

In [1]:

```
import keras
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import Dense, Dropout, LSTM
from sklearn.preprocessing import MinMaxScaler
```

Using TensorFlow backend.

In [2]:

```
base = pd.read_csv("dataset/petr4-treinamento.csv")
base = base.dropna() #Remove as linhas onde há colunas com valores faltantes

#Normalização de dimensionamento de recursos
base_treinamento = base.iloc[:, 1:2].values
normalizador = MinMaxScaler(feature_range=(0, 1))
base_treinamento_normalizada = normalizador.fit_transform(base_treinamento)

#Criando uma estrutura de dados com 90 timesteps e 1 output
x_treino = []
y_treino = []

for i in range(90, 1242):
    x_treino.append(base_treinamento_normalizada[i-90:i, 0])
    y_treino.append(base_treinamento_normalizada[i, 0])

x_treino, y_treino = np.array(x_treino), np.array(y_treino)

#Remodelando
x_treino = np.reshape(x_treino, (x_treino.shape[0], x_treino.shape[1], 1))
```

In [3]:

```
### Predição na base de teste ###

base_teste = pd.read_csv('dataset/petr4-teste.csv')
preco_real_teste = base_teste.iloc[:, 1:2].values

base_completa = pd.concat((base['Open'], base_teste['Open']), axis=0)

entradas = base_completa[len(base_completa) - len(base_teste)-90:].values
entradas = entradas.reshape(-1, 1)
entradas = normalizador.transform(entradas)

x_teste = []
for i in range(90, 112):
    x_teste.append(entradas[i-90:i, 0])

x_teste = np.array(x_teste)
x_teste = np.reshape(x_teste, (x_teste.shape[0], x_teste.shape[1], 1))
```

In [4]:

```
#Criando modelo
model = Sequential()

#Adicionando camadas LSTM e alguma regularização de Dropout
model.add(LSTM(units=50,return_sequences=True,input_shape=(x_treino.shape[1], 1)))
model.add(Dropout(0.2))
model.add(LSTM(units=50))
model.add(Dropout(0.2))
model.add(Dense(units=1, activation='linear'))

adam = keras.optimizers.Adam(learning_rate=0.001, beta_1=0.9, beta_2=0.999, amsgrad=False)
model.compile(optimizer = adam, loss='mean_squared_error', metrics=['mean_absolute_error'])
model.fit(x_treino, y_treino, epochs=100, batch_size=32)

preco_predito = model.predict(x_teste)
preco_predito = normalizador.inverse_transform(preco_predito)

#Evaluate the model
scores = model.evaluate(x_treino, y_treino)
```

```
print("%s: %.4f" % (model.metrics_names[1], scores[1]))
```

```
Epoch 1/100
1152/1152 [=====] - 5s 4ms/step - loss: 0.0264 - mean_absolute_error: 0.1133
Epoch 2/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0053 - mean_absolute_error: 0.0571
Epoch 3/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0045 - mean_absolute_error: 0.0508
Epoch 4/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0043 - mean_absolute_error: 0.0499
Epoch 5/100
1152/1152 [=====] - 4s 4ms/step - loss: 0.0038 - mean_absolute_error: 0.0477
Epoch 6/100
1152/1152 [=====] - 4s 4ms/step - loss: 0.0037 - mean_absolute_error: 0.0465
Epoch 7/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0037 - mean_absolute_error: 0.0457
Epoch 8/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0038 - mean_absolute_error: 0.0468
Epoch 9/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0033 - mean_absolute_error: 0.0435
Epoch 10/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0032 - mean_absolute_error: 0.0424
Epoch 11/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0033 - mean_absolute_error: 0.0432
Epoch 12/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0033 - mean_absolute_error: 0.0438A: 3s - loss: 0.0
Epoch 13/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0029 - mean_absolute_error: 0.0414
Epoch 14/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0031 - mean_absolute_error: 0.0427
Epoch 15/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0030 - mean_absolute_error: 0.0414
Epoch 16/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0030 - mean_absolute_error: 0.0407
Epoch 17/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0032 - mean_absolute_error: 0.0418
Epoch 18/100
1152/1152 [=====] - 4s 4ms/step - loss: 0.0027 - mean_absolute_error: 0.0396
Epoch 19/100
1152/1152 [=====] - 4s 4ms/step - loss: 0.0027 - mean_absolute_error: 0.0393
Epoch 20/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0027 - mean_absolute_error: 0.0393
Epoch 21/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0024 - mean_absolute_error: 0.0379
Epoch 22/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0026 - mean_absolute_error: 0.0387
Epoch 23/100
1152/1152 [=====] - 4s 4ms/step - loss: 0.0026 - mean_absolute_error: 0.0382
Epoch 24/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0026 - mean_absolute_error: 0.0389
Epoch 25/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0025 - mean_absolute_error: 0.0379
Epoch 26/100
1152/1152 [=====] - 4s 4ms/step - loss: 0.0025 - mean_absolute_error: 0.0379
Epoch 27/100
1152/1152 [=====] - 4s 4ms/step - loss: 0.0023 - mean_absolute_error: 0.0360
Epoch 28/100
1152/1152 [=====] - 5s 4ms/step - loss: 0.0022 - mean_absolute_error: 0.0359
Epoch 29/100
1152/1152 [=====] - 5s 4ms/step - loss: 0.0025 - mean_absolute_error: 0.0377
Epoch 30/100
1152/1152 [=====] - 5s 4ms/step - loss: 0.0023 - mean_absolute_error: 0.0363
Epoch 31/100
1152/1152 [=====] - 4s 4ms/step - loss: 0.0023 - mean_absolute_error: 0.0355
Epoch 32/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0020 - mean_absolute_error: 0.0344
Epoch 33/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0022 - mean_absolute_error: 0.0352
Epoch 34/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0022 - mean_absolute_error: 0.0348
Epoch 35/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0021 - mean_absolute_error: 0.0344
Epoch 36/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0022 - mean_absolute_error: 0.0358
Epoch 37/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0019 - mean_absolute_error: 0.0329
Epoch 38/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0020 - mean_absolute_error: 0.0336
Epoch 39/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0020 - mean_absolute_error: 0.0340
Epoch 40/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0019 - mean_absolute_error: 0.0326
```

