

# Sistemas inteligentes

## Práctica 2

30/11/2025

---

David Espinosa Gómez

Grupo 1

<b>Introducción</b>	<b>3</b>
<b>1. Mlp1</b>	<b>3</b>
<b>2. MLP2</b>	<b>5</b>
<b>3. MLP3</b>	<b>7</b>

# Introducción

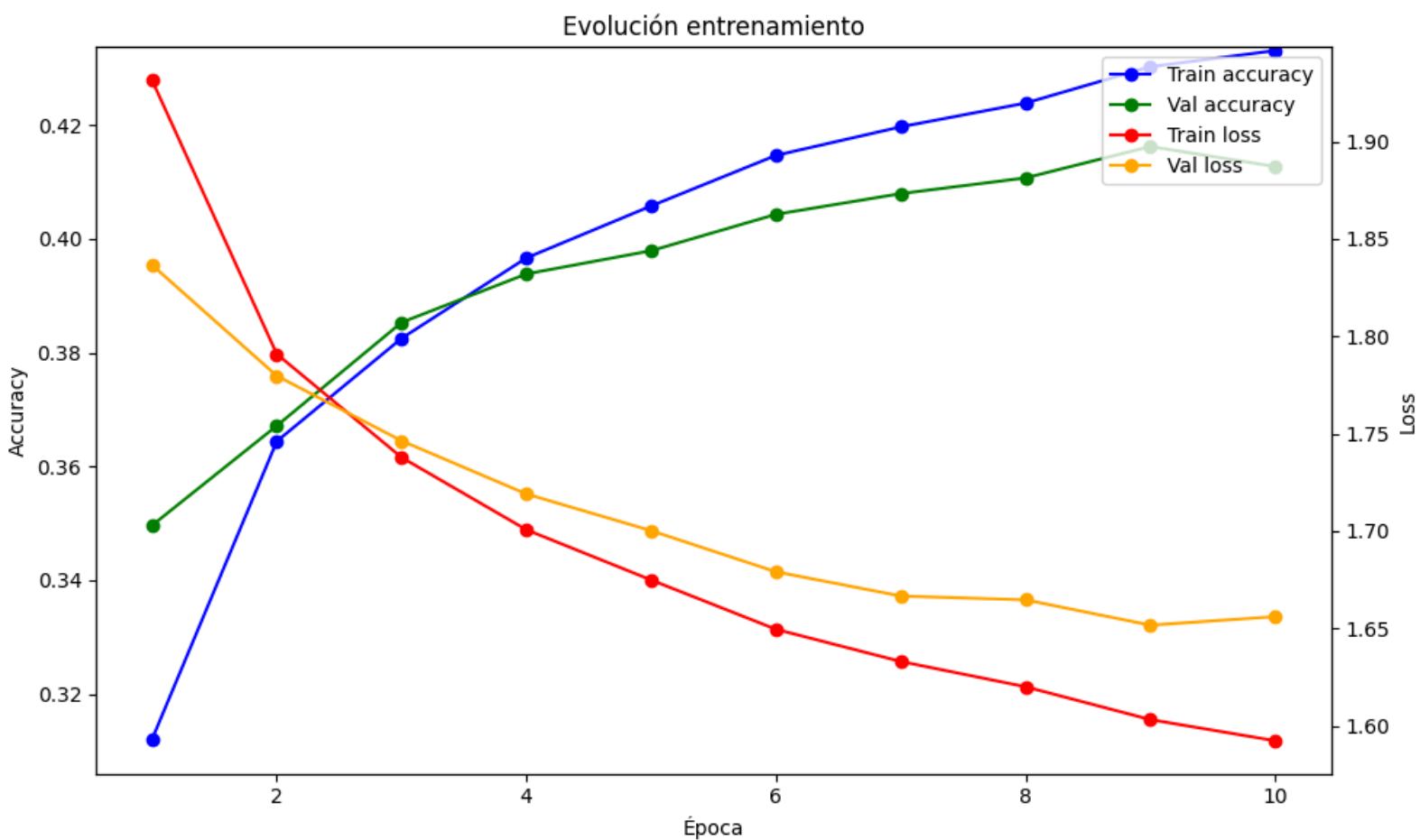
Para todos los mlps y modelos probados han sido compilados y entrenados 5 veces, siendo 5 modelos distintos con los mismos parámetros. Para el cálculo de todas las métricas se ha promediado entre los 5 entrenamientos, mientras que para la matriz de confusión se ha graficado la que corresponde al mejor resultado de los 5 entrenamientos (en términos de test accuracy).

