

Proyecto 2

Algoritmos

Genéticos

¿Se pueden resolver sudokus utilizando
Algoritmos Genéticos?

Stefan Quintana 1916030

Juan Galicia 20298

Índice

03 Introducción

04 Objetivo

05 Algoritmo

06 Métricas

07 Resultados

08 Discusión

09 Conclusiones

10 Recomendaciones

Introducción

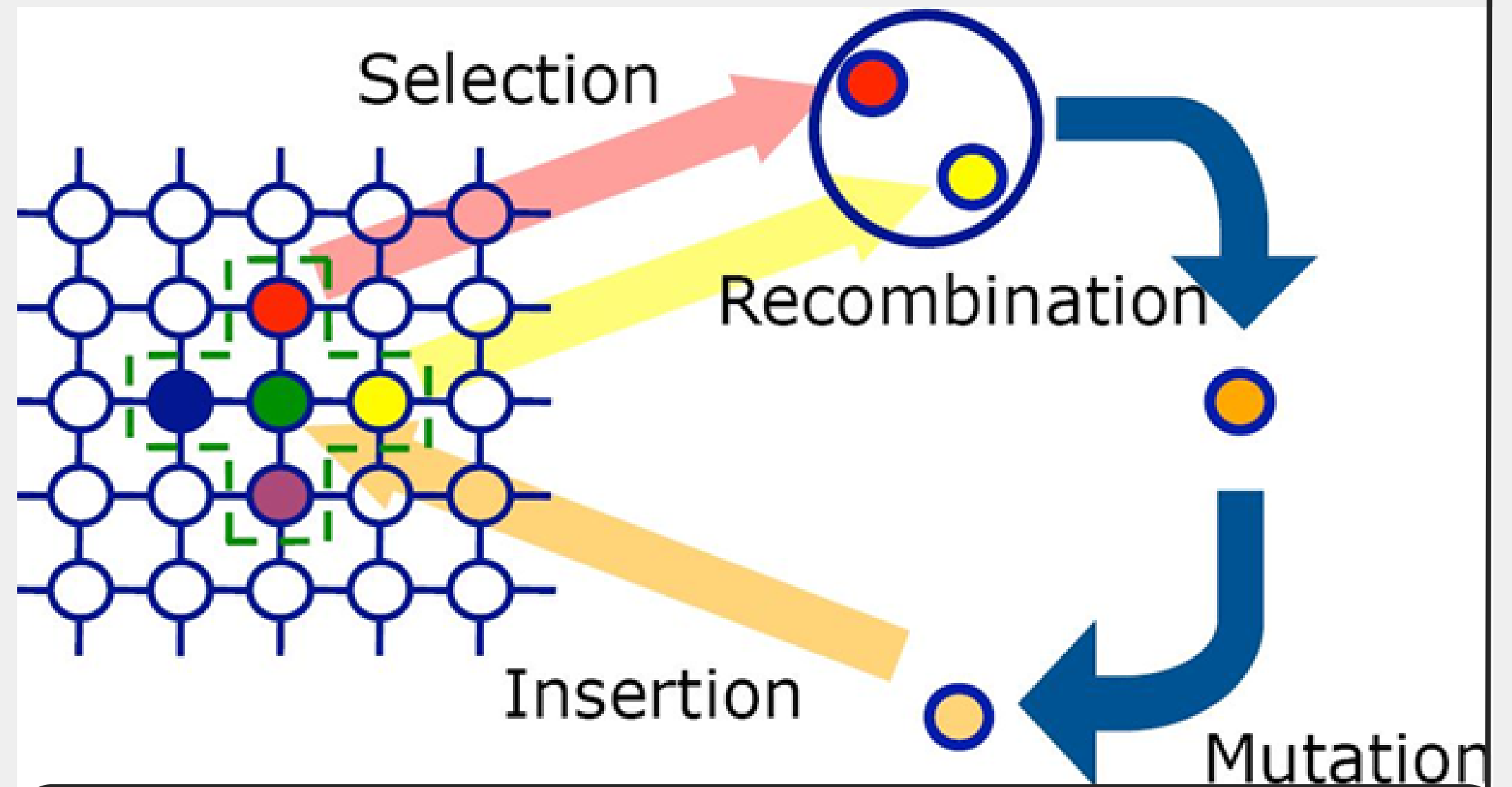
Los algoritmos genéticos buscan imitar la evolución, donde aquellos individuos con características que les ayudan a sobrevivir son los que se reproducen, así sucesivamente hasta tener individuos óptimos para sobrevivir.

8	7	2	3	1	6		5	
	1	5		8			2	
9	3		5				8	1
5	9	3	6	4		2		
2			1		9			4
		7			8	9		6
9	6				5			7
				6				
	2			9				

En el sudoku, nuestro objetivo es completar una cuadrícula 9x9 con números del 1 al 9, sin repeticiones en filas, columnas y bloques de tamaño 3x3.

Objetivos

- Realizar un algoritmo que pueda resolver de manera eficiente cualquier sudoku de 9x9



Lograr implementar diferentes métricas, diferentes maneras de mutar y poder compararlas para saber cuál es la combinación más eficiente.

Lograr verificar si existe una diferencia significativa en la eficiencia del algoritmo en resolver el sudoku según su dificultad impuesta para la resolución por un humano.

