

RECONOCIMIENTO FACIAL USANDO SVD: APLICACIÓN EN RENIEC

I N T E G R A N T E S :

CARLOS MARIÑOS, DEYSI
ZAVALETA Y PAULA RAMOS



ÍNDICE:

I. INTRODUCCIÓN

II. DESARROLLO DE CONCEPTOS

III. ESTADO DEL ARTE

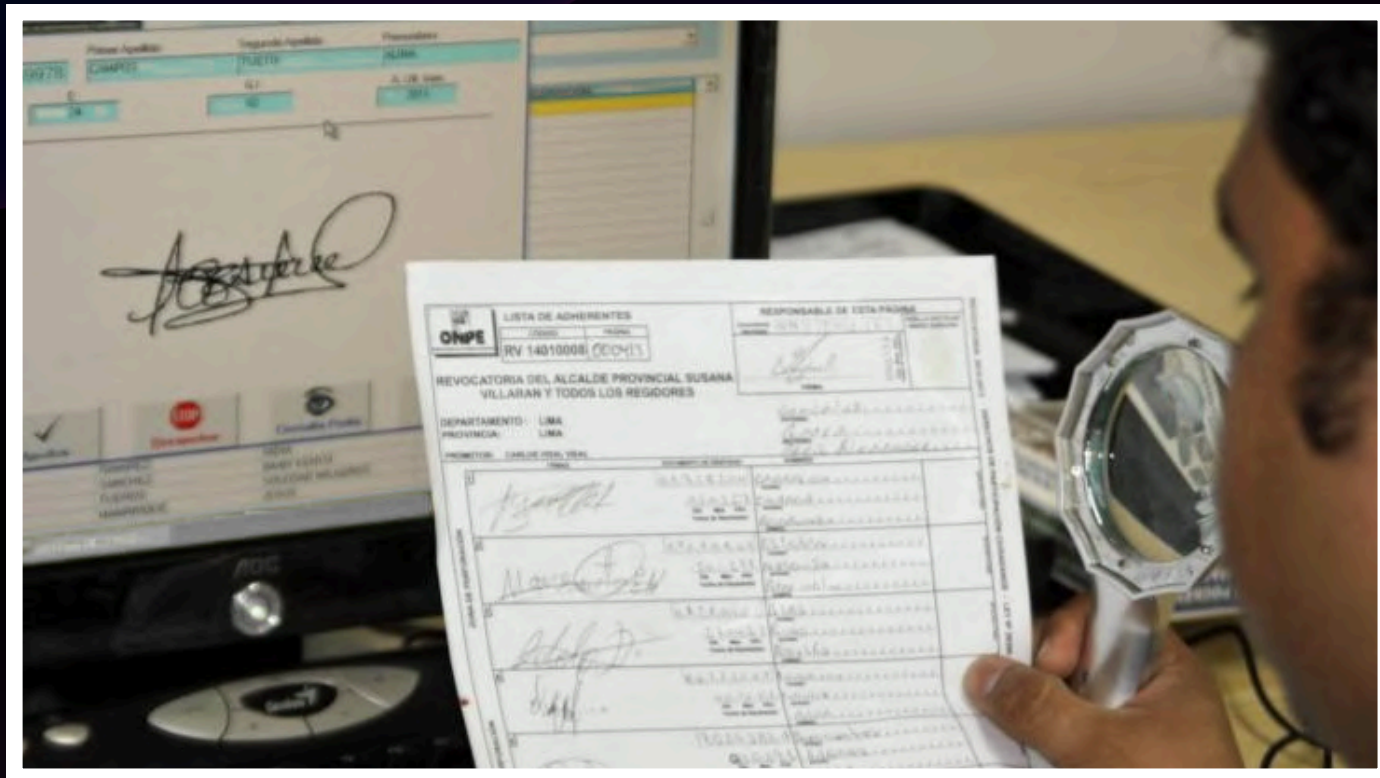
IV. METODOLOGÍA PROPUESTA

V. PROCEDIMIENTO Y RESULTADOS

VI. CONCLUSIONES Y RECOMENDACIONES



INTRODUCCIÓN



En abril de 2025, Perú enfrentó una crisis electoral tras descubrirse más de 300,000 **firmas falsas en la inscripción de partidos políticos**, lo que evidenció fallas en los métodos actuales de verificación de identidad. Ante esta situación, se propone implementar un sistema de reconocimiento facial que compare la foto del DNI con una imagen capturada en tiempo real, utilizando técnicas como:

- SVD (Descomposición en Valores Singulares)
- PCA (Análisis de Componentes Principales)

DESARROLLO DE CONCEPTOS



Verificación biométrica

En este caso, se aplicará el reconocimiento facial como medida adicional de seguridad para evitar fraudes.

Reconocimiento facial

Es un proceso automatizado que compara características faciales usando métodos como PCA o SVD, permitiendo identificar similitudes entre rostros con rapidez y precisión.

Singular Value Decomposition (SVD)

SVD descompone una imagen en tres matrices. La matriz Σ contiene los valores más relevantes del rostro. Comparando estos valores entre dos imágenes, se puede medir su similitud.



DESARROLLO DE CONCEPTOS



Principal Component Analysis (PCA)

PCA reduce la complejidad de los datos (como imágenes) extrayendo sus componentes principales. Estos conservan la mayor parte de la información y permiten comparaciones más eficientes. Se calcula a través de SVD.

Cálculo del error

Luego de extraer los vectores de características (por SVD o PCA), se calcula un error entre ellos. Si es bajo (o la similitud es alta), indica que las imágenes podrían ser de la misma persona.

Umbral de decisión

Se establece un valor límite (umbral) para decidir si dos imágenes son lo suficientemente parecidas. Si el valor de similitud supera ese umbral, se valida la identidad.

Aplicaciones en procesos electorales:

Integrar reconocimiento facial en la validación de firmas ayudaría a reducir la suplantación de identidad y fortalecer la transparencia en los procesos electorales.



ESTADO DEL ARTE

Tipos de reconocimiento

- Verificación (1 a 1): Compara imagen consulta vs. DNI (caso RENIEC).
- Identificación (1 a muchos): Compara imagen consulta con una base de datos masiva.

