



# MASTER EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

*Expositores:*  
**ING. LEONEL LINARES**  
**LIC. LESLY SALMERON**

**JUNIO 2025**



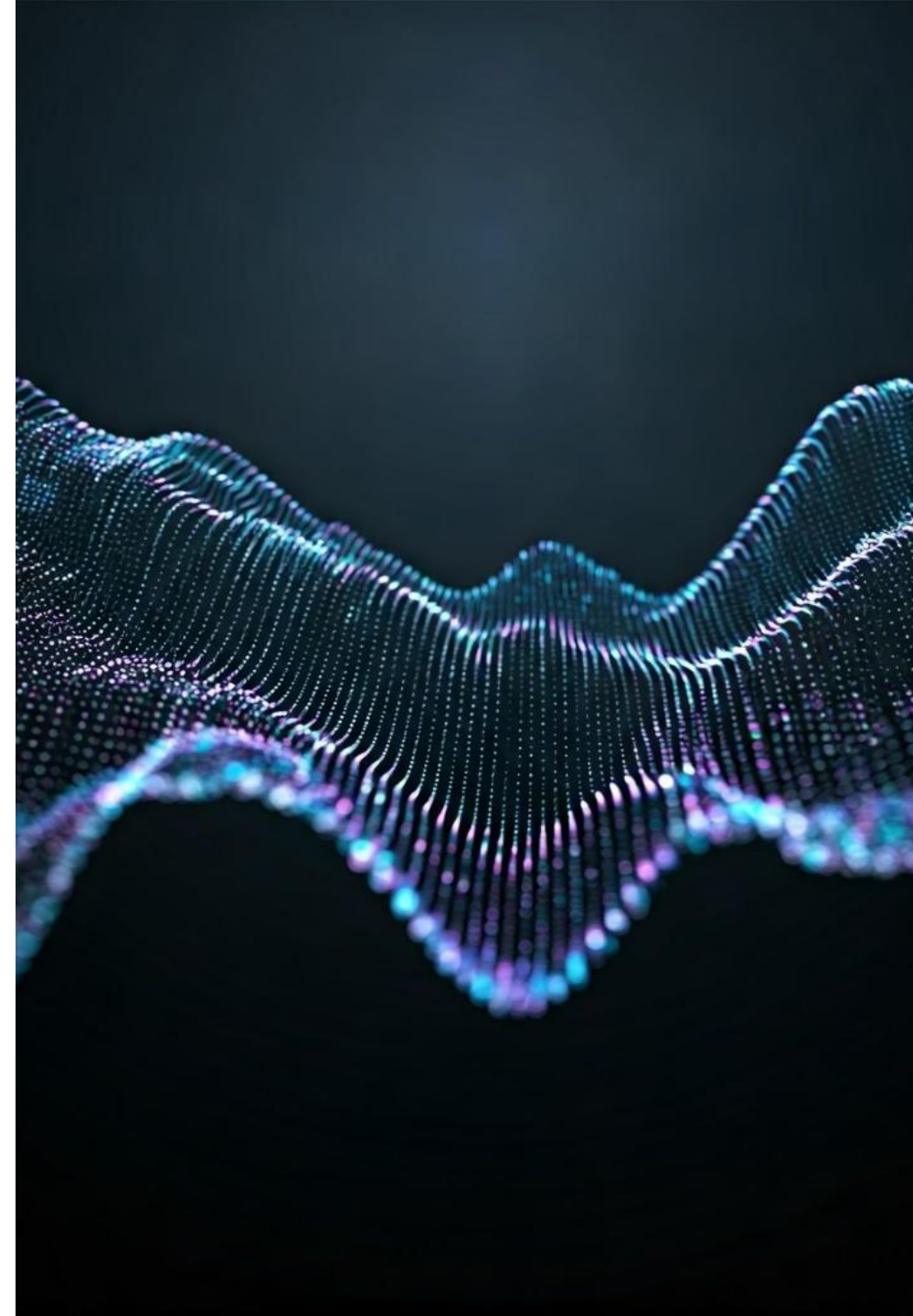
# RETO # 1

# Análisis y Modelado de Datos

## Datos para Predicción de Precios

**RETO:**  
**HOUSE PRICES - ADVANCED REGRESSION TECHNIQUES**

Presentación de nuestro proceso de análisis de datos y  
modelado para predecir precios



# DESCRIPCIÓN DEL RETO

- Predecir el Precio Final de viviendas mediante *Modelos Predictivos*.
- Utilizando un conjunto de datos que describen diversos aspectos de las propiedades, como características físicas, ubicación y entre otros.





# Conjunto de Datos



# Archivos

## Dos archivos CSV:

train.csv **1,460** Registros

test.csv: **1,459** Registros



## Variables

## 81 variables en total

**43** categóricas y  
**38** numéricas



## Variable Objetivo

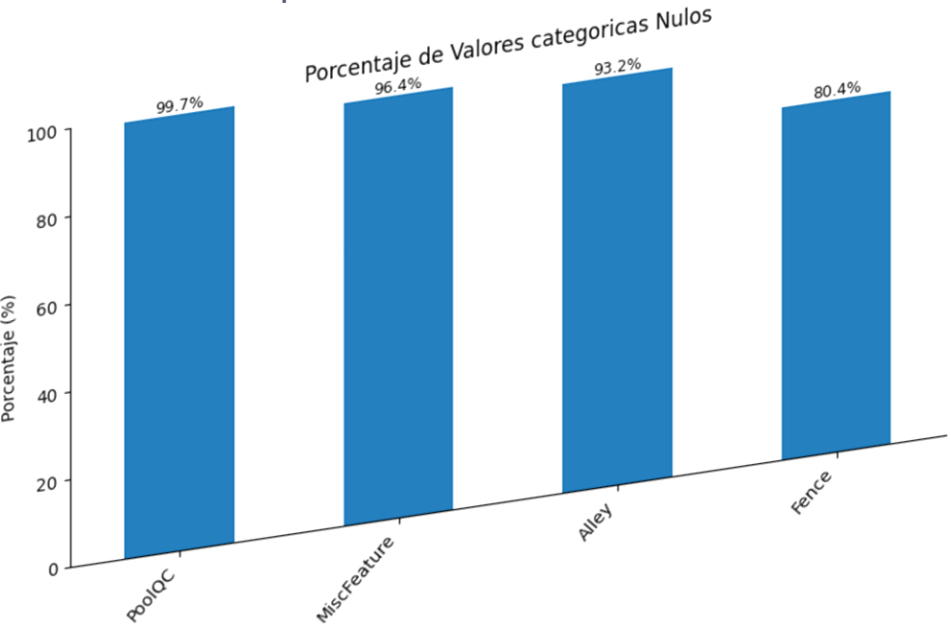
## Precio de Venta (SalePrice)

# Análisis de Variables Categóricas

## Variables con Valores Nulos

Identificamos variables categóricas con más del 80% de valores nulos.

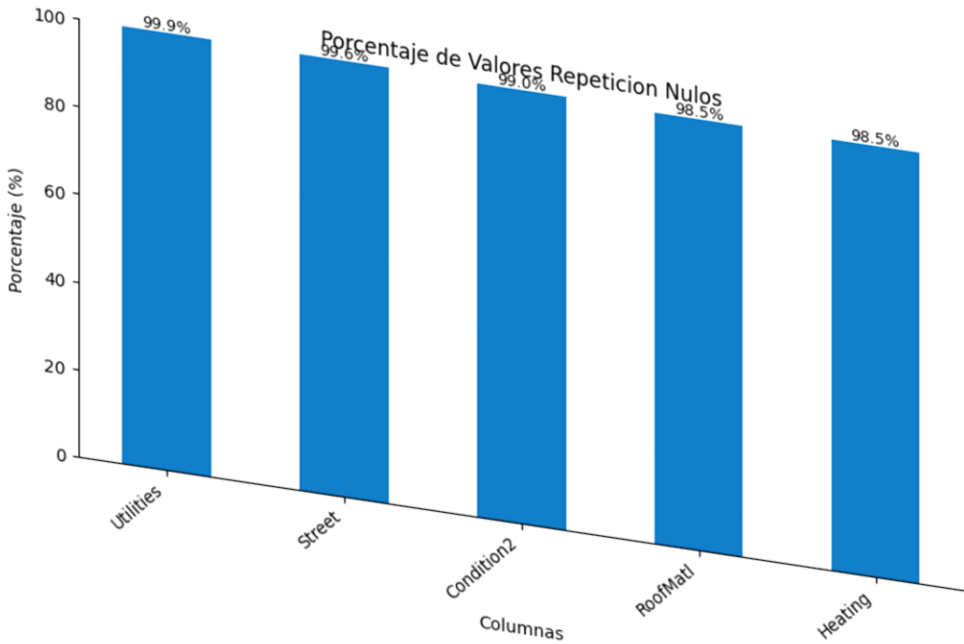
Esto nos permitió determinar qué variables requerían tratamiento especial.



## Variables con Alta Repetición

Detectamos variables categóricas con más del 95% de repetición de valores.

Estas variables aportan poca información al modelo.





# Análisis de Variables Numéricas Numéricas

## Valores Nulos

Identificamos variables numéricas con **más del 5% de valores nulos**.

## Valores Cero

Detectamos variables con **más del 95% de valores cero**.

## Correlación

Analizamos variables con **correlación superior al 80%** para aplicar PCA.

