

Participantes:

Andres Felipe Rivas Luna

Daniel Felipe Galeano Tabares

Jorge Andrés Salas Barrera

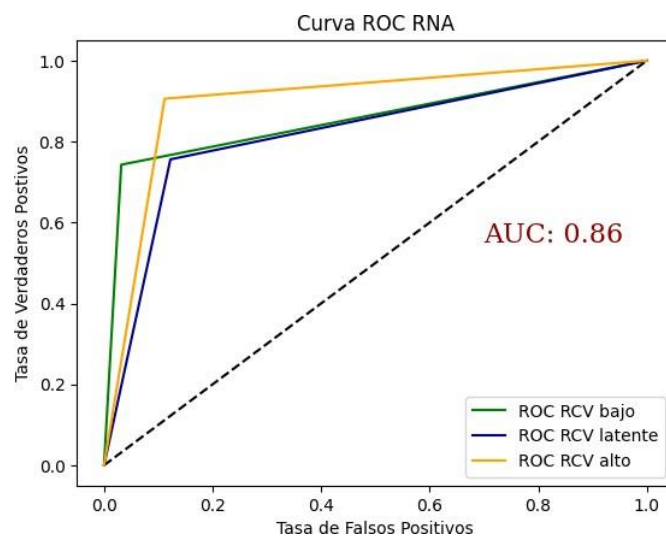
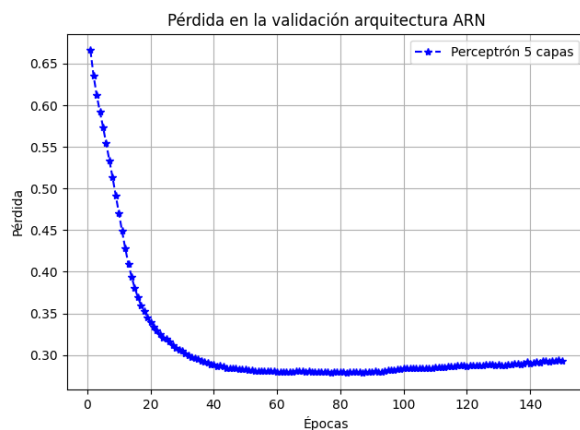
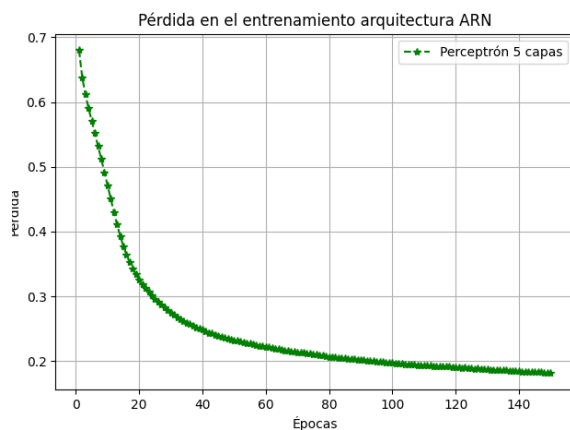
Introducción.

En el presente documento, se llevará a cabo una experimentación que involucra el uso de modelos de redes neuronales aplicados a datos relacionados con el riesgo cardiovascular. El objetivo primordial de este estudio es optimizar la arquitectura de la red neuronal de manera que el área bajo la curva (AUC) alcance el mayor valor posible.

