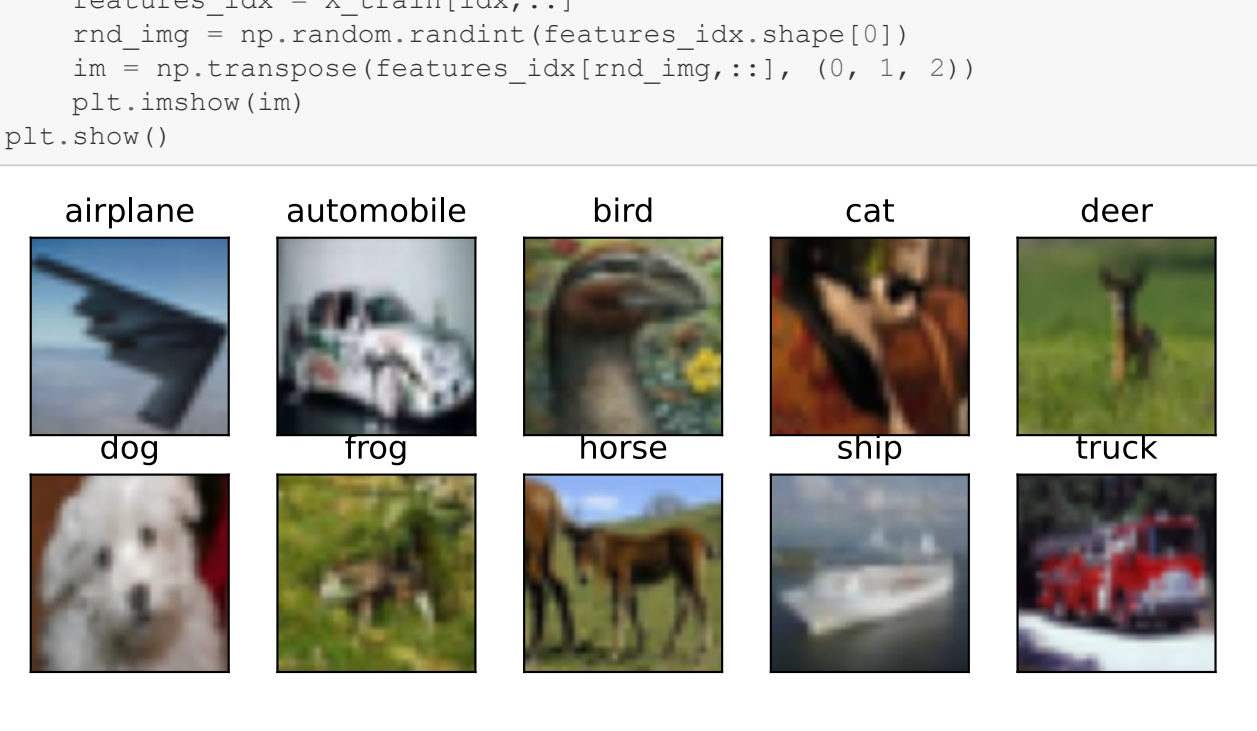
Carga de datos

In [64]:
```
# Load CIFAR-10 data set
from keras.datasets import cifar10
(X_train, y_train), (X_test, y_test) = cifar10.load_data()
```

In [65]:
```
# Show examples from each class
import numpy as np
import matplotlib.pyplot as plt

num_classes = len(np.unique(y_train))
class_names = ['airplane','automobile','bird','cat','deer','dog','frog','horse','ship','truck']
fig = plt.figure(figsize=(8,3))
for i in range(num_classes):
    ax = fig.add_subplot(2, 5, 1 + i, xticks=[], yticks=[])
    ax.set_title(class_names[i])
    idx = np.where(y_train[:]==i)[0]
    features_idx = X_train[idx,::]
    rnd_img = np.random.randint(features_idx.shape[0])
    im = np.transpose(features_idx[rnd_img,::], (0, 1, 2))
    plt.imshow(im)
plt.show()
```



In [66]:
```
# Data pre-processing
X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train /= 255.0
X_test /= 255.0

from keras.utils import np_utils
Y_train = np_utils.to_categorical(y_train, num_classes)
Y_test = np_utils.to_categorical(y_test, num_classes)
```

Utilidades

En esta sección se ejecutan todas las funciones que contienen las utilidades que se emplearán en la fase de experimentación

Dibujo de gráficas

La primera función sirve para dibujar las gráficas con el historial de entrenamiento

In [67]:
```
def plot_model_history(model_history):
    fig, axs = plt.subplots(1,2,figsize=(15,5))
    # summarize history for accuracy
    axs[0].plot(range(1,len(model_history.history['accuracy'])+1),model_history.history['accuracy'])
    axs[0].plot(range(1,len(model_history.history['val_accuracy'])+1),model_history.history['val_accura
cy'])
    axs[0].set_title('Model Accuracy')
    axs[0].set_ylabel('Accuracy')
    axs[0].set_xlabel('Epoch')
    axs[0].set_xticks(np.arange(1,len(model_history.history['accuracy'])+1,step=len(model_history.histo
ry['accuracy'])/10)
    axs[0].legend(['train', 'val'], loc='best')
    # summarize history for loss
    axs[1].plot(range(1,len(model_history.history['loss'])+1),model_history.history['loss'])
    axs[1].plot(range(1,len(model_history.history['val_loss'])+1),model_history.history['val_loss'])
    axs[1].set_title('Model Loss')
    axs[1].set_ylabel('Loss')
    axs[1].set_xlabel('Epoch')
    axs[1].set_xticks(np.arange(1,len(model_history.history['loss'])+1,step=len(model_history.history[
'loss'])/10)
    axs[1].legend(['train', 'val'], loc='best')
    plt.show()
```

Creación de modelos

A continuación se definen las funciones necesarias para la creación de los modelos que se utilizarán en la experimentación.

In [70]:
```
from tensorflow.keras import Sequential
from tensorflow.keras.models import Model
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten, BatchNormalization, Input, Con
v2D, MaxPooling2D
from tensorflow.keras.layers import experimental

def build_model_v1(dout=True, bn=True):
    model = Sequential([experimental.preprocessing.Rescaling(1./255, input_shape=(32, 32, 3)),
    experimental.preprocessing.RandomRotation(0.2),
    experimental.preprocessing.RandomZoom(.2),
    experimental.preprocessing.RandomFlip("horizontal")])

    model.add(Flatten())
    model.add(Dense(128))
    if bn:
        model.add(BatchNormalization())
    if dout:
        model.add(Dropout(.3))
    model.add(Activation('relu'))
    model.add(Dense(128))
    if bn:
        model.add(BatchNormalization())
    if dout:
        model.add(Dropout(.3))
    model.add(Activation('relu'))
    model.add(Dense(10))
    model.add(Activation('softmax'))
    return model

def build_model_v2(dout=True, bn=True):
    model = Sequential([experimental.preprocessing.Rescaling(1./255, input_shape=(32, 32, 3)),
    experimental.preprocessing.RandomRotation(0.2),
    experimental.preprocessing.RandomZoom(.2),
    experimental.preprocessing.RandomFlip("horizontal")])

    model.add(Flatten())
    model.add(Dense(128))
    if bn:
        model.add(BatchNormalization())
    if dout:
        model.add(Dropout(.3))
    model.add(Activation('relu'))
    model.add(Dense(64))
    if bn:
        model.add(BatchNormalization())
    if dout:
        model.add(Dropout(.3))
    model.add(Activation('relu'))
    model.add(Dense(32))
    if bn:
        model.add(BatchNormalization())
    if dout:
        model.add(Dropout(.3))
    model.add(Activation('relu'))
    model.add(Dense(16))
    model.add(BatchNormalization())
    if dout:
        model.add(Dropout(.3))
    model.add(Activation('relu'))
    model.add(Dense(10))
    model.add(Activation('softmax'))
    return model

def build_model_v3(dout=True, bn=True):
    model = Sequential([experimental.preprocessing.Rescaling(1./255, input_shape=(32, 32, 3)),
    experimental.preprocessing.RandomRotation(0.2),
    experimental.preprocessing.RandomZoom(.2),
    experimental.preprocessing.RandomFlip("horizontal")])

    model.add(Flatten())
    model.add(Dense(1024))
    if bn:
        model.add(BatchNormalization())
    if dout:
        model.add(Dropout(.3))
    model.add(Activation('relu'))
    model.add(Dense(512))
    if bn:
        model.add(BatchNormalization())
    if dout:
        model.add(Dropout(.3))
    model.add(Activation('relu'))
    model.add(Dense(64))
    model.add(BatchNormalization())
    if dout:
        model.add(Dropout(.3))
    model.add(Activation('relu'))
    model.add(Dense(64))
    if bn:
        model.add(BatchNormalization())
    if dout:
        model.add(Dropout(.3))
    model.add(Activation('relu'))
    model.add(Dense(10))
    model.add(Activation('softmax'))
    return model


def create_block(block_input, n_filtros, bn, bn_before):
    x = block_input
    for i in range(2):
        x = Conv2D(n_filtros, kernel_size=(3,3), padding="same")(x)
        if bn and bn_before:
            x = BatchNormalization()(x)
        x = Activation("relu")(x)
        if bn and not bn_before:
            x = BatchNormalization()(x)
    return x

def build_conv1(dout=True, bn=True, bn_before=True, extra_layer=False): # De https://www.kaggle.com/mah
tabshaan/autoencoder-as-feature-extractor-cifar10
    net_input = Input((32,32,3))
    x = net_input
    #x = experimental.preprocessing.Rescaling(1./255)(x)
    #x = experimental.preprocessing.RandomRotation(0.2)(x)
    #x = experimental.preprocessing.RandomZoom(.2)(x)
    #x = experimental.preprocessing.RandomFlip("horizontal")(x)

    block1 = create_block(x, 32, bn, bn_before)
    x = MaxPooling2D(pool_size=(2, 2))(block1)
    if dout:
        x = Dropout(0.2)(x)

    block2 = create_block(x, 64, bn, bn_before)
    x = MaxPooling2D(pool_size=(2, 2))(block2)
    if dout:
        x = Dropout(0.3)(x)

    block3 = create_block(x, 128, bn, bn_before)
    x = MaxPooling2D(pool_size=(2, 2))(block3)
    if dout:
        x = Dropout(0.3)(x)

    x = Flatten()(x)
    if extra_layer:
        x = Dense(2048)(x)
    output = Dense(10, activation='softmax')(x)
    return Model(net_input, output)
```

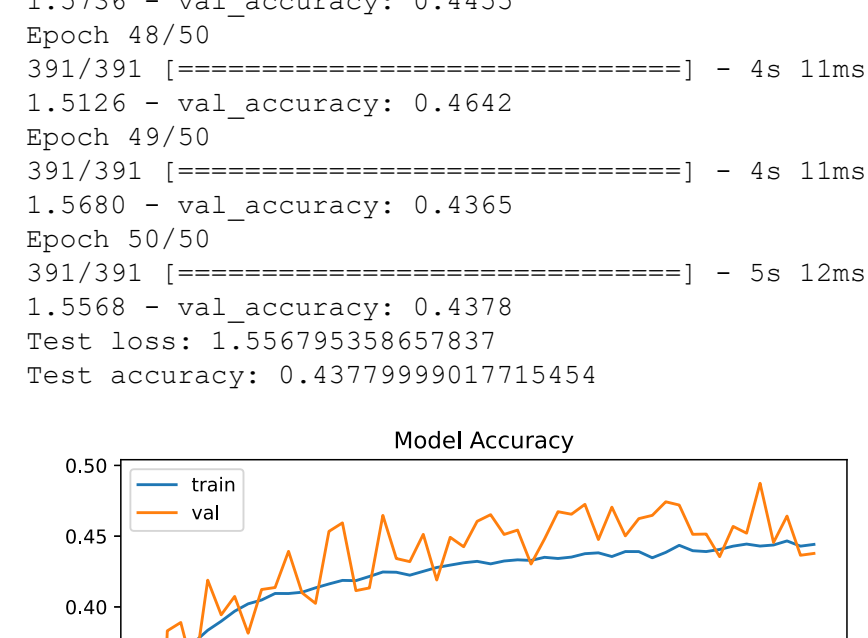
Experimentación

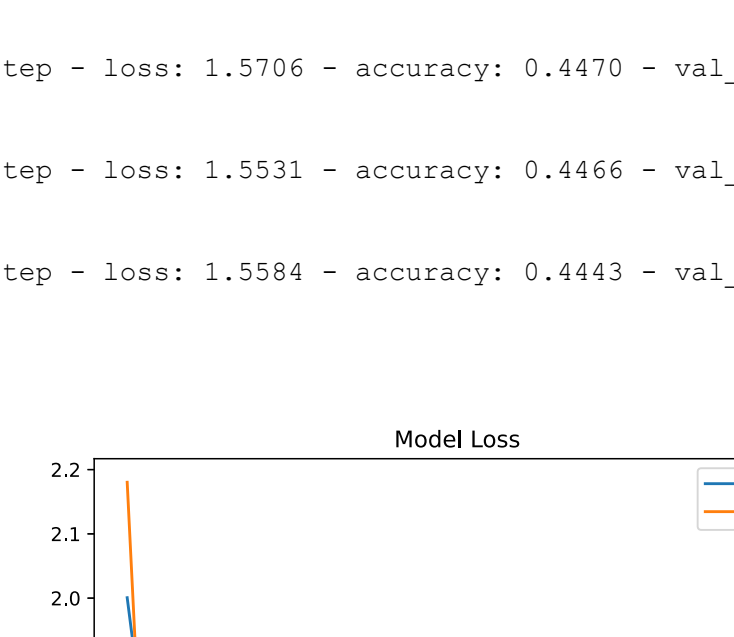
Modelo ffNN 1

Incluyendo capas de dropout y batch normalization

In [81]:
```
model = build_model_v1()

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Training
import time
start = time.time()
history = model.fit(X_train, Y_train, batch_size=128, epochs=50, verbose=1, validation_data=(X_test, Y_
test))
end = time.time()

loss, acc = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', acc)

plot_model_history(history)
print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "sequential_19"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescaling_41 (Rescaling)     (None, 32, 32, 3)         0
_________________________________________________________________
random_rotation_41 (RandomRo (None, 32, 32, 3)         0
_________________________________________________________________
random_zoom_41 (RandomZoom)  (None, 32, 32, 3)         0
_________________________________________________________________
random_flip_41 (RandomFlip)  (None, 32, 32, 3)         0
_________________________________________________________________
flatten_52 (Flatten)         (None, 3072)              0
_________________________________________________________________
dense_133 (Dense)            (None, 128)               393344
_________________________________________________________________
batch_normalization_210 (Bat (None, 128)               512
_________________________________________________________________
dropout_107 (Dropout)        (None, 128)               0
_________________________________________________________________
activation_281 (Activation)  (None, 128)               0
_________________________________________________________________
dense_134 (Dense)            (None, 128)               16512
_________________________________________________________________
batch_normalization_211 (Bat (None, 128)               512
_________________________________________________________________
dropout_108 (Dropout)        (None, 128)               0
_________________________________________________________________
activation_282 (Activation)  (None, 128)               0
_________________________________________________________________
dense_135 (Dense)            (None, 10)                1290
_________________________________________________________________
activation_283 (Activation)  (None, 10)                0
=================================================================
Total params: 412,170
Trainable params: 411,658
Non-trainable params: 512
_________________________________________________________________
Epoch 1/50
391/391 [==============================] - 7s 12ms/step - loss: 2.1124 - accuracy: 0.2479 - val_loss:
2.1803 - val_accuracy: 0.1545
Epoch 2/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8516 - accuracy: 0.3370 - val_loss:
1.7572 - val_accuracy: 0.3831
Epoch 3/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7905 - accuracy: 0.3585 - val_loss:
1.7231 - val_accuracy: 0.3890
Epoch 4/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7552 - accuracy: 0.3715 - val_loss:
1.7870 - val_accuracy: 0.3521
Epoch 5/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7296 - accuracy: 0.3805 - val_loss:
1.6360 - val_accuracy: 0.4189
Epoch 6/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7142 - accuracy: 0.3857 - val_loss:
1.7187 - val_accuracy: 0.3944
Epoch 7/50
391/391 [==============================] - 4s 11ms/step - loss: 1.6877 - accuracy: 0.3963 - val_loss:
1.6755 - val_accuracy: 0.4074
Epoch 8/50
391/391 [==============================] - 4s 11ms/step - loss: 1.6824 - accuracy: 0.3973 - val_loss:
1.7104 - val_accuracy: 0.3814
Epoch 9/50
391/391 [==============================] - 4s 11ms/step - loss: 1.6628 - accuracy: 0.4064 - val_loss:
1.6405 - val_accuracy: 0.4123
Epoch 10/50
391/391 [==============================] - 4s 11ms/step - loss: 1.6640 - accuracy: 0.4064 - val_loss:
1.6258 - val_accuracy: 0.4137
Epoch 11/50
391/391 [==============================] - 5s 12ms/step - loss: 1.6543 - accuracy: 0.4132 - val_loss:
1.5921 - val_accuracy: 0.4393
Epoch 12/50
391/391 [==============================] - 5s 12ms/step - loss: 1.6494 - accuracy: 0.4085 - val_loss:
1.6541 - val_accuracy: 0.4100
Epoch 13/50
391/391 [==============================] - 5s 12ms/step - loss: 1.6427 - accuracy: 0.4140 - val_loss:
1.6619 - val_accuracy: 0.4025
Epoch 14/50
391/391 [==============================] - 5s 12ms/step - loss: 1.6373 - accuracy: 0.4152 - val_loss:
1.5643 - val_accuracy: 0.4534
Epoch 15/50
391/391 [==============================] - 5s 12ms/step - loss: 1.6301 - accuracy: 0.4169 - val_loss:
1.5682 - val_accuracy: 0.4594
Epoch 16/50
391/391 [==============================] - 5s 12ms/step - loss: 1.6207 - accuracy: 0.4218 - val_loss:
1.6796 - val_accuracy: 0.4115
Epoch 17/50
391/391 [==============================] - 5s 12ms/step - loss: 1.6232 - accuracy: 0.4198 - val_loss:
1.6229 - val_accuracy: 0.4134
Epoch 18/50
391/391 [==============================] - 4s 11ms/step - loss: 1.6171 - accuracy: 0.4229 - val_loss:
1.5431 - val_accuracy: 0.4647
Epoch 19/50
391/391 [==============================] - 4s 11ms/step - loss: 1.6100 - accuracy: 0.4265 - val_loss:
1.6095 - val_accuracy: 0.4362
Epoch 20/50
391/391 [==============================] - 4s 11ms/step - loss: 1.6090 - accuracy: 0.4227 - val_loss:
1.5918 - val_accuracy: 0.4320
Epoch 21/50
391/391 [==============================] - 4s 11ms/step - loss: 1.6051 - accuracy: 0.4244 - val_loss:
1.5660 - val_accuracy: 0.4513
Epoch 22/50
391/391 [==============================] - 4s 11ms/step - loss: 1.6127 - accuracy: 0.4291 - val_loss:
1.6151 - val_accuracy: 0.4190
Epoch 23/50
391/391 [==============================] - 5s 12ms/step - loss: 1.6115 - accuracy: 0.4272 - val_loss:
1.5275 - val_accuracy: 0.4492
Epoch 24/50
391/391 [==============================] - 5s 12ms/step - loss: 1.5953 - accuracy: 0.4323 - val_loss:
1.5715 - val_accuracy: 0.4425
Epoch 25/50
391/391 [==============================] - 7s 18ms/step - loss: 1.5961 - accuracy: 0.4307 - val_loss:
1.5492 - val_accuracy: 0.4605
Epoch 26/50
391/391 [==============================] - 7s 17ms/step - loss: 1.5990 - accuracy: 0.4278 - val_loss:
1.5002 - val_accuracy: 0.4652
Epoch 27/50
391/391 [==============================] - 5s 12ms/step - loss: 1.5886 - accuracy: 0.4352 - val_loss:
1.5515 - val_accuracy: 0.4512
Epoch 28/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5968 - accuracy: 0.4338 - val_loss:
1.5485 - val_accuracy: 0.4543
Epoch 29/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5904 - accuracy: 0.4336 - val_loss:
1.5779 - val_accuracy: 0.4303
Epoch 30/50
391/391 [==============================] - 5s 12ms/step - loss: 1.5898 - accuracy: 0.4374 - val_loss:
1.5438 - val_accuracy: 0.4480
Epoch 31/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5836 - accuracy: 0.4364 - val_loss:
1.5019 - val_accuracy: 0.4673
Epoch 32/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5897 - accuracy: 0.4351 - val_loss:
1.5450 - val_accuracy: 0.4655
Epoch 33/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5823 - accuracy: 0.4393 - val_loss:
1.4923 - val_accuracy: 0.4725
Epoch 34/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5786 - accuracy: 0.4391 - val_loss:
1.5403 - val_accuracy: 0.4476
Epoch 35/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5836 - accuracy: 0.4348 - val_loss:
1.5075 - val_accuracy: 0.4705
Epoch 36/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5693 - accuracy: 0.4432 - val_loss:
1.5429 - val_accuracy: 0.4502
Epoch 37/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5835 - accuracy: 0.4391 - val_loss:
1.5056 - val_accuracy: 0.4624
Epoch 38/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5822 - accuracy: 0.4359 - val_loss:
1.5100 - val_accuracy: 0.4647
Epoch 39/50
391/391 [==============================] - 5s 12ms/step - loss: 1.5834 - accuracy: 0.4343 - val_loss:
1.5072 - val_accuracy: 0.4743
Epoch 40/50
391/391 [==============================] - 5s 12ms/step - loss: 1.5721 - accuracy: 0.4415 - val_loss:
1.5016 - val_accuracy: 0.4720
Epoch 41/50
391/391 [==============================] - 5s 12ms/step - loss: 1.5713 - accuracy: 0.4411 - val_loss:
1.5422 - val_accuracy: 0.4513
Epoch 42/50
391/391 [==============================] - 5s 12ms/step - loss: 1.5741 - accuracy: 0.4373 - val_loss:
1.5427 - val_accuracy: 0.4515
Epoch 43/50
391/391 [==============================] - 5s 12ms/step - loss: 1.5729 - accuracy: 0.4414 - val_loss:
1.5638 - val_accuracy: 0.4355
Epoch 44/50
391/391 [==============================] - 5s 12ms/step - loss: 1.5758 - accuracy: 0.4405 - val_loss:
1.5128 - val_accuracy: 0.4569
Epoch 45/50
391/391 [==============================] - 5s 12ms/step - loss: 1.5604 - accuracy: 0.4469 - val_loss:
1.5417 - val_accuracy: 0.4521
Epoch 46/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5636 - accuracy: 0.4443 - val_loss:
1.4474 - val_accuracy: 0.4874
Epoch 47/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5662 - accuracy: 0.4486 - val_loss:
1.5736 - val_accuracy: 0.4455
Epoch 48/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5706 - accuracy: 0.4470 - val_loss:
1.5126 - val_accuracy: 0.4642
Epoch 49/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5531 - accuracy: 0.4466 - val_loss:
1.5680 - val_accuracy: 0.4365
Epoch 50/50
391/391 [==============================] - 5s 12ms/step - loss: 1.5584 - accuracy: 0.4443 - val_loss:
1.5568 - val_accuracy: 0.4378
Test loss: 1.556795358657837
Test accuracy: 0.43779999017715454
```






```
Training MLP took 232.05204224586487 seconds
```

