



DermaScan AI: Plataforma de Clasificación de Enfermedades Cutáneas con IA

Julio, 2025

Universidad Politécnica Salesiana



Más que formar
PROFESIONALES,
Transformamos Vidas



Presentación de Resultados

Universidad Politécnica Salesiana



Más que formar
Profesionales,
Transformamos Vidas



Felipe Peralta

Andrés Torres

Kevin Chipantiza

Samantha Suquilanda

**Más que formar
Profesionales,
*Transformamos Vidas***

1.) Problema planteado

Incremento de casos de cáncer de piel en Ecuador (25.717 casos reportados entre 2006-2015).

Aumento global de la incidencia de melanoma y otros cánceres cutáneos.

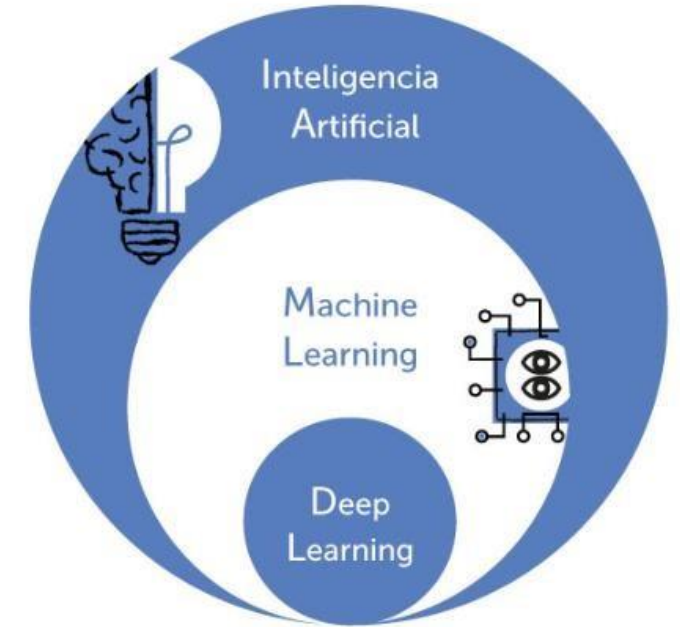
Afecta a mujeres entre 50-70 años y también a jóvenes desde los 30 años.

Falta de acceso a diagnósticos tempranos y especializados.

Exposición solar, hábitos ocupacionales y falta de educación preventiva.

2.) Fundamentos teóricos aplicados en DermaScan AI

En el desarrollo de DermaScan AI, aplicamos conceptos clave de **Inteligencia Artificial (IA)**, **Machine Learning (ML)** y **Deep Learning (DL)** :



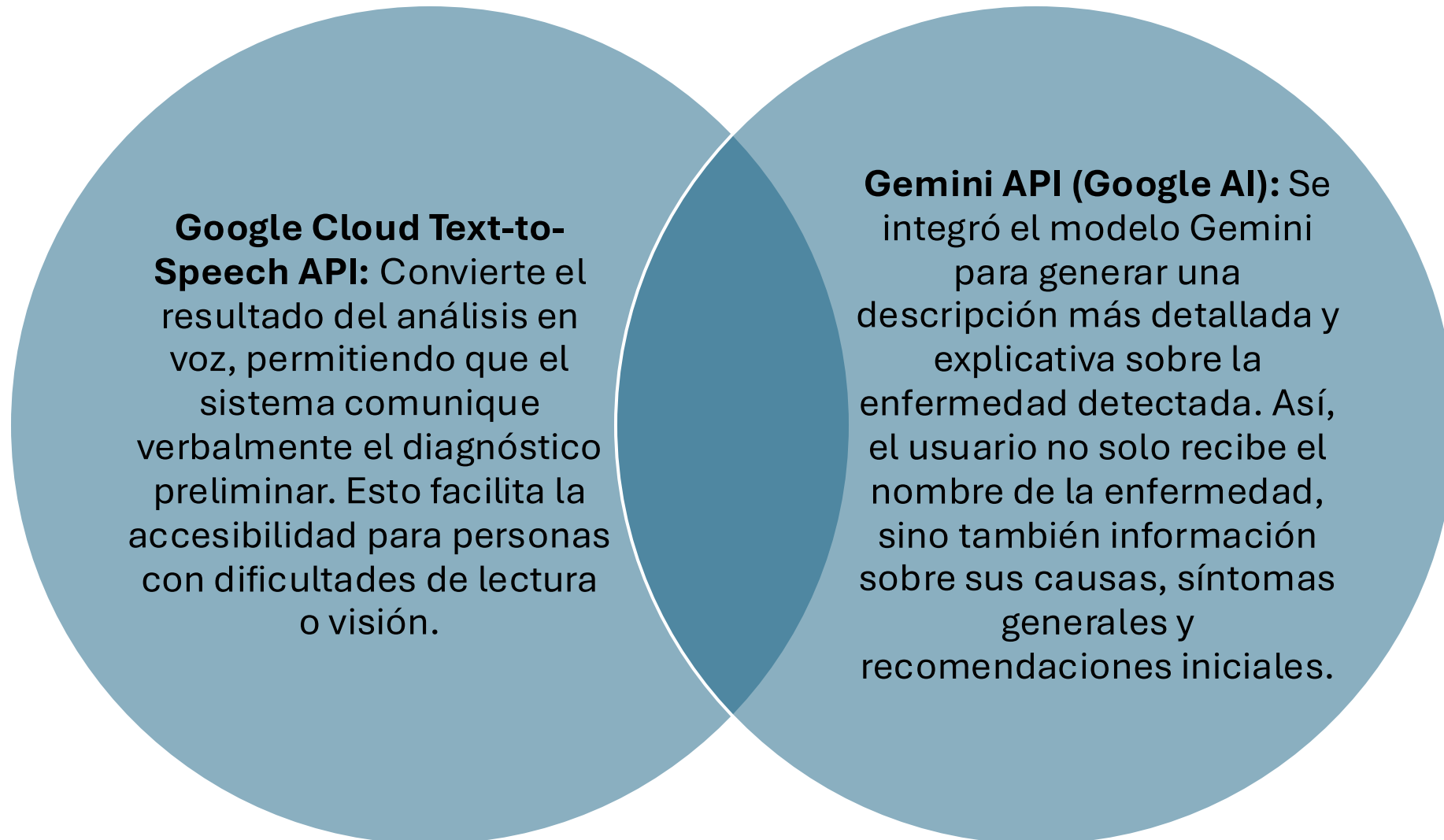
- **Inteligencia Artificial (IA):** Es la base que permite a DermaScan AI simular el razonamiento humano al analizar imágenes y proporcionar respuestas comprensibles a los usuarios. El objetivo fue que la aplicación no solo clasificara imágenes, sino que también ofreciera una explicación accesible del resultado.

