Construção de uma inteligência artificial utilizando o método NEAT(Neuroevolution of augmenting topologies)

Objetivo:

O objetivo do projeto é a criação de uma inteligência artificial usando o método "Neuroevolution of augmenting topologies" de **Kenneth O. Stanley** e **Risto Miikkulainen**, descrito no artigo homônimo.

Vamos criar o programa em java e usar o aplicativo Robocode como o treinamento das redes. Ao fim, esperamos que a inteligência treinada seja capaz de combater um robô programado à maneira clássica.

O projeto já está sendo codificado e pode se acessado na **página do GitHub**.

Diagrama de Casos De Uso:

O programa deverá ser executado por um usuário, que poderá entre genomas, ou treinar as redes.

Ao testar um genoma, o programa salvará uma rede (Usando a interface Serializable); essa classe será aberta pelo arquivo RobotusCodus, que, por sua vez, é executado pelo programa RoboCode, também executado automaticamente pela classe principal.

O treinamento acontece de forma semelhante, o processo de teste da rede da mesma maneira que a funcionalidade descrita acima. Exceto pelo fato dos testes serem executados exaustivamente, e seus resultados serem usados para a classificação, seleção, mutação e criação de novas redes.

Obs: O objeto deve ser serializado (salvo), pois o aplicativo é executado em outras estâncias de memória, então, seria de grande dificuldade a utilização da mesma classe e/ou objetos nos mesmos programas. E para a leitura de objetos externos no RoboCode, deve-se desabilitar a segurança usando comandos para a execução da VM.

