

# CENMLP

**Felipe Gutiérrez**

f.gutierrezlucero@uandresbello.edu

Ingeniería Civil Informática

**Santiago Salvador**

s.salvadorclarke@uandresbello.edu

Ingeniería Civil Informática

Abril 2024

Santiago, Chile



Facultad de  
Ingeniería

# ¿Qué es CENMLP?

También conocido como “Coordinador Eléctrico Nacional Machine Learning Project” es un proyecto desarrollado para el curso CINF104 (Aprendizaje de Máquina).



CEN



Mercado de Energía



Consumo Eléctrico

# Parte I: Análisis de la Calidad de los Datos

# Valores atípicos

Diagrama de caja para distribución de datos

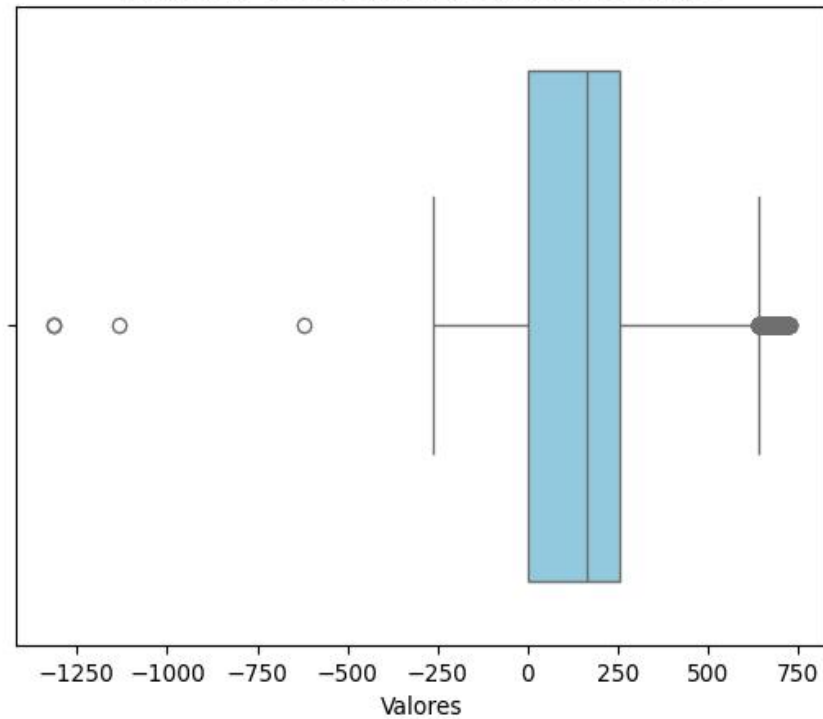
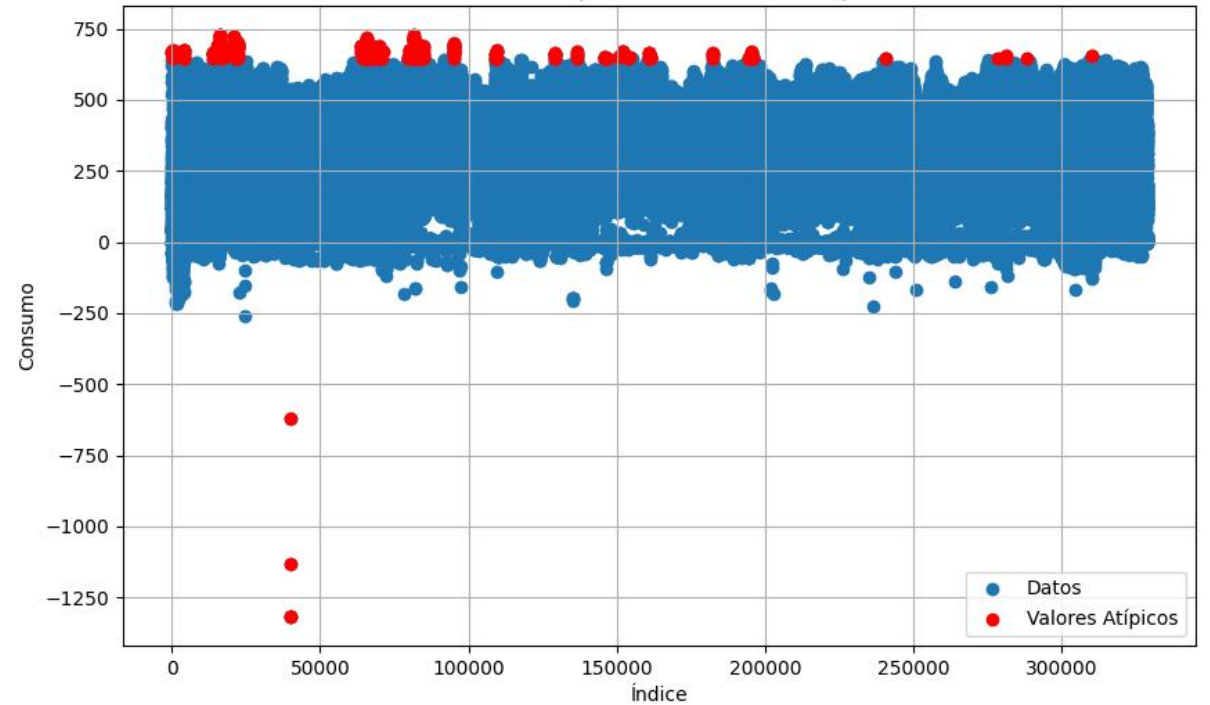
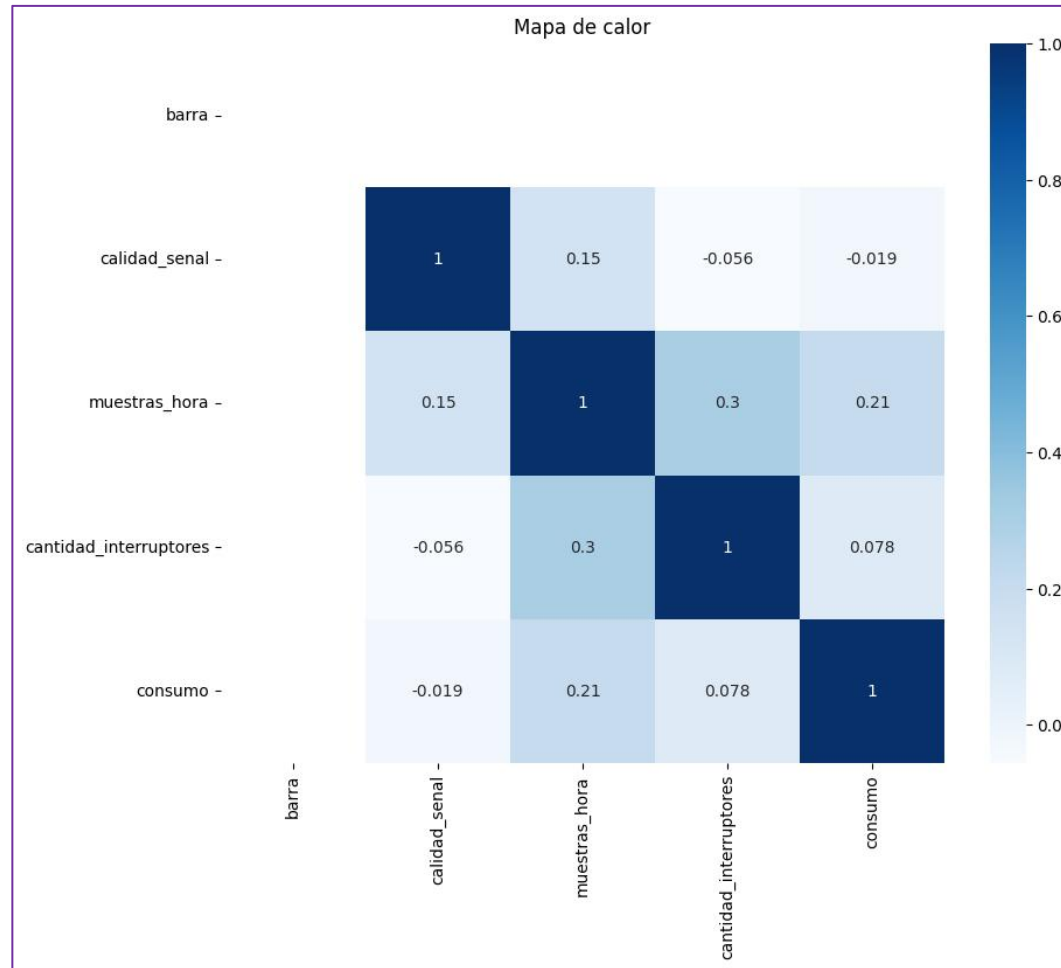


