

Frederico Papatella Guerino

Framework Solar Energy: Evoluções
Aplicadas a Sistemas Fotovoltaicos

Dissertação 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

Novembro de 2008

Livros Grátis

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

Milhares de livros grátis para download.

FICHA CATALOGRÁFICA

Elaborada pela Biblioteca da Pontifícia Universidade Católica de Minas Gerais

G932f	<p>Guerino, Frederico Papatella Framework solar energy: evoluções aplicadas a sistemas fotovoltaicos. / Frederico Papatella Guerino. – Belo Horizonte, 2008. 104f. : il.</p> <p>Orientador: Mark Alan Junho Song. Dissertação (Mestrado) – Pontifícia Universidade Católica de Minas Gerais, Programa de Pós-graduação em Informática. Bibliografia.</p> <p>1.Framework (Programa de computador) – Teses. 2. Energia solar – Programas de computador. I. Song, Mark Alan Junho. II. Pontifícia Universidade Católica de Minas Gerais. III. Título</p> <p style="text-align: right;">CDU: 681.3.06</p>
-------	---

Bibliotecário: Fernando A. Dias – CRB6/1084



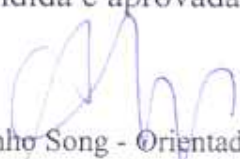
PUC Minas
Programa de Pós-graduação em Informática

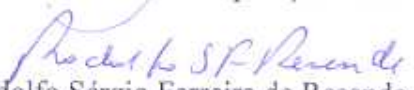
FOLHA DE APROVAÇÃO

"Framework Solar Energy: Evoluções Aplicadas a Sistemas Fotovoltaicos"

Frederico Papatella Guerino

Dissertação defendida e aprovada pela seguinte banca examinadora:


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


Prof. Rodolfo Sérgio Ferreira de Resende (UFMG-MG)
Doutor em Ciência da Computação - UCSB


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


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

Belo Horizonte, 28 de novembro de 2008.

À minha querida esposa, Érica
Aos meus pais, Antônio e Márcia
Às minhas irmãs, Sil e Ana
Aos meus cunhados, Rodrigo e Wanderson
Ao meu afilhado, Thiago
À todos os meus amigos e familiares

Agradecimentos

A Deus, fonte de sabedoria e luz.

À minha maravilhosa esposa, por compreender os momentos da minha ausência, por sempre me incentivar, pelo incondicional apoio e constante dedicação em me auxiliar nesta difícil jornada.

Aos meus queridos pais, exemplos de vida e superação.

Às minhas irmãs e aos meus cunhados pelos conselhos e torcida.

À D. Célia, pelas constantes orações e incentivo.

Ao meu orientador, Prof. Mark Alan Junho Song, por sempre indicar as decisões corretas a serem tomadas nos momentos de dificuldade.

Aos professores Elizabeth Pereira, Lauro Machado e Luiz Zarate pela amizade, apoio e aconselhamentos.

Ao GREEN (Grupo de Estudos em Energia Solar), na PUC-Minas, pelo auxílio, conhecimento e disponibilidade de equipamentos, especialmente, ao amigo Thiago Daniel Moura pela dedicação, disponibilidade, generosidade e empenho em me auxiliar durante todas as fases do projeto.

Aos alunos da iniciação científica, Igor Perdigão e Tiago Carvalho pelo auxílio no desenvolvimento das ferramentas.

À Cláudia Cabral, pela cordial prestação no desenvolvimento deste projeto.

Ao meu grande amigo Gustavo Guidoni por sua enorme colaboração, auxílio, por suas preciosas informações, dicas e pela sua grande disposição em colaborar com o sucesso desta empreitada.

Aos colegas de trabalho e diretoria do Datapuc, pelo incentivo e apoio.

Aos meus amigos e familiares e à todos aqueles, que, de alguma forma, contribuíram para a realização deste trabalho.

Resumo

O Brasil é um dos maiores consumidores de energia da América do Sul, entretanto, previsões pessimistas estão sendo feitas sobre a produção de energia através de combustíveis não renováveis. Neste cenário, a utilização de fontes de energia renováveis, tais como a energia solar, deve ser investigada. De fato, novas abordagens e pesquisas devem ser desenvolvidas a fim de ajudar a tomar decisões de forma adequada. A maioria dos *softwares* aplicados ao domínio da energia solar são complexos e difíceis de serem usados - não podem ser facilmente adaptados para tratar os dados relacionados a uma região específica. A Engenharia de *Software* pode resolver o problema, uma vez que abrange a aplicação dos princípios de engenharia para a construção de *softwares* mais flexíveis e adaptativos. O *framework* é um exemplo, pois simplifica o desenvolvimento de novas aplicações através do reaproveitamento de código fonte ou de parte do projeto. Esta técnica permite melhorias na produtividade, na qualidade e na manutenção dos *softwares*. Este trabalho apresenta uma evolução do *framework* orientado a objetos, chamado *Solar Energy*, inicialmente desenvolvido para controlar e simular processos de aquecimento de água por meio da energia solar, aplicando-o às especificações da energia solar fotovoltaica.

Abstract

Brazil is one of the largest energy consumer in South America, however, pessimistic forecasts are being made about the production of energy through non-renewable fuels. In this scenery, the use of renewable energy sources, such as the solar energy, must be investigated. In fact, research and new approaches must be developed to help make decisions in the proper way. Most softwares applied to the solar energy domain are complex and difficult to use - it cannot be easily adapted to handle data related to a particular region. The discipline of Software Engineering can be used to solve the problem, once it covers the application of engineering principles to build software more flexible and adaptive. The framework is an example. It simplifies the development of new applications by the reusing source code or any part of the project. This technique allows improvements in the productivity, quality and maintenance of the software. This dissertation presents the effort to evolve the object-oriented framework, named Solar Energy, originally developed to control and simulate processes of heating water using solar energy, in order to be applied to the photovoltaic processes.

Conteúdo

Lista de Figuras	v
Lista de Tabelas	vii
1 Introdução	1
1.1 Motivação	3
1.2 Objetivos	4
1.3 Principais contribuições	5
1.4 Estrutura da dissertação	5
2 Estado da Arte: <i>Frameworks</i>	6
2.1 Considerações iniciais	6
2.2 Conceitos básicos	6
2.2.1 Reuso de <i>software</i>	6
2.2.2 <i>Frameworks</i> orientados a objetos	7
2.2.3 <i>Frameworks</i> Caixa Branca, Caixa Preta e Caixa Cinza	8
2.3 Aplicações	9
2.4 A construção de <i>frameworks</i>	10
2.5 A instanciação de <i>frameworks</i>	12
2.6 Considerações finais	13
3 Estado da Arte: Geração Fotovoltaica	15
3.1 Considerações iniciais	15
3.2 Introdução	15
3.3 Ano padrão	16
3.3.1 Método de Festa e Ratto	17

3.3.2	Extensão no Método de Festa e Ratto	21
3.4	Radiação Solar	21
3.4.1	Métodos de Estimativa da Radiação Solar	22
3.5	Painel Fotovoltaico	29
3.6	Bateria	32
3.7	Controlador de Carga	33
3.8	Inversor	35
3.9	Comentários finais	35
4	<i>Framework Solar Energy</i>	36
4.1	Considerações iniciais	36
4.2	Requisitos de aplicativos fotovoltaicos	38
4.3	Arquitetura	39
4.3.1	Estrutura inicial	40
4.3.2	Evoluções aplicadas	42
4.4	Projeto	52
4.5	Implementação	54
4.6	Validação	54
4.7	Considerações Finais	55
5	Estudo de Caso: Evolução do Sol Brasil	56
5.1	Introdução	56
5.2	Requisitos	56
5.3	Solução proposta	57
5.3.1	Diagrama de execução	60
5.3.2	Definição da demanda	63
5.3.3	Execução	65
5.4	Resultados obtidos	66
5.5	Considerações Finais	73
6	Conclusões	74
6.1	Contribuições	75
6.2	Trabalhos Futuros	76

Bibliografia	77
Apêndice	81
A Fórmulas do cálculo da distância para identificação do ano padrão	82
B Banco de dados das matrizes de transição	83
C Fórmulas da Radiação Solar	87
D Coeficientes empíricos de Bunnett	89
E Dimensionamento Estocástico x Determinístico	90

Lista de Figuras

1.1	Componentes de um sistema fotovoltaico.	2
2.1	Representação esquemática de um <i>framework</i> caixa preta (a) e um caixa branca (b).	8
3.1	Diagrama esquemático dos métodos de estimativa de radiação solar uti- lizando os dados medidos do ano padrão.	23
3.2	Ângulo de declinação do sol.	24
3.3	Circuito equivalente de uma célula fotovoltaica real (modelo de um diodo).	30
4.1	Fluxograma para construção e evolução do <i>Solar Energy</i>	37
4.2	Modelo de casos de uso para um sistema fotovoltaico.	39
4.3	Diagrama inicial de pacotes do <i>Solar Energy</i>	40
4.4	Diagrama contendo as classes responsáveis pela execução da simulação.	41
4.5	Diagrama de classes do <i>hot spot IEnvironment</i>	44
4.6	Grafo de execução da simulação fotovoltaica.	46
4.7	Diagrama de seqüência de execução do processo de simulação fotovoltaica.	47
4.8	Diagrama de classes da implementação do padrão de projetos <i>Factory Method</i>	49
4.9	Diagrama de classes do <i>Solar Energy</i>	53
5.1	Diagrama de classes da aplicação Sol Brasil.	58
5.2	Tela principal do aplicativo <i>Sol Brasil</i>	59
5.3	Conjunto de dados climatológicos medidos para a cidade de Belo Horizonte para a identificação do Ano Padrão.	61
5.4	Interface para escolha do ambiente.	61
5.5	Gráfico representativo da curva horária de consumo.	64

5.6	Gráfico representativo da curva mensal de consumo.	65
5.7	Identificação do ano padrão para o período informado.	66
5.8	Relação entre a radiação extraterrestre e da superfície, em função do índice de claridade ao longo do ano.	67
5.9	Radiação horária média mensal no plano inclinado gerada pelo ano padrão.	68
5.10	Potência horária média mensal gerada pelos módulos fotovoltaicos ao longo do ano para 1 painel e 3 baterias.	69
5.11	Energia armazenada nas baterias durante todo o ano (horária média mensal) para 1 painel e 3 baterias.	69
5.12	Potência horária média mensal gerada pelos módulos fotovoltaicos ao longo do ano para 3 painéis e 1 bateria.	70
5.13	Energia armazenada nas baterias durante todo o ano (horária média mensal) para 3 painéis e 1 bateria.	71
5.14	Relação das falhas no dimensionamento.	72
E.1	Energia armazenada nas baterias pelo processo determinístico	92
E.2	Energia armazenada nas baterias pelo processo estocástico	93

Lista de Tabelas

3.1	Pesos atribuídos aos parâmetros meteorológicos.	21
3.2	Intervalos de \bar{K}_t e K_t associados a cada uma das matrizes de transição do banco de dados	26
5.1	Dados de entrada para Belo Horizonte	60
5.2	Especificação do Painel Solar ASE-100-ATF/17	62
5.3	Propriedades do componente bateria 12MC36	63
5.4	Desvio padrão aplicado a cada valor mensal de consumo	64
E.1	Cargas consideradas para o dimensionamento do sistema fotovoltaico	91
E.2	Especificação do Painel Solar I-100	91
E.3	Propriedades do componente bateria PVX-890T	92

Capítulo 1

Introdução

O uso da eletricidade está diretamente relacionado ao modo de vida das pessoas, seja na cidade ou no meio rural. O aumento da produtividade das indústrias, o aumento de vendas no comércio e a evolução nos meios de comunicações e transportes são exemplos desta relação, além de ser fator importante de fixação do homem ao campo, resultando em melhores condições de vida e de produção. O Instituto Nacional de Pesquisas Espaciais (INPE) estima um crescimento da demanda de energia elétrica em 4% a.a., para os países em desenvolvimento, como o Brasil, nos próximos 20 anos [MPAC05]. Este aumento, associado a crescente preocupação das conseqüências ambientais no consumo de combustíveis não renováveis fósseis, impulsiona a pesquisa, o desenvolvimento e o uso de tecnologias denominadas alternativas.

A geração de eletricidade através da energia solar pode contribuir de maneira significativa para a mitigação deste problema resultando em menores impactos ambientais.

Esta geração agrega vantagens na utilização, dentre as quais destacam-se: a possibilidade de geração de energia descentralizada e modular, que permite a construção de pequenos centros geradores localizados próximos ao ambiente de consumo; e a facilidade de instalação e manutenção de seus equipamentos. Além disso, é uma fonte de energia renovável e não poluidora.

Nos sistemas de energia fotovoltaicos, a radiação solar é transformada diretamente em eletricidade através das células fotovoltaicas semicondutoras. Para que uma energia proveitosa¹ seja gerada, as células são agrupadas em estruturas denominadas módulos fotovoltaicos ou módulos de fotoconversão.

Conforme apresentado pela Figura 1.1, um sistema fotovoltaico é constituído basicamente por componentes de *hardware*: um painel solar, uma unidade de armazenamento, que pode ser composta por uma ou várias baterias, uma unidade de controle e condiciona-

¹Tensão e corrente suficientemente capazes de alimentar um sistema elétrico.

mento de potência e um componente inversor do sinal elétrico. O principal objetivo de um painel solar é realizar o processo de geração de energia elétrica através da radiação solar.

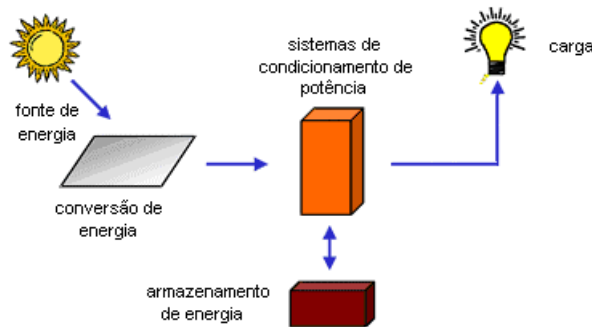


Figura 1.1: Componentes de um sistema fotovoltaico.

Associados a estes componentes estão os aplicativos de controle de processos como: análise estatística dos dados de radiação, do dimensionamento dos módulos fotovoltaicos, de apoio para os cálculos dos equipamentos e os aplicativos simuladores. Este conjunto de *software* é capaz de realizar ensaios do módulo fotovoltaico, informar resultados estatísticos e avaliar a eficiência do processo de conversão nas fases de projeto, construção e testes [SOL99].

Entretanto, alguns simuladores, disponíveis no mercado, não atendem completamente aos requisitos e as necessidades dos pesquisadores. Os estudos nesta área dependem fortemente das características climáticas e meteorológicas de cada região ou localidade e na maioria dos casos, os dados das localidades avaliadas não estão disponíveis.

Uma possível solução é o uso de instrumentos que coletem informações referentes à radiação solar para um curto período de tempo. Esta abordagem, embora simples, não é capaz de representar dados de longos períodos necessários para o dimensionamento dos sistemas [CCE00, MR05]. Destacam-se também, como fatores negativos, o alto custo para a implantação e adequação dos módulos simuladores e a dificuldade de manuseio de seus aplicativos.

Parte dos problemas elencados anteriormente podem ser resolvidos utilizando-se paradigmas e conceitos de Engenharia de Software, que possibilitam a geração de sistemas adaptáveis e flexíveis. O desenvolvimento de *frameworks* para aplicações de uma mesma categoria é um exemplo a ser citado. Em função da aplicação de conceitos de reutilização de código fonte em sistemas e projetos, são passíveis de adaptações atendendo as

necessidades de um domínio específico.

1.1 Motivação

O principal aspecto motivador do desenvolvimento deste trabalho é a ausência de ferramentas nacionais que auxiliem a pesquisa na área de energia fotovoltaica, associado ao elevado custo para a aquisição destes aplicativos.

A evolução do *framework* Solar Energy disponibiliza um arcabouço de funcionalidades para a realização de dimensionamentos e simulações [GPS⁺08, GUI07]. São vantagens desta abordagem:

- A capacidade de maximização do reuso e modularização do código fonte dos *softwares* simuladores de conversão de energia térmica ou fotovoltaica.
- A redução das manutenções e conseqüentemente do tempo de desenvolvimento das diversas aplicações provenientes do *framework*.
- Uma melhor consistência e compatibilidade entre os sistemas desenvolvidos.
- A capacidade de encapsular o conhecimento dos especialistas.
- Uma vez mantido o conhecimento no próprio *framework*, este pode ser utilizado e avaliado por diferentes pesquisadores auxiliando o desenvolvimento de aplicações no domínio de energia solar.

A solução proposta investiga formas de prover o reuso da análise e projeto para sistemas de energia solar definindo as características de cada novo componente, bem como a comunicação entre os mesmos e a apresentação de seus resultados.

Além disto, possibilita manipular dados referentes às pesquisas em energia solar com melhor qualidade, uma vez que se faz uso de aplicativos que já foram desenvolvidos, testados e validados. Simplifica, também, a criação e a utilização de novos componentes, a incorporação de novas funcionalidades às aplicações existentes, além de atualizar dados climatológicos, validar o comportamento do sistema e realizar simulações.

Há, porém, algumas desvantagens na utilização desta abordagem, destacando-se:

- A demanda por um maior esforço para o aprendizado em função da complexidade de construção de um *framework*.
- A dificuldade do processo de depuração do código fonte para a avaliação de possíveis falhas e/ou erros.

- A necessidade da criação de documentos extensos para manutenção e apoio das aplicações.
- A necessidade de informações a respeito dos artefatos criados, tanto no desenvolvimento do *framework*, quanto da aplicação.

Ressalta-se, entretanto, que os benefícios são mais relevantes por auxiliar o projeto de novos componentes, permitir o aprimoramento das fórmulas e proporcionar maior qualidade das informações de entrada dos sistemas.

1.2 Objetivos

O objetivo central desta dissertação é o desenvolvimento de aplicativos para uso na geração de energia solar fotovoltaica, através da utilização do paradigma de implementação por *framework*, potencializando o processo de reutilização de códigos fontes para a construção de *softwares* fotovoltaicos de controle, apoio e simulação.

Para se atingir este objetivo, evoluções e inovações são aplicadas ao *framework Solar Energy*. As alterações completam o arcabouço para sistemas que utilizam a radiação solar como fonte de energia.

O projeto está baseado na análise dos requisitos de sistemas simuladores de arranjos fotovoltaicos e máxima flexibilização dos pontos variáveis do *framework*.

As seguintes etapas foram realizadas neste trabalho:

1. Determinação do escopo das aplicações fotovoltaicas, bem como a compreensão do funcionamento do processo de conversão de energia solar em energia elétrica.
2. Compreensão do processo de geração do ano padrão pelo seqüenciamento sintético e aplicação no cálculo da radiação solar no plano inclinado.
3. Compreensão do processo de dimensionamento dos módulos, baterias, controladores de carga e inversores, utilizando uma abordagem determinística.
4. Evolução e aprimoramento das características do *framework Solar Energy*.
5. Implementação de um aplicativo simulador de um sistema fotovoltaico, como estudo de caso, para testes e validação do *framework*².

²Os procedimentos de definição e modelagem matemática dos componentes mencionados foram implementados com auxílio do Grupo de Estudos em Energia Solar (GREEN-PUC Minas)

1.3 Principais contribuições

A seguir são descritas as principais contribuições alcançadas com este trabalho:

- Projeto e implementação de uma ferramenta que realize simulações e análises do dimensionamento determinístico do processo de conversão de energia solar em elétrica e o controle da demanda da potência gerada pelo arranjo fotovoltaico, a partir de um conjunto de classes e interfaces abstratas.
- Atenuação do tempo e do custo no desenvolvimento, uma vez que os projetistas reutilizam uma solução testada e consolidada anteriormente.
- Facilidade de incorporação, utilização de novos componentes e alteração do código base de forma controlada e não invasiva com a criação e identificação dos pontos de flexibilização presentes em todas as interfaces abstratas do *framework*.
- Apresentação de estudos de casos envolvendo sistemas fotovoltaicos, que utilizam a solução proposta.

1.4 Estrutura da dissertação

Esta dissertação está assim organizada. No Capítulo 2, são fornecidos os conceitos que proporcionam o embasamento teórico, referente aos *frameworks*, necessário à compreensão desta dissertação, bem como os trabalhos relacionados ao domínio do problema. O Capítulo 3 descreve as características de energia fotovoltaica, bem como dos componentes responsáveis pela transformação da energia solar em energia elétrica. O Capítulo 4 apresenta as principais evoluções nos conceitos e processos do *framework Solar Energy*, bem como sua estrutura e a seqüência de execução. O Capítulo 5 apresenta as características, a criação, o funcionamento, a execução e os resultados obtidos com os estudos de caso realizados. Por fim, o Capítulo 6 apresenta as considerações finais e indica possíveis linhas de trabalhos futuros.

Capítulo 2

Estado da Arte: *Frameworks*

2.1 Considerações iniciais

Este capítulo apresenta o panorama da literatura técnica especializada relacionada a um tema da dissertação. As próximas seções abordam conceitos fundamentais de *frameworks*, sua construção e instanciação.

2.2 Conceitos básicos

2.2.1 Reuso de *software*

O reuso de *software* é um objetivo comum nas organizações desenvolvedoras, por melhorar a produtividade, a manutenibilidade e a qualidade tanto do produto final como do processo de desenvolvimento. Conforme mencionado por Silva e Price [SP98], a reutilização refere-se ao código e também a todo o trabalho gerado durante o processo de desenvolvimento, abrangendo desde a fase de análise e levantamento de requisitos, passando pela codificação, até os testes funcionais e de regras de negócio.

O reuso de *software* sofreu um grande impulso na década de 80 com a criação de sistemas de grandes proporções, resultando em avanços nas técnicas, nas bibliotecas e nos ambientes de apoio a reutilização [BAR01]. Entretanto, o reuso de *software* não é uma tarefa simples de ser realizada, tornando-se um dos principais desafios para os pesquisadores da área de Engenharia de Software. Uma proposta é a utilização de *frameworks* orientados a objetos em função de aplicarem o reuso em uma grande variedade de problemas de um mesmo domínio. A seção seguinte descreve esse paradigma.

2.2.2 Frameworks orientados a objetos

O enfoque principal das definições de *frameworks* está na possibilidade de facilitar o reuso. Do ponto de vista estrutural é definido como um conjunto de classes que contém o projeto abstrato de soluções para uma família de problemas relacionados, propiciando reuso com uma maior generalidade [JF88].

Um *framework* é composto por uma coleção de classes abstratas, denominada interfaces e por um conjunto que as implementam, denominadas classes concretas, representando o projeto de um subsistema [PPSS95]. O uso de classes abstratas possibilita a herança de código, preservando o comportamento semântico dos objetos instanciados a partir das classes concretas.

Quando essa coleção de classes abrange um sistema de *software* genérico para um domínio específico de aplicações, utiliza-se o termo *framework* de aplicação. Quanto ao propósito deste paradigma, trata-se de uma aplicação semi-completa reutilizável que, quando especializada, produz aplicações personalizadas.

Frameworks permitem aos desenvolvedores criar novas aplicações a partir de códigos fontes já implementados, agregando conhecimento à ferramenta de um determinado escopo. Podem ser considerados também, como uma técnica de reuso orientado a objetos que compartilha características de reuso de outras técnicas, como *templates* e os *esquemas* [JOH97].

Mohamed Fayad e Douglas Schmitd [FS97] destacam como características fundamentais para propiciar benefícios aos desenvolvedores que utilizem esta abordagem de programação:

- *Reusabilidade* \implies Componentes genéricos podem ser reutilizados para criarem novas aplicações a partir das interfaces.
- *Extensibilidade* \implies A extensão das interfaces é permitida em função dos métodos adaptáveis.
- *Modularidade* \implies As interfaces acessam os detalhes de implementação, que se encontram encapsulados. A modularidade melhora a qualidade do *software* por reduzir o esforço de manutenção tanto no projeto quanto na implementação.
- *Inversão de controle* \implies O *framework* realiza as atividades de gerência e seqüenciamento das atividades da aplicação.

A inversão de controle pode ser considerada como a característica mais marcante de um *framework*. No reuso tradicional de componentes ou bibliotecas, desenvolve-se a

aplicação específica com chamadas aos componentes que se deseja reutilizar. Assim, a responsabilidade pelas chamadas e pelo fluxo de controle é da aplicação.

Ao revés, o *framework* faz o papel do programa principal, coordenando e seqüenciando as atividades das aplicações. A invocação é encapsulada, ou seja, os métodos não são diretamente chamados pelo código das aplicações derivadas. Assim, determina-se o fluxo de controle do programa e sua estrutura geral [FS97].

Com relação a estrutura, o aspecto variável de um domínio de aplicação é denominado ponto variável (*hot spot*) [PPSS95]. As aplicações de um mesmo domínio são diferenciadas por um ou mais *hot spots*. Este conceito representa as partes específicas de um sistema e são projetados para serem genéricos e adaptáveis às necessidades de cada aplicação. Por sua vez, os pontos fixos (*frozen spots*) definem a arquitetura geral de um sistema de *software*, como seus componentes básicos e os relacionamentos entre os mesmos.

2.2.3 Frameworks Caixa Branca, Caixa Preta e Caixa Cinza

Um *framework* pode ser classificado quanto a técnica utilizada para estendê-lo. A diferenciação entre essas classificações se dá através da forma como os métodos são adicionados às subclasses de uma ou mais classes. São definidos como caixa branca (*white box*), caixa preta (*black box*) e caixa cinza (*gray box*) [FS97, JF88]. A Figura 2.1 representa de forma esquemática os *black box* (a) e os *white box* (b) *frameworks*.

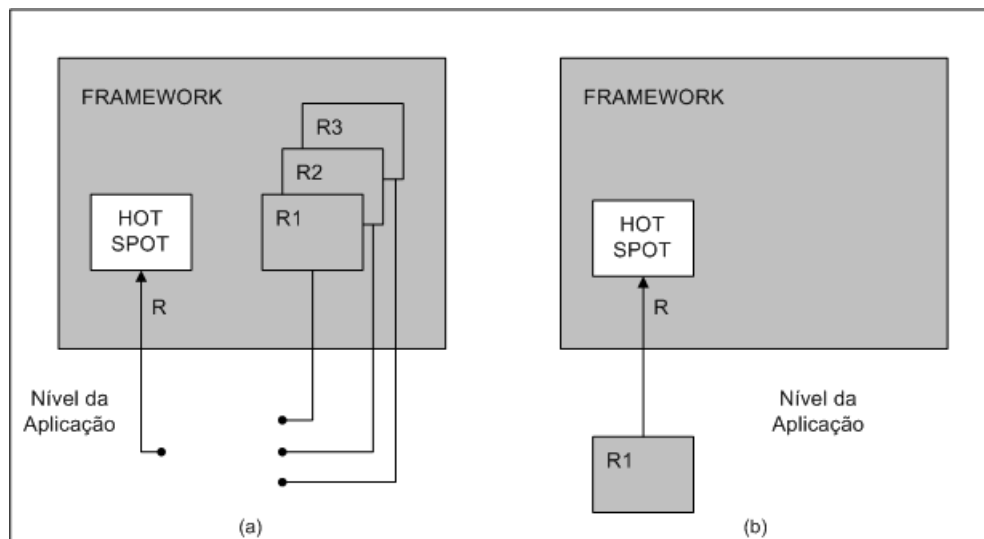


Figura 2.1: Representação esquemática de um *framework* caixa preta (a) e um caixa branca (b).

