

UNIVERSIDADE FEDERAL DE PERNAMBUCO

DEPARTAMENTO DE ENERGIA NUCLEAR

COMISSÃO NACIONAL DE ENERGIA NUCLEAR

CENTRO REGIONAL DE CIÊNCIAS NUCLEARES DO NORDESTE

Programa de Pós-graduação em Tecnologias Energéticas e Nucleares

**DESENVOLVIMENTO DE UM SOFTWARE PARA MODELAGEM DE
TOMÓGRAFOS POR EMISSÃO DE PÓSITRONS**

Igor Fagner Vieira

Orientador: Dr. Fernando Roberto de Andrade Lima

Co-Orientador: Dr. José Wilson Vieira

RECIFE, PE

Março de 2013

Igor Fagner Vieira

**DESENVOLVIMENTO DE UM SOFTWARE PARA MODELAGEM DE
TOMÓGRAFOS POR EMISSÃO DE PÓSITRONS**

Dissertação submetida ao Programa de Pós-Graduação em Tecnologias Energéticas e Nucleares, para obtenção do título de Mestre em Ciências. Área de Concentração: Dosimetria e Instrumentação Nuclear.

Orientador: Dr. Fernando Roberto de Andrade Lima

Co-Orientador: Dr. José Wilson Vieira

RECIFE, PE

Março de 2013

Catálogo na fonte

Bibliotecário Carlos Moura, CRB-4 / 1502

V658d

Vieira, Igor Fagner.

Desenvolvimento de um software para modelagem de tomógrafos por emissão de pósitrons. / Igor Fagner Vieira. - Recife: O Autor, 2013.

102 folhas, il., figs., tabs.

Orientador: Prof. Dr. Fernando Roberto de Andrade Lima.

Co-orientador: Prof. Dr. José Wilson Vieira.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Tecnologias Energéticas e Nucleares, 2013.

Inclui Referências e Apêndice.

1. GATE. 2. Python. 3. PET scanner. 4. Modelos

FOLHA DE APROVAÇÃO

DESENVOLVIMENTO DE UM SOFTWARE PARA MODELAGEM DE TOMÓGRAFOS POR EMISSÃO DE PÓSITRONS

IGOR FAGNER VIEIRA

APROVADO EM: 04/03/2013

ORIENTADOR: Prof. Dr. Fernando Roberto de Andrade Lima

CO-ORIENTADOR: Prof. Dr. José Wilson Vieira

COMISSÃO EXAMINADORA:

Prof. Dr. Fernando Roberto de Andrade Lima –CRCN-NE/CNEN

Prof.. Dr. Daniel Alexandre Batista Bonifácio – IRD/CNEN

Prof. Dr. Borko Stosick – UFRPE

Profa. Dra. Mércia Liane – CRCN-NE/CNEN

Visto e permitida a impressão

(Profa. Dra. Helen Jamil Khoury)
Coordenadora do PROTEN/DEN/UFPE

AGRADECIMENTOS

A Deus, que não este antropomórfico pintado pelas religiões seculares;

Aos meus pais, Alda Vieira e Marcondes Flores, pela Vida e por tudo o mais- e não poderia esquecer, as minhas tias-mãe Nair e Socorro que sempre exalaram o valor pelo conhecimento;

A minha namorada e companheira, Nazaré Fonseca, pela paciência, companheirismo e entusiasmo pela vida, elementos importantíssimos para o desenvolvimento deste trabalho;

Aos meus amigos, vivos e mortos, que sempre estiveram de alguma maneira por perto oferecendo muito mais que uma mão amiga, cujos nomes não caberiam nesta página, em especial: Paulo Vieira, Guilherme Amorim, amigos de longa data, com quem ao longo desses anos muito venho aprendendo;

Aos meus orientadores, Dr. Fernando Roberto de Andrade Lima e Dr. José Wilson Vieira não só pela orientação que tanto vem contribuindo para a minha formação científica, mas sobretudo pelo companheirismo durante o desenvolvimento deste trabalho, regado a muitas risadas e bom-humor em nossos momentos de descontração na hora do café ou do bar, ou mesmo no meio do trabalho. Estes momentos vividos me fazem pensar que a realização pessoal e grupal exige uma boa dose de humildade, leveza e paciência...

A todos amigos do CRCN-NE, da graduação em Física, do IFPE e do GDN- em especial ao Marcelo Santana por me iniciar no Linux, ao Alex Oliveira, ao Paulo Raully, ao Leanderson Pereira e claro, aos amigos Lindeval Fernandes, José de Melo, Viriato Leal, Rosa Chaves, Geneci Calado e Jonathan A. Melo pelo apoio e incentivo de sempre;

Aos professores do DEN Dr. Vinicius Saito, Dra. Helen Khoury e Dr. Vagner Cassola, pela abertura e disposição sempre demonstradas: algumas de nossas conversas nos corredores do DEN e/ou por email foram importantes em minha formação; À Profa. Dra. Mércia Liane de Oliveira, pelas sugestões durante o desenvolvimento deste trabalho.

Ao pessoal do IRD-RJ, especialmente a Dra. Lídia Vasconcelos Sá pela acolhida, colaboração e ao Dr. Daniel Bonifácio pela amizade e suporte técnico-científico, que muito vêm contribuindo para o desenvolvimento deste trabalho;

Ao CRCN-NE, à CNEN e FACEPE pelo apoio financeiro;

Ao IFPE, onde vive bons momentos;

À Universidade Federal de Pernambuco e ao Departamento de Energia Nuclear.

"[...]Fazer é tanto mistério e suor... [...]Por mais, eu sei, por mais que voemos neste mar, sempre há uma distância a cumprir-se! [...]"(Trecho do Livro "Frutos de Arribação" do também poeta, Wilson Vieira)

Por essa razão,

"[...]Precisamos ser capazes de olhar para o *nosso trabalho* com *Alegria, Satisfação e Simplicidade*."
(Richard Feynman)

RESUMO

DESENVOLVIMENTO DE UM SOFTWARE PARA MODELAGEM DE TOMÓGRAFOS POR EMISSÃO DE PÓSITRONS

Há uma tendência cada vez mais crescente na comunidade científica, ou mesmo dentro das grandes empresas da área médica, de utilizar códigos de transporte das radiações para validar resultados experimentais ou mesmo para projetar novos experimentos e/ou equipamentos. Neste trabalho, um método para modelagem de tomógrafo por emissão de pósitrons utilizando o GATE (*Geant4 Application for Tomographic Emission*) foi proposto e inicialmente validado. O GATE é uma plataforma internacionalmente reconhecida e utilizada para desenvolvimento de Modelos Computacionais de Exposição (MCE) no contexto da Medicina Nuclear, embora atualmente hajam módulos dedicados para aplicações em Radioterapia e Tomografia Computadorizada (TC). O GATE usa métodos Monte Carlo (MC) e tem uma linguagem de *script* própria. A escrita dos *scripts* para simulação de um PET *scanner* no GATE envolve um conjunto de passos interligados, sendo a acurácia da simulação dependente do arranjo correto das geometrias envolvidas, já que os processos físicos dependem destas, bem como da modelagem da eletrônica dos detectores no módulo *Digitizer*, por exemplo. A realização manual desse *setup* pode ser fonte de erros, sobretudo para usuários que não tenham experiência alguma no campo das simulações ou familiaridade prévia com uma linguagem de programação, considerando também o fato de todo este processo de modelagem no GATE ainda permanecer vinculado ao terminal do LINUX/UNIX, um ambiente familiar apenas para poucos. Isso se torna um obstáculo para iniciantes e inviabiliza o uso do GATE por uma gama maior de usuários, interessados em otimizar seus experimentos e/ou protocolos clínicos por meio de um modo mais acessível, rápido e amigável. O objetivo deste trabalho consiste, portanto, em desenvolver um software amigável para modelagens de Tomógrafos por Emissão de Pósitrons, chamado GUIGATE (Graphical User Interface for GATE), com módulos específicos e dedicados a controle de qualidade em PET *scanners*. Os resultados obtidos exibem os recursos disponíveis no GUIGATE, presentes em um conjunto de janelas que permitem ao usuário criar seus arquivos de entrada (os *inputs*), executar e visualizar em tempo real o seu modelo, bem como analisar seu arquivo de saída (os *outputs*) em um único ambiente, viabilizando assim de modo intuitivo o acesso a toda a arquitetura de simulação do GATE e ao analisador de dados do CERN, o ROOT.

PALAVRAS CHAVES: GATE, Python, PET scanner, Modelos Computacionais de Exposição, Monte Carlo.

ABSTRACT

SOFTWARE DEVELOPMENT FOR MODELING POSITRONS EMISSION TOMOGRAPH SCANNERS

There is an ever increasing trend in the scientific community, and among large medical companies, for the use radiation transport codes to validate experimental results and to design new experiments and/or equipments. In this dissertation, a method for modeling of positron emission tomography using the GATE (Geant4 Application for Tomographic Emission) was proposed and validated. GATE is an international platform recognized and used to develop Computational Model Exposure (CME) in the context of Nuclear Medicine, although currently there are dedicated modules for applications in Radiotherapy and Computed Tomography (CT). GATE uses Monte Carlo (MC) methods, and has a scripting language of its own. The writing of scripts for simulation of a PET scanner in GATE involves a number of interrelated steps, and the accuracy of the simulation is dependent on the correct setup of the geometries involved, since the physical processes depend on them, as well as the modeling of electronic detectors in module *Digitizer*, for example. The manual implementation of this setup can be a source of errors, especially for users without experience in the field of simulations or without any previous knowledge of a programming language, and also due to the fact that the modeling process in GATE still remains bounded to LINUX / UNIX based systems, an environment only familiar to a few. This becomes an obstacle for beginners and prevents the use of GATE by a larger number of users interested in optimizing their experiments and/or clinical protocols through a more accessible, fast and friendly application. The objective of this work is therefore to develop an user-friendly software for the modeling of Positron Emission Tomography called GUIGATE (Graphical User Interface for GATE), with specific modules dedicated to quality control in PET scanners. The results exhibit the features available in this first version of GUIGATE, present in a set of windows that allow users to create their input files, run and display in real time the model and analyze its output file in a single environment, allowing thus intuitive access to the entire architecture of the GATE simulation and to CERN's data analyzer, the ROOT.

KEY-WORDS: GATE, Python, PET scanner, Computational Models of Radiation Exposure (MCE), Monte Carlo.

LISTA DE FIGURAS

Figura 1 -	Visão do princípios básicos na geração da imagem PET.	18
Figura 2 -	(a) Diagrama básico do circuito coincidência (colimação eletrônica em PET). Os cristais cintiladores estão conectados a amplificadores (Amp) e analisadores de pulsos (AP) individuais. Quando um fóton interage em cada cristal, os sinais são amplificados e analisados para determinar se a energia está dentro de um certo limite inferior e superior. Se o critério de energia é satisfeito, um pulso é gerado pelo AP. Esses pulsos atingem o módulo de coincidência, se estiverem dentro do período 2τ , trata-se de uma coincidência; (b) Principais diferenças entre PET, SPECT e Gama Câmara.	19
Figura 3 -	Transdução de informação em um cristal cintilador.	20
Figura 4 -	Em PET, a probabilidade de detecção P_{detect} em uma aniquilação para uma dada linha de resposta entre dois detectores D_1 e D_2 independe do local atual onde ocorreu a aniquilação ao longo da linha de resposta. . .	22
Figura 5 -	Variação do coeficiente de atenuação mássico total (cm^2/g), em preto, para água (H_2O).	23
Figura 6 -	Variação do coeficiente de atenuação mássico total (cm^2/g), em preto, para o Óxido de Bismuto (BGO).	23
Figura 7 -	Representação dos efeitos (a) Compton e (b) Fotoelétrico.	25
Figura 8 -	(a) Diferentes tipos de eventos coincidências em PET e (b) Efeito qualitativo no contraste da imagem devido a uma redução na contagem de taxas de eventos aleatórios e espalhados.	27
Figura 9 -	(a) Fantoma NEMA e suas dimensões em uma visão lateral e axial. A fonte linear fica a 45 ± 1 mm abaixo do centro do fantoma. (b) Fantoma sendo posicionado no centro do campo de visão (FOV) para realização do teste.	29
Figura 10 -	Ilustra a influência de parâmetros geométricos (tamanho e posição do fantoma NCAT) nas curvas NECR, em função da atividade injetada (MBq). . .	30
Figura 11 -	Estrutura do GATE mostrando sua hierarquia funcional e áreas de aplicação.	32
Figura 12 -	Fantoma cilíndrico e um sistema <i>CylindricalPET</i> composto de 5 <i>rsectors</i> , 4 <i>modules</i> (repetidos ao longo do eixo Z), 3 <i>submodules</i> (repetidos ao longo do eixo Y), 64 <i>crystals</i> (8 x 8) and 2 <i>layers</i> (vermelho e amarelo). . .	33

Figura 13 -	Ilustra de modo simplificado a árvore de comandos fundamentais para a construção de um <i>cylindricalPET</i> . Os termos exibidos nesta figura se referem a estruturas do GATE, conforme consta no manual de usuários (USER'SGUIDE, 2012)	34
Figura 14 -	Parte do script do arquivo "scanner.mac" para modelar um PET <i>scanner</i> baseado no GE Advance, mostrando o conjunto de comandos a serem manualmente definidos. Exemplo completo no Apêndice A.	35
Figura 15 -	Mostra um script no arquivo " <i>phantom.mac</i> " para modelar um fantoma cilíndrico baseado no NEMA 2007 (fantoma de 20 cm de diâmetro, branco). Os outros dois são variações realizadas ao alterar o parâmetro mostrado na seta vermelha (Linha 17).	36
Figura 16 -	Mostra um script no arquivo "physics.mac" para modelar os processos físicos de interesse.	38
Figura 17 -	(a) Canais de processamento do módulo <i>digitizer</i> que organiza as <i>hits</i> (interações) para <i>singles</i> (eventos simples) e (b) As ações dos módulos <i>adder</i> e <i>readout</i> . O <i>adder</i> transforma os <i>hits</i> em um pulso dentro de cada volume individual, e então o <i>readout</i> soma um grupo desses pulsos formando um <i>single</i> no nível de volume (o chamado <i>depth</i> no GATE definido pelo usuário). (c) O papel dos processadores de pulso na geração dos eventos coincidência e simples, bem como a pilha de dados de saída representada pelo gerenciador de saída de dados	40
Figura 18 -	Mostra um script no arquivo "digitizer.mac" para modelar a eletrônica dos detectores.	41
Figura 19 -	Mostra um script no arquivo "output.mac" definindo o arquivo ROOT como saída padrão.	42
Figura 20 -	Síntese da arquitetura geral de simulação do GATE sob a forma de um passo a passo.	43
Figura 21 -	Arquitetura do <i>software</i> GUIGATE. Este foi criado afim de facilitar a interação do usuário com o GATE e com o ROOT, passando esse processo do terminal do Linux/Unix para uma interface gráfica. A classe <i>wxPython</i> fornece os métodos para criação das estruturas do software. As funcionalidades destas foram implementadas numa biblioteca de classe própria, a <i>GUIGATEClass</i> , toda baseada na arquitetura de simulação do GATE.	45
Figura 22 -	Diagrama Geral da Classe (<i>GUIGATEClass.py</i>) do <i>software</i> GUIGATE. Na superclasse <i>GUIGATESystems</i> apenas a subclasse <i>CylindricalPET</i> com seus métodos específicos foram implementados para essa primeira versão. As demais (<i>Scanner</i> , <i>CPET</i> , <i>OPET</i> , etc.) serão implementadas em versões futuras.	46
Figura 23 -	Árvore de <i>Menus</i> da janela principal. O menu " <i>IMAGE TOOLS</i> " aqui exibido não está com funções implementadas nesta primeira versão do GUIGATE.	47

Figura 24 - Diagrama de blocos e parte do código implementado para construção dos menus.	48
Figura 25 - Código implementado que permite o acesso a janela de criação dos arquivos de entrada, a qual deriva da classe-mãe " <i>MainFrame</i> ".	49
Figura 26 - Código implementado que permite a execução do GATE, quando selecionada a macro principal.	50
Figura 27 - Princípio geral de acesso a janela <i>GUIGATE Output Module</i>	51
Figura 28 - Diagrama geral implementado na <i>GUIGATEClass</i> para exibir na janela <i>GUIGATE Output Module</i> a perspectiva de análise selecionada: <i>NECR</i> , <i>Axial Sensitivity</i> , <i>Scatter Fractional Axial Distribution</i> , <i>Energy Distribution</i> ou <i>Sinogram2D</i>	52
Figura 29 - Janela Principal do GUIGATE mostrando o Menu <i>TOOLS</i> para acessar aos módulos de construção dos setups de entrada no GUIGATE, tais como: Scanner, Phantom, Physics etc.	54
Figura 30 - Janela de Inputs do GUIGATE.	55
Figura 31 - Mostrando a construção do <i>world</i> na janela de construção do <i>scanner</i>	55
Figura 32 - Janela de Construção do <i>scanner</i> , mostrando construção da geometria do sistema.	56
Figura 33 - Janela de Construção do <i>scanner</i> , mostrando a construção dos cinco níveis hierárquicos (<i>Rsector</i> , <i>Module</i> , <i>SubModule</i> , <i>Crystal</i> , <i>Layer</i>) que compõem o sistema.	57
Figura 34 - Janela de Construção do <i>scanner</i> , mostrando a construção de repetições de geometrias.	58
Figura 35 - Janela de Construção do <i>scanner</i> , mostrando (superior) a construção de <i>attach</i> de geometrias e (inferior) o arquivo " <i>Scanner.mac</i> " com parte dos parâmetros definidos na construção da geometria e salvos no mesmo.	59
Figura 36 - Janela de Construção dos Fantomas Cúbicos e Cilíndricos.	60
Figura 37 - Mostra arquivo " <i>Phantom.mac</i> " e parte dos parâmetros definidos na construção de um fantoma cilíndrico baseado no fantoma NEMA padrão.	61
Figura 38 - Janela de Construção dos possíveis processos físicos no GATE.	61
Figura 39 - Janela opcional aberta quando o usuário seleciona um material no "combobox" (" <i>set Materials</i> ") do campo " 3- Options ... ". Ao lado, alguns recursos de edição dos gráficos.	62
Figura 40 - Módulo Digitizer.	63
Figura 41 - Submódulo Add no GUIGATE.	64
Figura 42 - Submódulo AddCompton no GUIGATE.	64
Figura 43 - Submódulo Readout no GUIGATE.	65

Figura 44 - Submódulo "Energy Blurring"no GUIGATE, com os tipos de distribuição de probabilidade disponíveis.	66
Figura 45 - Submódulo " <i>THRESHOLDER/UPHOLDER</i> ", no GUIGATE. Parâmetros cruciais para reproduzir taxa de contagens e fração de espalhamento em simulações realistas de PET <i>scanners</i>	67
Figura 46 - Submódulo "Dead Time"no GUIGATE. Permite simular o tempo morto do sistema de detectores, ou seja, o tempo que o sistema de detecção fica inativo até estar apto para a subsequente detecção.	67
Figura 47 - Submódulo "Coincidences Sorter"no GUIGATE.	68
Figura 48 - Janela de Construção da Fonte: mostra diferentes opções projetadas para GUIGATE. Habilitada para esta versão apenas os emissores de pósitrons: Flúor-18 (F^{18}), Carbono-11(C^{11}), Oxigênio-15 (O^{15}).	70
Figura 49 - Janela de Construção da Fonte emissora de pósitrons: F^{18} , C^{11} , O^{15} . . .	70
Figura 50 - Janelas de simulação de radionuclídeos no GUIGATE. Na parte superior a modelagem de íons. Na parte inferior, a modelagem de fotons gamma com angulos de 180° graus.	71
Figura 51 - Input de entrada no GATE gerado no GUIGATE, que simula uma fonte cilíndrica emissora de pósitrons reproduzindo o F^{18}	71
Figura 52 - Caminho para definição do tipo de arquivo de saída no GUIGATE. . . .	72
Figura 53 - Etapas finais de construção do experimento no GUIGATE.	73
Figura 54 - Executando um experimento no GUIGATE: selecionando o arquivo na janela " <i>Choose your Main GATE Macro File</i> ".	74
Figura 55 - Opções de <i>menus</i> disponíveis para análises de arquivo ROOT.	75
Figura 56 - <i>Browser</i> do ROOT que é exibida ao clicar no <i>menu</i> " <i>Open ROOT Browser...</i> ". Cada <i>menu</i> deste <i>browser</i> tem uma série de funcionalidades nativas do ROOT.	75
Figura 57 - (a) Árvore gerada na simulação do GATE (Superior). (b) Distribuição de contagens da fonte ao longo do eixo-Z no modo padrão. (c) Mesma imagem que a anterior, exibida em um dos modos nativos de exibição de histograma no ROOT, o "Lego2"(3D)	76
Figura 58 - Janela geradora do NECR e outros parâmetros importantes em PET. . .	77
Figura 59 - Janela exibe histograma 2D da sensibilidade axial do <i>scanner</i> . Na "Tab"ao lado do gráfico é possível ver alguns recursos de edição de gráfico. . . .	78
Figura 60 - Janela exibe histograma 2D da fração de espalhamento axial do <i>scanner</i> . .	79
Figura 61 - Janela exibe histograma 2D da distribuição energética dos diferentes tipos de eventos coincidências verdadeira (vermelha), espalhada (azul) e total(amarela).	80

Figura 62 - Janela exibe sinograma 2D e distribuição espacial (3D) dos diferentes eventos registrados na matriz de detectores. Essa janela permite interações com rotação etc.	80
Figura 63 - Compara os resultados obtidos de modo automatizado com o GUIGATE e aqueles com o GATE, em seu modo padrão. Os resultados são exibidos para: <i>Transaxial Detection Position</i> e <i>Axial Detection Position</i>	92
Figura 64 - Compara os resultados obtidos de modo automatizado com o GUIGATE e aqueles com o GATE, em seu modo padrão. Os resultados são exibidos para: <i>Axial sensitivity</i> e <i>Axial scatter fraction</i>	93

LISTA DE TABELAS

Tabela 1 -	Propriedades físicas dos cristais cintiladores utilizados em PET.	21
Tabela 2 -	Interações possíveis de ocorrer para radiação eletromagnética ionizante e as consequências das interações.	26
Tabela 3 -	Recursos disponíveis nos códigos dedicados a simulações PET e SPECT estudados.	31
Tabela 4 -	Níveis Hierárquicos do <i>CylindricalPET</i>	34
Tabela 5 -	Tipos de arquivos de saída (padrão e específicos) dos diferentes sistemas para modelagem PET.	41
Tabela 6 -	Relação entre os componentes físicos do scanner e o nível hierárquico do <i>CylindricalPET</i>	53
Tabela 7 -	Diferentes modelos utilizados no GATE para simular múltiplas coincidências: descrição de cada um.	69
Tabela 8 -	Comparação dos resultados médios obtidos modelando o <i>benchPET</i> , usando método padrão no GATE e o GUIGATE.	81

SUMÁRIO

	Página
1 INTRODUÇÃO	14
2 REVISÃO DA LITERATURA	16
2.1 A TOMOGRAFIA POR EMISSÃO DE PÓSITRONS (PET) NO CONTEXTO DO DIAGNÓSTICO POR IMAGEM	16
2.2 PRINCÍPIO DE FUNCIONAMENTO DA PET	16
2.3 DETECTORES EM PET	20
2.4 PROCESSOS FÍSICOS EM PET	22
2.4.1 Principais Processos de Interação da Radiação com a Matéria em PET	22
2.4.2 Eventos Coincidências e a sua Influência na Imagem e Otimização da Dose	26
2.5 CÓDIGOS MONTE CARLO DEDICADOS À MODELAGEM PET	30
2.5.1 A Plataforma GATE: uma visão geral.	30
2.5.1.1 Arquitetura de Simulação do GATE	31
2.5.1.2 Inputs e Execução de um Modelo	32
3 MATERIAIS E MÉTODOS	44
3.1 HARDWARE E SOFTWARE	44
3.2 LINGUAGEM E FERRAMENTAS	44
3.3 AS INTERFACES	46
3.3.1 Principal	46
3.3.2 Secundárias:	48
3.3.2.1 Inputs	48
3.3.2.2 Outputs	51
3.4 PET SCANNER A SER SIMULADO	52
4 RESULTADOS	54
4.1 INPUTS NO GATE VIA GUIGATE	54
4.1.1 Passo 1: Acessando a Janela de Inputs	54
4.1.2 Passo 2: Construindo a Geometria do <i>Scanner</i>	55
4.1.3 Passo 3: Construindo a Geometria do <i>Fantoma</i>	60

4.1.4	Passo 4: Construindo Processos Físicos	61
4.1.5	Passo 5: Construindo a Eletrônica dos Detectores	63
	Adder/Addercompton	63
	Readout	64
	Blurring	64
	Thresholder/Upholder	66
	Dead Time	67
	Coincidences Sorter	68
4.1.6	Passo 6: Construindo a Fonte Emissora de Pósitrons	69
4.1.7	Passo 7: Definindo os Arquivos de Saídas	72
4.1.8	Passo 8: Definindo o <i>Start-Up</i> do Experimento	73
4.1.9	Passo 9: Executando o GATE via GUIGATE	73
4.2	OUTPUTS NO GATE VIA GUIGATE	74
5	CONCLUSÕES E PERSPECTIVAS	82
	REFERÊNCIAS	83
6	APÊNDICES	86
6.1	APÊNDICE A: SCRIPTS DE UM PET SCANNER CLÍNICO	86
6.2	APÊNDICE B: VALIDAÇÃO INICIAL DO GUIGATE	92

1 INTRODUÇÃO

A medicina moderna dispõe de uma ampla gama de recursos em sua busca de curar os mais diversos males que afetam a saúde humana. Na área do diagnóstico por imagem, os grandes avanços observados são resultados da combinação da biologia molecular e da ciência de formação da imagem (ZAID, 2003).

Nesse sentido, destacam-se os tomógrafos por emissão tipo SPECT (Single Photon Emission Computed Tomography) e PET (Positron Emission Tomography) (SÁ, 2010). No Brasil, atualmente, registra-se algo em torno de 58 tomógrafos emissores de pósitrons acoplados a tomógrafos computadorizados (TC), com perspectiva de crescer esse número dada a quantidade de ciclotrons, 15 no total, produzindo a principal matéria prima do PET, o emissor de pósitron flúor-18 (F^{18}) (<http://www.crcn.gov.br/>).

O PET utiliza como princípio básico blocos de detectores associados a todo um aparato eletrônico (amplificadores, discriminadores de altura de pulso, espaço e tempo) e a um circuito, também chamado circuito coincidência, que recebem a saída desse aparato e discriminam, utilizando sobretudo discriminadores de energia e tempo, os fótons advindos do mesmo par provenientes de diferentes tipos de eventos coincidência. Esse tipo de colimação eletrônica torna possível localizar o ponto onde ocorreu a aniquilação do par pósitron-elétron e a direção dos fótons de 511 keV.

A imagem final obtida nesse processo, além das informações fisiológicas da região examinada, reflete a qualidade do funcionamento de todo sistema de aquisição. Naturalmente, a otimização deste funcionamento influi na eficácia do diagnóstico PET e pode representar uma redução de dose injetada no paciente (WATSON et al., 2005).

O desafio no uso de tal tecnologia, consiste, por um lado, na tentativa de criar padronização de desempenho dos diferentes tipos de equipamentos, bem como definição de testes e controles mínimos, tal como proposto por Sá e colaboradores (2010). Na ausência de um consenso internacional para realização dos testes de aceitação dos PETs, os usuários seguem as recomendações dos protocolos NEMA (National Electrical Manufacturers Association) e/ou da IEC (International Electrotechnical Commission), sendo mais difundido, no Brasil, o uso do NEMA. Tais protocolos preconizados pela NEMA e IEC apresentam, de um modo geral, similaridades. As principais diferenças entre esses dois guias (NEMA e IEC) reside nos simuladores, procedimentos de aquisição e nos métodos de processamento (NEMA, 2007, 2008; IEC, 1998). Ambos têm o objetivo de assegurar que o equipamento se encontra de acordo com as especificações do fabricante, que é quem atualmente estipula os níveis de referência para seu equipamento. Os testes se baseiam no fato da tomografia por emissão de pósitrons ser uma tecnologia que usa isótopo radiativo de meia-vida curta (até 2h). Entre estes testes, está a estimativa do NECR (*Noise Equivalent Count Rate*), utilizado para avaliação de desempenho do sistema de detecção, com a indicação de ser realizado como o primeiro dos testes no caso de um novo equipamento ou ainda a cada nova alteração realizada no mesmo do ponto de vista de hardware e software. Atualmente este testes é realizado com um fantoma cilíndrico padrão (NEMA, 2008) com 20 cm de diâmetro e 70 cm de altura, dentro do qual é inserida uma fonte emissora de pósitrons também cilíndrica.

Do ponto de vista da rotina clínica, este é um aparato fácil de ser montado, o que torna a realização do teste algo prático. Por outro lado, alguns estudos levantam os limites da utilização dessa métrica no que diz respeito a sua aplicabilidade como um preditor único da qualidade da imagem PET, pelo fato da NECR não levar em conta, por exemplo, o impacto do algoritmo de reconstrução, os efeitos da resolução espacial (BADAWI, 2005). Além disso, outros estudos têm apontado que o NECR varia para um único paciente em função, por exemplo, da posição da mesa durante uma aquisição de corpo inteiro (WATSON et al., 2005; NICOLAS et al., 2008). Isso justifica estudos desse parâmetro com variações geométricas, como por exemplo, fantasmas de diâmetros maiores que 20 cm (RUTH et al., 2007), considerando que é preciso levar em conta os pacientes obesos, bem como a possibilidade de usar o NECR para construir uma função específica para pacientes individuais como proposto por Nicolas *et al* (2008), o que otimizaria a atividade injetada nos mesmos.

Buscando integrar em um único ambiente a possibilidade de reproduzir e analisar tais testes (modelando diferentes equipamentos) com um certo grau de flexibilidade, um software dedicado – denominado GUIGATE (*Graphical User Interface for GATE*)– foi construído neste trabalho. Para isso o GUIGATE executa o código Monte Carlo GATE (Geant4 Application for Tomography Emission), tornando ainda mais amigável o processo de simulação neste código dedicado consolidado. Atualmente, a maneira de executar o GATE se restringe ao terminal do Linux/Unix, e mesmo assim o usuário precisa, além de conhecer essas plataformas, digitar, comando por comando, todos os *setups* de entrada e saída da simulação. Para analisar seus dados de saída, na maioria dos casos, será preciso programar no ROOT (analisador de dados do CERN- *European Organization for Nuclear Research*) para visualizar suas informações gráficas e/ou numéricas, o que acaba por exigir do usuário, desde o início, por exemplo, amplo conhecimento de programação em C++ (BRUN; RADEMAKERS, 1997).

Pensando nisso, o GUIGATE foi projetado para que com simples eventos de *mouse* o usuário construa e visualize em 3D seu modelo de PET *scanner*, além de: (i) poder reproduzir testes NEMA, como o NECR com alterações que julgar necessárias quer seja em suas geometrias, quer seja em seus processos físicos; (ii) quantificar frações de espalhamento e os diferentes eventos coincidências inerentes ao PET; (iii) quantificar as distribuições energética e espacial desses mesmos eventos, entre outros, integrando para isso em uma janela de saída todas as funcionalidades do analisador de dados ROOT.

Ou seja, de um modo simples, rápido, intuitivo e amigável será possível construir todo o modelo do equipamento de modo realístico e assim estimar alguns parâmetros-chave, entre eles o NECR- que dá informações sobre os eventos coincidências os quais em PET, influenciam diretamente a qualidade da imagem, bem como a dose injetada no paciente (WATSON et al., 2005; BADAWI et al., 2005; NICOLAS et al., 2008).

2 REVISÃO DA LITERATURA

Nesse capítulo é dada uma visão geral sobre PET, abordando seu princípio de funcionamento, bem como os principais processos de interação da radiação com a matéria nesse contexto. Isso permite entender melhor como tais processos físicos e outros parâmetros podem influenciar nos eventos coincidências, e como este pode ser modelado usando Monte Carlo (MC). Para tal, são mostradas algumas ferramentas disponíveis, ficando claro o que motivou a escolha do código MC dedicado GATE como a ferramenta a ser utilizada.

2.1 A TOMOGRAFIA POR EMISSÃO DE PÓSITRONS (PET) NO CONTEXTO DO DIAGNÓSTICO POR IMAGEM

O diagnóstico por imagem, em sua ampla gama de características, é atualmente uma ferramenta imprescindível para a medicina moderna (ZAID, 2006).

As técnicas de diagnóstico podem ou não utilizar radiação ionizante, como por exemplo, a ressonância magnética (RM) que usa radiação não-ionizante (campos magnéticos de altíssima potência) para gerar as imagens. As que utilizam radiação ionizante podem ser classificadas como as que utilizam (i) fontes externas ao paciente (Tomografia Computadorizada-CT, Raio-X convencional, etc.) e as que utilizam (ii) fontes internas ao paciente, como é o caso dos diagnósticos realizados em Medicina Nuclear por meio de ferramentas como Gama Câmara, SPECT (Single-Photon Emission Computed Tomography) e PET (Positron Emission Tomography). No primeiro caso (i), as imagens adquiridas se restringem ao aspecto anatomopatológico, enquanto que, no segundo caso (ii), observa-se a possibilidade de se investigar aspectos fisiopatológicos, sendo este um grande diferencial da Medicina Nuclear.

Da necessidade de se investigar a anátomo-fisiopatologia, despontam no cenário dos equipamentos médicos, os sistemas híbridos (SPECT/CT, PET/CT, RM/PET). Dentre estes, destaca-se o PET/CT, dada sua maior utilização para o diagnóstico precoce do câncer bem como acompanhamento de pacientes-oncológicos submetidos a tratamentos complexos como o radioterápico e quimioterápico.

Sendo assim, entre o cabedal de equipamentos médicos disponíveis, dar-se-á ênfase aqui a um estudo do princípio de funcionamento e dos processos físicos básicos inerentes ao PET. Como neste caso a imagem final depende, sobretudo, da interação dos fótons com os meios (paciente e/ou detector), estes mesmos estudos se concentrarão numa visão geral da física das interações das radiações eletromagnéticas com a matéria neste mesmo contexto.

2.2 PRINCÍPIO DE FUNCIONAMENTO DA PET

A tomografia por emissão de pósitrons é uma tecnologia cujo uso tem se desenvolvido amplamente em todo o mundo. O princípio fundamental da PET é a detecção do evento coincidência de dois fótons de energia de 511 keV. Isso só é possível devido ao fato de um conjunto

de radioisótopos emitirem pósitrons (antipartícula do elétron). Nessa emissão, observa-se que a diferença de massa entre o elemento pai “ $M(Z,A)$ ” e o elemento filho “ $M(Z-1,A)$ ” é convertida em energia (E), conforme as expressões (BAILEY et al., 2003):

$$M(Z,A).c^2 > M(Z-1,A).c^2 + 2.m_e.c^2 \quad (2.1)$$

$$E = M(Z,A).c^2 - M(Z-1,A).c^2 \approx 2.m_e.c^2 \quad (2.2)$$

onde m_e é a massa de repouso do elétron, c a velocidade da luz, Z número atômico e A número de massa.

Essa energia é compartilhada entre o núcleo "filho" e o pósitron, que é então ejetado com energia cinética suficiente para se deslocar alguma distância do núcleo "pai". Nesse trajeto, o pósitron vai perdendo energia até se combinar com um elétron do meio e sofrer aniquilação.

Como esse pósitron pode atravessar meios diferentes no corpo do paciente ou do fantoma (ou seja, terá diferentes alcances), o conhecimento da distância média percorrida pelo pósitron é uma propriedade importante na escolha dos radioisótopos para aplicações médicas. Uma das razões diretas é que isso interfere na localização exata de onde ocorre a aniquilação, e por consequência, na resolução espacial da imagem (FAHEY, 2002).

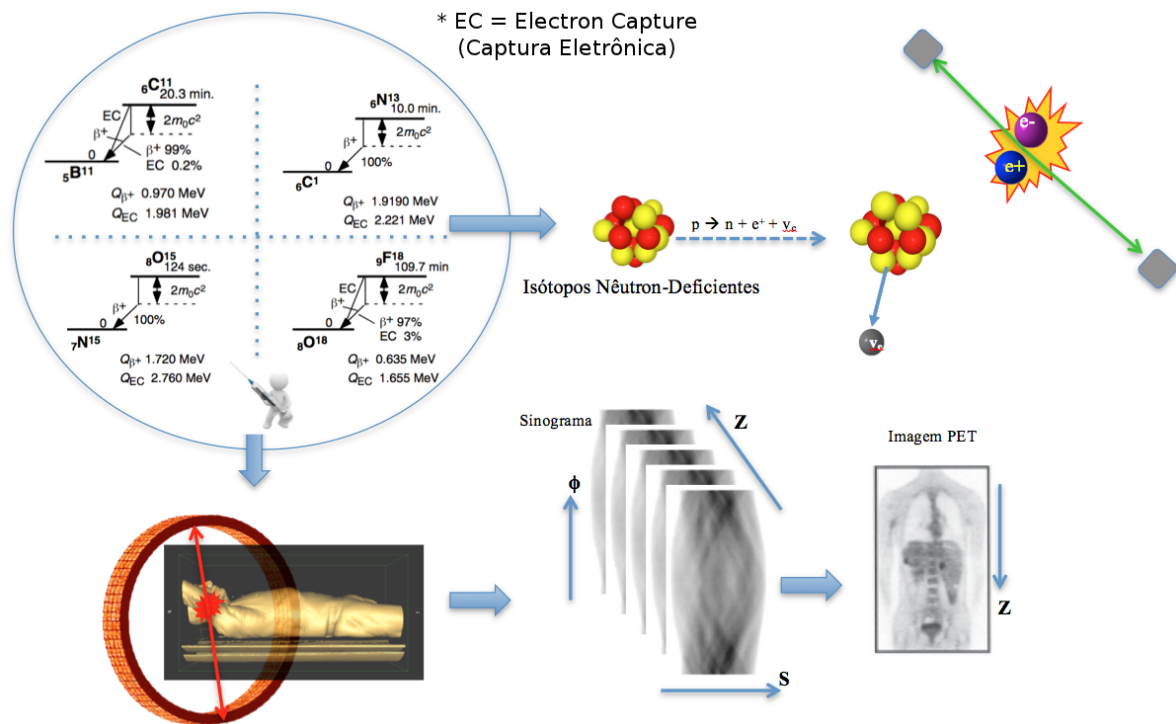
O resultado da aniquilação do pósitron é a produção de dois fótons cada um com energia correspondente à massa de repouso do elétron (511 keV). Esses dois fótons são emitidos a 180° com um desvio padrão de $\pm 0,5^\circ$, em virtude do momento do pósitron e do elétron serem diferentes de zero (PHELPS; SIMON, 2006). Uma síntese de todo esse processo está ilustrado na Figura 1, desde os principais radioisótopos pósitrons-emissores injetados nos paciente (e suas características gerais), a geração do sinograma e, por fim, a imagem gerada propriamente dita.