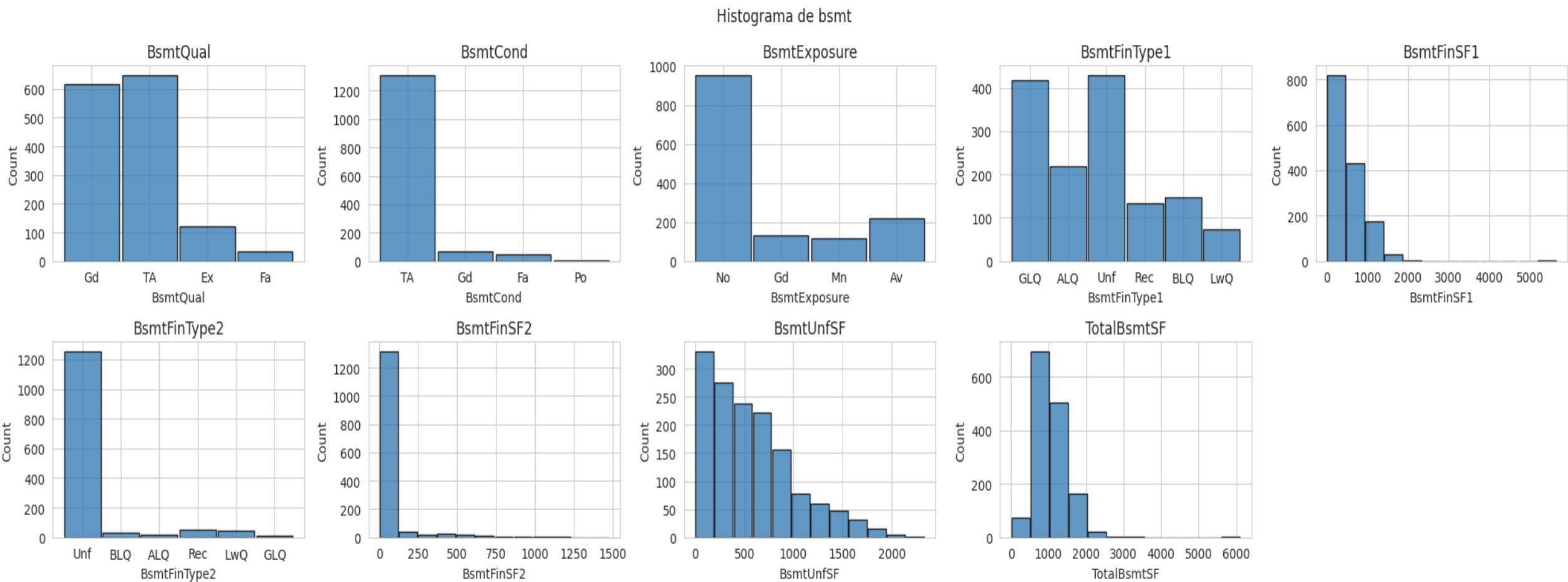
GarageCars	GarageArea	0.890
YearBuilt	GarageYrBlt	0.835
GrLivArea	TotRmsAbvGrd	0.808
TotalBsmtSF	1stFlrSF	0.802



# Análisis de Variables

*Representación gráfica de las variables numéricas y categóricas para identificar patrones*

## Distribución de las variables

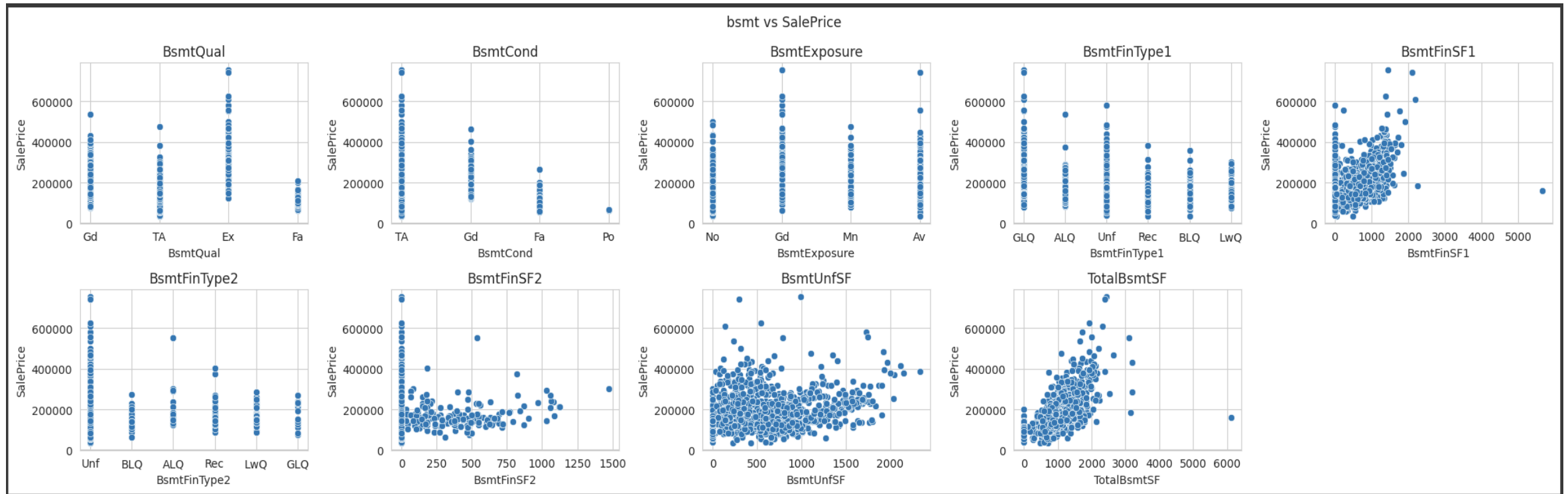




# Análisis de Variables

*Representación gráfica de las variables numéricas y categóricas para identificar patrones*

