# Clasificación de células de la sangre a través de aprendizaje automático

Paula Adaglio (Cs. de Datos)
Eric Lützow Holm (Biología - Computación)
Kevin Maldonado (Cs. de Datos)

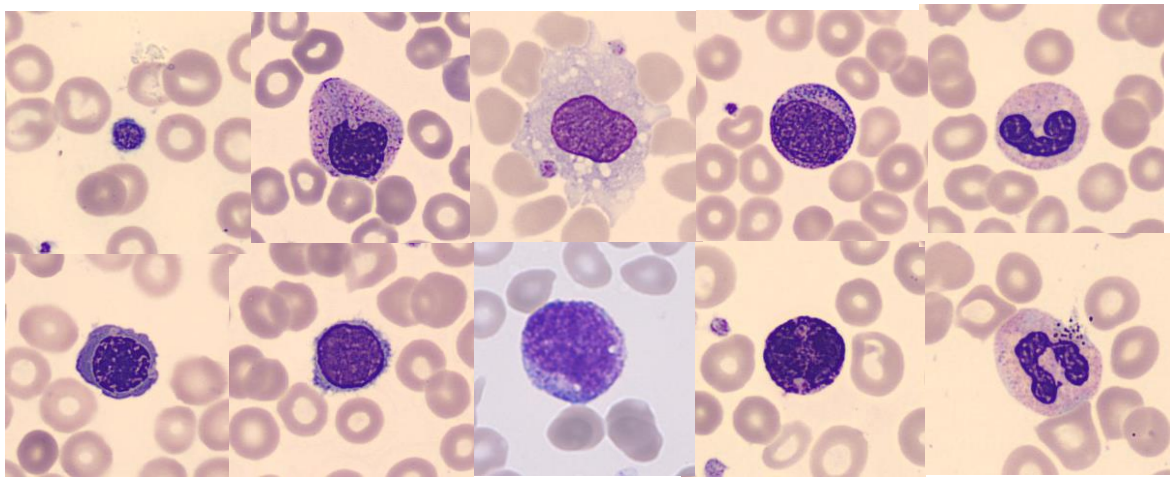
Laboratorio de datos, 1er cuatrimestre de 2021

# Agenda

1. **Objetivo**
2. **Datos**
3. **Preguntas iniciales**
1. **¿Cómo encaramos el problema?**
5. **Métodos**
6. **Resultados**
7. **Discusiones y conclusiones**

# 1.Objetivo

- Clasificar imágenes de frotis de sangre teñidas, vistas a través de un microscopio, según su forma y color.



# 2. Datos

- Aproximadamente 17 mil imágenes etiquetadas divididas en 8 clases no balanceadas.

- Casi todas las imágenes son de 363 x 360 píxeles, RGB. Por lo tanto, cada imagen se puede pensar como un vector de 392 mil dimensiones, con valores enteros entre el 0 y 255.

- Dataset descargado de Mendeley Data.

