

PROGRAMAÇÃO INTEGRADA APLICADA À AUTOMAÇÃO DE PROJETOS: UMA APLICAÇÃO EM EXAUSTORES PARA AVIÁRIOS

ALEXSSANDER ARIEL KUHN LIESENFELD¹, PAULO CÉSAR TONIN²

¹ Graduado em Tecnologia em Manutenção Industrial, Graduando em Ciência da Computação, Mestrando em Tecnologias Computacionais para o Agronegócio, UTFPR, Campus Medianeira – PR, Fone: (45) 9144-7742, alex.a7.93@hotmail.com.

² Doutor em Engenharia Agrícola, Professor da UTFPR – Câmpus Medianeira-PR, Fone: (45) 3240-8000, pctonin@utfpr.edu.br.

Apresentado no
XLV Congresso Brasileiro de Engenharia Agrícola - CONBEA 2016
24 a 28 de julho de 2016 - Florianópolis - SC, Brasil

RESUMO: Este trabalho apresenta uma alternativa na criação de projetos de exaustores utilizados em aviários. Para isto são utilizadas funções de programação orientada à objeto e esta programação é integrada entre os softwares de programação e de CAD 3D, fazendo com que este execute as atividades de modelagem e detalhamento necessárias. Ao final, o usuário poderá abrir o software de CAD 3D, executar a macro que foi criada para rodar o código e, assim, a interface lhe será apresentada. Nesta interface então, o usuário definirá as dimensões geométricas principais do exaustor e o projeto será criado automaticamente, gerando arquivos em formato original que podem ser abertos e editados no software de CAD 3D, além de arquivos em formato PDF para os detalhamentos.

PALAVRAS-CHAVE: Exaustor, SolidWorks[®] API, Visual Basic for Applications.

INTEGRATED PROGRAMMING FOR PROJECTS AUTOMATION: AN APPLICATION IN EXHAUSTS FOR POULTRY.

ABSTRACT: This paper presents an alternative in creating exhausts projects used in broiler houses. For this purpose, oriented-functions programming are used, and it is integrated between the programming software and 3D CAD, causing it to perform the modeling activities and its necessary detailing. Finally, the user can open the 3D CAD software, put the macro created into effect in order to run the code, and thus, the interface will be displayed. Using this interface, then, the user will define the exhaust's main geometrical dimensions and the project will be created automatically, forming files in original format to be opened and edited using the 3D CAD software, as well as PDF files for detailing.

KEYWORDS: Exhaust, SolidWorks[®] API, Visual Basic for Applications.

INTRODUÇÃO

O uso da tecnologia computacional vem aumentando a cada ano. Podemos citar diversos exemplos de utilizações dos avanços tecnológicos em várias áreas do conhecimento, como as diversas utilizações do GPS (*Global Positioning System*), utilização de sistemas

automatizados em lojas e pontos de venda, explorando a utilização de banco de dados, dentre várias outras utilidades de tecnologias computacionais.

Estes avanços podem também ser citados na área de projetos de equipamentos. O que anteriormente era feito mecanicamente por um projetista, utilizando lápis e papel, hoje é normalmente substituído pelos *softwares* computacionais de desenho, conhecidos como CAD (*Computer Aided Design*). Este desenvolvimento fez com que os projetos fossem concluídos com mais agilidade, de mais fácil detalhamento e entendimento por todas as partes envolvidas no projeto.

Os projetos e simulações em 3D tornam-se muito importantes para empresas de manufatura em geral. Na área metal-mecânica contribuem diretamente na determinação dos componentes e dos recursos a serem utilizados na fabricação e até mesmo na sequência de montagem do equipamento final (FARHAN, 2011).

Um grande problema encontrado em pequenas e médias empresas fabricantes de equipamentos para a área avícola é a falta de um projetista especializado que execute estes projetos, seja pela falta de profissional capacitado, ou pelo custo de se manter este funcionário na empresa. Oliveira (2007) relata que, em média, das empresas de pequeno e médio porte que trabalham no setor metal-mecânico no Brasil e utilizam algum tipo de desenho computacional em seus projetos, apenas 50% utiliza projetos em 3D, porém, é importante destacar que em modelagens 3D é possível verificar exatamente como as peças irão ficar depois de manufaturadas, diminuindo as chances de erro de execução e desperdícios de material (ALI et al., 2013).

Uma maneira eficiente encontrada para contornar este problema, é a automação de tarefas repetitivas nos projetos de equipamentos, ou seja, que possuem mesmo aspecto, mas com dimensões geométricas diferentes. Desta forma, o projeto é feito apenas uma vez de forma genérica e, a partir deste, o usuário tem a possibilidade de definir as dimensões que deseja para o equipamento e o *software*, utilizando destes dados, gera todo o projeto final.

O principal objetivo deste trabalho, portanto, foi desenvolver um programa que realize o projeto de um exaustor para aviários, de forma que seja apresentada ao usuário uma interface na qual ele deve introduzir as dimensões principais deste novo exaustor e o *software* modela automaticamente cada peça, faz as montagens destas peças e depois realiza os detalhamentos, entregando ao usuário como resultado final, os arquivos correspondentes à modelagem em 3D e os arquivos de detalhamento, no formato PDF, com as medidas determinadas inicialmente.

MATERIAL E MÉTODOS

Todo o desenvolvimento do trabalho foi realizado nas dependências do Campus Medianeira da Universidade Tecnológica Federal do Paraná, com exceção das fotografias, que foram retiradas em campo, em propriedade rural da cidade de Serranópolis do Iguaçu - Paraná. As imagens que foram coletadas serviram para definição do modelo inicial que teria seus procedimentos de projeto automatizados.

O *software* de CAD 3D utilizado na elaboração de todo o trabalho foi o SolidWorks® Premium 2013.

Com o modelo definido e realizada a modelagem do exaustor padrão no SolidWorks®, foram desenvolvidos os códigos responsáveis pela automação do projeto. Para tal, foi utilizada a ferramenta de ajuda da API (*Application Programming Interface*) do SolidWorks®, que é uma ferramenta utilizada para escrever um código em uma linguagem de programação dentro de outro aplicativo (FARHAN; BRIEN; RAD, 2012), de modo a encontrar os objetos e os seus respectivos métodos necessários por realizar cada uma das ações pretendidas dentro

da modelagem.

Outro recurso que é disponibilizado pelo SolidWorks® e foi utilizado, é a ferramenta de criação de macros, com a qual é possível gravar procedimentos repetitivos realizados dentro do *software* e, sempre que for necessário, rodar esta macro para que ela realize o mesmo procedimento gravado anteriormente. Com este recurso, é possível gravar as ações que são realizadas dentro da interface gráfica do SolidWorks® e editar o código que fora gerado por esta macro, assim, é possível verificar com facilidade quais métodos foram utilizados na realização de determinado recurso, e utilizar este método oportunamente dentro do código de automação do projeto.

É importante salientar que o SolidWorks® quando gera os códigos de macro, nem sempre utiliza os métodos mais recentes na construção do código, sabendo que versões mais recentes do *software* possuem alguns métodos modificados e otimizados com relação a versões anteriores, no entanto, cabe ao programador verificar a necessidade e utilizar os métodos que melhor convém. Outro importante ponto a ser destacado, é que os códigos gerados pelas macros são pouco simplificados e, portanto mais extensos, e de difícil compreensão. Sabendo disso, o trabalho foi desenvolvido diretamente no VBA, utilizando os métodos da API do SolidWorks® e valendo-se dos recursos de macro somente em casos específicos.

Cada peça, montagem e detalhamento do SolidWorks® foi tratado no código como um módulo, que é um espaço que armazena rotinas gerais e reutilizáveis dentro do programa (PERRY, 1999). Deste modo, foi possível escrever e testar cada um dos procedimentos de criação de peças, para então passar para as sub-montagens e montagens e, por fim, criar os arquivos de detalhamento, sempre testando e validando cada um destes módulos em sua etapa de criação. Foi gerado um outro módulo, considerado módulo inicial, o qual é responsável por “chamar” cada um dos outros módulos no momento em que devem ser executados, fazendo com que o projeto siga o roteiro necessário para a construção de todo o equipamento. Este último módulo, por fim, foi vinculado a um botão do formulário ao qual são inseridos os dados que o usuário deseja que sejam utilizados na modelagem do exaustor e este botão faz com que se inicie a execução de todos os códigos.

