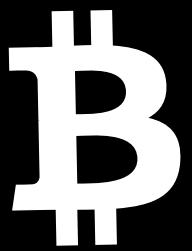


MAESTRÍA EN INTELIGENCIA ARTIFICIAL



# ANALISIS DE SERIES DE TIEMPO II

# BITCOIN

PREDICTION



DOCENTE:  
DR. CAMILO ARGOTY

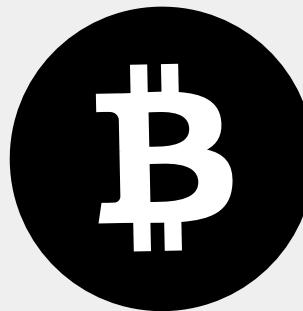
ALUMNOS:

NOELIA QUALINDI  
FABIAN SARMIENTO  
JORGE VALDEZ  
MATIAS MARANDO

# CONTENIDO

- 01** 1. Introducción
- 02** 2. Dataset
- 03** 3. EDA
- 04** 4. Ventanas y modelos elegidos
- 05** 5. NHiTS
- 06** 6. Transformer Vanilla
- 07** 7. Modelo TFT
- 08** 8. Informer
- 09** 9. Predicciones
- 10** 10. Métricas
- 11** 11. Conclusiones

# INTRODUCCIÓN



Predicción de criptomonedas  
muy volátiles



Volatilidad y  
sensibilidad a factores  
externos

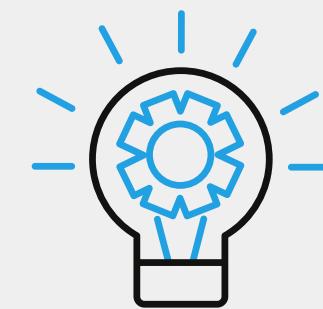


Desafío: predicción del Bitcoin a  
corto plazo

# DATASET



Cuenta con información del valor del bitcoin histórico desde el año 2012.



El dataset se actualiza todos los días. Se puede consumir por API.



Se acota la toma de datos desde el 1/01/2020 al 31/12/2023



**Bitcoin Historical Data**

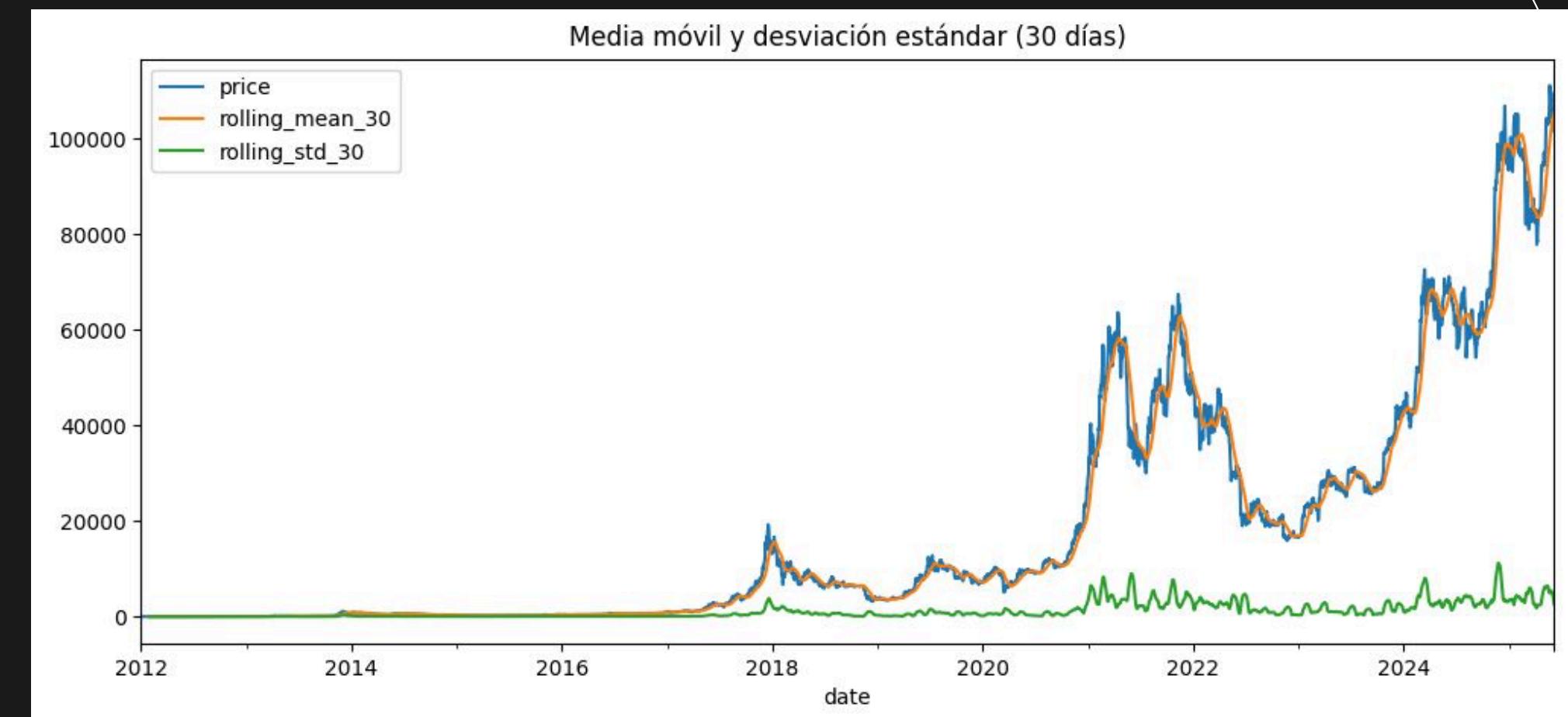
Bitcoin data at 1-min intervals from select exchanges, Jan 2012 to Present

[kaggle.com](https://www.kaggle.com)



# EDA Y PREPROCESAMIENTO

- LIMPIEZA DE DATOS
- CONVERSIÓN DE TIEMPOS
- ORDENAMIENTO TEMPORAL
- RESAMPLEO DIARIO
- NORMALIZACIÓN
- CREACIÓN DE SECUENCIA