Algoritmo

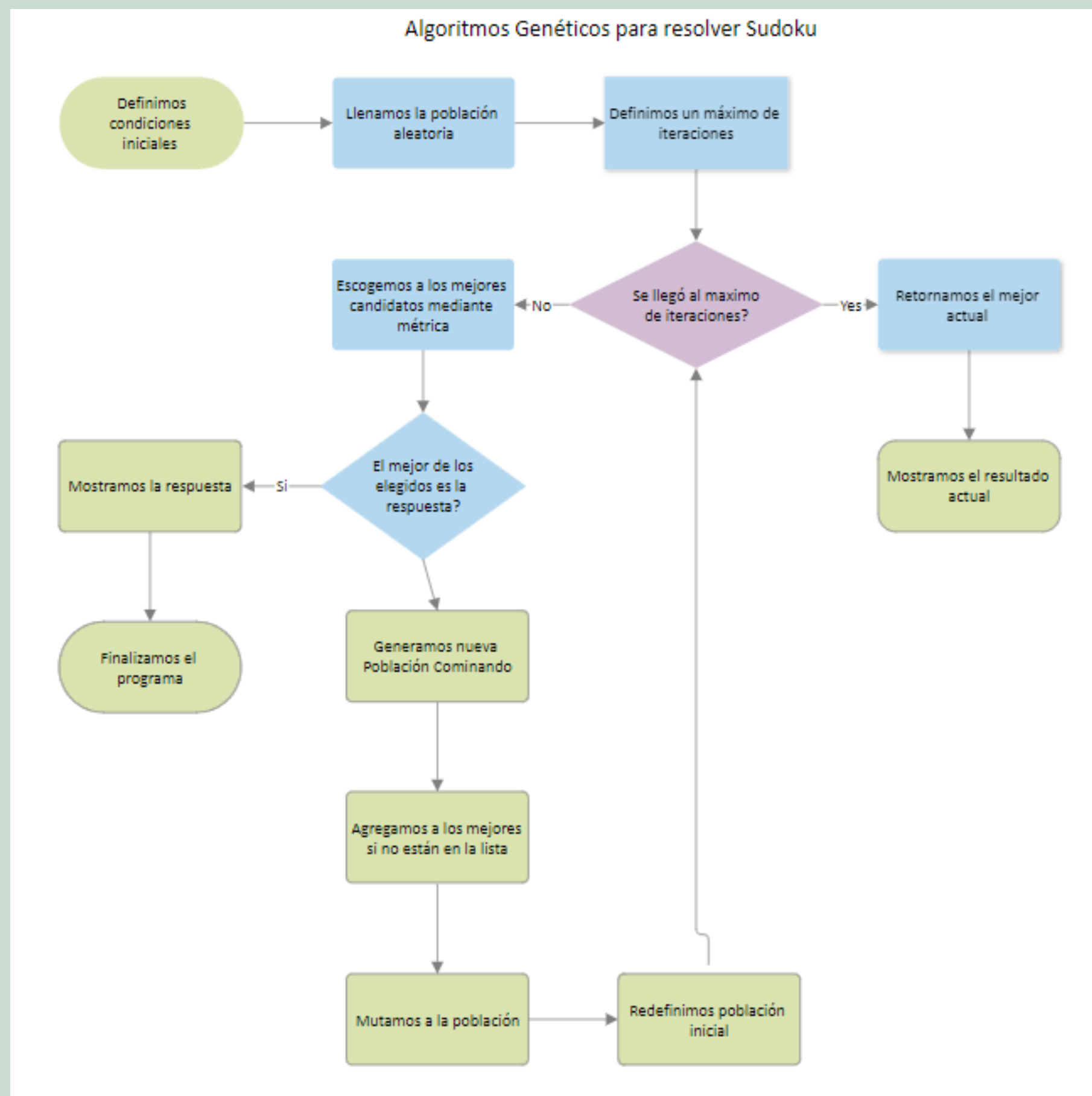
Enfoque:

- Población inicial: Aleatoriamente se resuelven los sudokus, solo respetando una de las 3 reglas nombradas.
- Escoger mejores: Se crea una métrica que cuente los repetidos en las filas y la columna.
- Combinar y generar nueva población: Se reemplaza la población con una población de combinaciones de los individuos que fueron elegidos como los mejores.
- Mutar: Para todos los individuos de la población con cierta probabilidad estos pueden mutar o no, habrán dos formas de mutar.
- Repetir hasta resolver el problema: Se realiza nuevamente el mismo algoritmo reemplazando la población inicial por la nueva población.



Pseudocódigo:

Algoritmo



Métricas

- Métrica 1:

Evalúa y penaliza la cantidad de elementos repetidos en columnas y bloques cuadrados.

Por lo tanto penaliza más fuertemente elementos individuos con mayor número de repeticiones.

- Métrica 2:

Se cuentan si existen elementos repetidos en columnas y bloques, pero el valor que se agrega al resultado es simplemente “1”. Por lo que penaliza igual a los individuos con tal que tengan repetidos.

```
def metrical(sudoku, n):
    t = n**2
    valor = 0
    for i in range(t):
        contador = Counter([sudoku[j][i] for j in range(t)])
        for elemento, cantidad in contador.items():
            if cantidad > 1:
                valor += (cantidad - 1)

    for square in iterar_cuadrados(sudoku, n):
        contador = Counter(square[0] + square[1] + square[2])
        for elemento, cantidad in contador.items():
            if cantidad > 1:
                valor += (cantidad - 1)

    return valor

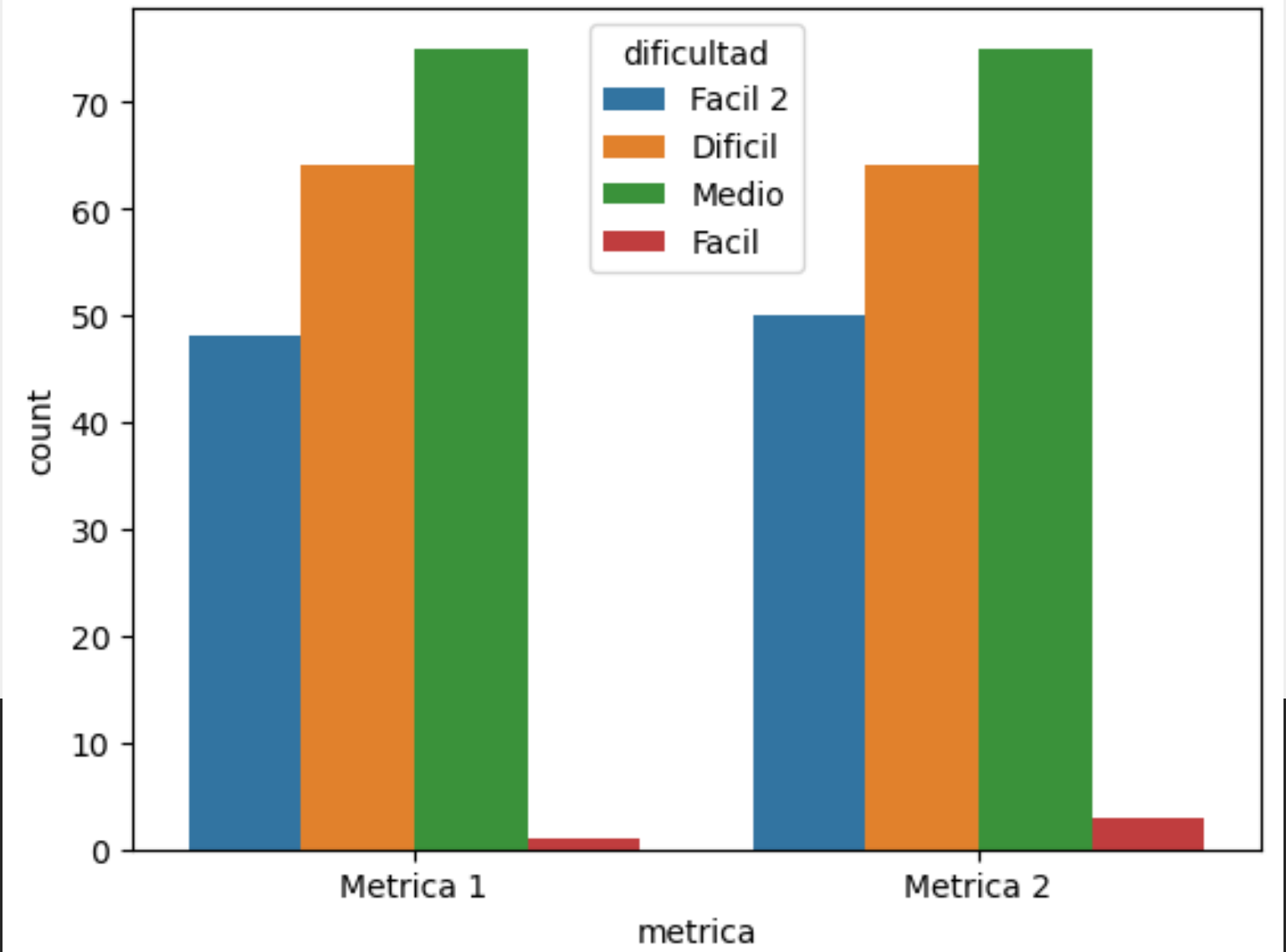
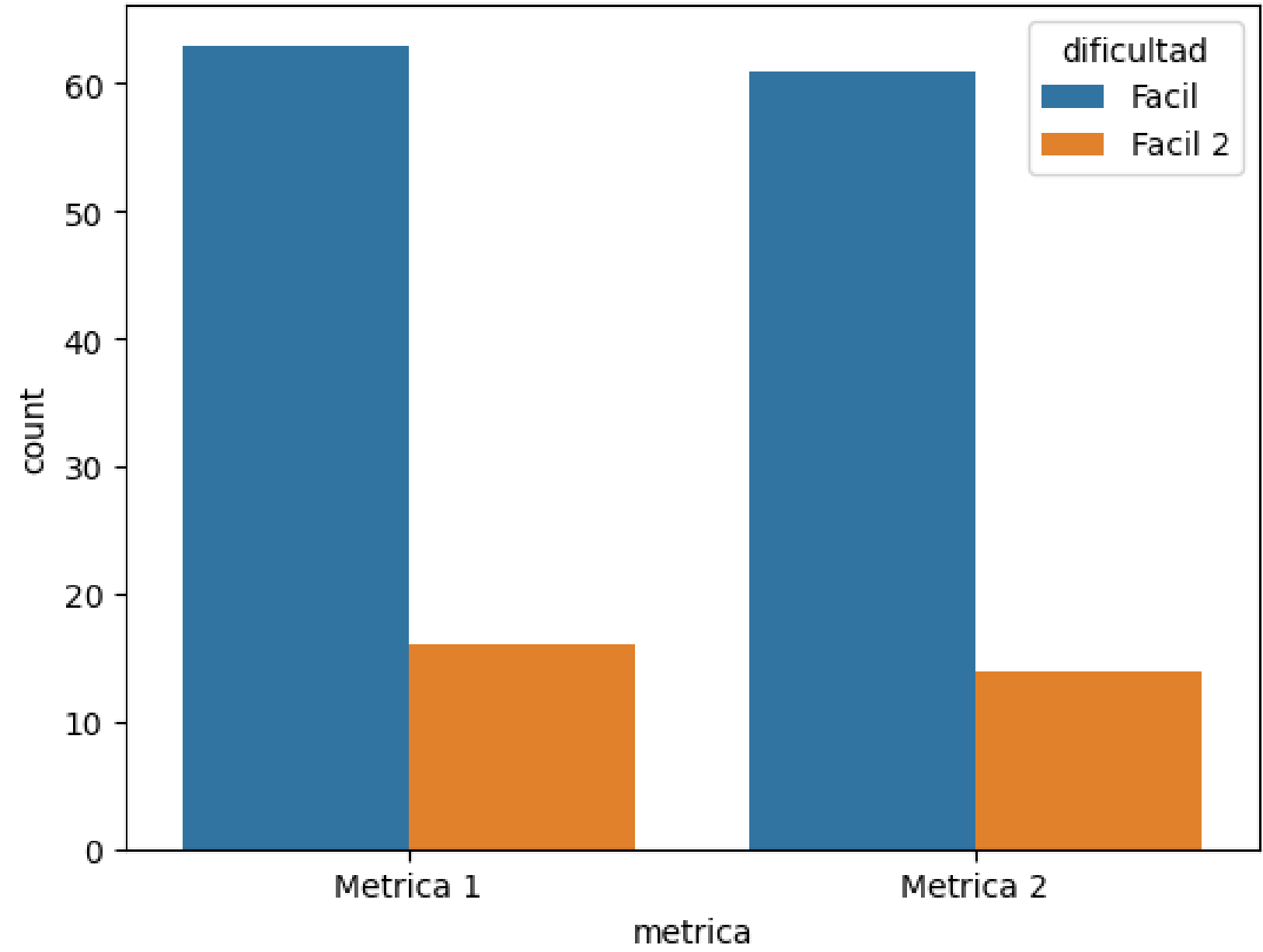
def metrica2(sudoku, n):
    t = n**2
    valor = 0
    for i in range(t):
        contador = Counter([sudoku[j][i] for j in range(t)])