2.) Fundamentos teóricos aplicados en DermaScan AI

- **Machine Learning (ML):** Utilizamos ML como técnica para que el sistema aprenda automáticamente a clasificar imágenes de lesiones cutáneas a partir de ejemplos. El modelo se entrenó con un gran número de imágenes etiquetadas, permitiéndole generalizar y clasificar imágenes nuevas no vistas.
- **Deep Learning (DL):** Dentro del ML, implementamos una **Red Neuronal Convolutiva (CNN)**, que es un tipo de DL especialmente diseñado para procesar imágenes. Gracias a sus múltiples capas, el modelo pudo identificar patrones complejos en las lesiones cutáneas (colores, texturas, formas) y clasificarlas con alta precisión.

3.) Servicios / APIs

La aplicación no solo realiza la clasificación, sino que también integra servicios que enriquecen la experiencia del usuario:



4. Descripción del Dataset

El dataset utilizado fue “**Skin Diseases**” disponible en Kaggle (ascanipek, 2022). Esta base de datos contiene imágenes clasificadas de manera precisa para diferentes tipos de enfermedades cutáneas, siendo adecuada para entrenar redes neuronales de clasificación.

Categorías del Dataset:

1. Enfermedades infecciosas

2. Eczema

3. Acné

4. Enfermedades pigmentarias

5. Tumores benignos

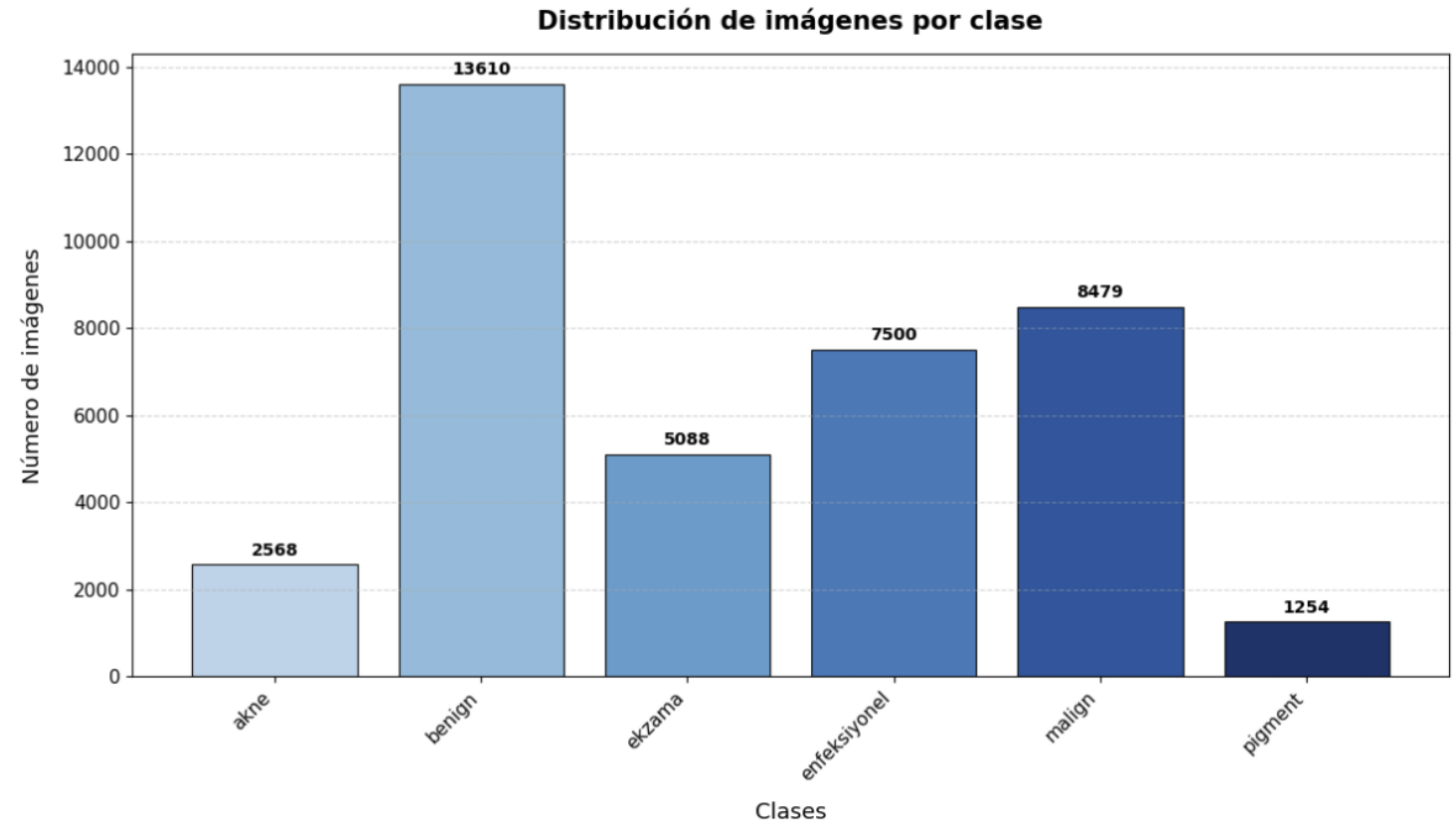
6. Tumores malignos

Cantidad de datos: Aproximadamente **38.760 imágenes**, distribuidas inicialmente de manera desigual entre las clases.