Con capa de dropout pero sin batch normalization

In [82]:
```
model = build_model_v1(bn=False)

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Training
import time
start = time.time()
history = model.fit(X_train, Y_train, batch_size=128, epochs=50, verbose=1, validation_data=(X_test, Y_
test))
end = time.time()

loss, acc = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', acc)

plot_model_history(history)
print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "sequential_20"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescaling_42 (Rescaling)     (None, 32, 32, 3)         0
_________________________________________________________________
random_rotation_42 (RandomRo (None, 32, 32, 3)         0
_________________________________________________________________
random_zoom_42 (RandomZoom)  (None, 32, 32, 3)         0
_________________________________________________________________
random_flip_42 (RandomFlip)  (None, 32, 32, 3)         0
_________________________________________________________________
flatten_53 (Flatten)         (None, 3072)              0
_________________________________________________________________
dense_136 (Dense)            (None, 128)               393344
_________________________________________________________________
dropout_109 (Dropout)        (None, 128)               0
_________________________________________________________________
activation_284 (Activation)  (None, 128)               0
_________________________________________________________________
dense_137 (Dense)            (None, 128)               16512
_________________________________________________________________
dropout_110 (Dropout)        (None, 128)               0
_________________________________________________________________
activation_285 (Activation)  (None, 128)               0
_________________________________________________________________
dense_138 (Dense)            (None, 10)                1290
_________________________________________________________________
activation_286 (Activation)  (None, 10)                0
=================================================================
Total params: 411,146
Trainable params: 411,146
Non-trainable params: 0
_________________________________________________________________
Epoch 1/50
391/391 [==============================] - 6s 11ms/step - loss: 2.2251 - accuracy: 0.1540 - val_loss:
1.9946 - val_accuracy: 0.2706
Epoch 2/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0302 - accuracy: 0.2526 - val_loss:
1.9351 - val_accuracy: 0.2959
Epoch 3/50
391/391 [==============================] - 4s 11ms/step - loss: 1.9816 - accuracy: 0.2756 - val_loss:
1.8946 - val_accuracy: 0.3150
Epoch 4/50
391/391 [==============================] - 4s 11ms/step - loss: 1.9575 - accuracy: 0.2860 - val_loss:
1.8598 - val_accuracy: 0.3264
Epoch 5/50
391/391 [==============================] - 4s 11ms/step - loss: 1.9385 - accuracy: 0.2971 - val_loss:
1.8419 - val_accuracy: 0.3389
Epoch 6/50
391/391 [==============================] - 4s 11ms/step - loss: 1.9181 - accuracy: 0.3044 - val_loss:
1.8361 - val_accuracy: 0.3383
Epoch 7/50
391/391 [==============================] - 4s 11ms/step - loss: 1.9045 - accuracy: 0.3100 - val_loss:
1.8009 - val_accuracy: 0.3571
Epoch 8/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8953 - accuracy: 0.3135 - val_loss:
1.7866 - val_accuracy: 0.3666
Epoch 9/50
391/391 [==============================] - 4s 10ms/step - loss: 1.8854 - accuracy: 0.3222 - val_loss:
1.7699 - val_accuracy: 0.3710
Epoch 10/50
391/391 [==============================] - 4s 10ms/step - loss: 1.8700 - accuracy: 0.3288 - val_loss:
1.7627 - val_accuracy: 0.3694
Epoch 11/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8627 - accuracy: 0.3282 - val_loss:
1.7361 - val_accuracy: 0.3808
Epoch 12/50
391/391 [==============================] - 4s 10ms/step - loss: 1.8536 - accuracy: 0.3319 - val_loss:
1.7237 - val_accuracy: 0.3873
Epoch 13/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8484 - accuracy: 0.3314 - val_loss:
1.7180 - val_accuracy: 0.3859
Epoch 14/50
391/391 [==============================] - 5s 12ms/step - loss: 1.8380 - accuracy: 0.3406 - val_loss:
1.7035 - val_accuracy: 0.3951
Epoch 15/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8334 - accuracy: 0.3428 - val_loss:
1.6922 - val_accuracy: 0.3979
Epoch 16/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8236 - accuracy: 0.3434 - val_loss:
1.6803 - val_accuracy: 0.4003
Epoch 17/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8187 - accuracy: 0.3484 - val_loss:
1.6780 - val_accuracy: 0.3982
Epoch 18/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8020 - accuracy: 0.3500 - val_loss:
1.6783 - val_accuracy: 0.4006
Epoch 19/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8143 - accuracy: 0.3505 - val_loss:
1.6596 - val_accuracy: 0.4082
Epoch 20/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8001 - accuracy: 0.3568 - val_loss:
1.6541 - val_accuracy: 0.4139
Epoch 21/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7920 - accuracy: 0.3541 - val_loss:
1.6499 - val_accuracy: 0.4131
Epoch 22/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7932 - accuracy: 0.3557 - val_loss:
1.6482 - val_accuracy: 0.4121
Epoch 23/50
391/391 [==============================] - 4s 10ms/step - loss: 1.7885 - accuracy: 0.3574 - val_loss:
1.6437 - val_accuracy: 0.4183
Epoch 24/50
391/391 [==============================] - 4s 10ms/step - loss: 1.7870 - accuracy: 0.3586 - val_loss:
1.6462 - val_accuracy: 0.4200
Epoch 25/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7767 - accuracy: 0.3614 - val_loss:
1.6318 - val_accuracy: 0.4170
Epoch 26/50
391/391 [==============================] - 4s 10ms/step - loss: 1.7768 - accuracy: 0.3635 - val_loss:
1.6322 - val_accuracy: 0.4228
Epoch 27/50
391/391 [==============================] - 4s 10ms/step - loss: 1.7818 - accuracy: 0.3614 - val_loss:
1.6323 - val_accuracy: 0.4195
Epoch 28/50
391/391 [==============================] - 4s 10ms/step - loss: 1.7702 - accuracy: 0.3660 - val_loss:
1.6278 - val_accuracy: 0.4212
Epoch 29/50
391/391 [==============================] - 4s 10ms/step - loss: 1.7740 - accuracy: 0.3653 - val_loss:
1.6246 - val_accuracy: 0.4236
Epoch 30/50
391/391 [==============================] - 5s 12ms/step - loss: 1.7664 - accuracy: 0.3664 - val_loss:
1.6213 - val_accuracy: 0.4312
Epoch 31/50
391/391 [==============================] - 4s 10ms/step - loss: 1.7642 - accuracy: 0.3673 - val_loss:
1.6282 - val_accuracy: 0.4195
Epoch 32/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7603 - accuracy: 0.3678 - val_loss:
1.6110 - val_accuracy: 0.4331
Epoch 33/50
391/391 [==============================] - 5s 12ms/step - loss: 1.7640 - accuracy: 0.3689 - val_loss:
1.6064 - val_accuracy: 0.4321
Epoch 34/50
391/391 [==============================] - 5s 12ms/step - loss: 1.7459 - accuracy: 0.3754 - val_loss:
1.6129 - val_accuracy: 0.4268
Epoch 35/50
391/391 [==============================] - 5s 12ms/step - loss: 1.7533 - accuracy: 0.3732 - val_loss:
1.6026 - val_accuracy: 0.4285
Epoch 36/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7524 - accuracy: 0.3692 - val_loss:
1.5985 - val_accuracy: 0.4342
Epoch 37/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7535 - accuracy: 0.3769 - val_loss:
1.6008 - val_accuracy: 0.4323
Epoch 38/50
391/391 [==============================] - 4s 10ms/step - loss: 1.7553 - accuracy: 0.3739 - val_loss:
1.6002 - val_accuracy: 0.4352
Epoch 39/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7503 - accuracy: 0.3703 - val_loss:
1.5993 - val_accuracy: 0.4311
Epoch 40/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7429 - accuracy: 0.3725 - val_loss:
1.5967 - val_accuracy: 0.4366
Epoch 41/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7353 - accuracy: 0.3783 - val_loss:
1.5908 - val_accuracy: 0.4404
Epoch 42/50
391/391 [==============================] - 4s 10ms/step - loss: 1.7414 - accuracy: 0.3739 - val_loss:
1.5944 - val_accuracy: 0.4355
Epoch 43/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7420 - accuracy: 0.3766 - val_loss:
1.5980 - val_accuracy: 0.4324
Epoch 44/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7365 - accuracy: 0.3766 - val_loss:
1.5872 - val_accuracy: 0.4368
Epoch 45/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7337 - accuracy: 0.3821 - val_loss:
1.5855 - val_accuracy: 0.4402
Epoch 46/50
391/391 [==============================] - 5s 12ms/step - loss: 1.7360 - accuracy: 0.3737 - val_loss:
1.5998 - val_accuracy: 0.4283
Epoch 47/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7349 - accuracy: 0.3797 - val_loss:
1.5850 - val_accuracy: 0.4361
Epoch 48/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7362 - accuracy: 0.3755 - val_loss:
1.5849 - val_accuracy: 0.4379
Epoch 49/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7323 - accuracy: 0.3784 - val_loss:
1.5885 - val_accuracy: 0.4311
Epoch 50/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7282 - accuracy: 0.3814 - val_loss:
1.5901 - val_accuracy: 0.4353
Test loss: 1.590074062347412
Test accuracy: 0.43529999256134033
```

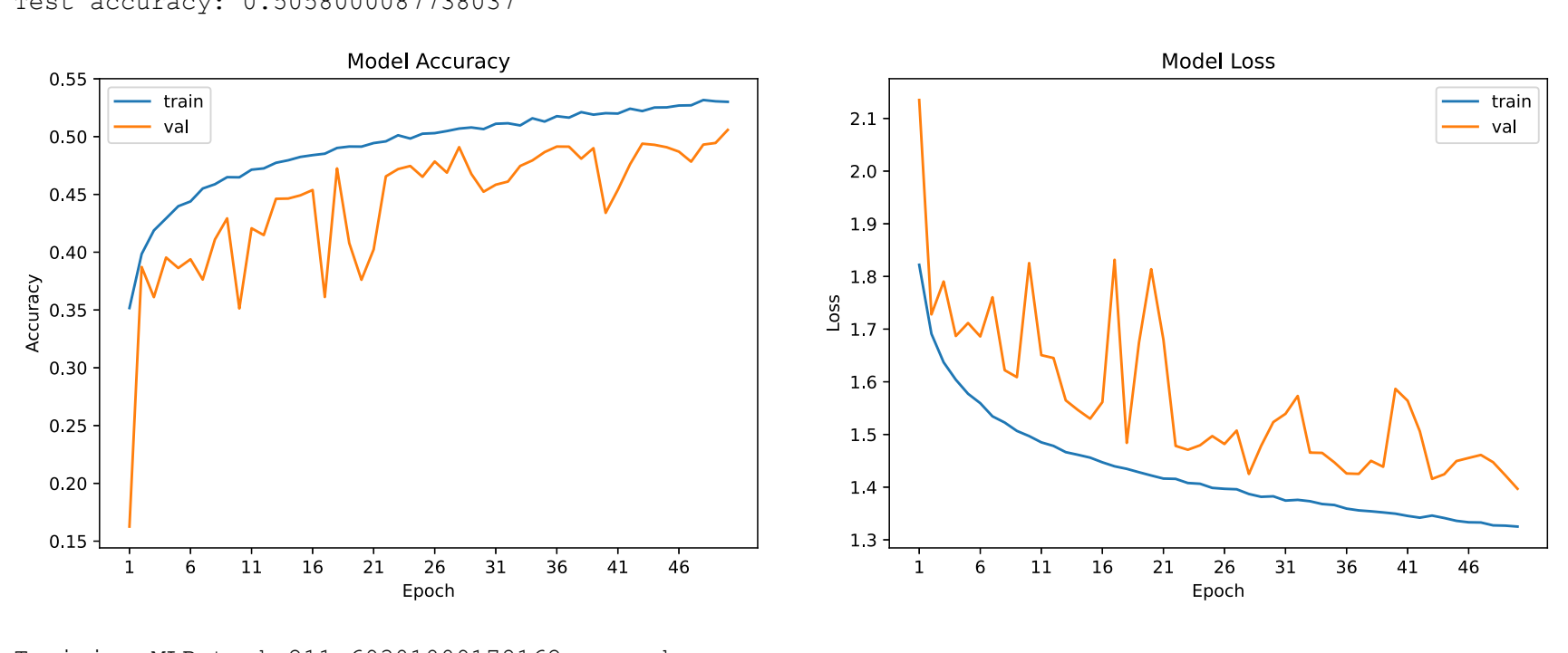


```
Training MLP took 213.21921491622925 seconds
```

Con capa de batch normalization pero sin dropout

In [83]:
```python
model = build_model_v1(dout=False)

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Training
import time
start = time.time()
history = model.fit(X_train, Y_train, batch_size=128, epochs=50, verbose=1, validation_data=(X_test, Y_test))
end = time.time()

loss, acc = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', acc)

plot_model_history(history)
print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "sequential_21"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescaling_43 (Rescaling)     (None, 32, 32, 3)         0
_________________________________________________________________
random_rotation_43 (RandomRo (None, 32, 32, 3)         0
_________________________________________________________________
random_zoom_43 (RandomZoom)  (None, 32, 32, 3)         0
_________________________________________________________________
random_flip_43 (RandomFlip)  (None, 32, 32, 3)         0
_________________________________________________________________
flatten_54 (Flatten)         (None, 3072)              0
_________________________________________________________________
dense_139 (Dense)            (None, 128)               393344
_________________________________________________________________
batch_normalization_212 (Bat (None, 128)               512
_________________________________________________________________
activation_287 (Activation)  (None, 128)               0
_________________________________________________________________
dense_140 (Dense)            (None, 128)               16512
_________________________________________________________________
batch_normalization_213 (Bat (None, 128)               512
_________________________________________________________________
activation_288 (Activation)  (None, 128)               0
_________________________________________________________________
dense_141 (Dense)            (None, 10)                1290
_________________________________________________________________
activation_289 (Activation)  (None, 10)                0
=================================================================
Total params: 412,170
Trainable params: 411,658
Non-trainable params: 512
_________________________________________________________________
Epoch 1/50
391/391 [==============================] - 6s 12ms/step - loss: 1.9234 - accuracy: 0.3122 - val_loss: 2.1350 - val_accuracy: 0.1626
Epoch 2/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7169 - accuracy: 0.3871 - val_loss: 1.7279 - val_accuracy: 0.3872
Epoch 3/50
391/391 [==============================] - 4s 11ms/step - loss: 1.6423 - accuracy: 0.4151 - val_loss: 1.7905 - val_accuracy: 0.3611
Epoch 4/50
391/391 [==============================] - 4s 10ms/step - loss: 1.6144 - accuracy: 0.4225 - val_loss: 1.6869 - val_accuracy: 0.3954
Epoch 5/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5861 - accuracy: 0.4363 - val_loss: 1.7116 - val_accuracy: 0.3863
Epoch 6/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5616 - accuracy: 0.4418 - val_loss: 1.6860 - val_accuracy: 0.3939
Epoch 7/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5421 - accuracy: 0.4542 - val_loss: 1.7606 - val_accuracy: 0.3763
Epoch 8/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5202 - accuracy: 0.4580 - val_loss: 1.6220 - val_accuracy: 0.4110
Epoch 9/50
391/391 [==============================] - 4s 10ms/step - loss: 1.5067 - accuracy: 0.4641 - val_loss: 1.6088 - val_accuracy: 0.4293
Epoch 10/50
391/391 [==============================] - 4s 10ms/step - loss: 1.4960 - accuracy: 0.4649 - val_loss: 1.8254 - val_accuracy: 0.3513
Epoch 11/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4843 - accuracy: 0.4725 - val_loss: 1.6506 - val_accuracy: 0.4207
Epoch 12/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4820 - accuracy: 0.4730 - val_loss: 1.6452 - val_accuracy: 0.4148
Epoch 13/50
391/391 [==============================] - 5s 12ms/step - loss: 1.4624 - accuracy: 0.4779 - val_loss: 1.5648 - val_accuracy: 0.4462
Epoch 14/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4630 - accuracy: 0.4804 - val_loss: 1.5464 - val_accuracy: 0.4464
Epoch 15/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4600 - accuracy: 0.4788 - val_loss: 1.5299 - val_accuracy: 0.4492
Epoch 16/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4400 - accuracy: 0.4841 - val_loss: 1.5614 - val_accuracy: 0.4538
Epoch 17/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4301 - accuracy: 0.4876 - val_loss: 1.8317 - val_accuracy: 0.3612
Epoch 18/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4311 - accuracy: 0.4918 - val_loss: 1.4840 - val_accuracy: 0.4725
Epoch 19/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4194 - accuracy: 0.4960 - val_loss: 1.6745 - val_accuracy: 0.4079
Epoch 20/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4211 - accuracy: 0.4911 - val_loss: 1.8138 - val_accuracy: 0.3761
Epoch 21/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4182 - accuracy: 0.4960 - val_loss: 1.6799 - val_accuracy: 0.4024
Epoch 22/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4091 - accuracy: 0.4974 - val_loss: 1.4783 - val_accuracy: 0.4656
Epoch 23/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4012 - accuracy: 0.5042 - val_loss: 1.4709 - val_accuracy: 0.4719
Epoch 24/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4017 - accuracy: 0.5000 - val_loss: 1.4795 - val_accuracy: 0.4746
Epoch 25/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3912 - accuracy: 0.5070 - val_loss: 1.4970 - val_accuracy: 0.4652
Epoch 26/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3903 - accuracy: 0.5041 - val_loss: 1.4819 - val_accuracy: 0.4785
Epoch 27/50
391/391 [==============================] - 5s 12ms/step - loss: 1.3928 - accuracy: 0.5064 - val_loss: 1.5074 - val_accuracy: 0.4688
Epoch 28/50
391/391 [==============================] - 5s 12ms/step - loss: 1.3827 - accuracy: 0.5080 - val_loss: 1.4250 - val_accuracy: 0.4909
Epoch 29/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3746 - accuracy: 0.5099 - val_loss: 1.4783 - val_accuracy: 0.4677
Epoch 30/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3811 - accuracy: 0.5070 - val_loss: 1.5235 - val_accuracy: 0.4523
Epoch 31/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3773 - accuracy: 0.5105 - val_loss: 1.5393 - val_accuracy: 0.4584
Epoch 32/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3745 - accuracy: 0.5123 - val_loss: 1.5731 - val_accuracy: 0.4611
Epoch 33/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3776 - accuracy: 0.5087 - val_loss: 1.4656 - val_accuracy: 0.4746
Epoch 34/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3683 - accuracy: 0.5143 - val_loss: 1.4650 - val_accuracy: 0.4794
Epoch 35/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3633 - accuracy: 0.5138 - val_loss: 1.4470 - val_accuracy: 0.4866
Epoch 36/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3588 - accuracy: 0.5178 - val_loss: 1.4259 - val_accuracy: 0.4914
Epoch 37/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3517 - accuracy: 0.5178 - val_loss: 1.4252 - val_accuracy: 0.4913
Epoch 38/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3554 - accuracy: 0.5192 - val_loss: 1.4500 - val_accuracy: 0.4809
Epoch 39/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3498 - accuracy: 0.5184 - val_loss: 1.4387 - val_accuracy: 0.4900
Epoch 40/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3510 - accuracy: 0.5178 - val_loss: 1.5867 - val_accuracy: 0.4340
Epoch 41/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3420 - accuracy: 0.5212 - val_loss: 1.5641 - val_accuracy: 0.4540
Epoch 42/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3429 - accuracy: 0.5249 - val_loss: 1.5060 - val_accuracy: 0.4761
Epoch 43/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3426 - accuracy: 0.5236 - val_loss: 1.4156 - val_accuracy: 0.4939
Epoch 44/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3406 - accuracy: 0.5227 - val_loss: 1.4245 - val_accuracy: 0.4929
Epoch 45/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3393 - accuracy: 0.5254 - val_loss: 1.4495 - val_accuracy: 0.4908
Epoch 46/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3294 - accuracy: 0.5305 - val_loss: 1.4554 - val_accuracy: 0.4870
Epoch 47/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3327 - accuracy: 0.5280 - val_loss: 1.4611 - val_accuracy: 0.4783
Epoch 48/50
391/391 [==============================] - 4s 10ms/step - loss: 1.3261 - accuracy: 0.5311 - val_loss: 1.4473 - val_accuracy: 0.4931
Epoch 49/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3199 - accuracy: 0.5295 - val_loss: 1.4225 - val_accuracy: 0.4945
Epoch 50/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3106 - accuracy: 0.5357 - val_loss: 1.3969 - val_accuracy: 0.5058
Test loss: 1.396878957748413
Test accuracy: 0.5058000087738037
```