Epoch 41/100  
1152/1152 [=====] - 3s 3ms/step - loss: 0.0019 - mean\_absolute\_error: 0.0328  
Epoch 42/100  
1152/1152 [=====] - 3s 3ms/step - loss: 0.0023 - mean\_absolute\_error: 0.0361  
Epoch 43/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0019 - mean\_absolute\_error: 0.0332  
Epoch 44/100  
1152/1152 [=====] - 4s 4ms/step - loss: 0.0019 - mean\_absolute\_error: 0.0330  
Epoch 45/100  
1152/1152 [=====] - 5s 4ms/step - loss: 0.0019 - mean\_absolute\_error: 0.0333  
Epoch 46/100  
1152/1152 [=====] - 4s 4ms/step - loss: 0.0019 - mean\_absolute\_error: 0.0330  
Epoch 47/100  
1152/1152 [=====] - 4s 4ms/step - loss: 0.0018 - mean\_absolute\_error: 0.0319  
Epoch 48/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0016 - mean\_absolute\_error: 0.0303  
Epoch 49/100  
1152/1152 [=====] - 3s 3ms/step - loss: 0.0017 - mean\_absolute\_error: 0.0306  
Epoch 50/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0017 - mean\_absolute\_error: 0.0305  
Epoch 51/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0019 - mean\_absolute\_error: 0.0320  
Epoch 52/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0018 - mean\_absolute\_error: 0.0316  
Epoch 53/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0017 - mean\_absolute\_error: 0.0306  
Epoch 54/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0018 - mean\_absolute\_error: 0.0311A: 0s - loss: 0.0017 - mean\_absolute\_error: 0.0306  
Epoch 55/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0017 - mean\_absolute\_error: 0.0301  
Epoch 56/100  
1152/1152 [=====] - 3s 3ms/step - loss: 0.0017 - mean\_absolute\_error: 0.0314  
Epoch 57/100  
1152/1152 [=====] - 4s 4ms/step - loss: 0.0014 - mean\_absolute\_error: 0.0285  
Epoch 58/100  
1152/1152 [=====] - 4s 4ms/step - loss: 0.0016 - mean\_absolute\_error: 0.0306  
Epoch 59/100  
1152/1152 [=====] - 4s 4ms/step - loss: 0.0015 - mean\_absolute\_error: 0.0292  
Epoch 60/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0014 - mean\_absolute\_error: 0.0277  
Epoch 61/100  
1152/1152 [=====] - 3s 3ms/step - loss: 0.0014 - mean\_absolute\_error: 0.0278  
Epoch 62/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0015 - mean\_absolute\_error: 0.0284  
Epoch 63/100  
1152/1152 [=====] - 4s 4ms/step - loss: 0.0014 - mean\_absolute\_error: 0.0282  
Epoch 64/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0015 - mean\_absolute\_error: 0.0289  
Epoch 65/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0015 - mean\_absolute\_error: 0.0285  
Epoch 66/100  
1152/1152 [=====] - 4s 4ms/step - loss: 0.0016 - mean\_absolute\_error: 0.0290  
Epoch 67/100  
1152/1152 [=====] - 4s 4ms/step - loss: 0.0015 - mean\_absolute\_error: 0.0286  
Epoch 68/100  
1152/1152 [=====] - 4s 4ms/step - loss: 0.0014 - mean\_absolute\_error: 0.0282  
Epoch 69/100  
1152/1152 [=====] - 3s 3ms/step - loss: 0.0014 - mean\_absolute\_error: 0.0279  
Epoch 70/100  
1152/1152 [=====] - 3s 3ms/step - loss: 0.0014 - mean\_absolute\_error: 0.0282A: 1s - loss: 0.0015 - mean\_absolute\_error: 0.0282  
Epoch 71/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0015 - mean\_absolute\_error: 0.0288  
Epoch 72/100  