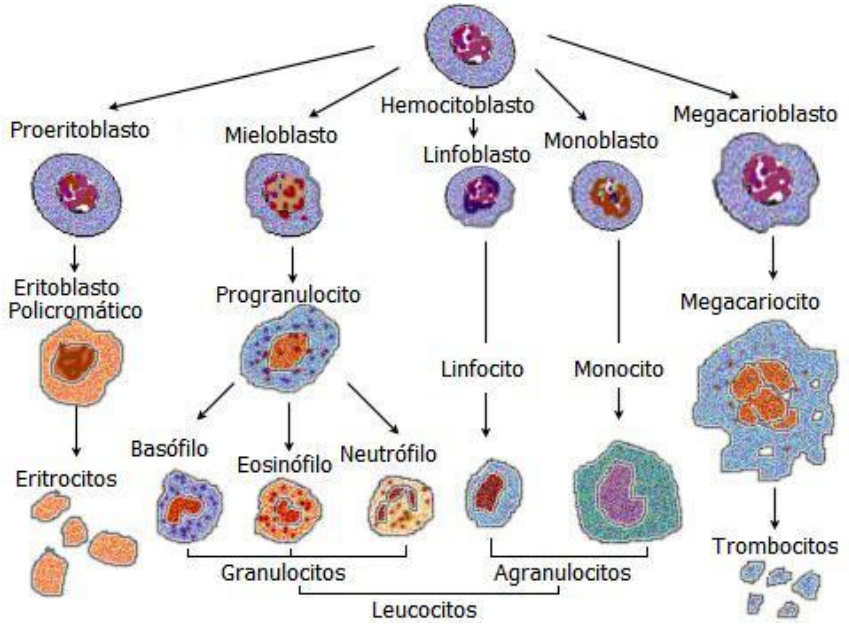
# 2. Datos

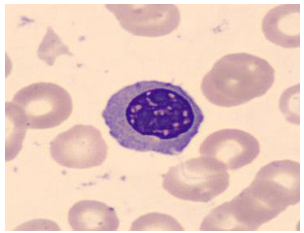
Las 8 clases son:

- Eritroblastos (precursores de glóbulos rojos)
- Trombocitos o plaquetas
- Basófilos
- Eosinófilos
- Neutrófilos
- Linfocitos
- Monocitos
- Granulocitos inmaduros

