

**UNIVERSIDADE CATÓLICA DE SANTOS  
MESTRADO EM INFORMÁTICA**

**DISSERTAÇÃO DE MESTRADO**

**INTEGRAMODEL: UM SERVIÇO DE SIMULAÇÃO DE EPIDEMIAS  
EM GRADES COMPUTACIONAIS**

**HENRIQUE FABRÍCIO GAGLIARDI**

**SANTOS  
2007**

**HENRIQUE FABRICIO GAGLIARDI**

**INTEGRAMODEL: UM SERVIÇO DE SIMULAÇÃO DE EPIDEMIAS  
EM GRADES COMPUTACIONAIS**

**Dissertação de mestrado apresentada à  
Universidade Católica de Santos, sob  
orientação do Prof. Dr. Fabrício Alves  
Barbosa da Silva e co-orientação do  
Prof. Dr. Domingos Alves, para  
obtenção do grau de mestre em Ciência  
da Computação na linha de Sistemas  
Distribuídos.**

**SANTOS  
2007**

*Dedicado àquele que esteve sempre presente em todos os momentos da minha vida e o maior mestre que já conheci. Muito obrigado por tudo, meu pai.*

## **AGRADECIMENTOS**

Mais uma vez, muitos são os agradecimentos que se fazem necessários para o desenvolvimento deste trabalho. Assim, agradeço:

Ao professor Fabrício, orientador e amigo, cujas horas de discussão de idéias renderam dois produtivos anos de vitórias profissionais e um aprendizado único como ser humano. Em minha humilde opinião, considero-o um exemplo de pesquisador sério no Brasil e acredito que todos nós temos muito a aprender com seu exemplo de profissionalismo.

Ao professor Domingos, amigo e empenhado companheiro de guerra cuja colaboração durante estes cinco anos rendeu grandes conquistas e alegrias. Em particular, ter a honra de ser co-orientado por um pesquisador tão apaixonado por seus ideais é algo que contagia, e nos faz lutar cada vez mais para conquistar também cada um de nossos sonhos.

Aos professores Ivan Torres Pisa e Hermes Senger, por aceitarem participar como membros da banca examinadora deste trabalho. Serei sempre muito grato por suas valiosas recomendações que contribuíram para o engrandecimento deste trabalho.

Ao professor Manoel Maria de Jesus Mendes por ter participado da banca de qualificação deste trabalho, contribuindo com sua experiência ao tecer sugestões e críticas sempre muito pertinentes.

Ao pessoal do grupo do laboratório da sala 402 (LCC-ASC): Juliane, Daniela, Kamilayne, Raquel, Fernanda e tantos outros que contribuíram para que as tardes fossem mais bonitas e agradáveis. Agradeço demais pela amizade de todas vocês.

À Maria Antónia e sua família pelo carinho e amizade. Seu exemplo de coragem, companheirismo e força de vontade nos motiva a lutar com todo o nosso empenho para cumprir com nossas metas.

Ao Eduardo Gallo pela amizade e as longas discussões sobre integração de bases de dados, mediadores, servidores de metadados e as demais conversas fáceis e descontraídas que povoaram o cotidiano destes dois anos de muito trabalho e dedicação.

E, finalmente, à CAPES pela bolsa de mestrado concedida para a realização deste trabalho durante estes últimos dois anos.

## **RESUMO**

O desenvolvimento de ferramentas para previsão e monitoramento do avanço de doenças infecto-contagiosas tornou-se atualmente uma necessidade por prover cenários e oferecer informações pertinentes à tomada de decisões por parte dos profissionais de saúde de maneira mais eficiente e eficaz. Diante disso, diversos modelos epidêmicos vêm sendo propostos, baseados em abordagens compartimentais (baseadas em equações diferenciais) ou baseadas no indivíduo (com uma rede local de contatos). Contudo, modelar a heterogeneidade das interações entre indivíduos ainda é um desafio para os epidemiologistas e, de fato, desenvolver modelos preditivos capazes de apresentar o comportamento de uma doença em particular baseado em dados reais ainda representa um desafio ainda maior. Além disso, um ambiente computacional robusto capaz de efetuar tais simulações também se torna uma necessidade nesse caso. Assim, a proposta deste trabalho é desenvolver um serviço de simulação para prever a dinâmica do espalhamento da dengue utilizando para tal uma plataforma de grades computacionais.

**Palavras-chave:** Epidemiologia, Autômatos Celulares, Grades Computacionais.

## **ABSTRACT**

The development of tools for forecasting and monitoring the spreading of infect-contagious diseases become a current need to provide scenarios and pertinent decision support information which would be used efficiently and efficaciously by health officers. Hence, several epidemics models have being proposed based on compartmental approaches (using differential equations) or on individual based approaches (with a local network of contacts). However, modeling the heterogeneity of interactions among individuals still represents a challenge for epidemiologists and, in fact, to develop such forecasting models capable to represent the behavior of a particular infectious disease is a even greater challenge. Moreover, a robust computational environment capable to realize such simulations also becomes a need in this case. Therefore, the main objective of this work is to develop a simulation service to foresee the dynamics of the urban dengue fever spreading on a grid computing platform.

**Keywords:** Epidemiology, Cellular Automata, Grid Computing.

## SUMÁRIO

INTRODUÇÃO.....	1
CAPÍTULO 1 – PLATAFORMAS DE GRADES COMPUTACIONAIS: INTRODUÇÃO E CONCEITOS FUNDAMENTAIS .....	5
1.1 INTRODUÇÃO HISTÓRICA ÀS GRADES COMPUTACIONAIS .....	9
1.2 CONCEITOS FUNDAMENTAIS DA GRADE COMPUTACIONAL.....	15
1.3 SERVIÇOS GRID.....	16
1.4 OPEN GRID SERVICES ARCHITECTURE (OGSA).....	17
1.5 O PROJETO GLOBUS.....	19
1.5.1 A arquitetura do Globus Toolkit 4.0.....	20
1.5.2 Os componentes do Globus Toolkit 4.0 .....	21
1.5.3 Os componentes comuns para execução .....	22
1.5.4 Os componentes de segurança.....	23
1.5.5 Os serviços de informação.....	24
1.5.6 Os serviços de acesso a dados .....	26
1.5.6.1 O GridFTP .....	26
1.5.6.2 RFT: a transferência de arquivos confiável.....	27
1.5.6.3 Replica Location Service (RLS).....	28
1.5.6.4 Data Replication Service (DRS).....	30
1.5.6.5 OGSA Data Access and Integration (OGSA-DAI).....	30
1.5.7 Os componentes de execução .....	31
1.5.7.1 WS GRAM .....	31
CAPÍTULO 2 – TÉCNICAS DE MODELAGEM DE PROCESSOS EPIDÊMICOS: UMA VISÃO GERAL.....	37

2.1	MODELOS EPIDÊMICOS COMPARTIMENTAIS .....	38
2.2.	MODELOS BASEADOS NO INDIVÍDUO .....	47
2.2.1	Pressupostos e construção de um autômato celular.....	49
2.2.1.1	Os estados do autômato celular e as regras de transição .....	52
2.2.2	O modelo probabilístico alternativo baseado em autômatos celulares básico para a disseminação de uma epidemia.....	54
2.2.2.1	Características do modelo alternativo baseado em autômatos celulares para uma população apenas.....	55
2.2.2.2	O modelo SIR na abordagem baseada em autômatos celulares .....	58
CAPÍTULO 3 – A MODELAGEM DO PROCESSO EPIDÊMICO DA DENGUE: SIMULAÇÃO E REALIDADE .....		68
3.1	UM MODELO PARA ESTUDAR A TRANSMISSÃO DA DENGUE.....	71
3.2	RESULTADOS PARTICULARES DO MODELO DA DENGUE .....	77
3.3	O MODELO DA DENGUE EM CIDADES VIRTUAIS.....	90
3.4	CONCLUSÕES E COMENTÁRIOS GERAIS.....	93
CAPÍTULO 4 – O SISTEMA DE PREVISÃO DE EPIDEMIAS INTEGRAMODEL .....		95
4.1	O PROJETO INTEGRAEPI .....	96
4.1.1	A arquitetura do sistema IntegraEPI .....	99
4.1.1.1	O serviço de acesso e integração de dados.....	100
4.1.1.2	Os serviços de análise IntegraAnalysis .....	103
4.1.1.3	O serviço de simulação IntegraModel.....	104
4.2	A ARQUITETURA DO SERVIÇO DE SIMULAÇÃO INTEGRAMODEL.....	105
4.3	CONSTRUINDO UM CLIENTE PARA O SERVIÇO INTEGRAMODEL.....	113
4.4	CONCLUSÕES E COMENTÁRIOS GERAIS .....	126

CAPÍTULO 5 – ESTUDO DE CASO: SIMULANDO A DENGUE PELO SISTEMA INTEGRAMODEL .....	129
CAPÍTULO 6 – CONCLUSÕES E PERSPECTIVAS .....	135
APÊNDICE A – O SIMULADOR DO MODELO DA DENGUE.....	139
REFERÊNCIAS BIBLIOGRÁFICAS .....	144

## LISTA DE FIGURAS

FIGURA 1.1 – ESQUEMA DA ARQUITETURA DO GLOBUS TOOLKIT MOSTRANDO ALGUNS DE SEUS COMPONENTES .....	20
FIGURA 1.2 – OS COMPONENTES DO GLOBUS TOOLKIT 4.0 ORGANIZADOS EM CATEGORIA E TIPO .....	21
FIGURA 1.3 – ESQUEMA DA RELAÇÃO ENTRE O JAVA WS CORE E SERVIÇOS DO GT4 .....	22
FIGURA 1.4 – ESQUEMA DE PUBLICAÇÃO DE RECURSOS NO MDS .....	25
FIGURA 1.5 – UMA TRANSFERÊNCIA ‘TERCEIRIZADA’ NO GRIDFTP .....	27
FIGURA 1.6 – DIAGRAMA ILUSTRATIVO DO FUNCIONAMENTO DA RFT .....	28
FIGURA 1.7 – A EXECUÇÃO DE UMA TAREFA PELO WS GRAM UTILIZANDO STAGING.....	33
FIGURA 1.8 – ESQUEMA DE SUBMISSÃO DE TAREFAS EM GRADES COMPUTACIONAIS DETALHANDO AS ETAPAS DE STAGING ....	35
FIGURA 2.1 – O MODELO SIR .....	40
FIGURA 2.2 – FRAÇÃO INFECTADA DA POPULAÇÃO EM UMA EPIDEMIA EM FUNÇÃO DO NÚMERO DE REPRODUÇÃO BÁSICA $R_0$ .....	42
FIGURA 2.3 – DESENHO ESQUEMÁTICO DE (A) UMA REDE UNIDIMENSIONAL, (B) UMA REDE BIDIMENSIONAL E (C) UMA REDE TRIDIMENSIONAL .....	49
FIGURA 2.4 – EXEMPLOS DE REDES COM VÁRIAS GEOMETRIAS .....	50
FIGURA 2.5 – ESQUEMA DOS TIPOS DE VIZINHANÇA .....	51
FIGURA 2.6 – A VIZINHANÇA DE VON NEUMANN (A) E A VIZINHANÇA DE MOORE (B) .....	51

FIGURA 2.7 –	CONDIÇÕES PERIÓDICAS DE CONTORNO EM UMA REDE UNIDIMENSIONAL E EM UMA REDE BIDIMENSIONAL.....	52
FIGURA 2.8 –	ESQUEMA DE UMA REDE QUADRADA BIDIMENSIONAL, DESTACANDO OS TRÊS DIFERENTES ESTADOS PARA O MODELO SIR, TOMADO AQUI COMO EXEMPLO.....	53
FIGURA 2.9 –	O ESPALHAMENTO GLOBAL E LOCAL EM UM AUTÔMATO CELULAR BIDIMENSIONAL ALTERNATIVO.....	56
FIGURA 2.10 –	AS TRANSIÇÕES DO MODELO SIR DISCRETO MODELADAS PELO AUTÔMATO CELULAR COM UMA COMPONENTE GLOBAL E UMA LOCAL.....	58
FIGURA 2.11 –	MODELO SIR DISCRETO UTILIZANDO-SE O ALGORITMO DESCRITO PARA O AUTÔMATO CELULAR.....	59
FIGURA 2.12 –	O MESMO QUE NA FIGURA 2.11, COM $\Gamma = 0.1$ E $\Lambda = 0.9$ .....	61
FIGURA 2.13 –	GRÁFICOS OBTIDOS A PARTIR DE 4 SIMULAÇÕES REALIZANDO 100 EXPERIMENTOS PARA CADA UMA COMEÇANDO COM APENAS UM INFECTADO NO CENTRO DA REDE.....	63
FIGURA 2.14 –	O MESMO QUE NA FIGURA 2.13, COMEÇANDO COM 10 INFECTADOS E 35% DA REDE SUSCEPTÍVEL.....	65
FIGURA 3.1 –	O ESQUEMA DO MODELO DE TRANSMISSÃO DA DENGUE PROPOSTO, REPRESENTANDO OS ESTADOS DA DOENÇA.....	73
FIGURA 3.2 –	ESQUEMA DA INTERAÇÃO ENTRE DOIS AUTÔMATOS REPRESENTANDO A POPULAÇÃO DE HUMANOS E A DE MOSQUITOS.....	74
FIGURA 3.3 –	SIMULAÇÃO DO MODELO BASEADO NA INTERAÇÃO ENTRE DOIS AUTÔMATOS.....	76

FIGURA 3.4 –	SIMULAÇÕES PARA $\lambda_h = 0,01$ CONFIGURADAS COMO DESCRITAS NA LEGENDA PARA A POPULAÇÃO DE HUMANOS.....	81
FIGURA 3.5 –	SIMULAÇÕES PARA $\lambda_h = 0,02$ , EM (A) E $\lambda_h = 0,03$ EM (B). AQUI OS DOIS RESULTADOS SÃO DISPOSTOS AO MESMO TEMPO PARA EFEITO DE COMPARAÇÃO .....	85
FIGURA 3.6 –	SIMULAÇÕES PARA $\lambda_h = 0,04$ , EM (A), $\lambda_h = 0,05$ EM (B) E $\lambda_h = 0,1$ EM (C) .....	89
FIGURA 3.7 –	ESQUEMA DE UMA REDE BASEADA NA GEOMETRIA DA CIDADE DE SANTOS-SP, NA REGIÃO METROPOLITANA DA BAIXADA SANTISTA.....	92
FIGURA 3.8 –	UMA SIMULAÇÃO COM UMA REDE USANDO A GEOMETRIA DO MUNICÍPIO DE SANTOS-SP .....	93
FIGURA 4.1 –	A ARQUITETURA DO SISTEMA INTEGRAEPI APRESENTADA DE MANEIRA SIMPLIFICADA .....	100
FIGURA 4.2 –	ESQUEMA ILUSTRATIVO DO SISTEMA DE MONITORAMENTO DE EPIDEMIAS FAZENDO USO DOS SERVIÇOS DE ANÁLISE DO INTEGRAEPI .....	104
FIGURA 4.3 –	O DIAGRAMA DE CLASSES DO SERVIÇO INTEGRAMODEL.....	106
FIGURA 4.4 –	A EXECUÇÃO DO SERVIÇO INTEGRAMODEL .....	108
FIGURA 4.5 –	O DIAGRAMA DE CASOS DE USO GERAL DO SISTEMA INTEGRAMODEL.....	115
FIGURA 4.6 –	O DIAGRAMA DE CASOS DE USO DO SISTEMA INTEGRAMODEL PARA A SUBMISSÃO DE SIMULAÇÕES .....	117
FIGURA 4.7 –	O DIAGRAMA DE CASOS DE USO DO SISTEMA INTEGRAMODEL	

	PARA VISUALIZAR RESULTADOS.....	118
FIGURA 4.8 –	O DIAGRAMA DE CLASSES DA APLICAÇÃO CLIENTE.....	119
FIGURA 4.9 –	ESQUEMA DE UM ARQUIVO CONTENDO A DESCRIÇÃO DE UM MAPA.....	120
FIGURA 4.10 –	ESQUEMA DE CONVERSÃO DE UMA FIGURA COM UM MAPA PARA UM ARQUIVO XML.....	121
FIGURA 4.11 –	A ESTRUTURA DO XML SCHEMA PARA O MODELO DE DADOS COMUM.....	122
FIGURA 4.12 –	A ESTRUTURA DO XML SCHEMA DOS RESULTADOS DO MODELO DA DENGUE.....	123
FIGURA 4.13 –	ESQUEMA DA EXECUÇÃO DA APLICAÇÃO CLIENTE.....	124
FIGURA 5.1 –	A TELA INICIAL DO CLIENTE.....	129
FIGURA 5.2 –	A TELA DE CONFIGURAÇÃO DE SIMULAÇÕES PARA A PREVISÃO DE EPIDEMIAS.....	130
FIGURA 5.3 –	OS RESULTADOS SIMULADOS DA DENGUE PARA A CIDADE DE SANTOS, VISUALIZADOS COMO UMA ANIMAÇÃO NO CLIENTE DO SISTEMA INTEGRAMODEL.....	132
FIGURA 5.4 –	UMA SIMULAÇÃO DA DENGUE EM GUARUJÁ SOBRE DOIS MAPAS (SETORES CENSITÁRIOS E LIMITE DE MUNICÍPIO) E APENAS SOBRE UM MAPA DO LIMITE DE MUNICÍPIO.....	133

## LISTA DE TABELAS

TABELA 1 –	LISTA DE ARGUMENTOS DE ENTRADA DO SIMULADOR (QUE NÃO SE REFERE A UMA POPULAÇÃO PARTICULAR) .....	140
TABELA 2 –	LISTA DE ARGUMENTOS DE ENTRADA PARA A POPULAÇÃO DE HUMANOS .....	140
TABELA 3 –	LISTA DE ARGUMENTOS DE ENTRADA PARA A POPULAÇÃO DE MOSQUITOS .....	141
TABELA 4 –	LISTA DE ARGUMENTOS DE ENTRADA PARA CONFIGURAÇÃO DA REDE E SAÍDA DA SIMULAÇÃO .....	142

## LISTA DE ARTIGOS PUBLICADOS RELACIONADOS AO PRESENTE

### TRABALHO

- ALVES, D., **GAGLIARDI, H. F.**, “Técnicas de Modelagem de Processos Epidêmicos e Evolucionários” in São Carlos, SP: Notas em Matemática Aplicada, v.26, 88p, 20,5cm, 2006, disponível em: <http://www.sbmac.org.br/notas.php>, acessível em 19/12/2006.
- **GAGLIARDI, H. F.**, SILVA, F. A. B., ALVES, D., “Automata networks applied to the epidemiology of urban dengue fever” in International Conference on Computational Science (ICCS 2006) – Workshop on Modelling of Complex Systems by Cellular Automata, 2006, Reading, UK. Lectures Notes in Computational Science. V.N. Alexandrov et al. (Eds.): ICCS 2006 Part III, LNCS 3993, p.297-304. Springer-Verlag Berlin Heidelberg 2006.
- SILVA, F. A. B., **GAGLIARDI, H. F.**, GALLO, E., CAVICCHIOLI NETO, V. ; PISA, I. T. ; ALVES, D. “IntegraEPI: a Grid-based Epidemic Surveillance System”. In: Healthgrid 2007 Conference, 2007, Genebra. Proceeding of Healthgrid 2007 (accepted for publication). Amsterdã : IOS Press, 2007.
- CAVICCHIOLI, V., **GAGLIARDI, H. F.**, REZENDE, A. S., GALLO, E., GONCALVES, E. S., SILVA, F. A. B., ALVES, D. “Data Access Service in a Computational Grid Platform Applied to the Monitoring and Control of Epidemics on Georeferenced Dynamic Maps” In: 2nd International IEEE Conference on e-science and the Grid, 2006, Amsterdam. Proceedings of the 2nd International IEEE Conference on e-science and the Grid.
- REZENDE, A. S., **GAGLIARDI, H. F.**, SILVA, F. A. B., ALVES, D. “IntegraEpi-GIS: A Geographic Information System to visualize and analyze the spatio-temporal patterns of the spread and control of epidemics” In: Spatial Epidemiology Conference, 2006, Londres. Proceedings Spatial Epidemiology Conference, 2006.

- ALVES, D., **GAGLIARDI, H. F.**, SILVA, F. A. B. “Epidemiology in a computational laboratory: preliminary validation, and samples of simulation results” In: Proceedings of 11th World Congress on Public Health, Rio de Janeiro 2006.
- FURLAN, L. B., **GAGLIARDI, H. F.**, SILVA, F. A. B., PISA, I. T. ; ALVES, D. . “MicroCASim: an automata network simulator applied to the competition between microparasites and host immune response” In: International Conference on Computational Science - Workshop on Modelling of Complex Systems by Cellular Automata, 2006, Reading. Lecture Notes on Computer Science, 2006.
- GALLO, E., **GAGLIARDI, H. F.**, SILVA, F. A. B., CAVICCHIOLI NETO, V., ALVES, D. GISE: A Data Access and Integration Service of Epidemiological Data for a Grid-Based Monitoring and Simulation System. Trabalho aceito para apresentação oral na 40th Annual Simulation Symposium, Norfolk, Virgínia-EUA, 2007.
- **GAGLIARDI, H. F.** ; ALVES, D. Redes Complexas e Modelagem de Epidemias. In: XXVIII Congresso Nacional de Matemática Aplicada e Computacional, 2005, SÃO PAULO. Anais do XXVIII Congresso Nacional de Matemática Aplicada e Computacional, 2005.
- MUCHA, M. ; RIBERTO, L. ; SERAFIM, R. C. ; CAVICCHIOLI NETO, V. ; **GAGLIARDI, H. F.** ; ALVES, D. . Análise Espaço-Temporal de um Mapa Ambiental Dinâmico com os Casos de dengue em Ribeirão Preto (SP). Saúde e Sociedade, v. 14, n. 1, p. 263, 2005.
- RIBERTO, L. ; MUCHA, M. ; SERAFIM, R. C. ; CAVICCHIOLI NETO, V. ; **GAGLIARDI, H. F.** ; ALVES, D. . Construção de indicadores Ambientais de risco de ocorrência e disseminação da dengue em mapas dinâmicos de Ribeirão Preto. Saúde e Sociedade, v. 14, n. 1, p. 264, 2005.

## INTRODUÇÃO

O desenvolvimento de modelos epidêmicos vem ganhando um enfoque cada vez maior atualmente por abordar um assunto de grande relevância no aspecto social como é o controle epidemiológico. Nesse sentido, mecanismos capazes de entender a dinâmica e prever o comportamento de determinada moléstia são cada vez mais desejados, de uma maneira geral, porque através deles podemos testar o impacto causado pelos parâmetros inerentes àquele tipo de epidemia. Entretanto, não existe uma solução única para modelar a dinâmica de uma epidemia. A solução ideal para cada caso deve ser buscada tentando responder o porquê, onde e quando do fenômeno epidêmico, através da integração das escalas temporal e espacial articulada com o modelo matemático e computacional definido para descrever a epidemia e prever sua evolução [PATERSON, VINEY, 2000]. Além disso, para que esses modelos possam ser utilizados, de forma preditiva, para auxiliar e avaliar uma tomada de decisão em saúde em nível municipal, estes devem ser cuidadosamente parametrizados e validados com dados epidemiológicos colhidos no local [LUZ et al, 2003].

Com efeito, a maioria dos modelos matemáticos e computacionais de surtos epidêmicos descrevem apenas as interações de grandes grupos de pessoas, vistos em conjunto [ALVES, HASS, CALIRI, 2003]. Ao se construir tais modelos não se leva em consideração o sinergismo existente entre os processos sociais e os ecossistemas sobre o espalhamento de doenças contagiosas específicas, ou não contam com a heterogeneidade de interação das doenças infecciosas com populações humanas dinâmicas, inseridas em estruturas urbanas complexas [ALVES, HASS, CALIRI, 2003; EUBANK, 2002; EUBANK et al, 2004]. Ora, as redes que sustentam o cotidiano e geram empregos, infra-estrutura de transportes, necessidades e luxos são as mesmas que as doenças infecciosas exploram para se espalhar

entre hospedeiros humanos. Modelando-as com certo detalhamento, podemos entender suas estruturas e descobrir como alterá-las para deter o alastramento das doenças, ao mesmo tempo em que infligimos o menor dano possível ao tecido da sociedade [EUBANK, 2002; ALVES, HASS, CALIRI, 2003].

Assim, para se tentar deter um surto epidêmico, diante de recursos e pessoal limitados de um serviço de saúde, é importante o desenvolvimento de um ambiente em que cenários do tipo “o que aconteceria se” possam ser simulados e testados da forma o mais realista possível antes que uma epidemia se instale [EUBANK, 2002]. Para que o desenvolvimento de tal ambiente venha ampliar a capacidade da vigilância epidemiológica no controle das doenças transmissíveis é necessário que ele esteja acoplado a ferramentas computacionais capazes de analisar e reconhecer padrões espaço-temporais provenientes de fontes diversas, em diferentes épocas, em múltiplas escalas; incorporando aspectos ambientais, identificadores de riscos [BAILEY e GATRELL, 1995] disponibilizados em bancos de dados necessários para os estudos de epidemias.

Daí, o grande desafio a que se propõe este projeto é oferecer um ambiente computacional capaz de reconhecer, monitorar e visualizar esses padrões de processos epidêmicos ocorrendo em um município, integrado a um laboratório em que se possa simular em detalhes as interações sociais complexas, em um ambiente urbano e ambiental de maneira realística.

Obviamente, que a integração desse conjunto de bases de dados e métodos de análise espaço-temporais com um modelo computacional baseado em padrões de contatos de indivíduos e estruturas urbano-ambientais reais exige um ambiente computacional de alto desempenho, o que poderia inviabilizar o projeto do ponto de vista técnico. Assim, escolhemos aqui a opção de se propor, implementar e avaliar uma plataforma de grade

computacional para ser o repositório dos resultados tecnológicos, os algoritmos e produtos de software intermediários, que foram construídos ao longo deste projeto.

O conceito de grade computacional surgiu historicamente como uma plataforma para provisionamento e gerência de recursos computacionais distribuídos para execução de aplicações científicas, que demandam grande poder computacional ou a utilização de equipamentos de uso específico na ciência [FOSTER, 2002]. Recentemente o enfoque tem mudado com o objetivo de transformar a grade computacional em uma plataforma conveniente para disponibilização de serviços em uma área geograficamente ampla. Estes serviços incluem acesso a recursos, serviços de segurança/autenticação, serviços de informações etc. Os serviços na grade foram originalmente projetados para integração de recursos distribuídos e geograficamente dispersos, levando-se em conta a heterogeneidade dos mesmos e a ocorrência de falhas.

Diante deste cenário, o objetivo deste trabalho é desenvolver uma primeira versão do mecanismo de previsão de epidemias do sistema IntegraEPI baseando-se em modelos epidêmicos para que possam ser efetuadas simulações a partir de dados reais disponíveis. Essas simulações realizadas dentro de um ambiente de grades computacionais, oferecerá aos agentes de saúde a capacidade de testar a eficácia de diferentes reações antes que surtos reais aconteçam ou mesmo decidir sobre um conjunto de contramedidas possíveis que podem ser estudadas previamente em uma população virtual modelada a partir de dados reais [EUBANK et al, 2004; BAILEY e GARTELL, 1995; AXELSON, 1999].

Assim, o presente trabalho está organizado da seguinte forma: no capítulo 1 são apresentados vários conceitos importantes sobre grades computacionais que servirão como base para a discussão sobre a implementação do sistema. Em seguida, no capítulo 2 é feito um levantamento bibliográfico sobre a modelagem de epidemias, apresentando conceitos e técnicas freqüentemente usadas, aproveitando para apresentar um estudo do modelo que

servirá como base para o modelo da dengue, apresentado no capítulo 3. O capítulo 4 apresenta o sistema de simulação de epidemias e por fim, o capítulo 5 apresenta um estudo de caso deste sistema.

## **CAPÍTULO 1 – PLATAFORMAS DE GRADES COMPUTACIONAIS: INTRODUÇÃO E CONCEITOS FUNDAMENTAIS**

Apesar de relativamente novas, as plataformas de grades computacionais são ambientes distribuídos que vêm ganhando destaque nos dias atuais por uma série de razões que estarão sendo discutidas ao longo deste capítulo. Suas aplicações variam do âmbito científico a aplicações na área da saúde e ensino a distância, destacando-se assim como uma tecnologia de grande importância no cenário mundial.

O termo *grid* (de *grid computing*) surgiu em meados da década de 90 no meio acadêmico e fazia uma analogia às redes elétrica e de abastecimento de água (*power grid* e *water grid*, em inglês), termo que no Brasil foi traduzido como ‘grade’, apesar do sentido da tradução mais correta ser ‘rede’. Ocorre que ‘rede’ já se refere a outro campo de estudo completamente distinto, sendo adotado, portanto o termo ‘grade computacional’ no Brasil para distinguir esse novo conceito.

Ainda nessa analogia, a idéia central era a de que tal sistema proveria recursos da mesma forma que em uma rede de distribuição de energia elétrica, ou seja, para se obter a eletricidade bastava que um eletrodoméstico fosse conectado a uma tomada, sem necessidade de se preocupar onde ou como ela foi gerada. Nesse caso, bastaria que um usuário (ou aplicação) acessasse tal sistema de grade computacional para prover dos recursos que ali se dispõe, sem necessidade de saber onde eles estão ou quais tecnologias ou linguagens, hardware ou sistemas operacionais que ali se encontram envolvidas. De fato, tal visão se apresenta como o objetivo final de tal abordagem, mas ainda se fazem necessários vários passos para isso, devendo ser considerados vários outros padrões em aspectos tanto tecnológicos quanto políticos ou de negócios para que esta visão se torne uma realidade.

Durante os últimos anos, com a evolução das grades computacionais e o começo da comercialização desta tecnologia por alguns, o conceito de grades computacionais sofreu algumas alterações do original, podendo ser visto hoje como um conjunto de recursos coordenados, controlados de maneira descentralizada através de interfaces e protocolos abertos, padronizados e de propósito geral com o intuito de prover serviços a seus usuários [FOSTER, 2002].

Atualmente grande parte do interesse neste tipo de tecnologia consiste no fato da grade computacional ser considerada uma tecnologia “sem limites”, uma vez que podem ser integrados em um sistema como esse todos os recursos que se encontrarem disponíveis (tais como estações de trabalho, PCs, supercomputadores, clusters de processamento, dispositivos móveis, dispositivos de armazenamento de dados, impressoras, sistemas computacionais, dentre vários outros), não importando quantos ou o que são, em um único sistema distribuído. Com isso, é possível obter um ambiente capaz de utilizar recursos ociosos tanto de processamento quanto de armazenamento ou mesmo executar cálculos complexos que poderiam requerer máquinas de um alto custo econômico, tal qual supercomputadores ou mainframes, pelo fato de possuir vários processadores ociosos distribuídos em uma área geograficamente ampla.

Este tipo de sistema de computação distribuída possui algumas características marcantes decorrentes por seu alto grau de distribuição. As grades computacionais são heterogêneas (por poder agregar múltiplos recursos computacionais, software e hardware de diferentes padrões e tecnologias), são escaláveis (por poder adicionar mais e mais recursos à grade, com a ressalva de que o maior desafio nesse caso é evitar que ocorra perda de desempenho e qualidade do serviço fornecido), é capaz de trabalhar conectando múltiplos domínios administrativos (com diferentes proprietários, políticas de acesso e uso congregando recursos de várias instituições), devem ser capazes de se adaptar a um ambiente susceptível a

falhas constantes por meio de um controle distribuído (a idéia nesse caso é não haver um ponto único de falhas para o gerenciamento de recursos) e compartilhar seus recursos de maneira segura (a grade não pode ser dedicada a uma única aplicação).

A computação em grade pode ser aplicada das mais variadas formas às necessidades das aplicações e geralmente as grades são categorizadas de acordo com o tipo de solução em que atuam. De uma maneira geral, podem-se classificar as grades em três categorias primárias [JACOB et al, 2006]:

- Grades computacionais – focadas no poder computacional e formadas em sua maioria por sistemas de alto desempenho;
- Grades de dados – responsáveis por gerenciar e prover o acesso a dados de múltiplas organizações, no qual o usuário não necessita saber a real localização daquela base de dados, bastando acessá-las através dos serviços de dados disponíveis;
- Grades re-aproveitáveis (*scavenging grid*) – geralmente utilizadas por computadores *desktop* e que consistem em grades capazes de reaproveitar os recursos ociosos de computadores que estão com um baixo nível de processamento.

Outro tipo de computação constantemente confundida como computação em grade é a computação aplicada a redes ponto-a-ponto. O fato é que não há um limite muito bem definido separando cada um dos tipos de grade, podendo elas co-existirem dentro de um mesmo contexto de aplicação, como ocorre no caso deste trabalho. A idéia aqui é a de um sistema de grade computacional interagindo com grades de dados de maneira que os recursos possam ser re-aproveitáveis ao longo da execução por parte das aplicações envolvidas.

Dessa forma, as grades computacionais podem ser vistas como uma plataforma para se integrar várias tecnologias e soluções em uma camada capaz de abstraí-las dos usuários finais,

organizada de forma a atender um ou mais domínios específicos de aplicação. Tal virtualização dos recursos compartilhados só se torna viável ao serem adotados padrões abertos, capazes de garantir que as aplicações ali envolvidas poderiam acessar de maneira transparente quaisquer recursos que se tornarem necessários.

Um ambiente como esse, capaz de compartilhar e tornar acessíveis tais recursos de maneira transparente certamente deve empregar a tecnologia adequada para tal virtualização, como também padrões e tecnologias nas áreas de escalonamento, gerenciamento de contas de usuário, gerenciamento de sistemas e segurança, dentre outras. Nesse sentido, um conjunto de padrões de serviços web e interfaces são o que tornam serviços (ou recursos computacionais) acessíveis.

Tal padronização de protocolos utilizados na grade computacional adiciona a esse tipo particular de sistema a faculdade de compartilhamento dinâmico e interoperabilidade de recursos com o usuário e sua aplicação, evitando desta forma a ocorrência de conflitos entre grupos de recursos que seriam incompatíveis entre si por questões de linguagem ou plataforma [FOSTER, KESSELMAN, TUECKE 2001; FOSTER et al, 2002].

Assim, se nos atermos às várias soluções computacionais que podem ser agregadas e transformadas em recursos da grade, gerenciados e organizados como serviços de maneira virtualizada, então as grades computacionais poderiam ser vistas como um grande sistema de computação distribuída que faz uso de recursos virtualizados. Dessa forma, a idéia das grades computacionais consiste justamente em se criar a ilusão em se tratar de um simples, porém grande e poderoso computador virtual composto por uma coleção de sistemas (heterogêneos ou não) conectados entre si e seus recursos.

É certo que tal sistema pode apresentar uma sobrecarga ao utilizar uma camada construída em software (conhecida como *middleware*) para gerenciar todos os recursos ou serviços que compõe tal arquitetura. Outro ponto relevante é o fato de tais recursos estarem

geograficamente distribuídos poderia gerar atrasos de comunicação consideráveis, além de se apresentar como um sistema bem mais complexo devido à sua heterogeneidade inerente, tornando bem mais difícil dessa forma a execução de aplicações paralelas ou distribuídas. Além disso, nem sempre o uso desta tecnologia se mostra apropriado ou vantajoso, variando muito de aplicação para aplicação e o contexto em que ela se insere.

Contudo, aplicando-se medidas estratégicas ou mesmo definindo-se políticas de acesso pertinentes a suas regras de negócio, além de um balanceamento de carga e armazenamento adequados, isso pode ser reduzido dramaticamente. Nesse caso, a replicação de recursos, delegação de tarefas em níveis por região ou *caching* de dados os mais utilizados exemplos de soluções cabíveis.

Tudo isso torna a grade uma tecnologia viável e ainda mais interessante no nível de negócios ou aplicações de uma maneira geral, por se poder “reciclar” recursos pré-existentes e agregar a eles poder computacional de maneira transparente e distribuída.

Portanto, ao longo deste capítulo vamos discutir sobre a evolução das grades computacionais até os dias atuais, a arquitetura das grades computacionais e os principais conceitos, componentes e funcionalidades da grade e o Globus Toolkit.

## 1.1 INTRODUÇÃO HISTÓRICA ÀS GRADES COMPUTACIONAIS

Desde seu surgimento, as grades computacionais foram sofrendo várias evoluções até os dias atuais. Tal evolução pode ser classificada em três gerações.

A primeira geração começou por volta do final da década de 80 e durou até meados dos anos 90 e tinha como objetivo a obtenção de grande poder computacional para aplicações que necessitavam do uso massivo de cálculos e simulações exigindo, portanto um ambiente de alto desempenho. Os principais projetos dessa geração foram o FAFNER e o I-WAY

[FAFNER, 2006; DEFANTI et al, 1996], ambos provendo ambientes de supercomputação através da integração de recursos.

O objetivo do projeto FAFNER [FAFNER, 2006] era testar os limites de segurança do algoritmo de criptografia RSA, provendo para isso um sistema heterogêneo capaz de ser executado em qualquer plataforma de software e hardware nos quais seus clientes seriam estações de trabalho, constituído por uma rede hierárquica de servidores web [ROURE et al, 2003].

O projeto I-WAY, por sua vez visava interconectar dezessete centros de supercomputação dos Estados Unidos através de um *backbone* de longa distância capaz de compartilhar e utilizar de maneira coordenada os recursos localizados em cada um desses centros de pesquisa. Com isso, várias tentativas foram feitas com o intuito de se criar soluções padronizadas tanto para aspectos de segurança, escalabilidade e heterogeneidade quanto com relação ao suporte e execução de aplicações paralelas que demandassem grande poder computacional.

A principal diferença entre os dois projetos consiste no fato do I-WAY trabalhar com aplicações de alto desempenho que necessitam de grande poder computacional e uma interconexão rápida e eficaz [FAFNER, 2006] ao passo que o FAFNER coordenava esforços distribuídos para executar a fatoração de grandes números de maneira paralela (e independentes entre si), mas não dependia, contudo, de uma conexão de alta velocidade para isso.

Ambos os projetos foram reconhecidamente inovadores e passaram a fazer parte da vanguarda da supercomputação, servindo como base para seus sucessores na grade computacional como o projeto Globus [FOSTER, KESSELMAN, 1998] e o projeto Legion [GRIMSHAW et al, 1997; ROURE et al, 2003], no caso do I-WAY e o SETI@Home no caso do FAFNER, respectivamente.

A segunda geração das grades computacionais começou em meados da década de 90 e durou até o início da década atual. Durante esta fase, vários projetos de pesquisa visavam suprir os desafios encontrados nesse tipo de ambiente, dentre eles, problemas relacionados à heterogeneidade e escalabilidade de um ambiente computacional como este, a gerência de recursos, sua utilização em múltiplos domínios administrativos e a tolerância à falhas.

Aqui a figura de um *middleware* surge como uma solução alternativa. O papel do *middleware* na segunda geração se tornou crucial por atuar como uma camada de software entre o usuário final e os recursos que compõem tal sistema, implementando diversos modelos de programação. Particularmente, o uso do *middleware* provê transparência sobre detalhes como sistema operacional envolvido em uso, capacidade da rede, linguagem de programação adotada na aplicação, localização geográfica do recurso em utilização, detalhes de arquitetura de hardware envolvidos tais como processadores, e memória disponível dentre outros.

Como comentado anteriormente, para se tornarem escaláveis tais aplicações deveriam ser adaptadas para suportar um alto grau de latência gerado pelas comunicações de longa distância ou diferenças quanto à qualidade da rede existente entre dois ou mais recursos. Portanto, estas aplicações deveriam ser capazes de explorar o conceito de localidade de recursos, políticas de replicação de fontes de dados, ou mesmo o *caching* de dados freqüentemente utilizados seriam alternativas nesse sentido.

Protocolos de autenticação e acesso a recursos foram então projetados nesta fase, por se apresentar necessária a figura de uma ambiente capaz de estabelecer políticas administrativas quanto a seu acesso e condições de uso, sendo esse mecanismo de segurança apenas um dos diversos aspectos envolvidos na construção dessa infra-estrutura computacional.

Conseqüentemente, para que este sistema se afirmasse como um ambiente coordenado, cooperativo e distribuído, muito esforço foi empregado para que alguns aspectos que se

tornaram necessários fossem definidos, tais como: um serviço de nomes, serviços de comunicação, serviços de informação, uma hierarquia administrativa, mecanismos de segurança e autenticação e ainda a elaboração de interfaces de acesso aos serviços, recursos e aplicações da grade de maneira fácil e intuitiva.

O **serviço de nomes** utiliza a mesma idéia do conhecido DNS (*Domain Name Service*) utilizado pela Internet, ou seja, os nomes utilizados pela grade computacional ou qualquer outro ambiente devem se referir a um objeto em particular existente em um determinado conjunto de maneira coerente (sua nomeação em todos os pontos do ambiente distribuído que o contém deve ser padronizada).

**Serviços de comunicação** devem ser capazes de prover mecanismos para a troca de informações, transferências de dados e requisições de recursos entre aplicações que fazem uso do ambiente de grade computacional capazes de trabalhar com diferentes formas de comunicação.

Por ser um sistema heterogêneo, a qualidade e disponibilidade dos recursos providos na grade podem variar constantemente, fazendo-se necessário, portanto, um mecanismo de informações sobre sua disponibilidade a cada instante que possa ser consultado pelo usuário quando necessário. Tais **serviços de informações** foram então projetados com o intuito de serem fáceis, precisos e confiáveis e constituem ainda hoje uma das características mais importantes das grades computacionais atuais.

Cada domínio administrativo possui suas próprias regras e políticas locais e não necessariamente uma grade impõe uma **hierarquia administrativa**. Uma política administrativa na grade computacional define regras de utilização comuns na grade computacional à medida que cada instituição passa a integrar aquela grade. Todo acesso e utilização devem então passar por mecanismos bem elaborados de **segurança e autorização** capazes de identificar o usuário que está fazendo tal requisição e verificar se o mesmo tem

autorização para acessá-lo. A infra-estrutura de segurança se apresenta como um mecanismo de extrema importância nesse tipo de ambiente distribuído e também um de seus pontos de maior complexidade, uma vez que a grade computacional deve ser capaz de administrar toda e qualquer transação ou ação do usuário de maneira que não cause danos ao material que ele compartilha, ao mesmo tempo em que buscam cobrir quaisquer falhas de segurança.

Normalmente cada usuário é registrado com um nome global no sistema, que é utilizado sempre que o mesmo pretende fazer uso de alguma funcionalidade existente. Para isso, contudo, tal usuário deve estar munido de uma chave privada e uma certificação gerada por uma autoridade certificadora, capaz de responder quanto à veracidade das informações relativas àquele usuário. A partir disso, sempre que ele fizer uso da grade, deverá ter uma credencial que deverá ser validada antes de qualquer ação ser aprovada para, somente assim, o acesso ser autorizado. Da mesma forma que o usuário, as máquinas que fazem parte daquela grade também requerem **autenticação e controle de acesso** similar para que possam acessar ou fazer parte daquele domínio administrativo, porque senão qualquer requisição vinda por parte dela será recusada pelo gerenciador de segurança da grade.

Por fim, a terceira geração consiste no estágio atual de evolução das grades computacionais e a primeira modificação mais marcante é a tendência cada vez maior de tornar sua estrutura praticamente orientada a serviços, o que conseqüentemente vem agregando grande importância à utilização de metadados. Tal combinação possibilita a criação de serviços flexíveis e extensíveis, uma vez que os metadados por si próprios descrevem dados que possibilitam os recursos descobrir automaticamente sobre sua disponibilidade e funcionalidades agregando-lhes assim maior autonomia e independência.

Em decorrência da visão de grades computacionais orientadas a serviço, várias comunidades de pesquisa começaram a ver a grade com grandes olhos, destacando-se entre elas as comunidades de arquiteturas orientadas a serviço, que apostam no sucesso da

tecnologia de serviços web, criada para resolver problemas de integração entre software e hardware. O aspecto mais interessante da utilização de serviços web é que com eles todo o processo de comunicação é capaz de abstrair a linguagem de programação, sistema operacional e arquitetura computacional existente.

Foi na terceira geração que se chegou a um padrão de arquitetura para grades computacionais. O chamado *Open Grid Services Architecture* (OGSA) [Foster et al, 2002] representa uma evolução no que diz respeito à utilização desses serviços, porque define comportamentos e conceitos chave que devem ser empregados para se definir uma grade computacional orientada a serviços.

Em decorrência desse novo conceito de grade computacional, os serviços web evoluíram para os chamados serviços grid, que são serviços web especializados para atender o contexto da grade computacional. A arquitetura OGSA e os serviços grid estarão sendo discutidos na próxima seção.

