



UNIVERSIDAD DE SALAMANCA

Escuela **politécnica** superior  
de **Zamora**

# Trabajo Fin de Grado

## DESARROLLO DE UNA HERRAMIENTA PARA LA APLICACIÓN DE ALGORITMOS DE ENJAMBRES COMO TÉCNICA DE CUANTIFICACIÓN DE COLOR

Autor:

Ángel Casado Nogueras

Tutora:

María Luisa Pérez Delgado

Junio 2025



VNiVERSiDAD  
D SALAMANCA

# Índice

1. Motivaciones .....	7
2. Introducción.....	8
2.1 Representación de imágenes en color.....	8
2.2 Los modelos de color.....	9
3. El problema de la cuantificación de color .....	9
3.1 Definición del problema .....	9
3.2 Aplicaciones prácticas .....	11
4. Los algoritmos de enjambres.....	12
4.1 Características generales.....	12
4.2 Algoritmo de optimización por enjambre de partículas (PSO) .....	13
4.2.1 Introducción.....	13
4.2.3 Variables.....	13
4.2.4 Definición del algoritmo.....	14
4.2.5 Parámetros PSO .....	16
4.3 Luciérnagas (FA).....	17
4.3.1 Introducción.....	17
4.3.3 Variables.....	17
4.3.4 Definición del algoritmo.....	17
4.4 Lobos (GWO).....	19
4.4.1 Introducción.....	19
4.4.2 Fundamentos.....	19
4.4.3 Definición del algoritmo.....	21
4.4.4 Variables.....	22
4.5 Ballenas (WOA) .....	23
4.5.1 Introducción.....	23
4.5.2 Fundamentos.....	23
4.5.3 Definición del algoritmo.....	24
4.6 Abejas (ABC) .....	26
4.6.1 Introducción.....	26
4.6.3- Fundamentos .....	27
4.6.4- Definición del algoritmo .....	28
5. Tecnologías empleadas.....	32
5.1 Python.....	32
5.2 NumPy .....	32
5.3 OpenCv.....	33
5.4 Scikit-learn .....	33

5.5 Scikit-image.....	33
5.6 Entornos virtuales .....	33
5.7 Control de versiones .....	34
5.8 Visual Studio Code.....	34
5.9 Otros .....	34
6. Pruebas .....	35
6.1 MSE .....	35
6.2 MAE .....	35
6.3 SSIM.....	35
6.4 UQI.....	36
6.5 FSIM.....	36
6.6 VIF .....	36
6.7 Criterios y conjuntos de pruebas .....	37
6.8 Análisis .....	39
7. Desarrollo del programa .....	99
7.1 Archivo main o ejecutor .....	99
7.2 Script.....	102
8. Trabajo futuro .....	105
9. Conclusiones.....	106
10.Bibliografía.....	107

## Índice de figuras

Ilustración 1 Imagen original [20] .....	10
Ilustración 2 Imagen resultante con 32 colores .....	10
Ilustración 3 Pirámide.....	19
Ilustración 4 Lobos Grises[22] .....	22
Ilustración 5 Logo Python [23].....	32
Ilustración 6 Logo Numpy[24].....	32
Ilustración 7 Logo GitHub[25].....	34
Ilustración 8 JSON de argumentos .....	34
Ilustración 9 Logo VisualStudioCode [26].....	34
Ilustración 10 Índices MSE para todos los algoritmos probados .....	85
Ilustración 11 Índices MAE para todos los algoritmos probados.....	87
Ilustración 12 Índices FSIM para todos los algoritmos probados .....	90
Ilustración 13 Índices SSIM para todos los algoritmos probados .....	92
Ilustración 14 Índices VIF para todos los algoritmos probados .....	95
Ilustración 15 Diagramas de barras de los tiempos de ejecución para todos los algoritmos probados .....	98
Ilustración 16 Argumentos .....	99
Ilustración 17 Obtener ruta de los directorios de imágenes.....	100
Ilustración 18 Comprobaciones de rutas e imágenes.....	100
Ilustración 19 Comprobaciones extra .....	101
Ilustración 20 Diccionarios Python de algoritmos y funciones.....	101
Ilustración 21 Método ejecutor de la herramienta .....	102
Ilustración 22 Constantes y directorios .....	102
Ilustración 23 Bucles .....	103
Ilustración 24 Sistema de checkpoints .....	103
Ilustración 25 Bucle Imagenes .....	104
Ilustración 26 Comienzo de la prueba .....	104
Ilustración 27 Ejecución de la herramienta .....	104
Ilustración 28 Fin del script.....	105

## Índice de tablas

Tabla 1 Resumen de pruebas.....	37
Tabla 2 Valores de los parámetros específicos para cada algoritmo de enjambre usado en las pruebas .....	38
Tabla 3 Tiempos PSO.....	39
Tabla 4 Tiempos FA.....	39
Tabla 5 Tiempos ABA.....	39
Tabla 6 Tiempos GWO.....	40
Tabla 7 Tiempos WOA.....	40
Tabla 8 32 colores - PSO.....	41
Tabla 9 32 colores - ABA.....	41
Tabla 10 32 colores - FA .....	41
Tabla 11 32 colores GWO .....	42
Tabla 12 32 colores - WOA.....	42
Tabla 13 64 colores - PSO.....	42
Tabla 14 64 colores - ABA.....	43
Tabla 15 64 colores - FA .....	43

Tabla 16 64 colores - GWO.....	43
Tabla 17 64 colores - WOA.....	44
Tabla 18 128 colores - PSO.....	44
Tabla 19 128 colores ABA.....	44
Tabla 20 128 colores - FA.....	45
Tabla 21 128 colores - GWO.....	45
Tabla 22 128 colores - WOA.....	45
Tabla 23 256 colores - PSO.....	46
Tabla 24 256 colores - ABA.....	46
Tabla 25 256 colores - FA.....	46
Tabla 26 256 colores - GWO.....	47
Tabla 27 256 colores - WOA.....	47
Tabla 28 Hipótesis Nula - Decisión - PSO_EMSE.....	48
Tabla 29 Resumen de la prueba de Friedman PSO_EMSE.....	48
Tabla 30 Comparaciones por parejas PSO_EMSE.....	48
Tabla 31 Hipótesis Nula - Decisión - PSO_EMAE.....	49
Tabla 32 Resumen de la prueba de Friedman PSO_EMAE.....	49
Tabla 33 Comparaciones por parejas PSO_EMAE.....	49
Tabla 34 Significación ajustada para PSO_EMAE.....	50
Tabla 35 Hipótesis Nula - Decisión - PSO_EFSIM.....	50
Tabla 36 Resumen de la prueba de Friedman PSO_EFSIM.....	50
Tabla 37 Comparaciones por parejas PSO_EFSIM.....	51
Tabla 38 Significación ajustada para PSO_EFSIM.....	51
Tabla 39 Hipótesis Nula - Decisión - PSO_ESSIM.....	52
Tabla 40 Resumen de la prueba de Friedman PSO_ESSIM.....	52
Tabla 41 Comparaciones por parejas PSO_ESSIM.....	52
Tabla 42 Hipótesis Nula - Decisión - PSO_EVIF.....	53
Tabla 43 Resumen de la prueba de Friedman PSO_EVIF.....	53
Tabla 44 Comparaciones por parejas PSO_EVIF.....	53
Tabla 45 Significación ajustada para PSO_EVIF.....	54
Tabla 46 Hipótesis Nula - Decisión - PSO_TIEMPO.....	54
Tabla 47 Resumen de la prueba de Friedman PSO_TIEMPOS.....	54
Tabla 48 Comparaciones por parejas PSO_TIEMPOS.....	55
Tabla 49 Hipótesis Nula - Decisión - ABA_EMSE.....	55
Tabla 50 Resumen de la prueba de Friedman ABA_EMSE.....	55
Tabla 51 Comparaciones por parejas ABA_EMSE.....	56
Tabla 52 Hipótesis Nula - Decisión - ABA_EMAE.....	56
Tabla 53 Resumen de la prueba de Friedman ABA_EMAE.....	56
Tabla 54 Comparaciones por parejas ABA_EMAE.....	56
Tabla 55 Hipótesis Nula - Decisión - ABA_EFSIM.....	57
Tabla 56 Resumen de la prueba de Friedman ABA_EFSIM.....	57
Tabla 57 Comparaciones por parejas ABA_EFSIM.....	57
Tabla 58 Hipótesis Nula - Decisión - ABA_ESSIM.....	58
Tabla 59 Resumen de la prueba de Friedman ABA_ESSIM.....	58
Tabla 60 Comparaciones por parejas ABA_ESSIM.....	58
Tabla 61 Hipótesis Nula - Decisión - ABA_EVIF.....	59
Tabla 62 Resumen de la prueba de Friedman ABA_EVIF.....	59
Tabla 63 Comparaciones por parejas ABA_EVIF.....	59
Tabla 64 Significación ajustada para ABA_EVIF.....	59
Tabla 65 Resumen de la prueba de Friedman ABA_ETIEMPO.....	60

Tabla 66 Comparaciones por parejas ABA_ETIEMPO.....	60
Tabla 67 Hipótesis Nula - Decisión - FA_EMSE .....	61
Tabla 68 Resumen de la prueba de Friedman FA_EMSE.....	61
Tabla 69 Comparaciones por parejas FA_EMSE.....	61
Tabla 70 Significación ajustada para FA_EMSE.....	61
Tabla 71 Hipótesis Nula - Decisión - FA_EMAE.....	62
Tabla 72 Resumen de la prueba de Friedman FA_EMAE .....	62
Tabla 73 Comparaciones por parejas FA_EMAE.....	62
Tabla 74 Significación ajustada para FA_EMAE .....	63
Tabla 75 Hipótesis Nula - Decisión - FA_EFSIM .....	63
Tabla 76 Resumen de la prueba de Friedman FA_EFSIM.....	63
Tabla 77 Comparaciones por parejas FA_EFSIM.....	64
Tabla 78 Significación ajustada para FA_EFSIM.....	64
Tabla 79 Hipótesis Nula - Decisión - FA_ESSIM .....	64
Tabla 80 Resumen de la prueba de Friedman FA_ESSIM.....	65
Tabla 81 Comparaciones por parejas FA_ESSIM.....	65
Tabla 82 Hipótesis Nula - Decisión - FA_EVIF .....	66
Tabla 83 Resumen de la prueba de Friedman FA_EVIF.....	66
Tabla 84 Comparaciones por parejas FA_EVIF .....	66
Tabla 85 Significación ajustada para FA_EVIF.....	66
Tabla 86 Hipótesis Nula - Decisión - FA_ETIEMPO.....	67
Tabla 87 Resumen de la prueba de Friedman FA_ETIEMPO .....	67
Tabla 88 Comparaciones por parejas FA_ETIEMPO .....	67
Tabla 89 Hipótesis Nula - Decisión - GWO_EMSE .....	68
Tabla 90 Resumen de la prueba de Friedman GWO_EMSE .....	68
Tabla 91 Comparaciones por parejas GWO_EMSE .....	68
Tabla 92 Hipótesis Nula - Decisión - GWO_EMAE .....	69
Tabla 93 Resumen de la prueba de Friedman GWO_EMAE.....	69
Tabla 94 Comparaciones por parejas GWO_EMAE.....	69
Tabla 95 Hipótesis Nula - Decisión - GWO_EFSIM.....	70
Tabla 96 Resumen de la prueba de Friedman GWO_EFSIM .....	70
Tabla 97 Comparaciones por parejas GWO_EFSIM .....	70
Tabla 98 Significación ajustada para GWO_EFSIM .....	71
Tabla 99 Hipótesis Nula - Decisión - GWO_ESSIM.....	71
Tabla 100 Resumen de la prueba de Friedman GWO_ESSIM .....	71
Tabla 101 Comparaciones por parejas GWO_ESSIM .....	71
Tabla 102 Significación ajustada para GWO_ESSIM .....	72
Tabla 103 Hipótesis Nula - Decisión - GWO_EVIF .....	72
Tabla 104 Resumen de la prueba de Friedman GWO_EVIF .....	72
Tabla 105 Comparaciones por parejas GWO_EVIF .....	72
Tabla 106 Significación ajustada para GWO_EVIF .....	73
Tabla 107 Hipótesis Nula - Decisión - GWO_ETIEMPO .....	73
Tabla 108 Resumen de la prueba de Friedman GWO_ETIEMPO.....	73
Tabla 109 Comparaciones por parejas GWO_ETIEMPO.....	74
Tabla 110 Hipótesis Nula - Decisión - WOA_EMSE .....	74
Tabla 111 Resumen de la prueba de Friedman WOA_EMSE .....	74
Tabla 112 Comparaciones por parejas WOA_EMSE .....	75
Tabla 113 Significación ajustada para WOA_EMSE .....	75
Tabla 114 Hipótesis Nula - Decisión - WOA_EMAE .....	76
Tabla 115 Resumen de la prueba de Friedman WOA_EMAE.....	76

Tabla 116 Comparaciones por parejas WOA_EMAE.....	76
Tabla 117 Significación ajustada para WOA_EMAE.....	77
Tabla 118 Hipótesis Nula - Decisión - WOA_EFSIM.....	77
Tabla 119 Resumen de la prueba de Friedman WOA_EFSIM .....	77
Tabla 120 Comparaciones por parejas WOA_EFSIM .....	78
Tabla 121 Significación ajustada para WOA_EFSIM .....	78
Tabla 122 Hipótesis Nula - Decisión - WOA_ESSIM.....	78
Tabla 123 Resumen de la prueba de Friedman WOA_ESSIM .....	78
Tabla 124 Comparaciones por parejas WOA_ESSIM .....	79
Tabla 125 Hipótesis Nula - Decisión - WOA_EVIF.....	79
Tabla 126 Resumen de la prueba de Friedman WOA_EVIF .....	79
Tabla 127 Comparaciones por parejas WOA_EVIF .....	80
Tabla 128 Significación ajustada para WOA_EVIF .....	80
Tabla 129 Hipótesis Nula - Decisión - WOA_ETIEMPO .....	81
Tabla 130 Resumen de la prueba de Friedman WOA_ETIEMPO.....	81
Tabla 131 Comparaciones por parejas WOA_ETIEMPO.....	81

# 1. Motivaciones

El desarrollo de este trabajo de fin de grado nace del creciente interés en las técnicas de optimización inspiradas por la naturaleza, en particular los algoritmos de enjambre, que han demostrado ser métodos bastante eficaces y versátiles en la solución de problemas complejos en diversas áreas.

También ha sido motivado por la asignatura de Sistemas Inteligentes con la cual personalmente disfruté mucho cursándola. En ella pudimos probar otro lenguaje de programación como Python, comprobar las diferencias que existen entre C y Python, e implementar código con un objetivo algo más visual que en el resto de la carrera.



## 2. Introducción

La cuantificación de color en imágenes es un problema complejo en estos tiempos en los que nos comunicamos frecuentemente con el uso de imágenes.

El objetivo principal es reducir el número de colores de una imagen a un conjunto más pequeño de colores representativos, a los que se llama **paleta cuantizada** de colores, con la menor pérdida de calidad posible. La imagen resultante, que se denomina **imagen cuantizada**, debe ser reconocible y lo más parecida posible visualmente a la original, pero con una representación mucho más simple y eficiente.

Puede parecer un proceso simple a primera vista, pero representa varios desafíos y requiere del uso de técnicas avanzadas, como algoritmos de enjambres, para obtener buenos resultados usando un tiempo de cómputo razonable. Algunas de las razones que explican por qué la cuantificación de color en imágenes es un problema complejo son:

- **Gran volumen de datos:** Ya que las imágenes que se procesan pueden contener miles de píxeles y cada píxel está representado por varios datos, procesar todos estos datos de manera eficiente y efectiva es un gran desafío.
- **Preservación de detalles importantes:** para la cuantificación de color es esencial preservar los detalles importantes de la imagen, como bordes y características distintivas. La simplificación excesiva puede llevar a la pérdida de información crítica y degradar la calidad visual de la imagen.
- **Selección de colores representativos:** Elegir los colores que representarán la imagen cuantizada de manera precisa es una tarea compleja. La selección de estos colores debe hacerse de manera inteligente para garantizar que la imagen cuantizada sea lo más fiel posible a la original.
- **Mantenimiento de la apariencia visual:** La reducción de color debe lograr un equilibrio entre la reducción del número de colores propiamente dicha y la retención de la apariencia visual general de la imagen. Los cambios agresivos en los colores pueden hacer que la imagen resultante sea irreconocible o menos atractiva.
- **Optimización de la eficiencia:** Procesar imágenes en tiempo real o en aplicaciones donde la velocidad es crucial (como la transmisión de video) requiere algoritmos de reducción de color eficientes que puedan realizar cálculos rápidos y precisos.
- **Evaluación de la calidad:** Medir la calidad de la imagen resultante es subjetivo y puede depender de la percepción humana. Evaluar y comparar algoritmos en términos de calidad de reducción de color es un aspecto importante de la investigación en este campo.

### 2.1 Representación de imágenes en color

La representación de imágenes en color en un ordenador es un proceso fundamental que implica codificar y almacenar la información de color de una imagen para luego procesarla, visualizarla y manipularla. Para lograr esto, los ordenadores utilizan modelos de color, siendo el modelo RGB (Red, Green, Blue) uno de los más comunes.

En el modelo RGB, cada píxel de una imagen se representa como una combinación de tres canales de color: rojo, verde y azul. Cada canal puede tomar valores entre 0 y 255 que representan la intensidad de ese color en un píxel específico.

La combinación de estos tres canales de color en cada píxel permite crear una gran cantidad de colores y tonalidades (16.777.216 colores posibles), lo que nos permite representar imágenes en color de alta calidad. La información de color de cada píxel se almacena en una matriz tridimensional, donde las dos primeras dimensiones representan la posición espacial del píxel en la imagen y la tercera dimensión almacena los valores de intensidad de rojo, verde y azul.

Esta representación de color en RGB es la base para todas las operaciones de procesamiento de imágenes en color, incluida la reducción de color. Durante este proceso, el objetivo es reducir el número de colores de la imagen, manteniendo al mismo tiempo la apariencia visual deseada. Los algoritmos de enjambres nos ayudan a “escoger” los colores adecuados para preservar la calidad en la imagen reducida.

## 2.2 Los modelos de color

Ya que se ha hablado de cómo un ordenador representa las imágenes en color utilizando el modelo de color RGB, me parece importante hablar brevemente de otros modelos que existen.

Uno de los más simples es la **escala de grises** que tiene un solo canal, en el que representa valores de grises del blanco al negro. Pueden tener una profundidad de 8 o de 16 bits por píxel.

Otro de ellos es el **indexado** en el que las imágenes tienen una gama de colores reducida para que sus archivos sean más pequeños. Las imágenes indexadas por lo general se suelen usar en páginas web o en gráficos con pocos colores. Solo cuentan con un canal de 8 bits.

Otros dos modelos bastante extendidos hoy en día son:

1. **CMYK:** Es un modelo de color de cuatro canales. Los valores representan los colores cian, magenta, amarillo y negro. Al igual que el RGB puede tener 8 o 16 bits por canal. Este modelo compone los colores de manera sustractiva, es decir, parte del blanco y va restando colores usando sus complementarios.

Este modelo suele ser usado por las impresoras. Al retocar las fotos en un ordenador sirve de ayuda para comprobar si los colores de la imagen en RGB serán los mismos en CMYK. Esto a veces no sucede ya que algunos colores RGB no se pueden conseguir en este modelo.

**LAB:** En este modelo existen 3 canales, L, A y B. L representa la información tonal o luminancia (intensidad de la luz en la imagen) y A y B la información de color o crominancia (cómo los colores se combinan y varían en una imagen), siendo A la que añade los colores en la línea del rojo y el verde y B la que añade el contenido en amarillo o azul.[1]

## 3. El problema de la cuantificación de color

### 3.1 Definición del problema

Una imagen digital se forma con una cantidad de píxeles representados usando un modelo de color concreto. Cuando este modelo de color es el RGB, como ya se ha indicado, cada píxel se representa con 3 números enteros entre el 0 y el 255 por lo cual este espacio de color nos permite usar  $256^3$  colores diferentes.

Como se ha dicho anteriormente la cuantificación de color trata de reducir este número de colores evitando la pérdida de información visual. Para conseguirlo realiza dos operaciones:

1. **Selección de colores representativos:** Esta operación implica la elección de un conjunto limitado de colores representativos que serán utilizados para reemplazar los colores originales de la imagen. Para seleccionar estos colores, se aplican algoritmos de agrupación de colores, uno de ellos es el algoritmo K-means, que agrupa los colores similares en clusters y utiliza los centroides de estos clusters como colores representativos. Este conjunto de colores representativos lo llamamos **paleta cuantizada**. Esta operación es esencial para simplificar la representación de color de la imagen y reducir el número de colores utilizados.
2. **Asignación de colores representativos a los píxeles:** Una vez que se han seleccionado los colores representativos, se lleva a cabo la operación de asignación de colores. En esta etapa, cada píxel en la imagen original se asigna al color representativo más cercano en términos de distancia de color seleccionado en la anterior operación. Esto implica calcular la diferencia de color entre el color original del píxel y los colores representativos de la paleta cuantizada y seleccionar el color representativo que minimice esta diferencia. Sin embargo, existen otros métodos de asignación de colores, y este es solo uno de ellos. Dependiendo del algoritmo o el objetivo, se podrían emplear técnicas alternativas para asignar los colores representativos a los píxeles. Esta operación es fundamental para reemplazar los colores originales por sus equivalentes representativos y, de esta manera, lograr la reducción de color en la imagen.

A continuación, en la Ilustración 1 se muestra una imagen del paquete CQ100 [18] y en la Ilustración 2 se muestra la imagen cuantizada resultante con 32 colores.



*Ilustración 1 Imagen original [20]*



*Ilustración 2 Imagen resultante con 32 colores*

## 3.2 Aplicaciones prácticas

La cuantificación de color tiene diversas aplicaciones prácticas en varios campos, tales como:

- **Procesamiento de imágenes y fotografía:** Se utiliza para reducir el número de colores en una imagen digital, lo que es útil para la compresión de imágenes y la optimización para dispositivos y medios con capacidad de color limitada.
- **Diseño gráfico y web:** Ayuda a los diseñadores a crear paletas de colores coherentes y limitadas para mantener la consistencia y mejorar la estética en diseños y sitios web.
- **Reconocimiento de patrones e identificación de objetos:** la cuantificación de color puede ayudar a simplificar la identificación y clasificación de objetos al reducir la complejidad de las imágenes, facilitando el reconocimiento de patrones.
- **Impresión y fabricación de textiles:** En la industria de la impresión y en la fabricación de textiles, la cuantificación de color es crucial para asegurar que los colores utilizados en los diseños sean reproducibles y consistentes en diferentes lotes de producción.
- **Cartografía y análisis geoespacial:** Se utiliza para simplificar la representación visual de mapas y datos geoespaciales, ayudando a resaltar características importantes y mejorar la legibilidad.
- **Análisis médico y de imágenes biomédicas:** En el campo de la medicina, la cuantificación de color se aplica en el análisis de imágenes biomédicas para mejorar la visualización y el diagnóstico de ciertas condiciones al resaltar variaciones sutiles en los tejidos.
- **Análisis de calidad y control en la industria alimentaria:** Se usa para evaluar la calidad y la madurez de alimentos basándose en su color, lo cual es especialmente útil en líneas de producción automatizadas.
- **Arte y restauración:** En el arte digital y la restauración de obras, la cuantificación de color puede ayudar a analizar y replicar los colores utilizados por los artistas originales, así como a detectar alteraciones o daños.
- **Seguridad y vigilancia:** La cuantificación de color puede mejorar el rendimiento de los sistemas de vigilancia al reducir la complejidad de las imágenes, facilitando la detección de movimientos o actividades sospechosas.
- **Investigación científica y análisis de datos:** En diversas áreas científicas, la cuantificación de color puede ser útil para visualizar y analizar datos, especialmente en representaciones gráficas de información compleja.

## 4. Los algoritmos de enjambres.

### 4.1 Características generales.

Los algoritmos de enjambres es la disciplina que se ocupa de los sistemas naturales y artificiales compuestos por muchos individuos que se coordinan mediante el control descentralizado y la autoorganización. Se trata de un subcampo de la inteligencia artificial que se centra en los comportamientos colectivos que resultan de las interacciones de los individuos entre sí y con su entorno o medio ambiente al igual que lo hacen los enjambres naturales.

Ciertos sistemas humanos también utilizan estos algoritmos. Algunos sistemas de varios robots, y diversos programas cuyo objetivo es resolver problemas de optimización y análisis de datos.[2]

Algunas propiedades de estos algoritmos son:

1. **La autonomía:** Cada agente (o partícula) en el enjambre opera de manera individual, tomando decisiones basadas en su información local y, posiblemente, en alguna información global disponible para el enjambre.
2. **Descentralización:** No existe un control centralizado que dirija las acciones de los agentes; en su lugar, el comportamiento global emerge de las interacciones locales entre los agentes y entre los agentes y su entorno.
3. **Distribución:** Los algoritmos de enjambres suelen ser inherentemente distribuidos, lo que los hace robustos y escalables, ya que el fallo de una partícula tiene un impacto limitado en el enjambre en su totalidad.
4. **Autoorganización:** Las interacciones entre las partículas y la aplicación de reglas sencillas dan lugar a un comportamiento colectivo "inteligente" y autoorganizado. Este comportamiento es emergente, es decir, surge de las acciones de las partículas individuales.
5. **Interacciones simples:** Las reglas que rigen las interacciones entre las partículas son simples, pero pueden dar lugar a comportamientos colectivos complejos y a la solución de problemas complejos.
6. **Retroalimentación positiva y negativa:** Los mecanismos de retroalimentación, tanto positiva como negativa, son muy importantes para el desarrollo y la estabilización de los patrones de comportamiento del enjambre. La retroalimentación positiva promueve la formación de estructuras o caminos para hallar una mejor solución mientras que la retroalimentación negativa ayuda a prevenir la saturación de datos o una mala solución rápida.
7. **Exploración y explotación:** Los algoritmos de enjambres equilibran entre la exploración del espacio de búsqueda para descubrir nuevas soluciones y la explotación de las mejores soluciones encontradas para afinarlas.
8. **Adaptabilidad:** La inteligencia de enjambre es altamente adaptable a cambios en el entorno o en los parámetros del problema, lo que permite al enjambre encontrar nuevas soluciones cuando las condiciones cambian.
9. **Robustez:** Los enjambres son robustos frente a fallos y variaciones, ya que la pérdida de algunas partículas generalmente no impide que el enjambre en su conjunto continúe funcionando eficazmente.

Algo que comparten todos los algoritmos de optimización es que realizan su búsqueda en un determinado espacio de búsqueda, en el caso de la cuantificación de color se puede definir claramente al estar determinado por el número de colores que se desea obtener en la paleta cuantizada, y cada color está representado por un vector de tres dimensiones que corresponden a las componentes RGB.

Por lo tanto, si se desea reducir la imagen a una paleta de  $n$  colores, el espacio de búsqueda sería:  $n * 3$  dimensiones.

## 4.2 Algoritmo de optimización por enjambre de partículas (PSO)

### 4.2.1 Introducción

La **Optimización por Enjambres de Partículas** (conocida como **PSO**, por sus siglas en inglés, **Particle Swarm Optimization**) es una técnica de optimización/búsqueda.

Este algoritmo fue descrito alrededor de 1995 por Kennedy y Eberhart, y se inspira en el comportamiento de los enjambres de insectos en la naturaleza.

En este algoritmo se trabaja con una función objetivo que queremos optimizar, denotada comúnmente como  $f(x_1, \dots, x_n)$ , esta función devuelve una medida de lo buena que es una solución en cierto punto del espacio de búsqueda.

El objetivo de este algoritmo es encontrar los valores óptimos de las variables  $x_1, x_2, \dots, x_n$  para maximizar (o minimizar) el valor de la función  $f$ . Esta función se denomina **función fitness**. [3]

### 4.2.3 Variables

- Se considera un conjunto de  $I$  individuos llamados en este algoritmo **partículas**. Cada una de las  $I$  partículas tendrá asociado un estado que se irá modificando con el tiempo. Estas partículas tendrán:
  - Posición ( $x_i$ ): representa el estado de la partícula, es una posible solución al problema.
  - Velocidad ( $v_i$ ): determina la actualización de la posición.
  - Mejor posición personal ( $b_i$ ): esta variable representa la mejor solución encontrada por la partícula durante la ejecución.
- La solución al problema viene dada por la mejor posición encontrada por el enjambre la cual llamamos la mejor posición global, representada como:  $g$ .

En este algoritmo las partículas (individuos) se mueven por el espacio de solución del problema guiadas por ellas mismas y por el conjunto de todo el enjambre.

Este movimiento hace variar su posición, velocidad y mejor posición personal.

## 4.2.4 Definición del algoritmo

### PSEUDOCODIGO

Inicializar la población de individuos

#### REPETIR

Evaluar el fitness de cada individuo.

Actualizar la mejor solución personal de cada individuo.

Actualizar la mejor solución global.

Actualizar la velocidad y posición de cada individuo. ([Ec.1](#), [Ec.2](#))

**HASTA** (condición de parada)

#### Inicializar la población de individuos

Al principio del algoritmo se inicializan las variables de cada individuo su **posición** y su **velocidad**.

Su posición será un valor aleatorio dentro del espacio de búsqueda y su velocidad será un valor aleatorio entre  $[v_{\min}, v_{\max}]$ , que son dos parámetros del algoritmo que se definen a priori.

#### Evaluar el fitness de cada individuo

Para calcular este valor se aplica la función objetivo del problema a la posición de la partícula. En el caso de la cuantificación de color se utilizarán funciones tales como MSE (error cuadrático medio), MAE (Error absoluto medio), SSIM (Índice de similitud estructural) o MS-SSIM (Índice de similitud estructural multiescala) (a partir de ahora función objetivo).

$$fitness_i \approx f(x_i)$$

Este valor llamado fitness determina la calidad de la solución cuya posición representa.

