

# Introdução à Inteligência Artificial

## Atividade 4

Universidade Federal da Paraíba

Aluno: Fábio Alexandre E. Melo

### 1.1 Objetivo e Predicado do Robô

O objetivo do robô `saddest_bot` é o de sobreviver pelo máximo de tempo possível, abatendo qualquer inimigo que se aproxime.

O Predicado do `saddest_bot` é “Fugir”.

### 1.2 Atributos Possíveis e Relevantes

Na modelagem deste robô, utiliza-se os atributos de estado: “Pontos de Vida (HP)”, valor inteiro de 0 a 100, sua Bateria\_Restante (valor inteiro de 0 a 100, porcentagem). o atributo booleano “Está\_seguro”, que indica que não há robôs próximos visíveis, e o atributo “Pode\_Atirar”, booleano que indica se há munição para atirar.

### 1.3 Feedback (Supervisão)

A Supervisão e aprendizagem ocorre pela observação direta dos padrões de movimentação e atitudes dos robôs adversários, para formular um modelo de rota que possa ser utilizado para otimizar as chances e o tempo de sobrevivência para formar um dataset que faça possível a previsão de tais movimentos.

## 1.4 Modelo de Dados de Entrada

Instância	Pontos de Vida (HP)	Bateria_Restante	Está_Seguro	Pode_Atirar	Supervisão do Predicado
1	100	100	true	true	não
2	80	80	false	true	não
3	100	40	false	true	sim
4	40	20	false	true	sim
5	20	100	false	true	sim
6	50	50	true	false	não
7	80	80	false	false	sim
8	60	80	true	true	não

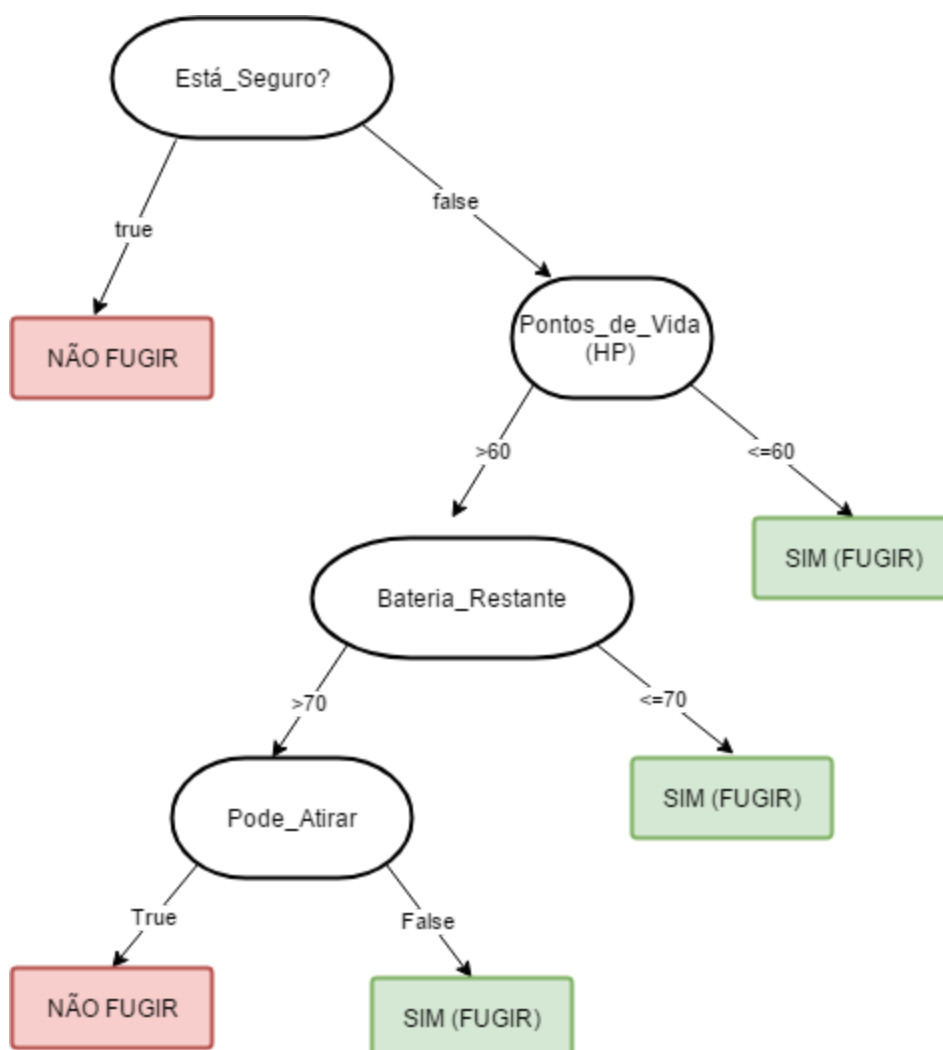
## 1.5 Captura de Exemplos

A Obtenção dos exemplos para a criação do dataset será feita pelas experiências do robô em batalha. o atributo “pontos\_de\_vida” e “bateria\_restante” são fornecidos pela API do jogo, já o “Pode\_Atirar”, da quantidade de munições disponíveis e se a habilidade de atirar não está de alguma forma impedida. já atributo com viés subjetivo “Está\_Seguro” será formulado por observações de localização, observação de outros robôs no campo de batalha.

## 1.6 Sobre os atributos “Ação sendo Executada” e “Tempo”

Não são considerados ação sendo executada e tempo de execução neste modelo de aprendizagem, e nesse exemplo em específico, são necessários apenas os estados atuais do robô, e a proximidade e rota dos adversários.

## 1.7 Árvore de Decisão



## 1.8 Sobre o Algoritmo ID3

No Github, há Alguns projetos interessantes que utilizam o algoritmo ID3, como o Data Mining Classifier (<https://github.com/yanqliuy/DataMiningClassifier>) que utiliza o ID3 em conjunção com técnicas de inferência bayesianas e K nearest neighbors” para classificar e ordenar datasets, e gerar suas respectivas árvores de decisão. e também esta implementação do ID3, que recebe um dataset como entrada, e automaticamente popula os campos incompletos nos sets de aprendizagem na árvore de decisão.

.(<https://github.com/danielrizea/ID3-Machine-Learning-Algorithm>)

A Evolução do algoritmo ID3 mais popular é a **C4.5**, que estende o ID3 com a capacidade de receber atributos contínuos, lidar nativamente com dados incompletos em sets de treinamento, lidar com “pesos” em atributos e diminuir a árvore de decisão removendo atributos considerados irrelevantes. O Software Weka possui uma implementação do algoritmo C4.5 em java (J48) em sua biblioteca padrão., Há também o **C5.0**, que estende a C4,5, e melhora a eficiência de memória, gera árvores de decisões ainda menores e dá suporte à descoberta e remoção de atributos que possam ser considerados inefetivos.

Na problemática do robocode, a implementação de um algoritmo baseado em ID3 deverá ser utilizado para gerar o set de dados mais leve e eficiente possível, por isso neste robô, o algoritmo ideal seria o C5.0.