Para automatizar os procedimentos a fim de fazer com que sejam utilizados efetivamente os dados inseridos pelo usuário, foram feitas modificações nos parâmetros de medidas que foram passados aos métodos, fazendo com que, ao invés de valores fixos, estes parâmetros recebam algum valor vindo de uma variável, que por sua vez está atribuída ao valor que o usuário determinar. Com isso, todas as medidas dependentes desta primeira medida alterada, devem se modificar de maneira proporcional à modificação feita pelo usuário, a fim de fazer com que o projeto continue funcionando corretamente. Estas modificações foram realizadas através de cálculos e testes, utilizando estas variáveis como base para a definição de todos os outros parâmetros dos módulos que, de alguma forma, estavam vinculados àquelas medidas.

Por fim, foram criadas as interfaces de interação com o usuário através do VBA. A primeira trata-se de uma apresentação do *software*, Figura 1, e a segunda é o formulário propriamente dito, no qual os dados serão inseridos, Figura 2.

Estas interfaces foram vinculadas às respectivas variáveis que posteriormente são utilizadas no código. Também foi adicionada uma referência à biblioteca do Microsoft Office Excel® 2013, para que fosse possível adicionar um botão no formulário que possibilita que o usuário utilize o explorador de pastas e arquivos do Microsoft Windows® para determinar o local de salvamento de seus arquivos e adicionar um formato de folha padrão para os detalhamentos, porém, estas informações de diretório podem ser passadas diretamente no campo especificado, de modo que não se faz necessário que o usuário possua o pacote Microsoft Office® citado, sendo necessário somente possuir instalado o SolidWorks® 2013.



Figura 1 - Interface de apresentação.
Fonte: Autoria própria.