Cada fóton envolvido no processo é contabilizado como um evento único (*single*) e a contagem total, em contagens por segundo (*cps*), é denominada taxa de contagem de fótons únicos (*single count rate*).

Os dois fótons de aniquilação só são detectados quando saem do corpo do paciente ou do fantoma. Se nenhum destes sofrer espalhamento, a linha que une os dois detectores (linha de resposta- da sigla inglesa LOR), irá passar pelo ponto onde ocorreu a aniquilação. Somente os fótons não espalhados (eventos verdadeiros) contribuem positivamente para a formação da imagem. Na prática, sabe-se que muitos fótons sofrem espalhamento (eventos espalhados) ao atravessar o corpo do paciente e o meio entre este e os detectores. Quando fótons não originados em uma mesma aniquilação são detectados numa mesma janela temporal, têm-se os chamados eventos aleatórios (*random events*) (PEDROSA, 2011). O como cada tipo de evento influencia a imagem será abordado com mais detalhe nos próximos itens, sendo o processo de distinção e contagem desses diferentes eventos uma das metas da ferramenta construída no presente trabalho.

Cada detector do sistema PET gera um pulso com o registro do fóton incidente. Esses pulsos são então combinados no chamado circuito coincidência e, se estiverem dentro da janela temporal pré-determinada (τ), da ordem de nanossegundos, são considerados como um sinal coincidente. Na Figura 2a está ilustrada, de um modo geral, a dinâmica dessa colimação eletrônica. Todo o processo de distinção dos diferentes tipos de eventos é realizado pela colimação

Figura 1 – Visão do princípios básicos na geração da imagem PET.



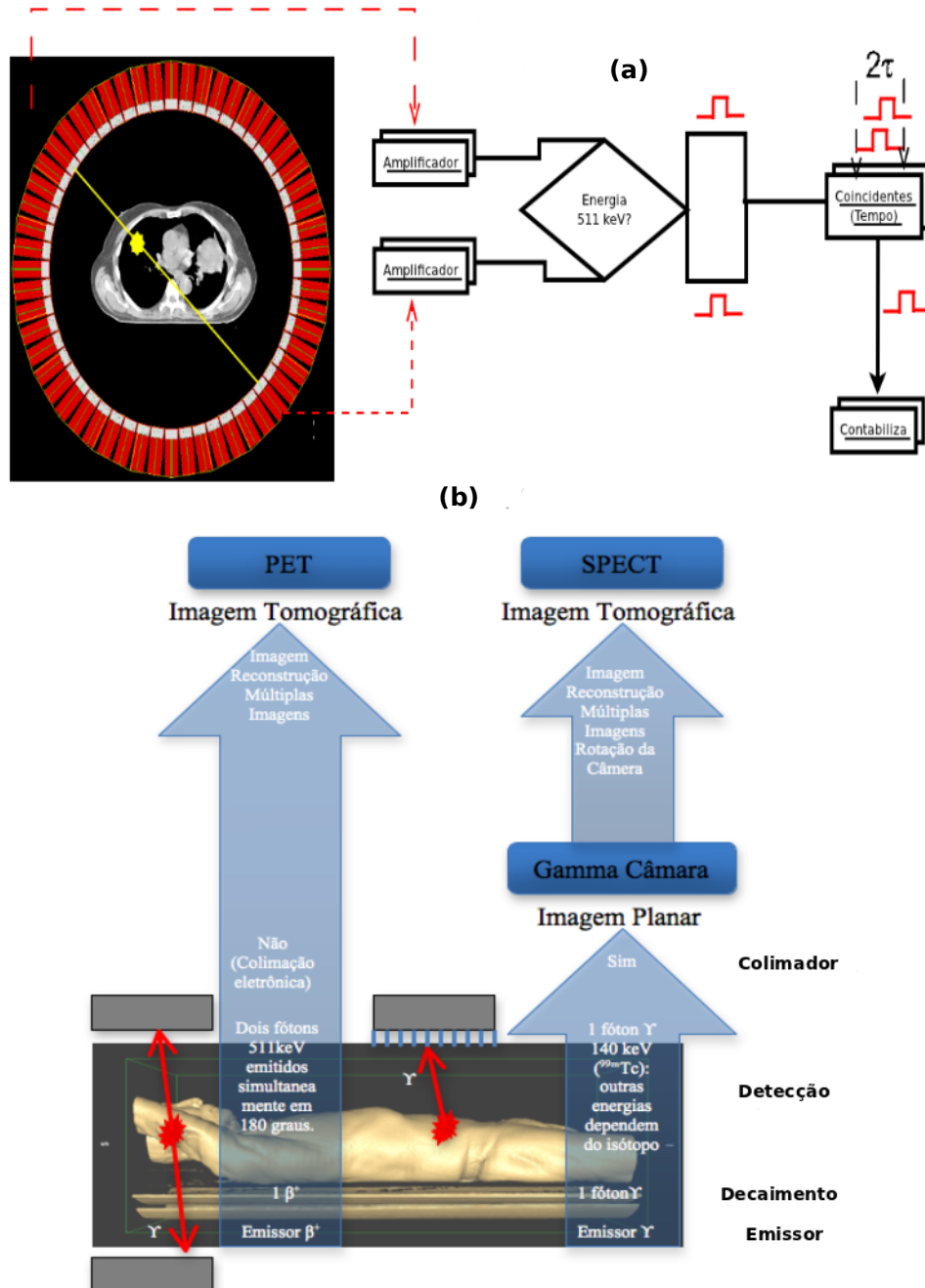
Adaptado: Physical Principles and Tecnology of Clinical PET Imaging, 2004.

eletrônica do PET, ao contrário, por exemplo, das gama câmaras convencionais e SPECT, que usam colimação feita com materiais absorvedores (PEDROSA, 2011), como está ilustrado na Figura 2b.

Os equipamentos PET modernos possuem milhares de detectores, e por conseguinte, milhões de linha de resposta (LOR). Por exemplo, o número de LORs do *scanner* da GE é da ordem de 37 milhões, as quais representam contagens coletadas e armazenadas para processamento posterior (SÁ, 2010). Essa matriz de armazenamento representando todas as LORs pode ser compactada por meio do reconhecimento de simetrias e pela combinação de contagens de linhas de respostas muito próximas em um mesmo local da memória. O arranjo de memórias usadas para estocar os dados é chamado de sinograma. Esse nome deve-se à forma do gráfico gerado com cada ponto das LORs, que formam um meio sino (PEDROSA, 2011; MOHAMMAD; XIAOYI; KLAUS, 2012).

No sinograma final, o valor de cada pixel representa uma coincidência entre um par de detectores opostos. Cada corte gera um sinograma que será posteriormente reconstruído formando a imagem (Figura 1). A construção de sinograma também irá compor o rol de opções ao usuário no software (GUIGATE), descrito a seguir.

Figura 2 – (a) Diagrama básico do circuito coincidência (colimação eletrônica em PET). Os cristais cintiladores estão conectados a amplificadores (Amp) e analisadores de pulsos (AP) individuais. Quando um fóton interage em cada cristal, os sinais são amplificados e analisados para determinar se a energia está dentro de um certo limite inferior e superior. Se o critério de energia é satisfeito, um pulso é gerado pelo AP. Esses pulsos atingem o módulo de coincidência, se estiverem dentro do período 2τ , trata-se de uma coincidência; (b) Principais diferenças entre PET, SPECT e Gama Câmara.



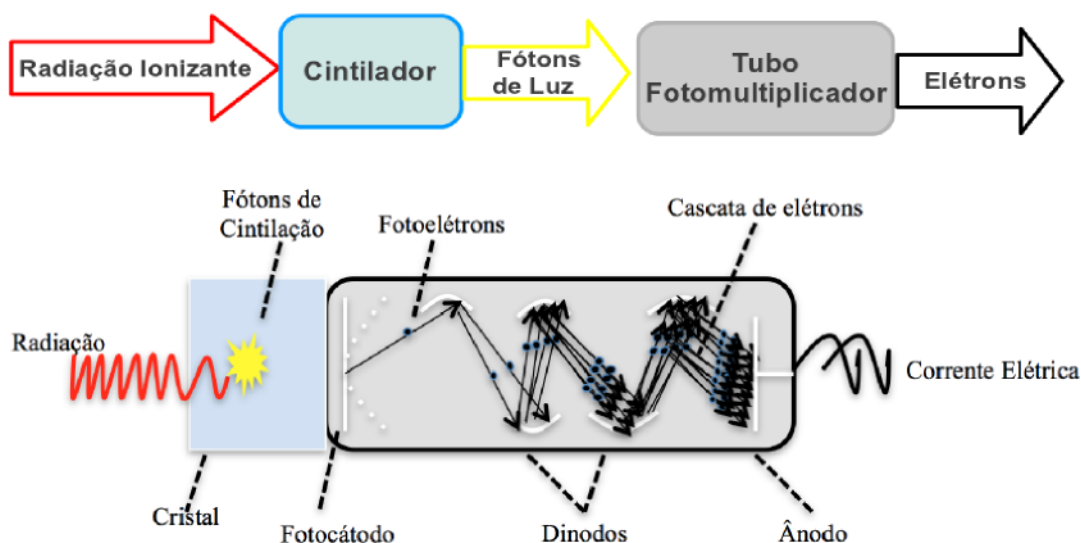
Adaptado: Nuclear Medicine Physics, 2011.

2.3 DETECTORES EM PET

Para tornar o processo de detecção dos fótons de 511 keV possível, os sistemas PET utilizam cristais cintiladores inorgânicos como detectores devido a alta densidade e número atômico, que permitem maior eficiência de detecção. O número de cristais em um sistema PET chega a ser da ordem de 10000, a depender do modelo do *scanner*.

Cada cristal tem como função realizar transdução de parte da radiação ionizante absorvida em fótons visíveis ou ultravioletas (MELCHER, 2000). Esse processo de transdução, chamado cintilação, dura cerca de nano a microssegundos, produzindo pulsos de luz com intensidade proporcional à energia depositada. Para registrar esse pulso, o cristal é acoplado a uma unidade fotomultiplicadora (Photomultiplier, PMT), que tem a função de gerar e amplificar o sinal elétrico em resposta à luz incidente. Essa luz incidente, remove elétrons do fotocátodo, os quais são acelerados devido a uma alta diferença de potencial entre os dinodos. Nos dinodos, os elétrons são multiplicados, atingindo por fim o ânodo, gerando um sinal de saída proporcional à radiação incidente. Esse processo é resumidamente mostrado na Figura 3. A vantagem de cintiladores como, por exemplo, o LSO, é que apresentam a máxima emissão em comprimentos de onda próximos a 400 nm, isto é, na banda ultravioleta. No caso do BGO, cujo comprimento de onda máximo é de cerca de 480 nm, a sensibilidade da PMT é menor (MELCHER, 2000).

Figura 3 – Transdução de informação em um cristal cintilador.



Adaptado: Física das Radiações, 2011.

Outro parâmetro importante aqui, como mostram MELCHER e colaboradores (2000), é o tempo de resposta (ou seja, o tempo que a intensidade da cintilação do cristal leva para ser reduzida à metade), que precisa ser a menor possível para rejeitar eventos aleatórios. Historicamente, os cristais mais utilizados em PET são o NaI(Tl) (constante de decaimento de 230 ns), o LSO (40 ns) e o BGO (300 ns). Na Tabela 1 são sumarizadas algumas propriedades físicas destes e de outros cintiladores utilizados em PET.

Tabela 1 – Propriedades físicas dos cristais cintiladores utilizados em PET.

Propriedades	NaI	BGO	LSO	YSO	GSO	BaF2
Densidade(g/cm^3)	3,67	7,13	7,4	4,53	6,71	4,89
Num.Atômico Efetivo(Z)	50,6	74,2	65,5	34,2	58,6	52,2
Constante de Decaimento(ns)	230	300	40	70	60	0,6
Saída de Luz Absoluta (fótons/keV)	41	9	26	66	8	2
Saída de Luz Relativa	100%	15%	75%	118%	25%	5%
Comprimento de Onda λ (nm)	410	480	420	420	440	220
Índice de Refração	1,85	2,15	1,82	1,8	1,91	1,56
Higroscópico?	Sim	Não	Não	Não	Não	Sim
Coef. Aten. Linear μ (cm^{-1})	0,3411	0,9496	0,8658	0,3875	0,6978	0,4545
Coef. Aten. Mássico $\frac{\mu}{\rho}$ (cm^2/mg)	0,0948	0,1332	0,117	0,853	0,104	0,0929

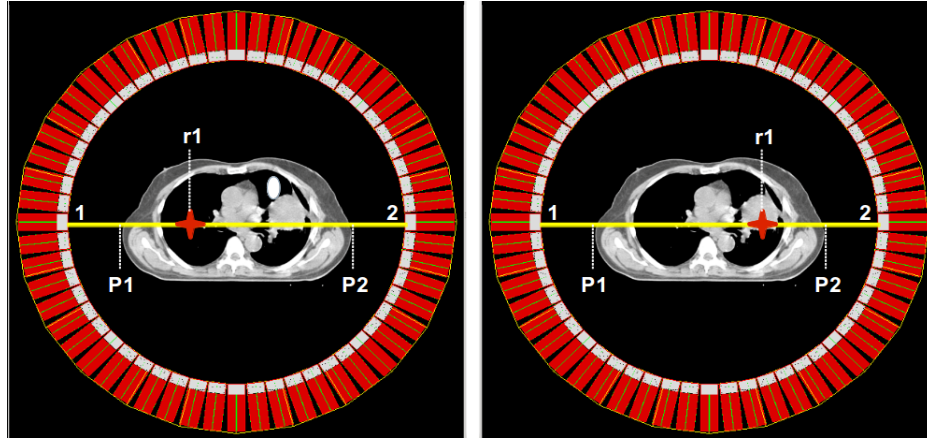
Adaptado: Nuclear Medicine Physics, 2011, pag. 150.

Independente do tipo de detector utilizado, a imagem PET final irá depender das interações fóton-meio, quer seja este meio o paciente ou fantoma, quer seja este meio o detector. Assim, as intensidades dos *pixels* na imagem estão por uma lado relacionadas com a captação do radiofármaco no órgão ou tecido e, por outro, com a ação dos chamados coeficientes de atenuação, que dependem da energia incidente e da composição do meio (PHELPS; SIMON, 2006). Todavia, a definição de coeficiente de atenuação não é única. Há, por exemplo, o conceito de coeficiente de atenuação linear, $\mu(E,t)(cm^{-1})$, parâmetro que representa a seção de choque total do tecido t para o fóton com energia E (TURNER, 2007). Já o termo coeficiente de atenuação mássico, $\mu(E,t)/\rho(cm^2/g)$, representa a seção de choque total de interação por unidade de massa.

Admitindo em PET que os fótons de 511 keV são mutuamente independentes, pode-se notar que a probabilidade total de detecção (P_{detct}) do par de fótons criado ao longo da linha de resposta amarela na Figura 4, independente do local onde ocorreu o evento de aniquilação e depende da espessura do paciente, como mostra os limites de integração na expressão (LEI, 2012):

$$P_{detct} = e^{-\int_{P_1}^{r_1} \mu(r)dr} \cdot e^{-\int_{r_1}^{P_2} \mu(r)dr} = e^{-\int_{P_1}^{P_2} \mu(r)dr} \quad (2.3)$$

Figura 4 – Em PET, a probabilidade de detecção P_{detect} em uma aniquilação para uma dada linha de resposta entre dois detectores D_1 e D_2 independe do local atual onde ocorreu a aniquilação ao longo da linha de resposta.



Adaptado: Tianhu Lei, Statistics Of Medical Imaging, 2012.

2.4 PROCESSOS FÍSICOS EM PET

2.4.1 Principais Processos de Interação da Radiação com a Matéria em PET

Como em MN a faixa de energia varia em geral de 80 a 511 keV, faz-se necessária uma análise cuidadosa dos processos físicos mais importantes a fim de se considerar nas simulações apenas aqueles mais significativos, o que pode reduzir o tempo computacional. Na faixa de energia do PET, os efeitos predominantes, em maior ou menor escala, são: efeito fotoelétrico, espalhamento Compton (ou Incoerente), espalhamento Rayleigh (ou Coerente) (PEDROSA, 2011).

Nas Figuras 5 e 6 é mostrada a predominância dos respectivos efeitos de interação para dois meios importantes nas simulações do presente trabalho (H_2O e BGO), em função da energia (10 keV a 1 MeV).

Na água (Figura 5), principal constituinte dos tecidos biológicos, pode-se ver a predominância do efeito Compton (espalhamento Incoerente, curva azul) para a faixa energética em questão, com destaque para os 511 keV, marcado na curva com um ponto vermelho. Nesse meio, o efeito fotoelétrico é bem menos significativo. Por outro lado, no caso do BGO (Figura 6), por exemplo, a probabilidade de efeito fotoelétrico chega a ser predominante ou equivalente ao efeito Compton, quando se observa faixas de energias menores que 511 keV. Para energias maiores ou iguais a 511 keV, o efeito Compton predomina, assim como no caso da água. Em ambos os casos, a 511 keV, a contribuição do espalhamento Rayleigh é menor como é mostrado nas curvas, porém, precisa este efeito deve ser levado em conta na construção do modelo, uma vez que sua probabilidade em um material como o BGO, que compõe os detectores de alguns equipamentos, é significativa para energias menores que 100 keV. Fótons com energia dessa ordem podem ser provenientes de espalhamentos nos componentes do *scanner* ou mesmo do fantoma ou paciente.

Figura 5 – Variação do coeficiente de atenuação mássico total (cm^2/g), em preto, para água (H_2O).

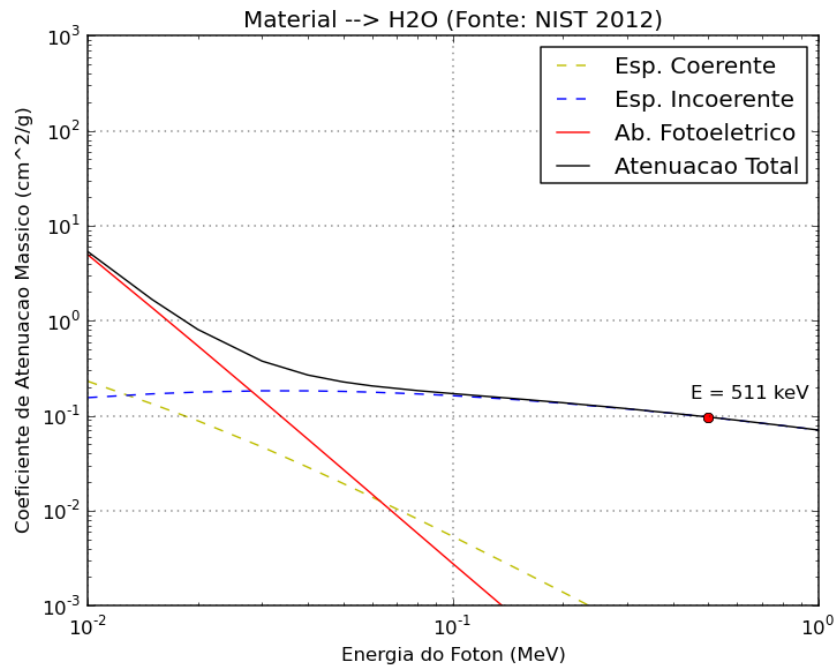
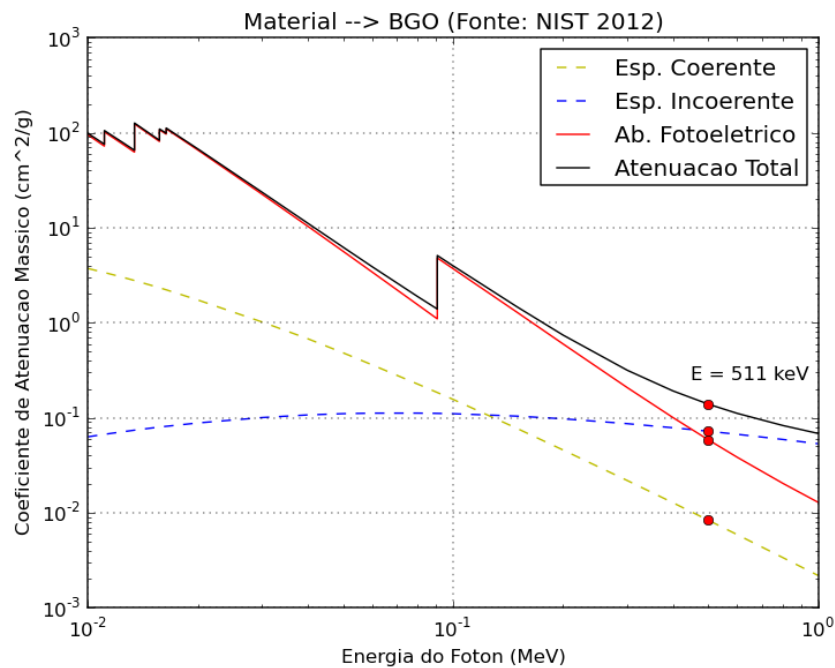


Figura 6 – Variação do coeficiente de atenuação mássico total (cm^2/g), em preto, para o Óxido de Bismuto (BGO).



O efeito fotoelétrico ocorre quando o fóton incidente transfere toda sua energia e é absorvido por um elétron de um átomo, o qual é ejetado para se mover no material. A energia cinética

(E_c) adquirida por esse elétron é a diferença entre a energia do fóton incidente ($h.v$) e a energia de ligação do elétron ao átomo (ϕ), ou seja, (PEDROSA, 2011):

$$E_c = h.v - \phi \quad (2.4)$$

A probabilidade de espalhamento, ou seção de choque, para o efeito fotoelétrico é dada pela seguinte expressão (MOHAMMAD; XIAOYI; KLAUS, 2012):

$$\sigma = \frac{32}{3} \sqrt{2\pi} . r_e^2 \alpha^4 . Z^5 . \left(\frac{m_e . c^2}{E_0} \right) \quad (2.5)$$

onde r_e é o raio clássico do elétron, α é a constante de estrutura fina, E_0 a energia do fóton antes da interação, m_e massa de repouso do elétron e c a velocidade da luz.

No efeito Compton, o fóton é espalhado (inelasticamente) ao interagir com um elétron livre ou fracamente ligado ao átomo. Assim, parte da energia e do momento do fóton é transferida para o elétron, e um fóton com a energia restante é espalhado de um ângulo θ em relação à direção de incidência. A relação entre a energia do fóton, E_0 , antes de interagir com o elétron livre ($m_e . c^2$) e a energia do fóton espalhado, E' , para um dado θ é obtida por (PEDROSA, 2011):

$$E' = \frac{E_0}{1 + \left(\frac{E_0}{m_e . c^2} \right) (1 - \cos\theta)} \quad (2.6)$$

A probabilidade de um fóton incidente ser espalhado, para um dado ângulo sólido $d\Omega$, não é equivalente em todas as direções, ou seja, o espalhamento Compton não é isotrópico. A distribuição angular dos fótons espalhados $\frac{d\sigma}{d\Omega} (m^2/rad)$ é dada pela expressão de Klein-Nishina (PEDROSA, 2011):

$$\frac{d\sigma}{d\Omega} = \frac{r_e^2}{2} (P_{E_0, \theta} - P_{E_0, \theta}^2 . \sin(\theta) + P_{E_{\gamma, \theta}}^3) \quad (2.7)$$

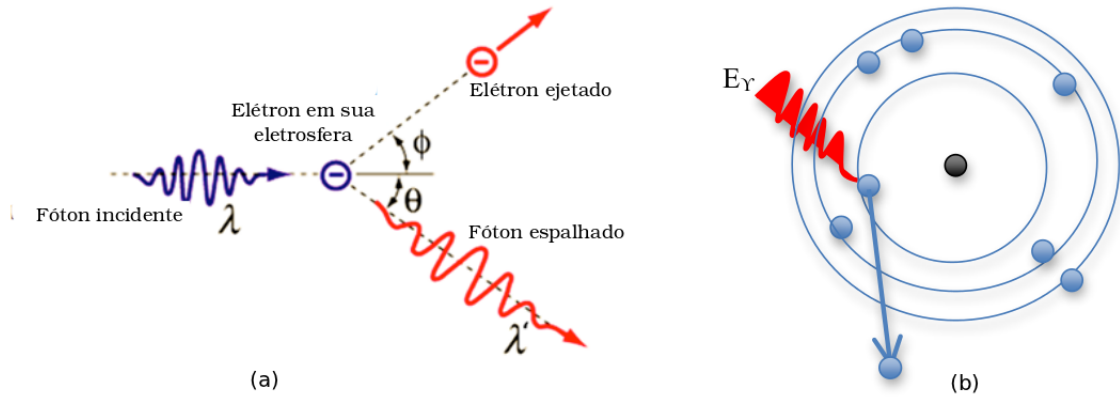
sendo:

$$P_{E_0, \theta} = \frac{E'}{E_0} = \frac{1}{1 + \left(\frac{E_0}{m_e . c^2} \right) (1 - \cos\theta)} \quad (2.8)$$

onde E'/E_0 representa a razão das energias antes e após o espalhamento Compton e r_e o raio clássico do elétron. O valor $\frac{d\sigma}{d\Omega}$ é usualmente conhecido como seção de choque diferencial e representa a probabilidade do fóton incidente ser espalhado em um ângulo sólido $d\Omega$.

A Figura 7 ilustra os efeitos fotoelétrico e Compton do ponto de vista microscópico.

Figura 7 – Representação dos efeitos (a) Compton e (b) Fotoelétrico.



Já o efeito Rayleigh, corresponde a uma interação de fótons de energia baixa com elétrons muito ligados, ocorrendo a interação com o átomo como um todo, sem que haja quase transferência de energia para o meio, exceto a pequena parcela necessária ao recuo do átomo, dada a conservação do momento. Sabe-se que nesse efeito o ângulo de espalhamento é, em geral, próximo da direção de incidência (JOHNS; CUNNINGHAM, 1983). A frequência do efeito e a direção do espalhamento dependem do número atômico (Z) e da energia do fóton, sendo mais provável para energia baixa e Z alto, i.e., (OKUNO; YOSHIMURA, 2010):

$$\sigma_{coe} \propto \left(\frac{Z}{h \cdot \nu}\right)^2 \quad (2.9)$$

Esse fenômeno é entendido como absorção do fóton incidente pelo átomo e sua reemissão em outra direção (YOSHIMURA, 2009). Como o ângulo de espalhamento é pequeno e a energia do fóton espalhado é praticamente a original, esse é um fenômeno difícil de ser observado experimentalmente.

A Tabela 2 mostra do ponto de vista microscópico, para cada efeito, quais são as consequências para o meio e para a radiação incidente, bem como quais são as radiações ionizantes produzidas ou liberadas em decorrência do tipo de interação.

A imagem PET final, portanto, será consequência, sobretudo, das interações fundamentais fótons-meio. Assim sendo, parte desses fótons serão absorvidos e parte não, interagindo a maioria deles por espalhamento Compton dada a faixa energética.

A seguir, será mostrado como os eventos coincidências, e consequentemente, o espalhamento, influenciam na qualidade da imagem.

Tabela 2 – Interações possíveis de ocorrer para radiação eletromagnética ionizante e as consequências das interações.

Interação	Mudança no Meio?	Radiação Incidente: Mudança?	Radiação Produzida	Varia com $h.v$	Varia com Z
Rayleigh	Não	Direção de Propagação	REM* espalhada ($E_0 = E_f$)	$(h.v)^{-2}$	Z^2
Fotoelétrico	Ionização e recuo do átomo, o elétron é ejetado	Fóton é absorvido	Elétron rápido, raios X característico, elétrons Auger	$(h.v)^{-3}$	Z^5
Compton	Ionização do átomo	Fóton perde energia e muda de direção	Elétron rápido, Fóton espalhado com menor energia raios X característico	$(h.v)$ baixo– cresce lento $(h.v)$ auto– decrece lento	Z^2 Z^1

Adaptado: Yoshimura, 2009.

2.4.2 Eventos Coincidências e a sua Influência na Imagem e Otimização da Dose

Como foi visto, detecção de eventos em PET baseia-se em colimação eletrônica. Um evento é considerado como válido se:

1. Os dois fótons são detectados dentro de uma janela temporal pré-definida chamada janela de coincidência;
2. A subsequente linha de resposta formada entre eles quando atingem os cristais está dentro de um ângulo aceito pelo tomógrafo;
3. A energia depositada no cristal por ambos os fótons estiver dentro da janela energética selecionada.

A terminologia comumente utilizada para descrever os vários eventos em PET é: (1) eventos simples, (2) coincidências verdadeiras, (3) coincidências aleatórias, (4) eventos espalhados e (5) múltiplos espalhamentos.

Um evento simples (*single events*) são eventos registrados que não foram processados pelo circuito coincidência. Um PET *scanner* tipicamente converte de 1% a 10% dos eventos simples em pares de eventos coincidências.

Um evento verdadeiro (*True events-T*) é consequência direta de uma única aniquilação pósitron-elétron. Os dois fótons assim gerados são detectados sem sofrer espalhamento, sendo registrados dentro de uma janela de energia e de coincidência temporal.

As coincidências aleatórias (*Random coincidences-R*) ocorrem quando dois núcleos decaem quase ao mesmo tempo. Após a aniquilação de ambos os pósitrons, quatro fótons são emitidos.

Dois desses fótons de diferentes aniquilações são contabilizados na mesma janela temporal e são considerados como provenientes do mesmo pósitron, enquanto os outros dois são perdidos. Inicialmente são considerados como válidos, mas não estão correlacionados com o decaimento do material radioativo em um mesmo ponto e, às vezes, podem ter como origem algum espalhamento sofrido no meio. Segundo Bailey e colaboradores (2003), tais eventos são uma função do número de desintegrações por segundo, sendo a taxa de contagem dos eventos aleatórios ($R_{\alpha\beta}$) entre dois detectores α e β dada por:

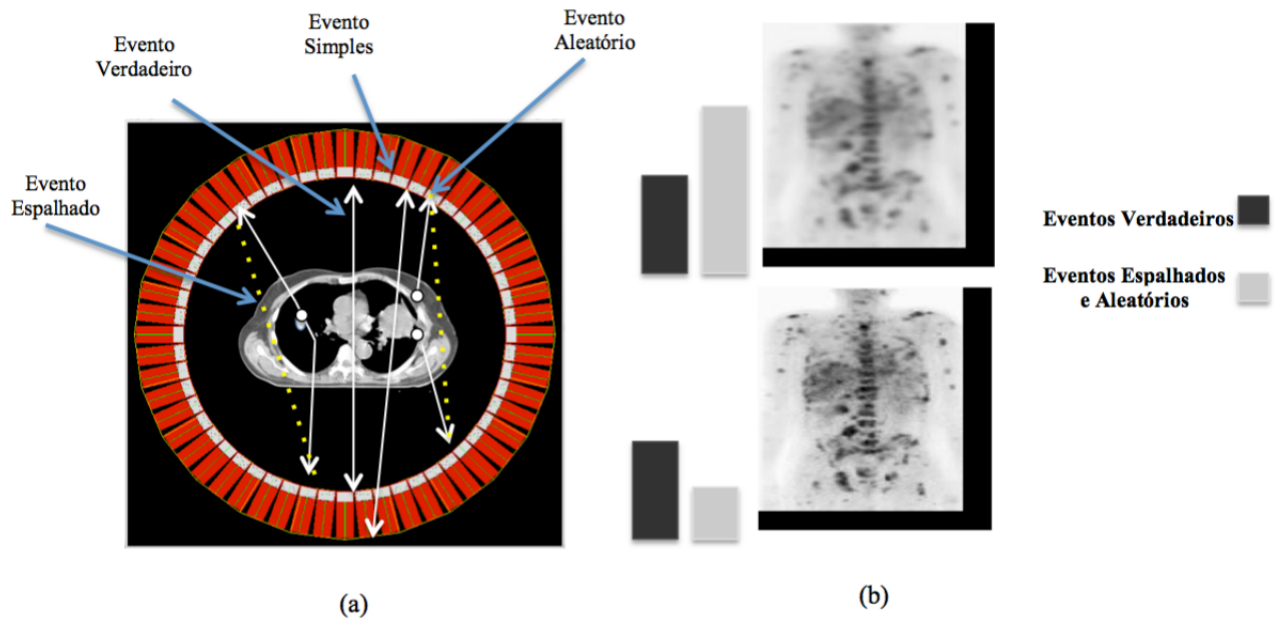
$$R_{\alpha\beta} = 2 \cdot \tau \cdot N_{\alpha} \cdot N_{\beta} \quad (2.10)$$

onde N_{α} e N_{β} são as taxas de eventos simples incidentes nos respectivos detectores α e β , e $2 \cdot \tau$ é a largura da janela temporal. Como normalmente $N_{\alpha} \simeq N_{\beta} = N$, considera-se que a taxa $R_{\alpha\beta}$ aumenta proporcionalmente a N^2 .

Os eventos espalhados (*scattered events-S*) surgem quando um ou ambos os fótons provenientes da mesma aniquilação são detectados em uma janela de coincidência, mas sofreram uma interação Compton, quer seja no fantoma (ou paciente) que contém o material pósitron-emissor, quer seja nos componentes do tomógrafo, interação esta que causa uma perda de energia no fóton e muda sua direção.

Como ilustrado na Figura 8a, as linhas de respostas (LOR) formadas pelos eventos espalhados e aleatórios (amarelas pontilhadas) não estão relacionadas com a origem do evento de aniquilação. Isso, além de causar inconsistência nos dados adquiridos, diminui o contraste e a acurácia final da imagem, como mostrado na Figura 8b.

Figura 8 – (a) Diferentes tipos de eventos coincidências em PET e (b) Efeito qualitativo no contraste da imagem devido a uma redução na contagem de taxas de eventos aleatórios e espalhados.



Adaptado: AIEA 2010.

Quando o sistema detecta três ou mais eventos simples na mesma janela temporal, estes são chamados Eventos Múltiplos e podem ou não ser rejeitados, a depender do interesse.

Na prática os tipos de eventos que vão compor o chamado *prompt* de coincidência são os eventos T, S e R. A partir da variedade de modelos de PET scanners existente atualmente, foi possível se chegar a uma relação entre os diferentes tipos de eventos em função de parâmetros como o raio do anel de detectores (D), o comprimento axial do volume ativo para a aquisição, o comprimento axial total (z) e do septo (L) do tomógrafo, assim como de outros fatores dependentes do operador (por exemplo, janela energética). Estudos analíticos mostram que esses mesmos parâmetros influenciam os eventos T, S e R da seguinte forma (DERENZO; ZAKLAD; BUDINGER, 1975), (DERENZO, 1980):

$$T \propto \frac{Z^2}{D}, \quad (2.11)$$

$$S \propto \frac{Z^3}{L \cdot D}, \quad (2.12)$$

$$R \propto \left(\frac{Z^2}{D}\right)^2 \quad (2.13)$$

Tais eventos (T,R e S) são úteis quando se trata dos testes de controle de qualidade dos *scanners*, conforme descritos no protocolo NEMA NU-2 2007 e no guia de saúde humana da IAEA(2009), que tem o NEMA como base.

Entre estes testes, um dos mais importantes é o teste de análise de perda de contagem e eventos aleatórios. Neste teste usa-se como métrica a taxa de contagem equivalente (NECR- *Noise Equivalent Count Rates*), que é útil do ponto de vista de projetos de novos tomógrafos PET, da avaliação de novos protocolos e de desempenho dos dispositivos de detecção. Este teste se baseia no fato de que as contagens seguem uma distribuição de Poisson (BAILEY et al., 2003) e que a NECR serve para estimar a contribuição dos eventos espalhados e aleatórios para a taxa de coincidência total, como mostra a expressão:

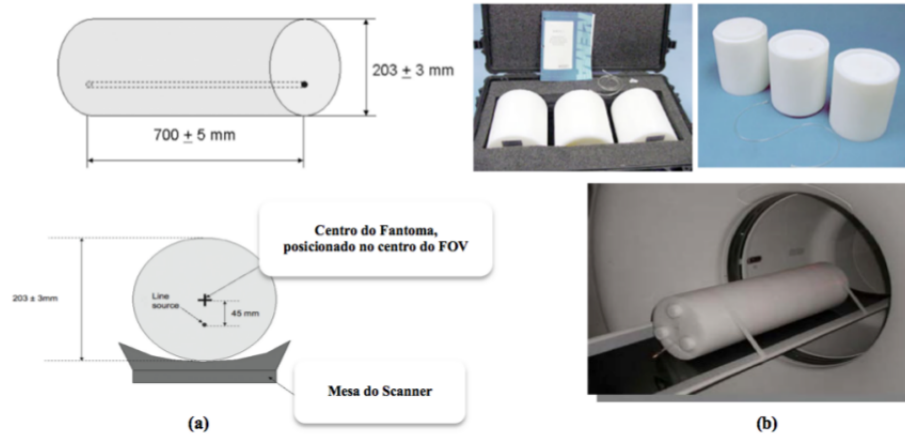
$$NECR = \frac{T^2}{T + S + (k + 1)R} \quad (2.14)$$

onde T,S e R se referem aos tipos de eventos coincidências e o fator k (varia entre 0 e 1) indica o método aplicado para estimar os eventos aleatórios (R): k próximo a zero implica que a estimativa dos *Randoms* foi baseada na contagem dos eventos individuais (*single rate*).

A NECR é medida experimentalmente utilizando um objeto teste padrão (NEMA, 2007): um cilindro de polietileno (densidade $0,96 \pm 0,01 \text{ g/cm}^3$) que tem 203 ± 3 mm de diâmetro e 700 ± 5 mm de altura, preenchido com água, dentro do qual é inserida uma fonte linear de diâmetro interno de $3,2 \pm 0,2$ mm e $4,8 \pm 0,2$ mm de diâmetro externo, emissora de pósitrons, uniformemente distribuída ao longo dos 700 ± 5 mm de altura (Figura 9). A massa total do cilindro é cerca de 20 kg, dividido em três peças desmontáveis.

Como demonstrado por Watson e colaboradores (2005), a NECR varia proporcionalmente com o quadrado do SNR (sinal–ruído) para um dado tempo de aquisição (Δt) (WATSON et al., 2005):

Figura 9 – (a) Fantoma NEMA e suas dimensões em uma visão lateral e axial. A fonte linear fica a 45 ± 1 mm abaixo do centro do fantoma. (b) Fantoma sendo posicionado no centro do campo de visão (FOV) para realização do teste.



Fonte: NEMA 2007.

$$SNR^2 = \left[\frac{T^2}{T + S + (k+1)R} \right] \cdot \Delta t = NECR \cdot \Delta t \quad (2.15)$$

Embora haja essa relação direta, Badawi e colaboradores (2005) mostram que a NECR *per se* é inadequada como métrica única preditora da qualidade da imagem, uma vez que o modelo não leva em conta, por exemplo, o impacto do algoritmo de reconstrução e os efeitos da resolução espacial que variam de equipamento para equipamento.

Além disso, o fantoma padrão de 20 cm de diâmetro e 70 cm de altura não serve para reprodução de casos de aquisição de corpo inteiro: alguns estudos mostram que a NECR varia para um único paciente em diferentes posições axiais durante uma aquisição de corpo inteiro, o que levou a estudos tentando otimizar a atividade injetada, avaliando a NECR para pacientes específicos (Figura 10), culminando na proposição de novos objetos-testes com diâmetros maiores que o padrão NEMA (LARTIZIEN et al., 2002; FAKHRI et al., 2007; NICOLAS et al., 2008).

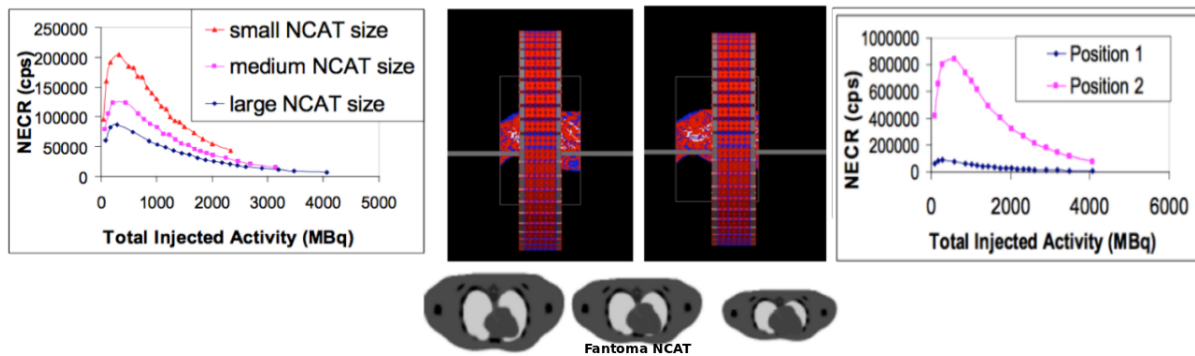
Esses trabalhos apontam para a necessidade de estudos que visem construir modelos para a estimativa da atividade ótima de radiofármaco a ser administrada em pacientes específicos, considerando por exemplo, (i) características pessoais (altura, idade, massa) e (ii) características do *scanner* (geometria e tipo de detector, algoritmo de reconstrução, resolução espacial, etc.).

Diversos fantasmas antropomórficos (de voxel ou de malhas poligonais) de diferentes idades podem ser encontrados na literatura (KRAMER et al., 2003, 2004, 2006; LIMA; BROWN, 2010; LIMA et al., 2011). Contudo, falta uma ferramenta na qual de modo prático e intuitivo um usuário iniciante (ou não) no GATE, possa modelar um PET *scanner* de modo realístico, com a máxima flexibilidade.

O GUIGATE foi projetado para este fim, servindo como uma interface amigável para testes de controle de qualidade em PET.

Dada a natureza estocástica dos fenômenos fundamentais no contexto em estudo, os métodos Monte Carlo (MC) são uma ferramenta fundamental na construção de modelos realísticos.

Figura 10 – Ilustra a influência de parâmetros geométricos (tamanho e posição do fantoma NCAT) nas curvas NECR, em função da atividade injetada (MBq).



Fonte: (NICOLAS et al., 2008).

2.5 CÓDIGOS MONTE CARLO DEDICADOS À MODELAGEM PET

A comunidade científica dispõe atualmente de alguns códigos Monte Carlo (MC) para modelagem de tomógrafos por emissão. Um primeiro desafio consiste, portanto, em definir aspectos e recursos técnicos fundamentais considerando o problema a ser resolvido. No caso aqui em estudo, este desafio consiste em escolher um código MC que simule realisticamente a geometria e os processos físicos inerentes aos atuais PETs, bem como sua eletrônica associada, e que ainda permita ao usuário a possibilidade de estimar dose em meios heterogêneos como o corpo humano. Esses códigos para aplicações em Medicina Nuclear se dividem em dois grupos: (1) códigos de uso geral EGS4 (KAWRAKOW et al., 2011), GEANT (CERN, 2010b), MCNP (ALAMOS, 2003) os quais simulam o transporte de partículas, desenvolvidos sobretudo para Física de alta energia e para Dosimetria; e (2) códigos dedicados criados especificamente para PET e/ou SPECT, como por exemplo, o SORTEO, SIMIND, SimSET e o GATE (VIEIRA et al., 2012).

Vale destacar que cada um deles têm suas vantagens e desvantagens e nesta seção, baseado sobretudo no manual do usuário, abordar-se-á com mais detalhes o GATE (USER'SGUIDE, 2012) por ser a ferramenta escolhida para o presente trabalho, dado o número maior de recursos oferecidos por este como será mostrado.

2.5.1 A Plataforma GATE: uma visão geral.

Embora seja um caminho possível a utilização dos códigos de uso geral para simular tomógrafos por emissão (BARRET et al., 2005) considerando, por exemplo, sua estabilidade e suporte a longo prazo, é preciso considerar que esses códigos MC apresentam umas desvantagens a nível de usuário: por serem projetados para interações de partículas em um contexto amplo, incluem muitas características irrelevantes para PET e SPECT, e acabam por inflar e complicar o seu uso para aplicações específicas, além de exigir do usuário uma programação intensiva, o que indica a necessidade do mesmo dominar previamente uma linguagem de programação (BUVAT; CASTIGLIONI, 2002).

Entre os códigos dedicados disponíveis gratuitamente aqui analisados, o GATE (*Geant4*

Application for Emission Tomography) é o que atualmente dispõe de mais recursos para modelagens de sistema PET e SPECT. A Tabela 3 mostra os recursos fundamentais para modelagem de tomógrafos por emissão, com destaque para a possibilidade de o usuário modelar fenômenos 4D (onde a quarta dimensão é o tempo), acoplar fantasmas de voxel e simular sistemas híbridos (PET/CT, SPECT/CT) com geometrias não convencionais. Entre outras vantagens, o GATE encapsula as bibliotecas do amplamente validado código geral GEANT4 (CERN, 2010b) possibilitando ao usuário e/ou desenvolvedor uma ampla gama de funcionalidades no que diz respeito à simulação realista de tomógrafos por emissão, entre elas a avaliação dosimétrica.

Tabela 3 – Recursos disponíveis nos códigos dedicados a simulações PET e SPECT estudados.

RECURSOS	SORTEO	Códigos Analisados		
		SIMIND	SimSET	GATE
Simula SPECT	NÃO	SIM	SIM	SIM
Simula PET	SIM	NÃO	SIM	SIM
Sistemas Híbridos(PET/CT, etc)	NÃO	NÃO	NÃO	SIM
Simula Fenômenos 4D	SIM (±)	SIM (±)	NÃO	SIM
Simula Geometrias Não-convencionais	NÃO	NÃO	NÃO	SIM
Utilização de Fantasmas Antropomórficos	SIM	SIM	NÃO	SIM
Disponibiliza Código-Fonte	SIM	SIM	SIM	SIM
Linguagem de Desenvolvimento	Não declarada	FORTRAN	C	C++

2.5.1.1 Arquitetura de Simulação do GATE

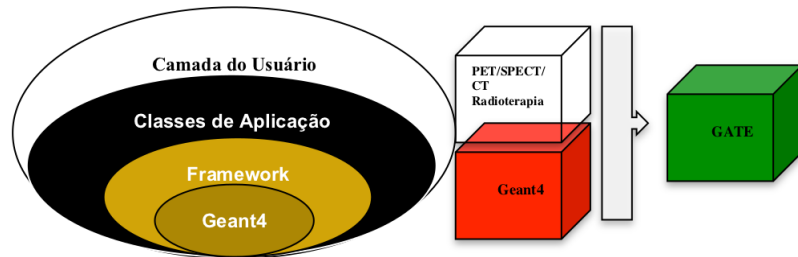
O GATE surgiu em 2001 e é fruto de uma colaboração internacional entre institutos e universidades, formando assim a *OpenGATECollaboration*(<http://www.opengatecollaboration.org>). Atualmente se encontra em sua versão estável 6.2 (Outubro de 2012), sendo regido pela licença *GNU Lesser General Public License* (LGPL). É importante frisar aqui que a versão utilizada no presente trabalho foi a versão estável 6.1.

De um modo geral, o GATE é escrito em C++ e apresenta uma estrutura modular dividida em três camadas principais (Figura 11):

- 1 A camada principal corresponde a camada acessível aos desenvolvedores (com o que e como podem fazer do ponto de vista de desenvolvimento). Define os mecanismos básicos (classes e métodos) disponíveis no GATE para, por exemplo, definição da geometria, dos fenômenos tempo-dependentes, definição da fonte, do módulo de mimetização da eletrônica do detector (digitizer) e dados de saída, como será visto a seguir;
- 2 A camada de aplicação é composta por classes derivadas das classes-bases para modelar objetos específicos ou processos;

- 3 A camada de usuário habilita o *setup* por meio do uso de *scripts* próprios que, representando comandos específicos, acessam as classes e métodos dos itens anteriores.

Figura 11 – Estrutura do GATE mostrando sua hierarquia funcional e áreas de aplicação.



Adaptado: GATE Training Materials, no site da OpenGATECollaboration.

Uma simulação completa no contexto da Medicina Nuclear usando o GATE, envolve um conjunto de passos, entre eles: (1) definição da geometria do *scanner* e do fantoma (analítico ou de voxel); (2) simulação dos processos físicos envolvidos; (3) definição da fonte (tipo, geometria, localização, atividade etc); (4) modelagem da eletrônica associada aos detectores, essencial em PET, realizado pelo módulo chamado *Digitizer*; (5) definição dos formatos dos arquivos de saídas (outputs); (6) processos de inicialização e (7) análise dos dados de saída, em geral, realizados no ROOT, um robusto analisador de dados do CERN.

Cada um dessas etapas reflete o aspecto modular do GATE. Isso significa na prática que precisam ser definidos diferentes arquivos de entrada (com a extensão *.mac*), um para cada módulo, envolvendo assim um conjunto de comandos e processos específicos.

São esses comandos específicos que viabilizam desde a construção do modelo até a execução (*inputs*) e criação dos dados de saída (*outputs*), usando para isso como *interface* o terminal do LINUX/UNIX.

A seguir será abordado com algum detalhe, para o caso específico de um PET *scanner*, todo o processo de simulação e possibilidades, ilustrando-o com partes de um *script* completo que se encontra disponível no apêndice A.

2.5.1.2 Inputs e Execução de um Modelo

(1) Construindo a geometria do *scanner*

A construção da geometria é um processo-chave na simulação, pois nesta etapa os componentes do *scanner* são reproduzidos. Além disso, as partículas interagem com esses componentes modelados, ou seja, os processos físicos ocorrem nestas estruturas.

Toda geometria no GATE fica inserida no volume chamado “*world*”, um cubo grande o suficiente para conter todas as geometrias da simulação.

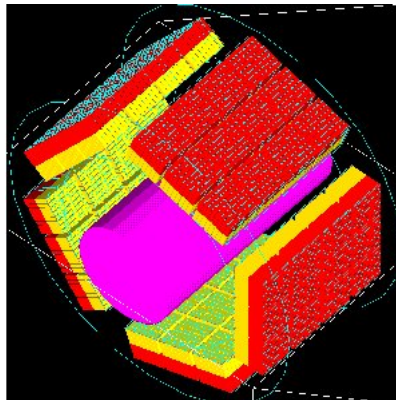
No que diz respeito à simulação da geometria do *scanner*, o próximo passo é definir as características gerais deste. Para isso, o GATE fornece *templates* pré-definidos, os chamados *systems*. Entre os *systems* atualmente disponíveis, encontram-se: *CylindricalPET*, *Scanner*, *ECATT*, *ECATTACCEL*, *CPET* e *OPET*, os quais já permitem simular os dispositivos de

tomógrafos PET disponíveis no mercado. Cada um desses *templates* possuem parâmetros específicos. Como nessa primeira versão do GUIGATE apenas o modelo *CylindricalPET* foi implementado, o foco estará centrado nas características deste mesmo *template*.

O *CylindricalPET* é um sistema PET que pode ser útil para descrever a maioria dos PET *scanners* para para corpo inteiro e pequenos animais, usados em pesquisas fundamentais. Apresenta uma geometria de simetria cilíndrica, e consiste de 5 níveis hierárquicos (Figura 12), do volume-mãe ao volume-filho, como se segue:

1. ***world CylindricalPET*** → Definido como um cilindro no world com um raio interno diferente de zero.
2. ***Rsector (depth=1)*** → Definido como uma caixa e pode ser repetido em forma anelar (ring repeater) no *CylindricalPET*.
3. ***Module (depth=2)*** → É uma caixa dentro do rsector. Pode ser repetido no formato cubicarray repeater, sem repetição ao longo do eixo X, fazendo `repeatNumberX = 1`. Esse nível é opcional.
4. ***Submodule (depth=3)*** → É uma caixa dentro do module. Pode ser repetido no formato cubicarray repeater, sem repetição ao longo do eixo X, fazendo `repeatNumberX = 1`. Esse nível é opcional.
5. ***Crystal (depth=4)*** → É uma caixa dentro do module. Pode ser repetido no formato cubicarray repeater, sem repetição ao longo do eixo X, fazendo `repeatNumberX = 1`.
6. ***Layer (depth=5)*** → É uma (ou várias, no caso de um sistema no formato *poswich*) caixa(s) arranjada(s) dentro do cristal. O comando “*repeater*” não deve ser usado para as camadas (“*layers*”), eles devem ser construídos, um por um ,na macro. Toda “*Layer*” deve ser definida como detector sensível utilizando o comando (“*/attachCrystalSD*”).

Figura 12 – Fantoma cilíndrico e um sistema *CylindricalPET* composto de 5 *rsectors*, 4 *modules* (repetidos ao longo do eixo Z), 3 *submodules* (repetidos ao longo do eixo Y), 64 *crystals* (8 x 8) and 2 *layers* (vermelho e amarelo).



(USER'SGUIDE, 2012)

Para cada uma dessas estruturas-filha são definidos ainda o local no volume-mãe onde serão posicionadas (por meio do recurso */placement*), o material, cor e tamanho de cada componente, tipo de visualização (se *solid* ou *wireframe*), modo de repetição da estrutura etc. Essa árvore de parâmetros básicos para as estruturas descritas acima e mostradas na Tabela 4 pode ser vista resumidamente no esquema na Figura 13.

Figura 13 – Ilustra de modo simplificado a árvore de comandos fundamentais para a construção de um *cylindricalPET*. Os termos exibidos nesta figura se referem a estruturas do GATE, conforme consta no manual de usuários (USER'SGUIDE, 2012)

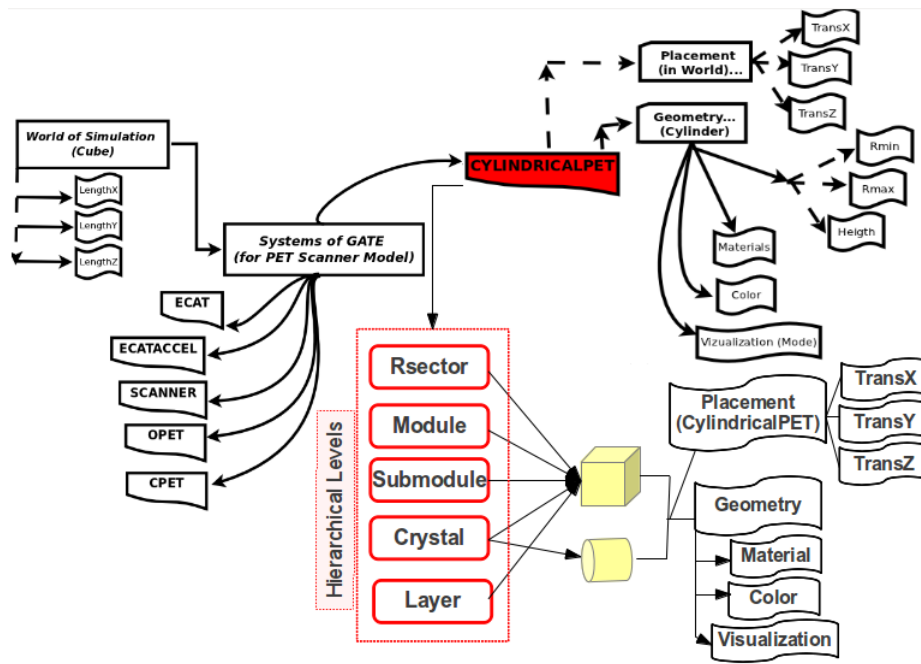


Tabela 4 – Níveis Hierárquicos do *CylindricalPET*.

Sistema	Níveis Hierárquicos	Geometria	Depth ID (Digitizer)	Formato Arquivo de Saída
CylindricalPET	"Rsector"	box	1	Básico: ASCII, ROOT and Raw. Específico: LMF
	"Module"	box	2	
	"Submodule"	box	3	
	"Crystal"	box, cilíndrico, etc	4	
	"layer[i], i=0,3"	box	5	

Um aspecto importante que deve ser levado em conta é a restrição do ponto de vista da construção da geometria: um volume-filho deve ser menor ou igual ao volume-mãe. O número de hierarquias e componentes destes certamente vai depender das características do *scanner* simulado.

A Figura 14 mostra um exemplo da construção da geometria de um PET *scanner* baseado no GE Advance cujas especificações constam no Apêndice A. Note que com exceção do cristal, todas as demais estruturas do *CylindricalPET* são paralelepípedos com repetição de volumes

(2) Construindo a geometria do fantoma

O fantoma, analítico ou de voxel, deve ser definido usando os materiais inclusos no banco de dados do GATE. Cada voxel pode ser feito de um material específico com sua densidade própria.

Para esta versão do GUIGATE, encontra-se funcionando a possibilidade de utilização dos fantasmas analíticos (formas geométricas simples ou composições destas).

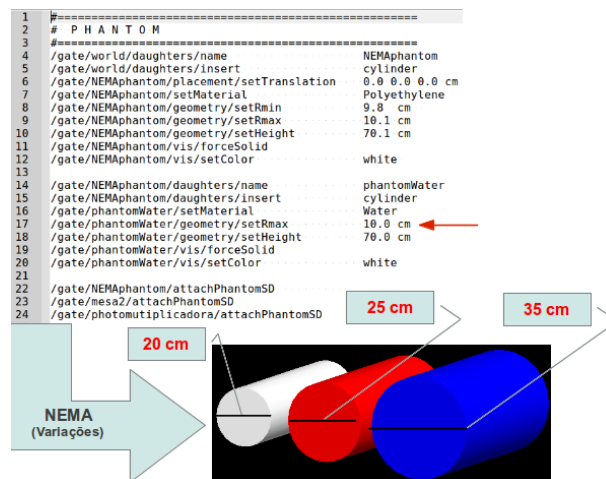
Um passo importante para qualquer geometria, seja o fantoma ou o *scanner*, é criar o chamado volume sensível, para que as interações que eventualmente ocorram neste sejam contabilizadas. O GATE registra e armazena informações relacionadas às interações apenas para aqueles volumes que estão como volumes sensíveis. Todas as interações que ocorrem em volumes não-sensíveis são perdidas. Dois tipos de volumes sensíveis estão disponíveis no GATE:

- 1- *crystalSD*: permite registrar informações de interações dentro de volumes pertencentes ao *scanner*, por exemplo cristais, mesa, etc.
- 2- *phantomSD*: pode ser usado para registrar informações das interações Compton e Rayleigh ocorridas no campo de visão (FOV) do *scanner*. Essa informação é usada para estimar se os fótons que atingem um detector é proveniente de um fóton espalhado ou não. Assim, em PET, o *phantomSD* atualmente é usado apenas como um modo de discriminar coincidências verdadeiras de espalhadas.

Uma simulação completa normalmente envolve volumes definidos como *crystalSD* e *phantomSD*.

A Figura 15 ilustra como criar um fantoma analítico baseado no fantoma NEMA para teste de controle de qualidade em PET.

Figura 15 – Mostra um script no arquivo "*phantom.mac*" para modelar um fantoma cilíndrico baseado no NEMA 2007 (fantoma de 20 cm de diâmetro, branco). Os outros dois são variações realizadas ao alterar o parâmetro mostrado na seta vermelha (Linha 17).



(3) Processos Físicos

Uma simulação usando Monte Carlo é constituída de um conjunto de passos. O objetivo da implementação dos processos físicos é decidir onde estes passos ocorrem e quais as interações mais prováveis em cada um deles.

As simulações das interações eletromagnéticas no GATE são provenientes das bibliotecas do GEANT4 (CERN, 2010a). Há dois tipos de pacotes disponíveis para modelar processos eletromagnéticos: SEP (Standard Energy Electromagnetic Processes) e LEP (Low Energy Electromagnetic Processes). A LEP, biblioteca utilizada para aplicações médicas, simula processos eletromagnéticos e hadrônicos com energias abaixo de 10 GeV (hadronterapia). Ambos os pacotes simulam elétrons, pósitrons, raios γ , raios X, múons, hádrons e íons. Quando usa-se a LEP, como é o caso do presente trabalho, assume-se fótons e elétrons com energia de corte de 250 eV e inclui-se espalhamento Rayleigh. Este limite de corte de 250 eV permite a ocorrência de emissões de raios X característicos da camada-K para $Z < 6$. Esses processos eletromagnéticos da LEP são utilizados para simular a interação de partículas com a matéria e inclui modelos para: espalhamento Compton, espalhamento Rayleigh, efeito fotoelétrico, ionização e Bremsstrahlung.

A implementação de todos esses processos são feitas em duas fases: (i) cálculo da seções de choque totais e (ii) geração dos estados finais.

A seção de choque total para uma dada energia (entre uma maior e menor) é calculada por interpolação, durante a inicialização, utilizando, por exemplo, entre outras, o seguinte tipo de interpolação:

$$\log(\sigma(E)) = \log(\sigma_1) \frac{\log(E_2) - \log(E)}{\log(E_2) - \log(E_1)} + \log(\sigma_2) \frac{\log(E) - \log(E_1)}{\log(E_2) - \log(E_1)} \quad (2.16)$$

O GATE utiliza o princípio de limites ou cortes para produção de partículas secundárias. Um primeiro corte permite definir a distância (alcance de parada) para a qual ainda se observará a produção de partículas secundárias. O segundo corte, convertido internamente após o corte anterior ter sido definido, consiste em definir a energia mínima que viabiliza a produção de partículas secundárias.

Em síntese, o processo se resume a:

- 1- Seleção da biblioteca de interação apropriada (STD ou LEP)
- 2- Habilitar ou desabilitar os processos físicos (efeitos fotoelétrico, Compton, Rayleigh etc.)
- 3- Definir, conforme o problema, os cortes para energia e alcance.

(4) Definição da fonte radioativa

O pacote GEANT4 *General Particle Source* (GPS) é usado para inicializar a simulação de partículas de um dado tipo e direção, com uma dada energia cinética. Uma fonte no GATE é definida pelo seu tipo de partícula, pela sua geometria (volume ou voxelizada), direção, energia,

Figura 16 – Mostra um script no arquivo "physics.mac" para modelar os processos físicos de interesse.

```

1  #=====
2  # PHYSICS
3  #=====
4  /gate/physics/addProcess PhotoElectric
5  /gate/physics/addProcess Compton
6  /gate/physics/addProcess GammaConversion
7  /gate/physics/addProcess LowEnergyRayleighScattering
8
9  /gate/physics/addProcess ElectronIonisation
10 /gate/physics/addProcess Bremsstrahlung
11 /gate/physics/addProcess PositronAnnihilationStd
12
13 /gate/physics/addProcess MultipleScattering e+
14 /gate/physics/addProcess MultipleScattering e-
15
16 /gate/physics/processList Enabled
17 /gate/physics/processList Initialized
18
19 #=====
20 # CUTS
21 #=====
22
23 /gate/physics/Gamma/SetCutInRegion ..... BGO 1.0 cm
24 /gate/physics/Electron/SetCutInRegion ..... BGO 1.0 cm
25 /gate/physics/Positron/SetCutInRegion ..... BGO 1.0 cm
26
27 /gate/physics/Gamma/SetCutInRegion ..... NEMAphantom 0.1 mm
28 /gate/physics/Electron/SetCutInRegion ..... NEMAphantom 0.1 mm
29 /gate/physics/Positron/SetCutInRegion ..... NEMAphantom 0.1 mm
30
31 /gate/physics/SetMaxStepSizeInRegion ..... NEMAphantom 0.1 mm

```

atividade e movimento, se necessário. A meia vida do material radioativo em geral é obtido do banco de dados do GEANT4, mas pode ser definido um valor qualquer pelo usuário. A atividade da fonte pode ser confinada em um volume específico (exemplo, dentro do volume do fantoma). Para modelagem da fonte no GATE, intervalos de tempo aleatório δt entre eventos que ocorrem num tempo t são amostrados usando uma distribuição exponencial (SILVA, 2010):

$$p(\delta t) = A(t) \cdot e^{-A(t) \cdot \delta t} \quad (2.17)$$

onde $A(t) = A_0 \cdot e^{-\frac{t-t_0}{\tau}}$ é a atividade da fonte a um tempo t , A_0 é a atividade da fonte definida pelo usuário em um instante inicial t_0 para uma meia-vida τ .

Fantomas de voxel ou dados de pacientes submetidos a PET podem ser usados, o primeiro como geometria a ser exposta, o segundo como fonte radiativa para reproduzir aquisições realistas. Os dados de emissão neste último caso são convertidos em níveis de atividade correspondentes.

(5) Definição do *Digitizer*

O propósito deste módulo é mimetizar o comportamento dos detectores do *scanner* e do canal de processamento. Ele converte as interações de partículas carregadas e fótons em energia armazenada, posição de detecção e coincidências. Para fazer isso, utiliza-se das informações registradas nos volumes sensíveis. O *digitizer* processa essas interações armazenadas e produz contagem de coincidências. O fluxo de dados típico é o seguinte:

- 1- Uma partícula é gerada com seus parâmetros como por exemplo tipo, momento e energia;
- 2- Um *step*, ou trajetória elementar entre duas interações discretas (fotoelétrico, Compton, produção de par etc.) é aplicado. Durante o *step* as mudanças na energia e momento da

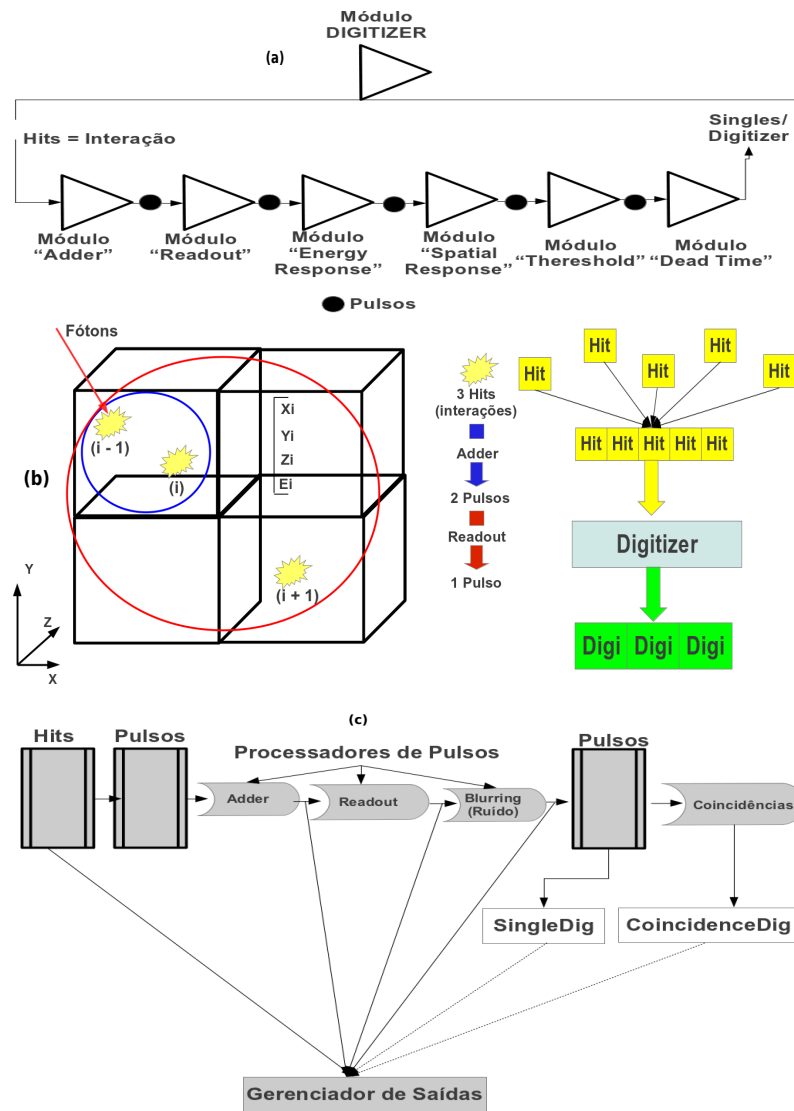
partícula são calculados. O comprimento do *step* depende do tipo da partícula e material, da natureza da interação etc;

- 3- Se esse *step* ocorre dentro de um volume correspondente a um volume sensível, como visto, ela será armazenada. Por exemplo, essa informação inclui a energia depositada, o momento antes e após a interação, e assim sucessivamente. Todo esse conjunto recebe o nome de *hit*;
- 4- Os passos 2 e 3 são repetidos até que a energia da partícula seja menor que o valor de corte pré-definido, ou caso a partícula saia dos limites geométricos do *world*. Essa série de *steps* que formam a trajetória da partícula é chamada de *track* no GEANT4;
- 5- A quantidade total de energia depositada no cristal é filtrada pelo *digitizer*. A saída desse módulo corresponde ao sinal depois que este é processado pelo *Front End Electronics (FEE)* do equipamento. O FEE, em geral, consiste de muitas unidades de processamento, trabalhando em série e/ou em paralelo.

Esses processos são repetidos para cada evento na simulação afim de produzir um ou mais conjuntos de eventos simples (*singles*). Esses eventos simples são armazenados no arquivo de saída (como uma árvore do ROOT, como será visto a seguir). No caso do PET, uma vez que essa lista inicial é criada, um segundo estágio de processamento pode ser inserido para organizar esses eventos simples numa lista de eventos coincidentes. Para fazer isso, o algoritmo procura numa lista contendo um conjunto de *singles*, aqueles que são detectados em um dado intervalo de tempo e de energia, criando os chamados eventos coincidências. Finalmente, se desejado, os dados coincidentes podem ser filtrados para reproduzir uma possível perda ocorrida no circuito global de coincidência ou durante o transporte de dados.

Como pode ser visto na Figura 17, o módulo *digitizer* apresenta um conjunto de submódulos.

Figura 17 – (a) Canais de processamento do módulo *digitizer* que organiza as *hits* (interações) para *singles* (eventos simples) e (b) As ações dos módulos *adder* e *readout*. O *adder* transforma os *hits* em um pulso dentro de cada volume individual, e então o *readout* soma um grupo desses pulsos formando um *single* no nível de volume (o chamado *depth* no GATE definido pelo usuário). (c) O papel dos processadores de pulso na geração dos eventos coincidência e simples, bem como a pilha de dados de saída representada pelo gerenciador de saída de dados .



Adaptado de (LAZARO et al., 2004a).

A Figura 18 ilustra no GATE como construir um canal de processamento no *Digitizer* usando *scripts*.

Figura 18 – Mostra um script no arquivo "digitizer.mac" para modelar a eletrônica dos detectores.

```

1  #####
2  # DIGITIZER: DETECTOR ELECTRONIC RESPONSE
3  #####
4  # A D D E R
5  /gate/digitizer/Singles/insert ..... adder
6
7  # R E A D O U T
8  /gate/digitizer/Singles/insert ..... readout
9  /gate/digitizer/Singles/readout/setDepth 1
10
11 # T R I G G E R
12 # transfer efficiency #photoelectron loss within the detector
13 /gate/digitizer/Singles/insert transferEfficiency
14 /gate/digitizer/Singles/transferEfficiency/chooseNewVolume Block
15
16 # DEFINE QUANTUM EFFICIENCY OF THE PHOTODETECTOR
17 /gate/digitizer/Singles/insert ..... quantumEf
18 /gate/digitizer/Singles/quantumEf/chooseQEVolume Block
19 /gate/digitizer/Singles/quantumEf/setUniqueQE 0.90
20
21 # E N E R G Y   B L U R R I N G == Resposta Energética
22 /gate/digitizer/Singles/insert ..... blurring
23 /gate/digitizer/Singles/blurring/setResolution 0.20
24 /gate/digitizer/Singles/blurring/setEnergyOfReference 511. keV
25
26 # N O I S E
27 /gate/distributions/name ..... gaussian
28 /gate/distributions/insert ..... Gaussian
29 /gate/distributions/gaussian/setMean 511 keV
30 /gate/distributions/gaussian/setSigma 1 keV
31
32 # B L O C K S
33 /gate/digitizer/Singles/name ..... blockdeadtime
34 /gate/digitizer/Singles/insert ..... deadtime
35 /gate/digitizer/Singles/blockdeadtime/setDeadTime 625 ns
36 /gate/digitizer/Singles/blockdeadtime/setMode ..... paralyzable
37 /gate/digitizer/Singles/blockdeadtime/chooseDTVolume Block
38
39 # E N E R G Y   W I N D O W
40 /gate/digitizer/Singles/insert ..... threshold
41 /gate/digitizer/Singles/threshold/setThreshold 375. keV
42 /gate/digitizer/Singles/insert ..... upholder
43 /gate/digitizer/Singles/upholder/setUphold 650. keV
44
45 # "MODULE"
46 /gate/digitizer/Singles/name ..... moduledeadtime
47 /gate/digitizer/Singles/insert ..... deadtime
48 /gate/digitizer/Singles/moduledeadtime/setDeadTime 425 ns
49 /gate/digitizer/Singles/moduledeadtime/setMode ..... paralyzable
50 /gate/digitizer/Singles/moduledeadtime/chooseDTVolume module
51
52 #####
53 # - C O I N C I D E N C E   S O R T E R
54 #####
55 /gate/digitizer/Coincidences/setWindow 6.25 ns
56 /gate/digitizer/name ..... delay
57
58 /gate/digitizer/insert ..... coincidenceSorter
59 /gate/digitizer/delay/setWindow 6.25 ns

```

(6) Definição e análise dos arquivos de saída

No GATE há diferentes tipos de formato de arquivos de saída, que podem ser habilitados, conforme mostrado na Tabela 5. Por padrão, o formato em comum para todos os sistemas são ASCII e ROOT. No caso de aplicações robustas, com alta demanda de análise estatística, é mais indicado o uso do ROOT o que contém as mesmas informações do ASCII, com a vantagem do tamanho do arquivo de saída de dados de ser mais compacto.

Tabela 5 – Tipos de arquivos de saída (padrão e específicos) dos diferentes sistemas para modelagem PET.

