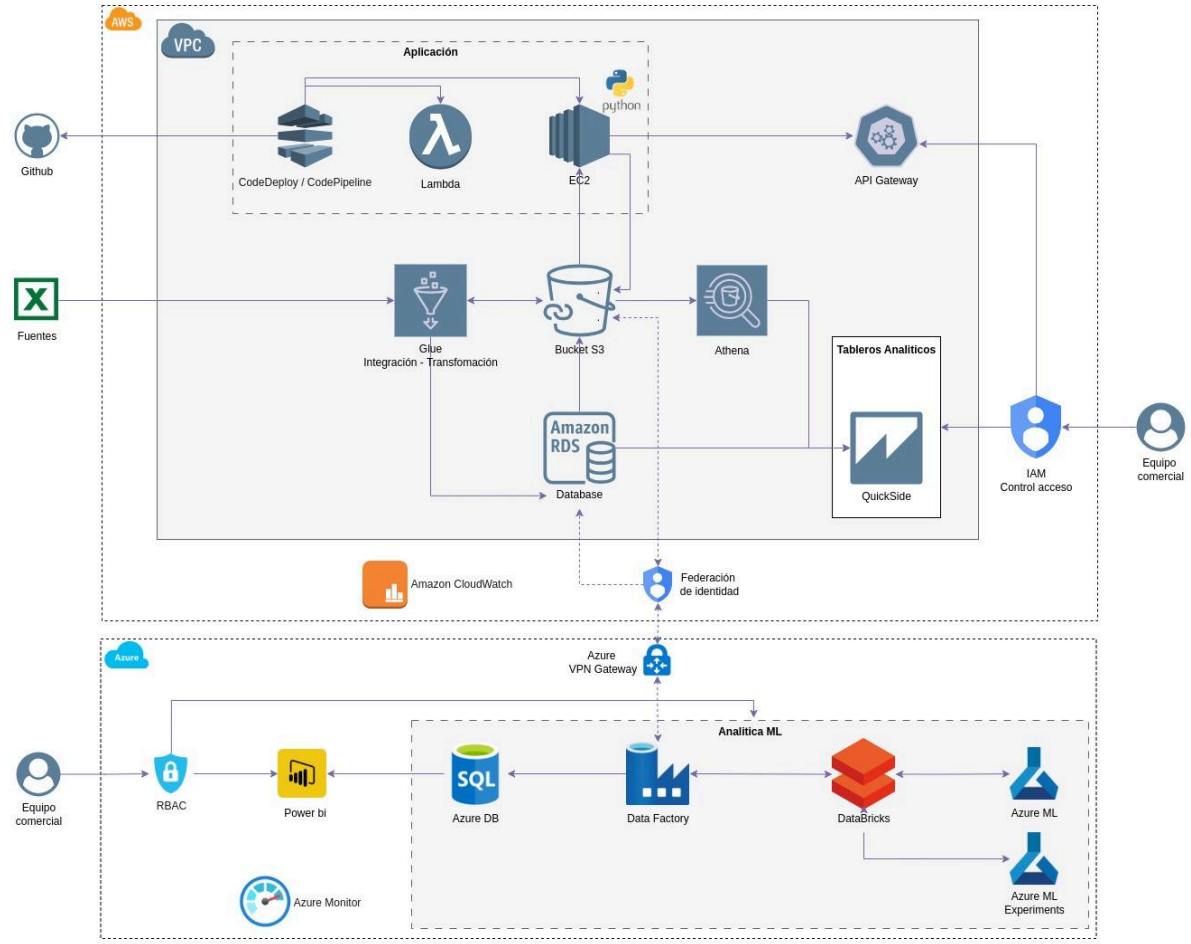
Parte 1. Diseño de Arquitectura

Documento Técnico – Arquitectura Híbrida AWS + Azure con Analítica y ML

1. Diagrama arquitectónico

El diseño integra servicios de **procesamiento, almacenamiento, despliegue, analítica y machine learning**, distribuidos entre AWS y Azure, conectados por una **VPN Gateway segura** y con autenticación federada entre identidades.



2. Componentes principales

En AWS

Componente	Función

GitHub	Repositorio de código fuente para la aplicación.
CodeDeploy / CodePipeline	CI/CD para desplegar funciones Lambda o instancias EC2.
Lambda / EC2 (Python)	Backend de la aplicación.
API Gateway	Expone servicios REST para el frontend u otros consumidores.
Amazon RDS	Almacenamiento estructurado de datos (relacional).
Glue	ETL para transformar y mover datos hacia S3.
S3	Lago de datos (almacenamiento de objetos).
Athena	Consulta interactiva sobre datos en S3 mediante SQL.
QuickSight	Visualización de dashboards analíticos para usuarios de negocio.
IAM	Control de acceso basado en roles (RBAC).
CloudWatch	Monitoreo de logs, métricas y alertas.
Federación de identidad	Permite que usuarios autenticados desde Azure accedan a AWS.

