

# CiberRato

**Artur Pereira / Nuno Lau**

Iris Lab/IEETA – DETI/Universidade de Aveiro



- **Micro-Rato em ambiente simulado**
  - Simulador implementa o labirinto e os corpos dos robôs virtuais
  - Sensores e atuadores disponíveis são iguais em todos os robôs
- **Participantes desenvolvem o software de controlo (cérebro) dos robôs**
  - Este software toma decisões autonomamente (agentes de software)
- **Destinado a equipas que se querem focar na componente de software de um robô**



- Labirinto com um ou mais faróis, áreas-alvo, paredes de alturas diferentes e uma grelha de pontos de partida



- Partir da posição na grelha de partida e fazer algo
  - por exemplo, visitar todas as áreas-alvo debaixo de faróis e regressar o mais perto possível ao ponto de partida



# Corpo do robô virtual

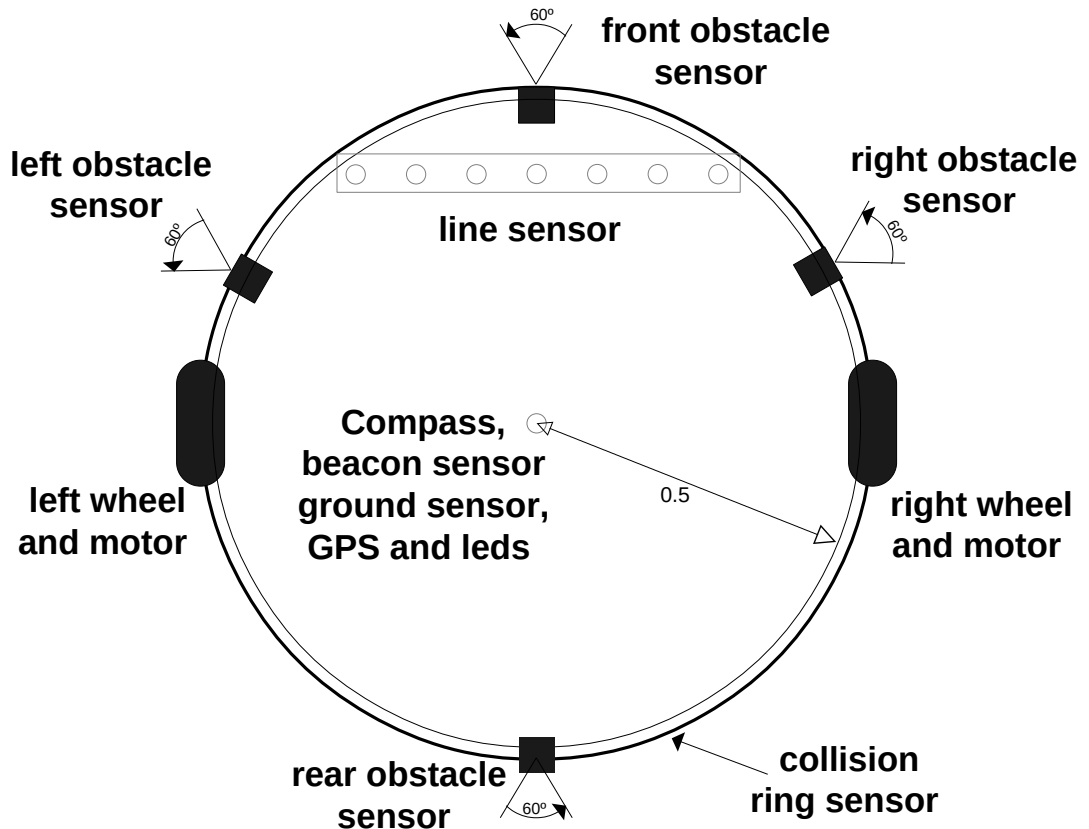
- O robô virtual está equipado com:

- **Sensors**

- Obstacles
- Beacon
- Ground
- Collisions
- Compass
- GPS
- Line
- Leds

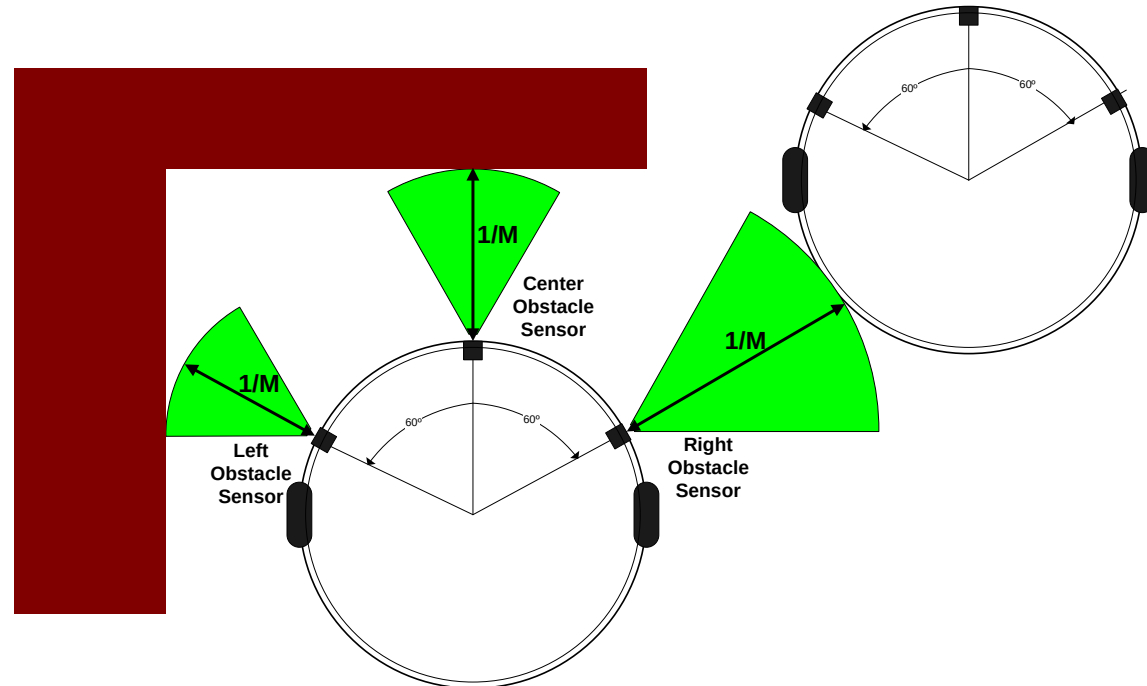
- **Actuators**

- 2 Motors
- Signaling Leds



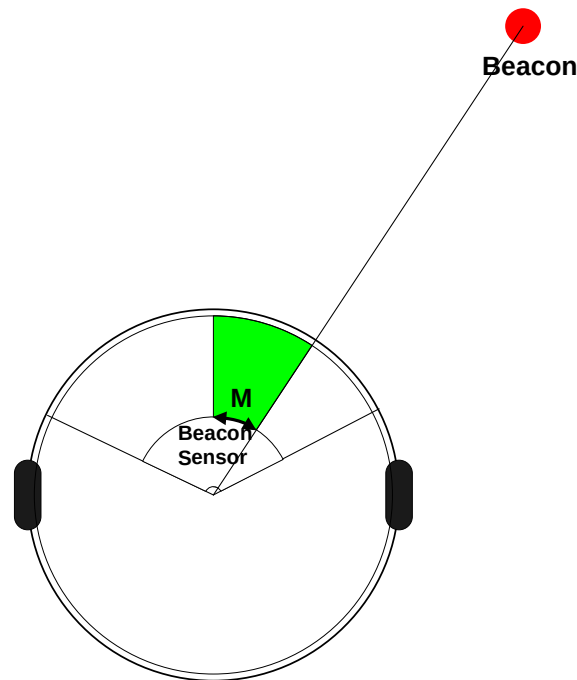
- Sensor de obstáculos

- Medida é inversamente proporcional à distância ao obstáculo mais próximo
- Ruído gaussiano aditivo
- Latência



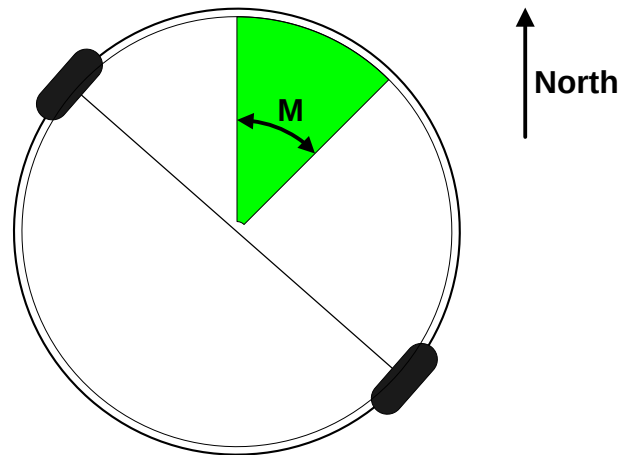
- **Sensor de farol:**

- Medida é igual à distância angular do eixo frontal do robô ao farol
- Ruído gaussiano aditivo
- Latência