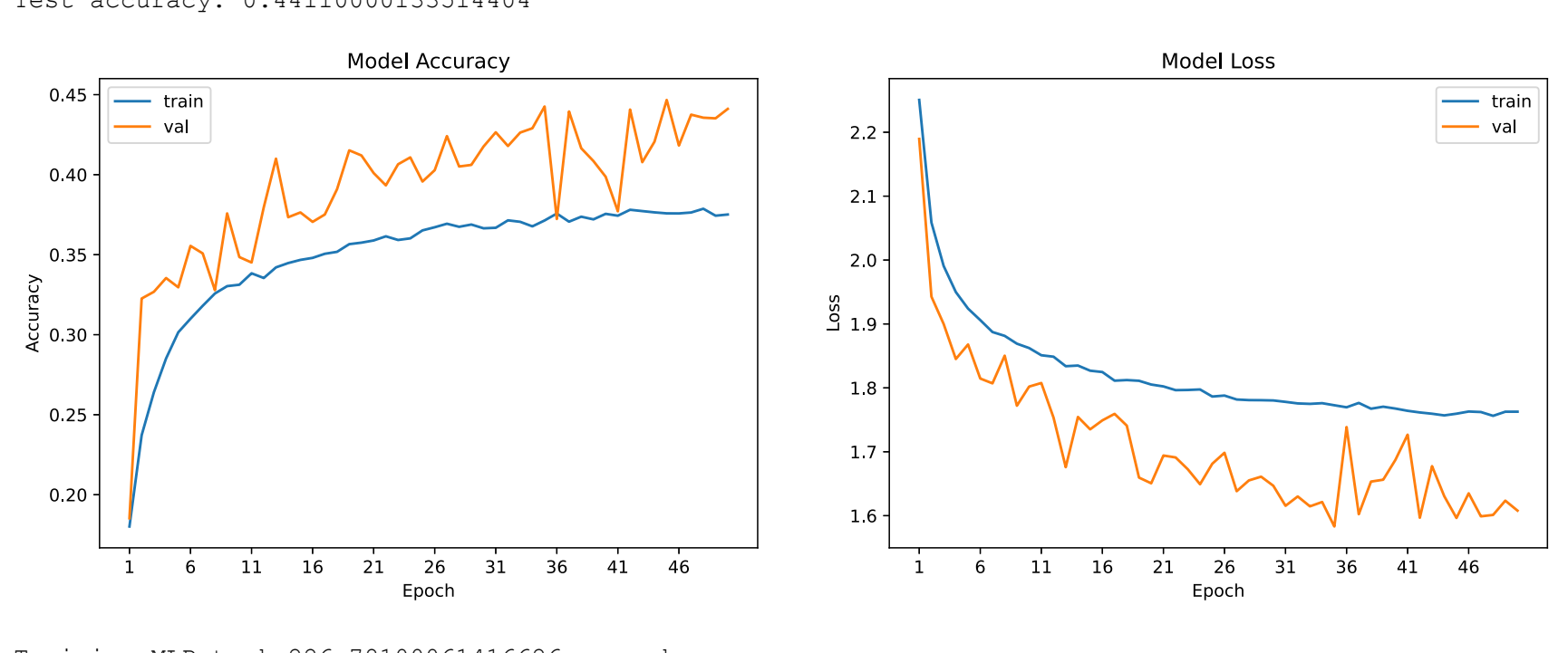

```
Training MLP took 211.69301009178162 seconds
```

Modelo ffNN 2

Incluyendo capas de dropout y batch normalization

In [84]:
```python
model = build_model_v2()

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Training
import time
start = time.time()
history = model.fit(X_train, Y_train, batch_size=128, epochs=50, verbose=1, validation_data=(X_test, Y_test))
end = time.time()

loss, acc = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', acc)

plot_model_history(history)
print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "sequential_22"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescaling_44 (Rescaling)     (None, 32, 32, 3)         0
_________________________________________________________________
random_rotation_44 (RandomRo (None, 32, 32, 3)         0
_________________________________________________________________
random_zoom_44 (RandomZoom)  (None, 32, 32, 3)         0
_________________________________________________________________
random_flip_44 (RandomFlip)  (None, 32, 32, 3)         0
_________________________________________________________________
flatten_55 (Flatten)         (None, 3072)              0
_________________________________________________________________
dense_142 (Dense)            (None, 128)               393344
_________________________________________________________________
batch_normalization_214 (Bat (None, 128)               512
_________________________________________________________________
dropout_111 (Dropout)        (None, 128)               0
_________________________________________________________________
activation_290 (Activation)  (None, 128)               0
_________________________________________________________________
dense_143 (Dense)            (None, 64)                8256
_________________________________________________________________
batch_normalization_215 (Bat (None, 64)                256
_________________________________________________________________
dropout_112 (Dropout)        (None, 64)                0
_________________________________________________________________
activation_291 (Activation)  (None, 64)                0
_________________________________________________________________
dense_144 (Dense)            (None, 32)                2080
_________________________________________________________________
batch_normalization_216 (Bat (None, 32)                128
_________________________________________________________________
dropout_113 (Dropout)        (None, 32)                0
_________________________________________________________________
activation_292 (Activation)  (None, 32)                0
_________________________________________________________________
dense_145 (Dense)            (None, 16)                528
_________________________________________________________________
batch_normalization_217 (Bat (None, 16)                64
_________________________________________________________________
dropout_114 (Dropout)        (None, 16)                0
_________________________________________________________________
activation_293 (Activation)  (None, 16)                0
_________________________________________________________________
dense_146 (Dense)            (None, 10)                170
_________________________________________________________________
activation_294 (Activation)  (None, 10)                0
=================================================================
Total params: 405,338
Trainable params: 404,858
Non-trainable params: 480
_________________________________________________________________
Epoch 1/50
391/391 [==============================] - 7s 12ms/step - loss: 2.3825 - accuracy: 0.1520 - val_loss: 2.1895 - val_accuracy: 0.1853
Epoch 2/50
391/391 [==============================] - 5s 12ms/step - loss: 2.0833 - accuracy: 0.2269 - val_loss: 1.9428 - val_accuracy: 0.3226
Epoch 3/50
391/391 [==============================] - 5s 12ms/step - loss: 2.0009 - accuracy: 0.2584 - val_loss: 1.9000 - val_accuracy: 0.3268
Epoch 4/50
391/391 [==============================] - 5s 12ms/step - loss: 1.9595 - accuracy: 0.2791 - val_loss: 1.8451 - val_accuracy: 0.3354
Epoch 5/50
391/391 [==============================] - 5s 12ms/step - loss: 1.9350 - accuracy: 0.2950 - val_loss: 1.8679 - val_accuracy: 0.3296
Epoch 6/50
391/391 [==============================] - 5s 12ms/step - loss: 1.9025 - accuracy: 0.3109 - val_loss: 1.8146 - val_accuracy: 0.3555
Epoch 7/50
391/391 [==============================] - 5s 12ms/step - loss: 1.8985 - accuracy: 0.3144 - val_loss: 1.8071 - val_accuracy: 0.3508
Epoch 8/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8793 - accuracy: 0.3263 - val_loss: 1.8504 - val_accuracy: 0.3278
Epoch 9/50
391/391 [==============================] - 5s 12ms/step - loss: 1.8683 - accuracy: 0.3316 - val_loss: 1.7721 - val_accuracy: 0.3758
Epoch 10/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8663 - accuracy: 0.3304 - val_loss: 1.8020 - val_accuracy: 0.3485
Epoch 11/50
391/391 [==============================] - 5s 12ms/step - loss: 1.8484 - accuracy: 0.3397 - val_loss: 1.8076 - val_accuracy: 0.3451
Epoch 12/50
391/391 [==============================] - 5s 12ms/step - loss: 1.8514 - accuracy: 0.3360 - val_loss: 1.7534 - val_accuracy: 0.3796
Epoch 13/50
391/391 [==============================] - 5s 12ms/step - loss: 1.8373 - accuracy: 0.3406 - val_loss: 1.6759 - val_accuracy: 0.4100
Epoch 14/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8323 - accuracy: 0.3470 - val_loss: 1.7545 - val_accuracy: 0.3734
Epoch 15/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8299 - accuracy: 0.3468 - val_loss: 1.7351 - val_accuracy: 0.3764
Epoch 16/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8179 - accuracy: 0.3494 - val_loss: 1.7491 - val_accuracy: 0.3705
Epoch 17/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8150 - accuracy: 0.3516 - val_loss: 1.7593 - val_accuracy: 0.3751
Epoch 18/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8104 - accuracy: 0.3514 - val_loss: 1.7409 - val_accuracy: 0.3909
Epoch 19/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8191 - accuracy: 0.3559 - val_loss: 1.6596 - val_accuracy: 0.4152
Epoch 20/50
391/391 [==============================] - 4s 11ms/step - loss: 1.8071 - accuracy: 0.3547 - val_loss: 1.6506 - val_accuracy: 0.4120
Epoch 21/50
391/391 [==============================] - 5s 12ms/step - loss: 1.7971 - accuracy: 0.3580 - val_loss: 1.6941 - val_accuracy: 0.4010
Epoch 22/50
391/391 [==============================] - 5s 12ms/step - loss: 1.7956 - accuracy: 0.3611 - val_loss: 1.6911 - val_accuracy: 0.3933
Epoch 23/50
391/391 [==============================] - 5s 12ms/step - loss: 1.7942 - accuracy: 0.3571 - val_loss: 1.6725 - val_accuracy: 0.4065
Epoch 24/50
391/391 [==============================] - 5s 12ms/step - loss: 1.8015 - accuracy: 0.3553 - val_loss: 1.6492 - val_accuracy: 0.4108
Epoch 25/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7848 - accuracy: 0.3652 - val_loss: 1.6812 - val_accuracy: 0.3957
Epoch 26/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7917 - accuracy: 0.3677 - val_loss: 1.6984 - val_accuracy: 0.4028
Epoch 27/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7805 - accuracy: 0.3675 - val_loss: 1.6383 - val_accuracy: 0.4241
Epoch 28/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7823 - accuracy: 0.3664 - val_loss: 1.6552 - val_accuracy: 0.4051
Epoch 29/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7826 - accuracy: 0.3671 - val_loss: 1.6610 - val_accuracy: 0.4061
Epoch 30/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7866 - accuracy: 0.3637 - val_loss: 1.6468 - val_accuracy: 0.4176
Epoch 31/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7851 - accuracy: 0.3638 - val_loss: 1.6154 - val_accuracy: 0.4265
Epoch 32/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7721 - accuracy: 0.3741 - val_loss: 1.6301 - val_accuracy: 0.4179
Epoch 33/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7782 - accuracy: 0.3686 - val_loss: 1.6146 - val_accuracy: 0.4263
Epoch 34/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7756 - accuracy: 0.3659 - val_loss: 1.6214 - val_accuracy: 0.4290
Epoch 35/50
391/391 [==============================] - 5s 12ms/step - loss: 1.7785 - accuracy: 0.3688 - val_loss: 1.5831 - val_accuracy: 0.4426
Epoch 36/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7779 - accuracy: 0.3713 - val_loss: 1.7387 - val_accuracy: 0.3723
Epoch 37/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7728 - accuracy: 0.3717 - val_loss: 1.6023 - val_accuracy: 0.4395
Epoch 38/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7671 - accuracy: 0.3772 - val_loss: 1.6533 - val_accuracy: 0.4165
Epoch 39/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7704 - accuracy: 0.3717 - val_loss: 1.6563 - val_accuracy: 0.4085
Epoch 40/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7636 - accuracy: 0.3735 - val_loss: 1.6873 - val_accuracy: 0.3987
Epoch 41/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7647 - accuracy: 0.3727 - val_loss: 1.7267 - val_accuracy: 0.3771
Epoch 42/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7619 - accuracy: 0.3746 - val_loss: 1.5968 - val_accuracy: 0.4407
Epoch 43/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7580 - accuracy: 0.3770 - val_loss: 1.6775 - val_accuracy: 0.4078
Epoch 44/50
391/391 [==============================] - 5s 12ms/step - loss: 1.7555 - accuracy: 0.3784 - val_loss: 1.6305 - val_accuracy: 0.4206
Epoch 45/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7561 - accuracy: 0.3757 - val_loss: 1.5964 - val_accuracy: 0.4467
Epoch 46/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7671 - accuracy: 0.3766 - val_loss: 1.6350 - val_accuracy: 0.4182
Epoch 47/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7608 - accuracy: 0.3772 - val_loss: 1.5990 - val_accuracy: 0.4375
Epoch 48/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7620 - accuracy: 0.3776 - val_loss: 1.6012 - val_accuracy: 0.4356
Epoch 49/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7647 - accuracy: 0.3727 - val_loss: 1.6235 - val_accuracy: 0.4352
Epoch 50/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7626 - accuracy: 0.3758 - val_loss: 1.6079 - val_accuracy: 0.4411
Test loss: 1.6078776121139526
Test accuracy: 0.44110000133514404
```

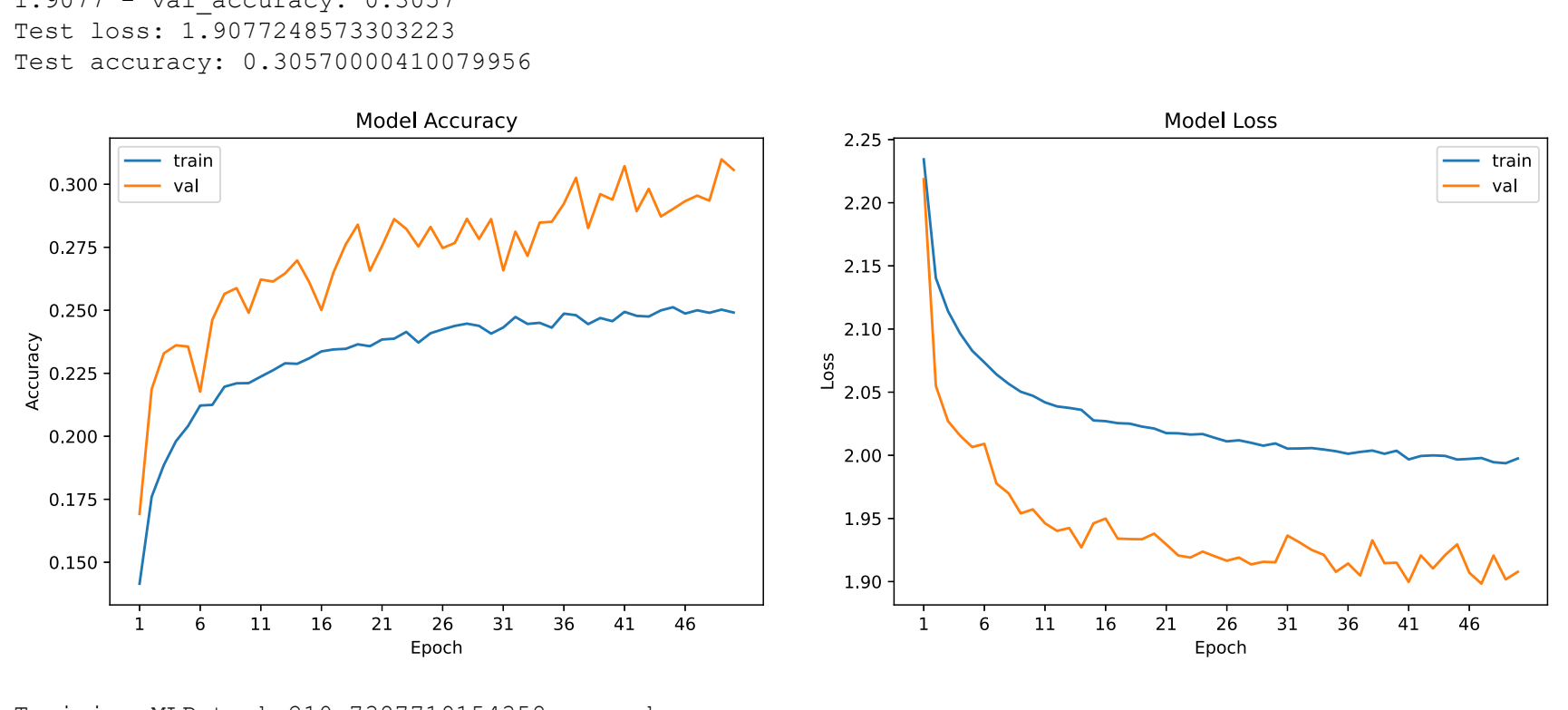



```
Training MLP took 226.78100061416626 seconds
```