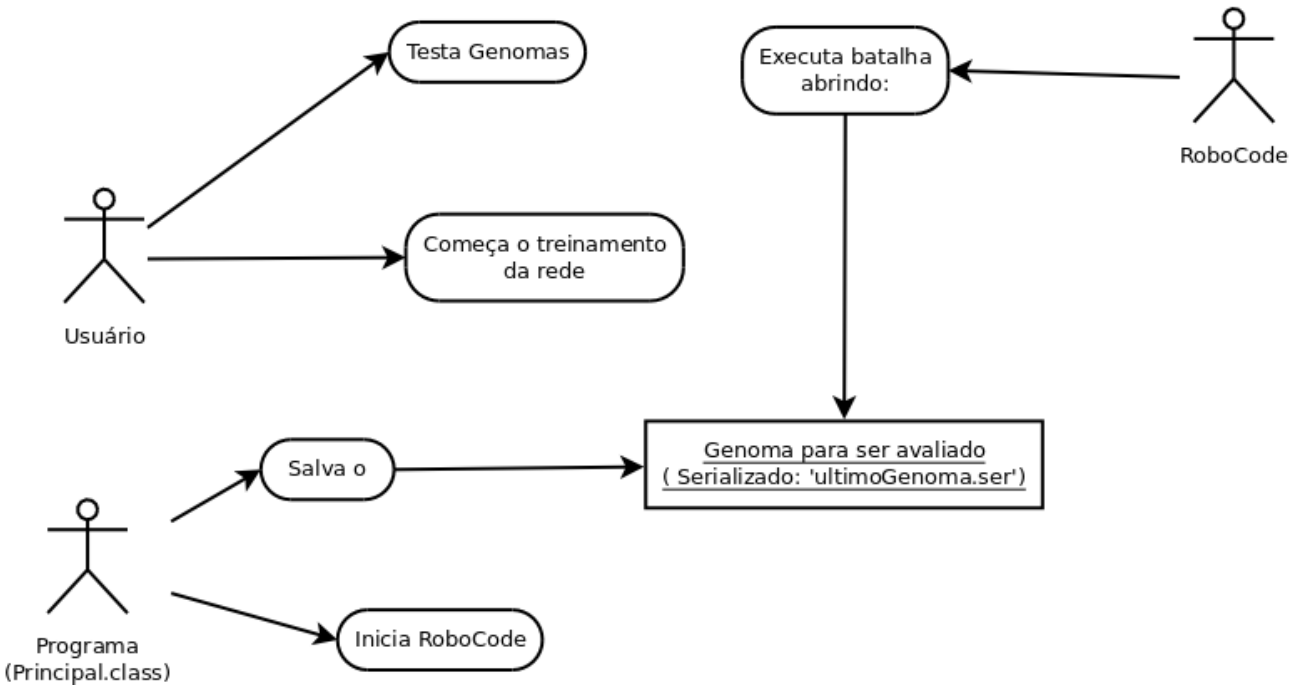
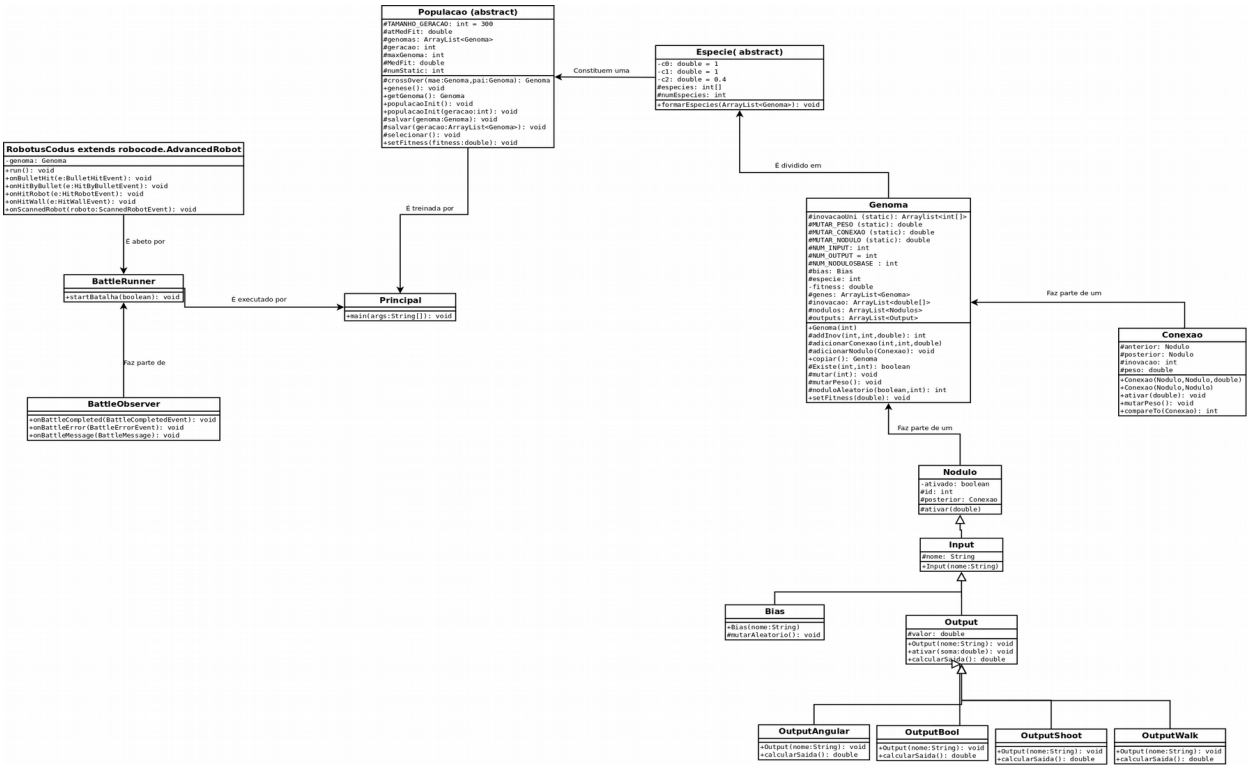