Gráfico de Dispersión con Valores Atípicos



V: Consumo medido en MWs

# Correlación



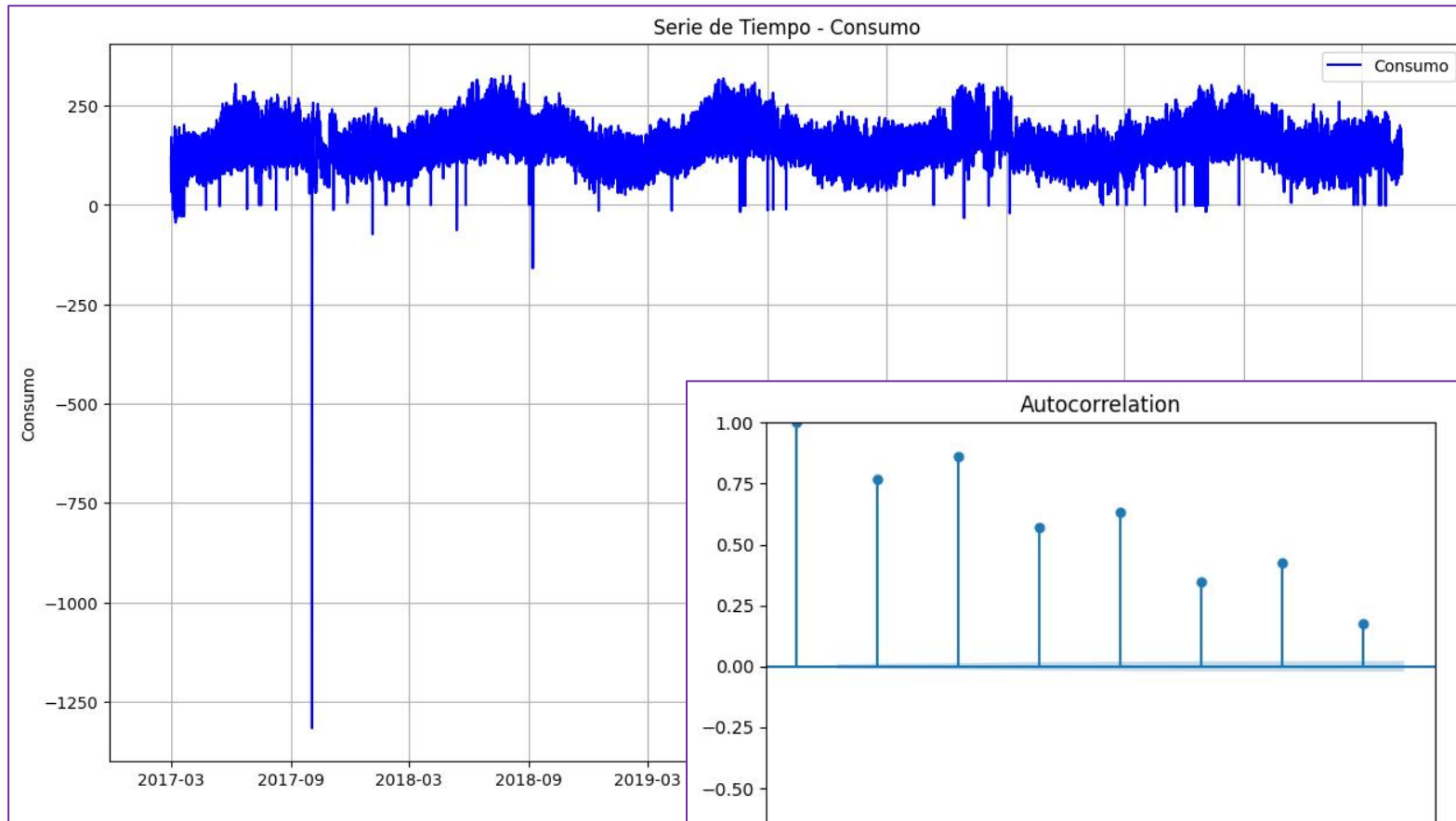
Variables:

- Barra: Variable constante, NS
- Muestras\_hora: Variable constante, NS
- Calidad\_senal: Corr.  $< \Delta$ , NS