Um *framework* caixa branca é aquele em que a integração é feita através da imple-

mentação, por herança, das classes ou das interfaces pré-definidas. Possuem a vantagem de serem mais flexíveis, permitindo implementações personalizadas, porém, exigem maior conhecimento sobre a estrutura, detalhes de implementação e funcionamento da ferramenta, dificultando o processo de aprendizado.

Já a classificação caixa preta refere-se àquela em que a integração é feita através da ligação de objetos do aplicativo aos objetos do *framework*. O reuso é por composição ou seja, o desenvolvedor combina diversas classes concretas ao *framework* para obter a aplicação desejada. Este conceito faz com que a interface de comunicação com o desenvolvedor se torne relativamente simples, não sendo necessário um conhecimento de sua estrutura. Há a necessidade de apenas conhecer a API ¹ da ferramenta. Uma vantagem é tornar o aprendizado mais rápido e eficiente, porém a flexibilidade é inferior aos *frameworks* caixa branca.

O *framework* caixa cinza incorpora os aspectos de caixa branca e caixa preta e o reuso é obtido por meio de herança, por ligação dinâmica e também pelas interfaces de definição.

Yassin e Fayad [YF00] relatam que a tendência atual é pela utilização dos *frameworks* caixa cinza em função dos problemas de implementação apresentados na exposição aos desenvolvedores pelos *frameworks* caixa branca e pela maior abstração e menor flexibilidade apresentada pelos *frameworks* caixa preta, tornando difícil a manutenção das aplicações derivadas.

Os *Frameworks* caixa cinza superam as barreiras impostas pelos outros dois tipos, pois possuem flexibilidade e facilidade de extensão, sem expor desnecessariamente informações que não interessam ao desenvolvedor da aplicação.

Neste trabalho, o *framework Solar Energy* caracterizava-se inicialmente como caixa branca, pelo fato de tratar apenas do processo de aquecimento de água. Porém, com as evoluções e inovações desenvolvidas, referentes ao processo de geração de energia elétrica, torna-se um *framework* caixa cinza.

2.3 Aplicações

Conforme apresentado por Fayad *et al* [FJS99], a maioria dos *frameworks* existentes aplica-se a domínios técnicos, tais como interfaces com o usuário e distribuição. Por exemplo, o MacApp, específico para aplicações Macintosh, o Lisa Toolkit, o Interviews e o Smalltalk Model-View-Controller (MVC). Este, o primeiro *framework* amplamente

¹ *Application Programming Interface* (Interface de Programação de Aplicativos). Significa um conjunto de rotinas e padrões estabelecidos por um *software* para utilização de suas funcionalidades por programas que não necessitam de envolvimento em detalhes da implementação.

utilizado através das técnicas de orientação a objetos, servindo para implementações de interfaces gráficas com o usuário.

Os autores apresentam também *frameworks* que estendem-se a outros domínios como os utilizados em algoritmos de roteamento VLSI (*Very Large Scale Integration*), sistemas hipermídia, sistemas operacionais e controle de manufaturas. Com o surgimento de novas linguagens, como *Java*, novas possibilidades são apresentadas, como o AWT (*Abstract Windows Toolkit*), integrante do JFC (*Java Foundation Classes*) e do *Java Beans*.

Há também, diferentes *frameworks* relacionados à área de negócios e gestão, dentre os quais destaca-se o *IBM SanFrancisco* [ABD⁺97]. Esta ferramenta foi desenvolvida para auxiliar na construção de aplicações relacionadas a gerência de compras e vendas, além do controle de clientes, produtos e serviços básicos ofertados.

Outro exemplo, para um domínio específico, é o GREN [BM04]. O *framework* trata da Gestão de Recursos de Negócios, sendo estruturado em três camadas: a primeira de persistência, a segunda de aplicação e a terceira de interface gráfica com o usuário. Ressalta-se que a criação de aplicações por meio deste *framework* é guiada pela aplicação da Linguagem de Padrões ². Em seguida, são obtidos os diagramas de classes da aplicação desejada. Com base nos diagramas, utilizam-se as classes pré-programadas do *Framework GREN* para criar o código da nova aplicação.

Por fim, relacionado ao domínio deste trabalho, cita-se o *Solar Energy* [GUI07], que foi desenvolvido com o propósito de colaborar com as pesquisas em energia solar. Foi definido como um arcabouço de aplicações relacionadas ao aquecimento de água e sua estrutura baseada em camadas, que permitem aos desenvolvedores realizar a expansão do código, disponibilizando classes e interfaces a serem estendidas e implementadas. As principais funcionalidades fornecidas são: o cadastro de novos dados climatológicos, essenciais ao funcionamento dos sistemas, a realização da conexão entre os dispositivos do dimensionamento, a transferência de informações, a gerência dos dados fornecidos pela radiação solar, a simulação de processos de aquecimento de água, a adição de novas interfaces gráficas e a disponibilização de componentes básicos, necessários ao domínio da aplicação.

2.4 A construção de *frameworks*

A construção de *frameworks* vem sendo alvo de pesquisas, resultando em diferentes propostas, como as descritas a seguir.

Prece [PRE99] apresenta um processo para construção que se inicia com a definição de

²É uma coleção estruturada de padrões que auxiliam na transformações de requisitos e restrições de uma aplicação em uma arquitetura [COP98].

um modelo de objetos de uma aplicação específica. O refinamento é dado por meio de um ciclo de construção composto pela identificação dos pontos variáveis do *framework* e por sua documentação. Os pontos variáveis são projetados e implementados e o *framework* testado, garantindo os requisitos do domínio. Novos pontos variáveis podem ser encontrados nestas etapas e o ciclo se repete. Ao término de cada iteração avalia-se o *framework* para determinar ou não um novo ciclo de refinamento. Esta abordagem sistematiza a construção do *framework*, mas não define o processo de instanciação por meio do qual as aplicações concretas são obtidas.

Outra técnica de construção, a partir de um modelo específico, é a apresentada por Schmid [SCH97]. Sugere-se que o *framework* seja desenvolvido por generalização sistemática tendo por base um modelo de classes de uma aplicação fixa. Inicialmente faz-se uma análise em alto nível para a determinação dos pontos variáveis, estabelecendo os principais aspectos do sistema. A seguir, cada ponto variável é analisado em detalhes, produzindo então, sua especificação, que passará ainda pelo projeto de alto nível, gerando vários subsistemas de pontos variáveis. No último passo, deve-se transformar o modelo de classes da aplicação fixa no modelo de classes do *framework*, por meio da substituição de grupos de classes do modelo original pelos subsistemas de pontos variáveis correspondentes. Esta abordagem também não define o processo de instanciação do *framework* resultante.

Bosch *et al* [BMM⁺99] apresenta uma forma de desenvolvimento composta de seis etapas. A primeira trata da análise do domínio, que é executada para descrever a cobertura realizada pelo *framework* e para captar seus principais conceitos e requisitos, resultando em um modelo de análise do domínio. Na segunda, cria-se o projeto arquitetural, baseado no modelo de requisitos. Na terceira fase, refina-se o projeto, que é implementado na quarta fase, utilizando uma linguagem de programação específica. Na quinta, tanto o *framework*, quanto um aplicativo derivado são testados para avaliar as funcionalidades e usabilidade das ferramentas. Finalmente, na sexta e última fase, são realizadas documentações por meio de um guia do usuário e manual do projeto. Nota-se nessa abordagem uma preocupação maior, em relação às abordagens anteriores, com a usabilidade do *framework*, visto que a fase de testes explicita a avaliação da facilidade de uso deste paradigma e que a fase de documentação produz um guia de utilização ao usuário desenvolvedor.

Há também, outra abordagem relacionada ao processo de construção definido por Johnson e Woolf [JW98]. Descreve-se por uma linguagem de padrões, denominada *Evolução de Frameworks*, aplicados à construção de um *framework* caixa preta com todos os recursos necessários para facilitar sua instanciação. Porém, inicialmente é sugerido que

as aplicações concretas sejam desenvolvidas e generalizadas de forma gradual, para a produção de um *framework* caixa branca. Posteriormente, padrões são aplicados para evoluir de forma progressiva em um *framework* caixa preta. Para facilitar a aplicação do método utiliza-se um construtor visual e um conjunto de ferramentas de linguagem.

As abordagens descritas por Pree, Schmid, Johnson e Woolf partem de modelos de aplicações particulares, incluindo a flexibilidade desejada posteriormente. Na abordagem proposta por Bosch *et al*, o modelo de análise do domínio é obtido no início do processo, o que torna os pontos variáveis do *framework* mais previsíveis.

A proposta desta dissertação é fazer uso de etapas que vão desde a captação das informações sobre o domínio das aplicações em energia solar, até o desenvolvimento do *framework*. Em particular, os pontos variáveis são identificados no início do processo para minimizar o número de ciclos de iteração e, ao mesmo tempo, facilitar o subsequente aprendizado e uso do *SolarEnergy*.

2.5 A instanciação de *frameworks*

O principal problema ao instanciar uma aplicação a partir de um *framework* é saber exatamente quais pontos variáveis devem ser adaptados e como adaptá-los. As abordagens propostas como solução deste problema contam com a documentação para obter a instanciação das aplicações específicas.

Considerando-se um *framework* caixa branca, essa documentação é, em geral, constituída da hierarquia de classes, das classes abstratas que precisam ser especializadas na nova aplicação, dos métodos a serem sobrepostos nessas classes e de exemplos de aplicações derivadas a partir do *framework*.

A instanciação se dá por documentação, por exploração de exemplos, por meio de padrões e por meio de *cookbooks*.

A primeira abordagem consiste do estudo da documentação do *framework*, composta de sua hierarquia de classes, código fonte e outros documentos. Isso pode ser feito com treinamento convencional ou por meio de tutoriais especialmente elaborados com essa finalidade. As desvantagens desta abordagem são o tempo necessário para o aprendizado do paradigma de programação e a dificuldade para determinar se o nível de compreensão é suficiente para iniciar o uso do *framework* [FRA01].

A exploração de exemplos, como proposto por Gangopadhyay e Mitra [GM95], consiste na análise de aplicações existentes, construídas a partir do *framework*, para identificar as adaptações necessárias para obtenção do sistema específico. A dificuldade desta abordagem é encontrar, entre os exemplos, uma aplicação similar àquela a ser desenvolvida já

que a análise é *ad-hoc*³.

Padrões de projeto podem ser empregados para documentar e instanciar *frameworks* uma vez que descrevem problemas recorrentes no projeto de sistemas e sua solução em termos de interfaces e objetos. Cada qual resolve um problema com ocorrências repetidas e descreve o núcleo da solução de tal maneira que possa ser reutilizado. Os padrões abrangem diferentes níveis de abstração, possuem grande aplicabilidade prática e são classificados em diversas categorias, de modo a facilitar a sua recuperação e utilização em um determinado problema [GHEV93]. Três trabalhos foram publicados nessa linha [BJ94, BS00, JOH92], mas não evoluíram rumo à definição de um processo a ser utilizado por outros desenvolvedores.

A instanciação por *Cookbooks* é um conjunto de tarefas, uma lista de passos a serem seguidos, exigidas para adaptar um *framework* para uma aplicação específica. Diversos trabalhos se baseiam nessa idéia [KP88, OCM00, PPSS95]. Entretanto, é pouco provável que as tarefas que compõem a lista possam ser executadas passo a passo, pois uma escolha feita durante o processo de especialização pode causar alterações nas etapas posteriores.

Tanto a abordagem de *cookbooks* quanto a abordagem de padrões têm o intuito de instanciar o *framework* com o mínimo de conhecimento do mesmo, ou seja, os usuários desenvolvedores concentram-se na adaptação dos pontos variáveis relativos a um problema específico.

Este trabalho baseia-se na idéia de verificar a documentação gerada pelo processo de construção do *framework* e também na lista de passos a serem seguidos para realizar a instanciação do *Solar Energy*.

2.6 Considerações finais

Inicialmente, apresentou-se, neste capítulo, uma breve introdução ao conceito de reuso e *frameworks*, necessários ao entendimento da proposta desta dissertação. Em seguida, foi feita uma revisão das principais pesquisas e trabalhos relacionados às etapas de criação e instanciação de *frameworks*.

Em todas as metodologias foram apresentados problemas nas fases de definição e execução das aplicações a partir de *frameworks*. A grande dificuldade encontrada é saber exatamente quais pontos variáveis devem ser adaptados e como realizar esta função para tornar a ferramenta flexível e adaptável a aplicações de um mesmo escopo.

³O termo *ad-hoc* é utilizado para representar ciclos completos de construção de *softwares* que não foram devidamente projetados, para assegurar que este esteja de acordo com os padrões e especificações de prazo, qualidade e custo. Geralmente está focado na necessidade específica de algum usuário ou situação.

Para identificar os possíveis pontos variáveis, deve-se analisar os conceitos referentes ao processo fotovoltaico bem como a modelagem matemática dos componentes usados no dimensionamento para a geração de energia elétrica por meio da radiação solar. O capítulo a seguir detalha estes conceitos.

Capítulo 3

Estado da Arte: Geração Fotovoltaica

3.1 Considerações iniciais

Este capítulo apresenta o panorama relacionado aos processos fotovoltaicos. As próximas seções abordam os conceitos e as modelagens matemáticas referentes aos dispositivos envolvidos.

3.2 Introdução

A denominação fotovoltaica engloba o conjunto das tecnologias que permitem a conversão direta da luz solar em eletricidade através dos dispositivos eletrônicos, denominados células solares, conectados em um módulo fotovoltaico ou de fotoconversão [CMOD04].

A geração de energia elétrica por meio da luz solar iniciou-se em 1958 com projetos espaciais para colocação de satélites em órbita. As primeiras aplicações em projetos terrestres ocorreram na década de 70, quando foram inauguradas empresas dedicadas às aplicações domiciliares. As primeiras usinas para geração fotovoltaica foram criadas em 1983 e, entre os anos 2002 e 2003, por exemplo, a Alemanha já continha algumas das maiores instalações solares, gerando potência da ordem de 4 a 5 MW [CAB06].

Conforme Galdino *et al* [GLRS04], nações como os Estados Unidos, alguns países da Europa e Japão investem maciçamente na área, além de incentivar a população a utilizar a energia solar como fonte geradora de energia elétrica. Como resultado, tornam-se referências mundiais neste campo de estudo.

No Brasil, há grandes vantagens de utilização deste tipo de energia. Por ocupar

uma grande extensão territorial compreendida entre a linha do Equador e o Trópico de Capricórnio há elevados índices de radiação solar por praticamente todo o ano. No entanto, a maioria dos sistemas fotovoltaicos instalados estão isolados.

A estimativa para o país é que a potência total seja da ordem 15 MW, sendo 70% localizado nas regiões Nordeste, Norte e Centro-Oeste [ZLS00]. No entanto, segundo o 19º Balanço Energético do Estado de Minas Gerais (BEEMG) com base no ano de 2003, a energia solar participou de forma pouco expressiva na matriz energética do estado [CEM04].

Um sistema fotovoltaico não emite poluentes durante sua operação e pode ser considerado como uma fonte promissora de energia sustentável. Entretanto, pode gerar impactos ambientais que devem ser considerados. Inatomi e Udaeta [IU05] apresentaram os seguintes impactos:

- Emissões de produtos tóxicos durante o processo da matéria-prima para a produção dos módulos e componentes periféricos, tais como ácidos e produtos cancerígenos,
- Ocupação de área para implementação do projeto e possível perda de habitat, (crítico apenas em áreas especiais),
- Impactos visuais,
- Riscos associados aos materiais tóxicos utilizados nos módulos fotovoltaicos (arsênico, gálio e cádmio) e outros componentes, ácido sulfúrico das baterias (incêndio, derramamento de ácido, contato com partes sensíveis do corpo),
- Necessidade de se dispor e reciclar corretamente as baterias (geralmente do tipo chumbo-ácido, e com vida média de quatro a cinco anos) e outros materiais tóxicos contidos nos módulos fotovoltaicos e demais componentes elétricos e eletrônicos, sendo a vida útil média dos componentes estimada entre 20 e 30 anos.

3.3 Ano padrão

O desempenho das aplicações, que visam o aproveitamento de energia solar, é altamente dependente dos aspectos climáticos. O custo computacional para realizar a análise dos dados meteorológicos provenientes de ensaios e simulações de vários anos é elevado, podendo também, gerar resultados inconclusivos.

Uma alternativa para o problema é o uso do conceito de ano de referência cujo objetivo é a caracterização das variáveis meteorológicas e a representação de um período de tempo

passado para a redução do custo de processamento nas simulações. Além disto, permite que o clima de uma determinada região seja caracterizado em um conjunto de dados correspondente a um único ano.

Diversas são as metodologias para se identificar um ano de referência, dentre as quais, destacam-se: os TRY (*Test Reference Year*), TMY (*Typical Meteorological Year*), DRY (*Design Reference Year*) e SRY (*Short Reference Year*). Todas visam caracterizar o clima de uma determinada localidade utilizando dados passados [FRD88].

O trabalho realizado por Argiriou *et al* [ALKB99] aplica as metodologias para identificar o melhor processo de seleção do ano de referência para diversos sistemas que utilizem a energia solar como fonte. Coleta dados meteorológicos para a cidade de Atenas por um período de vinte anos e os aplica em sistemas como: aquecimento de água, fotovoltaicos e construção sustentável.

Foi constatado que os melhores resultados na identificação do ano padrão para sistemas fotovoltaicos deve incluir critérios de seleção as variáveis climáticas utilizadas. A proposta consiste em utilizar a abordagem apresentada por Festa e Ratto [FR93] e estendê-la de forma a incluir pesos as variáveis climáticas analisadas com base na relevância do referido sistema. A seção seguinte descreve o método em detalhe.

3.3.1 Método de Festa e Ratto

Este método visa realizar uma padronização das variáveis climatológicas, comparar, na base de dados, a distribuição de frequência relativa de cada mês com a frequência a longo-prazo, considerar os valores diários subseqüentes para os parâmetros meteorológicos e por fim, identificar os meses padrão. Os passos utilizados por esta metodologia são descritos a seguir.

Análise das variáveis consideradas

A partir de dados climatológicos horários como temperatura, umidade relativa do ar, vento e radiação, para um determinado ano, chega-se aos dados diários conforme a matriz x , mostrada por 3.1:

$$x(y, m, d)(y = 1...ny; m = 1...12; d = 1...nd(m)) \quad (3.1)$$

onde:

$y \rightarrow$ ano;

$m \rightarrow$ mês;

$d \rightarrow$ dia;

$ny \rightarrow$ número de anos considerados;

$nd(m) \rightarrow$ número de dias de um mês.

Padronização das variáveis

Para as variáveis consideradas, deve-se calcular a média correspondente, além do desvio padrão. A média M_x é dada por:

$$M_x(m, d) = \frac{1}{ny} \sum_y x(y, m, d) \quad (3.2)$$

e o desvio padrão σ_x por:

$$\sigma_x(m, d) = \sqrt{\frac{1}{ny} \sum_y [x(y, m, d) - M_x(m, d)]^2} \quad (3.3)$$

A partir da metodologia proposta por Festa *et al* [FRD88], a suavização da média M_x e do desvio-padrão σ_x foram feitas utilizando uma base trigonométrica. Para todos os valores diários, realiza-se a padronização das variáveis com relação a tendência, sendo definido como o resíduo padronizado X e extraído através da seguinte matriz:

$$X(y, m, d) = \frac{x(y, m, d) - M_x(m, d)}{\sigma_x(m, d)} \quad (3.4)$$

Ajuste linear das variáveis

As séries temporais geradas pelos cálculos das médias e dos desvios padrões podem exibir fortes variações em suas escalas. Logo, deve-se aplicar um ajuste linear, também denominado suavização dos dados conforme relatado por Reboita [REB05]. Consiste em:

- Dividir a seqüência em M partes consecutivas onde M é o número de meses do ano.
- Calcular para cada seqüência de M sua média.
- Construir uma seqüência média \bar{x}_d com a condição de que os M valores médios parciais sejam os mesmos.

Como exemplo, caso seja informada uma seqüência de 12 médias mensais de parâmetros meteorológicos como radiação, chuva, temperatura e umidade, o ajuste linear é dado por:

$$\bar{x}_d = A_0 + \sum_{k=1}^5 \left(A_k \cos \frac{2\pi}{N} kd + B_k \sin \frac{2\pi}{N} kd \right) + B_6 \sin \left(\frac{2\pi}{N} 6d - \phi \right) \quad (3.5)$$

e

$$\langle \bar{x}_d \rangle_m = y_m (m = 1, 2, \dots, 12) \quad (3.6)$$

onde:

$\langle \rangle_m \rightarrow$ indica a média relativa a um determinado mês resultando em:

$$y_m = A_0 + \sum \left(A_k \left\langle \cos \frac{2\pi}{N} kd \right\rangle_m + B_k \left\langle \sin \frac{2\pi}{N} kd \right\rangle_m \right) + B_6 \left\langle \sin \frac{2\pi}{N} kd \right\rangle_m \quad (3.7)$$

e os coeficientes sendo calculados pelas seguintes expressões:

$$A_0 = \frac{1}{M} \sum_{m=1}^M y_m \quad (3.8)$$

$$A_k = \frac{2}{M} \sum_{m=1}^M y_m \cos \frac{\pi k(2m-1)}{M} \quad (3.9)$$

$$B_k = \frac{2}{M} \sum_{m=1}^M y_m \sin \frac{\pi k(2m-1)}{M} \quad (3.10)$$

$$B_{M/2} = -\frac{1}{M} \frac{\pi}{2} \sum_{m=1}^M (-)^m y_m \quad (3.11)$$

Correlação do tempo

O próximo passo é identificar, por meio da matriz z , a seguir, a correlação do tempo:

$$z(y, m, d) = X(y, m, d)X(y, m, d+1) \quad (3.12)$$

Os mesmos procedimentos de padronização e ajuste linear das variáveis, realizados para os dados de entrada, conforme as equações de 3.2 a 3.11, são aplicados aos dados da correlação do tempo resultando em um novo resíduo padronizado Z :

$$M_z(m, d) = \frac{1}{ny} \sum_y z(y, m, d) \quad (3.13)$$

$$\sigma_z(m, d) = \sqrt{\frac{1}{ny} \sum_y [z(y, m, d) - M_z(m, d)]^2} \quad (3.14)$$

$$Z(y, m, d) = \frac{z(y, m, d) - M_z(m, d)}{\sigma_z(m, d)} \quad (3.15)$$

Cálculo da distância

Para todas as variáveis x , para cada mês m , do ano y , são executados os cálculos das grandezas:

1. Cálculo da média mensal do resíduo padronizado,
2. Cálculo da média diária do resíduo padronizado,
3. Cálculo do desvio padrão mensal do resíduo,
4. Cálculo da média em longo prazo do resíduo,
5. Cálculo da desvio padrão em longo prazo do resíduo,
6. Distribuição cumulativa de frequência recuperando os valores para todas as variáveis,
7. Cálculo da distribuição de frequência para os resíduos,
8. Cálculo da distribuição de frequência para a média diária dos resíduos,
9. Cálculo da distância das médias, pela diferença entre as médias mensais,
10. Cálculo da distância dos desvios padrões, pela diferença entre as os desvios padrões mensais,
11. Cálculo da distância ponderada.

Estes passos são executados tanto para os resíduos padronizados de X , quanto para os resíduos padronizados de Z , ou seja, tanto para as os valores das variáveis, quanto para os valores da correlação do tempo, obtendo-se resultados semelhantes em ambos os casos. Todas as equações dos processos descritos acima são apresentadas no Apêndice A.

Processo de seleção

Após o cálculo da distância para todas as variáveis climáticas analisadas, deve-se realizar o processo de seleção, ou seja, a identificação dos 12 meses que irão compor o ano de referência. O método consiste em atribuir para cada mês o pior máximo entre as $2N$ distâncias. Supondo que este parâmetro seja definido por:

$$d_{\max}(y, m) \quad (3.16)$$

então para cada mês do ano tem-se:

$$d_{\min \max}(m, 1) = \min \{d_{\max}(y, m) | 1 \leq y \leq ny\} \quad (3.17)$$

3.3.2 Extensão no Método de Festa e Ratto

A fim de tratar os parâmetros meteorológicos com base na importância de sua influência em determinado sistema, Argiriou *et al* [ALKB99] estendeu o modelo de Festa e Ratto [FR93] e usou fatores de peso para cada parâmetro às distâncias ponderadas calculadas.

Este processo desconsidera apenas a aproximação mínima e máxima das variáveis e assegura que o mês, selecionado como padrão, tenha valores médios típicos e variações típicas para os parâmetros meteorológicos. A Tabela 3.1 apresenta os pesos associados aos

Tabela 3.1: Pesos atribuídos aos parâmetros meteorológicos.

Temperatura	0.1
Temperatura Máxima	0.05
Umidade relativa do ar	0.05
Velocidade do vento	0.05
Radiação	0.25

parâmetros climáticos. A proposta define que a soma de todos os pesos de cada parâmetro avaliado não seja inferior a 0 e nem superior a 1.

3.4 Radiação Solar

A radiação que atinge a atmosfera terrestre é denominada radiação extraterrestre. Esta energia possui uma orientação bem definida da direção Sol-Terra. Entretanto, os raios solares sofrem influência da nossa atmosfera, o que acarreta em modificações na intensidade, no espectro e na própria direção com que atingem a superfície [GP95].

Basicamente, a radiação global ¹ que incide sobre um corpo localizado no solo pode ser decomposta em três variações [DB91]:

1. *Radiação solar direta* \Rightarrow Radiação que atravessa a atmosfera sem sofrer alteração na sua direção;

¹Radiação global ou total é a soma das componentes direta e difusa da radiação solar incidente.

2. *Radiação solar difusa* \Rightarrow Radiação recebida por outro corpo em virtude da direção dos raios solares ter sido modificada por reflexão ou espalhamento na atmosfera;
3. *Radiação refletida* \Rightarrow Radiação que depende das características do solo e da vizinhança.

Em função do eixo de rotação da Terra em torno do Sol variar de acordo com a região que se analisa, a presença de nuvens e poluentes e as muitas variações climáticas, não há a possibilidade de se determinar, precisamente, o valor da radiação solar em um plano horizontal na superfície terrestre. Por isto, aplicam-se os métodos de estimativa da radiação, como descritos a seguir.

3.4.1 Métodos de Estimativa da Radiação Solar

Será apresentado nesta seção os modelos clássicos que estimam desde a radiação global em média mensal no plano horizontal até a radiação global horária no plano inclinado².

As equações, que descrevem os modelos, são estimativas de valores na superfície terrestre em função das variações climáticas que podem ocorrer nas horas, dias e meses que são analisados. Por exemplo, períodos de chuva para uma estação atípica acarretarão distorções nos valores coletados em relação aos valores calculados [GP95].

A Figura 3.1 representa um resumo dos métodos de estimativa baseados em dados históricos medidos e a identificação do provável ano de referência para as variáveis climáticas analisadas. Deve-se evidenciar que a radiação global pode ser calculada também baseada no número de horas de insolação solar, conforme realizado por Guidoni *et al* [GPS+08].

Radiação Global em média mensal no plano horizontal

A identificação do ano padrão foi realizada, conforme apresentado na Seção 3.3 e servirá de entrada para a estimativa mensal no plano horizontal. Contudo é necessária a inserção de coeficientes empíricos, os quais dependem da localidade e do dia do ano. A primeira equação para o cálculo da radiação global no plano horizontal é denominada equação de regressão do tipo Angstrom [DB91].

$$\frac{\bar{H}}{H_c} = a + b \frac{\bar{n}}{N} \quad (3.18)$$

onde:

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

²Inclinação relativa aos coletores térmicos ou painéis solares

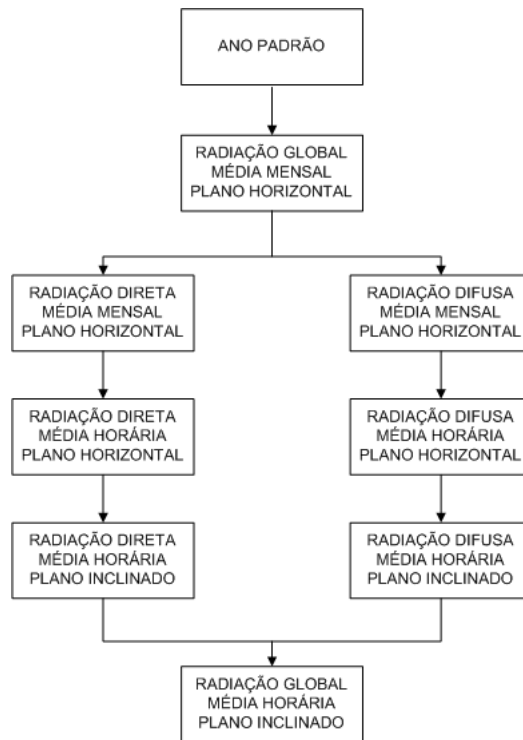


Figura 3.1: Diagrama esquemático dos métodos de estimativa de radiação solar utilizando os dados medidos do ano padrão.

\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,

\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 3.19.

$$\overline{N} = \frac{2}{15} \cos^{-1}(-\tan \phi \tan \delta) \quad (3.19)$$

onde:

ϕ → latitude da região,

δ → ângulo de declinação do sol, representada pela equação 3.20 e ilustrada pela Figura 3.2.

$$\delta = 23,45 \sin \left(2\pi \frac{284 + d}{365} \right) \quad (3.20)$$

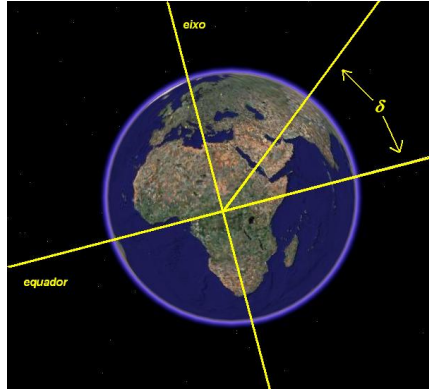


Figura 3.2: Ângulo de declinação do sol.

onde:

$d \rightarrow$ representa o dia do ano. Se d for igual a 1 indica o primeiro dia de Janeiro. Caso seja igual a 32 indica o primeiro dia de 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, Guimarães e Pereira [GP95] apresentaram um novo modelo formulado por Bennet, adicionando os efeitos relativos a altitude e época do ano. O modelo é amplamente utilizado nos dias atuais, porém, desenvolvido para o Hemisfério Norte. Sua utilização no Hemisfério Sul necessita de adaptações, conforme demonstrado pela Equação 3.21:

$$\frac{\bar{H}}{H_o} = a + b \frac{\bar{n}}{N} + ch \quad (3.21)$$

onde:

$\bar{H}_o \rightarrow$ radiação extraterrestre,

$h \rightarrow$ altitude da estação,

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

Calcula-se a radiação extraterrestre através da equação 3.22:

$$H_o = \frac{24 * 3600 G_{sc}}{\pi} \left(1 + 0,033 \cos \left(\frac{2\pi d}{365} \right) \right) (\cos \phi \cos \delta \sin \omega_s + \omega_s \sin \phi \sin \delta) \quad (3.22)$$

onde:

$G_{sc} \rightarrow 1367W/m^2$, 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 [DB91];

$\omega_s \rightarrow$ Hora angular do pôr-do-sol, definida pela equação 3.23.

$$\cos \omega_s = \cos^{-1}(-\tan \phi \tan \delta) \quad (3.23)$$

Conforme mencionado anteriormente, o processo de estimativa de radiação global média mensal no plano horizontal pode ser calculado mediante as médias mensais da radiação no processo de identificação de um ano meteorológico típico. Dessa forma, a aplicação da equação de regressão, apresentada pela Fórmula 3.21, não é necessária, pois não se trata da recuperação dos dados de insolação e sim da aplicação de dados medidos ao longo do tempo para uma determinada localidade.

Radiação direta e difusa em média mensal no plano horizontal

O cálculo da radiação difusa em média mensal foi proposto por Liu e Jordan em 1960 [GP95]. 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. A equação que descreve a radiação global diária incidente sobre uma superfície horizontal é apresentada em Cabral *et al* [CMOD04] e dado por:

$$H = H_o K_t \quad (3.24)$$

em que

$H_o \rightarrow$ Radiação diária extraterrestre incidente sobre uma superfície horizontal ($MJm^{-2}dia^{-1}$),

$K_t \rightarrow$ Índice de claridade diário.

O valor do índice de claridade diário foi estimado com base nas *cadeias de Markov* [ACPC88] e será posteriormente descrito. Algumas adaptações são feitas nas equações para incorporar a hora angular do pôr-do-sol, de acordo com o dia médio representativo do mês³ [DB91]:

$$\frac{\overline{Hd}}{H} = 0,775 + 0,00606(\omega_s - 90) - [0,505 + 0,0045(\omega_s - 90)] \cos(115\overline{K}_T - 103) \quad (3.25)$$

³Dia do mês cuja declinação se aproxima da média aritmética das declinações diárias.

onde:

$\overline{H}d \rightarrow$ Radiação difusa em média mensal,

$H \rightarrow$ Radiação global diária incidente sobre uma superfície horizontal.

Índice de claridade diário - K_t

Foi apresentado por Cabral [CAB06] uma estimativa de dados de irradiação horária global a partir de dados diários do índice de claridade (K_t), gerados sinteticamente, a partir de matrizes de transição de *Markov*.

Foram analisadas seqüências históricas de dados procedentes de diversas estações de medida e proposto um banco de dados universal. A cada matriz de *Markov* associa-se um intervalo de valores de K_t definido por seus valores máximo e mínimo, $K_{t\max}$ e $K_{t\min}$, respectivamente. Cada um destes intervalos foram divididos em 10 subintervalos, com mesmo número de linhas e colunas que cada matriz possui. A Tabela 3.2 apresenta os intervalos de \overline{K}_t - índice de claridade mensal - e K_t , associados a cada uma das matrizes de transição do banco de dados.

Tabela 3.2: Intervalos de \overline{K}_t e K_t associados a cada uma das matrizes de transição do banco de dados

Nº Matriz	1	2	3	4	5	6	7	8	9	10
$\overline{K}_t >$	0.000	0.300	0.350	0.400	0.450	0.500	0.550	0.600	0.650	0.700
$\overline{K}_t \leq$	0.300	0.350	0.400	0.450	0.500	0.550	0.600	0.650	0.700	1.000
$K_{t\min}$	0.031	0.058	0.051	0.052	0.028	0.053	0.044	0.085	0.010	0.319
$K_{t\max}$	0.705	0.694	0.753	0.753	0.807	0.856	0.818	0.846	0.842	0.865

O método de geração das seqüências de K_t - índice de claridade diário - demanda, exclusivamente, como dados de entrada, dos valores de irradiância global diária média mensal (\overline{H}). Os passos desta metodologia são apresentados a seguir:

1. Determinar os 12 valores de \overline{K}_t ,
2. Selecionar a matriz correspondente ao \overline{K}_t de janeiro e calcular seus 10 subintervalos,
3. Selecionar a linha correspondente ao subintervalo em que está compreendido o valor de K_t do dia anterior. No caso do cálculo do primeiro valor do mês de janeiro, foi determinado que este valor coincidissem com o \overline{K}_t do mês de dezembro,
4. Gerar um número aleatório z , compreendido entre 0 e 1, com distribuição uniforme,

5. Somar os elementos de cada coluna da linha i , selecionada, começando pelo elemento da primeira coluna da esquerda, $j = 1$, até que a somatória cumpra a seguinte condição:

$$\sum_{n=1}^{n=j} a_{in} > z, \quad (3.26)$$

6. O K_t do dia em questão é o valor intermediário do subintervalo j , correspondente ao número da última coluna contemplada na somatória anterior,
7. Repetir este procedimento até completar o mês em questão,
8. Selecionar a matriz correspondente ao próximo mês e repetir o processo.

Radiação difusa, direta e global em média horária no plano horizontal

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, dado pela equação a seguir [CAB06]:

$$r_t = \frac{I}{H} \quad (3.27)$$

onde:

$I \rightarrow$ radiação global horária;

$H \rightarrow$ radiação global diária.

A equação, expandida resulta em:

$$r_t = (a + b \cos \omega) \frac{\cos \omega - \cos \omega_s}{\sin \omega_s - \frac{\pi \omega_s}{180} \cos \omega_s} \quad (3.28)$$

onde:

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

$a, b \rightarrow$ coeficientes empíricos de Angstrom definidos pelas equações 3.29 e 3.30.

$$a = 0,409 - 0,5016 \sin(\omega_s - 60) \quad (3.29)$$

$$b = 0,6609 - 0,4767 \sin(\omega_s - 60) \quad (3.30)$$

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

$$r_d = \frac{I_d}{H_d} \quad (3.31)$$

onde:

$I_d \rightarrow$ radiação difusa horária;

$H_d \rightarrow$ radiação difusa diária.

Conforme apresentado no trabalho de Guimarães e Pereira, a seguinte equação foi proposta, tendo a razão das componentes difusa e diária:

$$r_d = \frac{\pi}{24} \left(\frac{\cos \omega - \cos \omega_s}{\sin \omega - \frac{\pi \omega_s}{180} \cos \omega_s} \right) \quad (3.32)$$

Radiação global em média horária incidente 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.

A radiação difusa é decomposta em [GP95]:

- 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:

$$I_T = I_b R_b + I_d \left(\frac{1 + \cos \beta}{2} \right) + I \rho_g \left(\frac{1 - \cos \beta}{2} \right) \quad (3.33)$$

onde:

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

$I \rightarrow$ 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;

ρ_g → refletância das superfícies próximas que podem ser vistas pela superfície incluída.

3.5 Painel Fotovoltaico

O painel é o componente de um sistema fotovoltaico que transforma a energia solar em energia elétrica. Este processo é resultante de um balanço de energia absorvida pelos componentes semicondutores que compõem o módulo solar. A geometria das células interfere diretamente no processo de transformação, pois quanto maior o espaço ocupado, maior será a absorção de energia. Em geral, as células quadradas ocupam melhor a área disponível. Entretanto as células circulares possuem a vantagem de não sofrerem perda de material devido à forma cilíndrica de crescimento, quando aquecidas [SOL99].

A curva que relaciona a tensão e corrente, geradas por um módulo fotovoltaico, é dada por $I \times V$ e está associada às condições de intensidade solar e temperatura em que foi obtida.

Normalmente, os fabricantes dos módulos fotovoltaicos apresentam uma família de curvas para as condições padrão de teste, temperaturas e diferentes intensidades solares. Informações sobre a tensão e corrente no ponto de máxima potência, tensão de circuito aberto, corrente de curto-circuito, coeficiente de temperatura para a corrente de curto-circuito na irradiância solar de referência e coeficiente de temperatura para a tensão de circuito aberto na irradiância solar de referência também são fornecidas. A irradiância solar de $1000W/m^2$ e a temperatura de $25^\circ C$ são as condições padrão de teste para o arranjo fotovoltaico.

O valor máximo de potência para uma determinada intensidade solar encontra-se no ponto de inflexão da curva característica do módulo. O desempenho dos módulos fotovoltaicos é influenciado, principalmente, pela intensidade luminosa e pela temperatura das células.

O aumento do nível de insolação aumenta a temperatura da célula e, conseqüentemente, tende a reduzir a eficiência do módulo [SOL99]. Por este motivo, as células solares cristalinas atingem a sua maior eficiência a baixas temperaturas, sendo que os coeficientes dependem do tipo de material. Ao revés, o aumento da temperatura diminui de forma significativa o valor da tensão gerada pelo módulo e acarreta uma pequena elevação no valor da corrente.

O modelo matemático utilizado para descrever uma célula fotovoltaica é baseado no seu circuito equivalente - modelo de um diodo [CMOD04]. O circuito é representado pela corrente fotogerada (I_L), resistências em série (R_s), resistência em paralelo (R_p), tensão (V) e corrente (I) de saída da célula, conforme visto pela Figura 3.3.

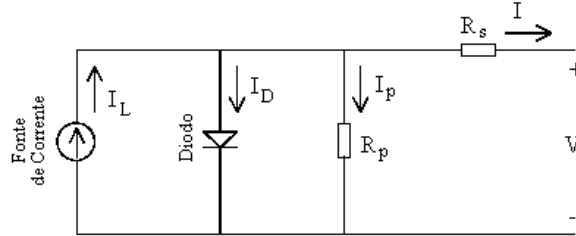


Figura 3.3: Circuito equivalente de uma célula fotovoltaica real (modelo de um diodo).

A equação que descreve o desempenho de geradores fotovoltaicos, relacionando tensão, corrente, intensidade solar e temperatura, é dada pela Fórmula 3.34:

$$I = I_{sc} \left\{ 1 - C_1 \left[\exp \left(\frac{V - \Delta V}{C_2 V_{oc}} \right) - 1 \right] \right\} + \Delta I \quad (3.34)$$

sendo:

$$C_2 = \frac{V_{mp}/V_{oc} - 1}{\ln(1 - I_{mp}/I_{sc})}$$

$$C_1 = \left(1 - \frac{I_{mp}}{I_{sc}} \right) \exp \left(- \frac{V_{mp}}{C_2 V_{oc}} \right)$$

$$\Delta I = \alpha \left(\frac{S_m}{S_{mref}} \right) \Delta T + \left(\frac{S_m}{S_{mref}} - 1 \right) I_{sc}$$

$$\Delta V = -\beta \Delta T - R_s \Delta I$$

$$\Delta T = T - T_{ref}$$

$$T = T_a + 0,02 S_m$$

onde:

V → tensão de saída do circuito;

V_{oc} → tensão de circuito aberto;

V_{mp} → tensão de máxima potência;

V_{ref} → tensão nas condições de referência;

$I \rightarrow$ corrente de saída do circuito;

$I_{sc} \rightarrow$ corrente de curto-circuito;

$I_{mp} \rightarrow$ corrente de máxima potência;

$I_{ref} \rightarrow$ corrente nas condições de referência;

$\alpha \rightarrow$ coeficiente de temperatura para a corrente de curto-circuito na irradiância solar de referência;

$\beta \rightarrow$ coeficiente de temperatura para a tensão de circuito aberto na irradiância solar de referência;

$R_s \rightarrow$ resistência série;

$S_m \rightarrow$ irradiância solar global no plano do gerador fotovoltaico (W/m^2);

$S_{mref} \rightarrow$ irradiância solar de referência ($1000W/m^2$);

$T \rightarrow$ temperatura da célula solar;

$T_a \rightarrow$ temperatura ambiente;

$T_{ref} \rightarrow$ temperatura de referência da célula solar ($25^\circ C$).

Novas tensões e correntes são calculadas a partir da tensão e da corrente nas condições de referência, sendo que estes novos valores de V e I serão dados de entrada na Equação 3.34 e as expressões são respectivamente, dadas por:

$$V = V_{ref} + \Delta V \quad (3.35)$$

$$I = I_{ref} + \Delta I \quad (3.36)$$

Para a determinação da resistência em série, utilizou-se a relação entre a potência, tensão e corrente, conforme mostrado na Fórmula 3.37. A potência de saída do módulo fotovoltaico é um produto entre sua tensão e a corrente de saída.

$$PP = VI \quad (3.37)$$

em que:

$PP \rightarrow$ potência de saída do gerador fotovoltaico;

$V \rightarrow$ tensão de saída do gerador fotovoltaico;

$I \rightarrow$ corrente de saída do gerador fotovoltaico.

De posse do valor da resistência em série, fornecido através da Equação 3.38, pode-se substituí-lo na Fórmula 3.34, que descreve o desempenho de geradores fotovoltaicos, permitindo então, simular a curva característica $I \times V$.

$$R_s = \frac{dS_{mref}}{bS_m(\alpha T - \alpha T_{ref} + I_{sc})} \quad (3.38)$$

sendo

$$d = V_{mp}b + cV_{mp} - bV - b\beta T + b\beta T_{ref} - cV_{oc}$$

$$b = \ln\left(\frac{I_{sc} - I_{mp}}{I_{sc}}\right)$$

$$c = \ln\left(\frac{-P + VI_{sc} + VaI_{sc} - VaI_{mp}}{V(I_{sc} - I_{mp})}\right)$$

$$a = \exp\left(-\frac{V_{mp} \ln\left(\frac{I_{sc} - I_{mp}}{I_{sc}}\right)}{V_{mp} - V_{oc}}\right)$$

$P \rightarrow$ máxima potência.

Os valores de V_{oc} , I_{sc} , V_{mp} , I_{mp} , α e β são fornecidos pelos fabricantes na especificação dos módulos, sendo utilizados no processo de simulação.

3.6 Bateria

Além dos geradores fotovoltaicos, o sistema de armazenamento de energia é outro componente de vital importância em um sistema fotovoltaico isolado. As baterias são uma conveniente forma de armazenamento da energia elétrica gerada pelos módulos. Atualmente, as baterias utilizadas são as produzidas de chumbo-ácido e níquel-cádmio, sendo que, para a maioria dos sistemas fotovoltaicos, instala-se as de chumbo-ácido, dando-se preferência para as abertas nos grandes sistemas e as seladas para os sistemas menores ⁴.

Os fatores de maior destaque no desempenho, capacidade e vida útil das baterias recarregáveis, são: profundidade de descarga a que são submetidas a cada ciclo, temperatura em que operam, número de ciclos e controle de carga e descarga, além da manutenção

⁴Diferenciação quanto à forma de confinamento do eletrólito - *Abertas* \rightarrow O eletrólito é líquido e livre e necessitam de verificação periódica do nível de água; *Seladas* \rightarrow Também conhecidas como "sem manutenção", porque não necessitam de adição de água e o eletrólito está confinado no separador.

periódica. Cabe ressaltar que as unidades de armazenamento se descarregam mesmo sem a utilização no arranjo fotovoltaico [SOL99].

Pode-se destacar o carregamento como uma operação crítica, que afeta diretamente o tempo de vida útil de uma bateria, ou seja, se existe um sistema de controle de carga eficiente, evita-se danos à bateria, causados pelo carregamento excessivo ou descarga prolongada. Logo, faz-se necessário conhecer os parâmetros que regem o sistema de armazenamento, a resistência interna e os coeficientes de variação da tensão e temperatura.

A demanda de carga e seus valores máximo e mínimo de profundidade de descarga devem ser quantificados nas simulações e são de grande importância para os dimensionamentos determinísticos. O funcionamento da unidade de armazenamento pode ser prejudicado caso o ciclo de carga alcance o valor máximo ou o ciclo de descarga ultrapasse o valor mínimo permitido [CMOD04].

Para o processo de simulação, parte-se do pressuposto que na primeira hora analisada a unidade de armazenamento esteja em sua carga máxima e esse valor deve ser fornecido através de uma estimativa realizada pelo fabricante, considerando a condição padrão de temperatura e os valores de tensão e corrente para um período específico. Além da carga inicial, os valores de profundidade de descarga e eficiência também devem ser fornecidos e são representados por:

$E_B \rightarrow$ energia armazenada nas baterias;

$E_{B\min} \rightarrow$ nível mínimo de descarga permitido;

$E_{B\max} \rightarrow$ nível máximo de carga permitido;

$\eta_{bat} \rightarrow$ eficiência das baterias.

O processo de carga e descarga nas baterias, durante uma hora t , deve respeitar à seguinte restrição:

$$E_{B\min} \leq E_B \leq E_{B\max}$$

Esta restrição é controlada pelo componente controlador e suas características serão descritas a seguir.

3.7 Controlador de Carga

O controlador de carga tem como principal objetivo facilitar a transferência de energia do arranjo fotovoltaico para o banco de baterias, protegendo contra cargas e descargas excessivas e, conseqüentemente, aumentando o tempo de vida útil do sistema.

Em um sistema fotovoltaico as baterias possuem uma relação direta com o controlador, ou seja, caso ocorra alguma falha, danos irreversíveis podem ser causados. Por sua vez, o controlador de carga depende do tipo de bateria a ser utilizado no arranjo, em função da eficiência de carga depender das especificidades de cada tipo [SOL99].

Este componente desconecta o módulo fotovoltaico quando a bateria atinge o seu nível máximo de carga permitido e ativa o fornecimento quando o estado de carga atingir o nível mínimo de segurança na descarga. Aumenta-se o desempenho do controlador de carga quando há um sensor de temperatura acoplado a ele, permitindo compensar o efeito da variação da temperatura nos parâmetros da bateria.

Para realizar uma escolha do controlador que melhor se adeqüe ao sistema fotovoltaico, devem-se analisar alguns aspectos do arranjo, como: a tensão do sistema, a corrente na carga e gerador fotovoltaico, o tipo e dimensão da bateria, os pontos de ajuste de regulação e desconexão da carga, as condições de operações ambientais, os indicadores do sistema, alarmes e medidores, os dispositivos de proteção e o custo de instalação.

O modelo matemático que descreve o funcionamento de um controlador de carga é descrito a seguir e leva em consideração as duas situações de profundidade de carga e descarga das baterias [CMOD04]:

$$E_B(t) = E_B(t - 1) + \left(E_G(t) - \frac{E_L(t)}{\eta_{inv}} \right) \eta_{bat} \quad (3.39)$$

$$E_B(t) = E_B(t - 1) - \left(\frac{E_L(t)}{\eta_{inv}} - E_G(t) \right) \quad (3.40)$$

em que:

$E_B(t)$ → energia armazenada nas baterias durante a hora t ;

$E_B(t - 1)$ → energia armazenada nas baterias durante a hora $t - 1$;

$E_G(t)$ → energia gerada pelo módulo fotovoltaico durante a hora t ;

$E_L(t)$ → energia demandada pela carga durante a hora t ou simplesmente consumo;

Após aplicar a restrição de carga e descarga da bateria, o controlador carrega até o limite máximo de eficiência de carga, se a energia gerada pelo arranjo fotovoltaico exceder a demandada pela carga. Este processo é demonstrado pela Fórmula 3.39. Caso contrário, se a energia demandada pela carga for maior que a energia gerada pelos módulos, as baterias serão descarregadas na quantidade necessária para suprir o déficit, conforme Equação 3.40.

3.8 Inversor

O inversor, também conhecido como conversor CC-CA (corrente contínua - corrente alternada), estabelece uma conexão entre o gerador fotovoltaico e a rede CA. A sua principal tarefa consiste em converter o sinal elétrico CC do gerador fotovoltaico em um sinal elétrico CA, e ajustá-lo para a frequência e o nível de tensão da rede a que está ligado. Ele usa um mecanismo de chaveamento para alterar o fluxo de corrente entre as direções positiva e negativa.

Para especificar um inversor é necessário considerar a tensão de entrada CC e a tensão de saída CA. A eficiência é dada pela relação entre a potência de entrada e a potência de saída, variando na faixa de 50% a 90% [SOL99]. O valor percentual de eficiência, geralmente especificado pelo fabricante, é o valor máximo, que pode ser obtido por esse componente. Seu dimensionamento é representado por η_{inv} e pode ser utilizado na modelagem matemática, conforme demonstrado pelas Equações 3.39 e 3.40. Deve-se, também, levar em consideração para o dimensionamento dois fatores: a tensão de entrada que deverá alimentar o componente por um determinado período de tempo e o a potência nominal de saída.

3.9 Comentários finais

Este capítulo descreveu os principais conceitos do processo de transformação de energia solar em elétrica. Um breve histórico sobre o processo fotovoltaico foi também apresentado. Em seguida, discutida a metodologia que caracteriza os dados climatológicos de um determinado local, utilizando informações de anos passados. Foram apresentados os procedimentos para geração de um ano de referência, também denominado ano padrão, baseado nos métodos propostos por Festa e Ratto e sua extensão realizada por Argiriou *et al*, além do procedimento de suavização de curvas proposta por Festa *et al*.

Posteriormente, foram vistos os conceitos de estimação da radiação solar horária sobre uma superfície inclinada para o dimensionamento determinístico de sistemas fotovoltaicos isolados.

E por fim, foram detalhados os métodos matemáticos para o entendimento da modelagem do arranjo fotovoltaico e seus principais componentes.

Cabe ressaltar que todos os métodos, conceitos e modelos matemáticos propostos foram implementados e são integrantes dos componentes do *framework*, que será apresentado no Capítulo seguinte.

Capítulo 4

Framework Solar Energy

4.1 Considerações iniciais

Conforme mencionado na Seção 2.2, *framework* de *software* compreende um conjunto de classes e interfaces, implementadas em uma linguagem específica, usadas para auxiliar no desenvolvimento de aplicações de um mesmo domínio.

Nesta dissertação propõe-se a utilização do *framework* orientado a objetos *Solar Energy*, desenvolvido por *Guidoni et al* [GPS⁺08], que permite ao desenvolvedor de sistemas criar aplicativos relacionados ao processo de simulação dinâmica de aquecimento de água por energia solar.

Foram aplicadas inovações e evoluções que possibilitaram a criação de sistemas para o dimensionamento e simulações do processo de conversão de luz solar em energia elétrica. Este *framework* tem como objetivo principal encapsular as regras de negócio das aplicações, controlar a troca de informações entre os objetos e abstrair os conceitos técnicos envolvidos na construção e manipulação dos componentes dos sistemas térmicos e fotovoltaicos. Além disto, prover interfaces gráficas capazes de facilitar o manuseio e a operacionalização dos *softwares* derivados.

A Figura 4.1 demonstra, através de um fluxograma, as cinco fases utilizadas para a evolução do *framework* e do aplicativo *Sol Brasil*. As seguintes etapas foram propostas e utilizadas pelo pesquisador no desenvolvimento deste trabalho:

1. Análise dos sistemas fotovoltaicos e a relação existente entre o processo térmico, descrevendo dessa forma, a cobertura realizada pelo *framework*, além de captação dos principais pontos variáveis, resultando em um modelo abrangente dos módulos que utilizam energia solar;
2. Estudo da arquitetura atual e proposta de inovações e evoluções nos principais

conceitos da estrutura, baseado no modelo de requisitos fotovoltaicos. Destaca-se nesta fase a utilização dos padrões de projeto, que auxiliam no desenvolvimento das principais mudanças arquiteturais da ferramenta;

3. Refinamento, resultando no projeto do *framework*;
4. Implementação, através da linguagem orientada a objetos *Java*, por meio da plataforma *NetBeans IDE 5.5*;
5. Manutenção na aplicação *Sol Brasil*, para a inclusão do processo de dimensionamento determinístico e simulação fotovoltaica, com o intuito de realizar testes tanto no *Solar Energy* quanto no aplicativo derivado.

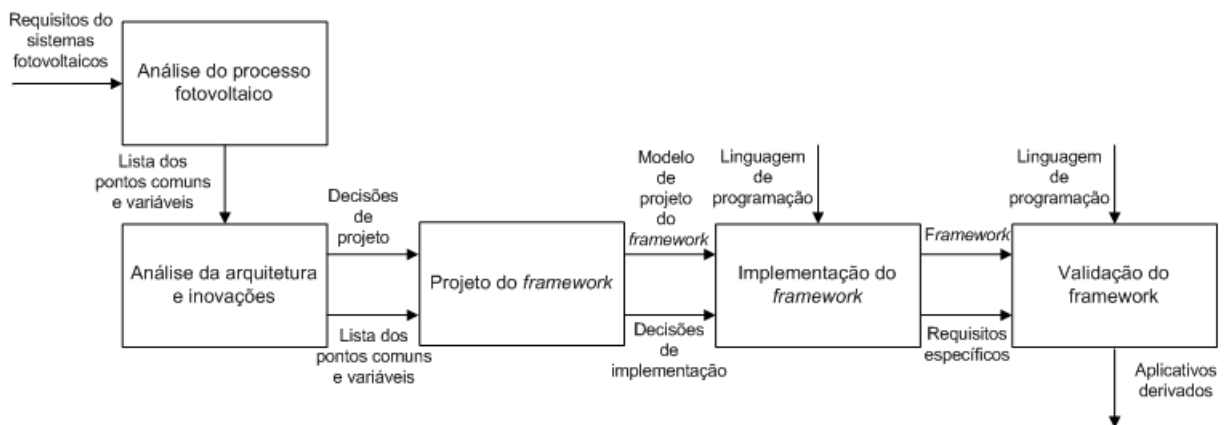


Figura 4.1: Fluxograma para construção e evolução do *Solar Energy*.

O restante deste capítulo descreve de forma detalhada as fases aplicadas para a construção do *Solar Energy* e está organizado da seguinte forma: A Seção 4.2 descreve os principais requisitos funcionais dos sistemas fotovoltaicos. A Seção 4.3 apresenta com mais detalhes a estrutura original do *framework* e as principais evoluções e inovações para permitir a construção de aplicações fotovoltaicas. As Seções 4.4, 4.5 e 4.6 abordam, respectivamente, de forma sucinta, os artefatos gerados para o refinamento do projeto, o processo de implementação e a forma de validação da ferramenta. Por fim, a Seção 4.7 apresenta os comentários finais e encerra o capítulo.

4.2 Requisitos de aplicativos fotovoltaicos

Para a realização da primeira fase de construção do *Solar Energy*, deve-se realizar a análise dos requisitos funcionais para a geração de energia elétrica e a relação com os requisitos relacionados ao aquecimento de água por meio da luz solar.

A identificação dos pontos variáveis e do conjunto de regras e funções equivalentes entre as aplicações térmicas e fotovoltaicas consiste na utilização das informações presentes na lista de requisitos levantada. O objetivo desta análise é encontrar os pontos comuns e as funcionalidades que não sejam usadas por outras aplicações pertencentes ao mesmo domínio.

Com base na avaliação dos sistemas de dimensionamento e simulação fotovoltaica definem-se as seguintes premissas a serem atendidas:

- **Adicionar novos componentes e funcionalidades:** Capacidade de incluir novos componentes e interfaces gráficas utilizando as características dos componentes e classes já criados;
- **Gerenciar dados climatológicos:** Permitir a carga, a exclusão e a alteração de dados climatológicos, previamente medidos, para o dimensionamento da radiação solar e outras variáveis climáticas, responsáveis pela identificação do ano de referência. As informações são relacionadas a uma determinada localidade/cidade;
- **Relacionar componentes:** Realizar a conexão entre os componentes do arranjo fotovoltaico e transferir as informações necessárias;
- **Dimensionar o sistema fotovoltaico:** Construir um diagrama, que represente, de forma gráfica, o funcionamento real de um módulo fotovoltaico, gerenciando as características e propriedades de todos os componentes envolvidos no processo;
- **Simular o sistema fotovoltaico:** Analisar a representação gráfica, controlando a ordem de execução e a troca de informações entre os componentes;
- **Gerar resultados estatísticos:** Possibilitar a verificação estatística do dimensionamento e da simulação tanto de forma sintética, quanto de forma analítica, por intermédio de relatórios.

A Figura 4.2 ilustra os principais requisitos funcionais, através do modelo de casos de uso. Deve-se levar em consideração dois princípios básicos nas simulações e dimensionamentos fotovoltaicos: a capacidade do módulo captar e armazenar luz solar e a capacidade de

transferir e utilizar essa energia por parte dos dispositivos que compõem o arranjo fotovoltaico. Estas características definem o forte acoplamento entre as aplicações térmicas e fotovoltaicas possibilitando a evolução do *Solar Energy* para se tornar um arcabouço completo em energia solar.

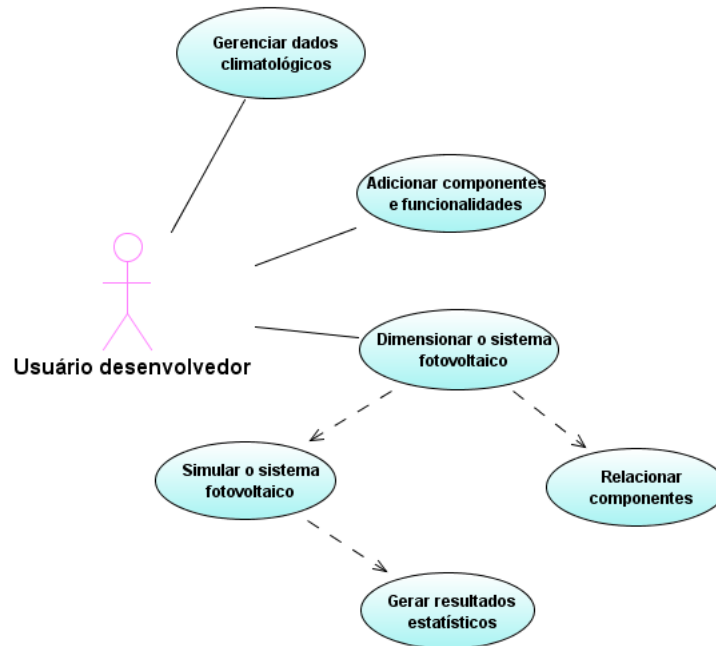


Figura 4.2: Modelo de casos de uso para um sistema fotovoltaico.

Os aspectos não funcionais das aplicações, como portabilidade, usabilidade, segurança e confiabilidade também devem ser considerados, uma vez que podem originar novos pontos variáveis. Também devem ser avaliados os aspectos de projeto e implementação que possam trazer maior flexibilidade ao *framework*, aumentando seu potencial de reuso.

4.3 Arquitetura

A partir da identificação dos pontos comuns em funcionalidades e dos aspectos únicos da geração fotovoltaica, fornecidos pelo levantamento de requisitos, aplicou-se a fase dois da metodologia. Nesta etapa foi analisada a estrutura do *framework* e incorporadas novas implementações. Cabe ressaltar que o forte acoplamento entre as aplicações do domínio solar permitiu um maior aproveitamento da arquitetura do *Solar Energy*. Entretanto, melhorias e evoluções foram aplicadas permitindo que as funcionalidades do *framework* se tornem mais abrangentes conforme descrito posteriormente.

4.3.1 Estrutura inicial

Para atender as premissas das aplicações térmicas, dividiu-se o *Solar Energy* em oito pacotes. A Figura 4.3 apresenta o diagrama inicial de pacotes do *framework*.

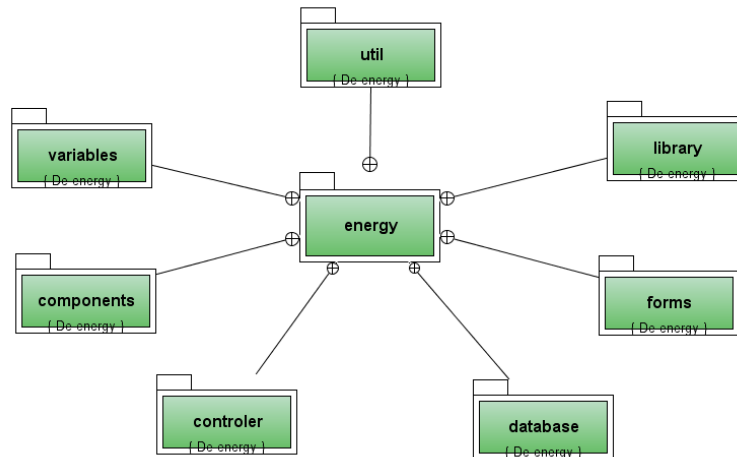


Figura 4.3: Diagrama inicial de pacotes do *Solar Energy*.

O pacote *Energy* foi definido como principal da estrutura. É composto pelas classes e interfaces responsáveis em aplicar a característica de inversão de controle dos aplicativos, além de incorporar todos os outros sete pacotes, que disponibilizam as funcionalidades necessárias ao desenvolvimento dos *softwares* derivados. As principais responsabilidades das classes que o compõem são:

- Gerar uma instância do *Solar Energy*;
- Incluir componentes;
- Adicionar interfaces gráficas;
- Manipular e exibir os componentes utilizados no dimensionamento e simulação;
- Aplicar e executar as regras de negócio envolvidas no processo de geração e transformação de energia pela luz solar;
- Fornecer dados da radiação solar através dos métodos de estimativa implementados;
- Garantir a correta organização e fluxo de execução dos dispositivos solares;
- Realizar a transferência de informações entre os componentes.

Este pacote é responsável por disponibilizar os *hot-spots*, que permitem inferir no comportamento dos aplicativos em função das necessidades dos usuários. Ressalta-se que o *framework* não permite que os componentes se comuniquem diretamente. Toda comunicação é realizada pelos respectivos pontos de conexão presentes na super-classe *EMCObject*. Os pontos representam as entradas e saídas das informações processadas por cada componente e um dispositivo pode conter vários pontos de conexão. Todos representados pela classe *EMCPoint*.

A interface *IRadiation*, também presente neste pacote, é responsável por manipular e fornecer dados da energia solar. Uma diferença essencial na classe que implementa esta interface é o fato de não participar do ciclo de execução da simulação.

A estrutura da ferramenta permite que todos os componentes disponibilizados sejam herdados da classe *EMCComponent*, presente no pacote *Components*. Gerar sub-classes a partir desta super-classe garante que os métodos essenciais para a execução dos processos sejam invocados dentro das configurações definidas para a classe ancestral.

A função principal de cada componente é receber informações de radiação, processá-las e transferi-las aos outros dispositivos.

Dentre os métodos existentes destacam-se os de preparação para a execução, definido como `prepareToRun()`, os de validação dos parâmetros de entrada, `validateData()` e os de processamento dos dados `calculate()`.

Os dados processados da radiação solar devem ser transferidos para todos os demais componentes presentes no diagrama de dimensionamento e simulação, antes de invocar o método de execução `calculate()`. Para que o *Solar Energy* reconheça as variáveis foi desenvolvida uma estrutura denominada *radiationVariables*, presente no sub-pacote *Variables*. Após a execução dos métodos de estimativa da radiação, os dados são adicionados a esta estrutura e distribuídos em função da necessidade.

Um pacote de fundamental importância para a estrutura do *framework* é o *Controler*. Este pacote, representado pelo diagrama da Figura 4.4, contém as classes cuja a responsabilidade está em controlar e executar o processo de simulação.

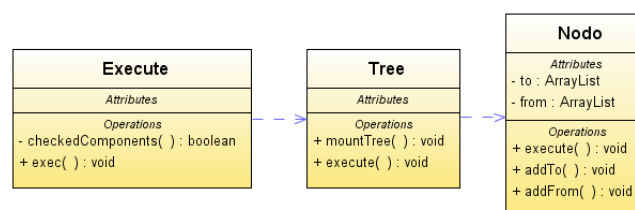


Figura 4.4: Diagrama contendo as classes responsáveis pela execução da simulação.

Os diagramas de execução são representados como grafos, onde cada objeto nodo (vértice) representa um dispositivo físico e as arestas equivalem aos pontos de conexão. Um caminhamento em profundidade representa a execução dos componentes.

Faz-se necessário a utilização de um banco de dados para gerenciar as informações relevantes ao desenvolvimento das aplicações. Logo, o pacote *Database* foi incorporado ao *framework* para prover as inclusões, alterações e exclusões de dados. Não faz parte das atribuições da ferramenta realizar a persistência das informações.

O *SolarEnergy* disponibiliza classes para o desenvolvimento de novas interfaces gráficas ou mesmo para a substituição da classe principal de um sistema derivado. Este conjunto de funcionalidades estão disponíveis no pacote *Forms*.

O pacote *Library* representa uma biblioteca de classes para os cálculos utilizados pelo *framework*. Estas classes são especializadas para realizar o processamento matemático dos dispositivos físicos e também dos métodos de estimativa da radiação. Desta forma, cria-se a flexibilidade de utilização e implementação de outros métodos presentes na literatura especializada em energia solar.

A seção seguinte descreve as inovações na ferramenta para tornar a infra-estrutura genérica em soluções para um conjunto de problemas relacionado ao domínio tratado.

4.3.2 Evoluções aplicadas

Visando agregar conhecimento ao *Solar Energy*, foram propostas evoluções em sua estrutura e em suas funcionalidades. Permite-se, com isso, criar as condições necessárias para que o desenvolvedor aplique o reuso de *software* e conseqüentemente melhore a produtividade, a qualidade, a manutenibilidade e obtenha maior flexibilidade nos aplicativos solares derivados. A seguir são descritas as inovações aplicadas ao *framework*, a partir da lista de requisitos dos aplicativos fotovoltaicos e definições dos principais pontos variáveis:

Hot Spot ambiente

Todo processo de simulação, seja térmico ou fotovoltaico, tem como fonte de dados a radiação solar. Esta energia é enviada aos componentes captadores, processada, manipulada e distribuída para os demais componentes presentes no diagrama de simulações.

Os requisitos para o aquecimento de água determinam que o processo de captação e transformação de energia sejam realizados a partir dos cálculos dos dados de insolação para uma determinada localidade ou região.

Por sua vez, para estimar a radiação solar no processo fotovoltaico, por meio de uma abordagem diferenciada, fez-se necessário o processamento de dados climatológicos me-

dados para uma cidade de referência, conforme visto na Seção 3.4.1. Para atender a esta premissa, uma mudança significativa no conceito do componente responsável em fornecer a luz solar foi realizada, resultando na criação do ponto variável ambiente, denominado *IEnvironment*. Esta interface possui os métodos que gerenciam o ambiente solar, como os relacionados aos cálculos de estimativa da radiação, definido por `calculate()`, os métodos para recuperação e manipulação do ambiente, além dos valores climatológicos necessários ao fluxo de execução dos processos, representado por `getRadiationVariables()` e `setEnvironment()`, respectivamente.

Conforme visto em seções anteriores, a identificação de um *hot spot* consiste em obter pontos adaptáveis, que necessitem ser especializados para um determinado escopo de aplicação e, para que o *framework* seja definido como uma arquitetura de *software* reutilizável, é necessário que haja colaboração entre as classes abstratas e o conjunto de pontos adaptáveis. Logo, a criação do *hot spot IEnvironment* exemplifica estes conceitos, porque permite que a manipulação dos dados solares se torne adaptável em função das definições feitas pelo usuário desenvolvedor para o ambiente da radiação, inicialmente, especializado para o aquecimento de água e, posteriormente, adaptado para a inclusão dos processos fotovoltaicos.

Responsabilidades comuns e tempo de ligação são duas características importantes para a definição do tipo de implementação de um *hot spot*. A primeira corresponde aos serviços que qualquer classe, que implemente o ponto variável, deva oferecer para o restante do *framework*. Já a segunda, define o momento em que as características do *hot spot* são selecionadas e associadas. Esta ligação pode ser feita pelo desenvolvedor durante a personalização do *framework* ou pelo usuário durante a execução da aplicação. Visando aplicar essas duas características, a interface *IEnvironment* foi implementada através da classe *EGYEnvironment*, que disponibiliza os serviços necessários ao controle e gerência do ambiente de simulação, bem como define o momento de acesso e manipulação a estes dados, cuja importância se aplica ao processo de execução.

A Figura 4.5 apresenta o ponto flexível *IEnvironment* e também a classe que implementa os principais conceitos e funcionalidades deste *hot spot*. O trecho de código a seguir ilustra a instanciação do objeto ambiente pelo *Solar Energy*. Por aplicar a inversão de controle, o *framework* é o responsável por criar este novo objeto.

```
public IEnvironment show(){
    if (environment == null)
        environment = EGYEnvironment.getInstanceOf();
    ...
}
```

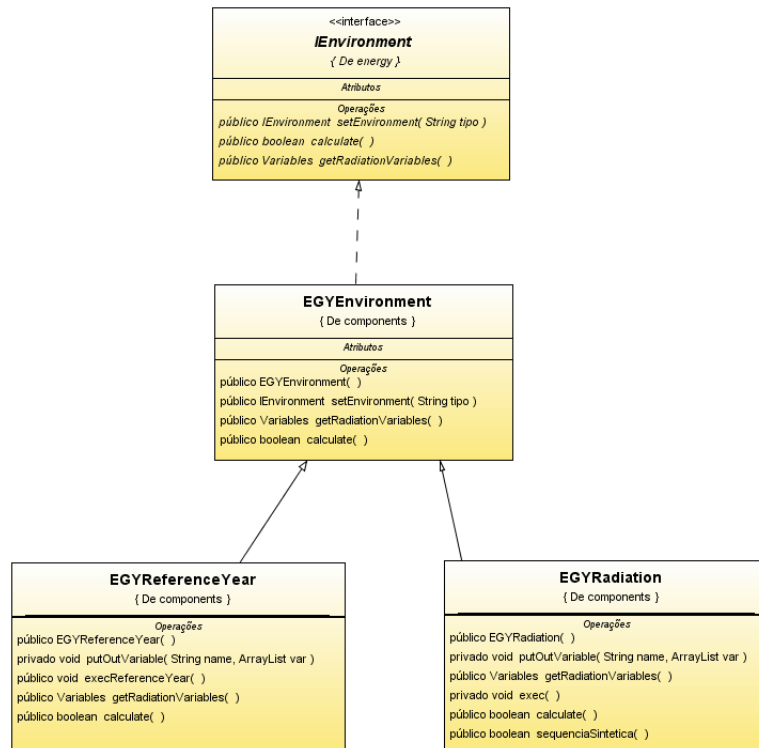


Figura 4.5: Diagrama de classes do *hot spot IEnvironment*.

Uma vez o ambiente criado, o usuário têm a opção de escolher qual a forma de estimar a radiação global média horária no plano inclinado, tendo como opções o cálculo pelos dados de insolação ou os dados do ano de referência, que foi obtido pela análise estatística dos dados climatológicas fornecidos.

Os métodos responsáveis em realizar estes cálculos foram agrupados em duas sub-classes herdadas da super-classe *EGYEnvironment* e definidas como *EGYRadiation* e *EGYReferenceYear*. Uma diretiva de projeto juntamente com uma interface gráfica auxiliar, disponível no *framework*, são responsáveis em recuperar e passar o controle ao ambiente escolhido.

O código seguinte é responsável por receber esta diretiva, definida pela variável `tipo` e instanciar, por uma das duas sub-classes, o objeto cujas funcionalidades representam a estimativa de radiação.

```

public IEnvironment setEnvironment(String tipo){
    environment = EnvironmentFactory.getInstanceOf(tipo, getEnvironmentGen());
    return environment;
}
  
```

O *hot spot IEnvironment* possibilita também, que novas formas de obtenção de energia possam ser agregadas ao *Solar Energy*. Para tal, basta ao desenvolvedor implementar uma sub-classe herdada de *EGYEnvironment*, informando uma instância desta nova classe.

Hierarquias de interfaces

Visando auxiliar a reutilização de código foi estabelecida uma hierarquia de interfaces. Para ilustrar esta evolução, em relação a estrutura inicial do *framework*, foi criada a interface *IPickupEnergy*. As sub-interfaces *ISolarPanel* e *ICollector* especializam, respectivamente, as funcionalidades dos dispositivos painel e coletor solar.

Outro exemplo que aplica o mesmo conceito é a super-interface *IStore* e as sub-interfaces *IReservoir* e *IBattery*. A implementação destas classes resultam, respectivamente, nos dispositivos reservatório e bateria, utilizados nos sistemas térmicos e fotovoltaicos.

Deve-se ressaltar que as super-interfaces também estendem as características da interface *IComponent*, definindo assim, uma estrutura única para qualquer componente criado e utilizado nos dimensionamentos e simulações.

Identificação de ciclos de execução

O pacote *Controler* utiliza a teoria de grafos para o dimensionamento das aplicações que envolvam tanto o aquecimento de água, quanto a conversão em energia elétrica e faz uso da heurística de busca em profundidade para a execução das simulações. Desta forma, um dispositivo, definido como nó raiz, é processado e todos os seus vizinhos próximos explorados. Cada um desses novos nós são processados e explorados, sucessivamente, até que todos os componentes tenham executado os cálculos necessários ao processamento no diagrama de simulação. Os componentes captadores de luz solar terão sempre a maior prioridade de execução sobre os demais dispositivos.

Conforme visto nas Seções 3.6 e 3.7 os dispositivos fotovoltaicos, bateria e controlador de carga, são dependentes no processo de execução. Esta forte relação deve-se ao fato da unidade de armazenamento ser uma fonte auxiliar de energia para os sistemas fotovoltaicos isolados e o controlador de carga realizar a transferência de energia, produzida pelo módulo, aos demais dispositivos, incluindo o banco de baterias e a demanda consumida. Caso a energia gerada pelo arranjo fotovoltaico não seja suficiente para alimentar o consumo, o controlador de carga utiliza a energia armazenada nas baterias. Caso contrário, o controlador disponibiliza o restante da potência gerada a toda a unidade de armazenamento de energia.

Em função desta premissa de dependência, foram identificados ciclos de execução no grafo que representa a simulação, onde o vértice inicial é o controlador de carga e todo o banco de baterias corresponde aos vértices que fecham o ciclo.

O grafo completo é representado por uma instância da classe *Tree* e os vértices, que correspondem os dispositivos, são gerados a partir da classe *Node*. Dois atributos, presentes nestes objetos e definidos por `visited` e `executed`, são responsáveis em auxiliar no caminhamento do ciclo. Há também o método responsável por realizar a análise de precedências, denominado `precedence()`.

A Figura 4.6 representa o grafo de execução da simulação fotovoltaica. O diagrama é composto por um painel solar, uma bateria, um controlador de carga e um inversor de corrente. Deve-se ressaltar que os componentes de transferência, cuja responsabilidade está em conectar e transferir as informações processadas para todos os dispositivos, são representados por *ICconnection* e correspondem a um vértice no grafo.

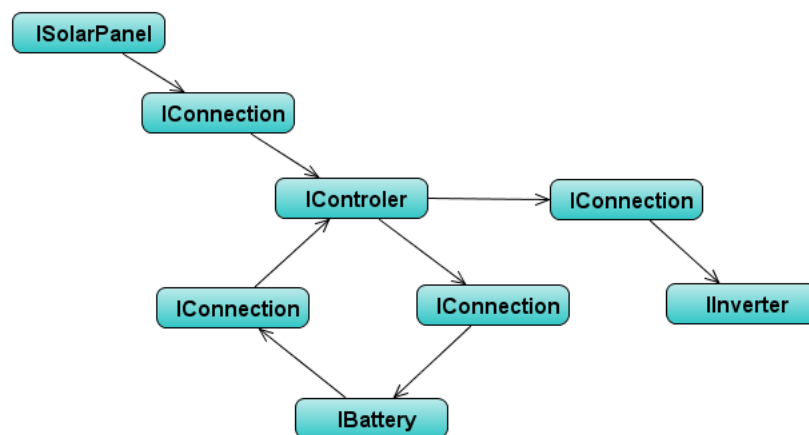


Figura 4.6: Grafo de execução da simulação fotovoltaica.

O diagrama, representado pela Figura 4.7, apresenta a seqüência de execução dos métodos no processo de simulação fotovoltaica. As classes *Execute* e *Tree* são responsáveis por montar o grafo de execução, realizar as consistências necessárias da simulação, recuperar os dados de radiação e iniciar a etapa de caminhamento no grafo. Cada iteração invoca os métodos correspondentes aos componentes, representados pelas classes *Node* e *Component*, na ordem que estão apresentados no diagrama.

