
# Lista 2
# CNN e RNN

Arthur Ricardo Ribeiro Lopes

Italo Nicacio dos Santos

João Pessoa - 2024

# Sumário

# Atividade 1
## MNIST

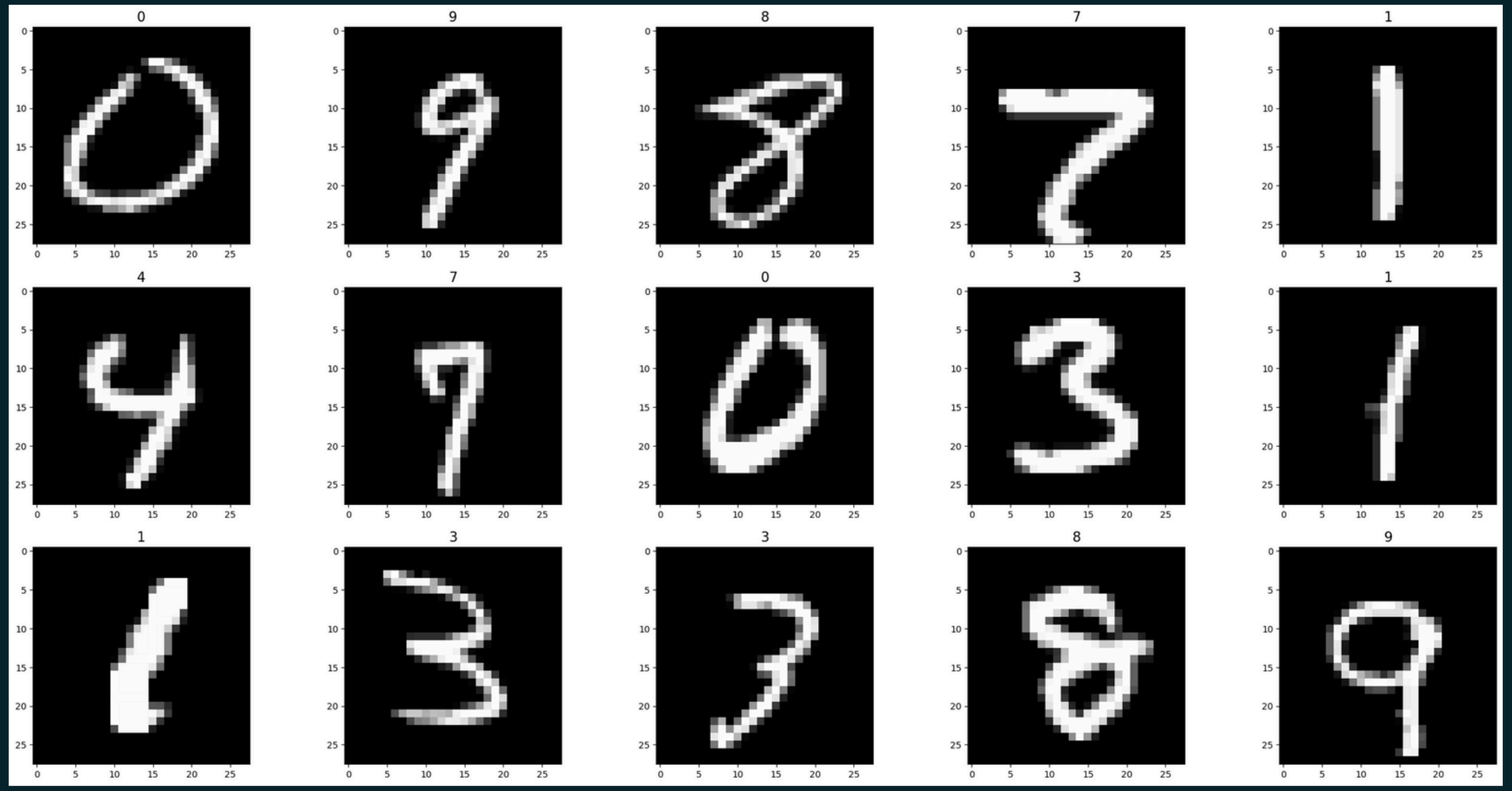
- 10 Classes
- Imagens de manuscritos de dígitos de 0 a 9

- Imagens 28x28

Dados:

- 6k por classe
- 60k Treinamento
- 10k teste.

# Atividade 1



# Atividade 1

## CNN

```
model_cnn = keras.Sequential([
  keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
  keras.layers.MaxPooling2D((2, 2)),
  keras.layers.Flatten(),
  keras.layers.Dense(10, activation='softmax')
])
```

- Otimizador:  Adam
- Loss: Categorical crossentropy
- Épocas:  5
- Batch Size: 32
- Parâmetros totais: 163.232