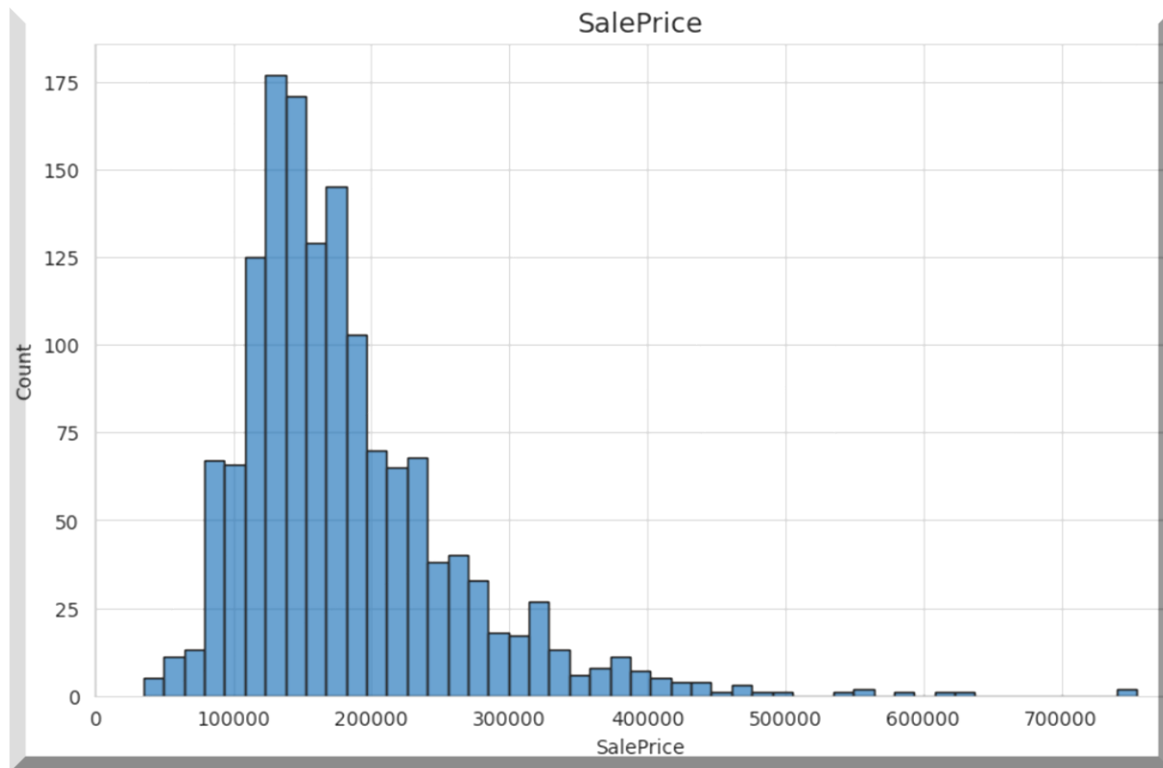
## Relación con la Variable Objetivo



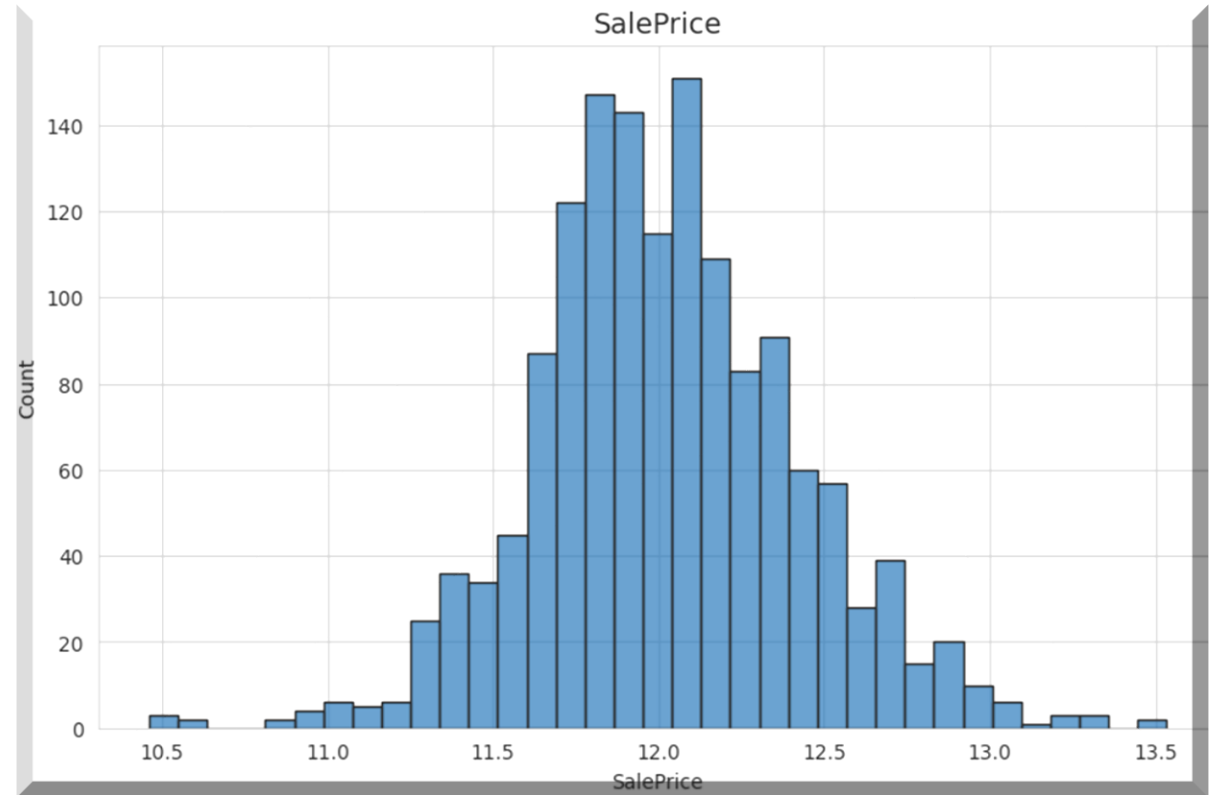
# Análisis de Variables

*Representación Gráfica de la Distribución de la Variable Objetivo*

## Escenario Inicial



## Aplicando Logaritmo



# Transformación de Variables



## Variables Numéricas con NULL

Se rellenaron con cero.



## Variables Categóricas con NULL

Aplicamos Ordinal encoding, Target encoding y encoding y One hot encoding.



## Variables Nuevas

Creamos variables con mayor correlación.



## Técnicas de Escalado

Utilizamos RobustScaler y StandardScaler.

# Entrenar

## Modelos Evaluados

- ☐ LinearRegression
- ☐ CatBoostRegressor
- ☐ GradientBoostingRegressor
- ☐ XGBoost
- ☐ LightGBM
- ☐ RandomForest



**Se implementó  
Grid Search**



**Se definió un conjunto de  
hiperparámetros asignando  
múltiples valores a cada variable,  
con el objetivo de explorar distinta  
combinaciones durante el proceso  
de ajuste de modelo.**



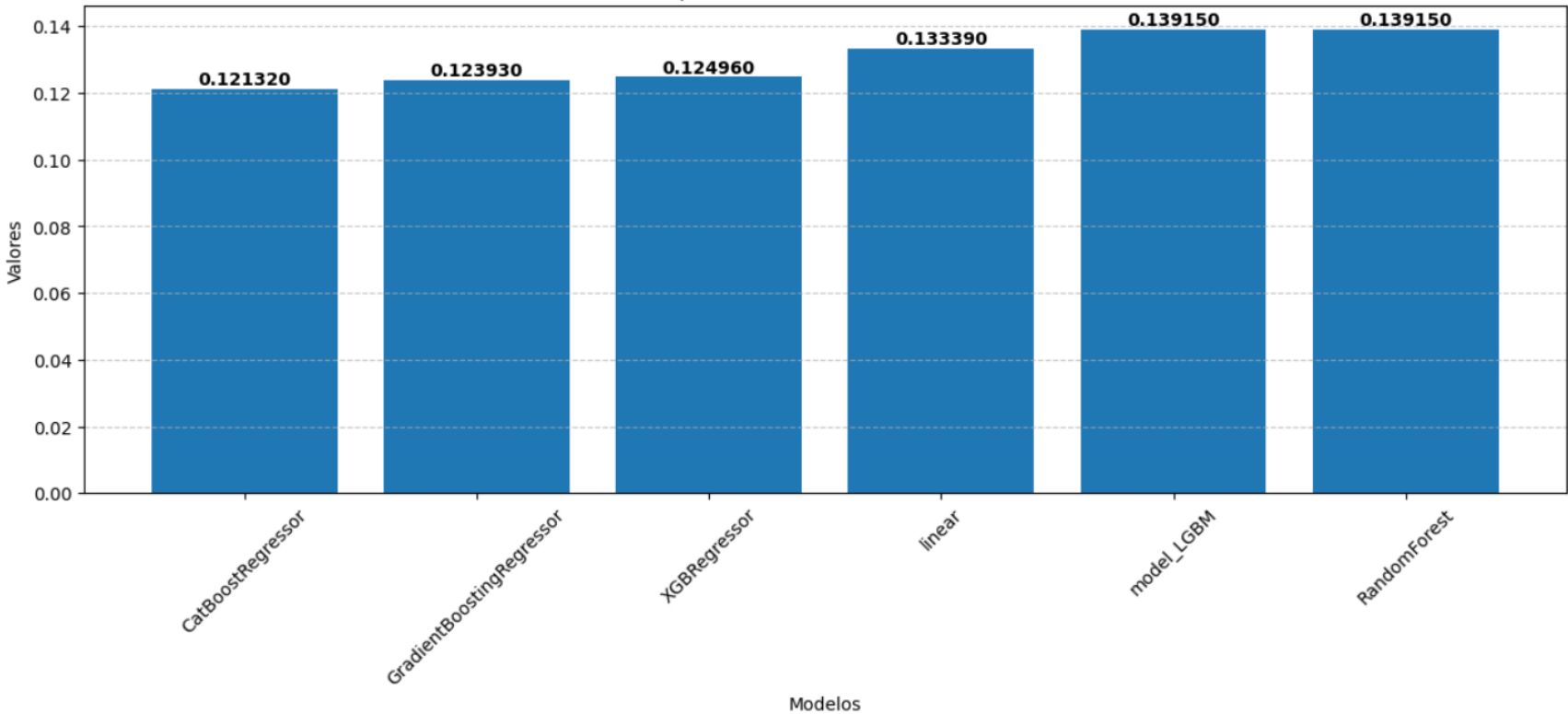
**La validación del modelo se realizó  
utilizando el Error Cuadrático  
Medio(RMSE).**





# Modelos Aplicados

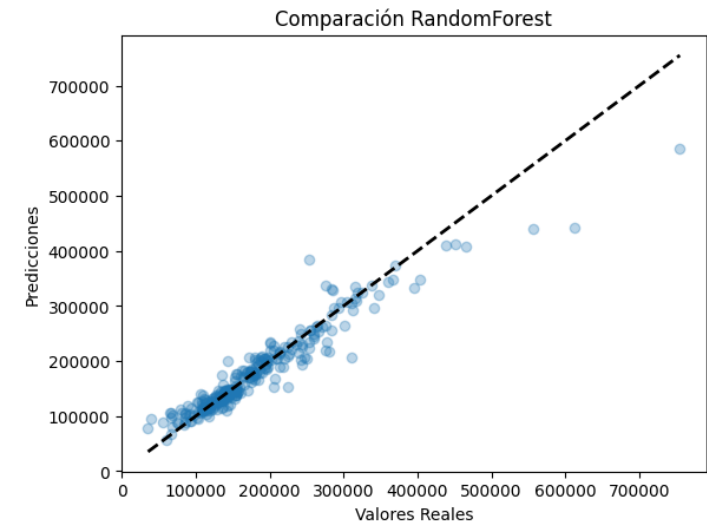
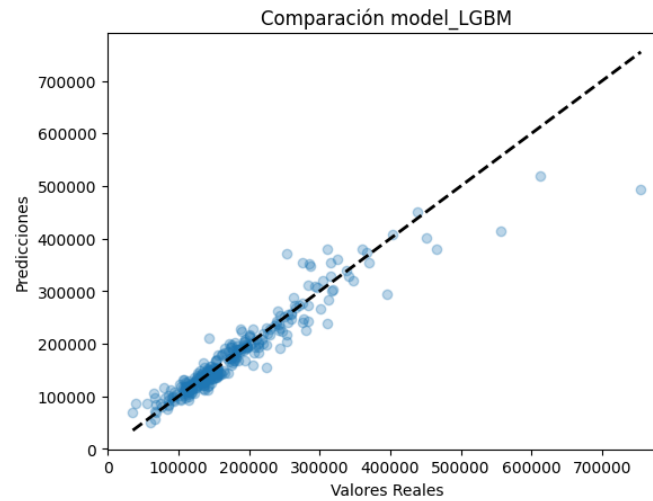
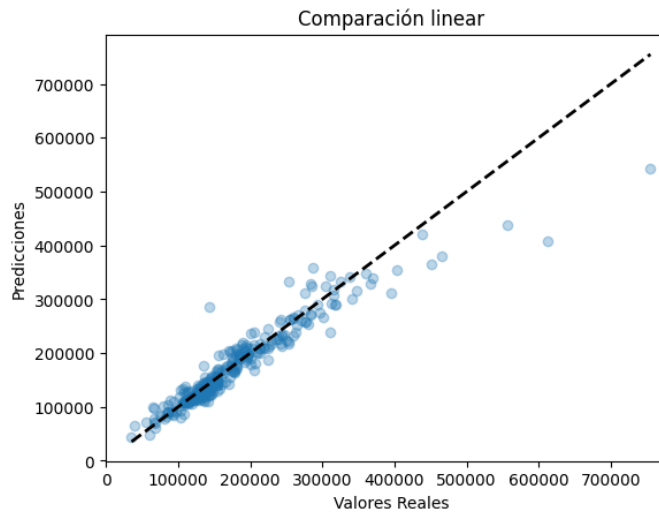
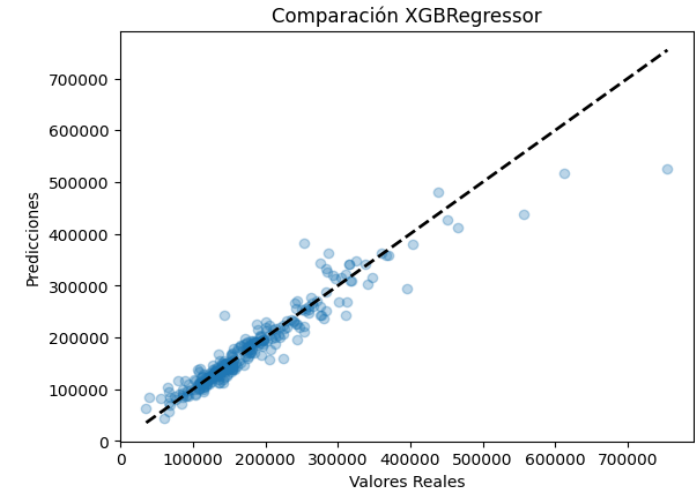
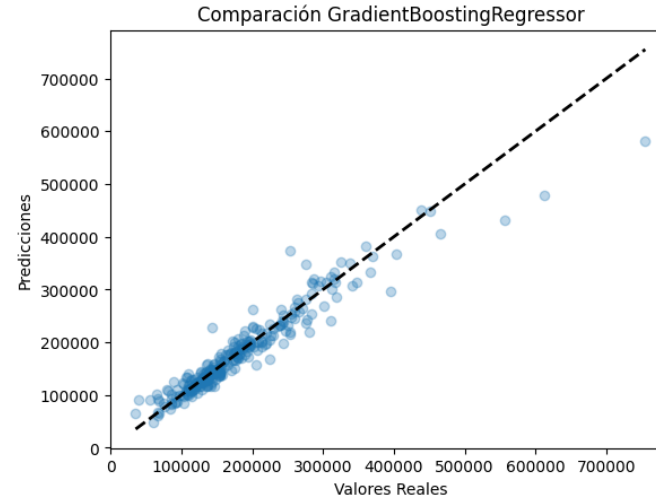
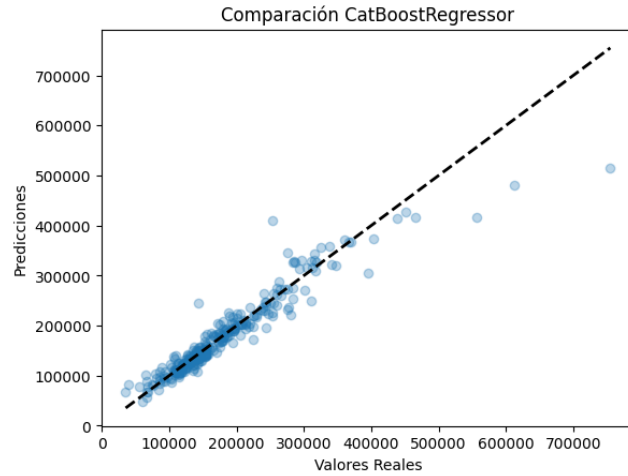
Comparación de Modelos Ordenados