Prueba 1: Inicialmente se crea una capa de entrada con 8 neuronas y 35 características de entrada, una capa oculta con 6 neuronas y por último la salida de la red, que tiene 3 neuronas y utiliza la función de activación softmax.

```
modelRNA = models.Sequential()
modelRNA.add(Dense(8, input_shape=(None, 35), activation='relu'))
modelRNA.add(Dense(6, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```

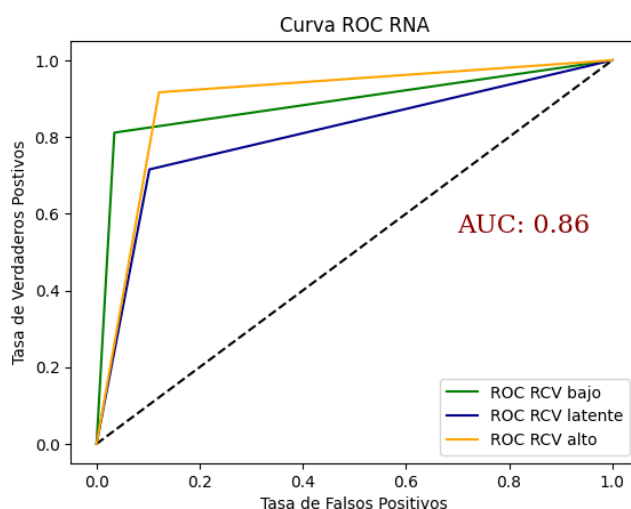
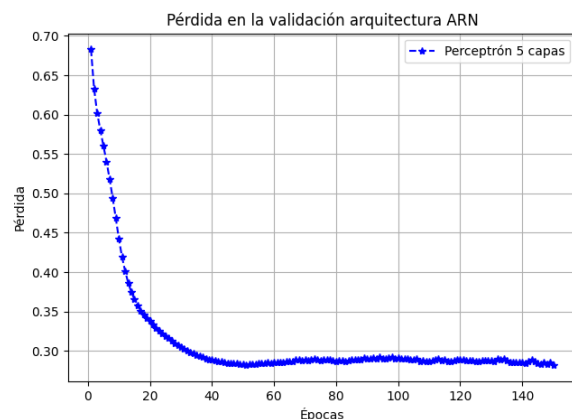
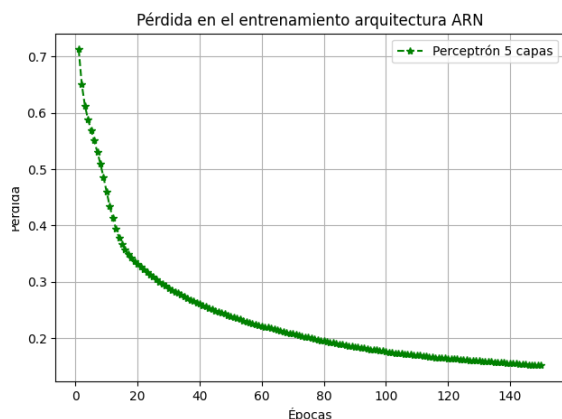
```
Epoch 140/150
53/53 [=====] - 0s 4ms/step - loss: 0.1840 - accuracy: 0.8976 - val_loss: 0.2901 - val_accuracy: 0.8595
Epoch 141/150
53/53 [=====] - 0s 4ms/step - loss: 0.1836 - accuracy: 0.8982 - val_loss: 0.2910 - val_accuracy: 0.8524
Epoch 142/150
53/53 [=====] - 0s 4ms/step - loss: 0.1835 - accuracy: 0.8946 - val_loss: 0.2914 - val_accuracy: 0.8595
Epoch 143/150
53/53 [=====] - 0s 4ms/step - loss: 0.1831 - accuracy: 0.8982 - val_loss: 0.2913 - val_accuracy: 0.8571
Epoch 144/150
