Problem Statement

By Sofia Cantu

You are a data scientist working for a school. You are asked to predict the GPA of the current students based on the following provided data:

- 0 StudentID int64
  1 Age int64
  2 Gender int64
  3 Ethnicity int64
  4 ParentalEducation int64
  5 StudyTimeWeekly float64 6 Absences int64
  7 Tutoring int64
  8 ParentalSupport int64
  9 Extracurricular int64
  10 Sports int64
  11 Music int64
  12 Volunteering int64
  13 GPA float64 14 GradeClass float64

The GPA is the Grade Point Average, typically ranges from 0.0 to 4.0 in most educational systems, with 4.0 representing an 'A' or excellent performance.

The minimum passing GPA can vary by institution, but it's often around 2.0. This usually corresponds to a 'C' grade, which is considered satisfactory.

You need to create a Deep Learning model capable to predict the GPA of a Student based on a set of provided features. The data provided represents 2,392 students.

In this excersice you will be requested to create a total of three models and select the most performant one.

1) Import Libraries

First let's import the following libraries, if there is any library that you need and is not in the list bellow feel free to include it

In [32]:
```
# Bibliotecas de Ciencia de Datos
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

```
# Bibliotecas de Visualización
import matplotlib.pyplot as plt

# Bibliotecas de Aprendizaje Automático y Deep Learning
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Conv1D, MaxPooling1D, Fl
from tensorflow.keras.regularizers import l2
```

2) Load Data

- You will be provided with a cvs (comma separated value) file.
- You will need to add that file into a pandas dataframe, you can use the following code as reference
- The file will be available in canvas

In [33]:
```
data = pd.read_csv("M2_A2_StudentPerformanceData.csv")
data.set_index('StudentID', inplace=True) # Index como StudentID
data
```

Out[33]:

	Age	Gender	Ethnicity	ParentalEducation	StudyTimeWeekly	Absences	T
StudentID							
1001	17	1	0	2	19.833723	7	
1002	18	0	0	1	15.408756	0	
1003	15	0	2	3	4.210570	26	
1004	17	1	0	3	10.028829	14	
1005	17	1	0	2	4.672495	17	
...	...	...	...	...	...	...	
3388	18	1	0	3	10.680555	2	
3389	17	0	0	1	7.583217	4	
3390	16	1	0	2	6.805500	20	
3391	16	1	1	0	12.416653	17	
3392	16	1	0	2	17.819907	13	

2392 rows × 14 columns

3) Review you data:

Make sure you review your data. Place special attention of null or empty values.

In [34]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2392 entries, 1001 to 3392
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   Age                2392 non-null   int64
 1   Gender             2392 non-null   int64
 2   Ethnicity          2392 non-null   int64
 3   ParentalEducation  2392 non-null   int64
 4   StudyTimeWeekly    2392 non-null   float64
 5   Absences           2392 non-null   int64
 6   Tutoring           2392 non-null   int64
 7   ParentalSupport    2392 non-null   int64
 8   Extracurricular    2392 non-null   int64
 9   Sports             2392 non-null   int64
 10  Music              2392 non-null   int64
 11  Volunteering       2392 non-null   int64
 12  GPA                2392 non-null   float64
 13  GradeClass         2392 non-null   float64
dtypes: float64(3), int64(11)
memory usage: 280.3 KB
```

4. Remove the columns not needed for Student performance prediction

- Choose only the columns you consider to be valuable for your model training.
- For example, StudentID might not be a good feature for your model, and thus should be removed from your main dataset, which other columns should also be removed?
- You can name that final dataset as 'dataset'

In [35]:
```
data.dropna(inplace=True)
X = data.drop('GPA', axis=1)
y = data['GPA']
```

5. Check if the columns has any null values:

- Here you now have your final dataset to use in your model training.
- Before moving foward review your data check for any null or empty value that might be needed to be removed

In [36]:
```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2392 entries, 1001 to 3392
Data columns (total 13 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Age                2392 non-null   int64  
 1   Gender             2392 non-null   int64  
 2   Ethnicity          2392 non-null   int64  
 3   ParentalEducation  2392 non-null   int64  
 4   StudyTimeWeekly    2392 non-null   float64
 5   Absences           2392 non-null   int64  
 6   Tutoring           2392 non-null   int64  
 7   ParentalSupport    2392 non-null   int64  
 8   Extracurricular    2392 non-null   int64  
 9   Sports             2392 non-null   int64  
 10  Music              2392 non-null   int64  
 11  Volunteering       2392 non-null   int64  
 12  GradeClass         2392 non-null   float64
dtypes: float64(2), int64(11)
memory usage: 261.6 KB
```

In [37]:

```
y.info()
```

```
<class 'pandas.core.series.Series'>
Index: 2392 entries, 1001 to 3392
Series name: GPA
Non-Null Count  Dtype  
--------------  -----  
2392 non-null   float64
dtypes: float64(1)
memory usage: 37.4 KB
```

6. Prepare your data for training and for testing set:

- First create a dataset named X, with all columns but GPA. These are the features
- Next create another dataset named y, with only GPA column. This is the label
- If you go to your Imports, you will see the following import: **'from sklearn.model_selection import train_test_split'**
- Use that *train_test_split* function to create: X_train, X_test, y_train and y_test respectively. Use X and y datasets as parameters. Other parameters to use are: Test Size = 0.2, Random State = 42.
- Standarize your features (X_train and X_test) by using the StandardScaler (investigate how to use fit_transform and transform functions). This will help the training process by dealing with normilized data.

Note: Your X_train shape should be around (1913, 10). This means the dataset has 10 columns which should be the input.

In [38]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran

scaler = StandardScaler() # Escala los datos
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

7. Define your Deep Neural Network.

- This will be a Sequential Neural Network.
- With a Dense input layer with 64 units, and input dimention of 10 and Relu as the activation function.
- A Dense hidden layer with 32 units, and Relu as the activation function.
- And a Dense output layer with 1 unit, do not define an activation function so it defaults to linear, suitable for regression tasks. e.g. Dense(1)

This last part of the output layer is super important, since we want to predict the GPA, this means that we want a regression and not a classification. Linear activation function is best for regression and Sigmoid is best for Binary Classification

In [39]:
```
model_1 = Sequential()
model_1.add(Dense(64, activation='relu', input_shape=(X_train.shape[1],)))
model_1.add(Dense(64, activation='relu'))
model_1.add(Dense(1))
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: U
serWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. Wh
en using Sequential models, prefer using an `Input(shape)` object as the fir
st layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

8. Compile your Neural Network

- Choose Adam as the optimizer
- And MSE as the Loss function
- Also add the following metrics: Mean Absolute Error

In [40]:
```
model_1.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae']
```

9. Fit (or train) your model

- Use the X_train and y_train datasets for the training
- Do 50 data iterations
- Choose the batch size = 10
- Also select a validation_split of 0.2
- Save the result of the fit function in a variable called 'history'

In [41]:
```
history_1 = model_1.fit(X_train, y_train, validation_split=0.2, epochs=100,
```