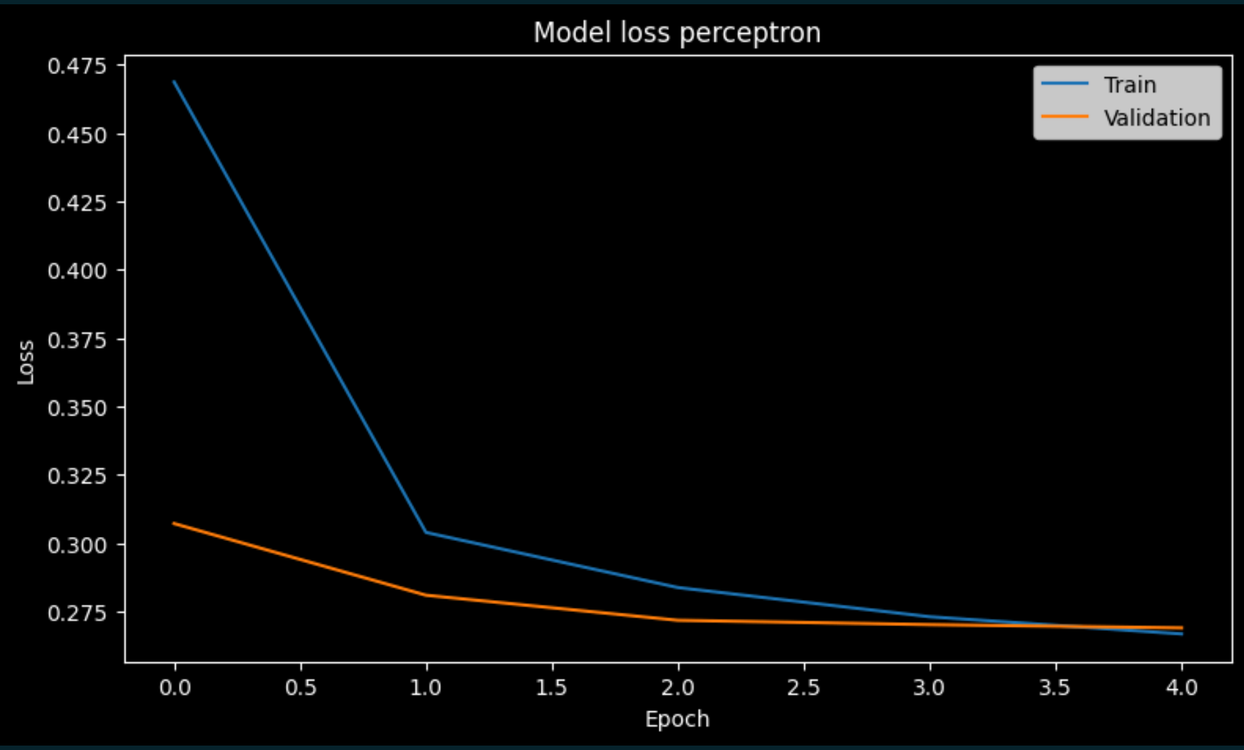
# Atividade 1

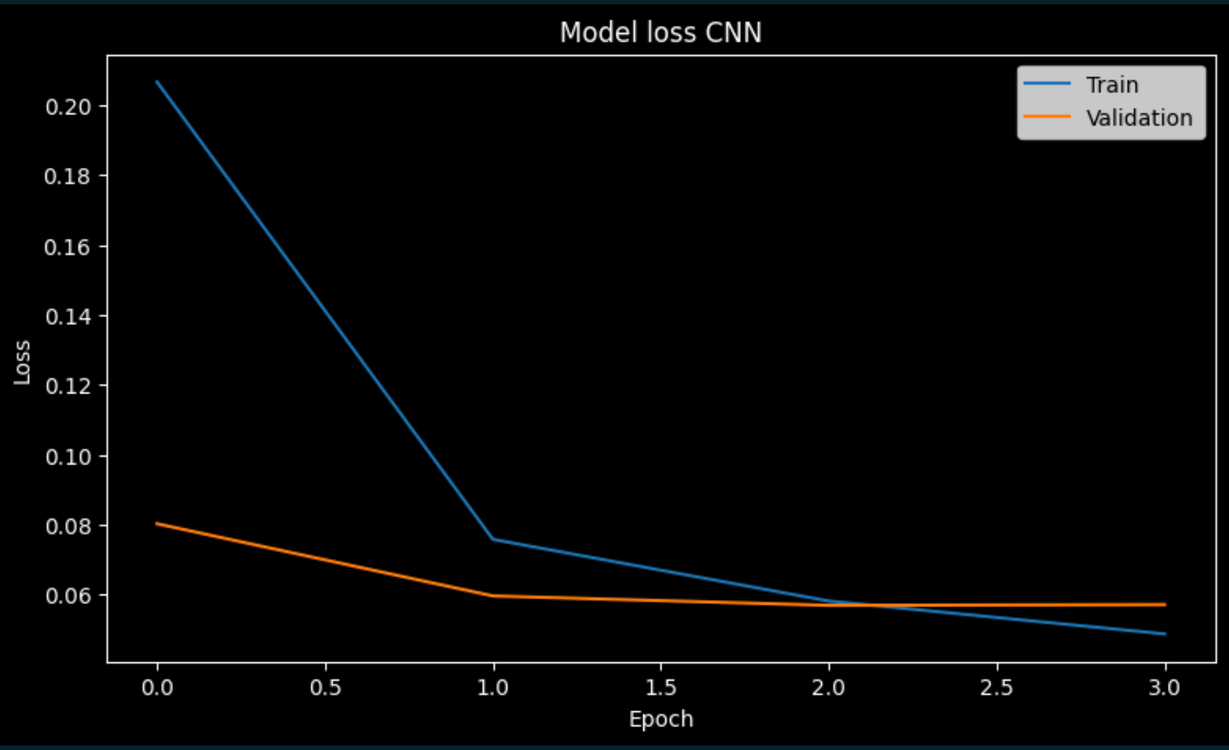
## Clássica

```
model = keras.Sequential([
  keras.layers.Flatten(),
  keras.layers.Dense(10, activation='softmax')
])
```

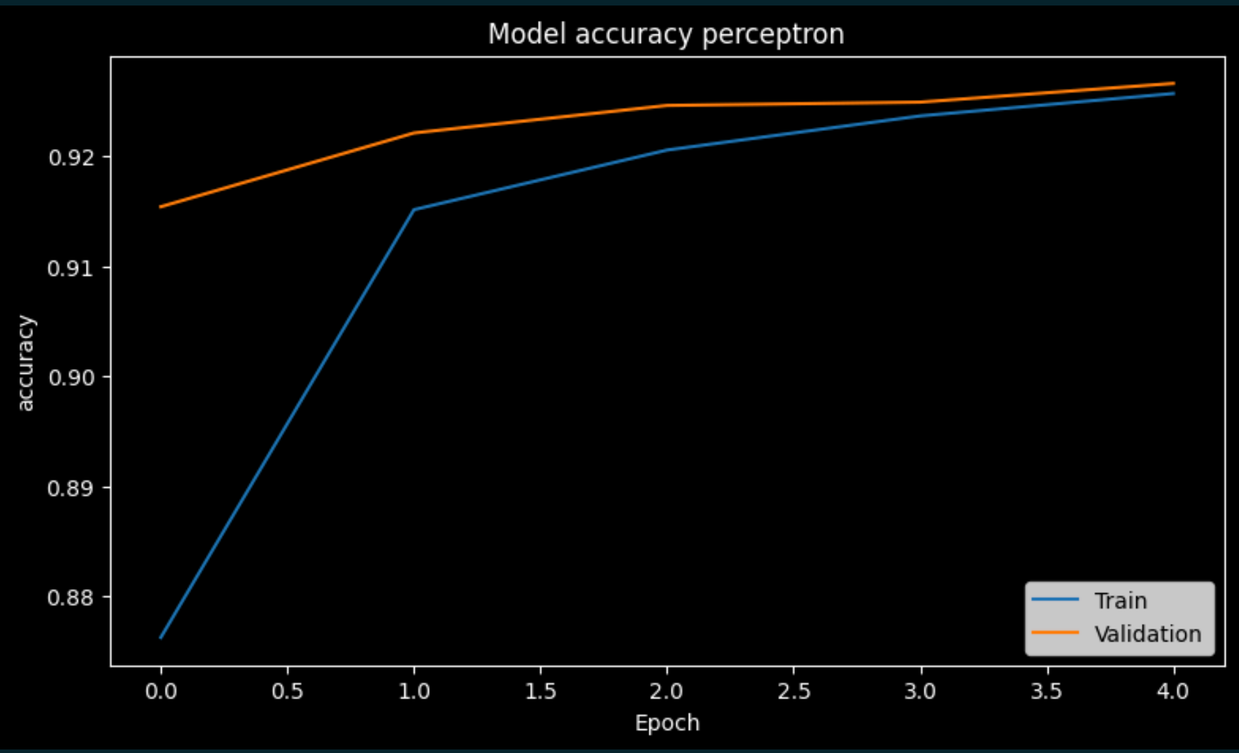
- Otimizador:  Adam
- Loss: Categorical crossentropy
- Épocas:  5
- Batch Size: 32
- Parâmetros totais: 23.552