- Cantidad\_interruptores: Corr.  $< \Delta$ , NS

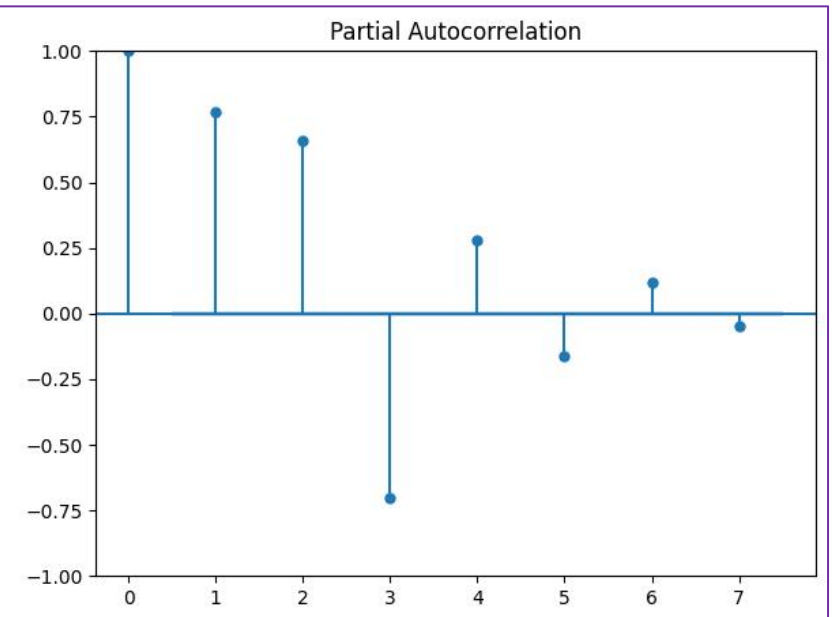
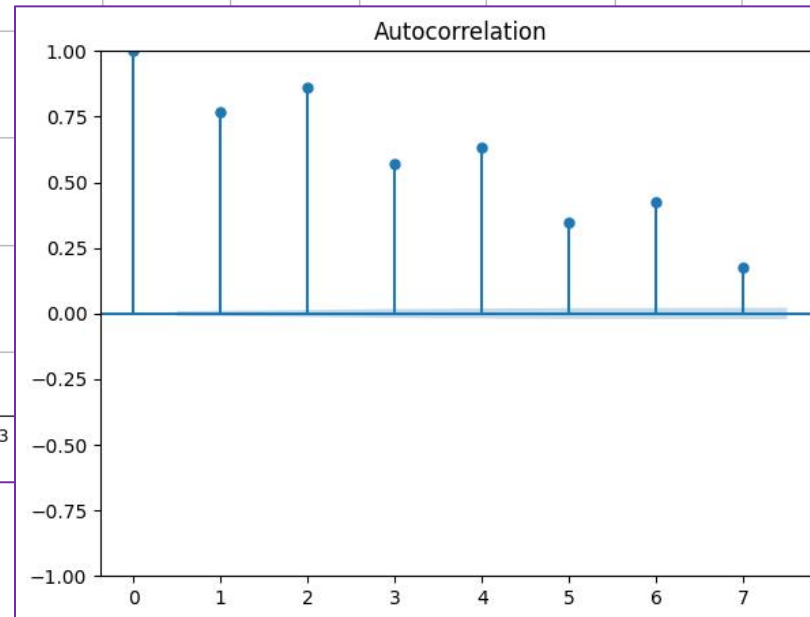
Por tanto **Consumo** y **Tiempo** únicas variables significativas y las cuales forman una serie de tiempo