5.) Preprocesamiento aplicado

a.) Análisis del Dataset original

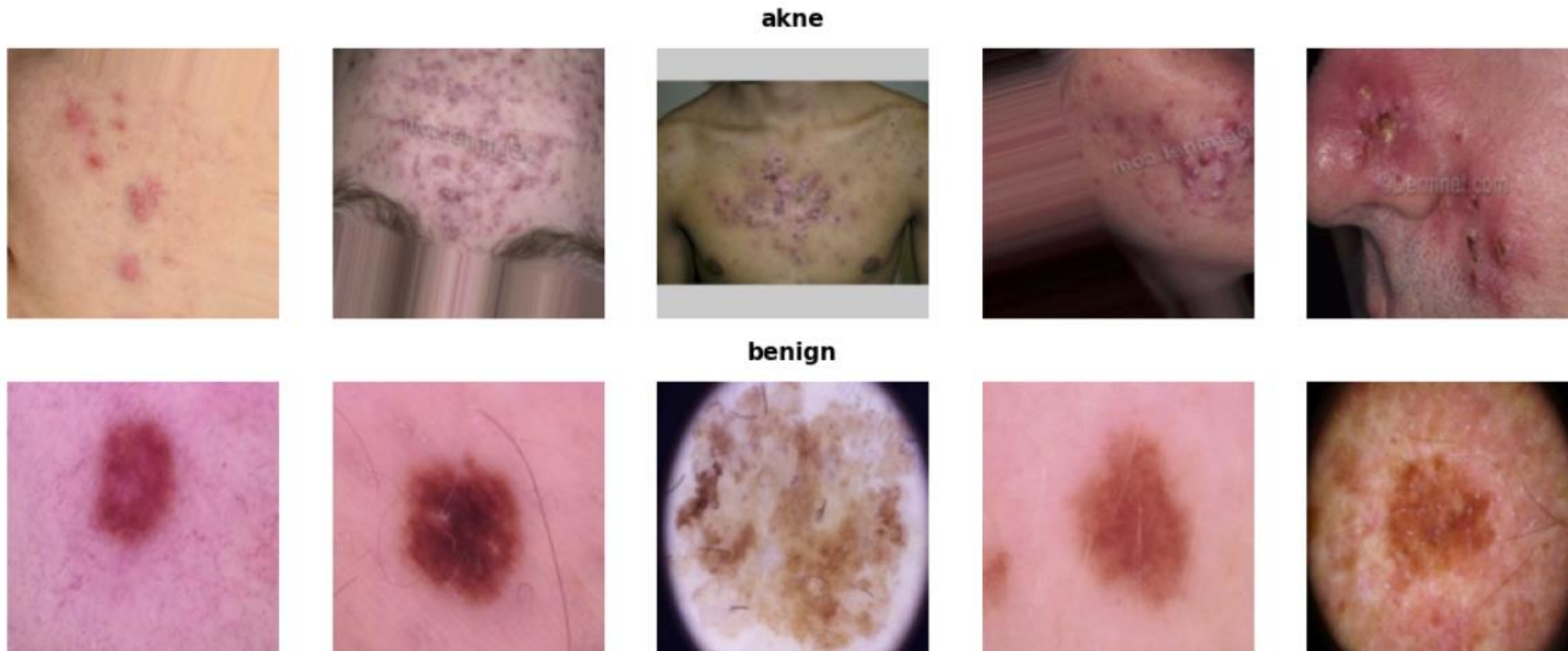
Histograma inicial: gráfico de barras que muestra el desbalance del dataset original. Algunas clases tenían muchas más imágenes que otras, lo que hubiera generado un sesgo durante el entrenamiento.



5.) Preprocesamiento aplicado

a.) Análisis del Dataset original

Muestra aleatoria: Se extrajeron **5 imágenes aleatorias por clase**, verificando visualmente la diversidad y calidad de las imágenes. Esto permitió confirmar que las clases eran visualmente distinguibles



5.) Preprocesamiento aplicado

a.) Análisis del Dataset original

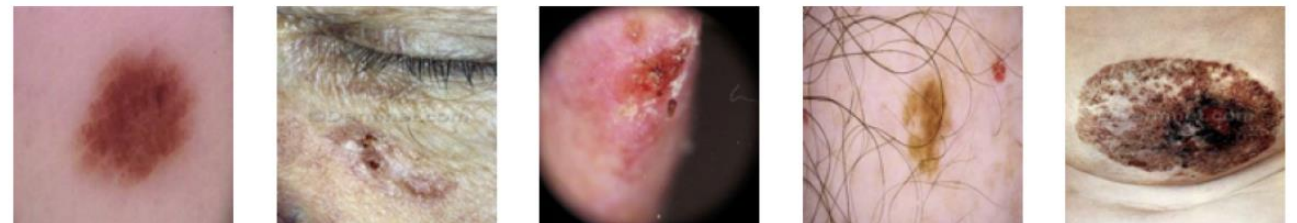
ekzama



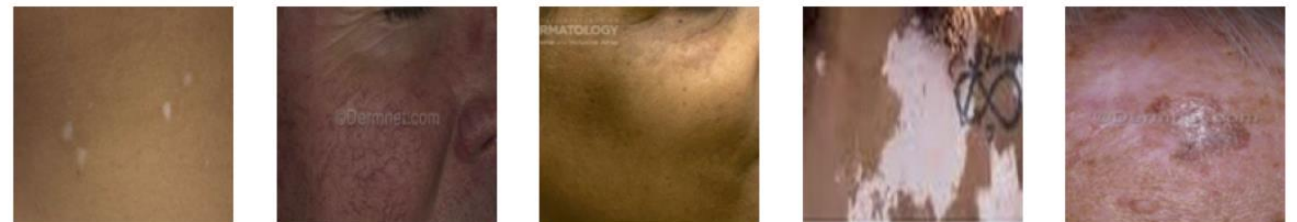
enfeksiyone



malign



pigment



5.) Preprocesamiento aplicado

b.) Data Augmentation

- Para equilibrar las clases se aplicaron técnicas de aumento de datos (**Data Augmentation**) como:

- Rotaciones aleatorias.
- Volteos horizontales.
- Zoom aleatorio.
- Cambios de brillo.

```
# DATA AUGMENTATION A 8000 IMAGENES

origen = "redimension" # carpeta original con clases
destino = "redimension_balanceado_8000" # carpeta destino
limite = 8000

# Crear carpeta destino
os.makedirs(destino, exist_ok=True)

# Aumentos a aplicar
augmentador = ImageDataGenerator(
    rescale=1./255,
    rotation_range=15,
    zoom_range=0.1,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True
)
```

