más significativa del proyecto, representando aproximadamente el 84% del coste total. Se han considerado dos perfiles: el ingeniero de datos encargado del desarrollo técnico íntegro del sistema, y el director del proyecto, responsable de la supervisión, orientación metodológica y validación de resultados.

Para el perfil de ingeniero de datos se han considerado 200 horas de trabajo efectivo, reflejo de la dedicación real acumulada a lo largo de los 7 meses de desarrollo del proyecto. El coste horario aplicado es de 25 €/hora, valor representativo del coste empresa de un perfil junior con formación de máster en el mercado de Madrid. Para el director del proyecto se estiman 40 horas dedicadas a reuniones de seguimiento, revisión de entregables y orientación técnica, con un coste horario de 80 €/hora, acorde al perfil de ingeniero senior o consultor especializado.

Tabla 1 - Costes de personal Año 0

Perfil	***Horas (h)***	***Coste horario (€/hora)***	***Total (€)***
Ingeniero de datos	200	25	5.000
Director del proyecto	40	80	3.200
Total personal	**240**		**8.200**

En cuanto al hardware, se ha utilizado un equipo informático personal compuesto por portátil y periféricos, valorado en 1.200 €. Dado que este equipo no se adquirió expresamente para el proyecto sino que se encontraba previamente disponible, el coste imputable al proyecto se

calcula mediante amortización lineal proporcional a la duración del desarrollo. Considerando una vida útil del equipo de 4 años (48 meses) y una duración del proyecto de 7 meses, la amortización imputable es la siguiente:

Tabla 2 - Costes de hardware Año 0

Equipo	***Valor (€)***	***Vida útil (meses)***	***Duración proyecto (meses)***	***Amortización (€)***
Portátil	1.000	48	7	145,83
Periféricos (ratón, teclado, pantalla)	200	48	7	29,17
Total hardware	**1.200**			**175**

Una de las ventajas del enfoque tecnológico adoptado en este proyecto es el uso íntegro de herramientas de código abierto o infraestructura ya disponible en el entorno corporativo, lo que elimina cualquier coste adicional de licenciamiento. Python y todas las librerías empleadas (pandas, scikit-learn, LightGBM, entre otras) son de distribución libre.

Finalmente, los costes indirectos recogen aquellos gastos de carácter general no directamente imputables al proyecto pero necesarios para su ejecución: electricidad, conexión a internet, uso de instalaciones y otros suministros. De acuerdo con la práctica habitual en la estimación de proyectos de ingeniería, se aplica un porcentaje del 15% sobre el total de costes directos.

Teniendo todo lo anterior en cuenta, el coste total estimado del proyecto asciende a 7.791,25 €, siendo el coste de personal la partida dominante. Este importe refleja exclusivamente el coste de desarrollo del sistema; los costes de mantenimiento, actualización de modelos y eventual migración al entorno productivo de Databricks constituirían partidas adicionales a considerar en una fase posterior de implantación.

Tabla 3 - Resumen de costes Año 0

Concepto	***Importe (€)***
Costes de personal	8.200
Costes de hardware	175
Costes de software	0
Subtotal costes directos	**8.375**
Costes indirectos (15%)	1.016,25
Total estimado del proyecto	**9.391,25**

Una vez concluida la fase de desarrollo, el sistema requiere una serie de costes recurrentes para garantizar su correcto funcionamiento, la vigencia de los modelos predictivos y la evolución de la plataforma. Se distinguen tres fases a lo largo del horizonte de cinco años contemplado: una fase inicial de despliegue en el entorno productivo existente de la empresa (años 1-2), una fase de migración a infraestructura cloud con Databricks y Azure Data Lake Storage (año 3), y una fase de explotación plena en cloud (años 4-5). Los costes se han actualizado anualmente con un IPC estimado del 2%

Se considera que el mantenimiento del sistema puede ser asumido por el mismo perfil de ingeniero de datos a razón de una dedicación parcial anual, incluyendo las tareas de reentrenamiento de modelos al incorporar nuevas obras finalizadas, actualización de librerías, resolución de incidencias y desarrollo de mejoras menores. En los años de migración a cloud (año 3) se estima una dedicación adicional puntual para las tareas de integración y configuración del entorno Databricks.

Durante los años 1 y 2 el sistema opera sobre la infraestructura corporativa existente de Arpada (SQL Server + servidor interno para Streamlit), por lo que el coste incremental de infraestructura es nulo. A partir del año 3 se contempla la migración al entorno cloud, con los costes asociados a Azure Data Lake Storage y Databricks Community, estimados en función de los volúmenes de datos y frecuencia de ejecución previstos. A partir del año 4 se añade la integración con Power BI Premium para la capa de visualización corporativa.

Se contempla el coste de la licencia de Power BI Premium por usuario a partir del año 4, así como los costes de soporte técnico de Microsoft Azure estimados de forma conservadora.

La tabla siguiente recoge el desglose completo de costes para el horizonte de cinco años, tomando el año 0 como el coste de desarrollo ya estimado anteriormente:

Tabla 4 - Costes de mantenimiento y explotación Año 0 + 5 primeros años

Concepto	***Año 0 (€)***	***Año 1 (€)***	***Año 2 (€)***	***Año 3 (€)***	***Año 4 (€)***	***Año 5 (€)***
Personal de mantenimiento						
Horas anuales (horas)	-	40	40	80	40	40
Coste horario (€/hora)	-	25	25,5	26,01	26,53	27,06
Total personal	-	1.000	1.020	2.080,80	1.061,20	1.082,40
Infraestructura						
SQL + servidor interno	0	0	0	-	-	-

Concepto	***Año 0 (€)***	***Año 1 (€)***	***Año 2 (€)***	***Año 3 (€)***	***Año 4 (€)***	***Año 5 (€)***
Azure Data Lake Storage	-	-	-	600	612	624,24
Databricks	-	-	-	1.200	1.224	1.248,48
Total infraestructura	0	0	0	1.800	1.836	1.872,72
Licencias y servicios						
Power BI	-	-	-	-	1.200	1.224
Soporte Azure	-	-	-	300	306	312,12
Total licencias	0	0	0	300	1.506	1.536,12
Costes indirectos (15%)	-	150	153	626,52	660,48	673,68
Total mantenimiento anual	**-**	**1.150**	**1.173**	**4.807,32**	**5.063,68**	**5.164,92**
Coste acumulado	**9.391,25**	**10.041,25**	**11.714,25**	**16.521,57**	**21.585,25**	**26.750,17**

Capítulo 5. SISTEMA DESARROLLADO

El sistema toma como punto de partida el histórico de curvas de producción reales de obras anteriores almacenado en los sistemas corporativos, junto con las características estáticas de cada proyecto, y produce como salida la predicción de cómo va a evolucionar el avance acumulado de cada código de compra a lo largo del ciclo de vida de una nueva obra. Esta predicción se presenta al usuario final a través de una interfaz web que permite consultar las curvas predichas de forma visual e interactiva sin necesidad de interactuar con el código.

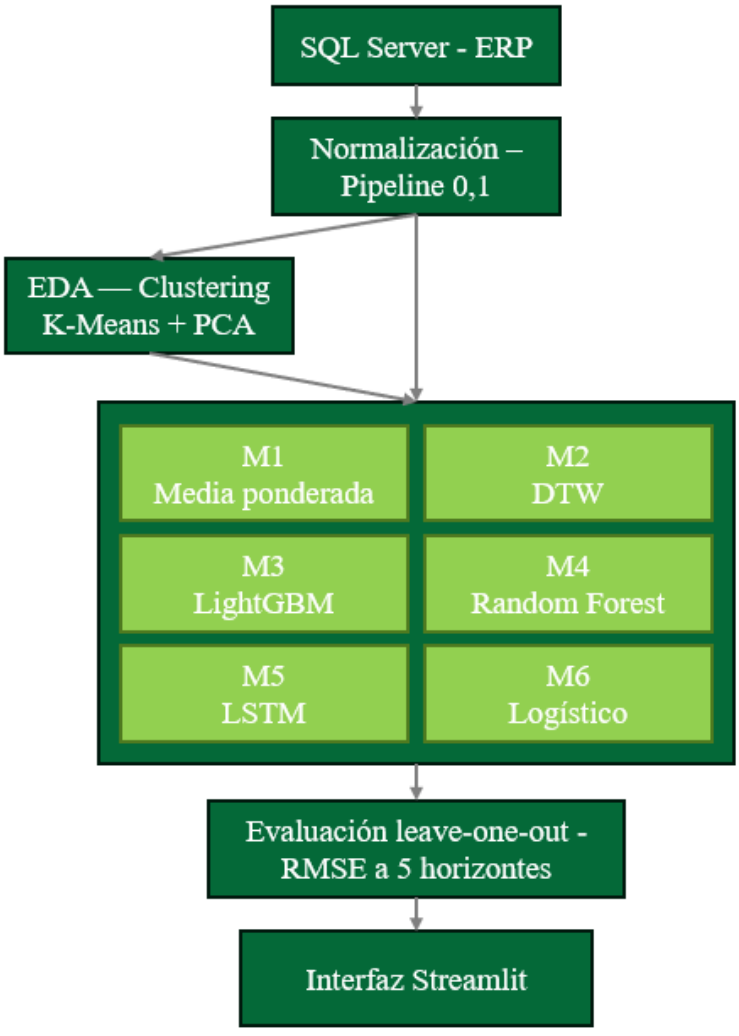


Figura 3 - Sistema desarrollado

El sistema se articula en cuatro etapas secuenciales que se ilustran en la Figura 3. La primera etapa es la extracción y normalización de datos: las curvas de producción históricas y las características de cada proyecto se obtienen directamente del ERP corporativo a través de consultas sobre SQL Server, y se normalizan en un rango 0-1 mediante un pipeline estandarizado que garantiza la comparabilidad entre proyectos de distinta escala y duración. La segunda etapa es el análisis exploratorio y la caracterización del dataset: se aplica un proceso de clustering mediante K-Means con reducción de dimensionalidad por PCA para identificar agrupaciones naturales entre proyectos y comprender la estructura del espacio de datos antes de entrenar los modelos. La tercera etapa es la predicción propiamente dicha, en la que seis modelos de distinta naturaleza generan de forma independiente la curva de producción esperada para cada código de compra de la obra nueva. La cuarta y última etapa es la presentación de resultados a través de una interfaz web desarrollada en Streamlit que permite al usuario consultar las predicciones de forma visual sin necesidad de conocimientos técnicos. En los siguientes apartados se detalla cada una de estas partes.

Capítulo 6. OBTENCIÓN Y TRATAMIENTO DE LOS DATOS

Este capítulo describe el proceso completo desde la fuente original de los datos hasta el dataset limpio y normalizado que se utiliza como entrada a los modelos. Se documenta el origen de cada tabla, la tipología de las variables, el proceso técnico de extracción y todas las transformaciones aplicadas antes del análisis exploratorio. El objetivo es que el proceso sea reproducible y que cada decisión de tratamiento quede justificada antes de que sus efectos sean visibles en el EDA.

6.1 ORIGEN DE LOS DATOS

Los datos utilizados en este trabajo provienen del sistema de gestión de proyectos de una empresa promotora y constructora del sector residencial español. No se trata de datos públicos ni de datasets de referencia estándar, sino de datos operacionales reales extraídos directamente del sistema de información corporativo. Esta naturaleza real de los datos es a la vez una fortaleza del trabajo, porque los resultados son directamente aplicables al negocio, y un reto metodológico, porque implica lidiar con problemas de calidad, inconsistencias y limitaciones de cobertura que no aparecen en datasets académicos curados.

Como se ha detallado anteriormente, el acceso a la información se realiza mediante consultas a una base de datos relacional, extrayendo la información necesaria a partir de dos fuentes diferenciadas según su estructura y nivel de detalle. Se pueden distinguir dos conjuntos de datos: unos centrados en la caracterización de los proyectos y otros centrados en el seguimiento de la ejecución.

El primer conjunto de datos presenta una granularidad de una fila por obra. Este conjunto recoge los atributos estáticos de cada proyecto, tales como sus dimensiones físicas y las

soluciones constructivas adoptadas. Esta información procede de la documentación técnica inicial y, aunque es susceptible de actualizaciones menores durante la ejecución, se trata a efectos de este estudio como un conjunto de características constantes para cada proyecto, permitiendo definir el contexto técnico de cada intervención de forma estable.

El segundo conjunto de datos ofrece un nivel de detalle significativamente mayor al estructurarse como una serie temporal desglosada por obra, código de compra y mes. Este nivel recoge el histórico de avance acumulado, indicando el porcentaje de ejecución certificado para cada partida presupuestaria en cada periodo. Esta fuente constituye la base para la generación de la variable objetivo del modelo. Su naturaleza, marcada por una serie temporal irregular (donde la disponibilidad de registros varía según el estado y la tipología de cada obra), condiciona de forma determinante tanto el diseño del preprocesado como la arquitectura de los modelos de aprendizaje automático empleados.

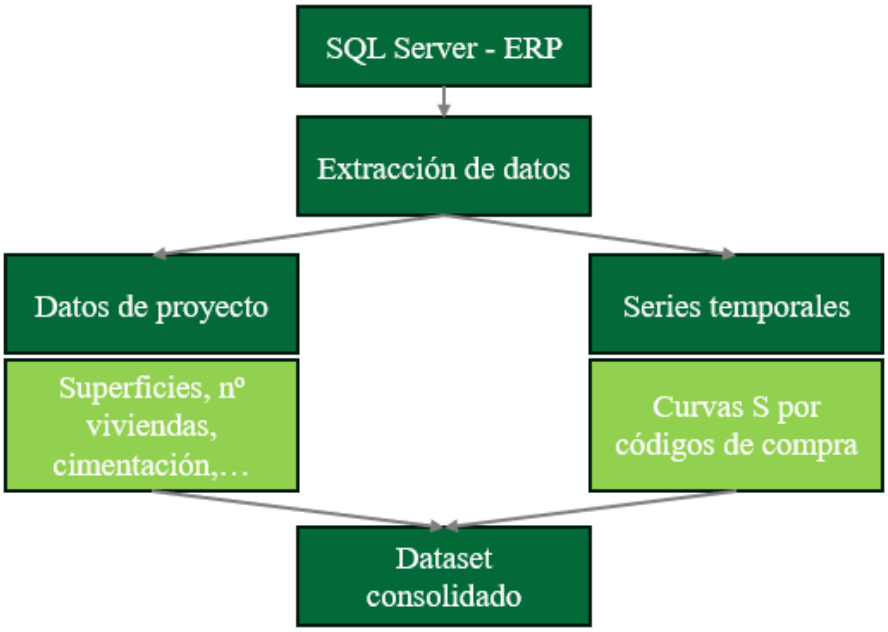


Figura 4 - Origen de los datos

6.2 TIPOLOGÍA DE LOS DATOS TRATADOS

Los datos del proyecto combinan tres tipos de variables (numéricas, categóricas y series temporales) con naturalezas estadísticas y requisitos de tratamiento muy distintos, lo que hace necesario un pipeline de preprocesado diferenciado para cada tipo.

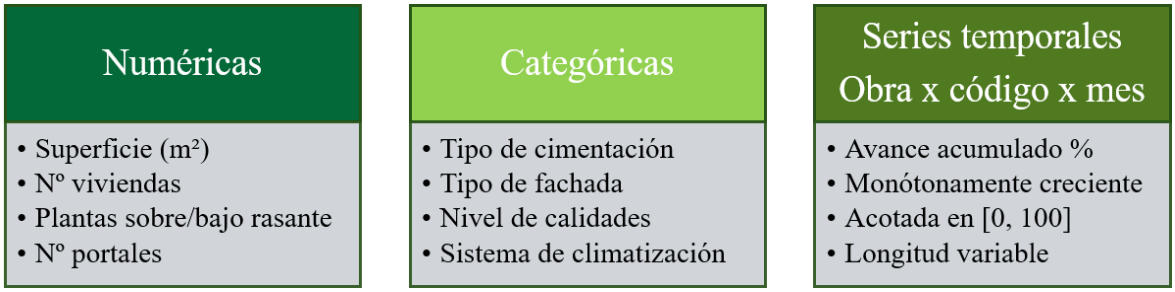


Figura 5 - Tipología de los datos tratados

6.2.1 VARIABLES NUMÉRICAS: SUPERFICIE, VIVIENDAS, DURACIÓN, PLANTAS

Las variables numéricas cuantifican atributos físicos de la obra que son directamente medibles y comparables entre proyectos. La superficie construida en metros cuadrados es la variable de tamaño más representativa y la que mayor correlación tiene con el presupuesto total y la duración estimada de la obra. El número de unidades de vivienda es relevante en promociones residenciales y está relacionado con la superficie aunque no de forma exacta, dado que el tamaño medio de las viviendas varía entre proyectos. Las plantas sobre rasante y bajo rasante determinan la complejidad estructural y condicionan la secuencia de ejecución de las partidas. El número de portales es un indicador de la distribución horizontal.

Estas variables requieren tratamiento específico antes de usarse en métricas de distancia por dos razones. La primera es la diferencia de escala: la superficie construida se mide en miles de metros cuadrados mientras que el número de portales raramente supera diez, y sin normalización la primera dominaría cualquier cálculo de distancia. La segunda es la asimetría de sus distribuciones, que hace que la distancia euclídea estándar sea sensible a outliers. Ambos problemas se abordan mediante normalización robusta basada en el rango intercuartílico antes de calcular similitudes entre obras.

6.2.2 VARIABLES CATEGÓRICAS: CIMENTACIÓN, FACHADA, CALIDADES, CLIMATIZACIÓN

Las variables categóricas codifican las decisiones de diseño y las soluciones constructivas adoptadas en cada obra. A diferencia de las variables numéricas, no tienen un orden natural ni una métrica de distancia obvia: la distancia entre "losa de cimentación" y "zapatas aisladas" no es cuantificable en la misma escala que la distancia entre 1000 y 2000 metros cuadrados.

El tratamiento de estas variables para el modelo de similitud requiere una decisión de codificación. La opción adoptada es la codificación one-hot binaria, donde cada categoría se convierte en una variable indicadora, combinada con una ponderación por entropía que da más peso en el cálculo de similitud a las variables con mayor poder discriminativo. Las variables con alta proporción de la categoría DESCONOCIDO se tratan de forma especial: no se interpreta como una categoría constructiva real sino como ausencia de información, y su similitud con cualquier otra categoría se establece en 0.5 en lugar de 0, reflejando incertidumbre en lugar de diferencia.

6.2.3 SERIES TEMPORALES: CURVAS S ACUMULADAS POR CÓDIGO DE COMPRA

Las series temporales de avance acumulado son el tipo de dato más complejo del proyecto porque combinan tres dimensiones simultáneamente: la obra, el código de compra y el tiempo. Cada serie es una secuencia de observaciones que describe la evolución del avance de una partida presupuestaria específica a lo largo de la ejecución de una obra concreta.

Las propiedades estadísticas más relevantes de estas series para el modelado son las siguientes. Son series acumuladas y por tanto monótonamente crecientes en condiciones ideales, lo que las distingue de series temporales generales donde los valores pueden oscilar libremente. Están acotadas en el intervalo [0, 100] porque representan un porcentaje. Tienen longitudes variables porque las obras tienen duraciones distintas. Y tienen una forma sistemática que se aproxima a una curva S logística, como se valida empíricamente en el EDA realizado durante el proyecto.

6.3 PROCESO DE EXTRACCIÓN: SQL SERVER VIA JDBC Y FALLBACK EXCEL

La extracción de los datos se ha realizado mediante conexión directa a la base de datos relacional corporativa (SQL server) utilizando protocolos de conectividad estándar. Esta aproximación permite ejecutar consultas de selección sobre las tablas sin necesidad de exportaciones manuales intermedias, lo que reduce el riesgo de errores en el volcado de datos y garantiza que la información procesada corresponda siempre a la versión más actualizada disponible en el sistema.

La conexión se establece mediante el controlador de base de datos nativo correspondiente, empleando un sistema de autenticación por usuario y contraseña (JDBC). Estas credenciales se mantienen seguras almacenándose en un fichero de configuración de variables de entorno, completamente externo al código fuente, evitando así su exposición en el control de versiones. Asimismo, la comunicación se configura con cifrado habilitado y aceptación automatizada del certificado del servidor, en consonancia con los estándares de seguridad habituales para entornos corporativos en red interna.

Durante la extracción, se han ejecutado consultas de selección completa sobre las distintas tablas, omitiendo la aplicación de filtros en origen. Esta decisión técnica busca preservar la flexibilidad necesaria para aplicar cualquier criterio de filtrado de forma programática durante la posterior fase de preprocesado, siguiendo el principio de realizar una única extracción consolidada para su filtrado múltiple en memoria, lo cual resulta eficiente dado el volumen de datos manejado. Adicionalmente, se ha diseñado un mecanismo de respaldo o fallback basado en la lectura de ficheros de hoja de cálculo para aquellos escenarios en los que la conexión a la red corporativa no se encuentre disponible. Estos archivos funcionan como réplicas estáticas de las tablas originales, conservando exactamente la misma estructura de columnas. De esta forma, el flujo de preprocesado posterior permanece inalterado con independencia de la fuente de origen empleada. La alternancia entre la fuente de datos principal en red y la fuente de respaldo local se gestiona de manera dinámica mediante un parámetro definido en la configuración del entorno.

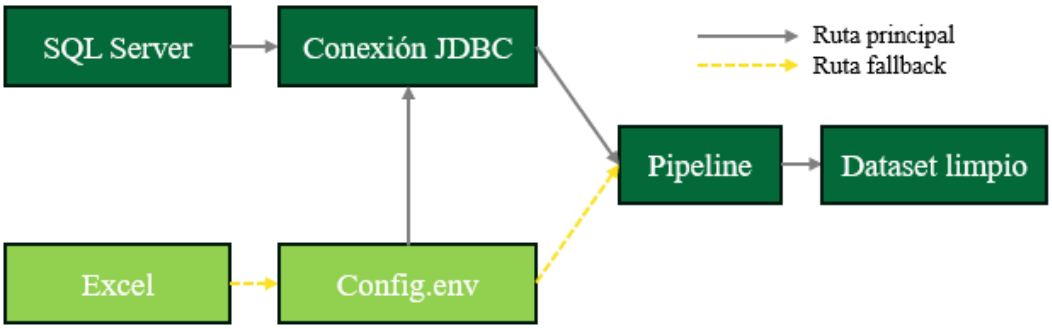


Figura 6 - Proceso de extracción de los datos

6.4 TRATAMIENTO, LIMPIEZA Y NORMALIZACIÓN

El pipeline de tratamiento de datos se ha diseñado con el principio de que todas las transformaciones sean explícitas, documentadas y reproducibles. Cada paso del pipeline modifica el dataset de una forma concreta y justificada, y el orden de los pasos es deliberado porque algunas transformaciones dependen de que otras se hayan aplicado previamente.

6.4.1 GESTIÓN DE VALORES FALTANTES Y SEMÁNTICOS VACÍOS

El primer paso del pipeline es la estandarización de los valores faltantes porque afecta a todos los pasos posteriores. Se define un conjunto canónico de valores semánticos vacíos que incluye las cadenas de texto "SIN DATOS", "N/A", "SIN DEFINIR", "ND", la cadena vacía, el espacio en blanco y las representaciones textuales de nulo como "nan" y "None". Todos estos valores se sustituyen por el valor nulo estándar de Python antes de cualquier otro procesado.

Una vez estandarizados los nulos, se calcula el porcentaje de faltantes por columna separando la contribución de nulos reales y nulos semánticos. Las columnas con más del 50% de faltantes se excluyen del conjunto de features del modelo de similitud y del modelo supervisado, aunque se conservan en el dataset para el análisis descriptivo del EDA. Las columnas entre el 20% y el 50% se imputan con la mediana para variables numéricas y con la moda para variables categóricas. Las columnas con menos del 20% de faltantes se imputan con el mismo criterio sin penalización adicional.

6.4.2 CORRECCIÓN DE CURVAS NO MONÓTONAS

La variable de avance acumulado debe ser monótonamente creciente por la naturaleza del proceso que describe. Para garantizar esta propiedad en todas las series del dataset se aplica una corrección basada en la acumulación del máximo: para cada punto de la serie, su valor corregido es el máximo entre su valor original y el valor corregido del punto anterior. Esta transformación es la menos invasiva posible porque solo modifica los puntos que violan la monotonía y lo hace de la forma más conservadora, preservando el valor máximo alcanzado hasta ese momento.

Además de la corrección de monotonía, se aplica clipping de la variable objetivo al rango [0, 100] para eliminar los valores físicamente imposibles. El clipping se aplica después de la corrección de monotonía para que la corrección opere sobre los valores originales y no sobre valores ya clippeados.

6.4.3 NORMALIZACIÓN TEMPORAL [0, 1]

La normalización temporal es la transformación más importante del pipeline porque hace posible comparar, promediar y modelar curvas de obras con duraciones muy distintas. Sin ella, el mes 12 de una obra de 12 meses y el mes 12 de una obra de 36 meses representan fases constructivas completamente distintas y no son comparables.

La transformación asigna a cada registro temporal de una serie el valor:

$$E.\ 2 \qquad t = \frac{fecha - fecha_{inicio}}{fecha_{fin} - fecha_{inicio}}$$

donde $fecha_{inicio}$ y $fecha_{fin}$ son el primer y último registro disponible para esa combinación obra-código. El resultado es un valor en el intervalo [0, 1] donde 0 representa el inicio de la ejecución de ese código en esa obra y 1 representa su finalización. Para series donde el primer punto no corresponde exactamente al inicio ($t > 0$) o el último no corresponde exactamente al final (t<1), se añaden artificialmente los puntos extremos con valores 0 y 100 respectivamente, bajo la hipótesis de que el código parte de cero y termina al 100%.

La normalización temporal se aplica por separado a cada combinación obra-código para que la referencia temporal sea local a cada serie y no dependa de fechas absolutas del calendario.

6.5 CATÁLOGO DE CÓDIGOS DE COMPRA: COBERTURA Y RELEVANCIA ECONÓMICA

Los códigos de compra son las partidas presupuestarias en las que se descompone el coste de cada obra. Cada código representa un tipo de trabajo o suministro concreto, como la estructura de hormigón, la fachada ventilada, la carpintería exterior o las instalaciones de climatización. El número total de códigos distintos presentes en el dataset son de 156 (con cobertura histórica suficiente), aunque su presencia varía enormemente entre obras.

La cobertura de un código se define como el porcentaje de obras del dataset que tienen al menos un registro de ese código. Los códigos con alta cobertura son partidas que aparecen en la práctica totalidad de los proyectos, lo que los convierte en los candidatos más robustos para el modelado porque disponen de más ejemplos históricos. Los códigos con baja cobertura son partidas especializadas que solo aparecen en ciertos tipos de obra, lo que limita la cantidad de datos disponibles para aprender su comportamiento.

