Algoritmos genéticos

Grupo 4

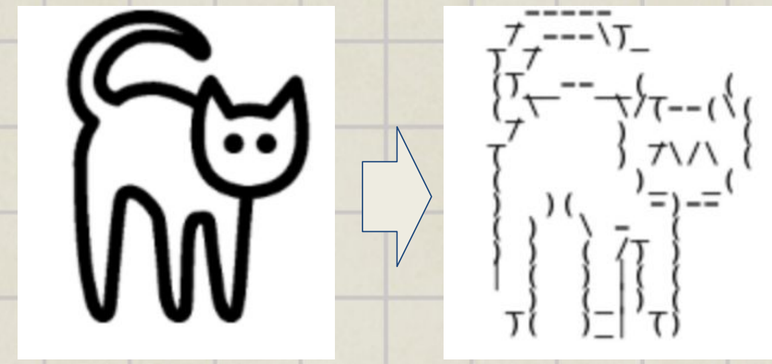
Facundo Criscuolo

Matías Pretel

Lucila Capurro

Ejercicio 1

Objetivo: mediante Algoritmos Genéticos, tomar una imagen cuadrada y representar de la mejor manera posible dicha imagen en un mapa de NxN caracteres ASCII.



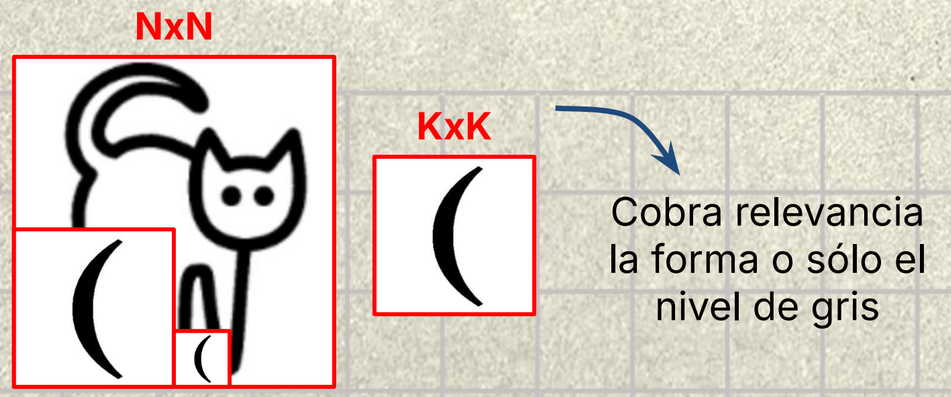
Consideraciones

¿Cuál es el la relación de tamaño entre la imagen y los caracteres?

¿Cuántos/qué caracteres ASCII se pueden usar?

¿Qué tamaño tiene un caracter en pixeles?

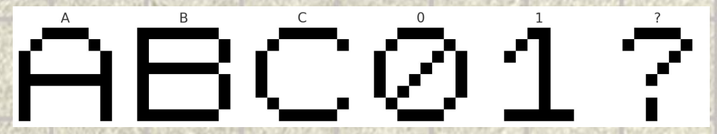
¿Se pueden superponer varios pixeles?



95 caracteres imprimibles: Letras (A-Z, a-z), dígitos (0-9), símbolos como !, @, #, $, etc.

Tamaño mínimo de la imagen para representar todos los caracteres con fuente monoespaciada:

8×8



Permitimos superposición de caracteres

O + 6 = 6

Consideraciones

Las diferentes partes de la imagen son independientes entre sí

Lo podemos dividir en subproblemas

Dividir la imagen en mini-imágenes y aplicar el algoritmo genético en cada una de ellas

Optimizar cada mimi-imagen llevará a la optimización global

Ejercicio 1

Individuo	Imagen de 8×8 caracteres
Gen	Cada individuo va a tener tantos genes como la cantidad de caracteres que se usan para formar la imagen. Cada gen es binario y permite mostrar u ocultar un determinado carácter. Se pueden habilitar una cantidad limitada de genes para tener superposición de caracteres.
Fitness	La función de fitness será una medida de similitud ambas imágenes.

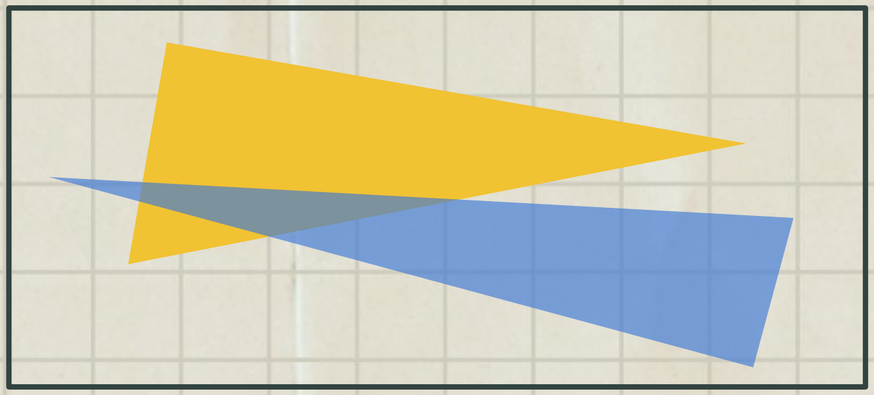
NOTA: Se puede pensar en genes de 8-bits para imágenes en escala de grises

Ejercicio 2

Objetivo: mediante Algoritmos Genéticos, implementar un compresor de imágenes que recibe una imagen y la aproxima con una cantidad de triángulos especificada sobre un canvas blanco.

Implementación

Individuo



Gen

Subgen

4 {color}

6 {posición}

8-bits c/u

Cromosoma

[{R, G, B, A}, {(x_1 y_2), (x_2 y_2), (x_3 y_3)},

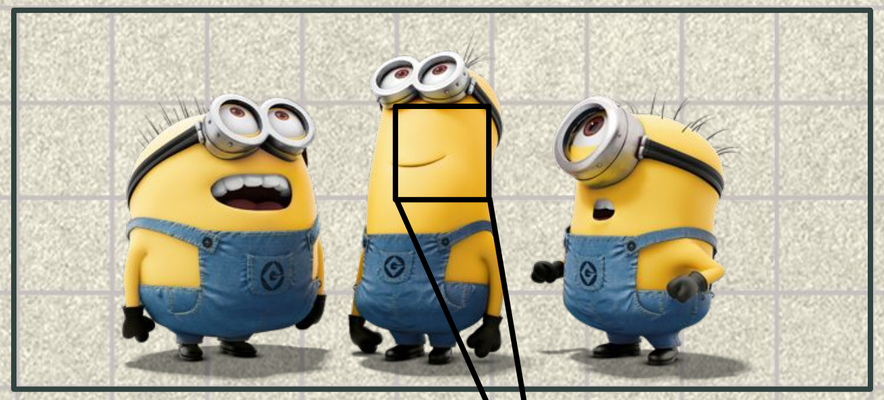
{R, G, B, A}, {(x_1 y_2), (x_2 y_2), (x_3 y_3)}]

Implementación

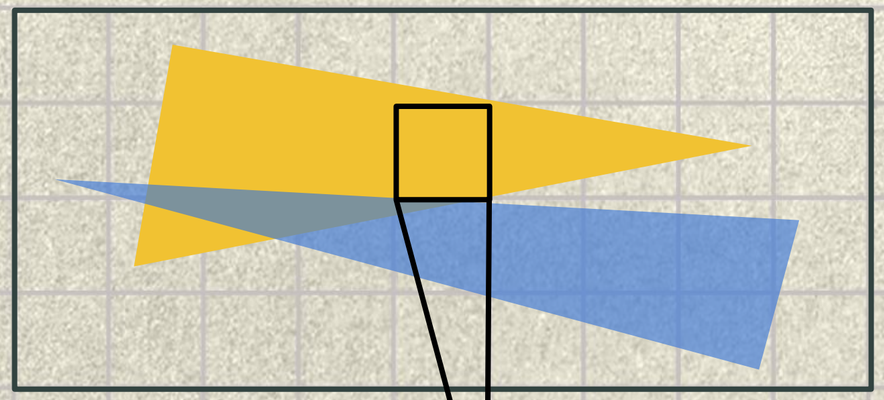
Métodos de selección	• Elite • Ruleta • Universal	• Boltzmann • Torneos • Ranking
Criterio para crear nuevas generaciones	• Tradicional	• Sesgo joven
Método de cruza	• Un punto	• Uniforme
Método de mutación	• Multigen (limitada y uniforme) • Completa	
Criterio de corte	• Cantidad de generaciones • Fitness mínimo • Tiempo de procesamiento	

Mean Absolute Error (MAE)

Target (t)



Render (r)

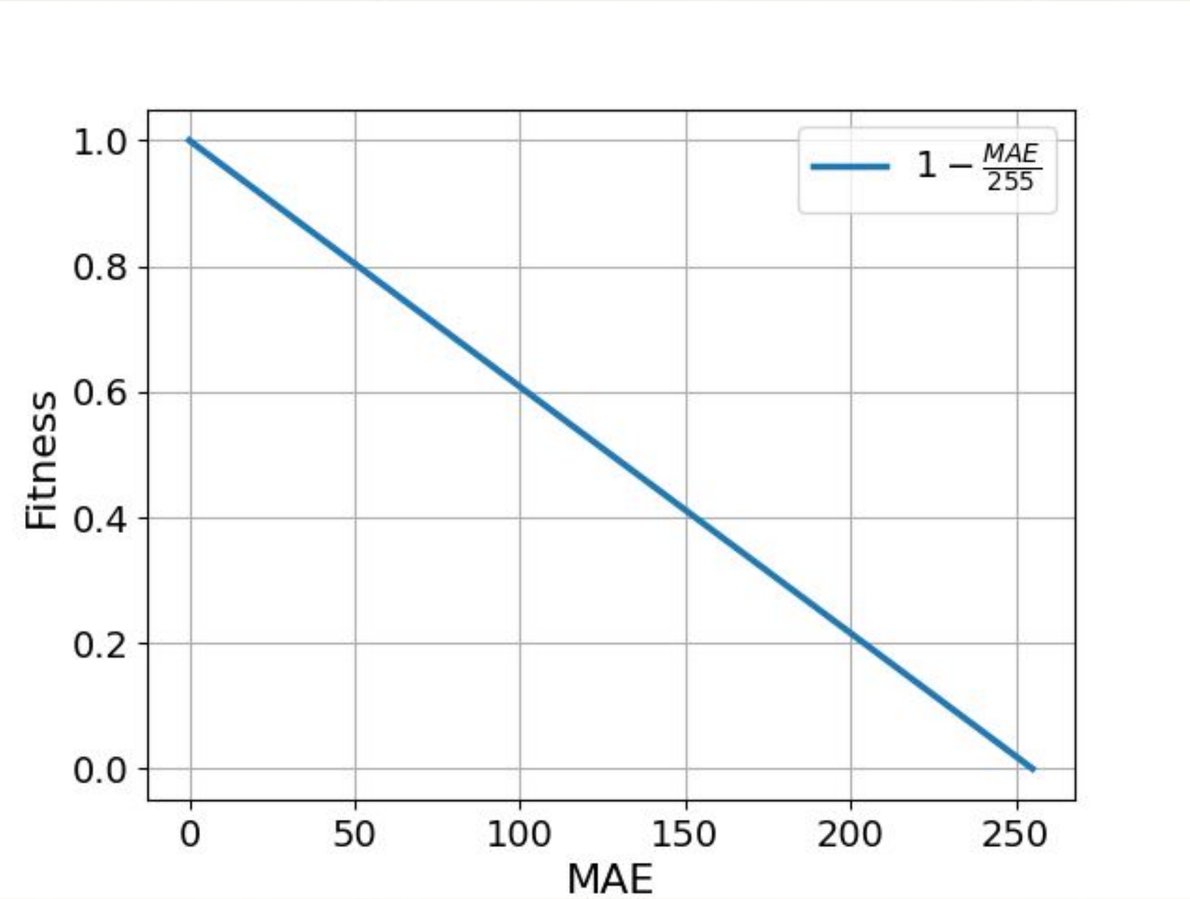


(244, 195, 79)

(241, 194, 50)

$$\mathrm{MAE} = \frac{1}{3MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{c \in \{R,G,B\}} \left| r_{i,j,c} - t_{i,j,c} \right|$$

Fitness



$$\text{Fitness} = 1 - \frac{\text{MAE}}{255}$$

- **Creciente con la disminución del MAE**
- **Lineal**
- **Normalizada**

Consideraciones

La imagen se puede reescalar para reducir su tamaño previo a la compresión	⇒ Reduce el tiempo de procesamiento sin perder resolución
Los 3 vértices del triángulo deben estar contenidos dentro del canvas	⇒ Los vértices pueden tomar cualquier píxel dentro de la imágen
Los triángulos se solapan sobre el canvas de mayor a menor A	⇒ Como queremos un compresor, no queremos que haya triángulos opacos tapando otros de menor A que estén atrás
Los triángulos tienen un alpha mínimo	⇒ Como queremos un compresor y la cantidad de triángulos es un input junto con la imágen, se limita A a un **valor mínimo** → no pueden desaparecer