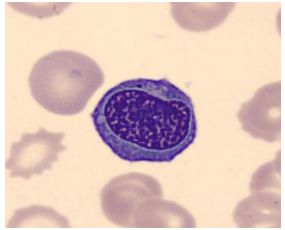
Granulocitos maduros

Son o darán glóbulos blancos

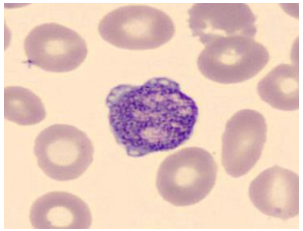


Eritroblasto



Basófilo

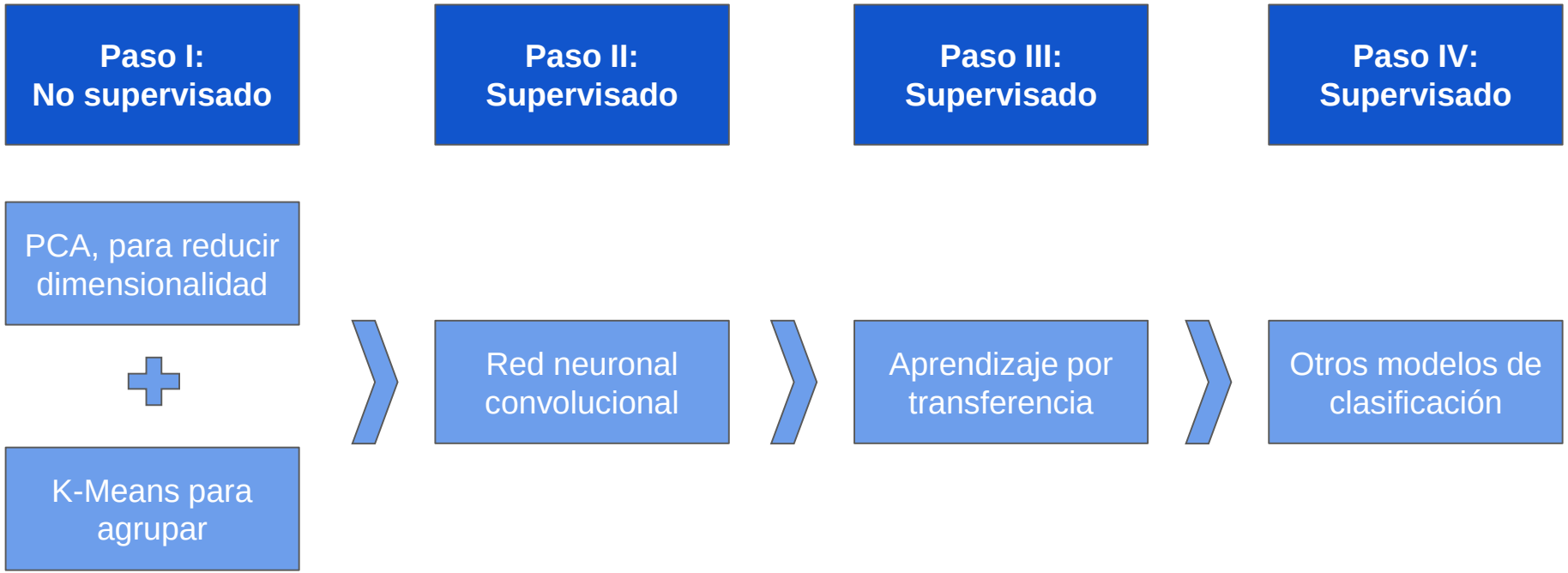


Plaqueta

# 3. Preguntas Iniciales

- ¿Podríamos separar correctamente las imágenes sin leer las etiquetas?
- ¿Podríamos entrenar a una máquina para reconocer los tipos celulares usando las etiquetas?
- ¿Cuán buenas son esas diferenciaciones o predicciones?

