



Universidade Federal de Campina Grande  
Unidade Acadêmica de Engenharia Elétrica  
Sistemas a Eventos Discretos - 2024.2  
Prof. Kyller Costa Gorgônio

## **Controle de Robôs Autônomos em um Armazém**

Gabrielle Pereira Barbosa  
Sarah Stella Borba Miguel  
Victor Gomes de Freitas Borge

Campina Grande - PB  
12 de março de 2025

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Descrição do Sistema</b>	<b>2</b>
2.1	Funcionamento do Sistema . . . . .	2
<b>3</b>	<b>Eventos do Sistema</b>	<b>3</b>
3.1	Eventos Controláveis . . . . .	4
3.2	Eventos Não Controláveis . . . . .	4
3.3	Regras do Sistema . . . . .	4
<b>4</b>	<b>Metodologia</b>	<b>5</b>
4.1	Autômatos Finitos . . . . .	5
<b>5</b>	<b>Implementação</b>	<b>5</b>
5.1	Ferramentas Utilizadas . . . . .	6
5.2	Síntese de Supervisores . . . . .	6
5.3	Casos de Teste e Simulação . . . . .	7

## 1 Introdução

O projeto tem como objetivo desenvolver um sistema de controle supervisionado para robôs móveis em um armazém automatizado. Utilizamos conceitos de autômatos finitos e controle supervisionado para garantir que os robôs realizem entregas dentro das restrições de segurança.

## 2 Descrição do Sistema

O sistema é composto por:

- **3 Robôs (R1, R2, R3):** Transportam caixas do BE para as máquinas de processamento.
- **4 Máquinas (M1, M2, M3, M4):** Recebem os insumos e processam as cargas.
- **Buffer de Entrada (BE):** Ponto de coleta de insumos pelos robôs.

Os robôs possuem rotas predefinidas e seguem regras que evitam conflitos no BE. Neste modelo, ignoramos colisões e eventos de espera, assumindo que cada robô sempre consegue realizar suas operações.

### 2.1 Funcionamento do Sistema

O fluxo do sistema ocorre da seguinte maneira:

1. Uma máquina (M1, M2, M3 ou M4) solicita um insumo quando precisa de material para processamento.

2. O sistema identifica qual robô está disponível para atender à solicitação, respeitando as seguintes restrições:
  - Apenas o robô R1 pode atender às máquinas M1 e M2.
  - Apenas o robô R2 pode atender às máquinas M3 e M4.
3. O robô designado desloca-se até o Buffer de Entrada (BE) para coletar a carga.
4. Após pegar a carga, o robô transporta o insumo até a máquina solicitante.
5. O robô entrega a carga à máquina e retorna ao BE para ficar disponível para novas solicitações.

Casos previstos no sistema incluem:

- **Requisição simultânea:** Se mais de uma máquina solicitar um insumo ao mesmo tempo, os robôs atenderão conforme a disponibilidade, respeitando suas rotas específicas.
- **Falha de robô:** Se um robô falhar durante o transporte, o robô reserva (R3) assume sua função.
- **Máquina sem requisição:** Se nenhuma máquina precisar de insumos, os robôs permanecem parados no BE.

### 3 Eventos do Sistema

Os eventos do sistema são definidos conforme a especificação e são divididos em controláveis e não controláveis.

### 3.1 Eventos Controláveis

- **move(Ri, BE, Mx)**: O robô  $R_i$  transporta uma caixa do BE para a máquina  $M_x$ .
- **unload(Ri, Mx)**: O robô  $R_i$  descarrega a caixa na máquina  $M_x$ .

### 3.2 Eventos Não Controláveis

- **request(BE, Mx)**: Uma máquina  $M_x$  solicita um insumo ao BE.
- **robotfail(Ri)**: O robô  $R_i$  apresenta falha.
- **robotreset(Ri)**: O robô  $R_i$  volta a funcionar após um reparo.

### 3.3 Regras do Sistema

O sistema segue as seguintes regras operacionais:

1. Uma máquina só pode solicitar insumos quando precisar processar uma carga.
2. As máquinas M1 e M2 só podem ser atendidas pelo robô R1.
3. As máquinas M3 e M4 só podem ser atendidas pelo robô R2.
4. Apenas um robô pode pegar uma carga no BE por vez.
5. O robô R3 só entra em operação quando R1 ou R2 falham.
6. Um robô só pode transportar uma carga por vez.
7. Após entregar uma carga, o robô deve retornar ao BE para aguardar novas solicitações.

## 4 Metodologia

A modelagem do sistema foi feita utilizando autômatos finitos, representando os estados e transições dos robôs, máquinas e do buffer de entrada. A ferramenta Supremica foi utilizada para a síntese de supervisores, garantindo que o sistema opere corretamente.

### 4.1 Autômatos Finitos

Cada componente do sistema foi modelado como um autômato finito:

- **Robôs (R1, R2, R3):** Responsáveis pelo transporte dos insumos.
- **Máquinas (M1, M2, M3, M4):** Mudam de estado conforme recebem e processam cargas.
- **Buffer de Entrada (BE):** Controla o acesso dos robôs para evitar congestionamento.

## 5 Implementação

Para o problema em questão, criamos 7 autômatos e um supervisor. Os autômatos foram: robô 1, robô 2, robô 3, máquina 1, máquina 2, máquina 3 e máquina 4. Escolhemos não usar um autômato para o BE porque estamos desconsiderando a possibilidade de colisão e considerando que ele ignora quando outro robô está pegando a caixa nele.

Para os autômatos do robô 1 e 2, temos o seguinte funcionamento: eles ficam em repouso até ter um request, após o request tem o move para eles carregarem a caixa do BE até a máquina requerida, seja a máquina 1 e 2, no caso do robô 1 ou máquina 3 e 4, para o robô 2. Quando chega na máquina tem o unload para depositar a caixa. As falhas podem acontecer

em qualquer momento do percurso, caso haja, o robô fica no estágio de falha e fica parado desse estágio até que seja resetado.

Nesse momento entra a ação do autômato do robô 3. Quando há falha no robô 1 ou 2, o robô 3 sai do repouso e assume a função do move. Quando o robô 1 quebra, ele recebe duas opções de ação: o request da máquina 1 ou 2, ou seja, a máquina que estava requerendo o robô antes da quebra; ou resetar o robô quebrado, onde R3 volta para o estado de descanso. Isso faz com que ou o robô quebrado seja consertado ou que haja a substituição pelo robô 3 no trajeto do robô que quebrou. Vale ressaltar que, para o que nós pensamos, se um robô quebra, o outro não pode quebrar.

Já os autômatos das máquinas estão no estado inicialmente paradas, até requererem que o robô associado a elas pegue a caixa no BE e leve até elas, ou pode haver a quebra desse robô. Caso haja a quebra, a máquina faz a solicitação do robô 3 para resolver o problema.

## 5.1 Ferramentas Utilizadas

Para o desenvolvimento do projeto, utilizamos:

- **Supremica:** Para modelagem e síntese dos supervisores.
- **LaTeX:** Para documentação do projeto.

## 5.2 Síntese de Supervisores

Na ferramenta Supremica, existe a aba *Analyzer* que nos dá a possibilidade de fazer uma síntese dos componentes do nosso sistemas. Essa síntese é o supervisório.

Com o supervisório, podemos ter uma visão geral de todos os componentes e todos os eventos do sistema, além de também ter o controle geral das possibilidades do sistema. O supervisório

Figura 1: Sistema supervisório.



Fonte: Autoria Própria, 2025.

possui 75 estados e 28 eventos que podem ocorrer. Entretanto, mesmo com esse número de estados, o supervisório é controlável e não bloqueante, duas características que eram buscadas ao ser construído.

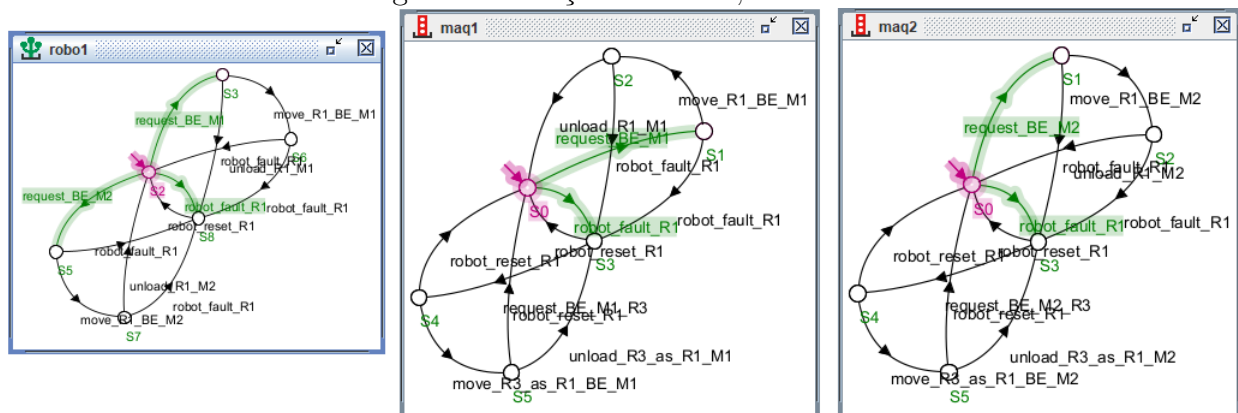
### 5.3 Casos de Teste e Simulação

Alguns testes podem ser feitos para testar o funcionamento do sistema:

1. Uma máquina (M1, M2, M3 ou M4) e seu respectivo robô (R1 ou R2).

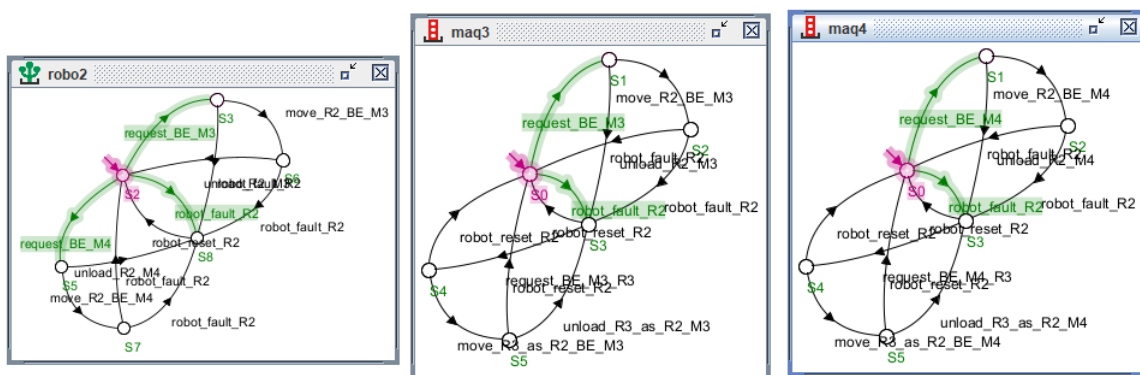


Figura 2: Relação entre R1, M1 e M2.



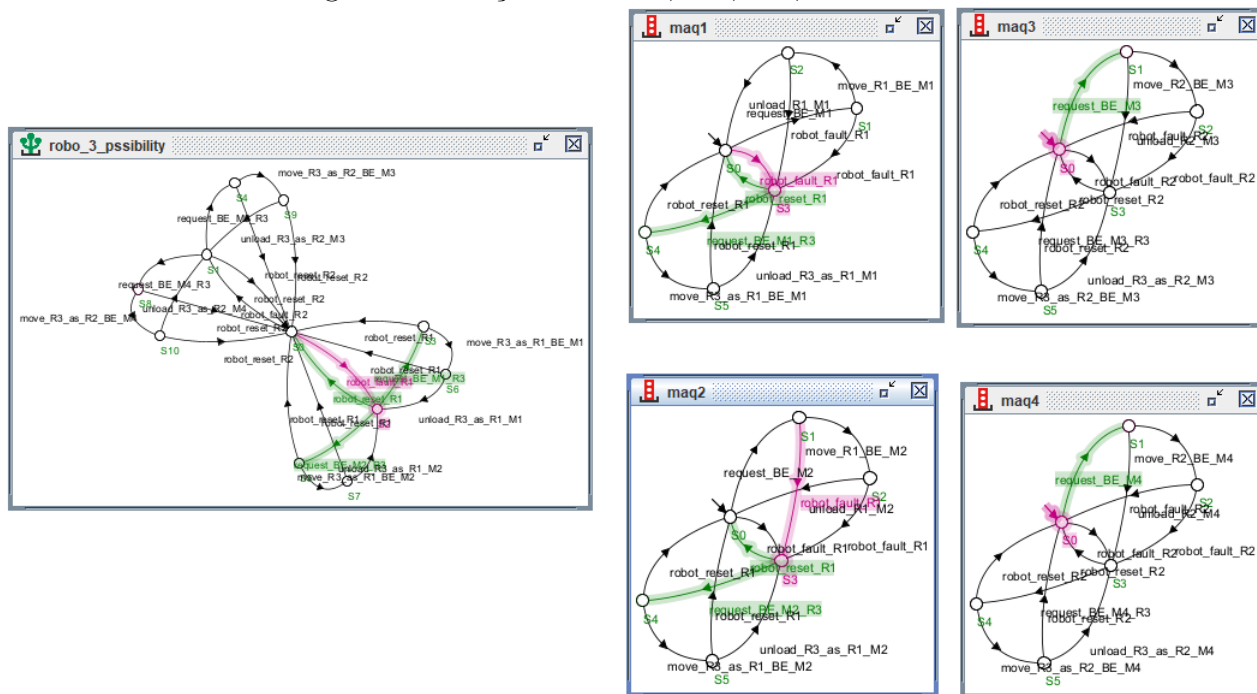
Fonte: Autoria Própria, 2025.

Figura 3: Relação entre R2, M3 e M4.



Fonte: Autoria Própria, 2025.

Figura 4: Relação entre R3, M1, M2, M3 e M4.

