

# PROYECTO FINAL TRADEMIND

TFM: INTELIGENCIA ARTIFICIAL Y BIG DATA

Mata Rodríguez, Yeray  
Ramírez Vaquero, Daniel  
Del Rosal Aguirre, Tristán Enrique

04/03/2025



# TABLA DE CONTENIDO

01. INTRODUCCIÓN	02. OBJETIVOS	03. RECOLECCIÓN, LIMPIEZA Y VISUALIZACIÓN DE DATOS
04. METODOLOGÍA	05. DESARROLLO	06. RESULTADOS Y ANÁLISIS
07. IMPLEMENTACIÓN	08. CONCLUSIONES	09. BIBLIOGRAFÍA

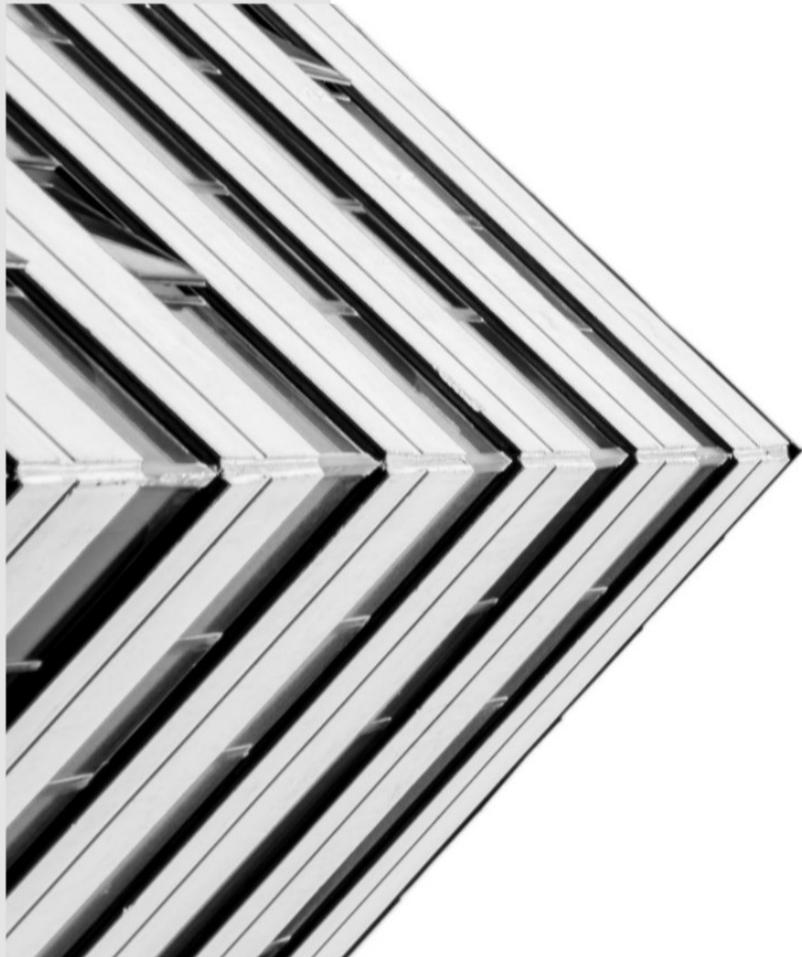




# INTRODUCCIÓN 01



## 01. INTRODUCCIÓN



### TRADEMIND

Trademind es un proyecto orientado a la predicción de precios de teléfonos móviles en el mercado de segunda mano. Surge ante la necesidad de contar con un sistema capaz de analizar datos históricos y variables relevantes para estimar el precio de reventa de dispositivos. Con ello, se busca optimizar los procesos de compra-venta y ayudar a usuarios y empresas a tomar decisiones más informadas.





# OBJETIVOS 02



## 02. OBJETIVOS



### GENERALES

- ✓ Predecir precios de móviles de segunda mano.
- ✓ Aplicar IA y Big Data en el análisis.
- ✓ Crear una interfaz de consulta interactiva.
- ✓ Comparar modelos de predicción.

### ESPECÍFICOS

- ◆ Agrupar y limpiar datos de precios.
- ◆ Evaluar distintos enfoques de predicción.
- ◆ Desarrollar la interfaz de usuario.
- ◆ Optimizar la precisión del modelo.



# RECOLECCIÓN, LIMPIEZA Y VISUALIZACIÓN DE DATOS

# 03

# RECOLECCIÓN Y LIMPIEZA



De donde hemos obtenido los datos?