Modelo	Tiempo (segundos)
LinearRegression	0.01380
CatBoostRegressor	1.51203
GradientBoostingRegressor	1.07605
XGBoost	0.95564
LightGBM	0.32938
RandomForest	3.38828



# Diagrama de dispersión



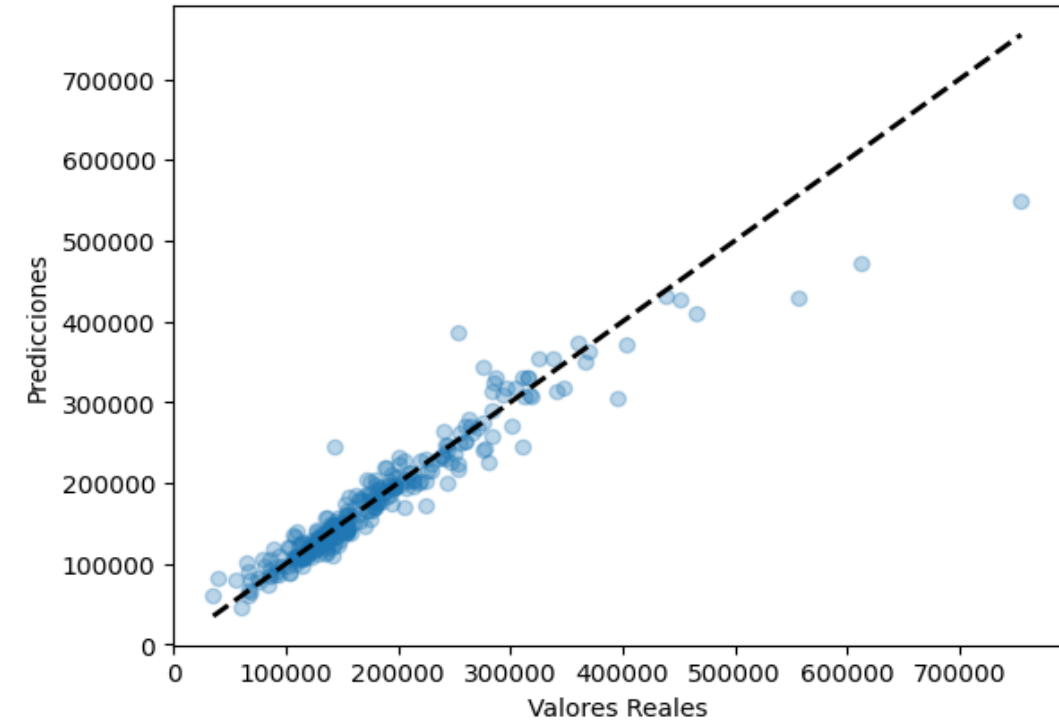
# Técnica de aprendizaje en conjunto (*ensemble learning*)

```
cv_fold = KFold(n_splits= 15, shuffle=True, random_state=12)

# CREATE STACKING REGRESSOR
model = StackingRegressor(
    estimators=[
        ('GradientBoostingRegressor', model_GBR),
        ('catboost', model_cat),
        ('linear', linear_model)
    ],
    final_estimator = RidgeCV(),
    cv=cv_fold
)
```

Modelos Combinados

- ✓ LinearRegression
- ✓ CatBoostRegressor
- ✓ GradientBoostingRegressor



4.89 segundos

**StackingRegressor** es una técnica de **aprendizaje en conjunto** que combina múltiples modelos de regresión para mejorar la precisión de las predicciones. En lugar de depender de un solo modelo, **StackingRegressor** apila varios modelos base y usa un **meta-modelo** para aprender de sus predicciones y generar una estimación final más robusta.



# Resultados y Conclusiones

0.11814

**Puntaje Final**

Métrica de evaluación del modelo

179

**Posición**

Entre 4,648 participantes

179

CesteGrupo42025



0.11814

1

1m

6  
**Modelos**

Algoritmos de regresión evaluados  
evaluados



# RETO # 2



# MASTER EN CIENCIA DE DATOS E INTELIGENCIA ARTIFICIAL

*Expositores:*

**ING. LEONEL LINARES**

**LIC. LESLY SALMERON**

**JUNIO 2025**



# Análisis y Modelado de Datos para Predicción de Tweets

**RETO:**  
**NATURAL LANGUAGE PROCESSING WITH DISASTER TWEETS**

Presentación de nuestro proceso de análisis de datos y modelado para predecir Tweets sobre desastres reales y no reales



# DESCRIPCIÓN DEL RETO

## PROCESAMIENTO DEL LENGUAJE NATURAL CON TWEETS DE DESASTRE

Construir un modelo de aprendizaje automático que prediga qué Tweets tratan sobre desastres reales y cuáles no.  
no.

Conjunto de datos de más 10 000 Tweets  
Tweets Clasificados Manualmente.





# Conjunto de Datos



## Archivos

## Dos archivos CSV:

train.csv **7,613** registros

test.csv 3,263 registros



## Variables

5 variables en total

### 3 categóricas

2 numéricas



## Variable Objetivo

Indicador de desastre(target)