```
Epoch 1/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.8399 - mae: 0.6876 - val_
loss: 0.1180 - val_mae: 0.2731
Epoch 2/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0879 - mae: 0.2376 - val_
loss: 0.0727 - val_mae: 0.2170
Epoch 3/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0685 - mae: 0.2127 - val_
loss: 0.0613 - val_mae: 0.1996
Epoch 4/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0516 - mae: 0.1833 - val_
loss: 0.0619 - val_mae: 0.2011
Epoch 5/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0459 - mae: 0.1705 - val_
loss: 0.0575 - val_mae: 0.1918
Epoch 6/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0399 - mae: 0.1617 - val_
loss: 0.0483 - val_mae: 0.1789
Epoch 7/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0370 - mae: 0.1530 - val_
loss: 0.0485 - val_mae: 0.1809
Epoch 8/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0337 - mae: 0.1486 - val_
loss: 0.0487 - val_mae: 0.1766
Epoch 9/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0302 - mae: 0.1371 - val_
loss: 0.0513 - val_mae: 0.1838
Epoch 10/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0309 - mae: 0.1420 - val_
loss: 0.0497 - val_mae: 0.1773
Epoch 11/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0255 - mae: 0.1270 - val_
loss: 0.0495 - val_mae: 0.1782
Epoch 12/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0269 - mae: 0.1333 - val_
loss: 0.0517 - val_mae: 0.1793
Epoch 13/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0230 - mae: 0.1196 - val_
loss: 0.0496 - val_mae: 0.1764
Epoch 14/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0277 - mae: 0.1340 - val_
loss: 0.0486 - val_mae: 0.1752
Epoch 15/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0222 - mae: 0.1169 - val_
loss: 0.0454 - val_mae: 0.1706
Epoch 16/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0214 - mae: 0.1168 - val_
loss: 0.0471 - val_mae: 0.1745
Epoch 17/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0232 - mae: 0.1206 - val_
loss: 0.0453 - val_mae: 0.1698
Epoch 18/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0216 - mae: 0.1160 - val_
loss: 0.0446 - val_mae: 0.1672
Epoch 19/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0185 - mae: 0.1073 - val_
```

```
loss: 0.0479 - val_mae: 0.1733
Epoch 20/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0197 - mae: 0.1109 - val_
loss: 0.0479 - val_mae: 0.1749
Epoch 21/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0197 - mae: 0.1104 - val_
loss: 0.0445 - val_mae: 0.1689
Epoch 22/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0174 - mae: 0.1047 - val_
loss: 0.0466 - val_mae: 0.1707
Epoch 23/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0163 - mae: 0.1004 - val_
loss: 0.0440 - val_mae: 0.1690
Epoch 24/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0156 - mae: 0.0981 - val_
loss: 0.0479 - val_mae: 0.1747
Epoch 25/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0162 - mae: 0.0988 - val_
loss: 0.0463 - val_mae: 0.1707
Epoch 26/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0151 - mae: 0.0993 - val_
loss: 0.0510 - val_mae: 0.1776
Epoch 27/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0155 - mae: 0.0993 - val_
loss: 0.0547 - val_mae: 0.1843
Epoch 28/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0168 - mae: 0.1031 - val_
loss: 0.0517 - val_mae: 0.1783
Epoch 29/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0149 - mae: 0.0963 - val_
loss: 0.0495 - val_mae: 0.1767
Epoch 30/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0141 - mae: 0.0951 - val_
loss: 0.0479 - val_mae: 0.1744
Epoch 31/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0136 - mae: 0.0921 - val_
loss: 0.0502 - val_mae: 0.1772
Epoch 32/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0133 - mae: 0.0902 - val_
loss: 0.0532 - val_mae: 0.1825
Epoch 33/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0115 - mae: 0.0842 - val_
loss: 0.0525 - val_mae: 0.1797
Epoch 34/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0135 - mae: 0.0911 - val_
loss: 0.0536 - val_mae: 0.1816
Epoch 35/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0120 - mae: 0.0878 - val_
loss: 0.0511 - val_mae: 0.1795
Epoch 36/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0114 - mae: 0.0834 - val_
loss: 0.0534 - val_mae: 0.1824
Epoch 37/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0116 - mae: 0.0859 - val_
loss: 0.0523 - val_mae: 0.1788
Epoch 38/100
```

```
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0108 - mae: 0.0826 - val_
loss: 0.0526 - val_mae: 0.1804
Epoch 39/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0108 - mae: 0.0829 - val_
loss: 0.0541 - val_mae: 0.1831
Epoch 40/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0106 - mae: 0.0804 - val_
loss: 0.0570 - val_mae: 0.1849
Epoch 41/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0106 - mae: 0.0825 - val_
loss: 0.0547 - val_mae: 0.1853
Epoch 42/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0097 - mae: 0.0776 - val_
loss: 0.0534 - val_mae: 0.1834
Epoch 43/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0095 - mae: 0.0779 - val_
loss: 0.0562 - val_mae: 0.1854
Epoch 44/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0095 - mae: 0.0770 - val_
loss: 0.0620 - val_mae: 0.1956
Epoch 45/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0118 - mae: 0.0867 - val_
loss: 0.0529 - val_mae: 0.1777
Epoch 46/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0099 - mae: 0.0786 - val_
loss: 0.0545 - val_mae: 0.1848
Epoch 47/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0093 - mae: 0.0758 - val_
loss: 0.0544 - val_mae: 0.1827
Epoch 48/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0085 - mae: 0.0727 - val_
loss: 0.0555 - val_mae: 0.1872
Epoch 49/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.0088 - mae: 0.0734 - val_
loss: 0.0564 - val_mae: 0.1859
Epoch 50/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.0083 - mae: 0.0723 - val_
loss: 0.0586 - val_mae: 0.1921
Epoch 51/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0092 - mae: 0.0765 - val_
loss: 0.0584 - val_mae: 0.1874
Epoch 52/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0098 - mae: 0.0783 - val_
loss: 0.0587 - val_mae: 0.1919
Epoch 53/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0086 - mae: 0.0732 - val_
loss: 0.0553 - val_mae: 0.1842
Epoch 54/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0079 - mae: 0.0708 - val_
loss: 0.0598 - val_mae: 0.1896
Epoch 55/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0088 - mae: 0.0746 - val_
loss: 0.0564 - val_mae: 0.1873
Epoch 56/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0071 - mae: 0.0679 - val_
loss: 0.0553 - val_mae: 0.1850
```

```
Epoch 57/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0072 - mae: 0.0670 - val_
loss: 0.0597 - val_mae: 0.1898
Epoch 58/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0080 - mae: 0.0691 - val_
loss: 0.0599 - val_mae: 0.1920
Epoch 59/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0083 - mae: 0.0730 - val_
loss: 0.0590 - val_mae: 0.1885
Epoch 60/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0073 - mae: 0.0678 - val_
loss: 0.0583 - val_mae: 0.1889
Epoch 61/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0071 - mae: 0.0676 - val_
loss: 0.0587 - val_mae: 0.1891
Epoch 62/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0079 - mae: 0.0699 - val_
loss: 0.0589 - val_mae: 0.1925
Epoch 63/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0057 - mae: 0.0594 - val_
loss: 0.0570 - val_mae: 0.1857
Epoch 64/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0069 - mae: 0.0656 - val_
loss: 0.0583 - val_mae: 0.1898
Epoch 65/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0064 - mae: 0.0629 - val_
loss: 0.0673 - val_mae: 0.2022
Epoch 66/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0070 - mae: 0.0657 - val_
loss: 0.0625 - val_mae: 0.1966
Epoch 67/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0064 - mae: 0.0632 - val_
loss: 0.0590 - val_mae: 0.1896
Epoch 68/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0055 - mae: 0.0586 - val_
loss: 0.0614 - val_mae: 0.1928
Epoch 69/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0056 - mae: 0.0591 - val_
loss: 0.0581 - val_mae: 0.1894
Epoch 70/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0061 - mae: 0.0615 - val_
loss: 0.0584 - val_mae: 0.1916
Epoch 71/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0065 - mae: 0.0630 - val_
loss: 0.0642 - val_mae: 0.1971
Epoch 72/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0061 - mae: 0.0630 - val_
loss: 0.0578 - val_mae: 0.1918
Epoch 73/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0059 - mae: 0.0603 - val_
loss: 0.0613 - val_mae: 0.1946
Epoch 74/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0063 - mae: 0.0628 - val_
loss: 0.0609 - val_mae: 0.1937
Epoch 75/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0054 - mae: 0.0581 - val_
```