Neste cenário, uma transformação também vem ocorrendo no que diz respeito à estrutura dos *middlewares*. O Globus Toolkit [FOSTER, 2005] na sua versão 3.x (GT3) já apresentava suporte ao padrão OGSA, através da especificação OGSi (*Open Grid Services Infrastructure*) [TUECKE et al, 2003] para a construção de grades orientadas a serviço. A interoperabilidade derivada desses padrões é tal que, teoricamente, permitiria a construção de sistemas de grades de grades computacionais.

Com a evolução da tecnologia de serviços web, a especificação OGSi foi sendo substituída pela especificação WSRF (*Web Services Resource Framework*) [CZAJKOWSKI et al, 2004], o que causou a necessidade de uma revisão do *middleware* globus, surgindo assim o Globus Toolkit 4.x (GT4) [FOSTER, 2005]. A versão do Globus GT4 será discutida na seção 1.5 deste trabalho.

Assim, após esta revisão geral sobre a evolução e os avanços na área de grades computacionais, estaremos discutindo na próxima seção os principais aspectos da arquitetura OGSA, bem como os padrões de serviços adotados para a elaboração de grades computacionais orientadas a serviço.

## 1.2 CONCEITOS FUNDAMENTAIS DA GRADE COMPUTACIONAL

Como discutido na seção 1.1, um ambiente de grade computacional deve prover o compartilhamento de um conjunto de recursos em uma organização. Faz-se necessário observar aqui que uma organização formada por pessoas, instituições e seus recursos é conhecida como organização virtual. Organizações virtuais atendem em geral a um ou mais domínios de aplicações, como é o caso do projeto *Healthgrid*, por exemplo, formada por várias organizações virtuais cujas aplicações em grade se aplicam à saúde [HEALTHGRID Project, 2006] ou mesmo o caso do projeto *IntegraEPI* [SILVA, GAGLIARDI et al, 2007; REZENDE, GAGLIARDI et al, 2006; CAVICCHIOLI, GAGLIARDI et al, 2006], cujas aplicações visam também atender a saúde pública no que se refere ao monitoramento e previsão de epidemias por parte dos agentes de saúde como estará sendo mais bem discutido no capítulo 4 deste trabalho.

A idéia de compartilhamento de recursos torna-se um diferencial da grade computacional quanto à tradicional abordagem de sistemas distribuídos orientados a objeto, porque diferentemente destes sistemas, a grade é capaz de gerenciar e compartilhar recursos heterogêneos em áreas geograficamente amplas, além de ser também, em essência, um sistema distribuído.

Vale lembrar que as aplicações disponíveis a partir dos recursos compartilhados podem variar de uma simples transferência de arquivos a uma aplicação mais complexa, capaz de interagir com vários recursos a fim de solucionar um problema em particular.

Um recurso pode ser qualquer componente tal como: um cluster de computadores, a capacidade de uma rede, uma autoridade certificadora de segurança, um escalonador de aplicações, bancos de dados, arquivos, executáveis dentre uma infinidade de outros exemplos. Teoricamente, qualquer coisa na computação poderia ser vista como um recurso de uma grade se necessário para aquele domínio de aplicação que a grade deseja abordar.

Para tornar tudo isso possível, os serviços grid desempenham papel fundamental nesse cenário, porque é a partir deles que tais recursos tornam-se acessíveis às aplicações que devem se enquadrar nos padrões definidos pela arquitetura OGSA. Assim, estaremos a seguir discutindo melhor esse tipo de serviço e a arquitetura OGSA.

### 1.3 SERVIÇOS GRID

De uma maneira geral, um serviço grid é uma versão adaptada de um serviço web para o grid e consiste basicamente em um serviço web com estado. Tanto o recurso que oferece como seu estado é gerenciado e controlado através do serviço ao qual ele corresponde. Um mesmo serviço de grid pode gerenciar mais de um recurso, como acontece, por exemplo, com um serviço de acesso a dados que pode estar vinculado a mais de uma base de dados. Da mesma forma, é possível que um único recurso possa ser vinculado a mais de um serviço de grid.

Dependendo do negócio que se pretende modelar, um serviço pode precisar interagir com outros serviços grid disponíveis e criar uma instância daquele recurso cada vez que invocado. Tal instância pode permanecer viva por um tempo sendo descartada após isso

(quando se trata de um serviço transiente), ou então permanecer acessível por outros serviços por tempo indeterminado.

Como os recursos aos quais tais serviços são vinculados podem estar nos ambientes mais heterogêneos possíveis, um *framework* se mostra necessário para abstrair suas especificidades de implementação e uma arquitetura orientada a serviço (SOA) é capaz de prover esse *framework*. Esta arquitetura, além de tornar a integração entre tais recursos mais fácil, também facilita o compartilhamento de recursos entre organizações virtuais e heterogêneas, como é desejado na computação em grade.

Conseqüentemente, um *framework* padrão foi adotado pelo *Global Grid Fórum* (GGF) [GGF, 2007] baseado neste tipo de arquitetura baseada em serviços, conhecido como OGSA (*Open Grid Services Architecture*) [FOSTER et al, 2002] para a implementação das grades computacionais.

Nesta arquitetura, os serviços grid são a figura central e modelam todos os recursos da grade, oferecendo com isso capacidades derivadas do modelo de mensagens dos serviços web, tais como descrição de serviços e descoberta. Portanto, a seguir estaremos apresentando esta arquitetura que passou a definir a estrutura das grades computacionais modernas.

#### 1.4 OPEN GRID SERVICES ARCHITECTURE (OGSA)

A arquitetura OGSA [FOSTER et al, 2002], tal como ocorre com os serviços web, baseia-se na infra-estrutura que utiliza os padrões XML (*eXtensible Markup Language*), SOAP (*Single Object Access Protocol*), WSDL (*Web Services Description Language*) e WSIL (*Web Services Interface Language*). Contudo, como discutido nas seções anteriores, algumas características práticas e conceituais adicionais se fizeram necessário se tratando de um ambiente dinâmico como o discutido neste capítulo, devido à necessidade de se prover

mecanismos para a criação e descoberta de instâncias de serviços um ou mais serviços personalizados capazes de se recuperar de falhas de maneira segura e controlada.

Existem quatro conceitos importantes na arquitetura OGSA que merecem destaque:

- ***Naming***: num ambiente de grade computacional, cada instância de serviço grid deve possuir um nome que seja único, global e duradouro. Este nome deve ser obtido a partir do *Grid Service Handle* (GSH).
- ***Factories***: devem ser capazes de criar instâncias do serviço grid que se deseja invocar sob demanda e prover um conjunto de dados sobre os serviços que oferece para serem consultados. As *factories* devem ser serviços permanentes e estarem associadas a um registro da grade para serem descobertas pelas aplicações. Esta arquitetura define como registros os locais onde podem ser armazenadas várias informações pertinentes a cada recurso da grade.
- ***Instances***: criadas pela factory podem ser acessíveis diretamente por meio do registro da factory que responde ao serviço grid do qual se trata a instância, onde consta como uma *Grid Service Reference* (GSR). Uma GSR é codificada como um documento WSDL, no qual devem constar as informações mínimas sobre como tal recurso pode ser acessado.
- ***Stateful***: cada serviço grid possui um estado associado que pode ser checado de tempos em tempos conforme a necessidade por sua referência GSR.

Com isso, esta arquitetura é capaz de gerenciar o estado dos serviços grid, seu ciclo de vida e nomenclatura. Dentre os serviços providos pela OGSA [FOSTER et al, 2002], são fornecidas especificações para acesso e integração de dados, *workflow*, segurança, além de possibilitar a criação de novos serviços específicos para cada aplicação, baseados nesta definição.

A um usuário que quiser interagir com determinado serviço grid estará somente visível uma interface, descrita a partir de uma Linguagem de Definição de Interface (IDL – *Interface Description Language*). Tal abstração dos mecanismos internos que propiciam a execução de um serviço em particular é que leva à virtualização dos recursos da grade, porque isso possibilita que um mesmo serviço tenha várias implementações destas funcionalidades nas mais diversas linguagens de programação, desde que implementem um mesmo padrão.

## 1.5 O PROJETO GLOBUS

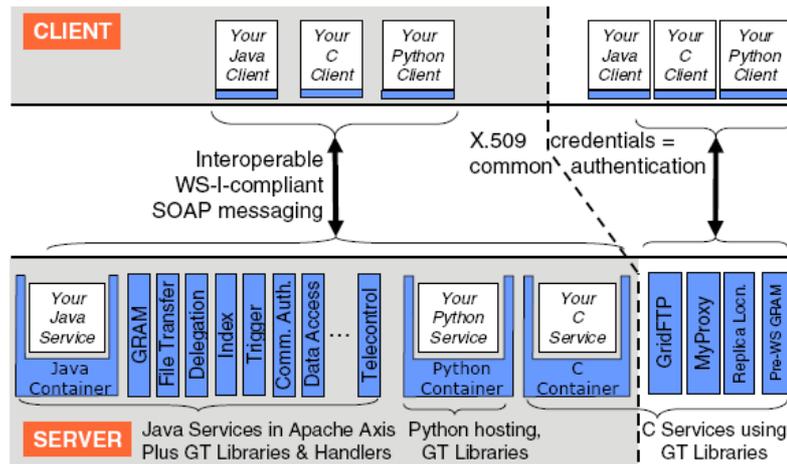
Criado por Ian Foster da Divisão de Matemática e Ciência da Computação do Laboratório de Argonne (EUA) e Karl Kesselman, do Instituto de Ciências da Informação da Universidade de Sul da Califórnia (EUA) [FOSTER, KESSELMAN, 1998], o projeto Globus tem como objetivo oferecer uma infra-estrutura para o compartilhamento de recursos heterogêneos e distribuídos. Este projeto atualmente disponibiliza um *toolkit* (um conjunto de ferramentas) para a configuração de uma grade computacional nos moldes da arquitetura OGSA [FOSTER et al, 2002] e se encontra na versão 4.1 (GT4).

O Globus Toolkit é um software de código aberto, extensível e sua implementação seu núcleo, o WS-CORE, se baseia fortemente na linguagem Java, sendo atualmente poucos recursos que ainda se mantêm na linguagem C desde sua versão 2.0. Devido a essas características, optou-se por utilizar este *toolkit* como substrato para a implementação dos serviços que compõem o sistema que estamos apresentando no capítulo 4 deste trabalho.

Por isso, a seguir estaremos apresentando os principais componentes da versão mais atual do Globus Toolkit, nos atendo aos aspectos mais pertinentes ao sistema IntegraEPI.

### 1.5.1 A arquitetura do Globus Toolkit 4.0

A arquitetura do Globus 4.x pode ser dividida em duas partes, conforme ilustrado na Figura 1.1: a parte do cliente e a parte do servidor.



**Figura 1.1** – Esquema da arquitetura do Globus Toolkit mostrando alguns de seus componentes.

[FOSTER, 2005].

Na parte do cliente residem as aplicações personalizadas que farão uso dos recursos dos serviços do Globus, podendo ser aplicações voltadas para a web ou aplicações *stand-alone*. A parte do servidor, por sua vez, abriga os principais componentes e serviços do Globus, e é composta por três *containers* (repositórios onde os serviços podem ser acessados): um para aplicações em Java, outro para aplicações em Python e um terceiro para aplicações em C. São nesses três *containers* onde os serviços criados especialmente para aplicações seguindo as regras de negócio planejadas estarão sendo disponibilizados aos usuários. É exatamente aí onde os serviços apresentados no capítulo 4 estarão disponíveis.

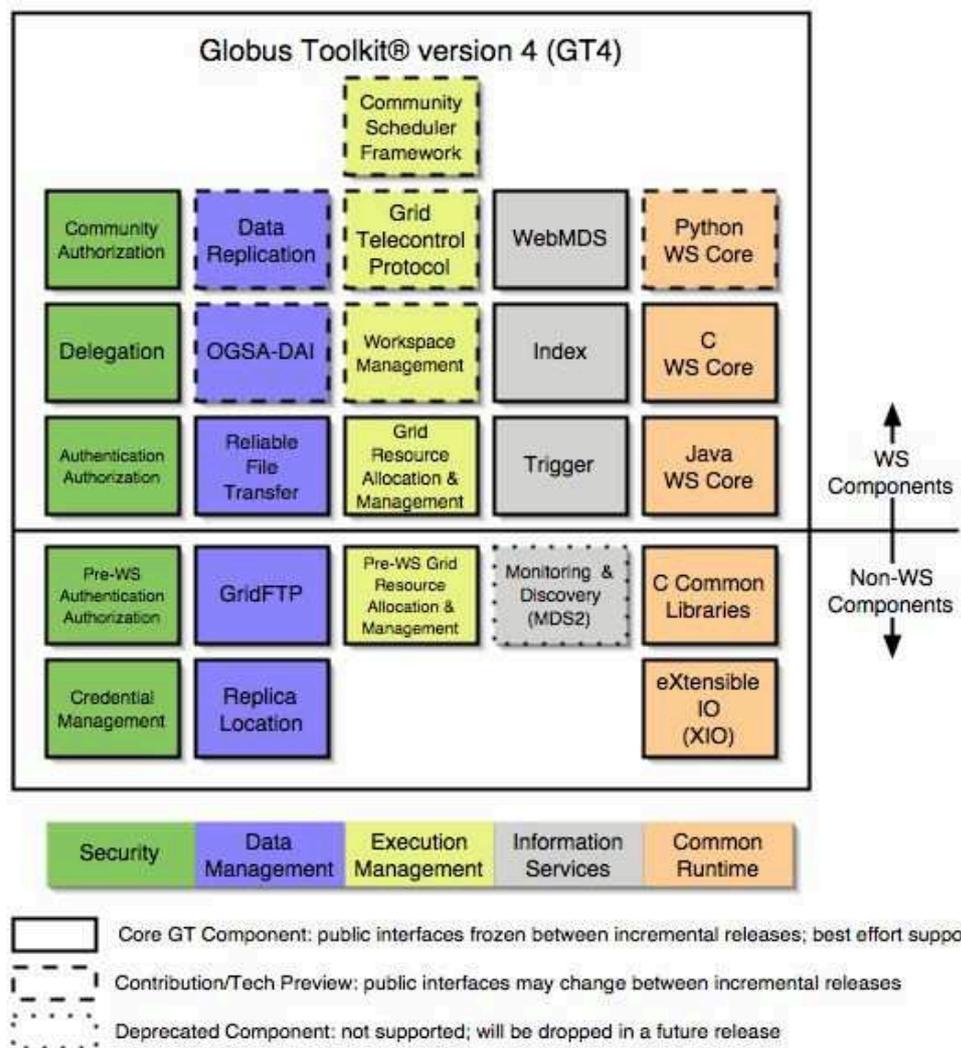
Assim, a parte do servidor é formada pelo conjunto de serviços úteis para o gerenciamento da infra-estrutura da grade. Estes serviços referem-se aos componentes de gerenciamento de execução (GRAM), transferência de e acesso a dados (GridFTP, RFT e OGSA-DAI), gerenciamento de réplicas (RLS e DRS), monitoramento e descoberta (Index,

Trigger e WebMDS), gerenciamento de credenciais (MyProxy, Delegation e SimpleCA), além de dar suporte à migração de aplicações das versões anteriores para a mais atual.

Todos os componentes supracitados serão apresentados em maiores detalhes na próxima seção.

### 1.5.2 Os componentes do Globus Toolkit 4

Como pôde ser visto na subseção anterior, o GlobusToolkit 4 é formado por uma coleção de componentes *open-source*.



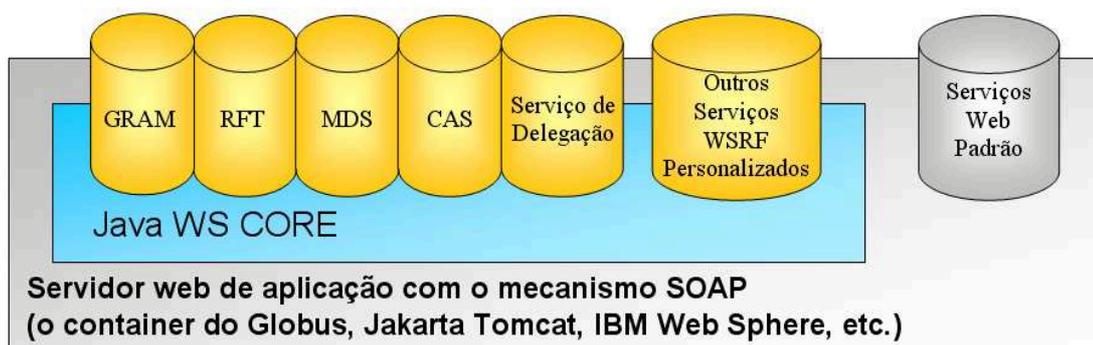
**Figura 1.2** – Os componentes do Globus Toolkit 4.0 organizados em categoria e tipo [GLOBUS, 2007].

É interessante observar mais uma vez que nem todos os componentes disponibilizados são implementações baseadas em serviços web, mantidos ainda na estrutura deste *toolkit* por motivos de compatibilidade. A Figura 1.2 ilustra a pilha de componentes oferecidos na sua versão 4.1.

O Globus Toolkit 4.0 (GT4) particularmente provê seus componentes classificando-os nas cinco categorias, a saber, (cada cor da Figura 1.2 representa uma dessas categorias): os componentes comuns para execução (*Common Runtime*, em laranja), os componentes de segurança (em verde), os componentes de gerenciamento de dados (em azul), os serviços de informação (em cinza) e os componentes de gerenciamento de execução (em amarelo). Cada um deles estará sendo discutido a seguir.

### 1.5.3 Os componentes comuns para execução

Em sua pilha de componentes, o GT4 inclui uma série de componentes comuns para execução. Tais componentes consistem em uma série de bibliotecas e ferramentas requeridas e utilizadas pelas implementações dos componentes que fazem parte do Globus Toolkit 4.



**Figura 1.3** – Esquema da relação entre o Java WS Core e serviços do GT4 [JACOB et al, 2006].

Tal como as demais categorias de componentes do Globus, os componentes comuns para execução são divididos em dois tipos: os que não são baseados em serviços web (compostos pelas bibliotecas de execução comuns da linguagem C e o componente eXtensible

IO, uma biblioteca para entrada e saída escrita exclusivamente para o Globus Toolkit em linguagem C) e os que são baseados em serviços web (compostos pelo Java WS Core, C WS Core e o Python WS Core).

O componente WS Core consiste nas APIs (*Application Programming Interfaces*) e ferramentas que implementam em linguagem Java a especificação *Web Services Resource Framework* (WSRF), atuando também como base para vários dos componentes disponíveis na arquitetura do Globus Toolkit 4.

É o Java WS Core quem provê também a base para a construção de serviços grid personalizados que são baseados no padrão *WSRF*. A Figura esquematiza essa relação entre os demais componentes e o em discussão nesta subseção.

Da mesma forma que o Java WS Core, os componentes C WS Core e Python WS Core consistem em uma série de ferramentas e APIs que implementam os padrões da especificação WSRF em linguagem C e Python, respectivamente, comportando-se de maneira similar a Java WS Core, contudo na linguagem correspondente.

#### **1.5.4 Os componentes de segurança**

Como a segurança é um dos componentes mais importantes nos ambientes de grades computacionais, o Globus Toolkit 4.0 disponibiliza vários componentes desta categoria como podemos ver em verde na pilha de protocolos da figura 1.2. De uma maneira geral, quando um recurso é compartilhado, devem-se definir políticas de acesso a eles para evitar acesso não autorizado.

Quanto a isso, a grade computacional provê a figura de um gerenciador de credenciais, capaz de dizer se aquele usuário é mesmo de quem ele diz ser, validando a certificação do usuário ou não conforme o caso, em um arquivo no padrão X.509 (Proxy) para que o acesso lhe seja dado. Para isso, o *Globus Toolkit 4.x* disponibiliza um serviço conhecido como

MyProxy, capaz de armazenar credenciais (como estas que são acessíveis apenas por meio de senhas configuradas previamente), e prover o acesso a essas credenciais remotamente.

De posse desta credencial, cada vez que o usuário solicitar o uso de um serviço ele será sempre validado de acordo com sua autenticidade. Tais certificações são obtidas por meio de um serviço de autenticação e autorização de usuários, capaz de criar certificados (por meio da figura de uma autoridade certificadora de confiança da organização virtual), a usuários e recursos da grade. Aí nesse ponto entra por vezes a figura do administrador da grade, que responde pelas requisições de certificação que as assinam e devolvem ao requerente como certificados.

Assim, o usuário terá uma identidade única e comum a todo o domínio da grade que fizer parte e realizará apenas um *logon* para todos os recursos (com prazo de validade determinado pelas credenciais que receber) [FOSTER, 2005]. Cada recurso da grade só poderá ser acessado desta forma se o usuário tiver direito a isso, conforme as políticas da grade que foram definidas. Nesse caso, toda vez que um usuário manifesta a necessidade de fazer uso de algum recurso, seus direitos de acesso são então delegados a tais serviços e sua permissão verificada para que sua utilização seja autorizada ou restringida.

### **1.5.5 Os serviços de informação**

Dado o alto grau de heterogeneidade presente num ambiente de grade computacional, serviços que provejam mecanismos para a descoberta de onde se encontra algum recurso ao mesmo tempo em que seu estado e tempo de vida pudessem ser monitorados é fundamental. Esse é justamente o papel desempenhado pelos serviços de informação.

Nesse sentido, o MDS (*Monitoring and Discovery Service*) é um componente do GT4 que simplifica a tarefa de monitoramento e descoberta de recursos computacionais a partir do

uso de metadados sobre os recursos que podem ser recuperados ou atualizados através de consultas e requisições.

Para isso, o MDS oferece um serviço de índices (Index service) exposto como um serviço WSRF capaz de coletar informações sobre vários recursos disponíveis e publicá-las para serem acessadas pelas demais aplicações cliente.

Além do serviço de índices, o MDS também provê um serviço gatilho (Trigger service), que executa uma aplicação pré-determinada quando a comparação entre uma série de condições descritas em um arquivo de configuração e os dados coletados for satisfeita.

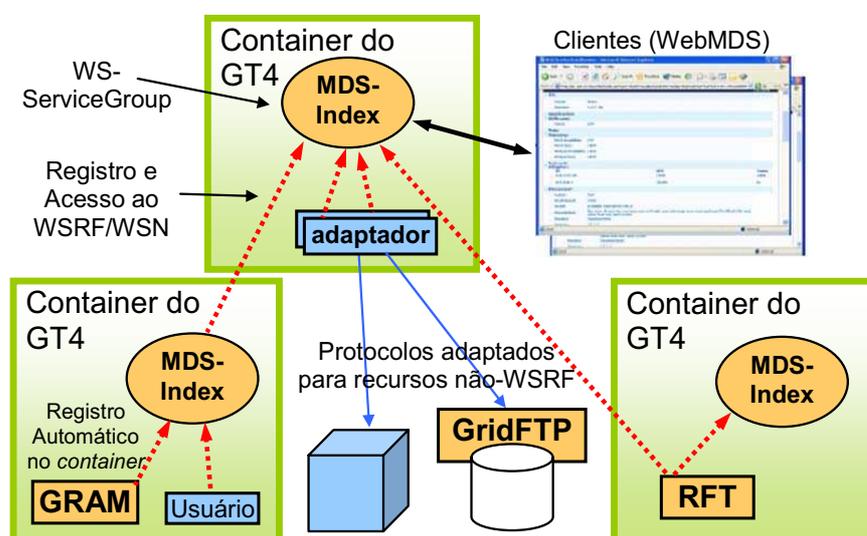


Figura 1.4 – Esquema de publicação de recursos no MDS [Foster, 2005].

O Globus Toolkit 4.0 possibilita uma interface de consulta dos recursos da grade computacional via web denominado WebMDS. No grid que instalamos e configuramos para o projeto IntegraEPI existe uma versão implementada do WebMDS em funcionamento acessível pelo endereço <http://serverpos.unisantos.edu.br:8080/webmds>.

A Figura 1.4 ilustra de maneira simplificada como o funcionamento do MDS. A idéia apresentada na figura mostra três *containers* do Globus Toolkit nos quais tanto os usuários quanto os recursos da grade são registrados ao serviço de índice que comentamos há pouco. É

interessante notar que mesmo os dois *containers* inferiores se registram também a um terceiro, que no caso da figura possui a interface WebMDS instalada.

### 1.5.6 Os serviços de acesso a dados

Prover acesso a dados de maneira distribuída consiste em uma das características mais úteis que a grade computacional pode oferecer, senão, das mais importantes, porque é através dela que uma aplicação é capaz de transferir um arquivo executável de maneira automática, ou mesmo coletar informações em bases de dados que se encontram distribuídas geograficamente (sem a necessidade referenciar manualmente o endereço de cada base de dados para sua aplicação).

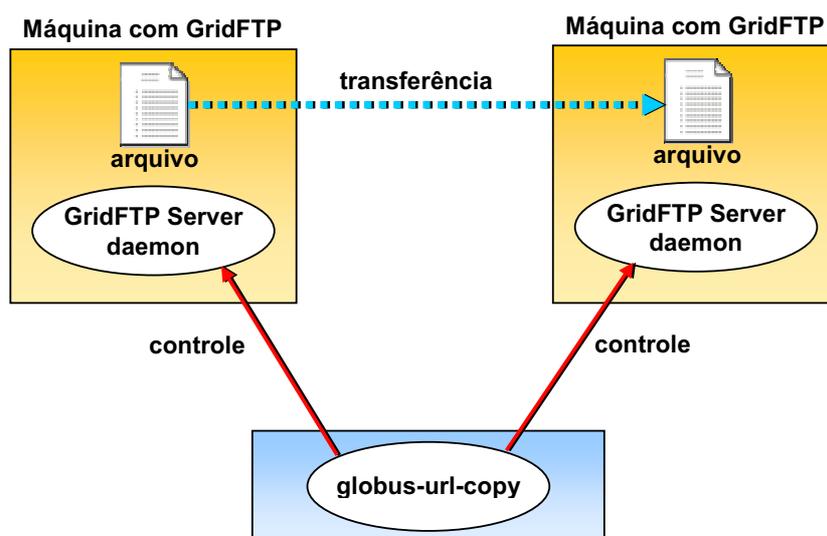
Particularmente, o Globus Toolkit disponibiliza para isso uma série de mecanismos de acesso a dados, tais como o mecanismo de transferência de arquivos como o GridFTP, o mecanismo de cópia confiável conhecido RFT (*Reliable File Transfer*) ou os chamados *grid data services* (GDS), providos a partir da solução de terceiros como é o caso do OGSA-DAI (*Open Grid Services Architecture – Data Access and Integration*) [OGSA-DAI, 2006] compatível ao Globus graças aos padrões definidos pela OGSA e WSRF.

Por isso, nesta seção estaremos detalhando melhor alguns dos componentes disponíveis nesta plataforma, porque estaremos nos referindo a eles mais adiante.

#### 1.5.6.1 O GridFTP

De uma maneira geral, o GridFTP é um serviço de FTP que possibilita a cópia segura e confiável de arquivos pela grade, com direito a autenticação dos usuários durante o processo.

Uma das características mais interessantes do GridFTP é a transferência ‘terceirizada’ (*third-party transfer*). Esse tipo de transferência é particularmente útil quando é desejado que um arquivo que se encontra remotamente em uma máquina seja diretamente transferido para outra máquina remota, ou seja, sem a necessidade de cópia à máquina do usuário para então passar à segunda máquina. Esse processo é ilustrado na Figura 1.5.



**Figura 1.5** – Uma transferência ‘terceirizada’ no GridFTP [JACOB et al, 2006].

O GridFTP não é um serviço baseado na tecnologia de serviços web, contudo é muito utilizado por vários outros serviços do Globus, incluindo entre eles o Reliable File Transfer (RFT) que estará sendo discutido a partir de agora.

#### 1.5.6.2 RFT: a transferência de arquivos confiável

O RFT é um serviço baseado na especificação WSRF e consiste basicamente num outro mecanismo de transferência de arquivos, porém mais sofisticado que o GridFTP. A Figura 1.6 apresenta um diagrama ilustrando a maneira como a RFT trabalha.

Em sua estrutura, a RFT utiliza-se de uma base de dados, em geral configurada no Sistema Gerenciador de Banco de Dados (SGBD) conhecido como PostgreSQL, onde são armazenadas informações sobre as transferências e seu estado.

Isso possibilita ao serviço a capacidade de recuperar o estado de uma transferência em caso de falhas ou caso tenha sido interrompida por outra razão, além de oferecer um mecanismo para apagar arquivos remotamente.

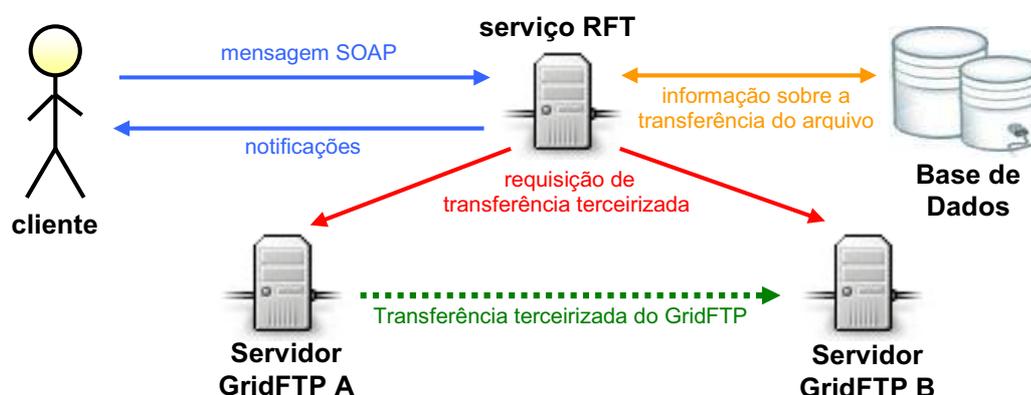


Figura 1.6 – Diagrama ilustrativo do funcionamento da RFT [JACOB et al, 2006].

Este tipo de transferência é muito importante por ser geralmente usado no processo de submissão de tarefas durante o processo conhecido como *staging* como será discutido mais adiante na discussão sobre o funcionamento do gerenciador de execução do Globus. Apesar da cópia de dados ser possível a partir de tais mecanismos apresentados até então, é de grande interesse que a grade consiga manter cópias de seus recursos como réplicas por questões de segurança ou mesmo desempenho. Nesse sentido o servidor de localização de réplicas RLS desempenha um papel de importância como componente do Globus Toolkit.

### 1.5.6.3 Replica Location Service (RLS)

Por serem sistemas distribuídos heterogêneos e poder apresentar uma distribuição de recursos geograficamente esparsa, as grades computacionais vêm a replicação como um

mecanismo de grande auxílio no que diz respeito ao acesso e obtenção de dados, porque algumas aplicações muitas vezes são obrigadas a trabalhar com acesso a terabytes de informações e transferir tal volume de dados toda vez que se fizer necessário. Isso pode, além de congestionar a rede, gerar atrasos enormes de execução e possivelmente até falhas durante a transferência.

Diante deste cenário, o RLS é capaz de mapear réplicas de dados pela rede e com isso a aplicação se torna capaz de explorar melhor o conceito de localidade dos recursos, evitando assim possíveis transferências de fontes de dados muito longínquas, o que poderia gerar grande latência e perda de desempenho do sistema. Além disso, a criação de réplicas torna maior a disponibilidade dos recursos, fazendo com que no caso de falhas ao acessar determinado recurso, o sistema consiga se recuperar mais prontamente simplesmente acessando uma das réplicas disponíveis para isso. Isso também torna a grade um ambiente com maior escalabilidade, porque novos recursos podem ser adicionados e replicados a cada instante explorando os conceitos discutidos até então. Deseja-se que tal replicação, contudo seja o mais transparente possível para a aplicação, ao mesmo tempo em que os dados replicados apresentem coerência entre si.

O RLS é constituído por dois componentes principais, a saber: o *Local Replica Catalogue* (LRC) e o *Replica Location Index* (RLI). O primeiro é um servidor no qual são mantidas as informações sobre as réplicas existentes e o segundo permite que os usuários possam executar consultas sobre determinado recurso entre vários servidores de réplica LRC.

Cada registro de um LRC é composto por um nome lógico (*Logical File Name* ou *LFN*) e um nome físico (*Physical File Name* ou *PFN*) para cada arquivo a ser replicado. Um LFN é basicamente uma string, capaz de identificar e representar um conjunto de arquivos iguais dispersos no grid. Cada LFN pode ter várias entradas de PFN, ou seja, um mesmo

nome lógico pode mapear mais de um arquivo de réplica. Da mesma forma, um mesmo nome físico PFN pode estar associado a mais de um nome lógico.

Os servidores LRC atualizam os seus dados constantemente a partir de mecanismos *soft state*, nos quais cada LRC envia informações de mapeamento para um ou mais RLIs, que após um tempo expiram. A vantagem dessa abordagem é que se um RLI falhar e por alguma razão voltar a funcionar, seu estado poderia ser reconstruído conforme as atualizações *soft state* fossem acontecendo, não sendo necessário, portanto, manter um estado persistente para ela.

#### 1.5.6.4 Data Replication Service (DRS)

O Data Replication Service é um serviço de replicação de arquivos no ambiente de grades computacionais utilizando GridFTP, RFT e o RLS para localizar e registrar as réplicas que cria. É construído como um serviço web compatível com a especificação WSRF.

#### 1.5.6.5 OGSA Data Access and Integration (OGSA-DAI)

O OGSA-DAI [OGSA-DAI, 2006] é um projeto patrocinado pelo e-Science [E-SCIENCE, 2006] e tem como participação a IBM e a Oracle, dentre outros que tem como objetivo a reutilização de bases de dados pré-existentes (relacionais ou até mesmo baseadas em XML) através da arquitetura OGSA.

O OGSA-DAI é constituído por um registro conhecido como *Grid Data Service Registry* (GDSR). O GDSR é um serviço permanente utilizado para localizar uma *Grid Data Service Factory* (GDSF). Uma GDSF é um serviço também permanente capaz de criar a instância de um *Grid Data Service* (GDS) quando requerido. O GDS é um serviço transiente que oferece ao cliente uma lista de operações possíveis de serem efetuadas sob o(s) recurso(s)

que nele se encontra(m) vinculado(s), tais como consultas, atualizações e até mesmo integração de dados, tudo isso a partir de um documento de entrada em XML.

A criação de tais serviços por parte do administrador da grade é um processo relativamente mais fácil, uma vez que o OGSA-DAI fornece um arquivo compatível com a tecnologia *ant* [ANT, 2006] em XML que oferece comandos e interfaces prontas para a criação de serviços, recursos de dados, e criação de vínculo entre um serviço e um recurso de dados.

Esta tecnologia é empregada no sistema IntegraEPI pela arquitetura GISE, proposta recentemente por Eduardo Gallo para a integração de dados [GALLO et al, 2006] e que estaremos discutindo no capítulo 4 em maiores detalhes.

### **1.5.7 Os componentes de execução**

O Globus Toolkit possui em sua pilha de componentes vários mecanismos muito pertinentes para a gerência de execução de recursos computacionais na grade. Dentre os componentes que formam esta categoria o *Grid Resource Allocation and Management* (GRAM) e sua versão em serviço web compatível com WSRF (WS GRAM) merecem destaque.

#### **1.5.7.1 WS GRAM**

O WS-GRAM é um serviço grid que provê a execução remota e gerenciamento do status de tarefas (ou *jobs*). Quando uma tarefa é submetida por um cliente, sua requisição é enviada a máquina remota através do protocolo SOAP, sendo então gerenciada pelo serviço WS-GRAM que ali se localiza. A partir de então o WS-GRAM é capaz de submeter a tarefa a

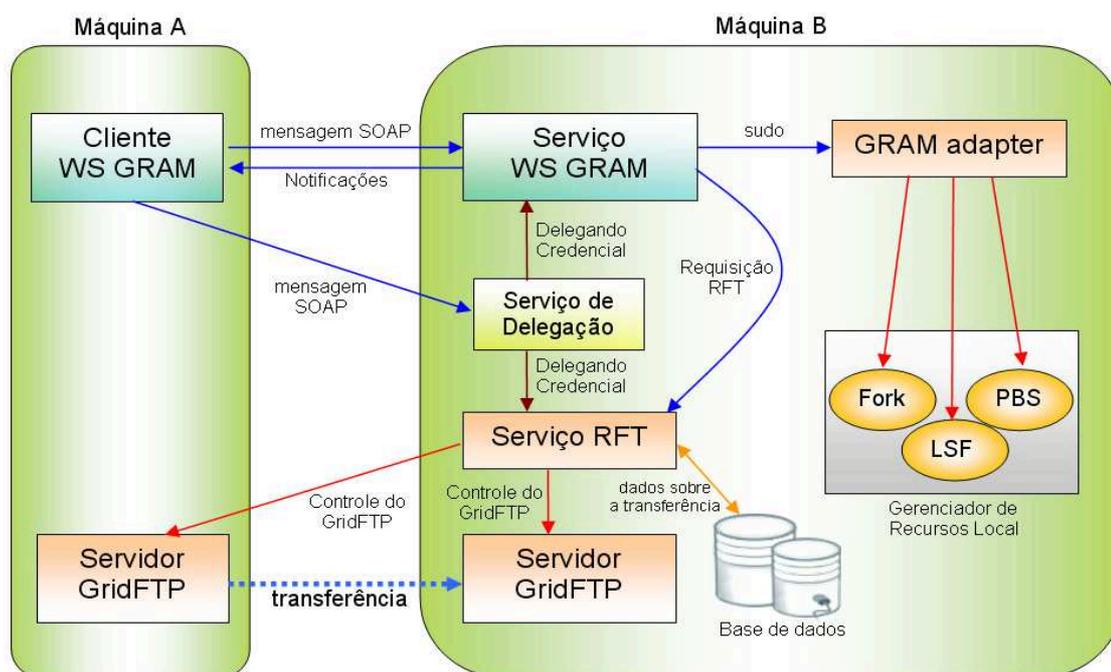
um escalonador de tarefas local como o Altair PBS, Condor ou a plataforma LSF e a partir de então é possível se obter o status da submissão da tarefa através do WSNotification.

Além disso, o GRAM tem como função processar scripts na linguagem RSL (contendo a definição das tarefas que se deseja executar) e também atualizar o serviço de informações do Globus sobre as capacidades e disponibilidades atuais do recurso que gerenciam.

Como estaremos discutido em detalhes no capítulo 4, o módulo de simulação de epidemias deste trabalho foi planejado para ser inicialmente executado como tarefas (ou jobs) independentes do tipo *bag-of-task*, a serem executadas em uma plataforma de grades computacionais. Nesta categoria cada tarefa não apresenta dependência com as demais, de maneira similar à discutida na primeira geração das grades computacionais quanto à aplicação que o FAFNER se propunha a solucionar. Por isso, estaremos a seguir discutindo os processos envolvidos na submissão de tarefas na grade.

Primeiramente, é válido lembrar que os sistemas de grades computacionais geralmente provêm vários mecanismos necessários para o monitoramento do estado de seus recursos. Tais ferramentas ou serviços de informação da grade computacional (tal qual o MDS4 ou Web-MDS, descritos anteriormente) provêm informações extremamente úteis à submissão de tarefas como, por exemplo, o nível de carga do sistema, quais recursos estão disponíveis ou em espera para a execução de mais tarefas, provendo para isso até mesmo interfaces gráficas (GUIs) ou comandos para a submissão de consultas, que são particularmente muito úteis na automatização de tarefas a serem executadas. Assim, um script poderia ser definido para primeiramente consultar o estado sistema, analisar a situação e decidir eventualmente se tal tarefa deve ser executado ou ainda permanecer em uma fila, por exemplo. O esquema de execução do WS-GRAM é apresentado na figura 1.7.

Na figura 1.7 é possível observar que o serviço WS-GRAM ao receber do cliente uma definição de tarefas a ser executado primeiramente é autenticado pelo serviço de delegação da grade e em seguida passa a solicitar a transferência dos arquivos envolvidos naquela execução via RFT, cuja transferência pode ser terceirizada (não necessariamente havendo necessidade de um servidor GridFTP na máquina do cliente para isso). Após o término da transferência, o serviço WS-GRAM envia aos escalonadores de fila locais a tarefa que é então executada.



**Figura 1.7-** A execução de uma tarefa pelo WS GRAM utilizando *staging* [JACOB et al, 2006].

De uma maneira geral, a submissão de tarefas utilizando o WS GRAM pode ser dividida em três etapas básicas: *stage in*, execução e *stage out*.

A primeira etapa consiste em enviar à entidade que executará a tarefa tanto os dados pertinentes àquela execução, consistindo potencialmente no executável ou mesmo em um script no qual são definidos passos pertinentes à execução do mesmo, bem como outros arquivos e dados envolvidos. Tal processo de envio do que se deseja executar para a máquina que fará esta execução é conhecido como *staging* (no sentido de transporte) dos dados de entrada (*stage in*).

Outra possibilidade é a de que os arquivos os programas requeridos já estejam previamente instalados nas máquinas da grade ou estejam acessíveis em um sistema de arquivos de rede (tal como o NFS).

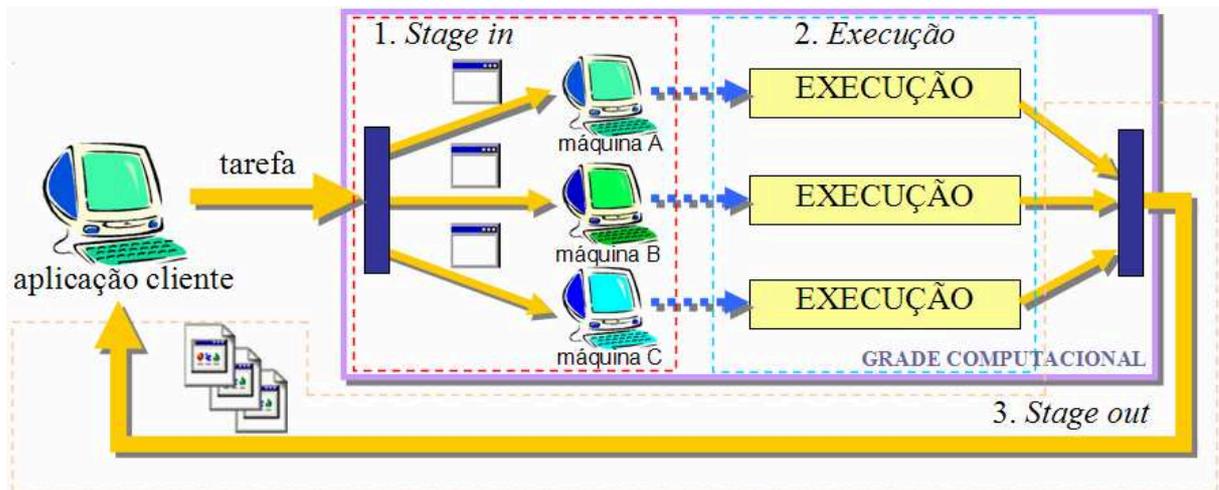
Como é previsto, um sistema de grades computacionais pode ser formado por um conjunto de máquinas heterogêneas, virtualizadas ou não (uma mesma máquina configurada para trabalhar como se fosse mais de uma, trabalhando como várias “máquinas virtuais” ou não), com os sistemas operacionais mais diversos, potencialmente incompatíveis entre si.

Por isso, é muito comum ser disponibilizada uma versão compilada para cada plataforma computacional em especial. Isso se torna mais interessante ainda no caso de alguns tipos de grade computacional que são capazes de registrar essas múltiplas versões do mesmo programa, uma vez que tal funcionalidade torna possível que o sistema de grade computacional consiga escolher automaticamente qual a versão do programa que se adapta de maneira mais adequada àquele ambiente de execução.

A segunda etapa consiste na execução da tarefa na grade pelo usuário local mapeado em cada máquina envolvida na execução a um identificador de usuário comum àquele tipo de grade computacional ou um nome comum de usuário daquela grade (*common name*, CN), como ocorre com o *globus toolkit*, no qual cada usuário é mapeado em um arquivo de mapeamento de usuários conhecido como *gridmap-file*, devendo haver restrições de acesso e escrita claramente definidas para tanto.

Na terceira e última etapa, os resultados são enviados de volta a quem submeteu tal tarefa e, dependendo da implementação, o usuário que submeteu tal tarefa seria capaz de ver os resultados intermediários ao longo da execução. Em alguns sistemas que não suportam esse processo conhecido como *stage out* (que basicamente é o inverso do processo de *stage in*), os resultados necessitam ser explicitamente enviados de volta ao usuário, podendo até mesmo ser utilizado um sistema de arquivos de rede para isso. A Figura 1.8 apresenta um diagrama

simplificado que resume para uma compreensão mais clara o processo de submissão de tarefas discutido até então.



**Figura 1.8** - Esquema de submissão de tarefas em grades computacionais detalhando as etapas de *staging*.

A submissão de uma série de tarefas pode ser definida através de scripts, nos quais são definidos os conjuntos de parâmetros aceitos pelo executável de acordo com o que se deseja investigar dentro daquele problema específico.

Isso é muito comum para aplicações do tipo *bag-of-task*, nas quais vários cenários podem ser obtidos apenas ajustando-se uma configuração de parâmetros diferente para um domínio de aplicação, como é o caso de uma simulação na qual se deseja descobrir os melhores locais para perfuração de poços de injeção e extração no caso do estudo de uma jazida de petróleo, por exemplo, ou mesmo no caso da aplicação que está sendo apresentada neste trabalho, na qual vários cenários distintos podem ser obtidos para o espalhamento da dengue, constituindo-se assim um mecanismo de grande valia para a tomada de decisões por parte dos agentes de saúde, uma vez que tal previsão seria possível ao configurar tais parâmetros do modelo.

Tais scripts podem muito bem ser feitos de maneira automática (criados como resultado da interação entre o usuário com uma interface gráfica qualquer) ou manual, na qual o usuário descreve em um arquivo as configurações de cada tarefa e submetidos a um

gerenciador de recursos da grade, de acordo com uma linguagem específica tal como a *Resource Specification Language* (RSL), comumente usada para descrever tarefas do *GRAM* (pré web services) pelas versões 2.0 e 3.0 do *Globus Toolkit* (GT2 e GT3) e a *WS-GRAM Job Description*, usado na versão mais recente do *Globus Toolkit 4.0* (GT4).

Quando o número de sub-tarefas é grande, um gargalo pode surgir à medida que os processos de *stage out* são executados, visto que tais resultados são geralmente centralizados na figura de um processo capaz de coletar os resultados obtidos e organizá-los de maneira adequada. No caso de muitas tarefas, uma coleta distribuída poderia ser uma alternativa para diminuir tal sobrecarga e tornar o processamento mais eficiente, ou seja, vários processos poderiam ser distribuídos na grade e cada um submeteria uma série de sub-tarefas, que seriam recolhidas e agregadas após seu término para então retornar tais resultados pré-organizados ao processo que o havia criado como em um escalonamento hierárquico.

Diante de todos os conceitos apresentados durante este capítulo, as grades computacionais se afirmam como a solução tecnológica escolhida para a construção do sistema de vigilância epidemiológica que estaremos apresentando nos próximos capítulos. Esta abordagem é muito interessante tanto pelo fator econômico quanto pela abstração possibilitada por este tipo de sistema, o que se enquadra perfeitamente com o perfil que se pretende oferecer nesta aplicação, porque nossa intenção é justamente disponibilizar ferramentas que possam auxiliar a tomada de decisões na área da saúde quanto ao combate de epidemias em um sistema que seja ao mesmo tempo robusto e poderoso, mas que seja barato e eficiente. Assim, no próximo capítulo estaremos prosseguindo para outro campo de estudo envolvido neste trabalho: o da modelagem de epidemias.

## **CAPÍTULO 2 – TÉCNICAS DE MODELAGEM DE PROCESSOS EPIDÊMICOS: UMA VISÃO GERAL**

Existe uma sofisticada literatura relacionada com a teoria de processos epidêmicos em uma população e a dinâmica de processos endêmicos e epidêmicos de uma doença infecto-contagiosa [KERMACK, MACKENDRICK, 1927; BAILEY, 1975; ANDERSON, MAY, 1991; BOCCARA, CHEONG, 1992; RHODES, ANDERSON 1996; CODEÇO, 2000; FERREIRA, YANG, 2003], mas não existe, entretanto, uma única maneira de se modelar o comportamento moléstia em particular. A resolução ideal para tal problema deve responder a vários questionamentos até que a partir de uma abordagem matemático-computacional um modelo seja elaborado visando expressar e prever o comportamento da epidemia em questão.

Desde seu surgimento, em meados do século XIX, a epidemiologia vem ampliando seu campo de atuação, focando-se cada vez mais no entendimento e combate às epidemias. De fato, ao se construir um modelo para a transmissão de epidemias deve-se levar em consideração vários aspectos físicos e sociais capazes de tornar sua dinâmica similar à observada na natureza. Nesse caso, uma das grandes dificuldades encontradas pelos teóricos dessa área ocorre ao se tentar mimetizar em um modelo as formas de interação que ocorrem entre os indivíduos de suas populações, tornando possível assim a transmissão de determinada epidemia entre eles [CLIFF, HAGGETT, SMALLMAN-RAYNOR 1993].

Para isso, o fenômeno saúde-doença só pode ser bem compreendido se considerada a complexidade e a dinâmica dos inter-relacionamentos entre diversos fatores operando ao mesmo tempo em múltiplas escalas espaço-temporais e organizacionais.

Em geral, os modelos podem ser classificados em vários eixos distintos, sendo que aqui estarão sendo distinguidos como modelos determinísticos ou estocásticos. Os primeiros,

em particular, são aqueles que descrevem a evolução espaço-temporal do fenômeno epidêmico empregando equações de diferença, diferenciais ou integro-diferenciais. Com isso, dadas as condições iniciais determinadas, as soluções são únicas para essa classe de modelos. Já os modelos estocásticos empregam probabilidades para a ocorrência dos eventos do modelo, cujas conclusões geralmente são consideradas em média sobre várias realizações.

Além disso, vários sistemas da natureza podem ser descritos por uma rede complexa de elementos, agrupados entre si por afinidade ou correlação espacial, como ocorre, por exemplo, com um círculo de amigos ou as cidades que formam um estado, respectivamente. Contudo, vários modelos epidêmicos baseados em redes são constituídos por indivíduos que interagem apenas com um determinado grupo limitado de elementos localmente, sendo baseados no indivíduo, ao passo que outros modelos matemáticos baseados em equações diferenciais (modelos compartimentais) baseiam-se na premissa de que tais indivíduos podem entrar em contato com qualquer outro que constitua uma população de maneira homogênea.

Assim, serão apresentadas e discutidas neste capítulo algumas formas mais simples de modelar fenômenos epidêmicos [ALVES, GAGLIARDI, 2006], sem pormenorizar os processos biológicos por detrás deles. Isso tornará possível a discussão mais refinada sobre conceitos básicos que serão tomados para aguçar a discussão ao apresentar o modelo epidêmico da dengue no capítulo 3.

## 2.1 MODELOS EPIDÊMICOS COMPARTIMENTAIS

A modelagem matemática de doenças infecto-contagiosas é antiga, podendo ser encontrada na literatura já em 1760 nos trabalhos de Daniel Bernoulli ao analisar a varíola [BAILEY, 1975]. De uma maneira geral, tal recurso apresentou-se nas últimas décadas como uma ferramenta importante para que se conseguisse entender melhor o comportamento de

epidemias e os efeitos causados com a imunização em populações (ao se adotar políticas de vacinação), por exemplo. Para isso, vários modelos descritos por equações diferenciais foram elaborados, no qual uma simples formulação compartimental dinâmica cuja estratégia é classificar a população em categorias (ou compartimentos) disjuntas de densidades de indivíduos é adotada [ANDERSON, MAY, 1991].

Dentro dessa abordagem determinística, o modelo SIR é um dos mais comuns na literatura, já que com ele é possível introduzir com simplicidade os principais mecanismos de um processo epidêmico, servindo assim como base para a formulação e desenvolvimento de modelos epidêmicos. De acordo com esse modelo, toda a população pode ser classificada nas seguintes categorias (ou compartimentos) disjuntas de densidades de indivíduos [BOCCARA, CHEONG, 1992]:

(S) a categoria dos *susceptíveis*, ou seja, dos indivíduos que não estão infectados, mas podem ser infectados;

(I) a categoria dos *infectados*, ou seja, daqueles indivíduos que são capazes de transmitir a doença aos susceptíveis;

(R) a categoria dos *recuperados* (ou *removidos/imunes*), ou seja, daqueles que já foram infectados pela doença e morreram (ou se isolaram) ou ficaram permanentemente imunes à doença.

Assim, em um instante de tempo  $t$  uma população é caracterizada pelas densidades  $S(t)$ ,  $I(t)$  e  $R(t)$  onde

$$N(t) = S(t) + I(t) + R(t) \tag{2.1.1}$$

é o número total de indivíduos da população a cada instante (tomando como constante no que segue). Partindo de tal lógica, basta adicionar mais estados (ou compartimentos) adotados para classificar os indivíduos da população em estudo para tornar mais realista a descrição de tal processo epidêmico.

É importante salientar aqui que tal classificação não significa em uma divisão física da população. Tal abordagem determinística particularmente não considera a existência do espaço entre os elementos da população, sendo esta uma dinâmica do tipo *campo-médio* [BOCCARA, CHEONG, 1992; KLECZKOWSKI, GRENFELL 1999], na qual cada indivíduo pode estar a qualquer instante em qualquer lugar. O que difere nesse tipo de modelagem são apenas os atributos de cada indivíduo. Isto quer dizer que os indivíduos desta população encontram-se *heterogeneamente misturados* (aleatoriamente misturados), permitindo que cada indivíduo da classe entre em contato com qualquer outro do sistema em qualquer instante.



**Figura 2.1** - O modelo SIR.

Com isso em mente, podemos entender os mecanismos do modelo através da transição entre esses estados que por sua vez define a dinâmica deste modelo (representado esquematicamente na figura 2.1), tomando três hipóteses básicas:

(1) O número de elementos infectados aumenta a uma taxa proporcional ao número de elementos infectantes e susceptíveis. Isto pode ser modelado por  $\beta S(t)I(t)$ , onde  $\beta$  é uma constante cuja finalidade é a de representar a taxa de infecção no sistema. Isto representa justamente a taxa em que o número de susceptíveis é reduzido em seu compartimento. De uma maneira geral, tal hipótese é herdada do estudo de cinética química ao considerar que uma reação química se dá a partir de um choque efetivo entre dois componentes e é conhecida como *lei de ação das massas*.

(2) A taxa em que indivíduos infectados são transferidos para a classe dos recuperados (removidos) é proporcional ao número de infectados. Isso pode ser modelado por  $\gamma I(t)$ , onde

$\gamma = 1/\tau$  também é uma constante, representando aqui a taxa de recuperação que no caso é justamente o inverso do tempo em que um indivíduo permanece infectado até se recuperar.

(3) Consideramos que neste modelo um indivíduo permanece por um tempo de incubação tão curto que ele é desprezível para a dinâmica geral que pretendemos modelar, fazendo com que os indivíduos transitem diretamente do estado susceptível para o estado infectado.

Tais mecanismos são descritos pelas seguintes equações diferenciais [MURRAY, 1993]:

$$\frac{dS}{dt} = -\beta S(t)I(t), \quad (2.1.2)$$

$$\frac{dI}{dt} = \beta S(t)I(t) - \gamma I(t), \quad (2.1.3)$$

$$\frac{dR}{dt} = \gamma I(t), \quad (2.1.4)$$

Combinando as duas primeiras equações temos:

$$I(t) = I(0) + (S(0) - S(t)) + \sigma \ln \frac{S(t)}{S(0)}, \quad (2.1.5)$$

onde é adotado  $\sigma = \gamma / \beta$  por simplicidade.

Aproveitando este mesmo modelo, vamos estabelecer quais condições são necessárias para que uma epidemia possa se estabelecer. Manipulando as equações 2.1.2 e 2.1.3, pode-se descrever o critério para que o número de infectados não cresça com o passar do tempo:

$$\frac{dI(t)}{dt} = \left( -1 + \sigma \frac{1}{S(t)} \right) \frac{dS(t)}{dt} < 0 \quad (2.1.6)$$

Tal relação se satisfaz quando  $S(t) < \sigma$ , pois em particular para  $S(0) < \sigma$  a epidemia não se estabelece. Para um melhor entendimento deste critério, estará sendo introduzido o conceito de número de reprodução básica  $R_0$  [ANDERSON, MAY, 1991]. Trata-se de um

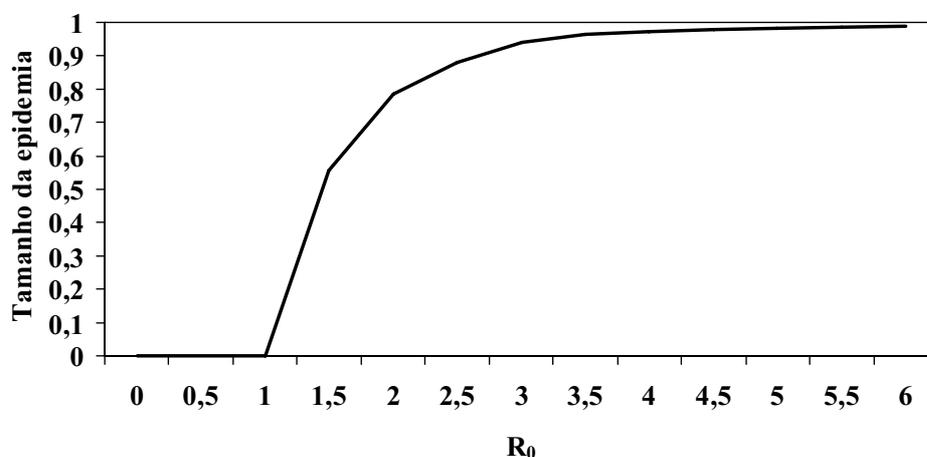
parâmetro adimensional que fornece o número de casos secundários gerados por um infectante enquanto ele está em seu período infeccioso. Em doenças de transmissão direta, este número é igual ao número médio de contatos adequados de um indivíduo durante seu período infeccioso. Por isso, para o modelo que estamos estudando aqui, podemos descrevê-lo da seguinte forma:

$$R_0 = \beta\tau = \frac{\beta}{\gamma} = \frac{1}{\sigma} \quad (2.1.7)$$

Trata-se de uma medida complexa e ambígua de fato, porém muito cara na modelagem de epidemias principalmente quando se parte do pressuposto compartimental, já que a mesma é capaz de resumir várias propriedades interessantes do sistema adotado. Denotando-se um valor crítico  $S_T$  que satisfaça esse critério e  $S$  como o número de susceptíveis da população, podemos escrever:

$$R_0 = \frac{S}{S_T} \Rightarrow R_0 > 1 \quad (2.1.8)$$

ou seja,  $R_0$  é dado nesse modelo pela razão entre o número de susceptíveis na população e a densidade limiar necessária para a persistência da doença. Esse resultado pode ser derivado de uma maneira mais geral e é conhecido como teorema do limiar epidêmico [KERMACK, MACKENDRICK, 1927]:



**Figura 2.2** - Fração infectada da população em uma epidemia em função do número de reprodução básica  $R_0$ .

O gráfico da Figura 2.2 ilustra as propriedades dessa grandeza, no qual podem ser observados os efeitos provocados à medida que o valor de  $R_0$  é aumentado. Além do mais, para o valor de transmissibilidade definido como  $R_0 = 1$  também fica clara que nos cenários em que  $R_0 < 1$  a epidemia não persiste, o que é de certa forma razoável visto que ela só poderia ser propagada a partir do momento em que um indivíduo infectado é capaz de infectar no mínimo um indivíduo da população enquanto em seu período infeccioso. Em outras palavras, se cada infectado não consegue infectar nenhum susceptível da rede durante seu período infeccioso, logo a doença não é transmitida.

Como comentado anteriormente, a partir deste modelo outros modelos puderam ser desenvolvidos com o intuito de estudar os mais variados tipos de doenças como o sarampo, dengue ou doenças venéreas, sendo necessário em alguns a inserção de outros compartimentos (ou estados), como é o caso do modelo SEIR compartimental, baseado no modelo SIR compartimental, por exemplo, no qual o período de incubação passa a ser considerado (na classe de (E) dos indivíduos expostos), caracterizado por armazenar a densidade de indivíduos que são infectados, porém ainda não infecciosos no sistema.

Outro modelo compartimental muito utilizado em epidemiologia é o modelo SIRS, que também se baseia nas categorias de indivíduos do modelo SIR compartimental, inserindo, contudo, uma modificação na transição dos elementos para possibilitar a volta dos recuperados para susceptíveis de maneira recorrente, como é o caso de doenças como a gripe, por exemplo, na qual os indivíduos podem voltar a ficar doente após um período de recuperação. Ao efetuar tal modificação no modelo SEIR obtém-se o modelo SEIRS, também muito utilizado.

No entanto, apesar das importantes contribuições desses modelos compartimentais para o desenvolvimento de teorias epidemiológicas, eles ainda apresentam suas limitações,

uma vez que, por mais que neles sejam agregadas informações, sempre serão regidos pelos dois seguintes pressupostos básicos [CODEÇO, 2000]:

- (1) Nos modelos compartimentais todos os indivíduos dentro de um compartimento são tidos como semelhantes entre si, isto é, para o modelo SIR, por exemplo, os elementos da categoria susceptível têm uma chance equivalente de passarem para a categoria dos infectados e desta para a categoria de recuperados. Sabe-se, entretanto, que há muitos fatores biológicos (idade, predisposição genética, estado emocional, dentre outros) e sociais (hábitos, costumes e círculo de amizades) que acabam influenciando para que um indivíduo adoça. Tais fatores podem ser, de certa forma, ainda inseridos na modelagem a partir da criação de sub-compartimentos menores que serviriam para classificá-los, contudo isso se demonstra inviável quando os fatores são muitos. Tal premissa é conhecida como homogeneidade dos indivíduos.
- (2) Nesse tipo de modelo todos os indivíduos dentro de um compartimento possuem a mesma chance de interagir com os de outro compartimento. Em outras palavras, todos têm uma mesma chance de interagir com os demais. Tal abordagem faz mais sentido se considerarmos uma população pequena e fechada, ao passo que o mesmo se torna bastante irreal se adotarmos uma população grande e com pouca mobilidade, visto que nesse último caso deveria haver maior probabilidade da transmissão ocorrer entre os vizinhos de cada indivíduo. Esta premissa é conhecida como homogeneidade de fluxo.

Com efeito, os modelos baseados em equações diferenciais, assumem que a população é totalmente misturada e, portanto desprezam fatores espaciais no espalhamento da epidemia. Conseqüentemente, esse tipo de modelagem não leva em consideração o fato de que a população é composta de indivíduos interagentes, ao considerar as populações como

entidades contínuas. Portanto, tal homogeneidade de fluxo produz resultados não realísticos para grandes populações.

Em adição, é interessante então observar que, de certa forma, tais pressupostos acabam limitando vários fatores importantes para o estudo de moléstias e que poderiam conduzir a uma nova visão sobre seu processo de espalhamento por considerar tudo de forma homogênea no sistema (interações e indivíduos), desconsiderando a individualidade de cada elemento. Portanto, para uma melhor previsão e definição de melhores estratégias no combate ao espalhamento de uma epidemia é necessário desenvolver novas metodologias matemáticas e computacionais.

De uma maneira geral esses modelos clássicos de epidemia, assim como os determinísticos e os estocásticos [BAILEY, 1975; GILLESPIE, 1976, 1977], falham em muitos aspectos para modelar adequadamente.

Particularmente, no caso estocástico é gerado um resultado em geral diferente do determinístico sendo que a curva epidêmica (ou, se preferir, o tamanho da epidemia) é em geral menor do que no análogo determinístico e os intervalos de tempo em que a epidemia ocorre também são diferentes nos dois casos [KEELING, GRENFELL 2000].

Como poderá ser observado em uma versão estocástica mais abrangente do modelo SIR, em alguns experimentos simplesmente a epidemia não se estabelece e então esses resultados se apresentam em média menor.

Assim, surgiu a necessidade de se pensar em uma outra forma de modelagem que se diferenciasse de tais características a fim de se compreender como esses pressupostos afetam a capacidade preditiva em tais modelos, o que levou à formulação dos modelos baseados no indivíduo (*individual-based models*). Nestes modelos considera-se a individualidade como o fator principal para que o espalhamento ocorra, sendo cada elemento diferente dos demais.

Em primeira instancia este modo de modelar atribui essas diferenças à localização de cada indivíduo no espaço.

Esse outro tipo de abordagem permite que o espalhamento agora atue no nível local ao invés de global, como era o caso dos modelos compartimentais, e assim, por meio de uma vizinhança pré-definida, faz com que o espalhamento seja efetivamente baseado no indivíduo.

No entanto, tal abordagem acaba geralmente considerando que todos os indivíduos das populações modeladas são estáticos, como se tivessem raízes ou vivessem presos em uma única posição no “mundo”, conhecendo apenas a um grupo de vizinhos que não sofre modificação com o passar do tempo, o que na natureza também é bastante irreal se aplicada em populações humanas ou de animais.

Surge daí a necessidade de se pensar em uma nova forma de modelagem, capaz de considerar a heterogeneidade espacial das relações populacionais a fim de possibilitar que elas atuassem localmente ao nível dos indivíduos sem desconsiderar a mobilidade na rede, de maneira global (como nos modelos do tipo campo-médio ou compartimentais, discutidos na seção anterior).

Tal modelo daria a possibilidade de todos os indivíduos poderem entrar em contato com os demais que a constituem sem importar sua posição, como ocorre quando uma pessoa viaja de um país para outro e acaba adquirindo uma doença (como uma transmissão em nível global) e, ao voltar ao seu país de origem, se torna alguém potencialmente capaz de retransmiti-la localmente.

Essa idéia é a base do modelo alternativo para o espalhamento de epidemias e o modelo da dengue em particular (que será o primeiro modelo disponibilizado no sistema de previsão de epidemias que estará sendo apresentado no capítulo 4), baseados em um tipo de rede de contato chamada autômatos celulares [ALVES, HASS, CALIRI, 2003], que será mais bem discutida e estudada ao longo deste trabalho.

## 2.2. MODELOS BASEADOS NO INDIVÍDUO

Vamos agora levar (ou tentar levar) em consideração no modelo SIR a heterogeneidade espacial das relações populacionais, ou seja, explicitar a dinâmica espacial do processo infeccioso, já que como vimos no SIR baseado na premissa compartimental não existe espaço.

Para fazer isso de uma maneira simples temos de alterar o pressuposto compartimental e formular modelos baseados no indivíduo, onde a unidade básica a ser modelada é o indivíduo e não mais uma população. Dessa maneira, agora devemos trabalhar com cada indivíduo explicitamente considerando sua individualidade mesmo que, em uma primeira análise, essa individualidade seja atribuída apenas pela distinção entre localizações espaciais, e que a transmissão entre eles ocorra de forma local e não mais global (como o pressuposto compartimental).

Esse tipo de abordagem pode ser melhor entendido com o uso de autômatos celulares [CODEÇO, 2000], nos quais cada ponto de uma grade é geralmente chamado de célula ou sítio, podendo conter no máximo um indivíduo. Estes podem ser classificados em categorias (ou estados) [OLIVEIRA 1999; CODEÇO, 2000], como susceptível, infectado e recuperado, se estiver sendo modelado o modelo SIR nesse autômato celular.

Introduzidos pelos matemáticos John von Neumann e Stanislaw Ulam em 1940, os autômatos celulares são uma ferramenta de grande auxílio para o estudo de sistemas que possuem fenômenos coletivos, como quebra de simetria, caos, ordenamento de fases e turbulência, tendo muitas aplicações na simulação de sistemas biológicos (processos de reprodução), fenômenos físicos (difusão) e sistemas sociais (formação de comunidades), dentre outras, por apresentarem a formação de padrões espaciais e temporais.

Os autômatos celulares podem ser definidos como sistemas dinâmicos discretos cujo comportamento é especificado em termos de relações (ou regras) locais. Formalmente, um autômato consiste de grafos [CHARTRAND 1985] com uma variável discreta em cada vértice. Essas variáveis por sua vez evoluem a passos de tempo discreto de acordo com um conjunto de regras envolvendo os valores das variáveis em cada vértice da vizinhança.

Assim, em um autômato celular é necessário que o espaço seja criado para que este possa conter os indivíduos que formarão a população que se pretende estudar e, para isso, usualmente utilizamos uma rede regular, na qual cada sítio (também chamado de célula ou vértice) assume um conjunto discreto de estados (caracterizando assim cada vértice um indivíduo da população). Dependendo do modelo que se pretende modelar utilizando esse tipo de rede, podemos ter regras determinísticas ou não determinísticas, que podem tornar o autômato como dito determinístico ou estocástico.

Nos autômatos celulares a dinâmica do modelo adotado (SIR, SIRS, SEIR ou SEIRS, dentre muitos outros) é definida através de regras de transição, nas quais são estabelecidos os comportamentos de cada estado específico e como eles influenciam as células vizinhas para que essas mudem de estado. Aqui, a escolha do número de vizinhos é importante, uma vez que determinará a dinâmica do espalhamento, nas quais um indivíduo susceptível pode se tornar infectado com uma probabilidade local

$$p_L = 1 - (1 - \lambda)^{n \text{ vizinhos infectados}}, \quad (2.2.12)$$

em que a probabilidade de um indivíduo ficar infectado é a probabilidade complementar deste não ficar infectado por cada vizinho que assim esteja, ou seja,

$$p_L = 1 - \underbrace{(1 - \lambda)(1 - \lambda) \dots (1 - \lambda)}_{n \text{ vizinhos infectados}},$$

onde  $\lambda \in [0,1]$ , representando a probabilidade de transmissão da doença. Assim, se adotarmos que  $\lambda = 1$ , ficamos com

$$1 - (1 - \lambda)^{n \text{ vizinhos infectados}} \Rightarrow 1 - 0^{n \text{ vizinhos infectados}} \Rightarrow 1 - 0 = 1,$$

no qual o espalhamento ocorrerá deterministicamente, por todos os vizinhos de um indivíduo infectado [HOLMES, 1997].

Existem vários outros mecanismos para que um espalhamento ocorra localmente, dentre os quais, podemos adotar um que, dado uma vizinhança de  $k_i$  indivíduos, e um número de  $k_{inf}$  vizinhos infectados, um indivíduo susceptível pode ficar infectado com uma probabilidade local

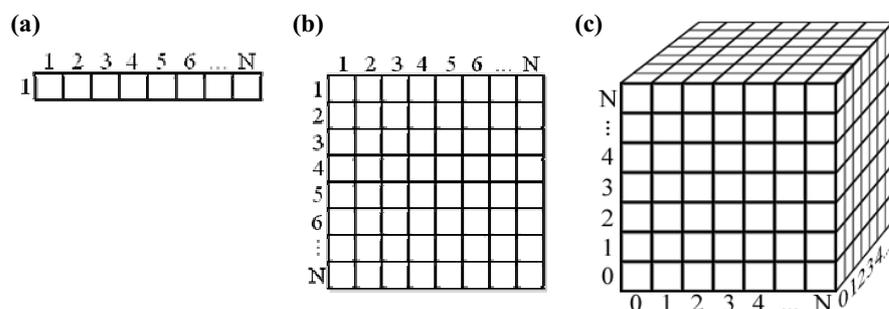
$$p_L = k_{inf}/k_i, \quad (2.2.13)$$

observando que desta forma, se o número de vizinhos infectados ao redor de um susceptível for igual a  $k_i$  este irá ficar infectado com probabilidade igual a 1.

### 2.2.1 Pressupostos e construção de um autômato celular

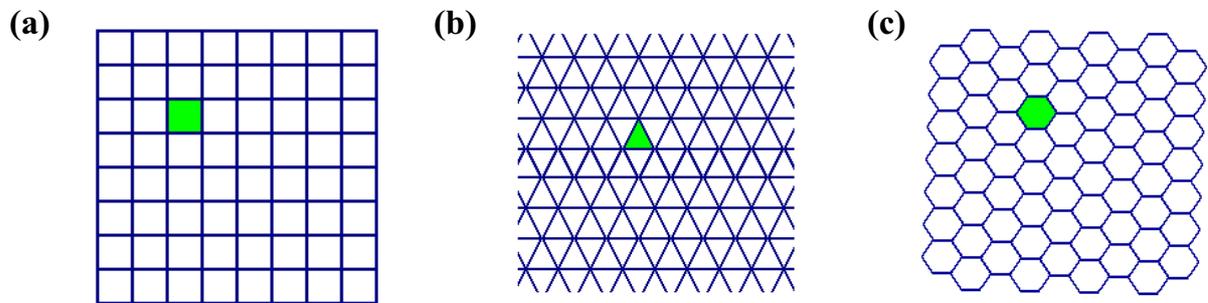
Para se construir um autômato celular é necessário se estabelecer quais são as regras de transição que deverão ser seguidas, uma vez que, uma pequena variação nestas regras pode alterar a solução do problema estudado. Tais regras dependem de uma série de escolhas que devemos fazer, que são:

1. *A dimensão da rede:* unidimensional, representada por uma rede  $1 \times N$ , onde  $N$  seria o número de células desta rede, bidimensional, com uma rede de  $N \times N$  células e tridimensional.



**Figura 2.3:** Desenho esquemático de (a) uma rede unidimensional, (b) uma rede bidimensional e (c) uma rede tridimensional. Cada célula receberá um atributo ou um estado característico.

2. *A geometria da rede*: deve-se especificar qual será a forma que a rede deve ter. Na figura a seguir estão sendo apresentados alguns exemplos utilizados de rede: quadrada, hexagonal ou triangular, dentre outras [EFROS, 1994].



**Figura 2.4:** Exemplos de redes com várias geometrias, onde (a) representa uma rede quadrada bidimensional, (b) representa uma rede triangular bidimensional e (c) uma rede hexagonal bidimensional, respectivamente. As células destacadas em verde representam uma célula desta rede.

3. *A vizinhança*: para qualquer célula de uma rede, há uma vizinhança que é formada por um conjunto de células que podem ou não afetar seu estado e que pode ser definida a partir da dimensão do autômato e a geometria de suas ligações. Os dois critérios mais freqüentemente utilizados para uma rede bidimensional são a vizinhança de Von Neumann e a vizinhança de Moore.

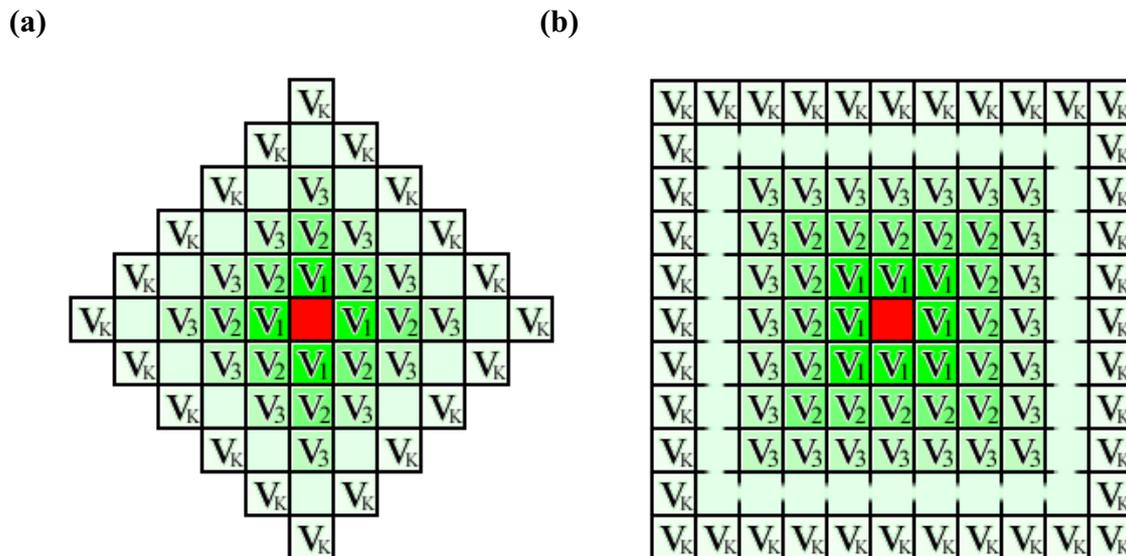
Na figura 2.5, estão ilustrados esses dois tipos de vizinhança que serão utilizadas em nossas simulações. Na figura 2.5a temos a representação esquemática da vizinhança de von Neumann, em que são consideradas como vizinhas às células que compartilhem as arestas de determinado indivíduo, o que no caso da rede quadrada, ilustrada na figura, são somados quatro vizinhos. A figura 2.5b mostra esquematicamente uma vizinhança de Moore para uma rede quadrada.

Observe-se que nesse caso foram acrescentados os vizinhos que têm vértices em comum com a célula que esta sendo considerada, elevando o número de vizinhos para oito.



**Figura 2.5:** Esquema dos tipos de vizinhança (em verde) que uma célula ou indivíduo (deixada em branco) em uma rede quadrada. (a) Vizinhança de von Neumann (b) Vizinhança de Moore.

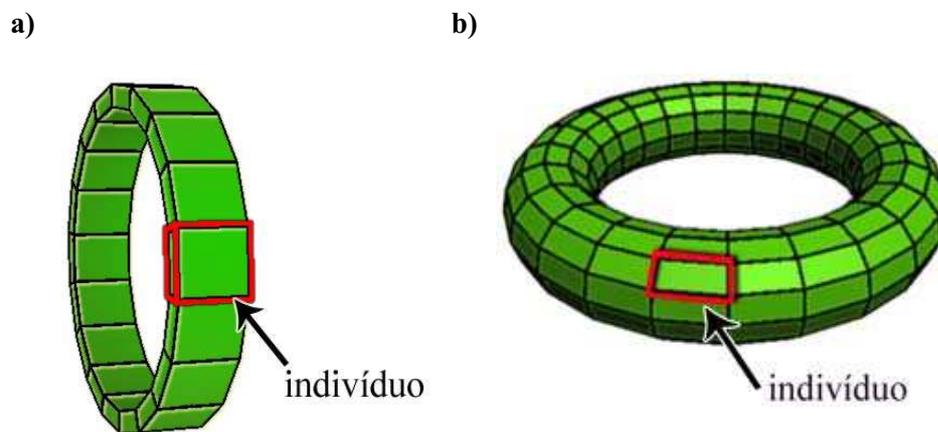
Além disso, esta vizinhança não necessariamente precisa ser aplicada aos elementos cuja ligação é exercida diretamente com um indivíduo específico, podendo ser estendida para  $K$  vizinhos mais próximos.



**Figura 2.6:** A vizinhança de von Neumann (a) e a vizinhança de Moore (b). Em ambas tem-se a representação de uma rede bidimensional configurada com uma vizinhança de  $K$  vizinhos mais próximos. Para cada representação, os quadrados marcados como  $V_i$ , onde  $i = \{0,1,2,3,\dots,K\}$  representam as células que constituem a vizinhança de cada célula da rede, tomada aqui separadamente em vermelho. Nesta representação foi adotada uma mudança gradual da cor das células que constituem a vizinhança, passando de verde para branco conforme o valor de  $K$  é acrescido, para mostrar que conforme isso ocorre, maior é a influência que cada célula pode exercer sobre sua vizinhança, no caso dela estar transmitindo uma doença.

4. *As condições de contorno:* como o tamanho da rede deve ser finito, uma célula quando se encontra na borda do autômato pode ter problemas *em saber* quem são seus

vizinhos e podem representar uma porcentagem muito grande da população quando esta for reduzida. Por exemplo, considerando uma rede de 10x10 células, 40% destes se encontram nas bordas. Assim, surge a necessidade de se estabelecer condições de contorno neste autômato, que podem ser periódicas, reflexivas ou fixas. As condições periódicas de contorno são muito utilizadas por proporcionarem uma situação equivalente a de simulações com redes infinitas. Na figura 2.7 é apresentado um desenho esquemático da aplicação das condições de contorno em uma rede unidimensional (figura 2.7a), gerando uma fita circular e em uma rede bidimensional (figura 2.7b) uma forma toroidal (um tórus).



**Figura 2.7:** Condições periódicas de contorno em uma rede unidimensional e em uma rede bidimensional, respectivamente.

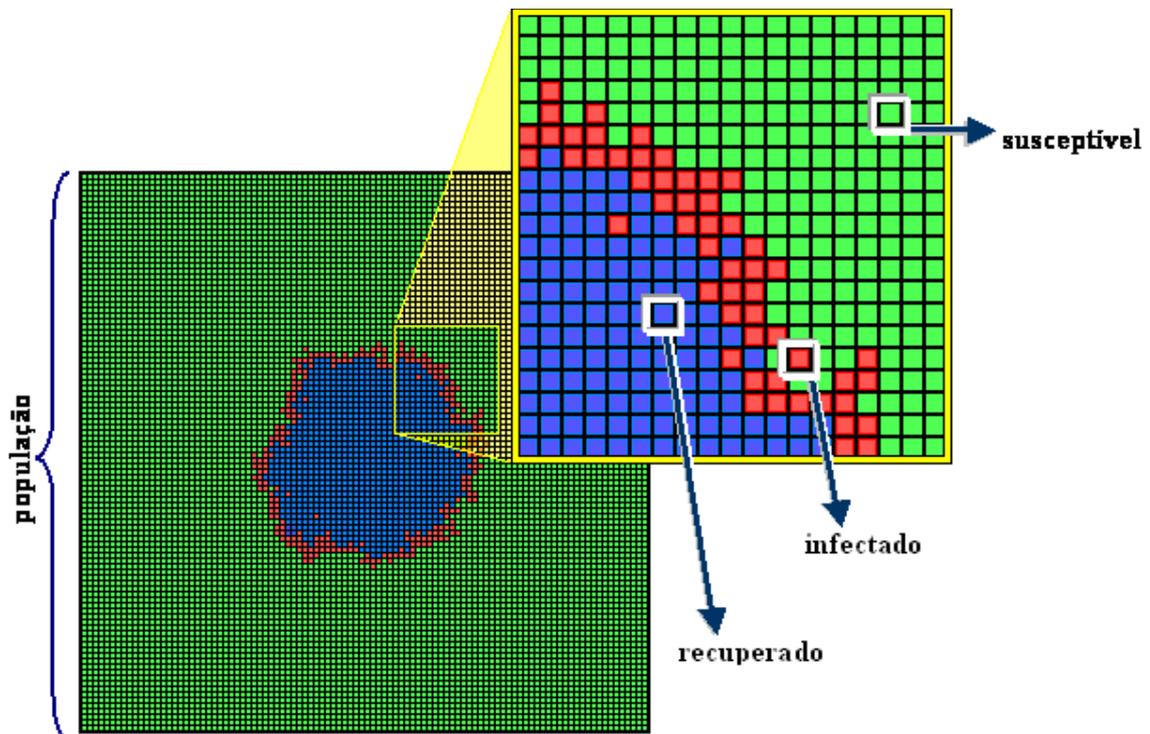
Em ambos os casos, devem-se destacar cada indivíduo na rede e observar que, ao aplicar as condições de contorno periódicas, as vizinhanças nas fronteiras da rede foram colocadas em contato.

### 2.2.1.1 Os estados do autômato celular e as regras de transição

Como comentado anteriormente o número de estados que uma determinada célula pode assumir é descrito por um número discreto e finito de estados, cada qual sendo definido por um número ou propriedade, conforme necessário. As regras de transição no autômato são

as regras que fazem com que um estado possa mudar em determinada célula que representa um indivíduo da população.

Essas transições dependem do estado atual desta célula e também dos estados em que seus vizinhos se encontram. Neste tipo de rede devem-se estabelecer regras que podem ser: determinísticas, quando há a dependência quanto à configuração do sistema (a partir da vizinhança); totalísticas, quando estas regras não dependem da configuração destas células vizinhas (dependem somente do número total de indivíduos em um estado específico) e regras probabilísticas, na qual não existe uma configuração específica para cada configuração da vizinhança, mas possibilita vários resultados associados a uma probabilidade.



**Figura 2.8:** Esquema de uma rede quadrada bidimensional, destacando os três diferentes estados para o modelo SIR, tomado aqui como exemplo.

Para uma abordagem do tipo SIR, baseada em autômatos celulares com transições probabilísticas, por exemplo, seriam utilizados três estados: suscetível (o indivíduo pode ficar infectado), infectado (o transmissor da doença na população, ou agente infeccioso) e recuperado (ou imune), de maneira que:

$$\sum S + \sum I + \sum R = N, \quad (2.3.14)$$

em que  $N$  representa o número de células (ou indivíduos) da rede, mantendo-se constante de maneira similar à vista na equação 2.1.1. Na figura 2.8 foram destacados tais estados enumerados para o modelo SIR utilizando autômatos celulares.

Desta forma, pensando em um modelo composto por  $M$  estados ( $S_1, S_2, S_3, \dots, S_M$ ),

$$\sum_{i=1}^M S_i = N, \quad (2.3.15)$$

deve ser verdade para  $M > 0$  e  $N > 0$ . É interessante notar que a onda em vermelho apresentada na figura (de elementos infectados) apresenta tal aspecto irregular devido a abordagem probabilística adotada no exemplo. No caso, se fosse adotado uma abordagem determinística, tais ondas apresentariam uma forma mais bem definida, de acordo com a vizinhança escolhida.

## **2.2.2 O modelo probabilístico alternativo baseado em autômatos celulares básico para a disseminação de uma epidemia**

Até então, nas seções anteriores foram discutidos aspectos relacionados à modelagem de epidemias, como os pressupostos necessários neste trabalho para a modelagem de epidemias utilizando autômatos celulares. Com isso em mente, pode-se enfim iniciar a discussão sobre o modelo alternativo para o espalhamento de epidemias desenvolvido a partir de autômatos celulares [ALVES, HASS, CALIRI, 2003], modeladas aqui em termos de contatos locais e globais entre os indivíduos, que nos baseamos para a estruturação do modelo da transmissão da dengue que estará sendo apresentado em grandes detalhes no capítulo 3 e que será o primeiro dos modelos epidêmicos que o sistema de previsão de epidemias que apresentamos no capítulo 4 disponibilizará.

Diante deste cenário, na próxima subseção serão apresentados características e mecanismos desse modelo alternativo de autômatos celulares para epidemias e, posteriormente, os resultados por ele obtidos.

### 2.2.2.1 Características do modelo alternativo baseado em autômatos celulares para uma população apenas

Como foi comentado ao longo deste trabalho, tomou-se como base para o desenvolvimento deste modelo alternativo (para uma população apenas) um autômato celular probabilístico tradicional (bidimensional), utilizando eventualmente uma rede quadrada, na qual cada célula representa um único indivíduo que possui uma posição  $(i,j)$ . Como este trabalho tem enfoque na dinâmica da disseminação de epidemias, foi adotado um conjunto de estados, que representam o *status* de saúde-doença de cada indivíduo em um determinado instante.

Contudo, a principal característica desse modelo, com relação aos autômatos celulares tradicionalmente utilizados na epidemiologia, é a utilização de dois tipos de interação entre seus indivíduos: as interações locais, que ocorrem devido à influência exercida pelos vizinhos infectados sobre os indivíduos susceptíveis do sistema, e as interações globais, nas quais todos os indivíduos têm igual probabilidade de entrar em contato entre si e, desta forma, possibilitar que tal disseminação ocorra por outras regiões da rede, distantes da onda local de infectados.

Dessa maneira, a probabilidade de qualquer susceptível tornar-se infectado é modelada como a superposição desses dois tipos de interações por meio da equação:

$$p_S = \Gamma p_G + \Lambda p_L \quad (2.3.16)$$

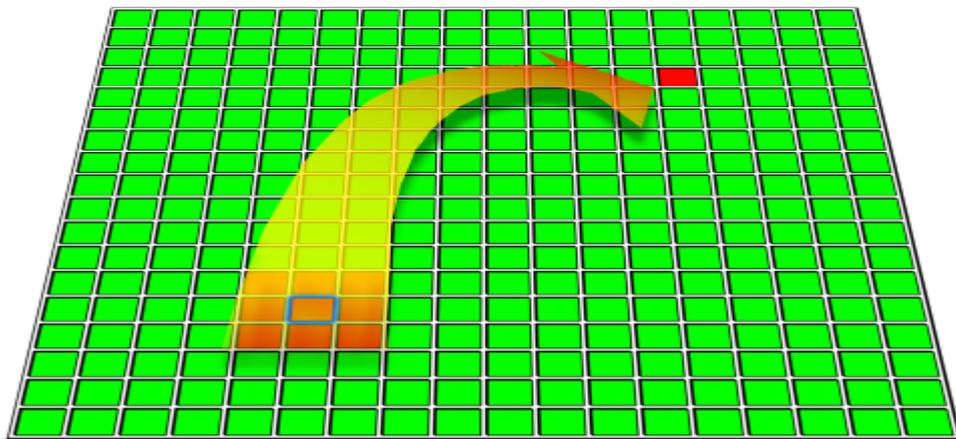
onde  $0 \leq p_S \leq 1$  e  $\Gamma$  e  $\Lambda$  são parâmetros utilizados para se ajustar as interações de curta (formação de *clusters*) e longa distância (do tipo campo-médio) [ANDERSON, MAY, 1991] entre indivíduos de uma mesma população, tal que  $\Gamma + \Lambda = 1$ .