#### Actualizar la mejor solución personal de cada individuo.

Si al calcular el fitness un individuo encuentra una mejor solución personal (es decir, encuentra una mejor posición lo que resulta en un mejor fitness) que la que tuviese almacenada para ella hasta el momento, actualiza la mejor solución con el valor de ésta.

Se da un nuevo valor a  $b_i$

#### Actualizar la mejor solución global.

Con la solución global ocurre lo mismo que con la mejor solución personal de cada individuo: si se ha encontrado una solución que mejora a la que hubiese almacenada como mejor solución global, se guarda como nueva mejor solución global.

Se da un nuevo valor a  $g$ .

#### Actualizar la velocidad y posición de cada individuo.

En este paso se realizan dos procedimientos:

1. *Actualizar velocidad de cada individuo.*

Se calcula una nueva velocidad de cada individuo  $i$ :

$$v_i(t+1) = w v_i(t) + f_1 e_1 [b_i(t) - x_i(t)] + f_2 e_2 [g - x_i(t)] \quad \text{Ec.1}$$

Cuyos parámetros son:

**e<sub>1</sub>, e<sub>2</sub>**: valores aleatorios en [0, 1].

**w**: inercia.

**f<sub>1</sub>**: parámetro cognitivo.

**f<sub>2</sub>**: parámetro social.

Esta actualización incluye tres componentes que son:

-Velocidad anterior:

$$w v_i(t)$$

Donde w (inercia) debe seleccionarse atendiendo a que si es grande se realizará una exploración global del espacio de solución, si es pequeña será exploración local. Se aconseja que este parámetro decrezca con las iteraciones.

-La distancia a la mejor posición personal:

$$f_1 e_1 [b_i(t) - x_i(t)]$$

En esta componente encontramos dos parámetros:

**e<sub>1</sub>**: será un valor aleatorio entre [0, 1].

**f<sub>1</sub>**: el parámetro cognitivo que determina la importancia que se da a la experiencia propia de cada individuo.

-La distancia a la mejor solución global:

$$f_2 e_2 [g - x_i(t)]$$

En esta componente tenemos los parámetros:

**e<sub>2</sub>**: será un valor aleatorio entre [0, 1].

**f<sub>2</sub>**: el parámetro social que determina la importancia que se da a la experiencia del enjambre.

Con el valor calculado de la velocidad, se debe ajustar al intervalo válido de la velocidad, es decir, nuestro nuevo valor:  $v_i(t+1)$  deberá estar entre  $[v_{min}, v_{max}]$ .

Para ello:

- Si  $v_i(t+1) > v_{max} \Rightarrow v_i(t+1) = v_{max}$

- Si  $v_i(t+1) < v_{min} \Rightarrow v_i(t+1) = v_{min}$

## 2. Actualizar posición de cada individuo.

Se calcula la nueva posición de cada individuo i.

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad \text{Ec.2}$$

Este nuevo valor para la posición de cada individuo depende de la actual posición y de la nueva velocidad que se acaba de calcular.



#### 4.2.5 Parámetros PSO

Como se ha hablado antes en el PSO existen varios parámetros principales que influyen en el comportamiento y en la eficacia del algoritmo, son: el coeficiente de **inercia**, y los parámetros **cognitivo** y **social**.

##### **Inercia:**

Este coeficiente afecta a la velocidad a la que los individuos se mueven en el espacio de búsqueda. Ayuda a controlar los resultados de la velocidad anterior del individuo sobre su velocidad actual. Un valor alto ayuda a que el individuo mantenga su dirección y velocidad anterior, lo que se traduce en una mejor exploración global. Por otro lado, un valor bajo hace que el individuo pierda velocidad por lo que ayuda a una exploración más local y detallada.

##### **Parámetro cognitivo:**

También conocido como parámetro personal o local, determina el grado en el que cada individuo se ve influido por sí mismo. Influye en cuánto se moverá un individuo hacia la mejor posición encontrada individualmente. Un valor alto de este componente dará a cada individuo una mayor motivación para seguir su propio camino hacia lo que percibe como una mejor solución, dando así lugar a una mayor variedad de soluciones exploradas por el enjambre.

##### **Parámetro social:**

Determina el grado de influencia de la mejor posición encontrada por todo el enjambre sobre el movimiento de cada individuo. Valores más altos de este componente resultan en una mayor convergencia del enjambre hacia las áreas del espacio de búsqueda que parecen más prometedoras según la experiencia del enjambre, es decir, los individuos preferirán moverse hacia la mejor solución encontrada por el conjunto de individuos.

## 4.3 Luciérnagas (FA)

### 4.3.1 Introducción

El algoritmo firefly, es un algoritmo metaheurístico de optimización, inspirado en el comportamiento del centelleo de las luciérnagas [4]. El propósito primario de una luciérnaga es generar destellos de luz para actuar como sistema de señal para atraer a otras luciérnagas. Xin-She Yang formuló este algoritmo con las siguientes premisas:

1. Todas las luciérnagas son "unisexuales", de modo que cualquier luciérnaga individual será atraída por todas las demás.
2. La atracción es proporcional a su brillo, y para cualquier par de luciérnagas, la menos brillante será atraída por (y por lo tanto se desplazará hacia) la más brillante; aun así, la intensidad (el brillo aparente) decrece cuando aumenta la distancia entre ambas;
3. Si no hay luciérnagas más brillantes que una dada, esta se mueve aleatoriamente.

El brillo es asociado con los valores de una función objetivo.

### 4.3.3 Variables

En este algoritmo se considera un conjunto de  $I$  individuos (luciérnagas). Cada uno de estos individuos representa una posible solución al problema. La calidad de esa solución dependerá del **brillo** que esté asociado a cada individuo.

Posición  $\rightarrow f_i = (x_{i1}, \dots, x_{ir})$

Brillo  $\rightarrow L_i \approx f(f_i)$

En este contexto el **brillo** de cada individuo será el fitness de este algoritmo, es decir, el individuo que posea el mayor brillo será la mejor solución al problema. Como en este algoritmo los individuos se mueven hacia el individuo con un mayor brillo, todos ellos se acercan a la mejor solución, menos el individuo más brillante.

La **atracción** entre los individuos disminuye con la distancia debido a la absorción de la luz. Este valor normalmente va cambiando para representar la disminución (visibilidad) del brillo con la distancia.

### 4.3.4 Definición del algoritmo

#### PSEUDOCODIGO

Generar la población inicial de luciérnagas

#### REPETIR

Mover cada luciérnaga hacia las más brillantes ([Ec.3](#) [Ec.4](#))

Mover la luciérnaga más brillante [Ec.5](#)

Actualizar el brillo de las luciérnagas

Ordenarlas por brillo y encontrar la mejor

**HASTA** (condición de parada)

### Generar la población inicial de luciérnagas

En este primer paso repartimos los individuos aleatoriamente en el espacio de búsqueda y calculamos el brillo inicial de cada uno.

Por cada iteración:

### Mover cada luciérnaga hacia las más brillantes

Cada luciérnaga (excepto la más brillante) mueve su posición hacia la más brillante, debido a esto se calcula una nueva posición para cada luciérnaga, en función de su posición previa y de la atracción ejercida por las otras luciérnagas.

Un individuo **I** se mueve hacia otro individuo **k**, que es más brillante que él, siguiendo esta ecuación:

$$f_i(t + 1) = f_i(t) + \beta(r_{ik})[f_k(t) - f_i(t)] + \varepsilon_i \quad \text{Ec.3}$$

Donde:

$f_i(t + 1)$  -> es la siguiente posición del individuo i

$f_i(t)$  -> es la posición actual del individuo i

$\beta(r_{ik})$  -> es el atractivo del individuo k sobre el i

$\varepsilon_i$  -> un valor aleatorio

Donde  $\beta(r_{ik})$  se calcula mediante:

$$\beta(r_{ik}) = \beta_0 e^{-\gamma r_{ij}^2} \quad \text{Ec.4}$$

$\beta_0$  -> atractivo a distancia cero

$\gamma$  -> coeficiente de absorción de la luz del medio

$r_{ij}$  -> distancia entre los individuos i y j

### Mover la luciérnaga más brillante

El individuo más brillante no se ve atraído por ningún otro, por lo que se mueve aleatoriamente.

Su nueva posición será:

$$f_{best}(t + 1) = f_{best}(t) + \varepsilon \quad \text{Ec.5}$$

### Actualizar el brillo de las luciérnagas

En este paso se calcula el fitness (brillo) de cada individuo mediante la función objetivo del problema.

### Ordenarlas por brillo y encontrar la mejor

En este último paso se ordenan los individuos por brillo y se determina el mejor (el que más brillo tenga). Para ello se utilizan diversos algoritmos de ordenación comunes (esta elección dependerá del número de individuos, la complejidad del algoritmo, y si el conjunto de datos está parcialmente ordenado).

## 4.4 Lobos (GWO)

### 4.4.1 Introducción

Este algoritmo de optimización metaheurística está basado en el comportamiento social y de caza de los lobos grises. Fue desarrollado por **Seyedeli Mirjalili** en 2014, simula tanto la estructura social de estos lobos como sus tácticas de caza para resolver complejos problemas de optimización. Estos lobos son conocidos por su comportamiento de caza estratégico y su jerarquía social, estos componentes se incluyen en este algoritmo para resolver estos problemas.

En esta jerarquía social, los individuos se clasifican en varias categorías: Alfas, Betas, Deltas y Omegas. Los Alfas son los que lideran la manada, estos toman las decisiones sobre la caza, el movimiento del grupo y su descanso. Los Betas, que son los colíderes, ayudan a los Alfas en la toma de decisiones y en la comunicación con el resto de los individuos de la manada. Los Deltas ocupan un rango inferior, estos se ocupan de cuidar o vigilar. Por último, los Omegas están en la base de la pirámide jerárquica, estos siguen las decisiones de los miembros de rango superior sin cuestionar sus decisiones.

Este algoritmo utiliza esta jerarquía para modelar la cooperación entre los lobos en la resolución de problemas de optimización. Los mejores candidatos que son los Alfas lideran la búsqueda, estos son seguidos por los Betas y los Deltas, mientras que los Omegas siguen las direcciones establecidas por los miembros superiores. Durante el proceso de optimización, todos los lobos, que son todas las posibles soluciones, van ajustando sus posiciones respecto a los lobos de mejor desempeño en su entorno (espacio de búsqueda).

Este enfoque le permite al GWO explorar eficazmente el espacio de búsqueda y explotar las áreas más prometedoras que se acerquen a las mejores soluciones encontradas. Posee un gran equilibrio entre la exploración y la explotación de las soluciones, y una gran capacidad para no quedarse estancado en mínimos locales, el GWO ha demostrado ser efectivo en una gran variedad de aplicaciones prácticas, desde la ingeniería hasta la ciencia de datos. Desde sus inicios, este algoritmo ha generado un gran interés en la comunidad de investigación por su robustez y rendimiento superior en comparación con otras técnicas de optimización.[5]

### 4.4.2 Fundamentos

El GWO introduce  $I$  lobos en la manada, desde aquí llamados  $I$  individuos.

- Cada individuo representa una posible solución al problema.
- La calidad o fitness de esa solución se calcula mediante la función objetivo del problema.

Esta jerarquía se representa de la siguiente manera:

Los tres mejores lobos de la manada estarán en los tres niveles superiores de la jerarquía, es decir, en cada uno de estos tres niveles habrá un solo lobo.

- El mejor lobo (mejor solución) se denota como  $\alpha$ . ->
- El segundo mejor lobo se denota como  $\beta$ . ->
- El tercer mejor lobo se denota como  $\delta$ . ->
- El resto de los lobos se consideran lobos  $\omega$ . ->

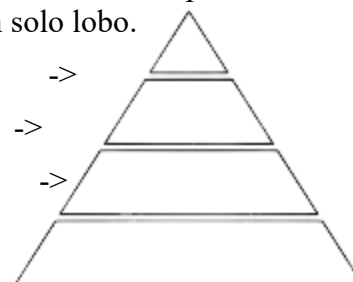


Ilustración 3 Pirámide

El proceso de optimización está representado por la operación de caza, en esta actividad los lobos de los tres primeros niveles guían, mientras que el resto siguen a éstos.

Este algoritmo posee varios coeficientes sociales ( $a$ ,  $A$  y  $C$ ).

El parámetro  $a$  o **factor de exploración** es el que guía el proceso de búsqueda. Este parámetro varía linealmente desde 2 hacia 0 con las iteraciones y se utiliza para calcular dos vectores aleatorios ( $A$  y  $C$ ) que serán los condicionantes de la exploración/explotación del algoritmo.

Estos vectores al depender de  $a$  también disminuirán con las iteraciones. Estos se calculan de la siguiente manera:

$$A = 2ar_1 - a \text{ Ec.6}$$

$$C = 2r_2 \text{ Ec.7}$$

Donde:

$r_1, r_2$  : son vectores aleatorios en  $[0, 1]$ .

$a$ : es el parámetro del algoritmo (factor de exploración)

Cuando el valor absoluto de estos vectores es mayor que uno se entiende que los individuos divergen respecto de la presa, se dice que están en fase de exploración.

Al converger a la presa, este valor será menor que uno, en este caso están en fase de explotación.[6]

### 4.4.3 Definición del algoritmo

En el GWO la búsqueda de la presa, es decir, la solución del problema se inicia con la creación de una población de  $I$  individuos al azar, igual que en el resto de los algoritmos.

Después de esto se realiza un proceso iterativo en el cual:

1. Los lobos: alfa, beta y delta, estiman la posición probable de la presa.
2. Cada lobo del grupo actualiza su distancia respecto de la presa.
3. Se decrementa el parámetro  $a$ .

#### PSEUDOCODIGO

```

Inicializar la población de lobos
Calcular el fitness de los lobos
Determinar los lobos alfa, beta y delta
REPETIR
    Actualizar la posición de cada lobo Ec.8
    Ajustar el parámetro  $a$  Ec.9
    Calcular el fitness de los lobos
    Determinar los lobos alfa, beta y delta
HASTA (condición de parada)
    
```

En el primer paso, **inicializar la población de individuos**, se escogen unos valores aleatorios del espacio de solución para la posición de cada individuo  $X_i$

Aparte de esto también se inicializa el parámetro  $a$ , este factor de exploración comienza con el valor de 2.

En el segundo paso, se calcula el **fitness** de cada individuo, esto se hace en base al valor de la función objetivo en  $X_i$

$$f_i \approx f(X_i)$$

Lo siguiente es decidir los individuos alfa, beta y delta que serán los individuos con mejor fitness.

$X_\alpha$ ,  $X_\beta$  y  $X_\delta$

En el bucle, el primer paso será actualizar la posición de cada individuo. Esta nueva posición será:

$$X_p(t+1) = \frac{X_1 + X_2 + X_3}{3} \quad \text{Ec.8}$$

$$X_1 = X_\alpha - A_1 D_\alpha \quad \text{Ec.4}$$

$$D_\alpha = |C_1 X_\alpha - X_p(t)| \quad \text{Ec.7}$$

$$X_2 = X_\beta - A_2 D_\beta \quad \text{Ec.5}$$

$$D_\alpha = |C_2 X_\beta - X_p(t)| \quad \text{Ec.8}$$

$$X_3 = X_\delta - A_3 D_\delta \quad \text{Ec.6}$$

$$D_\alpha = |C_3 X_\delta - X_p(t)| \quad \text{Ec.9}$$

En donde:

T: es la iteración actual.

$A_i, C_i$  ( $i=1, 2, 3$ ): vectores de coeficientes, calculados usando la [Ec6](#) y la [Ec7](#).

En el siguiente paso, se **actualiza el parámetro** del algoritmo.

El factor de exploración (a) se reduce linealmente en el intervalo  $[2,0]$  al avanzar las iteraciones, según:

$$a = 2 - 2 \frac{t}{max} \quad \text{Ec.9}$$

En donde:

t: es la iteración actual del algoritmo.

max: es el número máximo de iteraciones del algoritmo.[7]

Para **calcular el fitness de cada individuo** se aplica la función objetivo del problema.

Por último, para **encontrar los mejores individuos** (alfa, beta y delta) se toman los 3 individuos con mejor fitness.

#### 4.4.4 Variables

Como en cualquiera de estos algoritmos tenemos **el tamaño de la población** de individuos (I), en este caso es el número de lobos que existen en la población. Un mayor número de individuos puede mejorar la exploración del espacio de búsqueda, pero a su vez aumenta el costo computacional.



*Ilustración 4 Lobos Grises[22]*

En el GWO existen 3 parámetros o coeficientes sociales los cuales ya se han mencionado que son a, A y C.

- a: Disminuye linealmente de 2 a 0 a lo largo de las generaciones y se utiliza para calcular los coeficientes A y C.

- A: Calculado de la forma ([Ec6](#)), es un vector aleatorio cuyos valores están entre el [0,1]. Este coeficiente afecta a la amplitud del movimiento del individuo (lobo) hacia o desde la presa.
- C: Calculado de la forma ([Ec7](#)), este coeficiente influye en el reconocimiento y la localización de la presa, proporcionando un elemento de aleatoriedad en el acercamiento de los lobos a la presa.

#### Jerarquía de Lobos (Alfa, Beta, Delta)

- Alfa ( $\alpha$ ): El lobo líder, que posee la mejor solución encontrada hasta el momento.
- Beta ( $\beta$ ): El segundo mejor lobo, apoya al Alfa en la toma de decisiones y mantiene la disciplina dentro del grupo.
- Delta ( $\delta$ ): El tercero en la jerarquía, juega un rol subordinado pero esencial, facilitando la comunicación entre el Alfa y Beta con el resto de la manada.
- Omega ( $\omega$ ): El resto de los lobos, siguen las direcciones de los lobos con rango superior.

De la misma manera que el resto de los algoritmos, el GWO también intenta optimizar una función, llamada función objetivo que es la que determina la calidad de cada solución.

Por último, los criterios de terminación del problema, que son las condiciones bajo las cuales el algoritmo debe finalizar, ya pueden ser una solución concreta, el número máximo de iteraciones o una limitación del tiempo de ejecución.

## 4.5 Ballenas (WOA)

### 4.5.1 Introducción

El WOA simula la técnica de burbujeo empleada por estos animales, es una estrategia depredadora que utiliza la creación de burbujas en círculos o redes alrededor de su presa antes de atraparla. Este comportamiento es interesante desde un punto de vista de optimización, ya que engloba tanto los elementos de exploración como de explotación, que son las claves para la eficacia en la búsqueda de soluciones óptimas en espacios complejos y multidimensionales.

Este algoritmo posee una estructura simple pero potente que permite resolver una amplia gama de problemas de optimización en ingeniería, informática y más allá. La adaptabilidad y eficiencia del WOA se deben a su capacidad para imitar la dinámica natural de estas ballenas, ajustando sus posiciones en el espacio de búsqueda de manera que maximiza las posibilidades de localizar y converger hacia un óptimo global.

### 4.5.2 Fundamentos

El WOA se basa en el comportamiento de caza de las ballenas jorobadas, particularmente en la técnica conocida como "burbujeo", donde las ballenas crean burbujas en círculos concéntricos alrededor de su presa. Esta técnica no solo confina a la presa, sino que también reduce sus vías de escape, facilitando a la ballena capturarla eficientemente.

#### Modelo Matemático del WOA

El WOA utiliza dos mecanismos principales basados en este comportamiento: la encerrona por burbujeo y los movimientos en espiral hacia la presa. La elección entre estos dos comportamientos se modela mediante un enfoque probabilístico que depende de la proximidad de la solución candidata al mejor candidato actual (la presa).



### **Encerrona por Burbujeo**

En este enfoque, las posiciones de las ballenas se ajustan según la posición del mejor candidato actual. Donde la posición se mueve hacia el mejor candidato o se aleja ligeramente de él basándose en un coeficiente aleatorio.

### **Movimiento en Espiral**

Este comportamiento simula el movimiento en espiral de las ballenas alrededor de su presa. Se utiliza una ecuación helicoidal para actualizar la posición de la ballena en dirección a la presa.

### **Características del WOA**

#### **-Exploración y Explotación:**

El WOA equilibra eficazmente la exploración (buscando en nuevas áreas del espacio de búsqueda) y la explotación (intensificando la búsqueda cerca del óptimo conocido). La alternancia entre el comportamiento de encerrona por burbujeo y el movimiento en espiral permite este equilibrio, haciendo que el WOA sea robusto en encontrar soluciones globales.

#### **-Flexibilidad y Aplicabilidad:**

El algoritmo se ha aplicado en una variedad de problemas, desde optimización de funciones continuas hasta problemas de diseño industrial, demostrando su versatilidad y efectividad.

#### **-Simplicidad de Implementación:**

A pesar de su eficacia, el WOA es relativamente simple de implementar comparado con otros algoritmos metaheurísticos, lo que lo hace accesible para investigadores y profesionales.[8]

### **4.5.3 Definición del algoritmo**

En este algoritmo se comienza por inicializar una población de  $I$  individuos, seguido por la evaluación de estos individuos basada en la función objetivo y se actualizan sus posiciones.

## PSEUDOCODIGO

Inicializar la población de individuos.

Calcular el fitness de los individuos.

Determinar la mejor solución global.

### REPETIR

Se actualiza a. [Ec.10](#) [Ec.11](#) [Ec.12](#)

Actualizar la posición de cada individuo. [Ec.13](#) [Ec.14](#) [Ec.15](#) [Ec.16](#) [Ec.17](#)

Calcular el fitness de los individuos.

Actualizar la mejor solución global.

**HASTA** (condición de parada)

### Inicialización de la población

Para ello se inicializan todos los individuos con posiciones aleatorias en el espacio de búsqueda permitido, después se calcula el fitness de cada individuo según la función objetivo y por último se calcula la mejor solución (individuo con mejor fitness).

También en este paso se inicializan los parámetros del algoritmo, el parámetro  $a$  comienza con un valor de 2 y el  $b$  como 1.

Después de este primer paso se actualiza la posición cada individuo siguiendo las siguientes ecuaciones.

#### Actualización de $a$ :

$$a = 2 - \frac{2 \cdot t}{I} \quad \text{Ec.10}$$

Donde:

$t$  es la iteración actual.

$I$  es el número total de iteraciones permitidas.

#### Actualización de $A$

$$A = 2 a r - a \quad \text{Ec.11}$$

Donde:

$a$  es el parámetro anteriormente calculado

$r$  es un vector aleatorio en  $[0,1]$

#### Actualización de $C$

$$C = 2 r \quad \text{Ec.12}$$

Donde:

$r$  es un vector aleatorio en  $[0,1]$

**Se actualiza la posición respecto a la mejor solución:**

Se calcula el vector  $D$

$$D = |C X^*(t) - X(t)| \quad \text{Ec.13}$$

Donde:

$C$  es el parámetro calculado anteriormente mediante la Ec.12.

$X^*$  es la posición del mejor individuo

$X$  es la posición del individuo que se está evaluando

Con  $D$  se calcula la nueva posición del individuo

$$X(t + 1) = X^*(t) - A D \quad \text{Ec.14}$$

**Se selecciona un individuo aleatorio y se utiliza para actualizar la posición del individuo actual**, para ello se aplica la ecuación:

Para la cual se aplican esta ecuación para calcular  $D$

$$D = |C X^{rand}(t) - X(t)| \quad \text{Ec.15}$$

$$X(t + 1) = X^{rand}(t) - A D \quad \text{Ec.16}$$

Donde:

$X^{rand}$  es un individuo seleccionado aleatoriamente.

**Se mueve el individuo en un camino en espiral alrededor de la mejor solución** mediante esta ecuación que simula el movimiento en espiral hacia la presa:

$$X(t + 1) = D' e^{bl} \cos(2\pi l) + X^* \quad \text{Ec.17}$$

Dónde:

$b$  es un parámetro que usa este algoritmo para definir el movimiento en espiral

$l$  es un número aleatorio en el intervalo  $[-1,1]$

En la Ec.17 se utiliza la siguiente ecuación para calcular  $D'$

$$D' = |X^*(t) - X(t)| \quad \text{Ec.18}$$

## 4.6 Abejas (ABC)

### 4.6.1 Introducción

El Algoritmo de Colonia de Abejas Artificiales (Artificial Bee Colony, ABC) es una técnica de optimización metaheurística que fue desarrollada por Karaboga en 2005. Inspirado por el comportamiento de las abejas, el ABC simula las dinámicas de búsqueda de comida de una colmena de abejas para encontrar soluciones óptimas en problemas

complejos. Este algoritmo se basa en el modelo de división del trabajo y el reclutamiento mutuo entre abejas, características que permiten una exploración eficiente del espacio de soluciones.

Este proceso iterativo de exploración y explotación sigue hasta alcanzar una condición de terminación. Este algoritmo ha demostrado ser efectivo en problemas de optimización numérica y ha sido aplicado en muchos campos como la ingeniería, la investigación operativa y la inteligencia artificial. La simplicidad de este algoritmo sumada a su potente capacidad de optimización ha ayudado en gran parte a su popularidad y gran aceptación en la comunidad científica.[9], [10]

#### 4.6.3- Fundamentos

En el algoritmo ABC se define un conjunto de **F fuentes de alimento**, donde cada fuente representa una posible **solución al problema de optimización**. La posición de cada fuente está asociada a una solución y se considera como una variable clave en la búsqueda de soluciones óptimas.

- **Posición de la fuente:** Representa una solución factible dentro del espacio de búsqueda.
- **Néctar o calidad de la fuente:** La calidad o **fitness** de una fuente es una medida de su valor en términos de la función objetivo, es decir, cuanto mayor es la calidad de la fuente, mejor es la solución que representa.
- **Agotamiento de la fuente:** A medida que se explota una fuente, su **valor de calidad puede no mejorar**. Esto se modela a través de una variable que cuenta los intentos fallidos para mejorar dicha fuente. Si esta variable supera un cierto umbral, la fuente se considera agotada y se reemplaza por una nueva fuente aleatoria.

En el ABC, existen tres tipos de abejas que interactúan con estas fuentes de alimento:

- **Abejas exploradoras:** Se encargan de buscar nuevas fuentes de manera aleatoria, explorando áreas no visitadas previamente.
- **Abejas empleadas:** Se asignan a fuentes específicas y exploran en sus alrededores en busca de mejores soluciones.
- **Abejas observadoras:** Evalúan la información proporcionada por las abejas empleadas y seleccionan las fuentes más prometedoras para explotarlas.

Esta interacción entre los diferentes tipos de abejas y las fuentes de alimento permite al algoritmo buscar, explotar y reemplazar soluciones de manera eficiente a lo largo de las iteraciones.

#### 4.6.4- Definición del algoritmo

El algoritmo ABC sigue una estructura cíclica, donde diferentes tipos de abejas realizan operaciones sobre las fuentes de alimento, buscando mejorar la solución óptima. A continuación, se presenta el pseudocódigo simplificado que resume las principales etapas del algoritmo:

Inicializar el conjunto de fuentes de alimento

**REPETIR**

Operaciones de abejas empleadas [Ec.19](#)

Operaciones de abejas observadoras [Ec.20](#)

Operaciones de abejas exploradoras

Actualizar la mejor solución hasta el momento

**HASTA QUE** (condición de parada)

#### Inicialización de las fuentes de alimento

El proceso de inicialización consiste en seleccionar  $F$  fuentes de alimento dentro del espacio de búsqueda. A cada fuente se le asigna una posición aleatoria.

Además, a cada fuente se le asigna una variable llamada **limit**, que inicialmente comienza con el valor 0. Esta variable se utiliza para definir cuántos intentos se han realizado para mejorar la fuente sin éxito. Si este valor supera un valor previamente definido, la fuente es reemplazada.

#### Operaciones de abejas empleadas.

En el algoritmo ABC, cada **abeja empleada** está asignada a una fuente de alimento, cuya posición en el espacio de búsqueda representa una solución candidata al problema de optimización. Estas abejas exploran la vecindad de la fuente asignada en busca de nuevas soluciones mejores, basándose en la calidad de la fuente (valor de fitness). Si encuentran una fuente de mayor calidad, se mueven hacia ella; de lo contrario, incrementan el valor del contador **limit**, que indica cuántos intentos de mejora han hecho sin éxito. El pseudocódigo de este tipo de abejas sería:

### Pseudocódigo de las Abejas Empleadas:

**Para cada abeja empleada i:**

Obtener una **f fuente candidata**  $v_i$  próxima a la fuente actual  $x_i$  Ec. 19

**Si**  $\text{fitness}(v_i) < \text{fitness}(x_i)$ :

Moverse a la **nueva fuente**  $v_i$ .

**Sino:**

limit = limit + 1

Fin-si

Fin-para

Estas abejas para elegir la fuente candidata próxima, también llamada fuente vecina ( $v_i$ ), siguen esta ecuación:

$$v_i = x_i + \alpha_i(x_i - x_k) \quad \text{Ec. 19}$$

Donde:

$x_k$ : Fuente vecina elegida al azar con  $k \neq i$  (Fuente distinta a la actual)

$\alpha_i$ : número aleatorio en  $[-1,1]$

En cada iteración, las abejas empleadas generan una nueva fuente candidata  $v_i$ , desplazándose desde la posición  $x_i$  hacia la fuente vecina  $x_k$ , ajustada por el valor aleatorio  $\alpha_i$  si esta nueva fuente mejora la solución actual.

### Operaciones de abejas observadoras.

Las **abejas observadoras** eligen las fuentes de alimento que están siendo explotadas por las abejas empleadas, basándose en la calidad o **fitness** de dichas fuentes. La selección de la fuente se realiza en función de una **probabilidad** que refleja la calidad relativa de cada fuente en comparación con el resto. Una vez que una abeja observadora ha seleccionado una fuente, procede de manera similar a una abeja empleada, buscando una solución en la vecindad de la fuente seleccionada.

### Pseudocódigo de las Abejas Observadoras:

**Para cada abeja observadora i:**

Elegir una fuente de alimento  $x_i$  para explotar, en base a una probabilidad  $p_i$  ([Ec. 20](#))

Obtener una fuente candidata  $v_i$  próxima a la fuente actual  $x_i$ .

Si  $\text{fitness}(v_i) < \text{fitness}(x_i)$ :

    Moverse a la nueva fuente  $v_i$ .