53/53 [=====] - 0s 4ms/step - loss: 0.1827 - accuracy: 0.8964 - val_loss: 0.2924 - val_accuracy: 0.8571
Epoch 145/150
53/53 [=====] - 0s 4ms/step - loss: 0.1827 - accuracy: 0.9000 - val_loss: 0.2928 - val_accuracy: 0.8571
Epoch 146/150
53/53 [=====] - 0s 4ms/step - loss: 0.1823 - accuracy: 0.8982 - val_loss: 0.2919 - val_accuracy: 0.8595
Epoch 147/150
53/53 [=====] - 0s 5ms/step - loss: 0.1826 - accuracy: 0.8988 - val_loss: 0.2926 - val_accuracy: 0.8571
Epoch 148/150
53/53 [=====] - 0s 4ms/step - loss: 0.1821 - accuracy: 0.9006 - val_loss: 0.2935 - val_accuracy: 0.8595
Epoch 149/150
53/53 [=====] - 0s 4ms/step - loss: 0.1816 - accuracy: 0.8988 - val_loss: 0.2940 - val_accuracy: 0.8571
Epoch 150/150
53/53 [=====] - 0s 3ms/step - loss: 0.1813 - accuracy: 0.8970 - val_loss: 0.2925 - val_accuracy: 0.8571
```



Prueba 2: Se mantuvo la capa inicial de 8 neuronas de red, en la capa oculta aumentamos un poco el número de neuronas a 8 y la capa de salida también se mantuvo en el mismo valor. Evidenciamos que el valor de **AUC** se mantuvo estable.

