
NOTA TÉCNICA:**UTILIZAÇÃO DAS HEURÍSTICAS DE NIELSEN PARA DESENVOLVIMENTO DE UM SOFTWARE PARA AUXÍLIO AO DIMENSIONAMENTO DE BIODIGESTORES**

Luiz Henrique Faleiro Sidney¹, Jaqueline de Oliveira Castro², Tadayuki Yanagi Junior³, Vitor Hugo Teixeira⁴, Yamid Fabian Hernández Julio⁵

RESUMO

Objetivou-se com esse trabalho desenvolver um programa computacional para o auxílio ao dimensionamento de biodigestores modelos indiano e chinês. O programa foi desenvolvido em linguagem de programação Java, com base em análises de fluxogramas de cálculos que detalham os procedimentos de cada modelo. Para dimensionar o biodigestor, o usuário insere valores das variáveis necessárias para os cálculos, em telas distintas. Os processos do dimensionamento são distintos para o biodigestor do tipo chinês e do tipo indiano, bem como as telas apresentadas. Após o processamento, os resultados são apresentados de forma resumida ou memorial descritivo completo. O programa desenvolvido caracteriza-se pela interface interativa utilizando as heurísticas de Nielsen e processamento rápido, facilitando a sua utilização por profissionais envolvidos no dimensionamento de biodigestores.

Palavras-chave: biodigestores, aplicativo computacional, usabilidade

ABSTRACT**USE OF NIELSEN HEURISTICS TO DEVELOP A SOFTWARE TO AID BIODIGESTER DESIGN**

This study reports development of a computer program to help design models of Indian and Chinese biodigesters. The program was developed in Java programming language, based on analysis of calculation's flow charts that give details of procedures for each model. To calculate the biodigester, the user enters the required values in different screens, considering the peculiarities of each biodigester. After processing, the results are presented as a summary or as a complete descriptive report. The software is characterized by interactive interface using the Nielsen's heuristics and fast processing, that facilitates its use by professionals involved in designing biodigesters.

Keywords: biodigesters, computer software, usability

Recebido para publicação em 14/09/2011. Aprovado em 10/07/2012.

1- Graduado em Sistemas de Informação. luiz.henrique.faleiro@gmail.com

2- Doutoranda em Engenharia Agrícola na Universidade Federal de Lavras – UFLA. jaqueline.oliveiracastro@yahoo.com.br

3- Engenheiro Agrícola, Prof. Associado da Universidade Federal de Lavras – UFLA. yanagi@deg.ufla.br

4- Engenheiro Agrícola, Prof. Associado da Universidade Federal de Lavras – UFLA. vht@deg.ufla.br

5- Licenciatura em Informática Educativa e Médios Audiovisuais, Mestre em Engenharia de Sistemas. yafaheju@posgrad.ufla.br

INTRODUÇÃO

Resíduos agrícolas não tratados podem gerar vários problemas ambientais quando lançados, sem tratamento, em cursos de água (ALVAREZ; LIDÉN, 2008). O tratamento por meio de processos anaeróbios (SOUZA *et al.*, 2008), reduz os impactos ambientais, recicla os efluentes produzidos por sistemas de produção agrícola, possibilita o uso como biofertilizantes e produz o biogás, que é uma fonte alternativa de energia (COLDEBELLA *et al.*, 2006).

O desenvolvimento de tecnologias que utilizam fontes renováveis de energia são atrativas, tanto ambientalmente, quanto socialmente, pois possibilitam a criação de fontes de energia descentralizadas e em pequena escala, sendo, fundamental para o desenvolvimento sustentável em países desenvolvidos e em desenvolvimento (COLDEBELLA *et al.*, 2006). O biodigestor pode ser definido como uma câmara de fermentação onde a biomassa sofre digestão anaeróbica e que tem por finalidades a obtenção do biogás e o biofertilizante (MACINTYRE, 1996), apresenta-se como uma alternativa para o tratamento de resíduos.

Os biodigestores podem ser divididos de acordo com sua complexidade, país em que foi desenvolvido e a forma que será aplicado em cada situação. Os modelos mais comuns encontrados no Brasil são os modelos chinês e indiano, encontrados comumente em propriedades de pequeno e médio porte. Tais biodigestores são bastante dinâmicos, pois aceitam o uso de diferentes resíduos, além de seus projetos não demandarem alto custo. Em seus países de origem, China e Índia, eram destinados, na maioria das vezes, para uso doméstico, sendo empregados na produção de biogás e biofertilizante (BENICASA *et al.*, 1991).

A implantação de biodigestores para o tratamento de resíduos em áreas rurais requer conhecimento sobre digestão anaeróbica, dimensionamento, construção e manejo, além de se necessitar de área e resíduo suficiente para manutenção e funcionamento

do biodigestor (FLORENTINO, 2003). Tendo em vista a complexidade de dimensionamento de biodigestores, o desenvolvimento de um *software* de fácil utilização, que auxilie o dimensionamento, é de grande importância para a eliminação de erros nos cálculos e para agilizar o dimensionamento, possibilitando o estudo de diversas situações específicas.

As características necessárias para uma interface atingir a qualidade são denominadas “usabilidade”. A usabilidade é um componente importante para qualquer produto de *software*, pois possibilita aumento na produtividade e diminui os erros em sua utilização (ZEN *et al.*, 2008; SANTOS; TEIXEIRA, 2010). De acordo com Santos e Teixeira (2010), as heurísticas propostas por Nielsen, em 1994, englobam os principais atributos envolvidos para a qualidade da usabilidade de um produto de *software*.

