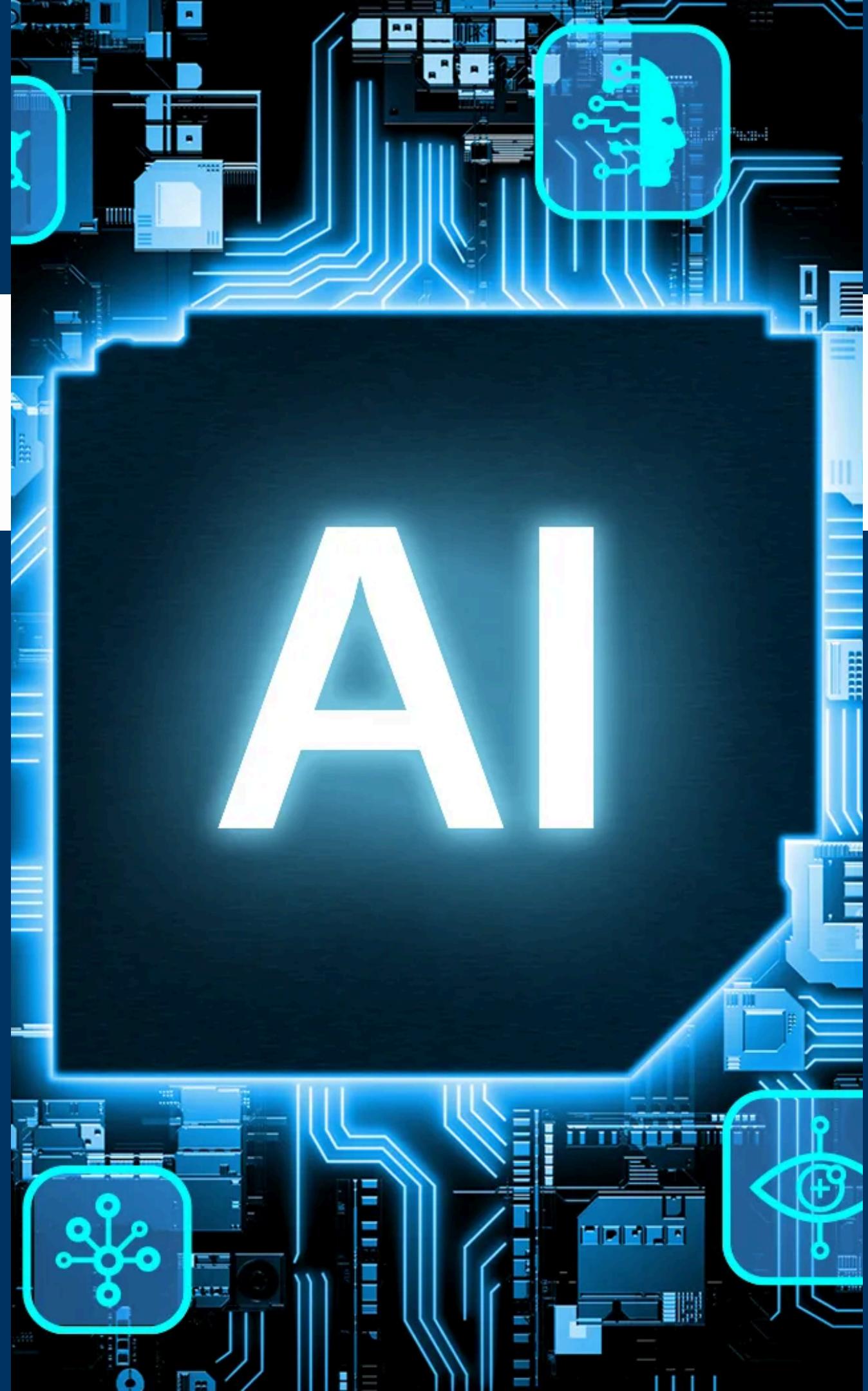


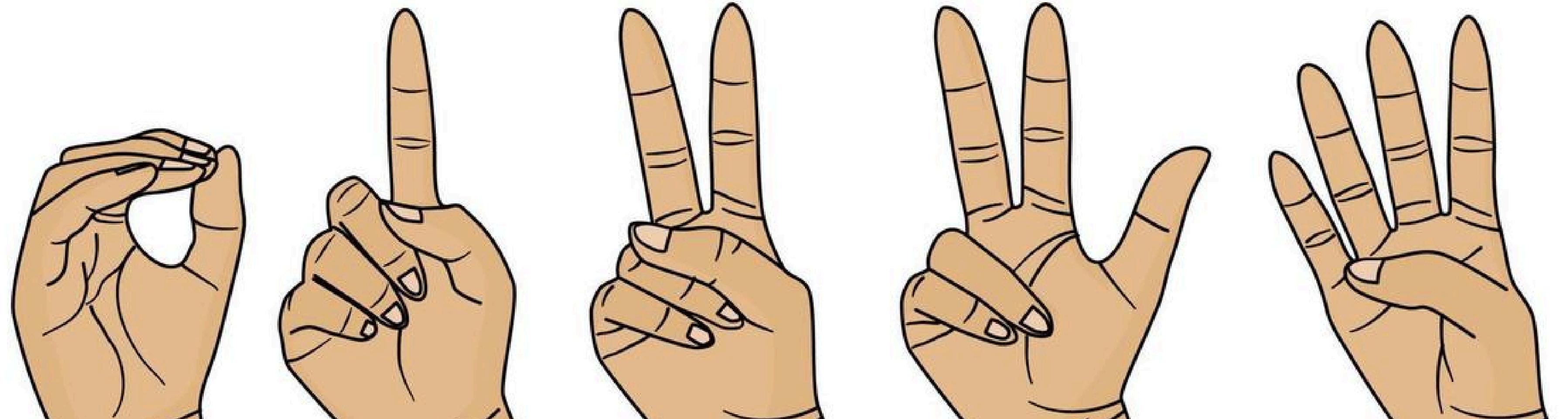
# RECONOCIMIENTO DE NÚMEROS EN EL LENGUAJE DE SEÑAS AMERICANO

Joan Sebastian Pedraza Nova - 2210100  
Juan Diego García Hernandez - 2210096



# INTRODUCCIÓN

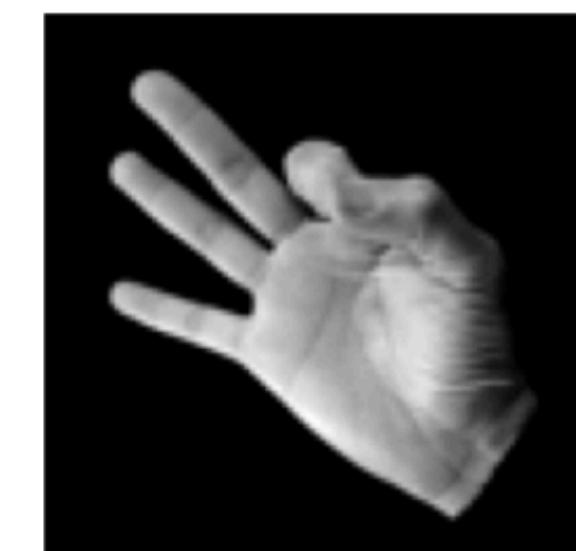
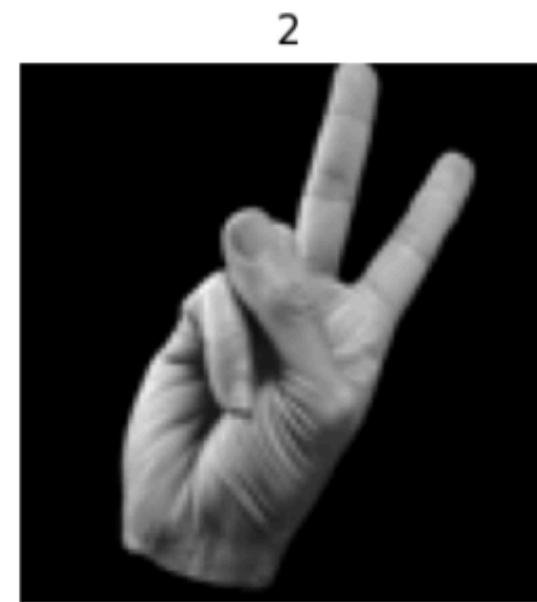
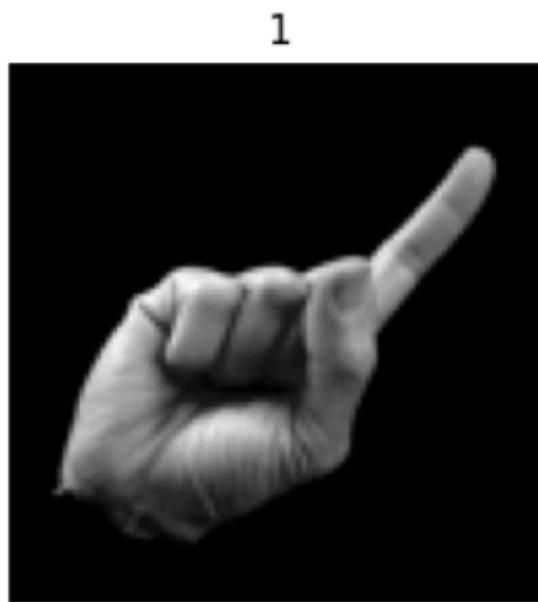
En este proyecto, utilizamos un conjunto de datos compuesto por imágenes que representan los dígitos del 0 al 9 en lenguaje de señas, cada una con un tamaño original de 150x150 píxeles. Estas imágenes fueron cuidadosamente etiquetadas para asociar cada representación con su correspondiente dígito, permitiendo su uso en tareas de clasificación. El objetivo principal es desarrollar y entrenar un modelo capaz de reconocer de manera precisa el dígito representado en cada imagen, contribuyendo así al reconocimiento automatizado del lenguaje de señas.



# DATASET

El dataset se llama “American Sign Language Digit Dataset” y contiene imágenes con los gestos para los dígitos de 0 al 9 en el lenguaje de señas americano.

# DATASET





# ESTIMADORES

1

Decision Tree

2

Random  
Forest

3

Support Vector  
Machine

# RESULTADOS

```
[ ] from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import numpy as np

x = np.array(x)
y = np.array(y)

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=21)

est = DecisionTreeClassifier()
est.fit(x_train, y_train)

y_pred = est.predict(x_test)

acc = accuracy_score(y_pred, y_test)
print("Tasa de aciertos: ")
print(acc)
```

→ Tasa de aciertos:  
0.806

# RESULTADOS

```
[ ] from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import *

X = np.array(X)
y = np.array(y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=21)

est = RandomForestClassifier()
est.fit(X_train, y_train)
y_pred = est.predict(X_test)

acc = accuracy_score(y_pred, y_test)
print("Tasa de aciertos: ")
print(acc)
```

→ Tasa de aciertos:  
0.973

# RESULTADOS

```
[ ] from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
import numpy as np

X = np.array(X)
y = np.array(y)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=21)

svc = SVC()
svc.fit(X_train, y_train)

y_pred = svc.predict(X_test)

acc = accuracy_score(y_pred, y_test)
print("Tasa de aciertos: ")
print(acc)
```