```
modelRNA = models.Sequential()
modelRNA.add(Dense(8, input_shape=(None, 35), activation='relu'))
modelRNA.add(Dense(8, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```

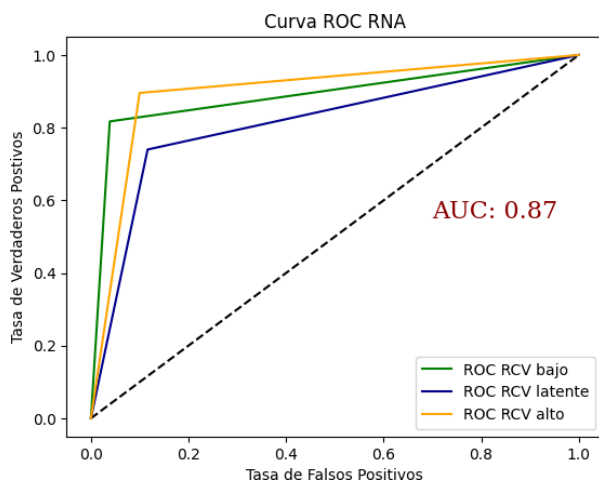
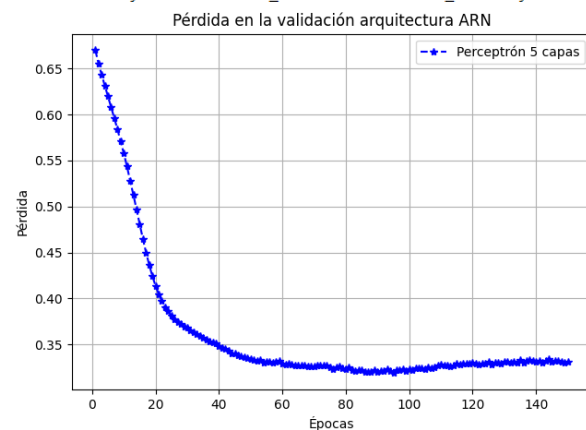
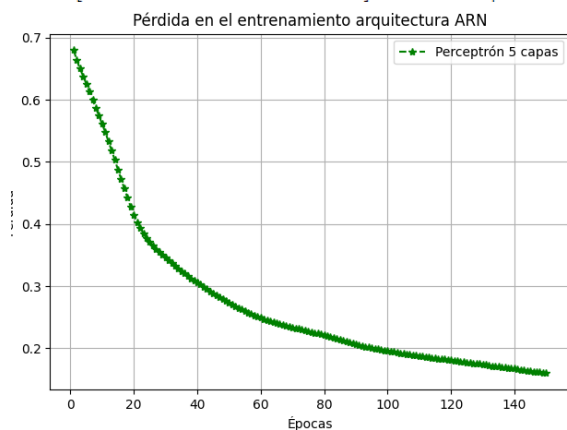
```
Epoch 140/150
53/53 [=====] - 0s 4ms/step - loss: 0.1552 - accuracy: 0.9232 - val_loss: 0.2852 - val_accuracy: 0.8548
Epoch 141/150
53/53 [=====] - 0s 4ms/step - loss: 0.1551 - accuracy: 0.9262 - val_loss: 0.2845 - val_accuracy: 0.8524
Epoch 142/150
53/53 [=====] - 0s 4ms/step - loss: 0.1552 - accuracy: 0.9244 - val_loss: 0.2865 - val_accuracy: 0.8524
Epoch 143/150
53/53 [=====] - 0s 2ms/step - loss: 0.1537 - accuracy: 0.9256 - val_loss: 0.2888 - val_accuracy: 0.8524
Epoch 144/150
53/53 [=====] - 0s 2ms/step - loss: 0.1542 - accuracy: 0.9268 - val_loss: 0.2867 - val_accuracy: 0.8524
Epoch 145/150
53/53 [=====] - 0s 2ms/step - loss: 0.1534 - accuracy: 0.9244 - val_loss: 0.2841 - val_accuracy: 0.8548
Epoch 146/150
53/53 [=====] - 0s 3ms/step - loss: 0.1527 - accuracy: 0.9250 - val_loss: 0.2834 - val_accuracy: 0.8548
Epoch 147/150
53/53 [=====] - 0s 2ms/step - loss: 0.1523 - accuracy: 0.9244 - val_loss: 0.2852 - val_accuracy: 0.8548
Epoch 148/150
53/53 [=====] - 0s 2ms/step - loss: 0.1523 - accuracy: 0.9286 - val_loss: 0.2835 - val_accuracy: 0.8548
Epoch 149/150
53/53 [=====] - 0s 2ms/step - loss: 0.1517 - accuracy: 0.9304 - val_loss: 0.2854 - val_accuracy: 0.8548
Epoch 150/150
53/53 [=====] - 0s 2ms/step - loss: 0.1520 - accuracy: 0.9298 - val_loss: 0.2817 - val_accuracy: 0.8571
```