(Crosswhite et al.)

1) PCA - Eigenfaces (Turk y Pentland, MIT):

- Convierte imágenes en vectores → calcula matriz de covarianza.
- Obtiene componentes principales (eigenfaces).
- Comparación eficiente, pero sensible a iluminación y expresión.

2) SVD - (Hsu y Chen):

- Mantiene estructura 2D de la imagen.
- Aplica SVD directamente: $A = U \Sigma V^t$.
- Precisión: 97.5% con 12 vectores.
- Mejor manejo de expresión e iluminación.



Figura 1: Ejemplo de Imágenes de 3 sujetos con diferentes expresiones faciales de la base de datos ORL|utilizada en el estudio de Hsu y Chen [7]

ESTADO DEL ARTE

3) SVD - (Zeng):

- Crea un "face space" con imágenes conocidas.
- Proyecta nuevas imágenes y compara con distancias.
- Alta eficiencia computacional, buen desempeño en condiciones controladas. [8]

Métodos modernos

- CNN y LDA (Fisherfaces):
- Más robustos a condiciones no controladas.
- Limitaciones:
- Requieren grandes datos, potencia computacional y son poco interpretables. [9]

$$A = U \Sigma V^T$$

Teniendo:

- U : vectores singulares "verticales" (filas)
- Σ : valores singulares
- V^T : vectores singulares "horizontales" (columnas)

Elección del estudio

SVD fue elegido por:

- Claridad matemática y computacional.
- Buen desempeño con baja variación.
- Bajo costo de almacenamiento y mayor interpretabilidad.



METODOLOGÍA PROPUESTA

Programa en Python que pueda verificar la identidad de un votante al comparar la imagen de su rostro en RENIEC con una foto capturada en tiempo real al inscribirse como ciudadano que apoya al partido.

Paso 1: Preprocesamiento de imágenes

Se considerarán sesenta imágenes para formar la matriz X destinada al entrenamiento.

$$X \in R^{60 \times 1000}$$

Paso 2: Aplicación de SVD

A cada imagen A_i , se le aplicará la descomposición en valores singulares:

$$X = U \Sigma V^T$$

donde $U \in R^{60 \times 60}$, $\Sigma \in R^{60 \times 60}$ y $V \in R^{60 \times 10000}$.

- La matriz Σ_i : valores singulares ordenados de forma descendente
- U y V : matrices ortogonales
- U : Valores singulares izquierdos de A_i
- V : valores singulares derechos de A_i .

Paso 3: Reducción de dimensionalidad

Con el fin de conservar sólo la información más relevante, se seleccionan los primeros k valores singulares de Σ_k y se construye una aproximación de la imagen.



METODOLOGÍA PROPUESTA

Funcionamiento de Numpy (np.linalg.svd)

Esta función calcula directamente los valores de las matrices U, Σ , V, que contienen lo mencionado previamente.

Paso 4: Construcción del vector de características

Para cada imagen, se proyecta su vector aplanado en el subespacio de características definido por los primeros k vectores de V. Estos serían los Eigenfaces con más varianza explicada

$$Firma = X[i] \cdot V_k^t$$

Paso 5: Calcular similitud

Para poder evaluar si dos imágenes son de la misma persona, se compararán los dos vectores (x_1 , x_2) utilizando la distancia euclidiana entre sus proyecciones.

$$d(x_1, x_2) = \sqrt{\sum_{i=1}^k (x_{1i} - x_{2i})^2}$$

Cuando d sea menor al umbral definido (3500) se considera que las imagenes pertenecen a la misma persona.

Paso 6: Visualización y validación

Se evalúa el sistema utilizando un conjunto de imagenes nuevo Xtest compuesto por 5 imagenes de la persona utilizada para el entrenamiento así como otras 5 imagenes diferentes a la persona.

- Se calcula el accuracy, matriz de confusión y otras métricas.



PROCEDIMIENTO Y RESULTADOS

Prueba preliminar:

1

Librerías

- LOpenCV
- NumPy
- Matplotlib
- OS

- Se utiliza la clase **VideoCapture** de **OpenCV** para capturar la imagen del ciudadano en tiempo real.
- Mediante un clasificador de CV2, `haarcascade_frontaldace_default.xml`, se reconoce el rostro en la imagen.



Figura 2: Captura del rostro del ciudadano usando OpenCV, con detección facial utilizando un clasificador Haar.

3

- Se carga la imagen del **DNI** del ciudadano (como matriz de imagen) la cual es recortada para priorizar el rostro.
- Ambas imágenes son convertidas en escala de grises y redimensionadas a 100 x 100 píxeles

4

Luego, se descomponen ambas imágenes con la función `numpy.linalg.svd` y se guardan las matrices U , S y V^T correspondientes a cada una, de modo que:



Figura 3: Rostro original, rostro reconstruido con 10 valores singulares y gráfico de caída de valores singulares.

- Se determinó que con 15 componentes se capturaba el 90% de la varianza.