# Análisis de Variables Categóricas

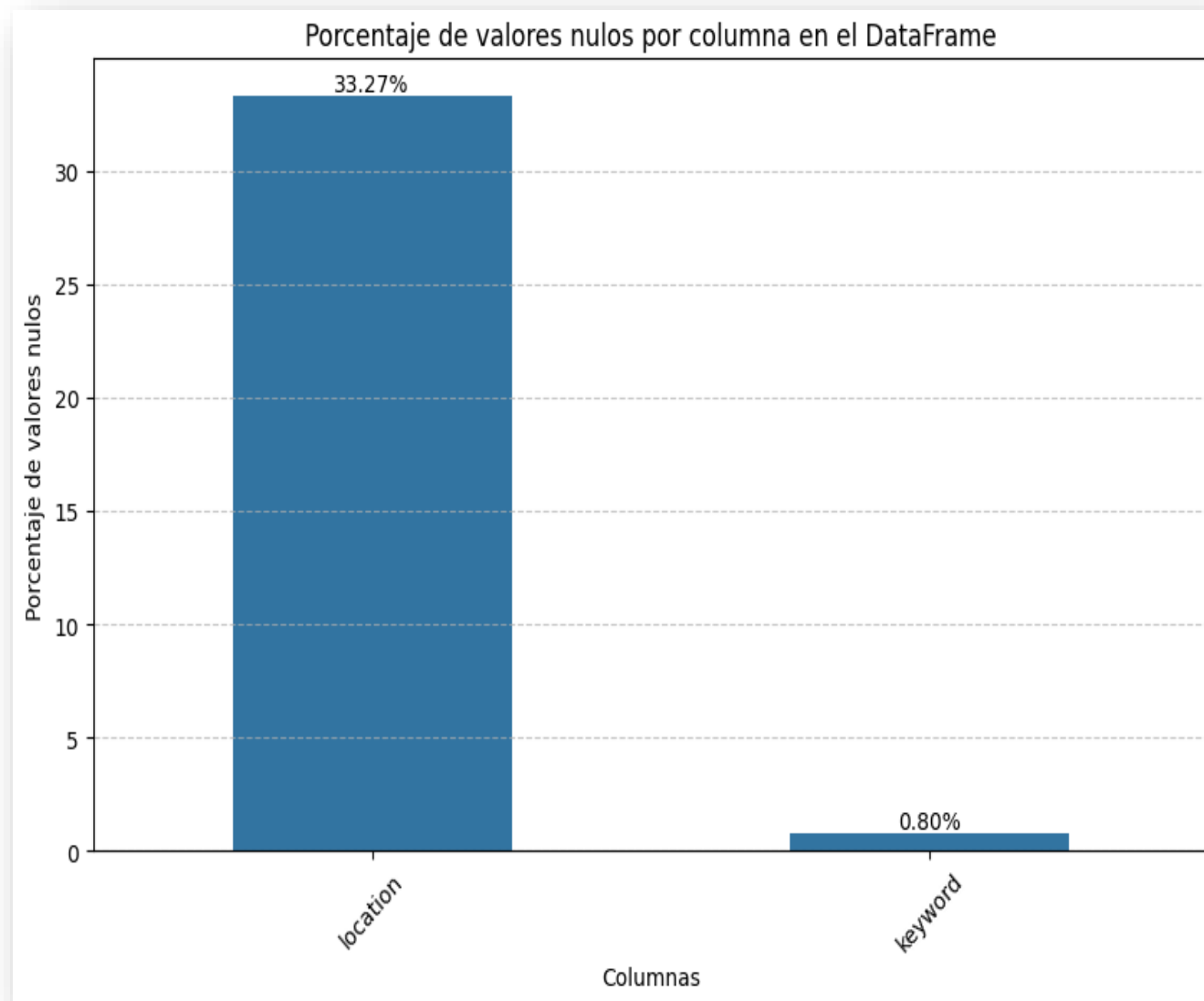
## Variables con Valores Nulos



Identificamos variables categóricas de valores nulos.

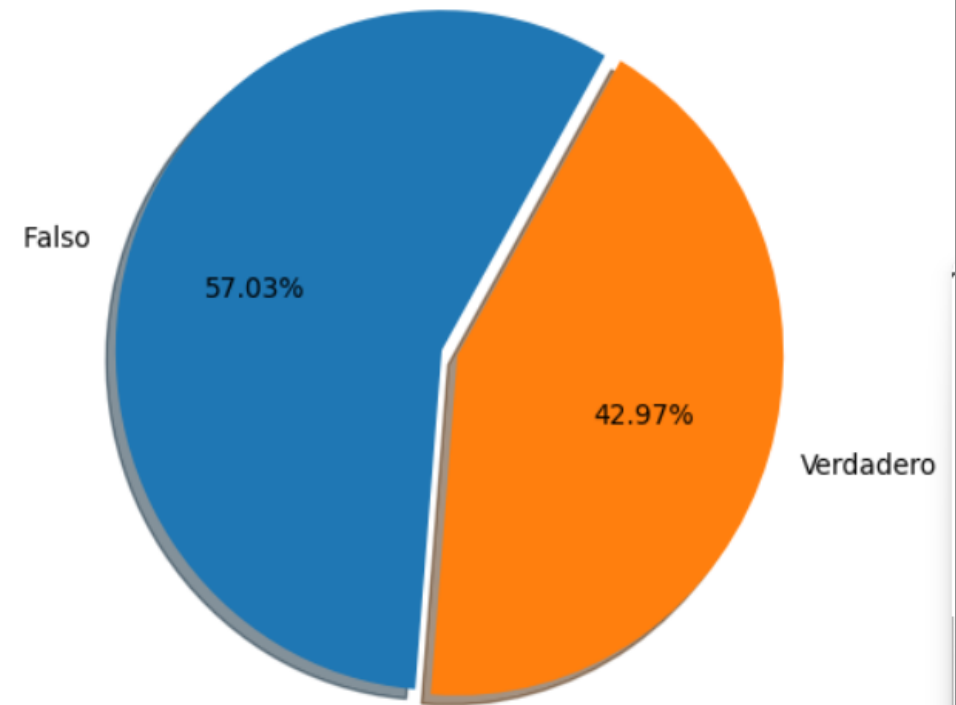
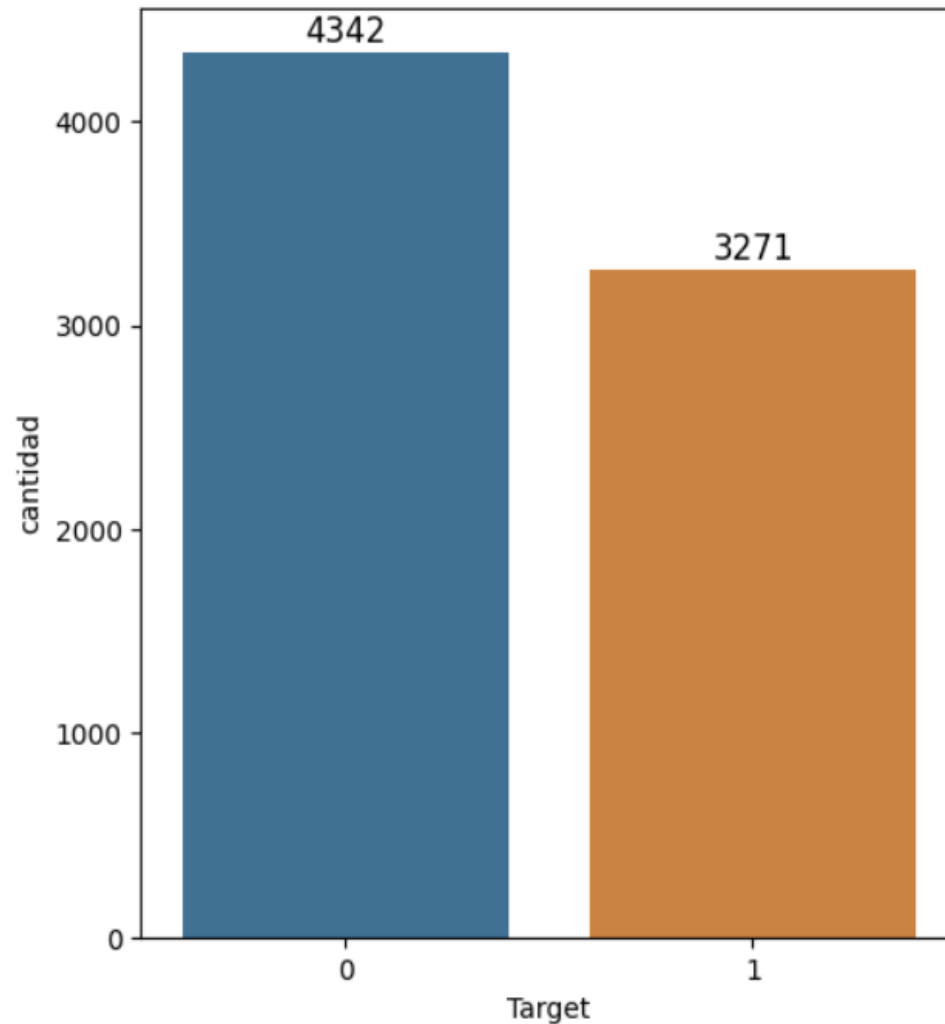


Esto nos permitió determinar qué variables requerían tratamiento especial.