→ Tasa de aciertos:  
0.987

# TUNNING DE PARAMETROS

1

Decision  
Tree

**MAX\_DEPTH,**  
**CRITERION**

2

Random  
Forest

**N\_ESTIMATORS,**  
**CRITERION**

3

Support  
Vector  
Machine

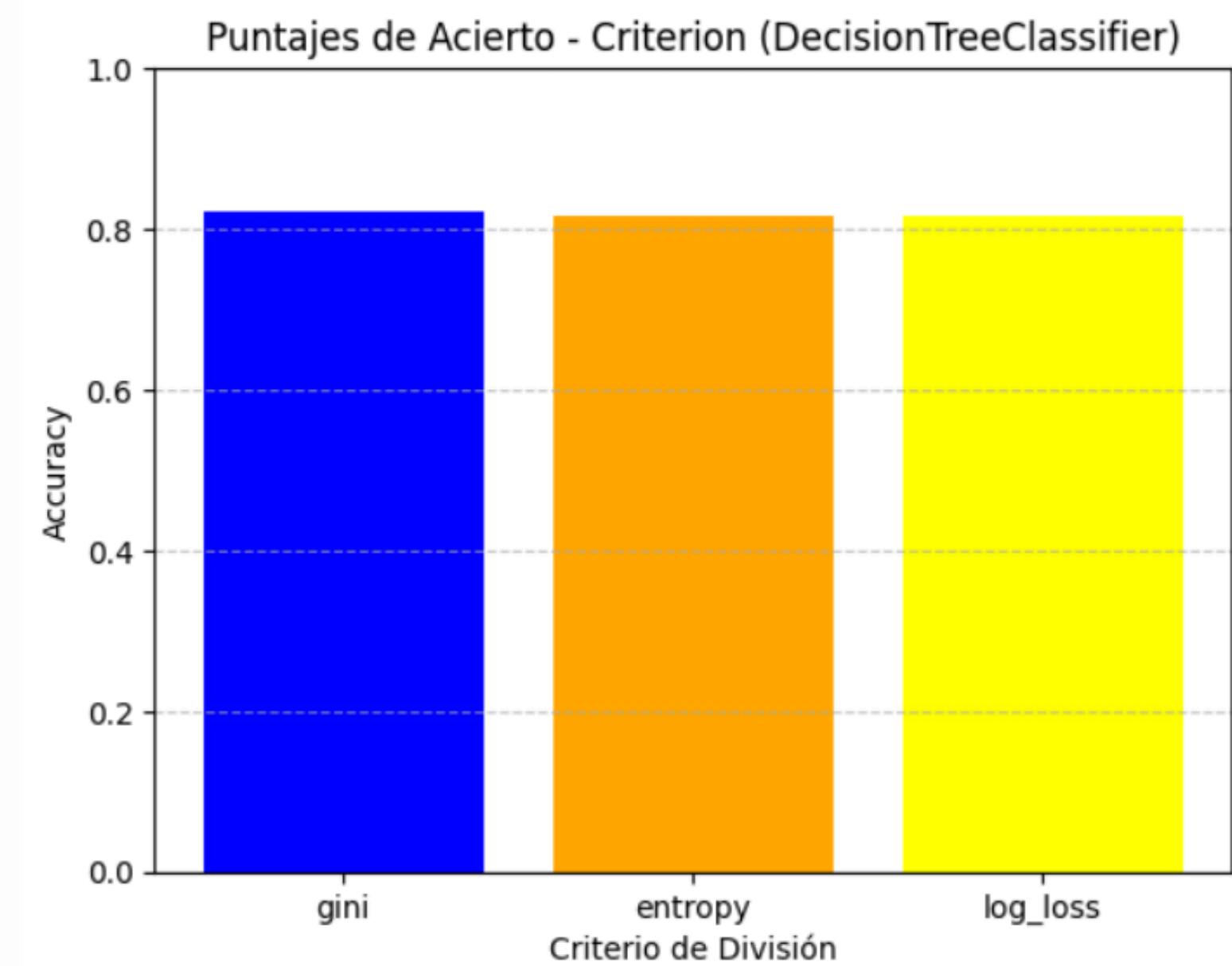
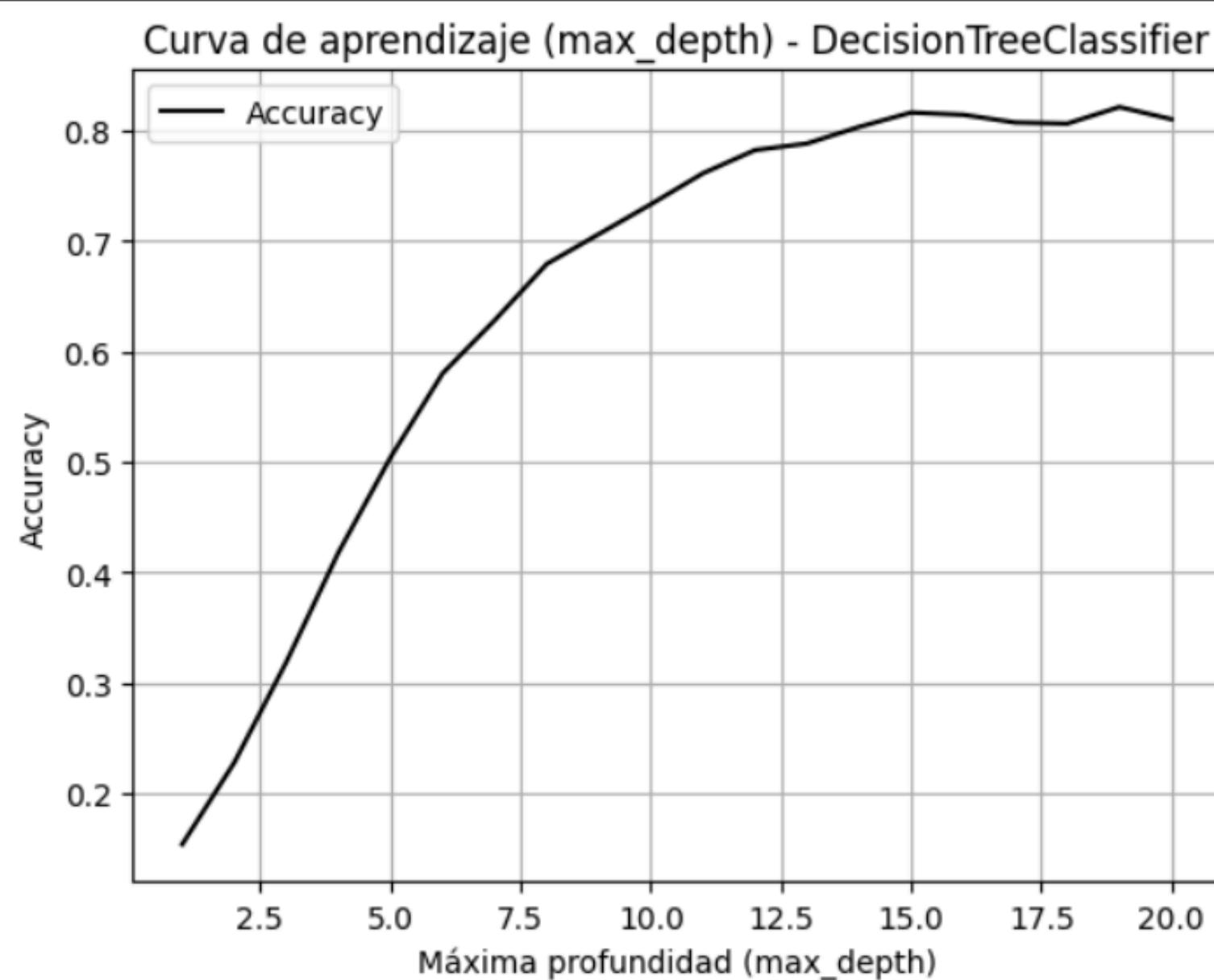
**KERNEL,**  
**GAMMA**