Incluyendo capas de dropout, pero sin batch normalization

In [85]:
```python
model = build_model_v2(bn=False)

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Training
import time
start = time.time()
history = model.fit(X_train, Y_train, batch_size=128, epochs=50, verbose=1, validation_data=(X_test, Y_test))
end = time.time()

loss, acc = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', acc)

plot_model_history(history)
print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "sequential_23"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescaling_45 (Rescaling)     (None, 32, 32, 3)         0
_________________________________________________________________
random_rotation_45 (RandomRo (None, 32, 32, 3)         0
_________________________________________________________________
random_zoom_45 (RandomZoom)  (None, 32, 32, 3)         0
_________________________________________________________________
random_flip_45 (RandomFlip)  (None, 32, 32, 3)         0
_________________________________________________________________
flatten_56 (Flatten)         (None, 3072)              0
_________________________________________________________________
dense_147 (Dense)            (None, 128)               393344
_________________________________________________________________
dropout_115 (Dropout)        (None, 128)               0
_________________________________________________________________
activation_295 (Activation)  (None, 128)               0
_________________________________________________________________
dense_148 (Dense)            (None, 64)                8256
_________________________________________________________________
dropout_116 (Dropout)        (None, 64)                0
_________________________________________________________________
activation_296 (Activation)  (None, 64)                0
_________________________________________________________________
dense_149 (Dense)            (None, 32)                2080
_________________________________________________________________
dropout_117 (Dropout)        (None, 32)                0
_________________________________________________________________
activation_297 (Activation)  (None, 32)                0
_________________________________________________________________
dense_150 (Dense)            (None, 16)                528
_________________________________________________________________
batch_normalization_218 (Bat (None, 16)                64
_________________________________________________________________
dropout_118 (Dropout)        (None, 16)                0
_________________________________________________________________
activation_298 (Activation)  (None, 16)                0
_________________________________________________________________
dense_151 (Dense)            (None, 10)                170
_________________________________________________________________
activation_299 (Activation)  (None, 10)                0
=================================================================
Total params: 404,442
Trainable params: 404,410
Non-trainable params: 32
_________________________________________________________________
Epoch 1/50
391/391 [==============================] - 5s 11ms/step - loss: 2.2792 - accuracy: 0.1217 - val_loss: 2.2186 - val_accuracy: 0.1692
Epoch 2/50
391/391 [==============================] - 4s 11ms/step - loss: 2.1467 - accuracy: 0.1720 - val_loss: 2.0550 - val_accuracy: 0.2187
Epoch 3/50
391/391 [==============================] - 4s 11ms/step - loss: 2.1180 - accuracy: 0.1869 - val_loss: 2.0271 - val_accuracy: 0.2329
Epoch 4/50
391/391 [==============================] - 4s 10ms/step - loss: 2.1050 - accuracy: 0.1940 - val_loss: 2.0156 - val_accuracy: 0.2361
Epoch 5/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0797 - accuracy: 0.2034 - val_loss: 2.0065 - val_accuracy: 0.2356
Epoch 6/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0771 - accuracy: 0.2094 - val_loss: 2.0091 - val_accuracy: 0.2177
Epoch 7/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0673 - accuracy: 0.2124 - val_loss: 1.9777 - val_accuracy: 0.2462
Epoch 8/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0573 - accuracy: 0.2239 - val_loss: 1.9699 - val_accuracy: 0.2565
Epoch 9/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0450 - accuracy: 0.2216 - val_loss: 1.9540 - val_accuracy: 0.2588
Epoch 10/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0489 - accuracy: 0.2213 - val_loss: 1.9572 - val_accuracy: 0.2490
Epoch 11/50
391/391 [==============================] - 4s 10ms/step - loss: 2.0385 - accuracy: 0.2214 - val_loss: 1.9462 - val_accuracy: 0.2622
Epoch 12/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0391 - accuracy: 0.2277 - val_loss: 1.9401 - val_accuracy: 0.2614
Epoch 13/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0371 - accuracy: 0.2283 - val_loss: 1.9424 - val_accuracy: 0.2646
Epoch 14/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0372 - accuracy: 0.2283 - val_loss: 1.9271 - val_accuracy: 0.2698
Epoch 15/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0280 - accuracy: 0.2298 - val_loss: 1.9461 - val_accuracy: 0.2611
Epoch 16/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0235 - accuracy: 0.2334 - val_loss: 1.9499 - val_accuracy: 0.2501
Epoch 17/50
391/391 [==============================] - 4s 10ms/step - loss: 2.0300 - accuracy: 0.2336 - val_loss: 1.9341 - val_accuracy: 0.2651
Epoch 18/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0233 - accuracy: 0.2368 - val_loss: 1.9337 - val_accuracy: 0.2761
Epoch 19/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0225 - accuracy: 0.2351 - val_loss: 1.9335 - val_accuracy: 0.2840
Epoch 20/50
391/391 [==============================] - 4s 10ms/step - loss: 2.0253 - accuracy: 0.2357 - val_loss: 1.9380 - val_accuracy: 0.2657
Epoch 21/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0147 - accuracy: 0.2379 - val_loss: 1.9295 - val_accuracy: 0.2755
Epoch 22/50
391/391 [==============================] - 4s 10ms/step - loss: 2.0200 - accuracy: 0.2379 - val_loss: 1.9207 - val_accuracy: 0.2862
Epoch 23/50
391/391 [==============================] - 4s 10ms/step - loss: 2.0165 - accuracy: 0.2420 - val_loss: 1.9191 - val_accuracy: 0.2823
Epoch 24/50
391/391 [==============================] - 4s 10ms/step - loss: 2.0164 - accuracy: 0.2367 - val_loss: 1.9238 - val_accuracy: 0.2753
Epoch 25/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0138 - accuracy: 0.2421 - val_loss: 1.9201 - val_accuracy: 0.2831
Epoch 26/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0119 - accuracy: 0.2415 - val_loss: 1.9165 - val_accuracy: 0.2747
Epoch 27/50
391/391 [==============================] - 4s 10ms/step - loss: 2.0135 - accuracy: 0.2437 - val_loss: 1.9190 - val_accuracy: 0.2767
Epoch 28/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0112 - accuracy: 0.2453 - val_loss: 1.9136 - val_accuracy: 0.2863
Epoch 29/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0088 - accuracy: 0.2417 - val_loss: 1.9157 - val_accuracy: 0.2783
Epoch 30/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0028 - accuracy: 0.2430 - val_loss: 1.9153 - val_accuracy: 0.2862
Epoch 31/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0063 - accuracy: 0.2432 - val_loss: 1.9365 - val_accuracy: 0.2658
Epoch 32/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0019 - accuracy: 0.2472 - val_loss: 1.9310 - val_accuracy: 0.2812
Epoch 33/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0046 - accuracy: 0.2447 - val_loss: 1.9251 - val_accuracy: 0.2716
Epoch 34/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0066 - accuracy: 0.2419 - val_loss: 1.9212 - val_accuracy: 0.2848
Epoch 35/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0060 - accuracy: 0.2401 - val_loss: 1.9077 - val_accuracy: 0.2851
Epoch 36/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0035 - accuracy: 0.2463 - val_loss: 1.9143 - val_accuracy: 0.2923
Epoch 37/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0056 - accuracy: 0.2494 - val_loss: 1.9048 - val_accuracy: 0.3026
Epoch 38/50
391/391 [==============================] - 4s 11ms/step - loss: 1.9973 - accuracy: 0.2459 - val_loss: 1.9327 - val_accuracy: 0.2826
Epoch 39/50
391/391 [==============================] - 4s 10ms/step - loss: 1.9984 - accuracy: 0.2490 - val_loss: 1.9146 - val_accuracy: 0.2961
Epoch 40/50
391/391 [==============================] - 4s 10ms/step - loss: 2.0084 - accuracy: 0.2425 - val_loss: 1.9150 - val_accuracy: 0.2939
Epoch 41/50
391/391 [==============================] - 4s 11ms/step - loss: 1.9909 - accuracy: 0.2493 - val_loss: 1.8997 - val_accuracy: 0.3072
Epoch 42/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0044 - accuracy: 0.2452 - val_loss: 1.9207 - val_accuracy: 0.2893
Epoch 43/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0019 - accuracy: 0.2497 - val_loss: 1.9104 - val_accuracy: 0.2982
Epoch 44/50
391/391 [==============================] - 4s 11ms/step - loss: 1.9979 - accuracy: 0.2537 - val_loss: 1.9210 - val_accuracy: 0.2872
Epoch 45/50
391/391 [==============================] - 4s 11ms/step - loss: 1.9950 - accuracy: 0.2490 - val_loss: 1.9295 - val_accuracy: 0.2902
Epoch 46/50
391/391 [==============================] - 4s 11ms/step - loss: 1.9935 - accuracy: 0.2488 - val_loss: 1.9070 - val_accuracy: 0.2933
Epoch 47/50
391/391 [==============================] - 4s 11ms/step - loss: 2.0050 - accuracy: 0.2487 - val_loss: 1.8983 - val_accuracy: 0.2955
Epoch 48/50
391/391 [==============================] - 4s 11ms/step - loss: 1.9923 - accuracy: 0.2500 - val_loss: 1.9207 - val_accuracy: 0.2935
Epoch 49/50
391/391 [==============================] - 4s 11ms/step - loss: 1.9921 - accuracy: 0.2477 - val_loss: 1.9017 - val_accuracy: 0.3099
Epoch 50/50
391/391 [==============================] - 4s 11ms/step - loss: 1.9971 - accuracy: 0.2438 - val_loss: 1.9077 - val_accuracy: 0.3057
Test loss: 1.9077248573303223
Test accuracy: 0.30570000410079956
```



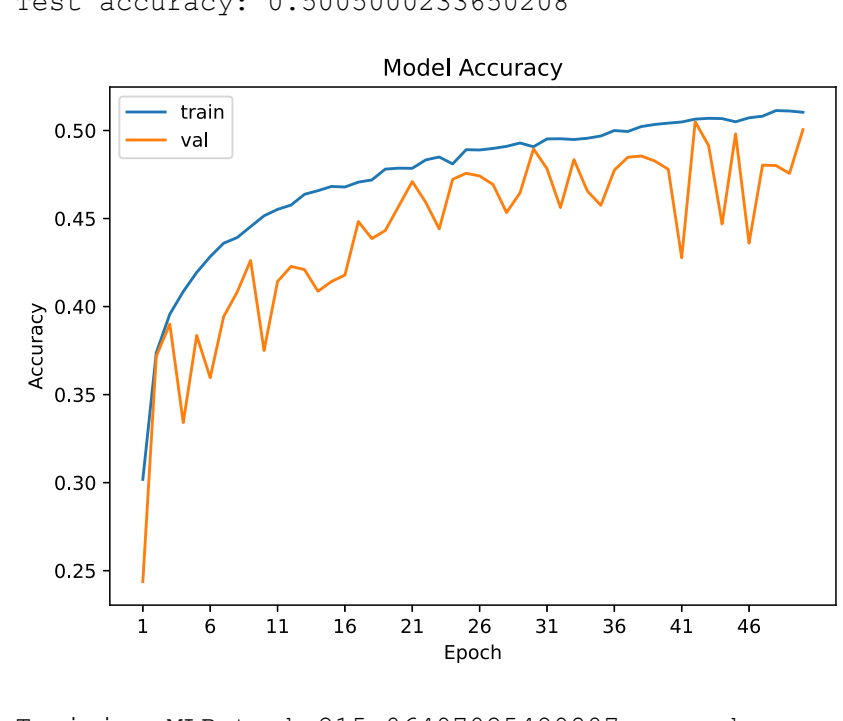

```
Training MLP took 210.7387719154358 seconds
```

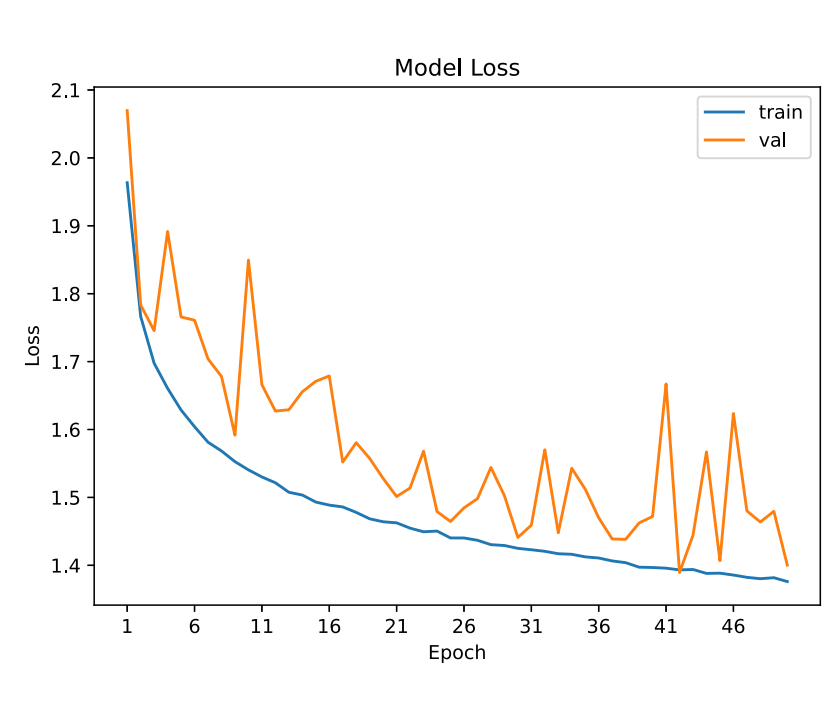
Incluyendo capas de batch normalization, pero sin dropout

In [86]:

```
model = build_model_v2(dout=False)

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Training
import time
start = time.time()
history = model.fit(X_train, Y_train, batch_size=128, epochs=50, verbose=1, validation_data=(X_test, Y_
test))
end = time.time()

loss, acc = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', acc)

plot_model_history(history)
print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "sequential_24"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescaling_46 (Rescaling)     (None, 32, 32, 3)         0
_________________________________________________________________
random_rotation_46 (RandomRo (None, 32, 32, 3)         0
_________________________________________________________________
random_zoom_46 (RandomZoom)  (None, 32, 32, 3)         0
_________________________________________________________________
random_flip_46 (RandomFlip)  (None, 32, 32, 3)         0
_________________________________________________________________
flatten_57 (Flatten)         (None, 3072)              0
_________________________________________________________________
dense_152 (Dense)            (None, 128)               393344
_________________________________________________________________
batch_normalization_219 (Bat (None, 128)               512
_________________________________________________________________
activation_300 (Activation)  (None, 128)               0
_________________________________________________________________
dense_153 (Dense)            (None, 64)                8256
_________________________________________________________________
batch_normalization_220 (Bat (None, 64)                256
_________________________________________________________________
activation_301 (Activation)  (None, 64)                0
_________________________________________________________________
dense_154 (Dense)            (None, 32)                2080
_________________________________________________________________
batch_normalization_221 (Bat (None, 32)                128
_________________________________________________________________
activation_302 (Activation)  (None, 32)                0
_________________________________________________________________
dense_155 (Dense)            (None, 16)                528
_________________________________________________________________
batch_normalization_222 (Bat (None, 16)                64
_________________________________________________________________
activation_303 (Activation)  (None, 16)                0
_________________________________________________________________
dense_156 (Dense)            (None, 10)                170
_________________________________________________________________
activation_304 (Activation)  (None, 10)                0
=================================================================
Total params: 405,338
Trainable params: 404,858
Non-trainable params: 480
_________________________________________________________________
Epoch 1/50
391/391 [==============================] - 6s 11ms/step - loss: 2.1057 - accuracy: 0.2487 - val_loss:
2.0697 - val_accuracy: 0.2438
Epoch 2/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7854 - accuracy: 0.3653 - val_loss:
1.7835 - val_accuracy: 0.3718
Epoch 3/50
391/391 [==============================] - 4s 11ms/step - loss: 1.7093 - accuracy: 0.3885 - val_loss:
1.7456 - val_accuracy: 0.3901
Epoch 4/50
391/391 [==============================] - 4s 11ms/step - loss: 1.6701 - accuracy: 0.4074 - val_loss:
1.8916 - val_accuracy: 0.3341
Epoch 5/50
391/391 [==============================] - 4s 11ms/step - loss: 1.6275 - accuracy: 0.4207 - val_loss:
1.7656 - val_accuracy: 0.3836
Epoch 6/50
391/391 [==============================] - 4s 11ms/step - loss: 1.6095 - accuracy: 0.4265 - val_loss:
1.7610 - val_accuracy: 0.3596
Epoch 7/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5829 - accuracy: 0.4350 - val_loss:
1.7460 - val_accuracy: 0.3943
Epoch 8/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5652 - accuracy: 0.4403 - val_loss:
1.6781 - val_accuracy: 0.4082
Epoch 9/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5524 - accuracy: 0.4435 - val_loss:
1.5916 - val_accuracy: 0.4262
Epoch 10/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5375 - accuracy: 0.4513 - val_loss:
1.8496 - val_accuracy: 0.3750
Epoch 11/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5362 - accuracy: 0.4537 - val_loss:
1.6660 - val_accuracy: 0.4142
Epoch 12/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5210 - accuracy: 0.4581 - val_loss:
1.6272 - val_accuracy: 0.4228
Epoch 13/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5050 - accuracy: 0.4631 - val_loss:
1.6290 - val_accuracy: 0.4210
Epoch 14/50
391/391 [==============================] - 4s 11ms/step - loss: 1.5006 - accuracy: 0.4666 - val_loss:
1.6555 - val_accuracy: 0.4087
Epoch 15/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4893 - accuracy: 0.4699 - val_loss:
1.6710 - val_accuracy: 0.4142
Epoch 16/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4883 - accuracy: 0.4689 - val_loss:
1.6789 - val_accuracy: 0.4179
Epoch 17/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4824 - accuracy: 0.4716 - val_loss:
1.5520 - val_accuracy: 0.4483
Epoch 18/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4783 - accuracy: 0.4736 - val_loss:
1.5806 - val_accuracy: 0.4386
Epoch 19/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4698 - accuracy: 0.4772 - val_loss:
1.5574 - val_accuracy: 0.4432
Epoch 20/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4621 - accuracy: 0.4790 - val_loss:
1.5280 - val_accuracy: 0.4571
Epoch 21/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4616 - accuracy: 0.4800 - val_loss:
1.5012 - val_accuracy: 0.4710
Epoch 22/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4585 - accuracy: 0.4817 - val_loss:
1.5140 - val_accuracy: 0.4592
Epoch 23/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4539 - accuracy: 0.4838 - val_loss:
1.5681 - val_accuracy: 0.4441
Epoch 24/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4391 - accuracy: 0.4851 - val_loss:
1.4791 - val_accuracy: 0.4723
Epoch 25/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4426 - accuracy: 0.4871 - val_loss:
1.4645 - val_accuracy: 0.4757
Epoch 26/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4403 - accuracy: 0.4881 - val_loss:
1.4846 - val_accuracy: 0.4742
Epoch 27/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4326 - accuracy: 0.4912 - val_loss:
1.4981 - val_accuracy: 0.4693
Epoch 28/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4231 - accuracy: 0.4939 - val_loss:
1.5441 - val_accuracy: 0.4534
Epoch 29/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4210 - accuracy: 0.4952 - val_loss:
1.5026 - val_accuracy: 0.4646
Epoch 30/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4276 - accuracy: 0.4905 - val_loss:
1.4411 - val_accuracy: 0.4896
Epoch 31/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4201 - accuracy: 0.4977 - val_loss:
1.4591 - val_accuracy: 0.4783
Epoch 32/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4116 - accuracy: 0.4971 - val_loss:
1.5702 - val_accuracy: 0.4562
Epoch 33/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4144 - accuracy: 0.4938 - val_loss:
1.4480 - val_accuracy: 0.4835
Epoch 34/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4166 - accuracy: 0.4955 - val_loss:
1.5429 - val_accuracy: 0.4656
Epoch 35/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4161 - accuracy: 0.4938 - val_loss:
1.5121 - val_accuracy: 0.4575
Epoch 36/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4121 - accuracy: 0.5000 - val_loss:
1.4701 - val_accuracy: 0.4775
Epoch 37/50
391/391 [==============================] - 4s 11ms/step - loss: 1.4042 - accuracy: 0.5003 - val_loss:
1.4388 - val_accuracy: 0.4907
Epoch 38/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3991 - accuracy: 0.5054 - val_loss:
1.4383 - val_accuracy: 0.4855
Epoch 39/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3971 - accuracy: 0.5033 - val_loss:
1.4624 - val_accuracy: 0.4827
Epoch 40/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3858 - accuracy: 0.5082 - val_loss:
1.4719 - val_accuracy: 0.4780
Epoch 41/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3854 - accuracy: 0.5080 - val_loss:
1.6671 - val_accuracy: 0.4277
Epoch 42/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3946 - accuracy: 0.5038 - val_loss:
1.3894 - val_accuracy: 0.5050
Epoch 43/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3858 - accuracy: 0.5107 - val_loss:
1.4442 - val_accuracy: 0.4914
Epoch 44/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3880 - accuracy: 0.5085 - val_loss:
1.5670 - val_accuracy: 0.4469
Epoch 45/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3791 - accuracy: 0.5062 - val_loss:
1.4068 - val_accuracy: 0.4981
Epoch 46/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3852 - accuracy: 0.5078 - val_loss:
1.6236 - val_accuracy: 0.4360
Epoch 47/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3843 - accuracy: 0.5086 - val_loss:
1.4801 - val_accuracy: 0.4803
Epoch 48/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3745 - accuracy: 0.5135 - val_loss:
1.4638 - val_accuracy: 0.4801
Epoch 49/50
391/391 [==============================] - 4s 11ms/step - loss: 1.3754 - accuracy: 0.5146 - val_loss:
1.4796 - val_accuracy: 0.4756
Epoch 50/50
391/391 [==============================] - 5s 12ms/step - loss: 1.3763 - accuracy: 0.5088 - val_loss:
1.4003 - val_accuracy: 0.5005
Test loss: 1.4002561569213867
Test accuracy: 0.5005000233650208
```






```
Training MLP took 215.06497025489807 seconds
```

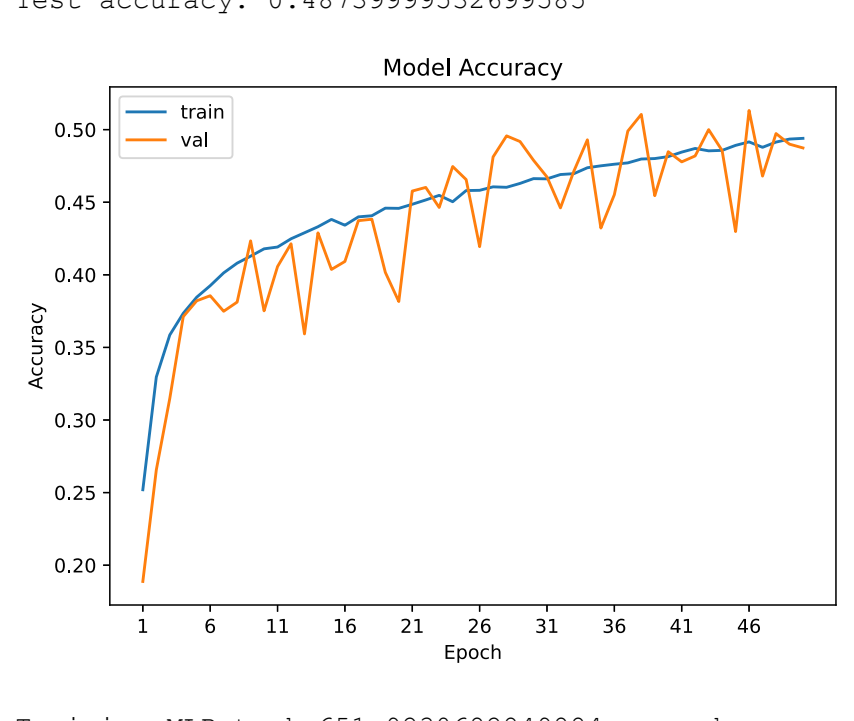
Modelo ffNN 3

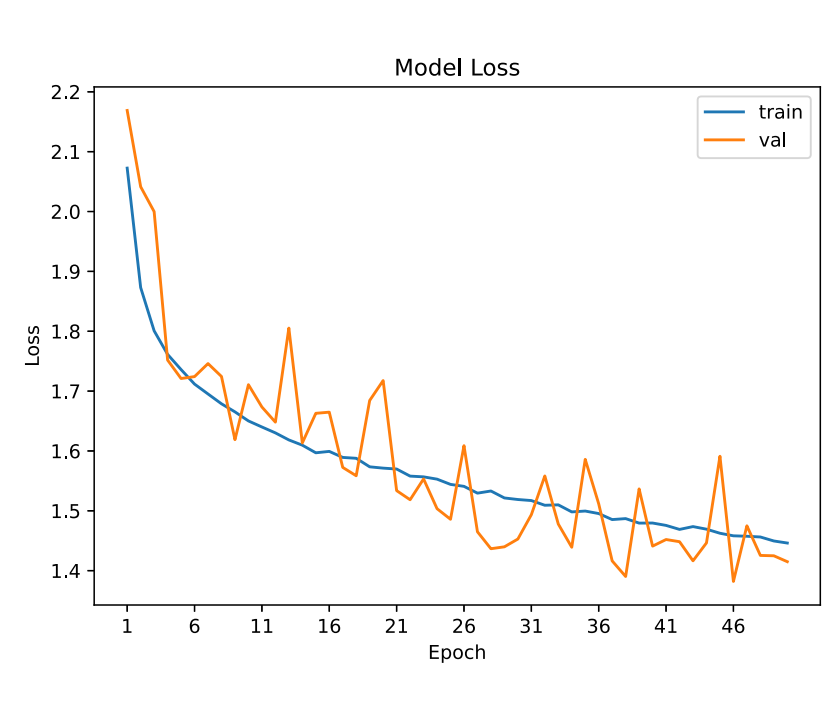
Incluyendo capas de dropout y batch normalization