# Análisis de Variables Numéricas

## Distribución de los Tweets



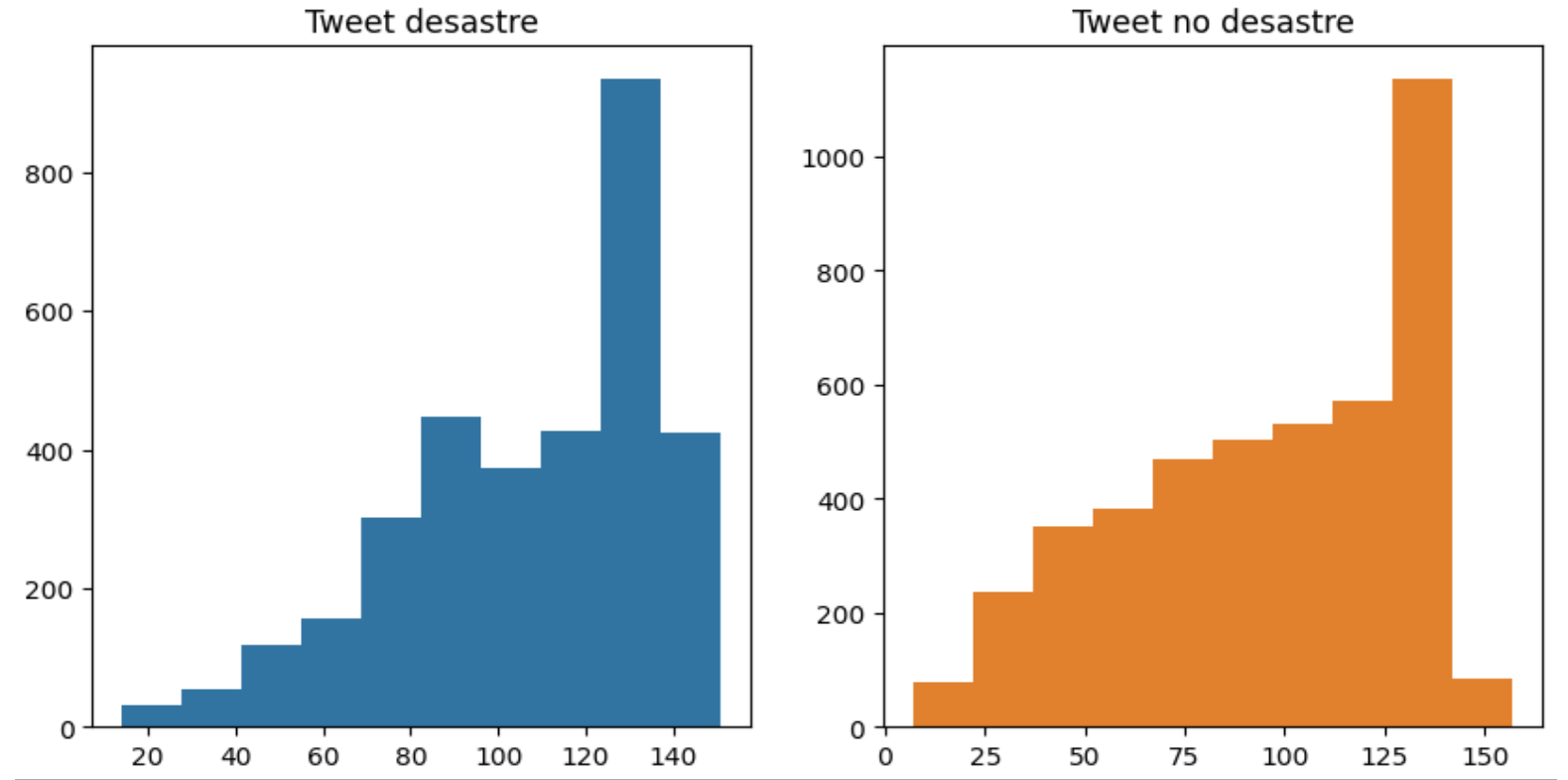


# Análisis de Variables

## *Histograma para identificar Patrones*

### Distribución de la longitud de los Tweets

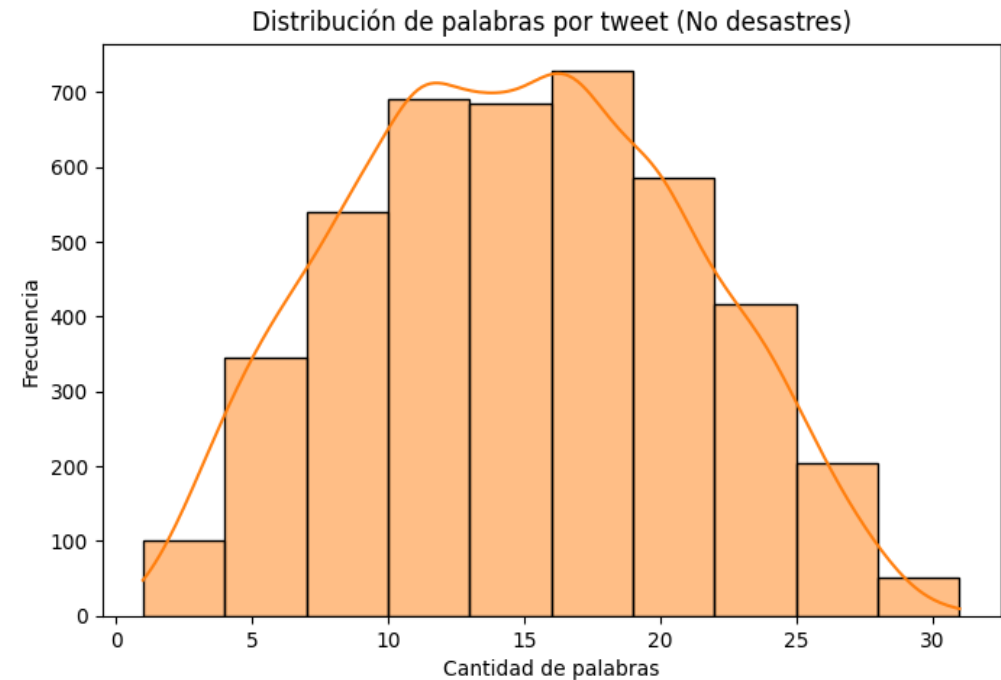
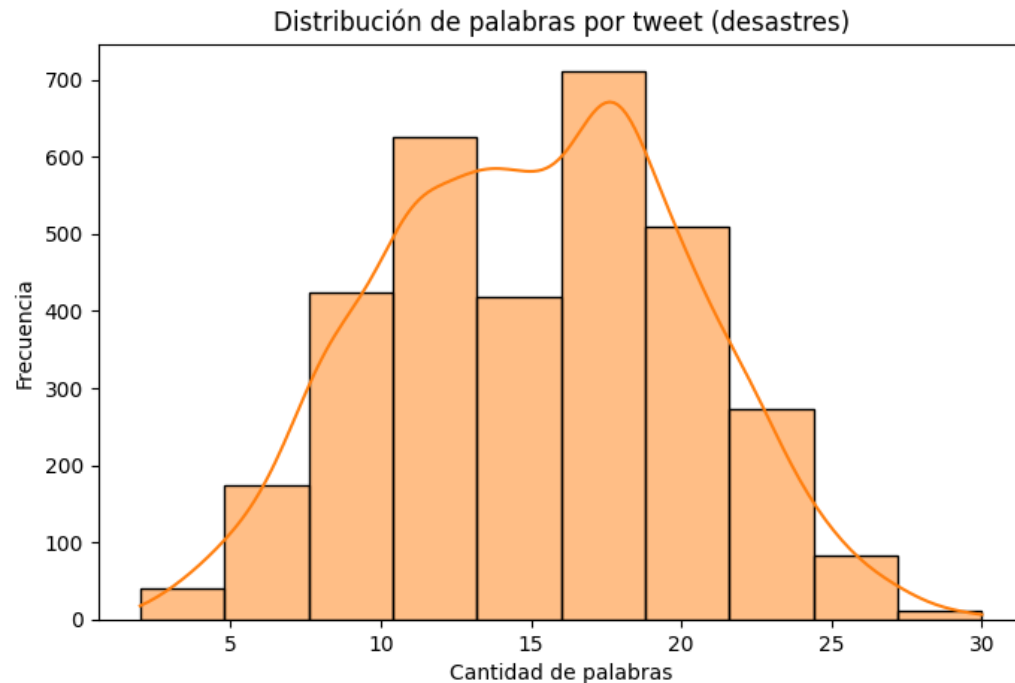
"La **longitud del tweet** es un detector de emergencias: Esto permite crear un **filtro de priorización** para salvar vidas."



# Análisis de Variables

## *Distribución grafica de palabras por cada cada Tweets*

**histograma** con una línea de densidad de kernel (KDE) superpuesta. Este gráfico ilustra cómo se distribuye la cantidad de palabras en los tweets.



# Limpieza de Datos



— Limpiamos caracteres extraños en el texto

Elimina URLs

Elimina emojis

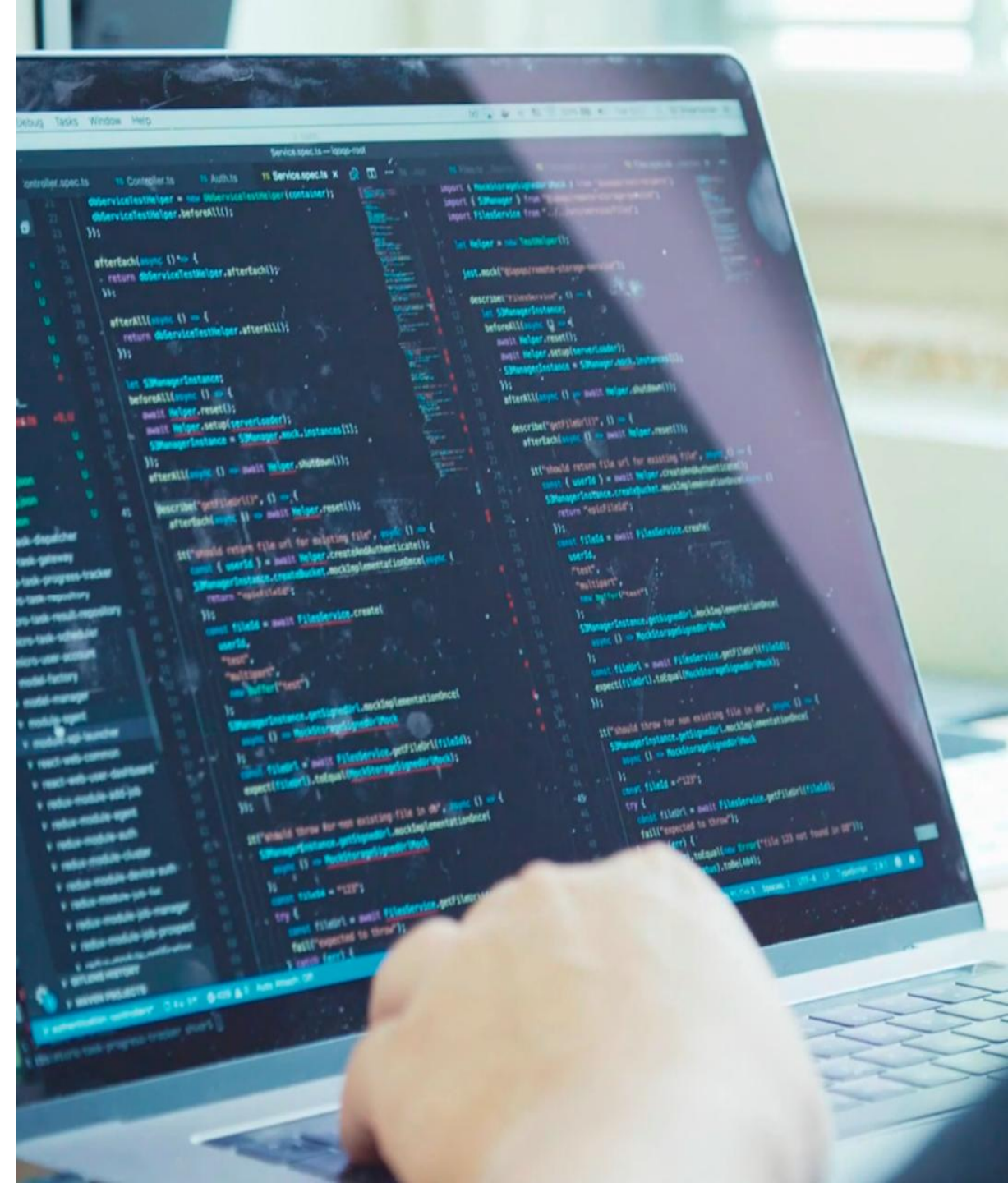
Elimina caracteres especiales(excepto letra ,números y espacios)