```
loss: 0.0630 - val_mae: 0.1963
Epoch 76/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0060 - mae: 0.0613 - val_
loss: 0.0617 - val_mae: 0.1970
Epoch 77/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0067 - mae: 0.0647 - val_
loss: 0.0582 - val_mae: 0.1909
Epoch 78/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0056 - mae: 0.0587 - val_
loss: 0.0663 - val_mae: 0.2016
Epoch 79/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0089 - mae: 0.0732 - val_
loss: 0.0649 - val_mae: 0.1985
Epoch 80/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0066 - mae: 0.0642 - val_
loss: 0.0591 - val_mae: 0.1922
Epoch 81/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.0060 - mae: 0.0618 - val_
loss: 0.0592 - val_mae: 0.1917
Epoch 82/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0051 - mae: 0.0563 - val_
loss: 0.0622 - val_mae: 0.1954
Epoch 83/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0049 - mae: 0.0550 - val_
loss: 0.0610 - val_mae: 0.1943
Epoch 84/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0046 - mae: 0.0535 - val_
loss: 0.0585 - val_mae: 0.1930
Epoch 85/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0048 - mae: 0.0554 - val_
loss: 0.0574 - val_mae: 0.1893
Epoch 86/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0050 - mae: 0.0554 - val_
loss: 0.0616 - val_mae: 0.1936
Epoch 87/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0041 - mae: 0.0501 - val_
loss: 0.0606 - val_mae: 0.1929
Epoch 88/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0048 - mae: 0.0554 - val_
loss: 0.0610 - val_mae: 0.1971
Epoch 89/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0065 - mae: 0.0626 - val_
loss: 0.0581 - val_mae: 0.1911
Epoch 90/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0052 - mae: 0.0570 - val_
loss: 0.0612 - val_mae: 0.1949
Epoch 91/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0050 - mae: 0.0554 - val_
loss: 0.0626 - val_mae: 0.1970
Epoch 92/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0053 - mae: 0.0572 - val_
loss: 0.0602 - val_mae: 0.1944
Epoch 93/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0039 - mae: 0.0486 - val_
loss: 0.0611 - val_mae: 0.1951
Epoch 94/100
```

```
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0044 - mae: 0.0517 - val_
loss: 0.0667 - val_mae: 0.2028
Epoch 95/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0062 - mae: 0.0615 - val_
loss: 0.0598 - val_mae: 0.1934
Epoch 96/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0048 - mae: 0.0552 - val_
loss: 0.0602 - val_mae: 0.1953
Epoch 97/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0041 - mae: 0.0507 - val_
loss: 0.0614 - val_mae: 0.1949
Epoch 98/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0043 - mae: 0.0509 - val_
loss: 0.0603 - val_mae: 0.1950
Epoch 99/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.0047 - mae: 0.0542 - val_
loss: 0.0630 - val_mae: 0.1973
Epoch 100/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0044 - mae: 0.0527 - val_
loss: 0.0618 - val_mae: 0.1945
```

10. View your history variable:

- Use Matplotlib.pyplot to show graphs of your model traning history
- In one graph:
  - Plot the Training Loss and the Validation Loss
  - X Label = Epochs
  - Y Label = Loss
  - Title = Training and Validation Loss over Epochs
- In a second graph:
  - Plot the Training MAE and the Validation MAE
  - X Label = Epochs
  - Y Label = Mean Absolute Error (MAE)
  - Title = Training and Validation MAE over Epochs

In [42]:

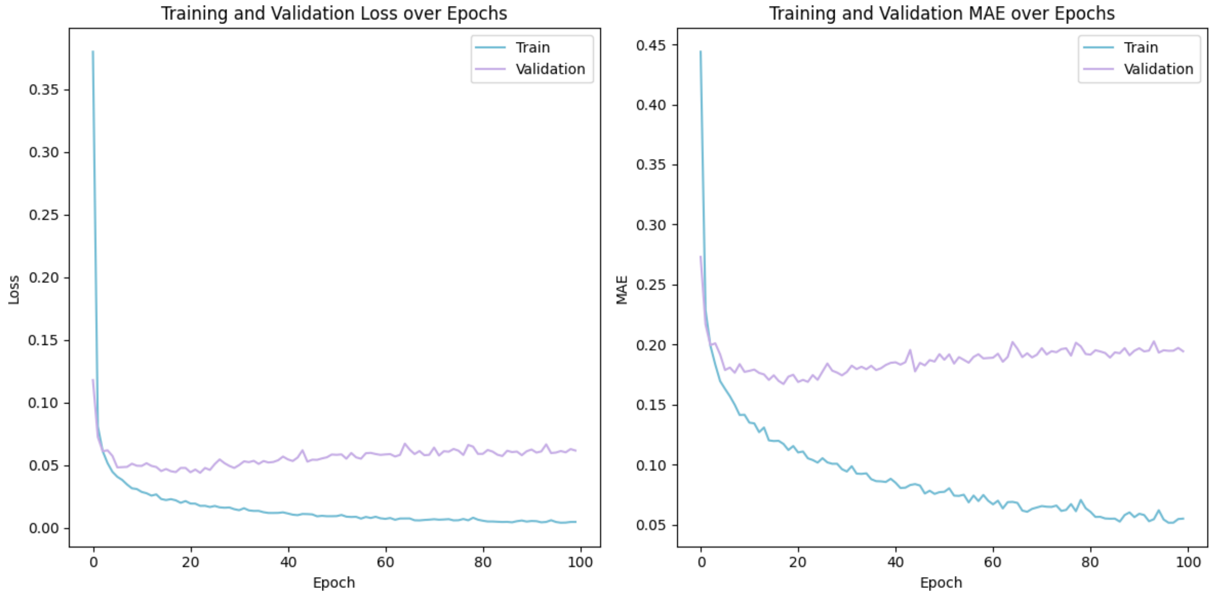
```python
plt.figure(figsize=(12, 6))

# Plot de valores de las pérdidas de entrenamiento y validación
plt.subplot(1, 2, 1)
plt.plot(history_1.history['loss'], color='#72bcd4')
plt.plot(history_1.history['val_loss'], color='#c6ade6')
plt.title('Training and Validation Loss over Epochs')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')

# Plot de formación y validación valores mae
plt.subplot(1, 2, 2)
plt.plot(history_1.history['mae'], color='#72bcd4')
plt.plot(history_1.history['val_mae'], color='#c6ade6')
plt.title('Training and Validation MAE over Epochs')
plt.ylabel('MAE')
```

```
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')

plt.tight_layout()
plt.show()
```



11. Evaluate your model:

- See the result of your loss function.
- What can you deduct from there?

In [43]:
```
test_loss, test_mae = model_1.evaluate(X_test, y_test, verbose=0)
print(f'Test loss: {test_loss}')
print(f'Test MAE: {test_mae}')
```