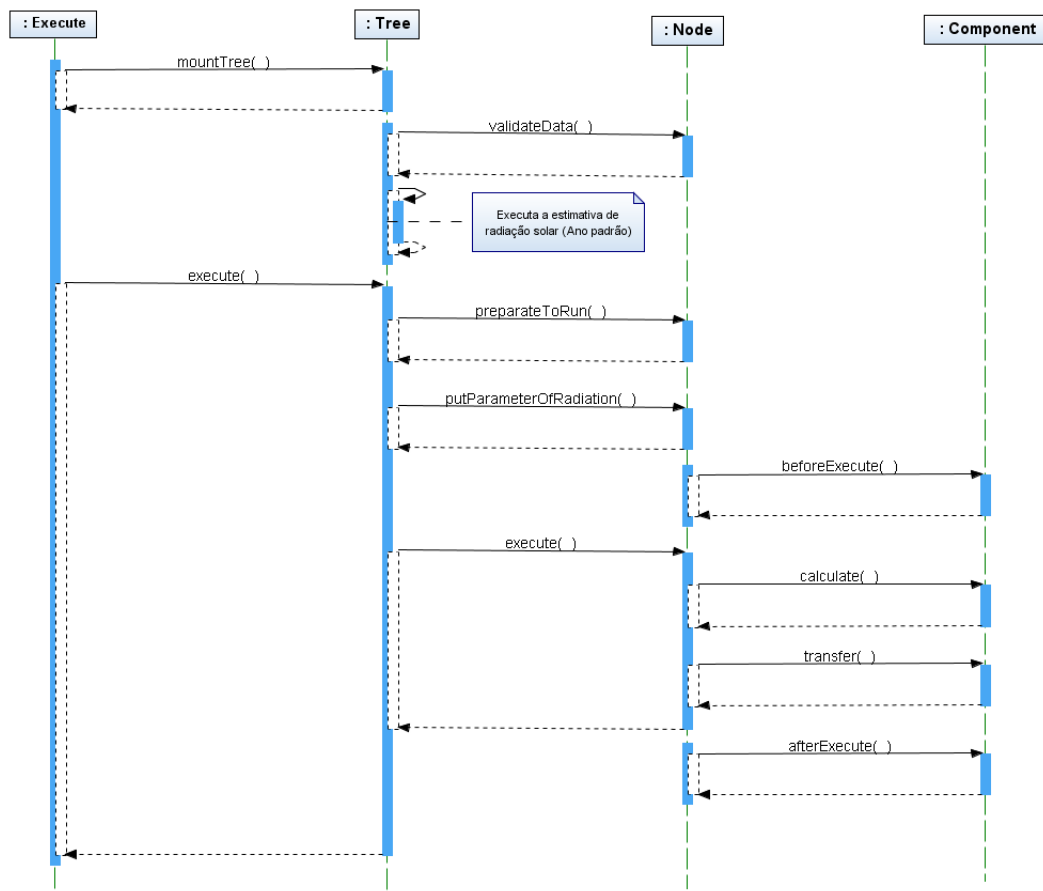


Figura 4.7: Diagrama de seqüência de execução do processo de simulação fotovoltaica.

Aplicação dos padrões de projeto

Um padrão de projeto nomeia, abstrai e identifica os aspectos chaves de uma estrutura comum para torná-la útil para a criação de um projeto orientado a objetos reutilizável. Por essa razão, foram usados no desenvolvimento do *framework* alguns padrões conforme a classificação necessária. Em geral, os padrões de projeto podem ser classificados em três diferentes tipos: de criação, que são utilizados para abstrair o processo de criação de objetos a partir da instanciação de classes; estruturais, que tratam a forma como as classes e os objetos são organizados; e comportamentais, relacionados à atribuição de responsabilidades entre os objetos [GHEV93]. A seguir são apresentados os padrões de criação que contribuíram para a evolução do *Solar Energy*:

- **Prototype:** este padrão tem como objetivo criar os componentes específicos a partir da instância de um protótipo e então, criar novos componentes através da cópia deste protótipo. A aplicação do padrão para um dispositivo cria outro com as mes-

mas funcionalidades, porém, com propriedades distintas. A vantagem de utilização está na possibilidade de adicionar ou excluir os componentes de um diagrama em tempo de execução, incorporando um novo dispositivo que esteja em desenvolvimento. Há também a possibilidade de se definir novos comportamentos através da composição de objetos, ou seja, de especificar valores para as variáveis de um dispositivo sem a necessidade de definir novas classes. Outra vantagem está na redução do número de sub-classes, pois ao invés de ser criada uma estrutura de hierarquia de classes para especializar os componentes, pode-se clonar um objeto protótipo. Em função da definição de uma estrutura única para representar qualquer dispositivo nos diagramas de execução do *framework*, o *Prototype* foi aplicado na super-classe *EMCComponent* e a implementação é mostrada pelo código a seguir

```
public Object clone() {
    return new EMCComponent(this);
}
```

No momento que um objeto do mesmo tipo seja adicionado ao diagrama, é executado o método descrito e gerado um novo dispositivo, onde seus atributos são definidos pelo construtor da classe que instancia o objeto informado.

- **Factory Method:** O *hot spot IEnvironment* faz uso deste padrão, denominado fábrica, que estabelece uma forma de desenvolver objetos responsáveis pela criação de outros objetos. Ele foi usado para fornecer uma interface do ambiente para a criação de formas correlatas ou dependentes sem a necessidade de especificar a classe concreta, ou seja, foi usado para criar um ambiente que implemente a interface *IEnvironment* de diferentes formas. É de responsabilidade da fábrica selecionar a implementação desejada. A principal vantagem na utilização do padrão *Factory* está na possibilidade de criar objetos ambiente, dinamicamente, sem ter a necessidade de conhecer a classe de implementação, somente a interface. Os trechos de códigos a seguir, exemplificam o padrão utilizado no *hot spot* e a Figura 4.8 mostra o diagrama de classes correspondente.

A interface que define o método `calculate()` é a *IEnvironment*.

```
public interface IEnvironment{
    public boolean calculate();
    ...
}
```

A classe *EGYRadiation* implementa a interface *IEnvironment* e realiza a estimativa da radiação solar através dos cálculos dos dados de insolação.

```

public class EGYRadiation extends EGYEnvironment {
    ...
    public boolean calculate(){
        boolean result = true;
        if ( verifica() )
            execRadiation();
        else result = false;
        return result;
    }
    ...
}

```

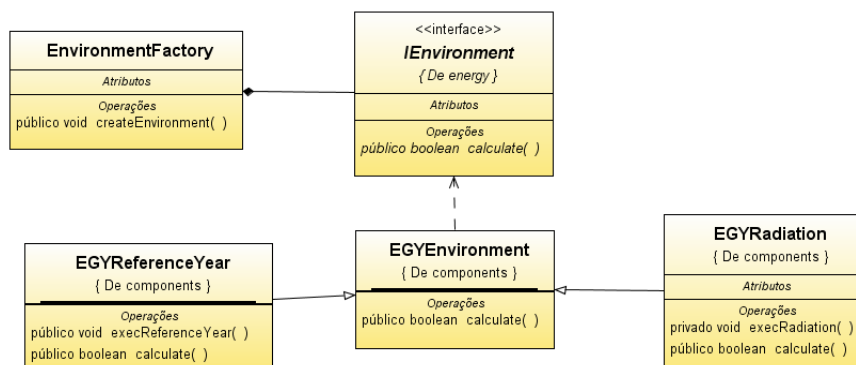


Figura 4.8: Diagrama de classes da implementação do padrão de projetos *Factory Method*.

Já a classe *EGYReferenceYear* implementa a interface e realiza a estimativa da radiação solar no plano inclinado a partir do ano de referência estimado pelos dados climatológicos medidos.

```

public class EGYReferenceYear extends EGYEnvironment{
    ...
    public boolean calculate(){
        boolean result = true;
        if ( verifica() )
            execReferenceYear();
        else result = false;
        return result;
    }
    ...
}

```

A responsabilidade de aplicar o padrão é da classe *EnvironmentFactory* e conforme a diretiva de projeto informada ao *Solar Energy*, retorna um objeto do tipo *IEnvironment*.

```
public class EnvironmentFactory {
    ...
    public static IEnvironment getInstanceOf(String tipo){
        if(tipo.equalsIgnoreCase("radiacao"))
            env = new EGYRadiation();
        else
            if(tipo.equalsIgnoreCase("anoPadrao"))
                env = new EGYReferenceYear();
        return env;
    }
}
```

Retirada do padrão de projeto *Singleton*

Fez-se necessária uma adaptação para realizar a retirada do padrão de projeto *Singleton*, utilizado na versão inicial do *framework*.

Em função das aplicações derivadas obterem dados de irradiância de apenas uma única fonte, este padrão foi aplicado para garantir que a classe referente a estimativa da radiação solar *EGYRadiation* tivesse uma única instância e um único ponto de acesso em todo o *framework*.

Os aspectos considerados no uso do *Singleton* foram o fato do limitador de acesso do construtor da classe ser privado e o método de recuperação da instância ser estático.

Porém, com a criação do *hot spot* ambiente e a possibilidade de estimar a radiação por mais de uma origem, foi invalidada a possibilidade de utilização deste padrão, isto é, com a herança da classe ambiente, existe a possibilidade das sub-classes herdadas instanciarem mais de um objeto da super-classe, inviabilizando a aplicação do padrão *Singleton*.

Para que essa adequação fosse realizada, o limitador de acesso do construtor da classe *EGYEnvironment* foi alterado para protegido, permitindo que qualquer sub-classe que esteja no pacote *Components* fizesse uso dos métodos e atributos da super-classe.

Entretanto, deve-se garantir que exista apenas um único ponto de acesso à variável ambiente. Esta garantia é implementada pela atribuição única da variável *environment*, mesmo que vários objetos sejam instanciados pelo desenvolvedor. Este atributo é considerado o ponto comum da ferramenta e definido no objeto derivado da classe *Energy*.

Os trechos de código, a seguir, mostram a aplicação desta abordagem na ferramenta.


```

public class Energy {
    private IEnvironment environment = null;
    public Energy(IBasicMain frm, EMCEEnvironment chooseenv){
        main = frm;
        ChooseEnv = chooseenv;
        startVariables();
    }
    public void show(){
        if (environment == null)
            environment = EGYEnvironment.getInstanceOf();
        ...
    }
    ...
}

public class EGYEnvironment extends EMCCComponent implements IEnvironment{
    protected EGYEnvironment() {
        radiationVariables = new Variables();
    }
    public static IEnvironment getInstanceOf(){
        IEnvironment env;
        env = new EGYEnvironment();
        return env;
    }
    ...
}

```

Pontos de conexão

Foram realizadas modificações nos pontos de conexão, disponibilizando pontos de entrada e saída. Desta forma, realiza-se uma verificação de conformidade das ligações entre os dispositivos em tempo de execução, além da determinação de variáveis de origem e destino na transferência de informações processadas entre os componentes. A principal vantagem desta evolução está na verificação de consistências no momento de execução da simulação, uma vez que as variáveis de origem e destino identificam o caminhamento no grafo de execução. As regras de negócio impedem as ligações entre entradas e entre saídas. A classe que implementa essa evolução é a *EMCJoinPoint* e os tipos são atribuídos a propriedade *typePoint*.

O *frozen spot Reports*

Análises estatísticas dos dados processados após a execução das simulações para o aquecimento de água ou fotovoltaicas são necessárias. Logo, foi desenvolvido um conjunto de classes concretas, agrupadas em um pacote, denominado *Reports*, com a finalidade de fornecer resultados visuais para qualquer aplicação derivada do *framework*. O objetivo principal deste *frozen spot* é gerar uma saída gráfica e genérica que contemple o resultado do processamento de todos os dispositivos presentes no diagrama de simulação. A evolução está no fato do desenvolvedor não necessitar conhecer a estrutura interna, nem a ferramenta de desenvolvimento em que os gráficos são gerados, sendo suficiente instanciar apenas os objetos correspondentes à aplicação desenvolvida.

4.4 Projeto

Identificados os requisitos dos aplicativos fotovoltaicos, os pontos comuns e variáveis, realizada a análise da arquitetura e uma proposta de inovações, inicia-se a terceira fase da metodologia, que é o projeto do *framework*. O projeto deve ser avaliado sobre dois aspectos: arquitetura e hierarquia de classes.

O aspecto arquitetural está relacionado com a persistência dos objetos, as interfaces gráficas utilizadas e as questões relacionadas a segurança e gerência das informações. Foram utilizadas a análise da arquitetura inicial e a proposta de inovações como premissa básica ao projeto. O segundo consiste da criação do modelo de classes e os relacionamentos entre as mesmas, analisando os aspectos comuns para a divisão em camadas. O *framework* foi dividido nas camadas de manipulação, negócios e componentes.

O diagrama resultante representa, basicamente, todas as funcionalidades cobertas pelo *Solar Energy*. Após a modelagem das classes, foi analisado o modelo arquitetural, para identificar outros elementos da estrutura que necessitam ser implementados por meio de novas classes não previstas inicialmente. Por exemplo, foram identificadas e agrupadas nos pacotes *Database*, *Forms* e *Utils* todas as classes relacionadas com a arquitetura de *software* escolhida, incluindo camadas de persistência, segurança e controle. O resultado dessa fase é o modelo completo de projeto do *framework*. A Figura 4.9 representa o diagrama contendo as principais classes e relacionamentos da ferramenta *Solar Energy*.

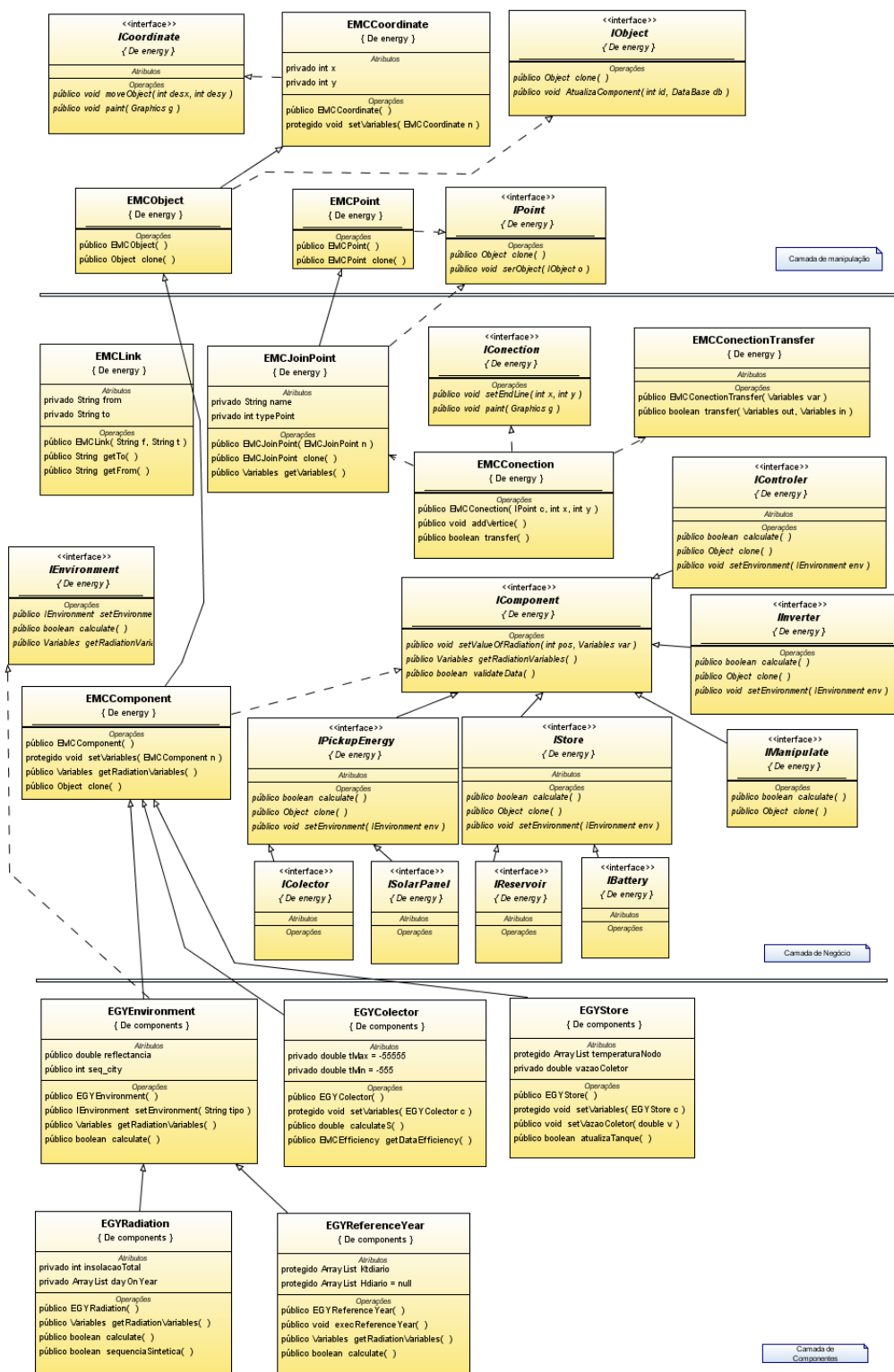


Figura 4.9: Diagrama de classes do Solar Energy.

4.5 Implementação

Definido o projeto, a próxima fase realizada foi a implementação do *Solar Energy*. Neste passo, as classes do *framework*, definidas no projeto, são implementadas usando a linguagem de programação orientada a objetos *Java*. O diagrama de classes forneceu o apoio necessário ao desenvolvimento, já que é uma fonte completa de informações sobre o domínio das aplicações. Outras decisões específicas sobre a implementação do sistema foram tomadas nesta fase como, a escolha das estruturas de dados, dos tipos de dispositivos utilizados para entrada de dados nas interfaces gráficas e o estilo e forma de saída dos dados processados pelos sistemas.

É de fundamental importância criar a documentação adequada do *Solar Energy*. As diretrizes fornecidas no processo proposto nesta dissertação conduzem a uma documentação, que fornece o mapeamento correto para as classes do *framework*. O propósito geral é facilitar a instanciação de aplicações específicas, fornecendo meios, sem a necessidade de um profundo conhecimento da estrutura do *framework*, para identificar novas classes a serem criadas ou especializadas e os métodos que devam ser sobrepostos.

4.6 Validação

A atividade final do processo de construção do *Solar Energy* é a validação, que tem como objetivo testar o desenvolvimento das aplicações derivadas. As entradas para este processo são o *framework*, a documentação dos processos de instanciação e a lista de requisitos para as aplicações específicas do domínio. Deve-se ressaltar que o processo de validação utiliza o mesmo processo de instanciação de aplicações, já que, para testar o *framework*, deve-se instanciar as aplicações concretas do domínio.

Para cada aplicação diferente a ser testada, seus requisitos devem ser analisados. Em seguida, deve-se realizar o mapeamento entre esse modelo e as classes do *framework*, utilizando o diagrama de classes do projeto. Depois a implementação das classes concretas da aplicação deve ser realizada, utilizando a mesma linguagem de programação orientada a objetos na qual o *Solar Energy* foi implementado. O resultado é o código da aplicação específica, que deve ser devidamente testado.

Este trabalho apresentou, como forma de validação, a evolução do aplicativo *Sol Brasil*, derivado do *Solar Energy*. Esta evolução está na necessidade de atender as demandas básicas do processo de conversão de energia solar em energia elétrica agregando conhecimento ao sistema existente, que atendia, inicialmente, apenas as necessidades do aquecimento de água.

Foi adotado como estratégia, testes que levaram em consideração as particularidades do *framework* e da aplicação, em diversos ciclos da execução, para que desta forma pudessem ser separados os erros da ferramenta e do *software* derivado.

4.7 Considerações Finais

Este capítulo apresentou as fases para construção de *frameworks* e a metodologia utilizada para a evolução do *Solar Energy*. A instanciação para sistemas específicos pode ser realizada com o mínimo de conhecimento sobre a ferramenta e a linguagem de programação na qual ela foi desenvolvida, pois a documentação gerada representa o mapeamento das funcionalidades principais utilizadas para o desenvolvimento das aplicações solares, isto é, pode-se consultar os documentos produzidos para obtenção do código da aplicação final desejada a partir do framework. O próximo capítulo apresenta o estudo de caso para a validação da etapa de criação e conseqüentemente do próprio *Solar Energy*.

Capítulo 5

Estudo de Caso: Evolução do Sol Brasil

5.1 Introdução

Este capítulo apresenta um estudo de caso baseado na documentação gerada da evolução do *framework Solar Energy* e no conjunto de passos a serem seguidos para adaptar a ferramenta em um arcabouço que, além de fornecer soluções para o dimensionamento e simulações do aquecimento de água por meio da energia solar, disponibilize também soluções para os problemas no domínio fotovoltaico.

5.2 Requisitos

O projeto de um sistema fotovoltaico é a relação entre a energia solar fornecida ao sistema e a demanda de energia gerada pela carga. Para a realização do dimensionamento utiliza-se a combinação dos seguintes critérios: energia gerada, confiabilidade do arranjo e eficiência do módulo.

Os métodos utilizados baseiam-se nas características dos componentes e valores típicos de radiação solar. A modelagem e a simulação computacional devem refletir satisfatoriamente a viabilidade da construção de projetos fotovoltaicos reais.

As etapas de desenvolvimento do projeto, que atendem as necessidades dos pesquisadores são:

- **Avaliação da radiação solar:** Na fase inicial do projeto, quantifica-se a radiação solar global incidente sobre o gerador fotovoltaico, de forma a calcular a energia gerada. Utiliza-se da definição do ano solar de referência para a estimativa da

radiação no plano horizontal. No dimensionamento determinístico, consideram-se os valores horários de energia.

- **Estimativa da curva de carga:** Necessita-se, também, da demanda de energia pela carga. O objetivo desta análise é a construção de uma curva de carga que expresse as eventuais sazonalidades. Este estudo pode implicar numa redução significativa do custo do sistema e prevenir contra efeitos de possíveis variações localizadas no comportamento da carga. A forma mais usual para se especificar uma carga para um projeto de sistema fotovoltaico é por meio do seu consumo diário de energia. É comum utilizar o valor de consumo correspondente aos dias em que a relação consumo/disponibilidade é máxima.
- **Escolha da configuração:** A escolha da configuração do sistema baseia-se nas características da carga e na disponibilidade de recursos energéticos. Ressalta-se que a configuração do sistema é o número de módulos e de baterias e a forma como serão arrançados (série ou paralelo).
- **Especificação dos demais componentes:** Os componentes controladores da demanda e do consumo devem ser configurados, além de definidos os dispositivos capazes de transformar a corrente gerada de forma contínua para a corrente alternada.

5.3 Solução proposta

A fase de validação do *framework* recebe como premissas de execução, a ferramenta implementada e a lista de requisitos da aplicação. É gerado o modelo de classes da aplicação, necessário ao projeto e implementação do aplicativo. A Figura 5.1 apresenta o referido diagrama.

Deve-se ressaltar que os *hot spots* são implementados tanto pela especialização das classes abstratas como pela composição dos componentes. As classes que exemplificam essa implementação são, respectivamente, relacionadas aos dispositivos do dimensionamento, como *SBFVSolarPanel*, *SBFVControler*, *SBFVBattery* e *SBFVInverter*, e as geradoras de saídas genéricas de resultado por relatórios gráficos como *ReportGraphics*, *SubReport* e *ReportSolarPanel*.

A derivação de um aplicativo pelo *framework Solar Energy* exige que a classe responsável pela inversão de controle seja instanciada no programa principal da aplicação. Pode-se informar a interface gráfica principal a ser utilizada ou fazer uso da interface padrão,

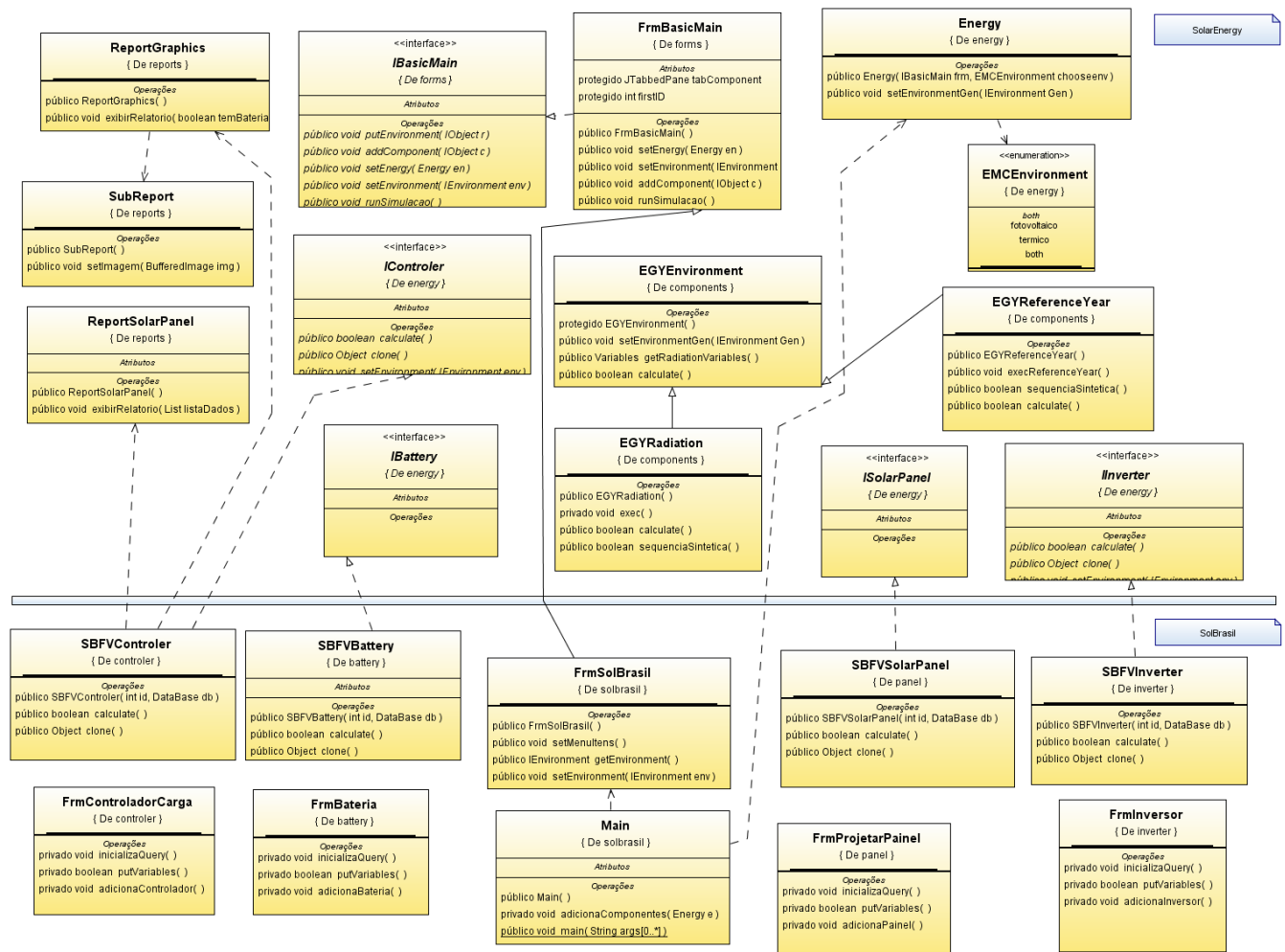


Figura 5.1: Diagrama de classes da aplicação Sol Brasil.

definida pela classe *Energy*. O trecho de código a seguir apresenta a instanciação da aplicação *Sol Brasil*:

```
public static void main(String[] args){
    File dirCorrente = new File(".");
    FrmSolBrasil frmPrincipal = new FrmSolBrasil();
    Energy e = new Energy(frmPrincipal, EMCEEnvironment.both);
    frmPrincipal.putComponents();
    e.show();
}
```

O argumento *both*, informado no construtor da classe *Energy*, permite que a aplicação derivada trabalhe com o dimensionamento e simulações tanto para o aquecimento de água, quanto para energia fotovoltaica. Entretanto não há a possibilidade de ocorrerem sim-

ulações em conjunto, ou seja, um arranjo fotovoltaico alimentar os equipamentos elétricos de um sistema de aquecimento de água. Deve-se ressaltar que para um processo evolutivo satisfatório, as funcionalidades dos sistemas térmicos não devem sofrer quaisquer alterações.