Elimina menciones a usuarios @usuario

Texto Original	Texto Limpio
"@user ¡Hola! Visita <a href="https://example.com">https://example.com</a> 😊 #Python"	"hola visita python"

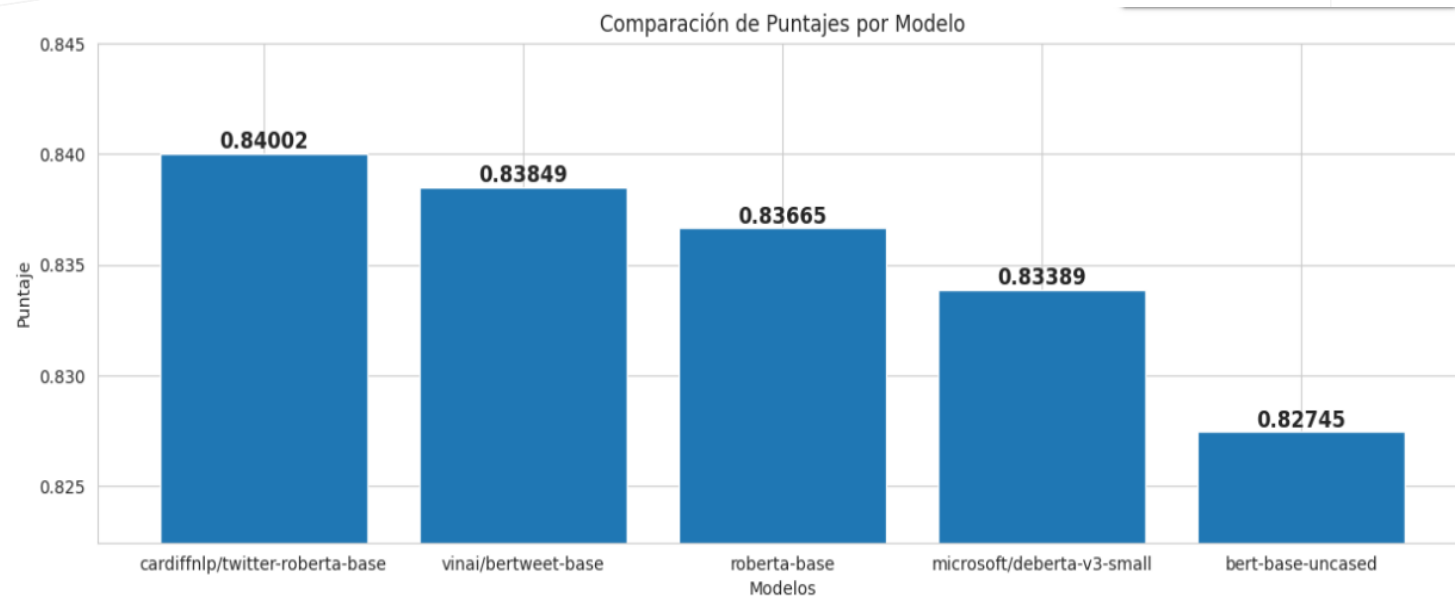
# Entrenar Modelo

- Modelos Evaluados
  - ✓ Vinai/bertweet-base
  - ✓ Cardiffnlp/twitter-roberta-base
  - ✓ Microsoft/deberta-v3-small
  - ✓ Bert-base-uncased
  - ✓ Roberta-base
- Se implementó **optimizador Adam**: permite ajustar dinámicamente la tasa de aprendizaje de cada parámetro.
- Para realizar la **validación** utilizamos el **error F1 Score**, garantizando así una evaluación más precisa del rendimiento del modelo.



# Modelos Aplicados

Modelo	Tiempo (segundos)
Vinai/bertweet-base	428.20
Cardiffnlp/twitter-roberta-base	432.55
Microsoft/deberta-v3-small	287.78
Bert-base-uncased	431.77
Roberta-base	429.58



MAX\_LEN=125 , BATCH=32 , EPOCHS=3 ,LEARNING\_RATE=2e-5



# Matriz de Confusión

Matriz de Confusión cardiffnlp/twitter-roberta-base

Valores Reales	Predicciones	
	No Desastre	Desastre
No Desastre	380	55
Desastre	67	260

Matriz de Confusión vinai/bertweet-base

Valores Reales	Predicciones	
	No Desastre	Desastre
No Desastre	389	46
Desastre	67	260

Matriz de Confusión roberta-base

Valores Reales	Predicciones	
	No Desastre	Desastre
No Desastre	387	48
Desastre	69	258

La matriz muestra información de  
Verdadero Negativo (VN):379  
Falso Positivos (FP):47  
Falso Negativos (FN):69  
Verdadero Positivo (VP):267  
ayuda a priorizar mejoras (ej:  
reducir FN si son críticos).






Matriz de Confusión microsoft/deberta-v3-small

Valores Reales	Predicciones	
	No Desastre	Desastre
No Desastre	389	46
Desastre	64	263

Matriz de Confusión bert-base-uncased

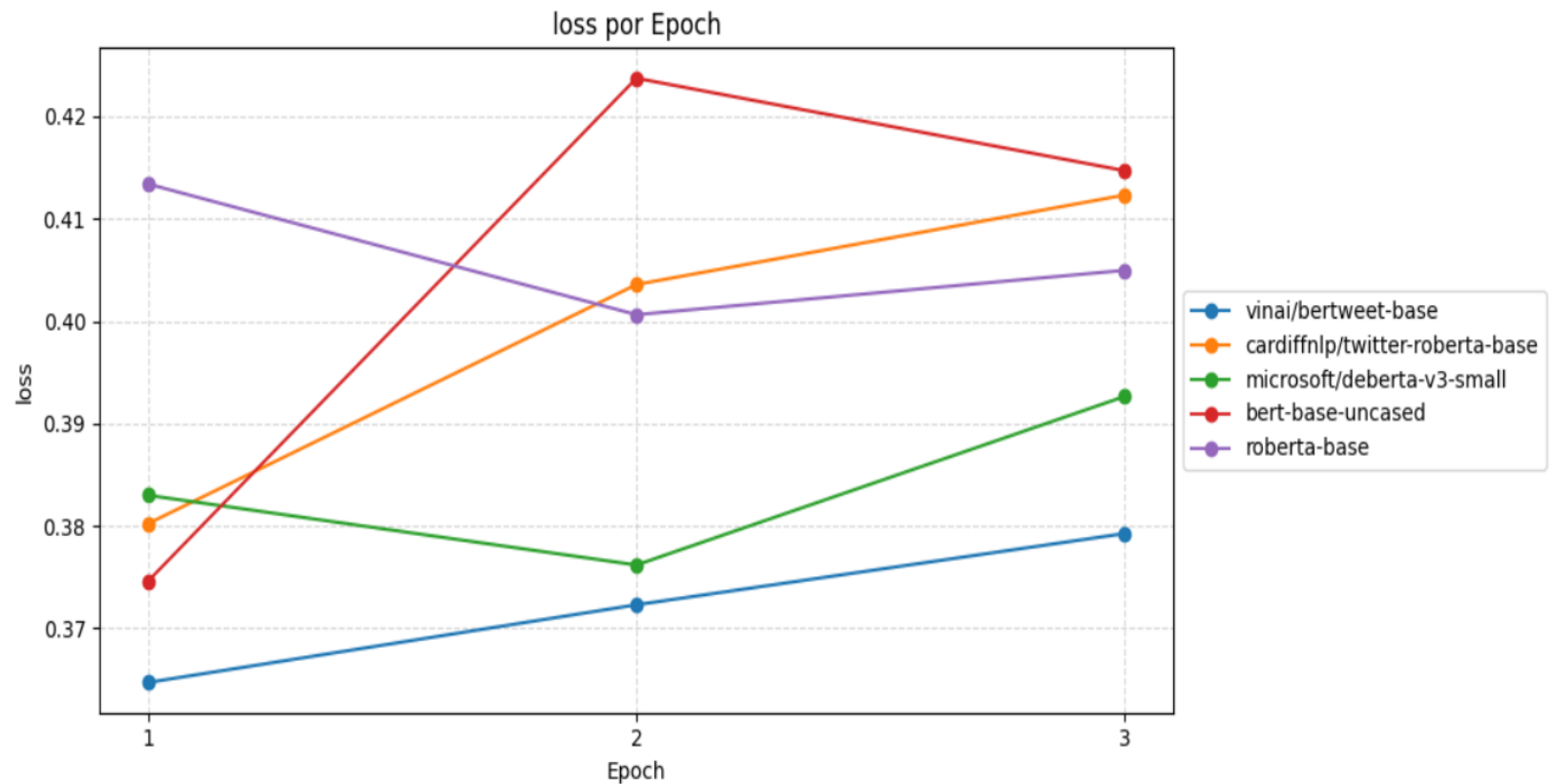
Valores Reales	Predicciones	
	No Desastre	Desastre
No Desastre	378	57
Desastre	66	261