# VENTANAS Y MODELOS ELEGIDOS PARA LAS SERIES

1 AÑO

6 MESES

3 MESES

1

2

3

4

NHiTS

Transformer

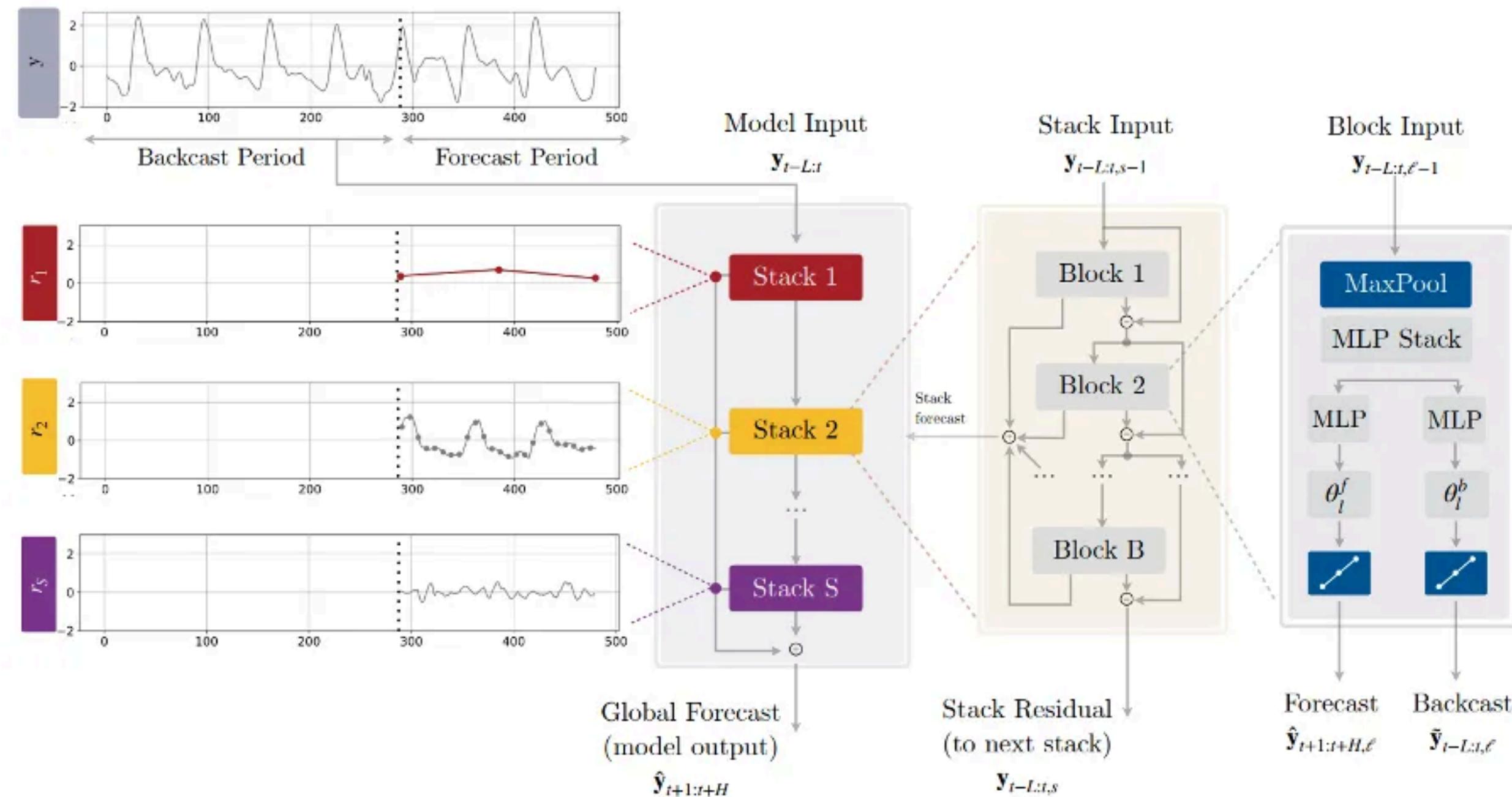
TFT

Informer



4

# Interpolación jerárquica neuronal para pronóstico de series de tiempo



- DESCOMPOSICIÓN JERÁRQUICA MULTIRESOLUCIÓN
- PREDICCIÓN POR BLOQUES Y APRENDIZAJE RESIDUAL
- INTERPOLACIÓN BASADA EN FUNCIONES BASE

# NHiTS

Interpolación jerárquica neuronal  
para pronóstico de series de tiempo

```
common_args = {
    'input_chunk_length': min(60, window_size//3),
    'output_chunk_length': PREDICTION_HORIZON,
    'random_state': SEED,
    'n_epochs': 100,
    'batch_size': 32,
    'optimizer_kwargs': {'lr': 1e-3},
    'model_name': f'{config_name}_model'
}
```

PREDICTION\_HORIZON = 10 (días)

SEED = 42

```
nhits = NHiTSModel(
    **common_args,
    num_blocks=2,
    num_layers=3,
    layer_widths=128,
    dropout=0.1
)
```

Librería utilizada: Darts ([forecasting.nhits](#)).

# TRANSFORMER VANILLA

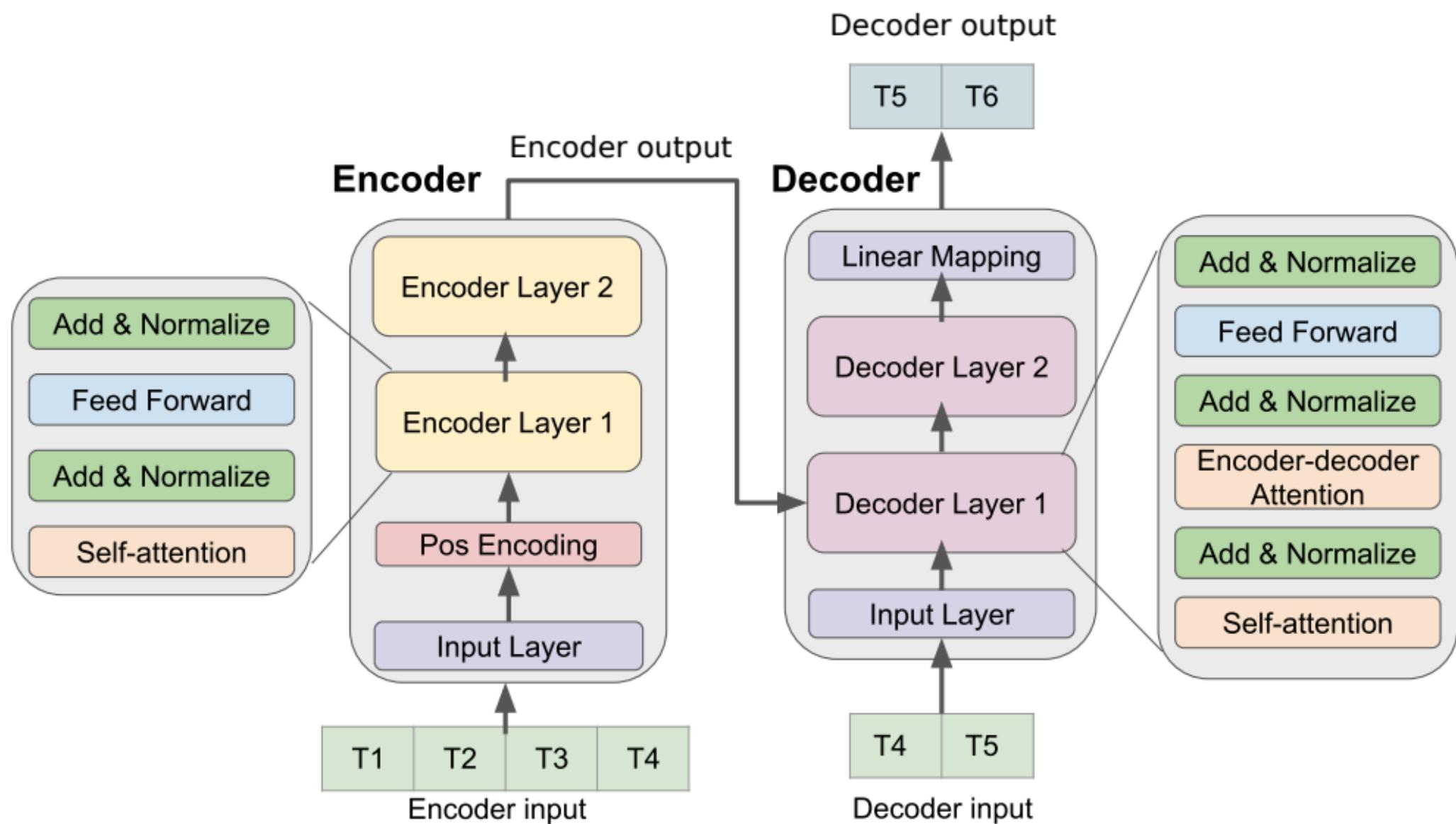


Figure 1. Architecture of Transformer-based forecasting model.

