

MAURILIO DE ARAUJO POSSI

**UMA FERRAMENTA PARA SIMULAÇÃO DO SISTEMA
IMUNOLÓGICO ATRAVÉS DE SISTEMAS MULTIAGENTES:
UM CASO DE ESTUDO DA AUTOIMUNIDADE**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

VIÇOSA
MINAS GERAIS – BRASIL
2012

**Ficha catalográfica preparada pela Seção de Catalogação e
Classificação da Biblioteca Central da UFV**

T

P856f
2012

Possi, Maurilio de Araujo, 1984-

Uma ferramenta para simulação do sistema imunológico através de sistemas multiagentes: um caso de estudo da autoimunidade / Maurilio de Araujo Possi. – Viçosa, MG, 2012.

xiv, 106f. : il. (algumas col.) ; 29cm.

Orientador: Alcione de Paiva Oliveira.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f. 104-106.

1. Agentes inteligentes (Software) - Planejamento.
 2. Software - Desenvolvimento. 3. Autoimunidade.
 4. Sistema imunológico. I. Universidade Federal de Viçosa.
- II. Título.

CDD 22. ed. 005.1

MAURILIO DE ARAUJO POSSI

**UMA FERRAMENTA PARA SIMULAÇÃO DO SISTEMA IMUNOLÓGICO
ATRAVÉS DE SISTEMAS MULTIAGENTES: UM CASO DE ESTUDO DA
AUTOIMUNIDADE**

Dissertação apresentada à
Universidade Federal de Viçosa, como
parte das exigências do Programa de
Pós-Graduação em Ciência da
Computação, para obtenção do título
de *Magister Scientiae*.

APROVADA: 17 de fevereiro de 2012



Ricardo dos Santos Ferreira



Marcelo Lobosco



Alcione de Paiva Oliveira
(Orientador)

*Aos meus pais,
Maurilio Divino Possi e Ione de Araujo Possi*

*A minha noiva,
Renata Maria De Mori Cezario Rezende*

“Estamos na posição de uma criancinha entrando em uma imensa biblioteca, repleta de livros em várias línguas. A criança sabe que alguém escreveu aqueles livros, mas ela não sabe como. Ela não compreende as línguas em que foram escritos. Tem uma vaga suspeita de que a disposição dos livros obedece a uma ordem misteriosa, mas não sabe qual é. Essa, ao que me parece, é a atitude até mesmo do mais inteligente dos seres humanos diante de Deus.”

(Albert Einstein)

AGRADECIMENTOS

Primeiramente, a Deus, por tudo.

Aos meus pais, Maurilio Divino Possi e Ione de Araujo Possi, por todo o amor, confiança e incentivo, que foram essenciais para essa conquista: vocês são exemplos de vida para mim e sem vocês nada disso seria possível.

A minha noiva, Renata De Mori Cezario Rezende, pelo amor, pela amizade, pelo companheirismo, pela compreensão, pela paciência e principalmente pelo apoio para ter ânimo nos momentos difíceis. Você é o meu tesouro.

Aos meus familiares, em especial a minha irmã Mirella de Araujo Possi e meu cunhado Vitor de Almeida Montenegro, e minhas tias Suely Nunes de Araujo e Margarida Posse, por todo o apoio.

Ao meu orientador, Alcione de Paiva Oliveira, um excelente professor e um exemplo como profissional e como pessoa, que acreditou em mim, ainda na graduação, e me incentivou a tentar o mestrado. Obrigado pelos conhecimentos, pelos conselhos e pelas boas conversas, e principalmente pela amizade, que foram fundamentais para a conclusão de mais esse projeto.

Aos professores Fabio Ribeiro Cerqueira, Ricardo dos Santos Ferreira, Vladimir Oliveira Di Iorio, José Luís Braga, e demais professores do Departamento de Informática (DPI) da UFV por toda a formação e ensinamentos recebidos, além das contribuições para esse trabalho.

Aos professores Cristina Maria Ganns Chaves Dias, Rodrigo Siqueira-Batista, Andréia P. Gomes, e demais professores do Departamento de Medicina e Enfermagem (DEM) da UFV por toda a atenção e auxílio no entendimento do SI. Em especial, ao professor Rodrigo, por me lembrar do termo “receptor” no momento certo.

Aos professores Jonas Marçal de Queiroz e Márcio Henrique Pereira Barbosa, grandes amigos e meus coordenadores nos estágios da graduação, pelo apoio e incentivo.

Aos meus chefes, Flávio Vieira Pontes, Rogério Souza e Leonardo Reis, pelo apoio, pela amizade, e principalmente pelos conselhos durante mestrado.

Aos funcionários do DPI, em especial ao Seu Altino Alves de Sousa Filho, secretário da Pós-Graduação, por todas as conversas para descontrair os momentos de tensão e principalmente pela ajuda com as burocracias do mestrado.

Aos amigos e colegas, em especial ao Daniel Rizzo de Carvalho, José Rui Sousa, Gleiph Ghiotto, Paulo Henrique Gonçalves, Carlos Antônio Bastos e Charles Silva, por todo o incentivo, amizade e contribuições para o trabalho.

A CAPES pelo apoio financeiro e à Universidade Federal de Viçosa.

Muito obrigado.

SUMÁRIO

LISTA DE FIGURAS	ix
LISTA DE TABELAS	xi
LISTA DE ABREVIACÕES	xii
RESUMO	xiii
ABSTRACT	xiv
1 Introdução	1
1.1 Motivação	1
1.1.1 Para a biologia	1
1.1.2 Para a computação	3
1.1.3 Por que utilizar modelos <i>in-silico</i> ?	4
1.1.4 Por que a autoimunidade?	6
1.2 Objetivos	6
1.3 Organização da dissertação	7
2 Referencial Teórico	8
2.1 Aspectos gerais do Sistema Imunológico	8
2.1.1 Imunidade Inata	10
2.1.1.1 Imunidade Inata Humoral	10
2.1.1.2 Imunidade Inata Celular	11
2.1.1.3 Limites da Imunidade Inata	14
2.1.2 Imunidade Adaptativa	14
2.1.2.1 Imunidade Adaptativa Humoral	23
2.1.2.2 Imunidade Adaptativa Celular	27
2.1.2.3 Autoimunidade	29
2.2 Modelando o Sistema Imunológico	32
2.2.1 Métodos de Modelagem	32
2.2.1.1 Equações Diferencias Ordinárias	33

2.2.1.2	Autômatos Celulares	34
2.2.1.3	Sistemas Multiagentes	34
2.2.2	Comparação entre os métodos	35
2.3	Sistemas Multiagentes	37
2.3.1	Ambiente	37
2.3.2	Agentes	38
2.4	Modelagem baseada em agentes	40
2.4.1	Criando modelos baseados em agentes	41
2.4.2	<i>Frameworks</i>	43
2.4.2.1	Repast Symphony	44
2.5	Trabalhos Correlatos	45
3	Proposta de modelo do SI para simulação da autoimunidade	47
3.1	Requisitos do modelo	47
3.2	<i>Framework</i> alvo	48
3.3	Ambiente	48
3.3.1	Tempo	49
3.3.2	Espaço	50
3.3.2.1	Projeções	50
3.3.3	Contextos	53
3.3.3.1	Parâmetros	54
3.3.4	Zonas	56
3.3.4.1	<i>Tissue</i>	56
3.3.4.2	<i>Lymphnode</i>	58
3.3.4.3	<i>Circulation</i>	58
3.3.4.4	<i>Bone Marrow</i>	58
Tolerância Central do Linfócito B	58
3.3.4.5	<i>Thymus</i>	59
Tolerância Central do Linfócito T	59
3.3.5	Migração entre zonas	60
3.3.6	Difusão de Substâncias	61
3.3.6.1	Citocinas	63
3.4	Agentes	65
3.4.1	Granularidade	65
3.4.2	Especificidade	67
Antígenos Próprios	67
3.4.3	Afinidade	67
3.4.4	Tipos de Agentes	69
3.4.4.1	<i>Agent</i>	70

3.4.4.2	<i>Antigen</i>	71
3.4.4.3	<i>Antibody</i>	71
3.4.4.4	<i>Virus</i>	72
3.4.4.5	<i>Portal</i>	73
3.4.4.6	<i>Cell</i>	74
3.4.4.7	<i>Parenchyma Cells</i>	75
3.4.4.8	<i>NK</i>	76
3.4.4.9	<i>ThCell</i>	77
3.4.4.10	<i>CTL</i>	78
3.4.4.11	<i>APC</i>	80
3.4.4.12	<i>Macrophage Cells</i>	80
3.4.4.13	<i>Dendritic Cells</i>	82
3.4.4.14	<i>B Cells</i>	83
4	Resultados	85
4.1	Implementação do modelo	85
4.2	Simulação do modelo	87
4.2.1	Teste 1: Resposta primária PAMP-dependente	87
4.2.2	Teste 2: Papel da NK na eliminação de reservatórios de infecção	89
4.2.3	Teste 3: Homeostasia	91
4.2.4	Teste 4: Especificidade e Memória	92
4.2.5	Teste 5: Autoimunidade	94
5	Conclusões	100
5.1	Trabalhos Futuros	101
	REFERÊNCIAS BIBLIOGRÁFICAS	104

LISTA DE FIGURAS

2.1	Os principais mecanismos da imunidade inata e adquirida	9
2.2	As células NK e o MHC-I	13
2.3	Fases da maturação dos linfócitos e a tolerância central	16
2.4	Detalhe do amadurecimento e tolerância central dos linfócitos T	17
2.5	Ciclo de desenvolvimento dos linfócitos	18
2.6	A necessidade de dois sinais para ativação dos linfócitos	18
2.7	Captura e apresentação de antígenos	20
2.8	Especificidade e memória das respostas imunológicas adaptativas	21
2.9	Fases das respostas imunológicas adaptativas	22
2.10	Cooperação entre as imunidades do SI	23
2.11	Opsonização mediada por anticorpos	24
2.12	Funções efectoras dos anticorpos	25
2.13	Ativação dos linfócitos B	26
2.14	Ativação dos linfócitos T	27
2.15	Funções efectoras dos linfócitos	29
2.16	Mecanismos postulados da autoimunidade	30
2.17	Mecanismos pelos quais os patógenos podem promover a autoimunidade	31
2.18	Modelo geral de um agente reativo com estado	40
2.19	Topologias dos agentes	42
3.1	Ilustração da estrutura do espaço definido por uma projeção do tipo Grid	51
3.2	Vizinhança de Moore	52
3.3	Ilustração de um espaço toroidal	52
3.4	Imagem da zona Tissue, implementada	57
3.5	Tolerância Central do Linfócito B	59
3.6	Tolerância Central do Linfócito T	60
3.7	Representação do gradiente de dispersão das citocinas	63
3.8	Classes de Agentes e sua organização	66
3.9	Exemplo de Sequência de <i>Bits</i>	67
3.10	Ilustração do cálculo do Grau de Afinidade	68
3.11	Regras do agente <i>Antibody</i>	71

3.12	Regras do agente <i>Virus</i>	72
3.13	Regras do agente <i>PC</i>	75
3.14	Regras do agente <i>NK</i>	76
3.15	Regras do agente <i>ThCell</i>	78
3.16	Regras do agente <i>CTL</i>	79
3.17	Regras do agente <i>Macrophage</i>	81
3.18	Regras do agente <i>Dendritic</i>	82
3.19	Regras do agente <i>BCell</i>	84
4.1	Ferramenta implementada: <i>Autosimmune</i> . Visão geral da zona <i>Tissue</i>	85
4.2	Ferramenta implementada: <i>Autosimmune</i> . Detalhe da zona <i>Tissue</i>	86
4.3	Evolução da Infecção de um patógeno <i>PAMP</i> -negativo	88
4.4	Evolução da Infecção de um patógeno <i>PAMP</i> -positivo	88
4.5	Evolução da Infecção com a atuação da <i>NK</i>	89
4.6	Evolução da Infecção sem a atuação da <i>NK</i>	90
4.7	Homeostasia do sistema na ausência de patógenos	91
4.8	Especificidade dos agentes <i>BCell</i>	92
4.9	Especificidade dos agentes <i>ThCell</i>	93
4.10	Especificidade dos agentes <i>CTL</i>	93
4.11	Especificidade e Memória dos agentes <i>CTL</i>	94
4.12	Autoimunidade dos linfócitos <i>CTL</i> , em situações normais	96
4.13	Autoimunidade dos linfócitos <i>CTL</i> , em condições normais, com mimetismo molecular	97
4.14	Autoimunidade dos linfócitos <i>CTL</i> , sem o mecanismo de tolerância	97
4.15	Autoimunidade dos linfócitos <i>CTL</i> , sem o mecanismo de tolerância, e com mimetismo molecular	98

LISTA DE TABELAS

3.1	Parâmetros do modelo	55
-----	--------------------------------	----

LISTA DE ABREVIACOES

AC	Autmato Celular
APC	clulas apresentadoras de antgenos, do Ingls <i>antigen-presenting cell</i>
B	linfcito B
BCR	receptor PRR do linfcito B
CAS	Sistema Adaptativo Complexo, do Ingls <i>Complex Adaptative System</i>
CD	do Ingls, <i>cluster of differentiation</i>
CTL	linfcito T citoltico
EDO	Equao Diferencial Ordinria
IAD	Inteligncia Artificial Distribuda
MHC-I	complexo de histocompatibilidade principal classe I, do Ingls <i>class I major histocompatibility complex</i>
MHC-II	complexo de histocompatibilidade principal classe II, do Ingls <i>class II major histocompatibility complex</i>
NK	clula exterminadora natural, do Ingls <i>Natural Killer</i>
PAMP	padro molecular associado a patgenos, do Ingls <i>pathogen-associated molecular pattern</i>
PRR	receptor de reconhecimento de padro, do Ingls <i>pattern recognition receptor</i>
SI	Sistema Imunolgico
SIA	Sistema Imunolgico Artificial
SMA	Sistema Multiagente
TCR	receptor PRR do linfcito T
Th	linfcito T auxiliar, do Ingls <i>T helper</i>

RESUMO

POSSI, Maurilio de Araujo, M.Sc., Universidade Federal de Viçosa, fevereiro de 2012, **Uma ferramenta para simulação do sistema imunológico através de sistemas multiagentes: um caso de estudo da autoimunidade**. Orientador: Alcione de Paiva Oliveira. Coorientador: Fábio Ribeiro Cerqueira.

O sistema imunológico é fundamental para a vida humana. Para a medicina, sua pesquisa pode trazer novas formas de tratamento para doenças até agora incuráveis. Quanto à Ciência da Computação, a inspiração nos mecanismos do sistema imunológico pode desenvolver novas soluções computacionais para os novos problemas com que a Informática precisa lidar. Vindo ao encontro dessa nova demanda, esse trabalho propõe um modelo do sistema imunológico para a simulação de fenômenos relacionados à autoimunidade. A escolha de construir um modelo baseado em agentes foi embasada na literatura. O modelo foi implementado utilizando o framework Repast Symphony, dando origem a uma ferramenta de simulação, chamada Autosimmune. Os testes realizados com a ferramenta evidenciaram o funcionamento coerente do modelo, em especial, em relação à autoimunidade, cujo teste foi capaz de simular a principal hipótese que relaciona a autoimunidade às doenças auto-imunes: o mimetismo molecular.

ABSTRACT

POSSI, Maurilio de Araujo, M.Sc., Universidade Federal de Viçosa, February, 2012, **A tool for simulating the immune system using multi-agent systems: a case study of autoimmunity**. Adviser: Alcione de Paiva Oliveira. Co-adviser: Fábio Ribeiro Cerqueira.

The immune system is fundamental to human life. For medicine, its research may bring new treatments for yet incurable diseases. For Computer Science, inspiration in the mechanisms of the immune system can help in the development of new computational solutions to new problems with which the Computer Science must deal. Coming to meet this new demand, this work proposes a model of the immune system, for the simulation of phenomena related to autoimmunity. The choice in building an agent-based model was based on the literature. The model was implemented using the framework Repast Symphony, giving rise to a simulation tool, called Autosimmune. The tests performed with the tool have shown consistent operation of the model, in particular in relation to autoimmunity. Computational experiments have shown that the tool was able to simulate the main hypothesis that relates the autoimmunity to the autoimmune diseases: the molecular mimicry.

CAPÍTULO 1

INTRODUÇÃO

1.1 Motivação

O sistema imunológico é de extrema importância para a vida humana. Isto é claramente evidenciado pela observação clínica de indivíduos que, por terem alguma deficiência em sua resposta imunológica, tornam-se suscetíveis a infecções graves, possivelmente fatais (ABBAS; LICHTMAN, 2003). Apenas por este motivo, o de tentar auxiliar na melhoria da qualidade de vida das pessoas, já se justificaria a busca por um maior entendimento deste sistema. No entanto, a pesquisa do sistema imunológico torna-se extremamente relevante por diversos outros aspectos, tanto para as ciências biológicas, como a medicina, quanto para as ciências tecnológicas, como a computação. A seguir, serão listados alguns dos possíveis benefícios que um melhor entendimento do sistema imunológico pode prover para essas áreas.

1.1.1 Para a biologia

Para a biologia, e em especial a medicina, um maior entendimento do sistema imunológico pode trazer diversos avanços e novos tratamentos para doenças ainda incuráveis. A seguir, serão mostrados alguns exemplos de possíveis benefícios provenientes da pesquisa desse sistema:

- **Vacinas:** segundo Abbas e Lichtman (2003), a vacinação é o método mais eficaz de proteger indivíduos contra infecções, além de ser a abordagem que levou à erradicação mundial da varíola. Hoje as pessoas estão inseridas em um mundo super-povoado, com constantes ameaças de pandemias, principalmente virais. Entender melhor o sistema imunológico é crucial para o avanço no desenvolvimento de novas vacinas, para que a humanidade tenha poder de resposta durante possíveis

crises. Além disso, testar as vacinas é uma tarefa crucial, uma vez que elas podem ter efeitos colaterais muito sérios. A utilização de modelos computacionais para simulação, conhecidos como sistemas *in-silico*, poderia iniciar estes testes, e detectar possíveis efeitos colaterais;

- **Transplantes de órgãos:** a medicina avançou muito na área de transplantes. No entanto, segundo Abbas e Lichtman (2003), a principal barreira ao sucesso de transplantes na atualidade se deve a rejeição de tecidos, ocasionada pela reação imunológica envolvida. Entender melhor essa reação pode ser a chave para contê-la sem no entanto prejudicar a resposta imunológica aos agentes infecciosos.
- **Tratamento do Câncer:** segundo Abbas e Lichtman (2003), prevenir e combater células cancerígenas é uma das competências do sistema imunológico. No entanto, às vezes essa proteção falha. Contudo, pesquisas recentes têm mostrado que, com certa manipulação, esse sistema consegue combater o câncer com eficácia. Em 2009 a agência reguladora de alimentos e medicamentos dos Estados Unidos, a FDA, aprovou uma “*vacina que usa o próprio sistema imune do paciente para combater o câncer de próstata avançado, que não responde mais à terapia hormonal*” (SNOWDEN, 2009). Já em 2011, em uma pesquisa desenvolvida por Porter *et al.* (2011), os pesquisadores modificaram os linfócitos T de um indivíduo com leucemia para que eles passassem a reconhecer as células cancerígenas como estranhas, e mobilizassem o sistema imunológico contra elas. Como resultado, o câncer do paciente regrediu;
- **Tratamento da Esclerose Múltipla:** A esclerose múltipla é uma doença degenerativa do sistema nervoso central, que atinge cerca de 2 milhões de pessoas em todo o mundo. Sabe-se que se trata de uma doença autoimune, mas no entanto, suas causas ainda não foram esclarecidas. Apesar disso, um estudo recente (SAWCER *et al.*, 2011) mostrou que existe uma grande associação entre a diferenciação do linfócito T-*helper* e a patogenia da esclerose múltipla. Ainda segundo o autor de Sawcer *et al.* (2011), os genes estudados na pesquisa demonstraram claras evidências de estarem associados a processos imunológicos. Novamente, um conhecimento mais aprofundado pode guiar os cientistas a encontrar a relação causa e efeito nas doenças relacionadas ao sistema imunológico;
- **Doenças Autoimunes:** Sem dúvida uns dos tópicos mais intrigantes da imunologia, a autoimunidade é responsável por diversas patologias, atingindo cerca de 2% da população mundial (ABBAS; LICHTMAN, 2007). Isso provavelmente sem contar as diversas outras patologias cujas origens na autoimunidade ainda não foram descobertas. Dentre as doenças mais conhecidas e comprovadamente relacionadas à autoimunidade, estão a esclerose múltipla, a diabetes latente autoimune do adulto, vitiligo, psoríase, lúpus eritematoso sistêmico, doença celíaca, hepatite autoimune,

púrpura autoimune, artrite reumatoide, entre várias outras. Um conhecimento mais profundo do sistema imunológico permitiria a cura de tais doenças, uma vez que sua causa está intimamente ligada a esse sistema.

Além destas, provavelmente existem diversas outras doenças que hoje são consideradas incuráveis ou de causa desconhecida, mas que têm enormes chances de serem vencidas, com um conhecimento aprofundado do sistema imunológico.

1.1.2 Para a computação

Já para a Engenharia, em especial para a Computação, o conhecimento desses sistemas complexos, entre eles o sistema imunológico, se tornou essencial para o avanço tecnológico dessas áreas. Isso porque a humanidade está vivendo uma época em que tudo é dinâmico: inclusive os problemas. O desenvolvimento da humanidade trouxe consigo também o desenvolvimento dos problemas com que esta necessita lidar. Problemas cada vez mais complexos e dinâmicos. Para conseguir resolvê-los, a Computação passou a ter que lançar mão de soluções cada vez mais sofisticadas, demandando sistemas computacionais cada vez mais complexos e dispendiosos.

Segundo Macal e North (2009), os sistemas (humanos, biológicos, econômicos, sociais) que precisam ser analisados e modelados estão se tornando muito complexos, em termos de suas interdependências. Diante disso, ferramentas convencionais já não estão sendo mais viáveis como eram antes (MACAL; NORTH, 2009).

Vindo ao encontro dessa demanda por novas soluções, é que surgiu uma nova abordagem na construção de soluções computacionais: os sistemas bio-inspirados. Esses sistemas têm como principal característica utilizar metáforas de sistemas biológicos naturais no desenvolvimento de suas soluções. Deste modo, buscam em sistemas biológicos a inspiração para resolver esses novos problemas com a mesma eficiência que a Natureza já o faz.

A capacidade de certos sistemas naturais, ou biológicos, vai muito além de qualquer estratégia tecnológica existente na atualidade. Sendo assim, elucidar e aplicar um conjunto geral de princípios que governam o comportamento destes sistemas naturais pode levar ao desenvolvimento de novas formas de engenharia.

(CASTRO, 2001, p. 39)

Dentre esses sistemas biológicos, destaca-se o sistema imunológico, devido as suas várias características interessantes do ponto de vista tecnológico. Os sistemas computacionais inspirados em metáforas do sistema imunológico são chamados de Sistemas Imunoló-

gicos Artificiais (SIA) (FLOREANO; MATTIUSI, 2008). Segundo Floreano e Mattiussi (2008), os SIAs se inspiram em funções, princípios e modelos do sistema imunológico biológico para a solução de problemas computacionais, como aprendizagem de máquina, detecção de anomalia, mineração de dados, segurança computacional, comportamento adaptativo, tolerância a falhas, reconhecimento de padrões, entre outros.

Além disso, a Sociedade Brasileira de Computação, SBC, em seu relatório intitulado “*Grandes Desafios da Pesquisa em Computação no Brasil - 2006 - 2016*” (CARVALHO *et al.*, 2006), destacou cinco “grandes desafios” que deveriam guiar a pesquisa da Computação no Brasil nos próximos anos. Dentre eles, citado como o 2º grande desafio, está a “*modelagem computacional de sistemas complexos artificiais, naturais e sócio-culturais e da interação homem-natureza*” (CARVALHO *et al.*, 2006, p. 6). Este trabalho, enquanto um modelo computacional do sistema imunológico, um sistema natural amplamente reconhecido por sua complexidade, encaixa-se neste item.

1.1.3 Por que utilizar modelos *in-silico*?

Na seção anterior foram mostrados alguns dos benefícios que podemos alcançar com um entendimento melhor do sistema imunológico. Contudo, entender melhor esse sistema não é uma tarefa fácil. Segundo Rapin *et al.* (2010), o sistema imunológico é um dos tópicos mais desafiadores da biologia, e muito embora muito já tenha sido descoberto, através de décadas de estudos *in-vivo* e *in-vitro*, muitos dos seus mecanismos, células, substâncias e interações continuam incógnitos para os cientistas, não podendo ser utilizados nem pela medicina, e nem pela computação.

Uma forma que os pesquisadores podem utilizar para tentar contornar esse problema é optar por uma outra abordagem de estudo, fazendo uso de modelos computacionais do sistema imunológico biológico para simular seu comportamento, a iteração entre suas células e substâncias, e assim tentar entender melhor os seus princípios. Esse tipo de abordagem é conhecida como pesquisa *in-silico*.

Segundo Folcik *et al.* (2007), a presença da bio-complexidade nas pesquisas biomédicas são bem reconhecidas. Como resultado, existe um grande interesse no desenvolvimento de ferramentas de pesquisa *in-silico* para serem usadas juntamente com métodos de pesquisa mais tradicionais. Assim também, Li *et al.* (2009) afirma que é imprescindível que se façam modelos do sistema imunológico para aprender as funções de cada um de seus componentes e mecanismos, de modo que se possa utilizar todo o seu potencial na medicina, e criar aplicações computacionais bio-inspiradas em suas muitas propriedades úteis. Deste modo, completa o autor, modelar o sistema imunológico serve tanto para sua melhor compreensão, como também para usar esses novos conhecimentos como inspiração

no desenvolvimento de soluções para problemas de outras áreas, como a Computação e a Engenharia.

Dentre outros aspectos que tornam a criação de modelos *in-silico* do sistema imunológico relevantes, Li *et al.* (2009) destaca ainda que, na medicina, por exemplo, existem muitas hipóteses sobre como o sistema imunológico reage às infecções, como também há a necessidade de avaliar a eficácia dessas hipóteses em descrever os fenômenos observados. Os modelos *in-silico* podem ajudar os pesquisadores a entender os mecanismos e verificar suas hipóteses. Além disso, podem utilizar o conhecimento adquirido para desenvolver novos tratamentos e, utilizando o mesmo modelo, testar sua eficácia.

Por fim, Li *et al.* (2009) afirma ainda que, além de mais baratos, os modelos *in-silico* são mais rápidos que os estudos in-vivo.

Já os autores Benoist e Mathis (2001) afirmam que, especificamente sobre a autoimunidade, existem diversas hipóteses de como as células T participam das doenças autoimunes. Uma delas é o “mimetismo molecular”, que diz que uma doença autoimune pode ser gerada após uma infecção por um patógeno que tenha um padrão molecular muito parecido com o padrão de um tecido do próprio organismo. No entanto, essas hipóteses ainda não possuem embasamento em dados suficientes para comprová-las, pois é difícil encontrar um padrão molecular com estas características. O autor conclui ainda dizendo que “*o que precisamos neste ponto é de novas abordagens para encontrar [padrões moleculares] candidatos à validação experimental*” (BENOIST; MATHIS, 2001, p. 800), e “*novas ferramentas computacionais e experimentais de triagem podem tornar a tarefa de encontrar padrões moleculares candidatos bem mais fácil*” (BENOIST; MATHIS, 2001, p. 800).

Mais ainda, modelos *in-silico* tornam-se também relevantes à luz das questões éticas. Com a discussão sobre o que é ético ou não, a ciência pode deixar de fazer descobertas cruciais para o desenvolvimento da humanidade. No entanto, é evidente que tais descobertas não justificam ações de natureza desumana. Sistemas *in-silico* que pudessem simular ambientes biológicos com certa exatidão, poderiam evitar que tais questionamentos atrasassem o desenvolvimento.

Para a computação, por sua vez, Li *et al.* (2009) afirma que através da modelagem *in-silico* do sistema imunológico é possível conhecer melhor suas funções e como funcionam suas interações microscópicas e não lineares a nível de célula, as quais podem prover diretrizes para a construção de métodos computacionais distribuídos, paralelos e adaptativos.

1.1.4 Por que a autoimunidade?

Segundo Abbas e Lichtman (2007), a autoimunidade é considerada a fonte de diversas doenças e condições patológicas. É estimado que pelo menos 2% da população mundial sofra de algum tipo de doenças autoimune (ABBAS; LICHTMAN, 2007). Apesar de muito já se ter aprendido sobre a autoimunidade através de modelos experimentais *in-vivo* e *in-vitro*, a etiologia das doenças autoimunes humanas ainda é desconhecida (ABBAS; LICHTMAN, 2007). Essa falta de conhecimento entre a autoimunidade e suas consequentes doenças autoimunes demonstra a necessidade do uso de novas abordagens, como os modelos *in-silico*, na tentativa de investigar e esclarecer as relações entre a autoimunidade e as doenças auto-imunes.

Além disso, ao iniciar este trabalho, foi pensado em um assunto que tivesse bastante documentação, diversas hipóteses, mas que ainda não tivessem alcançado um estado consistente, permitindo que novas abordagens tivessem o seu espaço. Neste caso, a autoimunidade correspondeu ao critério, por ter bastante documentação, algumas hipóteses de como ocorre, mas que ainda não foram suficientemente verificadas.

É nesse contexto que se apresenta este trabalho, uma abordagem *in-silico* de estudo, que sugere a utilização de um sistema multiagente para a simulação do comportamento do sistema imunológico, para tentar entendê-lo melhor, e assim contribuir tanto para a biologia quanto para a computação. Obviamente, é inviável, pelo menos por enquanto, simular o sistema imune como um todo. Ao invés disso, o que se propõe é simular uma parte relevante do sistema imune para o estudo de uma de suas características mais intrigantes, o reconhecimento do que é próprio e a autoimunidade.

1.2 Objetivos

O objetivo deste trabalho é propor um modelo computacional baseado em agentes de um sistema imunológico artificial, inspirado em uma parte do sistema imunológico humano, para o estudo dos fenômenos relacionados à autoimunidade, e às decorrentes doenças autoimunes.

Os objetivos específicos deste trabalho são:

- Reunir e revisar a literatura acerca do sistema imunológico humano e a autoimunidade;
- Identificar o método mais indicado para a modelagem do sistema imunológico artificial;

- Identificar os requisitos para o desenvolvimento de um modelo de sistema imunológico artificial que seja capaz de simular os fenômenos envolvidos nos processos de autoimunidade;
- Propor um modelo de sistema imunológico artificial;
- Desenvolver uma ferramenta para simulação do sistema imunológico, implementando o modelo proposto;
- Evidenciar, através de estudos de caso, a viabilidade do modelo proposto, utilizando da ferramenta desenvolvida;

1.3 Organização da dissertação

No Capítulo 2 estão descritos os conhecimentos adquiridos da literatura pesquisada, que são necessários para a construção deste trabalho. Especificamente, na Seção 2.1, estão os aspectos gerais do sistema imunológico. Já na Seção 2.2, estão descritos os principais métodos encontrados na literatura para a modelagem do sistema imunológico. Na Seção 2.3 estão descritos os principais aspectos dos Sistemas Multiagentes. Na Seção 2.4 estão descritas as indicações presentes na literatura no que tange o processo de modelagem baseada em agentes. E por fim, na Seção 2.5 estão descritos os trabalhos que estão relacionados ao modelo que esta dissertação apresenta.

No Capítulo 3 está a descrição do modelo proposto por este trabalho. Em particular, na Seção 3.1 estão os requisitos utilizados para que o modelo atenda à finalidade do trabalho. Na Seção 3.2 está a justificativa da escolha do *framework* utilizado como base do modelo e como alvo da implementação e sua descrição. Já na Seções 3.4 e 3.3 detalham o modelo em si.

Já no Capítulo 4 estão os resultados alcançados, que incluem a implementação da ferramenta baseada no modelo, e os testes realizados a fim de evidenciar a viabilidade do mesmo.

Finalmente, no Capítulo 5 são apresentadas as considerações finais deste trabalho, além das sugestões de trabalhos futuros.

CAPÍTULO 2

REFERENCIAL TEÓRICO

2.1 Aspectos gerais do Sistema Imunológico

O ser humano está inserido em um mundo hostil, cheio de agentes infecciosos, de diversas formas, tamanhos, composições e comportamentos, mas com um objetivo em comum: usar um **organismo hospedeiro** (como o organismo humano) para infectarem e se replicarem, propagando assim os seus genes. Ao fazerem isso, no entanto, podem causar danos ao organismo, gerando doenças e podendo até levá-lo à morte. Por este motivo, tais agentes infecciosos são comumente chamados de **patógenos** (do Grego, *Pathos*, “sofrimento”, e *Gene*, “gerar”).

Contudo, os organismos, em especial os vertebrados, desenvolveram uma série de mecanismos capazes de defendê-los destes patógenos e dos danos que podem causar. Estes mecanismos têm a capacidade de estabelecer um estado de **imunidade** contra os agentes infecciosos (do Latin *immunitas*, que quer dizer “livre de”). Ao conjunto destes mecanismos, células, e substâncias, deu-se o nome de **sistema imunológico** (SI), e seu estudo é o objetivo da ciência conhecida como imunologia (ROITT; DELVER, 2001).

A imunidade é definida como a resistência a doenças, mais especificamente às doenças infecciosas. O conjunto de células, tecidos e moléculas que são intermediários na resistência às infecções é chamado de sistema imunológico, e a reação coordenada dessas células e moléculas aos microrganismos infecciosos é chamada de resposta imunológica.

(ABBAS; LICHTMAN, 2007, p. 1)

O SI conta com um grande repertório de células especiais, chamadas de **leucócitos** (do grego, *leuko*, branco, e *kytos*, célula), e substâncias diversas, que são responsáveis pela comunicação entre estes, chamadas por isso de “mediadores moleculares”, conhecidas por **citocinas** (do grego, *kytos*, célula, e *kinos*, movimento).

A imunidade conferida pelo SI foi classificada em dois grupos: a **imunidade inata** (também chamada de natural ou nativa) e **imunidade adaptativa** (também chamada de específica ou adquirida). Assim também, os mecanismos que o SI utiliza para alcançar a imunidade foram classificados quanto ao modo de ação de suas respostas imunológicas: quando a eliminação do patógeno é feita diretamente por uma célula do SI, denomina-se a resposta como sendo celular, e quando a eliminação dos patógenos é mediada por moléculas que são secretadas pelos componentes do SI e circulam pelos espaços intracelulares, a resposta é denominada como sendo humoral (FLOREANO; MATTIUSI, 2008).

A imunidade inata é responsável pela proteção inicial do organismo, sendo a mais rápida. Já a imunidade adaptativa se desenvolve mais lentamente, sendo responsável pela proteção mais tardia, embora mais eficaz, contra as infecções (ABBAS; LICHTMAN, 2007). A Figura 2.1 ilustra os principais componentes de cada tipo de resposta e seu tempo de reação.

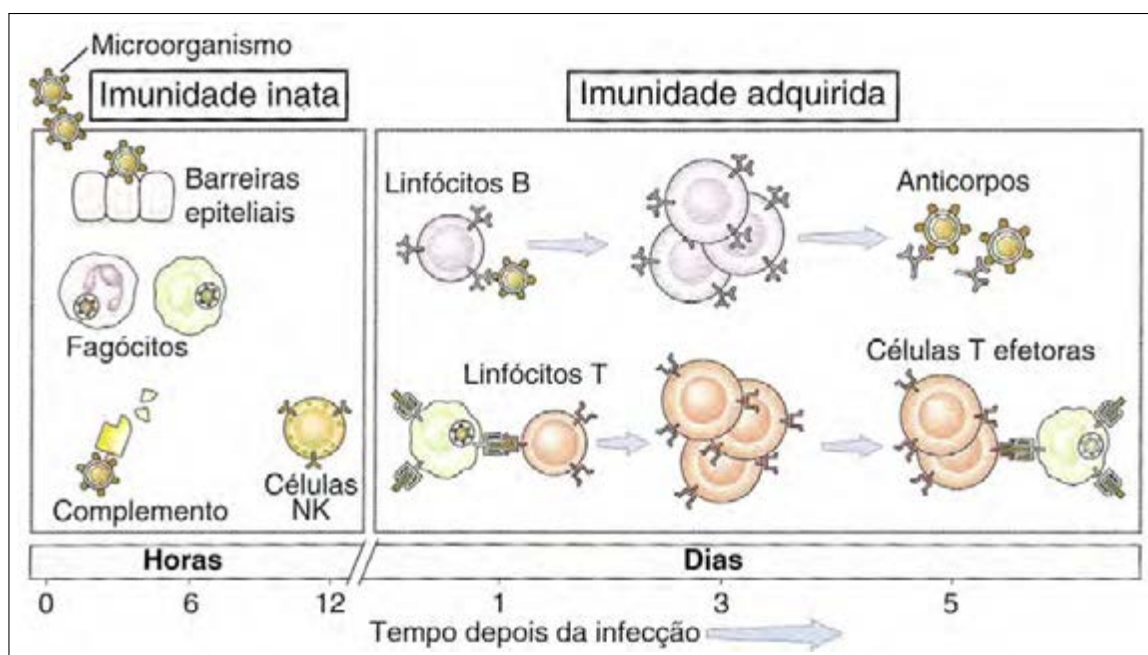


Figura 2.1. Os principais mecanismos da imunidade inata e adquirida.

Fonte: (ABBAS; LICHTMAN, 2007, p. 3)

O SI é constituído por diversas células, de diversas classes, que interagem em cooperação mútua, através de diversos mecanismos e substâncias. Não é o objetivo desta seção tentar descrever toda essa complexidade. Ao invés disso, será feita uma descrição geral desse sistema, tentando aprofundar apenas em suas partes mais relevantes para o entendimento do restante do trabalho.

2.1.1 Imunidade Inata

Segundo Roitt e Delver (2001), a imunidade inata diz respeito aos sistemas de imunidade que não são afetados, em sua eficiência, por contatos anteriores com os agentes infecciosos. Assim também, Floreano e Mattiussi (2008) definem a imunidade inata como sendo o conjunto de mecanismos de defesa que não se alteram durante a vida do organismo hospedeiro.

Os primeiros mecanismos da imunidade inata são as barreiras físico-químicas, que tentam impedir que os agentes infecciosos consigam entrar no organismo do hospedeiro. Entre estas barreiras, estão a pele (que produz ácidos lácticos e graxos, que inibem alguns organismos), as mucosas (e o muco, que age como uma barreira que impede a aderência dos agentes infecciosos às células), além das lágrimas, saliva, urina, ácidos estomacais, pelos, cílios, entre outros (ROITT; DELVER, 2001). No entanto, muitos agentes infecciosos evoluíram, desenvolvendo maneiras de vencer esses obstáculos.

Uma vez que esses agentes infecciosos tenham conseguido ultrapassar estas barreiras, e acessar o organismo do hospedeiro, entram em cena os próximos mecanismos da imunidade inata e suas células. Seu papel é reconhecer e destruir os agentes infecciosos, ou pelo menos, interferir na sua atividade, diminuindo os danos que podem causar, enquanto a imunidade adaptativa prepara sua resposta (FLOREANO; MATTIUSSI, 2008).

Os principais mecanismos são os efeitos tóxicos de algumas substâncias químicas contra os invasores (imunidade inata humoral), e a **fagocitose** (do Grego *phagein*, comer, devorar, *kytos*, vaso oco ou célula, e o sufixo (-osis), processo), processo pelo qual uma célula literalmente “devora” o patógeno (imunidade inata celular) (ROITT; DELVER, 2001).

2.1.1.1 Imunidade Inata Humoral

Segundo Floreano e Mattiussi (2008), o componente mais importante da imunidade inata humoral é o **sistema complemento**. Em resumo, esse sistema é constituído de uma família de moléculas que estão distribuídas pelo organismo do hospedeiro. Essas moléculas têm afinidade com a superfície das membranas celulares, tanto do hospedeiro, quanto dos possíveis patógenos. Uma vez que as primeiras moléculas estejam ligadas a essas membranas, elas permitem que outras moléculas do sistema complemento se liguem a elas, e assim por diante, em um processo que se auto-amplifica, culminando em uma reação capaz lisar essas membranas, ou seja, de criar um verdadeiro orifício em suas estruturas, rompendo-as, e levando a célula ou o patógeno à morte.

No entanto, as células do organismo hospedeiro são capazes de sintetizar outras subs-

tâncias, que agem como antídotos para o sistema complemento, impedindo que os primeiros estágios do processo logrem êxito, eliminando assim os demais estágios. Com isso, as células do hospedeiro não são afetadas pelo sistema complemento. Os patógenos, por sua vez, não conseguem sintetizar esses antídotos, uma vez que são substâncias bastante complexas. E mesmo que consigam evoluir para que as moléculas do sistema complemento não consigam perfurar suas membranas, só o fato de estarem impregnados com tais substâncias já é o suficiente para matá-los. Isto porque os elementos da imunidade inata celular reconhecem e matam (através da fagocitose) qualquer superfície que esteja impregnada com as moléculas do sistema complemento. Deste modo, a função do sistema complemento também é marcar os agentes infecciosos para que sejam destruídos pela imunidade inata celular (FLOREANO; MATTIUSI, 2008).

2.1.1.2 Imunidade Inata Celular

Quando um patógeno entra no organismo e começa a causar danos à células teciduais, estas liberam substâncias que são capazes de mobilizar o SI e recrutar leucócitos para região atacada. Esse acúmulo de leucócitos nos locais da infecção, juntamente com a alteração do tecido local (dilatação e aumento da permeabilidade vascular) é chamado de **inflamação** (ABBAS; LICHTMAN, 2003). As primeiras células a responder a esse chamado são as células da imunidade inata.

Segundo Roitt e Delver (2001), existem dois tipos de células principais nesse tipo de reação da imunidade inata, chamadas ainda de **fagócitos profissionais**: os neutrófilos e os macrófagos. Enquanto os neutrófilos provêm defesas contra bactérias piogênicas, através de seus grânulos com substâncias nocivas aos agentes infecciosos, os macrófagos são mais eficazes em combater bactérias, vírus e protozoários que são capazes de sobreviver no interior das células do hospedeiro. Ambas as células utilizando a fagocitose, ou seja, devorando os patógenos.

Os **neutrófilos** são os leucócitos mais abundantes no sangue, totalizando 10000 por mm^3 , subindo para até 20000 por mm^3 durante uma resposta à infecção. Eles são o primeiro tipo de célula do SI a responder a uma infecção. Agem através da fagocitose, ingerindo os patógenos e morrem após algumas horas apenas (ABBAS; LICHTMAN, 2003).

Já os **macrófagos** são menos abundantes (até 1000 por mm^3) e também agem ingerindo os patógenos. Além de matar os patógenos, os macrófagos são responsáveis por outras tarefas também importantes, entre elas, liberar substâncias que estimulam a cicatrização do tecido lesado, e outras que mobilizam as demais células do SI, além de fagocitar células mortas, literalmente limpando o local da infecção, para que a cicatrização possa ocorrer.

No entanto, essas células precisam saber o que é próprio e inofensivo e o que é estranho ao corpo, e provavelmente nocivo. Para isso, grande parte das células do SI possuem uma estrutura genericamente denominada de **receptores de reconhecimento de padrão** (PRR, do Inglês *pattern recognition receptors*) (FLOREANO; MATTIUSI, 2008). Qualquer estrutura molecular que possua uma parte que possa ser reconhecida por um PRR é chamada de **antígeno**, e a parte específica do antígeno que permitiu seu reconhecimento é chamada de determinante antigênico ou **epítopo** (FLOREANO; MATTIUSI, 2008). Mais ainda, a força, ou a precisão com que um antígeno é reconhecido por um PRR é chamada de **afinidade**. Um mesmo agente infeccioso pode ter diversos epítopos, podendo ser reconhecido por diversos PRRs (FLOREANO; MATTIUSI, 2008).

Contudo, assim como os agentes infecciosos, os tecidos do próprio organismo podem apresentar antígenos. Nesse caso, denomina-se os antígenos de estruturas moleculares do próprio organismo como **autoantígenos** (ABBAS; LICHTMAN, 2007). Os PRRs podem reconhecer os autoantígenos, assim como reconhecem os antígenos dos patógenos. Sendo assim, o SI evoluiu mecanismos diversos para se certificar que suas células diferenciem antígenos e autoantígenos, e não ataquem o próprio organismo.

Na imunidade inata, o SI se valeu do fato de os agentes infecciosos fazerem parte de uma linha evolutiva diferente do hospedeiro, e por isso, possuírem certas estruturas moleculares que não são encontradas no organismo saudável do hospedeiro. Por esta razão, os PRRs das células da imunidade inata são configurados de tal maneira que consigam reconhecer apenas antígenos não-próprios. Os macrófagos e os neutrófilos, por exemplo, possuem receptores PRR configurados de maneira a ter como alvo padrões moleculares típicos de agentes infecciosos apenas, e que não são produzidos pelo organismo do hospedeiro. Esses padrões moleculares são denominados de **padrões moleculares associados a patógenos** (PAMPs, do Inglês *pathogen-associated molecular patterns*) (ROITT; DELVER, 2001).

Como os PRRs da imunidade inata são codificados no DNA, eles não podem ser modificados durante a vida do organismo hospedeiro. Desta forma, eles devem ter como alvo PAMPs que não sejam facilmente modificados ou ocultados pelos agentes infecciosos. Por isso, bons alvos para os PRRs da imunidade inata são PAMPs responsáveis por funções cruciais, e que precisam estar expostos, como os padrões moleculares dos flagelos de uma bactéria, ou que são constituintes fundamentais da sua membrana celular (FLOREANO; MATTIUSI, 2008).

Desta forma, qualquer agente infeccioso que apresente um padrão molecular reconhecido como sendo um PAMP e que esteja acessível aos PRRs dos macrófagos e neutrófilos, será fatalmente fagocitado e eliminado. Assim também, os agentes infecciosos que estejam cobertos pelas proteínas do complemento, são reconhecidos como nocivos e também são

fagocitados.

Entretanto, alguns patógenos logram êxito em infectar uma célula do hospedeiro e ficam fora do alcance dos fagócitos. Uma vez lá, agem injetando seu código genético no seu interior, e transformando-a em uma replicadora, criando novos indivíduos semelhantes ao patógeno, transformando-a em um “depósito de infecção” (ABBAS; LICHTMAN, 2007). Nesse caso, de patógenos intracelulares, o SI inato conta com o auxílio de uma classe de leucócitos especializados em exterminar esse tipo de célula infectada: são as **células exterminadoras naturais** (NK, do Inglês *Natural Killer*).

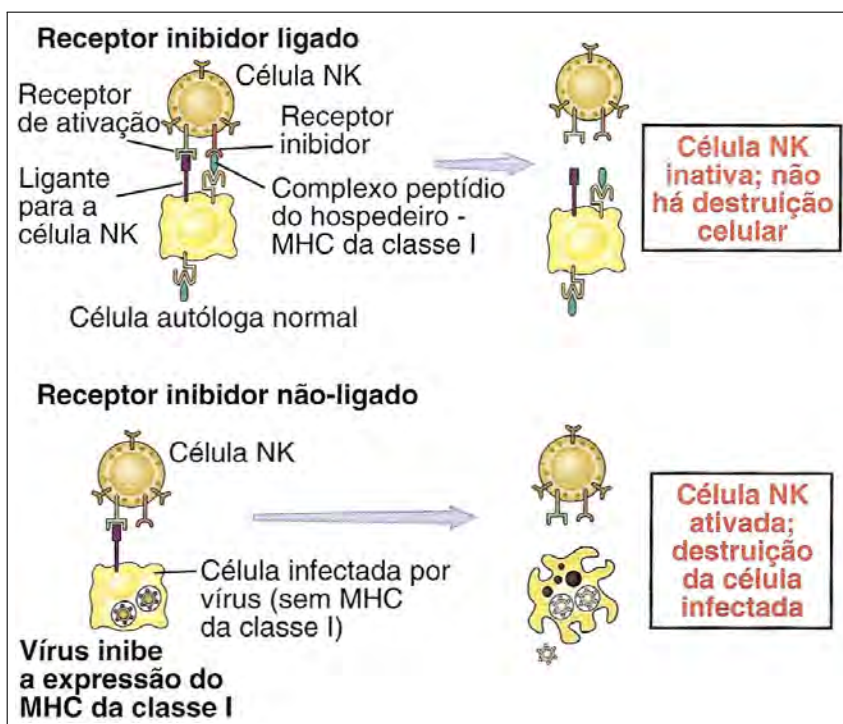


Figura 2.2. As células NK e o MHC-I.

Fonte: (ABBAS; LICHTMAN, 2007, p. 35)

As células NK são capazes de verificar se uma célula está infectada, e, caso esteja, liberam substâncias que levam a célula à apoptose, *i.e.*, ao processo de morte celular programada, eliminando assim o depósito de infecção e expondo os patógenos à ação dos fagócitos profissionais e do sistema complemento (ROITT; DELVER, 2001).

No entanto, para isso, as células NK precisam de um mecanismo para verificar se uma célula está infectada ou não. Para resolver isso, toda célula nucleada do organismo hospedeiro é capaz de apresentar, em sua superfície, uma amostra de todas as proteínas que estão sendo sintetizadas em seu interior. Isso é feito através de um conjunto de moléculas denominado **complexo de histocompatibilidade principal classe I** (MHC-I, do Inglês *class I major histocompatibility complex*).

Uma célula sadia produz e apresenta, através do MHC-I, apenas proteínas próprias do

organismo. Quando uma célula NK encontra a presença dessas proteínas em uma célula sadia, e do MHC-I em si, a célula NK é inibida, e não faz nada. Já em células infectadas, o MHC-I pode apresentar padrões moleculares que não fazem parte do organismo, mas que estão sendo sintetizados no interior da célula, provavelmente pelo agente infeccioso. Ou pior, alguns patógenos podem bloquear a apresentação do MHC-I, na tentativa de continuar ocultos. Mas ao encontrar uma célula que não esteja apresentando o MHC-I, o linfócito NK torna-se ativo, e mata a célula (ABBAS; LICHTMAN, 2003). A Figura 2.2 ilustra este mecanismo.

Além de todas essas características, os leucócitos ainda são capazes de reconhecer os diferentes tipos de patógenos e guiar a resposta imunológica para aquela que seja mais eficiente contra o tipo de agente infeccioso encontrado. Por exemplo, combater uma infecção de helmintos requer ações bem diferentes das usadas no combate a uma infecção viral (FLOREANO; MATTIUSI, 2008).

Por fim, além da função de eliminar os patógenos, os leucócitos da imunidade inata também desempenham um outro papel fundamental na resposta imunológica: a mobilização do SI adaptativo. Se o organismo hospedeiro estiver sofrendo um ataque massivo, e não conseguir eliminar os patógenos rapidamente através dos leucócitos presentes no local, estes são capazes de produzir citocinas que desencadeiam uma resposta inflamatória, que aumentam e focam a atividade do SI no local da infecção, e estimulam os elementos da imunidade adaptativa (FLOREANO; MATTIUSI, 2008).

2.1.1.3 Limites da Imunidade Inata

O SI inato é bastante eficaz na sua tarefa, mas por ser um sistema cujos alvos não podem ser modificados durante a vida do hospedeiro, possui seus limites. O maior deles é lidar com agentes infecciosos que evoluíram e conseguiram modificar suas estruturas, antes reconhecidas como PAMPs, ou ainda, que conseguiram esconder estas estruturas, passando a ser “invisíveis” pela imunidade inata (FLOREANO; MATTIUSI, 2008).

Diante disso, o SI também evoluiu, criando mecanismos que agem complementando a imunidade inata, mas com capacidade de se adaptar ao tipo de agente infeccioso, ao longo da vida do hospedeiro. A esses mecanismos, deu-se o nome de imunidade adaptativa.

2.1.2 Imunidade Adaptativa

As principais propriedades do SI adaptativo, e que o diferenciam do SI inato, são a **especificidade** para antígenos estruturalmente distintos e a **memória** das exposições anteriores (ABBAS; LICHTMAN, 2003).

Essas características são resultado da ação de leucócitos especiais, chamados de **linfócitos**. Os linfócitos são as únicas células que possuem receptores específicos para antígenos, sendo assim as principais células da imunidade adaptativa (ABBAS; LICHTMAN, 2007). Apesar de terem uma aparência semelhante, essas células são bem diferentes em sua linhagem e função. São classificadas quanto ao lugar onde são gerados, e quanto às proteínas de sua superfície, chamadas de CD (do Inglês, *cluster of differentiation*). Os principais linfócitos são o linfócito B e o linfócito T (ABBAS; LICHTMAN, 2007).

Os **linfócitos B** possuem esse nome pois são gerados, a partir de células-tronco, na medula óssea (B vem de *bone marrow*, medula óssea em Inglês). Eles são os únicos linfócitos capazes de produzir moléculas especiais, chamadas de **anticorpos**. Essas moléculas são projetadas de tal forma a reconhecer antígenos de patógenos com alta afinidade, ligando-se a eles, impedindo-os de se ligarem a outras estruturas, de infectarem células, enfim, neutralizando-os e marcando-os como nocivos.

Já os **linfócitos T** possuem esse nome pois sua linhagem final é gerada em um órgão chamado Timo (eles já foram chamados de timócitos). Possuem diversos sub-tipos de células, dentre as quais se destacam as células T CD4⁺, chamadas de **linfócitos T auxiliares** (Th, do Inglês *T helper*), porque ajudam os linfócitos B a produzirem anticorpos e aos fagócitos a ingerir patógenos; e as células T CD8⁺, chamadas de **linfócitos T citolíticos** ou citotóxicos (CTL), pois destroem células infectadas por patógenos intracelulares.

Como dito acima, esses linfócitos possuem receptores PRR especiais. Esses receptores são capazes de reconhecer virtualmente qualquer antígeno, mesmo os que ainda nem existam (FLOREANO; MATTIUSI, 2008). Segundo Floreano e Mattiussi (2008), o TCR, por exemplo, pode ter até 10¹⁸ combinações diferentes. No entanto, é inviável ter um linfócito T em circulação para cada combinação possível. Ao invés disso, o que o SI faz é gerar linfócitos com afinidades aleatórias, e renovar a população de linfócitos de tempos em tempos. Esse processo é chamado de **geração de diversidade** (ABBAS; LICHTMAN, 2007).

No entanto, como os receptores destes linfócitos são gerados aleatoriamente, e têm um potencial de identificação tão amplo, nada os impede de serem auto-reativos, *i.e.*, reativos contra os antígenos do próprio organismo. Por isso, não é viável colocá-los em circulação assim que são gerados. Antes disso, passam por uma espécie de seleção, cujos objetivos são eliminar linfócitos que sejam auto-reativos e aqueles que, por algum motivo em seu processo de maturação, não são capazes de reconhecer nenhum antígeno, *i.e.*, são inúteis.

O processo de se eliminar linfócitos que sejam auto-reativos é chamado de **tolerância imunológica** (ABBAS; LICHTMAN, 2007). Quando esse processo ocorre na geração dos linfócitos, é chamado de **tolerância central** (Figura 2.3 e 2.4). Ele possui duas etapas: a primeira, chamada de **seleção positiva**, testa se o linfócito é capaz de acessar

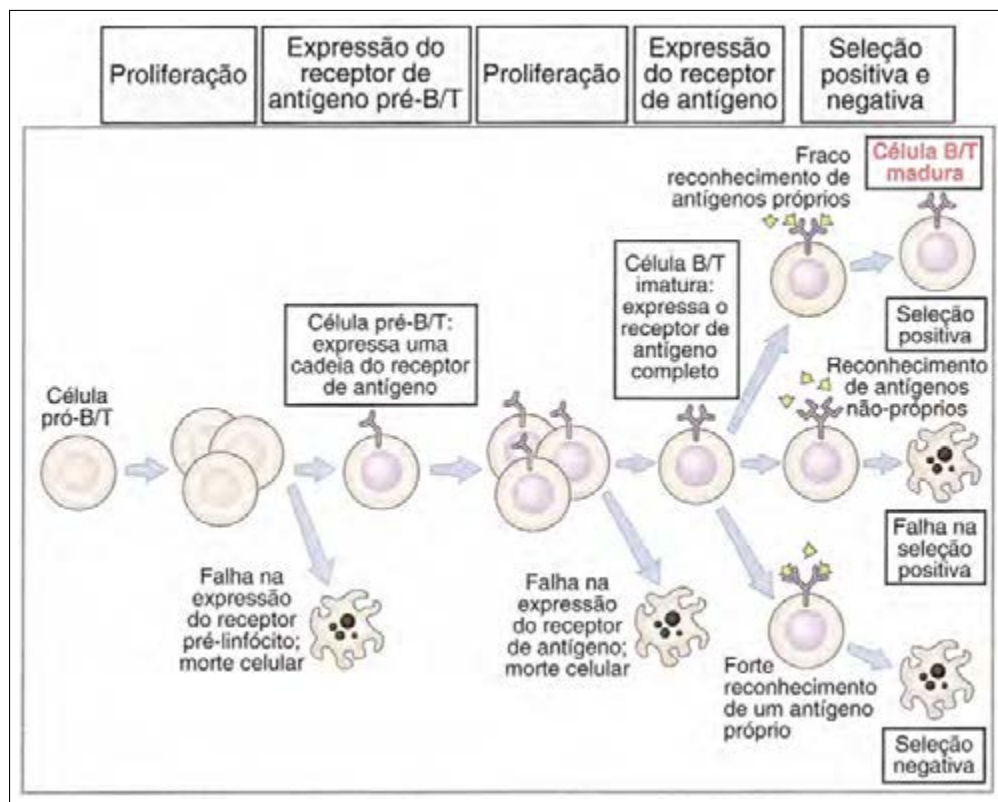


Figura 2.3. Fases da maturação dos linfócitos e a tolerância central.

Fonte: (ABBAS; LICHTMAN, 2007, p. 80)

as moléculas do MHC das células, e caso não seja, ele é eliminado. Já a segunda etapa, chamada de **seleção negativa**, expõe o linfócito a uma molécula de MHC apresentando autoantígenos. Caso ele reconheça esses autoantígenos fortemente, ele é eliminado. Caso ele reconheça fracamente, ele passa pelo teste: o SI baseia-se no fato de que, se o linfócito reconhece autoantígenos fracamente, possivelmente reconhecerá outros antígenos fortemente (FLOREANO; MATTIUSI, 2008). Os linfócitos B passam apenas pela seleção negativa, e mesmo quando é reconhecido como auto-reativo, é capaz de mudar sua especificidade, através de um mecanismo denominado editoramento de especificidade (FLOREANO; MATTIUSI, 2008).

Todos esses processos acontecem nos órgãos de geração dos linfócitos, chamados de **órgãos linfoides geradores**. Como já dito, o timo é o órgão linfoide gerador dos linfócitos T, e a medula óssea é o órgão linfoide gerador dos linfócitos B. Após serem gerados, receberem suas especificidades e passarem pela tolerância central, os linfócitos recém-criados migram (Figura 2.5) para os **órgãos linfoides periféricos**, como os linfonodos, o baço, as amígdalas, entre outras estruturas (ABBAS; LICHTMAN, 2007).

Os linfócitos que chegaram aos órgão linfoides periféricos já estão maduros, mas ainda não ativados. Isso porque eles ainda podem ser auto-reativos a autoantígenos que não estão presentes nos órgãos linfoides geradores, mas apenas em alguns locais muito es-

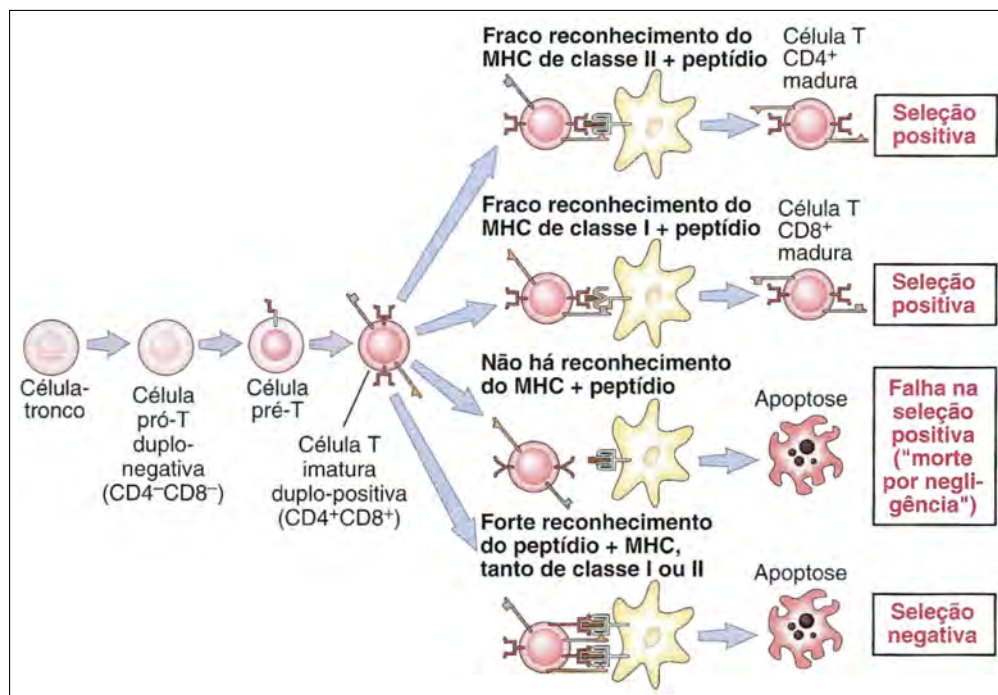


Figura 2.4. Detalhe do amadurecimento e tolerância central dos linfócitos T.

Fonte: (ABBAS; LICHTMAN, 2007, p. 87)

pecíficos do corpo, e por isso conseguiram passar pela tolerância central (FLOREANO; MATTIUSSI, 2008). Para lidar com esse problema, o SI possui uma “política” de ativação dos linfócitos, que se baseia no seguinte: todo linfócito, para ser ativado e iniciar sua resposta, precisa de dois sinais distintos, *i.e.*, dois estímulos (Figura 2.6). O primeiro sinal (sinal 1) é o encontro com o antígeno para o qual o linfócito é específico. O segundo sinal (sinal 2) consiste na presença de certas substâncias que, segundo o SI, podem indicar que o local de encontro com o antígeno é perigoso, e portanto, o antígeno provavelmente também é.

O sinal 1 garante que o linfócito só irá reagir contra antígenos para o qual foi projetado. Já o sinal 2 é conhecido como *danger signal* (FLOREANO; MATTIUSSI, 2008). É um sinal químico, que pode ser oriundo dos próprios patógenos (como PAMPs), ou de sua atividade: ao danificarem tecidos, as células atingidas podem liberar substâncias (sinais de estresse) consideradas sinais 2, ou ainda podem ser lisadas, expondo o conteúdo do seu citoplasma, o que também é considerado como sinal 2. Mais ainda, substâncias provenientes da imunidade inata também podem ser consideradas como sinais 2: uma das atribuições do SI inato é, durante sua resposta, marcar as regiões onde encontrou os patógenos com substâncias que são reconhecidas como sinais 2 (ABBAS; LICHTMAN, 2007).

Todas as regiões onde há a presença de substâncias consideradas sinais 2 são chamadas de *danger zones* (FLOREANO; MATTIUSSI, 2008). A necessidade da presença do sinal

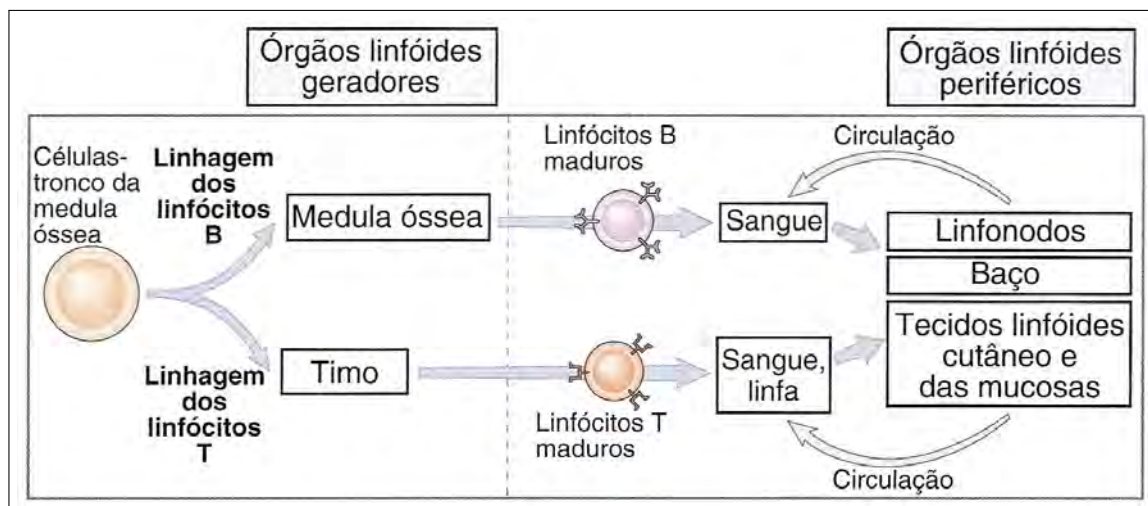


Figura 2.5. Ciclo de desenvolvimento dos linfócitos: órgão linfóides geradores e periféricos.

Fonte: (ABBAS; LICHTMAN, 2007, p. 13)

2 denota a ideia de ativação baseada em contexto do SI, evitando com que as respostas do SI adaptativo sejam desencadeadas por antígenos que, apesar de terem sido reconhecidos, provavelmente são inofensivos, por terem sido encontrados em áreas onde não havia danos ao tecido, ou inflamação (FLOREANO; MATTIUSI, 2008).

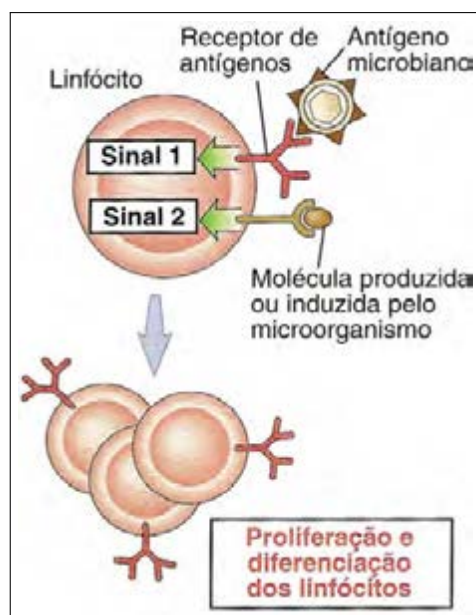


Figura 2.6. A necessidade de dois sinais para ativação dos linfócitos.

Fonte: (ABBAS; LICHTMAN, 2007, p. 10)

Além disso, se um linfócito encontra com um antígeno, mas sem a presença do sinal 2, esse linfócito é desativado ou eliminado. Isso porque o SI entende que se o linfócito é específico para um antígeno que foi encontrado em um lugar onde não havia sinais de perigo, lesão ou inflamação, então esse linfócito provavelmente é auto-reativo. Esse

mecanismo é chamado de **tolerância periférica**. Essa tolerância é crucial para o SI adaptativo. No entanto, pode acontecer de um autoantígeno ser encontrado, por acidente, em uma *danger zone*, e passar a ser considerado perigoso. Contudo, é muito mais provável que este autoantígeno seja encontrado em muitas outras regiões que não estão marcadas como sendo perigosas. Isso faz com que a probabilidade de um linfócito auto-reativo ser eliminado seja bem maior do que a probabilidade dele ser ativado. Esse mecanismo, que faz parte da tolerância periférica, é chamado de **tolerância baseada em frequência**. Por fim, os linfócitos que não encontrarem um antígeno para o qual têm afinidade por um certo período de tempo, também são eliminados, justamente para darem lugar a outros linfócitos, com especificidades diferentes (FLOREANO; MATTIUSI, 2008).

Mas encontrar antígenos não é uma tarefa fácil. O fato dos linfócitos ficarem concentrados nos órgãos linfoides periféricos diminui a chance deles encontrarem com um antígeno de um patógeno que eles reconheçam e que esteja em algum lugar do corpo. Para solucionar esse problema, o SI desenvolveu uma abordagem que consiste em concentrar, além dos linfócitos, também os antígenos nos órgãos linfoides periféricos. Para alcançar esse objetivo, o SI possui dois mecanismos: o primeiro consiste em drenar constantemente o líquido dos tecidos epiteliais, conjuntivos e parenquimatosos, chamado de **linfa**, por vasos especiais, chamados de **vasos linfáticos**, até os linfonodos. Conforme a linfa passa pelos linfonodos, ela traz consigo uma amostra de todos os antígenos que estão presentes nos arredores. Já o baço desempenha o mesmo papel que o linfonodo, mas com o objetivo de “filtrar” os patógenos que estão em circulação no sangue (ABBAS; LICHTMAN, 2007).

O segundo mecanismo consiste na utilização de células especiais, capazes de capturar antígenos em qualquer lugar do organismo, mesmo os mais distantes, e levá-los até os linfonodos, onde essas células são capazes de apresentá-los aos linfócitos (Figura 2.7). Essas células são chamadas de **células apresentadores de antígenos** profissionais (APC, do Inglês *antigen-presenting cell*) (ABBAS; LICHTMAN, 2007).

As APCs têm o papel de patrulhar o corpo capturando antígenos, levá-los aos linfonodos, e apresentá-los aos linfócitos. Para tanto, elas são equipadas com moléculas especiais em suas superfícies, capazes de estabelecer um canal de comunicação com os linfócitos, e informá-los sobre os antígenos que elas fagocitaram. Essas moléculas são o **complexo de histocompatibilidade principal classe II** (MHC-II, do Inglês *class II major histocompatibility complex*). Toda célula APC possui, além do MHC-I, esse tipo especial, o MHC-II, que é capaz de apresentar os antígenos aos linfócitos. Além disso, ao capturar os antígenos, as células APC também “sentem” o ambiente verificando a presença de sinais. 2. Caso eles existam, ela também irá informar isto aos linfócitos na hora de apresentar os antígenos.

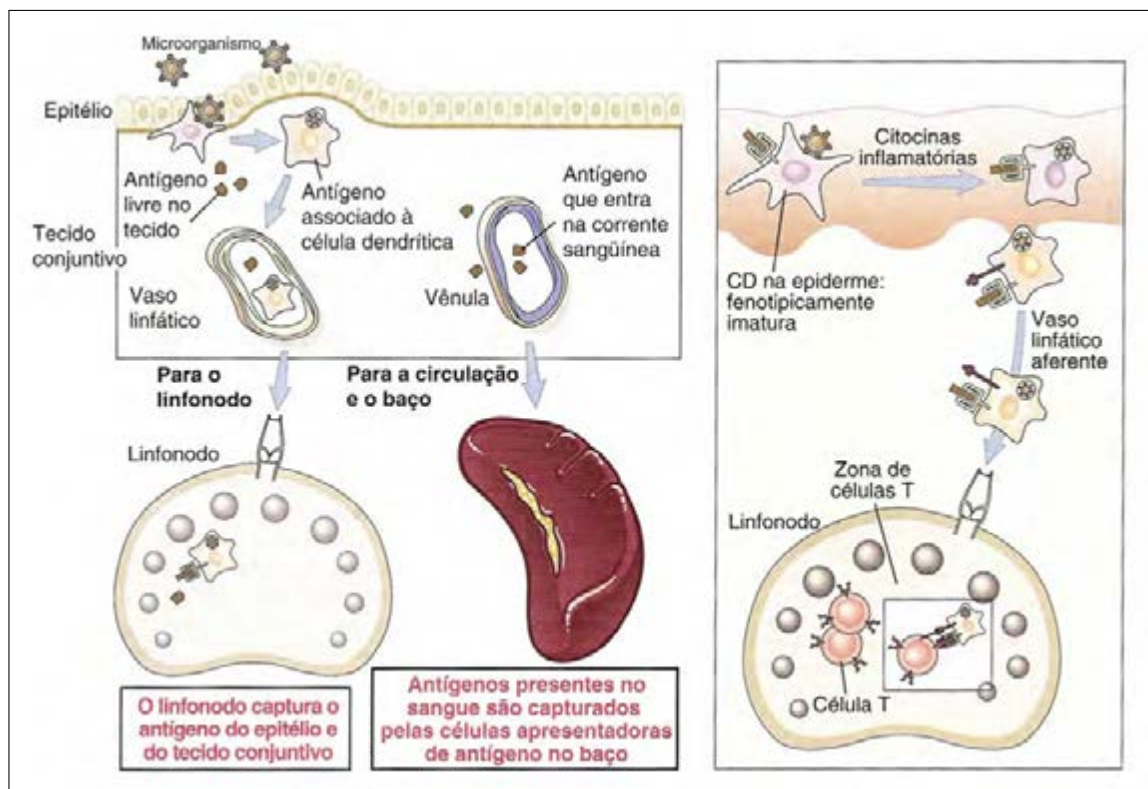


Figura 2.7. Captura e apresentação de antígenos.

Fonte (adaptado): (ABBAS; LICHTMAN, 2007, p. 48, 50)

Dentre as células APC, as de maior importância são as chamadas **células dendríticas** (DC, do Inglês *dendritic cells*) (ABBAS; LICHTMAN, 2007). Essas células possuem diversos PRRs capazes de reconhecerem os diferentes tipos de *danger signals*, como os gerados por tecidos lesados ou por PAMPs. Elas normalmente estão desativadas, e permanecem patrulhando o organismo em busca de antígenos (FOLCIK *et al.*, 2007). Quando ela identifica a presença de sinais 2, ela se torna ativada, e aumenta sua taxa de captura de antígenos, e à medida que captura mais antígenos, vai tornando-se mais inclinada a migrar para um linfonodo das redondezas. Quando o faz, leva consigo os antígenos e a informação sobre se eles foram capturados ou não em uma *danger zone* (ABBAS; LICHTMAN, 2007).

Além das DCs, os macrófagos também são células APCs (ABBAS; LICHTMAN, 2007), e por isso, são capazes de apresentar os antígenos dos patógenos que fagocitou aos linfócitos. Assim também os linfócitos B são capazes de apresentar antígenos aos linfócitos T (ABBAS; LICHTMAN, 2007).

Uma vez que os linfócitos reconheçam os antígenos apresentados, em conjunto com os co-estimuladores (sinais 2), eles se tornam ativados. Diferentes sub-tipos de linfócitos são ativados, dependendo de diferentes fatores, como os tipos de *danger signals* que a APC detectou quando foi ativada, qual tipo de APC que está ativando o linfócito e sinais

adicionais produzidos pelo tecido que está sendo atacado (FLOREANO; MATTIUSI, 2008).

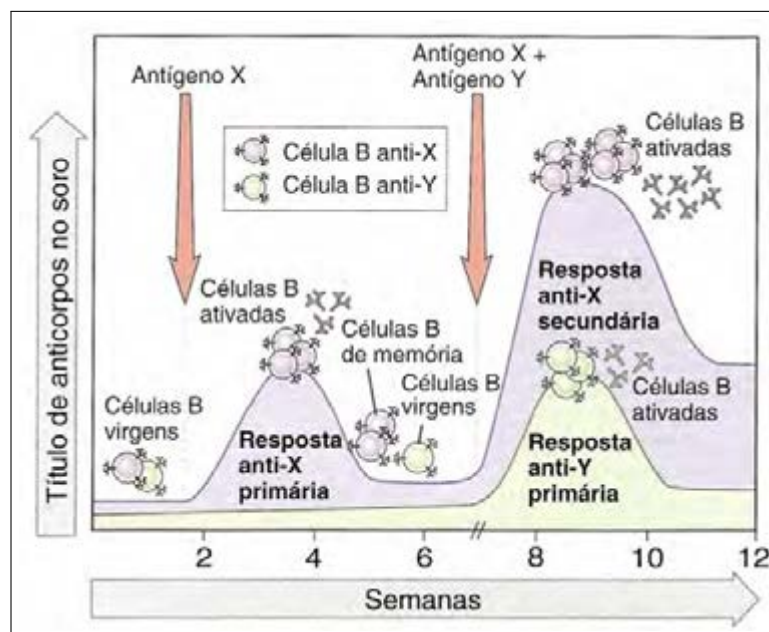


Figura 2.8. Especificidade e memória das respostas imunológicas adaptativas.

Fonte: (ABBAS; LICHTMAN, 2007, p. 7)

Já ativados, os linfócitos começam a proliferar, resultando em um aumento muito grande no número de linfócitos específicos para o antígeno que desencadeou a reação. Esse processo é chamado de **expansão clonal**. Para se ter uma ideia da amplitude desse processo, os autores Abbas e Lichtman (2007) destacam que o número de um linfócito T CTL específico para um determinado antígeno no organismo saudável é de aproximadamente 1 para 10^6 . Após o desencadeamento da resposta imunológica para uma infecção viral, por exemplo, até 20% de todos os linfócitos CTL do organismo serão linfócitos específicos para o vírus causador da infecção, o que significa que os clones específicos para o patógeno em questão aumentaram sua população em mais de 10000 vezes (ABBAS; LICHTMAN, 2007).

Os linfócitos ativados, depois da expansão clonal, são distribuídos pelo organismo, e são atraídos principalmente pelas áreas onde está ocorrendo a inflamação (quimiotaxia) (ABBAS; LICHTMAN, 2007). Uma vez no sítio de infecção, eles atuam eliminando os (ou estimulando a eliminação dos) patógenos e outras estruturas que apresentarem os antígenos que desencadearam a resposta imunológica.

À medida que os patógenos vão sendo eliminados, os sinais 2 vão diminuindo, as APCs vão apresentando menos antígenos, menos linfócitos vão sendo ativados, e os que já foram ativados, caso atinjam um limite de eliminações, ou caso fiquem muito tempo sem serem “re-ativados”, sofrem apoptose, e vão sendo eliminados. Dessa forma, de 1 a 2 semanas

após a infecção ter sido erradicada, toda a resposta imunológica já regrediu, e não há mais sinais de que tudo isso ocorreu, a não ser por um grupo especiais de células: os **linfócitos de memória** (ABBAS; LICHTMAN, 2007).

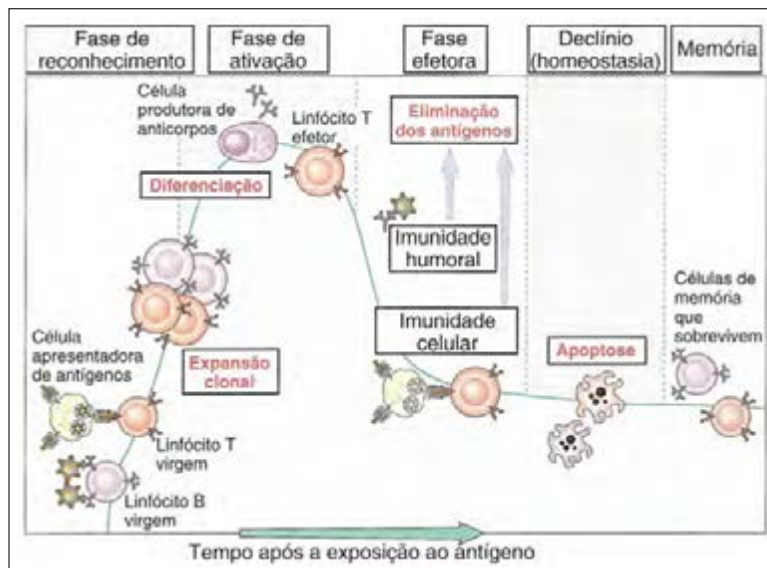


Figura 2.9. Fases das respostas imunológicas adaptativas.

Fonte: (ABBAS; LICHTMAN, 2007, p. 9)

Os linfócitos de memória podem sobreviver por anos e conferem ao SI adaptativo a chamada **memória imunológica**, de forma que ele possa desenvolver respostas imunológicas mais rápidas, mais acentuadas e mais eficazes à infecções recorrentes de agentes infecciosos já combatidos anteriormente (Figura 2.8). Cada encontro com o antígeno gera mais linfócitos de memória, o que fará com que a próxima reação seja ainda maior. A memória imunológica é um dos mecanismos dos quais se valem as vacinas para conferir proteção contra as doenças infecciosas (ABBAS; LICHTMAN, 2007).

Todos esses mecanismos ocorrem de forma sincronizada e distribuída em relação aos antígenos que desencadearam a reação. Essa sequência de eventos (veja Figura 2.9) é bem definida no SI adaptativo, e segundo Abbas e Lichtman (2007) consistem em:

1. reconhecimento do antígeno;
2. ativação dos linfócitos;
3. eliminação do antígeno;
4. declínio;
5. memória.

Assim como na imunidade inata, as respostas do SI adaptativo também são classificadas quanto ao tipo de mediador que utilizam: se os patógenos são eliminados diretamente

por alguma célula do SI, denomina-se o processo como imunidade adaptativa celular. Por outro lado, se a eliminação do patógeno se dá por meio de moléculas secretadas pelas células do SI, então denomina-se imunidade adaptativa humoral.

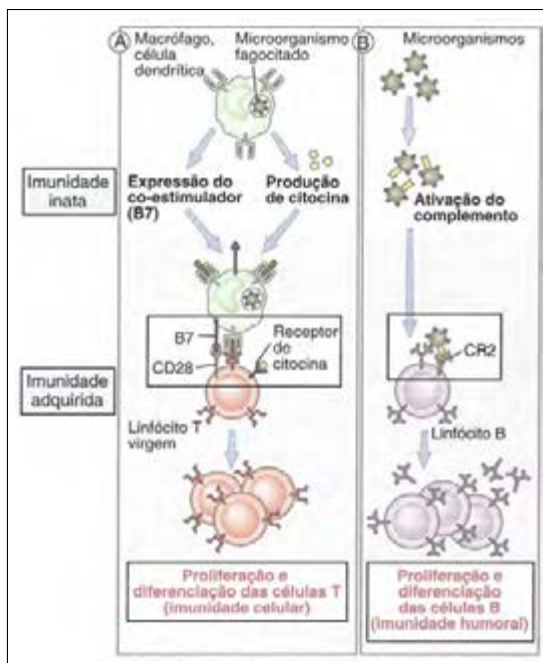


Figura 2.10. Cooperação entre as imunidades do SI.

Fonte: (ABBAS; LICHTMAN, 2007, p. 41)

Esses dois tipos de imunidade se comunicam com os seus processos semelhantes no SI inato, evidenciando a colaboração entre os sistemas. Dessa forma, tanto o SI inato estimula o SI adaptativo, quanto o SI adaptativo provê estímulos para o SI inato. Essa colaboração é ilustrada na Figura 2.10.

A seguir serão ilustradas em mais detalhes as características dos dois tipos de imunidades adaptativas conferidas durante a fase efetora: a imunidade humoral e a imunidade celular.

2.1.2.1 Imunidade Adaptativa Humoral

A imunidade adaptativa humoral é o braço do SI adaptativo responsável por combater as infecções causadas por patógenos extracelulares. Também é sua função eliminar as toxinas que os patógenos possam produzir (ABBAS; LICHTMAN, 2007).

A principal célula envolvida neste tipo de imunidade é o linfócito B. Como dito anteriormente, o linfócito B é a única célula capaz de produzir anticorpos. Segundo Abbas e Lichtman (2007), este tipo de imunidade, além de combater patógenos extracelulares, possui grande importância também por conseguir combater agentes infecciosos que não são reconhecidos pela imunidade celular.

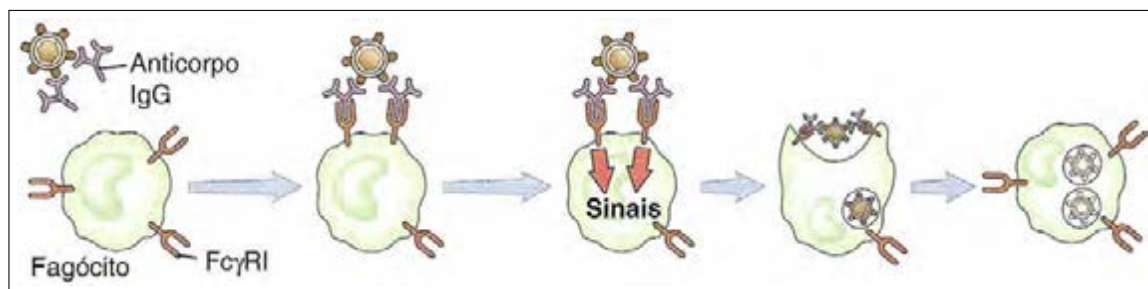


Figura 2.11. Opsonização mediada por anticorpos.

Fonte: (ABBAS; LICHTMAN, 2007, p. 162)

Quando um patógeno possui cápsulas de polissacarídeos, por exemplo, e ocultam suas PAMPs, eles conseguem escapar dos mecanismos da imunidade celular inata e adaptativa. Por exemplo, os macrófagos não conseguem fagocitar esse tipo de patógeno, e os receptores dos linfócitos T não reconhecem esse tipo de antígeno. Contudo, os linfócitos B são capazes de produzir anticorpos que reconhecem, além dos proteicos (reconhecidos pelos linfócitos T), os antígenos polissacarídeos e lipídicos. Ao serem detectados pela imunidade humoral do SI adaptativo, os linfócitos B produzem anticorpos específicos para o agente infeccioso em questão. Esses anticorpos percorrem todo corpo através da circulação, e quando encontram o patógeno, se ligam em sua superfície, impedindo-o que se ligue a outras estruturas, *i.e.*, neutralizando-o, e marcando-o para fagocitose, em um processo chamado de opsonização (ver Figura 2.11). Note que os anticorpos têm um papel semelhante ao do sistema complemento nesse ponto.

Por este motivo, os anticorpos são considerados o principal mecanismo efetor da imunidade adaptativa humoral. Veja na Figura 2.12 as principais funções efetoras dos anticorpos.

Como já mencionado, os anticorpos são produzidos pelos linfócitos B, e eles têm a mesma especificidade que elas. Assim, os linfócitos B só produzem anticorpos quando têm contato com um antígeno para o qual são específicas. Uma célula B ativada pode produzir até 4 mil células produtoras de anticorpos (plasmócitos), e estas podem produzir até 10^{12} partículas de anticorpos por dia. (ABBAS; LICHTMAN, 2007).

Diferentemente dos linfócitos T, estas células não precisam que os antígenos sejam apresentados a elas por APCs: elas possuem receptores especiais (BCR), que nada mais são do que anticorpos ligados à membrana do linfócito B, que são capazes de reconhecer os antígenos. Após reconhecerem o antígeno, o linfócito B se torna ativado. No entanto, dependendo da natureza do antígeno, só o seu reconhecimento não é o suficiente para que esse linfócito comece a produzir anticorpos: em alguns casos, os linfócitos B ainda necessitam de co-estimuladores oriundos dos linfócitos T ativados. Por este motivo, diz-se que os linfócitos B possuem dois tipos de ativação: dependentes de T ou independentes

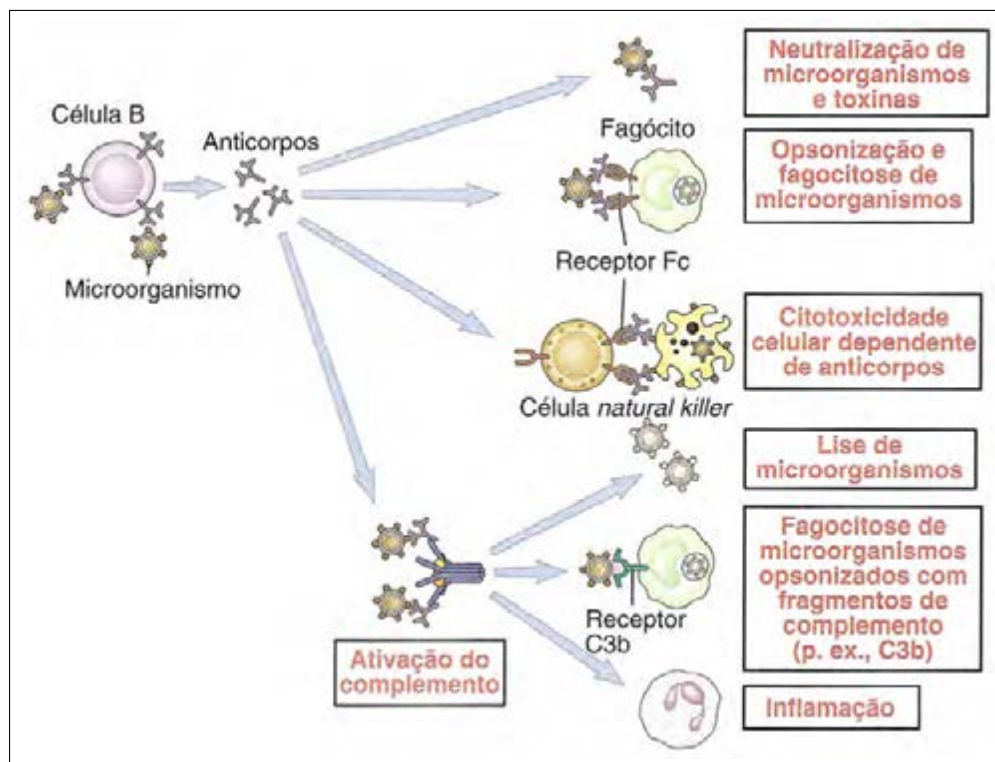


Figura 2.12. Funções efetoras dos anticorpos.

Fonte: (ABBAS; LICHTMAN, 2007, p. 159)

de T. A Figura 2.13 ilustra o processo de ativação do linfócito B.

Na ativação dependente de T, os antígenos normalmente são proteínas. Nestes casos, as APCs capturam os antígenos, apresenta-os aos linfócitos T que, caso ativados, vão de encontro aos linfócitos B, ligando-se a eles e os co-estimulando, para que comecem a produzir anticorpos. Essa interação (como todas as descritas na imunidade adaptativa) é antígeno-específica, ou seja, o linfócito T só ativa os linfócitos B que também são específicos para o antígeno em questão.

Já na ativação independente de T, os antígenos normalmente não são reconhecidos pelos linfócitos T. Eles são capturados pelos linfócitos B no linfonodo e fagocitados. Já no seu interior, eles são processados. A presença de moléculas de complemento aderidas ao antígeno fagocitado servem de sinal 2 para os linfócitos B (ABBAS; LICHTMAN, 2007). Os linfócitos B também são APCs, e possuem MHC-II para apresentação de antígenos. Assim, quando encontram os linfócitos T, são capazes de apresentar os antígenos capturados a estes. Os linfócitos T não irão reconhecer a mesma parte do antígeno que os linfócitos B reconheceram, mas podem reconhecer outras partes, que ficaram expostas depois do processamento. Caso isso ocorra, os linfócitos T são ativados.

Uma vez ativados, os linfócitos B diferenciam-se em células secretoras de anticorpos (plasmócitos) e podem migrar para os locais de infecção, ou permanecer no linfonodo (linfócito B germinativo). Os anticorpos produzidos são levados pela circulação, e circulam

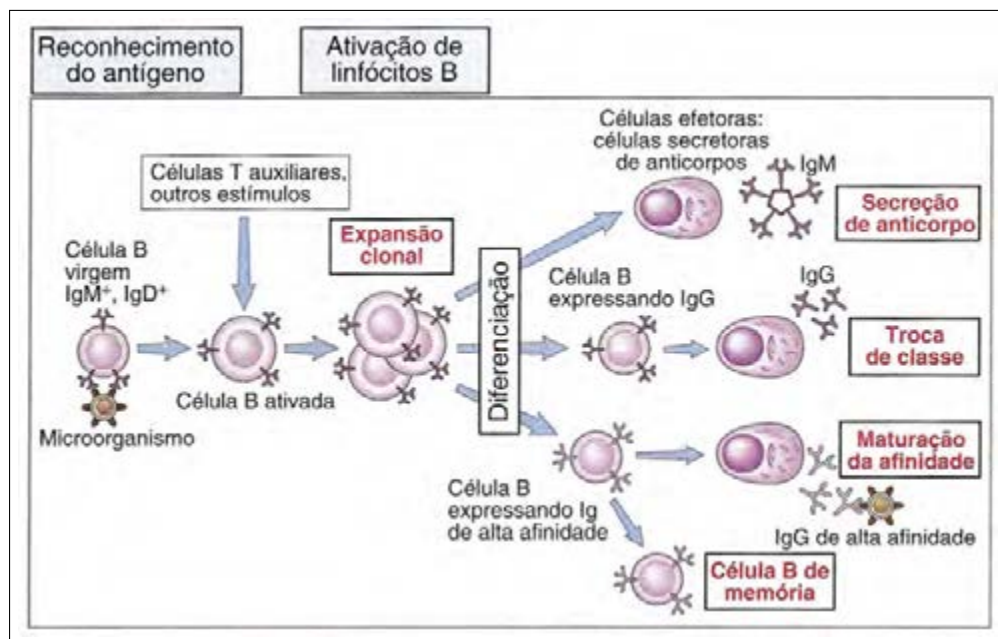


Figura 2.13. Ativação dos linfócitos B.

Fonte: (ABBAS; LICHTMAN, 2007, p. 136)

por todo o organismo.

À medida que os patógenos vão sendo neutralizados, os estímulos antigênicos vão cessando, e a resposta entra em declínio. Outro mecanismo utilizado pela célula B para o declínio da atividade imunológica é o reconhecimento dos chamados imunocomplexos, que são antígenos ligados a anticorpos. Quando uma célula B detecta um antígeno já ligado a um anticorpo, é estimulada de forma negativa, a não mais produzir anticorpos. Esse processo é chamado de *feedback* de anticorpo, e sua função é indicar à imunidade humoral quando uma quantidade suficiente de anticorpos já foi produzida (ABBAS; LICHTMAN, 2007).

Outras células, chamadas de células secretoras de anticorpos de longa duração, migram para a medula óssea onde podem viver muitos anos, e continuam a secretar pequenas quantidades de anticorpos. Caso haja necessidade (outra infecção), o patógeno já irá se encontrar com anticorpos prontos para neutralizá-lo. Já os linfócitos B de memória sobrevivem ao declínio e podem viver também por muitos anos. Estes não secretam anticorpos, mas ficam esperando por um novo contato com o antígeno para o qual são específicos e caso encontrem, se diferenciam rapidamente em células secretoras de anticorpos. Uns dos objetivos da vacinação é estimular a geração destes dois tipos de células para os patógenos da vacina (ABBAS; LICHTMAN, 2007).

2.1.2.2 Imunidade Adaptativa Celular

Segundo Abbas e Lichtman (2007), a imunidade adaptativa celular é o braço do SI adaptativo responsável por combater as infecções causadas por patógenos intracelulares. Os patógenos, por sua vez, podem se tornar intracelulares de duas formas distintas: primeiro, sendo propositalmente fagocitados por macrófagos, por exemplo, mas conseguindo resistir aos mecanismos de eliminação no seu interior, podendo até escapar para dentro do seu citoplasma; e, em segundo lugar, alguns patógenos conseguem aderir à membrana de algumas células e então infectá-las (ABBAS; LICHTMAN, 2007).

A principal célula responsável pela imunidade adaptativa celular é o linfócito T (ABBAS; LICHTMAN, 2007). Como visto, a ativação deste linfócito possui fases distintas, que começa com o reconhecimento do antígeno apresentado pelas APCs, sua subsequente ativação, ação contra os patógenos e eliminação, como parte do declínio da resposta imunológica. A Figura 2.14 ilustra esse processo.

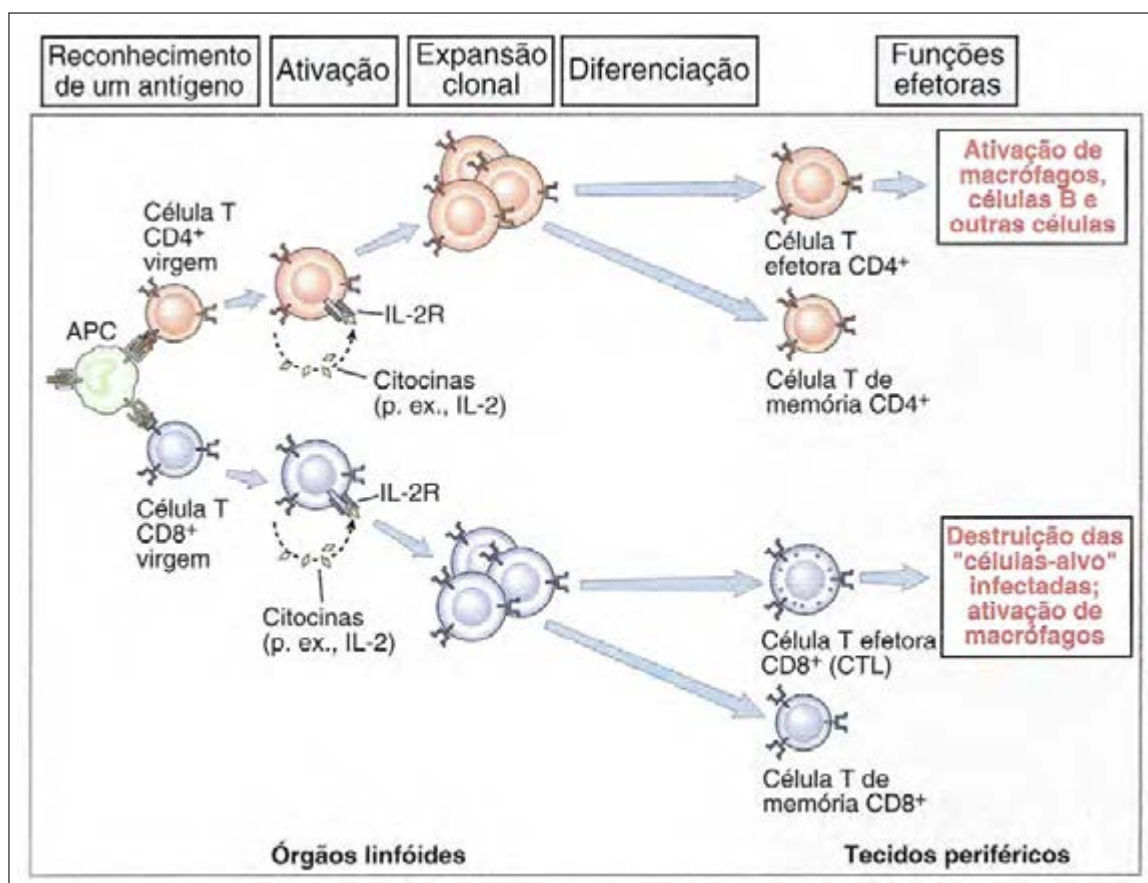


Figura 2.14. Ativação dos linfócitos T.

Fonte: (ABBAS; LICHTMAN, 2007, p. 93)

Na fase de reconhecimento, uma célula APC que migrou para o linfonodo se liga ao linfócito T apresentando os antígenos que ela capturou. Como dito anteriormente, a célula APC precisa estar ativada para que consiga estimular a ativação dos linfócitos T.

Os receptores das células Th reconhecem antígenos apresentados pelo MHC-II, enquanto os receptores das células CTL reconhecem apenas antígenos apresentados pelo MHC-I. As APCs podem apresentar antígenos tanto através do MHC-II quanto pelo MHC-I (no caso de antígenos presentes no citoplasma da mesma) (ABBAS; LICHTMAN, 2007). Mas como a APC é capaz de capturar antígenos de patógenos intracelulares? Segundo Abbas e Lichtman (2007), as APCs são capazes de fagocitar as células infectadas por inteiro, e processá-las, extraindo as informações sobre os patógenos que estavam presentes nelas, e então, apresentar os antígenos.

Após serem ativados, os linfócitos começam a expansão clonal, aumentando muito o número de células específicas para o antígeno que desencadeou a resposta. Durante esse processo, as células também sofrem diferenciação. Nesse processo, os linfócitos, antes células de reconhecimento de antígeno, se transformam em células efetoras, e ativam seus mecanismos de combate (Figura 2.15). Apesar de alguns permanecerem no linfonodo, os linfócitos ativados tendem a migrar para o local da infecção. Os linfócitos T e CTL possuem mecanismos efetores bem distintos, que serão discutidos a seguir.

A principal função dos linfócitos Th é estimular as demais células, e guiar o avanço da resposta imunológica. Por este motivo, seu principal mecanismo efetor é a expressão aumentada de algumas moléculas sinalizadoras em sua superfície, e a liberação de citocinas. Esses linfócitos atuam principalmente estimulando os macrófagos (a produzirem mais substâncias microbidas para que ele destrua os patógenos que fagocitou) e as APCs (fornecendo um “*feedback*” que prolonga a vida da APC, afinal ela está sendo útil). Eles também estimulam os linfócitos B (estimulando a produção de anticorpos), como foi visto anteriormente.

Já os linfócitos T CD8⁺ ativados se diferenciam em linfócitos T CTL, que produzem substâncias capazes de matar outras células. Esses linfócitos migram para os locais de infecção, onde são capazes de reconhecer antígenos apresentados pelas células do tecido, através do MHC-I. Caso as células estejam apresentando antígenos para os quais o CTL têm afinidade, então ele libera suas substâncias, eliminando a célula infectada.

É interessante notar que alguns patógenos desenvolveram estratégias para, após infectar uma célula, inibir a apresentação de antígenos através do MHC-I. Mas como discutido anteriormente, as células NK reconhecem células que não estão expressando o complexo MHC-I, e as destroem.

Uma parte dos linfócitos ativados se diferenciam em células T de memória. Essas células sobrevivem à erradicação da infecção e ao declínio da resposta imunológica, não sofrendo apoptose como outras células, tendo um período de vida bastante longo, podendo chegar a muitos anos. Elas podem ser encontradas nos tecidos linfóides, nas mucosas e na circulação (ABBAS; LICHTMAN, 2007), no entanto, não produzem mais citocinas,

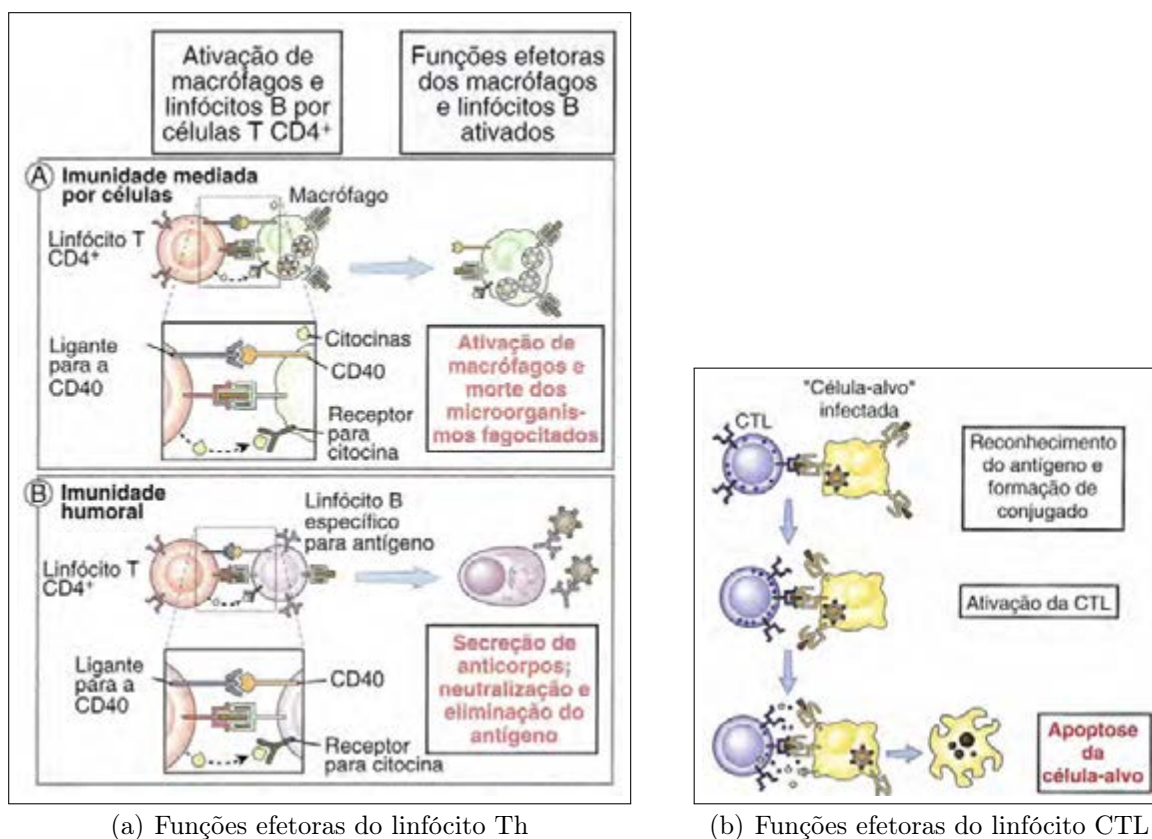


Figura 2.15. Funções efetoras dos linfócitos.

Fonte: (ABBAS; LICHTMAN, 2007, p. 105, 129)

nem destroem células infectadas. Um primeiro grupo de células, chamadas de células de memória central, permanecem nos órgãos linfóides e podem desencadear uma expansão clonal rapidamente caso encontrem com o antígeno que as originou. Outro grupo, chamada de células de memória efetoras, localizam-se nas mucosas, e podem reagir rapidamente contra antígenos que reaparecerem nestes locais (ABBAS; LICHTMAN, 2007).

2.1.2.3 Autoimunidade

Como foi mostrado, o SI saudável é capaz de reagir a uma quantidade infindável de antígenos. Contudo, é capaz de reconhecer os antígenos que são parte dos tecidos do seu próprio organismo, os antígenos próprios ou autoantígenos, e não reagir contra eles. Como discutido anteriormente, essa característica do SI é chamada de tolerância imunológica (ABBAS; LICHTMAN, 2007).

Como dito, o SI alcança essa propriedade através de mecanismos que eliminam linfócitos auto-reativos. Quando esses mecanismos atuam nos órgãos linfóides geradores, são chamados de tolerância central. Já quando esses mecanismos atuam nos linfócitos já em circulação, são chamados de tolerância periférica (ABBAS; LICHTMAN, 2007).

Apesar desses mecanismos serem bastante eficazes, diversos motivos podem fazer com que eles falhem, como problemas genéticos ou alguns tipos de infecções, permitindo que linfócitos auto-reativos sejam ativados. À existência e ativação de linfócitos auto-reativos deu-se o nome de **autoimunidade**. Segundo Abbas e Lichtman (2007), “a autoimunidade é definida como uma resposta imune contra antígenos próprios, e é uma causa importante de doenças” (ABBAS; LICHTMAN, 2007, p. 179).

Estima-se que cerca de 2% da população sofra com algum tipo de doença autoimune, ou seja, causada pela autoimunidade (ABBAS; LICHTMAN, 2007). Mais ainda, muitas doenças sem causa conhecida podem ser doenças autoimunes. Veja o caso descrito por Sawcer *et al.* (2011), cujo estudo da esclerose múltipla levou os autores a afirmarem que os genes encontrados durante o processo de mapeamento genético desta doença demonstraram claras evidências de estarem associados a processos imunológicos.

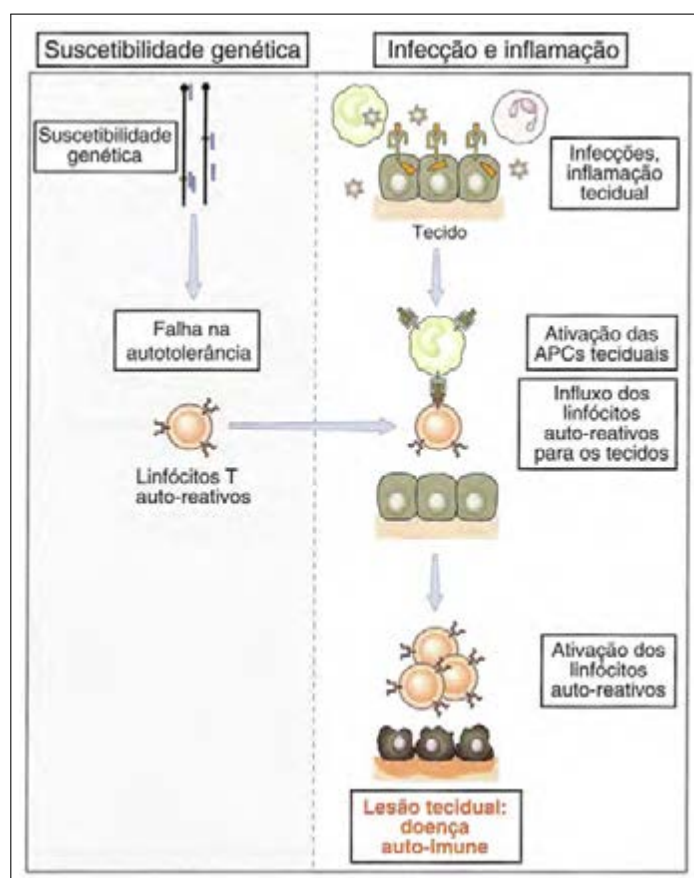


Figura 2.16. Mecanismos postulados da autoimunidade.

Fonte: (ABBAS; LICHTMAN, 2007, p. 181)

Isto demonstra que o conhecimento, principalmente a respeito da autoimunidade, ainda não está consolidado. Muito já se aprendeu sobre a autoimunidade, mas segundo Abbas e Lichtman (2007), nenhuma doença autoimune humana teve sua etiologia descrita ainda. Existem, contudo, diversas hipóteses que tentam explicar como essas doenças ocorrem.

Uma das hipóteses mais aceita é que fatores genéticos são responsáveis pela autoimunidade (Figura 2.16). Os autores de Abbas e Lichtman (2007) afirmam que dentre os múltiplos genes já estudados que demonstram ter alguma relação com o desenvolvimento de doenças autoimunes, destacam-se os genes do MHC. Segundo Abbas e Lichtman (2007), defeitos genéticos em “*alelos particulares do MHC podem contribuir para o desenvolvimento da autoimunidade, pois eles são ineficientes na apresentação de antígenos próprios, levando à seleção negativa defeituosa das células T*” (ABBAS; LICHTMAN, 2007, p. 191).

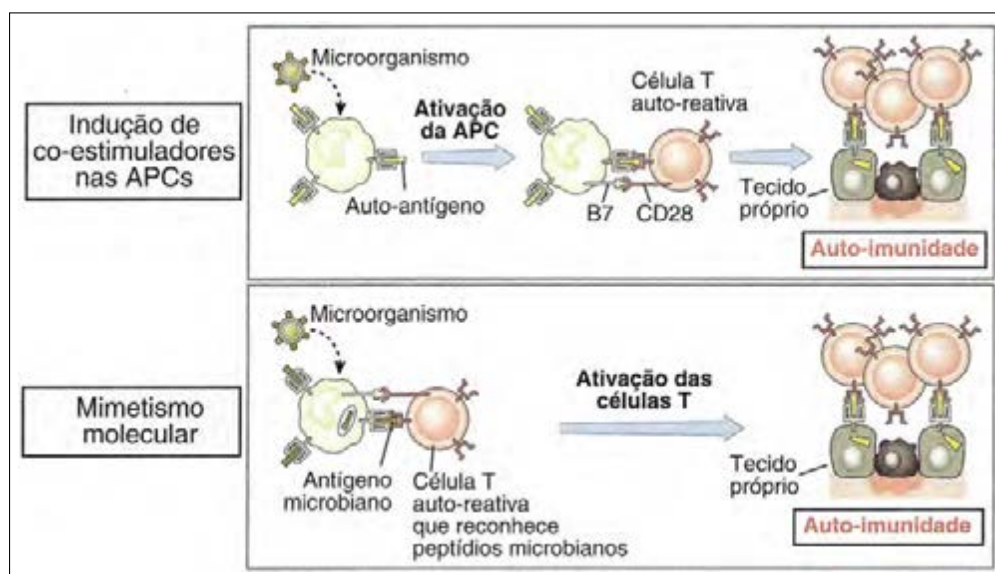


Figura 2.17. Mecanismos pelos quais os patógenos podem promover a autoimunidade.

Fonte: (ABBAS; LICHTMAN, 2007, p. 192)

Além disso, os linfócitos auto-reativos ainda precisariam ser ativados para efetivamente causarem algum dano. Sobre esta ativação, a principal hipótese diz que caberia a uma infecção o papel de burlar a tolerância periférica e ativar esses linfócitos auto-reativos, que iriam então atacar tecidos do próprio corpo, causando a doença autoimune. Para que essas infecções ativem os linfócitos auto-reativos, ainda existem duas hipóteses: a primeira diz que os patógenos deveriam, por si próprios, produzir substâncias que seriam identificadas como sinais 2 que fariam com que células APC, carregando autoantígenos, fossem ativadas, e conseguissem ativar os linfócitos auto-reativos. Já uma segunda hipótese, chamada de hipótese do “mimetismo molecular” (ABBAS; LICHTMAN, 2007; BENOIST; MATHIS, 2001), afirma que existem patógenos com antígenos muito semelhantes à antígenos do organismos. Quando uma resposta a esse antígeno é iniciada, alguns linfócitos, específicos para o patógeno podem acabar atacando tecidos do próprio organismo, por consequência da similaridade entre o antígeno do patógeno e um antígeno próprio. Essas hipóteses estão ilustradas na Figura 2.17.

É com base nesta demanda por maiores contribuições, e na pré-existência de hipóteses

a serem corroboradas, que o tema autoimunidade será escolhido como alvo para os testes da ferramenta proposta por este trabalho.

Descritos estes aspectos do SI, relevantes ao contexto do trabalho, serão agora discutidos os métodos, descritos na literatura, que poderão ser utilizados para desenvolver um modelo do SI que seja capaz de simular os mecanismos necessários para o estudo da autoimunidade.

2.2 Modelando o Sistema Imunológico

Uma vez tendo ilustrado a amplitude do SI e de seus componentes, esta seção destina-se a discutir qual a melhor maneira de se desenvolver modelos do SI.

Segundo Li *et al.* (2009), modelar o SI, ou mesmo parte dele, não é uma tarefa fácil: ele possui uma grande quantidade de iterações não-lineares entre suas células e tem habilidades como emergência de comportamento, co-evolução, agregação, variedade, e auto-organização em um ambiente dinâmico (LI *et al.*, 2009). Por ter estas características, o sistema imunológico é classificado como um Sistema Adaptativo Complexo (CAS, do Inglês *Complex Adaptive System*) (MACAL; NORTH, 2009; LI *et al.*, 2009). Tudo isso torna a tarefa de modelá-lo muito complexa e por isso normalmente apenas alguns mecanismos, células, tecidos e substâncias, que são relevantes ao que se pretende estudar, são incluídos no modelo. Estes limites de abstração serão discutido em seções posteriores.

A seguir serão mostrados alguns métodos que, segundo a literatura, são utilizados na modelagem de sistemas CAS, e algumas argumentações sobre as vantagens e desvantagens de utilizá-los na modelagem do SI.

2.2.1 Métodos de Modelagem

Segundo Li *et al.* (2009), os métodos de modelagem de sistemas CAS podem ser divididos, segundo sua abordagem de construção do modelo, em métodos que utilizam a abordagem *top-down* e métodos que utilizam a abordagem *bottom-up* (LI *et al.*, 2009).

Abordagem *top-down*: Segundo Li *et al.* (2009), esta abordagem soluciona o problema criando um modelo com um grande número de entidades, não dando ênfase às entidades microscópicas, mas sim estimando seu comportamento em um nível macroscópico. É o caso de modelos baseados em Equações Diferenciais Ordinárias (EDO), em que todo o comportamento é simulado através de equações que descrevem o avanço de uma determinada população de entidades, com base no avanço das demais. Questões como espaço e topologia são ignoradas;

Abordagem *bottom-up*: Segundo Li *et al.* (2009), esta abordagem dá ênfase às interações a nível microscópico. Através desta técnica, podemos observar as interações entre entidades específicas e estudar como cada entidade contribui, com o seu comportamento microscópico, para a emergência do comportamento global do sistema. No entanto, essa abordagem requer um grande poder computacional para conseguir simular um grande número de entidades e seus comportamentos. Portanto, conclui o autor, esta abordagem é capaz de fornecer uma representação mais precisa, mas com um custo computacional elevado, tanto de tempo como de memória.

Diante do exposto, serão apresentados agora alguns métodos que utilizam essas abordagens, sendo eles: as EDOs, os Autômatos Celulares (CA) e os Sistemas Multiagentes (SMA).

2.2.1.1 Equações Diferenciais Ordinárias

A modelagem utilizando EDOs é uma técnica de simulação contínua que usa a abordagem *top-down* (LI *et al.*, 2009). Através deste método, os modelos são feitos com base em equações que descrevem o comportamento de populações de entidades. Segundo Li *et al.* (2009), seu processo de modelagem possui quatro etapas:

1. determinar a granularidade do modelo, *i.e.*, quem serão as entidades, os indivíduos (homogêneos) que formam as populações, cujo comportamento se deseja estudar;
2. determinar (ou estimar) as relações entre as populações, *i.e.*, criar fórmulas matemáticas que relacionem o número de entidades em uma população com o número de entidades de outra;
3. formular as EDOs, *i.e.*, descrever a dinâmica de aumento e diminuição das populações em relação ao tempo;
4. analisar o modelo e retirar informações dos comportamentos observados. Segundo Li *et al.* (2009), existem diversos métodos de análise numérica que são úteis nessa etapa.

Esse método trata as entidades como sendo grupos homogêneos que formam populações, ignorando a estrutura espacial do sistema biológico em escala microscópica (LI *et al.*, 2009). Por este motivo, esta técnica não é muito adequada para estudar comportamentos microscópicos, e como estes afetam os comportamentos globais (LI *et al.*, 2009). Seus méritos, segundo Li *et al.* (2009) são o baixo custo computacional e a facilidade de modelagem e implementação.

2.2.1.2 Autômatos Celulares

Este método usa a abordagem *bottom-up*, e foi amplamente usado entre 1999 e 2003 (LI *et al.*, 2009). Ainda segundo Li *et al.* (2009), “*a função básica dos Autômatos Celulares é computar suas regras: não há autonomia*” (LI *et al.*, 2009, p. 79). Mesmo assim, ele é capaz de modelar muitos sistemas complexos, utilizando uma grande quantidade de células com regras específicas de interação.

A modelagem de sistemas baseados em AC utiliza um conjunto de células, que possuem estados e regras capazes de modificarem estes estados com base nos estados das células vizinhas.

Segundo Li *et al.* (2009), a principal vantagem no uso desse método para a modelagem do SI é a possibilidade de análise da estrutura espacial e as interações locais. No entanto, o autor afirma que as desvantagens do uso de ACs incluem o fato de não haver movimentação nas células, o que não está de acordo com o mundo real, tornando difícil extrair desses modelos a essência do que está acontecendo e descobrir regras gerais que descrevam os comportamentos do sistema. Mais ainda, segundo Tay e Jhavar (apud LI *et al.*, 2009) os modelos baseados em AC são “*muito simples para modelar o SI corretamente, e facilmente levam a falsos resultados*” (LI *et al.*, 2009, p. 79).

2.2.1.3 Sistemas Multiagentes

Os modelos baseados em agentes têm sido muito utilizados para simular, entender e prever comportamentos emergentes de sistemas CAS, incluindo o SI (LI *et al.*, 2009). Utilizam uma abordagem *bottom-up* em sua construção, e neles cada agente representa uma entidade do sistema real, sendo que estes agentes podem ser heterogêneos, no sentido de que cada um possa ter seus próprios estados e regras, podendo interagir com os demais agentes, ao contrário do que acontece com a utilização de EDOs. Segundo Li *et al.* (2009), as simulações baseadas em agentes descrevem com mais precisão as interações reais entre as entidades do sistema do que os modelos baseados em AC.

Segundo Hübner *et al.* (2004), na modelagem utilizando SMA com a abordagem *bottom-up*, o sistema é construído “*definindo os aspectos individuais, relacionados aos agentes, de tal forma que ocorra a emergência dos aspectos coletivos*”, desta forma, “*a interação e organização são definidos do ponto de vista dos agentes*” (HÜBNER *et al.*, 2004, p. 2). Ainda segundo o autor, “*o objetivo do SMA passa a ser então a definição de modelos genéricos de agentes, e regras de interações e organizações*” (HÜBNER *et al.*, 2004, p. 2).

Algumas das vantagens do uso de modelos baseados em agentes é que eles permitem

“*explorar a emergência de funções macroscópicas complexas e determinísticas a partir de interações microscópicas estocásticas*” (LI *et al.*, 2009, p. 80). Por este motivo, é possível verificar hipóteses de como células interagem entre si, e de como comportamentos emergem dessas interações.

Segundo Hübner *et al.* (2004), os modelos utilizando SMA são adaptativos e evolutivos, permitindo a adição ou a eliminação de agentes, e mudanças em sua organização, para que o modelo seja utilizado para propósitos diferentes. O autor também afirma que os SMA são uma metáfora natural para a modelagem de sistemas complexos e distribuídos (assim como o SI (LI *et al.*, 2009)), onde o controle, o conhecimento e os recursos estão distribuídos.

Além disso, Hübner *et al.* (2004) acrescenta que os agentes são concebidos independentemente do problema a ser resolvido, *e.g.*, o mesmo agente que representa um linfócito T pode ser utilizado em um sistema de teste de vacinas, ou em um sistema de estudo de doenças autoimunes. As regras de comportamento do linfócito são as mesmas.

No entanto, a principal desvantagem do uso de modelos baseados em agentes é o fato da grande demanda de poder computacional (de velocidade de processamento e de memória) que estes modelos têm, devido ao grande número de agentes que utilizam, às vezes com regras complexas de comportamento, e aos cálculos necessários para a simulação do ambiente, como difusão de substâncias, no caso dos modelos do SI (LI *et al.*, 2009).

2.2.2 Comparação entre os métodos

Segundo Li *et al.* (2009), a utilização de EDOs é a menos indicada para a modelagem do SI, se a intenção do modelo for estudar as interações microscópicas e o comportamento emergente resultante, o que é o caso deste trabalho. Entre outros motivos, Li *et al.* (2009) argumenta que os modelos baseados em EDOs utilizam técnicas de simulação contínua, o que torna difícil observar eventos discretos; esses modelos têm um escopo reduzido, tendo que representar células como populações ou sub-populações, e considerando-as homogêneas, o que pode fazer com que o modelo ignore fatores importantes; e por fim, esses modelos não levam em conta interações locais. Já o autor de Rapin *et al.* (2010), afirma que os modelos baseados em EDOs simplificam muito o sistema, fazendo com que uma população inteira de entidades discretas seja representada por uma única variável contínua. Para alguns propósitos, isso é muito positivo, como no caso dos modelos do HIV, os quais já foram simulados com muito sucesso através de modelos baseados em EDOs. Já quando o objetivo da pesquisa está em evidenciar comportamentos emergentes como consequências de alterações microscópicas em interações entre as entidades, esse tipo de modelagem não fornece as condições necessárias.

Já sobre os modelos baseados em AC, Li *et al.* (2009) argumenta que, embora esses modelos sejam mais apropriados para simular o SI do que os modelos baseados em EDOs, as regras do AC são muito simples para representar as interações complexas intrínsecas aos sistemas biológicos, como o SI, por não permitirem, por exemplo, movimentação real das células (LI *et al.*, 2009).

Por sua vez, sobre os modelos do SI baseados em agentes, Li *et al.* (2009) afirma que “até o momento, SMA é o melhor método para construir o modelo” (LI *et al.*, 2009, p. 80). Isso porque os modelos que utilizam SMA conseguem simular um grande número de interações estocásticas e têm um escopo de modelagem mais amplo, sendo possível criar modelos mais detalhados, de forma mais simples, descrevendo agentes em nível individual e descrevendo as interações entre agentes através de regras de comportamento.

Concluindo, o autor de Li *et al.* (2009) afirma que modelos baseados em EDOs são bons para responder perguntas do tipo “o que e quando” do fenômeno observado, enquanto os modelos baseados em SMA são bons para descobrir características do tipo “por que e como”.

O autor Wooldridge (2001), por sua vez, embora não faça comparação de métodos, e não fale especificamente sobre o SI, afirma que existem algumas condições que indicam se o uso de SMA na modelagem do problema é a mais apropriada, dentre elas:

- Quando o ambiente é altamente dinâmico, incerto ou complexo;
- Quando os agentes são uma metáfora natural para as entidades que serão modeladas;
- Quando os dados, o controle ou o conhecimento para resolver o problema (perícia) estão distribuídos entre as entidades.

O SI é dinâmico, incerto e complexo (FOLCIK *et al.*, 2007; RAPIN *et al.*, 2010), sendo que os dados (especificidade), o conhecimento (qual estratégia melhor para cada tipo de patógeno), e o controle (regulação) para eliminar a infecção está distribuído entre as células, as quais são como agentes, observando o ambiente e efetuando suas ações, em colaboração com os demais. Portanto, pode-se concluir que o SI atende às condições.

Diante do exposto, e tendo em vista a finalidade do trabalho que é o estudo da autoimunidade, como comportamento emergente de interações microscópicas entre células, o modelo que será apresentado no Capítulo 3 será baseado em agentes, *i.e.*, será um SMA.

2.3 Sistemas Multiagentes

Desde meados da década de 80, a Inteligência Artificial Distribuída (IAD) desenvolveu-se fortemente, sendo hoje um dos campos de pesquisa e aplicação mais promissores do ponto de vista tecnológico (WEISS, 1999). Esse campo tem como característica sua alta multidisciplinaridade, trazendo consigo conhecimentos de várias outras disciplinas, como Ciência da Computação e a Inteligência Artificial (IA), Sociologia, Economia, Etologia e Biologia (HÜBNER *et al.*, 2004; WEISS, 1999).

A principal ferramenta da IAD é a utilização de sistemas conhecidos como Sistemas Multiagentes, por serem compostos por entidades computacionais inteligentes e autônomas, chamadas de agentes.

O avanço dessa disciplina, segundo Weiß (1999), deve-se primariamente a duas razões: primeiro, pelo fato da evolução natural da humanidade, e conseqüentemente dos problemas com os quais ela tem que lidar. Esse desenvolvimento fez com que os sistemas computacionais se tornassem cada vez mais complexos, dinâmicos e distribuídos, e deixassem de ser programas únicos e isolados. Esse novo paradigma fez com que fosse necessária a utilização de tecnologias mais avançadas, que fossem capazes de agir de forma distribuída e autônoma, diante de problemas dinâmicos. E os SMA vieram ao encontro dessa nova demanda (WEISS, 1999).

Já, em segundo lugar, o autor de Weiß (1999) afirma que os SMA têm a capacidade de desempenhar um importante papel no desenvolvimento e análise de modelos e teorias (WEISS, 1999). Isso fez com que esse tipo de tecnologia tenha um grande apelo científico.

Formalmente, os **Sistemas Multiagentes** (SMA) são definidos como sistemas formados por (ou baseados em) um número de entidades autônomas que interagem entre si em um **ambiente**, cooperando mutuamente para resolverem um problema cuja solução está além da capacidade individual de cada componente (WOOLDRIDGE, 2001; HÜBNER *et al.*, 2004; WEISS, 1999). Os SMA têm como objetivo o estudo da coletividade, sendo o foco de sua atenção a forma de interação entre as entidades que formam o sistema (denominadas **agentes**) e sua organização.

A seguir serão discutidos esses dois conceitos-chave para um melhor entendimento dos SMA: ambiente e agente.

2.3.1 Ambiente

O ambiente é o meio, o contexto, no qual os agentes estão inseridos. É um mundo, e o agente assume que não existe nada além daquele universo.

A seguir serão apresentadas as principais propriedades de um ambiente e suas classificações, segundo Wooldridge (2001):

Acessível × Inacessível: um ambiente acessível é aquele do qual pode-se obter informações completas, precisas e atuais sobre o seu estado. A maioria dos ambientes não são acessíveis. No espaço, por exemplo, a informação sobre a posição de estrelas tem um atraso na escala de anos-luz.

Determinístico × Não-determinístico: em um ambiente determinístico, uma ação qualquer tem um único e garantido efeito, *i.e.*, uma ação determina o próximo estado do ambiente.

Estático × Dinâmico: em um ambiente estático, pode-se assumir que seu estado irá continuar imutável até que algum agente o modifique. Por outro lado, um ambiente dinâmico pode ter seu estado alterado por outros processos que estejam atuando sobre ele.

Discreto × Contínuo: um ambiente é dito discreto se existe um número finito de estados que ele possa assumir, mesmo que seja um número muito grande. Já um ambiente contínuo pode assumir um número infinito de estados.

Ao construir um SMA, é fundamental que se defina em que tipo de ambiente os agentes estarão inseridos, a fim de se definir que tipo de informação eles poderão obter do meio.

2.3.2 Agentes

Embora haja um consenso sobre a definição do SMA, o mesmo não ocorre sobre o termo agente autônomo. Segundo Wooldridge e Jennings (apud WOOLDRIDGE, 2001), “*um agente é um sistema computacional, situado em um ambiente, que é capaz de tomar ações autônomas nesse ambiente com a finalidade de atingir os objetivos para os quais fora construído*” (WOOLDRIDGE, 2001, p. 15). Outra linha de pensamento, utilizada por Weiß (1999), Hübner *et al.* (2004), afirma que um agente autônomo “*significa o fato de um agente ter sua existência independente dos demais, e até mesmo do problema a ser solucionado*” (HÜBNER *et al.*, 2004, p. 2), *i.e.*, “*para funcionar, um agente não precisa de outros agentes, mesmo que para alcançar seus objetivos ele eventualmente vá precisar da ajuda de outros.*” (HÜBNER *et al.*, 2004, p. 2).

Não faz parte do escopo deste trabalho entrar nesta discussão, por isso será utilizada a definição de que um agente autônomo é um sistema computacional, situado em um ambiente, capaz de percebê-lo e tomar ações baseadas nessa percepção, através de suas regras internas de comportamento.

Segundo Hübner *et al.* (2004), Wooldridge (2001), existem dois tipos de arquiteturas para agentes em um SMA: os **agentes cognitivos** (ou deliberativos) e os **agentes reativos**.

Os SMA baseados em agentes cognitivos normalmente possuem poucos agentes, e estes são sistemas sofisticados e computacionalmente complexos, por isso tê-los em grande quantidade no SMA seria inviável por causa do grande custo computacional (HÜBNER *et al.*, 2004). Esse tipo de agente possui uma representação explícita de conhecimento sobre o ambiente e os demais agentes, possuem memória e mecanismos de raciocínio lógico, que permitem a eles planejar suas ações. Além disso, esse tipo de agente pode se comunicar diretamente com os demais. A arquitetura mais difundida na construção de agentes cognitivos é a chamada BDI (crenças, desejos e intenções, do Inglês, *belief, desire and intention*) (WOOLDRIDGE, 2001; HÜBNER *et al.*, 2004).

Já os SMA baseados em agentes reativos surgiram da ideia de que um comportamento inteligente em um sistema emerge da interação entre um grande número de agentes muito simples, que não precisam especificamente terem representações explícitas de conhecimento ou mesmo algum tipo de mecanismo de raciocínio lógico (WOOLDRIDGE, 2001; HÜBNER *et al.*, 2004).

Desta forma, segundo Wooldridge (2001), Hübner *et al.* (2004), os agentes reativos “*têm comportamentos que podem ser descritos como autômatos finitos simples, possuindo um conjunto de regras que mapeiam percepções do ambiente diretamente em ações sobre estes, em um tipo de esquema estímulo-resposta.*” (HÜBNER *et al.*, 2004, p. 6).

Assim, os agentes reativos têm dois conceitos principais:

percepção: o agente é capaz de perceber alterações no ambiente;

ação: as alterações no ambiente são provenientes das ações que os agentes realizam constantemente no mesmo, sendo que normalmente os agentes não têm controle total do ambiente, mas podem apenas influenciá-lo de alguma forma;

Um sub-tipo dos agentes reativos são os **agentes reativos com estado** (WOOLDRIDGE, 2001), que, além de serem reativos, *i.e.*, agirem de acordo com o que observam no ambiente, eles são capazes de armazenar o seu estado, sendo que sua ação passa a ser uma função da sequência de estados e de sua percepção do ambiente. Portanto, suas decisões sofrem influência do seu histórico de ações. A Figura 2.18 ilustra a arquitetura desse tipo de agente.

As principais vantagens do uso de agentes reativos são a simplicidade na modelagem do agente e a economia de recursos de tempo e de computação, justamente por não terem

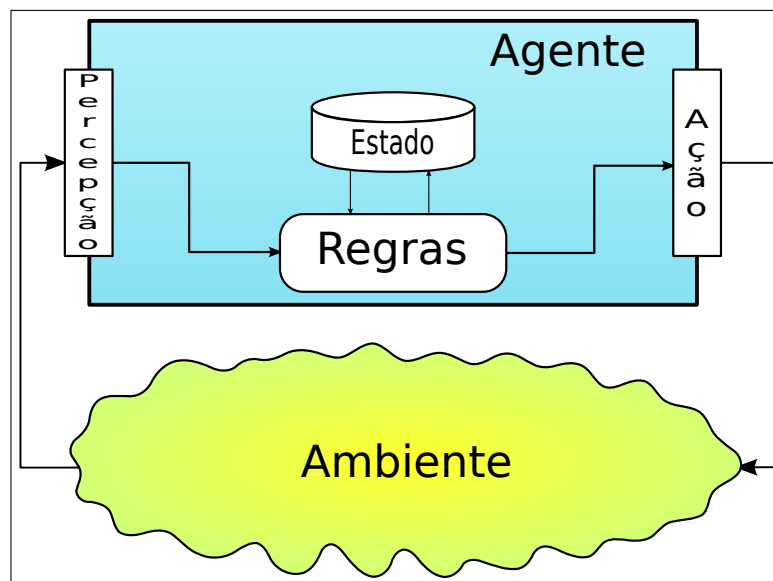


Figura 2.18. Modelo geral de um agente reativo com estado.

Fonte (adaptado): (WOOLDRIDGE, 2001, p. 36)

os mecanismos dos agentes cognitivos. No entanto, Wooldridge (2001) alerta para as dificuldades em se criar SMA reativos, como:

- Os agentes reativos precisam ter informações disponíveis no seu ambiente para tomar sua decisão;
- Esses sistemas se baseiam no fato de que o comportamento inteligente emerge das relações entre os comportamentos entre os agentes. No entanto, às vezes essas interações não são bem compreendidas, fazendo com que, para se alcançar o comportamento global desejado, seja necessário utilizar o método de experimentação, tentativa e erro para desenvolver os agentes. Essa segunda desvantagem é justamente o caso da modelagem do SI, onde as interações nem sempre são bem conhecidas. Exatamente por isso, em algumas ocasiões, as regras de interação dos agentes precisam ser definidas empiricamente.

Diante disso, e baseado no fato de os agentes reativos com estado serem, analogamente, mais próximos das células biológica do que os agentes deliberativos, o modelo proposto por este trabalho, no Capítulo 3, será baseado em agentes reativos com estados.

2.4 Modelagem baseada em agentes

A modelagem e simulação baseada em agentes é uma nova abordagem de pesquisa, que ganhou grande força nos últimos 20 anos, sendo chamada por alguns de “terceira

maneira de se fazer ciência” (MACAL; NORTH, 2009, p. 86). Trata-se da utilização do conceito de agentes inteligentes, descritos na Seção 2.3.2, para construir modelos abstratos de sistemas a serem estudados, onde os agentes interagem continuamente, através das simulações desse modelo.

A principal motivação do uso dessa abordagem está no fato do mundo estar se tornando um ambiente cada vez mais complexo e com isso os sistemas que precisam ser analisados e modelados também estão mais complexos. Nesse contexto, ferramentas de modelagem convencionais não são mais viáveis como eram anteriormente. Além disso, o poder computacional está avançando rapidamente, permitindo que modelos baseados em agentes, que antes eram inviáveis, possam agora ser utilizados (MACAL; NORTH, 2009).

Esse tipo de modelagem têm ligação direta com os Sistemas Adaptativos Complexos (CAS), se preocupando com a questão sobre como comportamentos complexos emergem da interação de agentes inteligentes (MACAL; NORTH, 2009). O conceito de CAS, segundo Macal e North (2009), surgiu originalmente com a finalidade de investigar a adaptação e emergência dos sistemas biológicos. Os autores de Macal e North (2009) afirmam ainda que ao desenvolver um modelo baseado em agentes, está-se desenvolvendo um modelo de um CAS.

Dessa forma, a ideia principal da modelagem utilizando agentes baseia-se no fato de que sistemas naturais aparentam exibir uma inteligência coletiva, emergente de regras simples, executadas por suas entidades, baseadas em informações locais, permitindo a esses sistemas naturais, não só sobreviver, mas também se adaptar ao seu ambiente, otimizando seu comportamento com o passar do tempo (MACAL; NORTH, 2009). Essa também é a ideia fundamental em que se baseia os sistemas bio-inspirados.

2.4.1 Criando modelos baseados em agentes

Segundo Macal e North (2009), identificar quem são os agentes, especificar seus comportamentos, e representar apropriadamente suas interações são a chave para se desenvolver modelos baseados em agentes com sucesso.

A primeira etapa no desenvolvimento diz respeito à identificação dos agentes e suas classes (MACAL; NORTH, 2009). Isso é feito definindo-se o nível de granularidade, *i.e.*, o nível de abstração do modelo, de forma que se consiga o nível mais grosso possível, mas que ainda mantenha os detalhes necessários para que o modelo apresente o comportamento desejado (FOLCIK *et al.*, 2007).

Uma vez que os agentes tenham sido identificados, suas regras são estabelecidas. Para tanto, é necessário que se tenha uma base teórica sobre o comportamento esperado dos agentes, de onde serão extraídas as suas regras.

Definidos os agentes e suas regras, devem-se definir os mecanismos que irão controlar seus relacionamentos, definindo como, com quem e quando os agentes irão interagir. Segundo Macal e North (2009), essa tarefa é crucial, sendo mais importante definir bem as relações e interações dos agentes, do que definir os próprios agentes e suas regras.

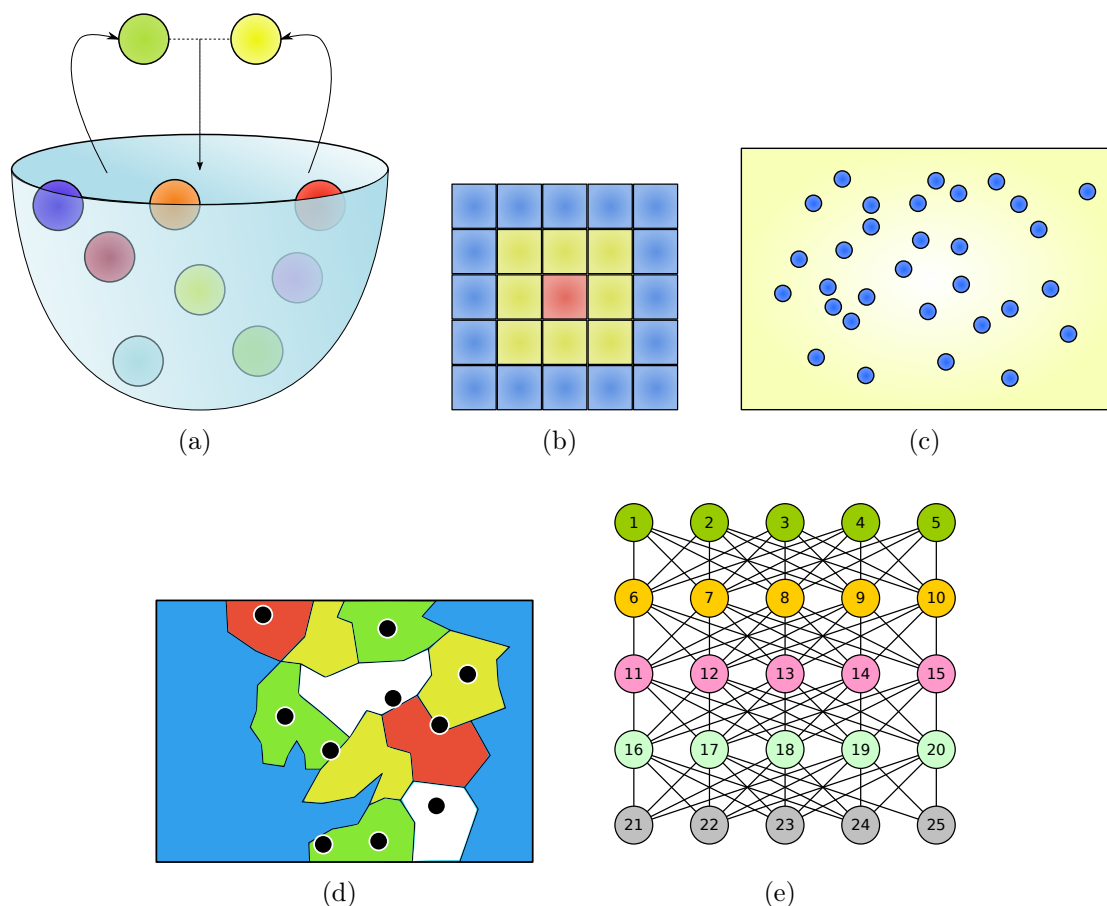


Figura 2.19. Ilustração das diferentes topologias dos agentes. Em (a), a topologia do tipo sopa, onde os agentes não têm uma relação espacial definida. Em (b), a topologia no tipo de autômato celular, onde cada agente permanece em uma célula. Em vermelho, a posição do agente, e em amarelo, sua Vizinhança de Moore. Em (c), a topologia do tipo espaço euclidiano, onde cada agente tem um posição em um espaço contínuo; Em (d), a topologia do tipo GIS, em que as posições dos agentes representam posições geográficas no mundo real, embora a ilustração não esteja muito fiel à realidade; Em (e), a topologia do tipo grafo, ou *network*, onde cada agente possui a informação sobre com quem pode interagir, não sendo necessariamente alguém próximo, no sentido espacial.

Fonte (adaptado): (MACAL; NORTH, 2009, p. 94)

A primeira tarefa em definir os relacionamentos dos agentes está em definir qual agente poderá conectar-se a outro. Segundo Macal e North (2009), existem várias topologias comuns, já utilizadas largamente, para definir essa relação, que são:

Sopa: nessa topologia, ilustrada na Figura 2.19(a), também conhecida como modelo espacial, os agentes não têm uma representação definida de espaço. Nesses modelos,

os agentes normalmente são escolhidos aos pares, interagem, e depois são devolvidos à sopa de agentes;

Autômato Celular: nessa topologia, ilustrada na Figura 2.19(b), os agentes estão contidos em uma estrutura de grade, onde cada agente ocupa uma célula (mas uma célula pode ser ocupada por vários agentes), e onde os agentes em posições ao seu redor são chamados de vizinhos;

Espaço Euclidiano: nesse tipo de topologia, ilustrada na Figura 2.19(c), os agentes estão em um espaço contínuo, bi-dimensional, tri-dimensional, ou multi-dimensional;

GIS: nessa topologia, ilustrada na Figura 2.19(d), chamada também de SIG (Sistema de Informação Geográfica, do Inglês GIS: *Geographic Information System*), os agentes se encontram em uma representação geográfica real, com coordenadas espaciais reais;

Grafo: nessa topologia, ilustrada na Figura 2.19(e), também chamada de rede, interconexões ou *network*, os agentes têm uma representação bem definida de com quem podem interagir, sendo que seus “vizinhos” são definidos de forma mais precisa;

Segundo Macal e North (2009) não importa qual topologia é usada, o importante é garantir aos agentes a possibilidade de interagir localmente, com um número limitado de agentes, em um período de tempo limitado, obedecendo às condições do modelo, evitando que os agentes possam interagir com todos os outros agente ao mesmo tempo.

2.4.2 Frameworks

Os modelos baseados em agentes podem ser construídos utilizando linguagens de programação de propósito geral, ou *frameworks* de construção de *software* genéricos. No entanto, é recomendável utilizar um *framework* próprio para o desenvolvimento de modelos baseados em agentes, visto que estes *toolkits* oferecem recursos e funcionalidades já prontos, que são comuns a esses modelos, o que pode facilitar muito o desenvolvimento (MACAL; NORTH, 2009).

A escolha de qual linguagem de programação ou *framework* utilizar também deve levar em conta o tamanho do modelo, que pode ser pequeno, podendo ser simulado em um computador apenas, ou de larga-escala, usando *clusters* de vários computadores interconectados para sua simulação.

Em Macal e North (2009), o autor descreve uma lista dos *frameworks* mais conhecidos, e suas características, dos quais se destacam Repast, NetLogo, StarLogo, Swarm, MASON e AnyLogic. Mais informações sobre esses e outros *frameworks* podem ser encontradas em Macal e North (2009).

Destes, o *framework* escolhido para a implementação do modelo foi o Repast. Essa escolha se baseia nos fatos de, em primeiro lugar, o Repast prover os mecanismos necessários para a implementação e simulação do modelo do SI baseado em agentes (COLLIER, 2010; MACAL; NORTH, 2009), e, em segundo lugar, pelo fato do BIS, apresentado em (FOLCIK *et al.*, 2007) e utilizado como inspiração para o modelo apresentado neste trabalho (ver Trabalhos Correlatos), também utilizar o *framework* Repast, em sua plataforma RepastJ, apresentando um excelente desempenho e aparência visual. No entanto, optou-se pela plataforma Repast Symphony, ao invés da utilizada pelo BIS, a RepastJ, uma vez que esta foi descontinuada.

Sendo assim, o *framework* escolhido foi o Repast, em sua plataforma mais atual, denominada Repast Symphony, em sua versão 1.2.0. Devido a isso, a descrição do modelo, no Capítulo 3, será feita já prevendo a sua implementação no *framework* citado, facilitando-a. Evidentemente, o modelo descrito poderá ser implementado em qualquer outro *framework*, desde que se faça as devidas adaptações, uma vez que os conceitos modelados são conceitos biológicos, independentes do *framework*.

2.4.2.1 Repast Symphony

Segundo Collier (2010), Macal e North (2009), o *framework* Repast, acrônimo de *Recursive Porous Agent Simulation Toolkit*, foi desenvolvido utilizando puramente a linguagem Java, e por este motivo, pode ser utilizado em diversas plataformas, como Microsoft Windows e o GNU Linux. Foi desenvolvido por um grupo de pesquisadores do laboratório *Argonne National Laboratory*, da Universidade de Chicago.

É um *framework* gratuito e de código aberto, usado amplamente em modelagem baseada em agentes (NORTH *et al.*, 2005). Possui diversas plataformas, entre elas: RepastJ, Repast Symphony ou RepastS, RepastPy, Repast.NET e a Repast HLC, para desenvolvimento de modelos de larga-escala, distribuídos em *clusters* de computação.

Esse *framework* provê diversos mecanismos facilitadores à modelagem dos agentes, suas regras e relações. Os agentes são criados em uma interface visual, onde podem ser organizados em seus ambientes. Também é possível criar diversos parâmetros a serem utilizados pelos agentes, e que podem ter seus valores definidos durante a simulação. Além disso, implementa todas as topologias supracitadas, fornecendo ainda um mecanismo para simulação de difusão de substâncias. Maiores detalhes sobre as características desse *framework* serão discutidas no Capítulo 3.

2.5 Trabalhos Correlatos

Aqui serão apresentados alguns trabalhos que estão relacionados ao modelo aqui apresentado. Não serão todos, mas apenas aqueles cujos modelos têm relevância à luz dos objetivos deste trabalho.

Os principais trabalhos correlatos encontrados na literatura são:

ImmSim É um dos mais conhecidos modelos do SI, escrito em APL. Seu modelo é baseado em duas entidades básicas: as entidades celulares (os linfócitos, por exemplo) e as entidades moleculares (citocinas, por exemplo). Esse modelo foi descrito por Celada e Seiden (apud RAPIN *et al.*, 2010) em 1992, há 20 anos atrás. Como o código-fonte do sistema foi liberado sob a licença GPL, várias outros sistemas foram derivados dele, como a versão em C ANSI, chamada C-ImmSim (RAPIN *et al.*, 2010), e a versão que utiliza programação paralela, chamada de ParImm. Todos estes utilizam uma cadeia de *bits* para representar a especificidade dos receptores e dos antígenos, e utilizam a Distância de *Hamming* entre duas cadeias de *bits* para representar a afinidade. No entanto, trata-se de um modelo em linha de comando, de difícil utilização.

Simmune é uma plataforma para simulações imunológicas. Segundo Rapin *et al.* (2010), esse sistema está mais para uma técnica e uma linguagem de modelagem, do que um modelo em si. Ele se baseia em uma rede tri-dimensional, onde todos os elementos são partículas, inclusive as substâncias, como as citocinas. As células são agentes reativos puros, que agem somente em resposta à estímulos.

BIS também conhecido como “*The Basic Immune Simulator*” (FOLCIK *et al.*, 2007), é um modelo baseado em agentes para estudar as interações entre células da imunidade inata e as células da imunidade adaptativa. Possui três espaços virtuais onde tipos básicos de células interagem. Tem sido utilizado tanto para uso educacional, como para pesquisas envolvendo temas como hipersensibilidade e câncer (RAPIN *et al.*, 2010).

Esses são os trabalhos baseados em agentes mais relevantes encontrados na literatura pesquisada. Embora todos eles têm lições importantes, o BIS foi o que mais próximo chegou do modelo ao qual este trabalho aspira. Utilizando a linguagem Java, uma linguagem moderna, portátil, e de ampla utilização, e o *framework* RepastJ, o BIS se mostrou um excelente trabalho, com uma interface amigável, de fácil utilização mesmo para os profissionais que não são da área da Computação, sendo possível ajustar os parâmetros da simulação sem a necessidade de se alterar o código-fonte do sistema, ou digitar parâmetros na linha de comando. Além disso, o autor do BIS (FOLCIK *et al.*, 2007), embora não

tenha disponibilizado o código-fonte do mesmo, disponibilizou em seu artigo (FOLCIK *et al.*, 2007) uma especificação completa do modelo utilizado no BIS, com diagramas de estado de cada agente responsável por uma célula.

Embora o BIS seja um excelente trabalho, e realize muito bem a tarefa a que se propõe, como o objetivo deste trabalho é o estudo a autoimunidade, o modelo apresentado pelo BIS se torna inviável, uma vez que não seria possível simular mecanismos de tolerância e até mesmo auto-reatividade. Isso porque o BIS apresenta um modelo monoclonal, *i.e.*, cada linfócito T têm apenas uma especificidade, e cada vírus apresenta apenas uma assinatura antigênica, de tal forma que todo linfócito T, por exemplo, reconhecerá o vírus, assim como todo vírus será reconhecido pelo linfócito, e nunca um linfócito irá “se enganar” e atacar uma célula própria do organismo. O BIS também não modela mecanismos como cálculo de afinidade, geração de diversidade, conjunto de antígenos próprios, tolerância, entre outros, que são essenciais para a simulações de fenômenos relacionados à autoimunidade. Além disso, o BIS modela anticorpos como sendo substâncias, como as citocinas, impedindo ainda mais a tentativa de se definir uma especificidade para eles.

Por este motivo, optou-se por se criar um novo modelo, baseado no trabalho apresentado por Folcik *et al.* (2007), mas que modele o conceito de especificidade (e os mecanismos relacionados), baseado na ideia de representar esta especificidade através de uma cadeia de *bits*, como descrito nos trabalhos publicados por Floreano e Mattiussi (2008) e Celada e Seiden (apud RAPIN *et al.*, 2010).

CAPÍTULO 3

PROPOSTA DE MODELO DO SI PARA SIMULAÇÃO DA AUTOIMUNIDADE

Neste capítulo será mostrado o modelo do SI proposto, para que seja usado na implementação da ferramenta de simulação *in-silico* do SI. É válido ressaltar que serão descritos apenas os itens do modelo: comportamentos resultantes das interações dos agentes, como memória, especificidade, declínio da resposta e autoimunidade, não são modelados explicitamente, mas sim, são consequência das regras internas de cada agente e de suas interações entre eles próprios e com o ambiente.

3.1 Requisitos do modelo

Como o objetivo do modelo é permitir o estudo da autoimunidade, os mecanismos que, evidentemente, serão requisitos para o modelo são justamente aqueles necessários para a prevenção da autoimunidade, exatamente os mecanismos cujas falhas podem levar à autoimunidade e ao desenvolvimento de doenças autoimunes.

Como descrito na Seção 2.1, tais mecanismos, chamados de mecanismos de tolerância imunológica, são a Tolerância Central e a Tolerância Periférica.

A Tolerância Central é formada por outros dois mecanismos: a Seleção Positiva, que elimina os linfócitos não funcionais, e a Seleção Negativa, que elimina os linfócitos auto-reativos. Nesse caso, a Seleção Negativa será o foco do modelo, no que diz respeito à Tolerância Central.

Já na Tolerância Periférica, existem diversos mecanismos e hipóteses sobre como eles funcionam. O mecanismo mais descrito na literatura pesquisada refere-se à necessidade de um segundo sinal, o *danger signal*, para informar ao linfócito se o antígeno é perigoso ou não. Evidentemente, quanto mais mecanismos forem modelados, mais preciso o modelo

irá ser. No entanto, por uma questão de escassez de tempo, apenas o mecanismo que envolve os *dangers signals* será modelado. Os demais serão inclusos como Trabalhos Futuros (Seção 5.1).

Além desses, também será necessário um mecanismo de Geração de Diversidade, responsável por gerar as diversas especificidades do modelo, e um banco de dados de antígenos próprios, *i.e.*, os autoantígenos, para serem usados durante o processo de tolerância.

Como pode ser visto, todos esses mecanismos se referem ao reconhecimento de antígenos através de receptores. Conseqüentemente, a modelagem da afinidade, *i.e.*, a força com que esse reconhecimento ocorre, torna-se fundamentalmente importante para os processos de tolerância. Por este motivo, os mecanismos de reconhecimento de antígenos, e cálculo de afinidade também são requisitos do modelo.

3.2 *Framework* alvo

Dados os *frameworks* e suas principais características, apresentados na Seção 2.4.2, o modelo apresentado por este trabalho foi desenvolvido visando sua implementação utilizando o *framework Repast Symphony*, em sua versão 1.2.0, para a plataforma Linux, embora o modelo tenha sido portado para a plataforma *Microsoft Windows* sem maiores problemas.

Dito isto, é válido ressaltar que as definições de espaço, tempo, representações de células, substâncias, tecidos, entre outros, foram concebidas já prevendo a utilização do *framework* em questão, por uma questão de praticidade.

No entanto, nada impede que tal modelo seja utilizado como base para a construção de adaptações que sejam compatíveis com outros *frameworks* e/ou plataformas, uma vez que os conceitos imunológicos e biológicos por trás do modelo não dependem da tecnologia utilizada para representá-los.

3.3 Ambiente

Com base nas características descritas na Seção 2.3.1, podemos definir o ambiente modelado, pela perspectiva dos agentes, como sendo:

Inacessível: os agentes não têm informação sobre o ambiente como um todo, apenas informação local, *e.g.*, de sua vizinhança;

Não-determinístico: um agente não pode prever o próximo estado do ambiente baseado apenas em sua ação. As ações de outros agentes, assim como ações externas (passagem do tempo, difusão de substâncias, por exemplo) podem interferir no próximo estado do ambiente;

Dinâmico: o ambiente pode modificar seu estado, mesmo se o agente escolher não tomar nenhuma ação;

Discreto: é fato que o número de estados que este ambiente pode assumir é muito alto, dado o número de agentes, de posições possíveis, de estados possíveis, de substâncias e dispersões possíveis, *etc.* No entanto, não é um número infinito de estados, portanto, trata-se de um ambiente discreto.

Dadas as características do ambiente, agora serão definidas duas grandezas de suma importância para o modelo: o tempo e o espaço.

3.3.1 Tempo

O transcorrer do tempo é modelado utilizando o conceito de unidade de tempo discreta disponibilizado pelo *framework*, chamado de *tick*.

Cada agente agenda um método próprio, por convenção chamado no modelo de *step*, utilizando o recurso *annotations*¹, para ser executado pelo *framework* quando possível. Ao anotar o método, o agente informa quando o método deve começar a ser chamado e o intervalo de cada chamada, ou seja, em qual *tick* o método deve começar a ser chamado e de quantos em quantos *ticks* ele deve ser chamado novamente.

Os *ticks* são intervalos de tempos necessários para a transição de um estado do ambiente para o próximo. Portanto, todos os eventos agendados devem ser executados e concluídos para que só então o próximo *tick* possa ocorrer, criando assim um sincronismo na simulação. Desta maneira, durante um *tick*, todos os agentes vão modificar suas posições, liberar substâncias, analisar sua vizinhança, baseados nas informações do *tick* anterior. Só quando todos tiverem realizado suas ações é que o *tick* termina e suas informações são atualizadas. Isso faz com que, na visão do agente, tudo esteja ocorrendo concorrentemente.

A duração real de cada *tick* varia muito em função do poder de processamento da máquina que estiver simulando o modelo, mas não é tão relevante, a não ser para fins de otimização, estudar esse tempo. Por outro lado, a abstração do *tick* em uma unidade

¹*Annotations* são uma forma sintática especial da linguagem de programação Java que permite adicionar informações, como propriedades especiais, a algumas outras formas sintáticas, como os métodos.

de tempo real no modelo tem grande relevância para a pesquisa em questão. Em outras palavras, definir se um *tick* no modelo representa (ou simula) uma hora, um dia, ou um mês no mundo real, é muito importante.

No entanto, devido a extensão do trabalho, o pouco tempo para desenvolvimento do mesmo e escassez de informações precisas na literatura pesquisada, quanto ao tempo necessário para que as células imunológicas realizem suas tarefas, essa abstração da unidade de tempo não será modelada, sendo incluída como sugestões para Trabalhos Futuros (Seção 5.1).

3.3.2 Espaço

O conceito de espaço, ou topografia dos agentes, é modelado utilizando os recursos espaciais do *framework*, dentre eles, o conceito de projeção. No *Repast*, os agentes ficam imersos no ambiente (veja Contexto, na Seção 3.3.3), como que em uma “sopa” de agentes: eles não têm uma relação espacial uns com os outros. Essa relação espacial é obtida através do uso de projeções.

3.3.2.1 Projeções

Existem diversos tipos de projeções pré-definidas no *Repast*, entre as quais está a projeção em espaço Euclidiano (espaço contínuo), a projeção em *network* (grafo) e a projeção em forma de autômato celular, chamada no *framework* de *Grid*.

Como também visto, o tipo de projeção escolhida deve levar em conta que tipo de relação que se deseja representar. Por exemplo, ao se utilizar uma projeção do tipo *network*, os agentes têm a informação de “quem está conectado a mim”. Já quando é utilizado uma projeção do tipo autômato celular, os agentes têm a informação de “onde estou” ou “quem está próximo a mim”.

No modelo em questão, deseja-se representar a relação espacial propriamente dita, *i.e.*, a coordenada de um determinado agente, sua distância em relação a outros agentes, e *etc.* Por analogia, o tipo de projeção que parece ser o mais viável seria o espaço Euclidiano, *i.e.*, uma representação espacial contínua 2D, onde os agentes teriam coordenadas cartesianas reais (x, y) indicando sua posição no espaço. No entanto, essa representação torna tarefas como difusão de substâncias, definição da relação de vizinhos, *etc.*, muito custosas computacionalmente, por tratar um espaço contínuo.

Por este motivo, optou-se por utilizar um espaço discreto, a projeção *Grid*, onde as linhas verticais e horizontais formam células espaciais bem definidas às quais estão

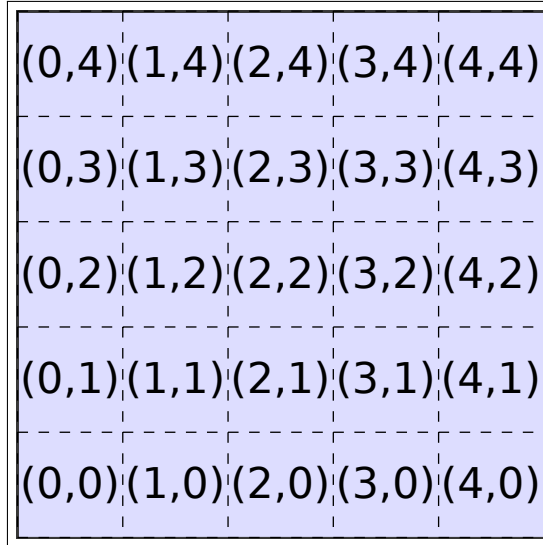


Figura 3.1. Ilustração da estrutura do espaço definido por uma projeção do tipo **Grid**.

vinculadas coordenadas espaciais i e j , definindo a coluna (i) e a linha (j) onde a célula espacial se encontra. A Figura 3.1 ilustra a estrutura de uma projeção do tipo **Grid**.

Através desse tipo de projeção, cada agente passa a ter uma posição (i, j) . A movimentação do agente é feita alterando-se sua posição, para uma posição em sua Vizinhança de Moore (Figura 3.2). Dessa forma, um agente não consegue “pular” posições: ele precisa percorrer um caminho até chegar a posição desejada. Na Figura 3.2 está ilustrado a estrutura da Vizinhança de Moore.

Essas projeções têm um tamanho definido, *i.e.*, uma largura (W) e uma altura (H). Desta forma, a posição $P(i, j)$ de um agente pode ser definida como:

$$P(i, j) = \{i \in \mathbb{N} : 0 \leq i < W, j \in \mathbb{N} : 0 \leq j < H\} \quad (3.1)$$

Quando um agente tenta se mover para uma posição que esteja fora desses limites, o *framework* permite duas opções: não mover, ou simular um espaço toroidal, *e.g.*, se um agente está na posição $P(W - 1, j)$ e tenta se mover para a posição $P(W, j)$, ele é transportado para a posição $P(0, j)$. Da mesma forma com as coordenadas das linhas. Em outras palavras, ao passar pela borda da direita, por exemplo, o agente aparece na mesma linha, só que na borda da esquerda.

Como foi observado que o espaço simulado (cerca de $1,21mm^2$, como descrito mais adiante, na Seção 3.3.4) é pequeno comparado ao espaço livre que uma célula possui em um organismo real, com o intuito de tentar simular um espaço maior e mais próximo da realidade, se cogitou aumentar o tamanho do espaço, mas foi evidenciado que isso seria computacionalmente caro. Ao invés disso, foi definido que seria utilizado a segunda

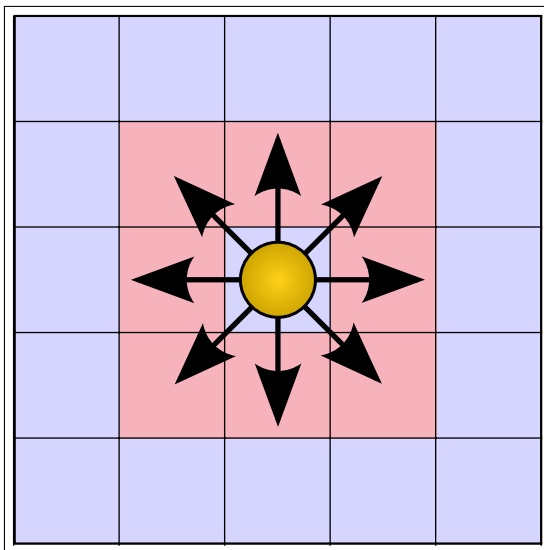


Figura 3.2. Vizinhança de Moore. Em uma estrutura de células bidimensional na forma de grade, a Vizinhança de Moore (de raio igual a um) compreende as oito células vizinhas à uma célula central. Seu nome é em homenagem à Edward F. Moore, um pioneiro no estudo dos autômatos celulares. Na figura, em amarelo, está o agente na célula central. As células em vermelho formam a Vizinhança de Moore de raio 1, e são as possíveis posições para as quais ele pode se movimentar durante o *tick*.

opção do *framework*, utilizando um espaço toroidal (Figura 3.3) para representar o espaço. Evidentemente, o espaço no organismo não é infinito, e esse não é o propósito desta escolha. Pelo contrário, o que se tenta é simular um espaço que pareça maior aos olhos do agente, mais perto assim da realidade.

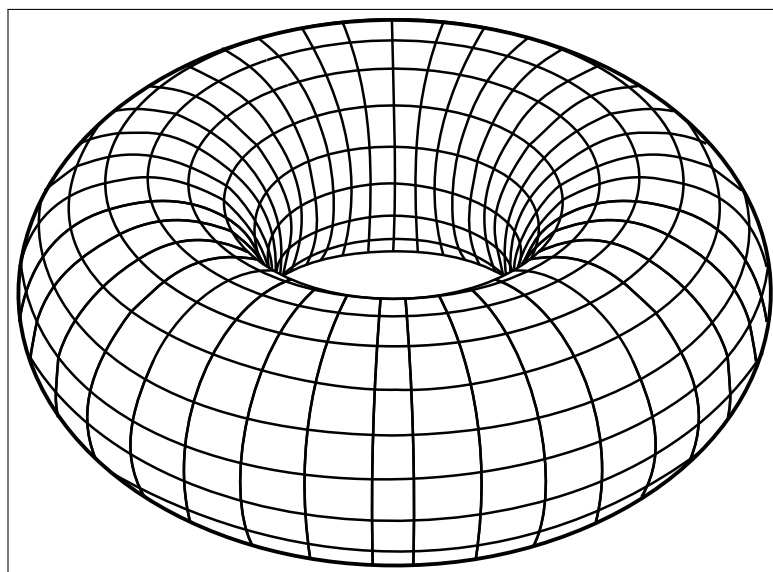


Figura 3.3. Ilustração de um espaço toroidal. ²

²Fonte da imagem: <http://bit.ly/yqXKJ4>. Acessado em: 21/04/2012.

Além disso, quando um agente tenta se mover para uma posição onde já existe um outro agente, foi definido que é permitido que eles ocupem o mesmo espaço. Dessa forma, tenta-se simular um espaço 3D, onde as células possam passar “por cima” ou “por baixo” uma das outras. Em outras palavras, é como observar um espaço 3D “de cima”, em uma projeção paralela ou ortogonal (e não em perspectiva).

3.3.3 Contextos

A funcionalidade do *framework* mais próxima do conceito de ambiente em SMA é chamada de `Context` (contexto, em Inglês). Segundo Collier (2010), o contexto é o conceito principal do *framework*, fornecendo uma estrutura de dados que permite a organização dos agentes, tanto do ponto de vista do modelo, quanto do ponto de vista do *software*. Dessa forma, o contexto é a estrutura de dados onde os agentes “existem”.

O contexto provê ainda inúmeras funcionalidades, como:

Parâmetros Os contextos podem ter parâmetros associados a eles. Os agentes podem acessar os parâmetros do contexto onde estão situados. Por exemplo, podemos ter dois contextos, representando dois planetas. E cada contexto pode ter uma variável chamada “aceleração da gravidade”. Quando o agente acessar o parâmetro “aceleração da gravidade”, este pode ter valores diferentes, dependendo do contexto em que o agente se encontra. Os parâmetros utilizados no modelo estão descritos na Seção 3.3.3.1;

Projeções Como vimos na Seção 3.3.2, ao criar os agentes e adicioná-los a um contexto, estes ficam livres no espaço, como em uma “sopa de agentes”. O componente responsável por impôr uma estrutura nestes agentes é chamada de `Projection`. Cada contexto pode ter várias projeções associadas a ele, viabilizando a definição de várias estruturas para o mesmo grupo de agentes, ou, em outras palavras, vários pontos de vista sobre eles;

Sub-contextos Os contextos também suportam a adição de sub-contextos. Dessa forma, os sub-contextos herdam todas as características dos seus ancestrais, e podem definir outras novas. Por exemplo, podemos ter um contexto chamado “sistema solar”, e cada planeta poderia ser representado por um sub-contexto deste.

Camada de Dados Em certas situações, um agente precisa de informações que mudam em relação a sua posição e o tempo. Por exemplo, quando um agente simula a liberação de uma substância, e o *framework* simula uma difusão sobre a mesma: a quantidade de substância irá depender do local onde o agente está e de quando ele

aferiu essa quantidade. Para atender a esse requisito comum a muitos modelos biológicos, o Repast oferece uma funcionalidade denominada `ValueLayers` (camada de valores ou dados, em Inglês). Os contextos permitem a adição de tais camadas de dados, que podem ser associadas às projeções. Por exemplo, o componente `GridValueLayer` adiciona uma camada de dados às projeções do tipo `Grid`, de tal forma que cada agente, além de poder saber a coordenada da sua posição atual, pode saber o valor instantâneo de uma camada de dados associado a sua posição atual. Para mais detalhes, veja a Seção 3.3.6;

No modelo, existem seis contextos, utilizados para definir as diversas zonas e níveis de simulação, baseadas no modelo apresentado por Folcik *et al.* (2007). Estes contextos podem ser agrupados em 3 tipos:

Contexto principal É o contexto-pai, responsável por gerenciar os demais contextos. Representa, de forma muito abstrata, o organismo;

Zonas Reais São contextos que representam as zonas propriamente ditas, e que são apresentados visualmente durante a simulação;

Zonas Virtuais São contextos que representam zonas que têm uma função cuja visualização não é relevante, não tem significado, ou simplesmente não pode ser exibida. Por isto, estes contextos não possuem visualização durante a simulação.

Os contextos são classes Java, que herdam da classe `DefaultContext` integrante do *framework*.

A classe responsável por implementar o contexto principal é chamada de `Global`. O contexto principal também tem a tarefa de criar os demais contextos, por isso, além de herdar da classe supracitada, ele implementa a interface `ContextBuilder`, cujo método `build` é responsável por construir o modelo assim que a simulação é iniciada.

Os demais contextos (as zonas) são também classes Java que herdam de uma classe abstrata auxiliar, chamada `Environment`, que por sua vez estende a classe `DefaultContext`.

No total então existe um contexto principal, três contextos de zonas reais, e 2 contextos de zonas virtuais. Tais zonas serão discutidas na seção (Seção 3.3.4).

3.3.3.1 Parâmetros

Como já tido anteriormente, os contextos possuem uma funcionalidade que permite ao modelo utilizar parâmetros. Tal funcionalidade aumenta muito a usabilidade da fer-

Tabela 3.1. Parâmetros do modelo

Parâmetros			
Rótulo	Identificador	Tipo	Valor Padrão
Affinity Threshold	affinityThreshold	INTEGER	5
Antibody Lifetime	antibodyLifetime	INTEGER	350
Antibody Self Pattern	antibodySelfPattern	STRING	01100110
BCell Antibody Count	bCellAntibodyCount	INTEGER	40
BCell CK1 Memory Threshold	bCellCK1MemoryThreshold	FLOAT	0,02
BCell Germinal Probability	bCellGerminalProbability	INTEGER	60
BCell Lifetime	bCellLifetime	INTEGER	400
BCell MK1 Memory Threshold	bCellMK1MemoryThreshold	FLOAT	0,02
BCell Self Pattern	bCellSelfPattern	STRING	01100110
Circulation Height	circulationHeight	INTEGER	250
Circulation Width	circulationWidth	INTEGER	250
CitokineValue	citokineValue	FLOAT	100,00
CTL CK1 Memory Threshold	cTLCK1MemoryThreshold	FLOAT	0,20
CTL Lymphnode CK1 Duration	cTLLymphnodeCK1Duration	INTEGER	100
CTL Lymphnode Lifetime	cTLLymphnodeLifetime	INTEGER	500
CTL PK1 Memory Threshold	cTLPK1MemoryThreshold	FLOAT	0,20
CTL Proliferation Count	cTLProliferationCount	INTEGER	40
CTL Self Pattern	cTLSelfPattern	STRING	01110111
CTL Tissue CK1 Duration	cTLTissueCK1Duration	INTEGER	50
CTL Tissue Lifetime	cTLTissueLifetime	INTEGER	300
Dendritic Cells Count	dendriticCellsCount	INTEGER	200
Dendritic Lymphnode Lifetime	dendriticLymphnodeLifetime	INTEGER	200
Dendritic Lymphnode MK1 Duration	dendriticLymphnodeMK1Duration	INTEGER	20
Dendritic MK1 Activation Threshold	dendriticMK1ActivationThreshold	FLOAT	0,50
Dendritic PK1 Activation Threshold	dendriticPK1ActivationThreshold	FLOAT	0,50
Dendritic Self Pattern	dendriticSelfPattern	STRING	01010101
Dendritic Tissue MK1 Duration	dendriticTissueMK1Duration	INTEGER	30
Difusion Constant	difusionConstant	FLOAT	0,50
Evaporation Rate	evaporationRate	FLOAT	0,92
Lymphnode Height	lymphnodeHeight	INTEGER	50
Lymphnode Width	lymphnodeWidth	INTEGER	50
Macrophage Lifetime	macrophageLifetime	INTEGER	20
Macrophage MK1 Duration	macrophageMK1Duration	INTEGER	20
Macrophage Self Pattern	macrophageSelfPattern	STRING	00010001
NK CK1 Duration	nKCK1Duration	INTEGER	10
NK Kill Limit	nKKillLimit	INTEGER	2
NK Lifetime	nKLifetime	INTEGER	30
NK NoKill Timeout	nKNoKillTimeout	INTEGER	20
NK Self Pattern	nKSelfPattern	STRING	00110011
PC Self Pattern	pcSelfPattern	STRING	01000100
PC Stress Max Time	pCStressMaxTime	INTEGER	25
PC Stress Threshold	pCStressThreshold	FLOAT	5,00
Portal Macrophage To Send	portalMacrophageToSend	INTEGER	3
Portal MK1 Threshold	portalMK1Threshold	FLOAT	1,00
Portal NK To Send	portalNKToSend	INTEGER	3
Portal PK1 Threshold	portalPK1Threshold	FLOAT	1,00
Receptor Pattern Lenght	receptorPatternLenght	INTEGER	5
Self Pattern Lenght	selfPatternLenght	INTEGER	8
Th CK1 Duration	thCK1Duration	INTEGER	50
Th CK1 Memory Threshold	thCK1MemoryThreshold	FLOAT	1,00
Th Lifetime	thLifetime	INTEGER	500
Th Memory Proliferation Count	thMemoryProliferationCount	INTEGER	10
Th Proliferation Count	thProliferationCount	INTEGER	20
Th Self Pattern	thSelfPattern	STRING	10001000
Tissue Height	tissueHeight	INTEGER	110
Tissue Quarter	tissueQuarter	INTEGER	9
Tissue Width	tissueWidth	INTEGER	110
Virus Initial	virusInitial	INTEGER	100
Virus Latency	virusLatency	INTEGER	10
Virus Self Pattern	virusSelfPattern	STRING	00111100
Virus Target Pattern	virusTargetPattern	STRING	01000100
Virus Virulency	virusVirulency	INTEGER	2

ramenta uma vez que permite que o usuário altere parâmetros do modelo através da interface gráfica e sem a necessidade de alterar o código-fonte da aplicação.

Quando um parâmetro é criado, ele é exibido na interface gráfica e pode ter seu valor ajustado antes de se iniciar a simulação. Por outro lado, no código-fonte da aplicação, é possível recuperar o valor desses parâmetros facilmente para utilizá-lo na implementação do modelo. Esses parâmetros são definidos por um rótulo, que irá identificá-lo na interface gráfica, um identificador, que irá identificá-lo no código-fonte, um tipo, que irá definir o tipo de dados que ele armazena, e um valor padrão, que será utilizado caso o usuário não especifique um outro valor.

A Tabela 3.1 contém a lista de todos os parâmetros utilizados no modelo, assim como seus valores padrões.

3.3.4 Zonas

Seguindo o conceito de zonas, descrito por Folcik *et al.* (2007), o modelo apresentado por este trabalho também apresenta 5 zonas distintas. No *framework*, as zonas são definidas como um contexto, com suas variáveis, uma projeção do tipo Grid, e quantas camadas de dados forem necessárias para representar as substâncias presentes nela (para mais detalhes, veja a Seção 3.3.6).

Em relação ao modelo, as zonas representam áreas do organismo onde as células irão desempenhar os seus papéis. Cada zona possui um tamanho específico, os agentes que podem estar nela, assim como um conjunto inicial de agentes; as substâncias que podem circular por ela, *etc.* O tamanho de cada zona é definido empiricamente como o menor possível, para não sacrificar o desempenho do sistema, mas que não comprometa a simulação de suas funções e a interação dos agentes.

As zonas definidas no modelo são: Tissue, Lymphnode, Circulation, BoneMarrow e Thymus. A seguir, estas zonas serão descritas em maiores detalhes.

3.3.4.1 Tissue

A zona *Tissue* é implementada pela classe `Tissue`. Ela é baseada na “Zona 1”, definida por Folcik *et al.* (2007), e representa uma fatia microscópica de um tecido parenquimatoso genérico (FOLCIK *et al.*, 2007), onde as células do organismo têm seu primeiro contato com o patógeno, e onde também ocorre a infecção, os possíveis danos da mesma, *etc.* É a zona central do modelo.

Os principais agentes (ou células) presentes nesta zona são as células parenquimatosas,

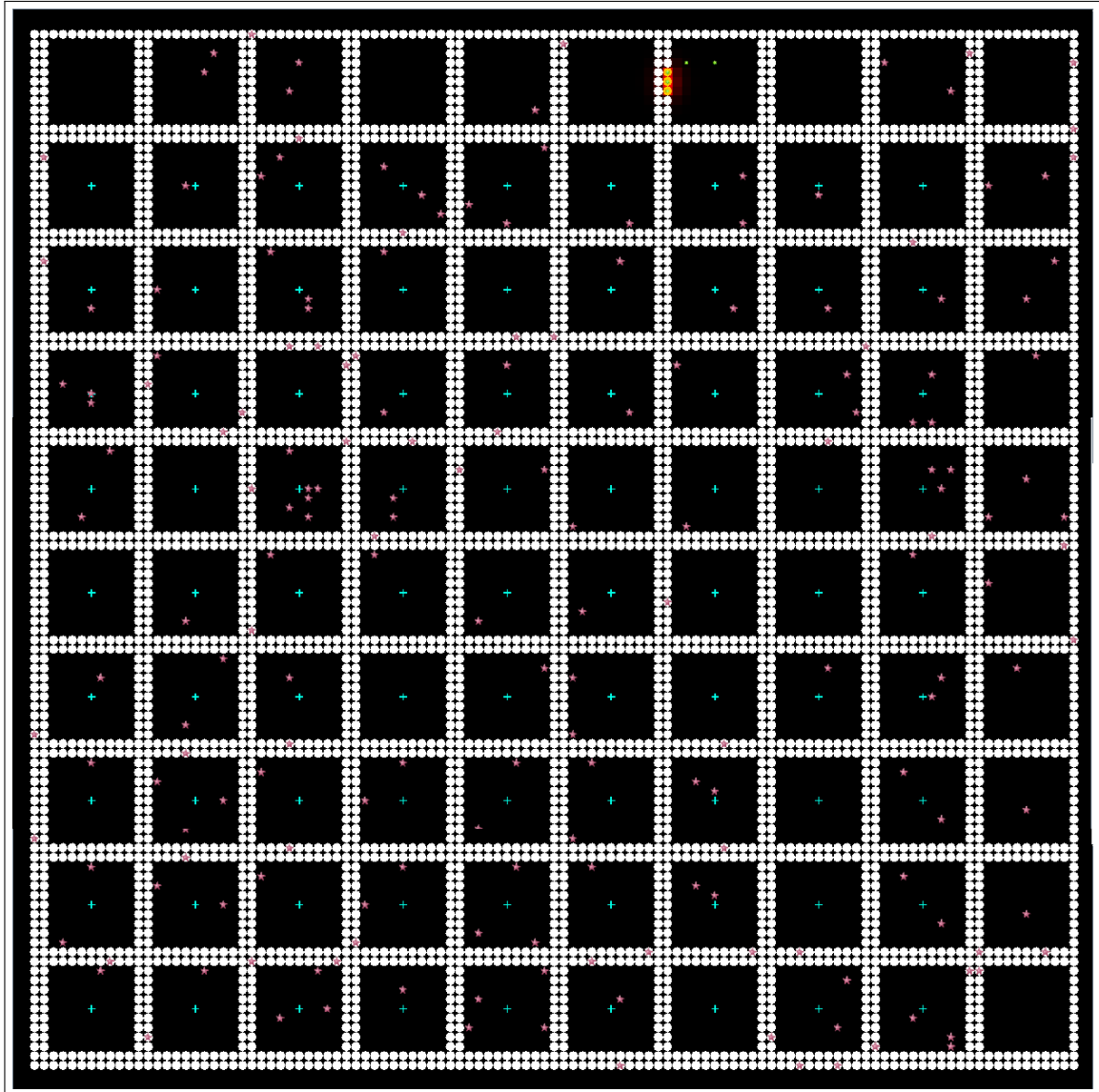


Figura 3.4. Imagem da zona **Tissue**, implementada.

representadas pelos agentes PC (Seção 3.4.4.7). Essas células formam uma estrutura de quarteirões que preenche todo o espaço da zona, representando a organização do tecido, como em um órgão. Essa estrutura está ilustrada na Figura 3.4.

Suas dimensões são definidas pelos parâmetros do contexto `tissueWidth` e `tissueHeight`.

Segundo um dado de Folcik *et al.* (2007), se considerarmos que uma célula possui $0.01mm$ de diâmetro em média e supondo que os parâmetros que definem o tamanho da zona *Tissue* estejam ajustados para o seu valor padrão que é de 110 unidades, essa zona representaria uma área de tecido de cerca de $1.21mm^2$.

3.3.4.2 *Lymphnode*

A zona *Lymphnode* é implementada pela classe *Lymphnode*. Ela é baseada na “Zona 2”, definida por Folcik *et al.* (2007), e representa o linfonodo. Segundo Folcik *et al.* (2007), esta zona pode representar também o baço. É nessa zona onde localizam-se inicialmente os linfócitos, onde os antígenos lhes são apresentados e onde eles se proliferam, uma vez que tenham reconhecido o antígeno.

Dessa forma, os principais agentes (ou células) presentes nesta zona são os linfócitos T *helper* (representados pelos agentes *ThCell*, Seção 3.4.4.9), os linfócitos T citolíticos (representados pelos agentes *CTL*, Seção 3.4.4.10) e os linfócitos B (representados pelos agentes *BCell*, Seção 3.4.4.14).

Suas dimensões são definidas pelos parâmetros do contexto *lymphnodeWidth* e *lymphnodeHeight*. Esses parâmetros têm o valor padrão de 50 unidades.

3.3.4.3 *Circulation*

A zona *Circulation* é implementada pela classe *Circulation*. Ela é baseada na “Zona 3”, definida por Folcik *et al.* (2007), e representa a circulação linfática e sanguínea. Segundo Folcik *et al.* (2007), a principal função dessa zona é representar a circulação de células que viajam do *Lymphnode* para o *Tissue*, por um período não conhecido e aleatório de tempo.

Dessa maneira, os principais agentes desta zona são os linfócitos que, vindos do *Lymphnode*, passam por ela até chegar ao *Tissue*.

Suas dimensões são definidas pelos parâmetros do contexto *circulationWidth* e *circulationHeight*. Esses parâmetros têm o valor padrão de 250 unidades.

3.3.4.4 *Bone Marrow*

A zona *Bone Marrow* é implementada pela classe *BoneMarrow*. Essa é uma nova zona, que representa uma medula óssea abstrata. Criada pela necessidade de se representar os mecanismos relacionados à tolerância central das células B, uma vez que os agentes agora têm uma representação de especificidade, esta zona é responsável por gerar os agentes que representam os linfócitos B, com especificidades aleatórias e testadas através dos mecanismos da tolerância central do linfócito B.

Tolerância Central do Linfócito B O funcionamento desse mecanismo está ilustrado na Figura 3.5. Toda vez que a geração de um linfócito B é necessária, a zona *BoneMarrow*

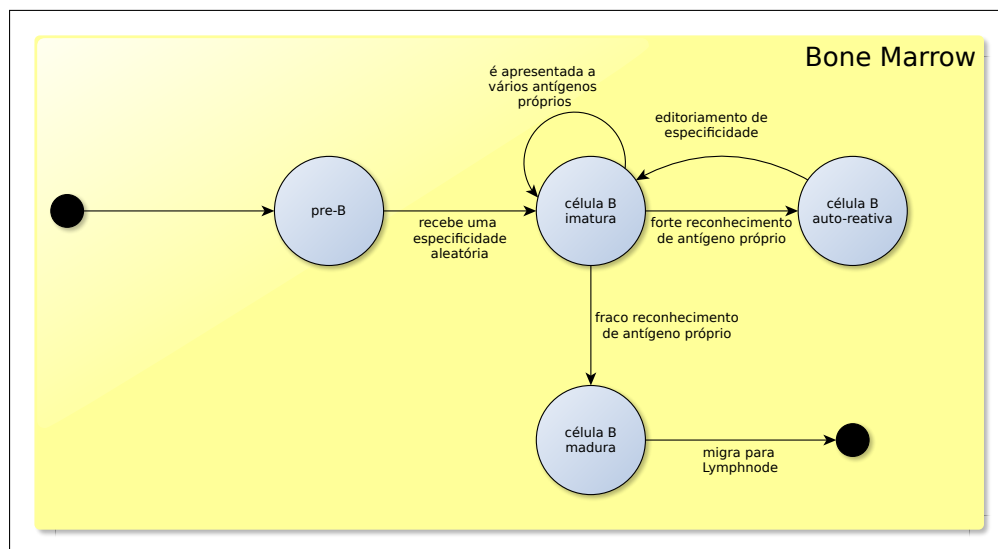


Figura 3.5. Tolerância Central do Linfócito B.

gera um linfócito pré-B, sem especificidade. Esse linfócito recebe uma especificidade aleatória, representando o mecanismo de **geração de diversidade**, passando a ser um linfócito B imaturo. Após isso, esse linfócito é exposto a autoantígenos, presentes no **banco de dados de antígenos próprios** (Seção 3.4.2): caso ele reconheça algum autoantígeno com alta afinidade, o linfócito sofre “editoriamento do receptor” (ABBAS; LICHTMAN, 2007), modificando seu receptor e retornando ao estado de linfócito B imaturo. Caso ele reconheça o antígeno próprio com baixa afinidade, ele passa pela seleção negativa e segue para os órgãos linfoides.

Esta zona é uma zona virtual, e portanto, não possui uma projeção e, conseqüentemente, não possui uma visualização, ou dimensões.

3.3.4.5 *Thymus*

A zona *Thymus* é implementada pela classe *Thymus*. Essa é uma nova zona, que representa o Timo, abstratamente. Criada pela necessidade de se representar os mecanismos relacionados à tolerância central das células T, uma vez que os agentes agora têm uma representação de especificidade, esta zona é responsável por gerar os agentes que representam os linfócitos T *helper* (*ThCell*) e T citolítico (CTL), com especificidades aleatórias e testadas através dos mecanismos da tolerância central do linfócito T.

Tolerância Central do Linfócito T O funcionamento desse mecanismo está ilustrado na Figura 3.6. Toda vez que a geração de um linfócito T *helper* ou T citolítico é necessária, a zona *Thymus* gera um linfócito pré-T, sem especificidade. Esse linfócito recebe então dois receptores TCR (o $CD4^+$ e o $CD8^+$) e passa a ser um linfócito T duplo-positivo. A

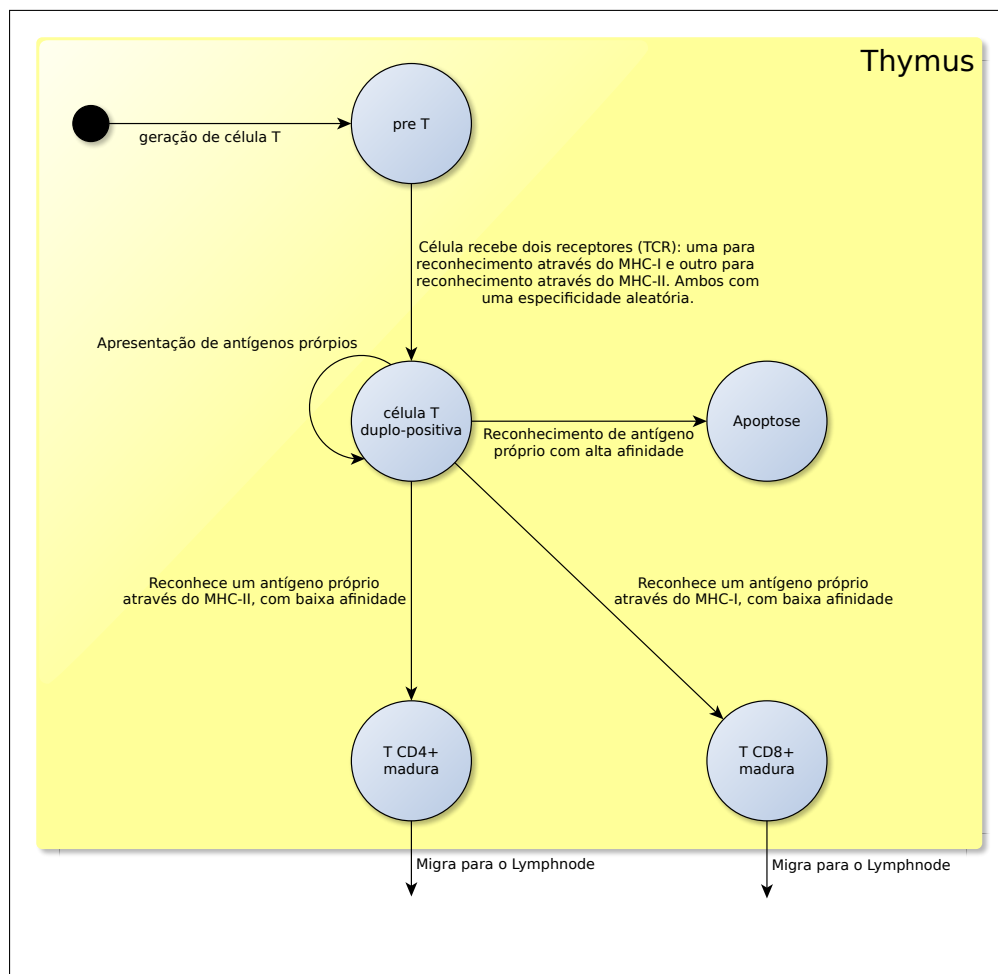


Figura 3.6. Tolerância Central do Linfócito T.

partir daí, vários autoantígenos, presentes no **banco de dados de antígenos próprios** (Seção 3.4.2), são apresentados a esse linfócito: caso ele reconheça o autoantígeno com alta afinidade, ele é eliminado. Caso ele reconheça o autoantígeno, com baixa afinidade, através do MHC-I, ele se diferencia em um linfócito T citolítico. Já se ele reconhecer o autoantígeno, com baixa afinidade, através do MHC-II, ele se diferencia em um linfócito T *helper*. Em ambos os últimos casos, o linfócito migra para o Lymphnode.

Esta zona, assim como a BoneMarrow, é uma zona virtual, e portanto, não possui uma projeção e, conseqüentemente, não possui uma visualização, ou dimensões.

3.3.5 Migração entre zonas

Como existem 5 zonas onde há agentes, é de se esperar que eles, em algum momento, tenham a necessidade de migrar de uma zona para outra. Por exemplo, os linfócitos recém criados nas zonas BoneMarrow e Thymus devem migrar para a zona Lymphnode. Assim como os linfócitos ativados do Lymphnode devem migrar para a zona Circulation, e

mais tarde, para a zona `Tissue`. E por fim, as células dendríticas (`Dendritic`, Seção 3.4.4.13) que, após capturarem antígenos na zona `Tissue`, (e mediante a detecção de *danger signals*), tendem a migrar para o `Lymphnode` afim de apresentá-los aos linfócitos lá contidos.

Essa migração é realizada e controlada por agentes especiais, chamados `Portals`, baseados nos agentes “*Portals*” descritos por Folcik *et al.* (2007). Segundo Folcik *et al.* (2007), esses agentes representam vasos linfáticos e sanguíneos responsáveis pela circulação das células e substâncias entre as zonas.

Cada zona possui seus próprios portais, que são conectados a portais em outras zonas. Por exemplo, a zona `Tissue` possui os seus portais, implementados pela classe `TissuePortal` que estende a classe `Portal`. Os portais do tipo `TissuePortal` podem estar conectados a portais do tipo `CirculationPortal`, por exemplo. Dessa forma, quando um agente alcança um `CirculationPortal`, e este agente atende aos requisitos para passar por ele até a zona `Tissue`, o agente é transportado para a localização de um `TissuePortal` dentro da zona `Tissue`, simulando a passagem por um vaso sanguíneo.

Assim também acontece com as substâncias que são liberadas em uma zona: quando atingem um portal, este notifica o portal que representa a sua outra extremidade em outra zona, que passa a emitir a mesma substância lá, simulando assim a passagem da substância pelo vaso.

Maiores detalhes sobre a migração serão discutidos na Seção 3.4.4.5 (*Portal*).

3.3.6 Difusão de Substâncias

Como visto na Seção 2.1, toda a comunicação entre as células é realizada através de mediadores químicos, chamados de citocinas. Essas substâncias, quando liberadas pelas células, sofrem um processo de difusão, espalhando-se pelos arredores do local onde foram liberadas, diminuindo sua concentração; e sofrem também um processo de decaimento, diminuindo sua quantidade com o tempo.

Para simular esse tipo de fenômeno, o *framework* provê um componente denominado `ValueLayer`. Este componente é uma interface, sendo implementada por diversos outros, dentre eles, o `GridValueLayer`³.

O `GridValueLayer` nada mais é do que uma camada abstrata de dados que, no momento de sua criação, é associada a uma projeção do tipo `Grid`. É possível associar várias camadas de dados à mesma projeção. Dessa forma, um agente pode saber qual a

³Para maiores detalhes sobre `GridValueLayer`, consulte a documentação do Repast.

concentração de dada substância, naquele instante de tempo, na posição (i, j) . Para cada substância, o agente irá selecionar os dados da camada específica para ela.

Para associar uma camada de dados a uma zona, basta invocar o método do modelo denominado `createCitokineLayer`. Para liberar uma certa quantidade de citocina em uma posição (i, j) basta invocar o método `releaseCitokine`. E para recuperar o valor da citocina em uma dada substância, basta invocar o método `getCitokineValue`.

Todo `GridValueLayer` tem associado a ele um componente denominado `ValueLayerDiffuser`⁴. Esse componente é responsável por simular a difusão e decaimento da substância, formando assim um gradiente de concentração. A cada *tick*, todas as camadas de dados têm seus valores atualizados.

A difusão das substâncias é calculada de acordo com a Equação (3.2).

$$V_{novo} = E * (V_{atual} + D * (\overline{V_{vizinhos}} - V_{atual})) \quad (3.2)$$

Onde:

E : constante de decaimento;

D : constante de difusão;

V_{novo} : valor da concentração atualizado na posição atual;

V_{atual} : valor da concentração antigo (do *tick* anterior) na posição atual;

$V_{vizinhos}$: valor da concentração antigo (do *tick* anterior) nas posições da Vizinhança de Moore, ao redor da posição atual.

Assim, os cálculos são feitos para todas as posições, e seus resultados são guardados em uma matriz auxiliar. Assim que todas as posições tiverem seus novos valores calculados na matriz auxiliar, os dados são copiados para a camada de dados, sobrescrevendo os valores antigos. Somente quando todas as camadas tiverem seus valores atualizados (e os agentes executarem suas tarefas agendadas) é que o próximo *tick* ocorre.

A Figura 3.7 ilustra como as camadas de dados são utilizadas para simular a difusão das substâncias nas zonas. Em 3.7(a) está ilustrado o gradiente contínuo, como é no mundo real e suas concentrações médias em cada célula. Pode-se notar que a liberação dessa substância ocorreu na célula onde sua concentração é igual a 10 unidades. Já em 3.7(b) é mostrado a discretização do gradiente, mostrando como os agentes veem a distribuição da substância.

⁴Para maiores detalhes sobre `ValueLayerDiffuser`, consulte a documentação do Repast.

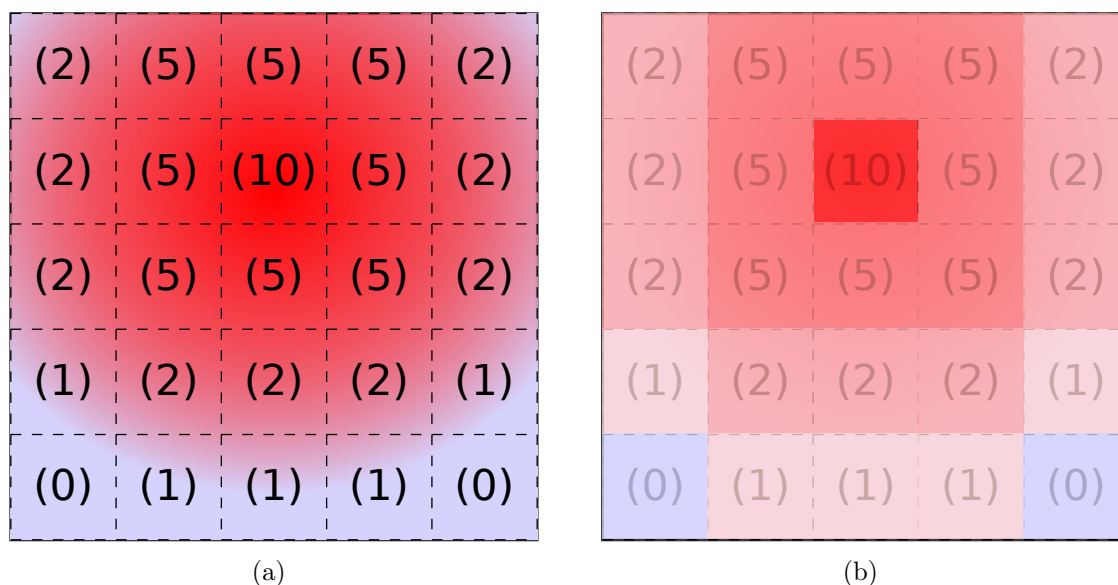


Figura 3.7. Ilustração da representação do gradiente de dispersão das citocinas através das camadas de dados. Em (a), a distribuição contínua da substância, como acontece no mundo real. Em (b), a discretização do gradiente, mostrando como os agentes o veem.

3.3.6.1 Citocinas

As substâncias que foram incluídas no modelo são representações abstratas das substâncias existentes no mundo real, e são baseadas em um sub-conjunto das substâncias existentes no modelo descrito por Folcik *et al.* (2007).

A regra geral para definir as substâncias no modelo, como descrito por Folcik *et al.* (2007), foi criar substâncias abstratas que representem as substâncias reais agrupadas por funcionalidade. As funcionalidades, por sua vez, foram abstraídas por Folcik *et al.* (2007) em duas funcionalidades abstratas básicas: as substâncias que amplificam a resposta inflamatória (pró-inflamatórias) e as substâncias que suprimem essa resposta (anti-inflamatórias).

O autor de Folcik *et al.* (2007) adotou uma regra de nomenclatura nas substâncias que diz que, quando um nome de substância termina com o numeral 1, então ela é uma substância pró-inflamatória; já se o nome da substância termina com o numeral 2, então trata-se de uma substância anti-inflamatória. Por achar que essa definição ficou simples e objetiva, optou-se por utilizar a mesma regra.

Desta forma, será descrito agora quais substâncias estão presentes no modelo, e quais substâncias reais elas representam, segundo o modelo descrito por Folcik *et al.* (2007).

As substâncias pró-inflamatórias incluídas no modelo são:

PK1 Baseada na substância “*Parenchymal-kine 1*” (PK1), descrita por Folcik *et al.* (2007), representa fatores de estresse liberados por tecidos que estão sofrendo danos em razão da infecção ou da resposta imunológica. Segundo Folcik *et al.* (2007), essa substância representa fatores de estresse como as proteínas de choque-térmico (HSP), ácido úrico, *etc*; além de quimiocinas como CX3CL1, CCL3, CCL5 e CCL6 (FOLCIK *et al.*, 2007);

MK1 Baseada na substância “*Mono-kine 1*” (MK1), descrita por Folcik *et al.* (2007). representa o conjunto de substâncias pró-inflamatórias presentes nas respostas imunológicas inatas. Segundo Folcik *et al.* (2007), essa substância representa as substâncias IL-12, IL-8, CCL3, CCL4, CCL5, CXCL9, CXCL10 e CXCL11 (FOLCIK *et al.*, 2007);

CK1 Baseada na substância “*Cytokine 1*” (CK1), descrita por Folcik *et al.* (2007), representa o conjunto de substâncias pró-inflamatórias presentes nas respostas imunológicas adaptativas. Segundo Folcik *et al.* (2007), essa substância representa as substâncias INF- γ , IL-2 e TNF- β (FOLCIK *et al.*, 2007);

NECROSIS Baseada na substância “*Necrosis factors*”, descrita por Folcik *et al.* (2007), representa os restos mortais de células que sofreram necrose, *i.e.*, morreram por algum processo traumático, e não por morte celular programada (FOLCIK *et al.*, 2007).

Já as substâncias anti-inflamatórias incluídas no modelo são:

MK2 Baseada na substância *Mono-kine 2* (MK2), descrita por Folcik *et al.* (2007), representa o conjunto de substâncias anti-inflamatórias presentes nas respostas imunológicas inatas. Segundo Folcik *et al.* (2007), essa substância representa as substâncias IL-10, CCL1, CCL17, CCL22, CCL11, CCL24, CCL26 (FOLCIK *et al.*, 2007);

CK2 Baseada na substância “*Cytokine 2*” (CK2), descrita por Folcik *et al.* (2007), representa o conjunto de substâncias anti-inflamatórias presentes nas respostas imunológicas adaptativas. Segundo Folcik *et al.* (2007), essa substância representa as substâncias TGF- β , IL-4, IL-5, IL-6, IL-10, IL-13 e INF- γ (FOLCIK *et al.*, 2007);

APOPTOSIS Baseada na substância “*Apoptotic bodies*”, descrita por Folcik *et al.* (2007), representa os restos mortais de células que sofreram morte celular programada, ou apoptose (FOLCIK *et al.*, 2007).

As demais substâncias, incluídas no modelo descrito por Folcik *et al.* (2007), não foram incluídas neste modelo, pois, primeiro, representam substâncias que, devido ao nível de

granularidade (Seção 3.4.1) escolhido por este trabalho, serão representadas por agentes; e, segundo, por representarem substâncias que não serão utilizadas no modelo.

Além disso, como a definição das substâncias foi feita antes da definição dos agentes, algumas substâncias, embora modeladas, não são utilizadas por nenhum agente. Contudo, por não serem utilizadas, não apresentam custo computacional e, portanto, serão deixadas no modelo (e na implementação) para facilitar trabalhos futuros.

3.4 Agentes

Nesta seção serão descritos os agentes do modelo e suas funções. Por questão de praticidade, os agentes serão descritos enquanto classes do sistema, para facilitar o entendimento, por representar algo mais tangível ao desenvolvedor que por ventura estiver utilizando este trabalho.

3.4.1 Granularidade

O nível de granularidade diz respeito a quão profundo será a abstração do modelo. Quanto mais fina a granularidade, mais detalhes serão modelados, e mais complexo será o modelo. Quando mais grossa, mais detalhes terão que ser abstraídos, e, conseqüentemente, corre-se o risco de deixar algo valioso de fora.

Por isso, enquanto o objetivo é modelar o máximo possível (granularidade mais fina o possível), é necessário que se alcance o nível de granularidade mais grosso possível, por questões de custo computacional, mas que ainda gere um modelo que consiga reproduzir o comportamento desejado, para que se possa realizar a simulação do fenômeno que se deseja estudar (FOLCIK *et al.*, 2007);

Como o objetivo deste trabalho é criar um modelo capaz de simular os mecanismos envolvidos na autoimunidade, é fundamental que os elementos que estejam envolvidos com especificidade, afinidade e tolerância estejam presentes no modelo.

O modelo descrito por Folcik *et al.* (2007) utilizou o nível de granularidade “*cell-as-agent*”, *i.e.*, toda célula será representada por um agente. Os patógenos (vírus, no caso) são modelados por Folcik *et al.* (2007) como substâncias, assim como os anticorpos. No entanto, este nível de granularidade não permitiria que o modelo apresentado neste trabalho tivesse, por exemplo, patógenos com especificidades diversas (para isso seria necessário uma camada de substância para cada especificidade desejada), ou anticorpos específicos para um patógeno em particular, entre outras limitações.

Por este motivo, o nível de granularidade escolhido foi o “*antigen-as-agent*” ou “*hasSpecificity-as-agent*”, *i.e.*, tudo aquilo que estiver incluído no modelo e for um antígeno, ou contiver antígenos, será modelado como um agente. Em outras palavras, todo componente que puder ser reconhecido por um PRR será considerado um agente.

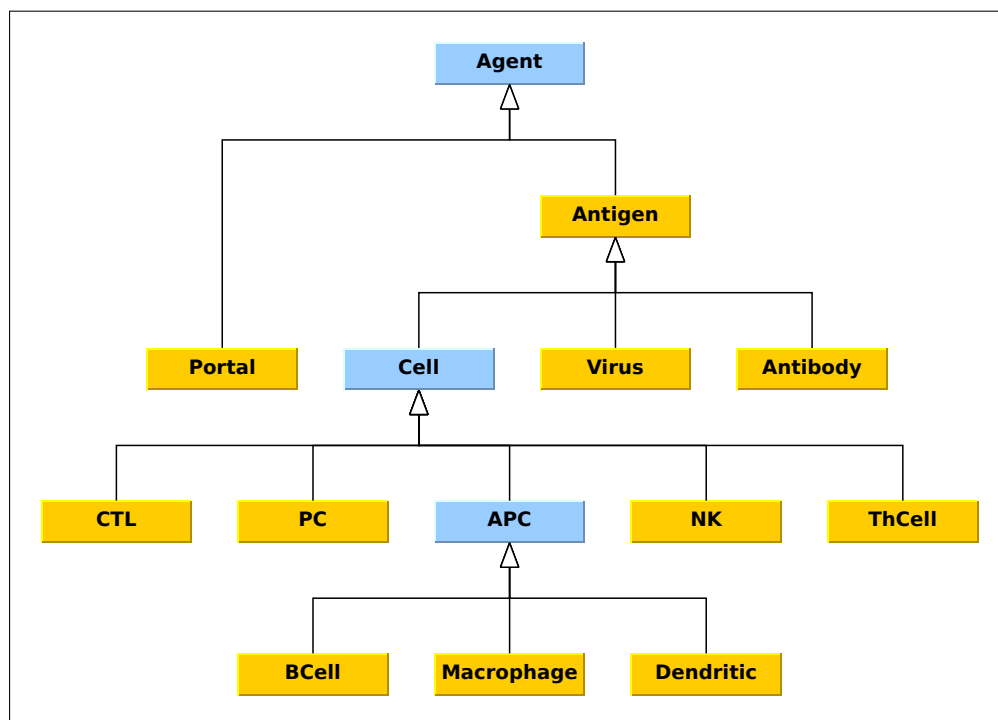


Figura 3.8. Classes de Agentes e sua organização resultante do nível de granularidade escolhido

As demais substâncias, como as citocinas, e outras moléculas cuja especificidade não seja relevante, foram modeladas como substâncias, utilizando a camada de dados, descrita na Seção 3.3.6.

A Figura 3.8 apresenta as classes representando quais elementos foram modelados como agentes, e sua organização, como resultado do nível de granularidade escolhido. Em azul, estão as classes abstratas, e em amarelo, as classes concretas. Mais detalhes sobre os diferentes tipos de agentes estão descritos na Seção 3.4.4.

É válido ressaltar que o fato do modelo apresentado por Folcik *et al.* (2007) não representar, por exemplo, vírus de várias especificidades distintas, não tira o mérito do excelente trabalho apresentado pelo autor, nem torna esse trabalho melhor ou pior que outros. Trata-se apenas de dois modelos, com focos diferentes, e que por isso escolheram o nível de granularidade que lhes era mais conveniente.

3.4.2 Especificidade

O termo especificidade refere-se ao fato das respostas imunológica adaptativas serem específicas para o patógeno invasor. Mais ainda, refere-se ao fato dos linfócitos serem específicos para antígenos distintos, *i.e.*, têm receptores específicos para diferentes antígenos. Dessa forma, assume-se no modelo que especificidade refere-se ao fato dos agentes terem um padrão (um antígeno) como alvo de suas ações, e como gatilho para sua ativação.

Assim, temos dois conceitos bem definidos no modelo:

Antígeno é o padrão que identifica qualquer instancia da classe `Antigen`, ou de uma classe que estenda a classe `Antigen`;

Especificidade é o padrão dos receptores dos linfócitos e dos anticorpos, o padrão para o qual eles são específicos, ou o padrão que os ativa.

Neste ponto, temos um conceito ainda a definir: o padrão. No mundo real, esse padrão é o padrão molecular, sua estrutura e a sua sequência de aminoácidos.