# Comparativa entre Modelos

Modelo	Mejor F1	Mejor Accuracy	Menor Val Loss	Comentario breve
vinai/bertweet-base	0.8219	0.8504	0.3794	 Mejor F1 y Accuracy
cardiffnlp/twitter-roberta	0.8074	0.8360	0.4089	 Buen rendimiento pero algo inconsistente
microsoft/deberta-v3-small	0.8076	0.8333	0.4263	 Muy estable, pero F1 ligeramente menor
bert-base-uncased	0.8137	0.8360	0.4171	 Pierde rendimiento en la última época
roberta-base	0.8111	0.8412	0.3929	 Consistente y competitivo

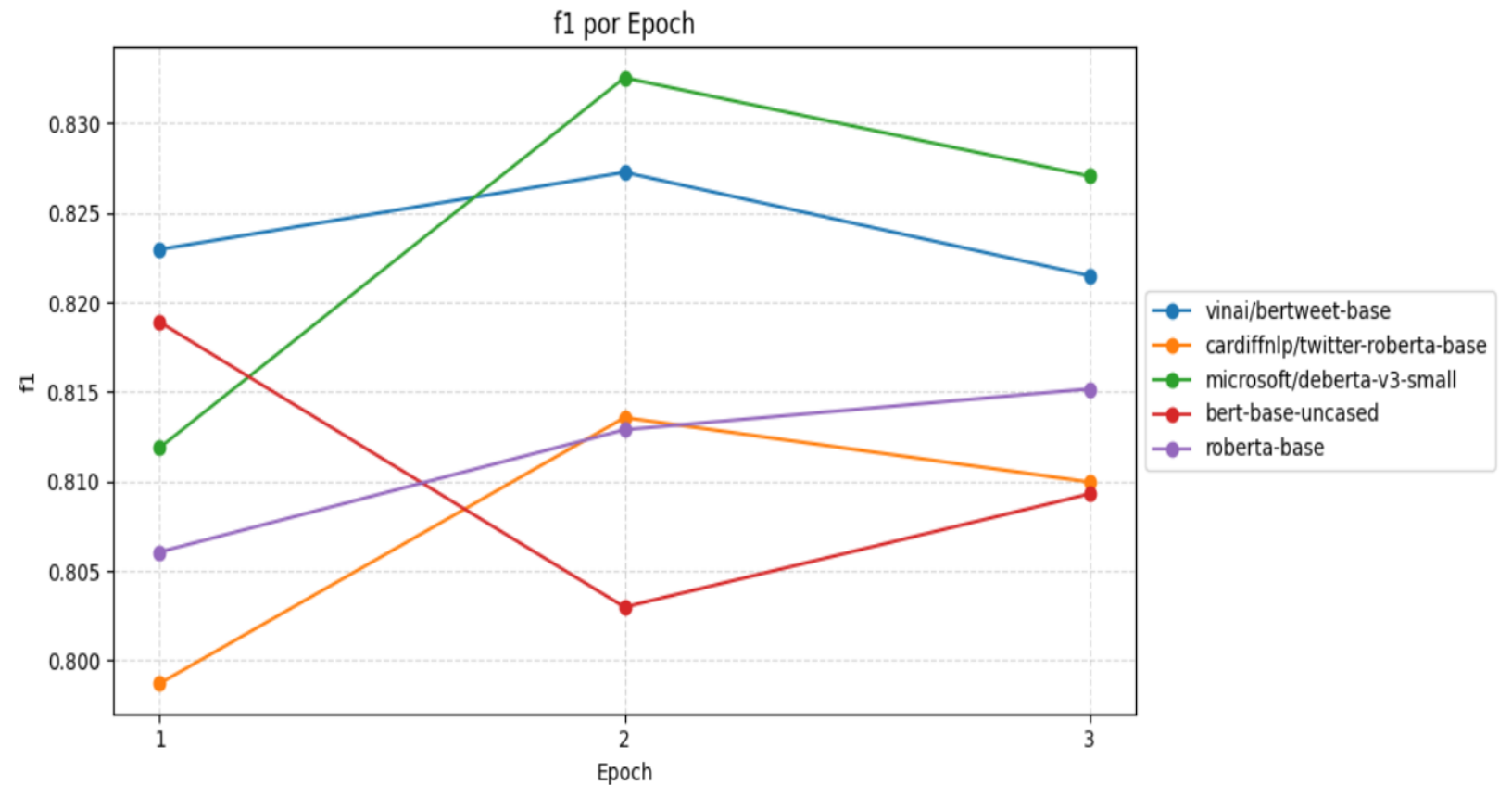
# Comparación de la validación de Pérdida por Época entre Diferentes Modelos de Lenguaje

Mide la diferencia entre las predicciones del modelo y los valores reales en el conjunto de validación. Un valor más bajo indica que el modelo está ajustándose bien sin sobreajustarse.



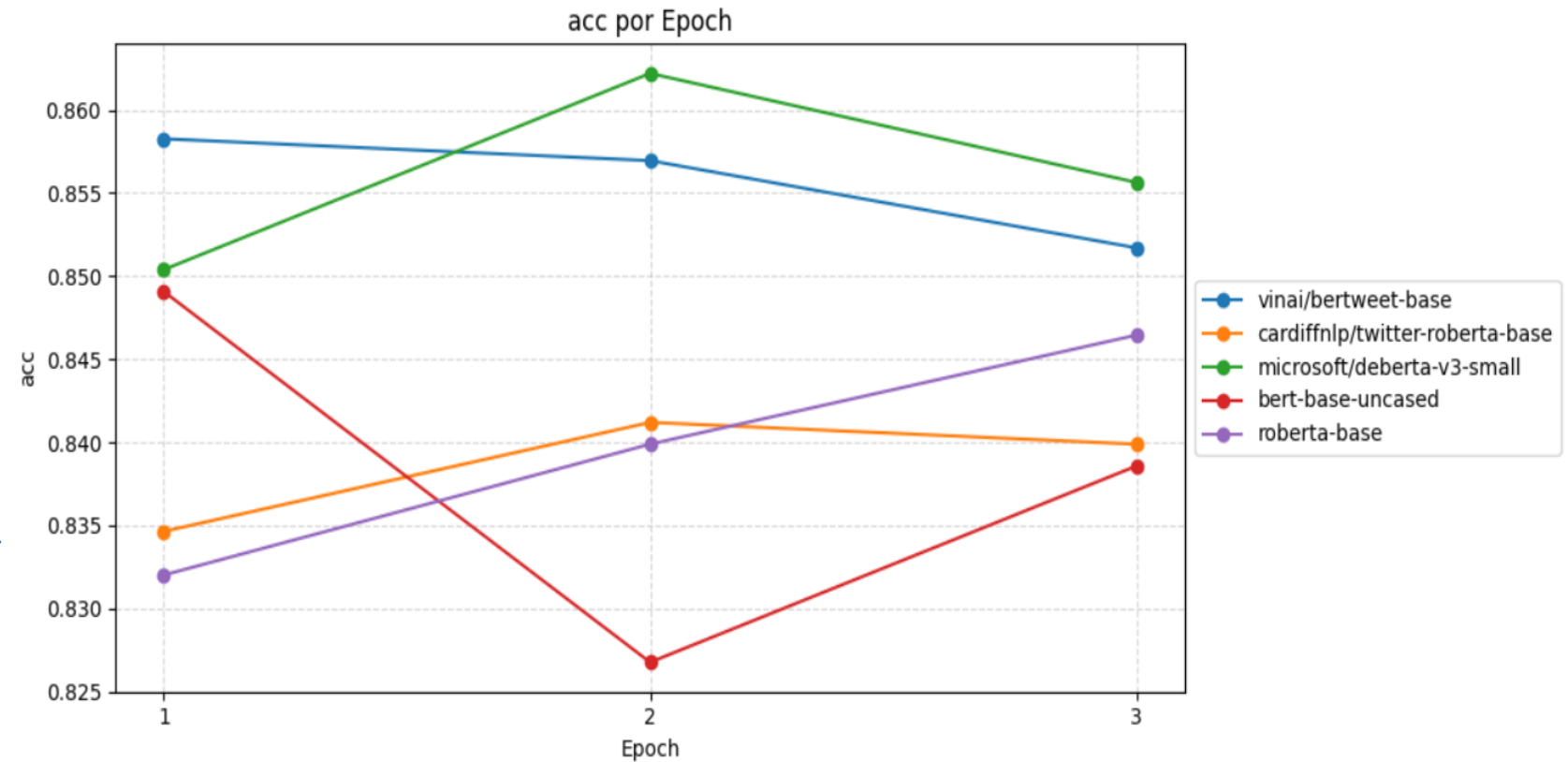
# Comparación del valor F1 de validación por Época entre Diferentes Modelos de Lenguaje

Es una métrica que combina precisión y exhaustividad (*precision & recall*) en un solo valor. Es útil cuando hay clases desbalanceadas, ya que captura tanto los falsos positivos como los falsos negativos de manera equilibrada



# Comparación del valor Accuracy por Época entre Diferentes Modelos de Lenguaje

Indica el porcentaje de ejemplos correctamente clasificados por el modelo en el conjunto de validación. Es una métrica más simple y directa, pero puede no ser la mejor si las clases no están bien distribuidas.



# Técnica de aprendizaje en conjunto

```
# Función para votación mayoritaria
def majority_vote(row):
    votes = [row[model] for model in MODELS if model in row]
    return Counter(votes).most_common(1)[0][0]
```

## Modelos:

- ✓ Vinai/bertweet-base
- ✓ Cardiffnlp/twitter-roberta-base
- ✓ Microsoft/deberta-v3-small

La votación mayoritaria (Majority Vote) es una técnica utilizada en Ensemble Learning para combinar múltiples modelos y obtener una predicción final más robusta. Básicamente, cada modelo individual genera una predicción y el resultado que recibe más votos es el que se elige como la decisión final.





# Resultados y Conclusiones

0.84308

**Puntaje Final**

Métrica de evaluación del modelo

51

**Posición**

Entre 965 participantes

51

CesteGrupo42025



0.84308

1

8s

5

**Modelos**

Algoritmos de evaluados