

# Optimización por Enjambre de Partículas



Morales Zepeda Ivan Yutlanih

Código: 215467762

Ingeniería en computación

El desarrollo de este programa en realidad se basa en el código de la tarea anterior, por lo que el cambio generado simplemente fue el del algoritmo proporcionado en clase. Sin embargo, aquí está la explicación nuevamente: Primero, las funciones a graficar son las mismas que vimos en una de las primeras tareas, por lo tanto, la lógica para representarlas con n cantidad de dimensiones no presentó un mayor reto. Por otra parte, conseguir los promedios fue una tarea que sí se empezó desde cero. Inicialmente, creé un arreglo que almacenaría los mejores almacené los mejores históricos:

```
#Guardar los mejores historicos para promediar y graficar
mejores_historicos.append(abs(self._mejor))
```

Después de las 5 ejecuciones, “mejores\_historicos” es un arreglo con 5 arreglos, cada uno con los mejores históricos de su respectiva ejecución. Posteriormente, evaluamos los promedios de las generaciones correspondientes:

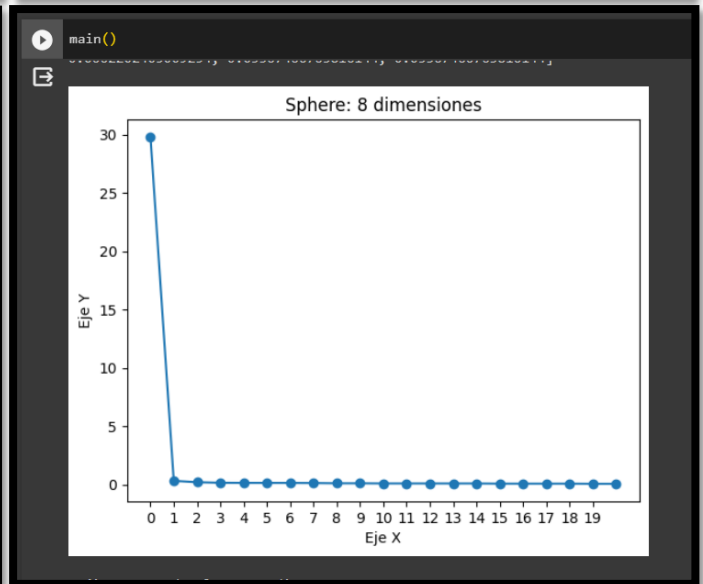
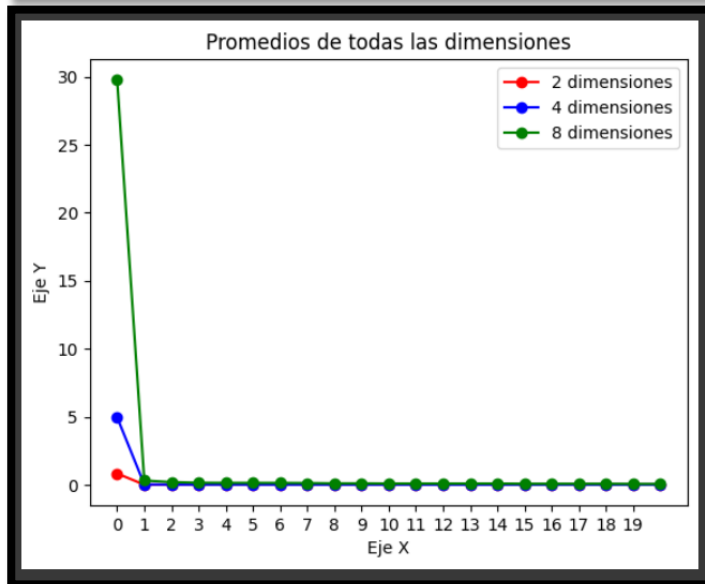
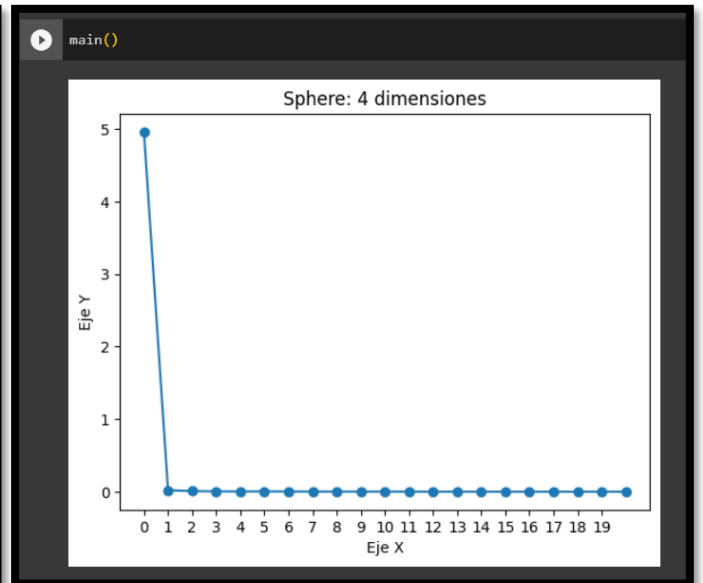
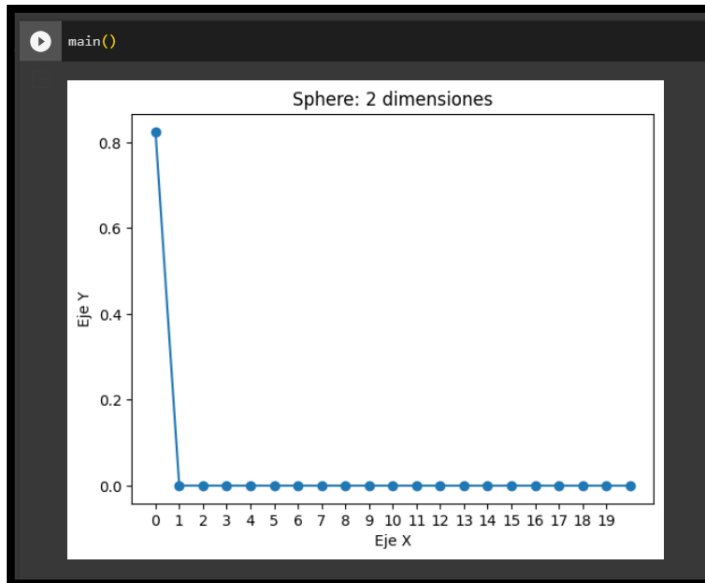
```
#Promediar
mejores_historicosG = [sum(x) / len(x) for x in zip(*mejores_historicosG)]
conjuntoPromedios.append(mejores_historicosG)
```

En esa línea de código, se generan tuplas con los elementos de los diferentes arreglos que se encuentran en la misma posición, se suman y se dividen entre la cantidad total de elementos. De esta forma se consigue el promedio. En la línea siguiente se guardan en un arreglo los promedios para cada cantidad de dimensiones que se graficarán tanto individualmente como en conjunto. Finalmente, se mandan a llamar las funciones para graficar:

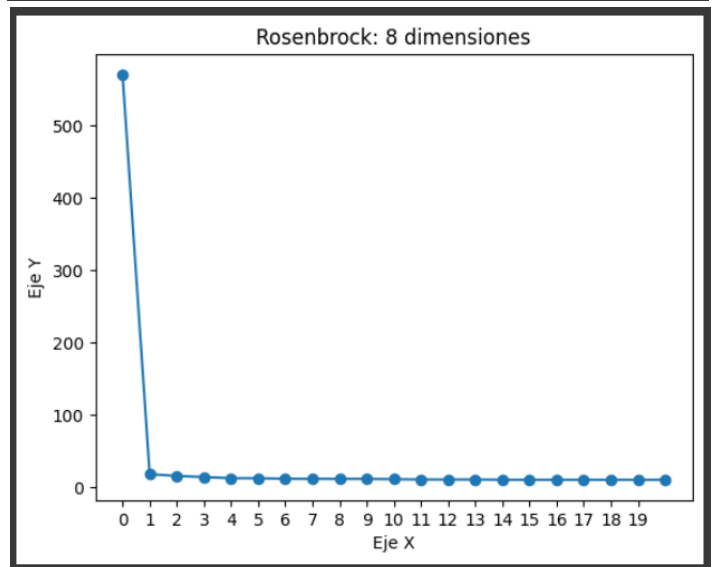
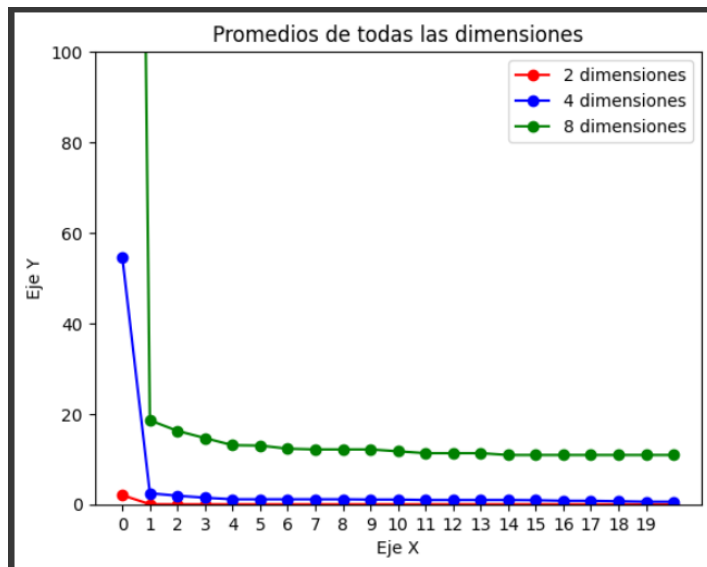
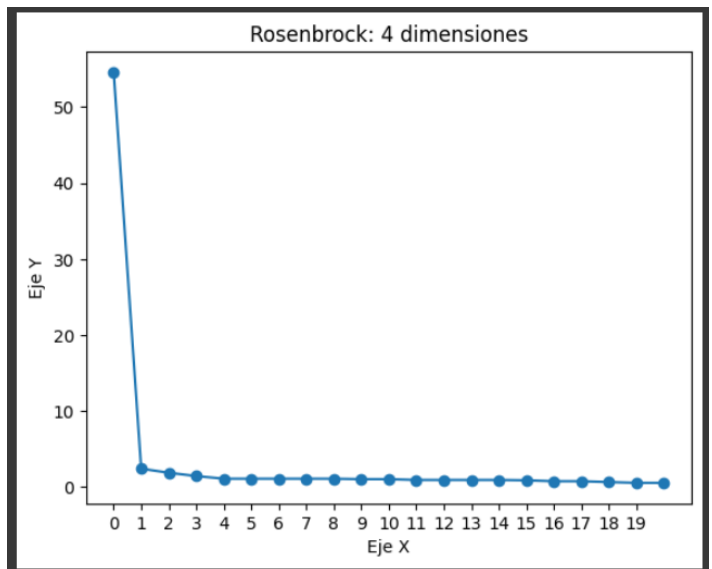
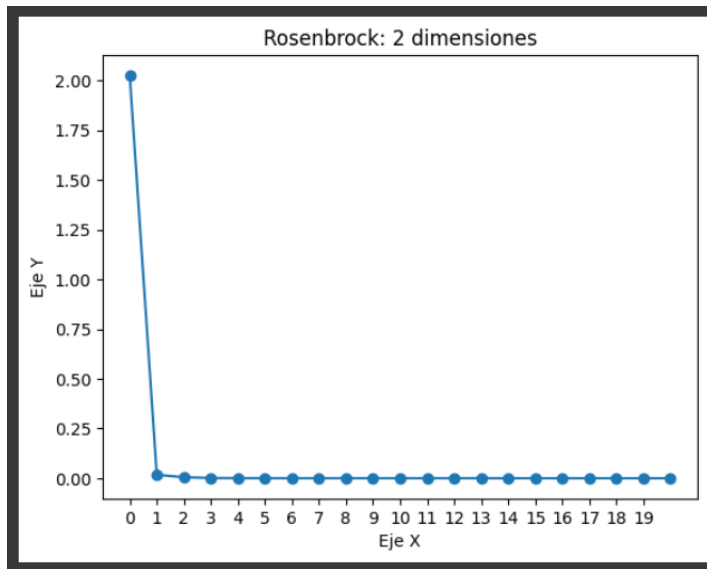
```
#Graficar individualmente
graficarIndividual(mejores_historicosG)
#Graficar promedios en conjunto
graficarConjunto(conjuntoPromedios)
```