En Azure

Componente	Función
Azure VPN Gateway	Canal seguro de comunicación entre nubes.
RBAC (Azure AD)	Gestión de permisos por roles para Power BI y DBs.
Power BI	Visualización avanzada, acceso a modelos en Azure y AWS.
Azure SQL DB	Repositorio adicional o intermedio para analítica.
Data Factory	Orquestación de datos desde AWS hacia Azure.
Databricks	Procesamiento avanzado, transformación, preparación de datos.
Azure ML + Experiments	Entrenamiento, validación y despliegue de modelos de machine learning.
Azure Monitor	Seguimiento de rendimiento, estado y seguridad de recursos.

3. Justificación de las tecnologías seleccionadas

Necesidad	Tecnología seleccionada	Justificación
CI/CD para despliegue rápido	CodePipeline + GitHub	Integración robusta y escalable para DevOps
Backend flexible y eficiente	Lambda + EC2 + API Gateway	Alternancia entre serverless y servidores dedicados
ETL y almacenamiento	AWS Glue + S3	Transformación de datos con alto rendimiento y bajo costo
Análisis en tiempo casi real	Athena + RDS + QuickSight	Consulta eficiente sobre S3 y RDS sin carga adicional
Dashboard multiplataforma	Power BI + QuickSight	Soporte para usuarios en ambas nubes
Procesamiento distribuido	Azure Databricks	Integración de Spark con escalabilidad nativa
Ciencia de datos e IA	Azure ML	Gestión centralizada de modelos con MLOps
Seguridad y acceso federado	IAM + RBAC + VPN + Federation	Control granular de accesos con conexión segura y auditoría

---

4. Beneficios técnicos y de negocio

Técnicos

- **Interoperabilidad cloud**: permite usar lo mejor de AWS y Azure sin depender de un solo proveedor.
- **Escalabilidad horizontal**: funciones serverless, clusters Spark y bases en la nube permiten escalar según demanda.
- **Automatización completa del ciclo de vida del dato**: desde la extracción hasta el consumo en dashboards y APIs.
- **Seguridad centralizada**: con RBAC, IAM y federación de identidades, se asegura trazabilidad y control de accesos.

De negocio

- **Mejor toma de decisiones**: dashboards en tiempo real (QuickSight y Power BI) permiten análisis dinámico.

- **Aprovechamiento de licenciamiento**: reutiliza suscripciones de Power BI e infraestructura existente.
- **Reducción de costos operativos**: automatización reduce trabajo manual y errores.
- **Adaptabilidad tecnológica**: plataforma abierta para agregar más fuentes, modelos y usuarios sin rediseño.

Propuesta Comercial y Ejecutiva

Solución Híbrida Multinube con Analítica y Machine Learning

3 de Junio del 2025

1. Resumen Ejecutivo

Presentamos una solución integral de **analítica avanzada y machine learning** construida sobre una **arquitectura híbrida entre AWS y Microsoft Azure**, que permite a la organización centralizar, procesar y visualizar sus datos desde diversas fuentes, con altos niveles de automatización, seguridad y escalabilidad.

Esta propuesta permite mejorar la **toma de decisiones basada en datos**, mediante dashboards en **QuickSight y Power BI**, así como impulsar capacidades predictivas a través de modelos de machine learning entrenados y desplegados en **Azure ML**.

2. Descripción de la Solución

La solución está compuesta por tres capas principales:

🔹 Capa de procesamiento y despliegue (AWS)

- Automatización del despliegue de aplicaciones usando **CodePipeline y Lambda**.
- Procesamiento de datos mediante **AWS Glue** y almacenamiento en **S3**.
- Análisis con **Athena** y visualización en **Amazon QuickSight**.
- Control de acceso granular con **IAM** y monitoreo con **CloudWatch**.

🔹 Capa de analítica avanzada y ML (Azure)

- Conexión segura con AWS mediante **Azure VPN Gateway**.
- Orquestación y transformación de datos con **Data Factory** y **Databricks**.
- Entrenamiento y operación de modelos con **Azure Machine Learning**.
- Acceso a dashboards de negocio mediante **Power BI**, controlado por **Azure RBAC**.

🔹 Interoperabilidad y seguridad

- Federación de identidad entre Azure AD y AWS IAM.
- Gestión unificada de usuarios y roles.
- Trazabilidad y monitoreo continuo en ambas nubes.

3. Beneficios de la Solución

Técnicos

- **Multinube real**: combina lo mejor de Azure y AWS sin bloqueo de proveedor.
- **Escalabilidad nativa**: servicios elásticos según demanda.
- **Automatización** de flujos de datos, despliegue y machine learning.
- **Alto nivel de seguridad y cumplimiento** (auditoría, control de accesos, cifrado).

De negocio

- **Acceso oportuno y visual a la información** para usuarios estratégicos.
- **Capacidad predictiva** para anticipar comportamientos y optimizar recursos.
- **Aprovechamiento de inversiones existentes** (licencias Power BI, infraestructura AWS).
- **Reducción de costos operativos** y mayor eficiencia en TI.