        for elemento, cantidad in contador.items():
            if cantidad > 1:
                valor += 1

    for square in iterar_cuadrados(sudoku, n):
        contador = Counter(square[0] + square[1] + square[2])
        for elemento, cantidad in contador.items():
            if cantidad > 1:
                valor += 1

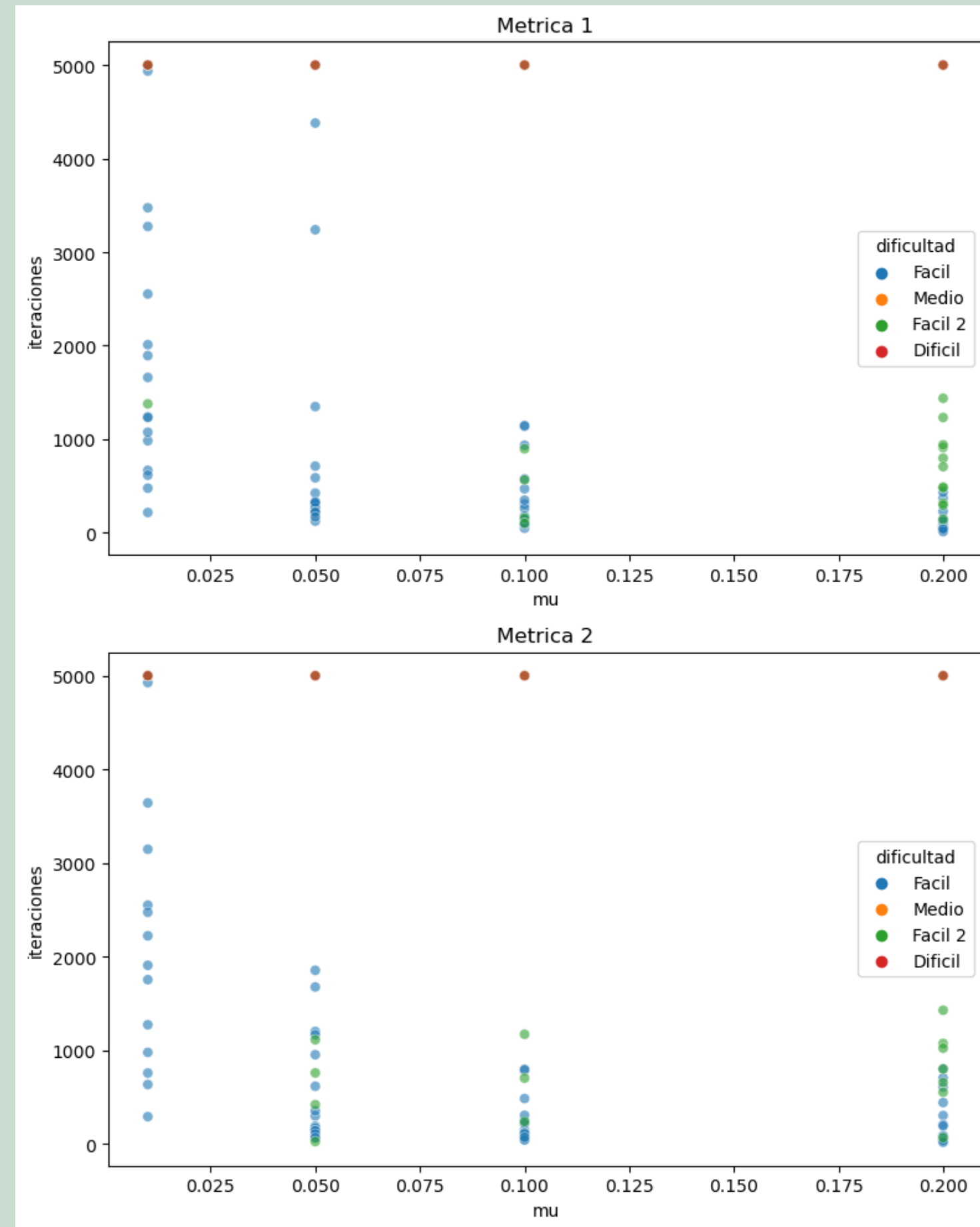
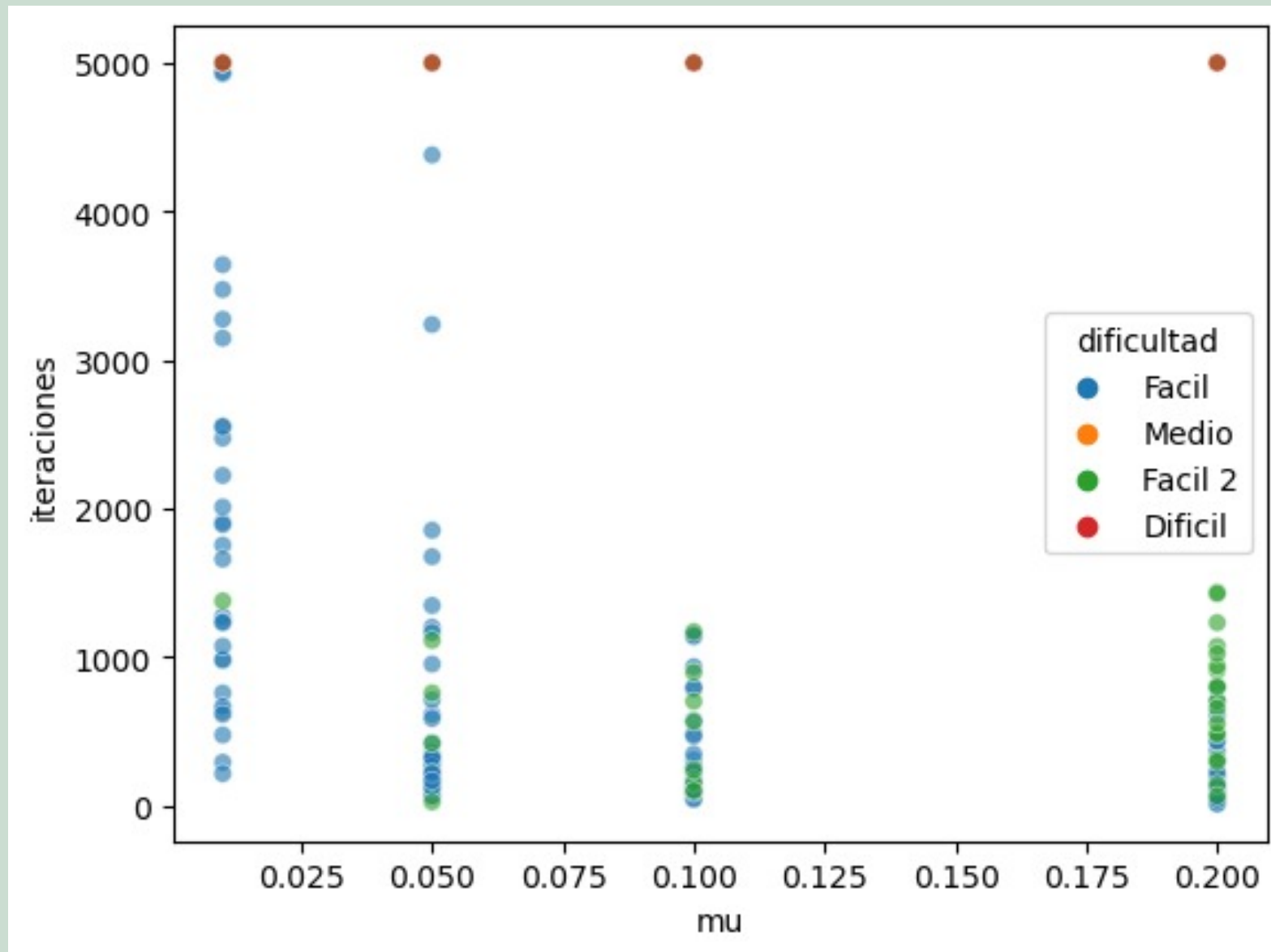
    return valor
```