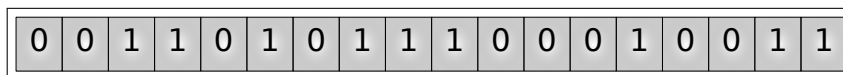


Figura 3.9. Ilustração de uma Sequência de *Bits*

Já no modelo, será utilizado um padrão abstrato, como sugerido por Floreano e Mattiussi (2008), representado por uma estrutura formada por uma sequência de símbolos binários 0 e 1 (*bits*), conhecida como *string* de *bits* (Figura 3.9). Essa estrutura é implementada pela classe `Pattern`. Todo agente que descende da classe `Antigen` possui um padrão próprio, uma instância da classe `Pattern`, chamado de *self*. Da mesma forma, os linfócitos têm um `Pattern` que representa a sua especificidade, chamado de *target*.

Antígenos Próprios Todo agente que descende da classe `Cell`, quando criado, adiciona o seu padrão *self* a um banco de dados de antígeno próprios, chamando de conjunto `SELF`. É este conjunto que é apresentado aos linfócitos durante a tolerância central.

3.4.3 Afinidade

Como visto na Seção 2.1, a afinidade é a propriedade que define a tendência entre as substâncias de interagirem. Mais especificamente, na imunologia, a afinidade é a força com a qual um antígeno é reconhecido por um receptor. Da mesma forma, no modelo,

a afinidade é definida com o grau de similaridade entre os padrões apresentados pelo antígeno e pelo seu receptor: quanto maior esse grau, maior será a afinidade.

Como visto na Seção 3.4.2, será utilizada uma sequência de bits para representar os padrões moleculares. Mas como calcular o grau de afinidade entre dois padrões formados por sequências de bits?

Para isso será utilizado o método sugerido por Floreano e Mattiussi (2008), chamado de “comprimento da maior subsequência comum”, em uma tradução livre de *longest common substring length*, cujo objetivo é calcular, dados dois padrões de sequências de bits A e B , o tamanho da maior subsequência de símbolos contíguos que esteja contida em A e em B simultaneamente, na mesma ordem.

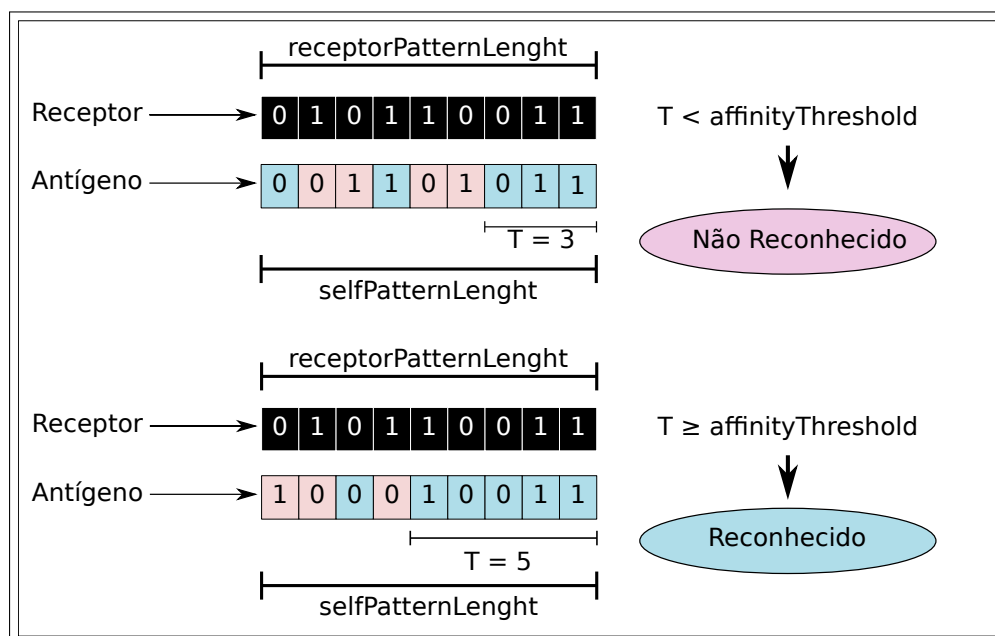


Figura 3.10. Ilustração do cálculo do Grau de Afinidade. Os símbolos da sequência de bits do antígeno em azul indicam que eles são iguais aos símbolos da sequência de bits do receptor de mesma posição. Os símbolos vermelhos indicam que são diferentes. A maior subsequência de símbolos comuns pode ser definida, na ilustração, como a maior sequência de símbolos azuis contíguos. As sequências não precisam, necessariamente, estarem alinhadas. No exemplo, supõe-se que o parâmetro `affinityThreshold` seja igual a 4.

Fonte (adaptado): (FLOREANO; MATTIUSSI, 2008, p. 381)

O grau de afinidade entre dois padrões então é definido como o comprimento da maior subsequência comum aos dois padrões. Para tanto, foi necessário a criação de três novos parâmetros no modelo: o primeiro é o `affinityThreshold`, que define o comprimento mínimo que precisa ser superado para que seja considerado que os dois padrões se reconhecem. O segundo parâmetro é o `receptorPatternLength`, que define o comprimento padrão das sequências de bits usadas para representar a especificidade dos receptores. E o terceiro parâmetro é o `selfPatternLength`, que define o comprimento padrão

das sequências de *bits* usadas para representar os antígenos. A Figura 3.10 ilustra o funcionamento do mecanismo de cálculo do grau de afinidade.

Como já dito, esses padrões estão em todos os agentes que estendem a classe `Antigen`, representando o fato de todos esses agentes poderem ser reconhecidos por algum receptor.

Já os agentes que estendem a classe `Cell` possuem o método `MHCI`, capaz de apresentar o padrão *self* do agente, e, no caso de células infectadas, o padrão do patógeno que as infectou. Esse método representa o complexo de histocompatibilidade principal classe I, descrito na Seção 2.1.

Os agentes que estendem a classe `APC`, por sua vez, possuem o método `MHCII`, capaz de apresentar o padrão *self* de agentes que foram fagocitados. Esse método representa o complexo de histocompatibilidade principal classe II, descrito também na Seção 2.1.

Por outro lado, os linfócitos `CTL`, `ThCell` e `BCell` têm receptores e, conseqüentemente, uma especificidade. Eles utilizam o cálculo do grau de afinidade para detectar se seus receptores reconhecem ou não o antígeno que está sendo apresentado por um agente `APC`.

Outros agentes que utilizam o cálculo do grau de afinidade, mas responsáveis pela imunidade inata, são o `Macrophage` e o `NK`. Eles utilizam esse cálculo para detectarem substâncias reconhecidas como PAMPs, descritas na Seção 2.1, e para inspecionar o MHC-I das células, respectivamente. Já sobre as PAMPs, elas são representadas por um conjunto de padrões que foram reservados e incluídos em um banco de dados de antígenos de PAMPs, chamado de conjunto `PAMP`, um conjunto semelhante ao conjunto `SELF`.

3.4.4 Tipos de Agentes

Os diferentes tipos de agentes, oriundos do processo de modelagem e do nível de granularidade escolhido, estão ilustrados na Figura 3.8. Todos esses agentes são definidos como classes da linguagem Java. Como discutido na Seção 2.3.2, os agentes foram modelados como agentes reativos com estado, por ser o tipo de agente cujo comportamento mais se assemelha ao comportamento real das células no organismo vivo.

As regras de comportamento dos agentes são baseadas na literatura específica sobre imunologia, em especial, nos trabalhos Abbas e Lichtman (2007), Roitt e Delver (2001) e também no modelo definido por Folcik *et al.* (2007). Essas regras foram extraídas da literatura, e os comportamentos foram descritos em máquinas de estado finito, como sugerido por Wooldridge (2001), Weiß (1999), Folcik *et al.* (2007). Posteriormente, essas máquinas de estado foram implementadas na forma de código-fonte Java, criando-se assim as classes Java que representam os agentes.

Cada agente possui um método principal, chamado arbitrariamente (ou tradicionalmente) de `step`. Como já mencionado, este método possui anotações que agendam a sua execução a cada *tick* da simulação, simulando assim a ação dos agentes: a cada *tick*, o agente percebe o ambiente, e reage, segundo as suas regras pré-definidas.

A seguir, serão descritos em mais detalhes os agentes, suas classes e suas principais funções.

3.4.4.1 *Agent*

A classe abstrata `Agent` é a representação básica do agente, em sua forma mais simples. Todo agente precisa estender essa classe para ser aceito pelos contextos (ambientes).

Os principais atributos do agente são:

zone Atributo do tipo `Environment`, responsável por armazenar o ambiente (espaço, zona) na qual o agente está situado;

x Atributo do tipo `Integer`, responsável por armazenar a coordenada x relativa à posição do agente no espaço definido pela projeção;

y Atributo do tipo `Integer`, responsável por armazenar a coordenada y relativa à posição do agente no espaço definido pela projeção;

ticks Número de *ticks* de idade do agente. Muitos agentes têm um tempo de vida limitado, definido por parâmetros específicos. Os agentes utilizam esse atributo para saber sua idade;

Os principais métodos do agente são:

randomWalk Faz com que o agente ande aleatoriamente pelo espaço, sempre 1 unidade por *tick*, dentro da Vizinhança de Moore;

getNeighbors Método que retorna todos os agentes que estão na Vizinhança de Moore do agente;

getSpecificNeighbors Método que retorna todos os agentes, do tipo especificado nos parâmetros do método, que estão na Vizinhança de Moore do agente. Esse método utiliza, por padrão, uma Vizinhança de Moore de raio 1. No entanto, é possível especificar um raio diferente. Isso é necessário para representar a Vizinhança de Moore nos agentes `Dendritic`;

step Método abstrato, que deve ser implementado pelas classes que herdam da classe `Agent`. Responsável por executar o comportamento dos agentes, baseado em suas regras;

die Método responsável pela exclusão do agente do contexto, e conseqüentemente, do modelo.

3.4.4.2 *Antigen*

A classe `Antigen` é uma classe concreta, pois pensou-se na possibilidade de ser necessário representar antígenos livres na circulação. Sua principal característica é ter um atributo do tipo `Pattern`, chamado `self` (ou `antigen`), que representa o padrão molecular do agente.

Ele possui um método chamado `getPattern` que informa qual é o padrão antigênico de qualquer agente que estenda essa classe.

Toda classe que estenda a classe `Antigen` também representa um antígeno. É válido ressaltar que antígeno é qualquer estrutura que possa ser reconhecida por um PRR (Seção 2.1).

3.4.4.3 *Antibody*

A classe `Antibody` é uma classe concreta, que representa as moléculas de anticorpo. São criados pelos agentes `BCell`, quando ativados. Possui um atributo do tipo `Pattern`, chamado `target`, que representa a especificidade do anticorpo.

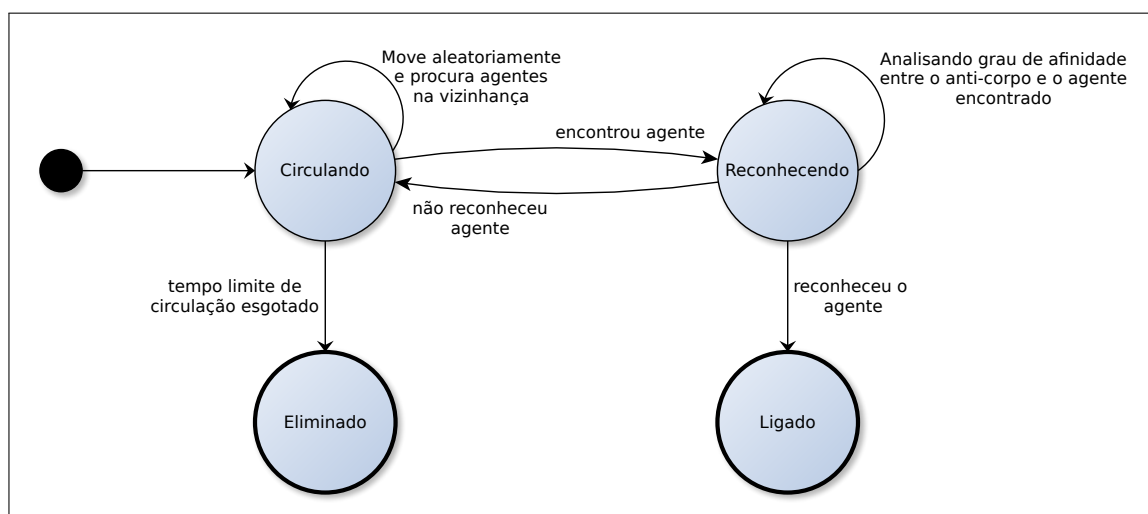


Figura 3.11. Regras do agente `Antibody`

Suas regras estão ilustradas na Figura 3.11.

3.4.4.4 *Virus*

A classe *Virus* é uma classe concreta, que representa os patógenos infecciosos que estão atacando o organismo. Ele possui um atributo do tipo *Pattern*, que representa o seu padrão antigênico. Além desse, possui outro atributo do tipo *Pattern*, também chamado de *target*, que define o padrão molecular para o qual o *Virus* têm afinidade, e irá tentar infectar.

Esse agente tem um comportamento similar a de um vírus real, e por isso recebeu esse nome: ele circula aleatoriamente, até encontrar uma célula hospedeira (que possua um padrão *self* reconhecido pelo padrão *target* do *Virus*). Quando a encontra, tenta infectá-la. Ele irá obter sucesso na infecção caso a célula já não esteja infectada por outro vírus. Essa regra foi criada apenas para não permitir que uma célula seja infectada por mais de um vírus, o que seria, no modelo, um desperdício de agentes.

O vírus que obtém sucesso em infectar a célula insere o seu padrão na mesma, que passa a apresentá-lo através do MHC I. Depois de infectada, e após um período de latência, a célula começa a produzir réplicas do *Virus*, com o mesmo padrão antigênico do *Virus* original. Em outros modelos, o padrão do *Virus* poderia mudar, se misturar com o da célula, mas neste modelo este comportamento não foi implementado.

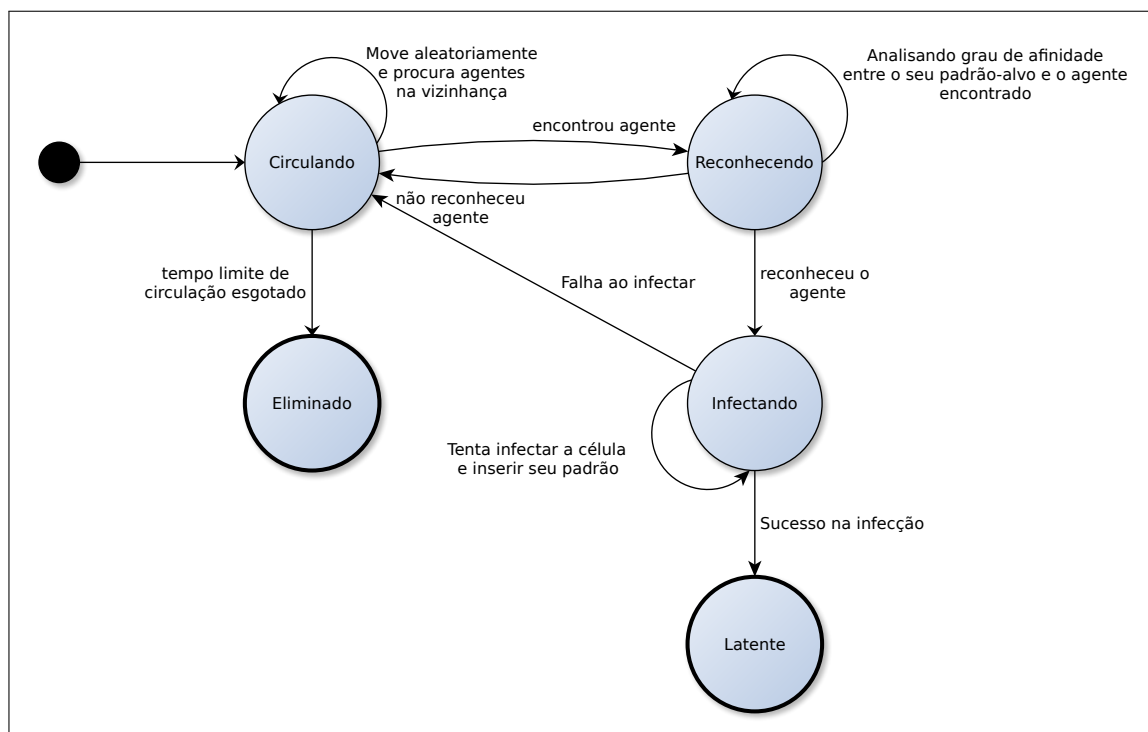


Figura 3.12. Regras do agente *Virus*

As regras do agente *Virus* estão ilustradas na Figura 3.12. Adicionalmente, o agente *Virus* possui um método chamado *ExecuteDNA*, que é executado uma vez que tenha

logrado sucesso em infectar uma célula.

3.4.4.5 *Portal*

Como descrito na Seção 3.3.5, os agentes do tipo `Portal` foram criados, baseados nos agentes `Portal` do modelo descrito por Folcik *et al.* (2007), para simular a migração dos agentes entre as zonas. Segundo Folcik *et al.* (2007), o `Portal` representa os vasos sanguíneos e linfáticos, e sua circulação.

A classe `Portal` é estendida por outras três classes, dependendo da zona em que o portal está situado. São elas:

TissuePortal: responsável por representar os vasos sanguíneos e linfáticos que irrigam o tecido. São conectados à portais presentes na zona `Circulation`. Esses portais também são responsáveis por criar os agentes `Macrophage` e `NK`, em resposta à detecção de níveis altos da citocina PK1 (sinal de estresse das células teciduais), representando a quimiotaxia dessas células em função da presença desses sinais químicos (FOLCIK *et al.*, 2007);

CirculationPortal: responsável por representar os vasos sanguíneos e linfáticos que capturam as células na circulação e às conduzem ao sítio de infecção. Alguns estão conectados aos portais da zona `Tissue`, e levam agentes para lá. Já outros, estão conectados aos portais da zona `Lymphnode`, e trazem os linfócitos de lá;

LymphnodePortal: responsável por representar os vasos sanguíneos e linfáticos que irrigam os linfonodos e o baço. (FOLCIK *et al.*, 2007) Estão conectados a portais da zona `Circulation`, para onde levam os linfócitos, quando ativados, e os anticorpos.

As únicas exceções são os agentes `Dendritic`, que são levados instantaneamente da zona `Tissue` para a zona `Lymphnode`, quando ativadas, justamente por ser o caminho contrário que todos os outros agentes fazem, *i.e.*, eles vão do `Tissue` para o `Lymphnode`, enquanto os demais vêm do `Lymphnode` para o `Tissue`. Assim como também os linfócitos, recém criados nas zonas `BoneMarrow` e `Thymus`, que não utilizam portais para migrarem para a zona `Lymphnode`, uma vez que essa migração acontece no primeiro *tick* de simulação, quando todos os linfócitos são gerados, não fazendo sentido utilizar a migração para isso.

3.4.4.6 *Cell*

A classe `Cell`, como o próprio nome já diz, é usada para modelar todos os agentes que representam células nucleadas. Dessa forma, todo agente que representa uma célula nucleada deve estender essa classe.

Sua principal tarefa é definir as funções utilitárias que todo agente, enquanto célula, deverá utilizar. Além disso, essa célula define o método `MHCI`, que representa a apresentação de peptídeos, sendo gerados no interior da célula, através do complexo de histocompatibilidade principal classe I, presente em todas as células nucleadas do organismo, como descrito na Seção 2.1.

Os principais métodos que essa classe define são:

`getCitokineValue`: este método retorna a quantidade de uma dada substância (uma citocina, por exemplo), na posição atual do agente;

`releaseCitokine`: este método libera uma quantidade de citocina pré-estabelecida por um parâmetro (`citokineValue`) do modelo (veja Seção 3.3.6) na posição atual do agente. Essa substância passa a sofrer difusão, de acordo com o que foi discutido também na Seção 3.3.6;

`followCitokineByGradient`: este método faz com que o agente (ou a célula, no caso) se mova na direção crescente do gradiente, *i.e.*, seguindo o gradiente de concentração no sentido em que a concentração da substância está aumentando;

`MHCI`: este método representa o complexo de histocompatibilidade principal classe I, informando os padrões que são produzidos pela célula, *i.e.*, o próprio padrão *self* da célula em questão, e o padrão do vírus, caso a célula esteja infectada;

`necrosis`: este método representa a morte da célula por processo traumático, *i.e.*, necrose. Executa ainda algumas tarefas como liberar a substância `NECROSIS`, que representa os restos mortais da célula. Este método não remove o agente da simulação;

`apoptosis`: este método representa a morte celular programada da célula, *i.e.*, apoptose. Após executar algumas tarefas, como liberar a substância `APOPTOSIS`, que representa a sinalização de uma morte celular programada, invoca o método `die` da classe `Agent`, removendo-o da simulação.

Todos os agentes que serão descritos a seguir derivam da classe `Cell`.

3.4.4.7 Parenchyma Cells

O agentes definidos pela classe PC representam as células do tecido parenquimatoso de um órgão abstrato, cuja função é manter sua estrutura de grade. Esse agente é baseado no agente PC, definido por Folcik *et al.* (2007).

Esses agentes são criados inicialmente na criação da zona Tissue. Eles são posicionados de maneira a formar uma estrutura bem definida, representando a funcionalidade dessas células em um órgão abstrato. Essa estrutura está ilustrada na Figura 3.4. São também o alvo dos patógenos no modelo, onde eles, após infectá-las, conseguem se reproduzir.

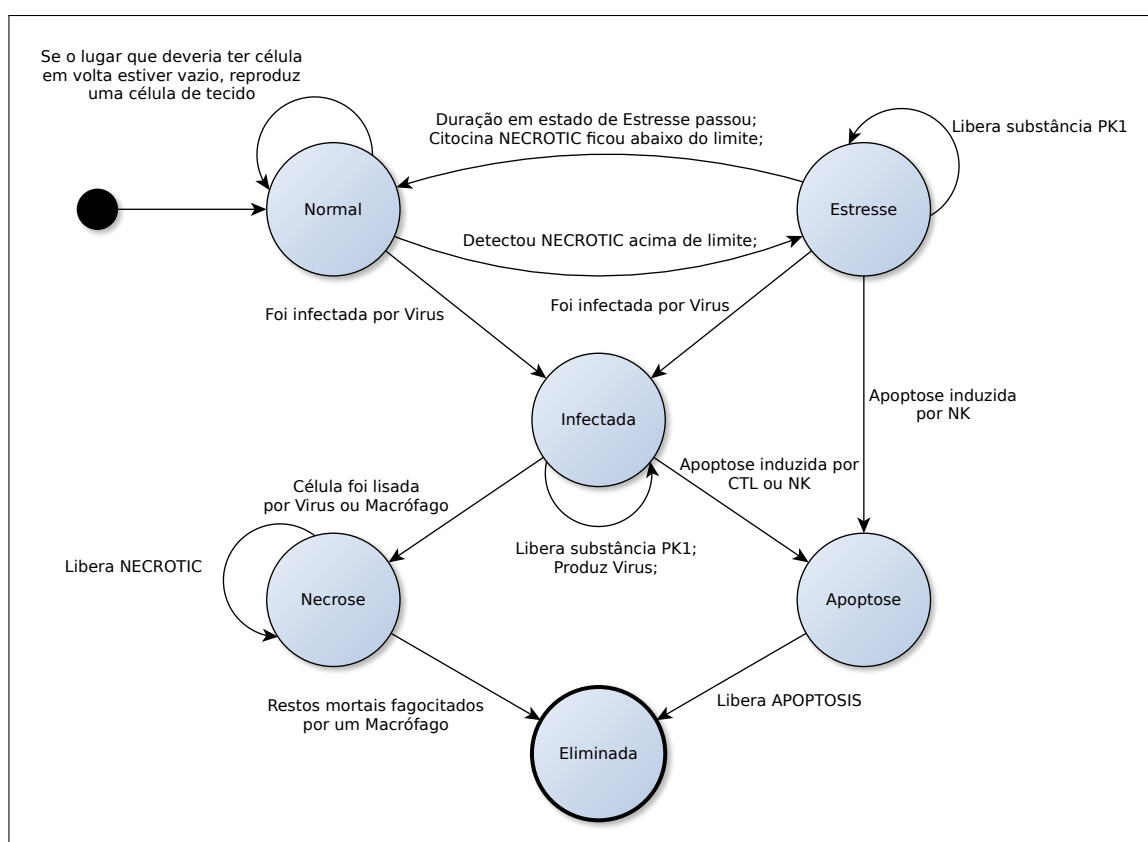


Figura 3.13. Regras do agente PC

As regras comportamentais desses agentes estão descritas na Figura 3.13. Como descrito no diagrama, inicialmente, permanecem em seu estado normal, quando verificam se suas células vizinhas (ou que deveriam ser suas vizinhas) estão funcionais. Caso não estejam, o agente cria clones seus, que vão para as posições onde deveriam haver células, para manter a estrutura. Essa função representa a regeneração do tecido. Caso alguma célula seja lisada, o seu conteúdo exposto pela lise, representado pela substância NECROSIS, pode “danificar” as células vizinhas, fazendo com que passem para um estado de estresse, liberando a substância PK1. Quando a substância NECROSIS se dissipa, ou quando a célula consegue se recuperar dos danos, o agente volta para seu estado normal.

Já quando um *Virus* logra sucesso em alcançar a célula e infectá-la, ela passa para o estado infectada. Neste estado, a célula libera a substância *PK1*, sinalizando ainda seu estado de estresse. Após ser infectada, podem acontecer duas situações: a célula pode ser eliminada por um linfócito T citolítico (*CTL*), por uma célula *NK*, ou fagocitada por um *Macrophage* ou *Dendritic*. Caso isso não ocorra, e o *Virus* vença o seu período de latência, definido pelo parâmetro *virusLatency*, a célula sofre lise, passa para o estado de necrose, e novos *Virus* são liberados. A quantidade de novos vírus é definida pelo parâmetro *virusVirulency*.

Tanto no estado de apoptose, quanto no estado de necrose, o agente *PC* continua no modelo, representando os restos mortais da célula. Em certo momento, esses restos mortais são limpos pelo *Macrophage*, e aí então, o agente *PC* é eliminado da simulação, para dar espaço a outro, gerado pelos agentes *PC* da vizinhança.

3.4.4.8 *NK*

O agente *NK* representa a célula *NK* (*Natural Killer*), e é baseado no agente “*NK*”, definido no modelo descrito por Folcik *et al.* (2007).

Como descrito na Seção 2.1, a principal função desse agente é eliminar os depósitos de infecção, *i.e.*, células infectadas que permitem a reprodução do vírus, enquanto a imunidade adaptativa se prepara. Como descrito por Folcik *et al.* (2007), embora a presença do complexo de histocompatibilidade principal classe I seja capaz de inibir sua ação, em meios pró-inflamatórios essa inibição é ignorada, e o *NK* mata a célula estressada.

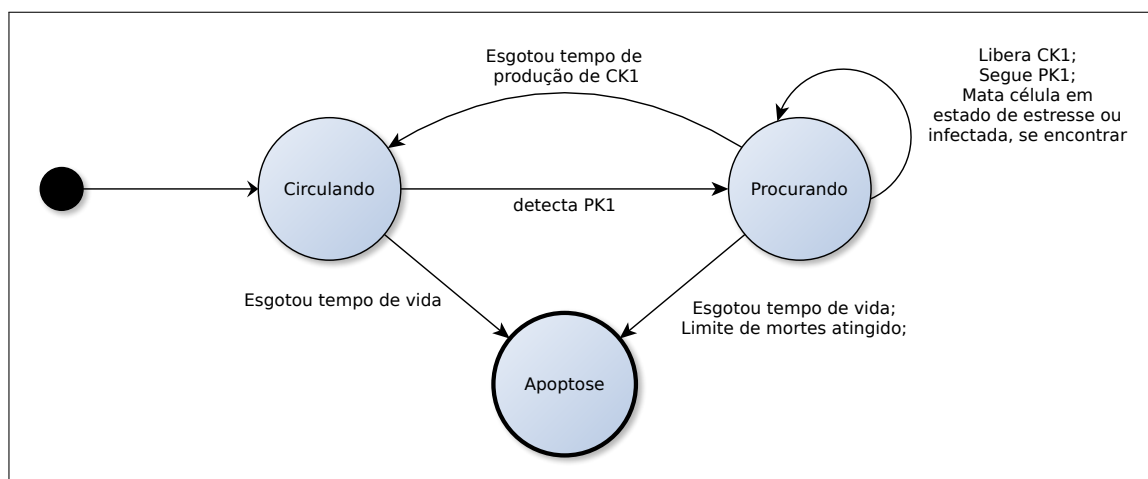


Figura 3.14. Regras do agente *NK*

As regras do agente *NK* estão descritas na Figura 3.14. Esse agente é introduzido na simulação na zona *Tissue*, quando um agente *TissuePortal* percebe a presença da citocina pró-inflamatória *MK1*, simulando assim a quimiotaxia dessas células. Ao che-

gar ao local, inicialmente ficam circulando pelo tecido. Ao detectarem a presença da substância PK1, o agente passa a procurar por células que estejam emitindo PK1, *i.e.*, célula estressadas e/ou infectadas. Ao encontrar uma célula que esteja nessas condições, e houver grande concentração de MK1 no local, então o NK mata a célula.

Quando seu tempo de vida termina, ou quando ele extermina um número limite de células, o agente NK sofre apoptose.

3.4.4.9 *ThCell*

O agente *ThCell* representa os linfócitos T *helper*, e é baseado nos agentes “T”, definidos no modelo descrito por Folcik *et al.* (2007), mas recebeu grandes modificações para que fosse possível simular sua especificidade. A principal função desse agente é prover os estímulos necessários para ativação dos agentes *BCell* e, mais tarde, os agentes *Macrophage*.

Como descrito na Seção 3.3.4, esse agente é gerado na zona *Thymus*, onde recebe sua especificidade (geração de diversidade) e é testado para garantir que não seja auto-reativo (tolerância central).

As regras do agente *ThCell* estão descritas na Figura 3.15. Ao chegar no *Lymphnode*, após ter sido gerado no *Thymus*, fica inativo, aguardando o contato com algum agente *APC*. Quando esse contato acontece, e o agente *APC* apresenta seu antígeno através do *MHCII*, o *ThCell* tenta reconhecê-lo. Em caso positivo, ele passa para o estado ativado.

Quando ativado, esse agente libera a citocina CK1, se prolifera, criando clones de si mesmo, que irão migrar para a zona *Tissue*, e se movimenta pelo *Lymphnode* à procura de linfócitos B, para ativá-los. Nesse estágio, cada novo reconhecimento de antígeno prolonga sua vida. Caso o nível de CK1 fique abaixo do limite, ele se torna uma célula de memória, capaz de responder rapidamente caso um agente *APC* a reative. Se, apesar da alta concentração de CK1, ele não reconhecer novamente algum antígeno, o seu tempo de vida se esgota, e ele sofre apoptose.

Já na zona *Tissue*, esse agente segue as citocinas MK1 e PK1, à procura do sítio de infecção. Ao reconhecer o antígeno apresentado por uma *APC* (no caso, um *Macrophage*), seu tempo de vida é estendido e inicia-se novamente a liberação de CK1. Caso isso não ocorra, ele sofre apoptose.

Os mecanismos de tolerância periférica, assim como os subtipos dos linfócitos T, em especial os ligados à supressão da resposta imunológica, não foram modelados, e serão incluídos na Seção 5.1 (Trabalhos Futuros).

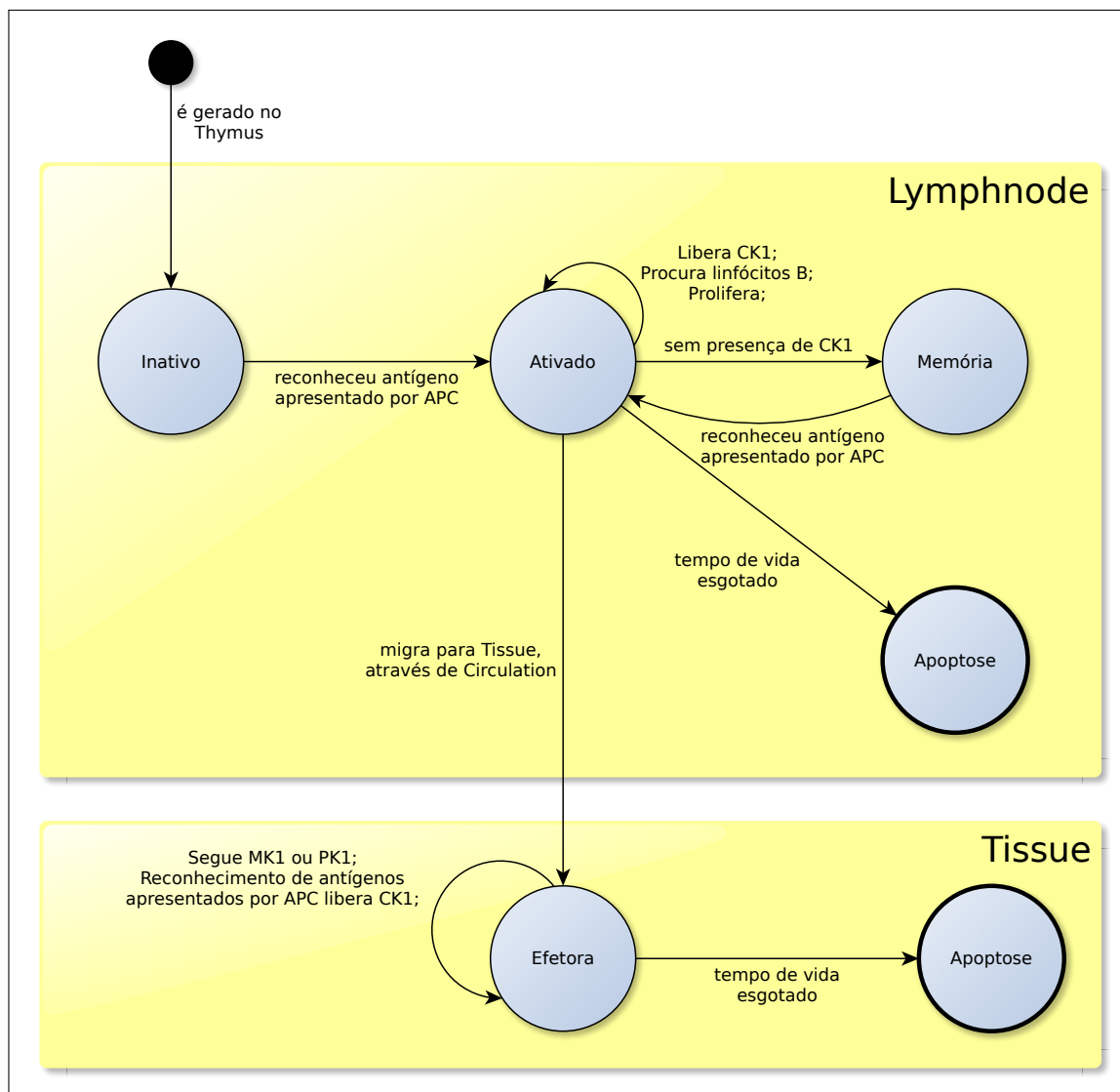


Figura 3.15. Regras do agente ThCell1

3.4.4.10 CTL

O agente CTL representa os linfócitos T citolíticos, e é baseado nos agentes “CTL” definidos no modelo descrito por Folcik *et al.* (2007), com adição de várias modificações para que possa ser possível representar sua especificidade. A principal função desse agente é, assim como o agente NK, eliminar os depósitos de infecção. Contudo, como descrito na Seção 2.1, os agentes NK eliminam as células cujo mecanismo de apresentação de antígenos pelo MHC I foi bloqueado pelo patógeno, enquanto o agente CTL precisa reconhecer o antígeno apresentado pelo MHC I para que elimine a célula infectada. É válido ressaltar que, como descrito por Folcik *et al.* (2007), a célula NK é capaz de ignorar a apresentação do MHC-I em situações pró-inflamatórias, eliminando a célula em estado de estresse mesmo assim.

Como descrito na Seção 3.3.4, esse agente é gerado na zona Thymus, onde recebe sua

especificidade (geração de diversidade) e é testado para garantir que não seja auto-reativo (tolerância central).

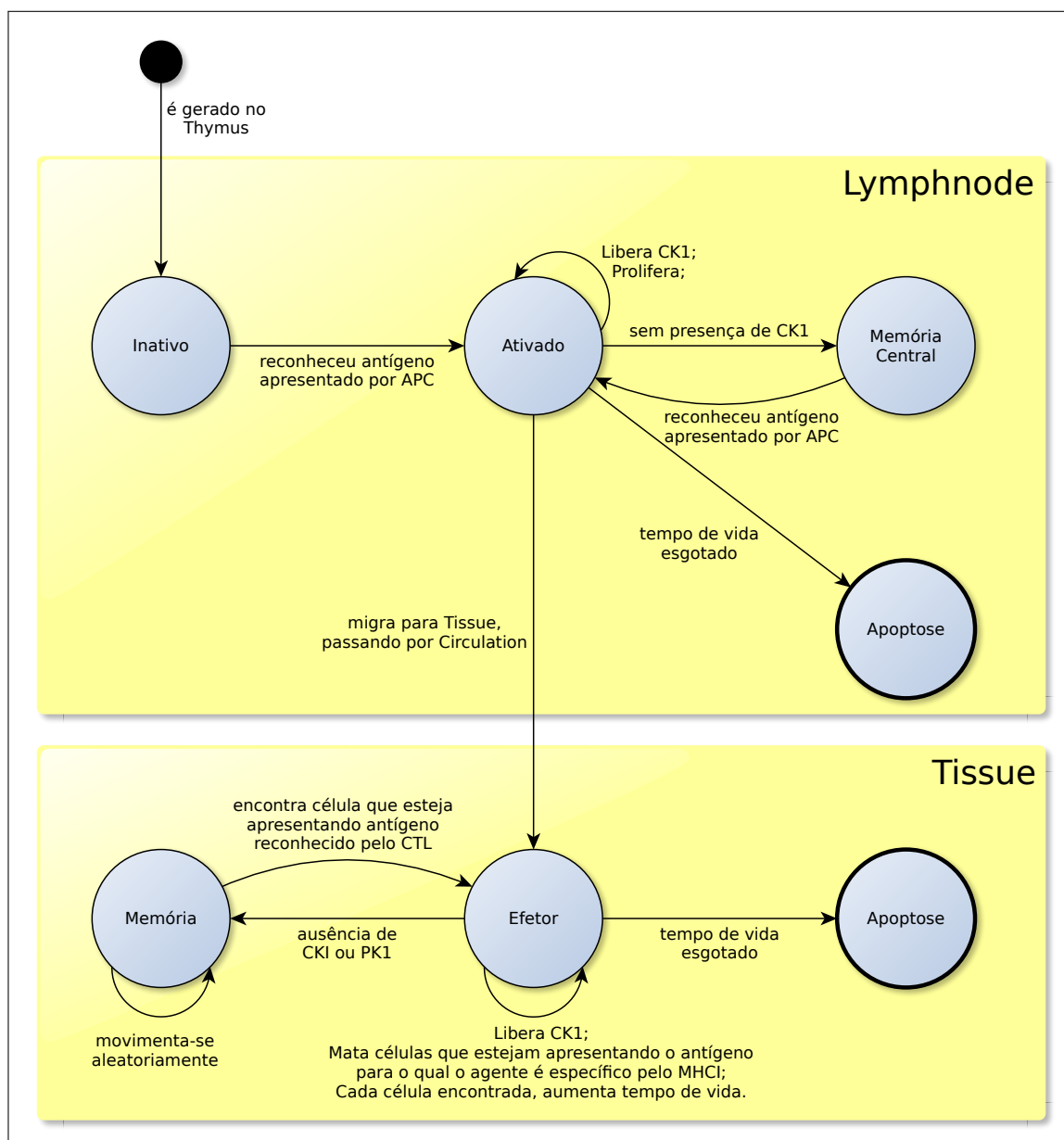


Figura 3.16. Regras do agente CTL

As regras do agente CTL estão descritas na Figura 3.16. Ao chegar ao Lymphnode, após ter sido gerado no Thymus, o agente CTL fica inativo, aguardando o contato com uma célula APC. Quando a encontra, ele tenta reconhecer o antígeno apresentado por ela, simulando assim o mecanismo denominado por apresentação cruzada (ou *cross-priming* em Inglês) (ABBAS; LICHTMAN, 2007; FOLCIK *et al.*, 2007).

Caso o CTL reconheça o antígeno, ele se torna ativado, e produz clones de si mesmo, que irão migrar para a zona Tissue. Também inicia a produção da citocina CK1. A partir daí, passa a andar aleatoriamente pelo Lymphnode a fim de aumentar suas chances

de contato com outra célula APC. Caso ocorra, gera novos clones e inicia produção de CK1 novamente. Caso ele fique um dado tempo, definido como um parâmetro do modelo, sem encontrar uma APC e sem sentir mais a presença de CK1, o agente torna-se uma célula de memória, que poderá ser ativada rapidamente por um novo reconhecimento de um antígeno apresentado por uma APC. Já se ele, mesmo na presença de CK1, não encontrar nenhuma APC da qual reconheça um antígeno, ele sofre apoptose.

Já no *Tissue*, o CTL torna-se um linfócito efetor, seguindo o PK1, examinando o MHC I das células e, caso reconheça algum antígeno, matando-as. Na ausência de citocinas inflamatórias, como PK1 e CK1, torna-se uma célula de memória, e passa a andar livremente pelo tecido, mas ainda examinando os antígenos apresentados pelo MHC I. Caso reconheça algum, retorna imediatamente para o estado efetor.