Sistema	Arquivos de Saída Disponíveis
CylindricalPET Scanner CPET ECAT ECATACCEL OPET	Padrão: ASCII e ROOT
CylindricalPET e OPET ECAT e ECATACCEL	Específico: LMF SINOGRAM ou ECAT7

Além disso, é possível contar com todos os recursos do analisador de dados do CERN, o framework ROOT, projetado para lidar com arquivos de saída grandes de modo eficiente. Utilizando o paradigma orientado a objeto, o ROOT dispõe de diversos métodos para construção e visualização de gráficos, tais como histogramas (2D ou 3D), *fitting* ou ajuste de dados etc.

No GATE, as saídas com extensão ".root" são também geradas usando scripts (Figura 19). As mesmas apresentam em sua estrutura interna uma *NTuple* chamada *GATE* e árvores ("TTrees"),

dai o nome "ROOT", onde são armazenados os eventos *Hits*, *Singles* e *Coincidences*. Por padrão acompanham esses pacotes 4 histogramas contendo as seguintes distribuições:

1. *Acolinea_Angle_Distribution_deg*: o ângulo (em graus) entre os dois fótons de aniquilação;
2. *Positron_Kinetic_Energy_MeV*: a energia cinética do β^+ ;
3. *Ion_decay_time_s*: o tempo de decaimento;
4. *Positron_annihil_distance_mm*: o alcance do β^+ .

Usando C++, o usuário pode gerar códigos para realizar uma completa análise dos dados, escolhendo quais desses deseja incluir em sua análise usando o compilador/interpretador do ROOT.

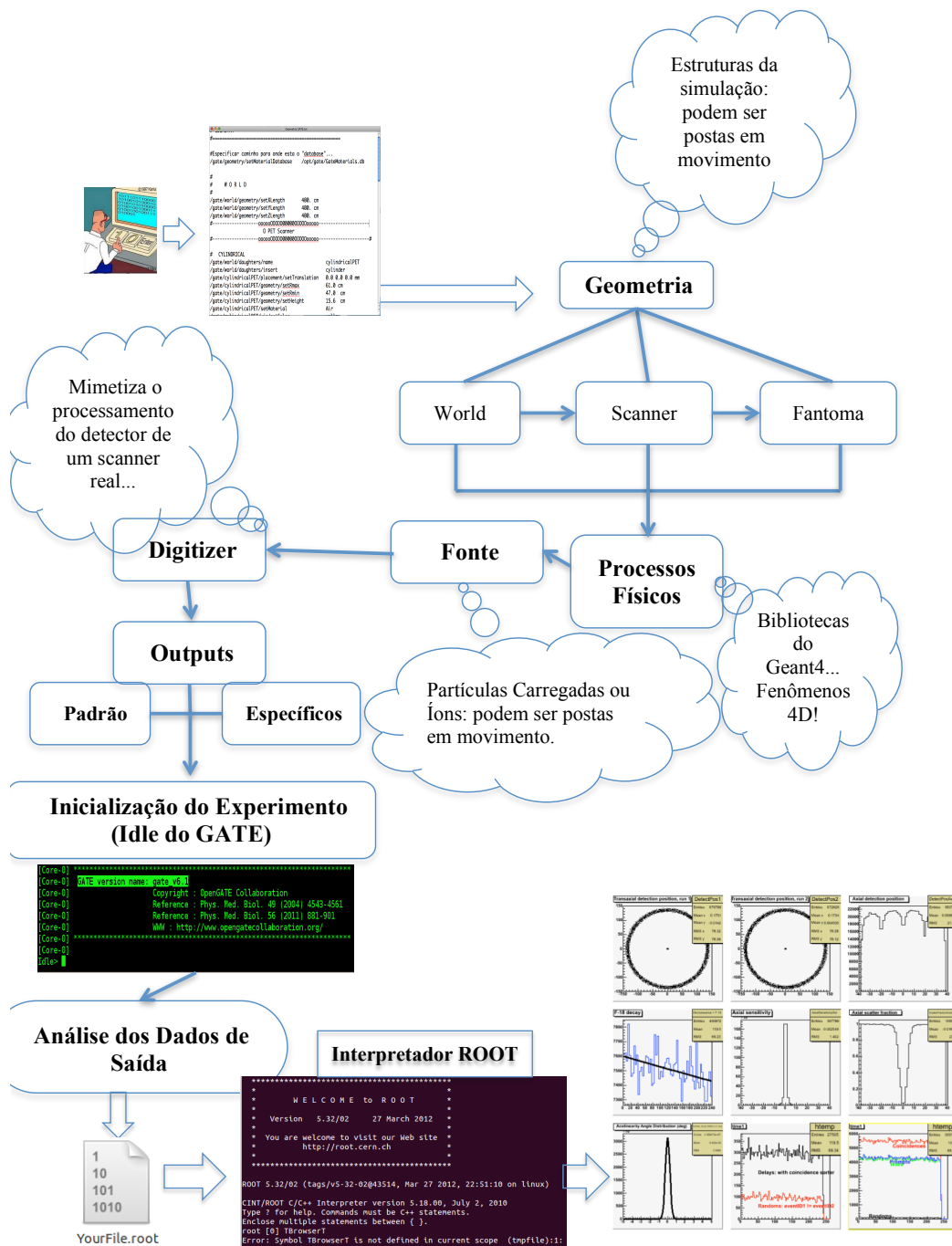
Figura 19 – Mostra um script no arquivo "output.mac" definindo o arquivo ROOT como saída padrão.

```
#####
# DATA OUTPUT #####
#####

/gate/output/root/enable
/gate/output/root/setFileName ..... YourFile
/gate/output/root/setRootHitFlag ..... 0
/gate/output/root/setRootSinglesFlag ..... 1
/gate/output/root/setRootCoincidencesFlag 1
/gate/output/root/setRootdelayFlag ..... 1
```

A Figura 20 sintetiza tudo o que foi descrito acima acerca da arquitetura de simulação do GATE, onde se pode notar a exigência de conhecimento prévio de manipulação do terminal do LINUX/UNIX, uma vez que tanto o acesso a inicialização do GATE, quanto a análise dos dados de saída (*.root*) no interpretador do ROOT, passam por usar o terminal diretamente.

Figura 20 – Síntese da arquitetura geral de simulação do GATE sob a forma de um passo a passo.



3 MATERIAIS E MÉTODOS

Neste capítulo são descritos os aspectos principais de como o GUIGATE foi criado e os recursos utilizados para tal. Os nomes de componentes do GUIGATE, incluindo as classes e comentários nos códigos, foram escritos em inglês para ser fiel às nomenclaturas do GATE e por assim também viabilizar uma ampla divulgação e potencial utilização pela comunidade científica internacional, como é o caso da *OpenGATE Collaboration*.

3.1 HARDWARE E SOFTWARE

O recurso de *hardware* utilizado foi um *laptop* da HP (Pavilion dv6- HD 500GB) com 4GB de memória RAM e processador AMD Phenom(tm) II N930 *quad-core*. O sistema operacional foi o Linux (Ubuntu 11.10) 32-bit.

Os softwares básicos instalados no Ubuntu foram: (i) as bibliotecas padrões da linguagem de programação Python, com (ii) *Numerical Python*, mais conhecido como Numpy (MAC-KINNEY, 2012) e (iii) *wxPython* (AWARE, 2010). Todas essas bibliotecas estão livremente disponíveis na árvore de distribuição do Ubuntu. Foram instaladas também (iv) o editor de texto e ambiente de desenvolvimento (IDE) Geany (<http://www.geany.org/>). Além disso, foram instalados (v) a versão 6.1 do GATE e os seguintes pré-requisitos para o mesmo: o GEANT 4.9.4, CLHEP 2.1.0.1 e o ROOT 5.32 habilitado para usar com python, por meio da biblioteca chamada pyROOT (<http://root.cern.ch/drupal/content/pyroot>).

3.2 LINGUAGEM E FERRAMENTAS

Para construção do software GUIGATE a linguagem de programação utilizada foi o Python, por meio da biblioteca *wxPython* (AWARE, 2010). Esta linguagem foi escolhida por ter uma sintaxe legível. Suas palavras-chave são fundamentalmente voltadas para a estruturação do programa (e não para instruir o computador como compilar ou interpretar trechos de código), o que facilita a manutenção e desenvolvimento de algoritmos anteriormente construídos. Além de livre, ela conta com uma comunidade internacional crescente de desenvolvedores ativos, que contribuem para construção de códigos-fonte robustos, um fator importante, com vastas bibliotecas de classes dispondo de recursos para aplicações que vão desde banco de dados a interfaces gráficas com o usuário (GUI).

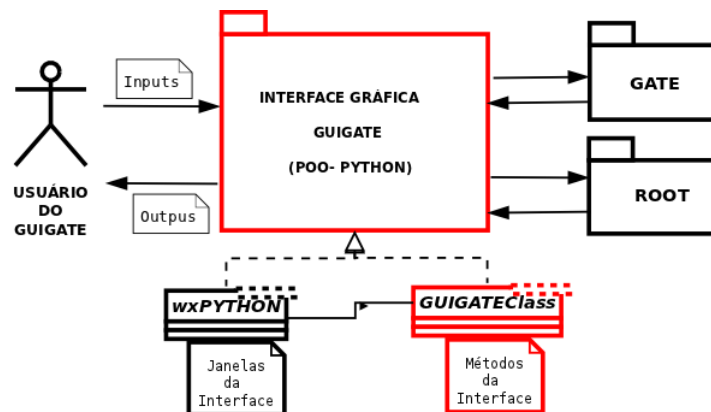
Todo o GUIGATE foi construindo utilizando o paradigma de programação orientada a objetos (POO), que se baseia na composição e interação entre as diversas unidades estruturais do software chamadas de objeto. Isso consiste na implementação de um conjunto de classes que servem para definir os objetos presentes no sistema de software. Cada classe determina, por sua vez, o comportamento (definido nos métodos ou funções) e os estados possíveis (atributos) de seus objetos, assim como a relação entre eles.

Assim sendo, no GUIGATE, os recursos da biblioteca de classes *wxPython* serviram para

construção das janelas e suas estruturas (botões, *checkbox*, *checkboxlist* etc.), formando assim um complexo de *interfaces* gráficas interagindo entre si por meio de métodos/eventos específicos. Estes métodos/eventos específicos foram implementados em uma ampla biblioteca de classe construída durante o presente trabalho, denominada *GUIGATEClass*.

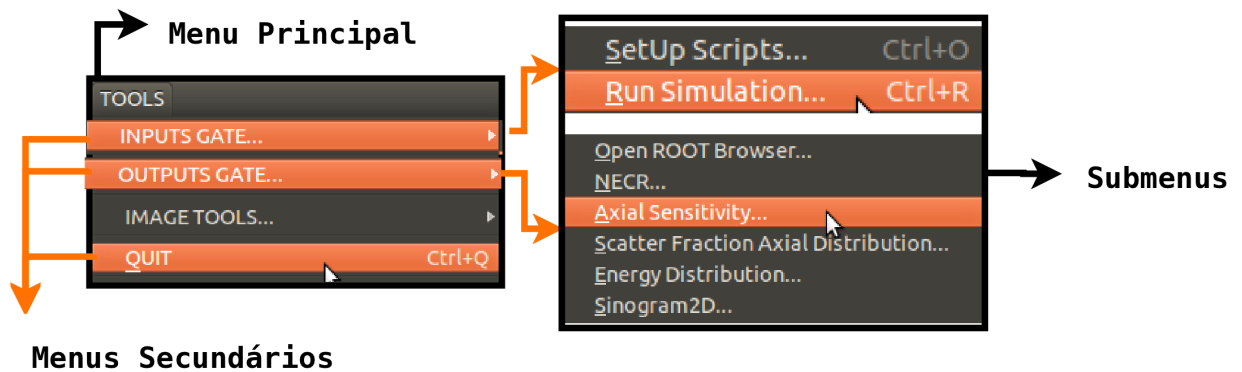
A Figura 21 ilustra como os processos no GUIGATE foram projetados, bem como o papel das bibliotecas de classes *wxPython* e *GUIGATEClass* na dinâmica do software.

Figura 21 – Arquitetura do *software* GUIGATE. Este foi criado afim de facilitar a interação do usuário com o GATE e com o ROOT, passando esse processo do terminal do Linux/Unix para uma interface gráfica. A classe *wxPython* fornece os métodos para criação das estruturas do software. As funcionalidades destas foram implementas numa biblioteca de classe própria, a *GUIGATEClass*, toda baseada na arquitetura de simulação do GATE.



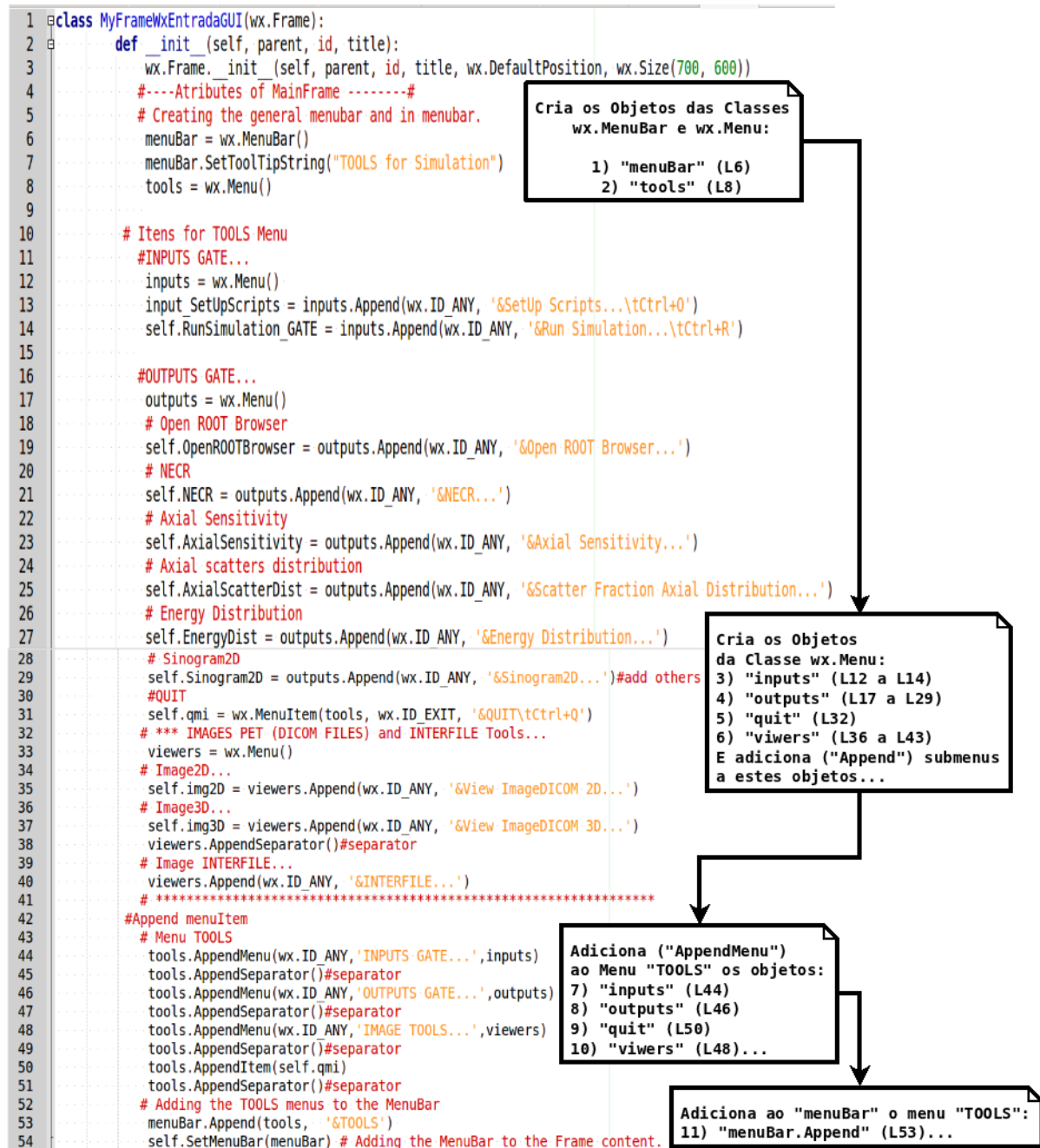
Para cumprir sua função, a *GUIGATEClass* foi construída considerando a arquitetura de simulação do GATE (Figura 20). Em seu escopo a *GUIGATEClass* fornece métodos específicos provenientes do GATE e/ou do ROOT como pode ser visto no diagrama da Figura 22.

Figura 23 – Árvore de *Menus* da janela principal. O menu "*IMAGE TOOLS*" aqui exibido não está com funções implementadas nesta primeira versão do GUIGATE.



A Figura 24 mostra um diagrama de blocos e o código implementado para criação dos menus mostrados na Figura 23.

Figura 24 – Diagrama de blocos e parte do código implementado para construção dos menus.



3.3.2 Secundárias:

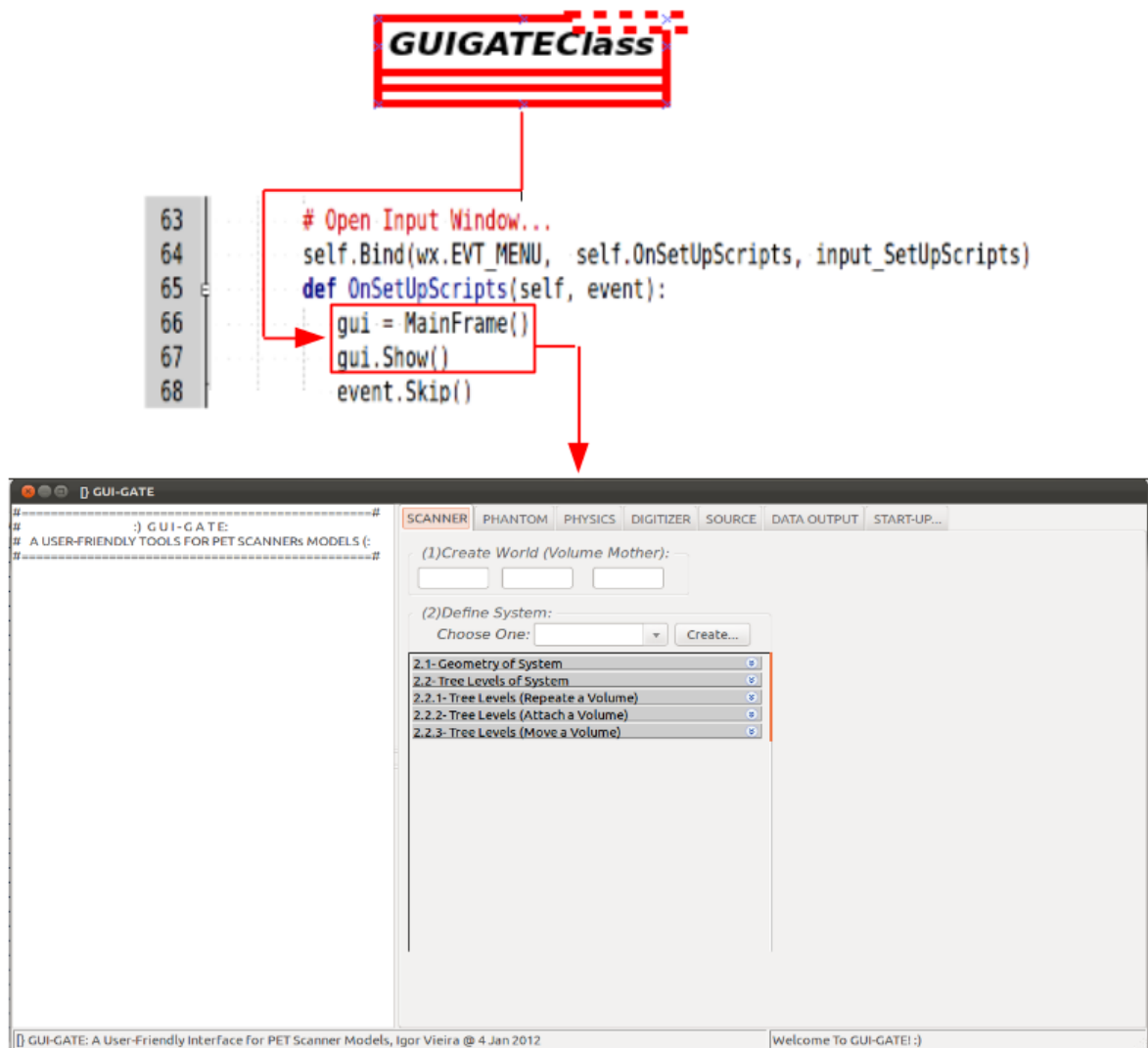
3.3.2.1 Inputs

Os processos associados ao que aqui denominou-se *inputs* se dividem na criação dos arquivos de entrada e na execução do modelo criado usando o GATE.

Para a criação dos arquivos de entrada, um evento de *click* no menu "SetUp Scripts..." da janela principal exhibe, graças a função mostrada na Figura 25, uma janela (chamada "GUI-

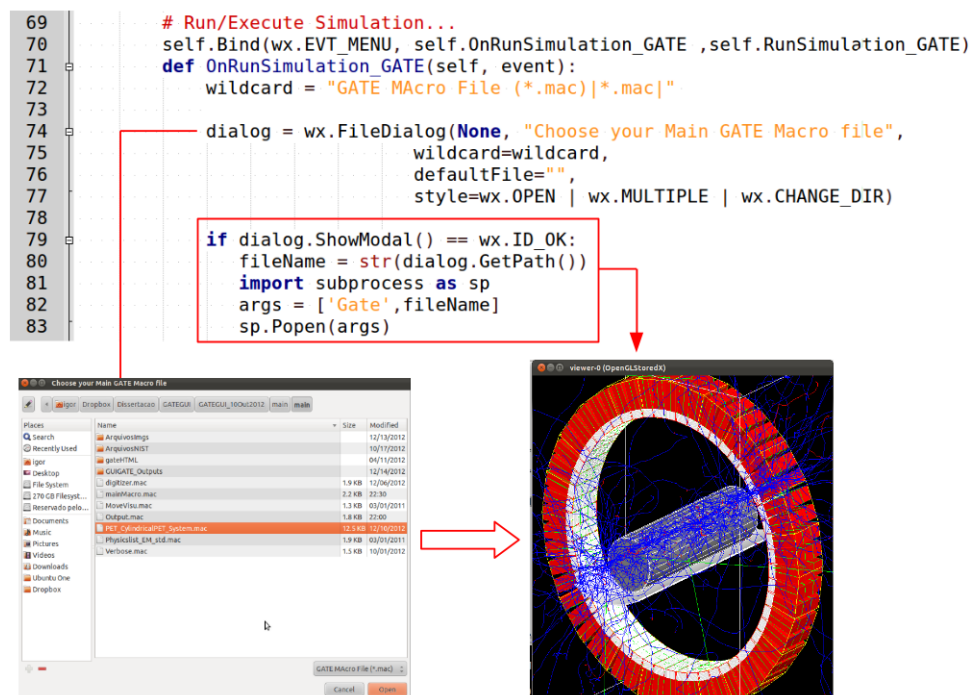
GATE Input"). Essa janela de entrada dispõe de muitos objetos (totalizando 5098 linhas de códigos) e foi dividida em sete *Tabs* principais, com cada uma correspondendo a uma classe, considerando o fluxo de simulação do GATE (Figura 20). Em síntese, esta janela acessa os métodos implementados na *GUIGATEClass* relativos a todas as classes agregadas à classe-mãe *GUIGATEImaging* (Figura 22). Isso lhe confere a possibilidade de criar comandos específicos do GATE em um arquivo de texto (".mac") e exibir para o usuário linha a linha no *panel* esses mesmos comandos, simplesmente por meio de eventos de *click* ou de *mouse* a medida em que vai ocorrendo a interação usuário-GUIGATE.

Figura 25 – Código implementado que permite o acesso a janela de criação dos arquivos de entrada, a qual deriva da classe-mãe "MainFrame".



No que se refere à execução do modelo, ao clicar no menu "Run Simulation...", a função mostrada na Figura 26 exibe uma janela do tipo "Open File Dialog" para o usuário selecionar na pasta "main", dentro da árvore de diretório do GUIGATE, a macro principal com extensão ".mac" criada na etapa anteriormente descrita. Após clicar "OK", o nome e o caminho do arquivo são armazenados na variável chamada *filename* (Linha 80) e a biblioteca nativa da linguagem Python, chamada *subprocess*, é usada (Linha 81 e 82). Como o próprio nome diz, essa biblioteca oferece métodos que permitem a interação com o sistema operacional. O método "Popen"(Linha 83), neste caso, inicializa o modelo selecionado executando o GATE.

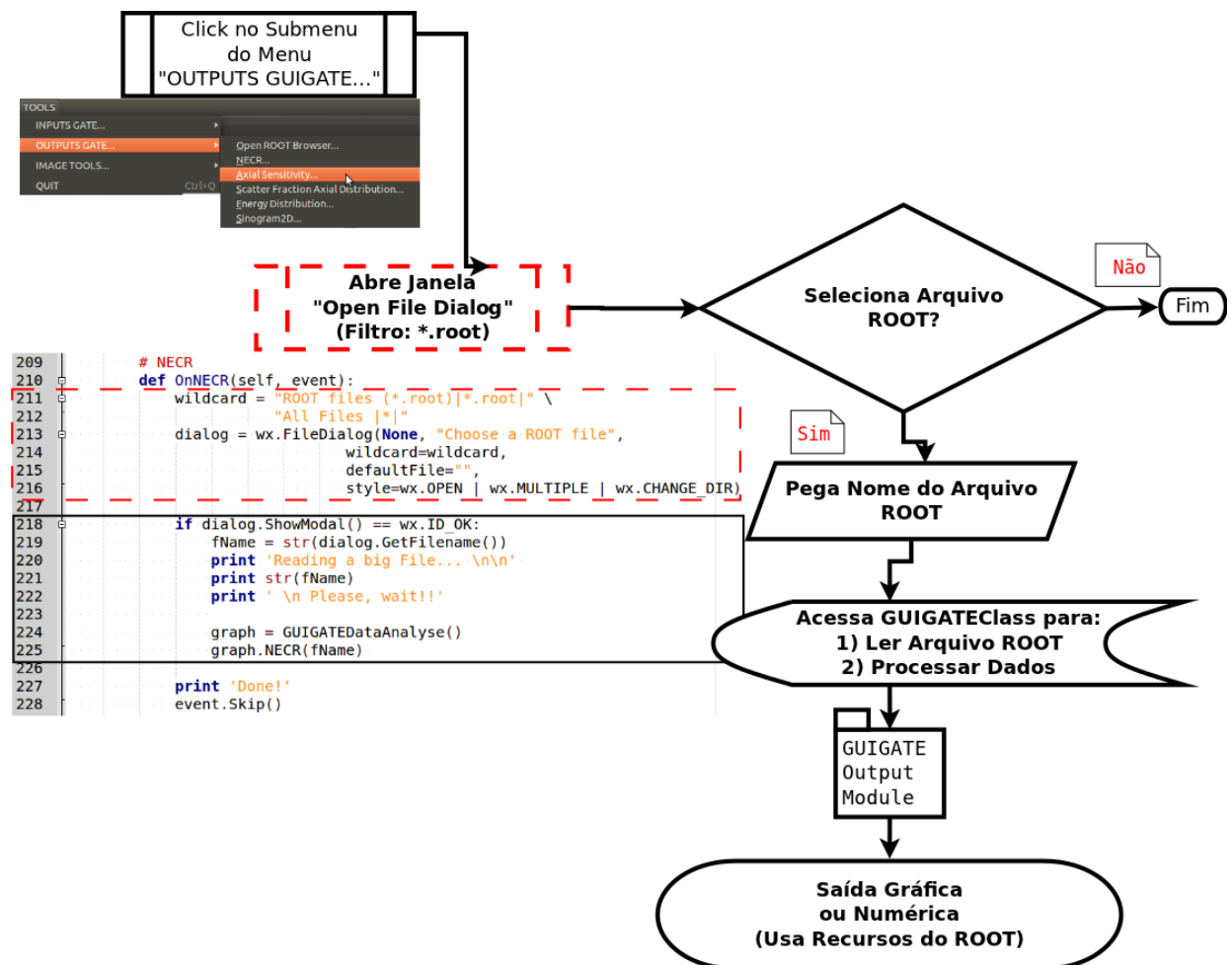
Figura 26 – Código implementado que permite a execução do GATE, quando selecionada a macro principal.



3.3.2.2 Outputs

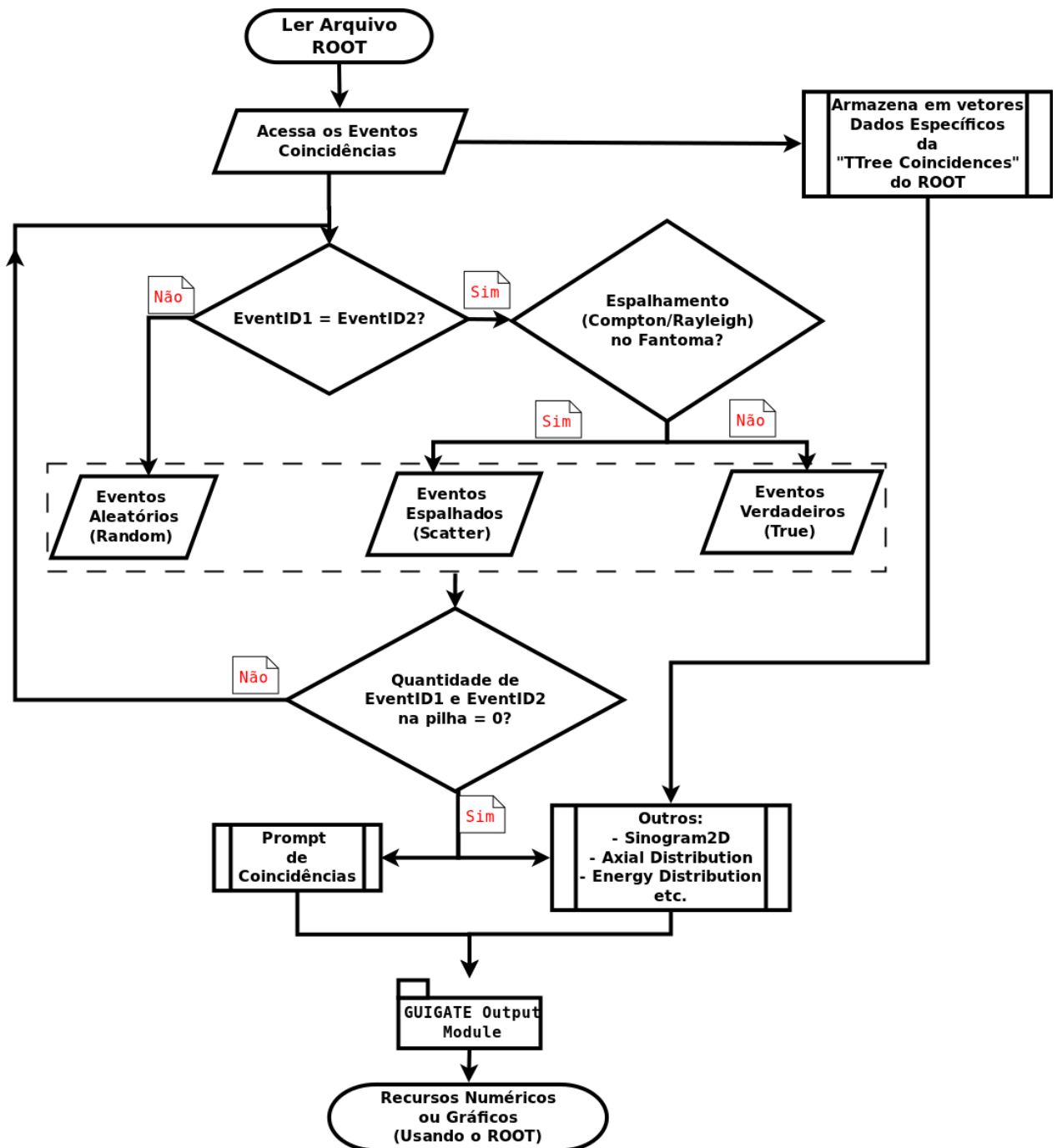
A criação do acesso a todas as janelas de análise do *output* segue um mesmo princípio: ao clicar em um dos *submenus* do menu *OUTPUTS GATE...* (Figura 23), uma janela do tipo *Open File Dialog* é aberta. Quando confirmado o arquivo selecionado, métodos da classe *GUIGATE-Class* são usados para ler e processar os dados contidos no arquivo ROOT ("**.root*"), fornecendo como saída recursos numéricos ou gráficos. Neste último caso, integrando no *GUIGATE Output Module* todas as funcionalidades do analisador de dados ROOT versão 5.32 (Figura 27).

Figura 27 – Princípio geral de acesso a janela *GUIGATE Output Module*.



O que vai diferenciar um algoritmo do outro nesses *submenus* é o modo de processamento dos dados, conforme as características do processo em estudo. O mais fundamental aqui é que independente do *submenu* selecionado, todos dependem dos processos básicos de eventos coincidências do PET simulado, ou seja, vão levar em conta um ou mais eventos coincidência, quer sejam verdadeiros (i.e, *NECR*, *Sinogram2D*, *Axial Sensitivity* etc.), espalhados (i.e, *NECR*, *Sinogram2D*) e/ou aleatórios (*NECR*). A Figura 28 ilustra esse processo geral sob a forma de um diagrama.

Figura 28 – Diagrama geral implementado na *GUIGATEClass* para exibir na janela *GUIGATE Output Module* a perspectiva de análise selecionada: *NECR*, *Axial Sensitivity*, *Scatter Fractional Axial Distribution*, *Energy Distribution* ou *Sinogram2D*.



3.4 PET SCANNER A SER SIMULADO

O PET Scanner a ser simulado afim de inicialmente validar o GUIGATE é um scanner de corpo inteiro que não corresponde a nenhum tomógrafo comercial existente e se encontra disponível na distribuição do GATE para testes gerais, o chamado *benchmark* (USER'SGUIDE,

2012). O modelo consiste de uma geometria cilíndrica, composta de oito arranjos de detectores (*heads*), com 88 cm de diâmetro e 40 cm de comprimento axial. Um único arranjo (*head*) é feito de 400 blocos (*blocks*), cada bloco com uma matriz 5 x 5 de dupla camada de cristais (BGO-LSO).

O sistema *cylindricalPET* foi usado para descrever a lógica de organização do scanner. A relação entre os componentes físicos do scanner e os níveis hierárquicos do modelo são descritos na Tabela 6.

Tabela 6 – Relação entre os componentes físicos do scanner e o nível hierárquico do *CylindricalPET*.

Componente Físicos	Nível do Sistema
head	rsector
module	module
block	submodule
crystal	crystal
LSO	layer0
BGO	layer1

Cada *head* está blindado no fim do campo de visão com 1 mm de uma chapa retangular de chumbo. Além disso, septos retangulares de 0.5 mm de tungstênio são colocados para reduzir a probabilidade de detecção de fótons espalhados.

Os cristais de BGO e LSO foram definidos como detectores sensíveis (*sensitive detectors*). Ruído Gaussiano com 26% de FWHM (*Full Width at Half Maximum*) a 511 keV foi aplicado para energia depositada nos cristais. Uma janela energética detecta os fótons entre 350 e 650 keV. Uma coincidência é definida quando os dois fótons são detectados na janela temporal, neste caso, 120 ns.

O fantoma, por sua vez, consiste de um cilindro de água com 20 cm de diâmetro e 70 cm de altura. O mesmo foi centralizado no centro do campo de visão. Dentro do fantoma de água, foram inseridas duas fontes lineares de 68 cm de altura e 1 mm de diâmetro, radialmente separadas por 2 cm do eixo do cilindro. A primeira fonte é formada de F^{18} (Flúor-18) e a segunda de O^{15} (Oxigênio-15), ambas com atividade inicial de 100 kBq. A meia-vida dos emissores de pósitrons simulados foram explicitamente definidas como 6586.2 s e 122.2 s, respectivamente.

O pacote padrão para simulação de processos eletromagnéticos foi usado para espalhamento Compton e efeito fotoelétrico. O pacote para baixa energia foi usado apenas para tratar do espalhamentos Rayleigh.

A simulação da aquisição do scanner foi definida com 4 minutos de duração, o que representa duas meia-vidas do O^{15} . Estas aquisições foram divididas em dois *frames* de 2 minutos cada.

Uma descrição detalhada de toda a simulação pode ser vista no Apêndice B.

4 RESULTADOS

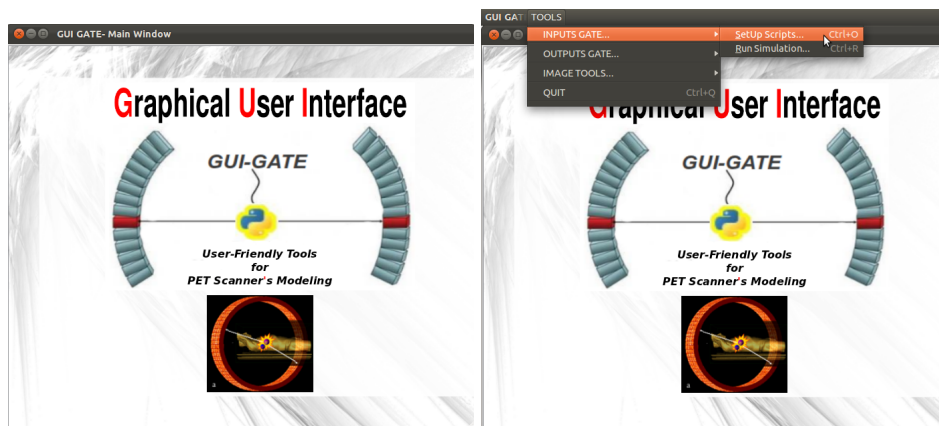
Neste capítulo são mostradas, passo a passo, as funcionalidades básicas dessa primeira versão do GUIGATE.