Este método se descartó



Análisis PCA

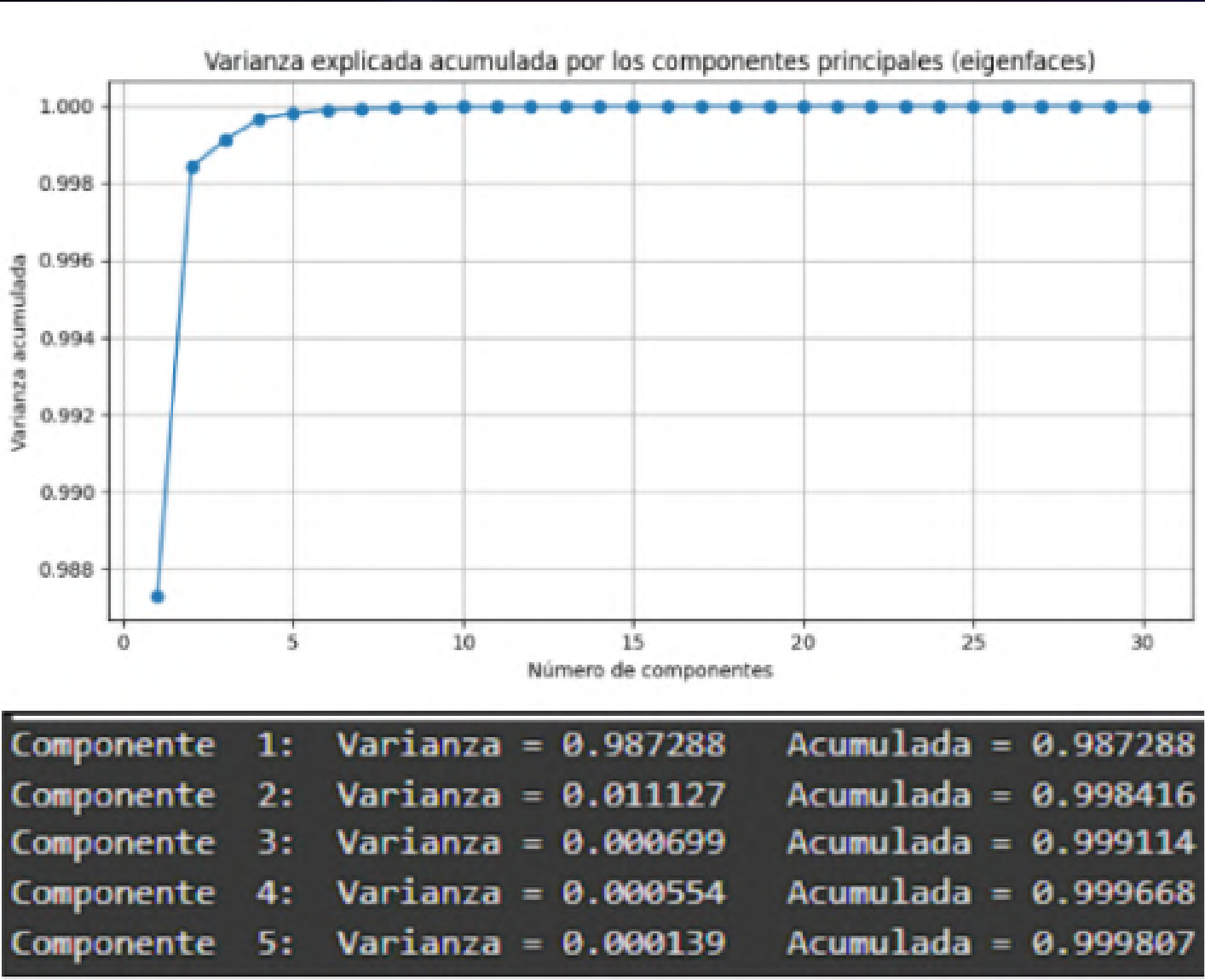
Si tenemos una matriz de datos $X \in \mathbb{R}^{n \times m}$, donde cada fila es una **imagen "aplanada" (vectorizada)**, el procedimiento clásico de PCA consiste en:

1. Centrar los datos:
Restar la media de cada característica (columna) para que la matriz tenga media cero.

2. Calcular la SVD de la matriz centrada:

$$X_{\text{centrada}} = U \Sigma V^T$$

3. Obtener los componentes principales :
Las columnas de V representan las direcciones de máxima varianza, es decir, los *eigenfaces* o "rostros característicos".

