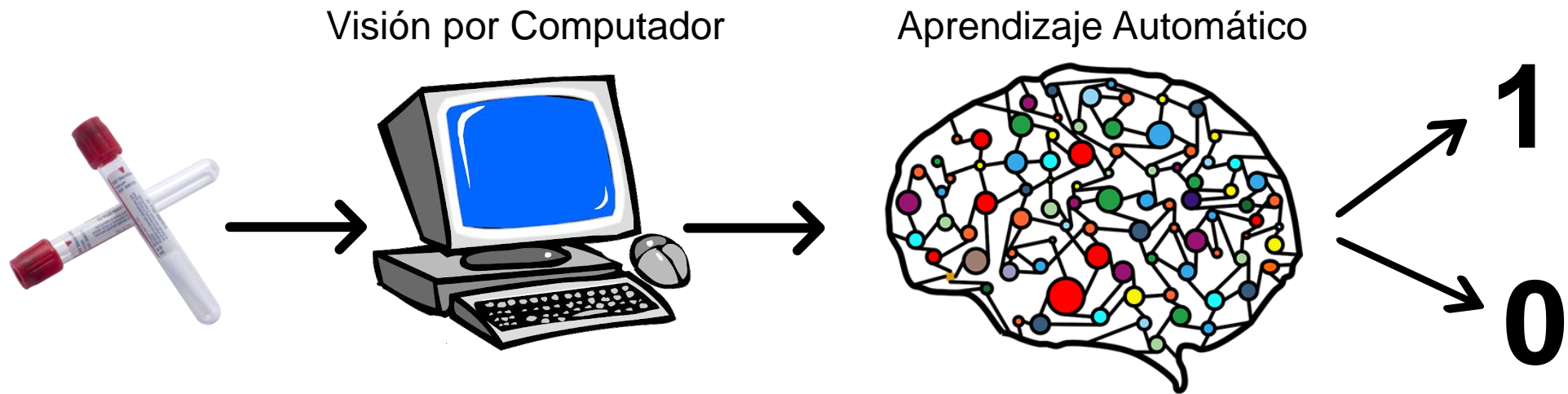


Análisis automático de imágenes de frotis de sangre periférica para diagnóstico de Leucemia