Desde el punto de vista económico, la relevancia de un código no depende solo de su cobertura sino también de su peso presupuestario medio sobre el coste total de la obra. Un código que aparece en pocas obras pero representa el 30% del presupuesto de cada una de ellas es más relevante económicamente que un código muy frecuente pero de bajo impacto. La combinación de cobertura y peso económico determina qué códigos se priorizan en el modelado y cuáles se tratan de forma secundaria.

Para los modelos de machine learning y deep learning se establece un umbral mínimo de cobertura por debajo del cual el código queda excluido del entrenamiento por falta de ejemplos suficientes. Los modelos heurísticos y el modelo paramétrico no tienen este umbral porque pueden operar con un número reducido de ejemplos históricos, aunque con mayor incertidumbre en la predicción.

Capítulo 7. ANÁLISIS EXPLORATORIO DE LOS DATOS (EDA)

El análisis exploratorio de datos constituye la base sobre la que se sustenta todo el trabajo de modelado posterior. Su objetivo no es meramente descriptivo sino fundamentalmente metodológico: cada hallazgo se traduce en una decisión de diseño concreta que afecta al preprocesado, al feature engineering o a la arquitectura de los modelos. En este capítulo se analiza en profundidad la estructura, calidad y comportamiento de los datos disponibles, con especial atención a la validación empírica de la hipótesis central del trabajo: que las curvas de producción por código de compra siguen una forma logística de tipo S.

7.1 DESCRIPCIÓN GENERAL DEL DATASET

El dataset está compuesto por dos tablas relacionadas entre sí mediante el identificador de obra. Comprender la estructura de ambas tablas y la relación entre ellas es el primer paso necesario antes de cualquier análisis, porque condiciona qué modelos son aplicables y sobre qué subconjunto de datos.

La primera tabla recoge las características estáticas de cada obra, con una fila por proyecto. Sus variables describen atributos conocidos en el momento de inicio de la obra: dimensiones físicas como la superficie construida, el número de viviendas, las plantas sobre y bajo rasante y el número de portales; y atributos cualitativos que describen las soluciones constructivas adoptadas, como el tipo de cimentación, la estructura, la fachada, las cubiertas, los acabados interiores y el sistema de climatización. Estas variables representan el conocimiento disponible sobre una obra en el momento en que se quiere generar una predicción, y por tanto son las features de entrada a los modelos supervisados. A continuación se muestra un ejemplo de algunas de las columnas (Tabla 5).

Tabla 5 - Características estáticas de las obras

ID Obra	*Ubicación*	Estado	*Tipo de edificación*	*Ud. Viviendas*
0001	Alcobendas	Obra terminada	Viv. colectiva	45
0002	Getafe	En ejecución	Unifamiliares	80
0003	Madrid	Obra terminada	Viv. colectiva	64

La segunda tabla contiene las curvas de producción históricas con granularidad obra × código de compra × mes. Para cada combinación de obra y código de compra existe una secuencia temporal de registros donde la variable registrada es el porcentaje de avance acumulado sobre el presupuesto total asignado a ese código en esa obra. Esta variable es el objetivo del modelado: dado que se conocen los primeros puntos de la serie, el modelo debe predecir su evolución futura hasta completar el 100%. En la Tabla 6 se muestra un ejemplo de algunas de las columnas que lo conforman.

Tabla 6 - Evolución de las obras por código de compra y mes

ID Obra	*Código compra*	Fecha	*Avance (€)*	*Avance acumulado (€)*
0001	1020	2024-05-01	10.320,12	15.500,21
0002	1040	2024-05-01	3.455,24	5.000,00
0003	2080	2024-06-01	2.825,45	3.456,78

Una consideración crítica es que la relación entre ambas tablas no es total. Existen obras con curvas históricas disponibles pero sin características registradas en la primera tabla, y obras con características pero sin curvas de producción. Solo el subconjunto de obras presentes en ambas tablas es utilizable para los modelos supervisados que aprenden la relación entre

características y forma de curva. Las obras sin características solo pueden usarse en modelos no supervisados o heurísticos que no requieren features de entrada.

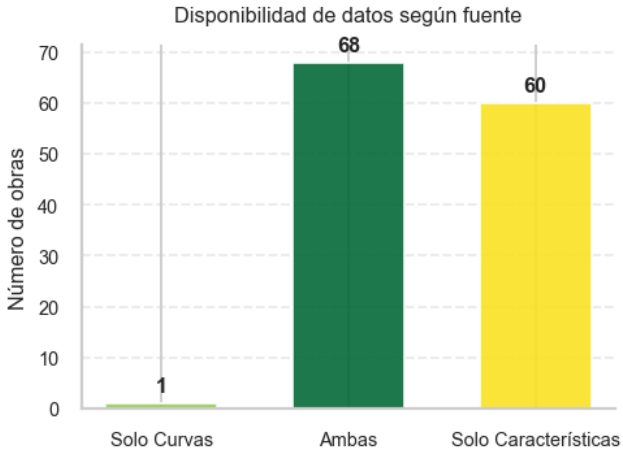


Figura 7 - Disponibilidad de los datos en cada una de las fuentes

7.2 CALIDAD DE LOS DATOS

El análisis de calidad de datos precede a cualquier análisis descriptivo porque los problemas no detectados en esta fase se propagan silenciosamente a todas las etapas posteriores, contaminando tanto las conclusiones del EDA como los resultados de los modelos. Este bloque documenta de forma sistemática todos los problemas encontrados, su magnitud y las decisiones tomadas para resolverlos.

7.2.1 VALORES FALTANTES

El análisis de valores faltantes debe distinguir entre dos tipos de ausencias con naturaleza distinta. Los nulos reales son celdas vacías en la base de datos, resultado de datos que nunca fueron registrados. Los nulos semánticos son valores que formalmente contienen texto pero que en la práctica significan ausencia de información: valores como "SIN DATOS", "N/A", "SIN DEFINIR" o "ND" son equivalentes a un nulo real desde el punto de vista del modelado y deben tratarse como tal. Ignorar los nulos semánticos y tratarlos como una categoría válida introduciría una categoría artificial sin significado real que contaminaría el modelo de similitud entre obras.

En la tabla de características (Figura 8 y Figura 9), el análisis revela una distribución muy desigual de los faltantes entre columnas. Algunas variables presentan porcentajes de ausencia superiores al 50%, lo que implica que más de la mitad de las obras no tienen ese dato registrado. Para estas columnas la decisión adoptada es excluirlas del modelado supervisado, porque introducen más ruido que señal: un modelo que aprende sobre una variable ausente en la mitad de los casos no puede generalizar de forma fiable. Se mantienen en el análisis descriptivo pero no entran como features. Las columnas con entre un 20% y un 50% de faltantes se imputan con la moda en el caso de variables categóricas y con la mediana en el caso de variables numéricas, decisión conservadora que preserva la distribución original sin introducir sesgos de escala. Las columnas con menos del 20% de faltantes se imputan de la misma forma sin penalización.

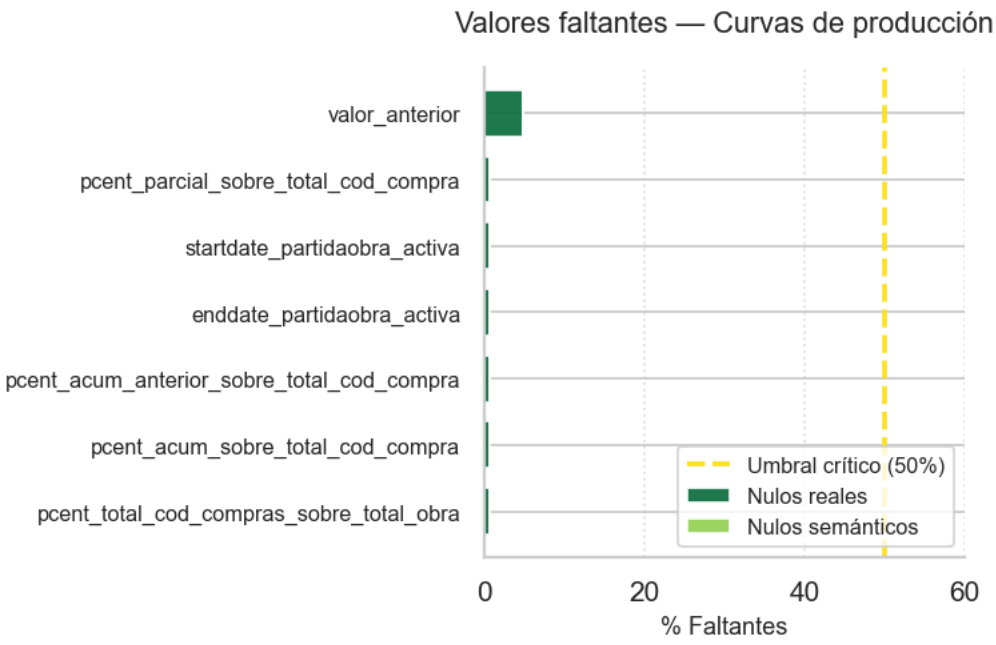


Figura 8 - Valores faltantes en las curvas de producción

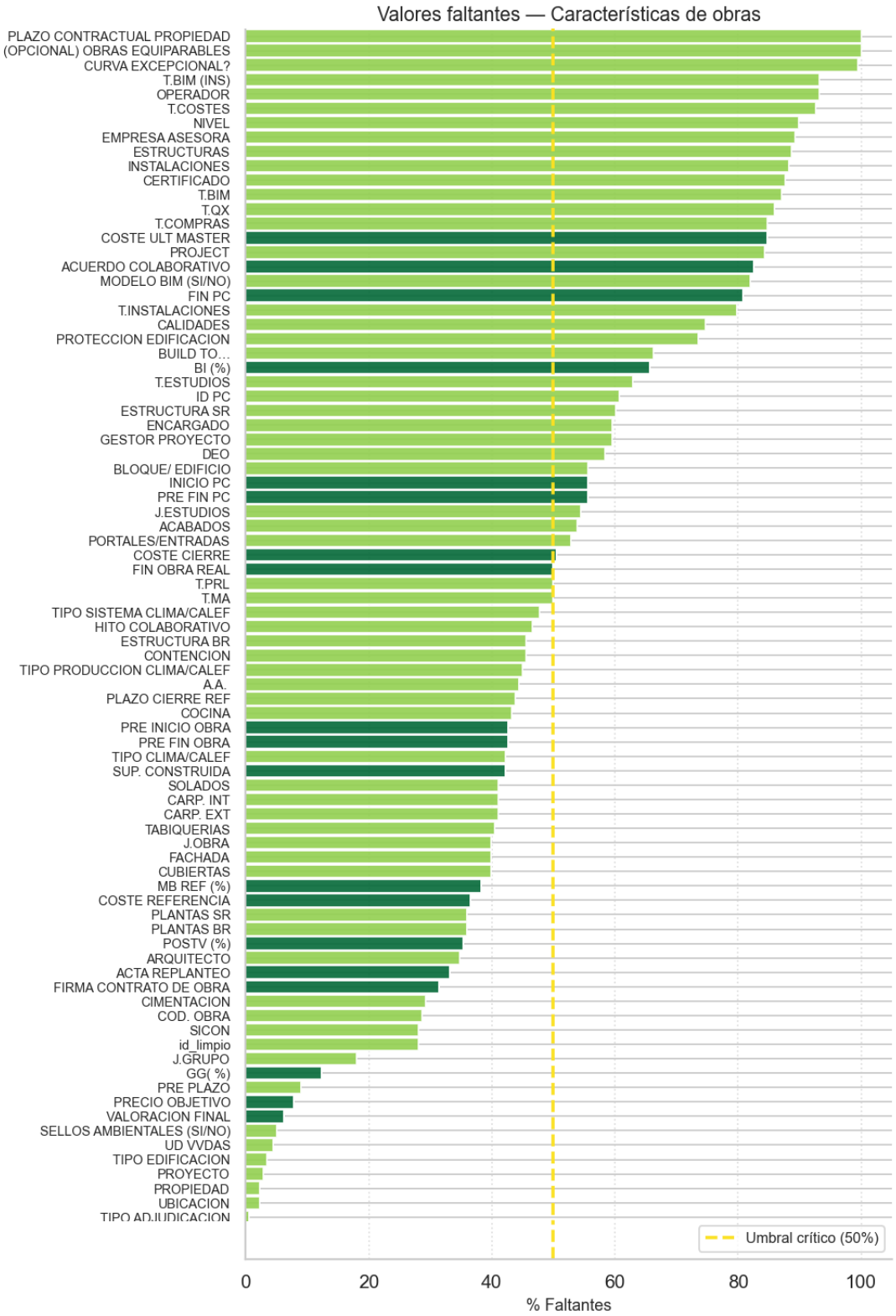


Figura 9 - Valores faltantes en las características de las obras

7.2.2 DUPLICADOS

En la tabla de curvas se detectaron duplicados funcionales, definidos como registros con la misma obra, el mismo código de compra y la misma fecha pero con valores de avance distintos. Este patrón no es un error de inserción aleatoria sino que responde a un comportamiento sistemático del sistema fuente: las correcciones de datos se registran como nuevas filas en lugar de sobrescribir el valor original. La decisión adoptada es conservar el último registro cronológico disponible para cada combinación obra-código-fecha, bajo la hipótesis de que el valor más reciente incorpora la corrección más actualizada y es por tanto el más fiable.

En la tabla de características se verifica que no existen obras duplicadas, es decir, que cada identificador de obra aparece una única vez.

```
Duplicados exactos — características : 0
Duplicados exactos — curvas          : 0
Duplicados funcionales curvas (obra+cod+fecha): 0
Obras duplicadas en tabla características: 71

Registros curvas tras eliminar duplicados: 109,110
```

Figura 10 - Comprobación de duplicados

7.2.3 RANGOS IMPOSIBLES EN LA VARIABLE OBJETIVO

El avance acumulado es una variable definida en el intervalo [0, 100] por construcción del negocio: representa un porcentaje de ejecución que no puede ser negativo ni superar el 100% de lo presupuestado. Sin embargo, el dataset contiene un pequeño porcentaje de valores fuera de este rango. Los valores negativos son atribuibles a ajustes contables que generan partidas de signo contrario. Los valores superiores a 100 pueden deberse a reasignaciones presupuestarias donde el denominador cambia retroactivamente. Todos estos valores se corrigen mediante clipping al rango [0, 100] antes de cualquier análisis o entrenamiento. Esta decisión es la más conservadora posible y es robusta como política general dado que no se dispone de información adicional sobre el origen de cada anomalía individual.

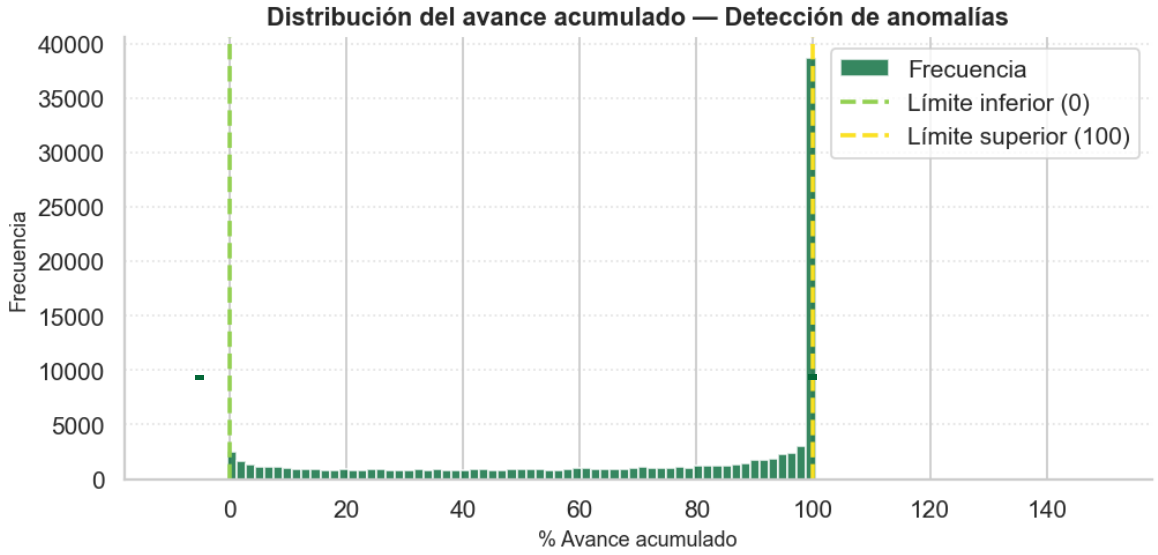


Figura 11 - Distribución del avance acumulado de las obras

7.2.4 INCREMENTOS NEGATIVOS Y MONOTONÍA

Una propiedad fundamental del avance acumulado es que debe ser monótonamente creciente: una vez ejecutado un porcentaje de trabajo no puede decrecer. Sin embargo, el análisis de los incrementos entre registros consecutivos de la misma serie revela que existe un porcentaje de pasos con retroceso superior a 0.5 puntos porcentuales. Estos retrocesos son inconsistentes con la naturaleza del proceso constructivo y se atribuyen a correcciones de medición, reasignaciones de partidas o errores de entrada de datos. No se eliminan los registros sino que se aplica durante el preprocesado una corrección de monotonía que acumula el máximo visto hasta cada punto temporal, garantizando que la serie resultante sea estrictamente no decreciente.

7.3 ANÁLISIS DE LAS CARACTERÍSTICAS DE OBRAS

7.3.1 VARIABLES NUMÉRICAS

Las variables numéricas disponibles describen fundamentalmente el tamaño y la configuración física de la obra: superficie construida, número de viviendas, plantas sobre rasante, plantas bajo rasante y número de portales o entradas. El análisis de sus distribuciones revela un patrón común a todas ellas: asimetría positiva pronunciada, con la mayoría de las obras concentradas en valores bajos y una cola derecha extendida correspondiente a obras de gran escala. Este patrón es esperable dado que en el sector residencial la distribución natural del tamaño de los proyectos es asimétrica: hay muchas más obras medianas que grandes complejos residenciales.

La asimetría tiene una implicación directa para el modelo de similitud entre obras. Si se utilizan estas variables en su escala original para calcular distancias, las obras grandes dominarán el cálculo porque sus diferencias absolutas son mucho mayores que las de las obras pequeñas, aunque proporcionalmente sean equivalentes. La solución adoptada es aplicar una transformación logarítmica antes de calcular distancias, lo que comprime la escala y hace comparables las diferencias relativas entre obras de distintos tamaños.

La correlación entre superficie construida y número de viviendas es moderada-alta, lo que es esperable porque ambas variables miden indirectamente la misma dimensión: el tamaño del proyecto. Esta redundancia implica que en el modelo de similitud no deben tratarse como dos dimensiones independientes con el mismo peso, porque estarían contando dos veces la misma información y sobrerrepresentarían el tamaño frente a otras características como el tipo constructivo.

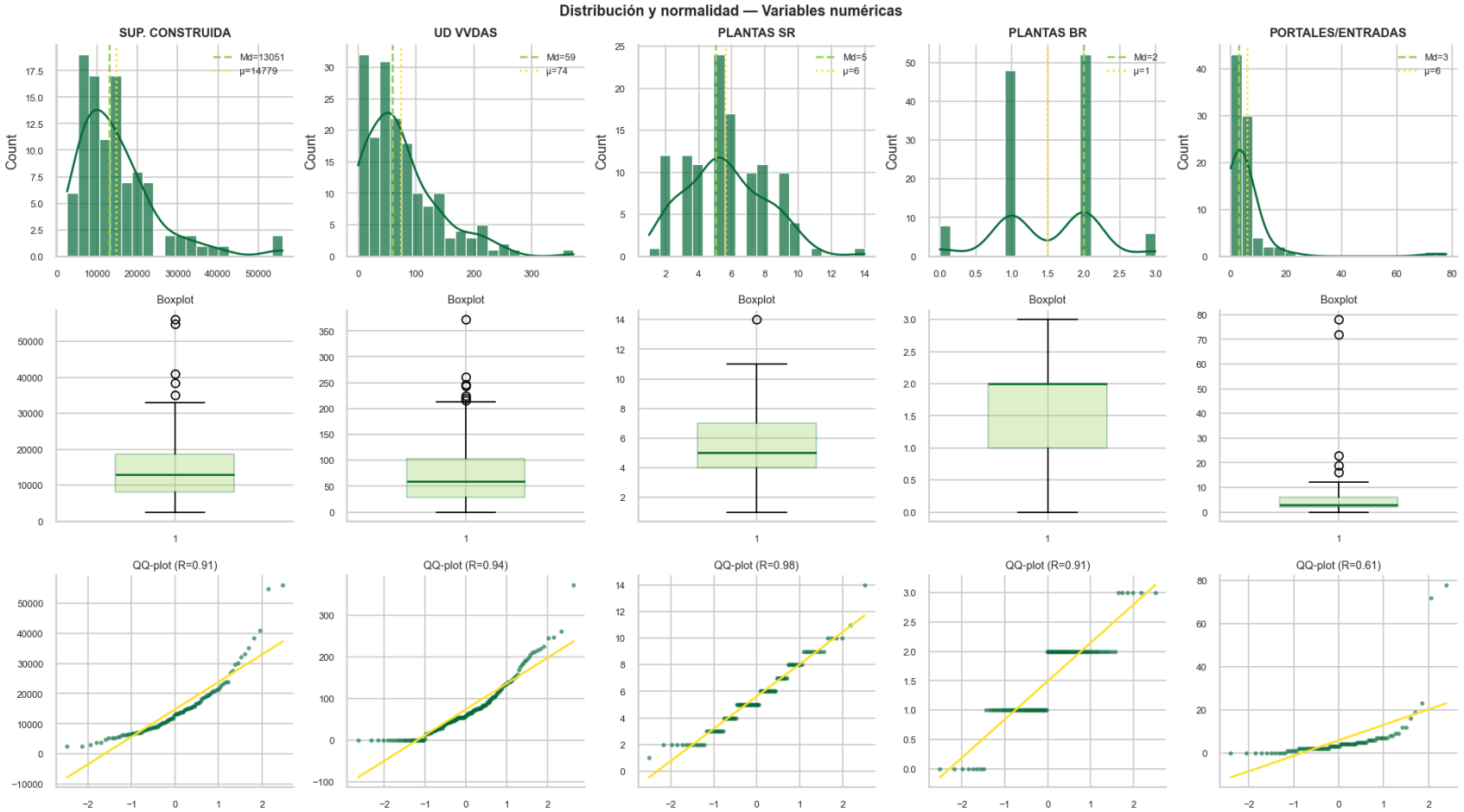


Figura 12 - Distribución y normalidad en las variables numéricas de las obras

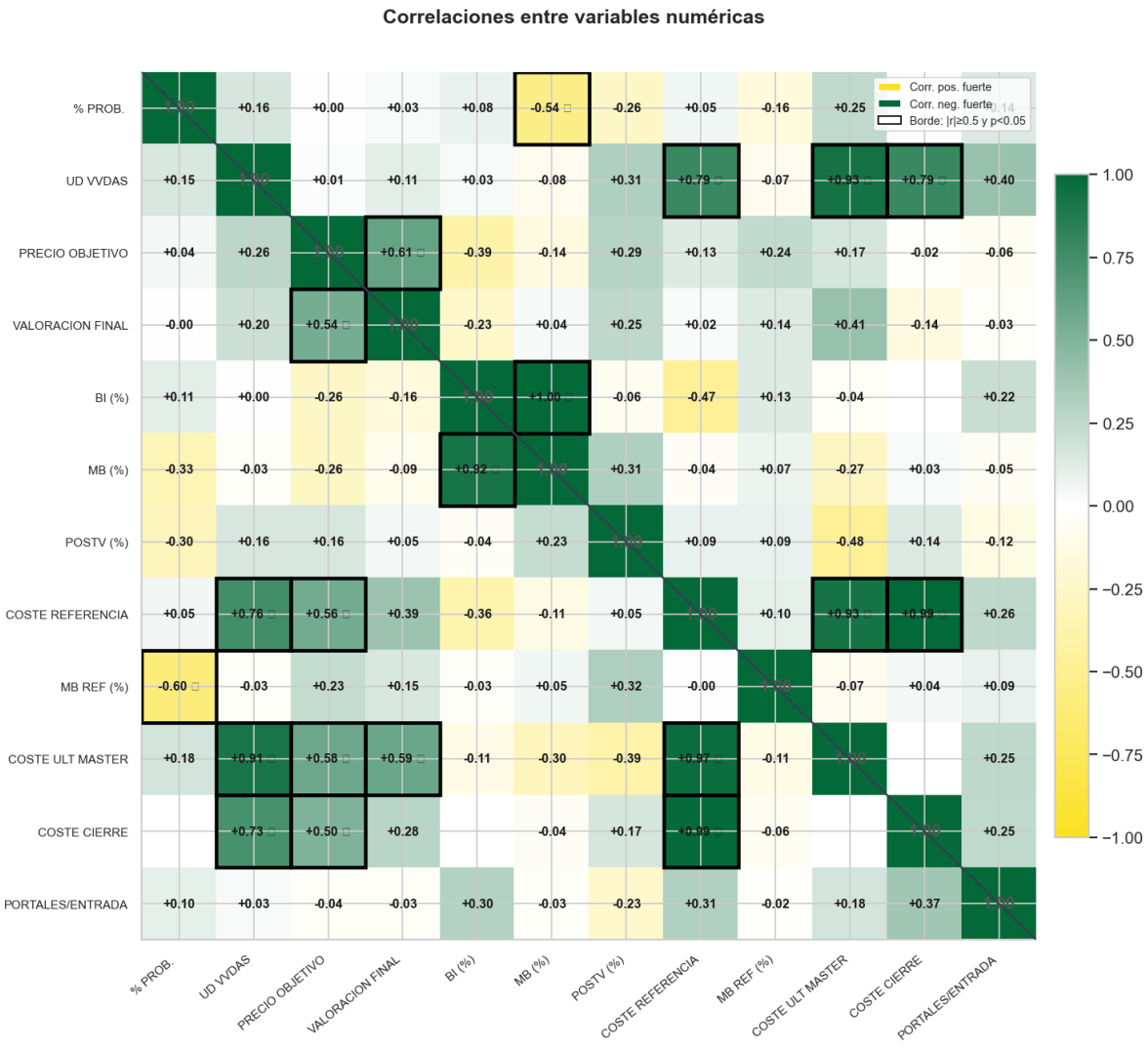


Figura 13 - Heatmap de correlaciones entre variables numéricas

7.3.2 VARIABLES CATEGÓRICAS

Las variables categóricas describen las soluciones constructivas adoptadas en cada obra. Su análisis revela dos problemas con implicaciones distintas para el modelado (Figura 14, Figura 15 y Figura 16). El primero es la alta proporción de la categoría DESCONOCIDO en varias variables, que en algunos casos supera el 50% del total de obras. Estas variables tienen utilidad muy limitada como features discriminativas porque la categoría mayoritaria no aporta información sobre las características reales de la obra. El segundo problema es la concentración excesiva en una única categoría en algunas variables: si el 80% de las obras

comparten la misma solución de cimentación, esa variable apenas discrimina entre proyectos y su aportación al modelo de similitud es marginal.