Código de Métricas

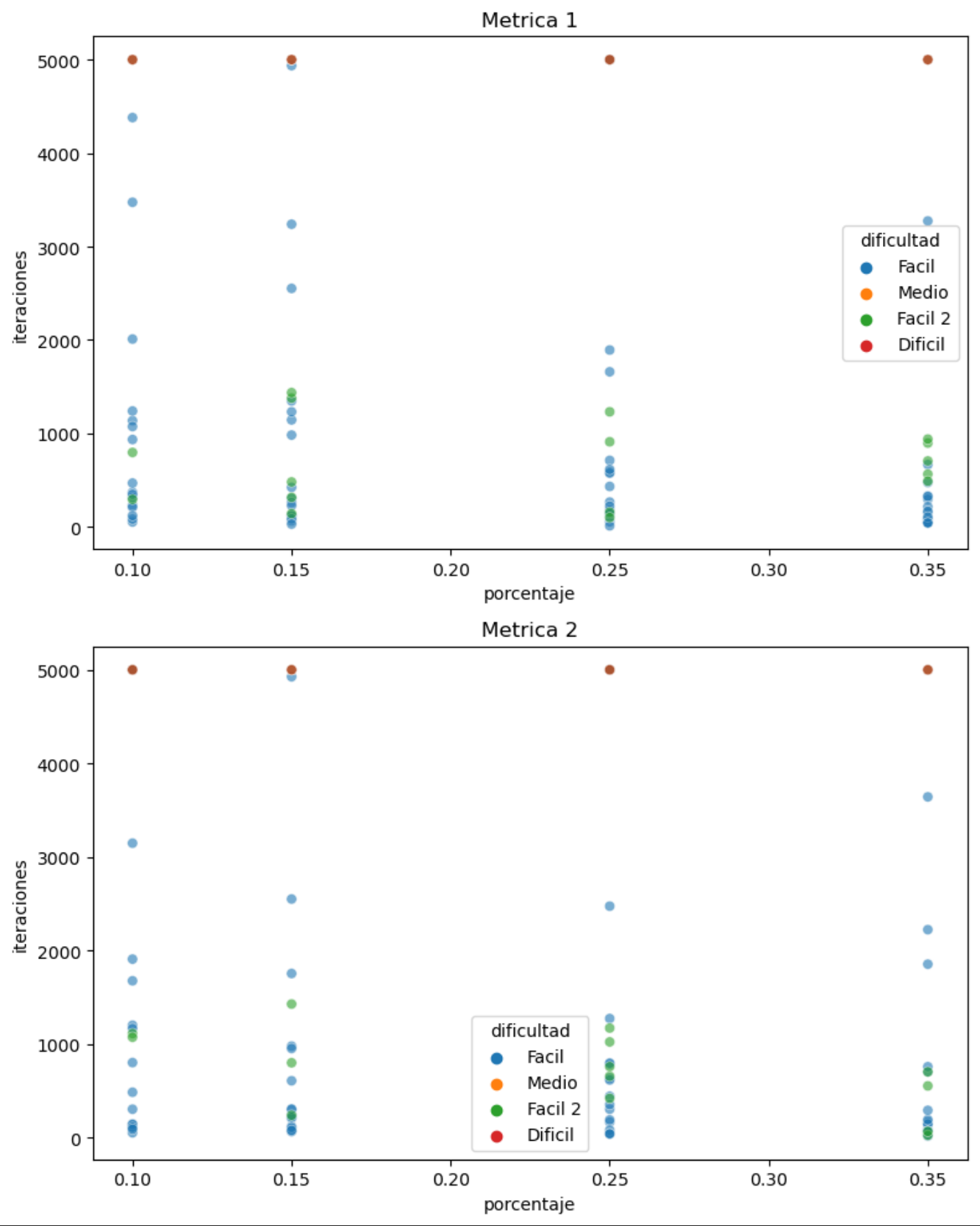
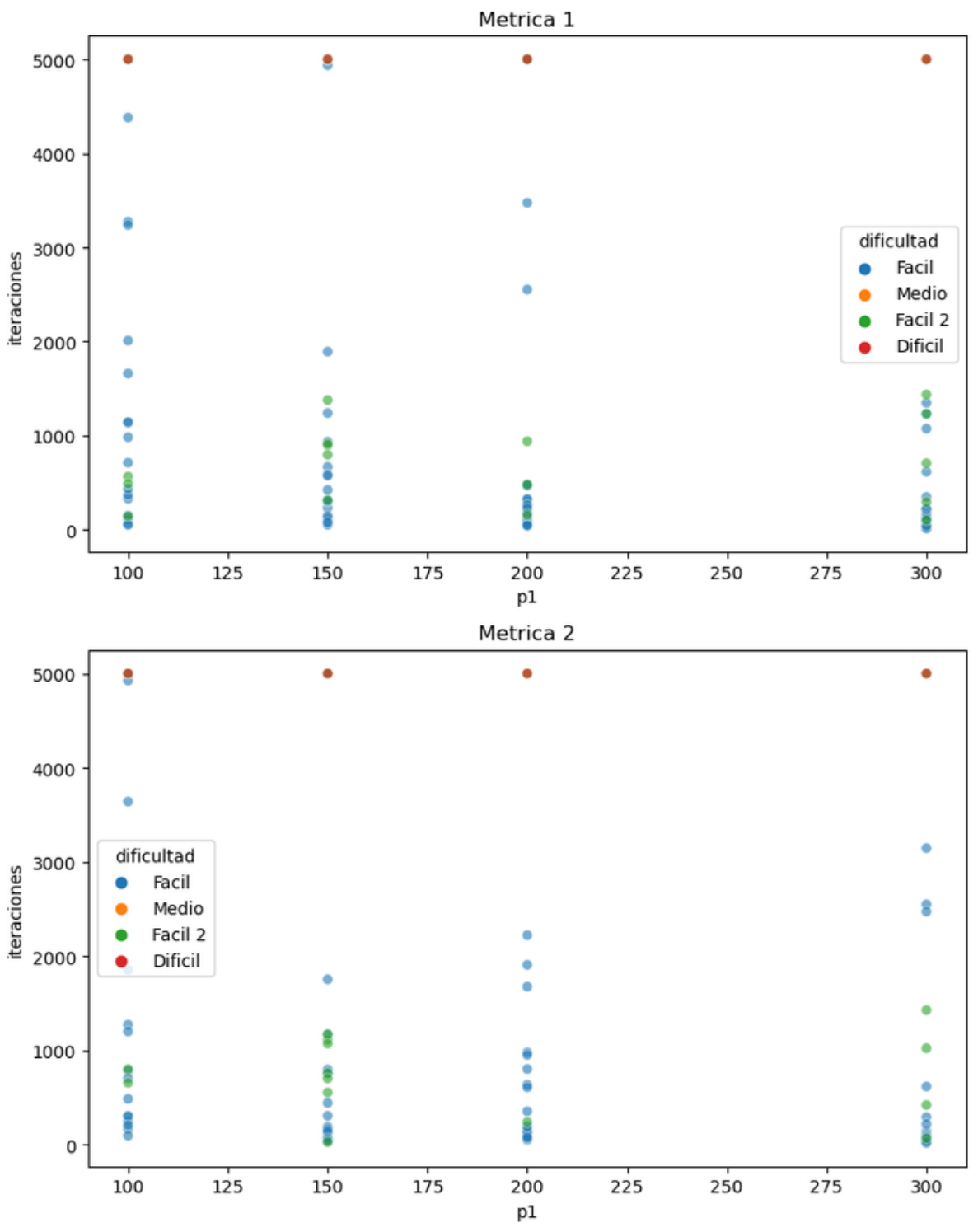
Anual 2024



Resultados



Resultados



Resultados

Se Muestra como se resuelve un sudoku de dificultad Fácil.