Por meio do *software Sol Brasil* é possível criar componentes, informando as principais propriedades exigidas nos cálculos (Capítulo 3) que são encapsulados pelas classes concretas definidas no desenvolvimento do *framework*. Os métodos de estimativa da radiação solar no plano inclinado são executados após a instanciação do objeto referente ao ambiente solar.

O modelo matemático de cada dispositivo presente no diagrama de simulação é executado no momento da aplicação da heurística de busca em profundidade em grafos com ciclos. A Figura 5.2 representa a tela principal do sistema *Sol Brasil*, em que a execução do dimensionamento determinístico real para um aplicativo fotovoltaico está detalhada. Os passos completos para a simulação serão apresentados a seguir.

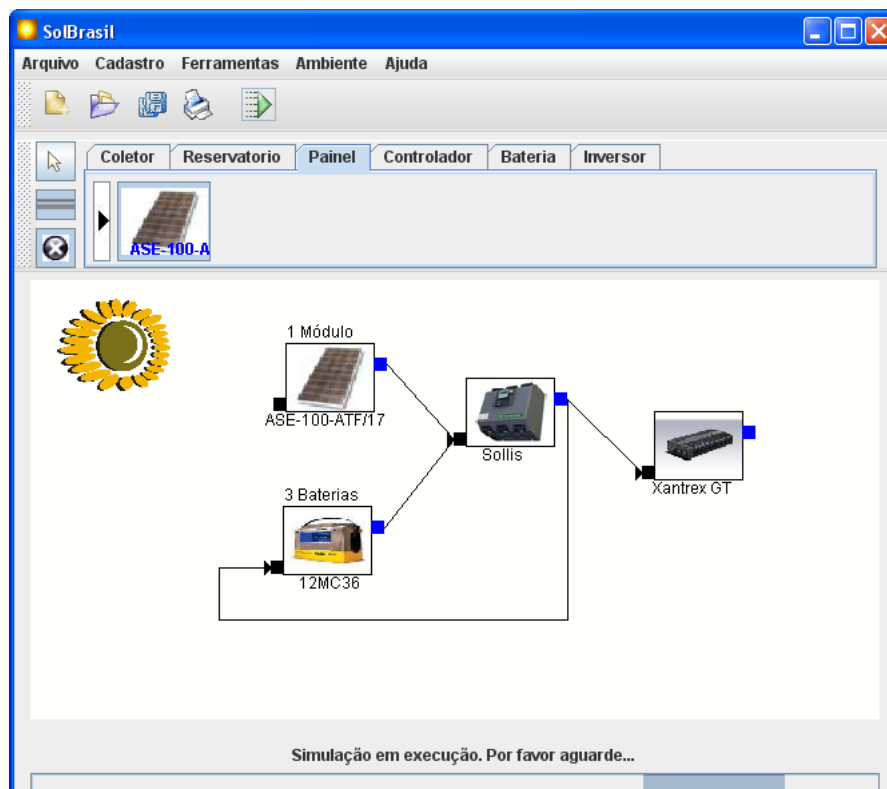


Figura 5.2: Tela principal do aplicativo *Sol Brasil*

5.3.1 Diagrama de execução

O projeto de sistemas fotovoltaicos depende da localização do sistema instalado. Dados sobre a latitude e altitude são importantes para o dimensionamento, pois o ângulo de inclinação do gerador fotovoltaico, que maximiza a sua geração de energia, varia conforme a época do ano. Isto significa que o mesmo gerador fotovoltaico pode ser dimensionado variando de região para região com diferentes valores estimados de radiação solar.

Os fatores que influenciam no balanço local de radiação e, conseqüentemente, na temperatura local do ar incluem: latitude, hora e dia do ano, cobertura de nuvens e a natureza da superfície. Estas variáveis determinam a intensidade e duração da irradiância solar, como também afetam o fluxo, o percentual da radiação absorvida e determinam o albedo¹.

A radiação global na superfície do painel foi obtida tendo como referência as propriedades físicas para a cidade de Belo Horizonte/Minas Gerais, visualizadas pela Tabela 5.1, e a aplicação do ano padrão para as variáveis climatológicas medidas para a referida cidade entre os anos de 2003 a 2006. A Figura 5.3 representa um exemplo de um conjunto de dados informado ao sistema para o processamento do ano de referência.

Tabela 5.1: Dados de entrada para Belo Horizonte

ϕ (Latitude)	-19,56'
Altitude (m)	850
β (Inclinação)	34°
Ângulo azimutal de superfície	180°
Reflectividade da vizinhança (Concreto)	0,22

Os dados medidos foram fornecidos pelo instituto meteorológico MGTempo - CEMIG contendo os valores climatológicos horários para cada dia do ano, tais como: radiação, temperatura máxima, média da temperatura, média da velocidade do vento e média para umidade relativa do ar. A interface responsável pela escolha do ambiente é apresentada pela Figura 5.4.

O processamento dos cálculos estatísticos é então realizado e identificado, dentre os anos informados como parâmetros, o ano de referência. Este ano é composto por vários meses padrão, cuja média estatística de cada mês mais se aproxima do padrão identificado no processo. Os anos passados passam a ter uma representação única composta pelos meses dos anos avaliados.

A partir da identificação do ano de referência e através da aplicação dos conceitos das matrizes de transição de *Markov*, apresentados na Seção 3.4.1 estima-se o indicador

¹Relação entre a radiação solar refletida ou dispersada pela superfície da Terra e o total que nela incide.

	Data	Hora	Temperatura Média	Temperatura Máxima	Unidade Média	Vento Médio	Radiação
1							
2	1/1/2003	0	19,1111	19,6246	66,2857	2,4857	0,0000
3	1/1/2003	1	18,7000	19,1668	67,2857	2,3857	0,0000
4	1/1/2003	2	18,3770	18,8344	70,2857	2,2571	0,0000
5	1/1/2003	3	18,1270	18,5262	72,7143	2,0286	0,0000
6	1/1/2003	4	17,9537	18,3184	73,8571	2,0143	0,0000
7	1/1/2003	5	17,7033	18,1070	75,8571	1,6714	0,0000
8	1/1/2003	6	17,5320	17,8750	78,2857	1,5000	0,0000
9	1/1/2003	7	18,2861	18,3336	71,1429	2,2714	35,0417
10	1/1/2003	8	19,9152	19,9955	64,1429	3,7429	479,1250
11	1/1/2003	9	21,3270	21,4557	60,2857	3,2286	1208,5000
12	1/1/2003	10	22,5303	22,7311	56,5714	2,8714	1835,0000
13	1/1/2003	11	23,4639	23,7541	49,8571	2,7000	2236,8750
14	1/1/2003	12	24,2631	24,6779	45,7143	2,6571	2498,6250
15	1/1/2003	13	24,8303	25,3148	42,7143	2,9571	2490,5417
16	1/1/2003	14	25,1180	25,7869	42,8571	2,3714	2247,3333
17	1/1/2003	15	25,1459	25,8377	41,0000	2,4000	1802,9167
18	1/1/2003	16	24,7082	25,5713	41,4286	2,4429	1182,0833
19	1/1/2003	17	23,5930	24,9709	42,7143	2,0857	414,0833
20	1/1/2003	18	22,2254	23,5930	49,4286	2,9143	15,3333
21	1/1/2003	19	21,3697	22,2500	57,7143	2,6143	0,0000
22	1/1/2003	20	20,7836	21,4328	59,7143	3,0143	0,0000
23	1/1/2003	21	20,2537	20,8623	61,8571	2,7143	0,0000
24	1/1/2003	22	19,7951	20,3574	63,1429	2,2714	0,0000
25	1/1/2003	23	19,4406	19,8934	64,7143	2,9000	0,0000
26	2/1/2003	0	19,2410	19,8304	67,9560	2,3791	0,0000
27	2/1/2003	1	18,7769	19,2924	69,3406	2,2714	0,0000
28	2/1/2003	2	18,4788	18,9010	72,1868	2,1142	0,0000
29	2/1/2003	3	18,2557	18,6319	74,4286	2,1341	0,0000
30	2/1/2003	4	18,0957	18,4324	75,4066	2,0901	0,0000
31	2/1/2003	5	17,8646	18,2372	77,2527	1,6198	0,0000
32	2/1/2003	6	17,6757	18,0231	79,5714	1,6308	0,6923
33	2/1/2003	7	18,3641	18,4156	72,9781	2,3582	44,0385

Figura 5.3: Conjunto de dados climatológicos medidos para a cidade de Belo Horizonte para a identificação do Ano Padrão.

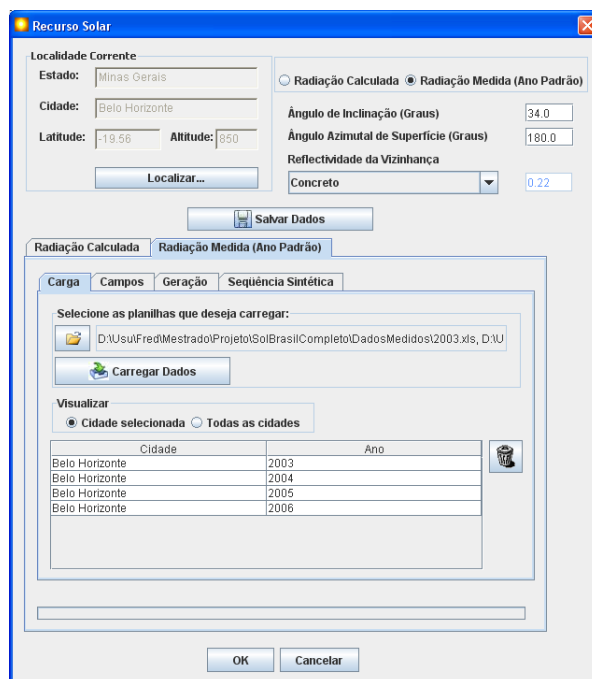


Figura 5.4: Interface para escolha do ambiente.

diário de claridade para cada dia do ano. Dessa forma identifica-se a seqüência sintética de dias chuvosos que afetam diretamente a estimativa da radiação e conseqüentemente o

dimensionamento realizado.

O diagrama de execução foi projetado de forma a atender um sistema residencial fotovoltaico isolado da rede elétrica e os dispositivos foram especificados conforme descrito a seguir.

Painel Solar

Deve-se evidenciar que para este trabalho só foram modelados os cálculos dos painéis solares com ligações em série.

O painel solar utilizado foi o *ASE-100-ATF/17* do fabricante *ASE Americas* e suas propriedades são apresentadas pela Tabela 5.2

Tabela 5.2: Especificação do Painel Solar ASE-100-ATF/17

Área do painel (m^2)	0,987
Número de células do módulo	36
Tensão de circuito aberto (V)	20
Tensão de máxima potência (V)	17,2
Corrente de curto-circuito (A)	6,4
Corrente de máxima potência (A)	5,8
Coefficiente alfa de temperatura (α)	0,00078
Coefficiente beta de temperatura (β)	-0,076
Potência de referência (W)	99,76

As condições padrão de operação para o painel fotovoltaico foram a irradiância solar de referência com o valor de $1000W/m^2$ e a temperatura solar de referência a $25^\circ C$.

A potência de saída é um parâmetro importante para o dimensionamento, pois o suprimento de energia para o consumo e para as baterias depende diretamente deste valor. A potência gerada pelo painel solar é fortemente dependente da insolação e da temperatura ambiente.

Bateria

A Tabela 5.3 apresenta as especificações da bateria utilizada na unidade de armazenamento. Deve-se ressaltar que o banco de baterias deve ser composto por dispositivos da mesma marca e modelo, ou seja, não é possível realizar o dimensionamento utilizando componentes distintos. A marca utilizada no estudo de caso foi a *MOURA* e o modelo *12MC36*.

A capacidade de armazenamento das baterias é utilizada para quantificar e expressar a máxima profundidade de descarga e a máxima carga a que pode ser submetida. O

Tabela 5.3: Propriedades do componente bateria 12MC36

Energia inicial de carga (Capacidade em 20h) (W)	432
Energia armazenada máxima (% prof. mín)	5
Energia armazenada mínima (% prof. máx)	70
Eficiência (%)	85

tempo de vida útil da unidade de armazenamento está diretamente relacionado aos ciclos diários de carga e descarga que as baterias são submetidas. Quanto mais profundo o ciclo de descarga, mais curta a vida da bateria. Há também uma redução de sua eficiência se o banco de baterias se aproximar do estado máximo de carga. A variação da temperatura também é fator determinante para a análise de eficácia deste dispositivo.

Controlador de carga

O componente controlador modelado pela aplicação foi o *Sollis* do fabricante *Variix*. Sua eficiência apresenta valor constante e tem a característica de consumir pouca quantidade da energia gerada pelo arranjo fotovoltaico. A principal função deste dispositivo é atuar no processo de execução de forma a não permitir que a unidade de armazenamento seja sobrecarregada ou que sofra uma descarga profunda, ou seja, realizar o balanço energético da relação carga/demanda e banco de baterias. O processo inverso também é controlado por este dispositivo, em que a potência das baterias é fornecida ao arranjo nos momentos de déficit da carga gerada pelo módulo fotovoltaico.

Inversor

O inversor, também conhecido como conversor de corrente contínua (CC) em corrente alternada (CA), estabelece uma conexão entre o gerador fotovoltaico e a rede ou carga de CA. A sua principal tarefa consiste em converter o sinal elétrico CC do gerador fotovoltaico em um sinal elétrico CA, e ajustá-lo para a frequência e o nível de tensão da rede a que está ligado. O modelo projetado como estudo de caso no *Sol Brasil* foi o *Xantrex GT*, do fabricante *XANTREX* e sua eficiência de conversão é de 95%. Este dispositivo também consome pouca quantidade de energia e sua demanda é considerada no momento do controle da carga gerada pelo módulo.

5.3.2 Definição da demanda

O funcionamento do sistema fotovoltaico depende da demanda informada ao diagrama. Visando atender a esta premissa, foi fornecido pela CEMIG-MG dados que representam

o consumo horário de uma residência rural em um determinado dia do ano e medido em W/h , conforme demonstrado pelo gráfico da Figura 5.5. Deve-se ressaltar que o estudo de caso realizado analisa o comportamento da estimativa da radiação solar para a cidade de Belo Horizonte sobre o arranjo fotovoltaico cuja a demanda cadastrada está relacionada ao consumo rural.

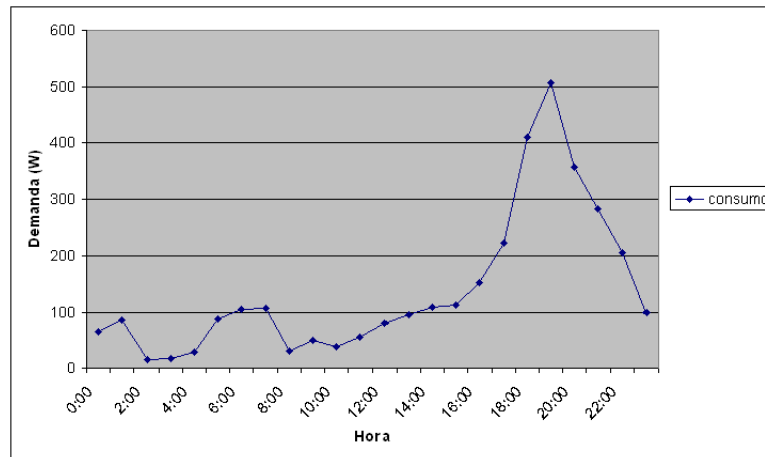


Figura 5.5: Gráfico representativo da curva horária de consumo.

O perfil de consumo é cadastrado pelo *frozen spot* $FrmConsumo$ e seus dados armazenados na estrutura de banco de dados correspondente. A identificação do ano padrão permite que o dimensionamento fotovoltaico seja diário ao longo do ano. Logo, foi necessário estender os valores horários para uma estimativa mensal conforme demonstrado pela Figura 5.6. Para a determinação da curva de demanda anual, realiza-se a soma dos valores diários lançados hora a hora e seu valor é medido em kW/h .

Foi aplicado a cada valor total mensal um desvio padrão de forma a garantir uma variação do consumo anual. A Tabela 5.4 apresenta a variação destes desvios.

Tabela 5.4: Desvio padrão aplicado a cada valor mensal de consumo

Janeiro	1.00	Julho	0.70
Fevereiro	0.95	Agosto	0.79
Março	0.92	Setembro	0.85
Abril	0.89	Outubro	0.92
Maio	0.85	Novembro	0.95
Junho	0.81	Dezembro	1.0

O *framework* permite uma flexibilidade no cadastro do consumo, por ser possível também informar a demanda através do cadastro de equipamentos e a quantidade de

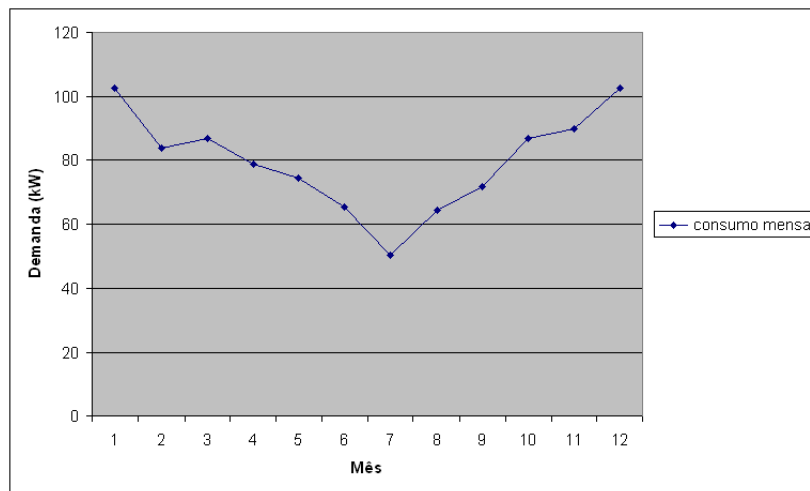


Figura 5.6: Gráfico representativo da curva mensal de consumo.

energia gasta, dada em W , por hora de utilização. Pode-se associar vários equipamentos por intervalo de hora, sendo o total calculado pelo somatório dos valores gastos de energia.

5.3.3 Execução

O modelo utilizado na aplicação *Sol Brasil* foi o de determinação da curva de máxima potência e o *framework Solar Energy* provê através da interface gráfica um conjunto de componentes, previamente cadastrados e organizados pela finalidade de cada dispositivo.

Cada componente é disposto nesta interface de forma a definir o diagrama de execução. O componente ambiente solar provê os dados solarimétricos necessários, sejam medidos ou calculados e transferem a fonte de energia para o componente captador e conversor. O perfil de consumo é informado ao sistema como também a quantidade de módulos que compõe o arranjo e a quantidade de baterias da unidade de armazenamento. Foram modelados duas configurações no processo de simulação, sendo o sistema avaliado para um painel e três baterias e, posteriormente, três módulos solares e apenas uma bateria.

O ciclo de execução do processo fotovoltaico é inicializado, onde todos os dispositivos tem suas funcionalidades executadas na medida que ocorre o caminhar no grafo de execução e o vértice correspondente é encontrado. O intervalo do ciclo corresponde as horas de cada dia do ano definido como padrão e a execução consiste em: determinar a potência máxima nas horas que ocorrem a radiação, realizar o balanço energético entre a demanda e os demais componentes, gerar resultados estatísticos da radiação horária média mensal no plano inclinado, da potência horária em média mensal, da energia armazenada nas baterias ao longo do ano e possíveis falhas no dimensionamento. Os resultados obtidos

são descritos na seção seguinte.

5.4 Resultados obtidos

O resultado apresentado para o ano padrão, dos dados climáticos, para a cidade de Belo Horizonte, é demonstrado pela Figura 5.7. Ele é formado pela composição dos meses padrão para os anos que compõem o período avaliado. Caso um novo ano seja acrescentado ao processo avaliativo, deve-se cadastrá-lo com as mesmas variáveis analisadas e executar novamente o processo de identificação. Os meses do ano incluído podem ou não compor o novo ano de referência.

Cidade	Mês	Ano
Belo Horizonte	1	2005
Belo Horizonte	2	2004
Belo Horizonte	3	2004
Belo Horizonte	4	2006
Belo Horizonte	5	2005
Belo Horizonte	6	2004
Belo Horizonte	7	2003
Belo Horizonte	8	2003
Belo Horizonte	9	2005
Belo Horizonte	10	2003
Belo Horizonte	11	2006
Belo Horizonte	12	2006

Figura 5.7: Identificação do ano padrão para o período informado.

A partir do ano padrão é possível gerar a sequência sintética do índice de claridade para os dias chuvosos, extraíndo, dessa forma, resultados de grande importância para a irradiância solar e conseqüentemente para o processo de simulação.

A Figura 5.8 representa a relação entre o índice de claridade ao longo do ano de referência e a radiação extraterrestre refletida na superfície. Esta é representada pela curva H_o e não possui influência de nenhum tipo de desvio causado pelas camadas da atmosfera,

portanto, seu comportamento é regular e sua intensidade elevada em praticamente todo o momento.

A radiação na superfície da terra H , é modelada principalmente em função do índice de claridade Kt , que representa a influência de nuvens e gases atmosféricos no brilho da radiação, ou seja, quanto mais nublado, menor a irradiância.

Percebe-se que principalmente nas extremidades do gráfico, onde a estação do verão para o hemisfério sul está representada, o efeito do Kt é bastante inerente, influenciando fortemente na radiação emitida na superfície terrestre e podendo ser observado através do comportamento da curva no gráfico. É importante salientar que os gráficos estão em escalas distintas, ou seja, os valores dos dados para a radiação H e H_o diferem do índice de claridade, sendo possível analisar e comparar o comportamento das curvas e não suas grandezas.

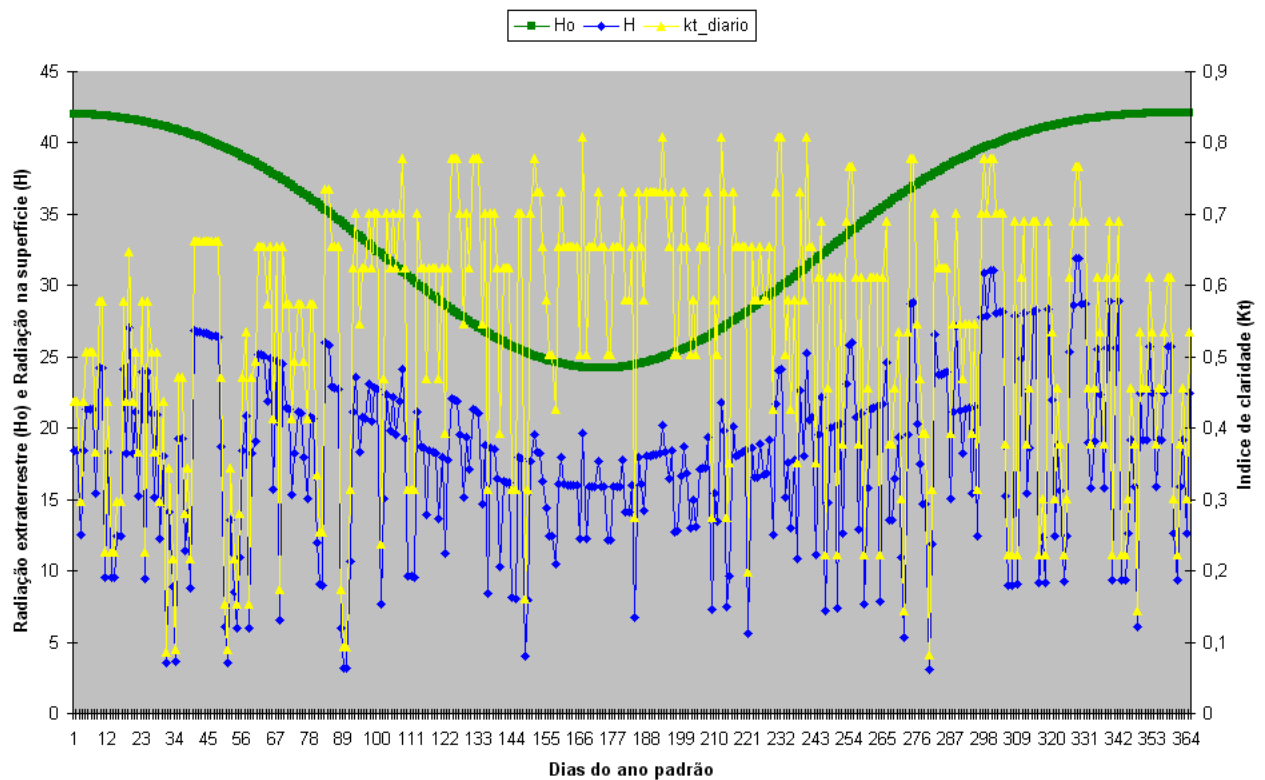


Figura 5.8: Relação entre a radiação extraterrestre e da superfície, em função do índice de claridade ao longo do ano.

A Figura 5.9 exhibe os dados simulados de irradiação global horária, para todos os dias do ano, para o plano inclinado a 34° . O gráfico apresenta a radiação horária média mensal com base no ano de referência. Pode-se perceber um aumento da irradiação na

parte central do gráfico para o plano inclinado, mostrando que a inclinação favoreceu uma melhor incidência dos raios solares sobre os módulos fotovoltaicos, nos meses com menor índice de irradiação e sobretudo com elevados índices de claridade. A radiação no plano horizontal é calculada como energia e dada em MJ/m^2 , entretanto, foi padronizado que o plano inclinado fosse calculado em W/m^2 , caracterizando potência.

