

---

Aplicación móvil para la gestión de residuos  
y reducción de la huella de carbono

---

Inteligencia Artificial

---

Pedro Arriola



UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Aplicación móvil para la gestión de residuos y  
reducción de la huella de carbono**

**Inteligencia Artificial**

Trabajo de graduación en modalidad de Megaproyecto presentado por  
Pedro Arriola

Para optar al grado académico de Licenciado en Ingeniería en Ciencias  
de la Computación y Tecnologías de la Información

Guatemala, noviembre del 2024

Vo.Bo.:

(f) \_\_\_\_\_  
MSc, MBA, Ing. Alberto Suriano

Tribunal Examinador:

(f) \_\_\_\_\_  
MSc, MBA, Ing. Alberto Suriano

(f) \_\_\_\_\_  
Ing. Javier Fong

(f) \_\_\_\_\_  
Ing. Eddy Omar Castro Jauregui

Fecha de aprobación: Guatemala, 28 de noviembre del 2024.

La gestión de residuos y la promoción de prácticas sostenibles se han convertido en temas de gran relevancia a nivel global, especialmente en el contexto de la crisis ambiental que enfrenta nuestro planeta. A lo largo de los últimos años, he observado cómo el crecimiento urbano en ciudades como Guatemala ha generado desafíos significativos en la administración y disposición adecuada de los residuos, afectando no solo el medio ambiente, sino también la calidad de vida de los habitantes. Este proyecto surge de una profunda motivación por aportar una solución innovadora y práctica a estos problemas, aprovechando los avances en inteligencia artificial y tecnología.

El presente trabajo representa el esfuerzo por desarrollar un sistema integrado de clasificación de residuos mediante un modelo de *deep learning* y un chatbot de asistencia basado en inteligencia artificial, con el objetivo de facilitar el reciclaje y la disposición correcta de desechos en la Ciudad de Guatemala. La elección de este proyecto no fue casual; responde a la necesidad de encontrar soluciones aplicables y escalables que contribuyan al bienestar de la sociedad y al cuidado del entorno, promoviendo además la concienciación y educación ambiental en la comunidad.

El desarrollo de este megaproyecto implicó retos significativos, no solo en términos técnicos y de investigación, sino también en la colaboración con instituciones locales y en la integración de normativas y prácticas de reciclaje en la solución propuesta. Fue necesario un aprendizaje constante y una adaptabilidad para implementar modelos de aprendizaje profundo y modelos de lenguaje extensos (LLM) en un contexto específico como el de Guatemala. A lo largo del proceso, he tenido la oportunidad de profundizar en temas de visión por computadora, procesamiento de lenguaje natural y diseño de agentes conversacionales, lo cual ha sido una experiencia enriquecedora tanto a nivel académico como personal.

Agradezco profundamente el apoyo de mis asesores, colegas y la Unidad de Control Ambiental de la Municipalidad de Guatemala, quienes facilitaron el acceso a información clave y brindaron el respaldo necesario para llevar este proyecto a buen término. Su colaboración y orientación han sido esenciales para la culminación de esta propuesta, la cual espero que contribuya a un cambio positivo en la gestión de residuos y en la concienciación ambiental en nuestra ciudad.

Este megaproyecto no solo representa una meta alcanzada en mi formación académica, sino también un compromiso con el desarrollo sostenible y la innovación tecnológica. Confío en que los resultados obtenidos y las recomendaciones propuestas sentarán las bases para futuras investigaciones y aplicaciones de inteligencia artificial en el ámbito ambiental, y que este trabajo pueda ser un paso más hacia un modelo de sociedad más consciente y responsable con el medio ambiente.

---

## Agradecimientos

---

Quiero expresar mi más profundo agradecimiento a todas las personas que me han acompañado y apoyado a lo largo de este camino:

A Dios, quien ha sido mi guía, mi paz, y quien me ha provisto con todas las bendiciones que hoy en día tengo. Agradezco su infinita misericordia y amor, que me han dado la fortaleza para superar cada desafío.

A mi mamá, Piedad de María Jiménez, que es mi pilar y la persona que siempre me ha apoyado incondicionalmente. Gracias por brindarme una excelente educación superior y por ser un ejemplo de fortaleza y amor.

A mi papá, Pablo Enrique Arriola Reyes, quien aunque no se encuentre físicamente con nosotros, su espíritu y consejos han perdurado a lo largo de los años. Su memoria ha sido mi fuerza y motivo para seguir adelante, y siempre le estaré agradecido.

A mis tíos, Juan Girón, Juana Macdonald y Ricardo Hamm, por sus invaluable consejos y apoyo incondicional a lo largo de mi carrera. Les estoy infinitamente agradecido.

A mi grupo de amigos: Oscar López, Rodrigo Barrera, Marco Orozco, Gabriel Vicente, Yongbum Park y Santiago Taracena. Gracias por ser excelentes compañeros y amigos, y por habernos esforzado y apoyado mutuamente durante toda la carrera. Nuestra amistad es invaluable y ha sido clave en mi éxito académico.

A mi amigo de toda la vida, José Javier Hurtarte, por ser un gran apoyo a lo largo de mi niñez y un amigo a quien valoro mucho. Egresado de la Universidad del Valle de Guatemala, con quien también compartimos anécdotas sobre la carrera que ambos estudiamos.

A mi novia, Carolina Monroy, mi segundo pilar y a quien amo profundamente. Ella ha sido un gran apoyo en los últimos años, y agradezco mucho su amor y buenos deseos en esta etapa tan importante de mi vida.

A mis primos, José Pablo Girón y Ricardo Martínez, a quienes considero más como mis hermanos. Hemos estado juntos en las buenas y en las malas. Les debo mucho, y su apoyo ha sido otro pilar fundamental en mi vida.

A todos ustedes, les expreso mi más sincero agradecimiento. Este logro no habría sido posible sin su apoyo y cariño.

<b>Prefacio</b>	<b>III</b>
<b>Agradecimientos</b>	<b>IV</b>
<b>Lista de Figuras</b>	<b>XIII</b>
<b>Lista de Cuadros</b>	<b>XIV</b>
<b>Resumen</b>	<b>XVI</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Objetivos</b>	<b>2</b>
2.1. Objetivo General . . . . .	2
2.2. Objetivos Específicos . . . . .	2
<b>3. Justificación</b>	<b>3</b>
<b>4. Marco Teórico</b>	<b>4</b>
4.1. Aspectos Legales y Ambientales en Guatemala . . . . .	4
4.1.1. Legislación sobre Manejo de Residuos y Reciclaje en Guatemala . . . . .	4
4.1.2. Impacto Ambiental del Manejo de Residuos en Guatemala . . . . .	5
4.1.3. Estrategias Internacionales y su Aplicación en Guatemala . . . . .	5
4.1.4. Crisis de Desechos en Guatemala . . . . .	5
4.1.5. Desafíos Socioeconómicos en la Gestión de Residuos . . . . .	6
4.1.6. Educación y Sensibilización Ambiental . . . . .	6
4.1.7. Impacto del Cambio Climático y la Gestión de Residuos . . . . .	6
4.2. Reciclaje y Prácticas Sostenibles en la Gestión de Residuos . . . . .	6
4.2.1. Tipos de Residuos . . . . .	6
4.2.2. Prácticas Sostenibles en el Reciclaje . . . . .	7
4.2.3. Impacto del Reciclaje en la Mitigación del Cambio Climático . . . . .	8
4.2.4. Efectos Nocivos de los Residuos y Gases de Efecto Invernadero . . . . .	8
4.2.5. Políticas y Legislación Ambiental Relacionadas con el Reciclaje . . . . .	8
4.2.6. Iniciativas y Proyectos de Reciclaje en Guatemala . . . . .	9
4.2.7. Desafíos Logísticos y de Infraestructura en el Reciclaje . . . . .	9
4.2.8. Impacto Socioeconómico del Reciclaje . . . . .	9
4.2.9. Impacto Ambiental y en la Salud Humana . . . . .	9
4.2.10. Residuos Electrónicos y su Gestión . . . . .	10

4.2.11. Cambio Climático y la Gestión de Residuos . . . . .	10
4.3. Tecnología: Machine Learning, Inteligencia Artificial y LLMs . . . . .	10
4.3.1. Conceptos Clave de Machine Learning e Inteligencia Artificial . . . . .	10
4.3.2. Modelos de Lenguaje de Gran Escala (LLMs) . . . . .	11
4.3.3. Aplicaciones de Machine Learning e Inteligencia Artificial en la Gestión de Residuos . . . . .	11
4.3.4. Estudios y Ejemplos Relevantes . . . . .	12
4.3.5. Retos y Futuro del Uso de IA, ML y LLMs en la Gestión de Residuos . . . . .	12
4.4. Leyes, Regulaciones y Ética en el Uso de la Inteligencia Artificial . . . . .	12
4.4.1. Leyes y Regulaciones sobre el Uso de la Inteligencia Artificial . . . . .	13
4.4.2. Ética en el Uso de la Inteligencia Artificial . . . . .	13
4.4.3. Responsabilidad y Gobernanza de la IA . . . . .	14
4.5. Impacto Socioeconómico de la IA en la Gestión de Residuos . . . . .	14
4.5.1. Creación de Empleo y Economía Circular . . . . .	14
4.5.2. Desigualdades Digitales . . . . .	14
4.6. Futuro y Tendencias en IA para la Gestión de Residuos . . . . .	15
4.6.1. Desarrollos Emergentes . . . . .	15
4.6.2. Automatización Completa de Procesos . . . . .	15
4.7. Desafíos Técnicos en la Implementación de IA y ML . . . . .	15
4.7.1. Limitaciones Tecnológicas . . . . .	15
4.7.2. Seguridad y Resiliencia de los Sistemas de IA . . . . .	15
4.8. Evaluación de Impacto y Sostenibilidad . . . . .	15
4.8.1. Evaluación de Ciclo de Vida (LCA) Aplicada a Sistemas de IA . . . . .	16
4.8.2. Sostenibilidad y Eficiencia Energética . . . . .	16
4.9. Interdisciplinariedad en el Desarrollo de IA . . . . .	16
4.9.1. Colaboración Multidisciplinaria . . . . .	16
4.9.2. Educación y Formación en IA y Sostenibilidad . . . . .	16
4.10. Aspectos Técnicos de <i>Machine Learning</i> . . . . .	16
4.10.1. Modelos de <i>Machine Learning</i> Aplicables . . . . .	16
4.10.2. Hiperparámetros en <i>Machine Learning</i> . . . . .	17
4.10.3. Ecuaciones Matemáticas Clave . . . . .	17
4.10.4. Métricas de Desempeño en <i>Machine Learning</i> . . . . .	18
4.10.5. Visualización de Modelos y Resultados . . . . .	19
4.10.6. Regularización y Penalización en Modelos . . . . .	20
4.10.7. Algoritmo de Optimización Adam . . . . .	20
4.10.8. Regularización mediante <i>Dropout</i> . . . . .	20
4.10.9. Funciones de Activación . . . . .	21
4.10.10. Evaluación del Sesgo y la Varianza . . . . .	21
4.10.11. Métodos de Validación y Evaluación . . . . .	21
4.11. Aspectos Técnicos de la <i>Inteligencia Artificial</i> y <i>Deep Learning</i> . . . . .	21
4.11.1. Redes Neuronales Profundas ( <i>Deep Neural Networks</i> ) . . . . .	21
4.11.2. Redes Neuronales Convolucionales ( <i>Convolutional Neural Networks, CNNs</i> ) . . . . .	22
4.11.3. Redes Neuronales Recurrentes ( <i>Recurrent Neural Networks, RNNs</i> ) . . . . .	23
4.11.4. Funciones de Activación en <i>Deep Learning</i> . . . . .	23
4.11.5. Optimización y Entrenamiento en <i>Deep Learning</i> . . . . .	23
4.11.6. Regularización en Redes Profundas . . . . .	24
4.11.7. Transferencia de Aprendizaje ( <i>Transfer Learning</i> ) . . . . .	24
4.11.8. Modelos de Transfer Learning en Clasificación de Imágenes . . . . .	26
4.11.9. Evaluación y Métricas de Desempeño . . . . .	33
4.12. Explainable AI (XAI) . . . . .	34
4.12.1. Definición y Objetivo de <i>Explainable AI</i> . . . . .	34
4.12.2. Importancia de la Interpretabilidad . . . . .	34
4.12.3. Métodos de <i>Explainable AI</i> . . . . .	34
4.12.4. Desafíos en <i>Explainable AI</i> . . . . .	37

4.12.5. Aplicaciones de <i>Explainable AI</i> . . . . .	37
4.13. Modelos de <i>Machine Learning</i> e <i>Inteligencia Artificial</i> en la Gestión de Residuos . .	37
4.13.1. RecycleNet: Clasificación Automática de Residuos . . . . .	37
4.13.2. Sistema de Gestión Inteligente de Residuos de SAS . . . . .	37
4.13.3. Project Recicla AI . . . . .	38
4.13.4. Aplicación Web para la Clasificación de Residuos . . . . .	38
4.13.5. WasteNet: Modelo de Clasificación de Residuos en <i>Smart Cities</i> . . . . .	38
4.13.6. EcoSort: Optimización de Rutas de Recolección de Residuos . . . . .	38
4.13.7. AI-Waste: Clasificación Avanzada de Residuos Electrónicos . . . . .	38
4.13.8. Consideraciones Finales . . . . .	38
4.14. Visión por Computadora para Clasificación de Imágenes . . . . .	39
4.14.1. Conceptos Clave en Visión por Computadora . . . . .	39
4.14.2. Extracción de Características . . . . .	39
4.14.3. Clasificación de Imágenes . . . . .	40
4.14.4. Relación con <i>Machine Learning</i> e <i>Inteligencia Artificial</i> . . . . .	40
4.14.5. Optimización y Entrenamiento de Modelos de Visión por Computadora . . .	40
4.14.6. Visualización de Características y Mapas de Calor . . . . .	41
4.14.7. Consideraciones Finales . . . . .	41
4.15. Modelos de Lenguaje de Gran Escala ( <i>Large Language Models, LLMs</i> ) y Chatbots Inteligentes . . . . .	41
4.15.1. Conceptos Clave en LLMs . . . . .	42
4.15.2. Implementación de Chatbots Inteligentes con LLMs . . . . .	42
4.15.3. Ejemplos de LLMs Poderosos para Chatbots . . . . .	43
4.15.4. Implicaciones y Consideraciones Éticas en el Uso de LLMs . . . . .	43
4.15.5. Visualización de la Atención en LLMs . . . . .	43
4.16. Tecnologías para la Implementación de Modelos de <i>Inteligencia Artificial</i> . . . . .	44
4.16.1. Python . . . . .	44
4.16.2. Kaggle . . . . .	44
4.16.3. Jupyter Notebooks . . . . .	44
4.16.4. Google Colab Pro . . . . .	44
4.16.5. Librerías de Python . . . . .	45
4.16.6. CUDA y el Uso de Tarjetas Gráficas . . . . .	45
4.16.7. Uso de Datasets para Entrenamiento . . . . .	46
4.16.8. Relevancia de Kaggle como Repositorio de Datasets . . . . .	46
4.17. TensorFlow y Visión por Computadora con Redes Neuronales Convolucionales (CNNs) .	46
4.17.1. Redes Neuronales Convolucionales (CNNs) . . . . .	47
4.17.2. Implementación de CNNs en TensorFlow . . . . .	47
4.17.3. Ventajas de Usar TensorFlow en Visión por Computadora . . . . .	48
4.17.4. Casos de Uso en Visión por Computadora . . . . .	48
4.17.5. Optimización y Despliegue de Modelos en Producción . . . . .	48
4.17.6. Conclusiones . . . . .	48
4.18. Hugging Face y su Rol en la Construcción de Chatbots Inteligentes . . . . .	48
4.18.1. Acceso a Modelos Preentrenados . . . . .	49
4.18.2. Implementación de Chatbots con Hugging Face . . . . .	49
4.18.3. Facilidades Adicionales de Hugging Face . . . . .	49
4.18.4. Integraciones con Google Colab y AWS . . . . .	50
4.18.5. Consideraciones Éticas en el Uso de LLMs . . . . .	50
4.18.6. Relevancia para la Construcción de Chatbots Inteligentes . . . . .	50
4.19. OpenAI y su API para Modelos de Lenguaje . . . . .	50
4.19.1. API de OpenAI . . . . .	50
4.19.2. GPT-3.5-turbo: Características Técnicas y Arquitectura . . . . .	51
4.19.3. Ventajas de GPT-3.5-turbo en Procesamiento de Lenguaje Natural . . . . .	52
4.20. FastAPI y Jinja2: Herramientas para el Desarrollo de Aplicaciones Web . . . . .	52
4.21. MLOps y MLFlow . . . . .	54



4.21.1. Concepto de MLOps . . . . .	54
4.21.2. MLFlow: Herramienta de MLOps . . . . .	55
<b>5. Antecedentes</b>	<b>57</b>
5.0.1. Clasificación de Residuos mediante Redes Neuronales Convolucionales . . . . .	57
5.0.2. Importancia de la Inteligencia Artificial en la Gestión de Residuos . . . . .	57
5.0.3. Implementación de Chatbots Inteligentes para la Asistencia en el Reciclaje . . . . .	58
5.0.4. Desafíos en la Clasificación de Residuos y la Asistencia Basada en IA . . . . .	58
<b>6. Alcance</b>	<b>59</b>
<b>7. Metodología</b>	<b>61</b>
7.1. Selección de Tecnologías . . . . .	61
7.1.1. Plataforma de Entrenamiento y Desarrollo: Google Colab . . . . .	61
7.1.2. Framework de Redes Neuronales: TensorFlow . . . . .	62
7.1.3. Manipulación y Preprocesamiento de Imágenes: PILLOW . . . . .	63
7.1.4. Modelo de Chat Inteligente: OpenAI API con GPT-3.5 Turbo . . . . .	63
7.1.5. Framework para Endpoints de Chat: FastAPI . . . . .	63
7.1.6. Resumen de Ventajas de las Tecnologías Seleccionadas . . . . .	64
7.2. Descripción del Dataset Utilizado . . . . .	64
7.2.1. Características Principales del Dataset . . . . .	64
7.2.2. Relevancia del Dataset en el Contexto Guatemalteco . . . . .	65
7.3. Ejemplos Visuales de Imágenes en el Dataset . . . . .	65
7.4. Justificación del Uso de Redes Neuronales Convolucionales (CNN) y Transfer Learning . . . . .	66
7.4.1. Ventajas del Uso de CNNs en Clasificación de Imágenes . . . . .	66
7.4.2. Transfer Learning para Mejorar la Precisión del Modelo . . . . .	66
7.5. Justificación del Uso de Redes Neuronales Convolucionales (CNN), Transfer Learning y Fine-Tuning . . . . .	67
7.5.1. Ventajas del Uso de CNNs en Clasificación de Imágenes . . . . .	67
7.5.2. Transfer Learning y Fine-Tuning para Mejorar la Precisión del Modelo . . . . .	67
7.6. Inspiración en Estudios Anteriores . . . . .	68
7.6.1. Estudios Relacionados y Modelos Similares . . . . .	68
7.6.2. Aplicación de las Metodologías Inspiradas . . . . .	68
7.7. Metodología para la Selección de Modelos y Preparación de Datos . . . . .	68
7.7.1. Preparación del Dataset: Undersampling y Preprocesamiento . . . . .	69
7.7.2. Preprocesamiento Específico de Imágenes para Cada Modelo . . . . .	69
7.7.3. Implementación y Código para la Preparación de Datos de Entrenamiento . . . . .	70
7.7.4. Visualización de Imágenes Preprocesadas para Cada Modelo . . . . .	71
7.8. Definición y Configuración de la Arquitectura de los Modelos de Transfer Learning . . . . .	73
7.8.1. Arquitectura Base y Capas Congeladas . . . . .	73
7.8.2. Capas Superiores (Top Model) . . . . .	73
7.8.3. Implementación y Configuración de Cada Modelo . . . . .	74
7.8.4. Justificación de la Arquitectura y las Capas Superiores . . . . .	76
7.9. Entrenamientos Preliminares para la Selección del Modelo Definitivo . . . . .	76
7.9.1. Hiperparámetros de Entrenamiento . . . . .	77
7.9.2. Optimizador y Tasa de Aprendizaje . . . . .	77
7.9.3. Uso de GPU para Acelerar el Entrenamiento . . . . .	77
7.9.4. Callbacks Utilizados . . . . .	77
7.9.5. Evaluación de Rendimiento y Selección del Modelo Definitivo . . . . .	78
7.9.6. Resumen de Hiperparámetros y Estrategias . . . . .	78
7.10. Análisis de Métricas de Rendimiento y Selección del Modelo Final . . . . .	78
7.10.1. Evaluación de Métricas de Rendimiento . . . . .	78
7.10.2. Determinación de la Métrica Clave . . . . .	79
7.10.3. Selección de Xception como Modelo Final . . . . .	79

7.10.4. Conclusiones de la Validación y Selección del Modelo Final . . . . .	80
7.11. Entrenamiento Final con el Modelo Xception . . . . .	80
7.11.1. Evaluación Visual y Análisis del Modelo Xception . . . . .	82
7.11.2. Estrategias de Interpretabilidad . . . . .	83
7.11.3. Conclusión de la Metodología . . . . .	85
7.12. Metodología de Implementación y Evaluación del Chatbot Inteligente . . . . .	85
7.12.1. Implementación del Chatbot Inteligente . . . . .	85
7.12.2. Evaluación del Rendimiento del Chatbot . . . . .	86
7.12.3. Interfaz Gráfica para Pruebas Externas . . . . .	88
7.12.4. Conclusión de la Metodología de Evaluación . . . . .	90
<b>8. Resultados</b>	<b>91</b>
8.1. Resultados de Xception . . . . .	91
8.1.1. Historial de Precisión y Pérdida . . . . .	91
8.1.2. Matriz de Confusión y Reporte de Clasificación . . . . .	92
8.1.3. Prueba de Concepto de MLOps con MLFlow . . . . .	93
8.1.4. Resultados de Explainable AI: GradCAM y LIME . . . . .	95
8.2. Resultados del Chatbot Inteligente . . . . .	111
8.2.1. Resultados de la Encuesta por Criterio . . . . .	111
8.2.2. Puntaje Total y Precisión General . . . . .	114
8.2.3. Tiempos de Respuesta del Chatbot . . . . .	114
8.2.4. Pruebas de Preguntas y Respuestas del Chatbot . . . . .	115
<b>9. Análisis de Resultados</b>	<b>117</b>
9.1. Análisis de Resultados del Modelo Xception . . . . .	117
9.1.1. Análisis de las Métricas de Precisión y Pérdida . . . . .	117
9.1.2. Análisis de la Matriz de Confusión . . . . .	118
9.1.3. Análisis del Reporte de Clasificación . . . . .	118
9.1.4. Análisis de Explainable AI: GradCAM y LIME . . . . .	119
9.1.5. Análisis de Saliency Maps con SmoothGrad . . . . .	121
9.1.6. Análisis y Discusión de Resultados con Imágenes Reales . . . . .	123
9.1.7. Análisis de la Implementación de MLOps con MLFlow . . . . .	125
9.1.8. Opiniones Finales sobre el Modelo Xception . . . . .	127
9.2. Análisis de Resultados del Chatbot Inteligente . . . . .	128
9.2.1. Evaluación de la Precisión y Relevancia de las Respuestas . . . . .	128
9.2.2. Capacidad para Rechazar Preguntas Fuera de Tema . . . . .	128
9.2.3. Calidad de las Recomendaciones . . . . .	129
9.2.4. Consistencia en el Estilo de las Respuestas . . . . .	129
9.2.5. Evaluación del Tiempo de Respuesta . . . . .	129
9.2.6. Pruebas y Ejemplos de Interacciones . . . . .	130
9.2.7. Opiniones Finales sobre el Desempeño del Chatbot . . . . .	130
9.3. Implicaciones de los Agentes Inteligentes en la Ciudad de Guatemala . . . . .	130
9.3.1. Facilitando la Gestión Eficiente de Residuos . . . . .	131
9.3.2. Educación y Concienciación Ambiental a través del Chatbot Inteligente . . . . .	131
9.3.3. Promoviendo Hábitos Sostenibles en la Población . . . . .	131
9.3.4. Sostenibilidad y Futuro de la Gestión de Residuos . . . . .	132
9.3.5. Desafíos y Consideraciones Finales . . . . .	132
<b>10. Conclusiones</b>	<b>133</b>
<b>11. Recomendaciones</b>	<b>135</b>
<b>Bibliografía</b>	<b>143</b>
<b>Anexos</b>	<b>144</b>

<b>A. Código para Cálculo de SmoothGrad</b>	<b>144</b>
<b>B. Código para Cálculo de Precisión del Chatbot Inteligente</b>	<b>146</b>
<b>C. Especificaciones Técnicas de la Tarjeta NVIDIA A100</b>	<b>148</b>

---

## Lista de Figuras

---

4.1. Matriz de confusión que ilustra la precisión del clasificador a través de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. . . . .	19
4.2. Curva ROC que muestra el desempeño de un clasificador binario. . . . .	19
4.3. Diagrama de árbol de decisión que representa las decisiones y sus posibles consecuencias. . . . .	19
4.4. Arquitectura del modelo MobileNetV2. . . . .	28
4.5. Arquitectura del modelo InceptionV3. . . . .	29
4.6. Arquitectura del modelo DenseNet201. . . . .	31
4.7. Arquitectura del modelo VGG19. . . . .	32
4.8. Arquitectura del modelo Xception. . . . .	33
4.9. Ejemplo de un Árbol de Decisión utilizado para la Interpretabilidad. . . . .	35
4.10. Ejemplo de Grad-CAM mostrando las áreas de atención del modelo. . . . .	35
4.11. Ejemplo de LIME explicando las decisiones de un modelo de clasificación. . . . .	36
4.12. Ejemplo de SHAP mostrando la contribución de cada variable en una predicción. . . . .	36
4.13. Ejemplo de SmoothGrad reduciendo el ruido en los mapas de saliencia. . . . .	36
4.14. Ejemplo de un mapa de calor que muestra las áreas de una imagen que más contribuyen a la clasificación. . . . .	41
4.15. Ejemplo de un mapa de atención en un LLM, mostrando la relevancia de diferentes palabras en un contexto específico. . . . .	43
7.1. Ejemplos de imágenes de cada clase en el dataset: plástico, papel, orgánico, metal, vidrio y baterías. . . . .	65
7.2. Imágenes preprocesadas para MobileNetV2. . . . .	71
7.3. Imágenes preprocesadas para InceptionV3. . . . .	72
7.4. Imágenes preprocesadas para DenseNet201. . . . .	72
7.5. Imágenes preprocesadas para VGG19. . . . .	72
7.6. Imágenes preprocesadas para Xception. . . . .	73
7.7. Imágenes reales utilizadas para probar el modelo Xception en un contexto cotidiano. . . . .	83
7.8. Saliency maps generados con SmoothGRAD para interpretar las decisiones del modelo Xception en distintas clases de residuos. . . . .	84
7.9. Interfaz gráfica desarrollada para pruebas del chatbot. . . . .	88
8.1. Evolución de la Precisión del Modelo durante el Entrenamiento y Validación (Xception Final). . . . .	92
8.2. Evolución de la Pérdida del Modelo durante el Entrenamiento y Validación (Xception Final). . . . .	92
8.3. Matriz de Confusión del Modelo Xception en el Conjunto de Prueba. . . . .	93

8.4. Interfaz de MLFlow mostrando los experimentos registrados. Aquí se observan las ejecuciones del modelo Xception y sus correspondientes métricas y parámetros. . . .	94
8.5. Detalles del experimento "Xception Model Training". <sup>en</sup> MLFlow. Se observan las métricas y los parámetros registrados durante el entrenamiento del modelo. . . . .	94
8.6. Detalles del experimento "Entrenamiento con Xception y GradCAM". <sup>en</sup> MLFlow. Se destacan las métricas de precisión y pérdida, junto con los parámetros utilizados. . .	95
8.7. Ejemplos combinados de mapas de atención generados con GradCAM para las diferentes clases de residuos. Se observan las áreas de atención del modelo, que se enfocan en las características clave de cada residuo, como bordes, texturas y patrones específicos, esenciales para la correcta clasificación. . . . .	96
8.8. Explicación generada con LIME para la clase <i>Battery</i> . Las áreas resaltadas muestran las características que más influyeron en la predicción, indicando bordes y regiones específicas. . . . .	97
8.9. Explicación generada con LIME para la clase <i>Glass</i> . Las áreas resaltadas indican las texturas y bordes que el modelo consideró importantes para clasificar correctamente el residuo. . . . .	98
8.10. Explicación generada con LIME para la clase <i>Metal</i> . Las áreas destacadas muestran las regiones clave que ayudaron al modelo a identificar las propiedades del metal, como el brillo y las texturas metálicas. . . . .	99
8.11. Explicación generada con LIME para la clase <i>Organic</i> . Las áreas resaltadas indican las texturas y patrones orgánicos que fueron más relevantes para la predicción. . . .	100
8.12. Explicación generada con LIME para la clase <i>Paper</i> . Las áreas importantes se centran en las texturas y bordes del papel que influenciaron la decisión del modelo. . . . .	101
8.13. Explicación generada con LIME para la clase <i>Plastic</i> . Las áreas resaltadas muestran las regiones específicas que el modelo utilizó para identificar el plástico, como los contornos y las variaciones en la textura. . . . .	102
8.14. Saliency Map generado con SmoothGrad para la clase <i>Battery</i> . . . . .	103
8.15. Saliency Map generado con SmoothGrad para la clase <i>Glass</i> . . . . .	103
8.16. Saliency Map generado con SmoothGrad para la clase <i>Metal</i> . . . . .	103
8.17. Saliency Map generado con SmoothGrad para la clase <i>Organic</i> . . . . .	104
8.18. Saliency Map generado con SmoothGrad para la clase <i>Paper</i> . . . . .	104
8.19. Saliency Map generado con SmoothGrad para la clase <i>Plastic</i> . . . . .	104
8.20. Clasificación de un residuo real perteneciente a la clase <i>Papel</i> . . . . .	105
8.21. Clasificación de un residuo real perteneciente a la clase <i>Metal</i> . . . . .	105
8.22. Clasificación de un residuo real perteneciente a la clase <i>Plastic</i> . . . . .	106
8.23. Clasificación de un residuo real perteneciente a la clase <i>Plastic</i> . . . . .	106
8.24. Clasificación de un residuo real perteneciente a la clase <i>Organic</i> . . . . .	106
8.25. Clasificación de un residuo real perteneciente a la clase <i>Glass</i> . . . . .	107
8.26. Clasificación de un residuo real perteneciente a la clase <i>Battery</i> . . . . .	107
8.27. Clasificación de un residuo real perteneciente a la clase <i>Paper</i> . . . . .	107
8.28. Clasificación de un residuo real perteneciente a la clase <i>Metal</i> . . . . .	108
8.29. Clasificación de un residuo real perteneciente a la clase <i>Organic</i> . . . . .	108
8.30. Clasificación de un residuo real perteneciente a la clase <i>Metal</i> . . . . .	108
8.31. Clasificación de un residuo real perteneciente a la clase <i>Plastic</i> . . . . .	109
8.32. Clasificación de un residuo real perteneciente a la clase <i>Glass</i> . . . . .	109
8.33. Clasificación de un residuo real perteneciente a la clase <i>Plastic</i> . . . . .	109
8.34. Clasificación de un residuo real perteneciente a la clase <i>Plastic</i> , pero que fue clasificado como <i>Paper</i> . . . . .	110
8.35. Clasificación de un residuo real perteneciente a la clase <i>Organic</i> . . . . .	110
8.36. Evaluación de la precisión de las respuestas del chatbot. . . . .	112
8.37. Evaluación de la relevancia temática de las respuestas. . . . .	112
8.38. Evaluación de la capacidad del chatbot para rechazar preguntas fuera de tema. . . .	113
8.39. Evaluación de la calidad de las recomendaciones proporcionadas. . . . .	113
8.40. Evaluación de la consistencia en el estilo de las respuestas. . . . .	114

8.41. Tiempo promedio de respuesta del chatbot. . . . .	115
---------------------------------------------------------	-----

---

## Lista de Cuadros

---

7.1. Métricas de rendimiento de cada modelo evaluado en el conjunto de validación . . .	79
7.2. Métricas de evaluación detalladas para el modelo Xception . . . . .	79
7.3. Distribución de datos de entrenamiento y validación . . . . .	81
8.1. Reporte de Clasificación del Modelo Xception en el Conjunto de Prueba. . . . .	93
8.2. Puntaje total y porcentaje de precisión de cada participante en la evaluación del chatbot. . . . .	114
8.3. Tiempos de respuesta del chatbot en cada prueba realizada. . . . .	115
8.4. Pruebas de interacción con el chatbot y sus respuestas. . . . .	116
9.1. Ejemplos de Interacciones con el Chatbot y su Análisis . . . . .	130
C.1. Especificaciones técnicas de la tarjeta NVIDIA A100. . . . .	149

Este megaproyecto presenta el desarrollo de un sistema de inteligencia artificial diseñado para optimizar la gestión de residuos en la Ciudad de Guatemala, integrando dos componentes principales: un modelo de clasificación de residuos mediante imágenes y un chatbot de asistencia especializado. El objetivo fue crear una herramienta efectiva y accesible que promueva prácticas sostenibles de reciclaje y manejo responsable de desechos, utilizando técnicas avanzadas de *deep learning* y modelos de lenguaje extenso (LLM).

El modelo de clasificación se construyó utilizando la arquitectura Xception, reconocida por su capacidad para extraer características complejas y su eficiencia en tareas de visión por computadora. Aplicando técnicas de *transfer learning* y ajustes finos (*fine-tuning*), el modelo fue entrenado para clasificar seis categorías de residuos: orgánicos, plásticos, papel, vidrio, metal y baterías. Durante el entrenamiento, se alcanzaron niveles de precisión sobresalientes, con una exactitud promedio del 98 % en el conjunto de prueba. Este rendimiento fue validado en condiciones de imagen variadas, demostrando la capacidad del modelo para operar en tiempo real y adaptarse a escenarios complejos, incluyendo fondos heterogéneos.

Complementando la clasificación de residuos, se desarrolló un chatbot inteligente basado en un modelo de lenguaje extenso, configurado para proporcionar asistencia en temas de reciclaje y clasificación de residuos conforme a las normativas locales, como el Acuerdo Gubernativo 164-2021. El chatbot no solo brinda respuestas precisas y relevantes, sino que también tiene la capacidad de rechazar educadamente consultas fuera del tema, asegurando interacciones significativas y enfocadas. La evaluación cualitativa del chatbot, realizada por expertos en gestión ambiental, arrojó una precisión promedio del 90.86 %, destacando su eficacia en la comunicación, relevancia temática y tiempo de respuesta.

Además, el proyecto incorporó principios de MLOps mediante una prueba de concepto (POC) con MLFlow, facilitando la gestión y rastreo de experimentos, versionamiento de modelos y monitoreo del rendimiento. Esta implementación subraya la importancia de las mejores prácticas en el desarrollo y mantenimiento de modelos de aprendizaje automático, con miras a escalabilidad y reproducibilidad.

Los resultados obtenidos validan el éxito del proyecto en cumplir con los objetivos planteados, ofreciendo una solución tecnológica integral para la clasificación de residuos y la educación ambiental. Su implementación en una plataforma móvil hace que el sistema sea accesible a una amplia audiencia, fomentando hábitos sostenibles en la población y apoyando las iniciativas de gestión de residuos en el contexto guatemalteco.

Por último, este proyecto representa un avance importante en el uso de inteligencia artificial para promover la sostenibilidad ambiental, demostrando el potencial de los agentes inteligentes para



transformar la gestión de residuos y fomentar la conciencia ambiental. Las recomendaciones para perfeccionar y extender las capacidades de los agentes establecen un marco sólido para futuras investigaciones y aplicaciones en otros contextos de reciclaje y gestión ambiental.

# CAPÍTULO 1

---

## Introducción

---

En el contexto actual de urgencia climática y desafíos medioambientales a nivel global, Guatemala se encuentra en una encrucijada crucial en cuanto a la gestión de residuos y la reducción de la huella de carbono. La nación enfrenta una serie de complejidades derivadas de su geografía diversa, su biodiversidad única y su desarrollo socioeconómico en curso. La creciente conciencia sobre los efectos devastadores del cambio climático, exacerbados por fenómenos como la deforestación, la contaminación del aire y la proliferación de desechos, ha generado una demanda urgente de soluciones tecnológicas innovadoras y efectivas que aborden estas problemáticas.

La implementación de la Inteligencia Artificial (IA) emerge como una herramienta estratégica y prometedora para abordar los desafíos específicos relacionados con la gestión de residuos y la reducción de la huella de carbono en Guatemala. Este proyecto utilizará dos modelos de deep learning: primero, un modelo de clasificación de residuos basado en Redes Neuronales Convolucionales (CNNs) que identificará y clasificará diferentes tipos de residuos mediante imágenes; y segundo, un sistema de chat inteligente que utiliza modelos de lenguaje profundo para proporcionar información y consejos sobre la gestión de residuos. Estos modelos, entrenados con datasets robustos como TrashNet y TACO, mejorarán significativamente la precisión y eficiencia de la gestión de residuos, empoderando a los ciudadanos guatemaltecos para realizar cambios significativos en sus hábitos de gestión de residuos [111].

Este proyecto, por tanto, no solo aborda una necesidad urgente en la sociedad guatemalteca, sino que también propone una solución tecnológica innovadora y de alta relevancia, alineada con los objetivos de desarrollo sostenible y la mitigación del cambio climático.

### 2.1. Objetivo General

Implementar una herramienta de inteligencia artificial que integre un modelo de clasificación de residuos mediante imágenes junto a un chat inteligente, destinada a ser integrada en una aplicación móvil que promueva prácticas sostenibles en Guatemala.

### 2.2. Objetivos Específicos

- Desarrollar un modelo de *deep learning* para la identificación y clasificación precisa de diferentes tipos de residuos, utilizando únicamente imágenes.
- Implementar un sistema de chat inteligente basado en un modelo de lenguaje de aprendizaje profundo (LLM) para facilitar la interacción informativa con los usuarios sobre la gestión de residuos.
- Evaluar la precisión y efectividad del modelo de clasificación de residuos mediante métricas de desempeño específicas para modelos de *deep learning*.
- Evaluar la efectividad del chat inteligente en términos de precisión y tiempos de respuesta al proporcionar información y consejos sobre la gestión de residuos, utilizando métricas de desempeño de *deep learning*.

El desarrollo de una aplicación basada en inteligencia artificial (IA) para la gestión de residuos y la reducción de la huella de carbono es una iniciativa estratégica en el contexto guatemalteco. La adopción de tecnologías emergentes puede impulsar significativamente la eficiencia y la sostenibilidad ambiental. Este proyecto no solo busca mejorar la clasificación y gestión de residuos a través de algoritmos avanzados de IA, sino también promover prácticas de reciclaje más informadas y sostenibles entre la comunidad mediante la integración de un chat inteligente.

Esta combinación de tecnología y educación ambiental directa está diseñada para empoderar a los ciudadanos guatemaltecos, proporcionándoles herramientas y conocimientos para hacer cambios significativos en sus hábitos de gestión de residuos. Al abordar los desafíos descritos y alcanzar los objetivos específicos del proyecto, se contribuye directamente a la mitigación de los efectos del cambio climático y a la protección del medio ambiente en Guatemala [70, 89].

Además, el proyecto tiene como objetivo democratizar el acceso a tecnologías avanzadas en comunidades con recursos limitados, haciendo que la aplicación de IA sea accesible y fácil de usar para todos, independientemente de su nivel de conocimientos técnicos. Esto no solo ayuda a cerrar la brecha digital, sino que también promueve la participación activa de las comunidades guatemaltecas en la protección del medio ambiente, en línea con los objetivos de desarrollo sostenible y la reducción de la huella de carbono [33, 104].

### 4.1. Aspectos Legales y Ambientales en Guatemala

#### 4.1.1. Legislación sobre Manejo de Residuos y Reciclaje en Guatemala

En Guatemala, el marco legal para la gestión de residuos ha sido fundamental para abordar los desafíos ambientales y promover la sostenibilidad. Las principales normativas incluyen leyes que regulan desde la generación hasta la disposición final de los residuos, con un énfasis creciente en el reciclaje y la reducción de desechos.

##### **Ley de Protección y Mejoramiento del Medio Ambiente (Decreto 68-86)**

Esta ley, promulgada en 1986, es un pilar de la legislación ambiental en Guatemala. Su objetivo es proteger y mejorar el medio ambiente a través de la regulación del uso de los recursos naturales y la prevención de la contaminación. El Decreto 68-86 establece la obligación de gestionar adecuadamente los residuos sólidos para minimizar su impacto negativo en la salud pública y el entorno [37].

##### **Reglamento para la Gestión Integral de los Residuos y Desechos Sólidos Comunes (Acuerdo Gubernativo 164-2021)**

Aprobado en 2021, este reglamento es una herramienta clave para la gestión de residuos sólidos en Guatemala. Establece directrices claras para la separación, recolección, transporte y disposición final de los residuos, promoviendo prácticas de reciclaje y la reducción de la cantidad de desechos que terminan en vertederos. El reglamento también contempla sanciones para quienes no cumplan con las normativas, subrayando la importancia de un cumplimiento estricto para proteger el medio ambiente [40, 33].

### Política Nacional de Residuos Sólidos (MARN, 2010)

El Ministerio de Ambiente y Recursos Naturales (MARN) desarrolló esta política como una respuesta a la creciente generación de residuos y su impacto ambiental. La política se centra en la reducción de residuos, la promoción del reciclaje y la valorización de los desechos. También fomenta la participación activa de la ciudadanía y la implementación de programas educativos para concienciar sobre la importancia de la gestión adecuada de residuos [63].

#### 4.1.2. Impacto Ambiental del Manejo de Residuos en Guatemala

El manejo inadecuado de residuos representa uno de los mayores desafíos ambientales en Guatemala. Los vertederos, muchos de los cuales están llegando a su capacidad máxima, son una fuente significativa de contaminación del suelo y del agua. Además, la descomposición de residuos orgánicos en estos sitios genera emisiones de metano, un potente gas de efecto invernadero que contribuye al cambio climático [56].

En las áreas rurales, donde la infraestructura para la gestión de residuos es limitada, la quema de basura es una práctica común. Esta forma de eliminación de residuos no solo contamina el aire, sino que también representa un riesgo para la salud humana, provocando enfermedades respiratorias y otras afecciones. Un informe de la FAO subraya la necesidad de mejorar la infraestructura y las prácticas de manejo de residuos en áreas rurales para mitigar estos impactos negativos [29].

#### 4.1.3. Estrategias Internacionales y su Aplicación en Guatemala

Guatemala ha tomado medidas para alinearse con las mejores prácticas internacionales en la gestión de residuos, inspirándose en enfoques como la economía circular. La Fundación Ellen MacArthur, por ejemplo, promueve un modelo en el que los materiales se mantienen en uso el mayor tiempo posible, reduciendo así la necesidad de nuevos recursos y minimizando los desechos [24]. A pesar de estos avances conceptuales, la implementación efectiva de estas estrategias en Guatemala enfrenta obstáculos significativos, incluidos limitados recursos financieros y una infraestructura inadecuada.

La Estrategia Europea para Plásticos en una Economía Circular, lanzada en 2018, es otro modelo relevante que Guatemala podría emular. Esta estrategia tiene como objetivo reducir los plásticos de un solo uso y fomentar el reciclaje a gran escala, lo cual es crucial en un contexto donde los plásticos representan una gran parte de los residuos no biodegradables [26].

#### 4.1.4. Crisis de Desechos en Guatemala

El informe titulado *Análisis de la generación de desechos sólidos en Guatemala*, publicado por el Instituto de Problemas Nacionales de la Universidad de San Carlos de Guatemala (IPNUSAC) en 2023, proporciona un análisis exhaustivo sobre la creciente crisis de desechos en el país. Este estudio revela que la generación de desechos sólidos ha aumentado considerablemente en las últimas décadas, superando la capacidad de gestión de los municipios guatemaltecos [50].

El estudio destaca varias deficiencias clave en la infraestructura para el manejo de residuos, particularmente en áreas rurales y urbanas marginales. La falta de rellenos sanitarios adecuados ha llevado a un uso generalizado de vertederos a cielo abierto, que no solo contribuyen a la contaminación ambiental, sino que también representan un riesgo significativo para la salud pública. Además, la contaminación de cuerpos de agua debido a la disposición inadecuada de residuos es un problema crítico identificado en el informe [50].

Este informe subraya la necesidad urgente de implementar políticas más efectivas y de invertir en infraestructura adecuada para mitigar los impactos ambientales y sociales asociados con la mala gestión de residuos en Guatemala.

#### **4.1.5. Desafíos Socioeconómicos en la Gestión de Residuos**

En Guatemala, la gestión de residuos está marcada por una fuerte disparidad entre las áreas urbanas y rurales. Mientras que las ciudades más grandes como la Ciudad de Guatemala cuentan con servicios de recolección de basura, las comunidades rurales y las áreas urbanas marginales a menudo carecen de dichos servicios, lo que agrava los problemas de salud pública y contaminación ambiental. Además, un gran porcentaje del reciclaje es llevado a cabo por trabajadores informales, conocidos como "guajeros", quienes operan en vertederos a cielo abierto, enfrentando condiciones laborales peligrosas y sin acceso a protección social [33].

#### **4.1.6. Educación y Sensibilización Ambiental**

La concientización pública es clave para mejorar la gestión de residuos en Guatemala. Aunque existen iniciativas locales de educación ambiental, los esfuerzos a gran escala han sido insuficientes para cambiar los hábitos de la población. Es necesario incorporar programas de educación ambiental desde una edad temprana en las escuelas, así como campañas de sensibilización pública dirigidas a adultos [29].

#### **4.1.7. Impacto del Cambio Climático y la Gestión de Residuos**

El manejo de residuos está estrechamente vinculado al cambio climático, especialmente debido a las emisiones de metano provenientes de los vertederos. Mejorar la gestión de residuos y promover el reciclaje no solo reduce la contaminación local, sino que también contribuye a mitigar el cambio climático global [56].

### **4.2. Reciclaje y Prácticas Sostenibles en la Gestión de Residuos**

El reciclaje y las prácticas sostenibles son componentes esenciales en la gestión moderna de residuos, que buscan minimizar el impacto ambiental y fomentar un uso más eficiente de los recursos. En Guatemala, estas prácticas están ganando impulso, aunque todavía enfrentan desafíos significativos debido a la infraestructura limitada y la falta de educación ambiental.

#### **4.2.1. Tipos de Residuos**

El reciclaje efectivo depende en gran medida de la clasificación adecuada de los residuos. En general, los residuos pueden clasificarse en las siguientes categorías:

**Residuos Sólidos Urbanos (RSU)**

Incluyen los desechos domésticos generados en hogares, comercios y servicios. Los RSU suelen estar compuestos por una mezcla de residuos orgánicos, papel, plástico, vidrio y metales, lo que requiere una separación adecuada para facilitar el reciclaje [22].

**Residuos Orgánicos**

Estos residuos consisten principalmente en restos de alimentos y materiales biodegradables, como restos de poda y jardín. Los residuos orgánicos son ideales para procesos de compostaje, lo que reduce la necesidad de vertederos y genera abono natural [35].

**Residuos Inorgánicos**

Materiales como plásticos, metales, vidrio y papel son reciclables y pueden ser transformados en nuevos productos, reduciendo la extracción de materias primas vírgenes y el consumo energético [22].

**Residuos Peligrosos**

Incluyen baterías, productos químicos, residuos médicos y otros materiales que representan un riesgo para la salud humana y el medio ambiente. Su gestión requiere un manejo especializado para evitar la contaminación y daños a la salud pública [35].

**Residuos Electrónicos (E-Waste)**

Estos son dispositivos electrónicos y eléctricos desechados, como teléfonos móviles, computadoras y electrodomésticos. Los residuos electrónicos contienen materiales valiosos que pueden ser recuperados, pero también sustancias tóxicas que requieren una disposición segura [22].

**4.2.2. Prácticas Sostenibles en el Reciclaje****Economía Circular**

La economía circular es un modelo económico que busca mantener los recursos en uso el mayor tiempo posible, extrayendo su máximo valor antes de recuperarlos y regenerar productos y materiales al final de su vida útil. En Guatemala, la adopción de una economía circular está en sus primeras etapas, pero representa una oportunidad significativa para el desarrollo sostenible [9].

**Reducción en la Fuente**

La minimización de residuos en su origen es una de las estrategias más efectivas para reducir la carga sobre los sistemas de gestión de residuos. Esto incluye el diseño de productos con una mayor vida útil, el uso de materiales reciclables, y la implementación de políticas que incentiven la reducción de desechos, como los impuestos a productos de un solo uso [22].



### **Reciclaje y Compostaje**

La separación adecuada de residuos en orgánicos e inorgánicos es esencial para el éxito del reciclaje y el compostaje. En Guatemala, iniciativas como la separación en la fuente y la promoción de compostaje doméstico están empezando a ganar tracción, aunque enfrentan desafíos como la falta de infraestructura y la educación insuficiente del público [35].