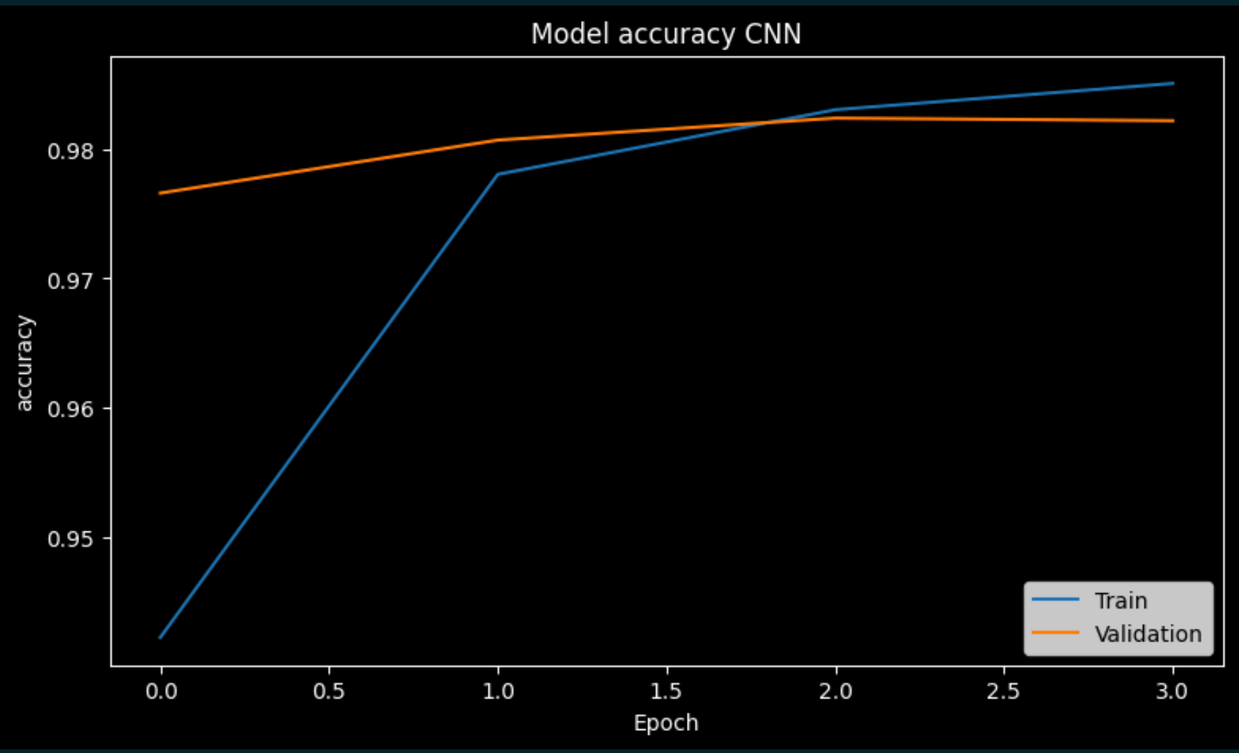
# Atividade 1 – Resultados



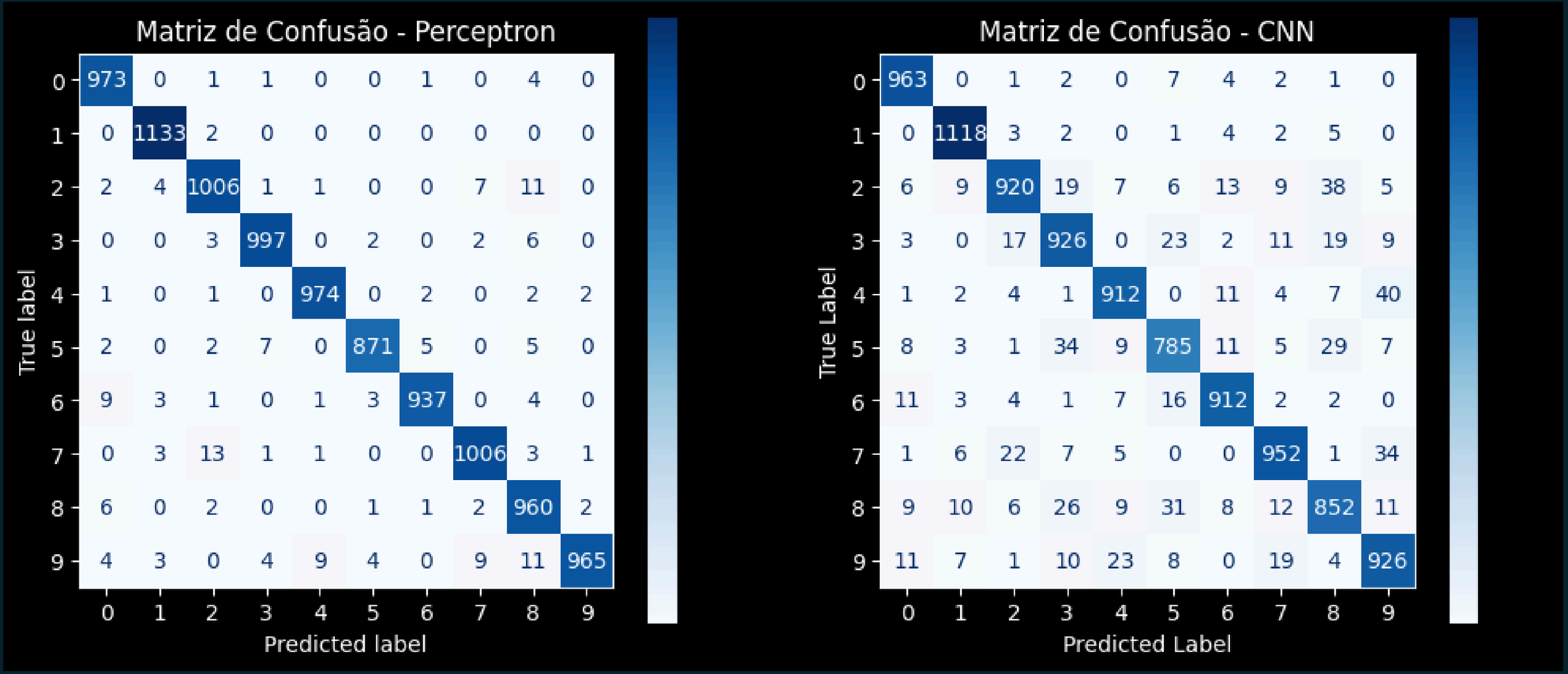



# Atividade 1 – Resultados



# Atividade 1 – Resultados



# Atividade 1 – Resultados

## CNN

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.98 | 0.99 | 0.99 | 980 |
| 1 | 0.98 | 1.00 | 0.99 | 1135 |
| 2 | 0.98 | 0.98 | 0.98 | 1032 |
| 3 | 0.99 | 0.98 | 0.99 | 1010 |
| 4 | 0.98 | 0.99 | 0.99 | 982 |
| 5 | 0.98 | 0.98 | 0.98 | 892 |
| 6 | 0.99 | 0.98 | 0.99 | 958 |
| 7 | 0.97 | 0.98 | 0.98 | 1028 |
| 8 | 0.98 | 0.98 | 0.98 | 974 |
| 9 | 0.99 | 0.95 | 0.97 | 1009 |
| micro avg | 0.98 | 0.98 | 0.98 | 10000 |
| macro avg | 0.98 | 0.98 | 0.98 | 10000 |
| weighted avg | 0.98 | 0.98 | 0.98 | 10000 |
| samples avg | 0.98 | 0.98 | 0.98 | 10000 |

## Clássica

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.96 | 0.98 | 0.97 | 980 |
| 1 | 0.98 | 0.97 | 0.98 | 1135 |
| 2 | 0.95 | 0.89 | 0.92 | 1032 |
| 3 | 0.92 | 0.91 | 0.91 | 1010 |
| 4 | 0.91 | 0.94 | 0.93 | 982 |
| 5 | 0.93 | 0.84 | 0.88 | 892 |
| 6 | 0.96 | 0.94 | 0.95 | 958 |
| 7 | 0.94 | 0.91 | 0.93 | 1028 |
| 8 | 0.89 | 0.88 | 0.89 | 974 |
| 9 | 0.95 | 0.84 | 0.89 | 1009 |
| micro avg | 0.94 | 0.91 | 0.93 | 10000 |
| macro avg | 0.94 | 0.91 | 0.92 | 10000 |
| weighted avg | 0.94 | 0.91 | 0.93 | 10000 |
| samples avg | 0.91 | 0.91 | 0.91 | 10000 |