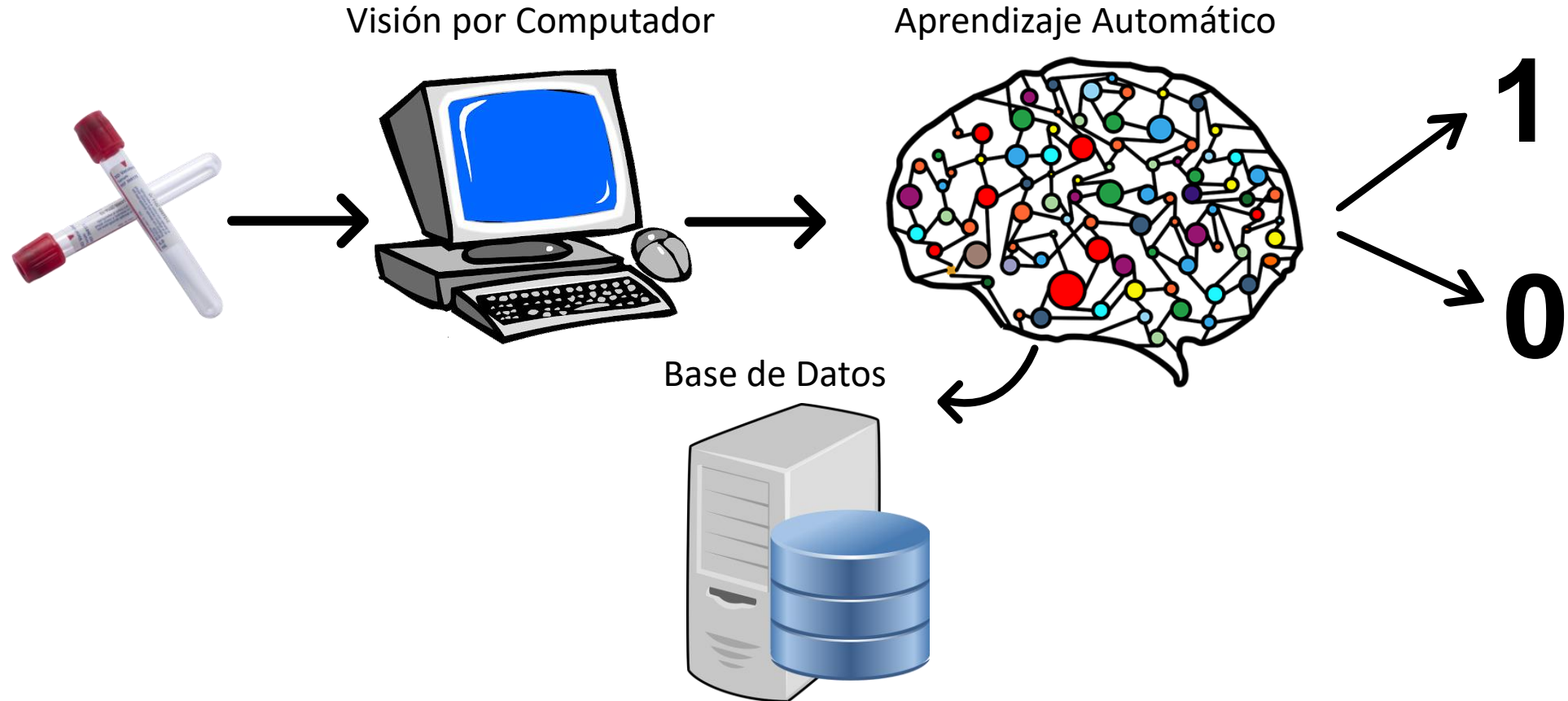
Alumno: Hodei Zia López

Director: José Antonio Sanz Delgado

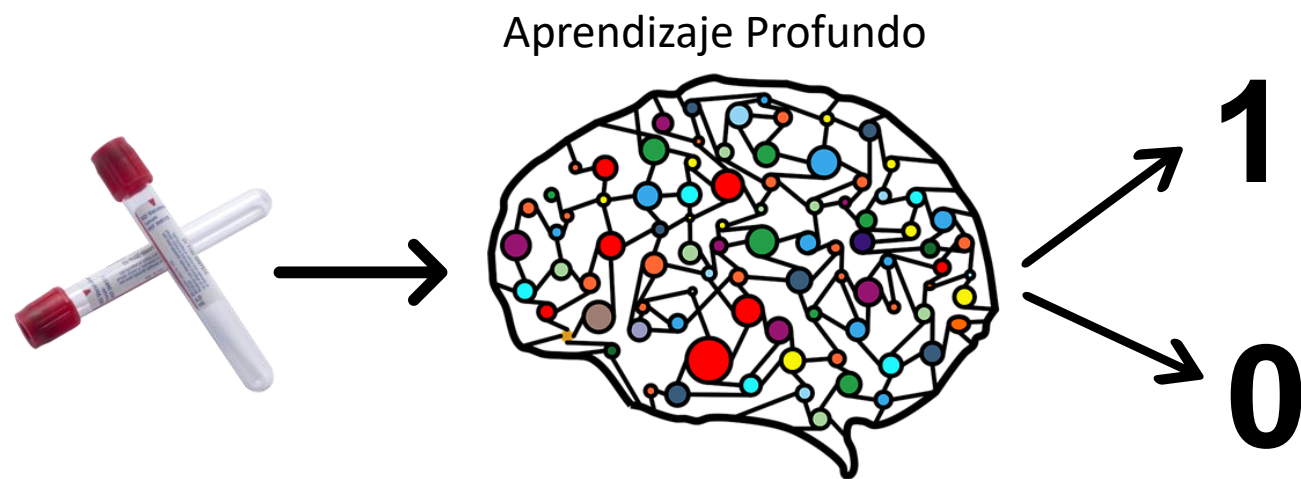
Introducción: Objetivo



Introducción: Objetivo

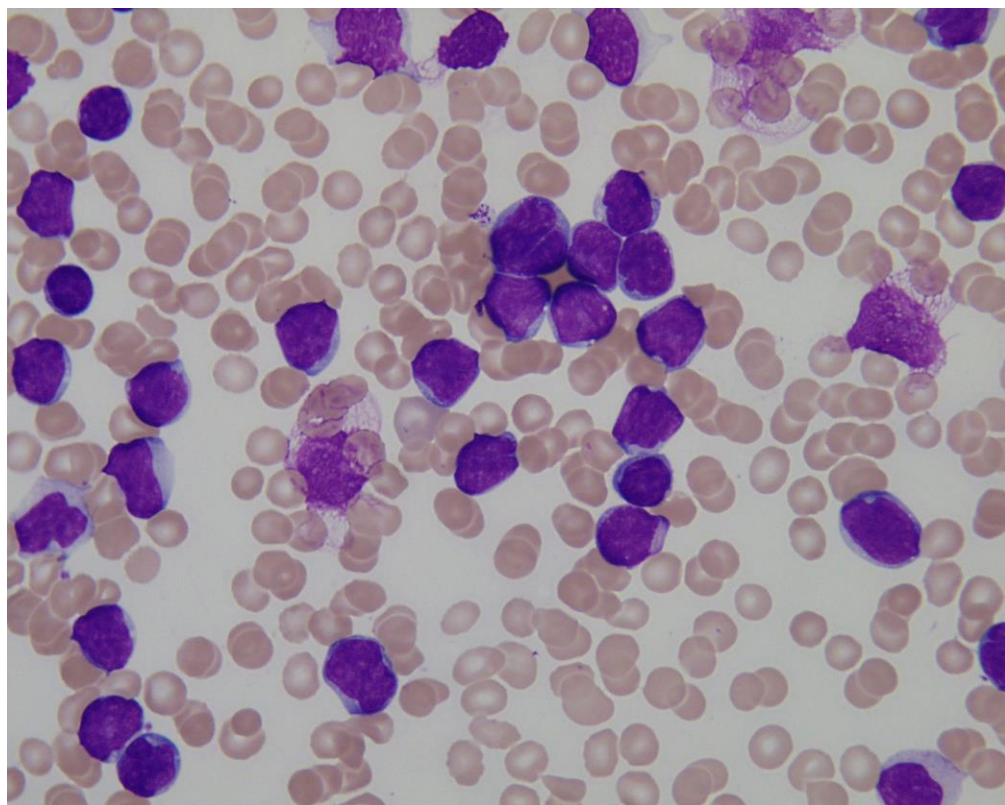


Introducción: Objetivo

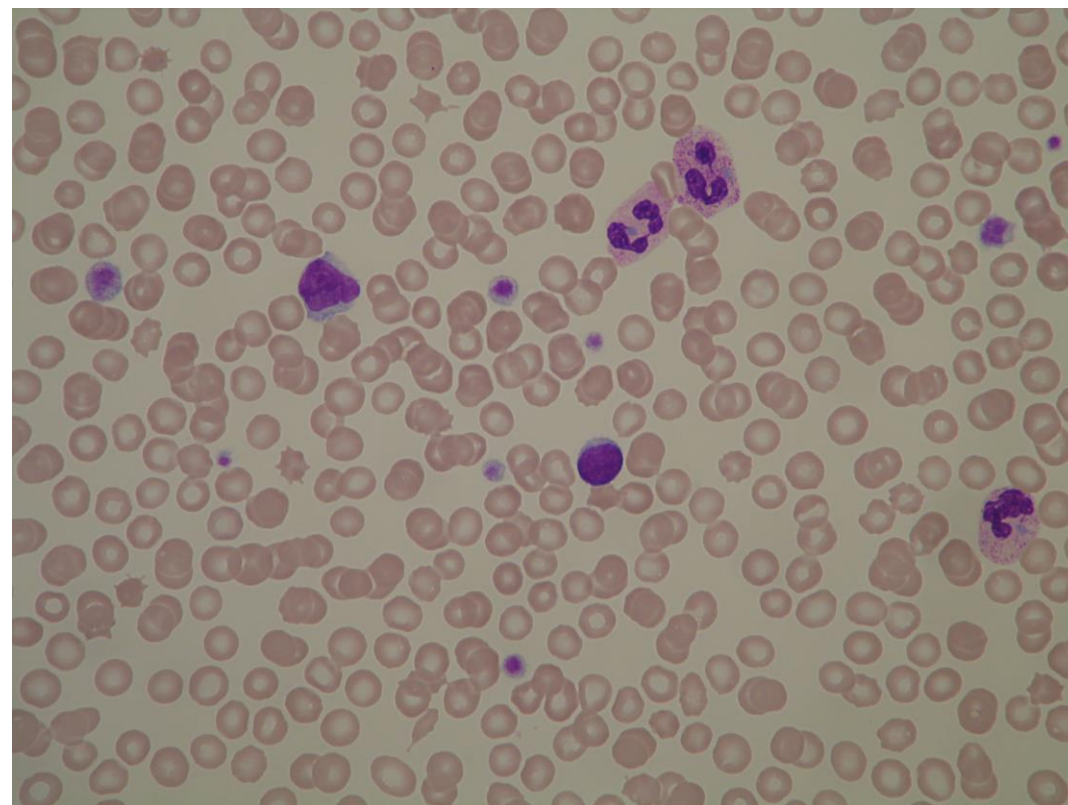


¿Qué es la Leucemia?