Imagen 1

1 triángulo negro

sobre un fondo blanco

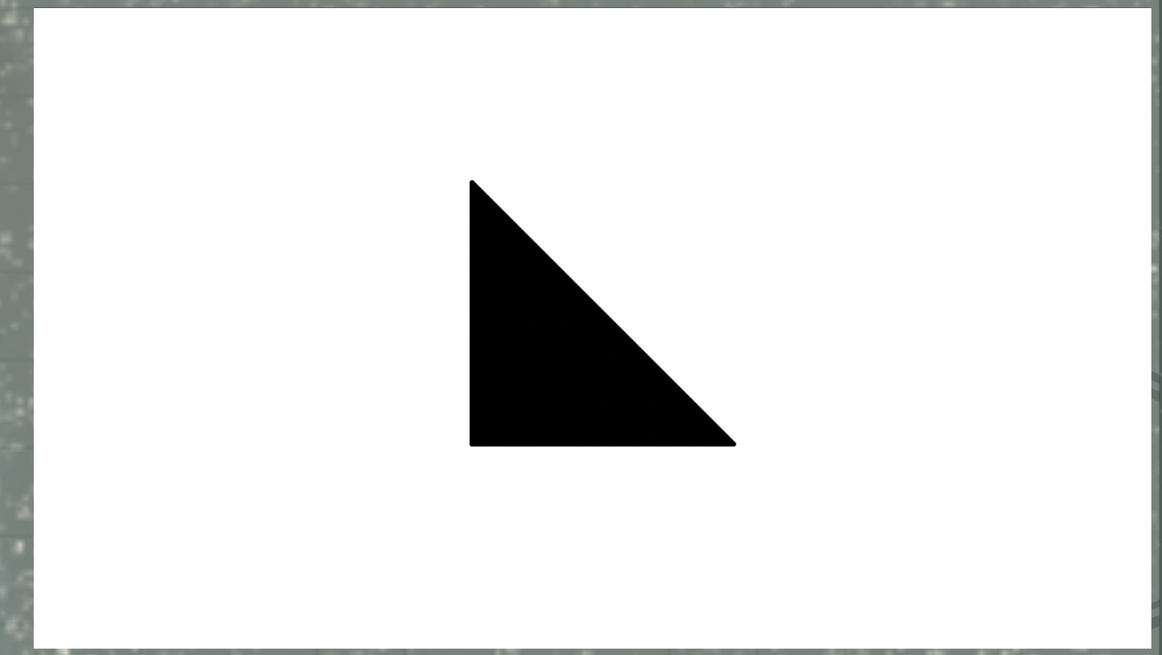
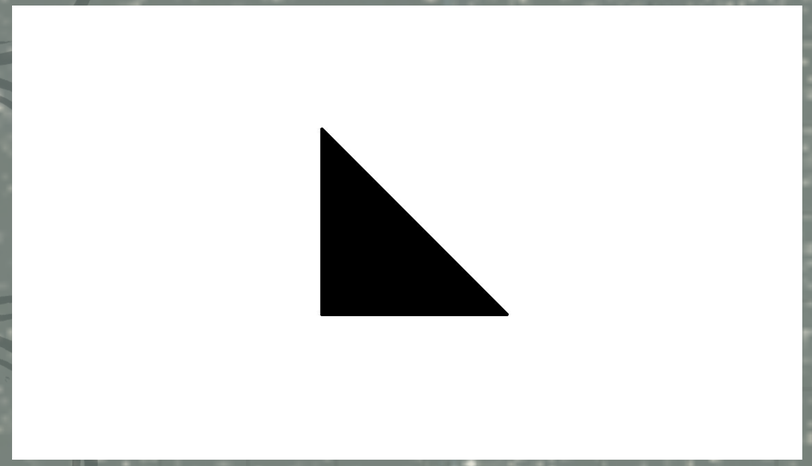


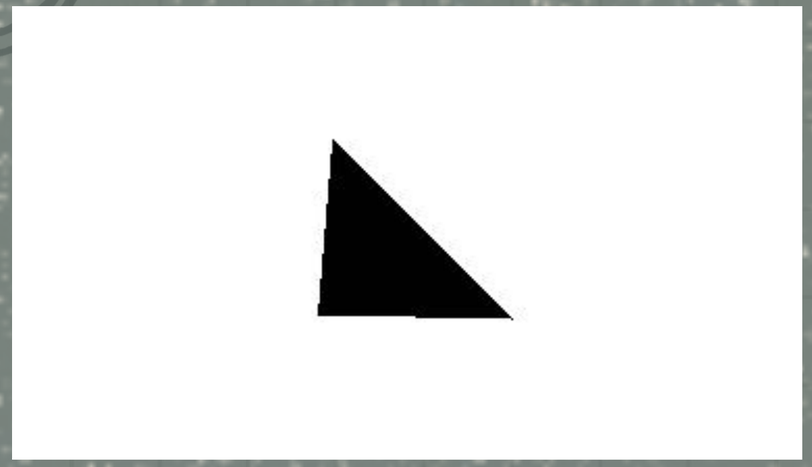
Imagen 1

Limited (p=0.1)

Input: n=1



Output



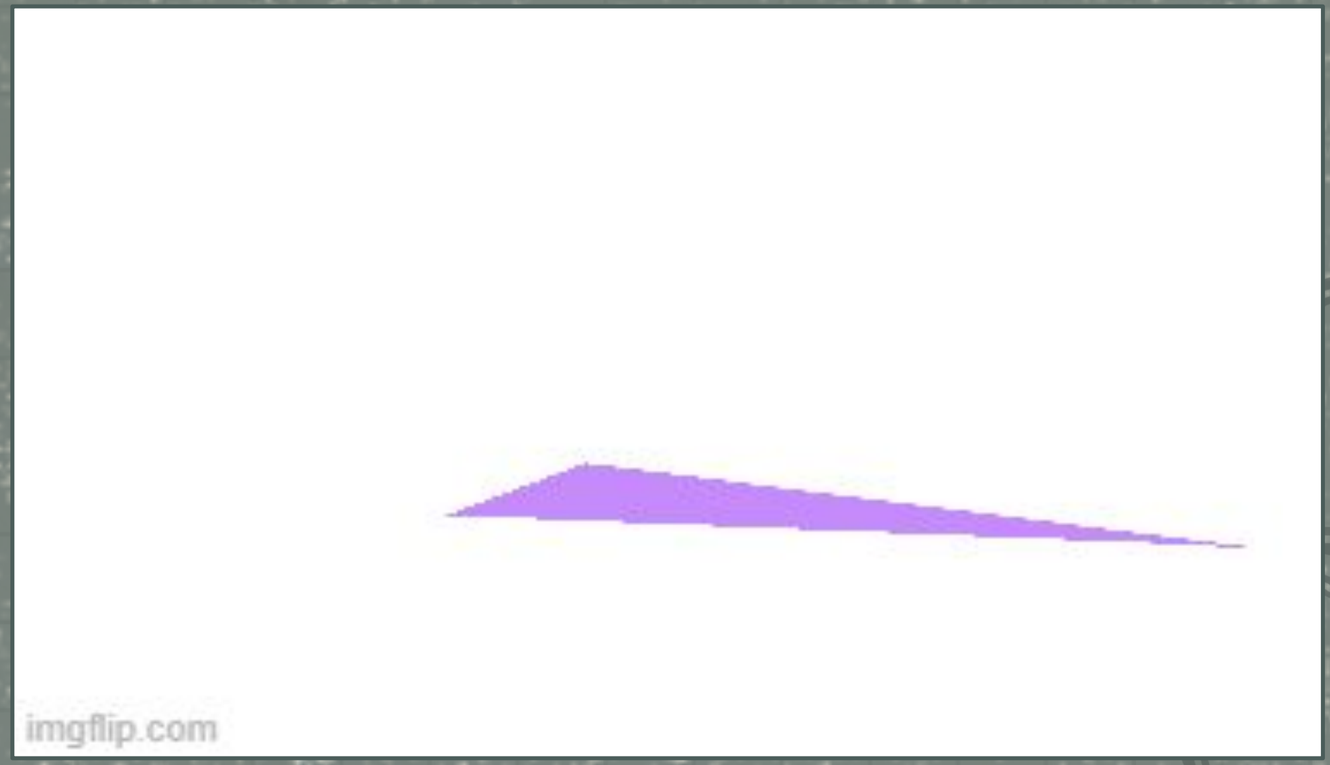
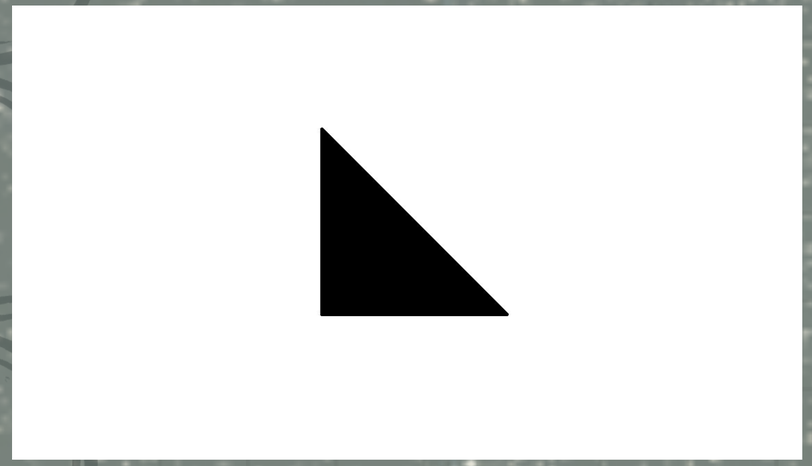


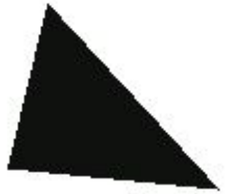
Imagen 1

Limited (p=0.01)

Input: n=1



Output



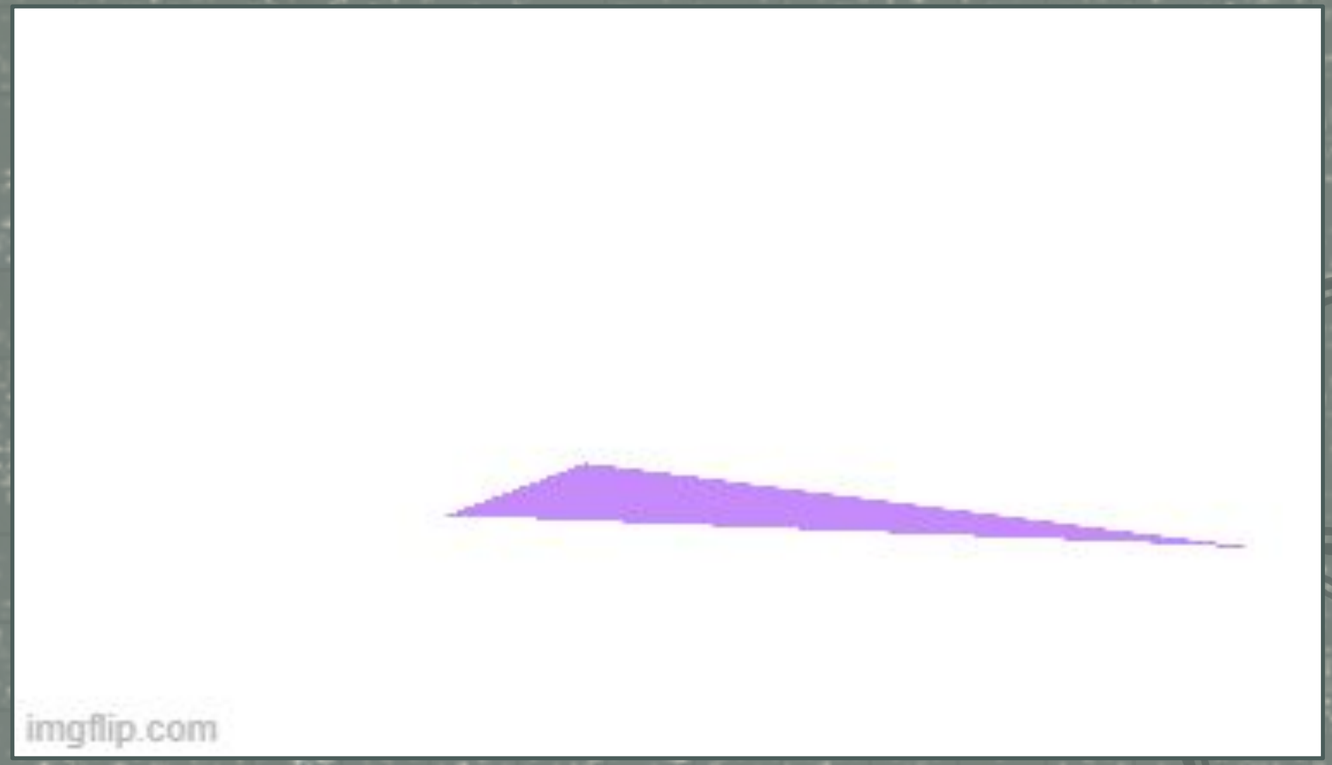
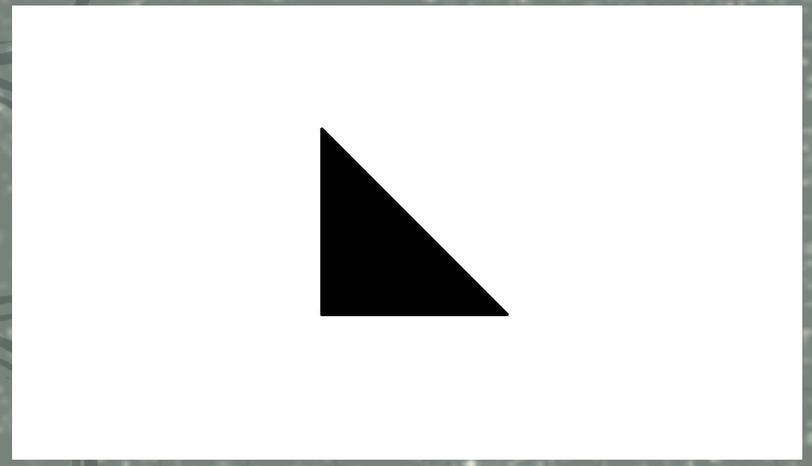