# Atividade 2
# Cifar-10

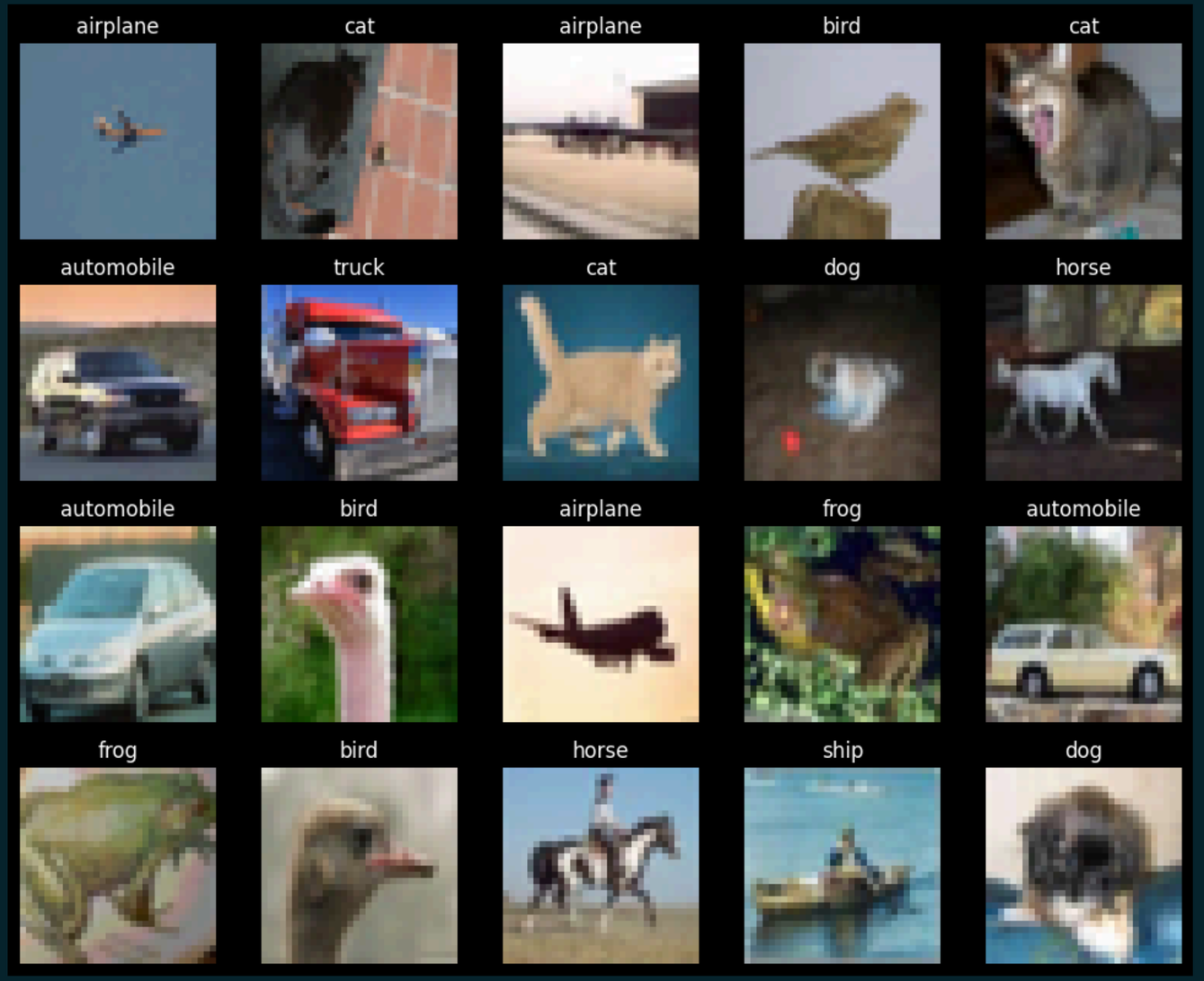
- Imagens 32x32

Dados:

- 6k por classe
- 50k Treinamento
- 10k teste.

- 10 Classes
  - airplane
  - automobile
  - bird
  - cat
  - deer
  - dog
  - frog
  - horse
  - ship
  - truck

# Atividade 2


airplane
cat
airplane
bird
cat
automobile
truck
cat
dog
horse
automobile
bird
airplane
frog
automobile
frog
bird
horse
ship
dog