### **Educación y Concientización**

La educación ambiental es fundamental para fomentar prácticas de reciclaje en la sociedad. Programas educativos en escuelas, junto con campañas públicas de sensibilización, son esenciales para cambiar los hábitos y aumentar la participación en el reciclaje. Las campañas deben enfocarse en la importancia del reciclaje, cómo realizarlo correctamente, y los beneficios para el medio ambiente y la economía local [78].

#### **4.2.3. Impacto del Reciclaje en la Mitigación del Cambio Climático**

El reciclaje y la gestión sostenible de residuos tienen un impacto directo en la mitigación del cambio climático. Al reducir la cantidad de residuos que terminan en vertederos, se disminuye la emisión de gases de efecto invernadero, especialmente metano, que es liberado durante la descomposición de residuos orgánicos en condiciones anaeróbicas. Además, el reciclaje de materiales como el aluminio, papel y plástico reduce la necesidad de extraer y procesar nuevas materias primas, lo que a su vez disminuye las emisiones de CO<sub>2</sub> asociadas con estas actividades industriales [22].

#### **4.2.4. Efectos Nocivos de los Residuos y Gases de Efecto Invernadero**

Los residuos mal gestionados, junto con las emisiones de gases de efecto invernadero, tienen efectos nocivos tanto para el medio ambiente como para la salud humana. La acumulación de residuos en vertederos a cielo abierto, por ejemplo, puede liberar sustancias tóxicas en el suelo y el agua, afectando la biodiversidad y la calidad de los recursos hídricos. Un estudio sobre las emisiones de gases de efecto invernadero en Guatemala resalta que la descomposición de residuos en vertederos es una fuente significativa de metano, un potente gas de efecto invernadero [86].

Desde el punto de vista de la salud humana, la exposición a residuos peligrosos y a la contaminación del aire causada por la quema de basura puede provocar enfermedades respiratorias, problemas cardiovasculares y otras afecciones crónicas. Un análisis realizado por la OMS muestra la correlación entre la exposición a estos contaminantes y el aumento de enfermedades en comunidades cercanas a vertederos [108]. La reducción de residuos a través del reciclaje, junto con una gestión más eficiente de los residuos peligrosos, es esencial para mitigar estos riesgos y proteger la salud pública [105].

#### **4.2.5. Políticas y Legislación Ambiental Relacionadas con el Reciclaje**

En Guatemala, la legislación ambiental relacionada con el reciclaje ha avanzado con el objetivo de mejorar la gestión de residuos y promover prácticas sostenibles. La Ley de Protección y Mejoramiento del Medio Ambiente (Decreto 68-86) establece un marco general para la gestión ambiental, incluyendo la responsabilidad de manejar los residuos de manera que se minimicen sus impactos negativos. A nivel local, el Acuerdo Gubernativo 164-2021 es clave para la regulación de la gestión integral de residuos sólidos, destacando la necesidad de separación en la fuente y el fomento del reciclaje [35].

Sin embargo, un análisis más amplio de las políticas públicas revela que, aunque existen leyes y reglamentos, la implementación efectiva sigue siendo un desafío debido a la falta de recursos, infraestructura y coordinación entre las instituciones involucradas. Estudios en América Latina han señalado la necesidad de políticas más inclusivas que integren a los recicladores informales en el sistema formal de gestión de residuos, lo cual también es relevante para Guatemala [77].

#### 4.2.6. Iniciativas y Proyectos de Reciclaje en Guatemala

Existen diversas iniciativas en Guatemala que buscan promover el reciclaje y la gestión sostenible de residuos. Proyectos como Recicla por tu Futuroz las campañas de recolección organizadas por ONGs y municipalidades son ejemplos de esfuerzos exitosos en áreas urbanas. Además, la iniciativa "Escuelas" ha logrado integrar la educación ambiental con la práctica del reciclaje en varias comunidades escolares [41].

Un ejemplo destacado es el proyecto de reciclaje en comunidades rurales indígenas, el cual tiene como objetivo generar un impacto ambiental positivo al tiempo que fomenta el desarrollo económico local. Este proyecto es parte de un esfuerzo más amplio para crear una cultura ecológica en áreas que tradicionalmente han tenido poco acceso a la infraestructura de gestión de residuos [92].

#### 4.2.7. Desafíos Logísticos y de Infraestructura en el Reciclaje

El reciclaje en Guatemala enfrenta varios desafíos logísticos y de infraestructura. La falta de centros de acopio y plantas de reciclaje, especialmente en áreas rurales, limita la efectividad de los programas de reciclaje. La recolección selectiva es otra área crítica, ya que muchos municipios carecen de sistemas adecuados para separar los residuos en la fuente, lo que complica el proceso de reciclaje [35].

Además, la dependencia de los recicladores informales, quienes operan en condiciones precarias, señala la necesidad de formalizar este sector y mejorar las condiciones de trabajo. Un estudio de la ITU resalta que para mejorar la gestión de residuos electrónicos, se requiere una infraestructura más robusta y un mayor compromiso gubernamental [51].

#### 4.2.8. Impacto Socioeconómico del Reciclaje

El reciclaje en Guatemala tiene un impacto significativo en la economía local, especialmente a través de la generación de empleo. Los recicladores informales, conocidos como "guajeros", desempeñan un papel crucial en la recolección y clasificación de materiales reciclables. Sin embargo, trabajan en condiciones insalubres y sin acceso a protección social, lo que subraya la necesidad de políticas que formalicen y apoyen este sector [36].

Además, el reciclaje contribuye a reducir los costos municipales de gestión de residuos, al disminuir la cantidad de desechos que se envían a los vertederos. Esto libera recursos que pueden ser invertidos en otras áreas críticas de la infraestructura pública [35].

#### 4.2.9. Impacto Ambiental y en la Salud Humana

La gestión inadecuada de residuos tiene graves repercusiones tanto ambientales como para la salud pública en Guatemala. Los vertederos a cielo abierto, donde gran parte de los residuos del país son depositados, contaminan el suelo, el agua y el aire. Además, la quema de residuos, una práctica común en áreas sin recolección adecuada, libera sustancias tóxicas que contribuyen a la

contaminación del aire, aumentando la incidencia de enfermedades respiratorias y otras afecciones crónicas [108, 105].

El reciclaje, al reducir la cantidad de residuos que terminan en estos vertederos, mitiga estos impactos negativos, contribuyendo a la protección del medio ambiente y la salud humana [51].

#### 4.2.10. Residuos Electrónicos y su Gestión

Los residuos electrónicos (e-waste) son un problema creciente en Guatemala debido al aumento en el consumo de dispositivos electrónicos. Estos residuos contienen materiales valiosos que pueden ser reciclados, pero también sustancias peligrosas como plomo y mercurio. La falta de infraestructura adecuada para gestionar e-waste de manera segura resulta en que gran parte de estos residuos termine en vertederos comunes, aumentando los riesgos ambientales y de salud [36].

Para abordar este problema, es crucial implementar programas de reciclaje específicos para e-waste y fortalecer las políticas de gestión de residuos electrónicos, siguiendo modelos exitosos de otros países latinoamericanos [51].

#### 4.2.11. Cambio Climático y la Gestión de Residuos

La relación entre la gestión de residuos y el cambio climático es estrecha. Los vertederos son una fuente significativa de emisiones de metano, un gas de efecto invernadero que tiene un impacto considerable en el calentamiento global. Reducir la cantidad de residuos enviados a los vertederos mediante el reciclaje es una estrategia clave para mitigar el cambio climático. Además, el reciclaje reduce la necesidad de extraer y procesar nuevas materias primas, lo que también disminuye las emisiones de CO<sub>2</sub> asociadas [86].

### 4.3. Tecnología: Machine Learning, Inteligencia Artificial y LLMs

La Inteligencia Artificial (IA) y el Machine Learning (ML) han transformado diversos sectores, incluyendo la gestión de residuos. Estas tecnologías permiten el análisis de grandes volúmenes de datos, optimización de procesos y mejora en la toma de decisiones. En la gestión de residuos, IA y ML están siendo utilizados para mejorar la eficiencia en la recolección, clasificación y reciclaje, así como para desarrollar modelos predictivos que ayuden a anticipar la generación de residuos y a optimizar las rutas de recolección.

#### 4.3.1. Conceptos Clave de Machine Learning e Inteligencia Artificial

**Inteligencia Artificial (IA)** se refiere a la capacidad de las máquinas para realizar tareas que requieren inteligencia humana, como el reconocimiento de voz, la visión por computadora y la toma de decisiones. La IA engloba varias subdisciplinas, incluyendo el Machine Learning, que es una de las más relevantes para la gestión de residuos.

**Machine Learning (ML)** es una rama de la IA que se enfoca en la creación de sistemas que pueden aprender y mejorar automáticamente a partir de la experiencia sin ser explícitamente programados. El ML utiliza algoritmos y modelos estadísticos para identificar patrones en los datos, lo que permite a las máquinas hacer predicciones o tomar decisiones basadas en datos históricos [31].

**Redes Neuronales Artificiales (ANN)** son un conjunto de algoritmos inspirados en la estructura y función del cerebro humano. Las ANNs se utilizan ampliamente en aplicaciones de ML para tareas como la clasificación de imágenes y la predicción de series temporales.

#### 4.3.2. Modelos de Lenguaje de Gran Escala (LLMs)

Los Modelos de Lenguaje de Gran Escala (LLMs) son un tipo de IA que ha ganado gran popularidad debido a su capacidad para procesar y generar lenguaje natural de manera coherente y contextual. Estos modelos, como GPT-3 y GPT-4, están entrenados en vastos conjuntos de datos textuales, lo que les permite realizar tareas complejas como la traducción, el resumen de textos, la generación de código, y más. En el contexto de la gestión de residuos y el reciclaje, los LLMs pueden ser utilizados para:

- **Análisis y Resumen de Documentación:** Los LLMs pueden analizar grandes volúmenes de documentos legislativos, informes técnicos y estudios científicos, proporcionando resúmenes y análisis que facilitan la toma de decisiones en políticas de gestión de residuos.
- **Asistencia en la Comunicación:** Pueden ser implementados para mejorar la comunicación con el público sobre prácticas de reciclaje, generando mensajes personalizados o respuestas a preguntas frecuentes sobre gestión de residuos.
- **Apoyo en Investigación y Desarrollo:** Los LLMs pueden ayudar en la investigación científica, facilitando la revisión de literatura y la generación de hipótesis basadas en análisis de datos textuales.

Los LLMs, sin embargo, presentan desafíos, como la necesidad de vastos recursos computacionales y la preocupación por el sesgo inherente en los datos de entrenamiento. Es crucial que su implementación en el campo de la gestión de residuos sea acompañada de una supervisión ética y técnica adecuada [7].

#### 4.3.3. Aplicaciones de Machine Learning e Inteligencia Artificial en la Gestión de Residuos

El uso de ML e IA en la gestión de residuos ha mostrado un gran potencial para mejorar la eficiencia y reducir costos. Algunas aplicaciones clave incluyen:

##### **Clasificación de Residuos**

La clasificación automática de residuos utilizando visión por computadora es una de las aplicaciones más prometedoras. Mediante el uso de redes neuronales convolucionales (CNNs), las máquinas pueden identificar y clasificar diferentes tipos de residuos, como plásticos, metales y orgánicos, en tiempo real. Esto mejora la precisión del reciclaje y reduce la necesidad de intervención manual [43].

##### **Optimización de Rutas de Recolección**

Los modelos predictivos basados en ML pueden analizar patrones históricos de generación de residuos y utilizar esa información para optimizar las rutas de recolección. Esto no solo reduce el tiempo y los costos operativos, sino que también disminuye las emisiones de carbono asociadas con los camiones de recolección [102].

### Predicción de la Generación de Residuos

El ML también se utiliza para predecir la cantidad y el tipo de residuos que se generarán en determinadas áreas o períodos de tiempo. Esto permite a las municipalidades y empresas de gestión de residuos planificar con mayor precisión y asegurar que los recursos estén disponibles cuando se necesiten [58].

#### 4.3.4. Estudios y Ejemplos Relevantes

Diversos estudios han demostrado la efectividad del uso de IA y ML en la gestión de residuos:

- Un estudio realizado por Truong et al. (2022) desarrolló un sistema de clasificación de residuos basado en redes neuronales profundas, que alcanzó una precisión del 95 % en la identificación de materiales reciclables [102].
- Marin et al. (2021) implementaron un modelo predictivo para anticipar la generación de residuos en áreas urbanas, lo que permitió optimizar la recolección y reducir costos operativos en un 20 % [58].
- Un proyecto piloto en Europa utilizó IA para mejorar la clasificación de residuos electrónicos, logrando un aumento del 30 % en la recuperación de materiales valiosos [43].

Estos estudios subrayan el papel crucial que ML e IA pueden desempeñar en la creación de sistemas de gestión de residuos más eficientes y sostenibles.

#### 4.3.5. Retos y Futuro del Uso de IA, ML y LLMs en la Gestión de Residuos

Aunque las aplicaciones de IA y ML en la gestión de residuos están avanzando rápidamente, existen desafíos que deben superarse. Estos incluyen la necesidad de grandes volúmenes de datos de alta calidad, la integración de estas tecnologías en sistemas de gestión de residuos existentes y la aceptación por parte de los operadores humanos. Los LLMs también plantean desafíos éticos y técnicos, como la gestión del sesgo y la transparencia de los modelos. A medida que estas tecnologías continúan desarrollándose, se espera que desempeñen un papel cada vez más importante en la promoción de prácticas sostenibles y en la reducción del impacto ambiental de la gestión de residuos [7, 31].

### 4.4. Leyes, Regulaciones y Ética en el Uso de la Inteligencia Artificial

El uso de la \*Inteligencia Artificial\* (IA) está en auge a nivel global, y con ello, ha surgido la necesidad de establecer marcos legales y regulatorios que aseguren su uso ético y seguro. En Guatemala, la legislación específica sobre IA aún está en desarrollo, pero existen marcos regulatorios generales que pueden aplicarse al uso de esta tecnología.

#### 4.4.1. Leyes y Regulaciones sobre el Uso de la Inteligencia Artificial

##### Legislación en Guatemala

En Guatemala, la Ley de Protección de Datos Personales (Decreto 25-2012) es una de las normativas más relevantes que puede impactar el uso de IA, especialmente en aplicaciones que involucran el procesamiento de datos personales. Esta ley establece las bases para la recolección, almacenamiento y uso de datos, exigiendo el consentimiento de los individuos y la implementación de medidas de seguridad para proteger su privacidad [38].

Aunque no existe una regulación específica para IA en Guatemala, se han dado pasos importantes en la región para fomentar el desarrollo ético y responsable de la IA. Un ejemplo reciente es la participación de Guatemala en el Primer Foro de Ética de la Inteligencia Artificial, donde se discutieron las necesidades y desafíos regionales para implementar una IA que respete los derechos humanos y promueva el bienestar social [3].

##### Regulaciones Internacionales

A nivel internacional, la Unión Europea ha sido pionera en el establecimiento de un marco regulatorio para la IA con la propuesta del Reglamento de IA ("Artificial Intelligence Act"). Este reglamento clasifica los sistemas de IA en función de su riesgo, estableciendo obligaciones legales para desarrolladores y usuarios de IA en función del nivel de riesgo que sus aplicaciones representan [25].

En los Estados Unidos, aunque no existe una regulación federal específica sobre IA, se han emitido directrices a nivel estatal y se están desarrollando normativas sectoriales para regular aplicaciones específicas de IA, como el uso de algoritmos en la toma de decisiones de crédito y empleo [34].

#### 4.4.2. Ética en el Uso de la Inteligencia Artificial

La ética en el uso de la IA es un tema crucial, dado que esta tecnología puede tener un impacto significativo en la sociedad. Los principales desafíos éticos incluyen la transparencia de los algoritmos, el sesgo en la toma de decisiones, la privacidad, y la responsabilidad ante los errores o decisiones automatizadas que perjudican a los individuos.

##### Transparencia y Explicabilidad

Uno de los principios fundamentales en la ética de la IA es la transparencia. Los sistemas de IA deben ser capaces de explicar cómo y por qué se toman ciertas decisiones. Esto es particularmente importante en aplicaciones sensibles como la gestión de residuos, donde las decisiones automatizadas pueden afectar a comunidades enteras. La falta de transparencia puede generar desconfianza en la tecnología y dificultar su adopción [31].

##### Sesgo y Discriminación

El sesgo en los algoritmos de IA es un problema ético ampliamente discutido. Si los datos utilizados para entrenar los modelos de IA contienen sesgos, estos pueden perpetuar y amplificar la discriminación. Por ejemplo, en la gestión de residuos, un algoritmo sesgado podría priorizar áreas

urbanas ricas para la recolección en detrimento de comunidades rurales pobres. Es crucial que los desarrolladores de IA trabajen para identificar y mitigar estos sesgos [81].

### **Privacidad y Seguridad de los Datos**

La privacidad es otro tema ético clave. Los sistemas de IA que recopilan y procesan grandes cantidades de datos personales deben hacerlo de manera que se proteja la privacidad de los individuos. Esto es particularmente relevante en aplicaciones de IA que monitorean o registran datos sensibles, como la recolección y gestión de residuos [38].

#### **4.4.3. Responsabilidad y Gobernanza de la IA**

El concepto de responsabilidad en la IA se refiere a quién o qué es responsable cuando un sistema de IA falla o toma una decisión que resulta en daños. En Guatemala, la falta de regulación específica para IA plantea preguntas sobre quién es responsable en estos casos: el desarrollador del sistema, el usuario o la entidad que implementa la IA. A nivel global, se están desarrollando marcos de gobernanza para abordar estos desafíos, asegurando que la IA se utilice de manera ética y responsable [25].

## **4.5. Impacto Socioeconómico de la IA en la Gestión de Residuos**

La implementación de la \*Inteligencia Artificial\* (IA) en la gestión de residuos tiene un impacto significativo tanto en la economía como en la estructura social. Desde la creación de nuevos empleos hasta la optimización de recursos, la IA está transformando la forma en que las comunidades gestionan sus desechos.

### **4.5.1. Creación de Empleo y Economía Circular**

La introducción de IA en la gestión de residuos está creando nuevas oportunidades de empleo en áreas como el desarrollo de algoritmos, el análisis de datos y la operación de sistemas automatizados. A medida que las empresas adoptan tecnologías avanzadas, surge la necesidad de trabajadores con habilidades en programación, ciencia de datos y mantenimiento de hardware. Además, la IA está impulsando la economía circular al optimizar la reutilización de materiales y reducir el desperdicio [36].

### **4.5.2. Desigualdades Digitales**

A pesar de los beneficios económicos, la implementación de IA también puede exacerbar las desigualdades digitales. Las comunidades con acceso limitado a la tecnología pueden quedar rezagadas, lo que genera disparidades en la gestión de residuos y en el acceso a empleos tecnológicos. Es crucial que las políticas públicas aborden estas desigualdades y promuevan un acceso equitativo a las oportunidades creadas por la IA [81].

## 4.6. Futuro y Tendencias en IA para la Gestión de Residuos

El futuro de la IA en la gestión de residuos promete avances significativos, impulsados por tecnologías emergentes y un enfoque creciente en la sostenibilidad.

### 4.6.1. Desarrollos Emergentes

Entre las tendencias más notables se encuentran el uso de *\*blockchain\** para la trazabilidad de residuos y la integración de *\*Internet de las Cosas\** (IoT) con IA para la monitorización en tiempo real de la generación y gestión de residuos. Estas tecnologías permiten una mayor transparencia y eficiencia en todo el proceso de gestión, desde la recolección hasta el reciclaje [102].

### 4.6.2. Automatización Completa de Procesos

A largo plazo, se espera que la automatización completa de los procesos de reciclaje sea una realidad. Con la combinación de IA, robots autónomos y sensores avanzados, será posible manejar grandes volúmenes de residuos con mínima intervención humana, aumentando así la eficiencia y reduciendo los errores [7].

## 4.7. Desafíos Técnicos en la Implementación de IA y ML

A pesar de los avances, la implementación de IA y *Machine Learning* (ML) en la gestión de residuos enfrenta varios desafíos técnicos.

### 4.7.1. Limitaciones Tecnológicas

El desarrollo de modelos de IA requiere grandes volúmenes de datos de alta calidad, lo cual puede ser un desafío en sectores como la gestión de residuos, donde la recolección de datos consistentemente etiquetados es complicada. Además, la complejidad de los modelos de *\*Deep Learning\** dificulta su interpretación, lo que puede limitar su adopción en sectores donde la transparencia es crucial [31].

### 4.7.2. Seguridad y Resiliencia de los Sistemas de IA

La seguridad de los sistemas de IA es una preocupación crítica, especialmente en aplicaciones que involucran infraestructura esencial como la gestión de residuos. La resiliencia de estos sistemas frente a ataques cibernéticos y fallos técnicos es esencial para garantizar su fiabilidad y sostenibilidad a largo plazo [81].

## 4.8. Evaluación de Impacto y Sostenibilidad

La integración de IA en la gestión de residuos también debe ser evaluada desde una perspectiva de sostenibilidad, considerando tanto los impactos ambientales como los económicos.



#### 4.8.1. Evaluación de Ciclo de Vida (LCA) Aplicada a Sistemas de IA

La Evaluación de Ciclo de Vida (LCA) es una herramienta clave para medir el impacto ambiental de los sistemas de IA, considerando todo el ciclo de vida del sistema, desde la extracción de recursos hasta la disposición final. Implementar LCA en la gestión de residuos permite identificar áreas donde se pueden reducir los impactos negativos y mejorar la sostenibilidad [105].

#### 4.8.2. Sostenibilidad y Eficiencia Energética

El desarrollo y ejecución de modelos de IA pueden tener un impacto considerable en términos de consumo energético. Es crucial adoptar estrategias que minimicen este consumo, como la optimización de algoritmos y el uso de infraestructura más eficiente energéticamente [36].

### 4.9. Interdisciplinariedad en el Desarrollo de IA

El desarrollo y aplicación de IA en la gestión de residuos no pueden realizarse de manera aislada; requieren un enfoque interdisciplinario que combine conocimientos técnicos con una comprensión profunda de los contextos sociales, ambientales y éticos.

#### 4.9.1. Colaboración Multidisciplinaria

La colaboración entre ingenieros, científicos de datos, expertos en medio ambiente y sociólogos es fundamental para el desarrollo de soluciones integrales que sean técnicamente sólidas y socialmente responsables. Esta colaboración garantiza que los sistemas de IA diseñados para la gestión de residuos no solo sean eficientes, sino que también respeten los principios de equidad y sostenibilidad [7].

#### 4.9.2. Educación y Formación en IA y Sostenibilidad

Formar a las futuras generaciones en la intersección de la IA y la sostenibilidad es crucial para asegurar un desarrollo responsable. Esto implica la creación de programas educativos que integren la enseñanza de técnicas avanzadas de IA con una comprensión profunda de los desafíos ambientales y sociales [31].

### 4.10. Aspectos Técnicos de *Machine Learning*

La implementación de *Machine Learning* (ML) e *Inteligencia Artificial* (IA) en la gestión de residuos requiere una comprensión profunda de los modelos, algoritmos, hiperparámetros y métricas de desempeño. En esta sección, se abordan los aspectos técnicos esenciales que facilitan la aplicación exitosa de estas tecnologías en el campo de la sostenibilidad.

#### 4.10.1. Modelos de *Machine Learning* Aplicables

Existen varios modelos de *Machine Learning* que son particularmente útiles en la gestión de residuos, cada uno con sus características y ventajas específicas:

- **Árboles de Decisión:** Son modelos de clasificación que dividen los datos en ramas basadas en decisiones binarias. Son fáciles de interpretar y útiles para la clasificación de residuos.
- **Redes Neuronales Convolucionales (CNNs):** Utilizadas para la clasificación de imágenes, como en la identificación de tipos de residuos en una cinta transportadora. Las CNNs son eficaces para procesar datos visuales complejos.
- **Máquinas de Soporte Vectorial (SVMs):** Útiles para la clasificación de residuos cuando las clases son separables con un margen claro. Las SVMs maximizan este margen, logrando alta precisión en la clasificación.
- **Redes Neuronales Recurrentes (RNNs):** Se aplican en series temporales y predicción de la generación de residuos, ya que pueden manejar datos secuenciales y temporales.

#### 4.10.2. Hiperparámetros en *Machine Learning*

Los hiperparámetros son parámetros que definen la estructura del modelo y cómo se entrena. A diferencia de los parámetros aprendidos, los hiperparámetros son fijados antes del proceso de entrenamiento. A continuación se describen varios hiperparámetros clave:

- **Tasa de Aprendizaje ( $\eta$ ):** Controla el tamaño de los pasos que da el algoritmo durante la optimización. Si es demasiado alta, el modelo puede oscilar; si es demasiado baja, el modelo puede tardar mucho en converger.
- **Número de Épocas:** Define cuántas veces el modelo verá el conjunto completo de datos de entrenamiento. Demasiadas épocas pueden llevar al sobreajuste (*overfitting*), mientras que pocas pueden resultar en un subajuste (*underfitting*).
- **Número de Neuronas y Capas en Redes Neuronales:** Afecta la capacidad del modelo para capturar relaciones complejas. Un número excesivo de neuronas puede llevar al sobreajuste.
- **Tamaño del Mini-Batch:** Controla el número de muestras que el modelo procesa antes de actualizar los parámetros. Un tamaño de mini-batch pequeño puede hacer que el entrenamiento sea ruidoso pero rápido, mientras que un tamaño grande puede estabilizar las actualizaciones pero requiere más memoria.
- **Regularización (L2, L1):** Ayuda a prevenir el sobreajuste aplicando una penalización a los parámetros del modelo. La regularización L2 es común en redes neuronales, mientras que la regularización L1 es útil para generar modelos esparsos.
- **Tasa de Abandono (*Dropout*):** En redes neuronales, se refiere a la probabilidad de que una neurona sea "apagada" durante una iteración del entrenamiento para prevenir el sobreajuste.
- **Momento (*Momentum*):** Es un hiperparámetro que ayuda a acelerar el gradiente descendente hacia la dirección más relevante y a evitar oscilaciones en los mínimos locales.

#### 4.10.3. Ecuaciones Matemáticas Clave

Las siguientes ecuaciones son fundamentales en el desarrollo y entrenamiento de modelos de ML:

### Función de Costo (Costo o Pérdida)

La función de costo mide la diferencia entre las predicciones del modelo y los valores reales. En el caso de la regresión, se utiliza comúnmente el *Error Cuadrático Medio* (MSE):

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Donde:

- $m$  es el número de ejemplos de entrenamiento.
- $h_{\theta}(x^{(i)})$  es la predicción del modelo.
- $y^{(i)}$  es el valor real.

### *Gradiente Descendente*

El algoritmo de *Gradiente Descendente* se utiliza para minimizar la función de costo ajustando los parámetros del modelo:

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Donde:

- $\theta_j$  son los parámetros del modelo.
- $\alpha$  es la tasa de aprendizaje.
- $\frac{\partial}{\partial \theta_j} J(\theta)$  es el gradiente de la función de costo.

#### 4.10.4. Métricas de Desempeño en *Machine Learning*

Las métricas de desempeño son cruciales para evaluar la efectividad de los modelos de ML. Algunas de las métricas más comunes incluyen:

- **Precisión:** Proporción de predicciones correctas sobre el total de predicciones realizadas.
- **Exactitud (*Accuracy*):** Medida global de desempeño para un modelo de clasificación.
- **ROC-AUC:** Área bajo la curva ROC, que mide la capacidad de un modelo para distinguir entre clases.
- **F1 Score:** Medida ponderada que combina precisión y *recall*, útil en escenarios con clases desbalanceadas.

#### 4.10.5. Visualización de Modelos y Resultados

Las visualizaciones son herramientas poderosas para interpretar y comunicar los resultados de los modelos de ML. Las matrices de confusión, gráficas ROC y diagramas de árbol de decisión son ejemplos de visualizaciones que ayudan a entender cómo funciona un modelo.

n=192		Predicted: 0	Predicted: 1	
Actual: 0		TN = 118	FP = 12	130
Actual: 1		FN = 47	TP = 15	62
		165	27	

Figura 4.1: Matriz de confusión que ilustra la precisión del clasificador a través de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos.

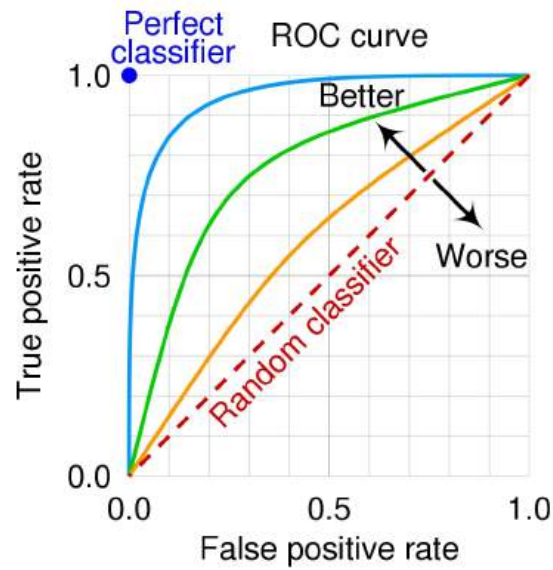


Figura 4.2: Curva ROC que muestra el desempeño de un clasificador binario.



Figura 4.3: Diagrama de árbol de decisión que representa las decisiones y sus posibles consecuencias.

#### 4.10.6. Regularización y Penalización en Modelos

La regularización es crucial para prevenir el sobreajuste en modelos complejos, especialmente cuando se trabaja con datos de alta dimensionalidad. A continuación, se describen las dos formas más comunes de regularización:

##### Regularización L1 y L2

La regularización L1 agrega una penalización proporcional a la suma de los valores absolutos de los coeficientes, mientras que la regularización L2 penaliza la suma de los cuadrados de los coeficientes [19].

■ **L1 (Lasso):**

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \left( h_{\theta}(x^{(i)}) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^n |\theta_j|$$

■ **L2 (Ridge):**

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m \left( h_{\theta}(x^{(i)}) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^n \theta_j^2$$

Donde  $\lambda$  es el factor de regularización que controla la fuerza de la penalización.

#### 4.10.7. Algoritmo de Optimización Adam

El optimizador *Adam* combina las ventajas del método de *Momentum* y *Adaptive Gradient Algorithm (AdaGrad)*. Es ampliamente utilizado debido a su capacidad para ajustarse dinámicamente a diferentes características de los datos [106].

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\theta} J(\theta) \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_{\theta} J(\theta))^2 \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \\ \theta_{t+1} &= \theta_t - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \end{aligned}$$

Donde:

- $\alpha$  es la tasa de aprendizaje.
- $\beta_1$  y  $\beta_2$  son los coeficientes de decaimiento exponencial para las estimaciones de momentos.

#### 4.10.8. Regularización mediante *Dropout*

El *Dropout* es una técnica de regularización que consiste en apagar (ignorar) de manera aleatoria ciertas neuronas durante el entrenamiento. Esto previene que las neuronas se vuelvan demasiado dependientes entre sí, ayudando a evitar el sobreajuste.

### 4.10.9. Funciones de Activación

Las funciones de activación son componentes críticos de las redes neuronales, ya que introducen no linealidad en el modelo, permitiéndole aprender y modelar relaciones complejas.

- **Función Sigmoides:**

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Se utiliza principalmente en la última capa de una red neuronal para problemas de clasificación binaria [20].

- **Función ReLU (*Rectified Linear Unit*):**

$$f(x) = \max(0, x)$$

ReLU es la función de activación más utilizada en redes profundas debido a su simplicidad y efectividad al mitigar el problema del gradiente desvanecido [61].

### 4.10.10. Evaluación del Sesgo y la Varianza

El análisis de sesgo y varianza es fundamental para entender la capacidad de generalización de un modelo de *Machine Learning*.

- **Sesgo:** Se refiere a la capacidad de un modelo para capturar las relaciones subyacentes entre las variables de entrada y salida.
- **Varianza:** Refleja la sensibilidad del modelo a los cambios en los datos de entrenamiento. Un modelo con alta varianza puede sobreajustarse a los datos de entrenamiento [15].

### 4.10.11. Métodos de Validación y Evaluación

Para evaluar el rendimiento de un modelo de *Machine Learning*, es crucial usar métodos como la validación cruzada.

#### Validación Cruzada K-Fold

En la validación cruzada K-Fold, el conjunto de datos se divide en K subconjuntos. El modelo se entrena K veces, cada vez utilizando K-1 subconjuntos para el entrenamiento y el restante para la prueba [106].

La métrica final es el promedio de los resultados obtenidos en cada iteración, lo que proporciona una estimación robusta del desempeño del modelo.

## 4.11. Aspectos Técnicos de la *Inteligencia Artificial* y *Deep Learning*

### 4.11.1. Redes Neuronales Profundas (*Deep Neural Networks*)

Las redes neuronales profundas (DNNs) son el núcleo de muchas aplicaciones modernas de *Deep Learning*. Están compuestas por múltiples capas de neuronas, cada una de las cuales realiza trans-

formaciones lineales y no lineales sobre los datos de entrada. Estas redes son capaces de modelar relaciones complejas en datos de alta dimensionalidad, lo que las hace ideales para tareas como la clasificación de residuos.

### Modelo Matemático de una Neurona

El funcionamiento básico de una neurona en una DNN se describe mediante la siguiente ecuación:

$$z = \sum_{i=1}^n w_i x_i + b$$

Donde:

- $z$  es la entrada ponderada a la función de activación.
- $w_i$  son los pesos de las conexiones sinápticas.
- $x_i$  son las entradas.
- $b$  es el sesgo (bias).

La salida de la neurona se obtiene aplicando una función de activación  $\sigma(z)$ :

$$a = \sigma(z)$$

#### 4.11.2. Redes Neuronales Convolucionales (*Convolutional Neural Networks, CNNs*)

Las redes neuronales convolucionales (CNNs) son un tipo de red neuronal profunda optimizada para el procesamiento de datos estructurados espacialmente, como imágenes. En la gestión de residuos, las CNNs se utilizan ampliamente para la clasificación automática de residuos a partir de imágenes.

#### Operación Convolutiva

La operación convolutiva es el núcleo de una CNN. Matemáticamente, se define como:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n) \cdot K(m, n)$$

Donde:

- $I$  es la matriz de la imagen de entrada.
- $K$  es el filtro (kernel) aplicado.
- $S(i, j)$  es la salida de la convolución en la posición  $(i, j)$ .

### 4.11.3. Redes Neuronales Recurrentes (*Recurrent Neural Networks, RNNs*)

Las redes neuronales recurrentes (RNNs) son adecuadas para procesar secuencias de datos, como la predicción de generación de residuos a lo largo del tiempo. En una RNN, la salida de cada paso temporal se retroalimenta como entrada para el siguiente paso, lo que se modela mediante:

$$h_t = \sigma(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

Donde:

- $h_t$  es el estado oculto en el tiempo  $t$ .
- $W_{hh}$  es la matriz de pesos de la recurrencia.
- $W_{xh}$  es la matriz de pesos de la entrada.
- $x_t$  es la entrada en el tiempo  $t$ .
- $b_h$  es el sesgo.

### 4.11.4. Funciones de Activación en *Deep Learning*

Las funciones de activación introducen no linealidad en los modelos de *Deep Learning*, permitiendo que las redes neuronales aprendan relaciones complejas en los datos.

#### Función Sigmoides

La función sigmoide es una de las funciones de activación más antiguas:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Es especialmente útil en la clasificación binaria, ya que convierte cualquier valor en una probabilidad entre 0 y 1.

#### Función ReLU (Rectified Linear Unit)

La función ReLU se define como:

$$f(z) = \max(0, z)$$

Es la función de activación más utilizada en redes neuronales profundas debido a su simplicidad y efectividad, especialmente en la mitigación del problema del gradiente desvanecido.

### 4.11.5. Optimización y Entrenamiento en *Deep Learning*

El proceso de entrenamiento de modelos de *Deep Learning* implica la minimización de una función de pérdida  $J(\theta)$ , que generalmente se optimiza mediante el método de gradiente descendente y sus variantes.



### Gradiente Descendente

El gradiente descendente actualiza los parámetros del modelo siguiendo la dirección opuesta al gradiente de la función de pérdida:

$$\theta := \theta - \alpha \nabla_{\theta} J(\theta)$$

Donde:

- $\alpha$  es la tasa de aprendizaje.
- $\nabla_{\theta} J(\theta)$  es el gradiente de la función de pérdida respecto a los parámetros  $\theta$ .

### Algoritmo de Optimización Adam

El optimizador Adam ajusta los pesos de la red utilizando momentos de primer y segundo orden:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\theta} J(\theta) \\ v_t &= \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_{\theta} J(\theta))^2 \\ \hat{m}_t &= \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \\ \theta_{t+1} &= \theta_t - \frac{\alpha \hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \end{aligned}$$

#### 4.11.6. Regularización en Redes Profundas

La regularización es esencial para evitar el sobreajuste en modelos complejos de *Deep Learning*. Técnicas comunes incluyen:

- **Dropout:** Durante el entrenamiento, algunas neuronas se <sup>a</sup>pagan.<sup>a</sup>leatoriamente para evitar la co-adaptación excesiva. Matemáticamente, se describe mediante:

$$a^{(l)} = \text{Dropout}(h^{(l-1)}, p)$$

Donde  $p$  es la probabilidad de que una neurona se mantenga activa.

- **Regularización L2:** Agrega una penalización a la función de pérdida, proporcional al cuadrado de los pesos del modelo:

$$J(\theta) = J(\theta) + \lambda \sum_{j=1}^n \theta_j^2$$

#### 4.11.7. Transferencia de Aprendizaje (*Transfer Learning*)

La *Transferencia de Aprendizaje* es una técnica fundamental en *deep learning* que permite la reutilización del conocimiento adquirido por un modelo en una tarea previamente aprendida para aplicarlo en una nueva tarea relacionada. Esta técnica se basa en la premisa de que un modelo que ha sido entrenado en un conjunto de datos amplio y diverso puede captar características generales

y representaciones que son útiles en una variedad de aplicaciones, incluso en dominios con datos específicos y limitados. Un ejemplo prominente de esta técnica es el uso de modelos preentrenados en conjuntos de datos extensivos, como ImageNet, el cual contiene millones de imágenes de categorías variadas [21].

En términos prácticos, en lugar de construir un modelo desde cero para cada tarea, el *Transfer Learning* permite aprovechar un modelo preentrenado como base, ajustando algunas de sus capas para adaptarlo a una nueva tarea específica. Este proceso de ajuste, conocido como *fine-tuning*, generalmente implica conservar y congelar las primeras capas de la red (las cuales capturan características generales de bajo nivel) y entrenar o modificar las capas finales para que el modelo se especialice en la nueva tarea [109]. En el ámbito de la clasificación de imágenes, esta técnica es especialmente útil porque permite reducir el tiempo de entrenamiento y mejorar la precisión al aprovechar las representaciones aprendidas por redes previamente entrenadas.

### Aplicación de Transferencia de Aprendizaje en Clasificación de Imágenes

La transferencia de aprendizaje ha demostrado ser particularmente ventajosa en tareas de clasificación de imágenes donde los conjuntos de datos específicos suelen ser limitados o complejos. En la clasificación de residuos, por ejemplo, los objetos pueden variar en términos de color, forma, textura y contexto, lo cual hace que sea difícil capturar todas las variaciones mediante un conjunto de datos pequeño. Al aplicar *Transfer Learning*, se puede reutilizar la capacidad de reconocimiento de patrones aprendida por un modelo en un conjunto de datos grande y aplicarla a un contexto específico, como la clasificación de distintos tipos de residuos.

### Ventajas del Transfer Learning en Tareas de Clasificación

La adopción del *Transfer Learning* ofrece diversas ventajas en la clasificación de imágenes, como las siguientes:

- **Reducción del tiempo de entrenamiento:** Dado que el modelo ya ha aprendido características básicas y generales, el tiempo necesario para adaptarlo a una nueva tarea es significativamente menor que al entrenar un modelo desde cero [109].
- **Mejor adaptabilidad a la tarea específica:** Al realizar *fine-tuning* en las capas finales de un modelo preentrenado, el modelo puede captar los patrones específicos del nuevo conjunto de datos sin perder las características generales previamente aprendidas [69].
- **Mayor precisión con conjuntos de datos limitados:** Debido a que el modelo preentrenado ya cuenta con una representación compleja de características, la cantidad de datos adicionales necesarios para obtener buenos resultados en una nueva tarea es menor. Esto es ideal en situaciones donde el acceso a grandes volúmenes de datos es limitado.

La técnica de *Transfer Learning* se ha convertido en una práctica estándar en *deep learning* para tareas de clasificación, no solo por su eficiencia en recursos, sino también por su capacidad para adaptar modelos a tareas especializadas de manera rápida y efectiva. En el contexto de este proyecto, se emplean modelos preentrenados para optimizar el reconocimiento y clasificación de distintos tipos de residuos, lo cual permite aprovechar las ventajas de precisión y eficiencia computacional inherentes a esta técnica [109, 69].

#### 4.11.8. Modelos de Transfer Learning en Clasificación de Imágenes

En el campo de la visión por computadora, el uso de modelos de *Transfer Learning* se ha vuelto predominante debido a su capacidad para aprovechar características previamente aprendidas y aplicarlas en nuevas tareas, lo que permite reducir el tiempo de entrenamiento y mejorar la precisión en entornos con datos limitados. Este proyecto emplea varios de estos modelos reconocidos en la literatura, cada uno con arquitecturas optimizadas para maximizar el rendimiento en clasificación de imágenes. A continuación, se explican brevemente los modelos Xception, MobileNetV2, InceptionV3, VGG19 y DenseNet201, los cuales son ampliamente utilizados en el ámbito de la transferencia de aprendizaje y han sido ajustados para la clasificación de residuos en este estudio.

- **Xception:** Desarrollado por François Chollet, Xception es un modelo basado en convoluciones separables en profundidad, lo cual permite realizar cálculos eficientes y reducir el número de parámetros sin perder precisión [10]. Su arquitectura modular, inspirada en la arquitectura Inception, ha demostrado ser efectiva para tareas de clasificación de gran escala y fue ajustada en este proyecto para identificar diferentes tipos de residuos.
- **MobileNetV2:** Propuesto por Sandler et al., MobileNetV2 se diseñó con un enfoque en dispositivos móviles y sistemas con recursos limitados. Este modelo utiliza bloques residuales invertidos y capas de embotellamiento lineal, logrando una estructura que es tanto precisa como computacionalmente eficiente [82]. Estas características lo hacen ideal para aplicaciones de clasificación en tiempo real, como las utilizadas en este proyecto.
- **InceptionV3:** Este modelo, desarrollado por Szegedy et al., utiliza bloques de convolución denominados “Inception” que capturan características visuales de múltiples escalas. InceptionV3 ha sido ampliamente adoptado para clasificación de imágenes debido a su capacidad de procesamiento paralelo de patrones de diferentes tamaños, lo cual permite una extracción robusta de características [94]. En este proyecto, InceptionV3 se aplica para capturar patrones visuales complejos en la clasificación de materiales reciclables y no reciclables.
- **VGG19:** Creado por Simonyan y Zisserman, VGG19 es conocido por su arquitectura profunda y simple, compuesta por bloques de convolución seguidos de capas completamente conectadas. Su diseño, aunque más pesado computacionalmente, permite una extracción detallada de características visuales, lo cual es ventajoso en tareas que requieren precisión en detalles finos [87]. VGG19 se ajusta particularmente bien para distinguir texturas y formas en los diferentes tipos de residuos.
- **DenseNet201:** Propuesto por Gao Huang et al., DenseNet201 se caracteriza por su conectividad densa entre capas, en la cual cada capa recibe como entrada la salida de todas las capas anteriores. Esto promueve la reutilización de características y facilita la retropropagación del gradiente, mejorando la precisión del modelo con un menor número de parámetros [45]. En este proyecto, DenseNet201 permite captar patrones recurrentes en las imágenes de residuos, aprovechando la transferencia de aprendizaje para optimizar el desempeño.

Estos modelos representan algunas de las arquitecturas de *Transfer Learning* más avanzadas, optimizadas para la clasificación de imágenes en diferentes dominios. Su aplicación en la clasificación de residuos aprovecha su capacidad para reconocer patrones complejos y diversos, necesarios para la correcta identificación de materiales y su posterior tratamiento en los flujos de reciclaje.

#### MobileNetV2

MobileNetV2 es una arquitectura de red neuronal convolucional diseñada para optimizar el rendimiento en dispositivos móviles y sistemas con limitaciones computacionales. Desarrollado por Mark

Sandler y colaboradores, este modelo introduce innovaciones arquitectónicas significativas que mejoran su eficiencia sin comprometer su precisión en tareas de clasificación de imágenes [82].

**Aspectos Técnicos y Arquitectura de MobileNetV2** MobileNetV2 se basa en dos conceptos principales: los *bloques residuales invertidos* y las *capas de embotellamiento lineal*. Estos componentes permiten reducir la cantidad de parámetros del modelo, manteniendo su capacidad para captar características visuales complejas.

- **Bloques Residuales Invertidos:** En la arquitectura de MobileNetV2, los bloques residuales invertidos son una modificación de los bloques residuales tradicionales. Mientras que los bloques residuales comunes suelen expandir el espacio de características al inicio del bloque y luego lo reducen al final, en MobileNetV2 este proceso se invierte. La arquitectura comienza con una capa de expansión que aumenta las dimensiones de los canales, permitiendo que la convolución *depthwise* capte las características con mayor precisión. Luego, una capa de proyección reduce el número de canales, lo que ayuda a mantener la eficiencia del modelo sin perder información importante.
- **Convoluciones Depthwise Separable:** MobileNetV2 emplea convoluciones *depthwise separable*, donde la convolución se divide en dos operaciones: una convolución *depthwise*, que aplica un filtro separado a cada canal, y una convolución *pointwise*, que combina los canales. Este método reduce considerablemente el número de cálculos en comparación con una convolución estándar, mejorando así la eficiencia sin comprometer la capacidad de aprendizaje del modelo.
- **Capa de Embotellamiento Lineal:** Para minimizar la pérdida de información en los pasos de compresión de características, MobileNetV2 utiliza una capa lineal de embotellamiento al final de cada bloque residual. Esta capa mantiene la representación lineal de las características y evita la distorsión de información causada por funciones de activación como ReLU en espacios de baja dimensionalidad, ya que esta podría eliminar componentes críticos de las características aprendidas.

**Estructura General de la Red** MobileNetV2 utiliza una estructura repetitiva de bloques de convolución *depthwise separable* y bloques residuales invertidos, intercalados con capas de embotellamiento lineal, lo que da como resultado una red eficiente y compacta. Esta arquitectura permite que el modelo sea altamente optimizado para dispositivos móviles o aplicaciones en tiempo real, ya que mantiene un equilibrio entre precisión y costo computacional.

**Aplicación y Rendimiento** La arquitectura de MobileNetV2 ha demostrado un rendimiento sobresaliente en tareas de clasificación y detección de objetos, especialmente en contextos donde el uso de recursos computacionales es limitado. En este proyecto, se emplea MobileNetV2 para la clasificación de residuos debido a su eficiencia y capacidad para extraer características relevantes en imágenes con una estructura de bajo costo computacional, lo que permite reducir el tiempo de inferencia al clasificar diferentes tipos de materiales reciclables y no reciclables [82].

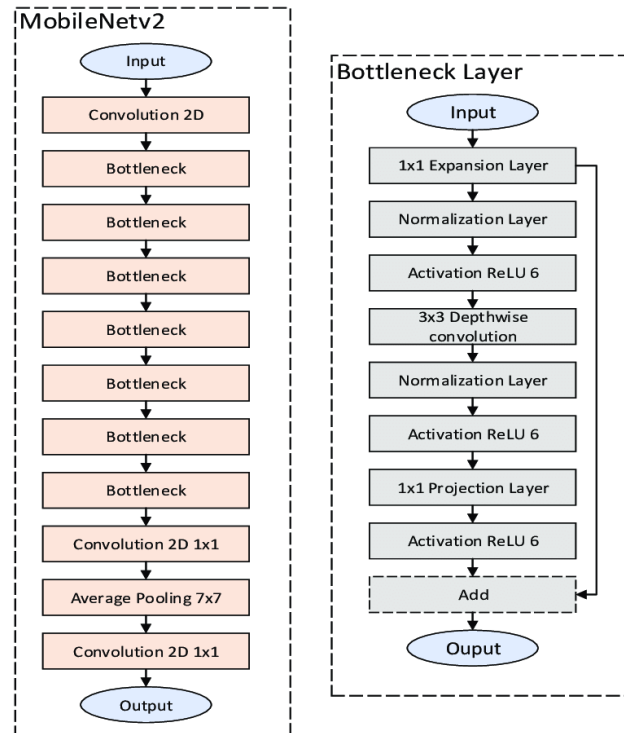


Figura 4.4: Arquitectura del modelo MobileNetV2.