# 4. ¿Cómo encaramos el problema?



# 5. Métodos I

## Aprendizaje no supervisado: PCA y K-means

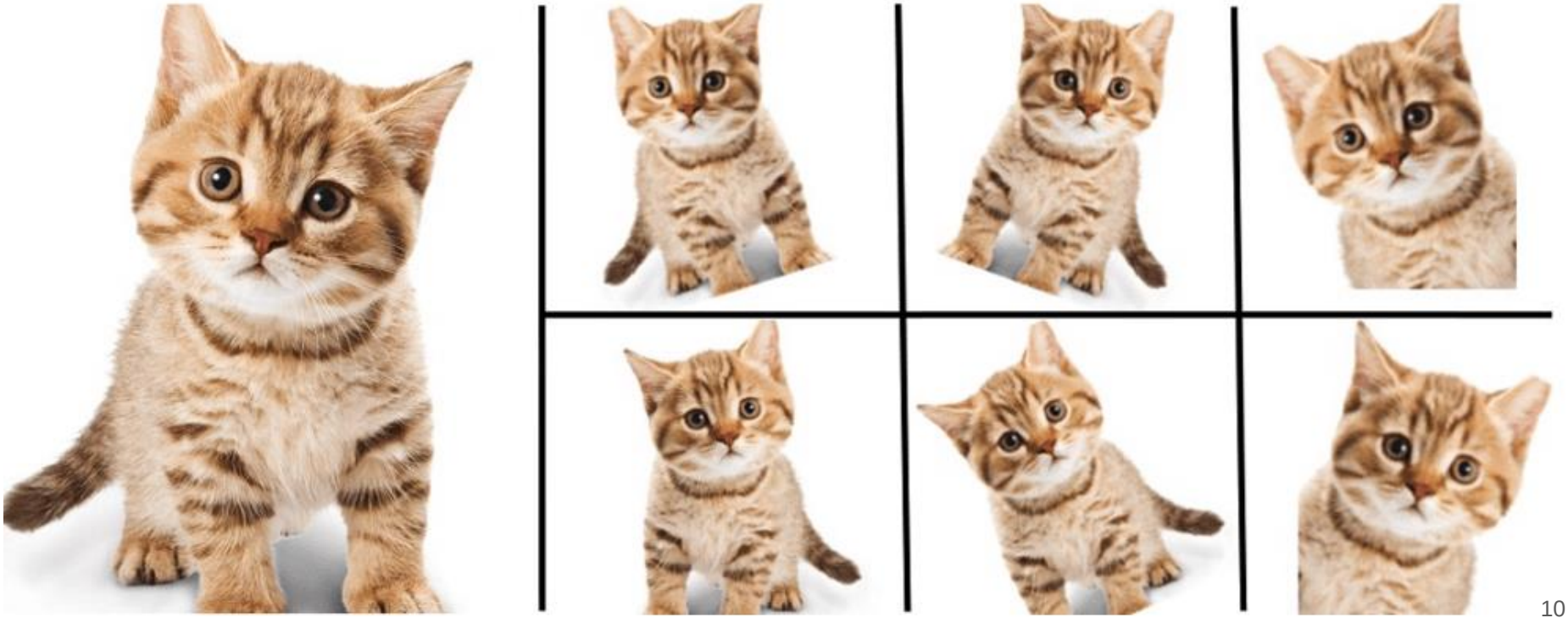
- Descargamos y extrajimos los archivos de Mendeley Data.
- Separamos los datos de manera aleatoria, con el objetivo de tener un set de entrenamiento, otro de validación y otro de prueba.
- A partir de un subconjunto de imágenes, realizamos una reducción de la dimensionalidad con PCA incremental de sklearn.
- Por último, nuevamente con la biblioteca sklearn, probamos varios modelos de agrupamiento utilizando K-means para 8 grupos.