La métrica más adecuada para cuantificar la utilidad de una variable categórica como feature discriminativa es la entropía de Shannon, que mide la dispersión de la distribución entre sus categorías.

$$E.\ 3 \qquad H(X) = -\sum_{i-1}^{n} p_i \, log_2(p_i)$$

Donde p_i es la proporción de obras en la categoría i y n es el número de categorías de la variable. El resultado se mide en bits.

Una entropía alta indica que las obras se distribuyen de forma relativamente uniforme entre las categorías disponibles, lo que significa que la variable distingue bien entre proyectos distintos. Una entropía baja indica concentración en pocas categorías y por tanto escasa capacidad discriminativa. Las variables con entropía baja y alto porcentaje de DESCONOCIDO se excluyen del modelo de similitud.

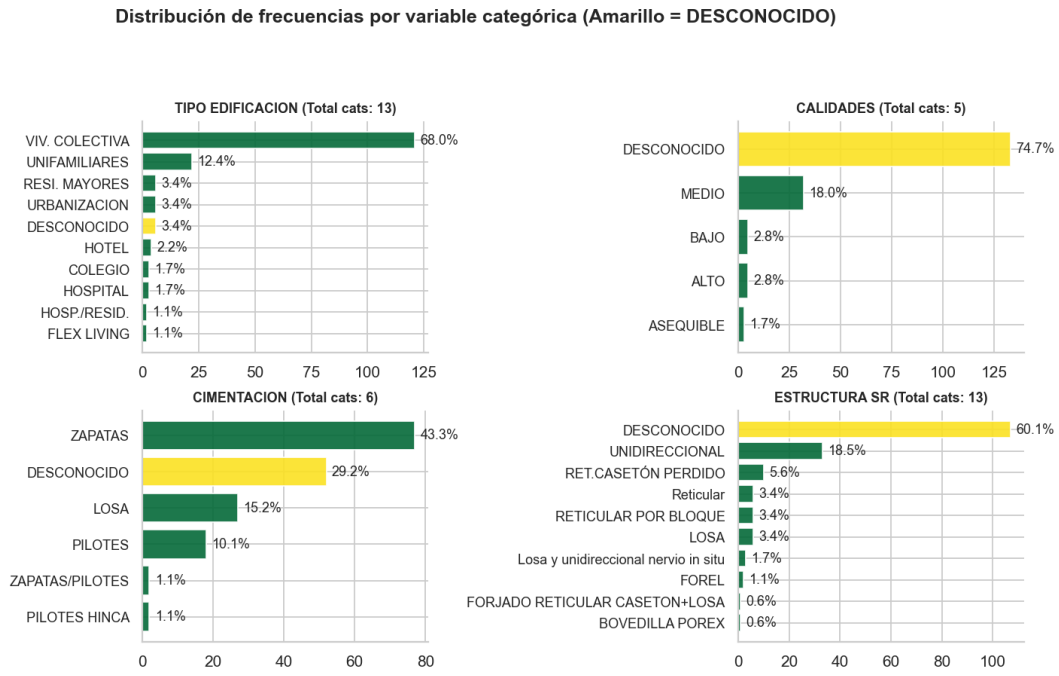


Figura 14 - Distribución de las frecuencias para cada variable categórica I

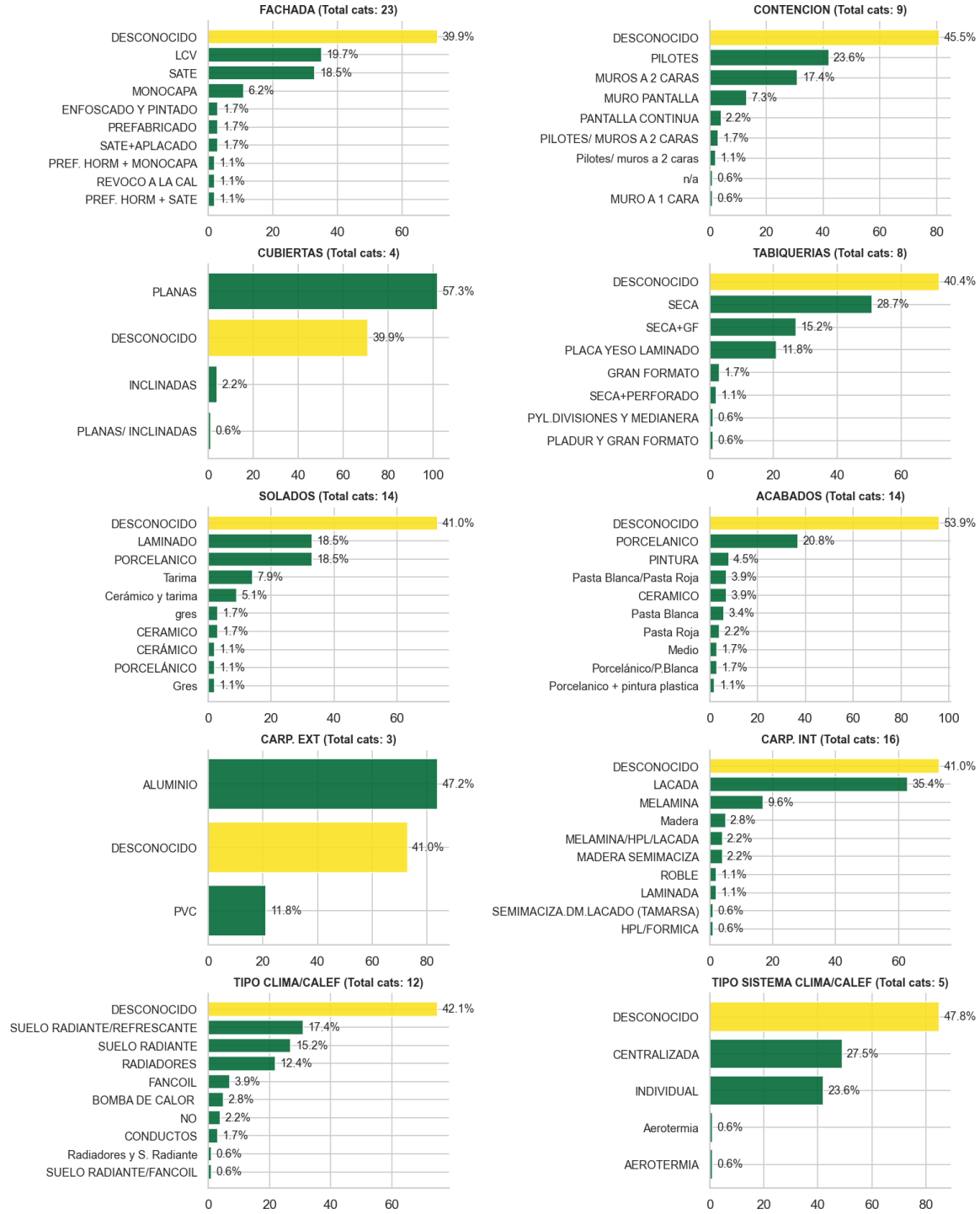


Figura 15 - Distribución de las frecuencias para cada variable categórica II

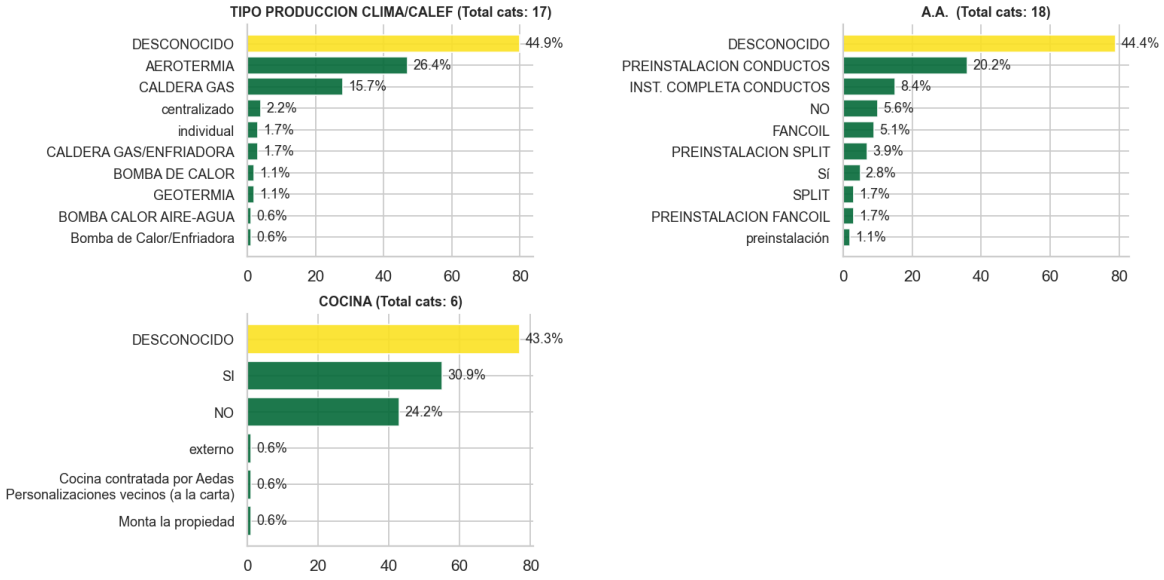


Figura 16 - Distribución de las frecuencias para cada variable categórica III

7.4 ESTRUCTURA TEMPORAL DE LAS CURVAS DE PRODUCCIÓN

7.4.1 DURACIÓN DE LAS OBRAS

La duración de las obras, medida como el intervalo entre el primer y el último registro disponible para cada obra, presenta una distribución con mediana en torno a 20 meses y una dispersión significativa (Figura 17, Figura 18 y Figura 19). Esta variabilidad no es un problema de calidad de datos sino una característica real del negocio: coexisten en el dataset obras de rehabilitación de corta duración con grandes promociones residenciales de varios años. Sin embargo, tiene una consecuencia metodológica fundamental: comparar o promediar curvas en tiempo absoluto carece de sentido cuando una obra dura 12 meses y otra dura 36, porque el mes 10 representa fases constructivas completamente distintas en cada caso. Esta observación es el argumento empírico más directo para justificar la normalización temporal al intervalo [0, 1], que transforma el eje temporal de meses absolutos a porcentaje de tiempo transcurrido sobre la duración total, haciendo las curvas comparables entre sí independientemente de su duración.

También se observa que comienzan más o menos el mismo número de obras cada año. Lo que destaca es la bajada de obras activas actualmente. Esto se puede justificar con una pequeña bajada de sector de la construcción o a faltas de actualización de la base de datos actual, lo que provoca que haya menos registros.

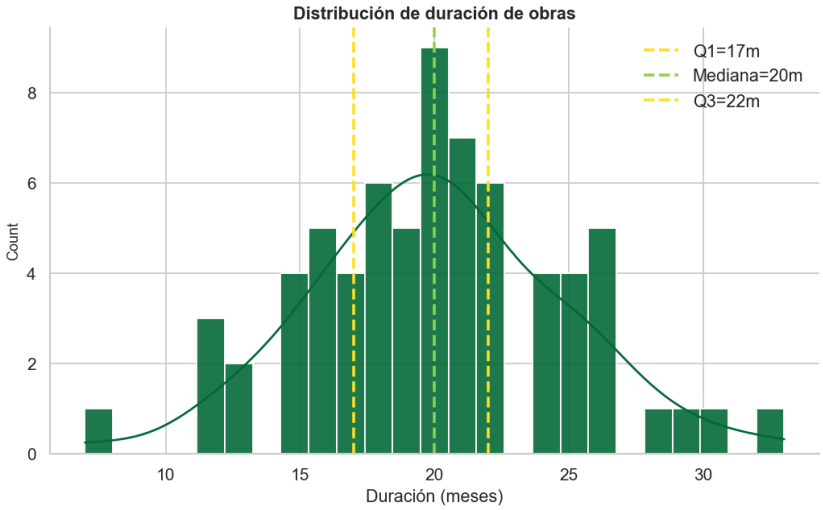


Figura 17 - Duración de las obras

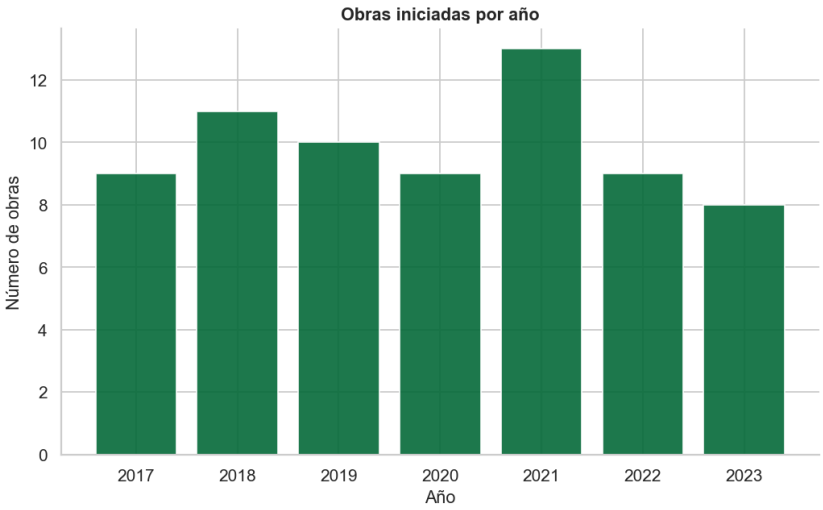


Figura 18 - Obras iniciadas por año

Figura 19 - Obras activas por mes

7.4.2 COBERTURA DE LOS CÓDIGOS DE COMPRA

No todos los códigos de compra aparecen en todas las obras. La cobertura de un código se define como el porcentaje de obras del dataset que tienen al menos un registro de ese código. La distribución de coberturas es fuertemente asimétrica: un subconjunto reducido de códigos aparece en la gran mayoría de las obras, mientras que la mayor parte de los códigos tienen presencia muy escasa, con aparición en menos del 10% de los proyectos (Figura 20).

Esta asimetría tiene consecuencias directas para el modelado. Los códigos con baja cobertura disponen de pocos ejemplos históricos para entrenar, lo que hace que los modelos de machine learning y deep learning sean poco fiables para ellos por falta de datos suficientes. Para estos códigos los modelos heurísticos basados en similitud entre obras son más robustos porque no requieren un volumen mínimo de ejemplos en sentido estricto: se apoyan en las obras más parecidas disponibles independientemente de cuántas sean. Se establece un umbral mínimo de cobertura por debajo del cual el código queda excluido del entrenamiento de los modelos ML y DL, aunque sigue siendo predicho por los modelos heurísticos.

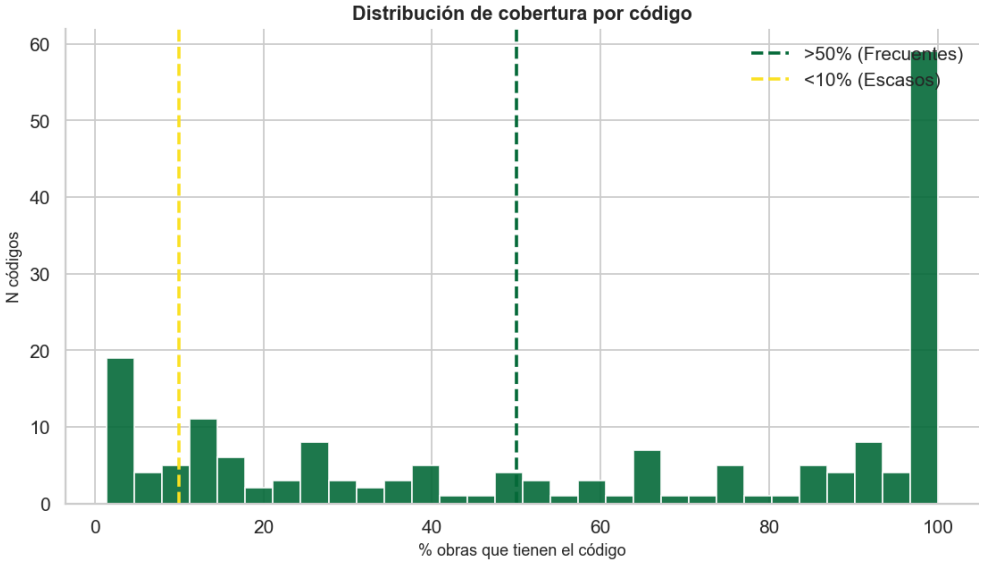


Figura 20 - Distribución de cobertura por código de compra

De esto sacamos la siguiente tabla resumen (Tabla 7). Esta sintetiza la disponibilidad de los datos según el umbral de cobertura exigido. Se observa que el paso de un umbral del 10% al 30% reduce el número de variables de entrada aproximadamente un 20%.

Tabla 7 - Resumen de cobertura de códigos y criterios de filtrado para modelado

Criterio de cobertura (≥%)	*N.º de códigos disponibles*	*Impacto en el modelo*
≥ 5%	162	Alta granularidad, riesgo de ruido
≥ 10%	154	Granularidad estándar
≥ 20%	134	Equilibrio óptimo
≥ 30%	122	Alta representatividad, menor dimensionalidad

7.4.3 DENSIDAD TEMPORAL Y REGULARIDAD DE LAS SERIES

La frecuencia de muestreo de las curvas es aproximadamente mensual, pero no perfectamente regular. El análisis de los gaps temporales entre registros consecutivos de la misma serie revela que una proporción de los intervalos corresponde a dos o tres meses, introduciendo irregularidad en la cadencia de las observaciones. Esta irregularidad es relevante especialmente para el modelo LSTM, que en su formulación estándar asume que los puntos de la secuencia están igualmente espaciados en el tiempo. Cuando los gaps son frecuentes, la solución adoptada es interpolar linealmente los valores faltantes para regularizar la serie antes de su uso en el entrenamiento del modelo secuencial.

Adicionalmente, el número de puntos temporales disponibles por serie varía considerablemente (Figura 21). Las series con menos de cuatro puntos no permiten caracterizar la forma de la curva con fiabilidad y se excluyen del entrenamiento de los modelos ML y DL, aunque se mantienen para los modelos heurísticos y paramétrico que pueden operar con pocos puntos.

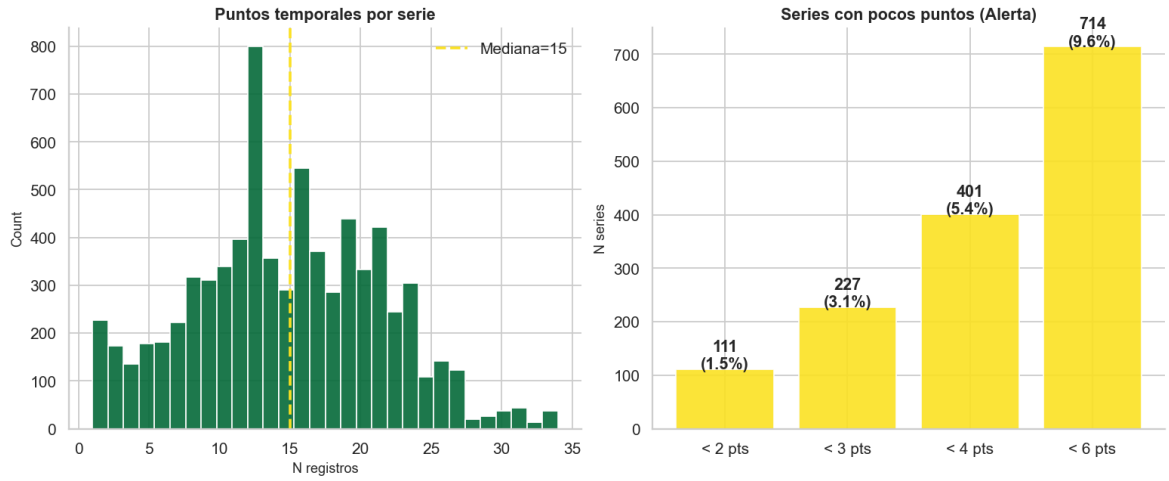


Figura 21 - Puntos temporales por serie (izquierda) y series con pocos puntos (derecha)

7.5 FORMA DE LAS CURVAS Y VALIDACIÓN DE LA HIPÓTESIS LOGÍSTICA

Este apartado constituye el núcleo analítico del EDA porque aborda directamente la pregunta que da sentido al diseño del modelo paramétrico: ¿siguen realmente las curvas de producción por código de compra una forma S logística, o esa hipótesis es únicamente teórica?

7.5.1 AJUSTE LOGÍSTICO Y DISTRIBUCIÓN DEL R^2

Para responder esta pregunta de forma empírica se ajustó una curva logística estándar a cada serie obra-código normalizada temporalmente mediante mínimos cuadrados no lineales. La función logística tiene tres parámetros con interpretación directa en el dominio constructivo. El parámetro L es el nivel máximo de avance alcanzado, que en condiciones ideales es 100. El parámetro k controla la velocidad o pendiente de crecimiento: valores altos producen una curva S muy pronunciada donde casi toda la producción se concentra en un periodo corto, mientras que valores bajos producen una curva más gradual. El parámetro x0 es el punto de inflexión, es decir, el momento del ciclo de obra expresado en tiempo normalizado en que la velocidad de producción es máxima y a partir del cual comienza a decrecer.

La bondad del ajuste se evaluó mediante el coeficiente de determinación R^2, que mide la proporción de la varianza del avance acumulado explicada por la curva logística ajustada. Un R^2 superior a 0.9 indica que el modelo logístico captura más del 90% de la variabilidad observada, lo que constituye un ajuste excelente para datos reales de obra.

El resultado empírico muestra que aproximadamente el 83,5% de las series obtienen un R^2 superior a 0.9 y el 92,3% superan 0.8. Este resultado valida la hipótesis teórica con datos reales: la forma S logística no es solo un supuesto teórico sino una descripción estadísticamente sólida del comportamiento real de la producción en la mayoría de los códigos y obras analizados. Las series con R^2 bajo corresponden mayoritariamente a tres patrones identificables: curvas con parones prolongados donde la obra se detiene y reanuda, comportamientos bimodales con dos picos de producción separados, y curvas de arranque muy tardío donde el código no se ejecuta hasta el último tramo de la obra. Estos casos no

invalidan la hipótesis general sino que identifican las excepciones estructurales que ningún modelo logístico simple puede capturar.

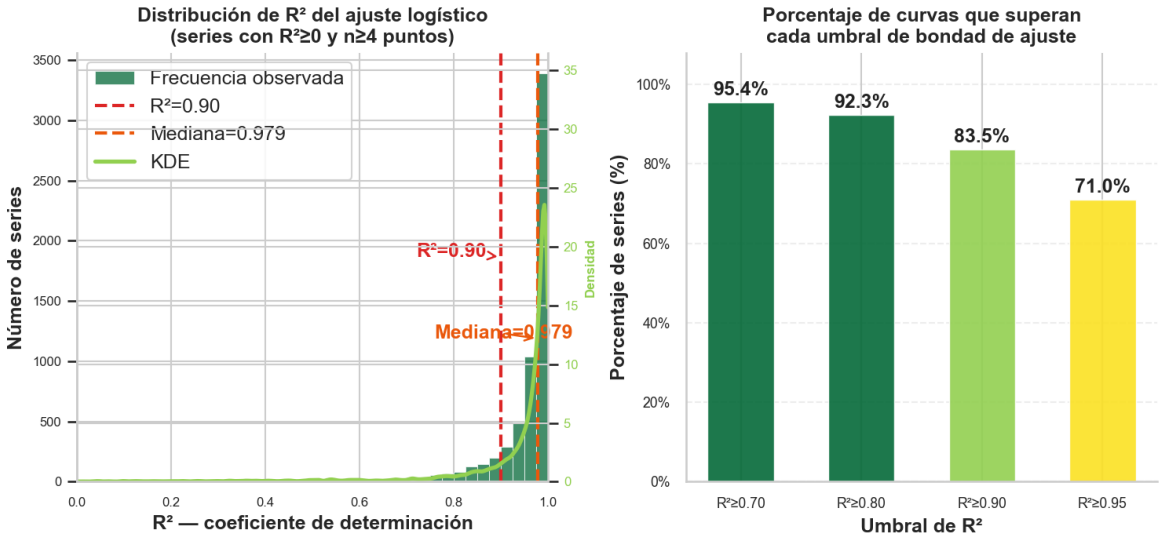


Figura 22 - Distribución del R^2

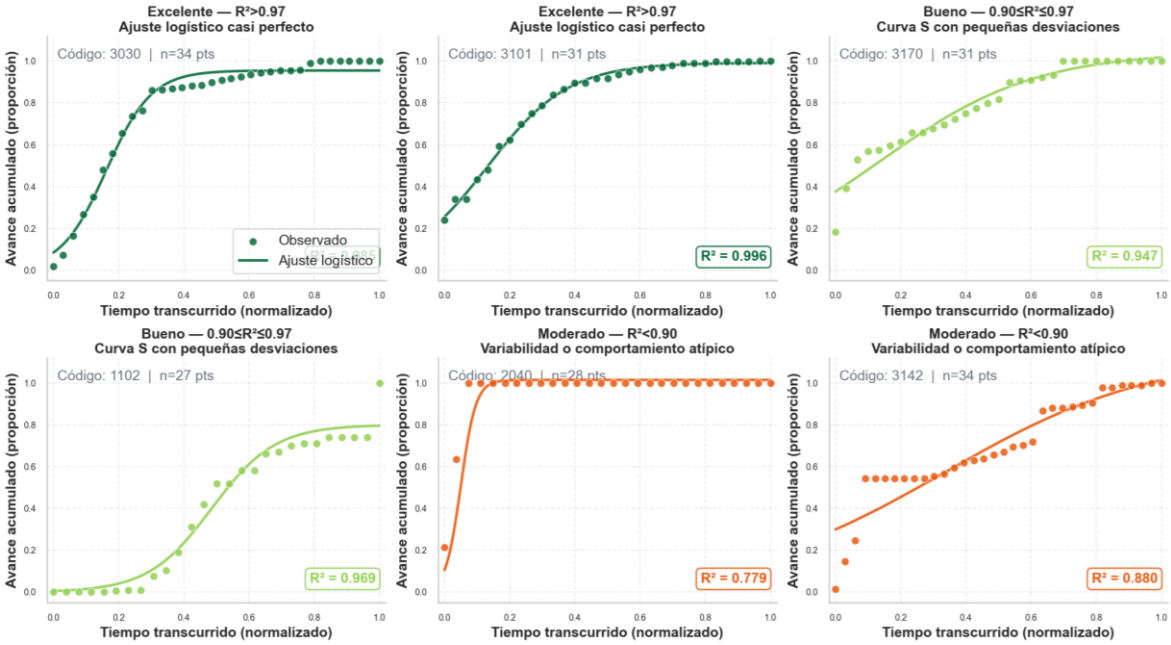


Figura 23 - Ejemplos de ajuste logístico por cada rango de R^2

7.5.2 DISTRIBUCIÓN E INTERPRETACIÓN DE LOS PARÁMETROS LOGÍSTICOS

Los parámetros del ajuste no son únicamente métricas de bondad sino que constituyen una representación compacta de la forma de cada curva y se convierten en variables objetivo intermedias para el modelo paramétrico: en lugar de predecir directamente la curva punto a punto, el modelo paramétrico predice k y x0 a partir de las características de la obra y reconstruye la curva analíticamente.

