Edgar Fuentes Lamigueiro, Sabela Yáñez López

Python Introductorio, Visión Artificial 1,

Aprendizaje Automático 1

Máster Universitario en Informática Industrial y Robótica

CLASIFICACIÓN SUPERVISADA

Técnicas de preprocesado, métodos de aprendizaje automático y técnicas de extracción de características

# ÍNDICE

[1 ÍNDICE 1](#_Toc193751278)

[1. OBJETO 2](#_Toc193751279)

[2. ETAPA DE DISEÑO 3](#_Toc193751280)

[2.1. DATASET 3](#_Toc193751283)

[2.2. TÉCNICAS DE PREPROCESADO 3](#_Toc193751284)

[2.3. TÉCNICAS DE APRENDIZAJE AUTOMÁTICO 4](#_Toc193751287)

[2.3.1 REGRESIÓN LOGÍSTICA 4](#_Toc193751290)

[2.3.2 CNN 4](#_Toc193751291)

[2.3.3 KNN 4](#_Toc193751292)

[2.3.4 ÁRBOL DE DECISIÓN 5](#_Toc193751293)

[2.3.5 RANDOM FOREST 5](#_Toc193751294)

[2.4. TÉCNICAS DE EXTRACCIÓN DE CARACTERÍSTICAS 6](#_Toc193751295)

[2.4.1 VGG16 6](#_Toc193751297)

[2.4.2 MOBILENETV2 6](#_Toc193751298)

[2.4.3 ANN + HOG 6](#_Toc193751299)

[2.4.4 SVM + HOG 6](#_Toc193751300)

[2.5. FUNCIONES AUXILIARES 7](#_Toc193751301)

[2.5.1 7](#_Toc193751303)

[3. ANÁLISIS DE RESULTADOS 8](#_Toc193751304)

[3.1. REGRESIÓN LOGÍSTICA 8](#_Toc193751309)

[3.2. CNN 10](#_Toc193751310)

[3.3. KNN 12](#_Toc193751311)

[3.4. ÁRBOL DE DECISIÓN 13](#_Toc193751312)

[3.5. RANDOM FOREST 14](#_Toc193751313)

[3.6. VGG16 17](#_Toc193751315)

[3.7. MOBILENETV2 17](#_Toc193751316)

[3.8. ANN + HOG 17](#_Toc193751317)

[3.9. SVM + HOG 17](#_Toc193751318)

[4. CONCLUSIONES 18](#_Toc193751319)

[4.1. Accuracy 18](#_Toc193751321)

[4.2. Recall 18](#_Toc193751322)

[4.3. Precision 19](#_Toc193751323)

[4.4. F1-score 19](#_Toc193751324)

# OBJETO

El objetivo del trabajo consiste en el desarrollo y entrenamiento de modelos de aprendizaje automático para la clasificación supervisada de un conjunto de imágenes.

A partir del dataset “[Flowers Multiclass Datasets](https://www.kaggle.com/datasets/alsaniipe/flowers-multiclass-datasets)” se aplican diferentes técnicas de aprendizaje automático con el fin de identificar la metodología más efectiva para clasificar los datos. Como se trabaja con datos de imagen, se implementan técnicas de preprocesado, además de diferentes estrategias de extracción de características.

El estudio experimental incluirá la evaluación de modelos mediante validación cruzada, calculando métricas como exactitud, sensibilidad, especificidad, precisión, f1-score y AUC, además de un análisis comparativo del impacto que tiene la variación de hiperparámetros en el rendimiento de cada modelo.

# ETAPA DE DISEÑO



## DATASET

El dataset contiene imágenes de cinto tipos de flores:

* Daisy (Margarita)
* Dandelion (Diente de león)
* Roses (Rosas)
* Sunflowers (Girasoles)
* Tulips (Tulipanes)

## TÉCNICAS DE PREPROCESADO



La aplicación de técnicas de preprocesado permite mejorar la calidad de los datos y optimizar el rendimiento de los modelos.