# 5. Métodos II

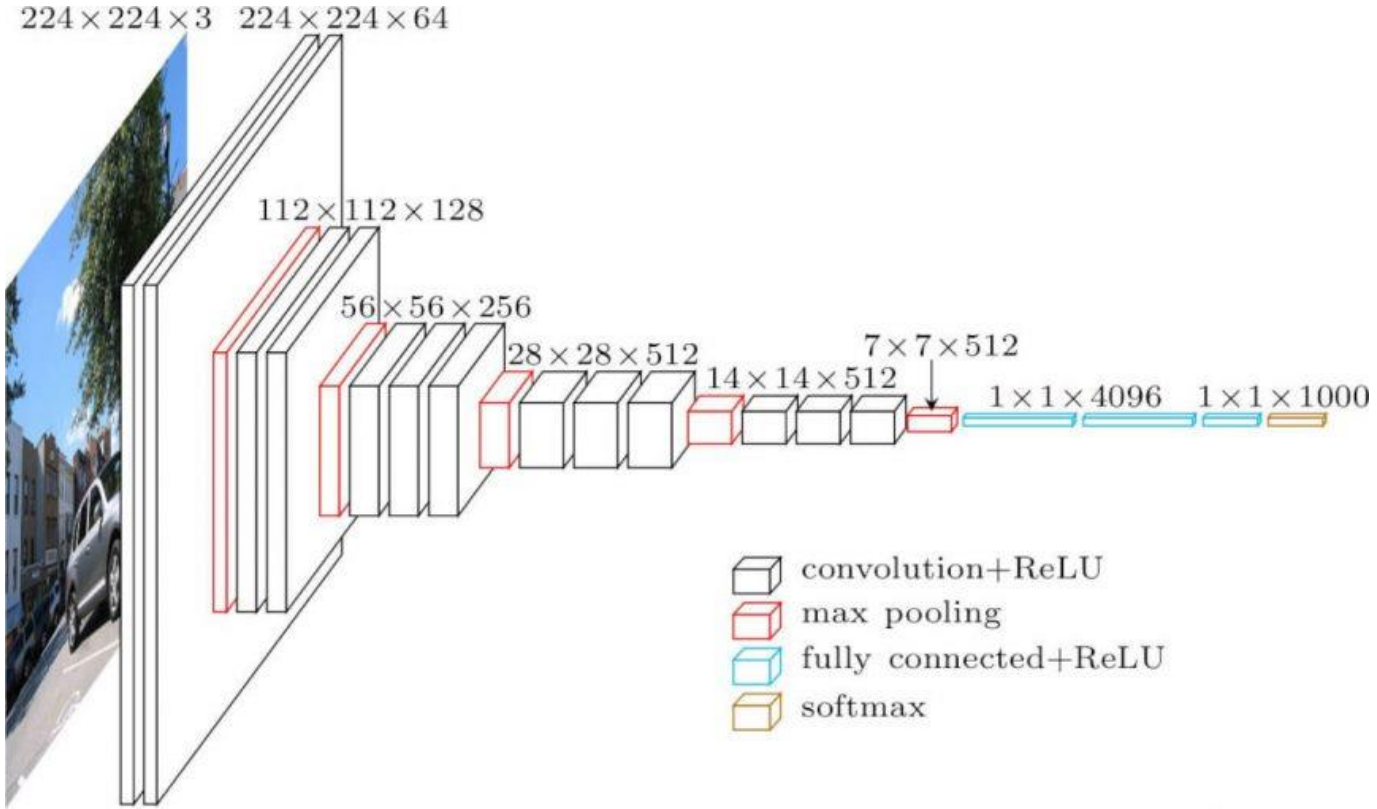
## Aprendizaje supervisado: redes neuronales y otros modelos