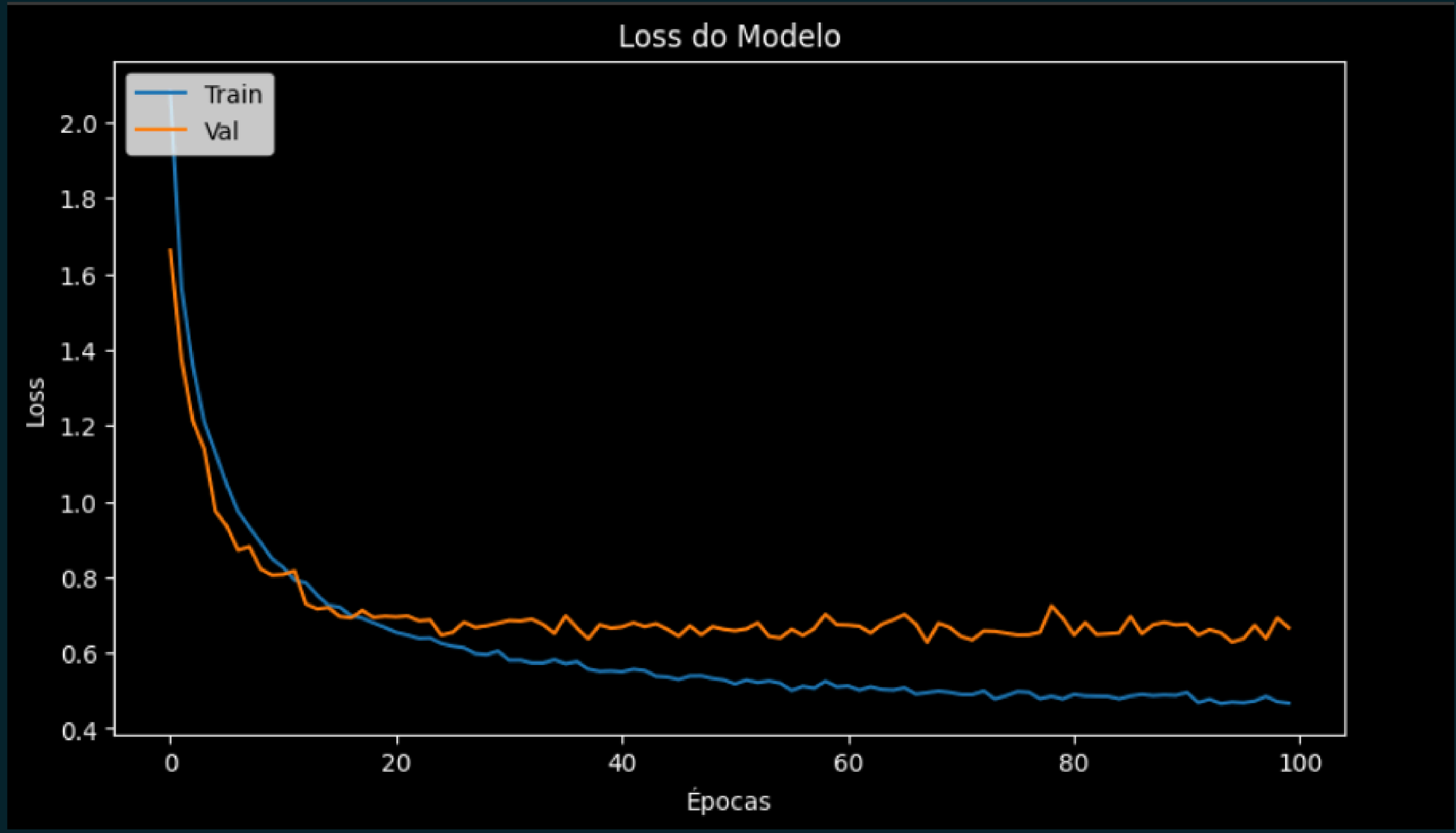
# Atividade 2

- Arquitetura e hiperparametros

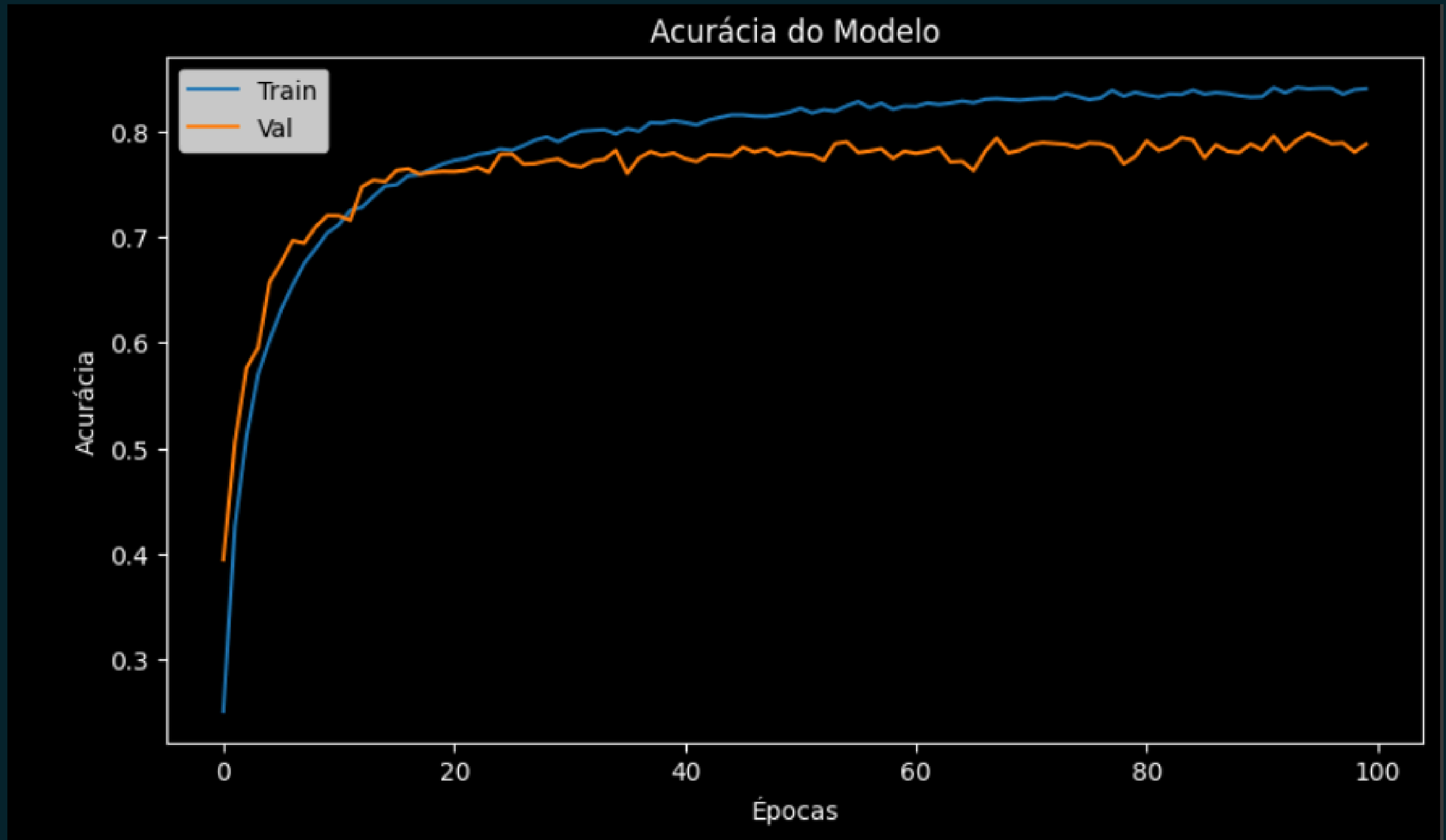
```
model = keras.models.Sequential([
    keras.layers.Input((32, 32, 3)),
    keras.layers.Conv2D(32, (3, 3), padding='same', activation='relu'),
    keras.layers.Conv2D(32, (3, 3), activation='relu'),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Dropout(0.25),

    keras.layers.Conv2D(64, (3, 3), padding='same', activation='relu'),
    keras.layers.Conv2D(64, (3, 3), activation='relu'),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Dropout(0.25),

    keras.layers.Conv2D(128, (3, 3), padding='same', activation='relu'),
    keras.layers.Conv2D(128, (3, 3), activation='relu'),
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Dropout(0.25),

    keras.layers.Flatten(),

    keras.layers.Dense(512, activation='relu'),
    keras.layers.Dropout(0.25),

    keras.layers.Dense(10, activation='softmax')
])
```

- Otimizador:  Adam
- Learning rate: 0.001
- Loss: Categorical crossentropy
- Épocas:  100
- Batch Size: 128
- Validation Split: 0.2