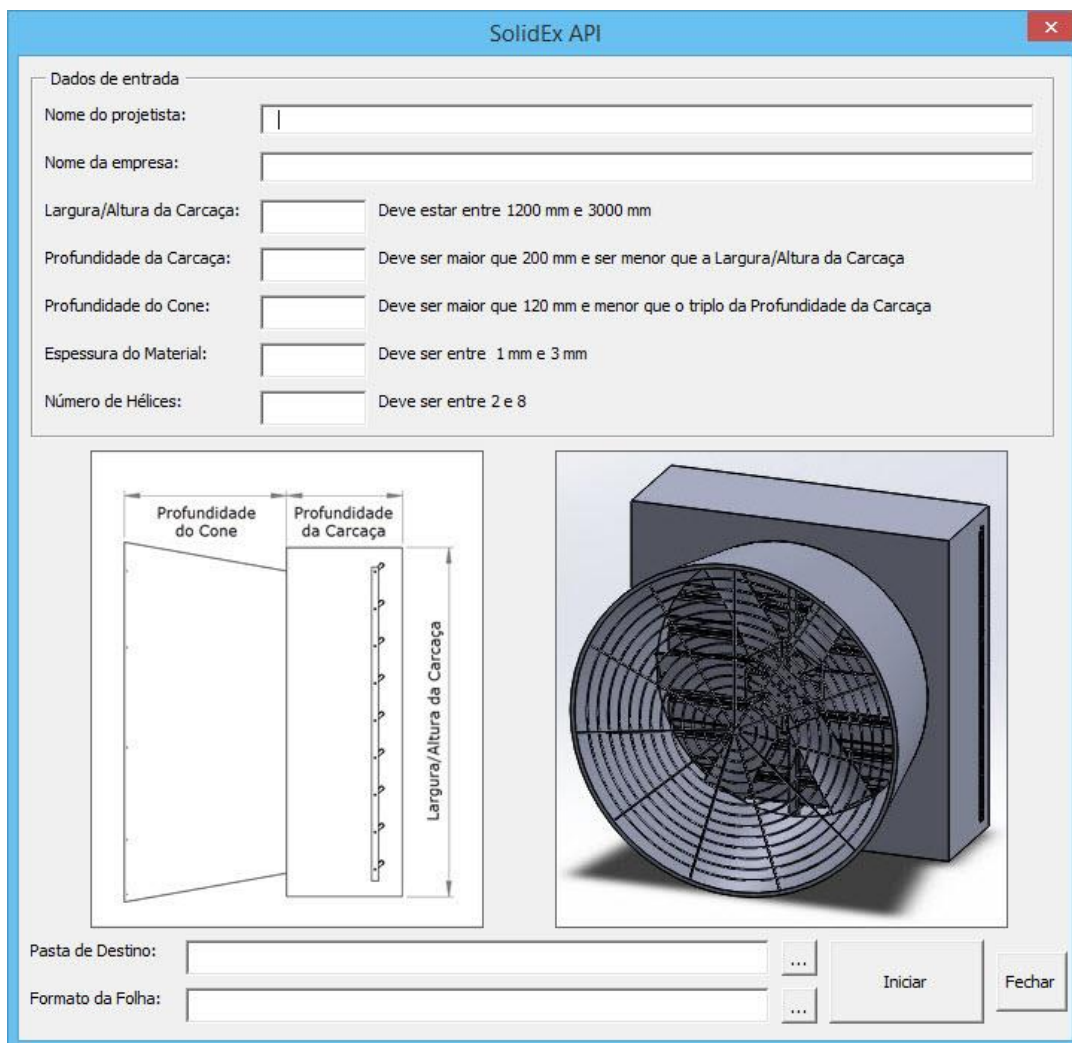


Figura 2 - Interface para definição de dados.
Fonte: Autoria própria.

RESULTADOS E DISCUSSÃO

A Figura 3 mostra a janela de propriedades do projeto, em que são listados todos os módulos que criam as peças do exaustor e os dois formulários de interação com o usuário. Dentre os módulos, as letras iniciais do nome de cada um deles identificam o tipo de modelagem que este código realiza no *Software* de CAD 3D. Letra “p” para peças, “m” para montagens e “d” para detalhamentos. A letra “a” foi utilizada no código inicial por conveniência para que seja a primeira da lista.



Figura 3 - Explorador de Projetos do VBA.
Fonte: Autoria própria.

A seguir, na Figura 4, é demonstrado um exemplo de uma parte do código escrito para realização da modelagem de uma das peças que, neste caso, é do cone do exaustor.

Desta mesma forma, todas as automações das criações das outras peças, montagens e detalhamentos foram codificadas, seguindo cada uma delas suas particularidades de execução,

seja na definição dos objetos, classes e métodos a serem utilizados em cada execução, ou das variáveis que devem ser inseridas com intuito de permitir a automação na modificação das medidas pelo usuário.

```
Sub Cone()  
  
Set swApp = Application.SldWorks  
swApp.NewPart  
Set swModel = swApp.ActiveDoc  
Set swSketchMgr = swModel.SketchManager  
Set swFeatureMgr = swModel.FeatureManager  
  
swModel.Extension.SelectByID2 strPlanoFrontal, "PLANE", 0, 0, 0, False, 0, Nothing, 0  
swModel.SketchManager.InsertSketch True  
swSketchMgr.CreateCircleByRadius 0, 0, 0, (dblCompCarcaca - 0.1)  
swSketchMgr.InsertSketch True  
  
swModel.Extension.SelectByID2 strPlanoFrontal, "PLANE", 0, 0, 0, False, Empty, Nothing, Empty  
swFeatureMgr.InsertRefPlane 8, dblCompCone, 0, 0, 0, 0  
  
...  
End Sub
```

Figura 4 - Exemplo de código escrito para execução de uma peça no VBA.

Fonte: Autoria própria.

As execuções dos códigos, então, irão gerar os componentes do exaustor e seus detalhamentos. A Figura 5 exibe a peça gerada pela execução do código mostrado anteriormente, enquanto a Figura 6 e a Figura 7 mostram a montagem final do exaustor e o detalhamento da carcaça do exaustor. Todos gerados automaticamente no momento da execução do código.