Muestra de un paciente enfermo

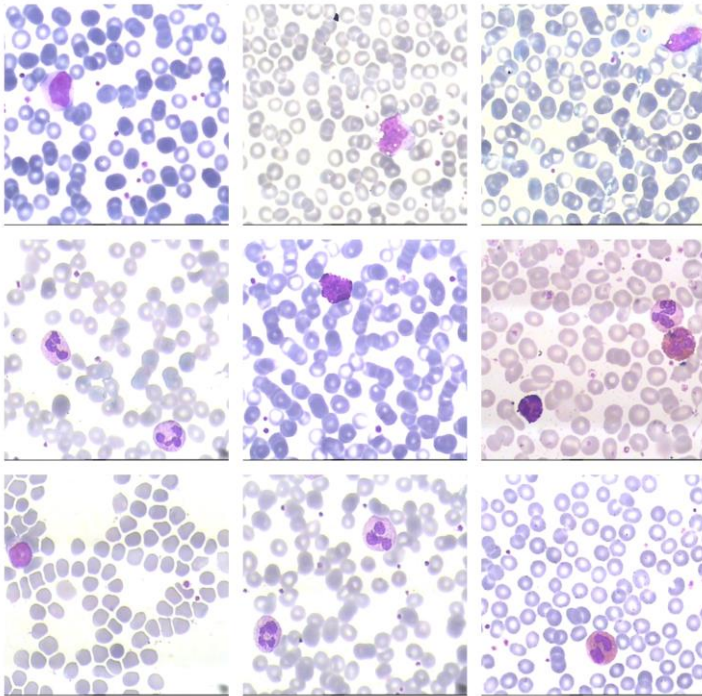


Muestra de un paciente sano

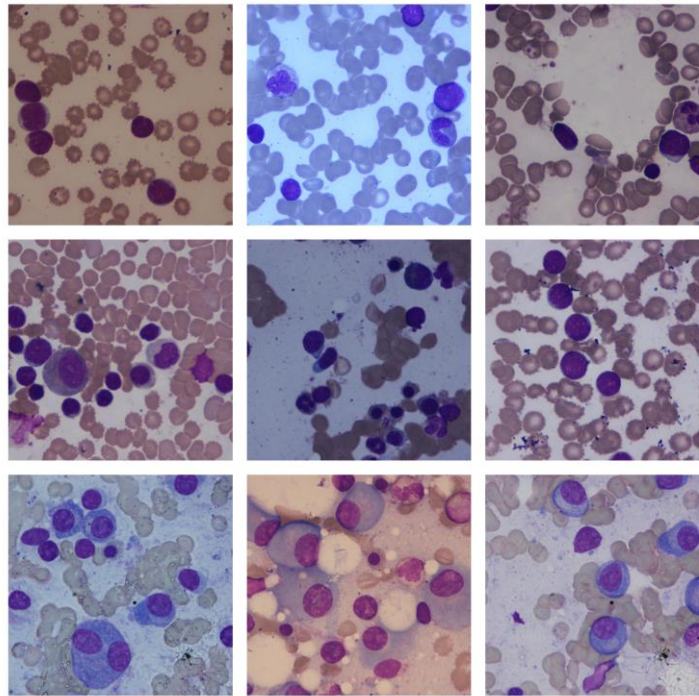


Datasets

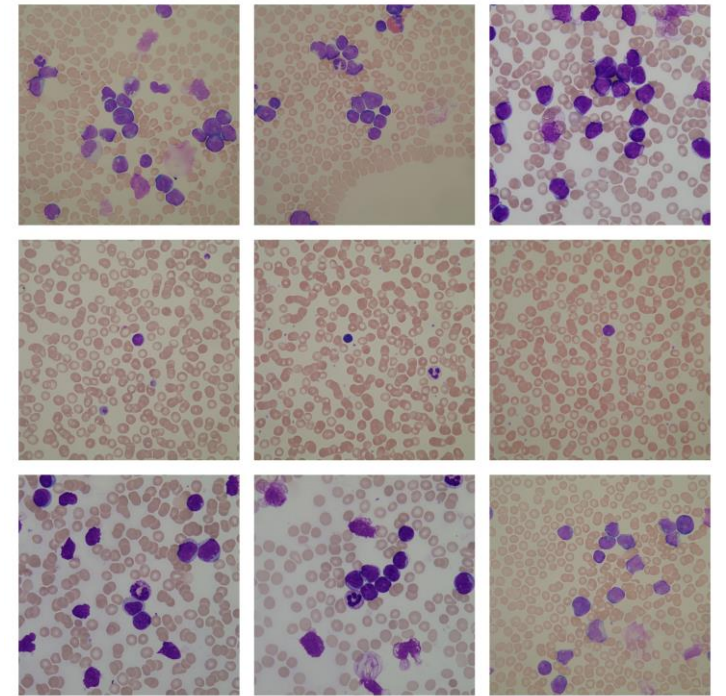
Dataset 1: LISC



Dataset 2: SN-AM



Dataset 3: ALL-IDB

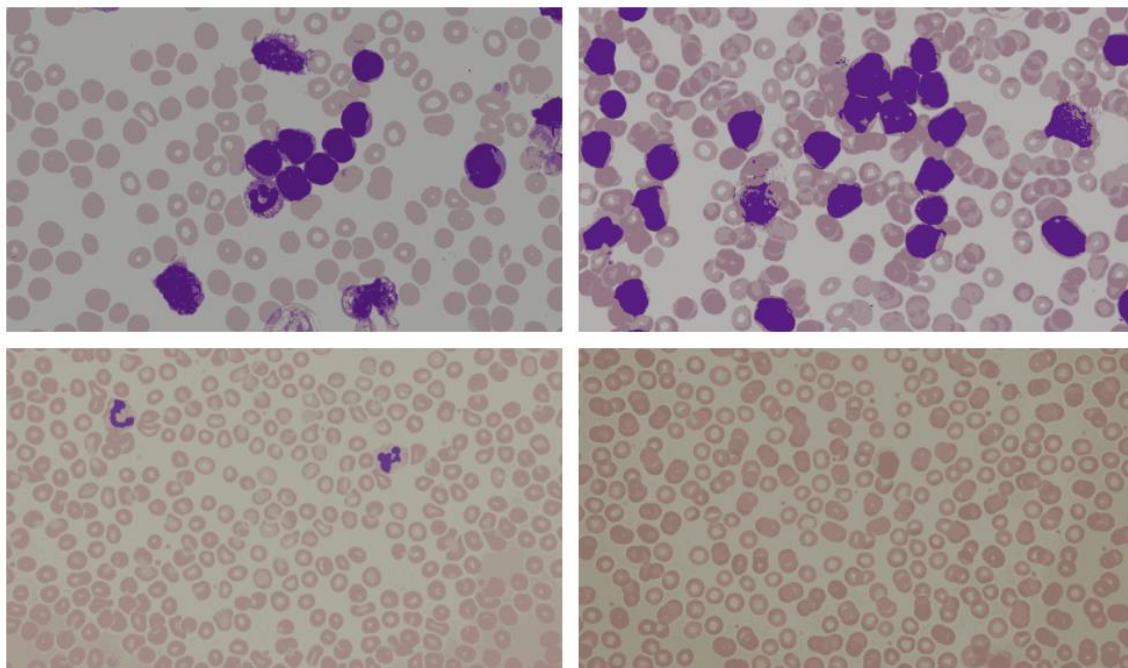


Implementación

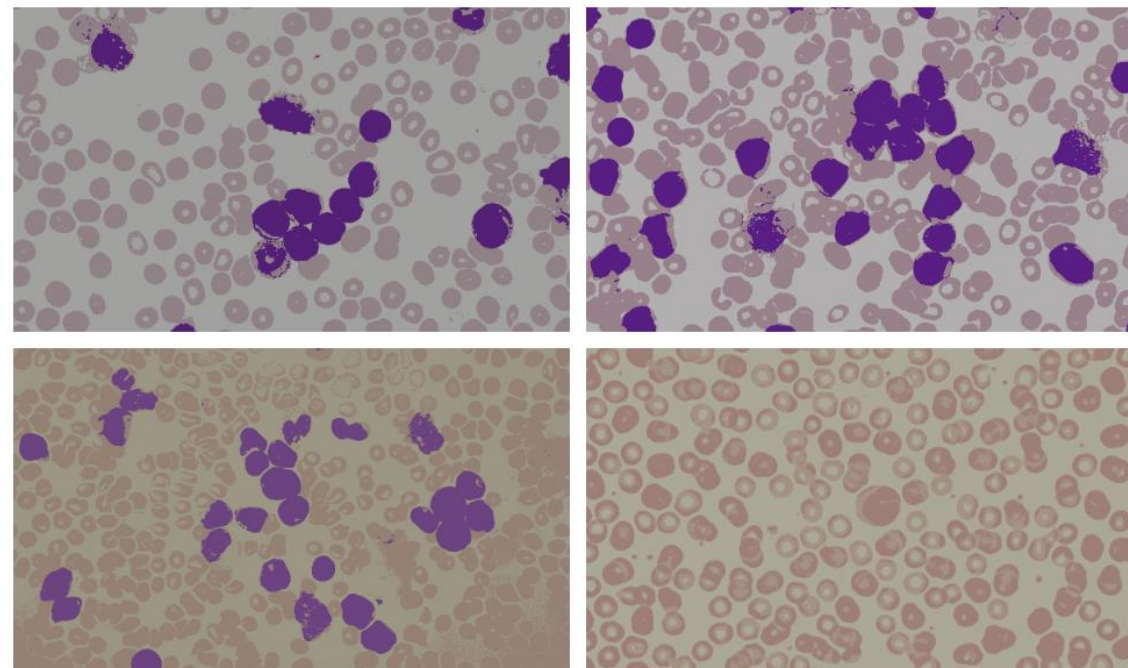
1. Fase 1: Preprocesamiento de Imágenes
2. Fase 2: Generación de Variables
3. Fase 3: Clasificación
4. Fase 4: Base de Datos e Interfaz Grafica
5. Fase 5: Deep Learning