## 1º ADICIÓN DE HEADERS

Los CSVs no traen headers por lo que hay que identificar los datos que incluye. Adicionalmente cortamos columnas añadidas a futuro, para evitar errores.



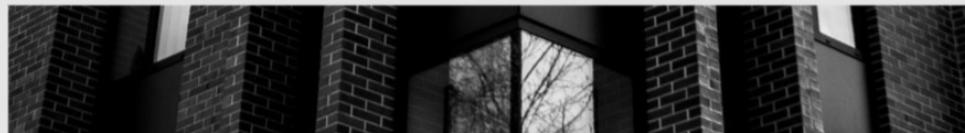
## 2º UNIÓN DE CSVs

Compactamos todos los CSVs en un unico CSV.



## 3º FILTRADO DE COLUMNAS

Eliminamos todas las columnas innecesarias del CSV dejando solamente las utiles.



# RECOLECCIÓN Y LIMPIEZA



## ALiqindoi

Los datos no han sido recolectados en el transcurso de este proyecto, han sido proporcionados por ALiqindoi Circular Solutions SL.



## 1º ADICIÓN DE HEADERS

Los CSVs no traen headers por lo que hay que identificar los datos que incluye. Adicionalmente cortamos columnas añadidas a futuro, para evitar errores.



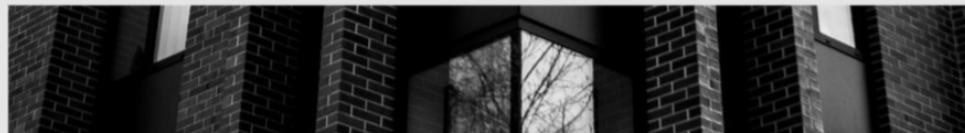
## 2º UNIÓN DE CSVs

Compactamos todos los CSVs en un único CSV.



## 3º FILTRADO DE COLUMNAS

Eliminamos todas las columnas innecesarias del CSV dejando solamente las útiles.



# RECOLECCIÓN Y LIMPIEZA



## 4º FILTRADO POR TIPOS

Entre los datos existentes, tenemos teléfonos, pero tambien un pequeña cantidad de otros tipos de dispositivos.

Filtraremos y solo guardaremos los teléfonos.



## 7º MODIFICACIÓN Y ADICIÓN DE COLUMNAS

En este paso añadimos datos de valor como "fecha de lanzamiento", "día de la semana",....

Pasamos columnas a tipo int para facilitar su uso por parte del modelo, tales como "color", "almacenamiento",....

Realizamos label encoding en la columna "modelo".



## 8º TIPADO DE COLUMNAS

Con el fin de evitar errores y optimizar el dataset, asignamos un tipado optimo a cada columna.



## 9º DIVISIÓN POR GRADOS Y MARCAS

Subdividimos cada fila del dataset en sus distintos grados y los agrupamos.

Posteriormente, dividiremos este CSV y generamos 1 CSV por marca.



## 6º RELLENADO DE NULOS

El dataset contiene vacios temporales en los precios de algunos dispositivos, con el uso de KNN hemos llenado estos espacio.





# METODOLOGÍA 04



## 04. METODOLOGÍA



### Algoritmos de Series Temporales

- ARIMA
- SARIMA
- SARIMAX



### Redes Neuronales

- LSTM



### Algoritmos de Aprendizaje Automatico

- XGBoost
- Random Forest

## 04. METODOLOGÍA



### Descartado

No captura bien la naturaleza escalonada de los precios.

Necesidad de entrenar un modelo por cada variante de teléfono, mas de 700



### Descartado

Las predicciones son demasiado lineales debido a que LSTM detecta los cambios bruscos como ruido.



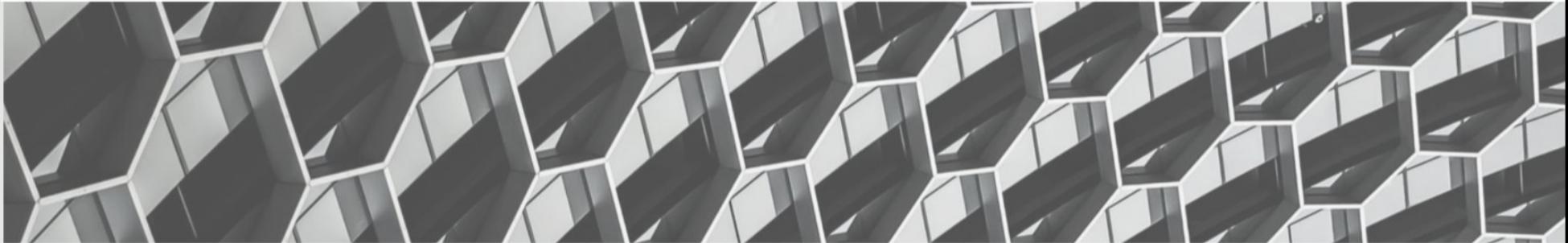
### Seleccionado

Mejor capacidad para manejar datos estacionarios con cambios bruscos en el tiempo.



