```
56              posf--;
57          }
58      } else {
59          // Seleciona um bloco dentro de []
60          if (Math.random() > 0.5) {
61              while (axiom.length() > posf + 1
62                      && axiom.charAt(posf + 1) != '['
63                      && axiom.charAt(posf + 1) != ']') {
64
65                  posf++;
66              }
67          }
68      }
69      return posf;
70  }
71
72  public String applyCrossover(String axiom2) {
73      String sbsaxiom;
74      int pos1, pos2, posf1, posf2, auxPos;
75      do {
76          pos1 = (int) Math.round(Math.random() * (axiom.length() - 1));
77          posf1 = calcPosf(pos1, axiom);
78          pos2 = (int) Math.round(Math.random() * (axiom2.length() - 1));
79          posf2 = calcPosf(pos2, axiom2);
80
81          if (posf1 < pos1) {
82              auxPos = pos1;
83              pos1 = posf1;
84              posf1 = auxPos;
85          }
86          if (posf2 < pos2) {
87              auxPos = pos2;
88              pos2 = posf2;
89              posf2 = auxPos;
90          }
91      } while (axiom.substring(pos1, posf1 + 1).equals("+")
92              || axiom.substring(pos1, posf1 + 1).equals("-")
93              || axiom2.substring(pos2, posf2 + 1).equals("+")
94              || axiom2.substring(pos2, posf2 + 1).equals("-"));
95
96      sbsaxiom = axiom.substring(pos1, posf1 + 1);
97
98      axiom = axiom.substring(0, pos1) + axiom2.substring(pos2, posf2 + 1)
99              + axiom.substring(posf1 + 1);
100     axiom2 = axiom2.substring(0, pos2) + sbsaxiom
101             + axiom2.substring(posf2 + 1);
102
103     return axiom2;
104 }
105
106 public double applyEvaluation() {
107     countLeaves(); // quantidade de folhas
108     calcHeight(); // altura da árvore
109
110     sizeOfIndividual = axiom.length();
111
```