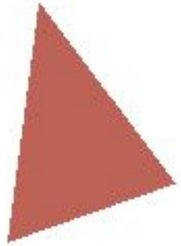
Imagen 1

Limited (p=0.7)

Input: n=1



Output



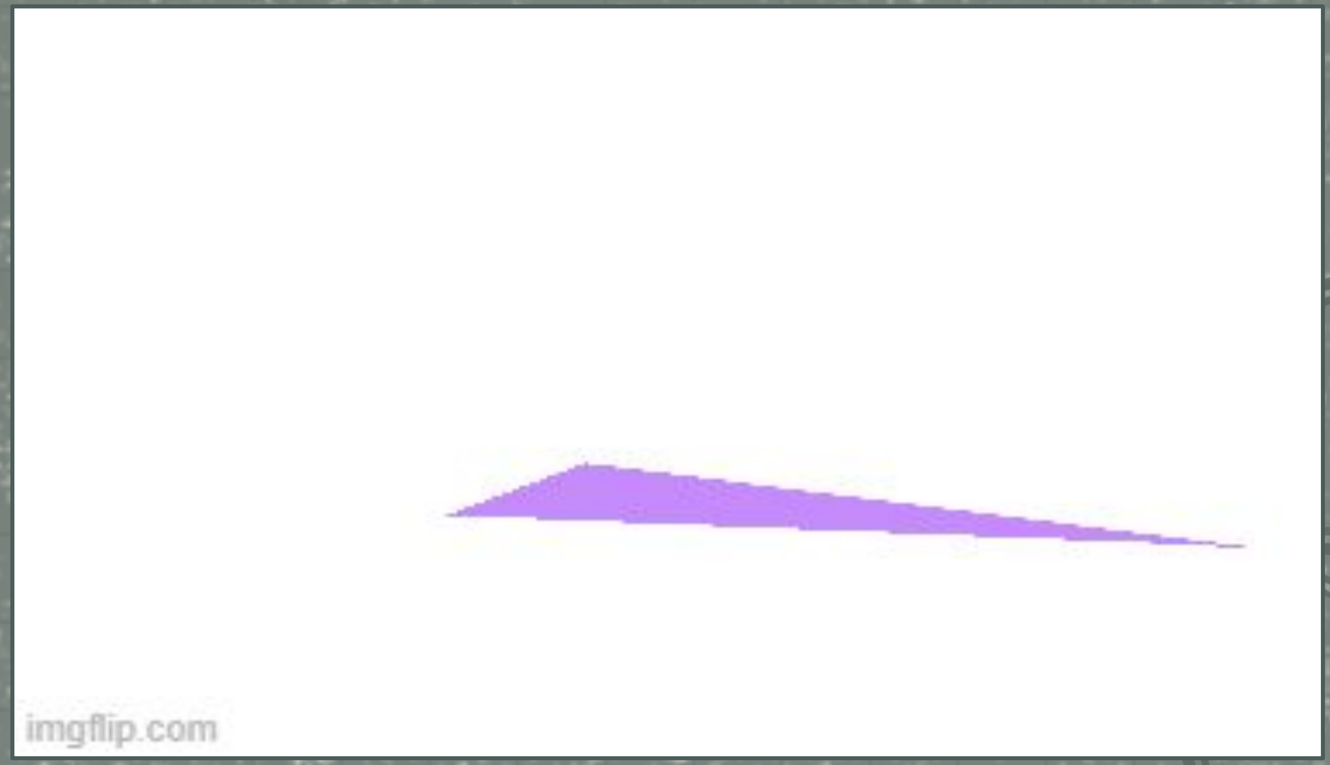
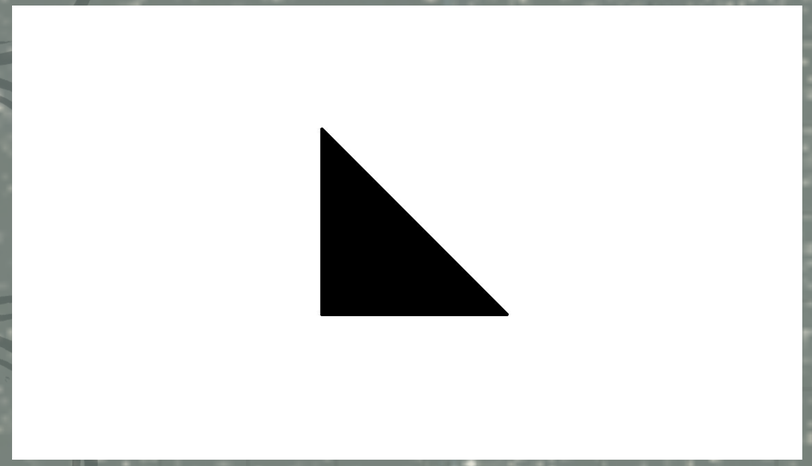


Imagen 1

Uniforme (p=0.1)

Input: n=1



Output

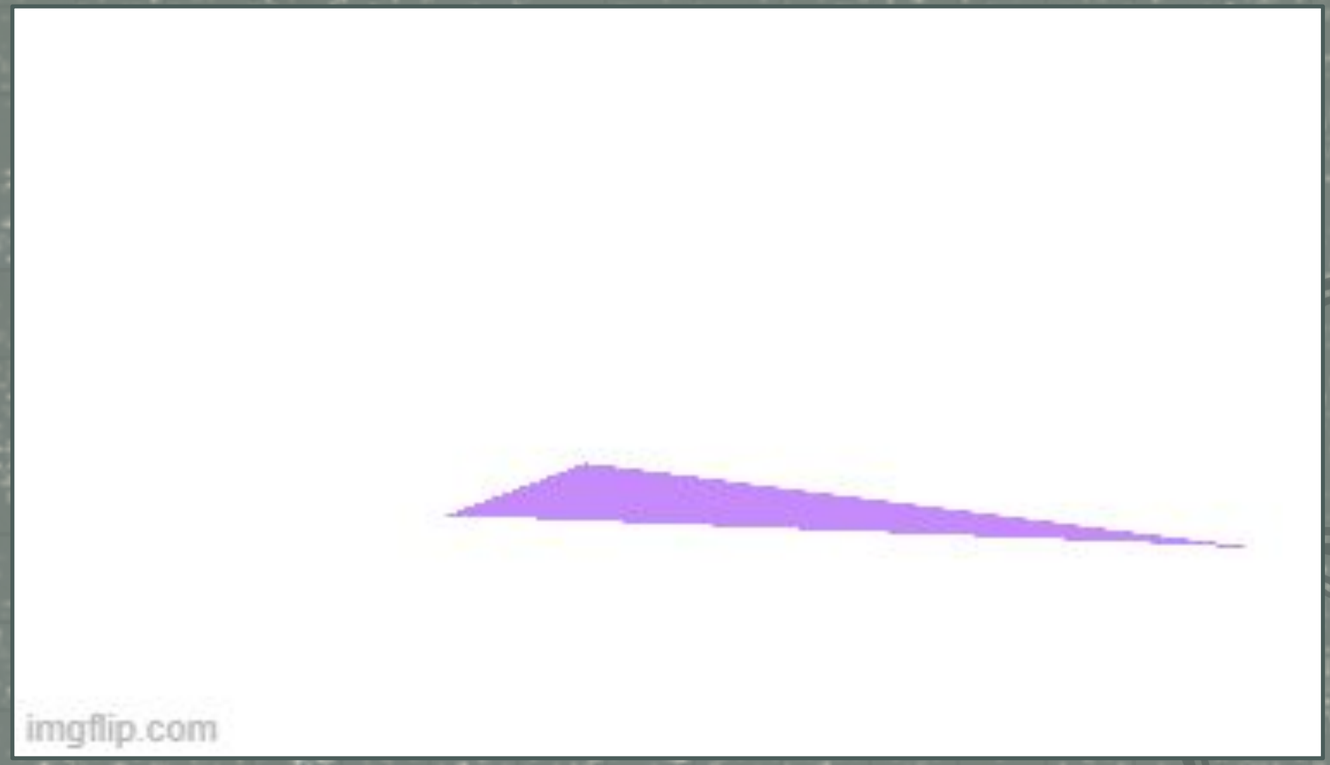


Imagen 1

A pesar de mutar, nunca disminuye el fitness porque usamos selección elitista

p=0,1 pareciera ser una buena tasa de mutación en este caso (0,01 es poco y 0,7 es mucho)

La limitada (muta todos los subgenes) genera más mutaciones que la uniforme (muta 1 solo subgen) en este caso (N=1) aún con la misma p

Inicialmente mejora rápidamente pero al mutar mucho le cuesta más acercarse a una solución mejor hasta pegar un salto

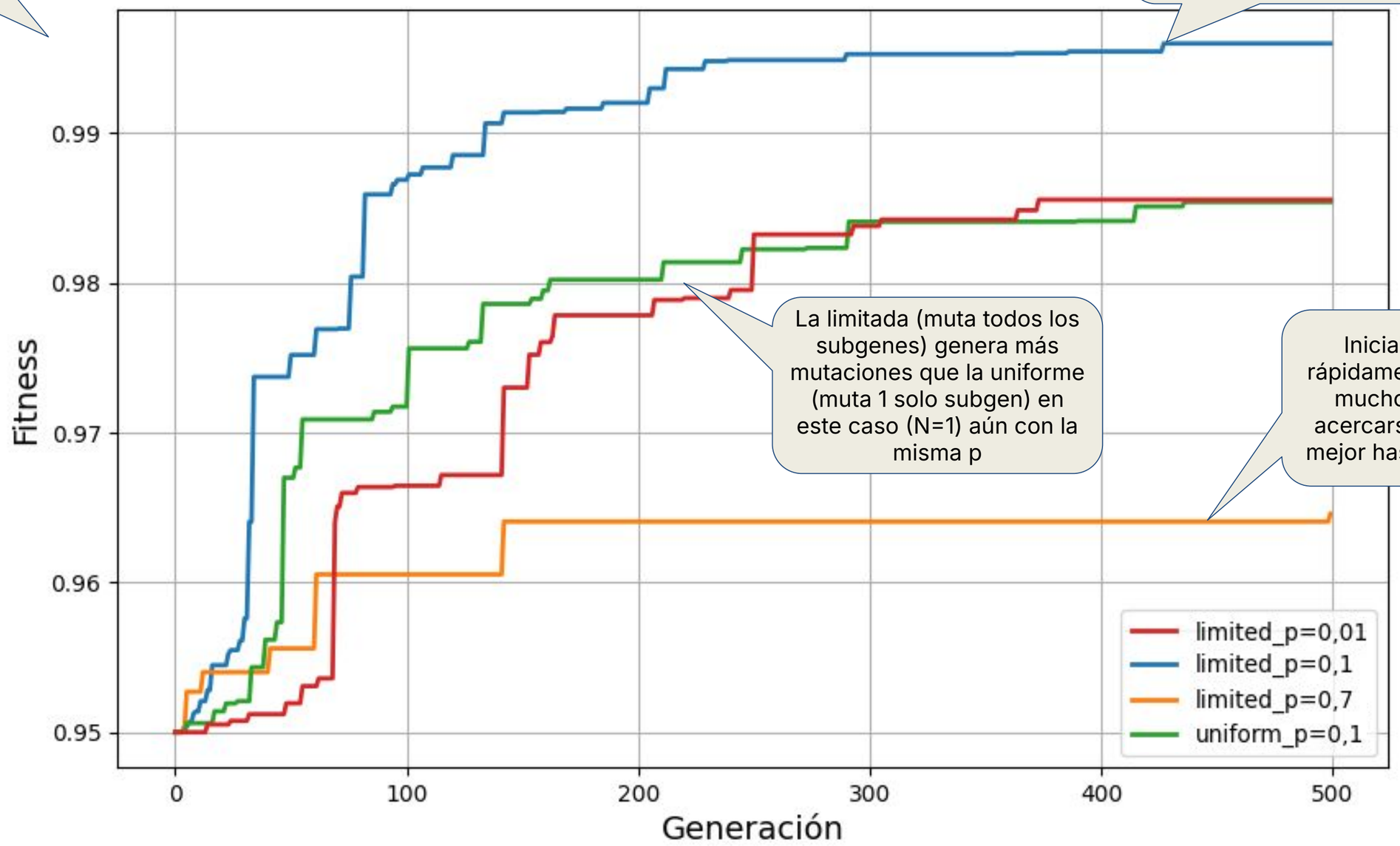
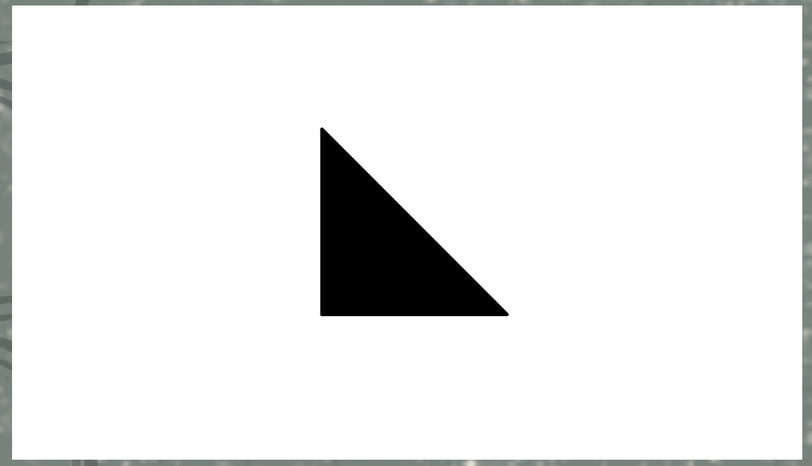