# DESARROLLO 05

## 04. DESARROLLO



### CONFIGURACIÓN TÉCNICA Y ENTORNO DE PROGRAMACIÓN

Desarrollado en VScode  
Usando entorno virtual de  
Python3.11.9



### ORGANIZACIÓN DE ARCHIVOS Y FLUJO DE TRABAJO

Todo el desarrollo se a  
unificado en un mismo  
cuaderno Jupyter.



### AJUSTES Y OPTIMIZACIÓN DEL MODELO SELECCIONADO

Comprobación de los  
resultados y ajuste de  
hiperparametros.



### VERSIONES ENTRENADAS DEL MODELO

Como resultado tenemos 9  
modelos guardados para cada  
marca.

## 04. DESARROLLO



```
# Configuración del modelo XGBoost con parámetros optimizados
modelo_marca = xgb.XGBRegressor(
    objective="reg:squarederror",      # Función de pérdida para problemas de regresión
    n_estimators=3000,                 # Número total de árboles en el modelo
    learning_rate=0.015,               # Tasa de aprendizaje pequeña para ajuste fino
    max_depth=9,                     # Profundidad máxima de cada árbol
    min_child_weight=2,               # Mínima suma de pesos necesaria en un nodo hijo
    subsample=0.8,                   # Fracción de muestras usadas en cada árbol (previene sobreajuste)
    colsample_bytree=0.85,            # Fracción de columnas usadas por cada árbol
    early_stopping_rounds=50,         # Parar entrenamiento si no hay mejora en 50 rondas
    gamma=0.5,                       # Reducción mínima de pérdida para hacer una partición
    reg_lambda=3,                    # Regularización L2
    reg_alpha=1.5,                  # Regularización L1 para modelos más parsimoniosos
    max_delta_step=2,                # Mejora estabilidad para datos desbalanceados
    scale_pos_weight=0.8,             # Balance para clases (precios) desproporcionadas
    tree_method="gpu_hist",          # Usa GPU para acelerar el entrenamiento si está disponible
    random_state=42                  # Semilla para reproducibilidad de resultados
```