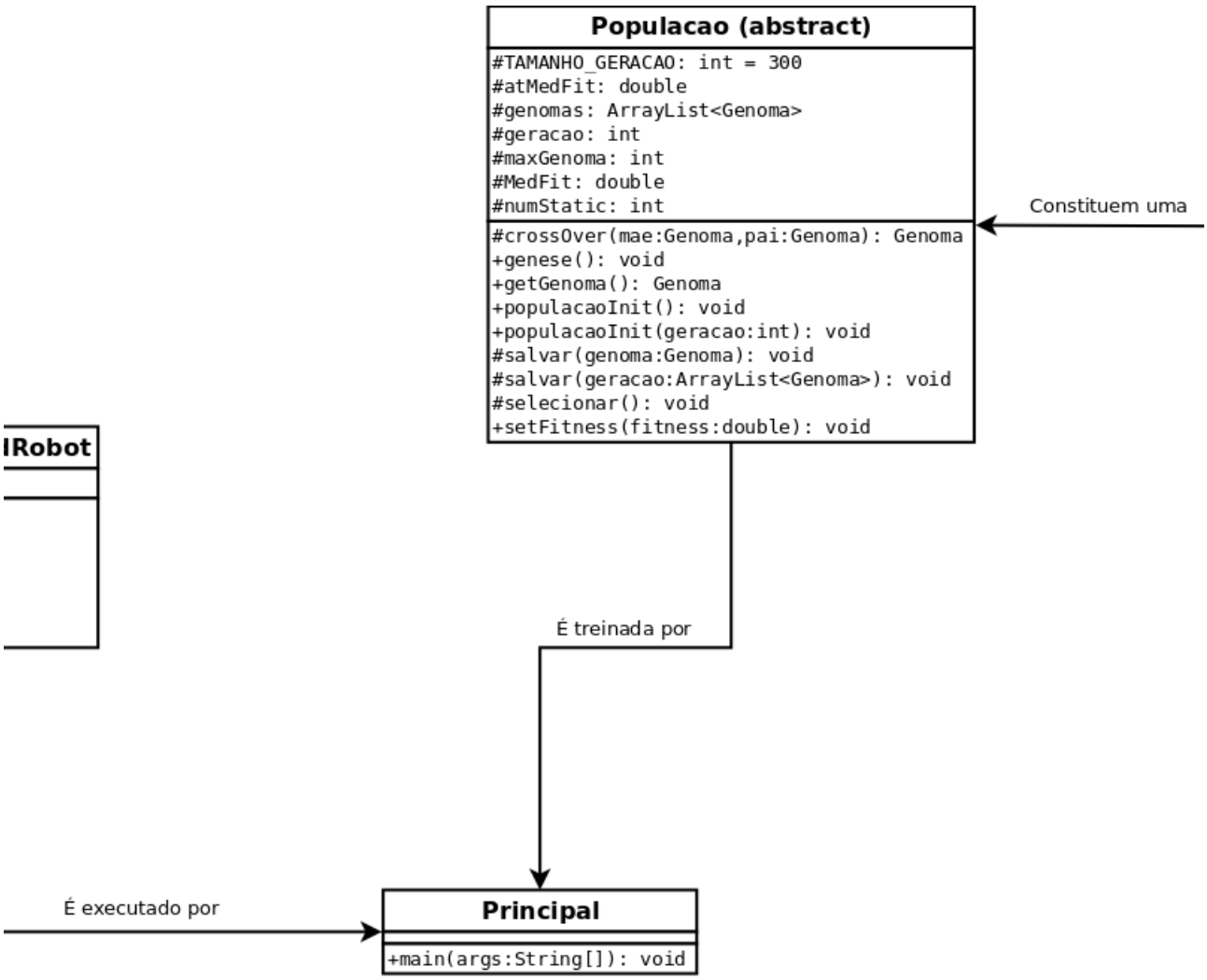


Diagrama de Classes:

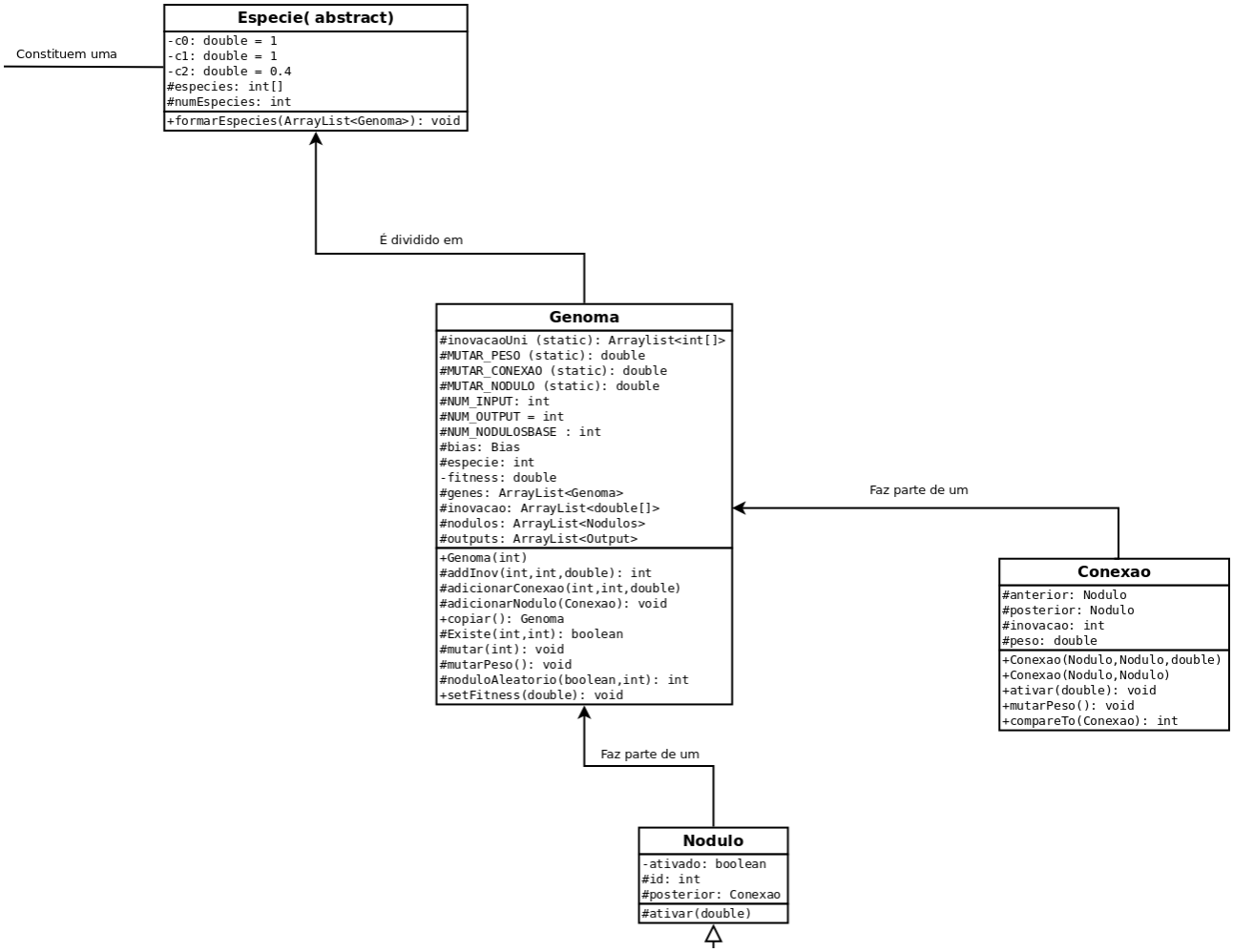
As classes foram criadas da maneira que é descrita no artigo NEAT. Salvo algumas exceções.