El punto de inflexión x0 tiene una mediana en torno al 0,28 (28% del tiempo total de la obra), con una dispersión que revela heterogeneidad real entre proyectos. Obras con x0 bajo tienen una producción concentrada en la primera mitad, lo que es típico de partidas de estructura o movimientos de tierra que se ejecutan al inicio. Obras con x0 alto tienen un arranque tardío con producción concentrada en la segunda mitad, característico de partidas de acabados e instalaciones que dependen de que la estructura esté terminada.

La pendiente k muestra también variabilidad considerable. Esta variabilidad no es ruido sino información: diferentes tipos de obra y diferentes códigos de compra tienen ritmos de producción naturalmente distintos, y k captura cuantitativamente esa diferencia. La distribución de k por código de compra, analizada en apartados anteriores, revela que ciertos códigos tienen pendientes sistemáticamente más altas que otros, lo que indica que son partidas que se ejecutan de forma intensa y concentrada independientemente de la obra.

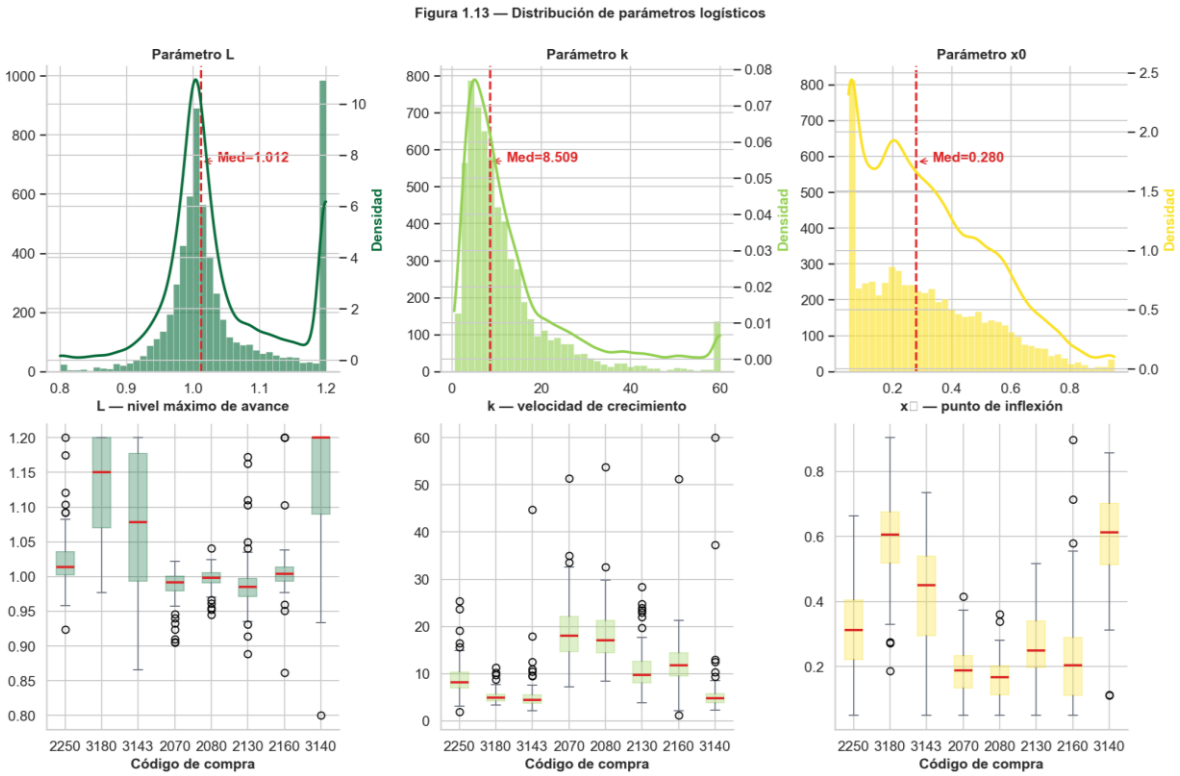


Figura 24 - Distribución de parámetros logísticos

7.5.3 VISUALIZACIÓN DE LAS CURVAS NORMALIZADAS

La superposición de todas las curvas normalizadas de un mismo código es la visualización más intuitiva del EDA porque comunica simultáneamente el patrón central compartido y la variabilidad existente entre obras. Cada obra se representa como una línea individual semitransparente, lo que permite ver tanto la densidad del conjunto como los casos atípicos que se alejan del patrón central.

La interpretación de estas figuras es directa para el modelado. Un código cuyas curvas se agrupan de forma compacta (indicado por un bajo coeficiente de variación CV) sugiere que el patrón central es representativo para la mayoría de los casos, permitiendo el uso de un modelo heurístico. Por el contrario, un código con curvas dispersas (alto CV y bandas P10-P90 amplias) evidencia que la variabilidad es intrínseca, obligando al modelo a condicionar sus predicciones sobre otras variables del proyecto para capturar esa heterogeneidad.

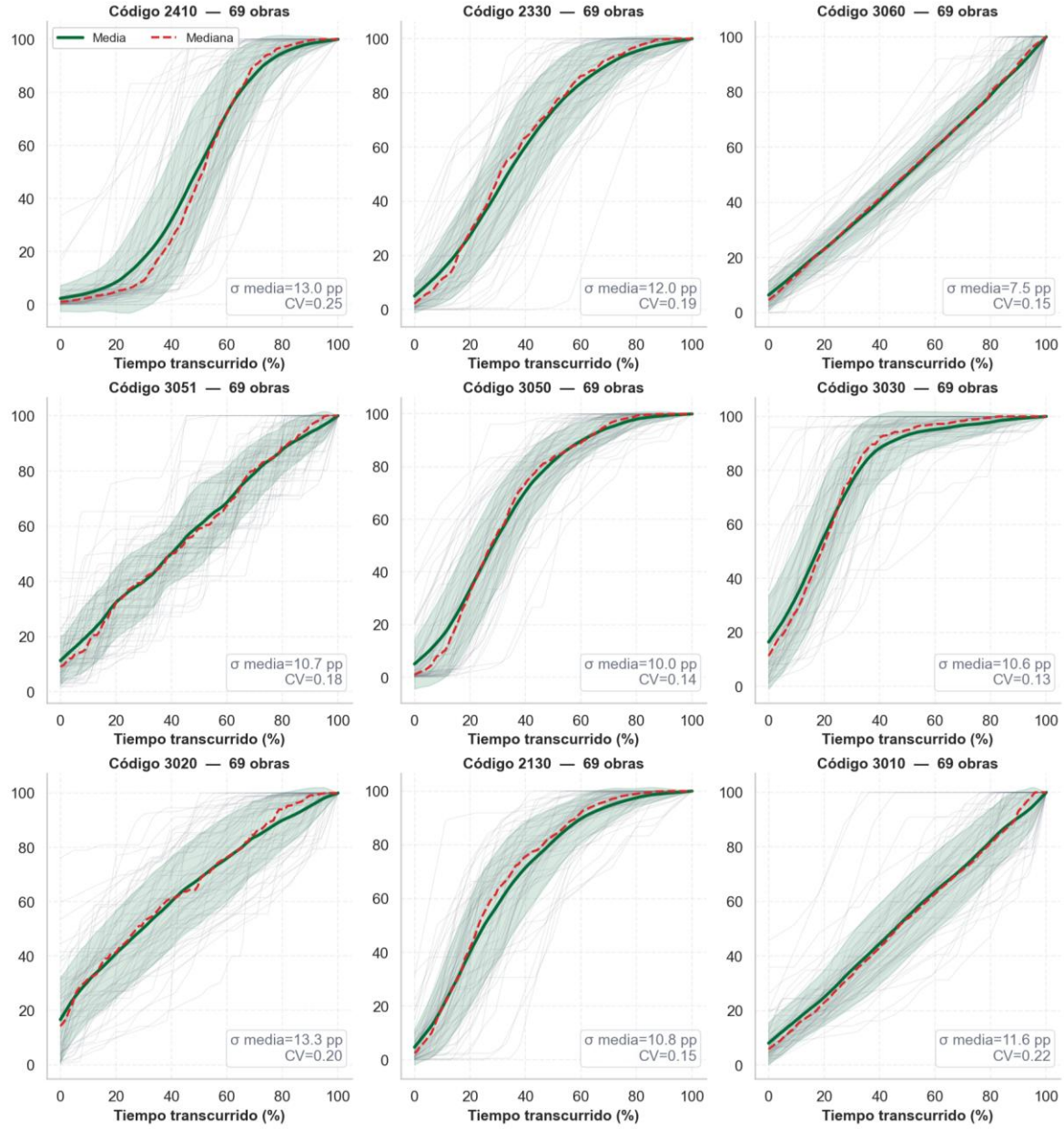


Figura 25 - Ejemplos de curvas de producción normalizadas

7.6 ARQUETIPOS DE CURVA S

La variabilidad observada en las curvas normalizadas plantea una pregunta natural: ¿es esa variabilidad continua y sin estructura, o existen grupos naturales de curvas con comportamientos diferenciados? Si existen grupos, identificarlos tiene valor tanto descriptivo, porque permiten caracterizar los distintos patrones de producción del negocio, como predictivo, porque el arquetipo al que pertenece una curva puede usarse como feature adicional en los modelos supervisados.

Para responder esta pregunta se aplicó K-Means sobre las curvas normalizadas representadas como vectores de 100 puntos en la grilla temporal [0, 1]. Esta representación vectorial hace que cada curva sea un punto en un espacio de 100 dimensiones, y K-Means agrupa los puntos más cercanos en el sentido euclídeo, es decir, las curvas con formas más similares.

La selección del número óptimo de clusters $K = 2$ se realizó mediante dos criterios complementarios que se evalúan conjuntamente porque cada uno tiene limitaciones individuales (Figura 26). El método del codo analiza la inercia intra-cluster en función de K y busca el punto donde la reducción marginal al añadir un cluster adicional deja de ser proporcional. Este método es intuitivo pero subjetivo cuando el codo no es pronunciado. El silhouette score complementa esta evaluación midiendo simultáneamente la cohesión intra-cluster, es decir, cuán similares son las curvas dentro de cada grupo, y la separación inter-cluster, cuán diferentes son los grupos entre sí. Su máximo indica $K = 2$, donde los clusters son más compactos internamente y más separados entre ellos, y es una métrica objetiva que no requiere interpretación visual.

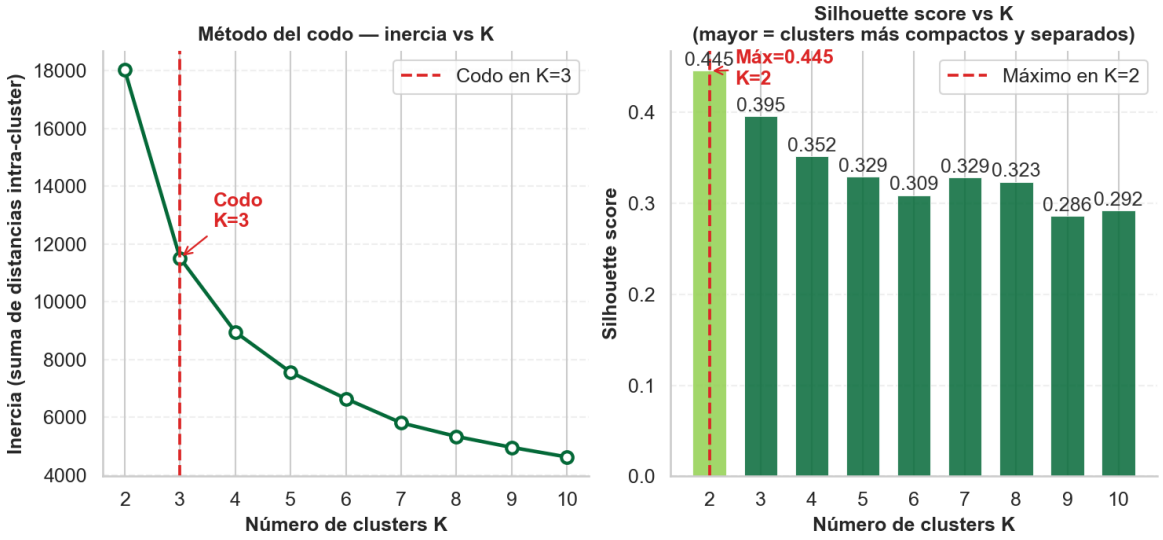


Figura 26 - Método del codo y silhouette score

Los K arquetipos identificados tienen interpretación directa en términos del proceso constructivo (Figura 27). Las curvas del primer arquetipo muestran un arranque lento seguido de una aceleración intensa en la segunda mitad de la obra, patrón característico de partidas que dependen de fases previas como los acabados interiores o las instalaciones. Las curvas del segundo arquetipo presentan una distribución más uniforme de la producción a lo largo del tiempo, típica de partidas que avanzan en paralelo con el progreso general de la obra. Las curvas del tercer arquetipo, si K lo permite, concentran la producción en la fase inicial, características de trabajos de obra civil, movimiento de tierras o estructura que se ejecutan antes de que comience el resto del proyecto. Esta interpretación semántica de los clusters es una de las aportaciones originales del trabajo porque conecta el análisis estadístico con el conocimiento del dominio constructivo.

La distribución de los arquetipos varía entre códigos de compra, lo que indica que los patrones no son aleatorios sino que responden a la naturaleza de cada partida presupuestaria. Que un código tenga el 80% de sus curvas en el arquetipo de arranque tardío es información valiosa para el modelado: indica que ese código tiene un comportamiento sistemático que puede predecirse con fiabilidad una vez identificado el arquetipo.

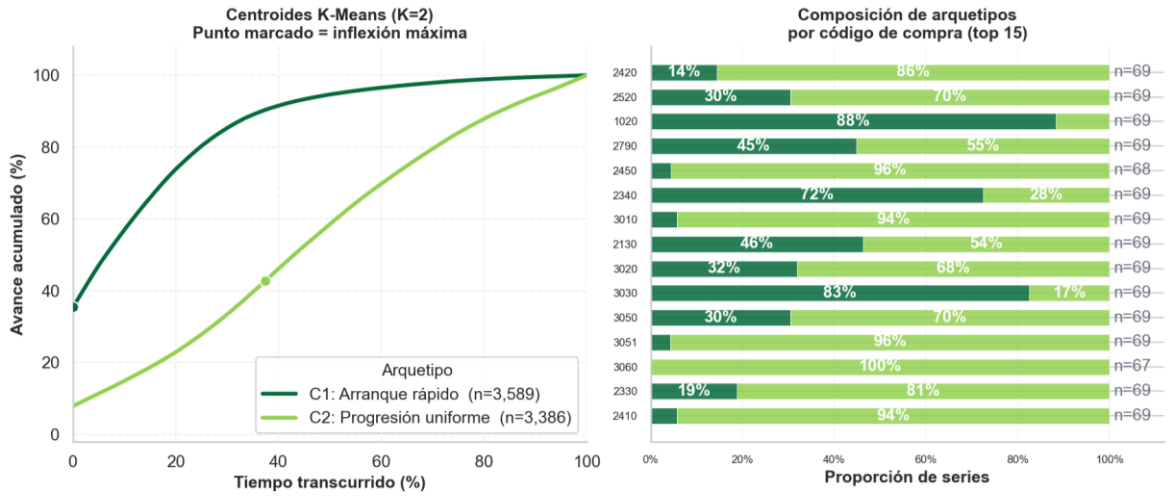


Figura 27 - Centroides y composición por código de compra

Adicionalmente se realizó un análisis de clustering jerárquico con enlace de Ward sobre una muestra aleatoria de curvas, cuyo dendrograma permite verificar la estructura de agrupamiento y confirmar que los clusters identificados por K-Means corresponden a agrupamientos naturales en los datos y no son un artefacto del algoritmo (Figura 28).

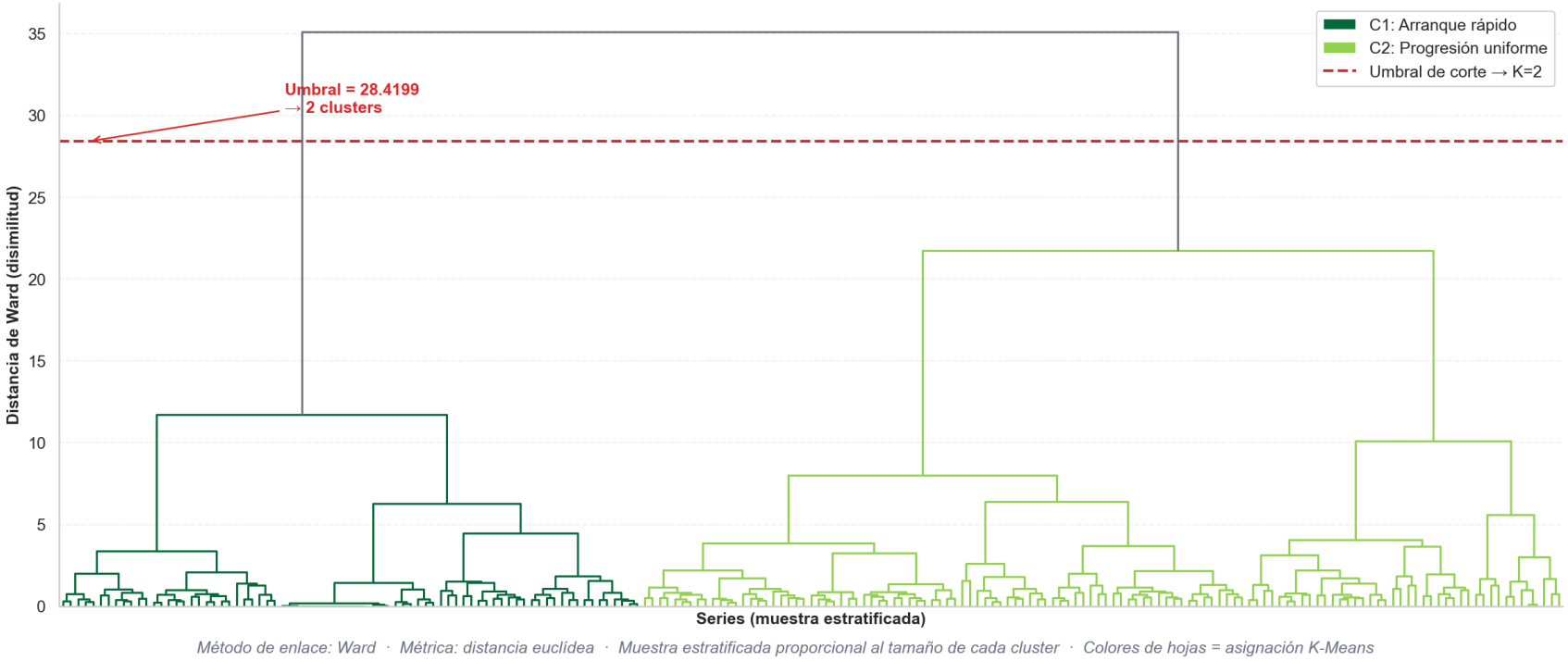


Figura 28 - Dendrograma jerárquico de 199 curvas y clústeres K-Means (K=2)

7.7 ANÁLISIS CRUZADO: CARACTERÍSTICAS DE OBRA VS FORMA DE CURVA

Este apartado es el puente analítico entre el EDA descriptivo y el diseño de los modelos supervisados. Su objetivo es identificar qué variables de características de obra tienen una relación estadísticamente significativa con la forma de la curva de producción, operacionalizada a través de los parámetros logísticos k y x0 obtenidos en el apartado 7.5. Si las características de una obra explican parte de la variabilidad en la forma de su curva, entonces los modelos que incorporan esas características como input deberían predecir mejor que los modelos heurísticos que las ignoran.

7.7.1 PARÁMETROS LOGÍSTICOS VS VARIABLES CATEGÓRICAS

Para cada combinación de variable categórica y parámetro logístico se aplicó un test ANOVA unidireccional, que contrasta si la media del parámetro difiere significativamente entre las categorías de la variable. Un p-valor inferior a 0.05 indica que esa variable categórica produce diferencias estadísticamente significativas en la forma de la curva y es por tanto informativa para el modelado supervisado.

Los resultados muestran que el tipo de edificación es la variable con mayor poder discriminativo sobre ambos parámetros. Las diferencias en el punto de inflexión x0 entre tipos de edificación tienen sentido constructivo intuitivo: las obras de uso residencial, terciario e industrial tienen secuencias de ejecución distintas porque sus partidas tienen pesos y dependencias diferentes, y eso se refleja en cuándo se concentra la mayor parte de la producción. Las diferencias en la pendiente k indican que el ritmo de producción también varía sistemáticamente entre tipos de obra, independientemente del código de compra específico que se esté analizando.

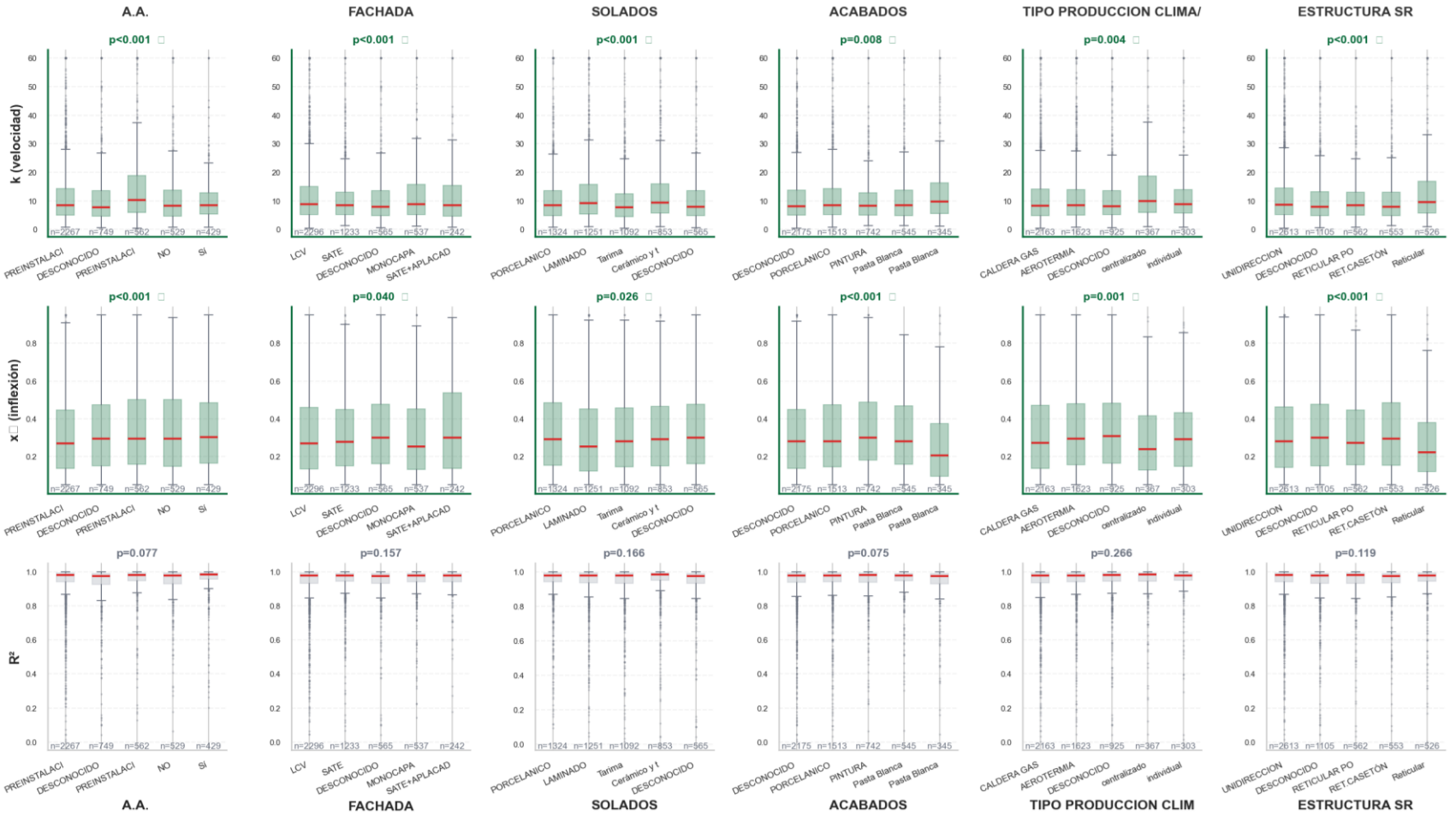


Figura 29 - Parámetros k, x0 y R² de las seis variables categóricas con mayor entropía identificadas en apartados anteriores

7.7.2 PARÁMETROS LOGÍSTICOS VS VARIABLES NUMÉRICAS

La relación entre los parámetros logísticos (k y x0) y las variables numéricas de características se analiza mediante scatter plots con línea de regresión y coeficiente de correlación de Pearson. Dado que algunas de estas variables presentan alta asimetría, se aplicó una escala logarítmica en el eje X para normalizar su visualización y facilitar la interpretación de las tendencias.

Las correlaciones observadas son en general moderadas, lo que indica que las variables numéricas de tamaño de obra tienen una relación no despreciable pero no dominante con la forma de la curva. Esto sugiere que el tamaño de la obra condiciona parcialmente cómo evoluciona la producción pero no la determina de forma determinista, lo que es coherente con la intuición de que obras de tamaño similar pueden ejecutarse con ritmos muy distintos dependiendo de otros factores como la organización del contratista o las condiciones del terreno.