A influência global é modelada pelo número total de indivíduos infectados que estão na rede, isto é, devido à presença e mobilidade de qualquer indivíduo infectado na população, descrita pela equação:

$$p_G = \frac{\rho}{N} \sum_{\{k,l\}} \delta_{i,\sigma(k,l)} \quad (2.3.17)$$

onde,  $0 \leq \rho \leq 1$  é um parâmetro do modelo que limita o valor máximo de  $p_G$  e está relacionado com a mobilidade intrínseca da população. Aqui, a notação  $\delta_{i,\sigma(k,l)}$  representa a célula (ou sítio) na posição  $(k,l)$  da rede e que se encontra no estado infectado. Assim, o que a somatória da equação 2.3.17 faz é simplesmente visitar cada um dos elementos que compõem aquela população, linha por linha e coluna por coluna, contando apenas aqueles que estão no estado infectado.

A influência local, por sua vez, pode ser calculada a partir da vizinhança de um determinado indivíduo, que pode conter um ou mais infectados. Caso isto seja verdade, a probabilidade de transmissão da doença por indivíduo infectado pode ser calculada da mesma forma como ocorre na equação 2.2.12 (o que não impede, contudo, que ao invés dela, tal probabilidade seja calculada pela equação 2.2.13), apresentada na seção 2.2.



**Figura 2.9:** O espalhamento global e local em um autômato celular bidimensional alternativo. A seta em destaque representa a forma como um indivíduo pode ser infectado globalmente, sendo que os quadrados em vermelho representam os infectados e verdes os susceptíveis. Pode-se notar aqui também o espalhamento local se manifestando em uma vizinhança de Moore.

Nesse modelo, portanto, a dinâmica na população de indivíduos é governada por regras de interação ou contato dadas por:

1. Qualquer indivíduo susceptível pode tornar-se infectado com uma probabilidade  $p_S$  (definida pela equação 2.3.16).
2. Indivíduos infectados tornam-se imunes depois de um período  $\tau$ , deterministicamente.
3. Uma vez na classe de indivíduos recuperados eles não participam do espalhamento da doença por um período de imunidade maior que o processo completo da epidemia.

Vale lembrar que a principal diferença entre o modelo clássico compartimental e o autômato celular (baseado no indivíduo) se dá pelo fato de não existir espaço no primeiro, podendo todos os indivíduos interagir com todos os demais, ao contrário do que ocorre com o SIR, modelado em um autômato celular, em que cada indivíduo possui uma localização definida no espaço, sendo seu processo de transmissão localizado no espaço.

Assim, a partir de tais equações é estabelecido um *link* entre esses dois cenários, podendo o sistema, a partir dos parâmetros de mobilidade  $\Gamma$  e  $\Lambda$ , comportar-se como um modelo baseado no indivíduo tradicional (para  $\Gamma = 0$ ), um modelo no qual efetivamente não há espaço (para  $\Gamma = 1$ ) e um cenário interessante, no qual ocorrem simultaneamente as duas abordagens (para  $0 < \Gamma < 1$ ), o que possibilita atribuir ao sistema um comportamento similar ao possibilitado pelo protocolo de re-conexão do modelo de mundo pequeno [WATTS, STROGATZ, 1998; ABRAMSON, KUPERMAN 2001; WATTS, STROGATZ 2004]. Contudo, vale lembrar que no modelo de mundo pequeno este processo de re-conexão ocorre apenas uma única vez logo no início de cada simulação, o que faz com que os atalhos

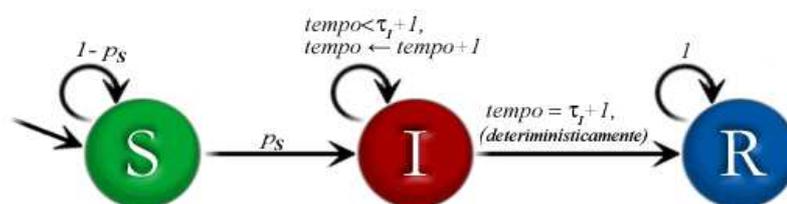
estabelecidos, mesmo entre indivíduos distantes entre si, permaneçam estáticos até o fim da simulação, o que já na abordagem probabilística apresentada nesta seção ocorre a cada instante. De fato, na vida real uma pessoa pode fazer novas amizades ou inimidades diariamente e, portanto, a abordagem baseada em autômatos celulares considerando estes dois tipos de contatos (local e global) se mostra mais coerente.

Diante do cenário, estará sendo apresentado a seguir um estudo mais detalhado sobre as características relacionadas a esse tipo de abordagem quando aplicadas ao modelo SIR discreto.

#### 2.2.2.2 O modelo SIR na abordagem baseada em autômatos celulares

Ao apresentar na seção 2.2.1 os pressupostos para o desenvolvimento deste autômato celular alternativo, foram comentadas e definidas algumas regras de interação para representar o modelo SIR (Susceptível  $\rightarrow$  Infectado  $\rightarrow$  Recuperado) que, como discutido na seção 2.1, é um modelo muito utilizado para se estudar vários tipos de moléstias.

As regras de transição adotadas aqui para cada indivíduo em particular podem ser esquematizadas pelo seguinte autômato finito:



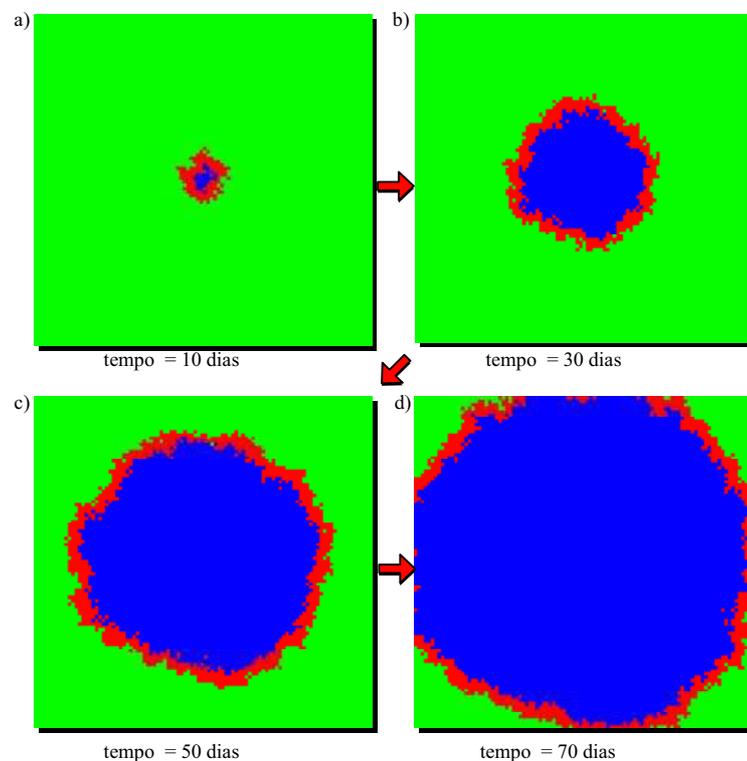
**Figura 2.10:** As transições do modelo SIR discreto modeladas pelo autômato celular com uma componente global e uma local. Portanto, o sistema é governado por  $p_s$  e  $\tau$ , e sua evolução temporal é determinada pela atualização da rede em cada passo de tempo (de maneira discreta).

Nas simulações do autômato celular alternativo proposto neste trabalho (ver seção 2.2.2) [ALVES, GAGLIARDI, 2006], mostradas a seguir, foram utilizadas redes

bidimensionais de 100.000 ( $N = 100^2$ ) indivíduos, começando com um único infectado no centro da rede.

Para ilustrar a diferença entre os efeitos causados pela forma normalmente utilizada de interações do autômato celular (apenas locais) e a superposição dos dois tipos de contatos (locais e globais), foram efetuados os experimentos apresentados nas figuras 2.11 e 2.12, respectivamente, utilizando o modelo probabilístico alternativo para ambas. Assim, adotou-se para o cálculo da probabilidade de espalhamento local uma probabilidade de transmissão  $\lambda = 0.23$  (equação 7), permanecendo o indivíduo infectado por um período de  $\tau = 5$  dias.

Na figura 2.11 considerou-se que no início do processo epidêmico ( $t = 0$ ), havia apenas um indivíduo infectado e  $S(0) = N - 1$  suscetíveis; o sistema evolui desta configuração inicial de tal forma que o número de suscetíveis diminui, à medida que são convertidos em recuperados e imunes.



**Figura 2.11:** Modelo SIR discreto utilizando-se o algoritmo descrito para o autômato celular, no qual foi considerado espalhamento local ( $\Lambda = 1$ ). Representa uma epidemia que se espalha devido aos contatos locais (físicos) em dois instantes de tempo distintos.

Após vários passos de tempo, chega-se ao final do processo epidêmico (quando  $I(t) = 0$ ) e de um experimento de simulação (uma população hipotética em que verificamos o espalhamento de uma epidemia). Nessa simulação foram adotados os parâmetros de controle como sendo  $\Gamma = 0$  e  $\Lambda = 1$  (apenas espalhamento local). Na figura os elementos em cor verde representam os indivíduos da população que se encontram no estado susceptível, os elementos em cor vermelha representam os infectados e os elementos em azul representam os recuperados em cada passo de tempo.

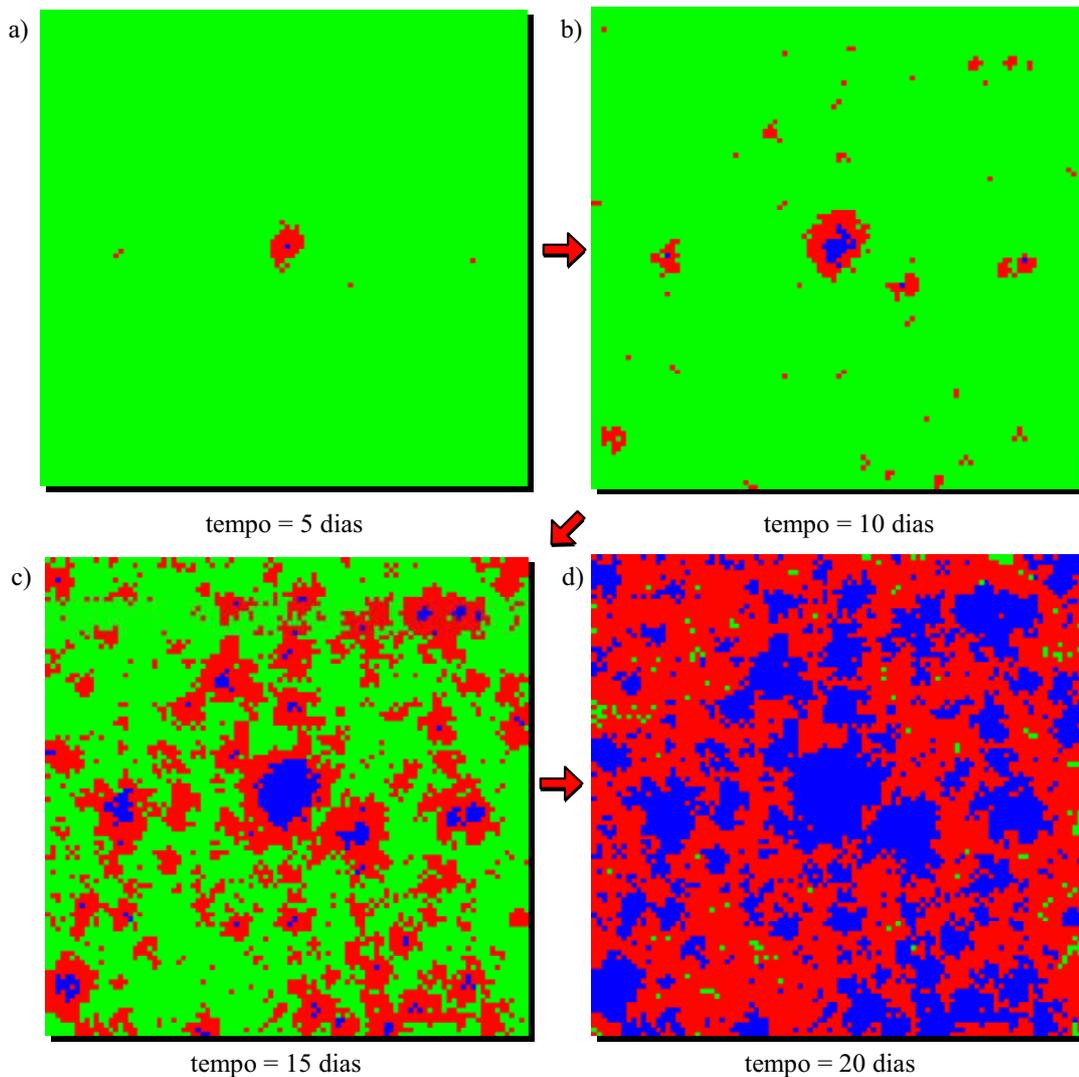
Neste tipo de espalhamento, no qual só existem contatos locais, o processo de contágio é mais lento do que quando estamos trabalhando com o espalhamento global porque o fato do indivíduo possuir mobilidade faz com que surjam novos focos de epidemia a cada instante, que dependendo dos valores ajustados para  $\Gamma$  e  $\Lambda$  pode se desenvolver utilizando somente o espalhamento global ou os dois ao mesmo tempo.

Na figura 2.12 foram utilizados os mesmos parâmetros da figura 2.11 assumindo  $\Gamma = 0.1$  e  $\Lambda = 0.9$ . Observe que a utilização dos contatos globais associados aos contatos locais acaba acelerando o processo de espalhamento de disseminação de uma epidemia pelo fato de criar novos focos infecciosos que, com o passar do tempo, acabam tornando possível um espalhamento distribuído na rede. Estas duas figuras analisadas em conjunto dão uma idéia mais ilustrativa do desenvolvimento de uma epidemia nesse contexto, mostrando a diferença na velocidade de espalhamento da epidemia quando considerada a probabilidade global.

É interessante observar que mesmo para valores pequenos de ( $\Gamma = 0.01$ ) a velocidade de espalhamento da epidemia aumenta bastante em relação a simulação da figura 2.11 mostrada anteriormente. Isto acontece, como já foi comentado, devido a mobilidade dos infectados na população, que aumenta o número de contatos entre os indivíduos na rede, da mesma forma que o protocolo de re-conexão utilizado no modelo de mundo pequeno

[WATTS, STROGATZ, 1998].

Esse efeito do aumento das correlações entre indivíduos na população pode ser observado mais drasticamente nas partes (c) e (d) da figura onde, em um intervalo de cinco dias, houve uma mudança muito brusca quanto ao número de susceptíveis do sistema.



**Figura 2.12:** O mesmo que na figura 2.11, com  $\Gamma = 0.1$  e  $\Lambda = 0.9$ .

Uma análise mais detalhada da evolução temporal da epidemia é mostrada na figura 2.13. Para uma população de  $N = 10^4$  indivíduos foram realizadas as simulações calculando a média sobre os resultados obtidos em 100 experimentos (cada experimento equivalendo a uma população diferente). Inicialmente será analisada a dinâmica do espalhamento de uma

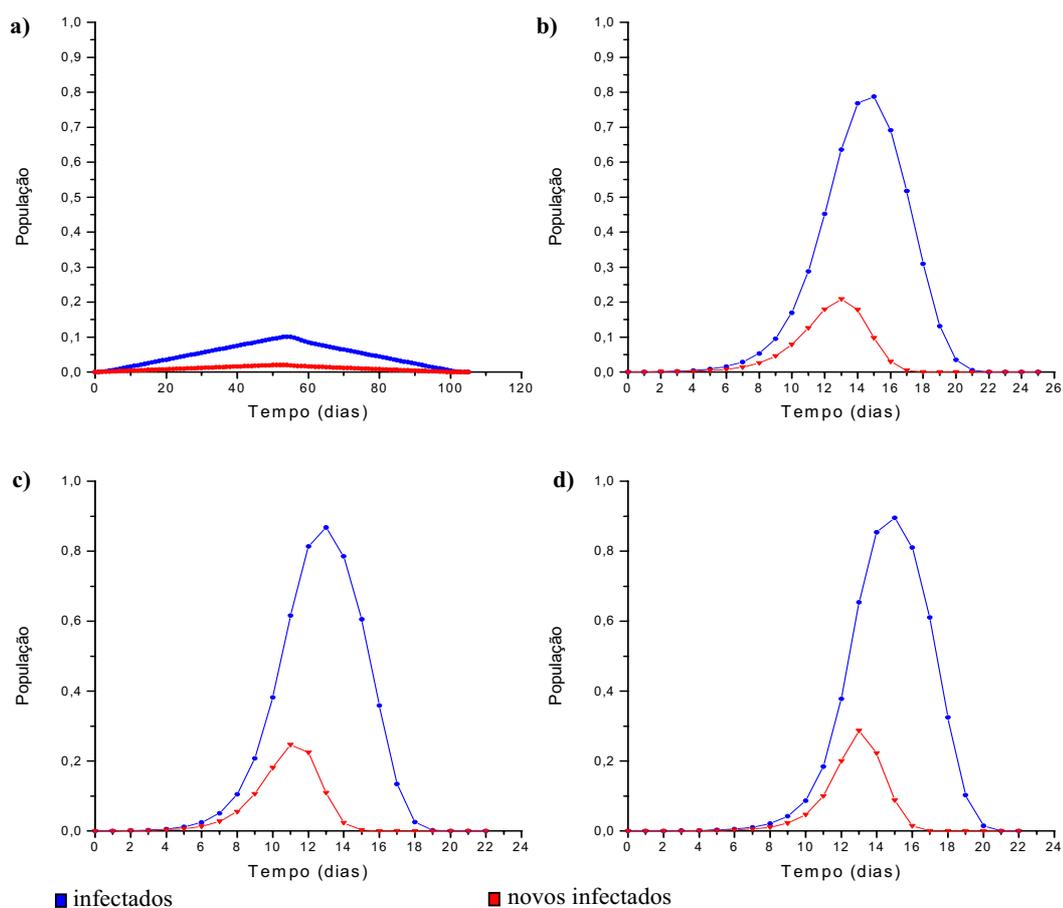
moléstia em populações configuradas inicialmente com apenas um único indivíduo infectado no centro da rede, sendo os demais indivíduos susceptíveis.

Para essas simulações foram utilizadas condições periódicas de contorno e adotado como critério de vizinhança 4 vizinhos diretos (vizinhança de von Neumann). Esta figura apresenta gráficos considerando  $\lambda = 0.9$  e, para se estabelecer uma comparação entre os três cenários possíveis nesse modelo (um espalhamento apenas local, um espalhamento apenas global e um espalhamento local e global), foram adotados os valores para  $\Gamma = 0$  (apenas espalhamento local),  $\Gamma = 0.096$ ,  $\Gamma = 0.1923$  (espalhamento local e global) e  $\Gamma = 1$  (apenas espalhamento global), respectivamente para (a), (b), (c) e (d). As linhas em azul e vermelho representam a média de infectados e de novos infectados no tempo, respectivamente. Aqui, entende-se por novos infectados o número de indivíduos que a cada dia de simulação acabaram de ser infectados.

Pode-se observar em (a) que, quando há apenas contatos locais, a doença se espalha mais lentamente pela população até que não reste mais nenhum infectado ( $I(t) = 0$ ), se comparada à curva (b), (c) e (d). Note que neste tipo de espalhamento, o pico da epidemia atinge um menor número de pessoas do que quando são utilizados contatos globais e locais ao mesmo tempo, como é o caso de (b) e (c) ou apenas contatos globais (em (d)).

Conforme o valor de  $\Gamma$  é acrescido, percebe-se que tal espalhamento ocorre cada vez em um espaço mais curto de tempo, bastando a inserção da mobilidade para que a epidemia levasse de 100 dias ( $\Gamma = 0$ ) para 20 dias ( $\Gamma = 0.096$ ), em média e se torna ainda mais rápida ao comparar (b) e (c). Isto acontece porque nestes dois gráficos tem-se a superposição dos dois tipos de contatos, o que faz com que um indivíduo possa, após ser infectado em qualquer ponto da rede, propagar a doença tanto local quanto globalmente, acelerando desta forma o contágio entre seus elementos.

A curva de novos infectados, mostrada em vermelho no gráfico desta figura, representa quantas pessoas acabaram de ficar doente em determinado dia da simulação, geralmente acompanhando o desenvolvimento da curva de infectados do sistema, que se caracteriza por ter um período de crescimento e após atingir o máximo começa a decair em virtude dos indivíduos que vão se tornando recuperados. Da figura, pode ser notado que a curva de novos infectados atinge o ápice dois dias antes da curva de infectados chegar ao máximo, o que se caracteriza por existirem menos pessoas susceptíveis no sistema para serem infectadas no período em que a epidemia atinge o seu ápice (em analogia a um incêndio numa floresta, seria como se a partir daquele momento faltasse árvores suficientes para que o fogo continuasse a se alastrar).



**Figura 2.13:** Gráficos obtidos a partir de 4 simulações realizando 100 experimentos para cada uma começando com apenas um infectado no centro da rede. As curvas em azul representam os indivíduos infectados e as curvas em vermelho representam o número de novos indivíduos que se infectaram em cada dia da simulação.

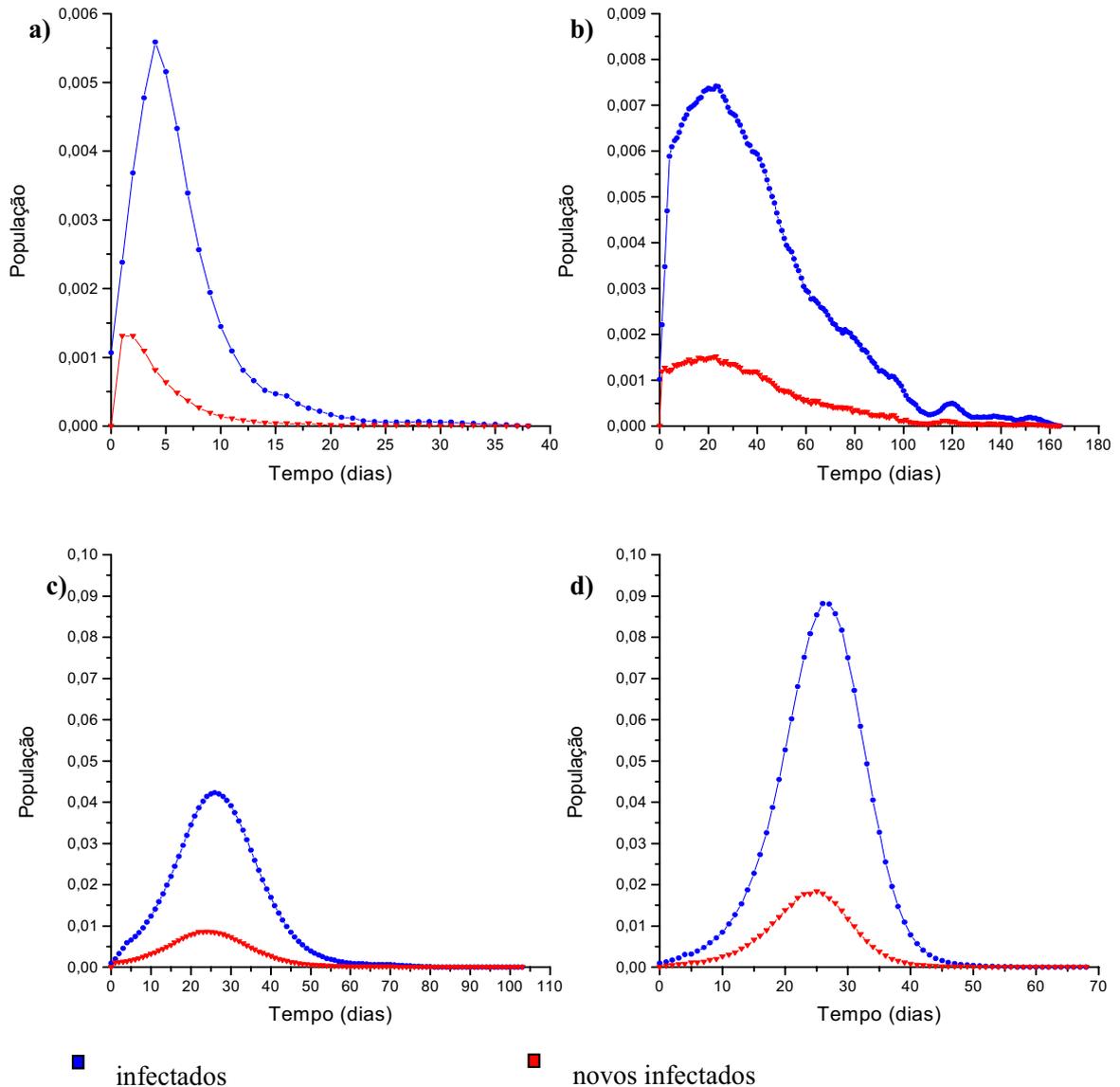
Na figura 2.14 são apresentados os gráficos obtidos a partir de médias calculadas com 35% da rede inicialmente formada por susceptíveis com 10 indivíduos infectados espalhados aleatoriamente pela rede, sendo que os demais indivíduos são mantidos no estado recuperado.

Nessas simulações foram adotadas as mesmas condições utilizadas para os resultados mostrados na figura 2.13. Na parte (a), pode-se notar que a curva de infectados tem um pico inferior ao observado em (b) devido ao fato das populações começarem com aproximadamente 65% de indivíduos recuperados, o que faz com que quando são assumidos apenas contatos locais no sistema estes acabem encontrando barreiras formadas por indivíduos recuperados, impedindo, assim, o avanço da doença. Tal barreira poderia ser vista como a taxa de indivíduos previamente medicado (ou já imunes a tal doença), de maneira similar à taxa de tratados inserida periodicamente [ALVES, GAGLIARDI, 2006].

Em (b), (c) e (d) percebe-se que à medida que é aumentada a mobilidade dada pelo parâmetro global  $\Gamma$ , tem-se um deslocamento do pico da epidemia no sentido que leva o espalhamento a ocorrer em menos tempo nas populações. Além disso, nota-se em (c) e (d) um aumento acentuado de indivíduos infectados quando a epidemia atinge este ápice uma vez que, com tal aumento no valor da variável global  $\Gamma$ , mais pessoas da rede acabam sendo infectadas simultaneamente na população, sendo quase toda a população atingida, como é o caso de (d), que apresenta um pico epidêmico de aproximadamente 850 pessoas infectadas ao mesmo tempo, visto que o resto da população formada de recuperados e susceptíveis que ainda não entraram em contato com a doença.

Novamente, ao compararmos (a) e (b), temos que em (a) o espalhamento ocorre mais rapidamente do que em (b), porém atingindo menos pessoas da rede, dado que nesta o espalhamento ocorre apenas localmente, ao contrário de (b), em que ocorre a superposição dos dois tipos de contatos, que caracterizam o efeito de *mundo pequeno* nesse modelo [WATTS, STROGATZ, 1998], onde cada indivíduo tem uma determinada mobilidade no

sistema e assim ocorre uma maior propagação desta doença pelo mundo que estamos simulando, sendo esta a principal característica desta nova maneira de se modelar.



**Figura 2.14:** O mesmo que na figura 2.13, começando com 10 infectados e 35% da rede susceptível (os indivíduos restantes são deixados como imunes à doença que se propaga, que pode ser visto como um efeito de vacinação parcial da população).

Portanto, ao longo deste capítulo foram apresentadas três maneiras para se modelar um mesmo sistema, o modelo SIR: a abordagem compartimental, a baseada em um modelo baseado em um autômato celular clássico (contatos apenas locais) e a abordagem alternativa baseada em autômatos celulares (apresentada neste trabalho na seção 2.2.2) [ALVES,

GAGLIARDI, 2006]. Neste momento, é importante salientar as principais diferenças entre cada um deles.

Como vimos anteriormente, o modelo SIR na visão compartimental é construído a partir de equações diferenciais e o que se consegue modelar com tal abordagem é um sistema baseado na população ao invés da figura do indivíduo, sendo um número o que descreve quantos indivíduos se encontram em cada compartimento do sistema, ou seja, não há uma discriminação de cada indivíduo. Consequentemente, esse sistema trabalha com uma massa populacional dividida em compartimentos, cujas interações intrínsecas consideram que todos estão em contato com todos a cada instante da simulação, ou seja, não há individualidade.

Na visão baseada na figura do indivíduo para o modelo SIR a situação já se inverte, ou seja, ao invés da população ser tratada como uma massa ela passa agora a ter indivíduos que possuem suas próprias características e localização no espaço e portanto, trata-se de um sistema com heterogeneidade espacial. O fato é que na abordagem clássica de autômatos celulares cada indivíduo interage apenas com seus vizinhos e a dinâmica destas interações ainda se mantém muito presa no entorno de cada indivíduo, que se mantém fixo no espaço, ou seja, apesar de haver a individualização do espaço, não há mobilidade associada a cada indivíduo. Este cenário também é tão irreal quanto o primeiro, uma vez que para se transmitir uma doença contagiosa neste tipo de rede é necessário que ela corra de vizinho a vizinho, o que torna muito lento o processo para que todos se comuniquem entre si e sejam infectados, como se o mundo fosse muito grande.

Assim, a principal diferença do modelo proposto nesta seção com relação às demais formas de se modelar o modelo SIR, a despeito desta também ser uma modelagem baseada no indivíduo, reside justamente na heterogeneidade das interações entre os indivíduos que compõem uma população. O simples fato de se atribuir contatos de longa distância entre os elementos desta população já torna possível a reprodução de cenários muito mais

interessantes, abrangendo espalhamentos muito rápidos (todos são atingidos pela doença em poucos dias) ou muito lentos (quanto menor for a taxa de espalhamento global, mais próximo do modelo clássico de autômato celular o sistema se comporta e mais local a dinâmica se apresenta). Particularmente, é tal heterogeneidade de interações, proporcionada a partir da utilização de contatos de longa e curta distância em populações, o que rege na vida real a transmissão de epidemias, sendo esta, portanto, a característica chave para a modelagem de doenças infecto-contagiosas de uma maneira geral.

Diante disto, os conceitos básicos foram sendo apresentados durante este capítulo com o propósito de contextualizar a base que utilizamos para o desenvolvimento de um modelo para simular o comportamento de transmissão da dengue em uma população [PESSOA, GAGLIARDI, ALVES 2004; GAGLIARDI, SILVA, ALVES, 2006] e que estará sendo apresentado no capítulo a seguir. Particularmente, o modelo da dengue é utilizado no serviço de simulação IntegraModel, discutido ao longo do capítulo 4.

### **CAPÍTULO 3 – A MODELAGEM DO PROCESSO EPIDÊMICO DA DENGUE: SIMULAÇÃO E REALIDADE**

No capítulo anterior apresentamos alguns pressupostos sobre a modelagem de epidemias de uma maneira geral na qual algumas das técnicas mais utilizadas no estudo de processos epidêmicos foram apresentadas e discutidas a fim de contextualizar a complexidade envolvida ao se modelar um processo epidêmico. Ainda no capítulo anterior, discorreremos sobre algumas soluções encontradas para caracterizar tal complexidade e então uma nova formulação alternativa, centrada na figura do indivíduo baseada em redes de autômatos celulares cujas interações entre seus elementos são descritas em termos de dois tipos de contatos distintos, locais e globais.

Nesse sentido, um estudo preliminar foi apresentado para essa nova forma de modelagem, no qual vários fenômenos de interesse foram discutidos de forma a possibilitar uma compreensão mais geral do comportamento desse tipo de sistema à luz das interações intra-populacionais e seus efeitos obtidos ao se adotar regras de transição de estados, tomando praticamente a grande maioria do capítulo anterior. Apesar de sua simplicidade, tais modelos tidos como hipotéticos são capazes de explicar razoavelmente bem vários padrões observados para alguns tipos de infecções do mundo real, tais como o sarampo quando o número de casos é anotado semanalmente ou diariamente.

Como já havíamos comentado ainda no capítulo anterior, podemos buscar maior realismo na modelagem ao incorporar outros esquemas compartimentais, ou porque mudamos o esquema de setas que esquematizam a transição entre um conjunto de estados que a população deve passar ou porque simplesmente acrescentamos novos estados possíveis para a população modelada.

Assim, por exemplo, podemos trabalhar com o modelo SIRS admitindo perda de imunidade, ou seja, um indivíduo recuperado permanece nesse estado por um período finito de tempo, voltando posteriormente ao estado susceptível e formando assim um ciclo. Tal modelo ainda bastante simples de se trabalhar é freqüentemente usado para o estudo de doenças recorrentes, onde uma pessoa pode ser contaminada mais de uma vez como é o caso da gripe, por exemplo.

Da mesma forma, podemos adicionar um período de latência E a esta forma de modelar, discriminando assim o tempo em que a doença se encontra no corpo do hospedeiro, mas ainda não se tornou contagiosa, surgindo assim um modelo SEIR (ou o modelo SEIRS, quando existe recorrência da doença).

Assim, o modelo SEIR apresenta um comportamento similar ao do modelo SIR (discutido no capítulo anterior), exceto pelo estado Exposto, que caracteriza esse período de incubação onde o indivíduo não atua como um agente infeccioso (não transmite a doença).

Vale salientar mais uma vez que tais categorias podem ser utilizadas em várias técnicas de modelagem distintas, como é o caso dos modelos compartimentais baseados em equações diferenciais parciais (também utilizadas para analisar a componente espacial nesses modelos como um processo de difusão) ou modelos baseados no indivíduo.

Outros aspectos ainda podem ser modelados utilizando esse ferramental teórico como, por exemplo, considerar a demografia da população e trabalhar como uma população aberta (deixar o número total de indivíduos variar) no sentido de que queremos estudar os efeitos da taxa de mortalidade causados pela doença na população (fator que pode ser essencial ao se modelar doenças como a peste em populações de idade média ou mesmo das várias ondas de gripe que acometeram as populações do começo do século XX).

Existe também uma vasta literatura de modelos que consideram a estrutura etária da população [ANDERSON, MAY, 1991; EUBANK, 2004], criando modelos mais complexos

para estudar epidemias, bem como é possível implementar e estudar esquemas de vacinação aproveitando o conceito de limiar epidêmico para definir qual parcela da população deve-se vacinar (quando uma vacina for disponível) de maneira que uma epidemia não se instale.

Uma aplicação mais ousada da formulação alternativa apresentada no capítulo anterior é a tentativa de se modelar a disseminação da dengue em uma população, o que acarretará em introduzir nesse modelo o efeito da população de vetores (o mosquito *Aedes aegypti*) interagindo com a população de humanos.

Antes de fazer isso, entretanto, deve-se lembrar brevemente que a dengue é uma doença comum em regiões tropicais do mundo causada por um arbovírus e transmitida a populações humanas (hospedeiros) de todas as idades e ambos os sexos, somente pela fêmea adulta do mosquito *Aedes aegypti* (o vetor).

Este ciclo epidemiológico sustenta-se em áreas urbanas e após a picada a um indivíduo infectante, o mosquito nunca se recupera da infecção, passando a infectar novos hospedeiros a cada refeição de sangue necessária para sua oviposição até o dia de sua morte, sem nada sofrer com a presença do vírus em seu organismo.

Assim, neste capítulo iremos discutir vários graus de complexidade que podemos incorporar a modelagem de um processo epidêmico, deixando vários desses aspectos de lado por ora em detrimento da simplificação e entendimento de um modelo para a transmissão da dengue, que estará sendo aqui proposto empregando autômatos celulares.

Particularmente, as regras de interação entre seus elementos adotadas no modelo da dengue são as mesmas do modelo alternativo apresentado no capítulo anterior, exceto pelo fato de que agora a interação ocorre entre duas populações distintas, cada qual com suas regras de transição de estados particular.

### 3.1 UM MODELO PARA ESTUDAR A TRANSMISSÃO DA DENGUE

Para descrever a dinâmica da transmissão do vírus da dengue dentro da comunidade de hospedeiros, devido à interação com a população de mosquitos, foram classificados individualmente os indivíduos inicialmente com uma estrutura similar aos modelos clássicos em epidemiologia de compartimentalização de uma população em classes de indivíduos, de acordo com o *status* da doença de cada um. Assim, a dengue pode ser modelada num primeiro momento admitindo-se que a população de hospedeiros com  $N_h$  indivíduos, no qual  $N_h = W_h \times H_h$ , sendo  $W_h$  e  $H_h$  a *largura* e a *altura* da rede bidimensional respectivamente. Em qualquer instante de tempo a população de hospedeiros pode ser dividida nas seguintes classes: os indivíduos susceptíveis que podem ser infectados pelo vírus através da picada do mosquito ( $S_h(t)$ ), os indivíduos que foram expostos ao vírus ao serem picados, mas que ainda estão incubando o vírus e por isso não infectam ( $E_h(t)$ ), os infectantes ( $I_h(t)$ ) e os indivíduos recuperados da infecção ao adquirirem imunidade ( $R_h(t)$ ). Uma simplificação adotada aqui é a de considerar que a população total de hospedeiros

$$N_h(t) = S_h(t) + E_h(t) + I_h(t) + R_h(t) \quad (3.2.1)$$

é mantida constante. Esta é uma hipótese realística desde que queremos modelar a infecção por um único sorotipo e que, portanto, não há mortalidade adicional causada pela doença (nesse caso a doença é benigna, e os indivíduos sintomáticos não são levados à morte [YANG, 2003; FERREIRA, YANG, 2003]).

No caso da população de *mosquitos adultos*, os  $N_m$  indivíduos também podem ser divididos em classes como: susceptíveis ( $S_m(t)$ ) que não contém o vírus da dengue, sendo inofensivo para a população de hospedeiro, infectados porém não infectantes ( $E_m(t)$ ) chamados de expostos e os infectantes ( $I_m(t)$ ). Particularmente, aqui é considerado que os

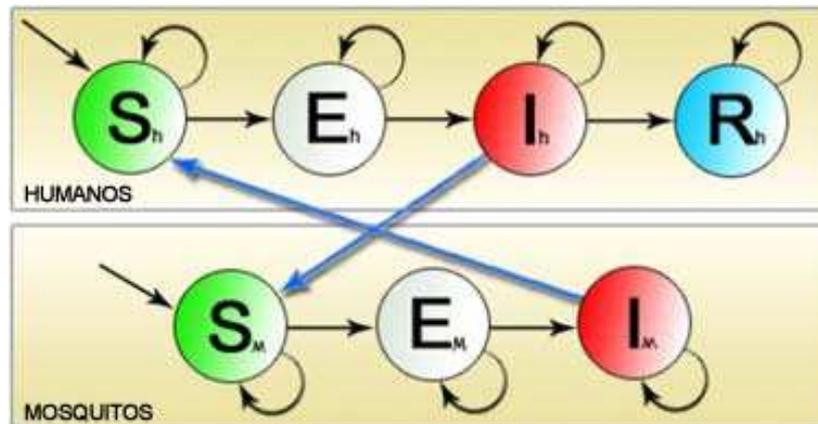
mosquitos, uma vez na classe dos infectantes, permanecem nesse estado até sua morte já que o vírus não interfere no ciclo de vida do vetor [YANG, 2003]. Diferentemente da consideração feita sobre a população de hospedeiros humanos, a população total de mosquitos

$$N_m(t) = S_m(t) + E_m(t) + I_m(t) \quad (3.2.2)$$

não é constante no tempo, uma vez que o tempo de vida do mosquito varia de alguns dias a meses, muito menor do que a vida média de um ser humano. Do que foi visto até o momento, a população de humanos é modelada por um esquema de transição de estado do tipo SEIR e a do mosquito, por um esquema do tipo SEI. Nesse esquema, a transmissão da doença começa no exato momento em que o mosquito infectado ( $I_m$ ) pica um indivíduo que não tenha adquirido a doença anteriormente e nem possua o vírus em seu corpo, ou seja, um indivíduo susceptível ( $S_h$ ). Esse indivíduo passará para o estado de incubação ( $E_h$ ) no qual permanecerá por  $\tau_{E_h}$  dias (de 5 a 7 dias, por exemplo), transitando então para o estado infectado ( $I_h$ ), podendo enquanto isso infectar mosquitos no estado susceptível ( $S_m$ ), caso entrem em contato com esse humano e se alimentem de seu sangue. O período de infectado na população de humanos  $\tau_{I_h}$  dura cerca de 3 a 7 dias e é chamado de período infeccioso (viremia). Após a transição ao estado recuperado ( $R_h$ ), o indivíduo aparentemente apresenta imunidade permanente ao vírus [FURLAN, GAGLIARDI, ALVES 2004; FURLAN, GAGLIARDI et al 2006].

A população de mosquitos comporta-se de maneira similar. Todo vetor que atinge a fase adulta é susceptível ( $S_m$ ) até que entre em contato com algum indivíduo humano que esteja no estado infeccioso ( $I_h$ ). Ao efetuar a picada, o vetor passa para o estado de incubação também chamado estado exposto ( $E_m$ ). O tempo que o vírus leva para alojar-se nas glândulas salivares e efetivar a transmissão, conhecido como período de incubação extrínseco ( $\tau_{E_m}$ ) e varia de 7 a 10 dias, variando bastante na literatura tal como em [LUZ et al, 2003] em que argumenta-se que para baixas cargas virais o período de incubação extrínseco pode levar até

25 dias. Ao se tornar efetivamente transmissor da doença, o mosquito permanece infectado até a sua morte no estado de infectante ( $I_m$ ). Logo, a figura 3.1 apresenta um diagrama do processo de transmissão descrito para a interação entre mosquitos e hospedeiros.



**Figura 3.1** - O esquema do modelo de transmissão da dengue proposto, representando os estados da doença. As setas mais espessas (em azul) representam a interação entre as duas populações para que o espalhamento ocorra de uma para a outra (através da picada do mosquito) e as setas mais finas (em preto) representam a transição de estados em cada população.

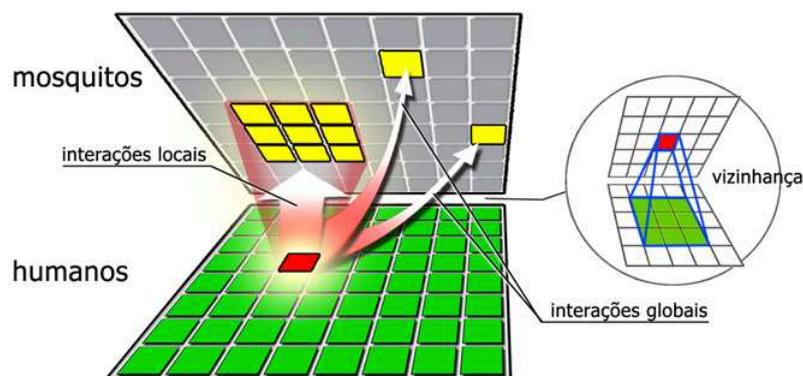
É válido notar que até então, nos modelos anteriormente apresentados, a transmissão ocorria dentro de uma mesma população, na qual um humano transmitia alguma doença para outro humano a partir de uma vizinhança em comum, podendo ser vista como uma transmissão horizontal, dado que ela ocorre no mesmo nível populacional. Ocorre que no modelo da dengue isso não é verdade, visto que o contágio de um humano a outro só é possível através do mosquito, ou seja, trata-se de uma transmissão vertical cruzada de dois estágios: humano  $\rightarrow$  mosquito e mosquito  $\rightarrow$  humano.

Portanto, tal transmissão agrega a esse modelo mais complexo uma característica tridimensional para a transmissão, uma vez que se torna necessário estabelecer o vínculo entre mundos diferentes (o dos humanos e o dos mosquitos), cada qual com sua dimensão distinta. Logo, para esse tipo de modelagem tanto a vizinhança quanto os contatos atuam em outro nível, agora não dentro de uma mesma população, mas de uma população para outra. Apesar disso, a idéia desses contatos continua sendo praticamente a mesma. O que muda é que agora

a vizinhança dos humanos se encontra na população de mosquitos, da mesma forma que a dos mosquitos está na população de humanos.

Com efeito, consideramos esta uma nova maneira de modelar em que além de se definir as regras de transição entre os estados em autômatos celulares para cada população interagente, definimos regras de interação entre esses dois autômatos. Tal forma de se modelar interações interindividuais empregando autômatos celulares é distinta na literatura e trata-se de uma novidade para esse tipo de problema que envolve a modelagem de sistemas biológicos de uma maneira geral [GAGLIARDI, SILVA, ALVES, 2006].

Nesse sentido, tivemos de pensar na definição de um novo esquema de vizinhança capaz de estabelecer um elo de ligação entre as duas redes interagentes. Ou seja, a posição geográfica da rede é o que determinará o seu elemento correspondente na outra rede. Para exemplificar o que acabou de ser discutido, se o elemento estiver, por exemplo, numa posição [49,60] na rede dos mosquitos, o seu correspondente será a área em volta do elemento [49,60] na rede dos hospedeiros.



**Figura 3.2** – Esquema da interação entre dois autômatos representando a população de humanos e a de mosquitos, na modelagem da transmissão da dengue. No destaque mostramos o novo esquema de vizinhança que tem de ser adotado nesse tipo de modelo.

Portanto, essa área será determinada pelo tamanho da vizinhança (como visto no capítulo anterior na discussão da figura 2.6), que corresponde ao raio de vizinhança deste indivíduo [49,60] (ver capítulo 2, seção 2.2.1 para maiores detalhes). Assim, todos os

indivíduos que estiverem dentro desse raio poderão transmitir a dengue a este indivíduo localmente. A Figura 3.2 apresenta um esquema da forma como tais interações ocorrem.

Ainda pensando na vizinhança, para populações de tamanhos diferentes interagindo entre si no modelo é mantida uma proporção entre os elementos e sua vizinhança. Logo, se  $N_m$  é maior do que  $N_h$ , por exemplo, como existem mais mosquitos interagindo com uma população de humanos, para cada humano haverá mais mosquitos interagindo em sua vizinhança e de forma inversamente proporcional, no caso da população de mosquitos, para um único mosquito haverá menos humanos interagindo. Com isso consegue-se trabalhar com uma densidade de mosquitos para cada humano da rede.

Assim, com esse novo protocolo de interação entre autômatos, manter-se-á para cada um as regras de transição estabelecidas na seção anterior. As probabilidades para cada população ficam definidas: para a população de humanos especificando-se que,

$$P_{L_h} = 1 - (1 - \lambda_m)^{I_m} \text{ e } P_{G_h} = \frac{\rho_m N_{I_m}}{N_m} \quad (3.2.3)$$

e de maneira similar, para a população de mosquitos tem-se que,

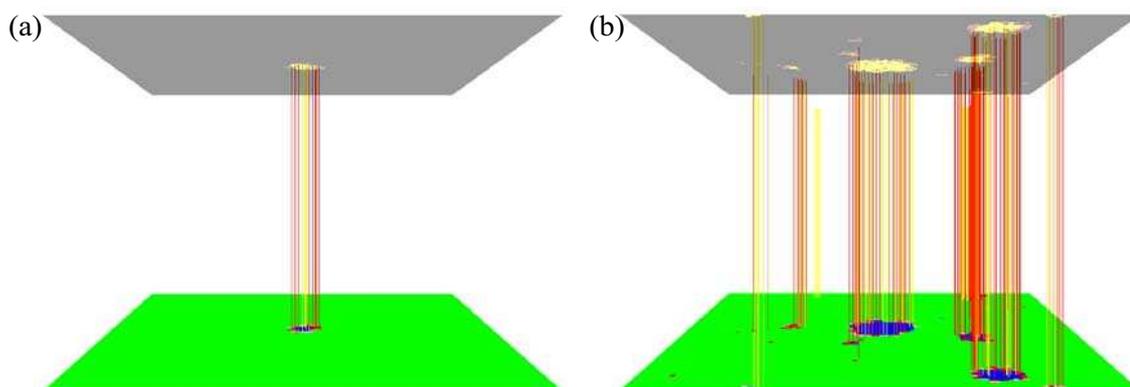
$$P_{L_m} = 1 - (1 - \lambda_h)^{I_h} \text{ e } P_{G_m} = \frac{\rho_h N_{I_h}}{N_h} \quad (3.2.4)$$

onde  $I_m$  é o número de mosquitos infectantes do autômato dos mosquitos que interagem como vizinhos de um indivíduo susceptível no autômato de humanos e  $I_h$  é o seu análogo representando o número de humanos infectados que podem ser picados por um mosquito susceptível. Para efeito de simplificação, estará sendo adotado em toda a discussão deste capítulo que  $\lambda_h = \lambda_m = \lambda$ , ou seja, a probabilidade de picada é a mesma do mosquito para o humano e do humano para o mosquito e será representada, portanto, apenas como  $\lambda$ . Nesse caso, a interpretação da probabilidade  $\lambda$  é fundamental já que ela passa a ser a probabilidade de picada, comum às duas populações. É interessante notar que nesse modelo as

probabilidades globais em cada população modelam as mobilidades que ambas podem ter na rede, em princípio distintas entre si.

A Figura 3.3 mostra a simulação desse modelo em dois instantes distintos de tempo em uma rede quadrada simples assumindo que inicialmente a rede tinha um humano infectado. Assim na parte (a) da figura podemos ver o espalhamento da epidemia nos primeiros instantes em que é detectado o aparecimento de um aglomerado de casos em torno desse primeiro caso.

A parte (b) da figura mostra a epidemia já instalada na população e o aparecimento de vários aglomerados (clusters) na rede em lugares distintos dessa cidade virtual.



**Figura 3.3** – Simulação do modelo baseado na interação entre dois autômatos, em dois instantes distintos de tempo.

Esse fato é o efeito da mobilidade das duas populações modeladas pelas probabilidades globais. Com esse ferramental é possível testar várias outras propriedades dos mecanismos mais gerais do espalhamento da dengue.

Realmente com esse laboratório virtual podemos fazer vários estudos adicionais em todo espaço de parâmetros [GAGLIARDI, SILVA, ALVES, 2006]. Por exemplo, ao testar o efeito de fixar uma porcentagem da população inicial de mosquitos no estado de infectantes, acarreta que a infecção se propaga preferencialmente de forma local, apenas os contatos mais próximos acabam por se infectar, e como os vetores estão distribuídos aleatoriamente, esta disposição dos vetores favorece o desenvolvimento da epidemia localmente.

De uma maneira geral, é possível observar também que o tamanho da rede (da nossa cidade virtual) acaba influenciando bastante no dano causado pela probabilidade de picadas, já que a rede aumentando o número de indivíduos acaba aumentando também o número possível de casos da doença e conseqüentemente mais gente se tornaria infectada, o que de certa forma faria que a mobilidade da população de humanos ganhasse maior força à medida que mais gente se torna agente infeccioso no sistema (é importante deixar claro, todavia que nesse modelo não são adotadas medidas de controle para o avanço das ondas infecciosas ainda). Mesmo assim, nesse caso podemos aferir os resultados frente a tamanhos de municípios diferentes, no mínimo, além de poder testar outros parâmetros espaciais importantes na dinâmica de transmissão da dengue.

Assim, a seguir, estaremos apresentando uma discussão sobre vários resultados obtidos através do modelo aqui apresentado nesta seção e uma posterior discussão confrontando-os com os dados reais da dengue, presentes na literatura atual.

### 3.2 RESULTADOS PARTICULARES DO MODELO DA DENGUE

Os resultados apresentados a seguir estão organizados de forma a possibilitar uma maior compreensão do comportamento do modelo da dengue. Para isso, foi inicialmente recuperado da literatura um intervalo de valores para a probabilidade de infecção  $0 < \lambda \leq 0.05$ , no qual a transição existente entre um cenário endêmico e um cenário epidêmico se apresenta de maneira bem tênue, ao mesmo tempo em que a mobilidade dos elementos, por sua vez, contribui de maneira contundente para sua ocorrência [LUZ et al, 2003; FENG e VELASCO-HERNÁNDEZ, 1997]. Contudo, à luz dos resultados que estarão sendo apresentados neste capítulo, ainda se faz necessária uma comparação com padrões de dados reais e portanto, o que estará sendo visto aqui humildemente são apenas indicações do

que seria um conjunto de parâmetros interessante para possíveis validações futuras mais detalhadas.

Aqui, são agrupados três conjuntos de dados obtidos a partir da média de 100 experimentos distintos em uma cidade virtual para os parâmetros do modelo de acordo com os valores apresentados nas legendas das figuras a seguir, de forma a destacar o dano médio provocado no sistema para cada configuração em particular, bem como o tempo médio para que o mesmo tenha ocorrido, além do número total de elementos atingidos em média para o referido número de realizações, de maneira similar ao estudo que Haas, Caliri e Silva efetuaram ao analisar a duração e o comportamento de uma epidemia [HAAS, CALIRI, SILVA, 1999]. Tal conjunto de resultados foi obtido justamente com o intuito de tentar validar e compreender melhor o comportamento do modelo, conhecendo quais configurações de parâmetros melhor se adequariam e reproduziriam um espalhamento cuja eficiência se apresentasse similar à do fenômeno da transmissão da dengue.

Todas as simulações apresentadas a seguir foram configuradas de forma a iniciar com uma rede tomada por elementos (humanos e mosquitos) susceptíveis, com apenas um infectado bem ao centro da rede.

Além disso, para todos os cenários a seguir a rede de mosquitos apresenta apenas contatos locais e as probabilidades de transmissão  $\lambda$  são as mesmas da população de humanos. Tanto o controle de vetores quanto a taxa de mortalidade de mosquitos e humanos para esse conjunto de resultados ainda não foi inserida. Vale explicar mais uma vez que o motivo que nos leva a adotar aqui tais restrições para esses dois mecanismos no modelo reside justamente na intenção de simplificar o estudo aqui elaborado em prol da compreensão dos dados obtidos.

No que se refere à modelagem dos mosquitos, é adotado um tempo de incubação de uma semana (sete dias) após o mosquito ter sido contaminado pelo sangue de um humano

contendo o vírus da dengue transitando ao estado infectado após isso, quando é capaz de transmitir a doença para outros humanos, no qual permanece até o término da simulação.

Mais uma vez, por motivos de simplificação, aqui foi modelado apenas um tipo de sorotipo do vírus transmissor da dengue. É interessante saber que quatro são os sorotipos de vírus catalogados que causam a dengue: dengue-1, dengue-2, dengue-3 e dengue-4, podendo em combinação provocar a dengue hemorrágica ou síndrome do choque da dengue, ou seja, após uma segunda infecção do humano pela doença. Contudo, aqui as regras de infecção como descritas anteriormente consideram que um humano e um mosquito qualquer podem contrair a doença apenas uma única vez, ainda por motivos de simplificação. Por isso na modelagem usada para os resultados a seguir, tanto a dengue hemorrágica quanto a síndrome do choque da dengue não são consideradas para simplificar o estudo do modelo.

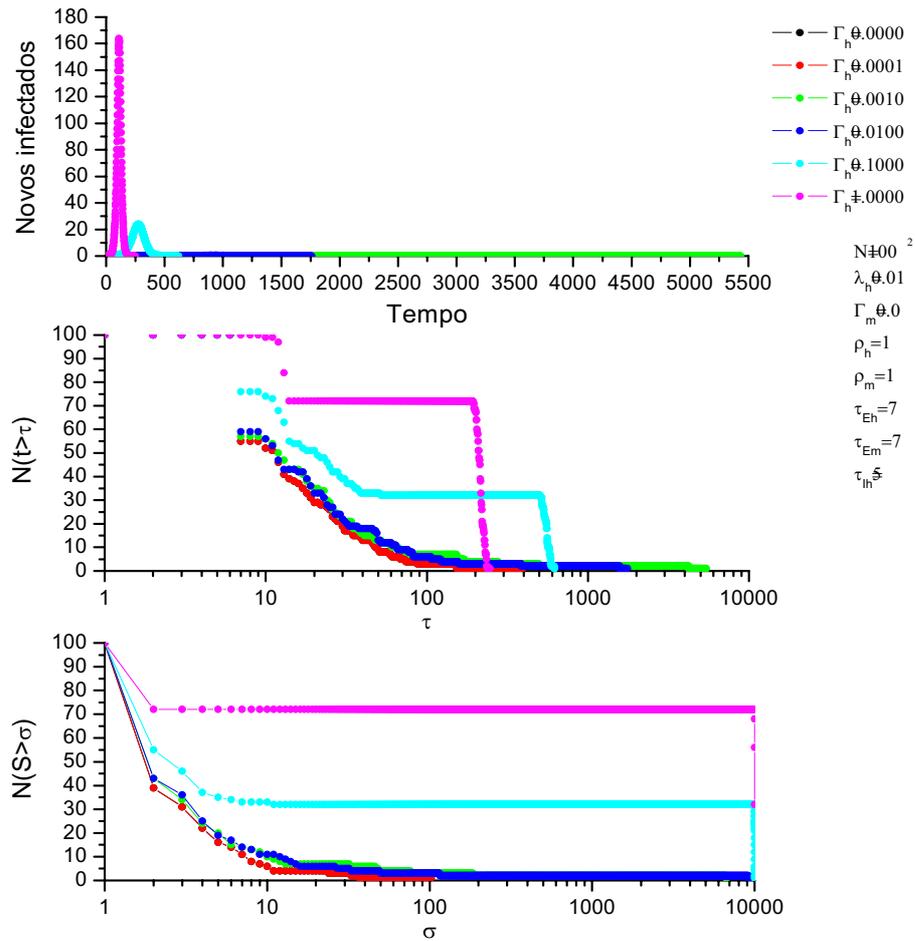
A escolha de um cenário como esse foi baseada na idéia de uma população virgem de infecção (que nunca houvera entrado em contato com a doença) cuja transmissão acontece a partir do momento que um caso é importado de outra região e ali se instala. O que se pretendeu com isso foi compreender melhor o comportamento do sistema ao se introduzir a mobilidade mesmo para cenários onde a epidemia se desenvolve com dificuldade e com isso compreender melhor qual é o conjunto de parâmetros chave que consegue caracterizar de maneira mais próxima no modelo proposto o comportamento da doença como encontrado na literatura [YANG, 2003; LUZ et al, 2003; FENG e VELASCO-HERNÁNDEZ 1997].

Surge com isso a necessidade de se pensar em como parametrizar adequadamente o modelo, sendo as escalas de tempo a serem adotadas, taxas de picadas, probabilidades de transmissão, tempos de incubação e infecção do humano e do mosquito, densidade de mosquitos para cada humano apenas alguns exemplos de fatores reais a serem considerados para podermos focar nossos estudos no ponto de real interesse de seu funcionamento. Nesse sentido, o primeiro ponto que se pensou foi o da escala de tempo, uma vez que o modelo

retorna apenas passos de tempo, sendo algo cuja dimensão poderia ser tanto segundos quanto anos, dependendo, portanto de outras referências para ser corretamente caracterizado.

Contudo, se pensarmos em uma taxa de picadas por dia que cada mosquito é capaz de efetuar, isso se torna uma medida bem mais palpável e conseqüentemente, torna possível determinar a escala de tempo obtida para as simulações desde que modelada corretamente com esses valores. Por isso, aqui foi adotado como intervalo de valores da probabilidade de transmissão  $\lambda$  o resultado do produto da taxa de picadas dos mosquitos pela probabilidade de infecção de um vetor ou humano tal como os parâmetros  $\alpha_i$  e  $\beta_i$  descritos por Feng e Velasco-Hernández em seu modelo baseado em equações diferenciais (equivalentes à  $\lambda_h$  e  $\lambda_m$  no modelo aqui proposto), no qual tal faixa de valores, variando de 0 a 0,05, já se apresenta na escala em que cada unidade de tempo representa um dia [FENG e VELASCO-HERNÁNDEZ, 1997]. Assim, em nosso modelo cada unidade de tempo da simulação representa o resultado de um dia de interação entre humanos e mosquitos sendo, portanto tais resultados representam amostras periódicas a cada 24 horas de simulação de como o sistema está se comportando.

Dentro desse contexto, os resultados obtidos indicam que para os valores adotados de  $\lambda$  o espalhamento não se desenvolve para algumas realizações, extinguindo-se em poucos dias. De uma maneira geral, para valores de  $\lambda$  menores do que 0,005 é o que ocorre para a grande maioria de simulações, sendo necessário um valor consideravelmente grande de  $\Gamma$  para que algum espalhamento ocorra, e mesmo assim a dinâmica passa a ser regida praticamente apenas pelos contatos globais que cada indivíduo possui. Mesmo assim, para  $\lambda \rightarrow 0$  nem mesmo o espalhamento global se torna eficaz para que o espalhamento consiga se propagar. Isso ocorre justamente no nos gráficos da Figura 3.4, e explica justamente porque a linha negra, que se refere à situação cujas duas populações possuem contatos apenas locais (ou seja,  $\Gamma_h = \Gamma_m = 0$ ), não é visível para os três gráficos apresentados.



**Figura 3.4** – Simulações para  $\lambda_h = 0,01$  configuradas como descritas na legenda para a população de humanos. A primeira parte da figura representa o número de novos infectados humanos, onde cada linha representa a média em 100 experimentos para cada valor de mobilidade  $\Gamma_h$  descrito. Na segunda parte é apresentado o número de experimentos  $N(t > \tau)$  que conseguiram atingir tantos dias descritos no eixo horizontal do gráfico (aqui em escala logarítmica para facilitar a visualização). Por fim, a parte inferior da figura mostra o número de experimentos  $N(S > \sigma)$  que conseguiram infectar o total de  $\sigma$  humanos do sistema.

Nesse caso ela durou pouco mais do que o período de tempo  $\tau_{lh}$ , o que pode levar o leitor à conclusão de que o espalhamento não atuou com força suficiente para a configuração correspondente a esse conjunto de realizações estocásticas em média para que o espalhamento se mantivesse ao longo do tempo, sendo portanto o número de reprodução para cada infectado do sistema em média inferior a 1 para a grande maioria de populações simuladas, o que

explicaria de certa forma tal circunstância. No entanto, aqui a mobilidade faz toda a diferença já para valores de  $\Gamma_h = 0,001$ .

O que os resultados indicam na Figura 3.4 de uma maneira geral é que exceto para  $\Gamma_h = 0$  e  $\Gamma_h = 0,0001$ , à medida que o valor de  $\Gamma_h \rightarrow 1$  menos tempo é requerido para que o processo de espalhamento se extinga, tornando-se proporcionalmente mais eficaz, ou seja, atingindo cada vez mais gente. Contudo, é interessante observar certas particularidades para a configuração apresentada nessa figura. É notável aqui o fato da curva vermelha representando as simulações com mobilidade  $\Gamma_h = 0,0001$  terminar mais rápido do que a curva verde representando as simulações com mobilidade  $\Gamma_h = 0,001$ .

Nesse sentido, de acordo com a dinâmica proporcionada pela heterogeneidade das interações adotada neste modelo e discutida no capítulo anterior, o que se poderia esperar é que a curva vermelha tivesse uma duração maior do que a curva verde, pelo simples fato do espalhamento na primeira ser muito mais dependente de contatos locais do que globais como já não acontece tanto na outra.

Ocorre que isto é o interessante para essa faixa de parâmetros adotados: aparentemente, nem todos os experimentos aqui conseguem ter força o suficiente para conseguir perdurar e se desenvolver, uma vez que o valor de  $\lambda$  aqui se situa em uma zona de transição em que o sistema passa de uma faixa (endêmica) onde a transmissão não tem força para continuar por ser menor ou igual a um indivíduo atingido a cada unidade de tempo, extinguindo-se em poucos dias, para uma situação ainda endêmica mas com maior velocidade no qual o fato do espalhamento ter conseguido chegar à cerca de 390 dias se deve à endemia provocada pela própria característica probabilística do sistema. Em outras palavras, aí existe uma transição de um sistema onde a doença não se propaga (como ocorreu para  $\Gamma_h = 0$ ) para um em que ela se propaga com muita dificuldade, permanecendo apenas com 1 ou 2 infectados simultaneamente enquanto a simulação dura até se extinguir (e isso ainda não

ocorrendo para a grande maioria das populações). Logo no início quase a metade dos experimentos para  $\Gamma_h = 0,0001$  consegue infectar mais elementos na rede a ponto deles conseguirem gerar novos elementos infectados ao longo do tempo. No entanto, o número de experimentos que a partir daí consegue ir se mantendo vai cada vez mais declinando à medida que o tempo passa, chegando ao ponto em que de 100 experimentos apenas 3 conseguem se manter até 150 dias e após isso apenas 1 deles consegue chegar até a marca de 393 dias, o que comprova esse comportamento probabilístico, atingindo 102 pessoas no máximo de um total de 10000 pessoas, o que nos possibilita verificar que, para essa realização, a taxa de espalhamento se manteve em média em torno de 0,26 novas pessoas infectadas a cada dia.

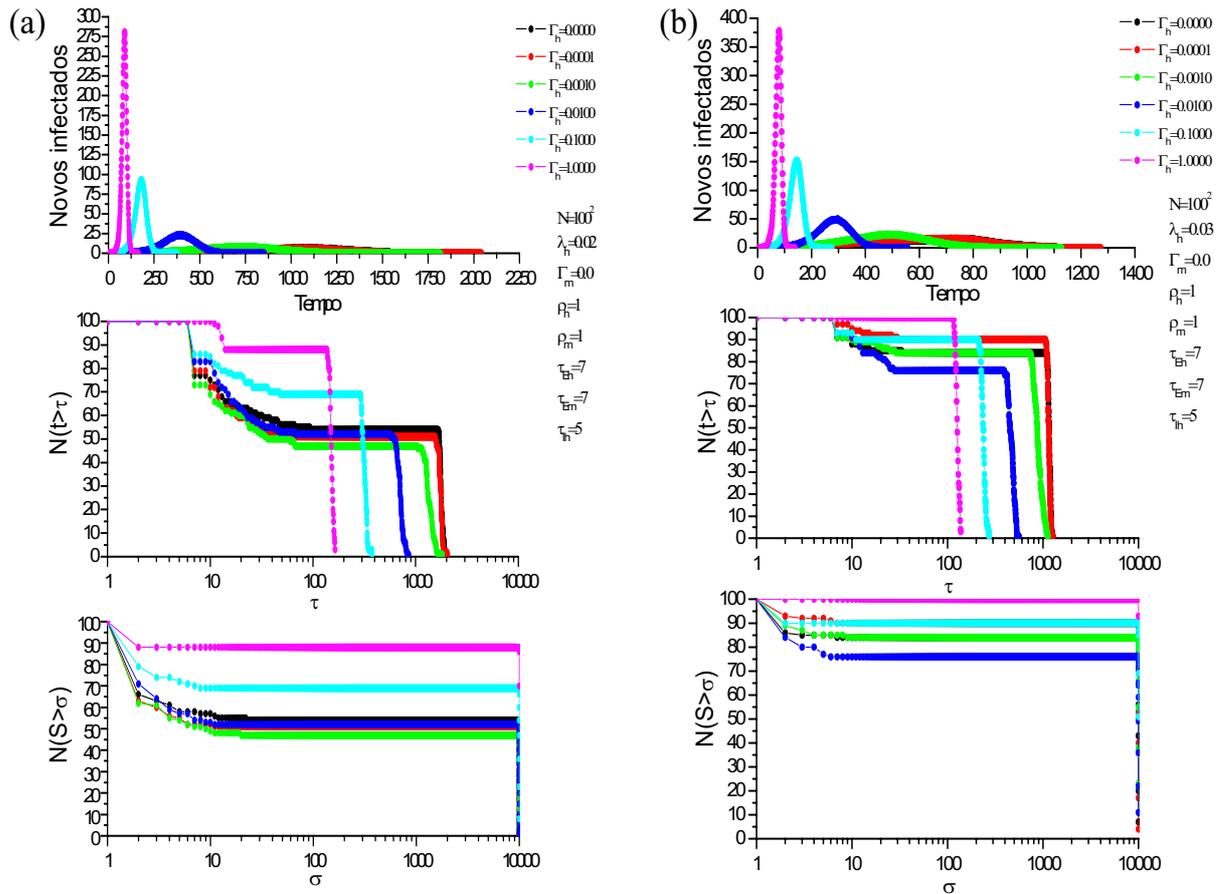
No caso do modelo para  $\lambda = 0,01$  com essa configuração, a probabilidade de que a epidemia se mantenha até atingir a todos da rede aparentemente até existe, mas essa probabilidade é tão pequena que torna mais difícil que isso aconteça e requer ainda mais experimentos para eventualmente um deles consiga atingir a rede por completo. Particularmente, para o cenário representado por essa linha vermelha, mesmo com algum espalhamento global a disseminação ainda apresenta-se ineficiente e ineficaz.

Continuando a analisar os gráficos da Figura 3.4, observa-se que aparentemente para as demais curvas (verde, azul, azul turquesa e rosa, respectivamente) que o sistema se torna mais eficiente à medida que o valor de  $\Gamma_h$  aumenta, visto que o espalhamento passa a ganhar mais força à medida que os contatos globais passam a atuar com mais força no espalhamento, tornando-se mais eficazes no que diz respeito ao número de pessoas atingidas da rede e eficientes no que diz respeito ao tempo de duração para que isso ocorra. Nota-se que a linha verde na maioria das simulações não passa de 10 indivíduos atingidos para 90% das realizações, contudo dentre os 10% restantes uma consegue se manter e atingir 67,05% da rede.

Nesse caso, o que é interessante é que o espalhamento, mesmo não sendo tão eficiente quanto o das curvas seguintes, consegue se manter por um longo período de 5423 dias estabilizando-se com uma taxa de reprodução média pouco mais de 1 indivíduo infectado na rede a cada  $\tau_I$  dias. Com isso, cerca de 6705 pessoas foram sendo atingidas pelo processo epidêmico dentre o total de 10000 pessoas, o que indica para essa realização uma taxa de espalhamento de 1,23 novas pessoas infectadas a cada dia, o que aparentemente caracteriza um processo endêmico e, de certa, forma bastante similar ao que se observa em alguns anos para a dengue em alguns municípios da Região Metropolitana da Baixada Santista de uma maneira geral como se observa a partir dos dados de casos obtidos do Sistema de Informação de Agravos de Notificação (SINAN) [SINAN, 2007] para algumas cidades como Ribeirão Preto, no interior do estado de São Paulo, no ano de 2004.

A partir da linha azul ainda na figura 3.4 (página 81), para  $\Gamma_h = 0,01$  já é possível observar que existe mais de uma realização que atinge cerca de 95% da população, sendo que o efeito obtido pelo espalhamento global a partir daí notado nas curvas azul turquesa ( $\Gamma_h = 0,1$ ) e rosa ( $\Gamma_h = 1$ ) é o de que as na primeira, 30% das simulações passam a atingir praticamente todas as pessoas da rede ao longo do processo de espalhamento da doença e para a última isso já se verifica para 72% das realizações, ao mesmo tempo em que o tempo necessário para isso decai exponencialmente à medida que o valor de  $\Gamma_h$  é acrescido em escala logarítmica mesmo para esse cenário particular do modelo.

A seguir, estará sendo observado como o sistema se comporta dentro da faixa de valores adotada para  $\lambda$  valores maiores. Mais uma vez vale lembrar que se espera aqui entender melhor como se apresenta a transição de um sistema endêmico para um sistema epidêmico, uma vez que essa transição é de muito interesse para a previsão de uma epidemia dentro do contexto deste trabalho. Nesse sentido, a Figura 3.5 apresenta alguns resultados muito interessantes:



**Figura 3.5** – Simulações para  $\lambda_h = 0,02$ , em (a) e  $\lambda_h = 0,03$  em (b). Aqui os dois resultados são dispostos ao mesmo tempo para efeito de comparação.

Ao observarmos a figura 3.5 (a), a primeira coisa que nos chama atenção é que agora, para todas as configurações de  $\Gamma_h$  em mais da metade das realizações, o espalhamento consegue atingir praticamente todos da rede, sobrando apenas uma pequena porcentagem para alguns valores menores de  $\Gamma_h$ , como se pode ver no gráfico inferior em (a).

Além disso, pode-se observar agora que o tempo que cada configuração leva para completar o espalhamento pela maioria da rede tende a seguir o comportamento esperado de acordo com o estudo de efeito de mundo pequeno apresentado no capítulo anterior, ou seja, à medida que a mobilidade dos humanos aumenta, menor é o tempo requerido para que o espalhamento se complete. De certa forma, a configuração mais demorada levou cerca de 1100 dias para atingir 10000 pessoas, o que dá uma taxa de espalhamento médio de 9 novas pessoas infectadas por dia, sendo o espalhamento nesse caso muito mais forte do que o

apresentado na Figura 3.4. O fato é que aqui mais uma vez, tanto para (a) quanto para (b) o espalhamento adquire um aumento de velocidade exponencial à medida que a mobilidade é apresentada de maneira logarítmica, o que indica possivelmente a ocorrência do fenômeno de mundo pequeno aqui e conseqüentemente a possibilidade de se compreender melhor como a mobilidade das interações entre as pessoas pode afetar significativamente no processo de transmissão da dengue.

O que é notável, e inesperado, é a maneira como as curvas se apresentam para os dois últimos gráficos de (a) e (b):  $N(t > \tau)$  e  $N(S > \sigma)$ . É possível perceber que o número de realizações que conseguiram se desenvolverem apresenta uma inversão quanto ao que seria o comportamento esperado. De fato, esperava-se aqui que à medida que o valor de  $\Gamma_h$  fosse acrescido, o número de realizações deveria se tornar proporcionalmente maior, ou seja, as linhas deveriam crescer obedecendo a seguinte ordem: preta, vermelha, verde, azul, azul turquesa e rosa.

Ocorre, no entanto, que em (a), a grande maioria das curvas se apresenta numa faixa, variando entre 47 a 54 experimentos, mas tanto a vermelha quanto a preta se apresentam acima da linha verde, o que nos leva a concluir que aparentemente aqui a mobilidade inicialmente acaba atrapalhando de certa forma a dinâmica do espalhamento para a partir de um valor de  $\Gamma_h$  passar a atuar com maior força. Em (b) o mesmo ocorre, no entanto para a linha azul, e a diferença agora entre o número de realizações se mostra maior não obstante o comportamento do sistema se apresentar como esperado dado nosso estudo anterior.

Contudo, apesar desta inversão, frente aos resultados observados pode-se pensar que a diferença aí se apresenta justamente devido à característica probabilística do sistema para o cenário em estudo e não necessariamente poderia ser tomada como uma regra, uma vez que o comportamento médio do sistema presente no primeiro gráfico em (a) e (b) se

mantém exatamente como previsto. É provável nesse caso que talvez sejam necessárias mais realizações para que o sistema entre em equilíbrio, mas de maneira geral isso não afeta de maneira significativa o comportamento do sistema, sendo apenas um reflexo da própria zona de transição em que se encontram os parâmetros adotados para tais experimentos.

Ainda nos gráficos apresentados na Figura 3.5, existem algumas tendências que são interessantes ressaltar e que pretendemos analisar nos próximos gráficos aqui apresentados. A primeira tendência refere-se ao número de realizações que são bem sucedidas no espalhamento. Observa-se ao comparar (a) e (b) da mesma figura que à medida que o valor de  $\lambda$  é acrescido, mais simulações conseguem se desenvolver, o que pode ser notado nos dois gráficos inferiores de (a) e (b). Isso quer dizer que a probabilidade de transmissão se tornando maior, mais eficaz o espalhamento se torna e conseqüentemente, mais factível o espalhamento se torna.

Uma vez que não é efetuado um controle sobre os novos casos que se apresentam, a tendência do espalhamento nesse caso é se expandir a partir do indivíduo inicial até as bordas da rede, como uma onda que cresce à medida que o tempo passa (para o espalhamento somente local) ou várias ondas que tornam tal espalhamento concorrente (como ocorre quando os espalhamentos locais e globais coexistem na configuração do modelo). Com isso, a transição entre um sistema endêmico (com poucos casos e constante) para um sistema epidêmico é facilmente observada dentro dessa faixa de valores adotados para  $\lambda$ .

São várias as análises possíveis ainda somente para essas duas figuras apresentadas até agora na presente discussão. É interessante observar a evolução do espalhamento para  $\Gamma_h = 0,0001$  que se apresenta agora com muito mais força e juntamente com  $\Gamma_h = 0$  passaram a ser as simulações mais demoradas que foram anotadas

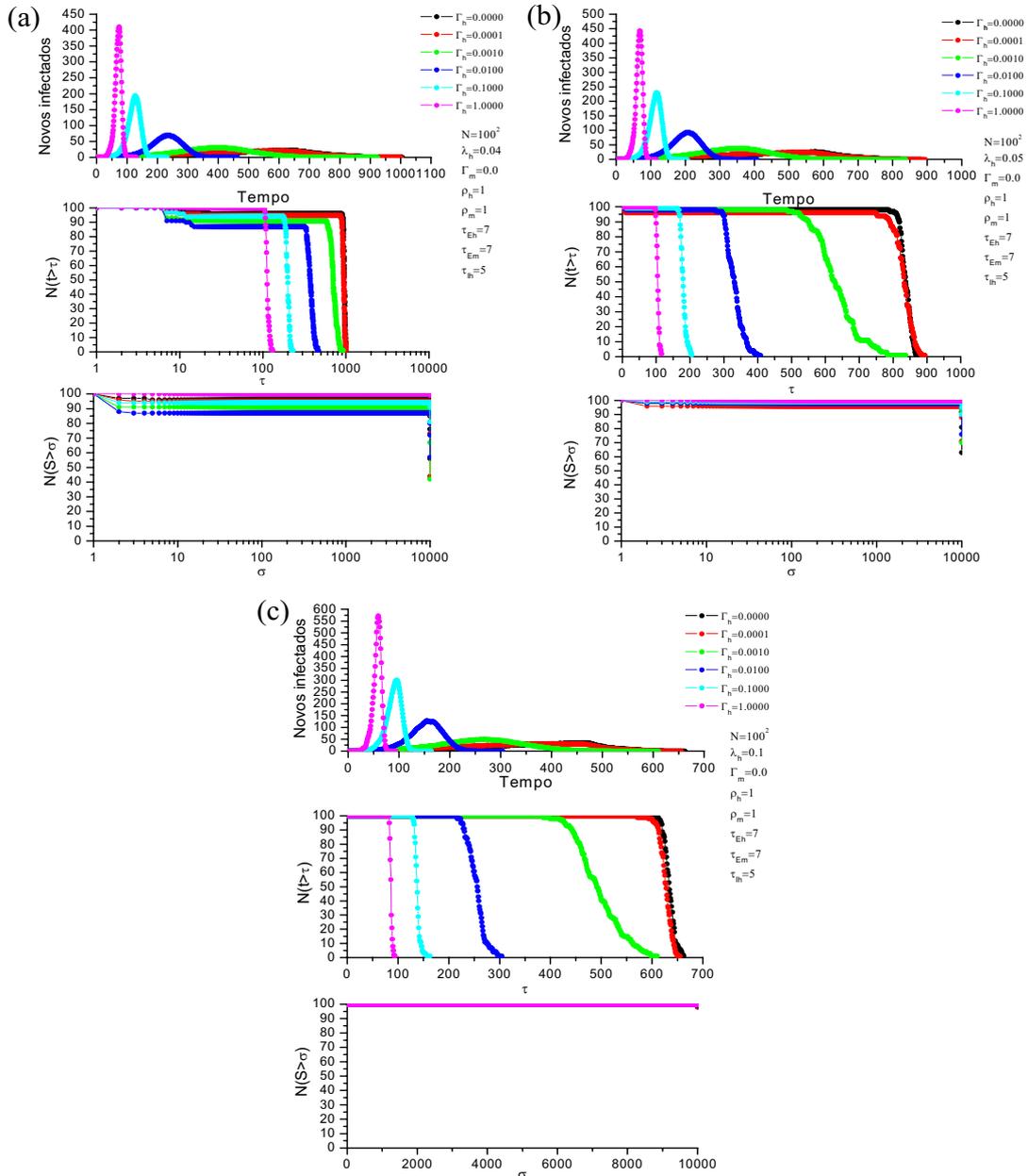
ao crescer o valor de  $\lambda$ . Isso nos leva a concluir que deve haver uma transição entre esses dois cenários apresentados para  $\lambda = 0,01$  e  $\lambda = 0,02$  que faz com que o sistema se torne mais eficaz já para uma situação onde a mobilidade se apresenta muito baixa na rede apenas alterando o valor de uma única variável  $\lambda$ . É um tanto quanto óbvio de certa forma que se aumentando a probabilidade de que o espalhamento ocorra, o mesmo se torne mais forte.

Contudo, o que é interessante aqui é que há uma transição entre um sistema onde a epidemia simplesmente se desenvolvia com muita dificuldade apenas para algumas realizações para um sistema onde mais da metade dos experimentos obtém sucesso em atingir a grande maioria da rede. Compreender como tal linha tênue existente entre esses dois cenários se comporta é algo de grande interesse no presente estudo, porque assim futuras medidas de controle podem ser inseridas a fim de reduzir a força do espalhamento até que esse limite seja rompido e então a epidemia extinta para tal configuração atual.

Para efeito de comparação a Figura 3.6 foi organizada de forma a apresentar como essa tendência discutida anteriormente se torna cada vez mais real à medida que o parâmetro  $\lambda$  é acrescido. Aqui, de (a) para (b), nota-se que a maioria dos experimentos agora se mostra com força suficiente para atingir toda a população com maior eficácia, caracterizando-se cada vez mais como uma epidemia ao invés de um processo endêmico de espalhamento.

Os gráficos em (c) foram inseridos à figura para que se possa estabelecer uma comparação com o comportamento epidêmico apresentado para um valor fora da faixa de parâmetros adotados para  $\lambda$ , e como o sistema em (a) e (b) tende para tal cenário, já se mostrando (b) bastante próximo de (c) quanto ao tempo e número de experimentos que conseguem atingir a maioria da rede, aqui colocados em escala de dias para facilitar a

comparação. Assim, já é possível observar em (c) que para  $\lambda = 0,1$  é notável que com o aumento da mobilidade da rede o sistema passa a se apresentar mais rápido e contagioso.



**Figura 3.6** – Simulações para  $\lambda_h = 0,04$ , em (a),  $\lambda_h = 0,05$  em (b) e  $\lambda_h = 0,1$  em (c).

Observando mais uma vez o dano causado para os três cenários apresentados em (a), (b) e (c), nota-se que o efeito de mundo pequeno [GAGLIARDI, ALVES, 2005] é catalisado à medida que a probabilidade de novos contatos globais entre o homem e o mosquito é acrescida. Isso é muito interessante pelo seguinte fato: grande parte do contágio ocorre não apenas na residência do indivíduo, mas em seu local de trabalho que pode se situar muitas

vezes próximo a regiões de risco de criação de mosquitos tais como cemitérios, áreas verdes da cidade, quartéis da polícia, lixões, lagos, dentre outros vários pontos estratégicos e imóveis especiais que possibilitam a criação do *Aedes aegypti* e por isso, quanto mais um indivíduo tenha acesso a locais como estes em seu cotidiano, maior a probabilidade de algum mosquito que ali se encontra adquirir a doença e assim transmiti-la a outra pessoa que ali se encontra.

Nesse caso, o papel da mobilidade do ser humano dentro de uma população é um fator de grande impacto para que o espalhamento ocorra e o que se observa nesses resultados é justamente um reflexo desse cenário, no qual o ser humano pode entrar em contato com várias pessoas ao longo do dia, não necessariamente conhecidas (pensando em um serviço de atendimento, por exemplo), e os mosquitos que podem ali existir piquem os clientes, transmitindo assim a doença para os atendentes.

Desta forma, a rede de mundo pequeno dos humanos fica centrada na figura do mosquito, que atua como um intermediário no processo de infecção para outros seres humanos que não necessariamente precisam ser amigos, mas sim que possuem algum grau de ligação baseado em locais em comum que freqüentam. Da mesma forma, o raciocínio inverso também é válido para a rede de mundo pequeno de mosquitos, uma vez que ela seria então centrada na figura do humano de quem contraem a doença.

Nesse caso, a densidade de mosquitos torna-se um fator chave para essa transmissão, mas o interessante é que o modelo já se mostra capaz de apresentar resultados condizentes com esse tipo de situação e também pode ser ajustado para se adequar a cenários mais particulares relacionados à realidade de cada município à medida que tais dados sobre densidade forem sendo aferidos.

### 3.3 O MODELO DA DENGUE EM CIDADES VIRTUAIS

Algumas considerações são muito importantes de serem discutidas nesta seção para que o modelo da dengue se adapte a realidade de cada município, aproveitando para comentar um dos recursos mais importantes do sistema que apresentaremos no próximo capítulo.

Para que o modelo possa prever uma epidemia num município é necessário que este também seja modelado e que a sua população (com as regras de interação descritas neste capítulo, para a dengue) seja, portanto, a população daquela cidade. Contudo, a tarefa de se modelar uma “cidade virtual” nem sempre é algo fácil e exige um estudo bem mais aprofundado sobre quais características são mais pertinentes de serem consideradas em detrimento de outras para a realidade de cada doença.

Nesse sentido, características sócio-econômicas da população, relevo, vegetação, clima, pontos estratégicos tal como praças, lagos, piscinas e regiões onde há maior concentração de mosquitos e pessoas são pontos que deveriam ser levados em consideração. Com isso, probabilidades diferentes poderiam ser aferidas a partir do cruzamento de tais informações, calculadas a partir de indicadores de risco [MUCHA et al, 2005; RIBERTO et al, 2005]. Contudo, isso seria um outro estudo e até então no prezado trabalho estamos tentando criar o que seria uma primeira aproximação estrutural de uma cidade virtual para ser aplicada à realidade da dengue.

Diante disso, para esta primeira versão do modelo foi criado um simulador capaz de ler um arquivo de definições de uma cidade qualquer e configurar o espaço das redes do modelo da dengue de acordo com a geometria desta cidade. A maneira como este arquivo é gerado e como se estrutura será apresentada em maiores detalhes no próximo capítulo e no apêndice A deste trabalho.

No modelo da dengue, cada indivíduo é capaz de armazenar probabilidades diferenciadas para cada posição em que ocupa, tomando como base o endereço de residência daquela pessoa. Apesar disso, como comentado anteriormente, até a conclusão deste trabalho

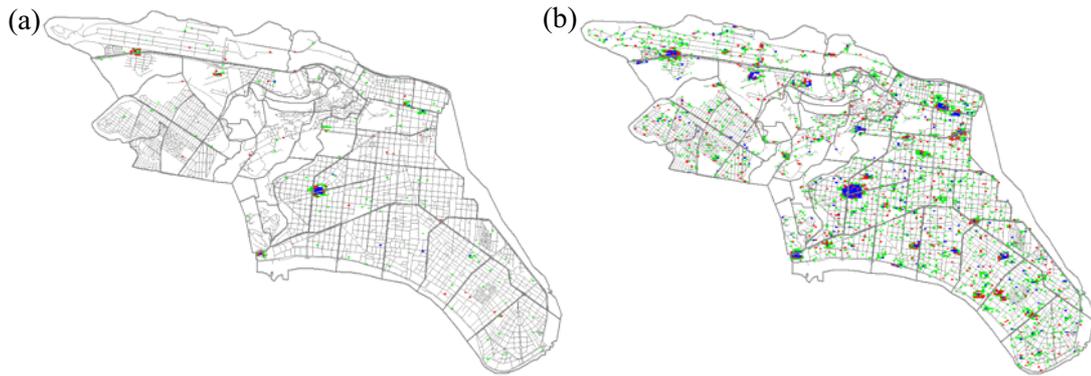
não havia um estudo ainda finalizado sobre a delimitação destas regiões para as cidades disponibilizadas para simulação apresentadas no estudo de caso do capítulo 5 e, portanto, esta susceptibilidade associada a cada indivíduo foi estimada de maneira arbitrária para testes.



**Figura 3.7** – Esquema de uma rede baseada na geometria da cidade de Santos-SP, na Região Metropolitana da Baixada Santista. Em (a) é apresentado o mapa de ruas e bairros da cidade. Em (b) pode se observar a forma da rede (em preto) obtida baseada nas ruas na cidade a partir do pressuposto de que onde há ruas existem casas ou locais onde a infecção pode acontecer que será usada para dar a forma final da rede, agora vista em (c) na qual as áreas em verde representam no mapa onde existem pessoas para que o espalhamento ocorra. Nesse caso, a figura (c) mostra a rede (em verde) com o mapa da cidade contendo os limites de bairros e ruas sobrepostos a ela.

Particularmente, o modelo da dengue possibilita a inserção de sítios vazios dentro do autômato. Isso torna possível a modelagem de regiões com mais ou menos mosquitos, verificando-se isso também para a população de humanos.

Assim, é possível definir uma rede cuja geometria não é mais quadrada como nos estudos anteriores, mas pode se assemelhar à de uma cidade real. Com isso, podemos definir vários níveis de susceptibilidade à rede, tal como seria considerado ao inserir ao modelo informações sobre regiões onde vivem pessoas com mais baixo nível de instrução, locais onde existem municípios mais populosos, pontos estratégicos da cidade, regiões onde existe um maior histórico de casos e então com isso tornar mais rica a descrição do *mundo* a ser simulado.