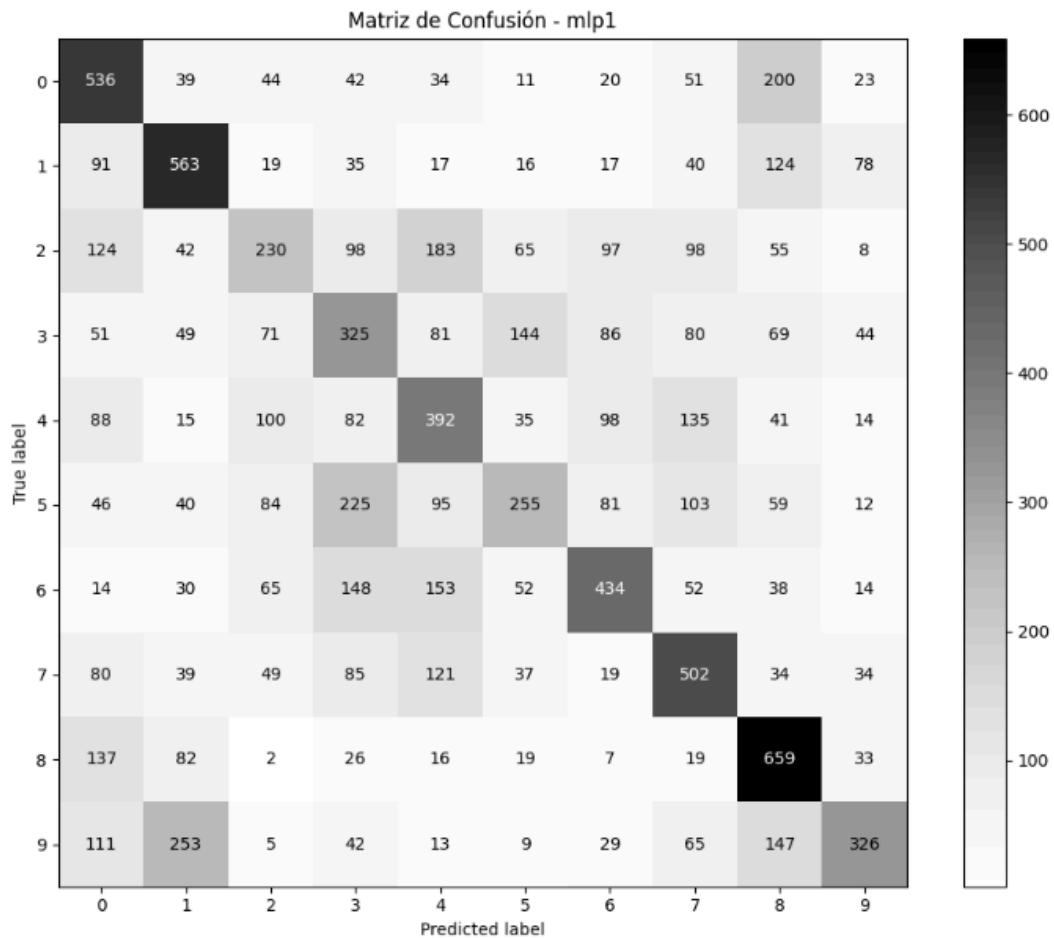
## 1.Mlp1

Implementación simple de acuerdo a la práctica.

Resultado obtenido:

```
Resultados finales de mlp1
Tiempo de entrenamiento (media): 11.84s
Test Accuracy (media): 0.4130
Test Loss (media): 1.6494
Val Accuracy final (media): 0.4127
Val Loss final (media): 1.6560
```





