In [1]:

```
# Não exibir warnings
import os
import sys
sys.stderr = open(os.devnull, "w")  # silence stderr
sys.stderr = sys.__stderr__  # unsilence stderr
```

In [2]:

```
from keras.models import Sequential
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.layers import Dense
import matplotlib.pyplot as plt
from sklearn.metrics import classification_report, confusion_matrix
import tensorflow as tf
import numpy as np
import pandas as pd
import seaborn as sns

from PIL import Image, ImageFile
ImageFile.LOAD_TRUNCATED_IMAGES = True
```

In [3]:

```
#Função de geração da matriz de confusão
def print_confusion_matrix(confusion_matrix, class_names, figsize = (10,7), fontsize=11
):
    df_cm = pd.DataFrame(
        confusion_matrix, index=class_names, columns=class_names,
    )
    fig = plt.figure(figsize=figsize)
    try:
        heatmap = sns.heatmap(df_cm, cmap="YlGnBu", annot=True, fmt="d")
    except ValueError:
        raise ValueError("Confusion matrix values must be integers.")
    heatmap.yaxis.set_ticklabels(heatmap.yaxis.get_ticklabels(), rotation=0, ha='right'
, fontsize=fontsize)
    heatmap.xaxis.set_ticklabels(heatmap.xaxis.get_ticklabels(), rotation=30, ha='righ
t', fontsize=fontsize)
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
    #return fig
```

In [4]:

```
batch = 32
#num_train = 5600
#num_validation = 2400
```

In [5]:

```
# Part 1 - Configuring the CNN

# Initialising the CNN
classifier = Sequential()

# Step 1 - Convolution
classifier.add(Conv2D(64, (3, 3), input_shape = (128, 128, 3), activation = 'relu'))

# Step 2 - Pooling
classifier.add(MaxPooling2D(pool_size = (2, 2)))

# Adding a second convolutional layer
classifier.add(Conv2D(64, (3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))

# Adding a second convolutional layer
classifier.add(Conv2D(128, (3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))

# Adding a second convolutional layer
classifier.add(Conv2D(128, (3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))

classifier.add(Conv2D(256, (3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))

# Step 3 - Flattening
classifier.add(Flatten())

# Step 4 - Full connection
classifier.add(Dense(units = 256, activation = 'relu'))
classifier.add(Dense(units = 2, activation = 'sigmoid')) #mudar unidades para numero de
classes

# Compiling the CNN
classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['a
ccuracy'])#adam
```

In [6]:

```
# Part 2 - Fitting the CNN to the images

from keras.preprocessing.image import ImageDataGenerator

#Conjunto de treinamento
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

training_set = train_datagen.flow_from_directory('classificador_A/train',
                                                 target_size = (128, 128),
                                                 color_mode="rgb",
                                                 batch_size = batch,
                                                 class_mode = 'categorical',
                                                 shuffle = True)

print (training_set.class_indices)
#Conjunto de validação
validation_datagen = ImageDataGenerator(rescale = 1./255)

validation_set = validation_datagen.flow_from_directory('classificador_A/validation/',
                                            target_size = (128, 128),
                                            color_mode="rgb",
                                            batch_size = batch, #alterado para 1
                                            class_mode = 'categorical',
                                            shuffle=True)
num_train = training_set.samples
num_validation = validation_set.samples
```

```
Found 4994 images belonging to 2 classes.
{'documentos': 0, 'nao_documentos': 1}
Found 2396 images belonging to 2 classes.
```