4. Alcance de la Propuesta

- Integración de fuentes de datos heterogéneas (archivos, bases de datos, APIs).
- Implementación de pipelines de datos en AWS y Azure.
- Entrenamiento, validación y despliegue de modelos ML.
- Desarrollo y publicación de dashboards interactivos.
- Configuración de acceso, seguridad y monitoreo multinube.

5. Tiempo de Implementación

El proyecto se desarrollará en un periodo estimado de **3 a 4 meses**, sujeto al acceso a las credenciales, fuentes y disponibilidad del equipo cliente.

Parte 2: Implementación Práctica

1. Preparación y Exploración de Datos

Limpieza de datos

- Se eliminan **valores nulos** y **filas duplicadas** del dataset original.
- Se corrige la columna de precio (`Price;;;`) y se convierte a tipo numérico.
- Se eliminan **outliers** del precio mediante el método **IQR (rango intercuartílico)** para mejorar la robustez del modelo.

Exploración inicial

- Se identifican los principales atributos categóricos que inciden en el precio: `Brand`, `Category`, `Color`, `Size`, `Material`.
- Se generan **todas las combinaciones posibles** de segmentos con esas variables, permitiendo evaluar precios en escenarios reales y simulados.
- Se prepara una visualización de la relación entre precios reales y predichos, junto con el análisis de los errores (residuos), para detectar sesgos o dispersiones significativas.

2. Modelamiento en Estrella

La solución está estructurada como un **modelo dimensional** tipo estrella, orientado al análisis de precios por atributos de productos de ropa. La tabla **`tbl_fact_precio_ropa`** actúa como **tabla de hechos**, relacionada con varias **tablas de dimensiones** que definen las características descriptivas de cada combinación de producto.

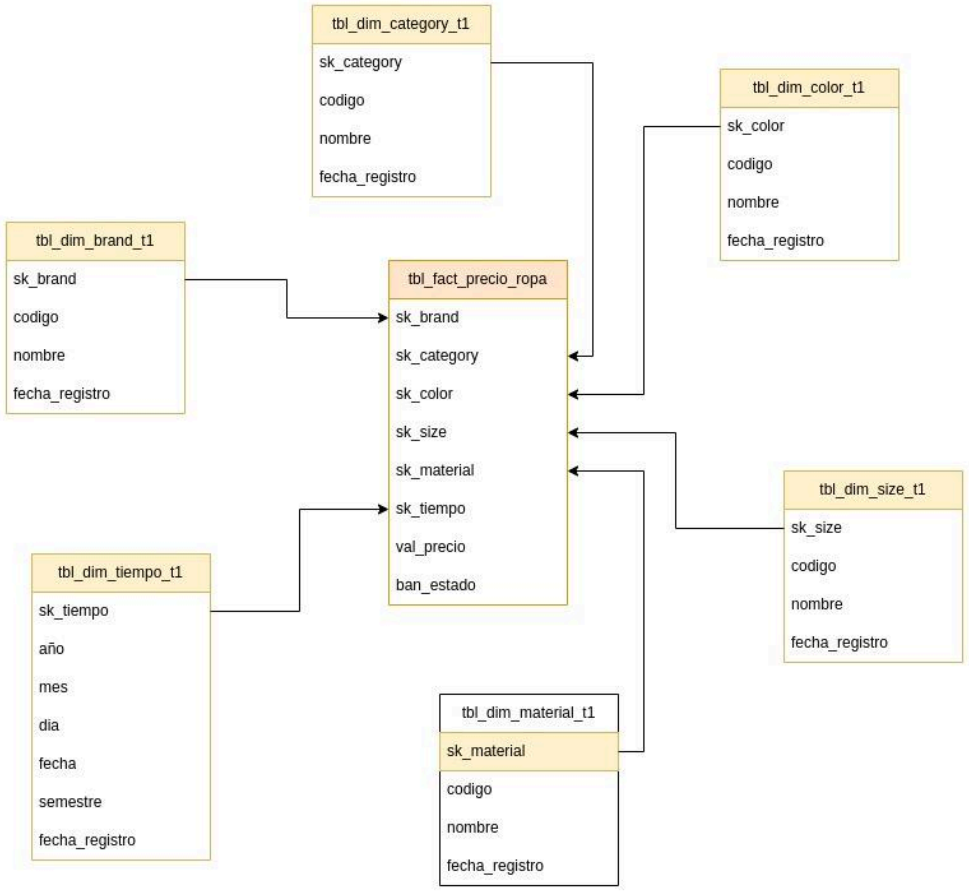


Tabla de hechos: `tbl_fact_precio_ropa`

Contiene los registros individuales de precios por combinación de dimensiones clave.

