

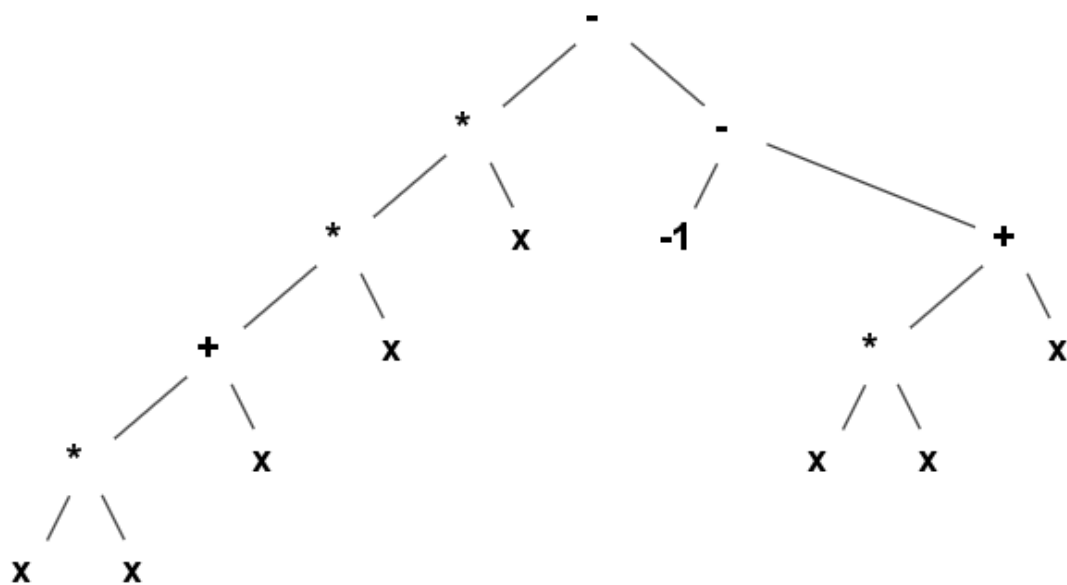
# Programación Evolutiva

## Practica 3

Ignacio Sánchez Santatecla

David Montoro Couso

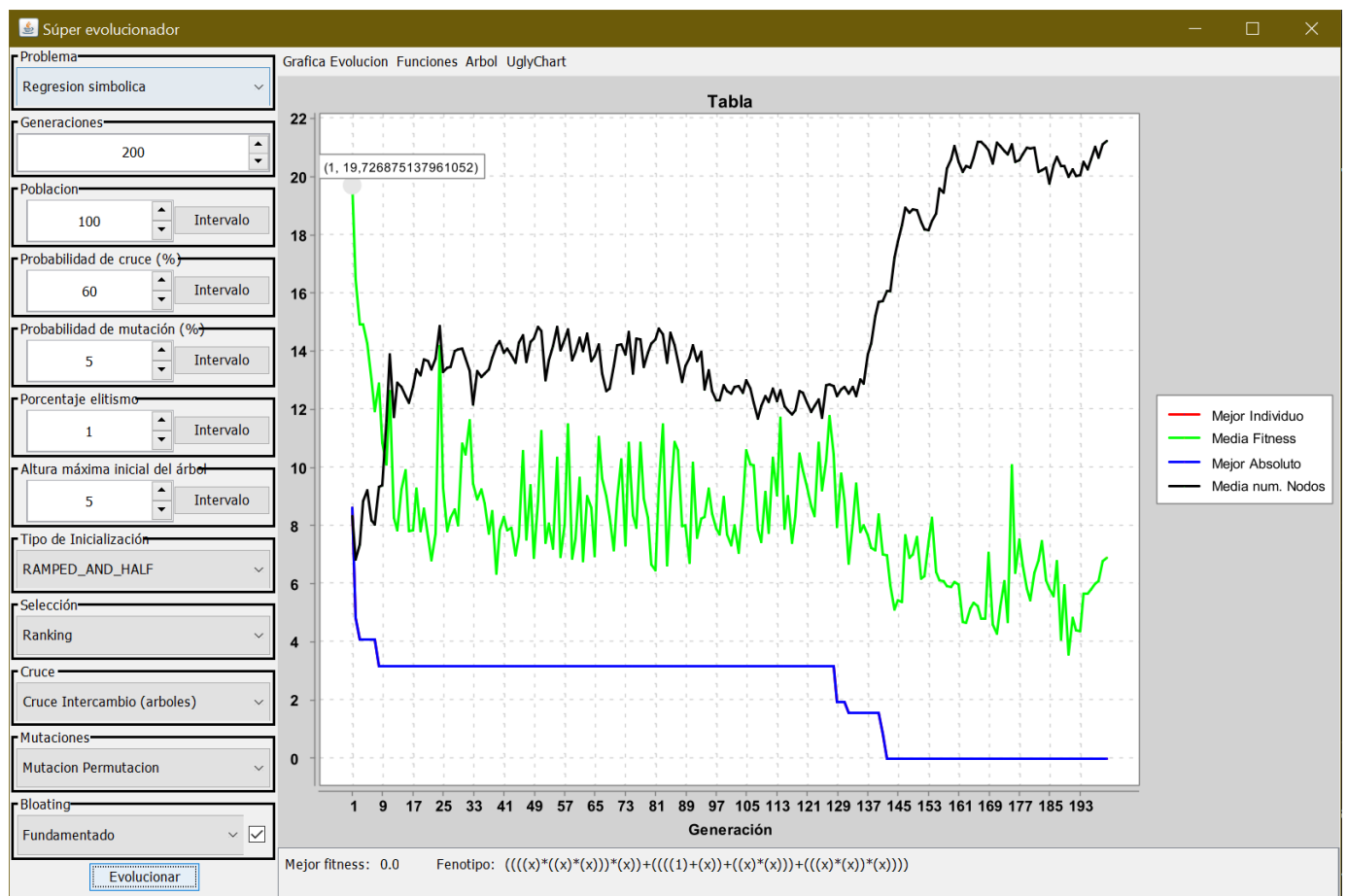
Grupo 7



## Observaciones generales:

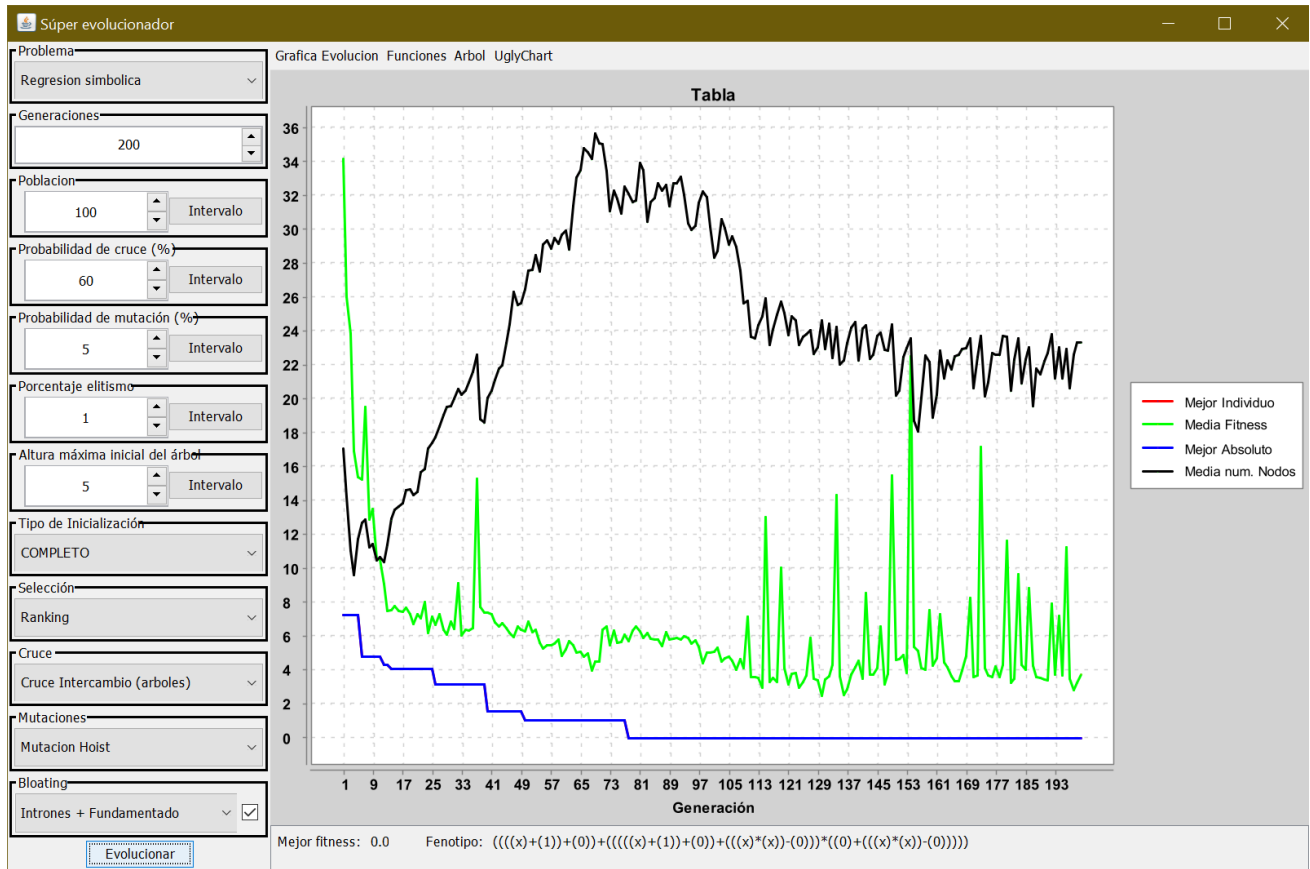
En la mayoría de gráficas que se van a ver a continuación hay un patrón, con la evolución del mejor individuo viene una subida en la media de número de nodos. Esto es debido a que, por lo general, los individuos que suelen alcanzar un fitness de cero están en torno a los 14-24 nodos, algo superior a los que tienen los individuos generados inicialmente. Debido al cruce o a las diversas mutaciones, veremos que los árboles se expanden hasta llegar a este número óptimo de nodos, y a partir de este momento la media de nodos se estabiliza. En casos más específicos que iremos remarcando se observa el efecto del control del bloating, que hace que, aun habiendo un pico en la media de nodos debido a la evolución del mejor individuo, la estabiliza evitando que crezcan desenfrenadamente o, en ocasiones, la hace disminuir.