Graficando el mejor de cada generación y verificando si este es la solución al problema según la métrica

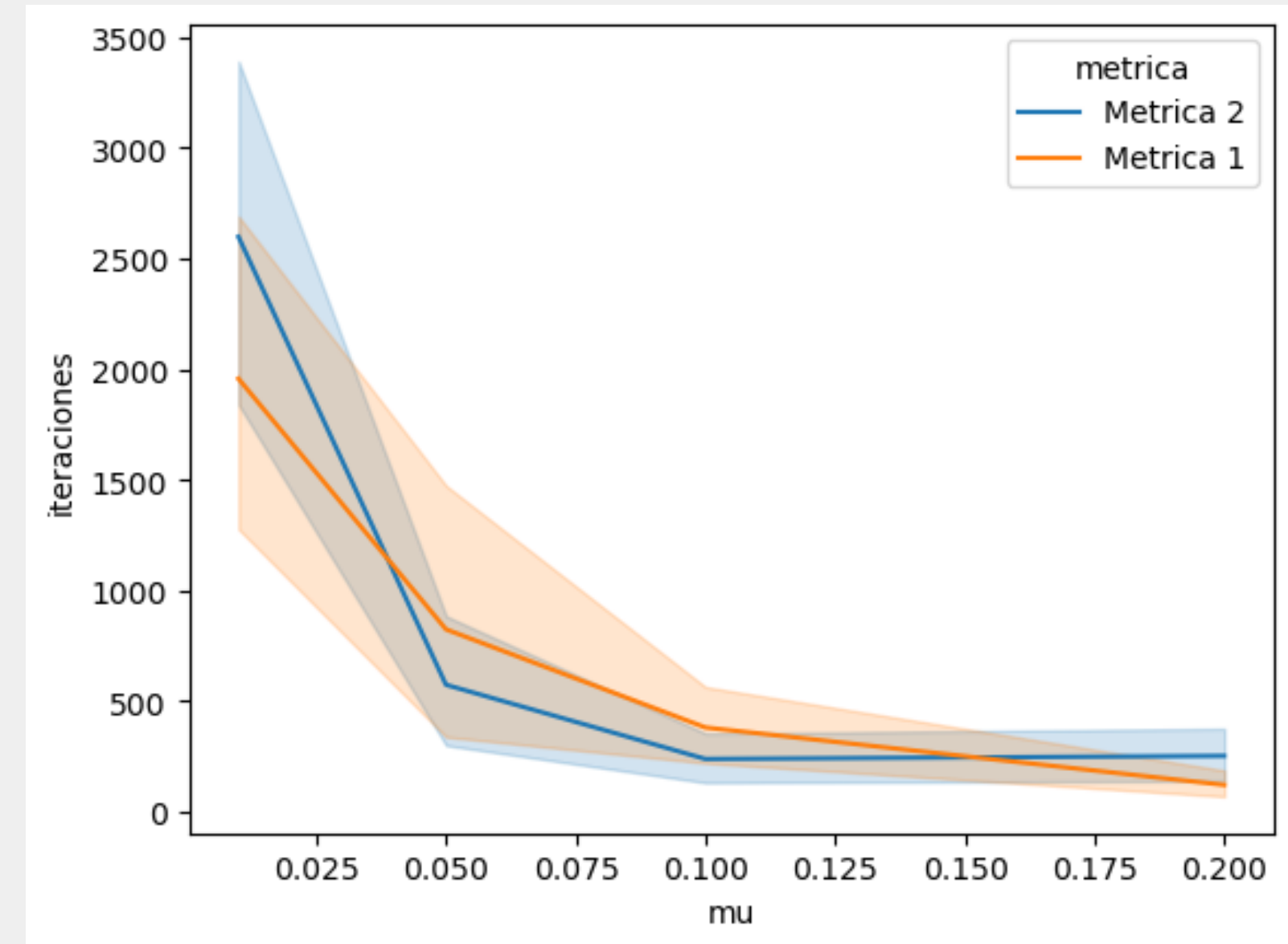
Sudoku por Algoritmos Genéticos

	5	1	8	3	6	2	9	7	4
	7	3	5	8	4	1	9	6	2
	9	2	6	7	5	8	3	1	4
	7	1	2	5	4	8	6	9	3
	3	4	8	6	7	9	5	2	1
	5	6	4	1	2	8	9	3	7
	9	2	1	7	8	5	3	6	4
	6	8	5	2	1	3	7	4	9
	4	3	7	5	9	6	2	8	1

Discusión

En General podemos observar que la métrica 2 da una mayor solidez al algoritmo. Mostrando una mejora más constante, pero significando un costo computacional más elevado.