```
Test loss: 0.060247279703617096
Test MAE: 0.19004300236701965
```

12. Use your model to make some predictions:

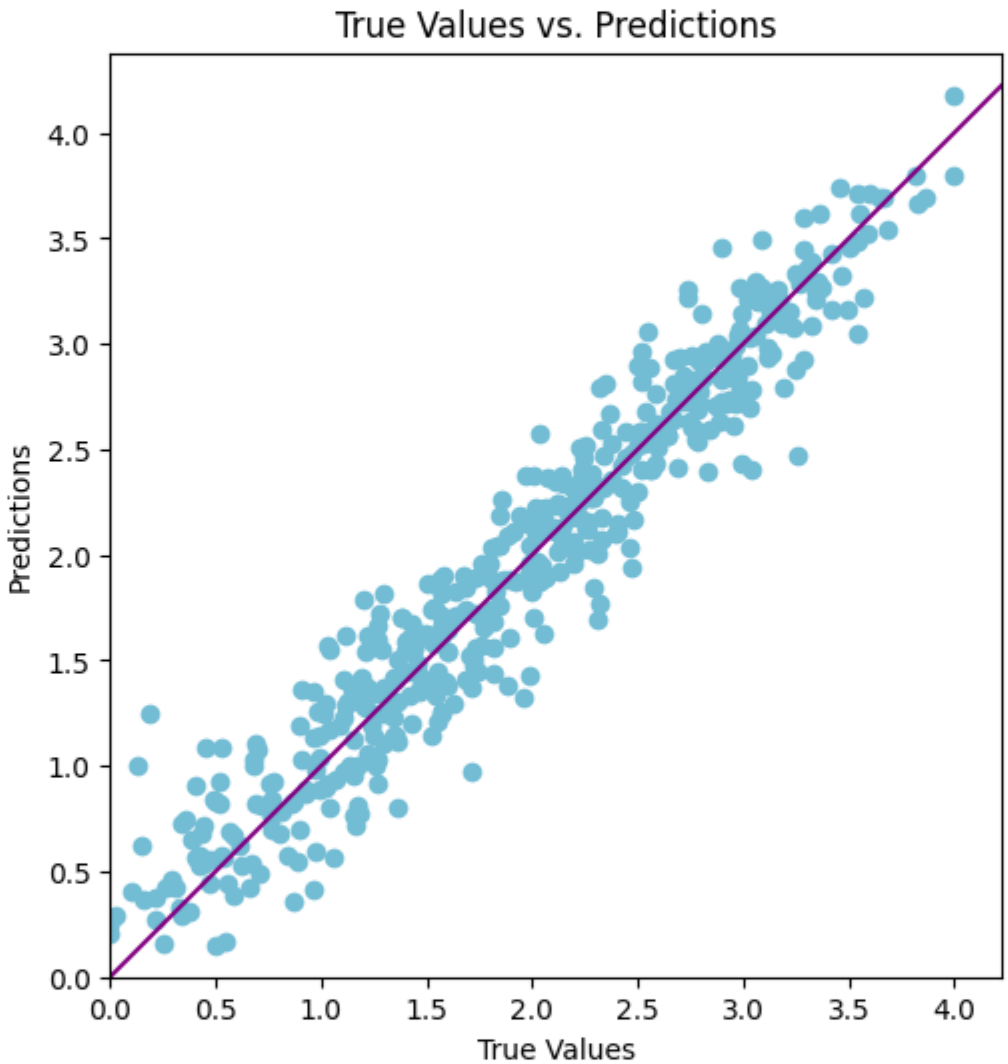
- Make predictions of your X_test dataset
- Print the each of the predictions and the actual value (which is in y_test)
- How good was your model?

In [44]:
```
y_pred = model_1.predict(X_test) # Predicciones

# Plot
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, color='#72bcd4')
plt.xlabel('True Values')
plt.ylabel('Predictions')
plt.title('True Values vs. Predictions')
plt.axis('equal')
plt.axis('square')
plt.xlim([0, plt.xlim()[1]])
```

```
plt.ylim([0, plt.ylim()[1]])
_ = plt.plot([-100, 100], [-100, 100], color='purple')
plt.show()
```

```
15/15 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
```



13. Compete against this model:

- Create two more different models to compete with this model
- Here are a few ideas of things you can change:
  - During Dataset data engineering:
    - You can remove features that you think do not help in the training and prediction
    - Feature Scaling: Ensure all features are on a similar scale (as you already did with StandardScaler)
  - During Model Definition:
    - You can change the Model Architecture (change the type or number of layers or the number of units)
    - You can add dropout layers to prevent overfitting
  - During Model Compile:

      - You can try other optimizer when compiling your model, here some optimizer samples: Adam, RMSprop, or Adagrad.
      - Try another Loss Function
    - During Model Training:
      - Encrease the number of Epochs
      - Adjust the size of your batch
- Explain in a Markdown cell which changes are you implementing
- Show the comparison of your model versus the original model

Model 2:

- Changes:
  - Dataset Data Engineering
  - Model Definition
  - Model Compile
  - Model Training

In [45]:
```
model_2 = Sequential()
model_2.add(Dense(16, activation='relu', input_shape=(X_train.shape[1], 1)))
model_2.add(Dense(8, activation='relu'))
model_2.add(Dense(1))

model_2.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae']
history_2 = model_2.fit(X_train, y_train, validation_split=0.2, epochs=100,
plt.figure(figsize=(12, 6))

# Plot training & validation valores de pérdida
plt.subplot(1, 2, 1)
plt.plot(history_2.history['loss'], color='#72bcd4')
plt.plot(history_2.history['val_loss'], color='#c6ade6')
plt.title('Training and Validation Loss over Epochs')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')

# Plot training & validation valores mae
plt.subplot(1, 2, 2)
plt.plot(history_2.history['mae'], color='#72bcd4')
plt.plot(history_2.history['val_mae'], color='#c6ade6')
plt.title('Training and Validation MAE over Epochs')
plt.ylabel('MAE')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')

plt.tight_layout()
plt.show()
```

```
Epoch 1/100
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/dense.py:87: U
serWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. Wh
en using Sequential models, prefer using an `Input(shape)` object as the fir
st layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

```
153/153 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - loss: 2.5537 - mae: 1.3429 - val_
loss: 1.0951 - val_mae: 0.8656
Epoch 2/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.9882 - mae: 0.8207 - val_
loss: 0.9582 - val_mae: 0.8200
Epoch 3/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.8660 - mae: 0.7774 - val_
loss: 0.9094 - val_mae: 0.8045
Epoch 4/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.8542 - mae: 0.7736 - val_
loss: 0.8991 - val_mae: 0.8033
Epoch 5/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8465 - mae: 0.7763 - val_
loss: 0.8950 - val_mae: 0.8023
Epoch 6/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8182 - mae: 0.7622 - val_
loss: 0.8932 - val_mae: 0.8018
Epoch 7/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.7909 - mae: 0.7353 - val_
loss: 0.8919 - val_mae: 0.8007
Epoch 8/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8581 - mae: 0.7787 - val_
loss: 0.8948 - val_mae: 0.7970
Epoch 9/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8095 - mae: 0.7586 - val_
loss: 0.9074 - val_mae: 0.8108
Epoch 10/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8064 - mae: 0.7513 - val_
loss: 0.8938 - val_mae: 0.8028
Epoch 11/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8325 - mae: 0.7638 - val_
loss: 0.8900 - val_mae: 0.8007
Epoch 12/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8220 - mae: 0.7593 - val_
loss: 0.8920 - val_mae: 0.8027
Epoch 13/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.7943 - mae: 0.7455 - val_
loss: 0.9058 - val_mae: 0.8097
Epoch 14/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7993 - mae: 0.7504 - val_
loss: 0.8896 - val_mae: 0.8010
Epoch 15/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8258 - mae: 0.7581 - val_
loss: 0.8887 - val_mae: 0.7985
Epoch 16/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8072 - mae: 0.7498 - val_
loss: 0.9031 - val_mae: 0.8083
Epoch 17/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8009 - mae: 0.7538 - val_
loss: 0.8990 - val_mae: 0.8062
Epoch 18/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8001 - mae: 0.7447 - val_
loss: 0.8891 - val_mae: 0.8002
Epoch 19/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7841 - mae: 0.7437 - val_
loss: 0.9059 - val_mae: 0.8092
```