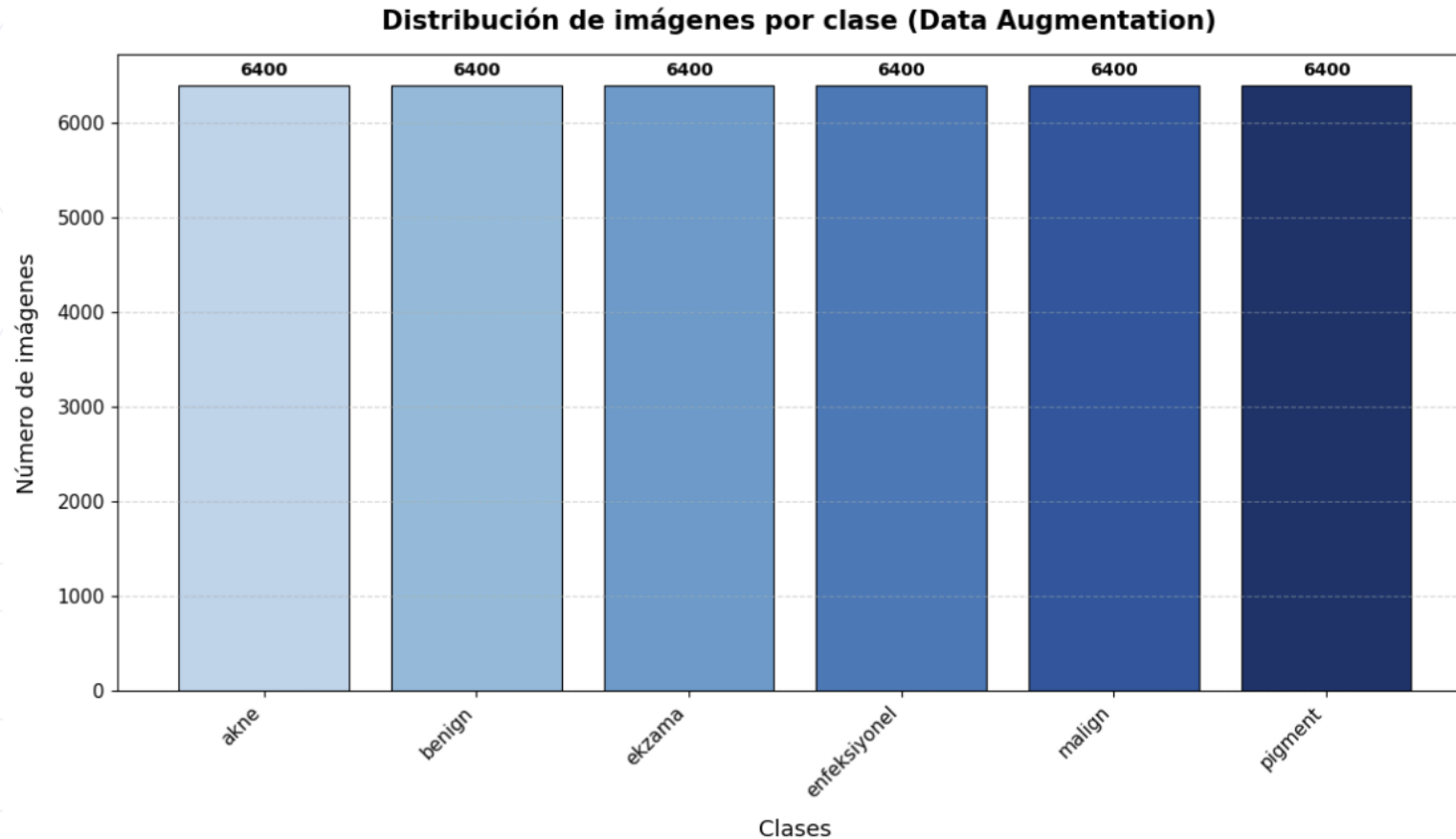
Resultado: Cada clase en redimension_balanceado_8000 contiene exactamente 8000 imágenes, balanceadas mediante una combinación de datos originales y aumentados, listas para su uso en modelos de clasificación.

b.) Redimensionamiento

1. Antes de entrenar el modelo, se redimensionaron todas las imágenes a un tamaño uniforme
2. El dataset fue dividido en dos subconjuntos principales:
 - 80% para entrenamiento, utilizado para aprender los patrones de las clases.
 - 20% para prueba, con datos no vistos.
3. Esta división se realizó de forma balanceada por clase, asegurando una distribución equitativa.
4. El 20% de prueba se dividió en partes iguales para formar los conjuntos de validación (10%) y test final (10%), lo que permitió ajustar hiperparámetros y evaluar el rendimiento del modelo de forma objetiva

5.) Preprocesamiento aplicado

e.) Análisis del Dataset balanceado



La gráfica muestra un dataset balanceado, con **6400** imágenes por clase. Se aplicó data augmentation a las clases minoritarias para evitar sesgos durante el entrenamiento y mejorar la generalización del modelo.

6.) Método Desarrollado

a.) Creación de DataLoader con muestreo balanceado

```
# Aplicar muestreo balanceado
targets = [label for _, label in train_dataset]
class_counts = np.bincount(targets)
class_weights = 1. / class_counts
weights = [class_weights[t] for t in targets]
sampler = WeightedRandomSampler(weights, num_samples=len(weights), replacement=True)

# DataLoaders
train_loader = DataLoader(train_dataset, batch_size=BATCH_SIZE, sampler=sampler)
val_loader = DataLoader(val_dataset, batch_size=BATCH_SIZE, shuffle=False)
test_loader = DataLoader(final_test_dataset, batch_size=BATCH_SIZE, shuffle=False)

print(" Celda completada: Sampler y DataLoaders listos")
```

Celda completada: Sampler y DataLoaders listos

Este muestreador ajusta la probabilidad de que cada imagen sea seleccionada durante el entrenamiento, de modo que todas las clases tengan la misma importancia. Luego, se definen los DataLoader para entrenamiento, validación y prueba

6.) Método Desarrollado

b.) Carga de modelo preentrenado (EfficientNet)

Se carga una versión preentrenada de EfficientNet-B0, una arquitectura eficiente y moderna para clasificación de imágenes. Se reemplaza la capa final (_fc) para ajustarse al número de clases del dataset dermatológico.

```
# Cargar modelo preentrenado  
model = EfficientNet.from_pretrained('efficientnet-b0')  
in_features = model._fc.in_features  
model._fc = nn.Linear(in_features, NUM_CLASSES)  
model = model.to(device)
```

6.) Método Desarrollado

c.) Definición de loss, optimizador y scheduler

Se configura la función de pérdida (CrossEntropyLoss) para clasificación multiclase, el optimizador Adam, y un scheduler que reduce la tasa de aprendizaje si la pérdida de validación no mejora.

```
# Entrenamiento
criterion = nn.CrossEntropyLoss()
optimizer = optim.Adam(model.parameters(), lr=1e-4)
scheduler = optim.lr_scheduler.ReduceLROnPlateau(optimizer, mode='min', patience=3)

print("Celda completada: Optimizador, loss y scheduler definidos")
```