En este trabajo se han aplicado las siguientes técnicas:

* Redimensionalidad
* Conversión a escala de grises
* Normalización
* Data augmentation

## TÉCNICAS DE APRENDIZAJE AUTOMÁTICO



### REGRESIÓN LOGÍSTICA

La regresión logística es un modelo de aprendizaje supervisado utilizado para clasificación. Este modelo permite clasificar cada imagen en una de las cinco categorías del dataset, transformando sus características en probabilidades de pertenencia a cada clase.

Al tratarse de datos no estructurados, las imágenes se convierten en vectores numéricos mediante flattening, y se aplica regresión logística multiclase y validación cruzada.

### CNN

Las Redes Neuronales Convolucionales son una arquitectura de aprendizaje profundo diseñadas para procesar y analizar datos de imágenes, extrayendo automáticamente características relevantes y eliminando la necesidad de diseñar manualmente atributos como bordes, texturas o formas.

Una CNN está compuesta por varias capas diseñadas para transformar y aprender características de las imágenes a diferentes niveles de abstracción.

En primer lugar, tenemos una función con una arquitectura de:

* Capas convolucionales: 2
* Capas de pooling: 2
* Capa de Global Average Pooling: 1
* Capas densas: 2 (una capa densa y una capa de salida)
* Capa de Dropout: 1

En segundo lugar, una función con:

* Capas convolucionales: 4
* Capas de pooling: 2
* Capa de normalización (BatchNormalization): 2
* Capas de Dropout: 3
* Capa de Global Average Pooling: 1
* Capa densa: 1
* Capa de salida (Softmax): 1

Para evaluar el desempeño de la CNN en la clasificación del dataset se aplica validación cruzada mediante “StratifiedKFold”, lo que permite analizar la variabilidad del rendimiento de la CNN en distintos subconjuntos de datos, asegurando que el modelo generaliza correctamente y no está sobreajustado.

### KNN

KNN es una técnica de aprendizaje automático supervisado que se emplea para problemas de clasificación y de regresión. Su funcionamiento está basado en la idea de que un punto de datos pertenece a la clase más común entre sus k vecinos más cercanos.

Además, se combina con PCA para reducir la dimensionalidad de las imágenes antes de aplicar el clasificador y se aplica validación cruzada utilizando “GridSearchCV”, que permite encontrar la mejor combinación de parámetros para mejorar el rendimiento del modelo.

### ÁRBOL DE DECISIÓN

Un árbol de decisión es un modelo de aprendizaje automático utilizado tanto para clasificación como para regresión. Su funcionamiento se basa en dividir los datos en subconjuntos más pequeños utilizando condiciones de decisión basadas en los valores de las características.

Pueden ser propensos al sobreajuste, por lo que se pueden limitar parámetros como: profundidad máxima (max\_depth), mínimo de muestras para dividir un nodo (min\_samples\_split), mínimo de muestras en un nodo hoja (min\_samples\_leaf), …

Durante el diseño, se proporcionan tres enfoques diferentes para entrenar el árbol de decisión:

1. Árbol de decisión base:

Es el modelo básico, sin preentreno ni extracción de características. Para evaluar el modelo se aplica GridSearchCV para encontrar los mejores hiperparámetros mediante validación cruzada

1. Árbol de decisión con VGG16 como extractor de características:

En este caso se usa una red preentrenada VGG16 para extraer características avanzadas de las imágenes. No se aplica validación cruzada debido al alto tiempo de simulación requerida.

1. Árbol de decisión con VGG16 y PCA:

En este caso, además de una red preentrenada VGG16, se aplica PCA (Análisis de componentes principales) para reducir la dimensionalidad de las características.

### RANDOM FOREST

Random Forest es un algoritmo de aprendizaje automático basado en árboles de decisión. Se compone de múltiples árboles de decisión entrenados con diferentes subconjuntos de datos, y la predicción final se obtiene mediante el voto mayoritario (para clasificación) o el promedio (para regresión).

Este algoritmo es menos propenso al sobreajuste que un solo árbol de decisión.