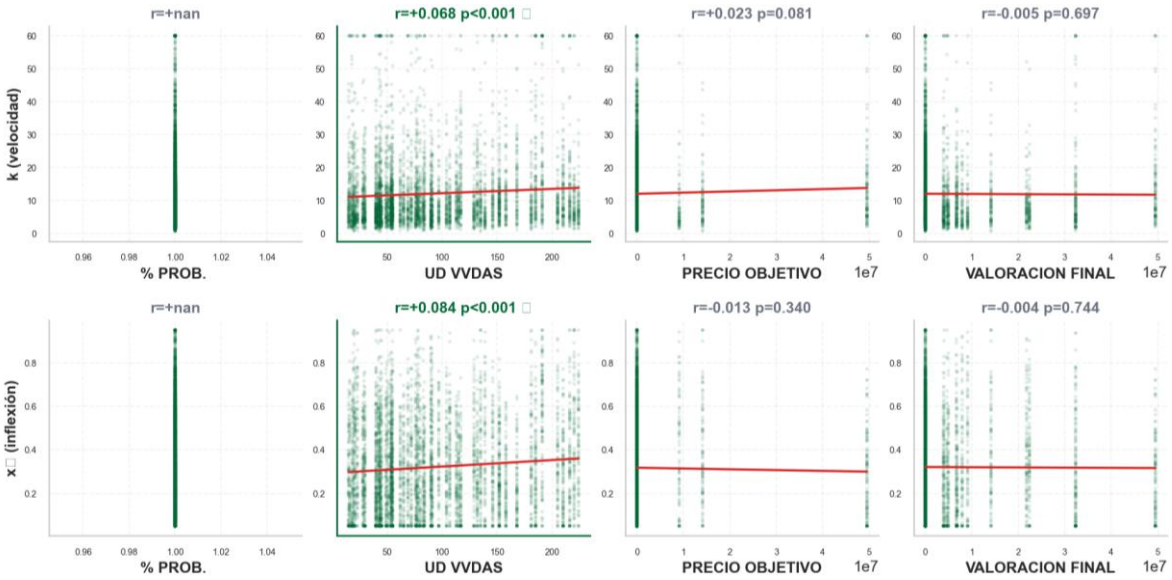


Figura 30 - k y x0 de las variables numéricas disponibles I

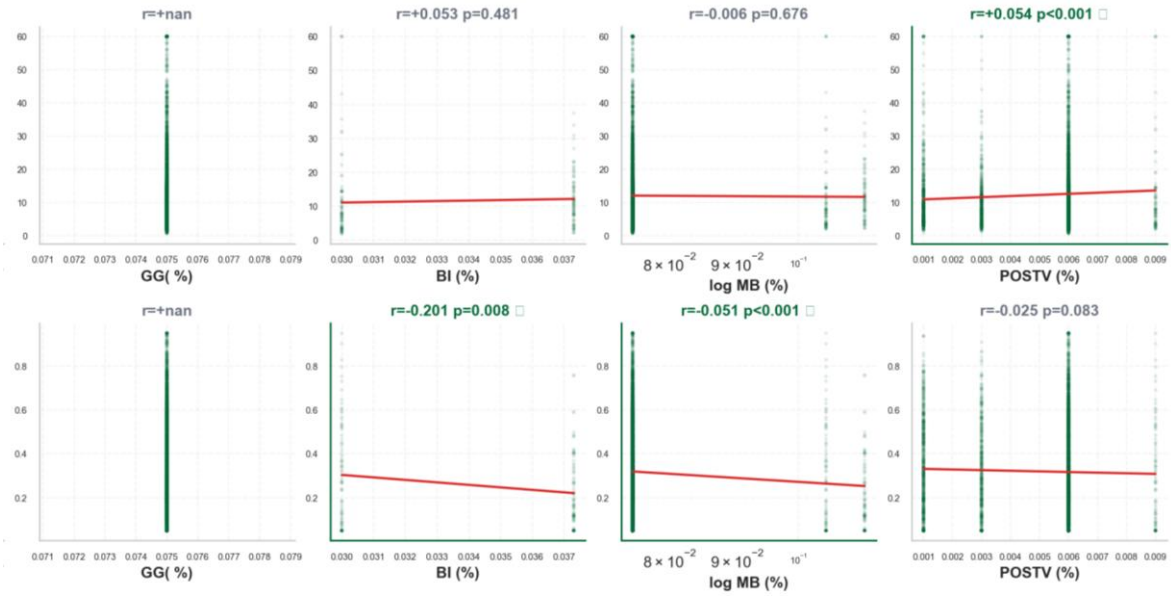


Figura 31 - k y x0 de las variables numéricas disponibles II

7.7.3 ASOCIACIÓN ENTRE VARIABLES CATEGÓRICAS

El análisis de asociación entre pares de variables categóricas mediante Cramér's V identifica qué pares de variables son redundantes entre sí. Esta información es relevante para el modelo de similitud entre obras: usar dos variables con alta asociación entre ellas equivale a contar la misma dimensión del proyecto dos veces, lo que sesga el cálculo de distancias hacia esa dimensión en detrimento de las demás.

Detectar estas dependencias es un paso crítico en el diseño del modelo, ya que permite seleccionar un conjunto de features ortogonales o no correlacionadas, garantizando que el cálculo de similitud sea equilibrado y representativo de la totalidad de las características de la obra.

En la Figura 32 se observa una alta densidad de asociaciones fuertes (cuadrantes de color oscuro con borde negro), especialmente en variables como Cubiertas, Tabiquerías Y Carpintería. Esto sugiere que, en el contexto de tus datos, estas variables son altamente dependientes entre sí (por ejemplo, el tipo de clima suele condicionar la carpintería o el tipo

de instalaciones). Esto valida que el proceso de selección de variables será vital para evitar la multicolinealidad en los modelos predictivos futuros.

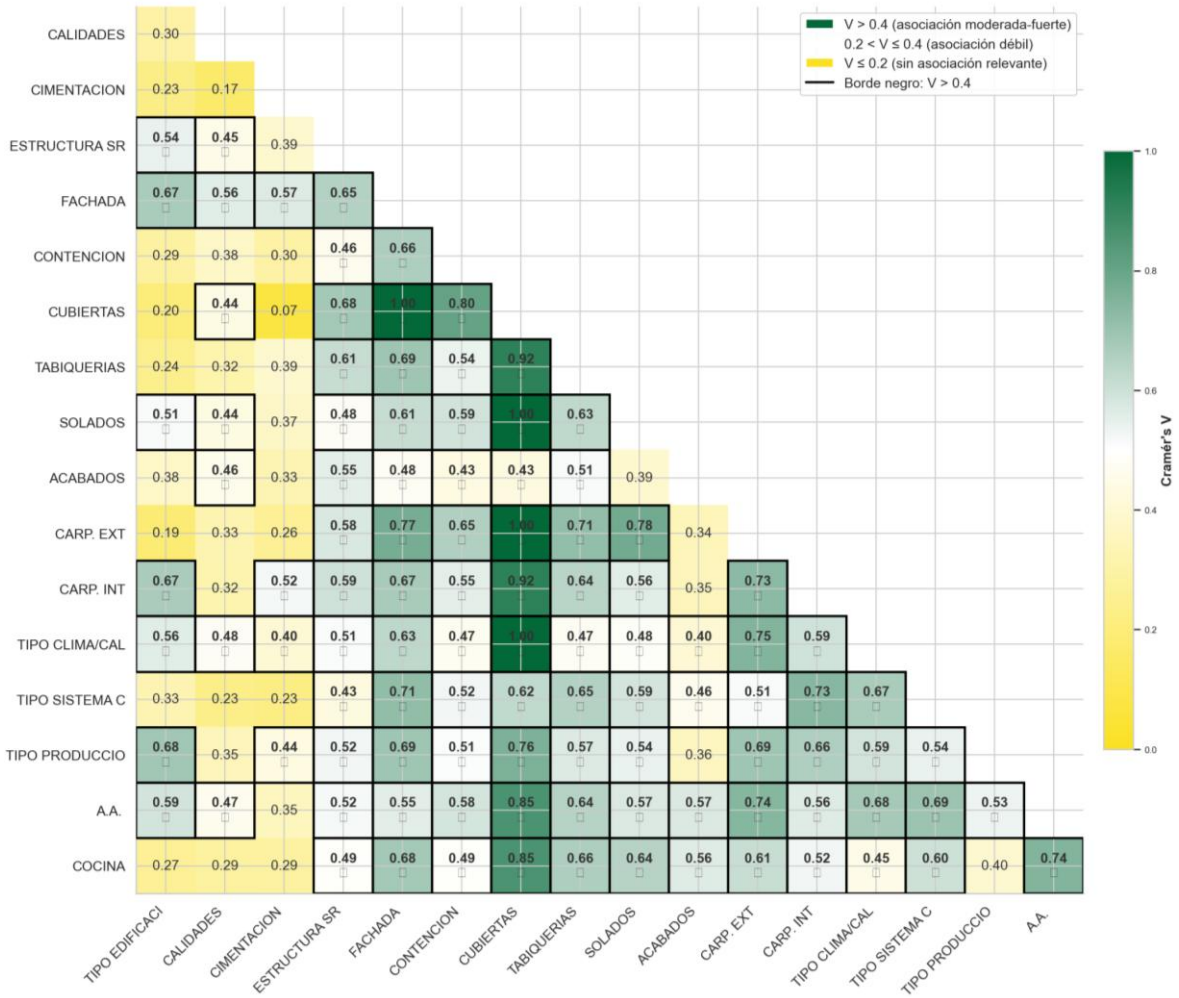


Figura 32 - Heatmap triangular inferior con los valores de Cramér entre todas las variables categóricas

7.7.4 ESTACIONALIDAD

El sector de la construcción tiene patrones estacionales bien documentados: reducción de actividad en agosto por el periodo vacacional y en diciembre-enero por festividades y condiciones climáticas adversas en determinadas regiones. Para verificar si estos patrones son visibles en los datos se analizó el incremento mensual de avance en función del mes del año sobre todo el dataset.

El análisis confirma la presencia de un efecto estacional consistente con el conocimiento del dominio. Los meses de agosto y diciembre muestran incrementos medianos inferiores al resto del año de forma persistente a lo largo de los distintos años del dataset. El efecto es moderado en magnitud pero suficientemente consistente en el tiempo como para justificar la inclusión del mes del año como variable de entrada en los modelos de machine learning y deep learning que generan predicciones en tiempo real, donde conocer el mes en que se hace la predicción puede mejorar la estimación del próximo incremento.

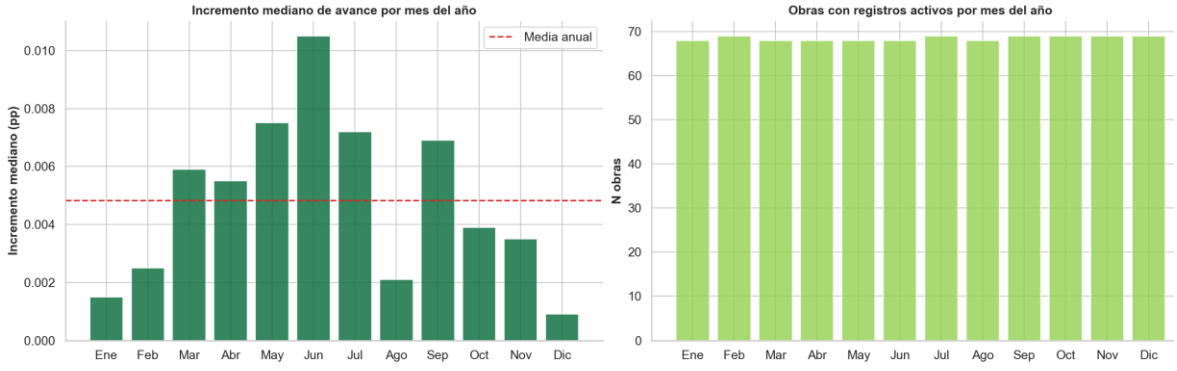


Figura 33 - Incremento mediano de avance por mes (izquierda) y obras con registros activos por mes (derecha)

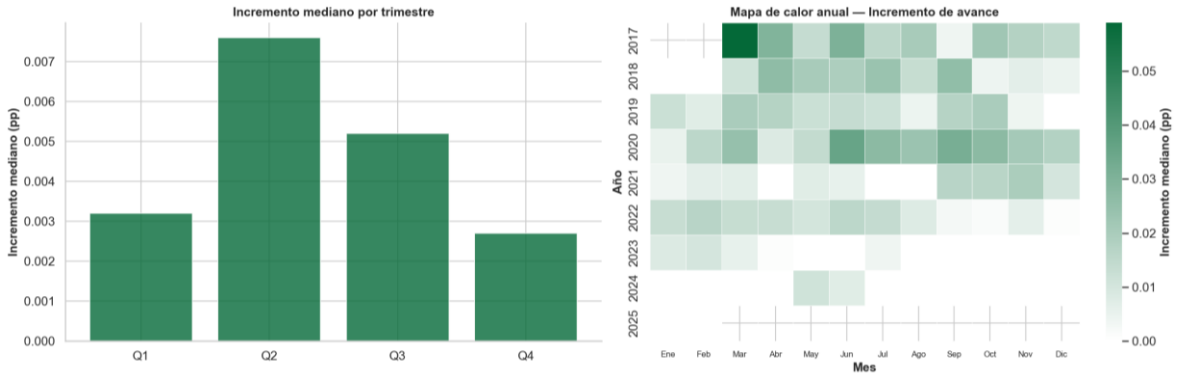


Figura 34 - Incremento mediano por trimestre (izquierda) e incremento de avance (derecha)

7.8 RESUMEN DE HALLAZGOS E IMPLICACIONES PARA EL MODELADO

El análisis exploratorio realizado permite extraer un conjunto de conclusiones concretas que estructuran el diseño metodológico del resto del trabajo. Se presentan organizadas por área temática con su implicación directa para el modelado.

Respecto a la calidad de los datos, la presencia de nulos semánticos, duplicados funcionales, valores fuera de rango e incrementos negativos requiere un pipeline de preprocesado explícito que se ejecute antes de cualquier entrenamiento. Este pipeline incluye la estandarización de nulos, la deduplicación conservando el último registro, el clipping de la variable objetivo al rango [0, 100] y la corrección de monotonía por acumulación del máximo.

Respecto a la normalización temporal, la variabilidad en la duración de las obras hace obligatorio transformar el eje temporal de meses absolutos a porcentaje de tiempo transcurrido. Sin esta normalización, la comparación y promediación de curvas carece de sentido estadístico.

Respecto a la hipótesis logística, la validación empírica confirma que el 83,5% de las series tienen un R^2 superior a 0.9 en el ajuste logístico. Este resultado justifica incluir el modelo paramétrico M6 como uno de los seis enfoques de predicción y utilizarlo como referencia de calidad para los demás modelos en el subconjunto de curvas bien ajustadas.

Respecto a la variabilidad intra-código, los códigos con baja variabilidad son bien predichos por el heurístico de media ponderada y no requieren modelos más complejos. Los códigos con alta variabilidad necesitan modelos que condicionen la predicción sobre las características de la obra para reducir el error. Esta diferencia justifica evaluar los modelos no solo globalmente sino también por código y por arquetipo.

Respecto a las features relevantes, el tipo de edificación es la variable con mayor poder discriminativo sobre la forma de la curva según el análisis ANOVA. La estacionalidad

mensual es moderada pero consistente y justifica incluir el mes como feature en los modelos ML y DL. Las variables numéricas tienen correlaciones moderadas con los parámetros logísticos y deben transformarse logarítmicamente antes de usarse en métricas de distancia.

Respecto al baseline de referencia, el RMSE de la media leave-one-out por código establece el umbral mínimo que cualquier modelo más complejo debe superar para justificar su uso. Un modelo que no mejora este baseline no aporta valor respecto al heurístico más simple.

Capítulo 8. MODELOS DE PREDICCIÓN DE LAS CURVAS S

En este capítulo se describen los seis modelos de predicción implementados para estimar la evolución del avance acumulado por código de compra a lo largo del ciclo de vida de una obra. Para cada modelo se presenta primero una explicación del enfoque metodológico, incluyendo su fundamento teórico y sus propiedades principales, y después se describe cómo se ha adaptado al problema concreto de predicción de curvas S, justificando las decisiones de diseño tomadas. El código completo de cada implementación se recoge en el Anexo correspondiente.

Todos los modelos comparten el mismo formato de entrada y salida para garantizar la comparabilidad de los resultados. La entrada es el vector de características estáticas de la obra y, cuando está disponible, la porción observada de la curva de producción hasta el momento de la predicción. La salida es siempre una curva de 100 puntos uniformemente distribuidos sobre la grilla temporal normalizada [0, 1], que representa la evolución completa del avance acumulado desde el inicio hasta la finalización del código de compra.

Todas las curvas históricas se normalizan temporalmente antes de ser usadas por cualquier modelo, transformando el eje temporal de fechas absolutas a porcentaje de tiempo transcurrido sobre la duración total de cada obra. Esta normalización es obligatoria dada la variabilidad en la duración de las obras documentada en el EDA, y permite comparar y promediar curvas de proyectos con duraciones muy distintas. Adicionalmente, se filtran las series con menos de cuatro puntos temporales por ser insuficientes para caracterizar la forma de la curva, y se excluyen los códigos con menos de cinco obras históricas disponibles para garantizar un mínimo de datos de referencia.

8.1 M1 - MEDIA PONDERADA

El modelo de media ponderada es el enfoque heurístico más simple posible y el primero en implementarse precisamente porque su función no es solo predictiva sino metodológica: establece el umbral mínimo de rendimiento que cualquier modelo más complejo debe superar para justificar su mayor coste de implementación. Su principio formaliza matemáticamente el proceso que seguía el departamento de estudios cuando estimaba manualmente la curva de una obra nueva: buscar proyectos anteriores similares y usar su comportamiento histórico como referencia.

Para cada obra nueva, el modelo calcula su similitud con todas las obras históricas del mismo código de compra y genera la predicción como la media ponderada de sus curvas. La similitud entre dos obras se mide como la inversa de la distancia euclídea entre sus vectores de características, de forma que las obras más parecidas reciben más peso en el promedio:

$$E.\ 4 \qquad \hat{y}(t) = \frac{\sum_{i-1}^{N} w_i \cdot y_i(t)}{\sum_{i-1}^{N} w_i}, \quad w_i = \frac{1}{1+d(x_{target},\ x_i)}$$

donde $y_i(t)$ es el avance de la obra histórica i en el punto temporal t, x_{target} y $x_i$son los vectores de características de la obra nueva y la obra histórica respectivamente, y $d(\cdot)$ es la distancia euclídea calculada sobre las variables numéricas transformadas con logaritmo y las variables categóricas codificadas numéricamente. Si no se dispone de características para alguna obra, su peso se fija en 1, equivalente a la media simple no ponderada.

El fragmento de código siguiente muestra el cálculo de la similitud y la construcción de la predicción ponderada:

```
def similitud_obras(obra1_features, obra2_features):
    diff = np.array(obra1_features) - np.array(obra2_features)
    dist = np.sqrt(np.sum(diff**2))
    return 1 / (1 + dist)

# Media ponderada sobre las curvas históricas
pesos = np.array([similitud_obras(feats_target, feats_i) for feats_i in
feats_hist])
pesos = pesos / pesos.sum()
prediccion = np.average(arr_curvas, weights=pesos, axis=0)
```

La evaluación sigue el protocolo leave-one-out: para predecir la curva de cada obra, el modelo usa únicamente las demás obras del mismo código. Su ventaja principal es la robustez: no requiere entrenamiento, no puede sobreajustarse y funciona con cualquier número de obras históricas disponibles. Su limitación estructural es que ignora completamente la curva parcialmente observada de la obra en curso, generando la misma predicción independientemente de si la obra lleva un 10% o un 70% del tiempo transcurrido. Esta limitación es precisamente lo que los modelos siguientes tratan de superar.

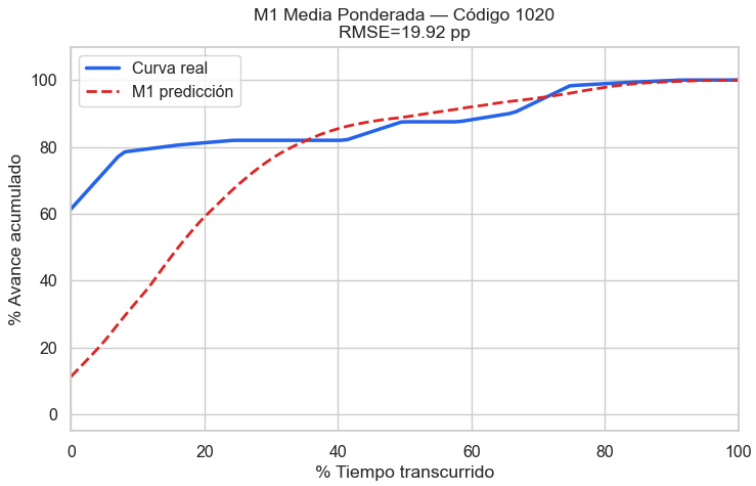


Figura 35 - Ejemplo de predicción M1 para código 1020

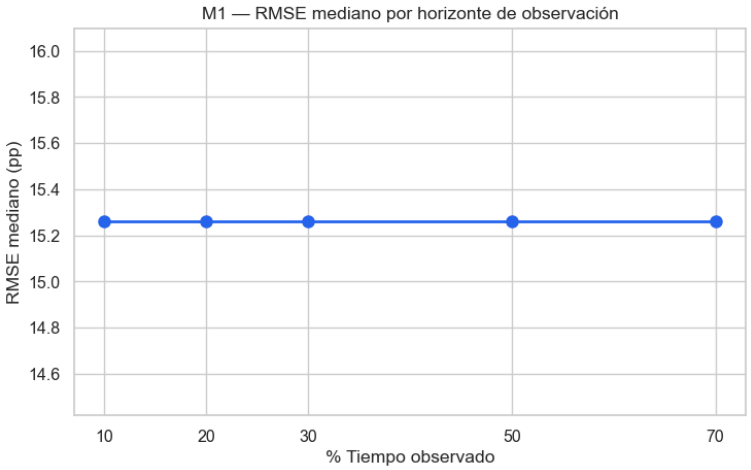


Figura 36 - RMSE mediano del modelo M1 en función del horizonte de observación

8.2 M2 - DTW

El Dynamic Time Warping, conocido por sus siglas DTW, es un algoritmo para medir la similitud entre dos secuencias temporales que pueden variar en velocidad o longitud, permitiendo detectar similitudes aunque una serie esté adelantada o atrasada respecto a la otra. A diferencia de la distancia euclídea punto a punto, que exige alineación temporal perfecta, DTW encuentra el alineamiento óptimo entre dos secuencias minimizando la distancia acumulada a lo largo de todos los posibles caminos de correspondencia entre sus puntos.

Formalmente, dados dos secuencias s_1 de longitud n y s_2 de longitud m, DTW construye una matriz de costes acumulados D de dimensiones $(n+1) \times (m+1)$ mediante la recurrencia:

$$E.\ 5 \qquad D(i,j) = |s_1(i) - s_2(j)| + min\,(D(i-1,j), D(i,j-1), D(i-1,j-1))$$

La distancia DTW entre las dos secuencias es el valor $D(n,m)$. Para acotar el espacio de búsqueda y reducir el coste computacional, se aplica una ventana de Sakoe-Chiba que restringe el camino a una banda diagonal de ancho fijo, implementada como sigue:

```
def dtw_distance_manual(s1, s2, window=10):
    n, m = len(s1), len(s2)
    w = max(window, abs(n - m))
    dtw_matrix = np.full((n+1, m+1), np.inf)
    dtw_matrix[0, 0] = 0
    for i in range(1, n+1):
        for j in range(max(1, i-w), min(m, i+w)+1):
            cost = abs(s1[i-1] - s2[j-1])
            dtw_matrix[i, j] = cost + min(
                dtw_matrix[i-1, j],
                dtw_matrix[i, j-1],
                dtw_matrix[i-1, j-1]
            )
    return dtw_matrix[n, m]
```

La diferencia fundamental respecto a M1 es que la similitud se calcula sobre la porción ya observada de la curva de la obra nueva, no solo sobre sus características estáticas. Dado que se conocen los primeros t% de puntos de la curva en curso, DTW mide la distancia entre esa porción y los primeros t% de cada curva histórica del mismo código. Las cinco obras con

menor distancia DTW se seleccionan como vecinos y sus curvas completas se promedian ponderando inversamente a la distancia:

$$E.\ 6 \qquad w_i = \frac{1}{d_{DTW}(obs_{target}\ ,\ obs_i)+\epsilon}$$

Esta adaptación tiene una implicación conceptual importante: a medida que avanza la obra y se dispone de más puntos observados, la similitud DTW se vuelve más informativa y la predicción mejora progresivamente. En los primeros estadios de la obra, con muy pocos puntos observados, DTW no tiene ventaja clara sobre M1 porque la porción observada es demasiado corta para discriminar entre patrones distintos. A partir del 20-30% del tiempo transcurrido, la similitud en forma de curva comienza a aportar información que M1 no puede capturar. Su limitación principal es el coste computacional cuando el número de obras históricas es grande, mitigado en la implementación mediante el uso de la librería dtaidistance con aceleración nativa cuando está disponible.

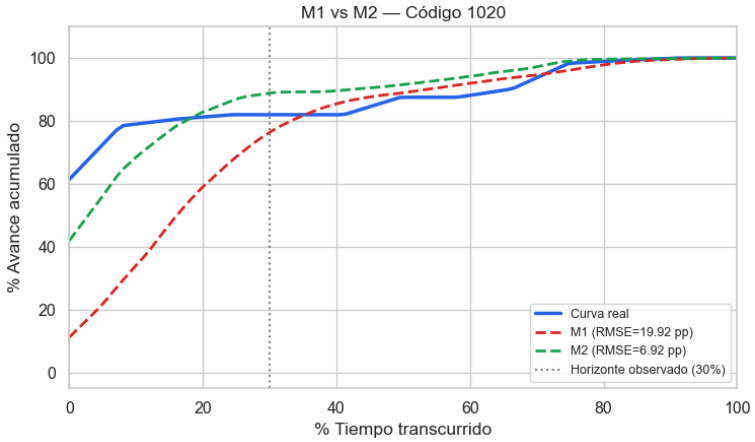


Figura 37 - Ejemplo de predicción M1 y M2 para código 1020

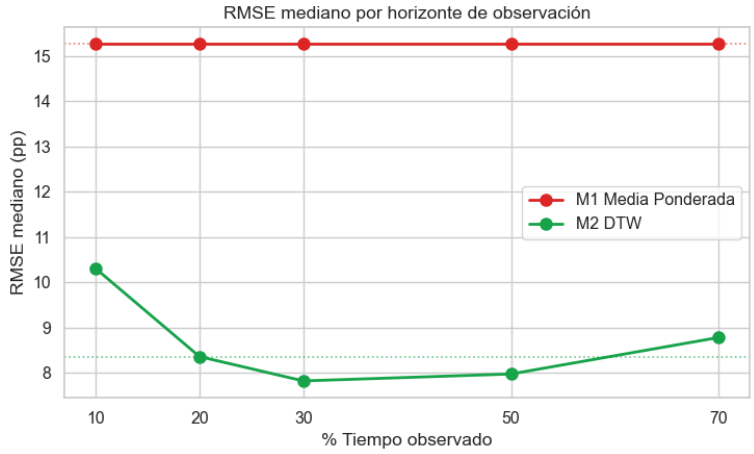


Figura 38 - RMSE mediano de los modelos M1 y M2 en función del horizonte de observación

8.3 M3 - LIGHTGBM

LightGBM es un algoritmo de gradient boosting que construye un ensemble de árboles de decisión de forma secuencial, donde cada árbol aprende a corregir los errores residuales del conjunto anterior. Sus características técnicas diferenciales son el crecimiento de árboles por hoja en lugar de por nivel, que permite modelos más expresivos con el mismo número de parámetros, y el uso de histogramas de features que agrupa los valores continuos en bins discretos para acelerar el cálculo de las divisiones óptimas. Estas características hacen que LightGBM sea significativamente más rápido que otros algoritmos de gradient boosting manteniendo una precisión comparable, lo que es especialmente relevante en el contexto de la evaluación leave-one-out donde el modelo debe reentrenarse para cada obra evaluada.