Prueba 3: Agregamos una capa oculta adicional con 3 neuronas y pudimos ver un leve aumento en el valor de **AUC**.

```
modelRNA = models.Sequential()
modelRNA.add(Dense(8, input_shape=(None, 35), activation='relu'))
modelRNA.add(Dense(8, activation='relu'))
modelRNA.add(Dense(3, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```

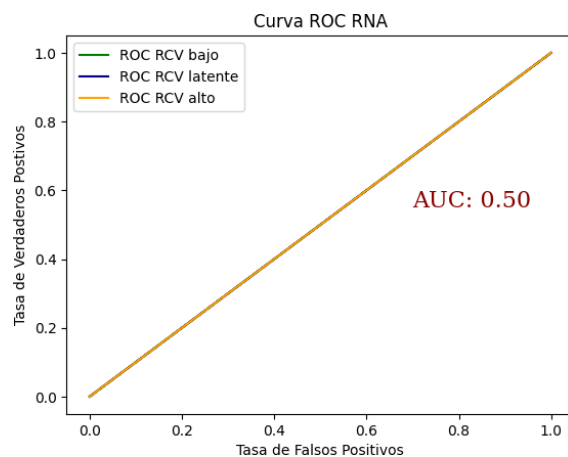
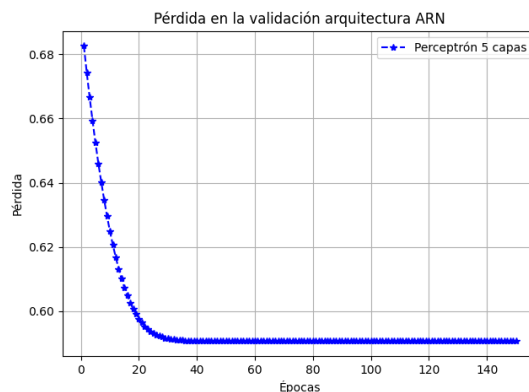
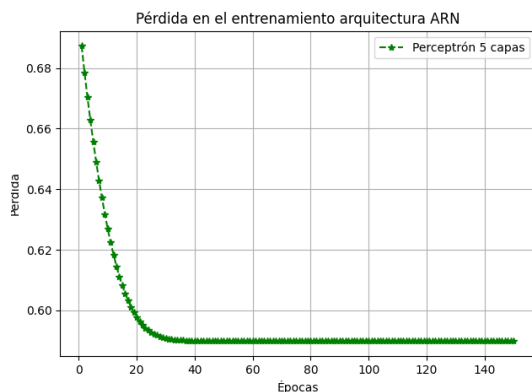
```
Epoch 140/150
53/53 [=====] - 0s 3ms/step - loss: 0.1672 - accuracy: 0.9143 - val_loss: 0.3307 - val_accuracy: 0.8381
Epoch 141/150
53/53 [=====] - 0s 2ms/step - loss: 0.1663 - accuracy: 0.9167 - val_loss: 0.3326 - val_accuracy: 0.8405
Epoch 142/150
53/53 [=====] - 0s 3ms/step - loss: 0.1654 - accuracy: 0.9167 - val_loss: 0.3297 - val_accuracy: 0.8452
Epoch 143/150
53/53 [=====] - 0s 3ms/step - loss: 0.1649 - accuracy: 0.9167 - val_loss: 0.3318 - val_accuracy: 0.8381
Epoch 144/150
53/53 [=====] - 0s 3ms/step - loss: 0.1638 - accuracy: 0.9167 - val_loss: 0.3338 - val_accuracy: 0.8405
Epoch 145/150
53/53 [=====] - 0s 3ms/step - loss: 0.1632 - accuracy: 0.9196 - val_loss: 0.3312 - val_accuracy: 0.8476
Epoch 146/150
53/53 [=====] - 0s 3ms/step - loss: 0.1625 - accuracy: 0.9173 - val_loss: 0.3326 - val_accuracy: 0.8476
Epoch 147/150
53/53 [=====] - 0s 3ms/step - loss: 0.1620 - accuracy: 0.9190 - val_loss: 0.3320 - val_accuracy: 0.8429
Epoch 148/150
53/53 [=====] - 0s 3ms/step - loss: 0.1614 - accuracy: 0.9208 - val_loss: 0.3308 - val_accuracy: 0.8405
Epoch 149/150
53/53 [=====] - 0s 3ms/step - loss: 0.1601 - accuracy: 0.9173 - val_loss: 0.3304 - val_accuracy: 0.8452
Epoch 150/150
53/53 [=====] - 0s 2ms/step - loss: 0.1601 - accuracy: 0.9196 - val_loss: 0.3314 - val_accuracy: 0.8381
```



Prueba 4: Se agregó una capa oculta adicional con 2 neuronas y el resultado fue negativo, en las iteraciones vimos valores del loss y el accuracy en un rango intermedio de 0.55, y el valor de la variable **AUC** bajo significativamente. Debido a este resultado en las siguientes pruebas se mantendrán sólo dos capas ocultas en la arquitectura y se modificarán sólo el número de neuronas.