**Figura 3.8** – Uma simulação com uma rede usando a geometria do município de Santos-SP. Aqui os pontos em verde são humanos expostos, os em vermelho são humanos infectados e em azul os humanos recuperados. Em (a) temos que apenas poucos casos da dengue após poucos dias em que o primeiro caso ocorreu. Em (b) temos aproximadamente três meses desde o primeiro caso sem o controle no município.

Contudo, é extremamente importante notar que mesmo sob uma descrição extremamente detalhada da cidade que se quer observar ou prever como a doença pode vir a se comportar reside um modelo simples, cuja idéia não difere muito daquela apresentada anteriormente no capítulo 2 deste trabalho, sendo, portanto essa característica um dos pontos-chave deste trabalho: a simplicidade do modelo. Tal simplicidade é o que torna possível sua aplicação para diversos cenários que se desejar, bastando para isso apenas configurar a rede de maneira a refletir a realidade da cidade que se pretende estudar.

### 3.4 CONCLUSÕES E COMENTÁRIOS GERAIS

Diante do apresentado durante este capítulo, um ponto importante de se reconhecer no modelo é o fato dele apresentar regras probabilísticas, o que proporciona um ganho computacional muito maior do que comparado com abordagens fortemente ancoradas na utilização de grafos para a descrição dos contatos entre os humanos [HUANG et al, 2005]. Aqui, como na vida, as iterações entre as pessoas ocorrem de maneira *ad hoc*, ou seja, uma pessoa a cada instante pode criar novas amizades ou mesmo se afastar delas, como na

vida real e isso confere a essa forma de modelar uma característica única na literatura, além de poder ser aplicado para outros tipos de doenças, tais como a doença de chagas ou a leishmaniose, por exemplo, pois a despeito da heterogeneidade das interações proporcionada por tais relações, basta simplesmente ajustar os parâmetros do modelo para aquela realidade.

A despeito desta nova maneira de se modelar a dengue, ainda se faz necessária uma validação mais cuidadosa, que seria o próximo passo que se pretende seguir a partir deste trabalho com o intuito de tornar este modelo uma ferramenta capaz de fornecer cenários para a tomada de decisão por parte dos agentes de saúde.

Além disso, a grande vantagem desse modelo é o ganho de desempenho computacional que ele proporciona frente a alguns sistemas para a previsão de doenças [EUBANK, 2004] que fazem uso intensivo de poderosos *mainframes* para sua simulação. Com a simples implementação deste modelo apresentada no Apêndice A deste trabalho, é possível executar simulações em um computador *desktop* simples e obter resultados em um tempo relativamente rápido, além de poder ser facilmente adaptado a um ambiente de grades computacionais, economizando assim tempo e dinheiro.

Diante disso, toda esta discussão serviu para que pudéssemos montar o cenário apropriado para o sistema que construímos ao longo deste trabalho, e que estará sendo apresentado a partir de agora.

## **CAPÍTULO 4 – O SISTEMA DE PREVISÃO DE EPIDEMIAS INTEGRAMODEL**

Nos capítulos anteriores conhecemos e estudamos os vários conceitos, técnicas e fundamentos computacionais necessários para iniciar então a descrição da aplicação que foi desenvolvida ao longo deste trabalho.

De fato, nos dias atuais a dengue se apresenta como uma doença de grande impacto social e prever seu comportamento é algo de grande interesse por parte da sociedade em geral, residindo aí a relevância da aplicação que será apresentada neste capítulo. Ocorre, porém que no Brasil muitas cidades ainda usam serviços de informações muito precários para analisar e armazenar informações relevantes à epidemias.

Em consequência da falta de recursos existentes, o tempo entre a coleta de informações (geralmente ainda usando formulários de papel) e a sua inserção em um sistema gerenciador de banco de dados é por vezes muito grande. Além disso, os sistemas existentes não são integrados com outras bases de dados municipais. Assim, há uma grande perda de qualidade da informação, demandando maior esforço para que análises adequadas possam ser realizadas com maior agilidade utilizando tais bases de dados.

Por vezes, o monitoramento de doenças infecto-contagiosas como a dengue é feito em um mapa de papel no qual os casos são marcados com alfinetes colocados manualmente um a um sobre os endereços de residência dos casos suspeitos, sendo pintados de outra cor quando confirmados, o que se mostra também muito precário, ainda mais por demandar que o gestor de saúde responsável consiga apresentar alguns resultados positivos diante de tais recursos funcionais.

Tal situação não possibilita análises mais elaboradas sobre tais dados, dificultando sua conseguinte manipulação e estudo, contribuindo dessa forma para a lentidão e falta de melhores estratégias na tomada de decisões como contramedidas ao espalhamento de tais doenças. Por muitas vezes o cadastro e monitoramento de tais ocorrências se trata ainda de um processo inteiramente manual nas secretarias municipais de saúde e a projeção dos impactos que os casos apresentados pode atingir na população ainda se apresenta de maneira muito abstrata e sem uma visualização eficiente dos cenários que a doença pode apresentar no futuro.

Ainda que campanhas de prevenção de doenças sejam recorrentes a cada vez que começam a aparecer novos casos de uma determinada doença tal como a dengue, a dificuldade encontrada por agentes epidemiológicos para entender como o processo epidêmico está se manifestando em sua cidade, prever os futuros locais mais críticos que a doença pode atingir a partir dali e conseqüentemente agir conforme o necessário ainda se mostra presente. Certamente um sistema computacional elaborado para isso não resolveria todos os problemas epidemiológicos existentes e muito menos tornaria tão fácil a vida dos gestores de saúde, porque muitas vezes é a higiene os hábitos da população que contribuem para uma epidemia acontecer ou não, mas a idéia que se quer apresentar aqui é a de que um sistema de computação pode ajudar na tomada de decisões, não que seja indispensável, mas que proveria ferramentas para tornar mais rápida a ação dos gestores se utilizado. É justamente neste cenário em que nosso projeto se insere.

#### 4.1 O PROJETO INTEGRAEPI

O IntegraEPI [SILVA et al, 2007] é um projeto em conjunto entre os programas de pós-graduação em Informática e pós-graduação em Saúde Coletiva da Universidade Católica

de Santos e atualmente abrange outras instituições como a Unifesp, Faculdades COC de Ribeirão Preto, Secretaria Municipal de Santos, Secretaria Municipal de Saúde de São Paulo e a Superintendência de Controle de Epidemias (SUCEN).

O principal objetivo do projeto IntegraEPI é contribuir para a modernização do sistema de vigilância epidemiológica a partir da construção de um sistema de monitoramento e previsão de doenças para ser aplicado como uma ferramenta de suporte a decisão na área da saúde pública [ALVES, GAGLIARDI, SILVA 2006].

Dentro deste contexto é que se insere o presente trabalho, por se tratar de uma das partes constituintes do sistema IntegraEPI: o módulo de previsão de doenças IntegraModel. Para isso, um ambiente de grade computacional foi escolhido para ser o repositório dos resultados tecnológicos, algoritmos e produtos de software intermediários, que serão construídos à medida que este projeto vier sendo desenvolvido.

Visando a melhoria da qualidade da informação na saúde pública, o sistema IntegraEPI deverá ser capaz de oferecer recursos para a comparação dos resultados das simulações detalhadas (como as discutidas no capítulo 3 deste trabalho) com o padrão experimental observado para o espalhamento de uma epidemia real, tanto temporal como geográfico, abstraindo certos graus de complexidade envolvidos no processo de análise aos usuários do sistema.

Para isso, este sistema deverá possibilitar a integração entre várias bases de dados referentes a dados populacionais e dados de notificação de casos dessas doenças de maneira transparente e clara.

Como comentado anteriormente o Globus Toolkit apresenta vários mecanismos atualmente que se mostram de grande interesse para esta aplicação, tanto na questão de possibilidade de tolerância a falhas, quanto em relação à capacidade de integrar dados geograficamente distribuídos de maneira transparente e distribuída, da mesma forma em que

oferece mecanismos de execução que dão suporte às simulações que são necessárias para a previsão de epidemias, sendo esta a razão que levou à escolha deste *toolkit*.

É certo que existem outras opções de *toolkit* ou mesmo outros escalonadores que seriam capazes de executar tais simulações, tendo sido alguns deles avaliados durante o processo de estudos do sistema. Contudo, o *Globus Toolkit* é capaz de abranger tais tecnologias como serviços grid baseados no padrão OGSA, sendo assim uma escolha pertinente à nossa aplicação.

A partir da integração desses modelos de previsão com os dados epidemiológicos, mapas incorporando tecnologias SIG [CAVICCHIOLI NETO, RIBERTO, ALVES, 2003; FELIPE et al, 2004] serão disponibilizados, com toda a infra-estrutura urbana e características ambientais a partir de dados públicos de cada município [ALVES, PESSOA, GAGLIARDI, 2004] para que os padrões do espalhamento da doença possam ser observados. Como consequência de tal estudo esperamos elucidar mecanismos de contágio em clusters epidemiológicos, reconstruindo no tempo e no espaço a história da transmissão e fazendo previsões sobre seu progresso futuro.

Apesar do estágio atual da aplicação conter apenas o modelo da dengue, que ainda está em desenvolvimento, será possível a adição de modelos para as mais diversas doenças contagiosas tais como a tuberculose, hanseníase, hepatite, leptospirose e meningite como recursos da grade, uma vez que o serviço será capaz de prover o substrato necessário para sua execução. Estas doenças são muito comuns na Região Metropolitana da Baixada Santista, apresentando-se tanto como doenças endêmicas quanto epidêmicas.

Diante deste contexto, será apresentado a seguir a arquitetura do sistema IntegraEPI e seus principais componentes, com ênfase no serviço de simulação de epidemias.

#### **4.1.1 A arquitetura do sistema IntegraEPI**

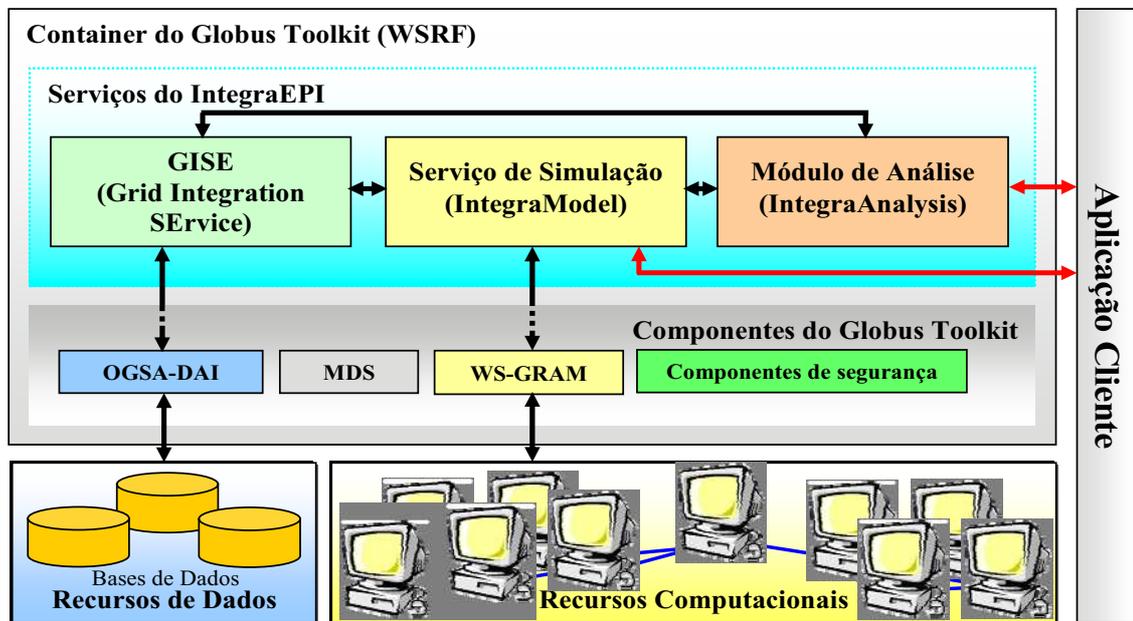
A arquitetura do sistema IntegraEPI é organizada como apresentado na figura 4.1. Em sua estrutura merecem destaque três serviços (colocados em destaque na figura): o serviço de dados e integração no grid GISE (Grid Integration Service) [GALLO, 2007; CAVICCHIOLI, GAGLIARDI et al, 2006], o serviço de simulação IntegraModel, e o serviço de Análise IntegraAnalysis [REZENDE, GAGLIARDI et al, 2006].

Cada um dos referidos serviços apresenta características muito importantes para a composição do sistema que se pretende elaborar, porque devem ser capazes de prover a integração de dados populacionais (e de casos notificados de uma doença em particular que se encontram geograficamente distribuídos de maneira transparente), um serviço de simulação dos modelos epidêmicos (criados a partir das regras discutidas nos capítulos 2 e 3 capazes de serem parametrizados a partir de dados reais providos pelo sistema GISE) e serviços que provêm mecanismos de análise de doenças (cujos dados podem ser obtidos a partir do serviço de integração de dados).

No momento da escrita desta dissertação, os serviços de análise ainda se encontram numa fase muito inicial de desenvolvimento e constituem outros trabalhos distintos dentro do escopo do projeto, contudo o serviço de integração já se encontra em um estágio mais avançado, necessitando, contudo, de alguns ajustes. O módulo de simulação se encontra em estágio de desenvolvimento e já apresenta algumas funcionalidades, mas como os demais módulos ainda não estão totalmente implementados, não foi possível ainda testá-las interagindo com os demais dois serviços.

Como se observa, os três serviços do sistema IntegraEPI deverão ser capazes de interagir entre si à medida que se encontrarem em um estágio mais avançado de implementação, visto que os dados providos pelo serviço de integração servirão como entrada

dos módulos de análise e simulação ao mesmo tempo em que o módulo de simulação poderia utilizar os serviços de análise para que cenários alarmantes do espalhamento pudessem ser detectados à medida que suas simulações forem sendo requeridas.



**Figura 4.1** –.A arquitetura do sistema IntegraEPI apresentada de maneira simplificada [CAVICCHIOLI, GAGLIARDI et al, 2006].

Após esta breve descrição dos componentes que constituirão o sistema IntegraEPI, estaremos detalhando discutindo cada um deles a seguir.

#### 4.1.1.1 O serviço de acesso e integração de dados

Para que os módulos de simulação e análise possam ser devidamente parametrizados, dados relativos à doença que se pretende estudar, bem como da cidade em que ela ocorre devem ser disponibilizados e conseqüentemente mecanismos de acesso se fazem necessários para isso. Da mesma forma, o volume e diversidade das bases de dados para essa tarefa pode ser significativa, dificultando assim a manipulação desses dados por precisariam ser interpretados e compatibilizados com o escopo da aplicação.

Nesse sentido, a integração desses dados se apresenta como uma solução interessante para esse cenário. Apresentado recentemente, o serviço de integração de dados [GALLO et al, 2006] apresenta uma arquitetura centrada na figura de um mediador capaz de manipular requisições de consultas distribuídas e retornar o resultado obtido das várias bases de dados como um único conjunto de dados.

Assim, no que diz respeito aos serviços de simulação e análise, o sistema IntegraEPI necessita de um mecanismo capaz de acessar tipos como mapas, informações geográficas e dados sócio-econômicos e nesse caso, os dados poderiam estar geograficamente distribuídos em diversos SGBDs, tais como o PostgreSQL, SQL Server e Oracle. A integração de dados atua, portanto na integração de dados de diferentes recursos disponíveis.

No momento este serviço ainda se encontra em fase de desenvolvimento e testes, não estando em uso ainda, portanto, pelos outros módulos do sistema IntegraEPI, inclusive o sistema de simulação que é o foco do presente trabalho, mas o objetivo do serviço GISE é prover os dados necessários para as aplicações deste sistema de maneira transparente. Sua aplicação, contudo, deve considerar uma série de problemas que poderiam surgir diante tais como:

- Grande número de fontes de dados, complicando ainda mais questões relacionadas à integração de esquemas e resolução de conflitos;
- A disponibilidade das fontes de dados varia dinamicamente. Neste caso, a adição e remoção de fontes de dados em uma plataforma de grade computacional devem ser feitas de maneira a causar um impacto mínimo em relação a seu esquema integrado;
- As fontes de dados podem ter diferentes capacidades computacionais, variando de simples arquivos a poderosos SGBDs. Quanto a isso, o OGSA-DAI oferece

vários tradutores (*wrappers*) que conseguem gerenciar os dados de simples arquivos a complexos bancos de dados relacionais ou orientados a XML.

A solução para resolver alguns dos problemas mencionados acima consiste na implementação de um Serviço de Catálogo de Metadados (MCS, *Metadata Catalogue Service*), aqui em nosso sistema implementado como um serviço grid (baseado no padrão OGSA) para garantir interoperabilidade entre os demais serviços que compõem a grade e o IntegraEPI.

Particularmente, decidiu-se por desenvolver os serviços grid de acesso a dados como serviços OGSA-DAI [OGSA-DAI, 2006], incluindo o MCS, devido a sua capacidade de gerenciar diversos SGBDs e prover às aplicações cliente um acesso claro e transparente quando às particularidades envolvidas no processo de acesso e integração de dados.

Outra característica muito interessante que justifica o uso do OGSA-DAI nestes serviços é a escalabilidade por ele proporcionada. É altamente desejado pela nossa aplicação que bases de dados de mapas, ou dados populacionais ou relacionados à saúde possam ser inseridas ou removidas de acordo com a necessidade de maneira que não afetem o comportamento de cada módulo do sistema, minimizando a necessidade de alterações ou atualizações. Conseqüentemente, a partir dos esquemas integrados providos pelo catálogo de metadados (MCS), o mediador sabe dizer se os dados estão disponíveis ou não.

A presença do Catálogo de Metadados no sistema é interessante, porque a partir dele o mediador do sistema de integração GISE é capaz de consultar sobre bases que estão integradas na grade que correspondem ao um assunto específico, detalhes particulares da estrutura de cada base local que compõem tal esquema integrado e um dicionário de sinônimos.

Quando uma consulta distribuída é enviada pela aplicação cliente (o IntegraModel ou o IntegraAnalysis, por exemplo), o mediador do GISE procura primeiramente o que existe integrado com relação àquele assunto no MCS, e após obter todas as informações pertinentes, fragmenta a *query* original em várias *sub-queries*, que são então submetidas aos demais serviços de acesso a dados do OGSA-DAI disponíveis.

Ao receber os resultados das consultas, o mediador então realiza um processo de desfragmentação das respostas, centralizando-as em uma única que é devolvida então à aplicação cliente que solicitou aquela consulta.

Portanto, quando o módulo de acesso a dados estiver concluído, ele apresentará uma das funcionalidades cruciais para o funcionamento dos demais componentes do IntegraEPI, tal como o módulo de análise, que estará sendo discutido a partir de agora.

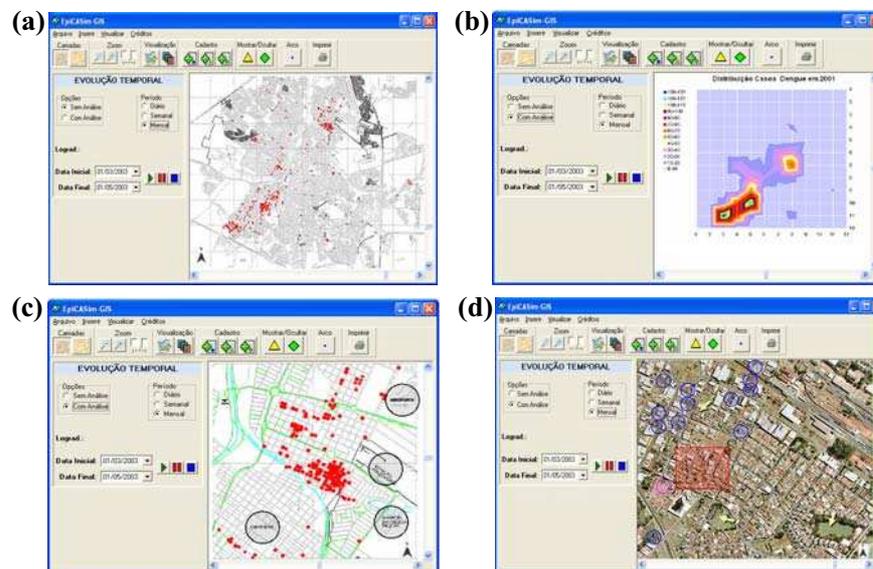
#### 4.1.1.2 Os serviços de análise IntegraAnalysis

O módulo de análise IntegraAnalysis ainda se encontra na fase inicial de desenvolvimento e a sua motivação é a de prover serviços de análise dos dados para com isso gerar indicadores capazes de determinar o risco de alguma doença se tornar uma epidemia e assim atuar como uma ferramenta para a geração de alarmes epidemiológicos no monitoramento de doenças.

Para ilustrar a idéia de como será o funcionamento do sistema de monitoramento de epidemias utilizando os serviços de análise, está sendo apresentada a figura 4.2.

Dentre os vários métodos que se pretende disponibilizar como serviços grid, destacam-se a detecção de *clusters* por área, técnica conhecida como *quadrat analysis* (ou análise por quadrantes) que consiste em se dividir a área do mapa de uma cidade em vários quadrantes nos quais é contado o número de casos de determinada doença por quadrante e comparados

em seguida a uma distribuição de probabilidade. Outra técnica bastante interessante seria a de análise de séries temporais (autocorrelações e análises espectrais).



**Figura 4.2** – Esquema ilustrativo do sistema de monitoramento de epidemias fazendo uso dos serviços de análise do IntegraEPI. Em sentido horário, podemos ver em (a) os casos da dengue sobre o mapa virtual da cidade de Ribeirão Preto, uma região “quente”, gerada a partir de um serviço de análise em (b), a visualização dos casos em conjunto com os rios e regiões mais propícias ao desenvolvimento da doença em (c) e a análise histórica por região em (d) [REZENDE, GAGLIARDI et al, 2006].

Num segundo momento, estas ferramentas de análise poderão interagir na calibração dos modelos do sistema. Mais ainda, este método poderia ser capaz de oferecer condições para a criação de indicadores ambientais com uma metodologia baseada em *scoring* (placar) para estratificar as áreas nas cidades com diferentes níveis de risco de ocorrência da dengue e sua transmissão.

#### 4.1.1.3 O serviço de simulação IntegraModel

O último módulo constituinte do sistema e que merece especial destaque na discussão dentro do IntegraEPI é o módulo de simulação IntegraModel. Como comentado até então, o serviço de simulação vem sendo construído para agir como um mecanismo de execução de

simulações que servirão para a previsão do comportamento de doenças infecto-contagiosas tendo como aplicações modelos epidêmicos como o apresentado no capítulo 3.

A despeito do funcionamento, estrutura e arquitetura deste módulo, foram desenvolvidos dois recursos principais: o serviço de simulação IntegraModel e um cliente capaz de submeter simulações ao serviço, recuperar seus resultados e conseqüentemente fornecer sua visualização. Assim, a seguir estaremos apresentando melhor a arquitetura e execução de cada um desses componentes.

#### 4.2 A ARQUITETURA DO SERVIÇO DE SIMULAÇÃO INTEGRAMODEL

Toda a implementação do serviço foi feita em dois estágios: a construção e disponibilização de um simulador do modelo da dengue, desenvolvido na linguagem C [MIZRAHI 1994a; MIZRAHI 1994b] e que consiste num programa fortemente estruturado, porém, parametrizável e configurável por meio de uma interface de comandos, e o serviço de simulação em questão, capaz de receber as requisições de tarefas das aplicações cliente, desenvolvido inteiramente em Java.

Portanto, o serviço aqui apresentado é constituído por duas partes: o(s) simulador(es) e o serviço de simulação IntegraModel. No momento, o sistema IntegraEPI possui apenas o simulador para o modelo da dengue apresentado neste trabalho. Da mesma forma, o serviço de simulação oferece métodos de submissão de simulações deste simulador apenas.

O serviço IntegraModel foi modelado como esquematizado no diagrama de classes apresentado na figura 4.3. Como se pode observar em sua constituição, o serviço faz uso de outras classes tais como a classe ClientGridFTP e a classe ClientOgsadai, que possibilitam a transferência de arquivos e o acesso transparente às bases de dados do sistema, respectivamente.

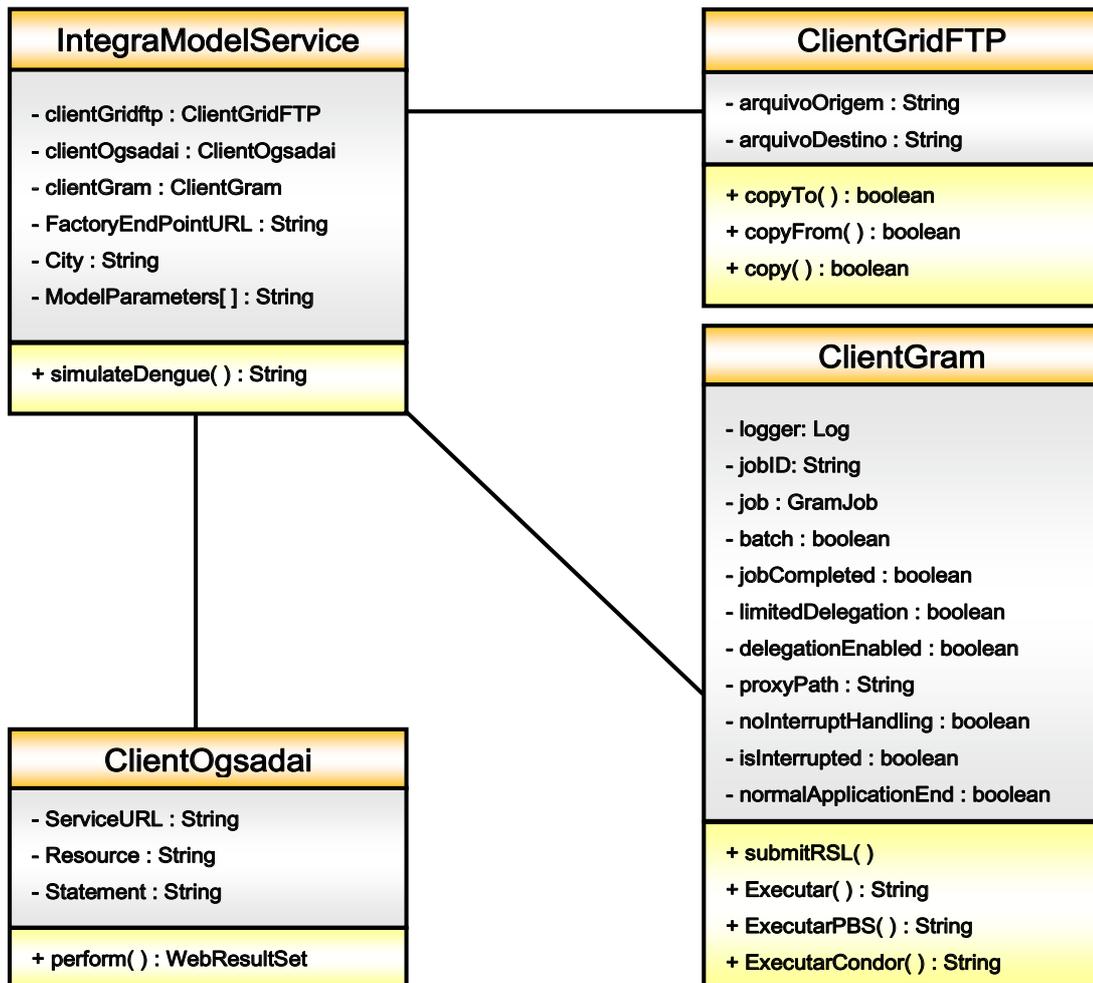


Figura 4.3 – O diagrama de classes do serviço IntegraModel.

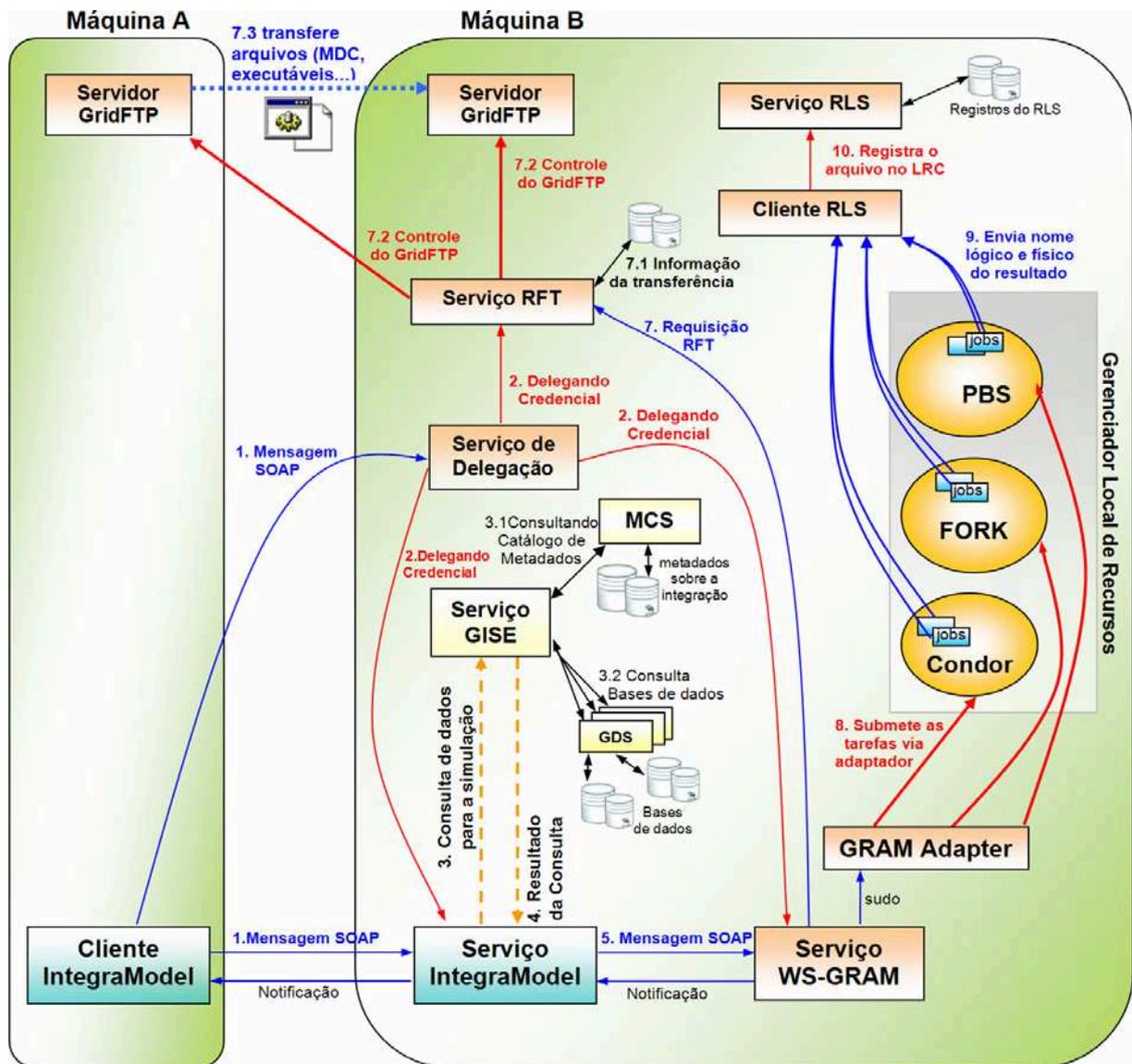
Como discutido anteriormente, no momento o sistema GISE ainda se encontra em fase de desenvolvimento e testes e por isso, o serviço IntegraModel está fazendo uso temporariamente de uma das classes básicas do GISE que é a responsável por interagir com os *grid data services* do OGSA-DAI que o IntegraEPI disponibiliza. O método *perform* particularmente é capaz de executar tanto consultas quanto inserções e atualizações nos dados presentes nas tabelas de maneira transparente ao usuário, bastando apenas para isso passar como argumentos do método a URL do serviço (e.g. “https://serverpos.unisantos.edu.br:8443/wsrf/services/IntegraEPI/dengueData”), o nome do recurso a ser acessado (e.g. dengueData) e o comando SQL desejado (e.g. “select \* from mapas”). Com a conclusão do sistema GISE, haverá uma substituição desta classe para a classe ClientGISE, que deverá ser capaz de interagir com o novo serviço.

A classe ClientGridFTP serve para transferir os arquivos de mapas e outros necessários para a configuração da entrada do simulador pelo serviço IntegraModel e disponibiliza três métodos básicos: CopyTo, CopyFrom e Copy. O método CopyTo é utilizado quando se deseja copiar um arquivo se encontra em outra máquina para a máquina onde o serviço está executando, ao passo que o método CopyFrom já executa o inverso, ou seja, copia um arquivo da máquina em que o serviço está para outra qualquer. O método Copy é justamente a versão mais geral dos dois métodos anteriores, possibilitando transferências terceirizadas, como as apresentadas na seção 1.5.6.1.

Particularmente, este serviço de simulação atua como um *wrapper* entre a grade e a aplicação desejada, por instanciar em seu conteúdo um cliente do WS-GRAM do globus (a classe ClientGram), e devido à modularidade que isso proporciona, seria possível que novos programas ou simuladores fossem adicionados ao sistema, escritos em qualquer linguagem de programação tal como Fortran, C, Java, Cobol, Pascal ou mesmo Java, desde que o serviço consiga interagir com ele a partir de uma interface de comandos que o simulador deve disponibilizar. Desta forma, outras soluções poderão também ser adicionadas ao sistema para complementar ou auxiliar o processo de previsão da(s) doença(s). Conseqüentemente, não apenas o modelo que foi apresentado ao longo deste trabalho mas muitos outros, mesmo baseados em equações diferenciais ou em outras técnicas de modelagem, poderiam entrar como recursos do serviço de simulação do sistema IntegraEPI. Portanto, o serviço IntegraModel se apresenta como um mecanismo escalável para a execução de simulações de doenças na grade computacional.

Para fazer uso do serviço IntegraModel o cliente deverá informar três parâmetros ao método que desejar, sendo eles: a máquina onde se pretende submeter a tarefa (que pode ser tanto a máquina que hospeda o serviço em si ou um *front end*), a cidade em que se deseja

simular e os parâmetros do modelo desejado, respectivamente. A forma como ocorre a execução do serviço IntegraModel é esquemática na Figura 4.4.



**Figura 4.4** – A execução do serviço IntegraModel. A linha pontilhada em azul representa a transferência do arquivo executável e o arquivo MDC no processo de *stage in*. As linhas tracejadas entre o serviço GISE e o IntegraModel representam a busca pelas informações do MDC para a criação da “cidade virtual” que será o mundo no qual a simulação será executada.

A primeira coisa que podemos observar na figura 4.4 é a maneira como o cliente do serviço IntegraModel, executando na “Máquina A” interage com o serviço IntegraModel. De fato, o que está ocorrendo aí é justamente a invocação do serviço pelo cliente como descrito anteriormente, passando os três parâmetros que definem a tarefa a ser executada. Também é possível notar uma seta vermelha ligando o cliente ao chamado Serviço de Delegação. O

serviço de Delegação é o responsável por verificar a validade de determinado usuário que se conecta à grade e seu papel será fundamental para a delegação de credenciais entre o cliente e o WS-GRAM, que necessita delas durante o processo de *staging in/out*.

Após o recebimento da mensagem SOAP enviada pelo cliente invocando-o, o serviço então processa a mensagem e passa para o próximo estágio da execução que consiste em buscar dados para a configuração da rede em que a simulação deverá rodar. A idéia básica aqui é que cada simulação deverá ser executada em um substrato capaz de representar a realidade de um dado município como se fosse uma “cidade virtual” e para isso, o serviço precisa obter os dados necessários para construir a entrada que irá descrever a configuração da rede para aquele cenário.

Neste “mundo” construído a partir das informações contidas nas bases disponíveis no sistema, aspectos como clima, vegetação, relevo, densidade demográfica, nível de carência social por região, regiões onde existem mais casos ou menos ocorrências de determinada doença, hidrografia, limites urbanos e rurais, pontos de maior propensão à criação de mosquitos (tais como lixões, quartéis de polícia, cemitérios, lagos, piscinas dentre outros no caso da dengue), enfim todos estes aspectos devem ser organizados, analisados e transformados de acordo com a realidade da doença que se pretende simular.

De uma maneira geral, cada doença tem seu próprio mundo, regras e maneira de atuar. Por isso, não está sendo aqui definido um único conjunto de dados que seriam acessados por todos os modelos que viessem a integrar este sistema. O que ocorre aqui é que para o modelo da dengue os mesmos dados que interessam, tais como densidade de mosquitos por área, temperatura, índice de chuvas, regiões onde existem mais áreas verdes ou formação de poças d’água dentre outros, podem não ser interessantes para a hanseníase ou a tuberculose, por exemplo. Desta forma, é necessária a definição de um Modelo de Dados Comum (MDC)

particular para cada domínio de aplicação que compor o sistema de simulação. Na seção 4.3 é apresentado um protótipo do modelo de dados comum para o modelo da dengue.

Isto não significa que será necessário construir um serviço para cada doença, mas somente duas coisas quando se quiser adicionar um novo modelo epidêmico no estágio atual do desenvolvimento: um simulador com as regras de interação adequadas, capaz de interpretar a entrada gerada pelo serviço IntegraModel e um arquivo com as definições básicas do MDC pertinentes àquela doença. Isso foi feito para a dengue.

Com isso, o modelo não precisará ser alterado toda vez que se quiser prever o comportamento da doença infecto-contagiosa em outra nova cidade, dado que ele será parametrizado de acordo com os dados daquela nova cidade, não havendo com isso a necessidade de se criar um novo programa para se adaptar àquela realidade. Em outras palavras, o modelo será capaz de se adaptar à realidade de qualquer cidade bastando apenas ser parametrizado de acordo com as características daquele local, dado que os mecanismos de espalhamento (como os descritos no capítulo 3) se manterão os mesmos para qualquer estrutura que se desejar.

Conseqüentemente, um mesmo modelo poderia ser aplicado para simular o comportamento de uma doença considerando vários cenários distintos. Contudo, cada doença deverá ter seu próprio modelo epidemiológico, com suas regras de transição particulares e um simulador próprio que deve ser construído de maneira modular e parametrizável.

Ainda na figura 4.4, após obter os dados necessários para que o simulador execute uma simulação na cidade desejada é criado um arquivo xml de acordo com o esquema apresentado na seção 4.3 do presente trabalho e então uma instância de um cliente do WS-GRAM é criada com a definição da(s) tarefa(s) que se pretende executar, bem como a definição do caminho onde o arquivo de entrada está localizado para ser transferido no processo de *stage in* da tarefa. Desta maneira, o serviço é capaz de interagir com o WS-

GRAM e verificar se o simulador se encontra na máquina do cliente para que o mesmo seja transferido para a “Máquina B” via RFT (que pode ser uma transferência terceirizada ou não, de acordo com a localização do simulador no sistema).

Após isso, o serviço WS-GRAM do *Globus Toolkit* submete aquele conjunto de tarefas ao adaptador do serviço GRAM, que se encarrega por sua vez de distribuir as requisições aos escalonadores de tarefas locais que se encontrarem disponíveis no momento. Na parte do sistema IntegraEPI que se encontra na Universidade Católica de Santos há os escalonadores PBS e Condor instalados, mas caso eles estejam indisponíveis, podem ser configurados jobs em *batch* utilizando o próprio mecanismo *Fork* dos sistemas operacionais que façam parte desta grade.

A simulação é então executada e seus resultados são definidos pelo *xml schema* apresentado na seção 4.3 (figura 4.11, página 122), que contém basicamente informações sobre o número de pessoas em cada estado a cada dia da simulação para cada realização e dados referentes à posição no espaço de cada novo humano infectado que surge durante a simulação para ser posteriormente visualizado sobre um mapa se tal funcionalidade estiver disponível na aplicação cliente.

No sistema IntegraModel cada simulação é uma tarefa única e independente, sendo parte de um conjunto de simulações que se caracterizam como uma aplicação *bag-of-task*. Cada simulação pode levar de alguns instantes a dias ou semanas para concluir sua execução, dependendo da parametrização que se fez, o que geralmente é proporcional ao tamanho da rede e o grau de mobilidade das pessoas e o número de realizações que se pretende que sejam calculadas para resultados em média, basicamente. Nos testes efetuados durante o estudo do modelo, as simulações com redes de 10000 habitantes e ausência de mobilidade humana e do mosquito duraram aproximadamente de 2 horas em média para um conjunto de 100 realizações, por exemplo, em uma máquina com processador Intel Pentium IV Hyper Thread,

com 1 GB de memória RAM, disco rígido de 80 GB e executando de maneira dedicada à aplicação no sistema operacional Linux Red Hat 9.0.

Ao mesmo tempo, uma simulação com os mesmos parâmetros mas apenas uma realização demorou cerca de 1 minuto em média. Contudo, para redes maiores, com um número de habitantes igual ao da cidade de Santos, por exemplo (cerca de 450 mil habitantes), uma realização nesse pior caso durou de 20 a 30 minutos em média para uma máquina com a mesma configuração.

De fato, pensando nesses números, o sistema foi construído para possibilitar resultados *offline* aos usuários, ou seja, resultados que poderão ser obtidos após o término das tarefas. Assim, tarefas de previsão de epidemias poderiam ser agendadas e colocadas em execução por um cliente e, tão logo estivessem prontas, o cliente poderia recuperar seus resultados e analisá-los de acordo.

Para isso, um mecanismo de indexação seria importante para que a recuperação das simulações fosse possível aos clientes do IntegraModel. Por isso, pensando em termos de escalabilidade e segurança dos dados disponíveis na grade adotou-se a idéia de registrar tais resultados no servidor de réplicas do Globus Toolkit 4, o *Replica Location Service*. Por possibilitar armazenar os resultados como réplicas, o sistema ganha em desempenho, segurança e escalabilidade, uma vez que vários servidores RLS podem ser utilizados por todo sistema, o que descentralizaria a tarefa de um único servidor estar responsável por cuidar desses resultados, ao mesmo tempo em que os resultados estarão distribuídos geograficamente pela grade, acessíveis para serem transferidos onde a entrada no LRC estiver se referindo.

Podemos observar isso ainda na figura 4.4 quando as instâncias do simulador se conectam ao cliente do RLS para registrar seus resultados. De fato, essa foi outra decisão adotada aqui pensando na escalabilidade do sistema. A idéia básica é que após a simulação terminar, a mesma poderia se registrar em um servidor RLS e assim evitar que um servidor se

tornasse um ponto único de falhas no sistema. Mesmo assim, no serviço IntegraModel também existe uma funcionalidade na qual é possível fazer com que o serviço, ao invés das próprias tarefas, registre os resultados no RLS.

Cada entrada no servidor de réplicas é constituída por um nome lógico e um nome físico. O nome lógico adotado foi o próprio ID da tarefa, constituído por uma concatenação entre o nome da cidade simulada, data da simulação e uma seqüência *hash* gerada a partir da classe UUIDGen disponível no pacote org.apache.axis.components.uuid (e.g. “Santos\_02-02-2007\_d844f4c0-d34a-11db-95fa-b82f185b40ce.xml”). O nome físico consiste basicamente numa URL do serviço GridFTP contendo o caminho para o arquivo físico armazenado na máquina indicada, como por exemplo: “gsiftp://serverpos.unisantos.edu.br:2811/tmp/Santos\_02-02-2007\_d844f4c0-d34a-11db-95fa-b82f185b40ce.xml”. Com isso, o serviço retorna ao cliente o nome lógico dos resultados daquela submissão e aí cabe ao cliente recuperar tais informações e trabalhá-las conforme necessário.

Assim, em linhas gerais nesta seção foram apresentadas as principais características atuais do serviço de simulações de epidemias do sistema IntegraEPI. De uma maneira geral é importante lembrar que o mesmo se trata de um mecanismo de simulação, mas não um mecanismo de análise e portanto, não foi pensado para este serviço mecanismos capazes de analisar os resultados e escolher dentre eles qual seria o mais adequado à realidade de determinada cidade, o que seria um passo posterior ao presente trabalho.

### 4.3 CONSTRUINDO UM CLIENTE PARA O SERVIÇO INTEGRAMODEL

Para efetivamente testar o funcionamento do serviço de simulação apresentado na seção anterior foi desenvolvido um cliente simplificado, cujas funcionalidades e maneira como interage com o serviço estarão agora em discussão.