Se, por outro lado, na presença de citocinas inflamatórias, esse agente não reconhecer o antígeno apresentado por nenhuma célula em um dado período de tempo, sofrerá apoptose, pois isso demonstra que, embora ainda haja infecção, o agente está sendo inútil.

3.4.4.11 *APC*

A classe APC é uma classe abstrata, que representa o grupo de células conhecidas como células apresentadoras de antígeno (APC), como descrito na Seção 2.1. A principal característica dessas células é o poder de capturar, processar e apresentar antígenos através do complexo de histocompatibilidade principal classe II.

Todo agente que estenda essa classe precisa implementar o método abstrato *MHCII*, responsável por apresentar os padrões antigênicos aos linfócitos.

Todas os agentes descritos a seguir estendem essa classe, e portanto, têm capacidade de apresentar antígenos.

3.4.4.12 *Macrophage Cells*

Os agentes definidos pela classe *Macrophage* representam a principal célula da imunidade inata celular, os macrófagos. Esse agente é baseado no agente *Macrophage*, definido por Folcik *et al.* (2007).

Esses agentes são criados pela classe *TissuePortal*, quando estes detectam a presença de MK1, e inseridos na zona *Tissue*, representando assim a atração dos macrófagos para o sítio de infecção, como descrito na Seção 2.1, e destacado por Folcik *et al.* (2007).

As regras desse agente estão ilustradas na Figura 3.17. Ao entrar na zona *Tissue*, o agente segue a substância sinalizadora de estresse celular PK1, até encontrar o sítio de

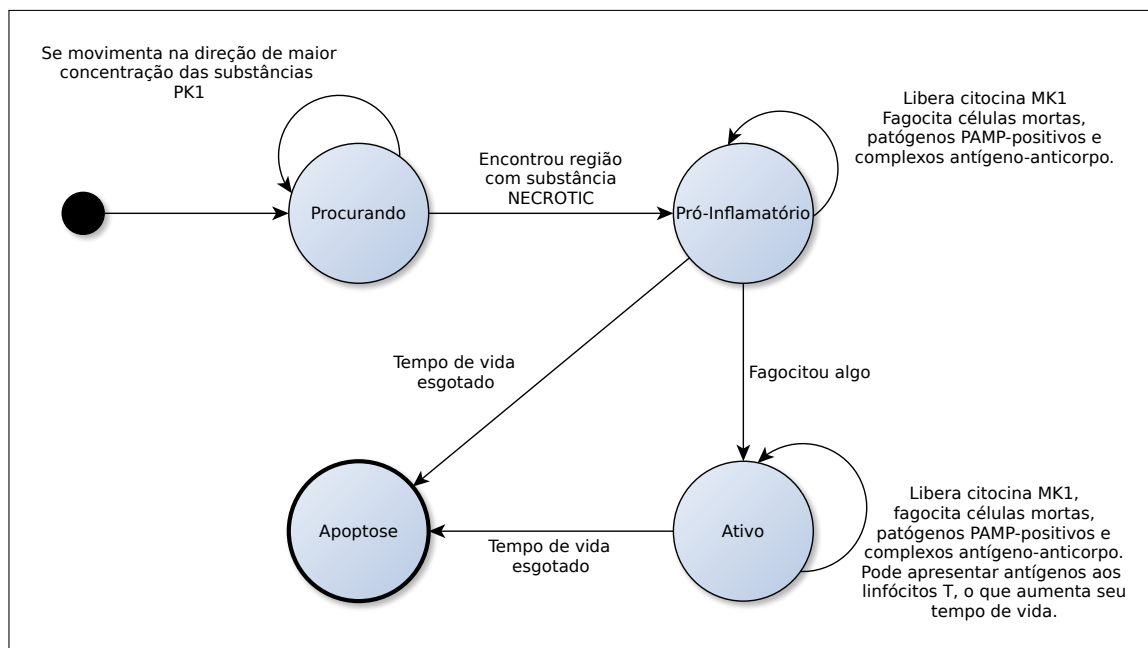


Figura 3.17. Regras do agente **Macrophage**

infecção. Ao encontrá-lo, e detectar a presença da substância NECROSIS, o agente passa para o estado pró-inflamatório, uma vez que encontrou evidência que as células estão realmente sendo danificadas. Ao entrar nesse estado, ele começa a produzir a citocina pro-inflamatória MK1, simulando a função do macrófago em iniciar o processo inflamatório, recrutando os demais elementos da imunidade. Nesse estado, ele também pode fagocitar células mortas, patógenos PAMP-positivos, e complexos antígeno-anticorpo, *i.e.*, anticorpos ligados a algum antígeno, marcando-o para fagocitose.

Ao fagocitar algum elemento, o Macrophage processa-o, extraíndo seus antígenos e passa apresentá-los pelo complexo de histocompatibilidade principal classe II (MHCII). Nesse momento, ele passa para o estado ativado. Quando ele encontra linfócitos T que reconheçam o antígeno que ele está apresentando, ele recebe fatores de sobrevivência, aumentando seu tempo de vida ativado, e fornece fatores de sobrevivência, para que o linfócito que reconheceu o antígeno também tenha seu tempo de vida aumentado.

Tanto no estado pró-inflamatório, quanto no estado ativado, o Macrophage têm um tempo de vida limitado, e se não receber fatores de sobrevivência que estendam este tempo, ele sofre um processo de morte celular programada, a apoptose, e é eliminado da simulação.

Os estados anti-inflamatórios do Macrophage não foram modelados, pois este modelo baseia-se apenas na hipótese de que o declínio da resposta imunológica acontece devido à ausência de antígenos que a estimule. Em contrapartida, existem outros estudos, citados inclusive no modelo descrito por Folcik *et al.* (2007), que apontam para a existência de células T reguladoras ou supressoras, que seriam responsáveis pelo declínio e supressão

da resposta imunológica. Essas células serão incluídas na Seção 5.1 (Trabalhos Futuros).

3.4.4.13 *Dendritic Cells*

Os agentes definidos pela classe *Dendritic* representam as células dendríticas, principal célula apresentadora de antígenos do modelo. Esse agente é baseado no agente “DC”, definido no modelo descrito por Folcik *et al.* (2007).

A principal função desse agente é patrulhar o tecido em busca de sinais de perigo (*danger signals*), e caso os encontre, capturar os antígenos presentes no local e apresentá-los aos linfócitos da imunidade adaptativa. É válido ressaltar que esse agente só fagocita elementos que estejam em *danger zones*, simulando assim uma tolerância periférica abstrata.

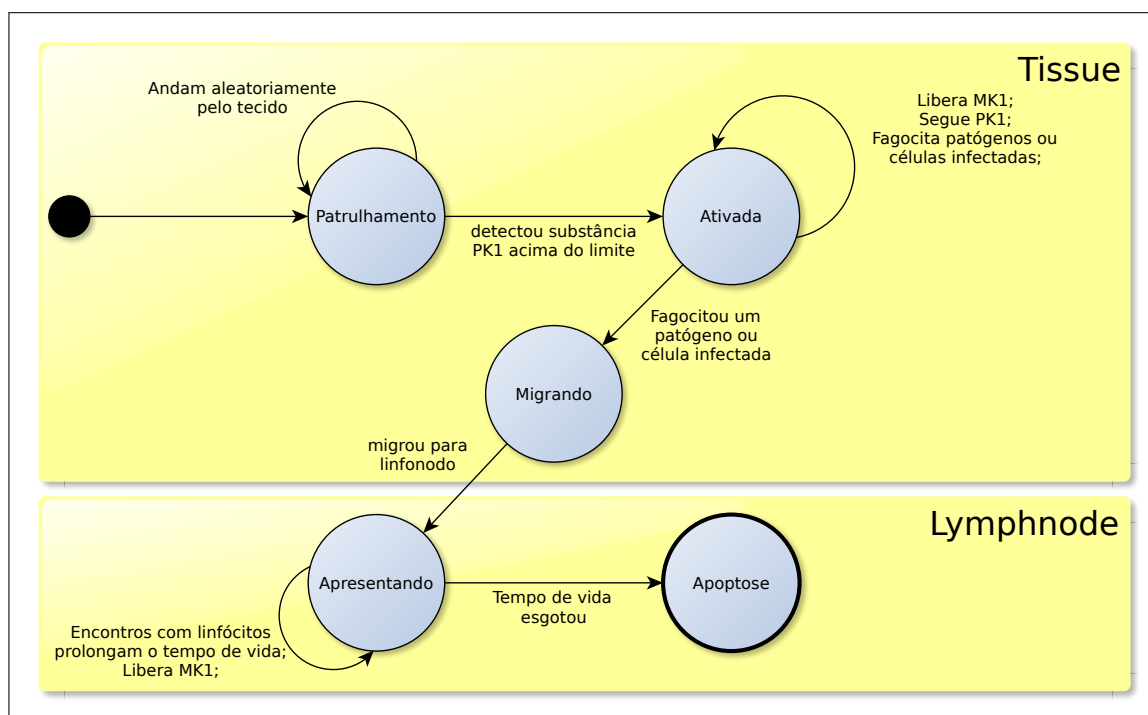


Figura 3.18. Regras do agente *Dendritic*

As regras do agente *Dendritic* estão descritas na Figura 3.18. Esse agente é criado inicialmente durante a criação da zona *Tissue*, e permanece lá andando aleatoriamente pelo tecido, patrulhando-o. Ao detectar a presença da substância sinalizadora de estresse celular, PK1, o agente *Dendritic* se torna ativado, e passa a concentrar-se no local da infecção capturando antígenos presentes no local (restos mortais de células, células infectadas e patógenos). Após ter fagocitado algum antígeno, começa a migração para o linfonodo.

Ao chegar ao linfonodo, sua função é estar disponível para que os linfócitos façam contato. Tenta se manter afastado de outros agentes *Dendritic* para aumentar a área

de cobertura. Como o `Dendritic` estende a classe `APC`, ele possui o método `MHCII`, utilizado para apresentar seus antígenos aos linfócitos. A cada apresentação bem sucedida, o tempo de vida do agente no linfonodo é aumentado. Ao fim desse tempo, ele sofre apoptose.

Como descrito por Folcik *et al.* (2007), esse agente tem uma grande capacidade de vasculhar o tecido, por esse motivo, ele é o único agente que utiliza a Vizinhança de Moore com raio igual a 2, *i.e.*, ao invés de olhar as suas 8 posições vizinhas, ele verifica 24 posições ao seu redor, em busca de antígenos.

3.4.4.14 *B Cells*

Os agentes definidos pela classe `BCell` representam os linfócitos B, a principal célula da imunidade adaptativa humoral. Essa classe é baseada no agente B definido no modelo proposto por Folcik *et al.* (2007), mas com grandes modificações, para podermos representar sua especificidade e geração de anticorpos.

Esses agentes são gerados na zona `BoneMarrow`, onde ganham sua especificidade aleatória (geração de diversidade), a qual é testada para se assegurar que não irá reconhecer antígenos próprios (tolerância central). Quando prontos, migram para a zona `Lymphnode`. Esses agentes também são responsáveis por gerar os agentes `Antibody`, quando necessário e passando para eles a sua especificidade.

As regras do agente `BCell` estão descritas na Figura 3.19. Ao chegar no `Lymphnode`, inicialmente esses agentes permanecem inativos. Ao reconhecerem um antígeno, que pode estar livre, ou sendo apresentado por um agente `APC` (como o `Dendritic`), o agente `BCell` muda para o estado ativado, quando começa a seguir a citocina `CK1`. Ao encontrar um agente `ThCell` ativado e de mesma especificidade, o agente `BCell` recebe o estímulo necessário para se diferenciar, entrando no estado germinativo, representando o linfócito B germinativo; ou plasma, representando os plasmócitos. Em ambos os casos, o agente `BCell` realiza sua proliferação, criando clones de si mesmo. O número de clones e a porcentagem com que ele se torna germinativo ou plasma são parâmetros do modelo.

Se o `BCell` se encontra no estado germinativo, então ele produz agentes `Antibody`, com a mesma especificidade que ele, até que seu tempo de vida tenha se esgotado, e ele passe para o estado de apoptose. No entanto, esse tempo pode ser aumentado caso o agente encontre com agentes `APC` apresentando o mesmo antígeno que o ativou. Se esses contatos ainda se repetirem uma certa quantidade de vezes, definida por um parâmetro do modelo, o agente se torna uma célula B de memória.

No estado de memória, esse agente pode ser ativado instantaneamente pela detecção da presença da citocina `CK1` no seu meio, ou pelo contato com um agente `ThCell` ativado,

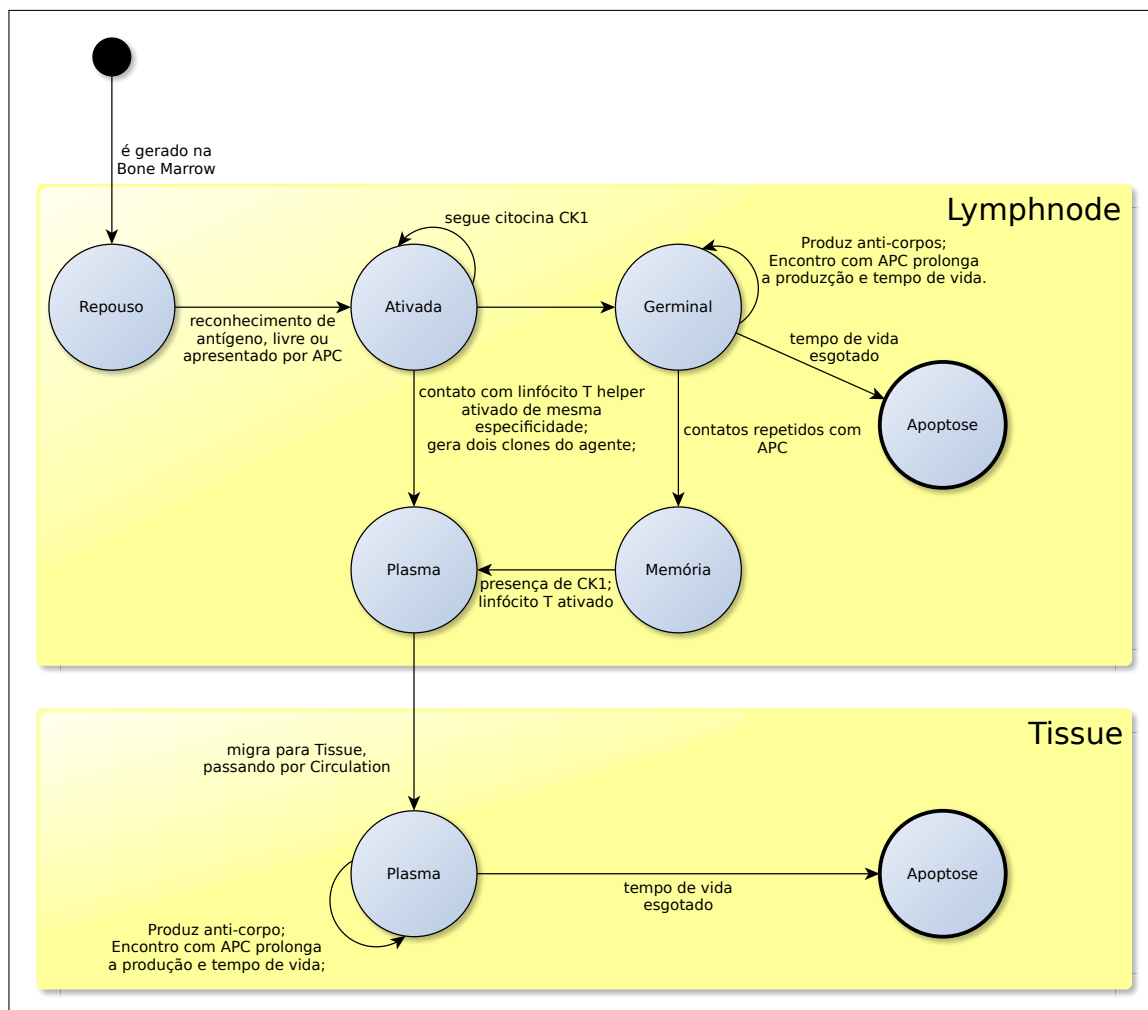


Figura 3.19. Regras do agente BCell

passando diretamente para o estado de plasmócito.

Já se o agente se encontra no estado plasma, representando os plasmócitos, ele migra até a zona Circulation. Lá, segue os sinais químicos emitidos pelos agentes CirculationPortal até encontrar um que o leve para o sítio de infecção, na zona Tissue. Ao chegar lá, começa sua produção de anticorpos, até que seu tempo de vida lá tenha se esgotado.

Mecanismos como maturação de afinidade não foram modelados, e serão incluídos na Seção 5.1 (Trabalhos Futuros).

CAPÍTULO 4

RESULTADOS

4.1 Implementação do modelo

Como já mencionado durante as seções anteriores deste trabalho, o modelo foi implementado utilizando o *framework* Repast Simphony, versão 1.2.0, em sua distribuição para Linux. Essa versão do Repast é disponibilizada como um *plug-in* para a plataforma de desenvolvimento Eclipse, e pode ser obtida pelo endereço <http://mirror.anl.gov/pub/repast-simphony>.

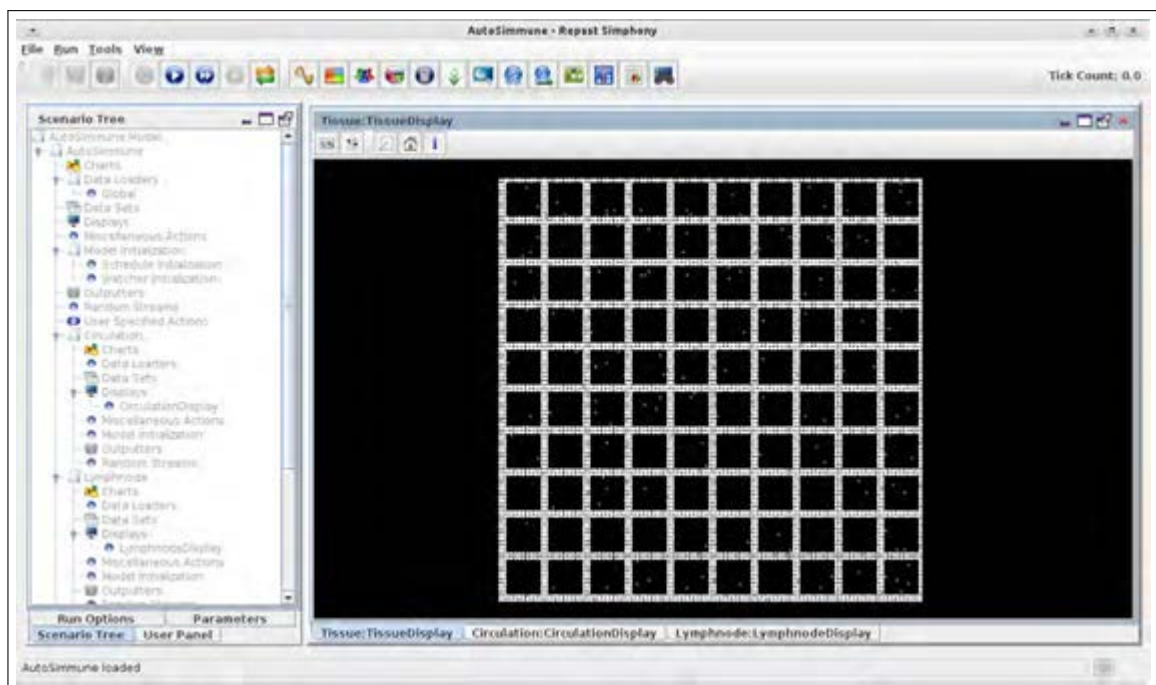


Figura 4.1. Ferramenta implementada: Autosimmune. Visão geral da zona **Tissue**.

O ambiente de desenvolvimento (IDE) utilizado foi o Eclipse Galileo (Eclipse 3.5.2, para Linux), em uma máquina executando o Ubuntu 11.04, e o Java Development Tool-

kit (JDK) para Linux, versão 1.6.0.26. Embora o ambiente de desenvolvimento seja na plataforma Linux, o *software* resultante, escrito na linguagem Java, pode ser executado em qualquer plataforma que suporte aplicativos Java Standard Edition (Java SE), como o Microsoft® Windows.

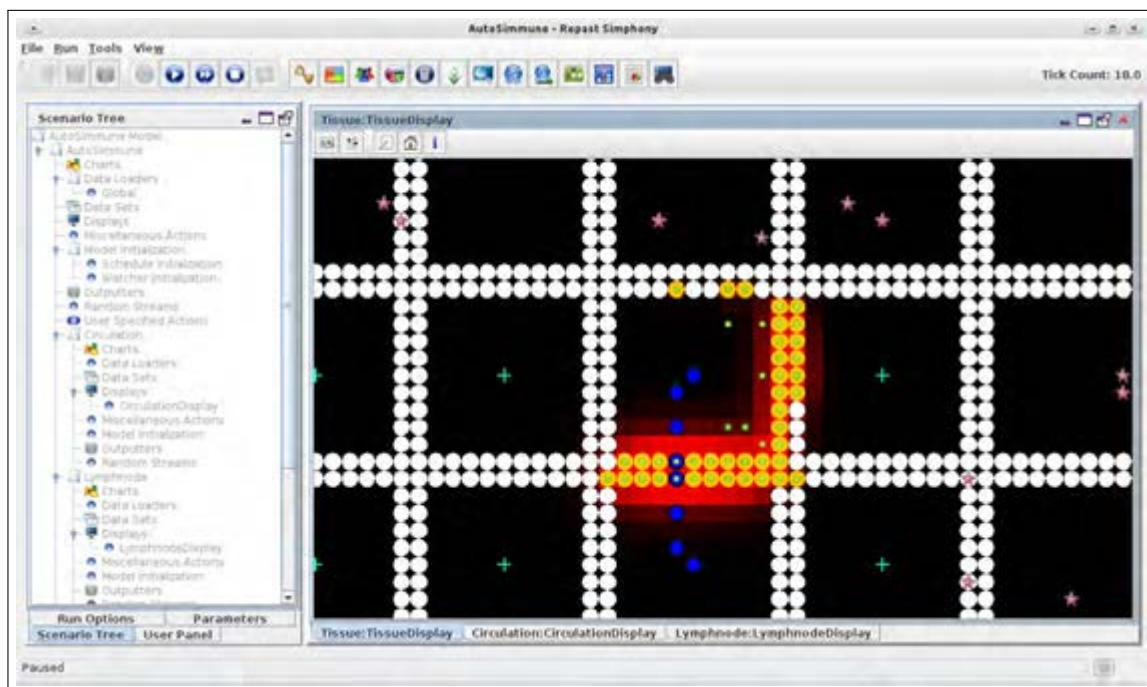


Figura 4.2. Ferramenta implementada: Autosimmune. Detalhe da zona **Tissue**.

A abordagem utilizada na implementação foi a *bottom-up*. Assim, primeiro foram desenvolvidas as classes que fariam o interfaciamento com as classes nativas do *framework*, como a classe `Environment` do modelo, que abstrai o uso das classes `Context`, `Projection` e `ValueLayer` do *framework*, assim como a classe `Agent` do modelo, que abstrai o uso dos métodos do *framework*, como `moveTo`, `add` e `remove`, por exemplo. A cada nova classe pronta, o sistema tinha suas novas funcionalidades testadas. Se estivesse tudo funcionando corretamente, novas classes eram adicionadas.

Os parâmetros utilizados pelo modelo, descritos na Seção 3.3.3.1, utilizam-se de uma facilidade do *framework*, que permite que seus valores sejam ajustados através da interface gráfica, aumentando assim a usabilidade da ferramenta.

A Figura 4.1 apresenta a aparência da ferramenta Autosimmune em funcionamento. À esquerda estão as configurações do modelo, que são ajustadas antes do início da simulação. À direita, e ocupando a maior parte da figura, estão as zonas: a zona selecionada e sendo exibida no momento é a zona **Tissue**. A estrutura em forma de grade representa a organização estrutural de um órgão abstrato. Já a Figura 4.2 apresenta a zona **Tissue** em detalhe: as células PC (em branco) sendo infectadas pelo **Virus** (em verde e menor). O gradiente vermelho representa a concentração da substância PK1, que sinaliza o estado

de estresse das células parenquimatosas. Em azul escuro, em formato circular, estão os `Macrophage`, iniciando a resposta inflamatória. Já em azul claro, em formato de cruz, estão os agentes `TissuePortal`: note que os `Macrophage` estão emergindo de dois destes portais. Em tom rosado, no formato de estrela, estão os agentes `Dendritic`.

A seguir serão descritos os resultados de simulações de cinco casos de uso, e análise dos seus resultados quanto à coerência do comportamento apresentado pelo modelo com o que se espera, segundo a literatura pesquisada. Os testes foram realizados no ambiente Linux Ubuntu 11.04, com Java SE versão 1.6.0.26, em um computador com processador Core 2 Duo de 2.0 GHz e com 3 GB de memória RAM.

4.2 Simulação do modelo

Essa seção do trabalho têm como objetivo evidenciar a viabilidade do modelo como ferramenta para a simulação de processos relacionados à autoimunidade no sistema imunológico. A seguir, serão apresentados alguns testes realizados, utilizando a implementação do modelo.

Os testes 1 e 2 foram realizados utilizando uma versão anterior da implementação, onde apenas as células responsáveis pela imunidade inata estavam presentes. Os resultados desses testes foram publicados no artigo Possi *et al.* (2011), apresentado na *XXXVII Conferência Latinoamericana de Informática (XXXVII CLEI)*. Os demais testes, por sua vez, foram realizados utilizando a implementação completa do modelo.

4.2.1 Teste 1: Resposta primária PAMP-dependente

Como visto na Seção 2.1, as células da imunidade inata utilizam-se das PAMPs para reconhecerem outros elementos como sendo nocivos e assim, reagirem contra eles. Neste teste, descrito por Possi *et al.* (2011), foram utilizados patógenos que se comportam como bactérias, se alimentando do tecido (células `PC`), destruindo-as e ganhando energia para se reproduzir.

Na primeira simulação, definiu-se o padrão molecular do patógeno como sendo um que não pertencesse ao conjunto PAMP. Após isso, a simulação foi iniciada, com um número inicial de 5 patógenos. A Figura 4.3 ilustra a evolução da infecção, mostrando o número de agentes presentes na simulação, em cada instante de tempo.

Segundo Possi *et al.* (2011), a análise da Figura 4.3 mostra que, embora as células da imunidade inata, representadas na simulação pelo `Macrophage` e pelo `NK`, tenham sido ativadas, pelo dano tecidual e conseqüente liberação das substâncias `PK1` e `NECROTIC`,

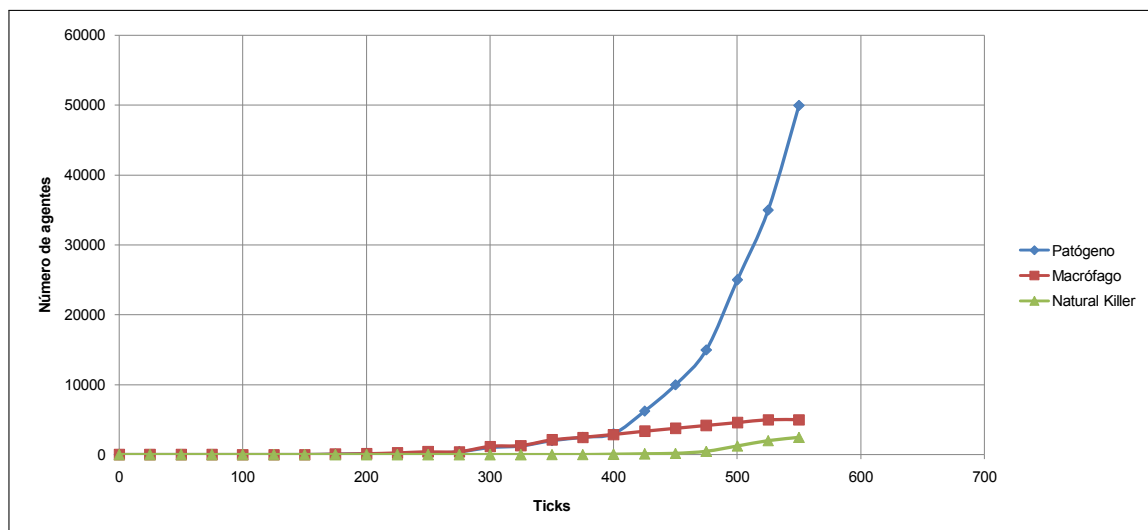


Figura 4.3. Evolução da Infecção de um patógeno PAMP-negativo.
 Fonte: (POSSI *et al.*, 2011, p. 10)

essas células não foram eficazes contra o patógeno, uma vez que ele não foi reconhecido pelo Macrophage como sendo algo hostil por não apresentar um padrão molecular reconhecido como um PAMP, e por este motivo, não pode fagocitá-lo.

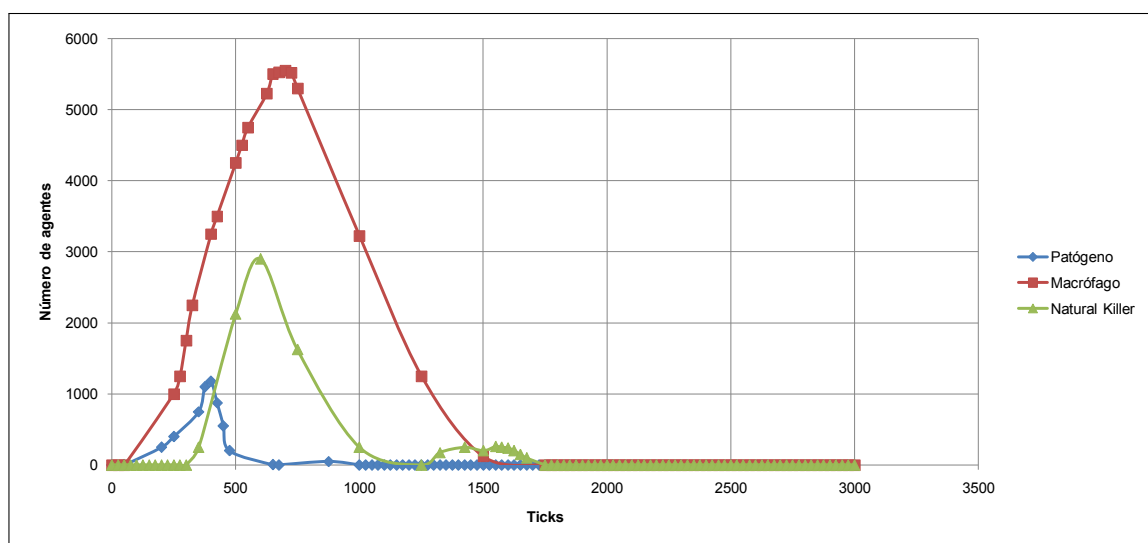


Figura 4.4. Evolução da Infecção de um patógeno PAMP-positivo.
 Fonte: (POSSI *et al.*, 2011, p. 10)

É válido ressaltar que, na implementação completa do modelo, as células *Dendritic* iriam fagocitar o patógeno (mesmo ele não apresentando PAMPs) e apresentariam seus antígenos aos linfócitos que, reconhecendo-os e tendo em vista que o patógeno fora capturado em uma *danger zone*, seriam ativados e eliminariam o patógeno.

Já na segunda simulação, segundo Possi *et al.* (2011), foram utilizados os mesmos parâmetros da primeira, exceto por um: definiu-se que o patógeno iria ter um padrão

molecular reconhecido como sendo um PAMP, e portanto, seria reconhecido como hostil pelas células imunológicas. A Figura 4.4 ilustra a evolução da infecção.

Segundo Possi *et al.* (2011), a análise da Figura 4.4 mostra que, da mesma forma, as substâncias PK1 e NECROTIC mobilizaram os macrófagos (Macrophage) que, por sua vez, iniciaram o processo inflamatório, liberando a citocina MK1, o que culminou no recrutamento das células NK, como na primeira simulação. A diferença foi que, desta vez, as células Macrophage foram eficazes contra os patógenos, pois conseguiram reconhecê-los como nocivos, fagocitando-os. Consequentemente, a infecção foi erradicada.

Segundo Possi *et al.* (2011), neste teste foi possível evidenciar, observando a movimentação dos agentes durante a simulação e os dados apresentados nos gráficos desta seção, que os mecanismos do modelo que envolvem quimiotaxia, difusão de substâncias, fagocitose e reconhecimento de padrões demonstram um comportamento coerente com a literatura.

4.2.2 Teste 2: Papel da NK na eliminação de reservatórios de infecção

Alguns patógenos, como os vírus, têm a capacidade de invadirem algumas células, onde conseguem se reproduzir. Tais patógenos são denominados intracelulares. No entanto, uma vez no interior da célula, mesmo apresentando PAMPs, estes patógenos ficam invisíveis aos receptores imunológicos. Os macrófagos, por exemplo, já não irão conseguir fagocitá-los. Pior ainda, essas células infectadas se tornam reservatórios de infecção, onde esses patógenos encontram o ambiente favorável para se multiplicarem.

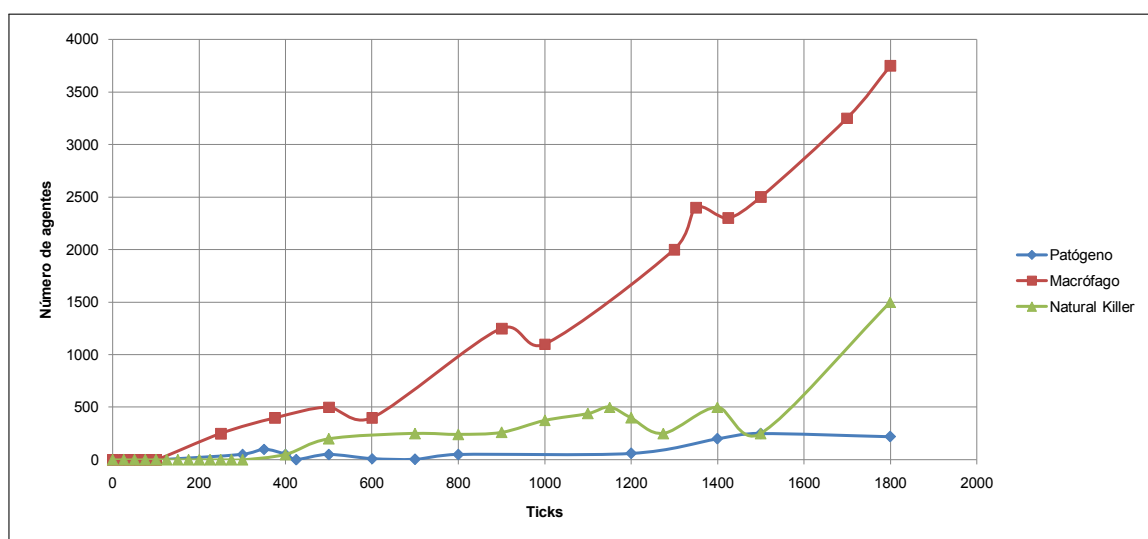


Figura 4.5. Evolução da Infecção com a atuação da NK.

Fonte: (POSSI *et al.*, 2011, p. 11)

Para solucionar esse problema, as células nucleadas do organismo possuem um mecanismo de apresentação de antígenos, através do MHC-I, pelo qual é apresentado um apanhado geral de tudo que é fabricado no interior da célula: caso estejam replicando vírus, provavelmente algum antígeno viral será apresentado. O MHC-I por sua vez é inspecionado pelos linfócitos CTL, cuja função é exatamente eliminar células cujo MHC-I esteja apresentando algum antígeno que este reconheça, eliminando assim os depósitos de infecção. Entretanto, os vírus são estruturas simples, capazes de se reproduzir muito rapidamente, e algumas vezes, não há tempo para esperar os linfócitos CTL proliferarem. Para resolver esse problema, existem as células NK, que, ainda na imunidade inata, conseguem eliminar os reservatórios de infecção.

Nesse teste, descrito por Possi *et al.* (2011), foi usado um patógeno com um comportamento similar ao de um vírus: ele infecta as células, passa por um período de incubação, se reproduz e mata a célula quando a deixa. Na primeira simulação, as células NK estavam presentes, para assim tentar evidenciar seu papel. A Figura 4.5 ilustra a evolução da infecção.

Analisando a Figura 4.5 é possível notar que as células Macrophage iniciaram a resposta à infecção, recrutando as células NK que, por sua vez, controlaram a carga viral, interferindo na replicação do vírus e eliminando os depósitos de infecção, como evidenciado pela diminuição da taxa de aumento da carga viral, em relação ao aumento do número de células NK, entre a iteração 350 e 400 (POSSI *et al.*, 2011).

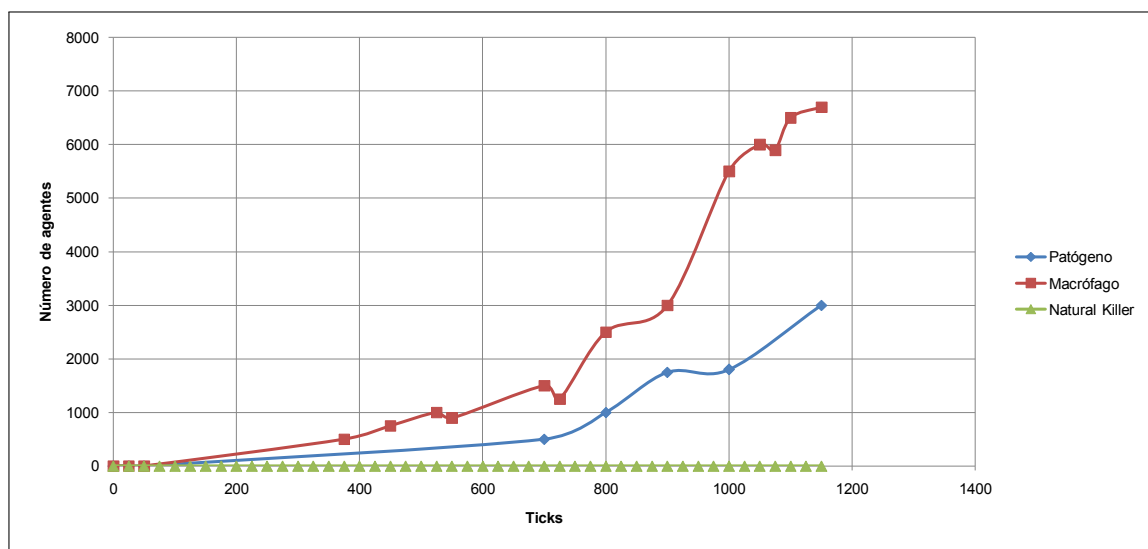


Figura 4.6. Evolução da Infecção sem a atuação da NK.

Fonte: (POSSI *et al.*, 2011, p. 12)

No entanto, a resposta imunológica inata não foi eficaz na eliminação completa do vírus. Contudo, Janeway *et al.* (apud POSSI *et al.*, 2011) apresenta resultados de pesquisas em ratos que mostram que as células NK têm o papel de controlar a replicação viral, mas

normalmente não conseguem eliminá-la. O estudo afirma ainda que tal eliminação deve ser alcançada pela ativação subsequente da resposta adaptativa, em especial, dos linfócitos CTL (que ainda não estava modelada nesse teste). Dessa forma, Possi *et al.* (2011) afirma que o resultado da simulação foi positivo, mostrando que o modelo demonstrou um comportamento esperado.

Já na simulação ilustrada na Figura 4.6, sem a presença das células NK, os vírus ficam ocultos nos reservatórios de infecção, fora do alcance dos mecanismos efetores das células Dendritic e Macrophage, de onde se multiplicam e alimentam a infecção. Segundo Possi *et al.* (2011), nota-se que, embora tenha havido um significativo aumento no número de células Macrophage em função da grande quantidade de lesões teciduais, estas células não conseguiram conter o aumento da carga viral, justamente por não conseguirem combater os depósitos de infecção, evidenciando a importância das células NK na imunidade inata.

4.2.3 Teste 3: Homeostasia

Neste teste, bastante simples, a ideia é evidenciar a homeostasia do modelo, *i.e.*, seu equilíbrio, de forma que, não havendo nenhum patógeno e, conseqüentemente, nenhum estímulo inflamatório, o número de células continua constante durante o tempo.

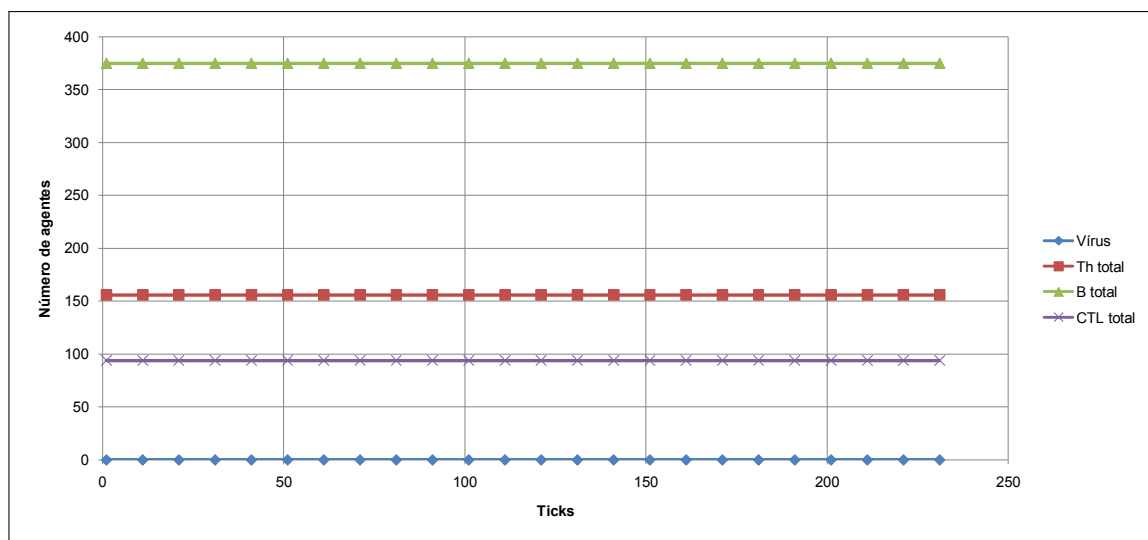


Figura 4.7. Homeostasia do sistema na ausência de patógenos.