Campo	Descripción	Tipo de dato
`sk_brand`	Clave sustituta de la marca	Entero (FK)
`sk_category`	Clave sustituta de la categoría	Entero (FK)
`sk_color`	Clave sustituta del color	Entero (FK)
`sk_size`	Clave sustituta de la talla/tamaño	Entero (FK)
`sk_material`	Clave sustituta del material	Entero (FK)
`sk_tiempo`	Clave temporal para análisis por fecha	Entero (FK)
`val_precio`	Valor del precio asignado a la combinación	Decimal / Float
`ban_estado`	Bandera lógica del estado del registro (activo)	Boolean / Tinyint

Tablas de dimensiones

`tbl_dim_brand_t1` – Dimensión Marca

Campo	Descripción	Tipo de dato
`sk_brand`	Clave sustituta (PK)	Entero
`codigo`	Código de marca original	Texto / String
`nombre`	Nombre de la marca	Texto
`fecha_registro`	Fecha de inserción del registro	DateTime

`tbl_dim_category_t1` – Dimensión Categoría

Campo	Descripción	Tipo de dato
`sk_category`	Clave sustituta (PK)	Entero
`codigo`	Código de categoría	Texto
`nombre`	Nombre de la categoría	Texto
`fecha_registro`	Fecha de inserción del registro	DateTime

`tbl_dim_color_t1` – Dimensión Color

Campo	Descripción	Tipo de dato
`sk_color`	Clave sustituta (PK)	Entero
`codigo`	Código de color	Texto
`nombre`	Nombre del color	Texto
`fecha_registro`	Fecha de inserción del registro	DateTime

`tbl_dim_size_t1` – Dimensión Talla

Campo	Descripción	Tipo de dato
`sk_size`	Clave sustituta (PK)	Entero
`codigo`	Código de talla	Texto
`nombre`	Descripción o valor de talla	Texto

`fecha_registro`	Fecha de inserción del registro	DateTime

`tbl_dim_material_t1` – Dimensión Material

Campo	Descripción	Tipo de dato
`sk_material`	Clave sustituta (PK)	Entero
`codigo`	Código del material	Texto
`nombre`	Tipo de material (algodón, poliéster, etc.)	Texto
`fecha_registro`	Fecha de inserción del registro	DateTime

`tbl_dim_tiempo_t1` – Dimensión Tiempo

Campo	Descripción	Tipo de dato
`sk_tiempo`	Clave sustituta (PK)	Entero
`año`	Año	Entero
`mes`	Mes	Entero
`día`	Día	Entero
`fecha`	Fecha completa	Date
`semestre`	Semestre del año	Entero (1 o 2)
`fecha_registro`	Fecha de inserción del registro	DateTime

3. Modelamiento Predictivo

Entrenamiento del modelo

- Se ofrece al usuario elegir entre tres algoritmos:
    - `LinearRegression`
    - `RandomForestRegressor`
    - `XGBoostRegressor`
- El pipeline incluye:
    - **Preprocesamiento**: codificación One-Hot para variables categóricas y escalado estándar para las numéricas.

    - **Entrenamiento supervisado**: los datos limpios se utilizan para entrenar el modelo elegido con hiperparámetros definidos.

Justificación del modelo

- **Random Forest** y **XGBoost** ofrecen buena capacidad de generalización, tolerancia a valores atípicos y robustez ante datos heterogéneos.
- En pruebas de validación interna, se calcula el **MSE (Mean Squared Error)** entre el precio real y el predicho para evaluar el rendimiento.
- Los modelos son persistidos usando `joblib` para permitir despliegue futuro como microservicio o dentro de un pipeline automatizado.

4. Almacenamiento y Resultados

Formatos de salida

- Se generan archivos en **Parquet** (optimizado para consultas analíticas) y opcionalmente en **JSON** (orientado a integraciones API o front-end) de ser necesario de acuerdo a selección.
- Se guarda el resultado general y también un subconjunto filtrado (solo combinaciones con precio original disponible) para validación.
    - predicciones_por_segmento.json
    - predicciones_por_segmento.parquet
    - predicciones_por_segmento_filtrado.json
    - predicciones_por_segmento_filtrado.parquet

Documentación Técnica - Predicción de Precios por Categorías

1. Objetivo

El objetivo principal de este proyecto es desarrollar un modelo predictivo capaz de estimar el precio de prendas de ropa a partir de atributos categóricos como marca, categoría, color, talla y material. Esto permite generar precios estimados para combinaciones nuevas o no observadas previamente en el histórico.

2. Alcance

- Lectura y limpieza de un dataset con atributos de ropa.
- Preprocesamiento de variables categóricas y numéricas.
- Entrenamiento y validación cruzada de tres modelos diferentes.
- Generación de predicciones para todas las combinaciones posibles de atributos.
- Evaluación de desempeño con MSE.
- Exportación de resultados a archivos Parquet y JSON.
- Visualización gráfica de predicciones y errores.

3. Flujo Funcional del Script Python

Flujo de ejecución:

1. **Inicio del script**
2. **Ingreso de ruta del archivo CSV** por el usuario.
3. **Lectura del archivo y transformación de columnas**:
    - Renombrar y convertir `Price;;;` a valor numérico.
4. **Limpieza de datos**:
    - Eliminación de nulos y duplicados.
    - Filtrado de outliers con método IQR.
5. **Separación de variables X (atributos) e y (precio)**
6. **Selección de modelo por el usuario**:
    - 0: LinearRegression
    - 1: RandomForest
    - 2: XGBoost
7. **Construcción del pipeline** con:
    - OneHotEncoder para variables categóricas.

    - StandardScaler para numéricas.
    - Modelo seleccionado.
8. **Entrenamiento del modelo**
9. **Cálculo de validación cruzada (5-fold)** con MSE.
10. **Generación de todas las combinaciones posibles de atributos**
11. **Predicción de precios para cada combinación**
12. **Cálculo del MSE solo con filas que tienen precio original conocido**
13. **Exportación de resultados en archivos `.parquet` y `.json`** (opcional)
14. **Visualización gráfica**:
    - Precio original vs predicho.
    - Distribución de errores.
15. **Fin del script**

4. Descripción de los Modelos

0 - LinearRegression

Descripción: modelo base que asume una relación lineal entre los atributos codificados y el precio.

Ventajas:

- Rápido de entrenar.
- Fácil de interpretar.

Desventajas:

- Asume independencia lineal entre variables.
- Poco robusto ante datos ruidosos o relaciones no lineales.

MSE obtenido: 2810.90

1 - RandomForestRegressor

Descripción: ensamble de árboles de decisión entrenados sobre subconjuntos aleatorios del dataset.

Hiperparámetros usados:

- `n_estimators=300`: número de árboles.
- `max_depth=15`: profundidad máxima de los árboles.
- `min_samples_split=5`: mínimo de muestras para dividir un nodo.
- `min_samples_leaf=2`: mínimo de muestras por hoja.

Buenas prácticas:

- Validar el overfitting si el número de árboles o profundidad es muy alto.
- Usar `n_jobs=-1` para acelerar entrenamiento.

MSE obtenido: 1335.81

2 - XGBoostRegressor

Descripción: técnica de boosting que entrena modelos secuenciales minimizando el error residual.

Hiperparámetros usados:

- `n_estimators=300`: árboles secuenciales.
- `max_depth=6`: profundidad de cada árbol.
- `learning_rate=0.05`: tasa de aprendizaje.
- `subsample=0.8`: fracción de muestras usadas por árbol.
- `colsample_bytree=0.8`: fracción de columnas por árbol.

Buenas prácticas:

- Evitar overfitting reduciendo `max_depth` y ajustando `learning_rate`.
- Monitorear el error en datos de validación.