4.1 INPUTS NO GATE VIA GUIGATE

4.1.1 Passo 1: Acessando a Janela de Inputs

Para criar seus dados de entrada, organizando-o didaticamente em um arquivo de texto, o usuário precisa acessar, na janela principal do GUIGATE (Figura 29), o seguinte caminho **TOOLS** → **Inputs GATE ...** → **Setup Scripts (Ctrl+O)...**. Feito isto, abrir-se-á a janela de **Inputs** mostrada na Figura 30.

Figura 29 – Janela Principal do GUIGATE mostrando o Menu **TOOLS** para acessar aos módulos de construção dos setups de entrada no GUIGATE, tais como: Scanner, Phantom, Physics etc.

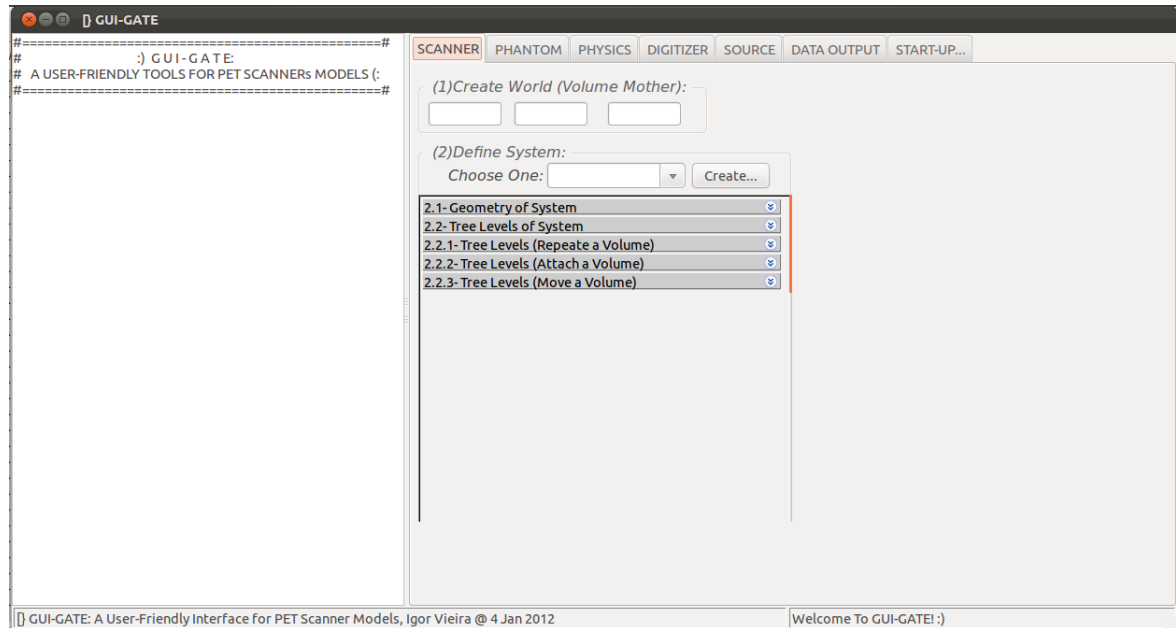


Como mostra a Figura 30, na janela o usuário define todos os *inputs* ao transitar numa janela única pelas abas: "**Scanner**", "**Phantom**", "**Physics**", "**Digitizer**", "**Source**", "**Data Output**" e "**Start-Up**". Enquanto os parâmetros de entrada vão sendo definidos no GUIGATE por meio de eventos de *click* ou mesmo da tecla *enter*, a depender do tipo de dado, dois processos estão ocorrendo em paralelo:

(1) o "*textbox*" ao lado do conjunto de abas mostra linha a linha todos os parâmetros definidos pelo usuário, auxiliando-o na aprendizagem dos comandos do GATE. Além disso, ao colocar o *mouse* sobre a *interface*, mensagens (por meio de *tooltips*) autoexplicativos dão instruções básicas sobre aquele tipo de dado, baseado no manual de usuário do GATE; e

(2) o conjunto de comandos definidos no GUIGATE em cada aba são automaticamente salvos em arquivos de texto no formato lido pelo GATE (extensão ".mac"). Por exemplo, os dados relativos a geometria do *scanner*, geram o arquivo "*Scanner.mac*" e os relativos à geometria do fantoma geram o "*Phantom.mac*" e assim sucessivamente.

Figura 30 – Janela de Inputs do GUIGATE.

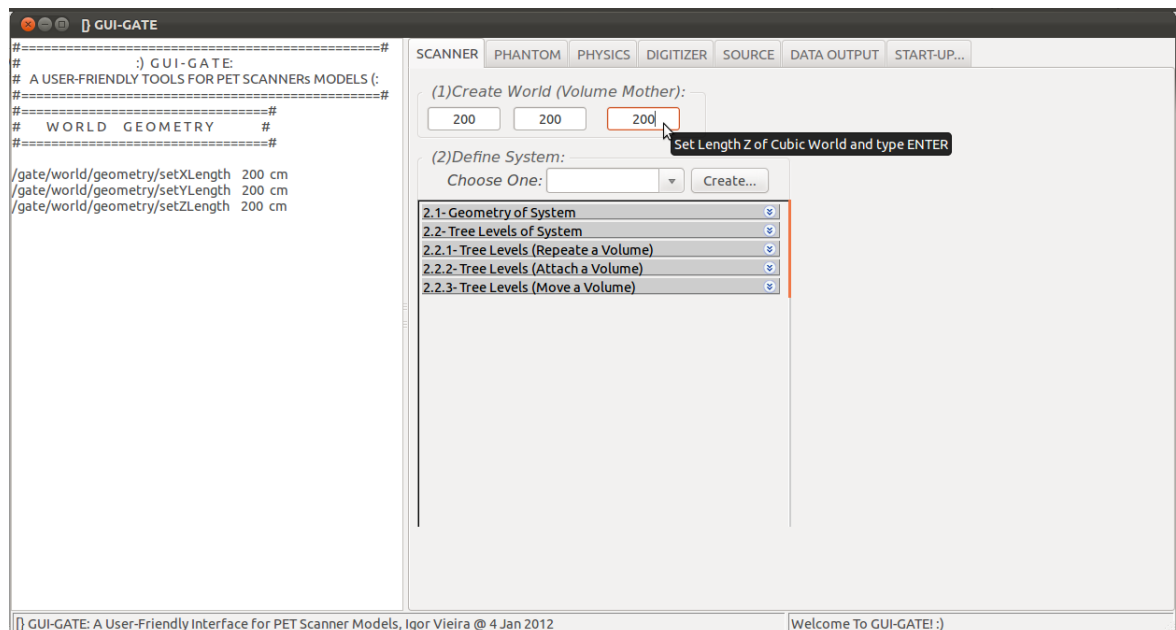


Nos passos subsequentes serão mostradas as abas já implementadas, bem como uma visão geral das funcionalidades disponíveis em cada uma delas.

4.1.2 Passo 2: Construindo a Geometria do Scanner

Por padrão no GATE, todas as geometrias são filhas de *world*, um cubo de tamanho suficiente para conter todas as geometrias da simulação. A definição do mesmo é realizada no campo "(1) Create world (Volume Mother)" como mostra a Figura 31, ao serem passados pelo usuário as arestas do cubo.

Figura 31 – Mostrando a construção do *world* na janela de construção do *scanner*.

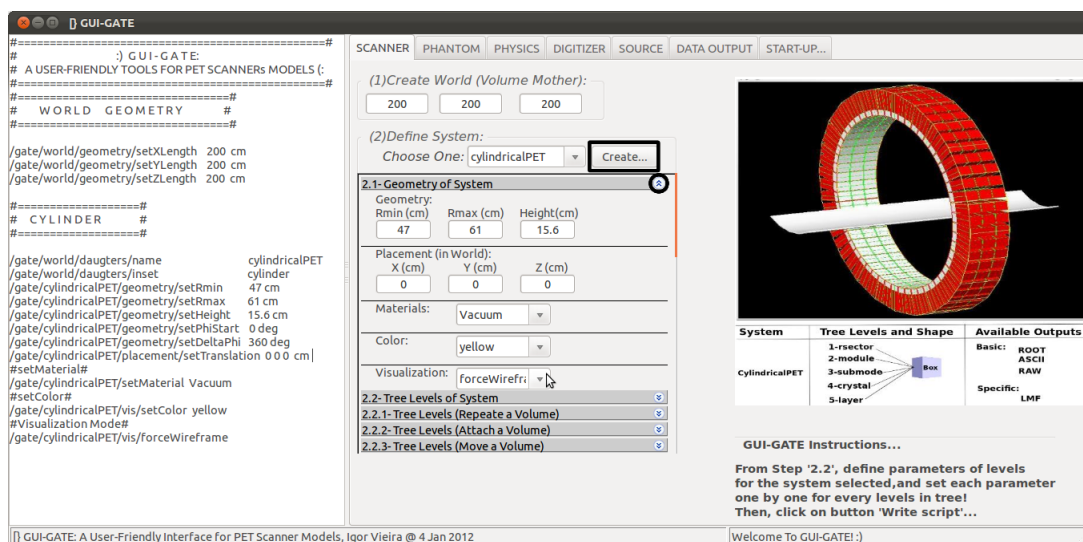


De todos os *frameworks* disponíveis para modelar sistemas PET no GATE, apresentar-se-á aqui a modelagem do "CylindricalPET", que pode ser usado como base para modelar muitos dos PETs clínicos e pré-clínicos existentes no mercado. Isto é realizado no *combobox* "Choose One" do campo "(2) Define System". Selecionado o sistema ("CylindricalPET", por exemplo), uma figura exemplo do sistema é mostrada com toda a estrutura hierárquica do mesmo, bem como informações gerais como mostra a Figura 32. Ao clicar no botão "Create...", habilita-se a construção do sistema selecionado com os eventos específicos do mesmo, nos diversos *FolderPanelBar*, estruturas enumeradas na Figura 32, a saber: (a) "(2.1) Geometry of System", (b) "(2.2) Tree Levels of System", (c) "(2.2.1) Tree Levels(Repeat a Volume)", (d) "(2.2.2) Tree Levels(Attach a Volume)", (e) "(2.2.3) Tree Levels(Move a Volume)".

(a) "(2.1)- Geometry of System"

Nesta barra, os parâmetros gerais relativos a geometria do sistema são definidos (Figura 32).

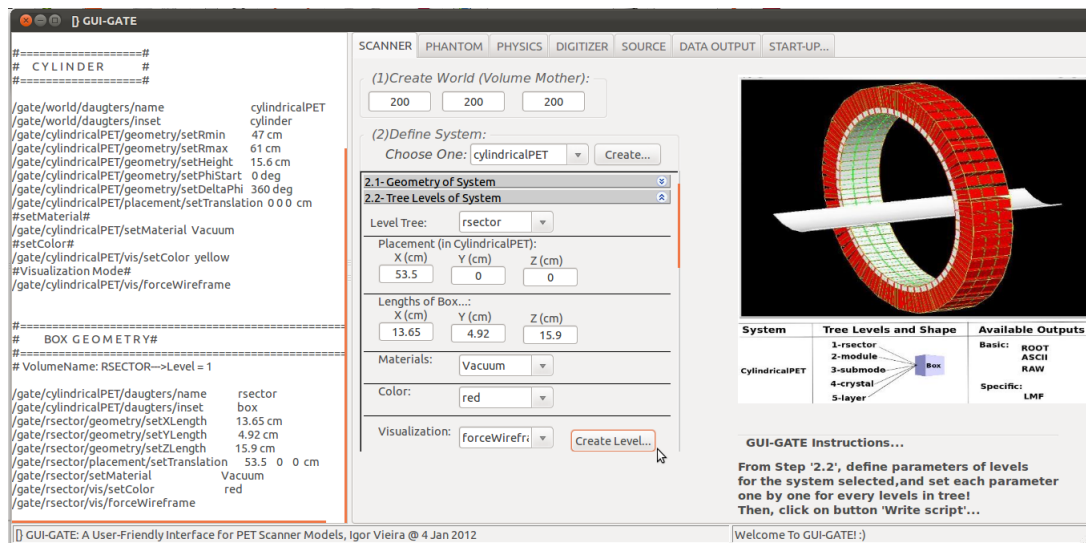
Figura 32 – Janela de Construção do *scanner*, mostrando construção da geometria do sistema.



(b) "(2.2)- Tree Levels of System"

Neste campo, por sua vez, por meio do *combobox* "**Level Tree**" (Figura 33) o usuário seleciona ordenadamente cada um dos cinco níveis hierárquicos (*Rsector, Module, SubModule, Crystal, Layer*) que compõem o sistema.

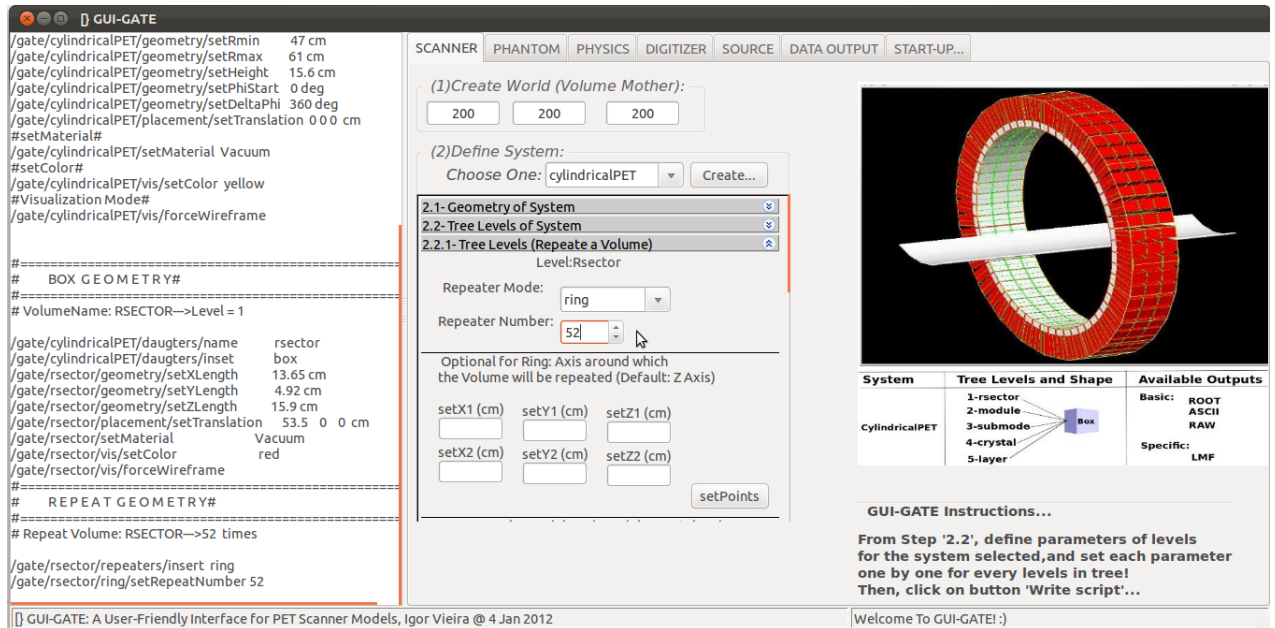
Figura 33 – Janela de Construção do *scanner*, mostrando a construção dos cinco níveis hierárquicos (*Rsector, Module, SubModule, Crystal, Layer*) que compõem o sistema.



(c) "(2.2.1)- Tree Levels(Repeat a Volume)"

Ao chegar nesse item, dispõe-se da possibilidade de repetição de volumes, conforme o nível hierárquico em construção. Na Figura 34, por exemplo, o nível é o *Rsector*, cujo modo de repetição é o modo *ring* por definição.

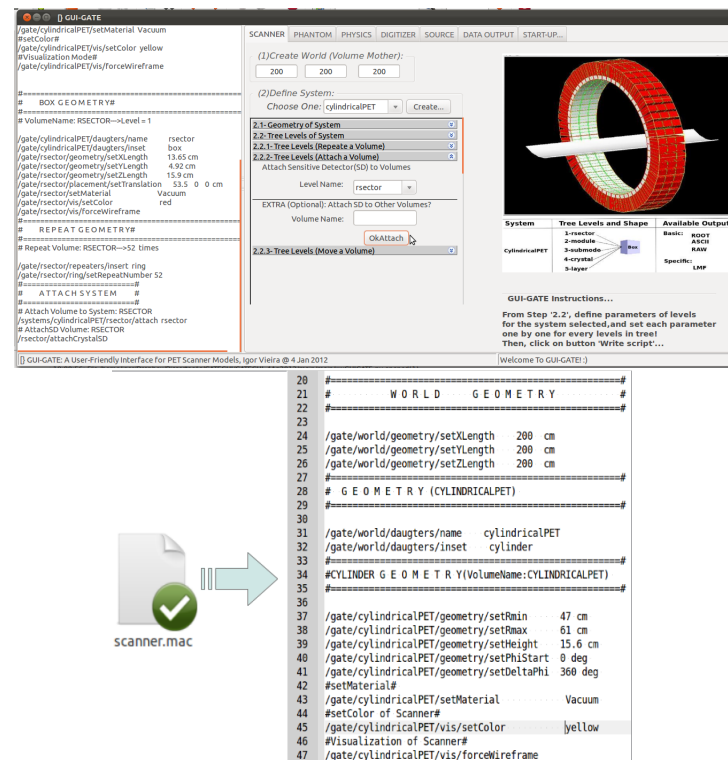
Figura 34 – Janela de Construção do *scanner*, mostrando a construção de repetições de geometrias.



(d) "(2.2.2)- Tree Levels(Attach a Volume)"

Uma etapa fundamental nas modelagens do GATE, como foi visto, é o *Attach Volume*: nesta etapa é definido se os processos físicos que eventualmente ocorram naquela geometria será armazenado ou não. A Figura 35 mostra como é realizada no GUIGATE essa escolha.

Figura 35 – Janela de Construção do *scanner*, mostrando (superior) a construção de *attach* de geometrias e (inferior) o arquivo "Scanner.mac" com parte dos parâmetros definidos na construção da geometria e salvos no mesmo.



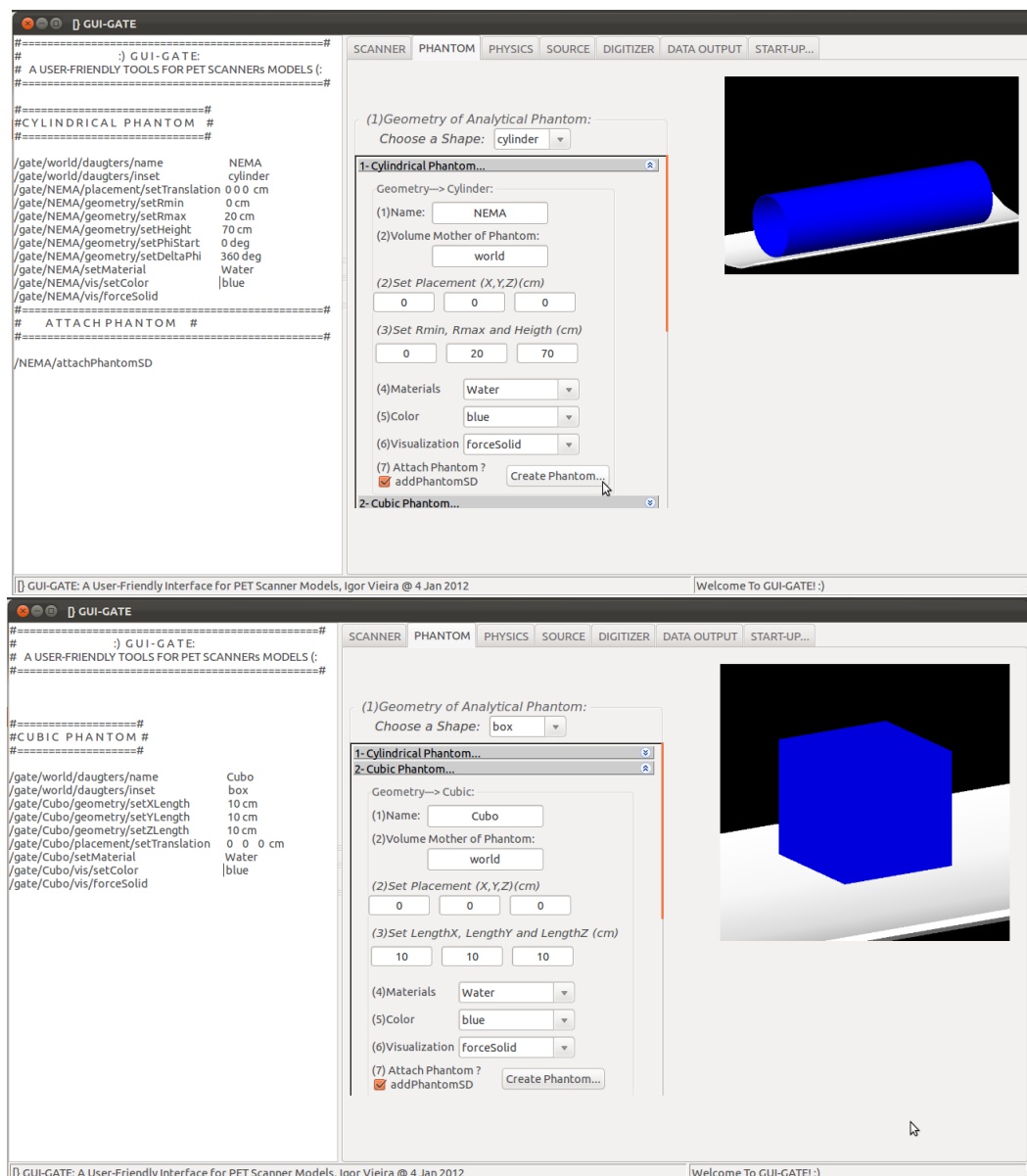
Ao finalizar esse passo, o usuário terá o seu modelo parcialmente construído e podendo explorar de modo didático o arquivo de texto criado ("Scanner.mac") se tiver o interesse de aprender sobre a estrutura do GATE na modelagem de *scanner*.

4.1.3 Passo 3: Construindo a Geometria do Fantoma

No GATE é possível simular dois tipos de fantasmas: (1) fantasmas analíticos (cubo, cilindros etc.) e (2) fantasmas de voxel. Na atual versão do GUIGATE, apenas a simulação dos fantasmas analíticos está disponível como mostra a Figura 36.

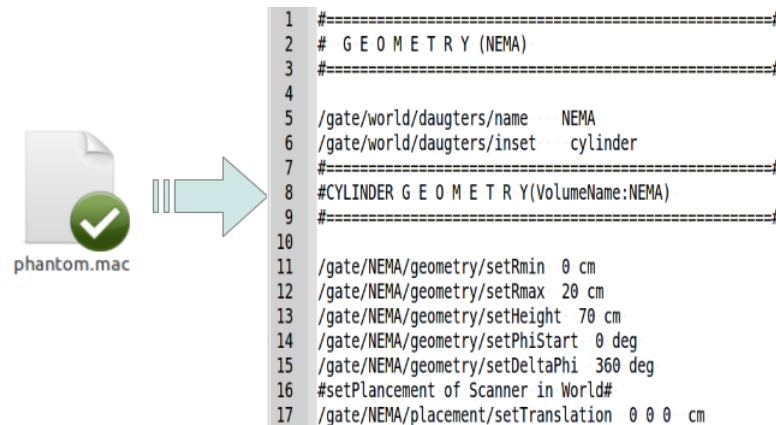
A principal razão para isso é que esta primeira versão foi planejada para simulação dos protocolos NEMA, cujos fantasmas são de geometria cilíndrica (com diâmetro fixo de 20 cm e altura de 70 cm). Contudo, além de fazer as alterações que julgar necessário em parâmetros como tipo de material ou dimensões dos simuladores NEMA, é possível testar formas geométricas diferentes como possíveis novos fantasmas.

Figura 36 – Janela de Construção dos Fantasmas Cúbicos e Cilíndricos.



A Figura 37 mostra o script gerado com os parâmetros relativos ao fantoma.

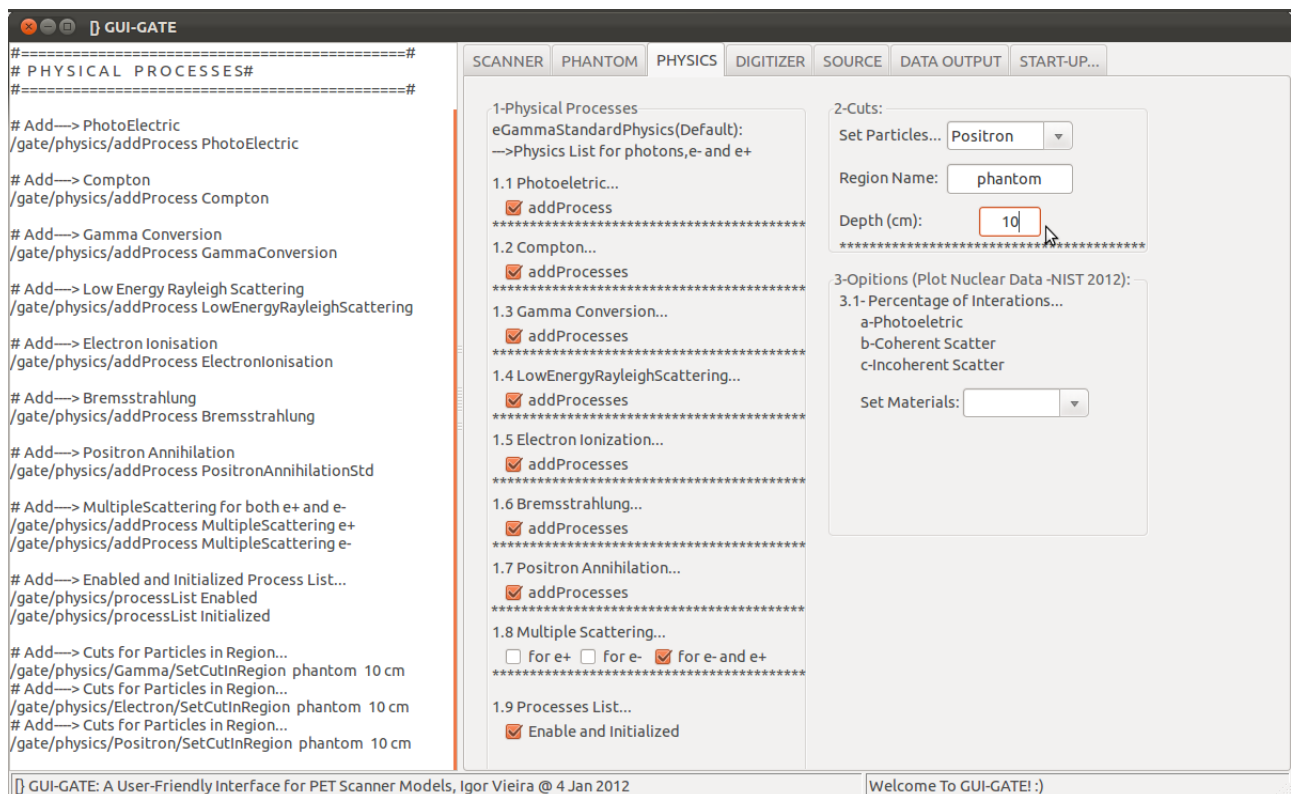
Figura 37 – Mostra arquivo "Phantom.mac" e parte dos parâmetros definidos na construção de um fantoma cilíndrico baseado no fantoma NEMA padrão.



4.1.4 Passo 4: Construindo Processos Físicos

Os processos físicos que podem ser simulados no GUIGATE estão contidos nos campos: "(1) Physical Process" e "(2) Cuts" e são mostrados na Figura 38. No campo (1) por meio de "checkbox" é possível selecionar os processos de interesse. No campo (2), por sua vez, as profundidades de corte são definidas na região de interesse considerando o tipo de partícula no "combobox" ("set Particles...").

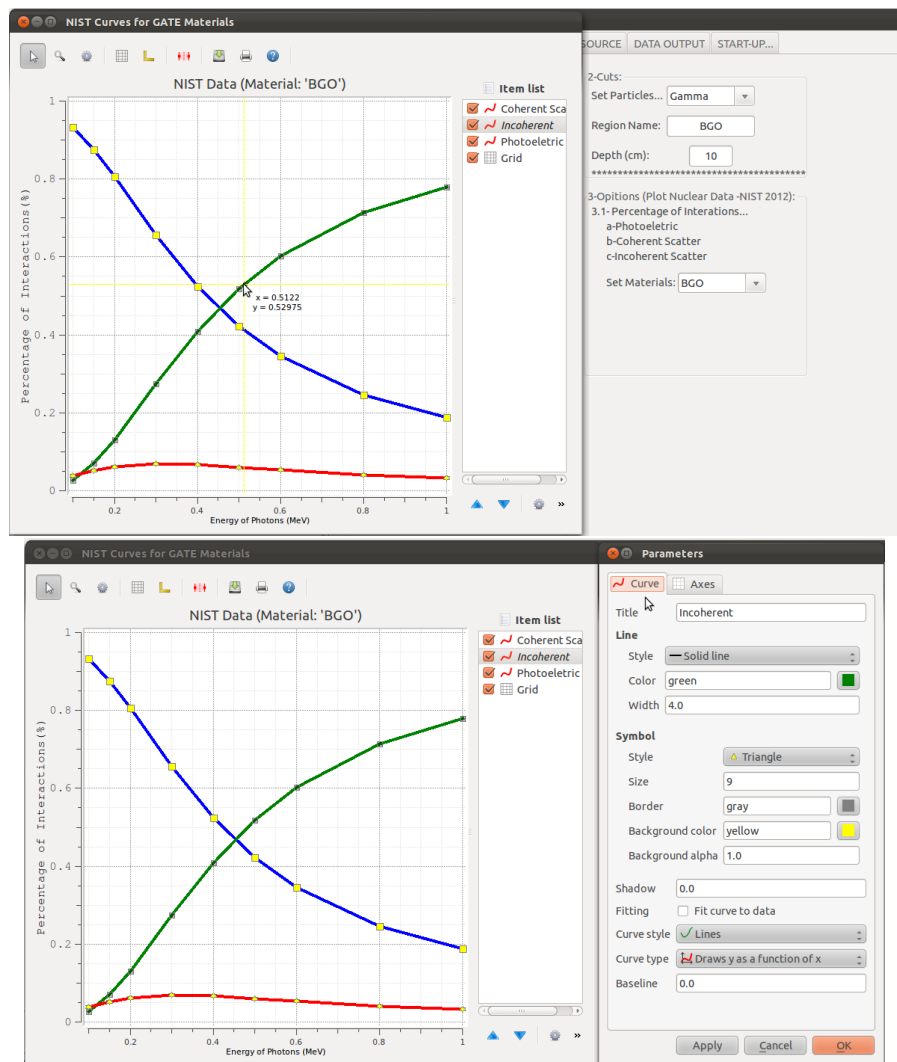
Figura 38 – Janela de Construção dos possíveis processos físicos no GATE.



No campo "3- Options ..." é possível visualizar curvas mostrando a probabilidade de interação do efeito Fotoelétrico, bem como espalhamentos coerentes e incoerentes em função da

energia para cada material simulado (Figura 39) ou qualquer outro de interesse. Os dados utilizados para tal são provenientes dos coeficientes atenuação mássico do NIST (<http://physics.nist.gov/PhysRefData/Xcom/html/xcom1.html>).

Figura 39 – Janela opcional aberta quando o usuário seleciona um material no "combobox" ("set Materials") do campo "3- Options ...". Ao lado, alguns recursos de edição dos gráficos.

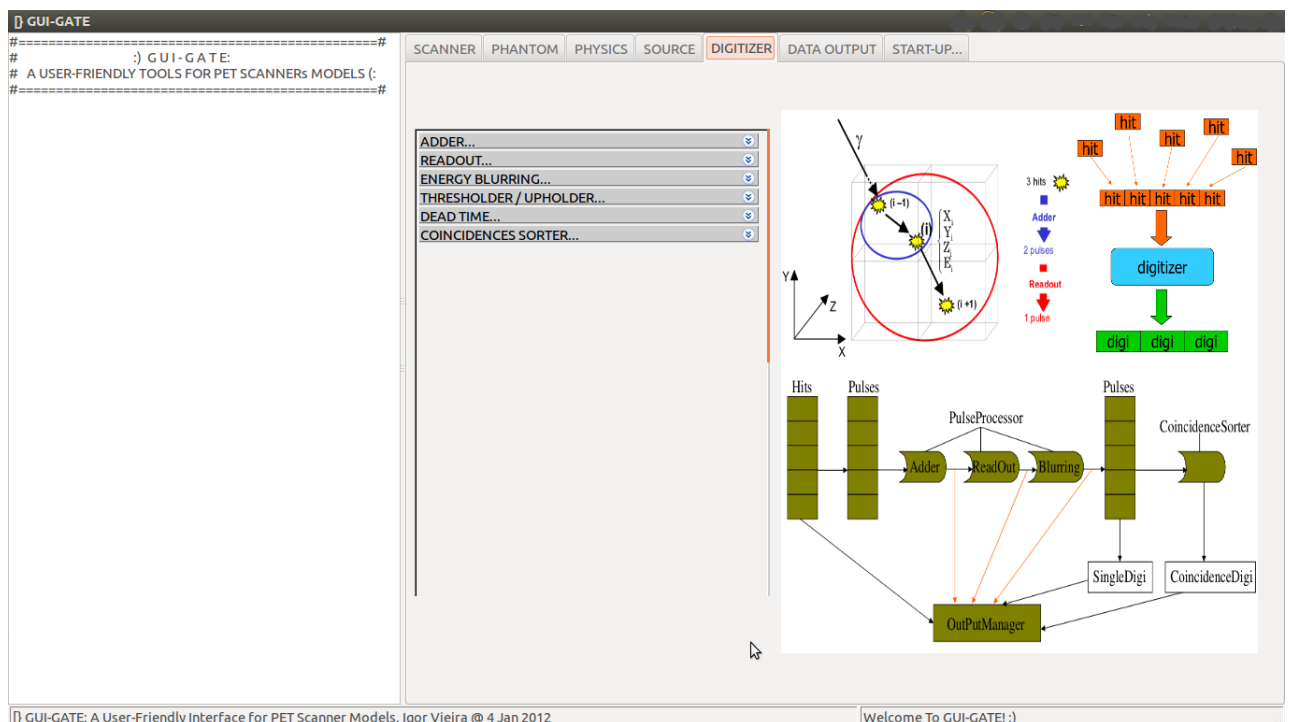


4.1.5 Passo 5: Construindo a Eletrônica dos Detectores

Um dos grandes diferenciais do GATE, quando comparado aos outros códigos dedicados à modelagem Monte Carlo de PET, é a possibilidade de transformar as interações ou processos físicos individuais (no GEANT chamado de *hit*) que ocorrem em uma dada geometria, em um sinal ou impulso elétrico tal como ocorre no detector. Essa mimetização do funcionamento de um detector real é realizado pelo módulo *Digitizer*.

No GUIGATE, os módulos disponíveis para modelar os canais de processamento do *Digitizer* são (Figura 40): "**Adder / AdderCompton**", "**Readout**", "**Blurring**", "**Thresholder / Upholder**", "**Dead Time**", "**Coincidences Sorter**".

Figura 40 – Modulo Digitizer.



A seguir, cada uma das possibilidades são brevemente analisadas.

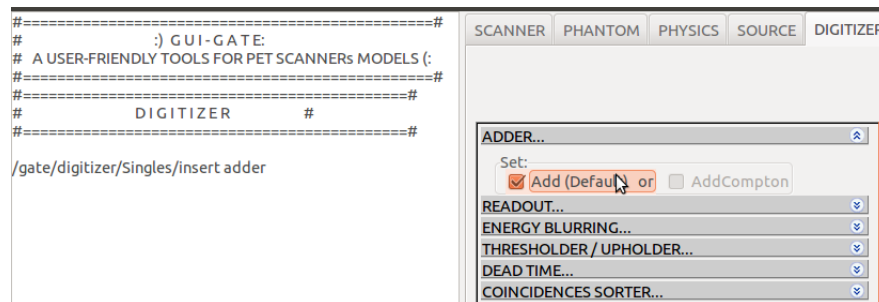
Adder/Addercompton

Este módulo reagrupa *hits* por volume. Por exemplo, se uma partícula que atinge o detector realiza 3 *hits* dentro de dois cristais diferentes antes de ser parada, a saída do módulo *Adder* consistirá de dois pulsos, como mostrado na Figura 41. Cada pulso é computado como se segue: a energia é considerada como total de energias em cada volume, a posição é obtida com um ponderação da posição de cada *hit*, tendo a energia de cada um destes como fator de peso. O tempo computado é igual ao tempo do primeiro *hit*.

Essas distinções se devem ao fato de o sinal eletrônico ser medido integralmente, sem uma resolução energética e temporal suficientes para distinguir entre interações de partículas dentro do cristal.

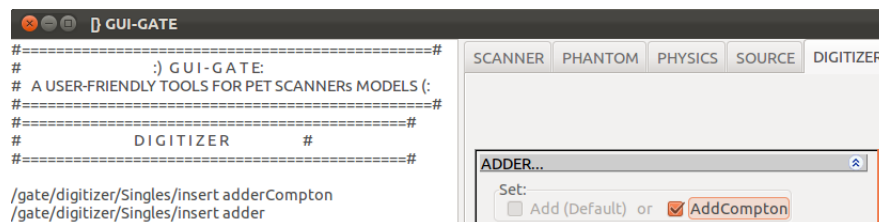
Isso pode ser feito no GUIGATE por meio da seleção de "*checkbox*"(Figura 41).