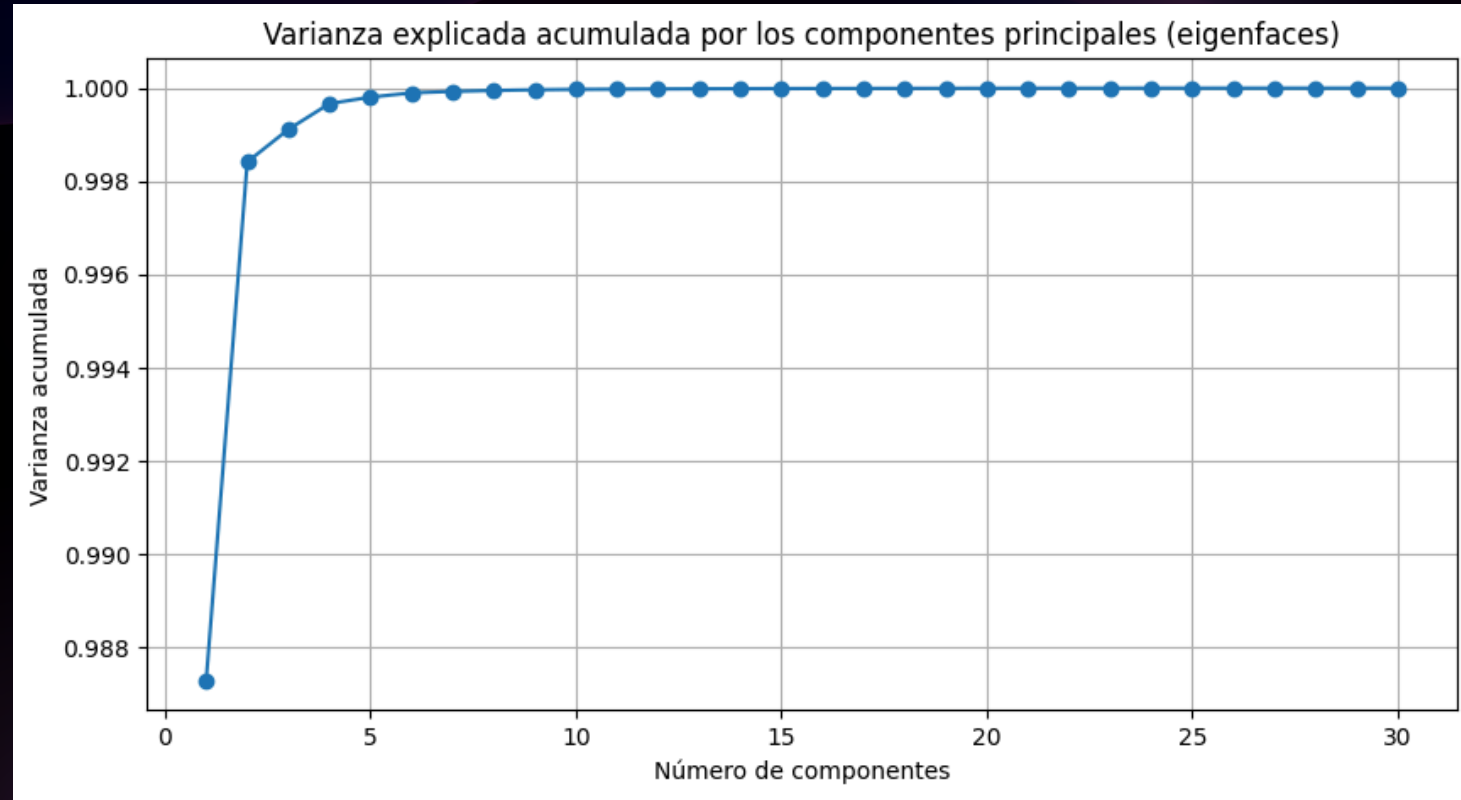
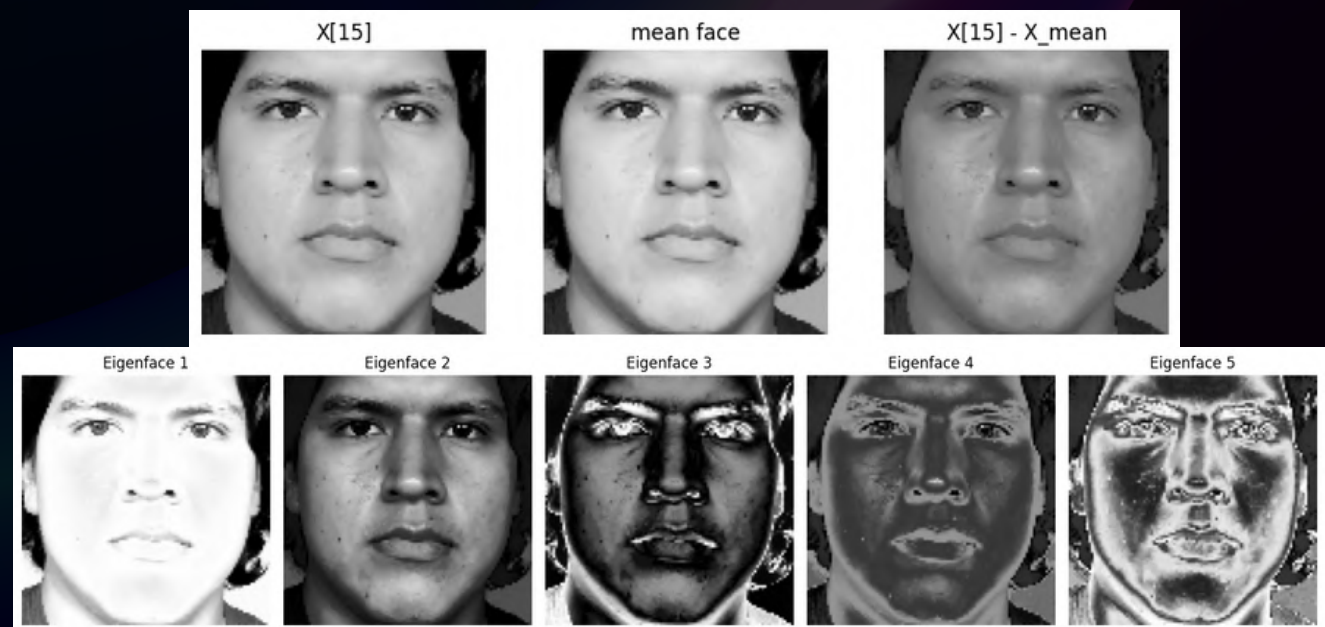


SVD es la base algebraica de PCA, ya que al aplicar SVD a una matriz de datos centrada X, los vectores de V son los autovectores de la matriz de covarianza , los componentes principales.
 Σ : a tovalores y representan la varianza explicada por cada componente.

$$\frac{1}{n-1} X^T X$$



X-Xmean=CaraCentrada CASO 1 IMAGEN



Componente 1:	Varianza = 0.987288	Acumulada = 0.987288
Componente 2:	Varianza = 0.011127	Acumulada = 0.998416
Componente 3:	Varianza = 0.000699	Acumulada = 0.999114
Componente 4:	Varianza = 0.000554	Acumulada = 0.999668
Componente 5:	Varianza = 0.000139	Acumulada = 0.999807

Comparando xtest[0] vs X[15]:
Distancia (SVD, k=5): 2717.35
✓ Probablemente la MISMA persona

X[0]

X[15]

Comparando xtest[1] vs X[15]:
Distancia (SVD, k=5): 2282.83
✓ Probablemente la MISMA persona

X[1]

X[15]

Comparando xtest[4] vs X[15]:
Distancia (SVD, k=5): 2003.11
✓ Probablemente la MISMA persona

X[4]

X[15]

Comparando xtest[5] vs X[15]:
Distancia (SVD, k=5): 3245.33
✓ Probablemente la MISMA persona

X[5]

X[15]

Comparando xtest[8] vs X[15]:
Distancia (SVD, k=5): 3150.67
✓ Probablemente la MISMA persona

X[8]

X[15]

Comparando xtest[9] vs X[15]:
Distancia (SVD, k=5): 4198.75
✗ Probablemente NO es la misma persona

X[9]

X[15]

Comparando xtest[6] vs X[15]:
Distancia (SVD, k=5): 5796.18
✗ Probablemente NO es la misma persona