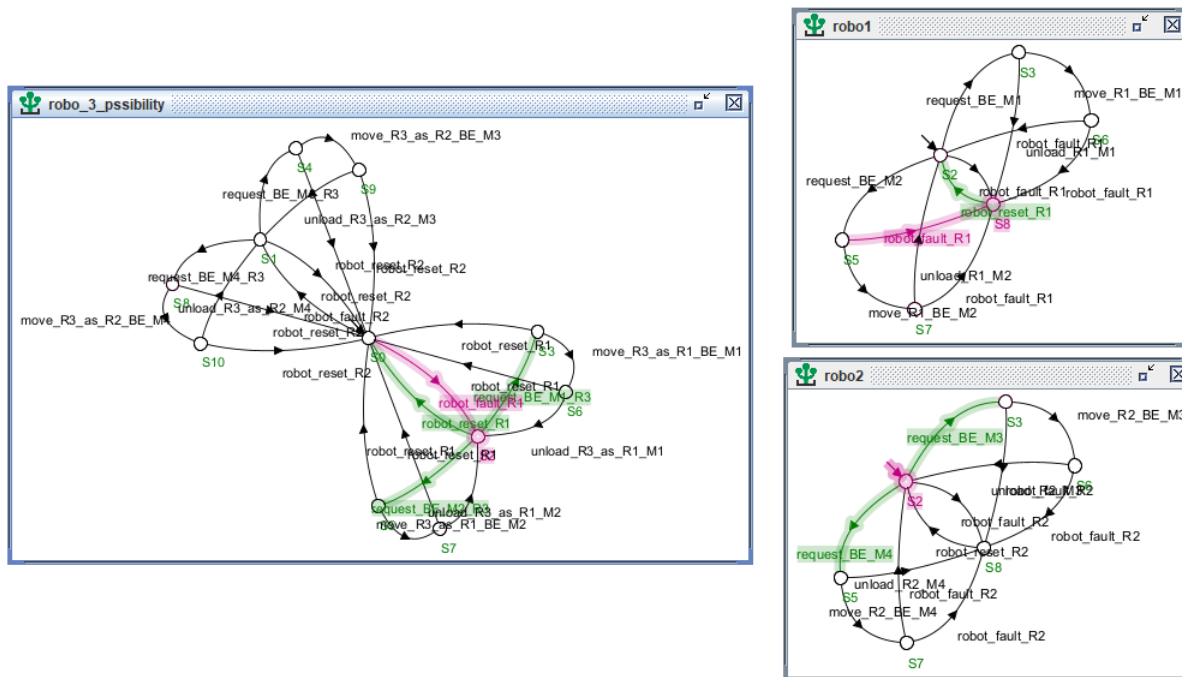


Fonte: Autoria Própria, 2025.

2. Uma máquina (M1, M2, M3 ou M4) e o robô 3 (R3).
3. Um robô (R1 ou R2) e o robô 3 (R3).
4. Um ou mais componentes (M1, M2, M3, M4, R1, R2 ou R3) e o sistema supervisorio.

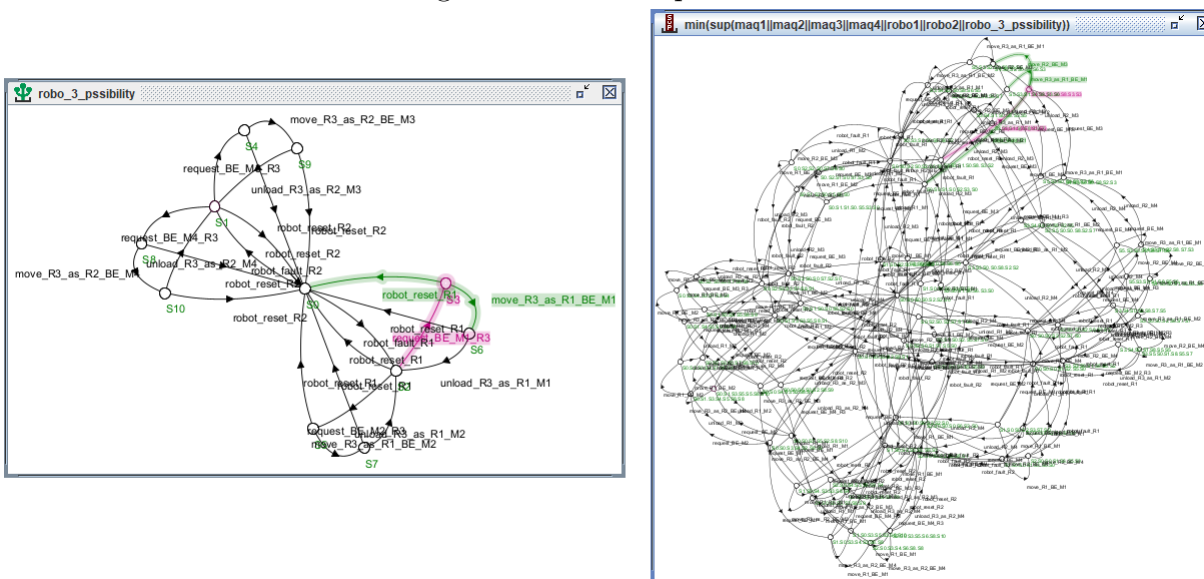
A partir desses testes, podemos observar o funcionamento do sistema, a relação entre cada um dos componentes e como o supervisorio funciona como um todo do sistema.

Figura 5: Relação entre os robôs.



Fonte: Autoria Própria, 2025.

Figura 6: R3 e o supervisor.



Fonte: Autoria Própria, 2025.