## 2.MLP2

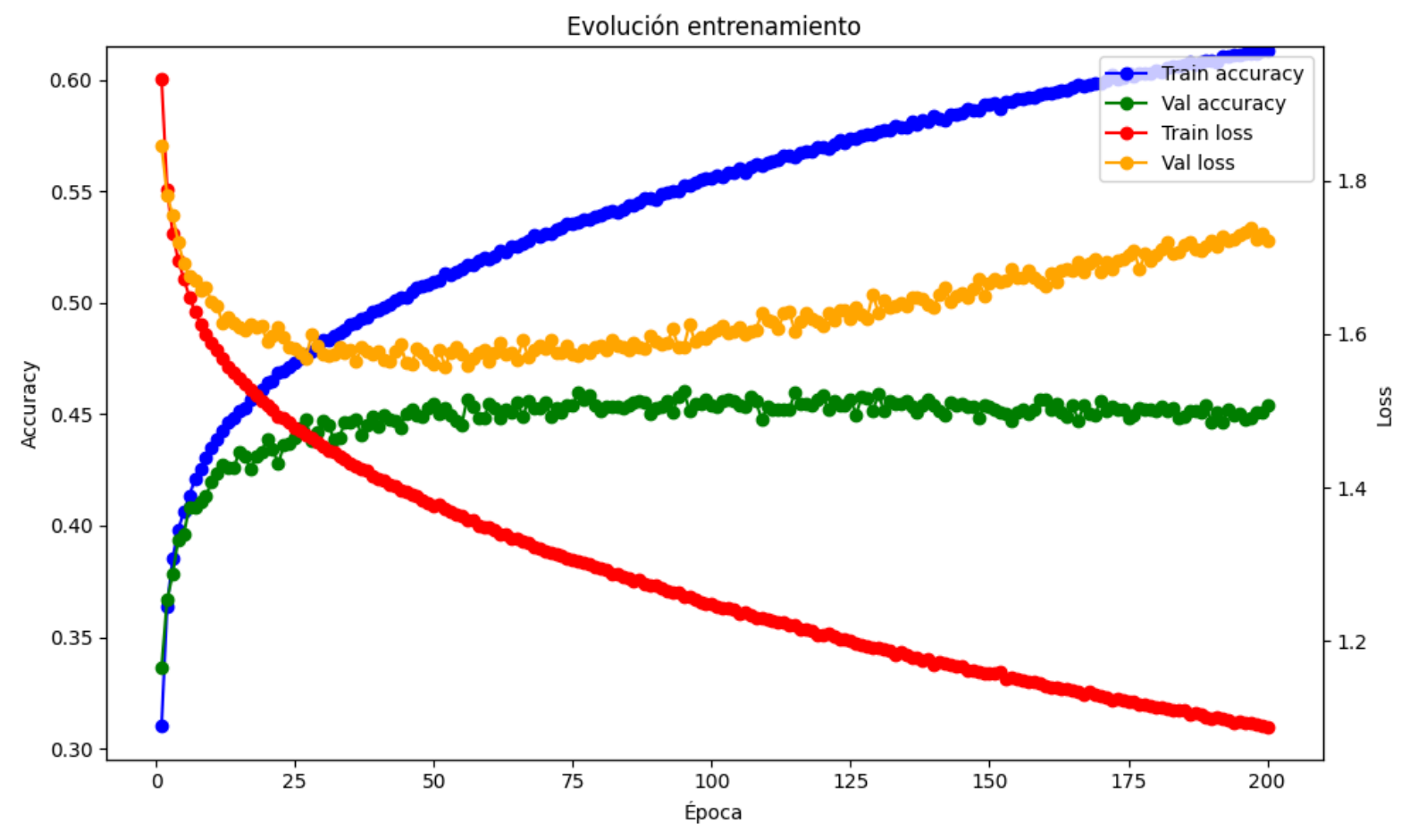
El mlp2 consiste en probar diferentes configuraciones de earlystopping para ver cual es el mejor número de épocas, teniendo en cuenta cuando comienza a pasarse el tiempo. El mlp2 como se observa es exactamente igual al mlp1 en la composición de las capas y función de activación, solamente he aumentado el número de épocas para intentar observar donde sucede el sobreentrenamiento. Para los callbacks he probado monitorizando tanto el validation loss como el validation accuracy, probando callbacks más agresivos (menos patience y un delta más alto) y callbacks menos agresivos (más patience y menos delta), siendo estos últimos los que permiten obtener una mayor accuracy pero tardando más también.