Nielsen criou dez princípios gerais, a fim de guiar a avaliação e criação de interfaces. De acordo com Nielsen (2005), estes princípios são denominados heurísticas, pois são melhor inseridas nas naturezas das regras de ouro do que nas diretrizes específicas de usabilidades.

Objetivou-se, com o presente trabalho, desenvolver um *software* de boa usabilidade para o dimensionamento de biodigestores indiano ou chinês, avaliado por meio das Heurísticas de Nielsen.

MATERIAL E MÉTODOS

A linguagem Java foi escolhida para o desenvolvimento do *software*, por ser orientada a objetos, pela sua robustez e versatilidade, além de poder ser executada em qualquer sistema operacional, desde que contenha a máquina virtual Java instalada (GOSLING *et al.*, 2000).

O modelo de processo do *software* desenvolvido foi em cascata, que é recomendado para o desenvolvimento de projetos simples, que não demandem grande equipe (SOARES, 2009). Este modelo é dividido classicamente em cinco fases: análise de requerimentos, *design* do *software*, implementação, integração e teste do sistema, validação e manutenção (Figura 1).

No desenvolvimento desse modelo, cada etapa deve ser concluída antes do início da próxima etapa (SOMMERVILLE, 2006; PETERSEN *et al.*, 2009).

Foram estabelecidos os requisitos e o design do software, que consiste na representação dos recursos necessários no software, nesse caso a capacidade de dimensionar biodigestores dos modelos indiano e chinês com uma interface de boa usabilidade. Para maior coesão e menor acoplamento no design, o software foi dividido em três camadas: na primeira camada foi desenvolvida a interface gráfica; na segunda camada o relatório, na qual foram desenvolvidos os possíveis relatórios a serem gerados pelo usuário, e; a terceira camada foram as regras de negócio, na qual foi desenvolvido os algoritmos para dimensionamento dos biodigestores (SOMMERVILLE, 2006; ZHAOYIN *et al.*, 2009).

A metodologia proposta por Teixeira (1998) foi utilizada para desenvolver os fluxogramas de cálculo dos dois diferentes modelos de biodigestores (chinês e indiano) que foram introduzidos no software (Figuras 2 e 3, respectivamente). Essa metodologia foi baseada no método de tentativa e erro. Foram atribuídos valores aos parâmetros iniciais, como H (altura do corpo cilíndrico, em metros) e Di (diâmetro interno, em metros) no modelo indiano e, caso esses ou outros valores não satisfizessem as restrições, novos valores seriam atribuídos aos parâmetros. Esse modelo foi escolhido em

função de possibilitar a interferência no projeto, possibilitando a flexibilidade do sistema.

Os modelos de biodigestor indiano e chinês possuem diversas variáveis para efetuação dos cálculos em seus dimensionamentos. No modelo chinês, H é a altura do corpo cilíndrico, Hf é a altura da calota do fundo, Hg é a altura da calota do gasômetro, Hs é a altura da caixa de saída, He é a altura da caixa de entrada, Dc é o diâmetro do corpo cilíndrico, Ds é o diâmetro da caixa de saída, De é o diâmetro da caixa de entrada, Rf é o raio da calota do fundo, Rg é o raio da calota do gasômetro, Pmax é a pressão máxima e A é a diferença entre a altura do corpo cilíndrico e a superfície do solo. Já no modelo indiano, H é a altura do nível do substrato, h é a altura da parede divisória, H1 é altura ociosa do gasômetro, H2 é a altura útil do gasômetro, Hc é a altura da caixa de carga, He é a altura do corpo cilíndrico, Db é o diâmetro de base (sapata) de sustentação do biodigestor, Dg é o diâmetro do gasômetro, Di é o diâmetro interno do biodigestor, Ds é o diâmetro da caixa de saída, Dcq é o diâmetro da caixa quadrada, Dc é o diâmetro da caixa de carga, Pg é o peso do gasômetro, De é o diâmetro externo do biodigestor e Ac é área da caixa de carga.

Os fluxogramas desenvolvidos foram utilizados para realizar procedimentos e funções do software que satisfizessem todos os requerimentos para o dimensionamento de biodigestores dos modelos indiano e chinês.

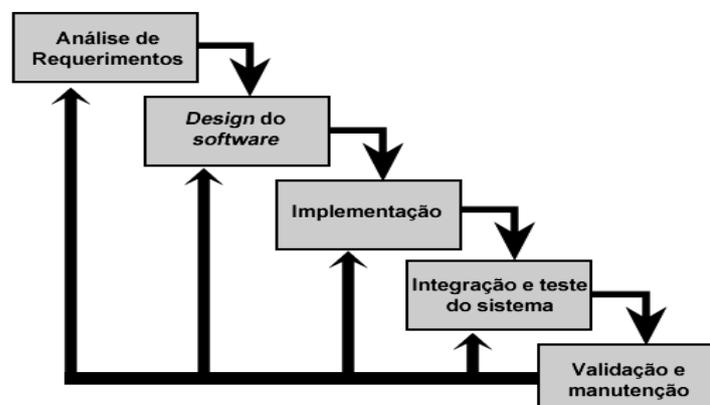


Figura 1. Modelo Cascata.