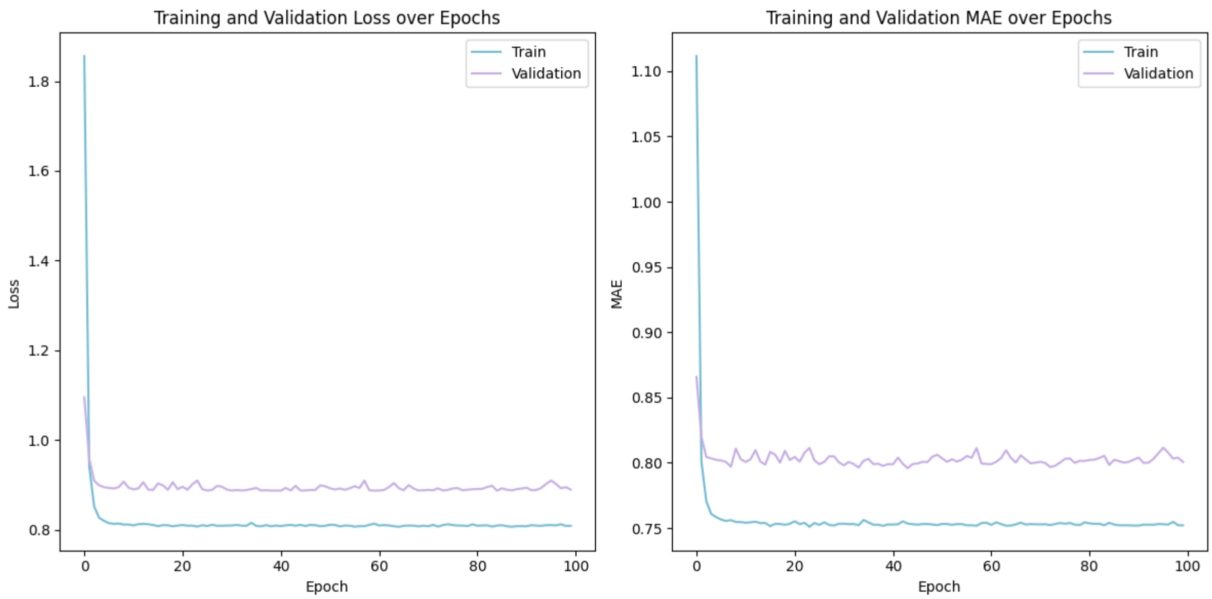
```
Epoch 20/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8106 - mae: 0.7500 - val_
loss: 0.8905 - val_mae: 0.8021
Epoch 21/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8146 - mae: 0.7558 - val_
loss: 0.8958 - val_mae: 0.8045
Epoch 22/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7857 - mae: 0.7427 - val_
loss: 0.8892 - val_mae: 0.8008
Epoch 23/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.7860 - mae: 0.7374 - val_
loss: 0.9013 - val_mae: 0.8078
Epoch 24/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7891 - mae: 0.7439 - val_
loss: 0.9094 - val_mae: 0.8114
Epoch 25/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8262 - mae: 0.7597 - val_
loss: 0.8905 - val_mae: 0.8018
Epoch 26/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.8329 - mae: 0.7659 - val_
loss: 0.8874 - val_mae: 0.7989
Epoch 27/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.7896 - mae: 0.7473 - val_
loss: 0.8887 - val_mae: 0.8007
Epoch 28/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.8147 - mae: 0.7593 - val_
loss: 0.8975 - val_mae: 0.8050
Epoch 29/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.8224 - mae: 0.7597 - val_
loss: 0.8958 - val_mae: 0.8049
Epoch 30/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.8380 - mae: 0.7698 - val_
loss: 0.8896 - val_mae: 0.8003
Epoch 31/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7814 - mae: 0.7376 - val_
loss: 0.8873 - val_mae: 0.7980
Epoch 32/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8209 - mae: 0.7628 - val_
loss: 0.8885 - val_mae: 0.8007
Epoch 33/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8016 - mae: 0.7420 - val_
loss: 0.8874 - val_mae: 0.7990
Epoch 34/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8116 - mae: 0.7536 - val_
loss: 0.8883 - val_mae: 0.7964
Epoch 35/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8191 - mae: 0.7581 - val_
loss: 0.8908 - val_mae: 0.8015
Epoch 36/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7834 - mae: 0.7352 - val_
loss: 0.8930 - val_mae: 0.8030
Epoch 37/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7905 - mae: 0.7469 - val_
loss: 0.8874 - val_mae: 0.7990
Epoch 38/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8253 - mae: 0.7583 - val_
```

```
loss: 0.8880 - val_mae: 0.7993
Epoch 39/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7809 - mae: 0.7422 - val_
loss: 0.8873 - val_mae: 0.7976
Epoch 40/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8172 - mae: 0.7539 - val_
loss: 0.8872 - val_mae: 0.7990
Epoch 41/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8158 - mae: 0.7591 - val_
loss: 0.8872 - val_mae: 0.7989
Epoch 42/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7994 - mae: 0.7503 - val_
loss: 0.8931 - val_mae: 0.8039
Epoch 43/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8241 - mae: 0.7665 - val_
loss: 0.8875 - val_mae: 0.7992
Epoch 44/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8365 - mae: 0.7644 - val_
loss: 0.8978 - val_mae: 0.7959
Epoch 45/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8048 - mae: 0.7434 - val_
loss: 0.8874 - val_mae: 0.7991
Epoch 46/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7961 - mae: 0.7455 - val_
loss: 0.8876 - val_mae: 0.7994
Epoch 47/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7938 - mae: 0.7494 - val_
loss: 0.8885 - val_mae: 0.8008
Epoch 48/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8007 - mae: 0.7515 - val_
loss: 0.8886 - val_mae: 0.8006
Epoch 49/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8011 - mae: 0.7453 - val_
loss: 0.8988 - val_mae: 0.8046
Epoch 50/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8404 - mae: 0.7697 - val_
loss: 0.8969 - val_mae: 0.8062
Epoch 51/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8215 - mae: 0.7636 - val_
loss: 0.8924 - val_mae: 0.8034
Epoch 52/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8136 - mae: 0.7486 - val_
loss: 0.8897 - val_mae: 0.8008
Epoch 53/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.7992 - mae: 0.7487 - val_
loss: 0.8919 - val_mae: 0.8027
Epoch 54/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7826 - mae: 0.7370 - val_
loss: 0.8890 - val_mae: 0.8010
Epoch 55/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8417 - mae: 0.7704 - val_
loss: 0.8921 - val_mae: 0.8023
Epoch 56/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.8165 - mae: 0.7556 - val_
loss: 0.8970 - val_mae: 0.8052
Epoch 57/100
```

```
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.8076 - mae: 0.7498 - val_
loss: 0.8930 - val_mae: 0.8039
Epoch 58/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.7972 - mae: 0.7462 - val_
loss: 0.9096 - val_mae: 0.8112
Epoch 59/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.8204 - mae: 0.7601 - val_
loss: 0.8880 - val_mae: 0.7995
Epoch 60/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8359 - mae: 0.7644 - val_
loss: 0.8873 - val_mae: 0.7991
Epoch 61/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8306 - mae: 0.7695 - val_
loss: 0.8875 - val_mae: 0.7989
Epoch 62/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.7892 - mae: 0.7420 - val_
loss: 0.8885 - val_mae: 0.8007
Epoch 63/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8271 - mae: 0.7652 - val_
loss: 0.8950 - val_mae: 0.8037
Epoch 64/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8193 - mae: 0.7598 - val_
loss: 0.9041 - val_mae: 0.8096
Epoch 65/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8363 - mae: 0.7678 - val_
loss: 0.8928 - val_mae: 0.8037
Epoch 66/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8009 - mae: 0.7505 - val_
loss: 0.8881 - val_mae: 0.8002
Epoch 67/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8264 - mae: 0.7674 - val_
loss: 0.8993 - val_mae: 0.8057
Epoch 68/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8089 - mae: 0.7538 - val_
loss: 0.8921 - val_mae: 0.8026
Epoch 69/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8068 - mae: 0.7561 - val_
loss: 0.8874 - val_mae: 0.7996
Epoch 70/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.7904 - mae: 0.7456 - val_
loss: 0.8879 - val_mae: 0.8001
Epoch 71/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8006 - mae: 0.7519 - val_
loss: 0.8888 - val_mae: 0.8007
Epoch 72/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8164 - mae: 0.7607 - val_
loss: 0.8879 - val_mae: 0.7998
Epoch 73/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8128 - mae: 0.7471 - val_
loss: 0.8922 - val_mae: 0.7967
Epoch 74/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8102 - mae: 0.7565 - val_
loss: 0.8874 - val_mae: 0.7977
Epoch 75/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8201 - mae: 0.7586 - val_
loss: 0.8889 - val_mae: 0.8002
```

```
Epoch 76/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8145 - mae: 0.7545 - val_
loss: 0.8919 - val_mae: 0.8031
Epoch 77/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8148 - mae: 0.7585 - val_
loss: 0.8927 - val_mae: 0.8033
Epoch 78/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8491 - mae: 0.7736 - val_
loss: 0.8879 - val_mae: 0.7999
Epoch 79/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7928 - mae: 0.7480 - val_
loss: 0.8892 - val_mae: 0.8015
Epoch 80/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8099 - mae: 0.7514 - val_
loss: 0.8901 - val_mae: 0.8014
Epoch 81/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8327 - mae: 0.7706 - val_
loss: 0.8908 - val_mae: 0.8022
Epoch 82/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.8120 - mae: 0.7536 - val_
loss: 0.8910 - val_mae: 0.8023
Epoch 83/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.8046 - mae: 0.7550 - val_
loss: 0.8952 - val_mae: 0.8037
Epoch 84/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.8322 - mae: 0.7692 - val_
loss: 0.8980 - val_mae: 0.8053
Epoch 85/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.8198 - mae: 0.7605 - val_
loss: 0.8870 - val_mae: 0.7983
Epoch 86/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7877 - mae: 0.7373 - val_
loss: 0.8922 - val_mae: 0.8023
Epoch 87/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8061 - mae: 0.7480 - val_
loss: 0.8892 - val_mae: 0.8012
Epoch 88/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.7710 - mae: 0.7303 - val_
loss: 0.8878 - val_mae: 0.8001
Epoch 89/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.7953 - mae: 0.7433 - val_
loss: 0.8903 - val_mae: 0.8008
Epoch 90/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8052 - mae: 0.7549 - val_
loss: 0.8917 - val_mae: 0.8024
Epoch 91/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8387 - mae: 0.7640 - val_
loss: 0.8937 - val_mae: 0.8039
Epoch 92/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8147 - mae: 0.7597 - val_
loss: 0.8881 - val_mae: 0.7999
Epoch 93/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.7762 - mae: 0.7328 - val_
loss: 0.8886 - val_mae: 0.8001
Epoch 94/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8121 - mae: 0.7502 - val_
```

```
loss: 0.8933 - val_mae: 0.8030
Epoch 95/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8090 - mae: 0.7548 - val_
loss: 0.9021 - val_mae: 0.8073
Epoch 96/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8420 - mae: 0.7761 - val_
loss: 0.9097 - val_mae: 0.8115
Epoch 97/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8276 - mae: 0.7601 - val_
loss: 0.9019 - val_mae: 0.8077
Epoch 98/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.7980 - mae: 0.7464 - val_
loss: 0.8925 - val_mae: 0.8033
Epoch 99/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - loss: 0.8077 - mae: 0.7463 - val_
loss: 0.8948 - val_mae: 0.8039
Epoch 100/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.8483 - mae: 0.7732 - val_
loss: 0.8893 - val_mae: 0.8007
```