6.) Método Desarrollado

d.) Entrenamiento principal del modelo

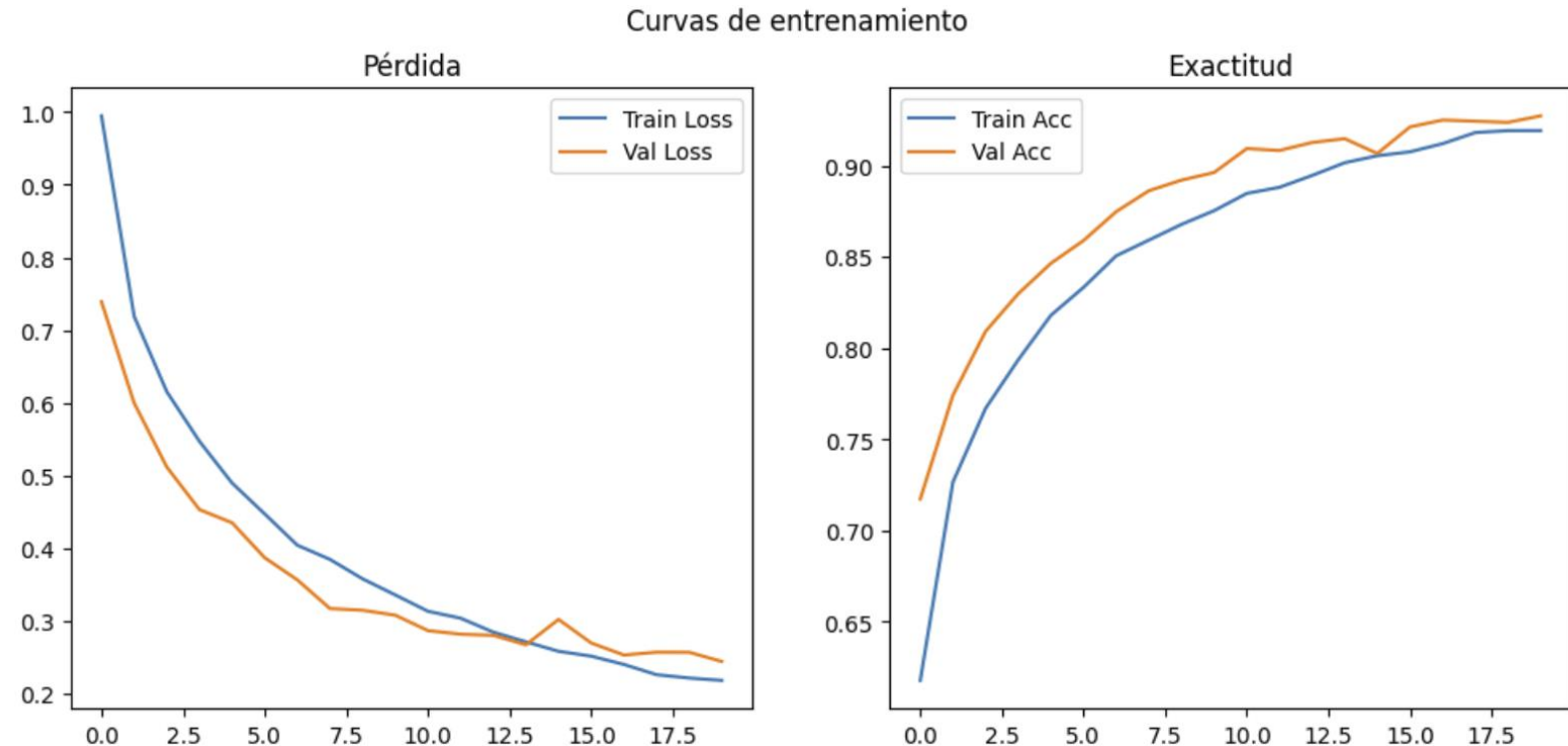
Se entrena el modelo por 20 épocas. Se realiza un ciclo de entrenamiento sobre los datos y luego una evaluación sobre el conjunto de validación. Se calcula y guarda el mejor modelo si la pérdida de validación mejora

```
Epoch 1/20 | Train Loss: 0.9946 | Train Acc: 0.6178 | Val Loss: 0.7392 | Val Acc: 0.7173 | ⌚ Tiempo: 874.00 s
Modelo guardado (nuevo mejor)
Epoch 2/20 | Train Loss: 0.7191 | Train Acc: 0.7266 | Val Loss: 0.6000 | Val Acc: 0.7742 | ⌚ Tiempo: 784.77 s
Modelo guardado (nuevo mejor)
Epoch 3/20 | Train Loss: 0.6156 | Train Acc: 0.7670 | Val Loss: 0.5120 | Val Acc: 0.8092 | ⌚ Tiempo: 788.77 s
Modelo guardado (nuevo mejor)
Epoch 4/20 | Train Loss: 0.5474 | Train Acc: 0.7936 | Val Loss: 0.4535 | Val Acc: 0.8298 | ⌚ Tiempo: 787.99 s
Modelo guardado (nuevo mejor)
Epoch 5/20 | Train Loss: 0.4900 | Train Acc: 0.8180 | Val Loss: 0.4354 | Val Acc: 0.8465 | ⌚ Tiempo: 790.76 s
Modelo guardado (nuevo mejor)
Epoch 6/20 | Train Loss: 0.4475 | Train Acc: 0.8333 | Val Loss: 0.3873 | Val Acc: 0.8590 | ⌚ Tiempo: 790.32 s
Modelo guardado (nuevo mejor)
Epoch 7/20 | Train Loss: 0.4045 | Train Acc: 0.8506 | Val Loss: 0.3569 | Val Acc: 0.8748 | ⌚ Tiempo: 789.70 s
Modelo guardado (nuevo mejor)
Epoch 8/20 | Train Loss: 0.3850 | Train Acc: 0.8592 | Val Loss: 0.3175 | Val Acc: 0.8862 | ⌚ Tiempo: 789.76 s
Modelo guardado (nuevo mejor)
Epoch 9/20 | Train Loss: 0.3583 | Train Acc: 0.8678 | Val Loss: 0.3152 | Val Acc: 0.8921 | ⌚ Tiempo: 789.60 s
Modelo guardado (nuevo mejor)
Epoch 10/20 | Train Loss: 0.3362 | Train Acc: 0.8753 | Val Loss: 0.3084 | Val Acc: 0.8962 | ⌚ Tiempo: 790.07 s
Modelo guardado (nuevo mejor)
Epoch 11/20 | Train Loss: 0.3138 | Train Acc: 0.8848 | Val Loss: 0.2872 | Val Acc: 0.9094 | ⌚ Tiempo: 790.47 s
Modelo guardado (nuevo mejor)
Epoch 12/20 | Train Loss: 0.3042 | Train Acc: 0.8881 | Val Loss: 0.2823 | Val Acc: 0.9083 | ⌚ Tiempo: 790.82 s
Modelo guardado (nuevo mejor)
Epoch 13/20 | Train Loss: 0.2851 | Train Acc: 0.8946 | Val Loss: 0.2807 | Val Acc: 0.9127 | ⌚ Tiempo: 789.73 s
Modelo guardado (nuevo mejor)
Epoch 14/20 | Train Loss: 0.2718 | Train Acc: 0.9016 | Val Loss: 0.2677 | Val Acc: 0.9148 | ⌚ Tiempo: 788.52 s
Modelo guardado (nuevo mejor)
Epoch 15/20 | Train Loss: 0.2587 | Train Acc: 0.9055 | Val Loss: 0.3025 | Val Acc: 0.9067 | ⌚ Tiempo: 790.40 s
Epoch 16/20 | Train Loss: 0.2521 | Train Acc: 0.9076 | Val Loss: 0.2702 | Val Acc: 0.9213 | ⌚ Tiempo: 790.45 s
Epoch 17/20 | Train Loss: 0.2407 | Train Acc: 0.9121 | Val Loss: 0.2536 | Val Acc: 0.9250 | ⌚ Tiempo: 789.93 s
Modelo guardado (nuevo mejor)
Epoch 18/20 | Train Loss: 0.2266 | Train Acc: 0.9181 | Val Loss: 0.2575 | Val Acc: 0.9244 | ⌚ Tiempo: 790.18 s
Epoch 19/20 | Train Loss: 0.2220 | Train Acc: 0.9192 | Val Loss: 0.2574 | Val Acc: 0.9237 | ⌚ Tiempo: 790.38 s
Epoch 20/20 | Train Loss: 0.2187 | Train Acc: 0.9192 | Val Loss: 0.2448 | Val Acc: 0.9273 | ⌚ Tiempo: 790.26 s
Modelo guardado (nuevo mejor)
Entrenamiento finalizado
```