In [87]:

```
model = build_model_v3()

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Training
import time
start = time.time()
history = model.fit(X_train, Y_train, batch_size=128, epochs=50, verbose=1, validation_data=(X_test, Y_
test))
end = time.time()

loss, acc = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', acc)

plot_model_history(history)
print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "sequential_25"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescaling_47 (Rescaling)     (None, 32, 32, 3)         0
_________________________________________________________________
random_rotation_47 (RandomRo (None, 32, 32, 3)         0
_________________________________________________________________
random_zoom_47 (RandomZoom)  (None, 32, 32, 3)         0
_________________________________________________________________
random_flip_47 (RandomFlip)  (None, 32, 32, 3)         0
_________________________________________________________________
flatten_58 (Flatten)         (None, 3072)              0
_________________________________________________________________
dense_157 (Dense)            (None, 1024)              3146752
_________________________________________________________________
batch_normalization_223 (Bat (None, 1024)              4096
_________________________________________________________________
dropout_119 (Dropout)        (None, 1024)              0
_________________________________________________________________
activation_305 (Activation)  (None, 1024)              0
_________________________________________________________________
dense_158 (Dense)            (None, 512)               524800
_________________________________________________________________
batch_normalization_224 (Bat (None, 512)               2048
_________________________________________________________________
dropout_120 (Dropout)        (None, 512)               0
_________________________________________________________________
activation_306 (Activation)  (None, 512)               0
_________________________________________________________________
dense_159 (Dense)            (None, 64)                32832
_________________________________________________________________
batch_normalization_225 (Bat (None, 64)                256
_________________________________________________________________
dropout_121 (Dropout)        (None, 64)                0
_________________________________________________________________
activation_307 (Activation)  (None, 64)                0
_________________________________________________________________
dense_160 (Dense)            (None, 64)                4160
_________________________________________________________________
batch_normalization_226 (Bat (None, 64)                256
_________________________________________________________________
dropout_122 (Dropout)        (None, 64)                0
_________________________________________________________________
activation_308 (Activation)  (None, 64)                0
_________________________________________________________________
dense_161 (Dense)            (None, 10)                650
_________________________________________________________________
activation_309 (Activation)  (None, 10)                0
=================================================================
Total params: 3,715,850
Trainable params: 3,712,522
Non-trainable params: 3,328
_________________________________________________________________
Epoch 1/50
391/391 [==============================] - 13s 28ms/step - loss: 2.2159 - accuracy: 0.2143 - val_los
s: 2.1688 - val_accuracy: 0.1888
Epoch 2/50
391/391 [==============================] - 11s 29ms/step - loss: 1.8947 - accuracy: 0.3218 - val_los
s: 2.0416 - val_accuracy: 0.2656
Epoch 3/50
391/391 [==============================] - 14s 36ms/step - loss: 1.8134 - accuracy: 0.3512 - val_los
s: 1.9998 - val_accuracy: 0.3148
Epoch 4/50
391/391 [==============================] - 13s 34ms/step - loss: 1.7600 - accuracy: 0.3718 - val_los
s: 1.9752 - val_accuracy: 0.3262
Epoch 5/50
391/391 [==============================] - 13s 34ms/step - loss: 1.7371 - accuracy: 0.3819 - val_los
s: 1.7210 - val_accuracy: 0.3821
Epoch 6/50
391/391 [==============================] - 14s 36ms/step - loss: 1.7185 - accuracy: 0.3896 - val_los
s: 1.7558 - val_accuracy: 0.3856
Epoch 7/50
391/391 [==============================] - 13s 34ms/step - loss: 1.7042 - accuracy: 0.3995 - val_los
s: 1.7460 - val_accuracy: 0.3749
Epoch 8/50
391/391 [==============================] - 13s 34ms/step - loss: 1.6759 - accuracy: 0.4114 - val_los
s: 1.7244 - val_accuracy: 0.3812
Epoch 9/50
391/391 [==============================] - 13s 34ms/step - loss: 1.6687 - accuracy: 0.4137 - val_los
s: 1.6188 - val_accuracy: 0.4236
Epoch 10/50
391/391 [==============================] - 13s 34ms/step - loss: 1.6568 - accuracy: 0.4155 - val_los
s: 1.7167 - val_accuracy: 0.3752
Epoch 11/50
391/391 [==============================] - 13s 34ms/step - loss: 1.6396 - accuracy: 0.4184 - val_los
s: 1.6734 - val_accuracy: 0.4057
Epoch 12/50
391/391 [==============================] - 13s 33ms/step - loss: 1.6309 - accuracy: 0.4266 - val_los
s: 1.6481 - val_accuracy: 0.4214
Epoch 13/50
391/391 [==============================] - 13s 34ms/step - loss: 1.6185 - accuracy: 0.4275 - val_los
s: 1.8053 - val_accuracy: 0.3593
Epoch 14/50
391/391 [==============================] - 13s 34ms/step - loss: 1.6098 - accuracy: 0.4335 - val_los
s: 1.6134 - val_accuracy: 0.4289
Epoch 15/50
391/391 [==============================] - 13s 33ms/step - loss: 1.5914 - accuracy: 0.4404 - val_los
s: 1.6629 - val_accuracy: 0.4037
Epoch 16/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5981 - accuracy: 0.4347 - val_los
s: 1.6648 - val_accuracy: 0.4092
Epoch 17/50
391/391 [==============================] - 13s 33ms/step - loss: 1.5894 - accuracy: 0.4405 - val_los
s: 1.5725 - val_accuracy: 0.4373
Epoch 18/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5837 - accuracy: 0.4420 - val_los
s: 1.5582 - val_accuracy: 0.4383
Epoch 19/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5761 - accuracy: 0.4449 - val_los
s: 1.6843 - val_accuracy: 0.4017
Epoch 20/50
391/391 [==============================] - 13s 33ms/step - loss: 1.5706 - accuracy: 0.4467 - val_los
s: 1.7176 - val_accuracy: 0.3816
Epoch 21/50
391/391 [==============================] - 13s 33ms/step - loss: 1.5658 - accuracy: 0.4520 - val_los
s: 1.5337 - val_accuracy: 0.4577
Epoch 22/50
391/391 [==============================] - 14s 37ms/step - loss: 1.5567 - accuracy: 0.4529 - val_los
s: 1.5165 - val_accuracy: 0.4602
Epoch 23/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5502 - accuracy: 0.4595 - val_los
s: 1.5532 - val_accuracy: 0.4464
Epoch 24/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5589 - accuracy: 0.4458 - val_los
s: 1.5084 - val_accuracy: 0.4746
Epoch 25/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5412 - accuracy: 0.4559 - val_los
s: 1.4858 - val_accuracy: 0.4656
Epoch 26/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5360 - accuracy: 0.4605 - val_los
s: 1.6088 - val_accuracy: 0.4194
Epoch 27/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5262 - accuracy: 0.4609 - val_los
s: 1.4650 - val_accuracy: 0.4812
Epoch 28/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5376 - accuracy: 0.4571 - val_los
s: 1.4367 - val_accuracy: 0.4907
Epoch 29/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5153 - accuracy: 0.4649 - val_los
s: 1.4398 - val_accuracy: 0.4918
Epoch 30/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5129 - accuracy: 0.4655 - val_los
s: 1.4528 - val_accuracy: 0.4789
Epoch 31/50
391/391 [==============================] - 14s 35ms/step - loss: 1.5233 - accuracy: 0.4621 - val_los
s: 1.4935 - val_accuracy: 0.4674
Epoch 32/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5084 - accuracy: 0.4676 - val_los
s: 1.5582 - val_accuracy: 0.4461
Epoch 33/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5024 - accuracy: 0.4756 - val_los
s: 1.4779 - val_accuracy: 0.4715
Epoch 34/50
391/391 [==============================] - 13s 34ms/step - loss: 1.4991 - accuracy: 0.4708 - val_los
s: 1.4391 - val_accuracy: 0.4930
Epoch 35/50
391/391 [==============================] - 13s 34ms/step - loss: 1.5013 - accuracy: 0.4741 - val_los
s: 1.5860 - val_accuracy: 0.4322
Epoch 36/50
391/391 [==============================] - 14s 35ms/step - loss: 1.4999 - accuracy: 0.4743 - val_los
s: 1.5117 - val_accuracy: 0.4552
Epoch 37/50
391/391 [==============================] - 14s 35ms/step - loss: 1.4823 - accuracy: 0.4768 - val_los
s: 1.4163 - val_accuracy: 0.4990
Epoch 38/50
391/391 [==============================] - 12s 32ms/step - loss: 1.4855 - accuracy: 0.4809 - val_los
s: 1.3903 - val_accuracy: 0.5105
Epoch 39/50
391/391 [==============================] - 12s 30ms/step - loss: 1.4685 - accuracy: 0.4831 - val_los
s: 1.5366 - val_accuracy: 0.4545
Epoch 40/50
391/391 [==============================] - 12s 31ms/step - loss: 1.4682 - accuracy: 0.4853 - val_los
s: 1.4410 - val_accuracy: 0.4848
Epoch 41/50
391/391 [==============================] - 12s 32ms/step - loss: 1.4761 - accuracy: 0.4835 - val_los
s: 1.4520 - val_accuracy: 0.4778
Epoch 42/50
391/391 [==============================] - 12s 30ms/step - loss: 1.4694 - accuracy: 0.4869 - val_los
s: 1.4484 - val_accuracy: 0.4819
Epoch 43/50
391/391 [==============================] - 12s 30ms/step - loss: 1.4724 - accuracy: 0.4860 - val_los
s: 1.4164 - val_accuracy: 0.5000
Epoch 44/50
391/391 [==============================] - 11s 29ms/step - loss: 1.4697 - accuracy: 0.4823 - val_los
s: 1.4464 - val_accuracy: 0.4856
Epoch 45/50
391/391 [==============================] - 12s 30ms/step - loss: 1.4728 - accuracy: 0.4873 - val_los
s: 1.5912 - val_accuracy: 0.4298
Epoch 46/50
391/391 [==============================] - 12s 31ms/step - loss: 1.4607 - accuracy: 0.4915 - val_los
s: 1.3820 - val_accuracy: 0.5132
Epoch 47/50
391/391 [==============================] - 12s 31ms/step - loss: 1.4477 - accuracy: 0.4897 - val_los
s: 1.4750 - val_accuracy: 0.4680
Epoch 48/50
391/391 [==============================] - 14s 35ms/step - loss: 1.4509 - accuracy: 0.4942 - val_los
s: 1.4254 - val_accuracy: 0.4973
Epoch 49/50
391/391 [==============================] - 14s 36ms/step - loss: 1.4437 - accuracy: 0.4976 - val_los
s: 1.4249 - val_accuracy: 0.4900
Epoch 50/50
391/391 [==============================] - 14s 36ms/step - loss: 1.4442 - accuracy: 0.4944 - val_los
s: 1.4150 - val_accuracy: 0.4874
Test loss: 1.4149885177612305
Test accuracy: 0.48739999532699585
```






```
Training MLP took 651.9230682849884 seconds
```

Incluyendo capas de dropout, pero sin batch normalization

In [88]:

```
model = build_model_v3(bn=False)

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Training
import time
start = time.time()
history = model.fit(X_train, Y_train, batch_size=128, epochs=50, verbose=1, validation_data=(X_test, Y_
test))
end = time.time()

loss, acc = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', acc)

plot_model_history(history)
print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "sequential_26"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
rescaling_48 (Rescaling)     (None, 32, 32, 3)         0
_________________________________________________________________
random_rotation_48 (RandomRo (None, 32, 32, 3)         0
_________________________________________________________________
random_zoom_48 (RandomZoom)  (None, 32, 32, 3)         0
_________________________________________________________________
random_flip_48 (RandomFlip)  (None, 32, 32, 3)         0
_________________________________________________________________
flatten_59 (Flatten)         (None, 3072)              0
_________________________________________________________________
dense_162 (Dense)            (None, 1024)              3146752
_________________________________________________________________
dropout_123 (Dropout)        (None, 1024)              0
_________________________________________________________________
activation_310 (Activation)  (None, 1024)              0
_________________________________________________________________
dense_163 (Dense)            (None, 512)               524800
_________________________________________________________________
dropout_124 (Dropout)        (None, 512)               0
_________________________________________________________________
activation_311 (Activation)  (None, 512)               0
_________________________________________________________________
dense_164 (Dense)            (None, 64)                32832
_________________________________________________________________
batch_normalization_227 (Bat (None, 64)                256
_________________________________________________________________
dropout_125 (Dropout)        (None, 64)                0
_________________________________________________________________
activation_312 (Activation)  (None, 64)                0
_________________________________________________________________
dense_165 (Dense)            (None, 64)                4160
_________________________________________________________________
dropout_126 (Dropout)        (None, 64)                0
_________________________________________________________________
activation_313 (Activation)  (None, 64)                0
_________________________________________________________________
dense_166 (Dense)            (None, 10)                650
_________________________________________________________________
activation_314 (Activation)  (None, 10)                0
=================================================================
Total params: 3,709,450
Trainable params: 3,709,322
Non-trainable params: 128
_________________________________________________________________
Epoch 1/50
391/391 [==============================] - 14s 33ms/step - loss: 2.1934 - accuracy: 0.1755 - val_los
s: 2.1491 - val_accuracy: 0.2533
Epoch 2/50
391/391 [==============================] - 12s 32ms/step - loss: 2.0218 - accuracy: 0.2613 - val_los
s: 1.8840 - val_accuracy: 0.3304
Epoch 3/50
391/391 [==============================] - 13s 32ms/step - loss: 1.9841 - accuracy: 0.2770 - val_los
s: 1.8752 - val_accuracy: 0.3262
Epoch 4/50
391/391 [==============================] - 12s 32ms/step - loss: 1.9665 - accuracy: 0.2867 - val_los
s: 1.8180 - val_accuracy: 0.3522
Epoch 5/50
391/391 [==============================] - 13s 32ms/step - loss: 1.9473 - accuracy: 0.2976 - val_los
s: 1.8344 - val_accuracy: 0.3435
Epoch 6/50
391/391 [==============================] - 12s 32ms/step - loss: 1.9272 - accuracy: 0.3049 - val_los
s: 1.7966 - val_accuracy: 0.3523
Epoch 7/50
391/391 [==============================] - 12s 32ms/step - loss: 1.9169 - accuracy: 0.3085 - val_los
s: 1.7526 - val_accuracy: 0.3707
Epoch 8/50
391/391 [==============================] - 12s 32ms/step - loss: 1.9092 - accuracy: 0.3115 - val_los
s: 1.7613 - val_accuracy: 0.3706
Epoch 9/50
391/391 [==============================] - 12s 30ms/step - loss: 1.8914 - accuracy: 0.3197 - val_los
s: 1.7503 - val_accuracy: 0.3818
Epoch 10/50
391/391 [==============================] - 13s 32ms/step - loss: 1.8926 - accuracy: 0.3216 - val_los
s: 1.7174 - val_accuracy: 0.3884
Epoch 11/50
391/391 [==============================] - 11s 28ms/step - loss: 1.8728 - accuracy: 0.3306 - val_los
s: 1.7334 - val_accuracy: 0.3851
Epoch 12/50
391/391 [==============================] - 12s 31ms/step - loss: 1.8704 - accuracy: 0.3320 - val_los
s: 1.7233 - val_accuracy: 0.3844
Epoch 13/50
391/391 [==============================] - 11s 29ms/step - loss: 1.8613 - accuracy: 0.3292 - val_los
s: 1.7106 - val_accuracy: 0.3886
Epoch 14/50
391/391 [==============================] - 11s 28ms/step - loss: 1.8588 - accuracy: 0.3355 - val_los
s: 1.7069 - val_accuracy: 0.3968
Epoch 15/50
391/391 [==============================] - 11s 28ms/step - loss: 1.8495 - accuracy: 0.3385 - val_los
s: 1.6933 - val_accuracy: 0.3922
Epoch 16/50
391/391 [==============================] - 11s 28ms/step - loss: 1.8586 - accuracy: 0.3351 - val_los
s: 1.6878 - val_accuracy: 0.4016
Epoch 17/50
391/391 [==============================] - 11s 27ms/step - loss: 1.8446 - accuracy: 0.3352 - val_los
s: 1.6711 - val_accuracy: 0.4062
Epoch 18/50
391/391 [==============================] - 11s 28ms/step - loss: 1.8440 - accuracy: 0.3425 - val_los
s: 1.6658 - val_accuracy: 0.4064
Epoch 19/50
391/391 [==============================] - 10s 27ms/step - loss: 1.8366 - accuracy: 0.3453 - val_los
s: 1.6642 - val_accuracy: 0.4105
Epoch 20/50
391/391 [==============================] - 11s 27ms/step - loss: 1.8229 - accuracy: 0.3546 - val_los
s: 1.6548 - val_accuracy: 0.4162
Epoch 21/50
391/391 [==============================] - 11s 28ms/step - loss: 1.8271 - accuracy: 0.3473 - val_los
s: 1.6435 - val_accuracy: 0.4164
Epoch 22/50
391/391 [==============================] - 11s 28ms/step - loss: 1.8217 - accuracy: 0.3509 - val_los
s: 1.6491 - val_accuracy: 0.4159
Epoch 23/50
391/391 [==============================] - 11s 28ms/step - loss: 1.8178 - accuracy: 0.3479 - val_los
s: 1.6525 - val_accuracy: 0.4124
Epoch 24/50
391/391 [==============================] - 11s 28ms/step - loss: 1.8246 - accuracy: 0.3488 - val_los
s: 1.6621 - val_accuracy: 0.4066
Epoch 25/50
391/391 [==============================] - 10s 27ms/step - loss: 1.8155 - accuracy: 0.3522 - val_los
s: 1.6449 - val_accuracy: 0.4133
Epoch 26/50
391/391 [==============================] - 10s 26ms/step - loss: 1.8150 - accuracy: 0.3523 - val_los
s: 1.6413 - val_accuracy: 0.4155
Epoch 27/50
391/391 [==============================] - 10s 27ms/step - loss: 1.8093 - accuracy: 0.3565 - val_los
s: 1.6360 - val_accuracy: 0.4212
Epoch 28/50
391/391 [==============================] - 10s 27ms/step - loss: 1.8122 - accuracy: 0.3547 - val_los
s: 1.6423 - val_accuracy: 0.4187
Epoch 29/50
391/391 [==============================] - 13s 33ms/step - loss: 1.7966 - accuracy: 0.3590 - val_los
s: 1.6325 - val_accuracy: 0.4244
Epoch 30/50
391/391 [==============================] - 12s 32ms/step - loss: 1.8058 - accuracy: 0.3579 - val_los
s: 1.6285 - val_accuracy: 0.4176
Epoch 31/50
391/391 [==============================] - 15s 38ms/step - loss: 1.7968 - accuracy: 0.3578 - val_los
s: 1.6260 - val_accuracy: 0.4206
Epoch 32/50
391/391 [==============================] - 12s 31ms/step - loss: 1.7974 - accuracy: 0.3577 - val_los
s: 1.6249 - val_accuracy: 0.4212
Epoch 33/50
391/391 [==============================] - 12s 31ms/step - loss: 1.7909 - accuracy: 0.3609 - val_los
s: 1.6358 - val_accuracy: 0.4184
Epoch 34/50
391/391 [==============================] - 11s 29ms/step - loss: 1.7955 - accuracy: 0.3600 - val_los
s: 1.6252 - val_accuracy: 0.4209
Epoch 35/50
391/391 [==============================] - 12s 32ms/step - loss: 1.7810 - accuracy: 0.3657 - val_los
s: 1.6410 - val_accuracy: 0.4086
Epoch 36/50
391/391 [==============================] - 13s 33ms/step - loss: 1.7822 - accuracy: 0.3624 - val_los
s: 1.6085 - val_accuracy: 0.4284
Epoch 37/50
391/391 [==============================] - 12s 30ms/step - loss: 1.7846 - accuracy: 0.3621 - val_los
s: 1.6070 - val_accuracy: 0.4286
Epoch 38/50
391/391 [==============================] - 12s 30ms/step - loss: 1.7858 - accuracy: 0.3650 - val_los
s: 1.6230 - val_accuracy: 0.4186
Epoch 39/50
391/391 [==============================] - 12s 31ms/step - loss: 1.7854 - accuracy: 0.3618 - val_los
s: 1.6107 - val_accuracy: 0.4298
Epoch 40/50
391/391 [==============================] - 11s 29ms/step - loss: 1.7834 - accuracy: 0.3672 - val_los
s: 1.6032 - val_accuracy: 0.4268
Epoch 41/50
391/391 [==============================] - 12s 30ms/step - loss: 1.7755 - accuracy: 0.3657 - val_los
s: 1.6139 - val_accuracy: 0.4290
Epoch 42/50
391/391 [==============================] - 12s 30ms/step - loss: 1.7773 - accuracy: 0.3725 - val_los
s: 1.6071 - val_accuracy: 0.4315
Epoch 43/50
391/391 [==============================] - 12s 30ms/step - loss: 1.7758 - accuracy: 0.3685 - val_los
s: 1.5996 - val_accuracy: 0.4345
Epoch 44/50
391/391 [==============================] - 12s 30ms/step - loss: 1.7711 - accuracy: 0.3671 - val_los
s: 1.5885 - val_accuracy: 0.4440
Epoch 45/50
391/391 [==============================] - 12s 29ms/step - loss: 1.7707 - accuracy: 0.3691 - val_los
s: 1.5957 - val_accuracy: 0.4311
Epoch 46/50
391/391 [==============================] - 12s 31ms/step - loss: 1.7642 - accuracy: 0.3706 - val_los
s: 1.5879 - val_accuracy: 0.4367
Epoch 47/50
391/391 [==============================] - 11s 29ms/step - loss: 1.7694 - accuracy: 0.3689 - val_los
s: 1.5922 - val_accuracy: 0.4320
Epoch 48/50
391/391 [==============================] - 12s 30ms/step - loss: 1.7691 - accuracy: 0.3697 - val_los
s: 1.6086 - val_accuracy: 0.4343
Epoch 49/50
391/391 [==============================] - 11s 29ms/step - loss: 1.7775 - accuracy: 0.3662 - val_los
s: 1.5785 - val_accuracy: 0.4358
Epoch 50/50
391/391 [==============================] - 12s 30ms/step - loss: 1.7657 - accuracy: 0.3711 - val_los
s: 1.5786 - val_accuracy: 0.4409
Test loss: 1.5786265134811401
Test accuracy: 0.4408999979496002
```