# Gráficas regresión simbólica:



## Parámetros estáticos

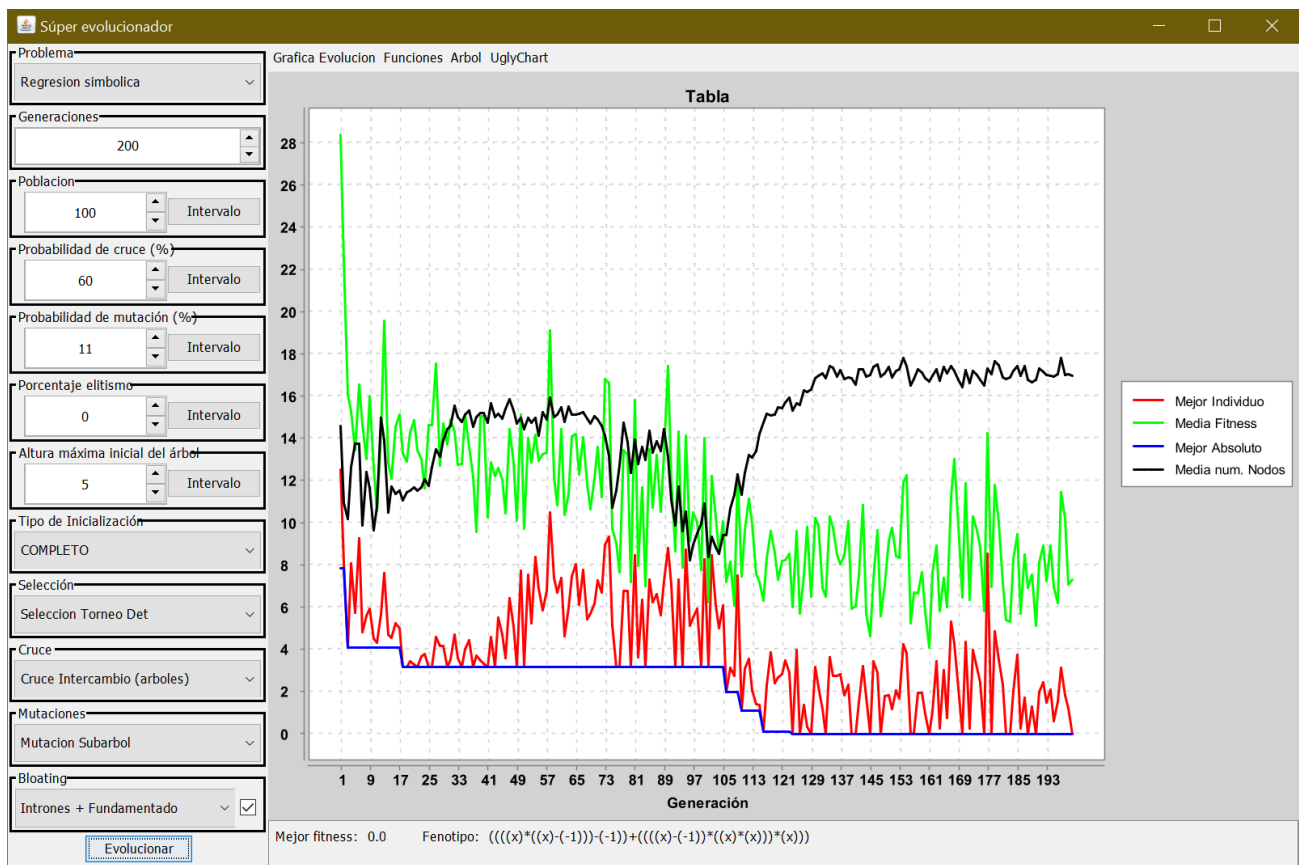
Generaciones:	200
Individuos:	100
Cruce	Intercambio 60%
Mutación:	Permutación (5%)
Elitismo:	1%
Inicialización:	Ramped and Half
Altura máxima:	5
Bloating:	Fundamentado