1152/1152 [=====] - 3s 3ms/step - loss: 0.0013 - mean\_absolute\_error: 0.0265  
Epoch 73/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0012 - mean\_absolute\_error: 0.0257  
Epoch 74/100  
1152/1152 [=====] - 4s 4ms/step - loss: 0.0013 - mean\_absolute\_error: 0.0274  
Epoch 75/100  
1152/1152 [=====] - 3s 3ms/step - loss: 0.0014 - mean\_absolute\_error: 0.0279  
Epoch 76/100  
1152/1152 [=====] - 3s 3ms/step - loss: 0.0013 - mean\_absolute\_error: 0.0279  
Epoch 77/100  
1152/1152 [=====] - 3s 3ms/step - loss: 0.0012 - mean\_absolute\_error: 0.0260  
Epoch 78/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0012 - mean\_absolute\_error: 0.0264  
Epoch 79/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0013 - mean\_absolute\_error: 0.0267  
Epoch 80/100  
1152/1152 [=====] - 3s 3ms/step - loss: 0.0011 - mean\_absolute\_error: 0.0245  
Epoch 81/100  
1152/1152 [=====] - 4s 3ms/step - loss: 0.0013 - mean\_absolute\_error: 0.0257

```

Epoch 82/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0012 - mean_absolute_error: 0.0257
Epoch 83/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0012 - mean_absolute_error: 0.0258
Epoch 84/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0011 - mean_absolute_error: 0.0250
Epoch 85/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0012 - mean_absolute_error: 0.0257
Epoch 86/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0012 - mean_absolute_error: 0.0252
Epoch 87/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0012 - mean_absolute_error: 0.0253
Epoch 88/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0013 - mean_absolute_error: 0.0265
Epoch 89/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0011 - mean_absolute_error: 0.0245
Epoch 90/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0011 - mean_absolute_error: 0.0244
Epoch 91/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0012 - mean_absolute_error: 0.0262
Epoch 92/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0011 - mean_absolute_error: 0.0246
Epoch 93/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0011 - mean_absolute_error: 0.0250
Epoch 94/100
1152/1152 [=====] - 4s 3ms/step - loss: 0.0011 - mean_absolute_error: 0.0243
Epoch 95/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0010 - mean_absolute_error: 0.0240- ETA: 0s - loss: 0.0010 - mean_abs
olute_error: 0.023
Epoch 96/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0010 - mean_absolute_error: 0.0242
Epoch 97/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0011 - mean_absolute_error: 0.0246
Epoch 98/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0011 - mean_absolute_error: 0.0251
Epoch 99/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0011 - mean_absolute_error: 0.0239
Epoch 100/100
1152/1152 [=====] - 3s 3ms/step - loss: 0.0010 - mean_absolute_error: 0.0241
1152/1152 [=====] - 1s 719us/step
mean_absolute_error: 0.0185

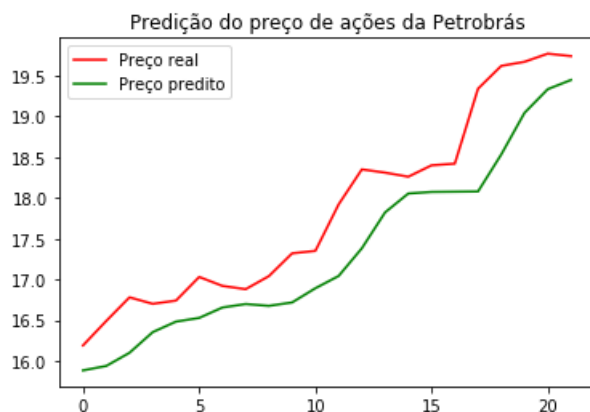
```

In [5]:

```

#Plotando resultados
plt.plot(preco_real_teste, color = 'red', label = 'Preço real')
plt.plot(preco_predito, color = 'green', label = 'Preço predito')
plt.title('Predição do preço de ações da Petrobrás')
plt.legend()
plt.show()

```



In []: