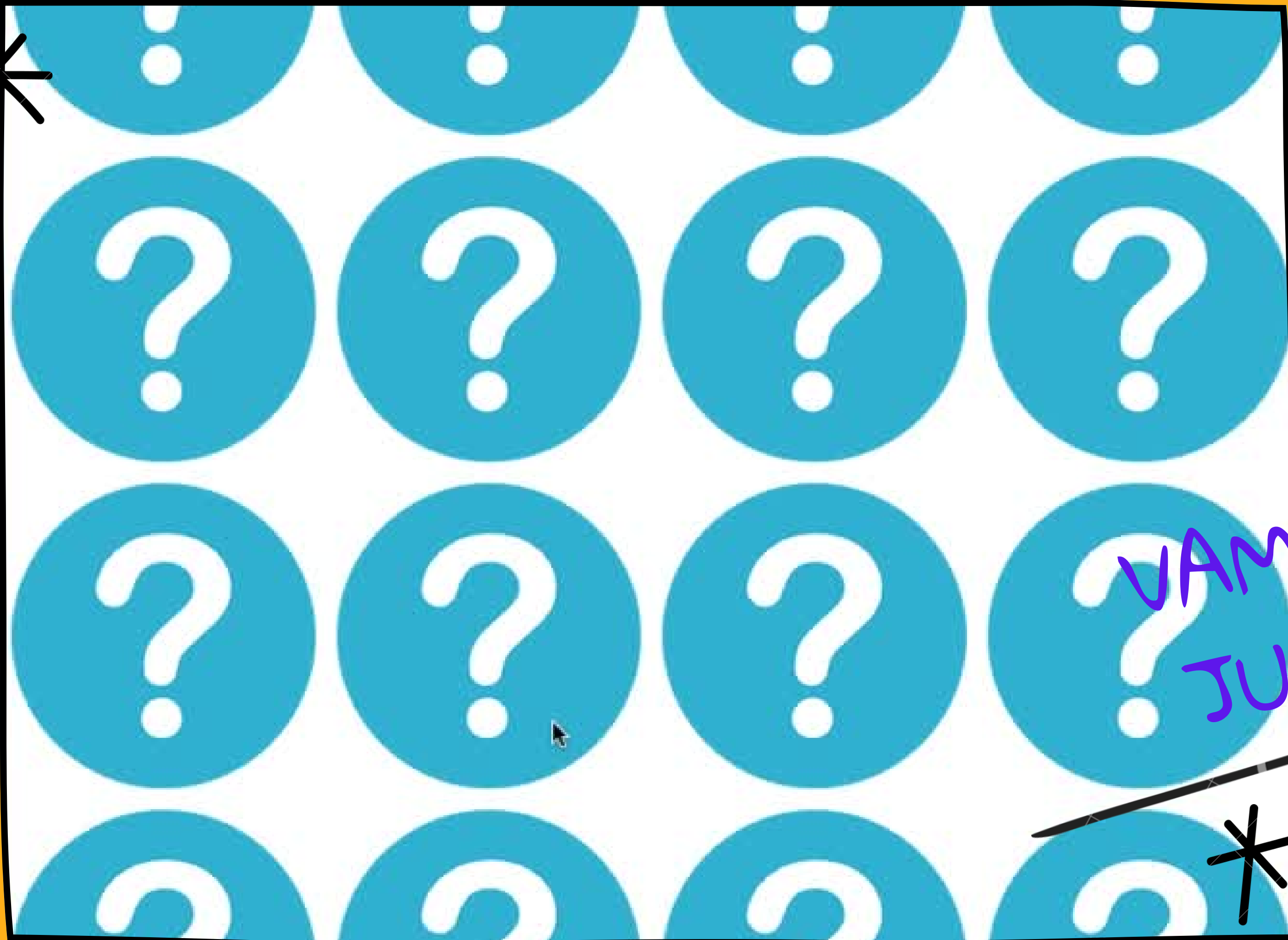


ANALISIS DE ALGORITMOS

# \* CARTAS DE \* MEMORIA \*

~~Canva~~  
ALEJANDRO DIAZ





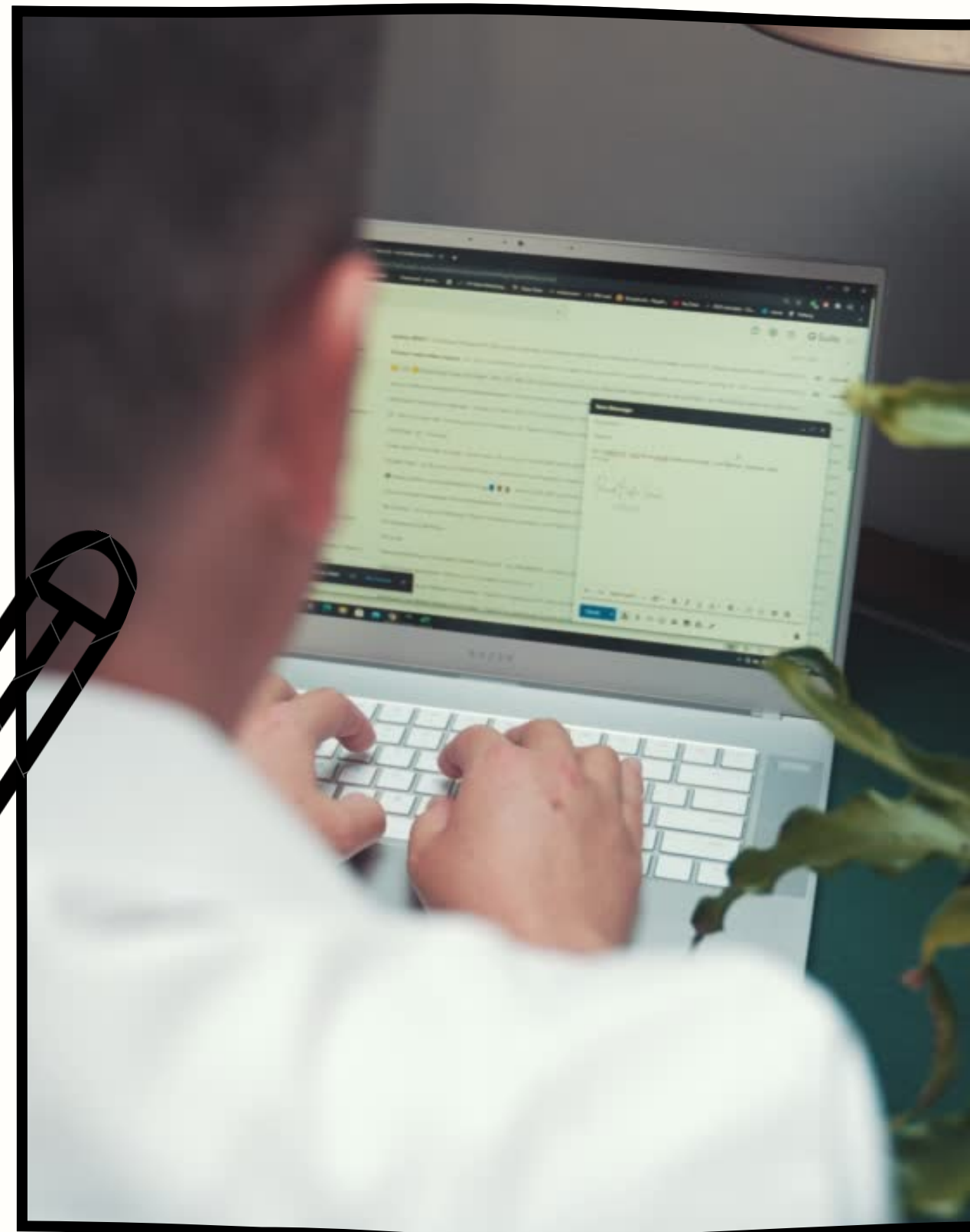
VAMOS A  
JUGAR!



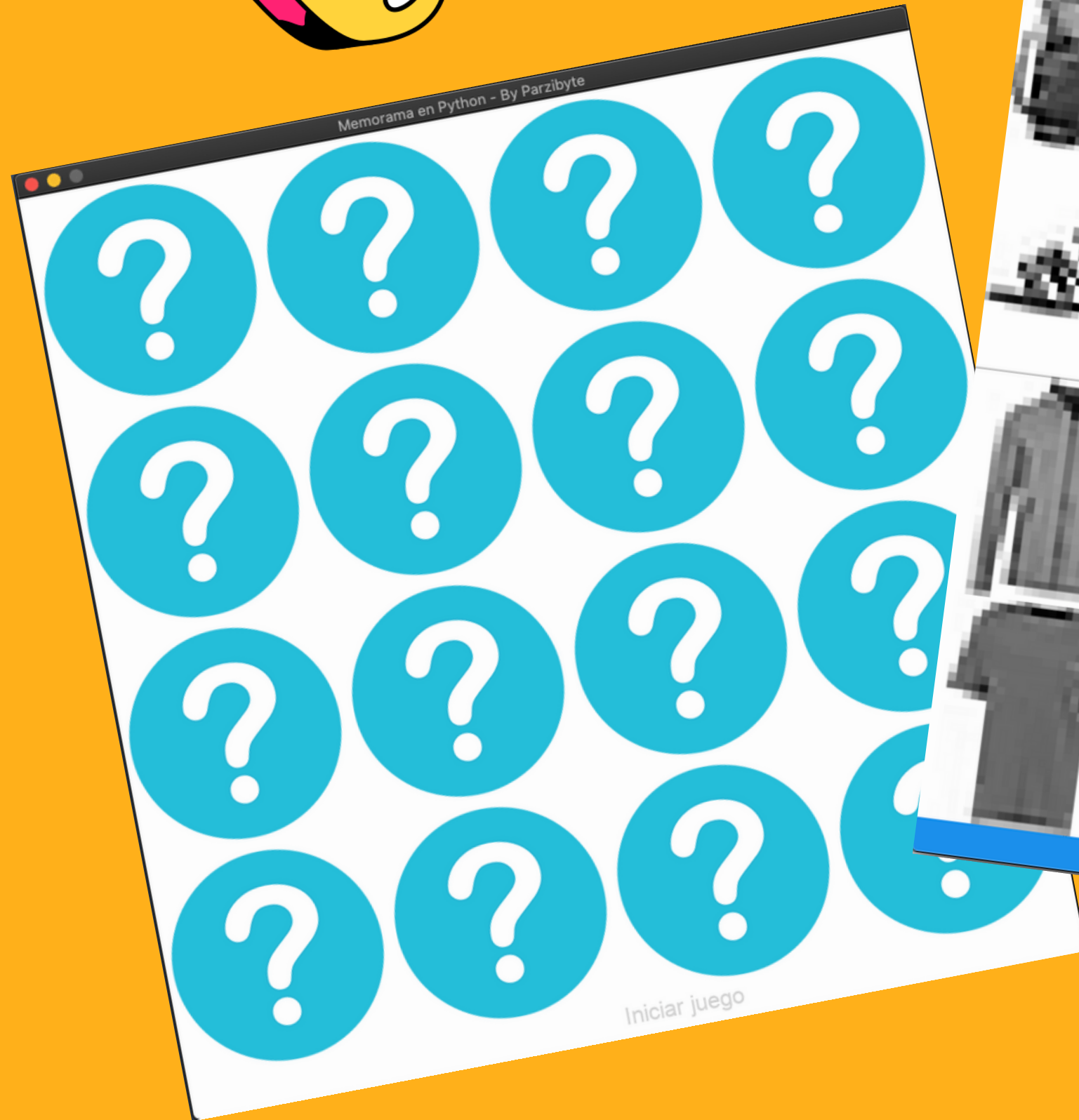
\* \*

# ¿QUÉ VAMOS A ANALIZAR?

✱

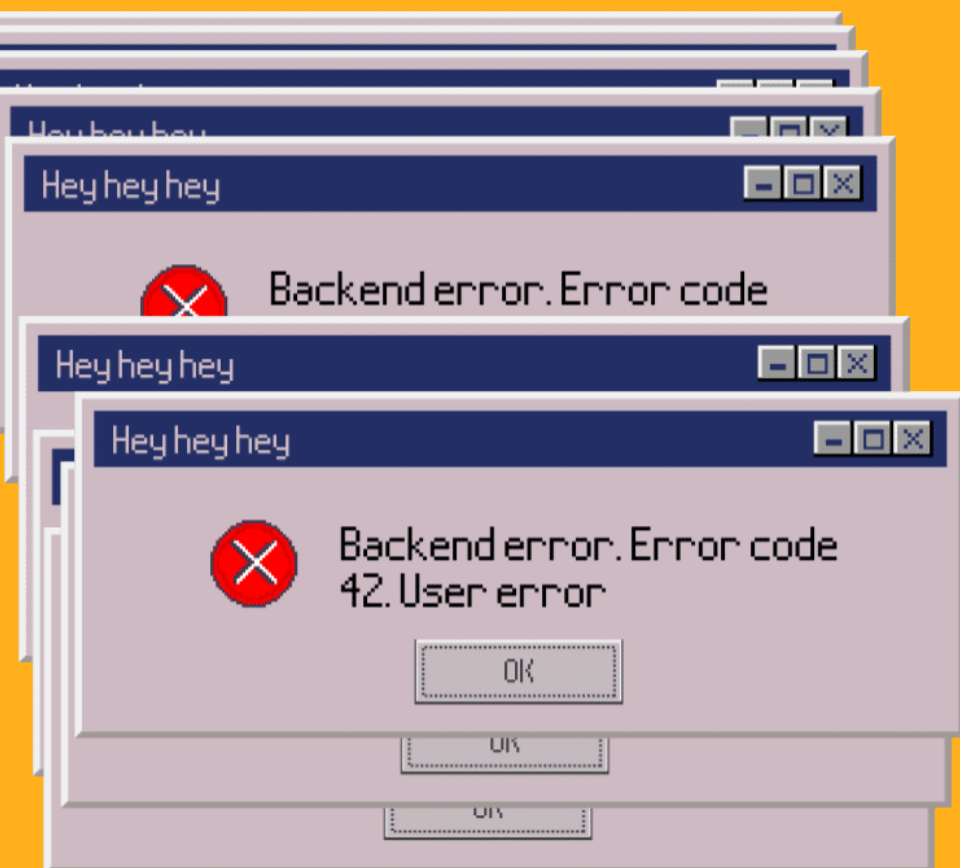






```
✓ [146] #Crear el modelo
0s
modelo = tf.keras.Sequential([
    tf.keras.layers.Flatten(input_shape=(28,28,1)), #1 - blanco y negro
    tf.keras.layers.Dense(50, activation=tf.nn.relu),
    tf.keras.layers.Dense(50, activation=tf.nn.relu),
    tf.keras.layers.Dense(10, activation=tf.nn.softmax) #Para redes de clasificacion
])
```

```
147] #Compilar el modelo
modelo.compile(
    optimizer='adam',
    loss=tf.keras.losses.SparseCategoricalCrossentropy(),
    metrics=['accuracy']
)
```



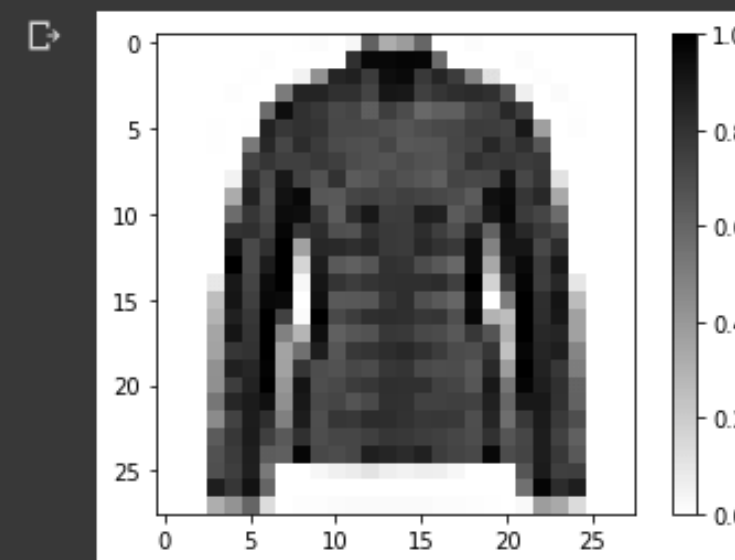
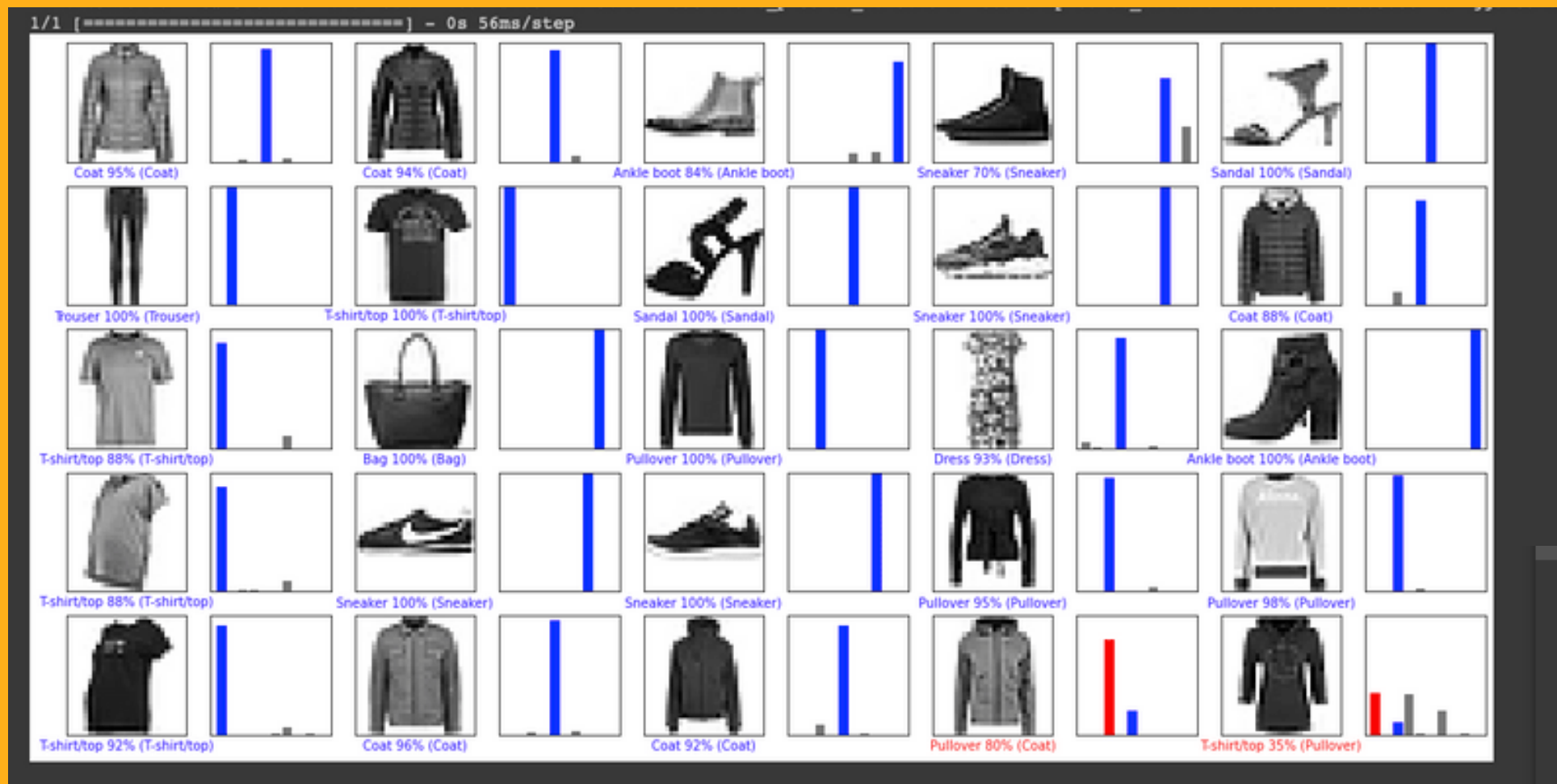
```
[149] #El trabajo por lotes permite que entrenamientos con gran cantidad de datos se haga de manera mas eficiente
TAMANO_LOTE = 32
```

```
#Shuffle y repeat hacen que los datos esten mezclados de manera aleatoria para que la red
#no se vaya a aprender el orden de las cosas
datos_entrenamiento = datos_entrenamiento.repeat().shuffle(num_ej_entrenamiento).batch(TAMANO_LOTE)
datos_pruebas = datos_pruebas.batch(TAMANO_LOTE)
```

```
import math
```

```
#Entrenar
historial = modelo.fit(datos_entrenamiento, epochs=5, steps_per_epoch= math.ceil(num_ej_entrenamiento/TAMANO_LOTE))
```

```
Epoch 1/5
1875/1875 [=====] - 11s 3ms/step - loss: 0.5158 - accuracy: 0.8179
Epoch 2/5
1875/1875 [=====] - 5s 3ms/step - loss: 0.3839 - accuracy: 0.8601
Epoch 3/5
1875/1875 [=====] - 6s 3ms/step - loss: 0.3489 - accuracy: 0.8719
Epoch 4/5
1875/1875 [=====] - 6s 3ms/step - loss: 0.3213 - accuracy: 0.8819
Epoch 5/5
1875/1875 [=====] - 5s 3ms/step - loss: 0.3058 - accuracy: 0.8866
```