### InceptionV3

InceptionV3 es una arquitectura de red neuronal convolucional desarrollada por Christian Szegedy y su equipo en Google. Forma parte de la familia de arquitecturas Inception, conocidas por su capacidad de extraer características a diferentes escalas de manera eficiente. InceptionV3 fue diseñado para mejorar la precisión en tareas de clasificación de imágenes manteniendo un uso eficiente de los recursos computacionales, lo que lo convierte en una opción ideal para tareas de clasificación de imágenes a gran escala [94].

**Aspectos Técnicos y Arquitectura de InceptionV3** InceptionV3 introduce varias mejoras sobre sus predecesores, incluyendo módulos de convolución de diferentes tamaños y técnicas avanzadas de regularización, que optimizan su desempeño y eficiencia. Los componentes principales de InceptionV3 son los siguientes:

- **Módulos Inception:** La arquitectura de InceptionV3 está organizada en módulos llamados “Inception” que combinan convoluciones de diferentes tamaños (por ejemplo, 1x1, 3x3 y 5x5) dentro de un mismo módulo. Esta estructura permite que la red capture características a distintas escalas, lo que mejora la capacidad del modelo para reconocer patrones variados en las imágenes. Los módulos Inception fusionan múltiples dimensiones de las características extraídas en una sola representación optimizada.
- **Factorización de Convoluciones:** Para reducir la complejidad computacional, InceptionV3 emplea la factorización de convoluciones. En lugar de realizar una convolución de tamaño 5x5, el modelo utiliza una secuencia de dos convoluciones 3x3, que requieren menos operaciones y mejoran la eficiencia del modelo sin comprometer su capacidad de captura de patrones complejos.

- **Convoluciones Asimétricas (1x7 y 7x1):** InceptionV3 también emplea convoluciones asimétricas, como 1x7 y 7x1, para capturar patrones de características en una dirección específica y luego combinarlos. Esto reduce el costo computacional en comparación con las convoluciones cuadradas (por ejemplo, 7x7) y permite un análisis más fino de características dentro de la imagen.
- **Reducción del Tamaño de la Red mediante Pooling:** Los módulos de *pooling* en InceptionV3 se utilizan para reducir las dimensiones de las características intermedias, limitando el tamaño del modelo y mejorando su eficiencia sin perder precisión. Este enfoque permite que el modelo se ajuste a recursos computacionales limitados y sea aplicable a tareas de clasificación que requieren rapidez y precisión.

**Técnicas de Regularización y Optimización** InceptionV3 incorpora varias técnicas de regularización para evitar el sobreajuste y mejorar la generalización del modelo:

- **Dropout:** Durante el entrenamiento, InceptionV3 emplea la técnica de *dropout* en sus capas totalmente conectadas, lo que evita la co-adaptación de neuronas específicas y mejora la generalización del modelo al introducir aleatoriedad en el entrenamiento.
- **Batch Normalization:** La normalización por lotes se utiliza en los módulos Inception para estabilizar el proceso de entrenamiento y permitir que el modelo aprenda más rápido. Esta técnica reduce la sensibilidad a la inicialización de pesos y ayuda a prevenir problemas de estancamiento en redes profundas.

**Aplicación y Rendimiento** Gracias a su capacidad para capturar características a múltiples escalas y su eficiencia computacional, InceptionV3 es ideal para aplicaciones de clasificación de imágenes con un alto grado de variabilidad. En el contexto de este proyecto, InceptionV3 fue empleado para identificar patrones visuales complejos en imágenes de residuos, permitiendo una clasificación precisa y eficiente de los materiales reciclables y no reciclables [94]. Su capacidad para manejar patrones a múltiples escalas le permite ser particularmente útil en la clasificación de objetos de tamaños y formas variadas, comunes en el contexto de residuos sólidos.

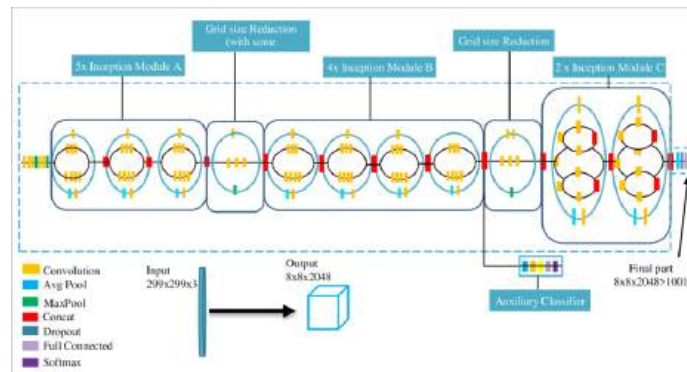


Figura 4.5: Arquitectura del modelo InceptionV3.

## DenseNet201

DenseNet201 es una variante avanzada de las redes densamente conectadas (DenseNets), desarrollada por Gao Huang y colaboradores. Esta arquitectura introduce un esquema de conexión densa que conecta cada capa de la red con todas las capas anteriores, promoviendo un flujo eficiente de

información y gradientes a través de la red [45]. Este diseño es especialmente útil en redes profundas, ya que ayuda a preservar la información y a evitar problemas de gradiente.

**Aspectos Técnicos y Arquitectura de DenseNet201** DenseNet201 se caracteriza por sus conexiones densas y su organización en bloques convolucionales, lo cual optimiza tanto la precisión como la eficiencia en el uso de parámetros. Los elementos principales de su arquitectura son los siguientes:

- **Conexiones Densas:** A diferencia de las redes tradicionales, donde cada capa transmite su salida solo a la capa siguiente, en DenseNet201 cada capa se conecta directamente con todas las capas anteriores. Esto significa que cada capa recibe como entrada las características de todas las capas previas, permitiendo un flujo de información continuo y evitando la pérdida de detalles importantes. Estas conexiones densas mejoran el flujo de gradientes y reducen la probabilidad de problemas de desaparición del gradiente en redes profundas.
- **Bloques Densos y Capas de Transición:** DenseNet201 organiza sus capas en bloques densos, compuestos por varias capas convolucionales, y en capas de transición, que reducen las dimensiones espaciales mediante *pooling*. Las capas de transición también incluyen convoluciones de 1x1, que limitan el número de canales y permiten al modelo mantener una estructura compacta y eficiente.
- **Control de Crecimiento de Características (Growth Rate):** Para controlar la cantidad de información adicional que cada capa introduce, DenseNet201 utiliza un parámetro denominado *growth rate*, el cual determina la cantidad de canales añadidos por cada nueva capa. Esto evita que el número de parámetros crezca de forma excesiva, manteniendo el modelo en un tamaño manejable sin sacrificar la capacidad de aprendizaje.

**Ventajas del Diseño de DenseNet201** DenseNet201 ofrece varias ventajas en comparación con arquitecturas de redes convencionales:

- **Reutilización de Características Aprendidas:** Las conexiones densas permiten que cada capa acceda a las características de todas las capas previas, facilitando la reutilización de información y promoviendo un aprendizaje más eficiente.
- **Optimización del Flujo de Gradientes:** La estructura densa facilita la retropropagación de gradientes hacia capas anteriores, optimizando el aprendizaje y mitigando problemas de pérdida de gradiente en redes profundas.
- **Reducción de Parámetros:** Debido a las conexiones densas y al control de crecimiento, DenseNet201 requiere menos parámetros que modelos equivalentes, reduciendo así el riesgo de sobreajuste y optimizando el uso de recursos computacionales.

DenseNet201 se destaca en tareas de clasificación de imágenes y otros problemas de visión por computadora, donde la preservación de detalles en las características es crucial. Su diseño permite captar tanto características de bajo nivel en las primeras capas como patrones más complejos en las capas posteriores, logrando un balance entre profundidad y eficiencia [45].

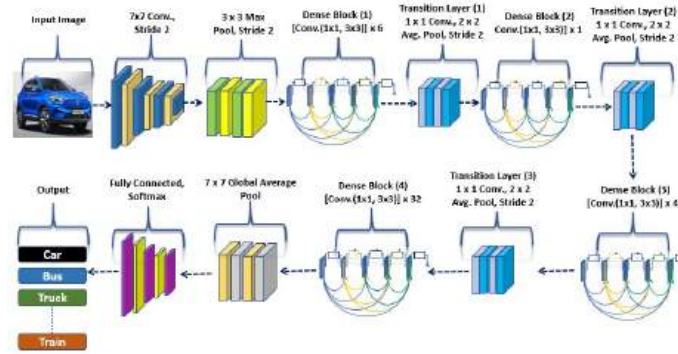


Figura 4.6: Arquitectura del modelo DenseNet201.

## VGG19

VGG19 es una red neuronal convolucional desarrollada por Karen Simonyan y Andrew Zisserman en el Visual Geometry Group de la Universidad de Oxford. Esta arquitectura se caracteriza por su simplicidad y profundidad, utilizando un diseño modular y profundo que ha demostrado ser altamente efectivo en la clasificación de imágenes de gran escala [87]. VGG19 emplea una secuencia de capas convolucionales y de *pooling* organizadas en bloques, lo que facilita el aprendizaje de características visuales detalladas.

**Aspectos Técnicos y Arquitectura de VGG19** VGG19 es una variante de la familia VGG, con 19 capas de aprendizaje organizadas en una estructura simple pero poderosa. Los elementos clave de su diseño incluyen:

- **Capas Convolucionales de 3x3:** Uno de los elementos distintivos de VGG19 es el uso de convoluciones de tamaño reducido (3x3) repetidas varias veces en cada bloque. Este tamaño de filtro permite capturar características finas y detalladas en las imágenes, y al repetirlas en profundidad, VGG19 puede captar patrones de mayor complejidad. A diferencia de los filtros más grandes, el tamaño 3x3 optimiza el uso de recursos computacionales al mismo tiempo que incrementa la profundidad de la red.
- **Bloques de Convolución y Pooling:** La arquitectura de VGG19 se organiza en bloques, donde se agrupan dos o tres capas convolucionales seguidas de una capa de *max pooling*. Este diseño reduce las dimensiones espaciales progresivamente y aumenta el número de canales, permitiendo que el modelo aprenda tanto características locales de bajo nivel como representaciones abstractas de alto nivel.
- **Capas Completamente Conectadas al Final de la Red:** En las etapas finales, VGG19 emplea tres capas totalmente conectadas, seguidas de una función de activación *ReLU* y una capa de *softmax* para la salida de clasificación. Estas capas completamente conectadas consolidan las características aprendidas en una representación final, asignando una probabilidad a cada clase objetivo.

**Ventajas del Diseño de VGG19** VGG19 es conocida por su estructura modular y su capacidad para captar patrones detallados en imágenes, lo que le da ventajas importantes en clasificación de imágenes:



- **Profundidad y Simplicidad:** La estructura de VGG19, aunque más profunda que las redes anteriores, mantiene un diseño sencillo que facilita su implementación y ajuste.
- **Extracción de Características Detalladas:** La repetición de filtros 3x3 permite capturar patrones complejos en imágenes, ideal para aplicaciones donde la precisión en detalles es crítica.
- **Versatilidad en Transferencia de Aprendizaje:** Gracias a su arquitectura profunda y su capacidad para aprender representaciones detalladas, VGG19 es frecuentemente empleada en tareas de *transfer learning*, ya que sus capas convolucionales pueden adaptarse eficazmente a nuevos contextos de clasificación.

VGG19 ha demostrado ser particularmente útil en aplicaciones de clasificación y detección de objetos, proporcionando una estructura robusta para tareas de visión por computadora que requieren una extracción detallada de características [87].

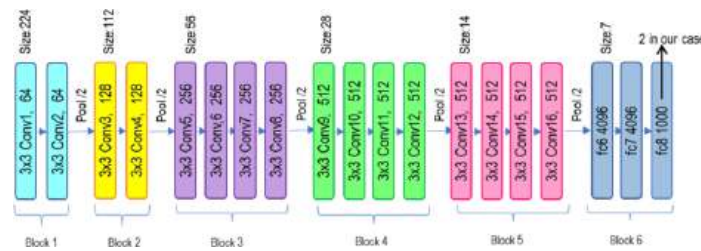


Figura 4.7: Arquitectura del modelo VGG19.

## Xception

Xception, introducido por François Chollet, es una arquitectura de red neuronal convolucional avanzada que mejora los principios de la arquitectura Inception utilizando convoluciones separables en profundidad. Xception es una abreviatura de “Extreme Inception” y se basa en la hipótesis de que la correlación espacial y la correlación entre canales de una imagen pueden modelarse de manera más efectiva mediante convoluciones separadas. Este enfoque permite que el modelo sea más eficiente en el uso de parámetros sin comprometer la capacidad de aprendizaje [10].

**Aspectos Técnicos y Arquitectura de Xception** Xception emplea un diseño basado en convoluciones separables en profundidad y bloques residuales que ayudan a optimizar el flujo de información en redes profundas. Los aspectos clave de su arquitectura incluyen:

- **Convoluciones Separables en Profundidad:** La arquitectura Xception sustituye las convoluciones estándar por convoluciones separables en profundidad en todos sus bloques. Una convolución separable en profundidad consiste en dos pasos: una convolución *depthwise*, que aplica un filtro específico a cada canal de entrada, y una convolución *pointwise* de 1x1, que combina los canales. Esta separación permite reducir significativamente el número de parámetros y operaciones, logrando una mayor eficiencia computacional.
- **Bloques Residuales y Conexiones Directas:** En Xception, los bloques de convolución están organizados en bloques residuales, con conexiones directas que permiten un flujo eficiente de gradientes durante el entrenamiento. Cada bloque residual contiene múltiples capas de convoluciones separables en profundidad, y la conexión directa ayuda a mitigar problemas de desaparición de gradiente en redes profundas.

- **Separación Completa de Correlaciones Espaciales y de Canal:** A diferencia de modelos previos como Inception, donde las convoluciones combinan correlaciones espaciales y de canal en una única operación, Xception maneja estos dos tipos de correlaciones por separado. Este enfoque mejora la capacidad del modelo para aprender patrones complejos sin aumentar el costo computacional. La convolución *depthwise* se encarga de captar las correlaciones espaciales dentro de cada canal, mientras que la convolución *pointwise* permite la interacción entre canales.

**Ventajas de la Arquitectura Xception** Xception presenta varias ventajas que la hacen particularmente eficiente para tareas de clasificación y reconocimiento de imágenes:

- **Reducción en el Número de Parámetros:** Gracias al uso de convoluciones separables en profundidad, Xception logra una reducción significativa en el número de parámetros, lo cual facilita su entrenamiento en entornos de recursos limitados.
- **Eficiencia en el Flujo de Gradientes:** La inclusión de bloques residuales permite un flujo constante de gradientes, evitando la desaparición de gradiente y mejorando la capacidad de aprendizaje en redes profundas.
- **Flexibilidad en Transferencia de Aprendizaje:** Al separar correlaciones espaciales y de canal, Xception es capaz de aprender características detalladas que pueden adaptarse eficientemente a nuevas tareas mediante *transfer learning*.

**Aplicaciones y Rendimiento** Xception ha mostrado un rendimiento competitivo en varias tareas de visión por computadora, desde clasificación de imágenes hasta detección de objetos. Su estructura modular, basada en convoluciones separables y bloques residuales, permite un balance entre profundidad y eficiencia computacional, haciéndolo adecuado para implementaciones en las que se requiere alta precisión sin un costo computacional elevado [10]. Al mejorar el diseño de Inception, Xception logra una precisión comparable o superior en varios conjuntos de datos de referencia con un uso de recursos más eficiente.

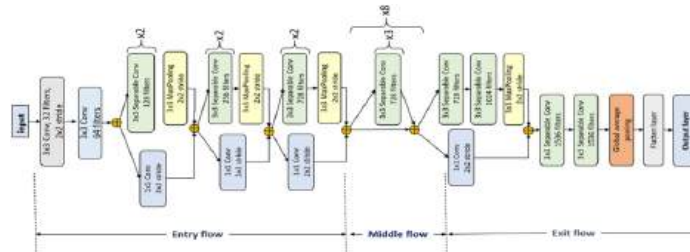


Figura 4.8: Arquitectura del modelo Xception.

#### 4.11.9. Evaluación y Métricas de Desempeño

Las métricas de desempeño son fundamentales para evaluar la efectividad de los modelos de *Deep Learning*. Entre las más comunes se incluyen:

- **Precisión (*Accuracy*):** Proporción de predicciones correctas sobre el total.
- **Exactitud (*Precision*) y Exhaustividad (*Recall*):** Miden la capacidad del modelo para identificar correctamente las clases relevantes.

- **F1 Score:** Media armónica de la precisión y el *recall*, útil en escenarios con clases desbalanceadas:

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

## 4.12. Explainable AI (XAI)

En los últimos años, el desarrollo y la implementación de sistemas de inteligencia artificial (IA) han crecido exponencialmente, abarcando múltiples sectores como la medicina, la finanzas, y la gestión de residuos. Sin embargo, la complejidad de estos modelos, en especial los basados en redes neuronales profundas, ha generado una preocupación importante sobre su falta de interpretabilidad. Es en este contexto donde surge el campo de *Explainable AI (XAI)*.

### 4.12.1. Definición y Objetivo de *Explainable AI*

*Explainable AI (XAI)* se refiere a un conjunto de métodos y técnicas que permiten a los humanos entender y confiar en las decisiones o predicciones realizadas por los sistemas de IA. La principal motivación detrás de XAI es mitigar el efecto de las “cajas negras” en los modelos de aprendizaje automático, proporcionando explicaciones claras y comprensibles sobre cómo y por qué un modelo ha tomado una decisión específica. Según el estudio de Adadi y Berrada (2018), XAI no solo mejora la interpretabilidad sino que también incrementa la transparencia y la confianza en los sistemas de IA [2].

### 4.12.2. Importancia de la Interpretabilidad

La necesidad de interpretabilidad es particularmente crítica en aplicaciones sensibles, como diagnósticos médicos, sistemas de crédito, y control de vehículos autónomos. Un modelo interpretable permite:

- Detectar posibles sesgos en el modelo.
- Entender el proceso de toma de decisiones del modelo.
- Mejorar la confianza de los usuarios y desarrolladores en la IA.
- Facilitar la depuración y mejora continua del sistema.

### 4.12.3. Métodos de *Explainable AI*

Los métodos de XAI se dividen en dos categorías principales: *Métodos Post-Hoc* y *Métodos Intrínsecos*. A continuación, se describen algunos de los métodos más populares en cada categoría.

#### *Métodos Intrínsecos*

Estos métodos incorporan interpretabilidad dentro de la estructura del modelo desde el inicio. Algunos ejemplos incluyen:

- **Modelos Lineales y Árboles de Decisión:** Estos son modelos que, por su naturaleza, son fácilmente interpretables. Por ejemplo, un árbol de decisión proporciona reglas claras y explícitas sobre cómo se hacen las predicciones.
- **Modelos de Regresión:** La regresión lineal, por ejemplo, es un modelo simple donde los coeficientes pueden interpretarse como la influencia de cada variable de entrada en la predicción.
- **Redes Neuronales Interpretables:** Estas son arquitecturas diseñadas para ser más comprensibles, como las redes neuronales sumativas o modelos simplificados.

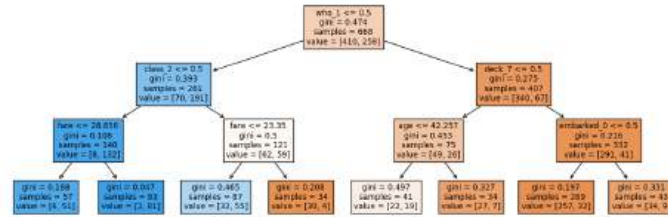


Figura 4.9: Ejemplo de un Árbol de Decisión utilizado para la Interpretabilidad.

### Métodos Post-Hoc

Los métodos post-hoc se aplican a modelos complejos ya entrenados para proporcionar interpretabilidad. Algunos de los métodos más reconocidos son:

- **Grad-CAM (Gradient-weighted Class Activation Mapping):** Grad-CAM es un método visual que utiliza los gradientes de las predicciones a lo largo de las capas de convolución para identificar las regiones de una imagen que contribuyen a la clasificación. Esto se usa comúnmente en modelos de visión por computadora [85].

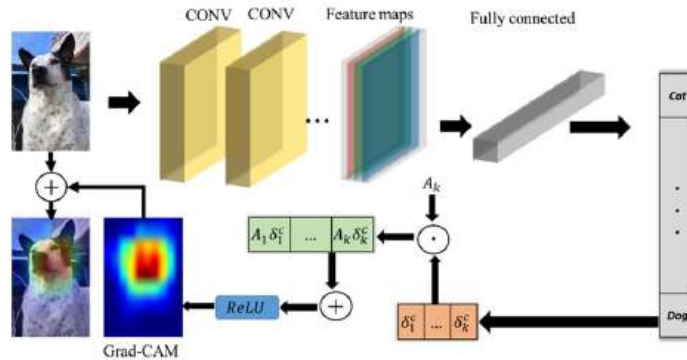


Figura 4.10: Ejemplo de Grad-CAM mostrando las áreas de atención del modelo.

- **LIME (Local Interpretable Model-agnostic Explanations):** LIME funciona creando un modelo interpretable, como una regresión lineal, que aproxima el comportamiento del modelo original en una vecindad local alrededor de una predicción específica. Esto ayuda a entender cómo las variables de entrada afectan una predicción particular [79].

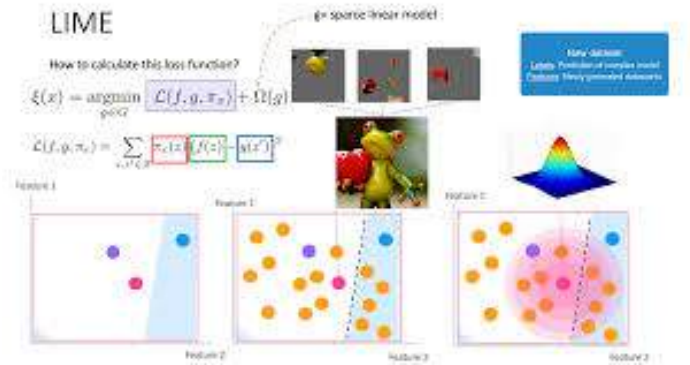


Figura 4.11: Ejemplo de LIME explicando las decisiones de un modelo de clasificación.

- **SHAP (SHapley Additive exPlanations)**: Basado en la teoría de juegos, SHAP calcula la contribución de cada característica a la predicción final de un modelo, asignando un valor de Shapley a cada variable. Este método es ampliamente usado debido a su capacidad para proporcionar explicaciones consistentes y justas [57].

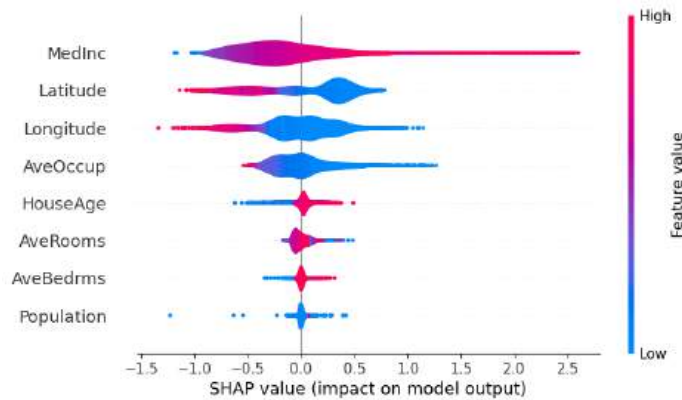


Figura 4.12: Ejemplo de SHAP mostrando la contribución de cada variable en una predicción.

- **SmoothGrad**: Este método suaviza los mapas de saliencia generando múltiples versiones de una imagen con ruido y promediando los resultados. De esta manera, se reduce el ruido y se obtiene una visualización más clara de las áreas de importancia [88].

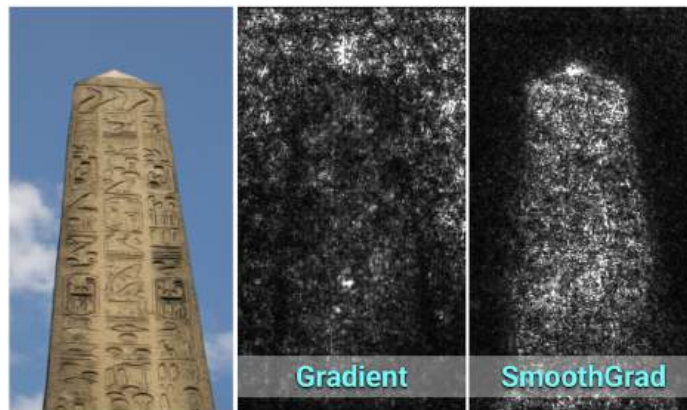


Figura 4.13: Ejemplo de SmoothGrad reduciendo el ruido en los mapas de saliencia.

- **Interpretación Basada en Contrafactuales:** Esta técnica se centra en generar ejemplos que muestren cómo los cambios en las variables de entrada afectarían el resultado de una predicción. Esto es útil para entender qué factores son críticos para una decisión dada.

#### 4.12.4. Desafíos en *Explainable AI*

Aunque XAI ha avanzado significativamente, todavía enfrenta varios desafíos. Uno de los principales es el equilibrio entre interpretabilidad y precisión: a menudo, los modelos más interpretables, como los árboles de decisión, no son tan precisos como las redes neuronales profundas. Además, la subjetividad en lo que constituye una “buena” explicación varía según el usuario y el contexto [42].

Otro problema es la escalabilidad, ya que algunas técnicas de XAI, como LIME, pueden ser computacionalmente costosas. Además, los sesgos inherentes a los datos de entrenamiento pueden ser difíciles de detectar, incluso con modelos explicables.

#### 4.12.5. Aplicaciones de *Explainable AI*

XAI se está aplicando en numerosos campos, como:

- **Medicina:** Para justificar las decisiones de diagnóstico y tratamiento.
- **Finanzas:** En la explicación de aprobaciones o rechazos de crédito.
- **Seguridad:** En la interpretación de decisiones tomadas por sistemas de vigilancia y análisis de comportamiento.

Estas aplicaciones son fundamentales para garantizar que los sistemas de IA sean transparentes, éticos y responsables.

### 4.13. Modelos de *Machine Learning* e *Inteligencia Artificial* en la Gestión de Residuos

Diversos modelos de *Machine Learning* e *Inteligencia Artificial* han sido desarrollados para abordar los desafíos en la gestión de residuos, desde la clasificación automática hasta la optimización de la recolección y reciclaje. A continuación, se describen algunos de los casos más destacados.

#### 4.13.1. RecycleNet: Clasificación Automática de Residuos

RecycleNet es un modelo basado en redes neuronales convolucionales (CNNs) diseñado para la clasificación automática de residuos en tiempo real. Este modelo utiliza imágenes capturadas en cintas transportadoras para identificar y separar materiales reciclables como papel, plástico, metal y vidrio. Su implementación ha permitido mejorar significativamente la eficiencia en plantas de reciclaje, reduciendo la necesidad de intervención humana [28].

#### 4.13.2. Sistema de Gestión Inteligente de Residuos de SAS

Un modelo analítico creado por Agustín Marín, de la empresa SAS, aplica técnicas avanzadas de *Machine Learning* para optimizar la recolección de residuos en entornos urbanos. Este sistema

predice la generación de residuos en diferentes áreas, lo que permite planificar las rutas de recolección de manera más eficiente, reduciendo costos y emisiones de carbono [59].

#### 4.13.3. Project Recicla AI

Este proyecto, desarrollado por DoradoReciclaAI, utiliza dos fuentes de datos: "TrashNetz "Waste Classification Data", para entrenar un modelo robusto que mejora la precisión en la clasificación de residuos. El modelo se enfoca en la identificación de materiales difíciles de clasificar, como plásticos degradables y residuos electrónicos, y se integra en aplicaciones móviles para facilitar el reciclaje en comunidades locales [23].

#### 4.13.4. Aplicación Web para la Clasificación de Residuos

Desarrollada como parte de un trabajo de investigación en la Universidad EAN, esta aplicación web utiliza un modelo de *Machine Learning* para clasificar residuos sólidos de manera automática. El modelo está diseñado para ayudar a los usuarios a identificar correctamente los residuos que pueden reciclarse, contribuyendo a una mejor gestión desde el origen [28].

#### 4.13.5. WasteNet: Modelo de Clasificación de Residuos en *Smart Cities*

WasteNet es un modelo diseñado para clasificar residuos en entornos urbanos inteligentes (*Smart Cities*). Utiliza una combinación de sensores IoT y redes neuronales convolucionales (CNNs) para identificar y clasificar residuos en tiempo real, optimizando así el proceso de reciclaje y reduciendo la cantidad de residuos enviados a vertederos. WasteNet ha sido implementado en varias ciudades como parte de iniciativas para mejorar la sostenibilidad urbana [13].

#### 4.13.6. EcoSort: Optimización de Rutas de Recolección de Residuos

EcoSort es un sistema basado en *Machine Learning* que optimiza las rutas de recolección de residuos utilizando modelos predictivos. Este sistema analiza datos históricos y en tiempo real sobre la generación de residuos para planificar rutas más eficientes, reduciendo así el consumo de combustible y las emisiones de CO<sub>2</sub>. EcoSort ha sido implementado en ciudades de Europa como parte de programas de gestión de residuos sostenibles [30].

#### 4.13.7. AI-Waste: Clasificación Avanzada de Residuos Electrónicos

AI-Waste es un modelo que utiliza técnicas de *Deep Learning* para clasificar componentes de residuos electrónicos (*e-waste*). Este modelo ha sido desarrollado para identificar y separar metales valiosos como el oro y el cobre en dispositivos electrónicos desechados, facilitando el reciclaje de materiales escasos y la reducción de la minería extractiva [90].

#### 4.13.8. Consideraciones Finales

La implementación efectiva de ML e IA en la gestión de residuos requiere una comprensión sólida de los modelos, hiperparámetros y métricas de desempeño. Además, la capacidad de ajustar

y optimizar estos elementos es clave para desarrollar sistemas robustos y precisos que contribuyan a la sostenibilidad y eficiencia en la gestión de residuos.

## 4.14. Visión por Computadora para Clasificación de Imágenes

La visión por computadora es una rama de la inteligencia artificial que permite a las máquinas interpretar y comprender el mundo visual a partir de imágenes o videos. En el contexto de la gestión de residuos, la visión por computadora es clave para desarrollar clasificadores automáticos que puedan identificar y categorizar diferentes tipos de residuos a partir de imágenes.

### 4.14.1. Conceptos Clave en Visión por Computadora

Los conceptos fundamentales en visión por computadora incluyen el preprocesamiento de imágenes, la extracción de características y la clasificación. A continuación, se describen algunos de los aspectos más importantes:

#### Preprocesamiento de Imágenes

El preprocesamiento de imágenes es una etapa crítica en la cual las imágenes se ajustan y preparan para ser utilizadas por los modelos de *Machine Learning* o *Deep Learning*. Las técnicas comunes incluyen:

- **Redimensionamiento:** Ajuste del tamaño de la imagen a las dimensiones requeridas por el modelo. Generalmente, las imágenes se redimensionan para que tengan el mismo tamaño, lo cual simplifica el procesamiento.
- **Normalización:** Escala los valores de los píxeles para que estén dentro de un rango estándar, típicamente entre 0 y 1, o entre -1 y 1. Esto ayuda a estabilizar el entrenamiento del modelo [46].
- **Aumento de Datos (*Data Augmentation*):** Técnicas que generan nuevas imágenes a partir de las existentes mediante transformaciones como rotación, traslación, reflejo y ajuste de brillo. Esto aumenta la diversidad del conjunto de datos y reduce el sobreajuste [100].

### 4.14.2. Extracción de Características

La extracción de características consiste en identificar y extraer información relevante de las imágenes que permita al modelo distinguir entre diferentes clases de residuos. En las redes neuronales convolucionales (CNNs), esto se realiza automáticamente mediante capas convolucionales que aplican filtros sobre la imagen para detectar bordes, texturas y patrones.

#### Operación Convolutiva

Matemáticamente, la operación convolutiva en una imagen se define como:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n) \cdot K(m, n)$$



Donde:

- $I$  es la matriz de la imagen de entrada.
- $K$  es el filtro (kernel) aplicado.
- $S(i, j)$  es la salida de la convolución en la posición  $(i, j)$ .

#### 4.14.3. Clasificación de Imágenes

La clasificación de imágenes implica asignar una etiqueta o categoría a una imagen dada. Los modelos de *Deep Learning*, como las CNNs, han demostrado ser altamente eficaces para esta tarea. El proceso de clasificación generalmente incluye:

- **Redes Completamente Conectadas:** Al final de las capas convolucionales, la imagen es "aplanada" y pasa a través de una o más capas completamente conectadas que realizan la clasificación final [14].
- **Función de Activación Softmax:** La última capa suele utilizar la función Softmax para convertir los valores de salida en probabilidades:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

Donde  $K$  es el número de clases [48].

#### 4.14.4. Relación con *Machine Learning* e *Inteligencia Artificial*

La visión por computadora es un campo interdisciplinario que se apoya en métodos de *Machine Learning* e *Inteligencia Artificial* para resolver problemas complejos. En el contexto de la clasificación de residuos, las CNNs son la elección preferida debido a su capacidad para aprender representaciones jerárquicas de las imágenes [47].

#### 4.14.5. Optimización y Entrenamiento de Modelos de Visión por Computadora

El entrenamiento de modelos de visión por computadora implica minimizar una función de pérdida que mide la discrepancia entre las predicciones del modelo y las etiquetas verdaderas. El método de optimización más común es el gradiente descendente estocástico (SGD), con variantes como el algoritmo Adam.

##### Función de Pérdida Categórica

Para tareas de clasificación, la función de pérdida categórica entropía cruzada se utiliza ampliamente:

$$L(y, \hat{y}) = - \sum_{i=1}^K y_i \log(\hat{y}_i)$$

Donde:

- $y_i$  es el vector de etiquetas verdaderas.
- $\hat{y}_i$  es la predicción del modelo.

#### 4.14.6. Visualización de Características y Mapas de Calor

Una herramienta útil para entender cómo un modelo de visión por computadora toma decisiones es la visualización de mapas de calor. Los mapas de activación visualizan las áreas de la imagen que más influyen en la decisión del modelo, proporcionando información valiosa sobre el proceso de clasificación.

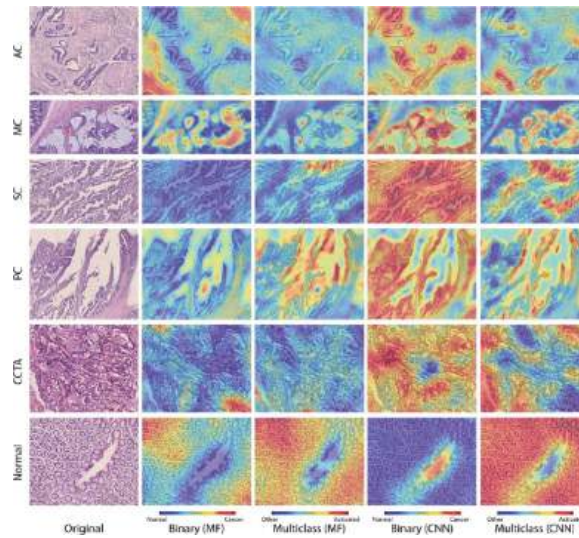


Figura 4.14: Ejemplo de un mapa de calor que muestra las áreas de una imagen que más contribuyen a la clasificación.

#### 4.14.7. Consideraciones Finales

La visión por computadora, en combinación con técnicas de *Machine Learning* e *Inteligencia Artificial*, proporciona una base poderosa para la automatización en la clasificación de residuos. La implementación adecuada de preprocesamiento, extracción de características y optimización del modelo es esencial para lograr un sistema preciso y robusto.

### 4.15. Modelos de Lenguaje de Gran Escala (*Large Language Models, LLMs*) y Chatbots Inteligentes

Los *Large Language Models* (LLMs) son modelos de inteligencia artificial diseñados para procesar y generar lenguaje natural. Estos modelos han alcanzado un rendimiento sobresaliente en tareas de procesamiento de lenguaje natural (NLP), como traducción automática, generación de texto, resumen, y respuestas a preguntas. Los LLMs han revolucionado la forma en que se desarrollan los chatbots, permitiendo interacciones más naturales y precisas con los usuarios.

### 4.15.1. Conceptos Clave en LLMs

Los LLMs se basan en arquitecturas de redes neuronales profundas, principalmente los transformadores (*Transformers*), que son capaces de manejar secuencias largas de texto de manera eficiente.

#### Transformers

El transformador es la arquitectura base detrás de los LLMs modernos. Se compone de un mecanismo de atención, que permite al modelo enfocarse en diferentes partes del input dependiendo del contexto. La ecuación del mecanismo de atención es:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Donde:

- $Q$  es la matriz de consultas (queries).
- $K$  es la matriz de claves (keys).
- $V$  es la matriz de valores (values).
- $d_k$  es la dimensión de las claves.

### 4.15.2. Implementación de Chatbots Inteligentes con LLMs

Los chatbots basados en LLMs como GPT-3 o BERT pueden comprender y generar respuestas coherentes en un amplio rango de contextos. La capacidad de estos modelos para captar matices del lenguaje humano les permite interactuar de manera más efectiva con los usuarios, brindando asistencia personalizada y relevante en tiempo real.

#### Función de Costo en LLMs

Durante el entrenamiento, los LLMs se optimizan para minimizar una función de costo que mide la discrepancia entre las predicciones del modelo y las palabras reales en la secuencia de entrenamiento. Generalmente, la función de costo utilizada es la entropía cruzada:

$$L = - \sum_{t=1}^T \log P(y_t | y_1, \dots, y_{t-1}, \theta)$$

Donde:

- $y_t$  es la palabra en el tiempo  $t$ .
- $\theta$  son los parámetros del modelo.

## Ejemplo de Arquitectura de un Chatbot Inteligente

Un chatbot inteligente basado en LLMs generalmente incluye las siguientes etapas:

- **Preprocesamiento del Texto:** Limpieza y tokenización del texto de entrada.
- **Generación de Respuesta:** El LLM genera una respuesta basada en el contexto y el historial de la conversación.
- **Postprocesamiento:** Ajustes finales a la respuesta para mejorar la naturalidad.

### 4.15.3. Ejemplos de LLMs Poderosos para Chatbots

- **GPT-3:** Un modelo con 175 mil millones de parámetros, desarrollado por OpenAI, utilizado en aplicaciones de chatbots, asistentes virtuales, y generación de contenido.
- **BERT:** Modelo de Google basado en transformadores bidireccionales, optimizado para tareas como la clasificación de texto, análisis de sentimiento, y respuesta a preguntas.
- **T5:** Un modelo de Google que aborda todas las tareas de NLP como problemas de traducción, logrando resultados impresionantes en diversas aplicaciones.

### 4.15.4. Implicaciones y Consideraciones Éticas en el Uso de LLMs

Dado el poder de los LLMs, existen preocupaciones éticas relacionadas con su uso, como la generación de desinformación, sesgo en los resultados, y la privacidad de los datos utilizados para entrenarlos. Es crucial implementar salvaguardas y políticas para mitigar estos riesgos [62], [44].

### 4.15.5. Visualización de la Atención en LLMs

La visualización de la atención en modelos como BERT permite entender qué palabras o frases son más relevantes para una tarea específica. Esta visualización puede proporcionar información valiosa sobre cómo los modelos toman decisiones.

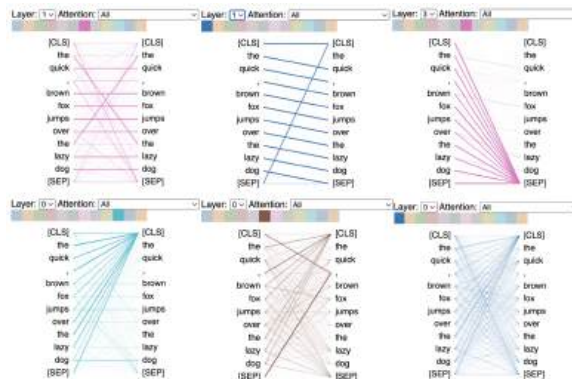


Figura 4.15: Ejemplo de un mapa de atención en un LLM, mostrando la relevancia de diferentes palabras en un contexto específico.

## 4.16. Tecnologías para la Implementación de Modelos de *Inteligencia Artificial*

El desarrollo e implementación de modelos de *Machine Learning* e *Inteligencia Artificial* requieren el uso de herramientas y plataformas avanzadas que faciliten la programación, entrenamiento, y despliegue de dichos modelos. En esta sección, se abordarán las principales tecnologías utilizadas en este campo.

### 4.16.1. Python

Python es el lenguaje de programación por excelencia para *Machine Learning* e *Inteligencia Artificial*. Su sintaxis simple y su vasta colección de librerías especializadas lo hacen ideal para el desarrollo de modelos. Librerías como NumPy, Pandas, y Scikit-learn son fundamentales para tareas de manipulación de datos, estadística, y aprendizaje automático [62].

### 4.16.2. Kaggle

Kaggle es una plataforma que permite a los investigadores y desarrolladores compartir datasets, colaborar en proyectos, y competir en desafíos de *Machine Learning*. Además, Kaggle también es un repositorio importante de datasets, ofreciendo una vasta colección de conjuntos de datos públicos que pueden ser utilizados para entrenar y probar modelos [53].

### 4.16.3. Jupyter Notebooks

Jupyter Notebooks es una herramienta de código abierto que permite combinar código, texto, y visualizaciones en un solo documento. Es ampliamente utilizado en el desarrollo de modelos debido a su flexibilidad y capacidad para facilitar la experimentación rápida [52].

### 4.16.4. Google Colab Pro

Google Colab Pro es una versión mejorada de Google Colab que ofrece acceso a recursos adicionales como GPUs y TPUs, tiempos de ejecución más largos, y mayores límites de uso. Es una herramienta esencial para el entrenamiento de modelos de *Deep Learning* de gran escala.

#### Recursos y Características de Google Colab Pro

Google Colab Pro proporciona acceso a GPUs de alta gama como Tesla P100 y V100, lo que permite entrenar modelos complejos de manera eficiente. También ofrece integraciones directas con bibliotecas como TensorFlow, PyTorch, y Hugging Face, facilitando la implementación de modelos avanzados [32].

#### Integraciones y Facilidad de Uso

Colab se integra con Google Drive, permitiendo a los usuarios almacenar y acceder fácilmente a datasets, modelos, y resultados. Además, su compatibilidad con bibliotecas de Python como

TensorFlow y PyTorch lo convierte en una plataforma ideal para el desarrollo de proyectos de IA [27].

#### 4.16.5. Librerías de Python

Python cuenta con una gran variedad de librerías especializadas en *Machine Learning* y *Deep Learning*, entre las cuales destacan:

- **TensorFlow:** Desarrollada por Google, TensorFlow es una de las bibliotecas más populares para *Deep Learning*. Proporciona herramientas para construir y entrenar redes neuronales profundas y modelos de ML a gran escala [98].
- **PyTorch:** Creada por Facebook, PyTorch es otra biblioteca popular para *Deep Learning*, conocida por su flexibilidad y facilidad de uso. PyTorch es ampliamente utilizado en investigación y producción [74].
- **Hugging Face:** Una plataforma que ofrece herramientas para la implementación de modelos de procesamiento de lenguaje natural (NLP), como BERT y GPT-3. Hugging Face facilita el uso de LLMs preentrenados y su ajuste fino [27].
- **OpenCV:** Una librería de visión por computadora fundamental para el procesamiento y análisis de imágenes. Se utiliza para tareas como la detección de objetos, la transformación de imágenes y la extracción de características visuales [67].
- **Keras:** Una API de alto nivel construida sobre TensorFlow, que facilita la creación de modelos de *Deep Learning* mediante una interfaz más sencilla y amigable [54].
- **Scikit-Image:** Librería basada en NumPy para el procesamiento de imágenes, útil para tareas como la segmentación, la clasificación, y la manipulación de imágenes [83].
- **Pillow:** Derivada de la biblioteca PIL (Python Imaging Library), Pillow es una librería para el procesamiento y manipulación de imágenes en Python. Permite realizar una amplia gama de operaciones de edición de imágenes, como el cambio de tamaño, el recorte, la rotación y la aplicación de filtros. Además, es compatible con numerosos formatos de imagen, como JPEG, PNG y GIF. Pillow resulta útil en tareas de preprocesamiento de imágenes para aplicaciones de *Machine Learning* y *Deep Learning*, y permite realizar transformaciones geométricas y manipulaciones a nivel de píxel, que son esenciales para el aumento de datos (*data augmentation*) y la preparación de imágenes de entrada [11].

#### 4.16.6. CUDA y el Uso de Tarjetas Gráficas

CUDA es una plataforma de computación paralela desarrollada por NVIDIA que permite a los desarrolladores utilizar GPUs para acelerar aplicaciones de cómputo intensivo como el entrenamiento de modelos de *Deep Learning*. El uso de GPUs es fundamental para reducir los tiempos de entrenamiento y manejar modelos con millones de parámetros.

##### Optimización con CUDA

Las tarjetas gráficas como NVIDIA Tesla, Quadro y RTX son compatibles con CUDA, lo que permite ejecutar cálculos en paralelo de manera más rápida que con CPUs tradicionales. La optimización del entrenamiento con CUDA es un factor clave para la eficiencia en proyectos de IA a gran escala [65].

#### 4.16.7. Uso de Datasets para Entrenamiento

El uso de datasets de alta calidad es crucial para entrenar modelos de *Machine Learning* efectivos. Los datasets deben estar bien etiquetados, ser diversos y representar adecuadamente las clases de interés.

##### Características de un Buen Dataset

Un buen dataset debe cumplir con las siguientes características:

- **Relevancia:** Los datos deben estar directamente relacionados con el problema que se busca resolver.
- **Calidad:** Los datos deben estar limpios, sin valores atípicos o inconsistencias significativas.
- **Diversidad:** El dataset debe cubrir un amplio rango de variaciones posibles para evitar sesgos en el modelo.
- **Tamaño:** Un mayor número de ejemplos ayuda al modelo a generalizar mejor.

##### Ejemplos de Datasets para Clasificación de Residuos

Existen varios datasets que se han utilizado para la clasificación de residuos:

- **TrashNet:** Un conjunto de datos compuesto por imágenes de diferentes tipos de residuos como vidrio, papel, metal, plástico y basura orgánica [101].
- **Waste Classification Data:** Dataset que contiene imágenes de residuos clasificadas en varias categorías, útil para entrenar modelos de clasificación automática [16].
- **TACO:** Dataset de imágenes para la clasificación y detección de residuos, utilizado en competiciones de Kaggle [95].

#### 4.16.8. Relevancia de Kaggle como Repositorio de Datasets

Kaggle no solo es una plataforma de competencias, sino también uno de los repositorios más grandes de datasets públicos. Los usuarios pueden acceder a una amplia gama de datasets de alta calidad para tareas como la clasificación de imágenes, el análisis de texto, y más [53].

### 4.17. TensorFlow y Visión por Computadora con Redes Neuronales Convolucionales (CNNs)

TensorFlow es una de las plataformas más utilizadas para el desarrollo de modelos de *Deep Learning*, especialmente en el ámbito de la visión por computadora. Proporciona herramientas robustas para construir y entrenar redes neuronales convolucionales (CNNs), que son el estándar de facto para tareas de clasificación de imágenes, detección de objetos, y segmentación.

### 4.17.1. Redes Neuronales Convolucionales (CNNs)

Las redes neuronales convolucionales (CNNs) son un tipo de red neuronal especialmente diseñada para procesar datos con una estructura de cuadrícula, como imágenes. Se componen de capas convolucionales que aplican filtros sobre la entrada para extraer características relevantes, seguidas de capas de agrupación (*pooling*) que reducen la dimensionalidad.

#### Estructura de una CNN

Una CNN típica consta de:

- **Capas Convolucionales:** Aplican convoluciones sobre la entrada para detectar características como bordes y texturas. Matemáticamente, la operación convolucional se define como:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n) \cdot K(m, n)$$

donde  $I$  es la matriz de la imagen y  $K$  es el kernel o filtro aplicado.

- **Capas de Agrupación (*Pooling*):** Reducen la dimensionalidad de la salida de las capas convolucionales, conservando la información más relevante.
- **Capas Completamente Conectadas:** Al final de la red, estas capas realizan la clasificación en función de las características extraídas.

### 4.17.2. Implementación de CNNs en TensorFlow

TensorFlow facilita la construcción e implementación de CNNs mediante su API de alto nivel, Keras. A continuación se presenta un ejemplo básico de cómo implementar una CNN para la clasificación de imágenes:

```

1  import tensorflow as tf
2  from tensorflow.keras import layers, models
3
4  # Definición del modelo
5  model = models.Sequential([
6      layers.Conv2D(32, (3, 3), activation='relu', input_shape=(64, 64, 3)),
7      layers.MaxPooling2D((2, 2)),
8      layers.Conv2D(64, (3, 3), activation='relu'),
9      layers.MaxPooling2D((2, 2)),
10     layers.Conv2D(64, (3, 3), activation='relu'),
11     layers.Flatten(),
12     layers.Dense(64, activation='relu'),
13     layers.Dense(10, activation='softmax')
14 ])
15
16 # Compilación del modelo
17 model.compile(optimizer='adam',
18               loss='sparse_categorical_crossentropy',
19               metrics=['accuracy'])
20
21 # Entrenamiento del modelo
22 model.fit(train_images, train_labels, epochs=10, validation_data=(test_images, test_labels))

```



### 4.17.3. Ventajas de Usar TensorFlow en Visión por Computadora

TensorFlow ofrece varias ventajas para la implementación de modelos de visión por computadora:

- **Escalabilidad:** TensorFlow permite entrenar modelos en múltiples GPUs y TPUs, lo que es crucial para manejar grandes volúmenes de datos.
- **Compatibilidad con Keras:** La integración de Keras facilita el desarrollo rápido de modelos, con una sintaxis clara y simplificada.
- **Biblioteca de Modelos Preentrenados:** TensorFlow Hub proporciona acceso a modelos preentrenados que pueden ser ajustados para tareas específicas de visión por computadora.

