Project 1
Machine Learning

Santiago, Chile
April 2024

Authors:

- **Felipe Gutiérrez**
- **Santiago Salvador**

Index

1. Introduction

In Chile, the Coordinador Eléctrico Nacional (CEN) is the entity in charge of managing the energy distribution regarding the national energy market (Sauma, 2018), this means, it performs a constant monitoring of the supply and demand of electricity in order to avoid blackouts or severe energy fluctuations throughout the country. In order to meet the energy demand objectives, a study of the predicted demand for the day is carried out based on the data collected up to the previous day, this data contains information on each hour of electricity consumption per meter in the different substations. On the other hand, when the demand is very high and a station cannot cover it completely, the complementary services market (SSCC) is activated as explained by (Coordinador Eléctrico Nacional, 2023), which distributes electricity from the reserve substations in order to supply the station that needs energy in these critical moments.

In this version, we aim to perform a statistical analysis of the data collected from the CEN where our study dataset contains information from the year 2017 to the year 2022. Based on the conclusions obtained from a previous analysis, our final objective is to propose a machine learning model, which can automatically anticipate the following periods and predict which parameters should be adjusted to meet the demand, and thus not generate cases of electricity shortages in certain substations.

This report will focus on the following key stages:

1. **Data quality:** An analysis of data completeness will be made and techniques will be applied to identify any null values or inconsistencies in the data set in addition to performing statistical analyses to determine the significant variables that will allow us to address our problem, we will also identify outliers and perform their appropriate treatments so that they do not influence our final results.

2. **Proposed selection of features and training sets:** At this point we will decide which variables are finally selected due to their relevance to the problem.

3. **Design of learning models:** Two different learning models will be designed to solve the problem, which will be trained and tested to see which of them is more suitable to solve the problem.

2. Description of the dataset

The initial data set consists of the following characteristics: a unique identifier for each input, the date and time of the measurement, the measurements recorded at the meters of each busbar, the data quality associated with each measurement, the number of samples taken per hour and the name of the switch associated with each busbar in the power system. This raw data is in the following form:

	0	1	2	3	4				5
0	177760211	2017-03-01 00:00:00-0300	-26.497207	1	4	AJAHUEL	110	H1	P
1	395102211	2017-03-01 00:00:00-0300	-148.774202	1	4	BUIN	110	HT1	MTP
2	395684211	2017-03-01 00:00:00-0300	82.685848	1	4	BUIN	110	H2	P
3	397670211	2017-03-01 00:00:00-0300	21.756566	1	4	LOSALME	110	H1	P
4	397693211	2017-03-01 00:00:00-0300	70.715223	1	4	LOSALME	110	H3	P
5	397712211	2017-03-01 00:00:00-0300	75.399084	1	4	LOSALME	110	H4	P
6	400203211	2017-03-01 00:00:00-0300	-6.712749	1	4	LOSALME	110	H2	P
7	433388211	2017-03-01 00:00:00-0300	139.467896	1	4	CHENA	110	H1	P
8	393655211	2017-03-01 00:00:00-0300	65.281774	1	4	BUIN	110	H1	P
9	433398211	2017-03-01 00:00:00-0300	111.997172	1	4	CHENA	110	H2	P

The challenge we wish to solve is determined by the measurements made by each switch busbar in the different substations. The total consumption per time slot (the measurements are made every one hour) then corresponds to the sum of all the measurements of the switches per substation, therefore, for the variable called "nombre_interruptor" we separate each of the subdata it has in the same string into independent variables, in this way we arrive at the following data set:

fecha	subestacion	barra	calidad_senal	muestras_hora	cantidad_interruptores	consumo
2017-03-01 03:00:00+00:00	AJAHUEL	110	1	4	4	121.980790
2017-03-01 03:00:00+00:00	BUIN	110	1	4	5	-1.150823
2017-03-01 03:00:00+00:00	CHENA	110	1	4	5	173.716561
2017-03-01 03:00:00+00:00	CNAVIA	110	1	4	9	292.470971
2017-03-01 03:00:00+00:00	ELSALTO	110	1	4	4	433.411947
2017-03-01 03:00:00+00:00	FLORIDA	110	1	4	8	31.084535
2017-03-01 03:00:00+00:00	LOSALME	110	1	4	4	161.158124
2017-03-01 04:00:00+00:00	AJAHUEL	110	1	4	4	101.512763
2017-03-01 04:00:00+00:00	BUIN	110	1	4	5	-1.101619
2017-03-01 04:00:00+00:00	CHENA	110	1	4	5	157.942591

Where:

- **consumo:** is the amount of MWs going out or coming in through the associated switch.
- **calidad_senal:** where 1 is a enabled switch and 0 disabled
- **fecha:** date where the measurement was performed (this is described in ISO 8601[1] format), it is used as an index in our dataset.
- **subestacion:** name of the substation where the measurement was taken.
- **muestras_hora:** number of samples taken within the hour of measurement.

[1] It is a standard date format used to represent dates and where the corresponding UTC time zone is also indicated.

- **cantidad_interruptores:** number of switches related to a busbar of the substation
- **barra:** name of the bus to which the switches are associated.

This final dataset is the one we will work with for the rest of our study, as it contains the information necessary for the explainability of the data.

3. Data quality study

This item sets out the statistical analysis performed on our dataset using different metrics, a 95% confidence interval (Δ=0.05) will be used to determine whether a variable is significant.

3.1 Missing data

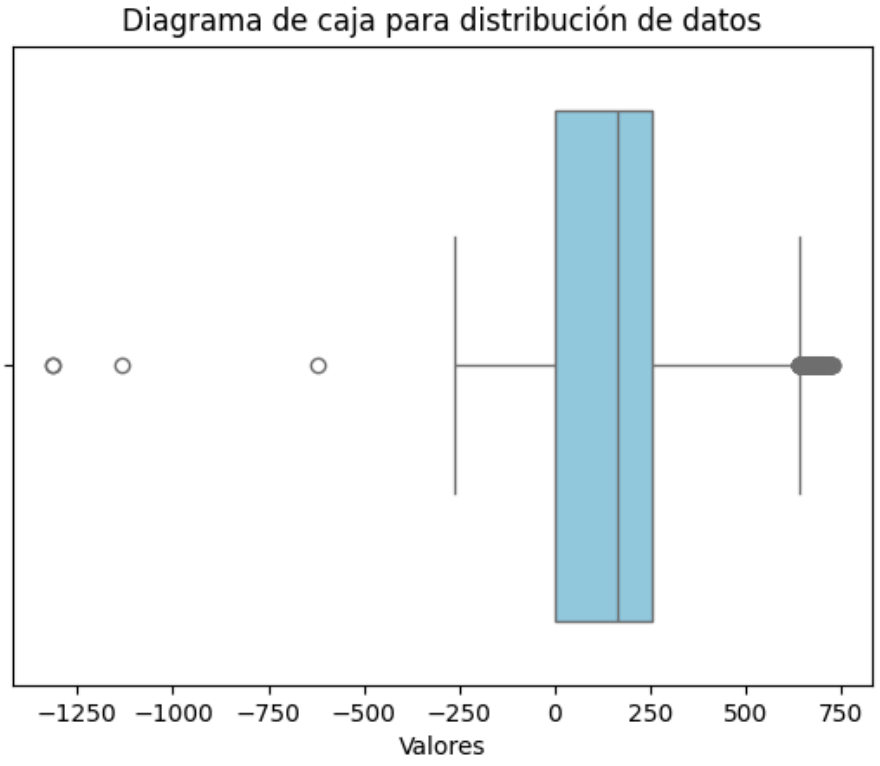
The dataset is checked for null values, as data integrity may generate outliers if not treated correctly:

```
numero de datos faltantes:
fecha                    0
subestacion              0
barra                    0
calidad_senal            0
muestras_hora            0
cantidad_interruptores   0
consumo                  0
dtype: int64
```

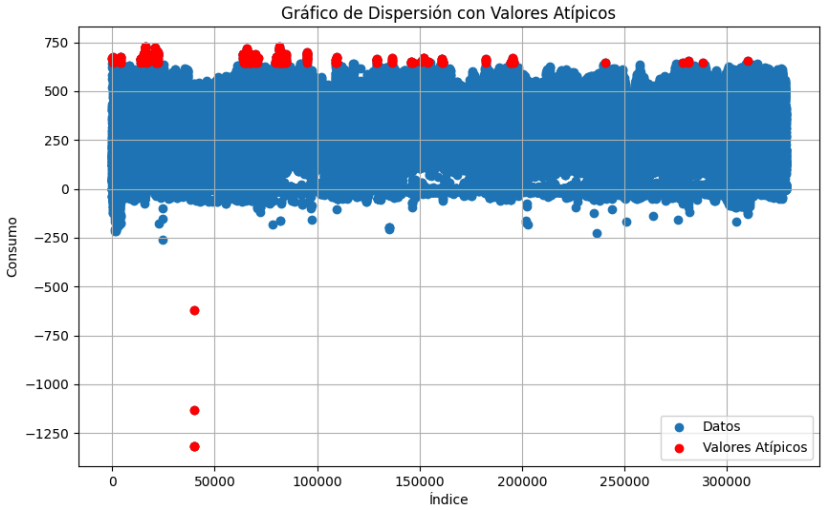
As there are no missing values, the data are left unchanged.

3.2 Outliers

To determine the number of outliers present in our dataset we made a box plot of the data distribution, which is depicted below:



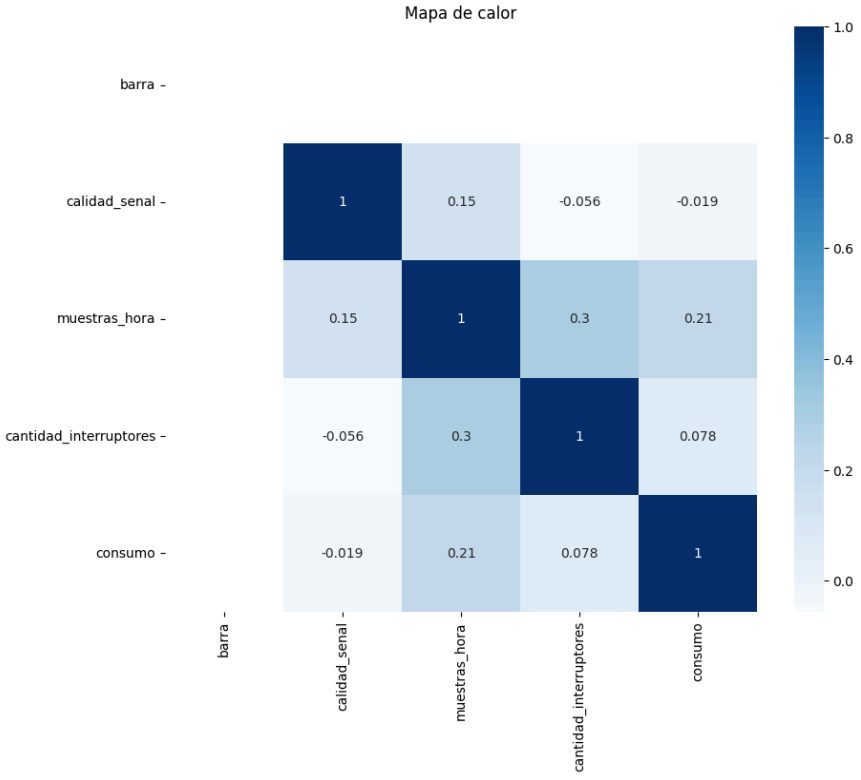
The box plot, shown above, gives us a visual idea of the distribution of the data for the variable 'consumption' and where it can be seen that there is an appreciable amount of outliers concentrated at the lower limit. Furthermore, we will use the **"z_score method"** with a threshold of 3 and for this indicator, we make a scatter plot in order to visualize these outliers:



We can see that the vast majority of consumption-related outliers (red colored points) are close to 750 MW and below -500 MW.

3.3 Descriptive data statistics

To contrast the above metrics we performed a heat map using the correlation matrix related to the numerical data, resulting in the following:



We can appreciate the following:

- The variable **“barra”** is not relevant for our study, as it always has the same value throughout the dataset and can therefore be discarded.
- The variable **“muestras_hora”** remains constant throughout our dataset with the same value, so it is not relevant for the study..
- The variable **“calidad_senal”** is related to hourly samples and can only take two possible values {0,1} as when the signal quality is inactive it means that no samples will be taken, otherwise this happens. But as “muestras_hora” is not relevant, this variable also has no direct relationship with the data as it has a correlation lower than delta.
- The variable **“numero_interruptores”** is related to “muestras_hora” and “calidad_senal” but for the variable “consumo” it has a lower delta than the accepted one, so it is not relevant for the study either.

The correlation matrix, therefore, indicates that the only variables that are significant for the study are “consumo” and “fecha”, which makes us appreciate that the problem we are facing is a time series prediction problem, which we will analyze below.

3.4 Timeseries analysis

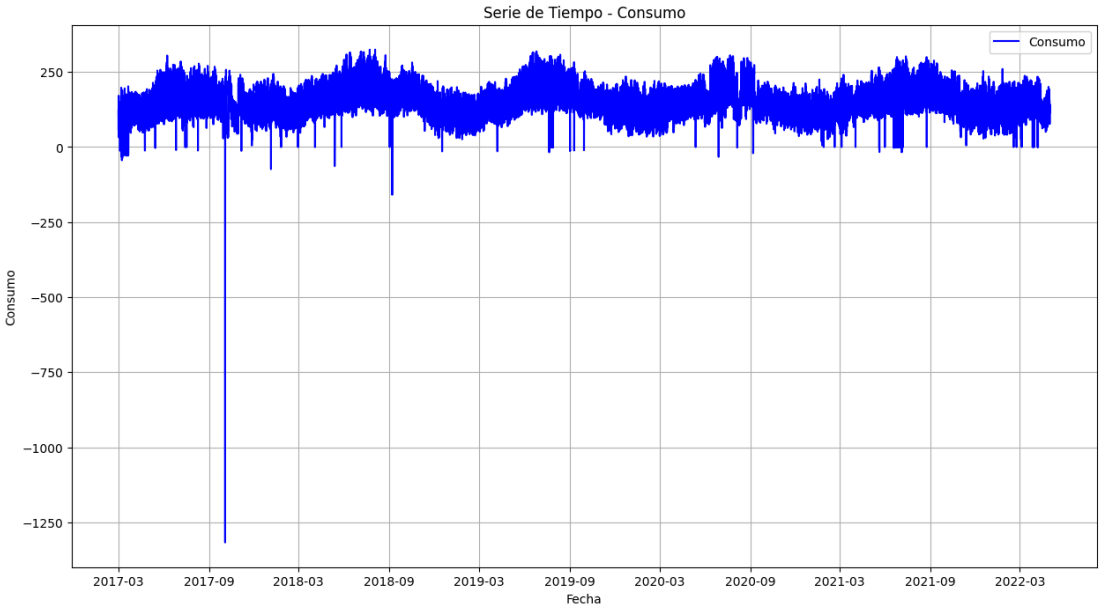
To carry out this part of the analysis, we analyzed each of the time series corresponding to each of the substations in our dataset, these are:

- AJAHUEL
- BUIN
- CHENA
- CNAVIA
- ELSALTO
- FLORIDA
- LOSALME

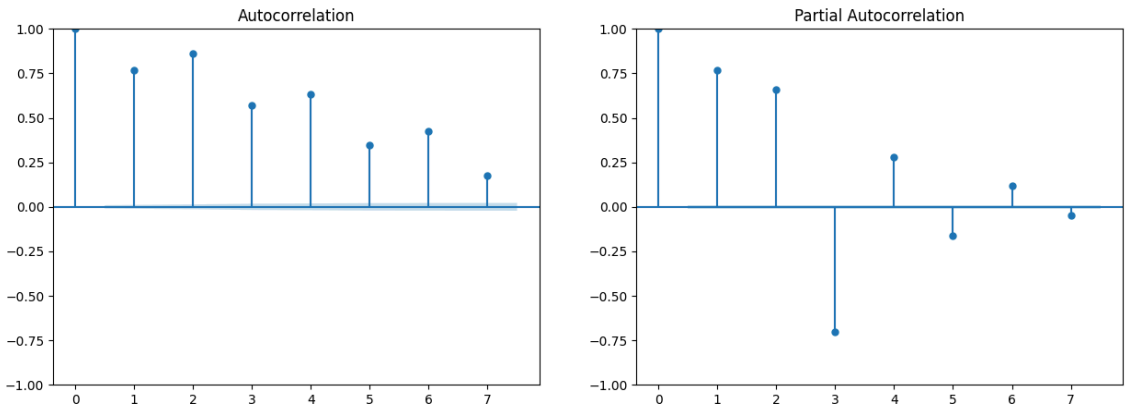
For which the analysis is shown below:

3.4.1 AJAHUEL Substation

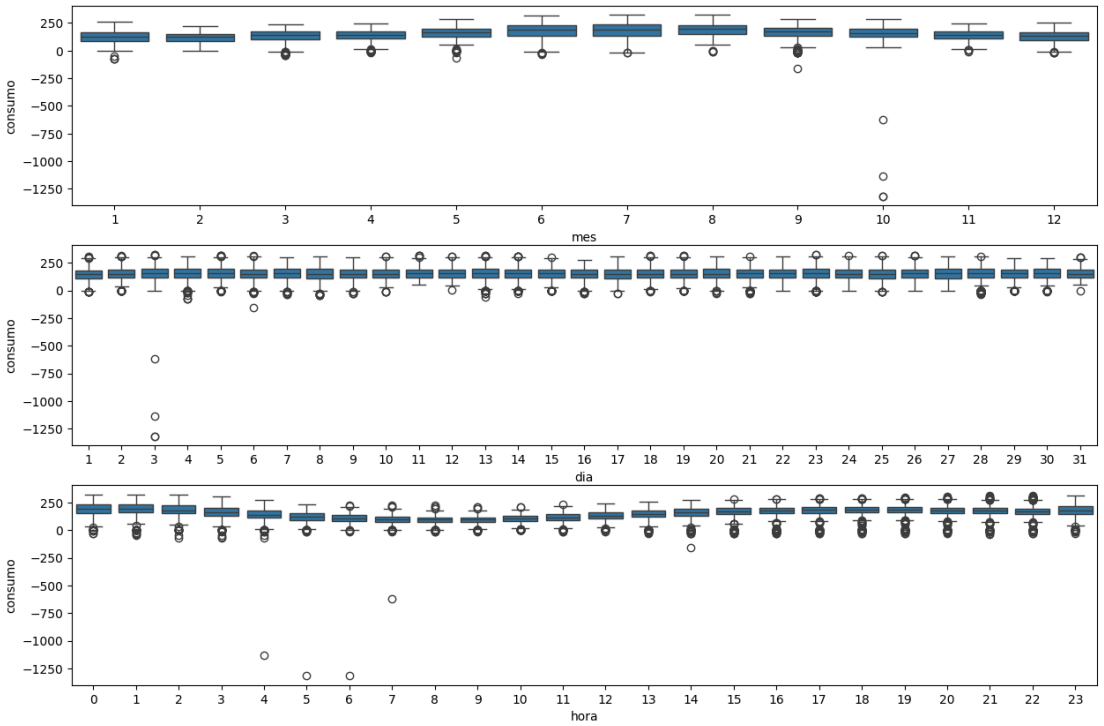
This item shows the consumption versus time data for the “AJAHUEL” substation, which are represented in the following graph:



We can appreciate that this time series has a stationary waveform, except for some outliers present in the interval of [2017-09, 2018-03] that do not follow the pattern. In addition, the autocorrelation and partial autocorrelation plots were performed, obtaining the following:



We can appreciate that the autocorrelation has the behavior almost of a linear function, since it goes up and down in some points but always maintaining its values on the same side of the axis. In addition to this, we made some box plots that allow us to analyze the evolution of this time series, with respect to the time parameters, subdivided into three graphs; month, day and hour, this graph is as follows:



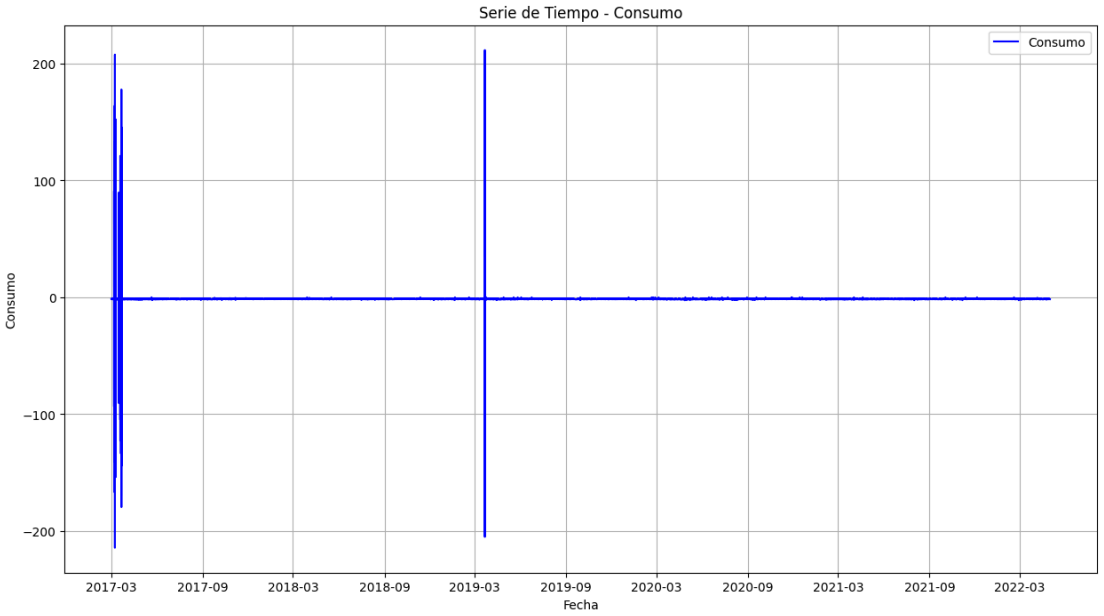
The box plots also indicate that there is a constant trend over the months and days, but for the hours there is a drop in consumption between the interval [3-15] hours, remaining constant throughout the day. According to (Cuellar, 2021) performing an Augmented Dickey-Fuller (ADF) test is a relevant metric to determine statistically whether a time series is stationary or not, and this test gave us the following result:

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

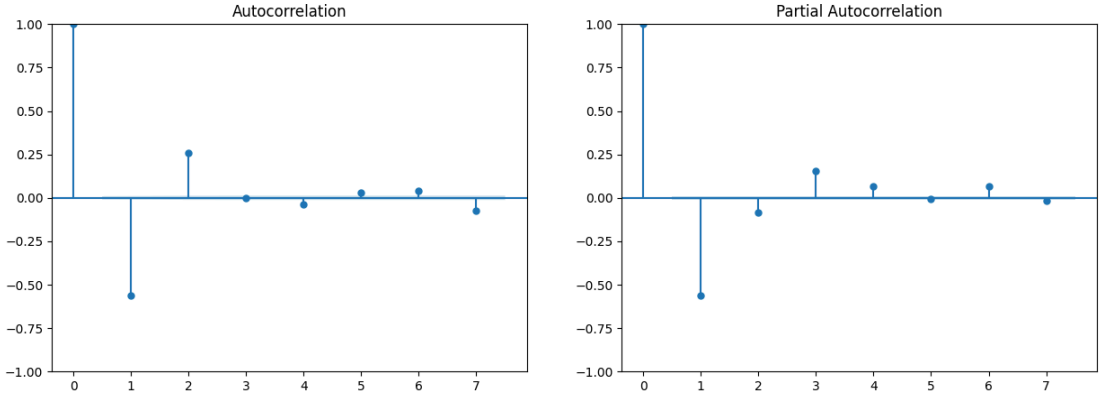
As the p-value obtained is lower than our delta, it is confirmed that the “AJAHUEL” substation has a stationary series behavior..