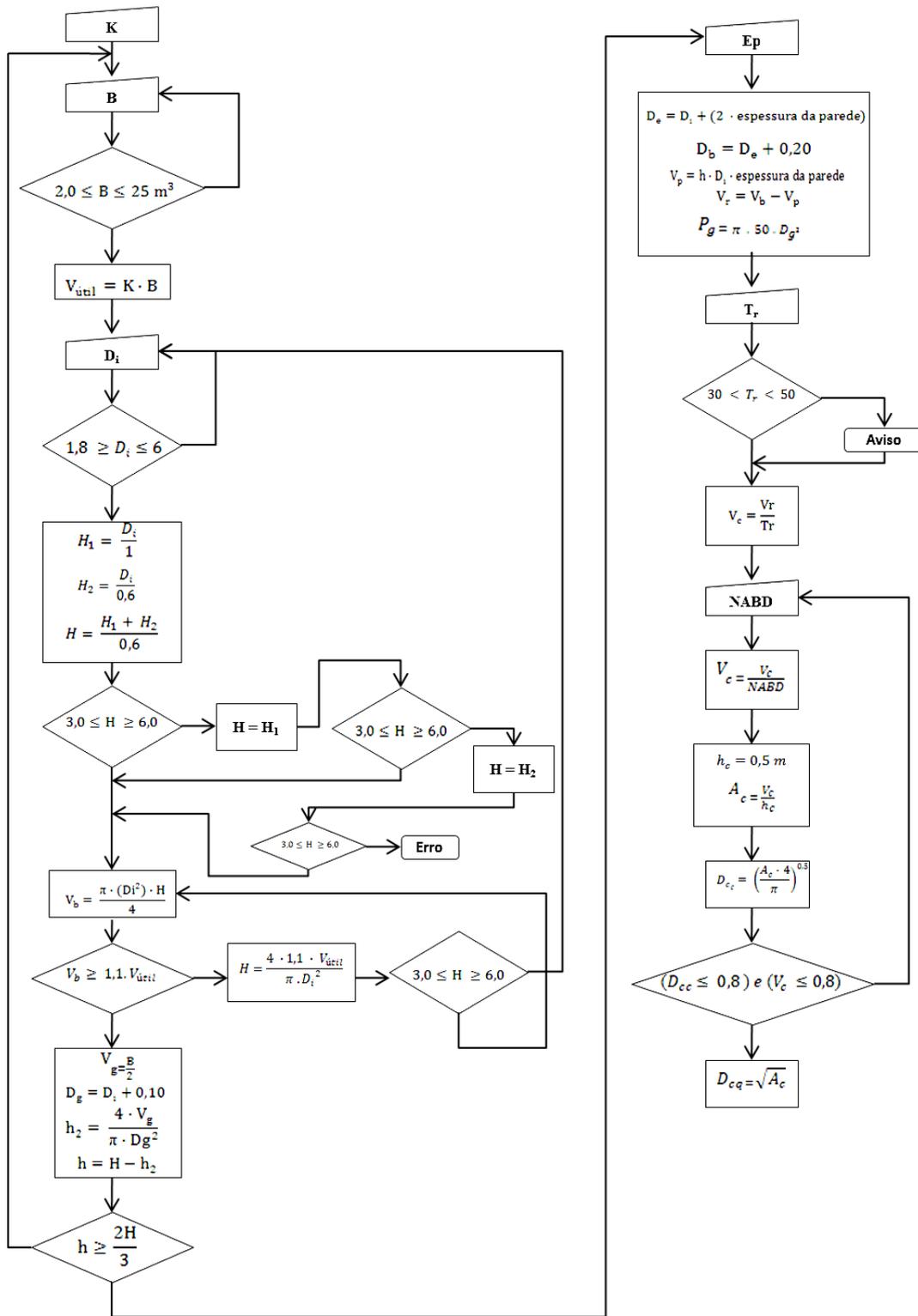


Figura 2. Fluxograma utilizado para o dimensionamento do biodigestor modelo indiano.