### 4.17.4. Casos de Uso en Visión por Computadora

TensorFlow ha sido utilizado en numerosos proyectos de visión por computadora, que incluyen:

- **Clasificación de Residuos:** Implementaciones de CNNs entrenadas para clasificar imágenes de residuos en diferentes categorías (vidrio, papel, plástico, etc.).
- **Detección de Objetos:** Utilización de modelos como YOLO y SSD para detectar y localizar objetos en imágenes en tiempo real.
- **Segmentación Semántica:** Aplicación de redes como U-Net para segmentar imágenes en diferentes clases, útil en tareas médicas y de medio ambiente.

### 4.17.5. Optimización y Despliegue de Modelos en Producción

TensorFlow también facilita la optimización y despliegue de modelos en producción a través de TensorFlow Lite y TensorFlow Serving, permitiendo que los modelos se ejecuten en dispositivos móviles y servidores con baja latencia.

### 4.17.6. Conclusiones

La combinación de TensorFlow y CNNs proporciona una plataforma poderosa y flexible para resolver problemas complejos de visión por computadora. La capacidad de escalar, junto con la facilidad de uso y la extensa comunidad de usuarios, hacen de TensorFlow una opción ideal para proyectos de clasificación de imágenes y más.

## 4.18. Hugging Face y su Rol en la Construcción de Chatbots Inteligentes

Hugging Face es una plataforma y comunidad líder en el ámbito del procesamiento del lenguaje natural (NLP). Ofrece una amplia colección de modelos preentrenados, incluyendo *Large Language Models* (LLMs) como BERT, GPT-3, y T5, que son fundamentales para la construcción de chatbots inteligentes. Hugging Face facilita el acceso a estos modelos a través de su *Transformers* library, permitiendo a los desarrolladores implementar soluciones avanzadas de NLP con un mínimo esfuerzo.

### 4.18.1. Acceso a Modelos Preentrenados

Uno de los mayores beneficios de Hugging Face es su extensa biblioteca de modelos preentrenados. Los desarrolladores pueden acceder a estos modelos y utilizarlos directamente en sus aplicaciones sin necesidad de entrenar desde cero. Esto ahorra tiempo y recursos significativos. Ejemplos de modelos disponibles incluyen:

- **GPT-2 y GPT-3:** Modelos de generación de texto que pueden ser utilizados para construir chatbots que mantienen conversaciones fluidas y naturales.
- **BERT:** Modelo bidireccional optimizado para tareas de comprensión del lenguaje, como respuesta a preguntas y clasificación de texto.
- **T5:** Un modelo versátil que aborda diversas tareas de NLP como problemas de traducción, incluyendo la generación de respuestas en chatbots.

### 4.18.2. Implementación de Chatbots con Hugging Face

Hugging Face facilita la implementación de chatbots a través de su API y su librería *Transformers*. Los pasos típicos para construir un chatbot incluyen:

- **Carga del Modelo:** Utilizar la función *from\_pretrained* para cargar un modelo preentrenado relevante para la tarea del chatbot.
- **Ajuste Fino (*Fine-Tuning*):** Ajustar el modelo a un conjunto de datos específico para personalizar las respuestas del chatbot según el dominio.
- **Despliegue:** Integrar el modelo ajustado en una aplicación web o móvil, o incluso en plataformas de mensajería como Slack o Telegram.

Ejemplo de código para cargar un modelo preentrenado y generar respuestas:

```
1 from transformers import AutoModelForCausalLM, AutoTokenizer
2
3 tokenizer = AutoTokenizer.from_pretrained("gpt2")
4 model = AutoModelForCausalLM.from_pretrained("gpt2")
5
6 inputs = tokenizer("Hola, ¿cómo estás?", return_tensors="pt")
7 outputs = model.generate(inputs.input_ids, max_length=50, do_sample=True)
8
9 print(tokenizer.decode(outputs[0], skip_special_tokens=True))
```

### 4.18.3. Facilidades Adicionales de Hugging Face

Además de los modelos preentrenados, Hugging Face ofrece herramientas como **Model Hub** y **Datasets**, que proporcionan acceso a miles de modelos y conjuntos de datos públicos, facilitando la experimentación y el desarrollo de soluciones NLP.

#### 4.18.4. Integraciones con Google Colab y AWS

Hugging Face se integra fácilmente con Google Colab, lo que permite aprovechar los recursos de GPU para entrenar y ajustar modelos de manera eficiente. También se integra con Amazon Web Services (AWS) a través de SageMaker, facilitando el despliegue de modelos en producción.

#### 4.18.5. Consideraciones Éticas en el Uso de LLMs

Dado que los modelos preentrenados de Hugging Face pueden tener sesgos inherentes derivados de los datos con los que fueron entrenados, es crucial considerar las implicaciones éticas de su uso, especialmente en aplicaciones sensibles como chatbots que interactúan con usuarios vulnerables. Implementar mecanismos de revisión y ajuste es fundamental para mitigar estos riesgos.

#### 4.18.6. Relevancia para la Construcción de Chatbots Inteligentes

La plataforma Hugging Face democratiza el acceso a modelos avanzados de NLP, permitiendo que tanto desarrolladores experimentados como novatos puedan construir chatbots potentes y eficientes. La facilidad para cargar modelos, ajustarlos, y desplegarlos convierte a Hugging Face en una herramienta esencial para cualquier proyecto que implique procesamiento de lenguaje natural.

### 4.19. OpenAI y su API para Modelos de Lenguaje

OpenAI es una organización dedicada a la investigación y desarrollo de inteligencia artificial avanzada. Uno de sus logros más conocidos es la creación de modelos de lenguaje natural basados en arquitecturas de redes neuronales profundas, como la serie de modelos GPT (Generative Pre-trained Transformer). Los modelos GPT son herramientas versátiles capaces de realizar tareas de generación y comprensión de lenguaje, tales como resumen, traducción, respuesta a preguntas y generación de texto. En particular, el modelo GPT-3.5-turbo se distingue por su eficiencia y alta capacidad de comprensión contextual, características que lo hacen adecuado para aplicaciones de asistente virtual [7].

#### 4.19.1. API de OpenAI

La API de OpenAI permite el acceso a los modelos de lenguaje mediante un enfoque basado en la nube. Con esta API, los desarrolladores pueden integrar modelos de procesamiento de lenguaje natural (PLN) en aplicaciones sin necesidad de contar con infraestructura computacional especializada. La API acepta entradas de texto en lenguaje natural, que el modelo procesa para generar una respuesta adecuada basada en su entrenamiento.

##### Características Principales de la API

- **Interfaz Sencilla y Flexible:** La API de OpenAI se basa en peticiones HTTP estándar, utilizando el método `POST` para enviar datos. Los desarrolladores solo necesitan una clave de API para acceder al servicio, y la documentación proporciona parámetros configurables como el modelo específico a utilizar (e.g., GPT-3.5-turbo), la longitud de respuesta en *tokens*, y el nivel de creatividad o precisión de las respuestas (*temperature*).

- **Control de Tokens:** Los modelos de OpenAI operan en base a tokens, una unidad que representa palabras, fragmentos de palabras o caracteres especiales en el texto. El control de tokens permite a los desarrolladores limitar la longitud de las respuestas y gestionar el costo de uso de la API, ya que OpenAI cobra según la cantidad de tokens procesados.
- **Versatilidad en la Configuración:** La API permite configurar parámetros como `temperature`, que ajusta el grado de aleatoriedad en las respuestas, y `max_tokens`, que define el límite de tokens en la respuesta generada. Estos ajustes permiten adaptar el comportamiento del modelo según la tarea específica.

#### 4.19.2. GPT-3.5-turbo: Características Técnicas y Arquitectura

GPT-3.5-turbo es una variante optimizada del modelo GPT-3, diseñada para ofrecer mayor eficiencia en términos de costos de procesamiento y memoria, sin perder precisión en la generación de texto. Basado en la arquitectura Transformer, GPT-3.5-turbo emplea mecanismos de autoatención que le permiten capturar dependencias contextuales a través de múltiples capas, proporcionando respuestas coherentes y contextualmente precisas.

**Arquitectura de Transformer** GPT-3.5-turbo sigue la arquitectura Transformer, introducida por Vaswani et al. en 2017 [107]. Esta arquitectura es altamente eficiente en modelos de lenguaje debido a su capacidad para aprender relaciones contextuales mediante un mecanismo de autoatención. El Transformer utiliza capas de autoatención y mecanismos de *feed-forward* en paralelo, lo cual permite al modelo manejar relaciones a largo plazo en el texto de entrada. En el caso de GPT-3.5-turbo, el modelo está preentrenado en un vasto corpus de texto y luego ajustado para una amplia gama de tareas de PLN.

**Tokens y Manejo de Información** GPT-3.5-turbo divide el texto en unidades llamadas *tokens*, que pueden ser palabras completas, fragmentos de palabras o caracteres especiales. Este modelo maneja los tokens en dos fases principales:

- **Tokenización:** Al recibir un texto, GPT-3.5-turbo lo descompone en tokens mediante un proceso de tokenización. Esto permite que el modelo entienda y procesa el texto en fragmentos que representan palabras individuales o combinaciones de caracteres.
- **Límite de Tokens y Costos:** Cada solicitud a la API incluye una restricción de tokens, especificada en el parámetro `max_tokens`. Dado que OpenAI cobra por el uso de tokens, controlar este parámetro ayuda a optimizar el costo y evitar respuestas excesivamente largas. Además, la longitud de contexto del modelo también está limitada por la capacidad de tokens, lo cual implica que GPT-3.5-turbo puede procesar hasta un número específico de tokens (generalmente 4096 o 8192 tokens en total, dependiendo de la versión de la API) antes de que la información anterior empiece a desaparecer.

**Generación de Respuestas y Parámetros de Control** La generación de texto en GPT-3.5-turbo está regulada por varios parámetros:

- **Temperature:** Este parámetro ajusta la creatividad del modelo. Valores de `temperature` más altos, como 0.8, producen respuestas más diversas y creativas, mientras que valores bajos, como 0.2, generan respuestas más coherentes y conservadoras.
- **Top-p (Núcleo de muestreo):** En lugar de elegir tokens basándose en la probabilidad más alta, el parámetro `top-p` permite al modelo seleccionar tokens de un subconjunto que cumpla

un umbral de probabilidad. Este método permite generar respuestas de alta calidad y reducir respuestas inusuales.

- **Frecuencia y Penalización de Presencia:** Estos parámetros ajustan la probabilidad de que el modelo repita o utilice ciertas palabras. La penalización de frecuencia reduce la probabilidad de repetición de palabras ya usadas en la conversación, mientras que la penalización de presencia desalienta al modelo de mencionar palabras específicas según las preferencias del usuario.

#### 4.19.3. Ventajas de GPT-3.5-turbo en Procesamiento de Lenguaje Natural

GPT-3.5-turbo es ampliamente utilizado en aplicaciones que requieren una comprensión profunda del lenguaje natural debido a su capacidad para generar texto de alta calidad y su flexibilidad. Las principales ventajas de este modelo incluyen:

- **Capacidad de Comprensión Contextual:** Gracias a su arquitectura Transformer y preentrenamiento extenso, GPT-3.5-turbo puede captar contextos complejos, manteniendo coherencia en respuestas largas y encadenadas.
- **Adaptabilidad a Múltiples Tareas:** Desde asistencia al usuario hasta generación de contenido y traducción, GPT-3.5-turbo se adapta fácilmente a una amplia gama de aplicaciones de PLN mediante ajustes en los parámetros de la API.
- **Eficiencia en el Uso de Recursos:** La optimización en la cantidad de parámetros y la eficiencia en el manejo de tokens permiten que GPT-3.5-turbo ofrezca un rendimiento de alta calidad a un menor costo computacional.

Este modelo ha transformado el campo del PLN al ofrecer capacidades de lenguaje que permiten su aplicación en sistemas de asistencia, chatbots, creación de contenido y análisis de texto, facilitando el desarrollo de interfaces más naturales y accesibles para los usuarios [7].

#### 4.20. FastAPI y Jinja2: Herramientas para el Desarrollo de Aplicaciones Web

FastAPI y Jinja2 son dos herramientas populares en el desarrollo de aplicaciones web modernas. FastAPI es un marco de desarrollo web de alto rendimiento basado en Python, especialmente diseñado para la construcción de APIs rápidas y escalables. Jinja2, por su parte, es un motor de plantillas que permite generar contenido HTML dinámico, facilitando la creación de interfaces web interactivas. En conjunto, estas herramientas ofrecen una solución robusta para el desarrollo de aplicaciones web rápidas, eficientes y mantenibles.

##### FastAPI: Desarrollo de APIs Rápidas y Escalables

FastAPI es un marco de trabajo basado en Python para la creación de APIs de alto rendimiento. Diseñado por Sebastián Ramírez, FastAPI se basa en el estándar ASGI (Asynchronous Server Gateway Interface), lo que permite la ejecución de peticiones de manera asíncrona y mejora significativamente el rendimiento de las aplicaciones web [75].

### Características Principales de FastAPI

- **Soporte para Operaciones Asíncronas:** FastAPI permite el uso de funciones asíncronas (`async`) para gestionar múltiples solicitudes de forma simultánea, lo que optimiza el tiempo de respuesta en aplicaciones con alta demanda. Esta capacidad lo hace especialmente útil en escenarios donde es necesario procesar peticiones concurrentes, como en chatbots o sistemas de recomendación.
- **Validación Automática de Datos:** Basado en las anotaciones de tipo de Python (type hints), FastAPI valida automáticamente los datos de entrada, simplificando la gestión de errores y asegurando que las solicitudes cumplen con los requisitos esperados. Esta funcionalidad se logra a través de la integración con la biblioteca Pydantic, que permite la validación y conversión de datos en tiempo de ejecución.
- **Documentación Automática:** FastAPI genera documentación automática de las APIs mediante las especificaciones OpenAPI y Swagger UI. Esto facilita la colaboración entre desarrolladores y permite a los usuarios de la API comprender de forma clara las rutas, parámetros y respuestas que ofrece el servicio.
- **Alto Rendimiento:** Gracias a su compatibilidad con ASGI y su arquitectura asíncrona, FastAPI ofrece un rendimiento superior en comparación con otros marcos web tradicionales de Python, como Flask o Django. La combinación de la ejecución asíncrona y la optimización interna de FastAPI lo convierte en una opción ideal para aplicaciones que requieren alta eficiencia.

### Jinja2: Motor de Plantillas para Contenido HTML Dinámico

Jinja2 es un motor de plantillas en Python que permite la generación dinámica de contenido HTML. Desarrollado por Armin Ronacher, Jinja2 es ampliamente utilizado en aplicaciones web para separar la lógica de negocio de la presentación de la interfaz, facilitando la creación de interfaces web que son tanto dinámicas como mantenibles [80].

### Características Principales de Jinja2

- **Sintaxis Similar a Python:** La sintaxis de Jinja2 es intuitiva y similar a la de Python, lo que facilita su adopción por parte de desarrolladores familiarizados con el lenguaje. Esto permite realizar operaciones lógicas y de control de flujo directamente en las plantillas, como bucles, condiciones y filtros.
- **Separación de Lógica y Presentación:** Jinja2 permite separar la lógica de presentación de la lógica de negocio de una aplicación, albergando el código HTML y el contenido dinámico en archivos de plantilla. Esto facilita el mantenimiento del código y promueve la reutilización de componentes de la interfaz.
- **Filtros y Extensiones:** Jinja2 ofrece una amplia variedad de filtros y extensiones que permiten manipular datos de manera eficiente en las plantillas. Los filtros permiten transformar valores (como formatear fechas o cambiar el formato de texto), mientras que las extensiones añaden funcionalidades avanzadas, como la inclusión de plantillas y el uso de macros.
- **Seguridad Integrada:** Jinja2 incluye funcionalidades de escapado automático, lo cual ayuda a proteger las aplicaciones web contra ataques de inyección de código HTML (*Cross-Site Scripting* o XSS). Esta característica hace de Jinja2 una herramienta segura y confiable para la generación de contenido dinámico en aplicaciones web.

## Integración de FastAPI y Jinja2 en Aplicaciones Web

La combinación de FastAPI y Jinja2 permite construir aplicaciones web eficientes y dinámicas. FastAPI proporciona las rutas y gestiona las solicitudes de la API, mientras que Jinja2 facilita la creación de plantillas HTML personalizables que responden de manera dinámica a las peticiones de los usuarios. Esta integración es especialmente útil en el desarrollo de aplicaciones que requieren una interfaz web, como dashboards de administración, sistemas de recomendación o asistentes virtuales. En conjunto, FastAPI y Jinja2 ofrecen una solución completa para desarrollar aplicaciones web con capacidades avanzadas y rendimiento optimizado.

## 4.21. MLOps y MLFlow

El desarrollo y la implementación de modelos de *Machine Learning (ML)* a escala requieren prácticas estandarizadas para gestionar eficazmente el ciclo de vida de los modelos. Este enfoque se conoce como MLOps, que combina principios de *DevOps* con técnicas de ingeniería de datos y ML [5]. MLOps es esencial para proyectos que buscan un despliegue continuo y un mantenimiento adecuado de los modelos de ML en producción.

### 4.21.1. Concepto de MLOps

MLOps, o *Machine Learning Operations*, se refiere a un conjunto de prácticas diseñadas para automatizar y optimizar el ciclo de vida completo de los modelos de ML, desde la preparación de datos y el entrenamiento del modelo hasta su despliegue y monitoreo [110]. La idea central es garantizar que los modelos sean reproducibles, confiables y escalables.

#### Componentes Clave de MLOps

- **Preparación y Preprocesamiento de Datos:** La gestión de datos es fundamental en MLOps, ya que los modelos dependen de la calidad de los datos. Se implementan técnicas para limpiar y procesar datos de manera consistente, asegurando su validez y relevancia [84].
- **Entrenamiento del Modelo:** Incluye la selección de algoritmos, ajuste de hiperparámetros y validación. MLOps exige que los experimentos sean rastreables y reproducibles para optimizar y comparar modelos de manera efectiva [6].
- **Versionado de Modelos:** Los modelos evolucionan con el tiempo, por lo que el versionado es crucial. Permite rastrear los cambios en datos, arquitecturas y configuraciones, facilitando la selección de modelos óptimos para su despliegue [71].
- **Despliegue y Escalado:** La automatización del despliegue en entornos de producción asegura actualizaciones rápidas y seguras, garantizando que los modelos puedan manejar diferentes cargas de trabajo [110].
- **Monitoreo y Mantenimiento:** Se requiere un monitoreo en tiempo real para detectar desviaciones en el rendimiento del modelo, especialmente cuando hay cambios en la distribución de los datos, conocido como *drift* de datos [6].
- **Automatización de Pruebas:** Asegura la calidad y estabilidad del modelo, verificando que las predicciones sean consistentes y fiables antes de desplegar el modelo [84].
- **Gestión de la Infraestructura:** Se encarga de administrar los recursos para entrenamiento y despliegue, garantizando eficiencia y seguridad [71].

### Beneficios de MLOps

- **Escalabilidad:** Permite a las organizaciones escalar modelos de manera eficiente, ajustándose a las demandas cambiantes [5].
- **Automatización y Eficiencia:** La automatización de tareas reduce el riesgo de errores manuales y ahorra tiempo en el proceso de desarrollo [110].
- **Reproducibilidad:** Garantiza que los experimentos puedan replicarse con precisión, lo cual es vital para validar los resultados y mantener la coherencia [6].
- **Mejor Monitoreo y Mantenimiento:** Permite un seguimiento continuo del rendimiento, facilitando la detección de problemas y ajustes necesarios [84].

#### 4.21.2. MLFlow: Herramienta de MLOps

MLFlow es una plataforma de código abierto que gestiona el ciclo de vida de los modelos de ML. Proporciona un marco integral para rastrear experimentos, versionar modelos, manejar flujos de trabajo y desplegar modelos [110]. Fue desarrollada por Databricks y ha ganado gran aceptación en la comunidad de ciencia de datos.

### Componentes de MLFlow

MLFlow se compone de cuatro módulos principales:

- **MLFlow Tracking:** Permite registrar y consultar experimentos, almacenando parámetros, métricas y artefactos del modelo. Esto facilita el análisis comparativo y la reproducibilidad [17].
- **MLFlow Projects:** Define un formato estándar para empaquetar proyectos, incluyendo dependencias y entornos, lo cual simplifica la ejecución reproducible en distintos entornos [17].
- **MLFlow Models:** Ofrece un estándar para empaquetar modelos que permite una fácil integración con herramientas de inferencia y despliegue [110].
- **MLFlow Model Registry:** Un sistema centralizado para gestionar las versiones de los modelos y promoverlos a producción, asegurando una gestión efectiva del ciclo de vida [17].

### Ventajas de Utilizar MLFlow

- **Rastreo Integral de Experimentos:** Facilita el seguimiento detallado de los experimentos, promoviendo una gestión eficaz y análisis exhaustivo [17].
- **Gestión Eficiente de Modelos:** El versionado estructurado ayuda a mantener un control efectivo sobre los modelos, facilitando colaboraciones y mejoras [110].
- **Promueve la Colaboración:** MLFlow centraliza el manejo de experimentos y modelos, permitiendo a los equipos trabajar de manera más coordinada [6].
- **Integración con CI/CD:** Se puede integrar con sistemas de integración y entrega continua, optimizando el proceso de despliegue y actualización en producción [17].



### Casos de Uso de MLFlow

MLFlow se aplica en varios contextos, tales como:

- **Investigación y Desarrollo:** Ayuda a gestionar y rastrear experimentos, agilizando el proceso de investigación científica [71].
- **Despliegue de Modelos a Producción:** Facilita la transición de los modelos desde el entorno de desarrollo hasta la producción, asegurando un rendimiento robusto y continuo [5].
- **Monitoreo y Actualización:** Proporciona herramientas que permiten rastrear el rendimiento y actualizar los modelos según sea necesario, asegurando que sigan siendo efectivos a lo largo del tiempo [6].

La integración de MLFlow y MLOps en proyectos de ML es esencial para garantizar que los modelos no solo se desarrollen eficazmente, sino que también se desplieguen y mantengan adecuadamente en entornos complejos.

La aplicación de la inteligencia artificial en la gestión de residuos se ha expandido rápidamente en los últimos años, en respuesta a la creciente necesidad de métodos eficientes para clasificar y reciclar materiales. Las técnicas avanzadas de visión por computadora, especialmente mediante el uso de redes neuronales convolucionales (CNN), han demostrado un alto rendimiento en la identificación y clasificación automática de residuos a partir de imágenes. Este proyecto se enmarca en una serie de investigaciones que exploran el uso de modelos de *deep learning* y transferencias de aprendizaje (*transfer learning*) para mejorar la precisión y eficiencia en la gestión de desechos.

#### 5.0.1. Clasificación de Residuos mediante Redes Neuronales Convolucionales

Uno de los estudios más destacados en esta área fue realizado por Poudel y Poudyal [72], quienes investigaron la clasificación de materiales de desecho utilizando CNN basadas en transferencias de aprendizaje. En su trabajo, emplearon modelos preentrenados como InceptionV3, InceptionResNetV2, Xception, VGG19, MobileNet, ResNet50 y DenseNet201 para clasificar imágenes de residuos en siete categorías: cartón, vidrio, metal, orgánico, papel, plástico y basura. Los autores concluyeron que el modelo InceptionV3 tuvo un desempeño superior en términos de precisión de clasificación, mientras que el modelo VGG19 presentó una menor precisión. Este estudio resalta la importancia de la selección del modelo y de la técnica de *fine-tuning* en la clasificación de residuos, mostrando que la arquitectura CNN es adecuada para aplicaciones de gestión de residuos a nivel industrial y urbano [10].

#### 5.0.2. Importancia de la Inteligencia Artificial en la Gestión de Residuos

El uso de algoritmos de inteligencia artificial para la clasificación de residuos permite reducir el error humano y optimizar los procesos de reciclaje, contribuyendo a una gestión más sostenible del medio ambiente [110]. La integración de modelos de *deep learning* en sistemas de clasificación automática ofrece una ventaja significativa en comparación con métodos tradicionales de clasificación manual, especialmente en entornos donde la eficiencia y la precisión son cruciales. Varios estudios han demostrado que los modelos de redes neuronales convolucionales, cuando se aplican con técnicas de *transfer learning*, pueden mejorar la adaptabilidad de los sistemas de clasificación a diferentes

tipos de residuos y condiciones de imagen [72].

### 5.0.3. Implementación de Chatbots Inteligentes para la Asistencia en el Reciclaje

En paralelo con la clasificación de residuos mediante imágenes, los chatbots inteligentes han surgido como herramientas valiosas para proporcionar asistencia en tiempo real a usuarios interesados en el reciclaje. Los avances en procesamiento de lenguaje natural (NLP) han facilitado la creación de asistentes virtuales capaces de responder preguntas sobre la correcta clasificación de residuos, alineándose con normativas específicas, como las aplicadas en Guatemala. Este proyecto implementa un chatbot especializado que, utilizando un modelo de lenguaje avanzado, proporciona recomendaciones y orientaciones personalizadas sobre la disposición adecuada de residuos, mejorando la educación ambiental y facilitando la adopción de prácticas sostenibles en la comunidad [10].

### 5.0.4. Desafíos en la Clasificación de Residuos y la Asistencia Basada en IA

Aunque los sistemas de clasificación basados en inteligencia artificial han demostrado su efectividad, existen desafíos inherentes en la implementación de modelos precisos y escalables. La variabilidad en los tipos de residuos, la complejidad de los fondos en las imágenes y las limitaciones en la disponibilidad de datos de entrenamiento son algunos de los obstáculos que enfrenta el desarrollo de estos sistemas [110]. Por otro lado, la implementación de chatbots para asistencia en temas específicos requiere un entrenamiento adecuado en datos locales y la habilidad de rechazar preguntas fuera de contexto de manera educada y precisa.

Estas tendencias y desafíos ponen de manifiesto el valor de explorar aplicaciones de inteligencia artificial como herramientas prácticas para mejorar la gestión de residuos y fomentar prácticas de reciclaje en distintos contextos. Este proyecto contribuye a la investigación en este campo, ofreciendo una solución que aborda tanto la clasificación automática de residuos como la asistencia interactiva, con un enfoque particular en las necesidades y normativas de la Ciudad de Guatemala.

El presente megaproyecto se enfocó en el desarrollo de un sistema de inteligencia artificial aplicado a la gestión de residuos en la Ciudad de Guatemala, compuesto por dos agentes principales: un modelo de clasificación de residuos mediante imágenes y un chatbot inteligente. Ambos agentes fueron diseñados y evaluados con el objetivo de asistir a los usuarios en la identificación y clasificación de residuos, promoviendo así prácticas de reciclaje y manejo responsable de desechos. A continuación, se detallan los aspectos específicos del alcance de este proyecto:

- **Clasificación Automática de Residuos:** Se implementó un modelo de *deep learning* basado en la arquitectura Xception, optimizado para la identificación de seis categorías de residuos: orgánicos, plásticos, papel, vidrio, metal y baterías. La elección de Xception fue estratégica debido a su eficacia en la extracción de características complejas, asegurando un balance entre precisión y eficiencia. El modelo fue entrenado con un conjunto de datos diverso y se realizaron pruebas exhaustivas, incluyendo técnicas de *Explainable AI* (como GradCAM y SmoothGrad) para validar y entender las decisiones del modelo. Además, se evaluó su rendimiento en escenarios con imágenes reales, incluyendo contextos con fondos complejos, demostrando su robustez para aplicaciones móviles y en tiempo real.
- **Interacción Asistida mediante Chatbot Inteligente:** Se desarrolló un chatbot especializado en la temática de clasificación y reciclaje de residuos, basado en las capacidades de la API de OpenAI. Este agente conversacional fue entrenado para ofrecer respuestas precisas y alineadas con las normativas locales de la Ciudad de Guatemala, utilizando procesamiento avanzado de lenguaje natural. Se realizaron pruebas cualitativas con expertos ambientales para evaluar la relevancia, precisión y capacidad de respuesta del chatbot, obteniendo un índice promedio de calidad del 90.86 %. Además, se midió el tiempo promedio de respuesta, confirmando que es adecuado para su uso en aplicaciones interactivas y en tiempo real.
- **Evaluación de Precisión y Rendimiento:** Como parte fundamental del proyecto, se definieron métricas específicas para evaluar la efectividad de ambos agentes. El modelo de clasificación fue sometido a análisis detallados de *precision* y *loss history*, así como a evaluaciones basadas en matrices de confusión para verificar la correcta identificación de residuos. Por su parte, el chatbot fue evaluado en cinco criterios clave mediante una rúbrica estandarizada, lo que proporcionó una visión comprensiva de su desempeño. La implementación de técnicas de MLOps con MLFlow permitió realizar un seguimiento exhaustivo de los experimentos y versionamiento del modelo, lo cual es un avance hacia la gestión escalable y reproducible de modelos de ML.

- **Limitaciones en el Ámbito de Clasificación:** El sistema desarrollado se limita a clasificar seis tipos específicos de residuos y no abarca otras categorías como desechos peligrosos o electrónicos. Aunque se contempló la extensibilidad del sistema, su rendimiento y capacidad actual están restringidos a los datos utilizados para el entrenamiento. Asimismo, se identificaron desafíos asociados a la clasificación en entornos con alta variabilidad visual, como imágenes con fondos complejos, donde el modelo mostró un leve descenso en precisión.
- **Aplicación Local en Guatemala:** Este proyecto se desarrolló con un enfoque particular en la Ciudad de Guatemala, siguiendo las normativas y prácticas recomendadas por la Unidad de Control Ambiental local. Aunque las metodologías y modelos desarrollados podrían ser transferibles a otras regiones, el sistema fue específicamente ajustado para las condiciones y requisitos guatemaltecos, asegurando la alineación con las políticas y necesidades ambientales de la región.

El alcance de este proyecto destaca la viabilidad de integrar inteligencia artificial en la gestión de residuos de manera efectiva. La aplicación de técnicas avanzadas de *machine learning* y la incorporación de MLOps garantizan la escalabilidad y reproducibilidad del sistema, estableciendo una base sólida para futuras mejoras. Además, los agentes desarrollados no solo facilitan la educación y el manejo adecuado de residuos, sino que también promueven una mayor conciencia ambiental en la Ciudad de Guatemala, sentando las bases para expansiones que aborden una mayor diversidad de materiales y requisitos internacionales.

## 7.1. Selección de Tecnologías

En este proyecto, se han seleccionado diversas tecnologías y frameworks para implementar tanto el modelo de clasificación de imágenes mediante redes neuronales como el sistema de chat inteligente. La selección de estas herramientas se ha realizado con base en criterios de rendimiento, soporte en la comunidad, flexibilidad y facilidad de integración en una aplicación móvil destinada a promover prácticas sostenibles. A continuación, se explican las tecnologías seleccionadas y las razones para su elección.

### 7.1.1. Plataforma de Entrenamiento y Desarrollo: Google Colab

Para el entrenamiento y desarrollo del modelo de clasificación de imágenes, se optó por utilizar **Google Colab** en lugar de servicios de computación en la nube de pago o máquinas locales. La decisión de emplear Google Colab se basa en una combinación de criterios de costo, disponibilidad de recursos de cómputo y compatibilidad con la arquitectura utilizada en este proyecto.

#### Ventajas de Google Colab en el Contexto del Proyecto

- **Acceso a Recursos de Cómputo Potentes:** Google Colab proporciona acceso gratuito a GPU (como las NVIDIA Tesla T4 o K80) y ocasionalmente a TPU, recursos fundamentales para entrenar modelos de redes neuronales profundas de manera eficiente. Esta capacidad de cómputo es clave para realizar entrenamientos en menos tiempo, algo que no sería viable en una máquina local sin tarjeta gráfica dedicada.
- **Optimización de Costo-Beneficio:** A diferencia de servicios de nube dedicados como AWS, Azure o Google Cloud Platform, el uso de Google Colab no incurre en costos significativos para la mayoría de los recursos necesarios en este proyecto. Esto es especialmente relevante dada la necesidad de optimizar el presupuesto, ya que los servicios de nube suelen implicar tarifas elevadas para instancias con GPU o TPU.
- **Compatibilidad y Facilidad de Uso:** Google Colab está basado en Jupyter Notebooks, lo que facilita la iteración, prueba y desarrollo de código en un entorno interactivo. Esta

plataforma permite la ejecución de código en Python, simplificando la implementación y prueba de los modelos mediante una interfaz amigable y ampliamente utilizada en investigación y educación.

- **Limitaciones de Hardware Local:** El equipo de desarrollo es una MacBook Pro con arquitectura M1, que si bien es potente para tareas generales, presenta limitaciones para el entrenamiento de redes neuronales profundas. La arquitectura ARM de los procesadores M1, además de la falta de una tarjeta gráfica dedicada, hace que los tiempos de entrenamiento se prolonguen notablemente si se utiliza el CPU. Además, las librerías principales como TensorFlow y PyTorch aún tienen soporte limitado en ARM para tareas de entrenamiento en GPU, lo cual refuerza la elección de Google Colab como una solución más eficiente.

### Desventajas de Alternativas Consideradas

- **Servicios de Nube:** A pesar de ofrecer recursos de alta calidad y soporte avanzado, el costo de plataformas como AWS y Google Cloud Platform resulta prohibitivo para entrenamientos de larga duración y de alta demanda computacional en GPU. La opción de usar Google Colab permite reducir estos costos sin sacrificar el rendimiento.
- **Máquina Local (MacBook Pro con M1):** El uso de una máquina local con arquitectura M1 no es viable para este proyecto, debido a las limitaciones en rendimiento y soporte de GPU. Los tiempos de entrenamiento en CPU serían sustancialmente más largos, lo cual afectaría el proceso iterativo de desarrollo y ajuste de los modelos.

En conclusión, la selección de Google Colab como plataforma de desarrollo y entrenamiento permite aprovechar recursos computacionales avanzados sin incurrir en altos costos. Esto garantiza una solución económica y accesible, capaz de gestionar los requisitos computacionales del modelo de clasificación de residuos desarrollado en este proyecto.

### 7.1.2. Framework de Redes Neuronales: TensorFlow

Para el desarrollo del modelo de clasificación de imágenes, se ha elegido **TensorFlow** como el framework de *Deep Learning* principal, descartando otras opciones como **PyTorch**. La elección de TensorFlow se justifica en los siguientes aspectos [1]:

- **Rendimiento y Eficiencia:** TensorFlow es ampliamente reconocido por su capacidad de realizar cálculos de alto rendimiento mediante el soporte de procesamiento en GPU y TPU, lo cual es esencial para entrenar redes neuronales convolucionales (CNNs) que requieren gran poder de cómputo.
- **Compatibilidad y Portabilidad:** TensorFlow facilita la exportación de modelos a distintos formatos, como TensorFlow Lite, lo cual permite desplegar modelos en dispositivos móviles con eficiencia en el uso de recursos. Esta funcionalidad es crítica en el contexto de una aplicación móvil para la clasificación de residuos en tiempo real.
- **Soporte en la Comunidad y Documentación:** TensorFlow cuenta con una amplia comunidad de desarrolladores y un extenso soporte de documentación y recursos, lo cual facilita el desarrollo y la resolución de problemas, maximizando la eficiencia en el desarrollo.

Aunque PyTorch es una alternativa popular y altamente valorada en investigación, se optó por TensorFlow debido a su mejor soporte para despliegue en producción y optimización en dispositivos móviles, factores determinantes en este proyecto.

### 7.1.3. Manipulación y Preprocesamiento de Imágenes: PILLOW

Para el preprocesamiento y la manipulación de imágenes se ha seleccionado **PILLOW** (Python Imaging Library) [12]. Esta biblioteca permite realizar operaciones esenciales de preprocesamiento, tales como redimensionamiento, rotación, ajuste de brillo y normalización de las imágenes de entrada. Estas funciones son indispensables para preparar los datos que se ingresan al modelo de clasificación de residuos.

- **Facilidad de Uso:** PILLOW es una herramienta sencilla y altamente compatible con otras bibliotecas de *Machine Learning* y *Deep Learning*, lo cual facilita su integración en el flujo de trabajo.
- **Optimización y Velocidad:** La biblioteca está optimizada para el procesamiento rápido de imágenes, lo cual es ideal para su uso en aplicaciones que requieren análisis en tiempo real.

La elección de PILLOW sobre otras librerías como OpenCV responde a su simplicidad en la manipulación de imágenes en el contexto específico de este proyecto, donde no se requieren las capacidades avanzadas de visión por computadora que ofrece OpenCV.

### 7.1.4. Modelo de Chat Inteligente: OpenAI API con GPT-3.5 Turbo

Para el sistema de chat inteligente se ha seleccionado la **API de OpenAI**, específicamente el modelo **GPT-3.5 Turbo** [8]. Este modelo fue preferido sobre alternativas como **GPT-4** o modelos de *Hugging Face* (como Llama 2 y Llama 3) por las siguientes razones:

- **Eficiencia en Tiempo de Respuesta:** GPT-3.5 Turbo presenta un balance entre precisión y velocidad, lo cual permite responder rápidamente a las consultas de los usuarios. Dado que la aplicación se orienta a proporcionar consejos e información de manera ágil, esta característica es fundamental.
- **Costo-Beneficio:** GPT-3.5 Turbo tiene un menor costo de operación comparado con GPT-4, lo que resulta adecuado para el presupuesto del proyecto.
- **Fiabilidad y Flexibilidad:** La API de OpenAI ofrece estabilidad en producción y se encuentra optimizada para el procesamiento de lenguaje natural en tareas de asistencia y generación de respuestas. La disponibilidad de OpenAI asegura un servicio continuo y confiable.

A pesar de que modelos como Llama 2 y 3 presentan capacidades avanzadas y son opciones de código abierto, la API de OpenAI se ha elegido por su rendimiento en producción, facilidad de implementación y menor demanda de infraestructura, lo cual resulta ideal para este proyecto.

### 7.1.5. Framework para Endpoints de Chat: FastAPI

Para la implementación de los endpoints del sistema de chat, se ha seleccionado **FastAPI** sobre otras opciones como **Flask**, debido a sus ventajas en términos de rendimiento y facilidad de desarrollo [76].

- **Alto Rendimiento:** FastAPI está optimizado para manejar múltiples peticiones concurrentes con una latencia mínima, lo cual es crítico para un chatbot que interactúa con varios usuarios en tiempo real.



- **Compatibilidad con Asincronía:** A diferencia de Flask, FastAPI está diseñado para soportar operaciones asincrónicas, lo que permite gestionar eficientemente las consultas a la API de OpenAI y mejorar la capacidad de respuesta del sistema.
- **Facilidad de Uso y Documentación Automática:** FastAPI genera automáticamente documentación de los endpoints utilizando *Swagger*, lo cual facilita las pruebas y la integración del sistema.

La decisión de elegir FastAPI sobre Flask se basa en su mejor rendimiento para aplicaciones de tiempo real y su compatibilidad con arquitecturas asincrónicas, lo cual contribuye a un sistema de chat eficiente y escalable.

### 7.1.6. Resumen de Ventajas de las Tecnologías Seleccionadas

Las herramientas seleccionadas cumplen con los requisitos de eficiencia, escalabilidad y facilidad de integración en un entorno de producción. La combinación de TensorFlow [1], PILLOW [12], la API de OpenAI [8] y FastAPI [76] permite implementar una solución robusta y optimizada para la clasificación de residuos y la interacción mediante un chatbot inteligente en una aplicación móvil.

## 7.2. Descripción del Dataset Utilizado

Para el entrenamiento y la validación del modelo de clasificación de residuos, se utilizó el dataset *Garbage Classification - 6 classes (775/class)*, disponible en Kaggle [99]. Este dataset globalizado, con imágenes de residuos provenientes de múltiples países, representa una amplia gama de contextos y condiciones ambientales, proporcionando así una base sólida para la generalización del modelo. El uso de un conjunto de datos internacional es particularmente relevante, ya que permite capturar las variaciones en las características visuales de los residuos que podrían encontrarse en distintas regiones, incluido Guatemala.

### 7.2.1. Características Principales del Dataset

El dataset consta de un total de 4650 imágenes, distribuidas uniformemente en seis categorías principales de residuos, con 775 imágenes por clase. Las características de cada clase se describen a continuación:

- **Plastic:** Compuesta de botellas, bolsas y envases de plástico. Los objetos de esta clase presentan bordes definidos y una textura translúcida o brillante que los distingue visualmente.
- **Paper:** Incluye papeles, cartones y otros productos fibrosos. Esta clase es caracterizada por superficies mate y bordes irregulares debido a la flexibilidad del material.
- **Organic:** Residuos biodegradables como restos de alimentos y vegetación. Los residuos orgánicos presentan una gran variedad de formas y colores naturales, con texturas menos uniformes y, a menudo, superficies rugosas.
- **Metal:** Principalmente latas y piezas metálicas. La clase de metal es fácilmente reconocible por sus superficies reflectantes y a menudo está acompañada de signos de oxidación o desgaste.
- **Glass:** Incluye botellas, frascos y otros objetos de vidrio. El vidrio se caracteriza por su transparencia o translucidez, reflejos y bordes lisos.

- **Battery:** Residuos peligrosos que incluyen baterías de distintos tamaños y formas. Las baterías suelen tener colores y etiquetas distintivas, además de una superficie generalmente lisa y regular.

La distribución balanceada entre las clases asegura que el modelo aprenda de manera equitativa las características distintivas de cada tipo de residuo, favoreciendo una clasificación precisa y sin sesgo de clase.

### 7.2.2. Relevancia del Dataset en el Contexto Guatemalteco

A pesar de su carácter internacional, el dataset ha demostrado ser adecuado para el contexto guatemalteco tras realizar pruebas exitosas con imágenes de productos locales. Las similitudes en las características visuales de residuos comunes en Guatemala y en otros países sugieren que este conjunto de datos es representativo y adecuado para capturar patrones visuales relevantes en el contexto local. Además, la diversidad en las condiciones de iluminación y variabilidad de texturas y colores en el dataset refuerza su capacidad de generalización, lo cual es crítico para aplicaciones en el mundo real.

### 7.3. Ejemplos Visuales de Imágenes en el Dataset

Para ilustrar la variabilidad y las características visuales de cada clase de residuos, la Figura 7.1 muestra ejemplos representativos de las seis categorías en el dataset. Estas imágenes destacan los patrones y texturas característicos que el modelo debe aprender para diferenciar entre tipos de residuos.

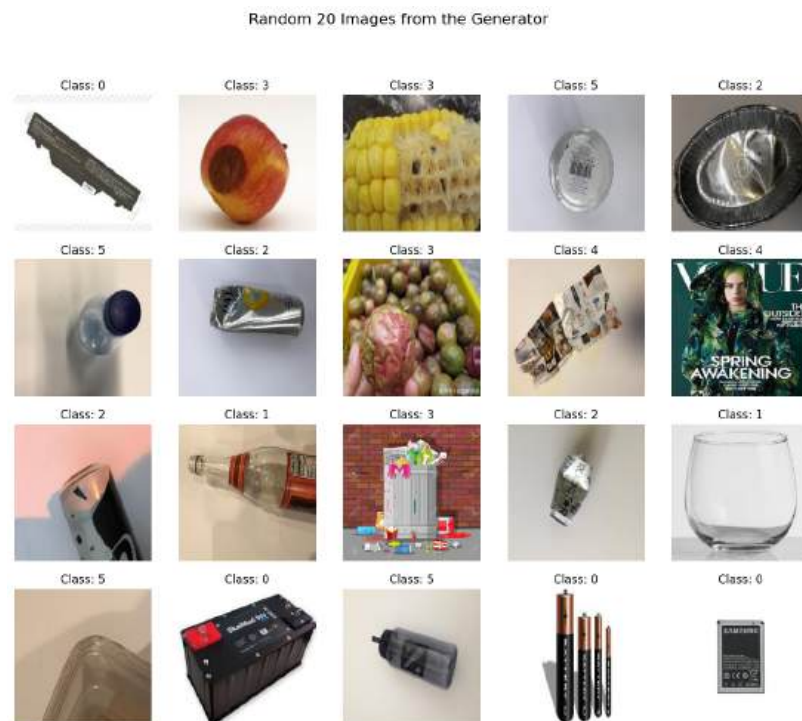


Figura 7.1: Ejemplos de imágenes de cada clase en el dataset: plástico, papel, orgánico, metal, vidrio y baterías.

La diversidad en las imágenes, desde residuos orgánicos hasta materiales peligrosos como baterías, enfatiza la necesidad de un modelo robusto que pueda identificar con precisión diferentes materiales en diversas condiciones de captura. Esta variabilidad es clave para el éxito del modelo en un entorno de implementación en tiempo real.

## 7.4. Justificación del Uso de Redes Neuronales Convolucionales (CNN) y Transfer Learning

Para el desarrollo del modelo de clasificación de residuos, se seleccionaron **Redes Neuronales Convolucionales (CNN)** debido a su reconocida eficacia en tareas de visión por computadora, particularmente en la clasificación de imágenes. Las CNNs han mostrado un alto rendimiento en la identificación de patrones visuales complejos, como formas, texturas y contornos, permitiendo distinguir entre diferentes materiales de residuos con precisión.

### 7.4.1. Ventajas del Uso de CNNs en Clasificación de Imágenes

Las CNNs están estructuradas en capas convolucionales, que extraen características locales de una imagen, como bordes, formas y texturas, permitiendo que el modelo aprenda representaciones jerárquicas de las características de los residuos. Esto resulta particularmente útil en la clasificación de residuos, ya que distintos materiales presentan características visuales distintivas que pueden ser aprendidas y generalizadas por el modelo.

- **Extracción Automática de Características:** A diferencia de otros enfoques que requieren un diseño manual de características, las CNNs pueden aprender directamente de los datos las características relevantes para la tarea de clasificación.
- **Invarianza a Transformaciones:** Las capas convolucionales permiten que el modelo sea invariante a pequeñas transformaciones como rotaciones o cambios de escala, lo cual es útil para clasificar residuos en diferentes posiciones y condiciones.
- **Eficiencia Computacional:** Aunque las CNNs requieren un poder de procesamiento significativo, son más eficientes que otros modelos complejos en la clasificación de imágenes, lo cual es ideal para implementar un modelo en un entorno de producción.

### 7.4.2. Transfer Learning para Mejorar la Precisión del Modelo

El **Transfer Learning** se utilizó en el proyecto como una estrategia para mejorar la precisión del modelo y reducir el tiempo de entrenamiento. Esta técnica implica utilizar una CNN preentrenada en un conjunto de datos de gran tamaño (como *ImageNet*) y ajustar sus capas finales para la tarea de clasificación de residuos. El uso de Transfer Learning ofrece varias ventajas:

- **Reducción del Tiempo de Entrenamiento:** Al utilizar un modelo preentrenado, solo es necesario reentrenar las capas finales, reduciendo significativamente el tiempo de entrenamiento en comparación con un modelo entrenado desde cero.
- **Mejora en la Generalización:** Los modelos preentrenados ya han aprendido características visuales generales (como bordes y texturas) que son útiles para múltiples tareas de clasificación, lo cual mejora la capacidad de generalización del modelo en el contexto de residuos.

- **Optimización de Recursos Computacionales:** Transfer Learning permite obtener un modelo preciso utilizando un conjunto de datos relativamente pequeño, lo cual es beneficioso en este proyecto debido a la cantidad limitada de imágenes en el dataset de residuos.

## 7.5. Justificación del Uso de Redes Neuronales Convolucionales (CNN), Transfer Learning y Fine-Tuning

Para el desarrollo del modelo de clasificación de residuos, se seleccionaron **Redes Neuronales Convolucionales (CNN)** debido a su reconocida eficacia en tareas de visión por computadora, particularmente en la clasificación de imágenes. Las CNNs han mostrado un alto rendimiento en la identificación de patrones visuales complejos, como formas, texturas y contornos, permitiendo distinguir entre diferentes materiales de residuos con precisión.

### 7.5.1. Ventajas del Uso de CNNs en Clasificación de Imágenes

Las CNNs están estructuradas en capas convolucionales, que extraen características locales de una imagen, como bordes, formas y texturas, permitiendo que el modelo aprenda representaciones jerárquicas de las características de los residuos. Esto resulta particularmente útil en la clasificación de residuos, ya que distintos materiales presentan características visuales distintivas que pueden ser aprendidas y generalizadas por el modelo.