- **Bússola**

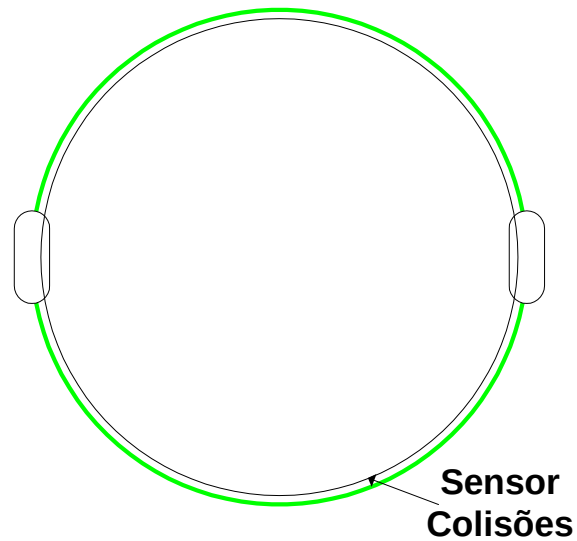
- Medida é a posição angular do eixo frontal do robô em relação ao Norte virtual
- Ruído gaussiano aditivo
- Latência





- **Sensor de contacto**

- Medida binária indica se o robô colidiu
- Sem ruído
- Sem latência

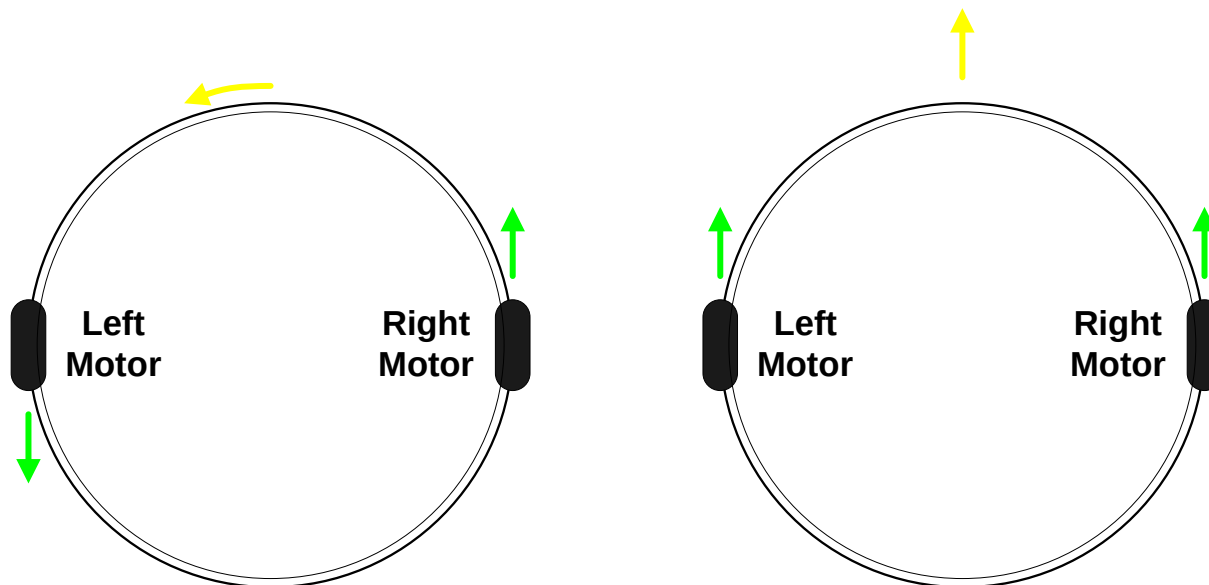


- Os sensores têm ruído e latência
- O número de sensores acessíveis por ciclo pode ser limitado
  - Agentes podem ter de indicar os sensores cujas medidas querem receber no ciclo seguinte

Sensor	Range	Resolution	Noise type	Deviation	Latency	On request
Obstacle s.	[ 0.0, 100.0 ]	0.1	aditive	0.1	0	yes
Beacon s.	[ -180, +180 ]	1	aditive	2.0	4	yes
Compass	[ -180, +180 ]	1	aditive	2.0	4	yes
Bumper	Yes/No	..... N/A .....			0	no
Ground s.	Yes/No	..... N/A .....			0	yes

- **Locomoção diferencial**

- Controlo do movimento do robô é feito através de 2 motores independentes
- Motores com ruído gaussiano e inércia



- **Modelo dos motores**
  - Aproximação de 1ª ordem

$$lin = \frac{out_{right} + out_{left}}{2} \quad rot = \frac{out_{right} - out_{left}}{diam}$$

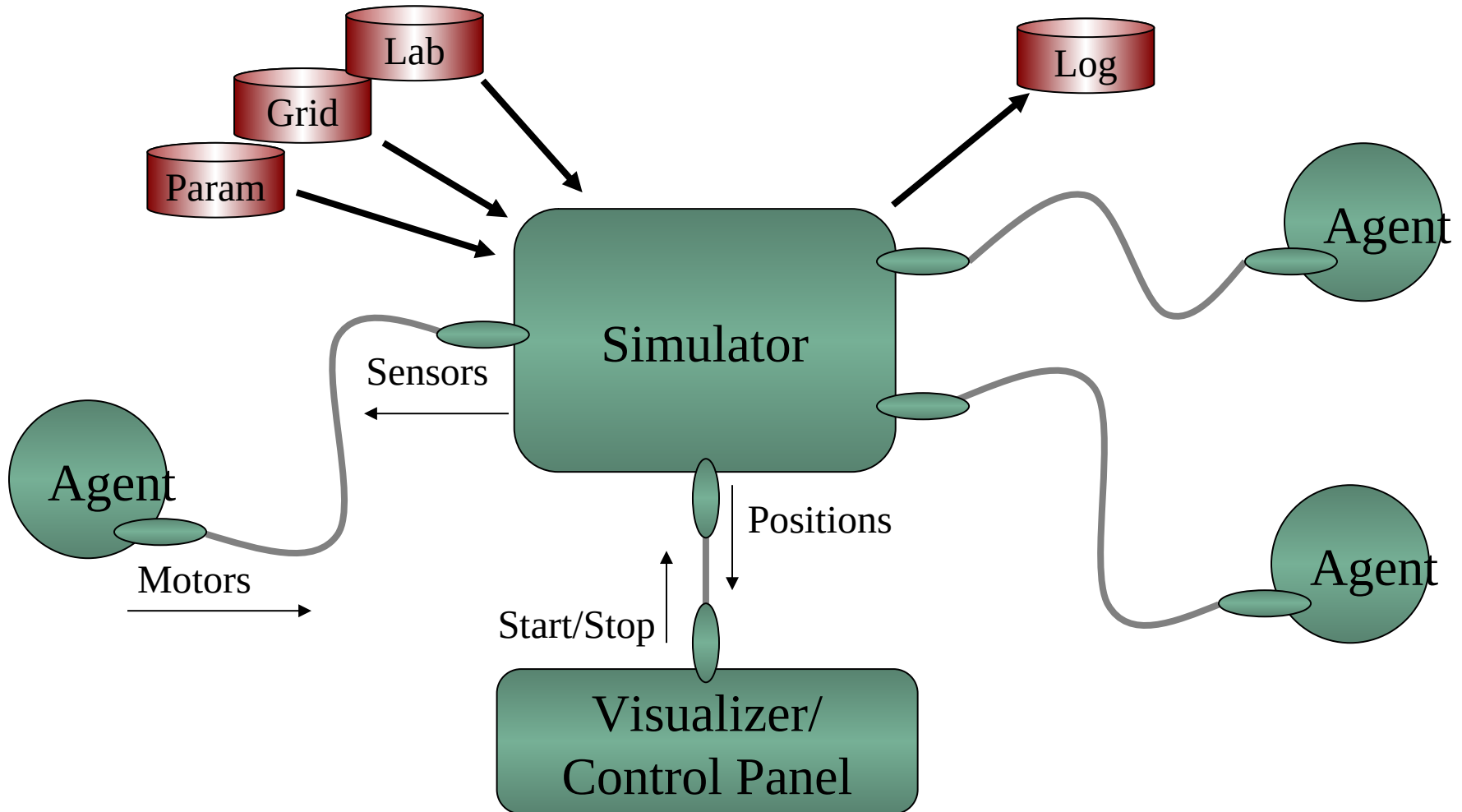
- Com inércia, simulada por um filtro IIR

$$out_t = (in_t * 0.5 + out_{t-1} * 0.5) * noise$$

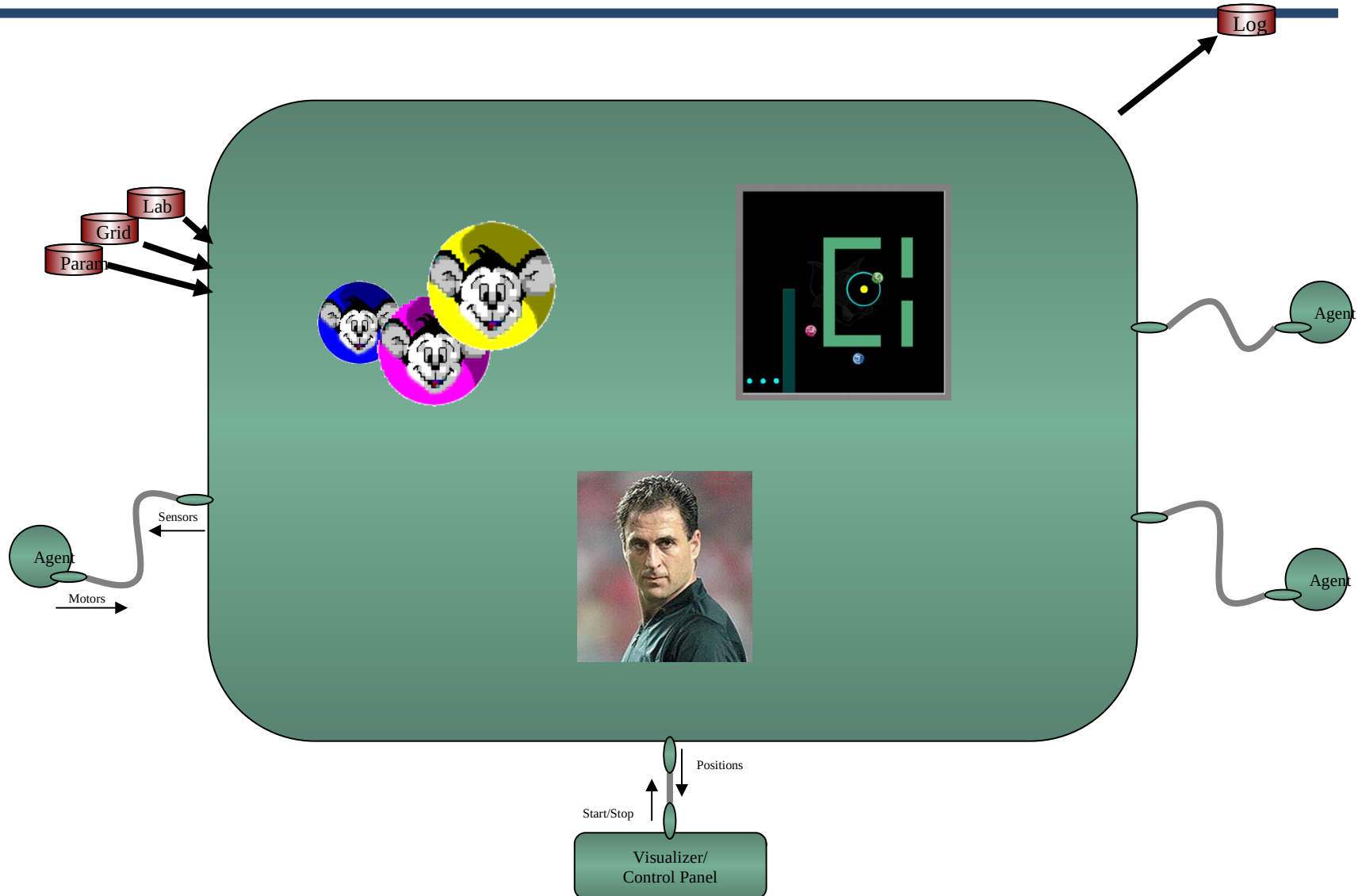
- Motores têm ruído e inércia
  - Simulador adiciona ruído às ordens de atuação
  - Inércia impede uma ordem de ter efeito imediatamente
- Os leds são usados para sinalizar eventos

Actuator	Range	Resolution	Noise type	Standard deviation
Motor	[ -0.15, +0.15 ]	0.001	multiplicative	1.5%
<i>End led</i>	On/Off	..... N/A.....		
<i>Return led</i>	On/Off	..... N/A.....		
<i>Beacon led</i>	On/Off	..... N/A.....		

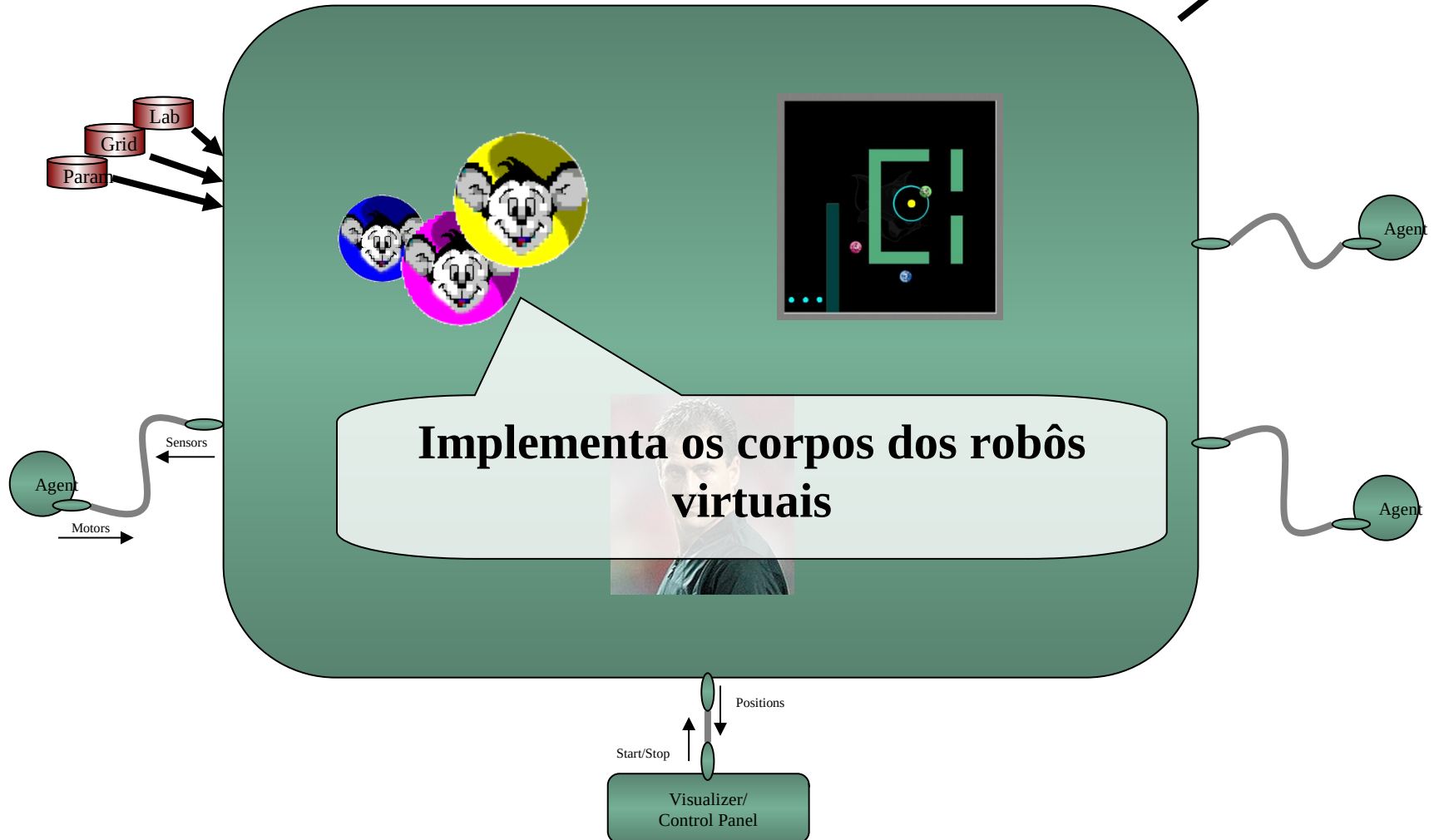
# Ambiente de simulação



# Simulador

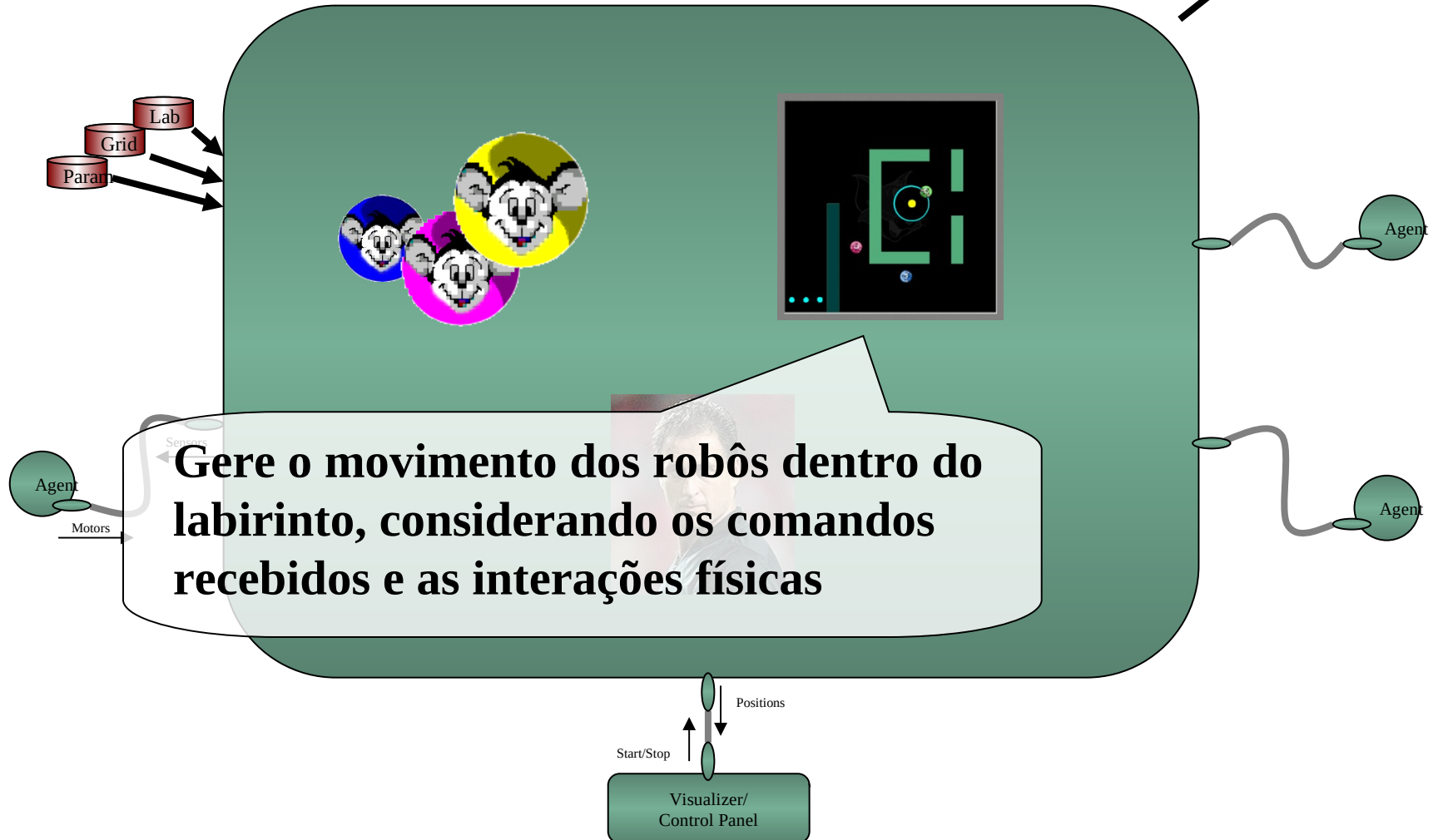


# Simulador

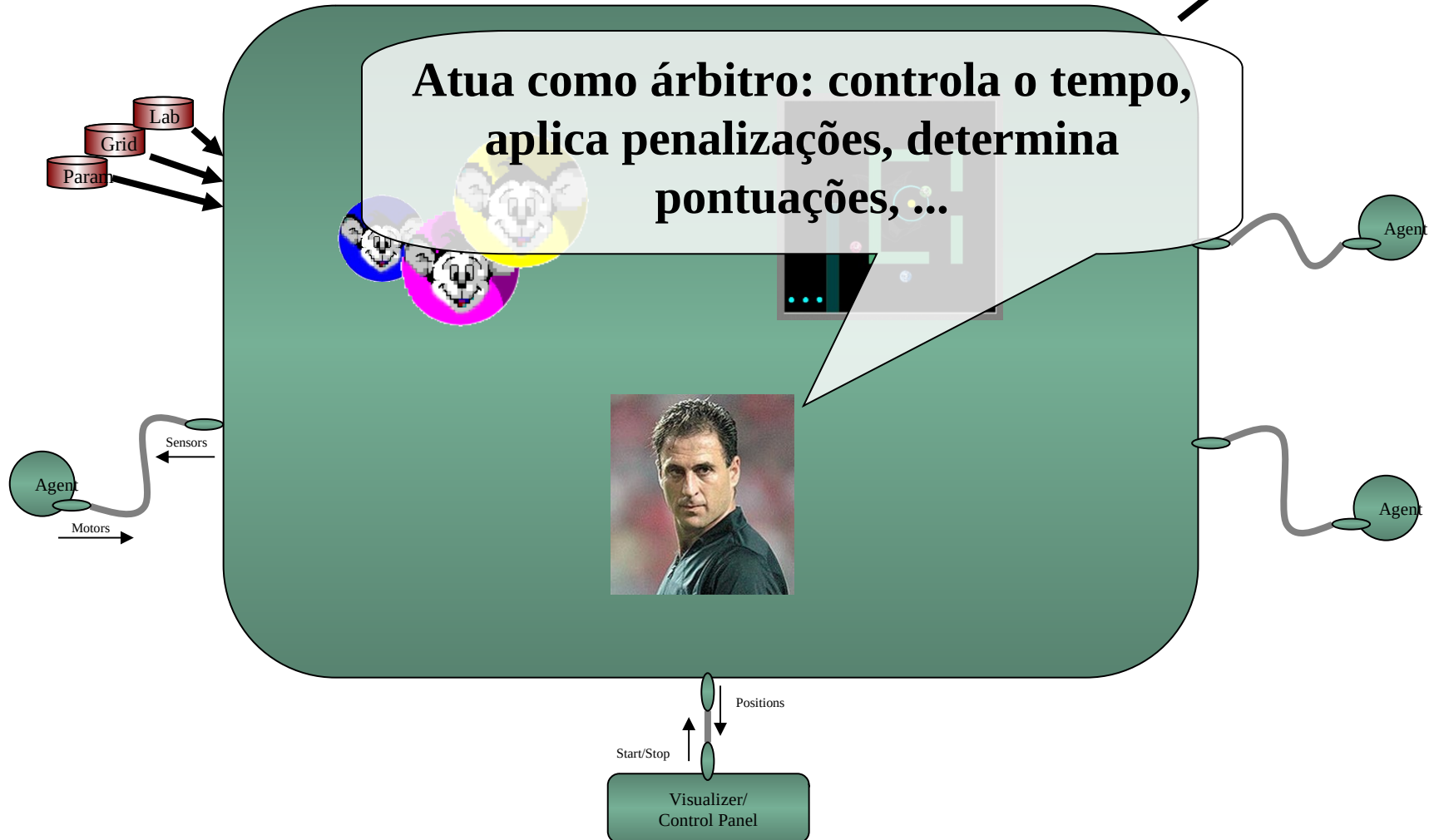




# Simulador

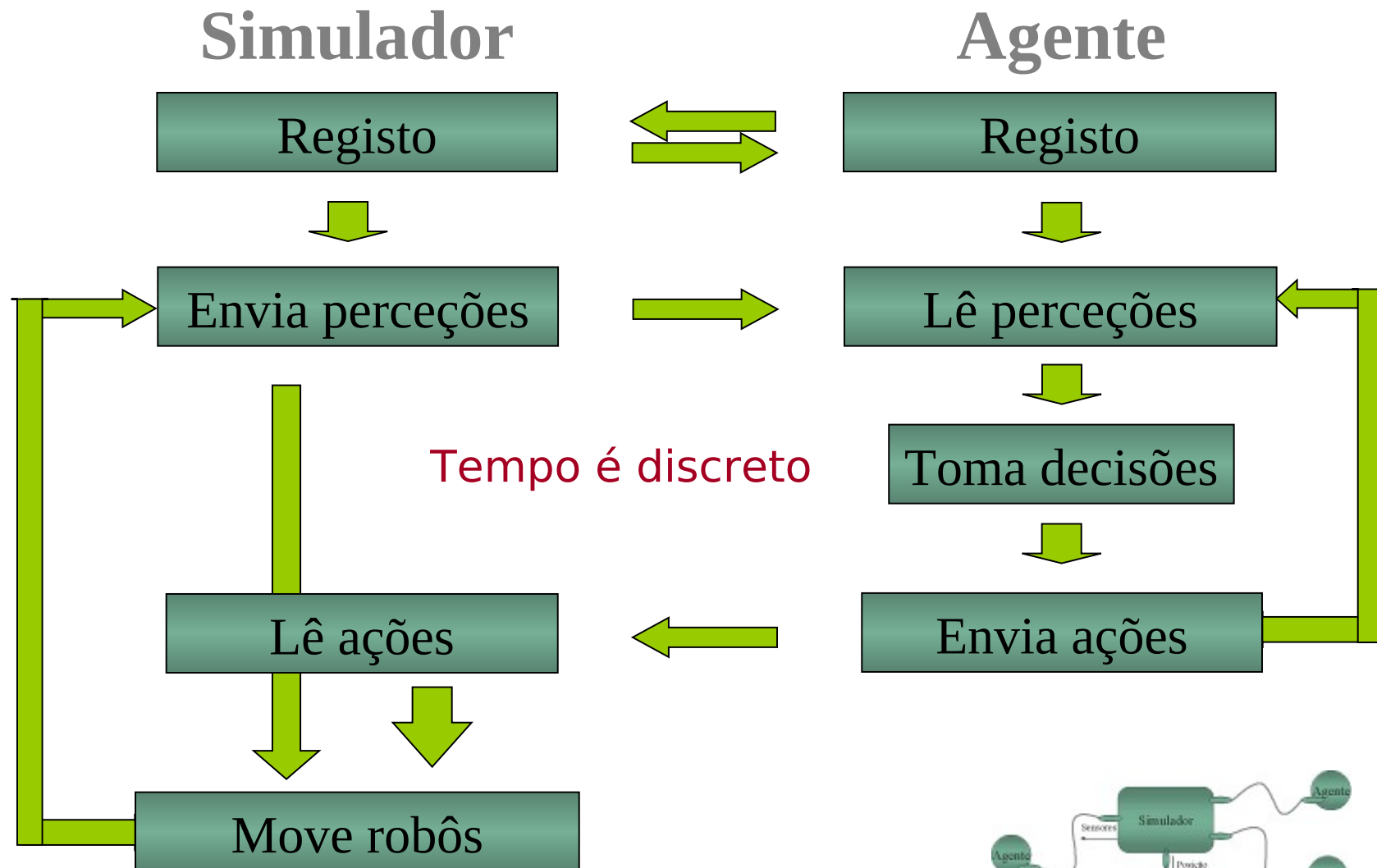


# Simulador



- Apresenta graficamente o estado da simulação
- Mostra o painel de pontuações





- Programável em qualquer linguagem e em qualquer sistema operativo
- Comunicação em IP/UDP
- Interação feita através de mensagens XML
  - Há bibliotecas em C, Prolog, Java, Visual Basic