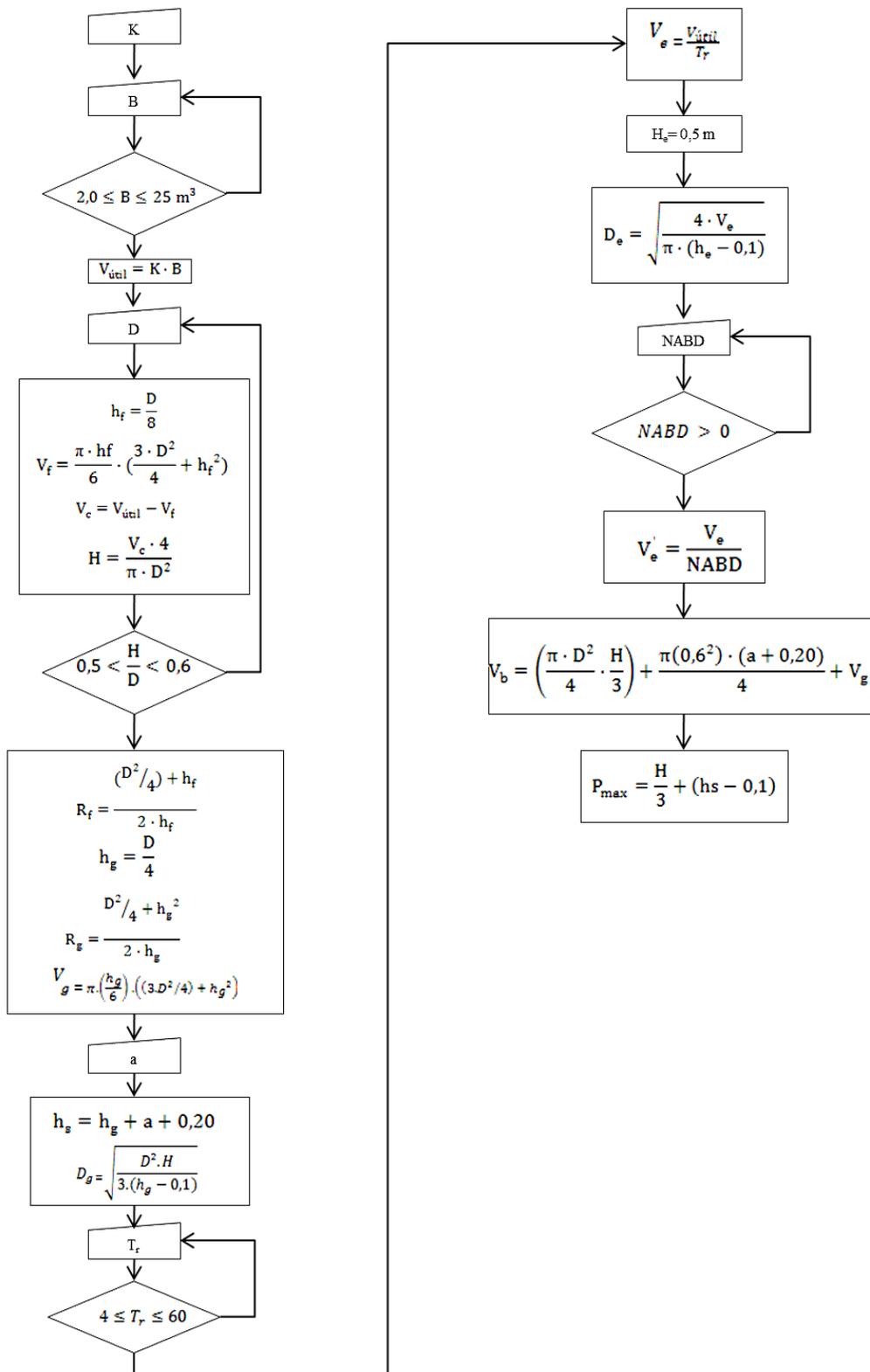


Figura 3. Fluxograma utilizado para o dimensionamento do biodigestor modelo chinês.

Na etapa de implementação, foi realizada toda a codificação do software. O ambiente de desenvolvimento integrado (IDE) NetBeans 6.8 (Oracle Corporation©, 2010) foi utilizado por ser um ambiente gratuito e que apresenta suporte ao desenvolvimento em linguagem Java. Neste ambiente, encontram-se ferramentas suficientes para criar aplicativos desktop, além de facilitar a criação das interfaces por meio do Swing, que é uma interface de programação de aplicativos (API) do Java para interfaces gráficas. Na API Swing, o desenvolvedor aloca os componentes desejados e destina onde eles devem ser alocados na interface gráfica do aplicativo e sua biblioteca gera automaticamente o código-fonte necessário.

Testes unitários foram implementados utilizando o JUnit, que é um framework open-source que auxilia a criação e execução de testes unitários sobre classes Java. Segundo Pocatilu (2008), o framework JUnit possui grande eficiência e é muito utilizado para a realização de testes em projetos de aplicações em Java.

A interface gráfica do software foi desenvolvida baseada nas heurísticas de Nielsen (NIELSEN, 2005). Estas preconizam que um programa deve apresentar visibilidade do status, compatibilidade entre sistema e mundo real, controle e liberdade para o usuário, consistência e padrões, prevenção de erros, reconhecimento em lugar de lembrança, flexibilidade e eficiência de uso, estética e design minimalista, auxiliar os usuários a reconhecer, diagnosticar e recuperar erros, além de ajuda e documentação, para que assim se crie uma interface de boa usabilidade.

Após a implementação do software, foi realizada a integração e teste do sistema, que constitui na junção dos módulos que foram construídos separadamente e do teste integral do sistema. A divisão do software em camadas (interface, relatório e regra de negócios) foi realizada com objetivo de se ter o máximo de acoplamento e o mínimo de coesão. Após sua junção foram feitos testes de caixa branca, para avaliar o comportamento interno dos componentes do software, e de caixa preta, com

objetivo de avaliar o comportamento externo do software.

No teste de caixa preta o software passa por baterias de teste sem que sejam avaliadas as funcionalidades em nível de usuário, ou seja, o testador utiliza o software como um usuário final com intuito de achar algum erro. No teste de caixa branca o testador avalia o *software* com relação ao código, testando se os métodos implementados retornam os valores desejados a partir de entradas controladas. Este tipo de teste é realizado com o monitoramento linha a linha durante a execução do *software*.

RESULTADOS E DISCUSSÃO

Iniciado o *software*, o usuário escolhe o modelo de biodigestor que deseja utilizar: indiano ou chinês (Figura 4).

Para o dimensionamento, o usuário insere os valores solicitados: fator de rendimento da biomassa (K , m^3 da biomassa dividido por m^3 do biogás produzido) (Figura 5), consumo diário de biogás (B , $m^3 \text{ dia}^{-1}$) (Figura 6), ou seja, quantidade diária de biogás necessária para manter em funcionamento todos os equipamentos; diâmetro interno do biodigestor (D_i , m), tempo de retenção (T_r , dias) e número de abastecimentos diários para equivaler ao volume total de carga (NABD). Devido às peculiaridades de cada biodigestor, foi necessária uma entrada diferente para cada modelo (chinês e indiano).