### Parámetros estáticos

<b>Generaciones:</b>	200
<b>Individuos:</b>	100
<b>Cruce</b>	Intercambio 60%
<b>Mutación:</b>	Hoist (5%)
<b>Elitismo:</b>	1%
<b>Inicialización:</b>	Completo
<b>Altura máxima:</b>	5
<b>Bloating:</b>	Intrones + Fundamentado

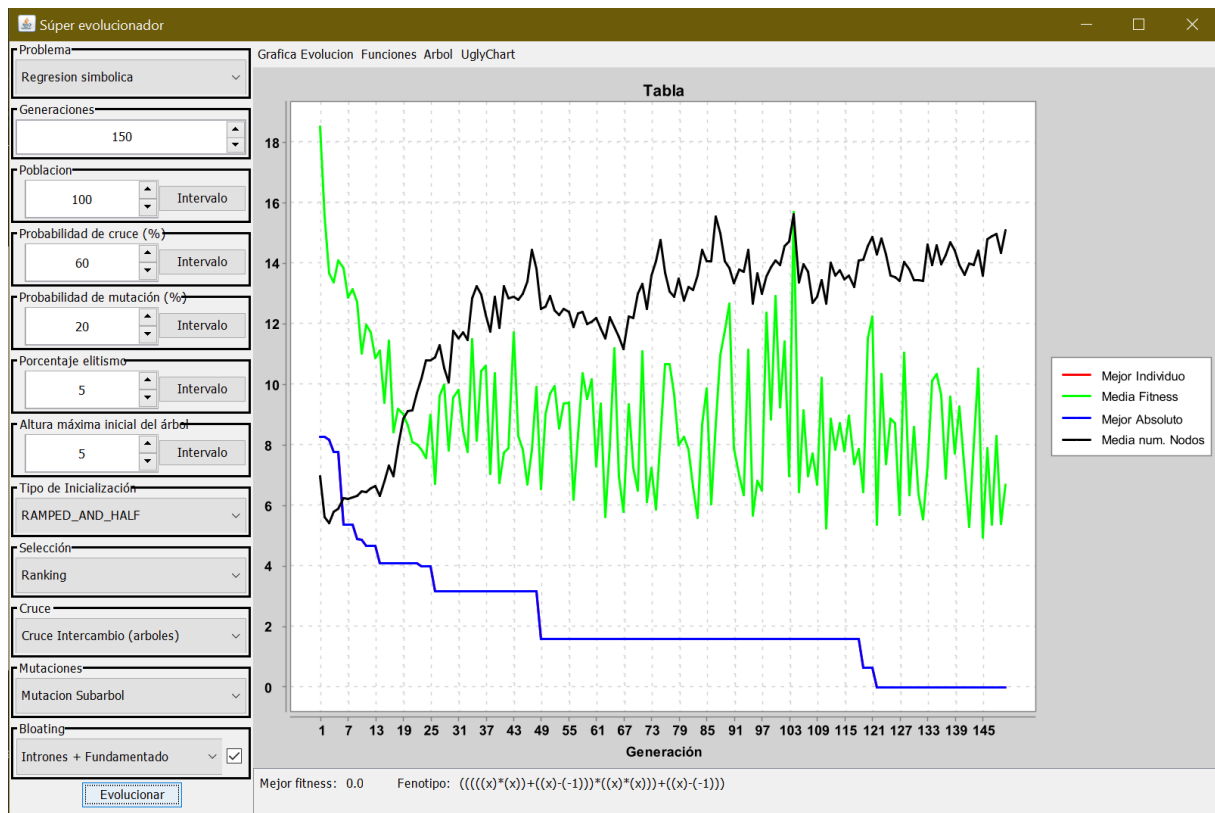
Como decíamos en las observaciones generales, se ve un pico de crecimiento en la media de nodos cuando el mejor individuo mejora, pero unas cuantas generaciones más adelante se estabiliza la media de nodos bajando drásticamente. También se puede apreciar la cantidad de ruido que introduce en la media del fitness la mutación de Hoist.