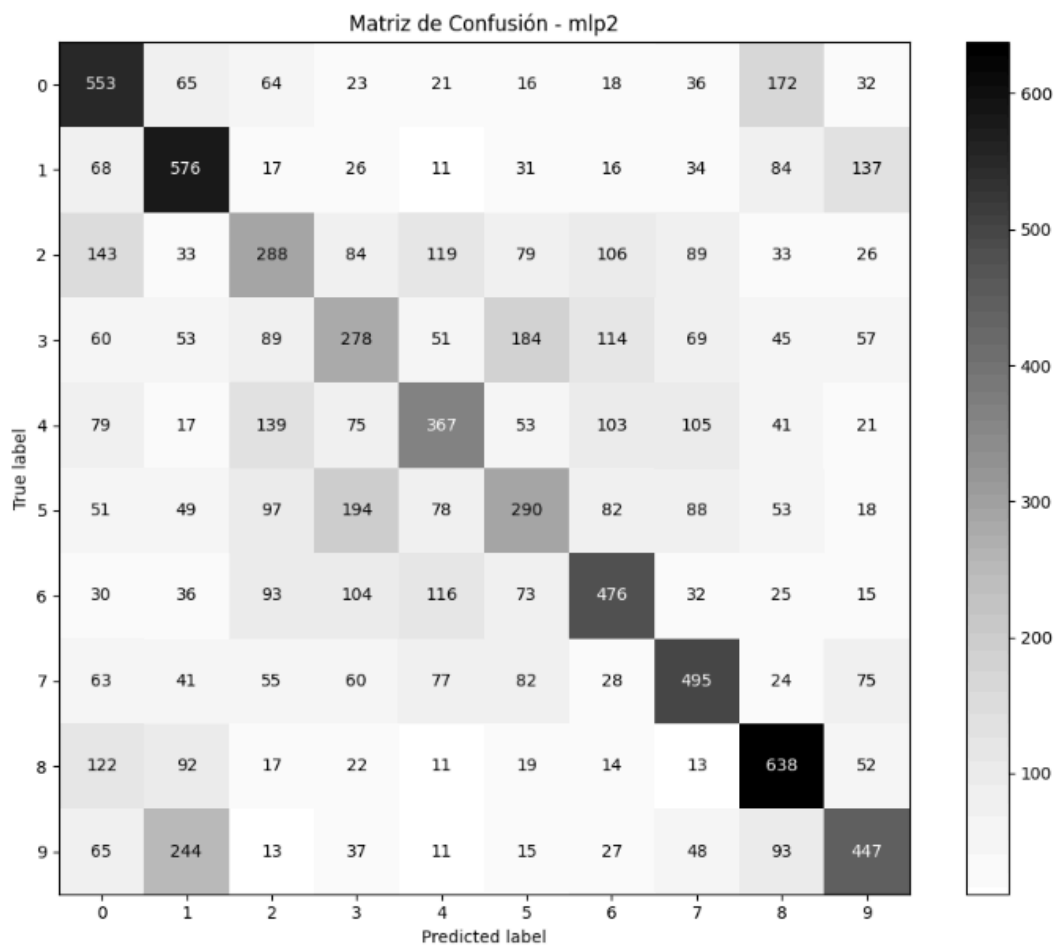
```

),
MLPConfig( #mlp 2, el callback para de media ahí
    nombre="mlp2",
    capas=[48],
    activation="sigmoid",
    epochs=50,
    batch_size=32,
    verbose=0
)

```

Primero voy a hacer una ejecución sin early\_stopping con 200 épocas y mostrar la gráfica de entrenamiento y la matriz de confusión.