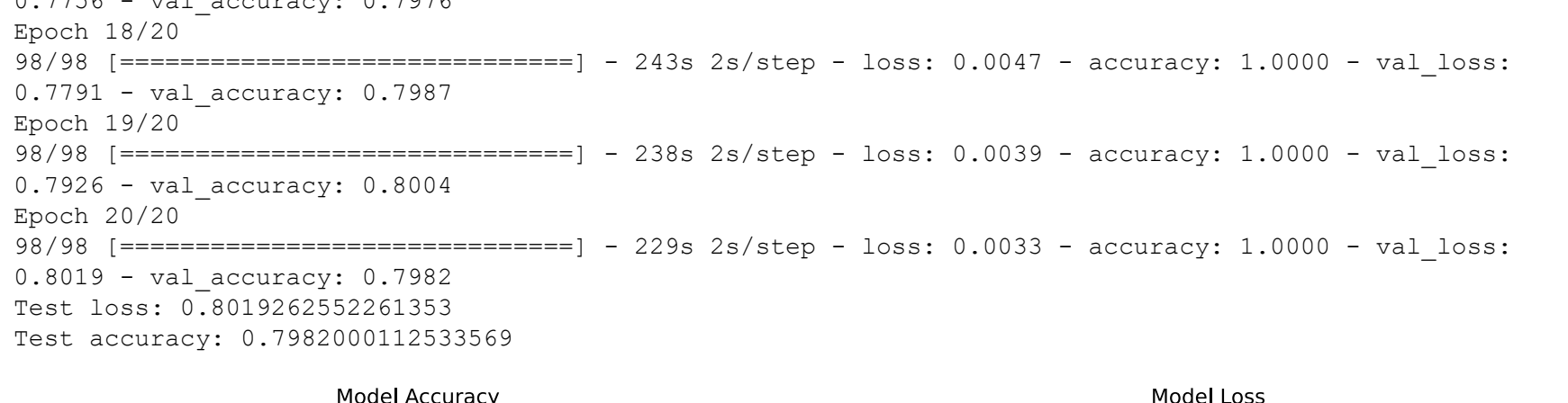


```
Training MLP took 586.0059037208557 seconds
```

Incluyendo capas de batch normalization, pero sin dropout

In [74]:
```python
model = build_conv1(dout=False, bn=True, bn_before=True, extra_layer=False)

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Training
start = time.time()
history = model.fit(X_train, Y_train, batch_size=512, epochs=20, verbose=1, validation_data=(X_test, Y_
test))
end = time.time()

loss, acc = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', acc)

plot_model_history(history)
print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "model_19"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_27 (InputLayer)        [(None, 32, 32, 3)]       0
_________________________________________________________________
conv2d_156 (Conv2D)          (None, 32, 32, 32)        896
_________________________________________________________________
batch_normalization_174 (Bat (None, 32, 32, 32)        128
_________________________________________________________________
activation_239 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_157 (Conv2D)          (None, 32, 32, 32)        9248
_________________________________________________________________
batch_normalization_175 (Bat (None, 32, 32, 32)        128
_________________________________________________________________
activation_240 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
max_pooling2d_78 (MaxPooling (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_158 (Conv2D)          (None, 16, 16, 64)        18496
_________________________________________________________________
batch_normalization_176 (Bat (None, 16, 16, 64)        256
_________________________________________________________________
activation_241 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_159 (Conv2D)          (None, 16, 16, 64)        36928
_________________________________________________________________
batch_normalization_177 (Bat (None, 16, 16, 64)        256
_________________________________________________________________
activation_242 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
max_pooling2d_79 (MaxPooling (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_160 (Conv2D)          (None, 8, 8, 128)         73856
_________________________________________________________________
batch_normalization_178 (Bat (None, 8, 8, 128)         512
_________________________________________________________________
activation_243 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
conv2d_161 (Conv2D)          (None, 8, 8, 128)         147584
_________________________________________________________________
batch_normalization_179 (Bat (None, 8, 8, 128)         512
_________________________________________________________________
activation_244 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
max_pooling2d_80 (MaxPooling (None, 4, 4, 128)         0
_________________________________________________________________
flatten_45 (Flatten)         (None, 2048)              0
_________________________________________________________________
dense_121 (Dense)            (None, 10)                20490
=================================================================
Total params: 309,290
Trainable params: 308,394
Non-trainable params: 896
_________________________________________________________________
Epoch 1/20
98/98 [==============================] - 221s 2s/step - loss: 1.8586 - accuracy: 0.3896 - val_loss:
2.9792 - val_accuracy: 0.1191
Epoch 2/20
98/98 [==============================] - 208s 2s/step - loss: 0.9776 - accuracy: 0.6560 - val_loss:
3.2510 - val_accuracy: 0.1434
Epoch 3/20
98/98 [==============================] - 245s 3s/step - loss: 0.7136 - accuracy: 0.7515 - val_loss:
2.4966 - val_accuracy: 0.2265
Epoch 4/20
98/98 [==============================] - 198s 2s/step - loss: 0.5858 - accuracy: 0.7988 - val_loss:
1.6289 - val_accuracy: 0.4418
Epoch 5/20
98/98 [==============================] - 252s 3s/step - loss: 0.4734 - accuracy: 0.8427 - val_loss:
1.3341 - val_accuracy: 0.5786
Epoch 6/20
98/98 [==============================] - 248s 3s/step - loss: 0.3810 - accuracy: 0.8733 - val_loss:
0.8497 - val_accuracy: 0.7135
Epoch 7/20
98/98 [==============================] - 258s 3s/step - loss: 0.3097 - accuracy: 0.9031 - val_loss:
0.8623 - val_accuracy: 0.7167
Epoch 8/20
98/98 [==============================] - 212s 2s/step - loss: 0.2379 - accuracy: 0.9280 - val_loss:
0.7537 - val_accuracy: 0.7533
Epoch 9/20
98/98 [==============================] - 210s 2s/step - loss: 0.1753 - accuracy: 0.9532 - val_loss:
0.8355 - val_accuracy: 0.7406
Epoch 10/20
98/98 [==============================] - 194s 2s/step - loss: 0.1350 - accuracy: 0.9666 - val_loss:
0.9216 - val_accuracy: 0.7326
Epoch 11/20
98/98 [==============================] - 195s 2s/step - loss: 0.1034 - accuracy: 0.9783 - val_loss:
0.9641 - val_accuracy: 0.7307
Epoch 12/20
98/98 [==============================] - 245s 3s/step - loss: 0.0654 - accuracy: 0.9903 - val_loss:
0.8028 - val_accuracy: 0.7687
Epoch 13/20
98/98 [==============================] - 248s 3s/step - loss: 0.0443 - accuracy: 0.9951 - val_loss:
0.8300 - val_accuracy: 0.7681
Epoch 14/20
98/98 [==============================] - 243s 2s/step - loss: 0.0269 - accuracy: 0.9985 - val_loss:
0.8990 - val_accuracy: 0.7570
Epoch 15/20
98/98 [==============================] - 241s 2s/step - loss: 0.0147 - accuracy: 1.0000 - val_loss:
0.7631 - val_accuracy: 0.7939
Epoch 16/20
98/98 [==============================] - 247s 3s/step - loss: 0.0084 - accuracy: 1.0000 - val_loss:
0.7796 - val_accuracy: 0.7919
Epoch 17/20
98/98 [==============================] - 263s 3s/step - loss: 0.0057 - accuracy: 1.0000 - val_loss:
0.7756 - val_accuracy: 0.7976
Epoch 18/20
98/98 [==============================] - 248s 3s/step - loss: 0.0047 - accuracy: 1.0000 - val_loss:
0.7791 - val_accuracy: 0.7987
Epoch 19/20
98/98 [==============================] - 238s 2s/step - loss: 0.0039 - accuracy: 1.0000 - val_loss:
0.7926 - val_accuracy: 0.8004
Epoch 20/20
98/98 [==============================] - 229s 2s/step - loss: 0.0033 - accuracy: 1.0000 - val_loss:
0.8019 - val_accuracy: 0.7982
Test loss: 0.8019262552261353
Test accuracy: 0.7982000112533569
```

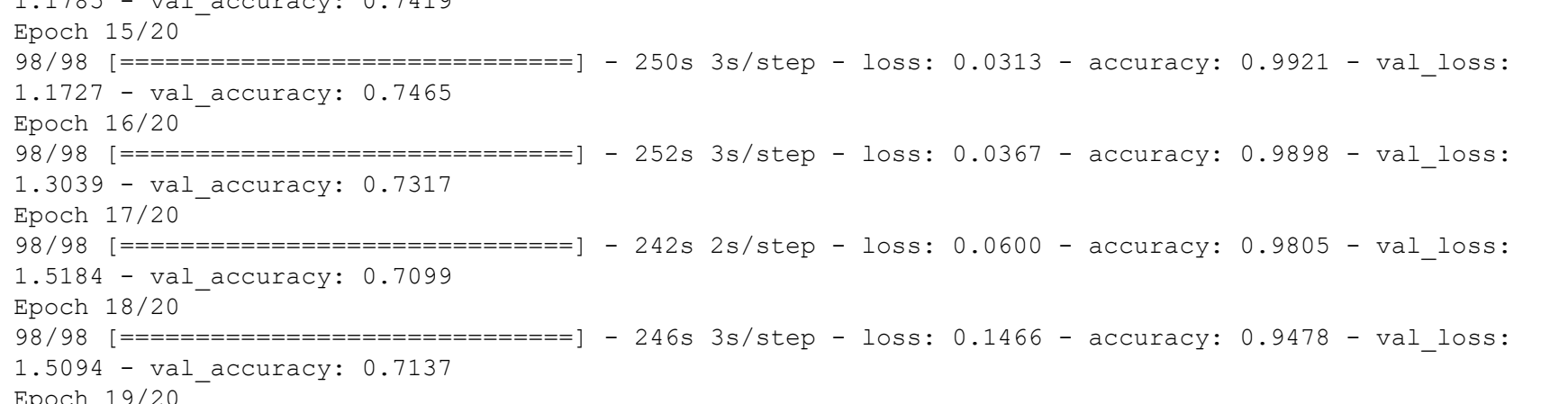



```
Training MLP took 4637.939530134201 seconds
```

dropput=no, batch_normalization=antes, capa_extra=no

In [75]:
```python
model = build_conv1(dout=False, bn=True, bn_before=False, extra_layer=False)

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Training
start = time.time()
history = model.fit(X_train, Y_train, batch_size=512, epochs=20, verbose=1, validation_data=(X_test, Y_
test))
end = time.time()

loss, acc = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', acc)

plot_model_history(history)
print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "model_20"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_28 (InputLayer)        [(None, 32, 32, 3)]       0
_________________________________________________________________
conv2d_162 (Conv2D)          (None, 32, 32, 32)        896
_________________________________________________________________
activation_245 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
batch_normalization_180 (Bat (None, 32, 32, 32)        128
_________________________________________________________________
conv2d_163 (Conv2D)          (None, 32, 32, 32)        9248
_________________________________________________________________
activation_246 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
batch_normalization_181 (Bat (None, 32, 32, 32)        128
_________________________________________________________________
max_pooling2d_81 (MaxPooling (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_164 (Conv2D)          (None, 16, 16, 64)        18496
_________________________________________________________________
activation_247 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
batch_normalization_182 (Bat (None, 16, 16, 64)        256
_________________________________________________________________
conv2d_165 (Conv2D)          (None, 16, 16, 64)        36928
_________________________________________________________________
activation_248 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
batch_normalization_183 (Bat (None, 16, 16, 64)        256
_________________________________________________________________
max_pooling2d_82 (MaxPooling (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_166 (Conv2D)          (None, 8, 8, 128)         73856
_________________________________________________________________
activation_249 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
batch_normalization_184 (Bat (None, 8, 8, 128)         512
_________________________________________________________________
conv2d_167 (Conv2D)          (None, 8, 8, 128)         147584
_________________________________________________________________
activation_250 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
batch_normalization_185 (Bat (None, 8, 8, 128)         512
_________________________________________________________________
max_pooling2d_83 (MaxPooling (None, 4, 4, 128)         0
_________________________________________________________________
flatten_46 (Flatten)         (None, 2048)              0
_________________________________________________________________
dense_122 (Dense)            (None, 10)                20490
=================================================================
Total params: 309,290
Trainable params: 308,394
Non-trainable params: 896
_________________________________________________________________
Epoch 1/20
98/98 [==============================] - 245s 2s/step - loss: 1.8790 - accuracy: 0.4127 - val_loss:
2.8086 - val_accuracy: 0.1024
Epoch 2/20
98/98 [==============================] - 236s 2s/step - loss: 0.9649 - accuracy: 0.6610 - val_loss:
3.2265 - val_accuracy: 0.1224
Epoch 3/20
98/98 [==============================] - 243s 2s/step - loss: 0.7243 - accuracy: 0.7515 - val_loss:
2.8489 - val_accuracy: 0.2116
Epoch 4/20
98/98 [==============================] - 240s 2s/step - loss: 0.5609 - accuracy: 0.8097 - val_loss:
1.9383 - val_accuracy: 0.3648
Epoch 5/20
98/98 [==============================] - 245s 3s/step - loss: 0.4444 - accuracy: 0.8482 - val_loss:
1.0935 - val_accuracy: 0.6285
Epoch 6/20
98/98 [==============================] - 235s 2s/step - loss: 0.3418 - accuracy: 0.8867 - val_loss:
0.9365 - val_accuracy: 0.6906
Epoch 7/20
98/98 [==============================] - 241s 2s/step - loss: 0.2485 - accuracy: 0.9182 - val_loss:
0.9284 - val_accuracy: 0.7090
Epoch 8/20
98/98 [==============================] - 245s 3s/step - loss: 0.1726 - accuracy: 0.9493 - val_loss:
0.9381 - val_accuracy: 0.7263
Epoch 9/20
98/98 [==============================] - 237s 2s/step - loss: 0.1228 - accuracy: 0.9650 - val_loss:
0.9338 - val_accuracy: 0.7430
Epoch 10/20
98/98 [==============================] - 241s 2s/step - loss: 0.0869 - accuracy: 0.9758 - val_loss:
0.9874 - val_accuracy: 0.7448
Epoch 11/20
98/98 [==============================] - 236s 2s/step - loss: 0.0662 - accuracy: 0.9818 - val_loss:
1.0267 - val_accuracy: 0.7423
Epoch 12/20
98/98 [==============================] - 236s 2s/step - loss: 0.0550 - accuracy: 0.9846 - val_loss:
1.0928 - val_accuracy: 0.7366
Epoch 13/20
98/98 [==============================] - 240s 2s/step - loss: 0.0417 - accuracy: 0.9901 - val_loss:
1.1042 - val_accuracy: 0.7522
Epoch 14/20
98/98 [==============================] - 243s 2s/step - loss: 0.0337 - accuracy: 0.9917 - val_loss:
1.1785 - val_accuracy: 0.7419
Epoch 15/20
98/98 [==============================] - 250s 3s/step - loss: 0.0313 - accuracy: 0.9921 - val_loss:
1.1727 - val_accuracy: 0.7465
Epoch 16/20
98/98 [==============================] - 252s 3s/step - loss: 0.0367 - accuracy: 0.9898 - val_loss:
1.3039 - val_accuracy: 0.7317
Epoch 17/20
98/98 [==============================] - 242s 2s/step - loss: 0.0600 - accuracy: 0.9805 - val_loss:
1.5184 - val_accuracy: 0.7099
Epoch 18/20
98/98 [==============================] - 246s 3s/step - loss: 0.1466 - accuracy: 0.9478 - val_loss:
1.5094 - val_accuracy: 0.7137
Epoch 19/20
98/98 [==============================] - 235s 2s/step - loss: 0.1208 - accuracy: 0.9562 - val_loss:
1.4203 - val_accuracy: 0.7233
Epoch 20/20
98/98 [==============================] - 239s 2s/step - loss: 0.0669 - accuracy: 0.9764 - val_loss:
1.1963 - val_accuracy: 0.7581
Test loss: 1.1962941884994507
Test accuracy: 0.7580999732017517
```

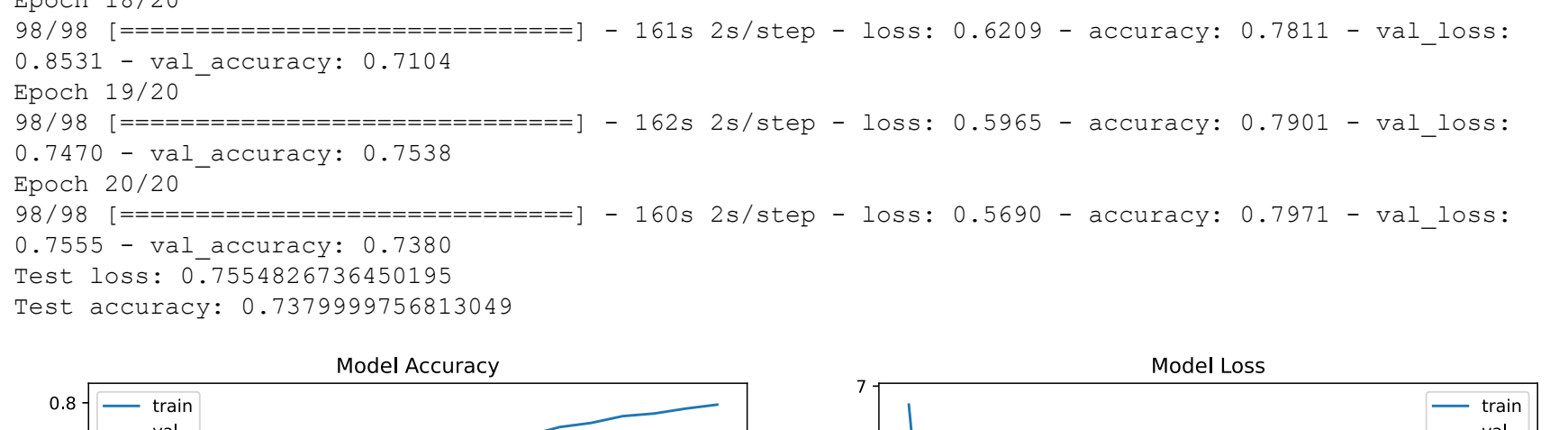



```
Training MLP took 4822.8211834430695 seconds
```