Durante el diseño, se proporcionan los siguientes enfoques:

1. Random Forest modelo base:

El modelo usa directamente los píxeles de las imágenes sin extracción avanzada de características. Entrena un modelo con n árboles y realiza validación cruzada para evaluar el rendimiento.

1. Randfom Forest con VGG16 y PCA:

Este modelo utiliza VGG16 como extractor de características en lugar de usar los píxeles sin procesar, y aplica PCA. Realiza validación cruzada para evaluar el rendimiento.

1. Random Forest con VGG16, PCA Y HOG:

Aplica VGG16+PCA según el modelo anterior, además de extraer otras características por HOG. El modelo pretende mejorar la precisión mediante la combinación de múltiples técnicas de extracción de características. Realiza validación cruzada para evaluar el rendimiento.

## TÉCNICAS DE EXTRACCIÓN DE CARACTERÍSTICAS



### VGG16

VGG16 es una red neuronal convolucional compuesta por 16 capas de convolución y capas completamente conectadas. El modelo está preentrenado en ImageNet, por lo que se emplea como extractor de características, facilitando la transferencia de aprendizaje.

Durante el diseño del modelo, se siguen los siguientes pasos:

* Creación del modelo

Se carga la VGG16 sin la parte superior para usarlo como extractor de características y se congelan las capas convolucionales para que no se actualicen en el entrenamiento inicial. Además, se agregan nuevas capas densas para adaptarse al nuevo aprendizaje.

* Entrenamiento del modelo

Se aplica augmentación de datos mediante rotación, desplazamiento horizontal y vertical y volteo horizontal, y se procede con el entrenamiento del modelo.

### MOBILENETV2

MobileNetV2 es una red neuronal convolucional diseñada para ser eficiente en términos de tamaño, velocidad y precisión, lo que la hace ideal para sistemas con recursos limitados.

Durante el diseño del modelo, se siguen los mismos pasos que con VGG16.

### ANN + HOG

ANN es un modelo de aprendizaje profundo que se inspira en el funcionamiento del cerebro humano. Está formado por capaz de neuronas artificiales conectadas entre sí, que aprenden a través de la retropropagación.

Durante el diseño, se construye y entrena una red neuronal con una capa densa de 512 unidades, seguidas de una capa de dropout y una capa de salida. Además, el modelo se entrena con los datos HOG.

### SVM + HOG

SVM es un clasificador supervisado que busca encontrar el hiperplano que mejor separa las clases en el espacio de características. Este modelo intenta maximizar el margen entre los ejemplos de diferentes clases para obtener una mejor generalización.

## FUNCIONES AUXILIARES



### 

# ANÁLISIS DE RESULTADOS







## 



## REGRESIÓN LOGÍSTICA

El modelo de *regresión logística rgb* obtuvo unos resultados durante la validación cruzada de:

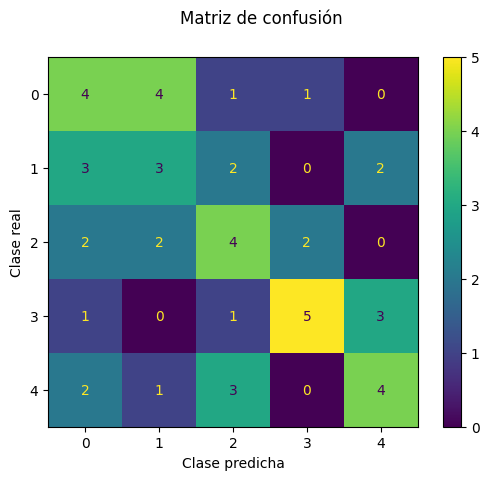
* **Accuracy:** 0.378249
* **Precision:** 0.377517
* **Recall:** 0.370069
* **f1:** 0.370564

Además, sus curvas roc\_auc y su matriz de confusión se pueden ver a continuación:

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.Gráfico

El contenido generado por IA puede ser incorrecto.