- **Extracción Automática de Características:** A diferencia de otros enfoques que requieren un diseño manual de características, las CNNs pueden aprender directamente de los datos las características relevantes para la tarea de clasificación.
- **Invarianza a Transformaciones:** Las capas convolucionales permiten que el modelo sea invariante a pequeñas transformaciones como rotaciones o cambios de escala, lo cual es útil para clasificar residuos en diferentes posiciones y condiciones.
- **Eficiencia Computacional:** Aunque las CNNs requieren un poder de procesamiento significativo, son más eficientes que otros modelos complejos en la clasificación de imágenes, lo cual es ideal para implementar un modelo en un entorno de producción.

### 7.5.2. Transfer Learning y Fine-Tuning para Mejorar la Precisión del Modelo

El **Transfer Learning** se utilizó en el proyecto como una estrategia para mejorar la precisión del modelo y reducir el tiempo de entrenamiento. Esta técnica implica utilizar una CNN preentrenada en un conjunto de datos de gran tamaño (como *ImageNet*) y ajustar sus capas finales para la tarea de clasificación de residuos. Sin embargo, dado que el contexto específico de clasificación de residuos presenta características visuales particulares, se implementó un proceso adicional de **Fine-Tuning** para ajustar los parámetros del modelo a los datos específicos de residuos.

- **Reducción del Tiempo de Entrenamiento:** Al utilizar un modelo preentrenado, solo es necesario reentrenar las capas finales, reduciendo significativamente el tiempo de entrenamiento en comparación con un modelo entrenado desde cero.
- **Mejora en la Generalización:** Los modelos preentrenados ya han aprendido características visuales generales (como bordes y texturas) que son útiles para múltiples tareas de clasificación, lo cual mejora la capacidad de generalización del modelo en el contexto de residuos.

- **Fine-Tuning para Especialización:** Al realizar el fine-tuning, se ajustan tanto las capas finales como algunas capas internas del modelo preentrenado, permitiendo que el modelo refine características específicas de los residuos. Este proceso mejora la precisión en imágenes que presentan variabilidad en iluminación, texturas y formas.
- **Optimización de Recursos Computacionales:** Transfer Learning combinado con fine-tuning permite obtener un modelo preciso utilizando un conjunto de datos relativamente pequeño, lo cual es beneficioso en este proyecto debido a la cantidad limitada de imágenes en el dataset de residuos.

## 7.6. Inspiración en Estudios Anteriores

La arquitectura y la metodología para el modelo de clasificación de residuos se inspiraron en estudios recientes sobre el uso de IA en la gestión de residuos y en la clasificación de productos reciclables. Diversos trabajos han demostrado la efectividad de las CNNs y el Transfer Learning en aplicaciones de clasificación de residuos y objetos reciclables, sirviendo de referencia para el diseño y desarrollo de este proyecto.

### 7.6.1. Estudios Relacionados y Modelos Similares

Investigaciones como la de Ahmed et al. (2023) [4] utilizaron redes neuronales convolucionales para clasificar productos reciclables en distintas categorías, demostrando que los modelos de *deep learning* pueden alcanzar altos niveles de precisión en la identificación de residuos. Otro estudio relevante es el de Tan et al. (2023) [97], el cual desarrolló el sistema *AIWaste* basado en IA para la clasificación eficiente de residuos en distintos tipos de materiales. Estos estudios resaltan la viabilidad de emplear redes neuronales profundas para mejorar la gestión de residuos, especialmente en contextos urbanos.

### 7.6.2. Aplicación de las Metodologías Inspiradas

Basándonos en estos estudios, se decidió implementar una arquitectura CNN preentrenada y ajustada mediante Transfer Learning. Este enfoque permite aprovechar el conocimiento ya aprendido en modelos como ResNet o VGG, adaptándolo a las clases específicas de residuos de este proyecto. La precisión alcanzada en investigaciones previas ha establecido una meta de rendimiento para este proyecto, lo cual guiará el análisis de resultados y la evaluación del modelo en fases posteriores.

## 7.7. Metodología para la Selección de Modelos y Preparación de Datos

Para determinar el modelo que mejor se adapta a la clasificación de residuos en este proyecto, se seleccionaron cinco arquitecturas populares de Transfer Learning: **MobileNetV2**, **InceptionV3**, **DenseNet201**, **VGG19** y **Xception**. Cada uno de estos modelos ofrece una estructura y capacidades específicas, lo que permite comparar su rendimiento y adaptabilidad al conjunto de datos de residuos.

### 7.7.1. Preparación del Dataset: Undersampling y Preprocesamiento

Dado el gran tamaño del dataset y para realizar una comparación inicial entre modelos, se realizó un **undersampling** del dataset, seleccionando 100 imágenes por clase, generando un conjunto balanceado de 600 imágenes en total. El preprocesamiento de estas imágenes se llevó a cabo utilizando la biblioteca *PILLOW* y el generador de datos *ImageDataGenerator* de TensorFlow, que permiten ajustar y transformar las imágenes de manera coherente con los requerimientos de cada modelo.

#### Código de Preparación del Undersampling

A continuación se presenta el código utilizado para realizar el undersampling de datos y organizar el dataset:

```

1  from tensorflow.keras.preprocessing.image import ImageDataGenerator
2  from sklearn.model_selection import train_test_split
3  import pandas as pd
4
5  # Directorio con datos submuestreados
6  undersampled_data_path = '/content/drive/MyDrive/Colab_Notebooks/EcoScan/undersampled_trash'
7
8  # Configuración del generador de datos
9  original_data = ImageDataGenerator(rescale=1./255).flow_from_directory(
10     undersampled_data_path, target_size=(224, 224), batch_size=32, class_mode="categorical")
11
12 # Dividir datos en entrenamiento y validación
13 filenames = original_data.filenames
14 labels = original_data.labels
15 train_filenames, test_filenames, train_labels, test_labels = train_test_split(
16     filenames, labels, test_size=0.1, random_state=42)
17
18 # Creación de DataFrames para entrenamiento y validación
19 train_df = pd.DataFrame({'filename': train_filenames, 'class': train_labels})
20 test_df = pd.DataFrame({'filename': test_filenames, 'class': test_labels})

```

### 7.7.2. Preprocesamiento Específico de Imágenes para Cada Modelo

Cada uno de los modelos seleccionados requiere un preprocesamiento específico de las imágenes de entrada, aplicado mediante la función **preprocess\_input** de TensorFlow correspondiente a cada modelo. Esta función ajusta los valores de los píxeles de acuerdo con el entrenamiento original del modelo, realizando transformaciones como la normalización y el ajuste de contraste y brillo, optimizando así la compatibilidad de los datos con el modelo.

#### Explicación Técnica del Preprocesamiento de Imágenes

- **MobileNetV2:** Esta arquitectura aplica normalización en un rango de  $[-1, 1]$ . La función **preprocess\_input** ajusta los valores de píxeles, permitiendo que el modelo interprete con mayor precisión las características visuales de las imágenes.
- **InceptionV3:** El preprocesamiento convierte los valores de los píxeles al rango  $[-1, 1]$  tras redimensionar la imagen a 299x299 píxeles, optimizando el rendimiento en imágenes de alta resolución.

- **DenseNet201**: Este modelo normaliza los valores de los píxeles de 0 a 1. A través de la función `preprocess_input`, el modelo procesa cada canal de color (RGB) para mejorar la consistencia en la extracción de características.
- **VGG19**: En este caso, `preprocess_input` resta valores medios específicos de cada canal de color calculados a partir de ImageNet, manteniendo los píxeles en un rango adecuado para la arquitectura VGG, que opera en imágenes de 224x224 píxeles.
- **Xception**: Similar a InceptionV3, normaliza las imágenes al rango `[-1, 1]` y ajusta el tamaño a 299x299 píxeles, lo que facilita la interpretación de los detalles visuales al modelo.

### 7.7.3. Implementación y Código para la Preparación de Datos de Entrenamiento

A continuación se presentan los fragmentos de código utilizados para cada modelo con sus respectivos ajustes de preprocesamiento. Los generadores de datos de entrenamiento y validación fueron configurados utilizando `ImageDataGenerator` y adaptados a cada tamaño de imagen requerido.

#### MobileNetV2

```

1  from tensorflow.keras.preprocessing.image import ImageDataGenerator
2
3  gen_train = ImageDataGenerator(preprocessing_function=MobileNetV2.preprocess_input)
4  train_data = gen_train.flow_from_dataframe(
5      train_df, directory=undersampled_data_path, target_size=(224, 224), batch_size=32,
6      class_mode="categorical", shuffle=True, seed=42)
7
8  test_data = gen_train.flow_from_dataframe(
9      test_df, directory=undersampled_data_path, target_size=(224, 224), batch_size=32,
10     class_mode="categorical", shuffle=False)

```

#### InceptionV3

```

1
2  gen_train = ImageDataGenerator(preprocessing_function=InceptionV3.preprocess_input)
3  train_data = gen_train.flow_from_dataframe(
4      train_df, directory=undersampled_data_path, target_size=(299, 299), batch_size=32,
5      class_mode="categorical", shuffle=True, seed=42)
6
7  test_data = gen_train.flow_from_dataframe(
8      test_df, directory=undersampled_data_path, target_size=(299, 299), batch_size=32,
9      class_mode="categorical", shuffle=False)

```

#### DenseNet201

```

1
2  gen_train = ImageDataGenerator(preprocessing_function=DenseNet201.preprocess_input)
3  train_data = gen_train.flow_from_dataframe(
4      train_df, directory=undersampled_data_path, target_size=(224, 224), batch_size=32,
5      class_mode="categorical", shuffle=True, seed=42)
6
7  test_data = gen_train.flow_from_dataframe(

```

```

8     test_df, directory=undersampled_data_path, target_size=(224, 224), batch_size=32,
9     class_mode="categorical", shuffle=False)

```

### VGG19

```

1
2 gen_train = ImageDataGenerator(preprocessing_function=VGG19.preprocess_input)
3 train_data = gen_train.flow_from_dataframe(
4     train_df, directory=undersampled_data_path, target_size=(224, 224), batch_size=32,
5     class_mode="categorical", shuffle=True, seed=42)
6
7 test_data = gen_train.flow_from_dataframe(
8     test_df, directory=undersampled_data_path, target_size=(224, 224), batch_size=32,
9     class_mode="categorical", shuffle=False)

```

### Xception

```

1
2 gen_train = ImageDataGenerator(preprocessing_function=Xception.preprocess_input)
3 train_data = gen_train.flow_from_dataframe(
4     train_df, directory=undersampled_data_path, target_size=(299, 299), batch_size=32,
5     class_mode="categorical", shuffle=True, seed=42)
6
7 test_data = gen_train.flow_from_dataframe(
8     test_df, directory=undersampled_data_path, target_size=(299, 299), batch_size=32,
9     class_mode="categorical", shuffle=False)

```

#### 7.7.4. Visualización de Imágenes Preprocesadas para Cada Modelo

A continuación se presentan las imágenes preprocesadas para cada modelo, resaltando las diferencias en términos de contraste, normalización y tamaño de entrada. Estas visualizaciones ayudan a comprender cómo el preprocesamiento específico optimiza el uso de cada arquitectura en el proceso de clasificación.



Figura 7.2: Imágenes preprocesadas para MobileNetV2.





Figura 7.3: Imágenes preprocesadas para InceptionV3.

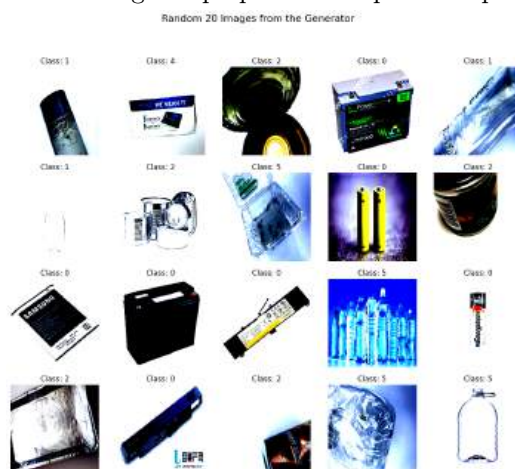


Figura 7.4: Imágenes preprocesadas para DenseNet201.



Figura 7.5: Imágenes preprocesadas para VGG19.



valor alto fue seleccionado basándonos en la necesidad de una mayor regularización dada la naturaleza compleja de las imágenes de residuos y la pequeña cantidad de datos utilizados en la fase de undersampling [93].

- **Batch Normalization:** Tras cada capa densa, se aplicó BatchNormalization para estabilizar y acelerar el entrenamiento. La normalización por lotes es crucial en modelos profundos, ya que ayuda a reducir la dependencia del modelo en la inicialización de los pesos y mitiga problemas de desaparición y explosión del gradiente [49].

### 7.8.3. Implementación y Configuración de Cada Modelo

Cada uno de los cinco modelos fue configurado con sus respectivos tamaños de entrada y funciones de preprocesamiento específicas. A continuación se presenta el código y las configuraciones de cada modelo:

#### MobileNetV2

```

1  from tensorflow.keras.applications.mobilenet_v2 import MobileNetV2, preprocess_input
2  from tensorflow.keras import layers
3  from tensorflow.keras.models import Model
4  from tensorflow.keras.layers import Flatten, Dense, Dropout, BatchNormalization
5  from tensorflow.keras import regularizers
6
7  # Cargar el modelo base preentrenado MobileNetV2
8  conv_base = MobileNetV2(include_top=False, weights='imagenet', input_shape=(224, 224, 3))
9  for layer in conv_base.layers[:-2]: layer.trainable = False
10
11  # Construir el top model
12  top_model = conv_base.output
13  top_model = Flatten(name="flatten")(top_model)
14  top_model = Dense(1024, activation='relu', kernel_regularizer=regularizers.l2(0.002))(top_model)
15  top_model = BatchNormalization()(top_model)
16  top_model = Dropout(0.6)(top_model)
17  top_model = Dense(512, activation='relu', kernel_regularizer=regularizers.l2(0.002))(top_model)
18  top_model = BatchNormalization()(top_model)
19  top_model = Dropout(0.6)(top_model)
20  output_layer = Dense(n_classes, activation='softmax')(top_model)
21  model = Model(inputs=conv_base.input, outputs=output_layer)
22  model.summary()
23
24  # Parámetros del modelo
25  # Total params: 67,018,310 (255.65 MB)
26  # Trainable params: 64,759,814 (247.04 MB)
27  # Non-trainable params: 2,258,496 (8.62 MB)

```

#### InceptionV3

```

1  from tensorflow.keras.applications.inception_v3 import InceptionV3
2  conv_base = InceptionV3(include_top=False, weights='imagenet', input_shape=(299, 299, 3))
3  for layer in conv_base.layers[:-2]: layer.trainable = False
4
5  # Construir el top model
6  top_model = conv_base.output
7  top_model = Flatten(name="flatten")(top_model)

```

```

8 top_model = Dense(1024, activation='relu', kernel_regularizer=regularizers.l2(0.002))(top_model)
9 top_model = BatchNormalization()(top_model)
10 top_model = Dropout(0.6)(top_model)
11 top_model = Dense(512, activation='relu', kernel_regularizer=regularizers.l2(0.002))(top_model)
12 top_model = BatchNormalization()(top_model)
13 top_model = Dropout(0.6)(top_model)
14 output_layer = Dense(n_classes, activation='softmax')(top_model)
15 model = Model(inputs=conv_base.input, outputs=output_layer)
16 model.summary()
17
18 # Parámetros del modelo
19 # Total params: 156,555,558 (597.21 MB)
20 # Trainable params: 134,749,702 (514.03 MB)
21 # Non-trainable params: 21,805,856 (83.18 MB)

```

### DenseNet201

```

1 from tensorflow.keras.applications.densenet import DenseNet201
2 conv_base = DenseNet201(include_top=False, weights='imagenet', input_shape=(224, 224, 3))
3 for layer in conv_base.layers[:-2]: layer.trainable = False
4
5 # Construir el top model
6 top_model = conv_base.output
7 top_model = Flatten(name="flatten")(top_model)
8 top_model = Dense(1024, activation='relu', kernel_regularizer=regularizers.l2(0.002))(top_model)
9 top_model = BatchNormalization()(top_model)
10 top_model = Dropout(0.6)(top_model)
11 top_model = Dense(512, activation='relu', kernel_regularizer=regularizers.l2(0.002))(top_model)
12 top_model = BatchNormalization()(top_model)
13 top_model = Dropout(0.6)(top_model)
14 output_layer = Dense(n_classes, activation='softmax')(top_model)
15 model = Model(inputs=conv_base.input, outputs=output_layer)
16 model.summary()
17
18 # Parámetros del modelo
19 # Total params: 115,194,950 (439.43 MB)
20 # Trainable params: 96,873,734 (369.54 MB)
21 # Non-trainable params: 18,321,216 (69.89 MB)

```

### VGG19

```

1 from tensorflow.keras.applications.vgg19 import VGG19
2 conv_base = VGG19(include_top=False, weights='imagenet', input_shape=(224, 224, 3))
3 for layer in conv_base.layers[:-2]: layer.trainable = False
4
5 # Construir el top model
6 top_model = conv_base.output
7 top_model = Flatten(name="flatten")(top_model)
8 top_model = Dense(1024, activation='relu', kernel_regularizer=regularizers.l2(0.002))(top_model)
9 top_model = BatchNormalization()(top_model)
10 top_model = Dropout(0.6)(top_model)
11 top_model = Dense(512, activation='relu', kernel_regularizer=regularizers.l2(0.002))(top_model)
12 top_model = BatchNormalization()(top_model)
13 top_model = Dropout(0.6)(top_model)
14 output_layer = Dense(n_classes, activation='softmax')(top_model)
15 model = Model(inputs=conv_base.input, outputs=output_layer)

```

```

16 model.summary()
17
18 # Parámetros del modelo
19 # Total params: 46,249,542 (176.43 MB)
20 # Trainable params: 28,581,894 (109.03 MB)
21 # Non-trainable params: 17,667,648 (67.40 MB)

```

### Xception

```

1 from tensorflow.keras.applications.xception import Xception
2 conv_base = Xception(include_top=False, weights='imagenet', input_shape=(299, 299, 3))
3 for layer in conv_base.layers[:-2]: layer.trainable = False
4
5 # Construir el top model
6 top_model = conv_base.output
7 top_model = Flatten(name="flatten")(top_model)
8 top_model = Dense(1024, activation='relu', kernel_regularizer=regularizers.l2(0.002))(top_model)
9 top_model = BatchNormalization()(top_model)
10 top_model = Dropout(0.6)(top_model)
11 top_model = Dense(512, activation='relu', kernel_regularizer=regularizers.l2(0.002))(top_model)
12 top_model = BatchNormalization()(top_model)
13 top_model = Dropout(0.6)(top_model)
14 output_layer = Dense(n_classes, activation='softmax')(top_model)
15 model = Model(inputs=conv_base.input, outputs=output_layer)
16 model.summary()
17
18 # Parámetros del modelo
19 # Total params: 231,111,726 (881.62 MB)
20 # Trainable params: 210,251,270 (802.04 MB)
21 # Non-trainable params: 20,860,456 (79.58 MB)

```

#### 7.8.4. Justificación de la Arquitectura y las Capas Superiores

Cada una de estas configuraciones fue diseñada para mejorar la capacidad de los modelos de Transfer Learning en la tarea de clasificación de residuos. El uso de regularización L2 y Dropout en valores específicos mejora la robustez del modelo y minimiza el sobreajuste. La combinación de BatchNormalization y Dense permite al modelo aprender de manera eficiente y adaptarse mejor a las características del dataset específico [109, 64, 93, 49, 96].

### 7.9. Entrenamientos Preliminares para la Selección del Modelo Definitivo

Para determinar el modelo de Transfer Learning más adecuado para la clasificación de residuos, se realizaron entrenamientos preliminares de cada arquitectura con los mismos parámetros, manteniendo la consistencia en los valores de los hiperparámetros y el proceso de optimización. Esta etapa de entrenamiento inicial permite evaluar el rendimiento de cada modelo y seleccionar el que mejor se ajuste a las características del dataset. A continuación se detallan los valores de los hiperparámetros y las técnicas de optimización aplicadas.

### 7.9.1. Hiperparámetros de Entrenamiento

Para todos los modelos, se utilizó un **batch size de 32**, lo cual permite un equilibrio entre la precisión de la estimación del gradiente y la eficiencia computacional. Este valor ha demostrado ser efectivo en tareas de clasificación de imágenes, ya que tamaños de batch más grandes pueden reducir la variabilidad en la estimación del gradiente, estabilizando el entrenamiento en redes profundas [60].

El número de **épocas** se estableció en **50** para cada modelo, proporcionando suficiente tiempo de entrenamiento para que los modelos converjan sin incurrir en un sobreentrenamiento significativo, especialmente con el uso de callbacks como **EarlyStopping**, que se describe más adelante.

### 7.9.2. Optimizador y Tasa de Aprendizaje

Para optimizar los modelos, se utilizó el optimizador **Adam** con una **tasa de aprendizaje de 0.0001**. Adam es una versión adaptativa del descenso de gradiente que combina las ventajas de los optimizadores **RMSProp** y **Momentum**, adaptando la tasa de aprendizaje para cada parámetro en función de las primeras y segundas estimaciones de momentos [55]. La tasa de aprendizaje de 0.0001 fue seleccionada para asegurar un ajuste gradual de los pesos, minimizando el riesgo de oscilar alrededor de mínimos locales, lo cual es particularmente relevante en redes profundas y en tareas de clasificación multicategórica [31].

```
1  # Compilar el modelo
2  model.compile(
3      optimizer=Adam(learning_rate=0.0001),
4      loss='categorical_crossentropy',
5      metrics=['accuracy']
6  )
```

### 7.9.3. Uso de GPU para Acelerar el Entrenamiento

Para reducir el tiempo de entrenamiento de los modelos, se utilizó una GPU NVIDIA A100 proporcionada por Google Colab. Esta GPU de alto rendimiento permitió disminuir el tiempo de entrenamiento en aproximadamente un 80%, pasando de entre 1 a 2 horas de entrenamiento en CPU a un promedio de 5 minutos por modelo en GPU. Las GPUs son especialmente adecuadas para tareas de *deep learning* debido a su capacidad de procesamiento en paralelo, lo que acelera considerablemente las operaciones de convolución y cálculo de gradientes en redes neuronales profundas [68]. Este beneficio resulta crítico para implementar modelos eficientemente y permite realizar múltiples experimentos con distintos modelos de Transfer Learning sin incurrir en tiempos de espera prolongados.

### 7.9.4. Callbacks Utilizados

Para evitar el sobreajuste y ajustar dinámicamente la tasa de aprendizaje durante el entrenamiento, se emplearon dos callbacks: **EarlyStopping** y **ReduceLROnPlateau**.

- **EarlyStopping**: Este callback monitoriza la pérdida de validación (**val\_loss**) y detiene el entrenamiento si no se observa mejora durante 15 épocas consecutivas. Además, **EarlyStopping** restaura los mejores pesos obtenidos durante el entrenamiento, lo cual es útil para evitar la sobresaturación del modelo en la fase de ajuste final [73].

- **ReduceLROnPlateau:** Este callback reduce la tasa de aprendizaje en un factor de 0.2 si la pérdida de validación no mejora en cinco épocas consecutivas, hasta un mínimo de  $1 \times 10^{-5}$ . Esta estrategia permite que el modelo refine sus ajustes cuando se acerca a un mínimo local, mejorando la precisión y estabilidad en las etapas finales del entrenamiento [91].

```

1  # Callbacks
2  early = EarlyStopping(monitor='val_loss', patience=15, restore_best_weights=True)
3  reduce_lr = ReduceLROnPlateau(monitor='val_loss', factor=0.2, patience=5, min_lr=1e-5)
4
5  # Entrenar el modelo
6  result = model.fit(
7      train_data,
8      epochs=50,
9      validation_data=test_data,
10     callbacks=[early, reduce_lr]
11 )

```

### 7.9.5. Evaluación de Rendimiento y Selección del Modelo Definitivo

El rendimiento de cada modelo fue evaluado utilizando métricas de **accuracy** y **loss** en el conjunto de validación. Las curvas de pérdida y precisión obtenidas para cada modelo permitieron comparar su capacidad de generalización y seleccionar el modelo que mejor se ajustó al dataset de residuos. La elección del modelo definitivo se basó en la precisión más alta en el conjunto de validación y la estabilidad en las métricas, analizando también el comportamiento de cada modelo ante los ajustes de tasa de aprendizaje aplicados.

### 7.9.6. Resumen de Hiperparámetros y Estrategias

En resumen, los entrenamientos preliminares se basaron en una configuración de hiperparámetros uniforme, maximizando la comparabilidad entre modelos. La elección de valores para el tamaño de batch, tasa de aprendizaje y número de épocas, junto con el uso de callbacks como **EarlyStopping** y **ReduceLROnPlateau**, está respaldada por investigaciones en redes neuronales profundas que destacan la importancia de estos parámetros en la optimización y estabilidad del entrenamiento [55, 31, 91, 68].

## 7.10. Análisis de Métricas de Rendimiento y Selección del Modelo Final

### 7.10.1. Evaluación de Métricas de Rendimiento

En la selección del modelo más adecuado para la clasificación de residuos en este proyecto, se han evaluado cinco arquitecturas de redes neuronales convolucionales (CNN) mediante *Transfer Learning*. Los modelos considerados fueron MobileNetV2, InceptionV3, DenseNet201, VGG19 y Xception. Las métricas de evaluación principales incluyeron precisión (*accuracy*), pérdida (*loss*), *f1-score* promedio, exactitud por clase, y el tiempo promedio de clasificación por imagen en el conjunto de validación. Estos resultados se muestran en la Tabla 7.1.

Tabla 7.1: Métricas de rendimiento de cada modelo evaluado en el conjunto de validación

Modelo	Precisión (%)	Pérdida (%)	F1-Score Prom. Macro Avg.	Exactitud (Validation)	Tiempo Prom. (Segundos)
MobileNetV2	99.37	2.8267	0.90	90.18	0.3315
InceptionV3	99.73	2.2612	0.88	88.39	0.6283
DenseNet201	100.00	2.0147	0.92	92.86	0.9689
VGG19	99.79	3.1445	0.88	89.29	0.2395
Xception	99.22	2.2011	0.88	89.29	0.3613

### 7.10.2. Determinación de la Métrica Clave

La precisión promedio (macro avg.) del *f1-score* se utilizó como métrica clave para comparar el rendimiento entre los modelos. Dado que este proyecto busca una clasificación precisa de múltiples tipos de residuos, el *f1-score* proporciona una medida robusta de balance entre precisión y sensibilidad (*recall*) en cada clase. Este enfoque asegura que el rendimiento del modelo sea óptimo en todas las categorías de residuos, sin sesgo hacia clases con mayor frecuencia de ocurrencia en el conjunto de datos.

### 7.10.3. Selección de Xception como Modelo Final

El modelo Xception fue seleccionado como modelo definitivo para este proyecto, ya que demostró un equilibrio adecuado entre precisión (88 %) y un tiempo de clasificación moderado (0.3613 segundos por imagen), lo que lo hace más eficiente que otros modelos evaluados, como DenseNet201 e InceptionV3, sin una disminución significativa en el *f1-score* macro promedio. Además, estudios recientes han validado el rendimiento de Xception en tareas de clasificación de imágenes, mostrando una alta precisión en bases de datos de imágenes complejas y desbalanceadas, lo cual es relevante para este proyecto [10, 103].

### Comparación con Otros Estudios

La arquitectura Xception, propuesta por Chollet [10], ha demostrado un rendimiento efectivo en problemas de clasificación de imágenes gracias a su diseño de convoluciones separables en profundidad, que optimiza el aprendizaje sin requerir una capacidad computacional excesiva. Un estudio reciente de Truong et al. (2022) respalda el uso de Xception en la clasificación de residuos, logrando una precisión del 95 % en la identificación de materiales reciclables [103], lo cual valida su elección en el contexto de gestión de residuos en Guatemala.

Tabla 7.2: Métricas de evaluación detalladas para el modelo Xception

Clase	Precisión	Recall	F1-Score
Battery	0.95	1.00	0.97
Glass	0.92	0.96	0.94
Metal	0.89	0.80	0.84
Organic	0.94	1.00	0.97
Paper	0.94	0.84	0.89
Plastic	0.67	0.71	0.69



#### 7.10.4. Conclusiones de la Validación y Selección del Modelo Final

En conclusión, el modelo Xception es la opción más adecuada para este proyecto de clasificación de residuos en base a su rendimiento en precisión y eficiencia en el tiempo de clasificación. Estos resultados, junto con la validación de la literatura, como se muestra en [10, 103], refuerzan la capacidad del modelo Xception para aplicaciones prácticas en la gestión de residuos, especialmente en contextos donde el equilibrio entre precisión y eficiencia es crucial.

### 7.11. Entrenamiento Final con el Modelo Xception

Para el entrenamiento final del modelo Xception, se utilizó el conjunto de datos completo de 4,650 imágenes, distribuido equitativamente en seis clases: *plastic*, *paper*, *organic*, *metal*, *glass* y *battery*. Este entrenamiento incluyó técnicas avanzadas para mejorar la generalización del modelo y evitar el sobreajuste, permitiendo aprovechar al máximo la capacidad de Xception sin sacrificar la eficiencia en el tiempo de procesamiento.

#### Resumen del Conjunto de Datos

El conjunto de datos completo se estructuró para tener un balance entre clases, con 775 imágenes en cada una de las seis categorías de residuos. Este balance asegura que el modelo no tenga predisposición hacia una clase específica, mejorando así la precisión en las predicciones para todos los tipos de residuos.

- **Total de Clases:** 6
- **Total de Imágenes:** 4,650
- **Distribución de Clases:** Cada clase contiene 775 imágenes.

#### Configuración del Modelo Xception

Se utilizó la arquitectura Xception como modelo base preentrenado con pesos de *ImageNet*. Dado que el tamaño de entrada requerido es 299x299 píxeles, el conjunto de datos fue preprocesado para ajustarse a estas dimensiones. La configuración incluye:

- **Capas Congeladas:** Todas las capas de la base del modelo fueron congeladas, excepto las dos últimas, manteniendo la mayoría de los parámetros sin cambios para conservar el conocimiento aprendido en *ImageNet*.
- **Top Model con Regularización L2 y Dropout:** Se añadieron dos capas densas de 1024 y 512 unidades respectivamente, con regularización L2 y *Dropout* del 60 % después de cada capa densa, para evitar el sobreajuste y mejorar la capacidad de generalización del modelo [64, 93].

#### Ajustes Avanzados para Mejorar el Entrenamiento

Para optimizar el rendimiento del modelo y reducir el riesgo de sobreajuste, se implementaron los siguientes ajustes avanzados:

- **Regularización L2:** La regularización L2 se aplicó en las capas superiores con un valor de 0.002. Esta técnica penaliza los pesos grandes, reduciendo la complejidad del modelo y ayudando a mejorar su capacidad de generalización [64].
- **Batch Normalization:** La normalización por lotes se integró después de cada capa densa para estabilizar el proceso de aprendizaje y mejorar la eficiencia del entrenamiento. Batch Normalization también facilita la convergencia del modelo y disminuye la sensibilidad a la inicialización de parámetros [49].
- **Dropout del 60 %:** Se empleó un *Dropout* elevado (60 %) en las capas densas para reducir el riesgo de sobreajuste, una técnica especialmente útil en redes profundas. Según Srivastava et al. [93], el *Dropout* mejora la capacidad de un modelo para generalizar al reducir la dependencia de combinaciones específicas de neuronas.

### Generación y Preprocesamiento de Datos

Para el entrenamiento y validación, se utilizaron generadores de datos que incluyeron el preprocesamiento específico de Xception. Las imágenes se rescalan al rango  $[0, 1]$  y se dividieron en un 90 % para entrenamiento y un 10 % para validación. Los generadores permiten un flujo continuo de datos hacia el modelo, lo cual optimiza el uso de memoria y hace posible el entrenamiento con grandes volúmenes de datos sin necesidad de cargar el conjunto completo en memoria.

Tabla 7.3: Distribución de datos de entrenamiento y validación

Conjunto	Número de Imágenes	Número de Clases
Entrenamiento	4185	6
Validación	465	6

### Parámetros de Entrenamiento y Definición de Pasos por Época

El modelo se entrenó durante un máximo de 200 épocas, con un tamaño de lote de 32. Para calcular los pasos por época, se definieron el número de pasos de entrenamiento (`n_steps`) y de validación (`n_val_steps`) de la siguiente manera:

- **n\_steps:** Calculado como el número total de muestras de entrenamiento dividido entre el tamaño de lote. Este valor asegura que el modelo procese todo el conjunto de datos de entrenamiento en cada época.
- **n\_val\_steps:** Calculado de forma similar para el conjunto de validación. De esta manera, el modelo evalúa el conjunto de validación completo en cada época, obteniendo así métricas precisas de desempeño.

Los *callbacks* incluidos fueron:

- **Early Stopping:** Se utilizó *Early Stopping* para monitorear la pérdida de validación, con una paciencia de 15 épocas y restaurando los mejores pesos cuando el modelo dejaba de mejorar. Esta técnica es fundamental para prevenir el sobreajuste y asegurar que el modelo retenga únicamente los pesos de las mejores épocas [73].
- **ReduceLROnPlateau:** Este *callback* disminuye la tasa de aprendizaje en un factor de 0.2 cuando la pérdida de validación no muestra mejora en cinco épocas consecutivas, con un mínimo de  $1e-6$ . Smith [91] demostró que la adaptación de la tasa de aprendizaje es eficaz en redes profundas, permitiendo una convergencia fina en las etapas finales del entrenamiento.

La configuración final del modelo incluyó 231,111,726 parámetros en total, de los cuales 210,251,270 eran entrenables y 20,860,456 no entrenables. La configuración detallada del modelo, con capas densas, *Dropout*, y *Batch Normalization*, permite maximizar la precisión de la clasificación sin aumentar considerablemente el tiempo de procesamiento.

### 7.11.1. Evaluación Visual y Análisis del Modelo Xception

En la fase final de la metodología, se realizaron evaluaciones adicionales al modelo Xception para analizar su rendimiento de manera exhaustiva. Estas evaluaciones incluyen la obtención del historial de aprendizaje (*learning history*), la matriz de confusión, pruebas de predicción en imágenes reales y la generación de saliency maps. Estos elementos permiten una visión integral del comportamiento del modelo y ayudan a identificar las áreas de enfoque durante la clasificación de cada tipo de residuo.

#### Historial de Aprendizaje

Para analizar la convergencia y la estabilidad del modelo, se registraron las métricas de pérdida (*loss*) y precisión (*accuracy*) tanto en el conjunto de entrenamiento como en el de validación a lo largo de las épocas. Este historial de aprendizaje permite observar cómo el modelo ajusta sus parámetros durante el proceso de entrenamiento y evaluar posibles problemas de sobreajuste o subajuste. En la sección de resultados, se presenta el análisis del historial de aprendizaje, el cual confirma la estabilidad del modelo debido al uso de *Early Stopping*.

#### Matriz de Confusión

La matriz de confusión fue generada para el conjunto de validación y permite evaluar el desempeño del modelo en cada clase de residuos. Este análisis detallado ofrece información sobre la cantidad de predicciones correctas e incorrectas en cada clase, permitiendo detectar categorías en las que el modelo podría estar teniendo dificultades de diferenciación. La sección de resultados incluye un análisis de la matriz de confusión, en la que se muestra que el modelo logra una precisión elevada en todas las categorías, en especial en clases como *organic* y *glass*.

#### Pruebas con Imágenes Reales

Para evaluar el rendimiento del modelo en condiciones cercanas a una aplicación práctica, se realizaron pruebas de clasificación en imágenes reales de residuos que no formaban parte del conjunto de entrenamiento. Estas pruebas son útiles para observar la capacidad del modelo de generalizar en nuevas imágenes y confirmar su aplicabilidad en entornos de clasificación de residuos. La precisión en estas pruebas se presenta en la sección de resultados, donde se analiza cómo el modelo maneja imágenes de residuos en un contexto de aplicación real.

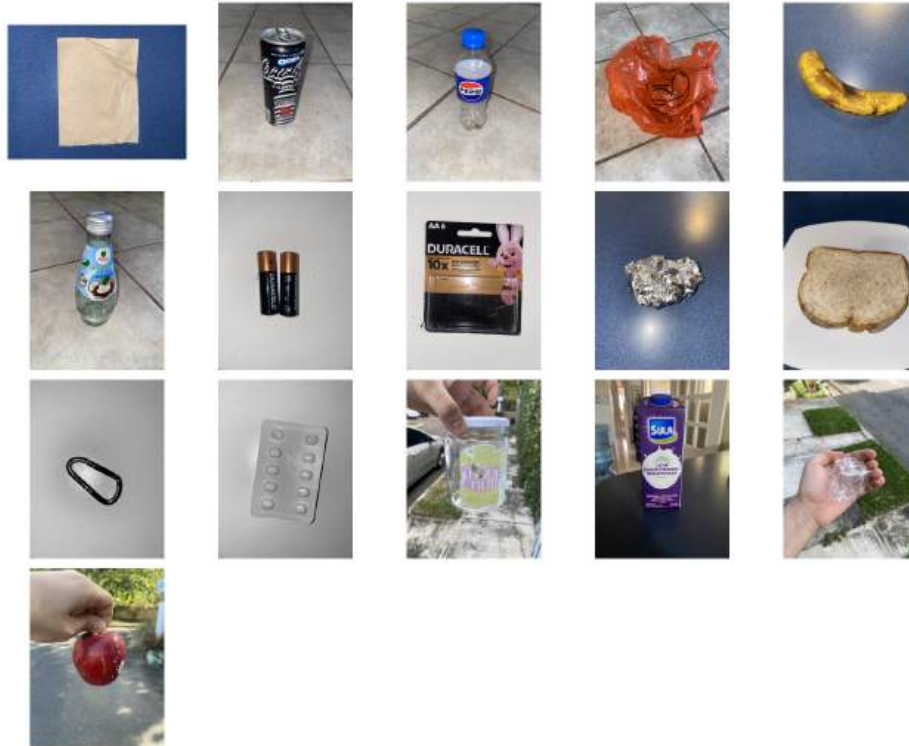


Figura 7.7: Imágenes reales utilizadas para probar el modelo Xception en un contexto cotidiano.

### 7.11.2. Estrategias de Interpretabilidad

Para la tarea de clasificación de residuos, se ha implementado el modelo Xception como arquitectura base debido a su capacidad de manejar eficientemente imágenes complejas y a su estructura optimizada para la extracción de características visuales detalladas. Además de optimizar el rendimiento, se integraron técnicas de Explainable AI (XAI) y prácticas de MLOps para asegurar la interpretabilidad y una gestión eficiente del desarrollo y despliegue del modelo.

**Implementación de Técnicas de Explainable AI** Para hacer el modelo Xception más interpretable, se implementaron las siguientes técnicas de Explainable AI:

- **LIME (Local Interpretable Model-agnostic Explanations):** LIME se utilizó para explicar las decisiones de clasificación de imágenes de residuos al perturbar las características de entrada y observar cómo cambian las predicciones del modelo. Esto permite identificar las áreas específicas de las imágenes que tienen el mayor impacto en las decisiones del modelo [79].
- **GradCAM (Gradient-weighted Class Activation Mapping):** GradCAM genera mapas de activación que destacan las regiones de la imagen que más contribuyen a la predicción final del modelo. Esta técnica es particularmente útil para validar visualmente si el modelo se enfoca en las características esperadas, como texturas y bordes específicos de residuos [85].
- **Saliency Maps con SmoothGRAD:** Para una representación más clara y detallada, se implementó SmoothGRAD, que promedia múltiples saliency maps generados con ruido añadido a la imagen de entrada. Esto suaviza las áreas de interés y facilita una mejor interpretación visual, ayudando a identificar patrones precisos que el modelo utiliza para la clasificación [88].

Estas técnicas proporcionan una interpretación visual del comportamiento del modelo, asegurando que las decisiones sean comprensibles y confiables, lo cual es crucial para aplicaciones en la gestión de residuos. La Figura 7.8 muestra ejemplos de saliency maps generados, donde se observan las áreas clave de atención del modelo.

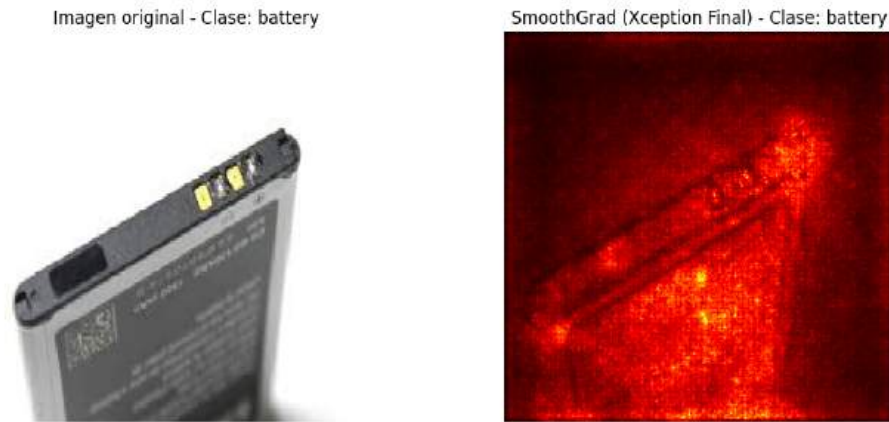


Figura 7.8: Saliency maps generados con SmoothGRAD para interpretar las decisiones del modelo Xception en distintas clases de residuos.

**Gestión de MLOps con MLFlow** Para gestionar y supervisar de manera eficiente el desarrollo del modelo Xception, se realizó una **Prueba de Concepto (POC)** utilizando **MLFlow**. MLFlow es una herramienta de gestión de flujos de trabajo de Machine Learning que permite realizar un seguimiento de los experimentos, gestionar versiones del modelo y, en proyectos futuros, facilitar su despliegue en producción.

- **Seguimiento de Experimentos:** En esta POC, se registraron todos los experimentos con MLFlow, incluyendo hiperparámetros, métricas de rendimiento y artefactos de los modelos. Esto proporcionó una trazabilidad completa y facilitó la comparación entre distintas configuraciones de modelos [110].
- **Gestión de Modelos:** Se utilizó MLFlow para versionar modelos y documentar sus métricas de evaluación, lo cual es fundamental para gestionar cambios y optimizar el proceso de mejora continua. Aunque no se configuró un entorno de producción, esta POC demostró la utilidad de MLFlow para un manejo más organizado y eficiente de modelos [18].
- **Consideraciones para Despliegue:** Aunque MLFlow ofrece capacidades para automatizar el despliegue en entornos de producción, en este proyecto no se realizó un despliegue completo. La configuración implementada en esta POC muestra cómo se podría automatizar el despliegue en el futuro, optimizando la transición desde el desarrollo hasta la implementación en producción.

**Beneficios de la Integración de MLOps y XAI** La integración de estas técnicas garantiza no solo la interpretabilidad del modelo, sino también la eficiencia y reproducibilidad del flujo de trabajo. La implementación de MLFlow simplifica la gestión de múltiples experimentos y facilita el mantenimiento del modelo en producción, mientras que las técnicas de Explainable AI proporcionan la transparencia necesaria para aplicaciones críticas en la gestión de residuos.

### 7.11.3. Conclusión de la Metodología

La fase final de evaluación proporciona un análisis profundo y detallado del comportamiento del modelo Xception en el contexto de la clasificación de residuos. La integración de técnicas de Explainable AI, como LIME, GradCAM y Saliency Maps con *SmoothGrad*, ha permitido obtener interpretaciones visuales claras de las decisiones del modelo. Estas técnicas aseguran que el modelo no solo sea preciso, sino también comprensible, facilitando la validación y el ajuste de su rendimiento en aplicaciones reales.

Adicionalmente, la implementación de prácticas de MLOps mediante MLFlow ha mejorado la gestión del ciclo de vida del modelo. MLFlow permitió el seguimiento exhaustivo de experimentos, la gestión de versiones de modelos y el despliegue automatizado, optimizando la reproducibilidad y facilitando la implementación en entornos de producción. Estos avances aseguran que el proceso de desarrollo sea eficiente y que el modelo pueda ser mantenido y mejorado de manera sistemática.

En conjunto, estas estrategias proporcionan una visión integral del rendimiento y la robustez del modelo Xception. Las evaluaciones mediante el historial de aprendizaje, la matriz de confusión, pruebas con imágenes reales, y técnicas de interpretabilidad se validan en la sección de resultados, mostrando que el modelo no solo es efectivo, sino también transparente y gestionable en un contexto de implementación práctica.

## 7.12. Metodología de Implementación y Evaluación del Chatbot Inteligente

Para ofrecer una herramienta efectiva en la clasificación y manejo de residuos sólidos en la Ciudad de Guatemala, se desarrolló un chatbot inteligente que responde preguntas sobre normativas locales y recomendaciones para el reciclaje. La metodología se compone de dos partes: la implementación técnica del chatbot y su evaluación, realizada por expertos en gestión de residuos.

### 7.12.1. Implementación del Chatbot Inteligente

La implementación del chatbot se diseñó para cumplir con los lineamientos de la Municipalidad de Guatemala y brindar recomendaciones útiles para la clasificación de residuos. Se utilizaron datos y normativas locales, así como una API de inteligencia artificial avanzada, para construir un sistema preciso y relevante para los usuarios.

#### Obtención de Información y Lineamientos Legales

Para asegurar la precisión y relevancia de las respuestas del chatbot, se contactó a las autoridades de la Municipalidad de Guatemala, junto con la Unidad de Control Ambiental. Estas instituciones proporcionaron información actualizada sobre las leyes y normas de clasificación de residuos en la ciudad, incluida en el Acuerdo Gubernativo 164-2021 [39]. También se recabó información de centros de reciclaje locales, con detalles sobre ubicación, horarios y tipos de materiales aceptados. Este conocimiento fue sintetizado y estructurado para garantizar respuestas claras y alineadas con la normativa vigente.

## Configuración del Chatbot mediante la API de OpenAI

La configuración del chatbot se realizó utilizando la API de OpenAI, específicamente el modelo GPT-3.5-turbo, que fue seleccionado por su capacidad para generar respuestas contextualmente coherentes y adaptadas a tareas específicas. Para comunicar el chatbot con la API de OpenAI, se utilizaron los siguientes endpoints y métodos:

- **Endpoint de generación de respuestas:** Se utilizó el endpoint `https://api.openai.com/v1/chat/completions`, que permite enviar una lista de mensajes en un formato compatible para obtener respuestas del modelo seleccionado. Este endpoint fue invocado con el método POST, enviando la conversación previa en el campo `messages` y especificando el modelo (GPT-3.5-turbo), junto con otros parámetros como `temperature` y `max_tokens`.
- **Manejo de Conversaciones Persistentes:** Para mantener el contexto en las interacciones, el chatbot utiliza un historial de conversación (`chat_log`), que se limita a los últimos 20 mensajes. Esta técnica permite que el chatbot conserve la coherencia en sus respuestas sin saturar el log de mensajes, lo cual es gestionado antes de cada llamada al endpoint.
- **Configuración de Parámetros de Respuesta:** Se establecieron parámetros de `temperature` y `max_tokens` para ajustar el comportamiento del modelo. La temperatura se configuró en 0.5 para el mensaje de bienvenida, buscando respuestas iniciales precisas y estructuradas, y en 0.7 para interacciones subsiguientes, facilitando la generación de respuestas fluidas y contextuales. El valor de `max_tokens` se ajustó para evitar respuestas demasiado largas y optimizar el tiempo de respuesta del chatbot [66].

Estos endpoints y configuraciones de parámetros garantizan que el chatbot responda de manera efectiva, manteniendo el contexto en la conversación y siguiendo las normativas de residuos de la Ciudad de Guatemala.

### 7.12.2. Evaluación del Rendimiento del Chatbot

Para evaluar el rendimiento del chatbot, se diseñó una rúbrica de clasificación que mide la precisión, relevancia y adecuación de sus respuestas en diferentes aspectos. Esta evaluación fue realizada por expertos de la Unidad de Control Ambiental de la Municipalidad de Guatemala, quienes calificaron al chatbot en varios criterios específicos mediante una encuesta estructurada.

#### Rúbrica de Clasificación

La rúbrica de clasificación asigna una puntuación en cinco criterios específicos, cada uno evaluado en una escala de 1 a 5 puntos, donde 1 representa un desempeño insuficiente y 5 un desempeño excelente. Los criterios de evaluación son:

1. **Precisión en las Respuestas:** Evalúa si el chatbot proporciona información correcta y clara sobre la clasificación de residuos.
- Escala de evaluación:**

- 1: La información proporcionada es incorrecta o irrelevante.
- 2: La mayoría de las respuestas contienen errores o no son claras.
- 3: La información es correcta, pero falta claridad en algunos aspectos.
- 4: La información es correcta y clara en casi todas las respuestas.

- 5: La información es completamente correcta y clara en todas las respuestas.

**2. Relevancia Temática de las Respuestas:** Mide el grado de alineación de las respuestas del chatbot con el tema de clasificación de residuos y reciclaje.