- Probamos una red neuronal convolucional basada en VGG16:
  - 1,6 millones de parámetros totales, todos entrenables
  - Lotes (*batches*) de 10, 10 períodos (*epochs*)
  - (Aproximadamente 2,5 horas de corrida en Google Colab)
- Luego, utilizamos una red neuronal pre-entrenada:
  - 3,2 millones de parámetros totales, 8 mil entrenables
  - *Data augmentation:* agregado de datos artificiales
  - Lotes (*batches*) de 32, 20 períodos (*epochs*) + 10 de ajuste fino
  - (Aproximadamente 1,5 + 2,5 horas de entrenamiento)
- Finalmente, comparamos estos con otros modelos: KNN, regresión logística y máquinas de soporte vectorial (SVM).

# Data augmentation

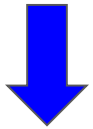


# Arquitectura de VGG 16

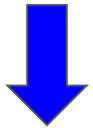


# 6. Resultados: descomposición y agrupamiento

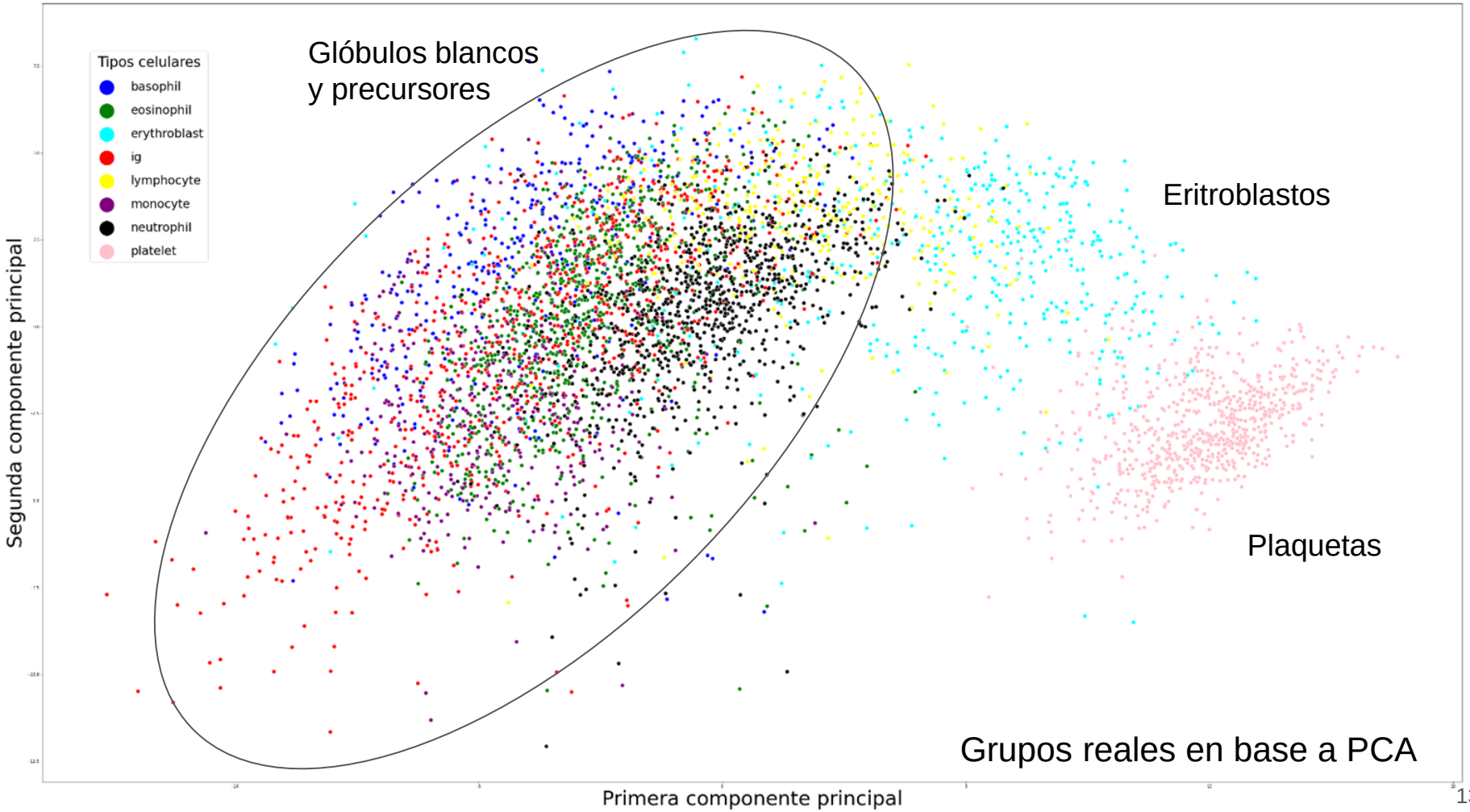
Los primeros dos componentes principales explican ~20% de la variabilidad.