# TRANSFORMER VANILLA

Librería utilizada: Darts ([forecasting.transformer\\_model](#))

## Transformer Model

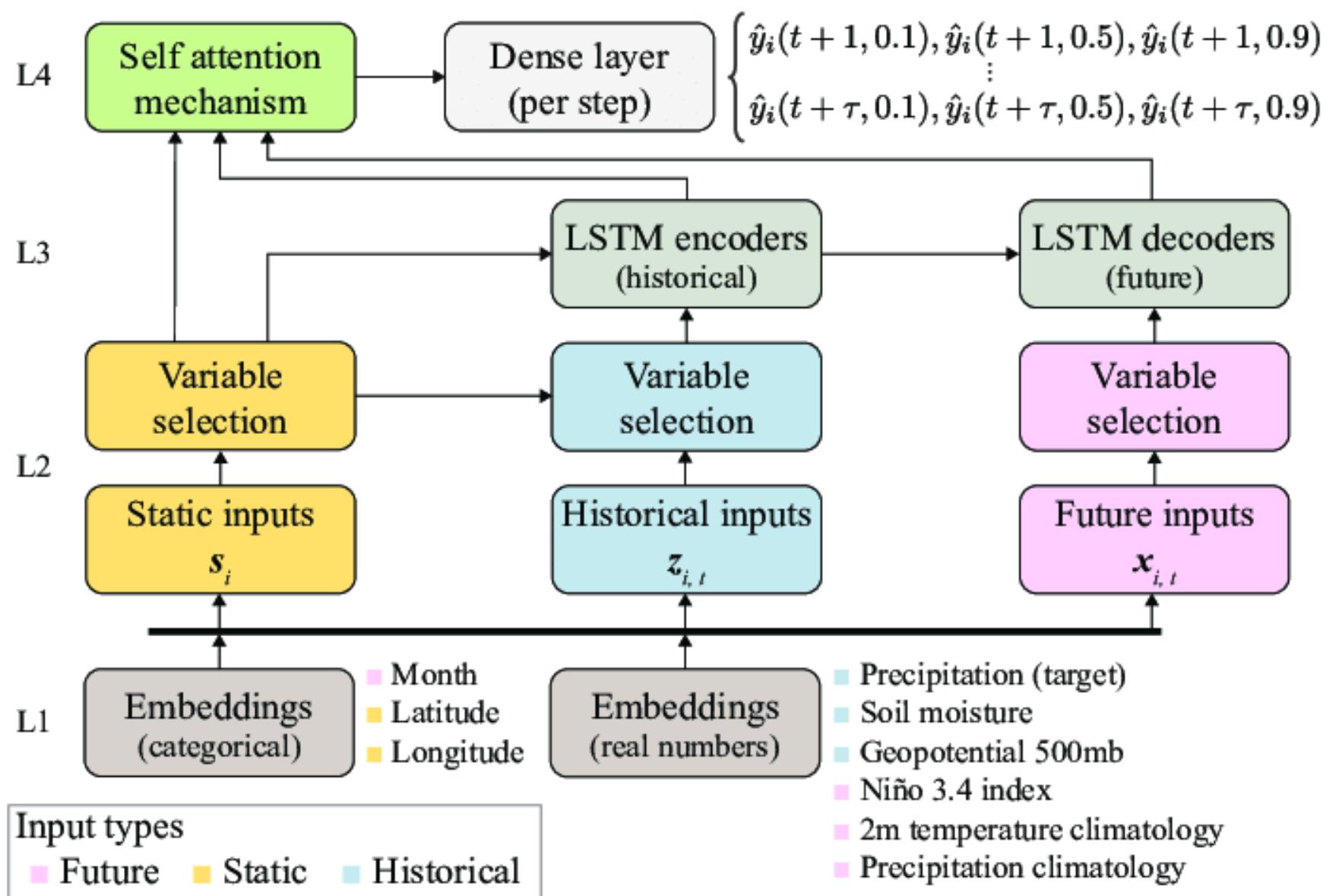
```
class
darts.models.forecasting.transformer_model.TransformerModel(input_chunk_length,
output_chunk_length, output_chunk_shift=0, d_model=64, nhead=4,
num_encoder_layers=3, num_decoder_layers=3, dim_feedforward=512, dropout=0.1,
activation='relu', norm_type=None, custom_encoder=None, custom_decoder=None,
**kwargs)
```

[source]

```
# Configuración común de modelos
common_args = {
    'input_chunk_length': min(60, window_size//3),
    'output_chunk_length': PREDICTION_HORIZON,
    'random_state': SEED,
    'n_epochs': 100,
    'batch_size': 32,
    'optimizer_kwargs': {'lr': 1e-3},
    'model_name': f'{config_name}_model'
}

# Inicializar modelos
transformer = TransformerModel(
    **common_args,
    d_model=64,
    nhead=4,
    num_encoder_layers=3,
    num_decoder_layers=3,
    dropout=0.1
)
```

# TEMPORAL FUSION TRANSFORMER (TFT)



Combinación de:

**LSTM** (procesa secuencias temporales) + **Multihead attention** (captura relaciones complejas entre variables)

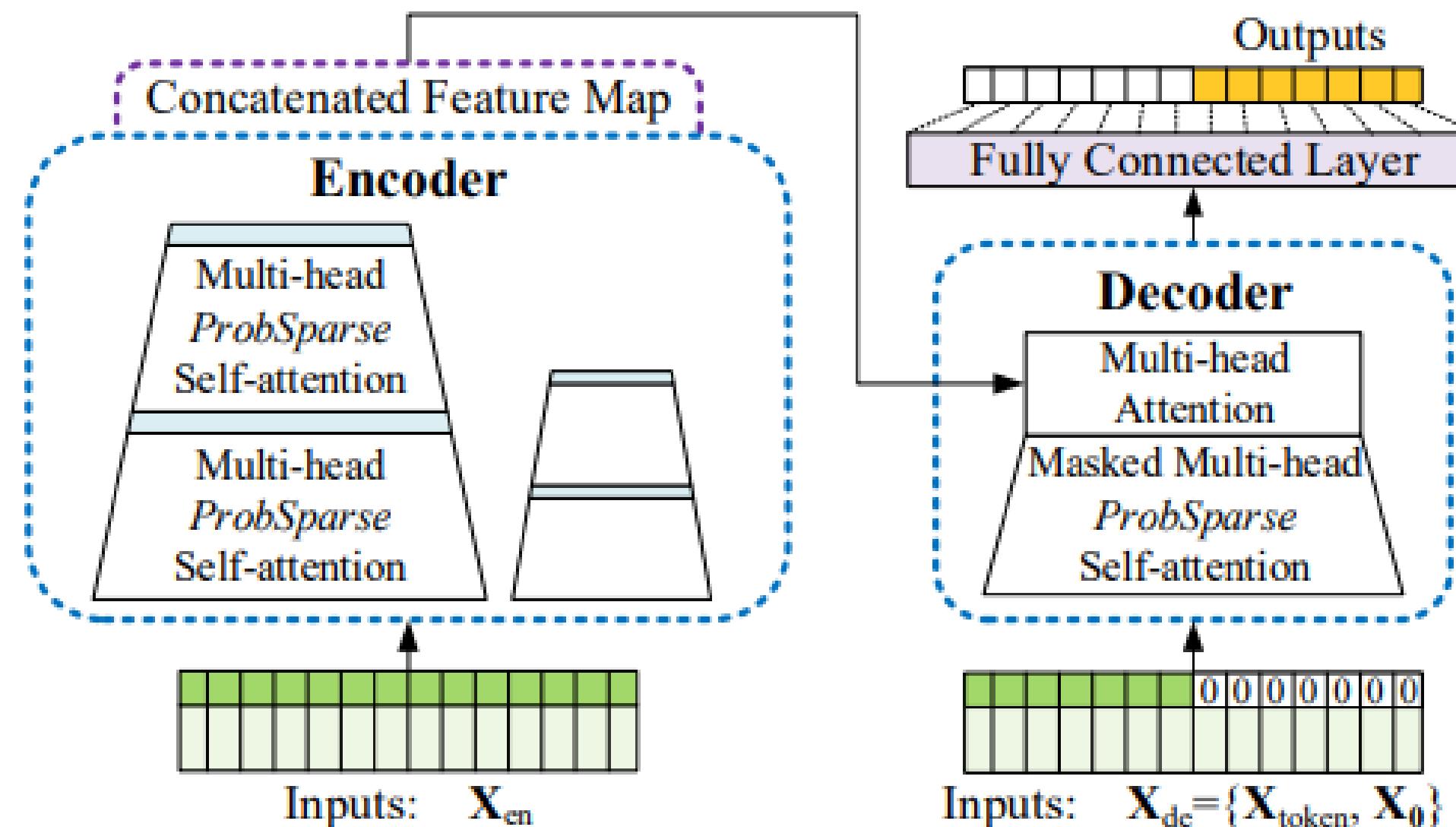
# TEMPORAL FUSION TRANSFORMER (TFT)

Librería utilizada: Darts ([forecasting.tft\\_model](#))

```
tft = TFTModel(  
    **common_args,  
    hidden_size=64,  
    lstm_layers=2,  
    dropout=0.1,  
    add_relative_index=True  
)
```

PREDICTION\_HORIZON = 10 (días)  
SEED = 42

# INFORMER



Diseñada específicamente para la predicción eficiente y escalable de series temporales, en particular para tareas de predicción de secuencias largas

# INFORMER

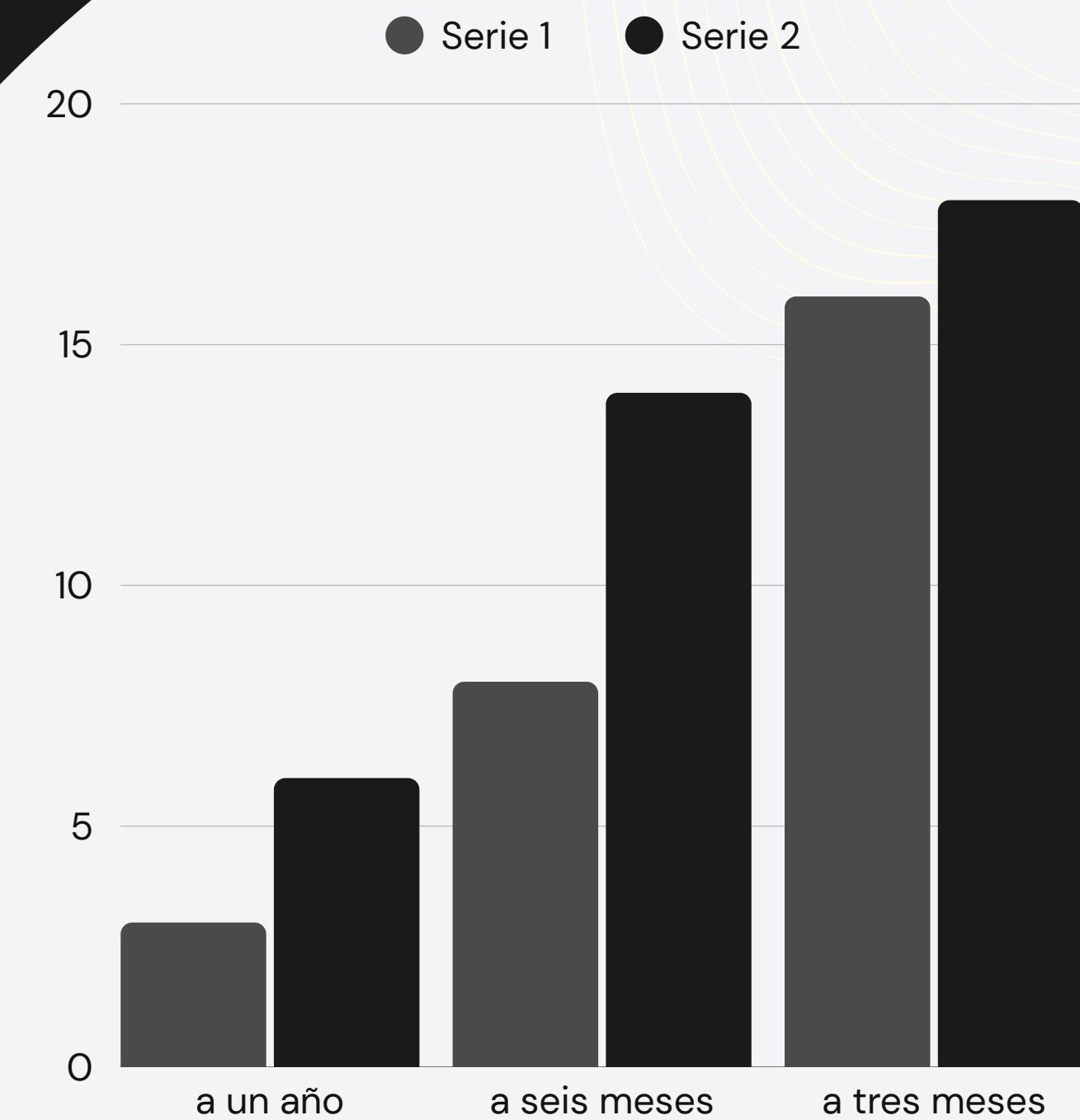
Librería utilizada: NeuralForecast ([neuralforecast.models.Informer](#))

```
# Configurar modelo Informer con mejores hiperparámetros
informer_model = Informer(
    h=PREDICTION_HORIZON,
    input_size=min(60, window_size//3),
    hidden_size=32, # Reducido para evitar overfitting
    conv_hidden_size=64, # Reducido
    n_head=2, # Reducido
    loss=MAE(),
    scaler_type='standard', # Cambio a standard scaler
    learning_rate=1e-5, # 1e-4, # Learning rate más bajo
    max_steps=500, # Más pasos de entrenamiento
    val_check_steps=25, # Validación más frecuente
    early_stop_patience_steps=5, # 5, # Más paciencia
    encoder_layers=2, # Menos capas para evitar overfitting
    decoder_layers=1,
    dropout=0.3, # 0.2, # Más dropout para regularización
    batch_size=8 # 16 # Batch size más pequeño
)
```