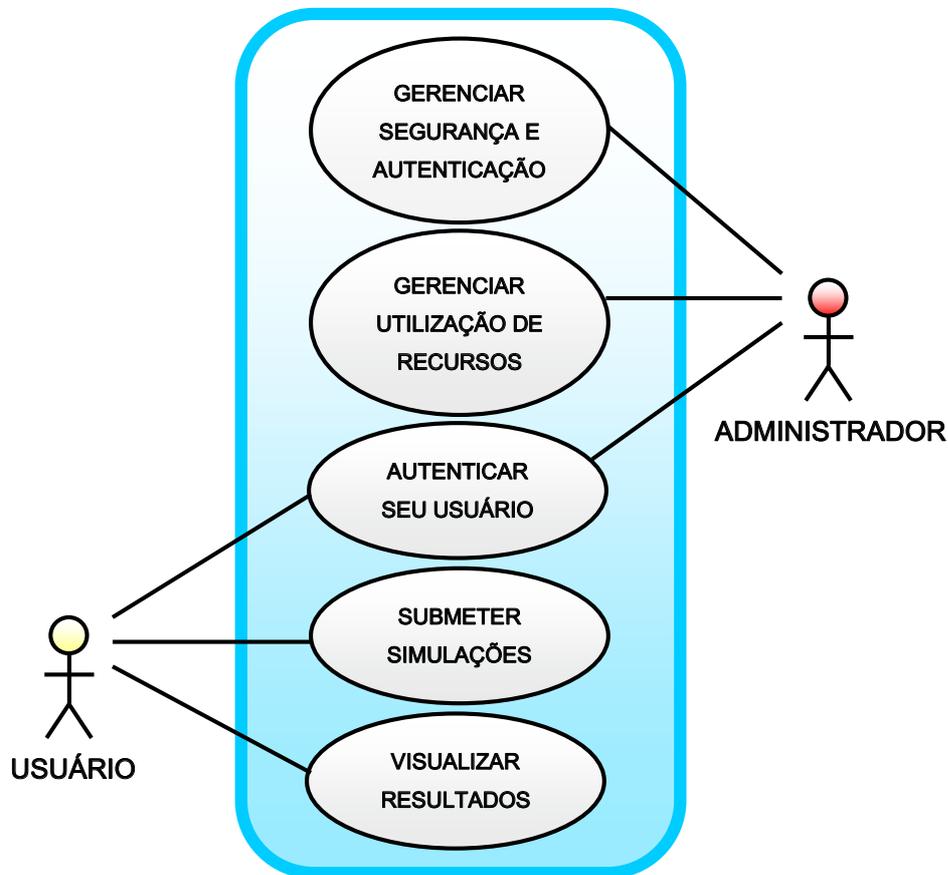
É importante lembrar que o sistema IntegraEPI consiste basicamente em um conjunto de serviços em epidemiologia que deverão estar disponibilizados em um ambiente de grades computacionais para que aplicações possam configurar o cenário de uso da melhor maneira que lhes conviesse, ao mesmo tempo em que vários clientes poderiam ser construídos para um mesmo alvo de aplicação, como é o caso da predição de epidemias. Por isso, o cliente que está sendo discutido nesta seção é apenas um exemplo de como seria uma aplicação capaz de fazer uso do serviço para a simulação de doenças.

Durante o processo de levantamento de requisitos do cliente, foi imaginado que a aplicação deveria ser capaz de submeter tarefas de simulação de doenças bem como recuperar os resultados de tais simulações para serem visualizados. Adotou-se como linguagem de programação a linguagem Java pelas facilidades que a mesma oferece tais como a reusabilidade, polimorfismo e encapsulamento [DEITEL, DEITEL, 2001]. De uma maneira geral, a possibilidade de se trabalhar com a programação orientada a objetos foi o ponto chave de escolha para o desenvolvimento.

De uma maneira geral, várias funcionalidades já haviam sido desenvolvidas para o serviço IntegraModel, tais como um cliente capaz de acessar bases de dados OGSA-DAI, outro capaz de efetuar transferências de arquivos e isso foi outro dos motivos que nos fez optar por Java para nossa linguagem de desenvolvimento. Além disso, o *Globus Toolkit 4.0* oferece vários recursos disponibilizados a partir da biblioteca Java WS-Core, que de uma maneira geral é a base para qualquer aplicação desenvolvida para grades computacionais usando este *toolkit*.

Uma aplicação capaz de oferecer um ambiente ao usuário para visualizar o espalhamento de doenças simuladas a partir dos modelos disponíveis sobre mapas de cidades reais foi outro dos requisitos definidos durante o processo de engenharia de software deste aplicativo [ODELL, MARTIN 1996] (aqui foram empregados os diagramas de caso de uso,

classes e colaboração da UML [FOWLER, SCOTT 2000; FURLAN 1998] para representar os vários aspectos do cliente nesta seção).



**Figura 4.5** – O diagrama de casos de uso geral do sistema IntegraModel.

Além disso, foram definidos até o momento dois papéis básicos para os usuários do sistema: o administrador e o usuário comum. O primeiro deverá ser capaz de gerenciar a segurança, autenticar novos usuários na grade e gerenciar a utilização dos recursos. O usuário comum por sua vez, deverá ser capaz de submeter simulações na grade e poder visualizar os resultados de alguma maneira, por meio de um cliente, por exemplo. Contudo, para que as duas classes de usuários possam efetuar seus papéis, ambos devem se autenticar como usuários válidos dentro da organização virtual IntegraGrid do projeto IntegraEPI. O diagrama da figura 4.5 esquematiza a maneira como foi planejada a interação das duas classes de usuários com o sistema.

Até o presente momento da escrita desta dissertação, havia sido definido que o projeto

IntegraEPI compreenderia a organização virtual IntegraGrid, e sua primeira unidade organizacional seria a IntegraGridUnisantos, atualmente instalada na Universidade Católica de Santos e também a responsável pela autenticação de usuários do projeto IntegraEPI. Contudo, com a colaboração de outras instituições é possível que tal organização ainda sofra alterações estruturais, havendo portanto a possibilidade da entrada de novas unidades organizacionais.

Para submeter simulações ao serviço IntegraModel, o cliente deverá ser capaz primeiramente de definir o conjunto de simulações que pretende executar, configurando cada uma delas de acordo com o que se pretende simular.

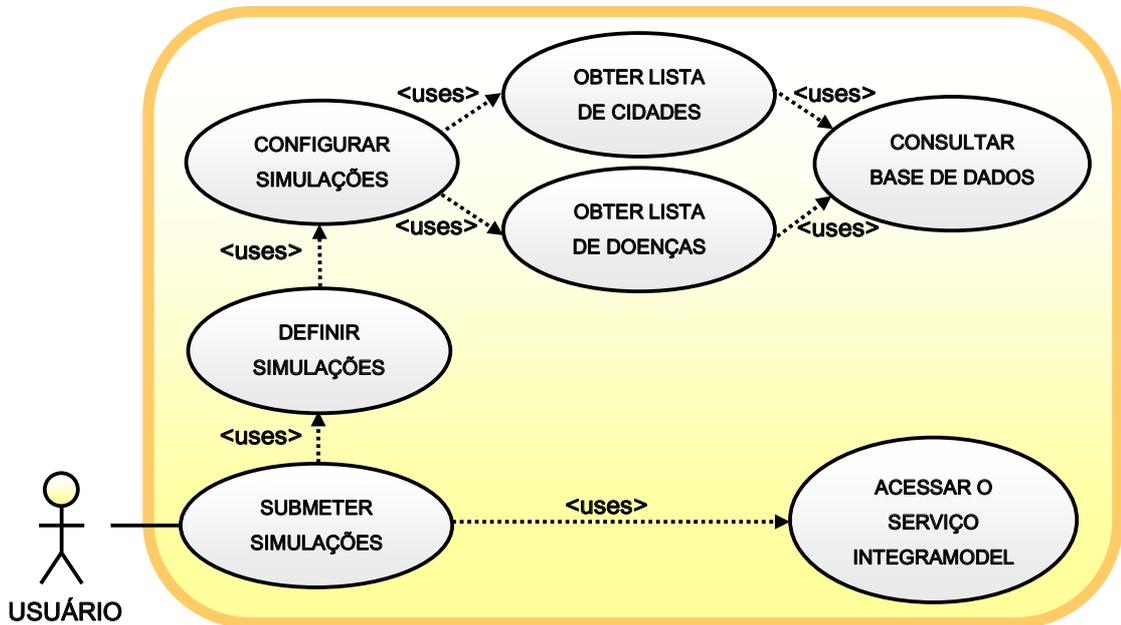
Neste caso, o cliente deve ser capaz de efetuar uma rápida consulta nas bases de dados de mapas do sistema IntegraEPI (disponibilizadas como serviços de acesso a dados do OGSA-DAI [OGSA-DAI, 2006]) para conhecer quais doenças possuem modelos de simulação disponíveis no momento bem como a lista de cidades que possuem informações para a execução de simulações.

Após isso, o cliente deve acessar o serviço IntegraModel e então submeter a definição de tarefas ao serviço. A forma como isso deve acontecer está esquematizada no diagrama de casos de uso da figura 4.6.

No cliente apresentado e discutido o usuário ainda precisa inserir no programa a url dos serviço de simulação e acesso a dados manualmente para que o sistema consiga simular e consultar as bases de dados com a localização dos mapas.

Contudo, esta é apenas uma primeira versão deste cliente que deverá ser melhorado. A intenção do sistema é que tanto este cliente quanto outros que façam uso do sistema IntegraModel atuar como *broker*, sendo capazes de acessar o serviço de informações MDS do *Globus Toolkit*, descobrir automaticamente a localização desses serviços na grade e selecionar

as máquinas apropriadas para o uso apropriado. Tal funcionalidade deverá ser implementada nas próximas versões do cliente do sistema IntegraEPI.

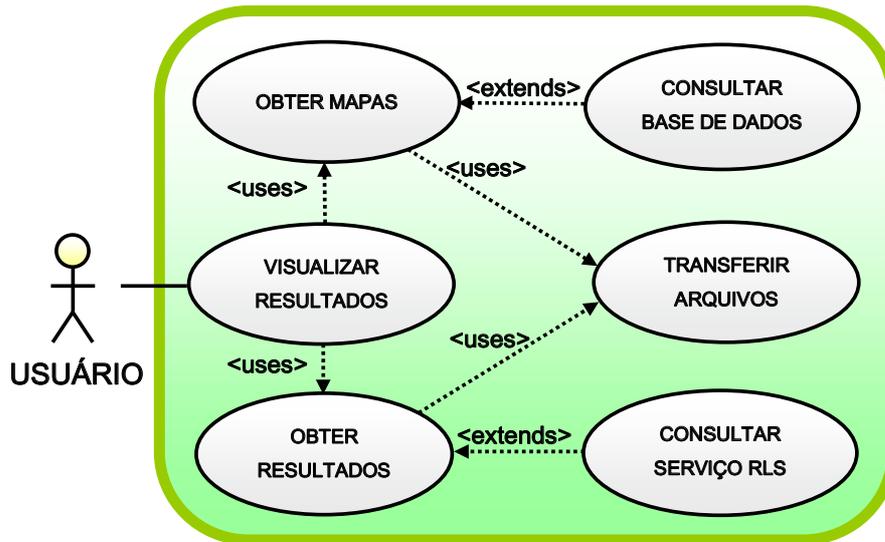


**Figura 4.6** – O diagrama de casos de uso do sistema IntegraModel para a submissão de simulações.

Após o término da execução das simulações o cliente deveria possibilitar ao usuário a visualização do resultado das simulações submetidas para a cidade escolhida.

Para isso, planejou-se que o cliente do serviço IntegraModel deveria ser capaz de obter os mapas e resultados. Contudo, primeiramente seria necessário efetuar consultas no sistema para encontrar onde se encontram tais dados e após isso transferir estes arquivos para a máquina onde a aplicação cliente está executando. A figura 4.7 apresenta um diagrama de classes [RUMBAUGH, BOOCH, JACOBSON 2000] esquematizando a forma como a visualização de dados foi pensada.

Foi planejado inicialmente que a informação sobre os mapas disponíveis e onde eles se encontram estaria sendo disponibilizada em bases de dados, como um servidor de mapas. Para o resultado das simulações entretanto, devemos lembrar que durante o projeto do serviço planejamos que as simulações se registrariam no serviço de réplicas do *Globus Toolkit 4*.



**Figura 4.7** – O diagrama de casos de uso do sistema IntegraModel para visualizar resultados.

Portanto, a aplicação deveria primeiramente consultar os resultados neste servidor de réplicas para então recuperar os resultados da simulação de alguma maneira, de preferência usando os recursos da grade computacional. Para isso, um cliente do serviço GridFTP do Globus Toolkit 4.0 poderia efetuar esta tarefa e foi então esta a escolha realizada para o processo de desenvolvimento deste cliente.

No momento o acesso a estas bases de dados é feito a partir de um serviço simples do OGSA-DAI criado para acessar uma base contendo informações sobre os mapas, mas a idéia é substituir este acesso para chamadas ao serviço GISE, no qual estes mapas estariam sendo listados em seus esquemas integrados de dados quando ele estiver disponível para uso.

Diante da definição das funcionalidades desejadas para o cliente, foi definido então o diagrama de classes da figura 4.8.

Este diagrama apresenta os dois pacotes criados para a organização das classes da aplicação cliente. O pacote `br.com.integraepi.integramodel.visual` é basicamente o principal pacote da aplicação. É este pacote o responsável por tratar das funcionalidades de submissão de tarefas, por oferecer ferramentas para a configuração de simulações e também, como o próprio nome diz, de cuidar da parte visual da aplicação cliente, tal como gerenciar a interface

com os usuários, manipular mapas e resultados de simulação e apresentá-los na tela em camadas, cuja exibição pode ser personalizada de maneira a permitir o controle da aplicação ao usuário.

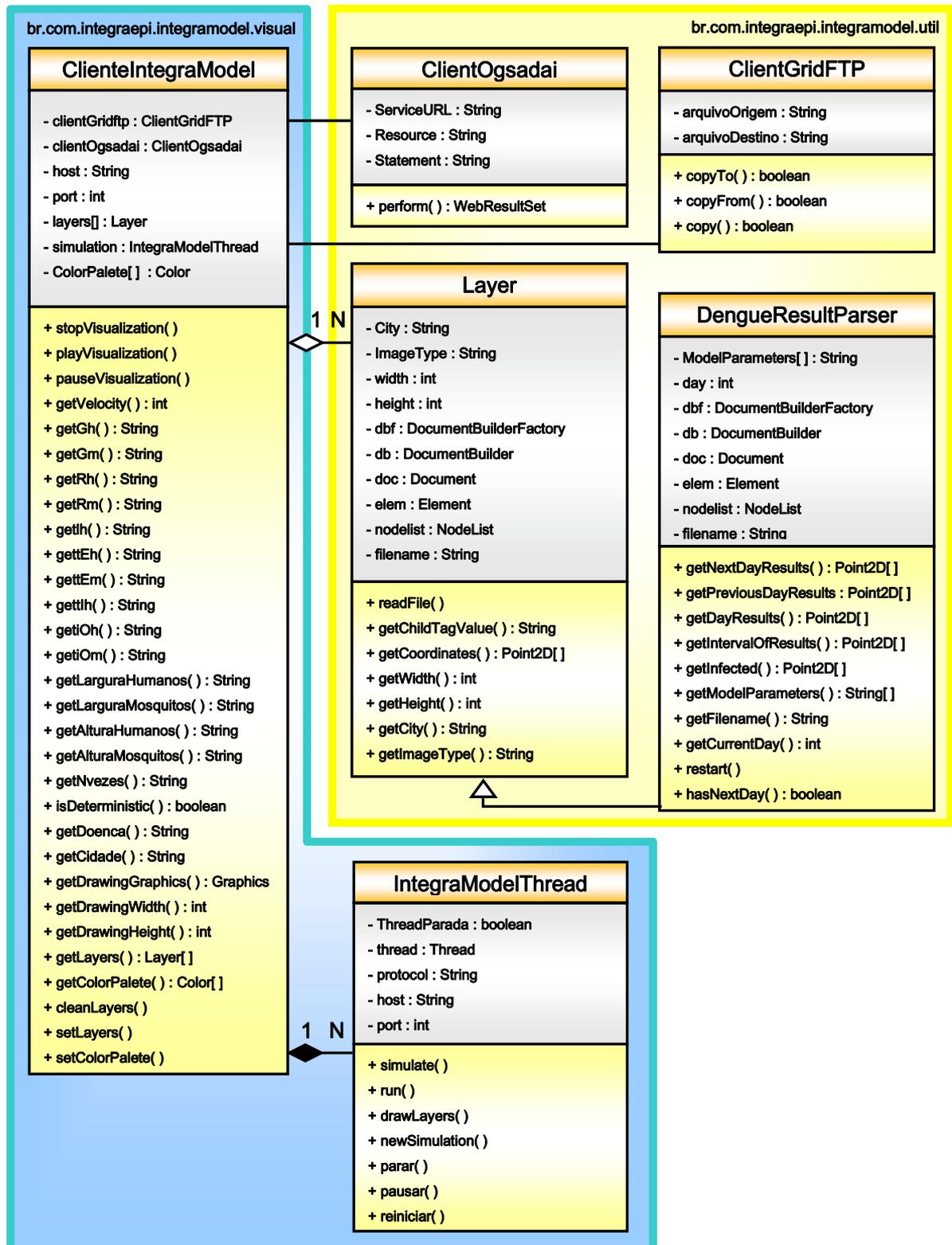
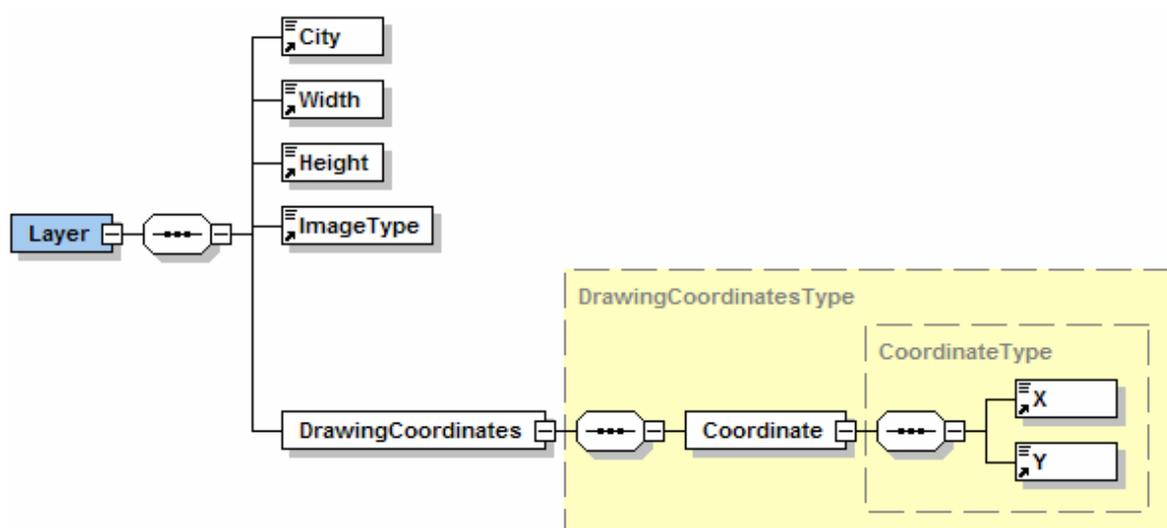


Figura 4.8 – O diagrama de classes da aplicação cliente.

O pacote `br.com.integraepi.integramodel.util`, por outro lado, já oferece ferramentas necessárias para que a aplicação consiga consultar a localização dos dados a serem exibidos bem como oferecer maneiras para transferi-los à máquina onde a aplicação cliente se encontra executando. Além disso, nesse pacote também se encontram as classes responsáveis por ler os arquivos transferidos e organizar seus dados como camadas (ou *layers*), como é o caso da classe `Layer` e da classe `dengueResultParser`, que também é uma `Layer`, mas possui funcionalidades extras para a manipulação dos resultados de simulação do modelo da dengue.

Particularmente, a classe mais importante da aplicação é a `ClienteIntegraModel` do pacote `br.com.integraepi.integramodel.visual`. Esta classe agrega objetos da classe `ClientOgsadai` que são utilizados durante o processo de consulta por modelos epidemiológicos disponíveis e das cidades disponíveis para simulação no sistema. Da mesma forma, a `ClienteIntegraModel` também agrega objetos da classe `ClientGridFTP` que são empregados para a transferência de dados como mapas e resultados de simulações realizadas.



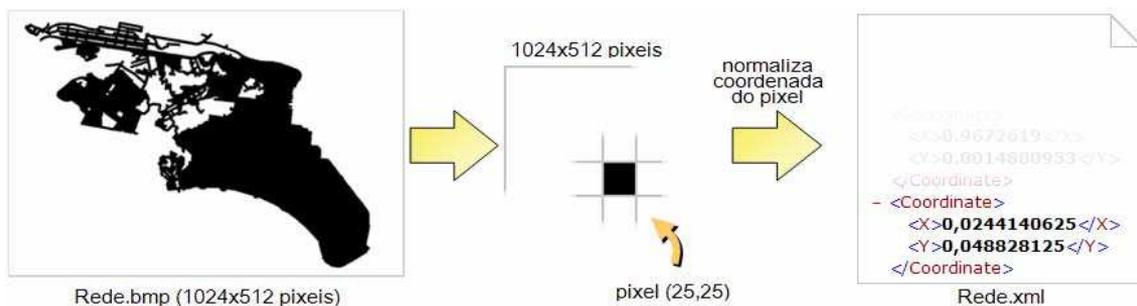
**Figura 4.9** – Esquema de um arquivo contendo a descrição de um mapa.

Todos os mapas após serem obtidos pela aplicação são então lidos e manipulados como instâncias da classe `Layer`. Esta classe emprega um *parser* DOM nos mapas, que são basicamente arquivos XML, para recuperar as informações sobre os mapas obtidos, tais como dimensão do mapa, tipo do mapa (ou seja, se é um mapa de setores censitários, ruas, bairros

ou limites de município), qual cidade a que se refere aquele mapa, bem como as coordenadas geográficas necessárias para sua *plotagem* pela classe ClientIntegraModel. Estes arquivos XML são estruturados como apresentado na figura 4.9.

Para entender melhor como utilizar e como estes mapas são criados, deve-se pensar novamente na idéia discutida rapidamente no capítulo 3 a respeito de sítios vazios e a geometria da rede (na Figura 3.6).

Para que o simulador conseguisse criar uma rede com o mesmo formato da cidade que se deseja modelar, foi criado um programa em C estruturado capaz de converter uma figura bitmap monocromática em um arquivo xml. O mecanismo empregado pelo programa criador de mapas é esquematizado na figura 4.10.

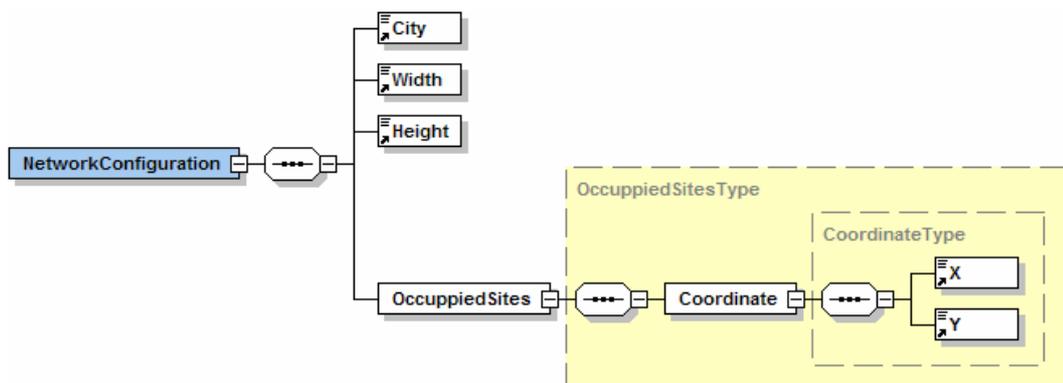


**Figura 4.10** – Esquema de conversão de uma figura com um mapa para um arquivo xml.

Inicialmente é passada uma figura como a que está na figura 4.10, totalmente branca e com o desenho que se pretende mapear em preto. A partir daí o programa que a converterá irá ler pixel por pixel desta imagem, pegar a coordenada (i,j) daquele pixel e normalizar de acordo com tamanho da imagem, ou seja, fazer  $X_n=i/altura$  e  $Y_n=j/largura$  da figura. O valor de  $X_n$  e  $Y_n$  é então salvo em um arquivo cujo esquema é igual ao da figura 4.9. A normalização é feita justamente para que aquela coordenada depois possa se adaptar a uma área qualquer de desenho, cabendo porém para isso adotar as devidas proporções para o tamanho de cada ponto, caso esta área seja maior do que a resolução original da figura.

O mesmo mecanismo é empregado para a definição de um arquivo xml do Modelo de Dados Comum para a dengue, com as coordenadas que determinariam as regiões válidas para

que a simulação ocorra. De fato, o arquivo mostrado na figura 4.10 (Rede.bmp) é um exemplo do que poderia ser considerado uma área válida para simulação em uma rede, ou seja, onde é preto existem pessoas que utilizam as regras do modelo descrito no capítulo 3 e onde está em branco na figura a rede é formada por elementos desocupados. O esquema para a criação deste tipo de rede é apresentado na figura 4.11 e está sujeito a modificações estruturais em trabalhos futuros.



**Figura 4.11** – A estrutura do xml schema para o Modelo de Dados Comum.

Os resultados de simulação são recuperados e lidos pela classe dengueResultParser de maneira similar aos mapas, mas é possível se obter uma lista de infectados por um único dia em particular ou por um intervalo de dias, que pode ser o tempo em que uma pessoa fica efetivamente espalhando a doença pela rede.

Além disso, a aplicação permite que os dados de casos de dengue simulados pudessem ser vistos de maneira cumulativa na tela ou apenas os casos de pessoas que se encontram infectados naquele momento. Um arquivo de resultados do modelo da dengue tem uma estrutura como a apresentada na figura 4.12.

A partir do resultado de uma tarefa de simulação é possível ser recuperada a configuração de parâmetros adotada e o conjunto de resultados para cada realização. Cada realização contém vários dias de simulação e cada dia apresenta a contagem naquele instante de simulação do número de elementos que se encontra em cada estado do modelo.

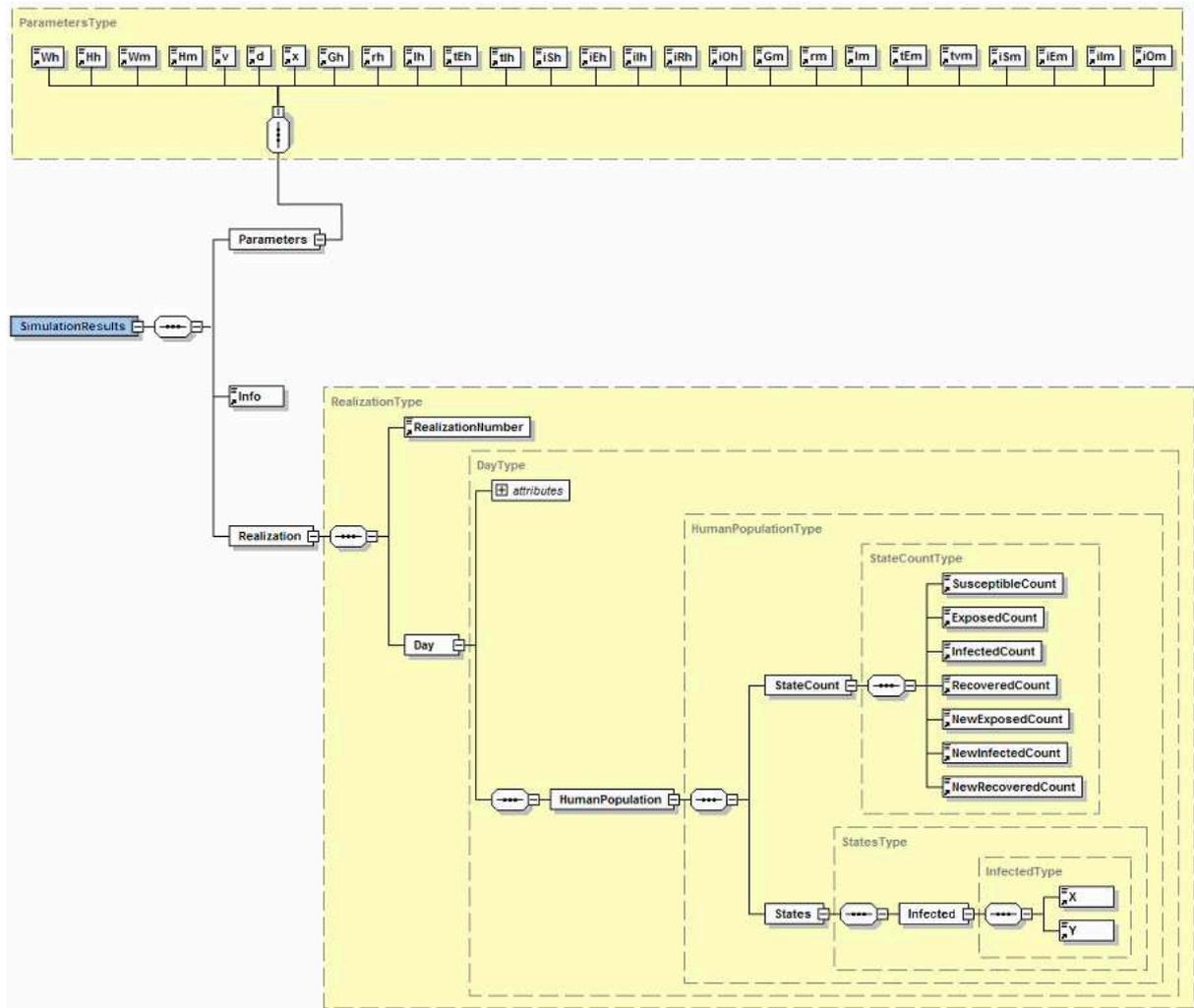


Figura 4.12 – A estrutura do xml schema dos resultados do modelo da dengue.

Também é salvo a lista dos novos infectados que apareceram na rede naquele instante para serem desenhados pelo cliente sobre o mapa. Estas coordenadas também são coordenadas normalizadas de acordo com a dimensão da rede de humanos, tal como ocorre no processo de criação de mapas.

Para a visualização dos resultados na tela foi necessária a criação de uma classe dedicada apenas à *plotagem* das camadas de dados na tela, dado que ocorre toda vez do usuário submeter uma simulação e a partir de então a aplicação permanece em espera até que aquela tarefa seja concluída. Por isso, a classe *IntegraModelThread* foi criada justamente para que as simulações e visualizações pudessem ser tratadas como uma *thread* separada da interface do usuário.

Assim, a classe ClientIntegraModel sempre que inicia uma nova simulação dispara uma nova instância da classe IntegraModelThread que se encarrega então de submeter a simulação configurada, buscar os resultados e atualizar a tela com as animações da evolução de casos simulados sobre os mapas recuperados, ficando livre para oferecer ao usuário o total controle da aplicação.

Para resumir definitivamente a maneira como o cliente discutido aqui se comporta, foi elaborado o diagrama da figura 4.13. Neste diagrama é possível ver como a submissão de tarefas ocorre e como após sua execução o cliente é capaz de recuperar os resultados de simulação e os mapas na seqüência. É interessante observar que apesar das máquinas ainda serem selecionadas manualmente no cliente apresentado no próximo capítulo, toda a execução deverá ser efetuada em várias máquinas que compõem o sistema de maneira automática, podendo ser exemplificado da maneira como apresentada no diagrama, o que caracteriza a idéia base de um sistema baseado em grades computacionais.

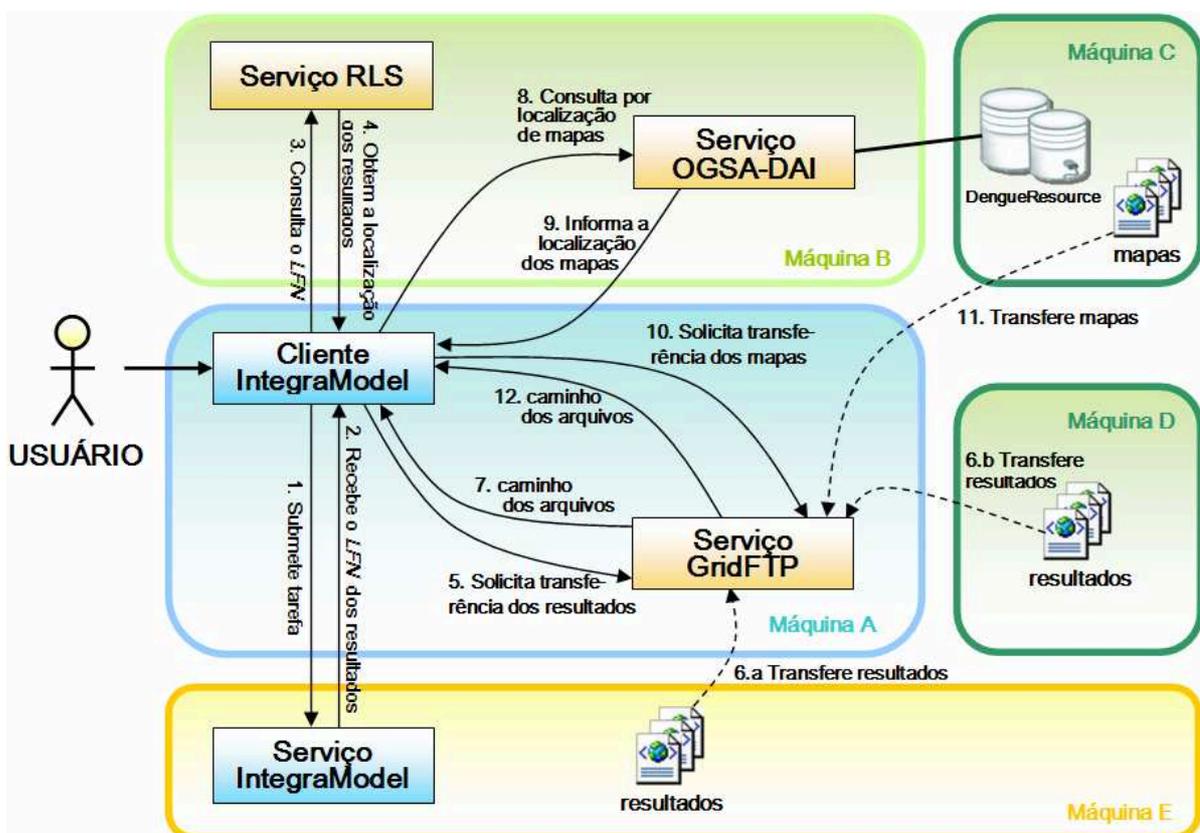


Figura 4.13 – Esquema da execução da aplicação cliente.

A partir da aplicação cliente há uma completa abstração da estrutura do sistema por parte do usuário, que desconhece a localização dos dados ou mesmo do serviço que está realizando suas simulações, oferecendo portanto a transparência que uma aplicação distribuída deve ter.

Como se pode notar na figura, a seqüência dos eventos está numerada de acordo com a ordem em que eles ocorrem. Portanto, é possível observar que o usuário utiliza a aplicação cliente inicialmente para submeter uma tarefa ao serviço IntegraModel.

Após a execução da tarefa pelo serviço (abstraindo na figura o processo que o serviço se utiliza para que a execução ocorra na grade), este retorna um nome lógico (*Logical File Name* ou *LFN*) para a aplicação recuperar o conjunto de resultados que foram obtidos. Este nome lógico é o mesmo comentado na seção 1.5.6.3 deste trabalho, uma vez que se trata do nome lógico em que estes arquivos foram registrados no servidor de réplicas do Globus Toolkit (RLS). O cliente então efetua uma rápida consulta no servidor RLS e obtém uma lista de endereços físicos (*Physical File Name* ou *PFN*), que durante o processo de registro pelo simulador foram salvos da maneira que se comentou na seção 5.2, ou seja, como uma URL do serviço GridFTP.

Na seqüência o cliente então passa a recuperar os arquivos que podem estar distribuídos por toda a grade através do serviço GridFTP. Vale notar que na figura estes arquivos estão nas máquinas D e E, mas eles não precisariam obrigatoriamente estar nestas máquinas toda vez que as simulações fossem executadas. A idéia apresentada neste diagrama é a de que os dados poderiam vir das diversas máquinas nas quais foram escalonadas as tarefas de simulação do modelo configurado pelo usuário.

Ainda na figura 4.13, após o cliente recuperar os dados simulados é então a vez de recuperar os mapas da cidade na qual a simulação foi efetuada. Para isso, o cliente consulta a partir da classe ClientOgsadai as bases de dados contendo informações sobre a localização

dos arquivos de mapas e mais uma vez solicita ao GridFTP por meio de uma instância da classe ClientGridFTP a transferência dos mapas disponíveis. Cada arquivo é copiado para a máquina onde o cliente está executando e então o cliente é capaz de ler os arquivos, fazer a interpretação de seus conteúdos por meio das classes Layer e dengueParserResult para então apresentar a visualização na tela para o usuário.

#### 4.4 CONCLUSÕES E COMENTÁRIOS GERAIS

Ao longo deste capítulo apresentamos o sistema IntegraModel, um dos módulos que constituem o sistema IntegraEPI cujo objetivo principal é o possibilitar aos agentes da vigilância epidemiológica a possibilidade de pré-visualizar o espalhamento de doenças em seu município de atuação. Assim, foram discutidos aqui os detalhes de projeto e desenvolvimento deste serviço bem como também exemplificamos a maneira como poderia ser projetado um cliente simples para a simulação e visualização de doenças usando as funcionalidades do serviço IntegraModel em um ambiente de grades computacionais. Como pôde ser visto, tal cliente ainda é apenas um simples exemplo de como poderíamos compor os vários elementos do sistema IntegraEPI para a construção de uma aplicação. Contudo, é extremamente útil lembrar que o sistema apresentado está longe de ser uma versão definitiva, cabendo ainda espaço ainda para muitos outros trabalhos que o tornarão ainda mais completo e funcional.

Uma das limitações do serviço IntegraModel é simplesmente o fato de que ainda se faz necessário um estudo muito mais aprofundado sobre as características de cada cidade para que o modelo apresente realmente resultados convincentes e satisfatórios. Contudo, isso só será possível a partir de uma parametrização dos modelos mais bem elaborada e, nesse sentido, o a construção de Modelo de Dados Comum (MDC) mais rico em detalhes é fundamental. Conseqüentemente, se torna necessário o desenvolvimento de um recurso a mais capaz de

intermediar estes dados, efetuar várias análises e inferências a partir dos dados obtidos nas bases de dados e posteriormente parametrizar o modelo após este procedimento de acordo com o MDC definido.

Uma vez que o mecanismo apresentado durante este trabalho ainda é algo bastante simples (capaz de considerar apenas a geometria da cidade para limitar as áreas habitadas e o número de pessoas), aliado ao fato de não se ter um ainda um estudo geográfico, histórico e populacional mais bem elaborado, a abordagem apresentada no que diz respeito à parametrização do modelo proposto ao longo deste trabalho se tornou mais superficial e simplificada, cabendo aí portanto, outro ponto do sistema a ser melhorado.

Outra falha, consiste no fato de que ainda não há um elemento capaz de analisar os resultados obtidos pelas simulações e identificar cenários de riscos, regiões onde há maior aglomeração de casos e possivelmente detectar a partir disso, padrões relacionados àquela situação. É necessário o desenvolvimento, portanto, de um outro serviço capaz de analisar tais resultados e selecionar os que se apresentaram mais pertinentes à realidade daquela cidade, por exemplo, conferindo ao sistema uma maior interação amigável com o usuário, que geralmente não é alguém especialista em tecnologia ou estatística.

Também seria muito interessante tornar a interface com o usuário o menos complicado possível no que diz respeito à parametrização dos modelos disponibilizados no sistema, uma vez que o mesmo apresenta vários parâmetros e requer um conhecimento mais aprofundado em como cada um deles se comporta para que uma simulação ocorra. Portanto, seria muito importante que a configuração de tais parâmetros fosse feita de maneira mais automatizada, sem muita intervenção com o usuário, porque nossa missão é facilitar e melhorar a qualidade do serviço de vigilância epidemiológica, sendo esta portanto uma das características chave para uma boa aceitação deste sistema.

Para exemplificar o funcionamento do mecanismo apresentado neste capítulo, será apresentado a seguir um estudo de caso deste sistema no qual discutiremos sua utilização por um cliente que deseja simular o espalhamento da dengue.

## CAPÍTULO 5 – ESTUDO DE CASO: SIMULANDO A DENGUE PELO SISTEMA INTEGRAMODEL

Neste capítulo estará sendo apresentado um estudo de caso da utilização do sistema IntegraModel para a simulação de doenças. A doença escolhida foi a dengue pelo fato do sistema no momento disponibilizar apenas o modelo para esta doença, apresentado no capítulo 3.

Inicialmente o cliente apresenta ao usuário uma tela inicial onde ele poderia acessar as três funcionalidades básicas do sistema IntegraEPI: previsão de epidemias, visualização de casos e análise de epidemias.



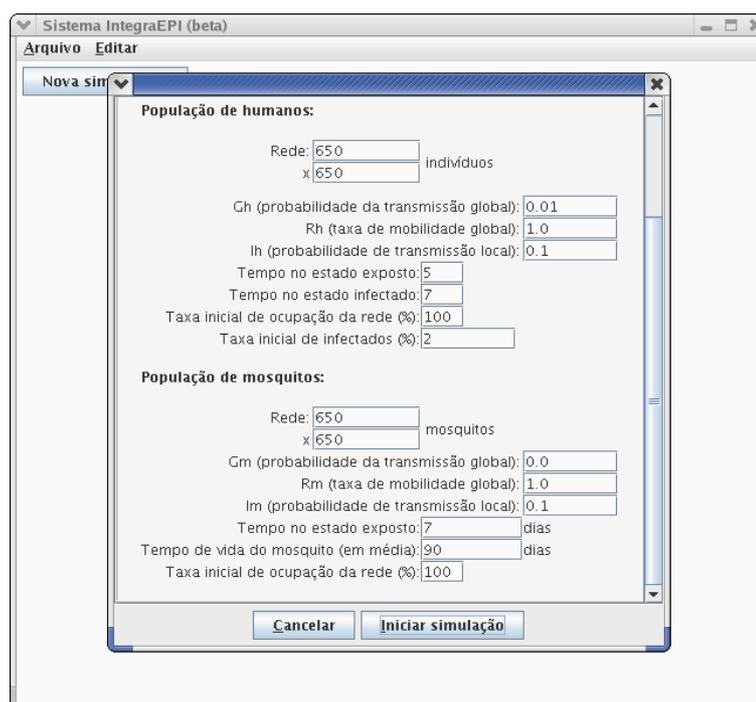
**Figura 5.1** – A tela inicial do cliente.

A única funcionalidade disponível no momento da escrita desta tese é a de previsão de epidemias porque é a que se encontra em estágio mais avançado até então e também foi a que apresentamos ao longo deste trabalho. As outras duas funcionalidades seriam disponibilizadas à medida que os módulos de análise e integração de dados estivessem concluídos. A previsão é de que com elas poderiam ser abertas ferramentas para visualizar a evolução histórica de casos de dados reais sobre mapas, bem como a visualizar dados atuais de ocorrências da dengue ao mesmo tempo em que ferramentas para se analisar os casos de doença e obter

indicadores de risco associados estariam auxiliando o trabalho de monitoramento e tomada de decisão por parte dos agentes de saúde. A figura 5.1 mostra a tela inicial comentada neste parágrafo.

Ao iniciar o módulo de previsão de epidemias do cliente o usuário terá a possibilidade de executar uma nova simulação. Clicando no botão “Nova Simulação” aparecerá uma janela na qual estarão disponíveis em duas caixas de seleção: as doenças disponíveis e as cidades que podem ser simuladas, respectivamente. No momento só há a doença dengue e as cidades de Santos e Guarujá para simular.

Mais abaixo, é mostrada a lista de parâmetros que o modelo da dengue oferece e seus valores padrões poderão ser alterados pelo usuário. Após configurar corretamente tais parâmetros, o usuário poderá iniciar a simulação clicando no botão “Iniciar Simulação”. No exemplo a seguir que estaremos apresentando o usuário primeiramente simulará o espalhamento da dengue na cidade de Santos começando com uma rede formada por 2% de humanos infectados (que transmitem a doença efetivamente). Os parâmetros definidos estão apresentados na figura 5.2.