### Parámetros estáticos

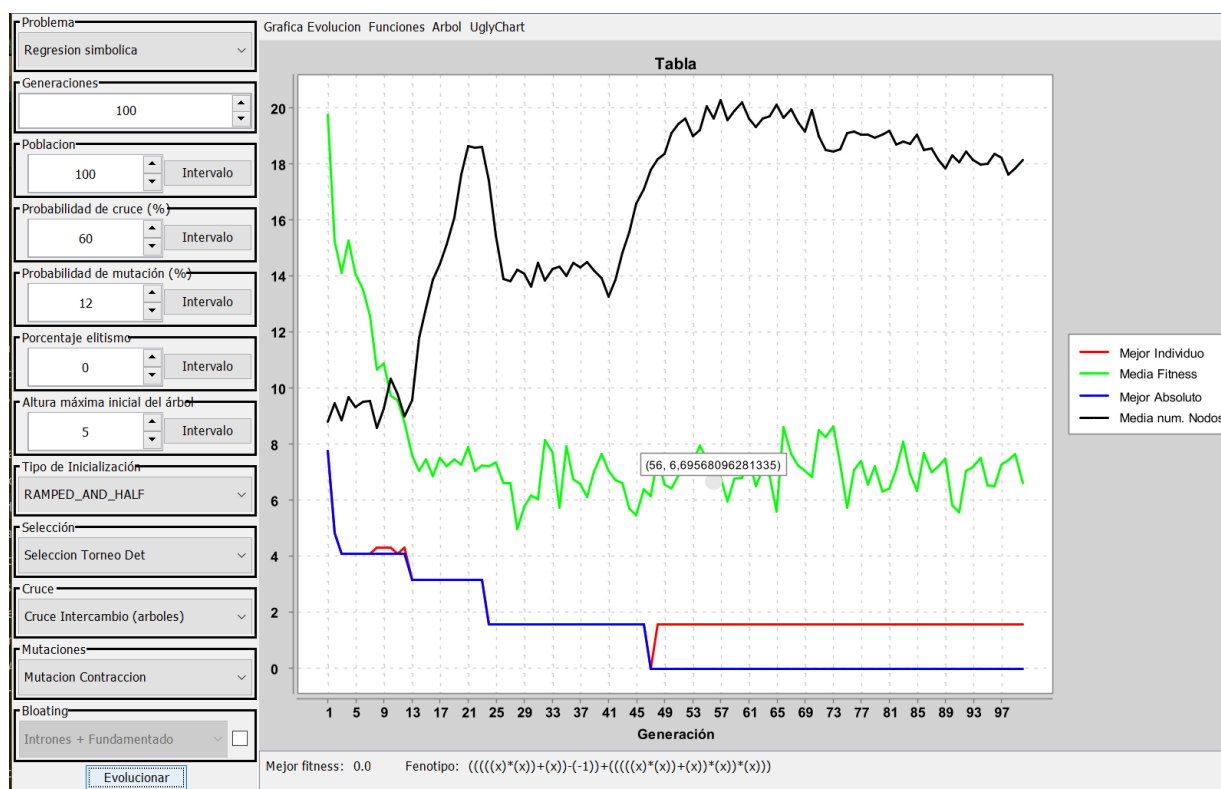
<b>Generaciones:</b>	200
<b>Individuos:</b>	100
<b>Cruce</b>	Intercambio 60%
<b>Mutación:</b>	Permutación (11%)
<b>Elitismo:</b>	0%
<b>Inicialización:</b>	Completo
<b>Altura máxima:</b>	5
<b>Bloating:</b>	Intrones + Fundamentado