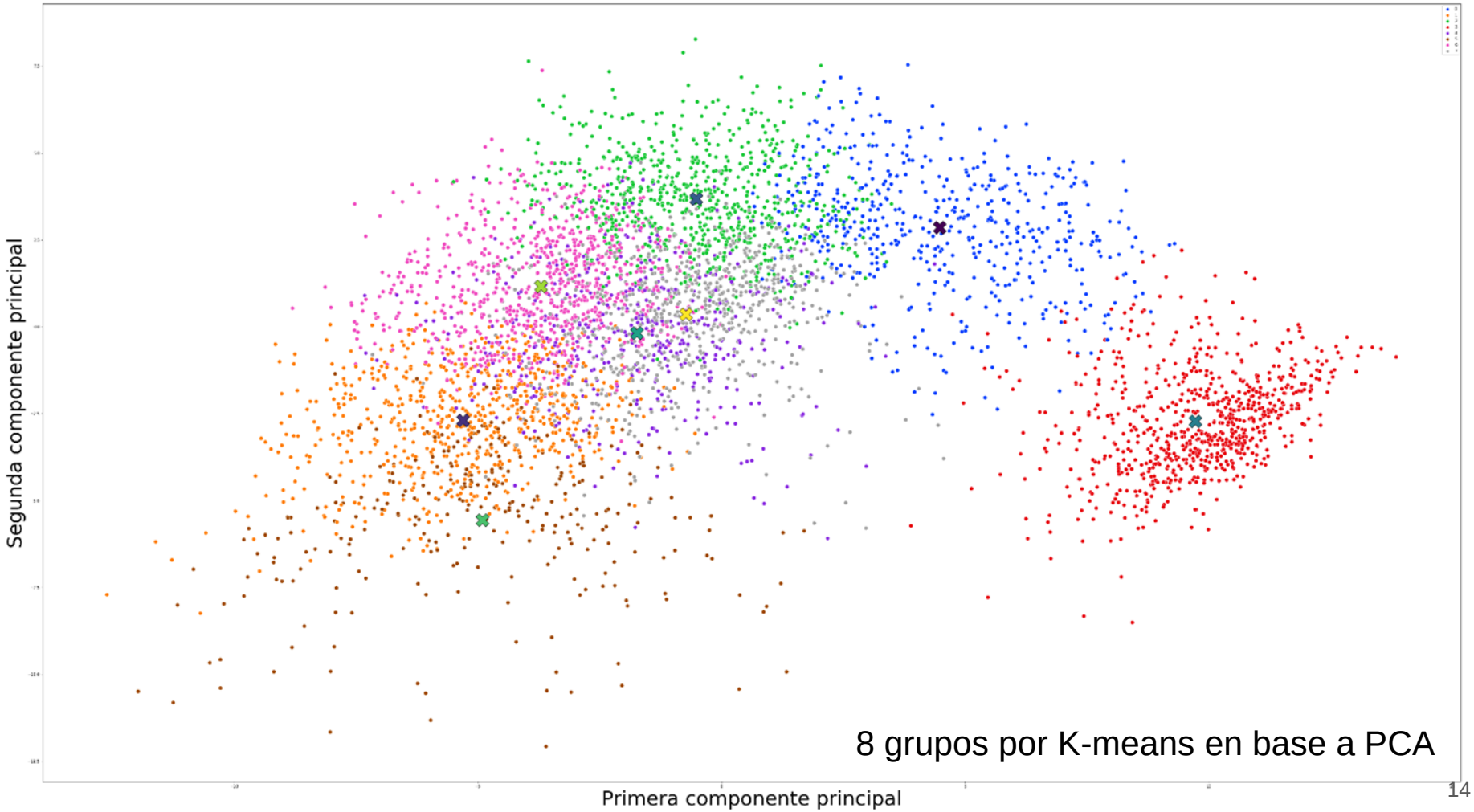


Se diferencian bien las plaquetas y los eritroblastos, pero los glóbulos blancos y sus precursores están muy solapados entre sí.



Aún con diferentes versiones de K means los agrupamientos son espurios, por lo que no alcanza con esto para diferenciarlos bien.

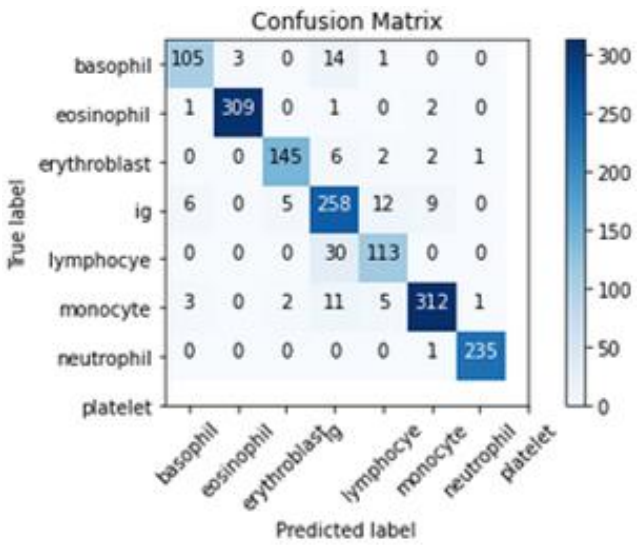

Tipos celulares
basophil
eosinophil
erythroblast
ig
lymphocyte
monocyte
neutrophil
platelet
Glóbulos blancos
y precursores
Eritroblastos
Plaquetas
Grupos reales en base a PCA
Segunda componente principal
Primera componente principal