6.) Método Desarrollado

d.) Visualización de curvas de pérdida y precisión

Se grafican las curvas de pérdida y exactitud (accuracy) durante las épocas, tanto para entrenamiento como validación. Estas gráficas permiten identificar si hay sobreajuste o subajuste en el modelo.



6.) Método Desarrollado

d.) Evaluación del modelo final

Se cargó el modelo previamente entrenado y guardado (best_model.pth) para evaluar su rendimiento sobre el conjunto de prueba. El código calcula las métricas de clasificación (precisión, recall, F1-score) y genera una matriz de confusión.

Resultados clave:

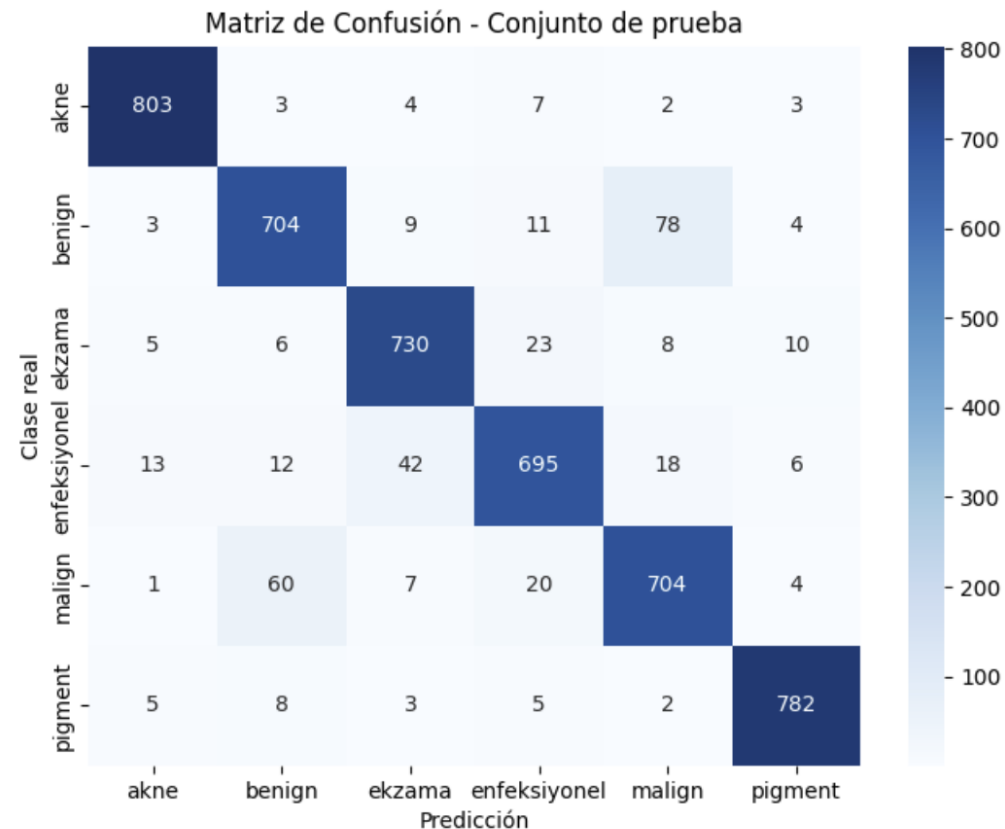
- Accuracy global: 92%
- F1-score por clase: entre 0.88 y 0.97
- El modelo presenta excelente desempeño en todas las clases, destacando especialmente en akne y pigment.

Esto confirma que el modelo generaliza bien y es confiable para clasificar imágenes dermatológicas.

6.) Método Desarrollado - Evaluación del modelo final

La matriz evidencia que el modelo logra una clasificación balanceada en todas las clases, aunque con margen de mejora en clases difíciles como los tumores malignos.

Clasificación por clase:				
	precision	recall	f1-score	support
akne	0.97	0.98	0.97	822
benign	0.89	0.87	0.88	809
ekzama	0.92	0.93	0.93	782
enfeksiyonel	0.91	0.88	0.90	786
malign	0.87	0.88	0.88	796
pigment	0.97	0.97	0.97	805
accuracy			0.92	4800
macro avg	0.92	0.92	0.92	4800
weighted avg	0.92	0.92	0.92	4800



Evaluación final realizada.

6.) Método Desarrollado - Ejemplo

Clase predicha: akne
Confianza: 84.37%

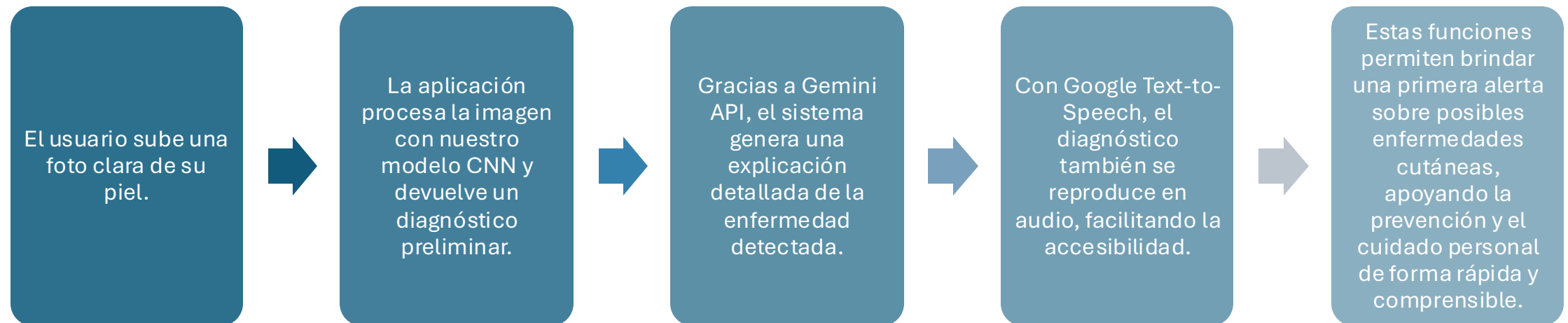
akne (84.37%)