**PREDICTION\_HORIZON = 10 (días)**  
**SEED = 42**

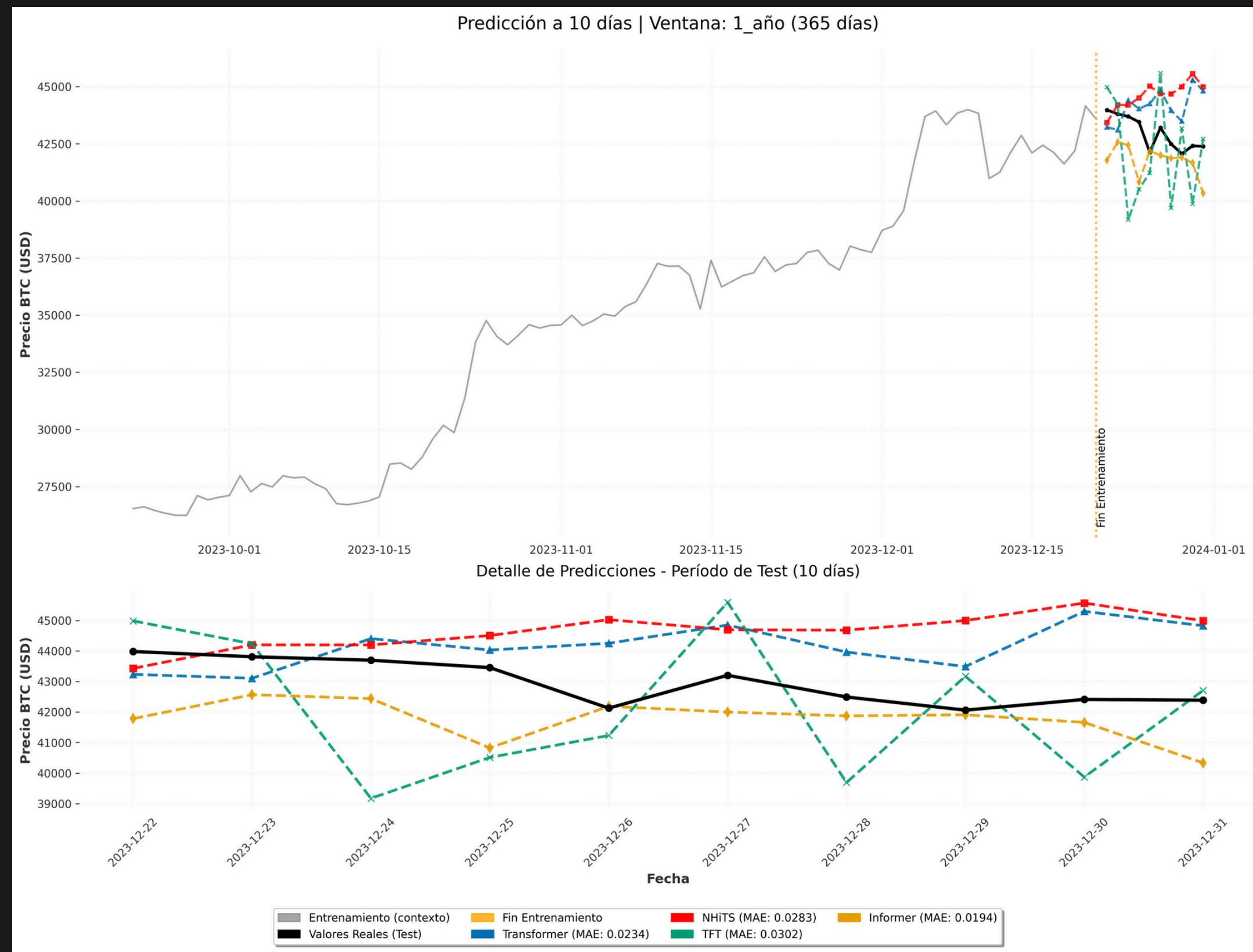
# PREDICCIONES

Se realizó la predicción a 10 días con las tres ventanas temporales elegidas.



