

**JOÃO PAULO DA SILVA FONSECA**

**REDES DE PETRI DE ALTO NÍVEL E PNRD  
INVERTIDA ASSOCIADAS AO CONTROLE DE  
ROBÔS MÓVEIS: UMA ABORDAGEM PARA  
OPERAÇÕES DE BUSCA E SALVAMENTO EM  
TRILHAS E TRAVESSIAS**



UNIVERSIDADE FEDERAL DE UBERLÂNDIA  
FACULDADE DE ENGENHARIA MECÂNICA

2018

**JOÃO PAULO DA SILVA FONSECA**

**REDES DE PETRI DE ALTO NÍVEL E PNRD INVERTIDA  
ASSOCIADAS AO CONTROLE DE ROBÔS MÓVEIS: UMA  
ABORDAGEM PARA OPERAÇÕES DE BUSCA E SALVAMENTO EM  
TRILHAS E TRAVESSIAS**

**Tese** apresentada ao Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos para obtenção do título de **DOUTOR EM ENGENHARIA MECÂNICA**.

Área de concentração: Mecânica dos Sólidos e Vibrações

Orientador: Prof. Dr. José Jean-Paul Zanlucchi de Souza Tavares

**UBERLÂNDIA - MG**

**2018**

Dados Internacionais de Catalogação na Publicação (CIP)  
Sistema de Bibliotecas da UFU, MG, Brasil.

---

F676r  
2018

Fonseca, João Paulo da Silva, 1988-  
Redes de Petri de alto nível e PNRD invertida associadas ao controle de robôs móveis: uma abordagem para operações de busca e salvamento em trilhas e travessias / João Paulo da Silva Fonseca. - 2018.  
146 f. : il.

Orientador: José Jean-Paul Zanlucchi de Souza Tavares.  
Tese (Doutorado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Mecânica.  
Disponível em: <http://dx.doi.org/10.14393/ufu.te.2018.782>  
Inclui bibliografia.

1. Engenharia mecânica - Teses. 2. Redes de Petri - Teses. 3. Robôs móveis - Teses. Tavares, José Jean-Paul Zanlucchi de Souza. II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Mecânica. III. Título.

CDU: 621

---

Maria Salete de Freitas Pinheiro – CRB6/1262



**ALUNO:** João Paulo da Silva Fonsêca

**NÚMERO DE MATRÍCULA:** 11323EMC003

**ÁREA DE CONCENTRAÇÃO:** Mecânica dos Sólidos e Vibrações

**LINHA DE PESQUISA:** Projetos de Sistemas Mecânicos

**PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA:** NÍVEL DOUTORADO

**TÍTULO DA TESE:**

***“Redes de Petri de Alto Nível e PNRD Invertida Associadas ao Controle de Robôs Móveis: Uma Abordagem para Operações de Busca e Salvamento em Trilhas e Travessias”***

**ORIENTADOR:** Prof. Dr. José Jean-Paul Zanlucchi de Sousa Tavares

A Tese foi **APROVADA** em reunião pública, realizada no Anfiteatro F - Bloco 5O, Campus Santa Mônica, em 18 de maio de 2018, às 14:00 horas, com a seguinte Banca Examinadora:

NOME		ASSINATURA
Prof. Dr. José Jean-Paul Zanlucchi S. Tavares	UFU	
Prof. Dr. Gina Maira Barbosa de Oliveira	UFU	
Prof. Dr. Rogério Sales Gonçalves	UFU	
Prof. Dr. Eric Rohmer	UNICAMP	
Prof. Dr. Pedro Manuel Gonzalez del Foyo	UFPE	

Uberlândia, 18 de maio de 2018



## DEDICATÓRIA

À minha esposa Regianne, que jamais deixou de me incentivar e restaurar meu ânimo durante essa jornada, permanecendo ao meu lado, sobretudo, nos momentos mais difíceis.

## AGRADECIMENTOS

Sinceros agradecimentos a Deus por abençoar a minha vida, cedendo-me a oportunidade de colaborar para o desenvolvimento científico e tecnológico do meu país, pelo privilégio da convivência com pessoas incríveis e por se fazer presente nas simples ações.

À minha esposa Regianne pelo amor, companheirismo, paciência e incentivo, sobretudo durante os momentos de dificuldade. Momentos ao seu lado revigoram o meu ânimo e estimulam o meu ser.

À toda minha família por compreenderem os incontáveis momentos de ausência, bem como pela confiança e estímulo para a conclusão deste projeto.

Ao meu orientador José Jean-Paul por acreditar na relevância do tema, pelo apoio técnico, científico e emocional, pelos ensinamentos e pelos conselhos que em muito auxiliaram o meu desenvolvimento acadêmico e pessoal.

À Universidade Federal de Uberlândia, à Faculdade de Engenharia Mecânica e ao Programa de Pós-graduação em Engenharia Mecânica por acreditarem neste trabalho, pelo espaço físico, materiais e equipamentos disponibilizados para a concretização do projeto.

À Universidade Federal de Goiás e à Escola de Engenharia Elétrica, Mecânica e de Computação pelo apoio ao permitir que eu orientasse parte de minha carga horária para a conclusão deste trabalho.

Aos colegas do *Manufacturing Automated Planning Lab* (MAPL), especialmente ao Alexandre pelo fundamental apoio no desenvolvimento da comunicação TCP/IP e testes com o robô, ao Rafael Froede pelo apoio no desenvolvimento e testes dos últimos robôs, ao Carlos Eduardo Silva pelo suporte na implementação da PNRD invertida, ao Gustavo Diniz pelo auxílio na montagem, ao Andreive Giovanni e ao Bernardo pelos projetos dos primeiros robôs, ao Bruno e ao Ivan, pela impressão de peças no Laboratório de Aeronaves Autônomas, ao Marco Vinícius por me receber de maneira hospitaleira tantas vezes em sua casa e aos demais pelo companheirismo, pela amizade e pelos momentos de descontração.

Aos colegas da EMC/UFG, especialmente ao Demóstenes e ao Felipe que possibilitaram distribuições de atividades que permitissem a minha dedicação ao projeto de doutorado e aos demais pelo apoio e companheirismo ao longo dos últimos dois anos.

À CAPES pelo apoio financeiro.

E a tantos outros que de alguma forma contribuíram para a conclusão deste trabalho.

FONSECA, J.P.S. **Redes de Petri de Alto Nível e PNRD Invertida Associadas ao Controle de Robôs Móveis: Uma Abordagem para Operações de Busca e Salvamento em Trilhas e Travessias**. 2018. 146 f. Tese de Doutorado, Universidade Federal de Uberlândia, Uberlândia.

## Resumo

A cada dia um número maior de pessoas passa a se interessar por atividades ao ar livre. Um exemplo comum no Brasil é caminhar em trilhas e travessias ao longo de biomas naturais, normalmente gerenciados por parques estaduais. Porém, nem sempre os principiantes dispõem de conhecimento da área ou estão acompanhados por um guia e muitos nem mesmo estão fisicamente preparados para completar o percurso. Consequentemente, percebe-se um aumento significativo no número de incidentes de pessoas perdidas ou feridas em trilhas e travessias no país. Devido ao apelo social, algumas áreas da tecnologia já auxiliam estudos científicos sobre operações de busca e salvamento, como é o caso da robótica móvel e da inteligência artificial. No entanto, soluções para ambientes de trilhas e travessias são pouco relatadas na literatura, mostrando-se um desafio para a área. Esta tese propõe uma solução, baseada em redes de Petri de alto nível e na tecnologia RFID, aplicada no controle de robôs móveis para operações de busca e salvamento em trilhas e travessias. Para isso, parte-se do conceito de uma técnica denominada rede de Petri inserida em base de dados RFID (PNRD), originalmente desenvolvida para rastreamento e controle de movimentação de objetos passivos ao longo de processos logísticos e de manufatura. Para ambientes característicos de busca e salvamento, nos quais há interesse em se rastrear e controlar a movimentação de agentes com comportamento ativo, propõe-se uma modificação estrutural da PNRD, denominada PNRD invertida ou iPNRD. A tese apresenta ainda duas arquiteturas de controle utilizadas para integrar os modelos em redes de Petri de alto nível, desenvolvidos na plataforma CPN Tools, tanto com dispositivos embarcados em robôs móveis, como com o ambiente de simulação desenvolvido na plataforma V-REP. Testes operacionais indicam que o método proposto tende a aumentar a rastreabilidade do usuário e direcionar as buscas para área reduzida, aumentando as chances de salvamento.

---

*Palavras-chave: Busca e salvamento. Robôs móveis. Arquitetura de controle. Redes de Petri de alto nível. PNRD invertida.*

FONSECA, J.P.S. **High-Level Petri Nets and Inverted PNRD Associated to Mobile Robot Control: An Approach to Search and Rescue on Trails and Crossings**. 2018. 146 f. Ph.D. Thesis, Universidade Federal de Uberlândia, Uberlândia.

### **Abstract**

Several people are interesting for outdoor activities. A common example in Brazil is walking on trails and crossings along natural biomes. However, beginners do not know the area or have a guide, and many are not physically ready to complete the trek. Consequently, the number of injured or lost people on trails and crossings are increasing. Due to the social appeal, some technological areas carry out scientific studies on search and rescue operations, such as mobile robotics and artificial intelligence. However, solutions for trekking environments are little reported in the literature. This thesis proposes a solution, based on high-level Petri nets and RFID technology, applied in the control of mobile robots for search and rescue operations on trails and crossings. For this, the concept of a technique called Petri net inside RFID database (PNRD), originally developed for tracking and controlling the movement of passive objects along logistic and manufacturing processes, is used. For characteristic search and rescue environments, where there is interest in tracking and controlling the movement of agents with active behavior, a structural modification of PNRD, called inverted PNRD or iPNRD, is proposed. The thesis also presents two control architectures used to integrate the models in high-level Petri nets, developed in the CPN Tools, with devices embedded in mobile robots, as well as with the simulation environment developed in the V-REP platform. Operational tests indicate that the proposed method tends to increase the user's traceability and direct the searches to a reduced area, increasing the chances of rescue.

---

*Keywords: Search and rescue. Mobile robots. Control architecture. High-level Petri nets. Inverted-PNRD.*

## LISTA DE FIGURAS

Figura 1.1 - Gráfico com número de eventos SAR e número de pessoas resgatadas por ano (1994-2014) com o sistema Cospas-Sarsat; adaptado de King (2013) .....	4
Figura 2.1 - Esquema de controle para sistemas robóticos móveis. Adaptado de Siegwart e Nourbakhsh (2004) .....	10
Figura 2.2 - Representação esquemática dos componentes de um sistema RFID. Modificado de Ahsan, Shah e Kingston (2010) .....	21
Figura 2.3 - Rede de Petri elementar com cinco lugares e quatro transições (SILVA, 2017b)	24
Figura 2.4 - Representação gráfica de uma mudança de estado em uma rede de Petri (SILVA, 2017b) .....	26
Figura 2.5 - Esquema ilustrativo do modo de operação de um sistema PNRD típico; baseado em Tavares e Saraiva (2010).....	28
Figura 2.6 - Esquema ilustrativo do modo de operação do DEMIS com o RETIM. Modificado de Tavares et al. (2017).....	29
Figura 2.7 - Exemplo de uma rede de Petri para o processo de verificação de conformidade de mancais (SILVA, 2017b) .....	30
Figura 2.8 - Esquema ilustrativo do sistema PNRD integrado ao sistema de posicionamento interno para controle de posição de robôs móveis; baseado em Tavares et al. (2017) .....	31
Figura 2.9 - Esquema ilustrativo do modo de operação da PNRD associada ao uso da biblioteca para Arduino; baseado em Silva (2017b) .....	31
Figura 2.10 - Distribuição de trabalhos publicados em periódicos e em anais, de 2005 a 2018, por combinação de palavras-chave.....	33
Figura 3.1 - Imagem do Google Earth mostrando os principais pontos da travessia Petrópolis-Teresópolis (GOOGLE, 2017).....	45
Figura 3.2 - Mapa esquemático da travessia Petrópolis-Teresópolis. Adaptado de (TRILHA & CIA, 2008) .....	46
Figura 3.3 - Representação simplificada da travessia Petrópolis-Teresópolis.....	46
Figura 3.4 - Associação dos termos da Eq. 2.4 com os elementos RFID para as abordagens PNRD clássica e iPNRD .....	48
Figura 3.5 - Esquema funcional do agente <i>Walker</i> para a composição do ambiente de investigação.....	49
Figura 3.6 - Esquema funcional do agente <i>GroundBot</i> para a composição do ambiente de investigação.....	50

Figura 3.7 - Esquema funcional do agente <i>AerialBot</i> para a composição do ambiente de investigação.....	50
Figura 3.8 - Esquema funcional do sistema multirrobo para a composição do ambiente de investigação (FONSECA et al., 2017).....	51
Figura 3.9 - Modelo comportamental do <i>Walker</i> em rede de Petri .....	55
Figura 3.10 - Modelo comportamental do <i>GroundBot</i> em rede de Petri .....	58
Figura 3.11 - Modelo comportamental do <i>AerialBot</i> em rede de Petri .....	60
Figura 3.12 - Representação clássica da arquitetura conhecida como pirâmide da automação. Adaptado de SMC (2017).....	62
Figura 3.13 - Relacionamento entre os três primeiros níveis da pirâmide da automação e a arquitetura de controle proposta .....	62
Figura 3.14 - Esquemático da arquitetura de controle proposta com detalhamento da interação entre os processos .....	63
Figura 3.15 - Esquema ilustrativo do modo de operação do sistema proposto .....	64
Figura 3.16 - Exemplo de uma rede ilustrando a utilização das funções COMMS/CPN .....	66
Figura 3.17 - Detalhe dos elementos de conexão e de comunicação na rede CPN .....	67
Figura 3.18 - Fotografia da placa STM32 <i>BluePill</i> (STM32DUINO, 2018) .....	68
Figura 3.19 - Modelo CAD do robô cartesiano apresentado por Rodrigues (2017) .....	68
Figura 3.20 - Modelo CAD do robô móvel apresentado por Silva (2017a).....	69
Figura 3.21 - Modelo do robô móvel (antes e após a aplicação da ferramenta <i>Extract the convex hull</i> ) em ambiente de simulação V-REP.....	69
Figura 3.22 - Modelo do robô cartesiano em ambiente de simulação V-REP.....	70
Figura 4.1 - Rede de Petri de alto nível com o modelo comportamental do <i>AerialBot</i> .....	75
Figura 4.2 - Detalhe da sub-rede associada à transição de substituição “ <i>MsgApoio</i> ”.....	75
Figura 4.3 - Rede de Petri de alto nível com o modelo comportamental do <i>GroundBot</i> .....	76
Figura 4.4 - Rede de Petri de alto nível com o modelo comportamental do agente <i>Walker</i> ..	77
Figura 4.5 - Ambiente de simulação composto por um robô móvel e um robô cartesiano....	78
Figura 4.6 - Imagem das janelas de configuração de uma coleção ( <i>Collection</i> ) de objetos e da ferramenta para planejamento de trajetória de um robô com desvio dos obstáculos associados ao elemento <i>Collection</i> .....	79
Figura 4.7 - Imagem da janela de configuração de um grupo de filtros aplicados ao sensor de visão para detecção de vítimas.....	80
Figura 4.8 - Imagem de um cenário do ambiente de simulação composto por três robôs móveis, um robô cartesiano, quatro elementos que simulam etiquetas RFID e um trajeto a ser executado por um dos robôs terrestres.....	81

Figura 4.9 - Representação esquemática dos componentes eletrônicos aplicados aos robôs móveis .....	82
Figura 4.10 - Fotografia detalhando um robô móvel antes e após a montagem dos componentes eletrônicos .....	82
Figura 4.11 - Sequência de imagens indicando a passagem do robô móvel pelos locais especificados na seção 3.2 durante navegação ao longo de uma trajetória planejada .....	84
Figura 4.12 - Sequência de imagens indicando a evolução do <i>token</i> em parte da rede de Petri do agente <i>Walker</i> , saindo do lugar “Desconectado” e entrando no lugar “Preparado”. .....	84
Figura 4.13 - Sequência de imagens indicando a evolução da rede de Petri do agente <i>Walker</i> para os passos 3 e 4, indicados na Fig. 4.11 .....	85
Figura 4.14 - Diagrama de sequência simplificado para execução do caso 1 em ambiente virtual de simulação .....	86
Figura 4.15 - Sequência de imagens indicando a passagem do robô móvel pelos pontos com <i>tags</i> RFID durante possível navegação em ambiente real de testes.....	87
Figura 4.16 - Detalhe do serial Monitor da Arduino IDE durante a execução dos testes funcionais de passagem do agente <i>Walker</i> pelas <i>tags</i> RFID associadas aos locais especificados na seção 3.2.....	88
Figura 4.17 - Sequência de imagens ilustrando o início da patrulha, identificação de vítima e chegada de apoio terrestre .....	89
Figura 4.18 - Sequência de imagens indicando a evolução da rede de Petri do agente <i>AerialBot</i> para os passos 1, 2 e 3, indicados na Fig. 4.17.....	90
Figura 4.19 - Sequência de imagens indicando a evolução da rede de Petri do agente <i>GroundBot</i> para os passos 1, 2 e 3, indicados na Fig. 4.17 .....	91
Figura 4.20 - Diagrama de sequência simplificado para execução do caso 2 em ambiente virtual de simulação .....	92
Figura 4.21 - Sequência de imagens ilustrando o início da busca, identificação de vítima e chegada de apoio aéreo .....	93
Figura 4.22 - Diagrama de sequência simplificado para execução do caso 3 em ambiente virtual de simulação .....	95



## LISTA DE TABELAS

Tabela 1.1 - Notícias sobre pessoas perdidas em trilhas do centro-sul brasileiro .....	3
Tabela 2.1 - Configurações de veículos equipados com rodas. Adaptado de Siegwart e Nourbakhsh (2004) .....	12
Tabela 2.2 - Breve comparação entre simuladores de robótica, modificado de Peralta et al (2016) .....	18
Tabela 2.3 - Resultados da comparação entre os simuladores V-REP e Gazebo baseada no trabalho de Nogueira (2014) .....	19
Tabela 2.4 - Características operacionais das <i>tags</i> ativas e passivas; modificado de Ahsan, Shah e Kingston (2010) .....	20
Tabela 2.5 - Resumo dos resultados da pesquisa bibliográfica a partir da combinação de quatro palavras-chave: “ <i>search and rescue</i> ”, “ <i>robot</i> ”, “ <i>Petri</i> ” e “ <i>RFID</i> ” .....	33
Tabela 2.6 - Resumo dos resultados da pesquisa bibliográfica a partir da combinação de quatro palavras-chave: “ <i>search and rescue</i> ”, “ <i>multi-robot</i> ”, “ <i>colored Petri net</i> ” e “ <i>RFID</i> ” .....	33
Tabela 2.7 - Comparativo das abordagens utilizadas pelos trabalhos semelhantes baseado na integração entre os assuntos correlatos a esta tese .....	42
Tabela 3.1 - Descrição dos lugares e transições utilizadas na modelagem do <i>Walker</i> .....	53
Tabela 3.2 - Descrição dos lugares e transições utilizadas na modelagem do <i>GroundBot</i> .....	57
Tabela 3.3 - Descrição dos lugares e transições utilizadas na modelagem do <i>AerialBot</i> .....	60
Tabela 3.4 - Lista das principais funções e ferramentas utilizadas para desenvolvimento deste trabalho na plataforma V-REP .....	72

## LISTA DE SÍMBOLOS

$A$	Matriz de incidência de uma rede de Petri
$A^t$	Função transporta da matriz de incidência $A$
$E$	Matriz (associada à função $I$ ) que identifica o conjunto de lugares de entrada para determinada transição de uma rede de Petri
$I$	Uma função de entrada que identifica o conjunto de lugares de entrada para determinada transição de uma rede de Petri
$i$	operador matricial associado aos lugares de uma rede de Petri
$j$	operador matricial associado às transições de uma rede de Petri
$k$	instante associado a determinada marcação em uma rede de Petri
$M$	Marcação ou estado da rede de Petri
$M_k$	Marcação ou estado da rede de Petri no instante $k$
$M_{k+1}$	Marcação ou estado da rede de Petri no instante $k + 1$
$m$	quantidade de lugares associados a uma rede de Petri
$n$	quantidade de transições associadas a uma rede de Petri
$O$	Uma função de saída que identifica o conjunto de lugares de saída para determinada transição de uma rede de Petri
$P$	Um conjunto finito com $m$ lugares relacionadas a uma rede de Petri
$P_i$	$i$ -ésimo lugar em uma rede de Petri
$R$	Uma tupla de cinco elementos que descrevem uma rede de Petri
$S$	Matriz (associada à função $O$ ) que identifica o conjunto de lugares de saída para determinada transição de uma rede de Petri
$T$	Um conjunto finito com $n$ transições relacionadas a uma rede de Petri
$T_j$	$j$ -ésima transição em uma rede de Petri
$u_k$	vetor de disparo associado ao instante $k$

## LISTA DE ABREVIATURAS

API	<i>Application Programming Interface</i>
CPN	<i>Colored Petri Nets</i>
CPN ML	<i>CPN Markup Language</i>
EJS	<i>Easy Java Simulations</i>
ERP	<i>Enterprise Resource Planning</i>
GPS	<i>Global Positioning System</i>
HF	<i>High Frequency</i>
IA	<i>Inteligência Artificial</i>
ID	<i>Identification</i>
IDE	<i>Integrated Development Environment</i>
IEC	<i>International Electrotechnical Commission</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IoT	<i>Internet of Things</i>
IP	<i>Internet Protocol</i>
iPNRD	<i>inverted Petri nets inside RFID distributed database</i>
ISO	<i>International Organization for Standardization</i>
LF	<i>Low Frequency</i>
MES	<i>Manufacturing Execution System</i>
PARNASO	<i>Parque Nacional da Serra dos Órgãos</i>
PC	<i>Personal Computer</i>
PLC	<i>Programmable Logic Controller</i>
PNRD	<i>Petri nets inside RFID distributed database</i>
RFID	<i>Radio Frequency Identification</i>
ROS	<i>Robot Operating System</i>
SAR	<i>Search and Rescue</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
SMA	<i>Sistemas Multiagentes</i>
TCP	<i>Transfer Control Protocol</i>
UHF	<i>Ultra High Frequency</i>
USAR	<i>Urban Search and Rescue</i>
V-REP	<i>Virtual Robot Experimentation Platform</i>
WSN	<i>Wireless Sensor Network</i>

# SUMÁRIO

<b>CAPÍTULO I - INTRODUÇÃO .....</b>	<b>1</b>
1.1. Motivação .....	2
1.2. Justificativa .....	6
1.3. Objetivos .....	7
1.4. Estrutura da Tese .....	8
<b>CAPÍTULO II - FUNDAMENTAÇÃO TEÓRICA E REVISÃO DA LITERATURA.....</b>	<b>9</b>
2.1. Robótica Móvel.....	9
2.1.1. <i>Sistemas multirrobôs</i> .....	12
2.1.2. <i>Sistemas multiagentes</i> .....	15
2.1.3. <i>Simulação em robótica móvel</i> .....	16
2.2. A Tecnologia RFID .....	18
2.2.1. <i>Etiqueta RFID</i> .....	19
2.2.2. <i>Leitor RFID</i> .....	20
2.2.3. <i>Infraestrutura de comunicação e software de aplicação</i> .....	20
2.3. A Tecnologia NFC .....	22
2.4. Redes de Petri.....	23
2.4.1. <i>Redes de Petri clássicas</i> .....	24
2.4.2. <i>Redes de Petri lugar / transição</i> .....	25
2.4.3. <i>Redes de Petri coloridas</i> .....	27
2.4.4. <i>A PNRD</i> .....	27
2.5. Trabalhos Relacionados .....	32
2.5.1 <i>Trabalhos semelhantes quanto à utilização de redes de Petri e RFID</i> .....	34
2.5.2 <i>Trabalhos semelhantes quanto à aplicação de tecnologia RFID em ambientes SAR</i> .....	36
2.5.3 <i>Trabalhos semelhantes quanto à aplicação de tecnologia RFID em robótica móvel</i> .....	38
2.5.4 <i>Trabalhos semelhantes quanto à aplicação de robôs em ambiente SAR</i> .....	38
2.5.5 <i>Trabalhos semelhantes quanto à aplicação de redes de Petri e robôs em ambientes SAR</i> .....	39
<b>CAPÍTULO III - METODOLOGIA E DESENVOLVIMENTO .....</b>	<b>43</b>
3.1. Metodologia .....	43
3.2. Análise do Problema.....	44
3.3. Proposta de Modificação da PNRD .....	47
3.4. Especificação do Sistema Multirrobô .....	49
3.5. Modelagem dos Agentes em Redes de Petri.....	51
3.5.1. <i>Walker</i> .....	52
3.5.2. <i>GroundBot</i> .....	55
3.5.3. <i>AerialBot</i> .....	59
3.6. Definição da Arquitetura de Controle .....	61
3.6.1. <i>Configuração do CPN Tools</i> .....	64
3.6.2. <i>Configuração do microcontrolador via Arduino IDE</i> .....	67
3.6.3. <i>Configuração do simulador V-REP</i> .....	68
<b>CAPÍTULO IV - IMPLEMENTAÇÃO E TESTES.....</b>	<b>73</b>
4.1. Implementação em CPN Tools .....	73

4.2.	Implementação em V-REP .....	78
4.3.	Implementação experimental.....	81
4.4.	Testes Funcionais e Discussão .....	83
4.4.1.	<i>Caso 1: Navegação do agente Walker</i> .....	83
4.4.2.	<i>Caso 2: identificação aérea e apoio terrestre</i> .....	88
4.4.3.	<i>Caso 3: patrulha terrestre e apoio aéreo</i> .....	93
<b>CAPÍTULO V - CONCLUSÕES E PROPOSTAS DE TRABALHOS FUTUROS .....</b>		<b>96</b>
5.1.	Contribuições da Tese.....	96
5.2.	Limitações da Solução Proposta.....	97
5.3.	Propostas de Trabalhos Futuros.....	98
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>		<b>100</b>
<b>APÊNDICE A - DETALHE DAS TRANSIÇÕES DE SUBSTITUIÇÃO DAS REDES DE PETRI DE ALTO NÍVEL DESENVOLVIDAS EM CPN TOOLS .....</b>		<b>118</b>
A.1.	Coleção Completa das Redes de Petri de Alto Nível Associadas ao Comportamento do Agente <i>AerialBot</i> .....	118
A.2.	Coleção Completa das Redes de Petri de Alto Nível Associadas ao Comportamento do Agente <i>GroundBot</i> .....	122
A.3.	Coleção Completa das Redes de Petri de Alto Nível Associadas ao Comportamento do Agente <i>Walker</i> .....	126

# CAPÍTULO I

## INTRODUÇÃO

Busca e salvamento, ou *search and rescue* (SAR) em inglês, consiste em um conjunto de atividades amplamente planejadas, treinadas e executadas por grupos responsáveis por procurar, identificar e resgatar pessoas ou animais perdidos, feridos ou em situação de risco. A existência de vítimas que necessitem de apoio especializado está normalmente associada a ocorrência de desastres naturais. Por outro lado, regiões com alta circulação de turistas e praticantes de atividades ao ar livre apresentam risco de desaparecimento de pessoas, como ocorre em trilhas e travessias em florestas, montanhas, campos, desertos, rios, lagos e mares.

Existem diversas ferramentas utilizadas para auxiliar buscas e salvamentos em trilhas, destacando-se o sistema internacional de satélites para busca e salvamento Cospas-Sarsat (KING, 2013). Entretanto, em regiões sem rede celular ou sem garantia de comunicação via satélite, as chances de salvamento são comprometidas. Nestes casos, a correta integração de recursos científicos e tecnológicos, associada a uma precisa descrição formal do ambiente de análise, pode apontar alternativas para a solução de problemas de busca e salvamento.

Apesar de a engenharia mecânica ser uma área amplamente técnica e especializada, este estudo revela seu papel substancial no auxílio a problemas sociais. Recursos científicos e tecnológicos como robótica móvel, redes de Petri e identificação por rádio frequência possuem características interessantes, porém pouco exploradas no que tange sua integração e correlação, para auxiliar operações de busca e salvamento de pessoas perdidas ou comprometidas fisicamente. Neste sentido, este trabalho avalia do ponto de vista acadêmico a associação de redes de Petri de alto nível e identificação por rádio frequência aplicadas ao controle de robôs móveis em cenários característicos de busca e salvamento em trilhas e travessias.

## 1.1. Motivação

Entre os desastres naturais ou tecnológicos recentes de grande proporção, estão os deslizamentos de terra na região serrana do Rio de Janeiro no início de 2011, o vazamento de óleo na bacia de Campos em novembro de 2011 e o rompimento da barragem de rejeitos de minério no distrito de Bento Rodrigues em novembro de 2015. Segundo o Relatório de danos materiais e prejuízos decorrentes de desastres naturais no Brasil: 1995 - 2014 (CEPED UFSC, 2016), desastres naturais custaram ao Brasil R\$ 182 bi em 20 anos, uma média de R\$ 800 mi por mês. Os prejuízos causados por secas, estiagens, inundações e enxurradas que atingiram o país entre 1995 e 2014 apontam uma necessidade de estratégias governamentais, focadas na especialização de profissionais e no desenvolvimento tecnológico, para gestão e prevenção de riscos e desastres.

Além da prevenção dos perigos associados a desastres naturais, é preciso atenção com outros tipos de risco. De acordo com Lack et al. (2012), de 1998 a 2011, o *Rocky Mountain Rescue Group* atendeu a 1857 incidentes SAR envolvendo 2198 vítimas nos Estados Unidos da América. Destes incidentes, 345 enquadram-se em casos ocorridos em escaladas (*climbing*) com 428 vítimas.

No Brasil, a prática do *trekking* tem aumentado significativamente, principalmente em travessias como a da Serra Fina, na divisa tríplice entre os estados de Minas Gerais, São Paulo e Rio de Janeiro, e a da Serra dos Órgãos, entre as cidades de Petrópolis e Teresópolis no estado do Rio de Janeiro. Entretanto, muitos praticantes nem sempre são orientados ou estão dispostos a solicitar ajuda de especialistas e guias turísticos, o que acaba aumentando a probabilidade de desaparecimento.

De acordo com o Corpo de Bombeiros Militares do estado de São Paulo, de 2013 para 2014, houve um aumento de 17,30% no número de ocorrências de pessoas perdidas em matas no Estado de São Paulo e um aumento de 21,42% na Baixada Santista. Em 2013 foram 52 ocorrências estaduais, sendo 14 no litoral. Já em 2014, o número estadual saltou para 61 e foram registrados 17 casos de pessoas perdidas na Baixada Santista (ROSSI, 2015). A Tab. 1.1 apresenta diversas manchetes de notícias sobre pessoas perdidas em trilhas localizadas na região centro-sul do Brasil.

A correta integração de recursos científicos e tecnológicos, associada a uma precisa descrição formal do ambiente de análise, pode apontar alternativas para a solução de problemas de busca e salvamento em trilhas e travessias.



Tabela 1.1 - Notícias sobre pessoas perdidas em trilhas do centro-sul brasileiro.

<b>Data</b>	<b>Região</b>	<b>Notícia</b>	<b>Fonte</b>
13/07/2017	<i>Parque Nacional do Caparaó (MG/ES)</i>	Professor desaparece durante trilha e é encontrado com vida após quatro dias	R7 (2017) ( <a href="http://bit.ly/2vIHm8F">http://bit.ly/2vIHm8F</a> )
26/12/2016	<i>Salto do Indaiá (GO)</i>	Corpo de professor que estava desaparecido é encontrado	(PINHEIRO; MEDEIROS, 2016)
18/07/2016	<i>Serra do Mar (SP)</i>	“Bombeiros buscam quatro pessoas perdidas em trilha na serra do mar”	Folha de S. Paulo (2016)
19/03/2016	<i>Serra Fina (MG/SP/RJ)</i>	“Guia local e bombeiros resgatam dois jovens na serra fina”	Borges (2016)
04/04/2015	<i>Serra Dona Francisca (SC)</i>	“Grupo de resgate em montanha localiza casal perdido na Serra Dona Francisca”	Diário Catarinense (2015)
19/07/2012	<i>Serra dos Órgãos (RJ)</i>	“Estudante é resgatado após dois dias perdido na Travessia Petrópolis – Teresópolis”	Azevedo (2012)
06/07/2011	<i>Serra do Mar (SP)</i>	“Homem que foi procurar grupo perdido na Serra do Mar, em SP, também se perde”	SPTV (2011)
24/05/2010	<i>Serra do Mar (SP)</i>	“Grupo de pessoas é resgatado na Serra do Mar”	Junior (2010)
02/11/2009	<i>Represa Billings (SP)</i>	“Grupo de pessoas é resgatado por bombeiros após passarem a noite perdidos em trilha”	Junior (2009)
27/07/2007	<i>Serra dos Órgãos (RJ)</i>	“Bombeiros resgatam casal no famoso pico de montanha <i>Dedo de Deus</i> ”	Junior (2007a)
11/06/2007	<i>Serra Fina (MG/SP/RJ)</i>	“Casal é resgatado na Serra da Mantiqueira”	Junior (2007b)
18/10/1999	<i>Serra dos Órgãos (RJ)</i>	“Professor cai de uma pedra a 70 m de altura enquanto pedia ajuda e morre na Serra dos Órgãos”	Lobato e Kfuri (1999)

O sistema Cospas-Sarsat, por exemplo, iniciou suas operações em 1982 com uma parceria entre Canadá, Estados Unidos da América, França e Rússia, ajudando a salvar mais de 33 mil vidas (dados até dezembro de 2012). Presente em 41 países, opera com dispositivos de identificação individual, os quais, na iminência de situações de perigo, emitem sinais de rádio que são capturados por satélites, analisados e direcionados aos centros de controle e planejamento de missão, possibilitando o escalonamento de novas operações de busca e salvamento. A Fig. 1.1 revela a evolução do número de eventos registrados e do número de pessoas resgatadas durante o período de 1994 a 2014.

Entretanto, é possível deparar-se com regiões remotas sem rede celular ou sem garantia de comunicação via satélite, nas quais o salvamento é um grande desafio, uma vez que a falta de conhecimento do local onde se encontra(m) a(s) vítimas(s) compromete consideravelmente a chegada do pessoal de apoio em tempo hábil para o pronto atendimento e resgate. Neste sentido, acredita-se que tais informações devam ser consideradas na especificação dos cenários característicos de busca e salvamento com multirrobôs, sejam em ambientes de simulação ou em ambientes para testes controlados.

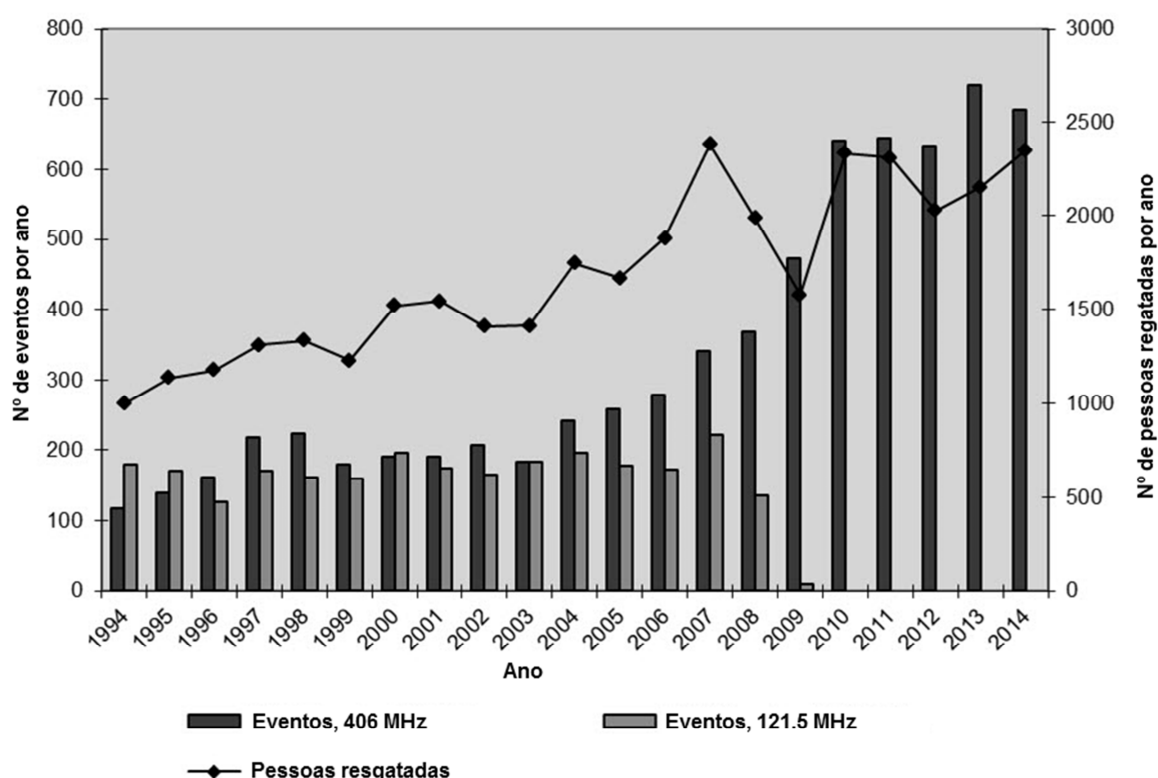


Figura 1.1 - Gráfico com número de eventos SAR e número de pessoas resgatadas por ano (1994-2014) com o sistema Cospas-Sarsat; adaptado de King (2013).

Este estudo revela, em parte, o papel substancial da engenharia mecânica no auxílio a problemas sociais. No caso específico de busca e salvamento de desaparecidos em trilhas e travessias, a utilização de recursos científicos e tecnológicos associados à engenharia mecatrônica pode auxiliar a investigação de cenários e propostas de novos modelos estratégicos para o resgate de pessoas perdidas.

Uma técnica amplamente difundida nas áreas computacionais de modelagem de eventos discretos e sistemas baseados em conhecimento é a rede de Petri (PETRI, 1962), que consiste em uma técnica clássica de modelagem gráfica com fundamentação matemática, tradicionalmente aplicada à modelagem de software, modelos organizacionais e comerciais, cadeias de suprimento, sistemas de manufatura flexível e sistemas de controle discreto. Redes de Petri de alto nível combinam as capacidades algébricas e gráficas das redes Petri com as capacidades de uma linguagem de programação de alto nível.

Apesar de ser bem fundamentada e possuir recursos interessantes para a investigação da utilização de robôs móveis em operações de busca e salvamento, a utilização das redes de Petri em aplicações desta natureza ainda é um desafio.

A identificação por rádio frequência (RFID), por sua vez, tem forte aplicação na rastreabilidade e controle de movimentação. Sistemas RFID são compostos, basicamente, por dois dispositivos: leitores e etiquetas eletrônicas, normalmente denominadas *tags*. Com esta tecnologia, é possível conferir identificação única e informação extra a objetos de interesse. E, por meio de leitores conectados a aplicações específicas, pode-se manipular um ambiente de informações fisicamente distribuídas. Tais características credenciam a tecnologia RFID como uma interessante ferramenta na análise de ambientes SAR.

Uma tecnologia que aproxima as duas anteriores é a rede de Petri inserida em base de dados RFID, ou *Petri nets inside RFID distributed database* (PNRD) (TAVARES; SARAIVA, 2010), uma estrutura formal de dados que integra redes de Petri e os componentes RFID. Esta abordagem permitiu que processos modelados em redes de Petri pudessem ter sua lógica de operação fisicamente distribuída em etiquetas e leitores.

A PNRD foi concebida para auxiliar sistemas logísticos e de manufatura, onde o foco está na identificação e monitoramento de agentes passivos (itens comerciais, peças e produtos, por exemplo). Por abordar situações relativamente simples, redes de Petri elementares foram a base desta abordagem clássica.

Sistemas SAR, por outro lado, requerem a identificação de agentes ativos (usuários e socorristas, por exemplo). Ao exigir modelos comportamentais mais complexos, comparados aos agentes passivos da PNRD clássica, há uma tendência de que redes de Petri elementares se tornem limitadas para a modelagem de agentes SAR.

A diferença na passividade dos agentes também revela a necessidade de reavaliação da estrutura e distribuição física da lógica de operação na PNRD. Para isso, propõe-se uma modificação da PNRD, a qual executa uma inversão dos locais de armazenamento dos dados da rede de Petri nos componentes RFID, quando comparado com a proposta original. A nova estrutura proposta é aqui denominada PNRD invertida ou iPNRD.

Após a motivação para o desenvolvimento deste trabalho, as próximas seções abordam os fatores técnicos e sociais que o justificam, os objetivos gerais e específicos que esperam-se alcançar com a sua conclusão e um detalhamento do conteúdo dos demais capítulos deste documento.

## 1.2. Justificativa

Encontram-se na literatura diversos trabalhos relacionados. Curiac (2016), por exemplo, ao pesquisar o estado da arte atual na integração de sistemas robóticos móveis com redes de sensores e atuadores sem fio, define um novo conceito integrado voltado para a percepção e controle de ambientes remotos ou sistemas complexos e distribuídos. O autor afirma tratar-se de um campo de pesquisa emergente e destaca necessidades críticas de pesquisa que ainda demandam maior investigação, tais como controle distribuído em tempo real; tomada de decisão; gestão de redundância; estratégias de enxames de robôs em redes de sensores sem fio; segurança, privacidade e questões legais.

Já Grieco et al. (2014) descrevem aplicações robóticas auxiliadas por internet das coisas, ou *Internet of Things* (IoT), como uma nova área de pesquisa e descreve suas implicações tecnológicas, levanta o estado da arte e apresenta os problemas em aberto. Uma revisão da literatura sobre os principais campos de pesquisa envolvidos reforça quão multidisciplinar e heterogêneo é o conhecimento necessário para lidar com esse tópico. Os autores ponderam que, para explorar plenamente o potencial da tecnologia nos próximos anos, é necessário um esforço sólido em protocolos e *design* das aplicações para levar em consideração questões relacionadas à tecnologia de comunicação de curto alcance, serviços baseados em semântica, informações de rede centralizada e segurança. Porém, assim como observado em Curiac (2016), nenhum estudo teórico ou prático é apresentado.

Outro ponto ao qual se justifica a investigação proposta nesta tese diz respeito à possibilidade de certas atividades ocorrerem com maior autonomia e sem depender exclusivamente de conexão de rede externa ou de um sistema de posicionamento global, ou *global positioning system* (GPS) em inglês. Uma alternativa consiste na utilização de robótica móvel e identificação por radiofrequência.

Diversas iniciativas vêm ocorrendo no sentido de se integrar robôs móveis em operações de busca e salvamento, como são os casos de (JENNINGS; WHELAN; EVANS, 1997), (ZLOT; STENTZ, 2006), (BALAKIRSKY et al., 2007) e (BAXTER et al., 2009). Do ponto de vista da modelagem do sistema, diversos estudos reportam a utilização de robôs móveis e redes de Petri como ferramentas para auxiliar a modelagem e análise de operações SAR. Por exemplo, uma técnica para modelar o comportamento robótico em um ambiente urbano de busca e salvamento que utiliza redes de Petri elementares para modelar o comportamento dos atores do sistema (ANDREWS; ABDELGAWAD; GARIO, 2016), e uma análise qualitativa do processo de triagem em operações de salvamento em zonas de desastres utilizando redes de Petri estocásticas (JIANJIE et al., 2012).

Porém, há uma clara lacuna de trabalhos que propõem ou verificam a utilização da tecnologia multirrobô associada à tecnologia RFID em operações SAR. Integrar estas tecnologias é fundamental para uma melhor investigação dos cenários de busca e salvamento em regiões sem garantia de comunicação e com pontos de energia escassos.

### **1.3. Objetivos**

Focando em operações de busca e salvamento em trilhas e travessias, esta tese apresenta a concepção de um sistema multirrobô que integra tecnologia RFID e redes de Petri de alto nível.

O principal objetivo deste projeto consiste em estruturar uma arquitetura de controle comportamental para robôs móveis baseada em redes de Petri de alto nível integrada com a iPNRD para auxiliar nas operações SAR. O trabalho avalia esta proposta em um estudo de caso específico (baseado em uma simplificação da travessia Petrópolis-Teresópolis).

Destacam-se, ainda, os seguintes objetivos específicos:

- 1) Apresentar o conceito da PNRD e a proposta de modificação denominada iPNRD;
- 2) Projetar o sistema multirrobô aplicado ao ambiente de análise;
- 3) Modelar o comportamento dos agentes em redes de Petri de alto nível
- 4) Desenvolver simulação computacional do ambiente de testes;
- 5) Definir cenários para testes e ensaios funcionais;
- 6) Realizar testes e ajustes necessários;
- 7) Avaliar o comportamento do sistema quanto às suas características, limitações e propostas de melhorias.

Por se tratar de um assunto amplo e recente, a integração desta tecnologia é abordada do ponto de vista da pesquisa exploratória, focando no aprofundamento do comportamento do sistema, levantamento de hipóteses, definição de métodos e procedimentos e na geração de tecnologia nacional. As soluções propostas são verificadas em um estudo de caso específico em ambiente de simulação e em um ambiente estruturado de testes, parcialmente apresentado em Fonseca e Tavares (2017) e Fonseca et al. (2017).

#### **1.4. Estrutura da Tese**

Os capítulos que descrevem o desenvolvimento, os resultados e as conclusões estão organizados neste documento da forma descrita a seguir.

No capítulo II, introduzem-se conceitos fundamentais para o entendimento do projeto. Além da robótica móvel na seção 2.1, abordam-se a tecnologia RFID na seção 2.2, a tecnologia NFC na seção 2.3 e as redes de Petri na seção 2.4. Por fim, trabalhos semelhantes encontrados na literatura recente são apresentados analisados na seção 2.5.

O sistema proposto, com a definição da estrutura do ambiente e dos agentes, é apresentado no capítulo III. Inicialmente, na seção 3.1, define-se a metodologia adotada, seguida pelos requisitos do sistema e as etapas necessárias para a implementação do projeto. Na sequência, a seção 3.2 aborda a análise do problema, seguida pela proposta de modificação da PNRD na seção 3.3, pela especificação do sistema multirrobô na seção 3.4, modelagem dos agentes na seção 3.5 e definição da arquitetura de controle na seção 3.6.

O capítulo IV reserva detalhes da implementação e testes do estudo de caso utilizado para aplicação do sistema proposto. A seção 4.1 descreve a implementação dos modelos em CPN Tools, enquanto a implementação em V-REP é descrita na seção 4.2 e a parte experimental na seção 4.3. A seção 4.4, por sua vez, detalha três cenários investigados em V-REP e um cenário avaliado com os dispositivos embarcados nos robôs móveis.

O capítulo V pondera as principais conclusões, limitações da solução proposta, sugestões de trabalhos futuros e contribuições associadas a esta tese. Por fim, apresentam-se as referências bibliográficas utilizadas e apêndices desenvolvidos.

## **CAPÍTULO II**

### **FUNDAMENTAÇÃO TEÓRICA E REVISÃO DA LITERATURA**

Este capítulo expõe uma breve revisão bibliográfica a respeito dos conceitos essenciais para o desenvolvimento deste projeto. Inicialmente, a seção 2.1 apresenta os principais conceitos relacionados à robótica móvel, abordando também noções de sistemas multirrobôs, sistemas multiagentes e simulação em robótica móvel, enquanto as tecnologias RFID são discutidas nas seções 2.2 e 2.3, respectivamente. A seção 2.4 aborda conceitos e formulações características das redes de Petri, passando por redes de Petri clássicas, redes de Petri lugar/transição, redes de Petri coloridas e pela PNRD. Finalmente, através da apresentação de trabalhos relacionados, a seção 2.5 levanta o estado da arte sobre a utilização das tecnologias multirrobô e RFID em aplicações SAR e aponta as principais lacunas desta linha de pesquisa.

#### **2.1. Robótica Móvel**

A robótica possui forte característica multidisciplinar, envolvendo conceitos fundamentais como o estudo do comportamento cinemático e dinâmico da estrutura, análise e tratamento de sinais provenientes de sensores e estratégias de controle para habilitação dos atuadores. Robôs industriais executam tarefas repetitivas como soldagem e pintura com alto nível de velocidade e precisão, no entanto, trabalham em um espaço limitado, geralmente presos a uma base fixa. A robótica móvel, por sua vez, além dos conceitos fundamentais, requer estratégias de localização, mapeamento e navegação.

Para Parker (1994) a questão chave para o desenvolvimento de robôs móveis é seu alto potencial para redução da necessidade de humanos em tarefas perigosas, tais como limpeza de lixo tóxico, desativação de usinas de energia nuclear, exploração espacial, missões de busca e salvamento, tarefas de segurança, vigilância e reconhecimento.



A concepção de robôs móveis exige a integração de diversos campos de conhecimento. Para Siegwart e Nourbakhsh (2004) esta interdisciplinaridade se dá porque, para solucionar os problemas de locomoção, os desenvolvedores de robôs móveis devem entender de mecanismos, cinemática, dinâmica e teoria de controle; para criar sistemas de percepção robustos requer-se áreas de conhecimento especializadas como análise de sinais e visão computacional para empregar corretamente uma grande variedade da tecnologia de sensores; e, por fim, localização e navegação demandam conhecimento de algoritmos computacionais, teoria da informação, inteligência artificial e teoria da probabilidade. A Fig. 2.1 apresenta um esquema de controle para robôs móveis baseado em Siegwart e Nourbakhsh (2004).

Os mecanismos de locomoção permitem que um robô se desloque por um determinado ambiente. Siegwart e Nourbakhsh (2004) destacam que existem pesquisas em robôs com mecanismos de locomoção biologicamente inspirados que permitem ao robô andar, pular, correr, deslizar, patinar, nadar, voar ou rolar. Entretanto, o mecanismo de locomoção mais utilizado em robótica é a roda - uma invenção humana que possui extrema eficiência em terrenos planos e cuja aplicação é mais simples do que a replicação de movimentos biológicos.

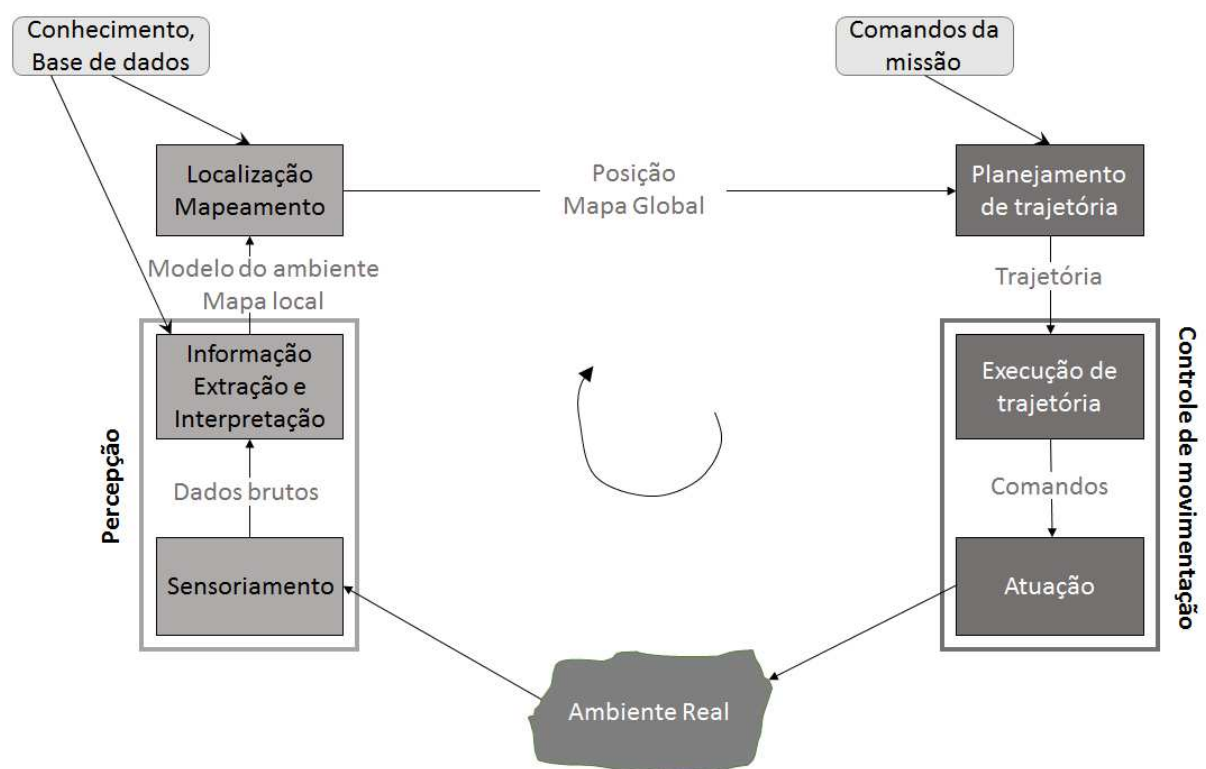


Figura 2.1 - Esquema de controle para sistemas robóticos móveis. Adaptado de Siegwart e Nourbakhsh (2004).

Existem diversos tipos de configurações de rodas em veículos. As mais comuns são duas rodas diferenciais com ou sem pontos de apoio, três rodas omnidirecionais, quatro rodas diferenciais e quatro rodas omnidirecionais. A Tab. 2.1 apresenta algumas configurações de rodas em robôs móveis e sua descrição.

A cinemática é a área de conhecimento que estuda como os mecanismos se comportam. Em robótica móvel é necessário entender o comportamento mecânico do robô tanto para projetar a sua estrutura quanto para criar softwares de controle capazes de garantir sua operação na execução de determinada tarefa. Cada configuração de robô móvel possui um modelo cinemático específico. Apesar da modelagem de robôs móveis não ser objeto de estudo deste projeto, maiores referências e exemplos de modelagem de algumas das configurações presentes na Tab. 2.1 podem ser encontrados na literatura.

Para Siegwart e Nourbakhsh (2004) a navegação é uma das competências mais desafiadoras que um robô móvel necessita. Uma navegação autônoma de sucesso exige sucesso em quatro tarefas: percepção, localização, cognição, e controle de trajetória.

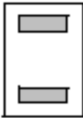
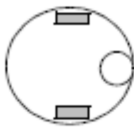
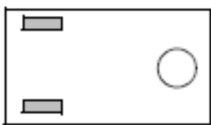
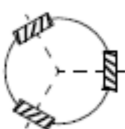
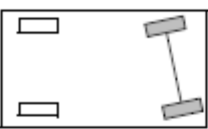
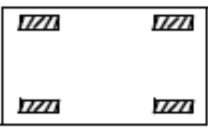
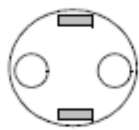
A maneira como o robô percebe o ambiente que o rodeia representa a capacidade de percepção do mesmo. A percepção depende de sensores bem como da aquisição e tratamento dos sinais provenientes destes sensores relacionando-os às características com as quais o robô consiga entender e identificar as condições do ambiente no momento quando as leituras foram realizadas.

O nível de cognição do robô é responsável por decidir a melhor maneira de habilitar os atuadores de forma a executar uma tarefa garantindo segurança e respeitando as restrições impostas pelo ambiente. No caso de robôs móveis, a cognição está diretamente relacionada com uma navegação robusta em termos das condições do ambiente (mapa), da localização do robô e dos objetivos (trajetória) que se almeja alcançar. Dado um conhecimento parcial a respeito do ambiente e uma posição objetivo, a navegação envolve a habilidade do robô agir baseado neste conhecimento e nos valores dos sensores de forma a atingir a posição desejada de forma mais eficiente e confiável.

Há alguns anos, os robôs eram máquinas grandes e de elevado custo, que em muitos casos precisavam estar conectados a computadores através de cabos. Atualmente, é fácil construir robôs móveis com atuadores, sensores e sistemas embarcados de baixo custo. (FABREGAS et al., 2016)

O projeto de robôs móveis pode ser mais profundo em uma competência e básico em outras. Como o intuito deste trabalho é o foco nas estratégias de busca e salvamento em ambientes de trilha e travessias utilizando a tecnologia PNRD invertida, as competências de cinemática, dinâmica, percepção e navegação foram abordadas de forma superficial.

Tabela 2.1 - Configurações de veículos equipados com rodas. Adaptado de Siegwart e Nourbakhsh (2004).

Pontos de contato com o solo	Arranjo	Descrição
2		Duas rodas com acionamento diferencial
3		Duas rodas centralizadas com acionamento diferencial e com um terceiro ponto de contato
3		Duas rodas diferenciais com uma roda esférica na extremidade oposta
3		Três rodas suecas igualmente espaçadas possibilitando movimentação omnidirecional
4		Duas rodas direcionais na parte frontal e duas rodas livres na traseira
4		Quatro rodas omnidirecionais
4		Duas rodas centralizadas com acionamento diferencial e dois pontos de contato

### 2.1.1. Sistemas multirrobo

Uma maneira de projetar sistemas autônomos é através do desenvolvimento de um único robô capaz de atingir objetivos particulares de um dado ambiente. Entretanto, a complexidade de diversos ambientes ou missões pode exigir uma mistura de capacidades robóticas que seria muito trabalhoso para projetar em um único robô. Além disso, restrições temporais podem requerer a utilização de múltiplos robôs trabalhando simultaneamente em diferentes aspectos da missão de modo a atingir o objetivo desejado. Neste caso, a construção de um time de robôs heterogêneos, que podem trabalhar juntos, torna-se mais simples e barata do que o projeto e desenvolvimento de um único robô, necessário para atingir o mesmo objetivo.

A execução de tarefas que exigem a operação de mais de um robô, requer técnicas que permitam que os agentes do sistema possam agir de forma coordenada e colaborativa para obtenção de um objetivo comum. O interesse no desenvolvimento de sistemas de controle colaborativo é bastante amplo. Um dos primeiros estudos desta área da robótica data de 1994 quando Lynne Parker apresentou sua tese de doutorado intitulada “Heterogeneous Multi-Robot Cooperation”. Desde então, diversas iniciativas vêm ocorrendo na pesquisa sobre sistemas robóticos colaborativos. Atualmente, a maioria dos países desenvolvidos ou emergentes possui algum tipo de programa de desenvolvimento de pesquisa em robótica móvel. Diversos centros de pesquisas ao redor do mundo unem esforços para desenvolver técnicas de cooperação entre times de robôs, como é o caso, por exemplo, do *AI and Robotics Lab at the University of New Hampshire* (EUA), do *AIRLab, Politecnico di Milano* (Itália), do *Australian Centre for Field Robotics* (Austrália), do *Information-oriented Control, Technische Universitaet Muenchen* (Alemanha), do *Multi-robot Systems Lab* (EUA), do *IRIDIA Lab, Université Libre de Bruxelles* (Bélgica) e do *GRASP Laboratory, University of Pennsylvania* (EUA). Dentre os laboratórios de pesquisa nacionais engajados no desenvolvimento de sistemas robóticos colaborativos, destacam-se o Laboratório de Robótica e Visão Computacional da Universidade Federal de Minas Gerais, o Centro Universitário da FEI, o Grupo de Controle, Automação e Robótica da Universidade Federal do Rio Grande do Sul e a Escola de Engenharia de São Carlos da USP.

Pesquisadores interessados em sistemas multirrobôs representam uma comunidade diversificada, já que são necessárias várias competências neste domínio, que vão desde sistemas de controle, projeto mecânico, percepção, comunicação coordenação e cooperação. Devido à forte característica multidisciplinar e ao grande interesse por parte de diversos pesquisadores ao redor do mundo, a Sociedade de Robótica e Automação do IEEE (*IEEE Robotics and Automation Society*) criou, em 2014, o Comitê Técnico em Sistemas Multirrobôs (*Technical Committee on Multi-Robot Systems*) com o objetivo de identificar e acompanhar as características, problemas e resultados da pesquisa com sistemas multirrobôs em seus diversos domínios.

De acordo com Yan, Jouandeau e Cherif (2013), no campo da robótica móvel o estudo de sistemas multirrobôs cresceu significativamente nos últimos anos. Com um progresso considerável nos problemas básicos envolvendo o controle de um único robô, na década de 80 alguns pesquisadores iniciaram o estudo da coordenação de sistemas compostos por mais de um robô. Desde então uma série de projetos foram realizados com sucesso, como é caso do ACTRESS (ASAMA et al., 1989), ALLIANCE (PARKER, 1994), MURDOCH (GERKEY; MATARIC, 2004), ASyMTRE (TANG; PARKER, 2005) e (AYANIAN, 2011).

Ao comparar a utilização de um único ou um grupo de vários robôs, Yan, Jouandeau e Cherif (2013) citam as seguintes vantagens de se utilizar um sistema multirrobô:

- possui uma melhor distribuição espacial;
- pode alcançar melhor performance. A métrica de performance pode ser o tempo total necessário para completar uma tarefa (BURGARD et al., 2000) ou o consumo de energia dos robôs (WAWERLA; VAUGHAN, 2010);
- é mais robusto, uma vez que é possível compartilhar informações e realizar fusão de dados. Por exemplo, múltiplos robôs podem localizar uns aos outros mais eficientemente se eles compartilharem informações sobre sua posição sempre que perceberem uns aos outros (FOX et al., 2000) (ROUMELIOTIS; BEKEY, 2002) (MADHAVAN; FREGENE; PARKER, 2002);
- pode ter um menor custo. Robôs menos complexos tendem a ser mais simples de programar e mais baratos para construir do que um único robô capaz de realizar uma tarefa complexa;
- pode exibir maior confiança, flexibilidade, escalabilidade e versatilidade. Robôs com habilidades diversas podem se unir para lidar com tarefas complexas e, em sistemas compostos por diversos robôs, ainda que um ou mais falhem, a conclusão da tarefa, mesmo que limitada, ainda é possível.

Um sistema multirrobô pode ser homogêneo ou heterogêneo. Em times de robôs homogêneos, as capacidades dos robôs individuais são idênticas (mesmo que a estrutura física seja diferente). Diversos trabalhos consideraram o uso de grupos de robô homogêneos, incluindo (VAUGHAN et al., 2002), (BAXTER et al., 2009) e (DAHL; MATARIĆ; SUKHATME, 2009). Em times de robôs heterogêneos as capacidades de cada robô individual são diferenciadas, sendo que cada um pode ser especializado para uma tarefa específica. Alguns trabalhos que consideraram o uso de robôs heterogêneos incluem (GERKEY; MATARIĆ, 2004), (BICCHI et al., 2008) e (PEI; MUTKA, 2012). De acordo com Yan, Jouandeau e Cherif (2013), em geral, sistemas com robôs heterogêneos são mais complexos do que aqueles com robôs homogêneos, uma vez que o planejamento de tarefas se torna mais difícil.

Um ambiente composto por vários robôs possui um comportamento coletivo. Segundo Yan, Jouandeau e Cherif (2013), comportamento coletivo inclui comportamentos cooperativo e competitivo. Para Barnes e Gray (1991) cooperação é a interação entre robôs que trabalham com um interesse comum. Dentre exemplos de cooperação multirrobô incluem-se os trabalhos sobre localização multirrobô (NERURKAR; ZHOU; ROUMELIOTIS, 2011) (PROROK; BAHR; MARTINOLI, 2012), exploração multirrobô (PEI; MUTKA, 2012) (FAIGL;

KULICH; PREUCIL, 2012), operações de busca e salvamento com múltiplos robôs (ZLOT; STENTZ, 2006) (BALAKIRSKY et al., 2007) e transporte com múltiplos robôs (KUBE; BONABEAU, 2000) (YAN; JOUANDEAU; CHERIF, 2012).

As situações nas quais os robôs competem entre si para melhor satisfazer interesses próprios são características de sistemas competitivos. Exemplos típicos de sistemas multirrobôs competitivos são jogos de xadrez e aplicações onde os robôs competem inclusive pelo espaço ocupado como na RoboCup (futebol de robôs) - embora robôs do mesmo time cooperem entre si para atingir o mesmo objetivo comum: vencer a partida (CANDEA et al., 2001) (CASTELPIETRA et al., 2000). A definição dos mecanismos de comunicação e de tomada de decisão no projeto do sistema multirrobô levarão em conta o tipo de comportamento presente nos robôs considerados.

O conceito de sistemas multirrobôs tem relação direta com outra área da ciência da computação, denominada sistemas multiagentes. A próxima subseção aborda noções básicas, mas essenciais para o entendimento de um sistema multirrobô como um sistema multiagente.

### *2.1.2. Sistemas multiagentes*

Tanto a Ciência da Computação como a Inteligência Artificial (IA) têm buscado inspiração em outras áreas do conhecimento como forma de se conceber sistemas que se aproximam da realidade. A orientação a objetos, por exemplo, está fundamentada na Matemática; Psicologia e Lógica são a base para a representação de conhecimento e raciocínio; e os sistemas evolutivos são inspirados na Biologia. Por outro lado, existe uma linha da IA que defende a concepção de sistemas com propriedades coletivas inspiradas no comportamento social. Estes sistemas, que buscam inspiração na Sociologia, são denominados Sistemas Multiagentes (SMA).

Ao contrário das técnicas tradicionais de IA, que buscam compreender e simular o comportamento individual isoladamente, os SMA movem sua atenção para a interação entre as entidades que formam o sistema e para a sua organização. Este paradigma é motivado pela observação de alguns sistemas naturais, nos quais se percebe o surgimento de um comportamento inteligente a partir da interação entre seus elementos. Uma rede de neurônios ou uma colônia de formigas são bons exemplos para mostrar que a coletividade possui características que não podem ser reduzidas aos componentes que a formam, mas que são essenciais para o comportamento bem adaptado que tais sistemas apresentam.

Neste sentido, para Wooldridge e Jennings (1995) agentes inteligentes podem ser encarados como módulos de software independentes que apresentam quatro características chave: autonomia, habilidade social, reatividade e pró-atividade.

O SMA, por sua vez, foi definido por Lesser et al. (2000) como sendo um sistema em que dois ou mais agentes interagem ou trabalham em conjunto de forma a desempenhar um determinado conjunto de tarefas ou satisfazer um conjunto de objetivos. Para Wooldridge (2002), os sistemas multiagentes são sistemas compostos por múltiplos elementos computacionais, conhecidos como agentes, que interagem entre si. Para Hubner (2003), a área de SMA estuda o comportamento de um grupo organizado de agentes autônomos que cooperam na resolução de problemas que estão além das capacidades de resolução de cada um individualmente.

Duas propriedades aparentemente contraditórias são fundamentais para os SMA (Briot; Demazeau, 2001):

- a autonomia dos agentes
- a sua organização

Ser autônomo significa que a existência de um agente independe dos demais ou da tarefa a ser executada. Por outro lado, a organização estabelece restrições ao comportamento dos agentes, visando estabelecer um comportamento grupal coeso. Segundo Hubner (2003), muitas das propriedades dos SMA vêm do equilíbrio destes dois pontos.

### *2.1.3. Simulação em robótica móvel*

Simuladores são essenciais quando se desejam testar teorias, ideias e arquiteturas antes da implementação em experimentos reais. De acordo com Peralta et al. (2016), no campo da robótica os simuladores têm maior importância porque robôs de alta performance são caros e normalmente escassos nos ambientes de pesquisa. Isto gera competitividade de recursos nas ocasiões em que os pesquisadores desejam realizar seus testes e experimentos ao mesmo tempo.

Para Peralta et al. (2016), os simuladores de sistemas robóticos mais utilizados atualmente são ARGoS (PINCIROLI et al., 2011), Webots (MICHEL, 2004), Gazebo (KOENIG; HOWARD, 2004), RFCSIM (FABREGAS et al., 2014) e V-REP (ROHMER; SINGH; FREESE, 2013).

O ARGoS consiste em um simulador projetado para simular enxames de robôs em tempo real. Sua abordagem é modular e permite que o usuário adicione recursos customizados em um ambiente virtual. Todos os componentes do ARGoS são *plugins* (modelos de robôs, sensores, atuadores, regras de comportamento dinâmico, etc.). Os usuários podem ampliar ou substituir os *plugins* existentes para criar robôs e implementar as simulações. A plataforma ARGoS está disponível para Linux e MacOS, mas não para Windows (PINCIROLI et al., 2011).



O Webots é um ambiente de desenvolvimento utilizado para modelar e simular robôs móveis. Uma vasta lista de sensores e atuadores encontra-se disponível para construir os robôs em um mundo virtual. Os controladores dos robôs podem ser programados internamente na própria interface de desenvolvimento – *integrated development environment* (IDE) – ou através de ambientes de desenvolvimento externos. Os programas do controlador também podem ser transferidos para robôs reais (GUYOT et al., 2011). A plataforma Webots está disponível para Linux, MacOS e Windows.

O RFCSIM consiste em um simulador desenvolvido em *Easy Java Simulations* (EJS). Seu maior propósito é prover experimentos para sistemas multirrobôs para educação em engenharia. Sua abordagem foca em algoritmos de desvio de obstáculos e controle de posição de robôs com rodas diferenciais. O usuário pode modificar tanto a lei de controle quanto o algoritmo de desvio de obstáculos. O RFCSIM está disponível para todas as máquinas que suportam *Java Virtual Machine* (FABREGAS et al., 2014).

O Gazebo é um simulador amplamente conhecido, principalmente por ser o simulador padrão do *Robot Operating System* (ROS). Segundo Nogueira (2014), nos últimos anos o ROS surgiu como um padrão para arquitetura de software em robótica. Vários robôs comerciais já possuem compatibilidade com ROS e Gazebo e existe a possibilidade de modelos particulares serem inseridos pelo usuário. O Gazebo é um projeto totalmente *open-source*, mas que roda apenas em sistemas operacionais Linux.

O V-REP consiste em uma plataforma versátil para a criação de simulações 3D em períodos de tempo relativamente curtos (ROHMER; SINGH; FREESE, 2013). Além de diversos exemplos, modelos de robôs, sensores e atuadores para a criação de ambientes virtuais estarem disponíveis, a plataforma permite a inserção de novos objetos com definição de propriedades dinâmicas. Com isso, modelos existentes podem ser modificados e modelos de robôs personalizados podem ser inseridos, tornando simples a criação de cenários de simulação customizados.

Farias et al. (2017) destacam que a IDE é baseada em uma arquitetura de controle distribuído, onde cada objeto do cenário pode ser individualmente controlado via um *script* de comportamento, um *plug-in*, um nó do ROS (ROS node), uma aplicação remota (*remote API*) ou uma solução customizada. Os controladores podem ser escritos em: C/C++, Python, Java, Lua, Matlab, Octave ou Urbi (COPPELIA ROBOTICS, 2018), conferindo maior autonomia para que o desenvolvedor adote sua linguagem de programação preferida.

A Tab. 2.2 apresenta uma breve comparação entre os simuladores ARGoS, Webots, RFCIM e V-REP, apresentada por Peralta et al. (2016) com relação aos parâmetros de usabilidade, escalabilidade, visualização, linguagem de programação, disponibilização de licença gratuita e interatividade com a simulação em execução.

Tabela 2.2 - Breve comparação entre simuladores de robótica, modificado de Peralta et al (2016).

Característica	ARGoS	Webots	RFCSIM	V-REP
Usabilidade	X	✓	✓	✓
Escalabilidade	✓	✓	✓	✓
Visualização	2D	✓	✓	✓
	3D	✓	X	✓
Linguagem de programação	C	X	-	✓
	C++	✓	-	✓
	Java	X	✓	✓
Licença gratuita	X	X	✓	✓
Interatividade	X	✓	✓	✓

Nogueira (2014) comparou os simuladores V-REP e Gazebo em uma aplicação de navegação e desvio de obstáculos com um robô comercial sob o ponto de vista de cinco métricas: Integração com ROS, grau de facilidade para modelar o ambiente, grau de facilidade para modificar e customizar o modelo do robô comercial, capacidade de controlar o ambiente de simulação utilizando uma API com linguagem clássica (C, C++, Java, Python) e quantidade de processamento necessária para a simulação acontecer. O autor conclui que o V-REP consiste em um simulador mais intuitivo e amigável, com uma quantidade de recursos disponíveis superior ao Gazebo. Por sua vez, além de ser totalmente *open source*, o Gazebo é mais integrado à plataforma ROS. Entretanto, carece de ferramentas externas disponíveis nas funcionalidades padrão do V-REP. Além disso, O Gazebo demanda maior processamento da CPU. Assim, de acordo com o autor, o usuário terá maiores chances de implementar e validar suas teorias cognitivas utilizando V-REP, especialmente se o mesmo não possuir experiência prévia em robótica. A Tab. 2.3, resume as conclusões do autor. Nela o simulador que apresentou melhor performance em cada critério está assinalado com um símbolo “✓”.

## 2.2. A Tecnologia RFID

Segundo Coskun, Ok e Ozdenizci (2011), RFID é uma tecnologia que utiliza comunicação via ondas de rádio para trocar dados entre um leitor e uma etiqueta eletrônica, tradicionalmente fixada em um objeto de interesse, principalmente para o propósito de identificação e rastreamento.

Tabela 2.3 - Resultados da comparação entre os simuladores V-REP e Gazebo baseada no trabalho de Nogueira (2014).

Métrica	V-REP	Gazebo
Integração com ROS		✓
Modelagem do ambiente	✓	
Modificações no modelo do robô	✓	
Controle programável via API em linguagem clássica (C, C++, Java, Python)	✓	✓
Consumo de processamento	✓	

De acordo com Ahsan, Shah e Kingston (2010), a tecnologia RFID recebeu forte investimento militar durante a segunda guerra mundial. Desde então, suas funcionalidades permitem a sua aplicação em diversas áreas civis, tais como rastreamento de itens em operações logísticas, movimentação de bagagens em companhias aéreas, controle de acesso em áreas restritas, controle da evasão de itens de alto valor, monitoramento das condições de saúde de pacientes em clínicas e hospitais, controle de pedágios em rodovias, controle de acesso a edificações e ferramentas de pagamento automático.

Um sistema RFID é composto basicamente por três componentes físicos:

- *Tag* ou etiqueta eletrônica;
- Antena;
- Leitor.

### 2.2.1. Etiqueta RFID

No sistema RFID, uma etiqueta é normalmente associada a um objeto de interesse, conferindo-lhe uma identificação única e inequívoca. Segundo Finkenzeller (2010), uma etiqueta (também conhecida como *tag* ou *transponder*) normalmente consiste de um elemento de acoplamento e um *microchip* eletrônico. Etiquetas sem alimentação própria são denominadas passivas, enquanto aquelas com bateria interna são denominadas *tags* ativas.

*Tags* RFID ativas e passivas também são diferenciadas de acordo com algumas características operacionais, como alcance, frequência de operação e quantidade de memória, tal como mostra a Tab. 2.4 apresenta diferenças entre as características operacionais de *tags* ativas e passivas.

Tabela 2.4 - Características operacionais das *tags* ativas e passivas; modificado de Ahsan, Shah e Kingston (2010).

Característica	<i>Tag</i> ativa	<i>Tag</i> passiva
Bateria	Sim	Não
Fonte de energia	Interna	Obtida do sinal RF emitido pelo leitor/antena
Energia disponível	Contínua	Somente no campo do leitor
Potência do sinal requerido para excitar a <i>tag</i>	Muito baixa	Muito alta
Alcance	Até 100 m	Até 5m
Leitura de múltiplas <i>tags</i>	Milhares de <i>tags</i> reconhecidas até 100 m	Poucas <i>tags</i> num raio de 3m
Armazenamento	Até 128 kB de leitura/escrita	Até 1kB de leitura/escrita
Frequência de operação	Micro-ondas UHF	LF, HF, UHF
Tempo de vida	5 a 10 anos	Até 20 anos

### 2.2.2. Leitor RFID

O leitor é responsável por coordenar as operações de leitura e escrita e enviá-las para a antena de maneira a serem executadas de acordo com o protocolo da *tag*. Consiste no *hardware* que controla diretamente a antena, de maneira a modular os comandos de leitura e escrita a serem enviados para as *tags* via RF, e demodular as respostas recebidas pela antena advindas da *tag*. Além disso, pode controlar o acesso e modificar informações em *tags* de interesse.

Os leitores RFID variam quanto ao número de antenas suportadas simultaneamente, método de comunicação, a frequência de operação, a energia de transmissão (consequentemente a distância máxima de leitura), velocidade (número de leituras por segundo) e o tipo de conexão (em geral, USB, *Wi-Fi* ou serial).

### 2.2.3. Infraestrutura de comunicação e software de aplicação

Ahsan, Shah e Kingston (2010) citam outros dois componentes, responsáveis por permitir o acesso, controle e manipulação das informações contidas nas etiquetas: a infraestrutura de comunicação e o software de aplicação. Uma representação esquemática dos componentes de um sistema RFID é apresentada na Fig. 2.2.

De acordo com Dias e Baladei (2012), a frequência de operação define a taxa de transferência de dados entre a etiqueta e o leitor. As três frequências mais utilizadas para o sistema RFID passivo são: baixa frequência (LF), de 125kHz, alta frequência (HF), 13,56 MHz, e ultra alta frequência (UHF), operando na faixa de 860 a 960MHz. Sistemas RFID ativos conseguem atender, também, frequências da ordem de 2,45 a 5,8 GHz, cuja faixa é denominada micro-ondas (FINKENZELLER, 2010).

Cada frequência tem um uso típico e distância máxima de operação. A distância máxima de operação do sistema cresce proporcionalmente à frequência empregada, já que uma onda de maior frequência é capaz de transmitir mais energia, sendo assim um sistema LF apresenta uma distância de operação muito menor que um sistema UHF. Porém, frequências maiores implicam em maiores interferências advindas de materiais metálicos ou de constante dielétrica elevada (líquidos em geral). Deste modo, antenas e tags operando em ambientes contendo objetos metálicos, que apresentam alto índice de reflexão das ondas, podem sofrer perdas significativas se não tratados corretamente.

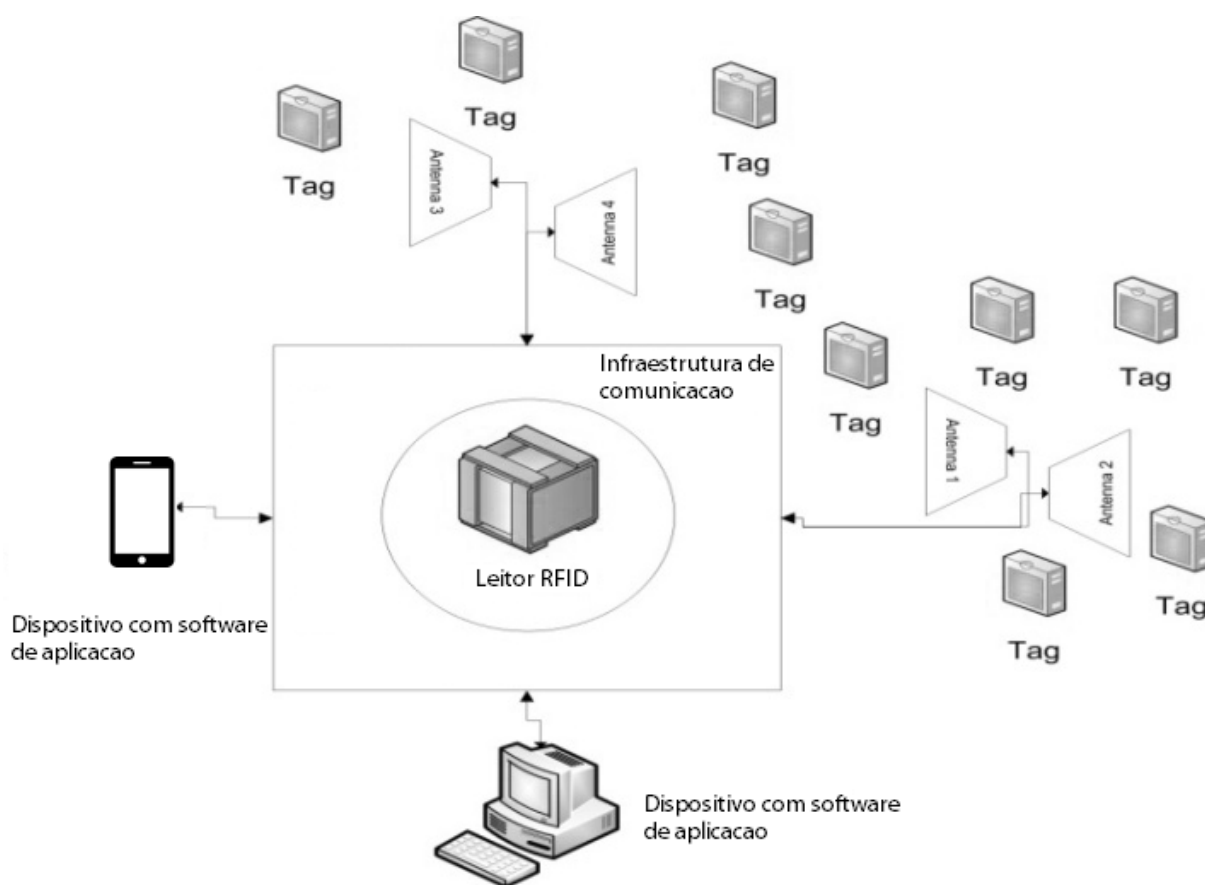


Figura 2.2 - Representação esquemática dos componentes de um sistema RFID. Modificado de Ahsan, Shah e Kingston (2010).

### 2.3. A Tecnologia NFC

Segundo Coskun, Ok e Ozdenizci (2011), a comunicação por campo próximo, ou *near-field communication* (NFC), é uma tecnologia emergente e promissora que terá um enorme impacto no ecossistema financeiro, bem como na tecnologia móvel em todo o mundo. Isso se justifica, pois diversos fabricantes de telefones celulares, operadores de redes móveis, instituições financeiras como bancos e empresas de tecnologia da informação têm investido fortemente em pesquisa e desenvolvimento nesta tecnologia nos últimos anos. Sendo uma tecnologia de comunicação sem fio de curto alcance que potencialmente facilita o uso de bilhões de pessoas em todo o mundo, a tecnologia NFC oferece uma enorme quantidade de casos de uso, incluindo cartões de crédito, chaves de acesso a veículos, hotéis, escritórios e casas, eventualmente integrando todos esses materiais em um único telefone celular.

Want (2011) destaca diversos exemplos de aplicativos que pesquisadores descreveram, prototiparam e publicaram desde meados da década de 1990, vinculando informações virtuais a dispositivos físicos por meio da proximidade

De acordo com Finkenzeller (2010), à primeira vista, NFC não se trata de um sistema RFID, mas uma interface *wireless* para comunicação entre dispositivos, similar ao infravermelho e ao *Bluetooth*. Entretanto, apresenta diversas características comuns aos sistemas RFID.

A transmissão de dados entre duas interfaces NFC usa campos magnéticos de alta frequência na faixa de 13,56 MHz e a distância máxima de comunicação para a transmissão de dados NFC é de 20 cm (FINKENZELLER, 2010). A transmissão de dados entre transponder e leitor é descrita na norma ISO 14443 (ISO/IEC, 2016).

A aplicação de dispositivos NFC depende de plataformas de hardware específicas para os protocolos de comunicação padronizados. O PN532 (NXP, 2007), por exemplo, é um módulo de transmissão altamente integrado para comunicação sem contato a 13,56 MHz, incluindo funcionalidade de microcontrolador com 40 Kbytes de ROM e 1 Kbytes de RAM.

O módulo PN532 NFC RFID (ELECHOUSE, 2013), por sua vez, é um módulo, projetado para “hobbystas”, que simplifica as funcionalidades do PN532 (NXP, 2007), tornando-o mais acessível e fácil de utilizar em plataformas de desenvolvimento, como o Arduino. Os desenvolvedores disponibilizam ainda uma biblioteca de programação que facilita a integração do módulo PN532 NFC RFID (ELECHOUSE, 2013) com plataformas Arduino.

## 2.4. Redes de Petri

Segundo Murata (1989), redes de Petri consistem em um modelo conceitual gráfico e matemático introduzido por Carl Adam Petri em 1962. Para Jensen e Kristensen (2009), as redes de Petri fornecem a base da notação gráfica e das primitivas básicas para modelagem de concorrência, comunicação e sincronização.

Outra definição (PETRINETS, 2018) destaca que as redes de Petri são definidas como uma técnica formal, gráfica e executável para a especificação e análise de sistemas a eventos discretos dinâmicos e concorrentes.

A palavra “formal” significa que a técnica é matematicamente definida. e pode ser usada para fornecer especificações e descrições inequívocas de aplicações.

O uso da palavra “gráfica” quer dizer que a técnica pertence a um ramo da matemática denominado teoria dos gráficos. Ou seja, uma rede de Petri pode ser representada tanto graficamente quanto matematicamente. A capacidade de visualização da estrutura e do comportamento de uma rede de Petri confere maior compreensão do comportamento do sistema modelado.

A palavra “executável” indica que uma rede de Petri pode ser executada e o seu comportamento observado graficamente. Com isso, protótipos desenvolvidos podem testar ideias de forma mais rápida e econômica.

Com uma abordagem gráfica intuitiva e técnicas de análise expressiva, as redes de Petri são adequadas para uma ampla gama de aplicações e cenários, e obtiveram ampla aceitação como técnica de modelagem em áreas como *design* de software e engenharia de controle (REISIG, 2013).

O campo de aplicação abrange uma ampla gama de sistemas técnicos, tais como manufatura, processos de negócios, software e hardware, redes de telecomunicações e sistemas de sinalização, sistemas de defesa, mecatrônica, serviços postais e aviônica para sistemas biológicos e sociotécnicos (ISO/IEC, 2004).

De acordo com Jensen e Kristensen (2009), o desenvolvimento de sistemas concorrentes é particularmente desafiador. Um dos principais motivos é que esses sistemas possuem simultaneidade e não determinismo. Exemplos recorrentes vão desde sistemas de grande escala, nas áreas de telecomunicações e aplicações baseadas em tecnologia da Internet, até sistemas de médio ou pequeno porte, na área de sistemas embarcados.

### 2.4.1. Redes de Petri clássicas

Redes de Petri clássicas são compostas por basicamente quatro tipos de elementos: lugares, transições, arcos e fichas (ABU-TAIEH, M.; EL-SHEIK, 2008). Lugares são tipicamente representados por círculos e transições por traços ou retângulos. Arcos direcionados ligam lugares a transições ou transições a lugares. Um conjunto de fichas, ou *tokens*, representam a marcação da rede, ou seja, o estado em que o sistema modelado se encontra atualmente. Os *tokens* são representados por pequenos círculos preenchidos, e cada lugar pode conter um ou mais *tokens*. A Fig. 2.3 ilustra um exemplo de rede de Petri com cinco lugares  $\{E_0, E_1, E_2, E_3, E_4\}$ , quatro transições  $\{T_0, T_1, T_2, T_3\}$  e um *token* no lugar  $E_0$ .

Bernardinello e De Cindio (1992) classificam as redes de Petri em três níveis distintos:

- Nível 1: redes de Petri caracterizadas por lugares que podem representar valores booleanos, isto é, um lugar é marcado por no máximo um *token* não estruturado.
- Nível 2: redes de Petri caracterizadas por lugares que podem representar valores inteiros, ou seja, um lugar é marcado por uma série de *tokens* não estruturados.
- Nível 3: redes de Petri caracterizadas por Lugares que podem representar valores de alto nível, ou seja, um lugar é marcado por um conjunto de *tokens* estruturados.

Trompedeller (1995) aponta que os sistemas evento/condição, os sistemas 1-*safe* e as máquinas de estado enquadram-se no nível 1, as redes de Petri lugar/transição enquadram-se no nível 2, enquanto no nível 3 enquadram-se as redes de Petri de alto nível com tipos de dados abstratos, como as redes predicado/transição e as redes de Petri coloridas. Noções básicas das redes de Petri lugar/transição e redes de Petri coloridas são abordadas nas seções 2.4.2 e 2.4.3.

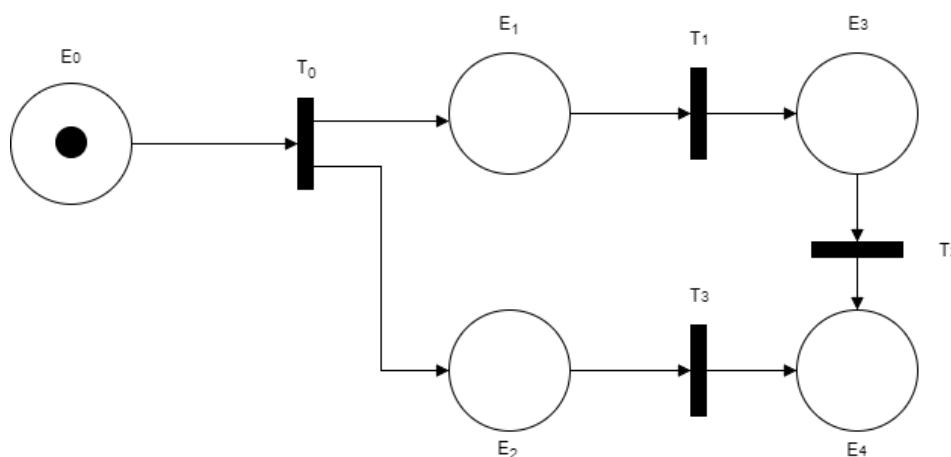


Figura 2.3 - Rede de Petri elementar com cinco lugares e quatro transições (SILVA, 2017b).



#### 2.4.2. Redes de Petri lugar / transição

Formalmente, uma rede de Petri lugar / transição (DESEL; REISIG, 1998) é definida como uma tupla de cinco elementos  $R = (P, T, I, O, M)$ , em que:

$P = \{P_1, P_2, \dots, P_m\}$  é um conjunto finito com  $m$  lugares,  $m > 0$ ,

$T = \{T_1, T_2, \dots, T_n\}$  é um conjunto finito com  $n$  transições,  $n > 0$ ,

O conjunto  $P \cap T = \emptyset$ ,

$I: T \rightarrow P$  é a função de entrada e identifica o conjunto de lugares de entrada para uma transição,

$O: T \rightarrow P$  é a função de saída e identifica o conjunto de lugares de saída para uma transição,

$M: P \rightarrow N$ , é a marcação ou estado da rede, onde  $N$  é um conjunto de inteiros incluindo o zero.

Através de notação matricial, as propriedades de uma rede de Petri podem ser algebricamente analisadas. As funções  $I$  e  $O$  são substituídas pelas matrizes  $E$  e  $S$  apresentadas na Eq. 2.1 e na Eq. 2.2, ambas com  $m$  linhas e  $n$  colunas. A matriz de incidência  $A$  é dada pela Eq. 2.3.

$$E[i, j] = (P_i, I(T_j)), \text{ for } i = 1, 2, 3, \dots, m \quad (2.1)$$

$$S[i, j] = (P_i, O(T_j)), \text{ for } j = 1, 2, 3, \dots, n \quad (2.2)$$

$$A[i, j] = E[i, j] - S[i, j] \quad (2.3)$$

A marcação armazena o estado atual da rede de Petri, enquanto o próximo estado é dado pela Eq. 2.4.

$$M_{k+1} = M_k + u_k \cdot A^t \quad (2.4)$$

Em que:

$M_k$  é um vetor coluna ( $m \times 1$ ) cujo elemento da  $i$ -ésima linha corresponde a quantidade de marcações no  $i$ -ésimo lugar, em determinado instante de tempo  $k$ ;

$M_{k+1}$  é o vetor de marcação resultante do disparo  $k$ ;

$A^t$  é a função transposta da matriz de incidência, que representa o conjunto de arcos da rede de Petri. A matriz de incidência possui  $m$  linhas e  $n$  colunas. O elemento  $a_{ij}$  terá valor  $-1$  quando o lugar correspondente à linha  $i$  for uma entrada da transição

correspondente à coluna  $j$ , e terá valor 1 quando for uma saída. Caso não exista arco ligando o lugar e a transição correspondentes, o elemento terá valor 0.

$u_k$  é um vetor linha ( $1 \times n$ ), ou vetor de disparo, cujo elemento da  $n$ ésima coluna corresponde à quantidade vezes que a transição  $n$  foi realizada no disparo.

Desse modo, a rede de Petri da Fig. 2.4 possui a matriz de incidência indicada na Eq. 2.5. Admitindo uma marcação inicial com uma ficha no lugar  $E_0$  e duas fichas no lugar  $E_3$ , pode-se construir o vetor de marcação inicial indicado na Eq. 2.6. Admitindo-se ainda que no instante 0 há o disparo das transições  $T_0$  e  $T_2$ , tem-se o vetor de disparos indicado na Eq. 2.7. Aplicando-se a equação de mudança de estado (Eq. 2.4) para o caso admitido, chega-se a um vetor de marcação para o instante  $k = 1$ , conforme mostrado na Eq. 2.8. A descrição gráfica da mudança de estado exemplificada pelas Eqs 2.5 a 2.8 está apresentada na Fig. 2.4.

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix} \quad (2.5)$$

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 0 \end{bmatrix} \quad (2.6)$$

$$u_0 = [1 \quad 0 \quad 1 \quad 0] \quad (2.7)$$

$$M_1 = M_0 + u_0 \cdot A^t = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 2 \\ 0 \end{bmatrix} + [1 \quad 0 \quad 1 \quad 0] \cdot \begin{bmatrix} -1 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (2.8)$$

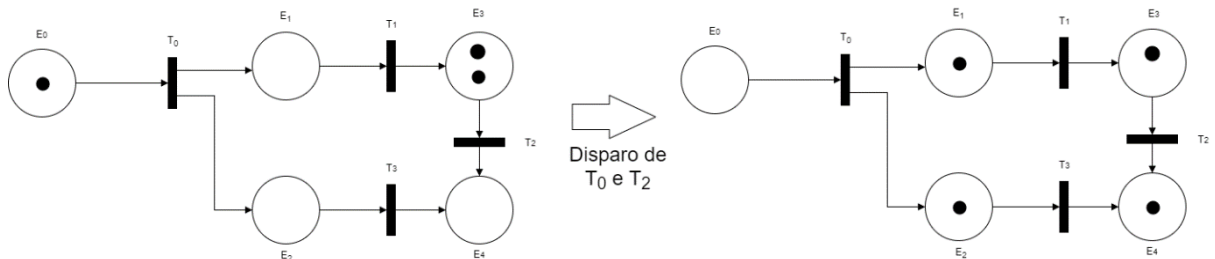


Figura 2.4 - Representação gráfica de uma mudança de estado em uma rede de Petri (SILVA, 2017b).

### 2.4.3. Redes de Petri coloridas

Classificada como uma rede de Petri de alto nível, a rede de Petri colorida (CPN) consiste em uma linguagem gráfica para a construção de modelos de sistemas concorrentes e análise de suas propriedades. Esta linguagem de modelagem de eventos discretos combina as capacidades das redes Petri com as capacidades de uma linguagem de programação de alto nível. As redes de Petri fornecem a base da notação gráfica e das primitivas básicas para modelagem de concorrência, comunicação e sincronização. A linguagem de programação CPN ML, baseada na linguagem de programação funcional *Standard ML* (MILNER et al., 1997), fornece as primitivas para a definição de tipos de dados, para descrever a manipulação de dados e para criar modelos compactos e parametrizáveis. A linguagem de modelagem da CPN é uma linguagem de modelagem de uso geral, ou seja, não se destina a modelar uma classe específica de sistemas, mas é voltada para uma classe muito ampla de sistemas que podem ser caracterizados como sistemas concorrentes. Domínios de aplicação típicos de redes de Petri coloridas são protocolos de comunicação, redes de dados, algoritmos distribuídos e sistemas embarcados. As CPNs também são aplicáveis em processos de negócios e fluxos de trabalho, sistemas de manufatura e sistemas de agentes (JENSEN; KRISTENSEN, 2009).

Modelos CPNs são orientados a estados e ações. Ou seja, descreve-se os estados do sistema e os eventos (transições) que podem fazer com que o sistema altere o estado. Ao realizar simulações do modelo CPN, é possível investigar diferentes cenários e explorar o comportamento do sistema.

O CPN Tools é uma ferramenta para edição, simulação, verificação e análise de desempenho de modelos CPN. É utilizado por mais de 8.000 usuários em 140 países e está disponível para Windows e Linux (JENSEN; KRISTENSEN, 2009).

Jensen e Kristensen (2009) reforçam que as redes de Petri são tradicionalmente divididas em redes de Petri de baixo nível e redes de Petri de alto nível, sendo a última um padrão internacional (ISO/IEC, 2004) (ISO/IEC, 2011). De acordo com os autores, a linguagem de modelagem CPN com suas ferramentas de suporte computacional estão em conformidade com este padrão.

### 2.4.4. A PNRD

Redes de Petri elementares inseridas em base de dados RFID é uma estrutura de dados formal a ser armazenada na etiqueta RFID de forma a integrá-la com sistema de controle (TAVARES; SARAIVA, 2010). Esta abordagem foi concebida para processos logísticos e de manufatura e pressupõe que o comportamento típico de um objeto possa ser modelado por meio de redes de Petri. Desse modo, o fluxo de trabalho da etiqueta RFID

associada ao objeto pode ser definido por uma matriz de incidência e o seu estado atual por um vetor de marcação. Essas duas informações, juntamente com a identificação do objeto (ID da etiqueta) e uma indicação de tempo, caracterizam a estrutura típica da PNRD. Sendo que a indicação de tempo é incluída de forma a possibilitar o sistema a realizar análises quantitativas de performance. As antenas e leitores, por sua vez, estão vinculadas ao vetor de disparos adequado a cada situação.

Na abordagem original da PNRD cada *tag* armazena a matriz incidência e o vetor de estados de uma rede de Petri referente ao processo ao qual o objeto identificado em questão participa; e cada leitor armazena uma lista de vetor de disparos. O funcionamento da PNRD original se baseia na captura do *tagID* seguido da matriz de incidência  $A^t$  e o vetor de estados  $M_k$  da *tag* pelo leitor. O software responsável pelo cálculo do próximo estado encontra o vetor de disparos  $u_k$  relativo ao *tagID*, estado da *tag*, antena, leitor, e outros dados adicionais que forem necessários, como intervalo de tempo, distância entre outros. Com todas essas informações é possível calcular o próximo vetor de estados  $M_{k+1}$ , conforme a Eq. 2.4.

O diagrama da Fig. 2.5 representa esquematicamente um sistema PNRD típico. Nele um objeto de interesse (como uma caixa, uma peça ou um pacote) têm seu fluxo de trabalho representado por uma matriz de incidência. O desenvolvedor implementa a rede de Petri associada ao sistema, em linguagem textual de programação, através de um software denominado DEMIS. A etiqueta associada ao objeto armazena informações referentes à matriz de incidência e ao estado atual do objeto (dados adicionais também podem ser armazenados), enquanto as informações do vetor de disparo (armazenadas no DEMIS) estão associadas às antenas posicionadas em pontos estratégicos (mudança de processos). Um leitor pode estar integrado a uma ou várias antenas distribuídas no ambiente e cada leitor comunica-se com o DEMIS via protocolo TCP/IP, viabilizando o controle do processo.

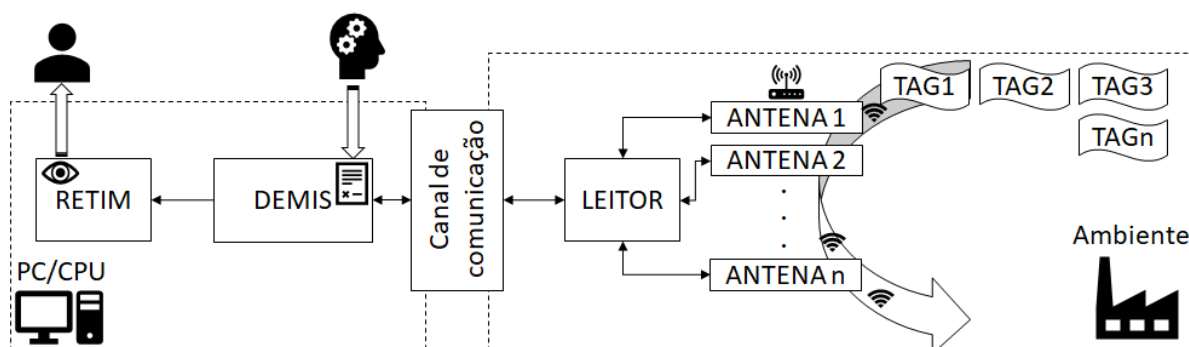


Figura 2.5 - Esquema ilustrativo do modo de operação de um sistema PNRD típico; baseado em Tavares e Saraiva (2010).

No momento em que um objeto passa por uma determinada antena, o mesmo pode ter seu estado atualizado. Inconsistências no fluxo de trabalho podem ser percebidas e exceções tratadas pelo sistema. Além disso, o DEMIS envia informações para outro software denominado RETIM, através do qual o usuário consegue visualizar, de forma gráfica, o comportamento da rede. A Fig. 2.6 ilustra o modo de operação entre DEMIS e RETIM.

Considerando o exemplo prático de uma fábrica produtora de mancais (SILVA, 2017b), uma peça usinada precisa de inspeção quanto a dois tipos de requisitos: a cilindricidade e a rugosidade do furo. Caso o teste de cilindricidade acuse desvio dos padrões aceitáveis, a peça é enviada para o setor de reciclagem. Estando dentro dos padrões, a peça é encaminhada para o teste de rugosidade. Caso o mancal esteja dentro dos limites aceitáveis de rugosidade, o mesmo é encaminhado para o estoque. E, caso seja reprovado no teste de rugosidade, a peça é colocada em uma fila para retrabalho. A representação gráfica deste exemplo está ilustrada na rede de Petri da Fig. 2.7.

Implementando a abordagem PNRD neste processo, cada peça usinada contém uma etiqueta RFID com informações da matriz de incidência e do vetor de marcação atual. Além disso, os pontos de mudança de processos estão associados a transições na rede de Petri. Dessa forma, grupos leitor/antena RFID são posicionados nos pontos de interesse, a saber: setores responsáveis pelos testes de cilindricidade e de rugosidade, setor de fabricação e retrabalho, estoque e reciclagem.

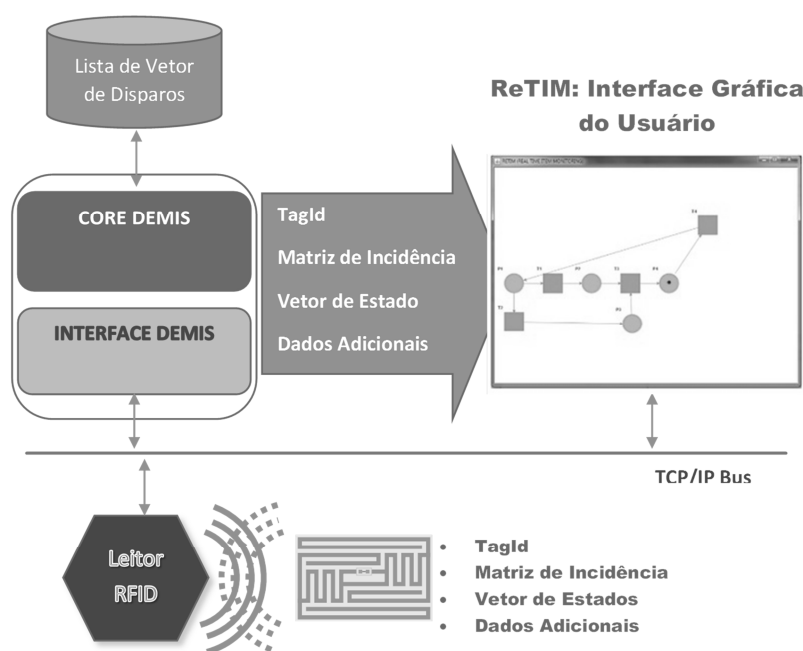


Figura 2.6 - Esquema ilustrativo do modo de operação do DEMIS com o RETIM. Modificado de Tavares et al. (2017).

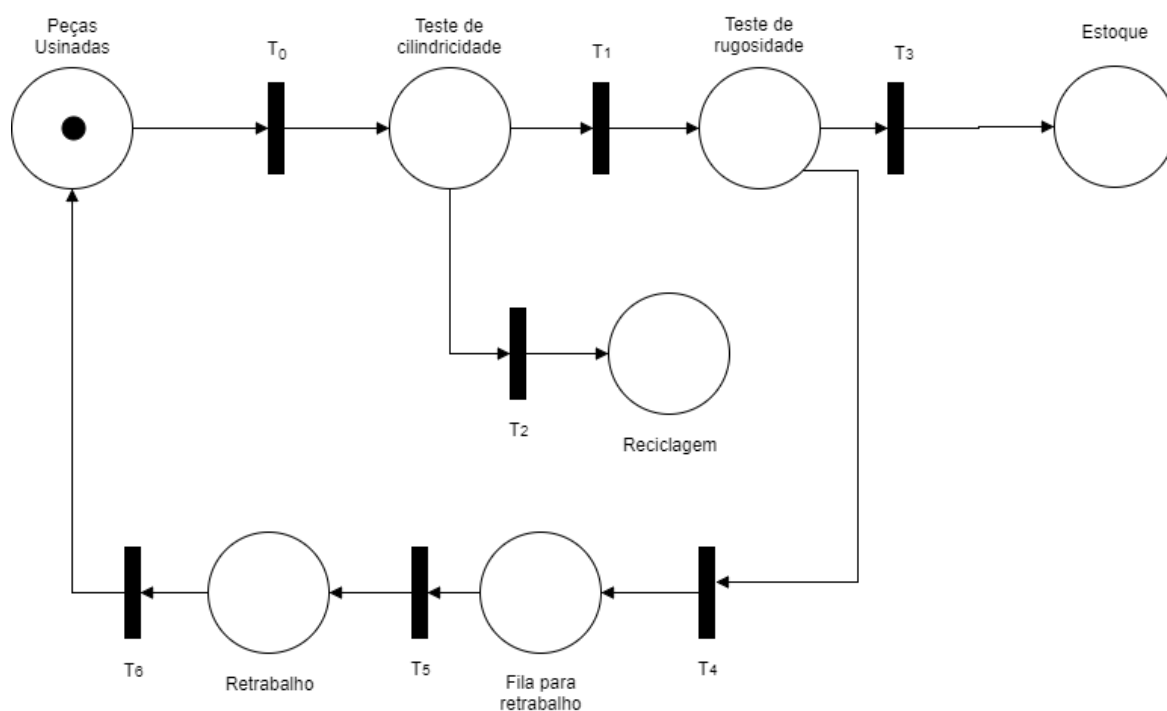


Figura 2.7 - Exemplo de uma rede de Petri para o processo de verificação de conformidade de mancais. (SILVA, 2017b)

Analizando o sistema, a máquina que verifica a cilindricidade do furo é dotada de uma antena RFID. O leitor ligado a essa antena, ao perceber uma nova peça, adquire as informações da mesma, isto é, seu vetor de marcação (estado atual “peça usinada”) e sua matriz de incidência, realiza o disparo da transição  $T_0$ , calcula o próximo estado e o grava na etiqueta. Em seguida, em concordância com o novo estado da etiqueta (teste de cilindricidade), o sistema encaminha um comando para que o teste seja iniciado. Dependendo do resultado do teste, o sistema envia uma mensagem ao aplicativo RFID que dispara a transição  $T_1$  ou a  $T_2$  e atualiza o novo estado da etiqueta.

Neste caso, a passagem do objeto em uma determinada antena não define a transição disparada. É necessário um conjunto de informações (que podem ou não incluir: dados externos, marcação inicial, a identificação da etiqueta e a identificação da antena) para identificar uma transição disparada. De forma semelhante, uma transição não está necessariamente ligada a uma única antena. No caso apresentado, poderiam haver várias máquinas verificadoras de medidas. No entanto, independentemente do número de máquinas, o fluxo de trabalho não é afetado. Isso permite que o sistema se expanda sem sofrer uma grande modificação no seu funcionamento. Nesse caso, a abordagem PNRD torna o sistema de controle da fábrica facilmente escalável (SILVA, 2017b).

Em uma linha distinta, Tavares et al. (2017) utilizaram a PNRD em sistemas de posicionamento interno para controle de posição de robôs móveis baseado na potência do

sinal recebido. O esquema da Fig. 2.8 ilustra o modo de operação típico desta aplicação, na qual o modelo comportamental do robô prevê comandos de navegação dependentes da distância entre *tag* e antena. Para modelar interfaces de comunicação na rede de Petri, os autores optaram pelo uso de *pseudo-box* (DEL FOYO; SILVA, 2004).

Estudos mais recentes (SILVA, 2017b) (SILVA; TAVARES; FERREIRA, 2018) levaram a PNRD para o nível dos dispositivos embarcados ao desenvolver uma biblioteca para a plataforma Arduino. O esquema da Fig. 2.9 ilustra o modo de operação da PNRD associada ao uso da biblioteca mencionada. O desenvolvedor implementa a rede de Petri associada ao sistema através da IDE do Arduino. Cada dispositivo de campo (Arduino) deve possuir um leitor/antena RFID integrado. No momento em que um objeto passa por determinado dispositivo leitor, o mesmo pode ler, interpretar e atualizar o estado do objeto. Passagens por locais não previstos no fluxo de trabalho do objeto podem ser percebidas e exceções tratadas pelo sistema. Esta aplicação não suporta a visualização da rede de Petri de forma gráfica.

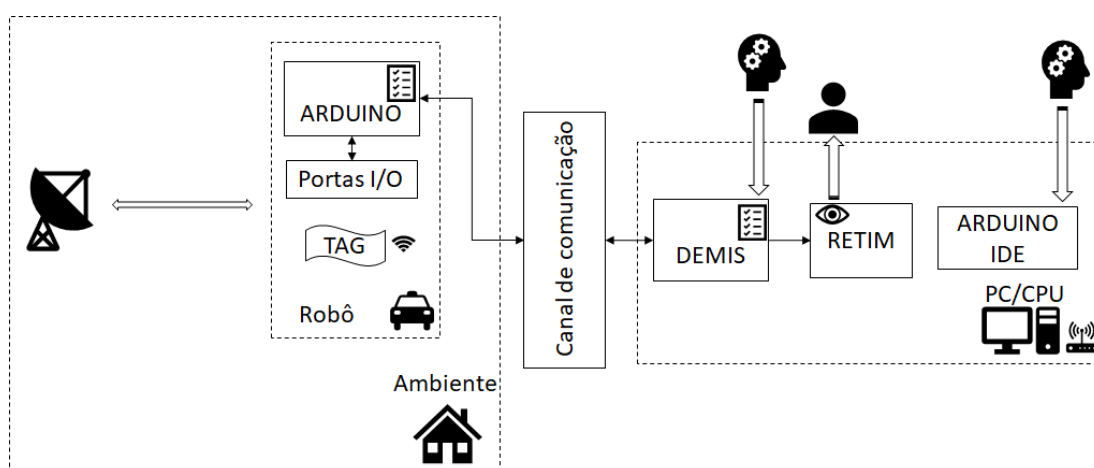


Figura 2.8 - Esquema ilustrativo do sistema PNRD integrado ao sistema de posicionamento interno para controle de posição de robôs móveis; baseado em Tavares et al. (2017).