Sino

$\text{limit}_i = \text{limit}_i + 1$

Fin-si

Fin-para

### Cálculo de la Probabilidad de Selección

La selección de la fuente de alimento por parte de las abejas observadoras se realiza utilizando una **probabilidad de selección**  $p_i$ , que está determinada por la calidad (fitness) de cada fuente en comparación con el resto de las fuentes. Esta probabilidad se calcula con la siguiente fórmula:

$$p_i = \frac{\text{fitness}_i}{\sum_j \text{fitness}_j} \quad \text{Ec.20}$$

$\text{fitness}_i$ : valor de fitness de la fuente i-ésima

$j$ : contador de fuentes de alimento.

Las abejas observadoras eligen fuentes de alimento de manera probabilística, de acuerdo con la calidad relativa de cada fuente. Fuentes con un mayor valor de fitness tienen una probabilidad más alta de ser seleccionadas.

Después de seleccionar una fuente, esta abeja busca una fuente candidata  $v_i$  cercana utilizando la [Ec. 19](#)

Si la fuente candidata mejora la solución actual, la abeja observadora se mueve hacia ella, de lo contrario, incrementa el contador  $\text{limit}_i$  asociado a la fuente actual.

### Operaciones de abejas exploradoras.

Cuando una fuente de alimento se agota, es decir, no se ha podido mejorar después de un número determinado de intentos, la fuente se abandona y es reemplazada por una nueva fuente generada aleatoriamente. En esta fase, la abeja empleada asignada a esa fuente agotada se convierte en una abeja exploradora y busca una nueva fuente de alimento en el espacio de búsqueda. Esta nueva fuente reemplaza a la fuente agotada.

**Para cada abeja empleada i:**

**Si**  $\text{limit}_i > L$

Generar una **nueva fuente de alimento**  $v_i$ , asignando una nueva posición aleatoria en el espacio de búsqueda.

Reemplazar la fuente actual  $x_i$  por la nueva fuente  $v_i$ .

$\text{limit}_i = 0$

**Fin-si**

**Fin-para**

Dónde  $L$  es la cantidad de intentos fallidos permitidos.



## 5. Tecnologías empleadas

En este apartado se describirán las tecnologías que se han utilizado para el desarrollo de este proyecto.

### 5.1 Python

Lenguaje de alto nivel elegido para la implementación de los algoritmos de optimización por enjambre y procesamiento de imágenes debido a su facilidad de uso, su rica colección de bibliotecas científicas, y su soporte para la creación de entornos virtuales.

Fue creado a finales de los años ochenta por Guido van Rossum en Stichting Mathematisch Centrum (CWI), en Países Bajos. Fue creado como sucesor del lenguaje ABC, capaz de manejar excepciones e interactuar con el sistema operativo Amoeba.[11], [12]

Es un lenguaje interpretado y de propósito general, conocido por su simpleza y legibilidad, lo que lo convierte en una opción popular tanto para principiantes como para desarrolladores experimentados. Se utiliza en una gran variedad de ámbitos, desde el desarrollo web hasta ciencia de datos, inteligencia artificial, blockchain ...



*Ilustración 5 Logo Python [23]*

### 5.2 NumPy

NumPy es una librería de Python especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos.

Incorpora una nueva clase de objetos llamados arrays que permite representar colecciones de datos de un mismo tipo en varias dimensiones, y funciones muy eficientes para su manipulación.

La ventaja de Numpy frente a las listas predefinidas en Python es que el procesamiento de los arrays se realiza mucho más rápido (hasta 50 veces más) que las listas, lo cual la hace ideal para el procesamiento de vectores y matrices de grandes dimensiones.[13]



*Ilustración 6 Logo Numpy[24]*

## 5.3 OpenCv

OpenCV es una librería de computación visual para el procesamiento de imágenes en Python. Esta biblioteca proporciona herramientas para realizar operaciones de procesamiento de imágenes, como el filtrado, la detección de bordes, el reconocimiento de características, el seguimiento de objetos, etc. Estas herramientas nos permiten desarrollar aplicaciones de visión artificial, como el reconocimiento facial, el seguimiento de objetos, etc. [14]

Esta librería ha sido muy útil en este proyecto debido a que permite el procesamiento y manipulación de imágenes. Es fundamental para la cuantificación de color en imágenes.

## 5.4 Scikit-learn

Es una biblioteca de Python que proporciona acceso a versiones eficaces de muchos algoritmos comunes. También proporciona una API propia y estandarizada. Por tanto, una de las grandes ventajas de Scikit-Learn es que una vez que se entiende el uso básico y su sintaxis para un tipo de modelo, cambiar a un nuevo modelo o algoritmo es muy sencillo. La biblioteca no solo permite hacer el modelado, sino que también puede garantizar los pasos de preprocesamiento.

La gran variedad de algoritmos y utilidades de Scikit-learn la convierten en la herramienta básica para empezar a programar y estructurar los sistemas de análisis de datos y modelado estadístico. Los algoritmos de Scikit-Learn se combinan y depuran con otras estructuras de datos y aplicaciones externas como Pandas o PyBrain.

La ventaja de la programación en Python, y Scikit-Learn en concreto, es la variedad de módulos y algoritmos que facilitan el aprendizaje y trabajo del científico de datos en las primeras fases de su desarrollo.[15]

## 5.5 Scikit-image

Esta es otra librería de procesamiento de imágenes de código abierto que está diseñada para operar con las bibliotecas numéricas y científicas de Python como Numpy o SciPy.

El proyecto scikit-image comenzó como scikits.image, de Stéfan van der Walt. Su nombre proviene de la idea de que es un "SciKit" (SciPy Toolkit), una extensión de terceros desarrollada y distribuida por separado para SciPy. El código base original fue posteriormente reescrito en profundidad por otros desarrolladores. De los diversos scikits, scikit-image y scikit-learn fueron descritos como "bien mantenidos y populares" en noviembre de 2012. [16]

## 5.6 Entornos virtuales

Esta característica me ha permitido aislar las dependencias del proyecto, asegurando que las versiones correctas de las librerías se utilicen sin conflictos con otros proyectos.

Este entorno virtual es un “mini-Python” encapsulado dentro de la carpeta del proyecto, al tener su propio intérprete y su propio directorio de paquetes donde se instalan las librerías hace que no afecten ni al Python global ni a otros proyectos en los que se pueda estar trabajando. También garantiza la reproducibilidad, es decir, cualquier colaborador o persona que clone el repositorio y configure el entorno ejecutará las mismas versiones de las librerías usadas.

Gracias al requirements.txt, una vez creado el entorno usando el comando **python -m venv 'nombre'** solo tendrá que ejecutar un **pip install -r requirements.txt** para instalar

las dependencias. Una vez instaladas se debe activar el entorno virtual creado con el comando **source nombreEntorno/bin/actívale** y a continuación ejecutar la herramienta como cualquier archivo Python.

## 5.7 Control de versiones

Para el desarrollo del proyecto se ha utilizado en todo momento la herramienta de GitHub lo que me ha permitido tener un control visual de los cambios que realizaba en el código y poder probar en diferentes entornos de trabajo con rapidez.



*Ilustración 7 Logo GitHub[25]*

## 5.8 Visual Studio Code

Ha sido el editor de código que he usado en todo el proyecto, permitiéndome la integración de los entornos virtuales de Python y la depuración del código.

También me facilitó una manera de pasarle diferentes argumentos requeridos por el programa desde el propio editor formando un JSON con todos ellos.



*Ilustración 9 Logo VisualStudioCode [26]*

```
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Python: Ejecutar Algoritmos",
      "type": "python",
      "request": "launch",
      "program": "${file}",
      "console": "integratedTerminal",
      "args": [
        "textile_market.ppm",
        "32",
        "PSO",
        "MSE",
        "15",
        "10",
        "--pintaImagen", "True"]
    }
  ]
}
```

*Ilustración 8 JSON de argumentos*

## 5.9 Otros

Aparte de todo lo anterior se realizó un script en bash para automatizar el proceso de ejecución de los algoritmos y la generación de resultados. Dicho script lanza el programa con diferentes argumentos y vuelca los resultados en diferentes txt para recuperar los datos para formar los resultados. En estas sucesivas llamadas al programa se cambia el algoritmo que se usa, la función con la que se calcula el fitness y también las imágenes que procesa.

## 6. Pruebas

En este apartado se exponen cómo se han realizado las pruebas de la herramienta desarrollada.

Se parte de que la valoración de la imagen resultante es objetiva, es decir se calcula un valor de error basado en una fórmula matemática.

Para esta herramienta se usan índices que utilizan una imagen de referencia (Full-reference indices) ya que se tiene la imagen original, aunque también existen índices que solo usan la imagen resultante.

Los índices utilizados se describen a continuación.

### 6.1 MSE

El MSE o error cuadrático medio mide el promedio de los errores al cuadrado, es decir, la diferencia entre el estimador y lo que se está estimando. Su ecuación es la siguiente:

$$ECM = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad \text{Ec.20}$$

Dónde:

$n$  es el número total de datos (píxeles, en el caso de imágenes).

$Y_i$  es el valor real del dato  $i$ .

$\hat{Y}_i$  es el valor predicho para el dato  $i$ .

### 6.2 MAE

El MAE o error absoluto medio es una medida de la diferencia entre dos variables continuas. A continuación, se muestra su ecuación:

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_i - \hat{Y}_i| \quad \text{Ec.21}$$

Dónde:

$n$  es el número total de datos (píxeles, en el caso de imágenes).

$Y_i$  es el valor real del dato  $i$ .

$\hat{Y}_i$  es el valor predicho para el dato  $i$ .

### 6.3 SSIM

El SSIM es una métrica diseñada para medir la similitud entre dos imágenes. A diferencia de otras métricas como el MSE, que simplemente cuantifican las diferencias absolutas entre los valores de píxeles, el SSIM se basa en el modo en que los humanos perciben las diferencias estructurales entre las imágenes.

Su ecuación es la siguiente:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad \text{Ec. 22}$$

Dónde:

$\mu_x$  y  $\mu_y$  son las medias de las imágenes  $x$  e  $y$ , respectivamente.

$\sigma_x^2$  y  $\sigma_y^2$  son las varianzas de  $x$  e  $y$ .

$\sigma_{xy}$  es la covarianza entre  $x$  e  $y$ .

$C_1$  y  $C_2$  son constantes para evitar divisiones por cero.

## 6.4 UQI

El UQI (Universal Quality Index) propuesto por Wang & Bovik, evalúa la calidad reconstruida comparando tres atributos que la vista humana identifica como “lo mismo” cuando observa dos imágenes. Basa la comparación entre imagen original y cuantizada en tres aspectos: luminancia, contraste y comparaciones estructurales. [17]

Su fórmula es la que sigue:

$$UQI(x, y) = \frac{4\mu_x\mu_y\mu_{xy}}{(\mu_x^2 + \mu_y^2)(\sigma_x^2 + \sigma_y^2)} \quad \text{Ec. 23}$$

Las variables de la ecuación anterior tienen la misma interpretación descrita para SSIM.

## 6.5 FSIM

FSIM compara dos imágenes fijándose en lo que el ojo realmente usa para reconocer la escena: los puntos de mayor detalle (bordes finos, esquinas, texturas llamativas). Primero detecta esos puntos, los mide para posteriormente comparar los de la imagen original con los de la cuantizada y da prioridad a los pixeles que eran muy salientes.

El resultado está entre 0 y 1. Cuanto más cerca de 1 esté, mejor será la imagen cuantizada.

No se incluye la ecuación del cálculo de este índice ya que utiliza ecuaciones auxiliares y se debería explicar múltiples conceptos que no se pueden abordar en esta memoria de TFG por su extensión. Estas ecuaciones se pueden encontrar en el artículo: FSIM: A feature similarity index for image quality assessment [18]

## 6.6 VIF

Este método da un paso más del FSIM ya que interpreta la imagen como “información visual” que viaja por un canal de comunicación, siendo la fuente la imagen original, el canal los errores producidos por el proceso y el receptor nuestro sistema visual. De esta manera el VIF puede conocer la fracción de la información visual de la imagen original llega intacta a nuestros ojos después del proceso.

Este índice varía entre 0 y 1, siendo 1 el mejor valor.

No se incluye la ecuación del cálculo de este índice ya que utiliza ecuaciones auxiliares y se debería explicar múltiples conceptos que no se pueden abordar en esta memoria de TFG por su extensión. Estas ecuaciones se pueden encontrar en el artículo: A VISUAL INFORMATION FIDELITY APPROACH TO VIDEO QUALITY ASSESSMENT [19]

## 6.7 Criterios y conjuntos de pruebas

El trabajo ha sido realizado en un equipo portátil **SLIMBOOK** que cuenta con las siguientes especificaciones técnicas:

- **Procesador:** Intel Core i7-13700H (13.<sup>a</sup> generación), con 14 núcleos / 20 hilos.
- **Memoria RAM:** 32 GB.
- **Tarjeta gráfica:** Gráficos integrados Intel (controlador Mesa Intel Graphics).

El equipo ejecuta un sistema operativo **GNU/Linux Debian**.

Se han realizado un total de veinte test independientes por cada una de las 100 imágenes del conjunto CQ-100 [20] para cada configuración ALGORITMO\_FUNCION y para cada tamaño de paleta, los tamaños elegidos son: 32, 64, 128 y 256.

La configuración ALGORITMO\_FUNCION se refiere a la combinación de todos los algoritmos de enjambre y funciones implementados (ABA, PSO, FA, GWO, WOA, MSE, MAE, SSIM, UQI).

En total se han hecho pruebas para los 5 algoritmos (ABA, PSO, FA, GWO, WOA) y para las 4 funciones objetivo (MSE, MAE, SSIM, UQI). Después se ha medido la calidad de la imagen cuantizada usando diferentes funciones de fitness (MSE, MAE, FSIM, SSIM, VIF). En total se han hecho las siguientes pruebas:

$$\begin{aligned} & \text{Algoritmos}(5) \times \text{Funciones objetivo}(4) \times \text{Tamaño de paleta}(4) \\ & \times \text{Imágenes}(100) \times \text{test independientes}(20) = 160.000 \text{ pruebas} \end{aligned}$$

Siguiendo estos criterios nos da un conjunto de 160000 pruebas independientes. En este trabajo no se llegaron a realizar todas estas pruebas debido al algoritmo de abejas, esto fue decidido conjuntamente con la tutora debido al tiempo adicional que me llevaría realizarlas, sin embargo, en vez de descartar totalmente dicho método, consideramos interesante incluir los resultados parciales que se habían calculado ya, pues dan una idea clara de lo lento que es el método comparado con los otros.

Como se mencionó anteriormente, para realizar las pruebas se ha utilizado el conjunto CQ-100 propuesto por la tutora de este proyecto [20], que incluye 100 imágenes almacenadas en formato PPM.

Nº imágenes originales	100
Funciones objetivo	MSE, MAE, SSIM, VIF
Tamaños de paleta (N)	32, 64, 128, 256
Algoritmos de enjambre aplicados	PSO, ABC, FA, WOA, GWO
Iteraciones de los algoritmos de enjambre (t)	15
Nº Individuos	20
Rango válido para cada componente de una solución factible de un algoritmo	[0, 255]

Tabla 1 Resumen de pruebas

PSO	$w = 0.729 \quad f_1 = 2.05 \quad f_2 = 2.05 \quad [V_{min}, V_{max}] = [-14, 14]$
FA	$\beta(r_{ik}) = 0.1 \quad \gamma = 0.03$
GWO	$a = 2 - 2 * \frac{t_i}{t}$
WOA	$a = 2 - 2 * \frac{t_i}{t} \quad b = 1$
ABC	$L = t_i * 0.2$

*Tabla 2 Valores de los parámetros específicos para cada algoritmo de enjambre usado en las pruebas*

## 6.8 Análisis

En este apartado se muestra un resumen de los resultados obtenidos en las pruebas realizadas. para ver información ampliada sobre los resultados de las pruebas se puede consultar el material complementario/anexos del proyecto (resumen\_resultados.xlsx).

En primer lugar, se muestra información sobre el tiempo de ejecución de los algoritmos. Se muestra la media, denotada por a (average en inglés) y la desviación típica denotada por d (standard deviation) de los resultados obtenidos por cada configuración que se ha probado.

Los rótulos de la tabla hacen referencia a la configuración probada, por ejemplo, PSO\_MAE es el resultado de ejecutar el algoritmo PSO optimizando la función MAE.

N		PSO_MAE	PSO_MSE	PSO_SSIM	PSO_UQI
32	a	7,08	8,38	20,64	31,21
	d	<b>0,71</b>	<b>0,77</b>	<b>3,88</b>	<b>1,99</b>
64	a	8,88	10,32	21,69	33,1
	d	<b>0,73</b>	<b>0,75</b>	<b>3,46</b>	<b>1,81</b>
128	a	12,29	14,22	25,44	36,75
	d	<b>0,67</b>	<b>12,42</b>	<b>3,38</b>	<b>1,82</b>
256	a	19,33	20,38	32,41	44,55
	d	<b>1,4</b>	<b>0,96</b>	<b>3,42</b>	<b>1,76</b>

Tabla 3 Tiempos PSO

N		FA_MAE	FA_MSE	FA_SSIM	FA_UQI
32	a	7,03	8,38	19,85	31,6
	d	<b>0,71</b>	<b>0,77</b>	<b>3,38</b>	<b>2,11</b>
64	a	8,86	10,42	21,89	33,44
	d	<b>0,7</b>	<b>2,15</b>	<b>3,58</b>	<b>1,91</b>
128	a	12,53	13,7	25,66	37
	d	<b>0,86</b>	<b>0,76</b>	<b>3,78</b>	<b>1,83</b>
256	a	19,2	20,48	32,42	44,93
	d	<b>1,1</b>	<b>0,93</b>	<b>3,54</b>	<b>1,83</b>

Tabla 4 Tiempos FA

N		ABA_MAE	ABA_MSE	ABA_SSIM	ABA_UQI
32	a	130,46	166,46	378,43	
	d	<b>6,62</b>	<b>14,68</b>	<b>46,21</b>	
64	a	170,93	205,77	422,97	
	d	<b>9,85</b>	<b>24,28</b>	<b>9,33</b>	
128	a	247,48	278,84	501,44	
	d	<b>37,83</b>	<b>21,01</b>	<b>10,82</b>	
256	a	407,83	440,52		
	d	<b>15,82</b>	<b>206,15</b>		

Tabla 5 Tiempos ABA



N		GWO_MAE	GWO_MSE	GWO_SSIM	GWO_UQI
32	a	6,97	8,48	20,39	30,39
	d	<b>0,66</b>	<b>0,78</b>	<b>3,64</b>	<b>1,59</b>
64	a	8,83	10,36	21,91	32,66
	d	<b>0,69</b>	<b>0,82</b>	<b>3,59</b>	<b>1,64</b>
128	a	12,39	13,71	25,55	37,61
	d	<b>0,77</b>	<b>0,73</b>	<b>3,67</b>	<b>8,33</b>
256	a	19,01	20,5	32,19	43,99
	d	<b>0,94</b>	<b>0,9</b>	<b>3,44</b>	<b>1,96</b>

Tabla 6 Tiempos GWO

N		WOA_MAE	WOA_MSE	WOA_SSIM	WOA_UQI
32	a	7,06	8,5	20,5	30,97
	d	<b>0,67</b>	<b>0,79</b>	<b>3,59</b>	<b>1,82</b>
64	a	8,92	10,19	22,18	33,41
	d	<b>0,76</b>	<b>0,69</b>	<b>3,58</b>	<b>8,11</b>
128	a	12,36	13,96	25,43	36,42
	d	<b>0,67</b>	<b>0,88</b>	<b>3,57</b>	<b>1,76</b>
256	a	19,16	20,75	32,63	43,93
	d	<b>0,85</b>	<b>1,17</b>	<b>3,66</b>	<b>2,04</b>

Tabla 7 Tiempos WOA

Los tiempos de todas las pruebas están medidos en segundos. Como se puede observar en las tablas 1, 2, 3, 4 y 5 el algoritmo más rápido en ejecutarse fue el GWO que utiliza MAE como función de fitness por una ligera ventaja. La función de fitness que menos tarda en cualquier caso siempre es MAE debido a que es una métrica cuyo calculo tiene bajo coste computacional.

Otra conclusión a la que podemos llegar es que el algoritmo más lento es el ABA sin ninguna duda ya que existe una gran diferencia con el resto de los algoritmos, también podemos observar que, aunque este algoritmo sea el más lento, se repite la conclusión de que el índice MAE es el más rápido de calcular de todos. Aun así, el ABA triplica el tiempo del resto de los algoritmos, lo que indica que el propio mecanismo de exploración/explotación de la colonia añade un coste considerable.

En cuanto al efecto de la paleta elegida podemos observar que el tiempo aumenta llegando a triplicar o cuadruplicar el tiempo si se escoge una paleta de mayor tamaño.

Por último, podemos observar que usar SSIM o UQI como funciones de fitness aumenta el tiempo de ejecución considerablemente, esto es debido a la manera de calcular estos índices ya que el SSIM o el UQI obligan a calcular medias, varianzas y covarianzas, mientras que el MAE o el MSE se resuelven con operaciones elementales.

Las siguientes tablas (de la 6 a la 25) muestran resultados relacionados con los índices de calidad de las imágenes cuantizadas (IQI) obtenidas en las pruebas. para cada una de las imágenes cuantizadas generadas por los algoritmos de enjambres se calcularon 5 índices de calidad: MSE, MAE, SSIM, FSIM y VIF. se observa que MSE, MAE y SSIM se usaron tanto como funciones de fitness como para calcular la calidad de las imágenes generadas. se muestran resultados para estos 5 índices porque están entre los más utilizados en la bibliografía sobre cuantificación de color.

Por cada paleta existen 5 tablas que corresponden a cada algoritmo probado ejecutado con cada una de las 4 funciones objetivo y la calidad de las imágenes cuantizadas medida con diversos índices.

IQI		PSO_MAE	PSO_MSE	PSO_SSIM	PSO_UQI
MSE	a	1436,92	1364,11	2128,31	1839,04
	d	<b>536,01</b>	<b>518,42</b>	<b>1135,22</b>	<b>980,04</b>
MAE	a	48,37	48,73	58,93	55,15
	d	<b>10,51</b>	<b>10,63</b>	<b>14,21</b>	<b>13,1</b>
SSIM	a	0,7349	0,7301	0,7416	0,7232
	d	<b>0,0877</b>	<b>0,093</b>	<b>0,0878</b>	<b>0,0919</b>
FSIM	a	0,854	0,8552	0,8475	0,8443
	d	<b>0,0466</b>	<b>0,0468</b>	<b>0,0495</b>	<b>0,0488</b>
VIF	a	0,355	0,3548	0,3733	0,3469
	d	<b>0,061</b>	<b>0,0618</b>	<b>0,0617</b>	<b>0,0612</b>

Tabla 8 32 colores - PSO

IQI		ABA_MAE	ABA_MSE	ABA_SSIM	ABA_UQI
MSE	a	1385,18	1346,07	1796,73	
	d	<b>399,23</b>	<b>380,1</b>	<b>670,53</b>	
MAE	a	47,85	48,76	54,52	
	d	<b>8,52</b>	<b>8,44</b>	<b>10,01</b>	
SSIM	a	0,7322	0,7257	0,745	
	d	<b>0,0801</b>	<b>0,085</b>	<b>0,0759</b>	
FSIM	a	0,8566	0,8562	0,8506	
	d	<b>0,045</b>	<b>0,0463</b>	<b>0,0497</b>	
VIF	a	0,3492	0,3485	0,355	
	d	<b>0,0571</b>	<b>0,0596</b>	<b>0,0601</b>	

Tabla 9 32 colores - ABA

IQI		FA_MAE	FA_MSE	FA_SSIM	FA_UQI
MSE	a	1856,15	1830,05	2200,41	2118,8
	d	<b>474,07</b>	<b>455,78</b>	<b>846,21</b>	<b>793,21</b>
MAE	a	58,24	58,64	62,9	61,67
	d	<b>7,75</b>	<b>8,07</b>	<b>10,56</b>	<b>10,07</b>
SSIM	a	0,6872	0,6858	0,697	0,6885
	d	<b>0,0949</b>	<b>0,0976</b>	<b>0,0911</b>	<b>0,0935</b>
FSIM	a	0,8349	0,8362	0,8323	0,8311
	d	<b>0,0479</b>	<b>0,048</b>	<b>0,0496</b>	<b>0,0489</b>
VIF	a	0,3269	0,3274	0,3303	0,3242
	d	<b>0,0607</b>	<b>0,0602</b>	<b>0,0609</b>	<b>0,0603</b>

Tabla 10 32 colores - FA

IQI		GWO_MAE	GWO_MSE	GWO_SSIM	GWO_UQI
MSE	a	964,49	919,31	1391,81	1130,93
	d	<b>342,64</b>	<b>330,68</b>	<b>604,56</b>	<b>570,91</b>
MAE	a	39,26	39,79	47,32	42,41
	d	<b>7,92</b>	<b>8,06</b>	<b>10,7</b>	<b>9,88</b>
SSIM	a	0,7713	0,7646	0,7871	0,7731
	d	<b>0,0776</b>	<b>0,0831</b>	<b>0,0755</b>	<b>0,0801</b>
FSIM	a	0,8829	0,884	0,876	0,8744
	d	<b>0,0436</b>	<b>0,043</b>	<b>0,0496</b>	<b>0,0468</b>
VIF	a	0,386	0,3864	0,4009	0,3778
	d	<b>0,0643</b>	<b>0,0633</b>	<b>0,065</b>	<b>0,0629</b>

Tabla 11 32 colores GWO

IQI		WOA_MAE	WOA_MSE	WOA_SSIM	WOA_UQI
MSE	a	1247,75	1197,97	1830,63	1685,63
	d	<b>375,8</b>	<b>370,21</b>	<b>724,13</b>	<b>599,42</b>
MAE	a	44,51	45,32	53,89	51,81
	d	<b>8,75</b>	<b>8,95</b>	<b>10,9</b>	<b>9,82</b>
SSIM	a	0,7495	0,7443	0,7614	0,7418
	d	<b>0,0751</b>	<b>0,0788</b>	<b>0,0729</b>	<b>0,079</b>
FSIM	a	0,8626	0,8631	0,8545	0,848
	d	<b>0,0469</b>	<b>0,0474</b>	<b>0,0503</b>	<b>0,0498</b>
VIF	a	0,3527	0,3536	0,362	0,342
	d	<b>0,0614</b>	<b>0,0618</b>	<b>0,0616</b>	<b>0,0616</b>

Tabla 12 32 colores - WOA

IQI		PSO_MAE	PSO_MSE	PSO_SSIM	PSO_UQI
MSE	a	1012,17	983,89	1451,79	1280,72
	d	<b>337,13</b>	<b>339,12</b>	<b>734,78</b>	<b>632,47</b>
MAE	a	41,15	41,85	48,82	46,54
	d	<b>8,21</b>	<b>8,44</b>	<b>10,94</b>	<b>10,07</b>
SSIM	a	0,7624	0,7573	0,7696	0,7538
	d	<b>0,0843</b>	<b>0,0892</b>	<b>0,0831</b>	<b>0,0869</b>
FSIM	a	0,8819	0,8829	0,8748	0,8737
	d	<b>0,04</b>	<b>0,0411</b>	<b>0,0443</b>	<b>0,0435</b>
VIF	a	0,3906	0,3893	0,4037	0,3808
	d	<b>0,0628</b>	<b>0,0631</b>	<b>0,0627</b>	<b>0,0621</b>

Tabla 13 64 colores - PSO

IQI		ABA_MAE	ABA_MSE	ABA_SSIM	ABA_UQI
MSE	a	931,1	907,14	1184,48	
	d	<b>243</b>	<b>233,97</b>	<b>418,25</b>	
MAE	a	39,51	40,12	44,53	
	d	<b>6,7</b>	<b>6,53</b>	<b>7,92</b>	
SSIM	a	0,7698	0,7651	0,7824	
	d	<b>0,0727</b>	<b>0,0765</b>	<b>0,0694</b>	
FSIM	a	0,8873	0,8883	0,8811	
	d	<b>0,0407</b>	<b>0,0399</b>	<b>0,0449</b>	
VIF	a	0,3882	0,3899	0,3957	
	d	<b>0,06</b>	<b>0,0597</b>	<b>0,0603</b>	

Tabla 14 64 colores - ABA

IQI		FA_MAE	FA_MSE	FA_SSIM	FA_UQI
MSE	a	1193,71	1183,71	1397,99	1369,88
	d	<b>281,49</b>	<b>275,74</b>	<b>499,08</b>	<b>504,86</b>
MAE	a	46,87	47,23	50,3	49,83
	d	<b>5,8</b>	<b>6,12</b>	<b>7,57</b>	<b>7,89</b>
SSIM	a	0,7325	0,7302	0,7392	0,731
	d	<b>0,0889</b>	<b>0,0909</b>	<b>0,0858</b>	<b>0,0885</b>
FSIM	a	0,8713	0,8714	0,8681	0,8673
	d	<b>0,0426</b>	<b>0,0426</b>	<b>0,0453</b>	<b>0,0444</b>
VIF	a	0,3693	0,369	0,3724	0,3659
	d	<b>0,062</b>	<b>0,0615</b>	<b>0,0626</b>	<b>0,0632</b>

Tabla 15 64 colores - FA

IQI		GWO_MAE	GWO_MSE	GWO_SSIM	GWO_UQI
MSE	a	695,21	669,31	974,36	826,36
	d	<b>230,81</b>	<b>225,26</b>	<b>400,46</b>	<b>347,2</b>
MAE	a	33,7	34,14	39,89	36,58
	d	<b>6,39</b>	<b>6,58</b>	<b>8,5</b>	<b>7,86</b>
SSIM	a	0,7986	0,792	0,8098	0,7994
	d	<b>0,0728</b>	<b>0,0788</b>	<b>0,0705</b>	<b>0,0745</b>
FSIM	a	0,9055	0,9062	0,8972	0,8965
	d	<b>0,0381</b>	<b>0,0384</b>	<b>0,0439</b>	<b>0,0422</b>
VIF	a	0,4236	0,424	0,4312	0,4121
	d	<b>0,0658</b>	<b>0,0652</b>	<b>0,0659</b>	<b>0,0651</b>

