```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import tensorflow as tf

# ── 1) Carga el modelo base (.h5) ──
model = tf.keras.models.load_model('celsius_a_fahrenheit.h5')
```

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until y
```

```
# ── 2) Genera 1 000 valores de Celsius y predice con el modelo ──
celsius = np.random.uniform(-200, 200, size=1000)
fahrenheit_pred = model.predict(celsius.reshape(-1,1)).flatten()

df = pd.DataFrame({
    'celsius': celsius,
    'fahrenheit': fahrenheit_pred
})
df.to_csv('dataset_model.csv', index=False)
print("Dataset generado: dataset_model.csv")
```

```
32/32 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
Dataset generado: dataset_model.csv
```

```
# ── 3) Split: 80% train / 20% test, y reserva 5% de train para validación ──
train_df, test_df = train_test_split(df, test_size=0.20, random_state=42)
train_df, val_df  = train_test_split(train_df, test_size=0.05, random_state=42)

train_df.to_csv('train.csv', index=False)
val_df.to_csv('val.csv',     index=False)
test_df.to_csv('test.csv',   index=False)

print(f"Splits creados → train: {len(train_df)}, val: {len(val_df)}, test: {len(test_df)}")
```

```
Splits creados → train: 760, val: 40, test: 200
```

```
# ── 4) Retraining: compila y entrena el modelo sobre el nuevo dataset ──
model.compile(
    optimizer=tf.keras.optimizers.Adam(0.01),
    loss='mean_squared_error',
    metrics=['mae']
)

history = model.fit(
    train_df['celsius'].values.reshape(-1,1),
    train_df['fahrenheit'].values,
    epochs=100,
    validation_data=(
        val_df['celsius'].values.reshape(-1,1),
        val_df['fahrenheit'].values
    ),
    verbose=1
)
```