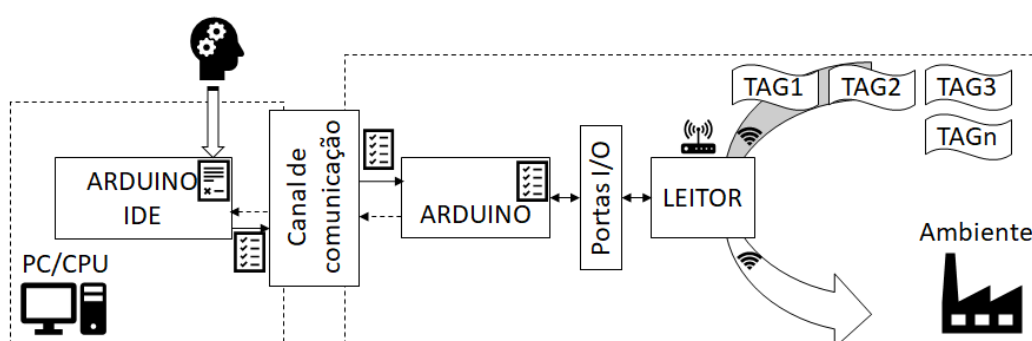


Figura 2.9 - Esquema ilustrativo do modo de operação da PNRD associada ao uso da biblioteca para Arduino; baseado em Silva (2017b).

## 2.5. Trabalhos Relacionados

Na última década, a Internet das Coisas (IoT) surgiu como um conceito de sistema inteligente, flexível, autônomo e distribuído. Em sistemas baseados em IoT, o nível de cognição deve ser fortemente integrado ao controle de baixo nível. Entretanto, esta integração ainda é um desafio (GIUSTO et al., 2010).

Na robótica móvel, por exemplo, o nível de cognição é responsável por decidir a melhor maneira de realizar a atuação, a fim de executar uma tarefa que garanta segurança e respeite as restrições impostas pelo ambiente (SIEGWART; NOURBAKHSH, 2004).

Como a pesquisa em robótica móvel é muito ampla, apontar estudos em áreas específicas nem sempre é trivial. Este trabalho relaciona-se com outros no ponto de vista da integração da tecnologia RFID em sistemas multirrobôs aplicados a operações de busca e salvamento. Além disso, a descrição do modelo comportamental em CPN e a sua integração com um simulador ou com os robôs reais também é investigada.

Neste sentido, realizou-se uma pesquisa bibliográfica inicial a partir da combinação de quatro palavras-chave: “*search and rescue*”, “*robot*”, “*Petri*” e “*RFID*”. A Tab. 2.5 apresenta os resultados referentes a manuscritos publicados dentro do período de 2005 a 2018. Percebe-se maior concentração de trabalhos que abordam SAR e sistemas RFID (42,6%), SAR e redes de Petri (18,5%) ou SAR, robótica e RFID (22%). Um percentual de 13% dos trabalhos aborda SAR, robótica e rede de Petri, enquanto apenas 3,7% do total de trabalhos envolvem a integração de robótica, redes de Petri e sistemas RFID. Não foi encontrado nenhum caso que contivesse as quatro palavras-chave em sua abordagem, tal como verificado com a combinação SAR, redes de Petri e sistemas RFID.

Adotando-se uma postura mais específica, substituíram-se as palavras-chave “*robot*” e “*Petri*” por “*multi-robot*” e “*colored Petri net*”, respectivamente, chegando-se a novas combinações, cujos resultados estão apresentados na Tab. 2.6. Nota-se uma escassez de abordagens que envolvam tais conceitos/sistemas na solução dos problemas de busca e salvamento. Foram encontrados apenas cinco trabalhos, dos quais a maioria (83,3%) envolve sistemas multirrobôs e CPN, seguidos por SAR com CPN (16,7%). As demais combinações apresentadas na Tab. 2.6 não retornaram nenhum trabalho associado, demonstrando uma grande lacuna na integração de tais conceitos e tecnologias. Uma representação gráfica da distribuição de trabalhos publicados em periódicos e em anais, entre 2005 e 2018, por combinação de palavras-chave, é apresentada na Fig. 2.10.



Tabela 2.5 - Resumo dos resultados da pesquisa bibliográfica a partir da combinação de quatro palavras-chave: “*search and rescue*”, “*robot*”, “*Petri*” e “*RFID*”.

Palavra-chave				Procedência	
“ <i>Search and Rescue</i> ”	“ <i>Robot</i> ”	“ <i>Petri</i> ”	“ <i>RFID</i> ”	Periódicos	Anais
✓			✓	5	18
✓		✓		3	7
✓	✓		✓	2	10
✓	✓	✓		1	6
	✓	✓	✓	0	2
✓		✓	✓	0	0
✓	✓	✓	✓	0	0

Tabela 2.6 - Resumo dos resultados da pesquisa bibliográfica a partir da combinação de quatro palavras-chave: “*search and rescue*”, “*multi-robot*”, “*colored Petri net*” e “*RFID*”.

Palavra-chave				Procedência	
“ <i>Search and Rescue</i> ”	“ <i>Multi-robot</i> ”	“ <i>Colored Petri Net</i> ”	“ <i>RFID</i> ”	Periódicos	Anais
	✓	✓	✓	0	0
✓	✓	✓		0	0
✓		✓	✓	0	0
✓	✓		✓	0	0
✓		✓		0	1
	✓	✓		1	4

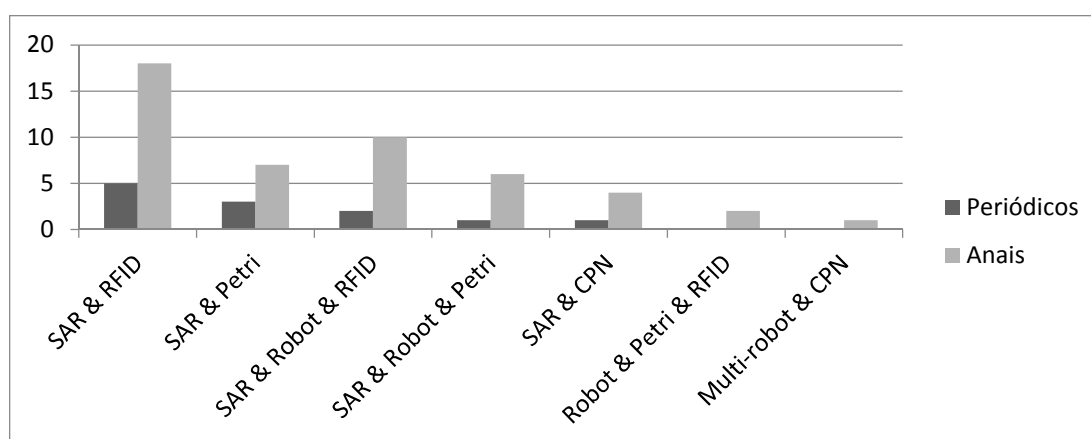


Figura 2.10 - Distribuição de trabalhos publicados em periódicos e em anais, de 2005 a 2018, por combinação de palavras-chave.

Além dos trabalhos relacionados à PNRD (ver subseção 2.4.4), outras aplicações e métodos encontrados na literatura são apresentados nesta seção. Como nem todos os trabalhos abrangem todos os tópicos listados, a revisão da literatura será apresentada em subseções distintas. Os trabalhos semelhantes ao desta tese quanto à utilização de redes de Petri e tecnologia RFID são discutidos na subseção 2.5.1, a seção 2.5.2 apresenta os trabalhos que mais se assemelham quanto à aplicação da tecnologia RFID em ambientes SAR, a seção 2.5.3 discute trabalhos semelhantes quanto à integração entre a tecnologia RFID e a robótica móvel, a seção 2.5.4 aborda trabalhos semelhantes quanto à aplicação de robôs em ambiente SAR e a seção 2.5.5 descreve trabalhos semelhantes quanto à aplicação de redes de Petri e robôs em ambientes SAR.

#### *2.5.1 Trabalhos semelhantes quanto à utilização de redes de Petri e RFID*

A literatura aponta diversas tentativas de integração entre redes de Petri e RFID. Wang e Ma (2012), por exemplo, abordam protocolos de segurança contra ataques para sistemas RFID. As propriedades de segurança são verificadas utilizando provas algébricas e redes de Petri coloridas. A robustez do sistema foi verificada através de modelos desenvolvidos e testados em CPN Tools. Todos os modelos com atividades sujeitas a ataques foram devidamente interrompidos.

Gao et al. (2013), por sua vez, abordam protocolos de segurança contra ataques para sistemas RFID. O protocolo de segurança proposto é modelado e verificado através de redes de Petri coloridas e o CPN Tools é utilizado para validar as redes associadas ao protocolo de segurança proposto. O estudo aborda a faixa de frequência 860 a 960 MHz (UFH) com capacidade de armazenamento de 216 bytes. Os autores não afirmam, mas pelas imagens percebe-se a utilização do CPN Tools como ferramenta para o desenvolvimento das redes de Petri coloridas e simulação de situações de ataque.

Embora os primeiros trabalhos citados abordem as questões de protocolos de segurança em sistemas RFID, a maioria dos trabalhos que tentam integrar redes de Petri e RFID, a exemplo do que ocorre com a PNRD, aplicam-se em ambientes de manufatura. Jiang et al. (2017), por exemplo, utilizam CPN para modelar agentes em ambientes de manufatura flexível integrados com sistemas RFID. Os autores associam agentes ativos (máquinas, robôs, AGVs) com leitores RFID e agentes passivos (peças, produtos) com *tags* RFID. *Tags* armazenam o estado e o fluxo de processo esperado para determinado agente passivo. Além de poder atualizar o estado dos agentes passivos, o modelo comportamental dos agentes ativos garante autonomia em tomada de decisão. A integração entre redes de Petri de agentes distintos é modelada através de *Interface Places*. Um exemplo de implementação reforça a metodologia de modelagem, integração e aplicação proposta. Os

autores não deixam claro se o sistema foi implementado na prática. Os modelos comportamentais não são utilizados diretamente para monitoramento do sistema, tal como no CPN Tools. Os autores propõem a sua conversão em XML para melhor manipulação e armazenamento.

Lv et al. (2012), por sua vez, exploram a possibilidade de utilização de CPN e RFID para análise de dados em tempo real. Os autores relacionam *tokens* coloridos com informações a respeito do produto armazenadas em *tags* RFID para monitoramento de qualidade. Resultados de simulação são discutidos quanto à capacidade de monitoramento do status de qualidade do produto, e tomadas de decisão. A integração do sistema RFID com CPN Tools não é clara. Não utiliza funções de comunicação, de modo que não se entende como as informações chegam e partem da CPN Tools, tornando os resultados questionáveis. O estudo de caso apresenta uma rede relativamente complexa, mas não detalha como os resultados foram obtidos, se a partir de dados históricos, se a partir de simulação com eventos estocásticos, ou através da integração online da CPN com as máquinas do chão de fábrica.

Já Chen (2012) adota a rede de Petri colorida como método para construir e analisar modelos de sistemas de manufatura flexível. Associa as informações de *tokens* coloridos com as informações a gravar em *tags* RFID. Utiliza uma célula de manufatura flexível virtual como estudo de caso para verificar a abordagem proposta. O *abstract* não deixa claro se a CPN opera em tempo real com a célula de manufatura virtual ou é apenas uma base para a programação do comportamento da célula. Resultados experimentais ou integração com sistemas físicos não são abordados.

De las Morenas, García-Higuera e García-Ansola (2017) implementam o controle distribuído do sistema logístico de um armazém utilizando a metodologia de agentes em controladores industriais. A chegada simultânea de produtos em regiões de cruzamentos das linhas de transporte (correias transportadoras) gera competição por recursos, que é gerenciada por um agente modelado em redes de Petri lugar/transição e implementados em CLP com linguagem *Ladder*. Cada produto possui uma etiqueta RFID com sua identificação, sua origem, destino e prioridades em casos de competição por recursos. Leitores RFID posicionados nas regiões de interesse possibilitam o acesso às informações necessárias para tomadas de decisão por parte dos agentes do sistema. Os agentes rodam em APIs Java e acessam informações dos CLPs via rede. Redes de Petri não controlam diretamente o comportamento do sistema, sendo necessária a tradução (*offline*) da sua lógica em linguagem *Ladder*. O comportamento dos produtos no sistema não é modelado em redes de Petri e, conseqüentemente, não há integração direta do sistema RFID com modelos em rede de Petri.

Sun, Chang e Mo (2009) propõem um método de monitoramento e controle do processo de montagem de produtos complexos, de modo a atingir a sincronia entre o fluxo logístico e o fluxo de informações e combinar dinamicamente materiais com tarefas de montagem. Eventos do sistema de montagem são descritos como lugares e mapeados em *tags* RFID. Estados de montagem e transporte são descritos como transições e são implementados por agentes logísticos e agentes de montagem. Informações das *tags* RFID são utilizadas para disparar tarefas de montagem. Os autores apresentam um estudo de caso para o processo de montagem de motores aeroespaciais. Baseado na ferramenta *aglets* da IBM *Japan* um protótipo digital com interface de monitoramento é apresentado. Os autores não discutem se o sistema deve ser automático ou apenas indicaria tarefas para os operadores das máquinas. Caso seja automático, o processo de criação de novas *tags* para identificar um produto montado não é discutido.

Por fim, em uma linha diferente, Zhang, Xu e Chou (2016) propõem uma nova metodologia de modelagem para interação entre dispositivos sensores e ambiente geográfico baseada em redes de Petri. O processo de modelagem é discutido através de exemplos em redes de Petri coloridas e o CPN Tools é utilizado para verificação dos modelos e simulações de execução. O sistema RFID não é abordado diretamente, mas pode ser considerado um tipo de implementação de rede de sensores sem fio, ou *wireless sensor network* (WSN).

### 2.5.2 Trabalhos semelhantes quanto à aplicação de tecnologia RFID em ambientes SAR

Alguns trabalhos abordam a utilização de fundamentos científicos e recursos tecnológicos, como tecnologia RFID, para a solução de problemas característicos de ambientes SAR. Wu e Vaidya (2010), por exemplo, desenvolveram uma estrutura de armazenamento de dados distribuída através de *tags* RFID passivas posicionadas em árvores para busca e salvamento em florestas. À medida que um usuário se move pela floresta, seu leitor RFID escreve sua identificação e um número sequencial nas *tags* ao alcance. Isto cria uma trajetória digital que facilita a busca de usuários perdidos. Os autores tratam *tags* com memória cheia através de quatro algoritmos de subscrição: (1) subscreve sua informação em cima da mais antiga, (2) subscreve sua informação de modo aleatório, (3) subscreve sua informação em cima da informação mais vista nas *tags* anteriores, (4) subscreve sua informação em cima daquela menos que ele menos apagou nas *tags* anteriores. Os algoritmos (3) e (4) requerem que o usuário consiga armazenar os dados lidos nas *tags* anteriores. Os algoritmos são simulados e avaliados em um exemplo simplificado baseado na trilha de Yosemite Valley no Yosemite National Park (Califórnia), situação na qual o algoritmo (3) apresentou melhor desempenho.

Vojtech et al. (2015), por sua vez, apresentam um sistema RFID de localização externo capaz de localizar centenas de *tags* ativas em uma área de 1 km<sup>2</sup>. O foco é identificação de *tags* de triagem em ambientes de desastre. Três estações de recebimento triangulam a posição aproximada de cada *tag* na região. Foram construídos protótipos de *tags* ativas e estações de recebimento, ambos testados em ambiente externo. A posição de cada elemento do sistema (estações de leitura e *transponders* identificados) é mostrada em um mapa em um terminal móvel.

Ganz et al. (2015) apresentam um sistema denominado DIORAMA que fornece consciência situacional para operações de busca e salvamento em áreas urbanas. O sistema usa tecnologia RFID ativa, bem como ferramentas de análise visual. Usando um *smartphone*, o DIORAMA permite que o usuário possa realizar uma triagem de pacientes com etiquetas RFID ativas, bem como marcar a localização de pacientes presos e a localização de pontos com fogo, derramamento de produtos químicos, colapso de edifícios, etc. Além disso, a localização dos pacientes marcados será automaticamente rastreada e atualizada na interface, de forma que o comandante da operação possa interagir com a equipe em solo diretamente pela interface do software. Este é outro caso que também não prevê equipes de campo formadas por robôs autônomos ou utilização da CPN para descrição e análise do sistema.

Hamp et al. (2014) apresentam resultados de um projeto de pesquisa com o objetivo de melhorar operações de busca e salvamento em áreas urbanas, ou *Urban Search and Rescue* (USAR). Duas tecnologias sem fio são utilizadas para a detecção e localização de vítimas inconscientes presas ou soterradas. As vítimas podem ser localizadas através do seu celular (GSM) se este estiver ativo. Vítimas sem celular não podem ser detectadas. Neste caso, os autores propuseram a utilização de um radar de onda contínua e penetrante no solo. Um dispositivo remoto denominado FRIEDAA permite a coleta, processamento e representação de informações relevantes, como resultados de pesquisa e locais de pessoal em tempo real. Este caso também não prevê equipes de campo formadas por robôs autônomos ou utilização da CPN para descrição e análise do sistema.

Finalmente, Wu et al. (2018) apresentam resultados da utilização de rede de sensores sem fio com roteamento oportunista para sistemas de busca e salvamento marítimos. Os autores destacam que as medidas de resgate existentes dependem apenas de equipamentos de comunicação por satélite, o que dificulta a confirmação de um posicionamento preciso e dos sinais vitais das pessoas que se afogam. Um dos principais desafios da utilização de WSN em busca e salvamento marítimos está no roteamento da rede. Os autores propõem uma nova técnica de roteamento denominada roteamento oportunista, onde as decisões de encaminhamento no roteamento baseiam-se apenas na

informação do vizinho. Maiores informações sobre as técnicas de roteamento analisadas podem ser encontradas na referência. Este caso não prevê equipes de campo formadas por robôs autônomos ou utilização da CPN para descrição e análise do sistema.

### 2.5.3 *Trabalhos semelhantes quanto à aplicação de tecnologia RFID em robótica móvel*

A aplicação da tecnologia RFID em robótica móvel também é relatada na literatura. Blatt e Szczerbicka (2016), por exemplo, utilizam sistema RFID como meio de comunicação indireta para robôs móveis em ambiente de simulação multiagente. Um algoritmo de mapeamento multiagente denominado MAF utiliza grid para salvar informações em posições específicas. O sistema RFID é proposto para implementar um grid de informações no ambiente mais fácil de ser implementado na prática. Os robôs são equipados com *tags* RFID capazes de armazenar os dados de interesse. Os robôs deixam *tags* ao longo do caminho com informações sobre a distância percorrida (distância da posição atual até o ponto de partida daquele robô em específico). O algoritmo foi adaptado para trabalhar com *tags* RFID, mas a análise foi apenas em simulação. Os autores não deixam claro como serão instaladas antenas e leitores RFID nem a dinâmica de inserção de novas *tags* no ambiente por parte de cada robô.

De La Cruz, Celeste e Bastos (2011), por sua vez, abordam uma técnica de localização e mapeamento simultâneos, ou *simultaneous localization and mapping* (SLAM), aplicada a robôs assistivos. Os autores utilizam um SLAM por Filtro de Kalman Estendido (EKF-SLAM), o qual é auxiliado por identificação de etiquetas por RFID estrategicamente posicionadas no ambiente e alcançam erros de odometria na ordem de centímetros.

### 2.5.4 *Trabalhos semelhantes quanto à aplicação de robôs em ambiente SAR*

A literatura também reserva trabalhos que relatam a aplicação de robôs em ambiente SAR, principalmente em meio acadêmico. Rohmer et al. (2010), por exemplo, desenvolveram o Quince, uma plataforma robótica móvel para pesquisa e desenvolvimento de robôs de resgate. Os autores descrevem a arquitetura modular de hardware e software do Quince, porém sem entrar em detalhes sobre modelagem comportamental de alto nível utilizando redes de Petri, ou integração da tecnologia RFID para resgate.

Pascucci e Setola (2011), por sua vez, apresentam uma estrutura de localização interna para equipes de resgate integrando robôs e operadores humanos. Uma rede composta por etiquetas RFID é usada para estruturar o meio ambiente e reduzir incertezas de posicionamento. O trabalho representa um estudo preliminar e apresenta principalmente uma arquitetura conceitual para a localização de resgate.

Danielsen (2017) apresenta um método de locomoção para robôs do tipo cobra orientado pela percepção e auxiliado por obstáculos. Exemplos de tarefas clássicas que estes robôs executam incluem missões de busca e salvamento, combate a incêndio e inspeção de tubos. O método de controle foi implementado em ROS e testado no simulador Gazebo. O robô cobra real foi controlado através de uma interface no LabView e uma interface de comunicação foi implementada entre ROS e LabView. Neste caso, o autor utilizou o ROS como *framework* para a descrição do modelo cognitivo do robô e modelos em CPN não foram discutidos. A tecnologia RFID também não faz parte do escopo deste trabalho.

Nesta mesma linha, Tuna, Gungor e Gulez (2014) utilizam robôs móveis autônomos para implantar uma rede de sensores sem fio para detecção de existência humana em caso de desastres. Através de simulações, os autores discutiram as vantagens de usar múltiplos robôs para a implantação da WSN em termos de exploração cooperativa e SLAM cooperativo, o benefício da implantação simultânea de nós de sensores sem fio durante a exploração de uma zona de implantação desconhecida e o uso da comunicação baseada em WSN como método de comunicação alternativo durante a exploração. O trabalho não utiliza a tecnologia RFID nem a CPN como ferramenta para descrição do comportamento do robô.

#### *2.5.5 Trabalhos semelhantes quanto à aplicação de redes de Petri e robôs em ambientes SAR*

Os demais artigos encontrados na literatura foram agrupados em aplicações que abordam redes de Petri em ambientes SAR, com ou sem a utilização de robôs móveis. Alguns trabalhos abordam apenas a modelagem do ambiente em redes de Petri, enquanto outros focam no modelo comportamental de agentes robóticos e alguns também em simulações e experimentos em ambientes controlados. Kong (2016), por exemplo, propõe redes de Petri estocásticas para modelagem da mobilização de processos em casos de resgates emergenciais. Segundo os autores, o modelo construído, além de permitir a visualização do processo, garante uma forma de analisar a performance do sistema através das probabilidades de ocupação de lugares e utilização de transições com o auxílio das propriedades das cadeias de Markov isomórficas características das redes de Petri estocásticas.

Qu e Wang (2016), por sua vez, propõem redes de Petri estocástica para modelagem do sistema logístico aplicado em casos de desastres e resgates emergenciais e utiliza cadeias de Markov para auxiliar as análises. Os autores concluem que as redes de Petri

permitem uma análise da performance do sistema de modo a apontar os gargalos e direciona tomadas de decisão para pontos específicos do sistema logístico.

Geng, Gong e Zhang (2017) discutem a utilização de modelos em redes de Petri para planejamento de trajetória de robôs de resgate. Estabelece-se o modelo em rede de Petri temporizada com base na força vital da vítima. Na sequência, utiliza-se Particle Swarm Optimization (PSO) para obtenção da melhor sequência de disparo de transições. Resultados de simulação mostram que o modelo estabelecido é razoável e o método proposto efetivo.

Yagoda e Coovert (2012) propõem redes de Petri como ferramenta de modelagem para interações humano-robô a partir da análise de vídeos de um operador de drones executando uma análise estrutural durante o furacão Katrina em New Orleans. São levantados três modelos: piloto, especialista de missões e do diretor de voo. Todas as redes de Petri são redes Lugar/Transição, as análises são apenas descritivas e nenhum software de análise é utilizado, a integração do modelo comportamental com as atitudes dos agentes de maneira automática ou supervisória não são abordadas e o foco é em modelagem e simulação posterior.

Em uma linha mais semelhante à adotada nesta tese, Calisi et al. (2007) abordam busca e exploração de ambientes utilizando abordagem multiobjectivos. A abordagem de busca e exploração proposta é baseada em uma representação explícita de alto nível dos planos dos robôs e é caracterizada por agir no nível estratégico da operação (direcionamento de ações e monitoramento). Petri *Net Plans* (ZIPARO et al., 2011) – uma plataforma avançada para especificação de planos para múltiplos agentes recentemente definida como uma plataforma para colaboração e coordenação em sistemas multi-robôs – é utilizado como formalismo de modelagem em nível estratégico. A abordagem é avaliada em ambiente de simulação com um simulador 3D para ambientes de salvamento (USARSim) e experimentalmente utilizando um robô Pioneer 3-AT equipado com *Laser Scan Range Finder*, câmera e sensor térmico. Descreve-se uma abordagem que provê uma maneira flexível de decidir quando é necessário considerar decisões e como escolher entre diferentes objetivos. Quatro cenários distintos foram avaliados em simulação e experimentalmente. Analisaram-se métricas de vítimas encontradas. Os autores não deixam claro como foi feita a integração entre o framework PNP e os robôs simulado e real, não sendo possível determinar se a rede de Petri foi utilizada somente para modelagem ou para controle de fato.

Por fim, neste mesmo sentido, Bertolaso et al. (2016) utilizam Petri *Net Plans*, para implementar comportamentos cooperativos complexos em um sistema multirrobô. Projetou-se uma estratégia efetiva para pouso cooperativo entre um UAV e um AGV, estratégia esta capaz de se adaptar a situações inesperadas (por exemplo, desvio inesperado de trajetória).



Segundo os autores, esta é a primeira aplicação de uma plataforma para especificação de planos orientados a equipes no controle cooperativo de cenários como o pouso coletivo. Os autores avaliaram a abordagem em um ambiente de simulação realístico utilizando ferramentas para controle e simulação de robôs pertencentes ao estado da arte. Para a implementação, ROS foi utilizado para conectar e controlar as plataformas simuladas (UAV e UGV) e V-REP foi utilizado para simular as duas plataformas e o ambiente.

A Tab. 2.7 detalha a abordagem dos principais trabalhos semelhantes baseado na integração entre nove assuntos:

1. Redes de Petri
2. Redes de Petri de alto nível
3. Tecnologia RFID
4. Rede de sensores sem fio
5. Busca e salvamento
6. Robótica
7. Múltiplos robôs
8. Verificação em ambiente de simulação
9. Verificação experimental

Nota-se que poucos autores integraram mais do que três assuntos, assim como poucos verificaram seu estudo experimentalmente ou em ambiente de simulação. Para facilitar a visualização das lacunas existentes e contribuições propostas, as características da abordagem adotada nesta tese foram inseridas na última linha da Tab. 2.7. A verificação experimental aparece destacada com um asterisco (\*), pois os resultados experimentais foram avaliados apenas do ponto de vista da operação dos dispositivos embarcados, sem a implementação da navegação e do controle reativo associados aos robôs móveis.

Tabela 2.7 - Comparativo das abordagens utilizadas pelos trabalhos semelhantes baseado na integração entre os assuntos correlatos a esta tese.

<b>Autor(es)</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
Wang e Ma (2012)	✓	✓	✓						
Gao et al. (2013)	✓	✓	✓						
Jiang et al. (2017)	✓	✓	✓						
Lv et al. (2012)	✓	✓	✓						
Chen (2012)	✓	✓	✓						
De las Morenas et al. (2017)	✓		✓						
Sun, Chang e Mo (2019)	✓		✓						
Zhang, Xu e Chou (2016)	✓	✓	✓						
Wu e Vaidya (2010)			✓		✓				
Vojtech et al. (2015)			✓		✓				✓
Ganz et al. (2015)			✓		✓				✓
Hamp et al. (2014)				✓	✓				✓
Wu et al. (2018)				✓	✓				✓
Blatt e Szczerbicka (2016)			✓			✓	✓	✓	
De La Cruz, Celeste e Bastos (2011)			✓			✓		✓	
Rohmer et al. (2010)					✓	✓		✓	✓
Pascucci e Setola (2011)					✓	✓	✓		
Danielsen (2017)					✓	✓			
Tuna, Gungor e Gulez (2014)				✓	✓	✓	✓	✓	
Kong (2016)	✓				✓				
Qu e Wang (2016)	✓				✓				
Geng, Gong e Zhang (2017)	✓	✓			✓	✓			
Yagoda e Coover (2012)	✓				✓	✓			
Calisi et al. (2007)	✓	✓			✓	✓		✓	✓
Bertolaso et al. (2016)	✓	✓				✓	✓	✓	
Fonseca et al. (2018)	✓	✓	✓		✓	✓	✓	✓	✓*

## **CAPÍTULO III**

### **METODOLOGIA E DESENVOLVIMENTO**

Este trabalho aborda a utilização de redes de Petri de alto nível e da iPNRD integradas a um sistema multirrobô aplicado em operações de busca e salvamento em trilhas e travessias. A intenção é manipular o conhecimento dos robôs, através de modelos comportamentais desenvolvidos em redes de Petri e, assim, orientar o comportamento dos agentes em campo baseado na troca de dados entre leitores e etiquetas RFID contendo o formato iPNRD. Este capítulo expõe a metodologia de projeto e o desenvolvimento do trabalho. Inicialmente, a metodologia é descrita na seção 3.1, seguida da análise do problema referente a uma simplificação da travessia Petrópolis-Teresópolis, na seção 3.2, e da proposta de modificação da PNRD, apresentada na seção 3.3. Na sequência, a seção 3.4 aborda a especificação do sistema multirrobô, enquanto a modelagem dos agentes em redes de Petri está presente na seção 3.5. Finalmente, a seção 3.6 apresenta a definição das arquiteturas de controle desenvolvidas.

#### **3.1. Metodologia**

A metodologia adotada para o desenvolvimento do trabalho foi o da pesquisa exploratória, focando no aprofundamento do comportamento do sistema, levantamento de hipóteses, definição de métodos e procedimentos e na geração de tecnologia nacional.

O método de desenvolvimento adotado segue os seguintes pontos:

- Análise do problema de busca e salvamento;
- Especificação da iPNRD associada ao sistema multirrobô para a solução do problema;
- Definição da arquitetura de controle para o método proposto;
- Preparação dos ambientes de desenvolvimento

- Preparação do ambiente de modelagem do CPN Tools;
- Preparação do ambiente de programação dos dispositivos embarcados;
- Preparação do ambiente de simulação do V-REP;
- Modelagem comportamental dos agentes do sistema em redes de Petri de alto nível;
- Desenvolvimento dos modelos dos robôs virtuais em V-REP;
- Desenvolvimento da API remota para comunicação entre V-REP e CPN Tools;
- Montagem dos cenários no simulador V-REP;
- Programação dos dispositivos embarcados aos robôs reais;
- Testes funcionais;
- Análises e conclusões.

O projeto aborda a integração da tecnologia multirrobô com a tecnologia RFID através da iPNRD em um estudo de caso específico baseado em uma simplificação da travessia Petrópolis-Teresópolis. Para atender esta necessidade, optou-se por construir um ambiente estruturado de ensaios que simula uma travessia com locais de entrada, saída e pontos de apoio. O grupo SAR é simulado por robôs aéreo e terrestres, que monitoram, buscam e resgatam usuários perdidos ou em situações de risco (os usuários, neste caso, são robôs terrestres sem funcionalidades SAR implementadas). Para isto, desenvolveram-se duas arquiteturas de controle que integram modelos desenvolvidos em CPN Tools, hora com o ambiente de simulação desenvolvido em V-Rep, hora com microcontroladores baseados na arquitetura ARM embarcados nos robôs. Para as funcionalidades do sistema RFID, instalaram-se leitores de alta frequência em cada robô terrestre e distribuíram-se etiquetas eletrônicas ao longo dos pontos de interesse. Em V-REP, as funcionalidades RFID foram modeladas como comunicação *Wi-Fi* com restrições no raio de comunicação.

Por dar enfoque no comportamento estratégico dos robôs, este trabalho não aborda modelos cinemáticos e dinâmicos ou, até mesmo, o próprio sensoriamento e atuação em níveis de controle reativo. Assim, funcionalidades como navegação, visão, mapeamento, localização e controle de trajetória não serão abordadas.

### 3.2. Análise do Problema

Criado em 1939, o Parque Nacional da Serra dos Órgãos (PARNASO) é o terceiro parque mais antigo do país. Considerado um dos melhores locais do país para a prática de esportes de montanha, como escalada, caminhada, rapel e outros, o PARNASO tem a maior rede de trilhas do Brasil. São mais de 200 km de trilhas em todos os níveis de dificuldade. A

Travessia Petrópolis-Teresópolis possui 30 km de subidas e descidas pela parte alta das montanhas. (ICMBIO, 2017)

A Fig. 3.1 apresenta uma imagem do Google Earth destacando os principais pontos da travessia Petrópolis-Teresópolis, a saber: portaria Petrópolis, portaria Teresópolis, Castelos do Açú e Pedra do Sino. Um mapa esquemático ilustrando outros pontos turísticos da travessia é mostrado na Fig. 3.2.

Os pontos intermediários contêm infraestrutura e pessoal qualificado para atendimento, e geralmente são utilizados para pernoite dos usuários. Normalmente, a travessia é realizada em três dias, com um pernoite no Castelos do Açú e outro na Pedra do Sino, mas há casos de usuários que completam a travessia em menor tempo. Nem todos os usuários pernoitam nos pontos de apoio, mas para completar a travessia, em algum momento devem passar por eles.

Para esta aplicação admitiu-se uma simplificação da travessia, definindo-se três regiões distintas:

- Região 1 - entre a portaria Petrópolis e os Castelos do Açú;
- Região 2 - entre os Castelos do Açú e a Pedra do Sino;
- Região 3 - entre a Pedra do Sino e a portaria de Teresópolis.

A caminhada no sentido Petrópolis – Teresópolis foi definida como ida (ou *ongoing*), enquanto a caminhada Teresópolis – Petrópolis, como retorno (ou *returning*).

Para avaliar a metodologia proposta, optou-se por construir um ambiente estruturado simulando a travessia passando por quatro pontos estratégicos: as portarias de Petrópolis e Teresópolis, e os pontos de apoio da Pedra do Sino e dos Castelos do Açú. Uma representação simplificada do ambiente de estudo é apresentada na Fig. 3.3.



Figura 3.1 - Imagem do Google Earth mostrando os principais pontos da travessia Petrópolis-Teresópolis (GOOGLE, 2017).

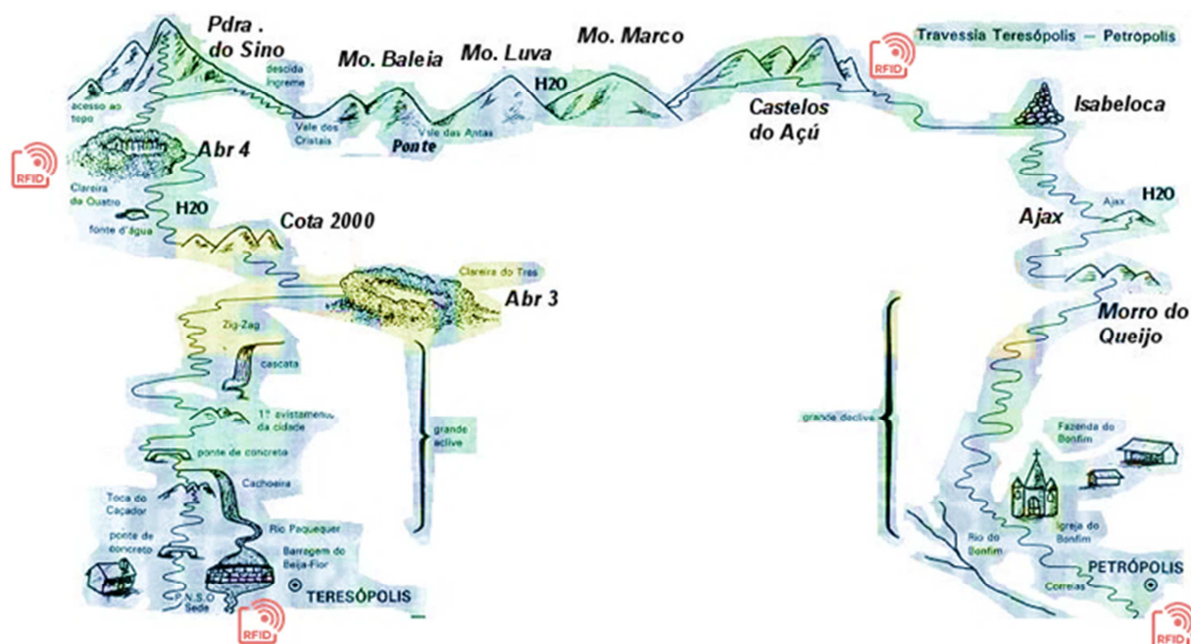


Figura 3.2 - Mapa esquemático da travessia Petrópolis-Teresópolis. Adaptado de (TRILHA & CIA, 2008).

Os agentes a serem utilizados no sistema deverão auxiliar os usuários das trilhas. Normalmente há agentes de resgate aéreos (helicóptero dos bombeiros) e agentes terrestres (guias ou pessoas com experiência no trajeto). Por se tratar de uma região vasta (mais de 30 Km de trilha) com geografia montanhosa, o agente aéreo normalmente tem acesso apenas aos pontos de apoio, cabendo aos agentes terrestres as atividades de busca e salvamento. A cada novo chamado de busca, não há precisão sobre a real posição da vítima. Para que a busca seja mais eficiente é necessária alguma forma de restringir o campo de busca para uma região com maior probabilidade de sucesso. A próxima seção aborda a proposta de modificação da PNRD como solução para o problema descrito.

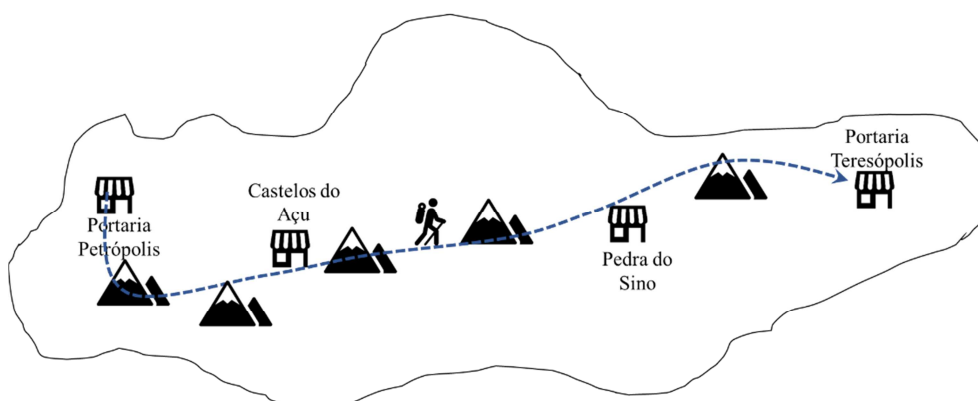


Figura 3.3 - Representação simplificada da travessia Petrópolis-Teresópolis.

### 3.3. Proposta de Modificação da PNRD

A PNRD clássica foca na identificação e monitoramento de agentes passivos, como itens comerciais, peças e produtos. A movimentação que um objeto deve realizar entre os processos associa-se ao seu modelo comportamental em rede de Petri. O controle do objeto se dá no nível dos sensores, com a distribuição física dos termos da Eq. 2.4 entre os elementos do sistema RFID. Informações referentes ao modelo comportamental (matriz de incidência  $A^t$  da rede de Petri) e ao estado atual do objeto (marcação da rede de Petri no instante atual  $M_k$ ) são armazenadas em uma *tag* passiva fixada ao mesmo, enquanto as informações referentes ao vetor de disparos  $u_k$  estão associadas aos leitores e antenas distribuídos no ambiente.

Em ambientes de busca e salvamento, onde os agentes que requerem identificação são ativos, como usuários, vítimas e socorristas, verifica-se que a estrutura clássica da PNRD seria incapaz de controlar o comportamento dos mesmos. Isso porque agentes ativos possuem comportamento autônomo e seus modelos comportamentais devem ir além do controle de movimentação ao longo dos pontos estratégicos. Esta limitação revela a necessidade de reavaliação da estrutura e distribuição física da lógica de operação da PNRD clássica.

Como os agentes ativos são normalmente microcontrolados, torna-se inviável associar sua identificação a uma etiqueta eletrônica, sendo mais razoável a utilização destas para a identificação de pontos estratégicos e locais de interesse dentro da trilha e a integração de leitores / antenas RFID ao corpo do agente ativo. Além disso, a utilização de *tags* passivas em locais remotos para identificar locais ao longo da trilha também se adequa melhor à solução para ambientes com pontos de energia escassos, haja vista a dificuldade para instalação de leitores e antenas nestas regiões. Desta forma, para que a abordagem PNRD seja capaz de controlar a movimentação de agentes ativos ao longo de pontos pré-determinados, faz-se necessária uma inversão dos locais de armazenamento dos termos da Eq. 2.4. Denomina-se PNRD invertida ou iPNRD esta nova abordagem. A Fig. 3.4 apresenta como os termos da Eq. 2.4 associam-se aos componentes do sistema RFID, tanto na PNRD clássica, quanto na iPNRD.

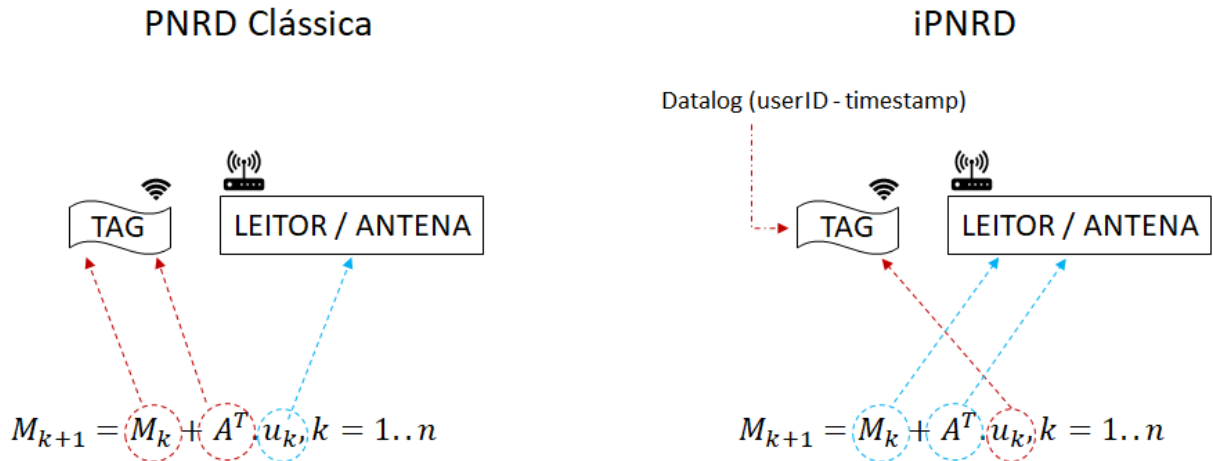


Figura 3.4 - Associação dos termos da Eq. 2.4 com os elementos RFID para as abordagens PNRD clássica e iPNRD.

Com esta inversão, a *tag* passa a armazenar informações referentes ao vetor de disparos  $u_k$  da rede, enquanto a matriz de incidência  $A^t$  e a marcação atual  $M_k$  passam a estar associados ao agente ativo portador do Leitor/antena RFID. O problema específico de busca e salvamento também pode ser beneficiado pela iPNRD através do armazenamento do histórico de usuários que passaram por um local associado a uma *tag*. Considera-se esta informação essencial nos casos de desaparecimento, pois pode-se direcionar as buscas para a região mais provável de se encontrar a vítima, reduzindo o tempo de procura e aumentando, consequentemente, as chances de salvamento.

Sugere-se que a iPNRD seja aplicada na representação simplificada da travessia, ilustrada na Fig. 3.3, definindo os pontos de entrada e apoio interno como locais a serem identificados com *tags*, enquanto usuários e agentes de busca terrestre seriam equipados com leitores RFID. As *tags* armazenariam o vetor de disparos  $u_k$  referente à rede de Petri dos usuários, além de registrar o histórico dos usuários por ali passaram. Dessa forma, as etiquetas operam como base de dados local para verificar se um usuário passou ou não por aquele ponto. Os usuários armazenam a matriz de incidência  $A^t$  da rede de Petri referente ao seu próprio comportamento e a marcação correspondente  $M_k$ , ou seja, em qual região do trajeto o usuário se encontra. O agente de busca terrestre verifica (por meio do leitor) em qual parte do trajeto o usuário perdido se encontra. A próxima seção aborda o processo de modelagem comportamental dos agentes do sistema em redes de Petri lugar / transição.



### 3.4. Especificação do Sistema Multirrobo

Para a composição do ambiente de investigação, definiram-se três agentes: *Walker*, *GroundBot* e *AerialBot*. Os praticantes de *trekking*, denominados *Walkers*, foram projetados de acordo com o esquema funcional da Fig. 3.5. Estes agentes são representados por robôs móveis equipados com motores DC, *encoders* e sensores ultrassônicos para navegação e desvio de obstáculos, um sensor de contato para a percepção dos agentes de resgate, um leitor RFID, para implementação da iPNRD, e LEDs (luzes de emergência) para indicar o estado do usuário dentro da trilha (em rota ou perdido), bem como para indicar seu estado de saúde.

Os agentes que compõem a equipe de busca e salvamento são denominados agentes SAR. Há um robô cartesiano (denominado *AerialBot*), cujo esquema funcional está representado na Fig. 3.6 e dois robôs móveis terrestres (denominados *GroundBots*), cujo esquema funcional está representado na Fig. 3.7. O robô cartesiano que implementa o *AerialBot* é equipado com motores de passo, possibilitando a patrulha aérea, e um sistema de visão para identificação da vítima, enquanto os robôs terrestres (*GroundBots*) são equipados com motores DC com *encoders* e sensores ultrassônicos para navegação e desvio de obstáculos, um leitor RFID, para implementação da PNRD invertida e um sensor de contato para a percepção da vítima. Para esta aplicação montada em laboratório, todos os agentes SAR possuem um módulo *Wi-Fi*, viabilizando a comunicação TCP/IP, entretanto, tecnologias de comunicação de longo alcance devem ser investigadas para aplicações em campo aberto. A Fig. 3.8 apresenta, de forma mais compacta, a integração dos esquemas funcionais dos três agentes descritos.

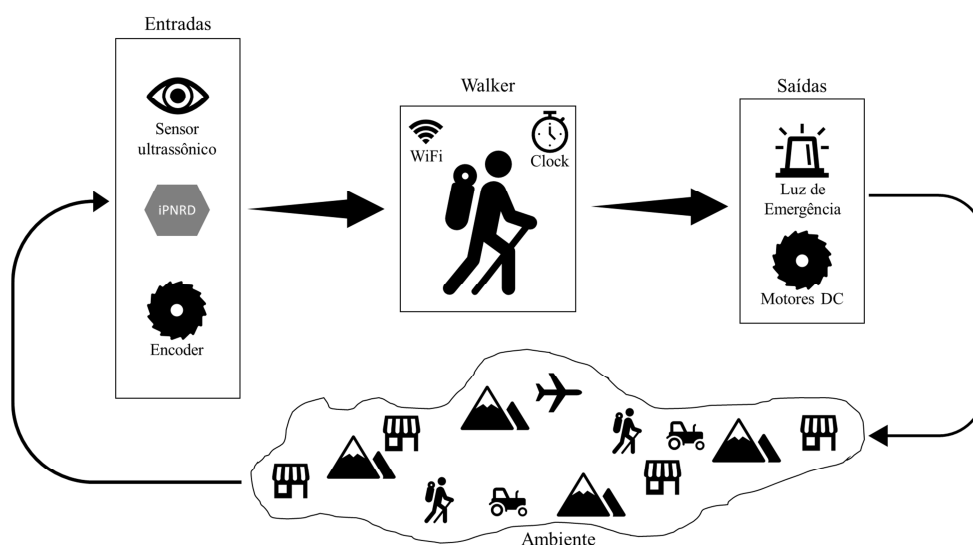


Figura 3.5 - Esquema funcional do agente *Walker* para a composição do ambiente de investigação.

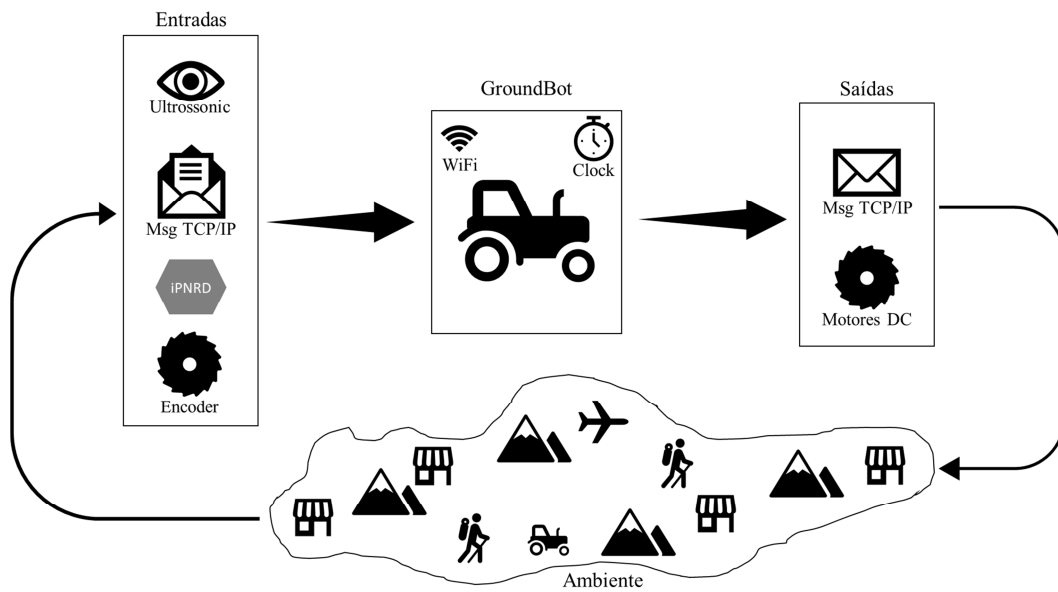


Figura 3.6 - Esquema funcional do agente *GroundBot* para a composição do ambiente de investigação.

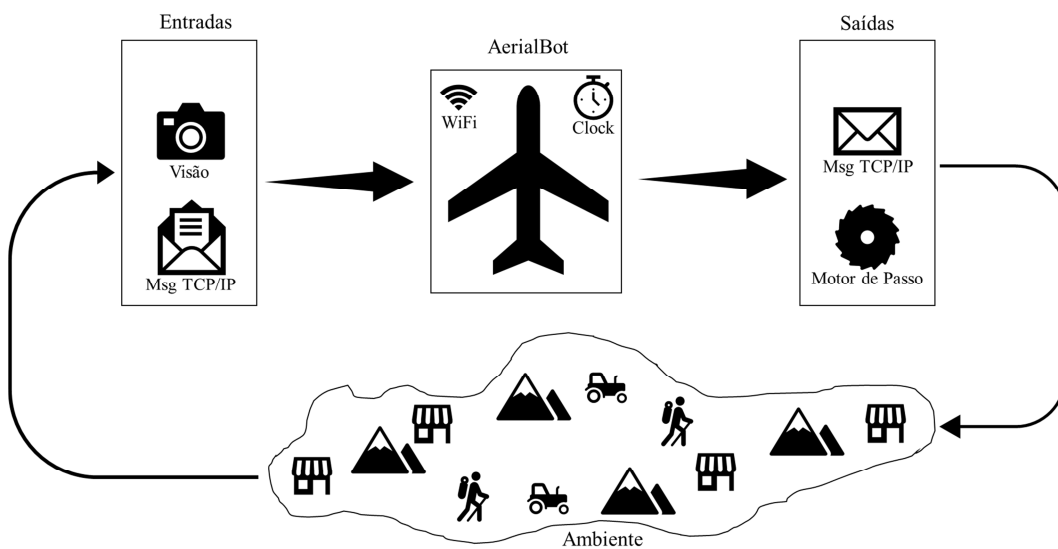


Figura 3.7 - Esquema funcional do agente *AerialBot* para a composição do ambiente de investigação.

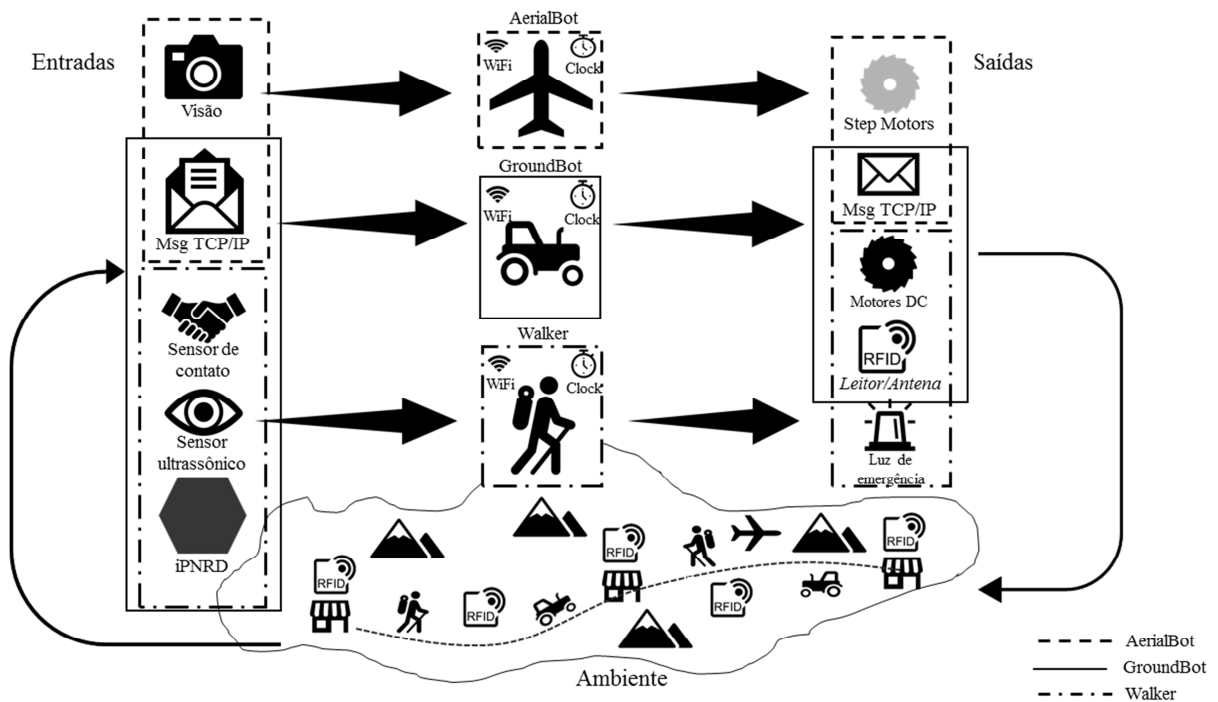


Figura 3.8 - Esquema funcional do sistema multirrobo para a composição do ambiente de investigação (FONSECA et al., 2017).

### 3.5. Modelagem dos Agentes em Redes de Petri

Para a composição do ambiente de investigação, definiram-se três agentes de busca e salvamento: *AerialBot*, *GroundBot* e *Walker*. As atitudes de cada agente foram definidas com base em suas características operacionais e no conjunto de atividades esperadas em situações de busca e salvamento em trilhas e travessias. Adotando uma abordagem baseada em eventos discretos, adotaram-se os conceitos das redes de Petri para o desenvolvimento dos modelos comportamentais de cada agente.

Para evitar que as características das redes de Petri de alto nível, como linguagem de programação, definição de tipos e comunicação com processos externos dificultasse a abstração do comportamento dos robôs, inicialmente os modelos foram concebidos em redes de Petri lugar/transição, os quais puderam ser verificados e analisados de maneira rápida e independente. Finalizado este processo e definidas as redes associadas a cada robô, inseriram-se características das redes de alto nível para viabilizar a implementação da arquitetura de controle em tempo de execução. Os modelos iniciais, desenvolvidos em redes de Petri lugar/transição, são apresentadas nas próximas subseções e suas equivalentes em redes de Petri de alto nível são melhor discutidas no capítulo IV.

### 3.5.1. Walker

Para o robô trilheiro, denominado Walker, assumiram-se oito lugares associados à posição do veículo na trilha: *Ongoing* 0% (em alguma posição dentro da Região 1, caminhando sentido Teresópolis), *Ongoing* 30% (em alguma posição dentro da Região 2, caminhando sentido Teresópolis), *Ongoing* 70% (em alguma posição dentro da Região 3, caminhando sentido Teresópolis), *Returning* 0% (em alguma posição dentro da Região 3, caminhando sentido Petrópolis), *Returning* 30% (em alguma posição dentro da Região 2, caminhando sentido Petrópolis), *Returning* 70% (em alguma posição dentro da Região 1, caminhando sentido Petrópolis), pronto para iniciar trilha (*Ready to trekking*) e trilha concluída (*Full trail*). Há quatro lugares indicando o status do usuário na trilha: Walker em rota (*On route mode*), Walker perdido (*Lost mode*), Walker resgatado (*Rescued mode*) e Walker passando por triagem (*Triage mode*). Outros dois lugares estão associados ao estado de saúde do trilheiro: Walker saudável (*Healthy mode*) e Walker ferido (*Injured mode*). O usuário inicia a travessia no modo Em Rota. Caso este não alcance o próximo ponto previsto dentro de um intervalo de tempo pré-determinado, passará para o modo Perdido, no qual permanecerá até que retorne à trilha ou seja resgatado por algum *GroundBot*. Estando em modo Resgatado e saudável, o agente Walker será liberado para retornar à trilha. Caso esteja em modo resgatado e ferido, o agente será encaminhado para triagem. Uma vez em modo triagem, o usuário permanecerá até que seja liberado para retornar à trilha. Etiquetas RFID espalhadas em posições estratégicas do trajeto auxiliam a atualização do status do robô na trilha e o rastreamento de possíveis usuários em perigo, perdidos ou feridos. Também foram previstas 16 transições, as quais estão associadas aos seguintes eventos:

- Passagem por etiqueta eletrônica associada à recepção de Petrópolis (*Petrópolis start*);
- Passagem por etiqueta eletrônica associada à recepção de Teresópolis (*Teresópolis start*);
- Passagem por etiqueta RFID associada ao ponto de parada do Castelos do Açú. Fronteira entre regiões 1 e 2 (*Castelos do Açú stop*);
- Passagem por etiqueta RFID associada ao ponto de parada da Pedra do Sino Fronteira entre regiões 2 e 3 (*Pedra do Sino stop*);
- Walker não alcança local desejado em intervalo de tempo esperado (*Time without trek progress*);
- Walker passa por ponto aguardado dentro do prazo definido (*New tag - trek progress*);

- Recebimento de mensagem solicitando busca local (*Lost/injured user - local search msg*);
- Chegada de resgate (*Rescue arrival*);
- Liberação para retornar à trilha (*Release to return to the trail*);
- Novo ferimento (função randômica) (*New hurt*);
- Cura espontânea (função randômica) (*Spontaneous healing*);
- Início da triagem (*Triage start*);
- Liberação da triagem (*Triage discharge*);
- Ponto final da trilha - recepção de Teresópolis (*Ongoing Exit*);
- Ponto final da trilha - recepção de Petrópolis (*Returning Exit*).

A Tab. 3.1 apresenta, de forma resumida, uma descrição dos lugares e transições utilizados na modelagem do agente *Walker*. O modelo comportamental possui a representação matemática descrita pelas Eqs. (3.1)-(3.4) e está graficamente apresentado na rede de Petri lugar/transição da Fig. 3.9. Transições com mensagens vinculadas serão abordadas no capítulo IV, ao apresentarem-se os modelos comportamentais em redes de Petri de alto nível.

Tabela 3.1 - Descrição dos lugares e transições utilizadas na modelagem do *Walker*.

Modelo	Lugar/Transição	Nomenclatura	Descrição
Walker	$P_{20}$	<i>Ready to trekking</i>	<i>Walker</i> pronto para iniciar trilha
	$P_{21} - P_{23}$	0%; 30%; 70% <i>Ongoing</i>	Distância percorrida (em %) no trecho Petrópolis-Teresópolis
	$P_{24} - P_{26}$	70%; 30%; 0% <i>Returning</i>	Distância percorrida (em %) no trecho Teresópolis-Petrópolis
	$P_{27}$	<i>On route mode</i>	<i>Walker</i> em rota
	$P_{28}$	<i>Lost mode</i>	<i>Walker</i> perdido
	$P_{29}$	<i>Rescued mode</i>	<i>Walker</i> resgatado
	$P_{30}$	<i>Healthy mode</i>	<i>Walker</i> saudável
	$P_{31}$	<i>Injured mode</i>	<i>Walker</i> ferido
	$P_{32}$	<i>Triage mode</i>	<i>Walker</i> passando por triage
	$P_{33}$	<i>Full trail</i>	Trilha concluída

	$T_{21}; T_{26}$	Petrópolis; Teresópolis <i>start</i>	Recepções de Petrópolis e Teresópolis, respectivamente
	$T_{22}; T_{24}$	Castelos do Açú <i>stop</i>	Ponto de parada do Castelos do Açú
	$T_{23}; T_{25}$	Pedra do Sino <i>stop</i>	Ponto de parada da Pedra do Sino
	$T_{27}$	<i>Time without trek progress</i>	<i>Walker</i> não alcança local desejado em intervalo de tempo esperado
	$T_{28}$	<i>New tag (trek progress)</i>	Walker passa por ponto aguardado dentro do prazo definido
	$T_{29}$	<i>Rescue arrival</i>	Chegada de resgate
	$T_{30}$	<i>Release to return to the trail</i>	Liberação para retornar à trilha
	$T_{31}$	<i>New hurt</i>	Novo ferimento (função randômica)
	$T_{32}$	<i>Spontaneous healing</i>	Cura espontânea (função randômica)
	$T_{33}$	<i>Triage start</i>	Início da triagem
	$T_{34}; T_{35}$	<i>Ongoing/Returning Exit</i>	Ponto final da trilha (recepção de Petrópolis ou Teresópolis, dependendo do sentido percorrido)

$$R = (P, T, I, O, M_0) \quad (3.1)$$

$$P = \{P_{20}, P_{21}, \dots, P_{33}\} \quad (3.2)$$

$$T = \{T_{21}, T_{22}, \dots, P_{35}\} \quad (3.3)$$

$$M_0 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0]^t \quad (3.4)$$

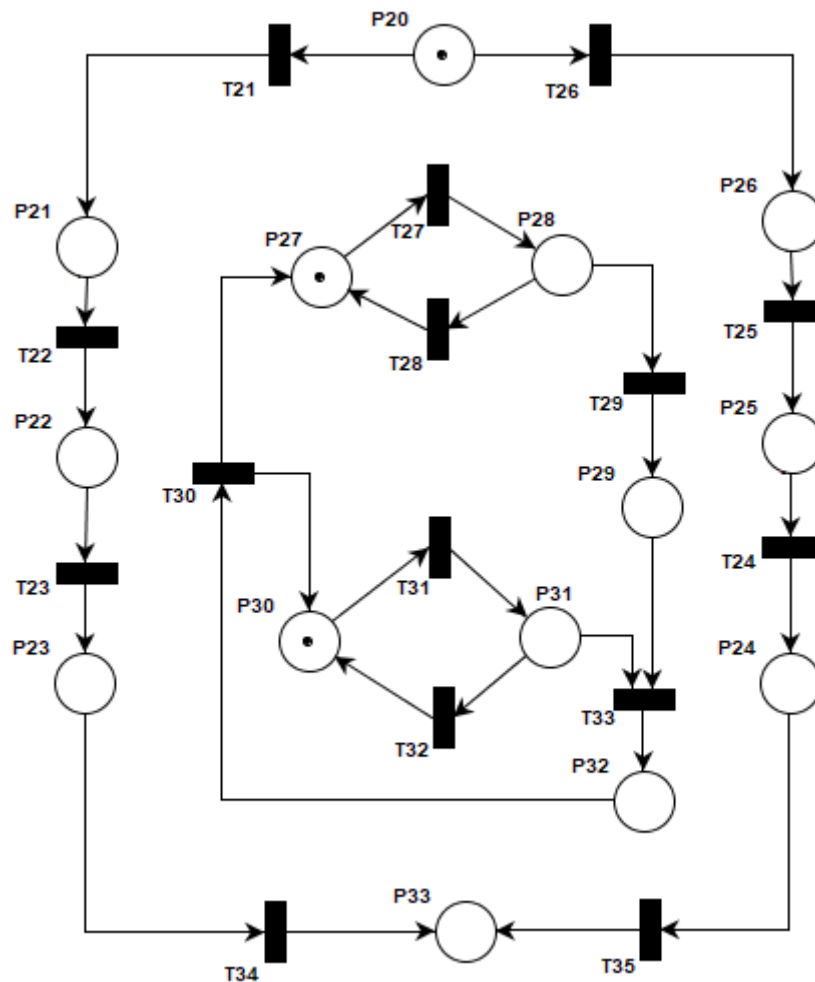


Figura 3.9 - Modelo comportamental do *Walker* em rede de Petri.

A iPNRD aplica-se à movimentação do *Walker* ao longo da trilha. Na rede da Fig. 3.9, este comportamento está associado ao conjunto de lugares  $\{P_{20}, P_{21}, \dots, P_{26}, P_{33}\}$  e ao conjunto de transições  $\{T_{21}, T_{22}, \dots, T_{26}, T_{34}, T_{35}\}$ . Os lugares indicam a posição do agente (Pronto para iniciar, Região 1, Região 2, Região 3 ou Trilha concluída), enquanto as transições estão associadas aos pontos estratégicos (Portarias de Petrópolis e Teresópolis, Castelos do Açú, e Pedra do Sino) onde são instaladas *tags* RFID. Os demais elementos da rede dizem respeito ao status do indivíduo na trilha (em rota, perdido, resgatado ou em triagem) e à sua condição de saúde.

### 3.5.2. *GroundBot*

Para o robô de salvamento terrestre, assumiram-se cinco modos de operação: modo disponível, modo busca local, modo auxílio, modo aproximação e modo resgate. Em modo disponível (*Available mode*), o veículo permanece preparado para atender qualquer

solicitação de busca, salvamento ou apoio terrestre. Caso o veículo terrestre receba uma mensagem de usuário ferido ou perdido, o mesmo entra no modo busca local (*Local search mode*), no qual permanece até localizar o indivíduo ou até receber uma mensagem abortando a busca. Em modo auxílio (*Aid mode*), o *GroundBot* atende uma solicitação de apoio enviada por outro agente SAR. Este modo é típico para buscas em grupo, resgate em grupo, entrega de mantimentos ou medicamento, entre outros. O veículo permanecerá neste modo até que uma mensagem dispensando o apoio seja recebida. Quando em modo busca local, ao localizar e identificar o indivíduo de interesse, o *GroundBot* entrará em modo Aproximação (*Approach*). Este modo é caracterizado por um movimento diferenciado, uma vez que a vítima pode encontrar-se em região vulnerável e de difícil acesso. Ao alcançar uma posição adequada, o veículo terrestre pode entrar em modo Resgate (*Rescue*), no qual permanecerá até que os procedimentos necessários sejam concluídos. Cada modo de operação descrito está associado a um lugar na rede de Petri. Há ainda três outros lugares associados à posição do veículo na trilha: Região 1 (entre a portaria de Petrópolis e o ponto de pernoite Castelos do Açú), Região 2 (entre os pontos de pernoite do Castelos do Açú e da Pedra do Sino) e Região 3 (entre o ponto de pernoite da Pedra do Sino e a portaria de Teresópolis). Também foram previstas 12 transições, as quais estão associadas aos seguintes eventos:

- Recebimento de mensagem solicitando busca local (*Lost/injured user - local search msg*);
- Recebimento de mensagem cancelando busca local (*Dismiss local search*);
- Recebimento de mensagem solicitando auxílio (*Request aid message*);
- Recebimento de mensagem cancelando auxílio (*Dismiss aid message*);
- Alvo identificado (*Target identified*);
- Alvo alcançado (*Target reached*);
- Resgate concluído (*Rescue completed*);
- Passagem por etiqueta RFID associada ao ponto de parada do Castelos do Açú. Fronteira entre regiões 1 e 2 (Castelos do Açú *totem*);
- Passagem por etiqueta RFID associada ao ponto de parada da Pedra do Sino Fronteira entre regiões 2 e 3 (Pedra do Sino *totem*).

A Tab. 3.2 apresenta, de forma resumida, uma descrição dos lugares e transições utilizados na modelagem do *GroundBot*. O modelo comportamental possui a representação matemática descrita pelas Eqs. (3.5)-(3.8) e está graficamente apresentado na rede de Petri lugar/transição da Fig. 3.10. As transições com mensagens vinculadas também serão abordadas no capítulo IV.



Tabela 3.2 - Descrição dos lugares e transições utilizadas na modelagem do *GroundBot*.

Modelo	Lugar/Transição	Nomenclatura	Descrição
<i>GroundBot</i>	$P_{11}$	<i>Available mode</i>	<i>GroundBot</i> disponível
	$P_{12}$	<i>Local search mode</i>	<i>GroundBot</i> em modo busca local
	$P_{13}$	<i>Aid mode</i>	<i>GroundBot</i> em modo auxílio
	$P_{14}$	<i>Approach mode</i>	<i>GroundBot</i> em modo aproximação
	$P_{15}$	<i>Rescue mode</i>	<i>GroundBot</i> em modo resgate
	$P_{16}$	<i>Initial state</i>	Estado inicial
	$P_{17}$	<i>GroundBot on Region 1</i>	<i>GroundBot</i> na região 1
	$P_{18}$	<i>GroundBot on Region 2</i>	<i>GroundBot</i> na região 2
	$P_{19}$	<i>GroundBot on Region 3</i>	<i>GroundBot</i> na região 3
	$T_{11}$	<i>Lost/injured user - local search msg</i>	Recebimento de mensagem solicitando busca local
	$T_{12}$	<i>Dismiss local search</i>	Recebimento de mensagem cancelando busca local
	$T_{13}$	<i>Request aid message</i>	Recebimento de mensagem solicitando auxílio
	$T_{14}$	<i>Dismiss aid message</i>	Recebimento de mensagem cancelando auxílio
	$T_{15}$	<i>Target identified</i>	Alvo identificado
	$T_{16}$	<i>Target reached</i>	Alvo alcançado
	$T_{17}$	<i>Rescue completed</i>	Resgate concluído
	$T_{18}$	<i>System start</i>	Início do sistema
	$T_{19}; T_{191}$	Castelos do Açú <i>totem</i>	Fronteira entre regiões 1 e 2 – Castelos do Açú
	$T_{20}; T_{201}$	Pedra do Sino <i>totem</i>	Fronteira entre regiões 2 e 3 – Pedra do Sino

$$R = (P, T, I, O, M_0) \quad (3.5)$$

$$P = \{P_{11}, P_{12}, \dots, P_{19}\} \quad (3.6)$$

$$T = \{T_{11}, T_{12}, \dots, T_{20}, T_{191}, T_{201}\} \quad (3.7)$$

$$M_0 = [0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0]^t \quad (3.8)$$

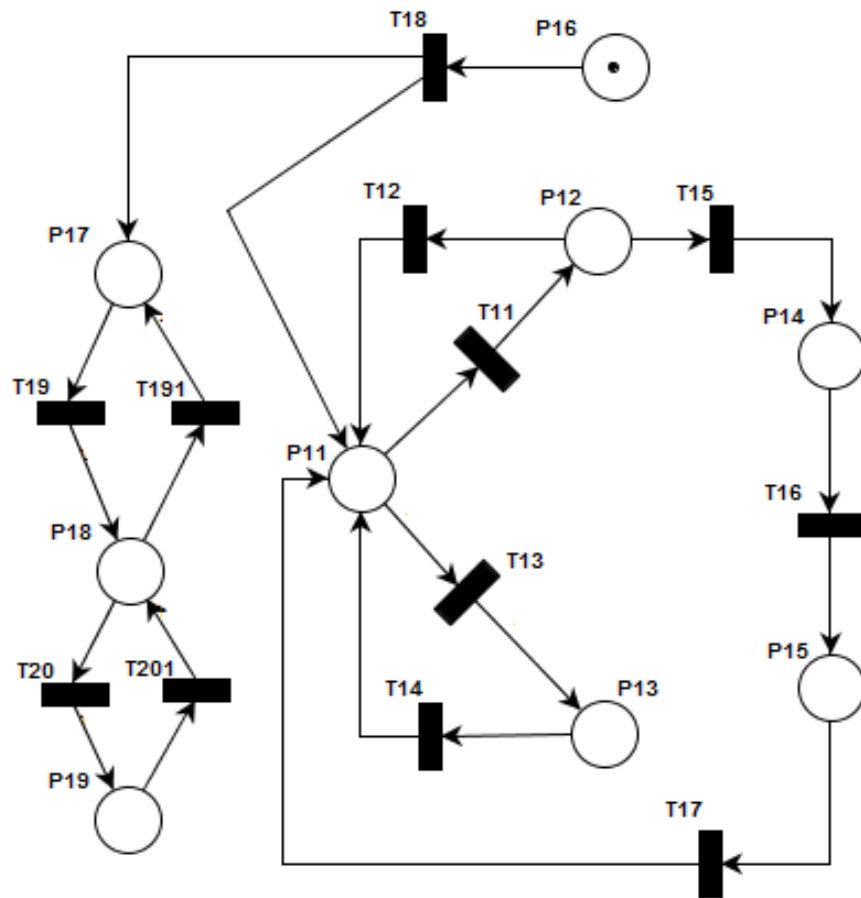


Figura 3.10 - Modelo comportamental do *GroundBot* em rede de Petri.

A iPNRD aplica-se à movimentação do *GroundBot* ao longo da trilha. Na rede da Fig. 3.13, este comportamento está associado ao conjunto de lugares  $\{P_{17}, P_{18}, P_{19}\}$  e ao conjunto de transições  $\{T_{19}, T_{20}, T_{191}, T_{201}\}$ . Os lugares indicam a posição do agente SAR (Região 1, Região 2 ou Região 3), enquanto as transições estão associadas aos pontos estratégicos (Castelos do Açú, e Pedra do Sino) onde são posicionadas *tags* RFID. Os demais elementos da rede dizem respeito ao modo de operação do agente SAR (disponível, em modo busca, em modo auxílio, em modo aproximação ou em modo resgate).

### 3.5.3. *AerialBot*

Para o robô aéreo, assumiram-se quatro modos de operação: modo patrulha, modo auxílio, modo disponível e aguardando chegada e auxílio terrestre. Em modo patrulha (*Patrol mode*), o veículo executa uma trajetória pré-definida, podendo interromper esta tarefa ao perceber usuários perdidos ou em perigo. Neste caso, o agente solicita apoio terrestre e entra no modo espera (*On hold mode*), no qual permanece até a chegada do auxílio em terra. Em modo auxílio (*Aid mode*), o *AerialBot* atende uma solicitação de apoio de algum agente terrestre. Este tipo de comportamento é comum em casos de entrega de mantimentos, entrega de medicamentos, iluminação aérea, entre outros. Uma vez neste modo, o veículo permanecerá até que uma mensagem dispensando o apoio seja recebida. Sempre que finalizar uma patrulha e não receber solicitação de apoio, o *AerialBot* entrará no modo Disponível (*Available mode*), no qual permanecerá até que seja necessário novo patrulhamento ou até que o mesmo receba uma mensagem solicitando apoio aéreo. Cada modo de operação está associado a um lugar na rede de Petri. As transições da rede estão associadas aos seguintes eventos:

- Recebimento de mensagem solicitando auxílio (*Request aid msg*);
- Recebimento de mensagem dispensando auxílio (*Dismiss aid msg*);
- Ronda concluída (*Patrol completed*);
- Tempo de espera finalizado (*Idle time*);
- Percepção de perigo (*Hazard perception*);
- Chegada de auxílio terrestre (*Aid arrived*).

A Tab. 3.3 apresenta, de forma resumida, uma descrição dos lugares e transições utilizados na modelagem do *AerialBot*. O modelo comportamental possui a representação matemática descrita pelas Eqs. (3.9)-(3.12) e está graficamente apresentado na rede de Petri lugar/transição da Fig. 3.11. As transições com mensagens vinculadas também serão abordadas no capítulo IV.

$$R = (P, T, I, O, M_0) \quad (3.9)$$

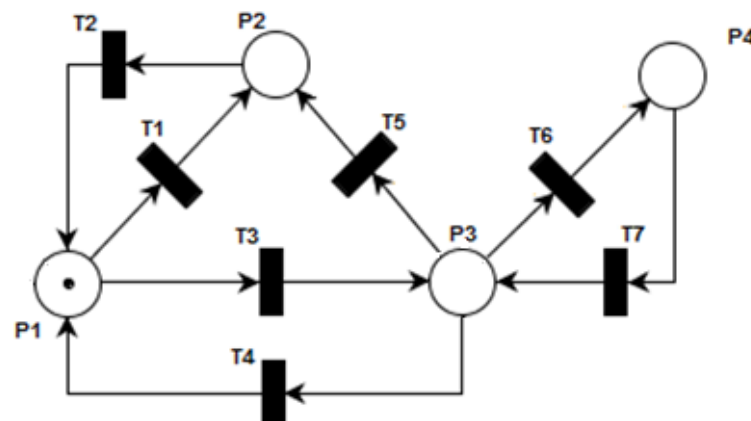
$$P = \{P_1, P_2, P_3, P_4\} \quad (3.10)$$

$$T = \{T_1, T_2, T_3, T_4, T_5, T_6, T_7\} \quad (3.11)$$

$$M_0 = [1 \ 0 \ 0 \ 0]^t \quad (3.12)$$

Tabela 3.3 - Descrição dos lugares e transições utilizadas na modelagem do *AerialBot*.

Modelo	Lugar/Transição	Nomenclatura	Descrição
<i>AerialBot</i>	$P_1$	<i>Available mode</i>	<i>AerialBot</i> disponível
	$P_2$	<i>Aid mode</i>	<i>AerialBot</i> em modo auxílio
	$P_3$	<i>Patrol mode</i>	<i>AerialBot</i> em modo patrulha
	$P_4$	<i>On hold mode</i>	<i>AerialBot</i> aguardando chegada de auxílio terrestre
	$T_1$	<i>Request aid msg</i>	Recebimento de mensagem solicitando auxílio
	$T_2$	<i>Dismiss aid msg</i>	Recebimento de mensagem dispensando auxílio
	$T_3$	<i>Idle time</i>	Tempo de espera finalizado
	$T_4$	<i>Patrol completed</i>	Ronda concluída
	$T_5$	<i>Request aid msg</i>	Recebimento de mensagem solicitando auxílio
	$T_6$	<i>Hazard perception</i>	Percepção de perigo dada pelo sistema de visão
	$T_7$	<i>Aid arrived</i>	Chegada de auxílio terrestre

Figura 3.11 - Modelo comportamental do *AerialBot* em rede de Petri.

### 3.6. Definição da Arquitetura de Controle

Devido a multidisciplinaridade da robótica móvel e de sistemas multirrobo, o sucesso das aplicações depende de diversos fatores e áreas de conhecimento distintas. Neste sentido, seria pretencioso e demasiadamente oneroso desenvolver um sistema multirrobo que focasse em todos os pontos descritos na seção 2.1.1.

Almejando-se manipular atitudes dos robôs através de modelos comportamentais desenvolvidos em redes de Petri e, assim, orientar o comportamento dos agentes em campo, presume-se que funcionalidades fundamentais de robôs móveis, como navegação, mapeamento, localização e controle de trajetória estejam bem desenvolvidas. Ratificando o que já foi exposto, por dar enfoque no comportamento estratégico dos robôs, este trabalho não aborda modelos cinemáticos e dinâmicos ou, até mesmo, o próprio sensorimento e atuação em níveis de controle reativo.

Por razões de aplicabilidade, a infraestrutura de controle proposta foi idealizada de modo a permitir que os modelos em redes de Petri fossem capazes de comandar o comportamento dos agentes de campo, sem a necessidade de abstrair uma representação diagramática em linguagem textual. Isto foi possível através da utilização do CPN Tools associado à biblioteca de comunicação COMMS/CPN (GALLASCH; KRISTENSEN, 2001), a qual permite aos modelos ali desenvolvidos, comunicação com aplicações e processos externos através do protocolo TCP/IP.

A arquitetura de controle proposta é baseada na pirâmide da automação, apresentada na Fig. 3.12. Os níveis estratégicos de execução da manufatura (MES) e planejamento de recursos (ERP) não são escopo deste projeto, mas sim os níveis de supervisão, controle, atuação e sensorimento. A Fig. 3.13 ilustra como cada módulo da arquitetura de controle proposta se relaciona com os três primeiros níveis da pirâmide da automação. Um esquemático com o detalhamento da interação entre os processos pode ser visualizado na Fig. 3.14.



Figura 3.12 - Representação clássica da arquitetura conhecida como pirâmide da automação. Adaptado de SMC (2017).

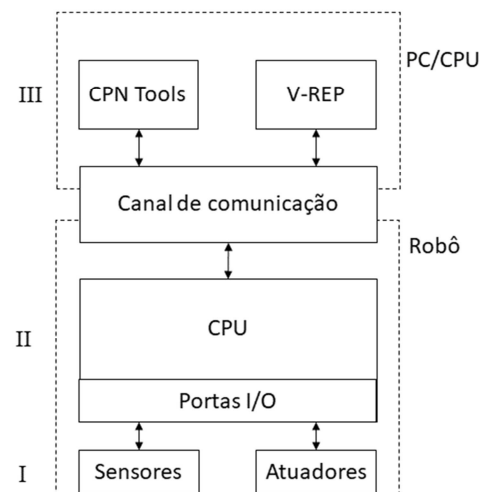
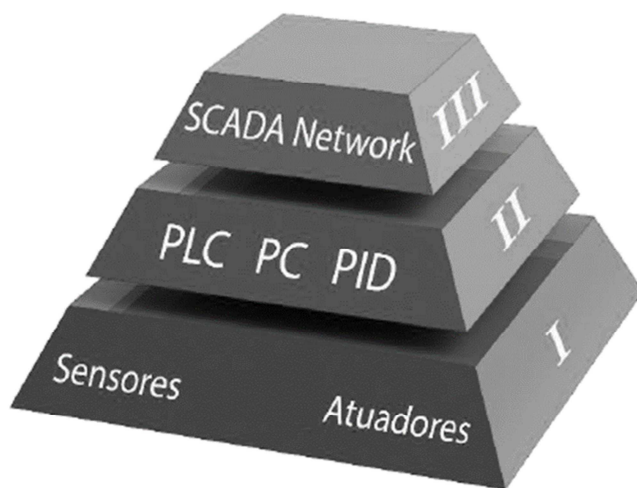


Figura 3.13 - Relacionamento entre os três primeiros níveis da pirâmide da automação e a arquitetura de controle proposta.

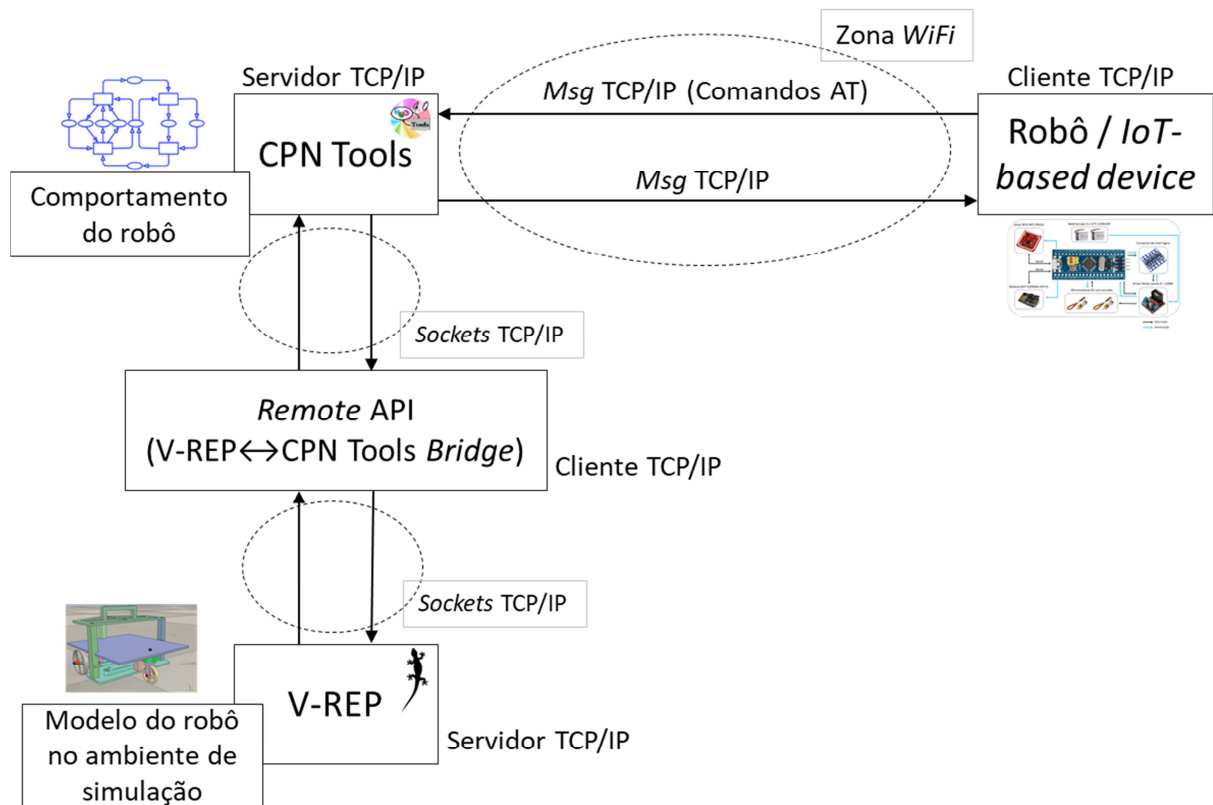


Figura 3.14 - Esquemático da arquitetura de controle proposta com detalhamento da interação entre os processos.

Os modelos comportamentais gerados no CPN Tools foram configurados para possibilitar o acesso de processos externos. Utilizaram-se dois tipos de processos: a plataforma de simulação V-REP (COPPELIA ROBOTICS, 2018) e uma placa de desenvolvimento, popularmente denominada “*BluePill*”.

Quando comparada às placas Arduino tradicionais, como UNO, MEGA ou NANO, a “*BluePill*” destaca-se no custo e poder de processamento. Ela opera com um microcontrolador de 32 bits da família STM32F103 (TECHSHOP, 2017), baseado na arquitetura ARM Cortex-M3. A comunicação entre dispositivos móveis e o CPN Tools se deu através de uma rede wireless, por meio do protocolo TCP/IP. A Fig. 3.15 ilustra uma representação esquemática da arquitetura proposta.

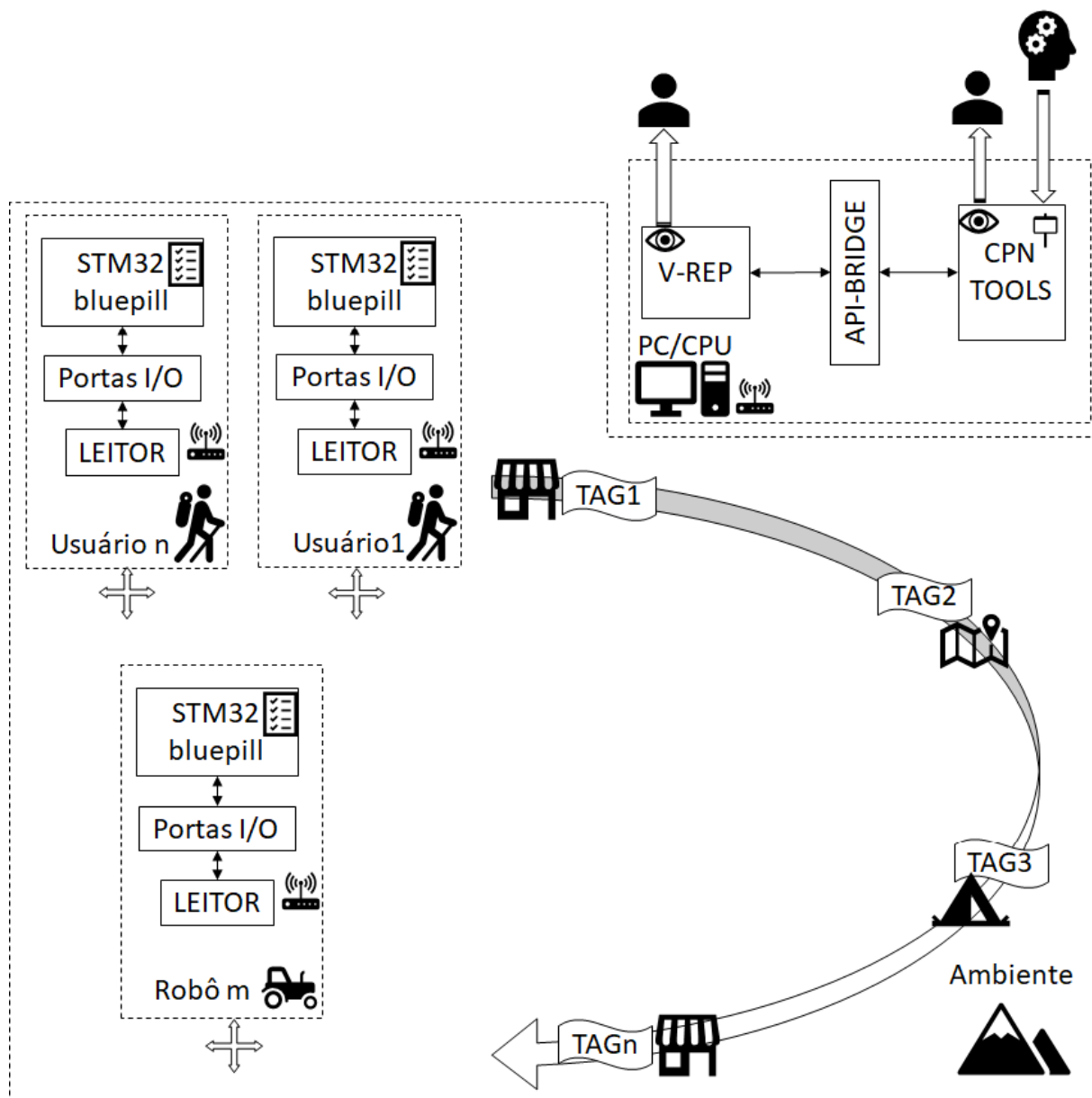


Figura 3.15 - Esquema ilustrativo do modo de operação do sistema proposto.

### 3.6.1. Configuração do CPN Tools

A biblioteca COMMS/CPN (GALLASCH; KRISTENSEN, 2001) permite que comunicação TCP/IP entre o CPN Tools e aplicativos e processos externos.

As principais funções desta biblioteca que permitem a configuração do CPN Tools para comunicação externa estão descritas a seguir (CPN TOOLS, 2018):

- **ConnManagementLayer.acceptConnection:** `string * int → unit`

Fornece comportamento de servidor e permite que os processos externos se conectem ao CPN Tools. O primeiro argumento deve ser uma *string* exclusiva a ser associada à conexão. O segundo argumento é um número de porta do tipo inteiro. A função



verifica se o identificador de conexão é exclusivo e depois escuta na porta fornecida para solicitações de conexão recebidas. Isso faz com que o CPN Tools seja bloqueado até receber uma solicitação de conexão recebida. Quando isso acontece, uma conexão é estabelecida com o processo externo que solicita a conexão.

- **ConnManagementLayer.send:** `string * 'a * ('a → Word8Vector.vector) → unit`

Permite que o usuário envie qualquer tipo de dado para processos externos. A função é polimórfica, no sentido de que os dados transmitidos para o envio podem ser de qualquer tipo, incluindo tipos definidos pelo usuário. Três parâmetros são passados para esta função como entrada: O primeiro é uma *string* identificando a conexão, o segundo é o dado a enviar e o terceiro é uma função para codificar os dados a serem enviados. O objetivo da função de codificação é codificar os dados para enviar para uma sequência de bytes. Isso permite que os dados sejam de qualquer tipo, desde que exista uma função de codificação para esse tipo. Exemplos de funções de codificação disponibilizadas pelo CPN Tools são *stringEncode* e *integerEncode*.

- **ConnManagementLayer.receive:** `string * (Word8Vector.vector → 'a) → 'a`

Permite que o usuário receba qualquer tipo de dado de um processo externo. A função de recepção é polimórfica da mesma maneira que a função **ConnManagementLayer.send**. Os parâmetros para esta função são uma *string* identificando uma conexão e uma função de decodificação, para decodificar o vetor de bytes recebidos para o tipo de dados apropriado. Os dados decodificados resultantes são retornados. Exemplos de funções de decodificação disponibilizadas pelo CPN Tools são *stringDecode* e *integerDecode*.

- **ConnManagementLayer.closeConnection:** `string → unit`

Permite que o usuário feche uma conexão. Uma *string* que identifica a conexão a ser fechada é passado como argumento para esta função.

Um exemplo de utilização destas funções é ilustrado na Fig. 3.16. Inicialmente a transição “Conectar” encontra-se habilitada devido à existência de uma ficha no lugar “Desconectado”. Ao disparar a transição “Conectar”, a função **acceptConnection** (“Conn1”, 9000) fornece comportamento de servidor ao CPN Tools, iniciando a conexão “Conn1” na porta 9000. Quando um cliente externo solicita conexão, uma ficha é retirada do lugar “Desconectado” e outras são inseridas nos lugares “Conectado” e “Recuando”. A partir deste momento as transições “Avançar” e “Desconectar” estarão habilitadas. Ao disparar-se a transição “Avançar”, o CPN Tools executa a função **send** (“Conn1”, “Go”, *stringEncode*), codificando e enviando a *string* “Go” para o processo externo através da conexão “Conn1”. Isto retira uma ficha do lugar “Recuando” e novas são

inseridas nos lugares “Avancando” e “Aguardando”. A partir deste momento as transições “Ler” e “Desconectar” estarão habilitadas. Ao disparar-se a transição “Ler”, o CPN Tools executa a função `receive("Conn1",integerDecode)`, decodificando o dado disponível na conexão “Conn1” e disponibilizando-o à rede em execução como um valor do tipo INT. Neste momento o valor recebido é comparado com um valor esperado. Caso a comparação obtenha sucesso, uma ficha é retirada do lugar “Aguardando” e outra é inserida no local “DadoOk”. Caso o valor lido não seja o valor esperado, o disparo da transição “Ler” consome uma ficha do lugar “Aguardando” e insere uma nova no mesmo lugar. Caso haja fichas nos lugares “Avancando” e “DadoOk”, a transição “Recuar” estará habilitada. Ao disparar-se transição “Recuar”, o CPN Tools executa a função `send("Conn1","Back",stringEncode)`, codificando e enviando a *string* “Back” para o processo externo através da conexão “Conn1”. Isto retira uma ficha do lugar “Avancando” e insere outra no lugar “Recuando”. A qualquer momento em que a transição “Desconectar” for disparada, a função `closeConnection("Conn1")` fechará a conexão “Conn1” e interromperá a comunicação com qualquer processo externo que utilizasse tal conexão. Isto retirará uma ficha do lugar “Conectado” e inserirá outra no lugar “Desconectado”.

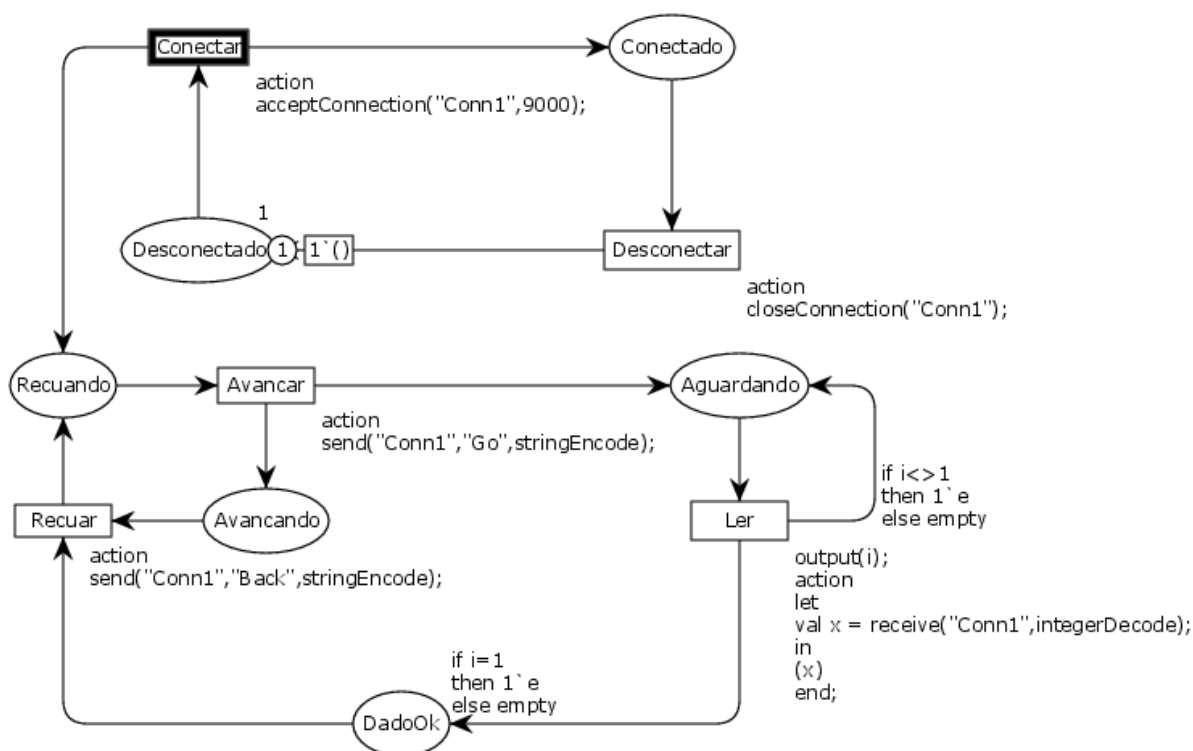


Figura 3.16 - Exemplo de uma rede ilustrando a utilização das funções COMMS/CPN.

Analisando melhor a Fig. 3.16, nota-se que a rede de Petri principal é composta por apenas dois lugares (“Recuando” e “Avançando”) e duas transições (“Avançar” e “Recuar”). Na Fig. 3.17 destacam-se os elementos de conexão e de comunicação necessários para que a rede de Petri principal seja executável em processos externos. A transição “Avançar” envia um comando e aguarda retorno do ambiente, por exemplo um sinal do sensor de proximidade. A transição “Recuar”, por sua vez, apenas envia e não aguarda informação dos sensores. Este simples exemplo evidencia o aumento na complexidade e dificuldade na leitura e entendimento da rede quando adicionados elementos de conexão e comunicação.

Uma maneira de melhorar a legibilidade da rede é lançando mão das funções de redes de Petri hierárquicas, disponibilizadas pelo CPN Tools. Esta solução será melhor discutida na seção 4.12.

### 3.6.2. Configuração do microcontrolador via Arduino IDE

A STM32 *BluePill* opera com um microcontrolador de 32 bits baseado na arquitetura ARM Cortex-M3, tornando-o superior às tradicionais placas Arduino. Há quem diga que seu poder de processamento pode ser comparado ao do Arduino DUE, mas com dimensões tão pequenas quanto às do Arduino Nano. Uma fotografia da placa STM32 *BluePill* é apresentada na Fig. 3.18.

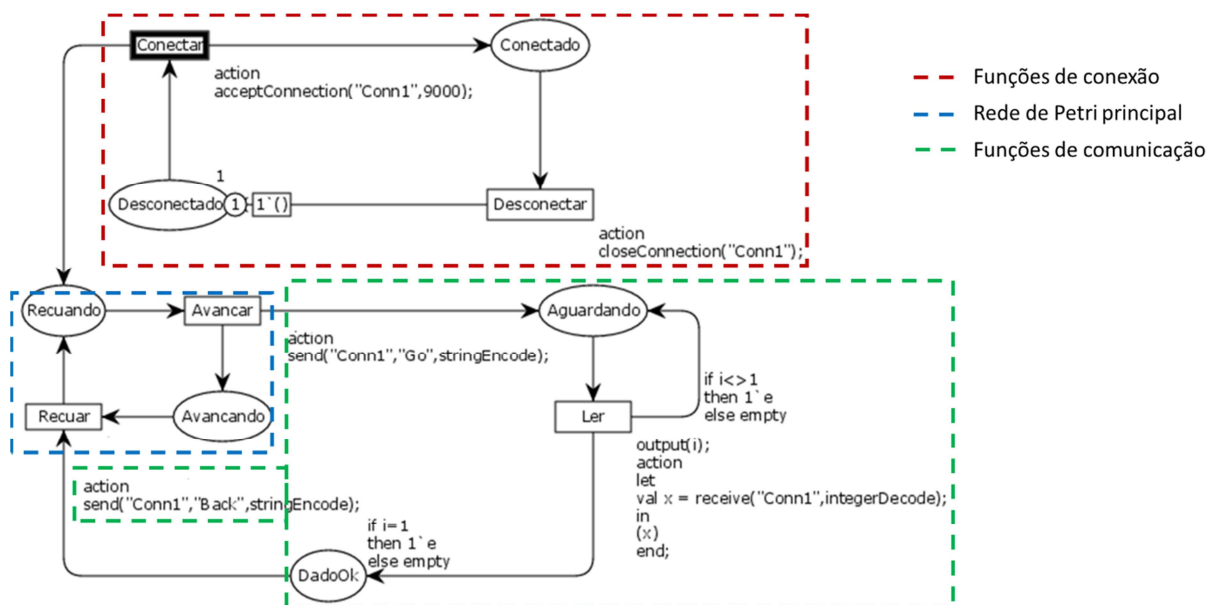


Figura 3.17 - Detalhe dos elementos de conexão e de comunicação na rede CPN.



Figura 3.18 - Fotografia da placa STM32 *BluePill* (STM32DUINO, 2018).

Apesar de não ser diretamente suportada pela Arduino IDE, a *BluePill* utiliza um microcontrolador de arquitetura semelhante à do Arduino DUE e pode ser acrescentada como um novo hardware dentro das configurações da IDE. Grande parte das informações sobre a configuração da Arduino IDE, de modo a oferecer uma interface de programação para placas *BluePill*, podem ser encontradas na página [stm32duino.com](http://stm32duino.com).

Uma vez configurada, é possível desenvolver programas na Arduino IDE e enviá-los via porta serial para a placa STM32 *BluePill*. A conexão com processos externos, como a CPN Tools é possível através da utilização de módulos wireless como o ESP8266. Desta forma, é possível programar o dispositivo móvel como um cliente TCP/IP que envia e recebe mensagens do servidor CPN Tools.

### 3.6.3. Configuração do simulador V-REP

Os modelos de robôs utilizados neste trabalho foram idealizados em projetos finais do curso de engenharia mecatrônica da Universidade Federal de Uberlândia. Rodrigues (2017) idealizou o robô cartesiano apresentado na Fig. 3.19, enquanto o robô móvel apresentado na Fig. 3.20 foi idealizado por Silva (2017a).

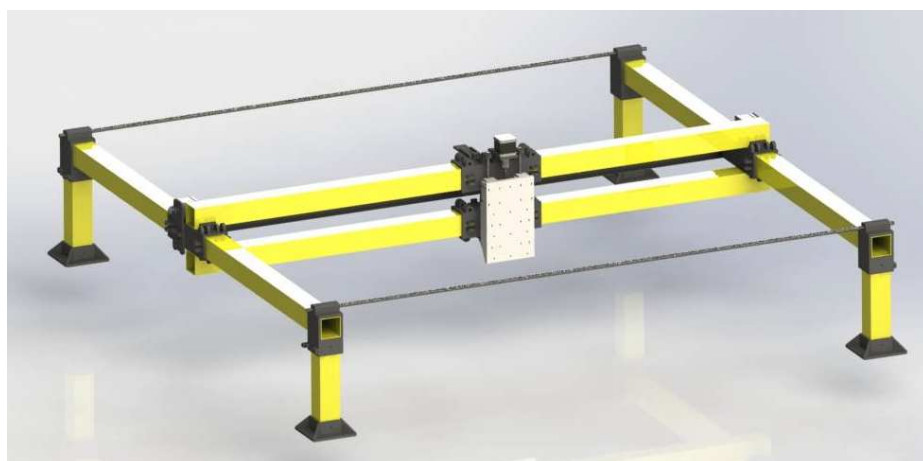


Figura 3.19 - Modelo CAD do robô cartesiano apresentado por Rodrigues (2017).

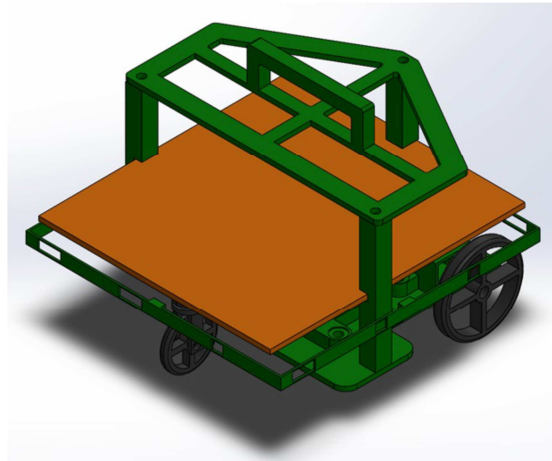


Figura 3.20 - Modelo CAD do robô móvel apresentado por Silva (2017a).

Ao construir um novo modelo, deve-se preocupar inicialmente com o seu aspecto visual. Pode-se criar formas primitivas em V-REP ou importar uma forma de uma aplicação externa. Formas primitivas são mais adequadas para o cálculo da resposta de colisão dinâmica. Ao importar dados CAD de um aplicativo externo, o mais importante é garantir que o modelo CAD não seja muito pesado, ou seja, que a malha não contenha muitos triângulos. Esse requisito é importante, pois um modelo pesado será lento na exibição, podendo também desacelerar vários módulos de cálculo que podem ser usados em um estágio posterior (por exemplo, cálculo de distância mínima ou dinâmica).

Algumas ferramentas, como *Decimate the mesh* e *Extract the convex hull*, permitem simplificar as malhas de objetos importados. Coppelia Robotics (2018) geralmente recomenda modelar um robô com malhas de 5 a 10 mil triângulos. Recomenda-se não ultrapassar 20 mil triângulos, pois acima disso uma cena de simulação pode rapidamente se tornar muito lenta. A Fig. 3.21 ilustra uma representação simplificada do robô móvel de Silva (2017a) antes e após a aplicação da ferramenta *Extract the convex hull*.

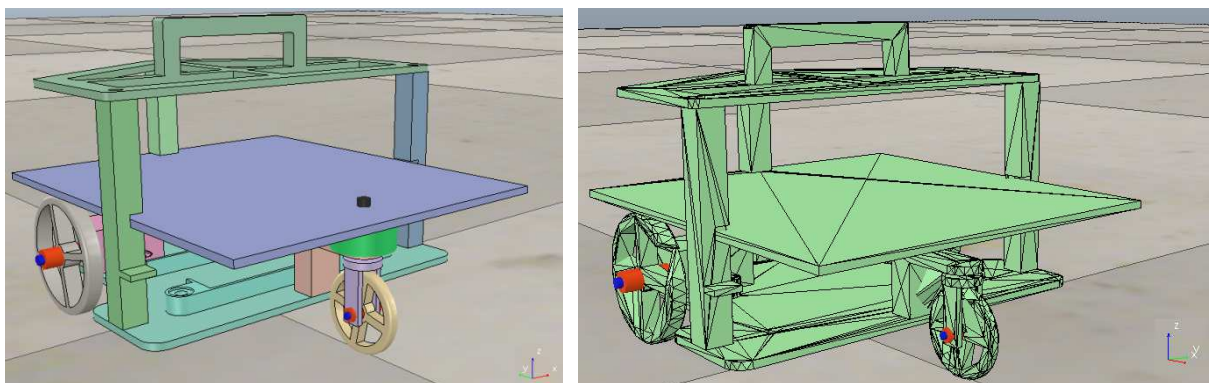


Figura 3.21 - Modelo do robô móvel (antes e após a aplicação da ferramenta *Extract the convex hull*) em ambiente de simulação V-REP.

A Fig. 3.22 detalha uma representação simplificada do robô cartesiano de Rodrigues (2017), inserida no ambiente V-REP, já com a simplificação das formas através da ferramenta *Extract the convex hull*.

Existem diversas maneiras de controlar um robô virtual inserido no ambiente V-REP (COPPELIA ROBOTICS, 2018):

- A maneira mais conveniente é escrever um *child script* associado ao comportamento de um determinado robô ou modelo.
- Outra maneira seria desenvolver um *plugin*, garantindo registro personalizado de funções Lua e acesso a bibliotecas de funções externas.
- Uma terceira maneira de controlar um robô ou uma simulação é criando um aplicativo cliente externo que acessa uma API remota. Maneira muito conveniente e relativamente fácil para executar códigos de controle de um aplicativo externo. Atualmente, existem sete idiomas suportados: C/C++, Python, Java, Matlab, Octave, Lua e Urbi.
- A quarta maneira é através de um ROS *node*. A plataforma ROS permite comunicação de vários processos distribuídos. Uma API remota destaca-se por ser muito leve e rápida, entretanto é capaz de comunicar-se apenas com o V-REP. O ROS, por outro lado, permite conectar praticamente qualquer número de processos entre si, além de disponibilizar uma grande quantidade de bibliotecas compatíveis, o que o torna mais pesado e mais complicado comparado a uma API remota.
- Uma quinta maneira de controlar um robô ou uma simulação é desenvolvendo um aplicativo externo capaz de se comunicar com um *plugin* ou um *script* V-REP por vários meios (por exemplo, *tubes*, *sockets*, porta serial, etc.). Duas vantagens deste método são a escolha da linguagem de programação e a flexibilidade. Neste caso também, o código de controle pode ser executado em um robô ou em um computador diferente. Esta maneira de controlar uma simulação ou um modelo é, no entanto, mais tediosa que o método com a API remota.

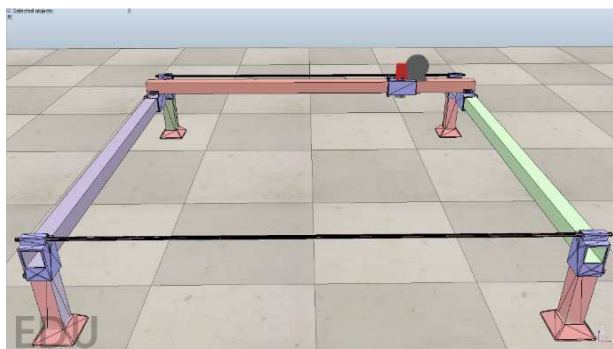


Figura 3.22 - Modelo do robô cartesiano em ambiente de simulação V-REP.

A escolha da maneira mais adequada dependerá das características de cada aplicação e das preferências do desenvolvedor. Para implementar uma API remota, é fundamental saber que esta é formada por duas entidades que se comunicam via socket (COPPELIA ROBOTICS, 2018): Uma aplicação cliente (desenvolvida em uma das sete linguagens suportadas) e um servidor (disponibilizado pelo próprio V-REP de modo contínuo ou temporariamente a cada nova simulação inicializada). Uma centena de funções específicas podem ser chamadas pela aplicação cliente, a qual interage com o servidor V-REP de forma oculta ao usuário, reduzindo atrasos e o carregamento da rede. A API remota pode permitir que uma ou várias aplicações externas interajam com o V-REP (servidores distintos operam em portas diferentes) de modo síncrono ou assíncrono (assíncrono por padrão).

As principais funções e ferramentas utilizadas para o desenvolvimento do ambiente no V-REP estão indicadas na Tab. 3.4. A documentação com o detalhamento para estas e outras funções está disponível no manual do usuário (COPPELIA ROBOTICS, 2018).

Tabela 3.4 - Lista das principais funções e ferramentas utilizadas para desenvolvimento deste trabalho na plataforma V-REP.

<b>Funções Regulares</b>	<b>Funções API Remota</b>	<b>Ferramentas</b>
<i>simGetObjectHandle</i>	<i>simxStart</i>	<i>Collection</i>
<i>simGetFloatSignal</i>	<i>simxGetObjectHandle</i>	<i>PathPlanning</i>
<i>simReadProximitySensor</i>	<i>simxStartSimulation</i>	<i>ExtractConvexHull</i>
<i>simClearStringSignal</i>	<i>simxStopSimulation</i>	<i>Group/Ungroup shapes</i>
<i>simSetJointTargetVelocity</i>	<i>simxGetFloatSignal</i>	
<i>simSendData</i>	<i>simxReadProximitySensor</i>	
<i>simReceiveData</i>	<i>simxClearStringSignal</i>	
<i>simGetJointPosition</i>	<i>simxSetJointTargetVelocity</i>	
<i>simSetJointTargetPosition</i>	<i>simxGetLastCmdTime</i>	
<i>simSetJointTargetVelocity</i>	<i>simxFinish</i>	
<i>simGetPathPlanningHandle</i>	<i>simxGetJointPosition</i>	
<i>simGetCollectionHandle</i>	<i>simxSetJointTargetPosition</i>	
<i>simGetSimulationTime</i>	<i>simxSetJointTargetVelocity</i>	
<i>simCheckCollision</i>		
<i>simPerformPathSearchStep</i>		
<i>simGetObjectOrientation</i>		
<i>simGetObjectOrientation</i>		
<i>simSetObjectOrientation</i>		
<i>simInitializePathSearch</i>		
<i>simPerformPathSearchStep</i>		
<i>simGetPathLength</i>		
<i>simGetObjectPosition</i>		
<i>simGetPositionOnPath</i>		
<i>simGetObjectMatrix</i>		
<i>simGetInvertedMatrix</i>		
<i>simMultiplyVector</i>		



## CAPÍTULO IV

### IMPLEMENTAÇÃO E TESTES

Este projeto aborda uma nova arquitetura de controle baseada em redes de Petri de alto nível e na utilização da iPNRD integradas a um sistema multirrobô aplicado em operações de busca e salvamento em trilhas e travessias. As etapas de implementação e testes estão apresentadas nas próximas seções. A implementação das redes de Petri de alto nível em CPN Tools associadas aos modelos comportamentais dos agentes *AerialBot*, *GroundBot* e *Walker* está descrita na seção 4.1. Toda a parte de implementação dos cenários de simulação em V-REP, a programação dos robôs e a integração da API remota com os processos do CPN Tools e do V-REP estão apresentados na seção 4.2. A seção 4.3 aborda a parte de implementação experimental com os dispositivos eletrônicos embarcados no robô terrestre. Finalmente, a seção 4.4 aborda a etapa de testes funcionais entre as redes de Petri de alto nível em CPN Tools e os robôs virtuais em V-REP via API-Bridge e entre as redes de Petri de alto nível em CPN Tools e os dispositivos móveis via ESP8266 e STM32 *BluePill*.

#### 4.1. Implementação em CPN Tools

Os modelos comportamentais dos agentes do sistema foram inicialmente concebidos através das redes de Petri lugar/transição apresentadas na seção 3.5. Estas redes puderam ser analisadas e verificadas quanto a sua operação e movimentação de fichas ao longo dos lugares. A aplicação do comportamento característico da rede em seu respectivo agente de campo está normalmente condicionada à abstração da representação diagramática em uma linguagem de programação do próprio robô.

Para que modelos comportamentais desenvolvidos em uma linguagem diagramática pudessem, de fato, controlar os agentes de campo, fez-se necessária uma arquitetura de

controle que integrasse o comportamento do modelo com as atitudes dos robôs. Para isso, modelos equivalentes aos da seção 3.5 foram concebidos em redes de Petri de alto nível através da CPN Tools, cujas funções de alto nível (como *acceptConnection*, *closeConnection*, *send* e *receive*) permitiram que informações provenientes dos modelos comportamentais alcançassem os robôs (ou os seus respectivos modelos em V-REP) e que as informações provenientes destes últimos pudessem ser lidas e interpretadas pelos modelos desenvolvidos em CPN Tools.

Basicamente, a arquitetura de controle demanda funções de comunicação em cada uma das transições do modelo. A inserção de novos elementos aumenta a complexidade das redes, comprometendo sua legibilidade e entendimento, situação já exemplificada na Fig. 3.17.

Além das funções de comunicação com processos externos, o CPN Tools oferece recursos de redes de Petri hierárquicas. Para melhor legibilidade e entendimento dos modelos desenvolvidos lançou-se mão da utilização das redes hierárquicas através das transições de substituição. Como cada transição de substituição possui uma sub-rede associada, as funções de comunicação puderam ficar ocultas às redes de Petri principais.

A rede de Petri com o modelo comportamental do *AerialBot* está apresentada na Fig. 4.1. A Fig. 4.2 ilustra a sub-rede associada à transição de substituição “*MsgApoio*”. Os modelos comportamentais do *GroundBot* e do *Walker* estão apresentados nas redes da Fig. 4.3 e Fig. 4.4. Todas as transições de substituição têm suas sub-redes detalhadas no Apêndice A.

Basicamente cada lugar associa-se a um modo de operação, que contém funções de baixo nível (executadas no microcontrolador do agente robótico). O lugar “*BuscaLocal*” no agente *GroundBot*, por exemplo, envolve funções referentes à operação de busca, tais como percepção, navegação, desvio de obstáculos, leitura das *tags* através da iPNRD e análise do histórico das mesmas para posterior tomada de decisão.

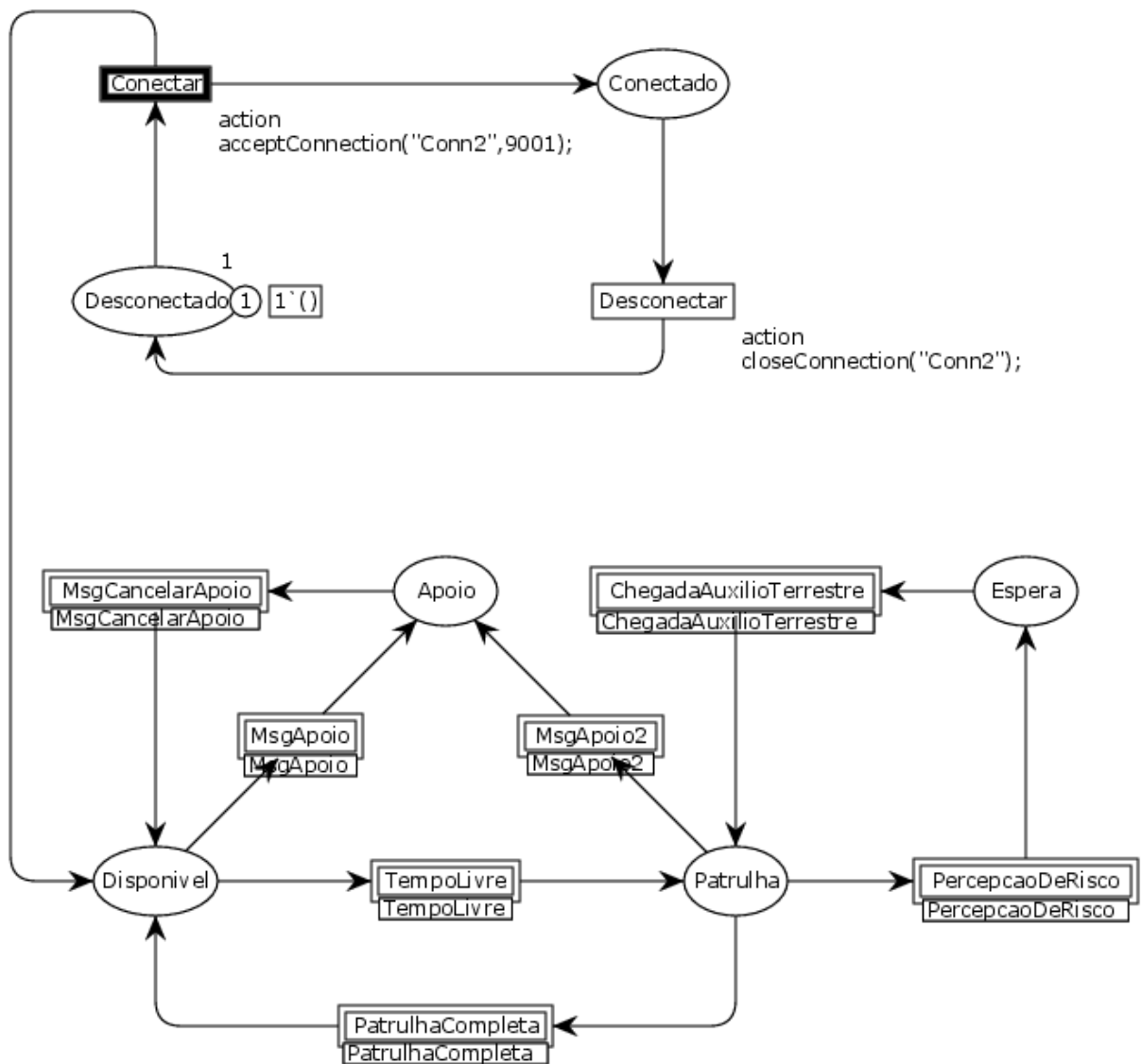


Figura 4.1 - Rede de Petri de alto nível com o modelo comportamental do *AerialBot*.

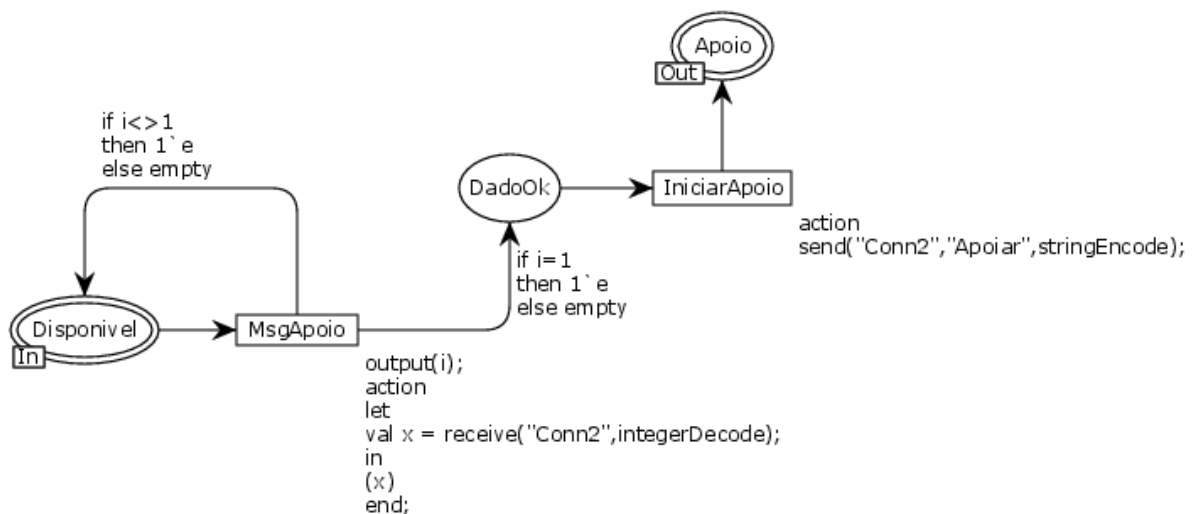


Figura 4.2 - Detalhe da sub-rede associada à transição de substituição "*MsgApoio*".

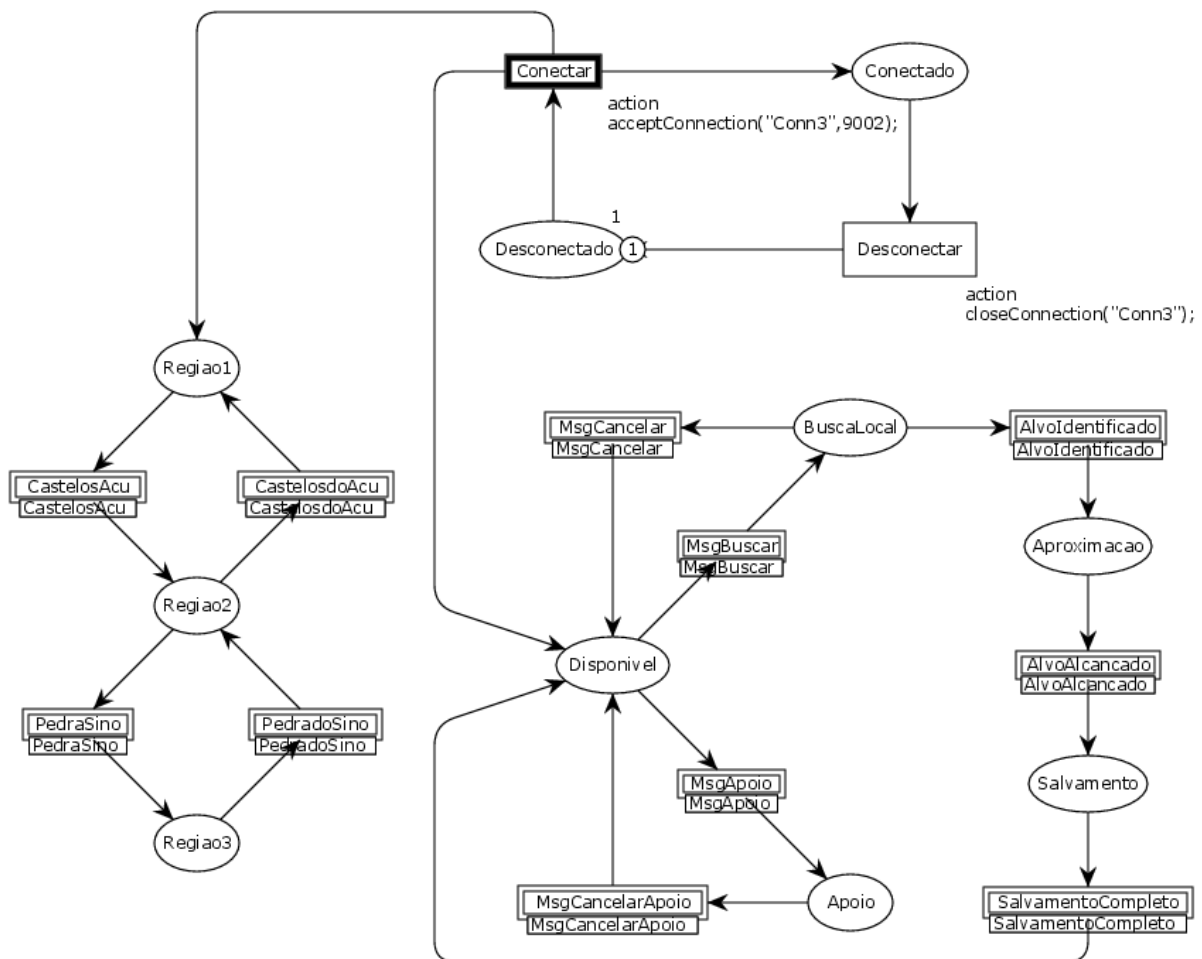


Figura 4.3 - Rede de Petri de alto nível com o modelo comportamental do *GroundBot*.

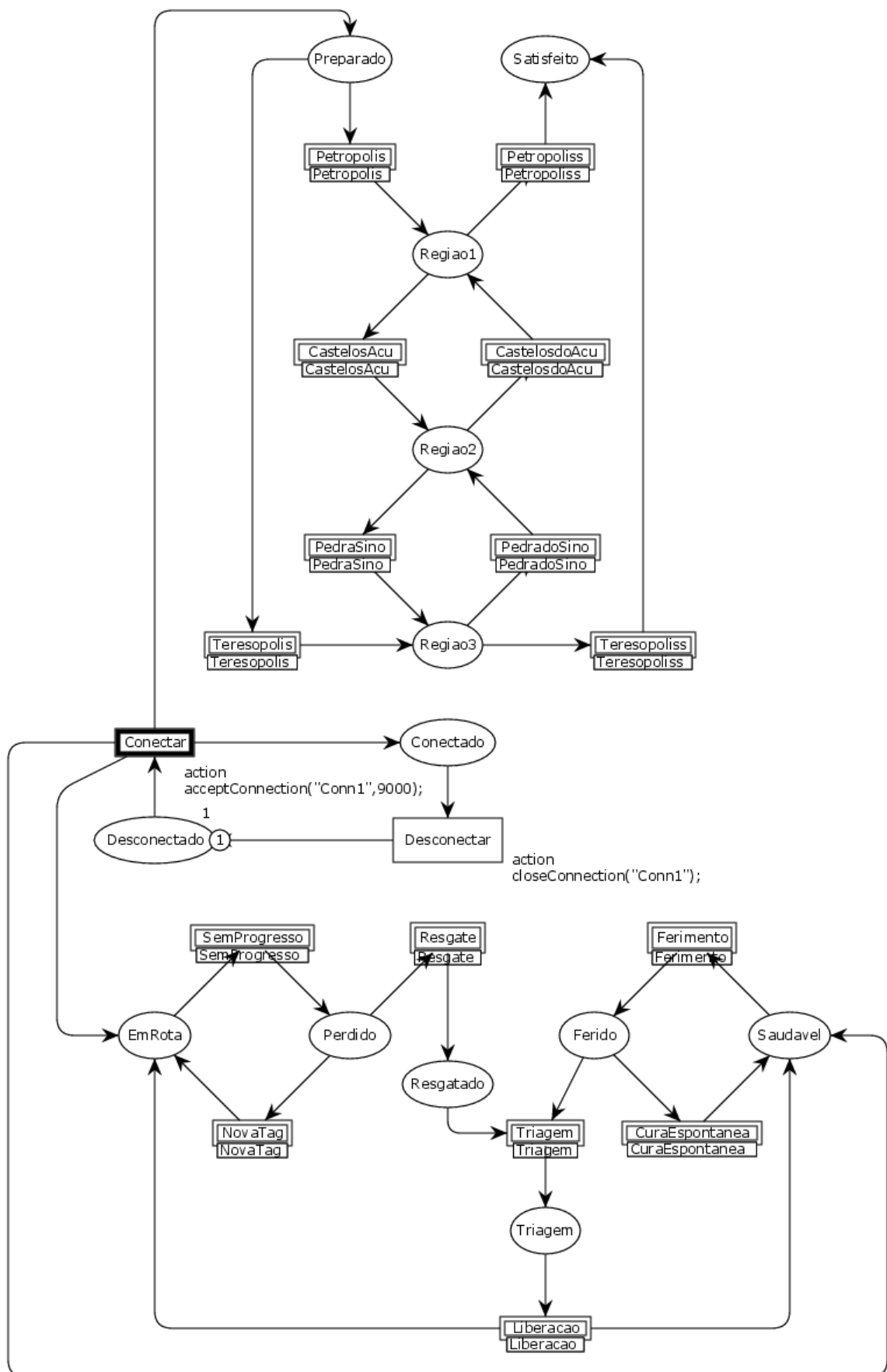


Figura 4.4 - Rede de Petri de alto nível com o modelo comportamental do agente *Walker*.

## 4.2. Implementação em V-REP

A construção dos cenários de simulação em V-REP iniciou-se com a criação dos modelos dos robôs terrestres e do robô cartesiano. Os robôs terrestres podem ser utilizados para implementar as funções tanto do agente *Walker*, quanto do agente *GroundBot*. E o agente *AerialBot* é implementado na estrutura do robô cartesiano. Na sequência, inseriu-se um ambiente de interação semelhante ao construído para testes experimentais. Este ambiente é composto por 49 células igualmente distribuídas em uma área de 1,05 x 1,05 m. As paredes internas podem ser facilmente removidas ou inseridas, possibilitando a construção de diversos cenários. A Fig. 4.5 ilustra um cenário composto pelo ambiente de testes, por um robô cartesiano e por um robô terrestre.

Todos os elementos foram inseridos de arquivos CAD e, devido ao elevado número de peças para montagem, a execução das simulações no V-REP torna-se lenta, fazendo-se necessária a aplicação da ferramenta *Extract the convex hull*.

Utilizaram-se cuboides primitivos para representar a estrutura dos leitores e etiquetas RFID. Os leitores foram posicionados na parte inferior dos robôs terrestres, enquanto as *tags* foram distribuídas no ambiente. Os elementos RFID utilizaram comunicação *Wireless* através das funções *simSendData* e *simReceiveData*, que foram implementadas nos scripts associados a cada elemento.

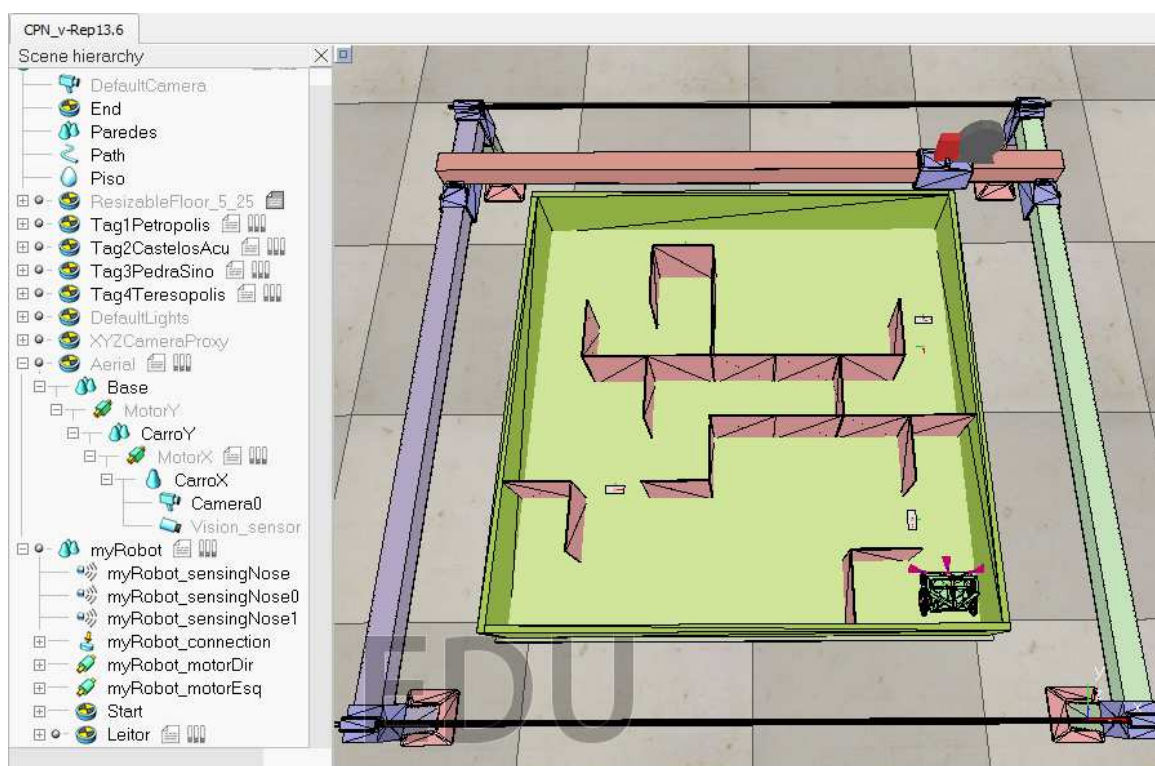


Figura 4.5 - Ambiente de simulação composto por um robô móvel e um robô cartesiano.

A Navegação dos robôs terrestres dentro do ambiente pode assumir dois comportamentos: um algoritmo seguidor de paredes ou, para casos com locais de origem e alvo definidos, a execução de uma trajetória fornecida pela ferramenta *PathPlanning*. A ferramenta *PathPlanning* deve ser configurada para planejamento de trajetória de objetos não-holonômicos (*Non-holonomic*) (objetos com restrições de movimentação em direções aleatórias, assim como robôs diferenciais), com ponto de início (*Start*), ponto alvo (*End*) e um objeto de caminho (*Path*) associados. A Fig. 4.6 apresenta as janelas de configuração de uma coleção (*collection*) de objetos e da ferramenta para planejamento de trajetória de um robô com desvio dos obstáculos associados ao elemento *Collection*.

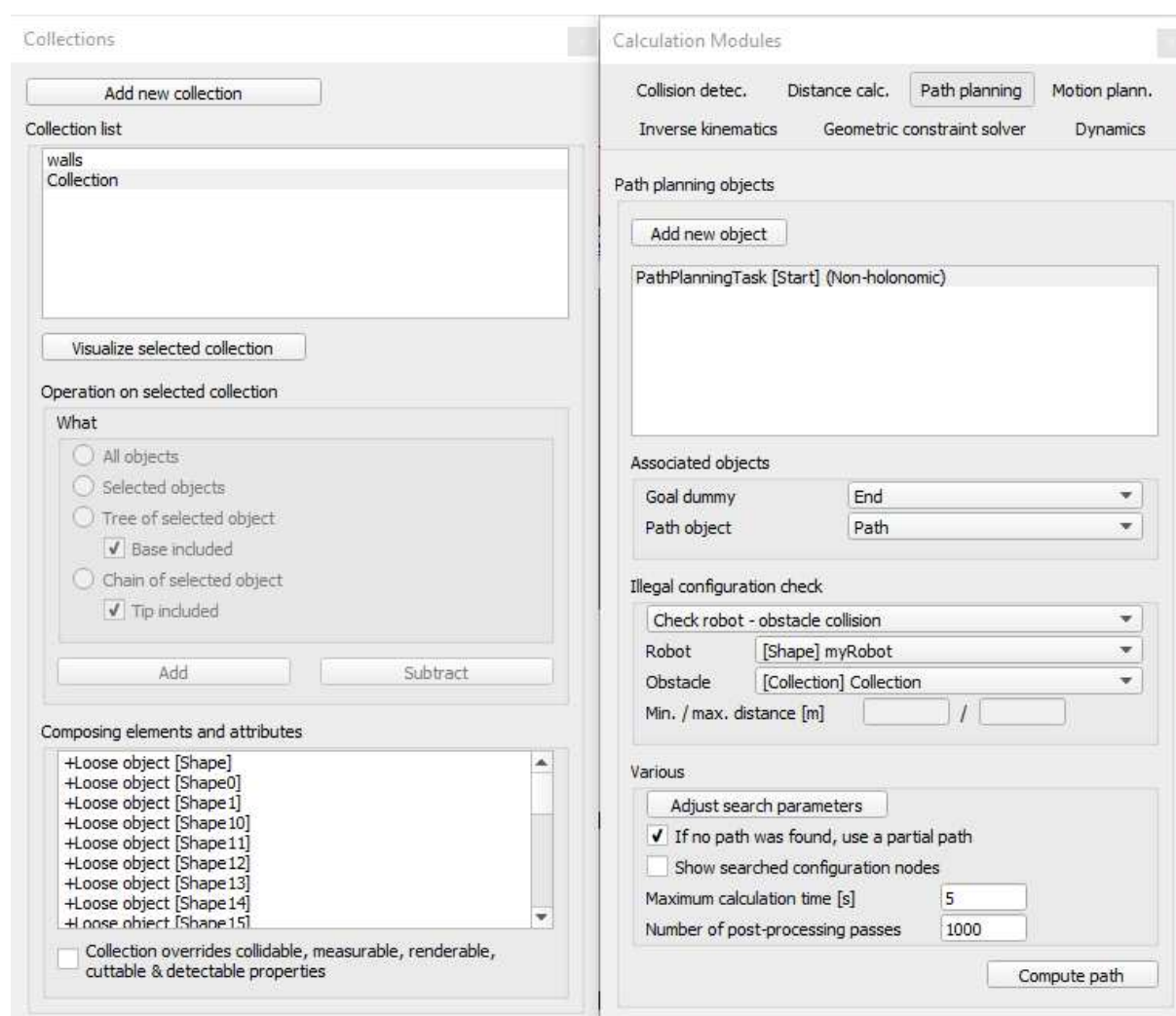


Figura 4.6 - Imagem das janelas de configuração de uma coleção (*Collection*) de objetos e da ferramenta para planejamento de trajetória de um robô com desvio dos obstáculos associados ao elemento *Collection*.

A identificação de vítimas depende de um sensor de visão associado ao robô cartesiano. Quando em patrulha aérea, as imagens coletadas pelo sensor são processadas através de uma composição específica de filtros integrados ao próprio V-REP. Um processamento de imagens através de API externa permitiria maior flexibilidade, porém, por questão praticidade e rapidez, optou-se pela utilização dos recursos integrados ao próprio simulador. A Fig. 4.7 ilustra a composição de filtros a serem aplicados à imagem adquirida. A imagem original é enviada a um buffer de processamento (*Original image to work image*); removem-se da imagem as partes de coloração verde (RGB {1.0, 0.7, 0.7}) e mantêm-se apenas as partes com coloração rósea (RGB {1.0, 0.5, 0.5}) (*Selective color on work image*); aplica-se uma detecção de formas com dimensões aproximadas à de um robô terrestre (*Blob detection on work image*); e disponibiliza-se a imagem processada (*Original image to work image* e *Work image to output image*). Detalhes adicionais da utilização dos sensores de visão podem ser consultados na bibliografia específica (COPPELIA ROBOTICS, 2018).

Um cenário do ambiente de simulação composto por três robôs móveis, um robô cartesiano, quatro elementos que simulam etiquetas RFID e uma trajetória a ser executada por um dos robôs terrestres, bem como a divisão deste ambiente nas áreas de interesse descritas na seção 3.2, é mostrada na Fig. 4.8. Neste caso onde cada agente robótico comporta-se segundo um modelo associado em redes de Petri de alto nível.

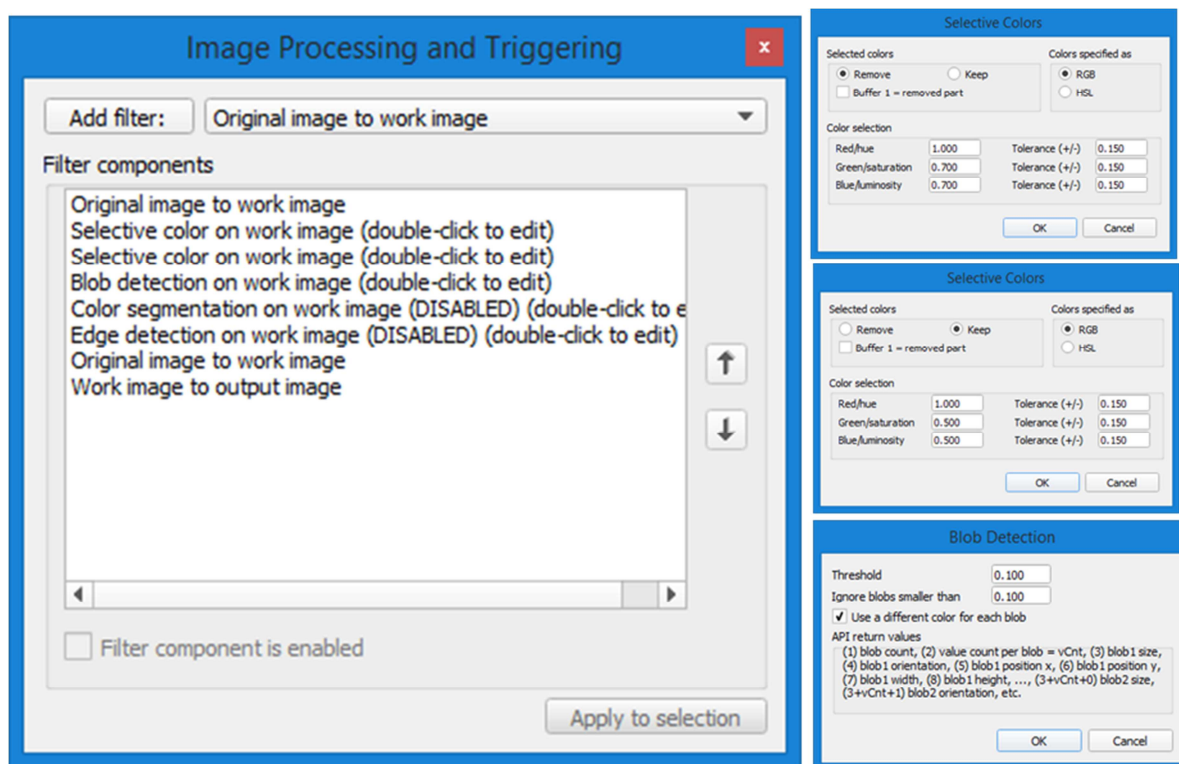


Figura 4.7 - Imagem da janela de configuração de um grupo de filtros aplicados ao sensor de visão para detecção de vítimas.



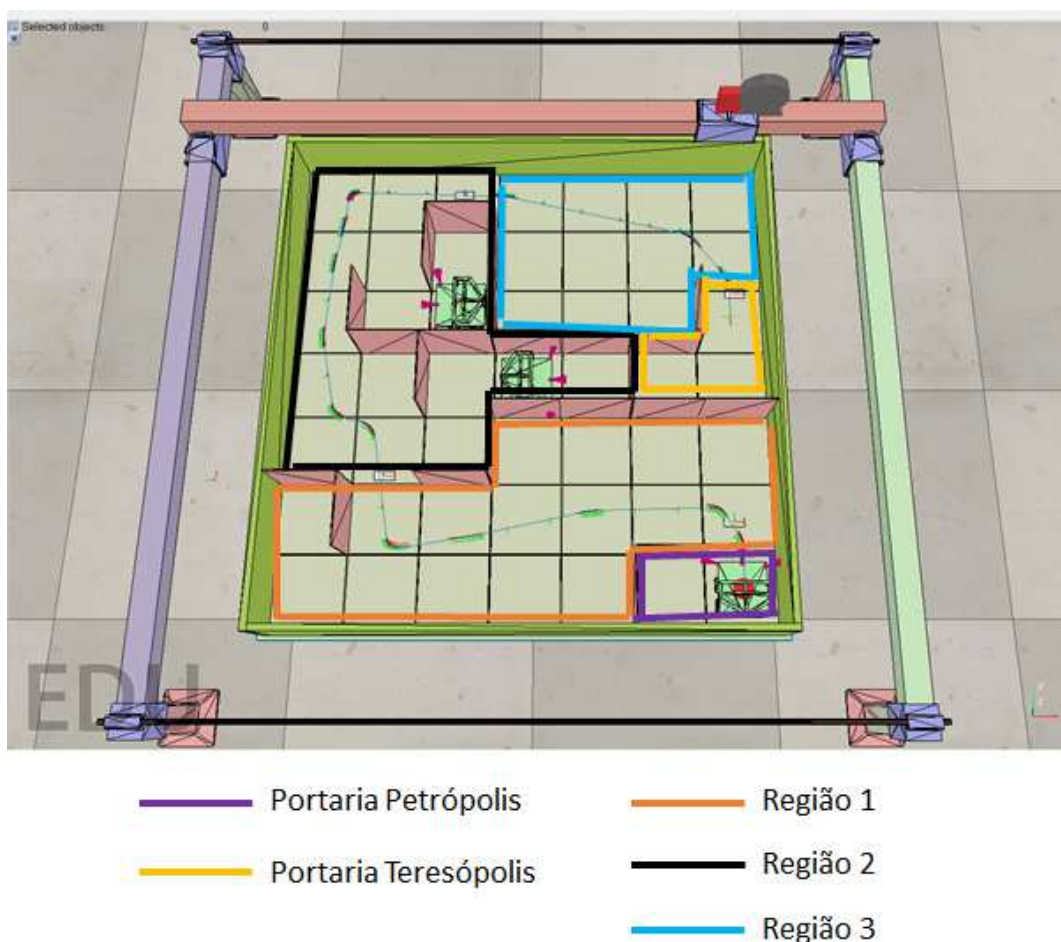


Figura 4.8 - Imagem de um cenário do ambiente de simulação composto por três robôs móveis, um robô cartesiano, quatro elementos que simulam etiquetas RFID e um trajeto a ser executado por um dos robôs terrestres.

### 4.3. Implementação experimental

Para os testes funcionais dos agentes robóticos, construíram-se dispositivos que respeitassem ao máximo os elementos apresentados na Fig. 3.8. Os robôs terrestres (*Walker* e *GroundBot*) foram concebidos em uma plataforma baseada no esquema da Fig. 4.9. As partes de sensoriamento ultrassônico e iluminação de emergência não foram inseridas nesta representação.

A parte estrutural foi desenvolvida através de peças produzidas através do processo de manufatura aditiva com impressoras 3D. A Fig. 4.10 apresenta fotografias do robô terrestre antes e após a montagem dos componentes eletrônicos especificados. Por razões operacionais, o comportamento do robô cartesiano não foi implementado experimentalmente.

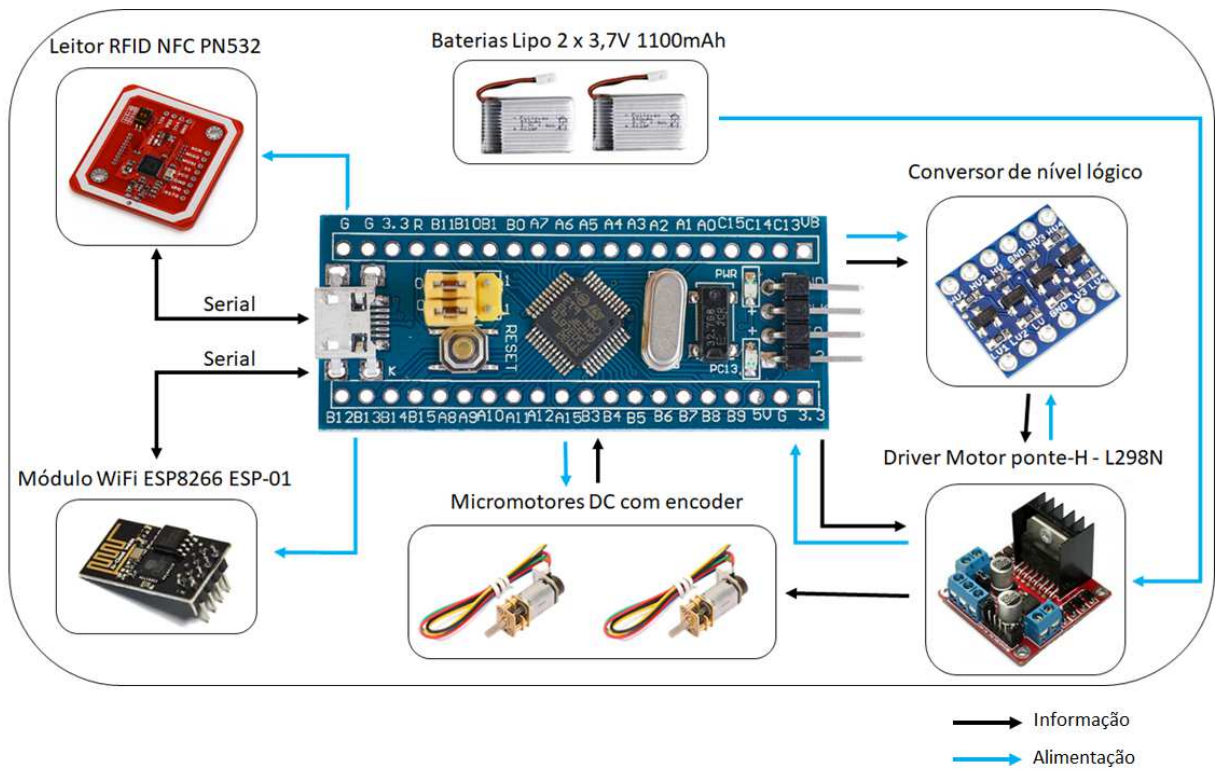


Figura 4.9 - Representação esquemática dos componentes eletrônicos aplicados aos robôs móveis.

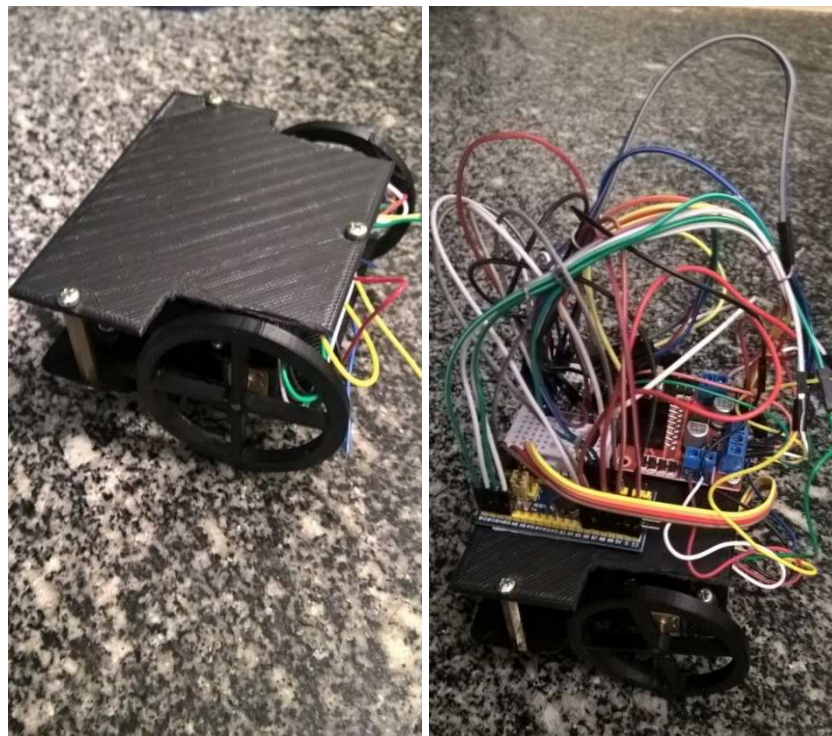


Figura 4.10 - Fotografia detalhando um robô móvel antes e após a montagem dos componentes eletrônicos.

#### 4.4. Testes Funcionais e Discussão

Verificou-se o funcionamento da nova arquitetura de controle baseada em redes de Petri de alto nível e na utilização da iPNRD integradas a um sistema multirrobo aplicado em operações de busca e salvamento em trilhas e travessias através da montagem de cenários, tanto em ambiente virtual na plataforma V-REP, quanto em ambiente estruturado de testes em laboratório. Foram preparados três casos em ambiente virtual e um caso em laboratório. Primeiramente analisou-se um cenário de navegação do agente *Walker* e atualização de informações gravadas nas etiquetas RFID. Na sequência, verificou-se o comportamento do sistema sujeito a uma tarefa de patrulha aérea, identificação de vítima e solicitação de apoio terrestre. O terceiro caso descreve um cenário de busca terrestre pelo agente *GroundBot*, seguido por identificação de vítima e solicitação de apoio aéreo. Implementaram-se ambos os casos em ambiente virtual e apenas o primeiro em laboratório. Detalhes dos ensaios realizados estão descritos nas próximas subseções.

##### 4.4.1. Caso 1: Navegação do agente Walker

Para o teste entre V-REP e CPN Tools, estabeleceu-se conexão através da *remoteAPI* denominada API-Bridge e, através das funções indicadas na Tab. 3.4, a lógica das redes de Petri de alto nível puderam comandar o comportamento dos agentes no ambiente de simulação criado no V-REP. A Fig. 4.11 apresenta uma sequência de imagens indicando a passagem do agente *Walker* pelas *tags* RFID associadas aos locais de interesse descritos na seção 3.2 durante a sua navegação ao longo de uma trajetória planejada no V-REP. O vídeo de um ensaio nestas configurações pode ser visualizado em: <https://1drv.ms/v/s!Au-cA30jRVEakoUqwLZ9dsGf8ONIHg>.

A evolução da rede de Petri do agente *Walker* durante a execução do teste pode ser visualizada na Fig. 4.12 e na Fig. 4.13. Para melhor legibilidade, optou-se por não revelar a passagem de *tokens* pelas sub-redes associadas às transições de substituição e por exibir apenas a evolução da rede de Petri principal.

O diagrama de sequência apresentado na Fig. 4.14 detalha simplificada e, para o cenário previsto no Caso 1, a interação entre os processos em execução: V-REP, CPN Tools e API-Bridge. Através da Fig. 4.14 percebe-se relevante dependência de comunicação entre processos, onde a informação recebida habilita ou não nova transição ou comportamento no processo destino. Apesar de não terem sido implementados critérios para controle de sincronia e redução dos atrasos de comunicação, a execução da simulação seguiu claramente ao modelo comportamental estruturado em rede de Petri de alto nível.

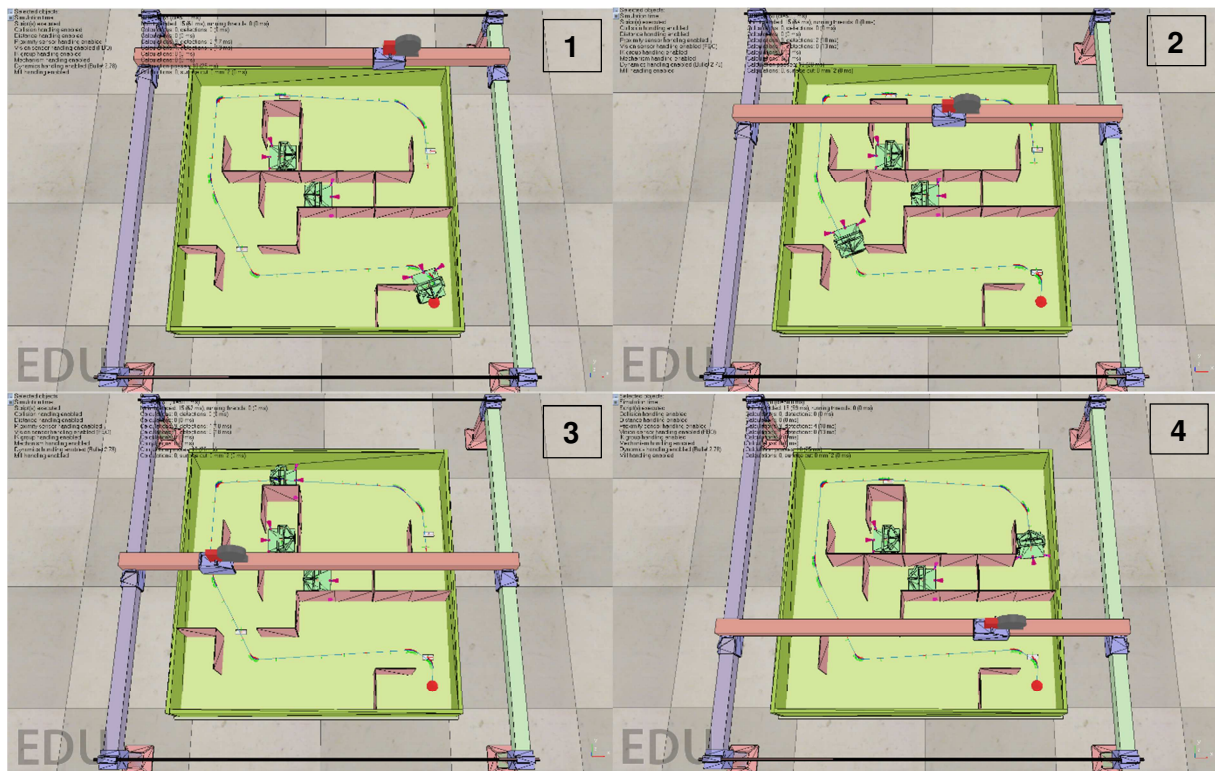


Figura 4.11 - Sequência de imagens indicando a passagem do robô móvel pelos locais especificados na seção 3.2 durante navegação ao longo de uma trajetória planejada.

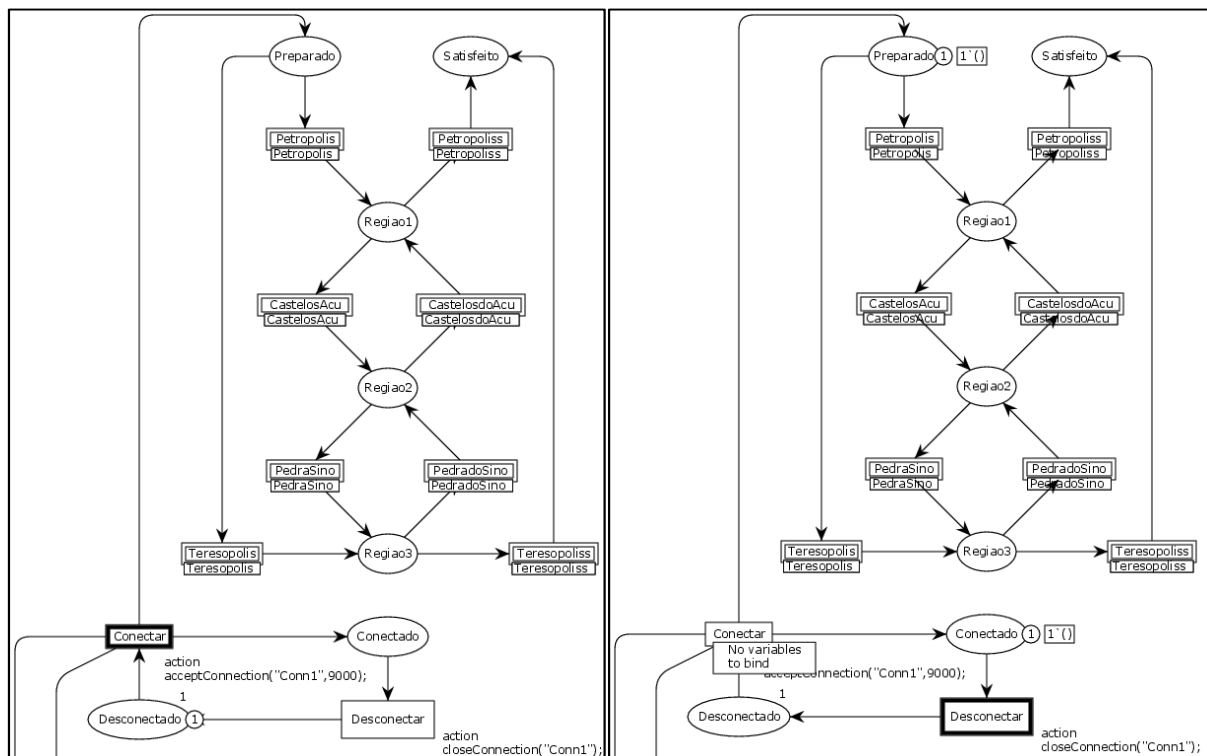


Figura 4.12 - Sequência de imagens indicando a evolução do *token* em parte da rede de Petri do agente *Walker*, saindo do lugar “Desconectado” e entrando no lugar “Preparado”.

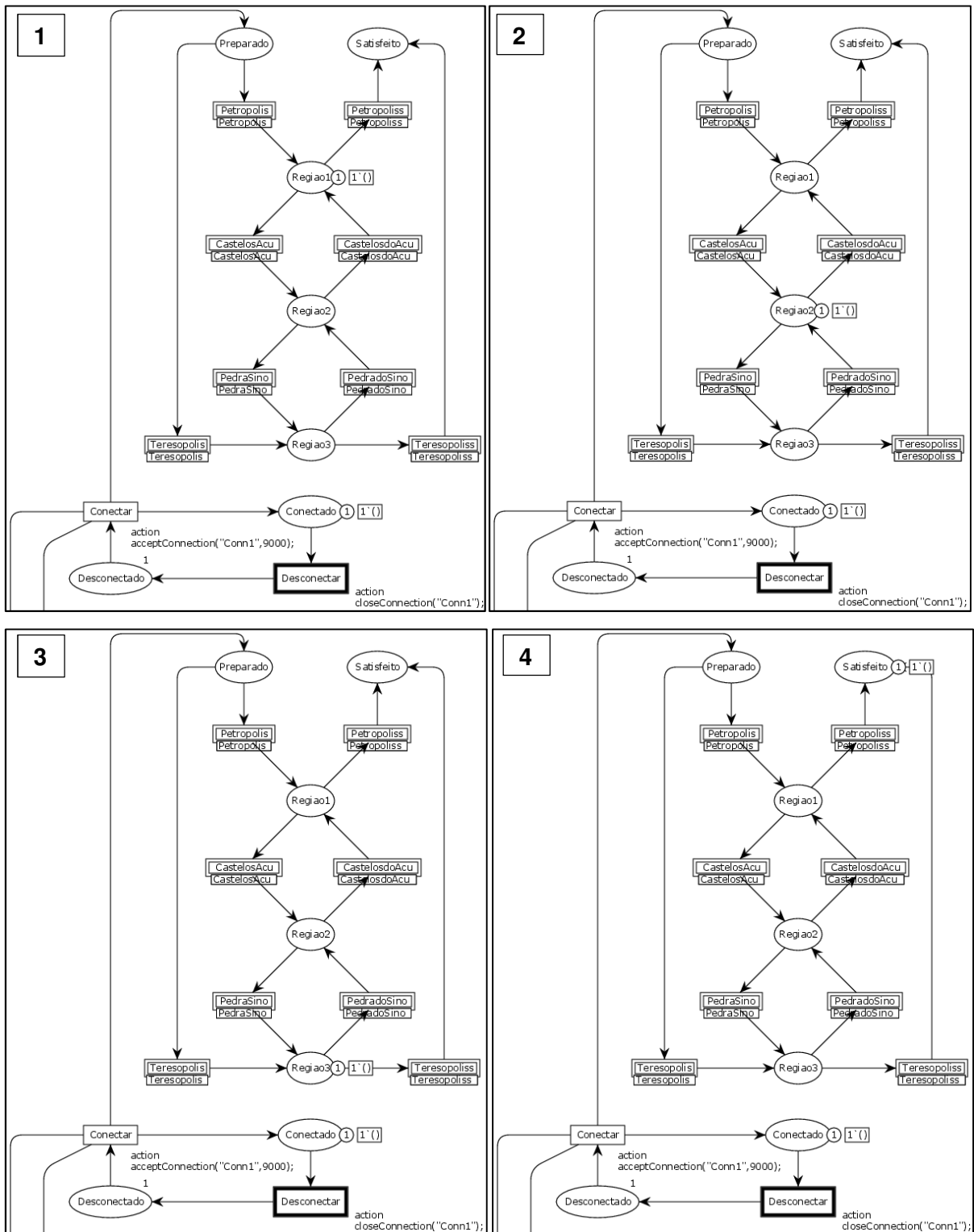


Figura 4.13 - Sequência de imagens indicando a evolução da rede de Petri do agente Walker para os passos 3 e 4, indicados na Fig. 4.11.