In [7]:

```
with tf.device('/gpu:0'): #rodar na GPU
    history = classifier.fit_generator(training_set,
                         steps_per_epoch = (num_train//batch),
                         epochs = 50,
                         validation_data = validation_set,
                         validation_steps = (num_validation//batch), verbose=1)


classifier.save('modelo_classificador_A_dist_01.h5')
```

```
Epoch 1/50
156/156 [==============================] - 505s 3s/step - loss: 0.3689 - a
cc: 0.8556 - val_loss: 0.2520 - val_acc: 0.9008
Epoch 2/50
156/156 [==============================] - 465s 3s/step - loss: 0.2723 - a
cc: 0.8902 - val_loss: 0.2273 - val_acc: 0.9154
Epoch 3/50
156/156 [==============================] - 466s 3s/step - loss: 0.2133 - a
cc: 0.9219 - val_loss: 0.1724 - val_acc: 0.9332
Epoch 4/50
156/156 [==============================] - 535s 3s/step - loss: 0.1734 - a
cc: 0.9367 - val_loss: 0.1723 - val_acc: 0.9387
Epoch 5/50
156/156 [==============================] - 632s 4s/step - loss: 0.1624 - a
cc: 0.9439 - val_loss: 0.1515 - val_acc: 0.9522
Epoch 6/50
156/156 [==============================] - 487s 3s/step - loss: 0.1347 - a
cc: 0.9507 - val_loss: 0.1769 - val_acc: 0.9370
Epoch 7/50
156/156 [==============================] - 555s 4s/step - loss: 0.1333 - a
cc: 0.9537 - val_loss: 0.2460 - val_acc: 0.9167
Epoch 8/50
156/156 [==============================] - 584s 4s/step - loss: 0.1148 - a
cc: 0.9593 - val_loss: 0.1371 - val_acc: 0.9539
Epoch 9/50
156/156 [==============================] - 580s 4s/step - loss: 0.1255 - a
cc: 0.9567 - val_loss: 0.1740 - val_acc: 0.9518
Epoch 10/50
156/156 [==============================] - 557s 4s/step - loss: 0.1002 - a
cc: 0.9647 - val_loss: 0.1487 - val_acc: 0.9514
Epoch 11/50
156/156 [==============================] - 591s 4s/step - loss: 0.1215 - a
cc: 0.9605 - val_loss: 0.1544 - val_acc: 0.9480
Epoch 12/50
156/156 [==============================] - 628s 4s/step - loss: 0.1075 - a
cc: 0.9617 - val_loss: 0.1294 - val_acc: 0.9590
Epoch 13/50
156/156 [==============================] - 630s 4s/step - loss: 0.0898 - a
cc: 0.9687 - val_loss: 0.1360 - val_acc: 0.9543
Epoch 14/50
156/156 [==============================] - 629s 4s/step - loss: 0.0800 - a
cc: 0.9718 - val_loss: 0.1343 - val_acc: 0.9556
Epoch 15/50
156/156 [==============================] - 520s 3s/step - loss: 0.0764 - a
cc: 0.9714 - val_loss: 0.1384 - val_acc: 0.9611
Epoch 16/50
156/156 [==============================] - 559s 4s/step - loss: 0.0767 - a
cc: 0.9756 - val_loss: 0.1527 - val_acc: 0.9569
Epoch 17/50
156/156 [==============================] - 462s 3s/step - loss: 0.0809 - a
cc: 0.9732 - val_loss: 0.1166 - val_acc: 0.9649
Epoch 18/50
156/156 [==============================] - 465s 3s/step - loss: 0.0618 - a
cc: 0.9786 - val_loss: 0.1508 - val_acc: 0.9645
Epoch 19/50
156/156 [==============================] - 459s 3s/step - loss: 0.0664 - a
cc: 0.9750 - val_loss: 0.1184 - val_acc: 0.9607
Epoch 20/50
156/156 [==============================] - 462s 3s/step - loss: 0.0516 - a
cc: 0.9806 - val_loss: 0.1657 - val_acc: 0.9615
Epoch 21/50
```