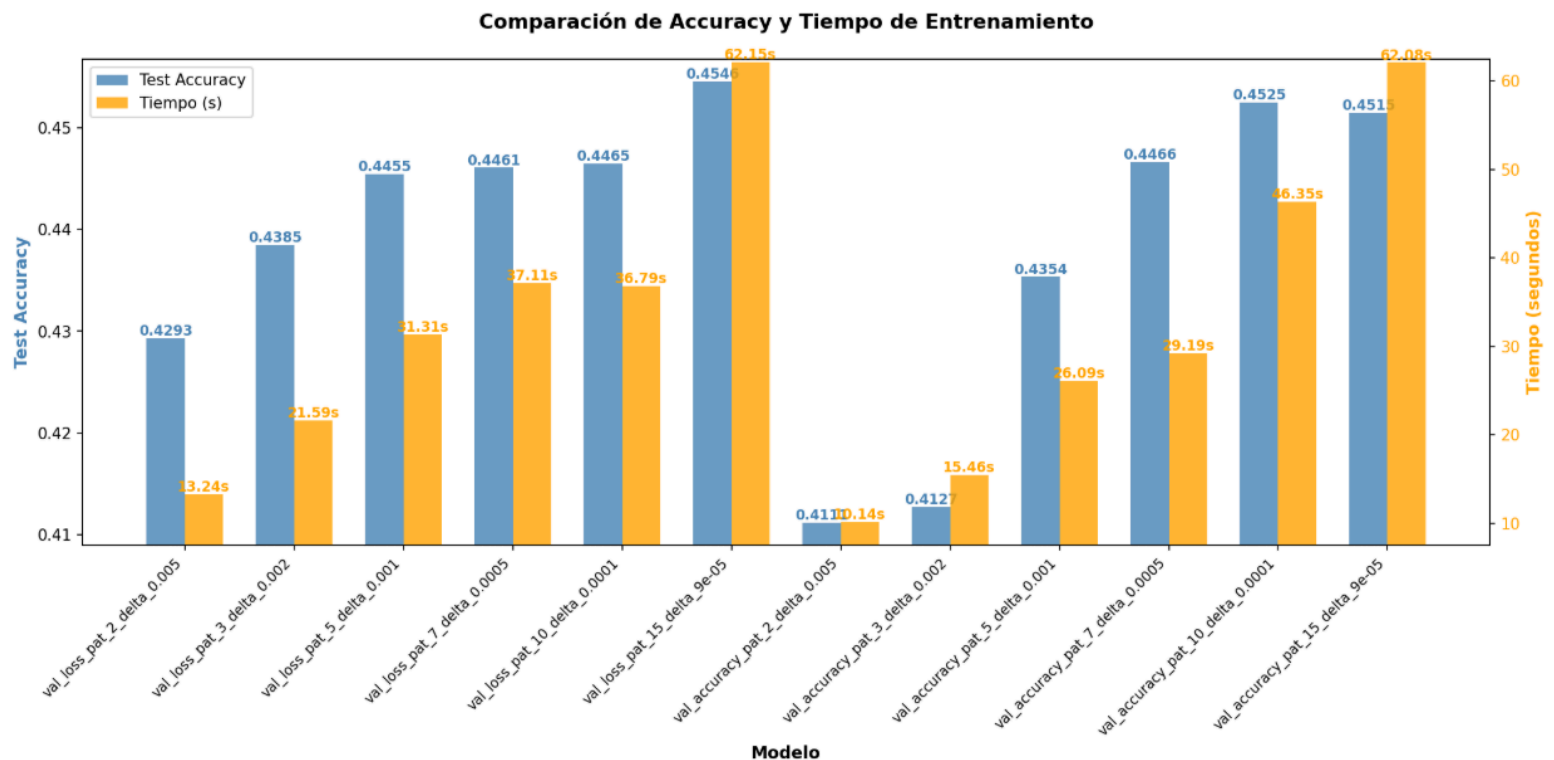




Los early\_stopping a probar son los siguientes:

```
early_stopping_configs = [ #earlystoppings a probar, grafica ya generada
    EarlyStoppingConfig(monitor='val_loss', patience=2, min_delta=0.005, verbose=0),
    EarlyStoppingConfig(monitor='val_loss', patience=3, min_delta=0.002, verbose=0),
    EarlyStoppingConfig(monitor='val_loss', patience=5, min_delta=0.001, verbose=0), #para mi este es el mejor
    EarlyStoppingConfig(monitor='val_loss', patience=7, min_delta=0.0005, verbose=0),
    EarlyStoppingConfig(monitor='val_loss', patience=10, min_delta=0.0001, verbose=0),
    EarlyStoppingConfig(monitor='val_loss', patience=15, min_delta=0.00009, verbose=0),
    EarlyStoppingConfig(monitor='val_accuracy', patience=2, min_delta=0.005, verbose=0),
    EarlyStoppingConfig(monitor='val_accuracy', patience=3, min_delta=0.002, verbose=0),
    EarlyStoppingConfig(monitor='val_accuracy', patience=5, min_delta=0.001, verbose=0), #para mi este es el mejor
    EarlyStoppingConfig(monitor='val_accuracy', patience=7, min_delta=0.0005, verbose=0),
    EarlyStoppingConfig(monitor='val_accuracy', patience=10, min_delta=0.0001, verbose=0),
    EarlyStoppingConfig(monitor='val_accuracy', patience=15, min_delta=0.00009, verbose=0)
```

El resultado obtenido ha sido:



Tras este resultado he decidido que el mejor early-stopping teniendo en cuenta la relación test accuracy y tiempo ha sido el callback que monitoriza el validation accuracy con un patience de 7, puesto que da un accuracy de un 44,66% tardando 29 segundos mientras que la máxima accuracy ha sido 45,46% tardando 62 segundos, por lo tanto considerando el aumento de precisión y tiempo creo que este callback es el mejor.