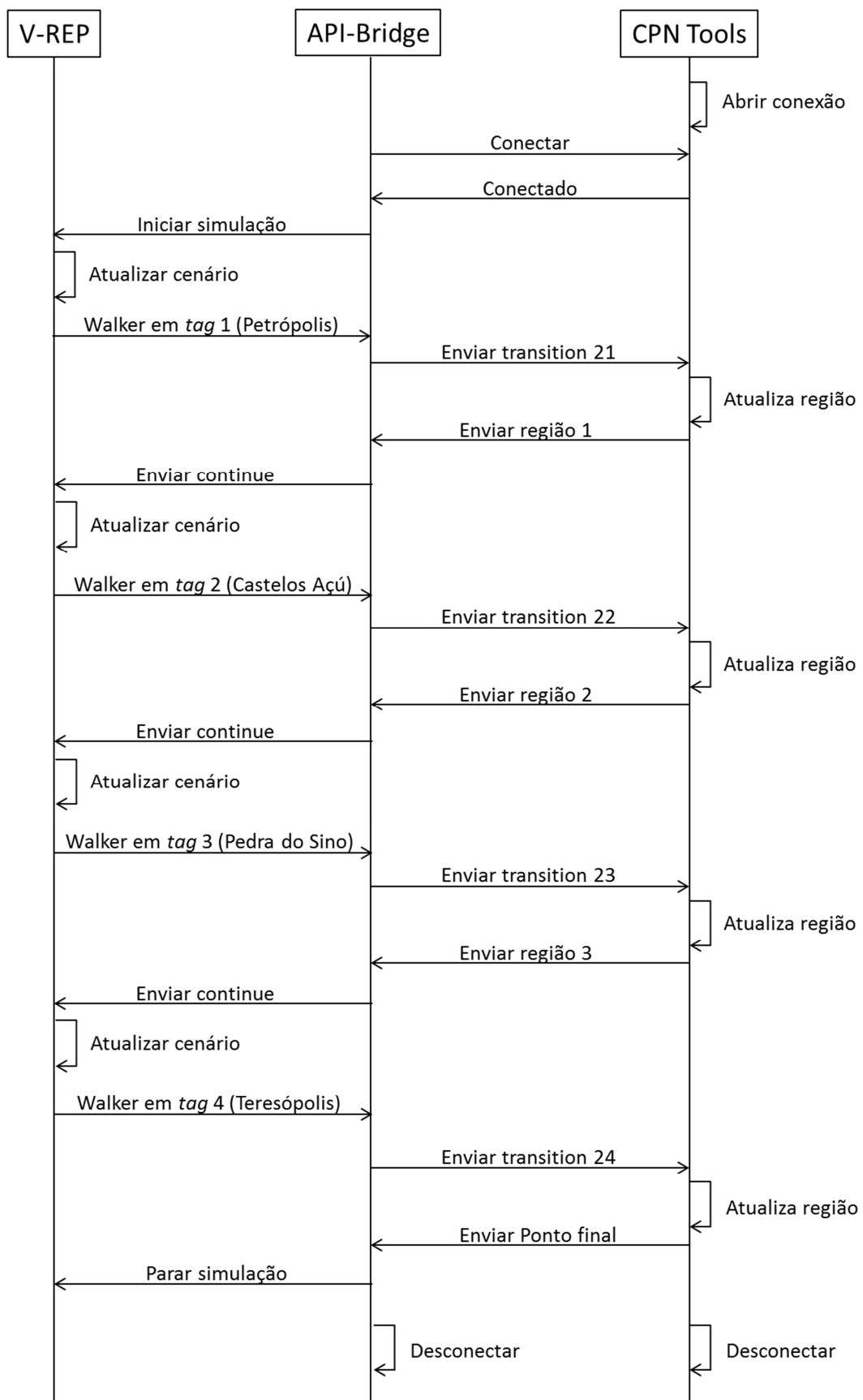


Figura 4.14 - Diagrama de sequência simplificado para execução do caso 1 em ambiente virtual de simulação.



Os testes com os robôs reais seguiram o mesmo procedimento. Inicialmente verificou-se a comunicação entre o robô e o CPN Tools. Neste caso, não houve a necessidade de uma aplicação intermediária, tal como a API-Bridge. A comunicação se deu diretamente pelo protocolo TCP/IP através da utilização do módulo wireless ESP8266-ESP01. O principal requisito, neste caso, é a existência de uma rede *Wi-Fi* disponível e acessível aos robôs e à máquina onde o CPN Tools está instalado.

Estabelecida a comunicação, associou-se o disparo das transições da rede de Petri com eventos externos, tal como a leitura de uma *tag* e verificou-se a evolução do sistema. Para os testes no ambiente estruturado, montou-se um cenário similar ao testado no V-REP (Fig. 4.8) e, impondo uma trajetória ao robô, encenou-se a sua passagem pelos pontos com etiquetas RFID instaladas, verificando se a evolução do robô ao longo da trajetória estava realmente associada à rede de Petri do agente *Walker*, em execução em outra máquina. A Fig. 4.15 apresenta uma sequência de fotografias indicando a passagem do robô móvel pelos pontos com *tags* RFID durante sua possível navegação no ambiente de testes real. A resposta verificada no serial Monitor da Arduino IDE, durante a execução dos testes funcionais, está detalhada na Fig. 4.16. O diagrama de sequência de execução do experimento é similar ao apresentado na Fig. 4.14, porém sem a existência do processo relacionado à API-Bridge, sendo por isso suprimido deste documento.

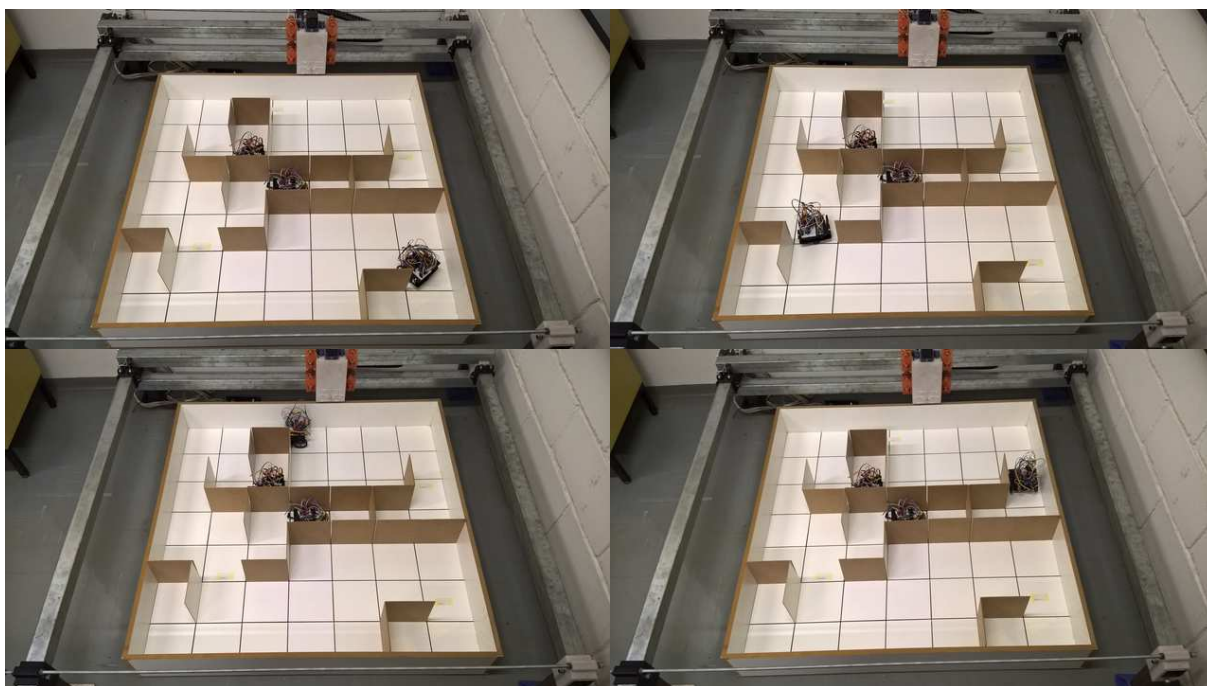


Figura 4.15 - Sequência de imagens indicando a passagem do robô móvel pelos pontos com *tags* RFID durante possível navegação em ambiente real de testes.

```

COM5 (Blue Pill)
Enviar

send=1
toSendCPN
01

Break
AAT+CIPSEND=4

OK
>Break
A
Recv 4 bytes

SEND OKrobo> Received data: Regiao1
Terminal disponivel
send=2
toSendCPN
02

Break
AAT+CIPSEND=4

OK
>Break
A
Recv 4 bytes

SEND OKrobo> Received data: Regiao2
Terminal disponivel
send=3
toSendCPN
03

Break
AAT+CIPSEND=4

OK
>Break
A
Recv 4 bytes

SEND OKrobo> Received data: Regiao3
Terminal disponivel
send=4
toSendCPN
04

Break
AAT+CIPSEND=4

OK
>Break
A
Recv 4 bytes

SEND OKrobo> Received data: Fim

☒ Auto-rolagem
Ambos, NL e CR
250000 velocidade

```

Figura 4.16 - Detalhe do serial Monitor da Arduino IDE durante a execução dos testes funcionais de passagem do agente *Walker* pelas *tags* RFID associadas aos locais especificados na seção 3.2.

#### 4.4.2. Caso 2: identificação aérea e apoio terrestre

O segundo caso verificou o comportamento do sistema sujeito a uma tarefa de patrulha aérea, identificação de vítima e solicitação de apoio terrestre. Para isso, definiram-



se vítimas como robôs terrestres de coloração rósea e fez-se uso do sensor de visão com os filtros indicados na Fig. 4.7 para a identificação. Uma vez em modo patrulha aérea, o agente *AerialBot* processa as imagens capturadas pelo sensor de visão e, ao identificar uma possível vítima, interrompe a patrulha, solicita apoio terrestre e entra em modo “Espera”. Um agente *GroundBot* em modo “Disponível”, ao receber solicitação de apoio, calcula sua rota até o ponto desejado e navega até a vítima para prestar o apoio solicitado. Neste momento, o agente *AerialBot* pode retomar o comportamento de Patrulha, enquanto o agente *GroundBot* presta o atendimento ao usuário ferido. A Fig. 4.17 apresenta uma sequência de imagens indicando o cenário inicial (onde a trajetória observada diz respeito à rodada de simulação anterior) com início da patrulha aérea (*snapshot 1*); identificação de vítima pelo robô cartesiano, solicitação de apoio terrestre, cálculo de trajetória e início de navegação (*snapshot 2*); e chegada do apoio terrestre com retomada de patrulha do veículo aéreo. O vídeo de um ensaio nestas configurações pode ser visualizado em: <https://1drv.ms/v/s!Au-cA30jRVEakoZcnA0aNOh6O4hQSQ>.

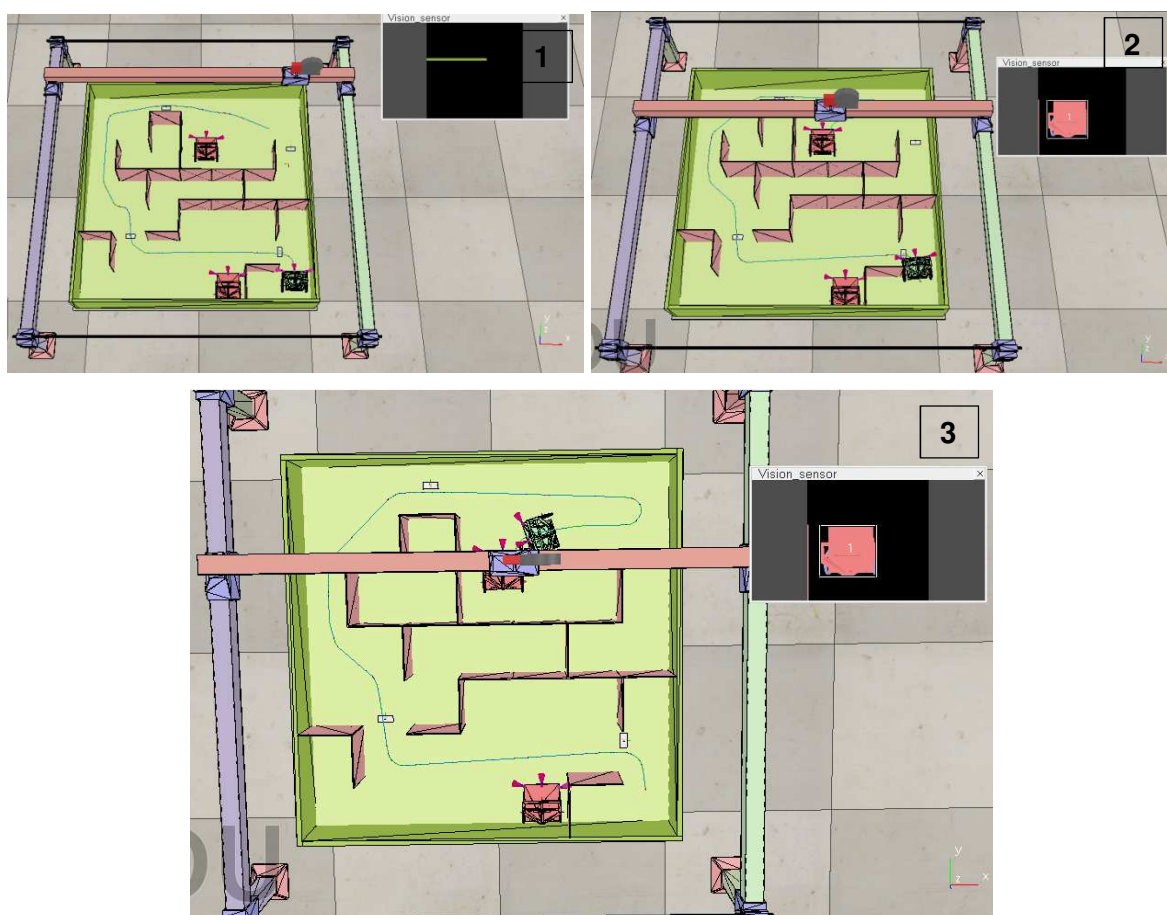


Figura 4.17 - Sequência de imagens ilustrando o início da patrulha, identificação de vítima e chegada de apoio terrestre.

A evolução das redes de Petri dos agentes *AerialBot* e *GroundBot* durante a execução do teste pode ser visualizada na Fig. 4.18 e na Fig. 4.19. Para melhor legibilidade, optou-se por não revelar a passagem de *tokens* pelas sub-redes associadas às transições de substituição e por exibir apenas a evolução da rede de Petri principal.

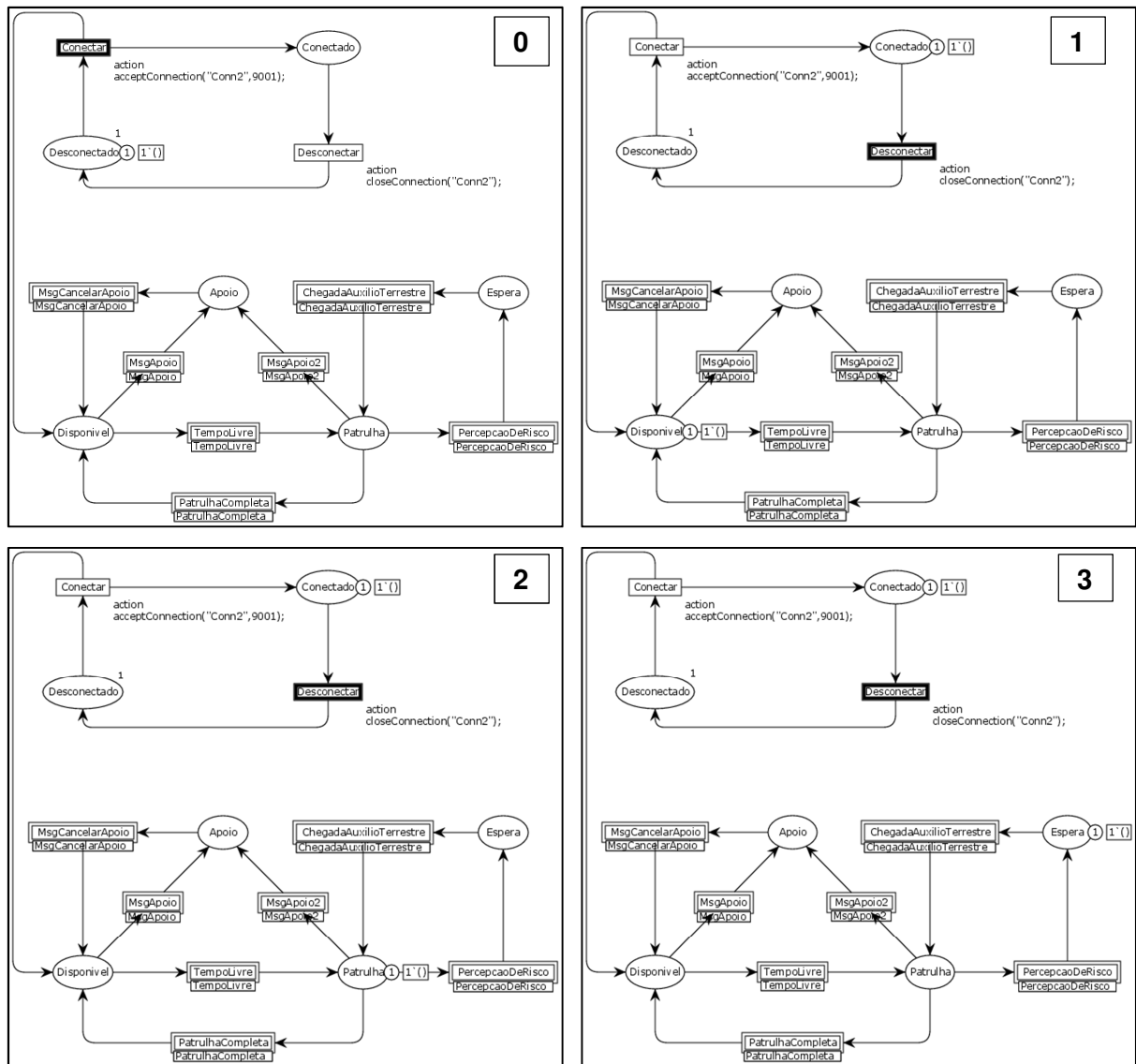


Figura 4.18 - Sequência de imagens indicando a evolução da rede de Petri do agente *AerialBot* para os passos 1, 2 e 3, indicados na Fig. 4.17.

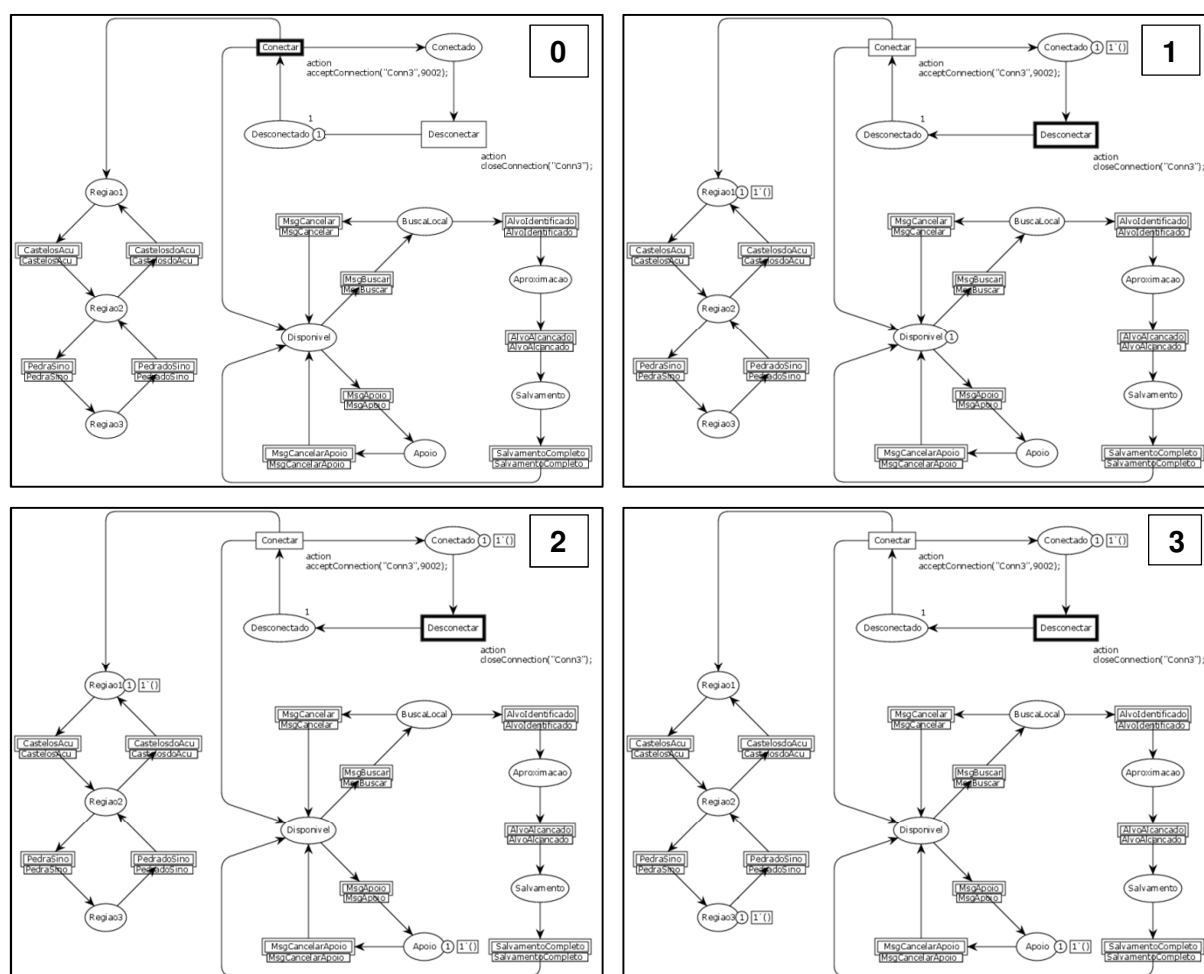


Figura 4.19 - Sequência de imagens indicando a evolução da rede de Petri do agente *GroundBot* para os passos 1, 2 e 3, indicados na Fig. 4.17.

O diagrama de sequência apresentado na Fig. 4.20 detalha simplificada, para o cenário previsto no Caso 2, a interação entre os processos em execução: V-REP, CPN Tools (*AerialBot* e *GroundBot*) e API-Bridge. Também neste caso, percebe-se relevante dependência de comunicação entre processos e, apesar de não terem sido implementados critérios para controle de sincronia e redução dos atrasos de comunicação, a execução da simulação seguiu claramente aos modelos comportamentais para os agentes *AerialBot* e *GroundBot*, estruturados em rede de Petri de alto nível.

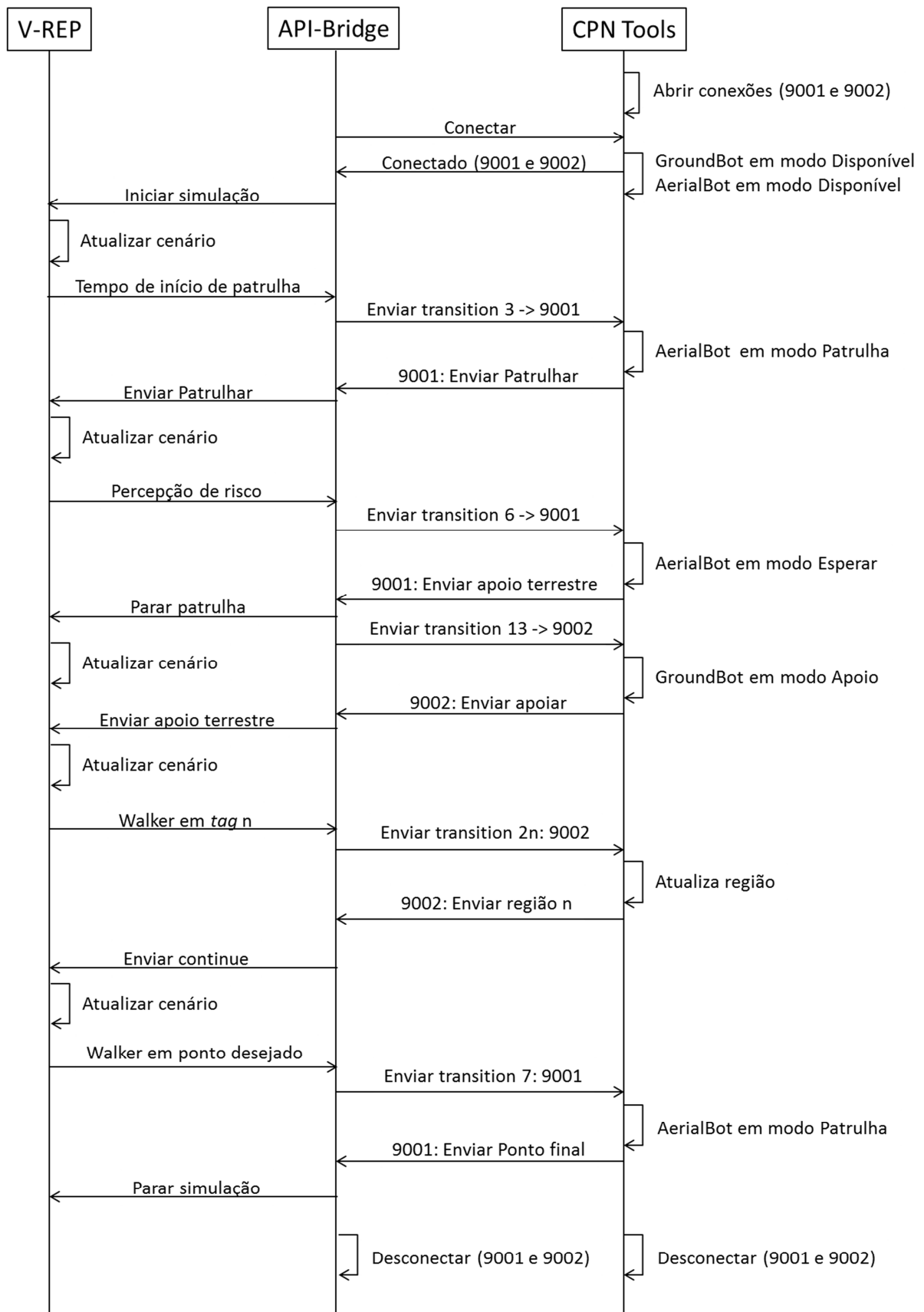


Figura 4.20 - Diagrama de sequência simplificado para execução do caso 2 em ambiente virtual de simulação.

#### 4.4.3. Caso 3: patrulha terrestre e apoio aéreo

O terceiro caso verificou o comportamento do sistema sujeito a uma tarefa de patrulha terrestre, identificação de possível vítima (dado pela análise das informações armazenadas nas *tags*) e solicitação de apoio aéreo. Neste cenário, um agente *Walker* (coloração verde quando saudável e coloração rósea quando perdido e/ou ferido) desenvolve uma trajetória equivocada e não consegue retornar à trilha. Em procedimento de patrulha terrestre, um agente *GroundBot* (coloração azul) planeja sua rota padrão (Petrópolis-Teresópolis), verificando as informações armazenadas nas *tags* em busca de possíveis usuários perdidos. Ao verificar que um usuário iniciou a trilha, mas não alcançou determinado ponto de passagem (registro na *tag* da portaria Petrópolis e falta de registro na(s) próxima(s) *tag*(s)), o agente *GroundBot* restringe sua área de busca e solicita auxílio aéreo. Ao receber a solicitação de apoio, o agente *AerialBot* interrompe sua atividade anterior e concentra a busca na região apontada pelo *GroundBot*. A Fig. 4.21 apresenta uma sequência de imagens indicando o cenário inicial (*snapshot 1*); navegação do agente *Walker* por percurso equivocado e ativação do modo perdido/ferido (coloração rósea) (*snapshot 2*); cálculo de trajetória padrão para patrulha terrestre e verificação de usuário que partiu da portaria de Petrópolis mas não alcançou o ponto de apoio Castelos do Açú (*snapshot 3*); e atendimento à solicitação de apoio aéreo com devida identificação de vítima pelo agente *AerialBot* (vídeo do ensaio disponível em: <https://1drv.ms/v/s!Au-cA30jRVEakoZdQ2wgduCLyoJivw>).

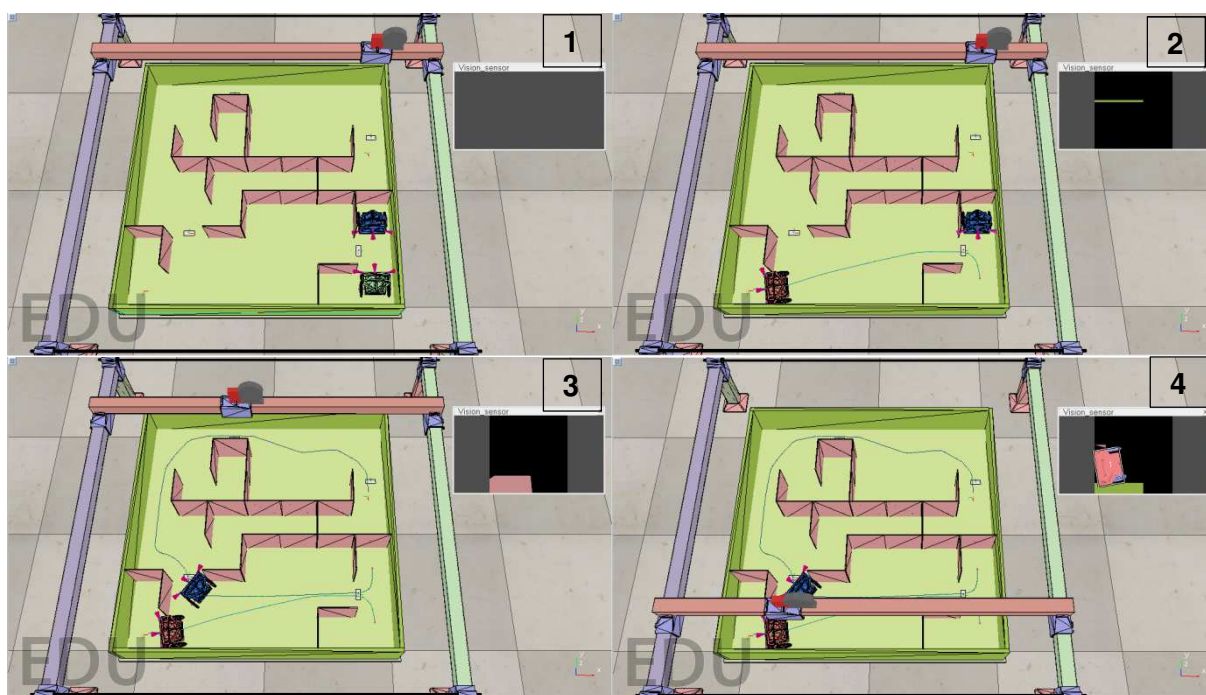


Figura 4.21 - Sequência de imagens ilustrando o início da busca, identificação de vítima e chegada de apoio aéreo.

Admitindo que os comportamentos de cada agente, representado por uma rede de Petri de alto nível em CPN Tools, já foram apresentados na seção 4.1 e exemplificados nos dois primeiros casos da seção 4.4, e, considerando a elevada quantidade de etapas características ao experimento relatado no Caso 3, para este caso em específico, optou-se por não apresentar a simples evolução dos *tokens* nas três redes de Petri dos agentes *Walker*, *GroundBot* e *AerialBot*.

Por outro lado, a interação entre os processos envolvidos foi representada através do diagrama de sequência ilustrado na Fig. 4.22, o qual detalha simplificada, para o cenário previsto no Caso 3, a interação entre os processos em execução: V-REP, CPN Tools (*Walker*, *GroundBot* e *AerialBot*) e API-Bridge. A conexão TCP/IP entre as redes de Petri associadas ao comportamento dos agentes *Walker*, *GroundBot* e *AerialBot* estabeleceu-se através das portas 9000, 9001 e 9002, respectivamente.

Um detalhe deste ensaio, apontado na mensagem “Atualizar cenário (patrulha terrestre)\*”, consiste na inserção de um comportamento de patrulha associado ao estado “Disponível” da rede de Petri do agente *GroundBot*. Tal comportamento não está devidamente previsto no modelo em rede de Petri, mas foi editado diretamente no *script* deste robô no próprio V-REP.

Novamente, percebe-se relevante dependência de comunicação entre processos e, apesar de não terem sido implementados critérios para controle de sincronia e redução dos atrasos de comunicação, a execução da simulação seguiu claramente aos modelos comportamentais para os agentes *Walker*, *GroundBot* e *AerialBot*, estruturados em rede de Petri de alto nível.

O ensaio do Caso 3 finaliza-se com a identificação da vítima pelo robô cartesiano e atualização do status do agente *Walker* para o modo triagem. Cenários mais complexos e com maiores tempos de execução requerem investigação mais detalhada e os comportamentos dos agentes ainda estão sujeitos a adaptações.

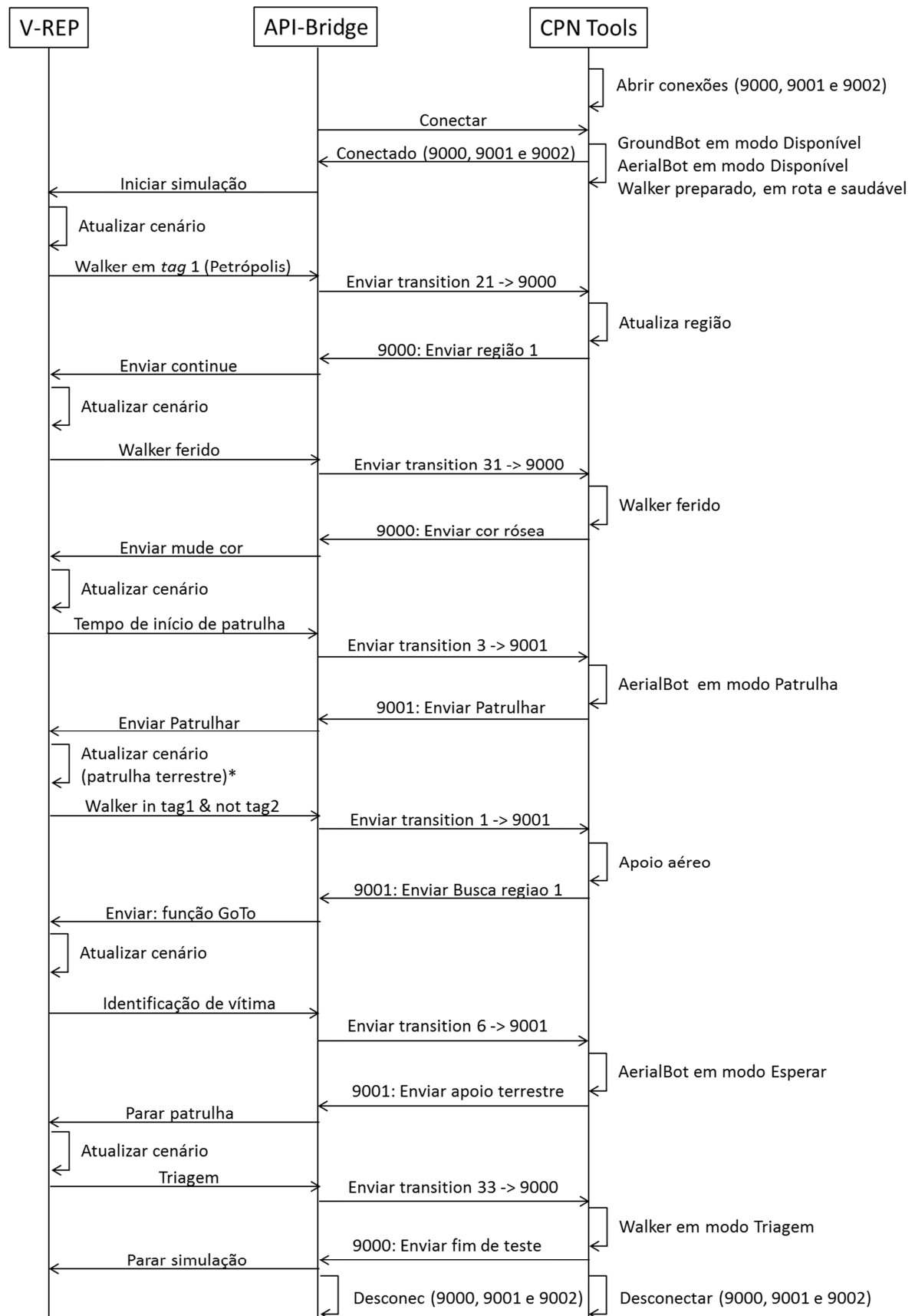


Figura 4.22 - Diagrama de sequência simplificado para execução do caso 3 em ambiente virtual de simulação.

## CAPÍTULO V

### CONCLUSÕES E PROPOSTAS DE TRABALHOS FUTUROS

Esta tese apresentou a concepção de um sistema multirrobô que integra tecnologia RFID e redes de Petri de alto nível. O objetivo principal, de estruturar uma arquitetura de controle comportamental para robôs móveis baseada em redes de Petri de alto nível integrada com a iPNRD para auxiliar operações de busca e salvamento em trilhas e travessias foi alcançado, pois definiu-se a iPNRD, implementou-se a iPNRD em dispositivos embarcados baseados na STM32 *Blue Pill*, projetou-se um sistema multirrobô para o ambiente de análise (simplificação da travessia Petrópolis-Teresópolis), modelou-se o comportamento dos agentes do sistema com redes de Petri de alto nível em CPN Tools, desenvolveu-se computacionalmente um simulador para o ambiente em V-REP, definiram-se cenários para ensaios funcionais, realizaram-se os testes de alguns cenários e discutiram-se os resultados.

As contribuições e limitações desta tese, bem como propostas de trabalhos futuros são discutidas nas próximas subseções.

#### 5.1. Contribuições da Tese

A aplicação da tecnologia RFID em sistemas multirrobô para operações de busca e salvamento é pouco abordada na literatura. Além disso, os trabalhos que consideram redes de Petri como modelos comportamentais para agentes SAR não saem das representações em alto nível para o nível das aplicações, atuação e controle dos agentes, seja em ambientes reais ou simulados. Já os trabalhos com ênfase experimental, não apresentam, ou não revelam, os modelos comportamentais de alto nível dos agentes através de linguagens estruturadas e padronizadas, como as redes de Petri de alto nível.



Além do desenvolvimento de uma arquitetura de controle, esta tese também demonstra, brevemente, um procedimento para aplicação de ações comportamentais provenientes de modelos em rede de Petri de alto nível em ações direcionadas a agentes de campo.

A solução apresentada permite que a PNRD expanda seu conceito de controle de agentes passivos e associe-se ao comportamento dinâmico de agentes ativos, conferindo maior autonomia ao sistema e consequente ineditismo da proposta.

A abordagem iPNRD, proposta nesta tese, e a sua integração com o CPN Tools, possibilita que os modelos comportamentais concebidos de forma diagramática em redes de Petri, sejam diretamente utilizados como ferramenta de controle para as ações operacionais dos agentes de campo, tornando comedida a simples abstração da representação diagramática da rede em uma linguagem de programação textual para implementação no microcontrolador do dispositivo.

Esta solução se mostra interessante do ponto de vista energético, pois as *tags* passivas associadas aos locais de interesse independem da escassez de energia, característica de lugares remotos, e os leitores RFID associados aos agentes de campo são portáteis e podem ser recarregados nos pontos de apoio.

A distribuição física das informações no ambiente e a manipulação destes dados, proporcionada pela iPNRD, permite que agentes SAR orientem suas ações de busca para uma região específica, podendo reduzir o tempo de busca e aumentar as chances de salvamento - situação exemplificada na seção 4.4.3.

Por fim, mas não menos importante, a iPNRD garante uma base de dados fisicamente distribuída no ambiente, permitindo que, independentemente da falta de rede de comunicação entre os agentes, ainda seja possível que operações de busca e salvamento aconteçam, baseadas no histórico de usuários recentes armazenados nas *tags* RFID.

## **5.2. Limitações da Solução Proposta**

A solução proposta e analisada foi implementada apenas através de cenários fictícios. As funcionalidades básicas de robôs móveis, como navegação, comunicação, localização e mapeamento, não foram desenvolvidas, o que pode aumentar tanto o custo computacional dos robôs, quanto as restrições de tempo real e exigir alteração do hardware.

O cenário de simulação é simplificado e não há garantia de que o mesmo represente fielmente o ambiente real. A cinemática e a dinâmica dos robôs, bem como o tempo de

operação simulados podem ser consideravelmente diferentes dos característicos da aplicação real.

Apesar dos testes funcionais em laboratório serem condizentes, esta solução não foi avaliada em ambientes externos. A arquitetura de controle carece de aperfeiçoamento, testes e análises em ambientes externos e a iPNRD carece de maiores investigações quanto a outros tipos de *hardwares* RFID.

Algumas funções de redes hierárquicas do CPN Tools apresentam restrições de operação, fazendo-se necessária a reinicialização da rede de Petri associada sempre que uma conexão for interrompida, o que prejudica as simulações e atrasa a realização de testes.

Nesta aplicação a comunicação entre agentes SAR deu-se via rede *Wi-Fi*. Entretanto, para aplicações em campo aberto deve-se prever a utilização de meios de comunicação de maior alcance, tal como ondas de rádio ou comunicação via satélite.

Por fim, limitações características da própria tecnologia RFID, como por exemplo, os desafios referentes a qualidade das *tags*, erros de leitura e gravação, interferência por proximidade de objetos metálicos e ou pela presença de água entre leitor e *tag*, não foram abordadas.

### 5.3. Propostas de Trabalhos Futuros

Espera-se que a aplicação da iPNRD em casos reais facilite a rastreabilidade do usuário e, em caso de incidentes, consiga apontar as prováveis regiões de busca com maior precisão, podendo reduzir o tempo de resgate e aumentar as chances de salvamento de usuários perdidos. Dentre os diversos trabalhos que surgem a partir dos desenvolvimentos apresentados nesta tese, destacam-se:

- O aperfeiçoamento dos modelos comportamentais dos agentes;
- A redução das simplificações adotadas e aperfeiçoamento dos modelos de simulação em V-REP;
- O aprimoramento de hardware e software com o objetivo de atender às funcionalidades básicas de robôs móveis, como navegação, comunicação, localização e mapeamento;
- A avaliação de cenários de maior complexidade: com maior número de bifurcações, maiores probabilidades de desvio da rota planejada, simulação de perda de conexão com a rede *Wi-Fi*, falha de funcionamento de algum agente, inserção de novos pontos de interesse e atualização automática da rede de Petri associada;

- O aperfeiçoamento dos modelos e lógica de programação para garantir o funcionamento com operações síncronas com tempos de comunicação reduzidos;
- A investigação das restrições de tempo real da arquitetura de controle proposta;
- A investigação da solução para hardwares RFID de melhor qualidade;
- Desenvolvimento da solução para outros meios e protocolos de comunicação *Wireless*, capazes de operar em campo, tal como *xBee*.
- O desenvolvimento de uma solução semelhante, utilizando *smartphones* como leitores RFID e investigação em cenários com agentes humanos;
- O desenvolvimento de um hardware específico contendo leitor RFID para auxiliar os humanos em operações SAR;
- A verificação da solução anterior em uma trilha real e comparação desta solução com os sistemas clássicos (sem rastreio RFID dos usuários).
- A expansão da solução baseada na iPNRD para outros tipos de aplicações da robótica móvel.

## REFERÊNCIAS BIBLIOGRÁFICAS

ABU-TAIEH, E. M. O.; EL-SHEIK, A. A. R. **Methodologies and Approaches in Discrete Event Simulation**. In: EL-SHEIK, A. A. R; AL-AJEELI, A; ABU-TAIEH, E. M. O. (Eds.). **Simulation and Modeling: Current Technologies and Applications**. Hershey: IGI Publishing, 2008. 451p.

ANDREWS, A.; ABDELGAWAD, M.; GARIO, A. World model for testing Urban Search and Rescue (USAR) robots using Petri Nets. In INTERNATIONAL CONFERENCE ON MODEL-DRIVEN ENGINEERING AND SOFTWARE DEVELOPMENT, 4, 2016, Roma. **Proceedings...** p. 1-8. Disponível em: <<http://ieeexplore.ieee.org/abstract/document/7954418/>>. Acesso em: 5 mar. 2018.

ASAMA, H.; MATSUMOTO, A; ISHIDA, Y. Design Of An Autonomous And Distributed Robot System: Actress. In: INTERNATIONAL WORKSHOP ON INTELLIGENT ROBOTS AND SYSTEMS, 1989, Tsukuba. **Proceedings...** p. 283-290. Disponível em: <<http://ieeexplore.ieee.org/document/637920/>>. Acesso em: 6 mar. 2018.

AHSAN, K.; SHAH, H.; KINGSTON, P. RFID Applications: An Introductory and Exploratory Study. **International Journal of Computer Science Issues**. Mahébourg, v. 7, i.1, n.3, p. 1-7, jan. 2010. Disponível em: <<http://www.ijcsi.org/papers/7-1-3-1-7.pdf>>. Acesso em: 23 dez. 2017.

AYANIAN, N. **Coordination of Multirobot Teams and Groups in Constrained Environments: Models, Abstractions, and Control Policies**, 2011. 185 f. Phd dissertation. University of Pennsylvania, Pennsylvania. Disponível em: <<https://repository.upenn.edu/edissertations/373/>>. Acesso em: 6 mar. 2018.

AZEVEDO, U. Estudante é resgatado por equipe do parnaso [Online]. In: **ICMBio/PARNASO**, jul. 2012. Disponível em: <<https://goo.gl/MZaxTB>>. Acesso em 28 abr. 2017.

BALAKIRSKY, F.; CARPIN, S.; KLEINER, A.; LEWIS, M.; VISSER, A.; WANG, J.; ZIPARO, V. A. Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from RoboCup rescue. **Journal of Field Robotics**. Los Angeles, v. 24, n. 11, p. 943-967, nov. 2007. Disponível em: <<https://doi.org/10.1002/rob.20212>>. Acesso em 5 mar. 2018.

BARNES, D. P.; GRAY, J. O. Behaviour synthesis for co-operant mobile robot control. In: INTERNATIONAL CONFERENCE ON CONTROL, 1991, Edinburgh. **Proceedings...** p. 1135–1140. Disponível em: <<http://ieeexplore.ieee.org/document/98610/>>. Acesso em: 6 mar. 2018.

BAXTER, J. L.; BURKE, E. K.; GARIBALDI, J. M.; NORMAN, M. Shared potential fields and their place in a multi-robot co-ordination taxonomy. **Robotics and Autonomous Systems**. Amsterdam, v. 57, n. 10, p. 1048-1055, out. 2009. Disponível em: <<https://doi.org/10.1016/j.robot.2009.07.023>>. Acesso em 5 mar. 2018.

BERNARDINELLO, L.; DE CINDIO, F. W. **A survey of basic net models and modular net classes**. In: ROZENBERG, G. (ed.) Advances in Petri Nets 1992. **Lecture Notes in Computer Science**, v. 609. Berlin, Heidelberg: Springer, 2012, p. 304-351.

BERTOLASO, A.; RAEISSI, M. M.; FARINELLI, A.; MURADORE, R. Cooperative UAV-UGV modeled by Petri Net Plans specification. In EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 22, 2016, Hague. **Proceedings...** p. 1720-1721. Disponível em: <<http://ebooks.iospress.nl/volumearticle/44999>>. Acesso em: 21 fev. 2018.

BICCHI, A.; DANESI, A.; DINI, G.; LA PORTA, S.; PALLOTTINO, L.; SAVINO, I. M.; SCHIAVI, R. Heterogeneous wireless multirobot system. **IEEE Robotics and Automation Magazine**. New York, v. 15, n. 1, p. 62-70, mar. 2008. Disponível em: <<http://ieeexplore.ieee.org/document/4476330/>>. Acesso em: 6 mar. 2018.

BLATT, F.; SZCZERBICKA, H. Realisation of Navigation Concepts for the Multi-agent Flood Algorithm for Search & Rescue Scenarios Using RFID Tags. In INTERNATIONAL SYMPOSIUM ON DISTRIBUTED SIMULATION AND REAL TIME APPLICATIONS, 20, 2016, London. **Proceedings...** p. 112-115. Disponível em: <<http://ieeexplore.ieee.org/document/7789879/>>. Acesso em: 20 fev. 2018.

BORGES, T. Guia local e bombeiros resgatam dois jovens na serra fina SP [Online]. In: Fé no pé, mar. 2016. Disponível em: <<https://goo.gl/tGEvDT>>. Acesso em: 28 abr. 2017.

BRIOT, J.-P.; DEMAZEAU, Y. **Principes et architecture des systèmes multi-agents**. Paris: Hermès-Lavoisier, 2001. 268p.

BURGARD, W.; MOORS, M.; FOX, D.; SIMMONS, R.; THRUN, S. Collaborative multi-robot exploration. In: INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2000, San Francisco. **Proceedings...** p. 476-481. Disponível em: <<http://ieeexplore.ieee.org/document/844100/>>. Acesso em: 6 mar. 2018.

CALISI, D.; FARINELLI, A.; IOCCHI, L.; NARDI, D. Multi-objective exploration and search for autonomous rescue robots. **Journal of field robotics**. Los Angeles, v. 24, n.9, p. 763-777, jan. 2007. Disponível em: <<http://onlinelibrary.wiley.com/doi/10.1002/rob.20216/abstract>>. Acesso em: 21 fev. 2018.

CANDEA, C.; HU, H.; IOCCHI, L.; NARDI, D.; PIAGGIO, M. Coordination in Multi-agent RoboCup Teams. **Robotics and Autonomous Systems**. Amsterdam, v. 36, n.2, p. 67-86, ago. 2001. Disponível em: <[https://doi.org/10.1016/S0921-8890\(01\)00137-3](https://doi.org/10.1016/S0921-8890(01)00137-3)>. Acesso em: 6 mar. 2018.

CASTELPIETRA, C.; IOCCHI, L.; NARDI, D.; PIAGGIO, M.; SCALZO, A.; SGORBISSA, A. Coordination among heterogeneous robotic soccer players. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2000, Takamatsu. **Proceedings...** p. 1385-1390. Disponível em: <<http://ieeexplore.ieee.org/document/893214/>>. Acesso em: 6 mar. 2018.

CENTRO DE ESTUDOS E PESQUISAS EM ENGENHARIA E DEFESA CIVIL DA UNIVERSIDADE FEDERAL DE SANTA CATARINA. **Relatório de danos materiais e prejuízos decorrentes de desastres naturais no Brasil: 1995-2014**. Florianópolis, 2016, 232 p. Relatório. Disponível em: <<http://www.ceped.ufsc.br/relatorio-de-danos-materiais-e-prejuizos-decorrentes-de-desastres-naturais-no-brasil-1995-2014/>>. Acesso em: 28 dez. 2017.

CHEN, K. Y. Cell controller design for RFID based flexible manufacturing systems. **International Journal of Computer Integrated Manufacturing**. Abingdon, v. 25, n.1, p. 35-

50, jan. 2012. Disponível em: <<http://www.tandfonline.com/doi/abs/10.1080/0951192X.2010.523845>>. Acesso em: 20 fev. 2018.

COPPELIA ROBOTICS. **Virtual Robot Experimentation Platform User Manual**. In: Coppel Robotics, GMBH, ?p. mar. 2018. Manual do usuário. Disponível em: <<http://www.coppeliarobotics.com/helpFiles/index.html>>. Acesso em: 6 mar. 2018.

COSKUN, V.; OK, K.; OZDENIZCI, B. **Near Field Communication (NFC): From Theory to Practice**. Chichester: John Wiley & Sons, 2011. 632 p.

CPN TOOLS. **Connection management functions**. In: cpntools.org, jan. 2018. Manual do usuário. Disponível em: <<http://cpntools.org/2018/01/09/connection-management-functions/>>. Acesso em: 6 mar. 2018.

CURIAC, D. I. Towards wireless sensor, actuator and robot networks: Conceptual framework, challenges and perspectives. **Journal of Network and Computer Applications**. Amsterdam, v. 63, p. 14-23, mar. 2016. Disponível em: <<https://doi.org/10.1016/j.jnca.2016.01.013>>. Acesso em: 20 mar. 2018.

DANIELSEN, S. G. **Perception-Driven Obstacle-Aided Locomotion for snake robots, linking virtual to real prototypes**. 2017. 81 f. Master thesis, Norwegian University of Science and Technology, Trondheim.

DAHL, T. S.; MATARIĆ, M. J.; SUKHATME, G. S. Multi-robot task allocation through vacancy chain scheduling. **Robotics and Autonomous Systems**. Amsterdam, v. 57, n. 6 p. 674-687, jun. 2009. Disponível em: <<https://doi.org/10.1016/j.robot.2008.12.001>>. Acesso em: 6 mar. 2018.

DEL FOYO, P. M. G.; SILVA, J. R. Towards a unified view of Petri nets and object oriented modeling. **ABCM Symposium Series in Mechatronics**. Rio de Janeiro, v. 1, p. 518-524, set. 2004.

DE LA CRUZ, C.; CELESTE, W., C.; BASTOS, T. F. A robust navigation system for robotic wheelchairs. **Control Engineering Practice**, v. 19, n. 6, p. 575-590, jun. 2011.

DE LAS MORENAS, J.; GARCÍA-HIGUERA, A.; GARCÍA-ANSOLA, P. Shop Floor Control: A Physical Agents Approach for PLC-Controlled Systems. **IEEE Transactions on Industrial Informatics**. New York, v. 13, n.5, p. 2417-2427, out. 2017. Disponível em: <<http://ieeexplore.ieee.org/document/7961283/>>. Acesso em: 20 fev. 2018.

