

## **DECISIONES DE DISEÑO**

A raíz de las interfaces dadas, hemos ido implementando las diferentes clases necesarias para la ejecución del algoritmo. Como dato destacable, hemos añadido en la clase nodo un campo que apunta al padre de dicho nodo, apuntando a null en caso de que el nodo sea la expresión (raíz) de un individuo. De esta manera, se facilita posteriormente el cruce.

Ya en la implementación del algoritmo, en el método que genera una nueva población, hemos decidido que el individuo con el mejor fitness pasa a la siguiente generación. Del resto de individuos, se coge un porcentaje (indicado al crear el objeto algoritmo), y, asegurándonos de que es un número par de individuos, los cruzamos dos a dos con el método de cruce. Por cada dos progenitores, se generan dos hijos, por lo que a la nueva generación pasan solo los hijos, eliminando a los padres de la población.

Para ampliar el diseño, bastaría con añadir mas clases que hereden de Función , Terminal y Dominio, por lo que se tendrían mayores probabilidades a la hora de generar los individuos y estudiarlos.