CONFIGURACIÓN  
ENTORNO DE

Desarrollado  
Usando ento

Python

ADAS

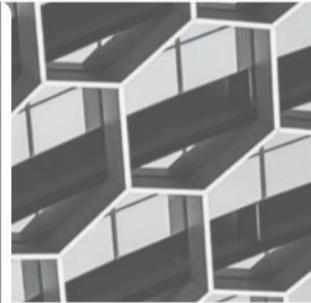
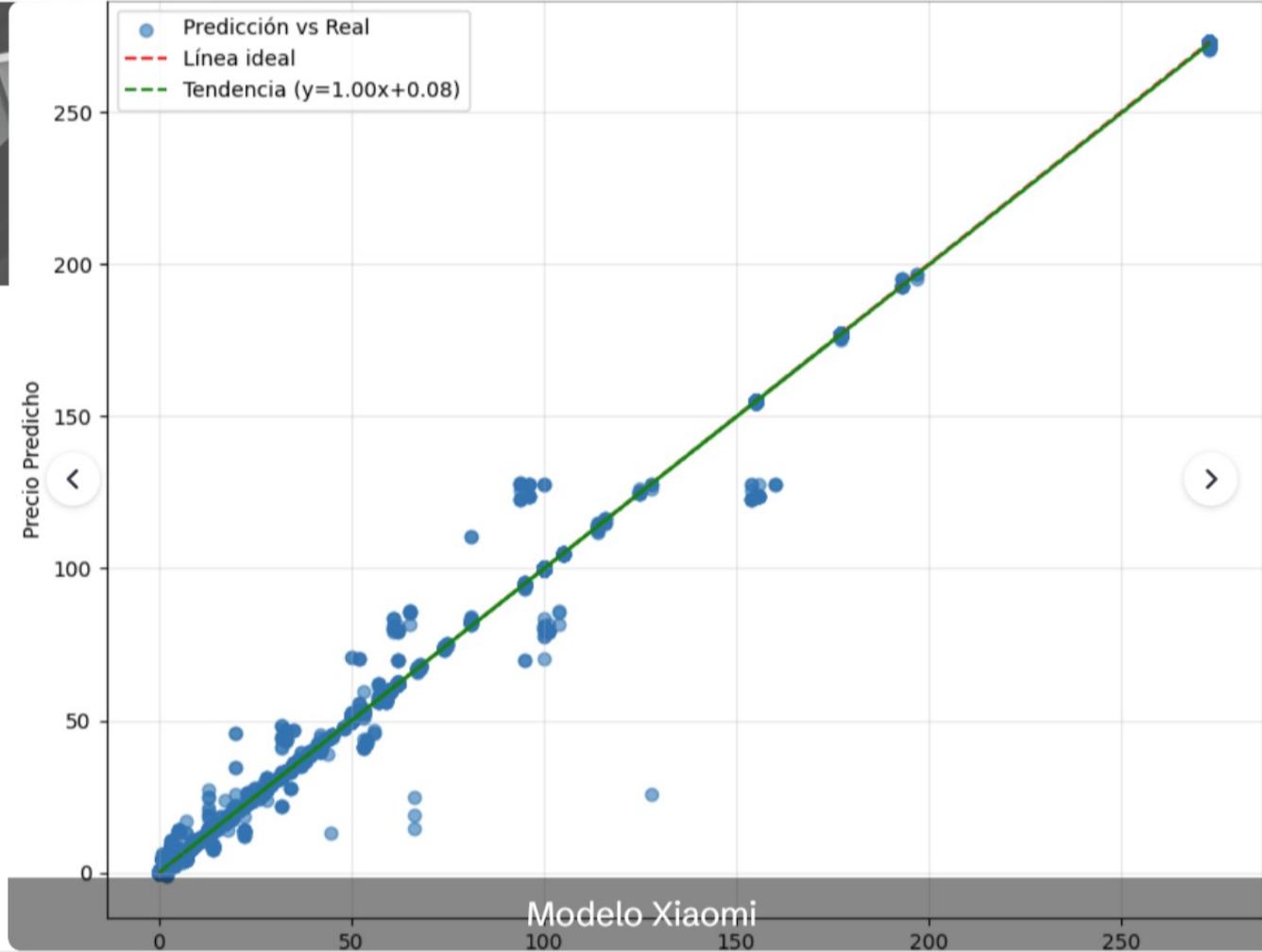
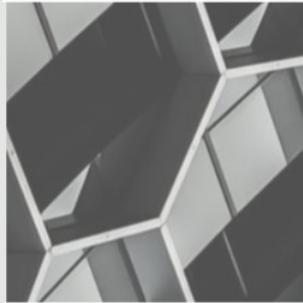
nos 9  
a cada