```
modelRNA = models.Sequential()
modelRNA.add(Dense(8, input_shape=(None, 35), activation='relu'))
modelRNA.add(Dense(4, activation='relu'))
modelRNA.add(Dense(5, activation='relu'))
modelRNA.add(Dense(2, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```

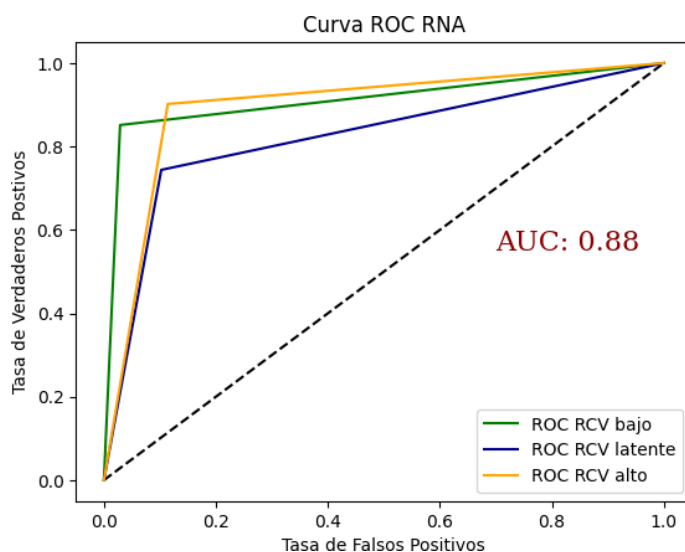
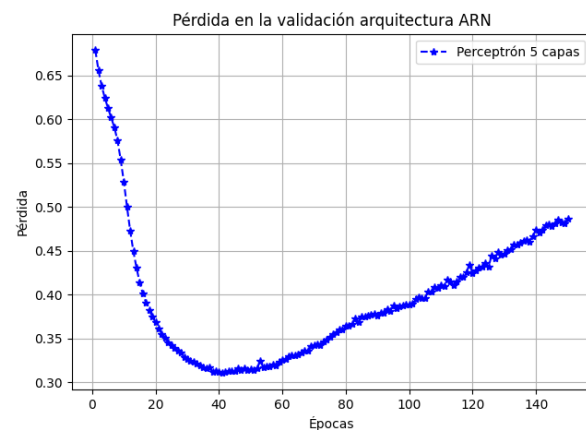
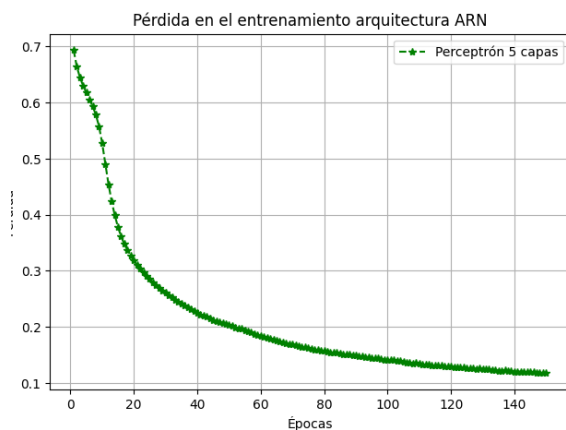
```
Epoch 140/150
53/53 [=====] - 0s 5ms/step - loss: 0.5899 - accuracy: 0.5327 - val_loss: 0.5907 - val_accuracy: 0.5310
Epoch 141/150
53/53 [=====] - 0s 4ms/step - loss: 0.5899 - accuracy: 0.5327 - val_loss: 0.5907 - val_accuracy: 0.5310
Epoch 142/150
53/53 [=====] - 0s 5ms/step - loss: 0.5899 - accuracy: 0.5327 - val_loss: 0.5907 - val_accuracy: 0.5310
Epoch 143/150
53/53 [=====] - 0s 4ms/step - loss: 0.5899 - accuracy: 0.5327 - val_loss: 0.5907 - val_accuracy: 0.5310
Epoch 144/150
53/53 [=====] - 0s 4ms/step - loss: 0.5899 - accuracy: 0.5327 - val_loss: 0.5907 - val_accuracy: 0.5310
Epoch 145/150
53/53 [=====] - 0s 4ms/step - loss: 0.5899 - accuracy: 0.5327 - val_loss: 0.5907 - val_accuracy: 0.5310
Epoch 146/150
53/53 [=====] - 0s 4ms/step - loss: 0.5899 - accuracy: 0.5327 - val_loss: 0.5907 - val_accuracy: 0.5310
Epoch 147/150
53/53 [=====] - 0s 5ms/step - loss: 0.5899 - accuracy: 0.5327 - val_loss: 0.5907 - val_accuracy: 0.5310
Epoch 148/150
53/53 [=====] - 0s 4ms/step - loss: 0.5899 - accuracy: 0.5327 - val_loss: 0.5907 - val_accuracy: 0.5310
Epoch 149/150
53/53 [=====] - 0s 5ms/step - loss: 0.5899 - accuracy: 0.5327 - val_loss: 0.5907 - val_accuracy: 0.5310
Epoch 150/150
53/53 [=====] - 0s 4ms/step - loss: 0.5899 - accuracy: 0.5327 - val_loss: 0.5907 - val_accuracy: 0.5310
```