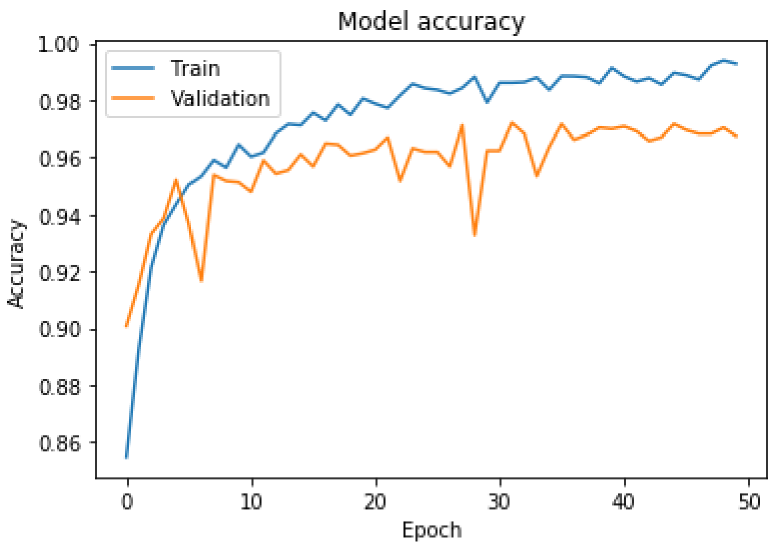
```
156/156 [==============================] - 454s 3s/step - loss: 0.0629 - a
cc: 0.9790 - val_loss: 0.1316 - val_acc: 0.9628
Epoch 22/50
156/156 [==============================] - 457s 3s/step - loss: 0.0684 - a
cc: 0.9774 - val_loss: 0.1257 - val_acc: 0.9670
Epoch 23/50
156/156 [==============================] - 456s 3s/step - loss: 0.0551 - a
cc: 0.9818 - val_loss: 0.1878 - val_acc: 0.9518
Epoch 24/50
156/156 [==============================] - 457s 3s/step - loss: 0.0467 - a
cc: 0.9858 - val_loss: 0.1311 - val_acc: 0.9632
Epoch 25/50
156/156 [==============================] - 459s 3s/step - loss: 0.0540 - a
cc: 0.9842 - val_loss: 0.1487 - val_acc: 0.9619
Epoch 26/50
156/156 [==============================] - 454s 3s/step - loss: 0.0453 - a
cc: 0.9838 - val_loss: 0.2092 - val_acc: 0.9619
Epoch 27/50
156/156 [==============================] - 459s 3s/step - loss: 0.0520 - a
cc: 0.9824 - val_loss: 0.1684 - val_acc: 0.9569
Epoch 28/50
156/156 [==============================] - 457s 3s/step - loss: 0.0450 - a
cc: 0.9844 - val_loss: 0.1143 - val_acc: 0.9712
Epoch 29/50
156/156 [==============================] - 458s 3s/step - loss: 0.0338 - a
cc: 0.9882 - val_loss: 0.2990 - val_acc: 0.9327
Epoch 30/50
156/156 [==============================] - 454s 3s/step - loss: 0.0665 - a
cc: 0.9794 - val_loss: 0.1814 - val_acc: 0.9624
Epoch 31/50
156/156 [==============================] - 456s 3s/step - loss: 0.0384 - a
cc: 0.9862 - val_loss: 0.1524 - val_acc: 0.9624
Epoch 32/50
156/156 [==============================] - 456s 3s/step - loss: 0.0366 - a
cc: 0.9862 - val_loss: 0.1233 - val_acc: 0.9721
Epoch 33/50
156/156 [==============================] - 458s 3s/step - loss: 0.0394 - a
cc: 0.9864 - val_loss: 0.1813 - val_acc: 0.9683
Epoch 34/50
156/156 [==============================] - 453s 3s/step - loss: 0.0313 - a
cc: 0.9880 - val_loss: 0.2974 - val_acc: 0.9535
Epoch 35/50
156/156 [==============================] - 462s 3s/step - loss: 0.0468 - a
cc: 0.9836 - val_loss: 0.1639 - val_acc: 0.9636
Epoch 36/50
156/156 [==============================] - 459s 3s/step - loss: 0.0353 - a
cc: 0.9886 - val_loss: 0.1346 - val_acc: 0.9717
Epoch 37/50
156/156 [==============================] - 460s 3s/step - loss: 0.0386 - a
cc: 0.9884 - val_loss: 0.1500 - val_acc: 0.9662
Epoch 38/50
156/156 [==============================] - 455s 3s/step - loss: 0.0317 - a
cc: 0.9882 - val_loss: 0.1693 - val_acc: 0.9679
Epoch 39/50
156/156 [==============================] - 459s 3s/step - loss: 0.0425 - a
cc: 0.9860 - val_loss: 0.1476 - val_acc: 0.9704
Epoch 40/50
156/156 [==============================] - 458s 3s/step - loss: 0.0272 - a
cc: 0.9914 - val_loss: 0.1640 - val_acc: 0.9700
Epoch 41/50
156/156 [==============================] - 458s 3s/step - loss: 0.0363 - a
```