# RESULTADOS Y ANALISIS 06

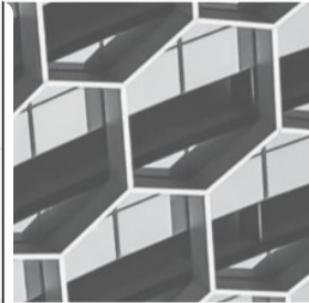
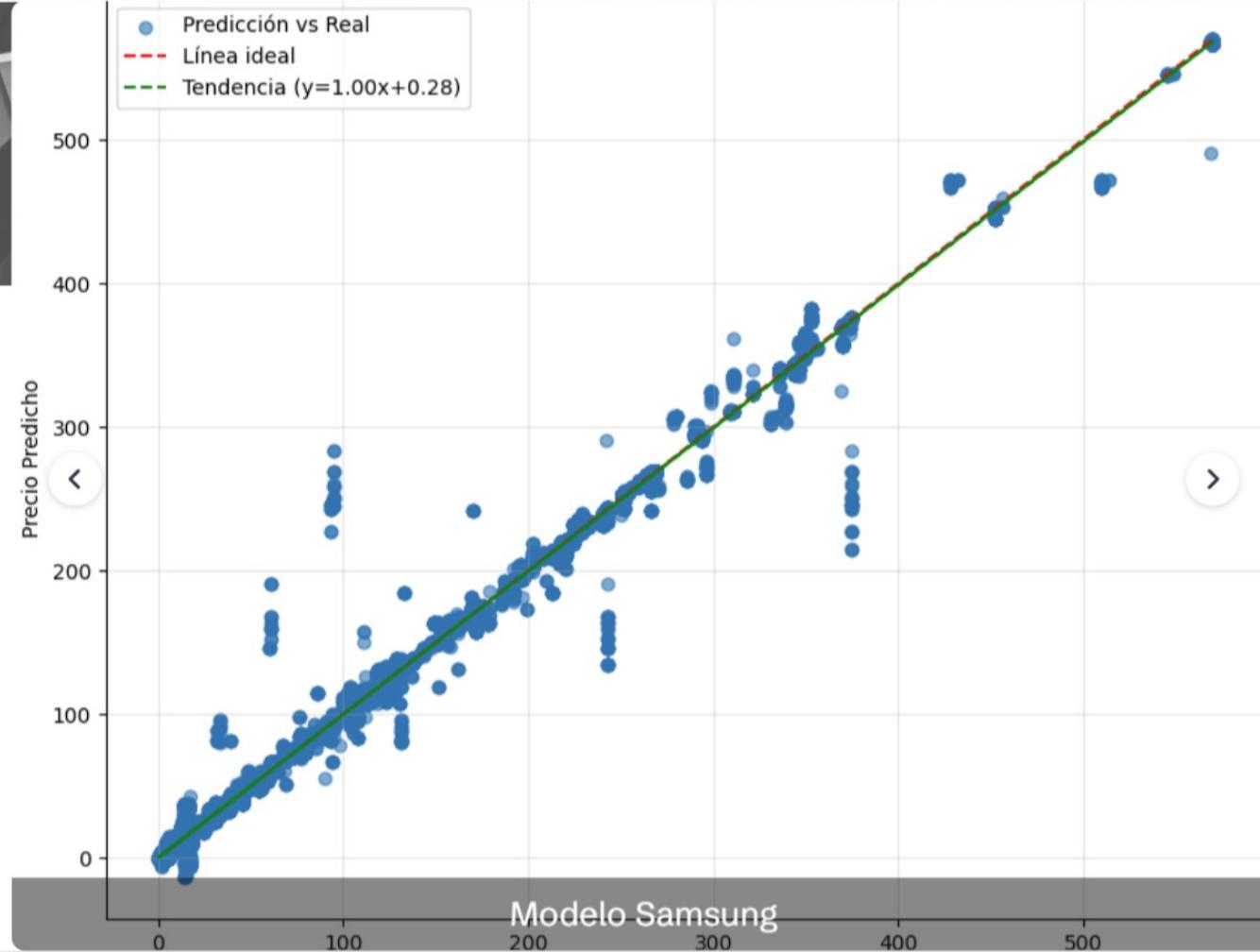
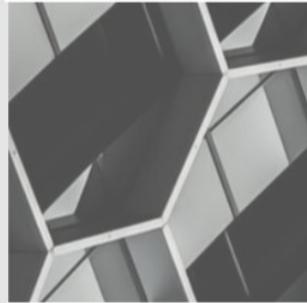