Ao optar pelo modelo chinês encontrar-se-á a variável “A”, que é a diferença, em metros, entre a altura do corpo cilíndrico e a superfície do solo. Já para a o cálculo do modelo indiano encontrar-se-á a variável “ E_p ”, que é a espessura da parede divisória em metros. Telas de entrada de dados distintas foram desenvolvidas devido à diferença entre os modelos, mas manteve-se a padronização das telas (mesma aparência).

Após o processamento, os resultados são apresentados de forma resumida ou na forma de memorial descritivo completo, permitindo a utilização do programa, tanto em projetos, quanto para fins didáticos.

O programa computacional desenvolvido

caracteriza-se pela interface interativa e processamento rápido, facilitando a sua utilização por profissionais envolvidos no dimensionamento de biodigestores. Em todas as diferentes telas criadas para interface gráfica do *software* foram utilizadas as heurísticas de Nielsen (NIELSEN, 2005), com objetivo de

criar uma boa usabilidade com o usuário. Para demonstrar a interface gráfica do programa, as heurísticas serão citadas e descritas como foram aplicadas no presente programa.

Na Figura 7 está apresentada a tela de entrada da variável D_i no dimensionamento do biodigestor do modelo indiano.

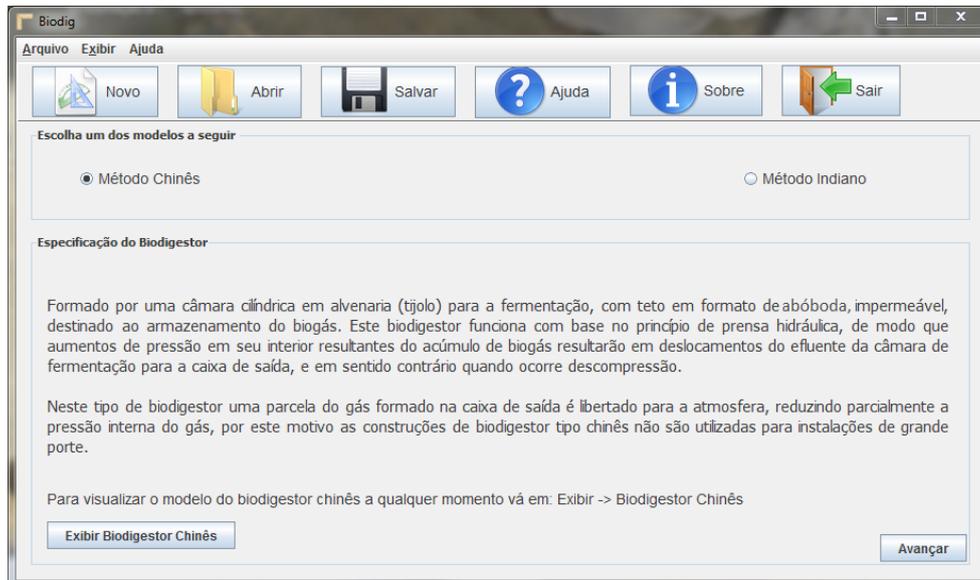


Figura 4. Tela de escolha de modelo de biodigestor a ser dimensionado.

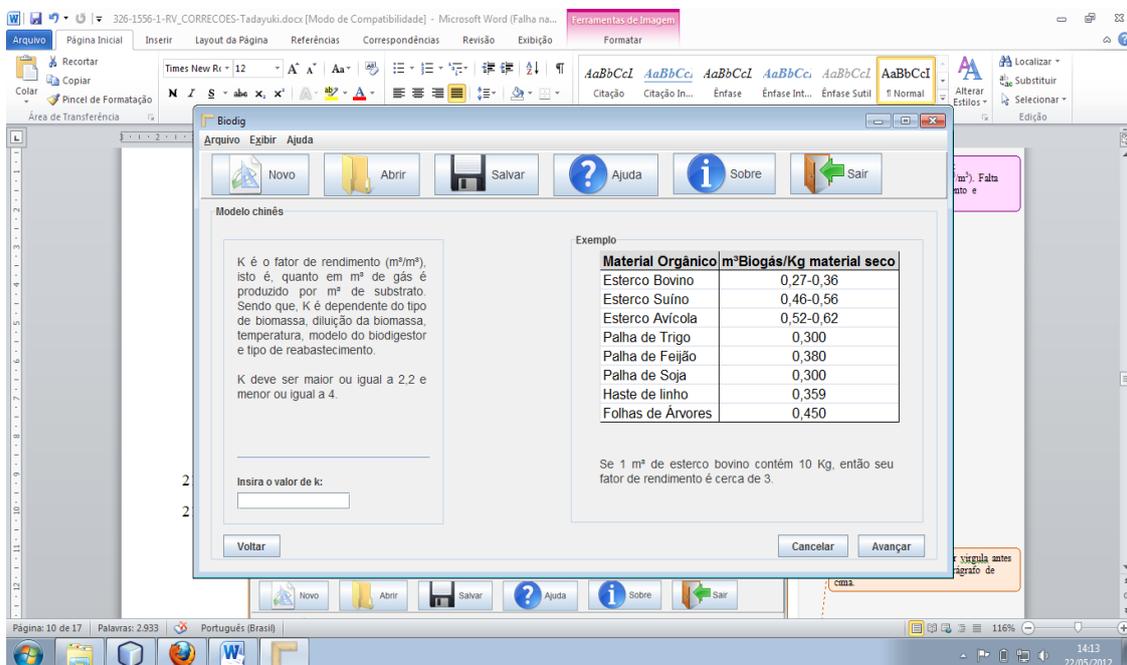


Figura 5. Entrada da variável “K”.