3.4.2 BUIN Substation

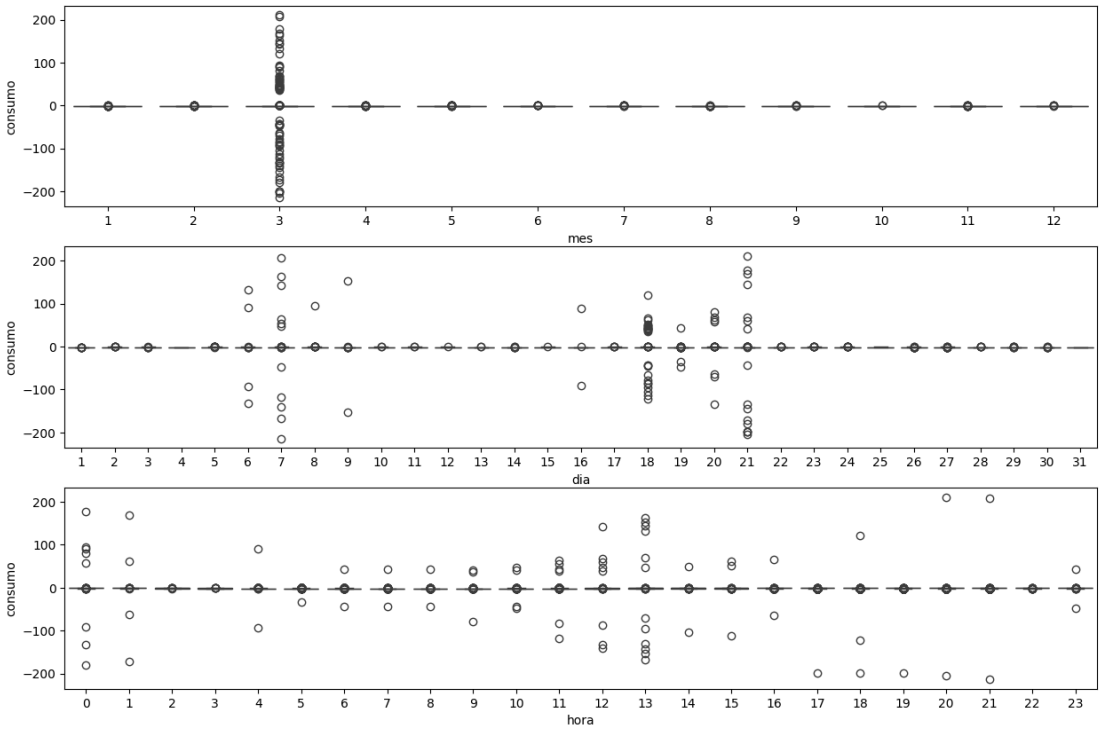
This item shows the consumption versus time data for the “BUIN” substation, which are represented in the following graph:



In this substation what is shown is that the consumption values are extremely small and close to zero, except in some peaks present in [2017-03] and [2019-03], which is where most of the information is concentrated. As these graphs are insufficient, the information is complemented with the autocorrelation and partial autocorrelation graphs of the model, obtaining the following:



We can see that the autocorrelation and partial autocorrelation are similar, but do not follow any pattern, which would indicate that some differentiation should be made for this data set in order to convert it into a periodic function. In addition to this we made some box plots showing the trend of the data over time, where the following graph was generated:



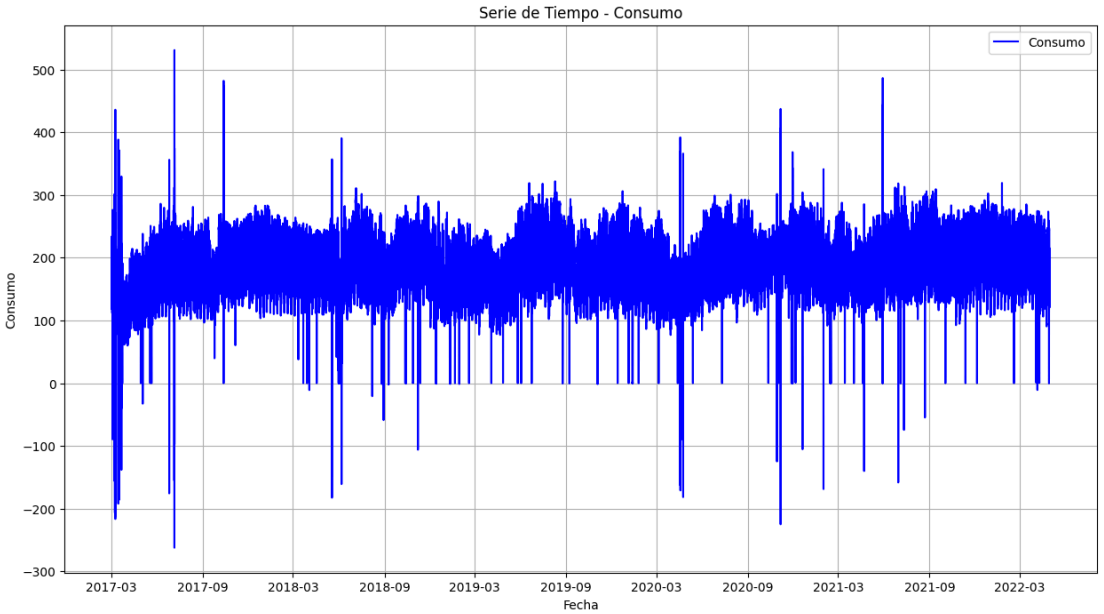
It can be seen that for this substation there are a large number of outliers, which could clarify why the graph of Time versus Consumption looked quite empty. Finally, we performed the ADF Test to statistically determine whether the above series is stationary or not, which gave us the following result:

```
ADF Statistic: -27.756376
p-value: 0.000000
Critical Values:
        1%: -3.430
        5%: -2.862
        10%: -2.567
```

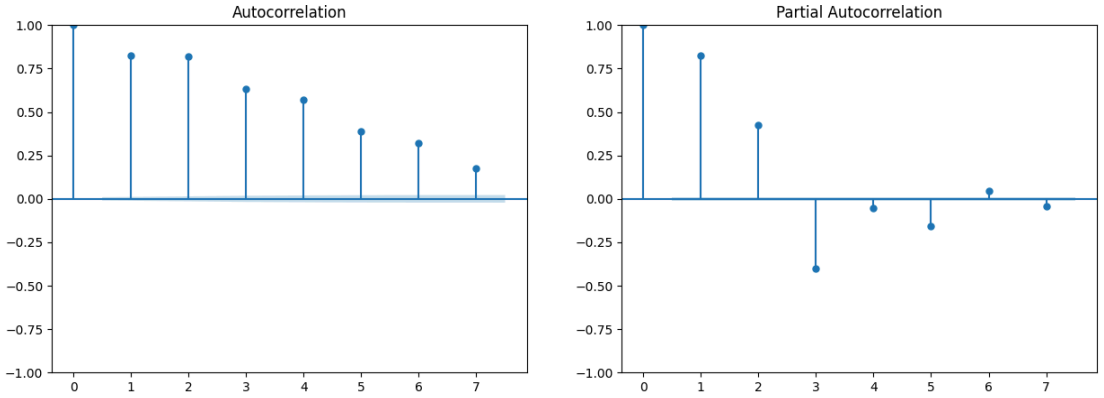
Although in this case the p-value is lower than our delta, a differentiation is still recommended for this time series due to the small amount of data present.

3.4.3 CHENA Substation

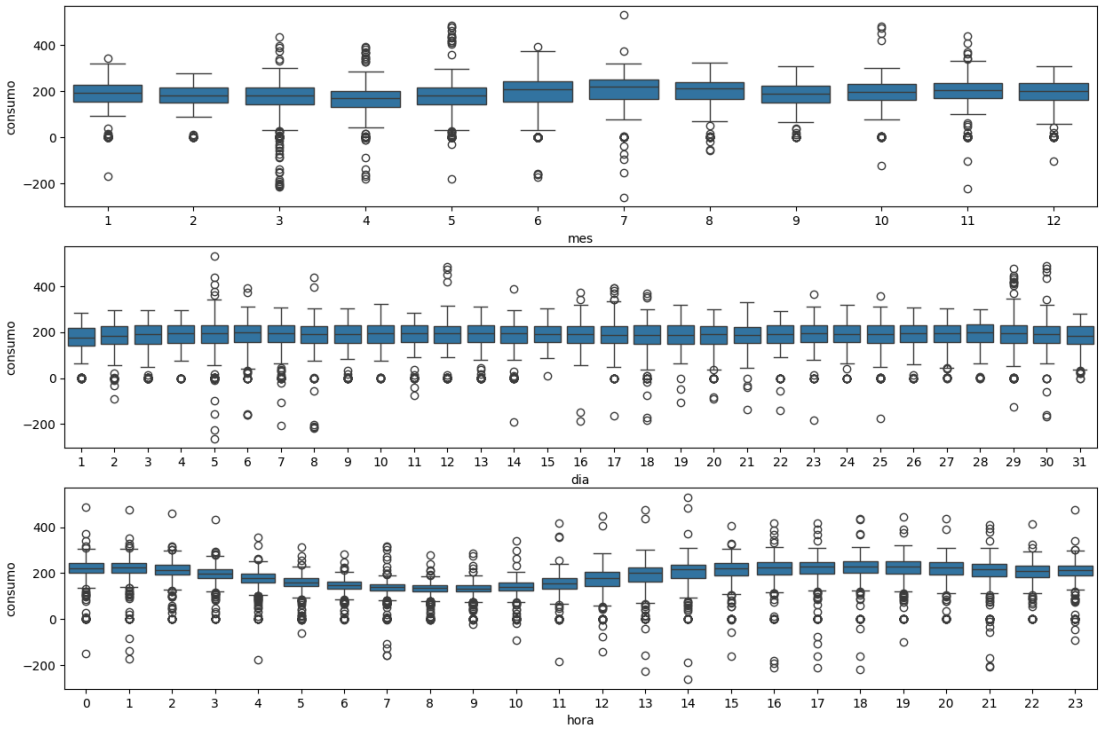
This item shows the consumption versus time data for the “CHENA” substation, which are represented in the following graph:



We can see that this time series generally follows the shape of a stationary wave with the exception of some peaks present over time. In addition, the autocorrelation and partial autocorrelation plots were performed and the following results were obtained:



We can see that the autocorrelation indicates a linear function behavior. Also to complement the study of the data for this substation, the graphs of the substations over time were also made as shown:



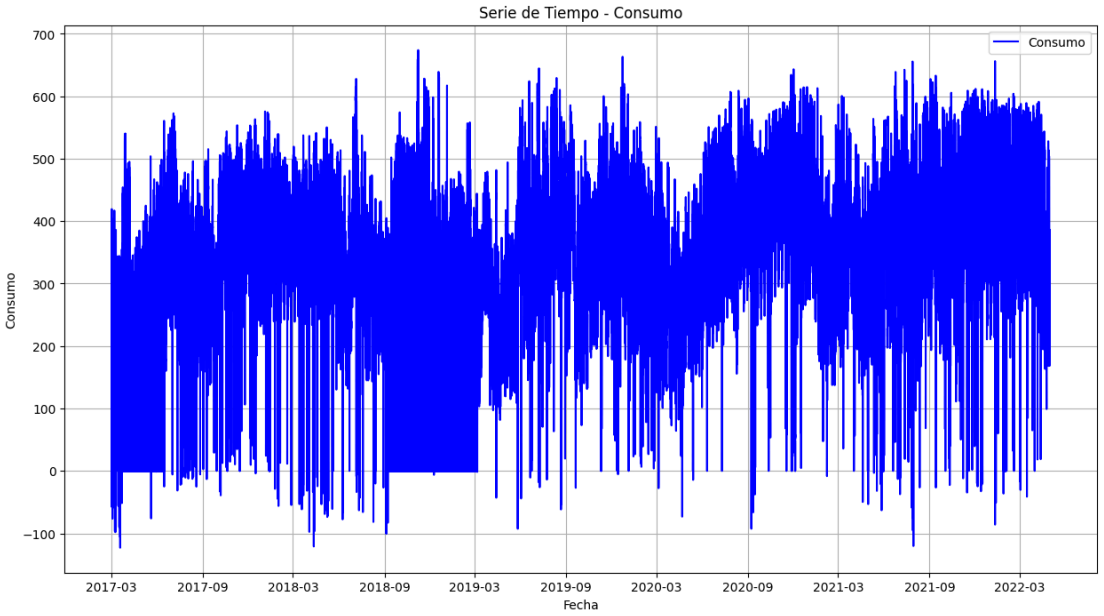
The box plots also indicate that there is a constant trend over the months and days, although for the hours there is a drop in consumption between the interval [3-12] hours, remaining constant throughout the day. Finally, we performed the ADF test to determine statistically whether the previous series is stationary or not, which gave us the following result:

```
ADF Statistic: -19.028391
p-value: 0.000000
Critical Values:
        1%: -3.430
        5%: -2.862
        10%: -2.567
```

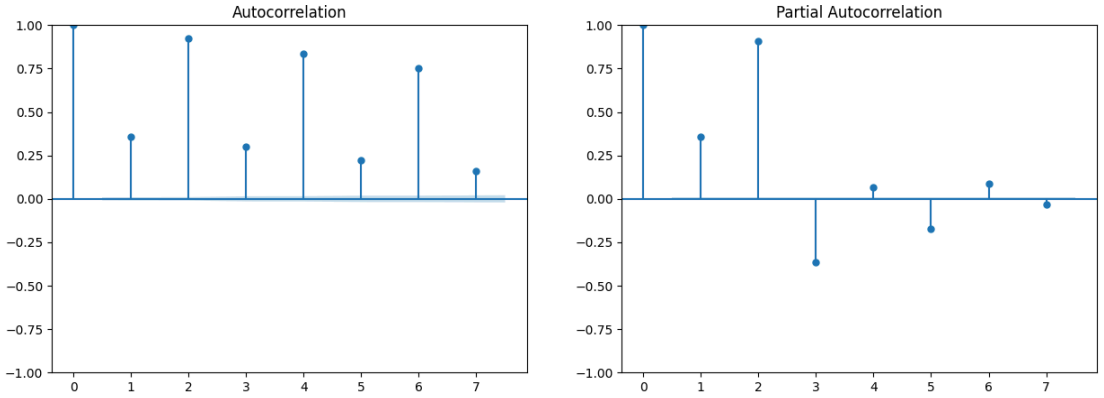
As the p-value obtained is lower than our delta, it is confirmed that the “CHENA” substation has a stationary series behavior.

3.4.4 CNAVIA Substation

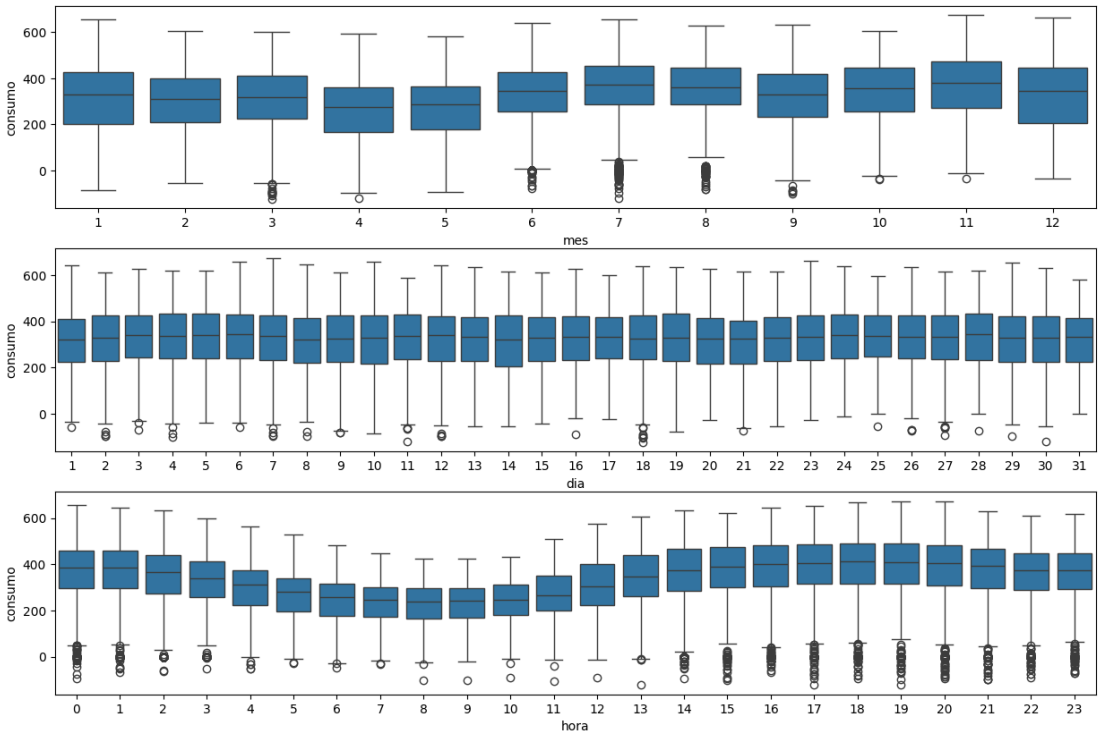
This item shows the consumption versus time data for the “CNAVIA” substation, which are represented in the following graph:



We can appreciate that this time series has a form of noise as it seems not to have a very significant pattern to interpret these data, therefore the autocorrelation and partial autocorrelation plots were made, obtaining the following:



The autocorrelation strongly indicates that this series has the behaviour of a stationary series over time. In addition to this, we make some box plots that allow us to analyse the evolution of this time series:



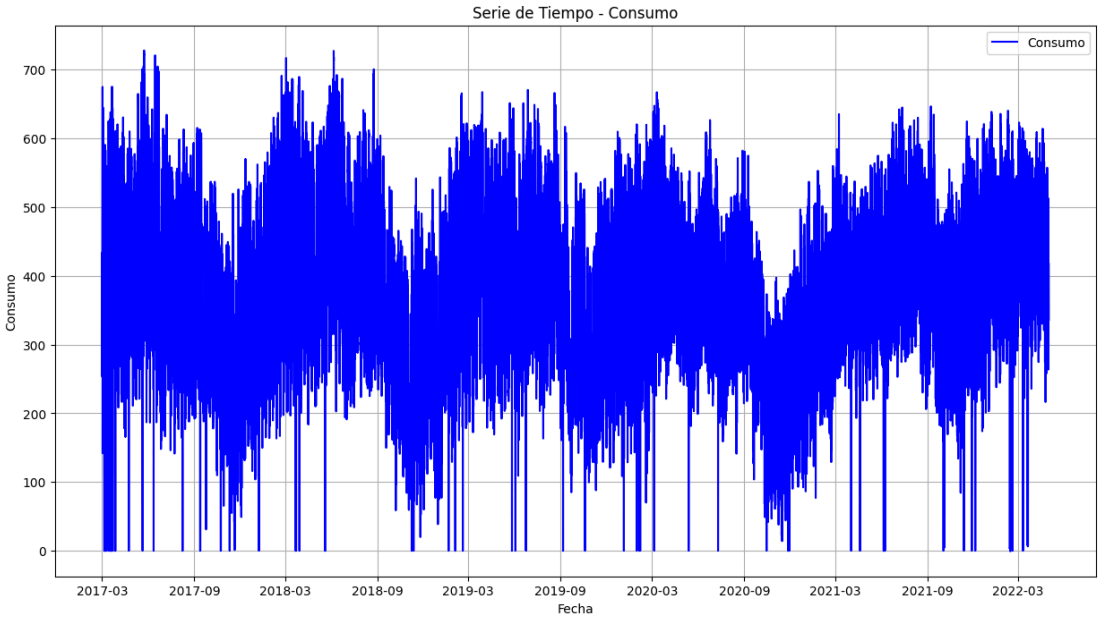
The box plots also indicate that there is a constant trend over the months and days, but for the hours there is a drop in consumption between the interval [3-15] hours, remaining constant throughout the day. Finally, we performed the ADF Test to determine statistically whether the above series is stationary or not, which gave us the following result:

```
ADF Statistic: -9.322051
p-value: 0.000000
Critical Values:
        1%: -3.430
        5%: -2.862
        10%: -2.567
```

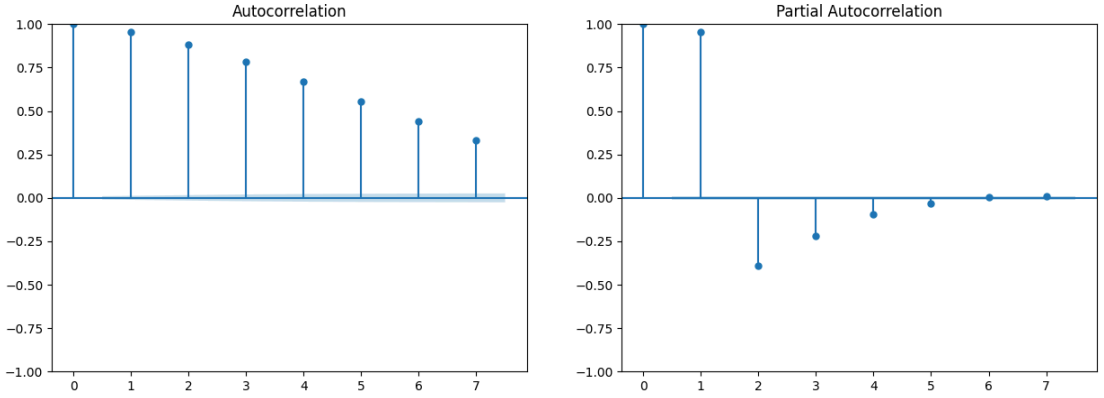
As the p-value obtained is lower than our delta, it is confirmed that the "CNAVIA" substation has a stationary series behavior.

3.4.5 ELSALTO Substation

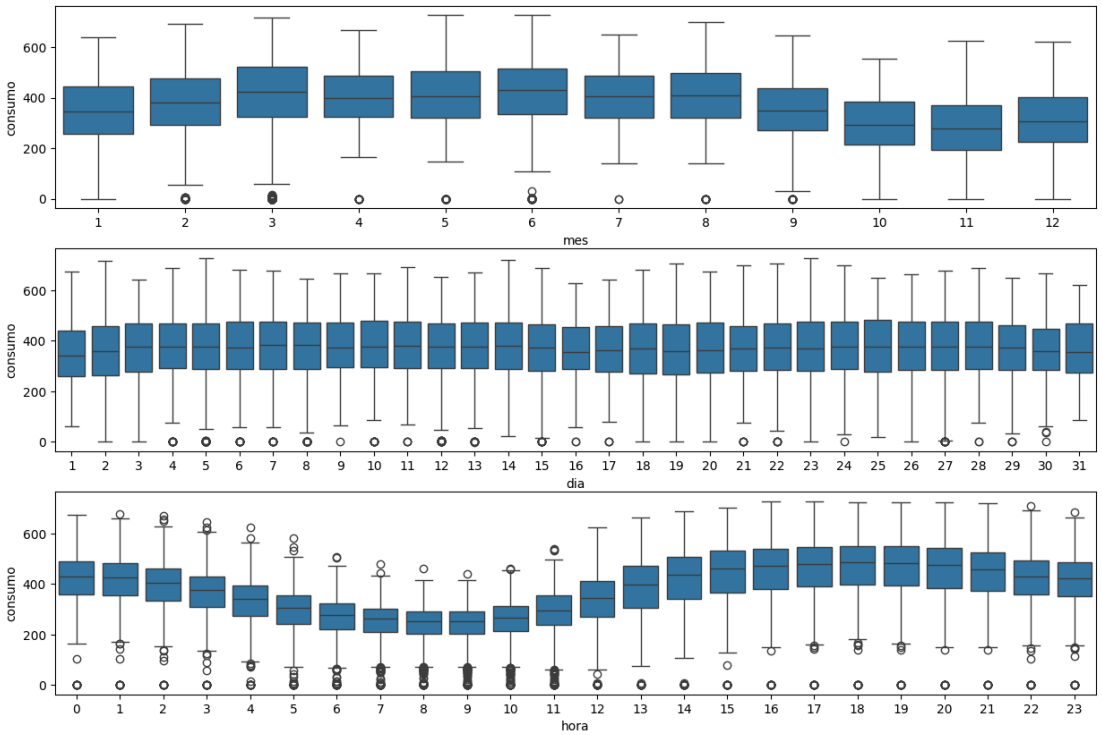
This item shows the consumption versus time data for the "ELSALTO" substation, which are represented in the following graph:



We can see that this time series has in general a stationary waveform although it has several outliers over time, so we performed the autocorrelation and partial autocorrelation plots to determine the shape of the data, obtaining the following:



We can see that the autocorrelation indicates the behavior of a linear function for this substation, as a complement to this we make the box plots of the evolution of this time series:



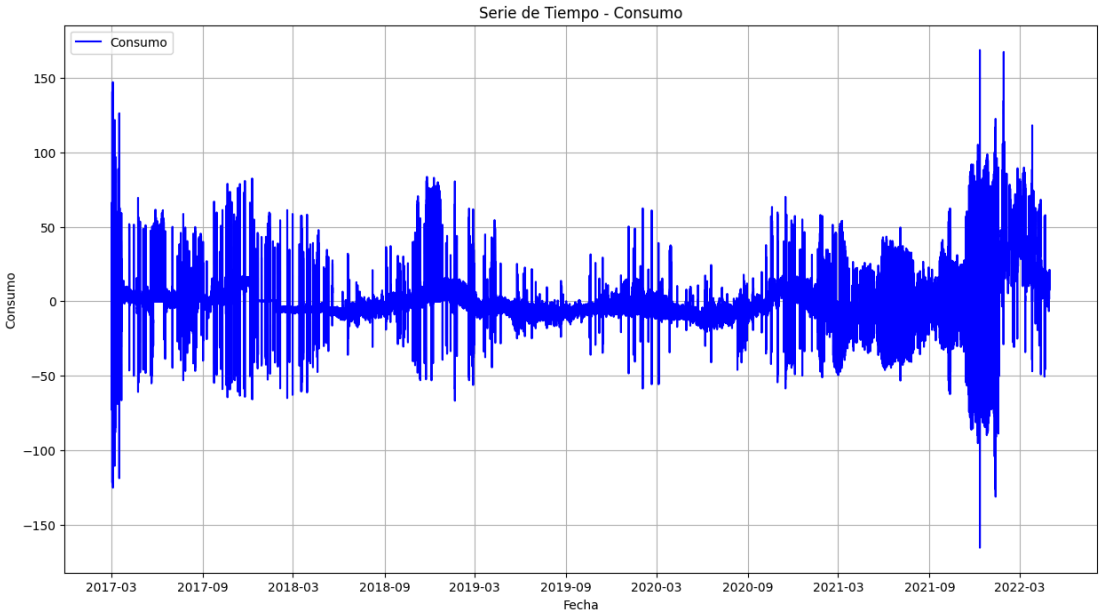
The box plots indicate that there is a constant trend over the months and days, but for the hours there is a drop in consumption between the interval [3-12] hours, remaining constant throughout the day. Finally, we performed the ADF Test to determine statistically whether the above series is stationary or not, which gave us the following result:

```
ADF Statistic: -16.868375
p-value: 0.000000
Critical Values:
        1%: -3.430
        5%: -2.862
        10%: -2.567
```

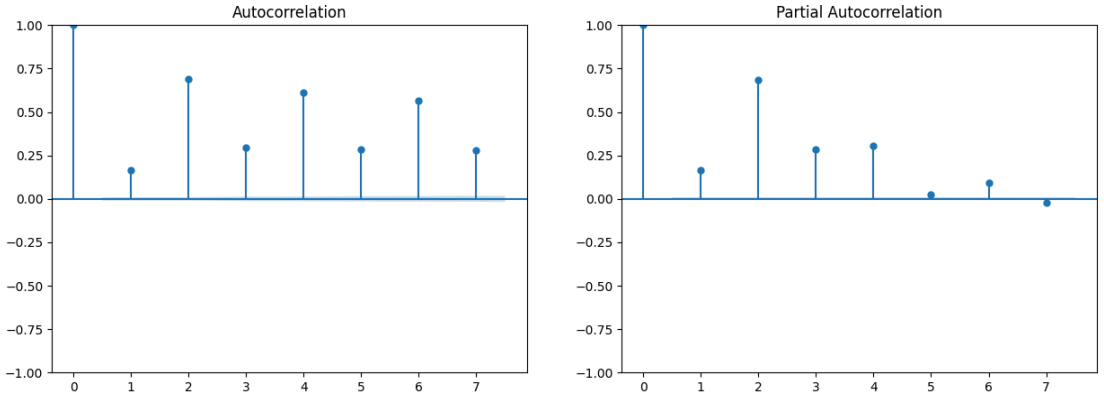
As the p-value obtained is lower than our delta, it is confirmed that the “ELSALTO” substation has a stationary series behavior.

3.4.6 FLORIDA Substation

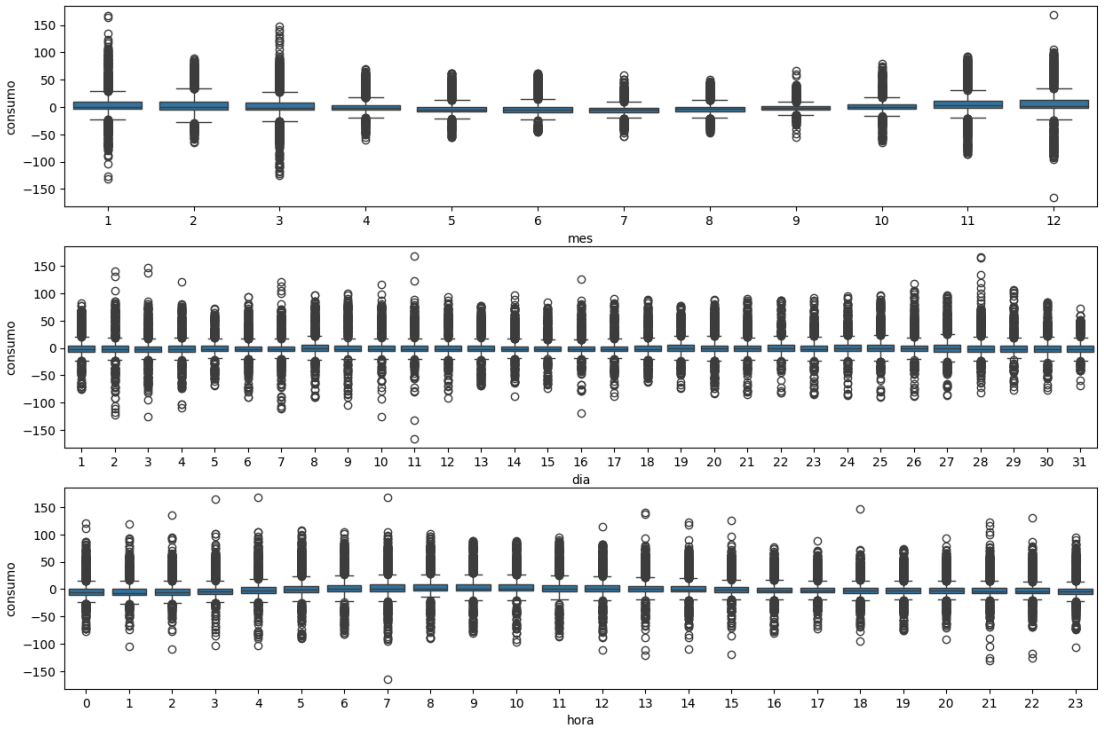
This item shows the consumption versus time data for the “FLORIDA” substation, which are represented in the following graph:



We can see that this time series has a rather particular form of noise, which is why autocorrelation and partial autocorrelation plots were made to determine the type of data we are dealing with:



We can see that the autocorrelation and partial autocorrelation are similar, confirming that this series is stationary over time. In addition to this, we make some box plots that allow us to visualize the evolution of consumption over time:



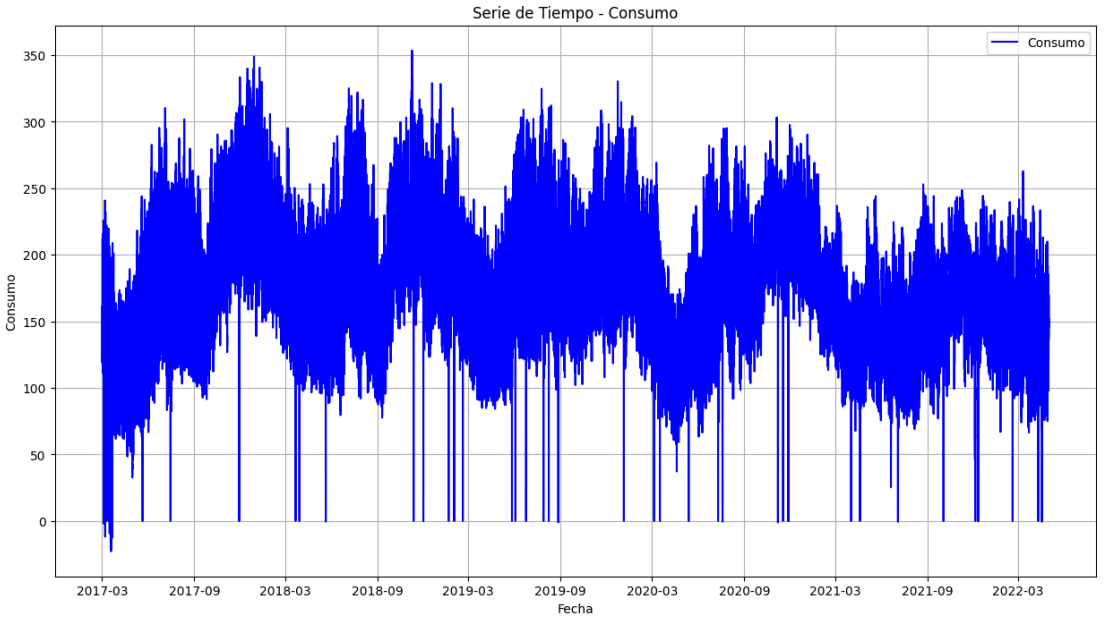
The box plots also show us that there are a significant number of outliers so we cannot perform a general study for these data and therefore it remains to perform the ADF Test to determine statistically whether the above series is stationary or not, which gave us the following result:

```
ADF Statistic: -7.998049
p-value: 0.000000
Critical Values:
        1%: -3.430
        5%: -2.862
        10%: -2.567
```

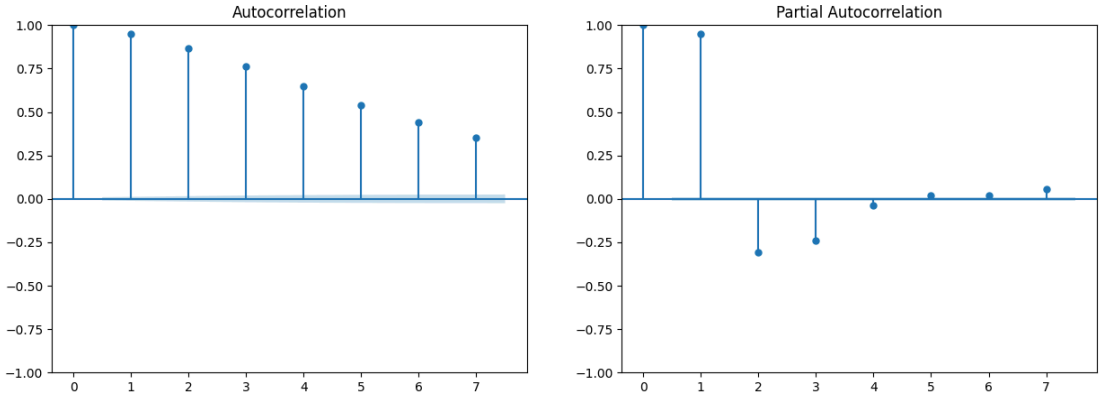
As the p-value obtained is lower than our delta, it is confirmed that the “FLORIDA” substation has a stationary series behavior.

3.4.7 LOSALME Substation

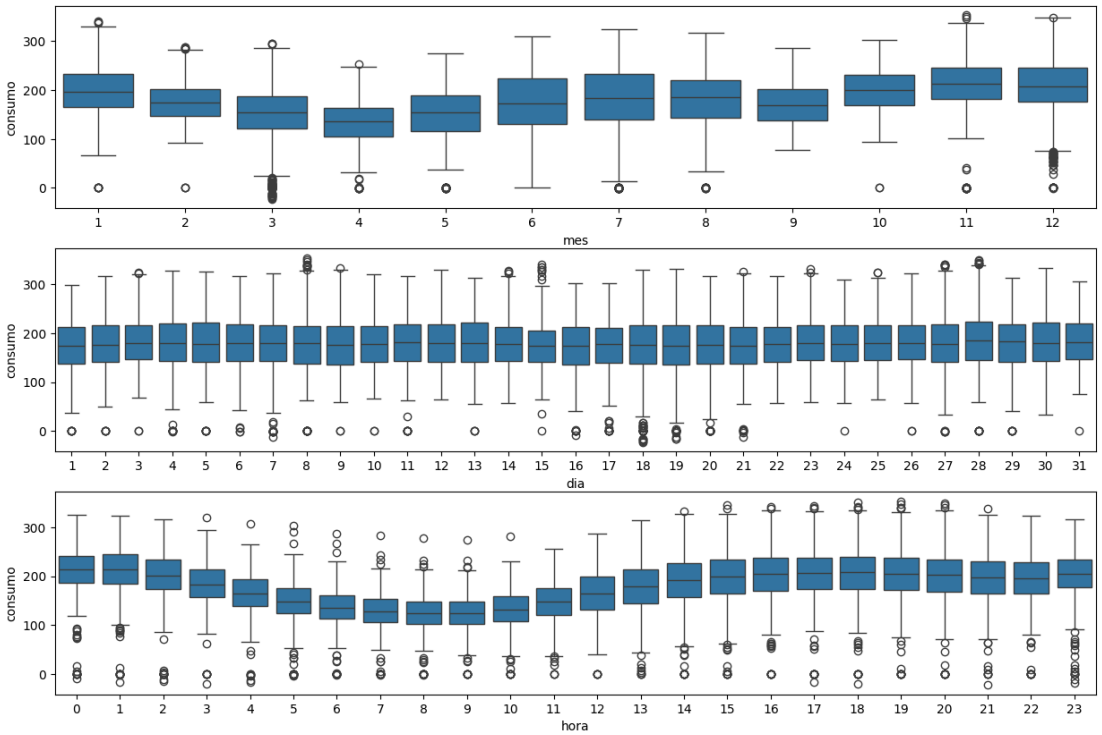
This item shows the consumption versus time data for the “LOSALME” substation, which are represented in the following graph:



We can appreciate that this time series has in general a stationary waveform with some outliers that are triggered in certain periods of time, therefore, it was necessary to perform the autocorrelation and partial autocorrelation plots to obtain more information about the shape of the data, obtaining the following:



We can see that the autocorrelation strongly indicates that the data follow the behavior of a linear function over time. In addition to this, we produced box plots that illustrate the evolution of consumption over time:



The box plots also indicate that there is a constant trend over the months and days, but for the hours there is a drop in consumption between the interval [3-12] hours, remaining constant throughout the day. Finally, we performed the ADF Test to determine statistically whether the above series is stationary or not, which gave us the following result:

```
ADF Statistic: -11.337786
p-value: 0.000000
Critical Values:
        1%: -3.430
        5%: -2.862
        10%: -2.567
```

As the p-value obtained is lower than our delta, it is confirmed that the “LOSALME” substation has a stationary series behavior.