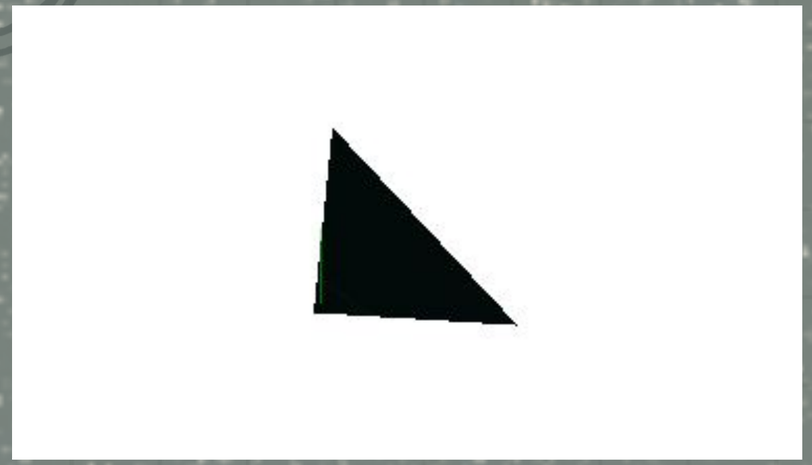
Imagen 1

Limited (p=0.1)

Input: n=5



Output



Para una compresión eficiente se debe evitar perder triángulos (memoria) ocultándolos detrás de otros o haciéndolos desaparecer

⇒Limitar área
⇒Limitar alfa
⇒Solapar triángulos según alfa

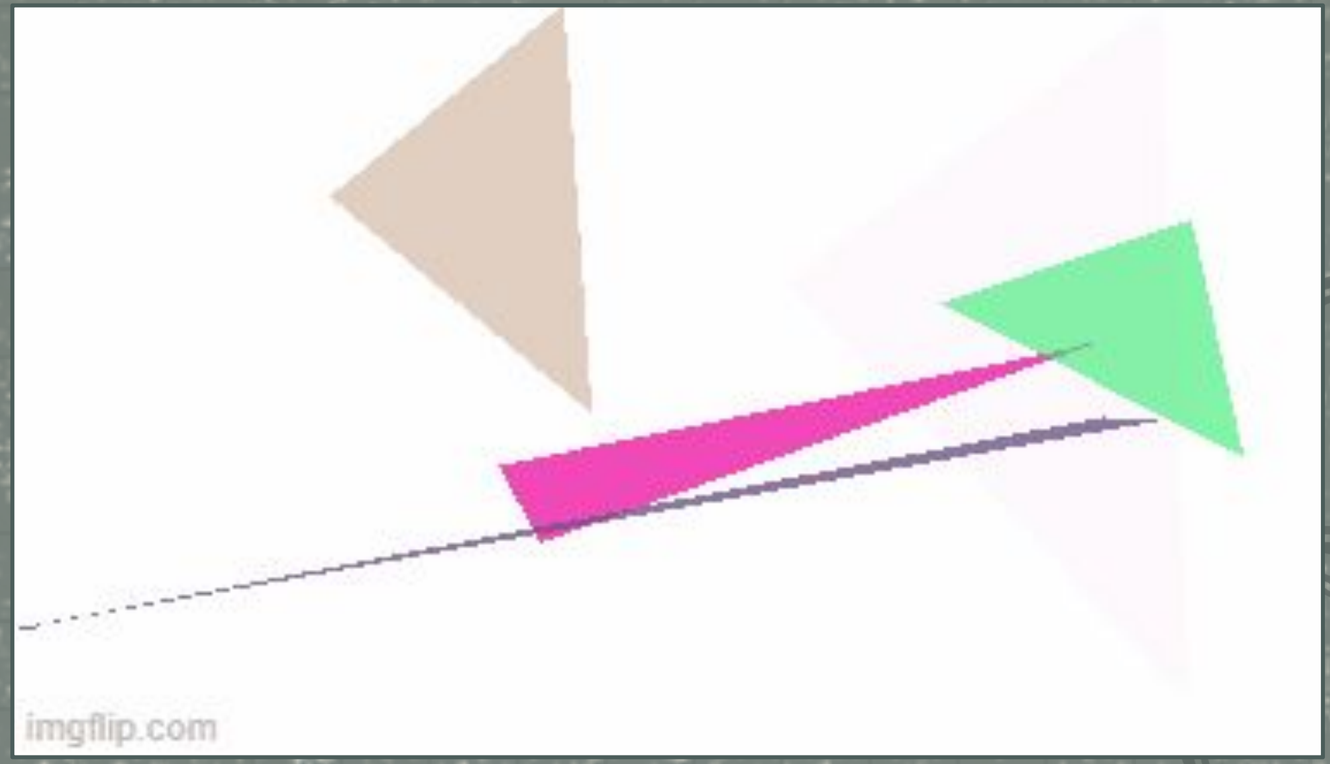


Imagen 2

3 triángulos

(**negro**, **rojo** y **verde**)

sobre un fondo blanco

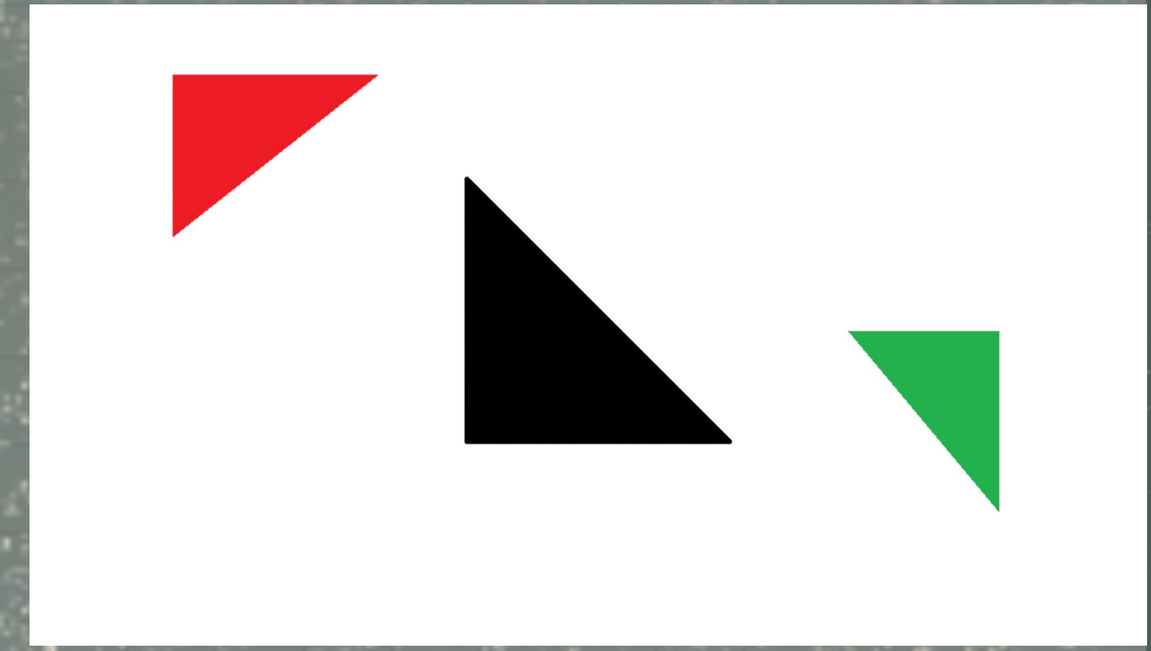
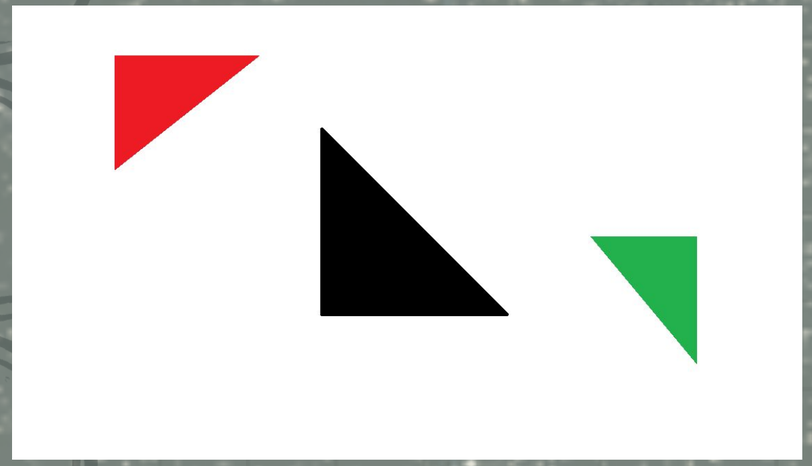


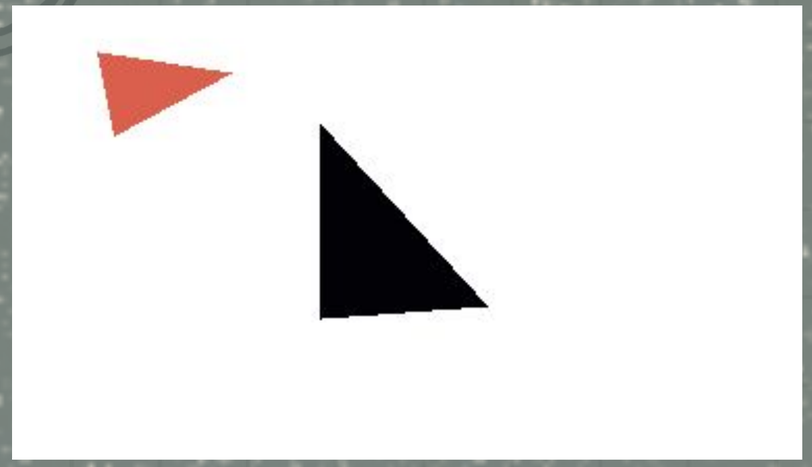
Imagen 2

Limited (M=1, p=0.1)

Input: n=3



Output



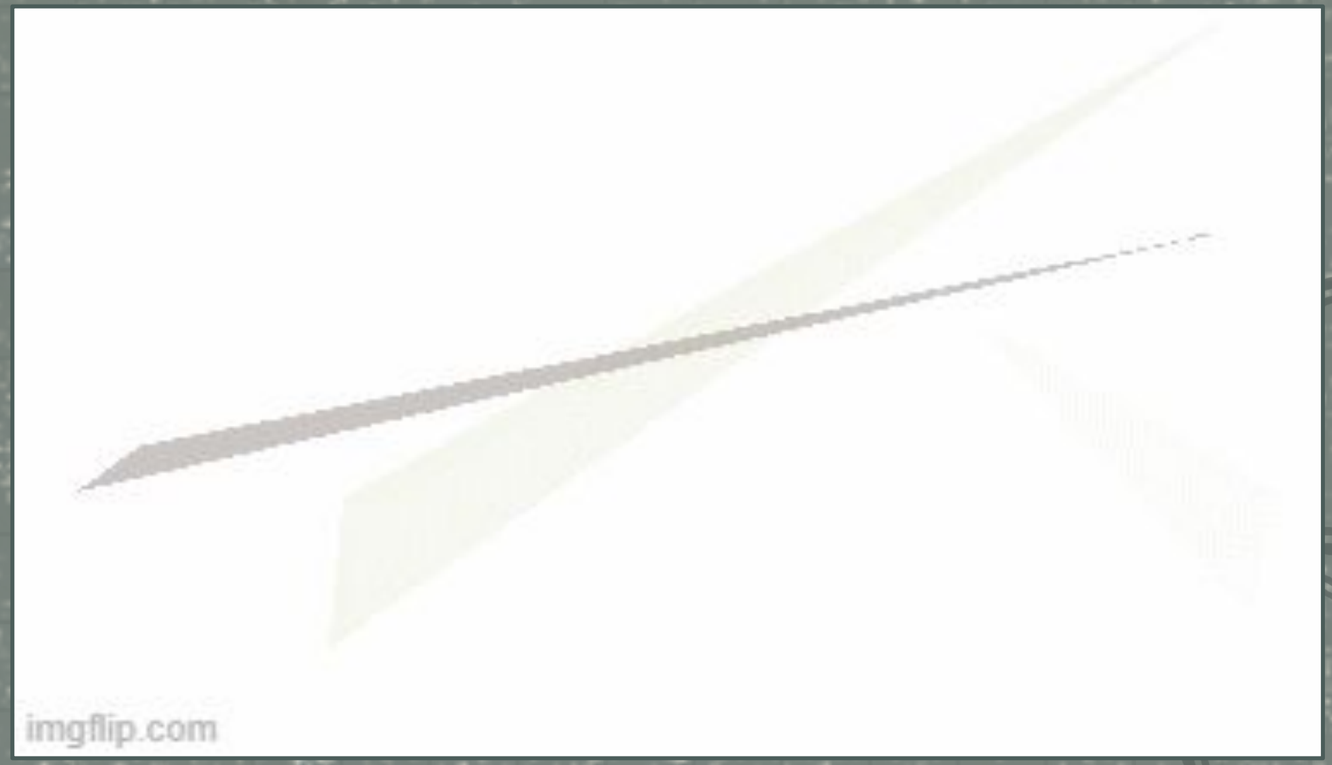
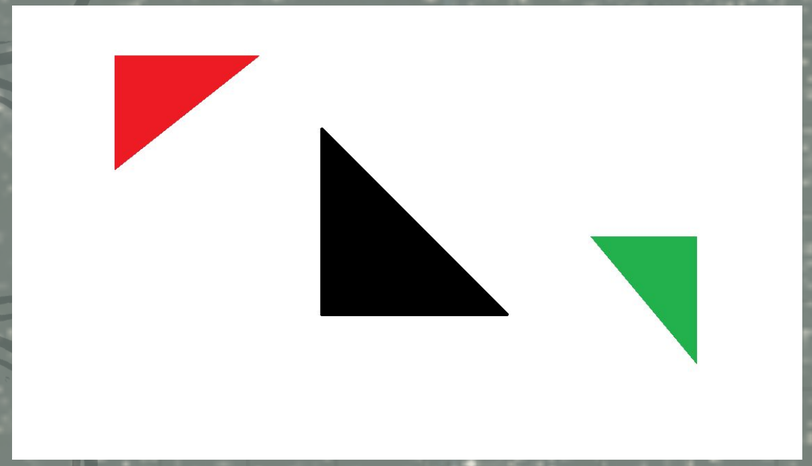