DESEL, J.; REISIG, W. Place/transition Petri Nets. In: Reisig W., Rozenberg G. (eds) **Lectures on Petri Nets I: Basic Models**. ACPN 1996. **Lecture Notes in Computer Science**, v. 1491. Berlin, Heidelberg: Springer, 1998, p. 122-173.

DIÁRIO CATARINENSE. Grupo de Resgate em Montanha localiza casal perdido na Serra Dona Francisca [Online]. In: **Diário Catarinense**, abr. 2015. Disponível em: <<https://goo.gl/9bTwQ2>>. Acesso em: 28 abr. 2017.

DIAS, R. R.; BALADEI, S. P. **Diferenças entre as frequências do sistema RFID passivo**. 2012. 2p. Artigo Técnico. Disponível em: <<http://brasil.rfidjournal.com/artigos/vision?9591/>>. Acesso em: 23 dez. 2017.

ELECHOUSE. **PN532 NFC RFID Module User Guide**. Elechouse product info, nov. 2013. Guia do usuário. Disponível em: <[goo.gl/dcEh9g](http://goo.gl/dcEh9g)>. Acesso em: 9 mar. 2018.

FABREGAS, E.; FARIAS, G.; DORMIDO-CANTO, S.; DORMINDO, S. RFCSIM Simulador Interactivo de Robótica Móvil para Control de Formación con Evitación de Obstáculos. In: CONGRESO LATINOAMERICANO DE CONTROL AUTOMÁTICO, 16, 2014, Cancún. **Memorias...** p. 1392–1397. Disponível em: <<http://ieeexplore.ieee.org/document/893214/>>. Acesso em: 6 mar. 2018.

FABREGAS, E.; FARIAS, G.; DORMIDO-CANTO, S.; GUINALDO, M.; SÁNCHEZ, J.; BENCOMO, S. D. Platform for Teaching Mobile Robotics. **Journal of Intelligent & Robotic Systems**. Cham, v. 81, n. 1, p. 131-143, jan. 2016.

FARIAS, G.; FABREGAS, E.; PERALTA, E.; TORRES, E.; DORMIDO, S. A Khepera IV library for robotic control education using V-REP. **IFAC-PapersOnLine**. Amsterdam, v. 50, n. 1, p. 9150-9155, out. 2017. Disponível em: <<https://doi.org/10.1016/j.ifacol.2017.08.1721>>. Acesso em: 6 mar. 2018.



FAIGL, J.; KULICH, M.; PREUCIL, L. Goal assignment using distance cost in multi-robot exploration. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2012, Vilamoura. **Proceedings...** 3741-3746. Disponível em: <<http://ieeexplore.ieee.org/document/6385660/>>. Acesso em: 6 mar. 2018.

FINKENZELLER, K. **RFID Handbook: Fundamentals and Applications in Contactless Smart Cards, Radio Frequency Identification and Near-Field Communication**. 3.ed. Chichester: John Wiley & Sons, 2010. 478 p.

FOLHA DE S. PAULO. Bombeiros buscaram quatro pessoas perdidas em trilha na serra do mar [Online]. In: **Folha de S.Paulo**, jul. 2016. Disponível em: <<https://goo.gl/zUmiA2>>. Acesso em: 28 abr. 2017.

FONSECA, J. P. S.; SILVA, C. E. A.; SOUSA, A. R.; TAVARES, J. J. P. Z. S. Strategies for search and rescue multirobot system based on place/transition Petri nets and RFID distributed database. In: 2017 LATIN AMERICAN ROBOTICS SYMPOSIUM (LARS) AND 2017 BRAZILIAN SYMPOSIUM ON ROBOTICS (SBR), 2017, Curitiba. **Proceedings...** p. 1-6. Disponível em: <<https://doi.org/10.1109/SBR-LARS-R.2017.8215338>>. Acesso em: 15 ago. 2017.

FONSECA, J. P. S.; TAVARES, J. J. P. Z. S. Petri Net with RFID Distributed Database for Autonomous Search and Rescue in Tracks and Crossings. In: INTERNATIONAL WORKSHOP ON PETRI NETS AND SOFTWARE ENGINEERING, 11, 2017, Zaragoza. **Proceedings...** p. 229-230. Disponível em: <<http://ceur-ws.org/Vol-1846/paper16.pdf>>. Acesso em: 15 ago. 2017.

FOX, D.; BURGARD, W.; KRUPPA, H.; THRUN, S. A Probabilistic Approach to Collaborative Multi-Robot Localization. **Autonomous Robots**. Cham, v. 8, n. 3, p. 325-344, jun. 2000. Disponível em: <<https://doi.org/10.1023/A:1008937911390>>. Acesso em: 6 mar. 2018.

GALLASCH, G.; KRISTENSEN, L. M. COMMS/CPN: A communication infrastructure for external communication with Design/CPN. In: WORKSHOP AND TUTORIAL ON PRACTICAL USE OF COLOURED PETRI NETS AND THE CPN TOOLS, 3, 2001, Aarhus. **Proceedings...** p. 75-89. Department of Computer Science, University of Aarhus, Denmark, 2001. Disponível em: <<https://tidsskrift.dk/daimipb/article/download/7111/6074>>. Acesso em: 7 mar. 2018.

GANZ, A.; SCHAFER, J. M.; TANG, J.; YANG, Z.; YI, J.; CIOTTONE, G. Urban Search and Rescue Situational Awareness using DIORAMA Disaster Management System. In: HUMANITARIAN TECHNOLOGY: SCIENCE, SYSTEMS AND GLOBAL IMPACT 2015, HUMTECH2015, 2015, Boston/Cambridge. **Procedia Engineering**. Amsterdam, v. 107, p. 349-356. Disponível em: <<https://doi.org/10.1016/j.proeng.2015.06.091>>. Acesso em: 1 jan. 2018.

GAO, L.; MA, M.; SHU, Y.; WEI, Y. A Security Protocol Resistant to Intermittent Position Trace Attacks and Desynchronization Attacks in RFID System. **Wireless Personal Communications**. Cham, v. 68, n.4, p. 1943-1959, fev. 2013. Disponível em: <<https://link.springer.com/article/10.1007/s11277-012-0770-8>>. Acesso em: 20 fev. 2018.

GENG, N.; GONG, D.; ZHANG, Y. Petri net model and its optimization for the problem of robot rescue path planning. In INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTING, 13, 2017, Liverpool. **Intelligent Computing Theories and Application. ICIC 2017. Lecture Notes in Computer Science**, v. 10361. Cham: Springer, 2017. p. 551-563. Disponível em: <[https://link.springer.com/chapter/10.1007/978-3-319-63309-1\\_49](https://link.springer.com/chapter/10.1007/978-3-319-63309-1_49)>. Acesso em: 21 fev. 2018.

GERKEY, B. P.; MATARIĆ, M. J. A. A Formal Analysis and Taxonomy of Task Allocation in Multi-Robot Systems. **The International Journal of Robotics Research**. Los Angeles, v. 23, n. 9, p. 939-954, set. 2004. Disponível em: <<https://doi.org/10.1177/0278364904045564>>. Acesso em: 6 mar. 2018.

GIUSTO, D.; IERA, A.; MORABITO, G.; ATZORI, L. **The Internet of Things: 20th Tyrrhenian Workshop on Digital Communications**. New York: Springer-Verlag New York, 2010. 442 p.

GOOGLE. **Google Earth Pro. Versão 7.3.0**. 2017. Parque Nacional da Serra dos Órgãos. Disponível em: <<https://www.google.com/earth/download/gep/agree.html>>. Acesso em: 18 out. 2017.

GRIECO, L. A.; RIZZO, A.; COLUCCI, S.; SICARI, S.; PIRO, G.; DI PAOLA, D.; BOGGIA, G. IoT-aided robotics applications: Technological implications, target domains and open issues. **Computer Communications**. Amsterdam, v. 54, p. 32-47, dez. 2014. Disponível em: <<https://doi.org/10.1016/j.comcom.2014.07.013>>. Acesso em: 20 mar. 2018.

GUYOT, L.; HEINIGER, N.; MICHEL, O.; ROHRER, F. Teaching robotics with an open curriculum based on the e-puck robot, simulations and competitions. In: INTERNATIONAL CONFERENCE ON ROBOTICS IN EDUCATION, 2, 2011, Viena. **Proceedings...** p. 53-58. Disponível em: <<https://www.cyberbotics.com/publications/RiE2011.pdf>>. Acesso em: 6 mar. 2018.

HAMP, Q.; ZHANG, R.; CHEN, L.; GORGIS, O.; OSTERTAG, T.; LOSCHONSKY, M.; REINDL, L. New technologies for the search of trapped victims. **Ad Hoc Networks**. Amsterdam, v. 13, Part A, p. 69-82, fev. 2014. Disponível em: <<https://doi.org/10.1016/j.adhoc.2012.06.005>>. Acesso em: 7 mar. 2018.

HUBNER, J. F. **Um Modelo de Reorganização de Sistemas Multiagentes**. 2003. 246 f. Tese de Doutorado, Universidade de São Paulo, São Paulo.

ICMBIO. **PARQUE NACIONAL DA SERRA DOS ÓRGÃOS (PARNASO)**. 2017. Disponível em: <<http://www.icmbio.gov.br/parnaserradosorgaos/>>. Acesso em: 10 out. 2017.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 14443-1:2016 - Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 1: Physical characteristics**. Geneva, 2016. 8p. Norma técnica. Disponível em: <<https://www.iso.org/standard/70170.html>>. Acesso em: 10 mar. 2017.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 14443-1:2016 - Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 2: Radio frequency power and signal interface**. Geneva, 2016. 44p. Norma técnica. Disponível em: <<https://www.iso.org/standard/66288.html>>. Acesso em: 10 mar. 2017.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 14443-1:2016 - Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 3: Initialization and anticollision**. Geneva, 2016. 63p. Norma técnica. Disponível em: <<https://www.iso.org/standard/70171.html>>. Acesso em: 10 mar. 2017.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 14443-1:2016 - Identification cards -- Contactless integrated circuit cards -- Proximity cards -- Part 4:**

**Transmission protocol.** Geneva, 2016. 55p. Norma técnica. Disponível em: <<https://www.iso.org/standard/70172.html>>. Acesso em: 10 mar. 2017.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 15909-1:2004 - Systems and software engineering -- High-level Petri nets -- Part 1: Concepts, definitions and graphical notation.** Geneva, 2004. 38p. Norma técnica. Disponível em: <<https://www.iso.org/standard/38225.html>>. Acesso em: 3 jan. 2017.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO/IEC 15909-2:2011 - Systems and software engineering -- High-level Petri nets -- Part 2: Transfer format.** Geneva, 2011. 102p. Norma técnica. Disponível em: <<https://www.iso.org/standard/43538.html>>. Acesso em: 3 jan. 2017.

JENNINGS, J. S.; WHELAN, G.; EVANS, W. F. Cooperative search and rescue with a team of mobile robots. In INTERNATIONAL CONFERENCE ON ADVANCED ROBOTICS, 8, 1997, Monterey. **Proceedings...** p. 193-200. Disponível em: <<http://ieeexplore.ieee.org/abstract/document/620182/>>. Acesso em: 5 mar. 2018.

JENSEN, K.; KRISTENSEN, L. M. **Coloured Petri Nets: Modelling and Validation of Concurrent Systems.** Berlin, Heidelberg: Springer-Verlag, 2009. 374 p.

JIANG, Z.; MINGCHENG E.; LIU, Y.; LIU, J.; LI, Y. Study of manufacturing resource perception and process control of a radio-frequency-identification-enabled decentralized flexible manufacturing system. **Advances in Mechanical Engineering.** Los Angeles, v. 9, n.1, p. 1-12, jan. 2017. Disponível em: <<http://journals.sagepub.com/doi/pdf/10.1177/1687814016687449>>. Acesso em: 20 fev. 2018.

JIANJIE, L.; ZHAOHUI, H.; XUAN, Y.; RAN Z.; CHENGAN, X.; YUAN, L. Analysis of Process of Triage in Disaster Rescue Action Using Stochastic Petri Net. In INTERNATIONAL CONFERENCE ON INDUSTRIAL CONTROL AND ELECTRONICS ENGINEERING, 1, 2012, Xi'an. **Proceedings...** p. 111-115. Disponível em: <<http://ieeexplore.ieee.org/document/6322327/>>. Acesso em: 5 mar. 2018.

JUNIOR. Bombeiros resgatam casal no Dedo de Deus [Online]. In: **Trilha & Cia**, jul. 2007. Disponível em: <<https://goo.gl/zys16T>>. Acesso em: 28 abr. 2017.

JUNIOR. Casal é resgatado na Serra da Mantiqueira [Online]. In: **Trilha & Cia**, jun. 2007. Disponível em: <<https://goo.gl/qgMtrw>>. Acesso em: 28 abr. 2017.

JUNIOR. Grupo é resgatado pelos bombeiros após passar noite perdido em trilha [Online]. In: **Trilha & Cia**, nov. 2009. Disponível em: <<https://goo.gl/kEdSND>>. Acesso em: 28 abr. 2017.

JUNIOR. Grupo é resgatado na Serra do Mar [Online]. In: **Trilha & Cia**, mai. 2010. Disponível em: <<https://goo.gl/V26okQ>>. Acesso em: 28 abr. 2017.

KING, J. V. Overview of the Cospas-Sarsat satellite system for search and rescue. In: THE RADIO CLUB OF AMERICA, INC, 85, 2013. **Proceedings...** p. 8-13. Disponível em: <<http://radioclubofamerica.org/wp-content/uploads/2015/07/Spring-2013-RCA-low-res.pdf>>. Acesso em: 31 dez. 2017.

KOENIG, N.; HOWARD, A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2004, Sendai **Proceedings...** p. 2149-2154. Disponível em: <<http://ieeexplore.ieee.org/document/1389727/>>. Acesso em: 6 mar. 2018.

KONG, H. The modeling of mobilization work process for emergency rescue based on stochastic Petri net. **ICIC Express Letters**. Kumamoto, v. 10, n.4, p. 977-983, abr. 2016. Disponível em: <<http://www.icicel.org/ell/contents/2016/4/el-10-04-31.pdf>>. Acesso em: 21 fev. 2018.

KUBE, C. R.; BONABEAU, E. Cooperative transport by ants and robots. **Robotics and Autonomous Systems**. Amsterdam, v. 30, n.1, p. 85-101, jan. 2000. Disponível em: <[https://doi.org/10.1016/S0921-8890\(99\)00066-4](https://doi.org/10.1016/S0921-8890(99)00066-4)>. Acesso em: 6 mar. 2018.

LACK, D. A.; SHEETS, A. L.; ENTIN, J. M.; CHRISTENSON, D. C. Rock climbing rescues: causes, injuries, and trends in boulder county, Colorado. **Wilderness & Environment Medicine**. Amsterdam, v. 23, n.3, p. 223-230, set. 2012. Disponível em: <<https://doi.org/10.1016/j.wem.2012.04.002>>. Acesso em: 6 mar. 2018.

LESSER, V.; HORLING, B.; KLASSNER, F.; RAJA, A.; WAGNER, T.; ZHANG, S. BIG: An agent for resource-bounded information gathering and decision making. **Artificial Intelligence** Amsterdarm, v. 118, n. 1, p. 197–244, abr. 2000. Disponível em: <[https://doi.org/10.1016/S0004-3702\(00\)00005-9](https://doi.org/10.1016/S0004-3702(00)00005-9)>. Acesso em: 6 mar. 2018.

LOBATO, E.; KFURI, L. Professor morre na serra dos Órgãos [Online]. In: **Folha de S.Paulo**, out. 1999. Disponível em: <https://goo.gl/UBNpzf>>. Acesso em: 28 abr. 2017.

LV, Y.; LEE, C. K. M.; CHAN, H. K.; IP, W. H. RFID-based colored Petri net applied for quality monitoring in manufacturing system. **International Journal of Advanced Manufacturing Technology**. Cham, v. 60, n.1, p. 225-236, abr. 2012. Disponível em: <<https://link.springer.com/article/10.1007/s00170-011-3568-z>>. Acesso em: 20 fev. 2018.

MADHAVAN, R.; FREGENE, K.; PARKER, L. E. Distributed heterogeneous outdoor multi-robot localization. In: INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2002, Washington, **Proceedings...** 374-381. Disponível em: <<http://ieeexplore.ieee.org/document/1013389/>>. Acesso em: 6 mar. 2018.

MICHEL, O. Cyberbotics Ltd. Webots™: Professional Mobile Robot Simulation. **International Journal of Advanced Robotics Systems**. Los Angeles, v. 1, n. 1, p. 39-42, mar. 2004. Disponível em: <<https://doi.org/10.5772/5618>>. Acesso em: 6 mar. 2018.

MILNER, R.; TOFTE, M.; HARPER, R.; MACQUEEN, D. **The Definition of Standard ML** (Revised). Cambridge: MIT Press, 1997.126p.

MURATA, T. Petri Nets: Properties, analysis and applications. **Proceedings of the IEEE**. New York, v. 77, n.4, p. 541-580. abr. 1989. Disponível em: <<http://ieeexplore.ieee.org/document/24143/>>. Acesso em: 30 jul. 2012.

NERURKAR, E. D.; ZHOU, K. X.; ROUMELIOTIS, S. I. A hybrid estimation framework for Cooperative Localization under communication constraints. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2011, San Francisco. **Proceedings...** p. 502-509. Disponível em: <<http://ieeexplore.ieee.org/document/6094684/>>. Acesso em: 6 mar. 2018.

NOGUEIRA, L. L. Comparative analysis between gazebo and v-rep robotic simulators. In: SEMINARIO INTERNO DE COGNICAO ARTIFICIAL, 4, 2014, Campinas. **Proceedings...** p. 1-5. Disponível em: <<http://www.dca.fee.unicamp.br/~gudwin/courses/IA889/2014/IA889-02.pdf>>. Acesso em: 31 dez. 2017.

NXP. **PN532 User Manual**. NXP Semiconductors docs, nov. 2007. Manual do usuário. Disponível em: <<https://www.nxp.com/docs/en/user-guide/141520.pdf>>. Acesso em: 9 mar. 2018.

PARKER, L. E. ALLIANCE: an architecture for fault tolerant, cooperative control of heterogeneous mobile robots. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 1994, Munich. **Proceedings...** p. 776-783. Disponível em: <<http://ieeexplore.ieee.org/document/407550/>>. Acesso em: 5 mar. 2018.

PASCUCCI, F.; SETOLA, R. An Indoor localization Framework for Hybrid Rescue Teams. In: IFAC WORLD CONGRESS, 18, 2011, Milan. **IFAC Proceedings Volumes**. Amsterdam, v.44, n.1, p. 4765-4770. Disponível em: <<https://doi.org/10.3182/20110828-6-IT-1002.02818>>. Acesso em: 1 jan. 2018.

PEI, Y.; MUTKA, M. W. Steiner traveler: Relay deployment for remote sensing in heterogeneous multi-robot exploration. In: INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2012, Saint Paul. **Proceedings...** p. 1551-1556. Disponível em: <<http://ieeexplore.ieee.org/document/6225347/>>. Acesso em: 6 mar. 2018.

PERALTA, E.; FABREGAS, E.; FARIAS, G.; VARGAS, H.; DORMIDO, S. Development of a Khepera IV Library for the V-REP Simulator. **IFAC-PapersOnLine**. Amsterdam, v. 49, n. 6, p. 81-86, jul. 2016. Disponível em: <<https://doi.org/10.1016/j.ifacol.2016.07.157>>. Acesso em: 6 mar. 2018.

PETRI, C. A. **Kommunikation mit Automaten**. 1962. Phd dissertation. Darmstadt Technical University, Darmstadt, Alemanha.

PETRINET. **Petri nets**. 2018. Disponível em: <<http://www.petrinets.info/>>. Acesso em: 3 jan. 2018.

PINCIROLI, C.; TRIANNI, V.; O'GRADY, R.; PINI, G.; BRUTSCHY, A.; BRAMBILLA, M.; MATHEWS, N.; FERRANTE, E.; DICARO, G.; DUCATELLE, F.; STIRLING, T.; GUTIERREZ, A.; GAMBARDELLA, L.; DORIGO, M. ARGoS: A modular, multi-engine simulator for heterogeneous swarm robotic. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, 2011, San Francisco. **Proceedings...** p. 5027-5034. Disponível em: <<http://ieeexplore.ieee.org/document/6094829/>>. Acesso em: 6 mar. 2018.

PINHEIRO, M.; MEDEIROS, B. Corpo de professor que estava desaparecido é encontrado [Online]. In: **Metrópoles**, dez. 2016. Disponível em: <[goo.gl/2p722C](http://goo.gl/2p722C)>. Acesso em: 28 abr. 2017.

PROROK, A.; BAHR, A.; MARTINOLI, A. Low-cost collaborative localization for large-scale multi-robot systems. In: INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2012, Saint Paul. **Proceedings...** p. 4236-4241. Disponível em: <<http://ieeexplore.ieee.org/document/6225016/>>. Acesso em: 6 mar. 2018.

QU, Y.; WANG, M. Modeling and analyzing performance of the emergency rescue logistics system based on Petri nets. **ICIC Express Letters**. Kumamoto, v. 10, n.6, p. 1293-1301, jun. 2016. Disponível em: <<http://www.icicel.org/ell/contents/2016/6/el-10-06-09.pdf>>. Acesso em: 21 fev. 2018.

R7. Professor desaparece durante trilha e é encontrado com vida após quatro dias SP [Online]. In: **R7 Notícias**, jul. 2017. Disponível em: <<http://bit.ly/2vIHm8F>>. Acesso em: 28 nov. 2017.

REISIG, W. **Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies**. Berlin: Springer-Verlag Berlin Heidelberg, 2013. 230p.

RODRIGUES, B. H. M. **Robô Cartesiano CNC: Projeto e construção de um robô cartesiano multifuncional**. 2017. 94 f. Monografia de Conclusão de Graduação, Universidade de Federal de Uberlândia, Uberlândia.

ROHMER, E.; SINGH, S. P. N.; FREESE, M. V-REP: A versatile and scalable robot simulation framework. In: INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS



AND SYSTEMS, 2013, Tokyo. **Proceedings...** p. 1321-1326. Disponível em: <<http://ieeexplore.ieee.org/abstract/document/6696520/>>. Acesso em: 6 mar. 2018.

ROHMER, E.; YOSHIDA, T.; OHNO, K.; NAGATANI, K.; TADOKORO, S.; KONAYAGI, S. Quince: A Collaborative Mobile Robotic Platform for Rescue Robots Research and Development. In: INTERNATIONAL CONFERENCE ON ADVANCED MECHATRONICS, 2010, Toyonaka. **Proceedings...** p. 225-230. Disponível em: <<https://doi.org/10.1299/jsmeicam.2010.5.225>>. Acesso em: 1 jun. 2018.

ROSSI, M. Número de pessoas perdidas em trilhas aumenta 20% no litoral de SP [Online]. In: **G1 Santos**, jan. 2015. Disponível em: <[goo.gl/1bVeyu](http://goo.gl/1bVeyu)>. Acesso em: 28 abr. 2017.

ROUMELIOTIS, S. I.; BEKEY, G. A. Distributed multirobot localization. **IEEE Transactions on Robotics and Automation**. New York, v. 18, n. 5, p. 781-795, dez. 2002. Disponível em: <<http://ieeexplore.ieee.org/document/1067998/>>. Acesso em: 6 mar. 2018.

SIEGWART, R.; NOURBAKHSI, I. R. **Introduction to Autonomous Mobile Robots**. Cambridge: The MIT Press, 2004. 335 p.

SILVA, A. G. **Projeto e construção de robô móvel tipo micromouse para bancada de busca e salvamento**. 2017. 63 f. Monografia de Conclusão de Graduação, Universidade de Federal de Uberlândia, Uberlândia. Disponível em: <<https://repositorio.ufu.br/handle/123456789/19691>>. Acesso em: 28 jan. 2018.

SILVA, C. E. A. **Desenvolvimento de biblioteca para aplicações de PNRD E PNRD invertida embarcadas em Arduino**. 2017. 102 f. Monografia de Conclusão de Graduação, Universidade Federal de Uberlândia, Uberlândia.

SILVA, C. E. A.; TAVARES, J. J. P. Z. S.; FERREIRA, M. V. M. Arduino Library Developed for Petri Net Inserted into RFID Database and Variants. In: INTERNATIONAL CONFERENCE ON APPLICATIONS AND THEORY OF PETRI NETS AND CONCURRENCY, 39, 2018, Bratislava. **Proceedings...** Cham: Springer-Verlag, 2018. No prelo.

SMC. Automation [Online]. In: **SMC International Training**. Disponível em: <<https://www.smctraining.com/en/webpage/indexpage/312>>. Acesso em: 25 dez. 2017.

SPTV. “Homem que foi procurar grupo perdido na Serra do Mar, em SP, também se perde [Online],” In: **O Globo**, jul. 2011. Disponível em: <<https://goo.gl/zixFLF>>. Acesso em: 28 abr. 2017.

STM32DUINO. **Blue Pill - STM32duino wiki**. [wiki.stm32duino.com](http://wiki.stm32duino.com), fev. 2018. Guia do usuário. Disponível em: <[http://wiki.stm32duino.com/index.php?title=Blue\\_Pill](http://wiki.stm32duino.com/index.php?title=Blue_Pill)>. Acesso em: 7 mar. 2018.

SUN, H.; CHANG, Z; MO, R. Monitoring and controlling the complex product assembly executive process via mobile agents and RFID tags. **Assembly Automation**. Bingley, v. 29, n.3, p. 263-271. 2009. Disponível em: <<http://www.emeraldinsight.com/doi/full/10.1108/01445150910972949>>. Acesso em: 21 fev. 2018.

TANG, F.; PARKER, L. E. ASyMTRe: Automated Synthesis of Multi-Robot Task Solutions through Software Reconfiguration. In: INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2005, Barcelona. **Proceedings...** p. 1501-1508. Disponível em: <<http://ieeexplore.ieee.org/document/1570327/>>. Acesso em: 6 mar. 2018.

TAVARES, J. J. P. Z. S.; MUROFUSHI, R. H.; SILVA, L. H. S.; SILVA, G. R. Petri Net Inside RFID Database Integrated with RFID Indoor Positioning System for Mobile Robots Position Control. In: INTERNATIONAL WORKSHOP ON PETRI NETS AND SOFTWARE ENGINEERING, 11, 2017, Zaragoza. **Proceedings...** p. 157-176. Disponível em: <<http://ceur-ws.org/Vol-1846/paper10.pdf>>. Acesso em: 15 ago. 2017.

TAVARES, J. J. P. Z. S.; SARAIVA, T. A. Elementary Petri net inside RFID distributed database (PNRD). **International Journal of Production Research**. London, v. 48, n. 9, p. 2563-2582, mar. 2010.

TECHSHOP. **STM32F103C8T6 Blue pill Arduino guide**. [Techshopbd.com](http://Techshopbd.com), fev. 2017. Guia do usuário. Disponível em: <[goo.gl/hi2sNm](https://goo.gl/hi2sNm)>. Acesso em: 6 mar. 2018.

TRILHA & CIA. Travessia Petrópolis x Teresópolis (Serra dos Órgãos) [Online]. In: **Mapas, Croquis e Tracklogs - Trilha & Cia**, fev. 2008. Disponível em: <<http://bit.ly/2vIHm8F>>. Acesso em: 28 nov. 2017.

TROMPEDELLER, M. A classification of Petri Nets [Online]. In: **Petri Nets World**. 1995. Disponível em: <<http://www.informatik.uni-hamburg.de/TGI/PetriNets/classification/>>. Acesso em: 25 abr. 2017.

TUNA, G.; GUNGOR, V. C.; GULEZ, K. An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters. **Ad Hoc Networks**. Amsterdam, v. 13, Part A, p. 54-68, fev. 2014. Disponível em: <<https://doi.org/10.1016/j.adhoc.2012.06.006>>. Acesso em: 6 mar. 2018.

VAUGHAN, R. T.; STOY, K.; SUKHATME, G. S.; MATARIĆ, M; J. LOST: Localization space trails for robot teams. **IEEE Transactions on Robotics and Automation**. New York, v. 18 n. 5 p. 796-812, out. 2002. Disponível em: <<http://ieeexplore.ieee.org/document/1067999/>>. Acesso em: 6 mar. 2018.

VOJTECH, L.; NERUDA, M.; SKAPA, J.; NOVOTNY, J.; BORTEL, R.; KORINEK, T. Design of RFID outdoor localization system: RFID locator for disaster management. IN INTERNATIONAL CONFERENCE ON THE INTERNET OF THINGS (IOT), 5, 2015, Seoul. **Proceedings...** p. 4-11. Disponível em: <<http://ieeexplore.ieee.org/document/7356542/>>. Acesso em: 22 fev. 2018.

WANG, B.; MA, M. A Server Independent Authentication Scheme for RFID Systems. **IEEE Transactions on Industrial Informatics**. New York, v. 8, n.3, p. 689-696, ago. 2012. Disponível em: <<http://ieeexplore.ieee.org/document/6140965/>>. Acesso em: 20 fev. 2018.

WANT, R. Near field communication. **IEEE Pervasive Computing**. Mahébourg, v. 10, n.3, p. 4-7, jul. 2011. Disponível em: <<http://ieeexplore.ieee.org/document/5958681/>>. Acesso em: 23 dez. 2017.

WAWERLA, J.; VAUGHAN, R. T. A fast and frugal method for team-task allocation in a multirobot transportation system. In: INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 2010, Anchorage. **Proceedings...** p. 1432-1437. Disponível em: <<http://ieeexplore.ieee.org/document/5509865/>>. Acesso em: 6 mar. 2018.

WOOLDRIDGE, M. **An introduction to MultiAgent Systems**. 2.ed. Chichester: John Wiley & Sons, 2002. 461p.

WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent Agents: Theory and Practice. **The Knowledge Engineering Review**. New York, v. 10, n. 2, p. 115-152, jun 1995. Disponível em: <<https://doi.org/10.1017/S0269888900008122>>. Acesso em: 6 mar. 2018.

WU, H.; WANG, J.; ANANTA, R. R.; KOMMAREDDY, V. R.; WANG, R.; MOHAPATRA, P. Prediction based opportunistic routing for maritime search and rescue wireless sensor network. **Journal of Parallel and Distributed Computing**. Amsterdam, v.111, p. 56-64, jan. 2018. Disponível em: <<https://doi.org/10.1016/j.jpdc.2017.06.021>>. Acesso em: 1 jan. 2018.

WU, V. K. Y.; VAIDYA, N. H. RFID Trees: A Distributed RFID Tag Storage Infrastructure for Forest Search and Rescue. In ANNUAL IEEE COMMUNICATIONS SOCIETY CONFERENCE ON SENSOR MESH AND AD HOC COMMUNICATIONS AND NETWORKS, 7, 2010, Boston. **Proceedings...** p. 1-8. Disponível em: <<http://ieeexplore.ieee.org/document/5508249/>>. Acesso em: 22 fev. 2018.

YAN, Z.; JOUANDEAU, N.; CHERIF, A. A. A Survey and Analysis of Multi-Robot Coordination. **International Journal of Advanced Robotics Systems**. Los Angeles, v. 10, n. 12, p. 1-18, dez. 2013. Disponível em: <<https://doi.org/10.5772/57313>>. Acesso em: 6 mar. 2018.

YAGODA, R. E.; COOVERT, M. D. How to work and play with robots: An approach to modeling human-robot interaction. **Computers in Human Behavior**. Amsterdam, v. 28, n.1, p. 60-68, jan. 2012. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S074756321100166X>>. Acesso em: 20 fev. 2018.

ZIPARO, V. A.; IOCCHI, L.; LIMA, P. U.; NARDI, D.; PALAMARA, P. F. Petri Net Plans - A framework for collaboration and coordination in multi-robot systems. **Autonomous Agents and Multi-Agent Systems**. Cham, v. 23, n. 3, p. 344-383, nov. 2011. Disponível em: <<https://link.springer.com/article/10.1007/s10458-010-9146-1>>. Acesso em: 26 fev. 2018.

ZHANG, F.; XU, Y.; CHOU, J. A Novel Petri Nets-Based Modeling Method for the Interaction between the Sensor and the Geographic Environment in Emerging Sensor Network. **Sensors**. Basel, v. 16, n.10, p. 1-41, set. 2016. Disponível em: <<http://www.mdpi.com/1424-8220/16/10/1571>>. Acesso em: 20 fev. 2018.

ZLOT, R.; STENTZ, A. Market-based Multirobot Coordination for Complex Tasks. **The International Journal of Robotics Research**. Los Angeles, v. 25, n. 1, p. 73-101, jan. 2006. Disponível em: <<https://doi.org/10.1177/0278364906061160>>. Acesso em: 5 mar. 2018.

## APÊNDICE A

### DETALHE DAS TRANSIÇÕES DE SUBSTITUIÇÃO DAS REDES DE PETRI DE ALTO NÍVEL DESENVOLVIDAS EM CPN TOOLS

#### A.1. Coleção Completa das Redes de Petri de Alto Nível Associadas ao Comportamento do Agente *AerialBot*

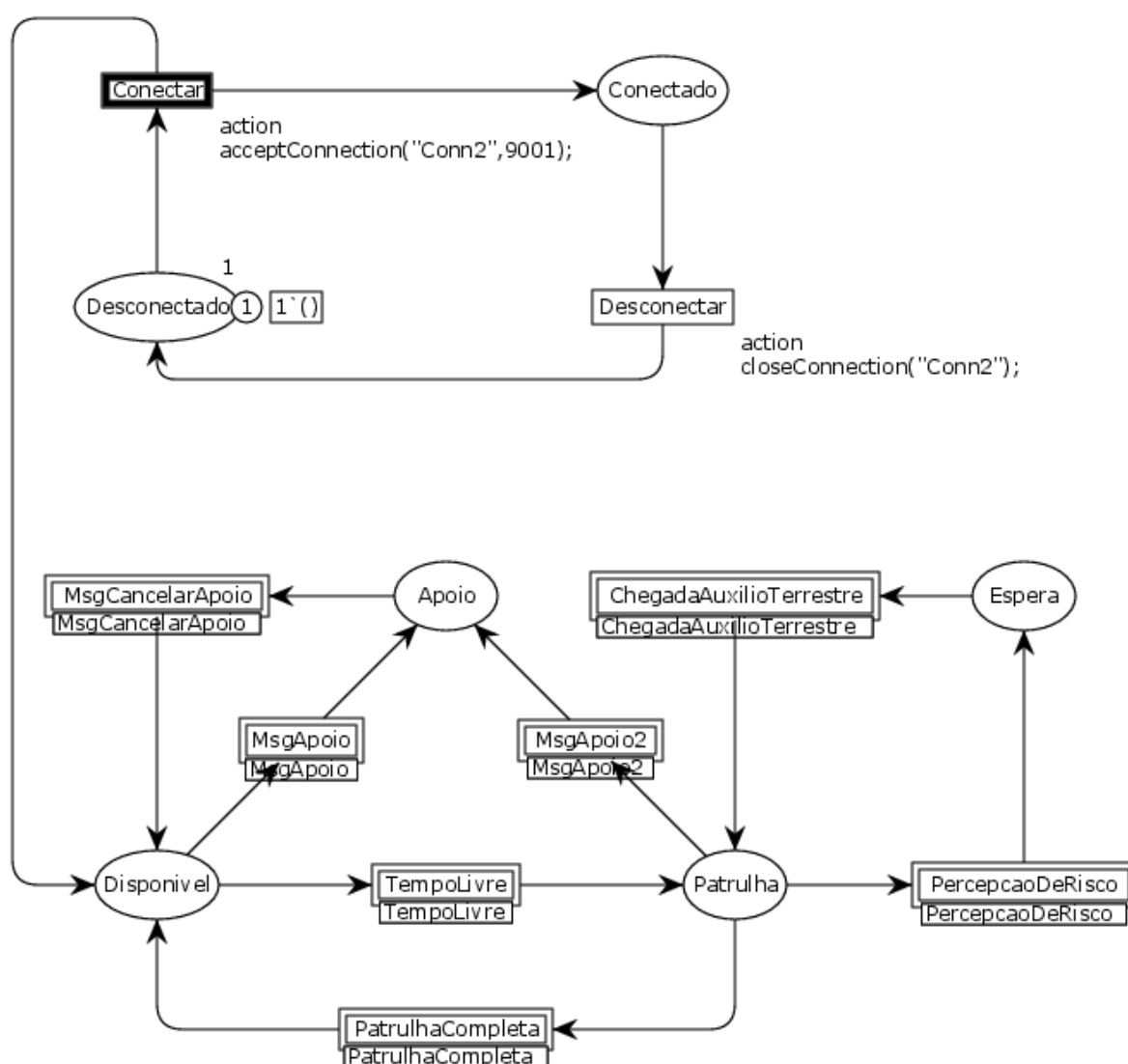


Figura A.1 - Rede de Petri de alto nível com o modelo comportamental do *AerialBot*.

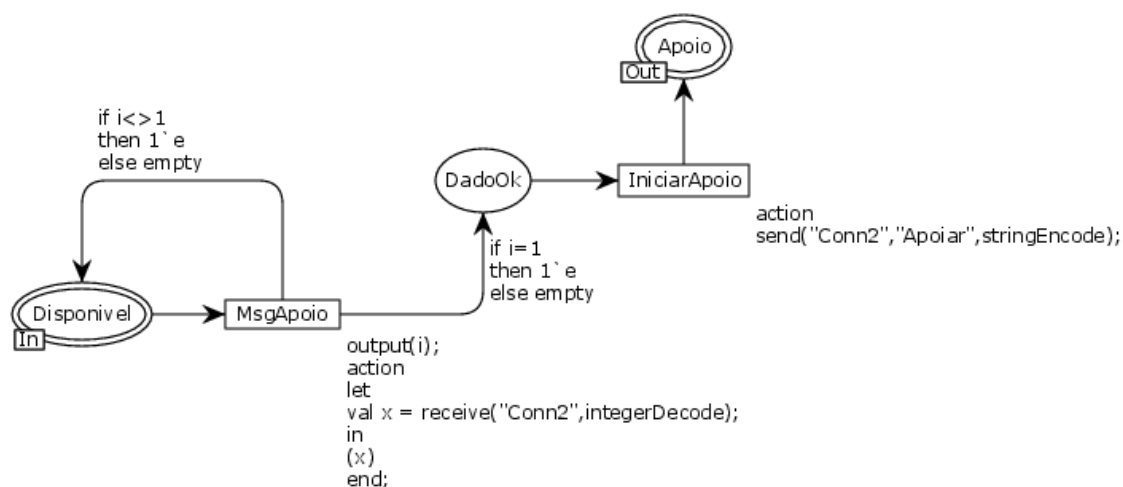


Figura A.2 - Detalhe da sub-rede associada à transição de substituição "MsgApoio".

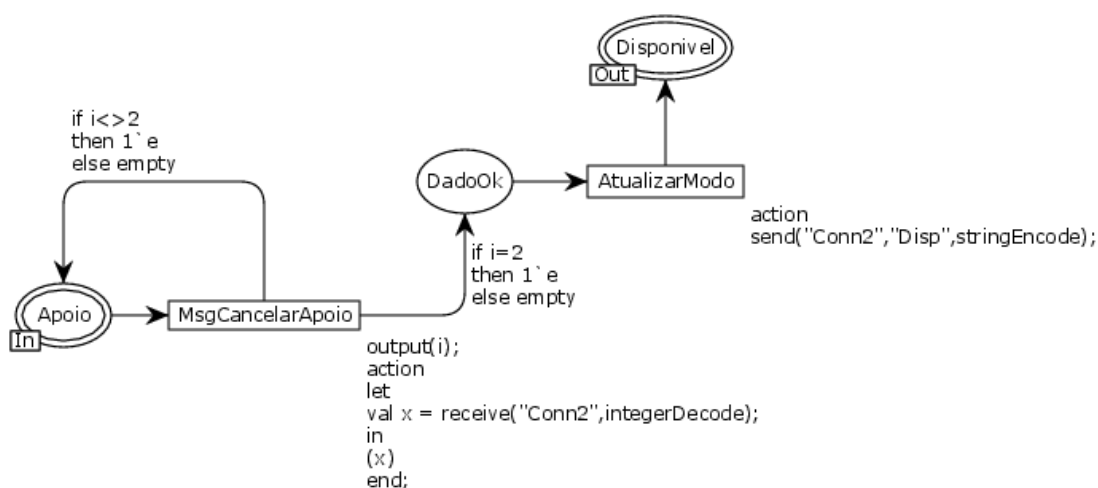


Figura A.3 - Detalhe da sub-rede associada à transição de substituição "MsgCancelarApoio".

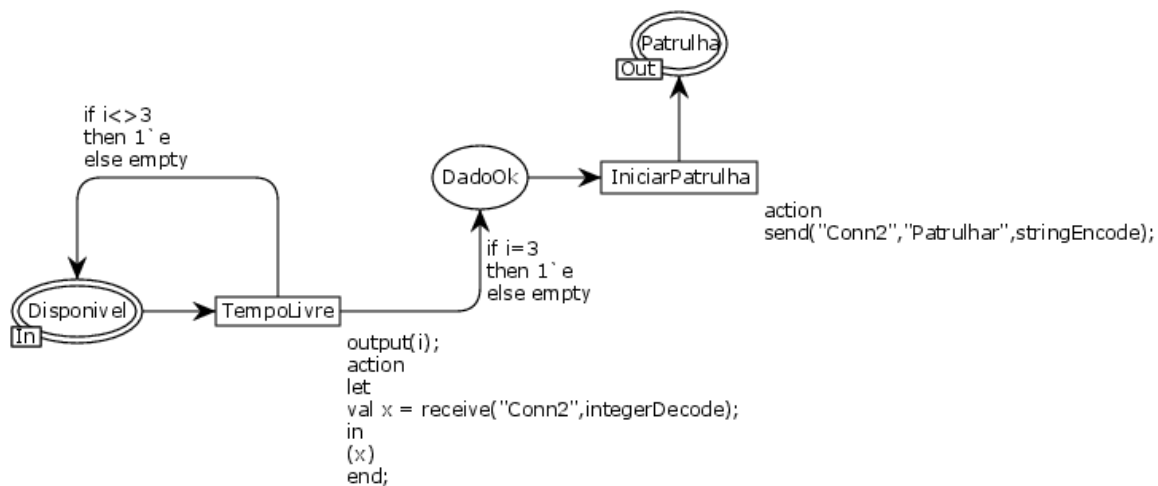


Figura A.4 - Detalhe da sub-rede associada à transição de substituição "TempoLivre".

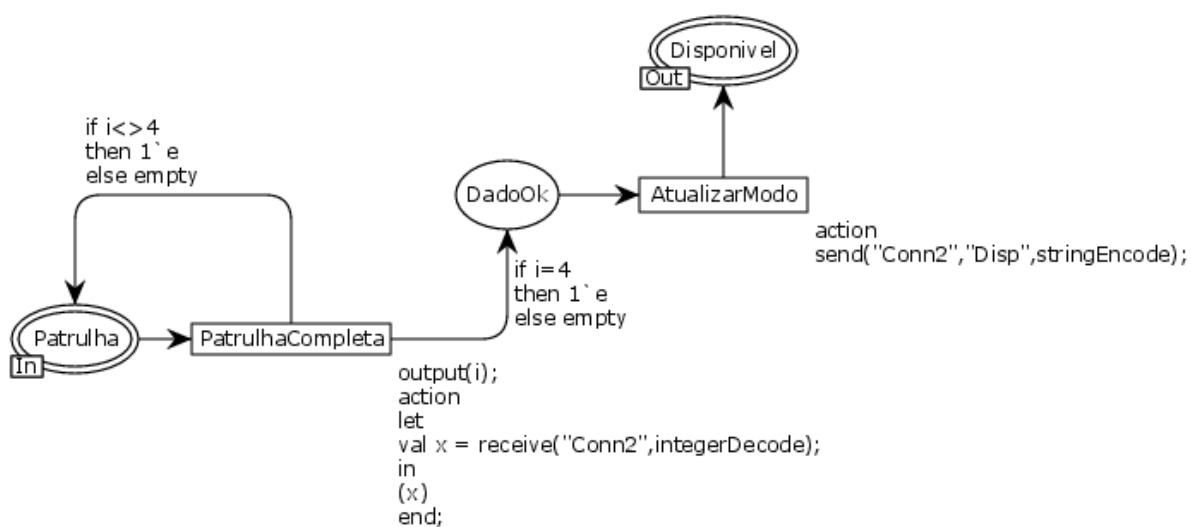


Figura A.5 - Detalhe da sub-rede associada à transição de substituição "PatrulhaCompleta".

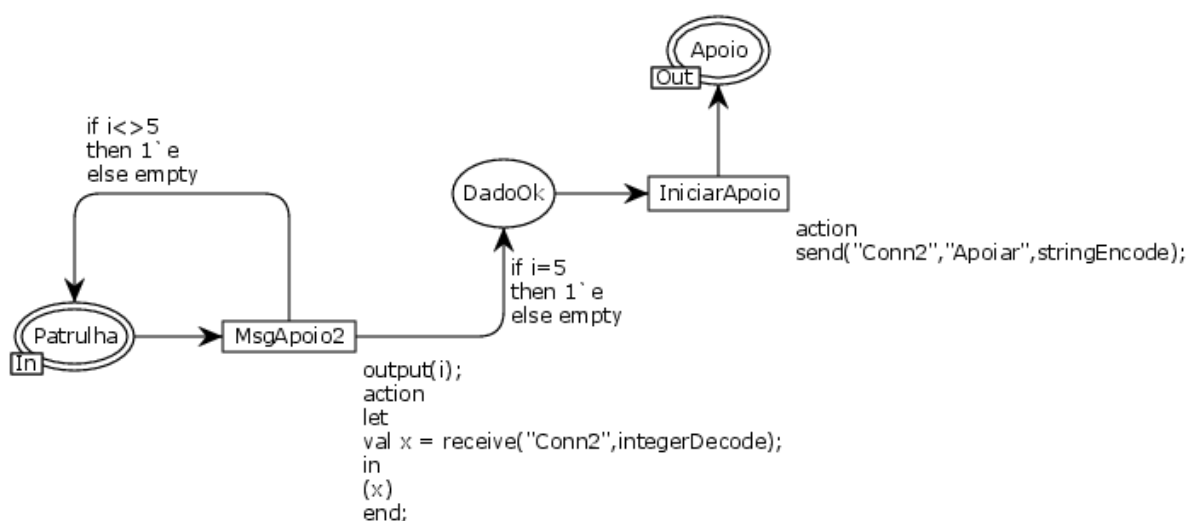


Figura A.6 - Detalhe da sub-rede associada à transição de substituição "MsgApoio2".



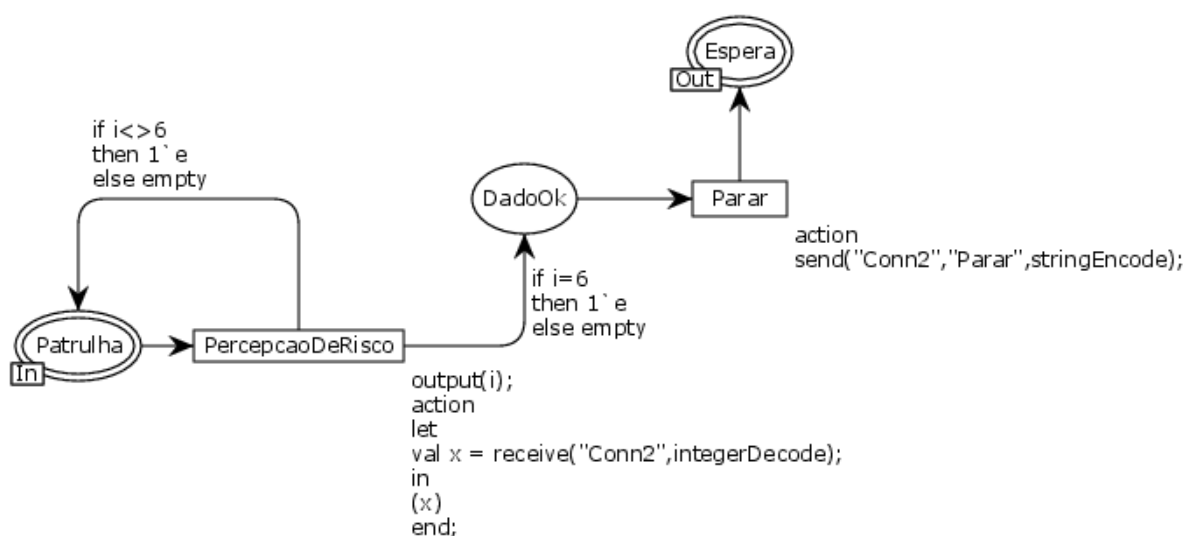


Figura A.7 - Detalhe da sub-rede associada à transição de substituição "PercepcaoDeRisco".

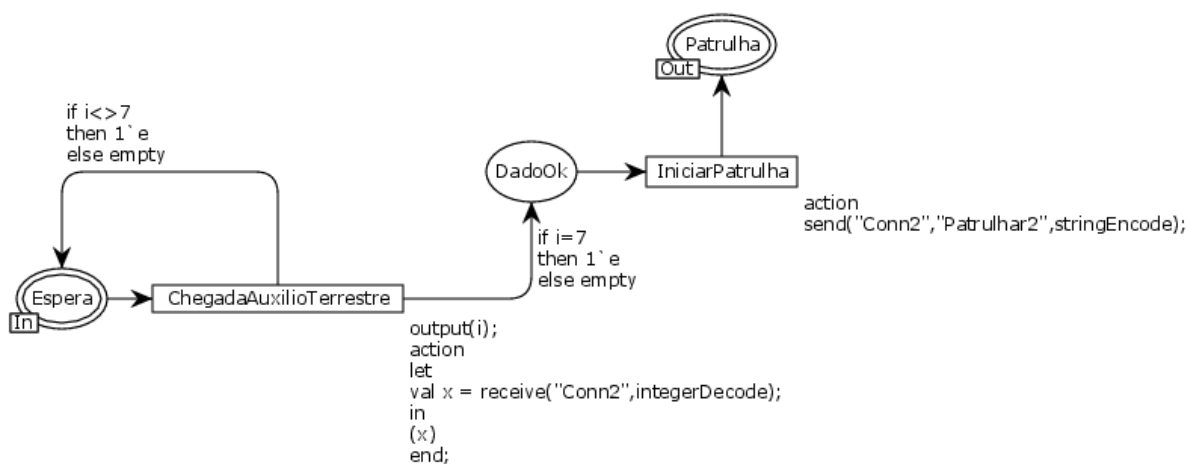


Figura A.8 - Detalhe da sub-rede associada à transição de substituição "ChegadaAuxilioTerrestre".

## A.2. Coleção Completa das Redes de Petri de Alto Nível Associadas ao Comportamento do Agente *GroundBot*

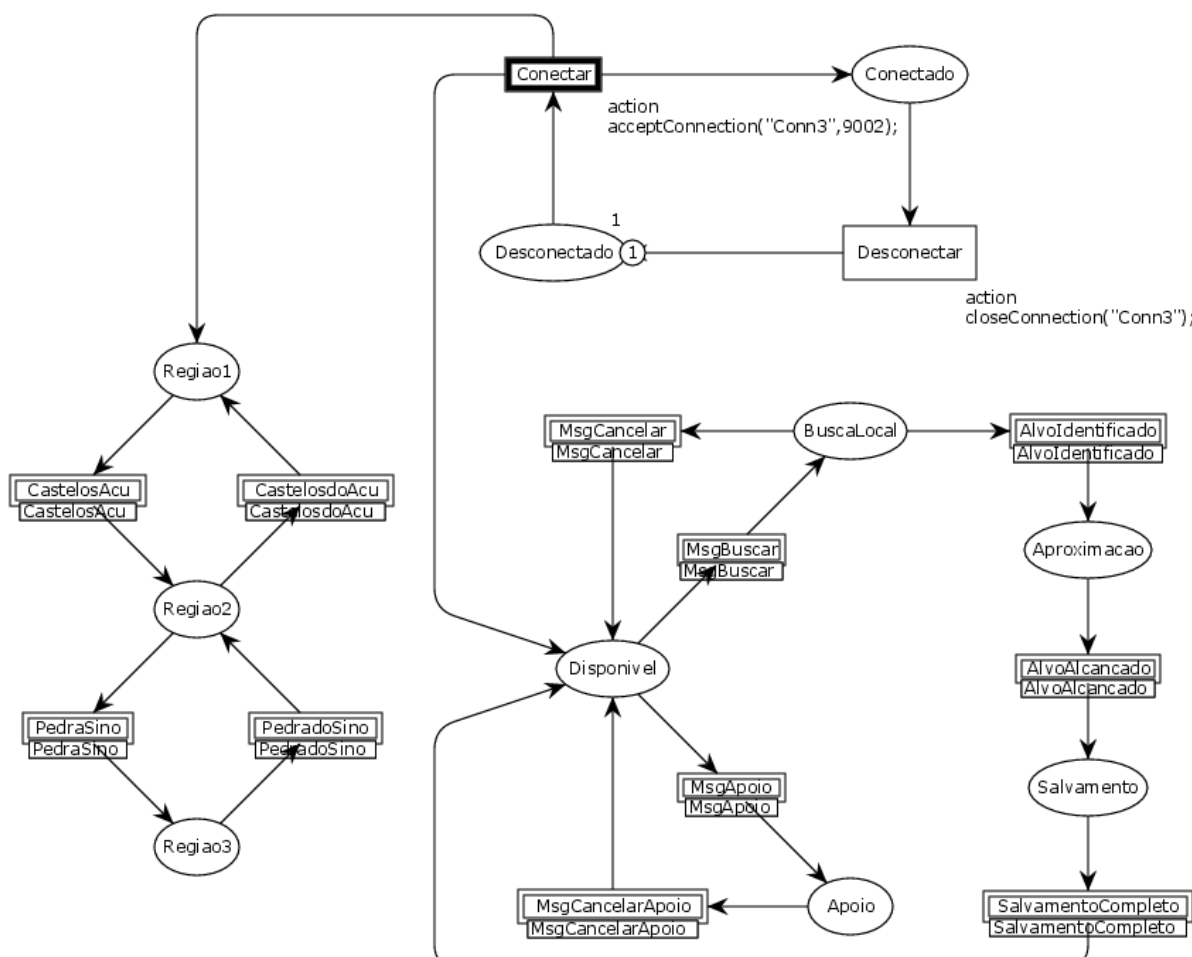


Figura A.9 - Rede de Petri de alto nível com o modelo comportamental do *GroundBot*.