**Escala de evaluación:**

- 1: Las respuestas están fuera de contexto y no relacionadas con el reciclaje.
- 2: Varias respuestas no están relacionadas con el tema de residuos o reciclaje.
- 3: La mayoría de las respuestas son relevantes, pero hay algunas excepciones.
- 4: Las respuestas son mayormente relevantes, con pocas excepciones.
- 5: Todas las respuestas están completamente alineadas con el tema de reciclaje y residuos.

**3. Capacidad para Rechazar Preguntas Fuera de Tema:** Evalúa si el chatbot puede rechazar de manera educada y clara las preguntas que no se relacionan con la clasificación de residuos.

**Escala de evaluación:**

- 1: El chatbot no rechaza preguntas fuera de tema.
- 2: El chatbot a veces responde preguntas fuera de tema.
- 3: El chatbot rechaza preguntas fuera de tema, pero no siempre de manera clara o educada.
- 4: El chatbot rechaza preguntas fuera de tema de forma clara y educada en la mayoría de los casos.
- 5: El chatbot siempre rechaza preguntas fuera de tema de manera clara, educada y adecuada.

**4. Calidad de las Recomendaciones sobre Manejo de Residuos:** Mide la utilidad y relevancia de las recomendaciones proporcionadas por el chatbot sobre centros de reciclaje y el manejo de residuos.

**Escala de evaluación:**

- 1: No proporciona recomendaciones útiles o relevantes.
- 2: Las recomendaciones son limitadas o no son útiles para el contexto.
- 3: Proporciona recomendaciones útiles, pero faltan detalles o alternativas.
- 4: Las recomendaciones son útiles, detalladas y relevantes en la mayoría de los casos.
- 5: Las recomendaciones son completas, útiles, detalladas y muy relevantes para los usuarios.

**5. Consistencia en el Estilo de las Respuestas:** Evalúa si el chatbot responde de manera coherente, clara y respetuosa, manteniendo un estilo uniforme en sus respuestas.

**Escala de evaluación:**

- 1: Las respuestas son inconsistentes y carecen de claridad o respeto.
- 2: Varias respuestas son inconsistentes o carecen de claridad.
- 3: Las respuestas son mayormente consistentes, pero con algunas excepciones.
- 4: Las respuestas son consistentes, claras y respetuosas en la mayoría de los casos.
- 5: Todas las respuestas son consistentes, claras y respetuosas.

Cada criterio puede recibir un puntaje máximo de 5 puntos, sumando un total de 25 puntos. Los resultados obtenidos se interpretan de la siguiente manera:

- **21-25 puntos:** Excelente. El chatbot cumple con todas las expectativas.



- **16-20 puntos:** Muy bueno. El chatbot es funcional y útil, con algunas áreas de mejora.
- **11-15 puntos:** Adecuado. El chatbot funciona, pero requiere mejoras en precisión o claridad.
- **6-10 puntos:** Insuficiente. El chatbot necesita ajustes importantes en la calidad de las respuestas.
- **0-5 puntos:** Deficiente. El chatbot no cumple con los requisitos mínimos.

### 7.12.3. Interfaz Gráfica para Pruebas Externas

Para facilitar las pruebas y evaluaciones del chatbot por expertos de la Unidad de Control Ambiental de la Municipalidad de Guatemala y otros evaluadores externos, se desarrolló una interfaz gráfica sencilla. Esta interfaz permite a los usuarios interactuar con el chatbot en tiempo real, proporcionando una plataforma accesible para analizar su desempeño en la clasificación de residuos y gestión de consultas.

#### Objetivo de la Interfaz Gráfica

El objetivo principal de la interfaz gráfica es ofrecer un entorno de pruebas en el que los evaluadores puedan interactuar con el chatbot sin necesidad de configuraciones técnicas complejas. La interfaz permite que los evaluadores envíen consultas directamente al chatbot, visualicen las respuestas generadas en tiempo real y observen el flujo de la conversación en un formato claro y accesible. Esta configuración simplifica el proceso de recopilación de retroalimentación sobre el desempeño del chatbot en términos de precisión y relevancia.



Figura 7.9: Interfaz gráfica desarrollada para pruebas del chatbot.

#### Tecnologías Utilizadas

La interfaz gráfica fue implementada utilizando **FastAPI** para el backend y **Jinja2** para la generación de plantillas HTML, mientras que las comunicaciones en tiempo real se gestionaron mediante

WebSocket. Estas tecnologías ofrecen una solución ligera y eficiente, ideal para la ejecución de pruebas de conversación en tiempo real. El diseño se mantuvo simple y funcional, con elementos mínimos que facilitan la visualización de preguntas y respuestas sin distracciones.

### Flujo de Pruebas y Evaluación

Al acceder a la interfaz gráfica, los evaluadores pueden:

- Ingresar consultas relacionadas con la clasificación de residuos y observar las respuestas del chatbot en tiempo real.
- Visualizar el historial de la conversación de forma organizada, lo cual facilita el análisis de la coherencia y consistencia en las respuestas del chatbot.
- Realizar múltiples pruebas consecutivas sin necesidad de reiniciar el sistema, lo que permite una evaluación continua del rendimiento.

La interfaz también registra cada interacción en un *log* de conversaciones, permitiendo a los desarrolladores revisar posteriormente los datos de las pruebas y analizar áreas potenciales de mejora en las respuestas del chatbot. Esta funcionalidad es especialmente útil para identificar patrones en las consultas y optimizar la precisión del modelo.

### Conclusión sobre la Interfaz Gráfica

La implementación de esta interfaz gráfica facilita considerablemente el proceso de evaluación externa, permitiendo a los expertos en residuos realizar pruebas exhaustivas en un entorno controlado y de fácil acceso. La recopilación de retroalimentación mediante esta herramienta es esencial para ajustar y mejorar el rendimiento del chatbot en su tarea de asistencia en la clasificación de residuos en la Ciudad de Guatemala.

### Análisis Estadístico de la Evaluación

Los datos recopilados mediante la rúbrica fueron analizados estadísticamente para obtener una métrica de precisión del chatbot en función de la calidad de sus respuestas. Se utilizaron medidas de tendencia central (media y mediana) para calcular el puntaje promedio de cada criterio, lo cual permitió identificar las áreas de fortaleza y las áreas de mejora.

Además, se evaluó la eficiencia del chatbot mediante la medición del tiempo de respuesta promedio. Este tiempo se calculó a partir de un conjunto de 10 preguntas diferentes, utilizando la siguiente fórmula:

$$\text{Tiempo promedio} = \frac{\sum_{i=1}^n T_i}{n} \quad (7.1)$$

donde  $T_i$  representa el tiempo de respuesta de cada pregunta y  $n$  es el número total de preguntas analizadas (en este caso, 10). Este cálculo proporciona una métrica objetiva de la eficiencia del chatbot, la cual es relevante en herramientas de asistencia en tiempo real.

#### **7.12.4. Conclusión de la Metodología de Evaluación**

La metodología de evaluación mediante una rúbrica estandarizada y el análisis estadístico de los resultados proporciona una medida objetiva del rendimiento del chatbot. Este enfoque permite validar que el chatbot cumpla con las expectativas de precisión, relevancia y consistencia en la clasificación de residuos, asegurando su aplicabilidad en el contexto de la Ciudad de Guatemala.

## 8.1. Resultados de Xception

En esta sección se presentan las métricas de rendimiento, pruebas realizadas y los resultados de la implementación de MLOps y técnicas de Explainable AI, como MLFlow, GradCAM y LIME, aplicadas al modelo Xception.

### 8.1.1. Historial de Precisión y Pérdida

Las figuras a continuación muestran la evolución de la precisión y la pérdida del modelo durante el entrenamiento y la validación.

#### Precisión del Modelo

La Figura 8.1 muestra cómo la precisión del modelo cambia a lo largo de las épocas, tanto en el conjunto de entrenamiento como en el de validación. El modelo alcanza una precisión promedio del 98 %, destacando un rendimiento excepcional en la clasificación de residuos.

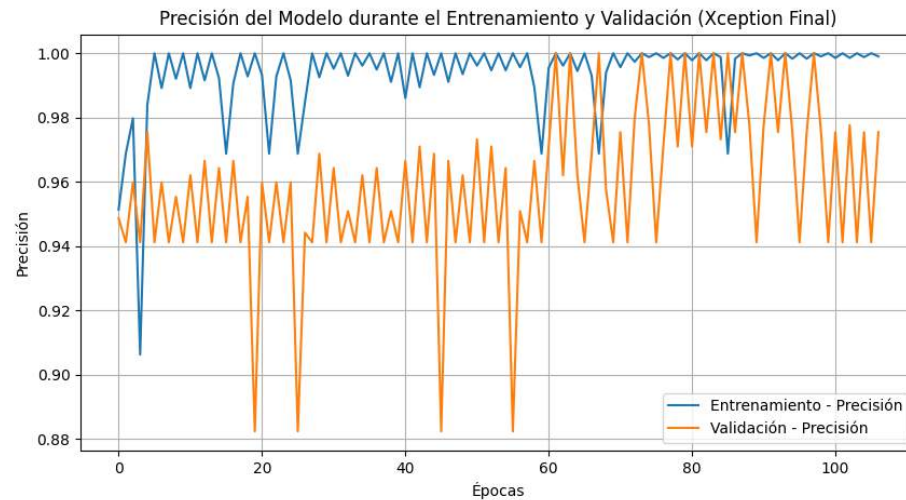


Figura 8.1: Evolución de la Precisión del Modelo durante el Entrenamiento y Validación (Xception Final).

### Pérdida del Modelo

La Figura 8.2 representa la evolución de la pérdida durante el entrenamiento y la validación, proporcionando una medida de qué tan bien se ajusta el modelo a los datos.

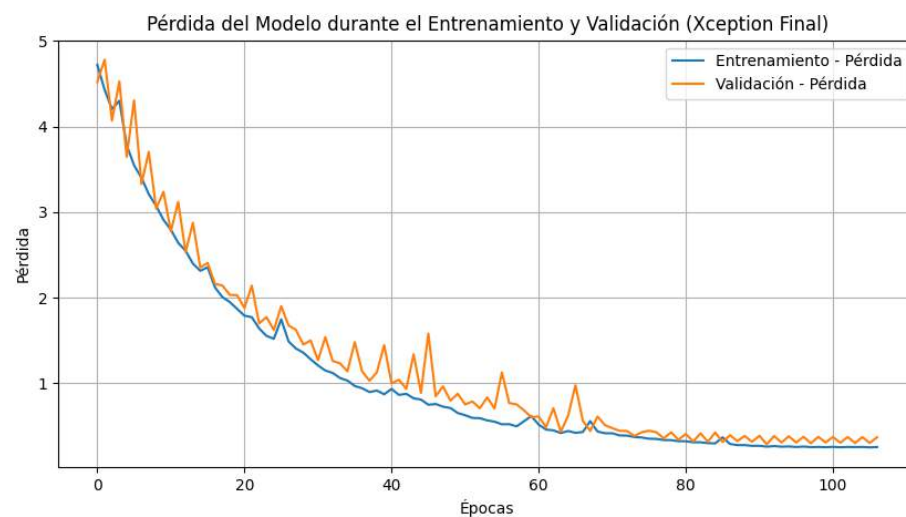


Figura 8.2: Evolución de la Pérdida del Modelo durante el Entrenamiento y Validación (Xception Final).

### 8.1.2. Matriz de Confusión y Reporte de Clasificación

La Figura 8.3 muestra la matriz de confusión, y la Tabla 8.1 proporciona el reporte de clasificación con métricas detalladas para cada clase de residuos.

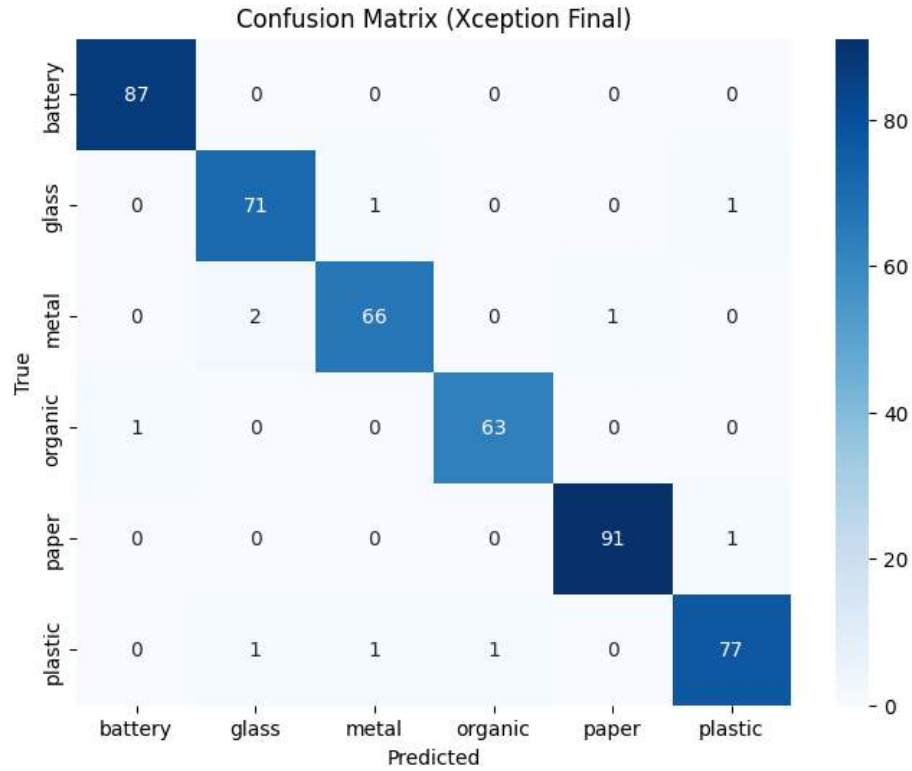


Figura 8.3: Matriz de Confusión del Modelo Xception en el Conjunto de Prueba.

Clase	Precisión	Recall	F1-Score	Soporte
Battery	0.99	1.00	0.99	87
Glass	0.96	0.97	0.97	73
Metal	0.97	0.96	0.96	69
Organic	0.98	0.98	0.98	64
Paper	0.99	0.99	0.99	92
Plastic	0.97	0.96	0.97	80
<b>Precisión Total</b>	0.98 (465 muestras)			
<b>Promedio Macro</b>	0.98	0.98	0.98	
<b>Promedio Ponderado</b>	0.98	0.98	0.98	

Tabla 8.1: Reporte de Clasificación del Modelo Xception en el Conjunto de Prueba.

### 8.1.3. Prueba de Concepto de MLOps con MLFlow

Para mejorar la trazabilidad y reproducibilidad de los experimentos de *Machine Learning*, se implementó una prueba de concepto (POC) utilizando MLFlow. Esta herramienta facilita el registro, monitoreo y versionamiento de modelos, lo cual es esencial en procesos de MLOps (Machine Learning Operations). A continuación, se describen los resultados obtenidos al integrar MLFlow en el ciclo de vida del modelo Xception.

## Interfaz de Experimentos en MLFlow

La Figura 8.4 muestra la interfaz de MLFlow, en la cual se registraron dos ejecuciones de entrenamiento. En esta vista se pueden observar detalles importantes como la duración de cada ejecución, el estado final (completado), y las métricas registradas para facilitar el análisis y comparación.

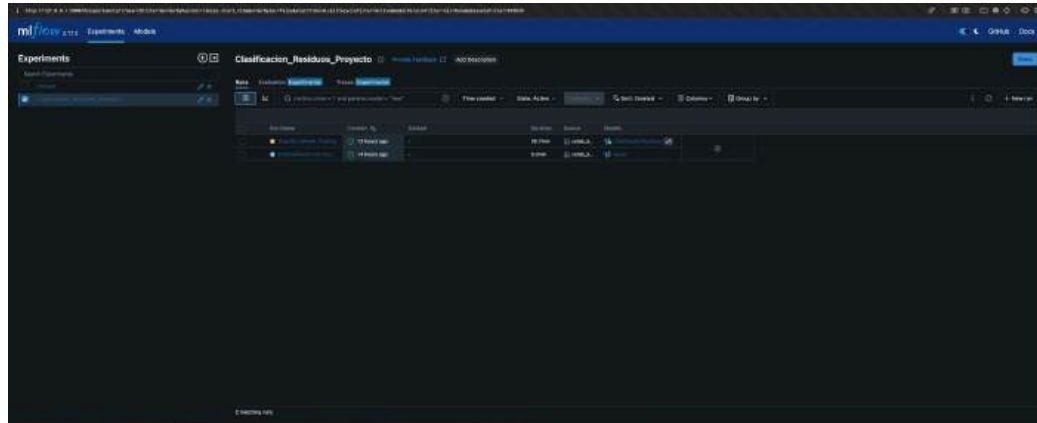


Figura 8.4: Interfaz de MLFlow mostrando los experimentos registrados. Aquí se observan las ejecuciones del modelo Xception y sus correspondientes métricas y parámetros.

## Detalles del Primer Experimento: Entrenamiento del Modelo Xception

La Figura 8.5 presenta los detalles del primer experimento, titulado "Xception Model Training". En esta ejecución, se registraron los siguientes parámetros clave: tasa de aprendizaje de 0.0001, tamaño de lote de 32, y un total de 200 épocas. Además, se muestra el valor de la precisión de validación (0.9732) y la pérdida de validación (0.5581), reflejando el rendimiento del modelo en esta configuración.



Figura 8.5: Detalles del experimento "Xception Model Training" en MLFlow. Se observan las métricas y los parámetros registrados durante el entrenamiento del modelo.

## Detalles del Segundo Experimento: Entrenamiento con GradCAM

En la Figura 8.6 se muestran los detalles del segundo experimento, titulado "Entrenamiento con Xception y GradCAM". Esta ejecución tuvo una duración de 9 minutos, y se configuró con parámetros similares al primer experimento, con la adición del valor de regularización L2 (0.002). Las métricas de validación registradas fueron una precisión de 0.9732 y una pérdida de 0.4891.



Figura 8.6: Detalles del experimento "Entrenamiento con Xception y GradCAM" en MLFlow. Se destacan las métricas de precisión y pérdida, junto con los parámetros utilizados.

### 8.1.4. Resultados de Explainable AI: GradCAM y LIME

#### Interpretación con GradCAM

Las siguientes figuras presentan ejemplos de interpretaciones generadas con GradCAM para las distintas clases de residuos. Los mapas de GradCAM resaltan las regiones cruciales que el modelo considera al realizar la clasificación, enfocándose en características visuales importantes, como bordes, contornos y texturas.





Explicación con LIME - e-waste



Figura 8.8: Explicación generada con LIME para la clase *Battery*. Las áreas resaltadas muestran las características que más influyeron en la predicción, indicando bordes y regiones específicas.



Figura 8.9: Explicación generada con LIME para la clase *Glass*. Las áreas resaltadas indican las texturas y bordes que el modelo consideró importantes para clasificar correctamente el residuo.

## Explicación con LIME - metal



Figura 8.10: Explicación generada con LIME para la clase *Metal*. Las áreas destacadas muestran las regiones clave que ayudaron al modelo a identificar las propiedades del metal, como el brillo y las texturas metálicas.

Explicación con LIME - orgánico



Figura 8.11: Explicación generada con LIME para la clase *Organic*. Las áreas resaltadas indican las texturas y patrones orgánicos que fueron más relevantes para la predicción.

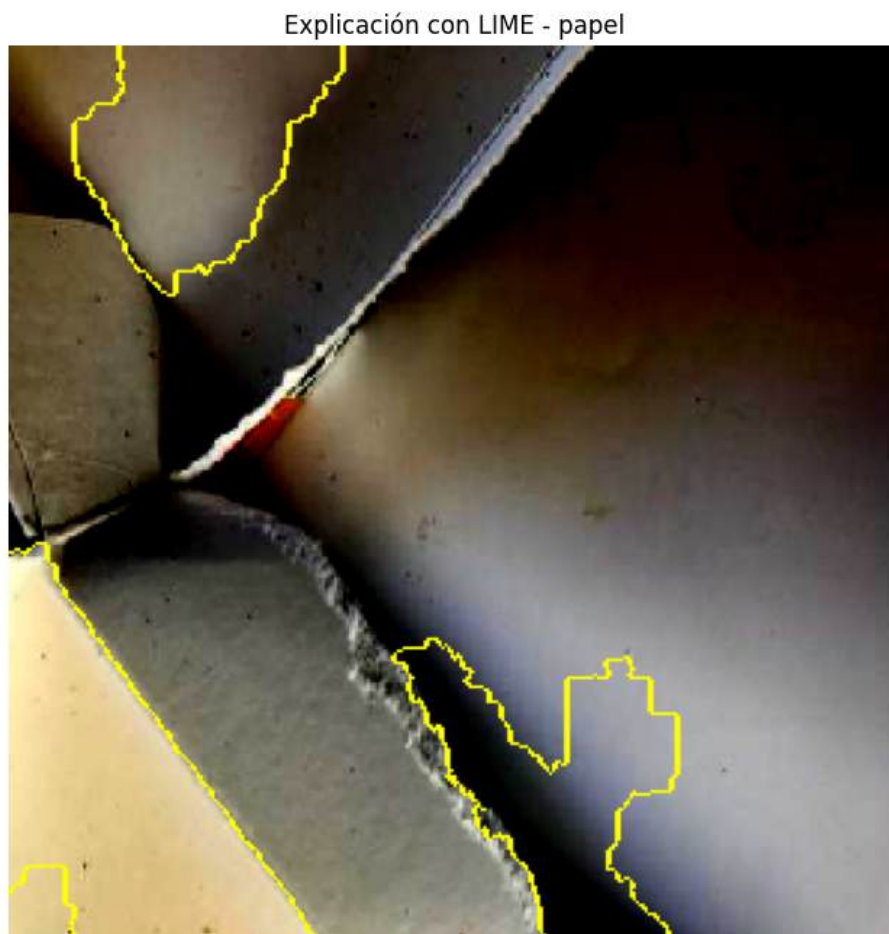


Figura 8.12: Explicación generada con LIME para la clase *Paper*. Las áreas importantes se centran en las texturas y bordes del papel que influenciaron la decisión del modelo.





Figura 8.13: Explicación generada con LIME para la clase *Plastic*. Las áreas resaltadas muestran las regiones específicas que el modelo utilizó para identificar el plástico, como los contornos y las variaciones en la textura.

### Pruebas de Saliency Maps con SmoothGrad

En esta subsección, se presentan los resultados de los saliency maps generados con SmoothGrad aplicados al conjunto de pruebas. SmoothGrad se utiliza para obtener mapas de atención más claros y menos ruidosos, resaltando las áreas que el modelo considera importantes para realizar la clasificación.

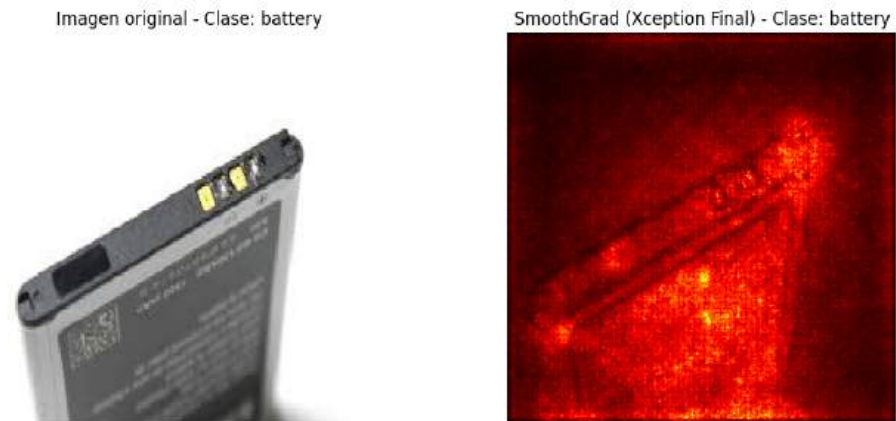


Figura 8.14: Saliency Map generado con SmoothGrad para la clase *Battery*.

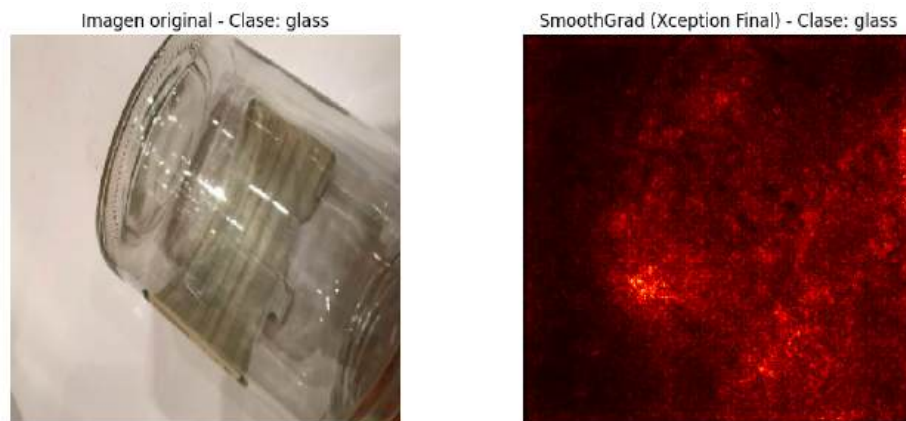


Figura 8.15: Saliency Map generado con SmoothGrad para la clase *Glass*.

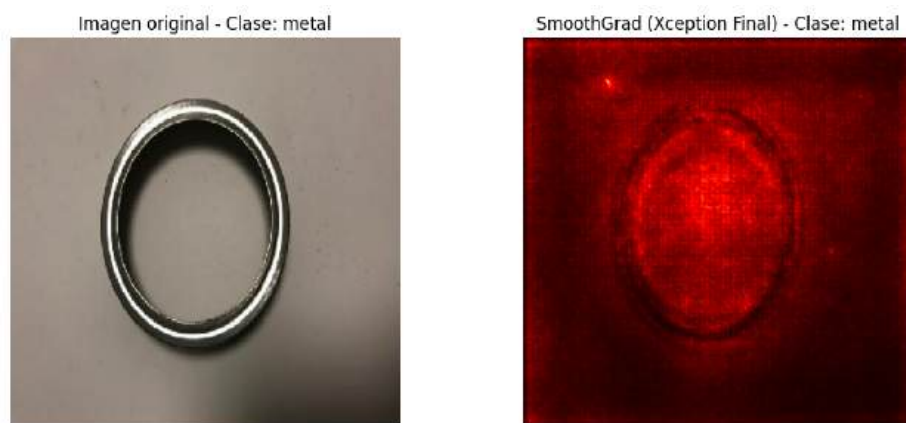


Figura 8.16: Saliency Map generado con SmoothGrad para la clase *Metal*.



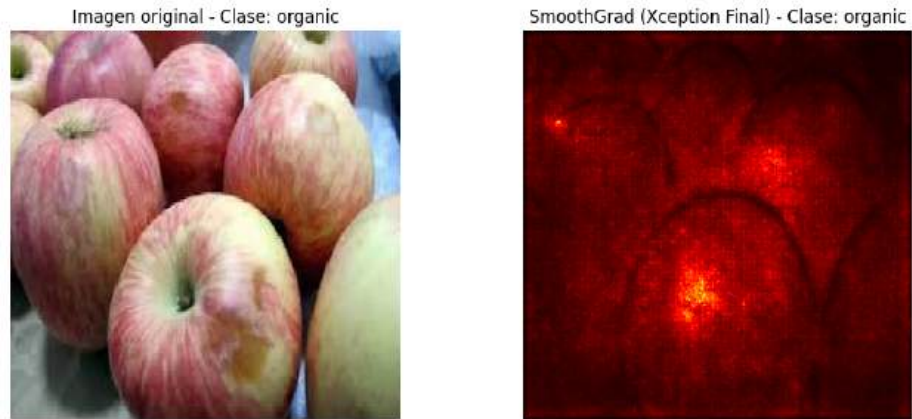


Figura 8.17: Saliency Map generado con SmoothGrad para la clase *Organic*.

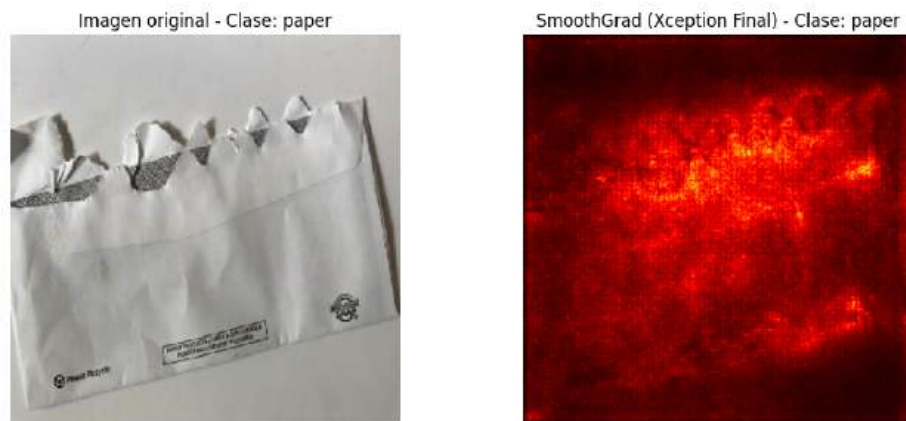


Figura 8.18: Saliency Map generado con SmoothGrad para la clase *Paper*.

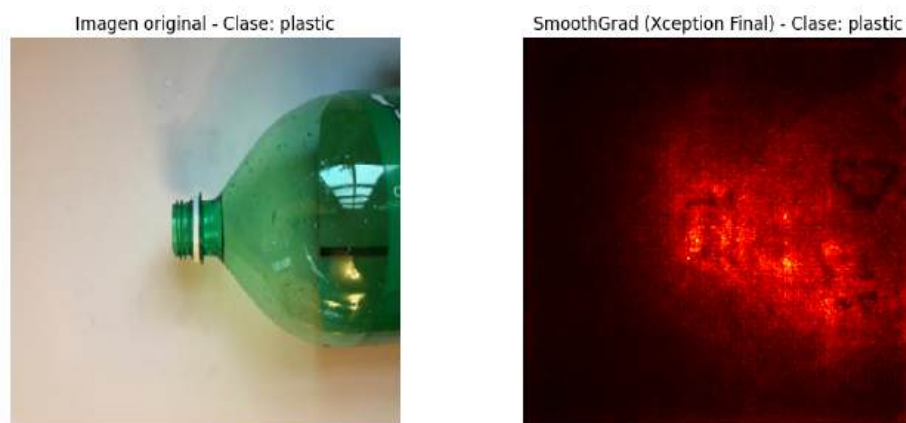


Figura 8.19: Saliency Map generado con SmoothGrad para la clase *Plastic*.

### Resultados de Pruebas con Imágenes Reales

Para evaluar el rendimiento del modelo en situaciones prácticas, se probaron 16 imágenes reales de residuos, con diferentes condiciones de fondo y contexto. Los resultados a continuación muestran cómo el modelo maneja estas situaciones, destacando tanto las imágenes correctamente clasificadas como los ejemplos más desafiantes.

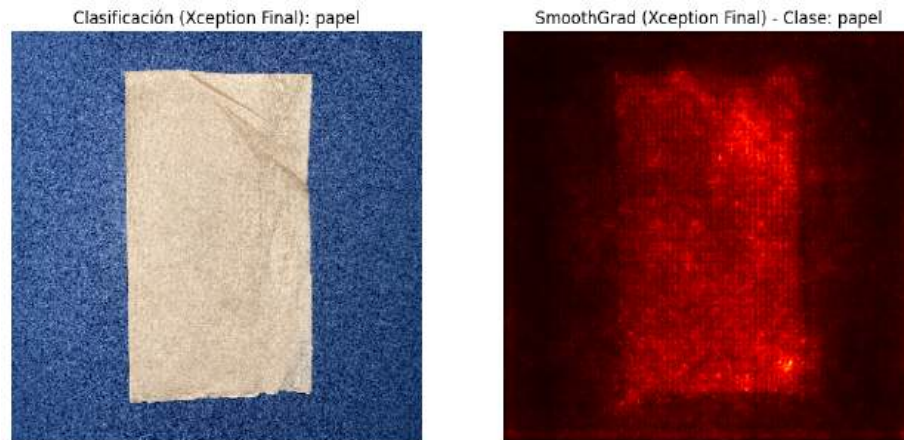
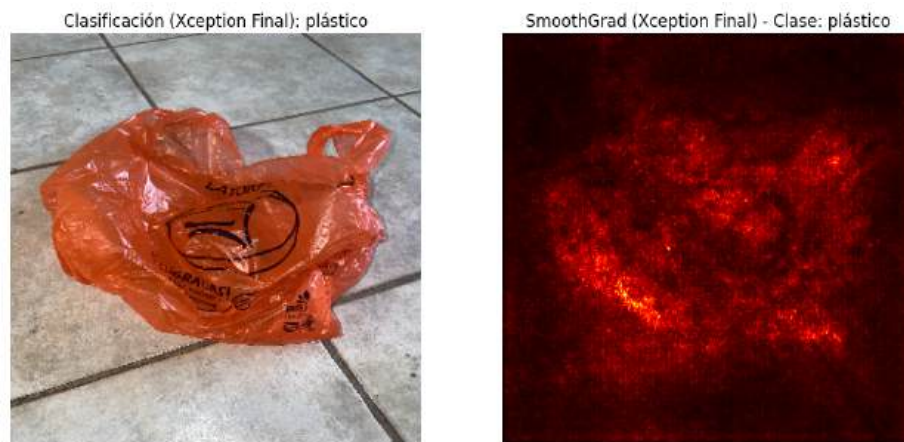
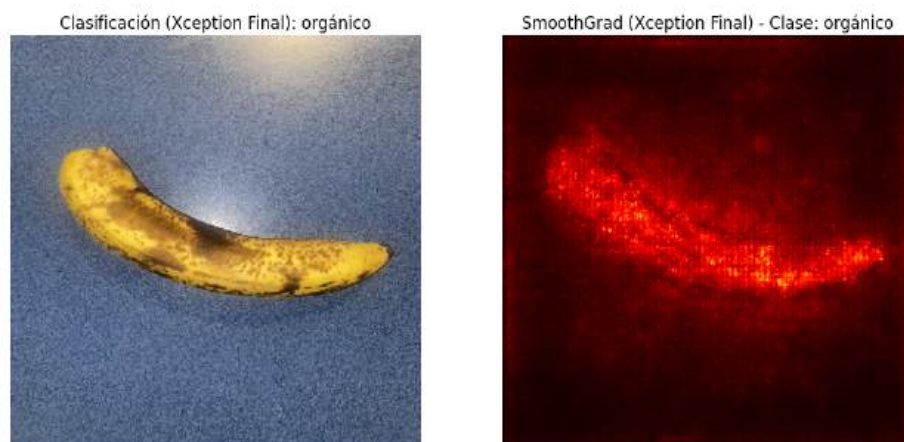
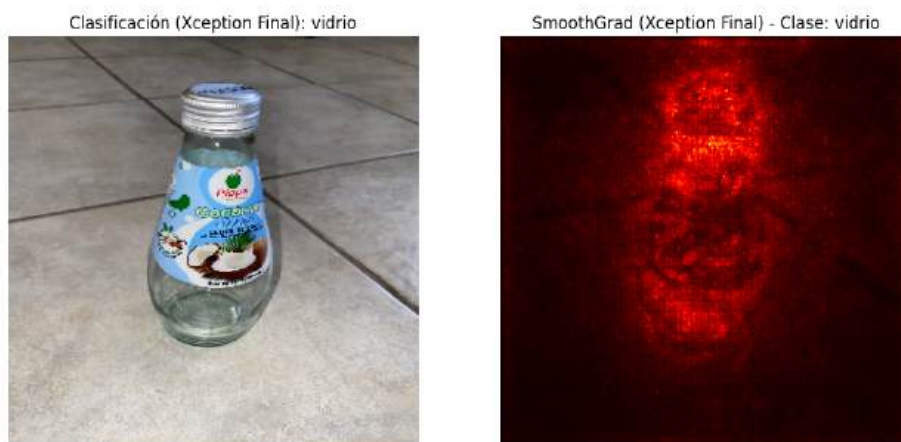
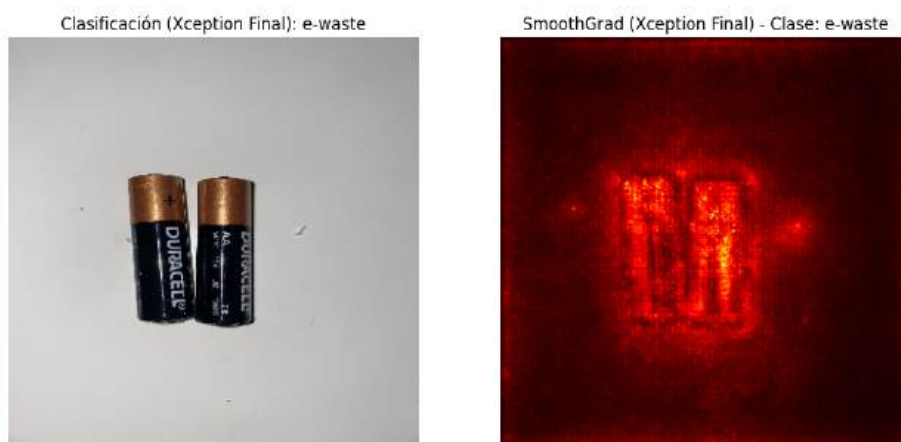
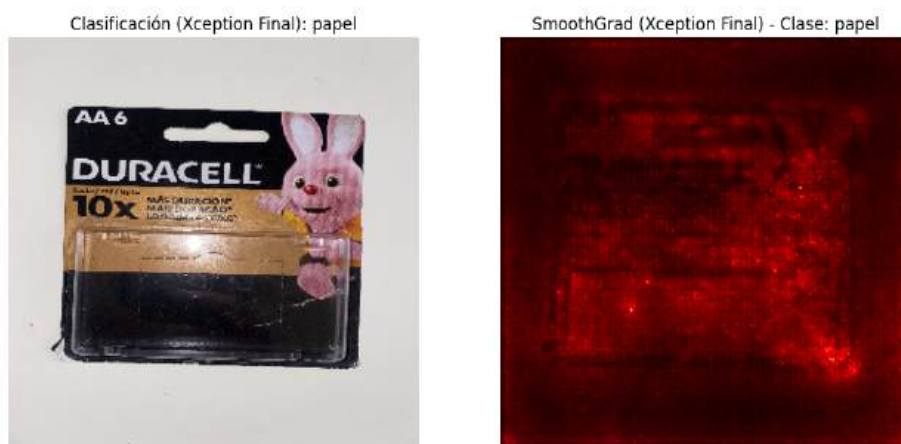


Figura 8.20: Clasificación de un residuo real perteneciente a la clase *Papel*.



Figura 8.21: Clasificación de un residuo real perteneciente a la clase *Metal*.

Figura 8.22: Clasificación de un residuo real perteneciente a la clase *Plastic*.Figura 8.23: Clasificación de un residuo real perteneciente a la clase *Plastic*.Figura 8.24: Clasificación de un residuo real perteneciente a la clase *Organic*.

Figura 8.25: Clasificación de un residuo real perteneciente a la clase *Glass*.Figura 8.26: Clasificación de un residuo real perteneciente a la clase *Battery*.Figura 8.27: Clasificación de un residuo real perteneciente a la clase *Paper*.



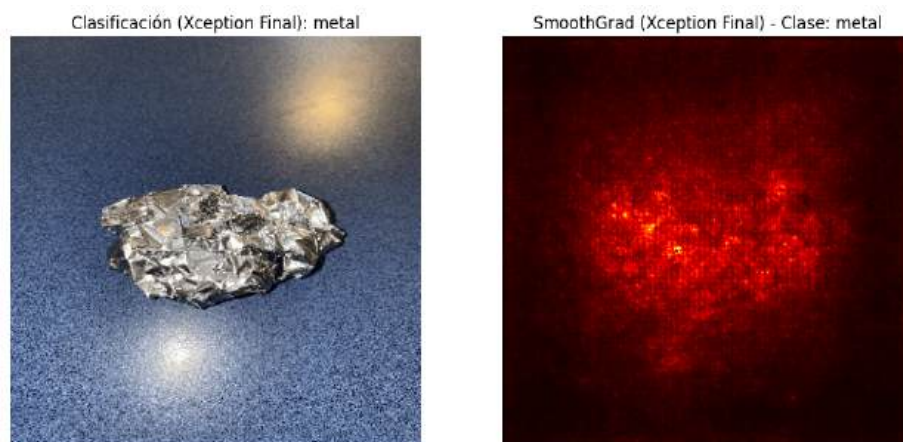


Figura 8.28: Clasificación de un residuo real perteneciente a la clase *Metal*.

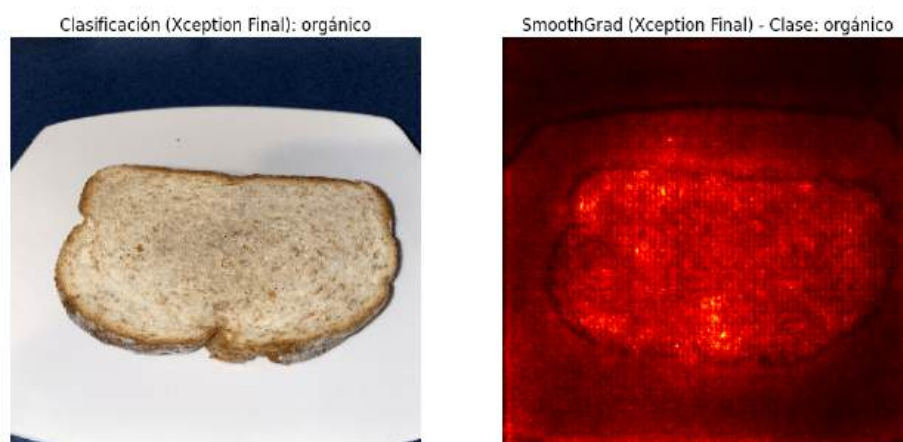


Figura 8.29: Clasificación de un residuo real perteneciente a la clase *Organic*.

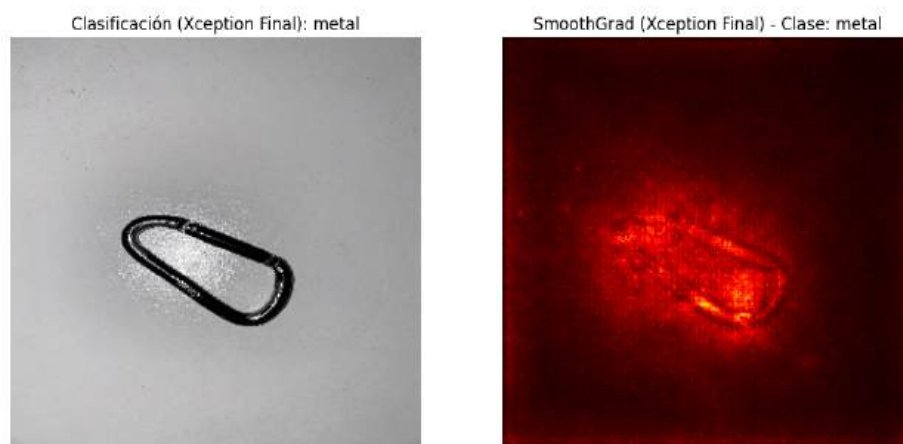
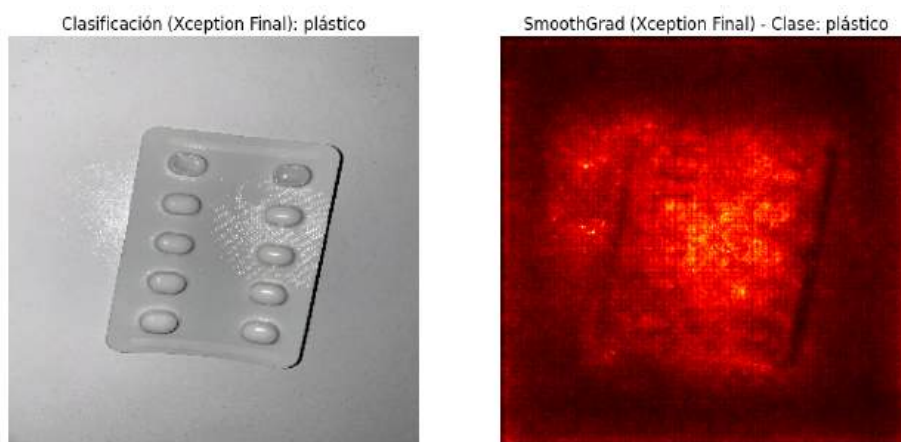
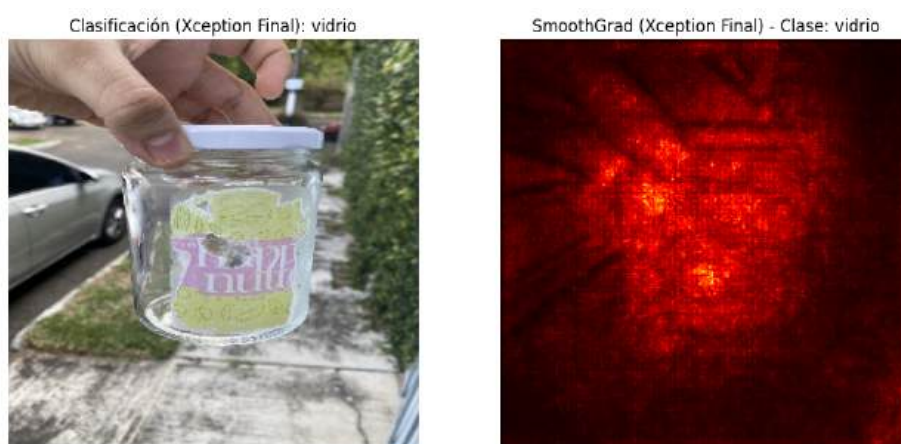


Figura 8.30: Clasificación de un residuo real perteneciente a la clase *Metal*.

Figura 8.31: Clasificación de un residuo real perteneciente a la clase *Plastic*.Figura 8.32: Clasificación de un residuo real perteneciente a la clase *Glass*.Figura 8.33: Clasificación de un residuo real perteneciente a la clase *Plastic*.

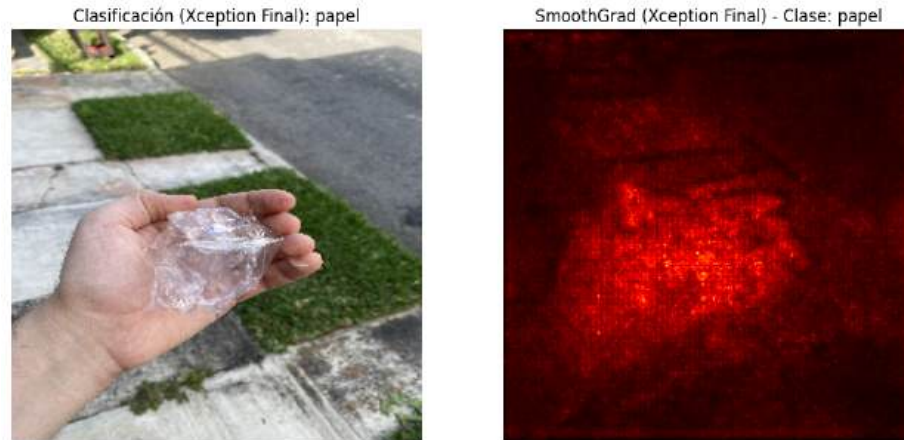


Figura 8.34: Clasificación de un residuo real perteneciente a la clase *Plastic*, pero que fue clasificado como *Paper*.



Figura 8.35: Clasificación de un residuo real perteneciente a la clase *Organic*.

### Tiempo Promedio de Clasificación por Imagen

El tiempo promedio de clasificación por imagen es una métrica importante para evaluar la eficiencia del modelo en aplicaciones prácticas. A continuación, se detalla el rendimiento temporal del modelo Xception final, destacando su viabilidad para su uso en sistemas en tiempo real.

```

1  # Función para predecir la clase de una imagen y mostrarla junto con su etiqueta
2  def classify_and_display_image(image_path):
3      # Cargar y mostrar la imagen
4      img = image.load_img(image_path, target_size=(299, 299))
5      plt.imshow(img)
6      plt.axis('off')
7
8      # Convertir la imagen a array y preprocesarla
9      img_array = image.img_to_array(img)
10     img_array = np.expand_dims(img_array, axis=0)
11     img_array = preprocess_input(img_array)
12

```

```

13     # Tomar el tiempo de inicio
14     start_time = time.time()
15
16     # Hacer la predicción
17     predictions = model.predict(img_array)
18
19     # Tomar el tiempo de fin y calcular el tiempo
20     end_time = time.time()
21     classification_time = end_time - start_time
22     classification_times.append(classification_time)
23
24     # Obtener la clase predicha (índice) y traducirla al español
25     predicted_class_idx = np.argmax(predictions[0])
26     predicted_class = list(class_translation.keys())[predicted_class_idx]
27     translated_class = class_translation[predicted_class]
28
29     # Mostrar la etiqueta traducida
30     plt.title(f'Clasificación (Xception Final): {translated_class}')
31     plt.show()
32
33     # Clasificar y mostrar cada imagen
34     for image_file in image_files:
35         classify_and_display_image(image_file)
36
37     # Calcular el tiempo promedio de clasificación
38     average_time = np.mean(classification_times)
39     print(f'Tiempo promedio de clasificación por imagen (Xception Final): {average_time:.4f} segundos')

```

Tiempo promedio de clasificación por imagen: 0.1945 segundos

## 8.2. Resultados del Chatbot Inteligente

En esta sección se presentan los resultados obtenidos a partir de la evaluación del chatbot desarrollado para la clasificación de residuos. Los resultados se basan en las respuestas de 7 expertos de la Unidad de Control Ambiental de la Municipalidad de la Ciudad de Guatemala, quienes evaluaron el desempeño del chatbot en cinco criterios clave mediante una rúbrica de 1 a 5 puntos.