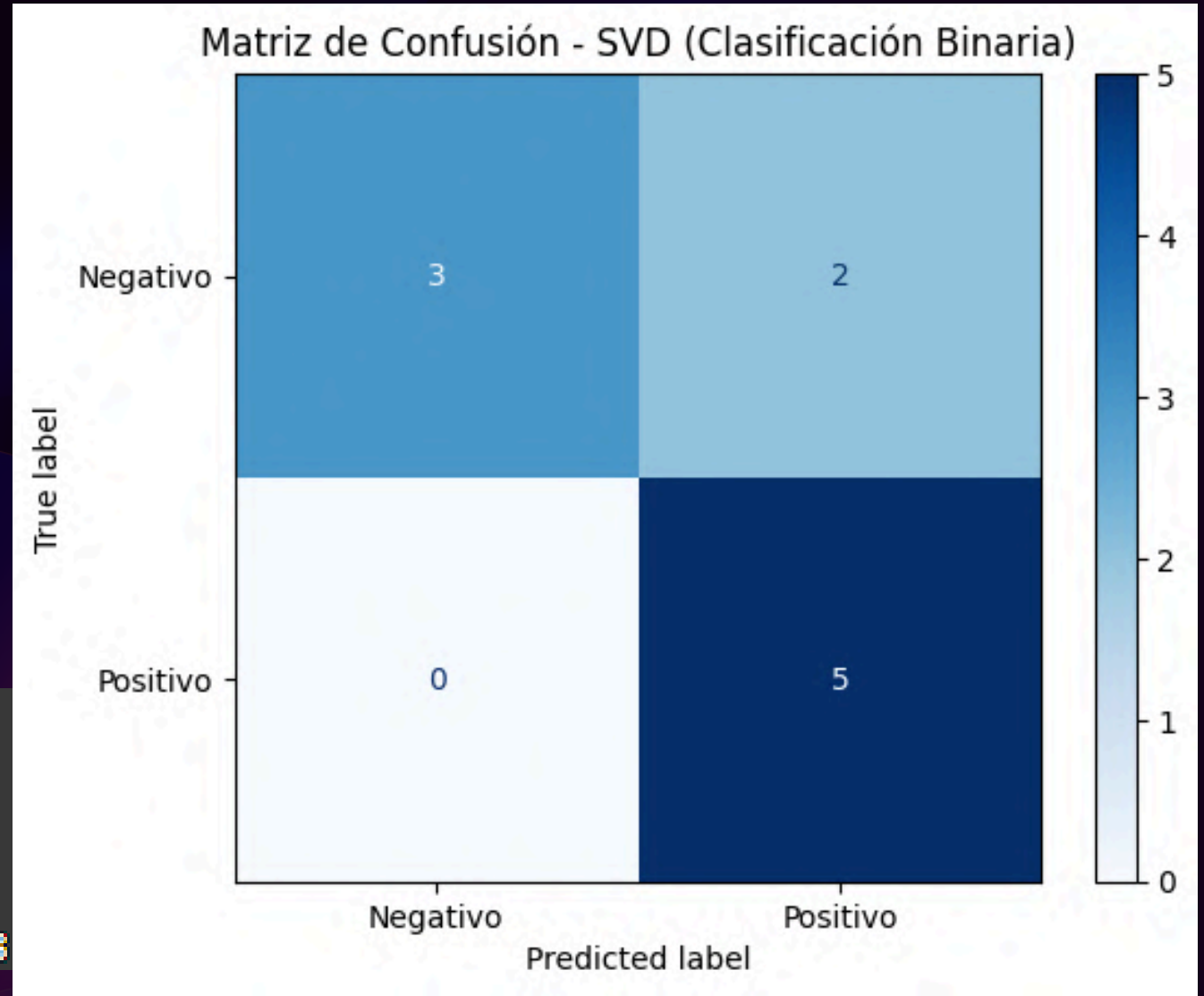
X[6]

X[15]

Comparando xtest[7] vs X[15]:
Distancia (SVD, k=5): 5546.60
✗ Probablemente NO es la misma persona

X[7]

X[15]



--- SVD ---
Accuracy: 0.80
Tiempo total: 0.01 s
Memoria total aprox: 2604.53 KB
Varianza explicada (k=5): 0.9998