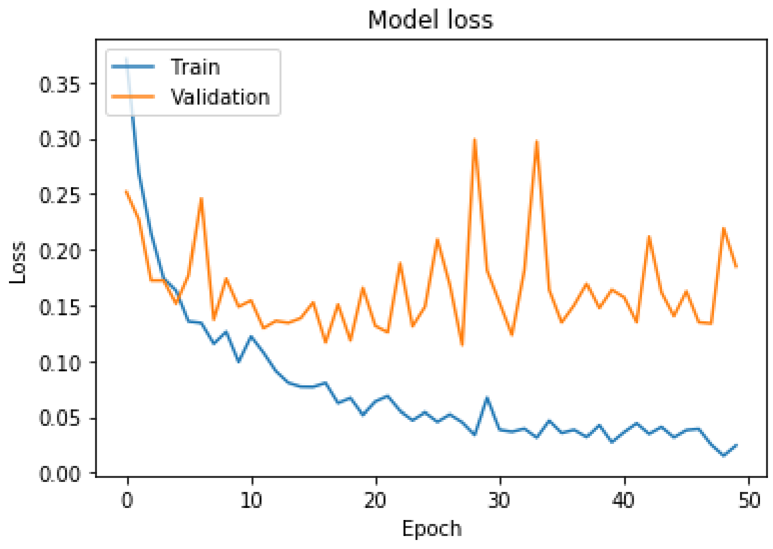
```
cc: 0.9884 - val_loss: 0.1572 - val_acc: 0.9708
Epoch 42/50
156/156 [==============================] - 456s 3s/step - loss: 0.0441 - a
cc: 0.9866 - val_loss: 0.1349 - val_acc: 0.9691
Epoch 43/50
156/156 [==============================] - 455s 3s/step - loss: 0.0347 - a
cc: 0.9878 - val_loss: 0.2117 - val_acc: 0.9657
Epoch 44/50
156/156 [==============================] - 457s 3s/step - loss: 0.0409 - a
cc: 0.9856 - val_loss: 0.1616 - val_acc: 0.9670
Epoch 45/50
156/156 [==============================] - 459s 3s/step - loss: 0.0317 - a
cc: 0.9896 - val_loss: 0.1404 - val_acc: 0.9717
Epoch 46/50
156/156 [==============================] - 454s 3s/step - loss: 0.0381 - a
cc: 0.9888 - val_loss: 0.1626 - val_acc: 0.9695
Epoch 47/50
156/156 [==============================] - 453s 3s/step - loss: 0.0390 - a
cc: 0.9874 - val_loss: 0.1348 - val_acc: 0.9683
Epoch 48/50
156/156 [==============================] - 460s 3s/step - loss: 0.0253 - a
cc: 0.9922 - val_loss: 0.1337 - val_acc: 0.9683
Epoch 49/50
156/156 [==============================] - 455s 3s/step - loss: 0.0151 - a
cc: 0.9940 - val_loss: 0.2193 - val_acc: 0.9704
Epoch 50/50
156/156 [==============================] - 458s 3s/step - loss: 0.0245 - a
cc: 0.9928 - val_loss: 0.1850 - val_acc: 0.9674
```

In [8]:

```
# Final accuracy and loss
print ("Train accuracy: %.3f" % (history.history['acc'][-1]))
print ("Train loss: %.3f" % (history.history['loss'][-1]),"\n")

print ("Validation accuracy: %.3f" % (history.history['val_acc'][-1]))
print ("Validation loss: %.3f" % (history.history['val_loss'][-1]))

# Plot training & validation accuracy values
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()

# Plot training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Validation'], loc='upper left')
plt.show()
```

```
Train accuracy: 0.993
Train loss: 0.025

Validation accuracy: 0.967
Validation loss: 0.185
```