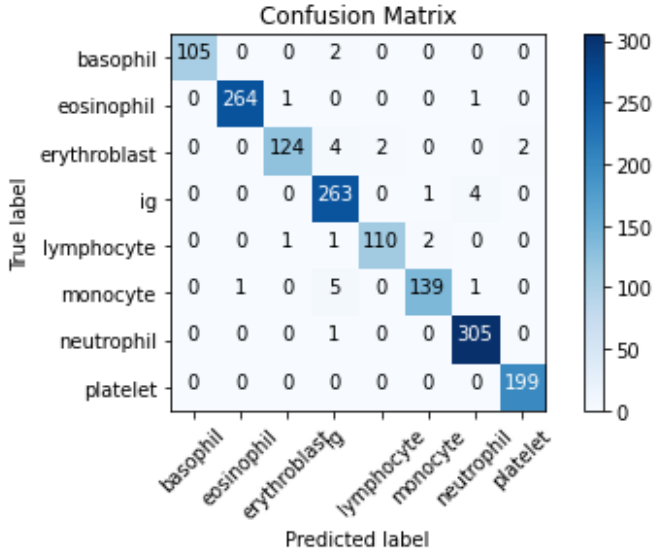

8 grupos por K-means en base a PCA

# 6. Resultados: Redes Profundas

- Primera red convolucional: 92% de aciertos.
- Segunda red convolucional: primero 96% y luego 98% de aciertos.

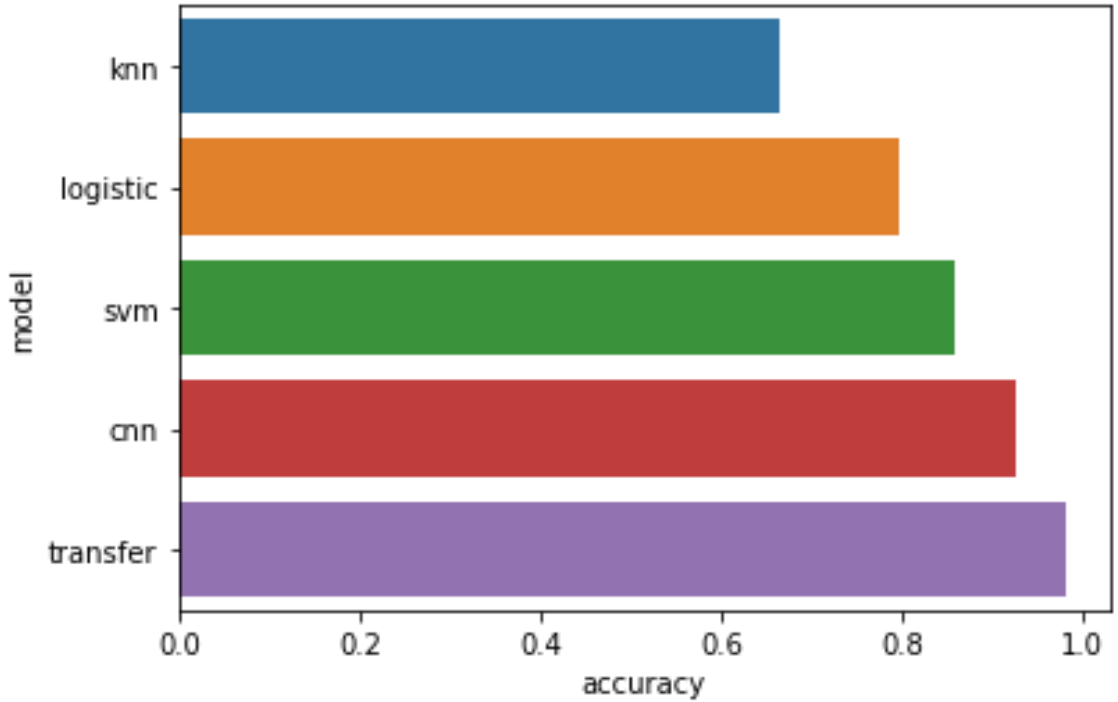



Matriz de confusión para la primera red




Matriz de confusión final para la segunda red

# 6. Resultados: Comparación de modelos



# 7. Discusiones y conclusiones

- Las primeras dos componentes principales con K-means no distinguen bien los grupos: los mezcla o los separa indebidamente.
- La red neuronal pre-entrenada fue la que mejor clasificó, pero también hay que tener en cuenta la eficiencia (tiempo/energía vs. puntaje).
- Otras cosas para probar: modificar los hiperparámetros, usar t-SNE en vez de PCA, centrar las imágenes automáticamente, random forests, otras formas de k-means, etcétera.
- Problemas al correr: conjunto muy grande, con muchos parámetros, que necesita de más poder de procesamiento que el que ofrece Google Colab.


¡Gracias!