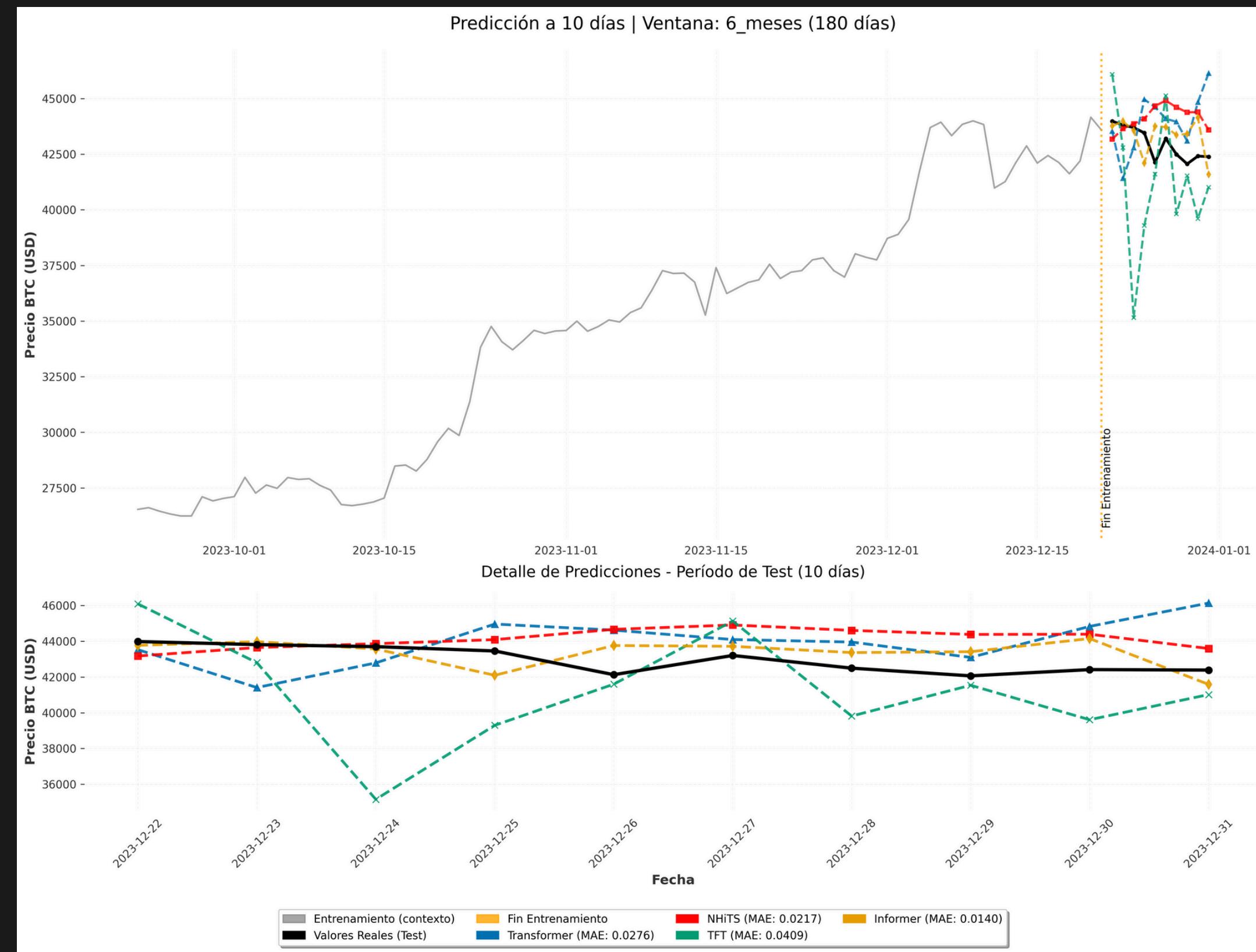
# PREDICCIONES

## VENTANA DE 1 AÑO



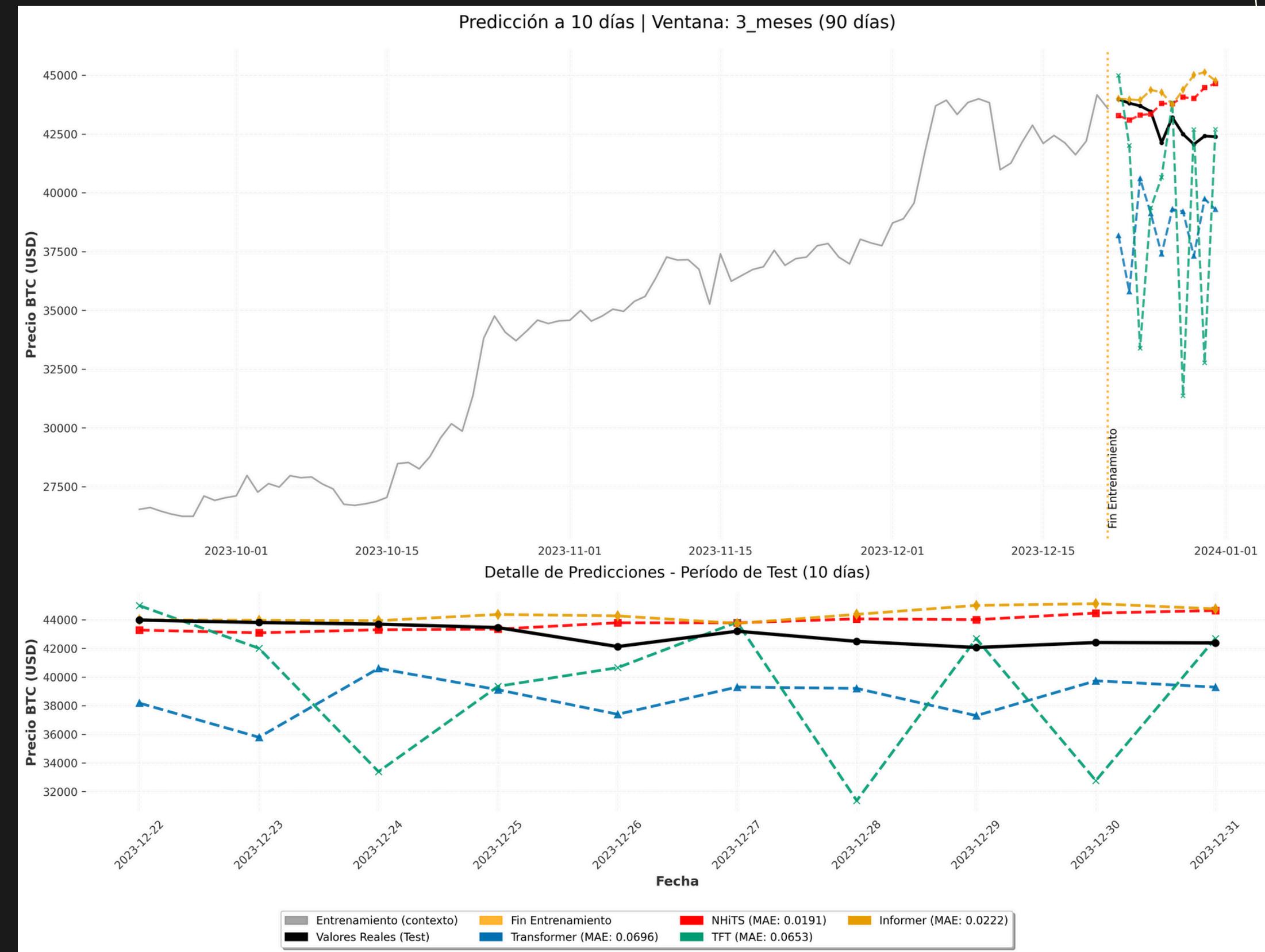