Parecido a la grafica anterior, se puede observar el efecto de la mutación subarbol, que al igual que hoist introduce mucho ruido debido a la regeneración de subárboles. Usamos una probabilidad de mutación del 11% para que se generen muchas variaciones y haya una mayor exploración de las posibles soluciones, de esta manera, si asumimos el ruido que va a generar, podemos llegar a la mejor solución sin necesidad de usar elitismo.



## Parámetros estáticos

<b>Generaciones:</b>	150
<b>Individuos:</b>	100
<b>Cruce</b>	Intercambio 60%
<b>Mutación:</b>	Subárbol (20%)
<b>Elitismo:</b>	5%
<b>Inicialización:</b>	Ramped and Half
<b>Altura máxima:</b>	5
<b>Bloating:</b>	Intrones + Fundamentado



## Parámetros estáticos

**Generaciones:** 100

**Individuos:** 100

**Cruce** Intercambio 60%

**Mutación:** Contracción (12%)

**Elitismo:** 0%

**Inicialización:** Ramped and Half

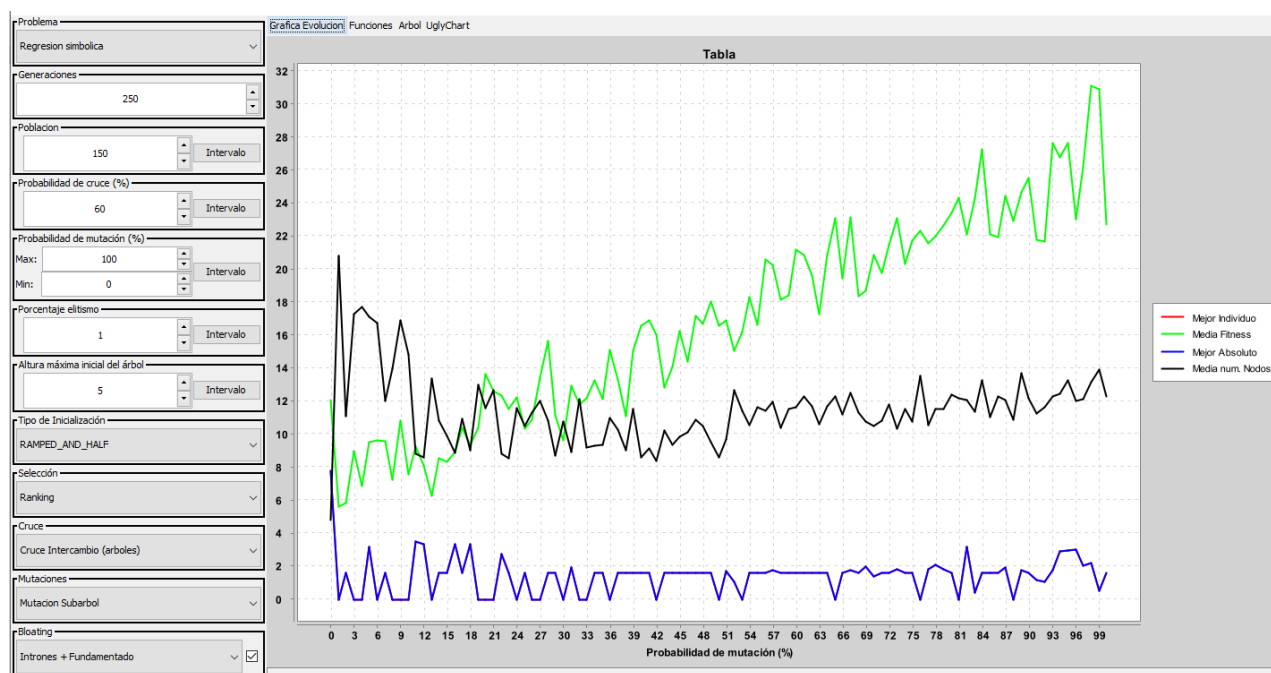
**Altura máxima:** 5

**Bloating:** NO

En esta grafica podemos observar dos fenómenos. Uno es el crecimiento de la media de nodos debido a la disminución del fitness, y otro, la mutación de contracción combinada con el cruce por intercambio. Se puede ver que el primer tramo de crecimiento en la media de nodos la provoca el cruce, generando individuos más expandidos, hasta que uno de ellos muta de la manera correcta, contrayendo su expresión y a la vez aproximándose a la función objetivo. Lo que provoca más copias de este individuo y por tanto, una reducción en la media de nodos. Lo mismo ocurre una segunda vez, acertando esta vez con la función objetivo exacta y estabilizándose entonces la media del número de nodos.