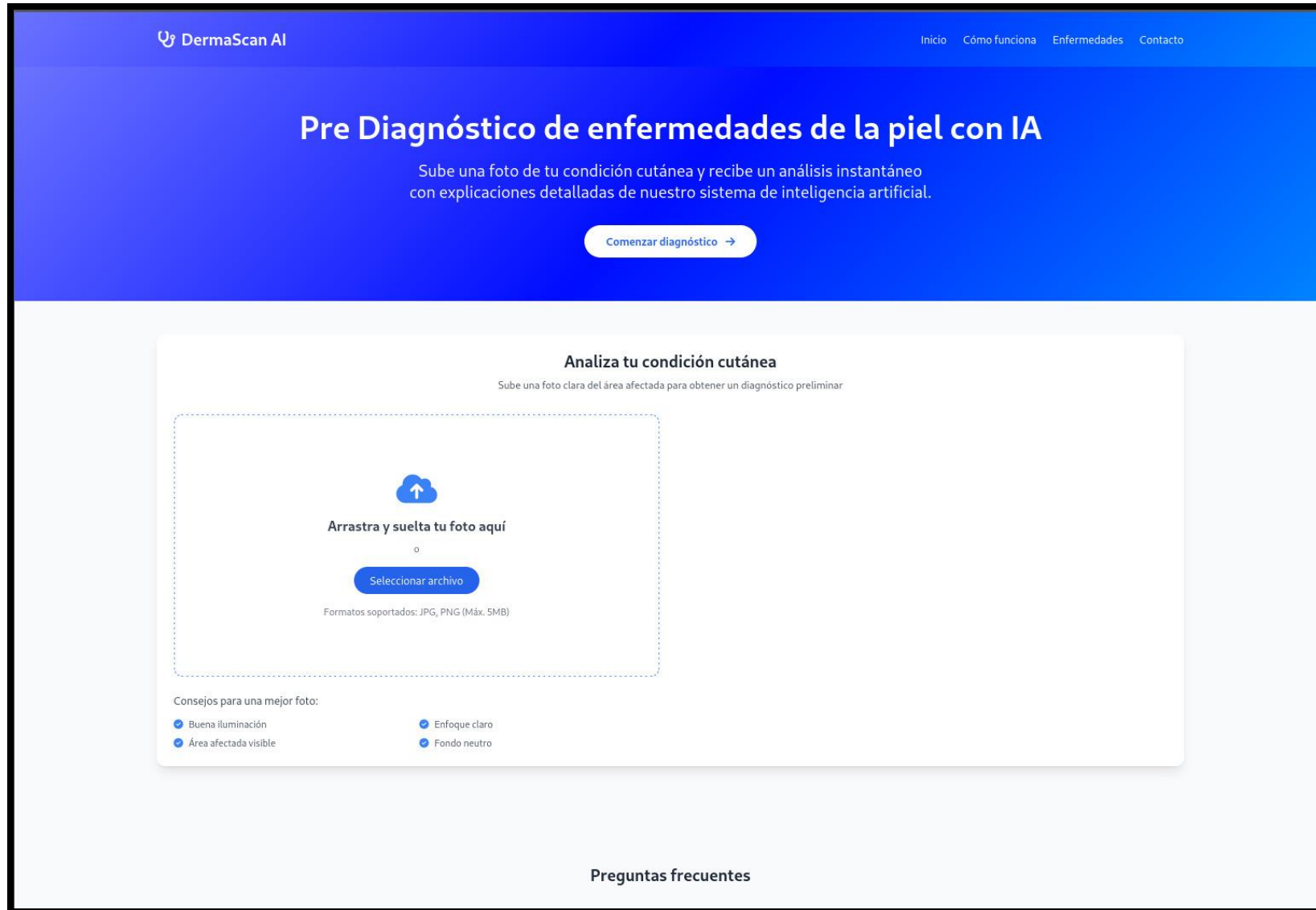
7.) Aplicación Desarrollada

DermaScan AI es una plataforma web intuitiva que permite a los usuarios subir imágenes de lesiones cutáneas y recibir un análisis instantáneo basado en inteligencia artificial.

El proceso es sencillo y accesible:



7.) Aplicación Desarrollada



7.) Aplicación Desarrollada

DermaScan AI

InicioCómo funcionaEnfermedadesContacto

Pre Diagnóstico de enfermedades de la piel con IA

Sube una foto de tu condición cutánea y recibe un análisis instantáneo con explicaciones detalladas de nuestro sistema de inteligencia artificial.

Comenzar diagnóstico →

Analiza tu condición cutánea

Sube una foto clara del área afectada para obtener un diagnóstico preliminar.

Arrastra y suelta tu foto aquí

o

Seleccionar archivo

Formatos soportados: JPG, PNG (Máx. 5MB)

Consejos para una mejor foto:

☒ Buena iluminación

☒ Área afectada visible

☒ Enfoque claro

☒ Fondo neutro

Opciones de análisis:

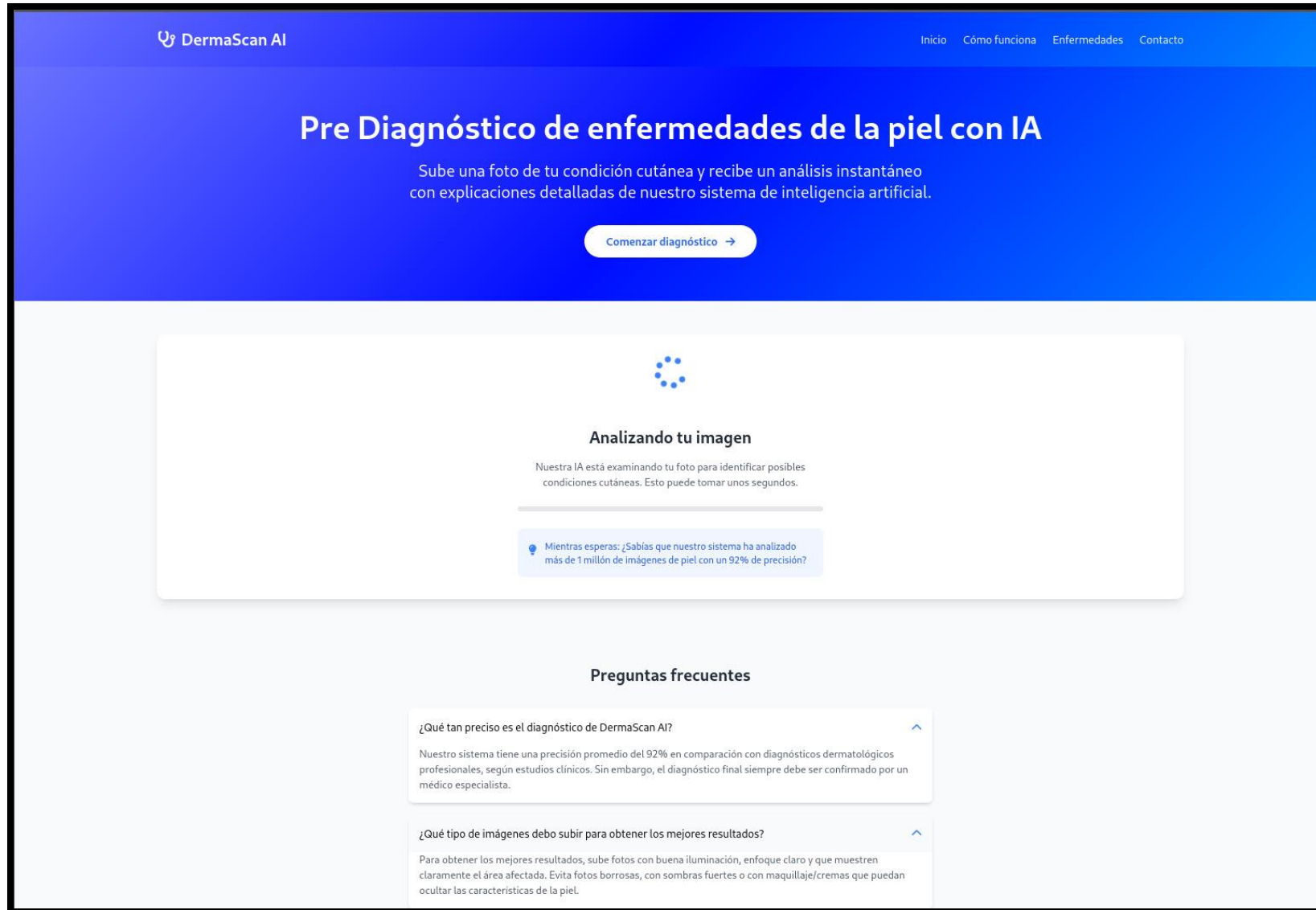
☒ Incluir diagnóstico de posibles condiciones

☒ Generar explicación detallada con IA

☐ Comparar con casos similares

Analizar imagen

7.) Aplicación Desarrollada



7.) Aplicación Desarrollada

Resultados del análisis


[Analizar otra imagen](#)

Diagnóstico principal

Eccema 100% de coincidencia

Resumen: Luego de la predicción, la imagen muestra características consistentes con Eccema.

Piel seca Eritema Prurito Inflamación



Ver en tamaño completo

Otras posibles condiciones

<p>Acné</p> <p>Clasificación interna: akne. Esta probabilidad es aproximada según la IA.</p>	0%	<p>Lesión benigna</p> <p>Clasificación interna: benign. Esta probabilidad es aproximada según la IA.</p>	0%
<p>Infección cutánea</p> <p>Clasificación interna: enfeksiyonel. Esta probabilidad es aproximada según la IA.</p>	0%	<p>Lesión maligna</p> <p>Clasificación interna: malign. Esta probabilidad es aproximada según la IA.</p>	0%
<p>Lesión pigmentada no específica</p> <p>Clasificación interna: pigment. Esta probabilidad es aproximada según la IA.</p>	0%		

Explicación detallada por IA

El eccema, o dermatitis atópica, es una enfermedad de la piel que causa inflamación, picazón intensa y sarpullido. Sus causas no se conocen completamente, pero se cree que una combinación de factores genéticos y ambientales juega un papel importante. Algunos desencadenantes comunes son los alérgenos (como ácaros del polvo, polen o alimentos), irritantes (como jabones fuertes o detergentes), estrés, y cambios en el clima. Las recomendaciones generales incluyen mantener la piel hidratada con cremas humectantes, evitar duchas o baños muy calientes o largos, usar jabones suaves e hipoalérgicos, identificar y evitar los desencadenantes personales, y seguir el tratamiento recetado por un médico si lo hay. Es fundamental no rascarse, ya que esto empeora la situación. Si tienes eccema, es importante consultar a un dermatólogo para un diagnóstico y tratamiento adecuado. Nota importante: Este análisis es preliminar y no sustituye una evaluación médica profesional. Si los síntomas persisten o empeoran, busca atención médica.

0:00 / 1:01

Conclusiones

1. DermaScan AI implementó una red neuronal EfficientNet-B0, logrando una clasificación precisa y confiable de enfermedades cutáneas.
2. El mayor desafío fue el desbalance de clases, solucionado mediante data augmentation, submuestreo y ajuste de pesos por clase, lo que permitió obtener un dataset balanceado con 8000 imágenes por categoría.
3. El modelo alcanzó una precisión global del 92%, con métricas equilibradas entre clases, lo que demuestra su capacidad para generalizar correctamente incluso en clases difíciles.
4. Se integró una herramienta práctica que permite al usuario analizar cualquier imagen nueva, facilitando su aplicación en el mundo real.

Recomendaciones

1. Ampliar el dataset con imágenes locales de Ecuador para adaptar el modelo a contextos regionales.
2. Validar el modelo clínicamente junto a dermatólogos para confirmar su precisión diagnóstica en entornos reales.
3. Explorar nuevas arquitecturas como EfficientNetV2 o Vision Transformers para optimizar el rendimiento.
4. Desarrollar una versión móvil u offline que permita el análisis sin conexión a internet, aumentando su accesibilidad.

Referencias

- Chollet, F., et al. (2015). **Keras**. Biblioteca de alto nivel para redes neuronales, construida sobre TensorFlow.
- Abadi, M., et al. (2016). **TensorFlow**. Plataforma de código abierto para aprendizaje automático.
- Pedregosa, F., et al. (2011). **Scikit-learn**: herramientas simples y eficientes de minería de datos y análisis de datos. *Journal of Machine Learning Research*, 12, 2825–2830.
- Reback, J., et al. (2020). **Pandas**: estructuras de datos para Python, con indexación etiquetada y operaciones de tiempo real.
- Harris, C. R., et al. (2020). **NumPy**: Fundamento para la computación científica en Python.
- Hunter, J. D. (2007). **Matplotlib**: biblioteca para gráficos estáticos y animados en Python.
- Ramírez, A., & otros. (2023). **FastAPI**: framework moderno y de alto rendimiento para construir APIs con Python 3.7+.
- Angular Team. (2023). **Angular**: plataforma para crear aplicaciones web.
- Google Cloud. (2023). **Text-to-Speech API**.
- Google AI. (2024). **Gemini API**.



**Más que formar
Profesionales,
*Transformamos Vidas***



www.ups.edu.ec