Tabla 16 64 colores - GWO

IQI		WOA_MAE	WOA_MSE	WOA_SSIM	WOA_UQI
MSE	a	846,95	811,1	1240,62	1187,56
	d	<b>232,21</b>	<b>230,12</b>	<b>525,21</b>	<b>427,01</b>
MAE	a	37	37,36	44,34	43,38
	d	<b>6,99</b>	<b>6,91</b>	<b>9,12</b>	<b>7,97</b>
SSIM	a	0,7854	0,7806	0,7941	0,776
	d	<b>0,0681</b>	<b>0,0719</b>	<b>0,0662</b>	<b>0,0722</b>
FSIM	a	0,8925	0,893	0,8822	0,8763
	d	<b>0,0416</b>	<b>0,0421</b>	<b>0,0464</b>	<b>0,0449</b>
VIF	a	0,3942	0,3958	0,3995	0,3802
	d	<b>0,0627</b>	<b>0,0647</b>	<b>0,064</b>	<b>0,0634</b>

Tabla 17 64 colores - WOA

IQI		PSO_MAE	PSO_MSE	PSO_SSIM	PSO_UQI
MSE	a	699,88	679,69	977,15	855,06
	d	<b>206,99</b>	<b>203,9</b>	<b>549,38</b>	<b>403,07</b>
MAE	a	34,7	35,09	40,13	38,53
	d	<b>6,38</b>	<b>6,2</b>	<b>8,73</b>	<b>7,57</b>
SSIM	a	0,7916	0,7867	0,7999	0,7836
	d	<b>0,0798</b>	<b>0,0837</b>	<b>0,0775</b>	<b>0,0812</b>
FSIM	a	0,9068	0,9078	0,9008	0,9009
	d	<b>0,035</b>	<b>0,0359</b>	<b>0,0398</b>	<b>0,0381</b>
VIF	a	0,4301	0,4292	0,4406	0,4207
	d	<b>0,0634</b>	<b>0,0635</b>	<b>0,0637</b>	<b>0,064</b>

Tabla 18 128 colores - PSO

IQI		ABA_MAE	ABA_MSE	ABA_SSIM	ABA_UQI
MSE	a	619,31	605,47	791,07	
	d	<b>150,32</b>	<b>146,58</b>	<b>325,29</b>	
MAE	a	32,37	32,84	34,98	
	d	<b>5,37</b>	<b>5,18</b>	<b>6,8</b>	
SSIM	a	0,8053	0,803	0,8136	
	d	<b>0,0657</b>	<b>0,0706</b>	<b>0,0499</b>	
FSIM	a	0,9136	0,9149	0,9116	
	d	<b>0,0353</b>	<b>0,0365</b>	<b>0,027</b>	
VIF	a	0,4333	0,4358	0,4463	
	d	<b>0,0616</b>	<b>0,0642</b>	<b>0,0619</b>	

Tabla 19 128 colores ABA

IQI		FA_MAE	FA_MSE	FA_SSIM	FA_UQI
MSE	a	778,23	765,1	878,51	884,99
	d	<b>173,69</b>	<b>168,75</b>	<b>280,87</b>	<b>311,08</b>
MAE	a	37,93	38,06	40,16	40,29
	d	<b>4,48</b>	<b>4,59</b>	<b>5,53</b>	<b>5,87</b>
SSIM	a	0,7709	0,7703	0,7773	0,7691
	d	<b>0,0829</b>	<b>0,0848</b>	<b>0,0808</b>	<b>0,0818</b>
FSIM	a	0,9008	0,9017	0,8989	0,8968
	d	<b>0,038</b>	<b>0,0375</b>	<b>0,0396</b>	<b>0,0394</b>
VIF	a	0,4137	0,4148	0,4175	0,4085
	d	<b>0,0648</b>	<b>0,0638</b>	<b>0,0649</b>	<b>0,0653</b>

Tabla 20 128 colores - FA

IQI		GWO_MAE	GWO_MSE	GWO_SSIM	GWO_UQI
MSE	a	497,82	482,99	678,51	602,25
	d	<b>152,11</b>	<b>149,56</b>	<b>275,72</b>	<b>232,25</b>
MAE	a	28,74	29,13	33,47	31,4
	d	<b>5,18</b>	<b>5,08</b>	<b>6,94</b>	<b>6,21</b>
SSIM	a	0,825	0,818	0,8338	0,8236
	d	<b>0,0672</b>	<b>0,0729</b>	<b>0,0653</b>	<b>0,07</b>
FSIM	a	0,9242	0,9251	0,9169	0,9161
	d	<b>0,0341</b>	<b>0,0337</b>	<b>0,039</b>	<b>0,0376</b>
VIF	a	0,4614	0,4623	0,4667	0,4495
	d	<b>0,0668</b>	<b>0,0666</b>	<b>0,0675</b>	<b>0,0675</b>

Tabla 21 128 colores - GWO

IQI		WOA_MAE	WOA_MSE	WOA_SSIM	WOA_UQI
MSE	a	563,33	547,97	849,04	831,23
	d	<b>142,86</b>	<b>145,26</b>	<b>405,78</b>	<b>289,2</b>
MAE	a	30,35	30,8	36,45	36,22
	d	<b>5,53</b>	<b>5,55</b>	<b>7,92</b>	<b>6,38</b>
SSIM	a	0,8202	0,815	0,826	0,808
	d	<b>0,0615</b>	<b>0,0641</b>	<b>0,0599</b>	<b>0,0652</b>
FSIM	a	0,9175	0,9181	0,9077	0,9018
	d	<b>0,0366</b>	<b>0,0368</b>	<b>0,0412</b>	<b>0,0392</b>
VIF	a	0,44	0,4401	0,4443	0,4221
	d	<b>0,0654</b>	<b>0,0657</b>	<b>0,0645</b>	<b>0,0644</b>

Tabla 22 128 colores - WOA

IQI		PSO_MAE	PSO_MSE	PSO_SSIM	PSO_UQI
MSE	a	469,81	458,49	629,88	578,69
	d	<b>123,64</b>	<b>120,76</b>	<b>312,31</b>	<b>283,07</b>
MAE	a	28,83	29,05	32,46	31,96
	d	<b>4,64</b>	<b>4,53</b>	<b>6,33</b>	<b>5,95</b>
SSIM	a	0,8196	0,8169	0,8283	0,8129
	d	<b>0,076</b>	<b>0,0772</b>	<b>0,0699</b>	<b>0,0764</b>
FSIM	a	0,929	0,9296	0,9241	0,9234
	d	<b>0,0305</b>	<b>0,0311</b>	<b>0,0334</b>	<b>0,0336</b>
VIF	a	0,4747	0,4739	0,4834	0,4636
	d	<b>0,0653</b>	<b>0,0663</b>	<b>0,0633</b>	<b>0,0658</b>

Tabla 23 256 colores - PSO

IQI		ABA_MAE	ABA_MSE	ABA_SSIM	ABA_UQI
MSE	a	411,27	398,46		
	d	<b>92,76</b>	<b>91,01</b>		
MAE	a	26,5	26,68		
	d	<b>4,24</b>	<b>4,11</b>		
SSIM	a	0,8389	0,8337		
	d	<b>0,0591</b>	<b>0,062</b>		
FSIM	a	0,9356	0,9354		
	d	<b>0,0304</b>	<b>0,0304</b>		
VIF	a	0,4834	0,4822		
	d	<b>0,0646</b>	<b>0,0641</b>		

Tabla 24 256 colores - ABA

IQI		FA_MAE	FA_MSE	FA_SSIM	FA_UQI
MSE	a	499,61	493,08	564,71	568,49
	d	<b>104,27</b>	<b>102,22</b>	<b>192,26</b>	<b>194,08</b>
MAE	a	30,54	30,6	32,25	32,36
	d	<b>3,4</b>	<b>3,45</b>	<b>4,55</b>	<b>4,45</b>
SSIM	a	0,808	0,8063	0,8123	0,8064
	d	<b>0,0776</b>	<b>0,0782</b>	<b>0,0745</b>	<b>0,0752</b>
FSIM	a	0,9259	0,9264	0,9238	0,9225
	d	<b>0,0328</b>	<b>0,0324</b>	<b>0,0339</b>	<b>0,0342</b>
VIF	a	0,4641	0,4642	0,4652	0,4585
	d	<b>0,0664</b>	<b>0,0664</b>	<b>0,0661</b>	<b>0,0666</b>

Tabla 25 256 colores - FA

IQI		GWO_MAE	GWO_MSE	GWO_SSIM	GWO_UQI
MSE	a	351,16	343,55	464,52	428,96
	d	<b>97,21</b>	<b>96,6</b>	<b>195,78</b>	<b>152,88</b>
MAE	a	24,29	24,63	27,7	26,66
	d	<b>4,07</b>	<b>4,06</b>	<b>5,71</b>	<b>5,04</b>
SSIM	a	0,8505	0,8443	0,8579	0,8482
	d	<b>0,0609</b>	<b>0,0652</b>	<b>0,0589</b>	<b>0,0632</b>
FSIM	a	0,9408	0,9413	0,9348	0,9333
	d	<b>0,0292</b>	<b>0,0292</b>	<b>0,034</b>	<b>0,0326</b>
VIF	a	0,5039	0,5037	0,5066	0,4904
	d	<b>0,0673</b>	<b>0,0678</b>	<b>0,0687</b>	<b>0,0684</b>

Tabla 26 256 colores - GWO

IQI		WOA_MAE	WOA_MSE	WOA_SSIM	WOA_UQI
MSE	a	375,47	363,4	569,56	574
	d	<b>91,17</b>	<b>90,58</b>	<b>325,9</b>	<b>211,26</b>
MAE	a	24,9	25,13	29,59	29,81
	d	<b>4,45</b>	<b>4,4</b>	<b>6,79</b>	<b>5,19</b>
SSIM	a	0,8519	0,8478	0,8562	0,8387
	d	<b>0,0541</b>	<b>0,0562</b>	<b>0,0533</b>	<b>0,0593</b>
FSIM	a	0,9384	0,9387	0,9296	0,9235
	d	<b>0,0311</b>	<b>0,0313</b>	<b>0,0364</b>	<b>0,0347</b>
VIF	a	0,489	0,4901	0,4927	0,4695
	d	<b>0,0665</b>	<b>0,0668</b>	<b>0,0654</b>	<b>0,0664</b>

Tabla 27 256 colores - WOA

Observando todas estas tablas podemos decir que el GWO es el algoritmo que mejores resultados ofrece. En particular, el algoritmo GWO ejecutado con la función objetivo MSE (GWO\_MSE) alcanza un error cuadrático medio (MAE) medio de 919,31 (paleta 32 colores) frente a los 1364,11 que ofrece el PSO\_MSE.

Ejecutar estos algoritmos con funciones objetivo más complejas como SSIM o UQI duplica o triplica el tiempo de ejecución sin garantizar mejores resultados en cuanto a la calidad de la imagen cuantizada. El caso más extremo de esta situación se da en el algoritmo ABA ejecutado usando SSIM como función objetivo (ABA\_SSIM), cuyo SSIM es de 0,745 (paleta 32 colores) mientras que el algoritmo PSO ejecutado con SSIM como función objetivo (PSO\_SSIM) da un resultado medio de 0,7416 pero esta última configuración es mucho más rápida

En todos los algoritmos el tiempo crece casi linealmente con el número de colores, mientras que la reducción de error se amortigua a partir de 128 colores; por tanto, 128 colores con GWO\_MSE ofrece un compromiso eficiente calidad/tiempo para la mayoría de las aplicaciones.



Se aplicó la prueba no paramétrica de Friedman para comprobar si la función objetivo empleada por cada algoritmo de enjambre afecta de manera significativa a la calidad de la imagen cuantizada. En los puntos siguientes se indica el algoritmo usado y el método con el que se midió la calidad de la imagen cuantizada. El valor p utilizado para la prueba fue de 0,05

### PSO-EMSE

Hipótesis Nula	Decisión
Las distribuciones de PSO_FMAE_EMSE, PSO_FMSE_EMSE, PSO_FSSIM_EMSE y PSO_FUQI_EMSE son iguales.	Rechace la hipótesis nula

Tabla 28 Hipótesis Nula - Decisión - PSO\_EMSE

N total	400
Estadístico de prueba	961,869
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 29 Resumen de la prueba de Friedman PSO\_EMSE

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
PSO_FMSE_EMSE- PSO_FMAE_EMSE	,538	,091	5,888	<,001	,000
PSO_FMSE_EMSE- PSO_FUQI_EMSE	-1,948	,091	-21,334	<,001	,000
PSO_FMSE_EMSE- PSO_FSSIM_EMSE	-2,455	,091	-26,893	<,001	,000
PSO_FMAE_EMSE- PSO_FUQI_EMSE	-1,410	,091	-15,446	<,001	,000
PSO_FMAE_EMSE- PSO_FSSIM_EMSE	-1,918	,091	-21,005	<,001	,000
PSO_FUQI_EMSE- PSO_FSSIM_EMSE	,508	,091	5,559	<,001	,000

Tabla 30 Comparaciones por parejas PSO\_EMSE

Esta prueba resultó altamente significativa lo que confirma que la función objetivo influye de forma notable en la calidad final.

El análisis post-hoc con corrección de Bonferroni muestra diferencias significativas entre todas las parejas de configuraciones ( $p < 0,001$  en los seis contrastes).

### PSO-EMAE

Hipótesis Nula	Decisión
Las distribuciones de PSO_FMAE_EMAE, PSO_FMSE_EMAE, PSO_FSSIM_EMAE y PSO_FUQI_EMAE son iguales.	Rechace la hipótesis nula

Tabla 31 Hipótesis Nula - Decisión - PSO\_EMAE

N total	400
Estadístico de prueba	898,202
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 32 Resumen de la prueba de Friedman PSO\_EMAE

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
PSO_FMAE_EMAE-PSO_FMSE_EMAE	-,374	,091	-4,094	<,001	,000
PSO_FMAE_EMAE-PSO_FUQI_EMAE	-1,983	,091	-21,717	<,001	,000
PSO_FMAE_EMAE-PSO_FSSIM_EMAE	-2,209	,091	-24,196	<,001	,000
PSO_FMSE_EMAE-PSO_FUQI_EMAE	-1,609	,091	-17,623	<,001	,000
PSO_FMSE_EMAE-PSO_FSSIM_EMAE	-1,835	,091	-20,101	<,001	,000
PSO_FUQI_EMAE-PSO_FSSIM_EMAE	,226	,091	2,478	,013	,079

Tabla 33 Comparaciones por parejas PSO\_EMAE

	MSE	MAE	SSIM	UQI
MSE	-			
MAE		-		
SSIM			-	0,079
UQI			0,079	-

Tabla 34 Significación ajustada para PSO\_EMAE

En el caso del PSO\_EMAE, se ve que los pares UQI y SSIM no ofrecen una diferencia significativa mientras que el resto sí.

### PSO-EFSIM

Hipótesis Nula	Decisión
Las distribuciones de PSO_FMAE_EFSIM, PSO_FMSE_EFSIM, PSO_FSSIM_EFSIM y PSO_FUQI_EFSIM son iguales.	Rechace la hipótesis nula

Tabla 35 Hipótesis Nula - Decisión - PSO\_EFSIM

N total	400
Estadístico de prueba	439,398
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 36 Resumen de la prueba de Friedman PSO\_EFSIM

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
PSO_FUQI_EFSIM-PSO_FSSIM_EFSIM	,330	,091	3,615	<,001	,002
PSO_FUQI_EFSIM-PSO_FMAE_EFSIM	1,348	,091	14,761	<,001	,000
PSO_FUQI_EFSIM-PSO_FMSE_EFSIM	1,618	,091	17,719	<,001	,000
PSO_FSSIM_EFSIM-PSO_FMAE_EFSIM	1,018	,091	11,146	<,001	,000
PSO_FSSIM_EFSIM-PSO_FMSE_EFSIM	1,288	,091	14,104	<,001	,000
PSO_FMAE_EFSIM-PSO_FMSE_EFSIM	-,270	,091	-2,958	,003	,019

Tabla 37 Comparaciones por parejas PSO\_EFSIM

	MSE	MAE	SSIM	UQI
MSE	-	0,019		
MAE	0,019	-		
SSIM			-	0,002
UQI			0,002	-

Tabla 38 Significación ajustada para PSO\_EFSIM

En el caso del FSIM para el PSO todos los pares de índices muestran diferencia significativa.

## PSO-ESSIM

Hipótesis Nula	Decisión
Las distribuciones de PSO_FMAE_ESSIM, PSO_FMSE_ESSIM, PSO_FSSIM_ESSIM y PSO_FUQI_ESSIM son iguales.	Rechace la hipótesis nula

Tabla 39 Hipótesis Nula - Decisión - PSO\_ESSIM

N total	400
Estadístico de prueba	570,159
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 40 Resumen de la prueba de Friedman PSO\_ESSIM

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
PSO_FUQI_ESSIM-PSO_FMSE_ESSIM	-,549	,091	-6,011	<,001	,000
PSO_FUQI_ESSIM-PSO_FMAE_ESSIM	1,013	,091	11,091	<,001	,000
PSO_FUQI_ESSIM-PSO_FSSIM_ESSIM	2,094	,091	22,936	<,001	,000
PSO_FMSE_ESSIM-PSO_FMAE_ESSIM	,464	,091	5,080	<,001	,000
PSO_FMSE_ESSIM-PSO_FSSIM_ESSIM	1,545	,091	16,925	<,001	,000
PSO_FMAE_ESSIM-PSO_FSSIM_ESSIM	-1,081	,091	-11,845	<,001	,000

Tabla 41 Comparaciones por parejas PSO\_ESSIM

En el caso del SSIM para el PSO ocurre lo mismo que con el anterior ya que todos los pares de índices muestran diferencia significativa.

## PSO-EVIF

Hipótesis Nula	Decisión
Las distribuciones de PSO_FMAE_EVIF, PSO_FMSE_EVIF, PSO_FSSIM_EVIF y PSO_FUQI_EVIF son iguales.	Rechace la hipótesis nula

Tabla 42 Hipótesis Nula - Decisión - PSO\_EVIF

N total	400
Estadístico de prueba	632,176
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 43 Resumen de la prueba de Friedman PSO\_EVIF

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
PSO_FUQI_EVIF- PSO_FMSE_EVIF	1,136	,091	12,447	<,001	,000
PSO_FUQI_EVIF- PSO_FMAE_EVIF	1,208	,091	13,227	<,001	,000
PSO_FUQI_EVIF- PSO_FSSIM_EVIF	2,291	,091	25,099	<,001	,000
PSO_FMSE_EVIF- PSO_FMAE_EVIF	,071	,091	,781	,435	1,000
PSO_FMSE_EVIF- PSO_FSSIM_EVIF	-1,155	,091	-12,652	<,001	,000
PSO_FMAE_EVIF- PSO_FSSIM_EVIF	-1,084	,091	-11,872	<,001	,000

Tabla 44 Comparaciones por parejas PSO\_EVIF

	MSE	MAE	SSIM	UQI
MSE	-	1		
MAE	1	-		
SSIM			-	
UQI				-

Tabla 45 Significación ajustada para PSO\_EVIF

Para el índice VIF se puede observar que todos los pares muestran diferencia significativa menos el par MAE, MSE.

### TIEMPOS PSO

Hipótesis Nula	Decisión
Las distribuciones de PSO_FMAE_TIEMPO, PSO_FMSE_TIEMPO, PSO_FSSIM_TIEMPO y PSO_FUQI_TIEMPO son iguales.	Rechace la hipótesis nula

Tabla 46 Hipótesis Nula - Decisión - PSO\_TIEMPO

N total	400
Estadístico de prueba	1188,114
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 47 Resumen de la prueba de Friedman PSO\_TIEMPOS

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
PSO_FMAE_ETiempo- PSO_FMSE_ETiempo	-,970	,091	-10,626	<,001	,000
PSO_FMAE_ETiempo- PSO_FSSIM_ETiempo	-1,980	,091	-21,690	<,001	,000
PSO_FMAE_ETiempo- PSO_FUQI_ETiempo	-2,980	,091	-32,644	<,001	,000
PSO_FMSE_ETiempo- PSO_FSSIM_ETiempo	-1,010	,091	-11,064	<,001	,000
PSO_FMSE_ETiempo- PSO_FUQI_ETiempo	-2,010	,091	-22,018	<,001	,000
PSO_FSSIM_ETiempo- PSO_FUQI_ETiempo	-1,000	,091	-10,954	<,001	,000

Tabla 48 Comparaciones por parejas PSO\_ TIEMPOS

Como se observa en la tabla 46 todos los pares muestran diferencias significativas.

#### ABA\_EMSE

Hipótesis Nula	Decisión
Las distribuciones de ABA_FMAE_EMSE, ABA_FMSE_EMSE y ABA_FSSIM_EMSE son iguales.	Rechace la hipótesis nula

Tabla 49 Hipótesis Nula - Decisión - ABA\_EMSE

N total	223
Estadístico de prueba	388,099
Grado de libertad	2
Sig. asintótica (prueba bilateral)	<,001

Tabla 50 Resumen de la prueba de Friedman ABA\_EMSE



Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
ABA_FMSE_EMSE- ABA_FMAE_EMSE	,731	,095	7,718	<,001	,000
ABA_FMSE_EMSE- ABA_FSSIM_EMSE	-1,852	,095	-19,556	<,001	,000
ABA_FMAE_EMSE- ABA_FSSIM_EMSE	-1,121	,095	-11,838	<,001	,000

Tabla 51 Comparaciones por parejas ABA\_EMSE

Como se muestra en la tabla 49 todos los pares de índices AMA\_MSE muestran diferencias significativas.

#### ABA\_EMAE

Hipótesis Nula	Decisión
Las distribuciones de ABA_FMAE_EMAE, ABA_FMSE_EMAE y ABA_FSSIM_EMAE son iguales.	Rechace la hipótesis nula

Tabla 52 Hipótesis Nula - Decisión - ABA\_EMAE

N total	223
Estadístico de prueba	374,861
Grado de libertad	2
Sig. asintótica (prueba bilateral)	<,001

Tabla 53 Resumen de la prueba de Friedman ABA\_EMAE

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
ABA_FMAE_EMAE- ABA_FMSE_EMAE	-,722	,095	-7,624	<,001	,000
ABA_FMAE_EMAE- ABA_FSSIM_EMAE	-1,821	,095	-19,225	<,001	,000
ABA_FMSE_EMAE- ABA_FSSIM_EMAE	-1,099	,095	-11,601	<,001	,000

Tabla 54 Comparaciones por parejas ABA\_EMAE

Como se puede ver en las tres anteriores tablas sucede lo mismo que en el caso anterior, todos los pares cuentan con diferencias significativas.

### ABA\_EFSIM

Hipótesis Nula	Decisión
Las distribuciones de ABA_FMAE_EFSIM, ABA_FMSE_EFSIM y ABA_FSSIM_EFSIM son iguales.	Rechace la hipótesis nula

Tabla 55 Hipótesis Nula - Decisión - ABA\_EFSIM

N total	223
Estadístico de prueba	149,581
Grado de libertad	2
Sig. asintótica (prueba bilateral)	<,001

Tabla 56 Resumen de la prueba de Friedman ABA\_EFSIM

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
ABA_FSSIM_EFSIM-ABA_FMAE_EFSIM	,969	,095	10,228	<,001	,000
ABA_FSSIM_EFSIM-ABA_FMSE_EFSIM	1,029	,095	10,867	<,001	,000
ABA_FMAE_EFSIM-ABA_FMSE_EFSIM	-,061	,095	-,639	,523	1,000

Tabla 57 Comparaciones por parejas ABA\_EFSIM

	MSE	MAE	SSIM	UQI
MSE	-	1		
MAE	1	-		
SSIM			-	
UQI				-

Tabla 56 Significación ajustada para ABA\_EFSIM

En este caso todos los pares tienen diferencia significativa menos los pares MSE, MAE.

## ABA\_ESSIM

Hipótesis Nula	Decisión
Las distribuciones de ABA_FMAE_ESSIM, ABA_FMSE_ESSIM y ABA_FSSIM_ESSIM son iguales.	Rechace la hipótesis nula

Tabla 5758 Hipótesis Nula - Decisión - ABA\_ESSIM

N total	223
Estadístico de prueba	244,074
Grado de libertad	2
Sig. asintótica (prueba bilateral)	<,001

Tabla 59 Resumen de la prueba de Friedman ABA\_ESSIM

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
ABA_FMSE_ESSIM-ABA_FMAE_ESSIM	,460	,095	4,854	<,001	,000
ABA_FMSE_ESSIM-ABA_FSSIM_ESSIM	-1,444	,095	-15,247	<,001	,000
ABA_FMAE_ESSIM-ABA_FSSIM_ESSIM	-,984	,095	-10,394	<,001	,000

Tabla 60 Comparaciones por parejas ABA\_ESSIM

En el caso de ABA\_ESSIM todos los pares cuentan con diferencia significativa.

## ABA\_EVIF

Hipótesis Nula	Decisión
Las distribuciones de ABA_FMAE_EVIF, ABA_FMSE_EVIF y ABA_FSSIM_EVIF son iguales.	Rechace la hipótesis nula

Tabla 61 Hipótesis Nula - Decisión - ABA\_EVIF

N total	223
Estadístico de prueba	34,533
Grado de libertad	2
Sig. asintótica (prueba bilateral)	<,001

Tabla 62 Resumen de la prueba de Friedman ABA\_EVIF

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
ABA_FMAE_EVIF-ABA_FMSE_EVIF	-,025	,095	-,260	,795	1,000
ABA_FMAE_EVIF-ABA_FSSIM_EVIF	-,493	,095	-5,209	<,001	,000
ABA_FMSE_EVIF-ABA_FSSIM_EVIF	-,469	,095	-4,948	<,001	,000

Tabla 63 Comparaciones por parejas ABA\_EVIF

	MSE	MAE	SSIM	UQI
MSE	-	1		
MAE	1	-		
SSIM			-	
UQI				-

Tabla 64 Significación ajustada para ABA\_EVIF

En este caso, igual que como pasa para los pares del PSO\_EVIF, los pares que no tienen diferencia significativa son MSE y MAE, mientras que el resto si tienen.

## ABA\_ETIEMPO

Hipótesis Nula	Decisión
Las distribuciones de ABA_FMAE_ETiempo, ABA_FMSE_ETiempo y ABA_FSSIM_ETiempo son iguales.	Rechace la hipótesis nula

Tabla 62 Hipótesis Nula - Decisión - ABA\_ETIEMPO

N total	223
Estadístico de prueba	442,036
Grado de libertad	2
Sig. asintótica (prueba bilateral)	<,001

Tabla 65 Resumen de la prueba de Friedman ABA\_ETIEMPO

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
ABA_FMAE_ETiempo-ABA_FMSE_ETiempo	-,982	,095	-10,370	<,001	,000
ABA_FMAE_ETiempo-ABA_FSSIM_ETiempo	-1,991	,095	-21,024	<,001	,000
ABA_FMSE_ETiempo-ABA_FSSIM_ETiempo	-1,009	,095	-10,654	<,001	,000

Tabla 66 Comparaciones por parejas ABA\_ETIEMPO

En este caso todos los pares también cuentan con diferencia significativa.

## FA\_EMSE

Hipótesis Nula	Decisión
Las distribuciones de FA_FMAE_EMSE, FA_FMSE_EMSE, FA_FSSIM_EMSE y FA_FUQI_EMSE son iguales.	Rechace la hipótesis nula

Tabla 67 Hipótesis Nula - Decisión - FA\_EMSE

N total	400
Estadístico de prueba	831,495
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 68 Resumen de la prueba de Friedman FA\_EMSE

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
FA_FMSE_EMSE-FA_FMAE_EMSE	,290	,091	3,177	,001	,009
FA_FMSE_EMSE-FA_FSSIM_EMSE	-1,993	,091	-21,827	<,001	,000
FA_FMSE_EMSE-FA_FUQI_EMSE	-1,998	,091	-21,882	<,001	,000
FA_FMAE_EMSE-FA_FSSIM_EMSE	-1,703	,091	-18,650	<,001	,000
FA_FMAE_EMSE-FA_FUQI_EMSE	-1,708	,091	-18,705	<,001	,000
FA_FSSIM_EMSE-FA_FUQI_EMSE	-,005	,091	-,055	,956	1,000

Tabla 69 Comparaciones por parejas FA\_EMSE

	MSE	MAE	SSIM	UQI
MSE	-	0.009		
MAE	0.009	-		
SSIM			-	1
UQI			1	-

Tabla 70 Significación ajustada para FA\_EMSE

Tal y como se muestra en la tabla 68 para este caso los pares que no tienen diferencia significativa son SSIM-UQI.

### FA\_EMAE

Hipótesis Nula	Decisión
Las distribuciones de FA_FMAE_EMAE, FA_FMSE_EMAE, FA_FSSIM_EMAE y FA_FUQI_EMAE son iguales.	Rechace la hipótesis nula

Tabla 71 Hipótesis Nula - Decisión - FA\_EMAE

N total	400
Estadístico de prueba	750,177
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 72 Resumen de la prueba de Friedman FA\_EMAE

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
FA_FMAE_EMAE-FA_FMSE_EMAE	-,305	,091	-3,341	<,001	,005
FA_FMAE_EMAE-FA_FUQI_EMAE	-1,896	,091	-20,772	<,001	,000
FA_FMAE_EMAE-FA_FSSIM_EMAE	-1,914	,091	-20,964	<,001	,000
FA_FMSE_EMAE-FA_FUQI_EMAE	-1,591	,091	-17,431	<,001	,000
FA_FMSE_EMAE-FA_FSSIM_EMAE	-1,609	,091	-17,623	<,001	,000
FA_FUQI_EMAE-FA_FSSIM_EMAE	,018	,091	,192	,848	1,000

Tabla 73 Comparaciones por parejas FA\_EMAE

	MSE	MAE	SSIM	UQI
MSE	-	0.05		
MAE	0.05	-		
SSIM			-	1
UQI			1	-

Tabla 74 Significación ajustada para FA\_EMAE

Para este caso, los contrastes post-hoc indican diferencias en casi todas las configuraciones como en el caso anterior.