$\Delta = 0.05$  (95% significancia)

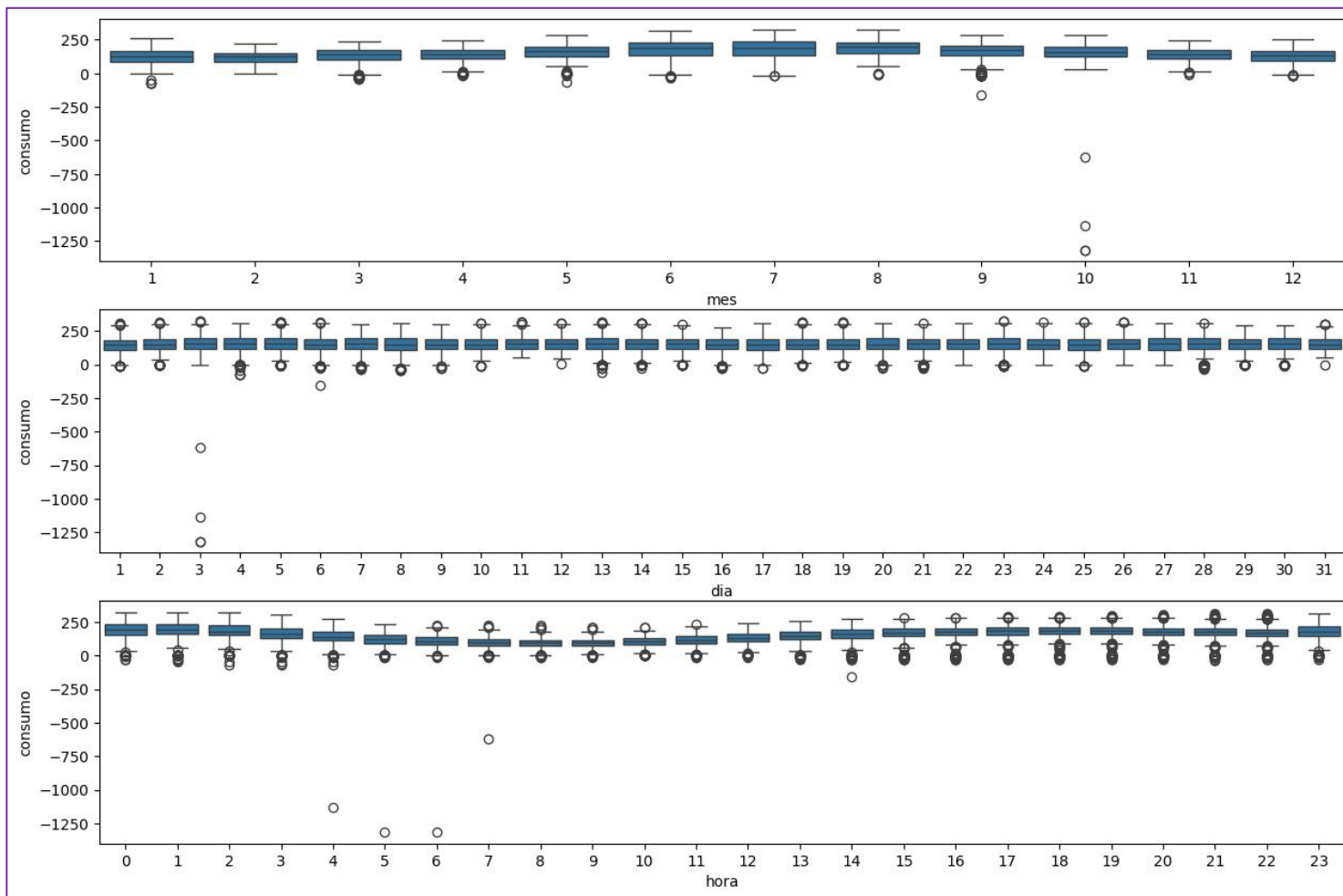
# Series de Tiempo



S: AJAHUEL



# Series de Tiempo



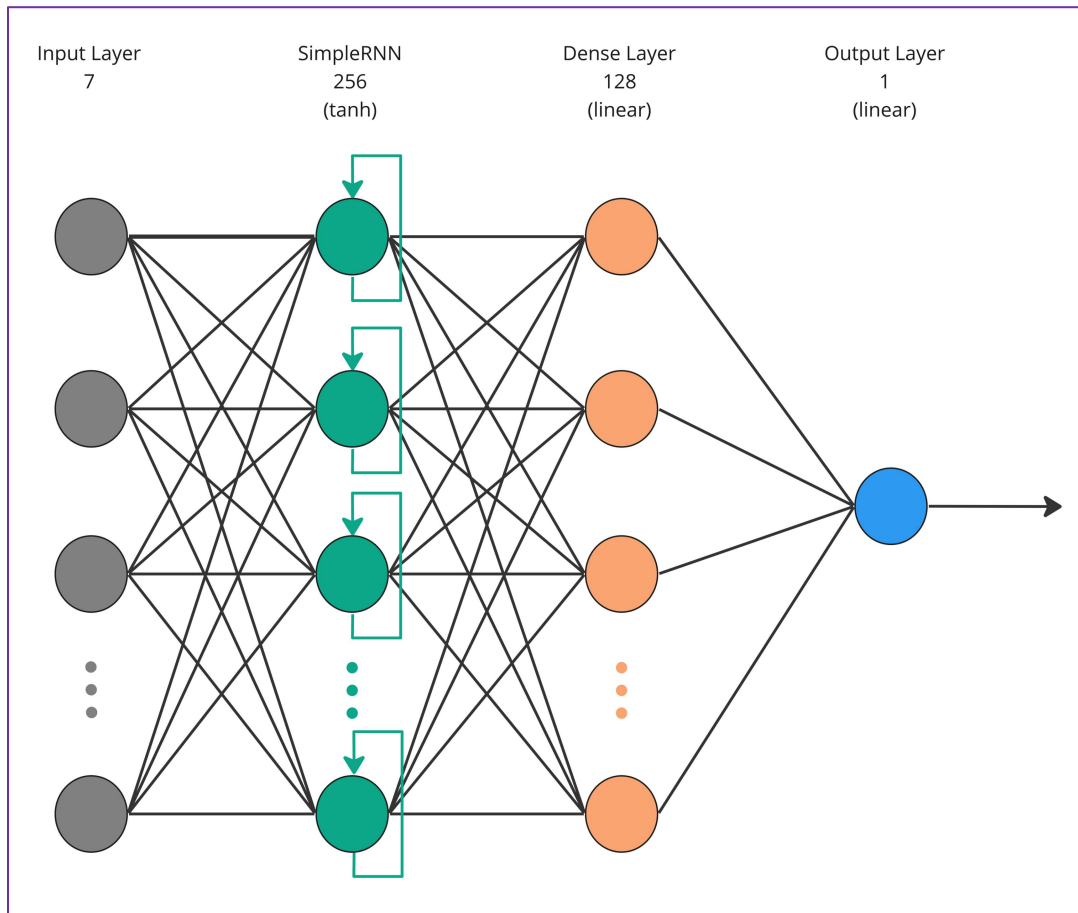
S: AJAHUEL

ADF Statistic: -12.940704  
p-value: 0.000000  
Critical Values:  
1%: -3.430  
5%: -2.862  
10%: -2.567

# Parte II: Diseño de Modelos de Aprendizaje



# Modelo 1 - RNN



S: AJAHUEL

```
# Definir conjuntos de datos de prueba y entrenamiento
train_mask = (df.index >= '2017-03-01') & (df.index <= '2021-03-31')
test_mask = (df.index >= '2021-04-01') & (df.index <= '2022-04-30')
```