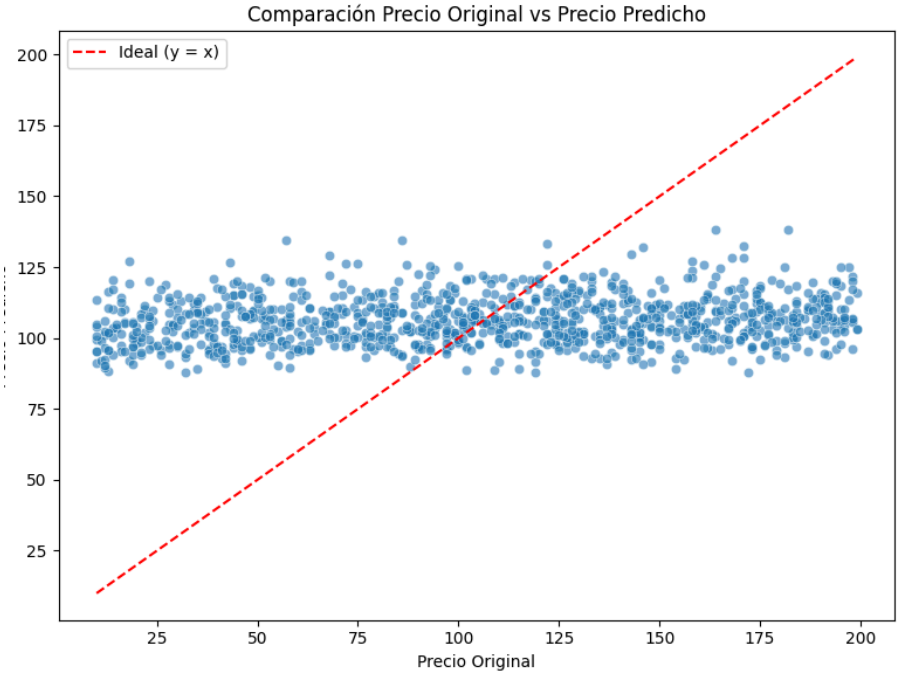
MSE obtenido: 839.06

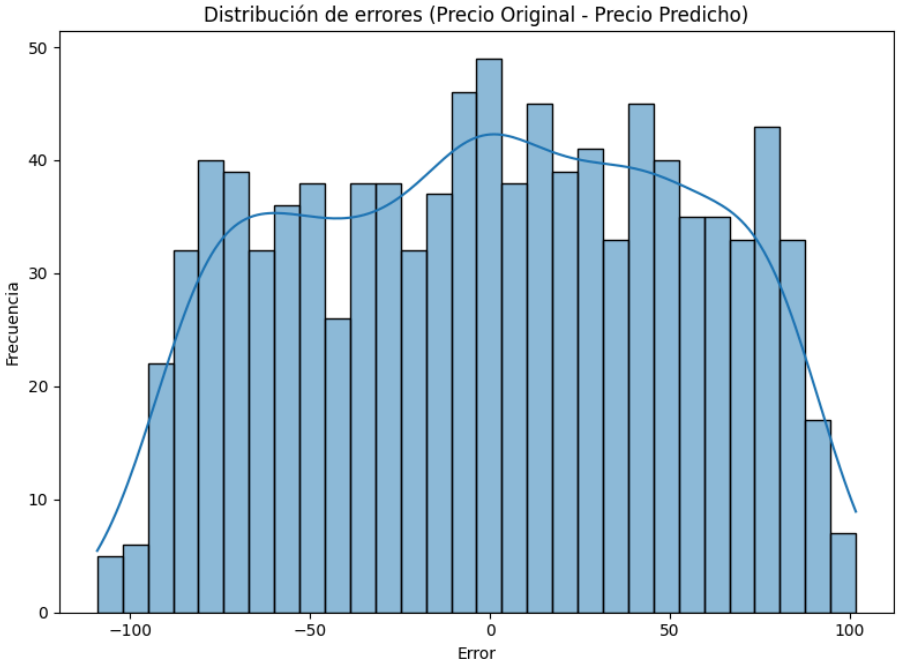
5. Comparación y Conclusión

Modelo	MSE	Observaciones
LinearRegression	2810.90	Alto error, no captura relaciones complejas.
RandomForest	1335.81	Mejora notable, robusto y no lineal.
XGBoost	**839.06**	Mejor desempeño, ideal para relaciones complejas.

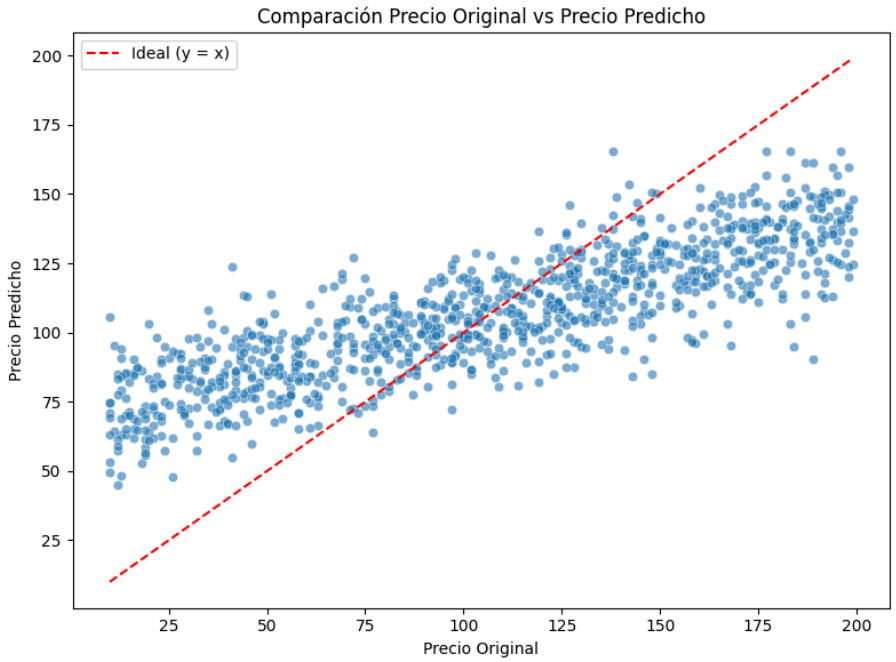
Conclusión: Para el problema de predicción de precios basado en múltiples atributos categóricos, el modelo **XGBoost** es el más recomendado, gracias a su capacidad para modelar relaciones no lineales y su bajo error cuadrático medio (MSE).