La formulación del problema para LightGBM requiere una decisión de diseño fundamental: cómo representar la predicción de una curva completa de 100 puntos como un problema de regresión tabular. La aproximación adoptada transforma cada punto de la grilla temporal en una fila independiente del dataset de entrenamiento. Cada fila contiene el vector de características de la obra, el valor de tiempo normalizado t en ese punto y el índice del punto normalizado por la longitud de la grilla. El target es el avance acumulado en ese punto:

$$E.\ 7 \qquad \hat{y}_i(t) = f_{LGBM}(x_i, t), \quad t \in \{0, \tfrac{1}{99}, \tfrac{2}{99}, \dots, 1\}$$

donde $x_i$es el vector de características de la obra iy f_{LGBM}es la función aprendida por el ensemble de árboles. Con esta formulación, el modelo aprende implícitamente la relación entre las características de la obra y la forma esperada de su curva en cada punto temporal, capturando cómo distintos tipos de obra tienen distintas velocidades de producción en distintas fases del ciclo. La configuración de hiperparámetros utilizada es la siguiente:

```
params_lgbm = {
    'objective'        : 'regression',
    'n_estimators'     : 300,
    'learning_rate'    : 0.05,
    'num_leaves'       : 31,
    'min_child_samples': 5,
    'feature_fraction' : 0.8,
    'bagging_fraction' : 0.8,
    'bagging_freq'     : 5,
    'verbose'          : -1,
    'random_state'     : 42
}
```

En inferencia, para predecir la curva completa de una obra nueva se evalúa el modelo en los 100 puntos de la grilla con las características de esa obra, y se aplica una corrección de monotonía acumulando el máximo para garantizar que la curva predicha sea no decreciente, propiedad que el modelo no garantiza por sí mismo. El early stopping a 30 rondas sin mejora evita el sobreajuste durante el entrenamiento leave-one-out. LightGBM es el modelo de machine learning principal por tres razones: su capacidad para manejar la heterogeneidad del dataset combinando variables numéricas de escala muy distinta con variables categóricas de alta cardinalidad, su eficiencia computacional que permite el reentrenamiento leave-one-out completo en tiempo razonable en CPU local, y la interpretabilidad de la importancia de variables que conecta los resultados del modelo con los hallazgos del EDA.

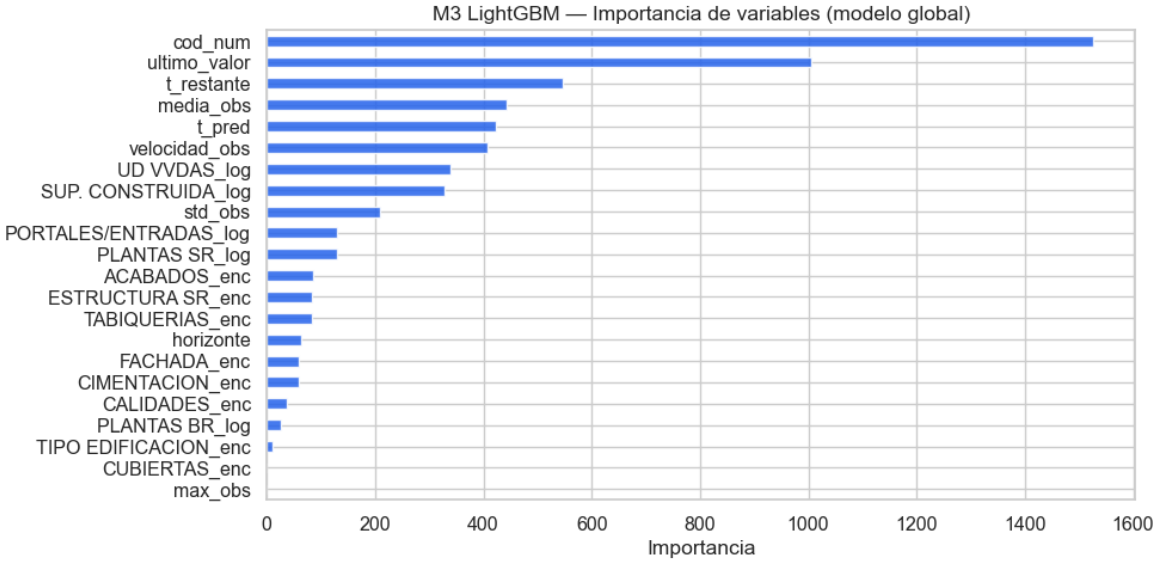


Figura 39 - Importancia de las variables del modelo M3

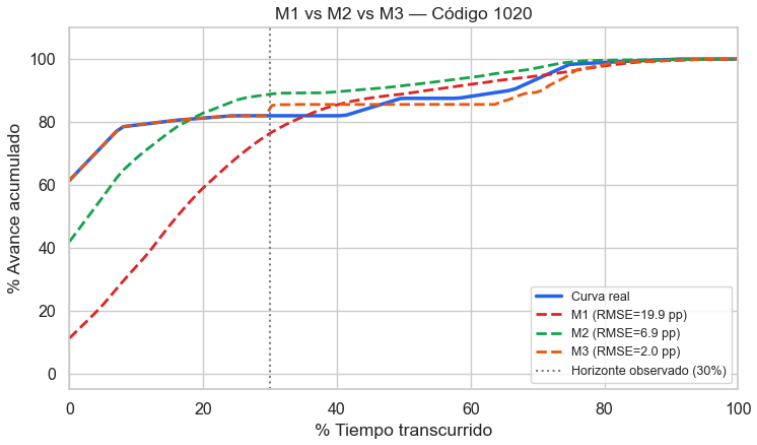


Figura 40 - Ejemplo de predicción M1, M2 y M3 para código 1020

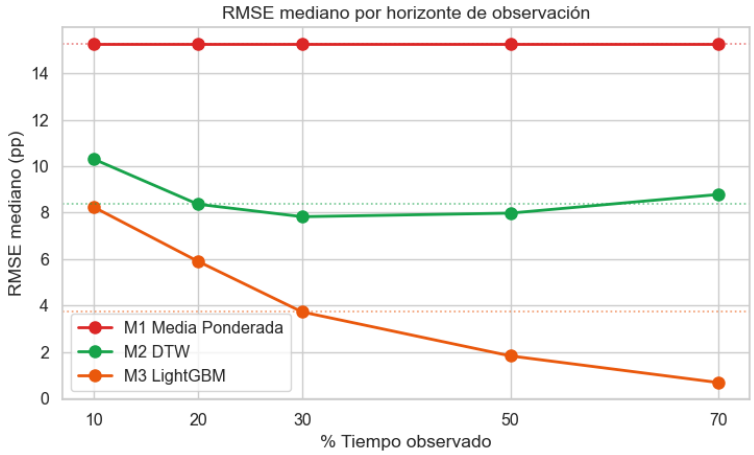


Figura 41 - RMSE mediano de los modelos M1, M2 y M3 en función del horizonte de observación

8.4 M4 - RANDOM FOREST

Random Forest es un ensemble de árboles de decisión entrenados en paralelo sobre submuestras aleatorias del dataset con reemplazo, técnica conocida como bagging. En cada división de cada árbol, solo se considera un subconjunto aleatorio de las features disponibles, lo que introduce diversidad entre los árboles y reduce la correlación entre ellos. La predicción final es la media de las predicciones de todos los árboles individuales:

$$E.\ 8 \qquad \hat{y}(t) = \frac{1}{B}\sum_{b-1}^{B} T_b\,(x,t)$$

donde Bes el número de árboles y $T_b(x,t)$ es la predicción del árbol $b$para las características $x$en el punto temporal t. A diferencia del gradient boosting, donde los árboles se construyen secuencialmente para corregir los errores anteriores, en Random Forest los árboles son independientes entre sí. Esto hace que el modelo sea más robusto frente a outliers en el target y menos sensible a la elección de hiperparámetros, pero generalmente menos preciso que el gradient boosting cuando el dataset es suficientemente grande.

La formulación del dataset de entrenamiento es idéntica a la de LightGBM para garantizar una comparación directa y limpia: mismas filas, mismas features, mismo protocolo leave-one-out y misma corrección de monotonía en la predicción. Los hiperparámetros utilizados son los siguientes:

```
params_rf = {
    'n_estimators'    : 200,
    'max_depth'       : 12,
    'min_samples_leaf': 5,
    'max_features'    : 'sqrt',
    'n_jobs'          : -1,
    'random_state'    : 42
}
```

Random Forest actúa como baseline de machine learning con un doble propósito. El primero es validar si la complejidad adicional de LightGBM, con su proceso de boosting secuencial, se traduce en una mejora real en este problema concreto: si ambos modelos tienen un rendimiento similar, la recomendación sería usar Random Forest por su mayor robustez y sencillez. El segundo propósito es proporcionar una segunda estimación de importancia de variables, ya que LightGBM y Random Forest pueden asignar importancias distintas a las mismas features y la coincidencia entre ambas refuerza la solidez del hallazgo.

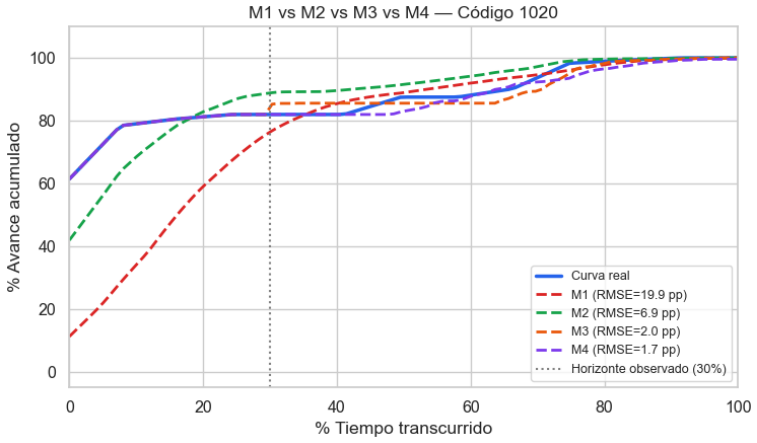


Figura 42 - Ejemplo de predicción M1, M2, M3 y M4 para código 1020

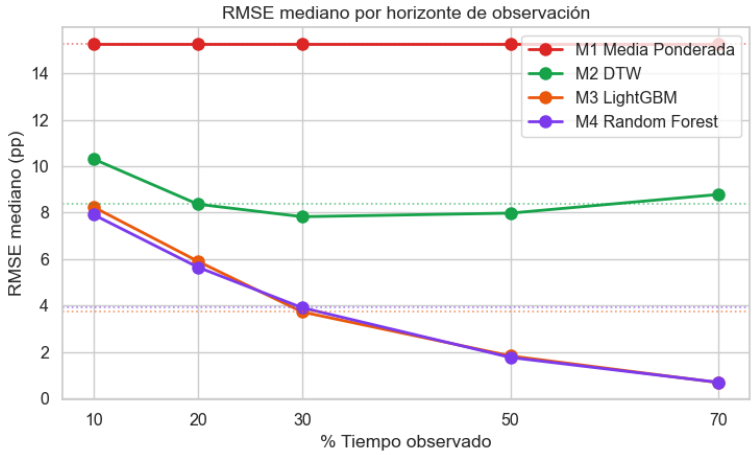


Figura 43 - RMSE mediano de los modelos M1, M2, M3 y M4 en función del horizonte de observación

8.5 M5 - LSTM

Las redes de memoria a largo y corto plazo, conocidas por sus siglas LSTM, son una variante de las redes neuronales recurrentes diseñada para aprender dependencias de largo alcance en secuencias temporales. El problema fundamental de las redes recurrentes estándar es el desvanecimiento del gradiente, que impide aprender relaciones entre puntos temporales muy separados en la secuencia. Las LSTM resuelven esto mediante un mecanismo de tres puertas que controla de forma diferenciada qué información nueva entra en el estado de la celda c_t, qué información antigua se descarta y qué parte del estado se transfiere al siguiente paso temporal:

$$E.\ 9 \qquad f_t = \sigma\big(W_f[h_{t-1}, x_t] + b_f\big) \quad (puerta\ de\ olvido)$$

$$E.\ 10 \qquad i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (puerta\ de\ entrada)$$

$$E.\ 11 \qquad \tilde{c}_t = tanh(W_c[h_{t-1}, x_t] + b_c) \quad (candidatos)$$

$$E.\ 12 \qquad c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (actualización\ del\ estado)$$

$$E.\ 13 \qquad o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \quad h_t = o_t \odot tanh\,(c_t) \quad (salida)$$

La arquitectura implementada sigue un diseño encoder-decoder específicamente adaptado al problema de completar una curva S a partir de su porción observada. El encoder es una red LSTM de dos capas que procesa la secuencia de puntos ya observados de la obra, donde cada paso temporal tiene dos dimensiones: el avance normalizado al intervalo [0,1] y el tiempo normalizado correspondiente. Las características estáticas de la obra se integran en el estado oculto del encoder mediante una subcapa densa que proyecta el vector de features al mismo espacio dimensional y lo combina con el estado oculto mediante una suma seguida de una activación tanh. El decoder es otra LSTM de dos capas que genera los puntos futuros usando como estado inicial el estado final del encoder, recibiendo en cada paso temporal la concatenación del contexto del encoder y el tiempo normalizado a predecir.

La definición de la arquitectura en PyTorch refleja directamente este diseño:

```
class LSTMCurvaS(nn.Module):
    def __init__(self, n_features_estaticas, hidden_size=64, n_layers=2,
dropout=0.3):
        super().__init__()
        self.encoder     = nn.LSTM(input_size=2, hidden_size=hidden_size,
                                   num_layers=n_layers, batch_first=True,
                                   dropout=dropout if n_layers > 1 else 0)
        self.fc_features = nn.Sequential(nn.Linear(n_features_estaticas,
hidden_size),
                                         nn.ReLU(), nn.Dropout(dropout))
        self.decoder     = nn.LSTM(input_size=hidden_size+1,
hidden_size=hidden_size,
                                   num_layers=n_layers, batch_first=True,
                                   dropout=dropout if n_layers > 1 else 0)
        self.fc_out      = nn.Sequential(nn.Linear(hidden_size, 32), nn.ReLU(),
                                         nn.Linear(32, 1), nn.Sigmoid())
```

El entrenamiento usa el optimizador Adam con reducción adaptativa de la tasa de aprendizaje cuando la pérdida deja de mejorar, gradient clipping con norma máxima de 1 para estabilizar el entrenamiento, y la función de pérdida de error cuadrático medio sobre los puntos futuros. El entrenamiento se realiza en Google Colaboratory con GPU dado el coste computacional del proceso.

El modelo LSTM es el único de los seis que puede aprender explícitamente de la secuencia temporal observada para adaptar la predicción al comportamiento específico de la obra en

curso. Mientras que LightGBM recibe el tiempo como una feature más sin capturar la dependencia entre puntos consecutivos, la LSTM procesa la secuencia completa y puede detectar patrones como que una obra con arranque más lento de lo habitual tenderá a compensar con mayor velocidad en la fase central. Esta capacidad de razonamiento secuencial justifica la complejidad adicional de la arquitectura, aunque su rendimiento depende de disponer de suficientes series de entrenamiento para que el encoder aprenda representaciones útiles del comportamiento parcial de la curva.

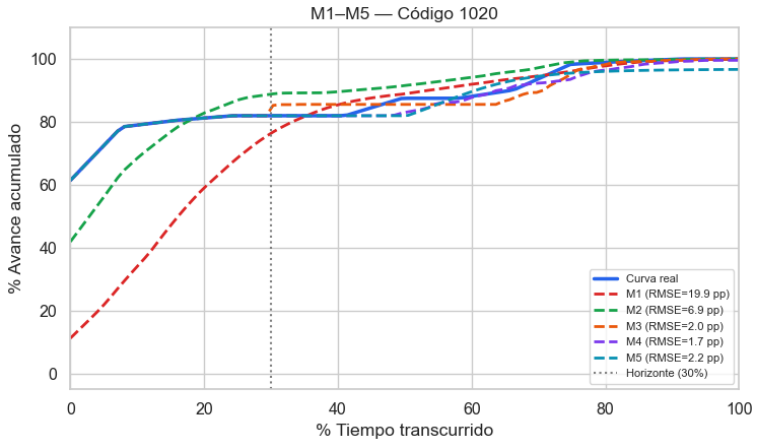


Figura 44 - Ejemplo de predicción M1, M2, M3, M4 y M5 para código 1020

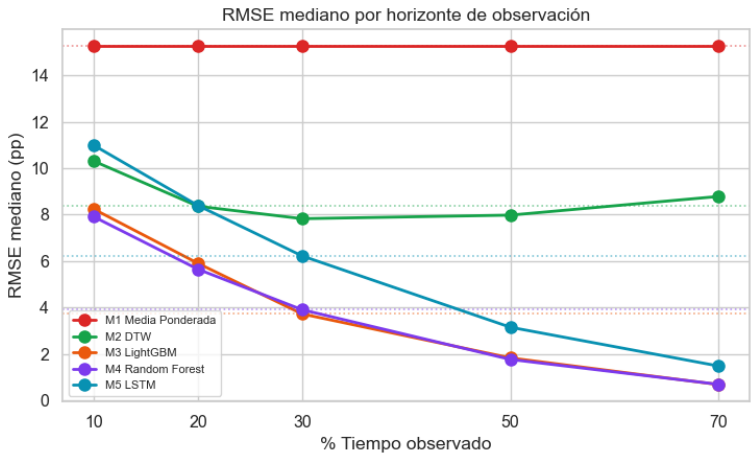


Figura 45 - RMSE mediano de los modelos M1, M2, M3, M4 y M5 en función del horizonte de observación

8.6 M6 - CURVA LOGÍSTICA PARAMÉTRICA

El modelo logístico paramétrico se apoya directamente en la validación empírica realizada en el EDA: la mayoría de las curvas de producción por código de compra se ajustan bien a una función logística de tres parámetros. A diferencia de los demás modelos, que predicen la curva punto a punto, este modelo predice los parámetros de la función analítica y reconstruye la curva completa a partir de ellos:

$$E.\ 14 \quad \hat{y}(t) = \frac{L}{1+e^{-k(t-x_0)}}$$

donde L es el nivel máximo de avance alcanzado, $k$es la pendiente que controla la velocidad de crecimiento y $x_0$es el punto de inflexión donde la velocidad de producción es máxima. Esta representación compacta les confiere dos propiedades únicas respecto a los demás modelos: la curva predicha es siempre matemáticamente suave y con forma S por construcción, y puede generar una predicción razonable con muy pocos puntos observados, porque ajustar tres parámetros requiere un mínimo de cuatro puntos mientras que los modelos de machine learning necesitan mucha más información.

El modelo implementa dos estrategias que se aplican según el horizonte de observación disponible. La primera es el ajuste directo: cuando se dispone de al menos cuatro puntos observados y el ajuste logístico sobre esos puntos produce un R^2 superior a 0.7, se usa la función ajustada para extrapolar el resto. El ajuste se realiza mediante mínimos cuadrados no lineales con el método Trust Region Reflective y restricciones de bounds que garantizan parámetros físicamente plausibles:

```
popt, _ = curve_fit(
    logistica, x_obs, y_obs,
    p0=[100, 5, 0.5],
    bounds=([80, 0.1, 0.0], [110, 50, 1.0]),
    maxfev=8000, method='trf'
)
```

La segunda estrategia se activa cuando el ajuste directo no es viable, principalmente en los primeros estadios de la obra cuando hay muy pocos puntos observados. En este caso, se

entrena un modelo de Random Forest sobre las obras históricas del mismo código usando sus características estáticas como features y los parámetros logísticos ajustados a sus curvas completas como target. Este Random Forest de parámetros predice los valores de k, x_0y *L*esperables para una obra con las características de la obra nueva, y la curva se reconstruye analíticamente. Si tampoco hay suficientes obras históricas con buen ajuste para entrenar este modelo secundario, se usa como fallback la mediana de los parámetros históricos disponibles.

El modelo logístico es especialmente valioso en dos situaciones identificadas en el EDA. La primera es cuando la obra está en sus fases muy tempranas, con menos del 20% del tiempo transcurrido, donde los demás modelos tienen poca información para trabajar pero el logístico puede generar una predicción inicial razonable basada en las características del proyecto. La segunda es cuando el código de compra tiene baja cobertura histórica con pocas obras disponibles para entrenar los modelos de machine learning. Su limitación estructural es que no puede capturar curvas bimodales, curvas con parones prolongados o cualquier comportamiento que se desvíe significativamente de la forma logística, casos que representan el subconjunto con $R^2<0.7$ identificado en el EDA.

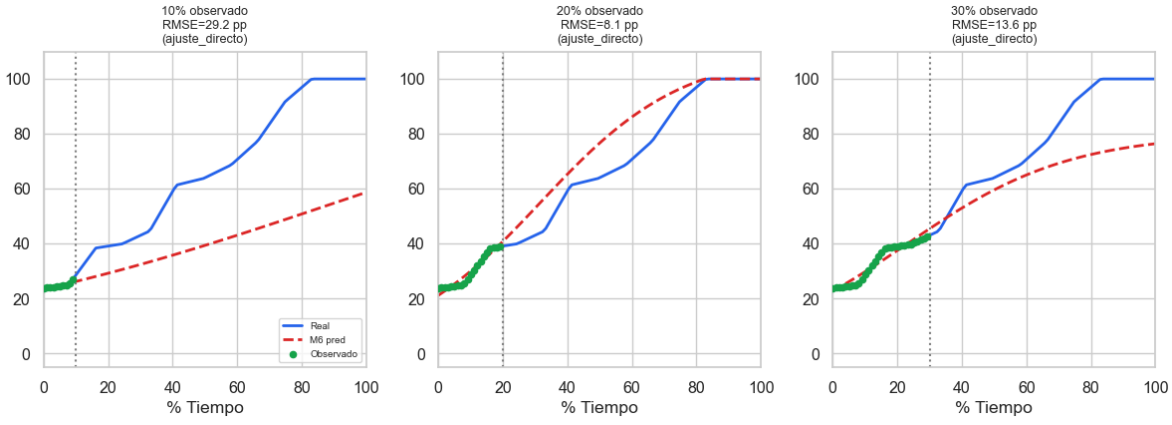


Figura 46 - Evolución de la predicción M6 a cinco horizontes I

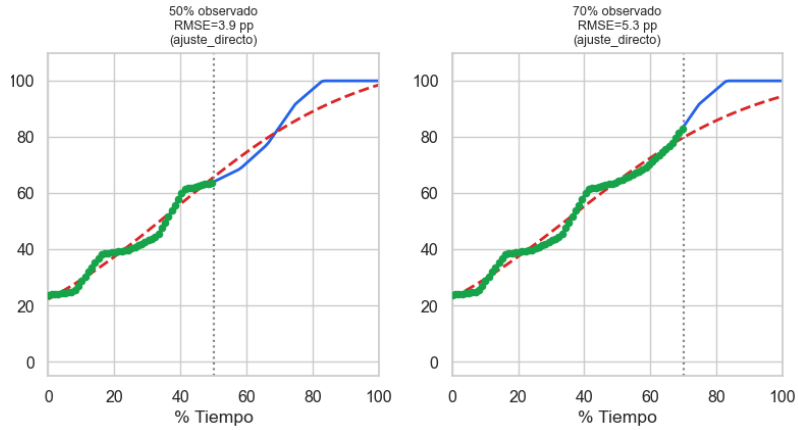


Figura 47 - Evolución de la predicción M6 a cinco horizontes II

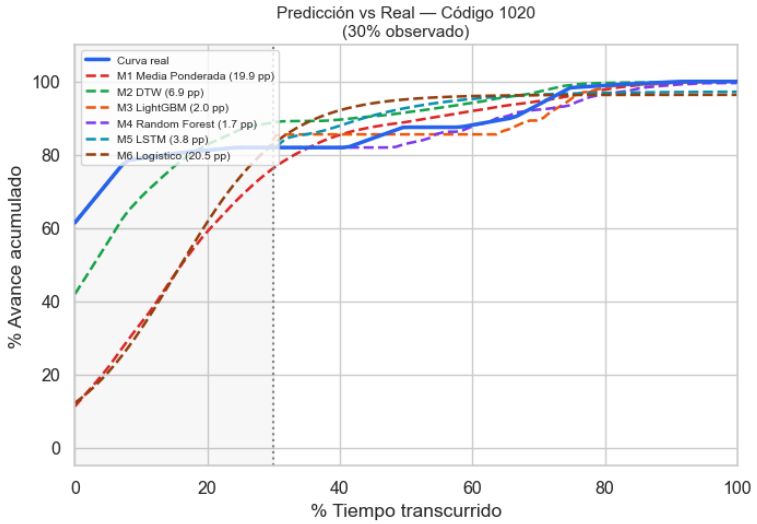


Figura 48 - Ejemplo de predicción M1, M2, M3, M4, M5 y M6 para código 1020

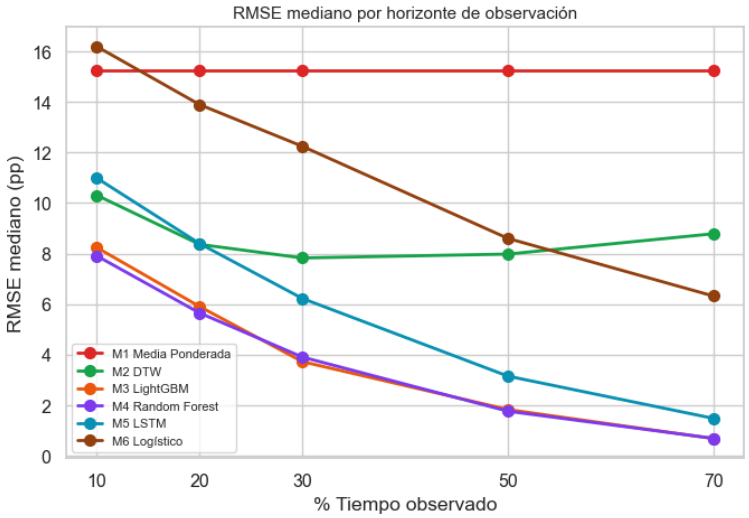


Figura 49 - RMSE mediano de los modelos M1, M2, M3, M4, M5 y M6 en función del horizonte de observación

8.7 EVALUACIÓN COMPARATIVA

La comparación entre los seis modelos se realiza mediante un protocolo leave-one-out aplicado de forma consistente a todos ellos. Para cada código de compra con suficiente cobertura, se evalúa cada modelo sobre cada obra disponible entrenando o calibrando el modelo exclusivamente con las demás obras del mismo código, sin usar ningún dato de la obra evaluada. Este protocolo simula el escenario real de uso: predecir el comportamiento de una obra nueva a partir únicamente del histórico de obras anteriores.