dropput=si, batch_normalization=despues, capa_extra=si

In [76]:
```python
model = build_conv1(dout=True, bn=True, bn_before=True, extra_layer=True)

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Training
start = time.time()
history = model.fit(X_train, Y_train, batch_size=512, epochs=20, verbose=1, validation_data=(X_test, Y_
test))
end = time.time()

loss, acc = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', acc)

plot_model_history(history)
print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "model_21"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_29 (InputLayer)        [(None, 32, 32, 3)]       0
_________________________________________________________________
conv2d_168 (Conv2D)          (None, 32, 32, 32)        896
_________________________________________________________________
batch_normalization_186 (Bat (None, 32, 32, 32)        128
_________________________________________________________________
activation_251 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_169 (Conv2D)          (None, 32, 32, 32)        9248
_________________________________________________________________
batch_normalization_187 (Bat (None, 32, 32, 32)        128
_________________________________________________________________
activation_252 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
max_pooling2d_84 (MaxPooling (None, 16, 16, 32)        0
_________________________________________________________________
dropout_98 (Dropout)         (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_170 (Conv2D)          (None, 16, 16, 64)        18496
_________________________________________________________________
batch_normalization_188 (Bat (None, 16, 16, 64)        256
_________________________________________________________________
activation_253 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_171 (Conv2D)          (None, 16, 16, 64)        36928
_________________________________________________________________
batch_normalization_189 (Bat (None, 16, 16, 64)        256
_________________________________________________________________
activation_254 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
max_pooling2d_85 (MaxPooling (None, 8, 8, 64)          0
_________________________________________________________________
dropout_99 (Dropout)         (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_172 (Conv2D)          (None, 8, 8, 128)         73856
_________________________________________________________________
batch_normalization_190 (Bat (None, 8, 8, 128)         512
_________________________________________________________________
activation_255 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
conv2d_173 (Conv2D)          (None, 8, 8, 128)         147584
_________________________________________________________________
batch_normalization_191 (Bat (None, 8, 8, 128)         512
_________________________________________________________________
activation_256 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
max_pooling2d_86 (MaxPooling (None, 4, 4, 128)         0
_________________________________________________________________
dropout_100 (Dropout)        (None, 4, 4, 128)         0
_________________________________________________________________
flatten_47 (Flatten)         (None, 2048)              0
_________________________________________________________________
dense_123 (Dense)            (None, 2048)              4196352
_________________________________________________________________
dense_124 (Dense)            (None, 10)                20490
=================================================================
Total params: 4,505,642
Trainable params: 4,504,746
Non-trainable params: 896
_________________________________________________________________
Epoch 1/20
98/98 [==============================] - 273s 3s/step - loss: 13.6607 - accuracy: 0.1343 - val_loss:
2.3084 - val_accuracy: 0.1408
Epoch 2/20
98/98 [==============================] - 261s 3s/step - loss: 2.0823 - accuracy: 0.2440 - val_loss:
2.7084 - val_accuracy: 0.1371
Epoch 3/20
98/98 [==============================] - 270s 3s/step - loss: 1.7005 - accuracy: 0.3808 - val_loss:
3.8148 - val_accuracy: 0.1343
Epoch 4/20
98/98 [==============================] - 274s 3s/step - loss: 1.5201 - accuracy: 0.4470 - val_loss:
3.3705 - val_accuracy: 0.1608
Epoch 5/20
98/98 [==============================] - 265s 3s/step - loss: 1.3712 - accuracy: 0.5041 - val_loss:
1.8322 - val_accuracy: 0.3803
Epoch 6/20
98/98 [==============================] - 259s 3s/step - loss: 1.2398 - accuracy: 0.5528 - val_loss:
1.8380 - val_accuracy: 0.3897
Epoch 7/20
98/98 [==============================] - 257s 3s/step - loss: 1.1107 - accuracy: 0.6027 - val_loss:
1.8236 - val_accuracy: 0.4052
Epoch 8/20
98/98 [==============================] - 282s 3s/step - loss: 1.0239 - accuracy: 0.6383 - val_loss:
1.2998 - val_accuracy: 0.5443
Epoch 9/20
98/98 [==============================] - 272s 3s/step - loss: 0.9658 - accuracy: 0.6649 - val_loss:
0.9779 - val_accuracy: 0.6523
Epoch 10/20
98/98 [==============================] - 192s 2s/step - loss: 0.8886 - accuracy: 0.6879 - val_loss:
1.2017 - val_accuracy: 0.5987
Epoch 11/20
98/98 [==============================] - 168s 2s/step - loss: 0.8295 - accuracy: 0.7051 - val_loss:
0.8470 - val_accuracy: 0.7031
Epoch 12/20
98/98 [==============================] - 171s 2s/step - loss: 0.8065 - accuracy: 0.7172 - val_loss:
0.9333 - val_accuracy: 0.6670
Epoch 13/20
98/98 [==============================] - 180s 2s/step - loss: 0.7627 - accuracy: 0.7294 - val_loss:
0.8081 - val_accuracy: 0.7158
Epoch 14/20
98/98 [==============================] - 176s 2s/step - loss: 0.7176 - accuracy: 0.7498 - val_loss:
0.8537 - val_accuracy: 0.7009
Epoch 15/20
98/98 [==============================] - 178s 2s/step - loss: 0.6719 - accuracy: 0.7645 - val_loss:
0.8021 - val_accuracy: 0.7202
Epoch 16/20
98/98 [==============================] - 162s 2s/step - loss: 0.6642 - accuracy: 0.7685 - val_loss:
0.7894 - val_accuracy: 0.7241
Epoch 17/20
98/98 [==============================] - 162s 2s/step - loss: 0.6221 - accuracy: 0.7817 - val_loss:
0.7599 - val_accuracy: 0.7351
Epoch 18/20
98/98 [==============================] - 161s 2s/step - loss: 0.6209 - accuracy: 0.7811 - val_loss:
0.8531 - val_accuracy: 0.7104
Epoch 19/20
98/98 [==============================] - 162s 2s/step - loss: 0.5965 - accuracy: 0.7901 - val_loss:
0.7470 - val_accuracy: 0.7538
Epoch 20/20
98/98 [==============================] - 160s 2s/step - loss: 0.5690 - accuracy: 0.7971 - val_loss:
0.7555 - val_accuracy: 0.7380
Test loss: 0.7554826736450195
Test accuracy: 0.7379999756813049
```

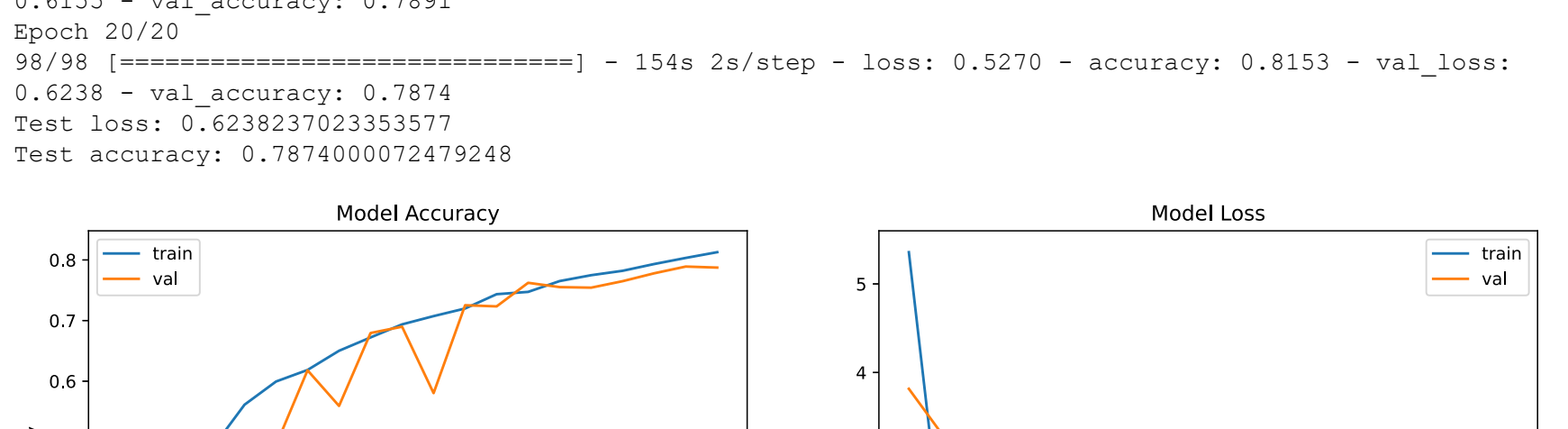



```
Training MLP took 4287.035219669342 seconds
```

dropput=si, batch_normalization=antes, capa_extra=si

In [77]:
```python
model = build_conv1(dout=True, bn=True, bn_before=False, extra_layer=True)

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Training
start = time.time()
history = model.fit(X_train, Y_train, batch_size=512, epochs=20, verbose=1, validation_data=(X_test, Y_
test))
end = time.time()

loss, acc = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', acc)

plot_model_history(history)
print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "model_22"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_30 (InputLayer)        [(None, 32, 32, 3)]       0
_________________________________________________________________
conv2d_174 (Conv2D)          (None, 32, 32, 32)        896
_________________________________________________________________
activation_257 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
batch_normalization_192 (Bat (None, 32, 32, 32)        128
_________________________________________________________________
conv2d_175 (Conv2D)          (None, 32, 32, 32)        9248
_________________________________________________________________
activation_258 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
batch_normalization_193 (Bat (None, 32, 32, 32)        128
_________________________________________________________________
max_pooling2d_87 (MaxPooling (None, 16, 16, 32)        0
_________________________________________________________________
dropout_101 (Dropout)        (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_176 (Conv2D)          (None, 16, 16, 64)        18496
_________________________________________________________________
activation_259 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
batch_normalization_194 (Bat (None, 16, 16, 64)        256
_________________________________________________________________
conv2d_177 (Conv2D)          (None, 16, 16, 64)        36928
_________________________________________________________________
activation_260 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
batch_normalization_195 (Bat (None, 16, 16, 64)        256
_________________________________________________________________
max_pooling2d_88 (MaxPooling (None, 8, 8, 64)          0
_________________________________________________________________
dropout_102 (Dropout)        (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_178 (Conv2D)          (None, 8, 8, 128)         73856
_________________________________________________________________
activation_261 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
batch_normalization_196 (Bat (None, 8, 8, 128)         512
_________________________________________________________________
conv2d_179 (Conv2D)          (None, 8, 8, 128)         147584
_________________________________________________________________
activation_262 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
batch_normalization_197 (Bat (None, 8, 8, 128)         512
_________________________________________________________________
max_pooling2d_89 (MaxPooling (None, 4, 4, 128)         0
_________________________________________________________________
dropout_103 (Dropout)        (None, 4, 4, 128)         0
_________________________________________________________________
flatten_48 (Flatten)         (None, 2048)              0
_________________________________________________________________
dense_125 (Dense)            (None, 2048)              4196352
_________________________________________________________________
dense_126 (Dense)            (None, 10)                20490
=================================================================
Total params: 4,505,642
Trainable params: 4,504,746
Non-trainable params: 896
_________________________________________________________________
Epoch 1/20
98/98 [==============================] - 156s 2s/step - loss: 10.4629 - accuracy: 0.2396 - val_loss:
3.8159 - val_accuracy: 0.1082
Epoch 2/20
98/98 [==============================] - 154s 2s/step - loss: 2.2856 - accuracy: 0.3654 - val_loss:
3.3293 - val_accuracy: 0.1480
Epoch 3/20
98/98 [==============================] - 154s 2s/step - loss: 1.9104 - accuracy: 0.4512 - val_loss:
2.4660 - val_accuracy: 0.1603
Epoch 4/20
98/98 [==============================] - 154s 2s/step - loss: 1.9018 - accuracy: 0.4832 - val_loss:
2.0876 - val_accuracy: 0.3033
Epoch 5/20
98/98 [==============================] - 154s 2s/step - loss: 1.4676 - accuracy: 0.5522 - val_loss:
2.3175 - val_accuracy: 0.3037
Epoch 6/20
98/98 [==============================] - 153s 2s/step - loss: 1.3111 - accuracy: 0.5907 - val_loss:
1.4977 - val_accuracy: 0.4945
Epoch 7/20
98/98 [==============================] - 154s 2s/step - loss: 1.2267 - accuracy: 0.6152 - val_loss:
1.0683 - val_accuracy: 0.6180
Epoch 8/20
98/98 [==============================] - 154s 2s/step - loss: 1.1245 - accuracy: 0.6474 - val_loss:
1.2720 - val_accuracy: 0.5592
Epoch 9/20
98/98 [==============================] - 155s 2s/step - loss: 1.0452 - accuracy: 0.6706 - val_loss:
1.1077 - val_accuracy: 0.6795
Epoch 10/20
98/98 [==============================] - 154s 2s/step - loss: 0.9530 - accuracy: 0.6942 - val_loss:
0.9882 - val_accuracy: 0.6899
Epoch 11/20
98/98 [==============================] - 155s 2s/step - loss: 0.8970 - accuracy: 0.7116 - val_loss:
1.3468 - val_accuracy: 0.5803
Epoch 12/20
98/98 [==============================] - 154s 2s/step - loss: 0.8600 - accuracy: 0.7209 - val_loss:
0.9206 - val_accuracy: 0.7253
Epoch 13/20
98/98 [==============================] - 154s 2s/step - loss: 0.7745 - accuracy: 0.7419 - val_loss:
0.8851 - val_accuracy: 0.7234
Epoch 14/20
98/98 [==============================] - 154s 2s/step - loss: 0.7645 - accuracy: 0.7504 - val_loss:
0.7113 - val_accuracy: 0.7623
Epoch 15/20
98/98 [==============================] - 155s 2s/step - loss: 0.6888 - accuracy: 0.7663 - val_loss:
0.7216 - val_accuracy: 0.7552
Epoch 16/20
98/98 [==============================] - 154s 2s/step - loss: 0.6601 - accuracy: 0.7784 - val_loss:
0.7314 - val_accuracy: 0.7543
Epoch 17/20
98/98 [==============================] - 154s 2s/step - loss: 0.6383 - accuracy: 0.7839 - val_loss:
0.6886 - val_accuracy: 0.7650
Epoch 18/20
98/98 [==============================] - 154s 2s/step - loss: 0.5864 - accuracy: 0.7947 - val_loss:
0.6585 - val_accuracy: 0.7780
Epoch 19/20
98/98 [==============================] - 155s 2s/step - loss: 0.5485 - accuracy: 0.8069 - val_loss:
0.6155 - val_accuracy: 0.7891
Epoch 20/20
98/98 [==============================] - 154s 2s/step - loss: 0.5270 - accuracy: 0.8153 - val_loss:
0.6238 - val_accuracy: 0.7874
Test loss: 0.6238237023353577
Test accuracy: 0.7874000072479248
```

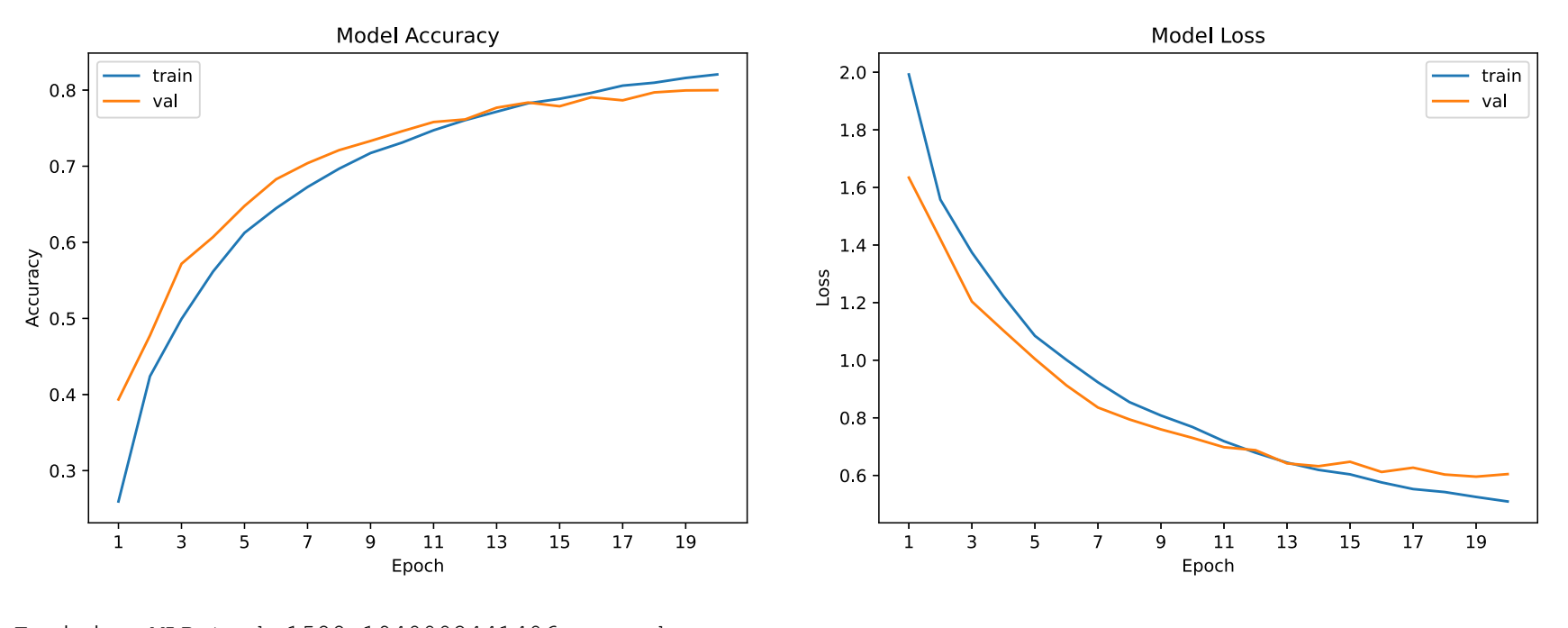



```
Training MLP took 3084.4385986328125 seconds
```

dropput=si, batch_normalization=no, capa_extra=si

In [78]:
```python
model = build_conv1(dout=True, bn=False, extra_layer=True)

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.summary()

# Training
start = time.time()
history = model.fit(X_train, Y_train, batch_size=512, epochs=20, verbose=1, validation_data=(X_test, Y_
test))
end = time.time()

loss, acc = model.evaluate(X_test, Y_test, verbose=0)
print('Test loss:', loss)
print('Test accuracy:', acc)

plot_model_history(history)
print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "model_23"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_31 (InputLayer)        [(None, 32, 32, 3)]       0
_________________________________________________________________
conv2d_180 (Conv2D)          (None, 32, 32, 32)        896
_________________________________________________________________
activation_263 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_181 (Conv2D)          (None, 32, 32, 32)        9248
_________________________________________________________________
activation_264 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
max_pooling2d_90 (MaxPooling (None, 16, 16, 32)        0
_________________________________________________________________
dropout_104 (Dropout)        (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_182 (Conv2D)          (None, 16, 16, 64)        18496
_________________________________________________________________
activation_265 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_183 (Conv2D)          (None, 16, 16, 64)        36928
_________________________________________________________________
activation_266 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
max_pooling2d_91 (MaxPooling (None, 8, 8, 64)          0
_________________________________________________________________
dropout_105 (Dropout)        (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_184 (Conv2D)          (None, 8, 8, 128)         73856
_________________________________________________________________
activation_267 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
conv2d_185 (Conv2D)          (None, 8, 8, 128)         147584
_________________________________________________________________
activation_268 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
max_pooling2d_92 (MaxPooling (None, 4, 4, 128)         0
_________________________________________________________________
dropout_106 (Dropout)        (None, 4, 4, 128)         0
_________________________________________________________________
flatten_49 (Flatten)         (None, 2048)              0
_________________________________________________________________
dense_127 (Dense)            (None, 2048)              4196352
_________________________________________________________________
dense_128 (Dense)            (None, 10)                20490
=================================================================
Total params: 4,503,850
Trainable params: 4,503,850
Non-trainable params: 0
_________________________________________________________________
Epoch 1/20
98/98 [==============================] - 80s 813ms/step - loss: 2.2148 - accuracy: 0.1764 - val_loss:
1.6340 - val_accuracy: 0.3935
Epoch 2/20
98/98 [==============================] - 79s 808ms/step - loss: 1.6101 - accuracy: 0.4019 - val_loss:
1.4203 - val_accuracy: 0.4776
Epoch 3/20
98/98 [==============================] - 79s 811ms/step - loss: 1.4052 - accuracy: 0.4838 - val_loss:
1.2038 - val_accuracy: 0.5718
Epoch 4/20
98/98 [==============================] - 79s 810ms/step - loss: 1.2539 - accuracy: 0.5483 - val_loss:
1.1032 - val_accuracy: 0.6069
Epoch 5/20
98/98 [==============================] - 80s 815ms/step - loss: 1.1000 - accuracy: 0.6054 - val_loss:
1.0045 - val_accuracy: 0.6438
Epoch 6/20
98/98 [==============================] - 80s 820ms/step - loss: 1.0147 - accuracy: 0.6396 - val_loss:
0.9127 - val_accuracy: 0.6828
Epoch 7/20
98/98 [==============================] - 80s 813ms/step - loss: 0.9388 - accuracy: 0.6676 - val_loss:
0.8358 - val_accuracy: 0.7040
Epoch 8/20
98/98 [==============================] - 79s 808ms/step - loss: 0.8667 - accuracy: 0.6905 - val_loss:
0.7945 - val_accuracy: 0.7211
Epoch 9/20
98/98 [==============================] - 79s 809ms/step - loss: 0.8135 - accuracy: 0.7151 - val_loss:
0.7599 - val_accuracy: 0.7332
Epoch 10/20
98/98 [==============================] - 79s 809ms/step - loss: 0.7691 - accuracy: 0.7316 - val_loss:
0.7304 - val_accuracy: 0.7460
Epoch 11/20
98/98 [==============================] - 79s 808ms/step - loss: 0.7160 - accuracy: 0.7483 - val_loss:
0.6979 - val_accuracy: 0.7581
Epoch 12/20
98/98 [==============================] - 80s 816ms/step - loss: 0.6780 - accuracy: 0.7596 - val_loss:
0.6877 - val_accuracy: 0.7615
Epoch 13/20
98/98 [==============================] - 80s 816ms/step - loss: 0.6447 - accuracy: 0.7735 - val_loss:
0.6418 - val_accuracy: 0.7769
Epoch 14/20
98/98 [==============================] - 79s 807ms/step - loss: 0.6186 - accuracy: 0.7828 - val_loss:
0.6321 - val_accuracy: 0.7837
Epoch 15/20
98/98 [==============================] - 79s 807ms/step - loss: 0.5997 - accuracy: 0.7890 - val_loss:
0.6475 - val_accuracy: 0.7788
Epoch 16/20
98/98 [==============================] - 79s 805ms/step - loss: 0.5816 - accuracy: 0.7934 - val_loss:
0.6120 - val_accuracy: 0.7905
Epoch 17/20
98/98 [==============================] - 79s 807ms/step - loss: 0.5454 - accuracy: 0.8080 - val_loss:
0.6269 - val_accuracy: 0.7866
Epoch 18/20
98/98 [==============================] - 79s 806ms/step - loss: 0.5490 - accuracy: 0.8079 - val_loss:
0.6032 - val_accuracy: 0.7970
Epoch 19/20
98/98 [==============================] - 79s 807ms/step - loss: 0.5181 - accuracy: 0.8173 - val_loss:
0.5958 - val_accuracy: 0.7996
Epoch 20/20
98/98 [==============================] - 79s 810ms/step - loss: 0.5034 - accuracy: 0.8221 - val_loss:
0.6046 - val_accuracy: 0.7999
Test loss: 0.6046326160430908
Test accuracy: 0.7999999953269958
```