Imagen 2

Alfa y área mín

Input: n=3



Output

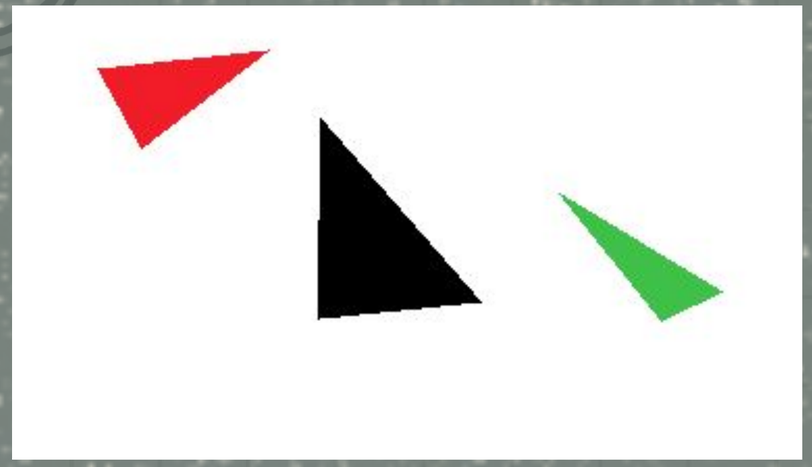
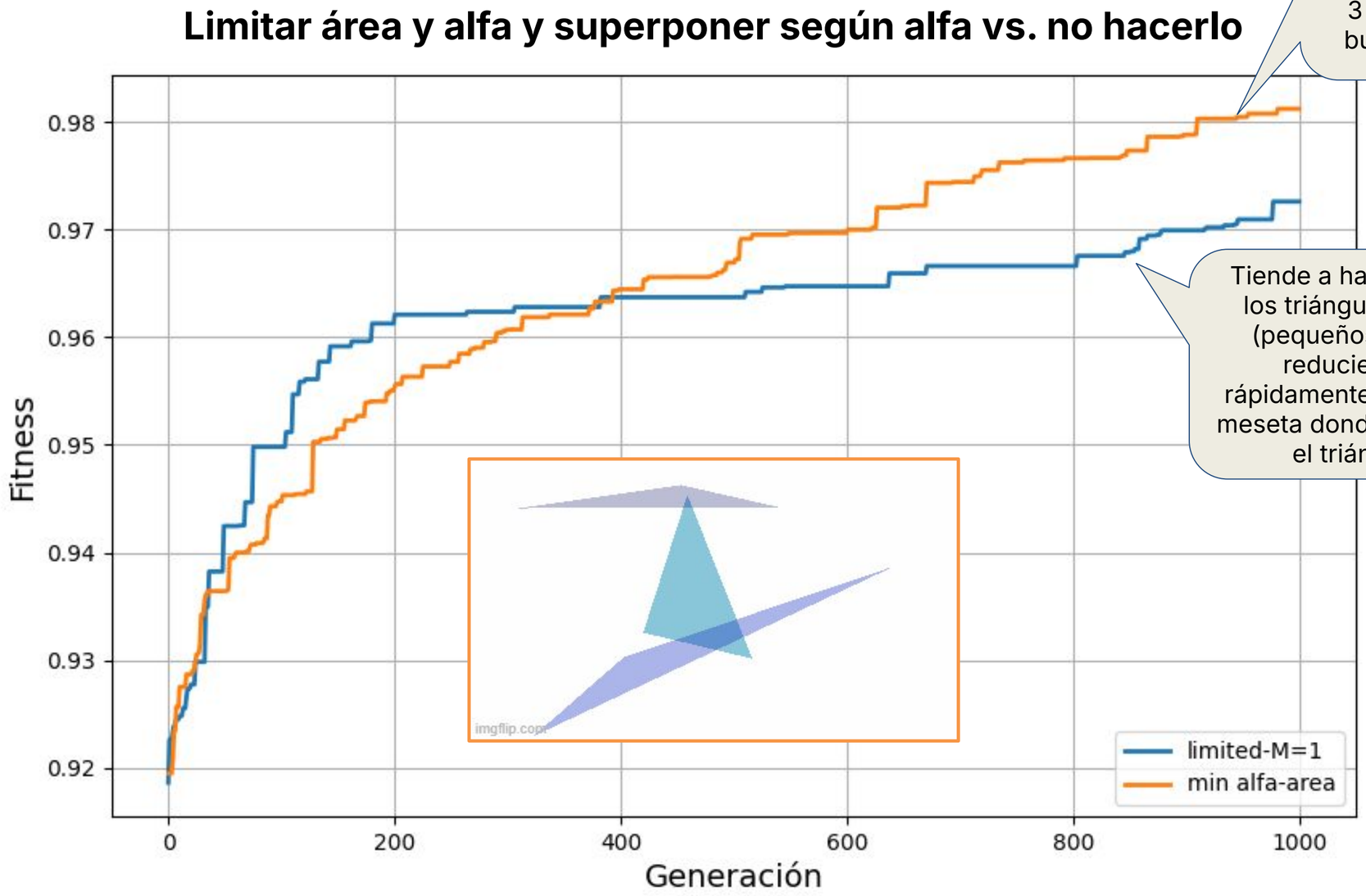


Imagen 2



Evita que desaparezcan los triángulos dando un menor fitness al principio pero permitiendo que encuentre los 3 triángulos alcanzando un buen fitness eventualmente

Tiende a hacer desaparecer los triángulos rojo y verde (pequeños/translúcidos) reduciendo el error rápidamente pero llega a una meseta donde sólo representa el triángulo negro