## 06. RESULTADOS



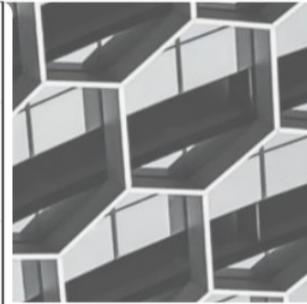
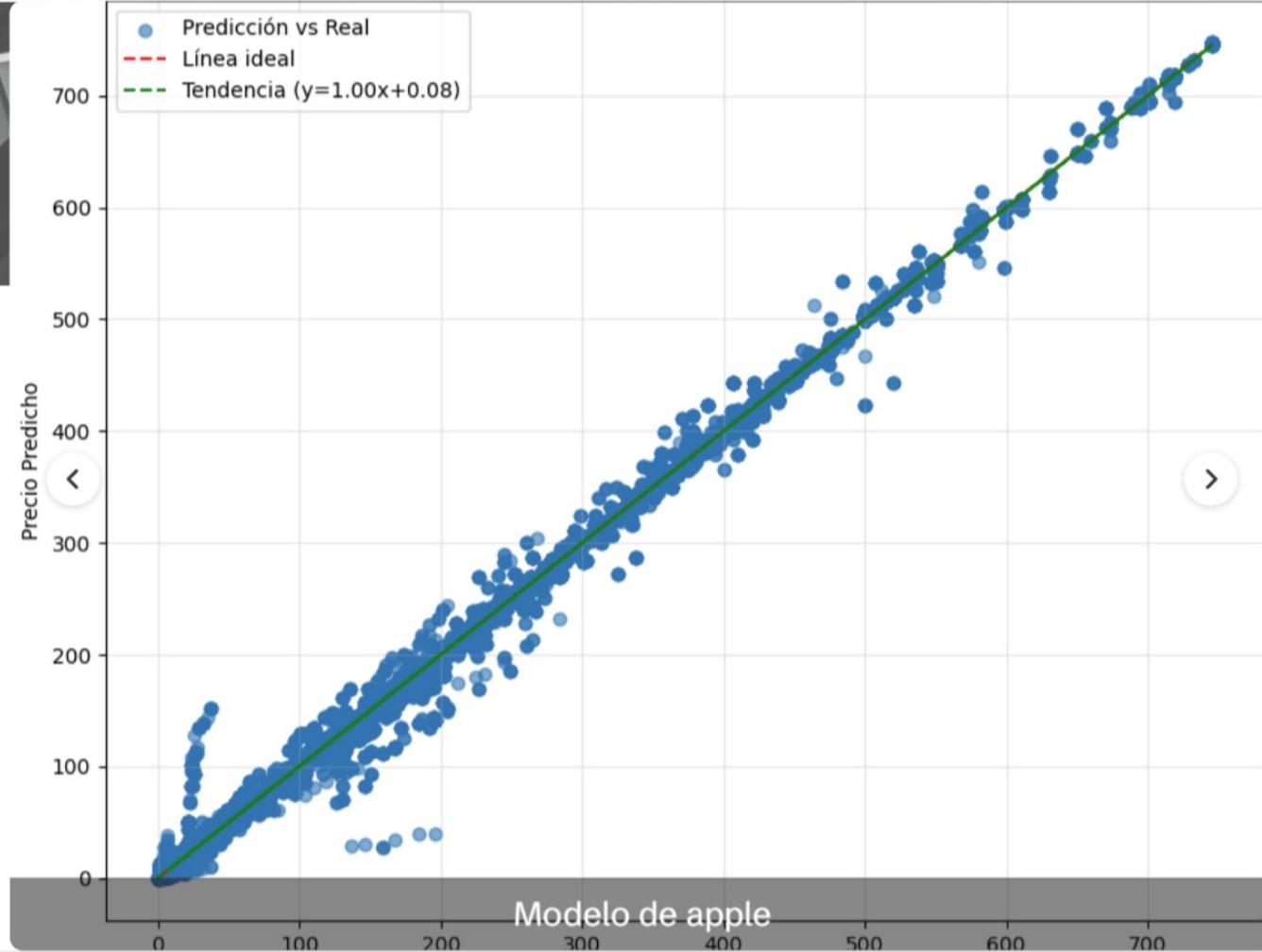
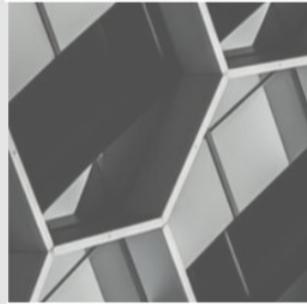
≡

## 06. RESULTADOS



≡

## 06. RESULTADOS



≡



# IMPLEMENTACIÓN 07

## 07. IMPLEMENTACIÓN



The screenshot shows a mobile application interface for "TradeMind". At the top, there's a header bar with the text "Venta de iPhone 12 Pro Max 256GB" and the "TradeMind" logo. On the left, a sidebar titled "Mis Conversaciones" lists recent conversations: "Venta de iPhone 12 Pro Max..." (Hoy) and "Venta de iPhone 13 Pro Max..." (Hoy). The main area is a chat window with the following exchange:

**¡Hola! Soy TradeMind 🤖**  
Agente especializado en compra-venta de smartphones  
¿En qué te puedo ayudar?

**Hola quiero vender mi iphone 12 pro max de 256gb**

:Perfecto! Para poder darte una estimación de precio, necesito evaluar el estado de tu iPhone 12 Pro Max de 256GB. Vamos a revisar tres aspectos: la pantalla, el cuerpo del dispositivo y su funcionalidad general.

Empecemos con la pantalla.

**Evaluación del Dispositivo**

**Estado de la Pantalla**

Podrías decirme si la pantalla presenta alguna de estas características?