Para los pares SSIM-UQI no muestra diferencia significativa y para el par MSE-MAE arroja un valor limítrofe ( $p = 0.05$ ), por tanto, este par puede ser tratado como estadísticamente equivalente en términos de error absoluto. El resto de los pares presentan diferencia significativa.

### FA\_EFSIM

Hipótesis Nula	Decisión
Las distribuciones de FA_FMAE_EFSIM, FA_FMSE_EFSIM, FA_FSSIM_EFSIM y FA_FUQI_EFSIM son iguales.	Rechace la hipótesis nula

Tabla 75 Hipótesis Nula - Decisión - FA\_EFSIM

N total	400
Estadístico de prueba	274,649
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 76 Resumen de la prueba de Friedman FA\_EFSIM



Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
FA_FUQI_EFSIM-FA_FSSIM_EFSIM	,398	,091	4,354	<,001	,000
FA_FUQI_EFSIM-FA_FMAE_EFSIM	1,111	,091	12,173	<,001	,000
FA_FUQI_EFSIM-FA_FMSE_EFSIM	1,321	,091	14,474	<,001	,000
FA_FSSIM_EFSIM-FA_FMAE_EFSIM	,714	,091	7,819	<,001	,000
FA_FSSIM_EFSIM-FA_FMSE_EFSIM	,924	,091	10,119	<,001	,000
FA_FMAE_EFSIM-FA_FMSE_EFSIM	-,210	,091	-2,300	,021	,129

Tabla 77 Comparaciones por parejas FA\_EFSIM

	MSE	MAE	SSIM	UQI
MSE	-	0,129		
MAE	0,129	-		
SSIM			-	
UQI				-

Tabla 78 Significación ajustada para FA\_EFSIM

En este caso los pares que no presentan diferencia significativa son MSE-MAE.

### FA\_ESSIM

Hipótesis Nula	Decisión
Las distribuciones de FA_FMAE_ESSIM, FA_FMSE_ESSIM, FA_FSSIM_ESSIM y FA_FUQI_ESSIM son iguales.	Rechace la hipótesis nula

Tabla 79 Hipótesis Nula - Decisión - FA\_ESSIM

N total	400
Estadístico de prueba	382,474
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 80 Resumen de la prueba de Friedman FA\_ESSIM

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
FA_FUQI_ESSIM-FA_FMSE_ESSIM	,351	,091	3,848	<,001	,001
FA_FUQI_ESSIM-FA_FMAE_ESSIM	,626	,091	6,860	<,001	,000
FA_FUQI_ESSIM-FA_FSSIM_ESSIM	1,688	,091	18,486	<,001	,000
FA_FMSE_ESSIM-FA_FMAE_ESSIM	,275	,091	3,012	,003	,016
FA_FMSE_ESSIM-FA_FSSIM_ESSIM	-1,336	,091	-14,638	<,001	,000
FA_FMAE_ESSIM-FA_FSSIM_ESSIM	-1,061	,091	-11,625	<,001	,000

Tabla 81 Comparaciones por parejas FA\_ESSIM

En este caso todos los pares cuentan con diferencia significativa.

## FA\_EVIF

Hipótesis Nula	Decisión
Las distribuciones de FA_FMAE_EVIF, FA_FMSE_EVIF, FA_FSSIM_EVIF y FA_FUQI_EVIF son iguales.	Rechace la hipótesis nula

Tabla 82 Hipótesis Nula - Decisión - FA\_EVIF

N total	400
Estadístico de prueba	243,587
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 83 Resumen de la prueba de Friedman FA\_EVIF

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
FA_FUQI_EVIF-FA_FMAE_EVIF	,933	,091	10,215	<,001	,000
FA_FUQI_EVIF-FA_FMSE_EVIF	,994	,091	10,886	<,001	,000
FA_FUQI_EVIF-FA_FSSIM_EVIF	1,364	,091	14,939	<,001	,000
FA_FMAE_EVIF-FA_FMSE_EVIF	-,061	,091	-,671	,502	1,000
FA_FMAE_EVIF-FA_FSSIM_EVIF	-,431	,091	-4,724	<,001	,000
FA_FMSE_EVIF-FA_FSSIM_EVIF	-,370	,091	-4,053	<,001	,000

Tabla 84 Comparaciones por parejas FA\_EVIF

	MSE	MAE	SSIM	UQI
MSE	-	1		
MAE	1	-		
SSIM			-	
UQI				-

Tabla 85 Significación ajustada para FA\_EVIF

En este los pares MSE-MAE no cuentan con diferencia significativa, el resto si.

## FA\_ETIEMPO

Hipótesis Nula	Decisión
Las distribuciones de FA_FMAE_ETiempo, FA_FMSE_ETiempo, FA_FSSIM_ETiempo y FA_FUQI_ETiempo son iguales	Rechace la hipótesis nula

Tabla 86 Hipótesis Nula - Decisión - FA\_ETIEMPO

N total	400
Estadístico de prueba	1191,188
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 87 Resumen de la prueba de Friedman FA\_ETIEMPO

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
FA_FMAE_ETiempo- FA_FMSE_ETiempo	-,960	,091	-10,516	<,001	,000
FA_FMAE_ETiempo- FA_FSSIM_ETiempo	-1,980	,091	-21,690	<,001	,000
FA_FMAE_ETiempo- FA_FUQI_ETiempo	-2,980	,091	-32,644	<,001	,000
FA_FMSE_ETiempo- FA_FSSIM_ETiempo	-1,020	,091	-11,174	<,001	,000
FA_FMSE_ETiempo- FA_FUQI_ETiempo	-2,020	,091	-22,128	<,001	,000
FA_FSSIM_ETiempo- FA_FUQI_ETiempo	-1,000	,091	-10,954	<,001	,000

Tabla 88 Comparaciones por parejas FA\_ETIEMPO

En el caso del tiempo para el algoritmo FA, todos los pares presentan diferencias significativas.

## GWO\_EMSE

Hipótesis Nula	Decisión
Las distribuciones de GWO_FMAE_EMSE, GWO_FMSE_EMSE, GWO_FSSIM_EMSE y GWO_FUQI_EMSE son iguales.	Rechace la hipótesis nula

Tabla 89 Hipótesis Nula - Decisión - GWO\_EMSE

N total	400
Estadístico de prueba	836,109
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 90 Resumen de la prueba de Friedman GWO\_EMSE

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
GWO_FMSE_EMSE-GWO_FMAE_EMSE	,593	,091	6,491	<,001	,000
GWO_FMSE_EMSE-GWO_FUQI_EMSE	-1,723	,091	-18,869	<,001	,000
GWO_FMSE_EMSE-GWO_FSSIM_EMSE	-2,385	,091	-26,126	<,001	,000
GWO_FMAE_EMSE-GWO_FUQI_EMSE	-1,130	,091	-12,379	<,001	,000
GWO_FMAE_EMSE-GWO_FSSIM_EMSE	-1,793	,091	-19,636	<,001	,000
GWO_FUQI_EMSE-GWO_FSSIM_EMSE	,663	,091	7,257	<,001	,000

Tabla 91 Comparaciones por parejas GWO\_EMSE

Como se observa en la tabla 89, todos los pares presentan diferencias significativas.

## GWO\_EMAE

Hipótesis Nula	Decisión
Las distribuciones de GWO_FMAE_EMAE, GWO_FMSE_EMAE, GWO_FSSIM_EMAE y GWO_FUQI_EMAE son iguales.	Rechace la hipótesis nula

Tabla 92 Hipótesis Nula - Decisión - GWO\_EMAE

N total	400
Estadístico de prueba	819,894
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 93 Resumen de la prueba de Friedman GWO\_EMAE

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
GWO_FMAE_EMAE-GWO_FMSE_EMAE	-,478	,091	-5,231	<,001	,000
GWO_FMAE_EMAE-GWO_FUQI_EMAE	-1,729	,091	-18,938	<,001	,000
GWO_FMAE_EMAE-GWO_FSSIM_EMAE	-2,294	,091	-25,127	<,001	,000
GWO_FMSE_EMAE-GWO_FUQI_EMAE	-1,251	,091	-13,707	<,001	,000
GWO_FMSE_EMAE-GWO_FSSIM_EMAE	-1,816	,091	-19,896	<,001	,000
GWO_FUQI_EMAE-GWO_FSSIM_EMAE	,565	,091	6,189	<,001	,000

Tabla 94 Comparaciones por parejas GWO\_EMAE

Tal y como se observa en la tabla 92 todos los pares presentan diferencias significativas.

## GWO\_EFSIM

Hipótesis Nula	Decisión
Las distribuciones de GWO_FMAE_EFSIM, GWO_FMSE_EFSIM, GWO_FSSIM_EFSIM y GWO_FUQI_EFSIM son iguales.	Rechace la hipótesis nula

Tabla 95 Hipótesis Nula - Decisión - GWO\_EFSIM

N total	400
Estadístico de prueba	448,338
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 96 Resumen de la prueba de Friedman GWO\_EFSIM

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
GWO_FUQI_EFSIM-GWO_FSSIM_EFSIM	,200	,091	2,191	,028	,171
GWO_FUQI_EFSIM-GWO_FMAE_EFSIM	1,318	,091	14,432	<,001	,000
GWO_FUQI_EFSIM-GWO_FMSE_EFSIM	1,573	,091	17,226	<,001	,000
GWO_FSSIM_EFSIM-GWO_FMAE_EFSIM	1,118	,091	12,242	<,001	,000
GWO_FSSIM_EFSIM-GWO_FMSE_EFSIM	1,373	,091	15,035	<,001	,000
GWO_FMAE_EFSIM-GWO_FMSE_EFSIM	-,255	,091	-2,793	,005	,031

Tabla 97 Comparaciones por parejas GWO\_EFSIM

	MSE	MAE	SSIM	UQI
MSE	-	0.031		
MAE	0.031	-		
SSIM			-	0.171
UQI			0.171	-

Tabla 98 Significación ajustada para GWO\_EFSIM

Como se muestra en la tabla 96 los pares que no muestran diferencia significativa son SSIM-UQI, mientras que el resto sí.

### GWO\_ESSIM

Hipótesis Nula	Decisión
Las distribuciones de GWO_FMAE_ESSIM, GWO_FMSE_ESSIM, GWO_FSSIM_ESSIM y GWO_FUQI_ESSIM son iguales.	Rechace la hipótesis nula

Tabla 99 Hipótesis Nula - Decisión - GWO\_ESSIM

N total	400
Estadístico de prueba	446,825
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 100 Resumen de la prueba de Friedman GWO\_ESSIM

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
GWO_FMSE_ESSIM-GWO_FUQI_ESSIM	-,238	,091	-2,602	,009	,056
GWO_FMSE_ESSIM-GWO_FMAE_ESSIM	,643	,091	7,038	<,001	,000
GWO_FMSE_ESSIM-GWO_FSSIM_ESSIM	-1,775	,091	-19,444	<,001	,000
GWO_FUQI_ESSIM-GWO_FMAE_ESSIM	,405	,091	4,437	<,001	,000
GWO_FUQI_ESSIM-GWO_FSSIM_ESSIM	1,538	,091	16,842	<,001	,000
GWO_FMAE_ESSIM-GWO_FSSIM_ESSIM	-1,133	,091	-12,406	<,001	,000

Tabla 101 Comparaciones por parejas GWO\_ESSIM



	MSE	MAE	SSIM	UQI
MSE	-			0.056
MAE		-		
SSIM			-	
UQI	0.056			-

Tabla 102 Significación ajustada para GWO\_ESSIM

Como se observa en la tabla 100 el único par que no muestra diferencia significativa para este caso es el par MSE-UQI.

### GWO\_EVIF

Hipótesis Nula	Decisión
Las distribuciones de GWO_FMAE_EVIF, GWO_FMSE_EVIF, GWO_FSSIM_EVIF y GWO_FUQI_EVIF son iguales.	Rechace la hipótesis nula

Tabla 103 Hipótesis Nula - Decisión - GWO\_EVIF

N total	400
Estadístico de prueba	417,702
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 104 Resumen de la prueba de Friedman GWO\_EVIF

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
GWO_FUQI_EVIF-GWO_FMAE_EVIF	1,133	,091	12,406	<,001	,000
GWO_FUQI_EVIF-GWO_FMSE_EVIF	1,156	,091	12,666	<,001	,000
GWO_FUQI_EVIF-GWO_FSSIM_EVIF	1,836	,091	20,115	<,001	,000
GWO_FMAE_EVIF-GWO_FMSE_EVIF	-,024	,091	-,260	,795	1,000
GWO_FMAE_EVIF-GWO_FSSIM_EVIF	-,704	,091	-7,709	<,001	,000
GWO_FMSE_EVIF-GWO_FSSIM_EVIF	-,680	,091	-7,449	<,001	,000

Tabla 105 Comparaciones por parejas GWO\_EVIF

	MSE	MAE	SSIM	UQI
MSE	-	1		
MAE	1	-		
SSIM			-	
UQI				-

Tabla 106 Significación ajustada para GWO\_EVIF

En este caso, tal y como se observa en la tabla 104, los pares que no cuentan con diferencia significativa son MSE-MAE.

### **GWO\_ETIEMPO**

Hipótesis Nula	Decisión
Las distribuciones de GWO_FMAE_ETiempo, GWO_FMSE_ETiempo, GWO_FSSIM_ETiempo y GWO_FUQI_ETiempo son iguales.	Rechace la hipótesis nula

Tabla 107 Hipótesis Nula - Decisión - GWO\_ETIEMPO

N total	400
Estadístico de prueba	1198,803
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 108 Resumen de la prueba de Friedman GWO\_ETIEMPO

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
GWO_FMAE_ETiempo-GWO_FMSE_ETiempo	-,995	,091	-10,900	<,001	,000
GWO_FMAE_ETiempo-GWO_FSSIM_ETiempo	-1,998	,091	-21,882	<,001	,000
GWO_FMAE_ETiempo-GWO_FUQI_ETiempo	-2,998	,091	-32,836	<,001	,000
GWO_FMSE_ETiempo-GWO_FSSIM_ETiempo	-1,003	,091	-10,982	<,001	,000
GWO_FMSE_ETiempo-GWO_FUQI_ETiempo	-2,003	,091	-21,936	<,001	,000
GWO_FSSIM_ETiempo-GWO_FUQI_ETiempo	-1,000	,091	-10,954	<,001	,000

Tabla 109 Comparaciones por parejas GWO\_ETIEMPO

Como se observa en la tabla anterior, todos los pares muestran diferencia significativa.

### WOA\_EMSE

Hipótesis Nula	Decisión
Las distribuciones de WOA_FMAE_EMSE, WOA_FMSE_EMSE, WOA_FSSIM_EMSE y WOA_FUQI_EMSE son iguales.	Rechace la hipótesis nula

Tabla 110 Hipótesis Nula - Decisión - WOA\_EMSE

N total	400
Estadístico de prueba	990,585
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 111 Resumen de la prueba de Friedman WOA\_EMSE

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
WOA_FMSE_EMSE-WOA_FMAE_EMSE	,628	,091	6,874	<,001	,000
WOA_FMSE_EMSE-WOA_FSSIM_EMSE	-2,285	,091	-25,031	<,001	,000
WOA_FMSE_EMSE-WOA_FUQI_EMSE	-2,308	,091	-25,277	<,001	,000
WOA_FMAE_EMSE-WOA_FSSIM_EMSE	-1,658	,091	-18,157	<,001	,000
WOA_FMAE_EMSE-WOA_FUQI_EMSE	-1,680	,091	-18,403	<,001	,000
WOA_FSSIM_EMSE-WOA_FUQI_EMSE	-,023	,091	-,246	,805	1,000

Tabla 112 Comparaciones por parejas WOA\_EMSE

	MSE	MAE	SSIM	UQI
MSE	-			
MAE		-		
SSIM			-	1
UQI			1	-

Tabla 113 Significación ajustada para WOA\_EMSE

Para este caso los pares que no cuentan con diferencia significativa son SSIM-UQI.

## WOA\_EMAE

Hipótesis Nula	Decisión
Las distribuciones de WOA_FMAE_EMAE, WOA_FMSE_EMAE, WOA_FSSIM_EMAE y WOA_FUQI_EMAE son iguales.	Rechace la hipótesis nula

Tabla 114 Hipótesis Nula - Decisión - WOA\_EMAE

N total	400
Estadístico de prueba	983,342
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 115 Resumen de la prueba de Friedman WOA\_EMAE

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
WOA_FMAE_EMAE-WOA_FMSE_EMAE	-,515	,091	-5,642	<,001	,000
WOA_FMAE_EMAE-WOA_FSSIM_EMAE	-2,233	,091	-24,456	<,001	,000
WOA_FMAE_EMAE-WOA_FUQI_EMAE	-2,263	,091	-24,784	<,001	,000
WOA_FMSE_EMAE-WOA_FSSIM_EMAE	-1,718	,091	-18,814	<,001	,000
WOA_FMSE_EMAE-WOA_FUQI_EMAE	-1,748	,091	-19,143	<,001	,000
WOA_FSSIM_EMAE-WOA_FUQI_EMAE	-,030	,091	-,329	,742	1,000

Tabla 116 Comparaciones por parejas WOA\_EMAE

	MSE	MAE	SSIM	UQI
MSE	-			
MAE		-		
SSIM			-	1
UQI			1	-

Tabla 117 Significación ajustada para WOA\_EMAE

Para este caso, como se puede observar en la tabla 115, los pares que no cuentan con diferencia significativa son SSIM-UQI.

### WOA\_EFSIM

Hipótesis Nula	Decisión
Las distribuciones de WOA_FMAE_EFSIM, WOA_FMSE_EFSIM, WOA_FSSIM_EFSIM y WOA_FUQI_EFSIM son iguales.	Rechace la hipótesis nula

Tabla 118 Hipótesis Nula - Decisión - WOA\_EFSIM

N total	400
Estadístico de prueba	771,914
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 119 Resumen de la prueba de Friedman WOA\_EFSIM

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
WOA_FUQI_EFSIM-WOA_FSSIM_EFSIM	,624	,091	6,833	<,001	,000
WOA_FUQI_EFSIM-WOA_FMAE_EFSIM	1,971	,091	21,594	<,001	,000
WOA_FUQI_EFSIM-WOA_FMSE_EFSIM	2,120	,091	23,223	<,001	,000
WOA_FSSIM_EFSIM-WOA_FMAE_EFSIM	1,348	,091	14,761	<,001	,000
WOA_FSSIM_EFSIM-WOA_FMSE_EFSIM	1,496	,091	16,391	<,001	,000
WOA_FMAE_EFSIM-WOA_FMSE_EFSIM	-,149	,091	-1,629	,103	,619

Tabla 120 Comparaciones por parejas WOA\_EFSIM

	MSE	MAE	SSIM	UQI
MSE	-	0.619		
MAE	0.619	-		
SSIM			-	
UQI				-

Tabla 121 Significación ajustada para WOA\_EFSIM

Para este caso los únicos pares que no cuentan con diferencia significativa es MSE\_MAE.

### WOA\_ESSIM

Hipótesis Nula	Decisión
Las distribuciones de WOA_FMAE_ESSIM, WOA_FMSE_ESSIM, WOA_FSSIM_ESSIM y WOA_FUQI_ESSIM son iguales.	Rechace la hipótesis nula

Tabla 122 Hipótesis Nula - Decisión - WOA\_ESSIM

N total	400
Estadístico de prueba	622,482
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 123 Resumen de la prueba de Friedman WOA\_ESSIM

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
WOA_FUQI_ESSIM-WOA_FMSE_ESSIM	,751	,091	8,230	<,001	,000
WOA_FUQI_ESSIM-WOA_FMAE_ESSIM	1,291	,091	14,145	<,001	,000
WOA_FUQI_ESSIM-WOA_FSSIM_ESSIM	2,208	,091	24,182	<,001	,000
WOA_FMSE_ESSIM-WOA_FMAE_ESSIM	,540	,091	5,915	<,001	,000
WOA_FMSE_ESSIM-WOA_FSSIM_ESSIM	-1,456	,091	-15,952	<,001	,000
WOA_FMAE_ESSIM-WOA_FSSIM_ESSIM	-,916	,091	-10,037	<,001	,000

Tabla 124 Comparaciones por parejas WOA\_ESSIM

Como se observa en la tabla 122, todos los pares cuentan con diferencia significativa.

### WOA\_EVIF

Hipótesis Nula	Decisión
Las distribuciones de WOA_FMAE_EVIF, WOA_FMSE_EVIF, WOA_FSSIM_EVIF y WOA_FUQI_EVIF son iguales.	Rechace la hipótesis nula

Tabla 125 Hipótesis Nula - Decisión - WOA\_EVIF

N total	400
Estadístico de prueba	525,379
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 126 Resumen de la prueba de Friedman WOA\_EVIF



Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
WOA_FUQI_EVIF-WOA_FMAE_EVIF	1,323	,091	14,487	<,001	,000
WOA_FUQI_EVIF-WOA_FMSE_EVIF	1,416	,091	15,514	<,001	,000
WOA_FUQI_EVIF-WOA_FSSIM_EVIF	2,026	,091	22,196	<,001	,000
WOA_FMAE_EVIF-WOA_FMSE_EVIF	-,094	,091	-1,027	,304	1,000
WOA_FMAE_EVIF-WOA_FSSIM_EVIF	-,704	,091	-7,709	<,001	,000
WOA_FMSE_EVIF-WOA_FSSIM_EVIF	-,610	,091	-6,682	<,001	,000

Tabla 127 Comparaciones por parejas WOA\_EVIF

	MSE	MAE	SSIM	UQI
MSE	-	1		
MAE	1	-		
SSIM			-	
UQI				-

Tabla 128 Significación ajustada para WOA\_EVIF

En este caso los pares que no cuentan con diferencia significativa son MSE-MAE.

## WOA\_ETIEMPO

Hipótesis Nula	Decisión
Las distribuciones de WOA_FMAE_ETiempo, WOA_FMSE_ETiempo, WOA_FSSIM_ETiempo y WOA_FUQI_ETiempo son iguales.	Rechace la hipótesis nula

Tabla 129 Hipótesis Nula - Decisión - WOA\_ETIEMPO

N total	400
Estadístico de prueba	1198,803
Grado de libertad	3
Sig. asintótica (prueba bilateral)	<,001

Tabla 130 Resumen de la prueba de Friedman WOA\_ETIEMPO

Sample 1-Sample 2	Estadístico de prueba	Error estándar	Estadístico de prueba estándar	Sig.	Sig. ajust. <sup>a</sup>
WOA_FMAE_ETiempo-WOA_FMSE_ETiempo	-,995	,091	-10,900	<,001	,000
WOA_FMAE_ETiempo-WOA_FSSIM_ETiempo	-1,998	,091	-21,882	<,001	,000
WOA_FMAE_ETiempo-WOA_FUQI_ETiempo	-2,998	,091	-32,836	<,001	,000
WOA_FMSE_ETiempo-WOA_FSSIM_ETiempo	-1,003	,091	-10,982	<,001	,000
WOA_FMSE_ETiempo-WOA_FUQI_ETiempo	-2,003	,091	-21,936	<,001	,000
WOA_FSSIM_ETiempo-WOA_FUQI_ETiempo	-1,000	,091	-10,954	<,001	,000

Tabla 131 Comparaciones por parejas WOA\_ETIEMPO

En este caso todos los pares cuentan con diferencia significativa.

En las siguientes páginas se muestran los diagramas de caja para cada índice de calidad, para todos los algoritmos de enjambre evaluados y, dentro de cada algoritmo, se muestran las cuatro paletas con las que se han realizado las pruebas. Para cada paleta se comparan las cuatro funciones objetivo con las que se ejecutó el algoritmo.

Como se puede ver en las siguientes ilustraciones (de la ilustración 10 a la 14) el GWO cuenta con ventaja ya que presenta los mejores resultados en todos los índices obtenidos, también muestra la dispersión (altura de las cajas) más reducida, lo que significa unos resultados consistentes entre imágenes.

El algoritmo de luciérnagas (FA) es el peor ya que ha obtenido las peores puntuaciones, además, sus diagramas de cajas muestran los bigotes más largos y varios valores atípicos, esto quiere decir que es menos robusto.

El algoritmo WOA obtiene un término medio interesante ya que mejora al PSO y ABA en error medio y obtiene un buen SSIM. El tiempo medio de ejecución sigue siendo moderado, por lo que podría ser otra buena alternativa.

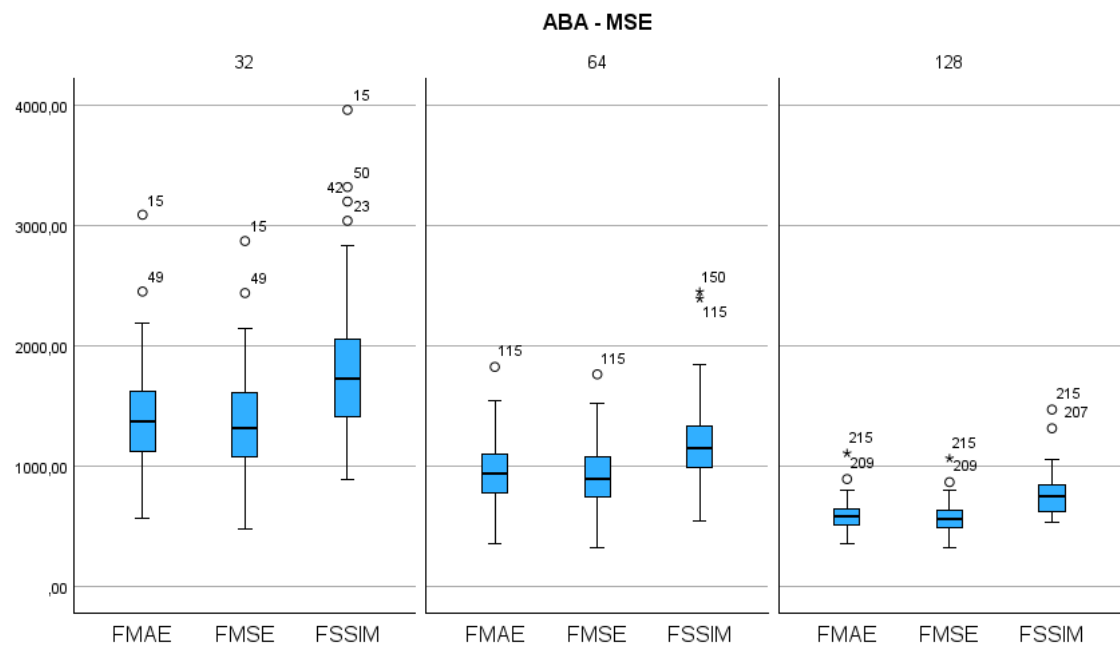
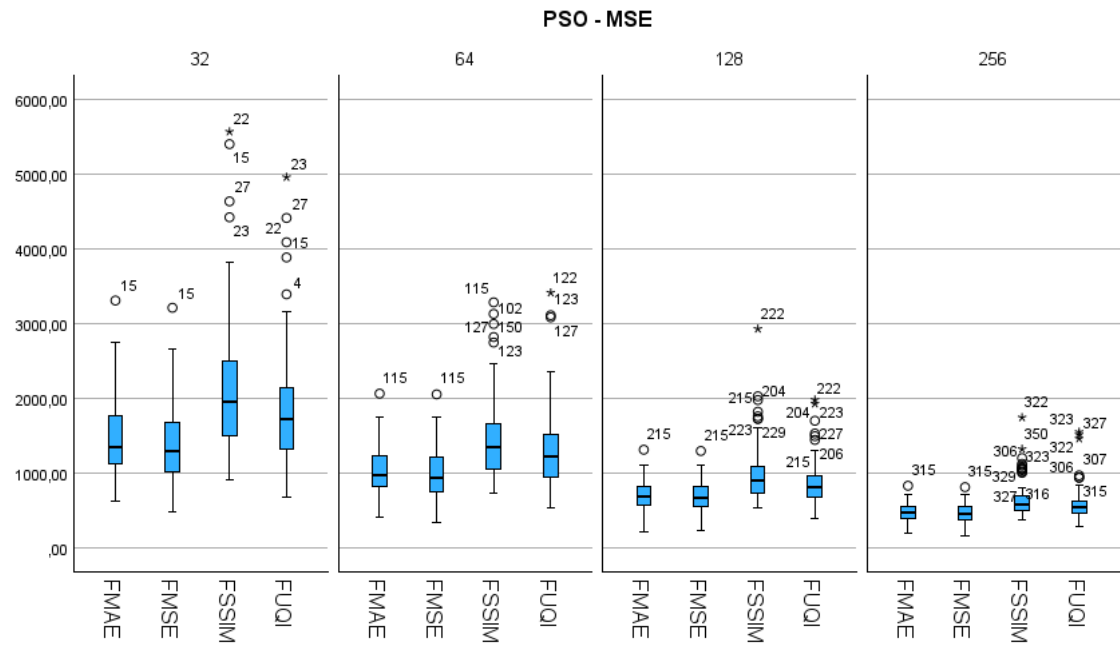
PSO y ABA quedan en un puesto intermedio ya que ambos obtienen errores similares y aceptables, pero no alcanzan la precisión del GWO ni la robustez del WOA.

Sobre que función objetivo ejecutar se encuentra una decisión clara, debe coincidir la función objetivo con el índice de evaluación, en todos los casos se obtiene la mejor mediana y menor dispersión.