Imagen 3

Bandera de la

República de Ghana

Imagen 3

Input: n=3

Output

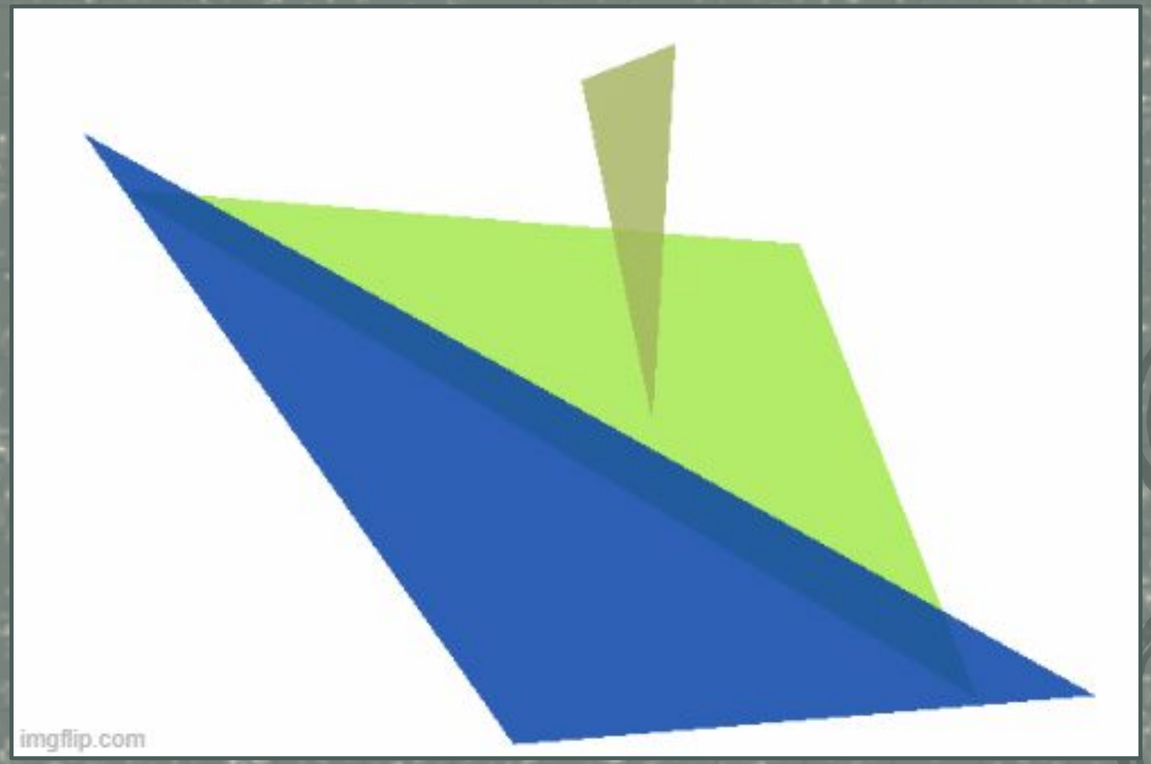


Imagen 3

Input: n=5

Output

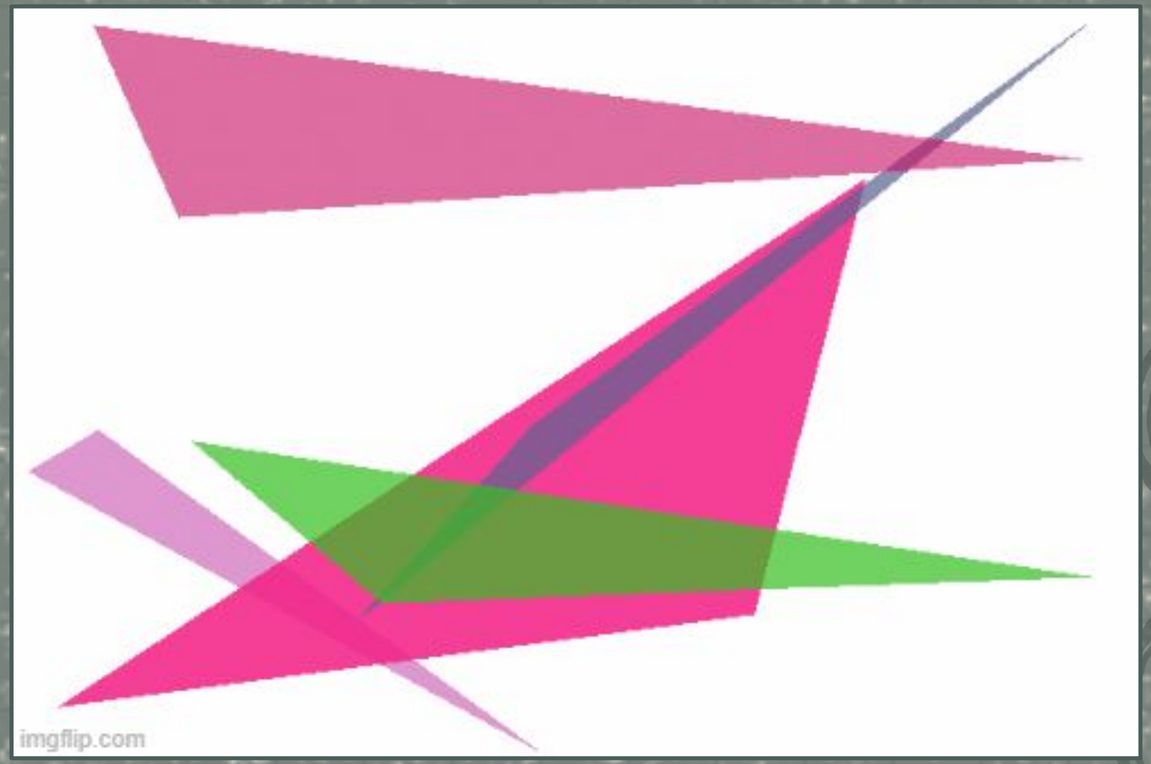


Imagen 3

Input: n=10

Output

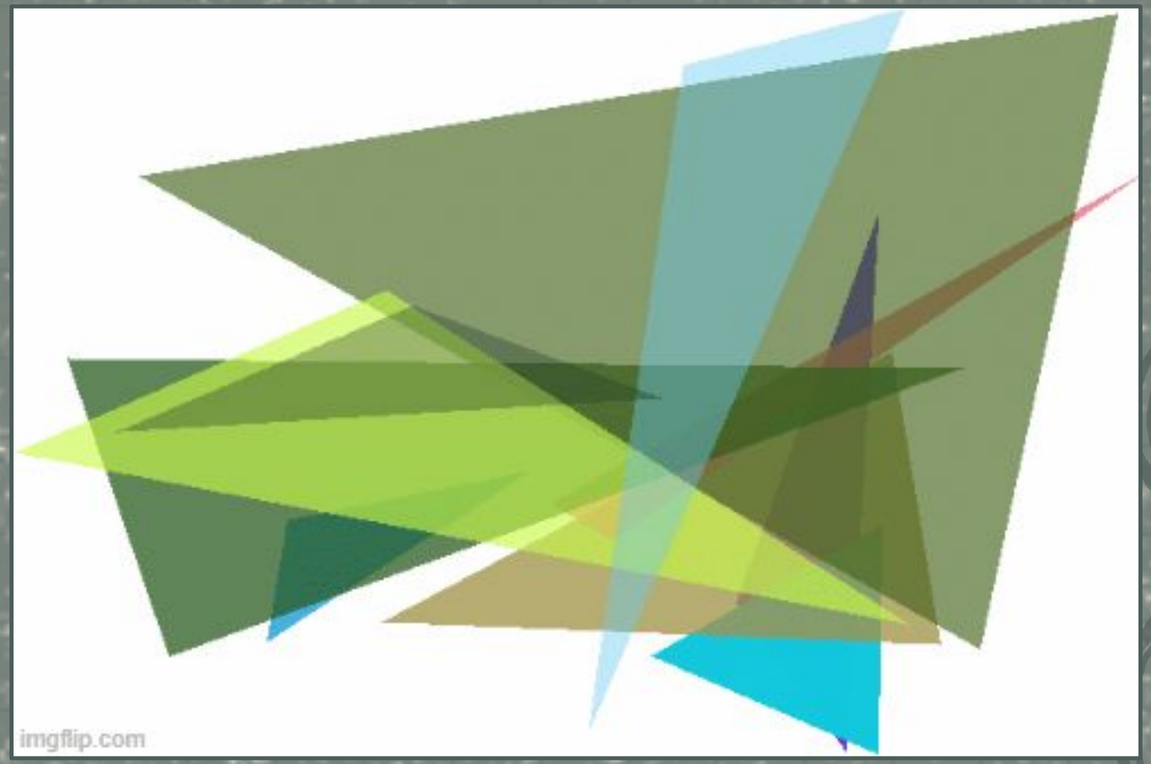
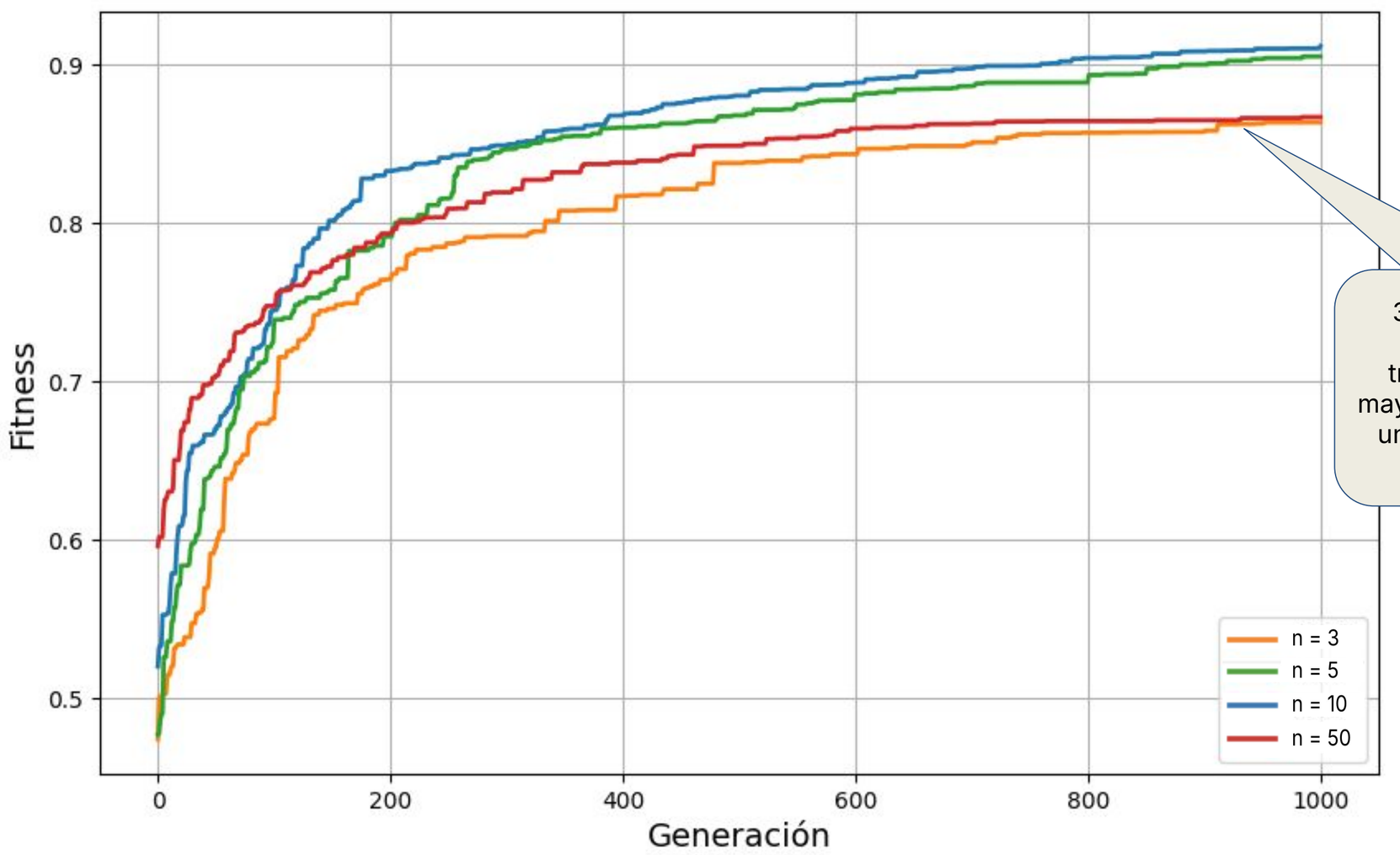


Imagen 3

Input: n=50

Output

Imagen 3



3 triángulos es poco para representarla bien y 50 triángulos requiere de una mayor cantidad de individuos o un criterio de corte superior para poder optimizarla

Imagen 4

Emilio Pettoruti:

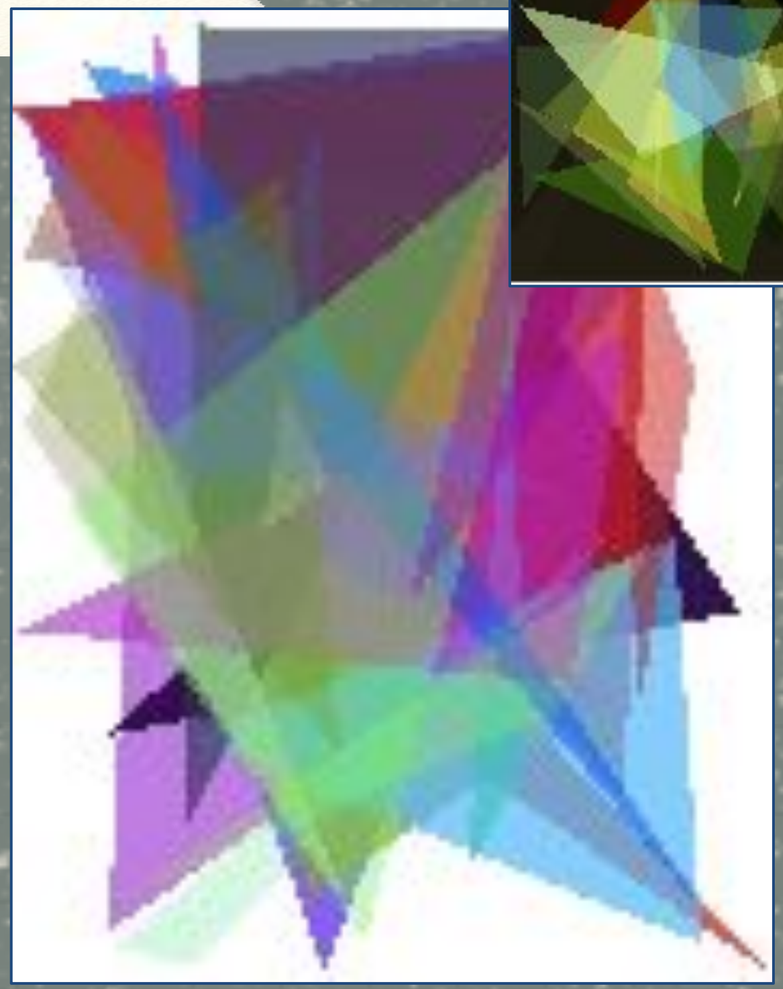
El hombre de la flor amarilla

Imagen 4

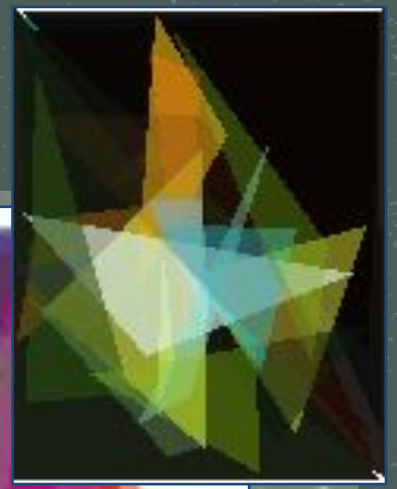
RGB

Input: n=25

3000 gens



Tradicional



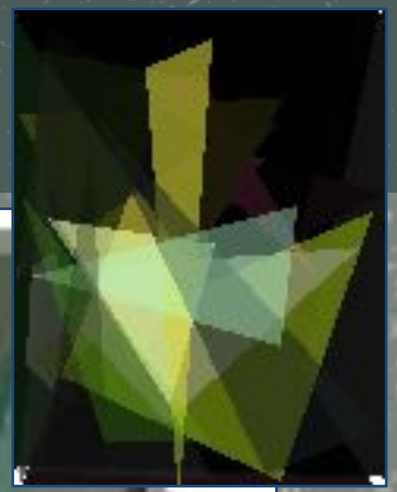
Joven

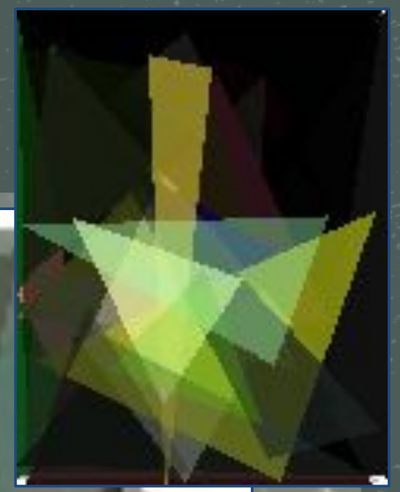
Imagen 4

HSV

Input: n=25

3000 gens





Tradicional

Joven

Imagen 4

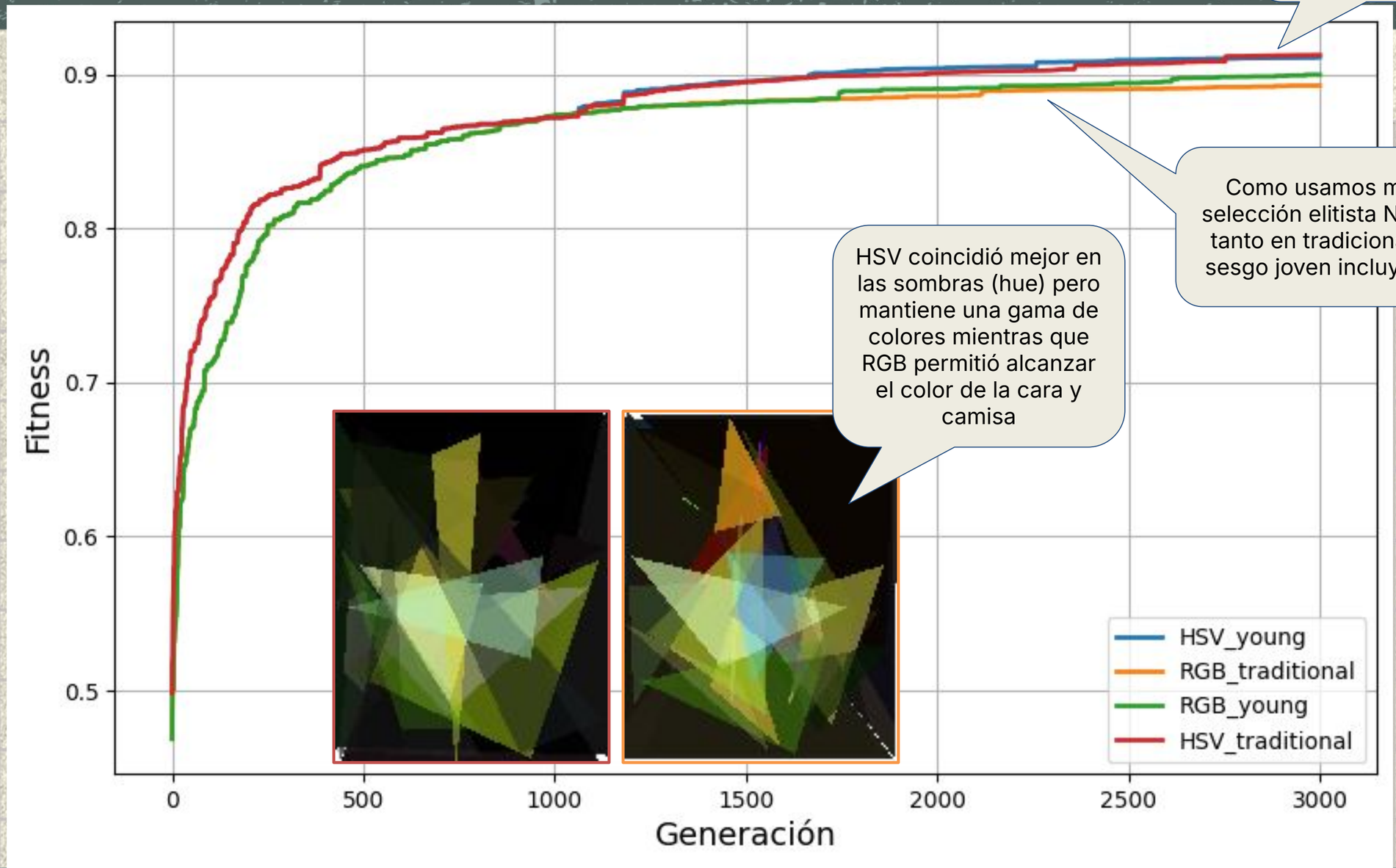

HSV es un poco superior en esta corrida pero no hay tanta diferencia
Como usamos método de selección elitista N=30 y K=10 tanto en tradicional como en sesgo joven incluyen al mejor
HSV coincidió mejor en las sombras (hue) pero mantiene una gama de colores mientras que RGB permitió alcanzar el color de la cara y camisa
Fitness
0.9
0.8
0.7
0.6
0.5
0
500
1000
1500
2000
2500
3000
Generación
HSV_young
RGB_traditional
RGB_young
HSV_traditional