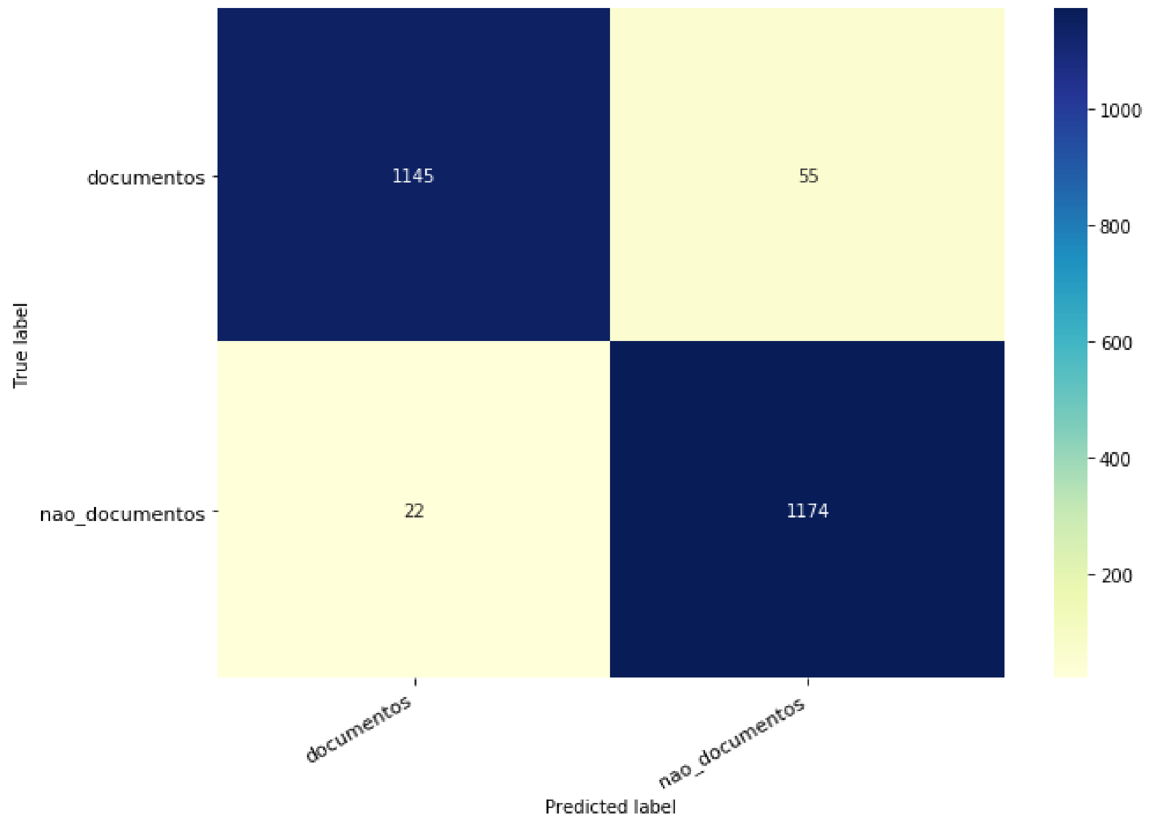




In [9]:

```
### Conjunto de Validação ###

print ("### Matriz de confusão para o conjunto de validação ###")

#Conjunto de validação
validation_datagen = ImageDataGenerator(rescale = 1./255)

validation_set = validation_datagen.flow_from_directory('classificador_A/validation/',
                                                        target_size = (128, 128),
                                                        color_mode="rgb",
                                                        batch_size = batch, #alterado para 1
                                                        class_mode = 'categorical',
                                                        shuffle= False)


#Confution Matrix
Y_pred = classifier.predict_generator(validation_set, num_validation // batch+1, verbos
e=1)

test_preds = np.argmax(Y_pred, axis=-1)
l=test_preds.shape[0]
test_trues = validation_set.classes
cm =confusion_matrix(test_trues[:l], test_preds)

print_confusion_matrix(cm, ["documentos", "nao_documentos"], figsize = (10,7), fontsize
=11)
```