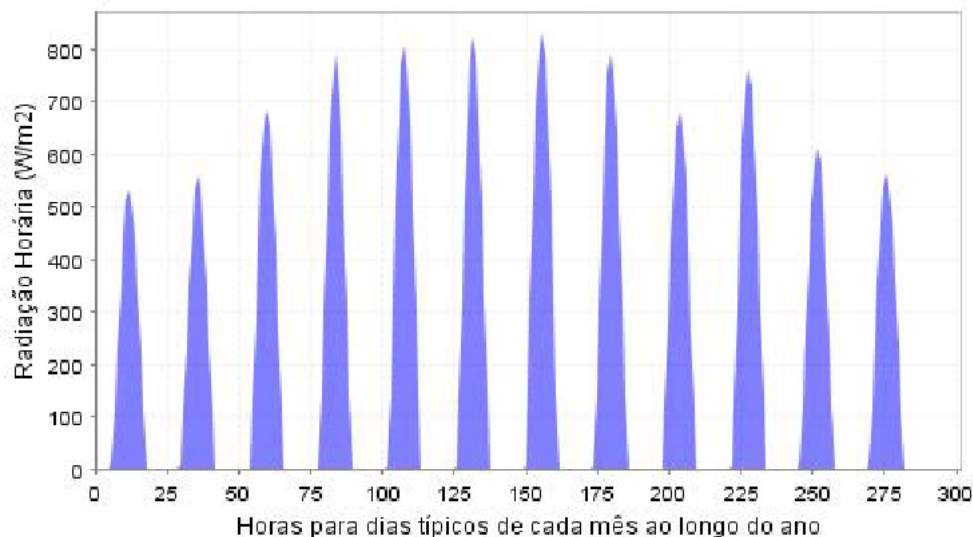


Figura 5.9: Radiação horária média mensal no plano inclinado gerada pelo ano padrão.

Os resultados referentes a etapa de identificação do ano padrão e estimativa da radiação foram confrontados com resultados apresentados pelo laboratório do GREEN com o intuito de avaliar os processos de iluminância². Os dados apresentaram-se equivalentes, comprovando a coerência da implementação.

Foi levado em consideração, para o processamento do sistema fotovoltaico o número de módulos e de baterias. Verifica-se o comportamento do dimensionamento com os geradores de potência (painel e bateria) realizando alterações em suas configurações e identificando a escolha da configuração adequada à demanda especificada. As Figuras 5.10 e 5.11 representam os gráficos para a potência fornecida por 1 módulo fotovoltaico e a energia armazenada em um banco contendo 3 baterias, respectivamente, ao longo do ano.

Após a geração de dados de irradiação, são realizados os cálculos para a estimativa do ponto de máxima potência do gerador fotovoltaico, que consiste em gerar curva da relação entre a corrente I e a tensão V para cada hora avaliada e recuperar o valor máximo na inflexão da curva. O gráfico da potência gerada pelo módulo também está em média horária mensal ao longo do ano.

²Fração visível do espectro solar

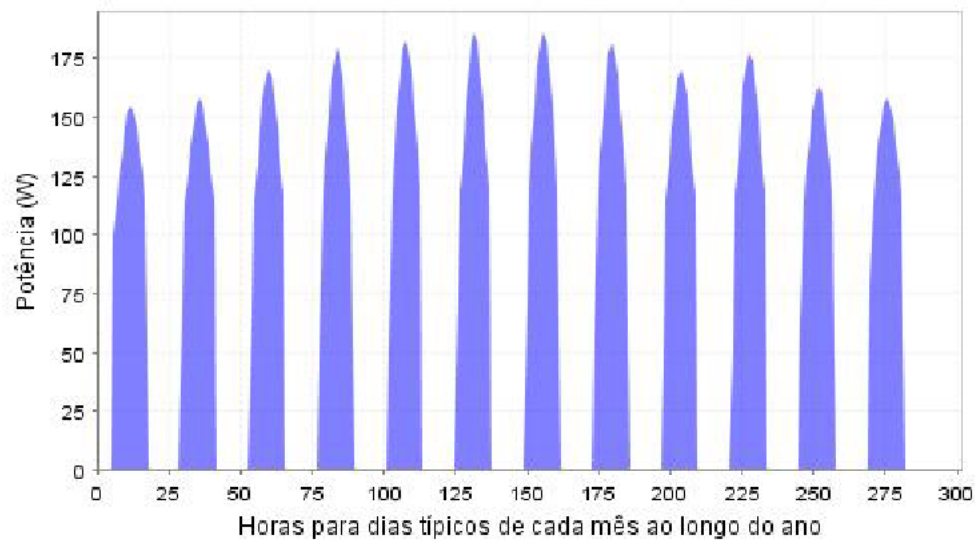


Figura 5.10: Potência horária média mensal gerada pelos módulos fotovoltaicos ao longo do ano para 1 painel e 3 baterias.

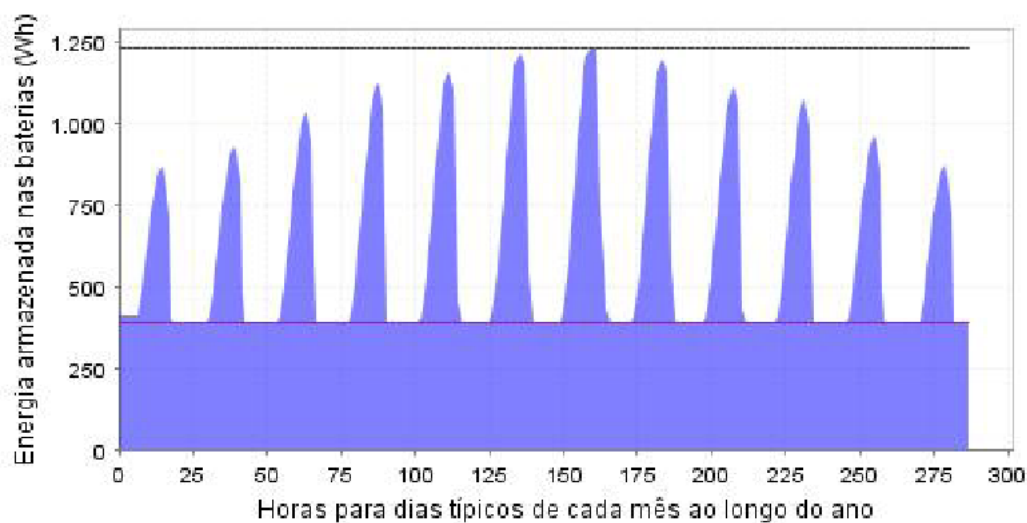


Figura 5.11: Energia armazenada nas baterias durante todo o ano (horária média mensal) para 1 painel e 3 baterias.

No gráfico da simulação para a energia armazenada no banco de baterias, a linha superior delimita a capacidade máxima de armazenamento, acima do qual o banco estaria sendo sobrecarregado, e a linha inferior delimita a profundidade máxima de descarga, abaixo da qual as baterias estariam sujeitas a uma descarga profunda, danificando-as. Observa-se, que a capacidade útil encontra-se entre estes dois limites, e que o controlador

de carga interfere atuando de forma a não permitir que o banco seja sobrecarregado ou que sofra uma descarga profunda.

Avalia-se que a energia armazenada ao longo do ano sofre uma variação diretamente proporcional ao valor de potência horária média mensal gerada pelos módulos, fazendo com que o banco de baterias não seja carregado o suficiente para restabelecer a energia máxima inicial, nos meses representados pelas extremidades do gráfico, o que reduziria o tempo de vida útil do banco de baterias.

Entretanto, ao realizar os testes utilizando 3 painéis e 1 bateria como unidade de armazenamento, percebe-se um aumento significativo da energia fornecida ao banco, mantendo-se praticamente constante, sem períodos de insuficiência de recarga. Este fato deve-se principalmente em função da potência gerada pelos módulos ser suficiente para alimentar o consumo e, também, recarregar o banco de baterias, após o período de utilização da carga em horários onde não há irradiância solar. Os gráficos que representam a geração de potência e a energia armazenada nas baterias, para a configuração correspondente, ao longo do ano, são apresentados pelas Figuras 5.12 e 5.13, respectivamente.

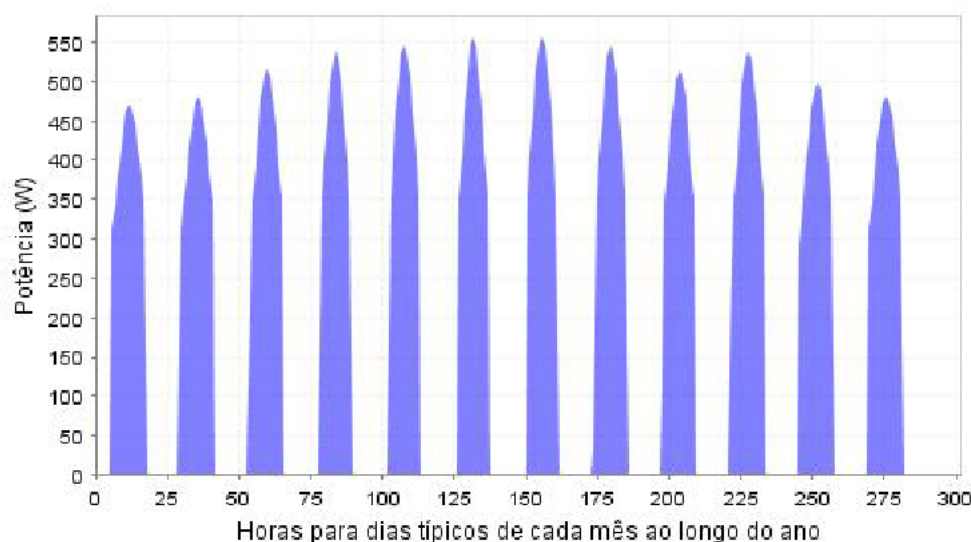


Figura 5.12: Potência horária média mensal gerada pelos módulos fotovoltaicos ao longo do ano para 3 painéis e 1 bateria.

Para que a análise do processamento seja completa, é gerado um relatório que lista a relação das falhas no dimensionamento ao longo de todas as horas de cada dia do ano padrão. O processo consiste na avaliação da relação existente entre o controlador de carga, a potência disponível pelo banco de baterias, a potência gerada pelo painel solar e o consumo. Deve-se ressaltar que a energia consumida pelos próprios dispositivos é

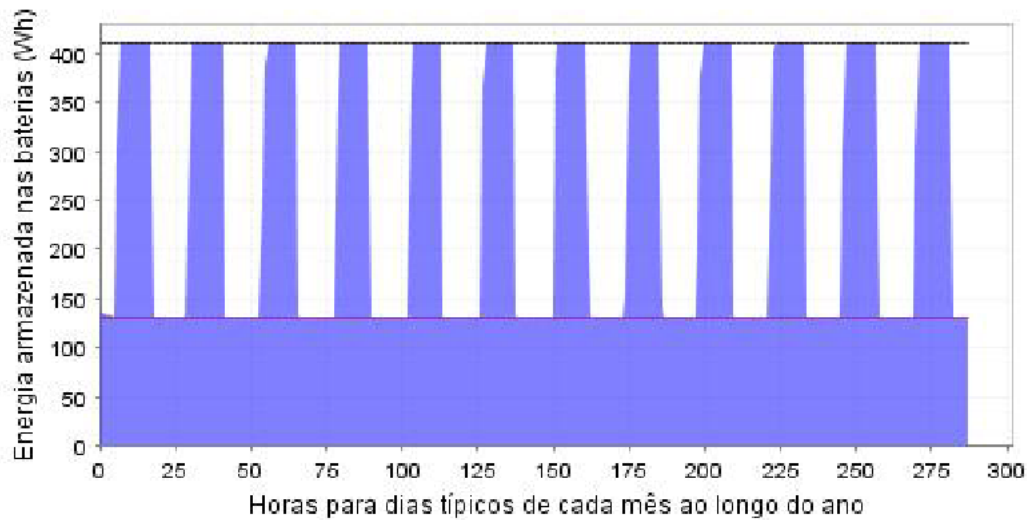



Figura 5.13: Energia armazenada nas baterias durante todo o ano (horária média mensal) para 3 painéis e 1 bateria.

levada em consideração, sendo acrescentada ao consumo cadastrado, conforme discutido anteriormente.

A Figura 5.14 apresenta parte do relatório de falhas para a execução da primeira configuração (1 painel e 3 baterias). Os problemas são apresentados caso ocorra uma das seguintes situações: o consumo ser superior à potência gerada pelo arranjo fotovoltaico; o somatório entre a potência fornecida pelo banco de baterias e a gerada pelo painel solar não for suficiente para o atendimento da demanda; e a diferença entre a potência fornecida pela unidade de armazenamento e a carga consumida for inferior a profundidade máxima de descarga ou nível mínimo de carga, o que resulta em danos irreversíveis ao equipamento.

Como exemplo, pode-se verificar pelo relatório, o período entre 19:00 e 20:00, para o primeiro dia de Janeiro. Neste intervalo, não há fornecimento de energia pelo módulo e o valor consumido resultará em uma descarga superior ao limite permitido pela bateria. Outro exemplo de falha do dimensionamento, corresponde ao período entre 06:00 e 07:00 para o terceiro dia de Janeiro, no qual o consumo de $105.24W$ é superior aos $104.09W$ fornecidos pelo módulo fotovoltaico e a unidade de armazenamento encontra-se em sua profundidade máxima de descarga.

A análise detalhada destas falhas é fator relevante para o desempenho de um sistema fotovoltaico, pois deve-se verificar o comportamento do sistema, mediante as várias combinações de módulos e baterias. Com isto, pode-se escolher os dispositivos mais ade-



Relação das falhas no dimensionamento

28/09/08 15:48

Perfil de Consumo Demanda residência-rural 2008

Arranjo Fotovoltaico 1 Painel(is) **Profundidade máxima de energia na(s) bateria(s)** 388.8 Wh

Banco de Baterias 3 Bateria(s) **Profundidade mínimo de energia na(s) bateria(s)** 1231.2 Wh

Janeiro

Dia 1

PERÍODO	POTÊNCIA MÓDULO(S)	POTÊNCIA BATERIA(S)	CONSUMO
19 - 20	0.0	649.26	508.38
20 - 21	0.0	388.8	356.85
21 - 22	0.0	388.8	281.11
22 - 23	0.0	388.8	205.6
23 - 0	0.0	388.8	98.43

Dia 2

PERÍODO	POTÊNCIA MÓDULO(S)	POTÊNCIA BATERIA(S)	CONSUMO
0 - 1	0.0	388.8	64.56
1 - 2	0.0	388.8	85.67
2 - 3	0.0	388.8	14.53
3 - 4	0.0	388.8	17.1
4 - 5	0.0	388.8	28.07
5 - 6	0.0	388.8	88.52
6 - 7	103.98	388.8	105.24
18 - 19	0.0	732.18	408.97
19 - 20	0.0	388.8	508.38
20 - 21	0.0	388.8	356.85
21 - 22	0.0	388.8	281.11
22 - 23	0.0	388.8	205.6
23 - 0	0.0	388.8	98.43

Dia 3

PERÍODO	POTÊNCIA MÓDULO(S)	POTÊNCIA BATERIA(S)	CONSUMO
0 - 1	0.0	388.8	64.56
1 - 2	0.0	388.8	85.67
2 - 3	0.0	388.8	14.53
3 - 4	0.0	388.8	17.1
4 - 5	0.0	388.8	28.07
5 - 6	0.0	388.8	88.52
6 - 7	104.09	388.8	105.24
18 - 19	0.0	628.0	408.97
19 - 20	0.0	388.8	508.38
20 - 21	0.0	388.8	356.85
21 - 22	0.0	388.8	281.11
22 - 23	0.0	388.8	205.6

Green - Grupo de Estudos em Energia Solar Página 1 de 120

Figura 5.14: Relação das falhas no dimensionamento.

quados ao dimensionamento para equilibrar de forma significativa o funcionamento de todo o processo.

5.5 Considerações Finais

Este capítulo apresentou como estudo de caso o desenvolvimento de uma aplicação fotovoltaica baseada no *framework Solar Energy*. Foram apresentadas as etapas de definição dos requisitos, além do processo de instanciação do *software Sol Brasil*. O dimensionamento e a simulação fotovoltaica foram baseadas em um modelo real e estenderam-se ao longo das horas para cada dia do ano padrão. O processamento incluiu desde a captação da irradiância solar e a geração da potência por meio do arranjo fotovoltaico, até a análise do consumo, inversão da corrente e balanceamento energético realizado pelos componentes controlador de carga e bateria, bem como as análises gráficas e analíticas dos resultados obtidos a partir das configurações informadas.

Visando realizar um comparativo entre os dimensionamentos determinístico, modelado por este trabalho, e estocástico, discutido e apresentado por Cabral [CAB06], foi realizado um estudo de caso confrontando os resultados das duas abordagens e utilizando as mesmas configurações de dimensionamento e demanda. O Apêndice E relata esta comparação.

Capítulo 6

Conclusões

A principal contribuição deste trabalho está na evolução de um arcabouço, originalmente construído com vistas à solução de problemas no domínio de aquecimento de água por meio da energia solar, para resolver também os problemas relacionados à geração de energia fotovoltaica. O desenvolvimento de um *framework* orientado a objetos colaborou para aplicação do reuso de *software*. Este conceito resolve um dos principais problemas que os engenheiros de *software* confrontam: o desenvolvimento de ferramentas reutilizáveis que possam ser adaptadas às famílias de aplicações dentro de um mesmo domínio específico.

Frameworks permitem aumentar a flexibilidade devido a uma maior facilidade de customização, de acordo com os requisitos da aplicação. Possibilitam também, o reuso da análise, do projeto e de parte ou todo código fonte, aumentando a qualidade do *software*, reduzindo o esforço e o tempo de desenvolvimento de novas aplicações, além de reduzir o número de manutenções.

O trabalho apresentou alguns etapas para a construção e uso do *framework Solar Energy*. Para obter um sistema específico, o desenvolvedor das aplicações concretas necessita basicamente aplicar os passos definidos para a construção e instanciação do novo aplicativo.

Para demonstrar a aplicação deste conceito, foi desenvolvido, como estudo de caso, a partir das evoluções aplicadas ao *framework*, o aplicativo *Sol Brasil*. As funcionalidades deste *software* estão relacionadas ao dimensionamento e simulações, que representam os sistemas reais de conversão de energia elétrica a partir da energia solar, por meio das células fotovoltaicas.

Esta abordagem permite minimizar os esforços de desenvolvimento para cada novo sistema relacionado ao domínio solar e resultante do *framework*, reduzindo assim, o código a ser gerado, isto é, o aplicativo desenvolvido conterá apenas o código necessário ao atendimento dos requisitos específicos e o código fonte genérico será encapsulado pela

aplicação.

A solução proposta permitiu que o desenvolvimento de aplicativos de controle, análises, simulações e ensaios realizados por pesquisadores da área de energia solar fossem simplificados, em função do conhecimento ser agregado à ferramenta. Além disso, permitiu que o tempo e o custo de desenvolvimento fossem reduzidos pelo fato dos projetistas reutilizarem uma solução consolidada e testada.

Este trabalho, pode ser considerado como uma proposta inovadora para a área de energia solar, porque foi desenvolvida uma metodologia capaz de realizar o dimensionamento determinístico de sistemas fotovoltaicos isolados da rede elétrica, analisando a radiação solar, o processo de captação e geração de potência, o balanço energético, os resultados estatísticos e a identificação de falhas.

Através desta abordagem, o engenheiro ou o pesquisador pode verificar, por meio de modelos próximos do ambiente real, as possíveis combinações de dispositivos que satisfaçam a confiabilidade da configuração desejada, facilitando e simplificando os ensaios em sistemas fotovoltaicos.

6.1 Contribuições

As contribuições obtidas com este trabalho de mestrado estão listadas a seguir

- Apresentação de uma solução para atenuar o problema de replicação de código fonte na construção de aplicativos,
- Aplicação de uma proposta para a construção de *frameworks* e adaptação de uma ferramenta existente por meio da utilização do novo código gerado. Essa evolução foi definida por etapas de execução, na qual cada fase gerou um artefato de saída necessário a entrada da próxima iteração,
- Desenvolvimento e evolução de um *framework* que possa ser adaptado para uma família de aplicações dentro do mesmo domínio solar,
- Desenvolvimento de uma aplicação que implementou a solução proposta e gerou dimensionamentos, simulações e análises da geração de energia fotovoltaica,
- Publicação do artigo *Solar Energy: A Framework for Solar Energy Applications* no INDIN 2008 [GPS⁺08] Este artigo apresentou a utilização do paradigma da *Engenharia de Software*, baseado em *frameworks* para o desenvolvimento de aplicações relacionadas ao domínio solar, especificamente os que tratam de aquecimento de água.

6.2 Trabalhos Futuros

Como trabalho futuro, pretende-se investigar a aplicação da solução proposta em sistemas relacionados a outras fontes de energia renovável, como por exemplo sistemas eólicos. A finalidade está em utilizar a solução proposta para dimensionar e simular aplicações que transformem a energia cinética dos ventos em energia mecânica de rotação e verificar se ocorreu redução significativa no tempo de desenvolvimento para as aplicações deste domínio.

Em relação ao domínio estudado outros estudos de caso serão úteis, principalmente, para:

- Considerar o desenvolvimento de sistemas híbridos, de forma que seja analisado um dimensionamento de um sistema fotovoltaico gerando carga necessária à demanda dos dispositivos elétricos de um sistema de aquecimento de água,
- Aplicar a estimativa da radiação solar, por meio dos cálculos de insolação, para as aplicações fotovoltaicas,
- Aplicar a estimativa da radiação solar, por meio da identificação do ano padrão proveniente dos dados climatológicos medidos, para as aplicações de aquecimento de água.

Além disso, pode-se agregar ao *framework* o estudo da viabilidade econômica, por meio da análise de custo dos dispositivos, para que seja avaliada qual a configuração mais vantajosa financeiramente para se utilizar no dimensionamento de um sistema real, levando-se em consideração a depreciação dos dispositivos que compõem o arranjo fotovoltaico ou os componentes necessários ao aquecimento de água.

Pode-se realizar também, uma avaliação das vantagens e desvantagens da utilização de padrões de projeto no desenvolvimento de *frameworks*. Deseja-se assim, avaliar se a utilização dos padrões reduz efetivamente o esforço de construção de aplicativos e se o reuso de código aumenta significativamente a capacidade de incorporar novos componentes. Esta avaliação pode ser realizada por meio de métricas específicas, que quantifiquem a qualidade dos sistemas de *software*.

Bibliografia

- [ABD⁺97] V. D. ARNOLD, R. J. BOSCH, E. F. DUMSTORFF, P. J. HELFRICH, T. C. HUNG, V. M. JOHNSON, R. F. PERSIK, and P. D. WHIDDEN. IBM Business Frameworks: San Francisco Project Technical Overview. *IBM Systems Journal*, 36(3):437–445, 1997.
- [ACPC88] R. J. AGUIAR, M. COLLARES-PEREIRA, and J. P. CONDE. Simple Procedure for Generating Sequences of Daily Radiation Values Using a Library of Markov Transition Matrices. *Solar Energy*, 40(3):269–279, 1988.
- [ALKB99] A. ARGIRIOU, S. LYKOUDIS, S. KONTOYIANNIDIS, and C.A. BALARAS. Comparison of Methodologies for TMY Generation Using 20 Years Data for Athens, Greece. *Solar Energy*, 66(1):33–45, May 1999.
- [BAR01] Álvaro C. P. BARBOSA. *Middleware para Integração de Dados Heterogêneos Baseado em Composição de Frameworks*. Doutorado, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil, 2001.
- [BJ94] K. BECK and R. JOHNSON. Patterns Generate Architectures. *European Conference on Object-Oriented Programming*, pages 139–149, 1994.
- [BM04] R. T. V. BRAGA and P. C. MASIERO. Finding Frameworks Hot Spots in Pattern Languages. *Journal of Object Technology*, 3(1):123–142, 2004.
- [BMM⁺99] J. BOSCH, P. MOLIN, M. MATTSSON, P. BENGTSSON, and M. FAYAD. Framework Problem and Experiences. In *M. Fayad, R. Johnson, D. Schmidt. Building Application Frameworks: Object-Oriented Foundations of Framework Design*, pages 55–82. John Wiley & Sons, 1999.
- [BS00] D. BRUGALI and K. SYCARA. Frameworks and Pattern Languages: an Intriguing Relationship. *ACM Computing Surveys*, 32(1):2–7, 2000.

- [CAB06] Cláudia Valéria Távora CABRAL. Análise de Dimensionamento Estocástico e Determinístico de Sistemas Fotovoltaicos Isolados. Doutorado, Universidade Federal de Viçosa, Viçosa, Brasil, 2006.
- [CCE00] Heitor S. COSTA, Rodrigo A. COSTA, and Myriam ECK. Análise Econômica Comparativa da Eletrificação Rural Comercial e Fotovoltaica. *XVII - Conferência Latino-Americana de Eletrificação Rural*, 2000.
- [CEM04] Companhia Energética de Minas Gerais CEMIG. 19º Balanço Energético do Estado de Minas Gerais - ano base 2003. Technical report, CEMIG, Minas Gerais, Brasil, 2004.
- [CMOD04] Cláudia V. T. CABRAL, Lauro V. B. MACHADO, Delly OLIVEIRA, and Antônia S. A. DINIZ. Modelagem e Simulação de Gerador Fotovoltaico. *Energia Solar*, pages 15–19, 2004.
- [COP98] James O. COPLIEN. Software Design Patterns: Common Questions and Answers. In *Rising L., (Ed.), The Patterns Handbook: Techniques, Strategies, and Applications*, pages 311–320. University Press, 1998.
- [DB91] J. A. DUFFIE and W. A. BECKMAN. *Solar Engineering of Thermal Processes*. John Wiley & Sons, New York, 1991.
- [FJS99] M. E. FAYAD, R. E. JOHNSON, and D. C. SCHMIDT. *Building Application Frameworks: Object-oriented Foundations of Framework Design*. John Wiley & Sons, New York, USA, 1999.
- [FR93] R. FESTA and C.F. RATTO. Proposal of a Numerical Procedure to Select Reference Years. *Solar Energy*, 50(1):9–17, January 1993.
- [FRA01] L. P. A. FRANCA. Geradores de Artefatos: Implementação e Instanciação de Frameworks. *XV Simpósio Brasileiro de Engenharia de Software*, pages 302–315, 2001.
- [FRD88] R. FESTA, C. F. RATTO, and D. DEGOL. A Procedure to Obtain Average Daily Values of Meteorological Parameters From Monthly Averages. *Solar Energy*, 40(4):309–313, October 1988.
- [FS97] Mohamed E. FAYAD and Douglas C. SCHMITD. Object-Oriented Application Frameworks. *Communications of the ACM*, 40(10):32–38, October 1997.

- [GHEV93] E. GAMMA, R. HELM, JOHNSON R. E., and J. M. VLISSIDES. Design Patterns: Abstraction and Reuse of Object-Oriented Design. In *ECOOP '93: Proceedings of the 7th European Conference on Object-Oriented Programming*, pages 406–431, London, UK, 1993. Springer-Verlag.
- [GLRS04] Marco A. E. GALDINO, Jorge H. G. LIMA, Cláudio M. RIBEIRO, and Eduardo T. SERRA. O Contexto das Energias Renováveis no Brasil. *Revista da DIRENG (Energias Renováveis)*, pages 17–25, 2004.
- [GM95] D. GANGOPADHYAY and S. MITRA. Understanding Frameworks by Exploration of Exemplars. *International Workshop on C.A.S.E, IEEE*, pages 90–99, 1995.
- [GP95] A. P. C. GUIMARÃES and E. M. D. PEREIRA. Estudos Solarimétricos com Base na Definição do Mês Padrão e Seqüencial de Radiação Diária a partir de uma Biblioteca das Matrizes de Markov. In *Congresso Brasileiro de Engenharia Mecânica - COBEM*, Belo Horizonte, Brasil, 1995.
- [GPS⁺08] G. L. GUIDONI, F. PAPATELLA, M. A. SONG, E. PEREIRA, L. ZARATE, S. YOSHIOKA, and I. REIS. Solar Energy: A Framework for Solar Energy Applications. *6th IEEE International Conference on Industrial Informatics (INDIN)*, July 2008.
- [GUI07] Gustavo Ludovico GUIDONI. *Framework Para Construção de Sistemas de Energia Solar*. Mestrado, PUC Minas, Belo Horizonte, Brasil, 2007.
- [IU05] T. A. H. INATOMI and M. E. M. UDAETA. Análise dos Impactos Ambientais na Produção de Energia Dentro do Planejamento Integrado de Recursos. In *II Workshop Brasil-Japão em Energia, Meio Ambiente e Desenvolvimento Sustentável*, Campinas, 2005.
- [JF88] R. JOHNSON and E. B. FOOT. Designing Reusable Classes. *Journal of Object Oriented Programming*, 1(5):22–35, 1988.
- [JOH92] Ralph E. JOHNSON. Documenting Frameworks Using Patterns. In *OOP-SLA '92: Conference Proceedings on Object-oriented Programming Systems, Languages, and Applications*, pages 63–76, New York, NY, USA, 1992. ACM.
- [JOH97] Ralph E. JOHNSON. How Frameworks Compare to Other Object-Oriented Reuse Techniques. *Communications of the ACM*, 40(10):39–42, 1997.

- [JW98] R. E. JOHNSON and B. WOOLF. *Type Object in Languages of Program Design 3*, pages 47–65. Addison-Wesley, 1998.
- [KP88] G. E. KRASNER and S. T. POPE. A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk-80. *J. Object Oriented Program.*, 1(3):26–49, 1988.
- [MPAC05] Fernando R. MARTINS, Enio B. PEREIRA, Samuel L. ABREU, and Sérgio COLLE. Mapas de Irradiação Solar para o Brasil. Resultados do Projeto SWERA. *XII Simpósio Brasileiro de Sensoriamento Remoto*, pages 3137–3145, 2005.
- [MR05] José A. MARINI and Luiz A. ROSSI. Sistematização do Dimensionamento Técnico e Econômico de Sistemas Fotovoltaicos Isolados por Meio de Programa Computacional. *Engenharia Agrícola*, 25:67–75, 2005.
- [OCM00] A. ORTIGOSA, M. CAMPO, and R. MORIY. Towards Agent-Oriented Assistance for Framework Instantiation. *SIGPLAN Not.*, 35(10):253–263, 2000.
- [PPSS95] W. PREE, G. POMBERGER, A. SCHAPPERT, and P. SOMMERLAD. Active Guidance of Framework Development. *Software - Concepts and Tools*, 16(3):136–148, 1995.
- [PRE99] W. PREE. *Hot-Spot-Driven Development in Building Application Frameworks: Object-Oriented Foundations of Framework Design*, pages 379–393. John Wiley & Sons, 1999.
- [REB05] M. S. REBOITA. Introdução à Estatística Aplicada à Climatologia - Parte IV - Análise Harmônica e Espectral. Technical report, USP - Universidade de São Paulo, 2005. Desenvolvimento de material didático ou instrucional - Estatística Básica.
- [SCH97] H. A. SCHMID. Systematic Framework Design by Generalization. *Communications of the ACM*, 40(10):48–51, 1997.
- [SOL99] GRUPO DE TRABALHO DE ENERGIA SOLAR. Manual de Engenharia para Sistemas Fotovoltaicos. Technical report, CRESESB/CEPEL - Centro de Referência para Energia Solar e Eólica Sérgio de Salvo Brito/Centro de Pesquisas em Energia Elétrica, Novembro 1999. Manual técnico. Edição Revisada e Ampliada pelo CRESESB.

- [SP98] R. P. SILVA and R. T. PRICE. A Busca de Generalidade, Flexibilidade e Extensibilidade no Processo de Desenvolvimento de Frameworks Orientados a Objetos. In *Proceedings of Workshop Iberoamericano de Engenharia de Requisitos e Ambientes de Software (IDEAS'98)*., volume 2, pages 298–309, Torres, Brasil, 1998.
- [YF00] A. YASSIN and M. E. FAYAD. *Application Frameworks: A Survey in Domain-Specific Application Frameworks: Frameworks Experience by Industry*, chapter 29, pages 615–632. John Wiley & Sons, 2000.
- [ZLS00] R. ZILLES, E. LORENZO, and P. M. N. SERPA. From Candles to PV Electricity: A Four Year Experience at Iguape-Cananeia, Brazil. *Progress in Photovoltaics*, 8(4):421–434, 2000.

Apêndice A

Fórmulas do cálculo da distância para identificação do ano padrão

$$M_x(y, m) = \frac{1}{nd(m)} \sum_d X(y, m, d)$$

$$\sigma_x(y, m) = \sqrt{\frac{1}{nd(m)} \sum_d [X(y, m, d) - M_x(y, m)]^2}$$

$$\mu_{M_x}(m) = \frac{1}{ny} \sum_y M_x(y, m)$$

$$\sigma_{M_x}(m) = \sqrt{\frac{1}{ny} \sum_y [M_x(y, m) - \mu_{M_x}(m)]^2}$$

$$F_{y,m}(X_0) = \text{fraction of } X - \text{value_less_than } X_0$$

$$G_m(X_0) = \text{fraction of } X - \text{value_less_than } X_0$$

$$d_{av}(y, m) = |M_x(y, m) - \mu_{M_x}(m)|$$

$$d_{sd}(y, m) = |\sigma_x(y, m) - \sigma_{M_x}(m)|$$

$$d_{ks}(y, m) = \max_{X_0} |F_{y,m}(X_0) - G_m(X_0)|$$

$$d_{y,m} = (1 - \alpha - \beta)d_{ks}(y, m) + \alpha d_{av}(y, m) + \beta d_{sd}(y, m)$$

$$\alpha \rightarrow 0.1$$

$$\beta \rightarrow 0.1$$

Apêndice B

Banco de dados das matrizes de transição

$\overline{K}_t \leq 0.30$									
0.229	0.333	0.208	0.042	0.083	0.042	0.042	0.021	0.000	0.000
0.167	0.319	0.194	0.139	0.097	0.028	0.042	0.000	0.014	0.000
0.250	0.250	0.091	0.136	0.091	0.046	0.046	0.023	0.068	0.000
0.158	0.237	0.158	0.263	0.026	0.053	0.079	0.026	0.000	0.000
0.211	0.053	0.211	0.158	0.053	0.053	0.158	0.105	0.000	0.000
0.125	0.125	0.250	0.188	0.063	0.125	0.000	0.125	0.000	0.000
0.040	0.240	0.080	0.120	0.080	0.080	0.120	0.120	0.080	0.040
0.000	0.250	0.000	0.125	0.000	0.125	0.125	0.250	0.063	0.063
0.000	0.250	0.000	0.125	0.250	0.000	0.250	0.000	0.000	0.125
0.000	0.000	0.000	0.000	0.000	0.000	0.500	0.250	0.000	0.250

$0.30 < \overline{K}_t \leq 0.35$									
0.000	0.000	0.091	0.000	0.364	0.091	0.182	0.000	0.273	0.000
0.118	0.118	0.176	0.118	0.059	0.118	0.176	0.059	0.059	0.000
0.067	0.267	0.067	0.200	0.067	0.000	0.133	0.133	0.000	0.067
0.118	0.235	0.000	0.235	0.059	0.176	0.118	0.000	0.059	0.000
0.077	0.154	0.308	0.077	0.154	0.077	0.000	0.077	0.077	0.000
0.083	0.000	0.167	0.250	0.083	0.167	0.000	0.083	0.167	0.000
0.222	0.222	0.000	0.111	0.111	0.000	0.111	0.222	0.000	0.000
0.091	0.182	0.273	0.000	0.091	0.273	0.000	0.091	0.000	0.000
0.111	0.111	0.111	0.222	0.000	0.000	0.000	0.222	0.111	0.111
0.000	0.000	0.000	0.000	0.000	0.000	0.500	0.000	0.000	0.500

0.35 < $\bar{K}_t \leq 0.40$									
0.206	0.088	0.176	0.176	0.088	0.029	0.176	0.029	0.029	0.000
0.120	0.100	0.140	0.160	0.120	0.220	0.100	0.000	0.020	0.020
0.077	0.123	0.185	0.123	0.077	0.139	0.092	0.123	0.061	0.000
0.048	0.111	0.095	0.206	0.206	0.190	0.095	0.048	0.000	0.000
0.059	0.137	0.118	0.137	0.098	0.118	0.118	0.157	0.059	0.000
0.014	0.097	0.139	0.153	0.125	0.139	0.208	0.056	0.042	0.028
0.073	0.101	0.116	0.145	0.087	0.159	0.203	0.087	0.029	0.000
0.019	0.037	0.111	0.056	0.074	0.111	0.185	0.296	0.074	0.037
0.035	0.069	0.035	0.000	0.035	0.103	0.172	0.138	0.379	0.035
0.000	0.167	0.167	0.000	0.167	0.000	0.000	0.333	0.000	0.167

0.40 < $\bar{K}_t \leq 0.45$									
0.167	0.167	0.167	0.000	0.083	0.125	0.000	0.167	0.125	0.000
0.117	0.117	0.150	0.117	0.083	0.117	0.200	0.067	0.017	0.017
0.049	0.085	0.134	0.158	0.098	0.110	0.134	0.134	0.061	0.037
0.039	0.090	0.141	0.141	0.167	0.141	0.090	0.141	0.039	0.013
0.009	0.139	0.074	0.093	0.194	0.139	0.167	0.093	0.074	0.019
0.036	0.018	0.117	0.099	0.144	0.180	0.180	0.117	0.072	0.036
0.000	0.046	0.061	0.061	0.136	0.159	0.273	0.167	0.098	0.000
0.016	0.056	0.080	0.128	0.104	0.080	0.160	0.208	0.136	0.032
0.011	0.053	0.021	0.043	0.128	0.096	0.074	0.223	0.277	0.074
0.000	0.074	0.037	0.000	0.074	0.074	0.074	0.074	0.333	0.259

0.45 < $\bar{K}_t \leq 0.50$									
0.120	0.200	0.160	0.120	0.120	0.120	0.080	0.000	0.040	0.040
0.100	0.080	0.120	0.140	0.140	0.200	0.180	0.040	0.000	0.000
0.046	0.114	0.068	0.171	0.125	0.171	0.080	0.159	0.057	0.011
0.015	0.061	0.084	0.099	0.191	0.153	0.153	0.115	0.115	0.015
0.024	0.030	0.098	0.098	0.165	0.195	0.195	0.140	0.043	0.012
0.015	0.026	0.062	0.124	0.144	0.170	0.170	0.222	0.062	0.005
0.000	0.013	0.045	0.108	0.112	0.175	0.188	0.224	0.117	0.018
0.008	0.023	0.054	0.066	0.093	0.125	0.191	0.253	0.183	0.004
0.006	0.022	0.061	0.033	0.067	0.083	0.139	0.222	0.322	0.044
0.000	0.046	0.091	0.091	0.046	0.046	0.136	0.091	0.273	0.182

0.50 < $\bar{K}_t \leq 0.55$									
0.250	0.179	0.107	0.107	0.143	0.071	0.107	0.036	0.000	0.000
0.133	0.022	0.089	0.111	0.156	0.178	0.111	0.133	0.067	0.000
0.064	0.048	0.143	0.048	0.175	0.143	0.206	0.095	0.079	0.000
0.000	0.022	0.078	0.111	0.156	0.156	0.244	0.167	0.044	0.022
0.016	0.027	0.037	0.069	0.160	0.219	0.230	0.160	0.075	0.005
0.013	0.025	0.030	0.093	0.144	0.202	0.215	0.219	0.055	0.004
0.006	0.041	0.035	0.064	0.090	0.180	0.337	0.192	0.049	0.006
0.012	0.021	0.029	0.035	0.132	0.123	0.184	0.371	0.082	0.012
0.008	0.016	0.016	0.024	0.071	0.103	0.159	0.270	0.309	0.024
0.000	0.000	0.000	0.000	0.059	0.000	0.059	0.294	0.412	0.176

0.55 < $\bar{K}_t \leq 0.60$									
0.217	0.087	0.000	0.174	0.130	0.087	0.087	0.130	0.087	0.000
0.026	0.079	0.132	0.079	0.026	0.158	0.158	0.132	0.158	0.053
0.020	0.020	0.020	0.040	0.160	0.180	0.160	0.200	0.100	0.100
0.025	0.013	0.038	0.076	0.076	0.139	0.139	0.266	0.215	0.013
0.030	0.030	0.050	0.020	0.091	0.131	0.162	0.283	0.131	0.071
0.006	0.006	0.013	0.057	0.057	0.121	0.204	0.287	0.185	0.064
0.004	0.026	0.037	0.030	0.093	0.107	0.193	0.307	0.167	0.037
0.011	0.009	0.014	0.042	0.041	0.071	0.152	0.418	0.203	0.041
0.012	0.022	0.022	0.038	0.019	0.050	0.113	0.281	0.360	0.084
0.008	0.024	0.039	0.039	0.063	0.039	0.118	0.118	0.284	0.268

0.60 < $\bar{K}_t \leq 0.65$									
0.067	0.133	0.133	0.067	0.067	0.200	0.133	0.133	0.067	0.000
0.118	0.059	0.059	0.059	0.059	0.118	0.118	0.235	0.118	0.059
0.000	0.024	0.024	0.049	0.146	0.073	0.195	0.244	0.195	0.049
0.026	0.000	0.026	0.026	0.053	0.184	0.263	0.184	0.237	0.000
0.014	0.000	0.042	0.056	0.069	0.097	0.139	0.306	0.278	0.000
0.009	0.009	0.052	0.069	0.052	0.112	0.215	0.285	0.138	0.060
0.009	0.009	0.026	0.017	0.094	0.099	0.232	0.283	0.210	0.021
0.010	0.014	0.016	0.019	0.027	0.062	0.163	0.467	0.202	0.019
0.004	0.007	0.031	0.017	0.033	0.050	0.086	0.252	0.469	0.050
0.000	0.000	0.015	0.046	0.031	0.046	0.077	0.123	0.446	0.215

0.65 < $\bar{K}_t \leq 0.70$									
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.250	0.250	0.500	0.000
0.000	0.000	0.000	0.000	0.250	0.000	0.000	0.375	0.250	0.125
0.000	0.000	0.000	0.083	0.000	0.167	0.167	0.250	0.333	0.000
0.000	0.000	0.042	0.042	0.042	0.083	0.083	0.292	0.292	0.125
0.000	0.000	0.032	0.000	0.000	0.032	0.129	0.387	0.355	0.065
0.000	0.000	0.000	0.038	0.038	0.075	0.047	0.340	0.415	0.047
0.004	0.004	0.007	0.007	0.011	0.030	0.052	0.141	0.654	0.089
0.000	0.000	0.000	0.000	0.061	0.061	0.030	0.030	0.349	0.470

$\bar{K}_t > 0.70$									
0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	1.000	0.000
0.100	0.100	0.100	0.100	0.100	0.100	0.100	0.100	0.100	0.100
0.000	0.000	0.000	0.250	0.000	0.000	0.000	0.500	0.250	0.000
0.000	0.000	0.143	0.143	0.000	0.143	0.143	0.429	0.000	0.000
0.000	0.000	0.000	0.200	0.000	0.000	0.200	0.400	0.200	0.000
0.000	0.000	0.000	0.000	0.000	0.000	0.222	0.444	0.333	0.000
0.000	0.000	0.000	0.000	0.080	0.080	0.080	0.480	0.240	0.040
0.000	0.000	0.027	0.009	0.027	0.018	0.135	0.523	0.252	0.009
0.000	0.000	0.000	0.022	0.000	0.043	0.043	0.326	0.511	0.054
0.000	0.000	0.000	0.143	0.000	0.000	0.000	0.143	0.714	0.000

Apêndice C

Fórmulas da Radiação Solar

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

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

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

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

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

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

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

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

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

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

$$I_d = r_d H_d$$

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

$$I = H r_t$$

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

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

$$\begin{aligned} \cos \theta = & \sin \delta \sin \phi \cos \beta - \sin \delta \cos \phi \sin \beta \cos \gamma + \cos \delta \cos \phi \cos \beta \cos \omega + \\ & \cos \delta \sin \phi \sin \beta \cos \gamma \cos \omega + \cos \delta \sin \beta \sin \gamma \sin \omega \end{aligned}$$

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

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

Apêndice D

Coeficientes empíricos de Bunnnett

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 E

Dimensionamento Estocástico x Determinístico

Conforme mencionado no Capítulo 5, foi realizado um estudo de casos para comparar os resultados obtidos entre os dimensionamentos determinístico, realizado por este trabalho, e estocástico, realizado por Cabral [CAB06]. As duas abordagens utilizam as mesmas configurações para os dispositivos no diagrama de execução e para a demanda. O processo será descrito a seguir.

Cabral define dimensionamentos determinísticos e estocásticos como:

- **Determinístico** Caso não exista no modelo de simulação qualquer variável probabilística, é denominado determinístico. Nestes modelos a saída é determinada de forma direta, uma vez que uma série de entradas e relações são especificadas.
- **Estocásticos** Modelos matemáticos que descrevem sistemas que variam no tempo de forma aleatória são chamados de estocásticos. Um processo estocástico é um conjunto de variáveis randômicas ordenadas no tempo, que são definidas sobre um espaço de amostragem comum. O conjunto de todos os possíveis valores que estas variáveis randômicas podem assumir é chamado de espaço de estado. Os sistemas mais complexos no mundo real, com elementos estocásticos, não podem ser precisamente descritos por um modelo matemático que possa ser avaliado analiticamente. Neste caso, a simulação é freqüentemente o único tipo de investigação possível. Entretanto, modelos de simulação estocástica fornecem saídas aleatórias e devem, por isso, ser tratados como somente uma estimativa das características verdadeiras do modelo.

Para exemplificar essas definições, serão apresentados, a seguir, os passos para a realização do estudo de casos.

Em ambos os dimensionamentos, foi utilizada a demanda de carga, conforme descrito na Tabela E.1.

Tabela E.1: Cargas consideradas para o dimensionamento do sistema fotovoltaico

Cargas	Quantidade	Potência unitária (W)	Potência total (W)
Lâmpadas 32W	8	32	256
Lâmpadas 16W	4	16	64

Esta demanda representa uma escola hipotética cujo o horário de funcionamento estende-se entre 18:00 e 22:00, de segunda à sexta-feira. Feriados e período de férias escolares foram considerados dias úteis, existindo demanda de carga para os mesmos. Finais de semana foram desconsiderados. Não foi aplicado nenhum desvio padrão para os valores de demandas mensais, portanto o consumo foi considerado constante ao longo do ano e a unidade de medida considerada foi W/h .

Foram utilizados por Cabral os dados de irradiância e conseqüentemente de identificação do ano de referência para a cidade de Viçosa - Minas Gerais. Porém, os dados medidos para estimativa de radiação, deste caso de testes, correspondem aos mesmos utilizados nas residências rurais, fornecidos pela *CEMIG*, na identificação do ano padrão para a cidade de Belo Horizonte. Pretende-se desta forma, avaliar o comportamento da demanda escolar com os respectivos dados de energia solar.

De posse dos dados de irradiância, pode-se escolher o módulo fotovoltaico a ser utilizado no dimensionamento. O diagrama de execução foi configurado pelos dispositivos descritos a seguir.

O painel solar utilizado foi o *I-100*, do fabricante *Isofoton* e suas propriedades são apresentadas pela Tabela E.2

Tabela E.2: Especificação do Painel Solar I-100

Área do painel (m^2)	0,857
Número de células do módulo	36
Tensão de circuito aberto (V)	21,6
Tensão de máxima potência (V)	17,4
Corrente de curto-circuito (A)	6,54
Corrente de máxima potência (A)	5,74
Coefficiente alfa de temperatura (α)	0,000416
Coefficiente beta de temperatura (β)	-0,0782
Potência de referência (W)	100

As condições padrão de operação para a irradiância e a temperatura solar de referência, respectivamente, são $1000W/m^2$ e $25^\circ C$.

A bateria dimensionada foi a do fabricante *Concorde*, modelo *PVX-890T*. Sua especificação é apresentada pela Tabela E.3.

Tabela E.3: Propriedades do componente bateria PVX-890T

Energia inicial de carga (Capacidade em 20h) (W)	1068
Energia armazenada máxima (% prof. mín)	5
Energia armazenada mínima (% prof. máx)	80
Eficiência (%)	95

O controlador de carga utilizado para este estudo de caso foi o mesmo da demanda fornecida pela *CEMIG*, modelo *Sollis* e fabricante *Varixx*.

O projeto do inversor corresponde ao modelo *Sine wave inverter*, desenvolvido pelo fabricante *XANTREX* e sua eficiência de conversão é de 88%.

As simulações foram executadas com a seguinte configuração: 2 painéis solares e 10 baterias para a unidade de armazenamento.

A Figura E.1 apresenta, como resultado da simulação no sistema *Sol Brasil*, o gráfico de energia armazenada nas baterias, por meio da carga produzida pelos módulos fotovoltaicos, no dimensionamento determinístico, para o ano de referência identificado.

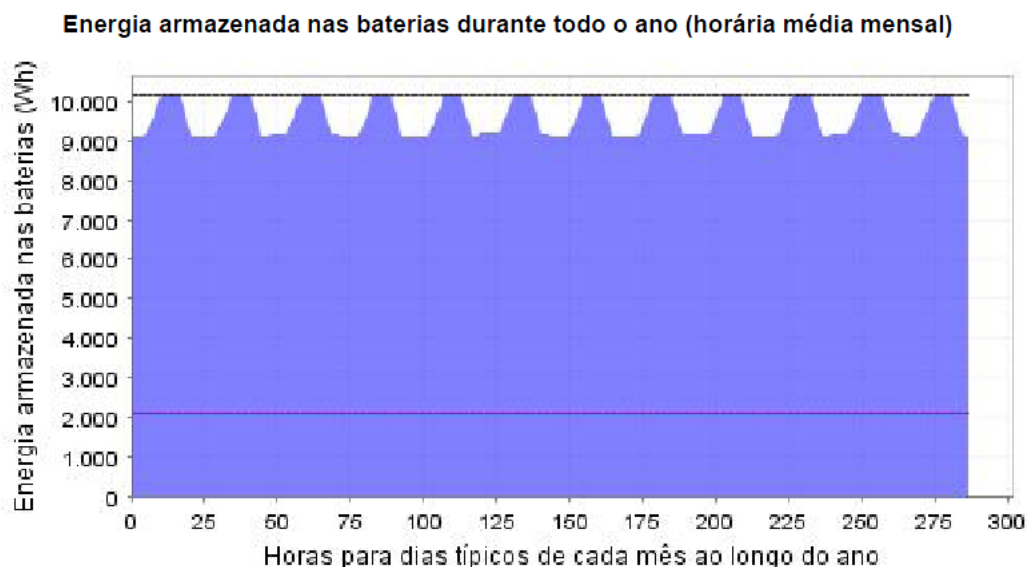


Figura E.1: Energia armazenada nas baterias pelo processo determinístico

Por sua vez, a Figura E.2 apresenta o gráfico de armazenamento de energia nas baterias, pela carga produzida através dos módulos fotovoltaicos, no dimensionamento realizado no trabalho proposto por Cabral. Os dados foram analisados por um programa

computacional, escrito na plataforma MatLab, versão 6.5.

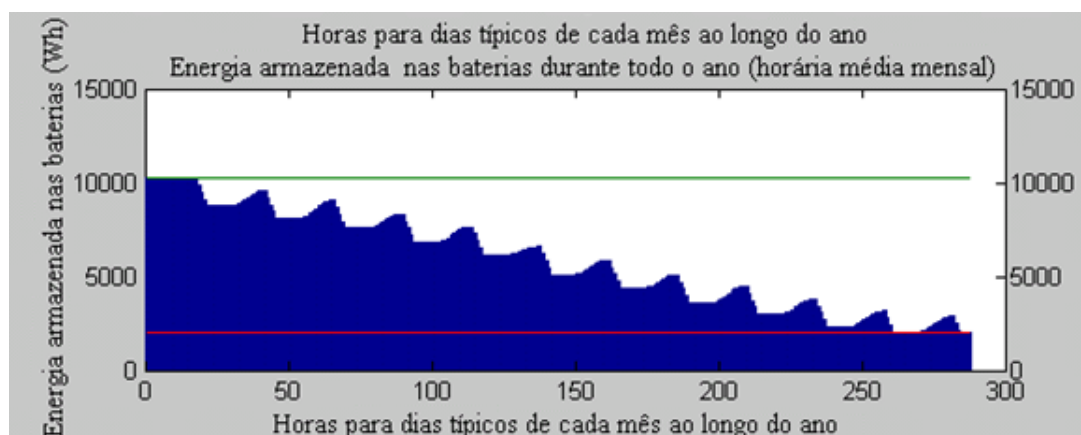


Figura E.2: Energia armazenada nas baterias pelo processo estocástico

Pode-se perceber que a energia total armazenada pelo banco de baterias, a profundidade máxima de descarga e o valor máximo de sobrecarga são equivalentes nos dois gráficos, comprovando a coerência das execuções.

Entretanto, o modelo estocástico leva em consideração um fator probabilístico de perda de fornecimento de energia, resultando na análise do processo de depreciação do dispositivo captador de luz solar e conseqüentemente na análise do custo e viabilidade das configurações informadas durante o dimensionamento. Já o modelo implementado por este trabalho, por não aplicar metodologias para análise da viabilidade econômica e também fatores estatísticos de perda de capacidade dos componentes, apresentou uma curva constante para o fornecimento de energia ao banco de baterias, ao longo do ano.

O modelo estocástico identifica os estados nos quais a demanda de carga é insatisfeita. Por sua vez, o dimensionamento determinístico não analisa o desempenho real dos dispositivos, ao longo de seu funcionamento, determinando insatisfatoriamente o projeto de sistema que melhor satisfaça o nível de desempenho do serviço requerido.

Pode-se concluir que o dimensionamento estocástico realiza uma melhor previsão do arranjo fotovoltaico por ser mais preciso, rico em detalhes e mais próximo do real.

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](#)