Imagen 5

Emilio Pettoruti:

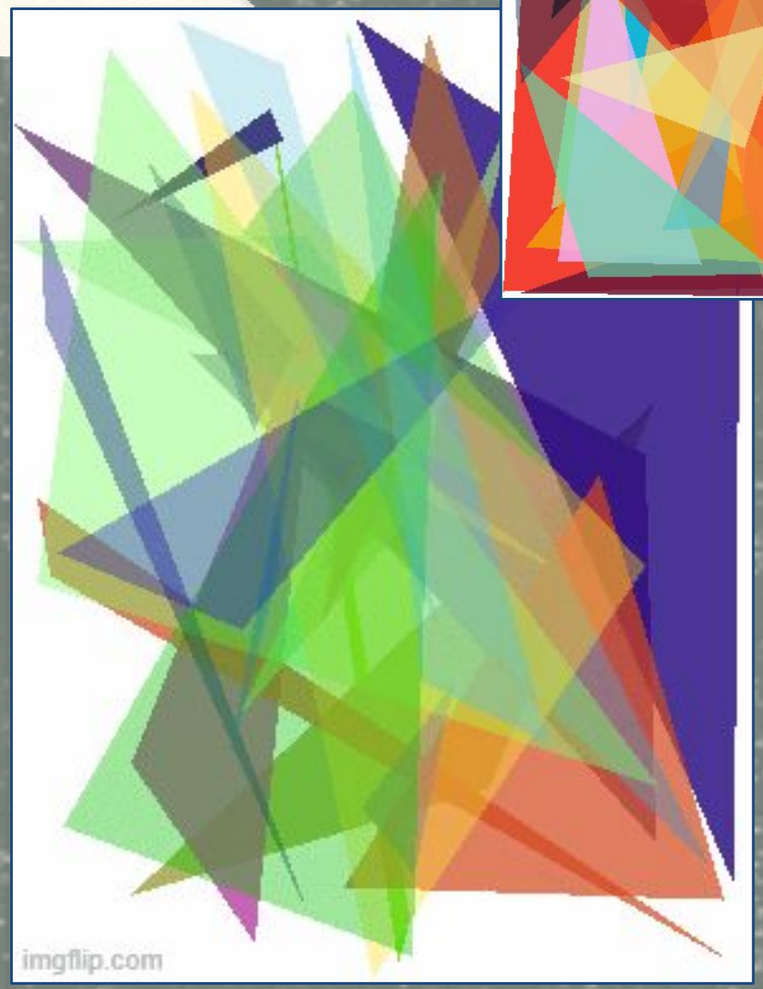
Payaso

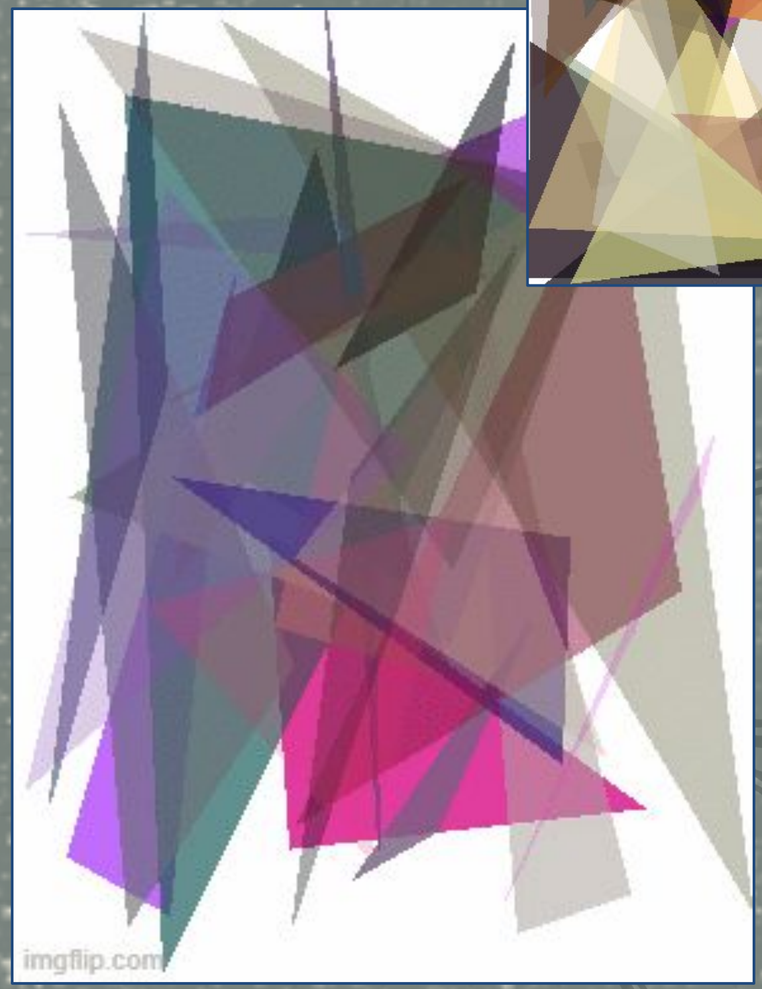
Imagen 5

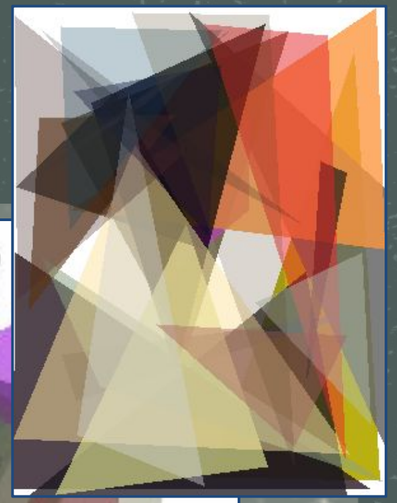
RGB vs HSV

Input: n=25

3000 gens



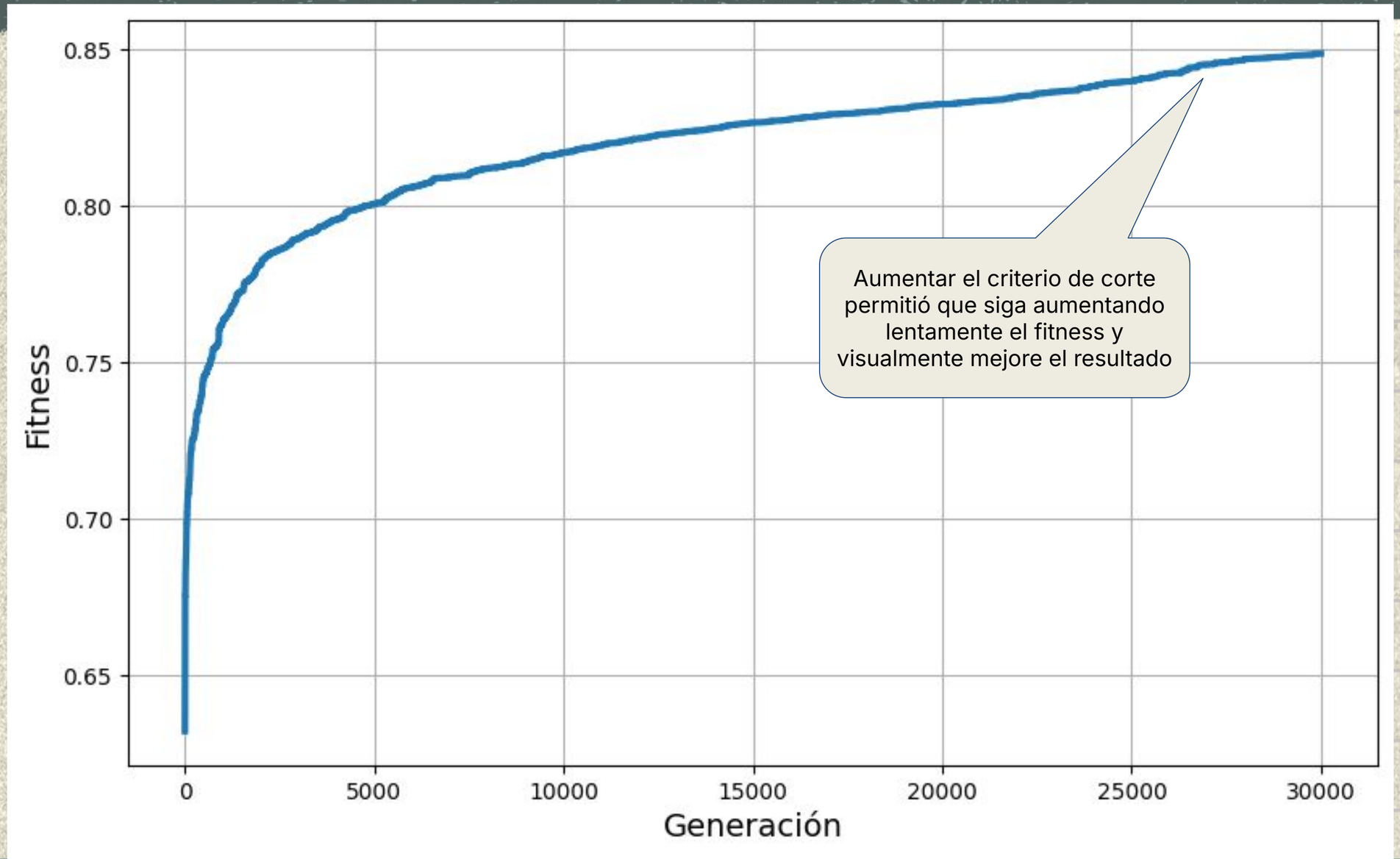




RGB

HSV

Imagen 5


0.85
0.80
0.75
0.70
0.65
Fitness
0
5000
10000
15000
20000
25000
30000
Generación
Aumentar el criterio de corte permitió que siga aumentando lentamente el fitness y visualmente mejore el resultado