A Figura 4.7 ilustra esse comportamento. Analisando-a é possível notar que o número de agentes (linfócitos BCell, ThCell e CTL) manteve-se constante na falta de um estímulo antígeno, pois não havia patógenos (Vírus).

Apesar disso, o modelo ainda não possui ferramentas para manter a homeostasia depois de várias infecções. Isso porque não há uma geração constante de linfócitos: todos são gerados no início da simulação, e os que são ativados, se proliferam. Seria interessante

que existisse um número médio de linfócitos, e que toda vez que esse número atingisse um limite mínimo, novos linfócitos fossem gerados pelos órgão linfoides geradores. Esse fato será incluído na Seção 5.1 (Trabalhos Futuros).

4.2.4 Teste 4: Especificidade e Memória

Como visto na Seção 2.1, as principais propriedades da imunidade inata adaptativa são sua especificidade, *i.e.*, somente as células que reconhecem o antígeno invasor se proliferam, evitando assim o gasto de recursos com células que serão inúteis no processo; e sua memória, *i.e.*, a cada encontro com o patógeno, a resposta imunológica se torna mais eficiente.

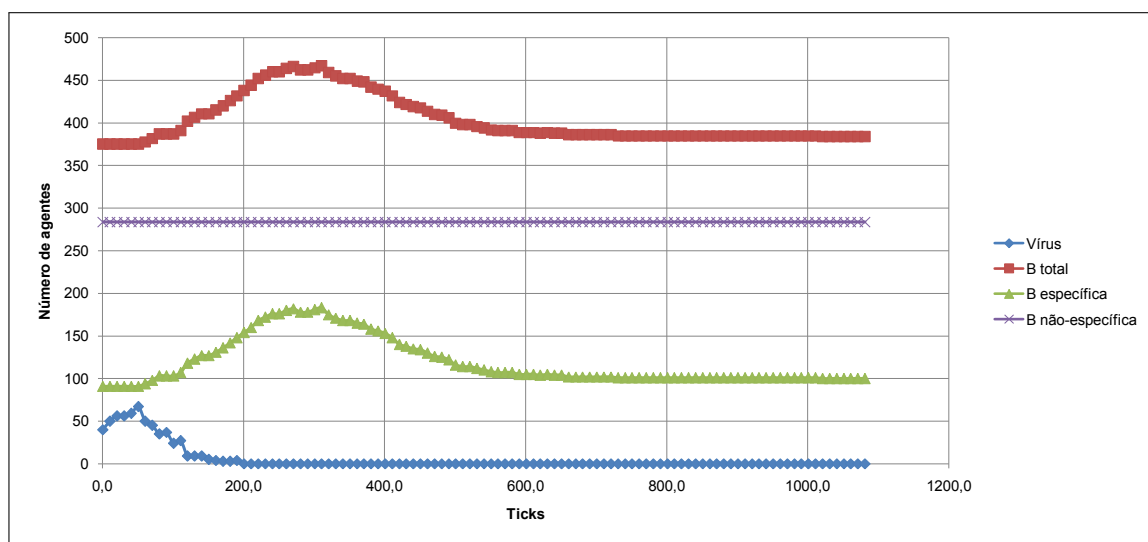


Figura 4.8. Especificidade dos agentes BCell.

Este teste tem a finalidade de evidenciar a especificidade e a memória como um comportamento do modelo. Ele foi realizado medindo a quantidade de linfócitos, dividindo-os em específicos e não específicos. É possível saber quais são específicos pois o padrão molecular do antígeno do patógeno é definido antes do início da simulação, e não se modifica durante a mesma. O número inicial de linfócitos específicos é aleatório, uma vez que o processo de geração de diversidade é um processo randômico.

Na Figura 4.8, são mostradas as quantidades do agente BCell total, específico para o antígeno do patógeno, não específico, e a quantidade de patógenos. Analisando-a é possível notar que apenas os linfócitos específicos proliferaram, em função do reconhecimento dos agentes infecciosos. Os linfócitos não específicos mantiveram sua quantidade inalterada. A contagem total de linfócitos BCell foi alterada apenas pelo aumento de linfócitos específicos, como se esperava. Após o declínio, o número de linfócitos específicos era maior do que no início: isso se deve a geração de células de memória, que sobreviveram

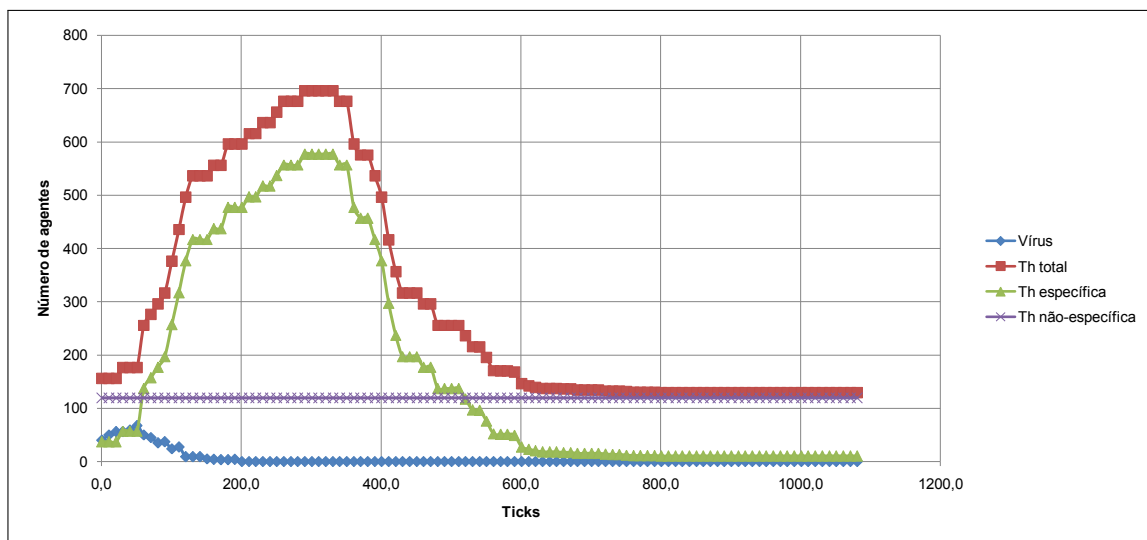


Figura 4.9. Especificidade dos agentes ThCell.

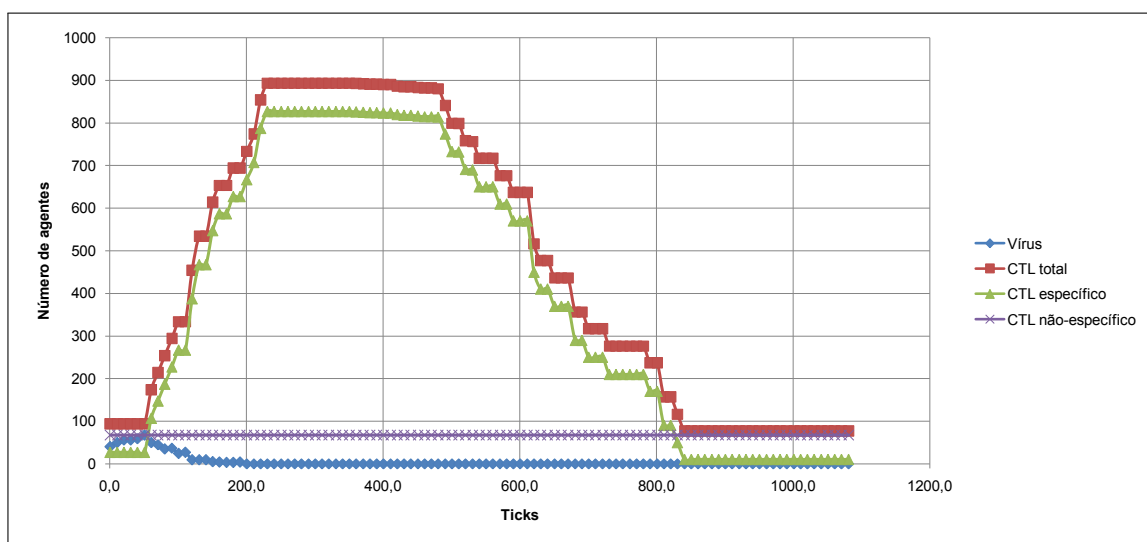


Figura 4.10. Especificidade dos agentes CTL.

ao declínio. Como será visto adiante, no modelo, o número de linfócitos específicos após o declínio nem sempre é maior do que no início. No entanto, os linfócitos BCell foram modelados para gerar um número maior de linfócitos de memória, representando o fato de que estas células podem diferenciar-se em dois tipos diferentes desses linfócitos de longa vida, como descrito na Seção 2.1. Os anticorpos gerados pelos linfócitos BCell ativados também serão específicos para o patógeno, pois “herdam” a especificidade da célula que os gerou.

Já na Figura 4.9, são mostradas as quantidades do agente ThCell. Da mesma forma, a análise desta figura demonstra que, mediante o estímulo antigênico, proveniente dos patógenos que foram inseridos no tecido, apenas os linfócitos específicos para o antígeno do patógeno foram ativados e proliferaram, como era esperado. Nota-se ainda que o número

de ThCell específicos após a fase de declínio é menor do que no início. Contudo, os linfócitos específicos que restaram são os linfócitos ThCell de memória, que são capazes de responder à infecções recorrentes do mesmo patógeno com muito mais eficiência.

Por fim, a Figura 4.10, por sua vez, apresenta as quantidades do agente CTL durante a resposta imunológica. Assim como os demais, apenas os linfócitos específicos se proliferaram, responderam ao patógeno e entraram na fase de declínio da resposta. Os que conseguiram condições favoráveis se diferenciaram em células de memória. A Figura 4.11 ilustra a simulação de uma infecção recorrente, na qual patógenos de mesmo padrão antigênico são inseridos na zona Tissue por duas vezes. Analisando-a é possível observar que a segunda resposta foi mais ampla que a primeira, e mais eficiente, o que pode ser evidenciado pelo fato do segundo pico de carga viral ser menor que o primeiro. Essa simulação também demonstrou um comportamento coerente com a literatura, comparando-se ao comportamento ilustrado pela Figura 2.8, da Seção 2.1.

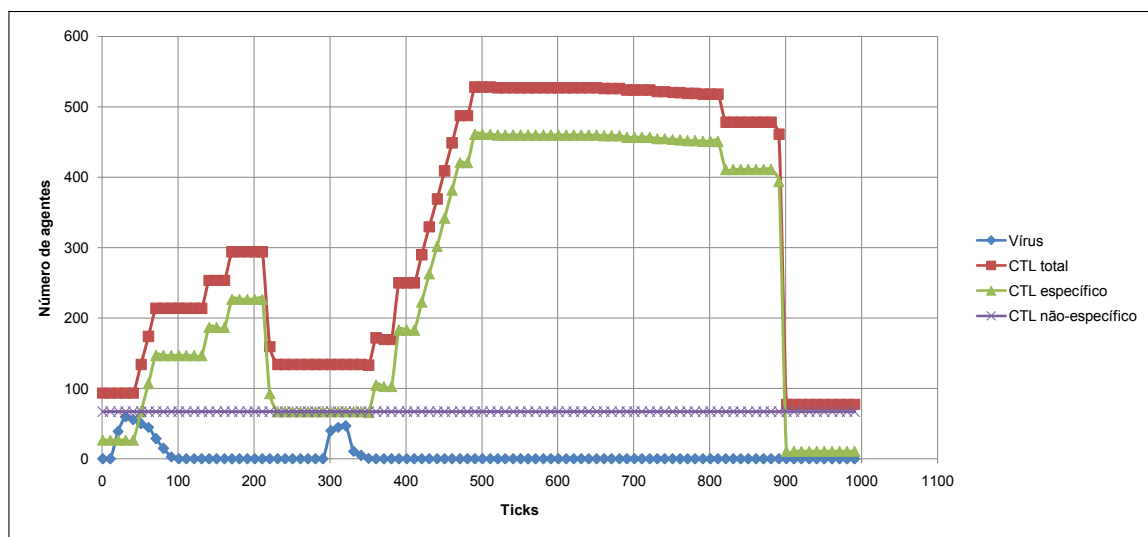


Figura 4.11. Especificidade e Memória dos agentes CTL.

Desta forma, é possível concluir que o modelo apresentou um comportamento satisfatório nas simulações de especificidade e memória, segundo os padrões comportamentais descritos na literatura da área.

4.2.5 Teste 5: Autoimunidade

O objetivo desse teste é evidenciar o comportamento do modelo, no que diz respeito à autoimunidade, simulando uma hipótese descrita na literatura sobre possíveis causas de doenças autoimunes, e observando se será exibido um comportamento autoimune, e mais ainda, se este irá culminar em uma doença auto-imune, caracteriza por haver alguma

lesão ao tecido sadio provocada por linfócitos auto-reativos. Por motivos de escassez de tempo, apenas os linfócitos CTL foram observados durante as simulações.

Como visto na Seção 2.1, as principais hipóteses para a causa das doenças autoimunes são (as hipóteses foram chamadas de Hipótese 1 e Hipótese 2, arbitrariamente, apenas por questões didáticas):

Hipótese 1 Linfócitos auto-reativos são gerados devido a defeitos genéticos, possivelmente nos genes do MHC, que afetaram o mecanismo de tolerância central. Esses linfócitos auto-reativos reconhecem autoantígenos apresentados por APCs, que podem ter capturado tais autoantígenos em uma *danger zone* (causada por um patógeno). O linfócito ativado ataca o tecido;

Hipótese 2 Linfócitos auto-reativos são gerados devido a defeitos genéticos, possivelmente nos genes do MHC, que afetaram o mecanismo de tolerância central. Esses linfócitos auto-reativos reconhecem antígenos, apresentados por APCs, de patógenos que têm um padrão molecular muito parecido ao padrão de um auto-antígeno do corpo, ativando o linfócito, que ataca o tecido do organismo.

Sendo assim, esse teste irá realizar quatro simulações, tentando evidenciar que o modelo possui os mecanismos necessários para representar fenômenos autoimunes, e permitir o estudo de suas patologias. As séries de dados retiradas do modelo (curvas do gráfico) são:

Vírus: quantidade de patógenos;

Lesão tecidual: número de células do tecido mortas por um linfócito CTL;

CTL total: número total de linfócitos CTL;

CTL específico: número de linfócitos CTL específicos para o agente infeccioso;

CTL não-específico: número de linfócitos CTL que não são específicos para o agente infeccioso;

CTL autoimune total: número de linfócitos CTL que são específicos para autoantígenos (especificamente, autoantígenos das células de tecido);

CTL autoimune tecidual: número de linfócitos CTL que são específicos para autoantígenos (especificamente, autoantígenos das células de tecido), que foram ativado e migraram para o zona Tissue.

Na primeira simulação, ilustrada na Figura 4.12, cujo objetivo é mostrar o funcionamento normal do modelo, foram utilizados os parâmetros normais, representando o comportamento de um SI saudável. Analisando a figura em questão, é possível notar que a infecção gerou uma resposta imunológica adaptativa normal, na qual apenas os linfócitos CTL específicos para o antígeno proliferaram. Note também que não existem linfócitos CTL auto-reativos, como se esperava.

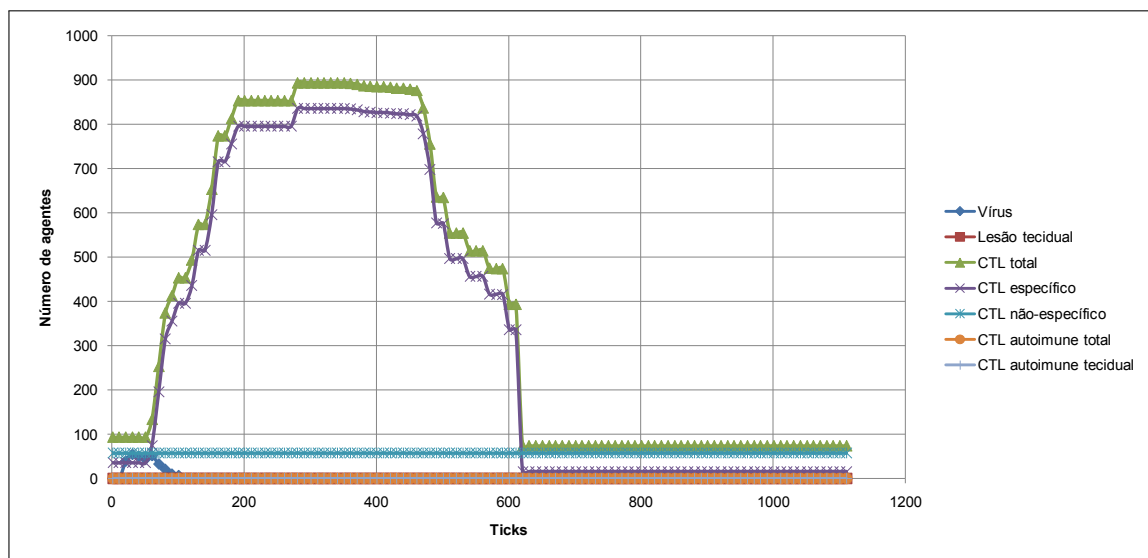


Figura 4.12. Autoimunidade dos linfócitos CTL, em situações normais.

Já na simulação, ilustrada na Figura 4.13, o objetivo é mostrar que apenas o mimetismo molecular não é capaz de gerar respostas autoimunes, e que para isso seria necessário uma suscetibilidade à geração de linfócitos auto-reativos. Nesta simulação o padrão molecular dos patógenos foi modificado para ser similar ao padrão das células PC. Note que as chances de existir um linfócito que reconhecesse esses patógenos eram bem pequenas, uma vez que, para isso, o linfócito teria que ser específico para um antígeno que é muito similar a um autoantígeno, mas não poderia ser específico para um autoantígeno em questão. Por isso, a resposta imunológica adaptativa, nesta simulação, nem chegou a ocorrer. O patógeno foi eliminado, com certo esforço, apenas pelos agentes da imunidade inata, que conseguiram tal feito devido ao baixo número inicial de patógenos utilizado, por motivos de escassez tempo (quanto mais agentes, mais lenta se torna a simulação). Contudo, o resultado foi como esperado.

Na simulação, ilustrada pela Figura 4.14, o objetivo é simular a Hipótese 1. Nesta simulação, apenas o mecanismo de tolerância dos linfócitos CTL foi desativado, simulando um problema genético que impediu apenas o seu funcionamento. Isso está de acordo com a literatura, pois, segundo Graser *et al.* (2000), os linfócitos CTL podem tornar-se ativos e desenvolverem uma resposta auto-reativa mesmo na ausência absoluta de linfócitos T *helper*. Todos os outros mecanismos de tolerância continuaram ativados.

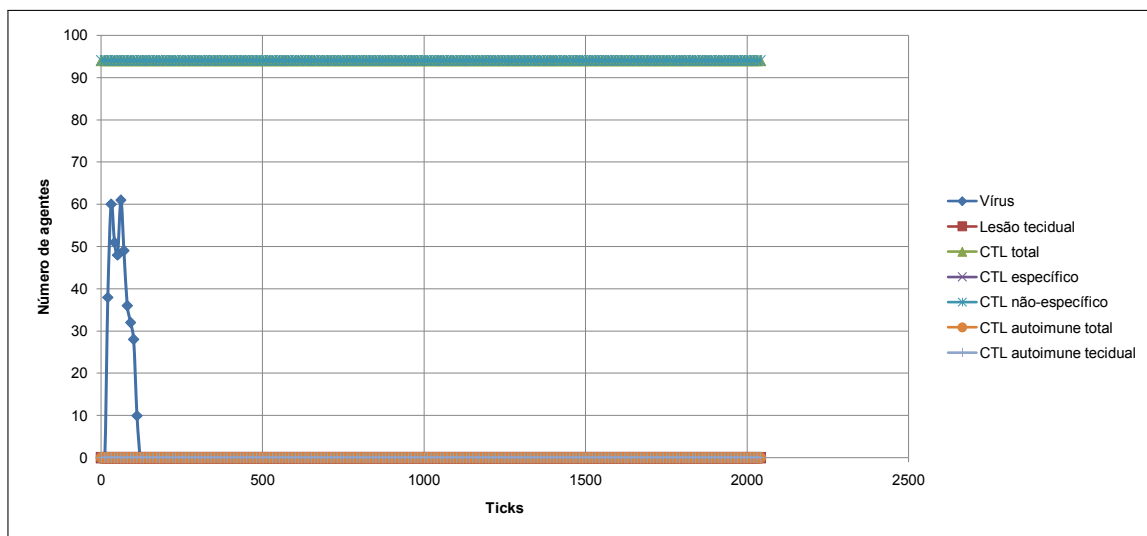


Figura 4.13. Autoimunidade dos linfócitos CTL, em condições normais, com mimetismo molecular.

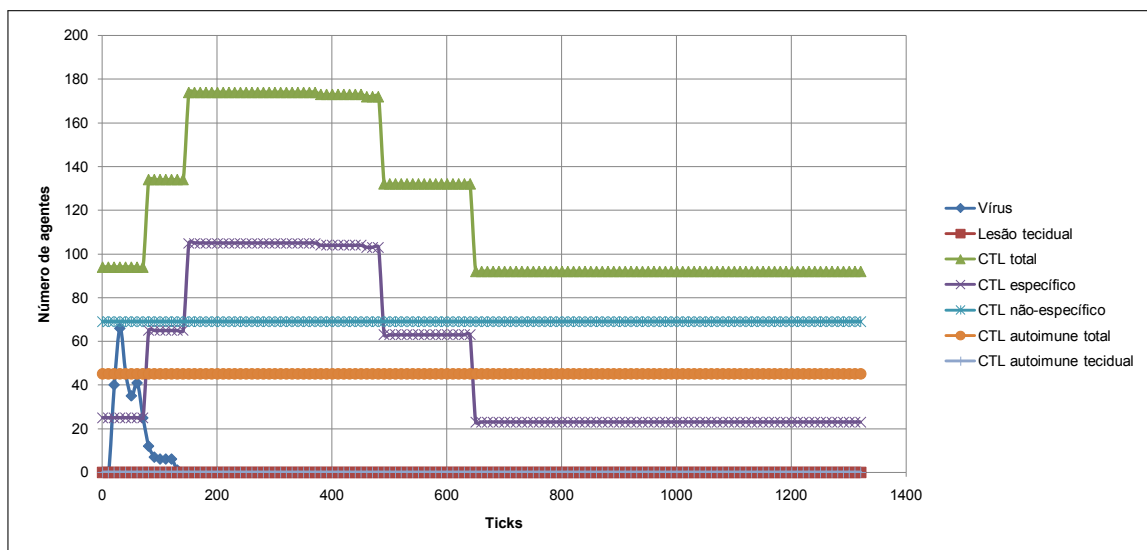


Figura 4.14. Autoimunidade dos linfócitos CTL, sem o mecanismo de tolerância.

É válido ressaltar que os agentes APC são capazes de fagocitar e apresentar qualquer antígeno. Mais ainda, eles são capazes de fagocitar células PC infectadas, e que, nesse caso, apresentam tanto os antígenos do patógeno, quanto os antígenos da própria célula PC, justamente para simular a captura e apresentação de autoantígenos. Neste caso, as células APC têm plena possibilidade de apresentar um autoantígeno capturado em uma *danger zone*.

Apesar disso, analisando a progressão do números de agentes durante a simulação (Figura 4.14), é possível notar que foram gerados linfócitos CTL auto-reativos, mas que estes não foram ativados. A conclusão deste trabalho à respeito desse resultado é que, apesar de ser possível, o SI trabalha sempre no intuito de manter as probabilidades de um

evento ocorrer sempre a seu favor. Neste caso, a probabilidade de uma célula APC capturar um autoantígeno em uma *danger zone*, e conseguir encontrar um linfócito auto-reativo específico para o autoantígeno parece ser muito pequena.

Para baixar o risco de autoimunidade, a imunidade adaptativa é implementada para deflagar um ataque autoimune somente se muitas condições improváveis e independentes ocorrerem. A probabilidade de um ataque autoimune é, portanto, o produto das probabilidades de cada condição em separado, e dessa forma, essa probabilidade é reduzida à valores suficientemente pequenos.

(FLOREANO; MATTIUSI, 2008, p. 344)

Além disso, Benoist e Mathis (2001) afirma que, embora isto ainda esteja longe de ser provado (e daí a motivação deste trabalho), a hipótese do “*mimetismo molecular tem sido a explicação favorita para explicar a tendência de infecções microbianas precipitarem reações autoimunes*” (BENOIST; MATHIS, 2001, p. 797).

Diante de tudo isso, considera-se que o resultado foi satisfatório, e que é coerente com o pensamento científico atual, embora ainda não seja um argumento científico válido para descartar a hipótese, pois a correção do modelo ainda não foi demonstrada.

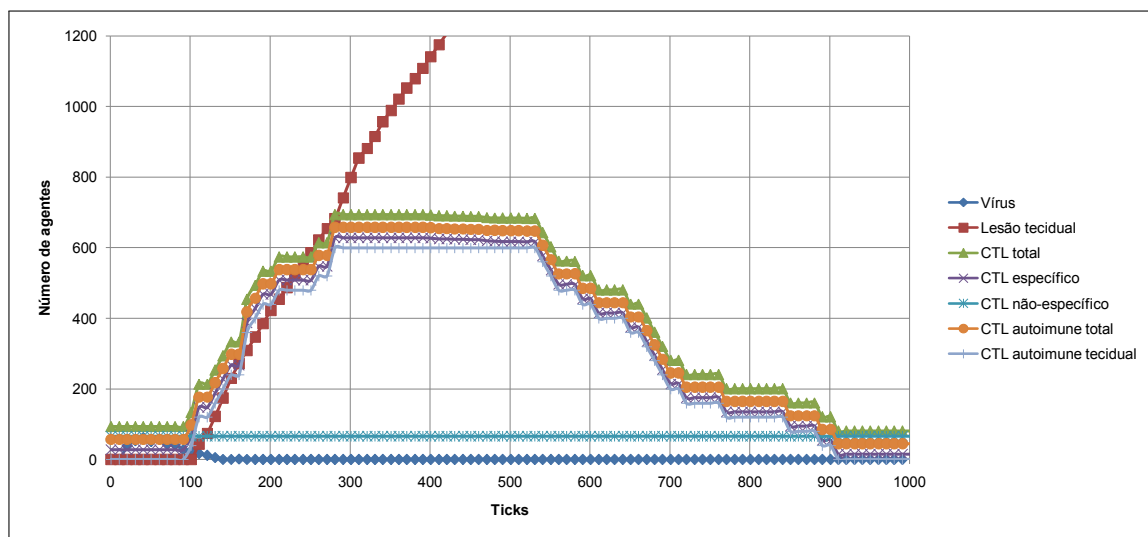


Figura 4.15. Autoimunidade dos linfócitos CTL, sem o mecanismo de tolerância, e com mimetismo molecular.

Por fim, na simulação, ilustrada pela Figura 4.15, além do mecanismo de tolerância do linfócito CTL estar desativado, o padrão do patógeno foi modificado para que fosse similar ao padrão presente nas células PC. Nesta simulação tenta-se demonstrar a Hipótese 2, do mimetismo molecular, que diz que, além dos linfócitos auto-reativos, é necessário que haja uma infecção, e que o agente infeccioso tenha antígenos similares aos antígenos próprios do organismos.

Analisando a progressão do número de agentes durante a simulação (Figura 4.15), é possível notar que, realmente, os linfócitos auto-reativos foram ativados, migraram para o tecido, onde, além de atacarem as células infectadas, atacaram também as células saudáveis, dado informado pela linha “Lesão tecidual”, configurando assim uma doença autoimune.

Como discutido anteriormente, era esperado que esse comportamento emergisse. Portanto, o resultado foi considerado satisfatório, mostrando que o modelo mostrou um comportamento coerente com a literatura, em relação às simulações de comportamentos relacionados à autoimunidade.

Entretanto, sobre a última simulação, é válido ressaltar que, primeiramente, a simulação em si não tem valor científico para corroborar a Hipótese 2 e, em segundo lugar, o fato do padrão molecular do patógeno ter sido alterado para ser similar ao das células PC foi o que determinou o sucesso na geração da doença auto-imune, mas isso é algo muito difícil de ocorrer na natureza, e, embora haja na literatura (BENOIST; MATHIS, 2001) casos de infecções microbianas que estejam ligadas a doenças auto-ímmunes, essa associação ainda não foi provada, justamente pela dificuldade em se ter acesso a esses padrões moleculares candidatos à experimentação.

Foi observado também que, ao matar as células saudáveis, os linfócitos CTL ainda recebiam estímulos para continuar ativos, o que gerava um processo de auto-alimentação, fazendo com que a resposta imunológica autoimune perdurasse por um tempo maior, mesmo depois que os patógenos foram eliminados.

CAPÍTULO 5

CONCLUSÕES

Indo ao encontro da proposta inicial do trabalho, e a partir de estudos e pesquisas realizadas acerca do sistema imunológico, principais métodos de modelagem e arquitetura baseada em agentes, foi proposto um modelo do sistema imunológico, baseado em agentes, que atendeu aos requisitos necessários para a simulação de processos ligados à autoimunidade.

Os objetivos específicos também foram atingidos, uma vez que foi identificado o método mais indicado, pela literatura, para a modelagem do sistema, foram levantados os requisitos necessários para um modelo que permita o estudo de processos autoimunes, o modelo foi implementado em uma ferramenta, chamada Autosimmune, e esta ferramenta foi capaz de apresentar um comportamento coerente com o esperado, segundo a literatura pesquisada.

O modelo apresentado neste trabalho é o primeiro do gênero no Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Viçosa, e seu principal papel foi dar início a um projeto maior que o procederá, cabendo a este evidenciar a viabilidade científica da pesquisa envolvendo a modelagem do sistema imunológico, através da proposta de um modelo inicial e de uma ferramenta que o implementasse, para que se fosse capaz de evidenciar a coerência de seu comportamento.

Após a especificação do modelo, o mesmo foi implementado em uma ferramenta de fácil uso, que pode ser utilizada em diversas plataformas, inclusive gratuitas, principalmente por estudantes da área da Ciência da Computação e Modelagem Computacional, no intuito de extrair lições úteis do processo utilizado durante a modelagem.

Essa ferramenta foi utilizada na simulação de diversos testes, descritos neste trabalho, os quais demonstraram que o modelo apresenta um comportamento coerente com a literatura da área, em especial em relação aos mecanismos relacionados à especificidade e autoimunidade.

Embora o modelo tenha demonstrado estar no caminho certo, apresentando as características que foram descritas como requisito para a simulação da autoimunidade, ele está em sua primeira versão e foi pouco experimentado. Por isso, é importante salientar que os resultados obtidos pela simulação do modelo não têm valor científico para a biologia: servem apenas para evidenciar o comportamento coerente do modelo, e não servem, ainda, como comprovação para as hipóteses da biologia.

As perspectivas a partir de agora são de que os novos ingressantes do Programa de Pós-Graduação em Ciência da Computação da UFV, em conjunto com os demais participantes do recém criado Grupo de Modelagem e Simulação do Sistema Imunológico (ModeSimune), possam dar continuidade ao trabalho, incrementando ainda mais o modelo, e efetuando ainda mais estudos de casos, para estabelecer a correteza do modelo, e permitir que seus resultados possam servir a tantas outras áreas ainda. Em particular, o trabalho do mestrando Carlos Antônio Bastos, que já está engajado na tarefa de acrescentar melhoramentos importantes ao modelo.

Vale mencionar que os resultados oriundos da pesquisa desenvolvida neste trabalho foram publicados e apresentados em congressos da área. Especificamente, a proposta inicial do modelo e seus requisitos foram apresentados no resumo “*A agent-based simulation tool of biological immune system: a case study of autoimmune diseases*” (POSSI *et al.*, 2010), apresentado no *Brazilian Symposium on Bioinformatics* (BSB2010). Já os primeiros resultados da simulação do modelo, especificamente a simulação da resposta inata do modelo, foram publicados no resumo estendido “*An in-silico Immune System Model for Investigating Human Autoimmune Diseases*” (POSSI *et al.*, 2011), apresentado na *XXXVII Conferência Latinoamericana de Informática* (XXXVII CLEI). Os novos resultados deverão ser submetidos à periódicos da área.

5.1 Trabalhos Futuros

Como discutido em diversos pontos do trabalho, a tarefa de modelar o sistema imunológico é algo grandioso, que não caberia neste trabalho. Ao contrário, o que se propôs foi escolher uma pequena parte do sistema, responsável por alguns mecanismos relacionados à autoimunidade, e propor um modelo. Evidentemente que o nível de abstração escolhido excluiu diversos elementos que poderiam contribuir ainda mais para o modelo.

A seguir estão algumas sugestões que este trabalho apresenta para trabalhos futuros envolvendo o modelo em questão:

Relacionar unidades de tempo e espaço com o mundo real: seria muito interessante se as unidades utilizadas no modelo, como tempo e espaço, tivessem uma

correlação com unidades do mundo real, de forma que fosse possível analisar os resultados das simulações do modelo em função de horas ou dias, e não em função de *ticks*. Também seria interessante definir a quantidade, tamanho, velocidade, tempo de vida, quantidade de substância gerada, e constantes de evaporação e difusão em função de dados reais, retirados da literatura. Essas informações ainda são muito escassas, mas existe um grande interesse nesses dados e em um futuro breve, talvez estejam disponíveis;

Tolerância periférica: alguns mecanismos da tolerância periférica, como as células T reguladoras, o processo de anergia, de deleção, não foram implementados. Tais processos podem ter relação com a autoimunidade, e por isso seriam uma contribuição importante para o modelo;

Subtipos de células T: as células T podem se diferenciar em diversos subtipos, como as células T *helper* tipo 1 e tipo 2, que são responsáveis por conduzir a resposta imunológica de acordo com o tipo de patógeno que a iniciou. Como o modelo utiliza apenas um tipo abstrato de vírus, tais células, assim como as diferentes reações dependentes do tipo do patógeno, não foram incluídas;

Outros processos patogênicos: também seria interessante que o modelo pudesse simular outros processos, além dos envolvidos com a autoimunidade. Processos ligados a reações de hipersensibilidade, à rejeição de transplantes, à processos oncológicos, seriam de grande valor. Assim como o acréscimo de outros tipos de patógenos, como bactérias, helmintos e até toxinas, seriam também interessantes;

Homeostasia: o modelo deve conseguir manter seu equilíbrio, seu número médio de células, seu nível normal de substâncias, antes e depois dos processos infecciosos. Esse equilíbrio é chamado de homeostasia. No modelo proposto, a homeostasia ainda é muito pouco desenvolvida, e seria uma boa contribuição acrescentar tais processos;

Maturação de afinidade: um processo interessante envolvendo as células B, chamado de maturação de afinidade, também não foi implementado. Tal processo é muito interessante para a computação inclusive, e poderia ser utilizado em heurísticas baseadas no sistema imunológico;

Desempenho: outro ponto importante nos modelos baseados em agentes é o seu alto custo computacional, que é inclusive indicado como um das suas principais desvantagens. Seria interessante a utilização de métodos, ou até de outros *frameworks*, como o Repast HLC, que possibilitem a simulação ocorrer em um ambiente paralelo e distribuído. Outra contribuição importante seria desenvolver uma ferramenta

que pudesse utilizar processadores gráficos (GPUs) para otimizar alguns processos, como a difusão de substâncias.

REFERÊNCIAS BIBLIOGRÁFICAS

ABBAS, A. K.; LICHTMAN, A. H. *Imunologia Básica: Funções e Distúrbios do Sistema Imunológico*. São Paulo: Revinter, 2003. ISBN 85-7309-729-9.

ABBAS, A. K.; LICHTMAN, A. H. *Imunologia Básica: Funções e Distúrbios do Sistema Imunológico*. 2^a. ed. São Paulo: Elsevier, 2007.

BENOIST, C.; MATHIS, D. Autoimmunity provoked by infection: how good is the case for t cell epitope mimicry? *Nature Immunology*, v. 2, n. 9, p. 797–801, Setembro 2001.

CARVALHO, A. C. P. L. F.; BRAYNER, A.; LOUREIRO, A.; FURTADO, A. L. *Grandes Desafios da Pesquisa em Computação no Brasil – 2006 – 2016*. São Paulo, maio 2006.

CASTRO, L. N. *Engenharia Imunológica: Desenvolvimento e Aplicação de Ferramentas Computacionais Inspiradas em Sistemas Imunológicos Artificiais*. Tese (Doutorado) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, Maio 2001.

CELADA, F.; SEIDEN, P. E. A computer model of cellular interactions in the immune system. *Immunol Today*, v. 13, p. 56–62, 1992.

COLLIER, N. T. *Repast Symphony Reference*. [S.l.], 2010. Disponível em: <<http://repast.sourceforge.net/docs/RepastReference.pdf>>. Acesso em: 07/01/2012.

FLOREANO, D.; MATTIUSI, C. *Bio-inspired artificial intelligence : theories, methods, and technologies*. [S.l.]: MIT Press, 2008. (Intelligent robotics and autonomous agents series). ISBN 978-0-262-06271-8.

FOLCIK, V.; AN, G.; OROSZ, C. The basic immune simulator: an agent-based model to study the interactions between innate and adaptive immunity. *Theoretical biology & medical modelling*, v. 4, p. 39, 2007.

GRASER, R. T.; DILORENZO, T. P.; WANG, F.; CHRISTIANSON, G. J.; CHAPMAN, H. D.; ROOPENIAN, D. C.; NATHENSON, S. G.; SERREZE, D. V. Identification of a cd8 t cell that can independently mediate autoimmune diabetes development in the

complete absence of cd4 t cell helper functions. *The Journal of Immunology*, v. 164, p. 3913–3918, 2000. ISSN 1550-6606.

HÜBNER, J. F.; BORDINI, R. H.; VIEIRA, R. *Introdução ao Desenvolvimento de Sistemas Multiagentes com Jason*. [S.l.], 2004. Disponível em: <<http://www.das.ufsc.br/~jomi/pubs/2004/Hubner-eriPR2004.pdf>>. Acesso em: 20/12/2011.

JANEWAY, C. A.; TRAVERS, P.; WALPORT, M. *Immunobiology: The Immune System in Health and Disease*. 5^a edição. ed. New York: Garland Science, 2001.

LI, X.; WANG, Z.; LU, T.; CHE, X. Modelling immune system: Principles, models, analysis and perspectives. *Journal of Bionic Engineering*, v. 6, n. 1, p. 77–85, 2009.

MACAL, C. M.; NORTH, M. J. Agent-based modeling and simulation. In: *Proceedings of the 2009 Winter Simulation Conference*. [S.l.: s.n.], 2009. p. 86–98.

NORTH, M. J.; HOWE, T. R.; COLLIER, N. T.; VOS, J. R. The repast symphony development environment. In: *Proceedings of the Agent 2005 Conference on Generative Social Processes, Models, and Mechanisms*. [S.l.: s.n.], 2005.

PORTER, D. L.; LEVINE, B. L.; KALOS, M.; BAGG, A.; JUNE, C. H. Chimeric antigen receptor–modified t cells in chronic lymphoid leukemia. *The New England Journal of Medicine*, n. 365, p. 725–733, 2011.

POSSI, M. A.; OLIVEIRA, A. P.; CHAVES, C. M. G.; CERQUEIRA, F. R.; ARROYO, J. E. C. An in-silico immune system model for investigating human autoimmune diseases. In: *XXXVII Conferencia Latinoamericana de Informática (XXXVII CLEI)*. Quito, Equador: [s.n.], 2011.

POSSI, M. A.; OLIVEIRA, A. P.; Di Iorio, V. O.; DIAS, C. M. G. C. A agent-based simulation tool of biological immune system: a case study of autoimmune diseases. In: *Brazilian Symposium on Bioinformatics*. Búzios, Brazil: [s.n.], 2010.

RAPIN, N.; LUND, O.; BERNASCHI, M.; CASTIGLIONE, F. Computational immunology meets bioinformatics: The use of prediction tools for molecular binding in the simulation of the immune system. *PLoS ONE*, v. 5, n. 4, abr. 2010.

ROITT, I. M.; DELVER, P. J. *Roitt's Essential Immunology*. [S.l.]: Blackwell Science, 2001. ISBN 0-632-05902-8.

SAWCER, S.; HELLENTHAL, G.; PIRINEN, M.; SPENCER, C. C. A.; DONNELLY, P.; COMPSTON, A. Genetic risk and a primary role for cell-mediated immune mechanisms in multiple sclerosis. *Nature*, n. 476, p. 214–219, 2011. ISSN 0028-0836.

SNOWDEN, R. V. Fda approves prostate cancer vaccine. *American Cancer Society News Center*, 2009. Disponível em: <http://www.cancer.org/docroot/NWS/content/NWS_1_1x_FDA_Approves_Prostate_Cancer_Vaccine.asp>. Acesso em: 20/10/2011.

TAY, J. C.; JHAVAR, A. Cafiss: A complex adaptive framework for immune system simulation. In: *Proceeding of the 2005 ACM Symposium on Applied Computing (SAC'05)*. Santa Fe, New Mexico: [s.n.], 2005. p. 158–164.

WEISS, G. (Ed.). *Multiagent systems: a modern approach to distributed artificial intelligence*. [S.l.]: The MIT Press, 1999. ISBN 0-262-23203-0.

WOOLDRIDGE, M.; JENNINGS, N. R. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, v. 2, n. 10, p. 115–152, 1995.

WOOLDRIDGE, M. J. *An introduction to multiagent systems*. 1. ed. [S.l.]: John Wiley & Sons, 2001. ISBN 047149691X.