Por último, se muestran 10 imágenes con su etiqueta real y su etiqueta predicha por el modelo:

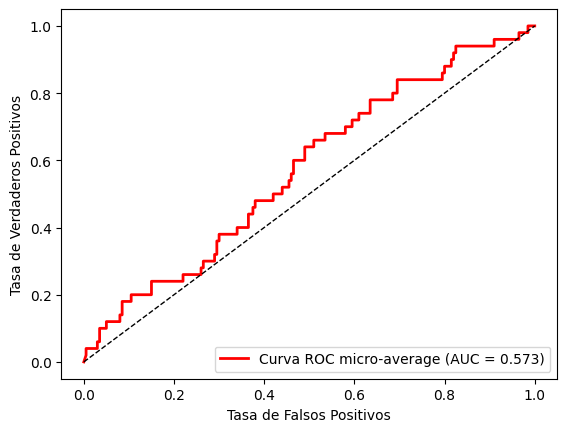
*Imagen de la pantalla de un celular con la imagen de una flor

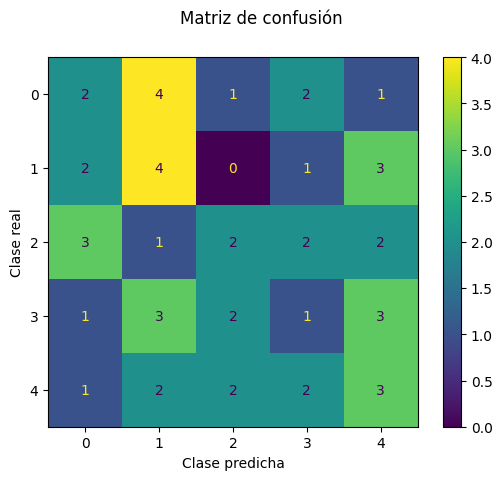
El contenido generado por IA puede ser incorrecto.*

El modelo de *regresión logística gray* obtuvo unos resultados durante la validación cruzada de:

* **Accuracy:** 0.267797
* **Precision:** 0.261721
* **Recall:** 0.259344
* **f1:** 0.257414

Además, sus curvas roc\_auc y su matriz de confusión se pueden ver a continuación:

Gráfico

El contenido generado por IA puede ser incorrecto.

Por último, se muestran 10 imágenes con su etiqueta real y su etiqueta predicha por el modelo:

Imagen que contiene Patrón de fondo

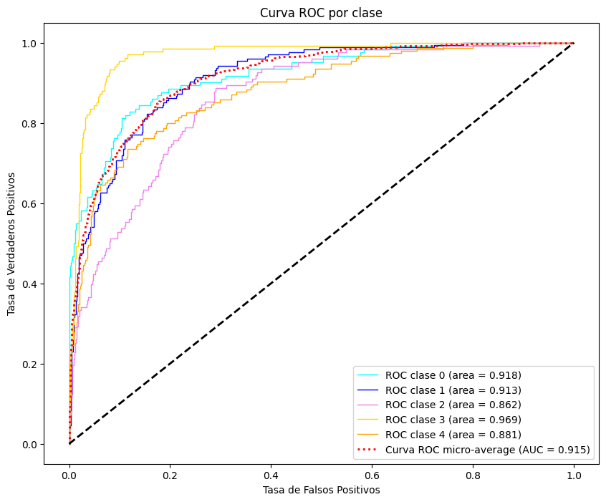
El contenido generado por IA puede ser incorrecto.

## CNN

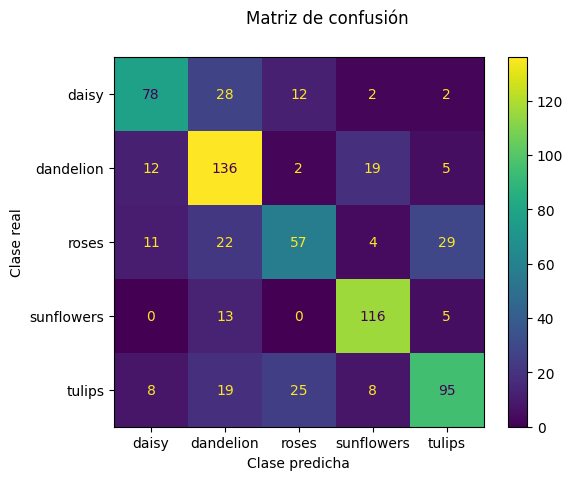
El modelo de *CNN1* obtuvo unos resultados durante la validación cruzada de:

* **Accuracy:** 0.685937
* **Precision:** 0.700000
* **Recall:** 0.713077
* **f1:** 0.700000

Además, sus curvas roc\_auc y su matriz de confusión se pueden ver a continuación:

Gráfico

El contenido generado por IA puede ser incorrecto.