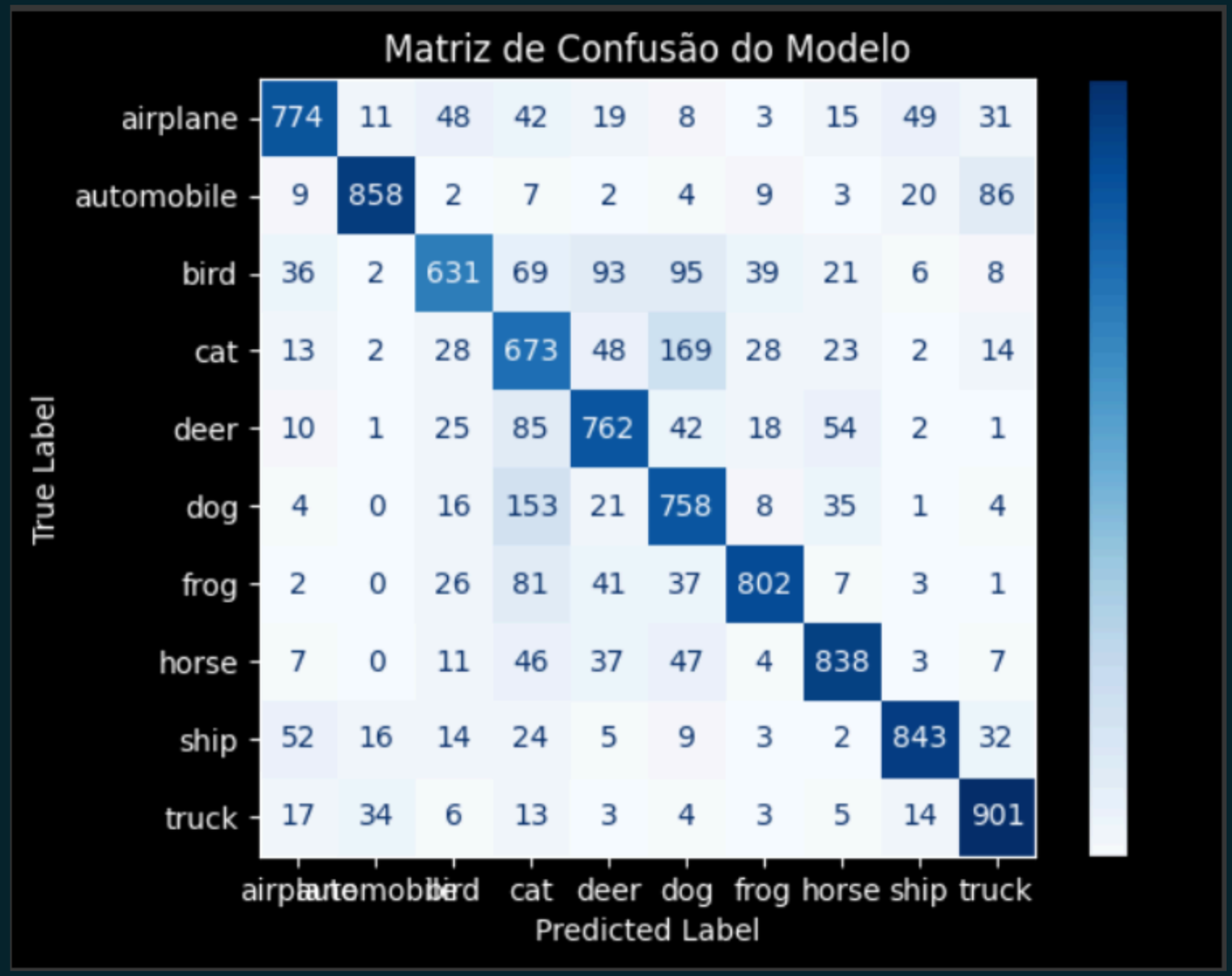
# Atividade 2 – Resultados



# Atividade 2 – Resultados



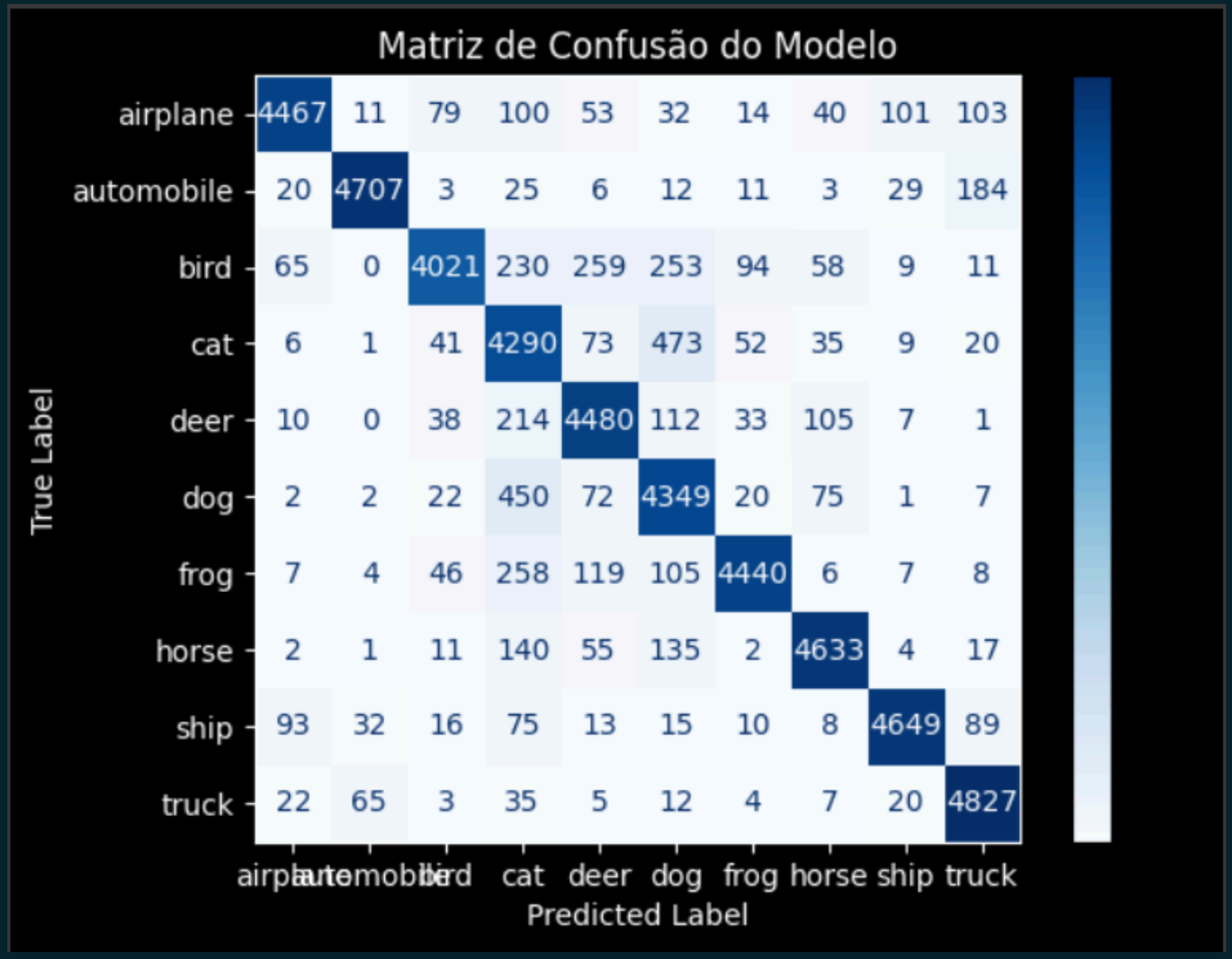
# Atividade 2 – Resultados – Teste



# Atividade 2 – Resultados – Teste

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| airplane | 0.84 | 0.77 | 0.80 | 1000 |
| automobile | 0.93 | 0.86 | 0.89 | 1000 |
| bird | 0.78 | 0.63 | 0.70 | 1000 |
| cat | 0.56 | 0.67 | 0.61 | 1000 |
| deer | 0.74 | 0.76 | 0.75 | 1000 |
| dog | 0.65 | 0.76 | 0.70 | 1000 |
| frog | 0.87 | 0.80 | 0.84 | 1000 |
| horse | 0.84 | 0.84 | 0.84 | 1000 |
| ship | 0.89 | 0.84 | 0.87 | 1000 |
| truck | 0.83 | 0.90 | 0.86 | 1000 |
| accuracy | | | 0.78 | 10000 |
| macro avg | 0.79 | 0.78 | 0.79 | 10000 |
| weighted avg | 0.79 | 0.78 | 0.79 | 10000 |