Prueba 5: Se modificó la arquitectura de la prueba anterior dejando en la primera capa oculta un total de 4 neuronas y en la segunda capa 5 neuronas. Evidenciamos un aumento significativo en el **AUC**.

```
modelRNA = models.Sequential()
modelRNA.add(Dense(8, input_shape=(None, 35), activation='relu'))
modelRNA.add(Dense(4, activation='relu'))
modelRNA.add(Dense(5, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```

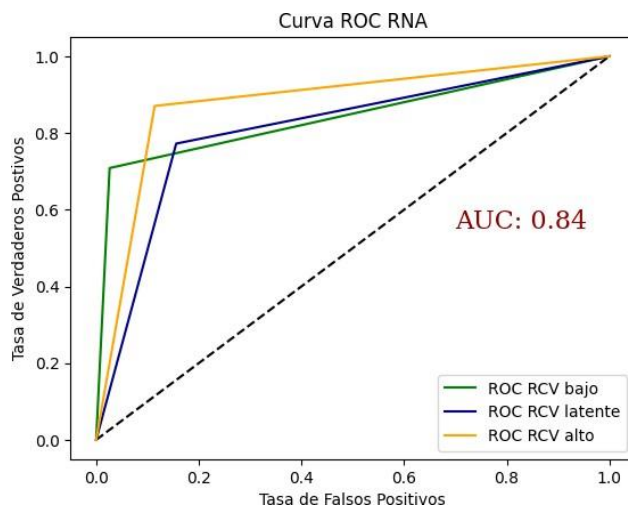
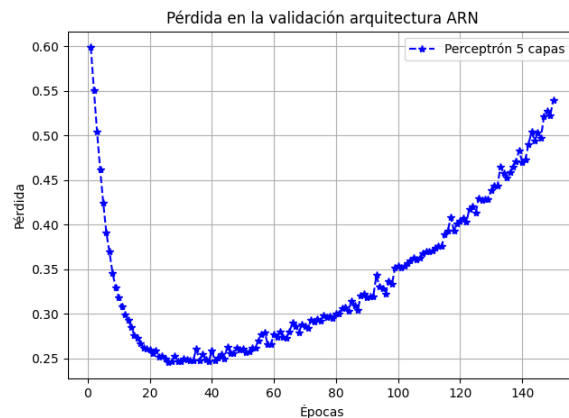
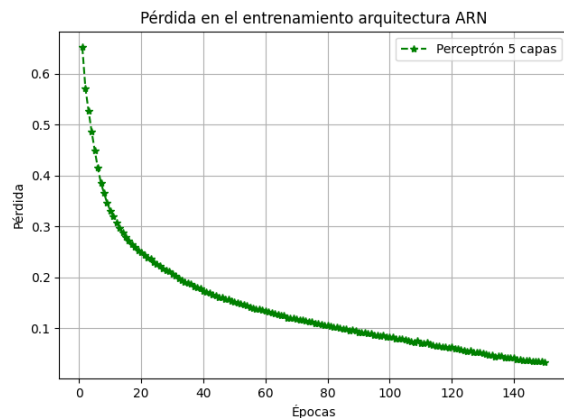
```
Epoch 140/150
53/53 [=====] - 0s 4ms/step - loss: 0.1206 - accuracy: 0.9399 - val_loss: 0.4742 - val_accuracy: 0.8405
Epoch 141/150
53/53 [=====] - 0s 5ms/step - loss: 0.1207 - accuracy: 0.9405 - val_loss: 0.4707 - val_accuracy: 0.8405
Epoch 142/150
53/53 [=====] - 0s 5ms/step - loss: 0.1205 - accuracy: 0.9411 - val_loss: 0.4737 - val_accuracy: 0.8476
Epoch 143/150
53/53 [=====] - 0s 6ms/step - loss: 0.1199 - accuracy: 0.9393 - val_loss: 0.4790 - val_accuracy: 0.8429
Epoch 144/150
53/53 [=====] - 0s 5ms/step - loss: 0.1204 - accuracy: 0.9417 - val_loss: 0.4803 - val_accuracy: 0.8452
Epoch 145/150
53/53 [=====] - 0s 5ms/step - loss: 0.1191 - accuracy: 0.9411 - val_loss: 0.4778 - val_accuracy: 0.8429
Epoch 146/150
53/53 [=====] - 0s 5ms/step - loss: 0.1202 - accuracy: 0.9417 - val_loss: 0.4816 - val_accuracy: 0.8476
Epoch 147/150
53/53 [=====] - 0s 5ms/step - loss: 0.1189 - accuracy: 0.9411 - val_loss: 0.4855 - val_accuracy: 0.8429
Epoch 148/150
53/53 [=====] - 0s 5ms/step - loss: 0.1189 - accuracy: 0.9387 - val_loss: 0.4821 - val_accuracy: 0.8405
Epoch 149/150
53/53 [=====] - 0s 5ms/step - loss: 0.1179 - accuracy: 0.9429 - val_loss: 0.4813 - val_accuracy: 0.8476
Epoch 150/150
53/53 [=====] - 0s 5ms/step - loss: 0.1182 - accuracy: 0.9417 - val_loss: 0.4866 - val_accuracy: 0.8524
```



Prueba 6: Se aumentó el número de neuronas de cada capa, aunque evidenciamos que el **loss** es más cercano a 0 y el **accuracy** se acercó más a 1, vimos una disminución en el valor de **AUC**.