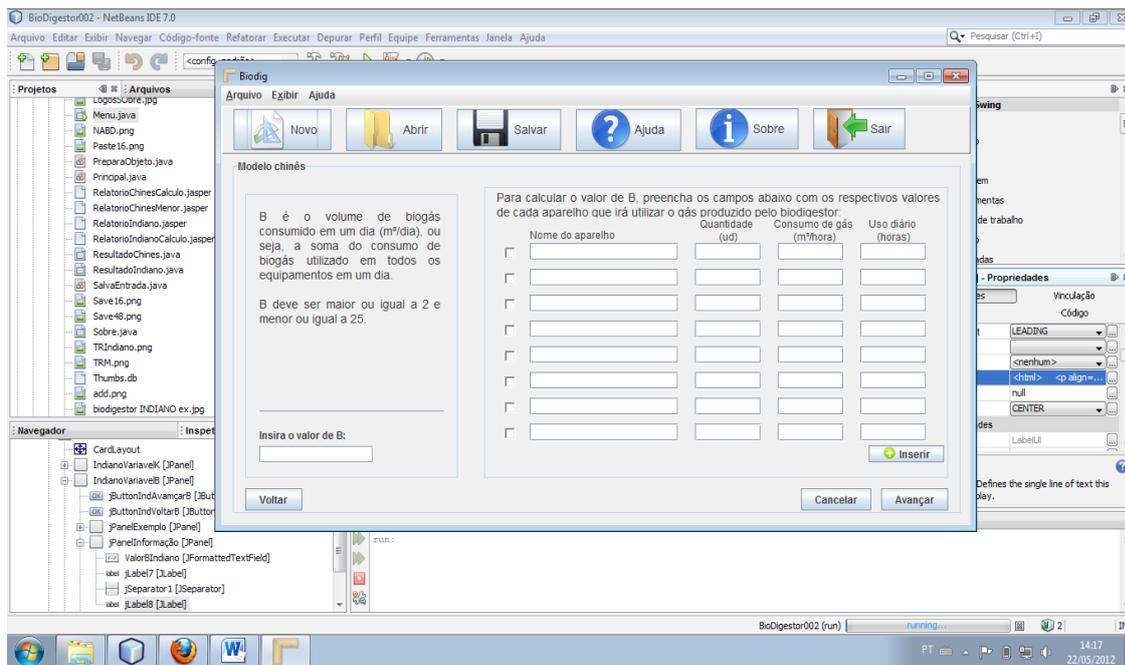


Figura 6. Entrada da variável “B”.

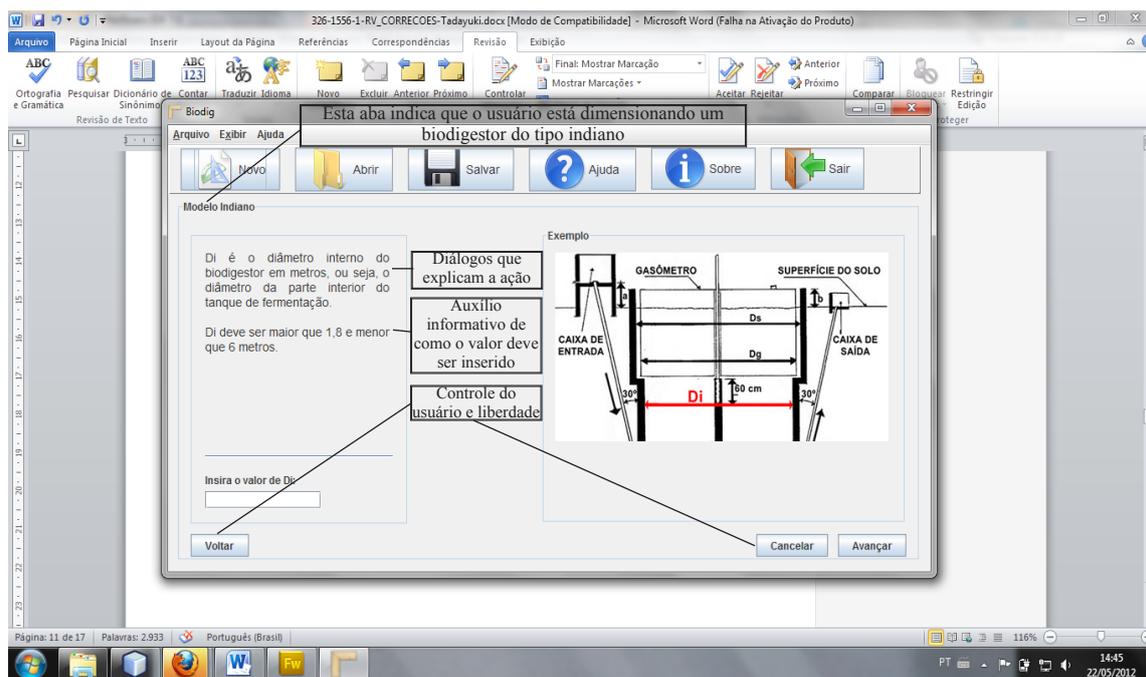


Figura 7. Tela de entrada da variável Di no dimensionamento do biodigestor.