PCA CASO 1 IMAGEN

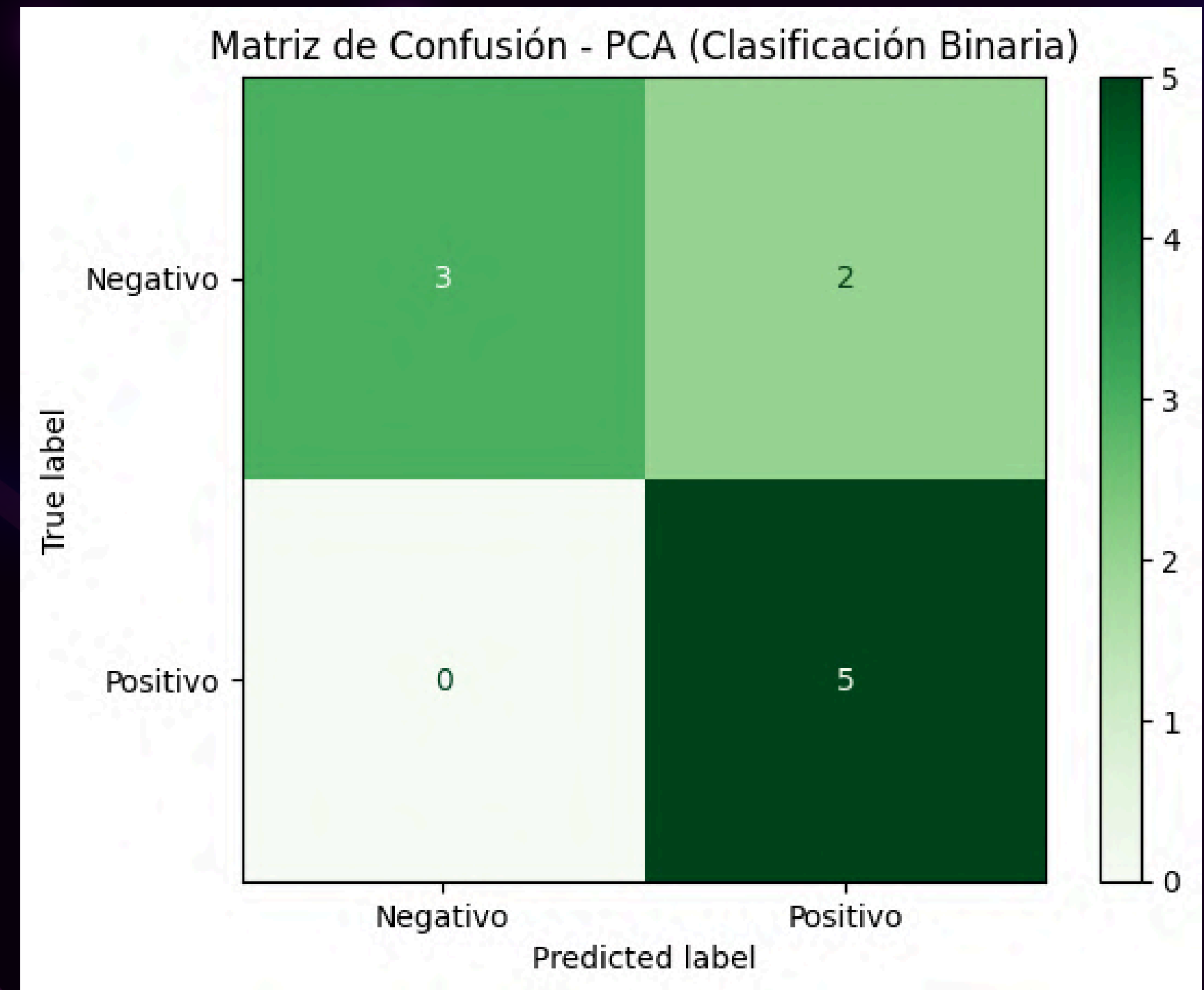
--- PCA ---

Accuracy: 0.80

Tiempo total: 0.05 s

Memoria total aprox: 3015.55 KB

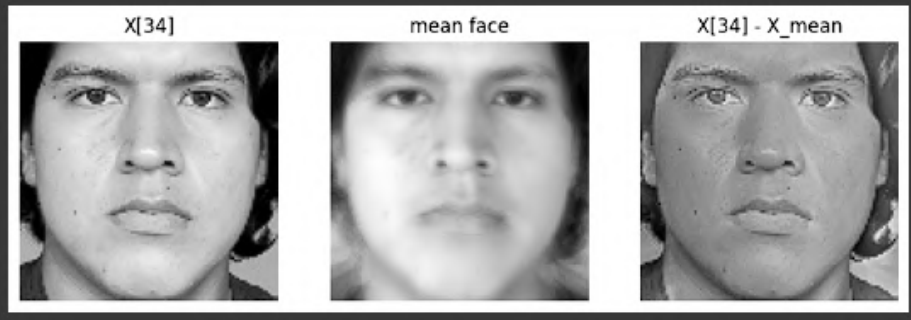
Varianza explicada (k=5): 0.9998





Análisis usando SVD y PCA realizando comparación de la foto del DNI
contra un conjunto de entrenamiento de 60 imágenes

X-Xmean=CaraCentrada



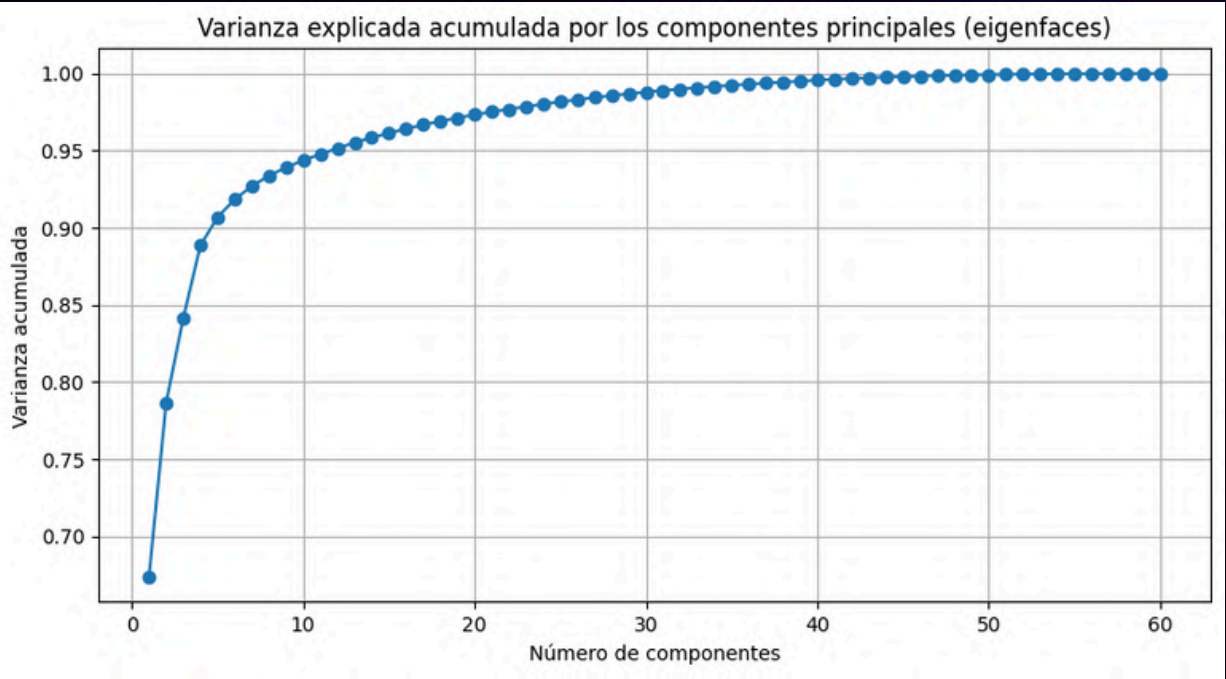
SVD(60 img en train)

Eigenfaces y la variabilidad que representa cada uno de los componentes principales