La evaluación se realiza a cinco horizontes de observación: 10%, 20%, 30%, 50% y 70% del tiempo total transcurrido. Para cada horizonte se simula que solo se dispone de los puntos de la curva hasta ese porcentaje de tiempo y se genera la predicción del resto. La métrica principal es el RMSE en puntos porcentuales de avance calculado sobre la curva completa predicha. Se calcula también el RMSE en el tramo final, correspondiente al último 30% del tiempo, que es el más relevante para la planificación del aprovisionamiento porque concentra los trabajos de acabado e instalaciones de mayor coste unitario. El RMSE del modelo M1

sirve como umbral de referencia: la mejora porcentual de cada modelo sobre este baseline cuantifica el valor añadido de cada nivel adicional de complejidad.

E. 15 $$RMSE = \sqrt{\frac{1}{100}\sum_{j-1}^{100}\left(\hat{y}(t_j) - y(t_j)\right)^2}, \quad Mejora_{M_k} = \frac{RMSE_{M1} - RMSE_{M_k}}{RMSE_{M1}} * 100\%$$

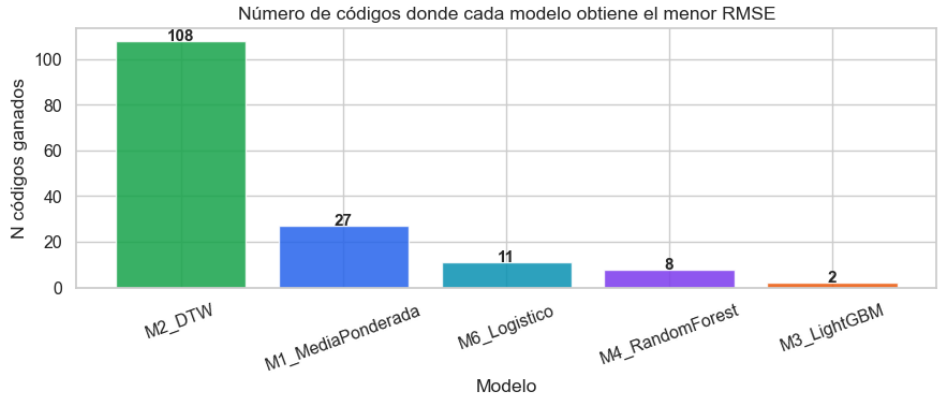


Figura 50 - Número de códigos donde cada modelo obtiene menor RMSE

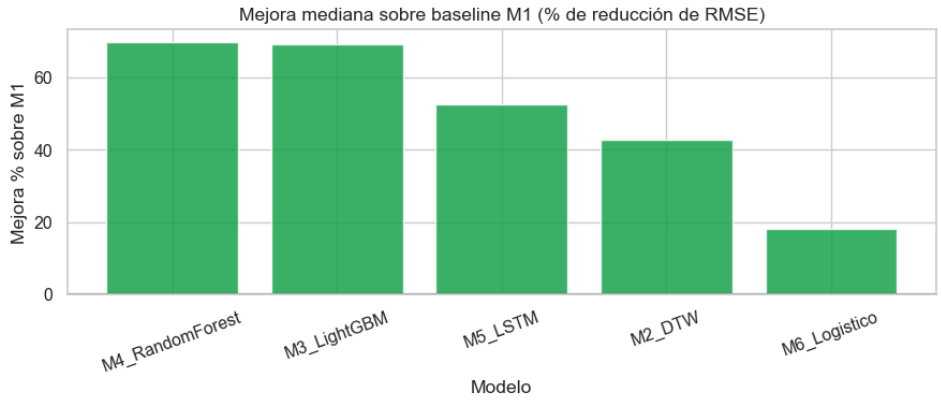


Figura 51 - Mejora de la mediana sobre el baseline M1

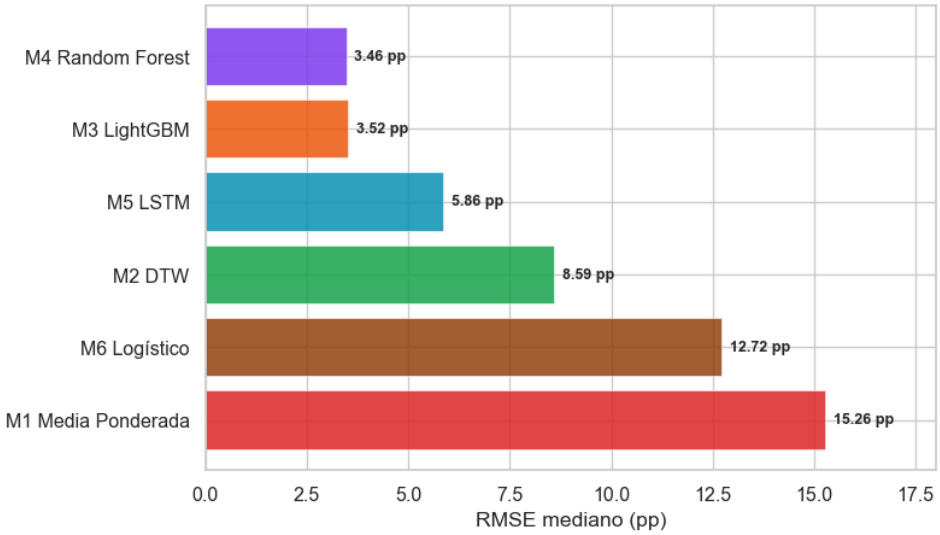


Figura 52 - RMSE mediano global (pp)

Capítulo 9. INTERFAZ WEB

En este capítulo se describe la interfaz web desarrollada como capa de acceso visual a los resultados del modelo predictivo de curvas S. El objetivo de esta herramienta es permitir que perfiles no técnicos, como responsables de obra o directores de proyecto, puedan explorar el avance de las obras, comparar las predicciones de los distintos modelos y consultar el cronograma de códigos de compra de forma intuitiva, sin necesidad de interactuar directamente con el código o los datos en bruto.

La aplicación ha sido implementada con Streamlit, un framework de Python orientado al desarrollo rápido de dashboards analíticos orientados a datos. Se conecta a la base de datos SQL Server corporativa mediante el driver JDBC de Microsoft, cargando las tablas de seguimiento de obras y curvas de producción, y expone toda la funcionalidad a través de una interfaz accesible desde el navegador. La aplicación se organiza en un panel lateral de configuración y cuatro pestañas principales, cada una orientada a un nivel distinto de análisis.

9.1 ELEMENTOS COMUNES

Antes de describir cada pestaña, se presentan los elementos de la interfaz que están presentes de forma permanente a lo largo de toda la sesión.

9.1.1 ESTILO VISUAL Y PALETA CORPORATIVA

La aplicación aplica un sistema de estilos personalizado mediante CSS inyectado con st.markdown(..., unsafe_allow_html=True). Se utilizan las tipografías DM Sans y DM Mono, y se adopta la paleta corporativa de Arpada, con verde #046938 como color primario, lima #92D050 como color de acento y amarillo #FBE122 como color de alerta. Todos los gráficos comparten una plantilla oscura coherente basada en Plotly, con fondo #0F1117 y rejilla #2A2F45, lo que proporciona una experiencia visual uniforme en todas las vistas. Los

elementos interactivos como botones, pestañas activas y scrollbars siguen la misma guía de color, reforzando la identidad de la herramienta.

9.1.2 PANEL LATERAL (SIDEBAR)

El panel lateral, visible en todo momento en el lado izquierdo de la pantalla, centraliza todos los controles de configuración global de la aplicación. Desde él, el usuario puede:

- Seleccionar la obra activa, a partir de un desplegable que muestra únicamente las obras que disponen de datos de curvas en la base de datos.
- Elegir los códigos de compra a analizar, mediante un selector múltiple que permite activar o desactivar categorías como Movimiento de Tierras, Cimentación, Estructura, Instalaciones, etc.
- Ajustar el horizonte observado, mediante un slider que permite definir qué porcentaje del tiempo de la obra se considera ya transcurrido, con un rango de 10 % a 90 % en pasos de 10 puntos.
- Seleccionar los modelos de predicción activos entre los disponibles: M1 Media Ponderada, M2 DTW, M3 LightGBM y M6 Logístico.

Adicionalmente, el panel muestra una ficha informativa de la obra seleccionada con su nombre, localidad, número de viviendas, superficie construida en m² y estado (Finalizada o En ejecución), indicado mediante un badge de color.

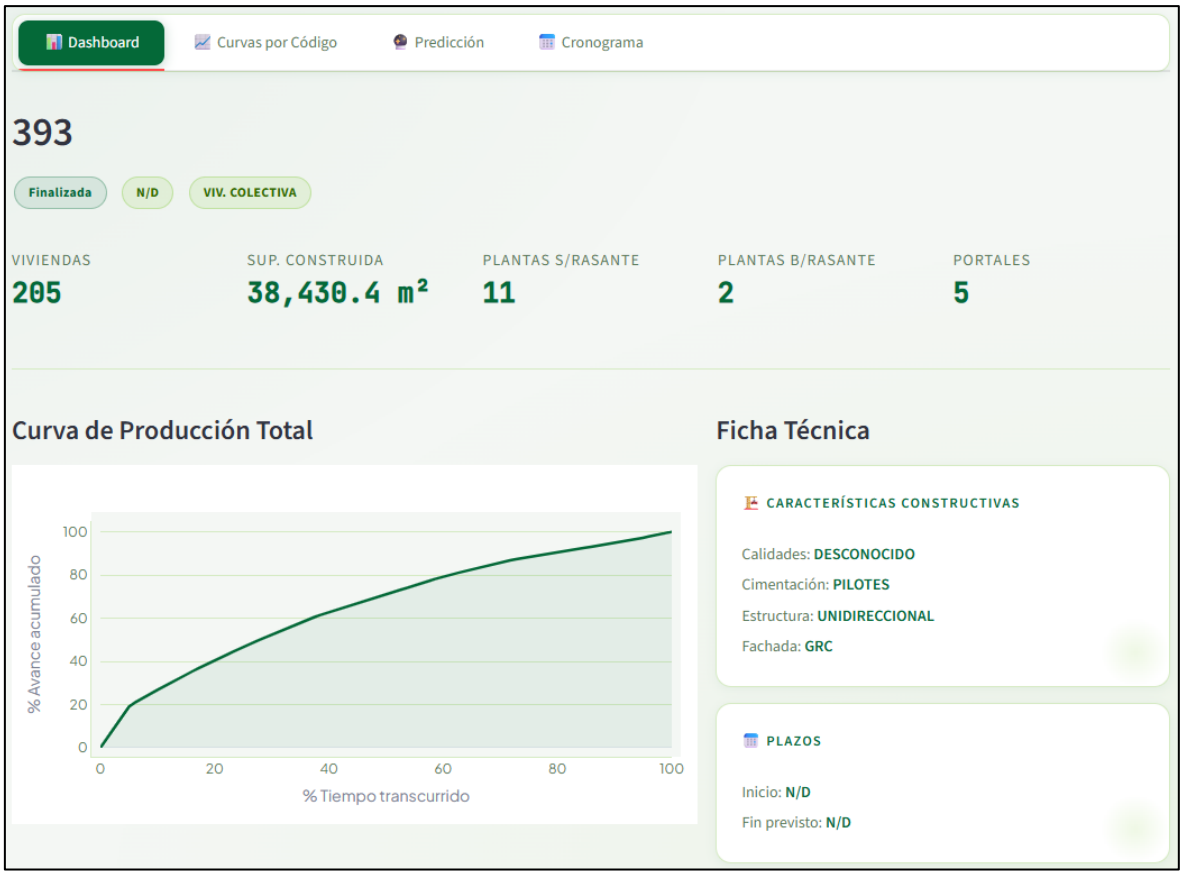
Figura 53 - Panel lateral de selección de obra y configuración

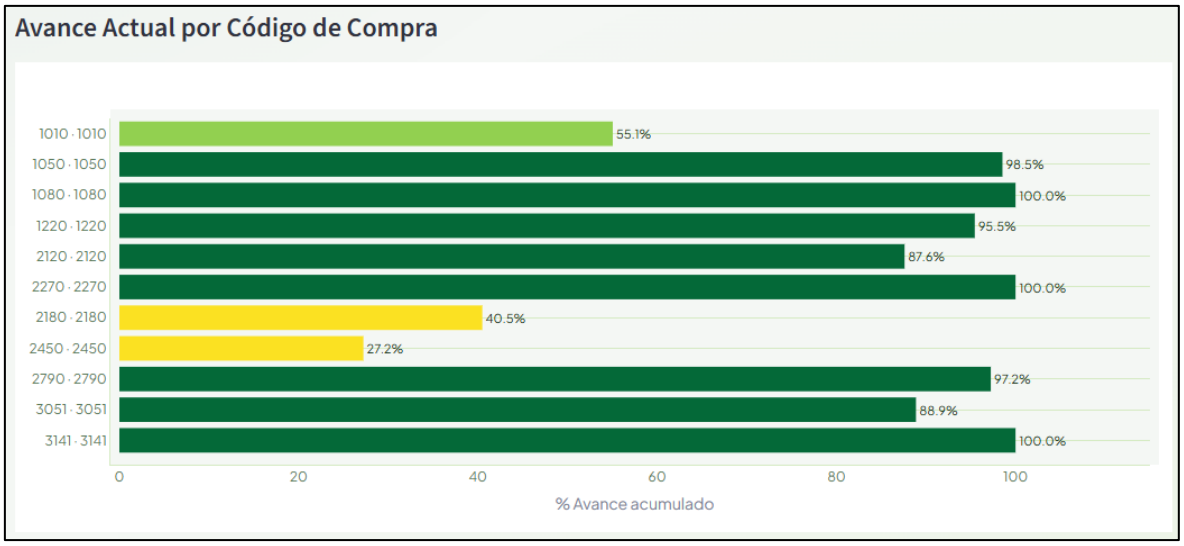
9.2 PESTAÑA DASHBOARD

La pestaña Dashboard ofrece una visión global del estado de la obra seleccionada en el momento definido por el horizonte observado. En la parte superior se muestran cinco métricas: número de viviendas, superficie construida, plantas sobre rasante, plantas bajo rasante y número de portales.

A continuación, el área principal se divide en dos columnas. La columna izquierda presenta la curva de producción total, calculada como la media de las curvas de todos los códigos de compra activos, representada con relleno bajo la curva y una línea vertical punteada que marca el horizonte actual. La columna derecha muestra una ficha técnica de la obra con sus características constructivas principales: tipo de calidades, cimentación, estructura y fachada, así como las fechas de inicio y fin previsto.

En la parte inferior, un gráfico de barras horizontales refleja el avance actual por código de compra, con un código de color que facilita la lectura rápida del estado de cada partida.





9.3 PESTAÑA CURVAS POR CÓDIGO

La pestaña Curvas por Código desciende al nivel de detalle individual de cada partida. Los gráficos se presentan en una cuadrícula de dos columnas, mostrando una curva por cada código de compra seleccionado en el sidebar. Cada gráfico indica en el subtítulo el avance actual de ese código en el horizonte configurado.

El comportamiento visual de cada curva varía según el estado de la obra. En obras finalizadas, se superpone sobre la curva real una línea discontinua con la media histórica de otras obras finalizadas para el mismo código, permitiendo comparar el comportamiento de la obra con el patrón general del conjunto. En obras en ejecución, la porción de la curva que queda por delante del horizonte observado se representa con trazo punteado y opacidad reducida, diferenciando visualmente los datos conocidos de los estimados.

Figura 54 - Curva real vs esperada de cada código

9.4 PESTAÑA PREDICCIÓN

La pestaña Predicción constituye el núcleo analítico de la aplicación. El usuario selecciona un código de compra concreto y la herramienta ejecuta sobre él los modelos de predicción activos, proyectando la evolución de la curva desde el horizonte observado hasta el 100 % del tiempo de obra.

Cada modelo se representa con un color y trazo diferenciado sobre el mismo gráfico, y en la leyenda se indica su RMSE calculado sobre el tramo predicho, expresado en puntos porcentuales. Una línea vertical punteada en amarillo marca el límite entre la zona observada y la zona predicha. Bajo el gráfico, se presentan tarjetas de métricas individuales por modelo que muestran tanto el RMSE del tramo predicho como el RMSE global sobre la curva completa. Finalmente, un panel expandible permite consultar una tabla de valores numéricos con los porcentajes de avance predichos en los deciles del tiempo (10 %, 20 %, ..., 100 %) para cada modelo activo.

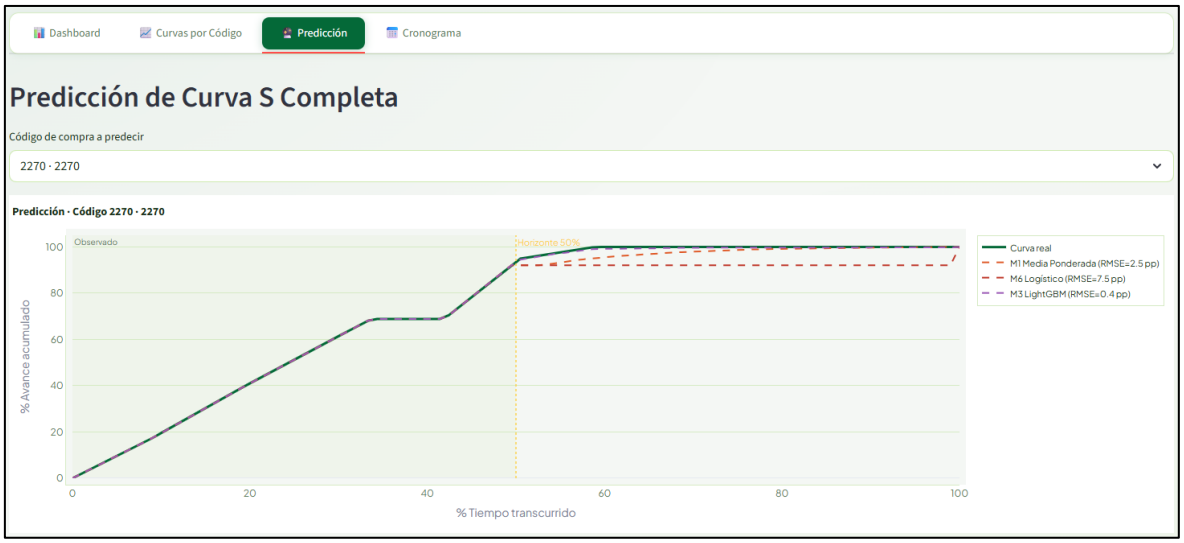


Figura 55 - Predicción curva S con modelos

9.5 PESTAÑA CRONOGRAMA

La pestaña Cronograma ofrece una perspectiva temporal de la actividad de cada código de compra dentro de la obra. Mediante un diagrama de Gantt estilizado, se visualiza la ventana de tiempo estimada en la que cada partida se encuentra activa, expresada como porcentaje del tiempo total de la obra en lugar de fechas absolutas, lo que facilita la comparación entre obras de distinta duración.

Cada barra se representa en dos capas: una exterior semitransparente que muestra la duración total estimada del código, y una interior opaca que representa el progreso completado hasta el horizonte actual. Una línea vertical punteada en amarillo señala el momento presente. El diagrama se complementa con una tabla resumen que indica, para cada código, el inicio y fin estimados, el avance alcanzado y el estado actual, clasificado automáticamente como Completado (avance superior al 95 %), En curso (entre el 5 % y el 95 %) o Pendiente (inferior al 5 %).

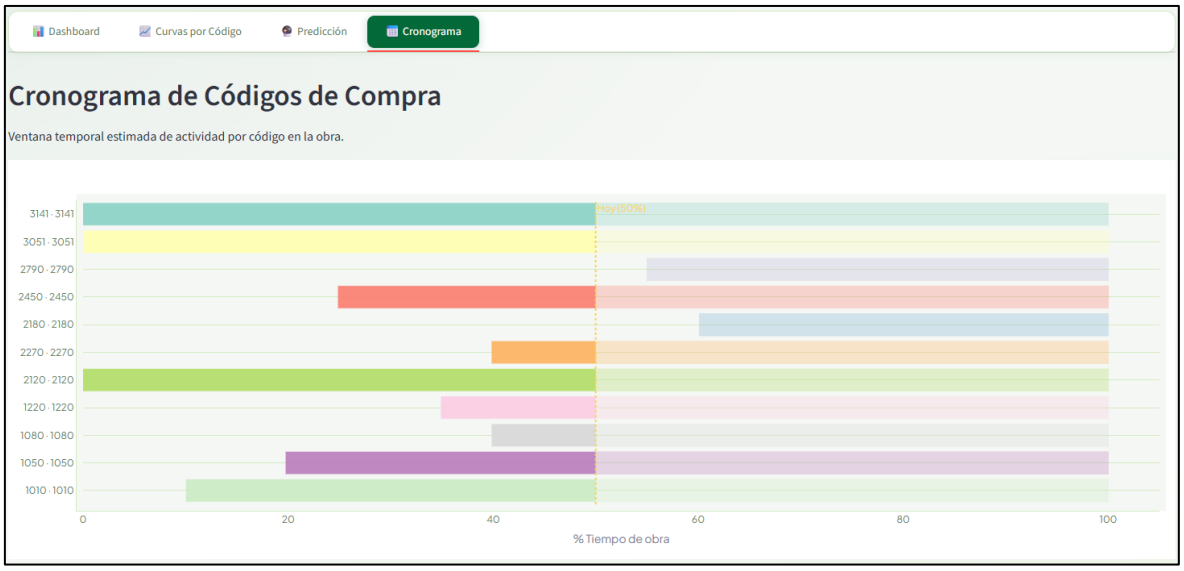
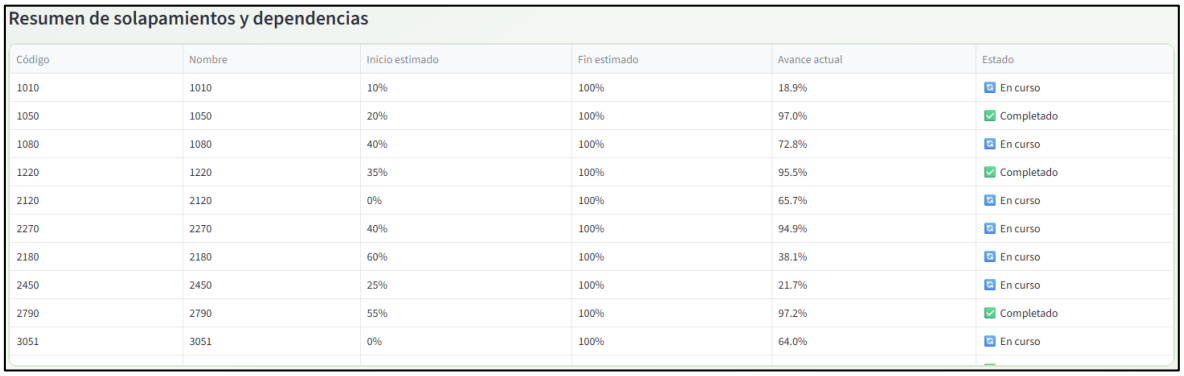


Figura 56 - Cronograma de los códigos de compra



Resumen de solapamientos y dependencias

Código	Nombre	Inicio estimado	Fin estimado	Avance actual	Estado
1010	1010	10%	100%	18.9%	En curso
1050	1050	20%	100%	97.0%	Completado
1080	1080	40%	100%	72.8%	En curso
1220	1220	35%	100%	95.5%	Completado
2120	2120	0%	100%	65.7%	En curso
2270	2270	40%	100%	94.9%	En curso
2180	2180	60%	100%	38.1%	En curso
2450	2450	25%	100%	21.7%	En curso
2790	2790	55%	100%	97.2%	Completado
3051	3051	0%	100%	64.0%	En curso

Figura 57 - Resumen con los solapamientos y dependencias de los códigos de compra

Capítulo 10. ANÁLISIS DE RESULTADOS

El sistema desarrollado ha sido evaluado mediante un protocolo leave-one-out aplicado de forma consistente a los seis modelos sobre el mismo dataset, compuesto por 69 obras, 156 códigos de compra con cobertura suficiente y 6.973 series temporales normalizadas. La evaluación se ha realizado a cinco horizontes de observación, del 10% al 70% del tiempo transcurrido, para simular el escenario real donde la predicción se genera en distintos momentos del ciclo de vida de la obra. La métrica principal es el RMSE en puntos porcentuales de avance acumulado sobre la curva completa predicha.

10.1 RESULTADOS POR MODELO

El modelo M1 de media ponderada, que actúa como baseline de referencia, obtiene un RMSE mediano global de 15.26 pp. Como era esperable por su naturaleza, este RMSE es completamente estable a lo largo de todos los horizontes de observación, manteniéndose en 15.26 pp independientemente de si se ha observado el 10% o el 70% del tiempo de la obra. Esto confirma el comportamiento teórico del modelo: al no usar los puntos observados de la obra en curso, su predicción no mejora conforme avanza la obra. Este resultado establece el umbral mínimo que cualquier modelo debe superar para justificar su implementación.

El modelo M2 de DTW obtiene un RMSE mediano global de 8.59 pp, lo que representa una mejora del 44% sobre el baseline M1. A diferencia de M1, su rendimiento sí varía con el horizonte de observación: parte de un RMSE de 10.30 pp cuando solo se ha observado el 10% del tiempo, mejora progresivamente hasta su mínimo de 7.82 pp al 30% de observación, y luego repunta ligeramente hasta 8.78 pp al 70%. Este comportamiento es coherente con la naturaleza del algoritmo: cuando hay muy pocos puntos observados la similitud DTW es poco discriminativa, mejora al acumularse información suficiente para caracterizar la forma de la curva, y el ligero repunte al 70% se explica porque las series muy cortas restantes a predecir tienen menos varianza sobre la que mejorar. La mejora de DTW sobre M1 valida la

hipótesis de que la información de la curva parcialmente observada aporta valor discriminativo real sobre las características estáticas de la obra.

Los modelos M3 de LightGBM y M4 de Random Forest son los que obtienen mejor rendimiento global, con RMSE medianos de 3.52 pp y 3.46 pp respectivamente, representando mejoras del 77% y el 77% sobre el baseline. Su comportamiento por horizonte es notablemente diferente al de los modelos heurísticos: a horizontes bajos (10%) tienen un RMSE de 8.23 pp y 7.91 pp respectivamente, cercano al de DTW, pero mejoran de forma muy pronunciada conforme aumenta el horizonte, llegando a RMSE de 0.67 pp y 0.68 pp al 70% de observación. Este comportamiento refleja que los modelos de machine learning aprovechan de forma muy eficiente la información de los puntos ya observados de la curva en curso, que se incluyen como features mediante variables como el último valor observado, la media de los puntos observados, la velocidad de avance y la desviación estándar. La importancia de variables del modelo LightGBM confirma este hallazgo: las tres variables más importantes son el código de compra, el último valor observado y el tiempo restante, que concentran la mayor parte de la capacidad predictiva del modelo.