Model 3:

- Changes:
  - Dataset Data Engineering
  - Model Definition
  - Model Compile
  - Model Training

In [46]:
```python
model_3 = Sequential()
model_3.add(SimpleRNN(64, activation='tanh', input_shape=(X_train.shape[1],
model_3.add(Dense(64, activation='tanh'))
model_3.add(Dense(1))
model_3.compile(optimizer='adam', loss='mean_squared_error', metrics=['mae']
history_3 = model_3.fit(X_train, y_train, validation_split=0.2, epochs=100,

plt.figure(figsize=(12, 6))
```

```
# Plot training & validation valores de pérdida
plt.subplot(1, 2, 1)
plt.plot(history_3.history['loss'], color='#72bcd4')
plt.plot(history_3.history['val_loss'], color='#c6ade6')
plt.title('Training and Validation Loss over Epochs')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')

# Plot training & validation valores mae
plt.subplot(1, 2, 2)
plt.plot(history_3.history['mae'], color='#72bcd4')
plt.plot(history_3.history['val_mae'], color='#c6ade6')
plt.title('Training and Validation MAE over Epochs')
plt.ylabel('MAE')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper right')

plt.tight_layout()
plt.show()
```

```
Epoch 1/100
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/rnn/rnn.py:204: Use
rWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When
using Sequential models, prefer using an `Input(shape)` object as the first
layer in the model instead.
  super().__init__(**kwargs)
```

```
153/153 ━━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - loss: 1.4747 - mae: 0.8469 - val_
loss: 0.0672 - val_mae: 0.2042
Epoch 2/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.0655 - mae: 0.2021 - val_
loss: 0.0625 - val_mae: 0.1976
Epoch 3/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - loss: 0.0621 - mae: 0.1965 - val_
loss: 0.0599 - val_mae: 0.1958
Epoch 4/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.0597 - mae: 0.1946 - val_
loss: 0.0514 - val_mae: 0.1879
Epoch 5/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0594 - mae: 0.1936 - val_
loss: 0.0630 - val_mae: 0.1977
Epoch 6/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0472 - mae: 0.1711 - val_
loss: 0.0561 - val_mae: 0.1928
Epoch 7/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0476 - mae: 0.1752 - val_
loss: 0.0775 - val_mae: 0.2233
Epoch 8/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0488 - mae: 0.1761 - val_
loss: 0.0557 - val_mae: 0.1904
Epoch 9/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0487 - mae: 0.1772 - val_
loss: 0.0519 - val_mae: 0.1824
Epoch 10/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0441 - mae: 0.1645 - val_
loss: 0.0561 - val_mae: 0.1919
Epoch 11/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0495 - mae: 0.1748 - val_
loss: 0.0459 - val_mae: 0.1709
Epoch 12/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0429 - mae: 0.1658 - val_
loss: 0.0545 - val_mae: 0.1881
Epoch 13/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0433 - mae: 0.1652 - val_
loss: 0.0498 - val_mae: 0.1781
Epoch 14/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0438 - mae: 0.1652 - val_
loss: 0.0555 - val_mae: 0.1870
Epoch 15/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0448 - mae: 0.1652 - val_
loss: 0.0439 - val_mae: 0.1677
Epoch 16/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0400 - mae: 0.1576 - val_
loss: 0.0489 - val_mae: 0.1752
Epoch 17/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0415 - mae: 0.1604 - val_
loss: 0.0557 - val_mae: 0.1902
Epoch 18/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0413 - mae: 0.1586 - val_
loss: 0.0612 - val_mae: 0.1982
Epoch 19/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0546 - mae: 0.1802 - val_
loss: 0.0525 - val_mae: 0.1819
```