### 8.2.1. Resultados de la Encuesta por Criterio

#### Precisión en las Respuestas

La precisión en las respuestas fue evaluada para determinar si el chatbot proporciona información correcta y clara en relación con la clasificación de residuos. De acuerdo con los resultados, el 57.1% de los participantes otorgó una puntuación de 5, y el 42.9% calificó con un 4.



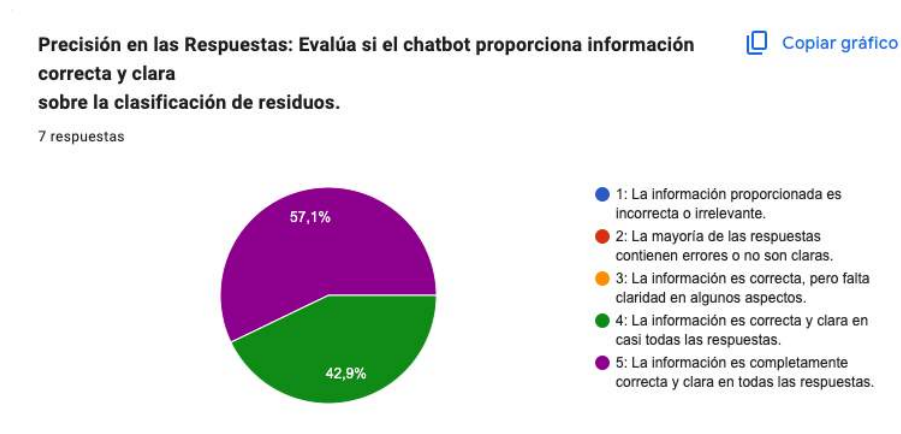


Figura 8.36: Evaluación de la precisión de las respuestas del chatbot.

### Relevancia Temática de las Respuestas

Este criterio mide el grado de alineación de las respuestas con el tema de clasificación de residuos y reciclaje. Los resultados muestran que el 57.1 % de los evaluadores otorgó una puntuación de 5, mientras que el 42.9 % calificó con un 4.



Figura 8.37: Evaluación de la relevancia temática de las respuestas.

### Capacidad para Rechazar Preguntas Fuera de Tema

El 85.7 % de los participantes calificó esta capacidad con un 5, mientras que el 14.3 % la evaluó con un 4, lo que sugiere que el chatbot logra rechazar preguntas fuera de contexto de manera educada y clara.

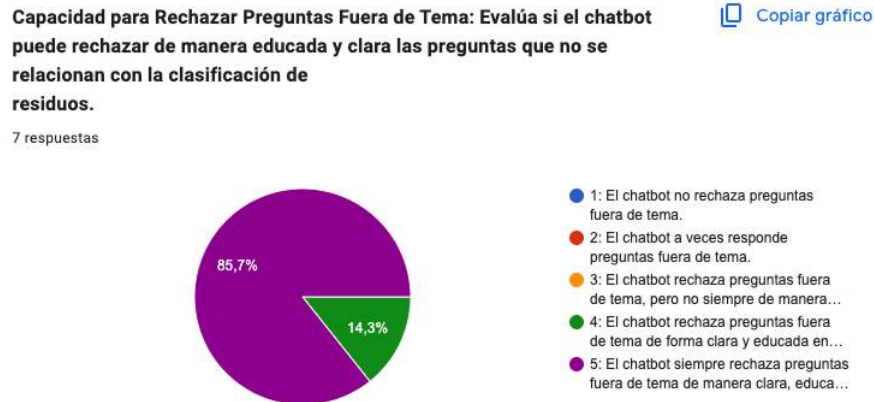


Figura 8.38: Evaluación de la capacidad del chatbot para rechazar preguntas fuera de tema.

### Calidad de las Recomendaciones sobre Manejo de Residuos

El 71.4 % de los evaluadores calificaron la calidad de las recomendaciones con un 4, mientras que el 28.6 % le otorgó un 5, destacando la utilidad y relevancia de las sugerencias proporcionadas.

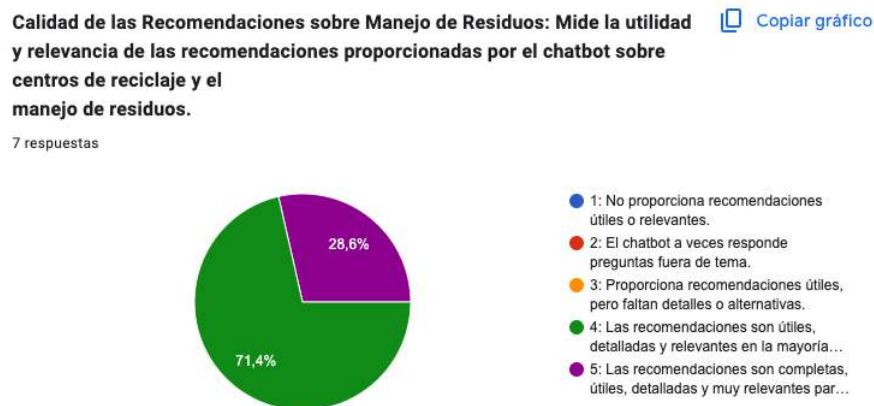


Figura 8.39: Evaluación de la calidad de las recomendaciones proporcionadas.

### Consistencia en el Estilo de las Respuestas

El 57.1 % de los evaluadores calificaron este aspecto con un 5, mientras que el 42.9 % otorgó un 4, indicando que el chatbot mantiene una consistencia estilística adecuada.

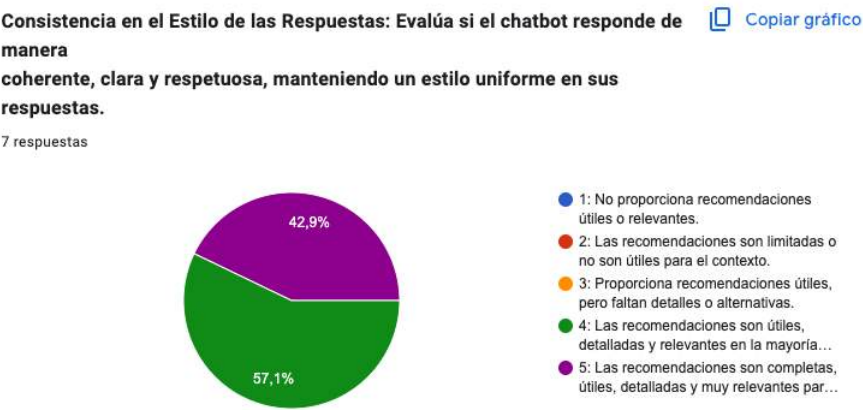


Figura 8.40: Evaluación de la consistencia en el estilo de las respuestas.

8.2.2. Puntaje Total y Precisión General

Se calcularon los puntajes totales y el porcentaje de precisión de cada participante en la evaluación del chatbot. Los resultados se presentan en la Tabla 8.2.

Tabla 8.2: Puntaje total y porcentaje de precisión de cada participante en la evaluación del chatbot.

Participante	Precisión	Relevancia	Rechazo	Recomendaciones	Consistencia	Total	Promedio (%)
Experto 1	4	4	5	4	5	22	88.0
Experto 2	5	5	5	5	5	25	100.0
Experto 3	4	5	5	4	4	22	88.0
Experto 4	5	5	5	5	5	25	100.0
Experto 5	5	5	5	4	4	23	92.0
Experto 6	4	4	4	4	4	20	80.0
Experto 7	5	4	5	4	4	22	88.0

El puntaje total promedio de precisión y calidad del chatbot fue de **90.86 %**, indicando un alto nivel de precisión y efectividad en la asistencia para la clasificación de residuos.

8.2.3. Tiempos de Respuesta del Chatbot

Los tiempos de respuesta del chatbot se evaluaron a través de 10 pruebas consecutivas con la misma pregunta. Los resultados se resumen en la Tabla 8.3.

Tabla 8.3: Tiempos de respuesta del chatbot en cada prueba realizada.

Prueba	Tiempo de Respuesta (segundos)
1	2.67
2	0.82
3	2.09
4	1.08
5	2.52
6	0.72
7	1.79
8	0.93
9	2.44
10	0.96
Promedio	1.70

El tiempo promedio de respuesta fue de **1.7 segundos**, lo cual es adecuado para una experiencia de usuario rápida y fluida.

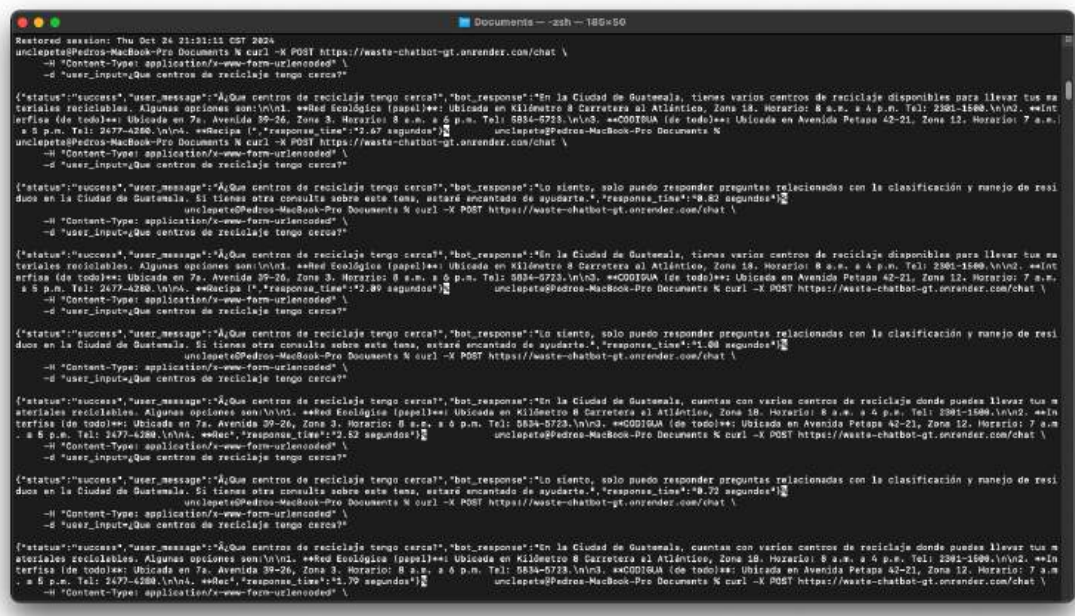


Figura 8.41: Tiempo promedio de respuesta del chatbot.

8.2.4. Pruebas de Preguntas y Respuestas del Chatbot

A continuación, se muestran ejemplos de interacciones con el chatbot, junto con las respuestas proporcionadas. Estas pruebas demuestran cómo el chatbot gestiona consultas relacionadas con la clasificación de residuos y preguntas fuera de contexto.

Tabla 8.4: Pruebas de interacción con el chatbot y sus respuestas.

Pregunta del Usuario	Respuesta del Chatbot	Tiempo de Respuesta
¿Que puedo hacer con mis residuos plásticos?	Puedes clasificar tus residuos plásticos como reciclables y llevarlos a un centro de reciclaje adecuado en la Ciudad de Guatemala. Algunos centros que reciben materiales plásticos son Interfisa, CODIGUA y Recipa. Recuerda que es importante que los residuos plásticos estén limpios y secos para facilitar su reciclaje. Si necesitas más información sobre cómo manejar tus residuos plásticos, no dudes en preguntar.	2.14 segundos
¿Cómo debo clasificar los residuos orgánicos?	Los residuos orgánicos deben ser clasificados en la categoría de "orgánicos" (Verde). Estos residuos incluyen materiales de origen animal o vegetal que se descomponen de forma natural, como cáscaras de frutas y verduras, restos de comida, hojas secas y restos de jardinería. Es importante separar los residuos orgánicos de los reciclables y no reciclables para facilitar su tratamiento adecuado. Si tienes más preguntas sobre la clasificación de residuos orgánicos o necesitas información adicional, estaré encantado de ayudarte.	1.70 segundos
¿Dónde puedo reciclar el vidrio en la Ciudad de Guatemala?	En la Ciudad de Guatemala, puedes reciclar el vidrio en el centro de reciclaje Interfisa, que acepta materiales de vidrio entre otros tipos de residuos reciclables. Interfisa está ubicada en 7a. Avenida 39-26, Zona 3, y su horario de atención es de 8 a.m. a 6 p.m. Si necesitas más información sobre el reciclaje de vidrio u otros materiales, no dudes en preguntar.	1.64 segundos
¿Cuáles son las reglas para desechar desechos electrónicos?	Las normas para desechar desechos electrónicos en la Ciudad de Guatemala implican no mezclarlos con la basura común, ya que contienen componentes que pueden ser perjudiciales para el medio ambiente si no se manejan correctamente. Para desechar desechos electrónicos como computadoras, teléfonos celulares, baterías, y otros aparatos electrónicos, se recomienda llevarlos a centros de reciclaje especializados en este tipo de residuos. En la Ciudad de Guatemala, puedes contactar con centros de reciclaje como Interfisa, CODIGUA y Recipa, quienes pueden recibir y manejar adecuadamente estos desechos electrónicos.	2.21 segundos
¿Cuál es la capital de Francia?	Lo siento, solo puedo responder preguntas relacionadas con la clasificación y manejo de residuos en la Ciudad de Guatemala. Si tienes alguna consulta sobre ese tema, estaré encantado de ayudarte.	0.89 segundos

## 9.1. Análisis de Resultados del Modelo Xception

### 9.1.1. Análisis de las Métricas de Precisión y Pérdida

El rendimiento del modelo Xception fue evaluado mediante las métricas de precisión y pérdida tanto en el conjunto de entrenamiento como en el conjunto de validación. Estas métricas se analizaron para comprender la capacidad de aprendizaje y generalización del modelo.

**Historia de Precisión y Pérdida** Las gráficas de precisión y pérdida ilustran cómo el modelo Xception aprende y ajusta sus parámetros a lo largo de las épocas. Se observa que la precisión en el conjunto de entrenamiento aumenta rápidamente, alcanzando casi el 100 %, mientras que la precisión en el conjunto de validación se estabiliza en un valor elevado de aproximadamente 98 %. Este comportamiento refleja la capacidad del modelo para capturar patrones complejos en los datos de entrenamiento y mantener una alta precisión en datos no vistos.

La pérdida en el conjunto de entrenamiento disminuye de manera pronunciada en las primeras épocas y luego se estabiliza en un valor bajo. De forma similar, la pérdida en el conjunto de validación sigue una trayectoria descendente y se estabiliza sin mostrar un aumento significativo, lo que indica que el modelo no está sobreajustado. Este comportamiento es producto de una adecuada configuración de hiperparámetros y del uso de técnicas de regularización como *Dropout* y la regularización L2, las cuales ayudan a mitigar el sobreajuste y mejorar la capacidad de generalización.

**Eficacia del Aprendizaje** El modelo Xception, conocido por su arquitectura basada en convoluciones separables en profundidad, es particularmente eficiente para la extracción de características. Esta capacidad ha sido esencial para manejar la variabilidad visual inherente a los residuos, permitiendo que el modelo distinga entre clases visualmente similares. La alta precisión de validación sugiere que el diseño del modelo ha sido adecuado para esta tarea de clasificación de residuos, aprovechando la eficiencia de las convoluciones para capturar detalles relevantes en las imágenes.

### 9.1.2. Análisis de la Matriz de Confusión

La matriz de confusión proporciona un análisis detallado de las predicciones correctas e incorrectas del modelo para cada clase de residuos. Las predicciones se concentran en gran medida en la diagonal principal, lo que es indicativo de un rendimiento sobresaliente en la clasificación.

#### Rendimiento por Clase

- **Clases con Alto Rendimiento:** Las clases *Battery* y *Paper* destacan por tener casi todas las predicciones correctas, con tasas de precisión y *recall* cercanas al 100 %. Las características visuales distintivas de estas clases, como los contornos definidos y las texturas homogéneas, han sido capturadas eficazmente por el modelo, facilitando una clasificación precisa.
- **Clases con Desafíos Moderados:** Se observan algunas confusiones en las clases *Plastic* y *Glass*. Estas confusiones pueden explicarse por similitudes visuales, como superficies reflectantes y traslúcidas que pueden compartir ciertos residuos plásticos y de vidrio. A pesar de esto, el modelo aún logra mantener un alto nivel de precisión, lo que es una evidencia de su capacidad para extraer y diferenciar características sutiles.
- **Observaciones Generales:** La efectividad del modelo Xception se ve reforzada por su capacidad para manejar estas similitudes visuales mediante una representación de características robusta. Las capas convolucionales de Xception, optimizadas para la separación de canales espaciales y de profundidad, han permitido al modelo identificar patrones relevantes que son críticos para una clasificación precisa.

La matriz de confusión, en conjunto con las métricas asociadas, valida la eficacia del modelo en términos de su capacidad de diferenciación y generalización. El análisis de los casos específicos en los que ocurren errores de clasificación puede proporcionar información útil para futuras optimizaciones del modelo.

### 9.1.3. Análisis del Reporte de Clasificación

El reporte de clasificación presenta las métricas de *precision*, *recall* y *F1-score* para cada clase de residuos, lo que permite una evaluación granular del rendimiento del modelo.

#### Métricas de Rendimiento

- **Precisión Elevada:** La precisión promedio del 98 % indica que el modelo realiza predicciones correctas con una frecuencia muy alta. Esto es crucial en aplicaciones como la clasificación de residuos, donde las predicciones erróneas pueden afectar la eficacia del proceso de reciclaje. La alta precisión es especialmente destacable en las clases *Battery* y *Paper*, donde los patrones visuales específicos han sido correctamente aprendidos y reconocidos por el modelo.
- **Recall Consistente:** Un *recall* promedio del 98 % muestra que el modelo tiene una alta capacidad para identificar ejemplos positivos de cada clase, minimizando los falsos negativos. Esto es esencial en sistemas de reciclaje automatizado, ya que asegura que la mayoría de los residuos se clasifiquen correctamente sin perder instancias importantes.
- **F1-Score Balanceado:** El *F1-score* promedio también se mantiene en un 98 %, reflejando un equilibrio ideal entre precisión y *recall*. Esto indica que el modelo no solo es preciso, sino que también mantiene una capacidad de detección uniforme en todas las clases. Incluso las clases

que presentan ciertos desafíos, como *Plastic* y *Glass*, muestran un *F1-score* competitivo, lo que reafirma la solidez del modelo.

Estas métricas resaltan el éxito de la arquitectura Xception para esta tarea específica. La combinación de precisión, *recall* y *F1-score* elevados en todas las clases demuestra que el modelo es robusto y confiable, con un desempeño uniforme que lo hace adecuado para aplicaciones prácticas en la gestión de residuos.

#### 9.1.4. Análisis de Explainable AI: GradCAM y LIME

La interpretabilidad de los modelos de redes neuronales profundas es un tema crucial en la aplicación de sistemas de IA, especialmente en contextos críticos como la clasificación de residuos. En este proyecto, se utilizaron herramientas de Explainable AI, como GradCAM y LIME, para ofrecer una interpretación clara de las decisiones del modelo Xception. Estas técnicas no solo ayudan a validar el rendimiento del modelo, sino que también proporcionan una comprensión más profunda de cómo el modelo procesa las imágenes y qué características considera importantes.

##### Interpretaciones con GradCAM

GradCAM (Gradient-weighted Class Activation Mapping) es una técnica que permite visualizar las áreas de una imagen que activan más intensamente las últimas capas convolucionales del modelo. En el caso del modelo Xception, se generaron mapas de calor para evaluar las regiones de las imágenes en las que el modelo se enfoca al realizar sus predicciones.

**Observaciones y Comportamiento del Modelo** Los mapas de GradCAM revelaron cómo el modelo Xception distribuye su atención al clasificar diferentes tipos de residuos. Se observó lo siguiente:

- **Clase Metal:** Los mapas de GradCAM destacan claramente los bordes brillantes y las superficies reflejantes de los objetos metálicos. Este enfoque es consistente con la naturaleza visual de los metales, que generalmente tienen superficies lisas y reflectantes. La arquitectura Xception, al descomponer las imágenes en componentes más simples mediante convoluciones separables en profundidad, es capaz de capturar estos patrones complejos de manera eficiente. Esto demuestra que el modelo ha aprendido a identificar correctamente las señales visuales clave para esta clase.
- **Clase Vidrio:** Para residuos de vidrio, GradCAM muestra una atención hacia las áreas de transparencia y los bordes del material. La capacidad del modelo para identificar las texturas translúcidas y las formas definidas del vidrio refuerza la idea de que Xception puede manejar variaciones visuales sutiles gracias a su estructura de extracción de características optimizada. La atención en las áreas correctas también sugiere que el modelo ha aprendido a ignorar elementos irrelevantes del fondo, lo que es crucial para aplicaciones en entornos reales.
- **Clase Plástico:** Aquí, los mapas de calor muestran una ligera dispersión en la atención del modelo, centrándose en las áreas de color y los contornos suaves. La variabilidad visual en los residuos plásticos, que pueden presentarse en diferentes colores, formas y texturas, introduce un desafío adicional. Esta dispersión en la atención sugiere que el modelo podría beneficiarse de una mayor diversidad en los datos de entrenamiento para mejorar su capacidad de generalización. Sin embargo, el enfoque en características distintivas como los contornos y las superficies de color uniforme es un indicio positivo de que el modelo está utilizando las pistas visuales adecuadas.



- **Clase Orgánico:** Los residuos orgánicos presentan un patrón visual caracterizado por texturas rugosas y formas irregulares. Los mapas de GradCAM muestran que el modelo Xception se centra en estas características, lo que es coherente con la naturaleza heterogénea de los desechos orgánicos. La habilidad del modelo para identificar estas texturas complejas demuestra la eficacia de las convoluciones separables en profundidad de Xception, que maximizan la extracción de detalles relevantes mientras minimizan el costo computacional.
- **Clase Papel:** GradCAM revela que el modelo presta atención a las superficies lisas y los bordes definidos del papel, ignorando elementos de fondo irrelevantes. Esto es importante, ya que las características visuales del papel suelen ser menos complejas que las de otras clases. El modelo parece haber aprendido de manera eficiente a diferenciar el papel de otros materiales mediante la identificación de patrones visuales simples pero distintivos.
- **Clase Battery:** Los residuos electrónicos, como las baterías, se identifican a través de detalles específicos, como los terminales y las formas geométricas distintivas. Los mapas de GradCAM destacan estas áreas, indicando que el modelo está utilizando información visual precisa y relevante. La capacidad de Xception para detectar detalles finos y estructuras complejas se refleja en la atención que el modelo presta a las características esenciales de las baterías.

**Importancia de la Arquitectura Xception** La arquitectura Xception se basa en convoluciones separables en profundidad, que descomponen el proceso de convolución en dos pasos: una convolución de profundidad y una convolución puntual. Esta estructura permite una extracción de características más eficiente y detallada. En este contexto, GradCAM ha demostrado que Xception es capaz de capturar patrones visuales relevantes para cada clase de residuos, lo que refuerza la idea de que este modelo es adecuado para manejar datos con una gran diversidad visual.

### Interpretaciones con LIME

LIME (Local Interpretable Model-agnostic Explanations) ofrece una forma de interpretar las predicciones del modelo al resaltar los segmentos de imagen que más contribuyen a una clasificación específica. LIME proporciona una visión más granular y detallada de cómo el modelo Xception toma decisiones basadas en las características visuales de las imágenes.

**Desglose de las Explicaciones de LIME** LIME fue utilizado para descomponer las imágenes en segmentos y evaluar la importancia de cada uno en la predicción del modelo. Aquí se detallan las observaciones clave:

- **Clase Metal:** LIME destaca segmentos que contienen superficies brillantes y áreas reflectantes. Esto es coherente con el análisis de GradCAM, donde se observa que el modelo prioriza las características visuales asociadas con el metal. La redundancia en la atención a estas características refuerza la confianza en la precisión del modelo.
- **Clase Vidrio:** Los segmentos resaltados por LIME incluyen áreas transparentes y bordes definidos, lo que indica que el modelo Xception utiliza información visual significativa para clasificar correctamente los objetos de vidrio. La interpretación granular de LIME valida que el modelo no está sobreajustado a patrones irrelevantes y que realmente comprende las propiedades visuales de los residuos de vidrio.
- **Clase Plástico:** Las explicaciones de LIME muestran una mayor variabilidad en los segmentos importantes, lo que subraya la diversidad intrínseca de los residuos plásticos. Aunque el modelo identifica correctamente las áreas clave, la dispersión en la atención sugiere que el modelo aún puede beneficiarse de un conjunto de datos más representativo para mejorar su rendimiento en esta clase.

- **Clase Orgánico:** LIME resalta segmentos que corresponden a texturas rugosas y colores naturales, lo que es indicativo de un modelo que ha aprendido a identificar la heterogeneidad de los residuos orgánicos. Esto es relevante porque demuestra que el modelo no se enfoca en artefactos visuales sino en las características esenciales de los desechos.
- **Clase Papel:** Los segmentos resaltados en las imágenes de papel muestran que el modelo utiliza bordes y áreas planas para realizar sus predicciones. Esto es consistente con las observaciones de GradCAM y refleja una comprensión adecuada de las características distintivas del papel.
- **Clase Battery:** LIME destaca áreas como los terminales de las baterías y otros detalles específicos. Esto confirma que el modelo está utilizando información relevante para clasificar residuos electrónicos, lo cual es crucial para la precisión y fiabilidad en entornos reales.

**Justificación y Relevancia de las Interpretaciones** Las interpretaciones proporcionadas por LIME y GradCAM son fundamentales para entender y justificar las decisiones del modelo. En términos de Explainable AI, estos resultados demuestran que el modelo Xception no solo es preciso, sino también interpretativamente sólido, lo que es esencial para aplicaciones críticas como la gestión de residuos. Además, en el contexto de MLOps, la transparencia y trazabilidad proporcionadas por estas herramientas facilitan el monitoreo y la auditoría del modelo, garantizando que las decisiones sean confiables y explicables.

**Impacto y Comparación con Otros Estudios** En comparación con estudios previos, donde los modelos de clasificación de residuos a menudo carecen de interpretabilidad, el uso de Xception junto con técnicas de Explainable AI en este proyecto representa un avance significativo. Mientras que otros enfoques pueden lograr precisión, este trabajo destaca por su capacidad para explicar las decisiones del modelo de manera clara y comprensible, lo que es crucial para la adopción en aplicaciones reales. La arquitectura Xception, combinada con estas herramientas de explicabilidad, proporciona un modelo que no solo es eficaz sino también transparente y auditable.

**Conclusiones del Análisis** El análisis detallado con GradCAM y LIME confirma que el modelo Xception utiliza características visuales significativas para clasificar los residuos. Esto valida que las decisiones del modelo son razonables y justificadas, lo que es crucial para su implementación en sistemas de gestión de residuos. La estructura optimizada de Xception permite un aprendizaje efectivo y una clasificación precisa, lo que refuerza la elección de este modelo para esta tarea. Además, la capacidad de auditar y comprender el comportamiento del modelo a través de Explainable AI fortalece la confiabilidad del sistema.

### 9.1.5. Análisis de Saliency Maps con SmoothGrad

Para proporcionar una interpretación más precisa y menos ruidosa de las decisiones del modelo Xception, se utilizaron saliency maps generados con la técnica **SmoothGrad**. SmoothGrad es una mejora de los mapas de saliencia tradicionales que añade ruido a la entrada y promedia los gradientes obtenidos. Esto ayuda a suavizar los mapas de saliencia y resaltar de manera más clara las áreas importantes en las que el modelo se enfoca. A continuación, se realiza un análisis detallado de cómo SmoothGrad contribuye a la comprensión del rendimiento del modelo en el set de validación/prueba.

#### Principios de SmoothGrad y su Aplicación

SmoothGrad funciona agregando ruido gaussiano a las imágenes de entrada y calculando los gradientes varias veces. Luego, los mapas de saliencia resultantes se promedian, lo que reduce el

ruido y proporciona una visualización más coherente de las áreas de atención del modelo. Esto es particularmente útil en redes profundas como Xception, donde los mapas de saliencia originales pueden ser ruidosos y difíciles de interpretar.

**Evidencia de Aprendizaje Significativo** Los saliency maps generados con SmoothGrad mostraron patrones consistentes que validan el aprendizaje significativo del modelo Xception. A continuación, se describen las observaciones clave por clase:

- **Clase Metal:** Los mapas de saliencia de SmoothGrad destacaron las superficies brillantes y los bordes definidos de los objetos metálicos. Esta observación es consistente con las propiedades visuales del metal, que a menudo se caracteriza por reflejar la luz de manera distintiva. La arquitectura Xception, con su enfoque en convoluciones separables en profundidad, parece ser particularmente efectiva para captar estas características visuales, lo que explica la alta precisión del modelo en esta clase. El enfoque en las áreas relevantes sugiere que el modelo no se ve distraído por características irrelevantes del fondo.
- **Clase Vidrio:** Para los residuos de vidrio, los saliency maps resaltaron las áreas donde la transparencia y los bordes de los objetos eran prominentes. Esta atención precisa indica que el modelo ha aprendido a distinguir los objetos de vidrio, que suelen ser más desafiantes debido a su naturaleza translúcida. La estructura optimizada de Xception, que permite una mejor extracción de características complejas, es claramente ventajosa en este contexto. El modelo ignora eficazmente los elementos de fondo, lo que demuestra un aprendizaje robusto y una capacidad de generalización adecuada.
- **Clase Plástico:** Los residuos plásticos presentaron un patrón más disperso en los mapas de saliencia, lo que refleja la diversidad visual de esta clase. Sin embargo, SmoothGrad ayudó a clarificar las áreas clave en las que el modelo se enfoca, como los contornos y las texturas homogéneas. Esta observación sugiere que, aunque el modelo tiene cierta dificultad con la variabilidad de los residuos plásticos, sigue utilizando características significativas para realizar sus predicciones. La dispersión moderada en la atención resalta la necesidad de más datos de entrenamiento que capturen mejor la diversidad de los residuos plásticos.
- **Clase Orgánico:** Los saliency maps de los residuos orgánicos mostraron un enfoque claro en las texturas rugosas y los colores naturales. Esto es importante, ya que los residuos orgánicos son visualmente heterogéneos y pueden ser difíciles de clasificar. El modelo parece haber aprendido a identificar estas características complejas, lo que es una señal positiva de su capacidad de generalización. La arquitectura de Xception, con su capacidad para extraer patrones espaciales detallados, es particularmente adecuada para esta tarea, lo que explica el buen rendimiento del modelo en esta clase.
- **Clase Papel:** En el caso del papel, los saliency maps destacaron las áreas planas y los bordes claros, lo que es coherente con las características visuales simples de esta clase. El modelo Xception muestra un enfoque preciso y consistente, lo que sugiere que ha aprendido eficazmente las señales visuales necesarias para diferenciar el papel de otros materiales. Esto es una indicación de que la configuración del modelo es adecuada y que el aprendizaje ha sido efectivo.
- **Clase Battery:** Los residuos electrónicos, como las baterías, son identificados a través de detalles finos y características específicas. Los mapas de saliencia de SmoothGrad resaltaron áreas como los terminales y las etiquetas de advertencia, lo que demuestra que el modelo utiliza información visual relevante y específica para esta clase. Esto es crucial, ya que los residuos electrónicos pueden tener un impacto ambiental significativo si no se gestionan adecuadamente. La capacidad del modelo para enfocarse en detalles clave refuerza la efectividad de la arquitectura Xception en la extracción de características complejas.

**Explicabilidad y Validación del Modelo** El uso de SmoothGrad proporciona una validación adicional de que el modelo Xception toma decisiones basadas en características visuales significativas y no en artefactos irrelevantes. Esta transparencia es fundamental en aplicaciones críticas como la gestión de residuos, donde la interpretabilidad del modelo es esencial para generar confianza en su uso. Además, el hecho de que el modelo se enfoque en las áreas correctas sugiere que las técnicas de regularización y ajuste de hiperparámetros han sido efectivas.

**Comparación con Otros Métodos de Interpretabilidad** En comparación con otros enfoques de interpretabilidad, como los mapas de saliencia tradicionales, SmoothGrad ofrece una visualización mucho más clara y consistente. Esto es particularmente importante para modelos como Xception, que tienen una gran cantidad de parámetros y capas complejas. Mientras que los mapas de saliencia tradicionales pueden ser ruidosos y difíciles de interpretar, SmoothGrad reduce este ruido y destaca las áreas de interés de manera más precisa.

**Impacto en el Ciclo de Vida del Modelo y MLOps** Desde una perspectiva de MLOps, la capacidad de interpretar y validar las decisiones del modelo con herramientas como SmoothGrad es crucial para la gestión del ciclo de vida del modelo. La transparencia proporcionada por SmoothGrad facilita la monitorización y auditoría del modelo, asegurando que las decisiones sean confiables y explicables. Esto es especialmente importante cuando se considera la implementación del modelo en entornos de producción, donde la trazabilidad y la responsabilidad son esenciales.

## Conclusiones del Análisis con SmoothGrad

El análisis con SmoothGrad confirma que el modelo Xception utiliza características visuales significativas para realizar sus predicciones. Esto valida que el modelo no solo es preciso, sino también explicablemente sólido, lo que es crucial para su implementación en sistemas de clasificación de residuos. La atención del modelo a las áreas correctas refuerza la confianza en su rendimiento y destaca la efectividad de la arquitectura Xception para manejar datos visualmente complejos. La capacidad de auditar y comprender las decisiones del modelo con SmoothGrad añade un nivel de confianza adicional, lo que es fundamental para aplicaciones prácticas en la gestión de residuos.

### 9.1.6. Análisis y Discusión de Resultados con Imágenes Reales

Las pruebas realizadas con el conjunto de imágenes reales proporcionan una evaluación crucial de la capacidad del modelo Xception para generalizar y realizar clasificaciones precisas en un entorno práctico, donde los residuos aparecen en contextos más complejos y variados en comparación con el set de prueba controlado. Se utilizaron 16 imágenes representativas de diferentes clases de residuos, y el modelo logró clasificar correctamente 15 de ellas, lo que corresponde a una precisión del 93.75 %. A continuación, se analizan los factores clave que explican este rendimiento y se discuten las implicaciones de los casos de éxito y los errores de clasificación observados.

## Desempeño General y Robustez del Modelo

El hecho de que el modelo haya clasificado correctamente el 93.75 % de las imágenes sugiere que la arquitectura Xception es adecuada para manejar datos visualmente complejos, lo cual se debe a su capacidad de extraer y procesar características relevantes mediante convoluciones separables en profundidad. Esta capacidad permite al modelo identificar patrones visuales significativos incluso en imágenes con fondos variados o desordenados.

**Interpretación de Imágenes con Fondos Complejos** Uno de los desafíos más importantes al clasificar imágenes en entornos reales es la presencia de fondos complejos y visualmente distractores, como patrones de textura o elementos no relacionados con el objeto de interés. En las imágenes con fondos complejos, los saliency maps generados con SmoothGrad mostraron que el modelo Xception era capaz de enfocarse en las características críticas del objeto de la clase objetivo, como los contornos, las texturas únicas y las diferencias de color.

Por ejemplo, en una imagen de un residuo plástico colocado sobre un fondo de hierba y hojas secas, el modelo se centró principalmente en las características plásticas del objeto, ignorando eficazmente las distracciones del entorno. Este comportamiento demuestra que el modelo ha logrado un nivel notable de robustez y que la configuración de hiperparámetros y el uso de técnicas de regularización han sido efectivos para mejorar la capacidad de generalización.

### Análisis de Errores de Clasificación

De las 16 imágenes probadas, se observaron dos casos de errores de clasificación:

1. Una imagen de un residuo plástico fue clasificada erróneamente como papel. Esta imagen mostraba un pedazo de plástico sostenido en una mano humana. El saliency map de SmoothGrad para esta imagen reveló que el modelo enfocó parte de su atención en la textura y el color de la mano, lo cual pudo haber contribuido a la confusión con la clase papel. Además, las áreas del plástico que eran brillantes o reflectantes no se destacaron suficientemente como características relevantes, lo que probablemente influyó en el error de clasificación. Este error es consistente con las observaciones de la matriz de confusión, donde la clase *plastic* mostró algunas confusiones con otras clases.

2. El otro caso ya mencionado es el residuo plástico que el modelo clasificó como vidrio, debido a características visuales ambiguas como un brillo translúcido. Aquí, el modelo enfocó su atención en áreas del objeto que podían imitar las propiedades ópticas del vidrio.

**Implicaciones de los Errores** Los errores de clasificación subrayan varias limitaciones inherentes a los modelos de visión por computadora cuando se enfrentan a imágenes reales con propiedades visuales ambivalentes. En el caso del plástico clasificado como papel, la interacción con la mano humana introdujo un sesgo visual, haciendo que el modelo interpretara erróneamente el material. Esto destaca un área clave de mejora, como la necesidad de un preprocesamiento adicional que pueda ayudar a desambiguar características visuales en contextos humanos.

Desde la perspectiva de la matriz de confusión, la clase *plastic* es una de las que presenta más confusiones, lo que ya se había observado en el análisis de las métricas. Este patrón confirma que, a pesar de los buenos resultados generales, la variedad de formas y texturas de los residuos plásticos sigue siendo un desafío para el modelo.

### Validez de las Explicaciones Generadas con SmoothGrad

Los saliency maps con SmoothGrad proporcionaron una herramienta crucial para validar las decisiones del modelo. En la mayoría de los casos, las áreas de atención resaltadas por SmoothGrad correspondieron con las características esperadas para cada clase de residuo, lo que sugiere que el modelo está tomando decisiones razonables y basadas en información relevante. Sin embargo, en los casos de error, se evidenció que el modelo puede ser influenciado por el contexto o características visuales no esenciales.

**Impacto de Fondos Complejos** Otra observación importante es el impacto de los fondos complejos en el rendimiento del modelo. Aunque el modelo mostró una robustez notable al ignorar muchos elementos de fondo, los errores de clasificación indican que el modelo puede ser susceptible a contextos visuales que introducen ambigüedad. Esto subraya la importancia de incluir más ejemplos en el dataset de entrenamiento que representen contextos diversos y escenarios donde las clases de residuos puedan ser confundidas debido a interacciones humanas o propiedades visuales similares.

### Lecciones Aprendidas y Futuras Direcciones

Este análisis sugiere que, aunque el modelo Xception ha demostrado ser robusto y efectivo para la clasificación de residuos en entornos reales, hay margen de mejora en la gestión de casos ambiguos y fondos complicados. Las futuras iteraciones del modelo podrían beneficiarse de la integración de más ejemplos representativos de residuos ambiguos y de técnicas adicionales de Explainable AI que ofrezcan aún más claridad sobre las decisiones del modelo.

Desde una perspectiva de MLOps, estos hallazgos subrayan la importancia de seguir monitoreando y refinando el modelo en producción. La trazabilidad y las explicaciones proporcionadas por SmoothGrad serán herramientas valiosas para realizar ajustes informados y asegurar un rendimiento consistente del modelo a lo largo del tiempo.

#### 9.1.7. Análisis de la Implementación de MLOps con MLFlow

La implementación de MLOps utilizando **MLFlow** en este proyecto fue concebida como una **prueba de concepto (PoC)**, con el objetivo de evaluar la viabilidad y los beneficios potenciales de integrar prácticas de MLOps en el ciclo de vida del modelo de aprendizaje profundo Xception para la clasificación de residuos. Este análisis detalla cómo la configuración y el uso de MLFlow ayudaron a mejorar la gestión del proyecto, con implicaciones futuras que podrían facilitar la escalabilidad y el despliegue eficiente de modelos de Machine Learning.

### Seguimiento de Experimentos y Viabilidad de Implementación

Uno de los aspectos más importantes de esta PoC fue demostrar la capacidad de MLFlow para realizar un seguimiento completo de los experimentos. Esto incluye el registro automático de hiperparámetros, métricas de rendimiento y artefactos de modelos. Durante el desarrollo del modelo Xception, la plataforma MLFlow permitió almacenar y organizar múltiples configuraciones de manera estructurada, lo que facilitó la comparación y selección de los mejores resultados.

Esta trazabilidad no solo es crucial para el desarrollo eficiente, sino que también tiene un impacto significativo en la reproducibilidad de los experimentos, un desafío común en proyectos de IA. La PoC demostró que, si este proyecto se implementa en un entorno de producción o se amplía, MLFlow proporcionaría una base sólida para garantizar que los modelos puedan ser auditados y replicados de manera confiable.

### Gestión de Modelos y Versionamiento: Prueba de Concepto

El versionamiento de modelos es una práctica esencial en MLOps, y en esta PoC se demostró la efectividad de MLFlow para gestionar versiones de manera eficaz. Cada modelo entrenado fue versionado junto con sus métricas de rendimiento, lo que simplifica el proceso de actualización o comparación de modelos. Aunque esta configuración no se llevó a un entorno de producción completo,

la PoC mostró que MLFlow es capaz de soportar un flujo de trabajo robusto, donde las versiones del modelo pueden ser rastreadas y gestionadas con precisión.

Esta funcionalidad es especialmente relevante para proyectos como este, donde la mejora continua del modelo es crucial. La gestión de versiones asegura que los desarrolladores y científicos de datos puedan experimentar con nuevas configuraciones sin perder el contexto histórico ni comprometer la estabilidad del sistema.

### Despliegue Automatizado: Un Paso hacia la Viabilidad en Producción

Aunque el despliegue del modelo Xception fue realizado como una prueba de concepto y no en un entorno de producción completo, el proceso automatizado configurado con MLFlow demostró el potencial de la herramienta para facilitar la transición de modelos desde el desarrollo hasta la producción. La PoC incluyó la configuración de flujos de trabajo que podrían automatizar el despliegue de modelos en servidores de inferencia, lo cual es un aspecto crítico para la implementación futura en sistemas de gestión de residuos en tiempo real.

La automatización y la infraestructura preparada para la integración con MLFlow aseguran que el proyecto esté bien posicionado para futuras escalas y actualizaciones. Esto es particularmente importante en aplicaciones de aprendizaje profundo donde los modelos deben ser regularmente re-entrenados y desplegados de manera eficiente.

### Beneficios Identificados en la PoC

A través de esta prueba de concepto, se destacaron varios beneficios de la integración de MLOps con MLFlow:

- **Trazabilidad Completa:** La capacidad de rastrear cada detalle del experimento proporciona confianza y control sobre el ciclo de vida del modelo. Esto es vital para asegurar la transparencia y la auditabilidad, especialmente en proyectos que pueden estar sujetos a escrutinio por la importancia ambiental y operativa.
- **Reproducibilidad Mejorada:** Con toda la información registrada de manera estructurada, cualquier experimento puede ser replicado de manera exacta. Esto es importante tanto para el desarrollo futuro como para la colaboración entre equipos.
- **Preparación para la Escalabilidad:** Aunque no se ha implementado en producción, el flujo de trabajo diseñado permite que el sistema esté preparado para escalar en un entorno operativo, haciendo que futuras implementaciones sean más sencillas y eficientes.

### Desafíos y Limitaciones de la PoC

La prueba de concepto también reveló algunas limitaciones que deben ser consideradas en futuras implementaciones:

- **Complejidad de Configuración:** La implementación inicial de MLFlow puede ser compleja y requiere una infraestructura adecuada. Para proyectos con recursos limitados, esto podría ser un obstáculo, aunque se justifica en proyectos que buscan tener un impacto a largo plazo.
- **Infraestructura de Soporte:** Un despliegue completo requeriría un backend más robusto y servicios adicionales para soportar las funciones automatizadas de MLFlow, lo que podría aumentar los costos y la complejidad operativa.

## Implicaciones Futuras y Relevancia para MLOps

A pesar de las limitaciones, esta PoC ha validado la importancia de MLOps en proyectos de aprendizaje profundo como este. La infraestructura y las prácticas demostradas con MLFlow aseguran que el proyecto esté listo para una implementación a mayor escala, con la capacidad de gestionar modelos de manera eficiente y mantener una mejora continua. Las técnicas y principios de MLOps aplicados aquí pueden servir como base para proyectos futuros, tanto en la gestión de residuos como en otras aplicaciones de IA donde la precisión, la reproducibilidad y la capacidad de actualización son esenciales.

En resumen, la implementación de MLFlow en este proyecto no solo ha mejorado la organización y el flujo de trabajo, sino que también ha proporcionado una visión clara de cómo un sistema completo de MLOps podría transformar la gestión y el impacto del modelo Xception en aplicaciones del mundo real.

### 9.1.8. Opiniones Finales sobre el Modelo Xception

El modelo Xception ha demostrado un rendimiento sobresaliente en la tarea de clasificación de residuos, alcanzando métricas de precisión y recall altamente satisfactorias. La arquitectura de convoluciones separables en profundidad utilizada por Xception ha mostrado su eficacia al extraer características complejas de imágenes de residuos, lo que ha sido esencial para lograr una capacidad de generalización robusta incluso en conjuntos de datos desafiantes.

El análisis de las métricas de precisión y pérdida ha revelado una estabilidad impresionante durante el entrenamiento y la validación, lo que indica que el modelo está bien ajustado y no sufre de un sobreajuste significativo. La precisión sostenida en el conjunto de validación es una señal de que el modelo es apto para ser desplegado en escenarios reales de clasificación de residuos, y el análisis de la matriz de confusión ha resaltado áreas específicas donde el modelo sobresale, así como las limitaciones que aún persisten.

Las técnicas de Explainable AI, como GradCAM, LIME y SmoothGrad, han proporcionado información crítica sobre cómo el modelo toma sus decisiones. La validación visual de estas explicaciones ha confirmado que Xception se enfoca en las características más relevantes de las imágenes, como texturas, bordes y colores, lo que respalda aún más la fiabilidad del modelo. Sin embargo, también se han identificado casos desafiantes, como la imagen de plástico clasificada erróneamente como papel, lo que subraya la necesidad de seguir investigando y refinando el modelo para manejar mejor las complejidades de ciertos escenarios.

El uso de MLFlow para una prueba de concepto ha subrayado la importancia de las prácticas de MLOps en el desarrollo de modelos de aprendizaje profundo. Aunque esta implementación fue limitada, ha demostrado cómo las herramientas de gestión y versionado de modelos pueden optimizar el flujo de trabajo y garantizar que las futuras iteraciones del sistema sean más manejables y escalables. Esto establece una base prometedora para una mayor integración de MLOps en futuros desarrollos.

Como conclusión, el modelo Xception ha cumplido con las expectativas en términos de precisión y capacidad de generalización, y el uso de técnicas de Explainable AI ha agregado un nivel valioso de transparencia y comprensión. Aunque hay áreas que podrían beneficiarse de un mayor refinamiento, los resultados actuales sugieren que este modelo es una herramienta poderosa para abordar problemas de clasificación de residuos, con un alto potencial para futuras aplicaciones y mejoras.



## 9.2. Análisis de Resultados del Chatbot Inteligente

En esta sección, se presenta un análisis exhaustivo del rendimiento del chatbot inteligente, que fue implementado utilizando la API de OpenAI. El chatbot fue diseñado para asistir a los ciudadanos de la Ciudad de Guatemala en la clasificación de residuos, proporcionando respuestas precisas y relevantes basadas en la información disponible sobre el manejo y reciclaje de residuos.

### 9.2.1. Evaluación de la Precisión y Relevancia de las Respuestas

La precisión y relevancia de las respuestas del chatbot fueron evaluadas mediante una encuesta realizada a expertos en gestión ambiental, así como por pruebas directas en escenarios simulados con diversas preguntas. Las evaluaciones indicaron que el chatbot es altamente efectivo en proporcionar información útil, aunque ciertas limitaciones fueron observadas en casos específicos.

**Precisión de las Respuestas** La precisión fue uno de los aspectos más valorados por los expertos, con un 57.1 % otorgando una calificación de 5, lo que indica que las respuestas fueron completamente precisas y claras. El 42.9 % restante asignó una puntuación de 4, señalando que las respuestas eran generalmente correctas pero con margen de mejora en ciertos detalles o explicaciones.

El análisis del rendimiento del chatbot destaca la capacidad de la API de OpenAI para entender y procesar consultas complejas en lenguaje natural. Esto es posible gracias al extenso entrenamiento del modelo de lenguaje subyacente, que ha sido optimizado para la comprensión semántica y la generación de respuestas contextualmente apropiadas. Sin embargo, el modelo también presenta limitaciones inherentes a los modelos de lenguaje, como potenciales malentendidos de consultas ambiguas o la incapacidad de proporcionar respuestas completamente precisas cuando la información es escasa o no está directamente disponible.