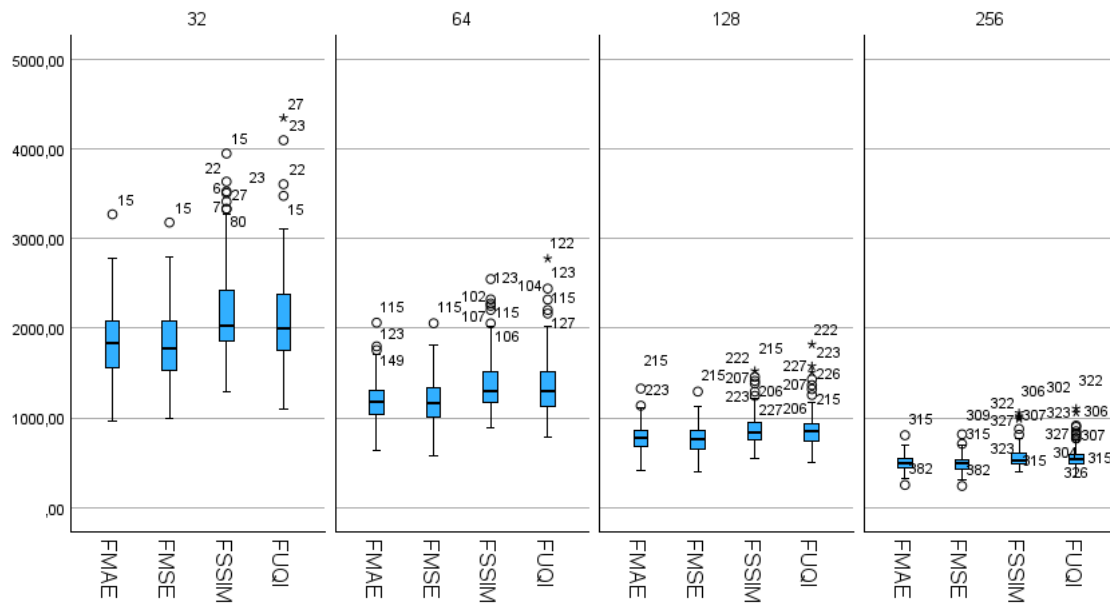
Si la función objetivo es el MSE no solo minimiza el MSE, sino que también deja unos valores del MAE muy cercanos a los que se obtienen cuando la función objetivo es el MSE. En cambio, si la función objetivo es SSIM/UQI se mejoran los índices SSIM y FSIM (similitud perceptual), este beneficio hace que se obtengan peores resultados medios de MSE y MAE en paletas de 32 y 64 colores. Esto ocurre porque el algoritmo ajusta la paleta para maximizar la similitud estructural de cada bloque de píxeles, aunque esto implique un error numérico mayor en algunos colores. Por tanto, optimizar SSIM/UQI es recomendable en aplicaciones donde la apariencia visual prima sobre la exactitud numérica (cuán idénticos son, número por número, los componentes RGB originales y los cuantizados).

En todos los casos se observa como al aumentar el número de colores de la paleta se obtienen mejores resultados a costa de un mayor tiempo de ejecución.

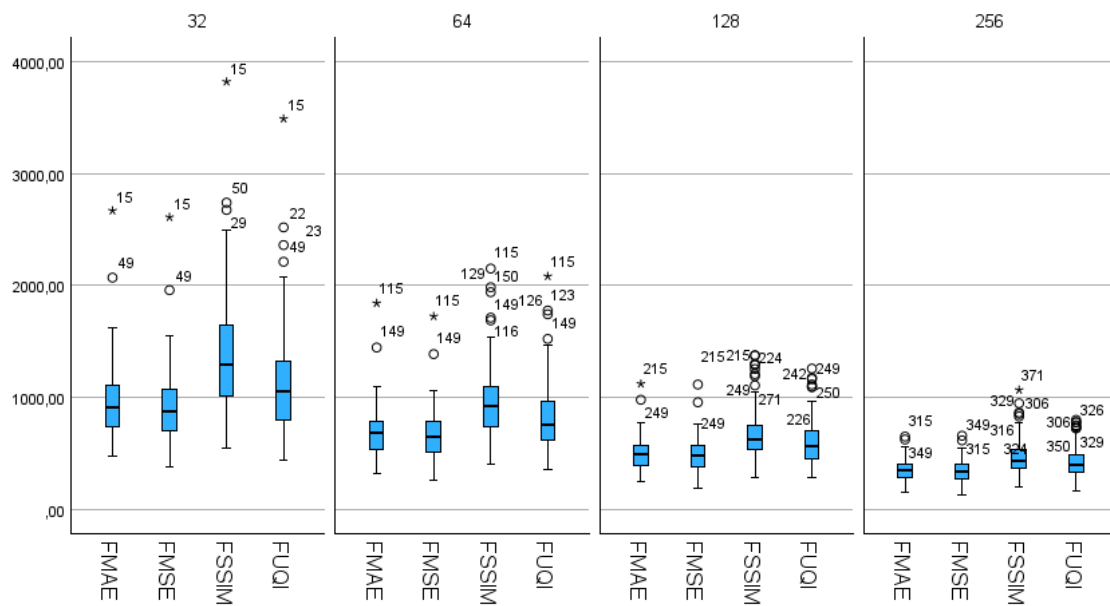
En cuanto al tiempo no hay mucha diferencia entre todos los algoritmos evaluados, a excepción del ABA que es mucho más lento. Según los resultados obtenidos no tiene sentido usarlo ya que tampoco ofrece una mejora de calidad perceptual. En cuanto a la función objetivo en todos los casos se obtiene que los índices SSIM y UQI son los más lentos.

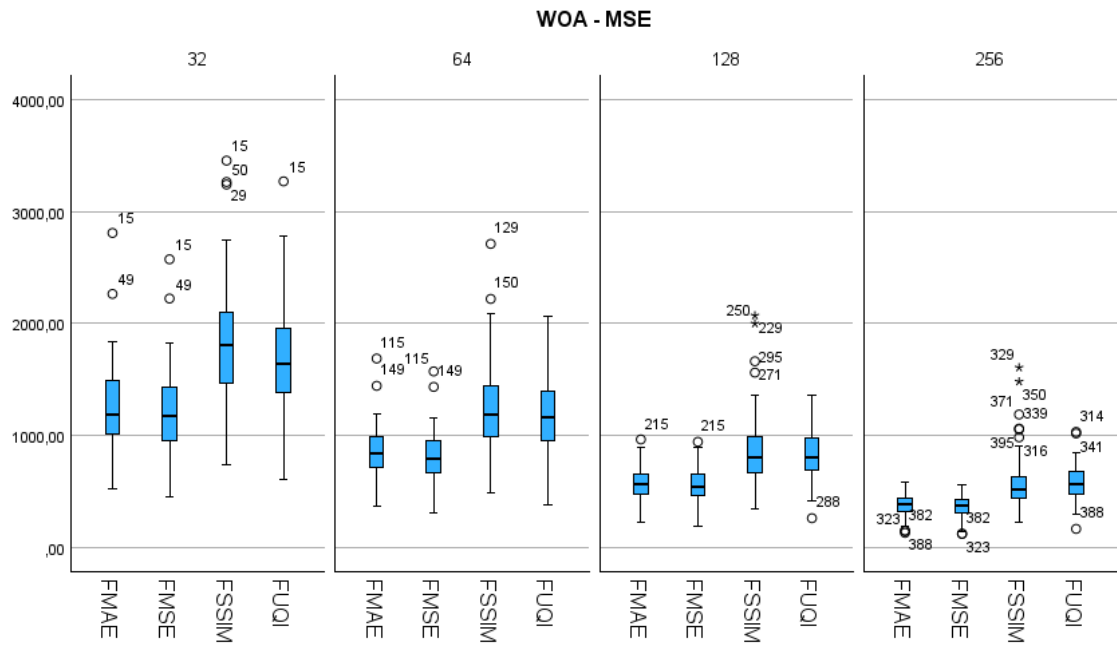


# FA - MSE

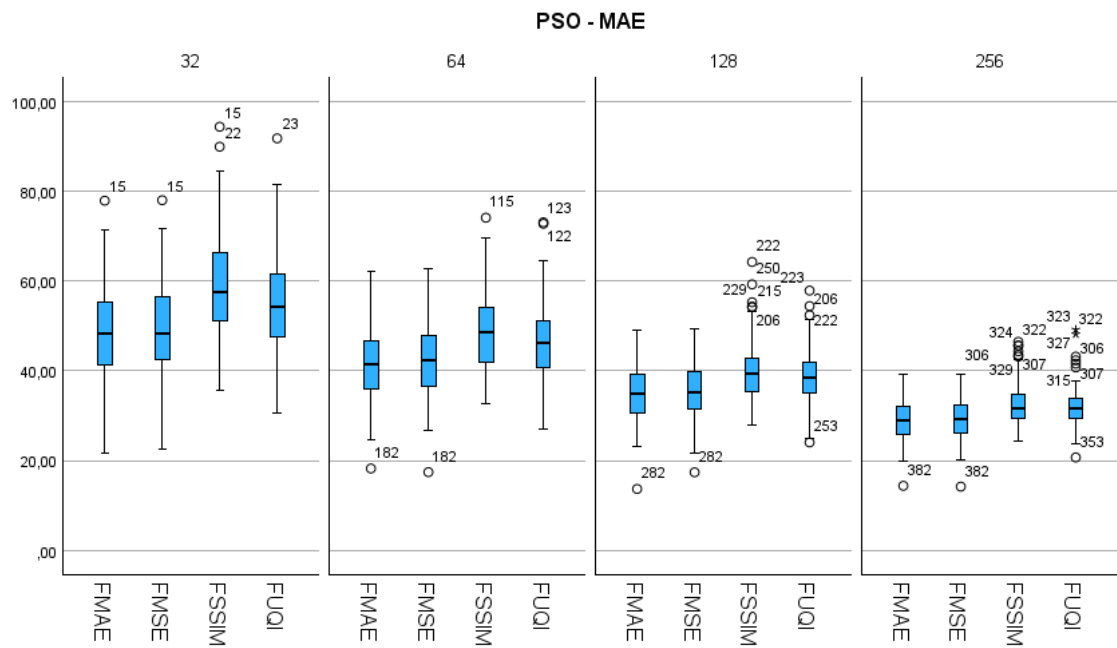


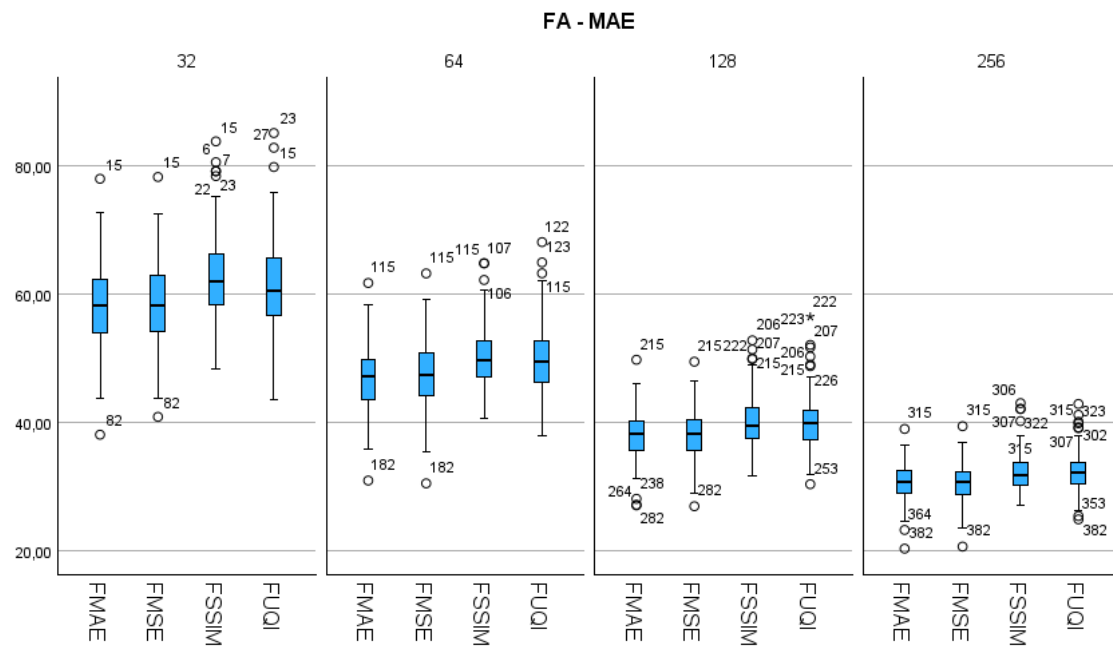
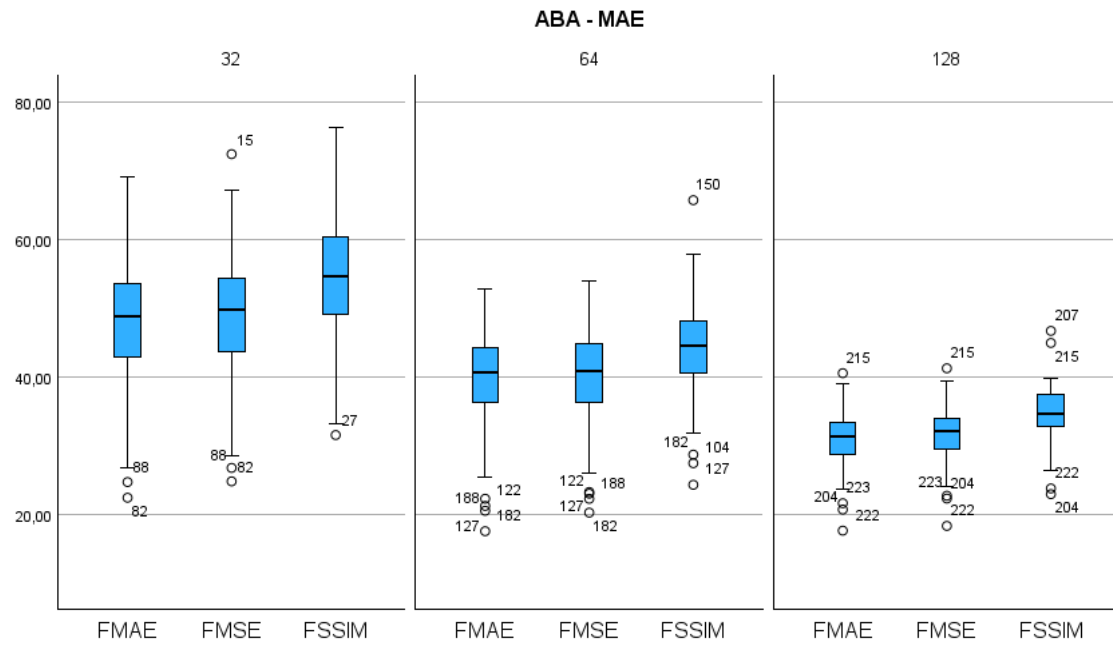
# GWO - MSE





*Ilustración 10 Índices MSE para todos los algoritmos probados*





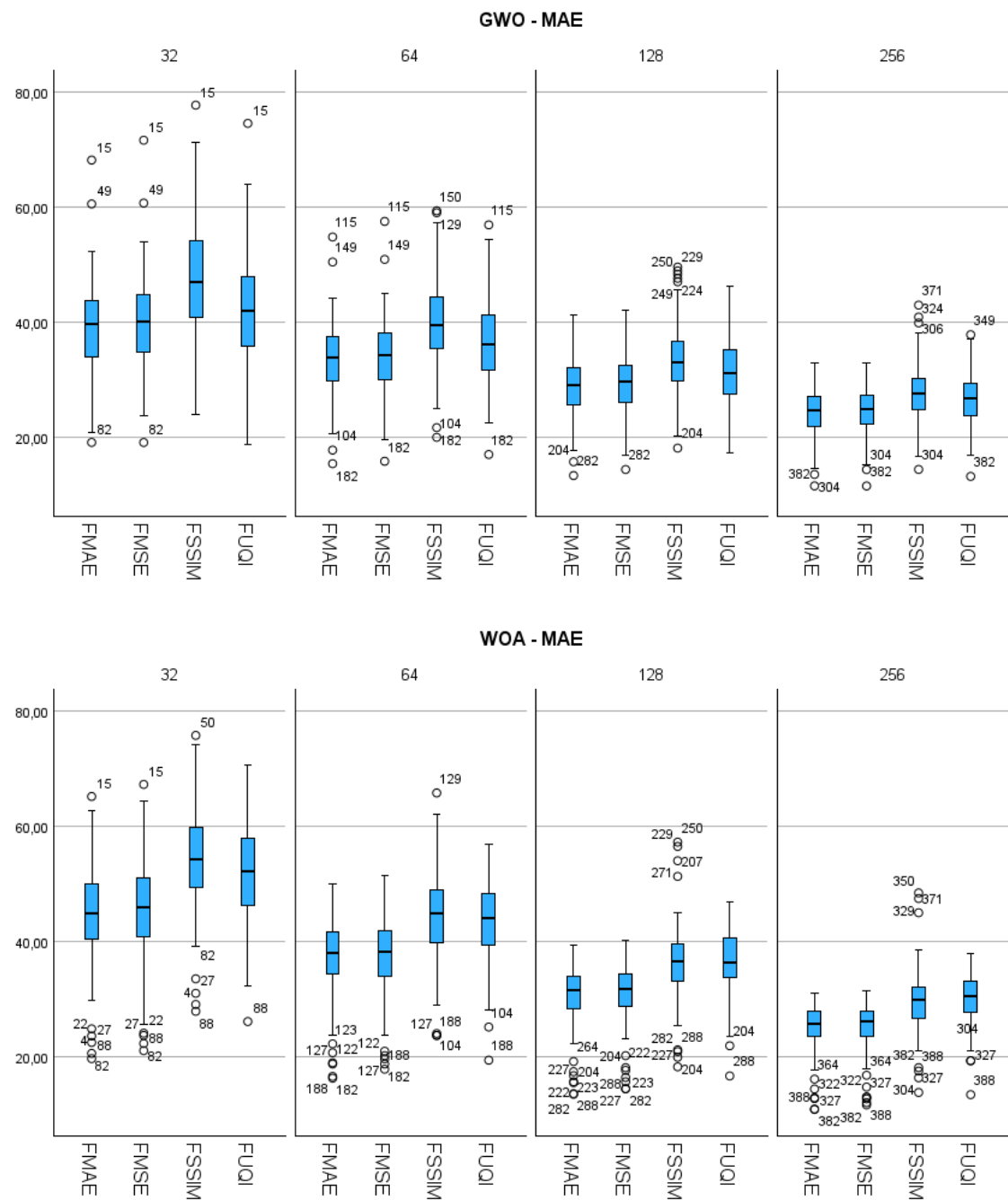
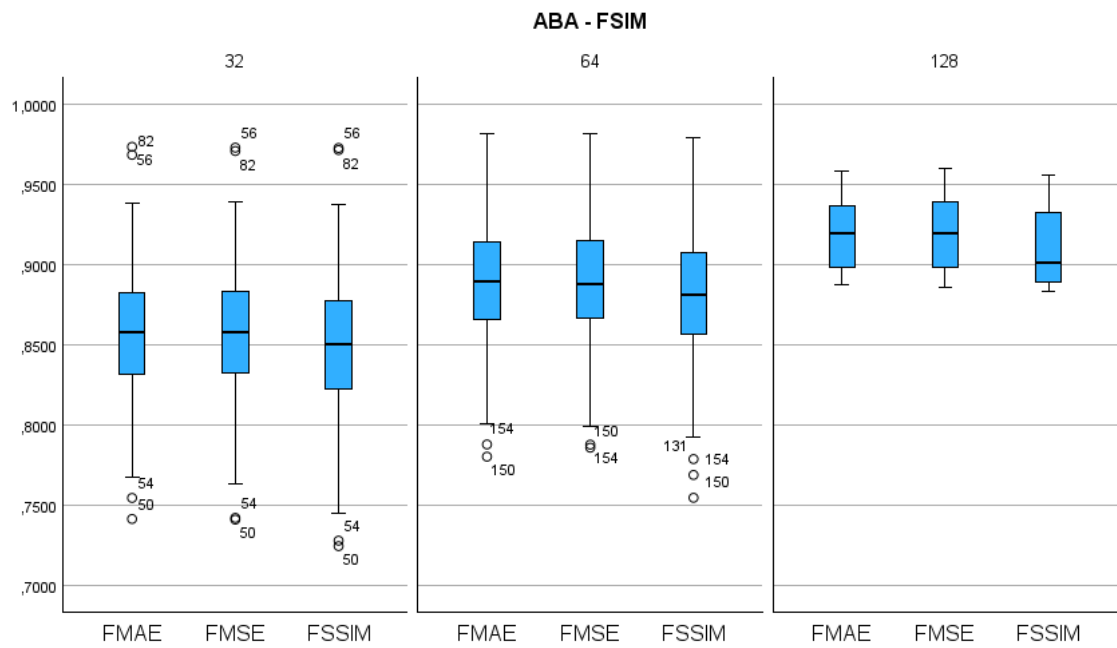
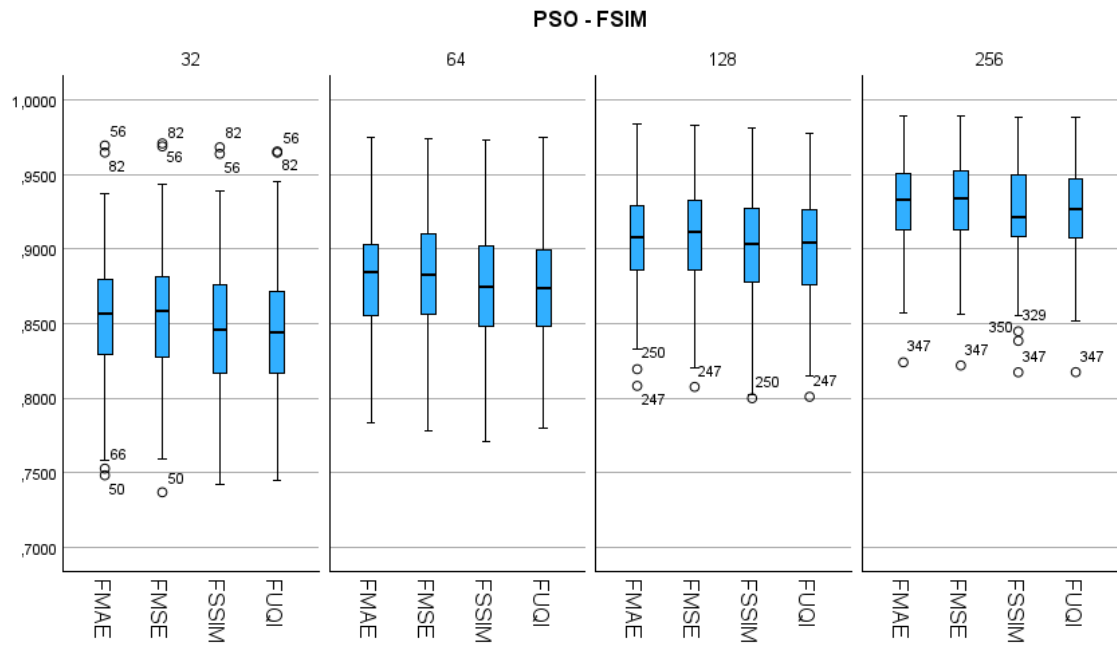
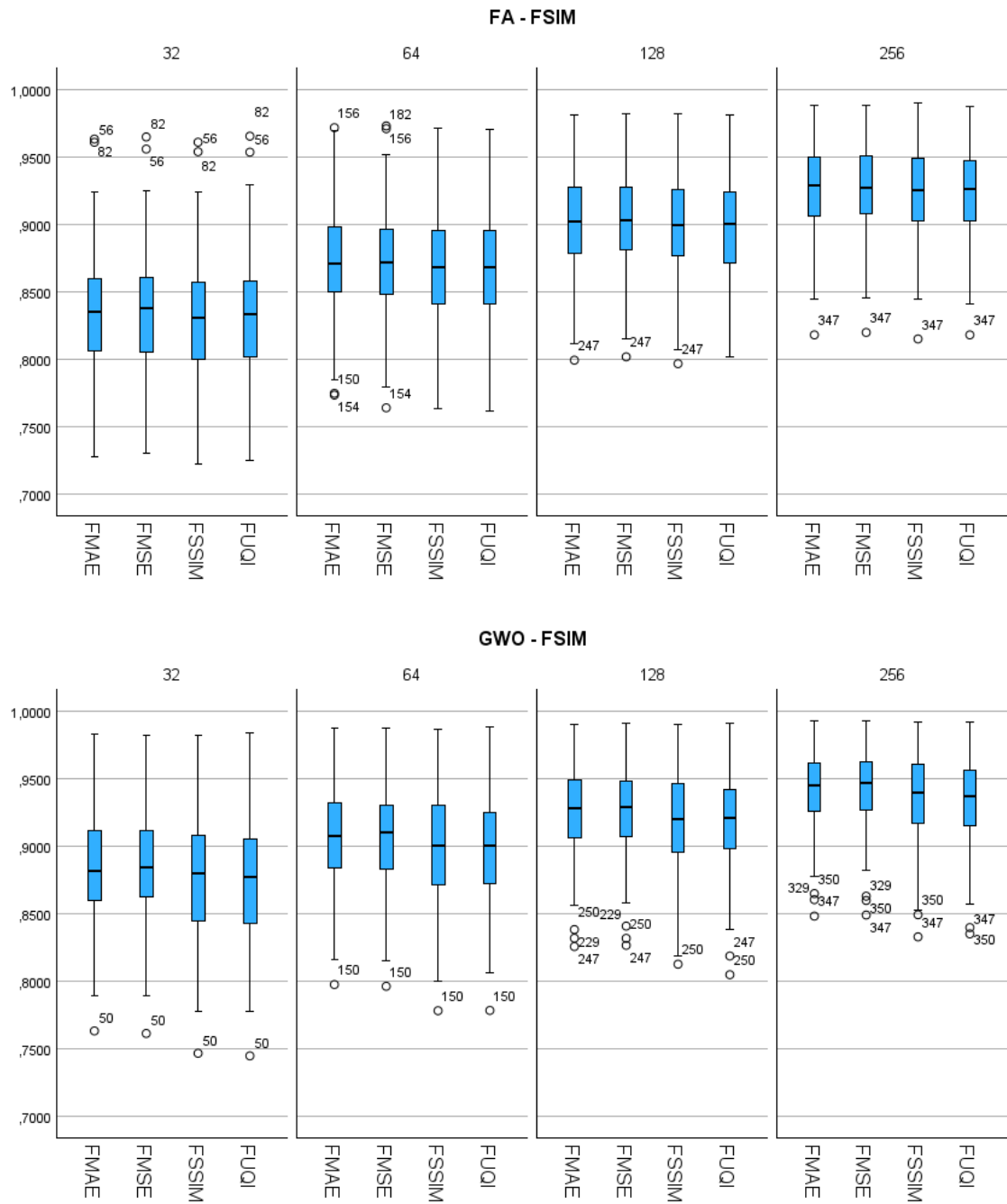