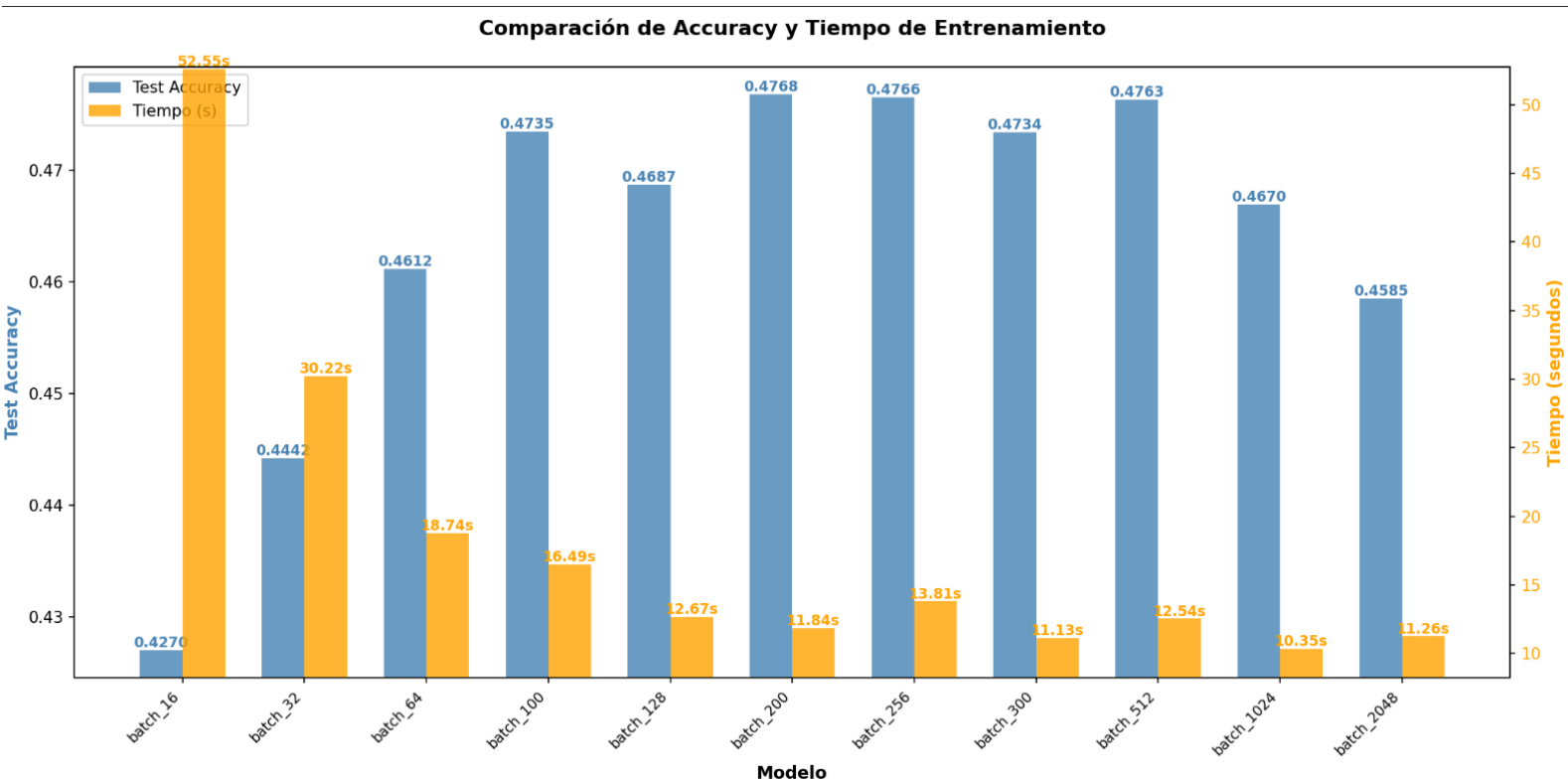
### 3. MLP3

Los batch sizes a probar son:

```
batch_sizes = [16,32,64,100,128,200,256,300,512,1024,2048]
```

Para realizar la prueba he usado 200 épocas y el callback obtenido en el mlp anterior.

Resultados obtenidos:



Como se observa en la imagen el mejor test accuracy ha sido con un batch size 200, además de haber tenido un tiempo de entrenamiento bajo. Por lo tanto he escogido 200 de batch size como el mejor resultado.

Gráficas de 200 de batch size:

