

Gustavo Ludovico Guidoni

# Framework para Construção de Sistemas de Energia Solar

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Informática da Pontifícia Universidade Católica de Minas Gerais, como requisito parcial para a obtenção do grau de Mestre em Informática.

Belo Horizonte

Junho, 2007

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



**PUC Minas**  
Programa de Pós-graduação em Informática

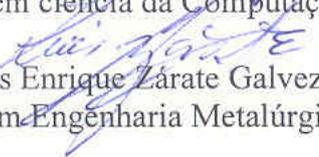
## **FOLHA DE APROVAÇÃO**

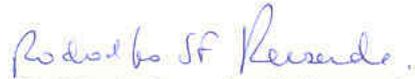
### **Framework para Construção de Sistemas de Energia Solar**

Gustavo Ludovico Guidoni

Dissertação defendida e aprovada pela seguinte banca examinadora:

  
Prof. Mark Alan Junho Song - Orientador (PUC Minas)  
Doutor em ciência da Computação – UFMG

  
Prof. Luis Enrique Zárate Galvez (PUC Minas)  
Doutor em Engenharia Metalúrgica e de Minas (UFMG)

  
Prof. Rodolfo Sérgio Ferreira de Resende (UFMG)  
Doutor em Computer Science (UCSB)

  
Profa. Elizabeth Marques Duarte Pereira (PUC Minas)  
Doutora em Química (UFMG)

Belo Horizonte, 15 de junho de 2007.

## FICHA CATALOGRÁFICA

Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

G948f      Guidoni, Gustavo Ludovico  
            Framework para construção de sistemas de energia solar / Gustavo  
Ludovico Guidoni. – Belo Horizonte, 2007.  
            86 f. : il.

            Orientador: Prof. Dr. Mark Alan Junho Song.  
            Dissertação (mestrado) – Pontifícia Universidade Católica de Minas  
            Gerais, Programa de Pós-graduação em Informática.  
            Bibliografia.

            1. Framework (Programa de computador). 2. Energia solar. I. Song,  
Mark Alan Junho. II. Pontifícia Universidade Católica de Minas Gerais.  
III. Título

CDU: 681.3.06

Bibliotecário: Fernando A. Dias – CRB6/1084

## Resumo

O aumento na demanda da energia elétrica e o interesse sobre o impacto causado pelo uso de combustíveis fósseis vêm estimulando a pesquisa e o desenvolvimento de sistemas relacionados às fontes alternativas de energia, como a energia solar. Há muitos estudos nesta área. Entretanto, os custos para a implantação, gerência e controle ainda são demasiado caros. Há diversos tipos de software dedicados à aplicação de energia solar como: Blast, Doe, Ener-Win, Energy Plus, Hot2000, Passport e Solar5. A disciplina da engenharia de *software* criou novos conceitos a fim de construir aplicações mais flexíveis e reusáveis. *Framework* é um exemplo. *Frameworks* são usados em diferentes contextos tais como interfaces gráficas (Java Swing, AWT), serviços de rede (Java RMI, Jeeves) e trabalhos comerciais (IBM's San Francisco). Aplicando a idéia do reuso do código e do projeto, este conceito auxilia os pesquisadores na geração e na simulação de novas aplicações. Neste trabalho um *frameworks*, denominado *SolarEnergy*, foi desenvolvido. Vários foram os benefícios obtidos com a criação e o uso de novos componentes, a incorporação de novas funcionalidades em aplicações existentes, além da atualização dos dados climáticos.

## **Abstract**

The increase in the demand of electric energy and the concern about the impact caused by fuel use are stimulating research and development of systems related to alternative sources of energy, as the solar one. There are many studies related to this area. However, the costs for implantation, management and control are still far too expensive. There are many types of software devoted to solar energy application such as Blast, Doe, Ener-Win, Energy Plus, Hot2000, Passport e Solar5. However, the software engineering discipline has created new concepts in order to make applications much more flexible and reusable. Framework is an example. Frameworks are used in very different contexts such as graphical interfaces (Java Swing, AWT), net services (Java RMI, Jeeves) and commercial works (IBM's San Francisco). By applying the idea of code and project reuse, it can assist the researchers in the generation and simulation of solar applications. In this work a framework, called SolarEnergy, was developed. It has provided many benefits such as the creation and use of new components, the incorporation of new functionalities into existing applications, and the updating of climatic data.

*A Deus por me dar pais maravilhosos, que sempre me apoiaram em todas  
as fases da minha vida.*

*Ao meu pai que é um exemplo de perseverança e superação,  
meu verdadeiro herói.*

*A minha mãe que é um exemplo de dedicação, que sempre me apoiou nos  
momentos em que precisava e nos que não precisava.*

*A minha esposa que me incentivou, compreendeu e  
apoiou para que fizesse o mestrado.*

# Agradecimentos

Agradeço ao meu irmão que sempre me ouviu, apoiou, incentivou e ajudou na superação dos obstáculos deste trabalho; meu exemplo de dedicação. Ao meu orientador que esteve ao meu lado nesta caminhada. A professora Elizabeth pela oportunidade de desenvolver este trabalho. Ao Frederico Papatella por sua disposição e generosidade em colaborar. A todos os outros que direta ou indiretamente contribuíram para a criação deste trabalho.

Sou a todos muito grato.

# Conteúdo

<b>Lista de Figuras</b>	<b>vii</b>
<b>Lista de Tabelas</b>	<b>ix</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 Objetivos . . . . .	4
1.3 Organização da Dissertação . . . . .	4
<b>2 Energia Solar</b>	<b>6</b>
2.1 Características . . . . .	6
2.2 Radiação Solar . . . . .	7
2.2.1 Métodos de Estimativa da Radiação Solar . . . . .	8
2.3 Coletor Solar . . . . .	15
2.4 Reservatório . . . . .	19
2.5 Conclusão . . . . .	21
<b>3 <i>Framework</i>: SolarEnergy</b>	<b>22</b>
3.1 <i>Framework</i> . . . . .	22
3.2 Requisitos do <i>SolarEnergy</i> . . . . .	25
3.3 O Pacote Energy . . . . .	29
3.4 O Pacote Components . . . . .	32
3.5 O Pacote Library . . . . .	35
3.6 O Pacote Controller . . . . .	35
3.7 Conclusão . . . . .	38

<b>4</b>	<b>Estudo de Caso: SolBrasil</b>	<b>39</b>
4.1	Requisitos . . . . .	39
4.2	Resultados Obtidos . . . . .	41
4.2.1	Radiação Solar . . . . .	41
4.2.2	Eficiência do Coletor Solar . . . . .	45
4.2.3	Dinâmica do Fluido no Reservatório . . . . .	46
4.2.4	Um Sistema de Circulação Forçada . . . . .	47
4.3	Conclusão . . . . .	50
<b>5</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>51</b>
	<b>Bibliografia</b>	<b>53</b>
	<b>Apêndice</b>	<b>56</b>
<b>A</b>	<b>Nomenclatura</b>	<b>57</b>
<b>B</b>	<b>Coeficientes empíricos de Bunnett</b>	<b>59</b>
<b>C</b>	<b>Fórmulas da Radiação Solar</b>	<b>60</b>
<b>D</b>	<b>Fórmulas do Coletor Solar</b>	<b>62</b>
<b>E</b>	<b>Documentação do <i>SolarEnergy</i></b>	<b>66</b>
E.1	Especificação de Requisitos do Framework . . . . .	66
E.1.1	Propósito . . . . .	66
E.1.2	Escopo . . . . .	66
E.1.3	Contexto do Sistema . . . . .	66
E.1.4	Principais Stakeholders . . . . .	67
E.1.5	Referências . . . . .	67
E.1.6	Restrições e Suposições . . . . .	67
E.1.7	Requisitos Funcionais . . . . .	68
E.2	Funcionamento Interno do <i>SolarEnergy</i> . . . . .	69
E.3	O Pacote <i>Controler</i> . . . . .	70
E.4	O Pacote <i>Variables</i> . . . . .	71

E.5	O Pacote <i>Forms</i> . . . . .	72
E.6	O Pacote <i>Database</i> . . . . .	72
E.7	O Pacote <i>Util</i> . . . . .	72

# Lista de Figuras

2.1	Diagrama esquemático dos métodos de estimativa de radiação solar. . . . .	9
2.2	Ângulo de declinação do sol. . . . .	11
2.3	Radiação solar no coletor [PRS <sup>+</sup> 97]. . . . .	16
2.4	Estratificação do reservatório em 3 nodos. . . . .	19
3.1	Inclusão de componentes em um <i>framework</i> . . . . .	25
3.2	Caso de Uso do <i>SolarEnergy</i> . . . . .	26
3.3	Pacotes do <i>SolarEnergy</i> . . . . .	28
3.4	Classes do pacote <i>Energy</i> . . . . .	29
3.5	Classe <i>Energy</i> . . . . .	32
3.6	Classes do pacote <i>Components</i> . . . . .	33
3.7	Criando componentes. . . . .	34
3.8	Clonagem de componentes. . . . .	34
3.9	Classes do pacote <i>Library</i> . . . . .	35
3.10	Classes do pacote <i>Controler</i> . . . . .	36
3.11	Grafo de execução. . . . .	36
3.12	Seqüência de execução do grafo. . . . .	37
4.1	Classes do Sol Brasil. . . . .	40
4.2	Resultados da radiação com $\gamma=-90^\circ$ . . . . .	42
4.3	Resultados da radiação com $\gamma=180^\circ$ . . . . .	43
4.4	Resultados da radiação com $\gamma=90^\circ$ . . . . .	44
4.5	Resultados da radiação com $\gamma=0^\circ$ . . . . .	45
4.6	Evolução do fator de eficiência do coletor ao longo do ano. . . . .	46
4.7	Evolução da temperatura do fluido no reservatório. . . . .	47

4.8	Sistema Termossifão. . . . .	48
4.9	Evolução da temperatura do fluido no reservatório. . . . .	48
4.10	Evolução da temperatura do fluido no reservatório. . . . .	49
4.11	Evolução da temperatura do fluido no reservatório. . . . .	49
E.1	Diagrama de classes . . . . .	74
E.2	Classes do pacote <i>Variables</i> . . . . .	75
E.3	Classes do pacote <i>Forms</i> . . . . .	75
E.4	Classes do pacote <i>Database</i> . . . . .	76
E.5	Classes do pacote <i>Util</i> . . . . .	76

# Lista de Tabelas

4.1	Dados utilizados no primeiro teste . . . . .	42
4.2	Radiação gerada com ângulo azimutal de $-90^\circ$ . . . . .	44
4.3	Radiação gerada com ângulo azimutal de $90^\circ$ . . . . .	44
4.4	Dados do Coletor Solar . . . . .	45
4.5	Dados do Reservatório . . . . .	47
E.1	Stakeholders . . . . .	67
E.2	Stakeholders . . . . .	68

# Capítulo 1

## Introdução

Este capítulo tem por objetivo apresentar o trabalho desenvolvido, assim como a motivação e justificativa para o desenvolvimento do mesmo e os objetivos propostos.

### 1.1 Motivação

O aproveitamento de energia solar como fonte alternativa de energia ainda é pouco explorada no Brasil [Sila]. Países como a China, além de incentivar a utilização da energia solar para o aquecimento da água (energia solar térmica) por parte da população, investe maciçamente na área e como resultado tem se tornado referência mundial neste campo de estudo [Bap06].

O Brasil, por exemplo, é um dos poucos países do mundo a utilizar o sistema de chuveiro elétrico como padrão para aquecimento de água. Nos EUA, Itália, França, Grécia, Alemanha, Austrália e Japão a energia solar é utilizada em massa para aquecimento de água e, em alguns países, o equipamento tem parte significativa doada pelo governo ou pela companhia energética. Em Israel, por exemplo, seu uso é obrigatório [Silb].

O Brasil possui uma incidência de energia solar favorável, pois se encontra, em sua maioria, entre o Equador e o Trópico de Capricórnio podendo produzir entre  $4kWh/m^2$ .dia e  $5kWh/m^2$ .dia em grande parte da sua extensão territorial [CRE]. Devido a este grande

potencial de produção de energia e com a necessidade crescente de produção de energia elétrica, o panorama da utilização de fontes renováveis no Brasil está mudando [Silb]. Além de uma maior produção de energia elétrica, este setor demanda pesquisas para a diminuição dos custos dos equipamentos agregado a uma maior produção de energia. Torna-se oportuno o surgimento de *software* que atenda as necessidades dos pesquisadores (construtores ou professores) na busca de seus objetivos.

A criação de *software* de apoio torna-se inevitável. E isto já acontece, pois pode-se observar o surgimento de diversos *software* aplicados a energia solar, dentre os quais destacam-se o BLAST, DOE (DOEPLUS, VISUAL DOE), ENER-WIN, ENERGY PLUS, HOT2000, PASSPORT e SOLAR5 [Has03] [Doe04] [ARA04]. Estas ferramentas, entretanto, não são customizáveis o suficiente para atender aos requisitos não funcionais dos pesquisadores como portabilidade, extensibilidade, escalabilidade e adaptabilidade. O TrnSys, por exemplo, embora amplamente utilizado em processos de simulação de energia solar é um *software* difícil de ser manuseado, pois necessita de conhecimentos de engenharia para ser operado. Isto não é um grande dificultador para o nicho de pesquisadores que este sistema aborda, entretanto é algo a mais que seus usuários devem lembrar ao operá-lo.

A maioria dos *software* disponíveis não são customizáveis o suficiente para se adaptarem à determinada característica de uma região ou mesmo para se adequarem às necessidades dos pesquisadores. Neste sentido, os especialistas têm procurado na Engenharia de Software novos paradigmas de desenvolvimento de forma a tornar os sistemas mais flexíveis e adaptativos.

Segundo Dias, a Engenharia de Software tem procurado novos conceitos de desenvolvimento para tornar as aplicações mais flexíveis, reusáveis e com maior capacidade de adaptação, dentre as quais destacam-se os *frameworks* [Dia04]. Embora existentes antes mesmo do paradigma de orientação por objetos, os *frameworks* atuais utilizam deste paradigma para encapsular regras de negócios específicas.

O desenvolvimento do paradigma de programação orientado a objetos tornou-se muito popular tanto nas fábricas de *software* como no meio acadêmico. Existem *frameworks* para os mais diferentes domínios de aplicações, dentre os quais citam-se interfaces gráficas (Java Swing, AWT), serviços de redes (Java RMI, Jeeves) e serviços comerciais (IBM's San Francisco) [FPR00].

Os sistemas computacionais de apoio e de simulação, na área de energias renováveis, não exploram todos os benefícios dos *frameworks* como, por exemplo, o reuso e a escalabilidade. Por outro lado esta técnica se mostra interessante visto que, definidos requisitos para um sistema de energia solar, estes podem ser reaproveitadas em qualquer outro sistema desta natureza.

Uma forma de prover o reuso de análise e projeto para sistemas de energia solar é definir as características de cada componente, bem como a comunicação entre os mesmos e a apresentação dos resultados. Fica a cargo do pesquisador apenas implementar as fórmulas ou características desejadas para a pesquisa.

Além de criar uma ferramenta flexível para o estudo de novos produtos ou componentes de energia solar, a reutilização de *software*, incorporada pelo *framework*, traz outros benefícios como a diminuição dos custos de desenvolvimento e manutenção. Possibilita a geração de resultados de pesquisa com melhor qualidade, uma vez que se faz uso de *software* já desenvolvidos, testados e validados. Facilita, também, a criação e a utilização de novos componentes, a incorporação de novas funcionalidades às aplicações existentes, além de permitir atualização dos dados climatológicos, validar o comportamento do sistema e realizar simulações.

Destarte, um *framework* torna-se muito útil no auxílio ao desenvolvimento de novos componentes, pois permite o aprimoramento das fórmulas que os descrevem e, até mesmo, uma maior qualidade das informações que alimentam estes sistemas.

## 1.2 Objetivos

Com o objetivo de auxiliar a pesquisa no campo de aquecimento solar, nesta dissertação foi projetado e desenvolvido um *framework* para a construção de *software* de transformação de energia solar em energia térmica. Na construção do *framework* proposto foram alcançados os seguintes objetivos específicos:

- Definir das características de sistemas aquecimento solar de água;
- Facilitar a incorporação de novos componentes de aquecimento solar;
- Desenvolver um sistema a partir do *framework* proposto que comprove sua usabilidade.
- Validar dos resultados obtidos.

Uma característica desta pesquisa foi a criação de um sistema que comprova a usabilidade, escalabilidade e confiabilidade do *framework* desenvolvido. O sistema, obtido a partir do *framework*, foi implementado com o auxílio do Grupo de Estudos de Energia Solar (Green) com sede na PUC Minas.

A responsabilidade do *framework* é fornecer recursos para que os componentes sejam implementados e adicionados, bem como a correta comunicação entre os mesmos e a apresentação dos resultados. O *framework* fornece recursos para que, além dos componentes, novas funcionalidades sejam incorporadas e relatórios sejam construídos de modo a atender as necessidades dos usuários.

## 1.3 Organização da Dissertação

Esta dissertação está assim organizada: este capítulo a introduz, apresentando as motivações e os principais objetivos alcançados. O Capítulo 2 descreve as características de energia solar de forma a apresentar alguns tipos de componentes e equações, bem como

os formatos e disposições das variáveis que o *framework* manipula. O Capítulo 3 descreve o *framework*, as classes criadas e seus relacionamentos, bem como os respectivos diagramas UML. O Capítulo 4 exhibe o sistema *SolBrasil*, desenvolvido a partir do *framework* proposto, algumas simulações e estudos de casos, e finalmente o Capítulo 5 conclui esta dissertação.

# Capítulo 2

## Energia Solar

Este capítulo apresenta um panorama da utilização de energia solar. Apresenta, também, os principais conceitos para entendimento de radiação solar, coletores e reservatórios - os principais componentes<sup>1</sup> de um processo térmico de captação de energia solar<sup>2</sup>.

### 2.1 Características

Atualmente é impossível imaginar a vida sem qualquer tipo de energia e suas aplicações, tanto em uso doméstico quanto comercial. Há uma necessidade social, econômica, política e ambiental para o surgimento de novas técnicas de geração de energia para que a mesma seja utilizada em pequenas comunidades, em áreas rurais ou, até mesmo, em áreas isoladas.

Uma fonte de energia renovável, limpa e não poluente, disponível em todo o planeta, é a energia solar. Esta energia incidente na superfície terrestre, em um único ano, é 10.000 vezes superior ao consumo anual de energia da humanidade [CRE]. Entretanto, o grande desafio, quanto a energia solar, é transformá-la em energia térmica ou elétrica com custos compatíveis.

---

<sup>1</sup>Dispositivos físicos que compõe um sistema termossifão.

<sup>2</sup>A dissertação aborda apenas a transformação da energia solar em térmica, através do aquecimento da água e o seu armazenamento.

O Brasil, por sua localização geográfica, possui um dos mais elevados índices mundiais de potencial do aproveitamento dessa fonte de energia. Para se ter uma idéia, a incidência solar na região Nordeste varia entre  $1752kWh/m^2$  e  $2190kWh/m^2$  por ano enquanto a superfície do deserto do Saara recebe cerca de  $2000KWh/m^2$  por ano [Pet].

A energia solar incidente no Brasil, em um ano, é de 15 trilhões de MWh, o que corresponde a 21.000 vezes a nossa produção anual de petróleo ou 50.000 vezes o consumo nacional de energia elétrica em 1999 [Sol].

A energia solar tem o potencial para fornecer cerca de 7000 vezes o total de energia consumida mundialmente. Caso fosse coberto 2,6% do deserto do Saara com células solares, todas as necessidades de energia elétrica do planeta seriam supridas [Sol].

Há duas forma de se aproveitar a energia solar incidente na superfície terrestre:

1. Conversão Fotovoltaica - é a geração de energia elétrica através de células fotovoltaicas. Também denominado processo de fotoconversão.
2. Conversão Térmica - é o aproveitamento do calor gerado pela radiação solar, normalmente utilizado para aquecimento de água. Também denominado processo de termoconversão.

Neste trabalho apenas o aproveitamento térmico foi abordado. As seções seguintes descrevem os principais componentes envolvidos neste processo.

## 2.2 Radiação Solar

A energia solar, antes de atingir a atmosfera terrestre, é denominada radiação extraterrestre. Esta radiação, segundo Guimarães, possui uma orientação bem definida da direção Sol-Terra [Gui95]. Entretanto, os raios solares sofrem influência da nossa atmosfera, o que acarreta modificações na intensidade, no espectro da radiação e na própria direção com que atingem a superfície.

Levando em consideração tais situações, a radiação solar global<sup>3</sup>, que incide sobre um corpo localizado no solo, pode ser decomposta em três tipos [Rif05]:

1. Radiação solar direta - é a radiação que atravessa a atmosfera sem sofrer alteração na sua direção;
2. Radiação solar difusa - é a radiação recebida por outro corpo em virtude da direção dos raios solares terem sido modificados por reflexão ou espalhamento na atmosfera;
3. Radiação refletida - é a radiação que depende das características do solo e da vizinhança.

Convém ressaltar que a radiação solar, em um plano horizontal na superfície terrestre, não pode ser determinada com precisão. Isto se deve a vários fatores, dentre eles o próprio eixo de rotação da Terra em relação ao plano da órbita em torno do Sol. Pode variar de acordo com a região, ou seja, com a latitude, a altitude e as condições meteorológicas. Outro fator a ser considerado é a imprecisão dos aparelhos utilizados nas medições - em geral, trabalha-se com uma margem de erro considerada aceitável (bimetálico > 10%; preto e branco J-10%; PSP -2 a 5%) [Rif05].

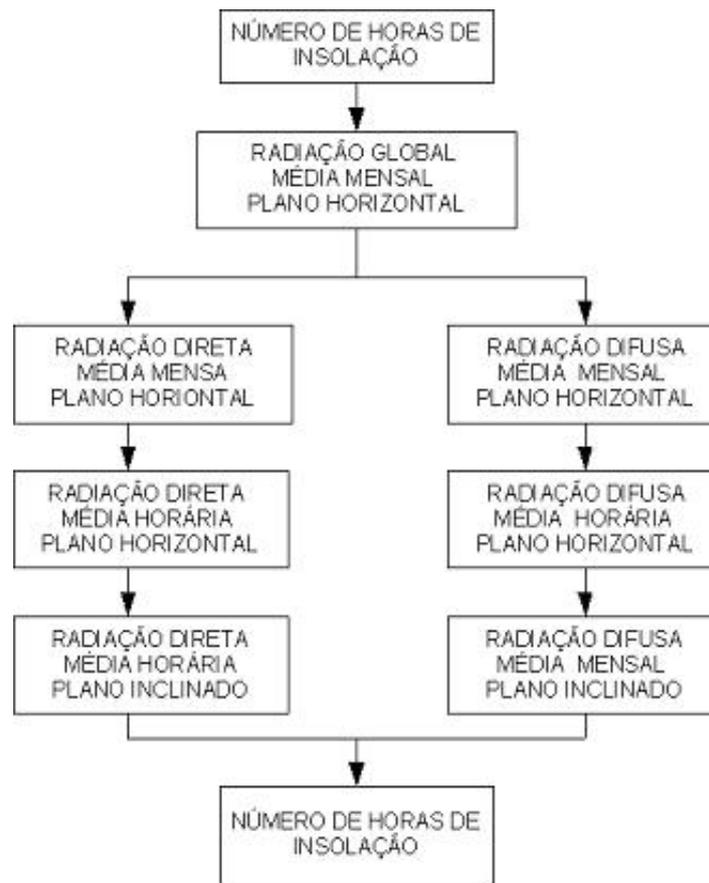
### 2.2.1 Métodos de Estimativa da Radiação Solar

Esta seção apresenta os modelos clássicos que estimam desde a radiação global em média mensal no plano horizontal (a partir do número de horas de insolação) até a radiação global horária em superfícies inclinadas, como mostrado na Figura 2.1.

É importante ressaltar que as equações são estimativas de valores na superfície terrestre, pois não é possível prever as variações climáticas que podem ocorrer em algumas horas, dias ou até meses. Por exemplo, períodos de chuva para uma estação atípica acarretarão distorções nos valores coletados em relação aos calculados. Portanto, é de se esperar que as estimativas funcionem com maior precisão em dias claros [Gui95].

---

<sup>3</sup>Radiação global ou total é a soma das componentes direta e difusa da radiação solar incidente.



**Figura 2.1:** Diagrama esquemático dos métodos de estimativa de radiação solar.

### Radiação Global em média mensal no plano horizontal

A radiação global pode ser calculada a partir do número de horas de insolação. Contudo é necessária a inserção de coeficientes empíricos, os quais dependem da localidade e do dia do ano [Gui95]. A primeira equação para o cálculo da radiação global no plano horizontal é denominada equação de regressão do tipo Angstrom [DB91].

$$\frac{\bar{H}}{H_c} = a + b \frac{\bar{n}}{N} \quad (2.1)$$

onde:

$\bar{H}$  → radiação global diária em média mensal incidente na superfície horizontal;

$\overline{H}_c$  → radiação global diária em média mensal, com condições de céu claro, para a localidade e mês em questão;

$a, b$  → coeficientes empíricos de Angstrom [DB91];

$\overline{n}$  → número de horas de insolação diária em média mensal;

$\overline{N}$  → número de horas de insolação teórico diária em média mensal, obtida pela equação 2.2.

$$\overline{N} = \frac{2}{15} \cos^{-1}(-\tan \phi \tan \delta) \quad (2.2)$$

onde:

$\phi$  → latitude da região;

$\delta$  → ângulo de declinação do sol, representada pela equação 2.3 e ilustrada na Figura 2.2;

$$\delta = 23,45 \sin \left( 2\pi \frac{284 + d}{365} \right) \quad (2.3)$$

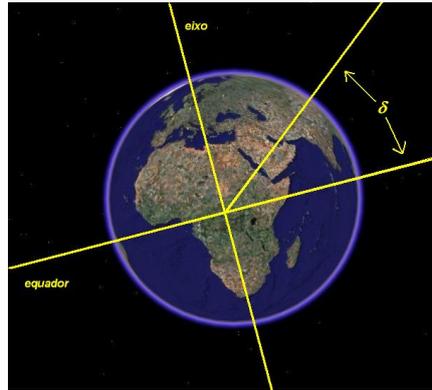
onde:

$d$  → representa o dia do ano - onde  $d=1$  indica o dia 01/janeiro; onde  $d=32$  indica o dia 01/fevereiro e, assim, sucessivamente;

Algumas alterações na equação de Angstrom foram necessárias para adequar o modelo, levando-se em consideração a radiação extraterrestre. Por exemplo, Bennett formulou um novo modelo adicionando os efeitos relativos a altitude e época do ano. O modelo é amplamente utilizado nos dias atuais, sendo entretanto, desenvolvido para o Hemisfério Norte. Sua utilização no Hemisfério Sul necessitou de adaptações [Gui95]:

$$\frac{\overline{H}}{\overline{H}_o} = a + b \frac{\overline{n}}{\overline{N}} + ch \quad (2.4)$$

onde:



**Figura 2.2:** Ângulo de declinação do sol.

$\overline{H}_o \rightarrow$  radiação extraterrestre;

$h \rightarrow$  altitude da estação;

$a, b$  e  $c \rightarrow$  coeficientes empíricos de Bennett.

A equação da radiação extraterrestre, apresentada por Guimarães ([Gui95]), destaca-se a seguir:

$$H_o = \frac{24 * 3600 G_{sc}}{\pi} \left( 1 + 0,033 \cos \left( \frac{2\pi d}{365} \right) \right) (\cos \phi \cos \delta \sin \omega_s + \omega_s \sin \phi \sin \delta) \quad (2.5)$$

onde:

$G_{sc} \rightarrow 1367 W/m^2$  [DB91], é a Constante Solar definida como a radiação solar incidente no topo da atmosfera terrestre por unidade de tempo e área normal à radiação para uma distância média Terra-Sol;

$\omega_s \rightarrow$  hora angular do pôr-do-sol, definida pela equação 2.6.

$$\cos \omega_s = \cos^{-1} (-\tan \phi \tan \delta) \quad (2.6)$$

### Radiação direta e difusa em média mensal

O cálculo da radiação difusa em média mensal foi proposto por Liu e Jordan em 1960 [Gui95]. O modelo calcula a componente difusa em média mensal baseando-se no índice de claridade - isto pode ser comprovado a partir da radiação direta, em média mensal.

Algumas adaptações foram feitas nas equações para incorporar a hora angular do pôr-do-sol, de acordo com o dia médio representativo do mês<sup>4</sup> [DB91]:

$$\frac{\overline{H}d}{\overline{H}} = 0,775 + 0,00606(\omega_s - 90) - [0,505 + 0,0045(\omega_s - 90)] \cos(115\overline{K}_T - 103) \quad (2.7)$$

onde:

$\overline{H}d \rightarrow$  é a radiação difusa em média mensal

$\overline{K}_T \rightarrow$  é o índice de claridade, dado pela equação 2.8.

$$\overline{K}_T = \frac{\overline{H}}{\overline{H}_o} \quad (2.8)$$

### Radiação difusa, direta e global em média horária

O próximo passo é a determinação da radiação direta e difusa global em média horária, ainda na superfície horizontal. Estudos estatísticos da distribuição temporal da radiação em superfícies horizontais levam à formulação do coeficiente  $r_t$  definido como a razão entre radiação horária e diária [Gui95]:

$$r_t = \frac{I}{H} \quad (2.9)$$

onde:

$I \rightarrow$  radiação global horária;

---

<sup>4</sup>É o dia do mês cuja declinação mais se aproxima à média aritmética das declinações diárias.

$H \rightarrow$  radiação global diária.

A equação, expandida por Collares-Pereira ([Gui95]), resulta em:

$$r_t = (a + b \cos \omega) \frac{\cos \omega - \cos \omega_s}{\sin \omega_s - \frac{\pi \omega_s}{180} \cos \omega_s} \quad (2.10)$$

onde:

$\omega \rightarrow$  hora angular do sol, definido como ângulo de deslocamento do sol. Considera-se  $15^\circ$  por hora, assumindo manhã negativo e tarde positivo.

$a, b \rightarrow$  componentes definidas pelas equações 2.11 e 2.12.

$$a = 0,409 - 0,5016 \sin(\omega_s - 60) \quad (2.11)$$

$$b = 0,6609 - 0,4767 \sin(\omega_s - 60) \quad (2.12)$$

Para se chegar à radiação difusa horária, quando se conhece a radiação difusa diária, utiliza-se o mesmo princípio:

$$r_d = \frac{I_d}{H_d} \quad (2.13)$$

onde:

$I_d \rightarrow$  radiação difusa horária;

$H_d \rightarrow$  radiação difusa diária.

Como mostrado por [Gui95], Liu e Jordan propuseram a seguinte equação, tendo a razão das componentes difusa e diária:

$$r_d = \frac{\pi}{24} \left( \frac{\cos \omega - \cos \omega_s}{\sin \omega - \frac{\pi \omega_s}{180} \cos \omega_s} \right) \quad (2.14)$$

### Radiação global em média horária insidente sobre superfície inclinada

Os modelos utilizados nas superfícies inclinadas são mais complexos, sendo necessário conhecer as direções às quais as componentes direta e difusa chegam à superfície de interesse.

Segundo [Gui95], a radiação difusa sob a abóbada celeste, função da nebulosidade e da claridade atmosférica, é decomposta em três componentes:

- Isotrópica - é a radiação recebida uniformemente sobre a superfície de toda a abóbada celeste;
- Circunsolar - é a radiação resultante do espalhamento da radiação que se concentra ao redor do sol;
- Banda do horizonte - é a radiação que se concentra próximo ao horizonte.

Liu e Jordan formularam um modelo levando em consideração o céu isotrópico. O modelo denominado difuso-isotrópico trata os componentes circunsolar:

$$I_T = I_b R_b + I_d \left( \frac{1 + \cos \beta}{2} \right) + I \rho_g \left( \frac{1 - \cos \beta}{2} \right) \quad (2.15)$$

onde:

$I_T$  → radiação solar global sobre a superfície inclinada, em média horária;

$I$  → radiação solar global incidente no plano horizontal;

$I_b$  → radiação direta horária na superfície horizontal;

$R_b$  → razão entre a radiação direta no plano inclinado e horizontal;

$\rho_g$  → reflectância das superfícies próximas que podem ser vistas pela superfície incluída.

## 2.3 Coletor Solar

Coletor solar é um componente capaz de transformar a energia da radiação solar em calor. A energia absorvida pela placa coletora é resultado de um balanço de energia absorvida pelos diversos elementos que compõe o coletor com o meio ambiente e respectivas perdas. Segundo [DB91], a energia total perdida pelo coletor é o somatório das perdas que ocorrem simultaneamente pelos seguintes modelos de transferência de calor :

- Condução  $\rightarrow$  é a transferência de calor entre dois corpos com temperaturas diferentes que estejam em contato físico.
- Convecção  $\rightarrow$  é a transferência de calor relacionado a troca de energia entre um fluido e uma superfície sólida.
- Radiação é a propagação de energia através de ondas eletromagnéticas.

As perdas de energia entre a placa e a cobertura do coletor ocorrem por convecção e por radiação. Em condições normais, a perda de energia entre a placa e a cobertura se iguala a perda de energia entre a cobertura e o ambiente. A radiação solar incidente no plano do coletor pode ser decomposta em três componentes:

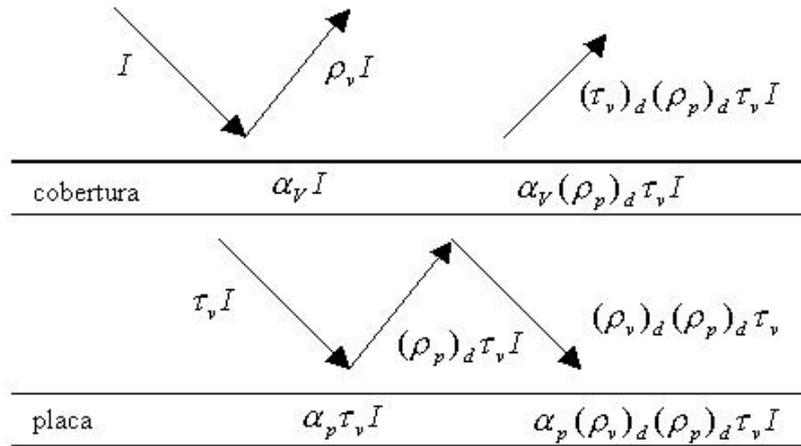
- Radiação direta (representado por  $b$ );
- Radiação difusa proveniente do céu (representado por  $d$ );
- Radiação difusa proveniente do solo ou albedo (representado por  $g$ ).

Na Figura 2.3 estão representados os caminhos da radiação solar no coletor - considere-se até a segunda reflexão pela cobertura, onde:

$I \rightarrow$  radiação incidente no plano do coletor, em média horária;

$\alpha, \rho, \tau \rightarrow$  são, respectivamente, as propriedades óticas da absorvidade, reflectividade e transmissividade dos materiais utilizados;

$v, p \rightarrow$  são as propriedades do vidro (cobertura) e da placa respectivamente.



**Figura 2.3:** Radiação solar no coletor [PRS+97].

As radiações direta, difusa e do albedo compõem a energia solar total no plano do coletor, dado pela equação 2.16:

$$S = \alpha_p [(\tau_v)_b I_{bt} + (\tau_v)_d I_{dt}] [1 + (\rho_v)_d (\rho_p)_d] + I_{gt} \alpha_p (\tau_v)_g \quad (2.16)$$

Onde:

$$I_{bt} = I_b R_b$$

$$I_{dt} = I_d \left( \frac{1 + \cos \beta}{2} \right)$$

$$I_{gt} = \rho_g (I_b + I_d) \left( \frac{1 - \cos \beta}{2} \right)$$

A energia útil absorvida pelo coletor ( $S$ ) é a energia solar total retirada as perdas térmicas, representada pelo coeficiente global de perdas ( $U_L$ ) e a diferença entre a temperatura da placa e do ambiente, dada pela equação abaixo:

$$Q_{util} = A_p [S - U_L (T_p - T_{amb})] \quad (2.17)$$

Onde :

$A_p \rightarrow$  área da placa;

$S \rightarrow$  energia absorvida pela placa;

$T_p \rightarrow$  temperatura da placa;

$T_{amb} \rightarrow$  temperatura do ambiente;

$U_L \rightarrow$  coeficiente global de perda, dado pela soma da energia perdida pelo topo ( $U_{topo}$ ), pela base ( $U_{base}$ ) e pela lateral ( $U_{lateral}$ ) do coletor:).

$$U_L = U_{topo} + U_{base} + U_{lateral} \quad (2.18)$$

A eficiência do coletor  $\eta$  (Equação 2.19) pode ser definida como a energia útil transferida ao fluido ( $Q_{util}$ ) sobre a energia incidente, a qual corresponde a soma da radiação direta, difusa e do albedo, na área do coletor. Dessa forma, quanto maior a quantidade de energia transmitida para a água maior a sua eficiência térmica, ou seja :

$$\eta = \frac{Q_{util}}{(I_{bt} + I_{dt} + I_{gt})A_{placa}} \quad (2.19)$$

Para se determinar a temperatura do fluido na saída do coletor ( $T_{fs}$ ) deve-se utilizar a equação 2.20. De forma análoga a equação 2.21 para temperatura média da placa e a equação 2.22 para a cobertura:

$$T_{fs} = T_{fi} + \frac{Q_{util}}{\dot{m}C_p} \quad (2.20)$$

Onde:

$T_{fi} \rightarrow$  temperatura do fluido na saída do coletor;

$\dot{m} \rightarrow$  vazão mássica do fluido;

$C_p \rightarrow$  calor específico do fluido;

$$T_p = T_{fi} + \frac{Q_{util}/A_p}{U_L F_R} \left(1 - \frac{F_R}{F'}\right) \quad (2.21)$$

Onde:

$F_R \rightarrow$  fator de remoção de energia;

$F' \rightarrow$  fator de eficiência;

$$T_v = T_p - \frac{U_{topo}(T_p - T_{amb})}{h_{p-v} + h_{r,p-v}} \quad (2.22)$$

Onde:

$h_{p-v} \rightarrow$  coeficiente de transferência de calor por convecção entre a superfície absorvedora e a cobertura;

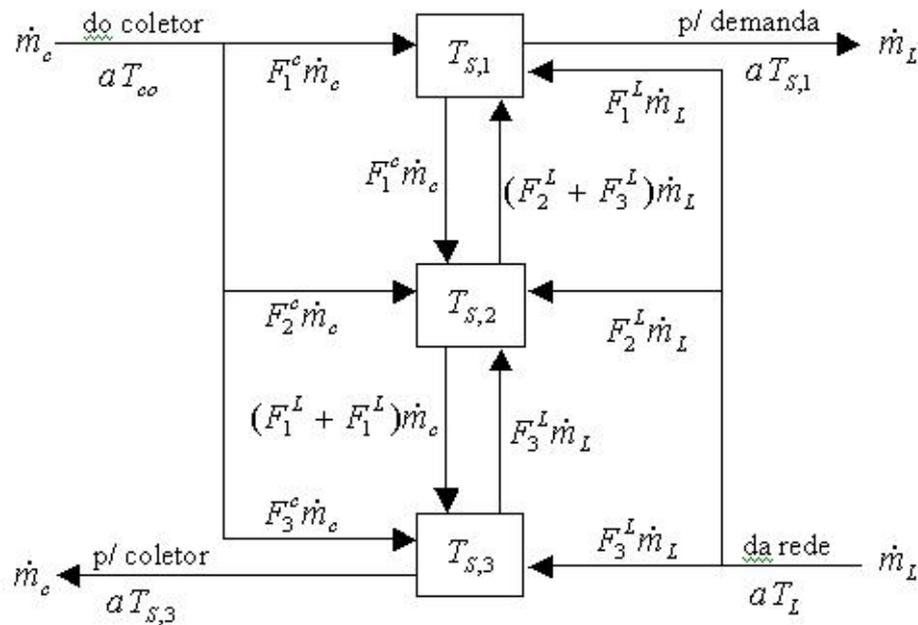
$h_{r,p-v} \rightarrow$  coeficiente de transferência de calor por radiação entre a superfície absorvedora e a cobertura;

Com o passar do dia a eficiência do coletor diminui. Isto se deve ao aumento da temperatura da água que circula pelo mesmo. Com isto a energia útil transferida para o fluido varia no decorrer do dia, conforme o aquecimento da água e da variação da radiação solar. Logo, a cada fração de tempo da simulação, devem ser recalculadas tanto a eficiência como as temperaturas finais instantâneas.

Para a determinação das temperaturas finais é possível a utilização de vários métodos numéricos. Neste trabalho utilizou-se a aproximação sucessiva, uma vez que existe garantia de convergência dos cálculos. Bastando ao se iniciar o processo definir adequadamente a temperatura inicial para o fluido, para a placa e para a cobertura.

## 2.4 Reservatório

Os reservatórios de água operam com níveis significativos de estratificação, ou seja, com a temperatura da água da parte superior do tanque mais quente que a inferior. O reservatório pode ser modelado dividindo-o em N nodos (seções), com balanços de energia definidos para cada nodo, como observado na Figura 2.4.



**Figura 2.4:** Estratificação do reservatório em 3 nodos.

Onde:

$\dot{m}_c \rightarrow$  vazão do coletor;

$\dot{m}_L \rightarrow$  vazão de consumo;

$T_{co} \rightarrow$  temperatura do fluido que chega do coletor;

$F^c \rightarrow$  função de controle do fluido proveniente do coletor nos nodos do reservatório;

$F^L \rightarrow$  função de controle do fluido proveniente da rede de abastecimento nos nodos do reservatório;

$T_{S,i}$  → temperatura no nodo “i”.

Restrições no modelo adotado, exigem que a demanda de água não ultrapasse mais que a metade do volume de um nodo - a equação 2.23 fragmenta  $\Delta t$  em frações de tempo menores reduzindo a quantidade de fluido retirado de um nodo. Para se chegar a temperatura em um nodo “i” utiliza-se a seguinte equação 2.23.

$$T_{S,i}^+ = T_{S,i} + \frac{\Delta}{m_i} \left[ F_i^c \dot{m}_c (T_{co} - T_{S,i}) - F_i^L \dot{m}_L (T_{S,i} - T_L + \dot{m}_c (T_{S,i-1} - T_{S,i}) \sum_{j=1}^{i-1} F_j^c - \dot{m}_L (T_{S,i} - T_{S,i+1}) \sum_{j=i+1}^N F_j^L - \left( \frac{UA}{C_p} \right) (T_{S,i} - T_a) + F_i^{aux} \left( \frac{P\Delta t}{C_p} \right) \right] \quad (2.23)$$

Onde:

$F_i^c$  → função de controle para indicar se o nodo i recebe água proveniente do coletor. Ela vale 1 se a água, que chega do coletor, entra pelo no nodo i e se a temperatura no nodo i ( $T_{S,i}$ ) for menor que a temperatura da água proveniente do coletor ( $T_{c,o}$ ). Nos demais casos esta função vale 0.

$F_i^L$  → função de controle para indicar se o nodo i recebe água proveniente da rede de abastecimento. Ela vale 1 se a água entrar pelo nodo i e se a temperatura da rede  $T_L$  for menor que a temperatura no nodo i ( $T_{S,i}$ ). Nos demais casos esta função vale 0.

$F_i^{aux}$  → função de controle para indicar se é necessário considerar o aquecimento auxiliar por resistência elétrica durante o balanço de energia do nodo que contiver a resistência. Ela vale 1 se i for o nodo que contém a resistência e se a temperatura no nodo i ( $T_{S,i}$ ) for menor que a temperatura mínima indicada no termostato ( $T_{ref}$ ). Nos demais casos esta função vale 0.

## 2.5 Conclusão

Este capítulo apresentou uma modelagem matemática dos principais componentes de um sistema termossifão. O próximo capítulo apresenta conceitos envolvendo *framework* e o próprio *framework* proposto: *SolarEnergy*.

## Capítulo 3

### *Framework: SolarEnergy*

Com a crescente pesquisa na área de energias renováveis, faz-se necessária a elaboração de ferramentas computacionais que possibilitem ao pesquisador, realizar experimentos de novos dispositivos, bem como testar alterações nos materiais ou mesmo nos cálculos realizados em instalações de captação desta energia. Este capítulo apresenta a descrição da estrutura e do funcionamento de SolarEnergy, um *framework* para projeto de aplicações de energia solar.

#### 3.1 *Framework*

Vários problemas surgem no desenvolvimento de um sistema: falhas na aplicação, implementação incompatível com especificação, dificuldade de manutenção e outros. Tais problemas surgem desde a fase de levantamento de requisitos gerando repetição de esforço e planejamento inadequado das atividades [Pie05].

Neste contexto, o reuso é importante por várias razões além da simples economia financeira. Programadores estão constantemente reinventando serviços. A simples cópia do código de outros sistemas é um fator negativo, pois em algumas situações exige a otimização do mesmo.

O reuso de *software* pode ser assemelhado, segundo Govoni [Gov99], com o reuso

na indústria automobilística. Para não *reinventar a roda*, ou partes de um automóvel como alternador e filtro de gasolina, partes comuns são aproveitadas na construção de outros modelos. Se cada modelo começar a *partir do zero* o custo de desenvolvimento será extremamente alto e conseqüentemente terá um preço de venda elevado.

Inúmeras partes de um sistema são desenvolvidas a *partir do zero*, desde o levantamento de requisitos até a codificação. O esforço necessário para se criar estas partes poderia ser melhor aproveitado se as mesmas já estivessem prontas, ou pelo menos parcialmente implementadas - componentes de um sistema, de um mesmo domínio, são parecidos ou até mesmo iguais.

Falar em reuso não é referir simplesmente ao reuso de código, mas sim de todo o trabalho gerado durante o processo de desenvolvimento, abrangendo desde a fase de levantamento de requisitos, codificações e até testes. O próprio processo de desenvolvimento pode ser reutilizado [Gov99].

A idéia de reuso é bastante antiga começando com a noção de sub-rotina, funções e bibliotecas. O reuso sofreu um grande impulso na década de 80 com a criação de sistemas de grandes proporções, onde se obteve vários avanços nas técnicas, nas bibliotecas e nos ambientes de apoio a reutilização [Bar01].

Pietrobon enfatiza alguns motivos pelo qual o reuso não é largamente utilizado nas empresas: a mudança organizacional exigida com a criação de grupos de desenvolvimento, as mudanças na forma de desenvolver *software*, as mudanças culturais tanto no desenvolvimento quanto no gerenciamento, além de serem necessários investimentos cujos retornos só serão obtidos a médio ou longo prazo [Pie05].

Diversas são as técnicas que pregam o reuso de *software*, dentre estas destaca-se a de *framework* que tem sido aplicada a uma grande variedade de domínios [FS97, Jon97, Fac01].

Várias são as conceituações de *framework*: coleções de classes abstratas, interfaces e padrões dedicados a solucionar uma classe de problemas, arquitetura flexível e extensível

que encapsula a parte crítica do projeto e outros [Gov99].

Um *framework* consiste: "em uma mini-arquitetura reutilizável que fornece a estrutura e comportamentos genéricos para uma família de abstração de software, com um contexto que especifica suas interações de uso dentro de um determinado domínio" [Cor04].

Pressman assim o define [Pre02]:

- Um projeto reutilizável de uma ou todas as partes de um sistema, representado por um conjunto de classes e suas interações;
- Um esqueleto, formado por elementos abstratos de uma família de aplicações, passível de ser adaptado para atender as necessidades de uma aplicação específica.

Um *framework* pode ser descrito como um sistema parcialmente completo que deve ser instanciado, além de fornecer uma arquitetura para uma família de sistemas com partes pré-definidas para sua construção [BMR<sup>+</sup>96].

*Frameworks* são um avanço real na reutilização de *software*, uma vez que o reuso não ocorre apenas em pequenas partes de programas, mas sim na parte fundamental do sistema, incluindo a reutilização do projeto. Um *framework* tem a propriedade de capturar as funcionalidades comuns a vários sistemas de um mesmo domínio [Lis00].

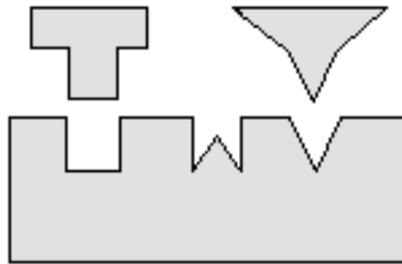
A modularidade, reusabilidade, extensibilidade e inversão de controle dos *frameworks* orientados a objetos, são características apresentadas por Cortês [Cor04]. A modularidade concentra os impactos das mudanças do projeto e de implementação, reduzindo o tempo para entender e manter o *software*, pois são implementados encapsulando detalhes específicos através de interfaces genéricas.

Mesmo havendo reaproveitamento de código e de projeto, os *frameworks* são mais voltados para os desenvolvedores pelo fato de serem expressos por uma linguagem de programação [Gov99].

Um *framework*, entretanto, não necessita estar implementado em uma linguagem de programação para fornecer uma solução parcial a uma família de problemas. Basta forne-

cer especificações e regras de interação entre os objetos para ser considerado um *framework* [Lis00].

Utilizando-se das portas deixadas pelo *framework* (hot-spots [FS97]), os programadores podem adicionar funcionalidades desenvolvidas para uma aplicação específica (Figura 3.1). Isto é possível porque a maioria das decisões arquiteturais já estão definidas - os programadores não necessitam se preocupar com os detalhes de como o *framework* soluciona o problema.



**Figura 3.1:** Inclusão de componentes em um *framework*.

Note, porem, que a utilização de uma nova abordagem de desenvolvimento exige uma curva de aprendizado que deve ser percorrida antes de se tornar útil. Se o aprendizado levar muito tempo há o perigo dos desenvolvedores encontrarem outra solução para o problema proposto

## 3.2 Requisitos do *SolarEnergy*

O principal objetivo do *framework* é possibilitar o desenvolvimento de aplicativos afins de modo que estes possam realizar simulações dinâmicas em sistemas de aquecimento de água utilizando energia solar. O *framework* desenvolvido, denominado *SolarEnergy*, tem o propósito de encapsular a regra de negócio da simulação, controlar a troca de informação, bem como abstrair os conceitos técnicos envolvidos na construção e manipulação dos componentes de um sistema termossifão. O objetivo é prover interfaces gráficas para a

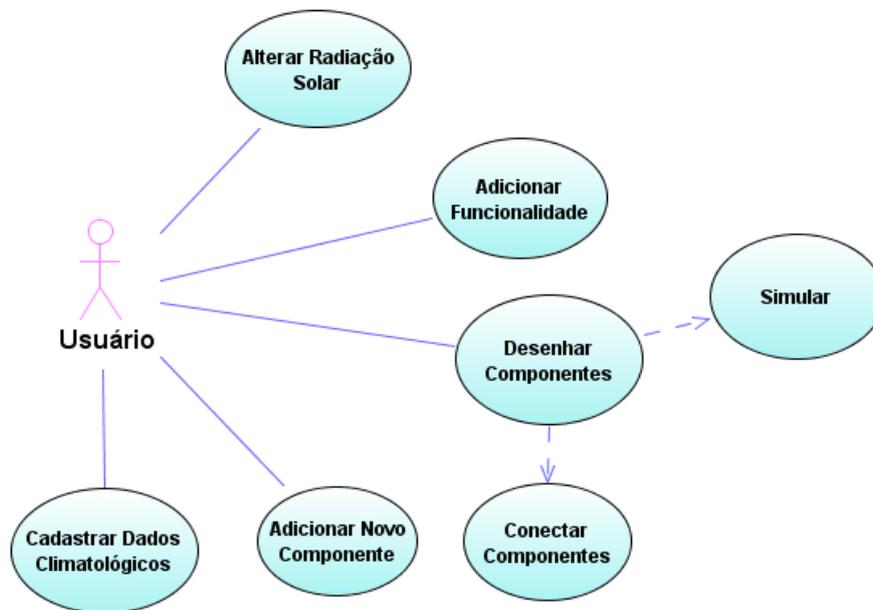
operacionalização tornando o *software* amigável e de fácil manuseio.

Há dois objetivos principais em qualquer simulação que envolva a captação de energia solar. O primeiro é a capacidade de se armazenar a energia capturada (eficiência). O segundo é a quantidade de energia que se consegue utilizar em relação ao capturado pelo dispositivo (eficácia).

*SolarEnergy* foi desenvolvido para simulações de sistemas termossifão. No seu desenvolvimento levou-se em consideração quatro elementos relevantes, dois dos quais críticos: a manipulação dos atributos do sistema e a estrutura de controle capaz de gerenciar as iterações entre os componentes.

Outros dois elementos presentes são: a estrutura de conexão dos componentes e a capacidade de se desenvolver qualquer novo tipo de componente para um sistema termossifão.

A Figura 3.2 ilustra os principais requisitos funcionais do sistema:



**Figura 3.2:** Caso de Uso do *SolarEnergy*.

**Adicionar Novo Componente** → Funcionalidade (*hot-spot*) capaz de expandir o *SolarEnergy* transformando-o em um sistema. Provê um conjunto de classes e interfaces a serem

extendidas e implementadas, de modo a possibilitar a adição de um novo componente (uma piscina por exemplo) sem causar danos no funcionamento do *framework*.

**Cadastrar Dados Climatológicos** → Possibilita a adição ou atualização dos dados climatológicos. Os dados são vinculados a uma cidade, pois para executar uma simulação, deve-se selecionar uma cidade.

**Conectar Componentes** → Tem a responsabilidade de conectar os componentes no sistema termossifão. A direção da informação é indicada por uma seta, a qual informa o destino dos dados.

**Construir Sistema Termossifão** → Adiciona visualmente componentes e conecta-os de modo a formar um sistema termossifão. O usuário pode dispo-los da maneira que desejar. Ao clicar em um componente, este deve disponibilizar suas propriedades para que possam ser alteradas.

**Gerenciar Iterações** → Controlar, através dos dados fornecidos da radiação solar, as iterações que ocorrerão em uma simulação.

**Simular Sistema Termossifão** → Controlar o processo de simulação. Isto implica em validar o sistema termossifão construído pelo usuário; controlar cronologicamente a ordem de execução dos componentes; validar a troca de informação entre os componentes.

**Transferir Valores entre Componentes** → Realiza a transferência dos dados entre os componentes. Caso um atributo esteja apenas na origem, este é desconsiderado não apresentando nenhum erro. O *SolarEnergy* realiza a transferência entre os atributos pelos seus respectivos nomes. Entretanto, é disponibilizado uma funcionalidade para que o usuário possa realizar as ligações entre os atributos da maneira que desejar.

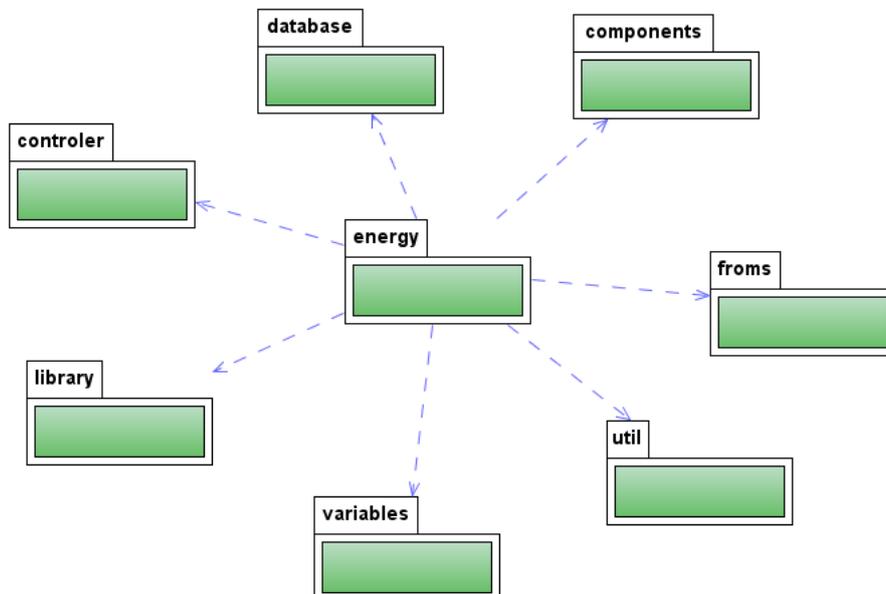
Do ponto de vista do desenvolvedor, o *framework* deve ter a capacidade de incorporar

novas funcionalidades ou mesmo expandir os componentes já existentes [Uch99]. Outros requisitos que dão suporte aos desenvolvedores na extensão do *SolarEnergy* são:

**Adicionar Novas Funcionalidades (telas)** → O *SolarEnergy* disponibiliza um conjunto de classes que devem ser extendidas para que novas interfaces gráficas sejam incorporadas.

**Disponibilizar componentes básicos** → O *SolarEnergy* disponibilizar três componentes básicos: radiação solar; coletor solar e reservatório, os quais podem ser expandidos. A radiação fornece os dados em médias mensais onde os cálculos utilizados foram propostos por Liu & Jordam. Os cálculos da eficiência do coletor solar, bem como os cálculos da dinâmica do fluido no reservatório foram propostos por Duffie ([DB91]).

Para atender estas premissas *SolarEnergy* foi dividido em sete pacotes (Figura 3.3) - as seções seguintes apresentam os principais.



**Figura 3.3:** Pacotes do SolarEnergy.



pular visualmente os componentes. A principal classe é *EMCObject* responsável por exibir e manipular os componentes da simulação. Todo componente desenvolvido é subtipo de *EMCObject* e possui “pontos de conexão” - entradas e saídas específicas para que se estabeleça a comunicação entre os componentes. Um objeto pode ter vários pontos de conexão especificados pelo método *addPoint()*. Os pontos de conexão são representados pela classe *EMCPoint*.

A ligação entre dois componentes, normalmente contém “joelhos” (curvas), que são representados no *SolarEnergy* como vértices. Desta forma a conexão entre dois pontos é feita através de ligações entre vértices. A classe *EMCPipeVertices* tem a responsabilidade de controlar a representação gráfica destes nós.

Na camada de Negócio são encontradas as classes responsáveis por garantir a correta execução de *SolarEnergy* e dos componentes adicionados ao mesmo. Toda regra de negócio envolvida no processo de simulação de um sistema termossifão encontra-se nesta camada.

Como mencionado no capítulo 2, toda e qualquer simulação tem como origem a radiação solar. A energia é capturada por componentes que a transformam em energia térmica ou elétrica, transmitida entre componentes até ser armazenada. Com isto, espera-se que a simulação utilize pelo menos três processos distintos:

1. Cálculo da radiação resultante;
2. Captura da radiação incidente;
3. Manipulação e distribuição da energia capturada.

O *SolarEnergy* foi particionado em três módulos distintos cada qual responsável por uma das etapas citadas acima correspondendo respectivamente as interfaces *IRadiation*, *ICapture* e *IManipulate*.

A interface *IRadiation* é responsável por fornecer dados da radiação solar. Uma diferença essencial desta interface das demais é o fato de não participar dos ciclos de uma simulação, mas sim ditar quantos ciclos a mesma terá.

A principal finalidade da interface *ICapture* é garantir a correta organização dos dispositivos de energia solar na simulação, pois com esta interface não há a necessidade de informar a ordem de execução dos componentes. Por exemplo, a simulação sempre inicia com os componentes de captura.

A interface *IManipulate* especifica o dispositivo físico ou lógico que trata, manipula ou armazena a energia solar capturada. Assume responsabilidades de inferir variações na energia capturada por um determinado dispositivo como, por exemplo, a dinâmica do fluido em um reservatório. Pode assumir, também, a responsabilidade de transporte, como de uma bomba hidráulica, da água aquecida do coletor solar até o reservatório, ou simplesmente monitorar dados que passam por um determinado componente da simulação.

Como as simulações são realizadas com apenas uma fonte de energia (Sol), não deve ser possível instanciar dois componentes de radiação solar. O uso do padrão *Singleton* [GHJV95] garante esta restrição implementada pela classe *EMCRadiation*.

Para que um componente possa manipular os dados recebidos, este deve buscá-los no seu ponto de conexão de entrada e, após efetuado os devidos cálculos, armazenar os resultados no ponto de conexão de saída - cada ponto de conexão é definido pela classe *EMCJoinPoint*.

A comunicação entre os pontos de conexão dos componentes *EMCJoinPoint* é de responsabilidade da classe *EMCPipe* que realiza a passagem dos valores ao invocar o método *transfer()* da classe *EMCPipeTransfer*. O mapeamento das variáveis, de origem e destino, é realizado pela classe *EMCLink*.

Convém ressaltar que um componente não se comunica com outro diretamente, mas seus pontos de conexão se comunicam com pontos de conexão dos demais componentes. O fluxo dos dados de um componente ao outro é assim obtido: o componente informa os valores das variáveis armazenadas em *Variables* no ponto de conexão de saída.

A classe *EMCPipe* informa as *Variables* de saída (origem) e de entrada (destino) dos pontos de conexão à classe *EMCTransfer*. A classe *EMCTransfer* realiza a passagem de

<b>Energy</b>
<i>Attributes</i>
- main : IBasicMain
- radiation : IRadiation
- execute : Execute
- dBase : DataBase
<i>Operations</i>
+ show( ) : void
+ getDatabase( ) : DataBase
+ exec( ) : void
+ addComponent( ) : void
+ setSolarRadiation( ) : void

**Figura 3.5:** Classe *Energy*.

valores entre as variáveis recebidas, utilizando a classe *EMCLink* para o mapeamento. Resta ao componente de destino selecionar os dados em seu ponto de conexão de entrada.

Outra classe especificada nesse pacote é a *Energy* (Figura 3.5). Através desta é possível instanciar *SolarEnergy*, adicionar componentes, telas e realizar simulações. É a classe responsável por disponibilizar os *hot-spots* que permitem alterar o comportamento do ambiente de acordo com as necessidades do pesquisador. Por exemplo o método *addComponet()* adiciona novos componentes ao ambiente, analogamente o método *setSolarRadiation()* substitui o componente radiação solar.

## 3.4 O Pacote Components

A terceira camada, composta pelo pacote *Components*, contém os componentes padrões disponibilizados pelo *framework*: *EGYRadiation* (radiação solar), *EGYColector* (coletor solar) e *EGYStore* (reservatório). A modelagem da terceira camada pode ser observada na Figura 3.6.

Para assegurar que um componente seja manipulado corretamente todo dispositivo desenvolvido deve estender *EMCComponent* e implementar uma das seguintes interfa-

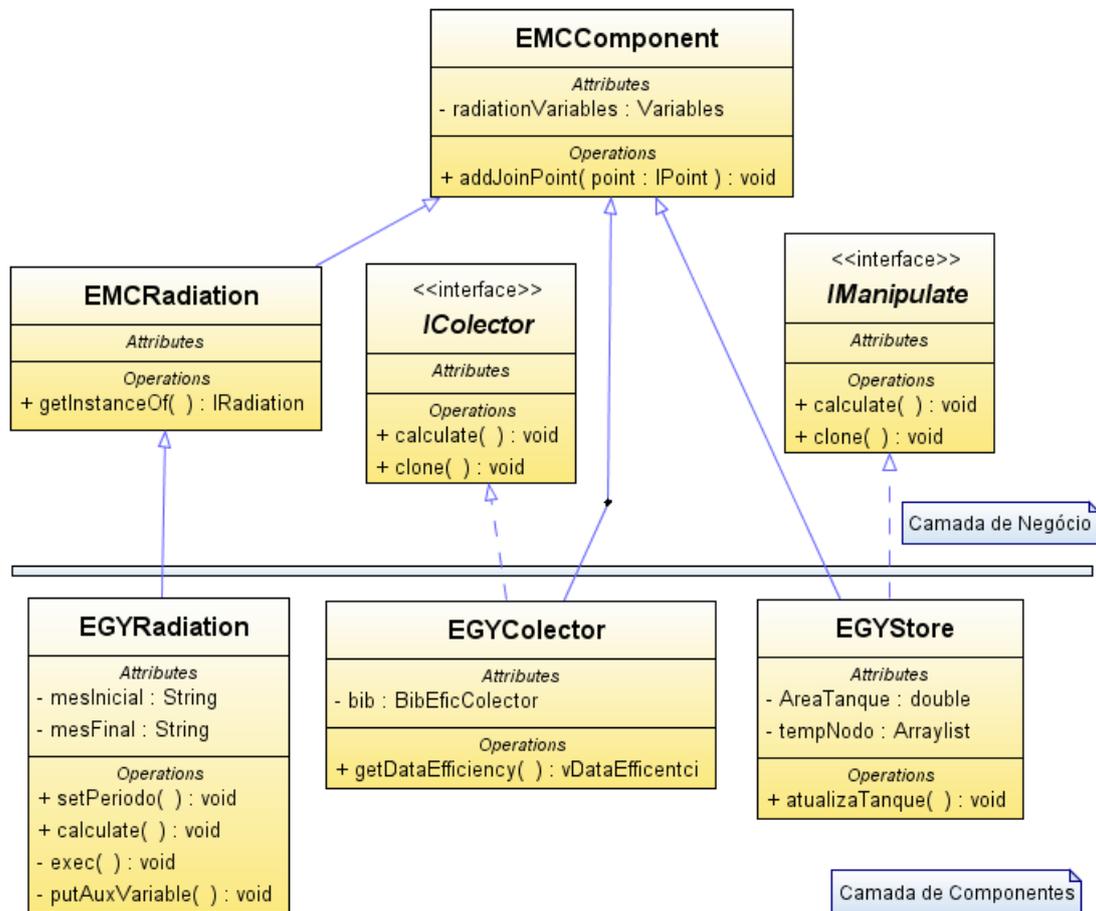


Figura 3.6: Classes do pacote *Components*.

ces: *ICapture* ou *IManipulate*. Assim, é possível garantir que métodos essenciais como *prepareToRun()*, *validateData()* ou *paint()* sejam invocados corretamente. A criação de componentes é ilustrada na Figura 3.7. Todo componente desenvolvido terá seu método *validateData()* e *prepareToRun()* invocados na simulação.

Para que o componente receba dados da radiação solar, basta adicionar a variável desejada em *radiationVariable*. Este atributo, do tipo *Variable*, pertence a classe *EMCComponent*.

O componente radiação solar é uma extensão de *EMCRadiation*. Devido ao padrão *Singleton* implementado, o construtor de *EMCRadiation* foi desenvolvido como privado à

```

public class EGYColector extends EMCCComponent implements IColector, Serializable{
    private Variables inVariables, outVariables;
    ...
    public EGYColector(){
        type = "Colector";
        // variáveis recebidas da radiação solar
        radiationVariables.addVariable("Ib", true, 0.0);
        ...
        inVariables.addVariable("tFluido", true, 0.0, "Temperatura do fluido");
        // adiciona pontos de conexão
        addConnectionPoint("In", 0,10,10,10, inVariables);
        addConnectionPoint("Out", 1,10,10,10, outVariables);
        ...
    }
    public void calculate(){
        double tf = inVariables.getValue("tfluido");
        ...
        // to implement the calculations of heating of the water
        outVariables.setValue("tfluido", eff.getTfs());
    }
}

```

**Figura 3.7:** Criando componentes.

classe, tornando impossível múltiplas instanciações. Para obter uma instância de radiação, basta invocar o método estático *getInstanceOf()*.

Antes de invocar o método *calculate()* de cada componente, *SolarEnergy* associa as variáveis fornecidas pela radiação solar às variáveis adicionadas em *radiationVariable* do respectivo componente. Uma outra premissa importante a todo dispositivo desenvolvido é a capacidade de se clonar. Isto se deve ao fato de um mesmo componente poder ser adicionado repetidas vezes à simulação. A Figura 3.8 apresenta o processo de clonagem.

```

public EGYColector (EGYColector c){
    super.setVariables((EMCCComponent)c);
    inVariables = c.inVariables;
    outVariables = c.outVariables;
    ...
}
public Object clone(){
    return new EGYColector (this);
}

```

**Figura 3.8:** Clonagem de componentes.

A princípio, o padrão de projeto *Abstract Factory* ou *Factory Method* poderia ser

utilizado em substituição ao método *clone()*. Entretanto, o *framework* se tornaria “caixa branca”, uma vez que o desenvolvedor deveria inserir linhas de código no *SolarEnergy* para instanciar novos componentes.

### 3.5 O Pacote Library

O pacote *Library* é uma biblioteca de *SolarEnergy* (Figura 3.4). Nela estão contidas classes especializadas para a resolução de cálculos específicos da radiação (*BibRadiacaoSolar*) e do coletor (*BibEficienciaColetor* e *BibRadiacaoColetor*).

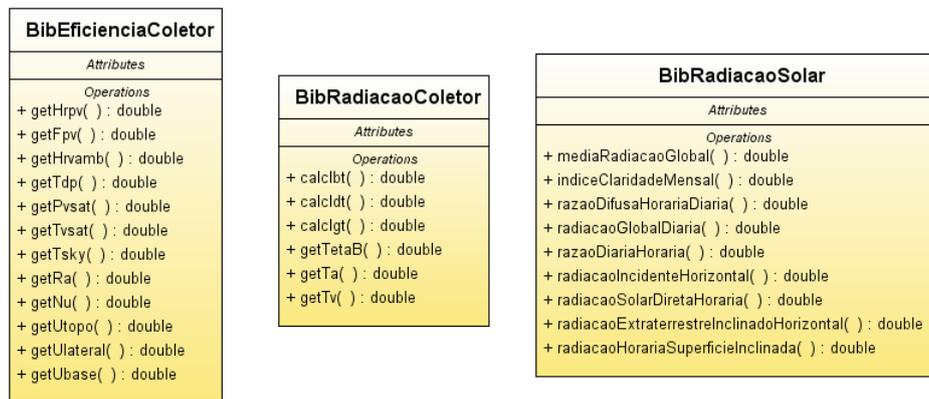


Figura 3.9: Classes do pacote *Library*.

### 3.6 O Pacote Controller

O pacote *Controller* é o cérebro do ambiente. É neste pacote que as simulações são controladas (Figura 3.10). As classes tem como objetivo encapsular as regras de uma árvore de execução controlando cronologicamente a criação, a ordem de validação e execução da mesma.

Algumas considerações quanto ao pacote *Controller*. A classe *Tree* é a responsável por criar, validar e executar o modelo. Dentre as validações estão:

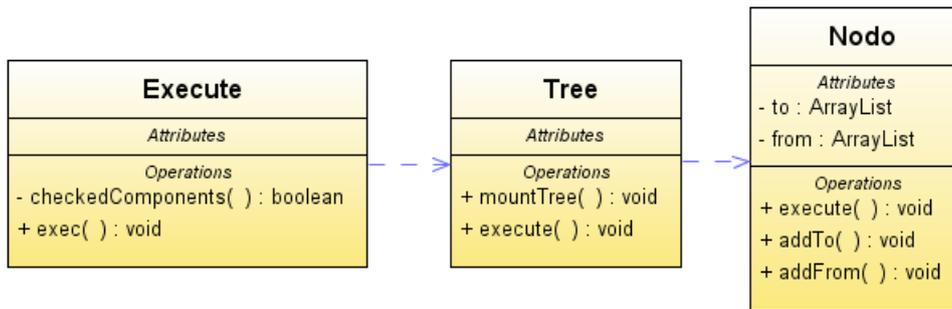


Figura 3.10: Classes do pacote *Controler*.

- **Componentes isolados.** Verificar a existência de componentes desconexos;
- **Atributos necessários.** Identificar se todo componente possui os respectivos dados exigidos para seu perfeito funcionamento;
- **Disposição dos componentes.** Verificar se o sistema a ser simulado apresenta, pelo menos, a existência de um componente de captura e de manipulação.

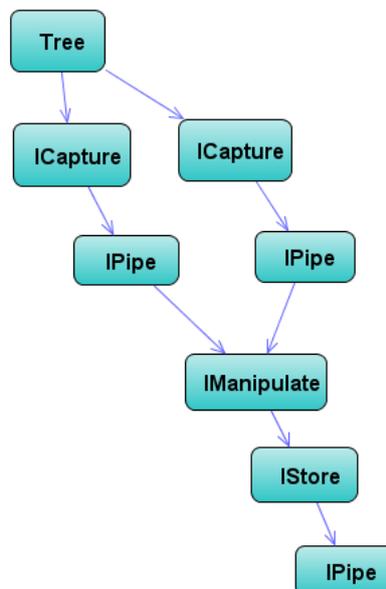
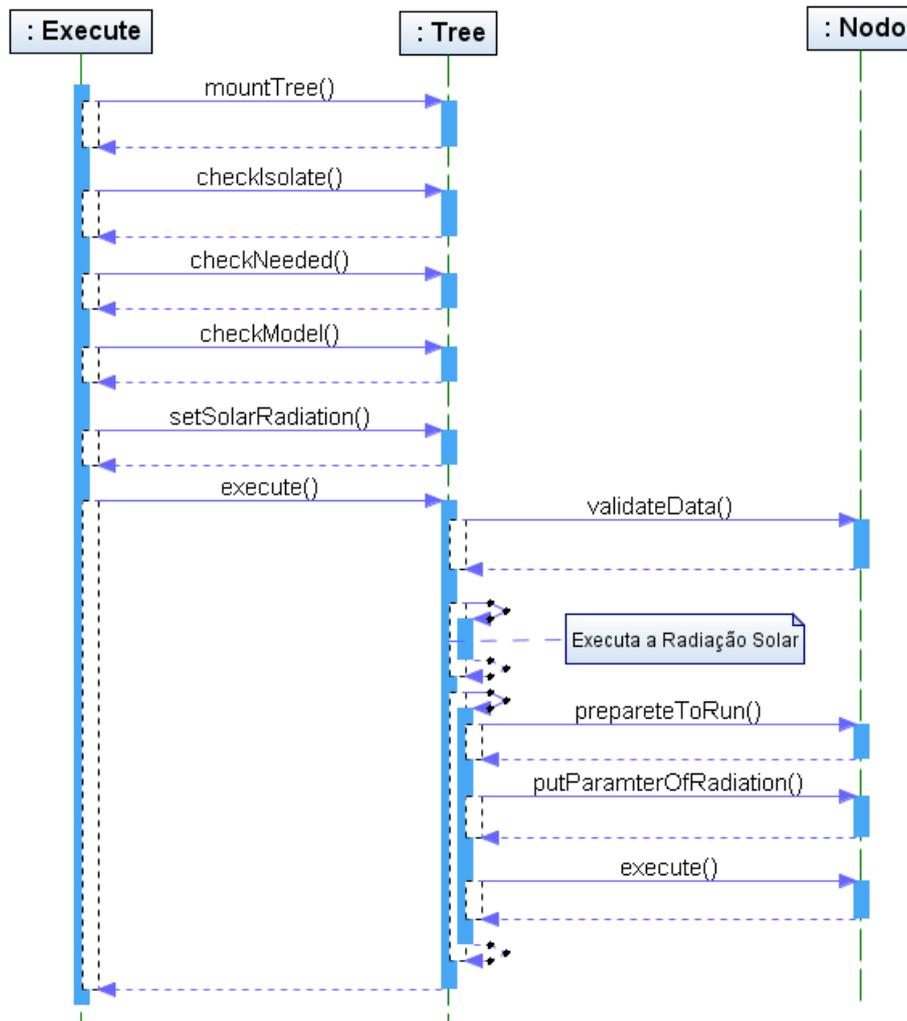


Figura 3.11: Grafo de execução.

O paralelismo entre os componentes é representado por um grafo (Figura 3.11) - cada nodo representa um componente físico e as arestas as conexões entre os mesmos. Toda simulação terá como partida os componentes de captura. A execução do grafo se dá invocando o método *exec()* da classe *Execute* - a Figura 3.12 descreve este processo.



**Figura 3.12:** Seqüência de execução do grafo.

## 3.7 Conclusão

Este Capítulo apresentou as principais estruturas, relacionamentos e organização utilizadas no desenvolvimento do framework, a documentação completa encontra-se em anexo. Baseado neste projeto foi implementado, em Java, o framework *SolarEnergy*. O Capítulo seguinte apresenta *SolBrasil* um sistema projetado a partir do framework proposto.

## Capítulo 4

# Estudo de Caso: SolBrasil

O estudo de caso, apresentado a seguir, foi particularizado para atender as necessidades básicas de desenvolvimento e simulações de conversão de energia solar em térmica. Este fato não impede a sua evolução considerando especificidades de novos componentes para outras aplicações solares. O objetivo principal deste capítulo é utilizar *SolarEnergy* no processo de aquecimento de água.

### 4.1 Requisitos

Os aplicativos disponíveis (Capítulo 2) para o manuseio da energia solar foram criados para engenheiros com profundo conhecimento solarimétrico. Este processo envolve desde a estimativa da radiação no plano do coletor até o armazenamento da água.

Através do *framework SolarEnergy* foi desenvolvido um aplicativo capaz de ser operado por qualquer usuário que saiba manusear o computador. O ambiente, entretanto, é robusto o suficiente para atender necessidades de um pesquisador.

O modelo utilizado nos experimentos foi o de termossifão. *SolarEnergy* já provê através de uma interface gráfica, um conjunto de componentes organizados por finalidades: coletores, reservatórios, bombas, e outros.

O componente da radiação solar *EGYRadiation* provê dados solarimétricos em médias

mensais para qualquer cidade do Brasil. Na implementação de *SolBrasil* os dados são trabalhados em médias horárias. Seguindo a idéia de frameworks, para que isto fosse possível, a partir de *SolarEnergy*, foi necessário apenas estender a classe *EGYRadiation* (figura 4.1).

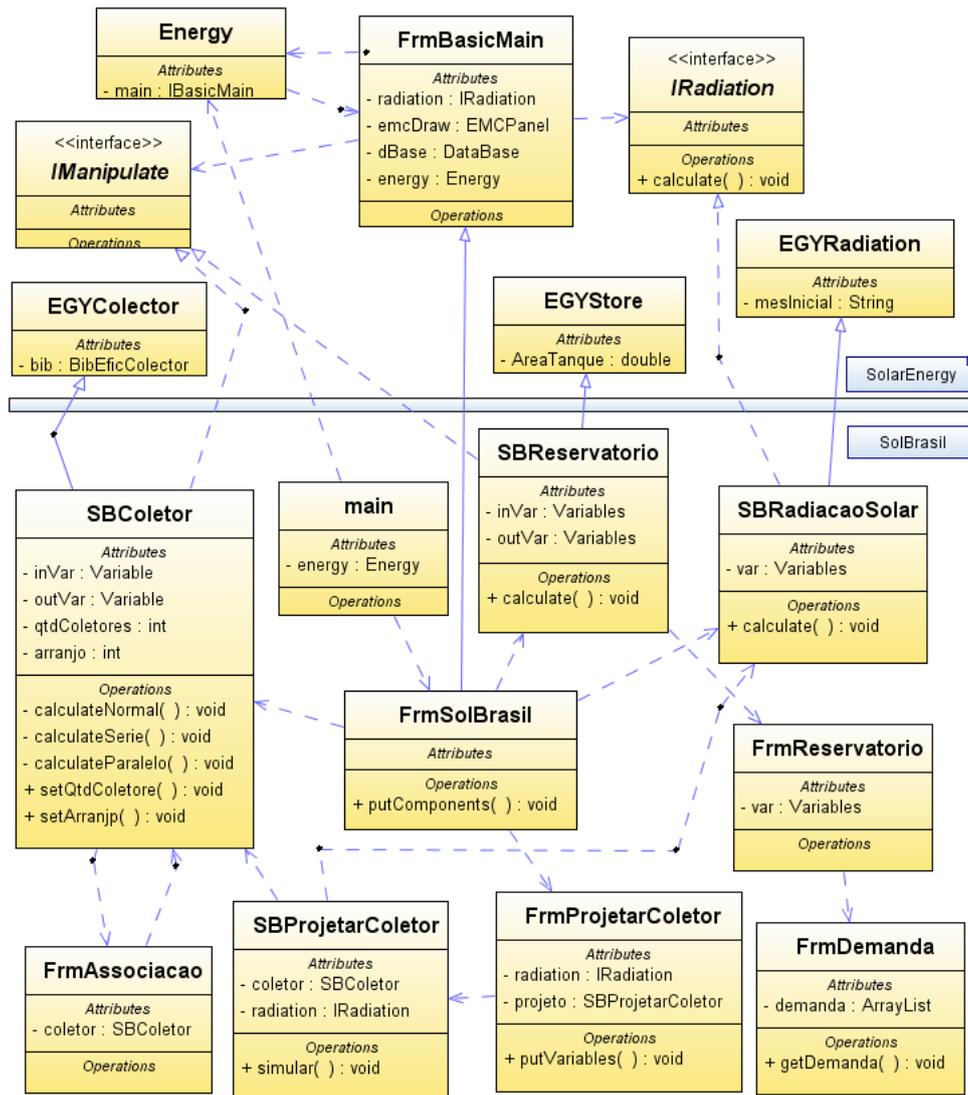


Figura 4.1: Classes do Sol Brasil.

Outro requisito do sistema é ter a capacidade de criar e testar coletores solares, formando uma base de dados por fabricante. O uso do componente *EGYColector* possibilitou estas simulações.

Uma outra característica de coletores é a organização em arranjos que podem variar em série ou paralelo, similar ao circuito elétrico. Quando um arranjo, formado por dois coletores, está em série significa que a água quente que sai de um coletor é utilizada como entrada de outro. Já o arranjo em paralelo significa que a vazão de água é distribuída igualmente entres os coletores, ou seja, a temperatura de entrada da água, para cada coletor, é a mesma. Novamente, a expansão da classe *EGYCollector* possibilitou atender esta necessidade.

## 4.2 Resultados Obtidos

Há diversos aplicativos para cálculos da radiação solar em sistemas termossifão. Entretanto, algumas especificidades dos componentes desenvolvidos para o *SolBrasil* não são contempladas nesses sistemas. Por exemplo, no Trnsys a eficiência do coletor é um dado a ser informado - não é possível calculá-la. No ambiente *SolBrasil* é possível informar as dimensões e materiais e, desta forma, calcular a eficiência esperada para diferentes coletores.

Um outro dificultador para o uso desta ferramentas refere-se aos dados relativos à radiação solar. Na maioria dos sistemas os dados coletados não podem ser facilmente atualizados para atender as necessidades do pesquisador. Com isto, simulações para determinadas regiões podem apresentar variações no resultado.

As seções seguinte apresentam os resultados obtidos de diferentes simulações utilizando *SolBrasil*. Inicialmente, os componentes são avaliados separadamente. A seguir, uma simulação completa é apresentada.

### 4.2.1 Radiação Solar

Foram realizados quatro testes para a obtenção da radiação global diária na superfície do coletor tendo como referência dados da radiação solar para cidade de Belo Horizonte

(Tabela 4.1).

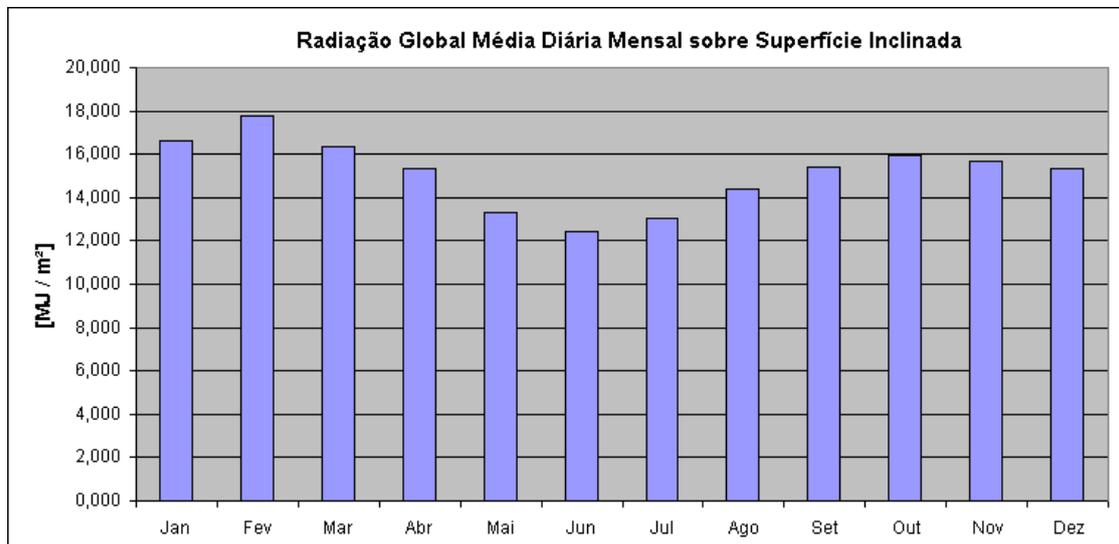
Os testes foram realizados de modo a obter dados que demonstrem o comportamento da radiação no plano do coletor quando o mesmo for posicionado de acordo com os quatro pontos geográficos: norte, sul, leste e oeste.

**Tabela 4.1:** Dados utilizados no primeiro teste

$\phi$ (Latitude)	-19,56'
$\beta$ (Inclinação)	34°
Altitude (m)	850
Reflectividade da vizinhança (Concreto)	0,22

No primeiro teste, privilegiou-se o sol da manhã. Para isto o coletor deve ser direcionado para o Leste, alterando o ângulo azimutal da superfície para  $-90^\circ$ .

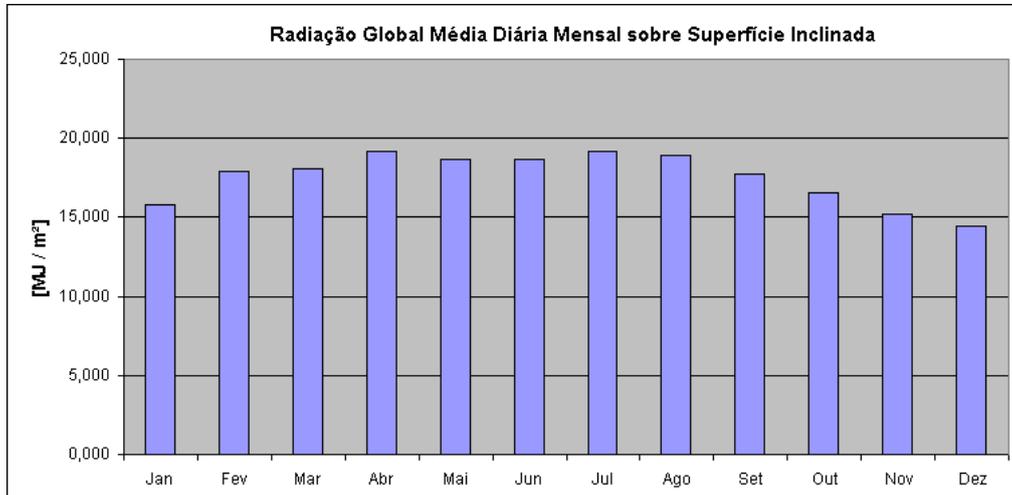
Foi observado que o período com menor incidência de radiação solar sobre o coletor corresponde ao inverno, pois é o período em que o Sol fica menos visível na abóbada celeste (Figura 4.2).



**Figura 4.2:** Resultados da radiação com  $\gamma=-90^\circ$ .

O resultado mostrou não ser uma boa escolha direcionar o coletor para o Sul, pois quando mais se precisa de água quente (no inverno), menor o aproveitamento da radiação.

No segundo teste privilegiou-se o sol do meio dia. O coletor foi direcionado para o Norte, alterando o ângulo azimutal para  $180^\circ$  (Figura 4.3).



**Figura 4.3:** Resultados da radiação com  $\gamma=180^\circ$ .

Embora tenha ocorrido uma redução em janeiro se comparado ao teste anterior, este é o período de menor necessidade de água quente. Observou-se um melhor equilíbrio da radiação solar no decorrer do ano. Este equilíbrio aconteceu pelo fato de, no verão, não ter ocorrido incidência direta da radiação sobre o coletor, enquanto no inverno este fato acontecia.

O terceiro teste privilegiou o sol da tarde direcionando o coletor para oeste. Neste caso o ângulo azimutal foi alterado para  $90^\circ$ . Foi observada uma variação semelhante ao primeiro teste (Figura 4.4). Isto se dá porque a radiação é melhor aproveitada pela manhã. No teste em questão a radiação é melhor aproveitada no período da tarde. Este fato pode ser melhor observado nas tabelas 4.2 e 4.2.

No último teste realizado o coletor solar foi direcionado para o Sul - "de costas para o Sol". Foi observado, no inverno quando o Sol fica mais baixo na abóboda celeste, que a radiação não consegue alcançar a cobertura do coletor nos extremos do dia, ocasionando um baixo aproveitamento da radiação (Figura 4.5).

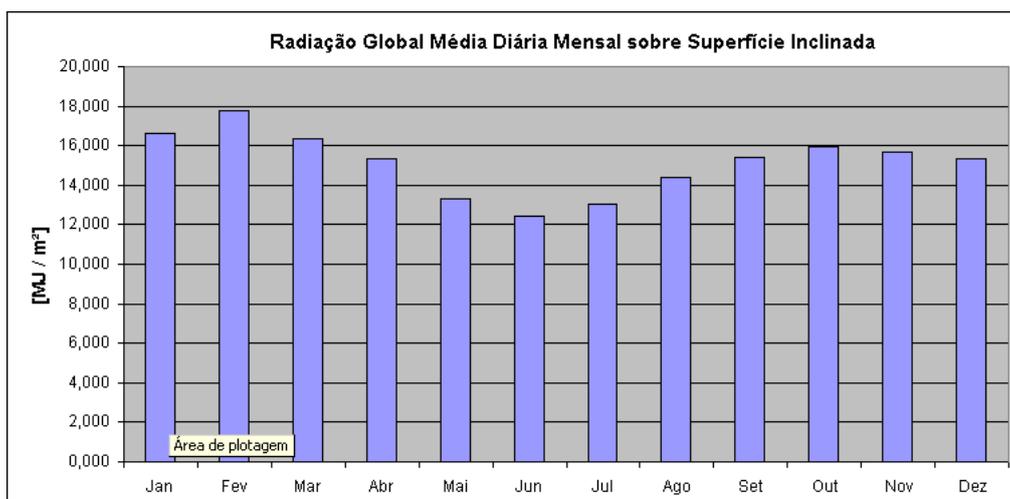


Figura 4.4: Resultados da radiação com  $\gamma=90^\circ$ .

Tabela 4.2: Radiação gerada com ângulo azimutal de  $-90^\circ$ .

Radiação Média Horária Global Sobre Superfície Inclinada [MJ/m²]												
Hora	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
6-7	0,705	0,766	0,650	0,585	0,463	0,417	0,459	0,533	0,586	0,634	0,638	0,625
7-8	1,218	1,347	1,230	1,182	1,028	0,967	1,027	1,114	1,152	1,172	1,133	1,093
8-9	1,696	1,886	1,775	1,745	1,566	1,494	1,568	1,664	1,685	1,677	1,595	1,530
9-10	2,065	2,300	2,194	2,176	1,978	1,898	1,982	2,086	2,096	2,067	1,954	1,871
10-11	2,268	2,520	2,415	2,397	2,186	2,100	2,188	2,300	2,313	2,278	2,154	2,063
11-12	2,274	2,509	2,400	2,365	2,150	2,059	2,145	2,266	2,297	2,277	2,164	2,078
12-13	2,087	2,273	2,156	2,090	1,876	1,783	1,861	1,992	2,055	2,067	1,988	1,920
13-14	1,741	1,857	1,729	1,624	1,420	1,328	1,393	1,531	1,634	1,690	1,659	1,617
14-15	1,297	1,332	1,196	1,054	0,869	0,782	0,830	0,970	1,111	1,212	1,235	1,225
15-16	0,825	0,783	0,647	0,479	0,322	0,245	0,275	0,408	0,575	0,712	0,785	0,803
16-17	0,389	0,287	0,163	0,000	0,000	0,000	0,000	0,000	0,108	0,263	0,369	0,410
17-18	0,036	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,035	0,088
<b>Total</b>	<b>16,600</b>	<b>17,860</b>	<b>16,555</b>	<b>15,697</b>	<b>13,858</b>	<b>13,073</b>	<b>13,727</b>	<b>14,864</b>	<b>15,611</b>	<b>16,048</b>	<b>15,708</b>	<b>15,323</b>

Tabela 4.3: Radiação gerada com ângulo azimutal de  $90^\circ$ .

Radiação Média Horária Global Sobre Superfície Inclinada [MJ/m²]												
Hora	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
6-7	0,036	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,035	0,088
7-8	0,389	0,287	0,163	0,000	0,000	0,000	0,000	0,000	0,108	0,263	0,369	0,410
8-9	0,825	0,783	0,647	0,479	0,322	0,245	0,275	0,408	0,575	0,712	0,785	0,803
9-10	1,297	1,332	1,196	1,054	0,869	0,782	0,830	0,970	1,111	1,212	1,235	1,225
10-11	1,741	1,857	1,729	1,624	1,420	1,328	1,393	1,531	1,634	1,690	1,659	1,617
11-12	2,087	2,273	2,156	2,090	1,876	1,783	1,861	1,992	2,055	2,067	1,988	1,920
12-13	2,274	2,509	2,400	2,365	2,150	2,059	2,145	2,266	2,297	2,277	2,164	2,078
13-14	2,268	2,520	2,415	2,397	2,186	2,100	2,188	2,300	2,313	2,278	2,154	2,063
14-15	2,065	2,300	2,194	2,176	1,978	1,898	1,982	2,086	2,096	2,067	1,954	1,871
15-16	1,696	1,886	1,775	1,745	1,566	1,494	1,568	1,664	1,685	1,677	1,595	1,530
16-17	1,218	1,347	1,230	1,182	1,028	0,967	1,027	1,114	1,152	1,172	1,133	1,093
17-18	0,705	0,766	0,650	0,585	0,463	0,417	0,459	0,533	0,586	0,634	0,638	0,625
<b>Total</b>	<b>16,600</b>	<b>17,860</b>	<b>16,555</b>	<b>15,697</b>	<b>13,858</b>	<b>13,073</b>	<b>13,727</b>	<b>14,864</b>	<b>15,611</b>	<b>16,048</b>	<b>15,708</b>	<b>15,323</b>

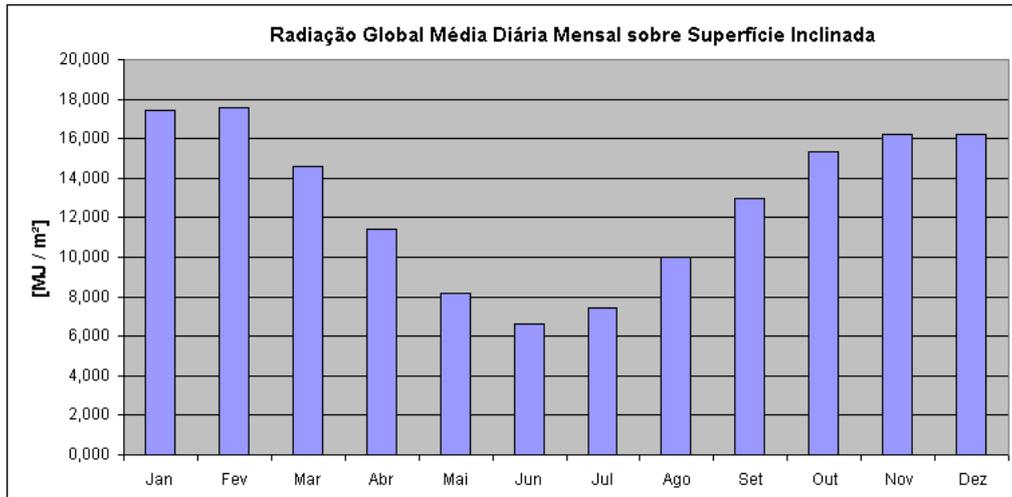


Figura 4.5: Resultados da radiação com  $\gamma=0^\circ$ .

#### 4.2.2 Eficiência do Coletor Solar

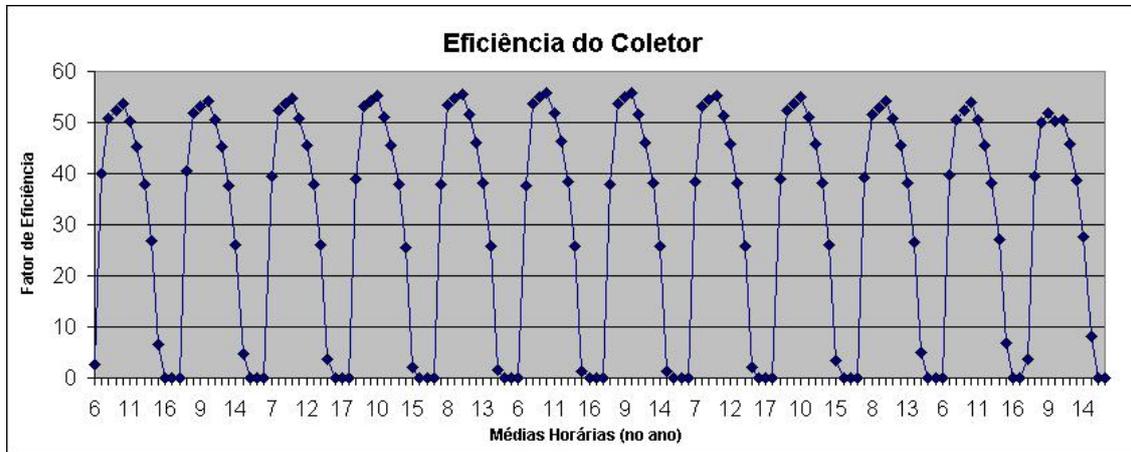
Para cálculo de eficiência do coletor utilizou-se como base os dados obtidos da radiação solar quando o mesmo estava direcionado para o norte. A especificação do coletor, utilizado neste teste, é definida na Tabela 4.4.

Tabela 4.4: Dados do Coletor Solar

Largura (m)	1,1	Comprimento (m)	1,5
Altura da lateral (mm)	55	Distância placa-cobertura (mm)	25
Largura área transparente (m)	0,99	Comprimento Atransparente (m)	0,98
Condutividade isolante	0,025	Número de tubos	8
Diâmetro externo do tubo (mm)	8	Diâmetro interno tubo (mm)	7
Espessura da placa (mm)	0,4	Espessura da cobertura (mm)	3
Espessura isolante base (mm)	25	Espessura isolante lateral (mm)	25
Coefficiente de Extinção	4	Índice de Refração	1,526
Emissividade da cobertura	0,88	Emissividade da tinta	0,9
Absortividade da tinta	0,9	Condutividade da placa	396
Fator de contato (%)	100	Vazão (l/m)	1,2

A Figura 4.6 exibe a variação da eficiência, em médias horárias, para os meses do ano. No início do dia, como a radiação é baixa, o coletor tem baixa eficiência - não aquece o fluido a uma temperatura maior que a inicial. Com o passar do dia este fator aumenta até o limite do coletor - quando não é possível transferir uma quantidade maior de calor

para o fluido. Ao final do dia a radiação diminui, reduzindo a eficiência progressivamente a zero.



**Figura 4.6:** Evolução do fator de eficiência do coletor ao longo do ano.

A eficiência do coletor é nula quando a temperatura de entrada da água for maior que a temperatura de saída, ou seja, o coletor não é capaz de aquecer a água a uma temperatura maior que aquela apresentada à entrada.

### 4.2.3 Dinâmica do Fluido no Reservatório

O balanço de energia, a dinâmica do fluido quente e frio no reservatório, impõe conceitualmente sua divisão em nodos. No *SolBrasil* é possível dividir o reservatório entre 3 a 10 nodos.

Restrições no modelo<sup>1</sup> demandam que dados fornecidos em médias horárias sejam transformados em valores correspondentes a fração 1/4 do volume do tanque - o ciclo de simulação de uma hora deve ser convertido em quatro ciclos de quinze minutos.

Nos testes realizados, o reservatório foi dividido em quatro nodos, com balanço de energia mostrado pela Figura 4.7. Os dados físicos do reservatório são apresentados na Tabela 4.5.

<sup>1</sup>Não é possível retirar mais que 25% do volume total do reservatório.



Figura 4.7: Evolução da temperatura do fluido no reservatório.

Tabela 4.5: Dados do Resrevatório

Espessura do isolante (cm)	10
Condutividade do isolante	0,03
Comprimento (m)	0,8
Diâmetro	0,4

#### 4.2.4 Um Sistema de Circulação Forçada

Nas seções anteriores foram apresentados estudos individuais para cada componente desenvolvido. Nesta seção apresenta-se um estudo de caso clássico de um sistema de circulação forçada envolvendo todos os componentes descritos anteriormente: *radiação solar, coletor e reservatório*.

As simulações foram realizadas levando-se em consideração o mês de janeiro. Os parâmetros de entrada de cada componente são os mesmos já apresentados. A Figura 4.8 ilustra o sistema termossifão. Neste exemplo, utilizam-se três coletores ligados em série e um reservatório. O coletor solar com a indicação *2S* informa um arranjo de coletores conectados em série.

Os resultados deste teste podem ser observados na Figura 4.9. Note que o aquecimento da água foi acelerado em função da existência de três coletores ligados em série.

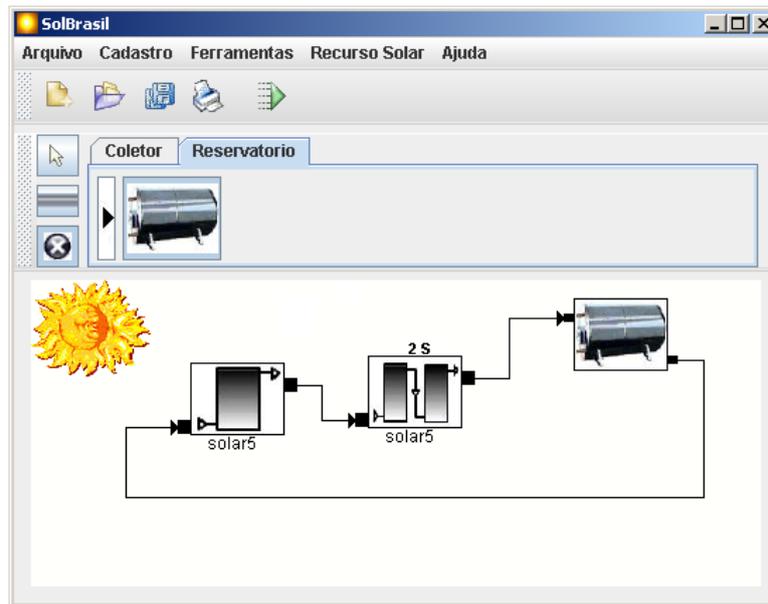


Figura 4.8: Sistema Termossifão.

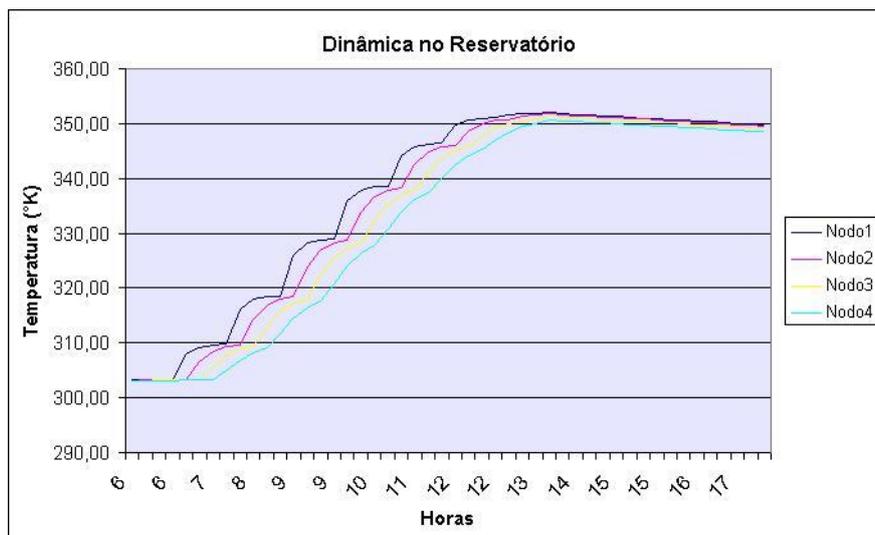
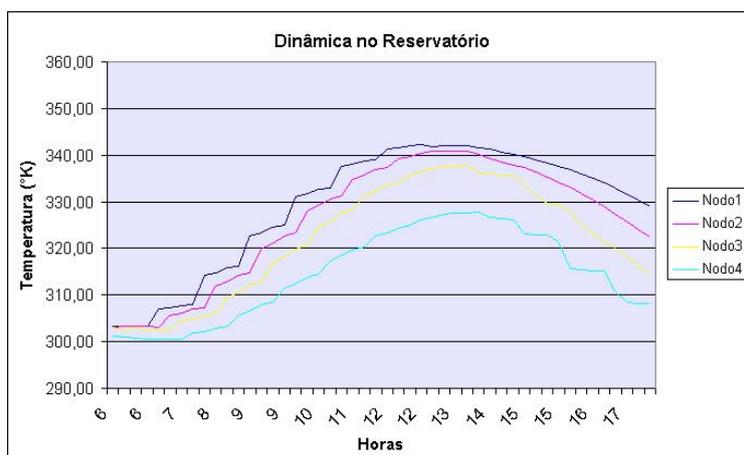


Figura 4.9: Evolução da temperatura do fluido no reservatório.

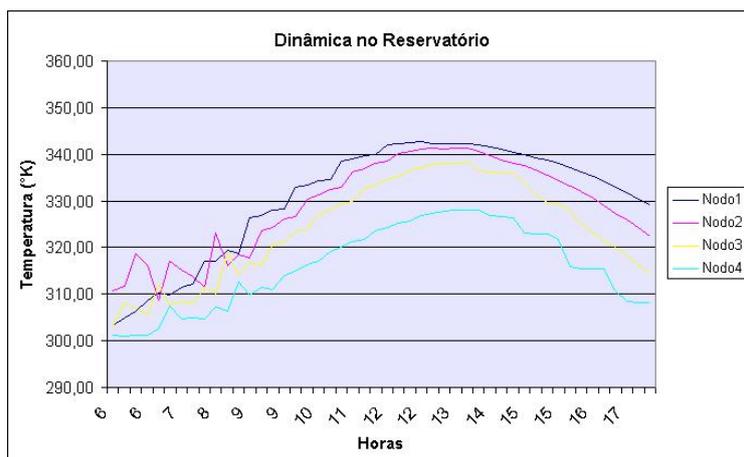
A Figura 4.10 apresenta o mesmo sistema termossifão acrescido de um perfil de consumo de vinte litros de água por hora. Nesta simulação foi observado que a temperatura do nodo quatro se distância dos demais. Este fato ocorre por ser o nodo quatro o elemento responsável pelo fornecimento de água para o coletor, e ao mesmo tempo responsável pela coleta de água fria proveniente da rede.



**Figura 4.10:** Evolução da temperatura do fluido no reservatório.

Ao final do dia os nodos tendem a acompanhar o nodo 4 na queda da temperatura. Isto se deve, principalmente, pelo consumo constante de vinte litros por hora e a perda de eficiência do coletor solar.

O último teste efetuado considera o mesmo sistema termossifão acrescentando, além de um consumo de vinte litros de água por hora, uma resistência elétrica instalada no segundo nodo. Esta resistência, com potência de 3600Wh, é ativada caso a água fique abaixo dos 40°C (Figura 4.11).



**Figura 4.11:** Evolução da temperatura do fluido no reservatório.

Observe que no início do dia a resistência foi acionada, tornando o nodo 2 e, em

alguns momentos o nodo 3, mais quente que o nodo 1. Com o passar do dia, mesmo com um consumo de vinte litros por hora, os coletores solares foram capazes de suprir as necessidades de consumo.

### 4.3 Conclusão

Este Capítulo apresentou *SolBrasil*, um aplicativo para simulação de sistemas termosifão. Pôde-se observar através deste a aplicação de *SolarEnergy*. O Capítulo seguinte apresenta as conclusões e trabalhos futuros.

## Capítulo 5

# Conclusões e Trabalhos Futuros

Frameworks relacionados ao domínio de utilização para energia solar são pouco comuns. Embora existam diferentes *software* de simulação para aquecimento de água, como o Trnsys exigem um amplo conhecimento do domínio do problema e realizam a criação e a incorporação de novos componentes através da utilização de plugins. Embora esse conceito seja bastante útil, existem algumas limitações para a reutilização de código fonte.

Este trabalho além de ter apresentado algumas definições de sistemas aquecimento solar de água, em especial as relacionadas ao processo de aquecimento de água, implementou o *framework* proposto, denominado *SolarEnergy*.

Pôde-se concluir, através da utilização deste, que o desenvolvimento de novos aplicativos de controle, simulações e ensaios realizados por pesquisadores da área de sistemas de energia solar foram simplificados, facilitando, assim, a incorporação de novos componentes. Além disso, o tempo e o custo do processo de desenvolvimento são reduzidos, uma vez que os projetistas reutilizarão uma solução proposta já testada e consolidada.

Para demonstrar a criação de um *software* a partir do paradigma estudado, foi desenvolvido, como estudo de caso, um aplicativo denominado *SolBrasil*. Foram implementadas todas as operações necessárias para um estudo detalhado de conversão de energia solar em térmica. Através do sistema desenvolvido (*SolBrasil*), o *framework* se mostrou escalável

e com uma curva de aprendizagem menor do que se contruísse todo este arcabolo para um experimento científico.

Os dados coletados pelo SolBrasil foram verificados com auxílio do grupo de estudos do GREEN, mostrando a confiabilidade do *SolarEnergy* na produção de dados.

Como trabalho futuro pretende-se estender o domínio de aplicação de *SolarEnergy* para a incorporação de aspectos de conversão de energia solar em energia fotovoltaica, ou seja, incorporar no framework aspectos relevantes a transformação da energia solar em elétrica.

Além disso, pode-se agregar ao framework o balanceamento do fluido em arranjos de coletores em paralelo. Neste caso, levando em consideração que cada coletor pode receber na entrada uma distribuição de água diferenciada tanto quanto ao volume quanto a temperatura.

Como existem diferentes métodos para extração da radiação solar há, também, a possibilidade de ser implementado um padrão strategy para que o próprio framework decida a melhor abordagem a ser utilizada.

## Bibliografia

- [ARA04] Omar M. Al-Rabghi and Mohammed M. Akyurt. A survey of energy efficient strategies for effective air conditioning. *Energy Convers Manage*, (45):p. 1643–1654, 2004.
- [Bap06] Alessandra Sleman Cardoso Baptista. Análise da viabilidade econômica da utilização de aquecedores solares de Água em resorts no nordeste do brasil. Mestrado, COPPE/UFRJ, Rio de Janeiro, 2006.
- [Bar01] Álvaro Cesar Pereira Barbosa. *Middleware para Integração de Dados Heterogêneos Baseado em Composição de Frameworks*. Doutorado, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2001.
- [BMR<sup>+</sup>96] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal, Peter Sommerlad, and Michael Stal. *Pattern-Oriented Software Architecture: A System of Patterns*. John Wiley and Sons, New York, NY, USA, 1996.
- [Cor04] Sérgio da Costa Cortes. *A Transaction Model for DBMS Integration to a Mobile Computation Environment*. Ph.d. thesis, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2004.
- [CRE] CRESESB. Centro de referência para energia solar e eólica sérgio de salvo brito. disponível em: <http://www.cresesb.cepel.br/faq-solar.htm> . acesso em: 15 mai. 2006.

- [DB91] J. A. Duffie and W. A. Beckman. *Solar Engineering of Thermal Processes*. John Wiley & Sons, New York, 1991.
- [Dia04] Viviane Cristina Dias. Proposta de um framework conceitual de dados e padrões de análise para a definição de um banco de dados em qualidade de energia elétrica. Mestrado, Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte, 2004.
- [Doe04] Ian Ross Doebber. Investigation of concrete wall systems for reducing heating and cooling requirements in single family residences. Master's thesis, Virginia Polytechnic Institute and State University, Virginia, 2004.
- [Fac01] P. W. Fach. Design reuse through frameworks and patterns. *Software, IEEE*, 18(5):71–76, 2001.
- [FPR00] Marcus Fontoura, Wolfgang Pree, and Bernhard Rumpe. Uml-f: A modeling language for object-oriented frameworks. *14th European Conference on Object Oriented Programming*, pages p. 63–82, 2000.
- [FS97] Mohamed E. Fayad and Douglas C. Schmidt. Object-oriented application frameworks. *Communications of the ACM*, 40(10), October 1997.
- [GHJV95] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Massachusetts, 1995.
- [Gov99] Darren Govoni. *Java application frameworks*. John Wiley & Sons, New York, NY, USA, 1999.
- [Gui95] A. P. C. Guimarães. Estudos solarimétricos com base na definição do mês padrão e seqüencial de radiação diária. Mestrado, Universidade Federal de Minas Gerais, Belo Horizonte, 1995.

- [Has03] Marwa M. Hassan. *Framework for Active Solar Collection Systems*. Tese phd, Department of Building Construction: Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 2003.
- [Jon97] R. E. Jonhson. How frameworks compare to other object-oriented reuse techniques. *Communications of the ACM*, pp. 39-42, 40(10), 1997.
- [Lis00] Jugurta Filho Lisboa. *Projeto Conceitual de Banco de Dados Geográficos através da Reutilização de Esquemas, Utilizando Padrões de Análise em um Framework Conceitual*. Doutorado, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.
- [Pet] Petrobras. Energia solar. disponível em: [http://www2.petrobras.com.br/petrobras/portugues/per\\_energrenov.htm](http://www2.petrobras.com.br/petrobras/portugues/perfil/per_energrenov.htm) . acesso em: 15 mai. 2006.
- [Pie05] C. A. M. et al Pietrobon. *SBC. (Org.). Escola Regional de Informática (ERI)*, volume v., chapter Reuso de Software com Qualidade, pages p. 185–227. Fumarc, Belo Horizonte, 2005.
- [Pre02] Roger S. Pressman. *Engenharia de Software*. McGraw-Hill, Rio de Janeiro, 5 ed edition, 2002.
- [PRS<sup>+</sup>97] E. M. D. PEREIRA, J. M. G. ROCHA, R. SCHIRM, J. F. FALEIRO, and F. M. MONTANDON. Siscos 3.0 - dimensionamento de instalações solares de médio e grande portes. *III Congresso Iberoamericano de Ingeniería Mecânica. III CIDIM*, 1997.
- [Rif05] Douglas Bressan Riffel. Unidade de osmose reversa acionada por energia solar fotovoltaica sem baterias: Simulação, projeto e validação experimental. Mestrado, Universidade Federal do Ceará, Fortaleza, 2005.

- [Sila] J. L. Silveira. Fontes alternativas podem completar abastecimento. disponível em: <http://www.comciencia.br/reportagens/energiaeletrica/energia04htm> . acesso em: 13 mar. 2006.
- [Silb] J. L. Silveira. O dilema da garantia de energia elétrica com baixo custos e reduzidos impactos ambientais: a solução para a crise energética do país. disponível em: <http://www.comciencia.br/reportagens/energiaeletrica/energia13.htm> . acesso em: 13 mar. 2006.
- [Sol] Soletrol. Energia solar. disponível em: [http://www.soletrol.com.br/educacional/voce\\_sabia.php](http://www.soletrol.com.br/educacional/voce_sabia.php) . acesso em: 15 mai. 2006.
- [Uch99] Elvira Maria Antunes Uchôa. *Framework para Integração de Sistemas de Banco de Dados Heterogêneos*. Doutorado, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 1999.

# Apêndice A

## Nomenclatura

$\mathbf{a, b, c}$  → constantes empíricas que dependem do mês do ano

$A_p$  → área da placa

$C_p$  → calor específico do fluido

$d$  → representa o dia do ano (se  $d=1$  indica o dia 01/01; se  $d=32$  indica o dia 01/02)

$F'$  → fator de eficiência

$F^c$  → função de controle do fluido proveniente do coletor nos nodos do reservatório

$F^L$  → função de controle do fluido proveniente da rede de abastecimento nos nodos do reservatório

$F_R$  → fator de remoção de energia

$G_{sc}$  → Constante Solar incidente no topo da atmosfera terrestre

$h$  → altitude da estação

$h_{p-v}$  → coeficiente de transferência de calor por convecção entre a superfície absorvedora e a cobertura

$h_{r,p-v}$  → coeficiente de transferência de calor por radiação entre a superfície absorvedora e a cobertura

$\bar{H}$  → radiação global diária em média mensal incidente na superfície horizontal

$\bar{H}_c$  → radiação global diária em média mensal

$H_d$  → radiação difusa diária

$\overline{H}_o$  → radiação extraterrestre

$I$  → radiação solar global incidente no plano horizontal

$I_b$  → radiação direta horária na superfície horizontal

$I_d$  → radiação difusa horária

$I_T$  → radiação solar global sobre a superfície inclinada, em média horária

$\overline{K}_T$  → índice de claridade

$\dot{m}$  → vazão mássica do fluido

$\dot{m}_c$  → vazão do coletor

$\dot{m}_L$  → vazão de consumo

$\bar{n}$  → número de horas de insolação diária em média mensal

$\overline{N}$  → número de horas de insolação teórico diária em média mensal

$T_{amb}$  → temperatura do ambiente

$T_{co}$  → temperatura do fluido que chega do coletor

$T_{fi}$  → temperatura do fluido na saída do coletor

$T_p$  → temperatura da placa

$T_{S,i}$  → temperatura no nodo “i”

$R_b$  → razão entre a radiação direta no plano inclinado e horizontal

$S$  → energia absorvida por unidade de área

$\alpha$  → absorvidade do material utilizado

$\delta$  → ângulo de declinação do sol

$\phi$  → latitude da região

$\rho_g$  → reflectância das superfícies próximas que podem ser vistas pela superfície incluída

$\rho$  → reflectividade do material utilizado

$\tau$  → transmissividade do material utilizado

$\omega_s$  → hora angular do pôr-do-sol

$\omega$  → hora angular do sol, definido como ângulo de deslocamento do sol. Considera-se 15° por hora, assumindo manhã negativo e tarde positivo.

## Apêndice B

### Coeficientes empíricos de Bunnett

Mês	a	b	c
Janeiro	0.0225	0.7812	0.0007
Fevereiro	0.221	0.5026	0.0006
Março	0.221	0.5142	0.0005
Abril	0.188	0.5574	0.0005
Maiο	0.197	0.5423	0.0004
Junho	0.235	0.4780	0.0004
Julho	0.264	0.4386	0.0004
Agosto	0.291	0.3768	0.0006
Setembro	0.260	0.4242	0.0006
Outubro	0.235	0.4744	0.0005
Novembro	0.207	0.4816	0.0007
Dezembro	0.237	0.4343	0.0007

# Apêndice C

## Fórmulas da Radiação Solar

$$\delta = 23,45 \sin\left(2\pi \frac{284+d}{365}\right)$$

$$\bar{H}_o = \frac{24 \cdot 3600 G_{sc}}{\pi} \left(1 + 0,033 \cos\left(\frac{2\pi d}{365}\right)\right) (\cos \phi \cos \delta \sin \omega_s + \omega_s \sin \phi \sin \delta)$$

$$\bar{N} = (2/15) \cos^{-1}(-\tan \varphi \tan \delta)$$

$$\bar{H} = \left(a + b \frac{\bar{n}}{\bar{N}} + ch\right) \bar{H}_o$$

$$Kt = \bar{H}/\bar{H}_o$$

$$\bar{H}_d = (0,775 + 0,00606(\omega_s - 90) - [0,505 + 0,00455(\omega_s - 90)] \cos(115Kt - 103)) \bar{H}$$

$$\bar{H}_b = \bar{H} - \bar{H}_d$$

$$a = 0,409 + 0,5016 \sin(\omega_s - 60)$$

$$b = 0,6609 - 0,4767 \sin(\omega_s - 60)$$

$$r_d = \frac{\pi}{24} \frac{\cos \omega - \cos \omega_s}{\sin \omega_s - \frac{\pi \omega_s}{180} \cos \omega_s}$$

$$I_d = r_d H_d$$

$$r_t = \frac{\pi}{24} (a + b \cos \omega) \frac{\cos \omega - \cos \omega_s}{\sin \omega_s - \frac{\pi \omega_s}{180} \cos \omega_s}$$

$$I = H r_t$$

$$I_b = \bar{I} - \bar{I}_d$$

$$\cos \theta_z = \sin \delta \sin \phi + \cos \delta \cos \phi \cos \omega$$

$$\cos \theta = \sin \delta \sin \phi \cos \beta - \sin \delta \cos \phi \sin \beta \cos \gamma + \cos \delta \cos \phi \cos \beta \cos \omega +$$

$$\cos \delta \sin \phi \sin \beta \cos \gamma \cos \omega + \cos \delta \sin \beta \sin \gamma \sin \omega$$

$$R_b = \frac{\cos \theta}{\cos \theta_z}$$

$$I_T = I_b R_b + I_b \left( \frac{1 + \cos \beta}{2} \right) + I_{\rho g} \left( \frac{1 - \cos \beta}{2} \right)$$

# Apêndice D

## Fórmulas do Coletor Solar

Cálculos das componentes solares incidentes na superfície do coletor solar.

$$\delta = \arccos \left( \frac{\sin \delta (\sin \phi \cos \beta - \cos \phi \sin \beta \cos \gamma) + \cos \delta \cos \omega (\cos \phi \cos \beta + \sin \phi \sin \beta \cos \gamma) + \cos \delta \sin \beta \sin \gamma \sin \omega}{\cos \delta \sin \beta \sin \gamma \sin \omega} \right)$$

$$\theta_d = (59,68 - 0,1388\beta + 0,001497\beta^2)\pi/180$$

$$\theta_g = (90 - 0,5788\beta + 0,002693\beta^2)\pi/180$$

$$\theta_{2b} = \arcsin \left[ \frac{\sin \theta_b}{n} \right]$$

$$\theta_{2d} = \arcsin \left[ \frac{\sin \theta_d}{n} \right]$$

$$\theta_{2g} = \arcsin \left[ \frac{\sin \theta_g}{n} \right]$$

$$\tau_{ab} = \exp \left( \frac{-K_v L_v}{\cos \theta_{2b}} \right)$$

$$\tau_{ad} = \exp \left( \frac{-K_v L_v}{\cos \theta_{2d}} \right)$$

$$\tau_{ag} = \exp \left( \frac{-K_v L_v}{\cos \theta_{2g}} \right)$$

$$r_b \perp = \frac{\sin^2(\theta_{2b} - \theta_b)}{\sin^2(\theta_{2b} + \theta_b)}$$

$$r_d \perp = \frac{\sin^2(\theta_{2d} - \theta_d)}{\sin^2(\theta_{2d} + \theta_d)}$$

$$r_g \perp = \frac{\sin^2(\theta_{2g} - \theta_g)}{\sin^2(\theta_{2g} + \theta_g)}$$

$$r_b \parallel = \frac{\tan^2(\theta_{2b} - \theta_b)}{\tan^2(\theta_{2b} + \theta_b)}$$

$$r_d \parallel = \frac{\tan^2(\theta_{2d} - \theta_d)}{\tan^2(\theta_{2d} + \theta_d)}$$

$$r_g \parallel = \frac{\tan^2(\theta_{2g} - \theta_g)}{\tan^2(\theta_{2g} + \theta_g)}$$

$$\tau_{rb} = 0,5 \left[ \frac{1-r_b \parallel}{1+r_b \parallel} + \frac{1-r_d \perp}{1+r_d \perp} \right]$$

$$\tau_{rd} = 0,5 \left[ \frac{1-r_d \parallel}{1+r_d \parallel} + \frac{1-r_b \perp}{1+r_b \perp} \right]$$

$$\tau_{rg} = 0,5 \left[ \frac{1-r_g \parallel}{1+r_g \parallel} + \frac{1-r_g \perp}{1+r_g \perp} \right]$$

$$\tau_{vb} = \tau_{ab} \tau_{rb}$$

$$\tau_{vd} = \tau_{ad} \tau_{rd}$$

$$\tau_{vg} = \tau_{ag} \tau_{rg}$$

$$\rho_{vd} = \tau_a (1 - \tau_r)$$

$$\rho_{pd} = 1 - \alpha_p$$

$$I_{bt} = I_b R_b$$

$$I_{dt} = I_d \left( \frac{1 + \cos \beta}{2} \right)$$

$$I_{gt} = \rho (I_b + I_d) \left( \frac{1 - \cos \beta}{2} \right)$$

Então é calculado  $S$

$$S = \alpha_p [\tau_{vd} I_{bt} + \tau_{vd} I_{dt}] [1 + \rho_{vd} \rho_{pd}] + I_{gt} \alpha_p \tau_{vg}$$

**Cálculos da eficiência do coletor solar.**

$h_w = 2,8 + V_{vento} \rightarrow$  Coeficiente convectivo entre a cobertura e o ambiente

$W =$  largura da placa/número de tubos  $\rightarrow$  distância entre dois tubos

$L = ((W - D_{ext})/2) \rightarrow$  distância média entre os tubos

$Rb = \frac{\ln\left(\frac{D_{ext} + 2E_{sp}placa}{D_{ext}}\right)}{2\pi(FC)L_p K_{placa}} \rightarrow$  FC é o fator de contato,  $L_p$  é o comprimento da placa,  $K_{placa}$  é a condutividade da placa

$T_{amb} = T_{max} - \frac{T_{max} - T_{min}}{2} + \left( \left( \frac{T_{max} - T_{min}}{2} \right) \cos \left( (\omega_s - 30) \frac{\pi}{180} \right) \right) \rightarrow$  temperatura do ambiente

$\alpha_{ar} = 0,0000001465t + 0,0000182 \rightarrow$  "t" é a temperatura média da placa com a cobertura em °C.

$\mu_{ar} = 0,0000004614286t + 0,000017 \rightarrow$  "t" é a temperatura média da placa com a cobertura em °C.

$\rho_{ar} = -0,003261t + 1,270643 \rightarrow$  "t" é a temperatura média da placa com a cobertura em °C.

$Visc_c = \frac{\mu_{ar}}{\rho_{ar}} \rightarrow$  viscosidade cinemática

$Ra = \frac{9,84 \left( \frac{1}{\left( \frac{t_p + t_c}{2} \right)} \right) (t_p - t_c) \delta_{p-v}^3}{\alpha_{ar} Visc_c} \rightarrow$  onde  $\delta_{p-v}$  é a distância entre a placa e a cobertura.

$$Nu = 1 + 1,44 \left[ 1 - \frac{1708}{Ra \cos \beta} \right]^+ \left( 1 - \frac{(\sin(1,8\beta))^{1,6}}{Ra \cos \beta} \right) \left[ \left( \frac{Ra \cos \beta}{5830} \right)^{1/3} - 1 \right]$$

$K_{ar} = 0,0000758928t + 0,024175 \rightarrow$  "t" é a temperatura média da placa com a cobertura em °C.

$$h_{p-v} = \frac{Nu K_{ar}}{\delta_{p-v}}$$

$x = \frac{a}{\delta_{p-v}} \rightarrow$  "a" é a largura da placa

$y = \frac{L_p}{\delta_{p-v}} \rightarrow L_p$  é o comprimento da placa

$$F_{p-v} = \frac{2}{\pi xy} \left( \begin{array}{l} \ln \left[ \frac{(1+x^2)(1+y^2)}{1+x^2+y^2} \right] + \\ x\sqrt{1+y^2} \operatorname{atan} \left( \frac{x}{\sqrt{1+y^2}} \right) + \\ y\sqrt{1+x^2} \operatorname{atan} \left( \frac{y}{\sqrt{1+x^2}} \right) - \\ -x \operatorname{atan}(x) - y \operatorname{atan}(y) \end{array} \right)$$

$h_{r,p-v} = \frac{\sigma(T_p+T_v)(T_p^2+T_v^2)}{\frac{1-\epsilon_p}{\epsilon_p} + \frac{4}{1+3F_{p-v}} + \frac{1-\epsilon_v}{\epsilon_v}}$   $\rightarrow \epsilon_p$  é a emissividade da placa e  $\epsilon_v$  é a emissividade do vidro

$P_{V\text{Sat}} = 0,0000006t^4 + 0,00009t^3 + 0,0019t^2 + 0,0395t + 0,6317 \rightarrow$  "t" é a temperatura ambiente em °C.

$$P1 = U * R * P_{V\text{Sat}}$$

$$T_{dp} = T_{V\text{Sat}} = -0,0195P1^4 + 0,429P1^3 - 3,7344P1^2 + 18,488P1 - 7,7101$$

$$T_{sky} = T_{amb} \left[ 0,8 + \frac{T_{dp}}{250} \right]^{1/4}$$

$$h_{r,v-amb} = \epsilon_v \frac{\sigma(T_v+T_{sky})(T_v^2+T_{sky}^2)(T_v-T_{sky})}{T_v-T_{amb}}$$

$$U_{topo} = \frac{1}{\frac{1}{h_{p-v}+h_{r,p-v}} + \frac{1}{h_w+h_{r,v-amb}}}$$

$$U_{base} = \frac{K_{isol}}{\delta_{isol}}$$

$$U_{lateral} = \frac{K_{isol}}{\delta_{isol}} \frac{A_{lat}}{A_p}$$

$$UL = U_{topo} + U_{base} + U_{lat}$$

$$m = \sqrt{\frac{UL}{K_{placa}Esp_{placa}}}$$

$F = \frac{\tan(mL)}{mL} \rightarrow L$  é a distância entre os tubos

$\mu_f = 0,000000203357t^2 - 0,00002946t + 0,00150652 \rightarrow$  quando a água estiver com menos de 60°C.

$\mu_f = 0,0000000497142t^2 - 0,0000125303t + 0,00103955 \rightarrow$  quando a água estiver entre 61 e 100°C

$$\rho_{agua} = -0,0030877t^2 - 0,1323t + 1002,313$$

$$K_{agua} = 0,000010017t^2 + 0,0023299t + 0,55196$$

$$\alpha_{agua} = 0,00000000039t + 0,000000134$$

$$Visc_{c-agua} = \frac{\mu_{agua}}{\rho_{agua}}$$

$$Vel_{fluido} = \frac{4\dot{m}}{N_{tubos}\pi D_{int}^2}$$

$$Re = \frac{Vel_{fluido}D_{int}}{Visc_{c-agua}}$$

$$Pr_f = \frac{Visc_{c-agua}}{\alpha_{agua}}$$

$$G_z = \frac{Re+Pr_f}{\left(\frac{L_p}{D_{int}}\right)}$$

$$nu = 3,66 + \frac{0,0668G_z}{1+0,04G_z^{2/3}}$$

$$h_f = \frac{nu*K_f}{D_{int}}$$

$$F_{coletor} = \frac{\frac{1}{UL}}{W \left[ \frac{1}{UL(D_{ext} + (W - D_{ext})^F)} + Rb + \frac{1}{\pi D_{int} h_f} \right]} \rightarrow W \text{ é a distância entre dois tubos.}$$

$$\dot{m} = \text{vazão} * \rho$$

$$C_p = 0,0081997t^2 - 0,5305t + 4187, t \rightarrow \text{“t” em } ^\circ\text{C}$$

$$Fr = \frac{\dot{m}C_p}{A_{placa}UL} \left[ 1 - \exp\left(-\frac{A_{placa}ULF_{coletor}}{\dot{m}C_p}\right) \right]$$

$$Q_{util} = (A_{plca}Fr(S - UL(T_{fi} - T_{amb}) * 3600/1000000)) * 1000000/3600$$

$$T_{fs} = T_{fi} + \frac{Q_{util}}{\dot{m}C_p}$$

$$T_p = T_{fi} + \frac{Q_{util}}{A_{placa}UL*Fr}(1 - Fr)$$

$$T_v = T_{placa} - \frac{U_{topo}(T_{placa} - T_{amb})}{h_{p-v} + h_{r,p-v}}$$

Todo o process acima ser repete até a diferença da temperatura inicial e final, da placa ou do viro, forem menor que 0,01.

$$Eff = \frac{Q_{util}}{(I_{bt} + I_{dt} + I_{gt})A_{placa} * 1000000/36000}$$

# Apêndice E

## Documentação do *SolarEnergy*

A documentação apresentada teve como base a fornecida pelo RUP (Rational Unified Process).

### E.1 Especificação de Requisitos do Framework

#### E.1.1 Propósito

O objetivo deste documento é definir os requisitos para o framework (daqui por diante, referido como SolarEnergy) e detalhar as especificações para as características, capacidades, atributos críticos, e características principais do framework proposto. Este documento deve ser lido com o objetivo de avaliar os benefícios e praticabilidade da aplicação proposta assim como fornecer uma base para a estimativa do tempo e de esforços necessários para construir, testar, implantar, e manter a aplicação. Este documento não descreve como, quando, ou onde qualquer destas atividades serão executadas ou quem irá fazê-las.

#### E.1.2 Escopo

O framework para o desenvolvimento de sistemas de energia solar será responsável pelo projeto (construção) de um sistema termossifão de maneira visual, onde o usuário poderá escolher o componente que participa do termossifão e conecta-los, informando a origem e o destino dos dados. Fornecer mecanismos que possibilitem sua expansão de modo a incorporar novos componentes e funcionalidades, além de fornecer uma base de dados climatológicos e permitir sua atualização.

#### E.1.3 Contexto do Sistema

O framework terá com principais usuários cientistas da área de energia solar, entretanto deverá ter a capacidade de ser manuseado por pessoas com baixo ou nenhum conhecimento técnico de energia solar.

**Tabela E.1: Stakeholders**

Função/Responsabilidade	Stakeholder Principal	Stakeholder Secundário
Dono	Elizabeth Pereira	Elizabeth Pereira
Colaboradores	Elizabeth Pereira	Alexandre Salomão; Daniele; Estagiários

### E.1.4 Principais Stakeholders

### E.1.5 Referências

- O Processo Unificado de Desenvolvimento de Software, Ivar Jacobson, Grady Booch, e James Rumbaugh
- Metodologia de Arquitetura SunTone , Sun Microsystems, Inc.,2002 [http://www.sun.com/service/sunps/jdc/suntoneam\\_wp5.24.pdf](http://www.sun.com/service/sunps/jdc/suntoneam_wp5.24.pdf) Relatório Fapemig fornecidos pelo GRE
- Java application frameworks, Govoni, Darren, 1999

### E.1.6 Restrições e Suposições

#### Processo de Desenvolvimento e Restrições de Equipe

Os componentes do framework serão desenvolvidos em etapas. Para iniciar uma próxima etapa a anterior terá de ser finalizada. A equipe de desenvolvimento se restringe a um aluno de mestrado e outro de graduação

#### Restrições Ambientais e Tecnológicas

O sistema final deverá ter manutenções evolutivas feitas por funcionários do GREEN. Para não exigir mão de obra especializada em uma determinada linguagem e para aproveitar o recurso de estagiários fornecidos pelo PUC, o framework foi desenvolvido em Java (linguagem lecionada na instituição).

Java <http://java.sun.com/>

Para obter maior produtividade no desenvolvimento de telas e confiabilidade utilizou-se o ambiente NetBeans. O SGBD será o MySQL, um servidor de gerenciamento de banco de dados relacional de livre.

#### Restrições de Entrega e Implantação

Deverá ser entregue um sistema completo ao final do projeto do CNPq.  
Minimização de Riscos

#### Riscos Tecnológicos

Não há riscos tecnológicos. O desenvolvedor tem que apenas saber desenvolver em Java.

### Risco de Recursos e Habilidades

As fórmulas a serem implementadas possuem alta complexidade e poucas pessoas com domínio no GREEN, sendo praticamente a professora Elizabeth Pereira a detentora do conhecimento. Risco altíssimo no cumprimento dos prazos, uma vez que ela é uma pessoa ocupadíssima.

### Risco de Requisitos

Não há riscos de requisitos para este projeto.

### Riscos Políticos

Não há riscos políticos para este projeto.

## E.1.7 Requisitos Funcionais

Características Essenciais:

- Manipular visualmente os componentes que formam um termosifão.
- Atualizar dados climatológicos.
- Incorporar novos componentes e funcionalidades.
- Realizar simulações, com base na radiação solar, dos componentes plotados pelo usuário.

### Casos de Uso

Após conversar com o dono do sistemas, chegou-se nas seguintes funcionalidades do framework.

**Tabela E.2:** Stakeholders

Nome do Caso de Uso	Prioridade	Número
Desenhar Componente	E	1
Cadastrar Dados Climatológicos	E	2
Conectar Componentes	E	3
Simular	E	4
Adicionar Novo Componente	E	5
Adicionar Funcionalidade	E	6
Alterar Radiação Solar	A	7

### Requisitos do Framework

Para cada caso de uso, segue os requisitos funcionais.

Código do Req.	Descrição do Requisito
E1-1	Adiciona um componente na área de plotagem.
E1-2	Manipular o componente: arrastá-lo para qualquer lado.
E1-3	Permitir selecionar vários componentes e manipulá-los igualmente.
E1-4	Excluir um componente adicionado na área de plotagem.
E1-5	Permitir que um mesmo componente seja desenhado várias vezes.
E2-1	Alterar, incluir ou excluir qualquer dado climatológico de uma cidade
E3-1	Realizar uma ligação entre dois componentes, informando o destino dos dados por uma ceta.
E3-2	Permitir que o usuário escolha os vínculos entre os dados (dados de origem e destino). Caso nenhum vínculo seja escolhido, o framework os vinculará pelo nome.
E3-3	Fornecer o acesso ao banco de dados.
E4-1	Os componentes adicionados na área de plotagem e a radiação solar são executados. A execução dos componentes deve obedecer a ordem das ligações feitas pelo usuário.
E4-2	Validação. Verificar se há componente que não será executado.
E4-3	Validação. Verificar se todo dado que necessita receber valores de outro componente para o seu funcionamento a terá.
E4-4	Para cada dado produzido pela radiação solar, todos os componentes deverão ser executados.
E4-5	Transmitir os valores entre os componentes e da radiação solar.
E5-1	Fornecer interfaces para que novos componentes sejam manipulados pelo framework
E5-2	Fornecer a implementação da manipulação dos componentes.
E6-1	Disponibilizar um conjunto de interfaces (telas) para que sejam extendidas e incorporadas no framework
E7-1	Disponibilizar uma implementação básica da radiação solar.
E7-2	Permitir que o componente da radiação disponibilizado seja extendido e alterado no framework.

## Diagramas de Classes do Framework

A Figura E.1 ilustra as classes que compõem a parte funcional do framework.

## E.2 Funcionamento Interno do *SolarEnergy*

Instanciando o *SolarEnergy*, através da classe *Energy*, é possível alterar a tela principal, só então é possível adicionar componente para serem utilizados pelo método *addComponent()* ou alterar a radiação solar com o método *setSolarRadiation()*. É na instanciação que a conexão com o banco de dados e o controlador das simulações são criados.

O dispõe de algumas operações básicas como salvar, abrir e imprimir o sistema de termo-sifão, plotado no painel de simulações.

Para construir um sistema de termo-sifão, basta clicar com o mouse no componente desejado, localizado na guia de componentes. Neste momento, o painel de simulações (*EMCPainel*) é informado pelo o *EMCObject* o componente que o usuário deseja adicionar. Nesta operação, a classe *EMCPainel* espera receber um *IObject*. Então basta que o usuário

clique no painel de simulações para que o componente seja adicionado. Ao adicionar um componente, o componente é clonado, e é este componente clonado que é adicionado.

Para realizar uma ligação entre dois componentes, é necessário informar ao painel de simulações (*EMCPainel*) que o usuário deseja desenhar uma linha (no caso do termo-sifão, o cano). O botão com desenho de um cano é quem transmite esta informação ao *EMCPainel*. O painel de simulações verifica se o clique do mouse foi em cima de um ponto de conexão (*EMCPoin*), se sim começa a desenhar uma linha, caso contrário não faz nada. Após localizado um ponto de conexão, cada clique que o usuário realiza com o mouse, um vértice é adicionado em *EMCPipeVertice*, com isto é possível desenhar arestas entre os componentes. O *SolarEnergy* termina de desenhar a linha quando o usuário clica em outro ponto de conexão, que não seja o mesmo que a origem.

Como ilustrado no exemplo acima, nenhuma classe contendo regras de negócio dos componentes é utilizada, mas sim suas classes básicas de manipulação de imagens, localizadas na primeira camada (vide Capítulo 3.3).

O *SolarEnergy* foi desenvolvido para realizar um número definido de ciclos por dado da radiação. A quantidade de ciclos padrão é quatro, podendo se alterada na tela de propriedades.

Formado um sistema termo-sifão é possível simulá-lo pressionando o botão *run*, representado por uma seta verde. Esta ação repassa os componentes e suas ligações para o *Controler*, através da classe *Energy* com o método *exec()*, para que a simulação tenha início. A partir desse momento que as atividades descritas na Sessão 3.6 são executadas.

Cada componente realiza seu trabalho independente de quem ele se conecta ou é conectado, funcionando como uma caixa preta. O *SoalrEnergy* não realiza nenhuma checagem dos tipos de componentes que se conectam, sendo que a única restrição de conexão são as variáveis que os componentes necessitam receber para o seu funcionamento. Se uma bomba é desenvolvida, e conectada após o coletor solar, quando deveria ser após o reservatório, por exemplo, a simulação ocorrerá de qualquer forma, desde que não haja nenhum restrição das variáveis recebidas pelos mesmos. Entretanto, poderá produzir resultados incoerentes.

### E.3 O Pacote *Controler*

Algumas considerações técnicas quanto ao pacote *Controler*.

O recurso de threads, para a execução dos componentes em paralelo, foi descartado após observado vários modelos de captura e armazenamento de energia solar - foi observado uma linearidade. Com isto, além de não se ter um ganho expressivo com a utilização do paralelismo (os modelos, em sua maioria, são pequenos), poder-se-ia até mesmo ocorrer perda de eficiência ao se instanciar threads desnecessariamente. Os componentes são executados de forma serial sem perda de dependências.

Cada nodo da árvore de execução representa um componente físico e as arestas do sistema termo-sifão. Os componentes serão executados como se estivesse em um caminho em largura para não haver a necessidade de re-execução do nodo, pois um componente pode fornecer dados para mais de um componente.

Toda simulação terá, como nodo de partida, o componente de captura. Mesmo que o

sistema termo-sifão possua mais de um coletor, o nodo raiz terá apenas um componente de captura. Quando dois ou mais componentes de captura são conectados em série, o *SolarEnergy* identifica o componente de captura independente dos demais, ou seja, que não recebe dados de nenhum outro componente de captura para adicioná-lo na raiz da árvore de execução.

Em muitos modelos termo-sifão, coletores são conectados em paralelo. Isto não é possível ser feito diretamente no *SolarEnergy*, pois não há uma padronização nos nomes das variáveis, para que o *framework* identifique a variável terá seu valor dividido entre os componentes conectados em paralelo. O componente também deveria informar ao *framework* o tipo da ligação: série ou paralelo, uma vez que os componentes são executados como uma “caixa-preta” para o *framework*.

O grafo de execução é criado da seguinte forma. A classe *Tree* recebe dois vetores, um contendo os componentes do modelo e outro contendo suas conexões, e uma referência para a radiação solar. Cada componente e conexão é adicionado em um nodo. Em seguida os nodos são conectados de acordo com as ligações originais dos componentes. Por fim, o nodo de captura é adicionado na raiz. Um exemplo de grafo é ilustrado na Figura 3.11. O grafo formado traz benefícios como uma melhor organização da ordem de execução e do processo que percorre os componentes.

O método *exec()* da classe *Execute*, além de executar o grafo, se encarrega de realizar validações no modelo. O processo pode ser visto em detalhes na Figura 3.12.

A classe *Nodo* do pacote *Controler* (Figura 3.10) é quem efetivamente executa um componente. Cabe a ela identificar se deve invocar o método *calculate()* de um componente ou o *transfer()* das conexões.

## E.4 O Pacote *Variables*

Na passagem de valores entre os componentes, há a necessidade de ter uma forma única de localizar as variáveis a serem transferidas. Para que o *SolarEnergy* conheça as variáveis, foi projetada a classe *Variables*.

Quando o *SolarEnergy* executa o componente radiação solar, este retorna um conjunto de valores que podem representar a radiação em meses, semanas, horas ou até minutos, ditando a quantidade de ciclos de uma simulação. Logo *Variables* também deve atender a esta necessidade. A Figura E.2 apresenta o projeto da estrutura de armazenamento da variáveis.

As variáveis são criadas pelo método *addVariabe()*, onde são armazenados em um vetor (*variables*) de *Records*. Com isto os dados podem ter a granularidade que o desenvolvedor desejar.

A classe *Record* é quem realmente armazena o valor da variável. Para armazenar vários valores sejam atribuídos a uma variável, a classe *Record* dispõe de um vetor de *DataRadiation*.

Toda variável, atribuída à *Variables*, é referenciada pelo *SolarEnergy* através do seu nome. O nome de uma variável tem o mesmo princípio de uma “chave primária” de uma tabela de banco de dados.

O método *setNeeded()* informa a necessidade do respectivo componente receber uma

variável, de modo a não comprometer sua execução. Caso o *SolarEnergy* identifique que um componente não recebe alguma variável rotulada como necessária, a simulação é abortada e nenhum cálculo será efetuado. Toda variável, por padrão, é necessária.

O método *setObject* associa um objeto “genérico” à classe *Record*, fornecendo maior flexibilidade para o desenvolvedor. Assim é possível transmitir qualquer valor e/ou classe para um componente.

A forma de se criar uma variável que conterà apenas um único valor e da que conterà vários valores é exatamente a mesma. Para isto, invocando-se o método *addVariable()*. A diferença está na ação de incluir e obter dos valores da variável. Em uma variável com apenas um único valor, isto se dá pelo método *setValue()* e *getValue()*. Havendo vários valores associados, o método a ser invocado é o *addValueOfRadiation()* e o *getValueOfRadiation()*.

Os dados da radiação são adicionados de forma consecutiva (ao final do vetor), não havendo nenhum tratamento temporal, ou seja, a ordem em que eles são adicionados é a mesma ordem que serão executados pelo *SolarEnergy*.

## E.5 O Pacote *Forms*

Tem a finalidade de agrupar as telas a serem exibidas pelo *SolarEnergy*.

*SolarEnergy* disponibiliza classes e interfaces (Figura E.3) para o desenvolvimento de novas telas ou mesmo para a substituição da tela principal, pois a necessidade do pesquisador pode não ser apenas de facilidade de implementação ou autonomia na construção de um componente, mas sim a necessidade de disponibilizar interfaces gráficas para que o usuário projete um componente (dado um conjunto de parâmetros).

Para desenvolver uma nova tela principal do *SolarEnergy* é necessário que seja herdada a classe *FrmBasicMain* para que métodos básicos contido em *IBasicMain*, como o *run()*, sejam invocados corretamente pelo *SolarEnergy*. Para desenvolver qualquer outra tela basta herdar de *FrmBasic*.

## E.6 O Pacote *Database*

Como o *SolarEnergy* permite atualização e incorporação de novos dados climatológicos, tornou-se útil a incorporação de um banco de dados ao projeto. Desta forma não faz parte da regra de negócio do *SolarEnergy* garantir a persistência dos dados.

De modo a facilitar o desenvolvimento, foi incorporado ao pacote a classe *GQuery* para prover as atualizações (inserções, alterações e exclusões) no banco de dados, como observado na Figura E.4.

## E.7 O Pacote *Util*

Para que testes comprovassem a utilização do *SolarEnergy*, o mesmo foi implementado na linguagem Java. Na sua codificação houve a necessário de extensão de algumas classes específicas do Java como a *EMCPanel*, para desenhar os componentes. O projeto do

*framework SolarEnergy* implementado em outra linguagem de programação, pode não haver a necessidades destas classes.

A classe *EMCPanel* estendida de *JPanel* é a responsável por desenhar e manipular visualmente os componentes no *SolarEnergy*. Também implementa o salvamento dos componentes adicionados. Para abrir um sistema termossifão já salvo, utiliza a classe *EMCFileFilter*.

Todo novo componente adicionado ao *SolarEnergy* reside em um botão (*EMCButton*) o qual informa ao *EMCPanel* o próximo componente a ser desenhado.

Para mapear os dados de origem e destino na transferência de valores em uma simulação, o *SolarEnergy* utiliza a classe *EMCPositionLine* e para desenhar as ligações na tela a classe *EMCLinkPanel*

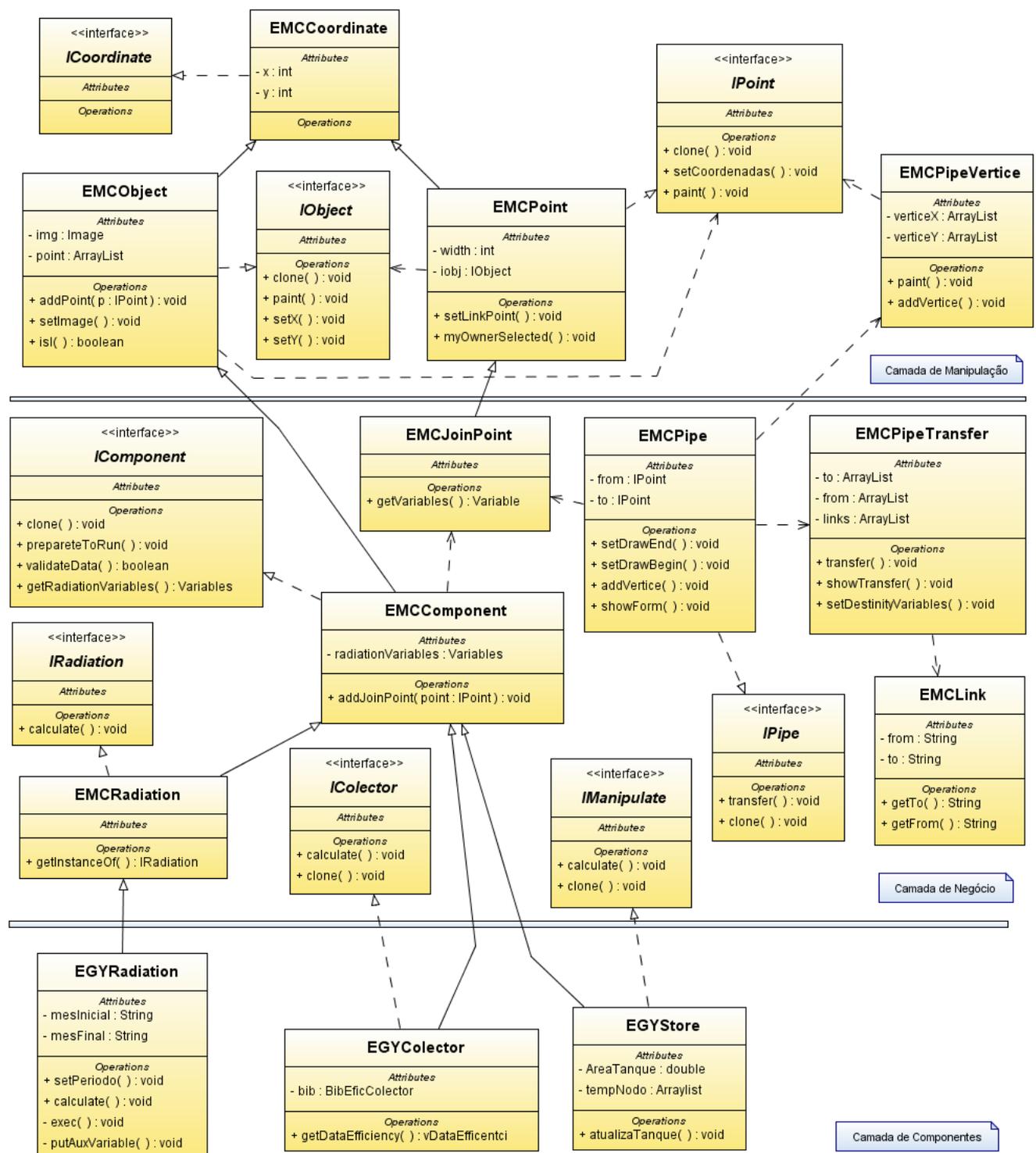


Figura E.1: Diagrama de classes

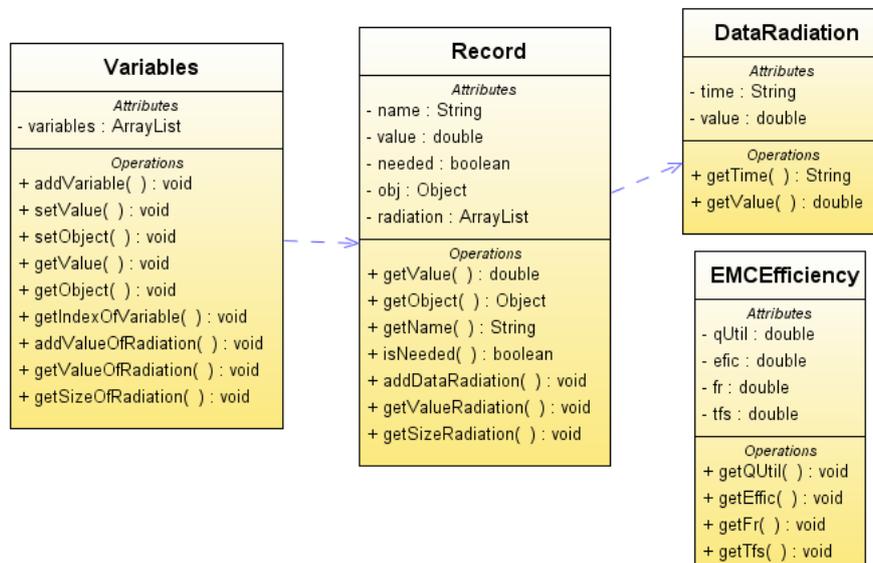


Figura E.2: Classes do pacote *Variables*.

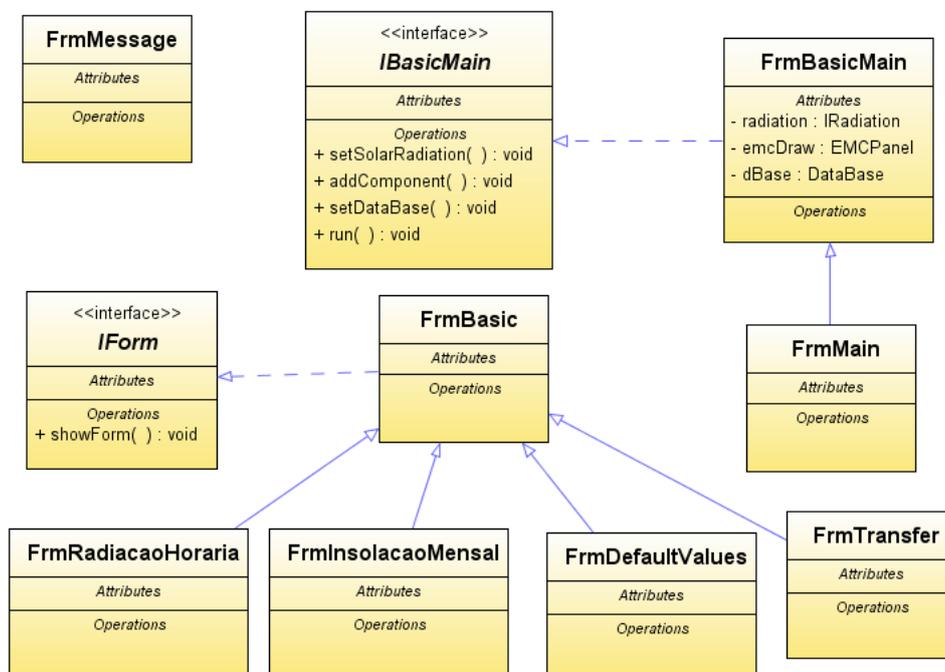


Figura E.3: Classes do pacote *Forms*.

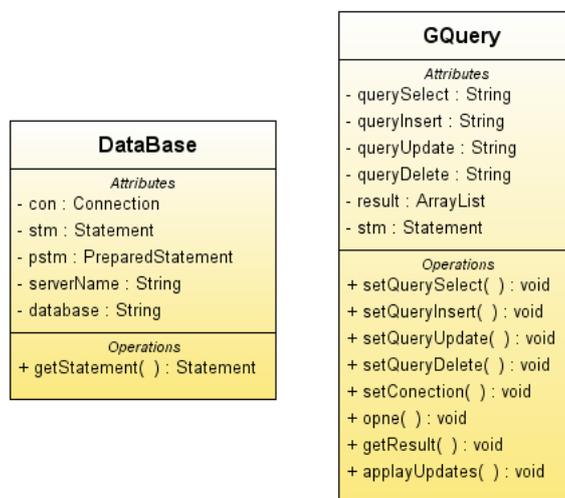


Figura E.4: Classes do pacote *Database*.

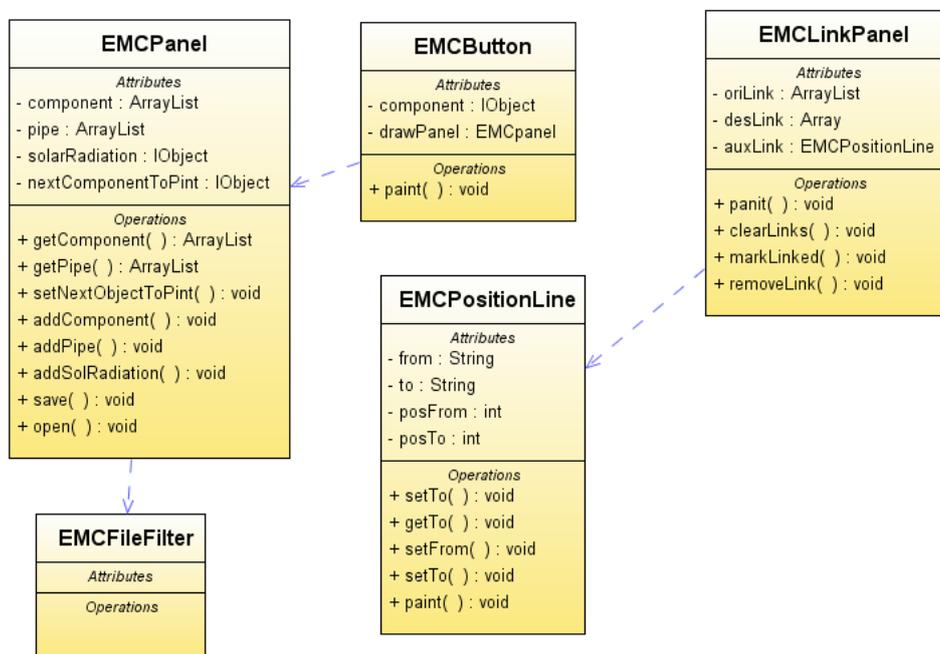


Figura E.5: Classes do pacote *Util*.

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)