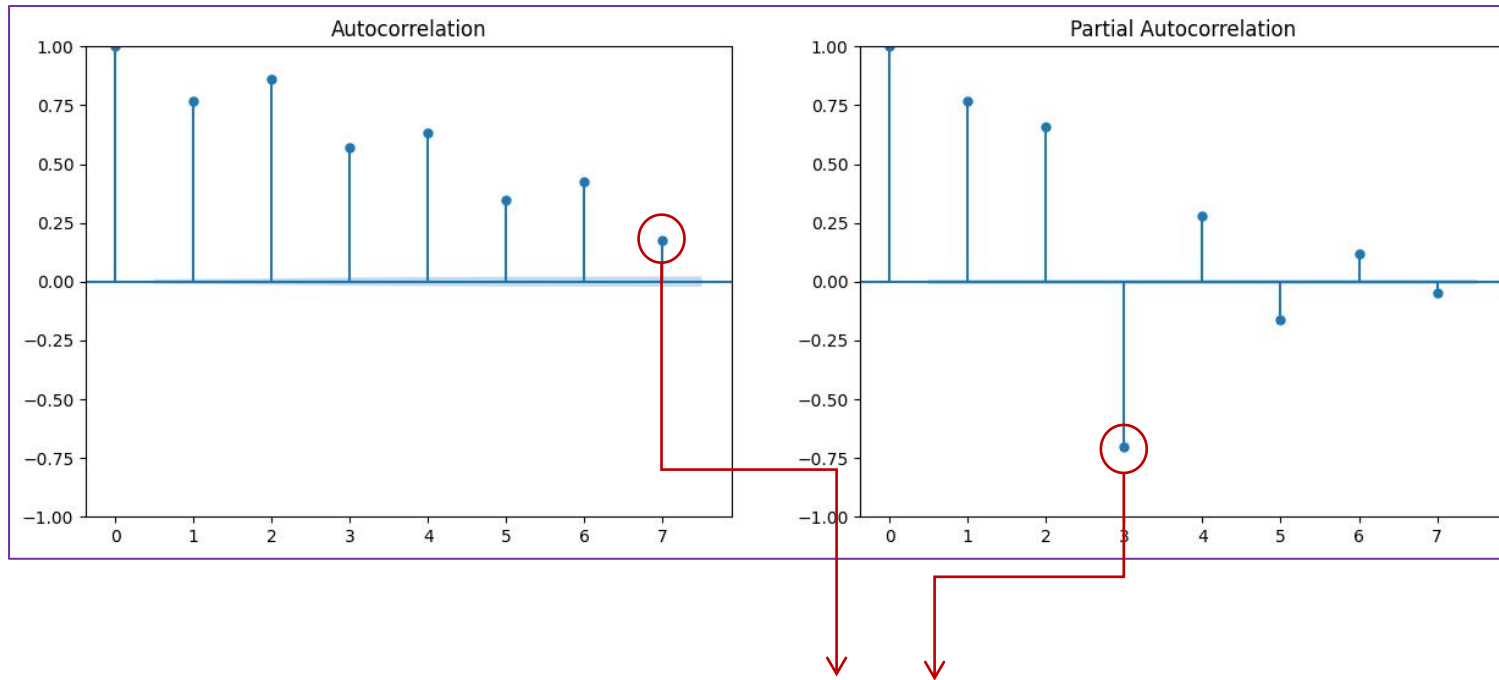
```
# Crear secuencias de tiempo
window_size = 7 # usaremos una ventana de 7 días
X_train, y_train = create_sequences(df[train_mask].values, window_size)
X_test, y_test = create_sequences(df[test_mask].values, window_size)

X_train, y_train = X_train[window_size:], y_train[window_size:]
X_test, y_test = X_test[window_size:], y_test[window_size:]
```

```
model.compile(optimizer=Adam(learning_rate=1e-3),
              loss='mse',
              metrics=['mse', 'mae'])
```

Layer (type)	Output Shape	Param #
simple_rnn (SimpleRNN)	(None, 256)	66,048
dense (Dense)	(None, 128)	32,896
dense_1 (Dense)	(None, 1)	129

# Modelo 2 - ARIMA



S: AJAHUEL

```
train = df.loc['2017-03-01':'2021-03-31'] # Filtrar datos  
test = df.loc['2021-04-01':'2022-04-30'] # Filtrar datos  
  
train.index = pd.to_datetime(train.index)  
train.index = train.index.strftime('%Y-%m-%d %H:%M:%S')  
  
test.index = pd.to_datetime(test.index)  
test.index = test.index.strftime('%Y-%m-%d %H:%M:%S')
```

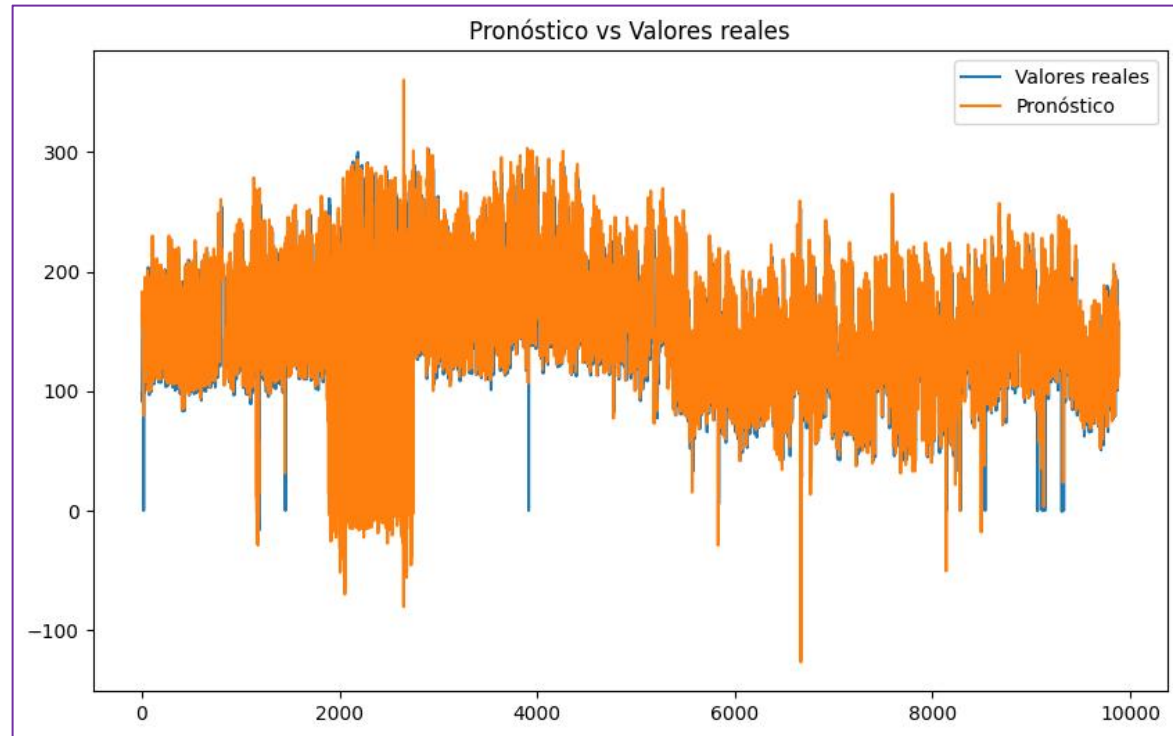
```
model = ARIMA(train["consumo"], order=(3, 0, 7)) # order=(p, d, q)  
results = model.fit()  
print("MSE :", results.mse)  
print("AIC :", results.aic)
```