# PREDICCIONES

## VENTANA DE 6 MESES

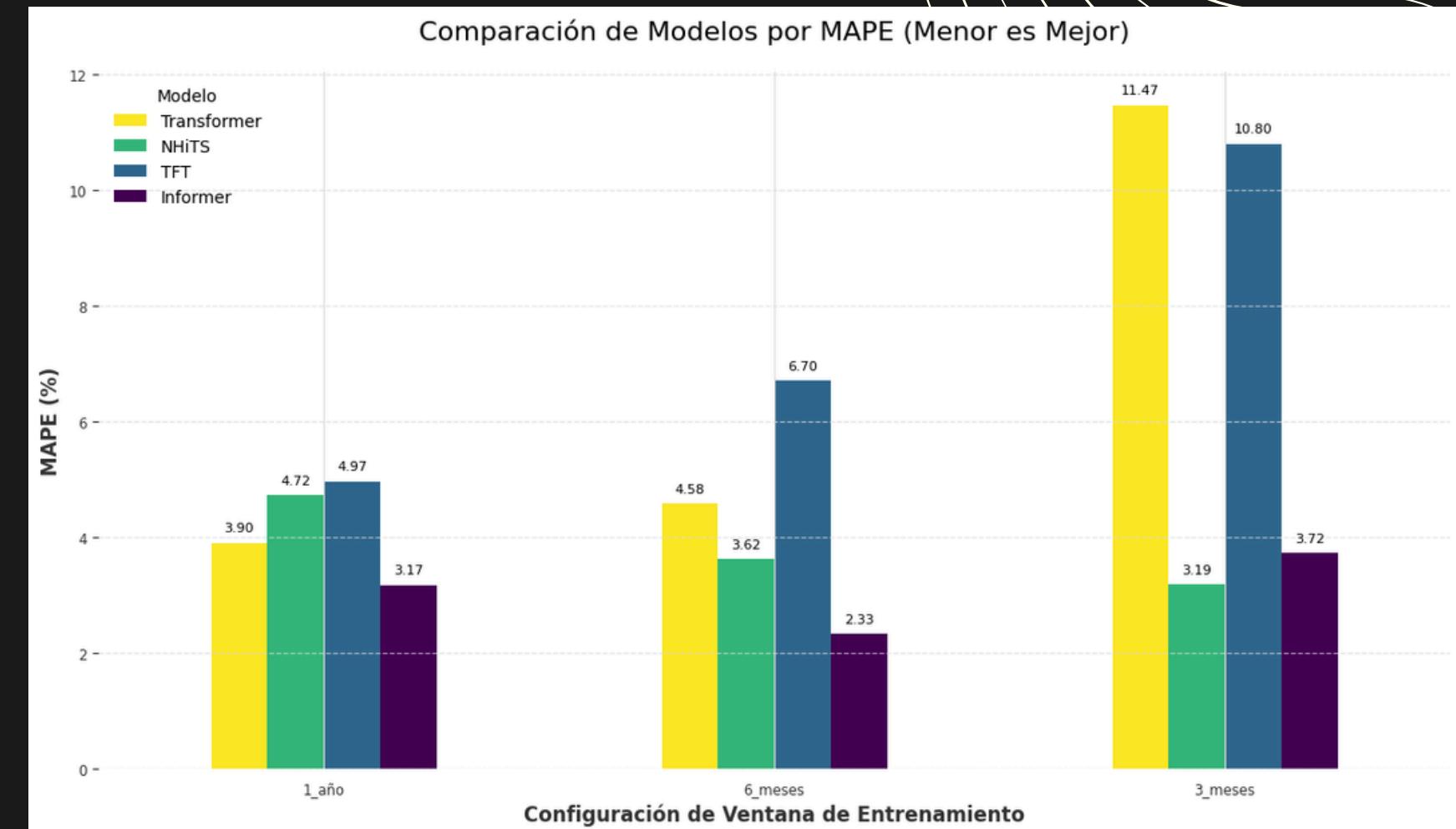
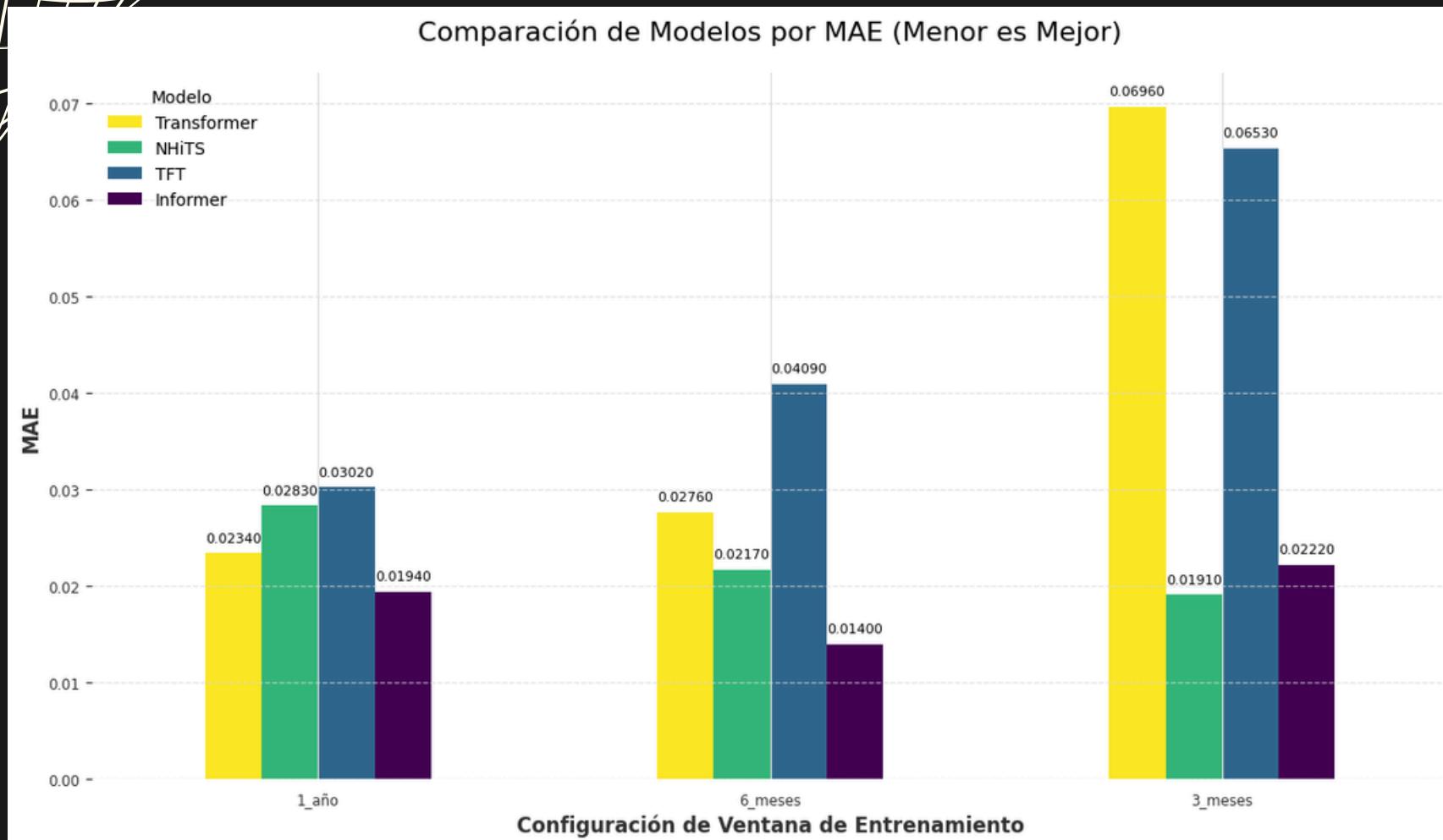