Eigenfaces y variabilidad

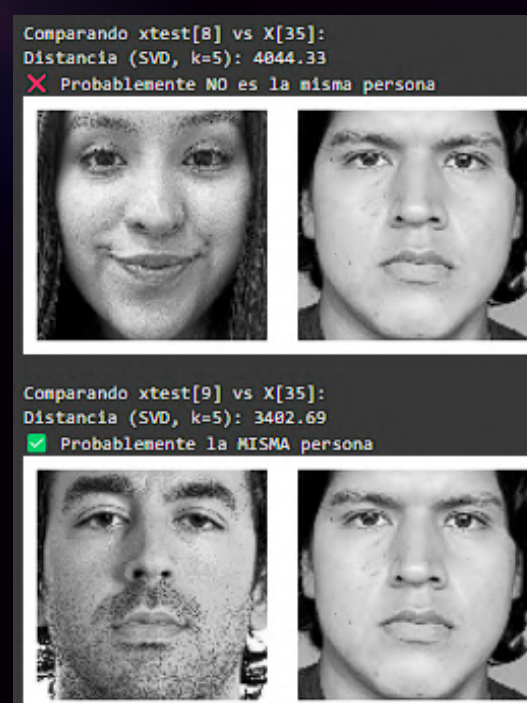
Se muestran los 5 eigenfaces hallados junto con un gráfico de varianza explicada acumulada por cada componente.



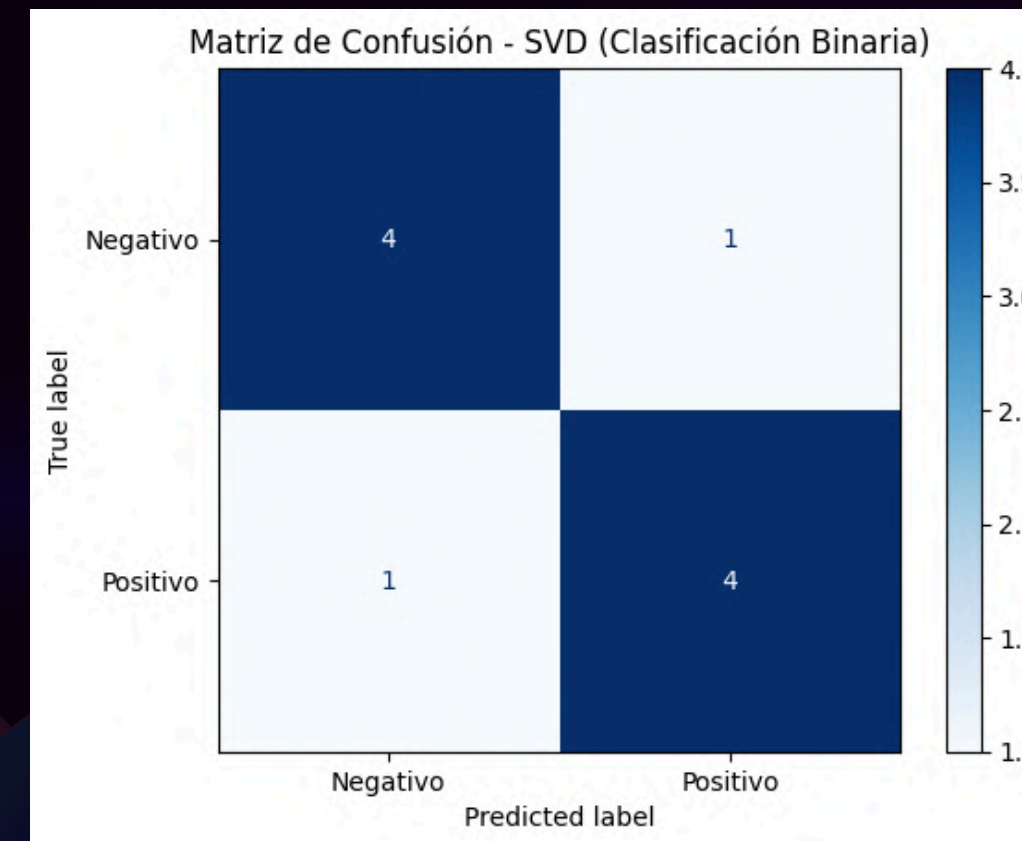
Componente 1:	Varianza = 0.674846	Acumulada = 0.674846
Componente 2:	Varianza = 0.112697	Acumulada = 0.786743
Componente 3:	Varianza = 0.054855	Acumulada = 0.841598
Componente 4:	Varianza = 0.046992	Acumulada = 0.888598
Componente 5:	Varianza = 0.018329	Acumulada = 0.906919

Análisis usando SVD y PCA realizando comparación de la foto del DNI contra un conjunto de entrenamiento de 60 imágenes

Cuatro comparaciones de imágenes de entrenamiento contra imágenes nuevas



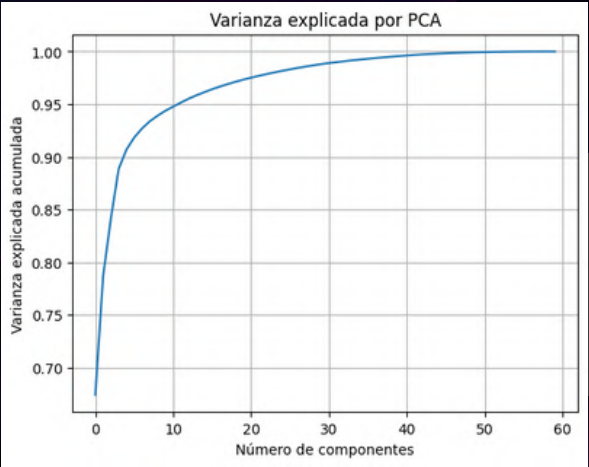
Rendimiento de svd(60)



Matriz de confusión para SVD cuando se utilizan 60 imágenes de entrenamiento

--- SVD ---
Accuracy: 0.88
Tiempo total: 0.00 s
Memoria total aprox: 1175.17 KB
Varianza explicada (k=5): 0.9869

PCA(60 img en train)



Componente 1:	Varianza = 0.674046	Acumulada = 0.674046
Componente 2:	Varianza = 0.112697	Acumulada = 0.786743
Componente 3:	Varianza = 0.054855	Acumulada = 0.841598
Componente 4:	Varianza = 0.046992	Acumulada = 0.888590
Componente 5:	Varianza = 0.018329	Acumulada = 0.906919

Rendimiento de pca60)