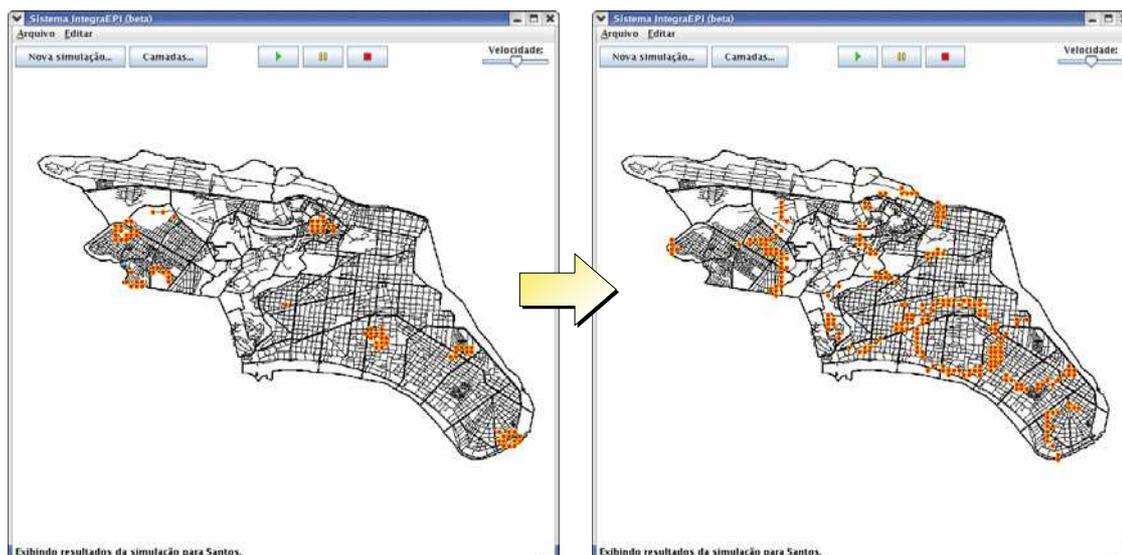
**Figura 5.2** – A tela de configuração de simulações para a previsão de epidemias.

É importante informar ao leitor que no prezado momento o modelo não possui ainda um estudo inicial sobre probabilidades heterogêneas para estas cidades. Por isso, as simulações aqui apresentadas não levam em conta condições particulares de uma determinada região da cidade, na qual há mais ferros-velhos, por exemplo, ou mesmo regiões onde ocorre um maior acúmulo de pessoas idosas (que são potencialmente pessoas que permanecem mais tempo em casa e, portanto, mais susceptíveis a entrarem em contato com o doméstico *Aedes aegypti*), dentre vários outros fatores que influiriam para o que seria um diferencial na parametrização de regiões em cada município no processo de criação da chamada “cidade virtual”.

Portanto, tal susceptibilidade da população infelizmente ainda não foi possível de ser simulada e validada, apesar do modelo já prever e possuir mecanismos para trabalhar com ela. Resta apenas que tais condições sejam devidamente mapeadas e os resultados obtidos sejam validados a partir dos dados reais para que o modelo possa fazer uso deste recurso devidamente.

Voltando ao nosso exemplo, após iniciar a simulação o sistema submete a tarefa ao serviço IntegraModel e aguarda o término da simulação. É importante observar que na versão apresentada aqui a aplicação só mostra resultados a partir do momento em que a simulação terminar sua execução, mas o usuário poderia ser capaz de utilizar outros módulos enquanto a mesma não terminasse. Ocorre que como esta é a única funcionalidade disponível no momento, o usuário deverá esperar pela tarefa, o que pode levar alguns minutos.

Ao terminar a simulação a aplicação recupera os dados simulados e passa a procurar pelos mapas da cidade de Santos disponíveis, que são transferidos logo na seqüência para a máquina onde o cliente está executando. A partir de então, uma animação é apresentada ao usuário com a evolução dos casos simulados sobre o mapa como mostrado na figura 5.3.



**Figura 5.3** – Os resultados simulados da dengue para a cidade de Santos, visualizados como uma animação no cliente do sistema IntegraModel. O usuário pode ser capaz de controlar a animação a partir dos três botões agrupados no topo da janela ao centro, que servem para reiniciar a animação, pausa-la ou pará-la para que possa ser reiniciada novamente. Da mesma forma, o usuário também pode controlar a velocidade da animação a partir da barra de velocidade (no topo da janela, à direita).

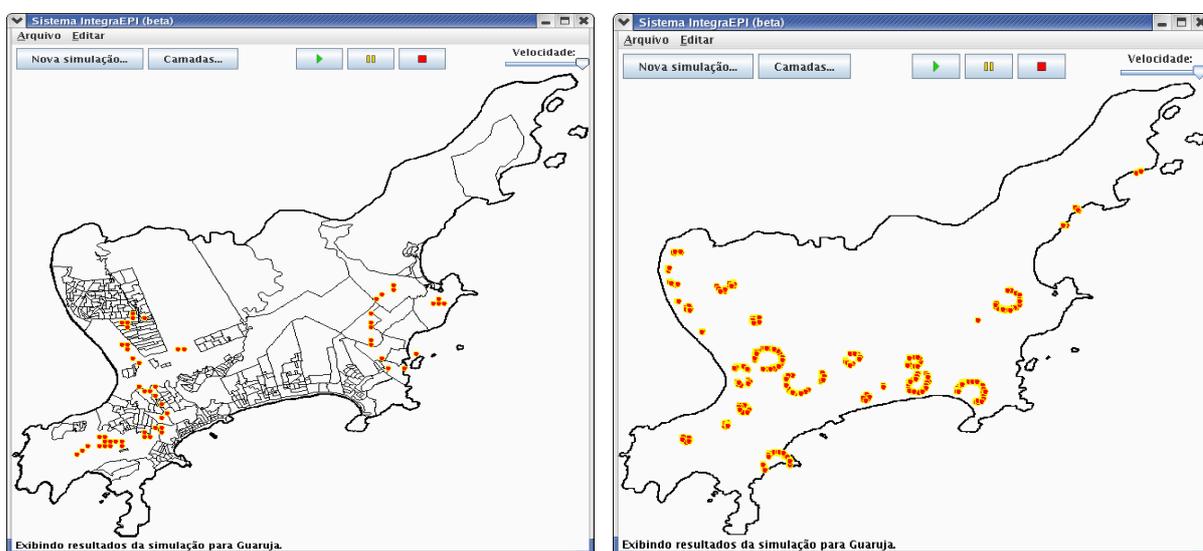
É possível observar que a partir de um cenário inicial no qual apenas 2% da população humana começa infectada pela doença, como se este fosse o quadro atual dos casos reais notificados na cidade de Santos na ocasião, a doença passa a se proliferar ao redor dos primeiros casos e a gerar ondas de espalhamento que vão se alastrando pela cidade.

É importante comentar aqui que nesta simulação nenhuma ação de controle está sendo aplicada para o espalhamento, o que implica que a doença está andando livremente pela rede como se os agentes epidemiológicos não tivessem feito realmente o controle a partir dali. Isso seria interessante para medir o quanto poderia ser grave o espalhamento se nenhuma ação fosse tomada.

Contudo, também vale comentar que para o caso apresentado ainda na figura 5.3, a rede de mosquitos também cobre totalmente a área populada, o que nem sempre ocorre, uma vez que pode haver nuvens de mosquitos localizadas apenas em algumas regiões distintas da cidade, formando locais com mais ou menos mosquitos.

Mais uma vez é importante explicar que para a população de mosquitos tal parametrização não foi ainda implementada mas o recurso já se encontra disponível na estrutura da rede de mosquitos do simulador (trabalhando-se com a idéia de um vértice da rede estar ocupado ou não se podem criar zonas nas quais há mosquitos e zonas onde não há mosquitos, da mesma forma como é montada a estrutura da rede com a geometria da cidade escolhida).

Tanto os resultados quanto os mapas são apresentados como camadas selecionáveis pelo usuário. No exemplo apresentado, a cidade de Santos possui nas bases de dados apenas o mapa de ruas disponível. Entretanto, para a cidade de Guarujá já está disponível um mapa no qual a mesma é dividida em setores censitários e outro mapa contendo o limite do município. Para ativar o menu de controle de camadas basta pressionar o botão “Camadas...” que aparece no topo da janela.



**Figura 5.4** – Uma simulação da dengue em Guarujá sobre dois mapas (setores censitários e limite de município) e apenas sobre um mapa do limite de município. O usuário é capaz de selecionar os mapas que quer exibir e com isso melhorar a visualização da ocorrência de futuros casos em regiões suspeitas.

A figura 5.4 apresenta um exemplo de como o usuário pode alterar a visualização de um conjunto de resultados simplesmente manipulando os mapas em camadas para outra simulação efetuada sobre a cidade de Guarujá.

Esse exemplo é interessante para mostrar que o simples fato de esconder uma camada (de setores censitários, como está na primeira tela desta figura, por exemplo) e passar a exibir apenas a camada de limite de município torna possível que o usuário possa confrontar várias instâncias de visualização de dados pertinentes ao problema da transmissão da dengue.

De uma maneira geral, o cruzamento de tais informações, como é o caso de uma camada com a hidrografia da cidade, por exemplo, e vegetação (se estivessem disponíveis no sistema), poderia ajudar na elucidação da formação de certos clusters em áreas urbanas e rurais da cidade. Com isso, o usuário poderia ser capaz de levantar de hipóteses, por exemplo, do porquê de alguma região ter mais casos ou uma maior concentração de mosquitos e, conseqüentemente, planejar um controle mais efetivo no combate de vetores com o intuito de prevenir futuros desastres epidêmicos.

## CAPÍTULO 6 – CONCLUSÕES E PERSPECTIVAS

A implementação de um sistema como o apresentado neste capítulo, projetado como serviços compatíveis com a plataforma de grade computacional *Globus Toolkit*, abre perspectivas completamente novas no que diz respeito ao acesso a dados em grandes populações e, conseqüentemente, permite a estratificação da epidemia metropolitana em larga escala e um aumento de poder computacional graças à utilização de máquinas que possuem recursos de processamento ociosos que passam a atuar como um supercomputador virtual.

Como discutido anteriormente, as vantagens desta tecnologia para uma área como a saúde coletiva no Brasil são muitas, figurando desde aspectos de tolerância a falhas, segurança e consistência dos dados ao baixíssimo custo de equipamentos necessários para isso comparada ao preço dos *mainframes* atuais. Particularmente, toda a infra-estrutura atual do serviço de saúde em um município ou mesmo no Estado poderia ser reutilizada para fazer parte de uma grade computacional tal como a que está sendo utilizada neste presente trabalho, por exemplo.

Além disso, o serviço de simulação proposto neste trabalho será de grande utilidade pública por se propor a prover mecanismos de previsão do comportamento de epidemias de grande apelo social tal como ocorre com a dengue.

No estágio atual de desenvolvimento ainda falta o projeto de uma interface gráfica mais elaborada para a visualização dos dados simulados, contudo, como comentado anteriormente o sistema ainda se encontra em desenvolvimento e nossa intenção é disponibilizar, conforme formos avançando com o projeto, uma interface muito mais amigável e fácil para a utilização por parte dos agentes de saúde.

Após o término desta dissertação pretendemos continuar nos empenhando quanto ao refinamento e validação do modelo da dengue, ao mesmo tempo em que os vários aspectos de projeto e implementação do sistema de simulação IntegraModel levantados ao longo deste capítulo serão trabalhados.

É importante destacar que os resultados esperados no desenvolvimento deste projeto não se aplicam somente a dados de saúde, e a epidemiologia é só uma de suas aplicações. Existe toda uma classe de problemas de natureza espacial e temporal, sobre os quais simular, detectar, monitorar e visualizar padrões é parte da resposta para o problema.

Com efeito, com o avanço deste projeto podemos pensar em criar e conectar simulações de outros sistemas, incluindo poluição ambiental e atmosférica, transporte, suprimento de água e redes de gestão municipal, a fim de fornecer laboratórios virtuais nos quais seja possível explorar soluções para uma ampla variedade de problemas do mundo real.

Ao longo deste trabalho foram sendo levantados todos os avanços efetuados até o prezado momento no desenvolvimento de um sistema capaz de prover simulações para modelos epidêmicos.

Da mesma forma, as limitações e tendências quanto aos próximos passos foram sendo levantados e vale lembrar que no estágio atual do serviço de simulação ainda há a necessidade de várias outras funcionalidades a serem implementadas a fim de complementar e tornar ainda mais robusto este mecanismo de simulação apresentado durante o presente trabalho.

No momento, mesmo o modelo da dengue precisa ainda continuar com o processo de validação ao nível em que reflita realmente de maneira satisfatória a realidade desta doença para um município que se deseja investigar.

Tal validação deve ser feita gradativamente estudando os efeitos e alterações que a inserção novos mecanismos e características forem sendo agregadas à dinâmica dos indivíduos que formam suas redes. Resultados bastante preliminares nos levaram a acreditar

que o modelo se mostra mais próximo dos resultados reais observados à medida que adicionamos mortalidade, natalidade e trabalhamos com uma taxa de ocupação de mosquitos heterogênea, cabendo aí uma direção de investigação interessante para o modelo atual.

Da mesma forma, um estudo mais completo sobre as características locais de cada cidade em que se pretende simular, considerando condições meteorológicas, geográficas e populacionais também se mostra muito importante tanto para a validação do modelo quanto para o sistema de uma maneira geral. O fato é que isto é que consolidará definitivamente um Modelo de Dados Comum que se aplique a realidade de cada município de uma maneira mais abrangente.

Além disso, ainda se faz necessário que o modelo continue sendo validado e que outros mecanismos sejam inseridos à medida que esta validação avance. Nesse sentido, medidas de controle da doença e estudos sobre o fluxo de entrada e saída de pessoas para cada cidade, interligando as redes simuladas para cada cidade e relacionando os resultados de uma simulação uns com os outros são exemplos de mecanismos interessantes de serem pensados e modelados em futuros trabalhos para o modelo da dengue.

É certo que cada local possui suas próprias características que o fazem únicos neste mundo e que é quase impossível que se crie um modelo que abranja todas as possibilidades de cenários existentes na realidade. Entretanto, pelo menos certas características chave deverão ser melhor levantadas e catalogadas para que o modelo possa ser devidamente parametrizado pelo sistema de simulação IntegraModel.

Além disso, estudos sobre o fluxo de entrada e saída de pessoas para cada cidade, bem como interligar as redes simuladas para cada cidade, relacionando os resultados de uma simulação uns com os outros é algo interessante de ser pensado em futuros trabalhos para o modelo da dengue.

Neste sentido, a maneira como a parametrização do modelo da dengue ocorre hoje também é algo que precisa ser revisado conforme o MDC da dengue for se tornando mais completo, uma vez que o simulador até o momento aceita arquivos de entrada para configuração como o MDC descrito anteriormente no capítulo 4. Portanto, no atual estágio de desenvolvimento deste sistema já está previsto que modificações no serviço IntegraModel e na aplicação do simulador da dengue acabarão ocorrendo com o avanço deste projeto para dar suporte às características que forem sendo adicionadas ao modelo.

A manipulação dos resultados obtidos pelo simulador também é algo que deve ser foco de futuras atenções durante o desenvolvimento deste projeto uma vez que seria extremamente interessante que após o término do módulo de análise houvesse um componente capaz de analisar os resultados obtidos pela simulação do modelo da dengue e separasse aqueles que se apresentarem mais interessantes para o controle de epidemias e os apresentasse para o usuário com maior prioridade.

De uma maneira geral, estamos muito otimistas com o projeto IntegraEPI, uma vez que se trata de uma iniciativa pioneira no que diz respeito a aplicações desse porte voltadas para a saúde coletiva no cenário nacional e esperamos com ele contribuir para a melhoria e modernização do sistema de vigilância epidemiológica presente no Brasil.

## APÊNDICE A – O SIMULADOR DO MODELO DA DENGUE

Neste apêndice estarão sendo discutidos aspectos relacionados à construção do simulador para o modelo da dengue e também a maneira como o usuário poderá interagir com o mesmo a partir da listagem dos comandos suportados até o momento.

### A.1 Construindo o simulador para o modelo da dengue

Durante o desenvolvimento do simulador da dengue, várias decisões foram sendo consideradas. A primeira delas foi a de se construir um programa capaz de oferecer uma interface baseada em comandos.

A linguagem de programação escolhida para isso foi a linguagem C. Tal recurso permite que o simulador seja configurável externamente, o que é algo altamente desejável para o serviço de simulação.

De uma maneira geral, esta aplicação foi construída para a realidade de aplicações *bag-of-task*, mas a paralelização da implementação deste modelo será tema de possíveis pesquisas futuras.

Para suportar a interação de redes populacionais de tamanhos diferentes entre si foi aplicada uma abordagem baseada nos conceitos da computação gráfica de resolução de pontos por área.

Da mesma forma como uma figura pode se adaptar a uma área e ser redimensionada para se adaptar àquela área, duas redes de tamanho distintos, tal como é o caso de uma população de humanos com uma rede 100x100 indivíduos interagindo com uma de mosquitos

de 200x200 indivíduos, é considerada a proporção de 1:4, o que faz com que para cada humano haja 4 mosquitos no sentido da interação de humanos para mosquitos (ver figura 3.2 no capítulo 3 para a explicação das duas formas de interação entre as populações). Isso possibilita então a configuração de redes com densidades diferentes de indivíduos interagindo entre si, o que é uma característica interessante para o espalhamento de doenças cuja ação depende da ação de um vetor transmissor.

Outra característica importante definida durante o projeto do simulador foi a de que cada elemento da população possuiria um conjunto de probabilidades particulares para que fosse possível posteriormente a configuração de diferentes graus de susceptibilidade no sistema.

A partir de uma estrutura de dados contendo as probabilidades comuns do modelo, é possível que cada elemento tenha sua probabilidade global e local particulares, o que facilita assim a configuração de uma rede com diferentes tipos de susceptibilidade visto que durante o processo de espalhamento, tal probabilidade particular para cada elemento é considerada e usada nas equações descritas no capítulo 3.

Como padrão o simulador aplica para todos os indivíduos da rede a mesma probabilidade inicial entrada na parametrização da simulação desejada. Para a parametrização, o simulador é capaz de ler um arquivo de entrada cuja estrutura é definida na figura 4.11 (apresentada no capítulo 4).

Portanto, o simulador é capaz de fazer o processo de *parsing* de um arquivo xml definido com a estrutura da figura 4.11 (capítulo 4) e então configurar a rede com o formato definido pelas coordenadas válidas. Cada coordenada destas é uma coordenada normalizada que é então multiplicada pela dimensão da rede, se adaptando portanto ao tamanho da população definida de maneira proporcional. As coordenadas que não estão descritas no modelo são tomadas como regiões inválidas na rede. O padrão do simulador quando nenhum

arquivo é informado é o de considerar a rede como uma rede quadrada com todos os sites ocupados (a menos que uma taxa de ocupação tenha sido definida pelo usuário). A saída de cada simulação obedece um esquema como o descrito na figura 4.12 do capítulo 4.

Assim, a seguir estarão sendo apresentadas as opções possíveis para a parametrização do simulador desenvolvido para esta aplicação.

## A.2 A interface de comandos do simulador para o modelo da dengue

O simulador da dengue não apresenta uma interface gráfica para a configuração de cada simulação que se deseja realizar, mas ele foi construído para aceitar os vários parâmetros de entrada. A sintaxe básica para se utilizar o simulador é a seguinte:

**dengue [<arg1 value> <arg2 value> ... <argN value>]**

Os argumentos possíveis do simulador são:

**Tabela 1**– Lista de argumentos de entrada do simulador (que não se refere a uma população particular).

-v	Modo verboso: apresenta o estado da simulacao passo a passo na tela.
-d	transicoes de estados deterministicas. Quando omitido, adota-se a probabilidade 1/(tempo total naquele estado)
-x <i>número_de_realizações</i>	Numero de realizações da simulação. Deve ser positivo e maior que zero.

O padrão do simulador é assumir que a simulação não está no modo verboso, não utiliza regras de transições determinísticas e tem número de realizações igual a 1. Para a configuração da população de humanos são aceitos os seguintes argumentos de entrada:

**Tabela 2**– Lista de argumentos de entrada para a população de humanos

-Wh valor	Largura da rede dos humanos. O valor deve ser um inteiro positivo maior que zero
-Hh valor	Altura da rede dos humanos. O valor deve ser um inteiro positivo maior que zero
-Gh valor	Probabilidade de mobilidade global dos humanos. O valor deve ser um decimal no intervalo de 0 a 1.
-rh valor	Taxa de mobilidade global dos humanos. O valor deve ser um decimal no intervalo de 0 a 1.
-lh valor	Probabilidade de infecção local dos humanos. O valor deve ser um decimal no intervalo de 0 a 1
-tEh valor	Tempo de incubação dos humanos. O valor deve ser um inteiro maior que zero.

-tlh valor	Tempo infeccioso dos humanos. O valor deve ser um inteiro maior que zero.
-iSh valor	Taxa inicial de susceptíveis humanos na rede. O valor deve ser um decimal no intervalo de 0 a 1.
-iEh valor	Taxa inicial de expostos humanos na rede. O valor deve ser um decimal no intervalo de 0 a 1.
-ilh valor	Taxa inicial de infectados humanos na rede. O valor deve ser um decimal no intervalo de 0 a 1.
-iRh valor	Taxa inicial de recuperados humanos na rede. O valor deve ser um decimal no intervalo de 0 a 1.
-iOh valor	Taxa de ocupação inicial dos humanos na rede. O valor deve ser um decimal no intervalo de 0 a 1.

O padrão adotado aqui é o de uma rede com  $G_h=0$ , para uma rede com  $W_h=H_h=100$ ,  $r_h=1$ ,  $l_h=0.1$ ,  $t_{Eh}=5$ ,  $t_{lh}=7$  e  $i_{Oh}=1$  (100%). Os argumentos  $i_{Sh}$ ,  $i_{Eh}$ ,  $i_{lh}$  e  $i_{Rh}$  quando não configurados fazem com que a configuração inicial da rede seja de uma rede tomada completamente por susceptíveis com apenas um único infectado ao centro da rede. Para a configuração da população de mosquitos, são aceitos os seguintes argumentos de entrada:

**Tabela 3** – Lista de argumentos de entrada para a população de mosquitos.

-Wm valor	Largura da rede dos mosquitos. O valor deve ser um inteiro positivo maior que zero
-Hm valor	Altura da rede dos mosquitos. O valor deve ser um inteiro positivo maior que zero
-Gm valor	Probabilidade de mobilidade global dos mosquitos. O valor deve ser um decimal no intervalo de 0 a 1.
-rm valor	Taxa de mobilidade global dos mosquitos. O valor deve ser um decimal no intervalo de 0 a 1.
-lm valor	Probabilidade de infecção local dos mosquitos. O valor deve ser um decimal no intervalo de 0 a 1
-tEm valor	Tempo de incubação dos mosquitos. O valor deve ser um inteiro maior que zero.
-tvm valor	Tempo de vida dos mosquitos. O valor deve ser um inteiro maior que zero.
-iSm valor	Taxa inicial de susceptíveis mosquitos na rede. O valor deve ser um decimal no intervalo de 0 a 1.
-iEm valor	Taxa inicial de expostos mosquitos na rede. O valor deve ser um decimal no intervalo de 0 a 1.
-iIm valor	Taxa inicial de infectados mosquitos na rede. O valor deve ser um decimal no intervalo de 0 a 1.
-iOm valor	Taxa de ocupação inicial dos mosquitos na rede. O valor deve ser um decimal no intervalo de 0 a 1.

Da mesma forma como para os humanos, a rede de mosquito é configurada por *default* como  $G_m=0$ ,  $W_m=H_m=100$ ,  $r_m=1$ ,  $l_m=0.1$  (a mesma probabilidade de  $l_h$ ),  $t_{Em}=7$ ,  $t_{vm}=-1$  (os mosquitos vivem eternamente durante a simulação) e  $i_{Om}=1$  (100%). Os argumentos  $i_{Sm}$ ,  $i_{Em}$  e  $i_{Im}$  quando não configurados fazem com que a configuração inicial da rede seja de uma

rede tomada completamente por susceptíveis com apenas um único infectado ao centro da rede. Para a parametrização da configuração inicial da rede e a definição do arquivo de saída da simulação, são disponibilizados os seguintes argumentos:

**Tabela 4** – Lista de argumentos de entrada para configuração da rede e saída da simulação.

-in arquivo	Carrega o arquivo .xml de definições (MDC do modelo).
-out arquivo	Salva os resultados da simulação no arquivo indicado.

O padrão para o simulador é sempre salvar a saída da simulação em um arquivo chamado output.xml quando nenhum nome é especificado.

### **A.3 Problemas conhecidos**

Esta versão do simulador pode apresentar problemas de falta de memória para dimensões de rede muito grandes para humanos ou mosquitos, tal como uma rede 100000x100000, por exemplo.

## REFERÊNCIAS BIBLIOGRÁFICAS

[ABRAMSON, KUPERMAN 2001] ABRAMSON, G., KUPERMAN, M., “Small World Effect in an Epidemiological Model”, **Physical Review Letters**, v.86, n.13, 2001.

[ALVES, GAGLIARDI, 2006] ALVES, D., GAGLIARDI, H.F., “Técnicas de Modelagem de Processos Epidêmicos e Evolucionários” in São Carlos, SP: **Notas em Matemática Aplicada**, v.26, 88p, 20,5cm, 2006, disponível em: <http://www.sbmac.org.br/notas.php>, acessível em 19/12/2006.

[ALVES, GAGLIARDI, SILVA 2006] ALVES, D., GAGLIARDI, H. F., SILVA, F. A. B. “Epidemiology in a computational laboratory: preliminary validation, and samples of simulation results”. **Proceedings of 11th World Congress on Public Health**, Rio de Janeiro 2006.

[ALVES, HASS, CALIRI, 2003] ALVES, D., HASS V. e CALIRI, A. “The predictive power of  $R_0$  in an epidemic probabilistic model”, **Journal of Biological Physics**, vol. 29, issue 1, pp. 63, 2003.

[ALVES, PESSOA, GAGLIARDI, 2004] ALVES, D., PESSOA, T. R. M., GAGLIARDI, H. F. “EpiCASim-dengue: Uma Ferramenta para Visualizar e Analisar a Dinâmica de Transmissão da dengue Baseada em Autômatos Celulares”. **VI Congresso Brasileiro de Epidemiologia**, 2004, Recife. *Revista Brasileira de Epidemiologia* (número especial). ABRASCO, 2004.

[ANDERSON, MAY, 1991] ANDERSON, R. M., MAY, R. M., “Infectious Diseases of Humans”. **Oxford University Press**, Oxford, 1991.

[ANT, 2006] ANT (The Apache Ant Project), disponível em <http://ant.apache.org/index.html>, acessado em 12/01/2007.

[AXELSON, 1999] AXELSON, O. “The character and the public health implications of ecological analyses” in: **Disease Mapping and Risk Assessment for Public Health** (A. Lawson, A. Biggeri, D. Böhning, E. Lesaffre, J.-F. Viel & R. Bertollini, ed.), pp. 301-309, New York: **Wiley and Sons**, 1999.

[BAILEY, 1975] BAILEY, N. T. J., “The Mathematical Theory of Infectious Diseases and its Applications” **Griffin**, London, 1975.

[BAILEY e GATRELL, 1995] BAILEY, T. C., GATRELL, A. C. “Interactive Spatial Data Analysis”. **Essex, England. Longman Scientific and Technical**, 1995.

[BOCCARA, CHEONG, 1992] BOCCARA, N., CHEONG, K., “Automata Network SIR Models for Spread of Infectious Diseases in Populations of Moving Individuals”. **J. Phys. A: Math and Gen.**, v.25, p.2447-2461, 2000.

[CAVICCHIOLI NETO, RIBERTO, ALVES, 2003] CAVICCHIOLI NETO, V., RIBERTO, L., ALVES, D., “A Geographic Information System (GIS) to visualize the spatio-temporal patterns of the spread and control of dengue epidemics in the Ribeirão Preto city”. 1st International Conference on Bioinformatics and Computational Biology - (I COBICOBI), 2003, Ribeirão Preto. **Proceedings of 1st International Conference on Bioinformatics and Computational Biology - (I COBICOBI)**, 2003.

[CAVICCHIOLI, GAGLIARDI et al, 2006] CAVICCHIOLI NETO, V. GAGLIARDI, H. F., REZENDE, A. S., GALLO, E., GONCALVES, E. S., SILVA, F. A. B., ALVES, D. “Data Access Service in a Computational Grid Platform Applied to the Monitoring and Control of Epidemics on Georeferenced Dynamic Maps”. 2nd International IEEE Conference on e-science and the Grid, 2006, Amsterdam. **Proceedings of the 2nd International IEEE Conference on e-science and the Grid.**

[CHARTRAND 1985] CHARTRAND, G., “Introductory Graph Theory”, New York: **Dover Publications**, 1985.

[CLIFF, HAGGETT, SMALLMAN-RAYNOR 1993] CLIFF, A. D., HAGGETT, P., SMALLMAN-RAYNOR, M. “Measles: An Historical Geography of a Major Human Viral Disease, from Global Expansion to Local Retreat”, Oxford: **Blackwell**, p.1840-1990, 1993.

[CODEÇO, 2000] CODEÇO, C. T., “Modelagem Computacional de Processos Infeciosos”. **Escola de Verão em Métodos Computacionais em Biologia**, Petrópolis: LNCC, 2000.

[CZAJKOWSKI et al, 2004] CZAJKOWSKI, K., FERGUSON, D. F., FOSTER, I., FREY, J., GRAHAM, S., SEDUKHIN, I., SNELLING, D., TUECKE, S., VAMBENEPE, W., “The WS-Resource Framework”, (March 2004), disponível em: <http://www-106.ibm.com/developerworks/library/ws-resource/ws-wsrf.pdf>, acessado em: 02/02/2007.

[DEFANTI et al, 1996] DEFANTI, T., FOSTER, I., PAPKA, M., STEVENS, R., KUHFUSS, T. “Overview of the I-WAY: Wide-area visual supercomputing”. **Int. J. Supercomput. Appl. High Perfo. Comput.**, v.10, 2, p.123-131, 1996.

- [DEITEL, DEITEL, 2001] DEITEL, H. M.; DEITEL P. J., “Java: Como Programar”. **Editora BookMan**. 3a Edição. 2001.
- [EFROS, 1994] EFROS, A. L.; “Física y Geometria del Desorden”, **URSS**, 1994.
- [E-SCIENCE, 2006] E-SCIENCE, disponível em <http://www.nesc.ac.uk/>, acessado em 08/02/2007.
- [EUBANK et al, 2004] EUBANK S., GUCLU H., KUMAR, V. S. A, MARATHE M. V., SRINIVASAN A., TOROCZKAI Z.e WANG, N. “Modelling disease outbreaks in realistic urban social networks”. **Nature**,v. 429, 2004, 180.
- [EUBANK, 2002] EUBANK, S. “Scalable, Efficient, Epidemiological Simulation”. Los Alamos National Labs. *ACM* Copyright 2002, Disponível em <http://www.ccs.lanl.gov/ccs5/documents/pdf/2001/LA-UR-01-5513.pdf>, acessado em 15/04/2006.
- [FAFNER, 2006] FAFNER, disponível em <http://www.npac.syr.edu/factoring.html>, acessado em 12/07/2006.
- [FELIPE et al, 2004] FELIPE, J. C., REZENDE, A., CAVICCHIOLI NETO, V., BERARDO, B., GUELERI, W. L., ALVES, D. “Utilizando Mapas Dinâmicos Georeferenciados para Suportar Monitoramento e Análise de Informações de um Sistema Municipal de Saúde”, IX Congresso Brasileiro de Informática em Saúde - CBIS, 2004, Ribeirão Preto. **Anais do IX Congresso Brasileiro de Informática em Saúde - CBIS**. Sociedade Brasileira de Informática em Saúde – SBIS, 2004.

- [FENG e VELASCO-HERNÁNDEZ, 1997] FENG, Z., VELASCO-HERNANDEZ, J.X.,  
“Competitive exclusion in a vector-host model for the dengue fever”. **J. Mathematical Biology**, 35, 523-544, 1997.
- [FERREIRA, YANG, 2003] FERREIRA, C. P., YANG, H. M., “Estudo da transmissão da dengue entre indivíduos em interação com a população de mosquitos *Aedes aegypti*”. **TEMA – Seleção do XXV CNMAC**, (A. Sri Ranga, ed.), 2003
- [FOSTER et al, 2002] FOSTER, I., KESSELMAN, C., NICK, J., TUECKE, S. “The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration”. **Open Grid Service Infrastructure WG**, Global Grid Forum, 2002.
- [FOSTER, 2002] FOSTER, I. “What is the Grid? A Three Point Checklist” in **GRID Today**, 2002.
- [FOSTER, 2005] FOSTER, I. “Globus Toolkit Version 4: Software for Service-Oriented Systems”. IFIP International Conference on Network and Parallel Computing, **Springer-Verlag LNCS 3779**, pp 2-13, 2005
- [FOSTER, KESSELMAN, 1998] FOSTER, I., KESSELMAN C., “The Globus Project: A Status Report”. **Proc. IPPS/SPDP '98 Heterogeneous Computing Workshop**, pp.4-18, 1998.
- [FOSTER, KESSELMAN, TUECKE 2001] FOSTER, I.; KESSELMAN, C.; TUECKE, C. “The Anatomy of the Grid: Enabling Scalable Virtual Organizations”. In: **International J. Supercomputer Applications** - 15(3). [S.I.: s.n], 2001.
- [FOWLER, SCOTT 2000] FOWLER, M., SCOTT K., “UML Essencial: Um Breve Guia para a Linguagem Padrão de Modelagem de Objetos”. **Editora BookMan**. 2a Edição. 2000

[FURLAN 1998] FURLAN, J. D., “Modelagem de Objetos Através da UML”. São Paulo: **Campus**, 1998.

[FURLAN, GAGLIARDI, ALVES 2004] FURLAN, L. B., GAGLIARDI, H. F., ALVES, D., “Estudo do Conceito de Memória Imunológica em um Modelo para a Competição e Co-evolução de Microparasitas e o Sistema Imune de um Hospedeiro”. **REIC - Revista Eletrônica de Iniciação Científica**, v.III, 2004, XXIV Congresso da Sociedade Brasileira de Computação SBC 2004, Salvador–BA.

[FURLAN, GAGLIARDI 2006] FURLAN, L. B., GAGLIARDI, H. F., SILVA, F. A. B., PISA, I. T. ; ALVES, D. “MicroCASim: an automata network simulator applied to the competition between microparasites and host immune response” In: International Conference on Computational Science - Workshop on Modelling of Complex Systems by Cellular Automata, 2006, Reading. **Lecture Notes on Computer Science**, 2006.

[GGF, 2007] Global Grid Forum, disponível em <http://www.ogf.org>, acessado em 10/07/2007.

[GAGLIARDI, SILVA, ALVES, 2006] GAGLIARDI, H. F., SILVA, F. A. B., ALVES, D., “Automata networks applied to the epidemiology of urban dengue fever” in International Conference on Computational Science (ICCS 2006) – Workshop on Modelling of Complex Systems by Cellular Automata, 2006, Reading, UK. **Lectures Notes in Computational Science**. V.N. Alexandrov et al. (Eds.): ICCS 2006 Part III, LNCS 3993, p.297-304. Springer-Verlag Berlin Heidelberg 2006.

- [GAGLIARDI, ALVES, 2005] GAGLIARDI, H. F., ALVES, D. Redes Complexas e Modelagem de Epidemias. In: XXVIII Congresso Nacional de Matemática Aplicada e Computacional, 2005, SÃO PAULO. **Anais do XXVIII Congresso Nacional de Matemática Aplicada e Computacional**, 2005.
- [GALLO, 2006] GALLO, E. “GISE – Grid Integration Service”, Tese de Mestrado em Informática da Universidade Católica de Santos na linha de Sistemas Distribuídos, 2006.
- [GALLO et al, 2007] GALLO, E., GAGLIARDI, H. F., SILVA, F. A. B., CAVICCHIOLI NETO, V., ALVES, D., “GISE: A Data Access and Integration Service of Epidemiological Data for a Grid-Based Monitoring and Simulation System”. Trabalho aceito para apresentação oral na **40th Annual Simulation Symposium**, Norfolk, Virgínia-EUA, 2007.
- [GILLESPIE, 1976] GILLESPIE, D. T., “A general method for numerically simulating the stochastic evolution of coupled”, **J. Comp. Phys.**, 22, 403–434 (1976).
- [GILLESPIE, 1977] GILLESPIE, D. T., “Exact stochastic simulation of coupled chemical reactions”, **J. Chem. Phys.**, 81, 2340–2361 (1977).
- [GLOBUS, 2007] GLOBUS, disponível em <http://www.globus.org/toolkit/docs/4.0>, acessado em 25/03/2007.
- [GRIMSHAW et al, 1997] GRIMSHAW, A. S., WULF, W. A., T. L. Team, “The legion: vision of a worldwide virtual computer”. **Communications of the ACM**, vol. 40, no. 1, pp. 39–45, 1997.

[HAAS, CALIRI, SILVA, 1999] HAAS, V. J., CALIRI, A, SILVA, M. A. A. “Temporal Duration and Event Size Distribution at the Epidemic Threshold”. **Journal of Biological Physics**, Holanda, v. 25, p. 309-324, 1999.

[HEALTHGRID Project, 2006] HEALTHGRID Project, disponível em <http://www.healthgrid.org/en/about-us/initiative.html>, acessado em 10/12/2006.

[HOLMES, 1997] HOLMES, E.E., “Basic Epidemiological concepts in a spacial context”. *Spacial Ecology: The role of space in population dynamics and interspecific interactions*, Princeton: **Princeton University Press**, 1997.

[HUANG et al, 2005] HUANG C.Y., SUN C.T., HSIEH J.L., CHEN Y.M.A., LIN H. “A Novel Small-World Model: Using Social Mirror Identities for Epidemic Simulations”. **Simulation: Transactions of The Society for Modeling and Simulation International**, 81(10), pp. 671-699, (SCI, NSC-92-2524-S-009-004), 2005.

[JACOB et al, 2006] JACOB, B., BROWN, M., FUKUI, K., TRIVEDI, N., “Introduction to Grid Computing”, disponível em: <http://www.redbooks.ibm.com/redbooks/pdfs/sg246778.pdf>, acessado em: 02/08/2006.

[KEELING, GRENFELL 2000] KEELING, M. J., GRENFELL, B. T., “Individual-based perspectives on  $R(0)$ ”. **J. Theor. Biol.**, 203, 51, 2000.

[KERMACK, MACKENDRICK, 1927] KERMACK, W. O., MACKENDRICK, A. G., “A contribution to the mathematical theory of epidemics”. **Proc. R. Soc. Lond. A**, v.115, p. 700, 1927.

- [KLECZKOWSKI, GRENFELL 1999] KLECZKOWSKI, A., GRENFELL, B. T., “Mean-Field-type equations for spread of epidemics: The ‘small world’ model”. **Physica A**, v.274, p.355-360, 1999.
- [LUZ et al, 2003] LUZ, P.M., CODEÇO, C.T., MASSAD, E, STRUCHINER, C.J.. “Uncertainties Regarding dengue Modeling In Rio De Janeiro, Brazil”. **Mem. Inst. Oswaldo Cruz**. p.871-878, 2003.
- [MIZRAHI 1994a] MIZRAHI, V. V., “Treinamento em Linguagem C++ - Módulo 1”. **Makron Books**. 1994.
- [MIZRAHI 1994b] MIZRAHI, V. V., “Treinamento em Linguagem C++ - Módulo 2”. **Makron Books**. 1994.
- [MUCHA et al, 2005] MUCHA, M. ; RIBERTO, L. ; SERAFIM, R. C. ; CAVICCHIOLI NETO, V. ; GAGLIARDI, H. F. ; ALVES, D. ., “Análise Espaço-Temporal de um Mapa Ambiental Dinâmico com os Casos de dengue em Ribeirão Preto (SP)”. **Saúde e Sociedade**, v. 14, n. 1, p. 263, 2005.
- [MURRAY, 1993] MURRAY, J. D., “Mathematical Biology”. Seattle: **Springer**, v.19, p.610-618, 1993.
- [ODELL, MARTIN 1996] ODELL, J., MARTIN, J., “Análise e Projeto Orientados a Objeto”. São Paulo: **Makron Books**, 1996.
- [OLIVEIRA 1999] OLIVEIRA, G. M. B.; “Dinâmica e Evolução de Autômatos Celulares Unidimensionais”, 220p., Tese de Doutorado em Ciência do Curso de Engenharia Eletrônica e Computação na Área de Informática, Instituto Tecnológico de Aeronáutica – ITA, 1999.

[OGSA-DAI, 2006] OGSA-DAI (Open Grid Services Architecture Data Access and Integration), disponível em <http://www.ogsadai.org.uk>, acessado em 03/06/2006.

[PATERSON, VINEY, 2000] Paterson, S. and Viney M. E. “The interface between epidemiology and population genetics”, **Parasitology Today**, Vol. 16, no. 12, pp. 528, 2000.

[PESSOA, GAGLIARDI, ALVES 2004] PESSOA, T. R. M., GAGLIARDI, H. F., ALVES, D., “Desenvolvimento de Autômato Celular Probabilístico para Estudar a Transmissão e Espalhamento de dengue”. **Anais do XXIV Congresso da Sociedade Brasileira de Computação (SBC 2004)**, v.1. p.319 – 331, 2004, XXIV Congresso da Sociedade Brasileira de Computação (SBC 2004), Salvador – BA.

[REZENDE, GAGLIARDI et al, 2006] REZENDE, A. S., GAGLIARDI, H. F., SILVA, F. A. B., ALVES, D. “IntegraEpi-GIS: A Geographic Information System to visualize and analyze the spatio-temporal patterns of the spread and control of epidemics”. Spatial Epidemiology Conference, 2006, Londres. **Proceedings Spatial Epidemiology Conference**, 2006.

[RIBERTO et al, 2005] RIBERTO, L. ; MUCHA, M. ; SERAFIM, R. C. ; CAVICCHIOLI NETO, V. ; GAGLIARDI, H. F. ; ALVES, D. . Construção de indicadores Ambientais de risco de ocorrência e disseminação da dengue em mapas dinâmicos de Ribeirão Preto. **Saúde e Sociedade**, v. 14, n. 1, p. 264, 2005.

[RHODES, ANDERSON 1996] RHODES, C. J., ANDERSON, R. M., “Power laws governing epidemics in isolated populations”, **Nature**, v.381, p.600-602, 1996.

- [ROURE et al, 2003] ROURE, D. D., BAKER, M. A., JENNINGS, N. R., SHADBOLT, N. R.. “The Evolution of the Grid”. Grid Computing – Making the Global Infrastructure a Reality, 2003.
- [RUMBAUGH, BOOCH, JACOBSON 2000] RUMBAUGH, J., BOOCH, G; JACOBSON, I., “UML: Guia do usuário”. São Paulo: **Makron Books**, 2000.
- [SILVA, GAGLIARDI et al, 2007] SILVA, F. A. B., GAGLIARDI, H. F., GALLO, E., CAVICCHIOLI NETO, V. ; PISA, I. T. ; ALVES, D. “IntegraEPI: a Grid-based Epidemic Surveillance System”. Healthgrid 2007 Conference, 2007, Genebra. Proceeding of Healthgrid 2007 (accepted for publication). Amsterdã : **IOS Press**, 2007.
- [SINAN, 2007] SINAN (Sistema de Informação de Agravos de Notificação), disponível em <http://dtr2004.saude.gov.br/sinanweb/index.php>, acessado em 03/02/2007.
- [TUECKE et al, 2003] TUECKE, S.; CZAJKOWSKI, K.; FOSTER, I.; FREY, J.; GRAHAM, S.; KESSELMAN, C.; MAQUIRE, T.; SANDHOLM, T.; SNELLING, D.; VANDERBILT, P. “Open Grid Services Infrastructure Specification”. [S.I.: s.n], 2003. Disponível em: <http://xml.coverpages.org/OGSI-SpecificationV110.pdf>. Acessado em: 20/11/2005. 2003.
- [WATTS, STROGATZ, 1998] WATTS, D. J., STROGATZ, S. H., “Collective Dynamics of ‘Small-World’ Networks”, **Nature**, v.393, p.440,1998.
- [WATTS, STROGATZ 2004] WATTS, D. J.; STROGATZ, W. M., “Small Worlds: The Dynamics of Networks”, **Princeton**, New Jersey: Princeton University Press, 2004.

[YANG, 2003] H. M. Yang, “Epidemiologia da transmissão da dengue”. **TEMA - Seleta do XXV CNMAC**, (A. Sri Ranga, ed.), 2003.