Segmentación con K-Means

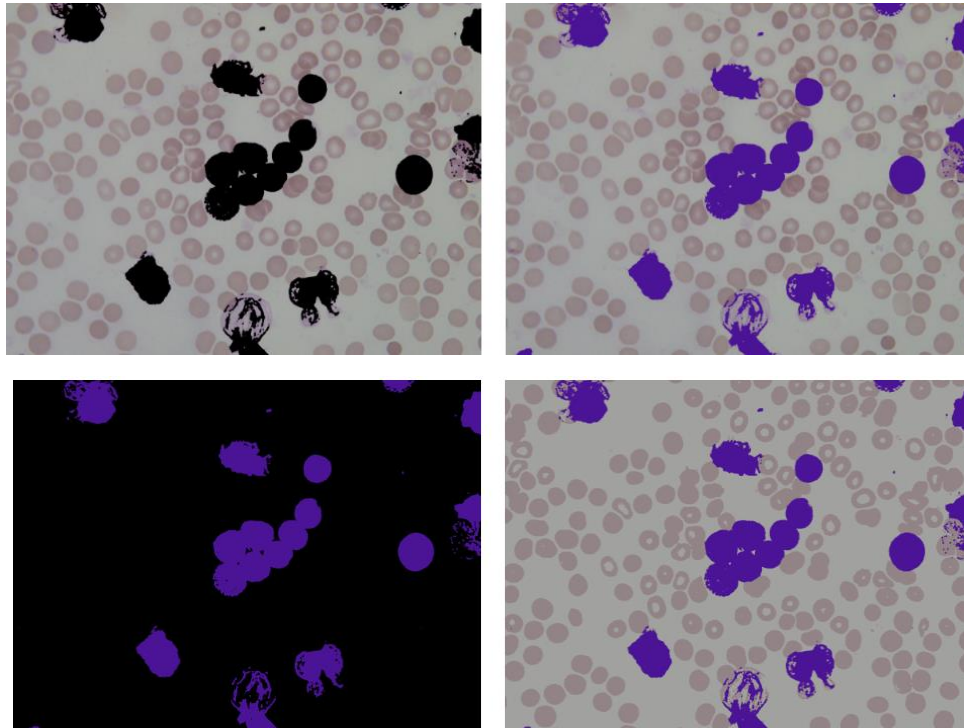
K-Means con $K = 3$



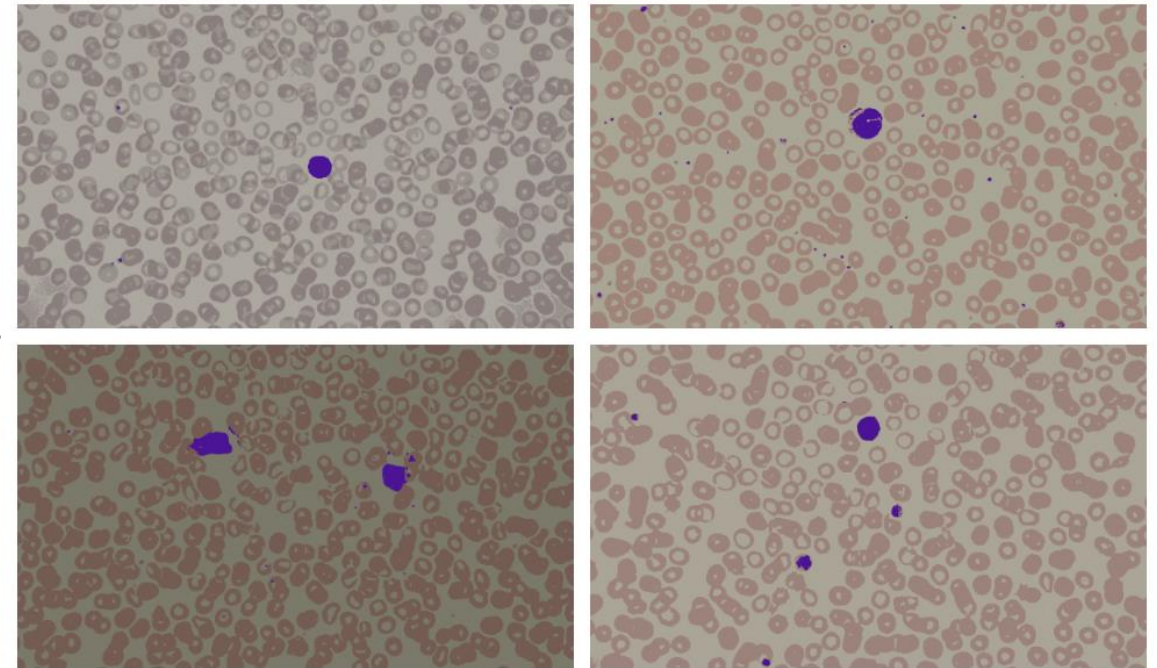
K-Means con $K = 4$



Segmentación con K-Means + Máscara

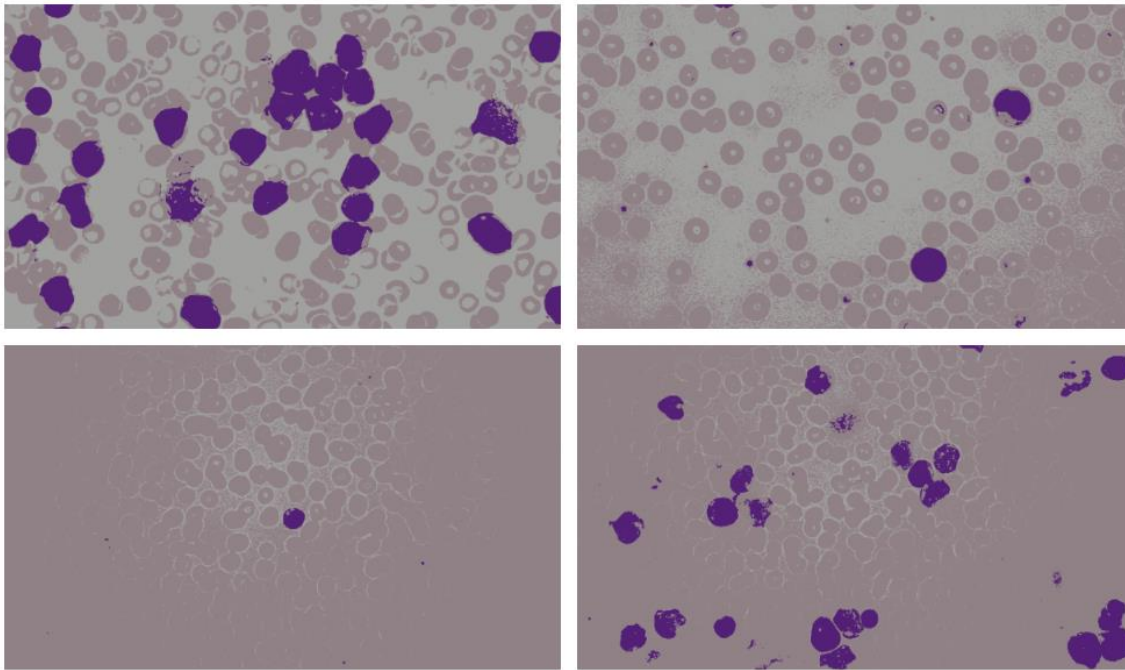


K-Means con $K = 3$ + Máscara

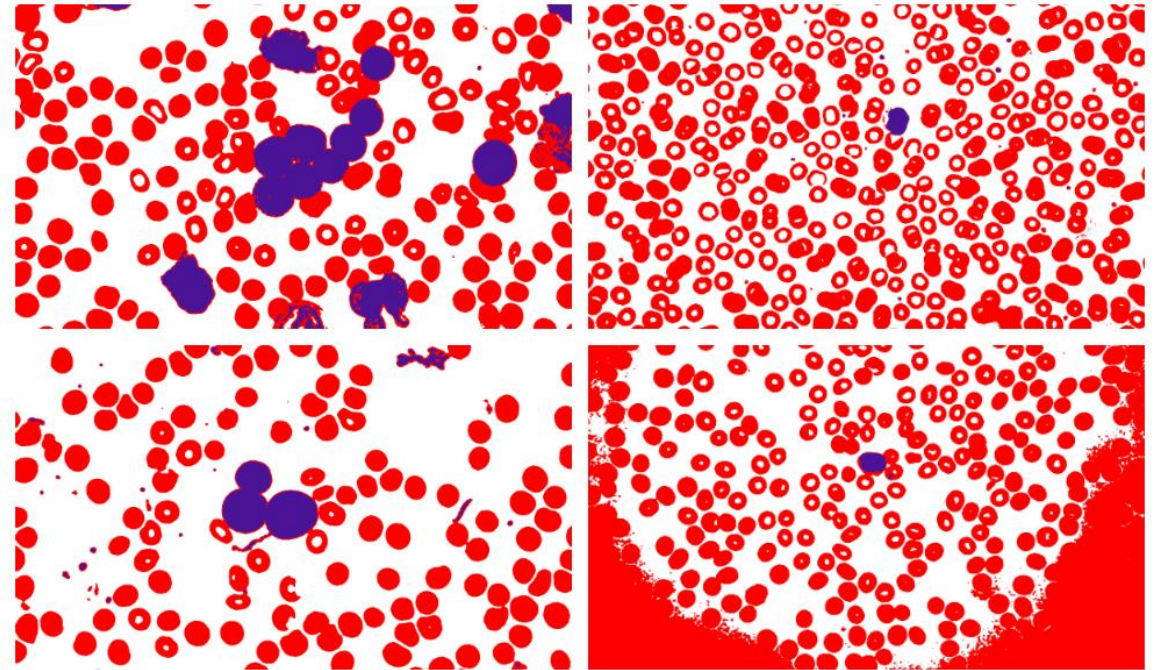


Normalización

Primera normalización

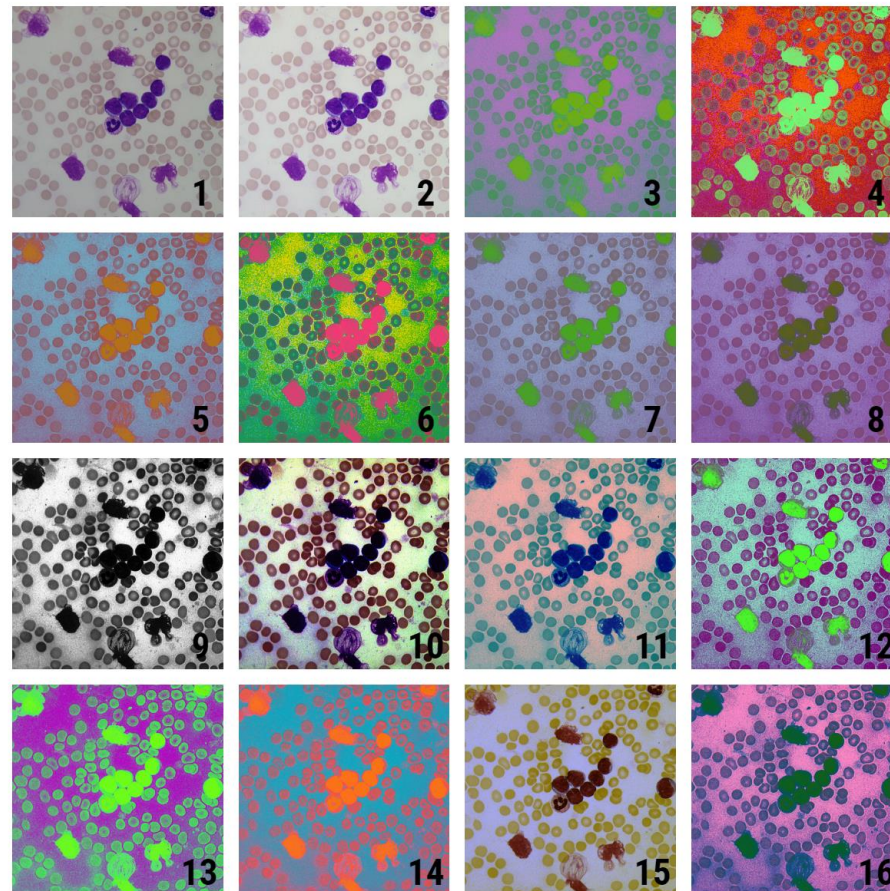


Segunda normalización



Conversión de espacios de color

Distintos espacios de color

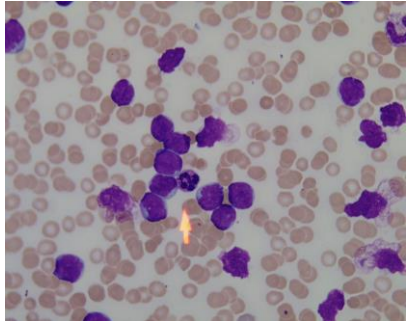


Implementación

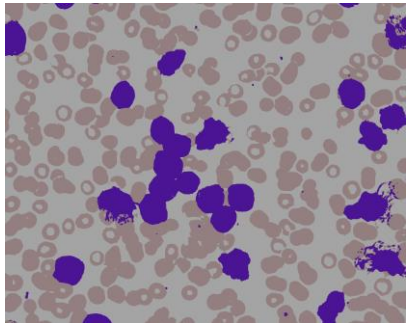
1. Fase 1: Preprocesamiento de Imágenes
2. Fase 2: Generación de Variables
3. Fase 3: Clasificación
4. Fase 4: Base de Datos e Interfaz Grafica
5. Fase 5: Deep Learning

Generación de Variables

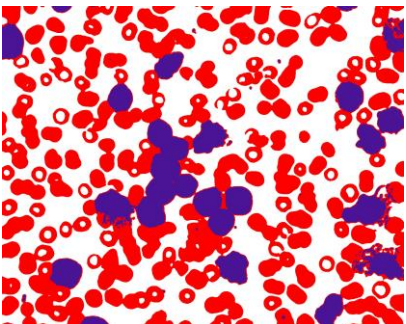
Tres datasets



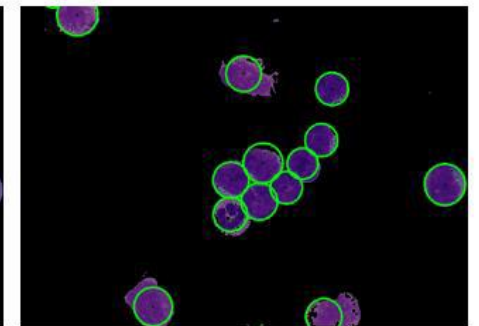
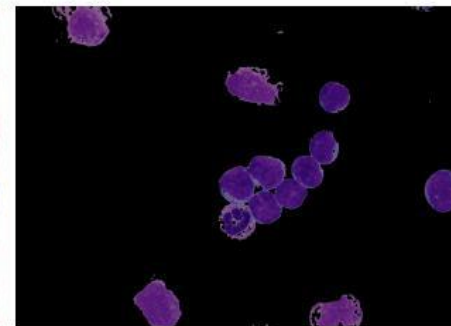
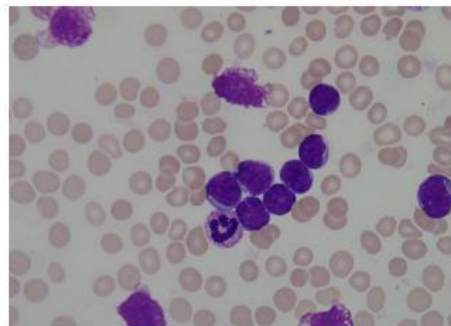
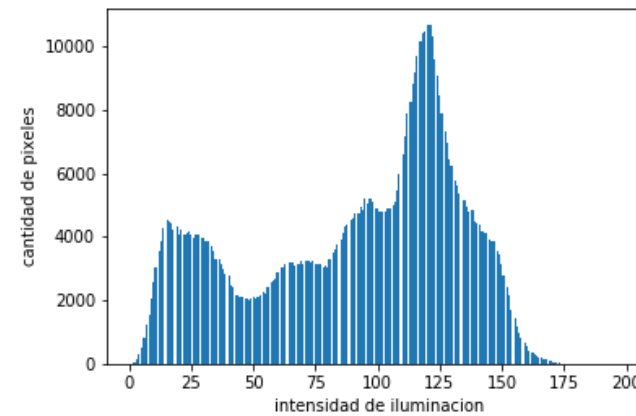
- 1 - HistMax1
- 2 - PixelesMax1
- 3 - HistMax2
- 4 - PixelesMax2
- 5 - DiferenciaPicos
- 6 - Leucocitos
- 7 - Porcentaje



- 1 - MoradosEnCluster
- 2 - HistSegMax1
- 3 - PixelesSegMax1
- 4 - HistSegMax2
- 5 - PixelesSegMax2



- 1 - Rojos
- 2 - Morados
- 3 - Blancos
- 4 - Porcentaje_Rojos
- 5 - Porcentaje_Morados
- 6 - Porcentaje_Blancos



Implementación

1. Fase 1: Preprocesamiento de Imágenes
2. Fase 2: Generación de Variables
3. Fase 3: Clasificación
4. Fase 4: Base de Datos e Interfaz Grafica
5. Fase 5: Deep Learning

Método 2

[illegible]

PORCENTAJES

[0.75930357	0.97458959	1.08274221	1.40737891	1.52454376	1.55528188
1.	1.60220861	1.70208216	2.04443336	2.09627151	2.4902463	2.52402425
2.	2.56595016	2.68469453	2.91906595	3.78655791	4.50739861	4.72759008
5.	1.09936	5.19589186	5.51836491	5.53432107	5.81148863	5.89035153
6.	2.23399019	6.33057356	6.63513541	6.88639283	6.9832027	7.48329759
7.	5.3902197	8.02398324	8.11342001	8.84616971	8.91571045	9.07487273
9.	3.765378	9.76574421	10.10868549	10.20590663	10.27392745	10.28197408
11.	0.6106639	11.11866832	11.68518662	11.80226803	12.15209365	13.00989985
16.	6.6637063]					

//////////////////////////////////NO LEUCEMIA:////////////////////////////////////

PORCENTAJES

```
[0.13431907 0.14338493 0.18216372 0.18496513 0.18748045 0.19417405
0.20659566 0.20737052 0.21495223 0.21830201 0.22062659 0.22386312
0.23671985 0.25315285 0.26079416 0.27236342 0.28121471 0.29768944
0.32799244 0.33105016 0.33481717 0.33529401 0.34563541 0.35081506
0.35377145 0.35726428 0.36792159 0.37252307 0.37496686 0.3832221
0.38486719 0.38756728 0.39290786 0.4224956 0.43116808 0.43412447
0.43450594 0.44087172 0.45258403 0.49211383 0.52893162 0.53083301
0.54278374 0.55165291 0.5599916 0.57092309 0.61069727 0.63036084
0.64865947 0.70496202 0.72976947 0.78502297 0.84604621 0.84604621
1.01607442 1.09092593 1.24292374 1.33304 1.42963529]
```

96.29%

Clasificación automática

Hiper parámetros por defecto

Clasificador	Rendimiento en train	Rendimiento en test	FScore
KNN	100.00%	96.96%	95.84%
Arboles de Decisión	100.00%	100.00%	100.00%
Random Forest	100.00%	96.96%	95.84%
Regresión Logística	100.00%	100.00%	100.00%
SVM	84.00%	84.84%	88.88%
Redes Neuronales	58.66%	45.45%	0.00%

Grid Search

Clasificador	Sin normalizar	MinMaxScaler	StandardScaler
KNN	99.09%	99.09%	99.09%
Arboles de Decisión	99.09%	99.09%	99.09%
Random Forest	100.00%	100.00%	100.00%
Regresión Logística	99.09%	99.09%	99.09%
SVM	99.09%	99.09%	100.00%
Redes Neuronales	100.00%	99.09%	99.09%

*Rendimiento en test

Selección de variables – Filtros

- Medidas de calidad utilizadas
 - Chi2
 - ANOVA
- Selección de mejores:
 - K-Best. [1,2,3,...,15,16,17] mejores variables.