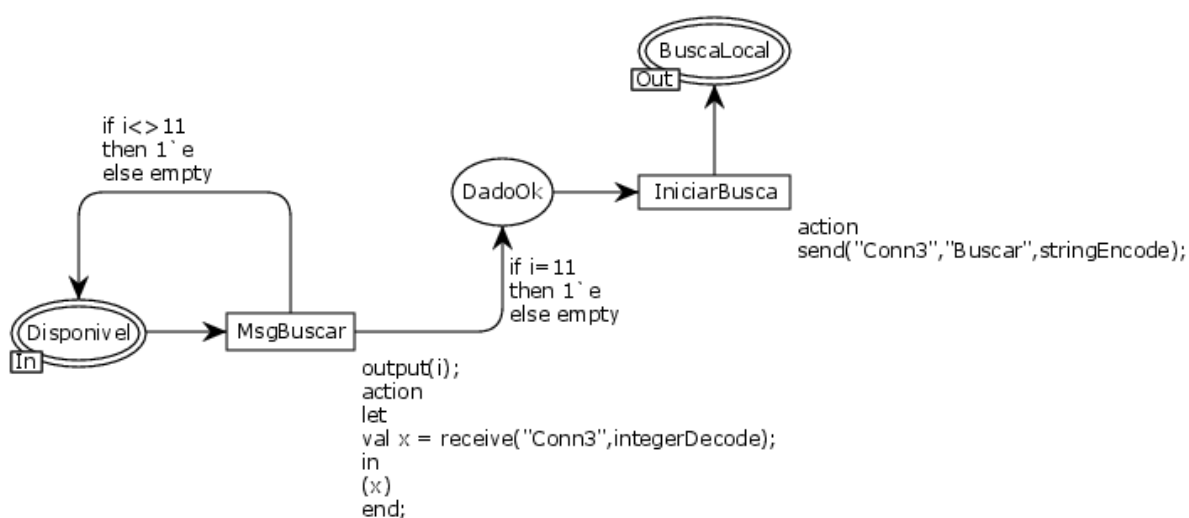


Figura A.10 - Detalhe da sub-rede associada à transição de substituição *"MsgBuscar"*.

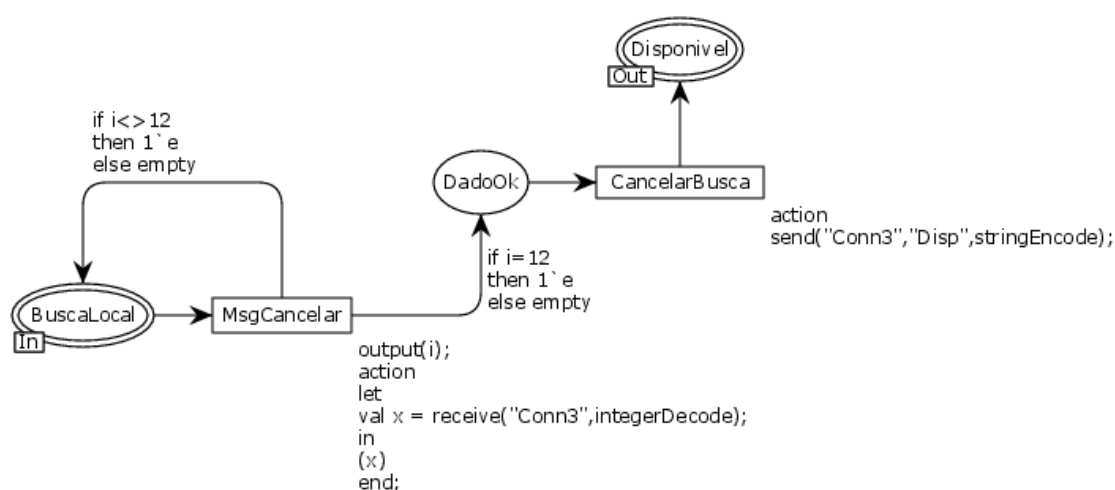


Figura A.11 - Detalhe da sub-rede associada à transição de substituição “*MsgCancelar*”.

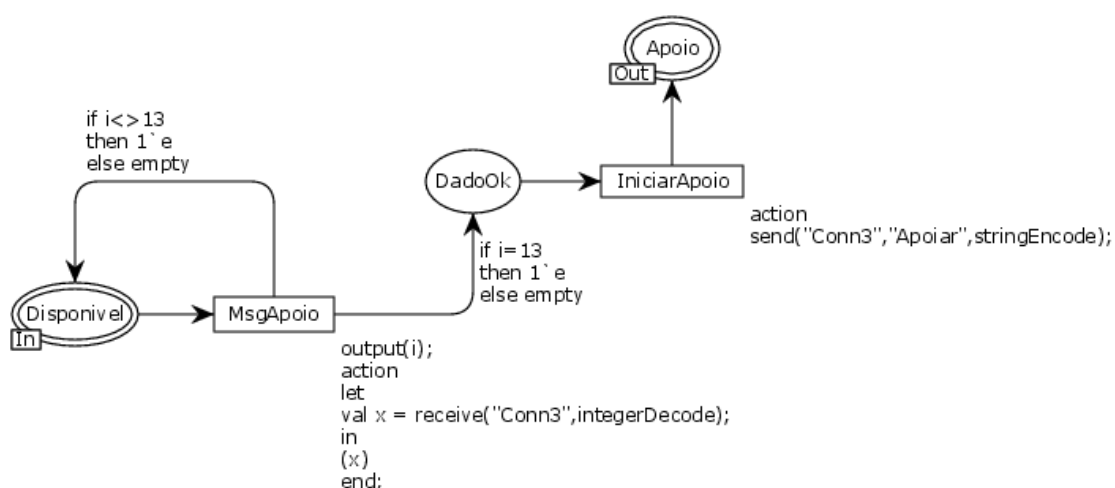


Figura A.12 - Detalhe da sub-rede associada à transição de substituição “*MsgApoio*”.

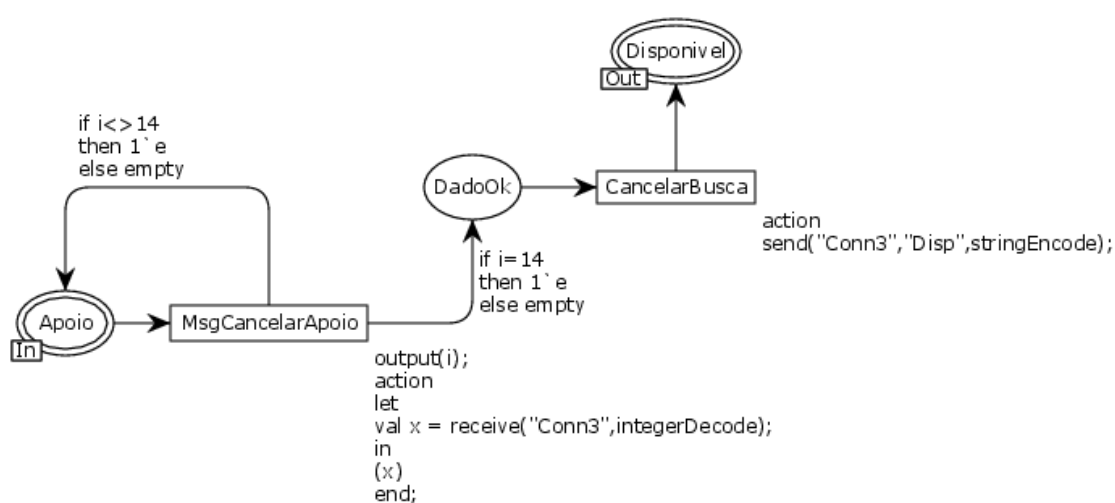


Figura A.13 - Detalhe da sub-rede associada à transição de substituição “*MsgCancelarApoio*”.

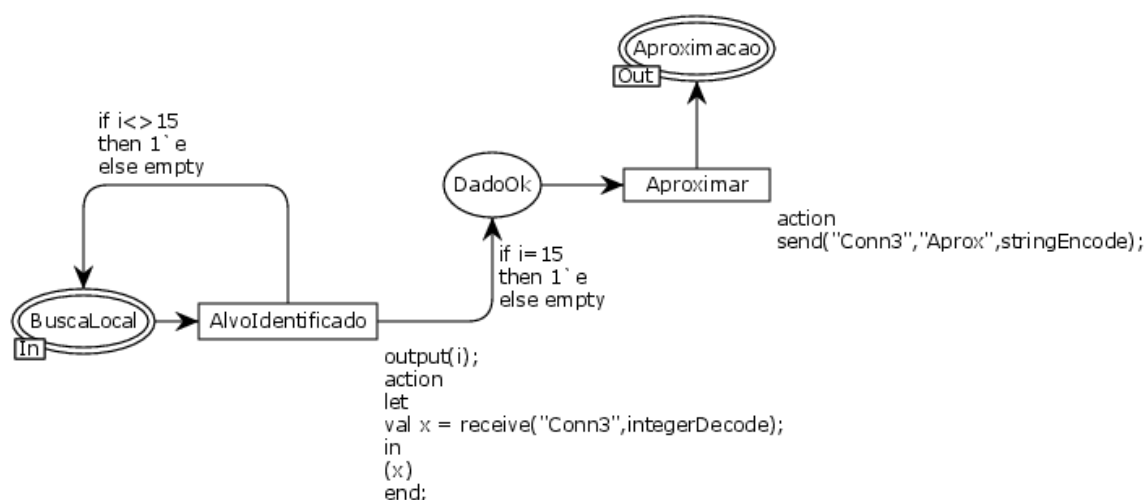


Figura A.14 - Detalhe da sub-rede associada à transição de substituição "AlvoIdentificado".

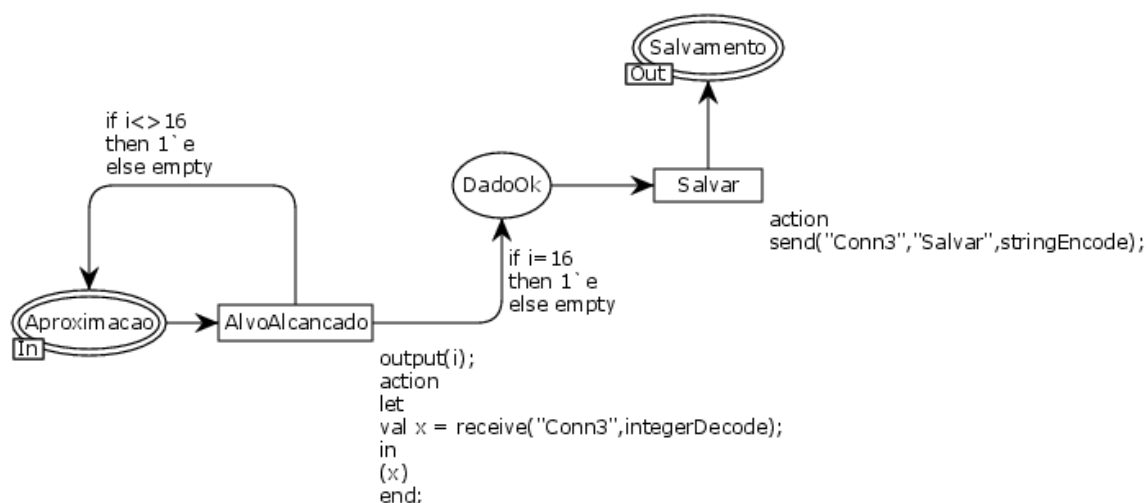


Figura A.15 - Detalhe da sub-rede associada à transição de substituição "AlvoAlcancado".

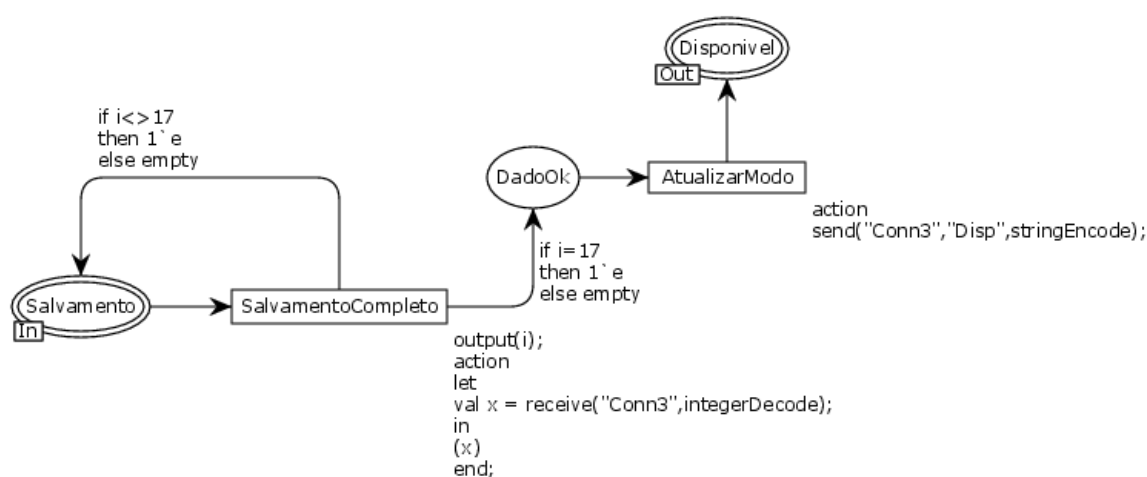


Figura A.16 - Detalhe da sub-rede associada à transição de substituição "SalvamentoCompleto".

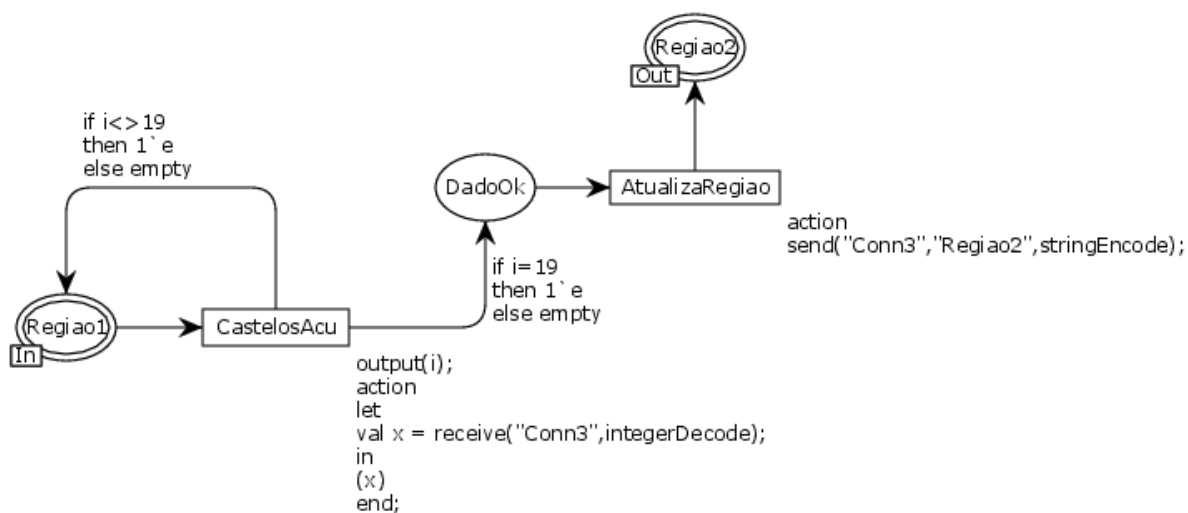


Figura A.17 - Detalhe da sub-rede associada à transição de substituição "CastelosAcu".

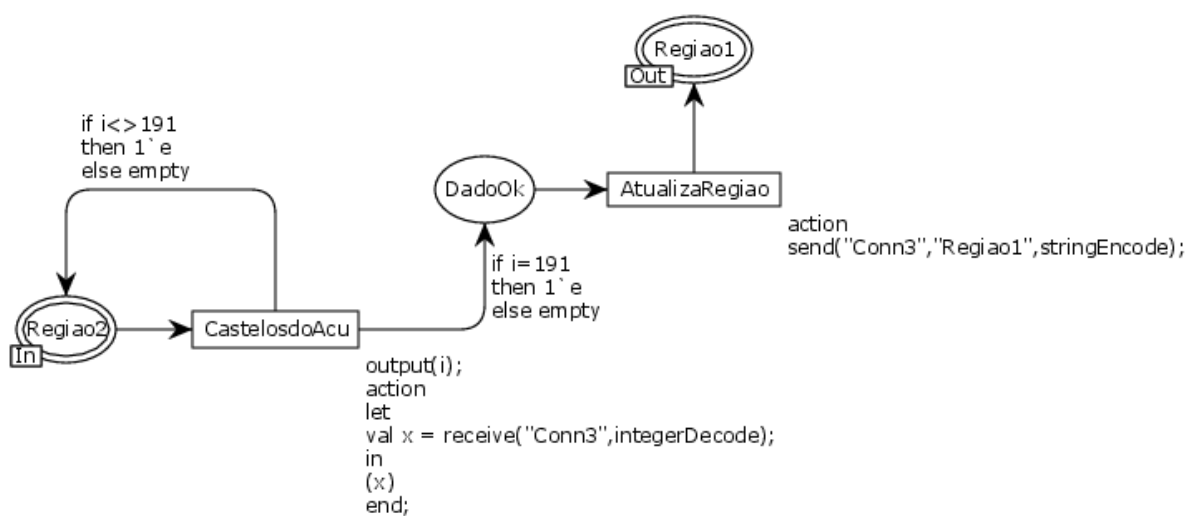


Figura A.18 - Detalhe da sub-rede associada à transição de substituição "CastelosdoAcu".

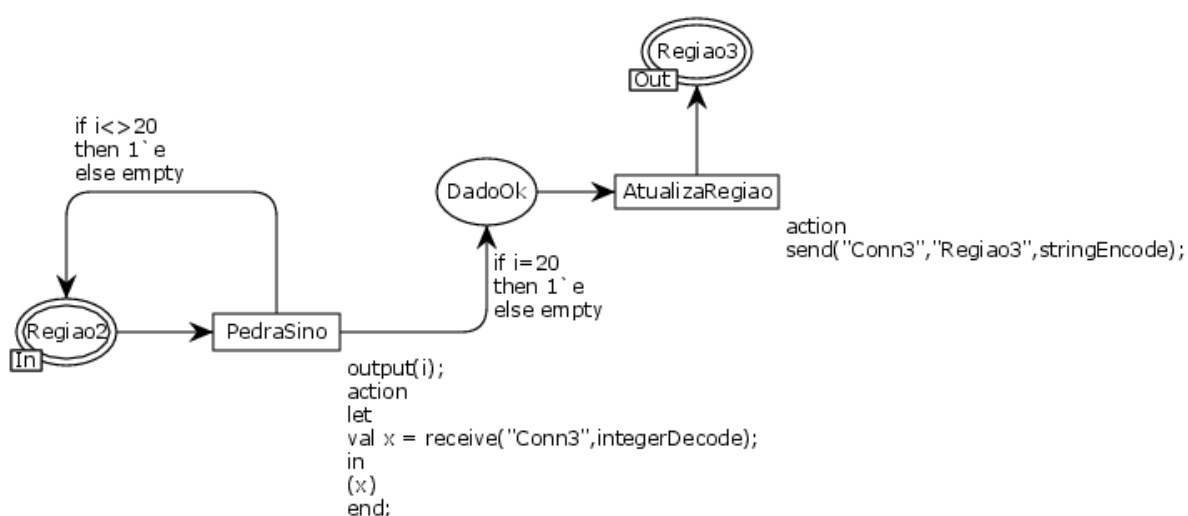


Figura A.19 - Detalhe da sub-rede associada à transição de substituição *"PedraSino"*.

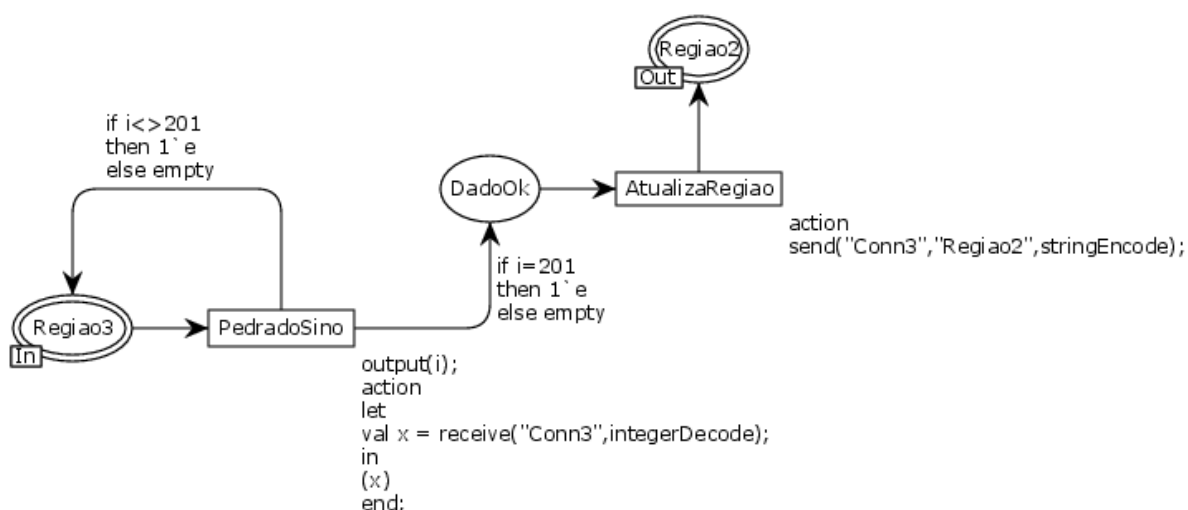


Figura A.20 - Detalhe da sub-rede associada à transição de substituição *"PedradoSino"*.

### A.3. Coleção Completa das Redes de Petri de Alto Nível Associadas ao Comportamento do Agente *Walker*

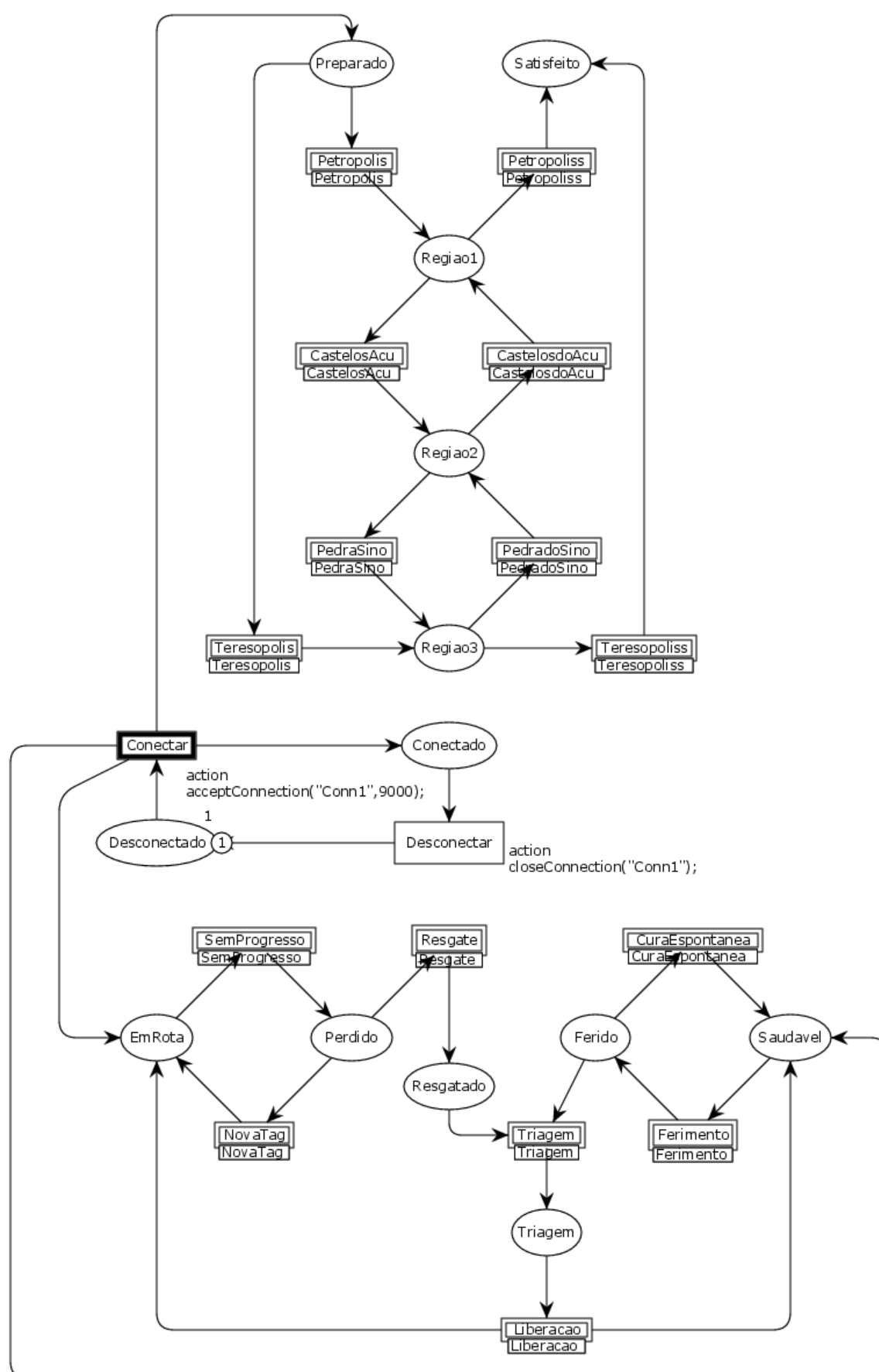


Figura A.21 - Rede de Petri de alto nível com o modelo comportamental do agente *Walker*.

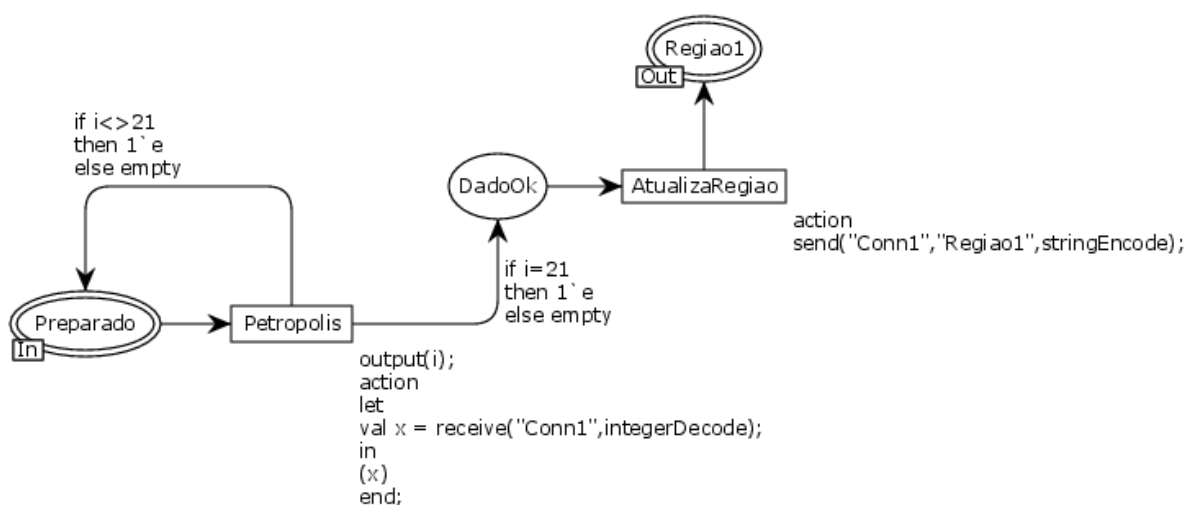


Figura A.22 - Detalhe da sub-rede associada à transição de substituição "Petropolis".

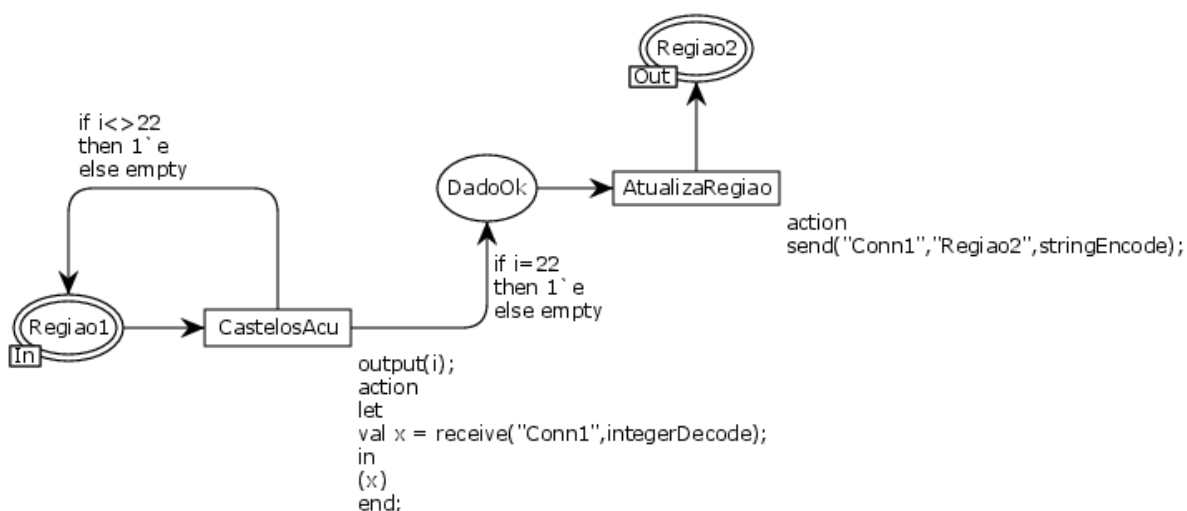


Figura A.23 - Detalhe da sub-rede associada à transição de substituição "CastelosAcu".

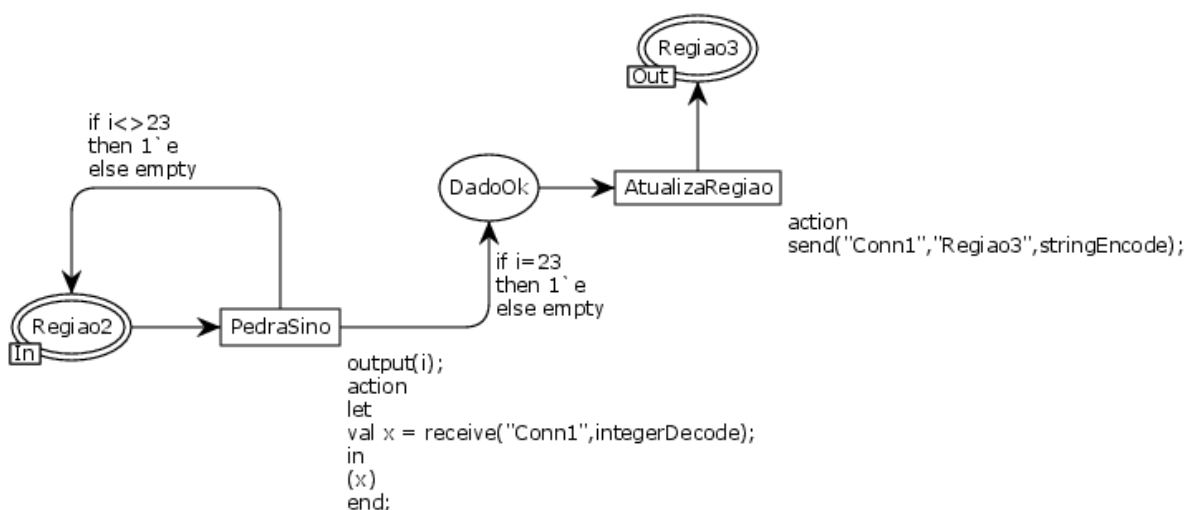


Figura A.24 - Detalhe da sub-rede associada à transição de substituição "PedraSino".



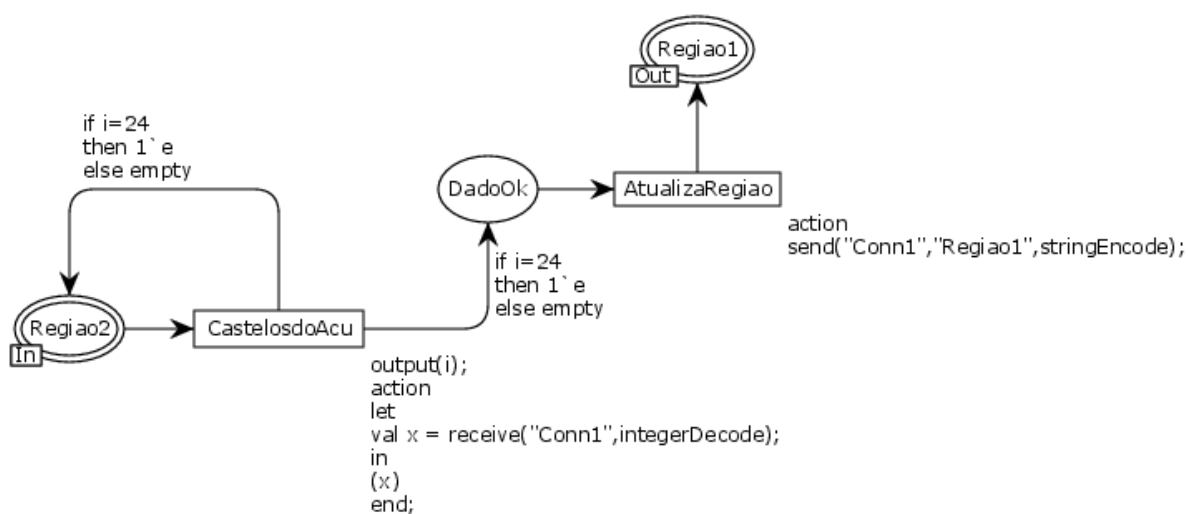


Figura A.25 - Detalhe da sub-rede associada à transição de substituição "CastelosdoAcu".

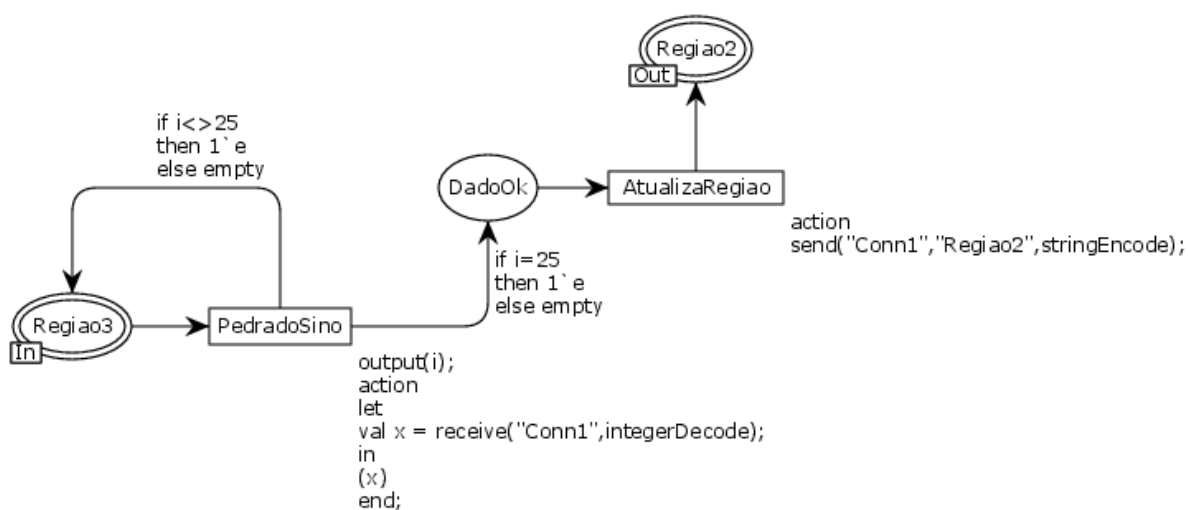


Figura A.26 - Detalhe da sub-rede associada à transição de substituição "PedradoSino".

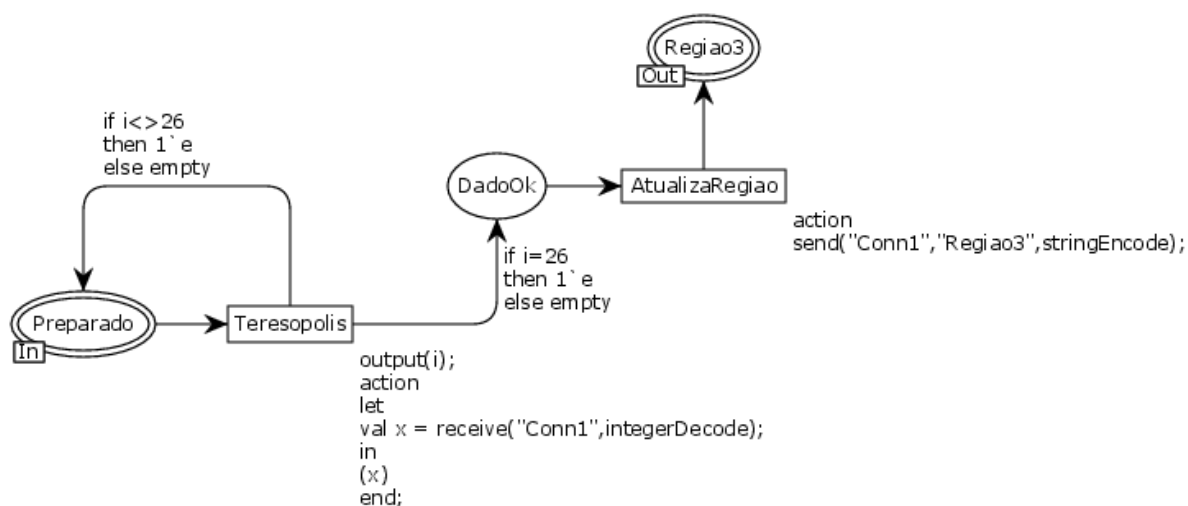


Figura A.27 - Detalhe da sub-rede associada à transição de substituição "Teresopolis".

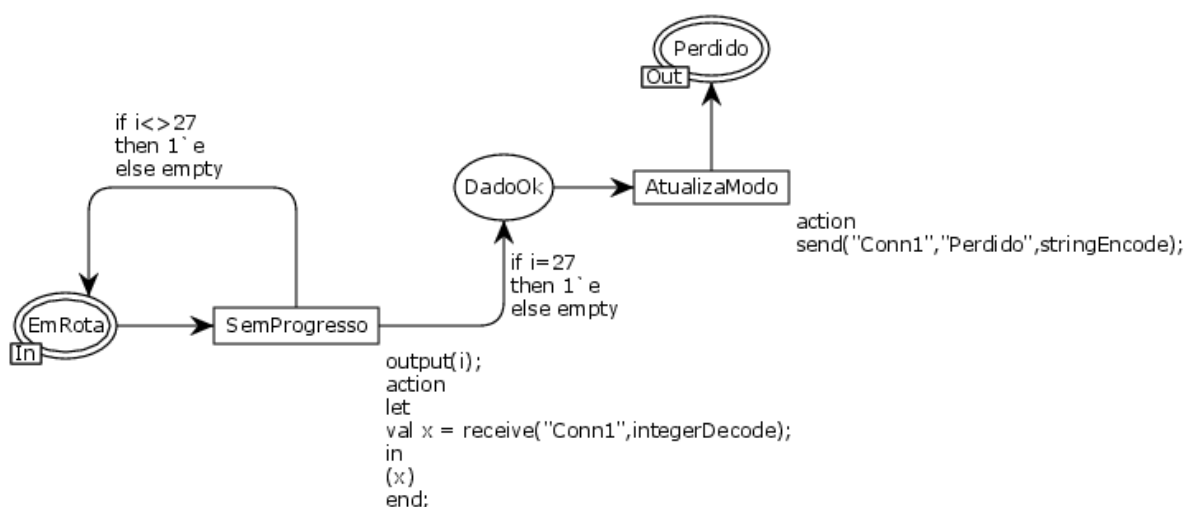


Figura A.28 - Detalhe da sub-rede associada à transição de substituição "SemProgresso".

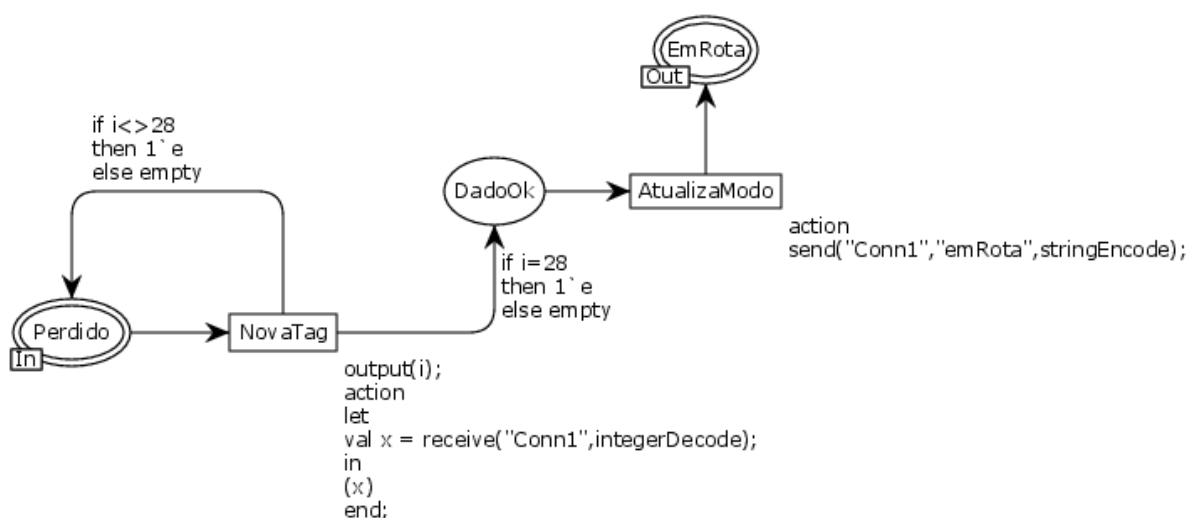


Figura A.29 - Detalhe da sub-rede associada à transição de substituição "NovaTag".

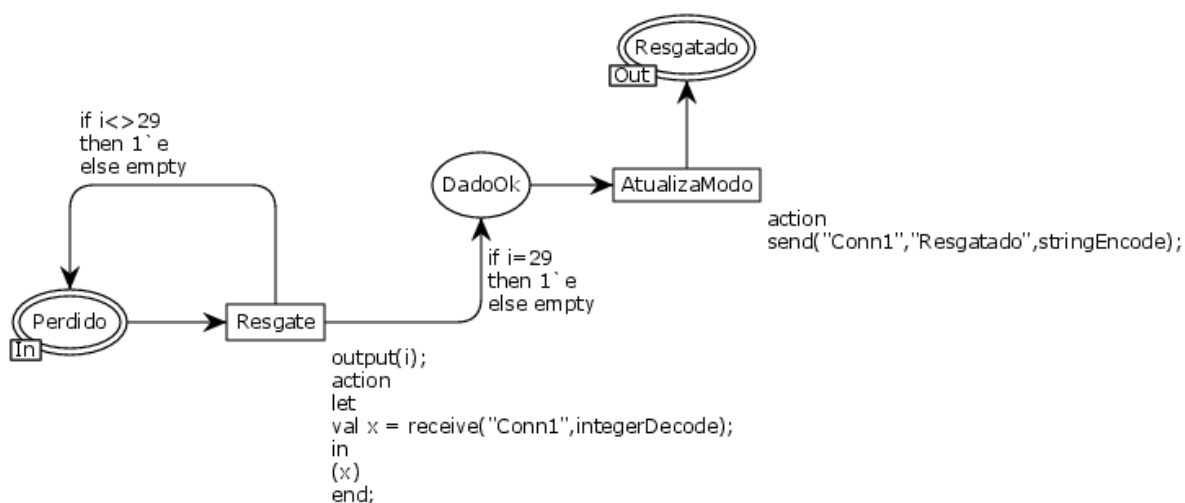


Figura A.30 - Detalhe da sub-rede associada à transição de substituição "Resgate".

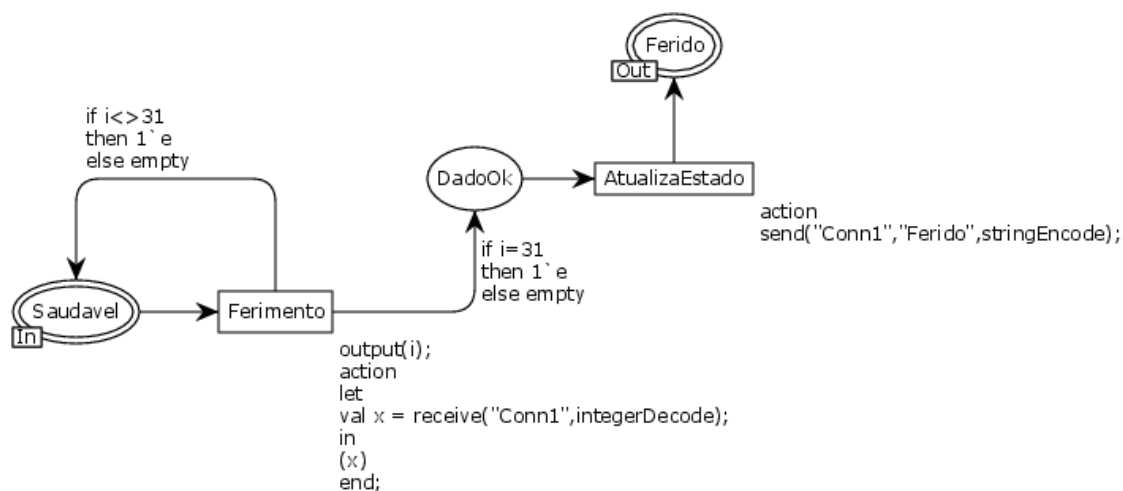


Figura A.31 - Detalhe da sub-rede associada à transição de substituição “Ferimento”.

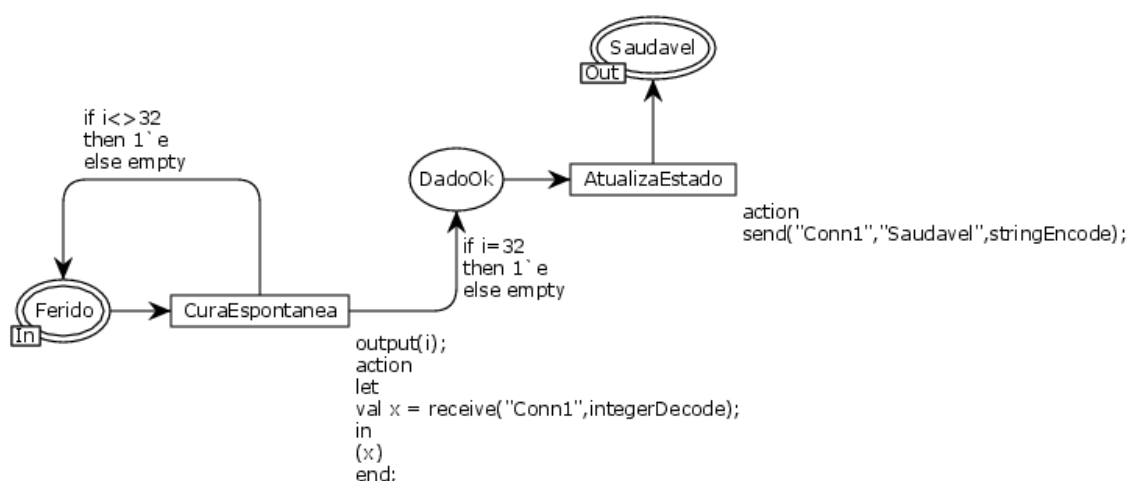


Figura A.32 - Detalhe da sub-rede associada à transição de substituição “CuraEspontanea”.

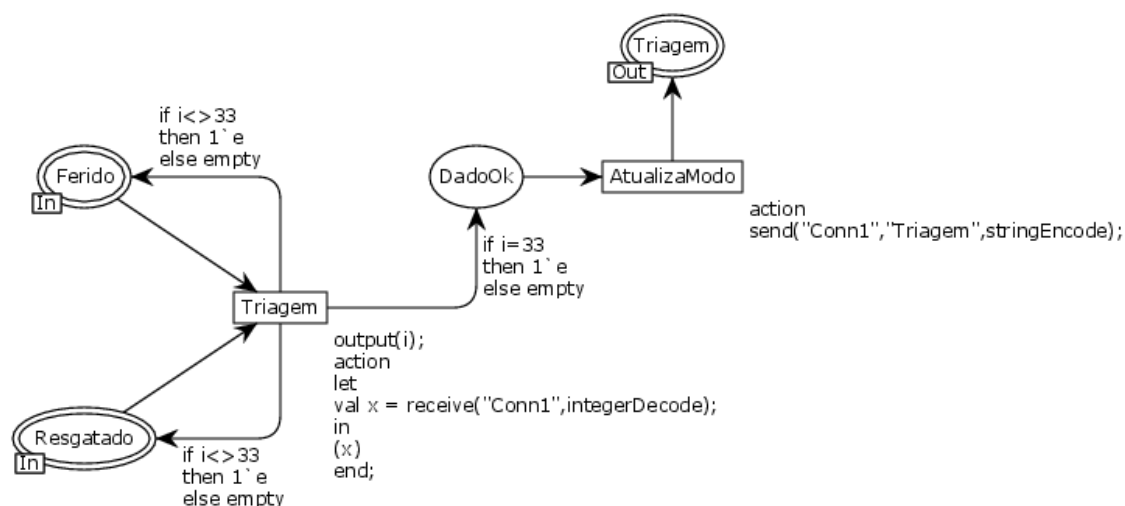


Figura A.33 - Detalhe da sub-rede associada à transição de substituição “Triagem”.

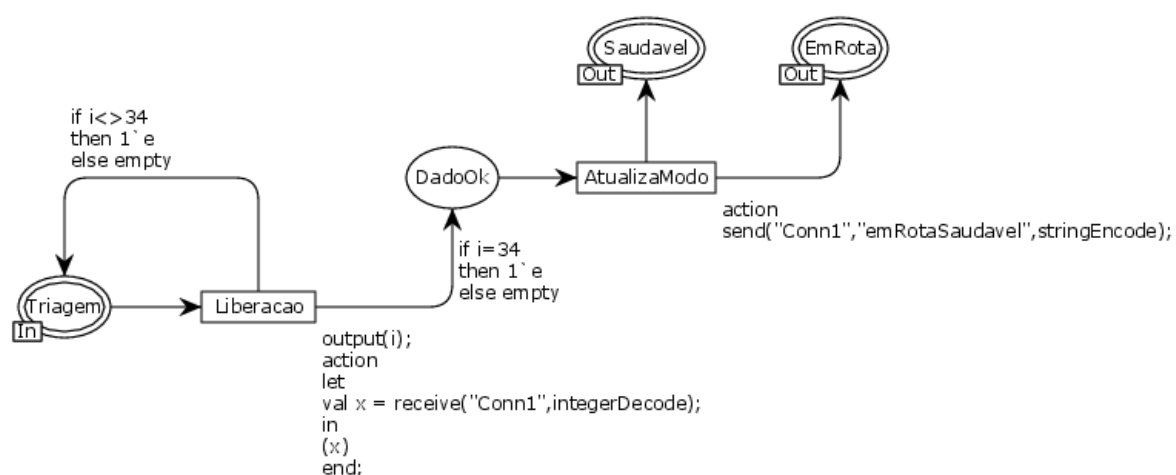


Figura A.34 - Detalhe da sub-rede associada à transição de substituição “*Liberacao*”.

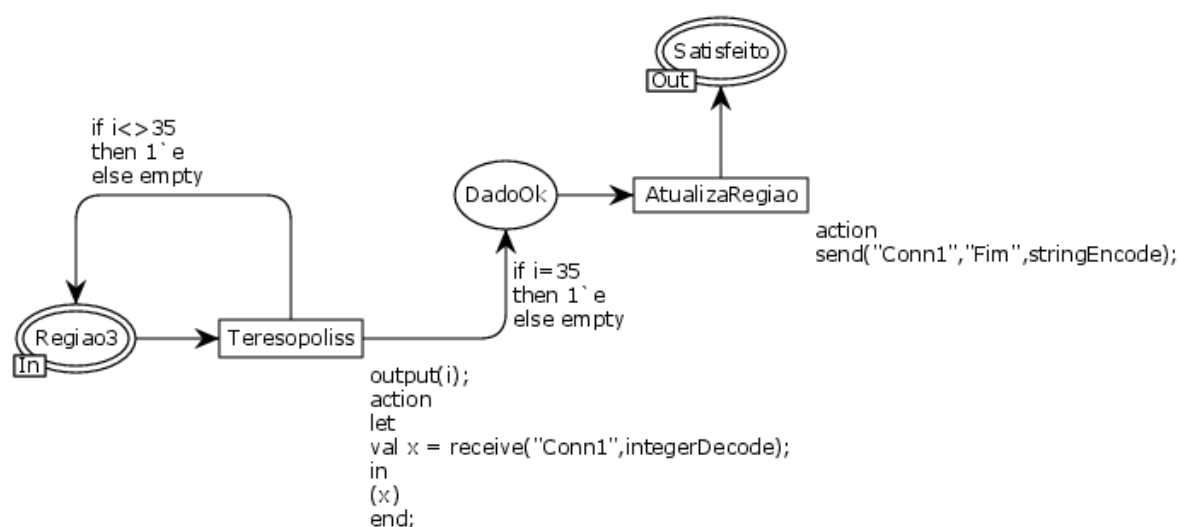


Figura A.35 - Detalhe da sub-rede associada à transição de substituição “*Teresopoliss*”.

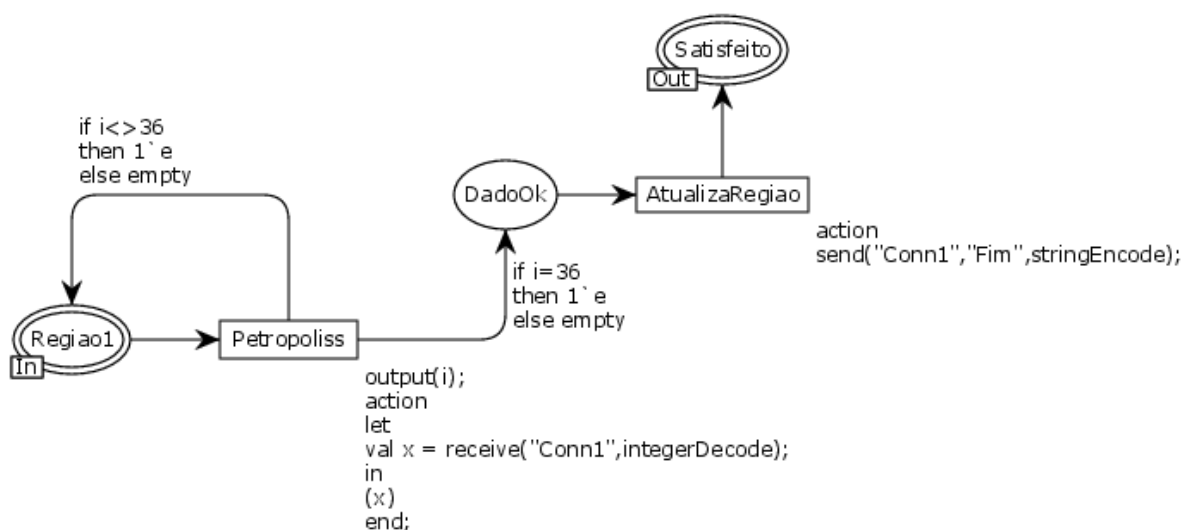


Figura A.36 - Detalhe da sub-rede associada à transição de substituição “*Petropoliss*”.