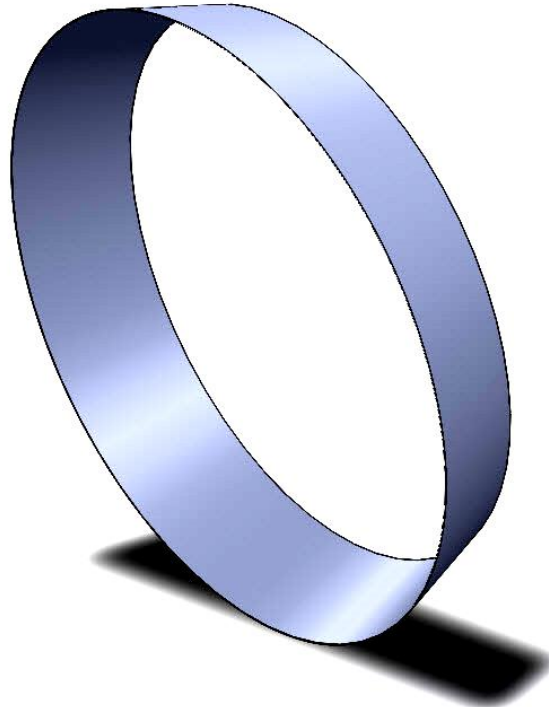


Figura 5 - Peça gerada pelo código.

Fonte: Autoria própria.

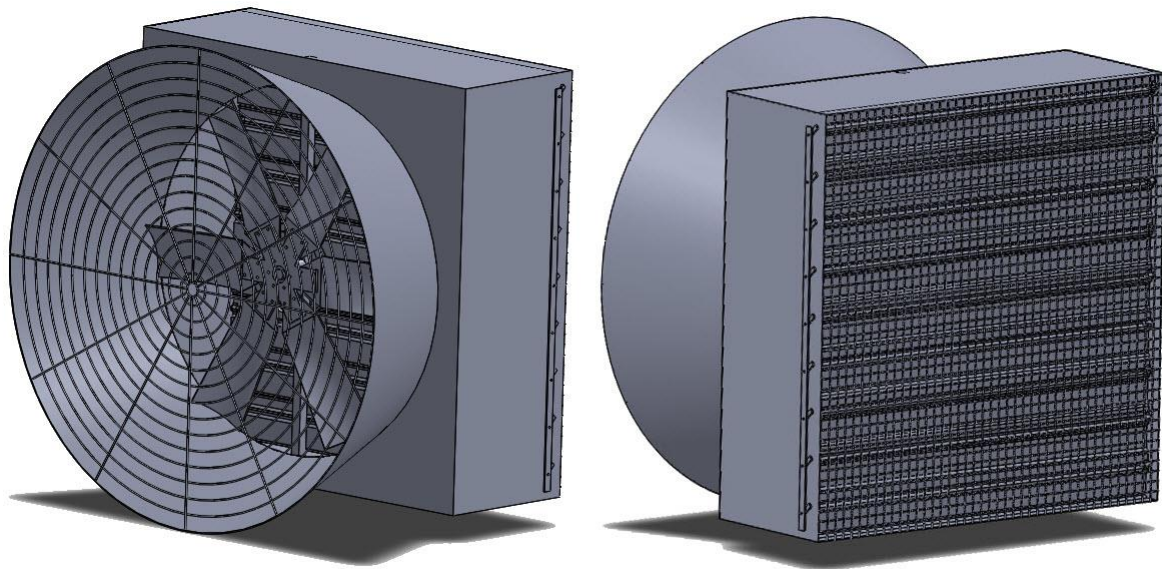


Figura 6 - Montagem gerada pelo código.
Fonte: A autoria própria.