# Atividade 2 – Resultados – Treino



# Atividade 2 – Resultados – Treino

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| airplane | 0.95 | 0.89 | 0.92 | 5000 |
| automobile | 0.98 | 0.94 | 0.96 | 5000 |
| bird | 0.94 | 0.80 | 0.87 | 5000 |
| cat | 0.74 | 0.86 | 0.79 | 5000 |
| deer | 0.87 | 0.90 | 0.88 | 5000 |
| dog | 0.79 | 0.87 | 0.83 | 5000 |
| frog | 0.95 | 0.89 | 0.92 | 5000 |
| horse | 0.93 | 0.93 | 0.93 | 5000 |
| ship | 0.96 | 0.93 | 0.95 | 5000 |
| truck | 0.92 | 0.97 | 0.94 | 5000 |
| accuracy | | | 0.90 | 50000 |
| macro avg | 0.90 | 0.90 | 0.90 | 50000 |
| weighted avg | 0.90 | 0.90 | 0.90 | 50000 |

# Atividade 3
# IMDB-50K-Movie-Reviews

- Texto do review

Dados:

- 50k de reviwes

- 2 Classes
  - Positivo
  - Negativo

- Treino e Teste
  - 80% Treino
  - 20% Teste

# Atividade 2

- Pré-processamento

| | review | sentiment |
|---|---|---|
| 0 | One of the other reviewers has mentioned that ... | positive |
| 1 | A wonderful little production. <br /><br />The... | positive |
| 2 | I thought this was a wonderful way to spend ti... | positive |
| 3 | Basically there's a family where a little boy ... | negative |
| 4 | Petter Mattei's "Love in the Time of Money" is... | positive |

# Atividade 2

- Pré-processamento

| | review | sentiment |
|---|---|---|
| 0 | One of the other reviewers has mentioned that ... | 1 |
| 1 | A wonderful little production. <br /><br />The... | 1 |
| 2 | I thought this was a wonderful way to spend ti... | 1 |
| 3 | Basically there's a family where a little boy ... | 0 |
| 4 | Petter Mattei's "Love in the Time of Money" is... | 1 |

# Atividade 2

- Pré-processamento
  - Embedding: bert-base-cased
    - Truncation: true
    - Padding: POST
    - Special tokens: true
    - max_len: 512