- Grietas o roturas
- Rasguños visibles
- Problemas con la pantalla táctil

## CREACIÓN DE LA INTERFAZ DE USUARIO.

La aplicación cuenta con una única ventana, donde podemos interactuar con el agente de IA a través de un chat. A la izquierda, se muestra un historial con las conversaciones recientes.





# FRAMEWORK WEB, LIBRERÍAS UTILIZADAS Y BBDD

## ANGULAR

Se utiliza para la interacción con el usuario y la visualización de los resultados. Permite una experiencia dinámica y adaptable en la consulta de predicciones.

## FAST API & POSTGRESQL

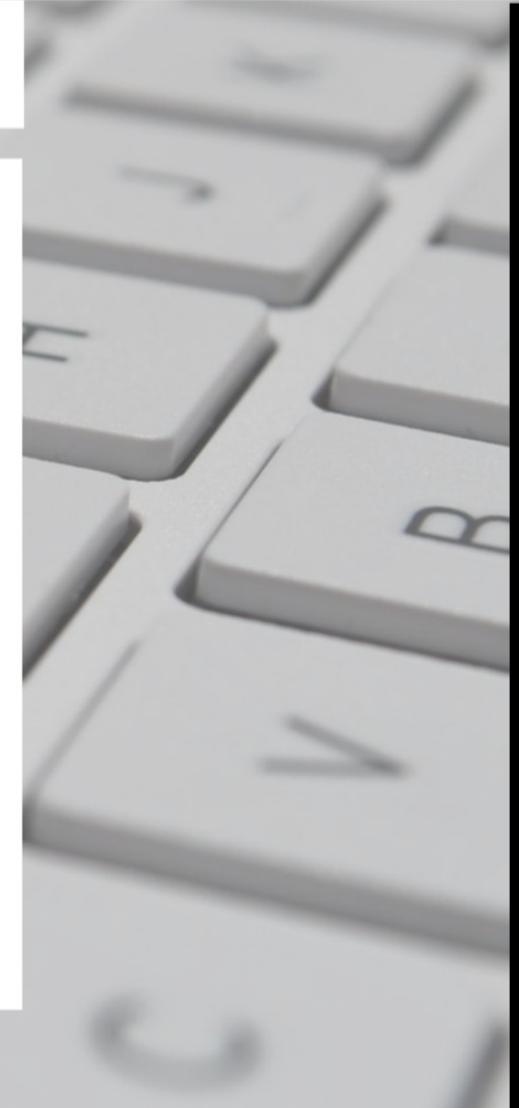
Se encarga de la recepción de solicitudes, la comunicación con el agente y gestión de la BBDD.

## LANGGRAPH Y LANGCHAIN

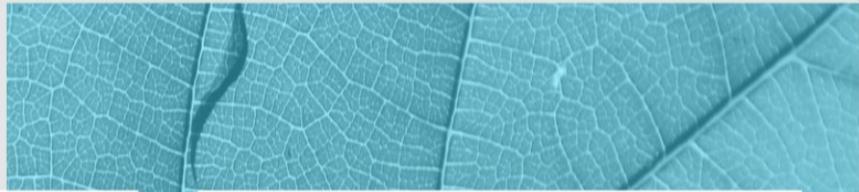
Se encarga de la orquestación avanzada de modelos de lenguaje con flujos conversacionales estructurados y agentes inteligentes.

## MATPLOTLIB Y SEABORN

Se utilizan para generar gráficos de tendencias y análisis visuales de las predicciones.



## 07. IMPLEMENTACIÓN



# FUNCIONALIDADES FINALES

## VENTA DE DISPOSITIVOS

El modelo recopila la información clave del dispositivo: modelo, marca, almacenamiento y realiza una serie de preguntas para determinar el grado de deterioro del dispositivo. A partir de estos datos, genera una predicción del precio de venta estimado en el mercado de segunda mano, proporcionando una referencia útil para vendedores.

## RECOMENDACIÓN DE DISPOSITIVOS

El usuario introduce un presupuesto y, opcionalmente, puede especificar modelo, marca, almacenamiento y grado de deterioro. Si se proporcionan todos los datos, la recomendación será más precisa; si no, el sistema explorará todas las combinaciones posibles y seleccionará el mejor dispositivo disponible dentro del rango de precio indicado.

## GENERACIÓN DE GRÁFICAS

Cuando el usuario lo solicita explícitamente, el agente genera una gráfica de evolución de precios en el rango temporal deseado para el dispositivo seleccionado. Esto permite visualizar tendencias, identificar patrones y tomar decisiones más informadas en la compra o venta de móviles.



