

# Programación genética

Alejandro Mendoza Puig

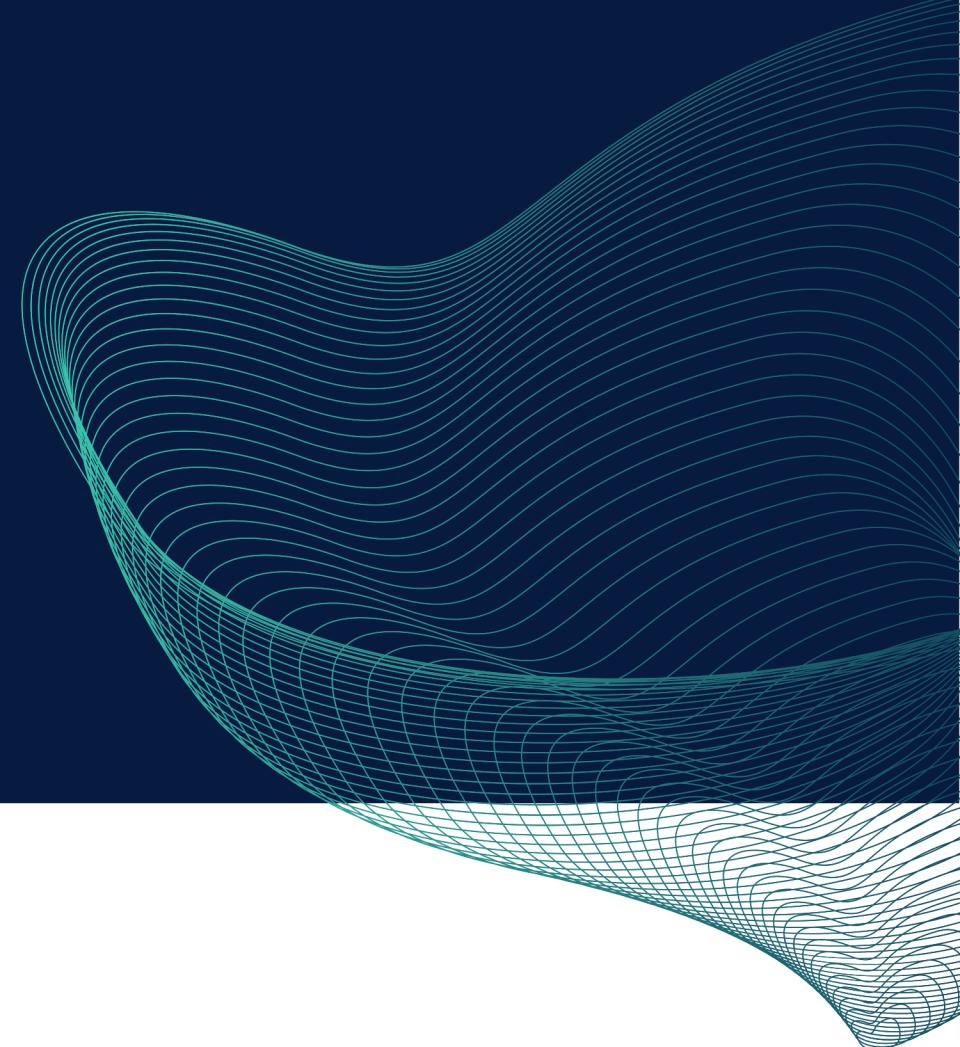
# Introducción

## Programación genética

- Rama de la computación evolutiva que funciona similar a los algoritmos genéticos (**cruza, mutación, selección y reemplazo**)
- Se realizan los mismos operadores que en AG pero a una función la cual esta buscando optimizar su comportamiento. La población pueden ser vistos como programas de computadora

## Problemas

- En esta tarea trabajamos sobre dos problemas, el primero es el de paridad, en el cual dados 3 bits se tiene que generar una respuesta binaria, 1 cuando el número de unos en los 3 bits son pares y 0 de lo contrario.
- El segundo problema es una regresión simbólica en la cual contamos con 21 valores de X junto con el resultado de  $f(x)$ .