--- PCA ---
Accuracy: 0.80
Tiempo total: 0.08 s
Memoria total aprox: 48466.31 KB
Varianza explicada (k=5): 0.9069

Comparando xtest[0] vs X[35]:
Distancia (PCA, k=5): 3663.97
✗ Probablemente NO es la misma persona



Comparando xtest[1] vs X[35]:
Distancia (PCA, k=5): 2669.38
✓ Probablemente la MISMA persona



Comparando xtest[4] vs X[35]:
Distancia (PCA, k=5): 2274.44
✓ Probablemente la MISMA persona



Comparando xtest[5] vs X[35]:
Distancia (PCA, k=5): 4532.56
✗ Probablemente NO es la misma persona

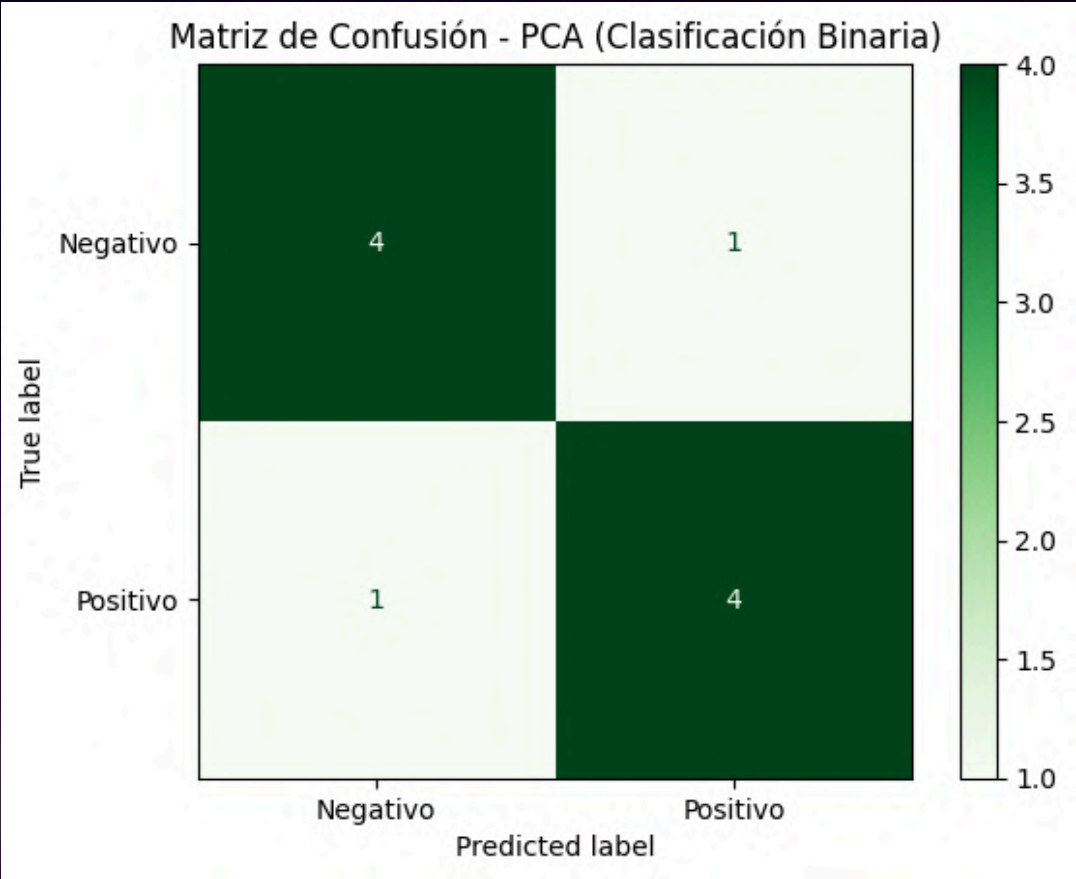


Comparando xtest[8] vs X[35]:
Distancia (PCA, k=5): 4044.33
✗ Probablemente NO es la misma persona



Comparando xtest[9] vs X[35]:
Distancia (PCA, k=5): 3402.69
✓ Probablemente la MISMA persona





BENCHMARK

Indicadores de
SVD usando 60
imágenes

Indicadores de PCA
usando 60 imágenes

```
--- SVD ---  
Accuracy: 0.80  
Tiempo total: 0.00 s  
Memoria total aprox: 1175.17 KB  
Varianza explicada (k=5): 0.9069
```

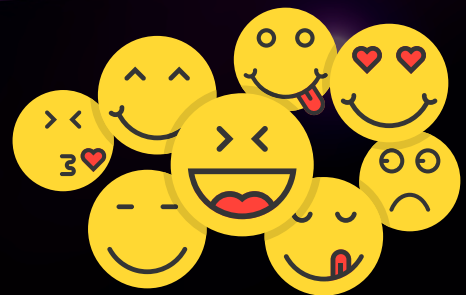
```
--- PCA ---  
Accuracy: 0.80  
Tiempo total: 0.08 s  
Memoria total aprox: 48466.31 KB  
Varianza explicada (k=5): 0.9069
```

accuracy: PCA Y SVD son iguales

Tiempo y la memoria utilizada: existe cambio en PCA y SVD (PCA puede llegar a demorar 8 veces más en la ejecución que SVD)

CONCLUSIONES Y RECOMENDACIONES

- Aplicar **SVD para reconocimiento facial es más eficiente que aplicar PCA**, ya que reduce el tiempo de procesamiento hasta 8 veces y disminuye significativamente el uso de memoria.



- Usar una base de entrenamiento con **varias imágenes por persona** (con diferentes poses y expresiones) **mejora el desempeño del sistema**, mientras que usar una sola imagen (como la del DNI) limita la capacidad de reconocimiento.

¿Qué recomendamos?

- Se recomienda que RENIEC capture múltiples fotos por ciudadano (con diferentes ángulos y expresiones) al emitir el DNI. Aunque el registro inicial tomaría más tiempo, usar SVD permite comprimir las imágenes y crear una imagen promedio, mejorando la precisión del reconocimiento facial y reduciendo errores que puedan afectar la credibilidad institucional.



GRACIAS