Por último, se muestran 10 imágenes con su etiqueta real y su etiqueta predicha por el modelo:

*Imagen de la pantalla de un celular con la imagen de una flor

El contenido generado por IA puede ser incorrecto.*

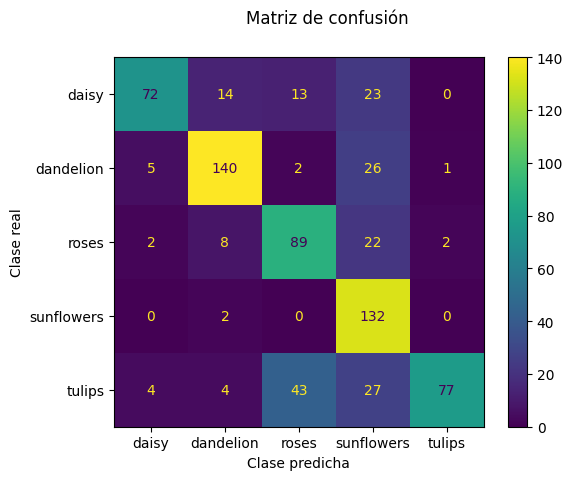
El modelo de *CN2* obtuvo unos resultados durante la validación cruzada de:

* **Accuracy:** 0.738559
* **Precision:** 0.744876
* **Recall:** 0.738559
* **f1:** 0.734516

Además, sus curvas roc\_auc y su matriz de confusión se pueden ver a continuación: Gráfico

El contenido generado por IA puede ser incorrecto.Gráfico

El contenido generado por IA puede ser incorrecto.



Por último, se muestran 10 imágenes con su etiqueta real y su etiqueta predicha por el modelo:

*Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.*

## KNN

El modelo de *KNN* obtuvo unos resultados durante la validación cruzada de:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PCA | N | accuracy | precision | recall | f1 | roc\_auc |
| **30** | **7** | **0,416206** | **0.435028** | |  |  | | --- | --- | | **0,435433** |  | | **0,420852** | **0.728536** |
| 50 | 7 | 0.411921 | 0.430429 | 0.414118 | 0.407869 | 0.723855 |
| 50 | 3 | 0.411864 | 0.425060 | 0.399954 | 0.397782 | 0.686011 |
| 30 | 3 | 0.410734 | 0.422788 | 0.400917 | 0.399539 | 0.686727 |

Según los valores obtenidos, el mejor modelo sería con N=7, PCA=30:

* **Accuracy:** 0.416206
* **Precision:** 0.435028
* **Recall:** 0.435433
* **f1:** 0.420852

Además, sus curvas roc\_auc y su matriz de confusión se pueden ver a continuación:

Por último, se muestran 10 imágenes con su etiqueta real y su etiqueta predicha por el modelo:

*Imagen de la pantalla de un celular con texto e imágenes de una flor

El contenido generado por IA puede ser incorrecto.*

## ÁRBOL DE DECISIÓN

El modelo de *Árbol decisión VGG16* obtuvo unos resultados durante la validación cruzada de:

* **Accuracy:** 0.460734
* **Precision:** 0.458119
* **Recall:** 0.451168
* **f1:** 0.451107

Además, sus curvas roc\_auc y su matriz de confusión se pueden ver a continuación:

Por último, se muestran 10 imágenes con su etiqueta real y su etiqueta predicha por el modelo:

*Una captura de pantalla de un celular con la imagen de una flor amarilla

El contenido generado por IA puede ser incorrecto.*

El modelo de *Árbol decisión VGG16 + PCA* obtuvo unos resultados durante la validación cruzada de:

* **Accuracy:** 0.418927
* **Precision:** 0.412061
* **Recall:** 0.412584
* **f1:** 0.411376

Además, sus curvas roc\_auc y su matriz de confusión se pueden ver a continuación:

Por último, se muestran 10 imágenes con su etiqueta real y su etiqueta predicha por el modelo:



## RANDOM FOREST

El modelo de *Random Forest* obtuvo unos resultados durante la validación cruzada de:

* **Accuracy:** 0.539831
* **Precision:** 0.546087
* **Recall:** 0.528290
* **f1:** 0.529463

Además, sus curvas roc\_auc y su matriz de confusión se pueden ver a continuación:

Por último, se muestran 10 imágenes con su etiqueta real y su etiqueta predicha por el modelo:



El modelo de *Random Forest VGG16* obtuvo unos resultados durante la validación cruzada de:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | n\_components | precision | recall | f1 | accuracy | roc\_auc |
| **Random forest VGG16 500** | **500** | **0,72676293** | **0,6972475** | **0,70131685** | **0,71327684** | **0,91560212** |
| Random forest VGG16 250 | 250 | 0,64205419 | 0,58876474 | 0,59539319 | 0,60875706 | 0,85691674 |

Según los valores obtenidos, el mejor modelo sería con n\_components = 500:

* **Accuracy:** 0.713277
* **Precision:** 0.726763
* **Recall:** 0.697247
* **f1:** 0.701317

Además, sus curvas roc\_auc y su matriz de confusión se pueden ver a continuación:

Por último, se muestran 10 imágenes con su etiqueta real y su etiqueta predicha por el modelo:

Una captura de pantalla de un celular con texto e imágenes

El contenido generado por IA puede ser incorrecto.

El modelo de *Random Forest VGG16 + HOG* obtuvo unos resultados durante la validación cruzada de:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | n\_components | precision | recall | f1 | accuracy | roc\_auc |
| **Random forest HOG 500** | **500** | **0,50378189** | **0,47564515** | **0,47654443** | **0,4940678** | **0,77343827** |
| Random forest HOG 250 | 250 | 0,50414202 | 0,47366143 | 0,47369658 | 0,49378531 | 0,77537725 |

Según los valores obtenidos, el mejor modelo sería con n\_components = 500:

* **Accuracy:** 0.494068
* **Precision:** 0.503782
* **Recall:** 0.475645
* **f1:** 0.476544

Además, sus curvas roc\_auc y su matriz de confusión se pueden ver a continuación:

Por último, se muestran 10 imágenes con su etiqueta real y su etiqueta predicha por el modelo:

Imagen que contiene Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

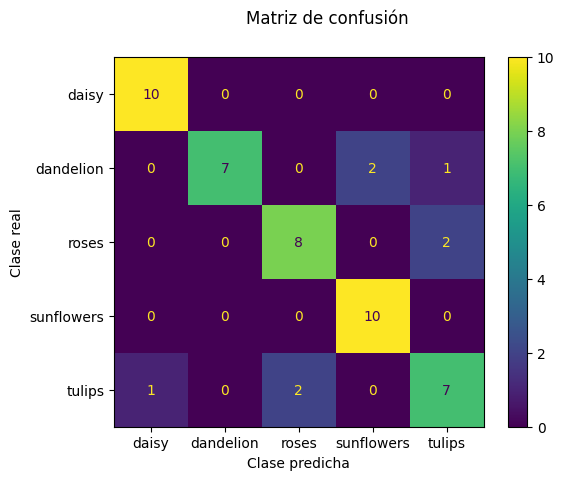


## VGG16

## MOBILENETV2

El modelo de *MobilenetV2* obtuvo unos resultados durante la validación de:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Modelo | Tamaño | dropout\_rate | learning\_rate | Precision |
| **1** | **256** | **0.5** | **0.001** | **0.86** |
| 2 | 64 | 0.5 | 0.001 | 0.66 |
| 3 | 64 | 0.5 | 0.0001 | 0.64 |
| 4 | 64 | 0.3 | 0.001 | 0.62 |