Conforme apresentado na Figura 7 observa-se que:

- Para atender a heurística “estética e design minimalista” os textos e imagens foram dispostos mostrando somente o necessário, a fim de exibir somente o que interessa ao usuário em determinado instante;
- Para atender a heurística “visibilidade e status do sistema”, em que o sistema deve informar ao usuário sobre o que ele está fazendo, os títulos nos frames internos do software foram dispostos de forma que o usuário saiba onde e o que ele está fazendo no sistema;

- Atendendo ao item sobre “compatibilidade entre sistema e o mundo real”, textos claros e diretos foram produzidos a fim de não deixar o usuário confuso ao utilizar o *software*, de forma que a terminologia utilizada foi baseada na linguagem do usuário e não do sistema;
- O item “controle do usuário e liberdade”, que diz respeito à liberdade de a qualquer momento o usuário poder abortar a operação no sistema, além de poder alterar um dado inserido em uma etapa anterior, foi atendido por meio do botão “voltar”, de um botão de cancelar e outro de fechar; e
- A interface foi formatada de modo que os erros pudessem ser prevenidos por meio de diálogos que ajudam o usuário a efetuar as tarefas da maneira correta, atendendo assim a

heurística “Prevenção de erros”.

Na Figura 8 está apresentada a exemplificação dos comandos e ações sempre na mesma localização e formatação. Para atender o item “consistência e padrões”, os comandos e ações estão dispostos sempre na mesma localização e formação, mantendo um padrão e assim facilitando o reconhecimento.

Na Figura 9 está apresentado uma tela na qual necessita da entrada da variável T_r , exemplificando o que é T_r . Na heurística “reconhecimento ao invés de lembrar”, o sistema deve permitir que o usuário faça suas escolhas sem a necessidade relembrá-lo de algo específico, para isso, exemplos são mostrados em todas as telas do programa como auxílio às ações do usuário.

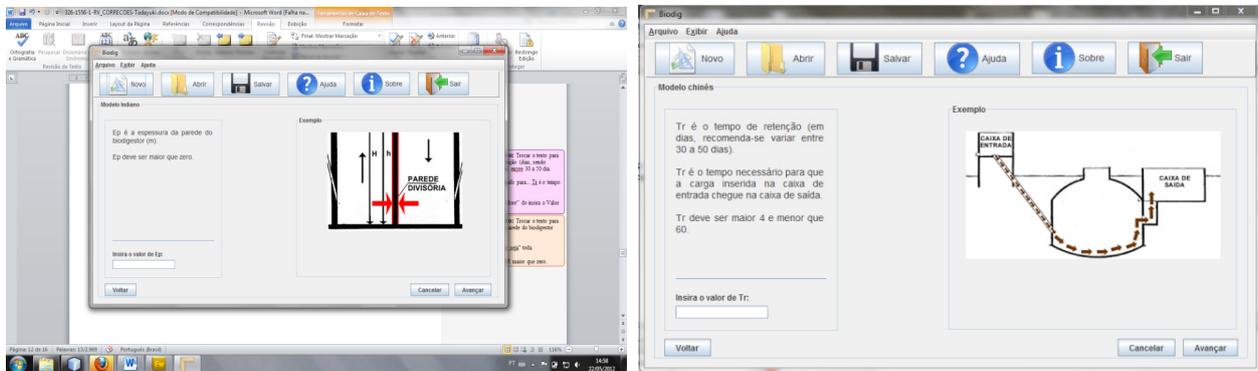


Figura 8. Exemplificação dos comandos e ações sempre na mesma localização e formatação.

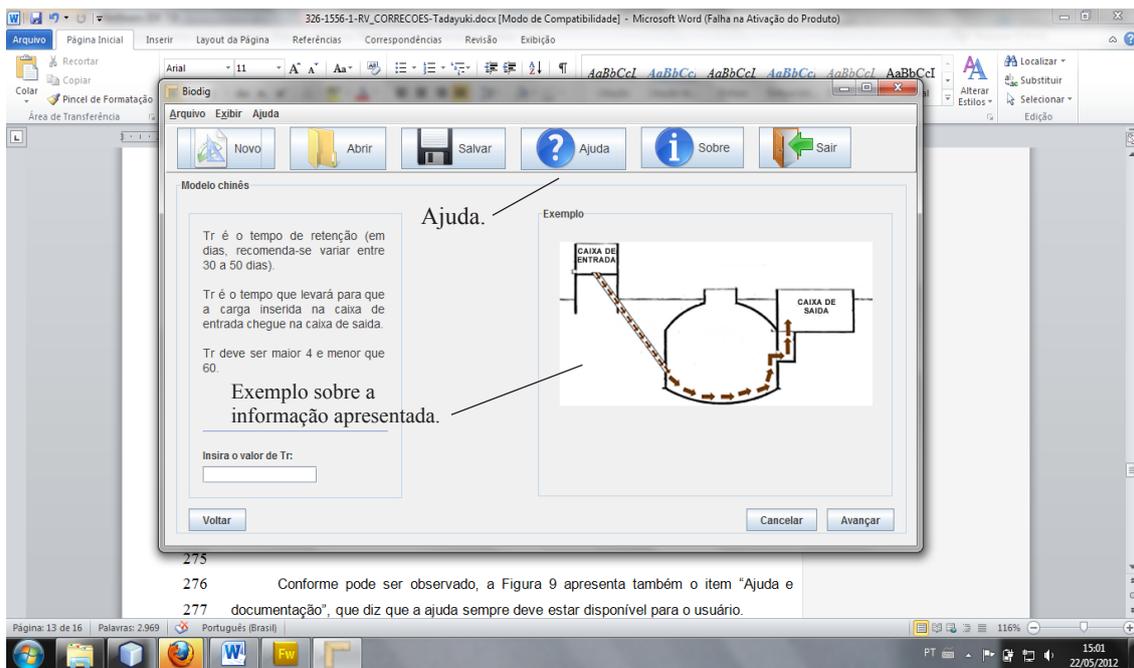


Figura 9. Tela que necessita da entrada da variável T_r , exemplificando o que é T_r .