```
modelRNA = models.Sequential()
modelRNA.add(Dense(16, input_shape=(None, 35), activation='relu'))
modelRNA.add(Dense(32, activation='relu'))
modelRNA.add(Dense(16, activation='relu'))
modelRNA.add(Dense(3, activation='softmax'))
```

```
Epoch 140/150
53/53 [=====] - 0s 4ms/step - loss: 0.0425 - accuracy: 0.9869 - val_loss: 0.4697 - val_accuracy: 0.8405
Epoch 141/150
53/53 [=====] - 0s 3ms/step - loss: 0.0392 - accuracy: 0.9851 - val_loss: 0.4732 - val_accuracy: 0.8381
Epoch 142/150
53/53 [=====] - 0s 4ms/step - loss: 0.0387 - accuracy: 0.9869 - val_loss: 0.4902 - val_accuracy: 0.8381
Epoch 143/150
53/53 [=====] - 0s 3ms/step - loss: 0.0373 - accuracy: 0.9875 - val_loss: 0.5042 - val_accuracy: 0.8333
Epoch 144/150
53/53 [=====] - 0s 4ms/step - loss: 0.0378 - accuracy: 0.9887 - val_loss: 0.4938 - val_accuracy: 0.8429
Epoch 145/150
53/53 [=====] - 0s 4ms/step - loss: 0.0369 - accuracy: 0.9875 - val_loss: 0.5030 - val_accuracy: 0.8357
Epoch 146/150
53/53 [=====] - 0s 3ms/step - loss: 0.0356 - accuracy: 0.9887 - val_loss: 0.4974 - val_accuracy: 0.8405
Epoch 147/150
53/53 [=====] - 0s 3ms/step - loss: 0.0359 - accuracy: 0.9875 - val_loss: 0.5215 - val_accuracy: 0.8310
Epoch 148/150
53/53 [=====] - 0s 4ms/step - loss: 0.0349 - accuracy: 0.9893 - val_loss: 0.5279 - val_accuracy: 0.8381
Epoch 149/150
53/53 [=====] - 0s 3ms/step - loss: 0.0345 - accuracy: 0.9875 - val_loss: 0.5223 - val_accuracy: 0.8452
Epoch 150/150
53/53 [=====] - 0s 3ms/step - loss: 0.0332 - accuracy: 0.9893 - val_loss: 0.5398 - val_accuracy: 0.8381
```



CONCLUSIONES

Tras la realización de múltiples pruebas, se ha observado que, a pesar de realizar numerosas modificaciones en la arquitectura de la red neuronal, no se logró aumentar el valor del **AUC** por encima de **0,88 (prueba 5)**. Aunque incrementar el número de neuronas o capas ocultas condujo a mejoras notables en las métricas de "**loss**" y "**accuracy**", el **AUC** no mostró mejoras significativas y, en algunos casos, disminuyó o se mantuvo constante. Tras completar el proceso experimental, se pudo concluir que es posible que al particionar los datos de manera diferente y/o al emplear una función de activación más adecuada, esto logre aumentar el valor del **AUC**.