Ilustración 11 Índices MAE para todos los algoritmos probados







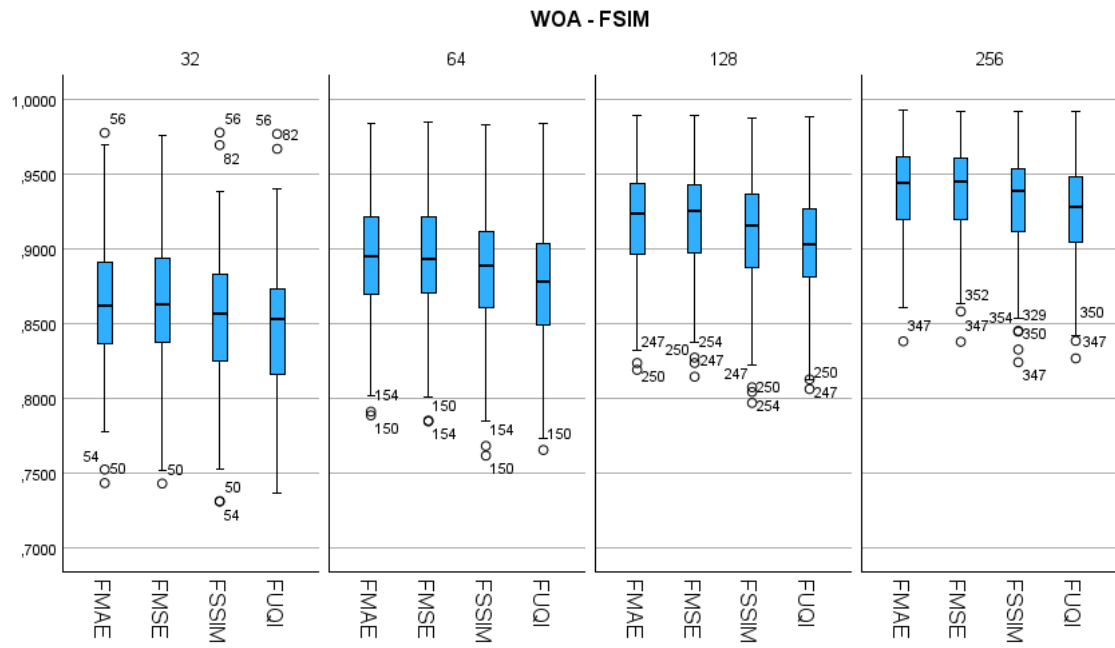
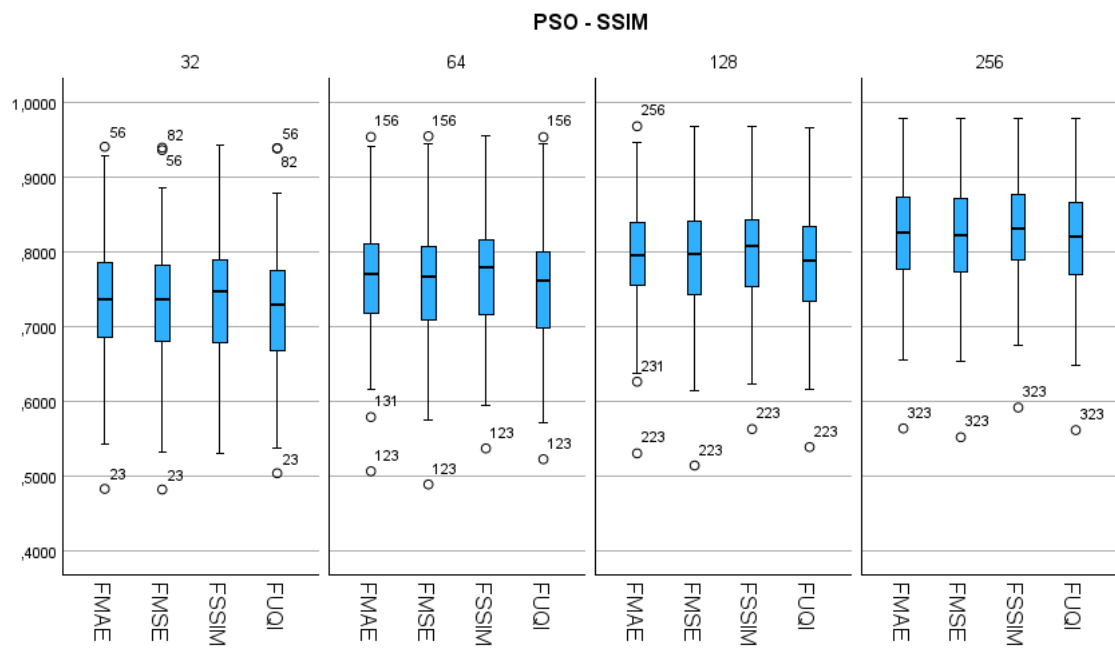
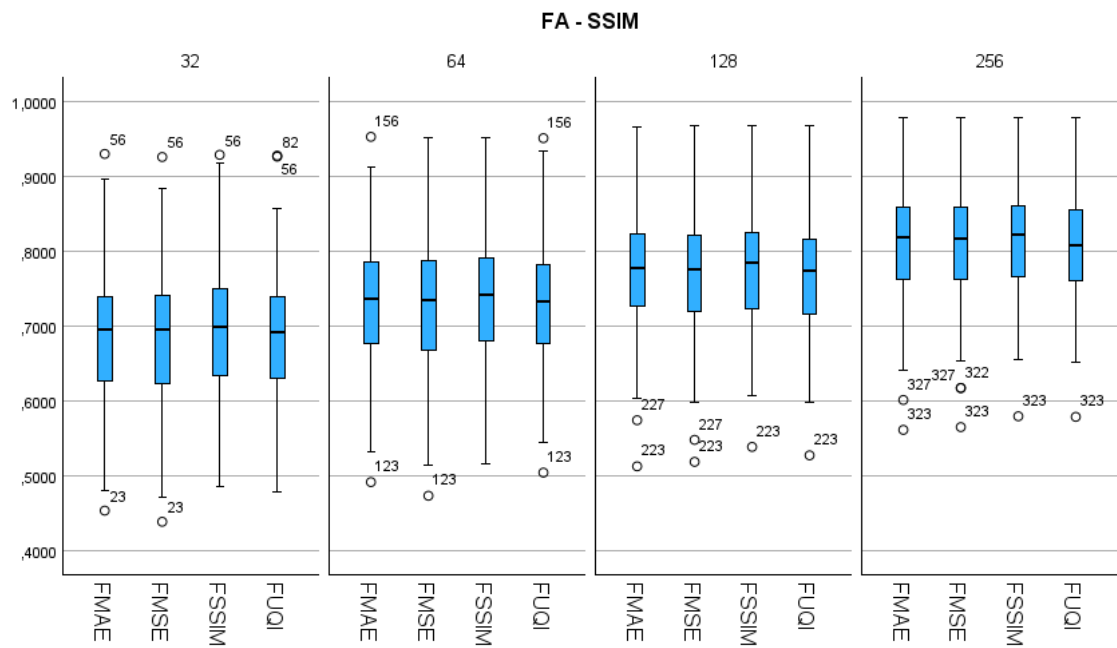
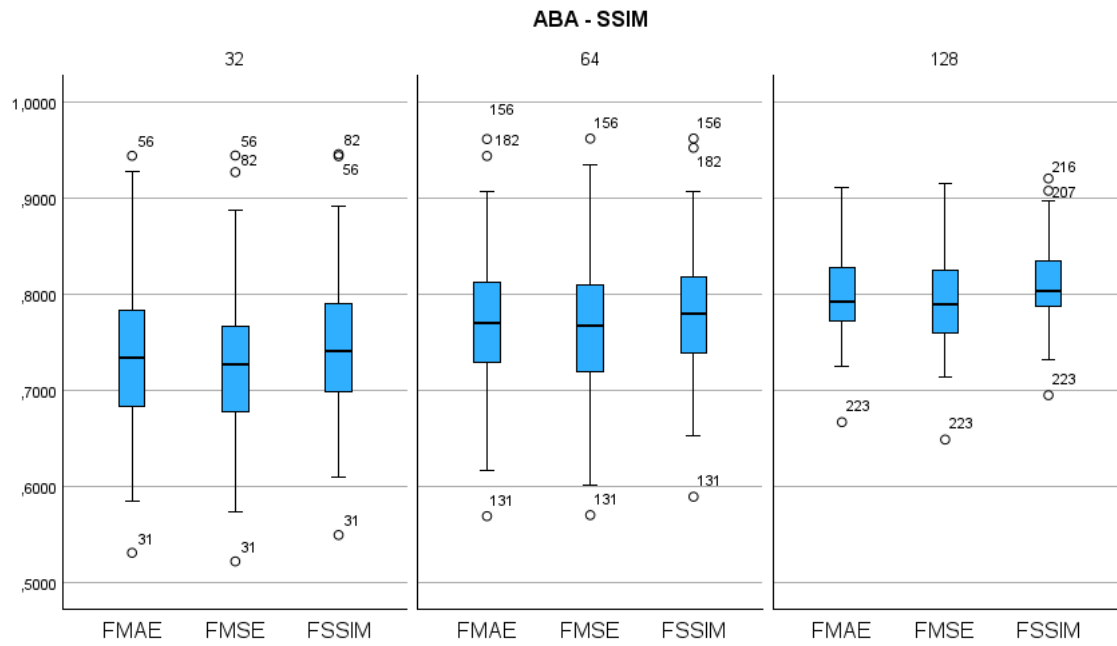
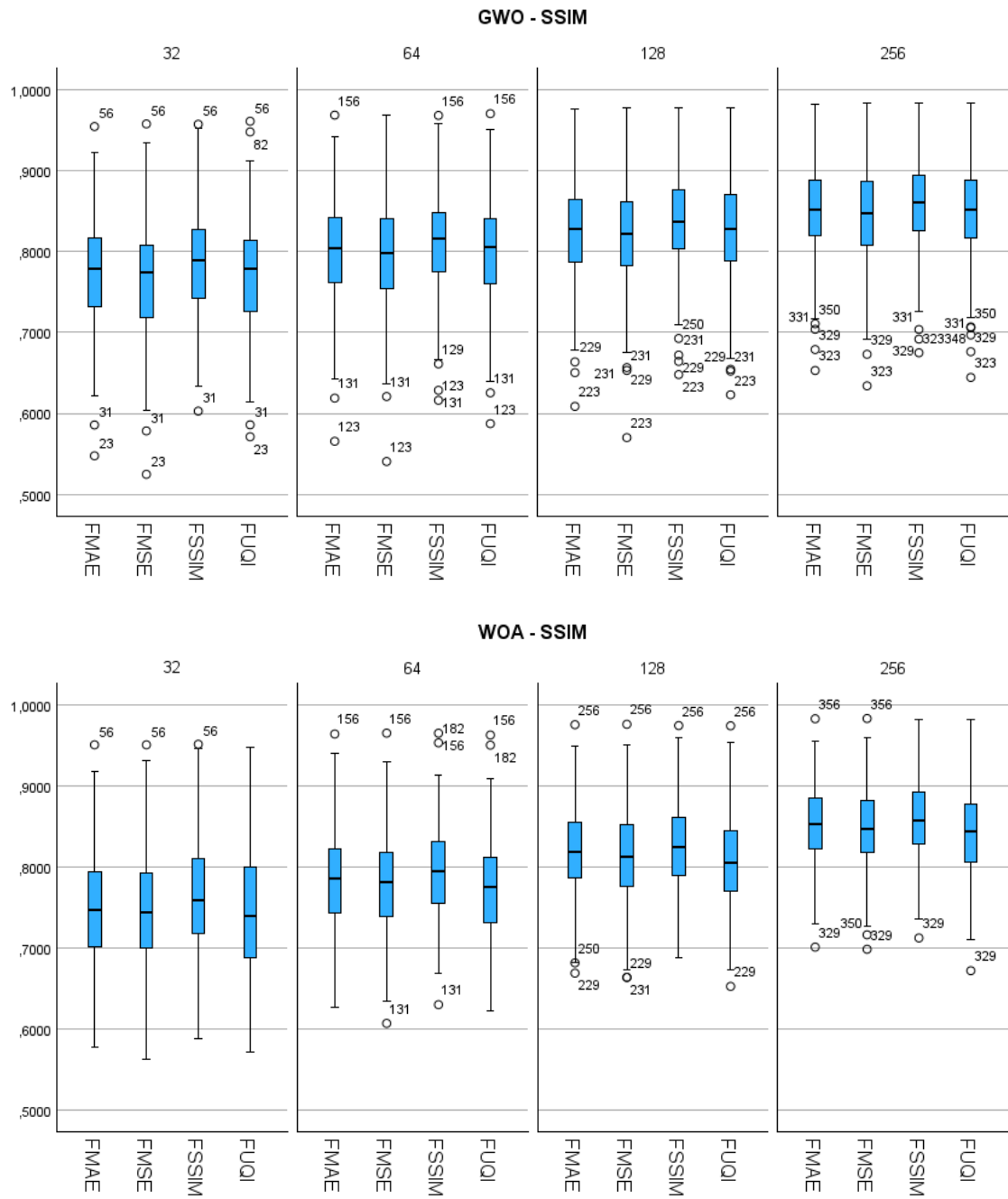


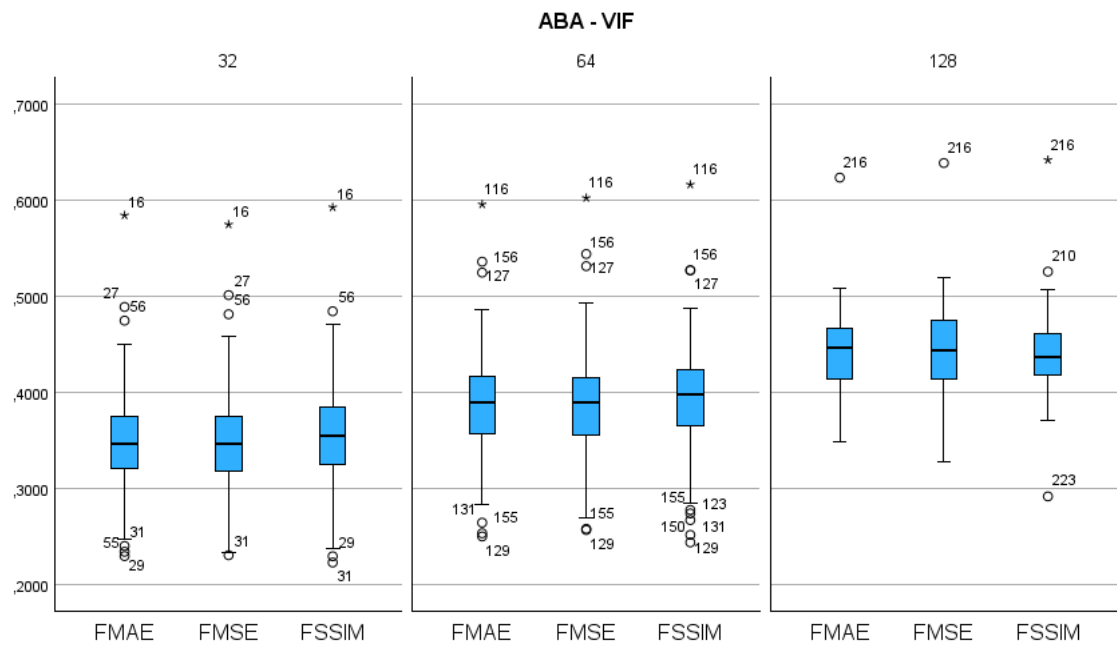
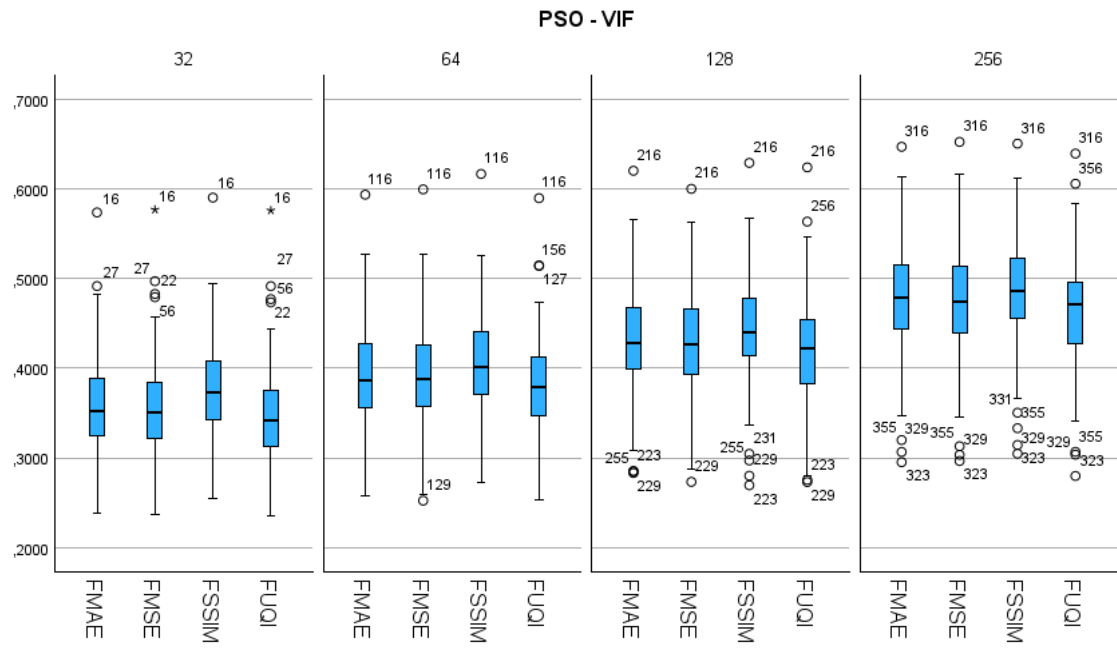
Ilustración 12 Índices FSIM para todos los algoritmos probados



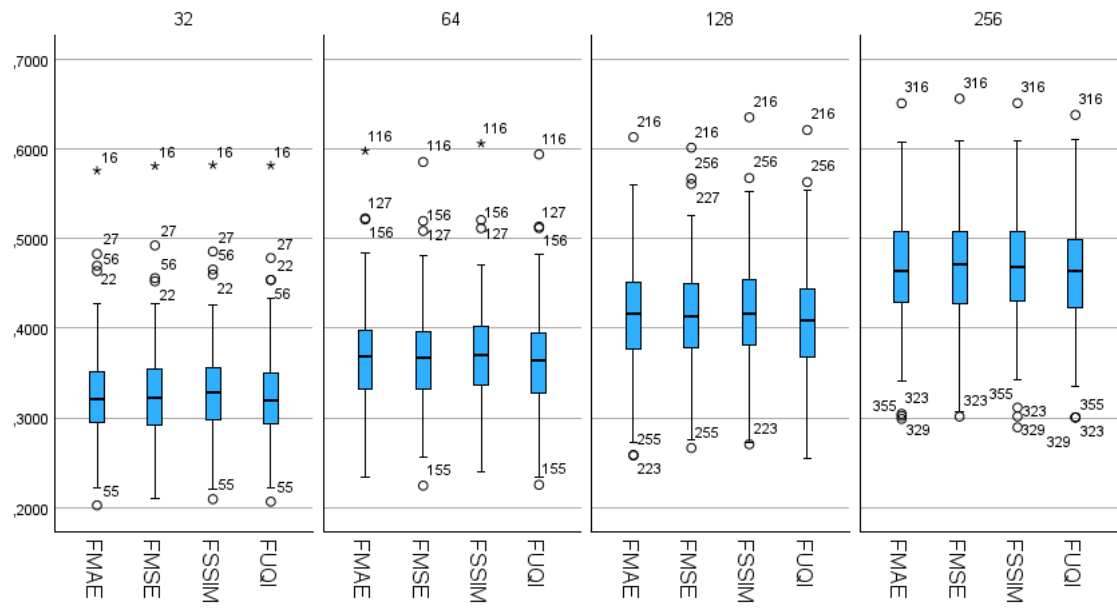




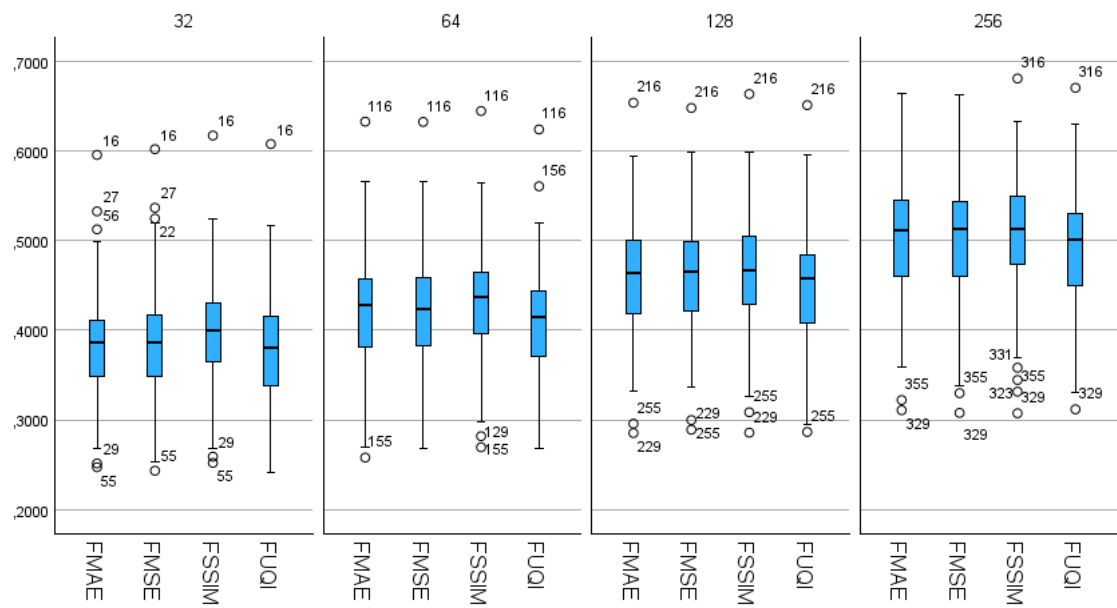
*Ilustración 13 Índices SSIM para todos los algoritmos probados*



# FA - VIF



# GWO - VIF



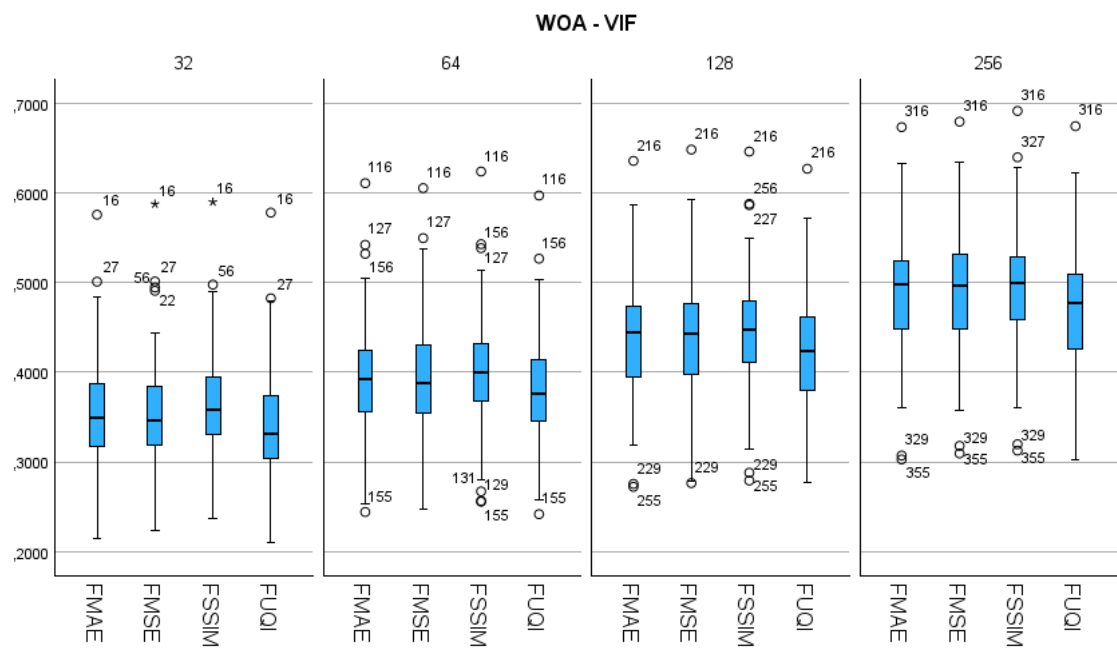
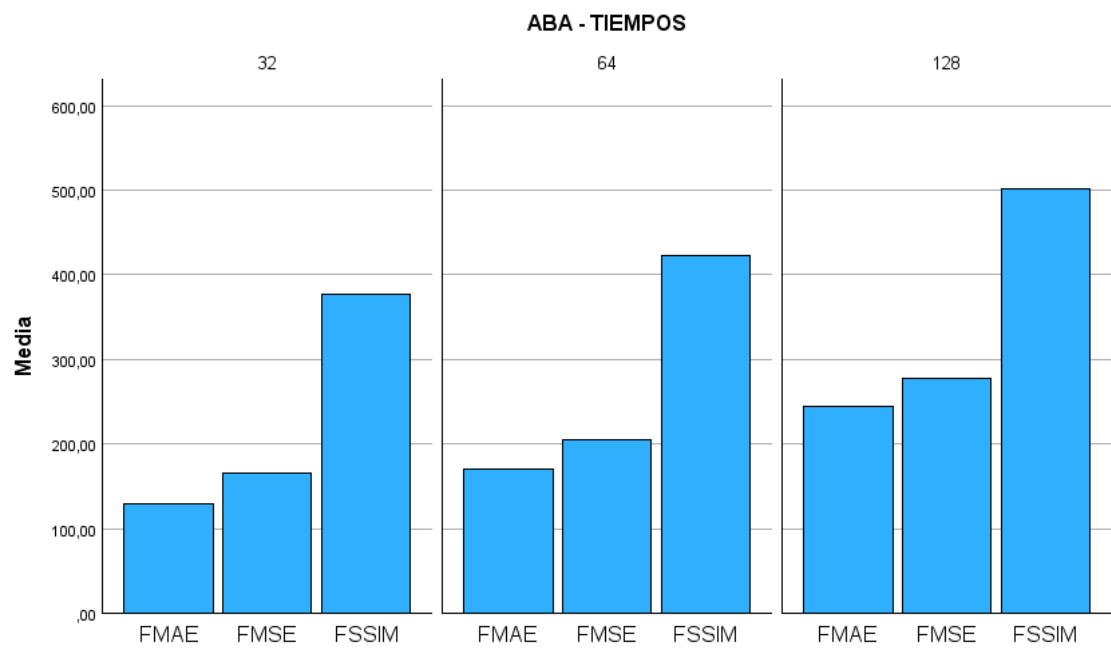
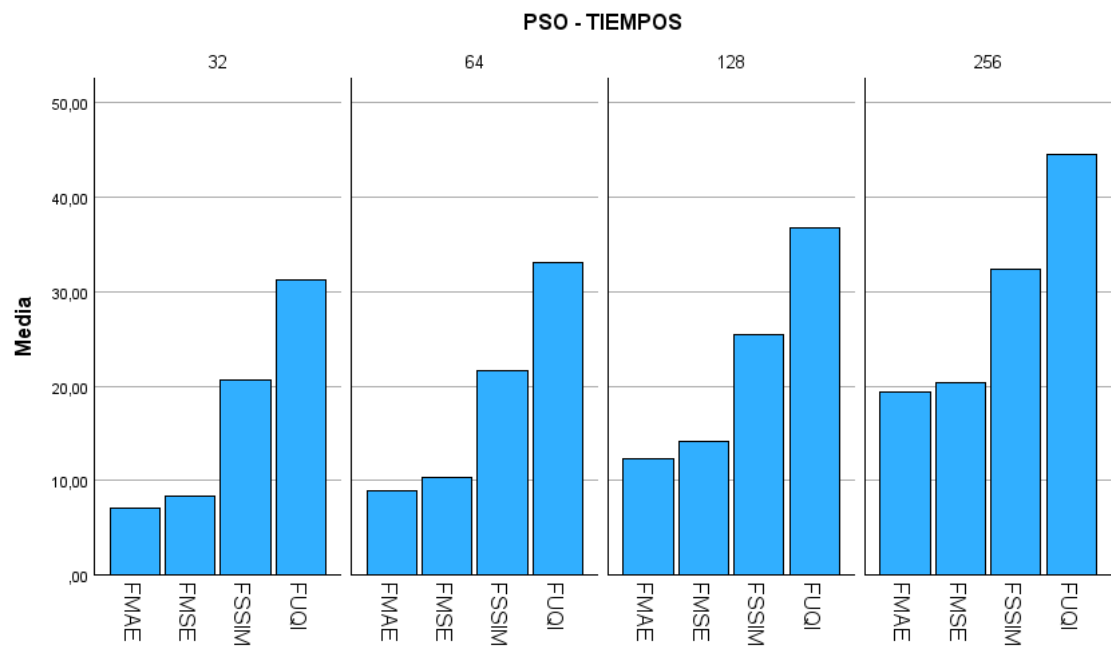
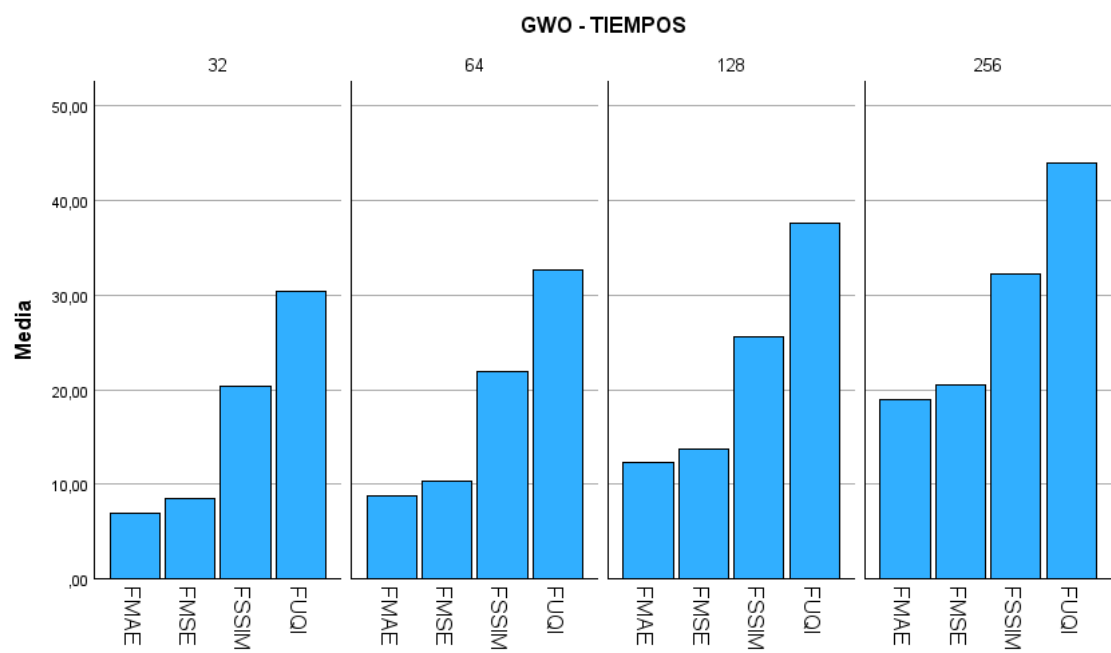
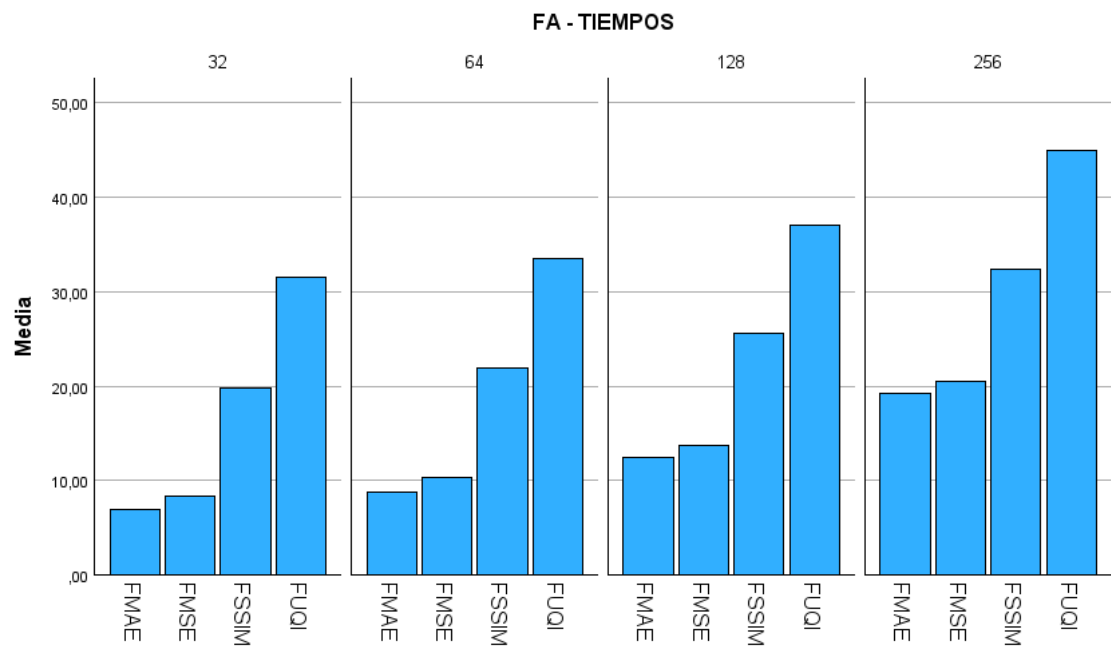
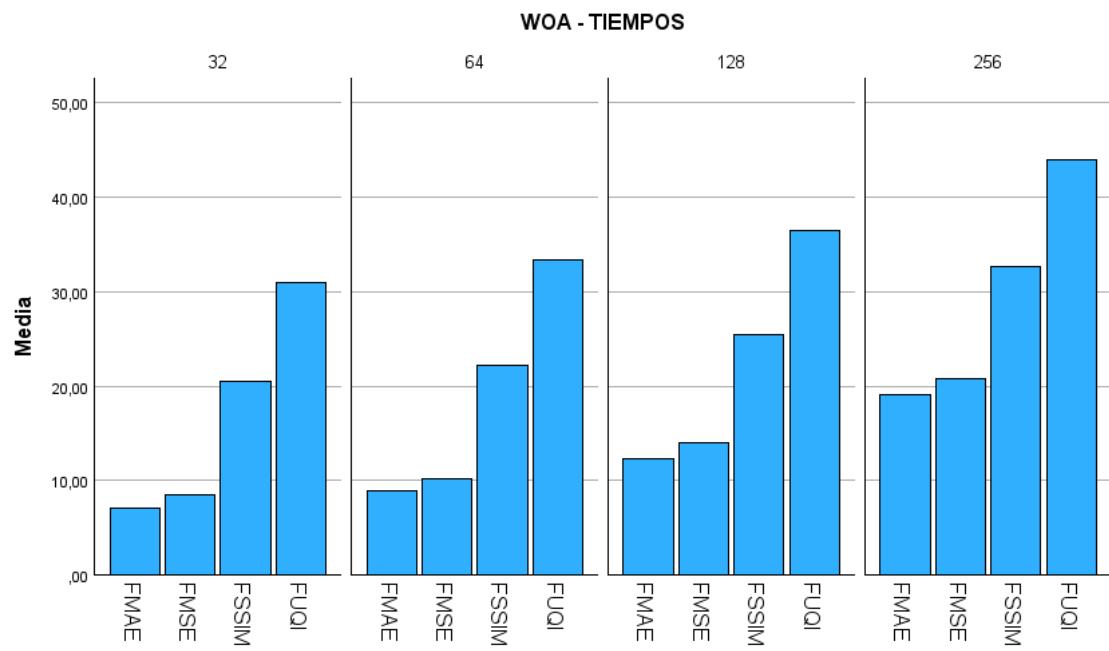


Ilustración 14 Índices VIF para todos los algoritmos probados









*Ilustración 15 Diagramas de barras de los tiempos de ejecución para todos los algoritmos probados*

## 7. Desarrollo del programa

Para el desarrollo como se ha mencionado antes se realizó un programa en Python que realiza las operaciones de los algoritmos de enjambre y genera imágenes cuantizadas usando cuatro funciones de fitness diferentes (MAE, MSE, SSIM, UQI), este programa se llama con diferentes argumentos que se explicarán en este apartado.