Resultados:

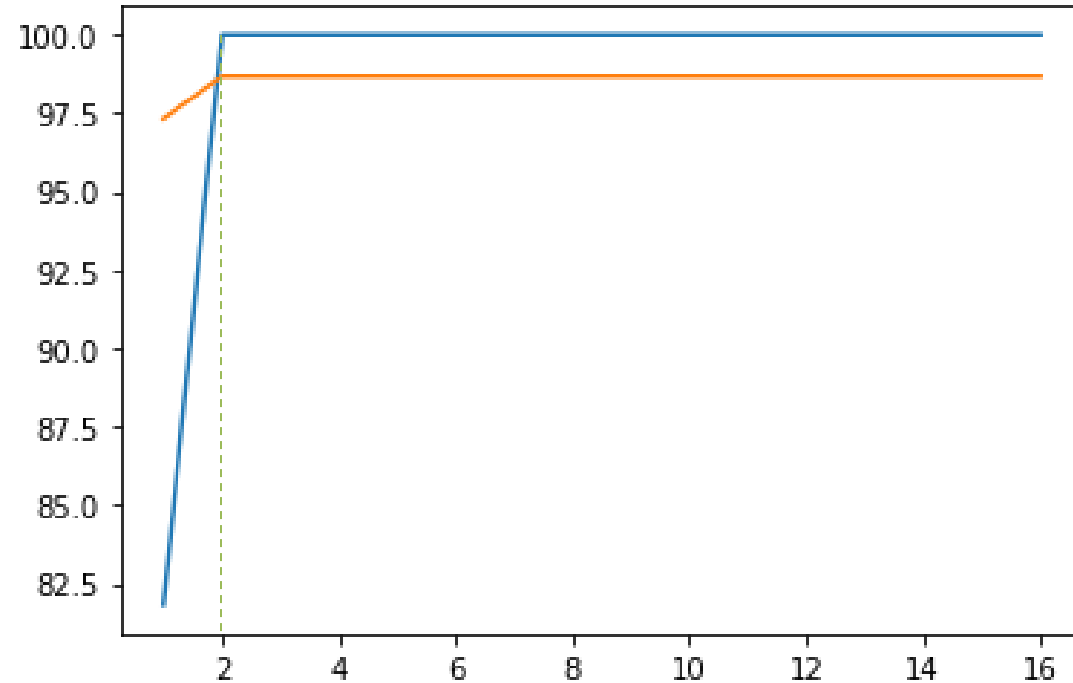
Tanto con Chi2 como con ANOVA obtenemos un **98.66%** en train y un **100.00%** en test.

Con Chi2: a partir de sus 6 mejores variables.

Con ANOVA: a partir de sus 3 mejores variables.

Selección de variables – PCA

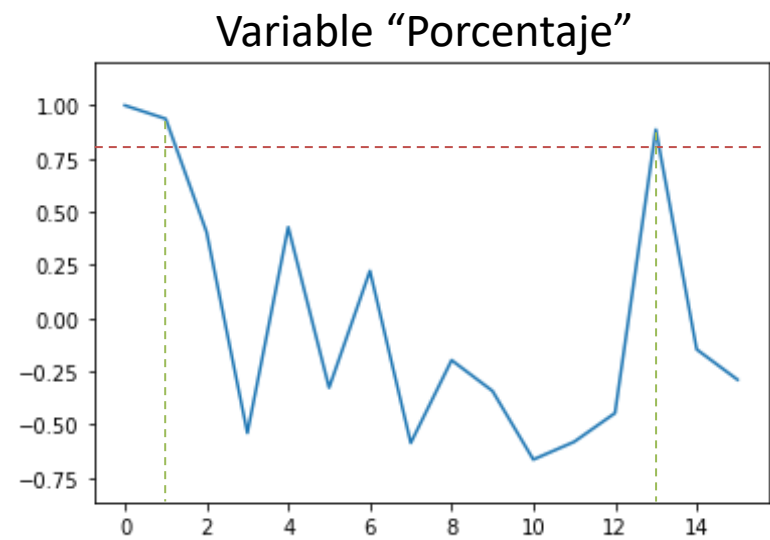
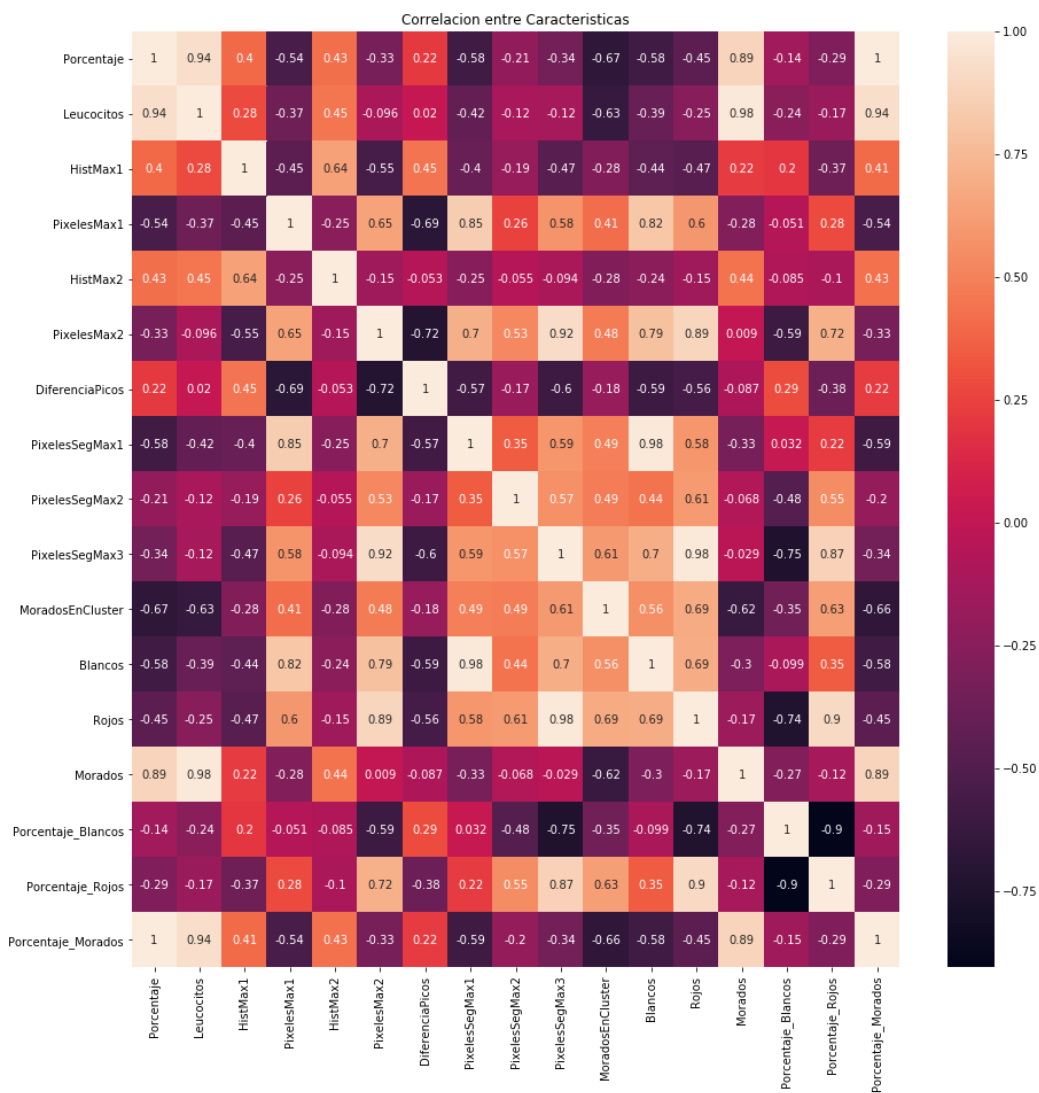
- Recorriendo las 17 variables con el PCA.



Resultados:

Con solamente dos variables podemos llegar
A un **98.66%** en train y **100.00%** test

Selección de variables – Correlaciones



Eliminamos las variables con correlación mayor al **umbral 0.8** con alguna de las variables de entrada