## Parámetros dinámicos

**Generaciones:** 250

**Individuos:** 150

**Cruce** Intercambio 60%

**Mutación:** Subárbol (12%)

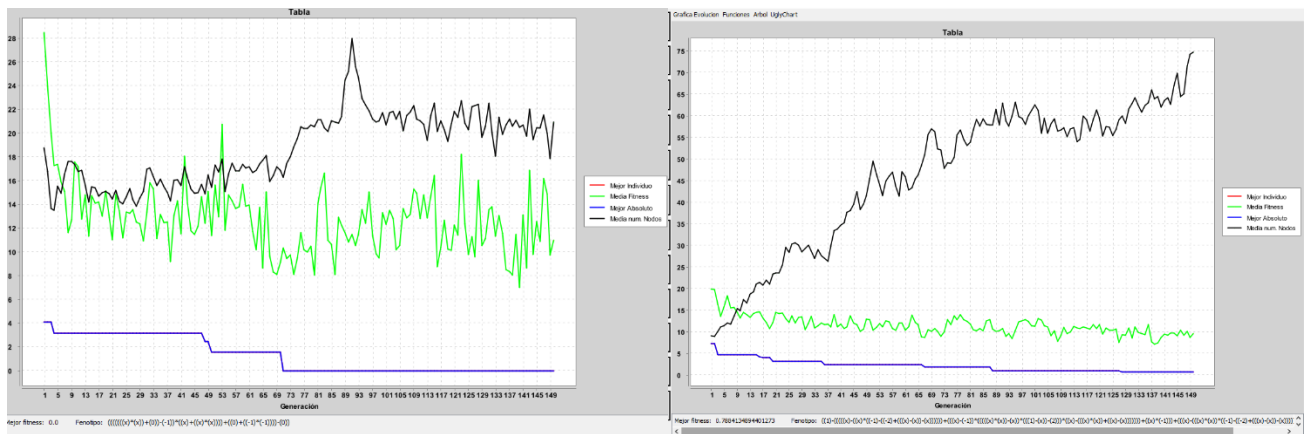
**Elitismo:** 1%

**Inicialización:** Ramped and Half

**Altura máxima:** 5

**Bloating:** Intrones + Fundamentado

Esta gráfica, comparada con la anterior, puede verse como a medida que aumenta la probabilidad de mutación, en el caso de expansión el número de nodos aumenta con la media del fitness, mientras que, en subárbol, aunque también introduce el mismo ruido al fitness, el número de nodos se mantiene estable.



## Parámetros estáticos

<b>Generaciones:</b>	150
<b>Individuos:</b>	100
<b>Cruce</b>	Intercambio 60%
<b>Mutación:</b>	Expansión (12%)
<b>Elitismo:</b>	1%
<b>Inicialización:</b>	Ramped and Half
<b>Altura máxima:</b>	5
<b>Bloating:</b>	Izq: Intrones + Fundamentado Der: NO

Aquí podemos ver nuestra combinación favorita (bueno, la de David), al igual que con la mutación subárbol, la de expansión nos permite explorar mucho el espacio de soluciones y a su vez, el control de bloating y de intrones impide que el número de nodos crezca descontroladamente, manteniéndose en un numero de nodos razonables hasta que encuentra la mejor solución, aumenta al encontrar la mejor solución, y luego vuelve a estabilizar el número de nodos.

En la segunda, se observa que sin el control del bloating, una vez que nuestro mejor individuo llega a un numero de nodos insano, es muy difícil llegar a la mejor solución, pues simplemente vamos a ir añadiendo nodos a una expresión sin sentido alejada de lo que buscamos.

## Detalles de la implementación (Regresión simbólica)

Hemos decidido modificar el bloating fundamentado para implementar el nuestro propio. Observando el comportamiento del algoritmo nos dimos cuenta de que la solución tenía un tamaño superior a la media de la población inicial, por tanto un control del bloating muy agresivo desde el principio impediría que se encontrase la mejor solución (aunque esto se pudiese contrarrestar con alguna mutación que ampliase el árbol como la de expansión), por esto, modificamos la penalización del bloating fundamentado para que dependa de la generación en la que estamos, siendo una mayor a medida que avanzan las generaciones.

Para nuestra propia técnica de control de bloating, hemos decidido implementar una función que “colapsa” el árbol y se deshace de algunos intrones. No reduce su expresión hasta la raíz, ni introduce nuevos terminales (solo actúa si el resultado del subárbol es uno de los iniciales: -2, -1, 0, 1, 2). Esto en sí no es una gran técnica de control del bloating, pues puede seguir produciendo árboles muy grandes si estos no tienen información redundante, por eso creemos que lo mejor es juntarla con alguna otra técnica como la bien fundamentada.

Una de las primeras cosas que vimos al probar nuestro algoritmo es la cantidad innecesaria de ruido que aportaban los ceros, pues la mayoría de veces alargaban el árbol sin aportar valor sintáctico a la expresión, por ello decidimos quitarlos de los valores posibles en la inicialización.

# Gráficas gramáticas evolutiva



## Parámetros estáticos

Generaciones:	250
Individuos:	150
Cruce	Monopunto 60%
Mutación:	Propia (12%)
Elitismo:	1%
Selección:	Truncamiento



## Parámetros estáticos

<b>Generaciones:</b>	150
<b>Individuos:</b>	100
<b>Cruce</b>	Uniforme 60%
<b>Mutación:</b>	Básica (12%)
<b>Elitismo:</b>	1%
<b>Selección:</b>	Ranking

# Detalles de la implementación (Gramáticas evolutivas)

Sin duda el mayor reto de esta práctica ha sido afinar con la gramática y con los valores de tamaño de cromosoma y wrapping adecuados. Desde el primer momento vimos un gran problema, los bucles. Nuestros individuos entraban en bucles infinitos de mientras desarrollaban las reglas de la gramática muy frecuentemente. También nos dimos cuenta que si usábamos un tamaño de cromosoma impar se decrecía el número de bucles, aunque muy a nuestro pesar, seguían produciéndose desarrollos repetitivos dentro de cada subexpresión.

Hemos probado a poner más reglas, menos, introducir signos para modificar las expresiones y las variables, y al final hemos sido incapaces de dar con una gramática que no genere ningún bucle. Para castigar los “Individuos” erróneos, le ponemos un fitness muy alto y confiamos en el trabajo de la selección.

La gramática final es:

$I \rightarrow \text{Exp}$

$\text{Exp} \rightarrow \text{Exp Op Exp} \mid \text{Var} \mid \text{Sign Exp}$

$\text{Op} \rightarrow + \mid - \mid *$

$\text{Sign} \rightarrow + \mid -$

$\text{Var} \rightarrow x \mid 1 \mid 2$

Cabe destacar la regla  $\langle I \rangle$  que hemos introducido para que nuestra expresión nunca sea un literal y siempre contenga al menos una operación.

Ya que la decodificación del individuo al final dependía del módulo del codón, pensamos que la mutación básica no era muy adecuada, puesto que, aunque escoja un número aleatorio entre todos los posibles, si este número tiene el mismo módulo, a efectos prácticos es como si no hubiese mutado, por tanto, hicimos nuestra propia mutación que suma o resta 1 al codón para asegurarnos de que se modifica su módulo.

## Reparto de Tareas

Al igual que en las entregas anteriores hemos trabajado codo con codo como buenos camaradas. Como ya hicimos los cambios a la GUI en la entrega anterior para esta solo teníamos que centrarnos en el problema en sí y sus funcionalidades correspondientes. David hizo la vista del árbol (que, aunque no funciona del todo bien, sigue aportando algo) y Nacho se encargó de hacer las modificaciones necesarias a la vista para incluir los nuevos selectores (problema y bloating) y pequeños detalles. En general todo este viaje vivido durante la asignatura ha unido a nuestros personajes a nivel emocional e incluso físico, evolucionando no solo el algoritmo sino la relación personal que los unía.