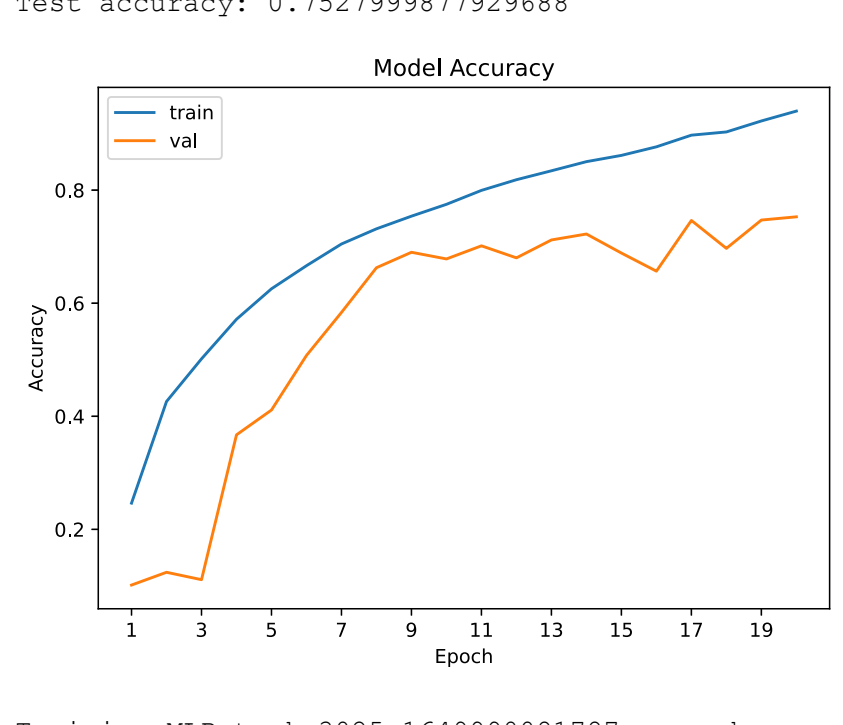


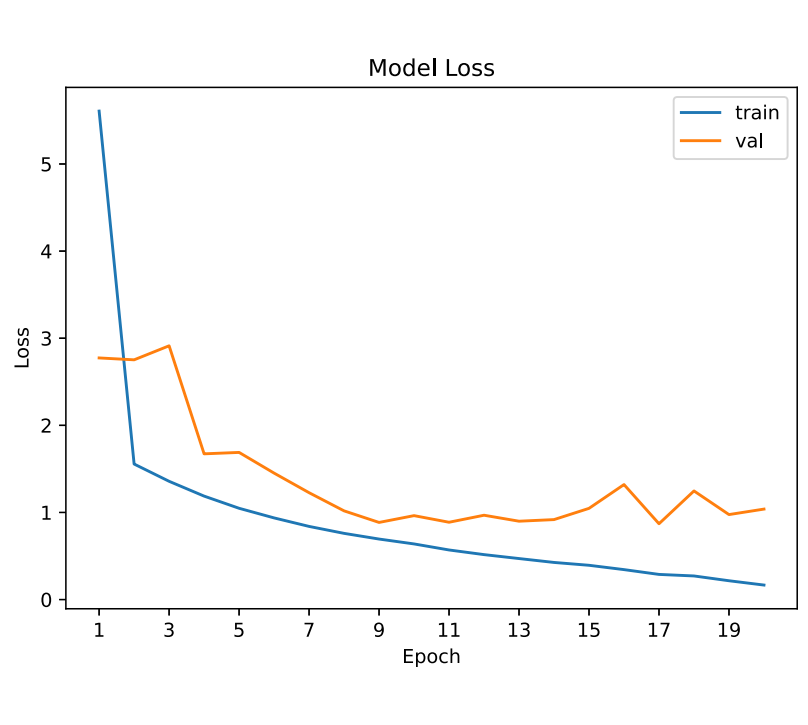

```
Training MLP took 1588.1940002441406 seconds
```

dropput=no, batch_normalization=despues, capa_extra=si

```
In [79]: model = build_conv1(dout=False, bn=True, bn_before=True, extra_layer=True)

         model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
         model.summary()

         # Training
         start = time.time()
         history = model.fit(X_train, Y_train, batch_size=512, epochs=20, verbose=1, validation_data=(X_test, Y_
         test))
         end = time.time()

         loss, acc = model.evaluate(X_test, Y_test, verbose=0)
         print('Test loss:', loss)
         print('Test accuracy:', acc)

         plot_model_history(history)
         print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "model_24"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_32 (InputLayer)        [(None, 32, 32, 3)]       0
_________________________________________________________________
conv2d_186 (Conv2D)          (None, 32, 32, 32)        896
_________________________________________________________________
batch_normalization_198 (Bat (None, 32, 32, 32)        128
_________________________________________________________________
activation_269 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_187 (Conv2D)          (None, 32, 32, 32)        9248
_________________________________________________________________
batch_normalization_199 (Bat (None, 32, 32, 32)        128
_________________________________________________________________
activation_270 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
max_pooling2d_93 (MaxPooling (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_188 (Conv2D)          (None, 16, 16, 64)        18496
_________________________________________________________________
batch_normalization_200 (Bat (None, 16, 16, 64)        256
_________________________________________________________________
activation_271 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_189 (Conv2D)          (None, 16, 16, 64)        36928
_________________________________________________________________
batch_normalization_201 (Bat (None, 16, 16, 64)        256
_________________________________________________________________
activation_272 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
max_pooling2d_94 (MaxPooling (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_190 (Conv2D)          (None, 8, 8, 128)         73856
_________________________________________________________________
batch_normalization_202 (Bat (None, 8, 8, 128)         512
_________________________________________________________________
activation_273 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
conv2d_191 (Conv2D)          (None, 8, 8, 128)         147584
_________________________________________________________________
batch_normalization_203 (Bat (None, 8, 8, 128)         512
_________________________________________________________________
activation_274 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
max_pooling2d_95 (MaxPooling (None, 4, 4, 128)         0
_________________________________________________________________
flatten_50 (Flatten)         (None, 2048)              0
_________________________________________________________________
dense_129 (Dense)            (None, 2048)              4196352
_________________________________________________________________
dense_130 (Dense)            (None, 10)                20490
=================================================================
Total params: 4,505,642
Trainable params: 4,504,746
Non-trainable params: 896
_________________________________________________________________
Epoch 1/20
98/98 [==============================] - 153s 2s/step - loss: 11.6732 - accuracy: 0.1837 - val_loss:
2.7736 - val_accuracy: 0.1015
Epoch 2/20
98/98 [==============================] - 151s 2s/step - loss: 1.6263 - accuracy: 0.4026 - val_loss:
2.7520 - val_accuracy: 0.1241
Epoch 3/20
98/98 [==============================] - 151s 2s/step - loss: 1.3786 - accuracy: 0.4932 - val_loss:
2.9129 - val_accuracy: 0.1110
Epoch 4/20
98/98 [==============================] - 152s 2s/step - loss: 1.2295 - accuracy: 0.5521 - val_loss:
1.6728 - val_accuracy: 0.3672
Epoch 5/20
98/98 [==============================] - 151s 2s/step - loss: 1.0704 - accuracy: 0.6176 - val_loss:
1.6891 - val_accuracy: 0.4111
Epoch 6/20
98/98 [==============================] - 151s 2s/step - loss: 0.9543 - accuracy: 0.6568 - val_loss:
1.4516 - val_accuracy: 0.5076
Epoch 7/20
98/98 [==============================] - 151s 2s/step - loss: 0.8655 - accuracy: 0.6946 - val_loss:
1.2267 - val_accuracy: 0.5837
Epoch 8/20
98/98 [==============================] - 152s 2s/step - loss: 0.7584 - accuracy: 0.7312 - val_loss:
1.0182 - val_accuracy: 0.6628
Epoch 9/20
98/98 [==============================] - 151s 2s/step - loss: 0.6803 - accuracy: 0.7604 - val_loss:
0.8854 - val_accuracy: 0.6901
Epoch 10/20
98/98 [==============================] - 151s 2s/step - loss: 0.6333 - accuracy: 0.7780 - val_loss:
0.9634 - val_accuracy: 0.6783
Epoch 11/20
98/98 [==============================] - 151s 2s/step - loss: 0.5610 - accuracy: 0.8023 - val_loss:
0.8875 - val_accuracy: 0.7015
Epoch 12/20
98/98 [==============================] - 151s 2s/step - loss: 0.5118 - accuracy: 0.8199 - val_loss:
0.9680 - val_accuracy: 0.6802
Epoch 13/20
98/98 [==============================] - 151s 2s/step - loss: 0.4629 - accuracy: 0.8366 - val_loss:
0.8995 - val_accuracy: 0.7119
Epoch 14/20
98/98 [==============================] - 151s 2s/step - loss: 0.4090 - accuracy: 0.8558 - val_loss:
0.9180 - val_accuracy: 0.7224
Epoch 15/20
98/98 [==============================] - 151s 2s/step - loss: 0.3783 - accuracy: 0.8669 - val_loss:
1.0470 - val_accuracy: 0.6887
Epoch 16/20
98/98 [==============================] - 153s 2s/step - loss: 0.3286 - accuracy: 0.8822 - val_loss:
1.3201 - val_accuracy: 0.6567
Epoch 17/20
98/98 [==============================] - 151s 2s/step - loss: 0.2712 - accuracy: 0.9029 - val_loss:
0.8703 - val_accuracy: 0.7464
Epoch 18/20
98/98 [==============================] - 151s 2s/step - loss: 0.2691 - accuracy: 0.9043 - val_loss:
1.2469 - val_accuracy: 0.6970
Epoch 19/20
98/98 [==============================] - 151s 2s/step - loss: 0.2079 - accuracy: 0.9255 - val_loss:
0.9756 - val_accuracy: 0.7471
Epoch 20/20
98/98 [==============================] - 152s 2s/step - loss: 0.1505 - accuracy: 0.9472 - val_loss:
1.0386 - val_accuracy: 0.7528
Test loss: 1.0385514497756958
Test accuracy: 0.7527999877929688
```



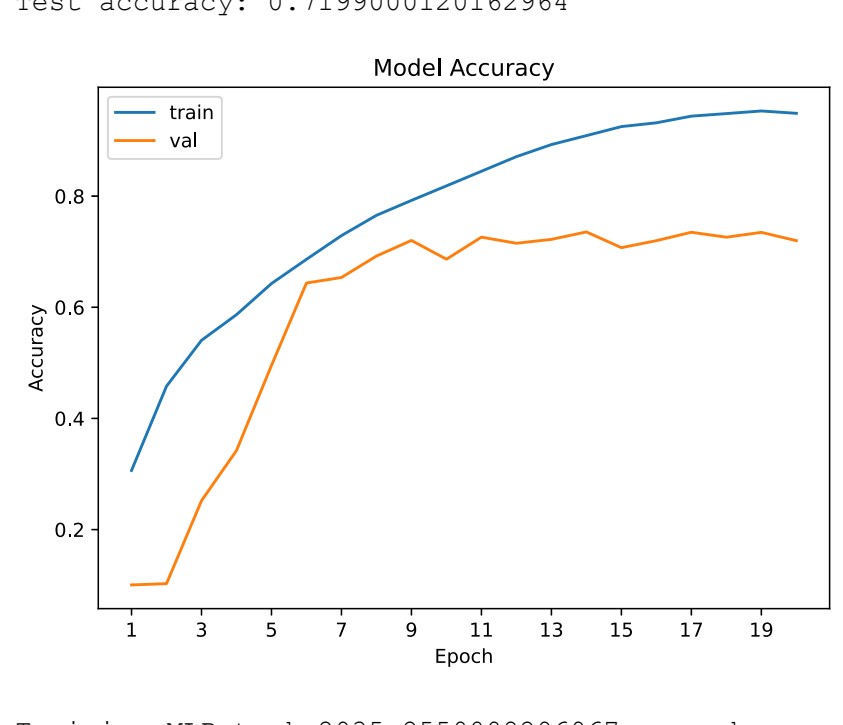


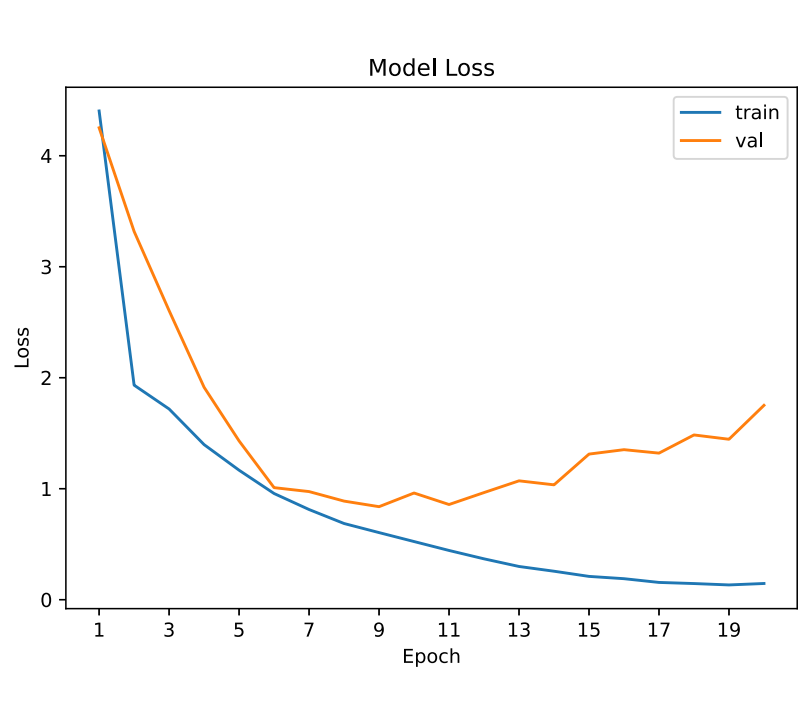

```
Training MLP took 3025.1649990081787 seconds
```

dropput=no, batch_normalization=antes, capa_extra=si

```
In [80]: model = build_conv1(dout=False, bn=True, bn_before=False, extra_layer=True)

         model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
         model.summary()

         # Training
         start = time.time()
         history = model.fit(X_train, Y_train, batch_size=512, epochs=20, verbose=1, validation_data=(X_test, Y_
         test))
         end = time.time()

         loss, acc = model.evaluate(X_test, Y_test, verbose=0)
         print('Test loss:', loss)
         print('Test accuracy:', acc)

         plot_model_history(history)
         print("Training MLP took " + str(end - start) + " seconds")
```

```
Model: "model_25"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_33 (InputLayer)        [(None, 32, 32, 3)]       0
_________________________________________________________________
conv2d_192 (Conv2D)          (None, 32, 32, 32)        896
_________________________________________________________________
activation_275 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
batch_normalization_204 (Bat (None, 32, 32, 32)        128
_________________________________________________________________
conv2d_193 (Conv2D)          (None, 32, 32, 32)        9248
_________________________________________________________________
activation_276 (Activation)  (None, 32, 32, 32)        0
_________________________________________________________________
batch_normalization_205 (Bat (None, 32, 32, 32)        128
_________________________________________________________________
max_pooling2d_96 (MaxPooling (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_194 (Conv2D)          (None, 16, 16, 64)        18496
_________________________________________________________________
activation_277 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
batch_normalization_206 (Bat (None, 16, 16, 64)        256
_________________________________________________________________
conv2d_195 (Conv2D)          (None, 16, 16, 64)        36928
_________________________________________________________________
activation_278 (Activation)  (None, 16, 16, 64)        0
_________________________________________________________________
batch_normalization_207 (Bat (None, 16, 16, 64)        256
_________________________________________________________________
max_pooling2d_97 (MaxPooling (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_196 (Conv2D)          (None, 8, 8, 128)         73856
_________________________________________________________________
activation_279 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
batch_normalization_208 (Bat (None, 8, 8, 128)         512
_________________________________________________________________
conv2d_197 (Conv2D)          (None, 8, 8, 128)         147584
_________________________________________________________________
activation_280 (Activation)  (None, 8, 8, 128)         0
_________________________________________________________________
batch_normalization_209 (Bat (None, 8, 8, 128)         512
_________________________________________________________________
max_pooling2d_98 (MaxPooling (None, 4, 4, 128)         0
_________________________________________________________________
flatten_51 (Flatten)         (None, 2048)              0
_________________________________________________________________
dense_131 (Dense)            (None, 2048)              4196352
_________________________________________________________________
dense_132 (Dense)            (None, 10)                20490
=================================================================
Total params: 4,505,642
Trainable params: 4,504,746
Non-trainable params: 896
_________________________________________________________________
Epoch 1/20
98/98 [==============================] - 147s 1s/step - loss: 7.9319 - accuracy: 0.2507 - val_loss:
4.2513 - val_accuracy: 0.1004
Epoch 2/20
98/98 [==============================] - 146s 1s/step - loss: 2.0367 - accuracy: 0.4284 - val_loss:
3.3177 - val_accuracy: 0.1027
Epoch 3/20
98/98 [==============================] - 146s 1s/step - loss: 1.8096 - accuracy: 0.5220 - val_loss:
2.6041 - val_accuracy: 0.2519
Epoch 4/20
98/98 [==============================] - 147s 1s/step - loss: 1.4932 - accuracy: 0.5818 - val_loss:
1.9135 - val_accuracy: 0.3423
Epoch 5/20
98/98 [==============================] - 147s 1s/step - loss: 1.1822 - accuracy: 0.6403 - val_loss:
1.4302 - val_accuracy: 0.4954
Epoch 6/20
98/98 [==============================] - 147s 1s/step - loss: 0.9767 - accuracy: 0.6797 - val_loss:
1.0093 - val_accuracy: 0.6437
Epoch 7/20
98/98 [==============================] - 146s 1s/step - loss: 0.8111 - accuracy: 0.7254 - val_loss:
0.9738 - val_accuracy: 0.6536
Epoch 8/20
98/98 [==============================] - 148s 2s/step - loss: 0.6800 - accuracy: 0.7660 - val_loss:
0.8881 - val_accuracy: 0.6922
Epoch 9/20
98/98 [==============================] - 147s 1s/step - loss: 0.5964 - accuracy: 0.7966 - val_loss:
0.8378 - val_accuracy: 0.7203
Epoch 10/20
98/98 [==============================] - 146s 1s/step - loss: 0.5053 - accuracy: 0.8236 - val_loss:
0.9610 - val_accuracy: 0.6865
Epoch 11/20
98/98 [==============================] - 146s 1s/step - loss: 0.4186 - accuracy: 0.8536 - val_loss:
0.8573 - val_accuracy: 0.7262
Epoch 12/20
98/98 [==============================] - 147s 2s/step - loss: 0.3336 - accuracy: 0.8825 - val_loss:
0.9650 - val_accuracy: 0.7152
Epoch 13/20
98/98 [==============================] - 147s 1s/step - loss: 0.2809 - accuracy: 0.8993 - val_loss:
1.0707 - val_accuracy: 0.7222
Epoch 14/20
98/98 [==============================] - 147s 1s/step - loss: 0.2322 - accuracy: 0.9177 - val_loss:
1.0346 - val_accuracy: 0.7356
Epoch 15/20
98/98 [==============================] - 147s 1s/step - loss: 0.1896 - accuracy: 0.9313 - val_loss:
1.3116 - val_accuracy: 0.7073
Epoch 16/20
98/98 [==============================] - 147s 2s/step - loss: 0.1824 - accuracy: 0.9345 - val_loss:
1.3514 - val_accuracy: 0.7199
Epoch 17/20
98/98 [==============================] - 147s 1s/step - loss: 0.1401 - accuracy: 0.9492 - val_loss:
1.3203 - val_accuracy: 0.7350
Epoch 18/20
98/98 [==============================] - 147s 1s/step - loss: 0.1277 - accuracy: 0.9544 - val_loss:
1.4841 - val_accuracy: 0.7261
Epoch 19/20
98/98 [==============================] - 147s 1s/step - loss: 0.1238 - accuracy: 0.9564 - val_loss:
1.4452 - val_accuracy: 0.7348
Epoch 20/20
98/98 [==============================] - 147s 1s/step - loss: 0.1322 - accuracy: 0.9539 - val_loss:
1.7509 - val_accuracy: 0.7199
Test loss: 1.7509175539016724
Test accuracy: 0.7199000120162964
```






```
Training MLP took 2935.2550008296967 seconds
```

```
In [ ]:
```