# PREDICCIONES

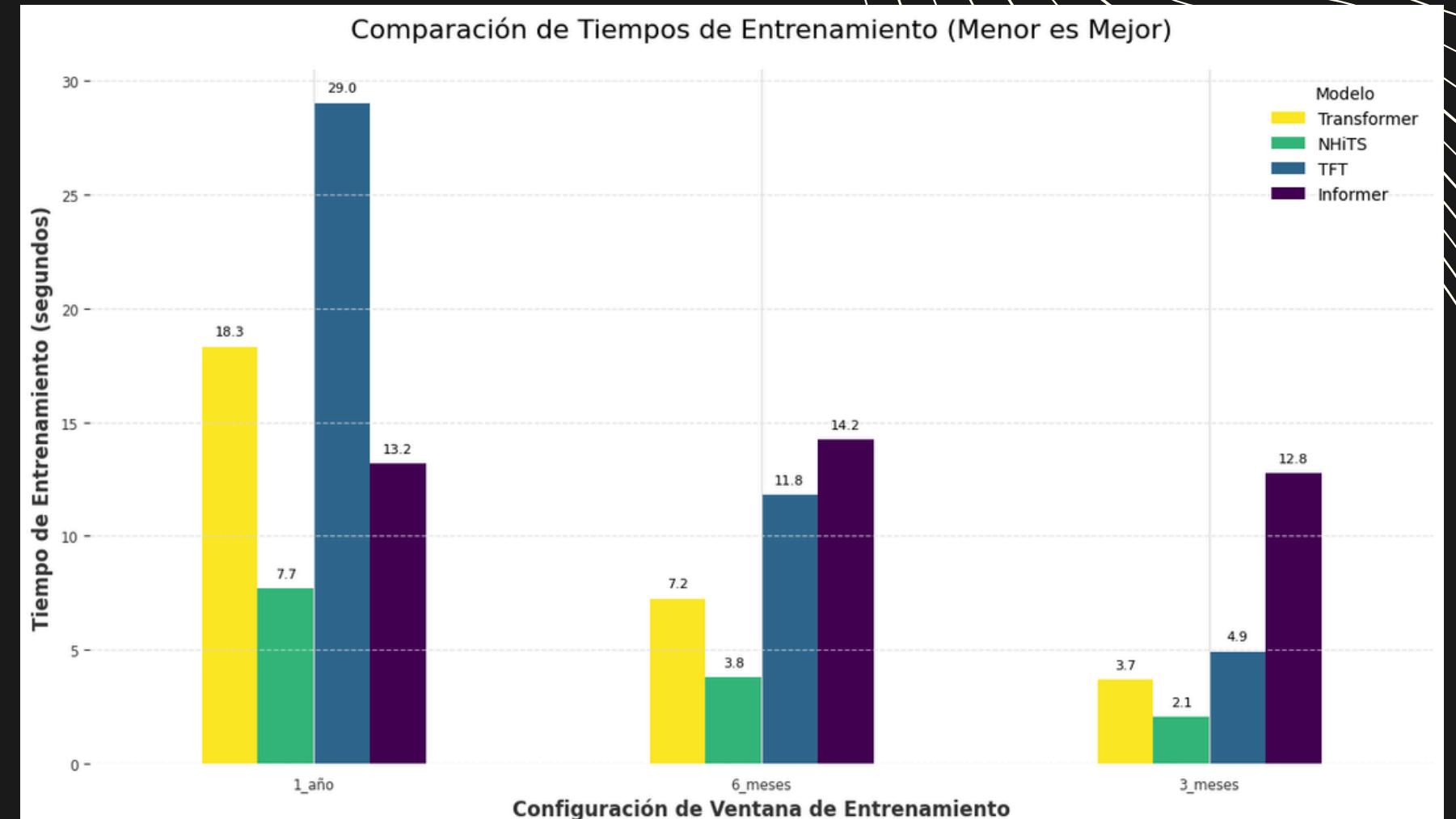
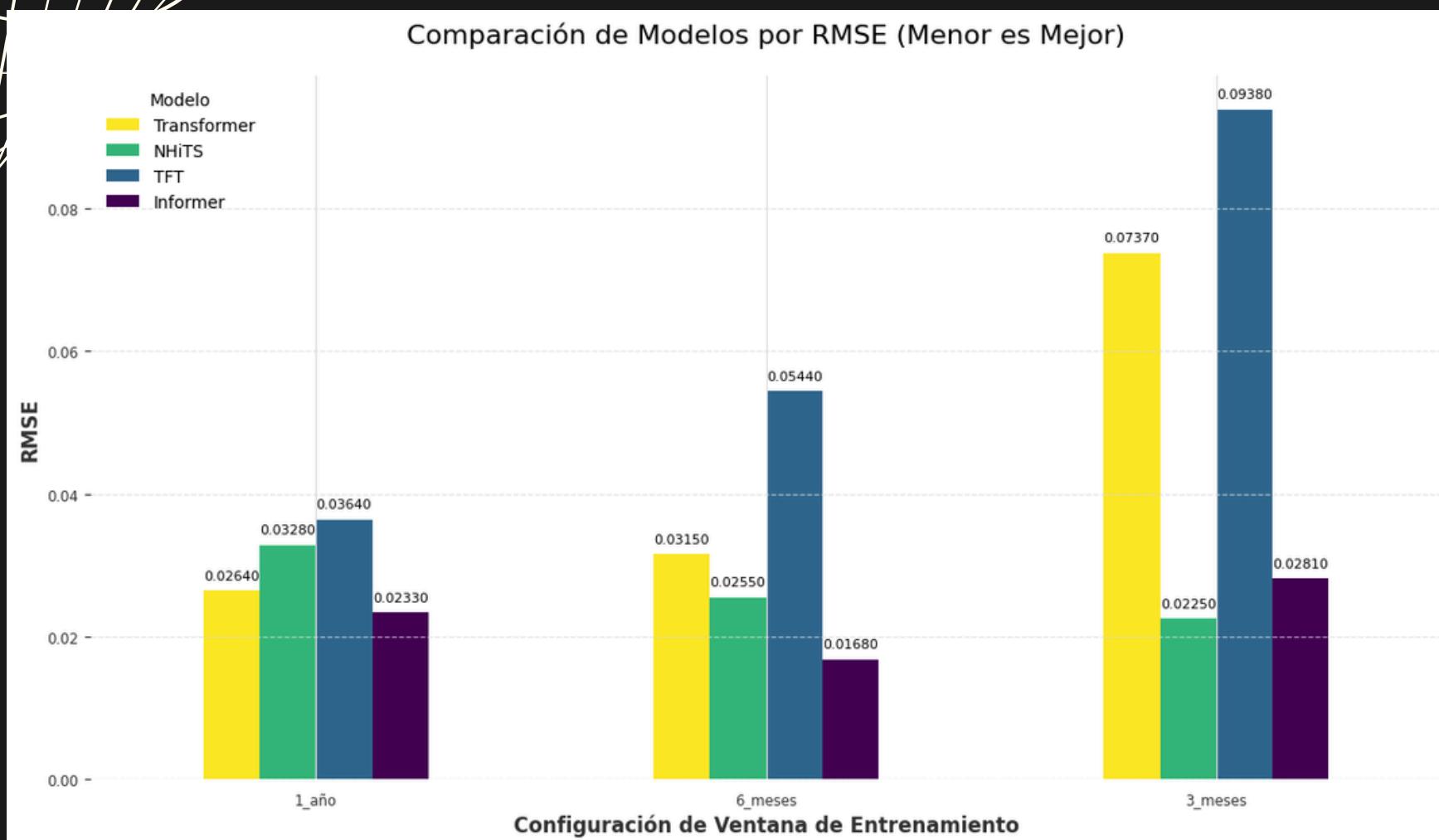
## VENTANA DE 3 MESES



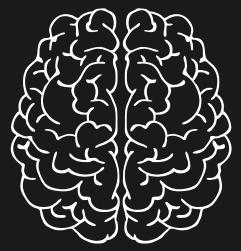
# MÉTRICAS



# MÉTRICAS



# CONCLUSIONES



MEJORES  
MODELOS  
**Informer**  
**NHiTS**



## PERFORMANCE

Los modelos mostraron  
dificultades durante  
eventos de alta  
volatilidad.



## SELF-ATTENTION

La arquitectura de  
atención demostró ser  
particularmente efectiva  
durante eventos de alta  
volatilidad.



MUCHAS  
GRACIAS!

PREGUNTAS?