4

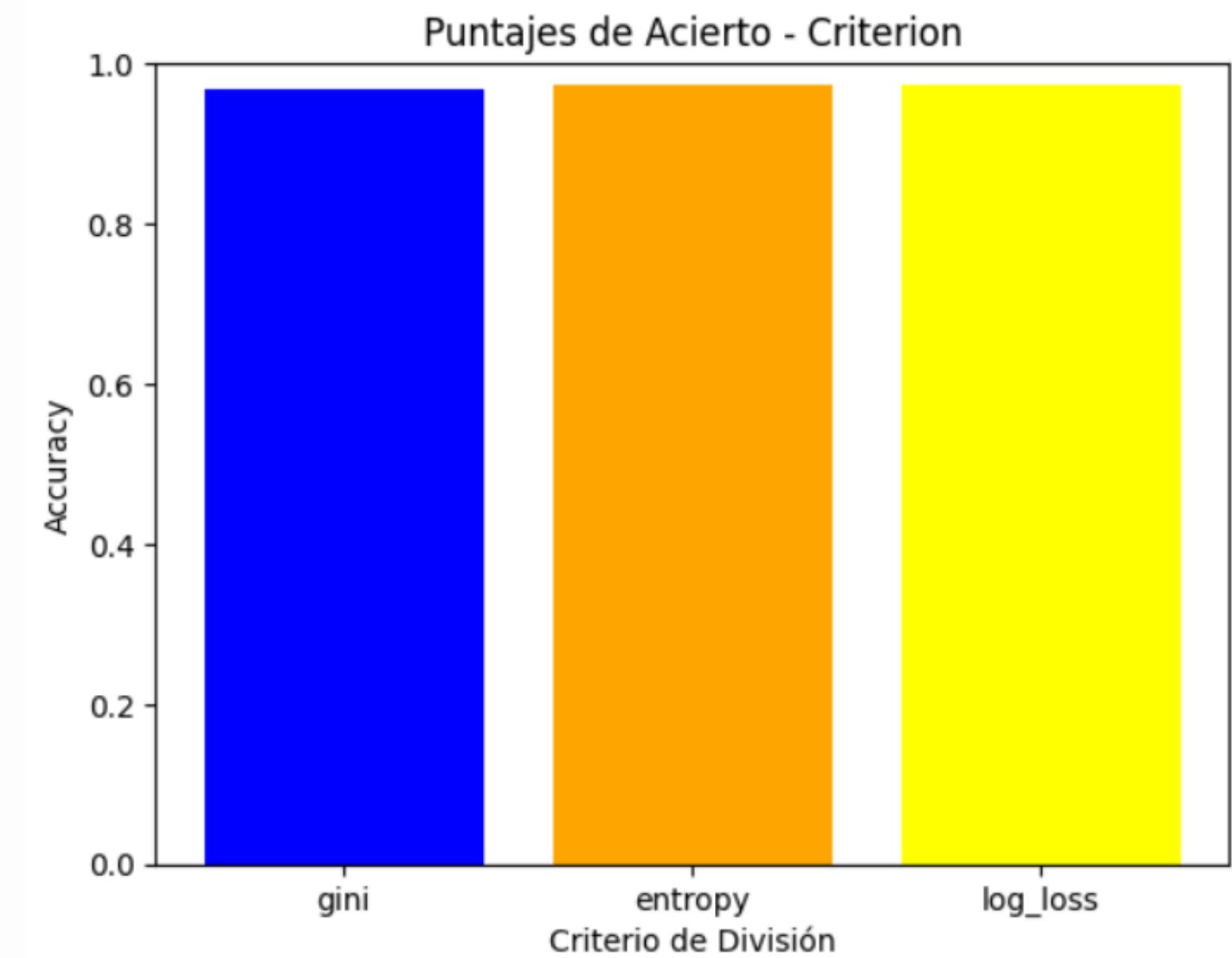
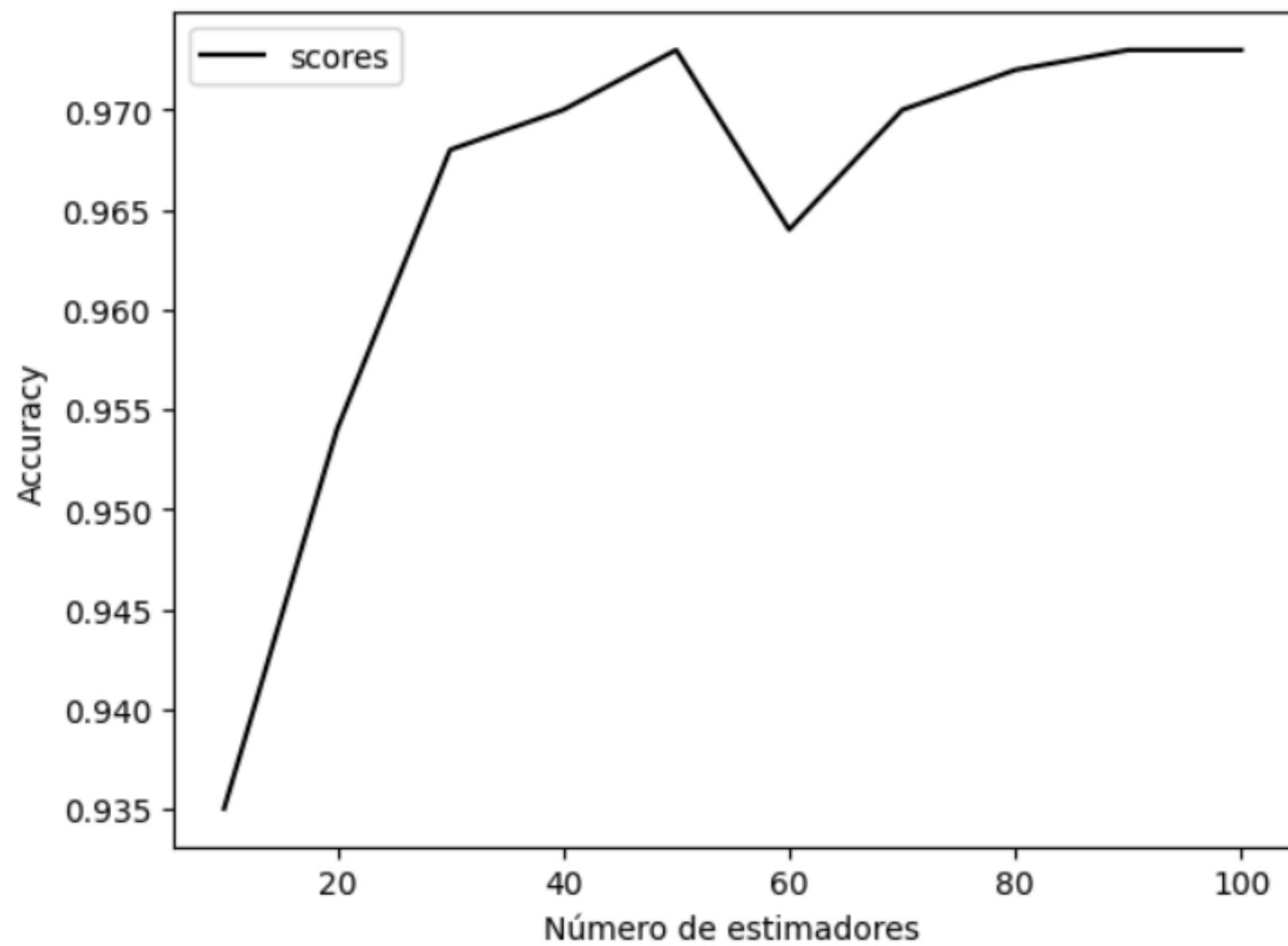
Cross  
Validation