## ANN + HOG

El modelo de *ANN + HOG* obtuvo unos resultados durante la validación de:

## SVM + HOG

El modelo de *SVM + HOG* obtuvo unos resultados durante la validación de:

# ANÁLISIS COMPARATIVO

En esta sección se presentan y analizan las métricas utilizadas para evaluar el desempeño de los modelos implementados. Se han considerado cuatro métricas clave: accuracy, recall, precisión y F1-score, que permitirán evaluar el rendimiento desde diferentes perspectivas.

Cada una de las métricas proporciona información relevante sobre el desempeño del modelo:

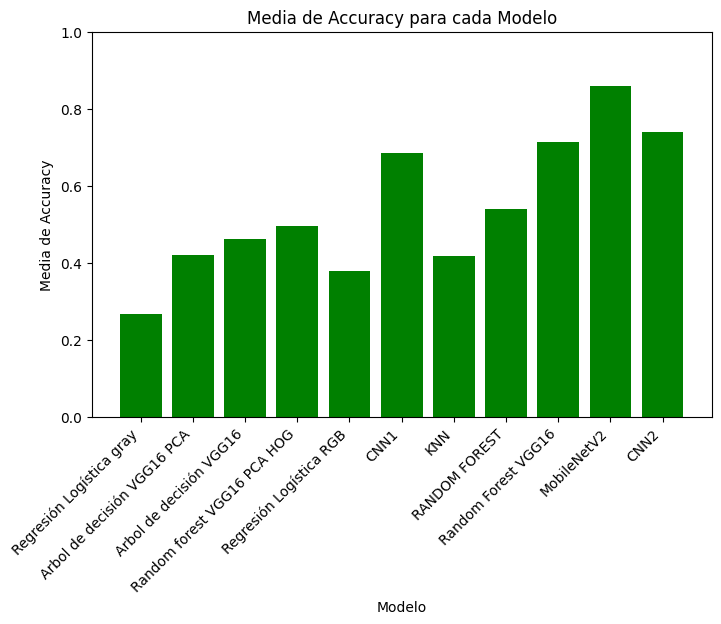
* Accuracy: mide la proporción de predicciones correctas sobre el total de instancias evaluadas
* Recall evalúa la capacidad del modelo para identificar correctamente las instancias positivas, siendo fundamental en problemas donde los falsos negativos tienen un alto costo.
* Precisión indica cuántas de las instancias clasificadas como positivas son realmente correctas.
* F1-score es una métrica que combina precisión y recall en una única medida equilibrada.



## Accuracy

En la gráfica comparativa se puede observar:

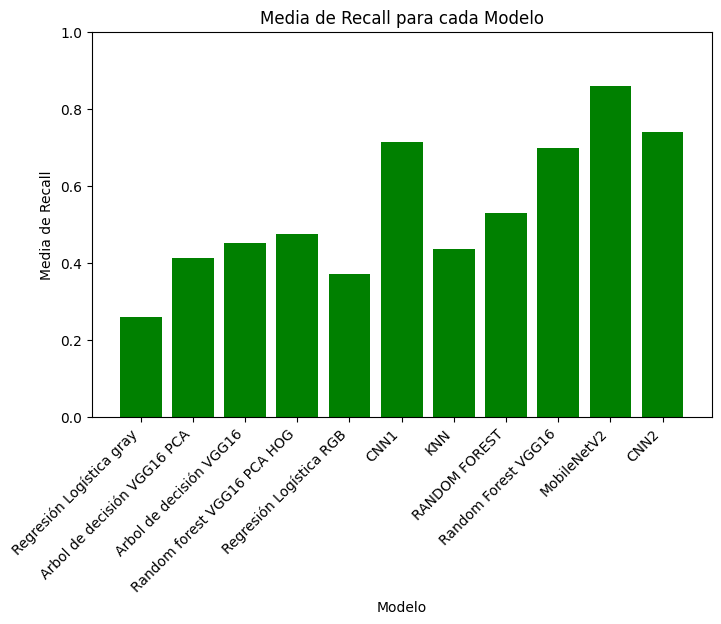
1. Modelos tradicionales como Regresión Logística y Árbol de decisión presentan valores bajos, por lo que presentan dificultades para generalizar los patrones
2. Los modelos basados en extracción de características mejorar los valores, lo que indica que el uso de estas técnicas mejora el rendimiento
3. Los modelos con redes neuronales profundas presentan los valores más alto. Esto sugiere que son capaces de aprender y hacer predicciones más precisas.