Conforme pode ser observado, a Figura 9 apresenta também o item “Ajuda e documentação”, dizendo que a ajuda sempre deve estar disponível para o usuário.

Atalhos foram adicionados para auxiliar o usuário experiente, permitindo navegação mais ágil através do sistema, atendendo, desta forma, o item “flexibilidade e eficiência de utilização”, no qual o sistema deve fornecer ao usuário experiente uma forma ágil de operar. Os atalhos foram inseridos nos itens do *menu* arquivo, sendo que, para o acesso de “Novo”, “Abrir”, “Salvar” e “Sair” foram usadas as teclas Ctrl-N, Ctrl-O, Ctrl-S e Ctrl-F4, respectivamente. Mensagens de erro e de alerta foram adicionadas ao *software*, atendendo dessa forma o item “Ajuda ao usuário para reconhecer, identificar e recuperar erros”, no qual, mensagens de erro ou falha devem ser escritas de forma clara e não técnica, com intuito de ajudar o usuário a reconhecer o erro e conseguir recuperá-lo, sem auxílio da ajuda. Dentre as mensagens de erro e de alerta disponíveis pode-se citar: “Campo vazio”, “Valor fora do limite” e “Deve-se adotar um valor maior para D”.

CONCLUSÕES

O programa desenvolvido facilita ao usuário dimensionar biodigestores nos modelos chinês e ou indiano, minimizando os erros e o tempo gasto nos cálculos, apresentando boa usabilidade, de acordo com a análise das heurísticas de Nielsen.

AGRADECIMENTOS

Os autores agradecem ao CNPq pelo suporte financeiro.

REFERÊNCIAS BIBLIOGRÁFICAS

ALVAREZ, R.; LIDÉN, G. Semi-continuous co-digestion of solid slaughterhouse waste, manure, and fruit and vegetable waste. **Renewable Energy**, Brighton, v. 33, n. 4, p.726-734, 2008.

BENICASA, M.; ORTOLANI, A.F.; JUNIOR, J.L. **Biodigestores convencionais**. Jaboticabal: Faculdade de Ciências Agrárias e Veterinárias,

Universidade Estadual Paulista, 1991. 25p.

COLDEBELLA, A.; SOUZA, S.N.M.; SOUZA, J.; KOHELER, A.C. Viabilidade da co-geração de energia elétrica com biogás da bovinocultura de leite. **Anais... XI Encontro de Energia no Meio Rural**. Campinas, 2006.

FLORENTINO, H.O. Mathematical tool to size rural digesters. **Scientia Agricola**, Piracicaba, v. 60 n. 1, p. 185-190, 2003.

GOSLING, J.J.; STEELE, B.; BRACHA, G. **Java language specification**, Second Edition: The Java Series. Addison-Wesley Longman Publishing Co., Boston, MA, USA, ISBN:0201310082, 2000.

MACINTYRE, A. J. **Instalações hidráulicas**. 3.ed. Rio de Janeiro: LTC, 1996. 596 p.

ORACLE CORPORATION. **NETBEANS IDE 6.8**. Disponível em: <<http://www.netbeans.org/community/releases/68/>> Acessado em: 10/03/2011

NIELSEN, J. (2005) **Ten Usability Heuristics**. Disponível em: www.useit.com/papers/heuristic/heuristic_list.html Acessado em: 10/04/2011

PETERSEN, K.; WOHLIN C.; BACA, D. The waterfall model in large-scale development. **Blekinge Institute of Technology**, Karlskrona, n.32, p.386–400, 2009.

POCATILU, P. Testing java me applications. **Informatica Economica Journal**, Bucharest, v. 3, n.47, p. 147-150, 2008.

SANTOS, S.L.; TEIXEIRA, F.G. Design de uma interface de interação tridimensional com foco na usabilidade e no desempenho gráfico. **Design & Tecnologia**, Porto Alegre, v. 1, n. 1, p. 39-50, 2010.

SOARES, M. Metodologias ágeis extreme programming e scrum para o desenvolvimento de software. **Revista Eletrônica de Sistemas de Informação**, Curitiba, v.3, n.1, p.8-13, 2004.

SOMMERVILLE, I. **Software Engineering** 8ª

edição, Rio de Janeiro: Pearson Addison Wesley. 2006.552 p.

SOUZA, C.F.; CAMPOS, J.A.; SANTOS C.R.; BRESSAN, W.S.; MOGAMI, C.A. Produção de dejetos de suínos. **Ciência e Agrotecnologia**, Lavras, v. 32, n. 12, p. 19-224, 2008.

TEIXEIRA, V.H. **Biogás**.Lavras: UFLA/FAEPE, 1998. 144 p.

ZEN, E.; NUNES, R.C.; OLIVEIRA, M.A.F.; FONSECA, M.S.; PEREIRA, S.N. Melhora da

Produtividade no uso do Prontuário Eletrônico do Paciente Utilizando a Engenharia da Usabilidade. **Anais...** XXVIII Congresso da Sociedade Brasileira de Computação, Belém do Pará, WIM - Workshop de Informática Médica. Porto Alegre: SBC, 2008.

ZHAOYIN Z.; YANFANG L.; CHEN. C. Software Requirement Analysis Research Based on Event-driven. In: INTERNATIONAL FORUM ON COMPUTER SCIENCE-TECHNOLOGY AND APPLICATIONS, Chongqing, 2009. **Anais...** Chongqing: IFCSTA, v. I, 2009. p. 247-250.