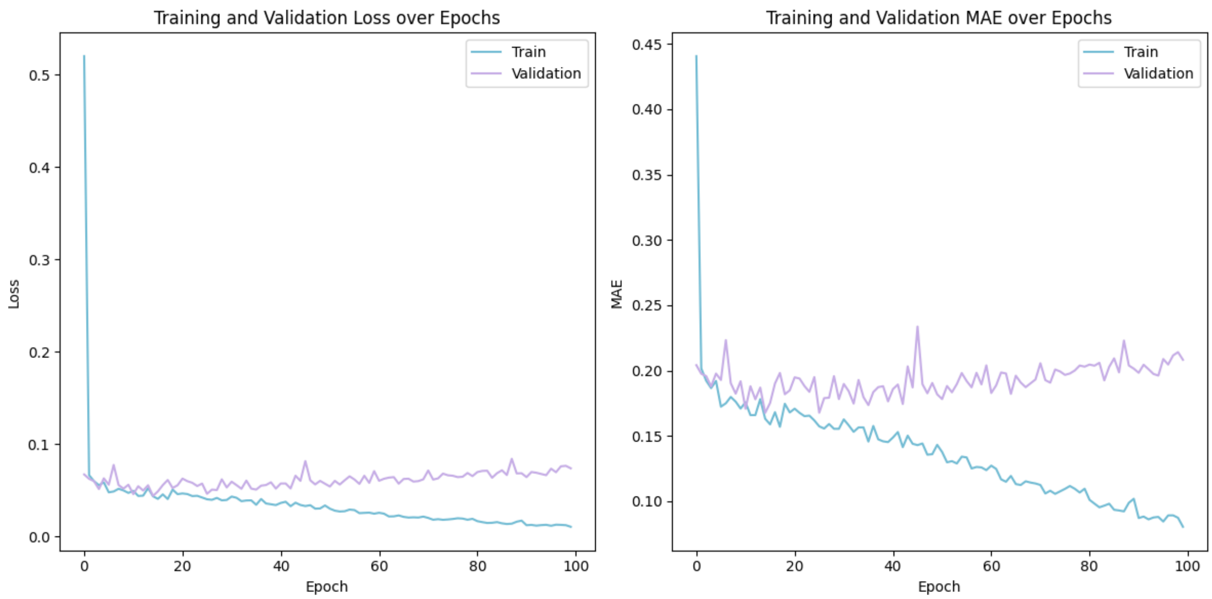
```
Epoch 20/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0426 - mae: 0.1629 - val_
loss: 0.0555 - val_mae: 0.1849
Epoch 21/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0467 - mae: 0.1706 - val_
loss: 0.0627 - val_mae: 0.1949
Epoch 22/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.0433 - mae: 0.1613 - val_
loss: 0.0599 - val_mae: 0.1939
Epoch 23/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.0416 - mae: 0.1625 - val_
loss: 0.0581 - val_mae: 0.1882
Epoch 24/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.0421 - mae: 0.1645 - val_
loss: 0.0547 - val_mae: 0.1838
Epoch 25/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0398 - mae: 0.1580 - val_
loss: 0.0573 - val_mae: 0.1950
Epoch 26/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0391 - mae: 0.1544 - val_
loss: 0.0463 - val_mae: 0.1678
Epoch 27/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0344 - mae: 0.1447 - val_
loss: 0.0508 - val_mae: 0.1790
Epoch 28/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0419 - mae: 0.1604 - val_
loss: 0.0503 - val_mae: 0.1793
Epoch 29/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0385 - mae: 0.1535 - val_
loss: 0.0619 - val_mae: 0.1958
Epoch 30/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0428 - mae: 0.1647 - val_
loss: 0.0531 - val_mae: 0.1782
Epoch 31/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0366 - mae: 0.1494 - val_
loss: 0.0594 - val_mae: 0.1898
Epoch 32/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0417 - mae: 0.1550 - val_
loss: 0.0556 - val_mae: 0.1843
Epoch 33/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0415 - mae: 0.1617 - val_
loss: 0.0517 - val_mae: 0.1748
Epoch 34/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0387 - mae: 0.1566 - val_
loss: 0.0605 - val_mae: 0.1929
Epoch 35/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0373 - mae: 0.1541 - val_
loss: 0.0519 - val_mae: 0.1797
Epoch 36/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0375 - mae: 0.1518 - val_
loss: 0.0508 - val_mae: 0.1736
Epoch 37/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0381 - mae: 0.1528 - val_
loss: 0.0551 - val_mae: 0.1836
Epoch 38/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0387 - mae: 0.1520 - val_
```

```
loss: 0.0558 - val_mae: 0.1874
Epoch 39/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0380 - mae: 0.1512 - val_
loss: 0.0587 - val_mae: 0.1881
Epoch 40/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0329 - mae: 0.1452 - val_
loss: 0.0519 - val_mae: 0.1765
Epoch 41/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.0345 - mae: 0.1442 - val_
loss: 0.0574 - val_mae: 0.1859
Epoch 42/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.0429 - mae: 0.1602 - val_
loss: 0.0574 - val_mae: 0.1894
Epoch 43/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.0314 - mae: 0.1391 - val_
loss: 0.0520 - val_mae: 0.1745
Epoch 44/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.0302 - mae: 0.1358 - val_
loss: 0.0658 - val_mae: 0.2033
Epoch 45/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - loss: 0.0337 - mae: 0.1433 - val_
loss: 0.0603 - val_mae: 0.1872
Epoch 46/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0319 - mae: 0.1402 - val_
loss: 0.0817 - val_mae: 0.2337
Epoch 47/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0361 - mae: 0.1504 - val_
loss: 0.0609 - val_mae: 0.1897
Epoch 48/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0282 - mae: 0.1324 - val_
loss: 0.0564 - val_mae: 0.1827
Epoch 49/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0281 - mae: 0.1302 - val_
loss: 0.0603 - val_mae: 0.1906
Epoch 50/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0344 - mae: 0.1446 - val_
loss: 0.0572 - val_mae: 0.1819
Epoch 51/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0298 - mae: 0.1356 - val_
loss: 0.0541 - val_mae: 0.1782
Epoch 52/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0276 - mae: 0.1305 - val_
loss: 0.0603 - val_mae: 0.1882
Epoch 53/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0267 - mae: 0.1313 - val_
loss: 0.0563 - val_mae: 0.1834
Epoch 54/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0244 - mae: 0.1228 - val_
loss: 0.0609 - val_mae: 0.1897
Epoch 55/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0306 - mae: 0.1377 - val_
loss: 0.0652 - val_mae: 0.1982
Epoch 56/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0285 - mae: 0.1332 - val_
loss: 0.0617 - val_mae: 0.1920
Epoch 57/100
```

```
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0262 - mae: 0.1281 - val_
loss: 0.0570 - val_mae: 0.1872
Epoch 58/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0234 - mae: 0.1200 - val_
loss: 0.0658 - val_mae: 0.1983
Epoch 59/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0238 - mae: 0.1209 - val_
loss: 0.0581 - val_mae: 0.1895
Epoch 60/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0230 - mae: 0.1208 - val_
loss: 0.0708 - val_mae: 0.2040
Epoch 61/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.0263 - mae: 0.1300 - val_
loss: 0.0604 - val_mae: 0.1829
Epoch 62/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.0246 - mae: 0.1246 - val_
loss: 0.0625 - val_mae: 0.1887
Epoch 63/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.0219 - mae: 0.1177 - val_
loss: 0.0638 - val_mae: 0.1986
Epoch 64/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0229 - mae: 0.1181 - val_
loss: 0.0643 - val_mae: 0.1978
Epoch 65/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0210 - mae: 0.1157 - val_
loss: 0.0572 - val_mae: 0.1822
Epoch 66/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0198 - mae: 0.1089 - val_
loss: 0.0623 - val_mae: 0.1961
Epoch 67/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0198 - mae: 0.1112 - val_
loss: 0.0626 - val_mae: 0.1909
Epoch 68/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0207 - mae: 0.1160 - val_
loss: 0.0595 - val_mae: 0.1873
Epoch 69/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0195 - mae: 0.1117 - val_
loss: 0.0600 - val_mae: 0.1903
Epoch 70/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0181 - mae: 0.1054 - val_
loss: 0.0620 - val_mae: 0.1933
Epoch 71/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0194 - mae: 0.1086 - val_
loss: 0.0715 - val_mae: 0.2057
Epoch 72/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0186 - mae: 0.1079 - val_
loss: 0.0617 - val_mae: 0.1928
Epoch 73/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0170 - mae: 0.1019 - val_
loss: 0.0629 - val_mae: 0.1908
Epoch 74/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0166 - mae: 0.1023 - val_
loss: 0.0681 - val_mae: 0.2009
Epoch 75/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0163 - mae: 0.1017 - val_
loss: 0.0663 - val_mae: 0.1994
```

```
Epoch 76/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0180 - mae: 0.1066 - val_
loss: 0.0658 - val_mae: 0.1967
Epoch 77/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0176 - mae: 0.1062 - val_
loss: 0.0643 - val_mae: 0.1978
Epoch 78/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0183 - mae: 0.1050 - val_
loss: 0.0647 - val_mae: 0.2001
Epoch 79/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0186 - mae: 0.1061 - val_
loss: 0.0688 - val_mae: 0.2039
Epoch 80/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0191 - mae: 0.1099 - val_
loss: 0.0654 - val_mae: 0.2030
Epoch 81/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.0157 - mae: 0.0988 - val_
loss: 0.0698 - val_mae: 0.2046
Epoch 82/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.0152 - mae: 0.0967 - val_
loss: 0.0710 - val_mae: 0.2038
Epoch 83/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - loss: 0.0148 - mae: 0.0961 - val_
loss: 0.0712 - val_mae: 0.2060
Epoch 84/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0147 - mae: 0.0958 - val_
loss: 0.0637 - val_mae: 0.1925
Epoch 85/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0146 - mae: 0.0939 - val_
loss: 0.0685 - val_mae: 0.2029
Epoch 86/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0143 - mae: 0.0925 - val_
loss: 0.0717 - val_mae: 0.2094
Epoch 87/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0147 - mae: 0.0973 - val_
loss: 0.0666 - val_mae: 0.1987
Epoch 88/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0119 - mae: 0.0864 - val_
loss: 0.0841 - val_mae: 0.2230
Epoch 89/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0157 - mae: 0.0968 - val_
loss: 0.0684 - val_mae: 0.2039
Epoch 90/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0157 - mae: 0.0972 - val_
loss: 0.0685 - val_mae: 0.2015
Epoch 91/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0129 - mae: 0.0895 - val_
loss: 0.0643 - val_mae: 0.1984
Epoch 92/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0120 - mae: 0.0848 - val_
loss: 0.0698 - val_mae: 0.2045
Epoch 93/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0121 - mae: 0.0874 - val_
loss: 0.0691 - val_mae: 0.2011
Epoch 94/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0113 - mae: 0.0825 - val_
```

```
loss: 0.0677 - val_mae: 0.1975
Epoch 95/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0134 - mae: 0.0893 - val_
loss: 0.0664 - val_mae: 0.1962
Epoch 96/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - loss: 0.0098 - mae: 0.0785 - val_
loss: 0.0734 - val_mae: 0.2089
Epoch 97/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.0138 - mae: 0.0927 - val_
loss: 0.0696 - val_mae: 0.2047
Epoch 98/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - loss: 0.0111 - mae: 0.0832 - val_
loss: 0.0760 - val_mae: 0.2117
Epoch 99/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0128 - mae: 0.0883 - val_
loss: 0.0765 - val_mae: 0.2141
Epoch 100/100
153/153 ━━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0115 - mae: 0.0836 - val_
loss: 0.0739 - val_mae: 0.2083
```