La similitud de resultados entre LightGBM y Random Forest es uno de los hallazgos más relevantes de la evaluación. La diferencia de RMSE entre ambos es de apenas 0.06 pp a nivel global, con perfiles por horizonte prácticamente idénticos, lo que indica que en este problema concreto la complejidad adicional del gradient boosting no aporta ventaja significativa sobre el ensemble de árboles paralelos. Ambos modelos alcanzan prácticamente la misma precisión con estrategias de aprendizaje distintas, lo que refuerza la robustez del resultado.

El modelo M5 de LSTM obtiene un RMSE mediano global de 5.86 pp, con un perfil por horizonte que parte de 10.99 pp al 10%, baja hasta 3.15 pp al 50% y alcanza 1.47 pp al 70%. Este comportamiento es el más pronunciado de todos los modelos: la red aprende a aprovechar la secuencia completa de puntos observados de forma progresiva, y su ventaja sobre los modelos tabulares se concentra en los horizontes intermedios y altos. Sin embargo, a nivel global el LSTM queda por detrás de LightGBM y Random Forest, lo que sugiere que el beneficio del modelado secuencial explícito no compensa el coste de entrenamiento

adicional y la mayor complejidad arquitectural cuando el volumen de datos de entrenamiento es limitado, con 69 obras en total.

El modelo M6 logístico paramétrico obtiene un RMSE mediano global de 12.72 pp, con un perfil por horizonte que parte de 16.19 pp al 10% y mejora progresivamente hasta 6.32 pp al 70%. Es el segundo modelo con peor resultado global, solo por encima del baseline M1, pero su interpretación requiere matizar el contexto. El modelo logístico es el único que puede generar una predicción razonable con tan solo 3 o 4 puntos observados sin necesidad de obras históricas de referencia, y su RMSE al 50% y al 70% de observación es competitivo con DTW. Su limitación principal es la falta de flexibilidad ante curvas que se desvían de la forma logística, que como el EDA mostró representan aproximadamente el 20-30% del dataset con R^2 inferior a 0.9.

10.2 ANÁLISIS COMPARATIVO

La tabla siguiente resume los RMSE medianos de cada modelo por horizonte de observación.

Tabla 8 - Comparación RMSE modelos utilizados

Modelo	*RMSE global*	*10%*	*20%*	*30%*	*50%*	*70%*
M1 Media Ponderada	15.26 pp	15.26	15.26	15.26	15.26	15.26
M2 DTW	8.59 pp	10.30	8.36	7.82	7.98	8.78
M3 LightGBM	3.52 pp	8.23	5.90	3.72	1.82	0.67
M4 Random Forest	3.46 pp	7.91	5.65	3.90	1.76	0.68
M5 LSTM	5.86 pp	10.99	8.39	6.22	3.15	1.47
M6 Logístico	12.72 pp	16.19	13.89	12.24	8.60	6.32

El resultado más relevante para el negocio es que los modelos M3 y M4 superan al baseline M1 en un 77% de reducción de RMSE, pasando de 15.26 pp a menos de 3.5 pp de error mediano. En términos prácticos esto significa que para una obra con un presupuesto de, por ejemplo, 10 millones de euros distribuidos en los distintos códigos de compra, el error de predicción del avance acumulado pasa de ser del orden de 15 puntos porcentuales con el método manual a menos de 4 puntos porcentuales con los modelos de machine learning, lo que mejora sustancialmente la calidad de las decisiones de aprovisionamiento.

El segundo hallazgo relevante es el comportamiento diferencial en función del horizonte. En los primeros estadios de la obra, con menos del 20% del tiempo transcurrido, DTW y los modelos ML tienen un rendimiento similar, y el logístico paramétrico es el peor. Conforme avanza la obra y se acumula más información, los modelos ML y LSTM se distancian claramente de DTW y del logístico, aprovechando el histórico observado para ajustar la predicción al comportamiento específico de esa obra. Este resultado sugiere una estrategia de uso en producción donde en las fases iniciales de la obra se priorice DTW por su robustez con pocos datos, y a partir del 20-30% de tiempo transcurrido se active el modelo LightGBM o Random Forest como predictor principal.

La importancia de variables de LightGBM revela que las features más informativas son, por orden: el código de compra, el último valor observado de la curva, el tiempo restante hasta el final previsto, la media y la velocidad de los puntos observados, y las unidades de vivienda y la superficie construida. Las variables categóricas de tipo constructivo tienen una importancia marginal, lo que sugiere que la forma de la curva está determinada principalmente por el comportamiento parcialmente observado de la propia curva y por el tipo de partida presupuestaria, más que por las características cualitativas de la solución constructiva.

10.3 VALORACIÓN DEL SISTEMA EN SU CONJUNTO

Los resultados obtenidos validan el planteamiento metodológico del trabajo en tres aspectos. Primero, que el histórico de curvas de producción almacenado en los sistemas corporativos

contiene suficiente información para predecir la evolución futura de una obra con una precisión sustancialmente superior al método manual basado en la experiencia experta, formalizado aquí como M1. Segundo, que la información de la curva parcialmente observada es el elemento más valioso para la predicción, más que las características estáticas de la obra, lo que justifica el diseño de modelos que la incorporen explícitamente. Tercero, que la complejidad metodológica tiene rendimientos decrecientes en este contexto: LightGBM y Random Forest, modelos de complejidad intermedia, superan al LSTM de mayor complejidad y al logístico de menor, lo que sugiere que el volumen de datos disponible favorece los modelos que aprovechan eficientemente la información sin requerir grandes cantidades de ejemplos para generalizar.

Capítulo 11. CONCLUSIONES Y TRABAJOS FUTUROS

En este último capítulo se presentan las conclusiones obtenidas a partir del trabajo realizado y se identifican las posibles líneas de desarrollo que puede tener este de manera que se consiga una herramienta más completa y que cubra nuevas necesidades.

11.1 CONCLUSIONES

Este Trabajo de Fin de Máster ha desarrollado un sistema de predicción de curvas de producción por código de compra para obras de construcción residencial, abordando un problema operativo real de la empresa con consecuencias económicas directas en la planificación del aprovisionamiento de materiales. El trabajo ha recorrido el ciclo completo de un proyecto de ciencia de datos aplicado: desde la comprensión del problema de negocio y la extracción de datos reales operacionales hasta la implementación y comparación de seis modelos de predicción y la integración de los resultados en una capa de visualización accesible para el usuario final.

La primera conclusión relevante es la validación empírica de la hipótesis de curva S logística. El análisis exploratorio ha demostrado que el porcentaje de series con R^2 superior a 0.9 en el ajuste logístico es elevado, lo que confirma con datos reales de la empresa lo que hasta ahora se asumía como hipótesis teórica basada en la experiencia del sector. Este hallazgo no solo justifica el modelo M6 paramétrico sino que proporciona una base cuantitativa para argumentar la regularidad del comportamiento productivo de las obras.

La segunda conclusión es que los modelos de machine learning, específicamente LightGBM y Random Forest, son los que ofrecen la mejor relación entre rendimiento y coste de implementación en este contexto. Con un RMSE mediano global de alrededor de 3.5 pp

frente a los 15.26 pp del baseline de la experiencia experta formalizada, representan una mejora del 77% que es directamente trasladable a decisiones de aprovisionamiento más precisas. La similitud de resultados entre ambos modelos, con una diferencia de tan solo 0.06 pp, es en sí misma un resultado relevante: indica que la ventaja del gradient boosting sobre el ensemble de árboles paralelos no se materializa en este problema con el volumen de datos disponible.

La tercera conclusión concierne al modelo LSTM. A pesar de ser el enfoque más sofisticado del trabajo, su rendimiento global queda por detrás de LightGBM y Random Forest. Esto no invalida el modelo sino que lo sitúa en su contexto adecuado: el LSTM alcanza errores de 1.47 pp al 70% de horizonte, muy competitivos con los de los modelos ML, y su arquitectura encoder-decoder está diseñada para escalar mejor cuando el volumen de datos de entrenamiento crezca al incorporar nuevas obras. En el escenario actual de 69 obras, los modelos tabulares aprovechan el dato más eficientemente.

La cuarta conclusión es de naturaleza operativa y conecta directamente con el problema de negocio planteado en la introducción. El sistema desarrollado convierte el conocimiento tácito acumulado durante años por el departamento de estudios en un modelo cuantitativo reproducible y escalable. Cuando la empresa inicia una nueva obra, el sistema puede generar en tiempo real una predicción de cómo va a evolucionar el avance de cada partida presupuestaria a lo largo del tiempo, permitiendo anticipar cuándo y en qué cantidad se va a necesitar cada tipo de material o trabajo. Esto reduce la dependencia del conocimiento individual de los técnicos y proporciona una base objetiva para las decisiones de aprovisionamiento, sin pretender sustituir el criterio experto sino complementarlo.

11.2 TRABAJOS FUTUROS

Los resultados del trabajo abren varias líneas de desarrollo que podrían extender su alcance y aumentar su impacto en producción.

La primera línea es la incorporación de un sistema de alertas automáticas. Una vez disponible la curva predicha y el avance real registrado mes a mes, es relativamente sencillo implementar un mecanismo que detecte cuándo el avance real se desvía de la predicción más de un umbral fijo durante varios meses consecutivos y genere una alerta al responsable de la obra. Este sistema de detección temprana de desviaciones tiene un valor operativo inmediato porque permite actuar sobre los problemas de aprovisionamiento antes de que se traduzcan en paradas de obra.

La segunda línea es la ampliación del dataset. El sistema está diseñado para mejorar progresivamente a medida que se incorporan nuevas obras finalizadas al histórico de entrenamiento. Con más datos, los modelos de machine learning podrán aprender patrones más específicos por tipo de obra y código de compra, y el LSTM podrá materializar su potencial en series largas. Se recomienda implementar un pipeline de reentrenamiento periódico que incorpore automáticamente las obras finalizadas en el último trimestre.

La tercera línea es la migración al entorno productivo Databricks con Azure Data Lake Storage. La arquitectura del sistema está diseñada desde el principio pensando en esta migración, que permitiría procesar el dataset completo sin las limitaciones de memoria del entorno local y orquestar el pipeline de predicción como un proceso batch mensual completamente automatizado.

La cuarta línea es la integración completa con Power BI. En el estado actual del sistema la predicción se genera bajo demanda a través de la interfaz web, pero en el entorno productivo sería más valioso calcular las predicciones de forma batch para todas las obras activas al inicio de cada mes y publicarlas en el datalake para su consumo desde los dashboards de seguimiento de obra ya existentes en la empresa.

La quinta línea es la exploración de arquitecturas más avanzadas para el modelo secuencial. La arquitectura Transformer, que ha demostrado resultados superiores a las LSTM en muchos problemas de series temporales gracias a su mecanismo de atención, podría capturar mejor las dependencias de largo alcance en las curvas de producción y aprovechar de forma más eficiente el contexto de toda la secuencia observada. Su evaluación requeriría un volumen de datos mayor que el disponible actualmente, pero sería factible en el escenario de crecimiento del dataset descrito en la segunda línea.

Capítulo 12. BIBLIOGRAFÍA

[1] Python Software Foundation. (s.f.). *Python 3 documentation*. https://docs.python.org/3/.

[2] GeeksforGeeks. (s.f.). SQL manual. https://cdncontribute.geeksforgeeks.org/wp-content/uploads/SQL-Manual.pdf.

[3] Microsoft. (s.f.). Visual Studio Code documentation. https://code.visualstudio.com/docs.

[4] Microsoft. (s.f.). Pylance. PyPI. https://pypi.org/project/pylance/.

[5] Project Jupyter. (s.f.). Project Jupyter documentation. https://docs.jupyter.org/en/latest/.

[6] Atlassian. (s.f.). Bitbucket guides. https://bitbucket.org/product/guides.

[7] Chacon, S., & Straub, B. (s.f.). Pro Git. Git-scm. https://git-scm.com/book/en/v2.

[8] Databricks. (s.f.). Databricks documentation. https://docs.databricks.com/aws/en/.

[9] Microsoft. (s.f.). Azure SQL Database. https://azure.microsoft.com/es-es/products/azure-sql/database.

[10] Microsoft. (s.f.). SQL Server. https://www.microsoft.com/es-es/sql-server.

[11] Databricks. (s.f.). Data import with JDBC. https://assets.docs.databricks.com/_extras/notebooks/source/data-import/jdbc.html.

[12] JPype Team. (s.f.). JPype documentation. https://jpype.readthedocs.io/en/latest/.

[13] PyPI. (s.f.). JayDeBeApi. https://pypi.org/project/JayDeBeApi/.

[14] LightGBM Team. (s.f.). LightGBM documentation. https://lightgbm.readthedocs.io/en/stable/.

[15] Inesdi. (s.f.). Random Forest: Qué es y cómo funciona. https://www.inesdi.com/blog/random-forest-que-es/.

[16] Codificando Bits. (s.f.). Redes neuronales LSTM. https://codificandobits.com/blog/redes-lstm/.

[17] Arsys. (s.f.). Google Colab: Qué es y cómo usarlo. https://www.arsys.es/blog/google-colab-que-es-y-como-usarlo.

[18] PyTorch Team. (s.f.). PyTorch 2.12 documentation. https://docs.pytorch.org/docs/2.12/index.html.

[19] Streamlit. (s.f.). Streamlit documentation. https://docs.streamlit.io/.

[20] Microsoft. (s.f.). ¿Qué es Azure?. https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-azure.

[21] Streamlit. (s.f.). Streamlit Community Cloud. https://docs.streamlit.io/deploy/streamlit-community-cloud.

[22] Muszynska, K. (2020). The S-curve as a tool for planning and controlling of construction process — Case study. Applied Sciences, 10(6), 2071. https://doi.org/10.3390/app10062071.

[23] Forbes, T., & Riso, T. (2024, 19 de junio). Guide to S-Curve Modeling in Construction. Procore. https://www.procore.com/library/s-curve-modeling-construction.

[24] Chao, L.-C., & Chen, H.-T. (2015). Predicting project progress via estimation of S-curve's key geometric feature values. Automation in Construction, 57, 33–41. https://doi.org/10.1016/j.autcon.2015.04.015.

[25] Gondia, A., Siam, A., El-Dakhakhni, W., & Nassar, A. H. (2020). Machine learning algorithms for construction projects delay risk prediction. Journal of Construction Engineering and Management, 146(1), 04019085. https://doi.org/10.1061/(ASCE)CO.1943-7862.0001736.

[26] Chao, L.-C., & Chien, C.-F. (2009). Estimating project S-curves using polynomial function and neural networks. Journal of Construction Engineering and Management, 135(3), 169–177. https://doi.org/10.1061/(ASCE)0733-9364(2009)135:3(169).

[27] Chao, L.-C., & Chien, C.-F. (2010). A model for updating project S-curve by using neural networks and matching progress. Automation in Construction, 19(1), 84–91. https://doi.org/10.1016/j.autcon.2009.09.006.

[28] Liao, Y. (2026). Data-driven multi-mode time–cost trade-off optimization for construction project scheduling using LightGBM. Processes, 14(8), 1311. https://doi.org/10.3390/pr14081311.

[29] Hwang, S. (2011). Time series models for forecasting construction costs using time series indexes. Journal of Construction Engineering and Management, 137(9), 656–662. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000350.

[30] Wikipedia. (2025). Dynamic time warping. https://en.wikipedia.org/wiki/Dynamic_time_warping.

[31] Alizadeh, E. (2020, octubre 11). An illustrative introduction to dynamic time warping. Towards Data Science. https://towardsdatascience.com/an-illustrative-introduction-to-dynamic-time-warping-36aa98513b98/.

[32] Glock, A.-C., Chmelina, K., Fürnkranz, J., & Hütter, T. (2025). Dynamic time warping for classifying long-term trends in time series. Data & Knowledge Engineering, 159, 102495. https://doi.org/10.1016/j.datak.2025.102495.

[33] Kafritsas, N. (2021, septiembre 14). Time series classification using dynamic time warping. Towards Data Science. https://towardsdatascience.com/time-series-classification-using-dynamic-time-warping-61dcd9e143f6/.

[34] Oracle. (s.f.). Primavera P6 Enterprise Project Portfolio Management. https://www.oracle.com/construction-engineering/primavera-p6/.

[35] RIB Software. (s.f.). Presto: Software de presupuestos y gestión de costes. https://www.rib-software.es/presto.

[36] Microsoft. (s.f.). Administración de proyectos integrada. Microsoft 365. https://www.microsoft.com/es-es/microsoft-365/project/project-management.

[37] ProNovos. (10 de julio de 2025). Understanding S-Curves in construction project forecasting. https://pronovos.com/understanding-s-curves-in-construction-project-forecasting/.

[38] Metodología Agile: la revolución en las formas de trabajo. BBVA. https://www.bbva.com/es/innovacion/metodologia-agile-la-revolucion-las-formas-trabajo/.

[39] Gantt Chart Basics. Asana. https://asana.com/es/resources/gantt-chart-basics.

[40] Naciones Unidas. (s.f.). Objetivos y metas de desarrollo sostenible. Objetivos de Desarrollo Sostenible. https://www.un.org/sustainabledevelopment/es/sustainable-development-goals/.

[41] Naciones Unidas. (s.f.). Trabajo decente y crecimiento económico. Objetivos de Desarrollo Sostenible. https://www.un.org/sustainabledevelopment/es/economic-growth/.

[42] Naciones Unidas. (s.f.). Industria, innovación e infraestructura. Objetivos de Desarrollo Sostenible. https://www.un.org/sustainabledevelopment/es/infrastructure/.

[43] Naciones Unidas. (s.f.). Ciudades y comunidades sostenibles. Objetivos de Desarrollo Sostenible. https://www.un.org/sustainabledevelopment/es/cities/.

[44] Naciones Unidas. (s.f.). Producción y consumo responsables. Objetivos de Desarrollo Sostenible. https://www.un.org/sustainabledevelopment/es/sustainable-consumption-production/.

[45] NumPy Developers. (s.f.). NumPy documentation. https://numpy.org/devdocs/.

[46] The pandas development team. (s.f.). pandas documentation. https://pandas.pydata.org/docs/.

[47] The Matplotlib development team. (s.f.). Matplotlib documentation. https://matplotlib.org/stable/index.html.

[48] Waskom, M. (s.f.). seaborn: statistical data visualization. https://seaborn.pydata.org/.

[49] SciPy Developers. (s.f.). SciPy documentation. https://docs.scipy.org/doc/scipy/.

[50] Scikit-learn developers. (s.f.). scikit-learn documentation. https://scikit-learn.org/stable/.

[51] LightGBM Contributors. (s.f.). LightGBM documentation. https://lightgbm.readthedocs.io/en/latest/index.html.

[52] PyTorch Team. (s.f.). PyTorch documentation. https://docs.pytorch.org/docs/2.12/index.html.

[53] Meert, W. (s.f.). DTAIDistance documentation. https://dtaidistance.readthedocs.io/en/latest/.

[54] JPype contributors. (s.f.). JPype documentation. https://jpype.readthedocs.io/en/latest/.

[55] JayDeBeApi contributors. (s.f.). JayDeBeApi. https://pypi.org/project/JayDeBeApi/.

[56] Python Software Foundation. (s.f.). re — Regular expression operations. https://docs.python.org/3/library/re.html.

[57] Python Software Foundation. (s.f.). itertools — Functions creating iterators for efficient looping. https://docs.python.org/3/library/itertools.html.

ANEXO A: ALINEACIÓN DEL PROYECTO CON LOS ODS

Los Objetivos de Desarrollo Sostenible (ODS) [40] constituyen un conjunto de 17 metas globales definidas por las Naciones Unidas en el marco de la Agenda 2030 (Tabla 9), orientadas a promover un desarrollo equilibrado desde el punto de vista social, económico y ambiental. Estos objetivos abordan desafíos fundamentales como la salud y el bienestar, la reducción de desigualdades, la innovación tecnológica y el fortalecimiento de las instituciones, estableciendo un marco común de actuación para gobiernos, organizaciones y entidades privadas. En el ámbito de los proyectos de ingeniería, la alineación con los ODS permite evaluar el impacto social y ético de las soluciones desarrolladas, asegurando que la aplicación de la tecnología contribuya de forma responsable a la mejora de la calidad de vida y al desarrollo sostenible.

Tabla 9 - ODS. Elaboración propia

ODS	*Objetivo*
ODS 1. Fin de la pobreza	Erradicar la pobreza en todas sus formas y en todo el mundo.
ODS 2. Hambre cero	Poner fin al hambre, lograr la seguridad alimentaria y promover una agricultura sostenible.
ODS 3. Salud y bienestar	Garantizar una vida sana y promover el bienestar para todas las personas en todas las edades.

ODS	*Objetivo*
ODS 4. Educación de calidad	Garantizar una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje permanente.
ODS 5. Igualdad de género	Lograr la igualdad entre los géneros y empoderar a todas las mujeres y niñas.
ODS 6. Agua limpia y saneamiento	Garantizar la disponibilidad y la gestión sostenible del agua y el saneamiento para todos.
ODS 7. Energía asequible y no contaminante	Garantizar el acceso a una energía asequible, segura, sostenible y moderna.
ODS 8. Trabajo decente y crecimiento económico	Promover el crecimiento económico sostenido, inclusivo y el empleo digno para todos.
ODS 9. Industria, innovación e infraestructura	Desarrollar infraestructuras resilientes y fomentar la innovación y la industrialización sostenible.
ODS 10. Reducción de las desigualdades	Reducir la desigualdad dentro de los países y entre ellos.
ODS 11. Ciudades y comunidades sostenibles	Lograr que las ciudades y comunidades sean inclusivas, seguras, resilientes y sostenibles.
ODS 12. Producción y consumo responsables	Garantizar modalidades de consumo y producción sostenibles.
ODS 13. Acción por el clima	Adoptar medidas urgentes para combatir el cambio climático y sus efectos.

ODS	*Objetivo*
ODS 14. Vida submarina	Conservar y utilizar de forma sostenible los océanos, mares y recursos marinos.
ODS 15. Vida de ecosistemas terrestres	Proteger y restaurar los ecosistemas terrestres y detener la pérdida de biodiversidad.
ODS 16. Paz, justicia e instituciones sólidas	Promover sociedades pacíficas, justas e inclusivas con instituciones eficaces.
ODS 17. Alianzas para lograr los objetivos	Fortalecer los medios de implementación y revitalizar la alianza mundial para el desarrollo sostenible.

El proyecto desarrollado se alinea con 4 de los ODS explicados anteriormente. En primer lugar con el ODS 8: Trabajo decente y crecimiento económico [41], al contribuir a la mejora de la eficiencia operativa en el sector de la construcción residencial mediante la optimización de la planificación del aprovisionamiento de materiales. La automatización de la predicción de curvas de producción por código de compra permite a los departamentos de compras reducir el tiempo dedicado a la estimación manual de necesidades, apoyando la toma de decisiones en datos históricos objetivos en lugar de en el criterio individual de cada técnico. Asimismo, la reducción del sobrestock y de las paradas de obra por falta de suministro se traduce en una mejora directa de la productividad y de las condiciones de trabajo en obra, contribuyendo a un crecimiento económico más sostenido y eficiente dentro del sector.

También contribuye al ODS 9: Industria, innovación e infraestructura [42] mediante la aplicación de técnicas avanzadas de machine learning y big data a un sector tradicionalmente poco digitalizado como es la construcción residencial. La implementación de una arquitectura basada en SQL Server, Python y Streamlit, diseñada para su migración a un entorno productivo con Azure Data Lake Storage y Databricks, supone un ejemplo de

modernización de la infraestructura digital empresarial. Además, el desarrollo de una solución modular y escalable, con seis modelos de predicción comparados bajo un protocolo de evaluación riguroso, fomenta la adopción de prácticas innovadoras que pueden ser replicadas en otras empresas constructoras, impulsando la transformación digital del sector.

En cuanto al ODS 11: Ciudades y comunidades sostenibles [43], el proyecto contribuye indirectamente a la mejora de la eficiencia en la construcción de vivienda residencial, un sector clave para garantizar el acceso a ciudades inclusivas y bien dotadas de infraestructura. Una planificación más precisa del aprovisionamiento de materiales permite reducir los retrasos en la entrega de obra, con impacto directo sobre los plazos de disponibilidad de vivienda. La reducción del desperdicio derivado de sobrecompras y de la gestión ineficiente del stock contribuye además a una construcción más racional en el uso de recursos, alineada con el objetivo de ciudades y comunidades más sostenibles.

Finalmente, también se vincula con el ODS 12: Producción y consumo responsables [44] a través de la optimización del ciclo de aprovisionamiento de materiales en obra. La predicción precisa de cuándo y en qué cantidad se necesitará cada partida presupuestaria permite ajustar los pedidos a la demanda real, reduciendo tanto el exceso de compra como el riesgo de obsolescencia o deterioro de materiales almacenados. De este modo, el sistema desarrollado contribuye a una gestión más eficiente y responsable de los recursos materiales en el proceso constructivo, minimizando el impacto ambiental asociado a la sobreproducción y al desperdicio en la cadena de suministro.

ANEXO B: LIBRERÍAS UTILIZADAS

En la Tabla 10 se muestran todas las librerías que se han utilizado en este proyecto, incluyendo una breve descripción del uso que se le ha dado a cada una.

Tabla 10 - Librerías utilizadas

Librería	*Tipo*	*Función*
NumPy [45]	Cálculo numérico	Operaciones matemáticas y manipulación de arreglos multidimensionales.
Pandas [46]	Manipulación de datos	Estructuras de datos (DataFrames) para limpieza y análisis.
Matplotlib [47] / Seaborn [48]	Visualización	Creación de gráficos estadísticos y representaciones visuales.
SciPy [49]	Computación científica	Optimización, interpolación, estadística y clustering jerárquico.
Scikit-learn [50]	Aprendizaje automático	Modelos predictivos (RandomForest), clustering, preprocesamiento y métricas.
LightGBM [51]	Machine Learning	Algoritmo de gradient boosting de alto rendimiento.
PyTorch [52]	Deep Learning	Creación, entrenamiento y gestión de redes neuronales.

Librería	*Tipo*	*Función*
Dtaidistance [53]	Análisis temporal	Cálculo de distancia de deformación temporal (DTW).
Jpype [54] / jaydebeapi [55]	Conectividad	Interfaz para conectar Python con bases de datos vía JDBC (Java).
Re [56]	Estándar de Python	Procesamiento de texto mediante expresiones regulares.
Itertools [57]	Estándar de Python	Funciones para la creación de iteradores eficientes.