# Parte III: Comparación de Modelos

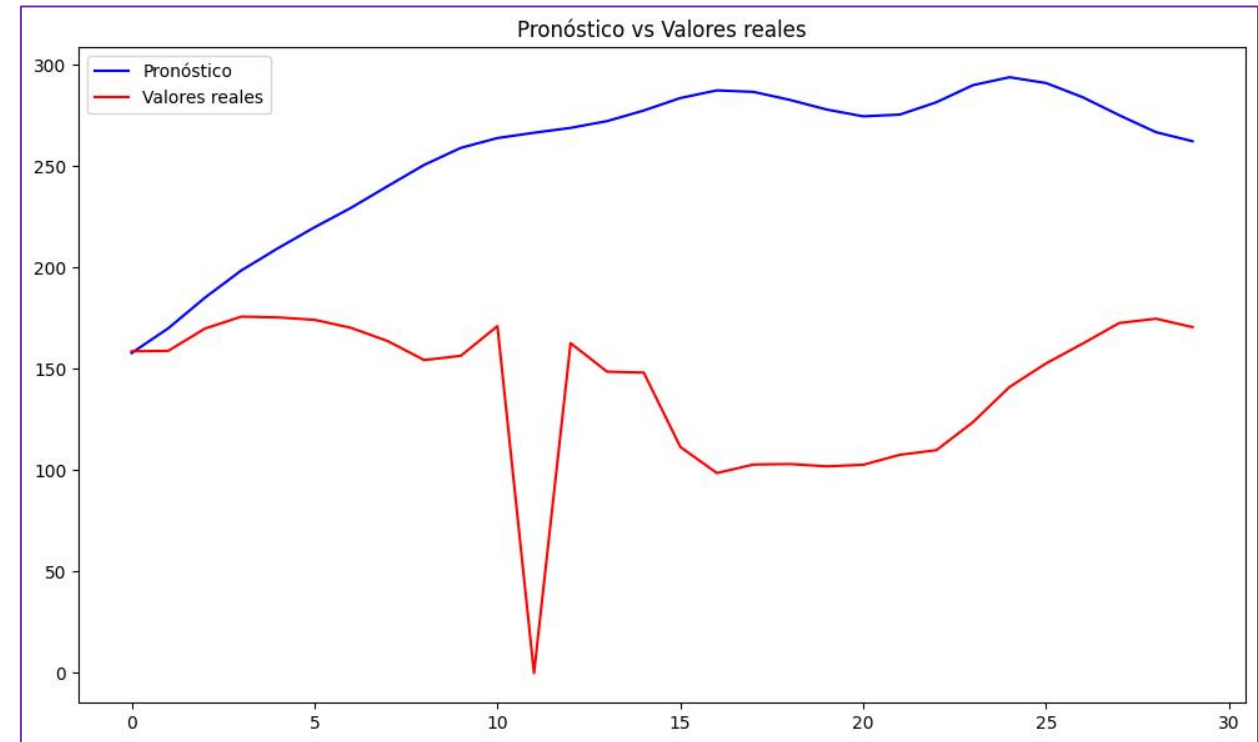
MAE: 66.14

MSE: 7647.85

# Modelo 1 - RNN



Utilizando model.predict()