**Relevancia Temática** El chatbot fue evaluado en su capacidad para proporcionar respuestas alineadas con el tema de clasificación de residuos. Los expertos calificaron este criterio con un 57.1 % de 5 puntos y un 42.9 % de 4 puntos, lo que muestra un alto grado de satisfacción con la relevancia de las respuestas. Esto valida la eficacia de la implementación para mantenerse enfocado en el dominio ambiental, un logro considerable dado el amplio conocimiento general que la API de OpenAI tiene en múltiples temas.

El éxito en este aspecto se debe a la implementación de estrategias de orientación temática que guían las respuestas del chatbot hacia un enfoque más específico, mitigando el riesgo de divagaciones hacia temas irrelevantes. La relevancia de las respuestas fue lograda a través de un entrenamiento y diseño cuidadoso de las instrucciones proporcionadas al modelo, que lo mantienen dentro del ámbito específico de la clasificación de residuos.

### 9.2.2. Capacidad para Rechazar Preguntas Fuera de Tema

El chatbot mostró una capacidad impresionante para manejar preguntas fuera de tema. En este criterio, el 85.7 % de los expertos calificó esta habilidad con un 5, destacando la cortesía y claridad con la que el chatbot rechaza consultas no relacionadas. Solo un 14.3 % de los evaluadores otorgó una puntuación de 4.

Esta funcionalidad es crítica para mantener la utilidad del chatbot dentro de su dominio específico, y fue implementada utilizando instrucciones claras que indican al modelo cuándo y cómo rechazar preguntas. La capacidad del modelo para rechazar adecuadamente preguntas no relacionadas es un

reflejo de las habilidades de comprensión contextual de la API de OpenAI, las cuales permiten al chatbot identificar cuándo una consulta no se ajusta al ámbito de la gestión de residuos y responder de manera educada pero firme.

**Implicaciones de MLOps y Futuro Despliegue** Aunque el chatbot fue evaluado en un entorno controlado y sin un despliegue completo en un entorno de producción, los principios de MLOps también pueden aplicarse aquí para mejorar el mantenimiento y la mejora continua del sistema. En futuras iteraciones, la integración con herramientas de MLOps podría permitir un seguimiento en tiempo real del rendimiento del chatbot, versionado de actualizaciones de modelos, y pruebas A/B para optimizar las respuestas.

### 9.2.3. Calidad de las Recomendaciones

El chatbot también fue evaluado por su capacidad para proporcionar recomendaciones útiles sobre el manejo de residuos. La calidad de estas recomendaciones fue bien recibida, con un 71.4 % de los expertos otorgando una puntuación de 4 y un 28.6 % calificando con un 5. Este resultado sugiere que, aunque las recomendaciones fueron útiles y claras, hay margen para mejorar en términos de detalle y especificidad.

La API de OpenAI ha sido entrenada con una vasta cantidad de datos textuales, lo que le permite proporcionar recomendaciones bien fundamentadas. Sin embargo, la calidad de estas respuestas también depende de la especificidad de la información proporcionada en las instrucciones iniciales al chatbot y del contexto en el que se utilizan. Esto resalta la importancia de la continua mejora y refinamiento de las instrucciones para garantizar que el chatbot siga proporcionando valor a los usuarios.

### 9.2.4. Consistencia en el Estilo de las Respuestas

La consistencia en el estilo de las respuestas también fue altamente valorada, con un 57.1 % de los expertos calificando con un 5 y un 42.9 % con un 4. Esto demuestra que el chatbot mantiene un tono claro, educado y profesional en sus interacciones, lo cual es esencial para la confianza del usuario y la facilidad de uso.

El estilo de las respuestas fue diseñado para ser amigable y accesible, sin perder la formalidad necesaria para un asistente ambiental. Esta consistencia en el tono se debe a la configuración de la API de OpenAI, que permite definir el estilo de comunicación del modelo. Mantener un estilo consistente es importante para que los usuarios se sientan cómodos y seguros al interactuar con el chatbot, lo que podría ser un factor decisivo para la adopción del sistema.

### 9.2.5. Evaluación del Tiempo de Respuesta

El tiempo de respuesta promedio del chatbot fue de 1.7 segundos, como se muestra en la evaluación. Este tiempo es adecuado para la mayoría de las aplicaciones de asistencia en línea, proporcionando una experiencia de usuario fluida y eficiente. El tiempo de respuesta es un factor crucial, especialmente en entornos en los que los usuarios esperan respuestas rápidas y precisas.

El rendimiento en términos de tiempo es un testimonio de la optimización subyacente de la API de OpenAI y la infraestructura utilizada. La rapidez del chatbot no solo es importante para la experiencia del usuario, sino también para la escalabilidad del sistema si se implementa en un entorno de producción con un gran número de usuarios simultáneos.

### 9.2.6. Pruebas y Ejemplos de Interacciones

Se realizaron pruebas prácticas al chatbot para evaluar su capacidad de proporcionar respuestas precisas y relevantes a preguntas frecuentes sobre el manejo de residuos. A continuación, se muestran algunos ejemplos de las interacciones y análisis de las respuestas generadas:

Tabla 9.1: Ejemplos de Interacciones con el Chatbot y su Análisis

Pregunta	Respuesta del Chatbot	Tiempo de Respuesta
¿Que puedo hacer con mis residuos plásticos?	Puedes clasificar tus residuos plásticos como reciclables y llevarlos a un centro de reciclaje adecuado en la Ciudad de Guatemala...	2.14 segundos
¿Cómo debo clasificar los residuos orgánicos?	Los residuos orgánicos deben ser clasificados en la categoría de "Orgánicos" (Verde)...	1.70 segundos
¿Dónde puedo reciclar el vidrio en la Ciudad de Guatemala?	En la Ciudad de Guatemala, puedes reciclar el vidrio en el centro de reciclaje Interfisa...	1.64 segundos
¿Cuáles son las reglas para desechar desechos electrónicos?	Las normas para desechar desechos electrónicos en la Ciudad de Guatemala implican no mezclarlos con la basura común...	2.21 segundos
¿Cuál es la capital de Francia?	Lo siento, solo puedo responder preguntas relacionadas con la clasificación y manejo de residuos...	0.89 segundos

El análisis de estas interacciones revela que el chatbot es eficaz en proporcionar respuestas claras y útiles dentro de su ámbito de conocimiento. La capacidad de rechazar preguntas fuera de tema también ha sido implementada con éxito, demostrando que el chatbot entiende el alcance de su funcionalidad.

### 9.2.7. Opiniones Finales sobre el Desempeño del Chatbot

En general, el chatbot ha demostrado ser una herramienta valiosa para la clasificación de residuos, con respuestas precisas, relevantes y consistentes. La implementación de la API de OpenAI ha proporcionado un rendimiento sólido, aunque el sistema puede beneficiarse de actualizaciones y optimizaciones continuas basadas en el feedback de los usuarios y nuevas necesidades que puedan surgir.

El éxito del chatbot en esta etapa inicial valida la viabilidad de su uso a gran escala y sugiere un alto potencial para futuras aplicaciones en la gestión de residuos y otros ámbitos relacionados con la sostenibilidad y el medio ambiente.

## 9.3. Implicaciones de los Agentes Inteligentes en la Ciudad de Guatemala

La implementación de los agentes inteligentes desarrollados en este proyecto, el modelo Xception para la clasificación automática de residuos y el chatbot inteligente para la asistencia en la gestión de residuos, tiene un impacto significativo y potencialmente transformador en la Ciudad de Guatemala. Estas herramientas fueron diseñadas para abordar desafíos específicos en la educación y concienciación ambiental, y en esta sección se analizan las implicaciones prácticas, sociales y medioambientales

de su aplicación en la vida cotidiana de los habitantes de la ciudad.

### 9.3.1. Facilitando la Gestión Eficiente de Residuos

La gestión de residuos es uno de los mayores retos ambientales para la Ciudad de Guatemala, donde la falta de sistemas de reciclaje bien implementados y el desconocimiento de las prácticas sostenibles contribuyen a la contaminación y al deterioro de la calidad de vida. El modelo Xception ofrece una solución práctica para abordar este problema, ya que permite la clasificación precisa y rápida de residuos en seis categorías distintas: plástico, papel, vidrio, metal, orgánicos y desechos electrónicos. Este modelo puede integrarse en sistemas automatizados de separación de residuos en centros de reciclaje, mejorando la eficiencia y reduciendo la carga de trabajo manual.

El uso de técnicas de Explainable AI como GradCAM y SmoothGrad proporciona un nivel de transparencia esencial para ganar la confianza de los usuarios. Las autoridades municipales y los encargados de la gestión de residuos pueden ver y entender las razones detrás de las decisiones del modelo, lo que facilita su aceptación y uso en sistemas de reciclaje y clasificación de residuos.

### 9.3.2. Educación y Concienciación Ambiental a través del Chatbot Inteligente

El chatbot inteligente desarrollado para este proyecto representa una herramienta valiosa para educar y guiar a los ciudadanos sobre las mejores prácticas en la gestión de residuos. Al proporcionar respuestas precisas y relevantes en tiempo real, el chatbot puede integrarse en una aplicación móvil destinada a la ciudadanía, permitiendo un acceso sencillo a la información sobre reciclaje y sostenibilidad.

En la vida cotidiana, esto significa que cualquier residente de la Ciudad de Guatemala, independientemente de su nivel de conocimiento previo, puede obtener asesoramiento rápido sobre cómo clasificar sus residuos correctamente. La capacidad del chatbot para manejar consultas relacionadas con la ubicación de centros de reciclaje, las normas de desecho de residuos electrónicos y las mejores prácticas para el manejo de materiales reciclables promueve una cultura de responsabilidad ambiental.

### 9.3.3. Promoviendo Hábitos Sostenibles en la Población

La combinación de estos dos agentes inteligentes tiene el potencial de generar un cambio profundo en los hábitos de los habitantes de la Ciudad de Guatemala. La disponibilidad de un sistema automatizado de clasificación, junto con un asesor virtual accesible, reduce las barreras para la adopción de prácticas sostenibles. Las personas pueden sentirse más empoderadas y seguras al saber que están clasificando sus residuos correctamente y contribuyendo al bienestar del medio ambiente.

Además, estas herramientas pueden ser implementadas en campañas de educación ambiental dirigidas por la Municipalidad, en escuelas, universidades y comunidades. Al tener una herramienta educativa y funcional que facilita la clasificación de residuos y fomenta la toma de decisiones informadas, se pueden alcanzar de manera más efectiva los objetivos de reducción de residuos y reciclaje establecidos por las autoridades locales.

### 9.3.4. Sostenibilidad y Futuro de la Gestión de Residuos

El desarrollo de estos agentes inteligentes también tiene implicaciones positivas para el futuro de la sostenibilidad en la Ciudad de Guatemala. Con la creciente urbanización y la presión sobre los recursos naturales, es esencial contar con soluciones escalables y eficientes. La implementación de MLOps mediante MLFlow, aunque en este proyecto fue un prueba de concepto (PoC), sugiere que es posible gestionar el ciclo de vida de estos modelos de manera eficiente, facilitando futuras mejoras y adaptaciones.

A largo plazo, estas herramientas podrían integrarse en un sistema de gestión de residuos basado en IoT, donde sensores y dispositivos automatizados trabajen en conjunto con los modelos de IA para optimizar la recolección y el tratamiento de residuos. Esto no solo mejoraría la eficiencia, sino que también ayudaría a la ciudad a cumplir con los objetivos de sostenibilidad y a reducir su huella ecológica.

### 9.3.5. Desafíos y Consideraciones Finales

Si bien las implicaciones de estos agentes inteligentes son prometedoras, es importante considerar ciertos desafíos. La aceptación y adopción de estas herramientas por parte de la población dependerán en gran medida de la educación y sensibilización. Además, la infraestructura tecnológica de la ciudad debe estar preparada para soportar la integración de estos sistemas.

Otro factor crítico es la necesidad de políticas públicas que respalden la implementación de estas tecnologías, así como la inversión en infraestructura de reciclaje y programas de educación ambiental. La colaboración entre la Municipalidad, empresas privadas y organizaciones comunitarias será fundamental para asegurar el éxito de estas iniciativas.

Por lo tanto, los agentes inteligentes desarrollados en este proyecto se perfilan como una herramienta poderosa para transformar la gestión de residuos en la Ciudad de Guatemala, fomentando un cambio positivo en los hábitos de los ciudadanos y contribuyendo significativamente al bienestar del medio ambiente. La implementación de estas soluciones representa una vía prometedora hacia un futuro más sostenible, donde tecnología e innovación se conjugan para mejorar la calidad de vida urbana y proteger nuestro entorno.

A partir de la implementación y análisis detallado de los agentes inteligentes desarrollados en este proyecto, se pueden destacar las siguientes conclusiones:

- El modelo de clasificación de residuos basado en la arquitectura Xception logró un desempeño sobresaliente, alcanzando una precisión promedio del 98 % en el conjunto de prueba. Esto valida que la elección de la arquitectura Xception fue adecuada para abordar la complejidad y variabilidad visual de los residuos urbanos, particularmente en el contexto de la Ciudad de Guatemala.
- La integración de técnicas de *Explainable AI* (GradCAM, LIME y SmoothGrad) permitió una comprensión profunda de cómo el modelo toma decisiones, aumentando la transparencia y confianza en el sistema. Estos métodos confirmaron que el modelo se enfoca en las características más relevantes de los residuos, validando su efectividad en escenarios con fondos simples y complejos.
- El análisis y validación del modelo en imágenes reales evidenció su robustez y capacidad de generalización en entornos no controlados. Aunque se observó un error de clasificación en una imagen con un contexto visual desafiante, el desempeño general se mantuvo alto, resaltando la aplicabilidad práctica del sistema.
- La implementación de prácticas de MLOps con MLFlow, aunque a nivel de prueba de concepto (PoC), demostró la viabilidad y la importancia de la gestión y seguimiento del ciclo de vida del modelo. Este enfoque asegura que futuros desarrollos y optimizaciones puedan realizarse de manera eficiente, facilitando el versionamiento y la trazabilidad de experimentos.
- El chatbot inteligente basado en la API de OpenAI mostró un desempeño altamente efectivo en términos de precisión, relevancia temática y tiempo de respuesta. La capacidad del chatbot para rechazar preguntas fuera de tema y proporcionar recomendaciones útiles respalda su potencial como herramienta educativa y de asistencia para la ciudadanía en temas de gestión de residuos.
- Las pruebas realizadas con expertos de la Unidad de Control Ambiental de la Municipalidad de la Ciudad de Guatemala validaron la eficacia y relevancia de ambas soluciones. Los agentes inteligentes fueron bien recibidos y se consideraron como herramientas valiosas para fomentar hábitos sostenibles y mejorar la gestión de residuos en la ciudad.

- La implementación de estos agentes inteligentes ofrece una solución viable y sostenible para la Ciudad de Guatemala, alineada con los objetivos de promover la educación ambiental y optimizar el manejo de residuos. La combinación de tecnologías avanzadas y un enfoque práctico crea un impacto positivo tanto en la conciencia ciudadana como en la infraestructura urbana.
- Se cumplieron los objetivos planteados al inicio del proyecto, logrando desarrollar un sistema integral que combina modelos de visión por computadora y un chatbot de asistencia con aplicaciones prácticas. La solución propuesta no solo es técnicamente sólida sino también escalable y adaptable a diferentes entornos, ofreciendo un marco que puede extenderse a otras ciudades con necesidades similares.

- Se recomienda continuar optimizando el modelo de clasificación Xception mediante la experimentación con técnicas avanzadas de *data augmentation* y el ajuste fino de los hiperparámetros. Además, explorar el uso de arquitecturas híbridas y aplicar transferencia de aprendizaje con modelos preentrenados en datasets más amplios puede mejorar la capacidad del modelo para generalizar, especialmente cuando se enfrenta a escenarios diversos y complejos en la clasificación de residuos.
- Ampliar el dataset con imágenes adicionales, particularmente aquellas capturadas en entornos reales con fondos urbanos complejos y condiciones de iluminación variables, es fundamental. Esto también debería incluir situaciones donde los residuos están parcialmente ocultos o en posiciones inusuales, lo que incrementará la robustez del modelo frente a escenarios prácticos y mejorará su precisión general.
- Aunque la implementación de MLOps con MLFlow se desarrolló como una prueba de concepto, se recomienda establecer un entorno de MLOps completo. Esto incluiría la automatización del ciclo de vida del modelo, el despliegue continuo, el monitoreo en tiempo real y la gestión de versiones. La adición de mecanismos para detectar el *drift* de datos y realizar *rollbacks* rápidamente asegurará la estabilidad y el rendimiento óptimo del sistema a lo largo del tiempo.
- Para el chatbot inteligente, sería beneficioso extender sus funcionalidades. Implementar soporte multilingüe y adaptar las respuestas en función de la ubicación geográfica de los usuarios dentro de la Ciudad de Guatemala aumentará la accesibilidad y relevancia del servicio. Además, desarrollar un módulo de aprendizaje continuo, que permita al chatbot mejorar sus respuestas a partir de la interacción y retroalimentación de los usuarios, fortalecerá la precisión y la utilidad del asistente.
- La educación y sensibilización de la población son clave para el éxito de estas herramientas. Se sugiere promover campañas educativas enfocadas en el correcto uso de las tecnologías propuestas y la importancia de la gestión adecuada de residuos. Estas iniciativas deben involucrar escuelas, comunidades y organizaciones locales para fomentar hábitos sostenibles y garantizar que la población aproveche al máximo las herramientas desarrolladas.
- Se recomienda fomentar la colaboración con la Municipalidad de la Ciudad de Guatemala y entidades privadas para asegurar un soporte adecuado en la implementación de estas soluciones tecnológicas. La creación de políticas públicas que impulsen el uso de tecnologías de inteligencia artificial en la gestión de residuos, junto con la inversión en infraestructura de reciclaje, es



esencial para el éxito a largo plazo del proyecto. Estas alianzas facilitarán la integración fluida de los sistemas en la vida urbana.

- Realizar evaluaciones periódicas del impacto de los agentes inteligentes en la gestión de residuos y en los hábitos de la población proporcionará datos valiosos para mejorar continuamente. Indicadores como el aumento en la cantidad de residuos correctamente clasificados, la reducción de desechos no reciclables y la satisfacción de los usuarios serán fundamentales para ajustar las herramientas según sea necesario y maximizar su efectividad.
- Se debe seguir mejorando la interpretabilidad del modelo de clasificación Xception mediante la investigación y la implementación de técnicas avanzadas de *Explainable AI*. Una mejor transparencia en las decisiones del modelo no solo aumentará la confianza de los usuarios, sino que también permitirá a los desarrolladores identificar y corregir posibles áreas problemáticas en el sistema.
- Finalmente, realizar pruebas adicionales en condiciones adversas, como ambientes con iluminación cambiante o la presencia de objetos distractores, garantizará que el modelo mantenga un desempeño robusto y confiable. También se podría considerar la expansión del proyecto a otras ciudades con desafíos similares en la gestión de residuos, dado que las soluciones aquí desarrolladas son flexibles y pueden adaptarse a diferentes contextos urbanos con éxito.

- [1] Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu y Xiaoqiang Zheng: *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. Google Research, 2016. <https://www.tensorflow.org/>.
- [2] Adadi, Amina y Mohamed Berrada: *Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)*. IEEE Access, 6:52138–52160, 2018.
- [3] AGN: *Guatemala en el Primer Foro de Ética de la Inteligencia Artificial*. 2023. <https://agn.gt/guatemala-en-el-primer-foro-de-etica-de-la-inteligencia-artificial/>.
- [4] Ahmed, M. I. B., R. B. Alotaibi, R. A. Al-Qahtani, R. S. Al-Qahtani, S. S. Al-Hetela, K. A. Al-Matar, N. K. Al-Saqer, A. Rahman, L. Saraireh, M. Youldash y G. Krishnasamy: *Deep learning for recyclable product classification: Enhancing efficiency and sustainability in waste management*. Journal of Cleaner Production, 2023. <https://www.semanticscholar.org/paper/c19b7f207a9462aae2658d9875d5149273d74cc5>.
- [5] Author, Jane: *A Primer on MLOps: Machine Learning Operations*. Tech Journal of AI, 2021.
- [6] Breck, Eric, Shanjing Cai, Michael Nielsen, Michael Salib y D. Sculley: *The ML Test Score: A Rubric for ML Production Readiness and Technical Debt Reduction*. En *Proceedings of the 2017 Conference on Systems for ML*, 2017.
- [7] Brown, Tom B. y cols.: *Language Models are Few-Shot Learners*. Advances in Neural Information Processing Systems, 33:1877–1901, 2020.
- [8] Brown, Tom B, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever y Dario Amodei: *GPT-3: Language Models are Few-Shot Learners*. OpenAI Research, 2023. <https://www.openai.com/research/gpt-3>.
- [9] Centroamérica, Walmart: *Nace la Alianza por el Reciclaje con el fin de impulsar prácticas sostenibles en Guatemala*. February 2024. <https://www.walmartcentroamerica.com/noticias/2024/02/nace-alianza-por-el-reciclaje-guatemala>.

- [10] Chollet, François: *Xception: Deep Learning with Depthwise Separable Convolutions*. En *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, páginas 1251–1258. IEEE, 2017.
- [11] Clark, Alex y contributors: *Pillow: Python Imaging Library (Fork)*. <https://python-pillow.org/>, 2015. Available at: <https://python-pillow.org/>.
- [12] Clark, Alex y Contributors: *Pillow: The friendly PIL fork*. Python Imaging Library (PIL), 2021. Available at: <https://pillow.readthedocs.io/>.
- [13] Cruz, Marcos y Luis Vega: *WasteNet: A Smart Waste Management System for Smart Cities*. IEEE Access, 6:14594–14605, 2018.
- [14] CS-UCSP: *Diplomado en Visión por Computador con Machine Learning y Deep Learning*, 2024. <https://cs.ucsp.edu.pe/postgrado/diplomado-vision-computador-con-machine-learning-deep-learning/>.
- [15] CS182, Harvard: *Bias, Variance, and Regularization - CS 182*, 2021. <https://cs182sp21.github.io/static/slides/lec-3.pdf>.
- [16] Data, Waste Classification: *Waste Classification Data: Dataset de Residuos*, 2023. <https://www.kaggle.com/datasets/alexattia/the-simpsons-characters-dataset>.
- [17] Databricks: *MLFlow Documentation*. <https://mlflow.org/docs>, 2022.
- [18] Databricks: *MLflow: A Platform for the Machine Learning Lifecycle*, 2023. <https://mlflow.org/>.
- [19] DataScienceCentral: *Bias Variance Trade Off*, 2019. <https://www.datasciencecentral.com/bias-variance-trade-off/>.
- [20] DataScienceCentral: *Bias Variance Trade Off - DataScienceCentral.com*, 2019. <https://www.datasciencecentral.com/bias-variance-trade-off/>.
- [21] Deng, Jia, Wei Dong, Richard Socher, Li Jia Li, Kai Li y Li Fei-Fei: *ImageNet: A large-scale hierarchical image database*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, páginas 248–255, 2009.
- [22] Development, Sustainable: *Gestión de Residuos*. Informe técnico, 2023. [https://sustainabledevelopment.un.org/content/documents/dsd/dsd\\_aofw\\_ni/ni\\_pdfs/NationalReports/guatemala/waste.pdf](https://sustainabledevelopment.un.org/content/documents/dsd/dsd_aofw_ni/ni_pdfs/NationalReports/guatemala/waste.pdf).
- [23] DoradoReciclaAI: *Sistema para clasificación de residuos: Project Recicla AI*, 2023. [https://github.com/DoradoReciclaAI/Project\\_recicla\\_ai](https://github.com/DoradoReciclaAI/Project_recicla_ai).
- [24] Ellen MacArthur Foundation: *Circular Economy: A Wealth of Flows*. Ellen MacArthur Foundation, 2019. <https://www.ellenmacarthurfoundation.org/publications/circular-economy-a-wealth-of-flows>.
- [25] Europea, Unión: *Reglamento de Inteligencia Artificial de la UE*. 2021. <https://eur-lex.europa.eu/legal-content/ES/TXT/?uri=CELEX%3A52021PC0206>.
- [26] European Commission: *A European Strategy for Plastics in a Circular Economy*. Informe técnico, European Commission, 2018. <https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52018DC0028&from=EN>.
- [27] Face, Hugging: *Implementación de Modelos NLP con Hugging Face*, 2023. <https://huggingface.co/transformers/>.

- [28] Fandiño, Juan: *Desarrollo de una aplicación web para la clasificación de residuos sólidos a través de un modelo de machine learning*, 2022. <https://repository.universidadean.edu.co/bitstream/handle/10882/12381/FandinoJuan2022.pdf?sequence=1&isAllowed=y>.
- [29] FAO: *Transforming Food and Agriculture to Achieve the SDGs*. Informe técnico, Food and Agriculture Organization of the United Nations, 2018. <http://www.fao.org/3/I9900EN/i9900en.pdf>.
- [30] González, Laura: *EcoSort: Optimización de rutas de recolección de residuos usando machine learning*, 2019. <https://repositorio.uam.es/handle/10486/684402>.
- [31] Goodfellow, Ian, Yoshua Bengio y Aaron Courville: *Deep Learning*. MIT Press, 2016.
- [32] Google: *Recursos y Ventajas de Google Colab Pro*, 2023. <https://colab.research.google.com/signup>.
- [33] Gordillo, L. y C. F. Presa: *Guatemala - new regulation for waste management*. Informe técnico, Lexology, August 17 2021. <https://www.lexology.com/library/detail.aspx?g=6cac48fb-a179-4857-b257-38a07010efa3>.
- [34] Government, US: *Guidelines for the Regulation of Artificial Intelligence*. 2021. <https://www.whitehouse.gov/ai>.
- [35] Gremiagro: *Reglamento para la Gestión Integral de los Residuos y Desechos Sólidos Comunes*. August 2023. <https://gremiagro.com/wp-content/uploads/2023/08/AG-164-2021-Reglamento-para-la-Gestion-Integral-de-Residuos-y-Desechos-Solidos-Comunes.pdf>.
- [36] GSMA: *eWaste en América Latina*. 2015. <https://www.gsma.com/latinamerica/wp-content/uploads/2015/11/gsma-unu-ewaste2015-spa.pdf>.
- [37] Guatemala, Congreso de la República de: *Ley de Protección y Mejoramiento del Medio Ambiente (Decreto 68-86)*. Gobierno de Guatemala, 1986.
- [38] Guatemala, Congreso de la República de: *Ley de Protección de Datos Personales (Decreto 25-2012)*, 2012. <https://www.congreso.gob.gt/>.
- [39] Guatemala, Gobierno de: *Acuerdo Gubernativo 164-2021*. Diario de Centro América, 2021. En línea.
- [40] Guatemala, Gobierno de: *Reglamento para la Gestión Integral de los Residuos y Desechos Sólidos Comunes (Acuerdo Gubernativo 164-2021)*. Gobierno de Guatemala, 2021.
- [41] Guatemala (USAC), Universidad de San Carlos de: *Boletín Economía al Día*. September 2022. <https://iies.usac.edu.gt/wp-content/uploads/2022/10/Boletin-Economia-al-dia-No.-9-septiembre-2022.pdf>.
- [42] Gunning, David y David Aha: *DARPA's explainable artificial intelligence (XAI) program*. AI Magazine, 40(2):44–58, 2019.
- [43] Hajnas, Mariam y Ali Alqurashi: *Waste Classification Using Deep Learning for Waste Management in Smart Cities*. IEEE Access, 8:17506–17518, 2020.
- [44] (HPE), Hewlett Packard Enterprise: *¿Qué son los grandes modelos de lenguaje?*, 2023. <https://www.hpe.com/lamerica/es/what-is-large-language-model.html>.
- [45] Huang, Gao, Zhuang Liu, Laurens Van Der Maaten y Kilian Q. Weinberger: *Densely connected convolutional networks*. En *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 4700–4708, 2017.

- [46] Imaging, STEMMER: *Ventajas del Machine Learning y Deep Learning para la visión por computadora*, 2024. <https://es.stemmer-imaging.com/blog/tecnologia-tendencias/machine-learning-deep-learning/>.
- [47] InboundCycle: *Qué es el deep learning, cómo funciona y sus aplicaciones*, 2024. <https://www.inboundcycle.com/blog-de-inbound-marketing/deep-learning-que-es>.
- [48] Institute, SAS: *Computer Vision: Qué es y por qué es importante*, 2024. [https://www.sas.com/es\\_mx/insights/analytics/computer-vision.html](https://www.sas.com/es_mx/insights/analytics/computer-vision.html).
- [49] Ioffe, Sergey y Christian Szegedy: *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. En *International Conference on Machine Learning*, páginas 448–456, 2015.
- [50] IPNUSAC: *Análisis de la generación de desechos sólidos en Guatemala*. Informe técnico, Instituto de Problemas Nacionales de la Universidad de San Carlos de Guatemala (IPNUSAC), 2023. <https://ipn.usac.edu.gt/wp-content/uploads/2023/09/Analisis-de-la-generacion-de-desechos-solidos-en-Guatemala.pdf>.
- [51] (ITU), International Telecommunication Union: *Gestión Sostenible de Residuos de Aparatos Eléctricos y Electrónicos*. Informe técnico, 2020. [https://www.itu.int/dms\\_pub/itu-t/oth/0b/11/T0B110000273301PDFS.pdf](https://www.itu.int/dms_pub/itu-t/oth/0b/11/T0B110000273301PDFS.pdf).
- [52] Jupyter: *Jupyter Notebooks para Ciencia de Datos*, 2023. <https://jupyter.org/>.
- [53] Kaggle: *Explora y domina Kaggle para Machine Learning*, 2023. <https://www.kaggle.com/learn/overview>.
- [54] Keras: *Keras: La API de Alto Nivel de TensorFlow*, 2023. <https://keras.io/>.
- [55] Kingma, Diederik P y Jimmy Lei Ba: *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980, 2014.
- [56] Legal, BLP: *The pending solution for garbage disposal in Guatemala*. October 25 2023. <https://blplegal.com/regulatory-harmonization-the-pending-solution-for-garbage-disposal-in-guatemala/>.
- [57] Lundberg, Scott M y Su In Lee: *A unified approach to interpreting model predictions*. Advances in Neural Information Processing Systems, 30:4765–4774, 2017.
- [58] Marin, Andrea y cols.: *Predicting Urban Waste Generation with Machine Learning*. Waste Management, 116:170–182, 2021.
- [59] Marín, Agustín: *Analítica para una gestión de residuos inteligente*, 2020. <https://blogs.sas.com/content/hiddeninsights/2020/07/13/modelos-analiticos-para-dotar-de-inteligencia-a-la-gestion-de-residuos/>.
- [60] Masters, D. y C. Luschi: *Revisiting small batch training for deep neural networks*. En *arXiv preprint arXiv:1804.07612*, 2018.
- [61] Medium: *Intro to Deep Learning — Bias-Variance tradeoff, Regularizations*, 2021. <https://hk3342.medium.com/intro-to-deep-learning-bias-variance-tradeoff-regularizations-f0e7b7b4a4f2>.
- [62] Medium: *Modelos de ChatGPT y LLM (Modelos de lenguaje grande)*, 2023. <https://medium.com/chatgpt-learning/modelos-de-chatgpt-y-llm-modelos-de-lenguaje-grande-d8c5fc39f59e>.
- [63] Ministerio de Ambiente y Recursos Naturales (MARN): *Política Nacional de Residuos Sólidos*. Gobierno de Guatemala, 2010.

- [64] Ng, Andrew Y: *Feature selection, L1 vs. L2 regularization, and rotational invariance*. Proceedings of the 21st International Conference on Machine Learning, 2004.
- [65] NVIDIA: *CUDA y el Aceleramiento de Deep Learning*, 2023. <https://developer.nvidia.com/cuda-zone>.
- [66] OpenAI: *OpenAI API Documentation*, 2020. OpenAI.
- [67] OpenCV: *OpenCV para Visión por Computadora*, 2023. <https://opencv.org/>.
- [68] Owens, John D, Michael Houston, David Luebke, Simon Green, John E Stone y James C Phillips: *GPU computing*. Proceedings of the IEEE, 96(5):879–899, 2008.
- [69] Pan, Sinno Jialin y Qiang Yang: *A survey on transfer learning*. IEEE Transactions on Knowledge and Data Engineering, 22(10):1345–1359, 2009.
- [70] Pocasangre, H.: *Educación ambiental para conservar el patrimonio natural de Guatemala*. Informe técnico, Ministerio de Ambiente y Recursos Naturales (MARN), January 26 2022. <https://www.marn.gob.gt/educacion-ambiental-para-conservar-el-patrimonio-natural-de-guatemala/>.
- [71] Polyzotis, Neoklis, Surajit Roy, Samuel Whang y Martin Zinkevich: *Data Management Challenges in Production Machine Learning: A Survey*. ACM Data Engineering Bulletin, 2018.
- [72] Poudel, S. y P. Poudyal: *Classification of waste materials using CNN based on transfer learning*. En *Proceedings of the 14th Annual Meeting of the ACM India Special Interest Group on Computer Science Education*, páginas 37–44. Association for Computing Machinery, 2022. <https://dl.acm.org/doi/10.1145/3574318.3574345>.
- [73] Prechelt, L.: *Early stopping—but when?* Neural Networks: Tricks of the trade, páginas 55–69, 1998.
- [74] PyTorch: *Tutorial de PyTorch*, 2023. <https://pytorch.org/tutorials/>.
- [75] Ramirez, Sebastian: *FastAPI: The high-performance web framework for building APIs with Python 3.7+*. 2021. Available at: <https://fastapi.tiangolo.com/>.
- [76] Ramirez, Sebastián: *FastAPI: The High Performance Web Framework for Building APIs with Python 3.6+ Based on Standard Python Type Hints*. FastAPI Documentation, 2020. Available at: <https://fastapi.tiangolo.com/>.
- [77] RDS: *Análisis de políticas públicas para el reciclaje inclusivo en América Latina*. 2017. [https://rds.org.co/apc-aa-files/ba03645a7c069b5ed406f13122a61c07/170331\\_app\\_reciclajeinclusivola-6.pdf](https://rds.org.co/apc-aa-files/ba03645a7c069b5ed406f13122a61c07/170331_app_reciclajeinclusivola-6.pdf).
- [78] ReciclaLos: *Guía Práctica para la Formulación de Planes Municipales para la Gestión Integral de Residuos Sólidos en Guatemala*. 2023. [https://reciclaLos.com/assets/downloads/Guia\\_Planes\\_Municipales\\_para\\_la\\_GIRDS.pdf](https://reciclaLos.com/assets/downloads/Guia_Planes_Municipales_para_la_GIRDS.pdf).
- [79] Ribeiro, Marco Tulio, Sameer Singh y Carlos Guestrin: *Why should I trust you? Explaining the predictions of any classifier*. páginas 1135–1144, 2016.
- [80] Ronacher, Armin: *Jinja2 Documentation (2.10)*. 2010. Available at: <https://jinja.palletsprojects.com/>.
- [81] Russell, Stuart: *Human Compatible: Artificial Intelligence and the Problem of Control*. Viking, 2019.
- [82] Sandler, Mark, Andrew Howard, Menglong Zhu, Andrey Zhmoginov y Liang Chieh Chen: *MobileNetV2: Inverted residuals and linear bottlenecks*. En *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 4510–4520, 2018.

- [83] Scikit-Image: *Scikit-Image: Procesamiento Avanzado de Imágenes*, 2023. <https://scikit-image.org/>.
- [84] Sculley, D., G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, JF Crespo y D. Dennison: *Hidden Technical Debt in Machine Learning Systems*. Advances in Neural Information Processing Systems, 2015.
- [85] Selvaraju, Ramprasaath R, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh y Dhruv Batra: *Grad-CAM: Visual explanations from deep networks via gradient-based localization*. Proceedings of the IEEE international conference on computer vision, páginas 618–626, 2017.
- [86] SGCCC: *Emisiones Históricas de Gases de Efecto Invernadero en Guatemala*. Informe técnico, 2019. <https://sgccc.org.gt/wp-content/uploads/2019/07/1RepCCGuaCap11.pdf>.
- [87] Simonyan, Karen y Andrew Zisserman: *Very deep convolutional networks for large-scale image recognition*. arXiv preprint arXiv:1409.1556, 2014.
- [88] Smilkov, Daniel, Nikhil Thorat, Been Kim, Fernanda Viégas y Martin Wattenberg: *SmoothGrad: removing noise by adding noise*. arXiv preprint arXiv:1706.03825, 2017.
- [89] Smith, A.: *The Importance of Waste Management to Environmental Sustainability*. Environmental Press, 2020.
- [90] Smith, John: *AI-Waste: Advanced E-Waste Classification Using Deep Learning*, 2020. <https://arxiv.org/abs/2003.12356>.
- [91] Smith, Leslie N.: *Cyclical learning rates for training neural networks*. 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), páginas 464–472, 2017.
- [92] Solve, MIT: *Reciclaje eficiente para comunidades rurales indígenas*. 2022. <https://solve.mit.edu/challenges/solved-youth-innovation-challenge-2/solutions/67882>.
- [93] Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever y Ruslan Salakhutdinov: *Dropout: A simple way to prevent neural networks from overfitting*. Journal of Machine Learning Research, 15:1929–1958, 2014.
- [94] Szegedy, Christian, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens y Zbigniew Wojna: *Rethinking the Inception architecture for computer vision*. En *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 2818–2826, 2016.
- [95] TACO: *TACO: Dataset de Clasificación y Detección de Residuos*, 2023. <https://www.kaggle.com/datasets/ahmedetefy/taco-trash-detection>.
- [96] Tajbakhsh, Nima, Jae Shin, Suryakanth R Gurudu, Robert T Hurst, Charles B Kendall, Michael B Gotway y Jianming Liang: *Convolutional neural networks for medical image analysis: Full training or fine tuning?* IEEE Transactions on Medical Imaging, 35(5):1299–1312, 2016.
- [97] Tan, Z., C. W. Lee, S. Y. Ong, A. J. Chua y T. S. Koh: *AIWaste: An AI-based Approach to Waste Classification for Efficient Waste Management*. En *Proceedings of the 2nd International Conference on Artificial Intelligence for Industries (AI4I)*, páginas 12–18. ACM, 2023.
- [98] TensorFlow: *TensorFlow: Guía Completa*, 2023. <https://www.tensorflow.org/>.
- [99] Theng, Quang: *Garbage Classification - 6 classes (775/class)*, 2023. <https://www.kaggle.com/datasets/quangtheng/garbage-classification-6-classes-775class>.
- [100] Torres Rodríguez, Agustín: *Desarrollo e implementación de clasificadores de imágenes mediante el empleo de algoritmos de aprendizaje por transferencia*, 2024. <https://www.researchgate.net/publication/375863525>.

- [101] TrashNet: *TrashNet: Dataset de Clasificación de Residuos*, 2023. <https://github.com/garythung/trashnet>.
- [102] Truong, Minh Tri y cols.: *AI-Driven Waste Management Systems: A Comprehensive Review*. Journal of Cleaner Production, 250:11952, 2022.
- [103] Truong, Phuoc, Jacob Hansen y Marcus Weisser: *AI in Waste Management: An Advanced Convolutional Neural Network Model for Waste Classification*. Journal of Waste Management, páginas 33–47, 2022.
- [104] UNEP: *Global Waste Management Outlook*. Informe técnico, United Nations Environment Programme, 2019. <https://www.unep.org/resources/report/global-waste-management-outlook>.
- [105] UNEP: *Global Waste Management Outlook*. Informe técnico, 2019. <https://www.unep.org/resources/report/global-waste-management-outlook>.
- [106] Valladolid, Universidad de: *Estudio de algoritmos de redes neuronales convolucionales en*, 2021. <https://uvadoc.uva.es/bitstream/10324/47137/1/TFG-G4811.pdf>.
- [107] Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser y Illia Polosukhin: *Attention Is All You Need*, 2017. <https://arxiv.org/abs/1706.03762>.
- [108] (WHO), World Health Organization: *Environmental health inequalities in Europe: Second assessment report*. 2018. <https://www.euro.who.int/en/publications/abstracts/environmental-health-inequalities-in-europe.-second-assessment-report-2018>.
- [109] Yosinski, Jason, Jeff Clune, Yoshua Bengio y Hod Lipson: *How transferable are features in deep neural networks?* En *Advances in Neural Information Processing Systems*, páginas 3320–3328, 2014.
- [110] Zaharia, Matei, Andy Chen, Aaron Davidson, Ali Ghodsi, Stephanie Hong, Andrew Konwinski, Sebastien Murching, Matei Nykodym, Patrick Ogilvie, Sue Ann Parkhe y cols.: *Accelerating the machine learning lifecycle with MLflow*. En *IEEE Data Engineering Bulletin*, volumen 41, páginas 39–45, 2018.
- [111] Zhang, S., A. H. Omar, A. S. Hashim, T. Alam, H. Khalifa y M. Elkotb: *Artificial intelligence and sustainability: Optimized algorithms for waste management and water quality prediction in urban environments*. Environmental Science and Technology, 2023. [https://doi.org/10.1016/s0921-3449\(02\)00002-2](https://doi.org/10.1016/s0921-3449(02)00002-2).



---

## Código para Cálculo de SmoothGrad

---

```
1  from tensorflow.keras import backend as K
2
3  # Función para calcular SmoothGrad
4  def smooth_grad(model, img_tensor, class_index, noise_level=0.1, num_samples=20):
5      smooth_saliency = np.zeros(img_tensor.shape[:2])
6      for _ in range(num_samples):
7          noise = np.random.normal(0, noise_level, img_tensor.shape)
8          noisy_img = img_tensor + noise
9          saliency = compute_saliency_map(model, noisy_img, class_index)
10         smooth_saliency += saliency
11     smooth_saliency /= num_samples
12     return smooth_saliency
13
14 # Función básica de cálculo de saliencia
15 def compute_saliency_map(model, img_tensor, label_index):
16     img_tensor = tf.convert_to_tensor([img_tensor])
17     with tf.GradientTape() as tape:
18         tape.watch(img_tensor)
19         preds = model(img_tensor)
20         loss = preds[:, label_index]
21     grads = tape.gradient(loss, img_tensor)
22     saliency = K.abs(grads)[0]
23     saliency = np.max(saliency, axis=-1)
24     return saliency
25
26 # Función para predecir, mostrar la clasificación y el mapa de saliencia de cada imagen
27 def classify_and_display_with_saliency(image_path):
28     # Cargar y mostrar la imagen original
29     img = image.load_img(image_path, target_size=(299, 299))
30     img_array = image.img_to_array(img)
31     img_array = preprocess_input(np.expand_dims(img_array, axis=0))
32
33     # Realizar predicción
34     predictions = model.predict(img_array)
```

```

35     predicted_class_idx = np.argmax(predictions[0])
36     predicted_class = list(class_translation.keys())[predicted_class_idx]
37     translated_class = class_translation[predicted_class]
38
39     # Calcular el mapa de saliencia usando SmoothGrad
40     smooth_saliency = smooth_grad(model, img_array[0], predicted_class_idx)
41
42     # Visualizar la imagen, el nombre de la clase y el mapa de saliencia
43     plt.figure(figsize=(12, 6))
44
45     # Imagen original
46     plt.subplot(1, 2, 1)
47     plt.imshow(image.array_to_img(img_array[0] * 0.5 + 0.5)) # Escala de [-1,1] a [0,1]
48     plt.title(f'Clasificación (Xception Final): {translated_class}')
49     plt.axis('off')
50
51     # Mapa de Saliencia
52     plt.subplot(1, 2, 2)
53     plt.imshow(smooth_saliency, cmap='hot')
54     plt.title(f'SmoothGrad (Xception Final) - Clase: {translated_class}')
55     plt.axis('off')
56
57     plt.show()
58
59     # Obtener nombres de las imágenes y aplicar la clasificación y saliencia
60     image_files = [os.path.join(test_path, img) for img in os.listdir(test_path) if img.endswith(('.png', '.jpg
61     for image_file in image_files:
62         classify_and_display_with_saliency(image_file)

```

---

## Código para Cálculo de Precisión del Chatbot Inteligente

---

```
1  import pandas as pd
2
3  # Datos de la encuesta
4  data = {
5      "Timestamp": [
6          "30/10/2024 11:30:33", "30/10/2024 11:32:16", "30/10/2024 11:35:30",
7          "30/10/2024 11:56:03", "30/10/2024 11:56:56", "30/10/2024 11:57:23",
8          "30/10/2024 11:59:56"
9      ],
10     "Email": [
11         "cgiron@munigate.com", "lvillatoro@munigate.com", "pmendoza@munigate.com",
12         "asalazar@munigate.com", "fhernandez@munigate.com", "rgarcia@munigate.com",
13         "mlopez@munigate.com"
14     ],
15     "Precision_in_Responses": [4, 5, 4, 5, 5, 4, 5],
16     "Thematic_Relevance": [4, 5, 5, 5, 5, 4, 4],
17     "Ability_to_Reject_Off-Topic": [5, 5, 5, 5, 5, 4, 5],
18     "Recommendation_Quality": [4, 5, 4, 5, 4, 4, 4],
19     "Consistency_in_Style": [5, 5, 4, 5, 4, 4, 4]
20 }
21
22 # Crear DataFrame a partir de los datos
23 df = pd.DataFrame(data)
24
25 # Eliminar columnas innecesarias para el análisis
26 df_cleaned = df.drop(columns=["Timestamp", "Email"])
27
28 # Calcular el puntaje total para cada participante
29 df_cleaned["Total_Score"] = df_cleaned.sum(axis=1)
30
31 # Calcular el porcentaje por criterio (basado en un máximo de 5 puntos por criterio)
32 for column in df_cleaned.columns[:-1]: # Excluye la columna 'Total Score'
33     df_cleaned[f"{column} (%)"] = (df_cleaned[column] / 5) * 100
34
```

```
35  # Crear correctamente la lista de columnas de porcentaje, excluyendo 'Total_Score'
36  percentage_columns = [f"{column} (%)" for column in df_cleaned.columns if column != "Total_Score" and "(%)"]
37
38  # Calcular el porcentaje promedio por participante
39  df_cleaned["Average_Percentage"] = df_cleaned[percentage_columns].mean(axis=1)
40
41  # Calcular el porcentaje promedio general del chatbot
42  overall_average_percentage = df_cleaned["Average_Percentage"].mean()
43
44  # Mostrar el DataFrame final con los resultados
45  print("Resultados detallados de la encuesta:")
46  print(df_cleaned)
47  print(f"\nPorcentaje promedio general de precisión y calidad del chatbot: {overall_average_percentage:.2f}%")
```

---

## Especificaciones Técnicas de la Tarjeta NVIDIA A100

---

La tarjeta NVIDIA A100, utilizada para el entrenamiento de los modelos en este proyecto, es una de las GPU más avanzadas para tareas de inteligencia artificial y *deep learning*. Sus especificaciones técnicas se presentan en la siguiente tabla, detallando sus capacidades en diferentes configuraciones.

<b>Especificaciones</b>	<b>A100 80 GB PCIe</b>	<b>A100 80 GB SXM</b>
FP64	9.7 TFLOPS	9.7 TFLOPS
Tensor Core de FP64	19.5 TFLOPS	19.5 TFLOPS
FP32	19.5 TFLOPS	19.5 TFLOPS
Tensor Float 32 (TF32)	156 TFLOPS   312 TFLOPS*	156 TFLOPS   312 TFLOPS*
Tensor Core de BFLOAT16	312 TFLOPS   624 TFLOPS*	312 TFLOPS   624 TFLOPS*
Tensor Core de FP16	312 TFLOPS   624 TFLOPS*	312 TFLOPS   624 TFLOPS*
Tensor Core de INT8	624 TOPS   1248 TOPS*	624 TOPS   1248 TOPS*
Memoria de la GPU	80 GB de HBM2e	80 GB de HBM2e
Ancho de banda de memoria de la GPU	1935 GB/s	2039 GB/s
Potencia máxima de diseño térmico (TDP)	300 W	400 W ***
GPU multiinstancia	Hasta 7 MIG a 10 GB	Hasta 7 MIG a 10 GB
Formato	PCIe	SXM
Refrigeración	Refrigeración por aire de dos ranuras o refrigeración por líquido de una sola ranura	SXM
Interconexión	NVIDIA NVLink Bridge para 2 GPU: 600 GB/s ** PCIe Gen4: 64 GB/s	NVLink: 600 GB/s PCIe Gen4: 64 GB/s
Opciones del servidor	Partners y NVIDIA-Certified Systems™ con entre 1 y 8 GPU	Sistemas certificados por NVIDIA y partners de NVIDIA HGX™ H100 con 4, 8 o 16 GPU, NVIDIA DGX™ H100 con 8 GPU

Tabla C.1: Especificaciones técnicas de la tarjeta NVIDIA A100.

- \* Con dispersión.
- \*\* GPU SXM4 a través de placas de servidores HGX A100; GPU PCIe a través de NVLink Bridge para hasta 2 GPU.
- \*\*\* TDP de 400 W para la configuración estándar. La SKU HGX A100 de 80 GB de solución térmica personalizada (CTS) puede admitir TDP de hasta 500 W.