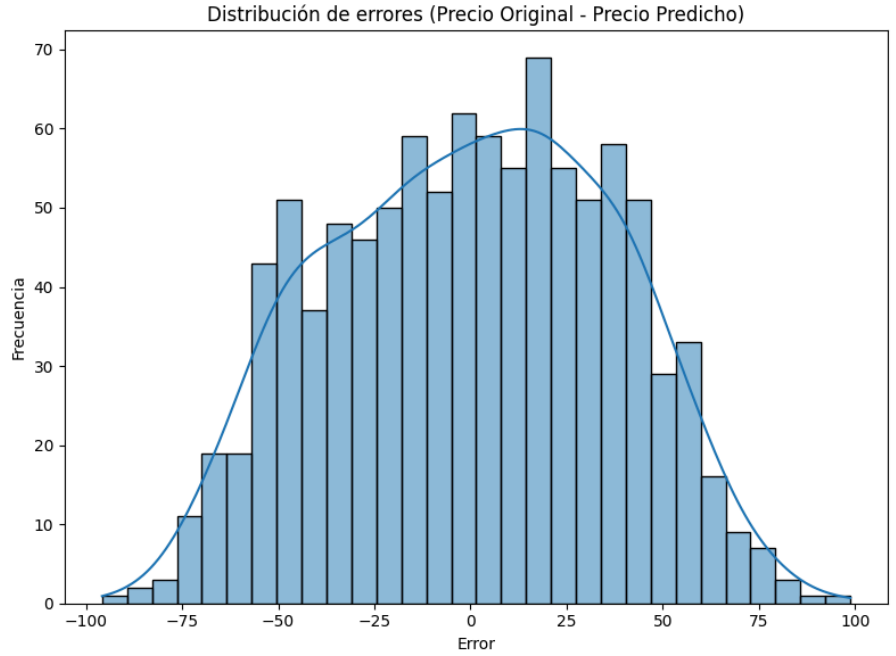
LinearRegression



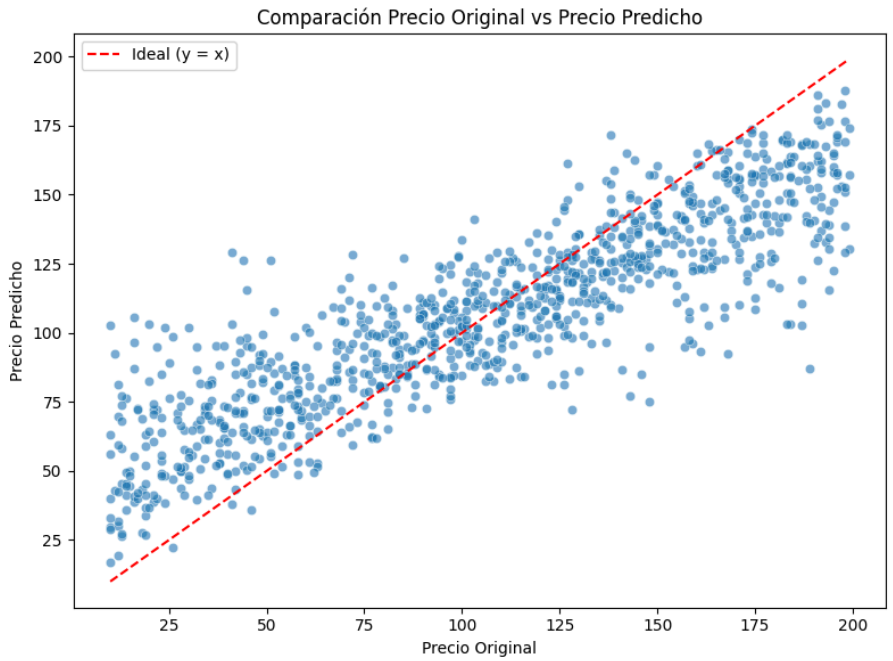


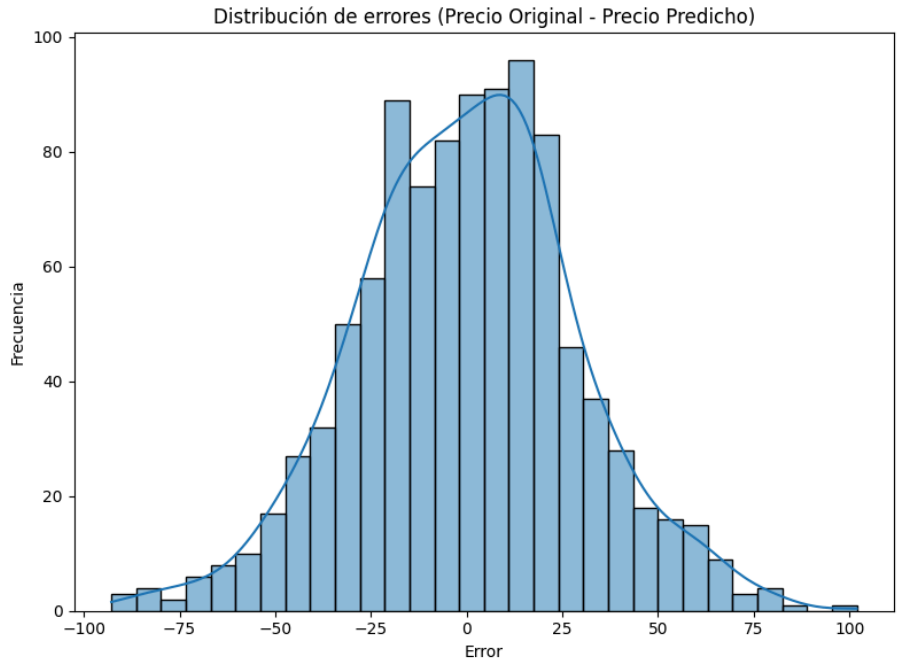
RandomForest


Comparación Precio Original vs Precio Predicho
Ideal (y = x)
Precio Predicho
Precio Original
200
175
150
125
100
75
50
25
25
50
75
100
125
150
175
200