```
Epoch 1/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - loss: 130.1956 - mae: 7.9754 - val_loss: 17.5880 - val_mae: 3.6667
Epoch 2/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 14.4061 - mae: 2.9772 - val_loss: 0.6222 - val_mae: 0.6954
Epoch 3/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - loss: 1.2889 - mae: 0.8960 - val_loss: 0.0520 - val_mae: 0.1914
Epoch 4/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.1122 - mae: 0.2539 - val_loss: 0.0201 - val_mae: 0.1241
Epoch 5/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.0127 - mae: 0.0862 - val_loss: 3.8509e-04 - val_mae: 0.0172
Epoch 6/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.0011 - mae: 0.0253 - val_loss: 3.1931e-04 - val_mae: 0.0156
Epoch 7/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 1.3609e-04 - mae: 0.0090 - val_loss: 2.9415e-06 - val_mae: 0.0014
Epoch 8/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 1.0587e-05 - mae: 0.0025 - val_loss: 1.5075e-06 - val_mae: 0.0010
Epoch 9/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 1.2084e-06 - mae: 8.6553e-04 - val_loss: 1.5995e-07 - val_mae: 3.5341e-04
Epoch 10/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 7.8736e-08 - mae: 2.2712e-04 - val_loss: 9.4577e-09 - val_mae: 7.4792e-05
Epoch 11/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 5.1469e-08 - mae: 1.7026e-04 - val_loss: 2.1433e-09 - val_mae: 3.4904e-05
Epoch 12/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 7.8593e-09 - mae: 6.4750e-05 - val_loss: 9.1718e-10 - val_mae: 2.3699e-05
Epoch 13/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 1.1771e-09 - mae: 2.5053e-05 - val_loss: 8.7880e-10 - val_mae: 2.2841e-05
```

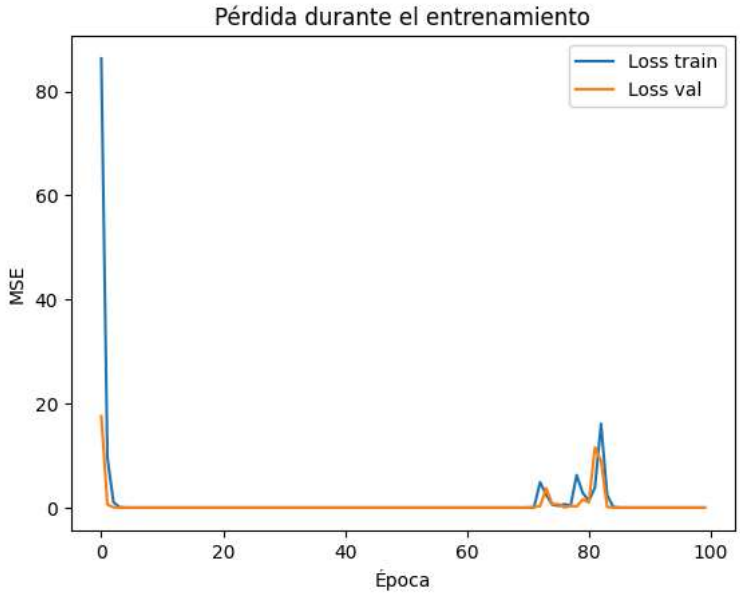
```
Epoch 14/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 1.0792e-09 - mae: 2.3759e-05 - val_loss: 5.2789e-10 - val_mae: 1.8048e-05
Epoch 15/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 8.4272e-10 - mae: 2.2139e-05 - val_loss: 4.7150e-10 - val_mae: 1.6999e-05
Epoch 16/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 8.5549e-10 - mae: 2.1936e-05 - val_loss: 8.3787e-10 - val_mae: 2.2125e-05
Epoch 17/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 8.4721e-10 - mae: 2.1583e-05 - val_loss: 1.2733e-09 - val_mae: 2.6131e-05
Epoch 18/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 1.2715e-09 - mae: 2.6401e-05 - val_loss: 6.5159e-10 - val_mae: 1.9872e-05
Epoch 19/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 1.0281e-09 - mae: 2.4008e-05 - val_loss: 6.8208e-10 - val_mae: 1.9693e-05
Epoch 20/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 1.0727e-09 - mae: 2.4844e-05 - val_loss: 1.2539e-09 - val_mae: 2.7585e-05
Epoch 21/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 1.0651e-09 - mae: 2.4622e-05 - val_loss: 1.4119e-09 - val_mae: 2.9707e-05
Epoch 22/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 1.2432e-09 - mae: 2.6208e-05 - val_loss: 1.1307e-09 - val_mae: 2.5964e-05
Epoch 23/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 1.2863e-09 - mae: 2.6409e-05 - val_loss: 1.5746e-09 - val_mae: 3.0863e-05
Epoch 24/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 2.4369e-09 - mae: 3.6557e-05 - val_loss: 2.2826e-09 - val_mae: 3.8135e-05
Epoch 25/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 2.7647e-09 - mae: 3.9221e-05 - val_loss: 1.2944e-09 - val_mae: 2.7418e-05
Epoch 26/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 1.0249e-09 - mae: 2.3165e-05 - val_loss: 1.0999e-09 - val_mae: 2.1553e-05
Epoch 27/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 1.0505e-09 - mae: 2.4550e-05 - val_loss: 1.2396e-09 - val_mae: 2.3842e-05
Epoch 28/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 1.1718e-09 - mae: 2.5999e-05 - val_loss: 1.6536e-09 - val_mae: 2.9922e-05
Epoch 29/100
24/24 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 1.0065e-09 - mae: 2.4245e-05 - val_loss: 1.5806e-09 - val_mae: 2.6977e-05
```

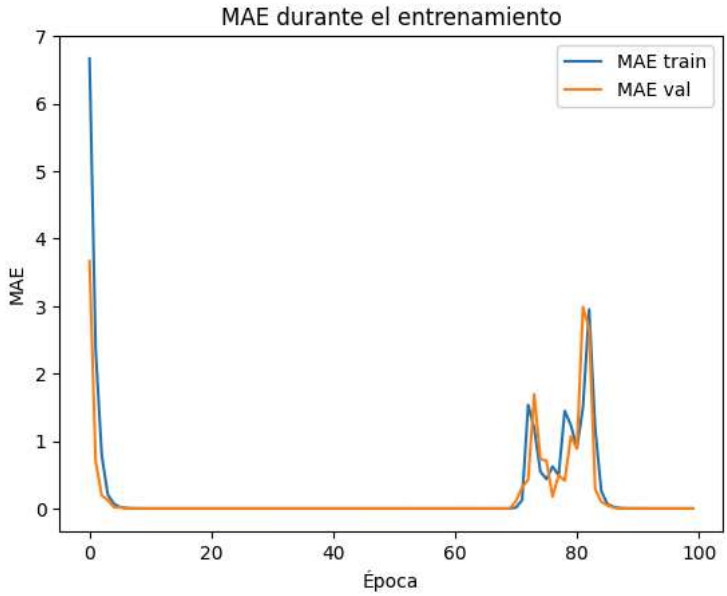
```
# ── 5) Evaluación sobre test set ──
loss, mae = model.evaluate(
    test_df['celsius'].values.reshape(-1,1),
    test_df['fahrenheit'].values,
    verbose=1
)
print(f"Test Loss: {loss:.4f}, Test MAE: {mae:.4f}")
```

```
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 2.3053e-09 - mae: 3.4758e-05
Test Loss: 0.0000, Test MAE: 0.0000
```

```
# ── 6) Gráficas de pérdida y MAE ──
plt.figure()
plt.plot(history.history['loss'],     label='Loss train')
plt.plot(history.history['val_loss'], label='Loss val')
plt.xlabel('Época')
plt.ylabel('MSE')
plt.legend()
plt.title('Pérdida durante el entrenamiento')
plt.show()

plt.figure()
plt.plot(history.history['mae'],      label='MAE train')
plt.plot(history.history['val_mae'],  label='MAE val')
plt.xlabel('Época')
plt.ylabel('MAE')
plt.legend()
plt.title('MAE durante el entrenamiento')
plt.show()
```


Pérdida durante el entrenamiento
Loss train
Loss val
MSE
Época
0
20
40
60
80
100


MAE durante el entrenamiento
MAE train
MAE val
MAE
Época
0
1
2
3
4
5
6
7
20
40
60
80
100