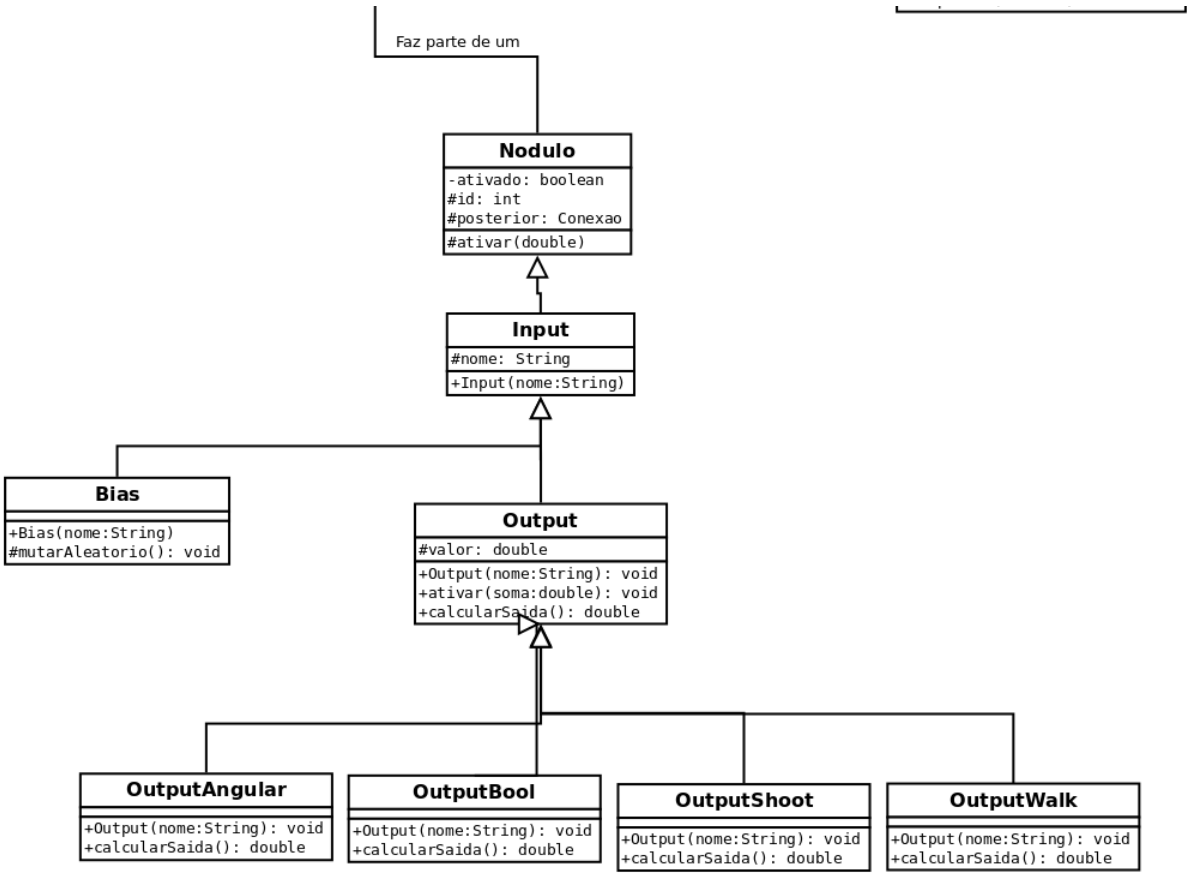


A classe “Principal” é onde é executado o aplicativo.

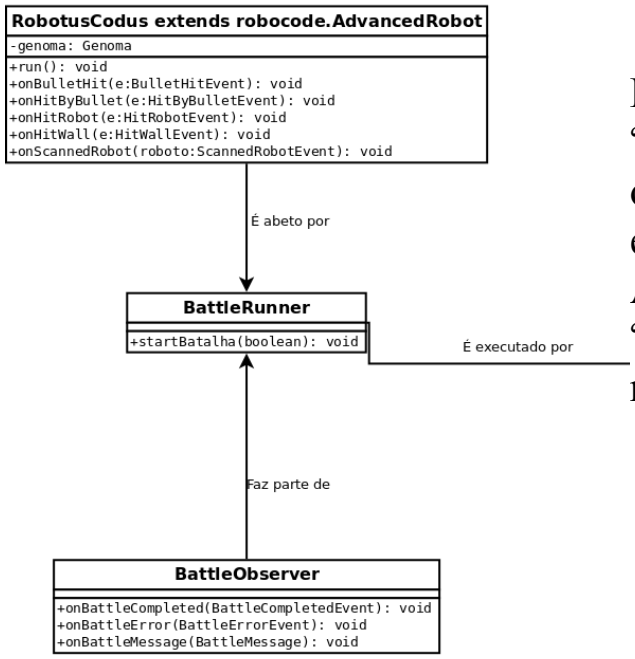
A classe “Populacao” é a qual controla e a seleção, criação e mutação de Genomas, que são as redes. Também é onde são Serializadas as gerações e redes.



Os genomas são constituídos por “Nodulos”e “Conexoes”, responsáveis por conectar os “Inputs” e os “Outputs”; e separados em “Especies”, que é responsável por proteger as inovações topológicas.



Os nódulos podem ser normais, “Inputs”, “Outputs”(e suas variações) ou o “Bias”. Os Inputs recebem as informações; os Outputs retornam a informação, formatada para cada caso; e o bias se conecta com todos os nódulos que não são Inputs, a fim de oferecer controle a rede.



Por fim, a classe “BattleRunner” executa o aplicativo Robocode executa as avaliações. A classe “RobotusCodus” é o robô que usa a rede.