## Recall

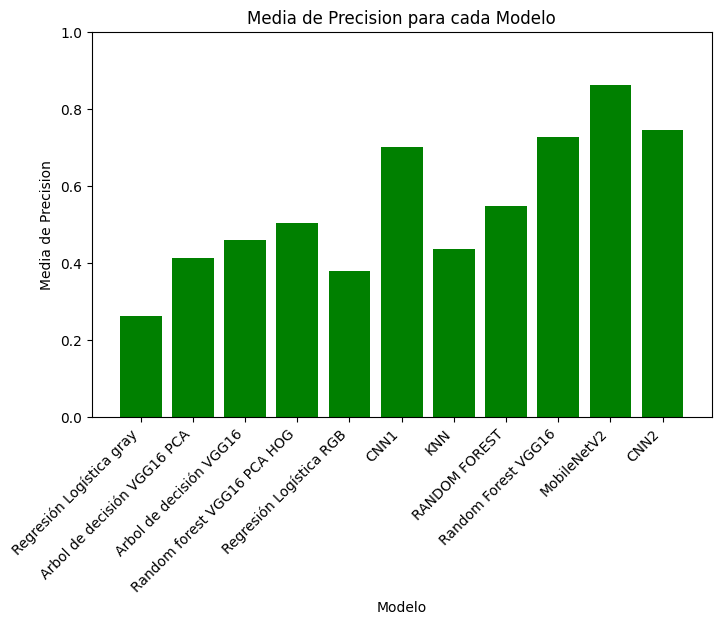
En la gráfica comparativa se puede observar:

1. Los modelos tradicionales presentan valores bajos, por lo que no logran identificar una gran proporción de los casos positivos
2. Los modelos basados en extracción de características mejoran los valores
3. Las redes neuronales profundas presentan los valores más altos.



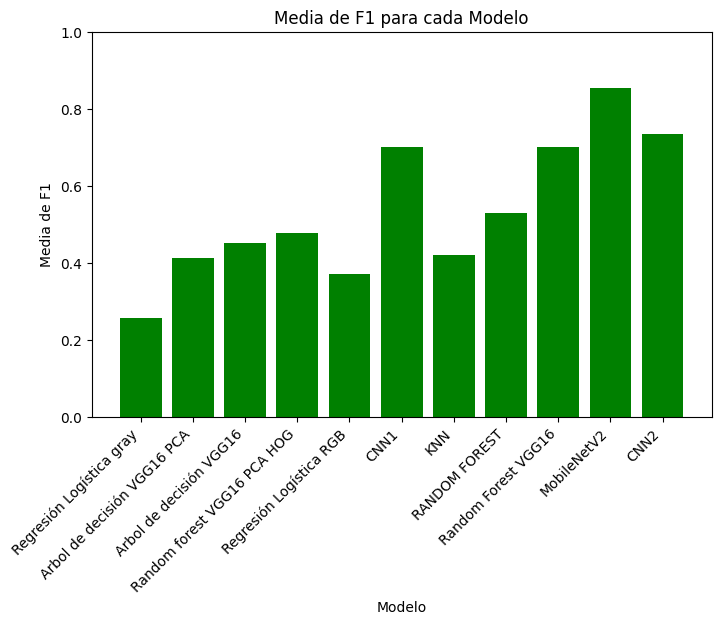
## Precision

En la gráfica comparativa se puede observar:



## F1-score

En la gráfica comparativa se puede observar:



# CONCLUSIONES