Resultado: Bajamos de 17 a 9 variables

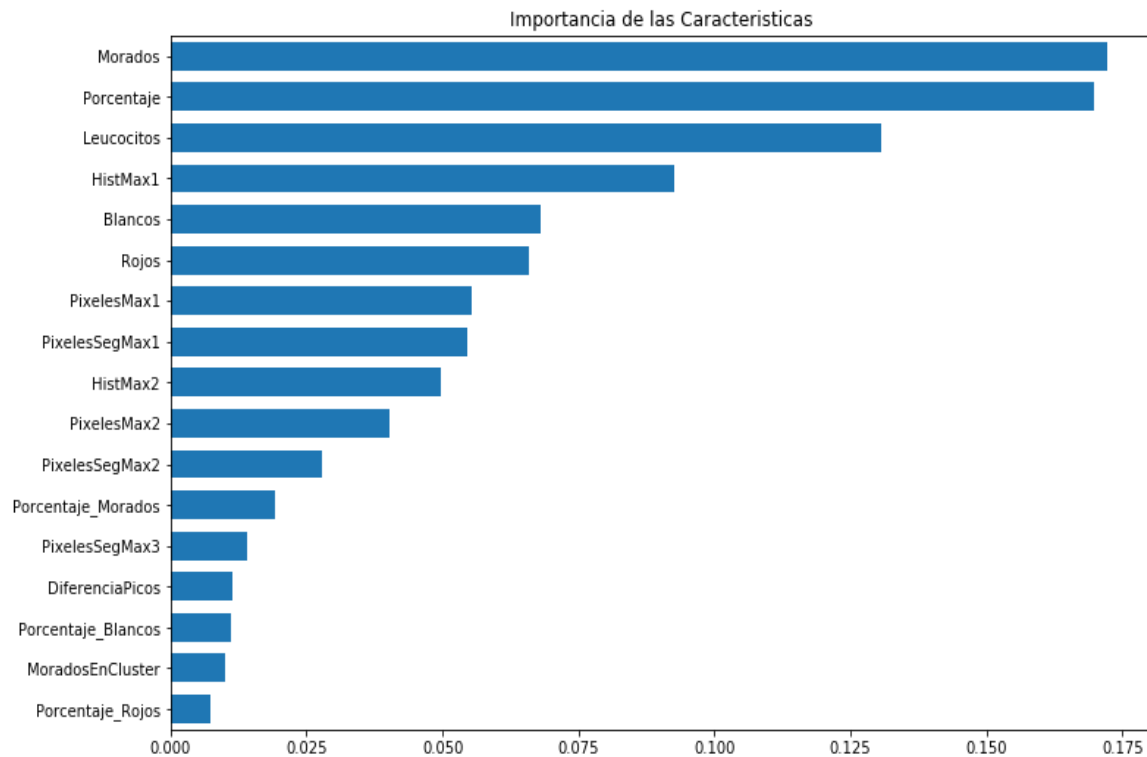
Selección de variables – Correlaciones

Grid Search con las 9 variables

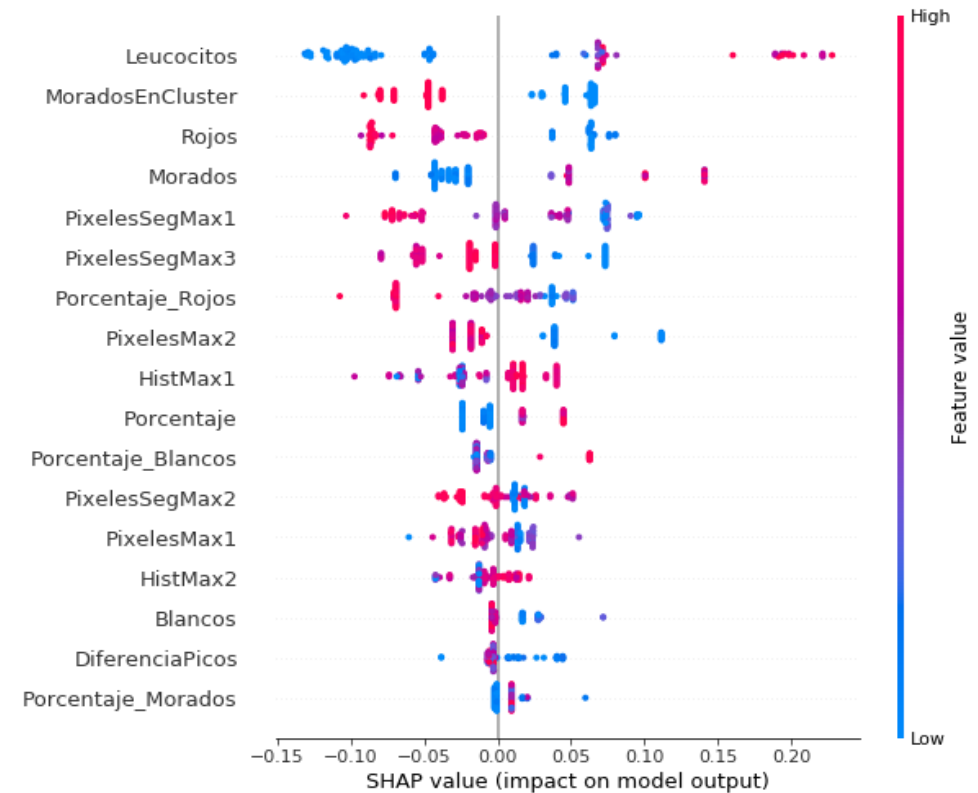
Clasificador	Sin normalizar	MinMaxScaler	StandardScaler
KNN	90.72%	99.09%	99.09%
Arboles de Decisión	98.18%	98.18%	98.18%
Random Forest	100.00%	100.00%	100.00%
Regresión Logística	90.45%	99.09%	99.09%
SVM	86.00%	99.09%	100.00%
Redes Neuronales	82.45%	99.09%	99.09%

Selección de variables – Feature Importances y SHAP Values

- Feature Importance con Random Forest



- SHAP Values

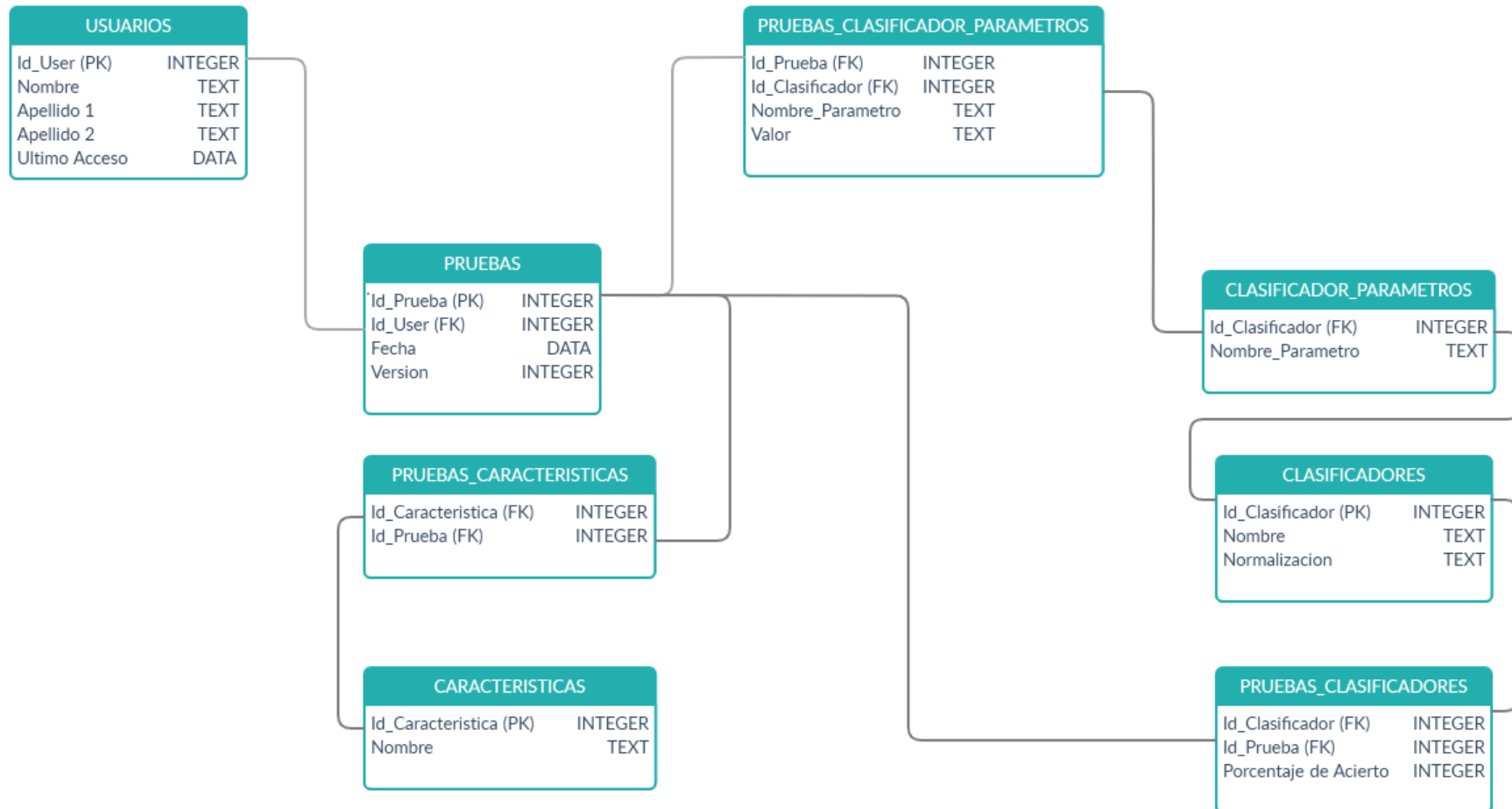


Implementación

1. Fase 1: Preprocesamiento de Imágenes
2. Fase 2: Generación de Variables
3. Fase 3: Clasificación
4. Fase 4: Base de Datos e Interfaz Grafica
5. Fase 5: Deep Learning