Comparación de los tres modelos

In [47]:
```python
# Evaluar los modelos con los datos de prueba
test_loss_1, test_mae_1 = model_1.evaluate(X_test, y_test, verbose=0)
test_loss_2, test_mae_2 = model_2.evaluate(X_test, y_test, verbose=0)
test_loss_3, test_mae_3 = model_3.evaluate(X_test, y_test, verbose=0)

# Tabla comparativa
comparison_data = {
    'Model': ['Model 1', 'Model 2', 'Model 3'],
    'Test Loss': [test_loss_1, test_loss_2, test_loss_3],
    'Test MAE': [test_mae_1, test_mae_2, test_mae_3]
}

comparison_df = pd.DataFrame(comparison_data)
comparison_df.set_index('Model', inplace=True)
comparison_df
```

Out[47]:

Model	Test Loss	Test MAE
Model 1	0.060247	0.190043
Model 2	0.809933	0.752570
Model 3	0.064943	0.198279

In [48]:

```
students = np.random.choice(data.index, 5, replace=False)

# Obtener los datos de los exámenes de los alumnos seleccionados
X_test_real = data.loc[students].drop('GPA', axis=1)
y_test_real = data.loc[students, 'GPA']
X_test_real = scaler.transform(X_test_real) # Escalar los datos de la prueba

# Hacer predicciones utilizando los tres modelos
predictions1 = model_1.predict(X_test_real).flatten()
predictions2 = model_2.predict(X_test_real).flatten()
predictions3 = model_3.predict(X_test_real).flatten()

comparison_df = pd.DataFrame({
    'Estudiante': [1, 2, 3, 4, 5],
    'Modelo 1': predictions1[:5],
    'Modelo 2': predictions2[:5],
    'Modelo 3': predictions3[:5],
    'Actual': y_test_real[:5]
})

comparison_df.set_index('Estudiante', inplace=True) # Alumno como index
comparison_df
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 21ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 72ms/step
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 142ms/step
```

Out[48]:

Estudiante	Modelo 1	Modelo 2	Modelo 3	Actual
1	2.737846	2.072214	2.718173	2.813806
2	1.018899	1.702772	1.038768	1.182565
3	1.752803	1.928937	1.658082	1.641341
4	0.444845	1.917436	0.277790	0.415931
5	2.497147	1.924489	2.394563	2.215087