Figura 41 – Submódulo Add no GUIGATE.



O módulo *AddCompton*, por sua vez, tem um comportamento diferente do *Adder* clássico, que usa a energia como fator de ponderação para determinar a posição, adicionando todos os *hits* provenientes de elétrons ou fótons. Ao invés disso, para cada energia depositada por elétrons, esta é adicionada ao *hit* por fótons prévios no mesmo volume ID (ou descartado se fora), mas a localização permanece a da interação do fóton. Desse modo, as interações Compton tornam-se registradas para interações por fótons, viabilizando estudos futuros. O usuário deve usar o *Adder* clássico depois do *AddCompton*, para lidar com múltiplas interações de fótons no mesmo volume. A Figura 42 mostra a implementação no GUIGATE.

Figura 42 – Submódulo AddCompton no GUIGATE.



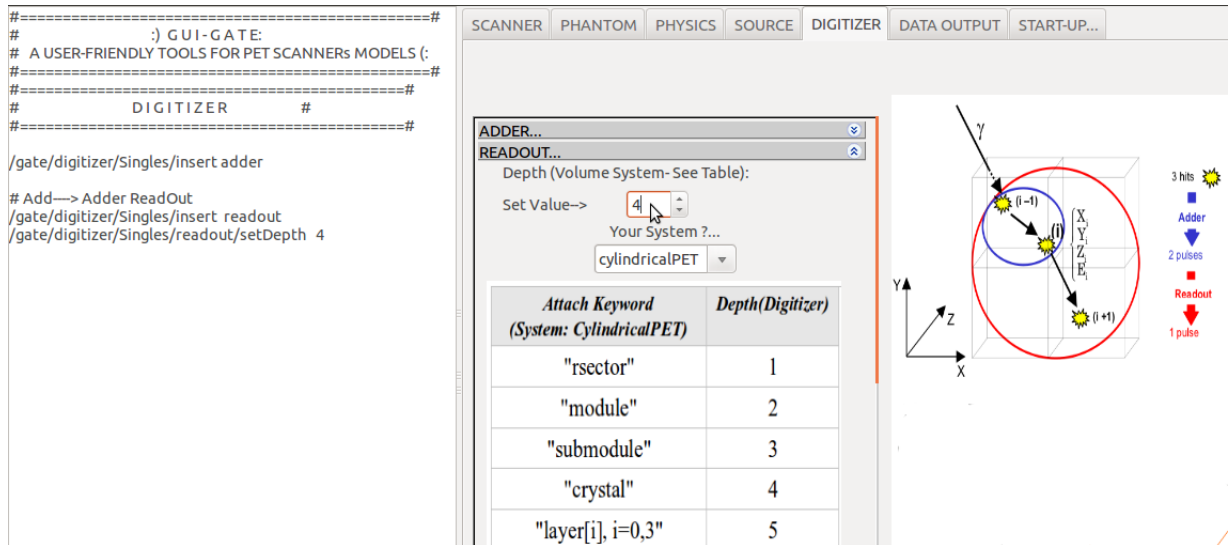
Readout

O módulo **readout** reagrupa os pulsos por bloco (grupo de detectores sensíveis). Esse agrupamento tem que ser determinado pelo usuário por meio da variável chamada **depth**, que reflete o nível dentro da hierarquia da geometria onde ocorrerá a contagem. Usando essa variável os pulsos são somados se os ID's do volume são do mesmo nível da variável **depth**. Os resultados desse módulo são a energia total por bloco e a posição do pulso com máxima energia.

O resultado que viabiliza a implementação desse processo no GUIGATE é mostrado na Figura 44. No combobox o usuário seleciona seu sistema pré-definido, e pode visualizar em uma tabela que é exibida os níveis hierárquicos da geometria do sistema, facilitando assim a definição do parâmetro **readout**, que é fundamental para uma precisão maior na modelagem.

Blurring

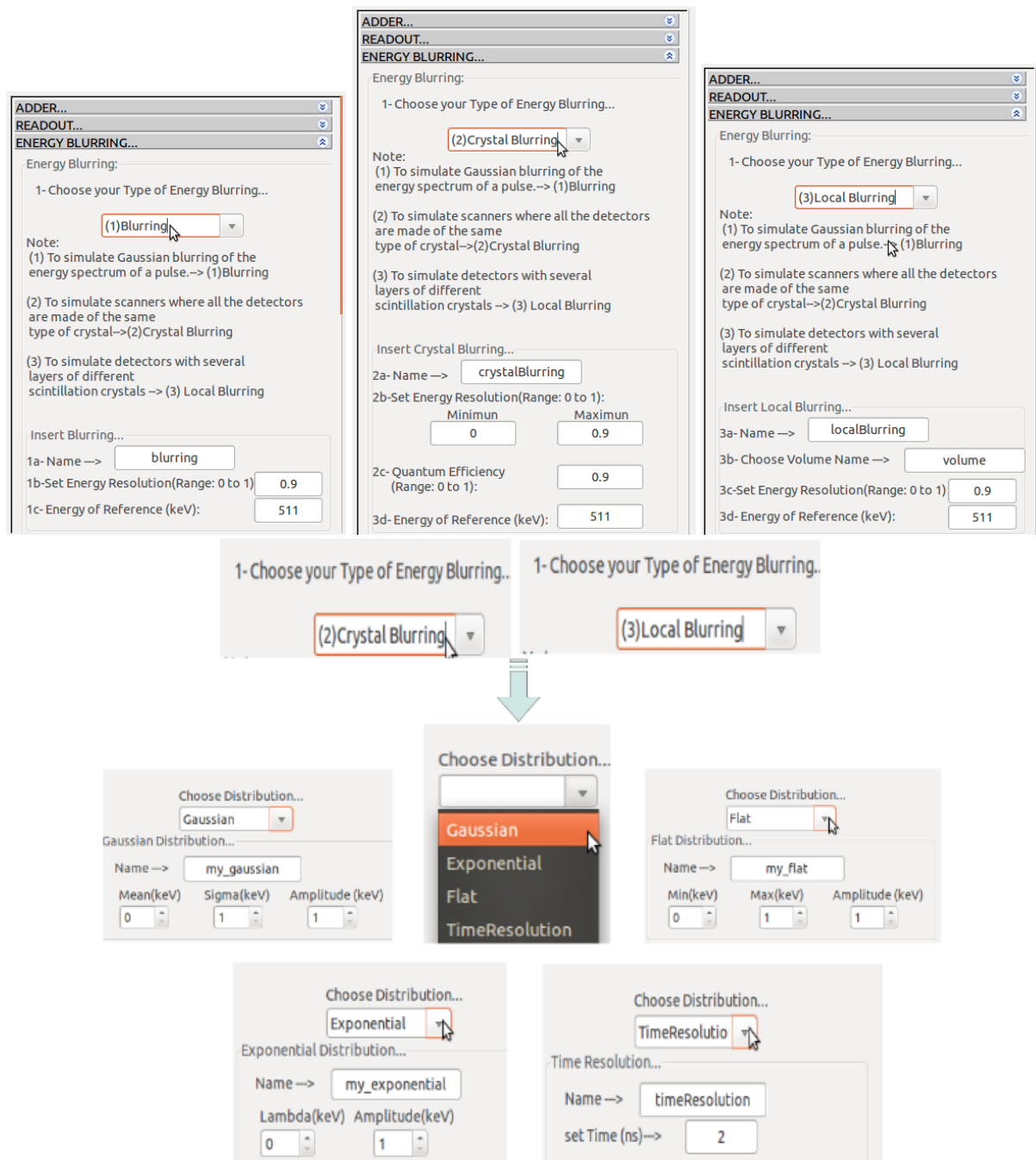
Após o "*ReadOut Module*", o usuário poderá simular: (1) "*Blurring*" afim de projetar um espectro energético de pulsos com tendência Gaussiana, (2) "*Crystal Blurring*" no caso de *scanners* com todos os detectores feitos do mesmo tipo de cristal e (3) "*Local Blurring*" no caso do PET *scanner* possuir matrizes de detectores com diferentes cristais cintiladores.

Figura 43 – Submódulo Readout no GUIGATE.

Em cada um desses casos o usuário definirá a faixa da resolução energética, a eficiência quântica (QE) bem como a energia de referência (keV). Especificamente no caso (3), será necessário definir o nome do volume ou camada para a qual aqueles valores serão atribuídos.

Por outro lado, enquanto no caso (1) o usuário dispõe apenas da possibilidade de simular uma distribuição de probabilidade com tendência Gaussiana; nos casos (2) e (3) as possibilidades são um pouco maiores, sendo possível simular, além de uma distribuição Gaussiana ("*Gaussian Distribution*"), distribuições Exponenciais ("*Exponential Distribution*"), Flat ("*Flat Distribution*") e Temporal ("*Temporal Distribution*"), cada uma com seus respectivos parâmetros de entrada (Figura 44).

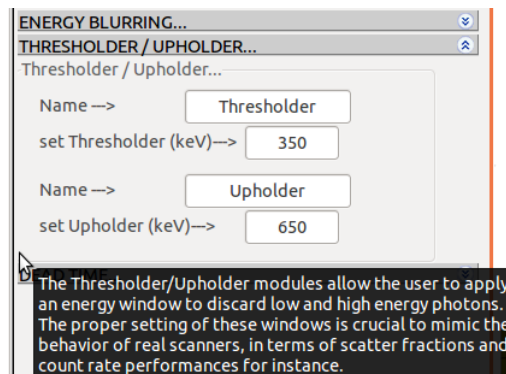
Figura 44 – Submódulo "Energy Blurring"no GUIGATE, com os tipos de distribuição de probabilidade disponíveis.



Thresholder/Upholder

O "*THRESHOLDER/UPHOLDER*" permite a definição de uma janela energética, abaixo ou além da qual, os fótons que atingem os detectores são descartados. A menor energia de corte definida pelo usuário representa o "*Threshold*" e a maior o "*Upholder*". São parâmetros cruciais para reproduzir taxa de contagens e fração de espalhamento, por exemplo (Figura 45).

Figura 45 – Submódulo "*THRESHOLDER/UPHOLDER*", no GUIGATE. Parâmetros cruciais para reproduzir taxa de contagens e fração de espalhamento em simulações realistas de PET scanners.

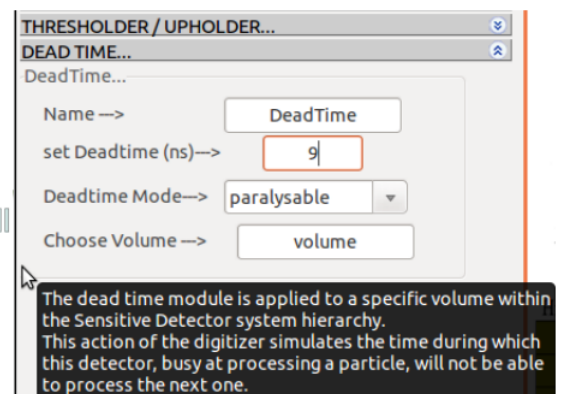


Dead Time

O tempo morto ("*dead time*") de detecção pode ser simulado no GUIGATE como mostra a Figura 46. Os dois modos de tempo morto estão disponíveis: *paralysable* e *nonparalysable*. Esses modelos podem ser implementados evento por evento durante a simulação, o que a torna mais realista.

Figura 46 – Submódulo "Dead Time" no GUIGATE. Permite simular o tempo morto do sistema de detectores, ou seja, o tempo que o sistema de detecção fica inativo até estar apto para a subsequente detecção.

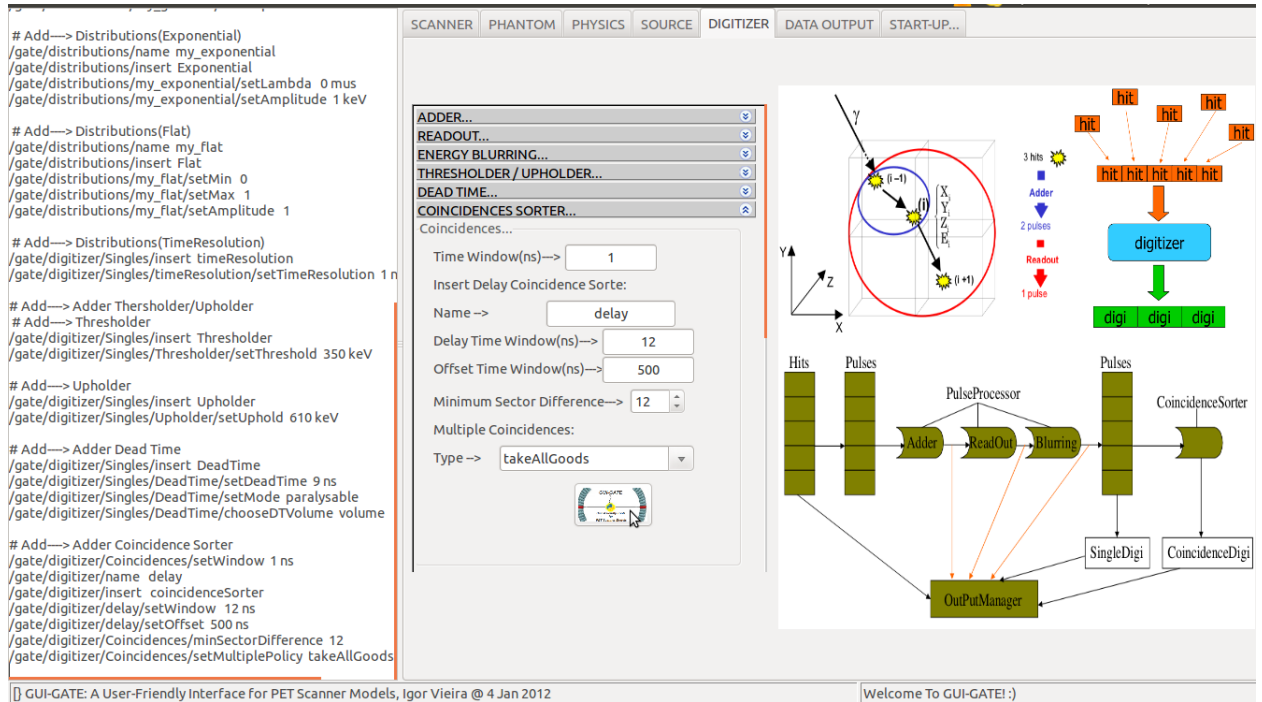
```
# Add—> Adder Dead Time
/gate/digitizer/Singles/insert DeadTime
/gate/digitizer/Singles/DeadTime/setDeadTime 9 ns
/gate/digitizer/Singles/DeadTime/setMode paralysable
/gate/digitizer/Singles/DeadTime/chooseDTVolume volume
```



Coincidences Sorter

Para definir os critérios de busca da ocorrência de eventos coincidências na matriz de eventos simples da simulação, os parâmetros estão mostrados na Figura 47.

Figura 47 – Submódulo "Coincidences Sorter" no GUIGATE.



Vale destacar aqui a possibilidade de simulação de múltiplos eventos coincidências chegando ao detectores, o que é esperado na prática. Esses tipos de múltiplos eventos são: (1) takeAllGoods, (2) takeWinnerOfGoods, (3) takeWinnerIfIsGood, (4) takeWinnerIfAllAreGoods, (5) keepIfOnlyOneGood, (6) keepIfAnyIsGood, (7) keepIfAllAreGoods, (8) killAllIfMultipleGoods, (9) killAll. A descrição de cada um pode ser visto na Tabela 7.

Tabela 7 – Diferentes modelos utilizados no GATE para simular múltiplas coincidências: descrição de cada um.

Modos/Casos	Descrição/Tipos
takeAllGoods	Bons pares são considerados
takeWinnerOfGoods	Apenas bons pares com energias maiores é considerado.
takeWinnerIfIsGood	Apenas bons pares com energias maiores é considerado, caso contrário, descarta-o.
takeWinnerIfAllAreGoods	Se todos os pares são bons, considera um com a maior energia.
keepIfOnlyOneGood	Se exatamente um par é bom, mantém todas as múltiplas coincidências.
keepIfAnyIsGood	Se, pelo menos, um par é bom, mantém todas as múltiplas coincidências.
keepAllIfMultipleGoods	Se mais de um par é bom, o evento é visto como um real "múltiplo" e então, todos os eventos são descartados.
killAll	Múltiplas coincidências não são aceitas, não importa quantos pares bons estejam presentes.

Adaptado: GATE User's Guide V6.1: Digitizer and Readout parameters.

4.1.6 Passo 6: Construindo a Fonte Emissora de Pósitrons

Vários tipos de fontes estão disponíveis no GATE, entre elas: voxelized, linacBeam, phaseSpace, PencilBeam, TPSPencilBeam e GPS.

No GUIGATE, está disponível nessa primeira versão, o modelo de fonte *GPS* (*General Particle Source*), que é usada para lançar partículas de um dado tipo, em uma dada direção, com determinada energia cinética.

Os tipos projetados no GUIGATE para o modelo GPS são (Figura 48):

1. "1- Positron Emitters..."
2. "2- Radioactive Isotope(Ion)..."
3. "3- Two Gamma Back to Back..."

É possível ainda simular os seguintes emissores de pósitrons: Flúor-18(F^{18}), Carbono-11(C^{11}) e Oxigênio-15(O^{15}) (Figura 49).

Figura 48 – Janela de Construção da Fonte: mostra diferentes opções projetadas para GUIGATE. Habilitada para esta versão apenas os emissores de pósitrons: Flúor-18 (F^{18}), Carbono-11 (C^{11}), Oxigênio-15 (O^{15}).

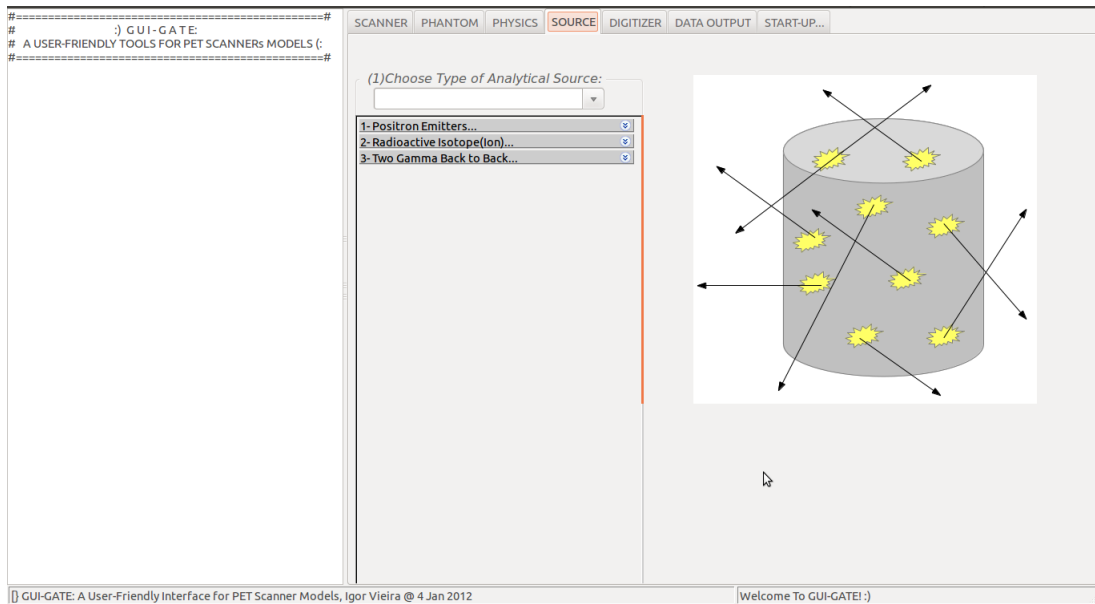
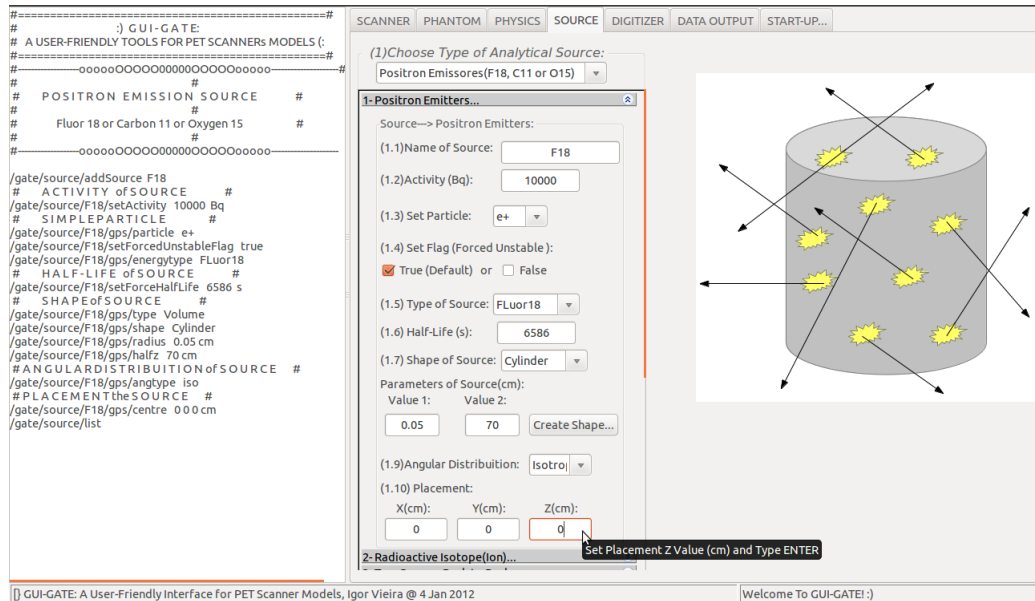


Figura 49 – Janela de Construção da Fonte emissora de pósitrons: F^{18} , C^{11} , O^{15} .



A Figura 50 em sua parte superior mostra o tipo de fonte que pode ser simulado nessa janela. O usuário precisa, obrigatoriamente, definir: o número atômico (Z), a massa atômica (A), a carga em unidades de energia (Q), e a energia de excitação (E) em keV. Este é o modo mais realista de simular radionuclídeos; contudo, é também o mais lento.

A Figura 51 mostra o input gerado no arquivo criado automaticamente ao serem definidos os parâmetros mostrados na Figura 49.

Figura 50 – Janelas de simulação de radionuclídeos no GUIGATE. Na parte superior a modelagem de íons. Na parte inferior, a modelagem de fótons gamma com ângulos de 180° graus.

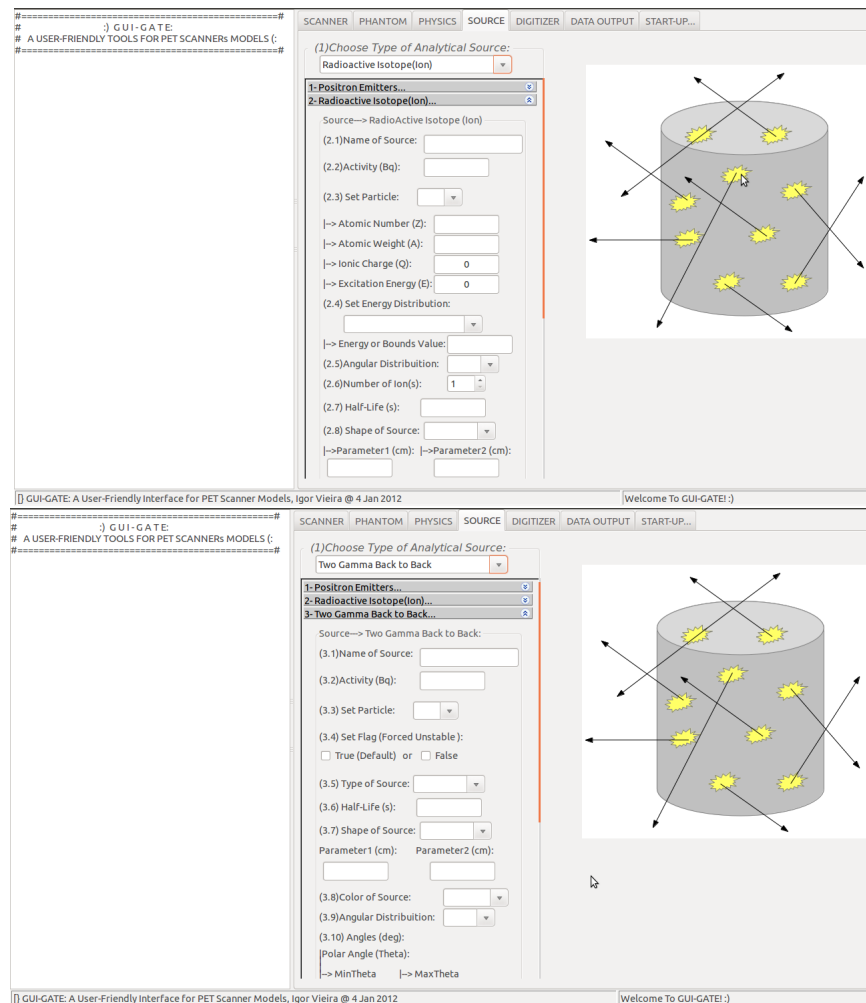


Figura 51 – Input de entrada no GATE gerado no GUIGATE, que simula uma fonte cilíndrica emissora de pósitrons reproduzindo o F^{18} .

```

20 #=====
21 # SOURCE #
22 #=====
23 # Add----> Source (GPS --> General Particle Source)
24 #=====
25 /gate/source/addSource F18
26 /gate/source/F18/setActivity 10000 Bq
27 /gate/source/F18/gps/particle e+
28 /gate/source/F18/setForcedUnstableFlag true
29 /gate/source/F18/gps/energytype Fluor18
30 /gate/source/F18/setForceHalfLife 6586 s
31 /gate/source/F18/gps/type Volume
32 /gate/source/F18/gps/shape Cylinder
33 /gate/source/F18/gps/radius 0.05 cm
34 /gate/source/F18/gps/halfz 70 cm
35 /gate/source/F18/gps/angtype iso
36 /gate/source/F18/gps/centre 0 0 0 cm
37 # Add----> List...
38 /gate/source/list

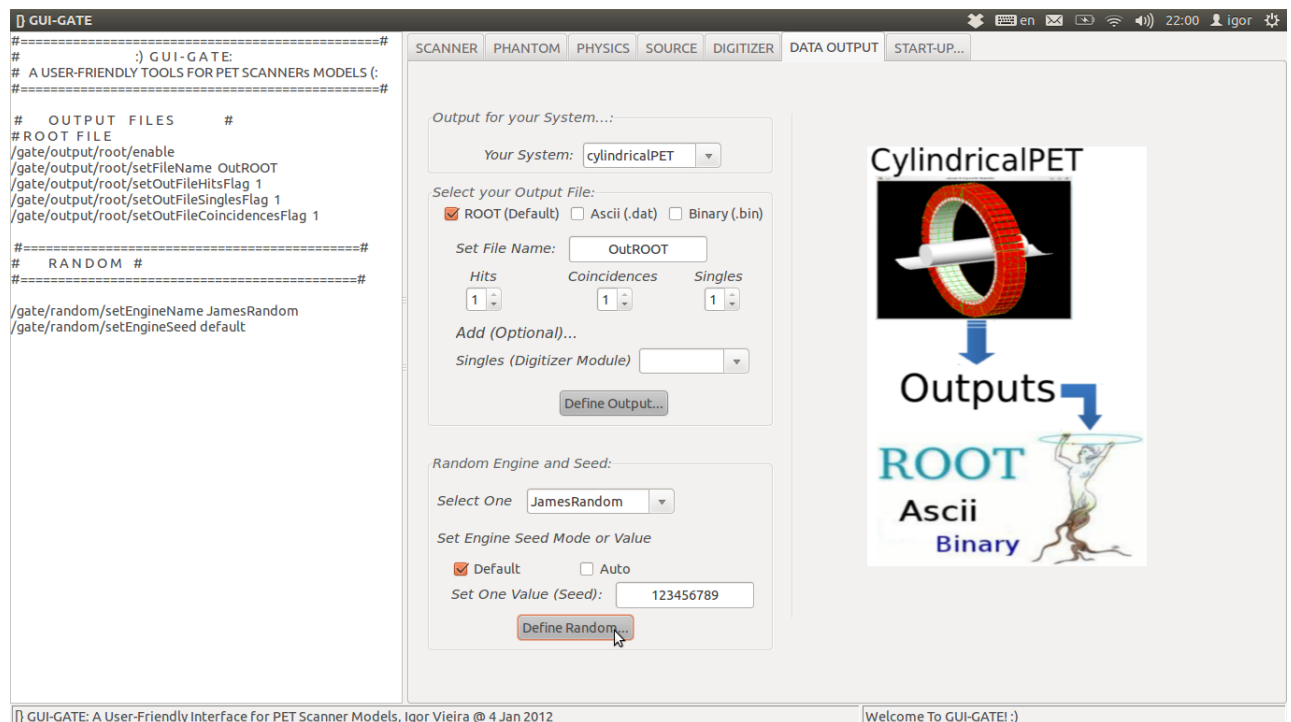
```

4.1.7 Passo 7: Definindo os Arquivos de Saídas

Esta etapa consiste em definir o formato do arquivo de saída ("**ROOT**", "**ASCII**" ou "**Binary**") em função do tipo de *scanner* a ser simulado (no caso um *CylindricalPET*) e o tipo de gerador de número aleatório ("**JamesRandom**", "**RanLux64**", "**MersenneTwister**").

Um exemplo desse processo, que segue o mesmo padrão no GUIGATE independentemente do tipo de arquivo definido, é mostrado na Figura 52. A diferença observada entre um e outro tipo de output definido se apresenta nos scripts em si, e não em como isto é feito no GUIGATE.

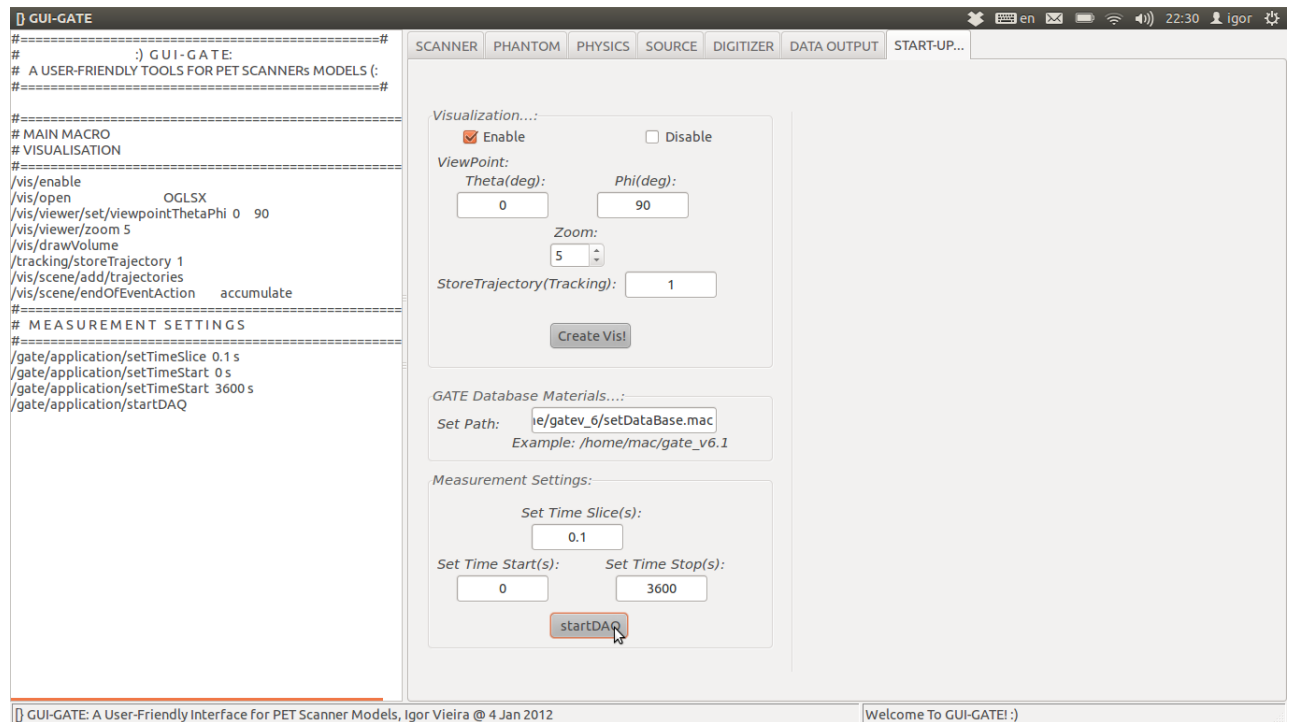
Figura 52 – Caminho para definição do tipo de arquivo de saída no GUIGATE.



4.1.8 Passo 8: Definindo o *Start-Up* do Experimento

Três etapas esperam ser definidas para inicialização do experimento: (i) os parâmetros de visualização da cena, (ii) o caminho do arquivo onde se encontra o banco de dados de materiais disponíveis no GATE e (iii) o tempo de cada "slice", assim como o tempo total da simulação (Figura 53).

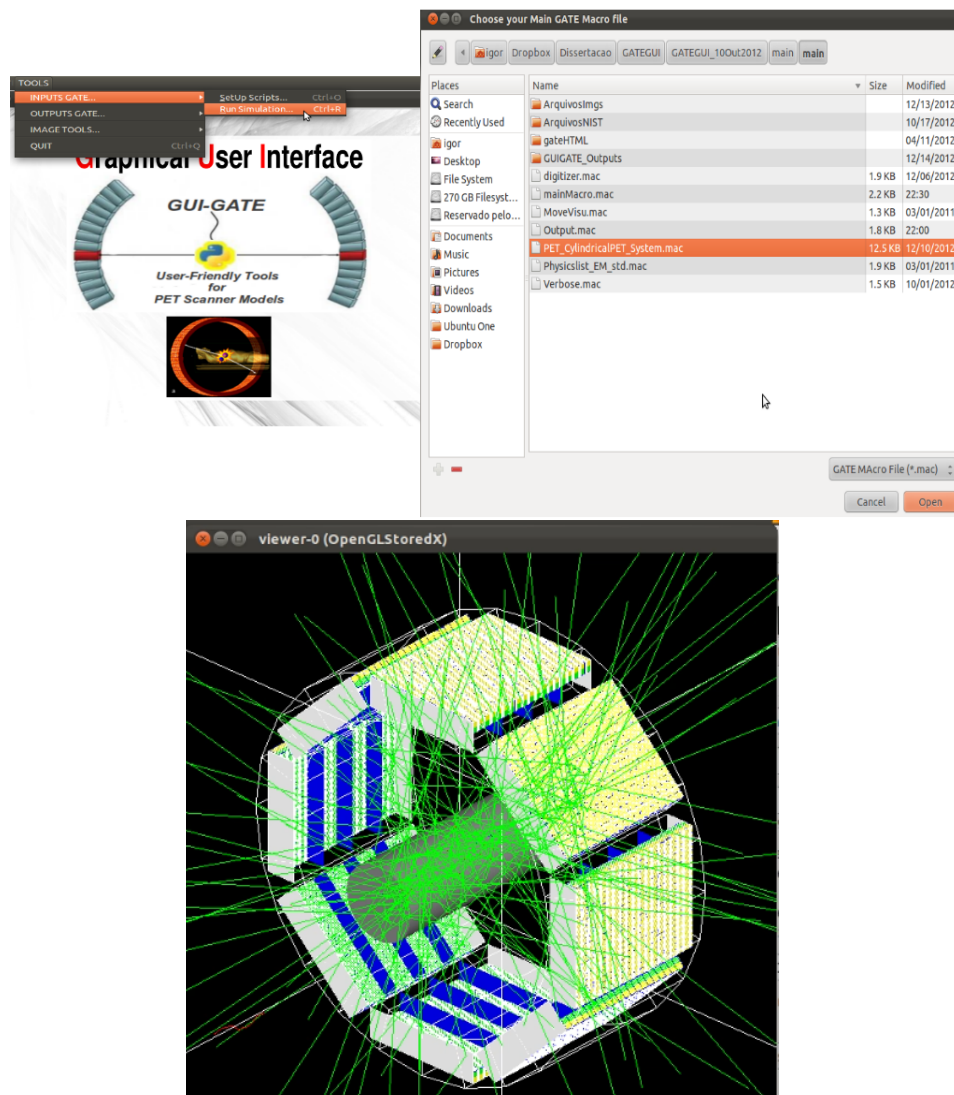
Figura 53 – Etapas finais de construção do experimento no GUIGATE.



4.1.9 Passo 9: Executando o GATE via GUIGATE

Ao serem definidos todos os arquivos de entrada nos passos anteriores, o usuário, ao retornar 'a janela principal ("**Main Window**")', pode executar seu modelo ao selecionar a macro principal. Após isso, caso tenha ativado o processo de visualização, uma janela mostra, usando OpenGL (biblioteca gráfica padrão do GATE), a simulação em tempo real: decaimento da fonte, movimento dos detectores etc.(Figura 54).

Figura 54 – Executando um experimento no GUIGATE: selecionando o arquivo na janela "Choose your Main GATE Macro File".