4. Proposed selection of features and training sets

For the present dataset, the features selected to design our learning models are "fecha" and "consumo" including the name of the substation to which each measurement belongs.

Furthermore, we define the training and test sets from the following time periods[2]:

- **Training set:** These are the data that belong to the period [2017-03, 2021-03].
- **Test set**: These are the data that belong to the period [2021-04, 2022-04].

These sets will be the ones we will use to create our learning models.

[2] These periods follow the format YYYY-MM, therefore the set [A1-A2, B1-B2] starts from month A2 of year A1 and ends in month B2 of year B1.

5. Conceptual description of used Machine Learning Models

For the present study, two machine learning models will be used to solve the same problem, namely:

5.1 Model 1 - RNN

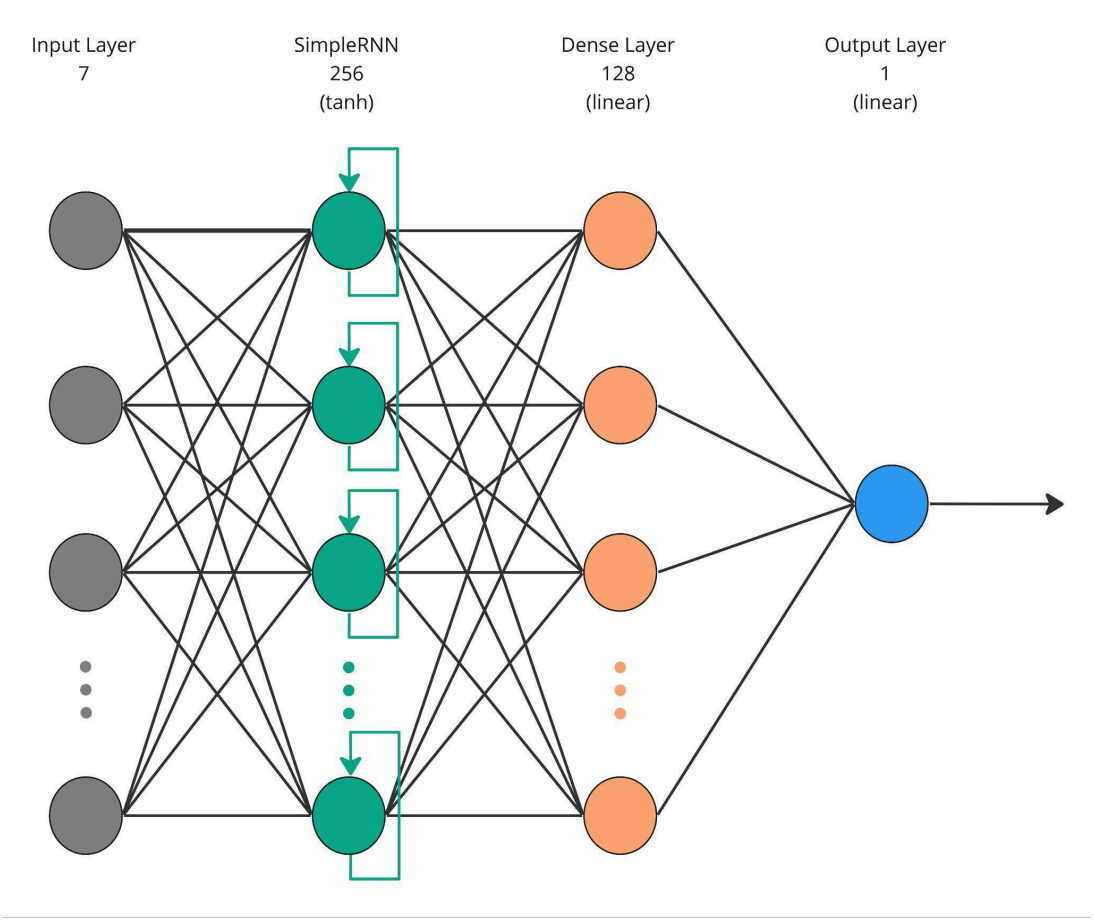
The model used is a 4-layer recurrent neural network (RNN), explained below:

- ***Input layer:*** This layer receives the set of characteristics that allow predicting the "consumo" variable and takes as a reference a 7-day window to make the predictions, therefore 7 inputs per iteration are entered.
- ***Recurrence layer:*** For this case we use SimpleRNN[3] to define a recurrence layer with 256 memory cells using the 'tanh' activation function. This layer processes the information from the previous layer and using the BPTT algorithm[4] allows the result computed for each memory cell to reduce the model error before sharing this data to the next layer.
- ***Dense layer:*** This layer has 128 neurons and processes the data using a linear activation function, because we want to predict the optimal consumption, which is variable and cannot be contextualized as a classification problem.
- ***Output layer:*** This last layer has a single output and a linear activation function was used.

Its operation can be described in the following diagram:

[3] SimpleRNN is a Keras class that allows the creation of a recurrence layer and accepts as parameters the form of the input, the amount of memory cells to be used and the activation function used.

[4] Backpropagation Through Time (BPTT) is an algorithm commonly used in neural networks to propagate the error and adjust the model with respect to the target variable at each training epoch.



5.2 Model 2 - ARIMA

The model used was ARIMA (Autoregressive Integrated Moving Average) as this statistical model is useful for predictions based on time series, the settings for this model were as follows:

- The model was trained to consider “consumo” as the variable to be predicted.
- The *autoregressive component (p)* was defined with the value suggested by the partial autocorrelation plots of each substation.
- The *integration component (d)* was defined based on whether the substation is stationary or not.
- The *moving average component (q)* was defined with the value suggested by the autocorrelation plots for each substation.

6. Procedure used for the generation of the training, validation and test set.

As mentioned in the previous item, we will work with two different models, which is why the data preparation procedure for the models varies from one model to the other:

6.1 Model 1 - RNN

First, we define the substation we are going to work with to create our learning model, in this case we show how the parameters of our RNN would be implemented for the "AJAHUEL" substation, although in order to generate the conclusions of the models we changed the name of the substation to which we will be performing the analysis:

```
# Probar con la subestacion AJAHUEL
df = df.loc[df['subestacion'] == 'AJAHUEL']
df = df.drop('subestacion', axis=1)
```

We define the training and test sets according to the time periods already proposed:

```
# 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') #
```

From the mask that we have already defined for each set, we set the sets from each time period and we also establish that the prediction window will be 7 days to carry out our study:

```
# 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:]
```

Next, we set a seed to generate our RNN data, we define the input data to be a time series which will be determined by a 7-day window to predict consumption, with respect to the other design specifications we already mentioned from our model we should arrive at something like the following:

```
seed=123456
rd.seed(seed)
np.random.seed(seed)
tf.random.set_seed(seed)

model = Sequential()
model.add(InputLayer(batch_input_shape=(None, window_size, 1), name="serie"))
model.add(SimpleRNN(256))
model.add(Dense(128))
model.add(Dense(1, activation='linear'))
model.summary()
```

When this block is executed, it generates the summary table illustrating the shape of our RNN:

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

Based on the recommendations of we use ***"tensorboard_callback"*** as a callback function to store the values of the metrics of my model and use the parameter ***"shuffle=True"*** to add variability to the data in each training epoch, contributing to improve the generalization of the model. We do this in the following way:

```
tag="rnn"+datetime.now().strftime("%Y%m%d-%H%M%S")
log_dir = "logs/fit/" + tag
tensorboard_callback = TensorBoard(log_dir=log_dir, histogram_freq=1)
rnn = model.fit(X_train, y_train, batch_size=256, epochs=200, shuffle=True, verbose=1, callbacks=[tensorboard_callback])
model.save(log_dir+'/model_{0}.keras'.format(tag))
model.save("generated/rnn.keras")
```

Finally, for the creation of our neural network we used Adam[5] with a learning rate of "1e-3" as an optimiser, this was done based on recommendations from (Aggarwal, 2018, 286) to avoid problems such as *"vanishing gradients"*. Also, the loss function was defined in relation to the MSE metric, which has similar properties to MAE for training time series:

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

6.2 Model 2 - ARIMA

As in the previous model, we define the training and test sets, defining the substation we are analyzing, to be used within the model:

[5] Adam's algorithm (Adaptive Moment Estimation) is a variant of the Stochastic Gradient Descent (SGD) optimizer, which is generally used to train neural networks because of its computational efficiency and good generalization to new data.

```
# Probar con la subestacion AJAHUEL
df = df.loc[df['subestacion'] == 'AJAHUEL']
df = df.drop('subestacion', axis=1)

# Definir conjuntos de datos de prueba y entrenamiento
df.set_index('fecha', inplace=True)

train = df.loc['2017-03-01':'2021-03-31'] # Filtrar dato
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')
```

Once we have our test and training sets assembled, we define the variable we are going to work with in the model ("consumo") and adjust the model according to the parameters already conceptually defined:

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

# guardar modelo
results.save('generated/arima.pkl')
```

Once the training of our model is finished, we save it for later use in the testing phase.

7. Metrics used to evaluate the quality of the generated models.

To evaluate the quality of the applied models, the MAPE (Mean Absolute Percentage Error) metric was selected since according to the recommendations of (Machine Learning in Plain English, 2023) this metric is robust to outliers, which means that it does not penalize errors excessively, which is extremely useful for regressive data as it happens for time series analysis. Therefore, we describe the results of the models as follows:

7.1 Model 1 Results

To perform this part of the analysis, we analyzed each of the time series corresponding to each of the substations in our dataset and performed 200 epochs training for all of them using the MSE as loss function and MAE, MSE and MAPE as analysis metrics. The analysis performed for each of the substations is shown below:

7.1.1 AJAHUEL Substation

This item shows the training results for the "AJAHUEL" substation, where the results obtained in the first 10 epochs are shown here:

```
Epoch 1/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 3s 9ms/step - loss: 8807.2637 - mae: 67.6286 - mse: 8807.2637
Epoch 2/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 477.7117 - mae: 9.6849 - mse: 477.7117
Epoch 3/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 374.1333 - mae: 8.3151 - mse: 374.1333
Epoch 4/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 336.3947 - mae: 7.7299 - mse: 336.3947
Epoch 5/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 2s 13ms/step - loss: 317.4271 - mae: 7.4951 - mse: 317.4271
Epoch 6/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 305.6710 - mae: 7.3556 - mse: 305.6710
Epoch 7/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 2s 11ms/step - loss: 295.0911 - mae: 7.1865 - mse: 295.0911
Epoch 8/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - loss: 290.5732 - mae: 7.0913 - mse: 290.5732
Epoch 9/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - loss: 289.6692 - mae: 7.0118 - mse: 289.6692
Epoch 10/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - loss: 285.6490 - mae: 7.0052 - mse: 285.6490
```

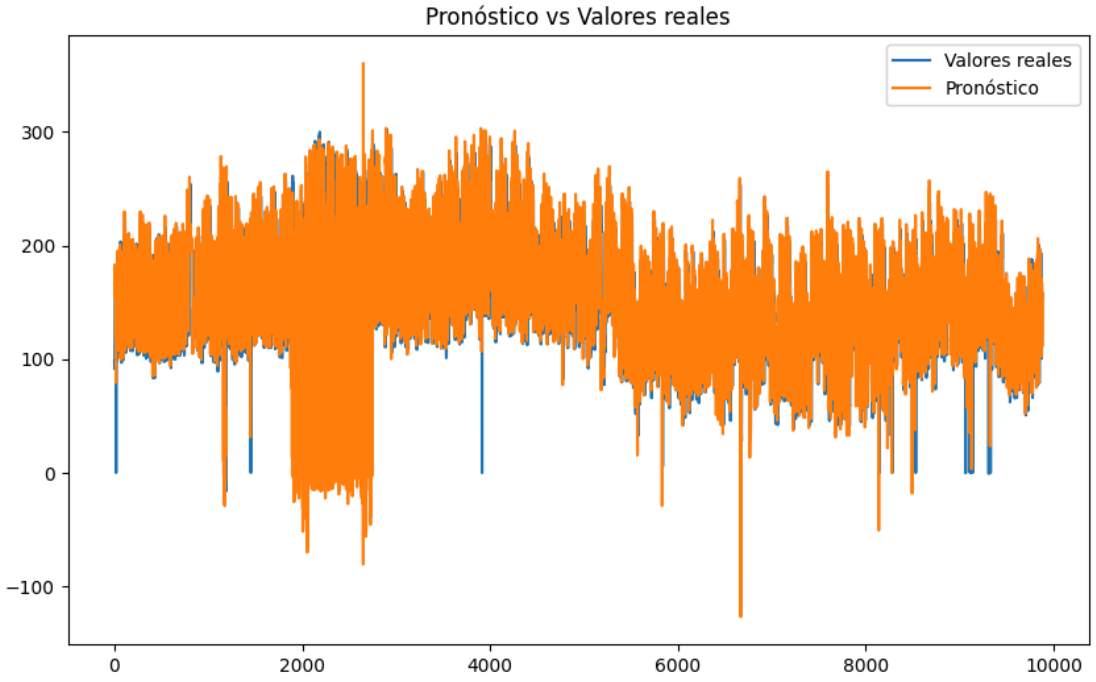
And the results obtained in the last 2 epochs:

```
Epoch 199/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 2s 11ms/step - loss: 158.8646 - mae: 6.9304 - mse: 158.8646
Epoch 200/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 2s 12ms/step - loss: 164.2780 - mae: 7.0010 - mse: 164.2780
```

Finally, the evaluation of the training set with the test set is performed, obtaining the following result:

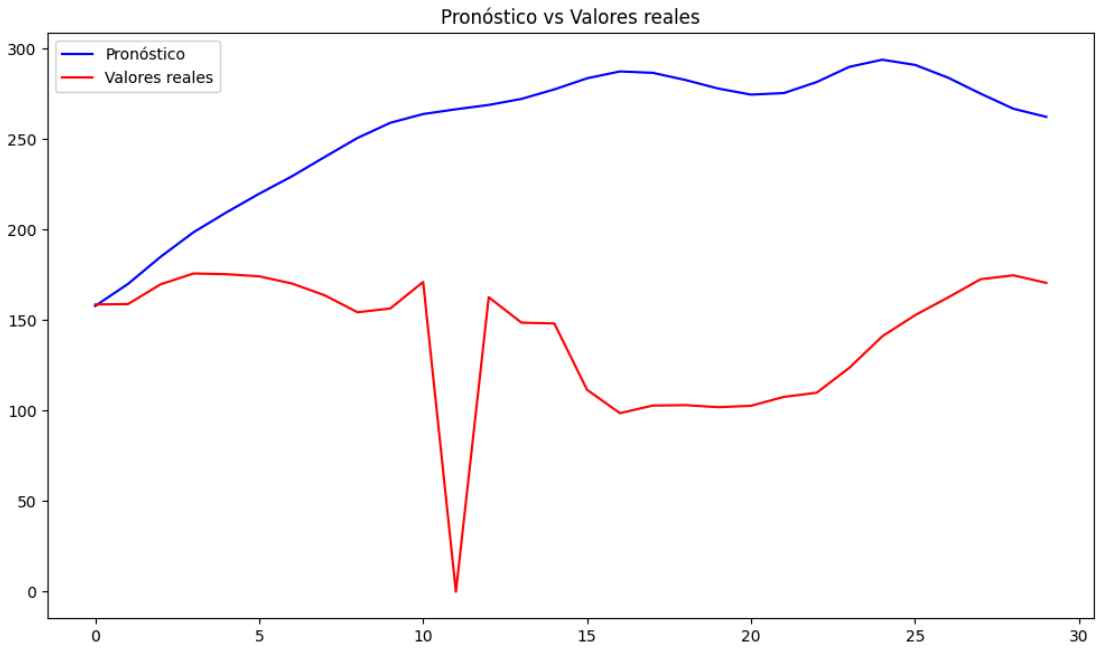
```
309/309 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step
MAE: 66.14071893371647
MSE: 7647.855354541793
MAPE: 43.31%
```

We can see that we obtained for the model predictions with respect to the test set a MAPE of 43.1%, which means that it predicted on average this percentage correctly with respect to the real data. In addition, to contrast this information, we can make a graph showing the actual data versus the correctly predicted data, as illustrated here:



We can see that our model has a tendency to predict the data quite well in general, except for some peaks shown in the graph where it generates erroneous predictions.
To contrast this information we will perform a Time Series Forecasting[6] to predict the data of our neural network using as reference the model of our RNN and using a maximum of 30 steps to predict the time series with respect to the historical, thus obtaining the following graph:

[6] Time Series Forecasting is a methodology used in time series analysis to discover patterns in the data and trends over time.



Here we can see that our model is not good at predicting time series in small amounts of data, that is why the predicted data are far away from the real values.

7.1.2 BUIN Substation

This item shows the training results for the "BUIN" substation, where the results obtained in the first 10 epochs are shown here:

```
Epoch 1/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 4s 16ms/step - loss: 17.3594 - mae: 0.7384 - mse: 17.3594
Epoch 2/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 2s 14ms/step - loss: 15.0563 - mae: 0.6620 - mse: 15.0563
Epoch 3/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 2s 12ms/step - loss: 14.9427 - mae: 0.6491 - mse: 14.9427
Epoch 4/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 2s 12ms/step - loss: 14.8537 - mae: 0.6268 - mse: 14.8537
Epoch 5/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 2s 12ms/step - loss: 14.9426 - mae: 0.6944 - mse: 14.9426
Epoch 6/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 2s 11ms/step - loss: 14.6039 - mae: 0.6365 - mse: 14.6039
Epoch 7/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 2s 11ms/step - loss: 14.0983 - mae: 0.6067 - mse: 14.0983
Epoch 8/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 2s 12ms/step - loss: 13.6893 - mae: 0.6071 - mse: 13.6893
Epoch 9/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 3s 14ms/step - loss: 13.6255 - mae: 0.5741 - mse: 13.6255
Epoch 10/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 2s 12ms/step - loss: 13.4828 - mae: 0.5500 - mse: 13.4828
```

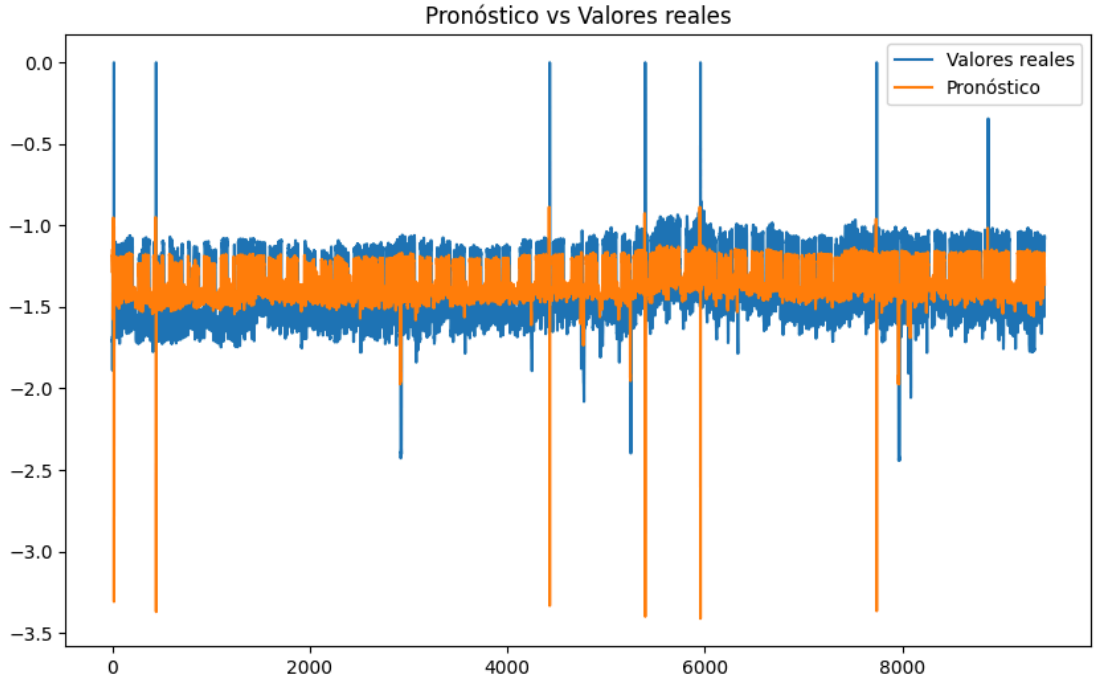
And the results obtained in the last 2 epochs:

```
Epoch 199/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 6.9747 - mae: 0.4399 - mse: 6.9747
Epoch 200/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - loss: 7.1613 - mae: 0.4172 - mse: 7.1613
```

Finally, the evaluation of the training set with the test set is performed, obtaining the following result:

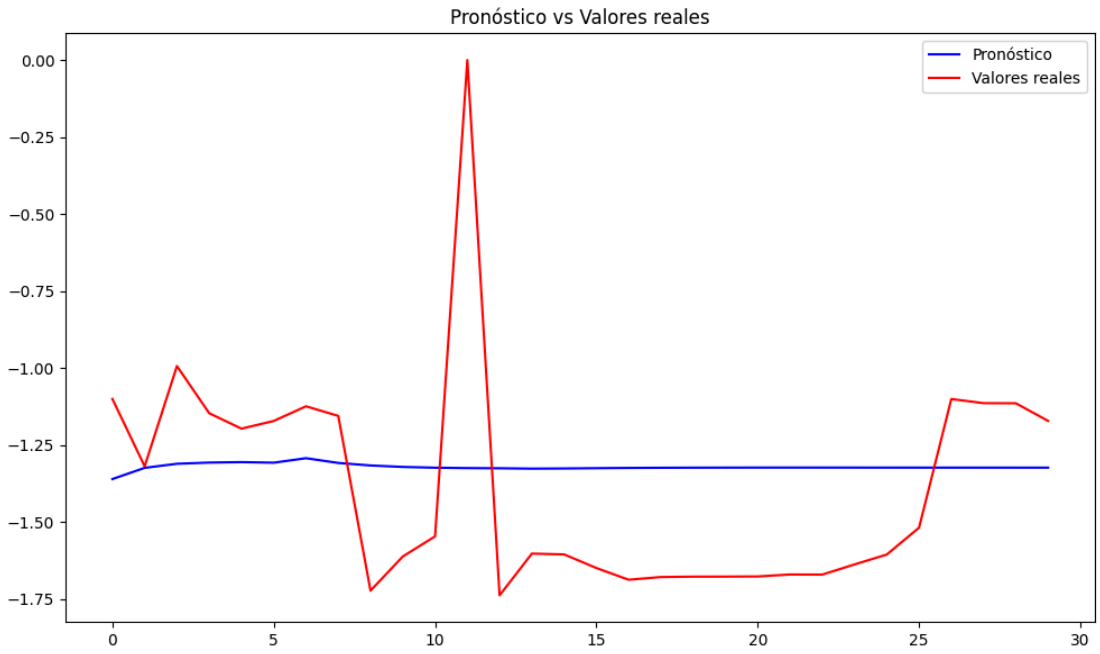
```
295/295 ━━━━━━━━━━━━━━━━━━━━ 1s 1ms/step
MAE: 0.21991606221162294
MSE: 0.07255888624679242
MAPE: 16.18%
```

We can see that we obtained for the model predictions with respect to the test set a MAPE of 16.18%, which means that it predicted on average this percentage correctly with respect to the real data. In addition, to contrast this information, we can make a graph showing the actual data versus the correctly predicted data, using the values saved with model.predict() as shown here:



We can see that our model tends to predict the data quite well in general, except for some peaks shown in the graph where it generates erroneous predictions.

To contrast this information we will perform a Time Series Forecasting using a maximum of 30 steps to predict the time series with respect to the historical, as shown here:



Here we can see that our model is not good at predicting time series in small amounts of data, although it makes a better attempt than the substation seen above.

7.1.3 CHENA Substation

This item shows the training results for the "CHENA" substation, where the results obtained in the first 10 epochs are shown here:

```
Epoch 1/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 4s 14ms/step - loss: 13336.9883 - mae: 90.9611 - mse: 13336.9883
Epoch 2/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 2s 14ms/step - loss: 563.0425 - mae: 11.5691 - mse: 563.0425
Epoch 3/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 2s 14ms/step - loss: 446.5167 - mae: 9.2182 - mse: 446.5167
Epoch 4/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 2s 15ms/step - loss: 407.6531 - mae: 8.3799 - mse: 407.6531
Epoch 5/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 2s 11ms/step - loss: 391.2427 - mae: 8.1193 - mse: 391.2427
Epoch 6/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 2s 12ms/step - loss: 372.8074 - mae: 7.9265 - mse: 372.8074
Epoch 7/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 2s 14ms/step - loss: 363.3827 - mae: 7.8137 - mse: 363.3827
Epoch 8/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 2s 13ms/step - loss: 355.3550 - mae: 7.7482 - mse: 355.3550
Epoch 9/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 3s 15ms/step - loss: 347.5765 - mae: 7.6637 - mse: 347.5765
Epoch 10/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 2s 16ms/step - loss: 342.1131 - mae: 7.6850 - mse: 342.1131
```

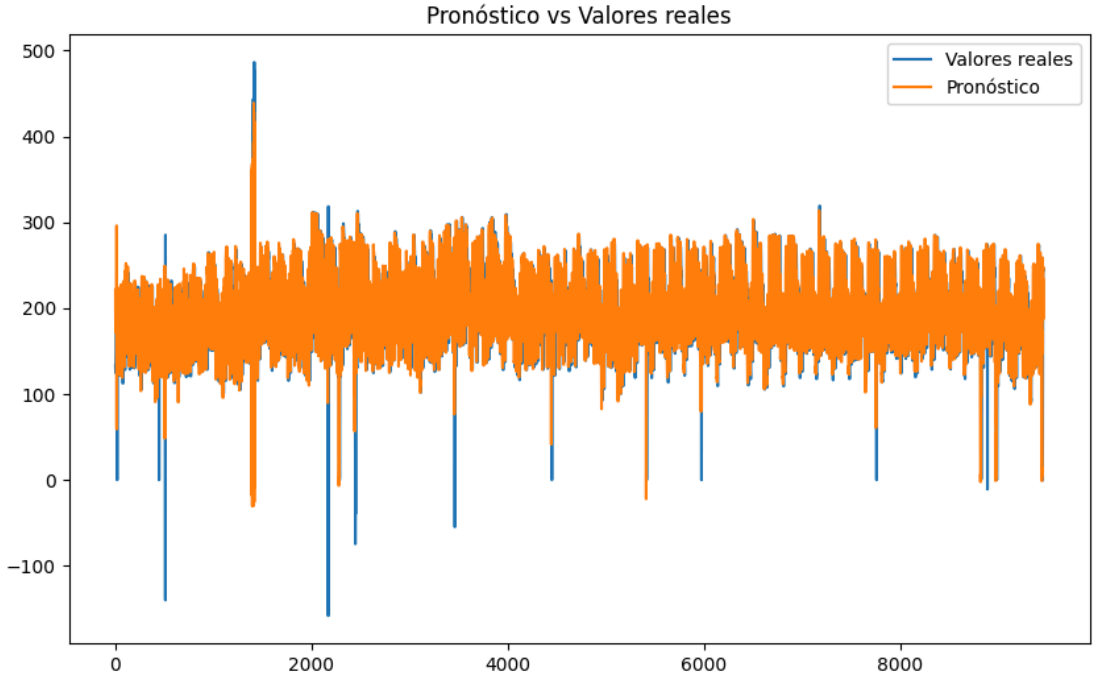
And the results obtained in the last 2 epochs:

```
Epoch 199/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 2s 17ms/step - loss: 204.4157 - mae: 6.7751 - mse: 204.4157
Epoch 200/200
141/141 ━━━━━━━━━━━━━━━━━━━━ 3s 18ms/step - loss: 210.8737 - mae: 6.8956 - mse: 210.8737
```

Finally, the evaluation of the training set with the test set is performed, obtaining the following result:

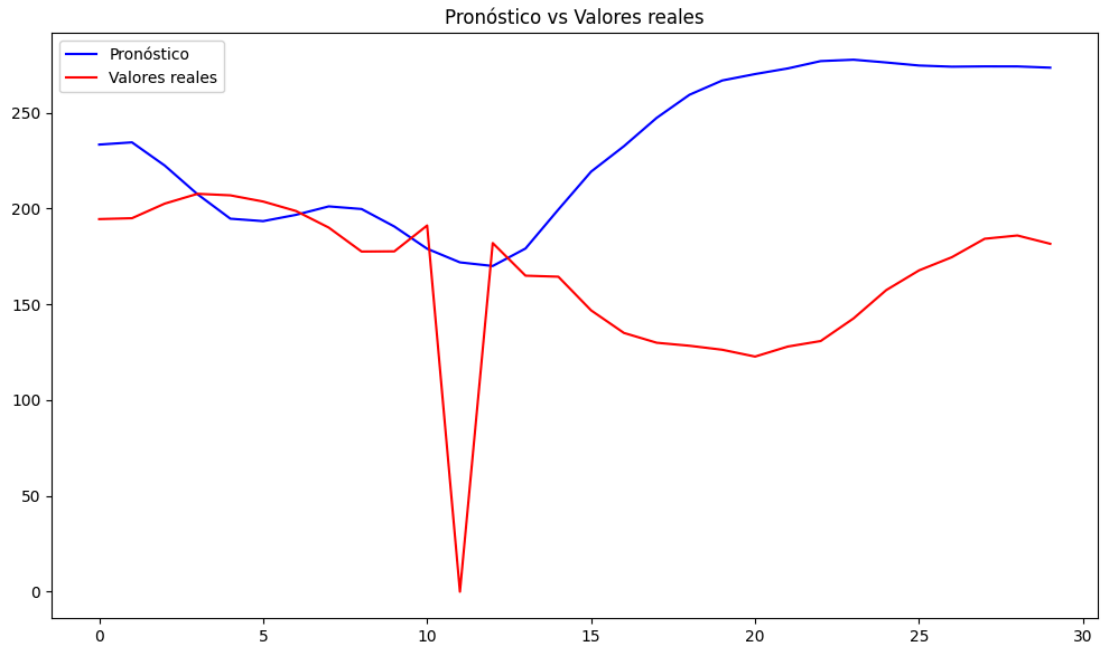
```
296/296 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step
MAE: 53.870331099308025
MSE: 4545.820527178043
MAPE: 26.7%
```

We can see that we obtained a MAPE of 26.7% for the predictions of the model with respect to the test set, which means that it predicted on average this percentage correctly with respect to the real data. To contrast these results we made a graph showing the real data versus the correctly predicted data, as illustrated here:



We can see that our model tends to predict the data quite well in general, except for some peaks shown in the graph where it generates erroneous predictions.

To contrast this information we will perform a Time Series Forecasting using a maximum of 30 steps to predict the time series with respect to the historical, as shown here:



Here we can see that our model is not good at predicting time series in small amounts of data, although it makes a good attempt in the first 10 steps to predict but after that it generates wrong predictions.

7.1.4 CNAVIA Substation

This item shows the training results for the "CNAVIA" substation, where the results obtained in the first 10 epochs are shown here:

```
Epoch 1/200
167/167 ━━━━━━━━━━━━━━━━━━━━ 3s 7ms/step - loss: 56569.9336 - mae: 197.2488 - mse: 56569.9336
Epoch 2/200
167/167 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 11134.2705 - mae: 77.1935 - mse: 11134.2705
Epoch 3/200
167/167 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 2834.9170 - mae: 32.4858 - mse: 2834.9170
Epoch 4/200
167/167 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 2236.3877 - mae: 25.4051 - mse: 2236.3877
Epoch 5/200
167/167 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 2025.4860 - mae: 23.2444 - mse: 2025.4860
Epoch 6/200
167/167 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 1923.1600 - mae: 21.9088 - mse: 1923.1600
Epoch 7/200
167/167 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 1853.5463 - mae: 21.0838 - mse: 1853.5463
Epoch 8/200
167/167 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 1831.9480 - mae: 20.7418 - mse: 1831.9480
Epoch 9/200
167/167 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 1794.1256 - mae: 20.2252 - mse: 1794.1256
Epoch 10/200
167/167 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 1770.3749 - mae: 19.8486 - mse: 1770.3749
```

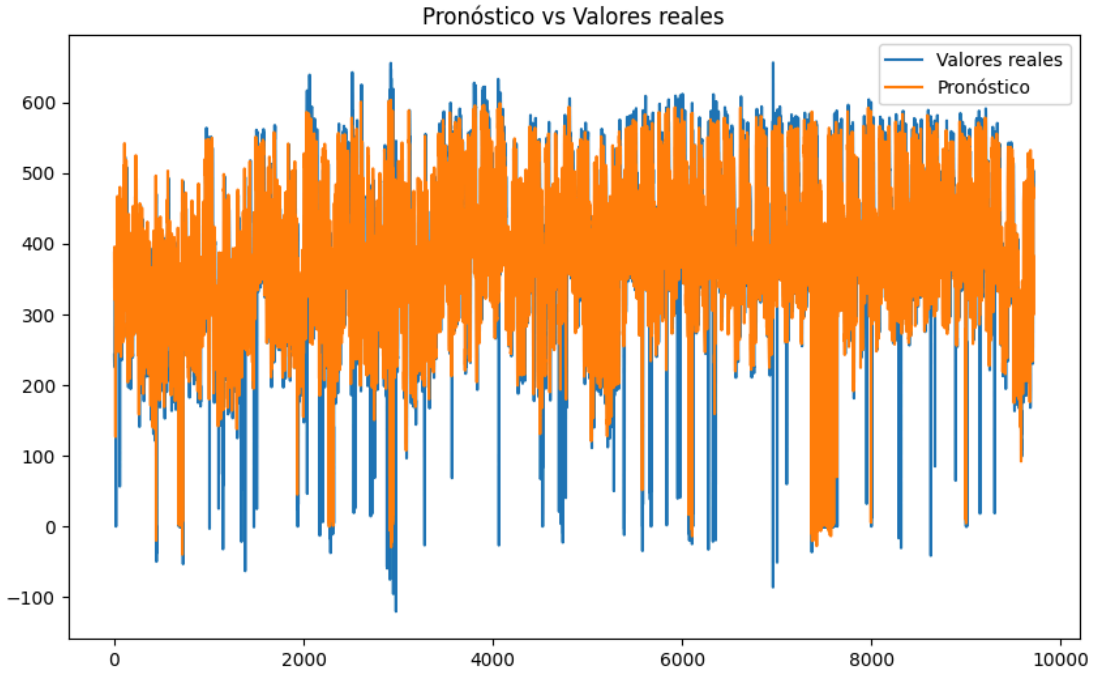
And the results obtained in the last 2 epochs:

```
Epoch 199/200
167/167 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 1437.1862 - mae: 17.7780 - mse: 1437.1862
Epoch 200/200
167/167 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 1435.7983 - mae: 17.9122 - mse: 1435.7983
```

Finally, the evaluation of the training set with the test set is performed, obtaining the following result:

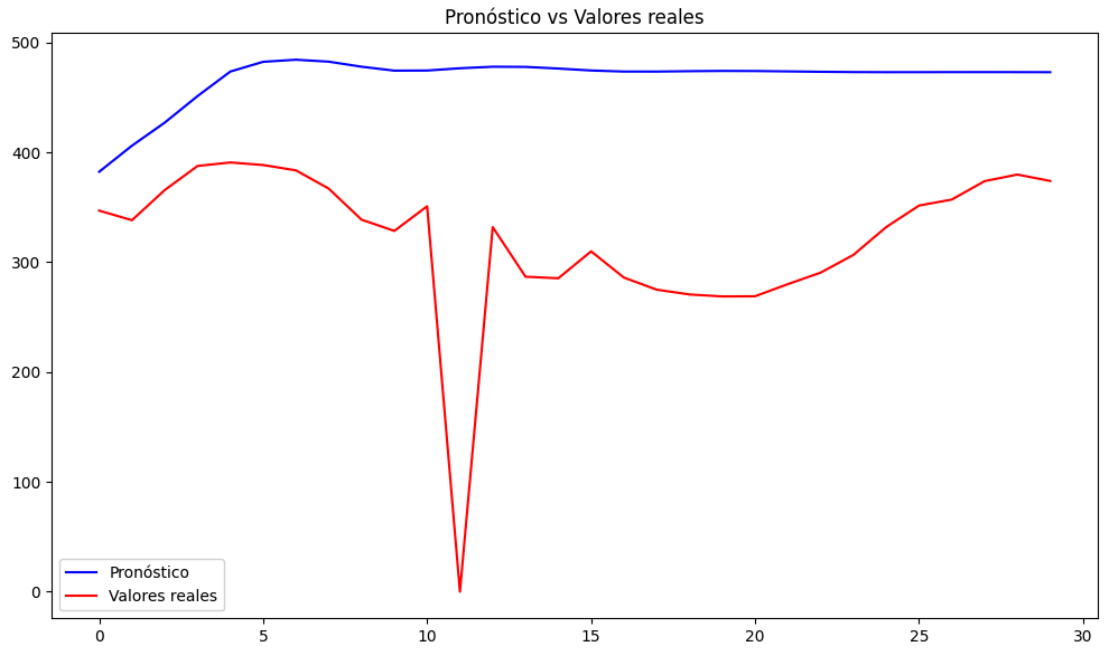
```
304/304 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step
MAE: 122.13145037008677
MSE: 23818.971144230145
MAPE: 31.67%
```

We can see that we obtained a MAPE of 31.67% for the predictions of the model with respect to the test set, which means that it predicted on average this percentage correctly with respect to the real data. To contrast these results we made a graph showing the real data versus the correctly predicted data, as illustrated here:



We can see that our model tends to predict the data quite well in general, except for some peaks shown in the graph where it generates erroneous predictions.

To contrast this information we will perform a Time Series Forecasting using a maximum of 30 steps to predict the time series with respect to the historical, as shown here:



Here we can see that our model is not good at predicting time series in small amounts of data, since in this case the graphs do not cross at any time.

7.1.5 ELSALTO Substation

This item shows the training results for the "ELSALTO" substation, where the results obtained in the first 10 epochs are shown here:

```
Epoch 1/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 3s 7ms/step - loss: 75698.1328 - mae: 224.4645 - mse: 75698.1328
Epoch 2/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 1230.2106 - mae: 20.0593 - mse: 1230.2106
Epoch 3/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 948.3771 - mae: 16.9496 - mse: 948.3771
Epoch 4/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 917.1420 - mae: 16.5071 - mse: 917.1420
Epoch 5/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 900.8382 - mae: 16.2104 - mse: 900.8382
Epoch 6/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 890.2756 - mae: 16.0879 - mse: 890.2756
Epoch 7/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 881.5571 - mae: 15.9411 - mse: 881.5571
Epoch 8/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 882.4318 - mae: 15.9768 - mse: 882.4318
Epoch 9/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 875.1244 - mae: 15.8116 - mse: 875.1244
Epoch 10/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 870.8934 - mae: 15.7508 - mse: 870.8934
```

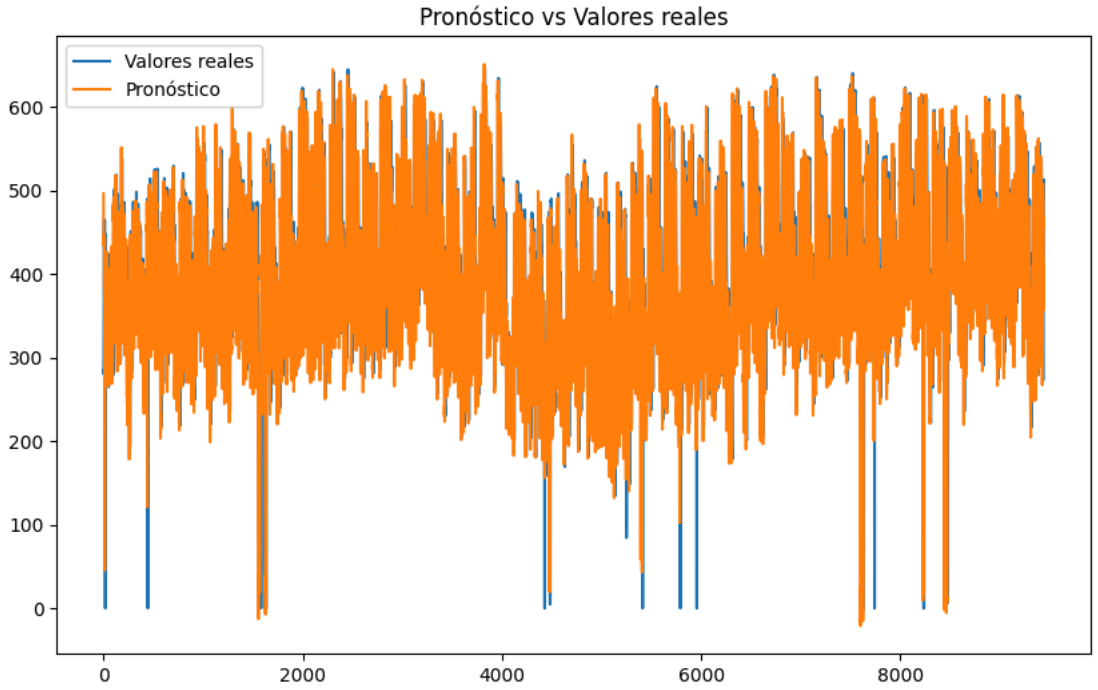
And the results obtained in the last 2 epochs:

```
Epoch 199/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 797.7300 - mae: 15.4999 - mse: 797.7300
Epoch 200/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 790.1578 - mae: 15.2850 - mse: 790.1578
```

Finally, the evaluation of the training set with the test set is performed, obtaining the following result:

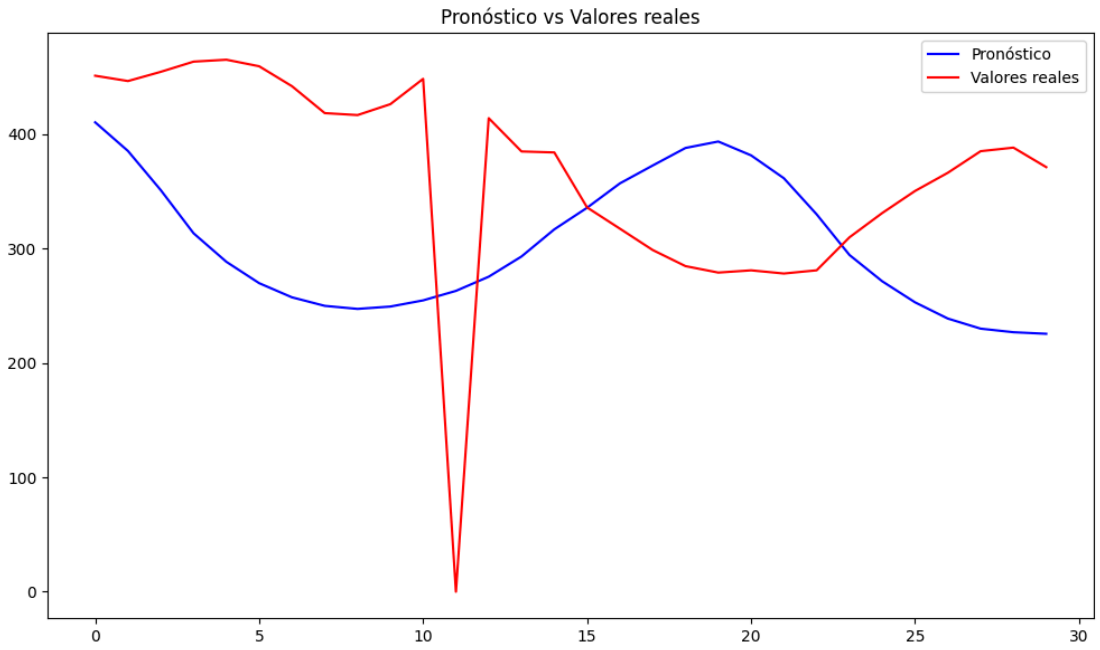
```
295/295 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step
MAE: 111.94887945913801
MSE: 17955.857388895947
MAPE: 27.51%
```

We can see that we obtained a MAPE of 27.51% for the predictions of the model with respect to the test set, which means that it predicted on average this percentage correctly with respect to the real data, and to contrast this information we made a graph showing the real data versus the correctly predicted data, as illustrated here:



We can see that our model tends to predict the data quite well in general, except for some peaks shown in the graph where it generates erroneous predictions.

To contrast this information we will perform a Time Series Forecasting using a maximum of 30 steps to predict the time series with respect to the historical, as shown here:



Here we can see that our model is not good at predicting time series in small amounts of data, since the forecast versus actual values seem to form a wave as the curves cross each other for different times.

7.1.6 FLORIDA Substation

This item shows the training results for the "FLORIDA" substation, where the results obtained in the first 10 epochs are shown here:

```
Epoch 1/200
152/152 ━━━━━━━━━━━━━━━━━━━━ 3s 7ms/step - loss: 82.5436 - mae: 4.1055 - mse: 82.5436
Epoch 2/200
152/152 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 47.7768 - mae: 2.7482 - mse: 47.7768
Epoch 3/200
152/152 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 44.7090 - mae: 2.6979 - mse: 44.7090
Epoch 4/200
152/152 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 42.6402 - mae: 2.5282 - mse: 42.6402
Epoch 5/200
152/152 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 41.4527 - mae: 2.4695 - mse: 41.4527
Epoch 6/200
152/152 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 40.6563 - mae: 2.4384 - mse: 40.6563
Epoch 7/200
152/152 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 40.1248 - mae: 2.4349 - mse: 40.1248
Epoch 8/200
152/152 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 39.5590 - mae: 2.4212 - mse: 39.5590
Epoch 9/200
152/152 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 38.8401 - mae: 2.3927 - mse: 38.8401
Epoch 10/200
152/152 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 38.2550 - mae: 2.3786 - mse: 38.2550
```

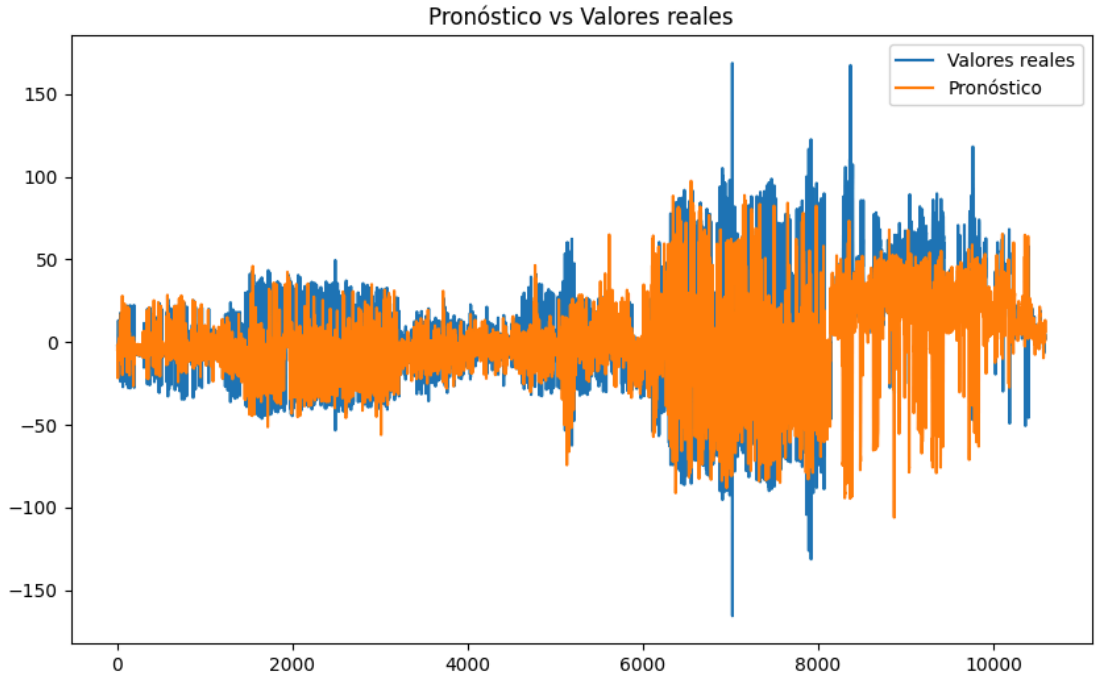
And the results obtained in the last 2 epochs:

```
Epoch 199/200
152/152 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 18.1681 - mae: 1.8604 - mse: 18.1681
Epoch 200/200
152/152 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 17.1680 - mae: 1.8057 - mse: 17.1680
```

Finally, the evaluation of the training set with the test set is performed, obtaining the following result:

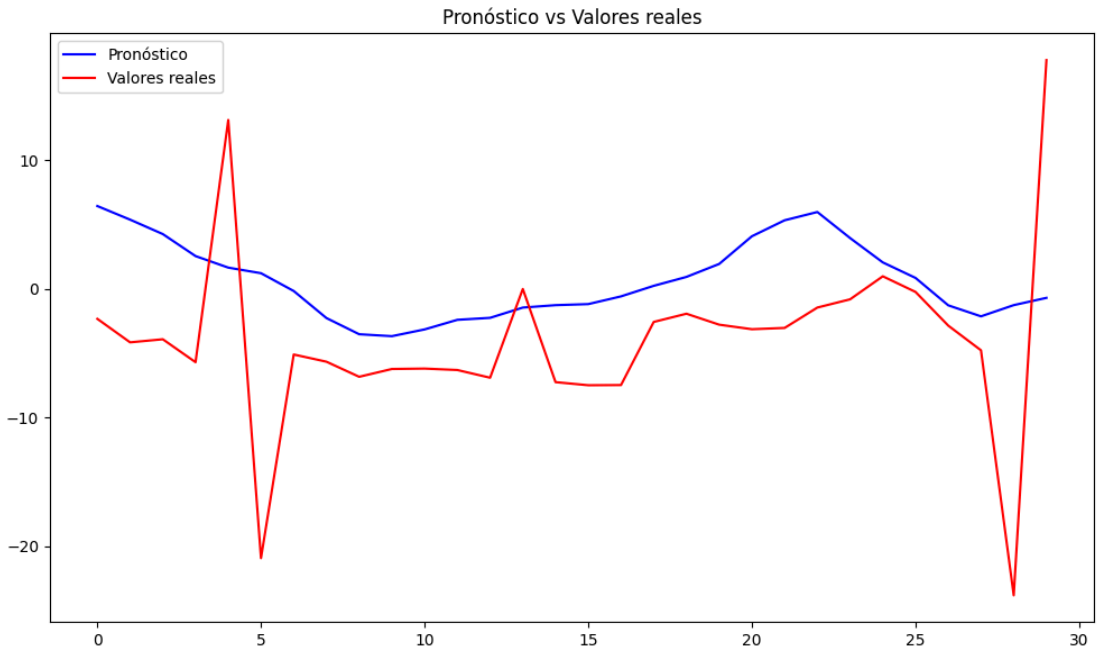
```
332/332 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step
MAE: 17.60071467422195
MSE: 868.598343560301
MAPE: 346.94%
```

We can see that we obtained for the model predictions with respect to the test set a MAPE of 346.9%. This is obviously not statistically possible and in order to contrast these values we made a graph showing the real data versus the correctly predicted data, as illustrated here:



We can see that our model tends to predict the data quite well in general, except for some peaks shown in the graph where it generates erroneous predictions.

To contrast this information we will perform a Time Series Forecasting using a maximum of 30 steps to predict the time series with respect to the historical, as shown here:



Here we can see that our model is not good at predicting time series in small amounts of data, although here it tries to get close enough it does not predict the real values well.

7.1.7 LOSALME Substation

This item shows the training results for the "LOSALME" substation, where the results obtained in the first 10 epochs are shown here:

```
Epoch 1/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 3s 7ms/step - loss: 13429.5479 - mae: 90.4054 - mse: 13429.5479
Epoch 2/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 260.2667 - mae: 9.8784 - mse: 260.2667
Epoch 3/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 200.4670 - mae: 8.2140 - mse: 200.4670
Epoch 4/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 194.0248 - mae: 8.0231 - mse: 194.0248
Epoch 5/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 191.7170 - mae: 7.9048 - mse: 191.7170
Epoch 6/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 186.7912 - mae: 7.8398 - mse: 186.7912
Epoch 7/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 183.0933 - mae: 7.7889 - mse: 183.0933
Epoch 8/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 180.3551 - mae: 7.7517 - mse: 180.3551
Epoch 9/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 178.4369 - mae: 7.7234 - mse: 178.4369
Epoch 10/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 176.3807 - mae: 7.6962 - mse: 176.3807
```

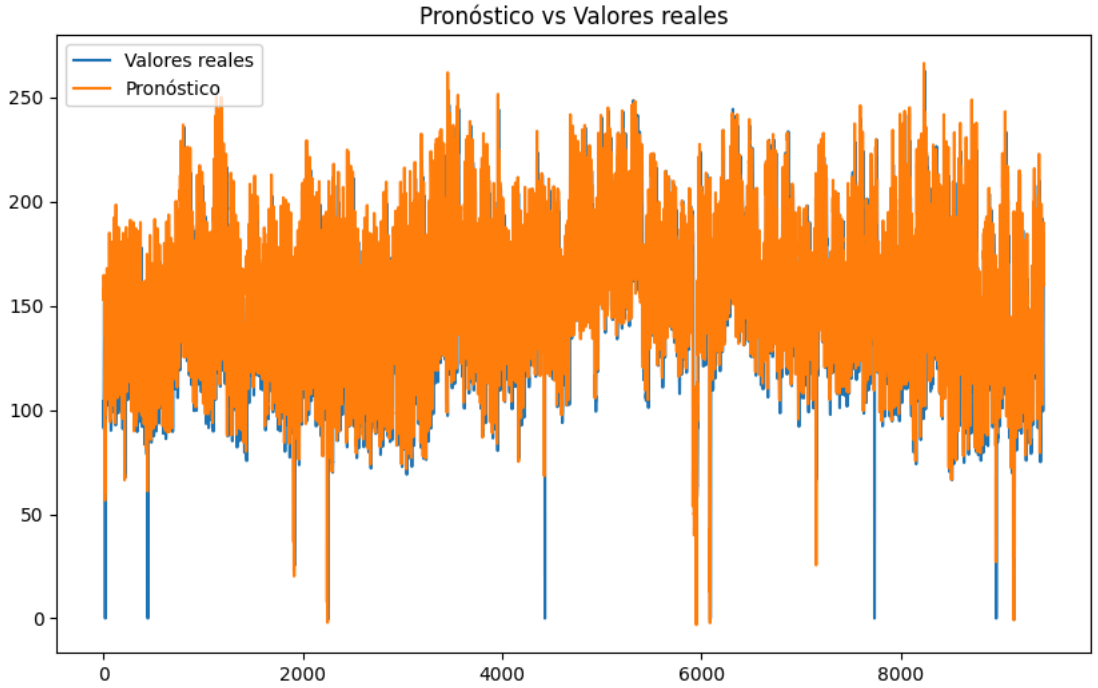
And the results obtained in the last 2 epochs:

```
Epoch 199/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 143.7197 - mae: 7.4850 - mse: 143.7197
Epoch 200/200
140/140 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - loss: 145.8290 - mae: 7.5712 - mse: 145.8290
```

Finally, the evaluation of the training set with the test set is performed, obtaining the following result:

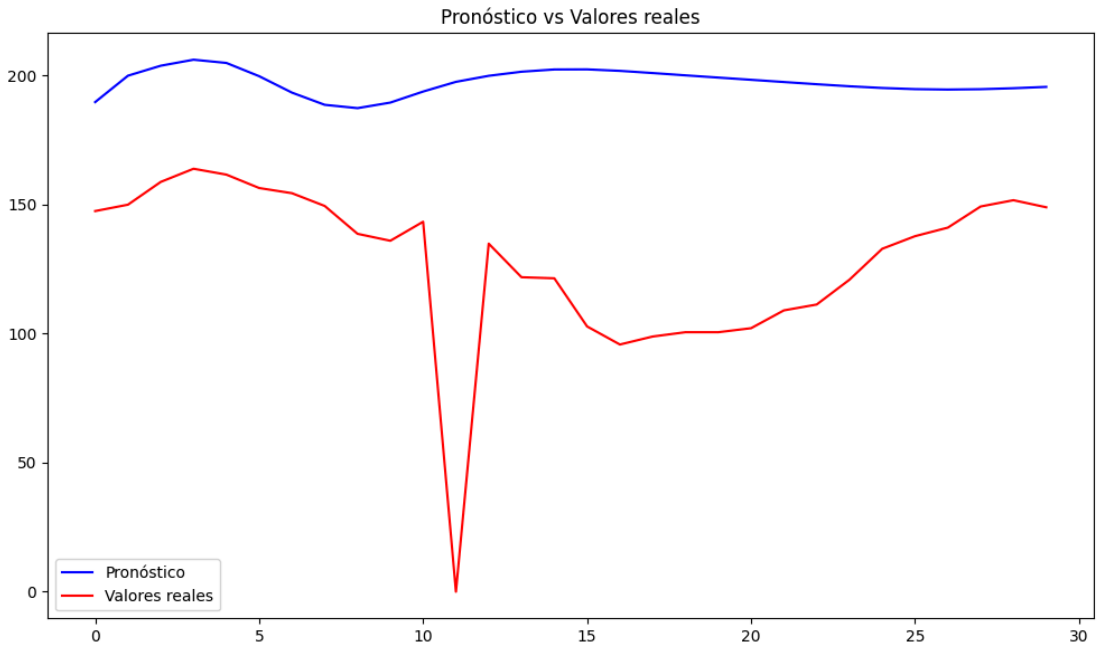
```
295/295 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step
MAE: 40.09750247726953
MSE: 2399.7977670678347
MAPE: 25.91%
```

We can see that we obtained a MAPE of 25.91% for the predictions of the model with respect to the test set, which means that it predicted on average this percentage correctly with respect to the real data, so to contrast this information we made a graph showing the real data versus the correctly predicted data, as illustrated here:



We can see that our model tends to predict the data quite well in general, except for some peaks shown in the graph where it generates erroneous predictions.

To contrast this information we will perform a Time Series Forecasting using a maximum of 30 steps to predict the time series with respect to the historical, as shown here:



Here we can see that our model is not good at predicting time series in small amounts of data, since it never crosses the prediction curves with the curve of the real data.

7.2 Model 2 Results

On the other hand, for the ARIMA we specify the test set and make the predictions by enabling the "levels" option, a parameter recommended for working with time series, so that our model can correctly interpret the input data:

```
# Realizar predicciones
start = 0
end = len(test) - 1
pred = results.predict(start=start, end=end, typ='levels')
```

We also analyze the time series of each substation in the dataset, in order to adjust the parameters of our model. We used "MSE" and "AIC" to calculate the performance of the model in the training and as evaluation metrics we used the "MAPE", which will be fundamental as a criterion for comparison with the other substations. The analysis performed for each substation is shown below.

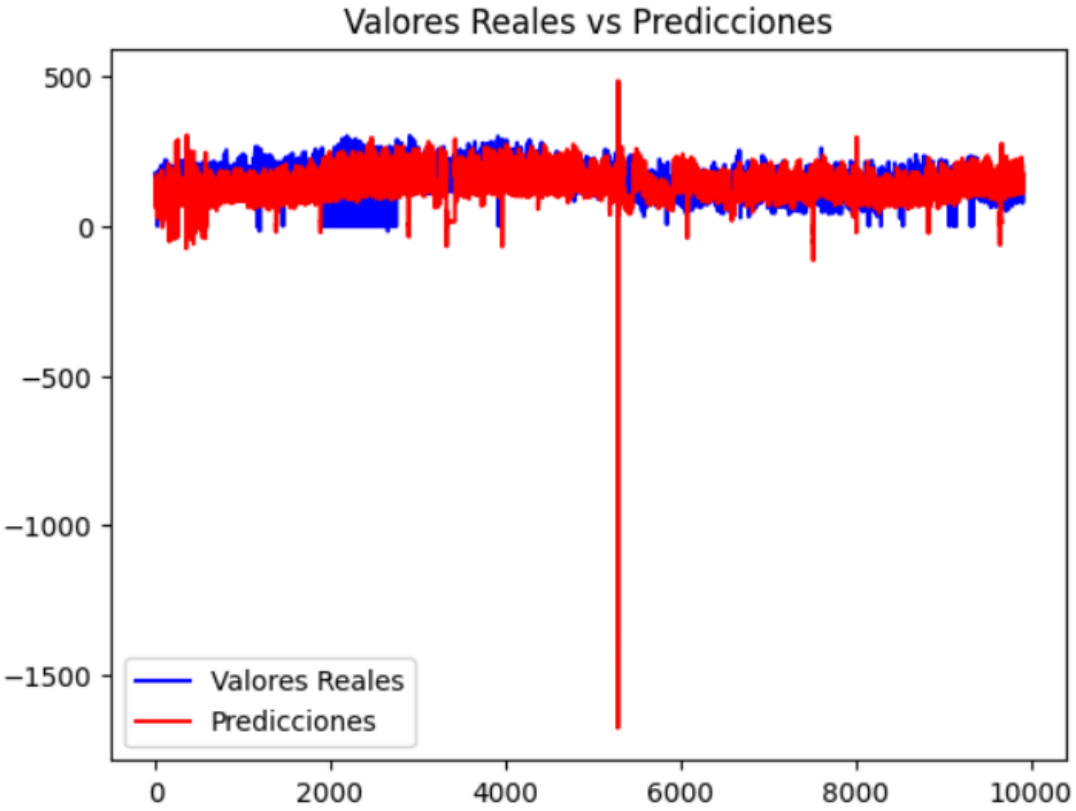
7.2.1 AJAHUEL Substation

This item shows the training results for the “LOSALME” substation, where the results obtained on the end of training were:

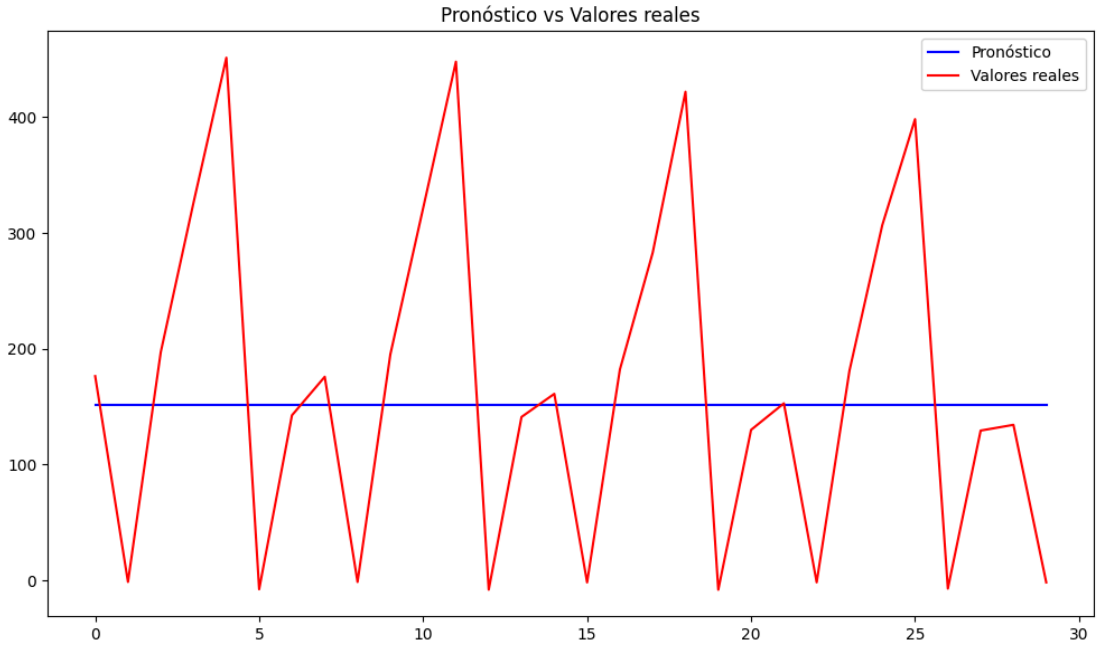
```
MSE : 317.55606149480366
AIC : 309174.67396604986
```

The results of the evaluation metrics indicate an MSE that differs a value of 57.16 MWs from the real values, which can provide us with relevant information in the future when purchasing with the following models, since the MSE strongly penalizes

a prediction error with respect to the real values. On the other hand, we compared the actual data versus the predictions, obtaining the following graph:



It can be seen in the graph that the model in general tends to predict accurately, except for some values that are in the [4000-6000] interval, which in a certain part the real values are notoriously far away. We also performed a forecast with a maximum value of 30 steps, obtaining the following:



We can see that this model is not suitable for predicting small amounts of data, since the forecast line remains constant throughout the forecast.
Finally, we evaluate how well it predicts the data by calculating the metrics obtained by results.predict() where obtained:

```
MAE: 57.162527081849156
MSE: 6307.355710800036
MAPE: 37.46%
```

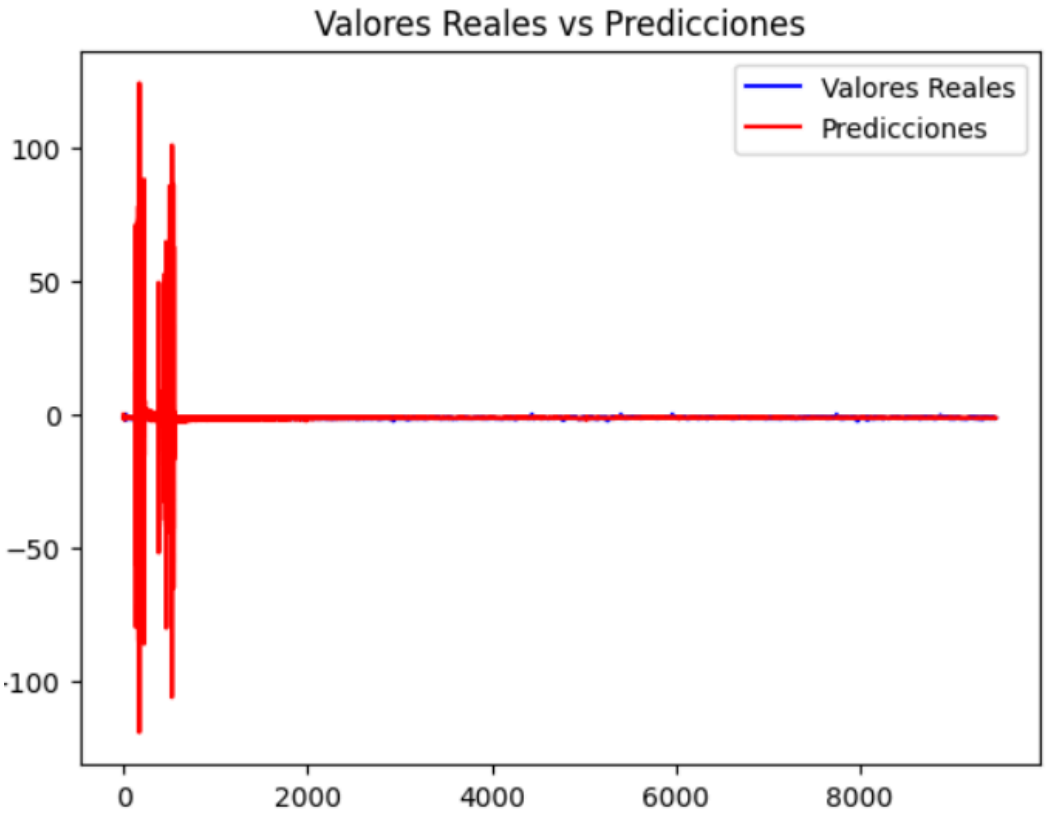
It was obtained that the ARIMA model has an error of 37.46% with respect to the real data for this substation, based on the results of the MAPE.

7.2.2 BUIN Substation

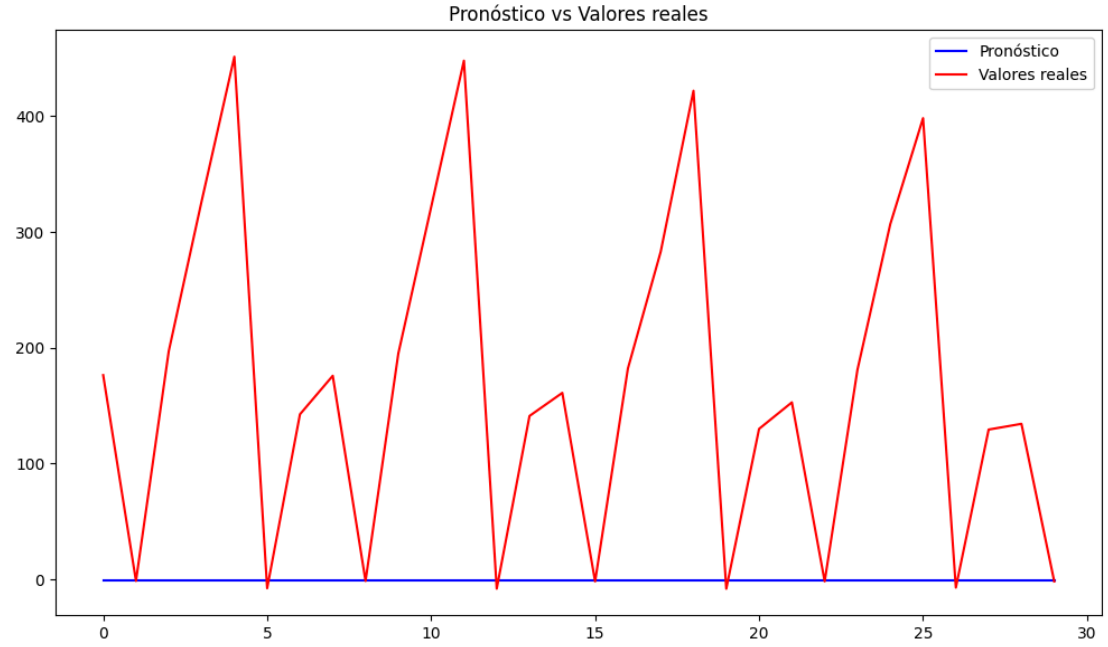
This item shows the training results for the "BUIN" substation, where the results obtained on the end of training were:

```
MSE : 33.35509182779764
AIC : 226558.46342007793
```

In this substation we have an MSE of 33.3, very low compared to the previous substation, this indicates that the values differ little, giving as an idea that the predictions are quite accurate with respect to reality. On the other hand, we compare the real data versus the predictions, obtaining the following graph:



As mentioned above, the MSE has a minimal difference, which can be clearly seen in the graph, the blue values of the real data being almost invisible, since they follow a very similar trend. We also performed a forecast with a maximum value of 30 steps, obtaining the following:



We can see that this model is not suitable for predicting small amounts of data, since the forecast line remains constant throughout the forecast.

Finally, we evaluate how well it predicts the data by calculating the metrics obtained by results.predict() where obtained:

```
MAE: 0.7550406106478804
MSE: 31.25700538544103
MAPE: 55.53%
```

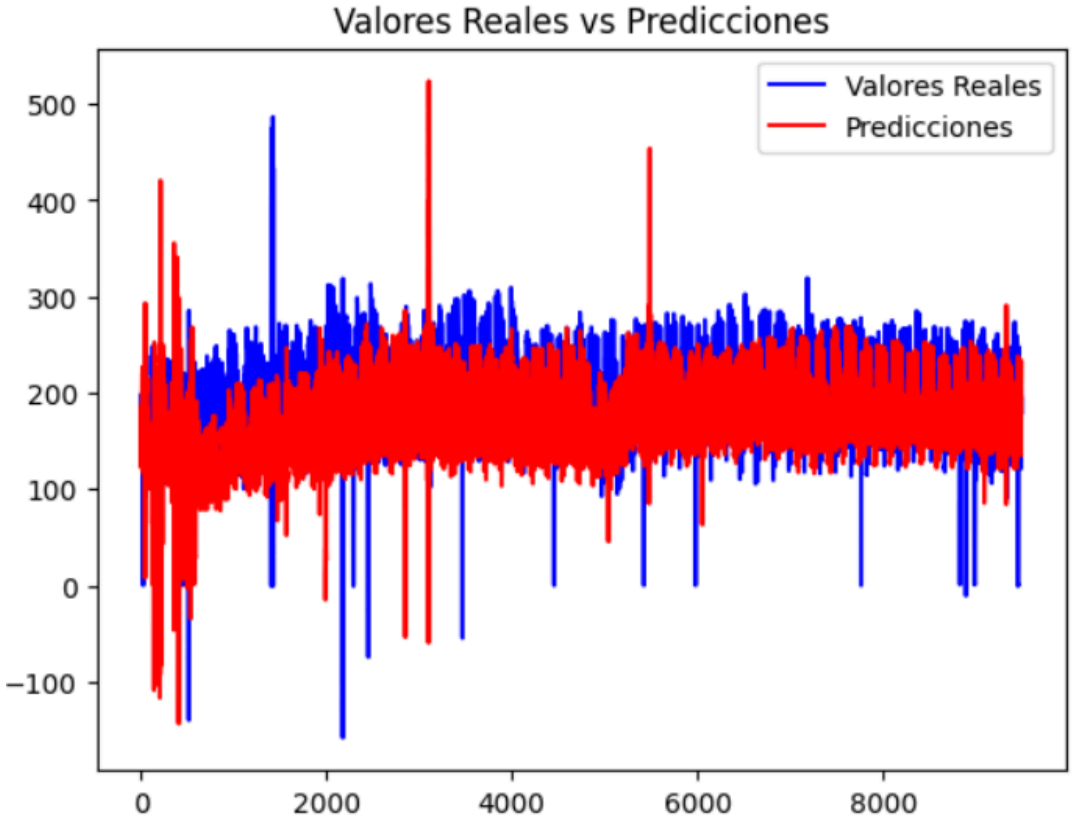
It was obtained that the ARIMA model has an error of 55.53% with respect to the real data for this substation, based on the results of the MAPE.

7.2.3 CHENA Substation

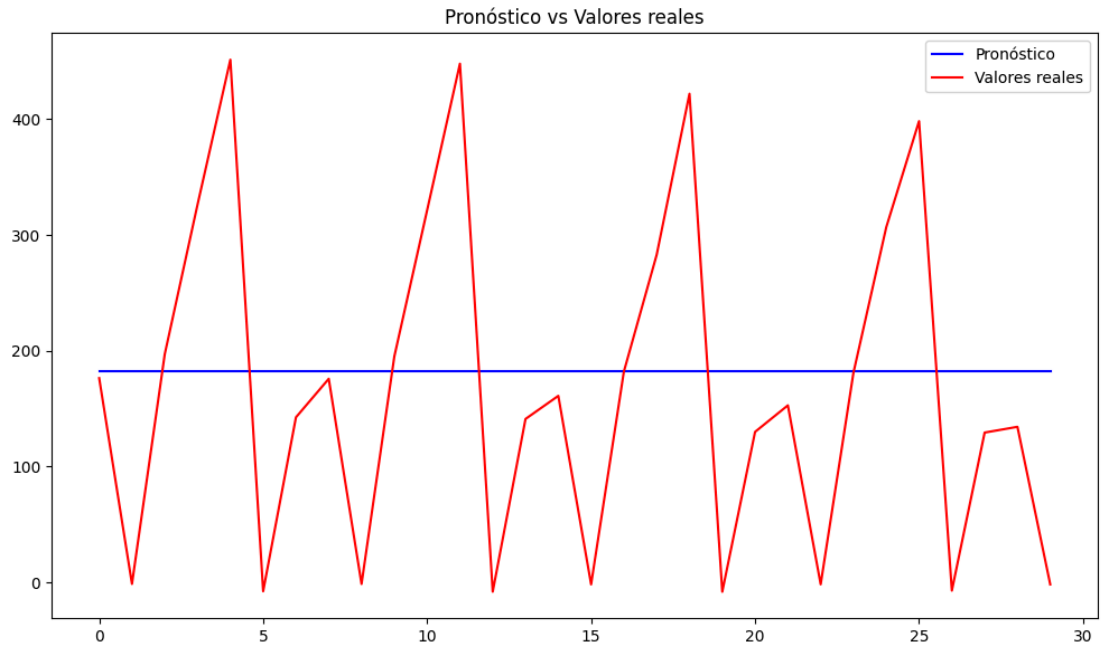
This item shows the training results for the "LOSALME" substation, where the results obtained on the end of training were:

```
MSE : 590.0139831688231
AIC : 331422.8441218626
```

The results of the evaluation metrics indicate that the MSE differs a value of 590.01 MWs with the actual values. On the other hand, we compared the actual data versus the predictions, obtaining the following graph:



As can be seen, the graph indicates that there was a fairly accurate prediction with respect to the actual data except for some peaks shown in the graph where it generates erroneous predictions. We also performed a forecast with a maximum value of 30 steps, obtaining the following:



We can see that this model is not suitable for predicting small amounts of data, since the forecast line remains constant throughout the forecast.
Finally, we evaluate how well it predicts the data by calculating the metrics obtained by results.predict() where obtained:

```
MAE: 55.07471780504029
MSE: 4846.421105512357
MAPE: 27.3%
```

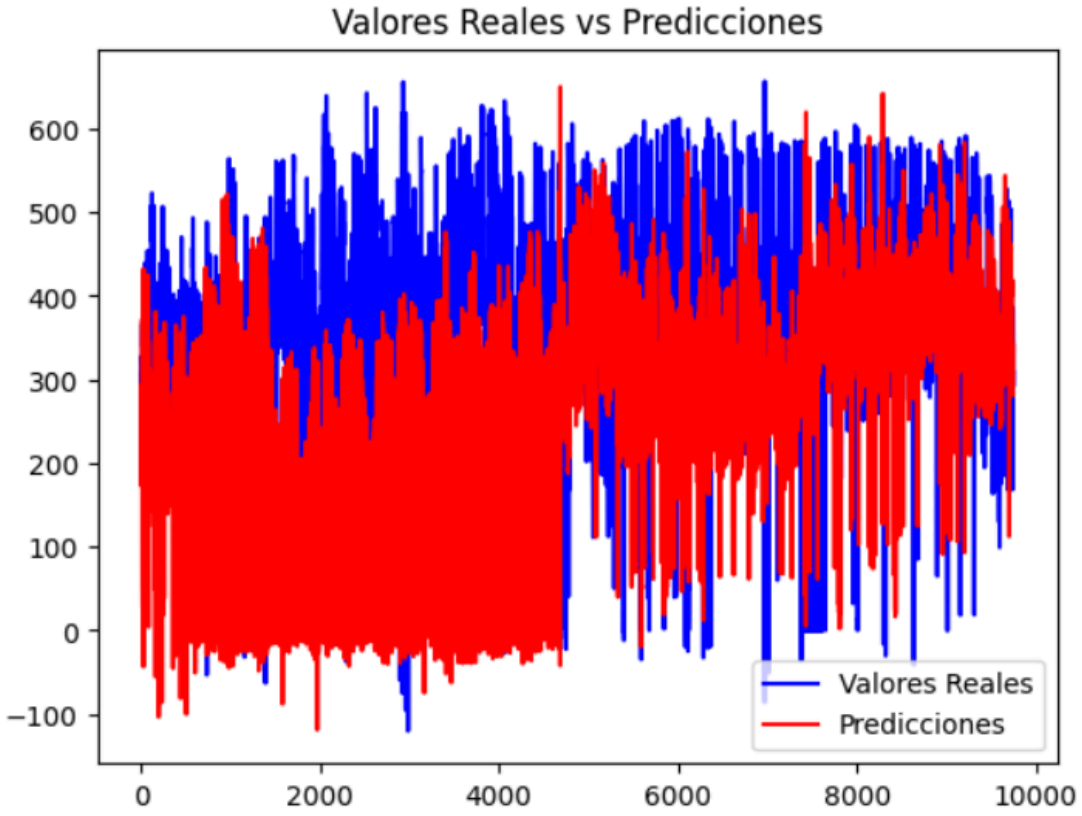
It was obtained that the ARIMA model has an error of 27.3% with respect to the real data for this substation, based on the results of the MAPE.

7.2.4 CNAVIA Substation

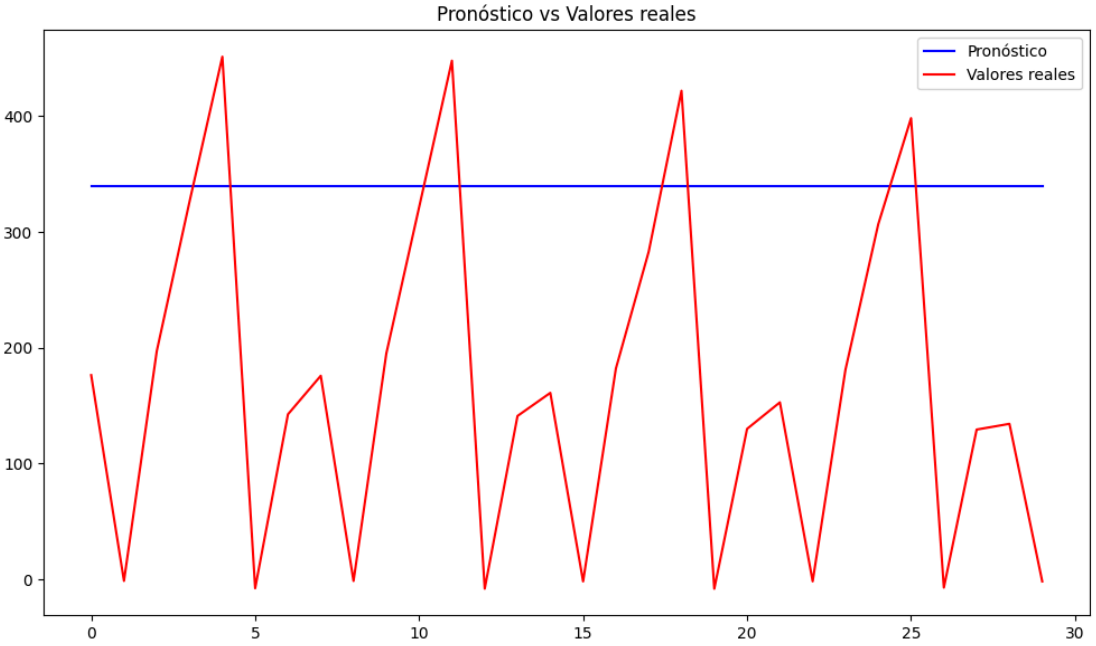
This item shows the training results for the “LOSALME” substation, where the results obtained on the end of training were:

```
MSE : 2760.1800911299506
AIC : 458786.4628488623
```

With respect to the results of the metrics used, it can be seen that the data differ quite a lot, with an MSE of 2760.18 MWS difference with reality. On the other hand, we compared the actual data versus the predictions, obtaining the following graph:



It can be seen that the predictions are not entirely accurate; in different ranges of the graph there are low peaks, where the predictions are quite far from reality. In general, the accuracy is somewhat irregular and this can be verified with the metrics mentioned above. We also performed a forecast with a maximum value of 30 steps, obtaining the following:



We can see that this model is not suitable for predicting small amounts of data, since the forecast line remains constant throughout the forecast.

Finally, we evaluate how well it predicts the data by calculating the metrics obtained by results.predict() where obtained:

```
MAE: 185.08509453398344
MSE: 54626.473813977536
MAPE: 48.03%
```

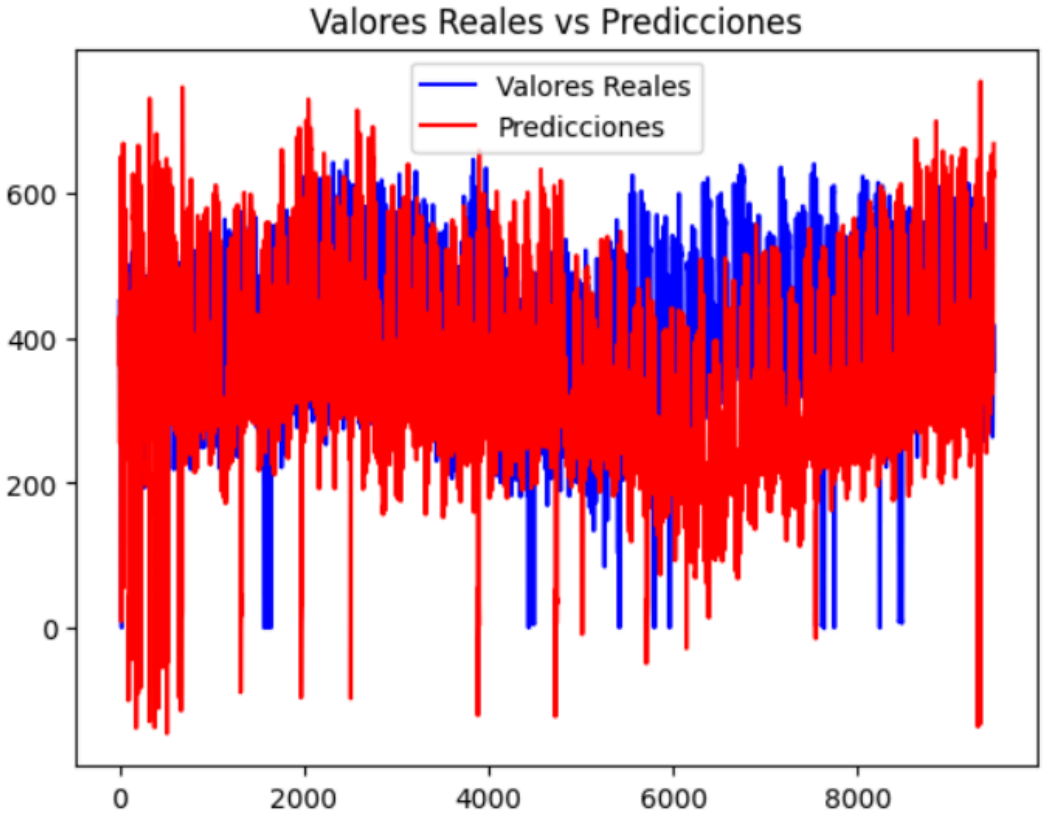
It was obtained that the ARIMA model has an error of 48.03% with respect to the real data for this substation, based on the results of the MAPE.

7.2.5 ELSALTO Substation

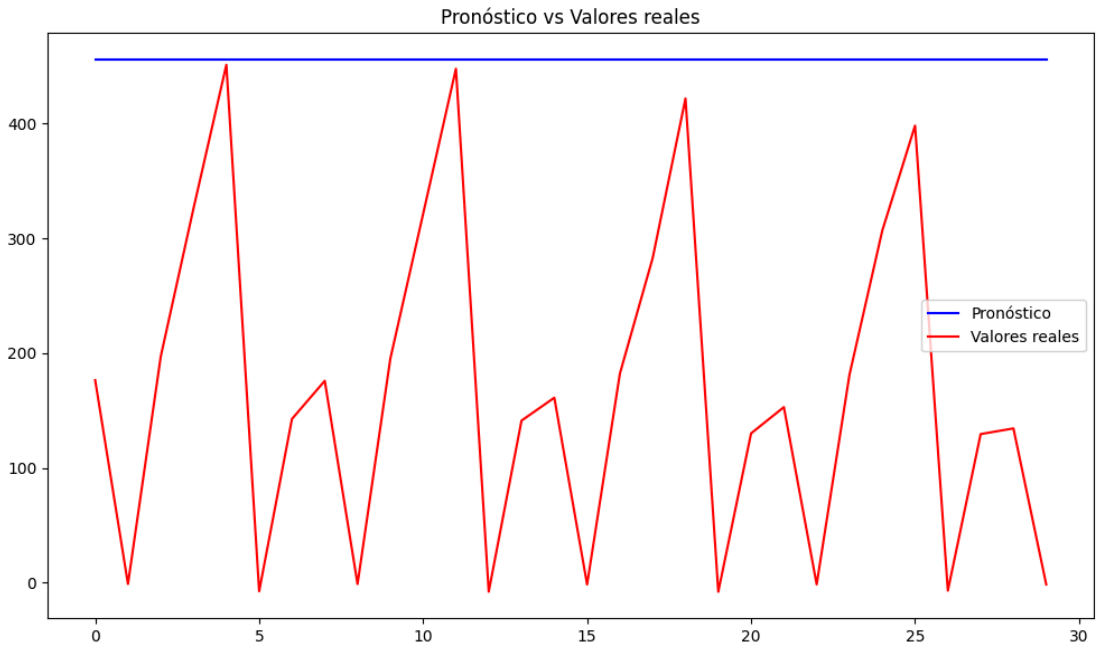
This item shows the training results for the "LOSALME" substation, where the results obtained on the end of training were:

```
MSE : 1108.802766971194
AIC : 351862.39970750874
```

With respect to the results of the metrics used, it can be seen that the data differ quite a lot, with an MSE of 1108.8 MWs difference with reality. On the other hand, we compared the actual data versus the predictions, obtaining the following graph:



We can see that our model has a tendency to predict the data generally well, except for some peaks shown in the graph where it generates erroneous predictions. We also performed a forecast with a maximum value of 30 steps, obtaining the following:



We can see that this model is not suitable for predicting small amounts of data, since the forecast line remains constant throughout the forecast.
Finally, we evaluate how well it predicts the data by calculating the metrics obtained by results.predict() where obtained:

```
MAE: 127.33477447621851
MSE: 25469.297064723116
MAPE: 31.3%
```

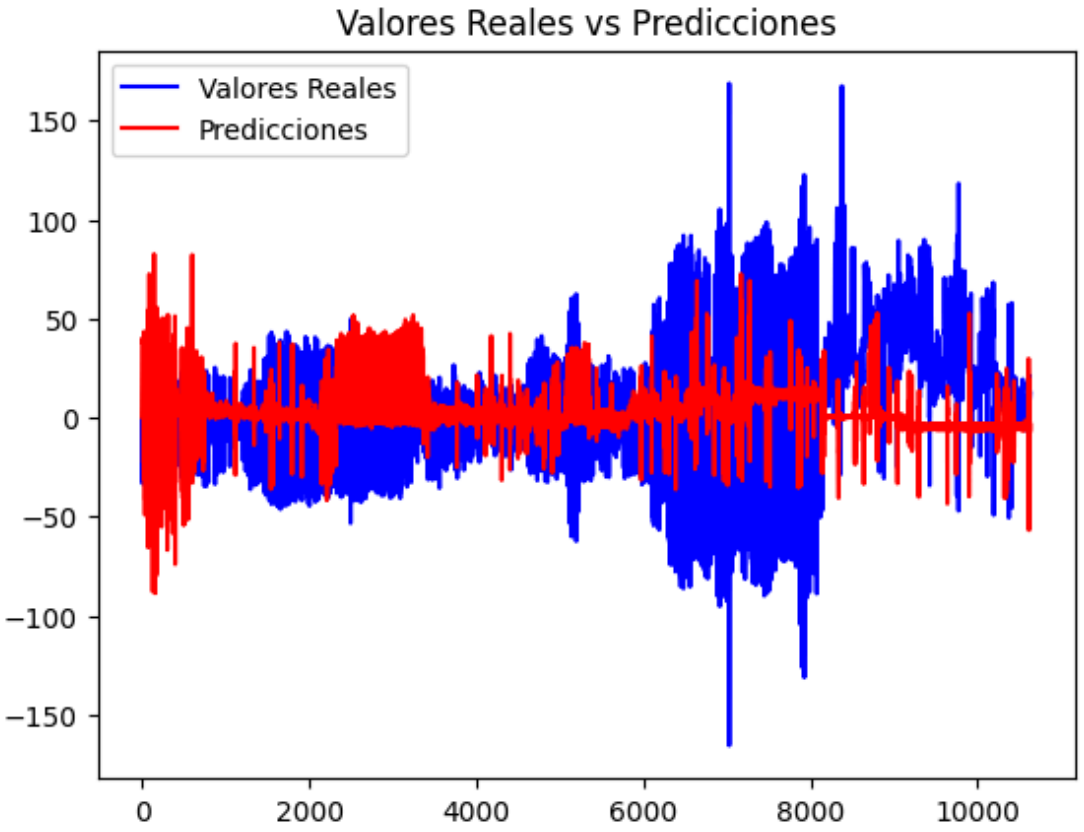
It was obtained that the ARIMA model has an error of 31.3% with respect to the real data for this substation, based on the results of the MAPE.

7.2.6 FLORIDA Substation

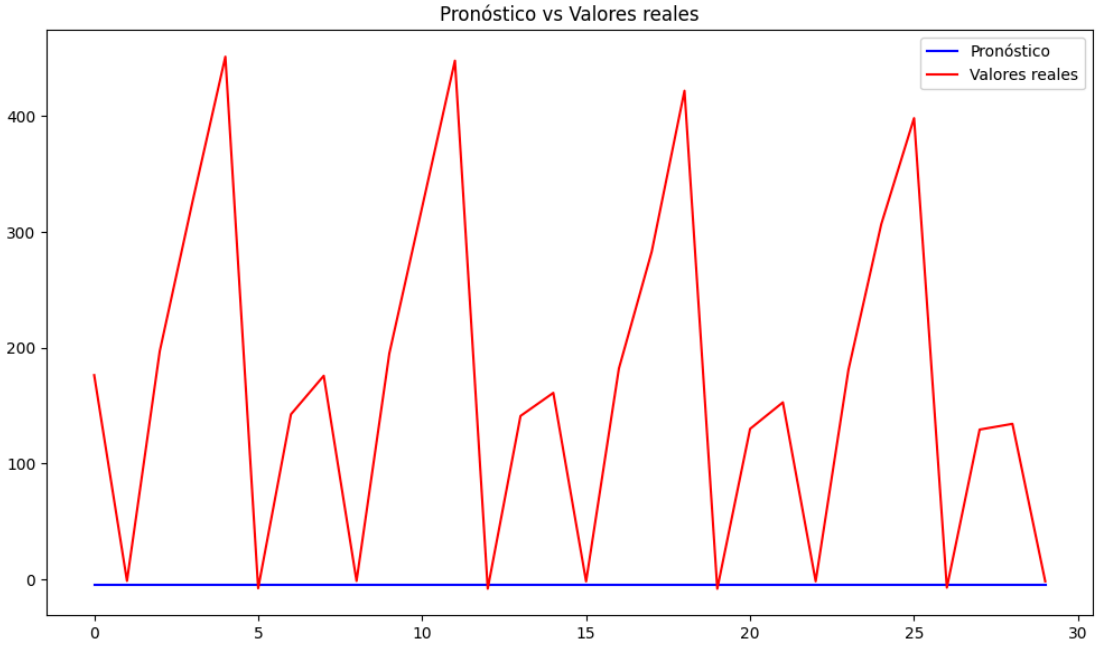
This item shows the training results for the “LOSALME” substation, where the results obtained on the end of training were:

```
MSE : 72.28950617303573
AIC : 275860.78143284674
```

With respect to the results of the metrics used, it can be seen that the data differ quite a lot, a low MSE of 72.28 is shown, but it is not a reliable metric to determine the behavior for this substation. Therefore, we compare the actual data versus the predictions, obtaining the following graph:



We can see that our model cannot accurately predict the data for this substation, this is also due to the fact that FLORIDA in particular has a large number of outliers. We also performed a forecast with a maximum value of 30 steps, obtaining the following:



We can see that this model is not suitable for predicting small amounts of data, since the forecast line remains constant throughout the forecast.
Finally, we evaluate how well it predicts the data by calculating the metrics obtained by results.predict() where obtained:

```
MAE: 21.942064697090853
MSE: 917.5569675109081
MAPE: 431.4%
```

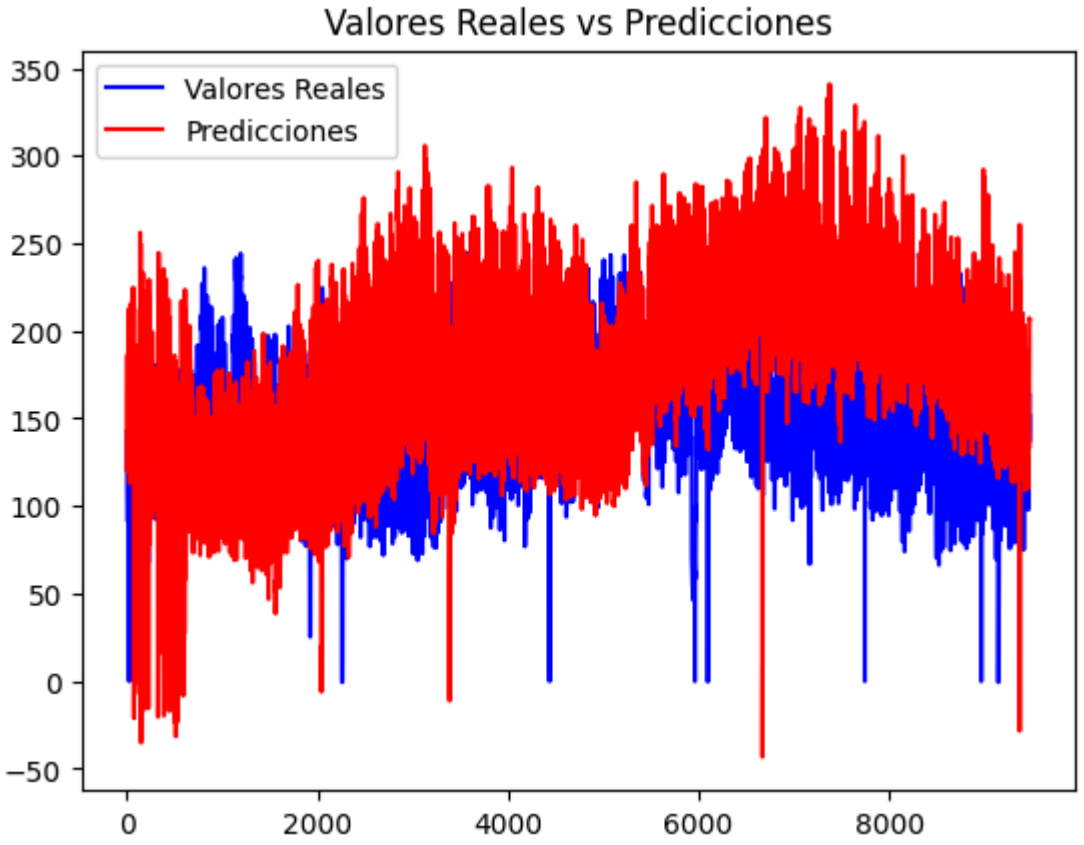
We can see that the ARIMA model has an error of 431.4%, which is mathematically not possible and therefore indicates that the data for this substation may have data collection problems, since the same evaluation problem was obtained when performing the RNN model for this substation.

7.2.7 LOSALME Substation

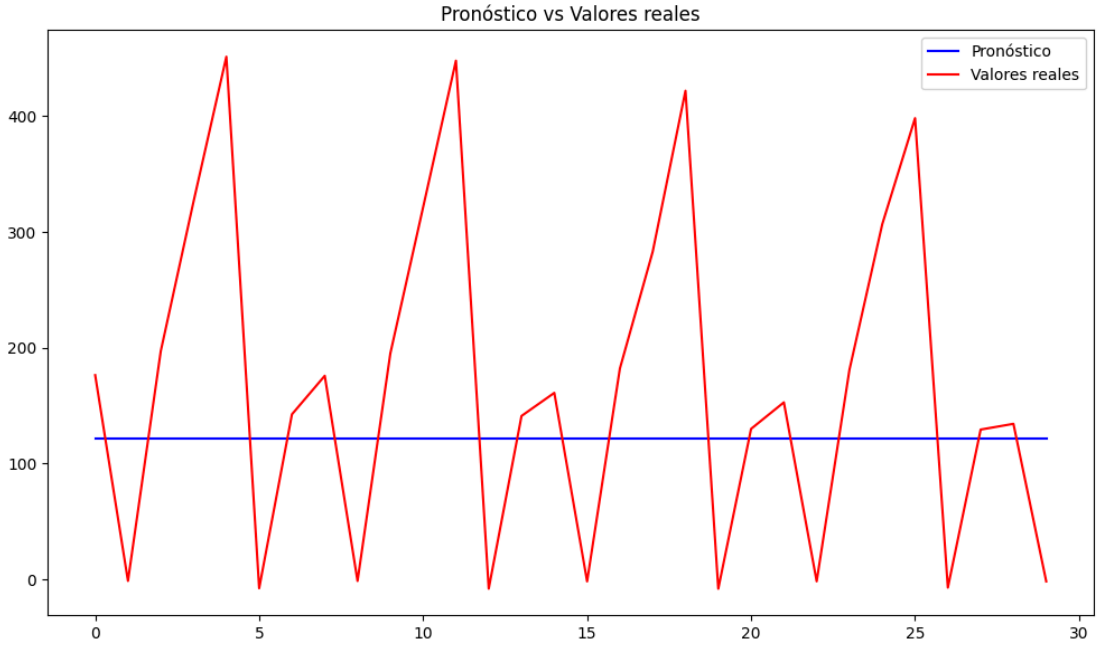
This item shows the training results for the "LOSALME" substation, where the results obtained on the end of training were:

```
MSE : 262.13132975760976
AIC : 300798.7870729921
```

With respect to the results of the metrics used, it can be seen that the data differ quite a lot, an MSE of 262.13 is shown but we need more information about the model. Therefore, we compare the actual data versus the predictions, obtaining the following graph:



We can see that our model predicts the model data quite well. Also, we performed a forecast with a maximum value of 30 steps, obtaining the following:



We can see that this model is not suitable for predicting small amounts of data, since the forecast line remains constant throughout the forecast.
Finally, we evaluate how well it predicts the data by calculating the metrics obtained by results.predict() where obtained:

```
MAE: 52.542672745956686
MSE: 4231.869024082596
MAPE: 33.96%
```

It was obtained that the ARIMA model has an error of 33.96% with respect to the real data for this substation, based on the results of the MAPE.

8. Choice of model

Regarding the results obtained between both models we must consider that if our prediction data is a time series with a small amount of data (considering up to n=1000) probably none of the models seen before will serve to correctly predict this data. On the other hand, if our n>1000 then to determine which of the models is better we will analyze each one with respect to the metrics obtained, the following table shows the data obtained for Model 1:

Substation\Metric	MAE	MSE	MAPE
AJAHUEL	66.14	7657.86	43.31%
BUIN	0.22	0.07	16.18%
CHENA	53.87	4545.82	26.7%
CNAVIA	122.13	23818.97	31.67%
ELSALTO	111.94	17955.85	27.51%
LOSALME	40.1	2399.8	25.91%
PROMEDIO			28.55%

And in this other table, the metrics obtained for Model 2 are shown:

Substation\Metric	MAE	MSE	MAPE
AJAHUEL	57.16	6307.36	37.46%
BUIN	0.75	31.25	55.53%
CHENA	55.07	4846.42	27.3%
CNAVIA	185.1	54626.47	48.03%
ELSALTO	127.33	25469.29	31.3%
LOSALME	52.54	4231.87	33.96%
PROMEDIO			38.93%

The "FLORIDA" substation was not considered for model evaluation since it has problems in data collection, thus generating evaluation problems when training any of the models presented.
Regarding the metrics previously obtained we can determine that the model that best predicts in all datasets is Model 1 (RNN) since we obtained an average MAPE of 28.55% versus an average MAPE of 38.93% obtained in ARIMA, this means that

Model 1 has a lower error percentage than ARIMA to predict the data, although it is true that if we wish to obtain a reliable model we should try to decrease that percentage at least to a MAPE value $\leq 10\%$, which leaves room for improvement for this model.

9. Bibliography

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning: A Textbook*. Springer International Publishing.

Coordinador Eléctrico Nacional. (2023, April 19). *Webinar: Los avances de la nueva Programación Intradiaria*. YouTube. Retrieved May 1, 2024, from https://www.youtube.com/watch?v=6PQ-KSx1P4o

Cuellar, J. (2021, December 4). *Prueba de raíz unitaria de Dickey & Fuller*. RPubs. Retrieved May 1, 2024, from https://rpubs.com/JessicaCuellar/843574

Field, A., Miles, J., & Field, Z. (2012). *Discovering Statistics Using R*. SAGE Publications.

Keras Team. (n.d.). *Keras 3 API documentation*. Keras. Retrieved May 1, 2024, from https://keras.io/api/

Machine Learning in Plain English. (2023, June 14). *Deep Learning Course — Lesson 11: Model Evaluation Metrics | by Machine Learning in Plain English*. Medium. Retrieved May 1, 2024, from https://medium.com/@nerdjock/deep-learning-course-lesson-11-model-evaluation-metrics-d85d0b85bcca

Sauma, E. (2018, September 21). *¿Cómo está regulado el mercado eléctrico chileno?* Clase Ejecutiva UC. Retrieved May 1, 2024, from https://www.claseejecutiva.uc.cl/blog/articulos/como-esta-regulado-el-mercado-electrico-chileno/

Schwarzenberg, P. (n.d.). *CINF104*. Github. Retrieved May 1, 2024, from https://github.com/pabloschwarzenberg/CINF103

Shumway, R. H., & Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples*. Springer International Publishing.