# Funciones

Para la implementación se desarrollaron las siguientes funciones:

## Clase Nodo

Clase para cada función/terminal.  
Variables: valor, hijos

## Crear población

Función recursiva para la generación de la población, nodos con aridad=1 solo tienen un hijo

## Cruza

Sección con problemas de recursión. Se seleccionan dos padres, un punto aleatorio en cada uno y se intercambian los “sub árboles”

## Mutación

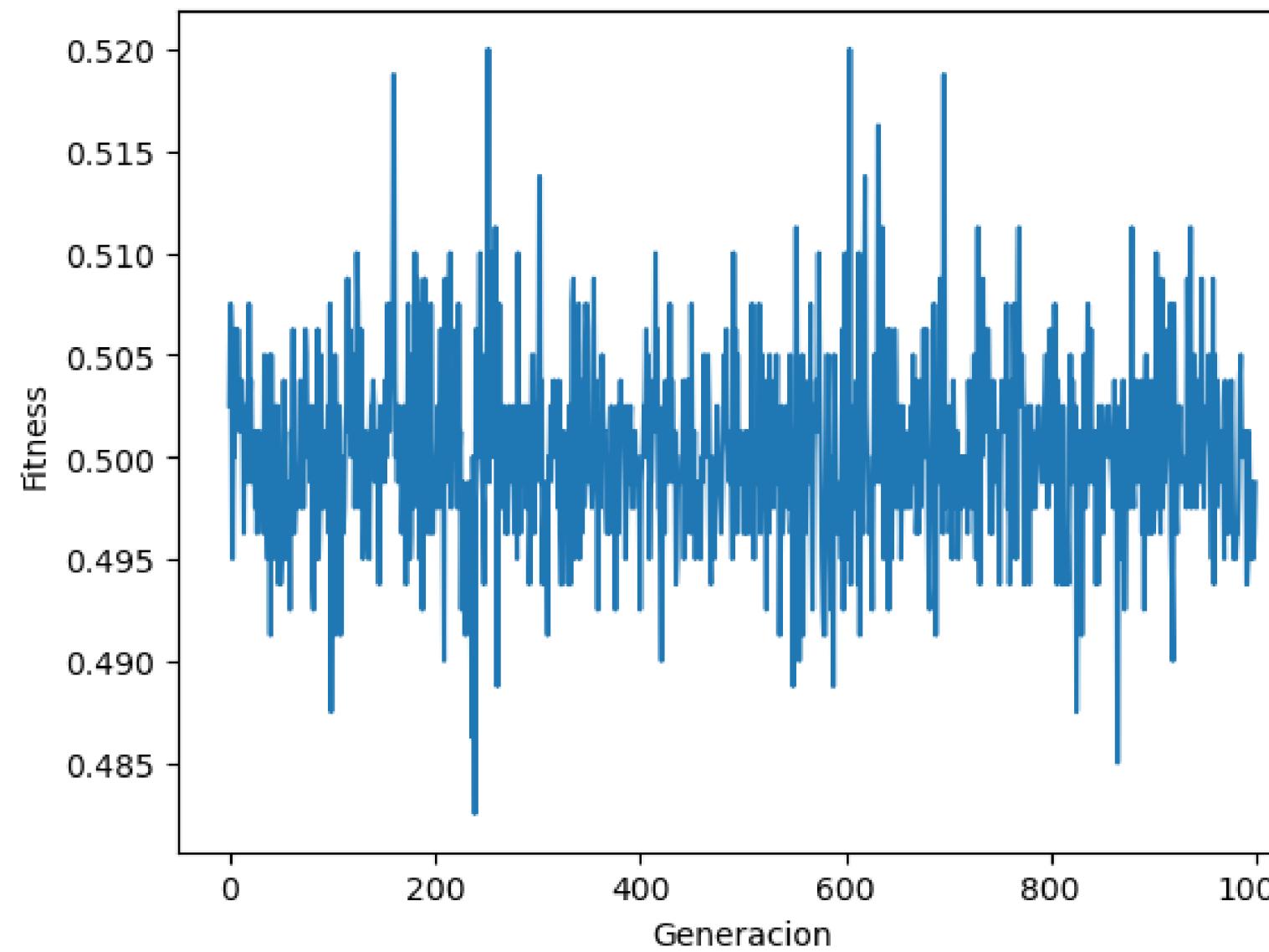
Función que busca un nodo aleatorio en el árbol, genera uno nuevo y lo coloca en el punto aleatorio

## Selección

La implementación fue método de ruleta.

# Resultado

- Los resultados generados muestran un comportamiento aleatorio debido al problema de recursión que he tenido.



# Conclusiones

A pesar de haber tenido problemas de implementación pude fortalecer los conceptos de la programación genética y también pude correr unas pruebas con DEAP una paquetería para algoritmos genéticos.

Estoy iniciando una nueva implementación utilizando la notación polaca para poder realizar la mutación y crusa únicamente recorriendo una lista en vez de un árbol

```
Cell In[5], line 43
 41 nombre_funcion = individuo[0]
 42 if nombre_funcion == 'AND':
...
--> 43     return AND(evaluuar_individuo(individuo[1], entradas), evaluuar_individuo(individuo[2], entradas))
 44 elif nombre_funcion == 'OR':
 45     return OR(evaluuar_individuo(individuo[1], entradas), evaluuar_individuo(individuo[2], entradas))

RecursionError: maximum recursion depth exceeded
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```