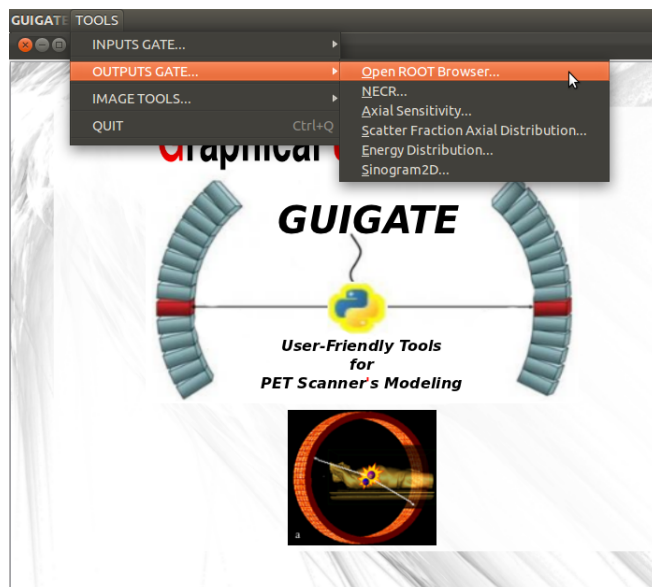


4.2 OUTPUTS NO GATE VIA GUIGATE

Entre os diferentes arquivos de saída disponíveis no GATE, apenas a análise automatizada de arquivos com extensão *.root* foi implementada para essa primeira versão do GUIGATE.

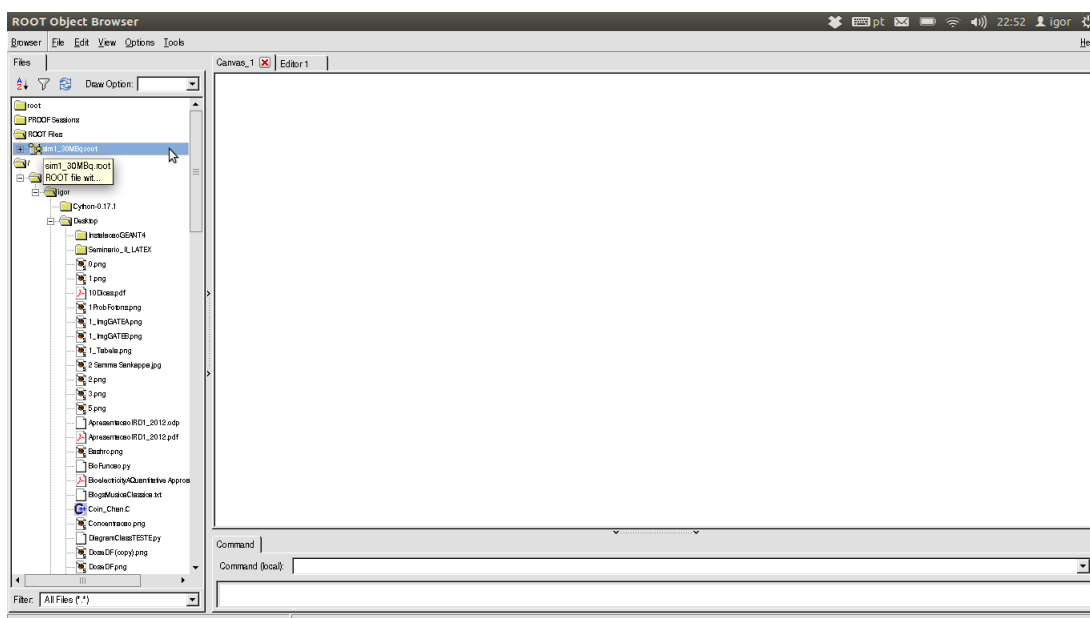
Assim sendo, uma vez criado o modelo e adquirido os arquivos de saída, através do caminho mostrado na Figura 55, o usuário pode apenas por meio de eventos de *mouse*, selecionando um dos *menus*, analisar seus dados com todos os recursos do ROOT, numa janela chamada "GUIGATE Output Module". Ao clicar em qualquer um dos *menus* que dão acesso a tal janela, abri-se a janela "Choose ROOT File", na qual o usuário seleciona o arquivo a ser analisado. Após isso, vários recursos de análise e edição de gráficos estão disponíveis, conforme mostrado a seguir.

Figura 55 – Opções de *menus* disponíveis para análises de arquivo ROOT.



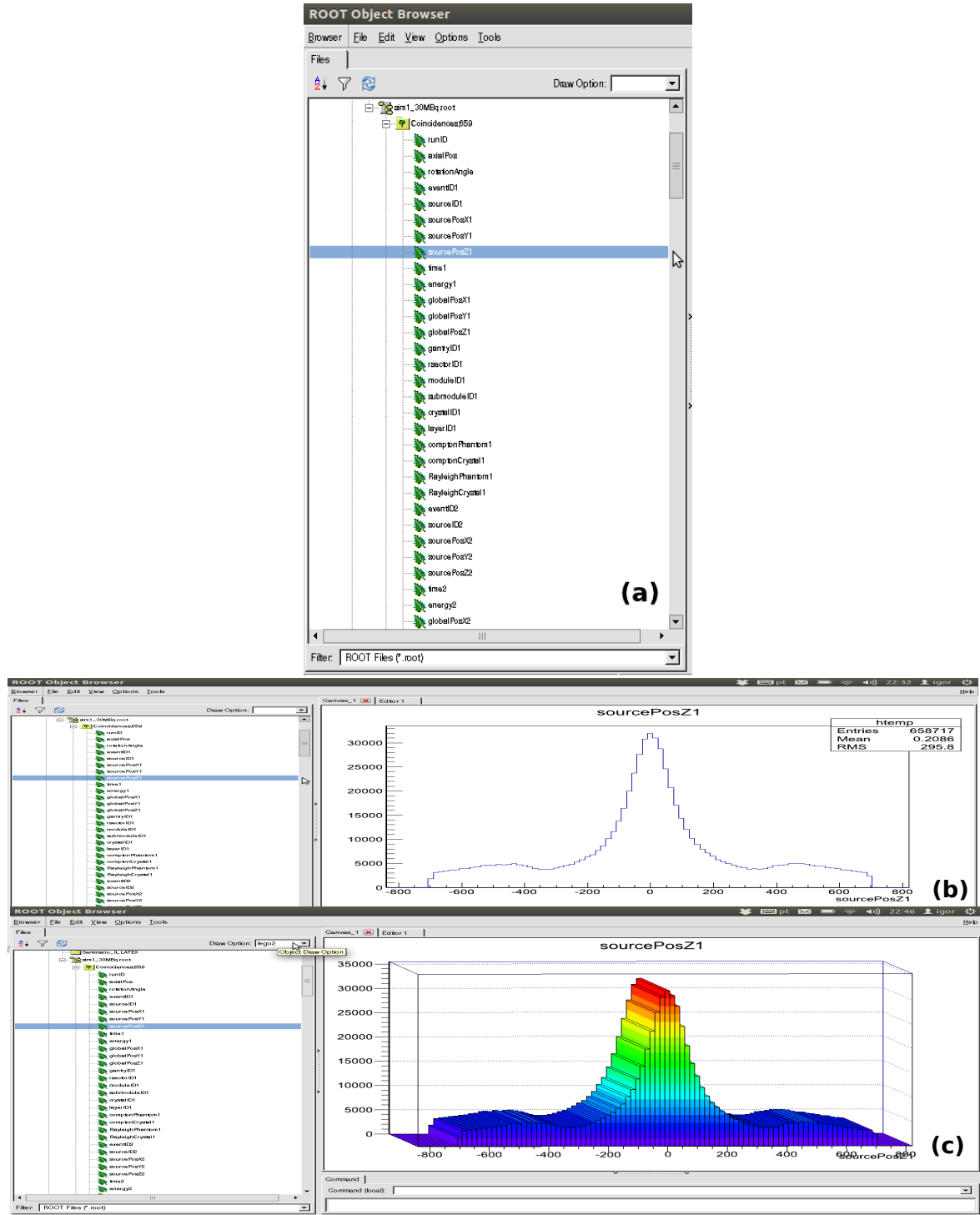
O acesso ao primeiro deles pode ser por meio do *menu* "**Open ROOT Browser...**". Como o próprio nome diz, ele permite abrir o *browser* do ROOT e assim navegar as árvores (*TTree*) existentes no mesmo. Ao clicar no *menu* em destaque na Figura 55 o *browser* do ROOT é exibido como mostra a Figura 56.

Figura 56 – *Browser* do ROOT que é exibida ao clicar no *menu* "**Open ROOT Browser...**". Cada *menu* deste *browser* tem uma série de funcionalidades nativas do ROOT.



Ao selecionar o arquivo com extensão *.root*, é possível navegar entre suas árvores e acessar alguns dados sob a forma de histogramas gerados automaticamente em tempo de execução, em diferentes modos de exibição entre aqueles disponíveis no ROOT, ou seja, normal (2D) ou *Lego2*(3D) como é chamado (Figura 57). Estes histogramas fornecem informações como por exemplo o número de contagens por cristal (representado por um número identificador ou ID), o espectro de energia para os pares coincidência, entre outros.

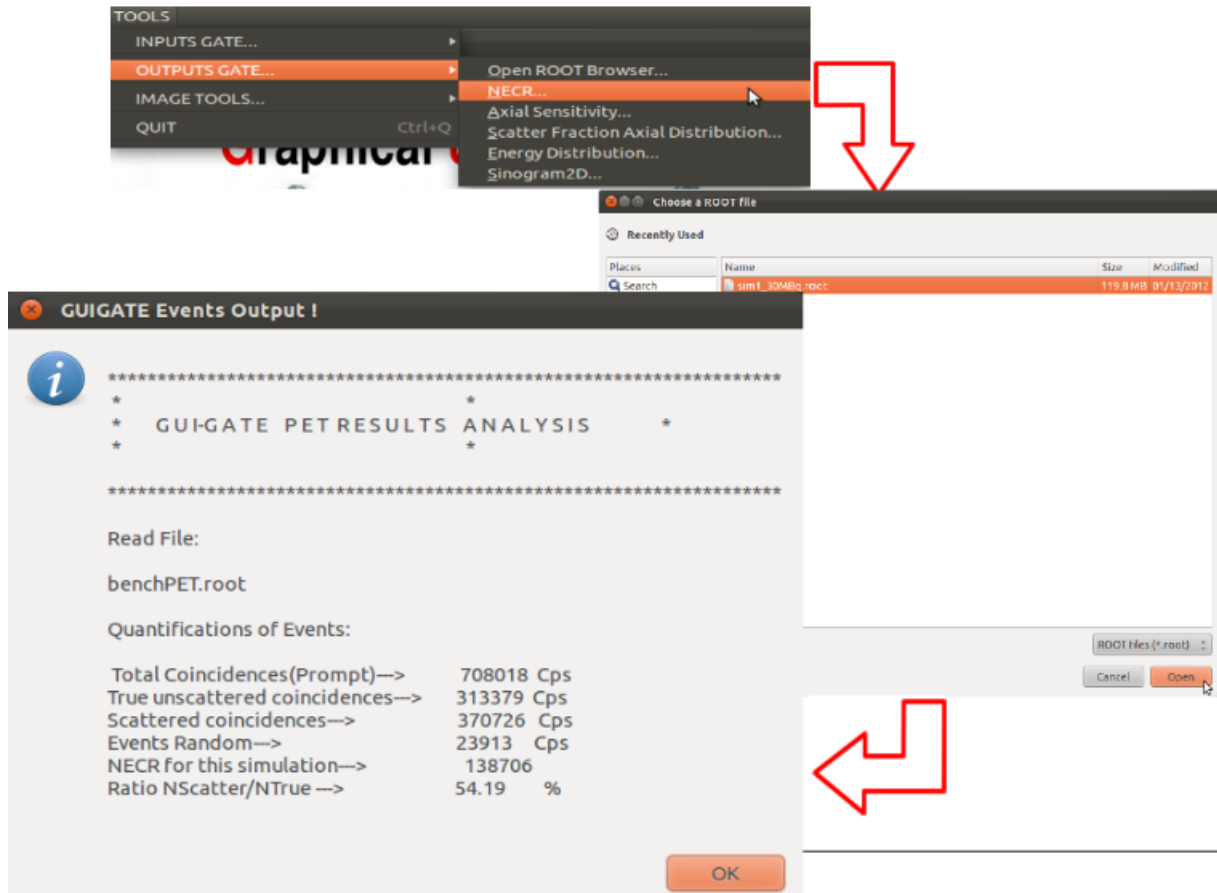
Figura 57 – (a) Árvore gerada na simulação do GATE (Superior). (b) Distribuição de contagens da fonte ao longo do eixo-Z no modo padrão. (c) Mesma imagem que a anterior, exibida em um dos modos nativos de exibição de histograma no ROOT, o "Lego2"(3D)



O segundo recurso disponível é a análise do *NECR*. Embora o *menu* tenha este nome, a estimativa do *NECR* não é o único recurso disponível como mostra a Figura 58. Nessa mesma janela é possível notar outros parâmetros importantes em PET. Entre eles se encontram: o número total de eventos coincidências, o número de eventos verdadeiros, eventos espalhados,

eventos aleatórios, assim como a fração de espalhamento e o próprio *NECR*.

Figura 58 – Janela geradora do *NECR* e outros parâmetros importantes em PET.



A Figura 59, por sua vez, mostra um histograma 2D exibindo a sensibilidade axial do *scanner* ao utilizar-se o *menu* exibido nesta mesma figura. No *menu* "Tools" do *browser* da janela "GUIGATE Output Module", é possível acessar o *Fit Panel* e, utilizando os recursos disponíveis nessa janela, encontrar o melhor modelo ou tendência que representa os dados plotados no histograma.

A opção seguinte permite obter a fração de espalhamento axial, através do seguinte caminho na janela principal do GUIGATE (Figura 55): "**TOOLS**" → "**Outputs GATE...**" → "**Scatter Fraction Axial Distribution...**" → "**Choose ROOT File**" → "**Ok**". A janela resultante é exibida na Figura 60.

Figura 59 – Janela exibe histograma 2D da sensibilidade axial do *scanner*. Na "Tab" ao lado do gráfico é possível ver alguns recursos de edição de gráfico.

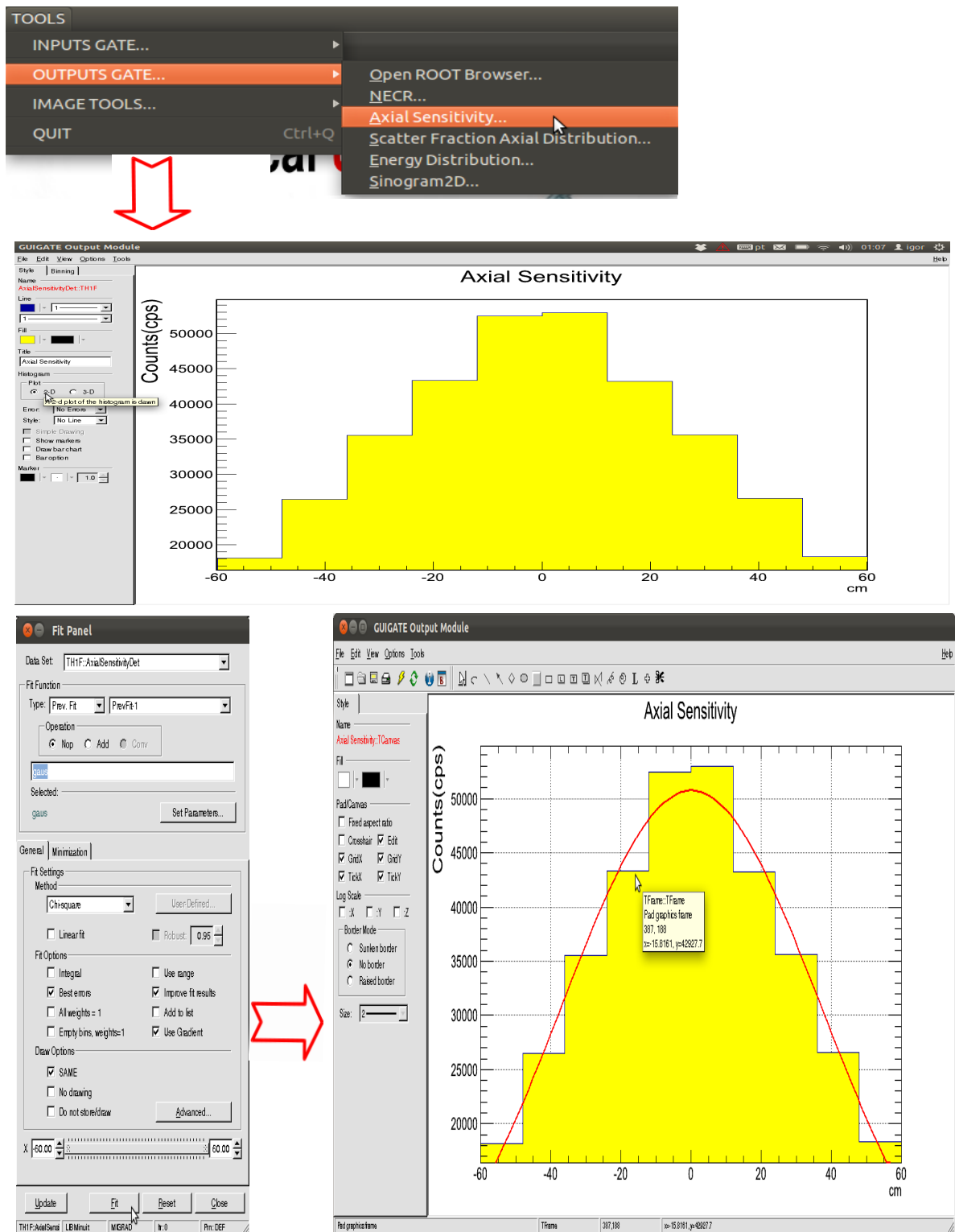
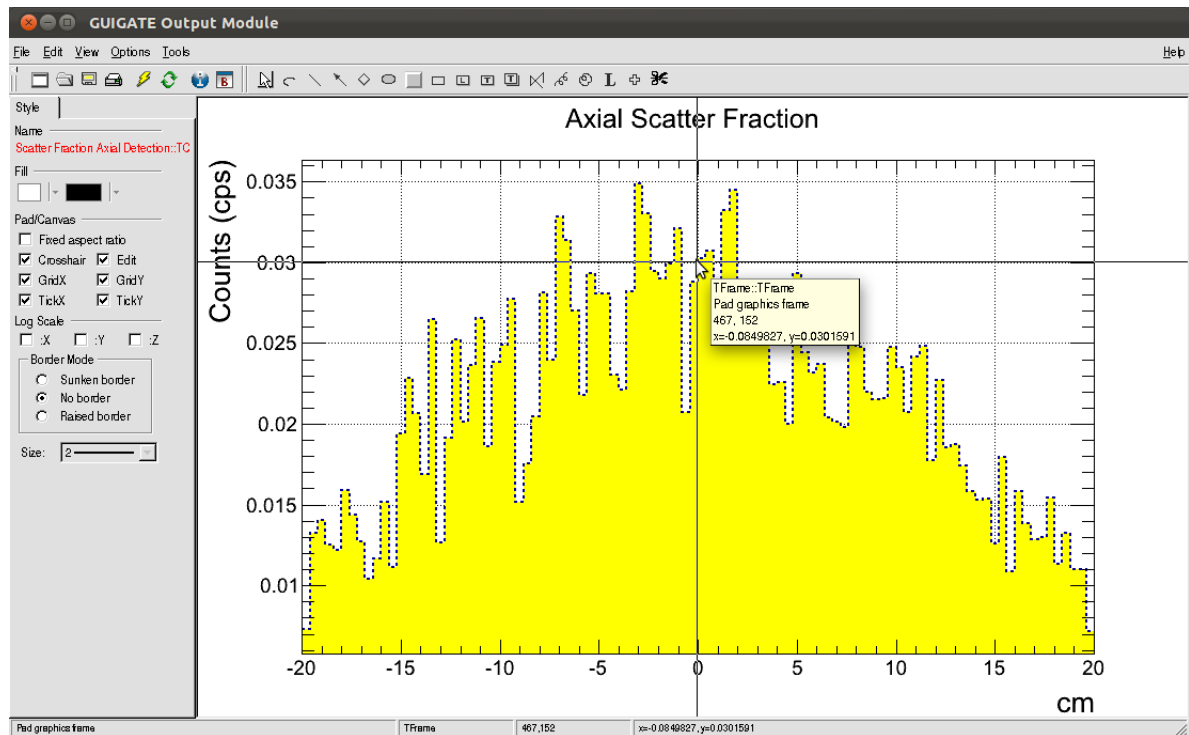
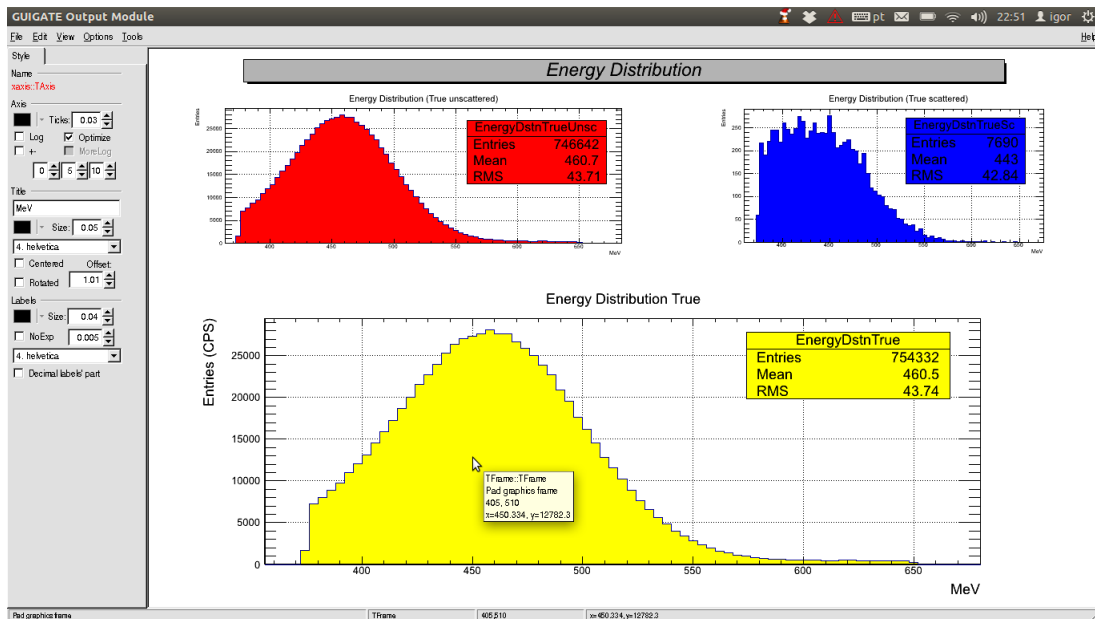


Figura 60 – Janela exibe histograma 2D da fração de espalhamento axial do *scanner*.



Uma outra importante ferramenta disponível é a geração da distribuição energética dos eventos únicos (*single*) contabilizados como evento coincidência: verdadeiro (em vermelho), espalhado (em azul) e total (em amarelo). Espera-se que as contagens se realizem considerando a janela energética definida anteriormente na simulação (no caso aqui, de 350 a 650 keV). Pode-se notar nos *Stats Box* (caixa com informações estatísticas) das curvas vermelha e amarela que a diferença entre uma e outra foi insignificante para a simulação em análise, onde a fração de espalhamento foi algo menor que 1%, o que se faz refletir nos valores encontrados para a curva azul (Figura 61). A "Tab", de nome "Style", ao lado dos gráficos, exibe a possibilidade de uma outra ferramenta de edição, neste caso, dos eixos dos gráficos.

Figura 61 – Janela exibe histograma 2D da distribuição energética dos diferentes tipos de eventos coincidências verdadeira (vermelha), espalhada (azul) e total(amarela).



Por fim, o menu "*Sinogram2D*" fornece não somente o sinograma 2D, mas também a distribuição espacial (3D) dos eventos registrados nos detectores. Isto permite avaliar a tendência de detecção dos eventos coincidências, sobretudo dos eventos espalhados (pontos brancos) nos detectores (Figura 62) ao se considerar na simulação a modelagem de estruturas como a mesa do equipamento, por exemplo.

Figura 62 – Janela exibe sinograma 2D e distribuição espacial (3D) dos diferentes eventos registrados na matriz de detectores. Essa janela permite interações com rotação etc.

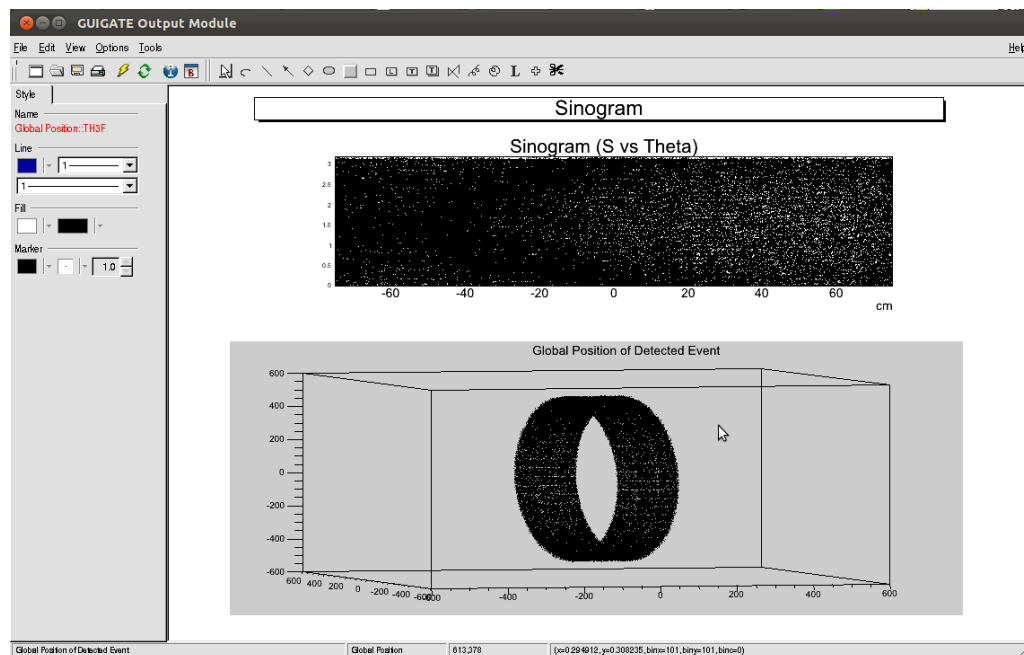


Tabela 8 – Comparação dos resultados médios obtidos modelando o *benchPET*, usando método padrão no GATE e o GUIGATE.

Grandezas Comparadas	GATE	GUIGATE
Decaimentos Esperados	$3,6813 \cdot 10^7 \pm 0.01\%$	$3,6813 \cdot 10^7 \pm 0.01\%$
Coincidências Aleatórias	$2,3536 \cdot 10^4 \pm 0.20\%$	$2,3913 \cdot 10^4 \pm 0.65\%$
Coincidências Espalhadas	$3,70116 \cdot 10^5 \pm 0.12\%$	$3,70726 \cdot 10^5 \pm 0.16\%$
Coincidências Verdadeiras	$3,12725 \cdot 10^5 \pm 0.10\%$	$3,13379 \cdot 10^5 \pm 0.18\%$

A Tabela 8 compara os resultados gerais obtidos na GUIGATE, com os valores publicados na *OpenGATECollaboration* para o chamado *benchPET*. Estes valores se encontram disponíveis na distribuição do GATE para testes gerais (USER'SGUIDE, 2012). Observou-se durante a aquisição de 4 minutos, dividida em 2 *frames*, aproximadamente $3,7 \cdot 10^7$ decaimentos e algo em torno de $7,08018 \cdot 10^5$ eventos coincidências registradas no arquivo de saída. A diferença média para os eventos coincidência foi de 1,14%, quando comparado ao valor de referência da distribuição. O tempo total da simulação foi algo em torno de 5 horas.

Em todos os casos, o erro associado foi menor que 1,00%, como pode ser visto na Tabela 8, e a precisão alcançada com a GUIGATE neste experimento, se deve ao fato de todo o processo de geração dos arquivos de entrada, execução e análise dos arquivos de saída estarem conforme o padrão exigido pelo GATE.

5 CONCLUSÕES E PERSPECTIVAS

Há uma tendência cada vez mais crescente na comunidade científica, ou mesmo dentro das grandes empresas da área médica, de utilizar códigos de transporte das radiações para validar resultados experimentais ou mesmo para projetar novos experimentos e/ou equipamentos. Neste trabalho, um método para modelagem de tomógrafo por emissão de pósitrons utilizando o GATE foi proposto e inicialmente validado.

O GUIGATE, em sua primeira versão, está apto a modelar PET scanner integrando as funcionalidades do GATE, no que diz respeito a utilização do *framework cylindricalPET*, o qual permite modelar os tomógrafos existentes no mercado, especialmente os dedicados a pequenos animais. Estes em geral são inicialmente projetados e/ou utilizados para pesquisas de desenvolvimento de novos radiofármacos para PET, bem como para desenvolvimento de novos tomógrafos, dado seu menor tamanho, o que significa um menor número de cristais detectores e, consequentemente, menor custo de fabricação.

Além disso, o GUIGATE congrega em um único ambiente e de modo amigável, o vasto analisador de dados do CERN, o ROOT, viabilizando assim uma série de análises de parâmetros fundamentais em controle de qualidade de PET *scanners*, o que aponta a possibilidade de uso do GUIGATE neste sentido.

Portanto, a comunidade internacional de usuários do GATE (a *OpenGATECollaboration*) e interessados em modelagem PET, agora dispõem de um método mais prático e rápido de projetar e analisar seus experimentos computacionais.

Para a próxima versão do GUIGATE, contudo, vem sendo implementado recursos para do mesmo modo, em um único ambiente, ser possível visualizar e acoplar fantasmas de voxel nos formatos lido pelo GATE (*interfile*, *analyze* e *DICOM*), afim de que esse processo também possa ser feito de modo amigável, prático e intuitivo, abrangendo assim a possibilidade de aplicação do GUIGATE tanto em instrumentação em PET *scanners*, quanto em dosimetria numérica.

REFERÊNCIAS

- ALAMOS, L. *MCNP — A General Monte Carlo N-Particle Transport Code*. <http://mcnp.lanl.gov/>, April 2003.
- AWARE, A. *wxPython 2.8 Application Development Cookbook*. 1. ed. [S.l.]: Packt Publishing, 2010.
- BADAWI, D. M. Nec: Some coincidences are more equivalent than others. *J.Nucl. Med.*, v. 46, p. 1767–1768, 2005.
- BAILEY, D.; TOWNSEND, D. W.; VALK, P. E.; MAISEY, M. N. *Positron Emission Tomography Basic Sciences*. 1. ed. [S.l.]: Springer, 2003.
- BARRET; CARPENTER, T. A.; CLARK, J. C.; ANSORGER, E.; FRYER, T. D. Monte carlo simulation and scatter correction of the ge advance pet scanner with simset and geant4. *Phys. Med. Biol.*, v. 50, p. 4823–4840, 2005.
- BRUN, R.; RADEMAKERS, F. Root- an object oriented data analysis framework. *Nucl. Instrum. Methods Phys.*, Res.A 389, p. 81–86, 1997.
- BUVAT, I.; CASTIGLIONI, I. Monte carlo simulations in spect and pet. *J.Nucl. Med.*, v. 46, p. 48–61, 2002.
- CERN. *Geant4 Physics Reference*. Switzerland, 2010. Versão: Geant4 9.4.
- CERN. *Geant4 User's Guide-For Application Developers*. Switzerland, 2010. Versão: Geant4 9.4. Disponível em: <<http://cern.ch/geant4>>. Acesso em: Marco 2012.
- DERENZO, S. Method for optimizing side shielding in positron emission tomographs and for comparing detector materials. *J.Nucl. Med.*, v. 21(10), p. 971–7, 1980.
- DERENZO, S.; ZAKLAD, H. .; BUDINGER, T. Analytical study of a high-resolution positron ring detector system for transaxial reconstruction tomography. *J.Nucl. Med.*, v. 16(12), p. 1166–73, 1975.
- FAHEY, F. H. Data acquisition in pet imaging. *J. of Nucl. Med. Tech.*, v. 30(2), p. 239–49, 2002.
- FAKHRI, G.; SANTOS, P.; BADAWI, R.; HOLDSWORTH, C.; A.D., K. M. V. D. A. Impact of acquisition geometry, image processing, and patient habitus on tumor detection in whole-body fdg-pet. *J.Nucl. Med.*, v. 48, p. 1951–1960, 2007.
- IEC. *International Standard IEC 61675-1: Radionuclide Imaging Devices – Characteristics and Test Conditions. Part 1: Positron Emission Tomographs*. [S.l.], 1998.
- JOHNS, H. E.; CUNNINGHAM, J. R. *The Physics of Radiology*. 2. ed. [S.l.]: Springfield: Charles C Thomas, 1983.

KAWRAKOW, I.; E., M.-H.; D.W.O., R.; F., T.; B.R.B., W. *The main EGSnrc manual*. <http://irs.inms.nrc.ca/software/egsnrc/documentation/pirs701/index.html>, May 9 2011.

KRAMER, R.; VIEIRA, J. W.; KHOURY, H. J.; LIMA, F. R. A.; FUELLE, D. All about max: a male adult voxel phantom for monte carlo calculations in radiation protection dosimetry. *Physics in Medicine and Biology*, v. 48, p. 1239–62, 2003.

KRAMER, R.; VIEIRA, J. W.; KHOURY, H. J.; LIMA, F. R. A.; LIMA, V. J. M.; LOUREIRO, E. C. M.; HOFF, G. All about fax: a female adult voxel phantom for monte carlo calculations in radiation protection dosimetry. *Physics in Medicine and Biology*, v. 49, p. 5203–16, 2004.

KRAMER, R.; VIEIRA, J. W.; KHOURY, H. J.; LIMA, V. J. M. Max 06 and fax 06: update of two adult human phantoms for radiation protection dosimetry. *Physics in Medicine and Biology*, v. 51, p. 3331–3346, 2006.

LARTIZIEN, C.; COMTAT, C.; KINAHAN, N.; FERREIRA, B.; BENDRIEM, R.; TRE-BOSSSEN. Optimization of injected dose based on noise equivalent count rates for 2- and 3-dimensional whole-body pet. *J Nucl Med.*, v. 43, p. 1268 –1278, 2002.

LAZARO; BUVAT, I.; LOUDOS, G.; STRUL, D.; SANTIN, G.; GIOKARIS, N.; DONNARIEIX, D.; MAIGNE, L.; SPANOUDAKI, V.; STYLIARIS, S.; STAELENS, S.; BRETON, V. Validation of the gate monte carlo simulation platform for modeling a csi(tl) scintillation camera dedicated to small-animal imaging. *Phys. Med. Biol.*, v. 49, p. 271–285, 2004a.

LEI, T. *Statistics Of Medical Imaging*. 1. ed. [S.l.]: Chapman and Hall CRC Interdisciplinary Statistics, 2012.

LIMA, V. J. de M.; CASSOLA, V. F.; KRAMER, R.; LIRA, C. A. B. de O.; KHOURY, H. J.; VIEIRA, J. W. Development of 5- and 10-year-old pediatric phantoms based on polygon mesh surfaces. *Med. Phys*, v. 38(8), p. 4723–36, 2011.

LIMA, V. J. Kramer R.; Cassola V. F.; Khoury H. J.; Vieira J. W.; de M.; BROWN, K. . R. Fash and mash: female and male adult human phantoms based on polygon mesh surfaces: Ii. dosimetric calculations. *Physics in Medicine and Biology*, v. 55, p. 163–189, 2010.

MACKINNEY, W. W. *Python For Data Analysis*. 1. ed. [S.l.]: O'Reilly Media, 2012.

MELCHER, C. L. Scintillation crystal for pet. *J. Nucl. Med.*, v. 41(6), p. 1051–1055, 2000.

MOHAMMAD, D.; XIAOYI, J.; KLAUS, S. *Correction Techniques in Emission Tomography*. 1. ed. [S.l.]: CRC Press, 2012.

NEMA. *NEMA NU-2 Standards Publication NU2-2007: Performance Measurements of Positron Emission Tomographs*. [S.l.], 2007.

NEMA. *NEMA NU-4 Standards Publication NU4-2008: Performance Measurements of Small Animal Positron Emission Tomographs*. [S.l.], 2008.

NICOLAS, A. K.; ANASTASIOS, P.; GEORGE, L.; N., K. S. A simulation study of the counting-rate performance of clinical pet systems applying a methodology for optimizing the injected dose. In: *IEEE International workshop on Imaging Systems and Techniques*. [S.l.: s.n.], 2008.

OKUNO, E.; YOSHIMURA. *Física das Radiações*. 1. ed. [S.l.]: S. Paulo: Oficina de Textos, 2010.

PEDROSA, J. L. *Nuclear Medicine Physics*. 1. ed. [S.l.]: CRC Press, 2011.

PHELPS, M. E.; SIMON, N. *PET: Physics, Instrumentation and Scanner*. 1. ed. [S.l.]: Springer, 2006.

SILVA, S. *Small Animal PET Imaging Using GATE Monte Carlo Simulations: implementation of physiological and metabolic information*. Tese (Doutorado) — Departamento de Física, Universidade de Lisboa, 2010.

SÁ, L. V. *Controle de Qualidade de Imagem em Tomografia Por Emissão de Pósitrons*. Tese (Doutorado) — COOPE/UFRJ, 2010.

TURNER, J. E. *Atoms, Radiation, and Radiation Protection*. 2. ed. [S.l.]: WILEY-VCH, 2007.

USER'S GUIDE, G. *OpenGATE Collaboration User's Guide V6.1*. 2. ed. http://wiki.opengatecollaboration.org/index.php/Users_Guide_V6.1, Acessado em Agosto 2012.