**KFOLDS**

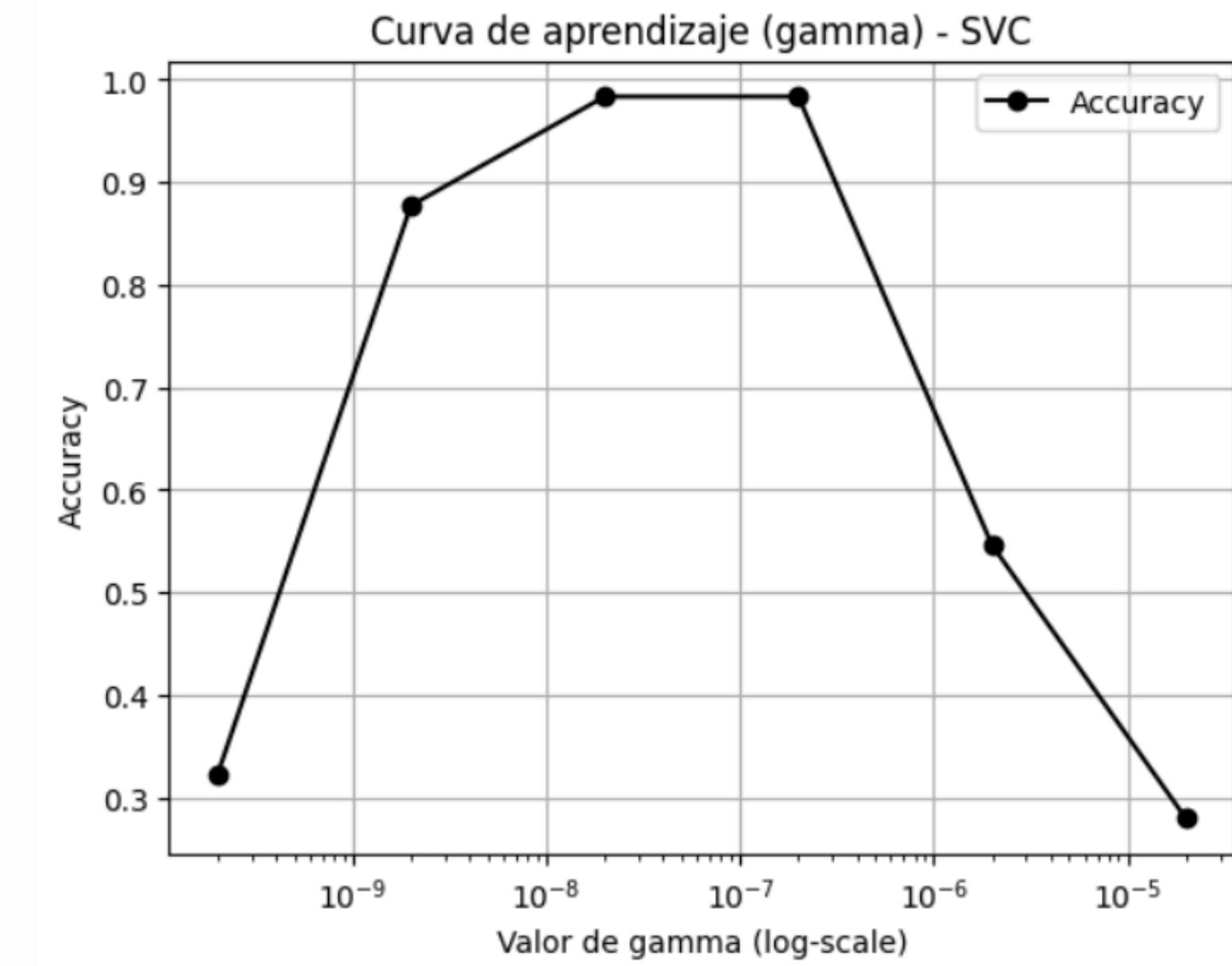
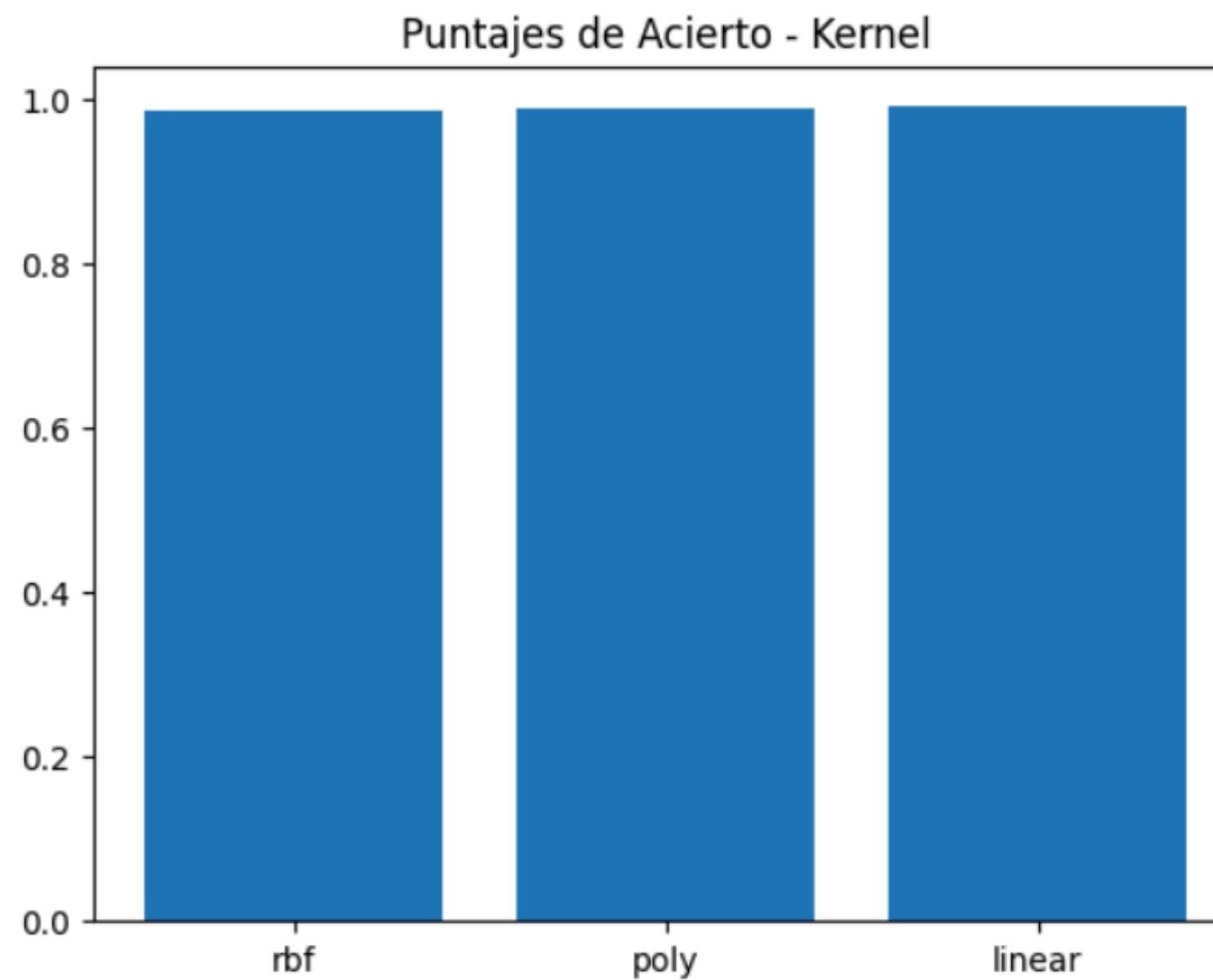
# RESULTADOS