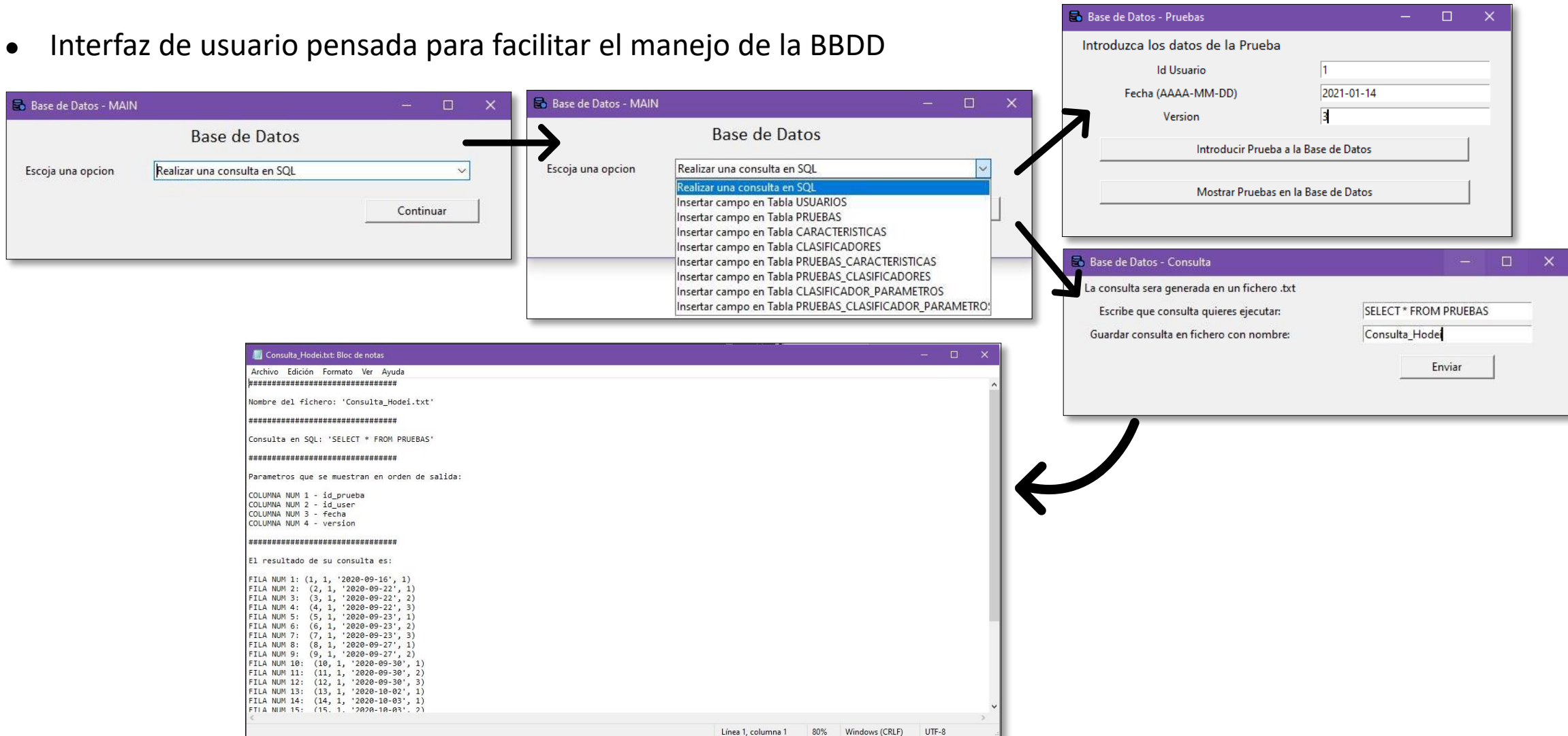
Base de Datos

- Arquitectura de la BBDD creada con más de 2500 instancias



Interfaz Gráfica

- Interfaz de usuario pensada para facilitar el manejo de la BBDD



Implementación

1. Fase 1: Preprocesamiento de Imágenes
2. Fase 2: Generación de Variables
3. Fase 3: Clasificación
4. Fase 4: Base de Datos e Interfaz Grafica
5. Fase 5: Deep Learning

Deep Learning

- Reducción de imágenes
 - 1944x1458 (75.00% de resolución respecto a la original)
 - 1458x1093 (56.25% de resolución respecto a la original)
 - 1093x820 (42.16% de resolución respecto a la original)
 - 820x615 (31.63% de resolución respecto a la original)
 - 615x461 (23.72% de resolución respecto a la original)

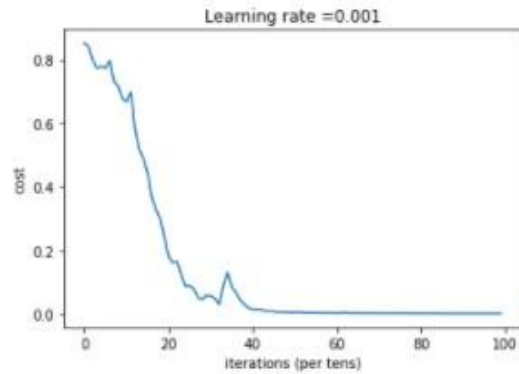
Neural Networks and Deep Learning: <https://coursera.org/share/cb49c81b3212ddc608416c6522fe47f4>

Improving Deep Neural Networks: Hyperparameter Tuning, Regularization and Optimization: <https://coursera.org/share/35de54f21a76ec1e2b5357ef751dcfac>

Structuring Machine Learning Projects: <https://coursera.org/share/00db9e5354d5d173a0b365449f159118>

Deep Learning

- 615x461



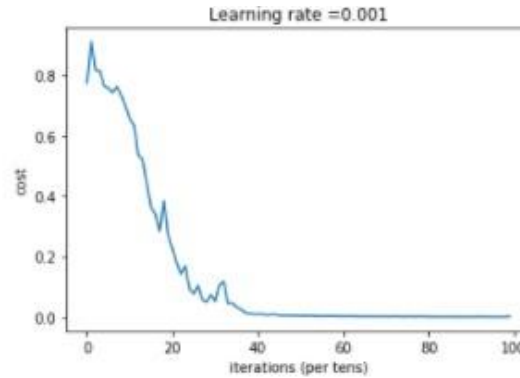
Tensor("Mean_1:0", shape=(), dtype=float32)
Train Accuracy: 1.0
Test Accuracy: 1.0

Resultado:

100.00% en Train

100.00% en Test

- 820x615



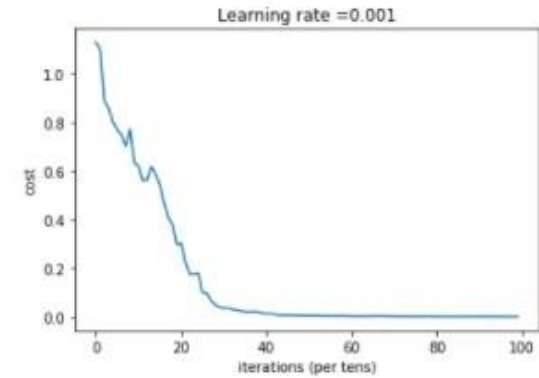
Tensor("Mean_1:0", shape=(), dtype=float32)
Train Accuracy: 1.0
Test Accuracy: 0.93939394

Resultado:

100.00% en Train

93.93% en Test

- 1093x820



Tensor("Mean_1:0", shape=(), dtype=float32)
Train Accuracy: 1.0
Test Accuracy: 0.969697

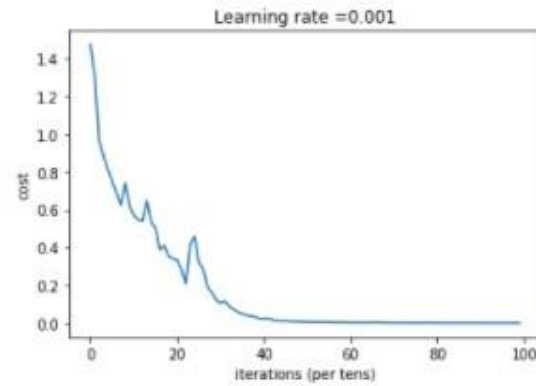
Resultado:

100.00% en Train

96.96% en Test

Deep Learning

- 1458x1093

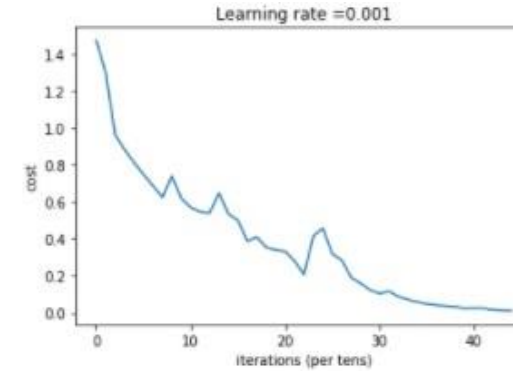


Tensor("Mean_1:0", shape=(), dtype=float32)
Train Accuracy: 1.0
Test Accuracy: 0.969697

Resultado:

100.00% en Train
96.96% en Test

- 1944x1458



Tensor("Mean_1:0", shape=(), dtype=float32)
Train Accuracy: 1.0
Test Accuracy: 0.969697

Resultado:

100.00% en Train
96.96% en Test

Conclusiones y líneas futuras

- No todo es el porcentaje de acierto.
 - Deep Learning funciona muy bien con imágenes.
 - No todos los errores son iguales.
-
- Facilitar el acceso a estos análisis a todo el mundo.
 - Detectar la enfermedad antes de que se expanda.
-
- Detectar que tipo de leucocitos tienen mayor relación con la enfermedad.
 - Reducir la combinatoria de los hiper parámetros del Grid Search.
 - Aumentar el dataset con muestras reales.
 - Trabajar sobre el terreno para mejorar el proyecto e intentar aplicarlo.