VIEIRA, I.; GOMES, M.; VIEIRA, J.; LIMA, F. Códigos dedicados a modelagem monte carlo de pet e/ou spect: uma visão geral par iniciantes. In: *III Congresso de Proteção Contra Radiações de Países e Comunidades de Língua Portuguesa*. [S.l.: s.n.], 2012.

WATSON, C.; CASEY, M.; BENDRIEM, B.; JP, C.; DW, T.; S, E.; S, M.; FP, D. Optimizing injected dose in clinical pet by accurately modeling the counting-rate response functions specific to individual patient scans. *J. Nucl. Med.*, v. 46, p. 1825–1834, 2005.

YOSHIMURA. Física das radiações: Interação da radiação com a matéria. *Revista Brasileira de Física Médica*, v. 3(1), p. 57–67, 2009.

ZAID, H. *Therapeutic Application of Monte Carlo Calculation in Nuclear Medicine*. 1. ed. [S.l.]: Institute of Physics, Series in Medical Physics and Biomedical Engineering, 2003.

ZAID, H. *Quantitative Analysis in Nuclear Medicine Imaging*. 1. ed. [S.l.]: New York, Springer, 2006.

6 APÊNDICES

6.1 APÊNDICE A: SCRIPTS DE UM PET SCANNER CLÍNICO

```
#-----oooooooooooooooooooooooo-----#
# ESTE SCIPT MOSTRA UM PET GE ADVANCE EM MODO 3D
#
# ( I M A G I N G A P P L I C A T I O N S )
#
#-----oooooooooooooooooooooooo-----#
#=====
# VISUALISATION
#=====
/vis/disable
/vis/open OGLSX
/vis/viewer/set/viewpointThetaPhi 40 30
/vis/viewer/zoom 3.0
/vis/drawVolume
/tracking/storeTrajectory 1
/vis/scene/add/trajectories
/vis/scene/endOfEventAction accumulate

#=====
# GEOMETRIA
#=====
#Especificar caminho para onde está o "database"...
/gate/geometry/setMaterialDatabase GateMaterials.db
# W O R L D /gate/world/geometry/setXLength 200. cm
/gate/world/geometry/setYLength 200. cm
/gate/world/geometry/setZLength 200. cm
#-----oooooooooooooooooooooooo-----#
# P E T S C A N N E R M O D E L #
#-----oooooooooooooooooooooooo-----#
#CYLINDRICALPET
/gate/world/daughters/name cylindricalPET
/gate/world/daughters/insert cylinder
/gate/cylindricalPET/placement/setTranslation 0.0 0.0 0.0 cm
/gate/cylindricalPET/geometry/setRmax 61.0 cm
/gate/cylindricalPET/geometry/setRmin 47.0 cm
```

```

/gate/cylindricalPET/geometry/setHeight 15.6 cm
/gate/cylindricalPET/setMaterial Air
/gate/cylindricalPET/vis/setColor yellow
/gate/cylindricalPET/vis/forceWireframe
# R SECTOR
/gate/cylindricalPET/daughters/name rsector
/gate/cylindricalPET/daughters/insert box
/gate/rsector/placement/setTranslation 53.5 0.0 0.0 cm
/gate/rsector/geometry/setXLength 13.65 cm
/gate/rsector/geometry/setYLength 4.92 cm
/gate/rsector/geometry/setZLength 15.9 cm
/gate/rsector/setMaterial Air
/gate/rsector/vis/forceWireframe
/gate/rsector/vis/setColor red
# M O D U L E
/gate/rsector/daughters/name module
/gate/rsector/daughters/insert box
/gate/module/placement/setTranslation 0.0 0.0 0.0 cm
/gate/module/geometry/setXLength 13.65 cm
/gate/module/geometry/setYLength 4.92 cm
/gate/module/geometry/setZLength 15.9 cm
/gate/module/setMaterial Air
/gate/module/vis/forceWireframe
/gate/module/vis/setColor red
#BLOCK
/gate/module/daughters/name Block
/gate/module/daughters/insert box
/gate/Block/placement/setTranslation 0.0 0.0 0.0 cm
/gate/Block/geometry/setXLength 13.65 cm
/gate/Block/geometry/setYLength 2.48 cm
/gate/Block/geometry/setZLength 5.0 cm
/gate/Block/setMaterial Plastic
/gate/Block/vis/forceWireframe
/gate/Block/vis/setColor green
# P H O T O M U T I P L I C A D O R A
/gate/Block/daughters/name photomultiplicadora
/gate/Block/daughters/insert box
/gate/photomultiplicadora/placement/setTranslation 1.6 0.0 0.0 cm
/gate/photomultiplicadora/geometry/setXLength 10.7 cm
/gate/photomultiplicadora/geometry/setYLength 2.48 cm
/gate/photomultiplicadora/geometry/setZLength 2.4 cm
/gate/photomultiplicadora/setMaterial Glass
/gate/photomultiplicadora/vis/forceSolid
/gate/photomultiplicadora/vis/setColor red
#C R Y S T A L
/gate/Block/daughters/name crystal
/gate/Block/daughters/insert box
/gate/crystal/placement/setTranslation -5.32 0.0 0.0 cm

```

```

/gate/crystal/geometry/setXLength 3.0 cm
/gate/crystal/geometry/setYLength 0.40 cm
/gate/crystal/geometry/setZLength 0.84 cm
/gate/crystal/setMaterial BGO
/gate/crystal/vis/setColor white
/gate/crystal/vis/forceSolid
# L A Y E R BGO
/gate/crystal/daughters/name BGO
/gate/crystal/daughters/insert box
/gate/BGO/placement/setTranslation 0.0 0.0 0.0 cm
/gate/BGO/geometry/setXLength 3.0 cm
/gate/BGO/geometry/setYLength 0.40 cm
/gate/BGO/geometry/setZLength 0.84 cm
/gate/BGO/setMaterial BGO
/gate/BGO/vis/setColor white
/gate/BGO/vis/forceSolid
# Mesa de Exame
/gate/cylindricalPET/daughters/name mesa
/gate/cylindricalPET/daughters/insert cylinder
/gate/mesa/placement/setTranslation 0.0 20.0 0.0 cm
/gate/mesa/geometry/setRmax 30.6 cm
/gate/mesa/geometry/setRmin 30.0 cm
/gate/mesa/geometry/setHeight 100.0 cm
/gate/mesa/geometry/setPhiStart 240 deg
/gate/mesa/geometry/setDeltaPhi 60 deg
/gate/mesa/setMaterial Polyethylene
/gate/mes2/vis/setColor white
/gate/mesa/vis/forceSolid
# R E P E A T MODULE /gate/rsector/repeaters/insert ring
/gate/rsector/ring/setRepeatNumber 56
# R E P E A T B L O C K
/gate/Block/repeaters/insert cubicArray
/gate/Block/cubicArray/setRepeatNumberX 1
/gate/Block/cubicArray/setRepeatNumberY 2
/gate/Block/cubicArray/setRepeatNumberZ 3
/gate/Block/cubicArray/setRepeatVector 13.0 2.5 5.3 cm
# R E P E A T P H O T O M U L T I P L I E R
/gate/photomultiplicadora/repeaters/insert cubicArray
/gate/photomultiplicadora/cubicArray/setRepeatNumberX 1
/gate/photomultiplicadora/cubicArray/setRepeatNumberY 1
/gate/photomultiplicadora/cubicArray/setRepeatNumberZ 2
/gate/photomultiplicadora/cubicArray/setRepeatVector 13.0 0.0 2.5 cm
# R E P E A T C R Y S T A L
/gate/crystal/repeaters/insert cubicArray
/gate/crystal/cubicArray/setRepeatNumberX 1
/gate/crystal/cubicArray/setRepeatNumberY 6
/gate/crystal/cubicArray/setRepeatNumberZ 6
/gate/crystal/cubicArray/setRepeatVector 3.0 0.41 0.843 cm

```

```

# A T T A C H S Y S T E M
/gate/systems/cylindricalPET/rsector/attach rsector
/gate/systems/cylindricalPET/module/attach module
/gate/systems/cylindricalPET/crystal/attach crystal
/gate/systems/cylindricalPET/layer0/attach BGO
# A T T A C H C R Y S T A L S D
/gate/BGO/attachCrystalSD
/gate/systems/cylindricalPET/describe
# M O V I M E N T O D O S C A N N E R
/gate/cylindricalPET/moves/insert orbiting
/gate/cylindricalPET/orbiting/setSpeed 30 deg/s
/gate/cylindricalPET/orbiting/setPoint1 0 0 0 cm
/gate/cylindricalPET/orbiting/setPoint2 0 0 4.0 cm
#=====
# P H A N T O M
#=====
# Phantom for NEMA scatter
/gate/world/daughters/name NEMAphantom
/gate/world/daughters/insert cylinder
/gate/NEMAphantom/placement/setTranslation 0.0 0.0 0.0 cm
/gate/NEMAphantom/setMaterial Polyethylene
/gate/NEMAphantom/geometry/setRmin 9.8 cm
/gate/NEMAphantom/geometry/setRmax 10.1 cm
/gate/NEMAphantom/geometry/setHeight 70.1 cm
/gate/NEMAphantom/vis/forceWireframe
/gate/NEMAphantom/vis/setColor white
/gate/NEMAphantom/daughters/name phantomWater
/gate/NEMAphantom/daughters/insert cylinder
/gate/phantomWater/setMaterial Water
/gate/phantomWater/geometry/setRmax 10.0 cm
/gate/phantomWater/geometry/setHeight 70.0 cm
/gate/phantomWater/vis/forceSolid
/gate/phantomWater/vis/setColor gray
/gate/NEMAphantom/attachPhantomSD
#=====
# PHYSICS
#=====
/gate/physics/addProcess PhotoElectric
/gate/physics/addProcess Compton
/gate/physics/addProcess GammaConversion
/gate/physics/addProcess LowEnergyRayleighScattering
/gate/physics/addProcess ElectronIonisation
/gate/physics/addProcess Bremsstrahlung
/gate/physics/addProcess PositronAnnihilationStd
/gate/physics/addProcess MultipleScattering e+
/gate/physics/addProcess MultipleScattering e-
/gate/physics/processList Enabled
/gate/physics/processList Initialized

```

```

#=====
# CUTS
#=====
/gate/physics/Gamma/SetCutInRegion BGO 1. cm
/gate/physics/Electron/SetCutInRegion BGO 1. cm
/gate/physics/Positron/SetCutInRegion BGO 1. cm
/gate/physics/Gamma/SetCutInRegion NEMAphantom 1. cm
/gate/physics/Electron/SetCutInRegion NEMAphantom 1. cm
/gate/physics/Positron/SetCutInRegion NEMAphantom 1. cm
/gate/physics/SetMaxStepSizeInRegion NEMAphantom 2. cm
#=====
# INITIALISATION
#=====
/gate/run/initialize
#=====
# D I G I T I Z E R: DETECTOR ELECTRONIC RESPONSE
#=====
/gate/digitizer/Singles/insert adder
/gate/digitizer/Singles/insert readout
/gate/digitizer/Singles/readout/setDepth 1
# D E F I N E Q U A N T U M E F F I C I E N C Y O F T H E P H O T O D E T E C T O R
/gate/digitizer/Singles/insert quantumEfficiency
/gate/digitizer/Singles/quantumEfficiency/chooseQEVolume module
/gate/digitizer/Singles/quantumEfficiency/setUniqueQE 0.90
#
/gate/digitizer/Singles/insert blurring
/gate/digitizer/Singles/blurring/setResolution 0.20
/gate/digitizer/Singles/blurring/setEnergyOfReference 511. keV
# NOISE
/gate/distributions/name gaussian
/gate/distributions/insert Gaussian
/gate/distributions/gaussian/setMean 511.0 keV
/gate/distributions/gaussian/setSigma 30.0 keV
# CRYSTALBLURRING
/gate/digitizer/Singles/insert crystalblurring
/gate/digitizer/Singles/crystalblurring/setCrystalQE 0.90
/gate/digitizer/Singles/crystalblurring/setCrystalEnergyOfReference 511.0 keV
# /gate/digitizer/Singles/name moduledeadtime
/gate/digitizer/Singles/insert deadtime
/gate/digitizer/Singles/moduledeadtime/setDeadTime 625 ns
/gate/digitizer/Singles/moduledeadtime/setMode paralyzable
/gate/digitizer/Singles/moduledeadtime/chooseDTVolume module
# /gate/digitizer/Singles/name rsectordeadtime
/gate/digitizer/Singles/insert deadtime
/gate/digitizer/Singles/rsectordeadtime/setDeadTime 625 ns
/gate/digitizer/Singles/rsectordeadtime/setMode paralyzable
/gate/digitizer/Singles/rsectordeadtime/chooseDTVolume rsector
# /gate/digitizer/Singles/insert thresholder

```



```

/gate/digitizer/Singles/threshold/setThreshold 375. keV
/gate/digitizer/Singles/insert upholder
/gate/digitizer/Singles/upholder/setUphold 650. keV
#=====
# C O I N C I D E N C E S O R T E R
#=====
/gate/digitizer/Coincidences/setWindow 6.25 ns
/gate/digitizer/Coincidences/MultiplesPolicy takeAllGoods
/gate/digitizer/name delay
/gate/digitizer/insert coincidenceSorter
/gate/digitizer/delay/setWindow 6.25 ns
/gate/digitizer/delay/setOffset 500. ns
/gate/digitizer/delay/MultiplesPolicy takeAllGoods
#=====
# P A R T I C L E S O U R C E
#=====
/gate/source/addSource F18LineSource
/gate/source/F18LineSource/setActivity 0.001 Ci
# /gate/source/F18LineSource/gps/particle e+
/gate/source/F18LineSource/setForcedUnstableFlag true
/gate/source/F18LineSource/setForcedHalfLife 6586.2 s
/gate/source/F18LineSource/gps/energytype Fluor18
/gate/source/F18LineSource/gps/type Volume
/gate/source/F18LineSource/gps/shape Cylinder
/gate/source/F18LineSource/gps/radius .5 mm
/gate/source/F18LineSource/gps/halfz 700.0 mm
/gate/source/F18LineSource/gps/angtype iso
/gate/source/F18LineSource/gps/centre 0. +45. 0. mm
/gate/source/list
#=====
# D A T A O U T P U T
#=====
# ROOT Output Format
/gate/output/root/enable
/gate/output/root/setFileName GEOutput
/gate/output/root/setRootHitFlag 1
/gate/output/root/setRootSinglesFlag 1
/gate/output/root/setRootCoincidencesFlag 1
/gate/output/root/setRootdelayFlag 1
#=====
# R A N D O M E N G I N E A N D S E E D
#=====
# JamesRandom or Ranlux64 or MersenneTwister
/gate/random/setEngineName JamesRandom
/gate/random/setEngineSeed default
/gate/random/verbose 1
#=====
# M E A S U R E M E N T S E T T I N G S

```

```
#=====
/gate/application/setTimeSlice 10. s
/gate/application/setTimeStart 0. s
/gate/application/setTimeStop 20. s
/gate/application/startDAQ
```

6.2 APÊNDICE B: VALIDAÇÃO INICIAL DO GUIGATE

Os *scripts* a seguir foram criados pelo método automatizado implementado no GUIGATE e são baseados no *benchmark* do GATE, com fins de uma validação inicial dos recursos do GUIGATE. A precisão obtida com o GUIGATE pode ser vista nas Figuras 63 e 64.

Figura 63 – Compara os resultados obtidos de modo automatizado com o GUIGATE e aqueles com o GATE, em seu modo padrão. Os resultados são exibidos para: *Transaxial Detection Position* e *Axial Detection Position*

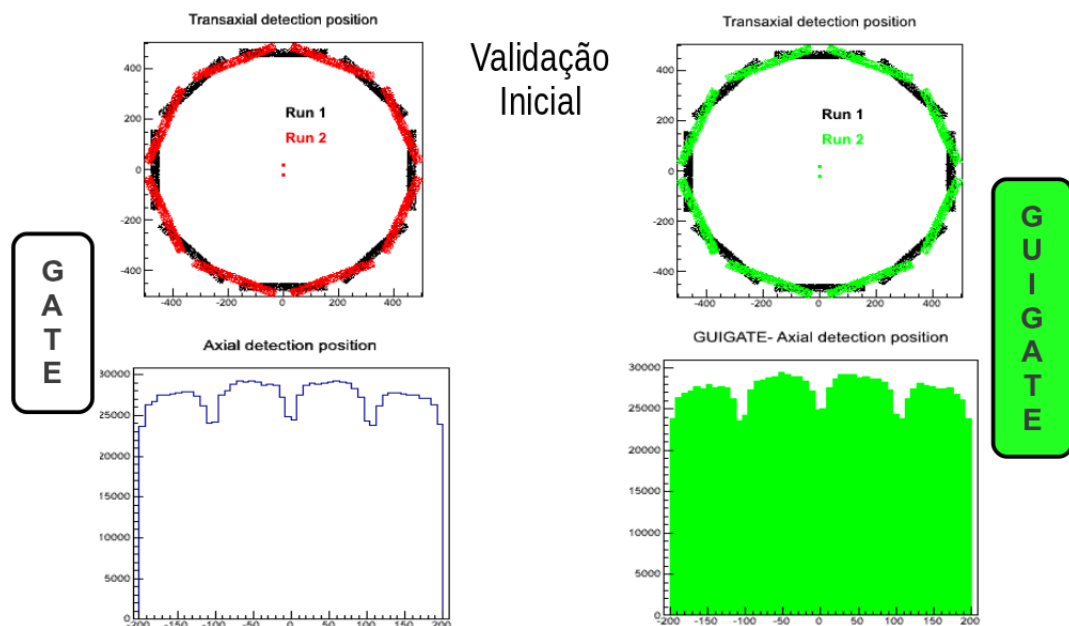
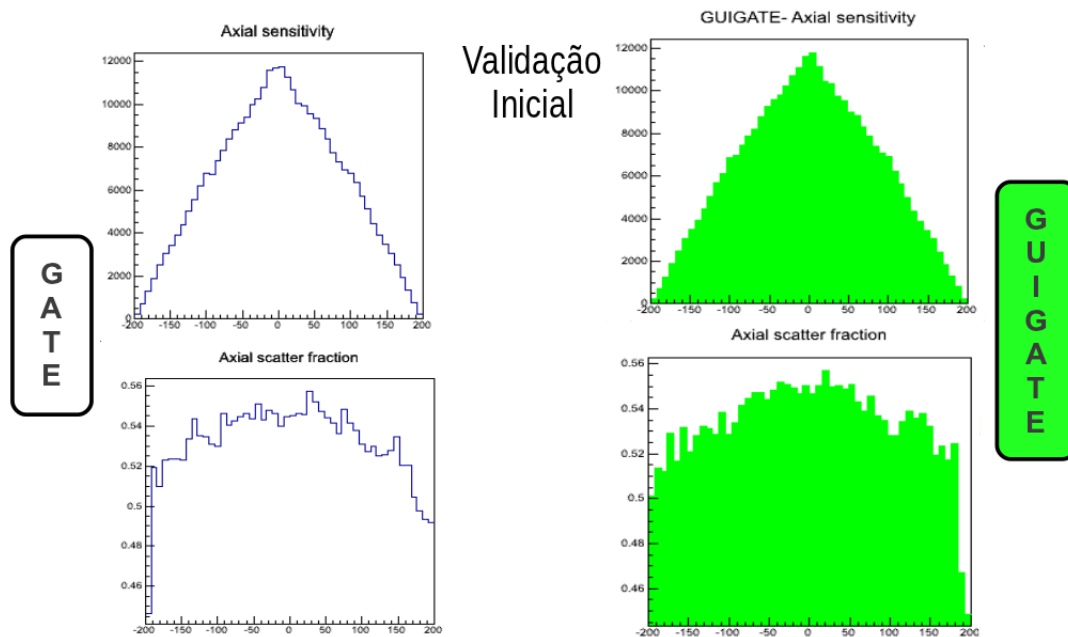


Figura 64 – Compara os resultados obtidos de modo automatizado com o GUIGATE e aqueles com o GATE, em seu modo padrão. Os resultados são exibidos para: *Axial sensitivity* e *Axial scatter fraction*.



Arquivo → scanner.mac ...

```
#=====#
# :) G U I - G A T E : A USER-FRIENDLY TOOLS FOR PET SCANNERs MODELS (:
#=====#
#*****#
# DISCLAIMER
#
# The author of this software system, nor their employing
# institutes, nor the agencies providing financial support for this
# work make any representation or warranty, express or implied,
# regarding this software system or assume any liability for its use.
#
# This code implementation is the intellectual property of the
# Igor Fagner Vieira, Recife - PE - Brazil.
# (Contact: igoradiologia@gmail.com)
# By copying, distributing or modifying the Program (or any work
# based on the Program) you indicate your acceptance of this
# statement, and all its terms.
# *****#
#=====#
# WORLDGEOMETRY#
#=====#
/gate/world/geometry/setXLength 150 cm
/gate/world/geometry/setYLength 150 cm
/gate/world/geometry/setZLength 150 cm
#=====#
```

```

# G E O M E T R Y (CYLINDRICALPET)
#=====# /gate/world/daughters/name cylindricalPET
/gate/world/daughters/insert cylinder
#=====#
#CYLINDER G E O M E T R Y (VolumeName:CYLINDRICALPET)
#=====#
/gate/cylindricalPET/geometry/setRmin 39.9 cm
/gate/cylindricalPET/geometry/setRmax 52.0 cm
/gate/cylindricalPET/geometry/setHeight 40.2 cm
/gate/cylindricalPET/geometry/setPhiStart 0 deg
/gate/cylindricalPET/geometry/setDeltaPhi 360 deg
/gate/cylindricalPET/placement/setTranslation 0.0 0.0 0.0 cm
/gate/cylindricalPET/setMaterial Air
/gate/cylindricalPET/vis/setColor white
/gate/cylindricalPET/vis/forceWireframe
#=====#
# G E O M E T R Y (HEAD)
#=====#
/gate/cylindricalPET/daughters/name head
/gate/cylindricalPET/daughters/insert box
#=====#
# BOX G E O M E T R Y (VolumeName:HEAD)
#=====#
/gate/head/geometry/setXLength 8.0 cm
/gate/head/geometry/setYLength 32.0 cm
/gate/head/geometry/setZLength 40.2 cm
/gate/head/placement/setTranslation 44.0 0.0 0.0 cm
/gate/head/setMaterial Air
/gate/head/vis/setColor white
/gate/head/vis/forceWireframe
#=====#
# G E O M E T R Y (ENDSHIELDING)
#=====#
/gate/head/daughters/name endshielding
/gate/head/daughters/insert box
#=====#
# BOX G E O M E T R Y (VolumeName:ENDSHIELDING)
#=====#
/gate/endshielding/geometry/setXLength 8.0 cm
/gate/endshielding/geometry/setYLength 32.0 cm
/gate/endshielding/geometry/setZLength 0.1 cm
/gate/endshielding/placement/setTranslation 0.0 0.0 0.0 cm
/gate/endshielding/setMaterial Lead
/gate/endshielding/vis/setColor white
/gate/endshielding/vis/forceSolid
/gate/endshielding/repeaters/insert cubicArray
/gate/endshielding/cubicArray/setRepeatNumberX 1
/gate/endshielding/cubicArray/setRepeatNumberY 1

```

```

/gate/endshielding/cubicArray/setRepeatNumberZ 2
/gate/endshielding/cubicArray/setRepeatVector 0.0 0.0 40.1 cm
#=====#
# G E O M E T R Y (SEPTA)
#=====#
/gate/head/daughters/name septa
/gate/head/daughters/insert box
#=====#
# BOX G E O M E T R Y (VolumeName:SEPTA)
#=====#
/gate/septa/geometry/setXLength 5.0 cm
/gate/septa/geometry/setYLength 32.0 cm
/gate/septa/geometry/setZLength 0.05 cm
/gate/septa/placement/setTranslation -1.5 0.0 0.0 cm
/gate/septa/setMaterial Tungsten
/gate/septa/vis/setColor blue
/gate/septa/vis/forceSolid
/gate/septa/repeaters/insert cubicArray
/gate/septa/cubicArray/setRepeatNumberX 1
/gate/septa/cubicArray/setRepeatNumberY 1
/gate/septa/cubicArray/setRepeatNumberZ 3
/gate/septa/cubicArray/setRepeatVector 0.0 0.0 10.0 cm
#=====#
# G E O M E T R Y (MODULE)
#=====#
/gate/head/daughters/name module
/gate/head/daughters/insert box
#=====#
# BOX G E O M E T R Y (VolumeName:MODULE)
#=====#
/gate/module/geometry/setXLength 3.0 cm
/gate/module/geometry/setYLength 8.0 cm
/gate/module/geometry/setZLength 10.0 cm
/gate/module/placement/setTranslation 2.5 0.0 0.0 cm
/gate/module/setMaterial Air
/gate/module/vis/setColor blue
/gate/module/vis/forceWireframe
#=====# # G E O M E T R Y (BLOCK) #=====#
/gate/module/daughters/name block
/gate/module/daughters/insert box
#=====#
# BOX G E O M E T R Y (VolumeName:BLOCK)
#=====#
/gate/block/geometry/setXLength 3.0 cm
/gate/block/geometry/setYLength 1.59 cm
/gate/block/geometry/setZLength 1.99 cm
/gate/block/placement/setTranslation 0.0 0.0 0.0 cm
/gate/block/setMaterial Air

```

```

/gate/block/vis/setColor blue
/gate/block/vis/forceWireframe
#=====#
# G E O M E T R Y (CRYSTAL)
#=====#
/gate/block/daughters/name crystal
/gate/block/daughters/insert box
#=====#
# BOX G E O M E T R Y (VolumeName:CRYSTAL)
#=====#
/gate/crystal/geometry/setXLength 3.0 cm
/gate/crystal/geometry/setYLength 0.3 cm
/gate/crystal/geometry/setZLength 0.38 cm
/gate/crystal/placement/setTranslation 0.0 0.0 0.0 cm
/gate/crystal/setMaterial Air
/gate/crystal/vis/setColor white
/gate/crystal/vis/forceWireframe
#=====#
# G E O M E T R Y (LSO)
#=====#
/gate/crystal/daughters/name LSO
/gate/crystal/daughters/insert box
#=====#
# BOX G E O M E T R Y (VolumeName:LSO)
#=====#
/gate/LSO/geometry/setXLength 1.5 cm
/gate/LSO/geometry/setYLength 0.3 cm
/gate/LSO/geometry/setZLength 0.38 cm
/gate/LSO/placement/setTranslation -0.75 0.0 0.0 cm
/gate/LSO/setMaterial LSO
/gate/LSO/vis/setColor green
/gate/LSO/vis/forceSolid
#=====#
# G E O M E T R Y (BGO)
#=====#
/gate/crystal/daughters/name BGO
/gate/crystal/daughters/insert box
#=====#
# BOX G E O M E T R Y (VolumeName:BGO)
#=====#
/gate/BGO/geometry/setXLength 1.5 cm
/gate/BGO/geometry/setYLength 0.3 cm
/gate/BGO/geometry/setZLength 0.38 cm
/gate/BGO/placement/setTranslation +0.75 0.0 0.0 cm
/gate/BGO/setMaterial BGO
/gate/BGO/vis/setColor yellow
/gate/BGO/vis/forceSolid
/gate/crystal/repeaters/insert cubicArray

```

```

/gate/crystal/cubicArray/setRepeatNumberX 1
/gate/crystal/cubicArray/setRepeatNumberY 5
/gate/crystal/cubicArray/setRepeatNumberZ 5
/gate/crystal/cubicArray/setRepeatVector 0.0 0.32 0.4 cm
/gate/block/repeaters/insert cubicArray
/gate/block/cubicArray/setRepeatNumberX 1
/gate/block/cubicArray/setRepeatNumberY 5
/gate/block/cubicArray/setRepeatNumberZ 5
/gate/block/cubicArray/setRepeatVector 0.0 1.6 2.0 cm
/gate/module/repeaters/insert cubicArray
/gate/module/cubicArray/setRepeatNumberX 1
/gate/module/cubicArray/setRepeatNumberY 4
/gate/module/cubicArray/setRepeatNumberZ 4
/gate/module/cubicArray/setRepeatVector 0.0 8.0 10.0 cm
#REPEATERS VOLUME (Mode: Ring)#
/gate/head/repeaters/insert ring
/gate/head/ring/setRepeatNumber 8
# MOVE VOLUME
/gate/cylindricalPET/moves/insert orbiting
/gate/cylindricalPET/orbiting/setSpeed .1875 deg/s
/gate/cylindricalPET/orbiting/setPoint1 0 0 0 cm
/gate/cylindricalPET/orbiting/setPoint2 0 0 1 cm
# A T T A C H S Y S T E M
/gate/systems/cylindricalPET/rsector/attach head
# A T T A C H S E N S I T I V E D E T E C T O R
# A T T A C H S Y S T E M
/gate/systems/cylindricalPET/module/attach module
# A T T A C H S E N S I T I V E D E T E C T O R
# A T T A C H S Y S T E M
/gate/systems/cylindricalPET/submodule/attach block
# A T T A C H S E N S I T I V E D E T E C T O R
# A T T A C H S Y S T E M /gate/systems/cylindricalPET/crystal/attach crystal
# A T T A C H S E N S I T I V E D E T E C T O R
# A T T A C H S Y S T E M /gate/systems/cylindricalPET/layer0/attach LSO
# A T T A C H S E N S I T I V E D E T E C T O R
/gate/LSO/attachCrystalSD
# A T T A C H S Y S T E M
/gate/systems/cylindricalPET/layer1/attach BGO
# A T T A C H S E N S I T I V E D E T E C T O R
/gate/BGO/attachCrystalSD

```

Arquivo → phantom.mac ...

```

#=====#
# G E O M E T R Y (NEMA CYLINDER)
#=====#
/gate/world/daughters/name NEMACylinder
/gate/world/daughters/insert cylinder

```

```
#=====#
#CYLINDER G E O M E T R Y(VolumeName:NEMACYLINDER)
#=====#
/gate/NEMACylinder/geometry/setRmin 0.0 cm
/gate/NEMACylinder/geometry/setRmax 10.0 cm
/gate/NEMACylinder/geometry/setHeight 70.0 cm
/gate/NEMACylinder/geometry/setPhiStart 0 deg
/gate/NEMACylinder/geometry/setDeltaPhi 360 deg
/gate/NEMACylinder/placement/setTranslation 0.0 0.0 0.0 cm
/gate/NEMACylinder/setMaterial Water
/gate/NEMACylinder/vis/setColor gray
/gate/NEMACylinder/vis/forceSolid
# A T T A C H S E N S I T I V E D E T E C T O R
/gate/NEMACylinder/attachPhantomSD
/gate/endshielding/attachPhantomSD
/gate/septa/attachPhantomSD
```

Arquivo → physics.mac ...

```
#=====#
# P H Y S I C A L P R O C E S S E S #
#=====#
#Add→ PhotoElectric
/gate/physics/addProcess PhotoElectric
/gate/physics/processes/PhotoElectric/setModel StandardModel
#Add→ Compton
/gate/physics/addProcess Compton
/gate/physics/processes/Compton/setModel StandardModel
#Add→ Gamma Conversion
/gate/physics/addProcess GammaConversion
/gate/physics/processes/RayleighScattering/setModel PenelopeModel
#Add→ Low Energy Rayleigh Scattering
/gate/physics/addProcess LowEnergyRayleighScattering
#Add→ Electron Ionisation
/gate/physics/addProcess ElectronIonisation /gate/physics/processes/ElectronIonisation/setModel
StandardModel e-
/gate/physics/processes/ElectronIonisation/setModel StandardModel e+
#Add→ Bremsstrahlung
/gate/physics/addProcess Bremsstrahlung
/gate/physics/processes/Bremsstrahlung/setModel StandardModel e-
/gate/physics/processes/Bremsstrahlung/setModel StandardModel e+
#Add→ Positron Annihilation(Std)
/gate/physics/addProcess PositronAnnihilationStd
#Add→ MultipleScattering for e+
/gate/physics/addProcess MultipleScattering e+
#Add→ MultipleScattering for e-
/gate/physics/addProcess MultipleScattering e-
#Add→ Enable and Initialized List
```



```

/gate/physics/processList Enabled
/gate/physics/processList Initialized
#CUTS IN REGIONS...
/gate/physics/Gamma/SetCutInRegion NEMACylinder 1.0 cm
#CUTS IN REGIONS...
/gate/physics/Electron/SetCutInRegion NEMACylinder 1.0 cm
#CUTS IN REGIONS...
/gate/physics/Positron/SetCutInRegion NEMACylinder 1.0 cm
#CUTS IN REGIONS...
/gate/physics/Gamma/SetCutInRegion LSO 1.0 cm
#CUTS IN REGIONS...
/gate/physics/Electron/SetCutInRegion LSO 1.0 cm
#CUTS IN REGIONS...
/gate/physics/Positron/SetCutInRegion LSO 1.0 cm
#CUTS IN REGIONS...
/gate/physics/Gamma/SetCutInRegion BGO 1.0 cm
#CUTS IN REGIONS...
/gate/physics/Electron/SetCutInRegion BGO 1.0 cm
#CUTS IN REGIONS...
/gate/physics/Positron/SetCutInRegion BGO 1.0 cm

```

```

    Archivo → source.mac ...
#=====#
# S O U R C E #
#=====#
#Add→ Source (GPS -> General Particle Source)
/gate/source/addSource F18
#Add→ Activity...
/gate/source/F18/setActivity 100000 Bq
#Add→ Particle e+
/gate/source/F18/gps/particle e+
#Add→ Unstable Flag...
/gate/source/F18/setForcedUnstableFlag true
#Add→ Types...
/gate/source/F18/gps/energytype Fluor18
#Add→ Half-Life...
/gate/source/F18/setForcedHalfLife 6586.2 s
#Add→ Type(Volume)...
/gate/source/F18/gps/type Volume
#Add→ Type(Cylinder)...
/gate/source/F18/gps/shape Cylinder
/gate/source/F18/gps/radius 0.05 cm
/gate/source/F18/gps/halfz 34.0 cm
#Add→ Ang. Type...
/gate/source/F18/gps/angtype iso
#Add→ Placement...
/gate/source/F18/gps/centre 0.0 -2.0 0.0 cm

```

#Add—> Source (GPS -> General Particle Source)

/gate/source/addSource O15

#Add—> Activity...

/gate/source/O15/setActivity 100000 Bq

#Add—> Particle e+

/gate/source/O15/gps/particle e+

#Add—> Unstable Flag...

/gate/source/O15/setForcedUnstableFlag true

#Add—> Types...

/gate/source/O15/gps/energytype Oxygen15

#Add—> Half-Life...

/gate/source/O15/setForcedHalfLife 122.24 s

#Add—> Type(Volume)...

/gate/source/O15/gps/type Volume

#Add—> Type(Cylinder)...

/gate/source/O15/gps/shape Cylinder

/gate/source/O15/gps/radius 0.05 cm

/gate/source/O15/gps/halfz 34.0 cm

#Add—> Placement...

/gate/source/O15/gps/centre 0.0 +2.0 0.0 cm

#Add—> List...

/gate/source/list

Arquivo —> digitizer.mac ...

#=====#

D I G I T I Z E R

#=====#

#Add—> Adder

/gate/digitizer/Singles/insert adder

#Add—> Adder ReadOut

/gate/digitizer/Singles/insert readout

/gate/digitizer/Singles/readout/setDepth 1

#Add—> Blurring

/gate/digitizer/Singles/insert blurring

/gate/digitizer/Singles/blurring/setResolution 0.26

/gate/digitizer/Singles/blurring/setEnergyOfReference 511 keV

#Add—> Thresholder

/gate/digitizer/Singles/insert thresholder

/gate/digitizer/Singles/thresholder/setThreshold 350.0 keV

#Add—> Upholder

/gate/digitizer/Singles/insert upholder

/gate/digitizer/Singles/upholder/setUphold 650.0 keV

#Add—> Coincidences Sorter

/gate/digitizer/Coincidences/setWindow 120 ns

/gate/digitizer/Coincidences/MultiplesPolicy takeWinnerOfGoods

/gate/digitizer/name delay

/gate/digitizer/insert coincidenceSorter

```

/gate/digitizer/delay/setWindow 120. ns
/gate/digitizer/delay/setOffset 500. ns
/gate/digitizer/delay/MultiplesPolicy takeWinnerOfGoods

```

Arquivo → mainMacro.mac ...

```

#=====
# VISUALISATION
#=====
/vis/disable
#/vis/enable
#/vis/open OGLSX
#/vis/viewer/set/viewpointThetaPhi 60 60
#/vis/viewer/zoom 2
#/vis/drawVolume
#/tracking/storeTrajectory 1
#/vis/scene/add/trajectories
#/vis/scene/endOfEventAction accumulate
#=====
# GEOMETRY #=====
/gate/geometry/setMaterialDatabase GateMaterials.db
#=====
# C O N T R O L E X E C U T E
#=====
/control/execute scanner.mac
#=====
# C O N T R O L E X E C U T E
#=====
/control/execute phantom.mac
#=====
# C O N T R O L E X E C U T E
#=====
/control/execute physics.mac
#=====
# C O N T R O L E X E C U T E
#=====
/control/execute digitizer.mac
#=====
# INITIALISATION
#=====
/gate/run/initialize
#=====
# C O N T R O L E X E C U T E
#=====
/control/execute source.mac
#=====
# M E A S U R E M E N T S E T T I N G S
#=====

```

```
/control/execute Output.mac
#=====
# M E A S U R E M E N T S E T T I N G S
#=====
/gate/application/setTimeSlice 120. s
#=====
# M E A S U R E M E N T S E T T I N G S
#=====
/gate/application/setTimeStart 0.0 s
#=====
# M E A S U R E M E N T S E T T I N G S
#=====
/gate/application/setTimeStop 240. s
/gate/application/startDAQ
```