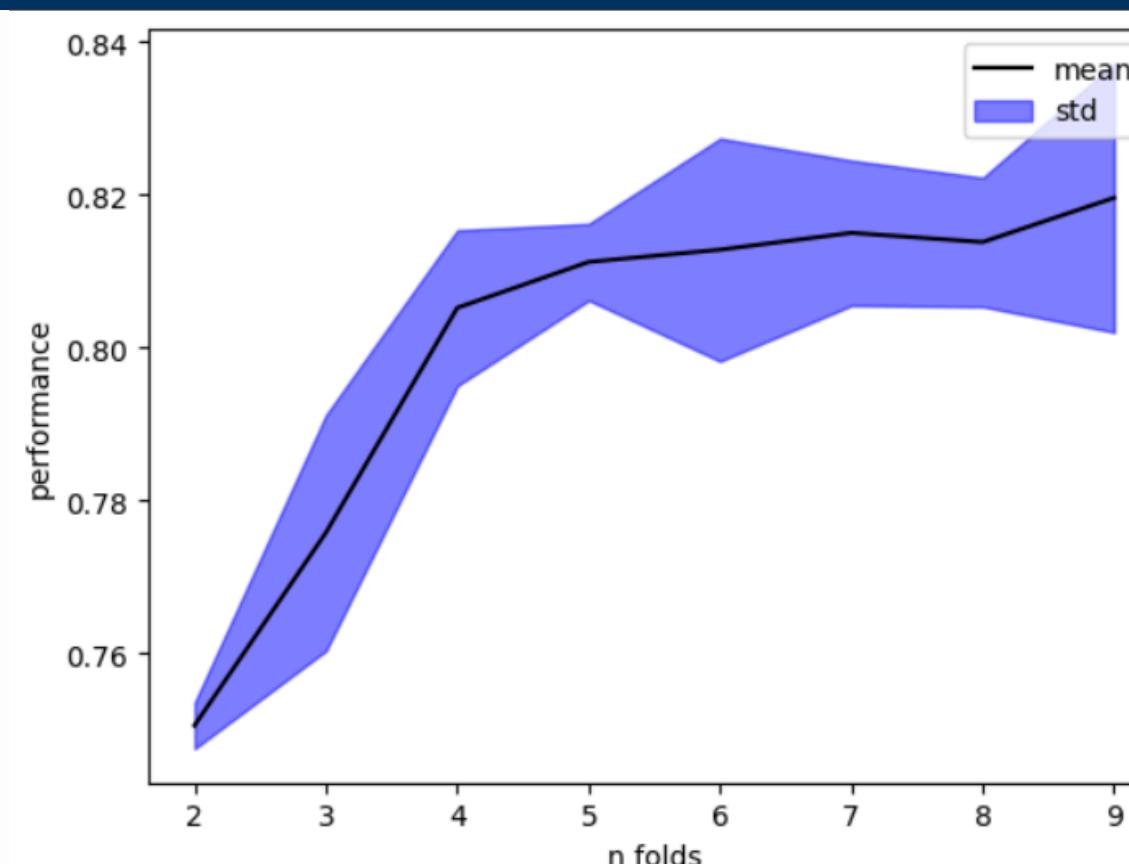
# RESULTADOS



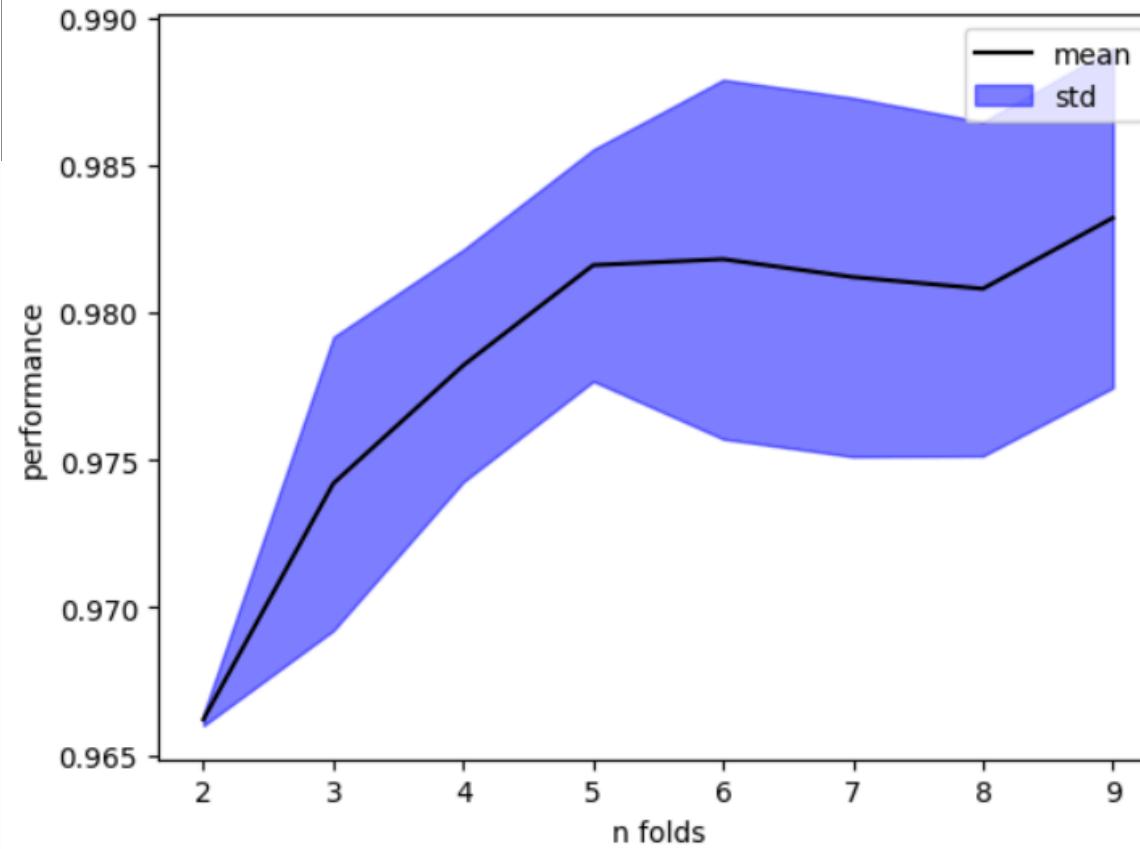
# RESULTADOS



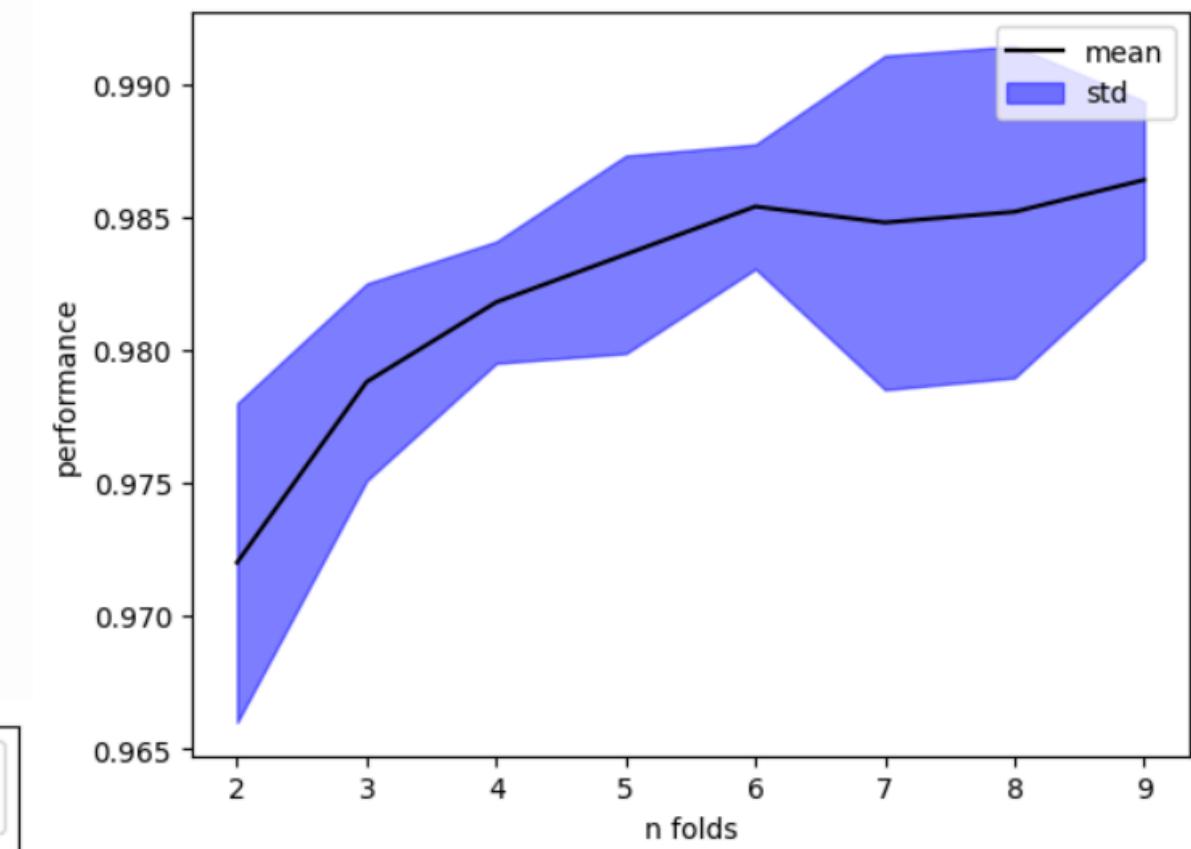
# RESULTADOS



Decision Tree



Random Forest



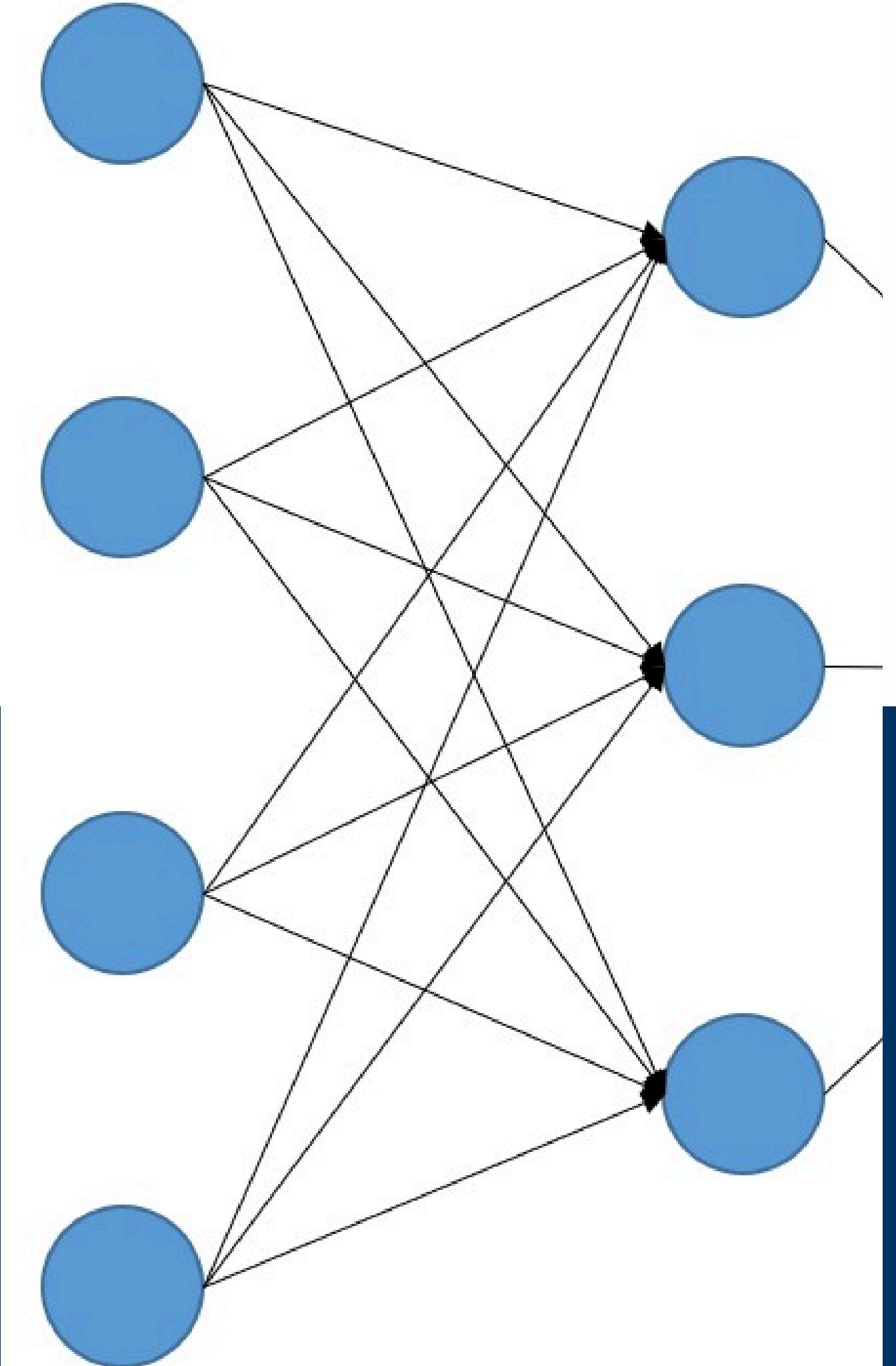
Support Vector  
Machine

# DEEP LEARNING: PERCENTRON MULTICAPA

3 capas ocultas

6 capas ocultas

10 capas ocultas



# RESULTADOS

## ▼ Puntaje de precisión

```
[ ] #@title Puntaje de precisión  
from sklearn.metrics import accuracy_score  
y_pred = np.argmax(model.predict(x_test), axis=-1)  
accuracy = accuracy_score(y_test, y_pred)  
print("\nPrecisión calculada con accuracy_score:", accuracy)
```

→ 32/32 ━━━━━━━━ 1s 18ms/step

Precisión calculada con accuracy\_score: 0.987

3 capas ocultas

# RESULTADOS

## ▼ Puntaje de precisión

```
[ ] #@title Puntaje de precisión
y_pred = np.argmax(model.predict(x_test), axis=-1)
accuracy = accuracy_score(y_test, y_pred)
print("\nPrecisión calculada con accuracy_score:", accuracy)
```

→ 32/32 ————— 1s 39ms/step

Precisión calculada con accuracy\_score: 0.968

6 capas ocultas

# RESULTADOS

## ▼ Puntaje de precisión

```
[ ] #@title Puntaje de precisión  
y_pred = np.argmax(model.predict(x_test), axis=-1)  
accuracy = accuracy_score(y_test, y_pred)  
print("\nPrecisión calculada con accuracy_score:", accuracy)
```

→ 32/32 ━━━━━━━━ 1s 22ms/step

Precisión calculada con accuracy\_score: 0.961

10 capas ocultas

**!MUCHAS  
GRACIAS!**