## Resultados:

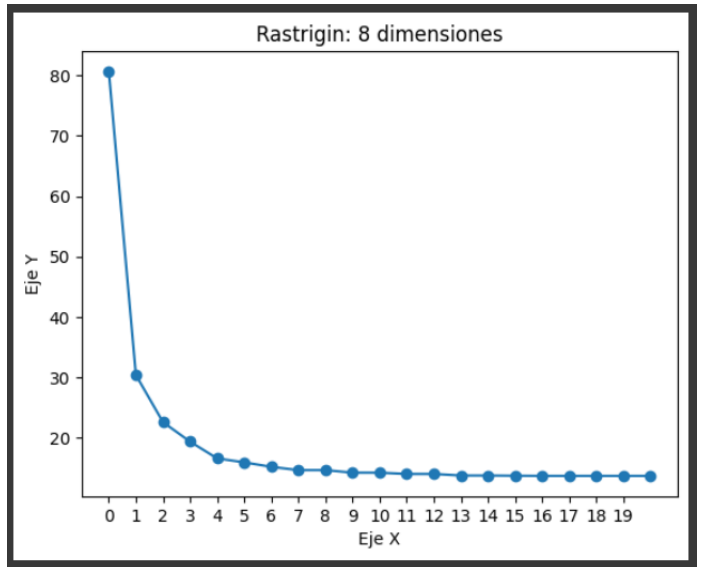
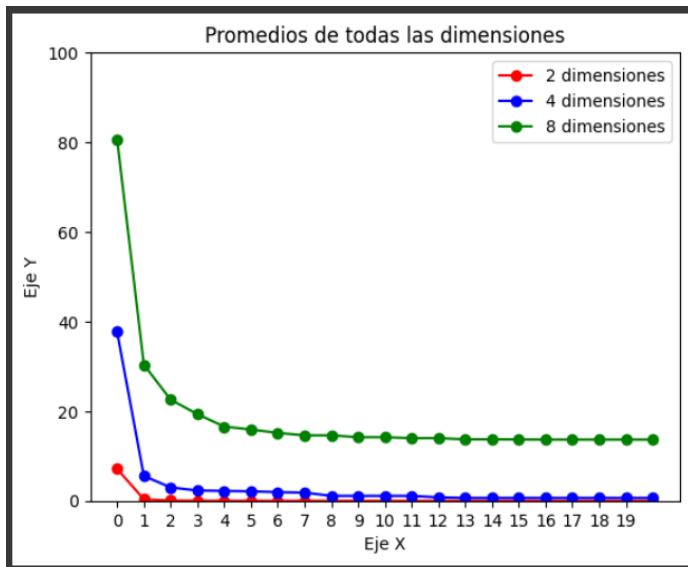
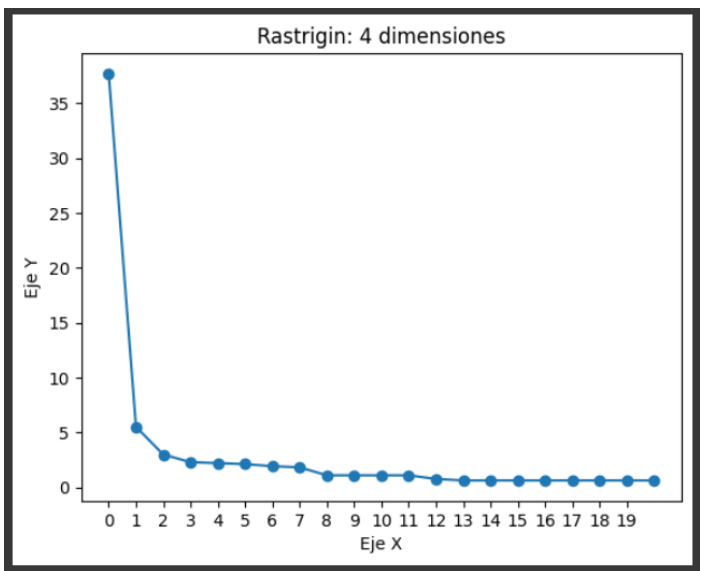
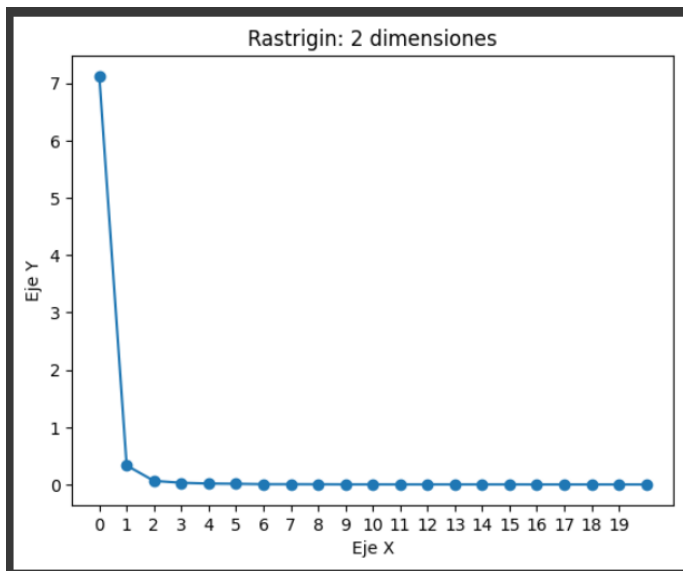
### - Sphere:



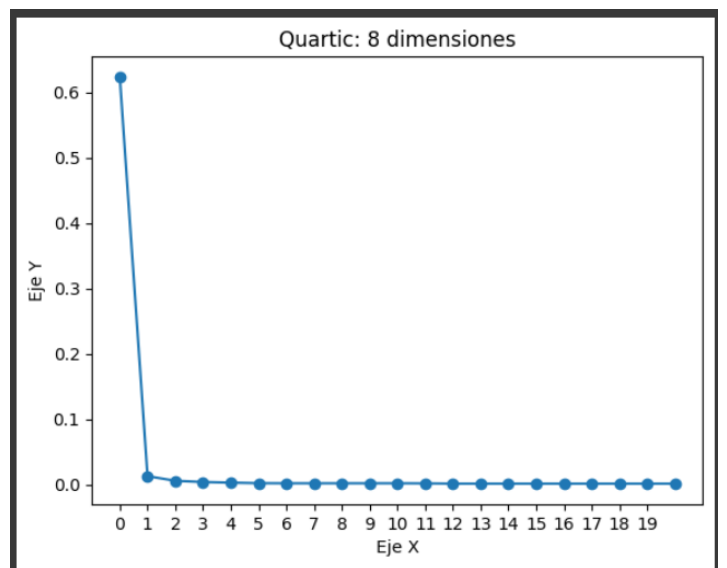
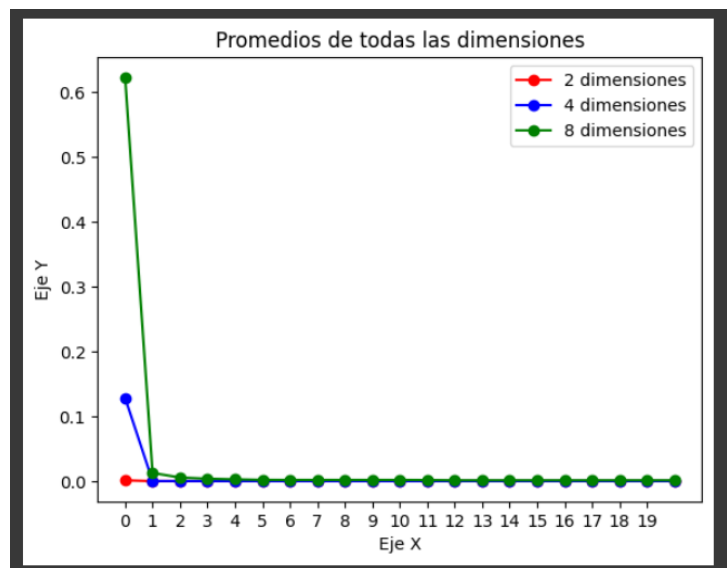
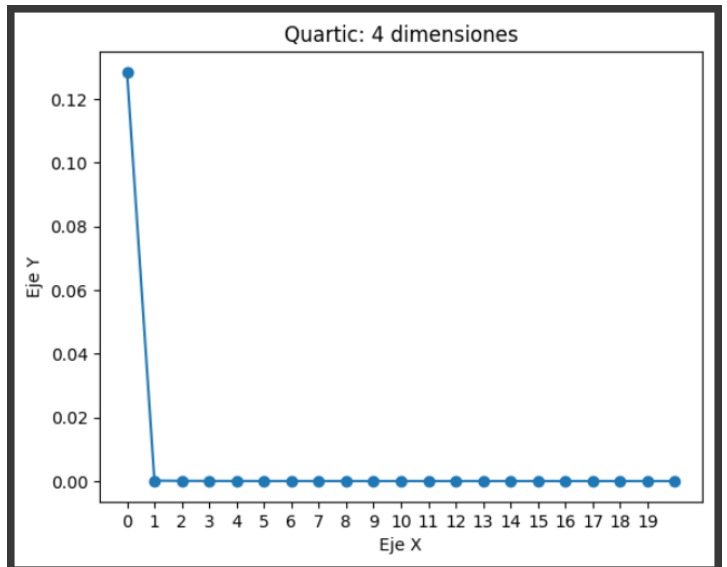
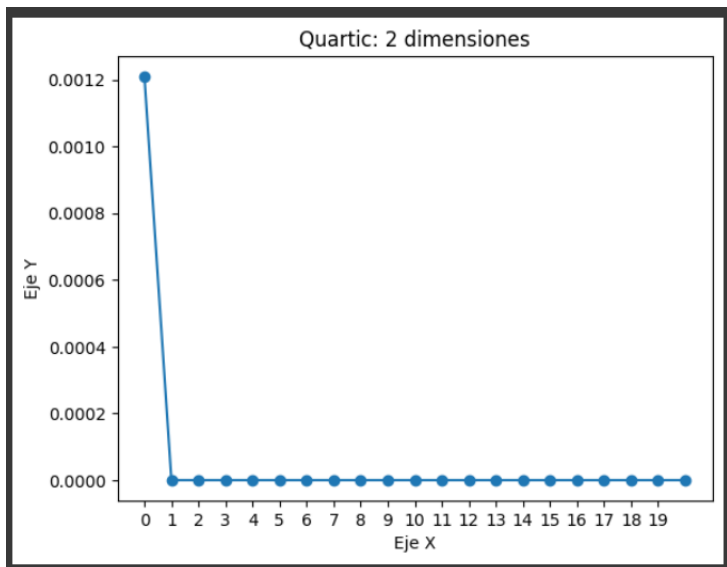
## - Rosenbrock:



## - Rastrigin:



## - Quartic:



El algoritmo de Optimización por Enjambre de Partículas resultó ser bastante bueno, los resultados que entregó, en su gran mayoría se acercaban bastante al valor esperado. En cuanto a la implementación, simplemente se adaptó este nuevo código a los ejemplos y funciones del algoritmo anterior. En conclusión, la Optimización por Enjambre de Partículas ha sido de momento el algoritmo que mejor desempeño ha mostrado.