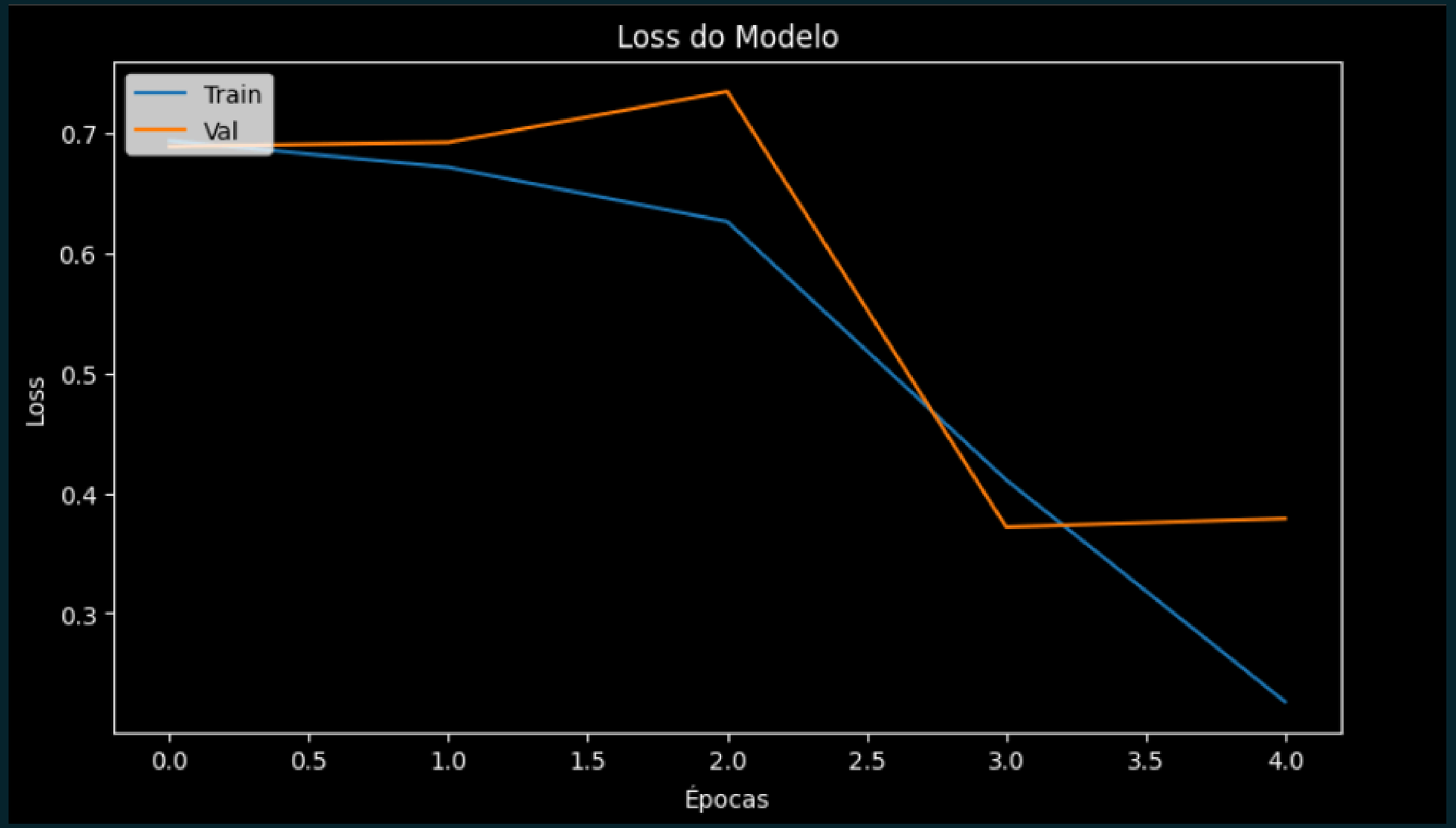
# Atividade 2

- Arquitetura e hiperparametros

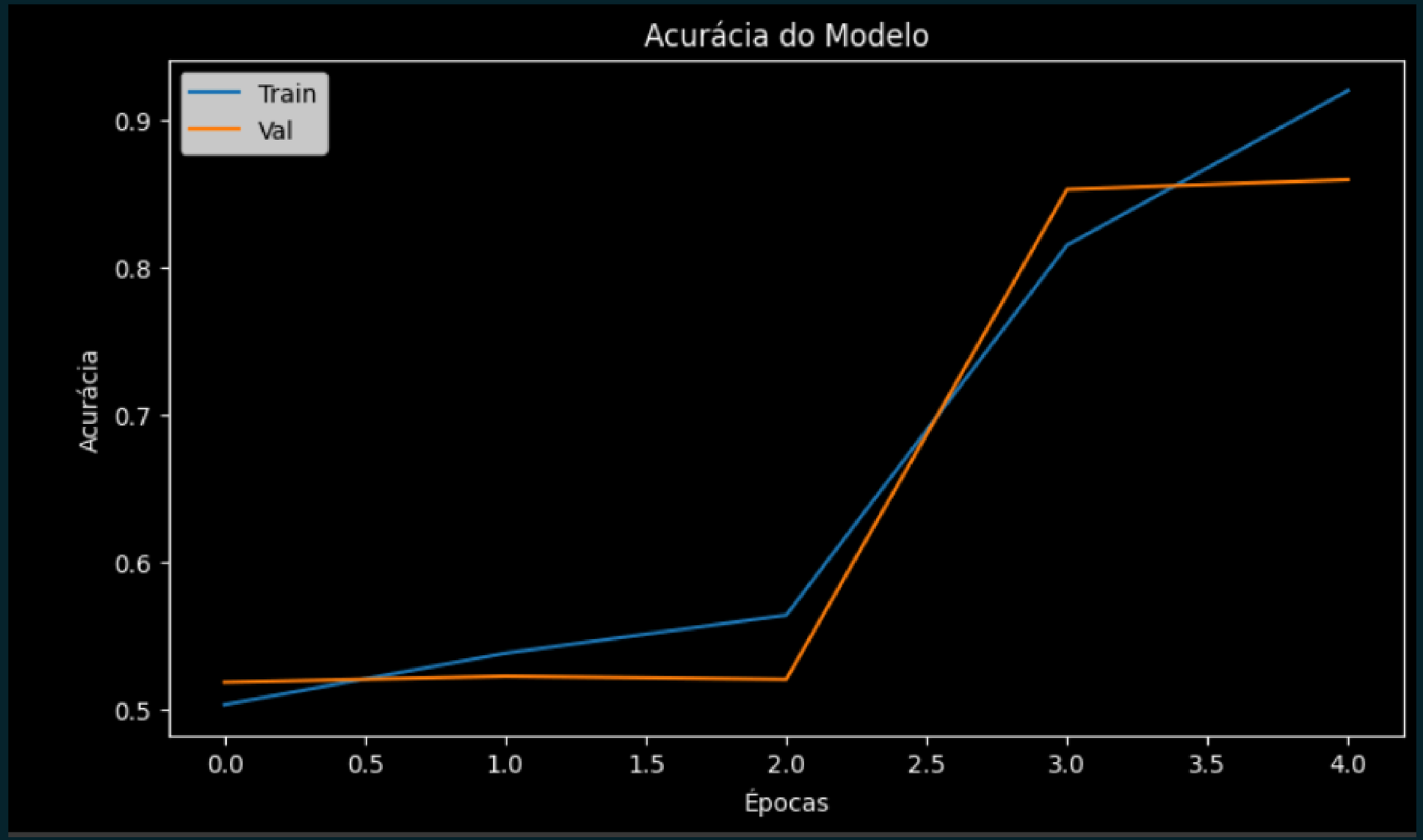
```
model = keras.Sequential([
    Embedding(input_dim=vocab_size, output_dim=embedding_dim, input_length=max_len),
    LSTM(32, return_sequences=False, input_shape=(32, 512)),
    keras.layers.Dropout(0.3),
    Dense(32, activation='relu'),
    keras.layers.Dropout(0.3),
    Dense(1, activation='sigmoid')
])
```

- Otimizador:  Adam
- Learning rate: Default(0.001)
- Loss: Binary Crossentropy
- Épocas:  5
- Batch Size: 32
- Validation Split: 0.2

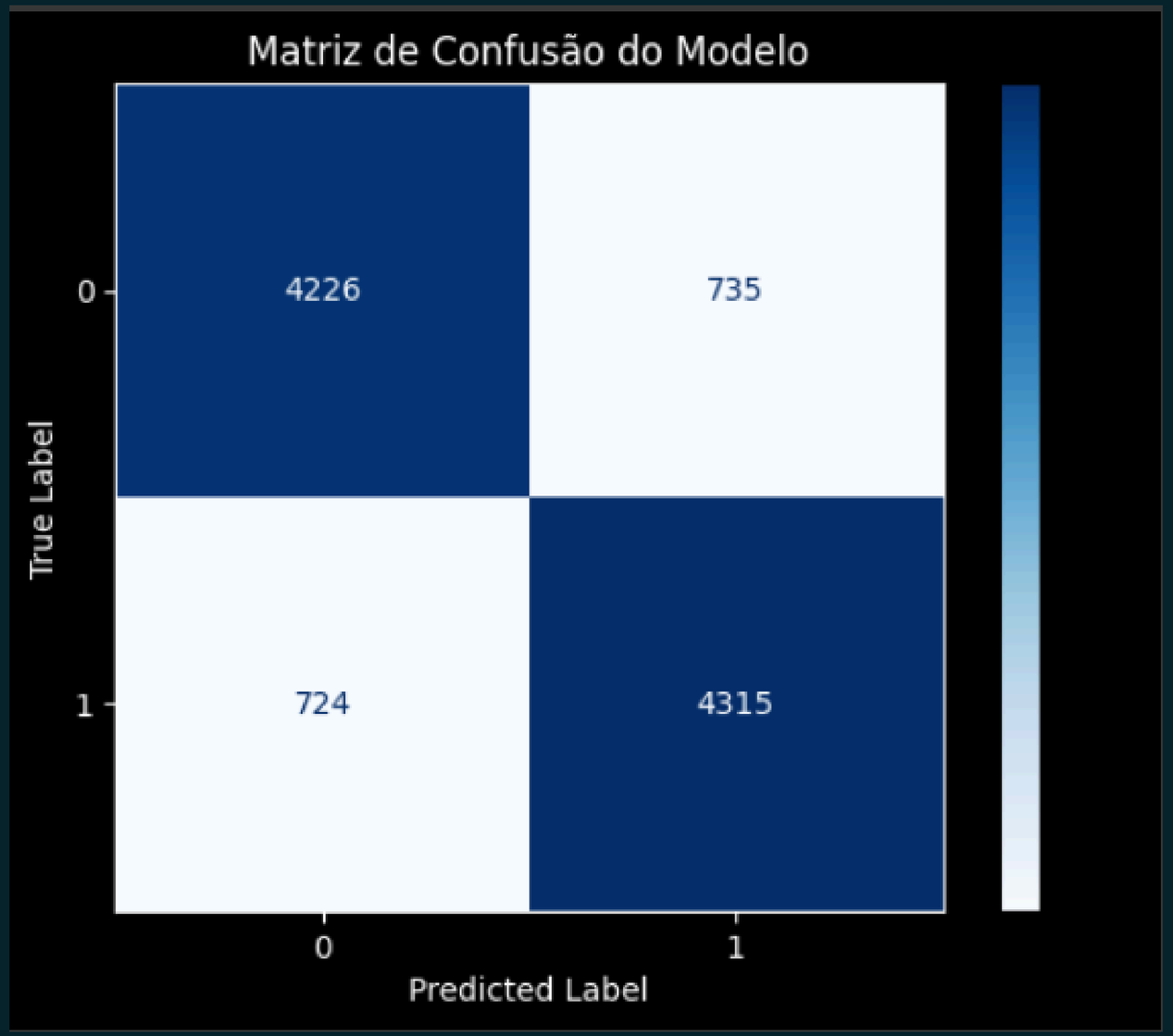
# Atividade 2 – Resultados



# Atividade 2 – Resultados



# Atividade 2 – Resultados



# Atividade 2 – Resultados

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.85 | 0.85 | 0.85 | 4961 |
| 1 | 0.85 | 0.86 | 0.86 | 5039 |
| accuracy | | | 0.85 | 10000 |
| macro avg | 0.85 | 0.85 | 0.85 | 10000 |
| weighted avg | 0.85 | 0.85 | 0.85 | 10000 |

# Dúvidas?