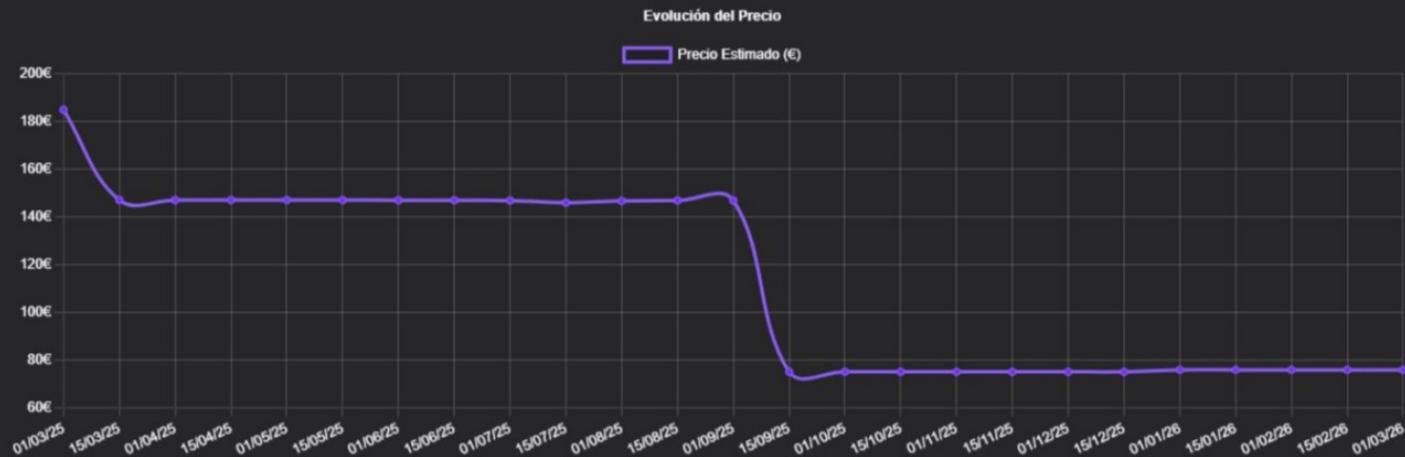
# 07. IMPLEMENTACIÓN



## Gráfica de Precios

- Dispositivo: Apple iPhone 12 Pro Max 256GB
- Rango de fechas: Del 01-03-2025 al 01-03-2026
- Porcentaje de depreciación: -59.02%

Esto significa que el valor de tu dispositivo ha disminuido en un 59.02% durante el último año. ¿Hay algo más en lo que te pueda ayudar?



## VENTA DE

El modelo resulta clave del dispositivo. Almacenamiento preguntas para el deterioro de estos datos. del precio de mercado proporcionar para vendedores.

## ICAS

a genera una ciencias en el para el esto as, r en la





CONCLUSIONES Y TRABAJO FUTURO

08



# 08. CONCLUSIONES Y TRABAJO FUTURO



## ◆ SÍNTESIS DEL PROYECTO

Hemos desarrollado un modelo de predicción de precios para móviles de segunda mano, aplicando técnicas de Machine Learning y Big Data. A través de la exploración de distintos enfoques, seleccionamos XGBoost como la mejor opción por su precisión y capacidad de manejar datos con cambios abruptos.

## ◆ RESULTADOS CLAVE

El modelo final proporciona predicciones precisas basadas en datos históricos, permitiendo a los usuarios estimar el valor de un dispositivo en el mercado de reventa. La implementación en una interfaz interactiva facilita la consulta y visualización de precios.

## ◆ LIMITACIONES Y MEJORAS

A pesar del éxito del modelo, la precisión podría mejorar incorporando más variables como tendencias de mercado, volumen de ventas o fechas de lanzamiento de nuevos modelos. También se podría optimizar el rendimiento del sistema y ampliar su uso a otros tipos de dispositivos.



# BIBLIOGRAFÍA 09

## REFERENCIAS BIBLIOGRÁFICAS

- Documentacion oficial de xgboost  
<https://xgboost.readthedocs.io/en/stable/>
- Documentacion de la API de Chat-GPT  
<https://platform.openai.com/docs/api-reference/introduction>
- Documentacion de Angular 19  
<https://angular.dev/>
- Langraph y Langchain  
<https://www.langchain.com/langgraph>
- Prime NG  
<https://primeng.org/>
- Chat-GPT, Claude & Copilot  
Mención honorifica