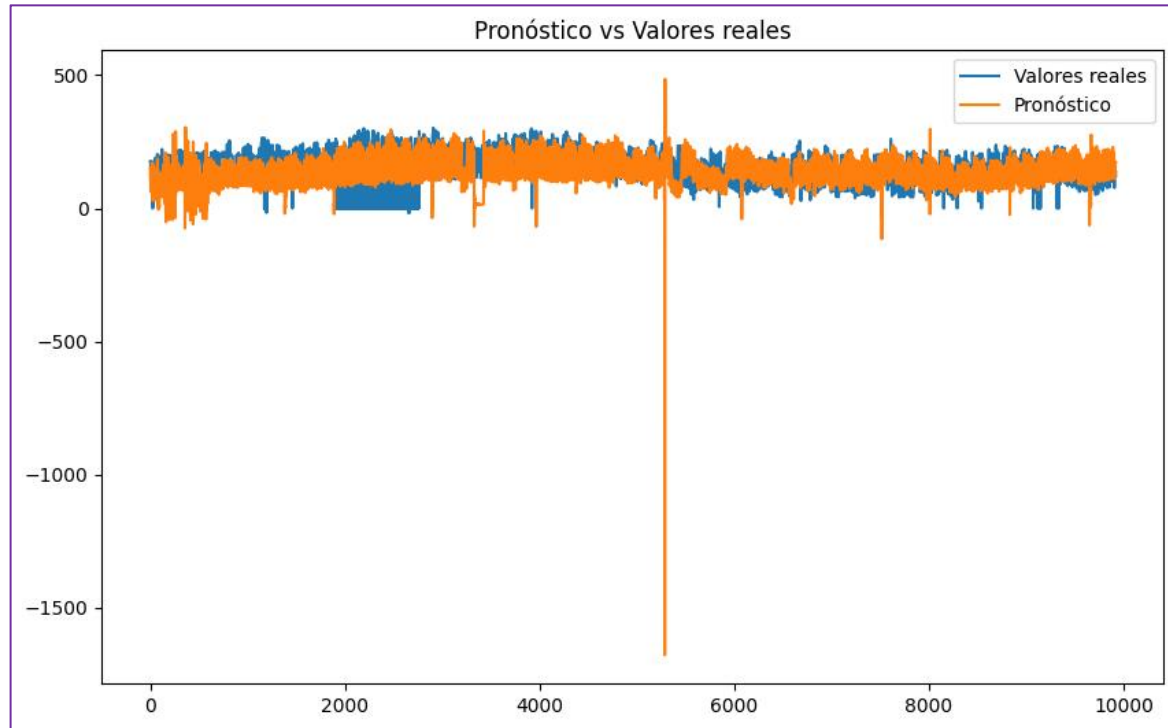


Time Series Forecasting

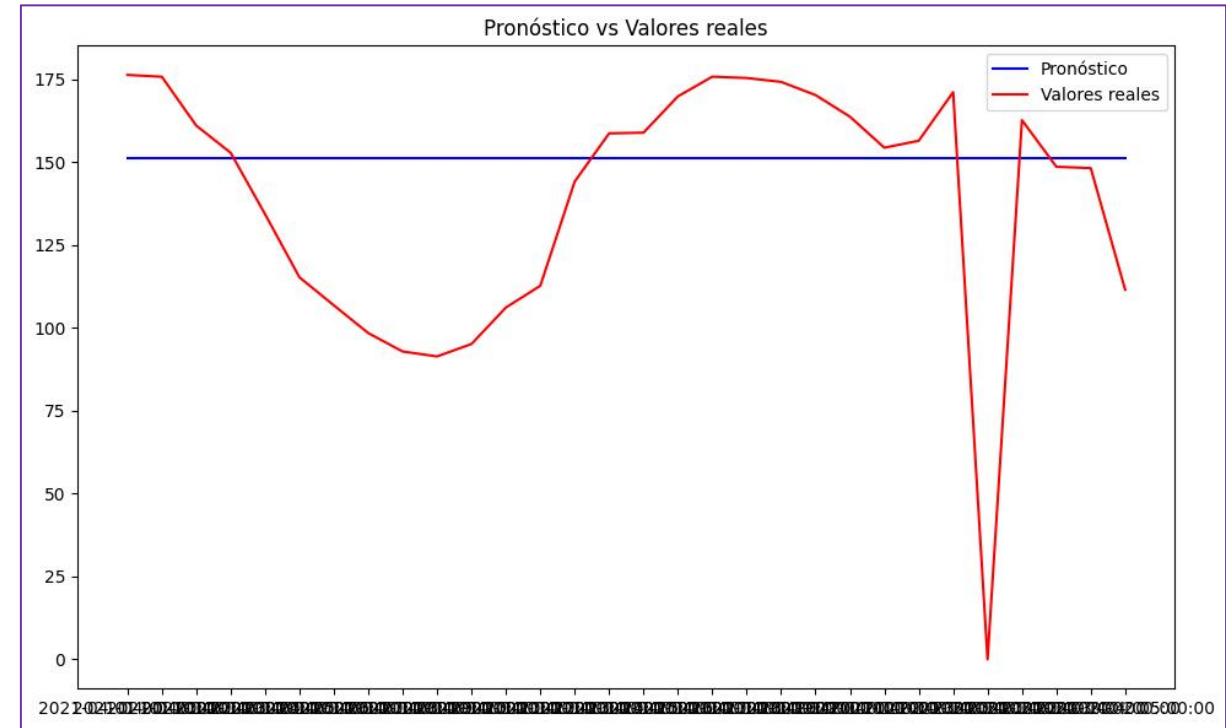
MAE: 57.16

MSE: 6307.35

# Modelo 2 - ARIMA



Utilizando `model.predict()`



Time Series Forecasting