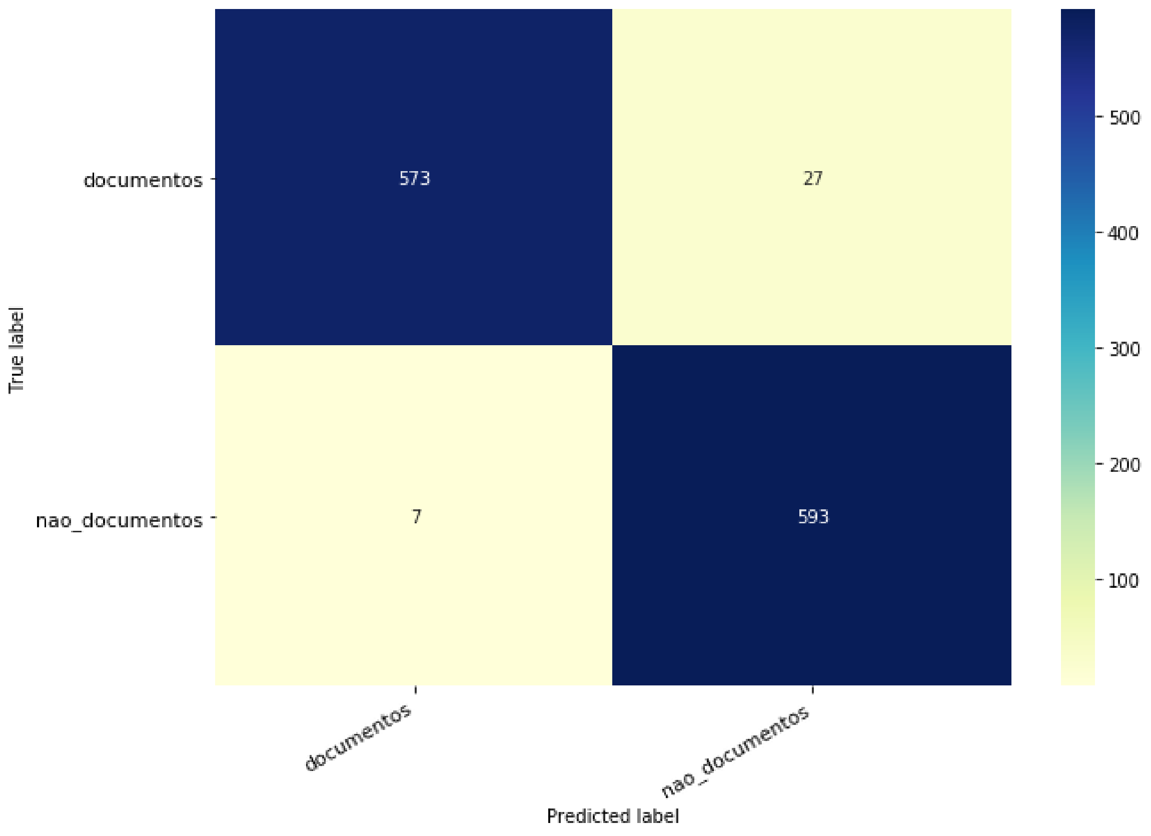
```
### Matriz de confusão para o conjunto de validação ###
Found 2396 images belonging to 2 classes.
75/75 [==============================] - 86s 1s/step
```



In [10]:

```
### Conjunto de Teste ###

print ("### Matriz de confusão para o conjunto de teste ###")

test_datagen = ImageDataGenerator(rescale = 1./255)

test_set = test_datagen.flow_from_directory('classificador_A/test/',
                                            target_size = (128, 128),
                                            color_mode="rgb",
                                            batch_size = 1,
                                            class_mode = 'categorical',
                                            shuffle=False)
num_test = test_set.samples


#Confution Matrix
Y_pred = classifier.predict_generator(test_set, num_test, verbose=1)

test_preds = np.argmax(Y_pred, axis=-1)
l=test_preds.shape[0]
test_trues = test_set.classes
cm =confusion_matrix(test_trues[:l], test_preds)

print_confusion_matrix(cm, ["documentos", "nao_documentos"], figsize = (10,7), fontsize
=11)

# Accuracy and Loss for the Test set
loss, acc = classifier.evaluate_generator(test_set, num_test, verbose=1)

# Final accuracy and loss
print ("Test accuracy: %.3f" % acc)
print ("Test loss: %.3f" % loss)
```

```
### Matriz de confusão para o conjunto de teste ###
Found 1200 images belonging to 2 classes.
1200/1200 [==============================] - 56s 47ms/step
1200/1200 [==============================] - 55s 46ms/step
Test accuracy: 0.972
Test loss: 0.149
```


True label
documentos
nao_documentos
documentos
nao_documentos
Predicted label