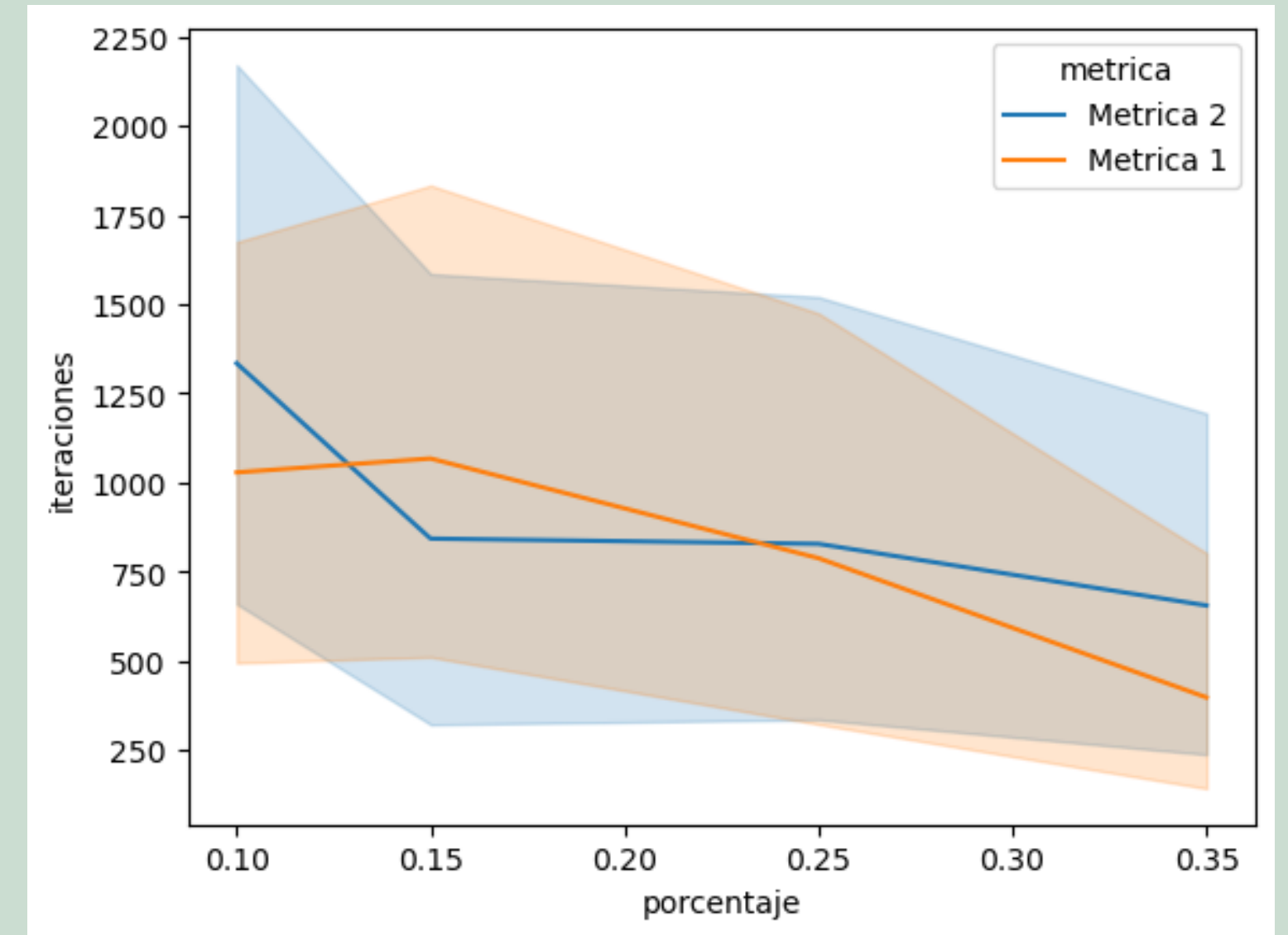
Asimismo, observamos que a medida que se aumenta el Mu, la población 1 y el porcentaje, las iteraciones tienden a bajar.



Podemos observar la eficiencia de la métrica 2
Pero teniendo en cuenta su costo computacional

Discusión

amet volutpat consequat mauris nunc congue nisi vitae
suscipit tellus mauris a diam maecenas sed enim ut
sem viverra aliquet eget sit amet tellus cras adipiscing
enim eu turpis egestas pretium aenean pharetra magna
ac placerat vestibulum lectus mauris ultrices



amet volutpat consequat mauris nunc congue nisi vitae
suscipit tellus mauris a diam maecenas sed enim ut sem
viverra aliquet eget sit amet tellus cras adipiscing enim eu
turpis egestas pretium aenean pharetra magna ac placerat
vestibulum lectus mauris ultrices

Conclusiones del Proyecto

Utilizar parámetros altos para este tipo de problemas aumenta el costo computacional, pero disminuye la cantidad de generaciones para resolver el problema.

La métrica a utilizar si es un factor determinante en la solución del problema.

La dificultad del Sudoku fue un impedimento en cuestión de gasto computacional, ya que se requerían más generaciones que las utilizadas.

	5	1	8	3	6	2	9	7	4
	7	3	5	8	4	1	9	6	2
	9	2	6	7	5	8	3	1	4
	7	1	2	5	4	8	6	9	3
	3	4	8	6	7	9	5	2	1
	5	6	4	1	2	8	9	3	7
	9	2	1	7	8	5	3	6	4
	6	8	5	2	1	3	7	4	9
	4	3	7	5	9	6	2	8	1

Recomendaciones

Para poder realizar esta solución es recomendable utilizar una cantidad de generaciones máxima entre 5000 y 8000. Si se quiere llegar a una solución en niveles más complejos

Es Python es recomendable declarar cada sudoku dentro del ciclo ya que por la funcionalidad de punteros de memoria, este devuelve siempre la misma ubicación

Utilizar una métrica que penalice varios aspectos a tomar en cuenta en la solución.