

Implementación del árbol con ArrayList

```
public class Cromosoma {  
  
    . . .  
  
    public static String terminales[];  
    public static final String terminales6[] = { "A0", "A1", "D0", "D1", "D2", "D3" };  
    public static final String funciones[] = { "AND", "OR", "NOT", "IF" };  
  
    private Arbol arbol;  
    private double fitness;  
    private double fitness_bruto; //Aptitud antes de transformarla  
    private double punt;  
    private double puntAcum;  
    private String fenotipo;  
  
    public Cromosoma(int profundidad, int tipoCreacion, boolean useIf, int tipoMultiplexor) {  
        arbol = new Arbol(profundidad, useIf);  
        switch(tipoCreacion){  
            case 0:  
                arbol.inicializacionCreciente(0);  
                break;  
            case 1:  
                arbol.inicializacionCompleta(0,0);  
                break;  
            case 2:  
                int ini = new Random().nextInt(2);  
                if(ini == 0) arbol.inicializacionCreciente(0);  
                else arbol.inicializacionCompleta(0,0);  
                break;  
        }  
    }  
    . . .  
  
    public class Arbol{  
  
        private String valor;  
        private ArrayList<Arbol> hijos;  
        private int numHijos;  
        private int numNodos;  
        private int max_prof;  
        private int profundidad;  
        private boolean useIF;  
        private boolean esHoja;  
        private boolean esRaiz;  
  
        . . .  
  
        . . .  
  
        // Devuelve el arbol en forma de array  
        public ArrayList<String> toArray(){  
            ArrayList<String> array = new ArrayList<String>();  
            toArrayAux(array, this);  
            return array;  
        }  
    }  
}
```

```

// Insertar un valor en el arbol (nodo simple)
public Arbol insert(String v, int index){
    Arbol a = new Arbol(v);
    if(index == -1){
        hijos.add(a);
        numHijos = hijos.size();
    }
    else
        hijos.set(index, a);
    return a;
}

// Insertar un arbol en otro arbol.
public void insert(Arbol a, int index){
    if(index == -1){
        hijos.add(a);
        numHijos = hijos.size();
    }
    else
        hijos.set(index, a);
}

public Arbol at(int index){
    return at(this, 0, index);
}

private Arbol at(Arbol a, int pos, int index){
    Arbol s = null;
    if(pos >= index) s = a;
    else if(a.getNumHijos() > 0)
    {
        for(int i = 0; i < a.getNumHijos(); i++)
            if(s == null) s = at(a.getHijos().get(i), pos+i+1, index);
    }
    return s;
}

private void toArrayAux(ArrayList<String> array, Arbol a){
    array.add(a.valor);
    for(int i = 0; i < a.hijos.size(); i++){
        toArrayAux(array, a.hijos.get(i));
    }
}

public int inicializacionCompleta(int p, int nodos){
    int n = nodos;
    int nHijos = 2;
    if(p < max_prof){
        setProfundidad(p);
        Random rnd = new Random();
        int func = 0;

        if(useIF){
            func = rnd.nextInt(Cromosoma.funciones.length);
        }else{
            func = rnd.nextInt(Cromosoma.funciones.length-1);
        }

        this.valor = Cromosoma.funciones[func];
        this.setEsRaiz(true);
    }
}

```