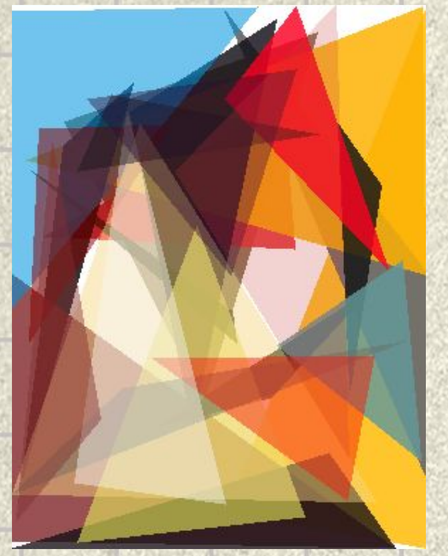


Imagen 6

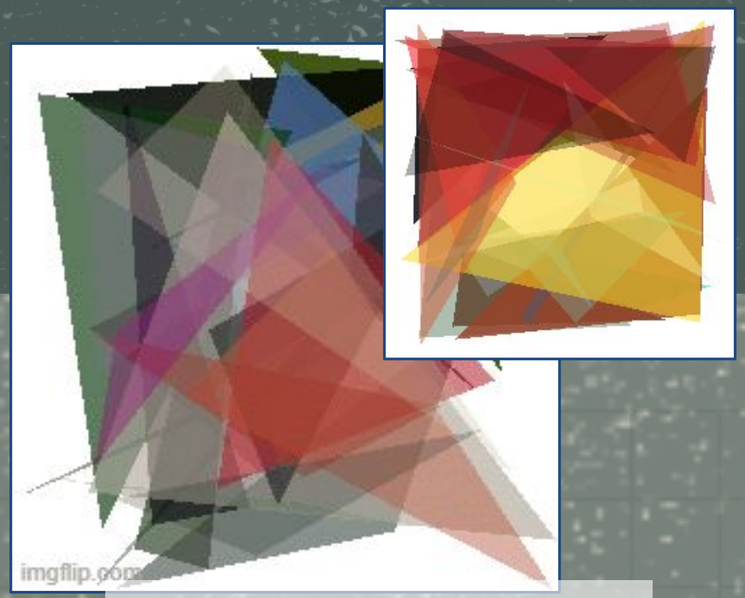
Caja del juego de mesa Catan

sobre un fondo blanco

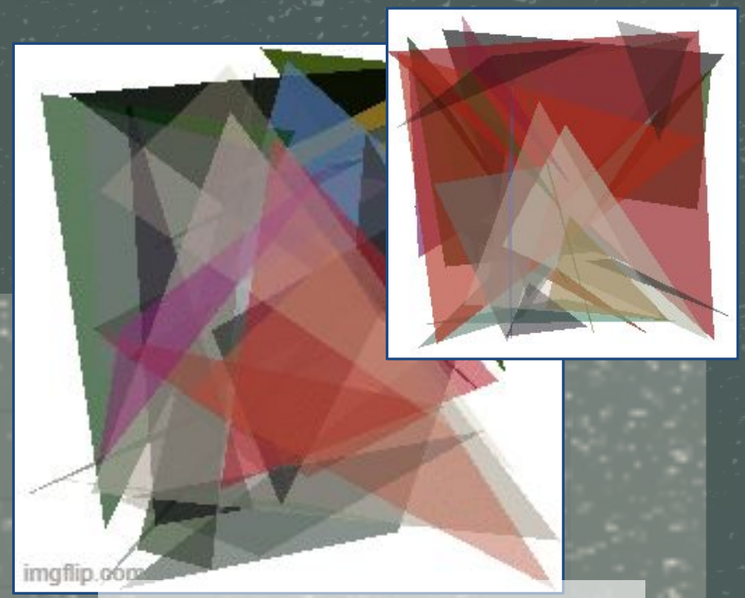
Imagen 6

Input: n=25

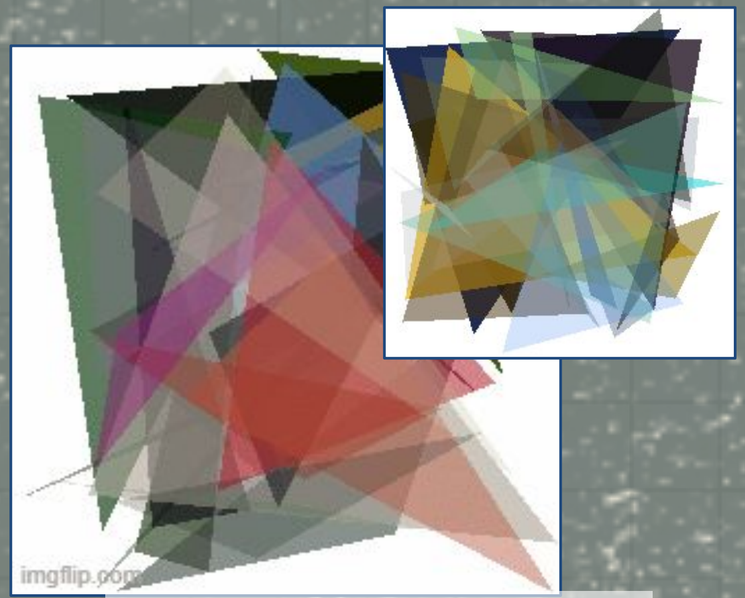
3000 gens



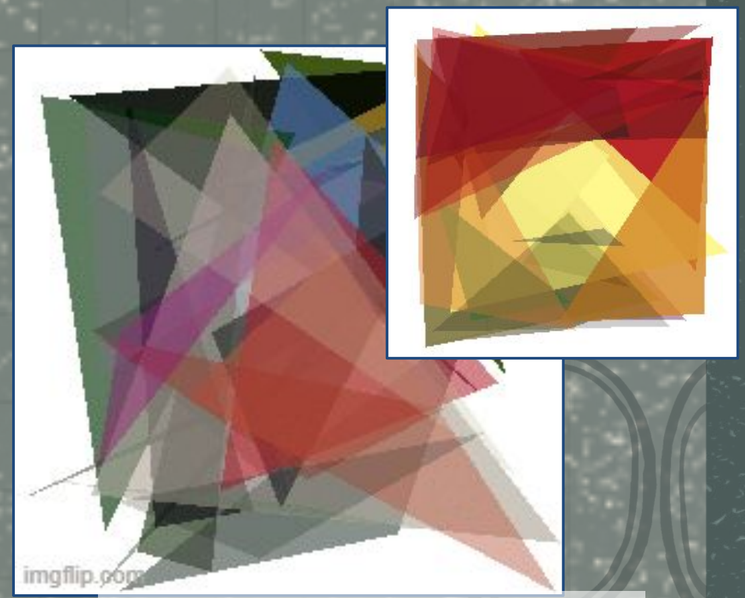
Full Elitism



Elitism-Roulette



Full Roulette



Elitism-Det. Tourn.

Imagen 6

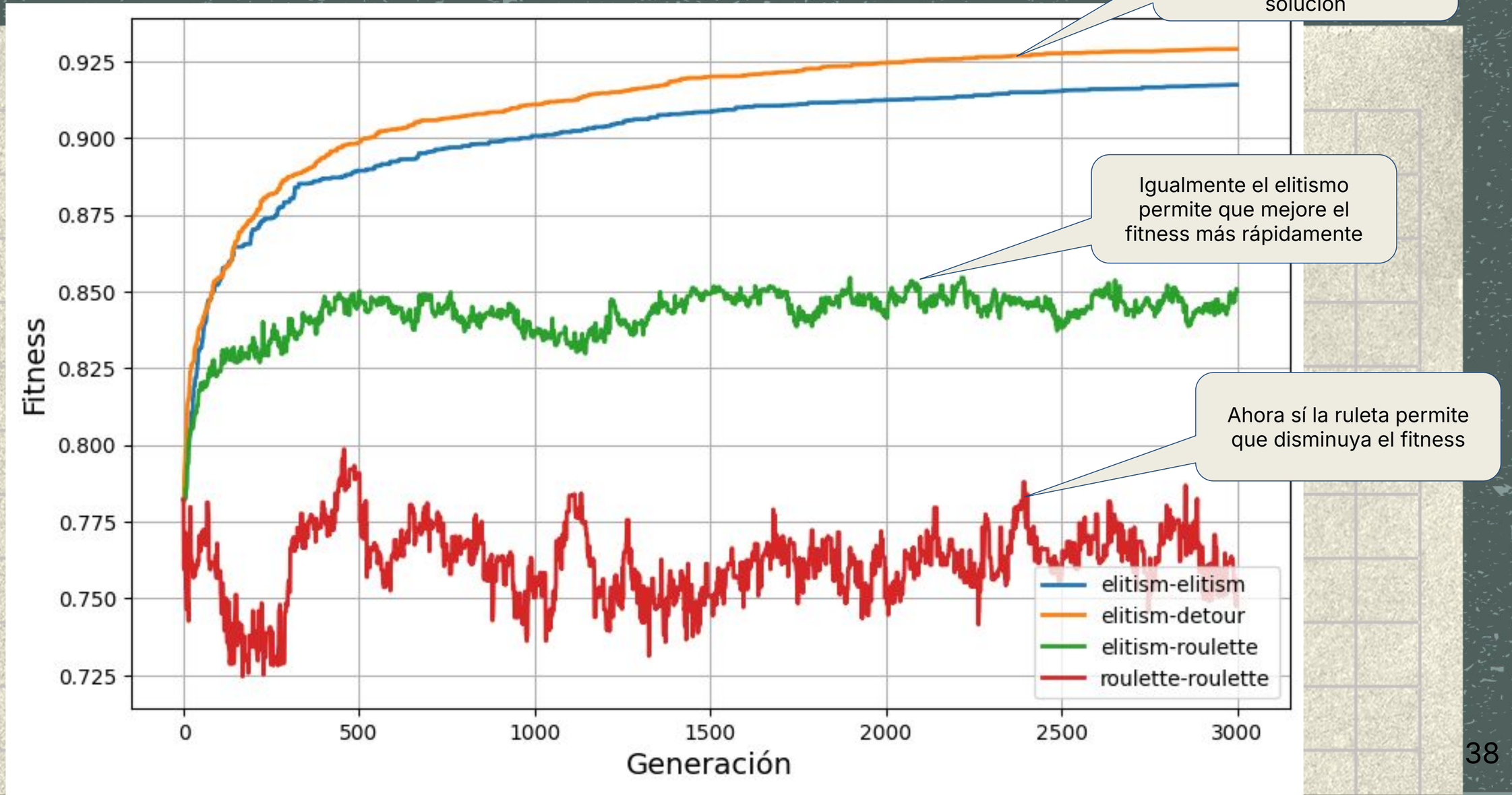
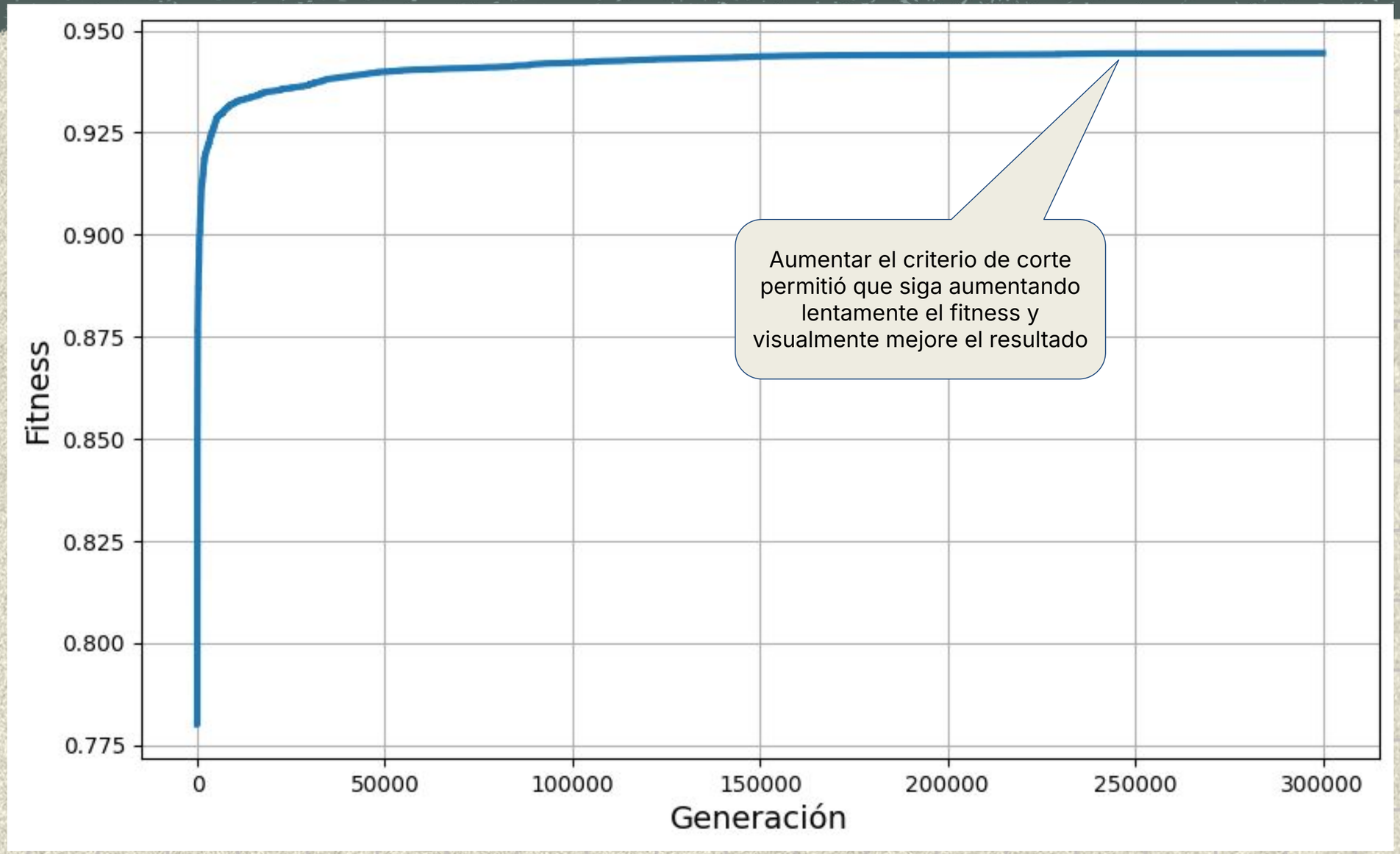
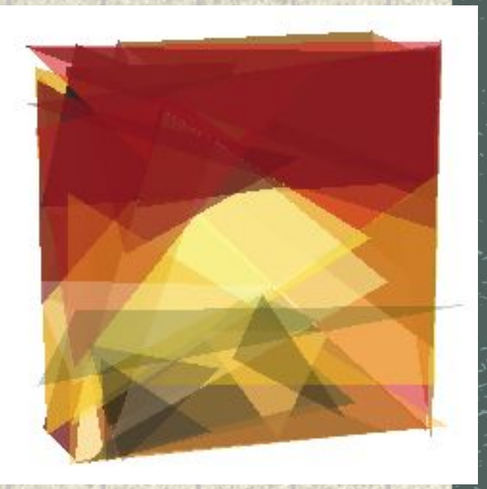

El torneo permite explorar más que el elitismo, encontrando una mejor solución
Igualmente el elitismo permite que mejore el fitness más rápidamente
Ahora sí la ruleta permite que disminuya el fitness
Fitness
0.925
0.900
0.875
0.850
0.825
0.800
0.775
0.750
0.725
0
500
1000
1500
2000
2500
3000
Generación
elitism-elitism
elitism-detour
elitism-roulette
roulette-roulette

Imagen 6


0.950
0.925
0.900
0.875
0.850
0.825
0.800
0.775
Fitness
0
50000
100000
150000
200000
250000
300000
Generación
Aumentar el criterio de corte permitió que siga aumentando lentamente el fitness y visualmente mejore el resultado


KLAUS TEUBER
CATAN
EL JUEGO
Coloniza
Comercia
Construye
DEVIR



Conclusiones

- **Balance entre exploración y explotación**
- **Eficiencia del compresor**
- **Información *a priori* de la imagen**

Incluir heurísticas