```
______
exp = Experimento_1b_1_3
modelo = 3
samples_per_class = 100
opti = Adam
batch = 128
epochs = 15
run = True
Creando juego de datos \dots
Entrenando ...
Tiempo de entrenamiento: ** 66.0 **
                 Accuracy
                                                      Loss
         training accuracy
          validation accurac
 0.4
0.2
                                      SS0
3
  0.0
                                                                 12.5
         2.5
              5.0
                      10.0
                           12.5
                                15.0
                                          0.0
                                              2.5
                                                   5.0
                                                            10.0
Evaluando ...
Exactitud validate: 40.0 %, Test: 43.33 %
Tiempo de ejecución del experimento: 67.0
                                             _____
exp = Experimento_1b_1_4
modelo = 4
samples_per_class = 100
opti = Adam
batch = 128
epochs = 15
run = True
Entrenando ...
Tiempo de entrenamiento: ** 40.0 **
                 Accuracy
 0.4
accnracy
0.2
                                      055
                     training accuracy
                     validation accuracy
         25
              5.0
                  7.5 10.0 12.5 15.0
                                          0.0
                                              2.5
                                                   5.0
                                                            10.0
                                                                 12.5
                   epoch
Evaluando ...
Exactitud validate: 42.4 %, Test: 43.44 %
Tiempo de ejecución del experimento: 41.0
exp = Experimento_1b_2_3
modelo = 3
samples_per_class = 100
opti = rmsprop
batch = 128
epochs = 15
run = True
Entrenando ...
Tiempo de entrenamiento: ** 66.0 **
                 Accuracy
                                                      Loss
  0.8
 accuracy
                                       3
                                      055
  0.2
                     training accuracy
                                       1
                     validation accuracy
         2.5
              5.0
                  7.5 10.0 12.5 15.0
                                          0.0
                                              2.5
                                                   5.0
                                                            10.0 12.5
                   epoch
Evaluando ...
Exactitud validate: 42.4 %, Test: 45.0 %
Tiempo de ejecución del experimento: 67.0
exp = Experimento_1b_2_4
modelo = 4
samples_per_class = 100
opti = rmsprop
batch = 128
epochs = 15
run = True
Entrenando ...
Tiempo de entrenamiento: ** 42.0 **
                                                      Loss
  0.6
                                     3
  0.4
                                       2
                     training accuracy
  0.2
                      validation accuracy
                      10.0 12.5 15.0
                                                                 12.5
         2.5
              50
                  75
                                         0.0
                                              2.5
                                                   5.0
                                                        75
                                                            10.0
```

```
Evaluando ...
Exactitud validate: 53.54 %, Test: 55.42 %
Tiempo de ejecución del experimento: 43.0
Tiempo de ejecución todo el batch: 261.0
```

▼ Resultados Lote 1a

15

Experimento	Muestras*Clase	Optimizador	Modelo	Batch Size	epocas	Tiempo Entrenamiento	Exac. val.	Exact. test
Experimento_1b_1_3	100	Adam	3	128	15	66	40.00%	43.33%
Experimento_1b_1_4	100	Adam	4	128	15	40	42.40%	43.44%
Experimento_1b_2_3	100	rmsprop	3	128	15	66	42.40%	45.00%
Experimento_1b_2_4	100	rmsprop	4	128	15	42	53.54%	55.42%

Si comparamos con los resultados anteriores podemos observar que el **batch_size=32** funciona mejor en la mayoria de los casos excepto en el **Experimento_1b_2_4**.

Experimento	Muestras*Clase	Optimizador	Modelo	Batch Size	epocas	Tiempo Entrenamiento	Exac. val.	Exact. test
Experimento_1_3	100	Adam	3	32	15	73	44.38%	48.44%
Experimento_1_4	100	Adam	4	32	15	44	50.10%	52.08%
Experimento_2_3	100	rmsprop	3	32	15	71	47.71%	53.44%
Experimento_2_4	100	rmsprop	4	32	15	43	48.12%	52.60%

```
1 \;\#\; definimos los experimentos del lote 1
 2 number_of_classes = 102
 3 experimentos = [
                  {"exp":"Experimento_1b_2_4","modelo":4,"samples_per_class":100,"opti":"rmsprop","batch":2
 5
7 start time = timeit.default timer()
 8 hacer_experimento_gec(experimentos)
9 print("Tiempo de ejecución todo el batch:",round(timeit.default_timer() - start_time,0))
   ______
    exp = Experimento_1b_2_4
    modelo = 4
    samples per class = 100
    opti = rmsprop
    batch = 256
    epochs = 15
    run = True
    Creando juego de datos ...
    Entrenando ...
    Tiempo de entrenamiento: ** 40.0 **
                   Accuracy
                                                     Loss
      0.6
                                        4
     accuracy
                                      850
      0.2
                       training accuracy
                    - validation accuracy
                     7.5
                        10.0 12.5 15.0
                                                               12.5
            2.5
                 5.0
                                          0.0
                                              2.5
                                                   5.0
                                                           10.0
                     epoch
    Evaluando ...
    Exactitud validate: 42.29 %, Test: 45.42 %
    Tiempo de ejecución del experimento: 41.0
    Tiempo de ejecución todo el batch: 76.0
1 model = models.Sequential()
2 print("Input dimensions: ",X_train.shape[1:])
4 #model.add(layers.Conv2D(32, (3, 3), input_shape=X_train.shape[1:]))
 5 model.add(layers.Conv2D(32, (3, 3),use_bias=False, input_shape=X_train.shape[1:]))
 6 model.add(layers.BatchNormalization()) #######
 7 model.add(layers.Activation('relu'))
 8 model.add(layers.MaxPooling2D(pool_size=(2, 2)))
10 #model.add(layers.Conv2D(32, (3, 3)))
11 model.add(layers.Conv2D(32, (3, 3),use_bias=False))
12 model.add(layers.BatchNormalization()) #######
13 model.add(layers.Activation('relu'))
14 model.add(layers.MaxPooling2D(pool_size=(2, 2)))
```