Por otro lado, también se realizó un script para la automatización de la obtención de resultados que se encarga de llamar al programa repetidas veces para ejecutarlo con todos los algoritmos, todas las funciones de fitness y 4 tamaños de paletas diferentes (32,64,128,256).

### 7.1 Archivo main o ejecutor

Este archivo es el principal del programa encargado de recibir los argumentos que se le pasen y de elegir o “mapear” estos argumentos con el algoritmo de enjambre que toque junto con la función de fitness que toque.

Para recibir estos argumentos se ha utilizado la librería por defecto de Python llamada `argparse`.

```
parser= argparse.ArgumentParser()
#Agrego argumento de la imagen
parser.add_argument('imagen', type=str, help="Nombre de la imagen a procesar. Debe estar dentro de la carpeta images en el proyecto.")
parser.add_argument('numeroColores', type=int, help="Numero de colores de la imagen cuantizada. No puede ser menor que 1. Ej. 32")
parser.add_argument('algoritmo', type=str, help="Algoritmo a procesar. Opciones: PSO,FA,GWO,ABA,WOA")
parser.add_argument('funcion', type=str, help="Funcion a usar por algoritmo. Opciones: MSE,MAE,SSIM,MSSIM")
parser.add_argument('iteraciones', type=int, help="Numero de iteraciones que realizara el algoritmo. Debe ser mayor que 0.")
parser.add_argument('individuos', type=int, help="Numero de individuos del algoritmo. Ej. 5 o 10.")
parser.add_argument('--pintaImagen', type=bool, default=False,help="Argumento para saber si se dibuja la imagen cuantizada al final del algoritmo (depuracion); Si viene se pinta.")
parser.add_argument('-a', type=int,default=0, help="Numero para saber si se ajusta la paleta generada nuevamente. Si es 1 se ajusta.")
args = parser.parse_args()
```

*Ilustración 16 Argumentos*

En la ilustración 10 se ve cómo se añaden los argumentos de tipo cadena o de tipo entero, estos argumentos por orden son:

- Imagen: imagen que se va a procesar en el algoritmo.
- NumeroColores: el número de colores de la nueva imagen o el tamaño de la paleta que se va a usar.
- Algoritmo: algoritmo que usará el programa para la cuantificación de color. Estos pueden ser: PSO, FA, BA, GWO, ABA que respectivamente representan a los algoritmos: pso, luciérnagas, ballenas, lobos y abejas.
- Función: función que calculará el fitness de cada iteración del algoritmo, las opciones son: MSE, MAE, SSIM, MSSIM.
- Iteraciones: número entero que representa el número de iteraciones que realizara el algoritmo escogido.
- Individuos: número entero que representa el tamaño de la población de individuos que utiliza el algoritmo.
- PintaImagen: este argumento booleano indica si se muestra por pantalla la imagen cuantizada o no.
- a: este último argumento indica si se ajusta la paleta generada. Si recibe el valor de 1 se ajusta. Este ajuste vuelve a calcular una paleta nueva para que se asemeje mas a la imagen original.

```
# Construir la ruta completa del directorio 'imagenes' y 'imagenesCuantizadas'
rutaDirectorioImagenes = os.path.join(os.path.dirname(__file__), 'imagenes')

rutaDirectorioCuantizadas=os.path.join(os.path.dirname(__file__), 'imagenesCuantizadas')

# Obtener la ruta completa de la imagen
ruta_imagen = os.path.join(os.path.dirname(__file__), 'imagenes', args.imagen)
```

*Ilustración 17 Obtener ruta de los directorios de imágenes*

En la ilustración 11 se ve la línea que obtiene la ruta de la imagen que se le ha pasado, de igual manera se realiza para el directorio de las imágenes cuantizadas. Se ha usado el módulo ‘os’ para obtenerla de manera que funcione bien en cualquier sistema operativo. Esta operación se realiza debido a que las imágenes que se procesan en el algoritmo están en una carpeta ‘imágenes’ que está ubicada al mismo nivel que el archivo ‘ejecutor’, si estuviesen al mismo nivel que el archivo ‘ejecutor’ no haría falta, pero esto sería más caótico.

```
# Verificar si el directorio imagenes existe
if not os.path.exists(rutaDirectorioImagenes):
    print(f"Error: No se puede encontrar el directorio 'imagenes'. Creandolo ...")
    os.makedirs(rutaDirectorioImagenes) # Crear el directorio 'imagenes'
    print(f"Directorio creado. Añada imagenes.")
    quit()

# Verificar si el directorio imagenesCuantizadas existe
if not os.path.exists(rutaDirectorioCuantizadas):
    print(f"Error: No se puede encontrar el directorio 'imagenesCuantizadas'. Creandolo ...")
    os.makedirs(rutaDirectorioCuantizadas) # Crear el directorio 'imagenes'

# Verificar si el archivo existe
if not os.path.isfile(ruta_imagen):
    print(f"Error: No se puede encontrar el archivo '{ruta_imagen}'")
    quit()

# Intentar leer la imagen con OpenCV
img = cv2.imread(ruta_imagen, cv2.IMREAD_COLOR)
if img is None:
    print(f"Error: No se puede abrir o leer el archivo '{ruta_imagen}'")
    quit()
```

*Ilustración 18 Comprobaciones de rutas e imágenes.*

En la ilustración 12 se ven algunas comprobaciones necesarias por si no se pudiese encontrar el archivo en la carpeta imagenes y la comprobación de que se puede leer o abrir la imagen pasada.

```

#Verificar si numero de iteraciones es válido
if (args.iteraciones <= 0):
    print(f"Error: Numero de iteraciones no válido. Debe ser un número mayor o igual a 1.")
    quit()

#Verificar si numero de colores es válido
if (args.numeroColores <= 1):
    print(f"Error: Numero de colores de la imagen cuantizada no válido. Debe ser un número mayor que 1.")
    quit()

#Verificar si algoritmo es válido
if args.algoritmo not in algoritmos:
    print(f"Error: Algoritmo " + args.algoritmo + " no válido. Debe ser uno de los algoritmos permitidos.")
    print(f"Algoritmos permitidos: ")
    for alg in algoritmos.keys():
        print(alg, end=" ")
    quit()

#Verificar si la funcino introducida es válida
if args.funcion not in funcionesObjetivo:
    print(f"Error: Funcion " + args.funcion + " no válida. Debe ser una de los funciones permitidas.")
    print(f"Funciones: ")
    for fn in funcionesObjetivo.keys():
        print(fn, end=" ")
    quit()

#Verificar si numero de colores es válido
if (args.individuos < 1):
    print(f"Error: Numero de individuos no válido. Debe ser un número mayor que 0.")
    quit()

```

*Ilustración 19 Comprobaciones extra*

En la ilustración 13 se muestran las comprobaciones del resto de argumentos. Comprueba si se ha recibido un número válido de iteraciones, colores e individuos.

También comprueba si el algoritmo y la función de fitness recibidos están dentro de los valores permitidos.

```

#DICCIONARIOS FUNCIONES ALGORITMOS
#En este punto se pueden añadir nuevas funciones y algoritmos. En los algoritmos solo hace falta definir los argumentos que se le pasan
# Definir funciones disponibles para el algoritmo
funcionesObjetivo = {}
"RMSE": func.getRmse,
"MAE": func.getMae,
"SSIM": func.getSsim,
"UQI": func.getUqi,
"FSIM": func.getFsim,
"VIF": func.getVifp,
"MSSIM": func.getMssim

# Definir los algoritmos disponibles
algoritmos = {}
"PSO": lambda indiv, func, col, img, it, ajuste: SwarmPackagePy.pso(indiv, func, 0, 255, 3, it, col, args.pintaImagen, -14, 14, w=0.729, f1=2.05, f2=2.05, imagen=img, ajuste=ajuste),
"FA": lambda indiv, func, col, img, it, ajuste: SwarmPackagePy.fa(indiv, func, 0, 255, 3, it, col, args.pintaImagen, beta0=0.1, gamma=0.03, norm0=0, norm1=0.1, imagen=img, ajuste=ajuste),
"GWO": lambda indiv, func, col, img, it, ajuste: SwarmPackagePy.gwo(indiv, func, 0, 255, 3, it, col, args.pintaImagen, imagen=img, ajuste=ajuste),
"ABA": lambda indiv, func, col, img, it, ajuste: SwarmPackagePy.abejas(indiv, func, 0, 255, 3, it, col, args.pintaImagen, imagen=img, ajuste=ajuste),
"WOA": lambda indiv, func, col, img, it, ajuste: SwarmPackagePy.woa(indiv, func, 0, 255, 3, it, col, args.pintaImagen, imagen=img, ajuste=ajuste)
}

```

*Ilustración 20 Diccionarios Python de algoritmos y funciones*

En la ilustración 14 se muestran los diccionarios Python que hacen referencia a las funciones y algoritmos implementados. Estos diccionarios nos permiten acceder a las diferentes configuraciones de la herramienta, también nos simplifica el desarrollo de nuevas funciones y algoritmos en un futuro, ya que lo único que se debe hacer es añadirlo a estos diccionarios de la misma manera que están los algoritmos implementados.

En el diccionario de las funciones objetivo se debe poner el valor que debe recibir la herramienta como una cadena y el valor real que tiene, es decir, el método que hace referencia a la función objetivo desarrollado en el módulo de funciones.

En el diccionario de algoritmos se debe añadir el valor que recibe la herramienta como una cadena y la función lambda a la que hace referencia, es decir, la referencia al constructor del algoritmo que se esté implementando.

```

#Para instalar librerias -> pip install -r requirements.txt
#ejemplo de ejecucion: python3 ejecutor.py maya_beach.ppm 64 PSO FSIM 15 15 --pintaImagen True -a 0

# Función que ejecuta el algoritmo correspondiente
def ejecutar_algoritmo(algoritmo, funcion, individuos, iteraciones, numero_colores, imagen, ajuste):
    if algoritmo not in algoritmos:
        print(f"Error: Algoritmo '{algoritmo}' no reconocido")
        quit()

    if funcion not in funcionesObjetivo:
        print(f"Error: Funcion '{funcion}' no reconocida")
        quit()

    # Ejecutar el algoritmo
    alg_func = algoritmos[algoritmo]
    func_fitness = funcionesObjetivo[funcion]
    alg_func(individuos, func_fitness, numero_colores, imagen, iteraciones, ajuste)

```

*Ilustración 21 Método ejecutor de la herramienta*

En la ilustración 15 se muestra el método que llama al algoritmo indicado con la función objetivo que se le ha pasado. Realiza comprobaciones y escoge el valor pedido recuperándolo de los diccionarios que se mencionaron anteriormente.

Todas estas operaciones constituyen la primera parte del programa, donde se reciben argumentos, se comprueban y se “escoge” el algoritmo a usar junto a la función de fitness seleccionada.

## 7.2 Script

Una vez definido como se ejecuta la herramienta se explica el script realizado para obtener todas las pruebas.

Como resumen, este script llama repetidas veces a la herramienta ejecutando una serie de bucles que componen todas y cada una de las posibles configuraciones que puede realizar la herramienta para componer todos los resultados.

```

iteraciones=15
individuos=20

# Directorios y archivos
dir_trabajo=$(pwd)
dir_checkpoints="$dir_trabajo/checkpoints"
dir_imagenesCuantizadas="$dir_trabajo/imagenesCuantizadas"
dir_piq="$dir_trabajo/piq-master"

# Crear directorio de checkpoints si no existe
mkdir -p "$dir_checkpoints"

```

*Ilustración 22 Constantes y directorios*

En la anterior ilustración se muestran las constantes del script, todos los algoritmos se ejecutaron con 15 iteraciones y 20 individuos, También se pueden ver los directorios, la propia ruta donde está el script, el directorio de checkpoints, y los directorios auxiliares tanto de las imágenes cuantizadas como el directorio donde se ubica el piq-master que es la herramienta usada para calcular los diferentes índices de error con los que se han generado los resultados [21].

```
for ALGO in "PSO" "WOA" "GWO" "FA" "ABA"
do

# Bucle para recorrer todas las funciones que acepta el programa
for FUNC in "MSE" "MAE" "SSIM" "UQI"
do
|

# Separo los resultados de cada tamaño de paleta en un fichero diferente
for C in 32 64 128 256; # para cada tamaño de paleta cuantizada
do

#Bucle para decidir el ajuste o no ajuste de la paleta
for AJUSTE in 0;
do
```

*Ilustración 23 Bucles*

En la ilustración 17 se muestran los bucles que componen las diferentes configuraciones de la herramienta en cuanto al algoritmo, función de fitness, paleta y ajuste requeridos.

```
| # Archivo de checkpoint para esta configuración
config_checkpoint="${dir_checkpoints}/config_${ALGO}_${FUNC}_${C}_${AJUSTE}.done"
image_checkpoint="${dir_checkpoints}/image_${ALGO}_${FUNC}_${C}_${AJUSTE}.txt"

# Leer el último checkpoint de imagen si existe
last_image_processed=""
if [[ -f "$image_checkpoint" ]]; then
|   last_image_processed=$(cat "$image_checkpoint")
fi

# Verificar si ya se ha completado esta configuración
if [[ -f "$config_checkpoint" ]]; then
|   echo "Configuración completada: $ALGO $FUNC $C $AJUSTE"
|   continue
fi

found_last_image=false
[[ -z "$last_image_processed" ]] && found_last_image=true
```

*Ilustración 24 Sistema de checkpoints*

Debido al tiempo necesario para ejecutar todas las pruebas de este trabajo, tuve que implementar un pequeño sistema de checkpoints que me permitieran guardar los puntos que ya tenía probados. Este sistema se divide en comprobaciones tanto para conocer si una configuración ya está ejecutada, es decir, si se ha ejecutado el PSO con la función



MAE con una paleta de 32 colores, por ejemplo, y la comprobación de la imagen que se ha probado.

```
# Para cada imagen del conjunto de 100 CQ-100
for F in adirondack_chairs.ppm astro_bodies.ppm astronaut.ppm bali
do
if [[ "$found_last_image" == true ]]; then
```

*Ilustración 25 Bucle Imagenes*

En la ilustración 19 se muestra el bucle de las imágenes del conjunto CQ-100 y la comprobación de si ya se ha procesado.

```
# mensaje para el terminal (método que voy a probar)
echo "algoritmo->$ALGO Cuantizada->$C Funcion-> $FUNC Imagen -> $F"

# NÚMERO DE TEST INDEPENDIENTES EJECUTADOS PARA UNA MISMA CONFIGURACIÓN (5, DE MOMENTO) POR A
for TEST_INDEPEN in {1..20}
do
```

*Ilustración 26 Comienzo de la prueba*

En esta ilustración se muestra el mensaje que se escribe por la terminal para conocer el estado en el que se encuentra el script, y un bucle que hace se ejecuten 20 test independientes para cada imagen, para generar el banco de datos.

```
#Antes de ejecutar el programa capturo el tiempo del inicio de la ejecucion para luego saber lo que tardo en ejecutarse el algoritmo
tiempoIni=$(date +%s)

# Ejecuto el código python que aplica un algoritmo de enjambre.
# Los resultados de la ejecución del programa se vuelcan en un fichero con un
# nombre de la forma:
# salida_PSO_32.txt salida_PSO_64.txt salida_PSO_128.txt o salida_PSO_256.txt , en este ejemplo
python3 ejecutor.py ${F} ${C} ${ALGO} ${FUNC} ${iteraciones} ${individuos} -a ${AJUSTE} >> salida_${ALGO}_${FUNC}_${C}.txt

# Capturar el tiempo de fin
tiempoFin=$(date +%s)
# Calcular la duración en milisegundos
duration=$((tiempoFin - tiempoIni)) # Dividir por 1,000,000 para convertir nanosegundos a milisegundos

#Escribo en el txt correspondiente la duracion de la ejecucion en milisegundos
echo $duration >> salida_${ALGO}_${FUNC}_${C}.txt
#y un salto de linea para la siguiente ejecucion
echo >> salida_${ALGO}_${FUNC}_${C}.txt

# Calculo múltiples medidas de error sobre la imagen cuantizada que acabo de generar
python3 errores_cq.py ${F} ${ALGO}_${C}_${F} >> IQI_${ALGO}_${FUNC}_${C}.txt

cd piq-master
python3 image_metrics.py ${F} ${ALGO}_${C}_${F} >> ../IQI_${ALGO}_${FUNC}_${C}.txt
cd ..
#A los índices de error generados en una línea le añado el tiempo de ejecucion de un algoritmo enterno
echo $duration >> IQI_${ALGO}_${FUNC}_${C}.txt
```

*Ilustración 27 Ejecución de la herramienta*

En la ilustración 21 se muestra la ejecución de la herramienta llamando a ‘ejecutor.py’, y el cálculo de los índices de error llamando a ‘errores\_cq.py’ donde se calculan algunos de ellos y a ‘image\_metrics.py’ que es del paquete piq-master donde se calcula el resto de ellos. También se captura el momento en el que empezó a ejecutarse para posteriormente poder conocer el tiempo que se tardó en ejecutar, el cual se escribe al final de cada línea en los txt de resultados.

```
# Borro la imagen cuantizada, que ya no necesito (guardarlas todas me ocupa
# mucho disco duro)
cd imagenesCuantizadas
rm ${ALGO}_${C}_${F}
cd ..

done #tests sucesivos
echo "$F" > "$image_checkpoint" # Actualiza el archivo de checkpoint con la última imagen procesada
```

*Ilustración 28 Fin del script*

En esta ilustración se muestra el borrado de la imagen cuantizada después de calcular los índices de error y la actualización del archivo de checkpoints.

La imagen cuantizada se borra debido a que almacenar todas ocuparía mucho espacio en disco.

## 8. Trabajo futuro

Existen varias áreas en las que este trabajo podría expandirse y mejorar, algunos de ellos son:

- **Optimización de Algoritmos:** Aunque los algoritmos de enjambre como PSO, GWO han sido aplicados exitosamente, se podrían explorar versiones avanzadas de estos algoritmos, como versiones híbridas o adaptativas que ajusten sus parámetros dinámicamente durante la ejecución. Esto podría mejorar la convergencia y reducir el tiempo de procesamiento.
- **Ampliación de la Biblioteca de Algoritmos:** Actualmente, la herramienta incluye un conjunto de algoritmos específicos. Una dirección interesante sería la incorporación de nuevos algoritmos de optimización basados en la naturaleza, como el algoritmo de murciélagos, el algoritmo de cucos o variantes de redes neuronales evolutivas, ampliando las opciones para la cuantificación de color.
- **Paralelización y Uso de GPU:** A medida que las imágenes procesadas son de mayor tamaño o mayor resolución, los tiempos de cómputo aumentan considerablemente. La paralelización del código o el uso de procesamiento en GPU podría reducir de manera significativa el tiempo de ejecución de los algoritmos, haciéndolos más eficientes en escenarios de grandes volúmenes de datos.
- **Evaluación de Nuevas Métricas:** Las métricas de calidad utilizadas, como el MSE, MAE y SSIM ofrecen una visión del rendimiento, pero podrían incluirse nuevas métricas que tomen en cuenta aspectos específicos de la percepción humana del color. Métricas más sofisticadas que evalúen la fidelidad del color y las características visuales serían un buen complemento.

## 9. Conclusiones

El desarrollo de esta herramienta para la aplicación de algoritmos de enjambre en la cuantificación de color ha demostrado la efectividad y versatilidad de estas técnicas bioinspiradas en el ámbito del procesamiento de imágenes. A lo largo del proyecto, se han implementado y probado diversos algoritmos de optimización como PSO (Optimización por Enjambre de Partículas), GWO (Optimización por Lobos Grises), FA (Luciérnagas), WOA (Ballenas) y ABA (Abejas Artificiales).

La principal ventaja de los algoritmos de enjambre reside en su capacidad para explorar eficientemente el espacio de búsqueda y encontrar soluciones factibles en problemas complejos, como la cuantificación de color. Además, estos algoritmos son altamente flexibles, permitiendo su ajuste y personalización para adaptarse a diferentes escenarios y requisitos de precisión.

Si se necesitase usar estos algoritmos en problemas de cuantificación de color, gracias a los resultados obtenidos en este trabajo, podemos afirmar que existen algoritmos que obtienen mejores resultados como el GWO y otros que, aunque no llegan a la misma precisión que este, también se deben de tener en cuenta como el WOA.

La función objetivo que optimizar depende del uso que se le quiera dar, ya que algoritmos ejecutados con la función objetivo MSE minimizan el MSE y el MAE con un tiempo de ejecución bastante bueno, mientras que, si se optimiza SSIM o UQI se maximiza la similitud perceptual a costa de incrementar el tiempo de ejecución, aunque sigue siendo asumible.

El algoritmo que no se recomienda es el ABA puesto que su calidad de resultados no justifica el tiempo de ejecución.

## 10. Bibliografía

- [1] «Los modos de color de la imagen digital». Accedido: 28 de enero de 2024. [En línea]. Disponible en: <https://www.fotonostra.com/fotografia/modoscolor.htm>
- [2] «Inteligencia de enjambre e inteligencia artificial - Fundación Aquae». Accedido: 5 de febrero de 2024. [En línea]. Disponible en: <https://www.fundacionaquae.org/la-inteligencia-enjambre-y-la-inteligencia-artificial/>
- [3] «PSO: Optimización por enjambres de partículas». Accedido: 21 de abril de 2024. [En línea]. Disponible en: <https://www.cs.us.es/~fsancho/Blog/posts/PSO.md>
- [4] «Algoritmo firefly - Wikipedia, la enciclopedia libre». Accedido: 5 de mayo de 2024. [En línea]. Disponible en: [https://es.wikipedia.org/wiki/Algoritmo\\_firefly](https://es.wikipedia.org/wiki/Algoritmo_firefly)
- [5] S. Mirjalili, S. M. Mirjalili, y A. Lewis, «Grey Wolf Optimizer», *Advances in Engineering Software*, vol. 69, pp. 46-61, mar. 2014, doi: 10.1016/J.ADVENGSOFT.2013.12.007.
- [6] «GWO». Accedido: 20 de mayo de 2024. [En línea]. Disponible en: <https://seyedalimirjalili.com/gwo>
- [7] H. Faris, I. Aljarah, M. A. Al-Betar, y S. Mirjalili, «Grey wolf optimizer: a review of recent variants and applications», *Neural Comput Appl*, vol. 30, n.º 2, pp. 413-435, jul. 2018, doi: 10.1007/S00521-017-3272-5/TABLES/4.
- [8] J. Nasiri y F. M. Khiyabani, «A whale optimization algorithm (WOA) approach for clustering», *Cogent Math Stat*, vol. 5, n.º 1, p. 1483565, ene. 2018, doi: 10.1080/25742558.2018.1483565.
- [9] «AN IDEA BASED ON HONEY BEE SWARM FOR NUMERICAL OPTIMIZATION».
- [10] «Algoritmo colonia de abejas artificiales - Wikipedia, la enciclopedia libre». Accedido: 4 de agosto de 2024. [En línea]. Disponible en: [https://es.wikipedia.org/wiki/Algoritmo\\_colonia\\_de\\_abejas\\_artificiales](https://es.wikipedia.org/wiki/Algoritmo_colonia_de_abejas_artificiales)
- [11] *Why was Python created in the first place?* General Python FAQ. Accedido: 4 de septiembre de 2024. [En línea]. Disponible en: <http://docs.python.org/faq/general#why-was-python-created-in-the-first-place>
- [12] «artima - The Making of Python», *www.artima.com*, Accedido: 4 de septiembre de 2024. [En línea]. Disponible en: <https://www.artima.com/articles/the-making-of-python>
- [13] «La librería Numpy | Aprende con Alf». Accedido: 4 de septiembre de 2024. [En línea]. Disponible en: <https://aprendeconalf.es/docencia/python/manual/numpy/>
- [14] «Procesamiento de imágenes con OpenCV en Python». Accedido: 4 de septiembre de 2024. [En línea]. Disponible en: <https://imaginaformacion.com/tutoriales/opencv-en-python>
- [15] «Scikit-Learn, herramienta básica para el Data Science en Python». Accedido: 4 de septiembre de 2024. [En línea]. Disponible en: <https://www.master-data-scientist.com/scikit-learn-data-science/>
- [16] S. Van Der Walt *et al.*, «scikit-image: image processing in Python», *PeerJ*, vol. 2:e453, n.º 1, p. e453, jul. 2014, doi: 10.7717/peerj.453.

- [17] A. León-Batallas, J. Bermeo-Paucar, J. Paredes-Quevedo, y H. Torres-Ordoñez, «Una revisión de las métricas aplicadas en el procesamiento de imágenes», *RECIMUNDO*, vol. 4, n.º 3, pp. 267-273, sep. 2020, doi: 10.26820/recimundo/4.(3).julio.2020.267-273.
- [18] L. Zhang, L. Zhang, X. Mou, y D. Zhang, «FSIM: A feature similarity index for image quality assessment», *IEEE Transactions on Image Processing*, vol. 20, n.º 8, pp. 2378-2386, ago. 2011, doi: 10.1109/TIP.2011.2109730.
- [19] H. R. Sheikh y A. C. Bovik, «A VISUAL INFORMATION FIDELITY APPROACH TO VIDEO QUALITY ASSESSMENT».
- [20] M. E. Celebi y M.-L. Pérez-Delgado, «CQ100: A High-Quality Image Dataset for Color Quantization Research», vol. 2, 2022, doi: 10.17632/VW5YS9HFXW.2.
- [21] «GitHub - photosynthesis-team/piq: Measures and metrics for image2image tasks. PyTorch.» Accedido: 15 de junio de 2025. [En línea]. Disponible en: <https://github.com/photosynthesis-team/piq>
- [22] «Así son los LOBOS GRISES ¡Impresionantes! - Impacto Venezuela». Accedido: 15 de junio de 2025. [En línea]. Disponible en: <https://impactove.com/asi-son-los-lobos-grises-impresionantes/>
- [23] «Python logo - Iconos Social Media y Logos». Accedido: 15 de junio de 2025. [En línea]. Disponible en: <https://icon-icons.com/es/icono/python-logo/168886>
- [24] «File:NumPy logo 2020.svg - Wikipedia». Accedido: 15 de junio de 2025. [En línea]. Disponible en: [https://en.m.wikipedia.org/wiki/File:NumPy\\_logo\\_2020.svg](https://en.m.wikipedia.org/wiki/File:NumPy_logo_2020.svg)
- [25] «1000 Logos - The Famous logos and Popular company logos meaning». Accedido: 15 de junio de 2025. [En línea]. Disponible en: <https://1000logos.net/>
- [26] «Visual Studio Code Logo PNG vector in SVG, PDF, AI, CDR format». Accedido: 15 de junio de 2025. [En línea]. Disponible en: <https://logowik.com/visual-studio-code-vector-logo-1-5273.html>