Distribución de errores (Precio Original - Precio Predicho)
Frecuencia
Error
70
60
50
40
30
20
10
0
−100
−75
−50
−25
0
25
50
75
100

XGBoost





Evidencia de ejecución

antonio@antonio-ZenBook:~/Documentos/Prueba$ python3 predecir_precios_grafica.py
Ingresa la ruta completa del archivo CSV de entrada:
/home/antonio/Documentos/Prueba/clothes_price_prediction_dat.csv

¿Qué modelo deseas usar?
0 - LinearRegression
1 - RandomForest
2 - XGBoost

Escribe 0 o 1 o 2: 2
Modelo XGBoost entrenado y guardado en
/home/antonio/Documentos/Prueba/modelo_precio_XGBoost.pkl.

20 primeras filas de las predicciones por segmento:

	Brand	Category	Color	Size	Material	Precio_Original	Precio_Predicho
0	New Balance	Dress	White	XS	Nylon	182.0	169.653152
1	New Balance	Dress	White	XS	Silk	NaN	125.704857
2	New Balance	Dress	White	XS	Wool	NaN	106.744492
3	New Balance	Dress	White	XS	Cotton	NaN	125.907082
4	New Balance	Dress	White	XS	Polyester	NaN	107.162193
5	New Balance	Dress	White	XS	Denim	NaN	138.374985
6	New Balance	Dress	White	M	Nylon	NaN	117.281494
7	New Balance	Dress	White	M	Silk	NaN	105.662979
8	New Balance	Dress	White	M	Wool	NaN	105.079094
9	New Balance	Dress	White	M	Cotton	NaN	105.550987
10	New Balance	Dress	White	M	Polyester	NaN	118.516724
11	New Balance	Dress	White	M	Denim	NaN	146.350311
12	New Balance	Dress	White	XL	Nylon	NaN	119.757828
13	New Balance	Dress	White	XL	Silk	NaN	112.694351
14	New Balance	Dress	White	XL	Wool	NaN	98.199661
15	New Balance	Dress	White	XL	Cotton	NaN	105.951141
16	New Balance	Dress	White	XL	Polyester	NaN	84.522636
17	New Balance	Dress	White	XL	Denim	NaN	111.347252
18	New Balance	Dress	White	XXL	Nylon	164.0	146.766037
19	New Balance	Dress	White	XXL	Silk	NaN	112.400726

MSE en filas con precio original conocido: 839.06

¿Quieres ver sólo filas con precio original conocido? (s/n): s

20 primeras filas con precio original conocido:

	Brand	Category	Color	Size	Material	Precio_Original	Precio_Predicho
0	New Balance	Dress	White	XS	Nylon	182.0	169.653152
18	New Balance	Dress	White	XXL	Nylon	164.0	146.766037
22	New Balance	Dress	White	XXL	Polyester	57.0	88.499992
23	New Balance	Dress	White	XXL	Polyester	86.0	88.499992
25	New Balance	Dress	White	S	Nylon	146.0	140.544296
33	New Balance	Dress	White	L	Wool	122.0	117.327408
47	New Balance	Dress	Black	M	Polyester	178.0	146.830933
54	New Balance	Dress	Black	XL	Denim	48.0	65.189026
63	New Balance	Dress	Black	S	Wool	107.0	126.904907
88	New Balance	Dress	Red	XL	Cotton	190.0	132.466339
95	New Balance	Dress	Red	XXL	Polyester	158.0	137.653961
106	New Balance	Dress	Red	L	Cotton	44.0	126.375412
112	New Balance	Dress	Green	XS	Cotton	109.0	129.349777
116	New Balance	Dress	Green	M	Silk	99.0	127.906944
122	New Balance	Dress	Green	XL	Silk	141.0	125.341019

131	New Balance	Dress	Green	XXL	Polyester	94.0	103.819504
143	New Balance	Dress	Green	L	Polyester	85.0	126.939697
146	New Balance	Dress	Yellow	XS	Silk	150.0	142.694016
157	New Balance	Dress	Yellow	XL	Nylon	39.0	65.651619
171	New Balance	Dress	Yellow	S	Wool	198.0	151.143112

Guardado archivo parquet filtrado:
/home/antonio/Documentos/Prueba/predicciones_por_segmento_filtrado.parquet

¿Deseas guardar también el archivo JSON sólo con filas filtradas? (s/n): s
Guardado en JSON filtrado:
/home/antonio/Documentos/Prueba/predicciones_por_segmento_filtrado.json
Guardado archivo parquet:
/home/antonio/Documentos/Prueba/predicciones_por_segmento.parquet

¿Deseas guardar también en JSON el archivo completo? (s/n): s
Guardado en JSON: /home/antonio/Documentos/Prueba/predicciones_por_segmento.json

¿Deseas generar gráficas de análisis? (s/n): s