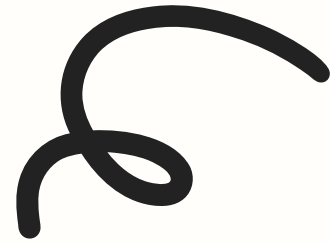
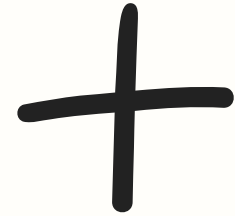
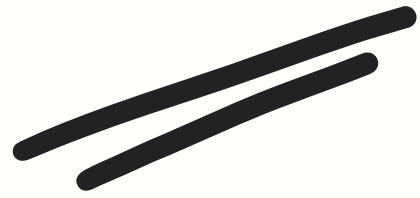
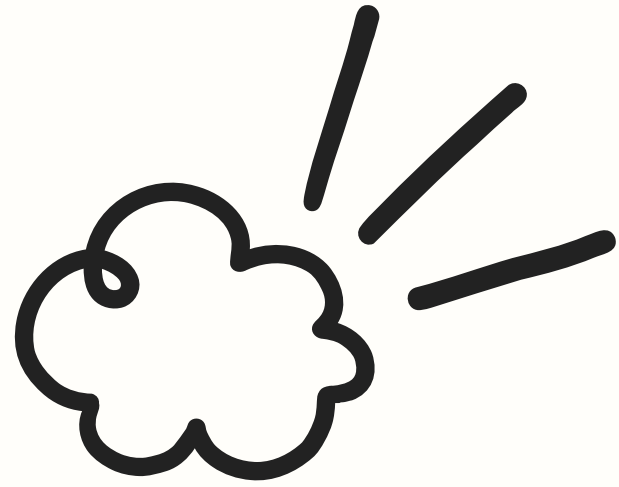
Etiqueta real=>Coat

```
[170] #Probar una imagen suelta
i=1
imagen = imagenes_prueba[i]
imagen = np.array([imagen])
prediccion = modelo.predict(imagen)

print("Prediccion: " + nombres_clases[np.argmax(prediccion[0])])
```

1/1 [=====] - 0s 71ms/step  
Prediccion: Coat





GRACIAS!