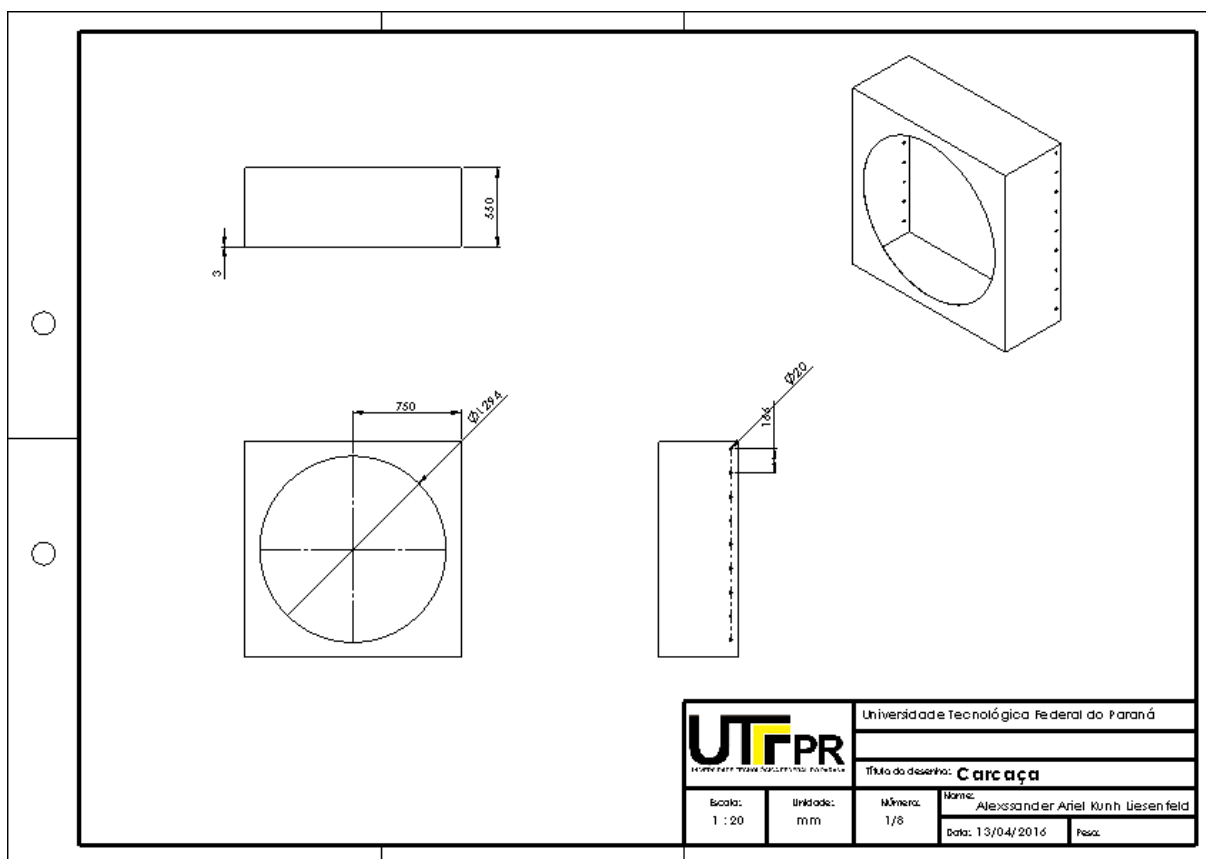


Figura 7 - Detalhamento gerado pelo código.
Fonte: A autoria própria.

Ao final da execução do projeto, o usuário possuirá, no local definido por ele para ser a pasta de destino, todos os arquivos gerados pelo *software* de CAD 3D para que sejam abertos no próprio *software*, além dos detalhamentos em formato PDF, de modo que, para visualizar os arquivos de detalhamentos das peças e montagens, não seja necessário que o computador possua o SolidWorks® instalado.

Um exemplo de como os arquivos serão gravados e disponibilizados no diretório definido pelo usuário, depois da execução, é mostrado na Figura 8.

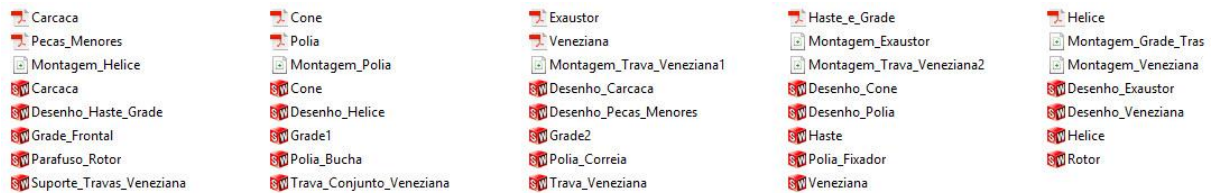


Figura 8 - Arquivos gerados ao final da execução.

Fonte: Autoria própria.

CONCLUSÕES

A pesquisa se apresenta como um trabalho eficiente na tarefa de automatizar procedimentos de criação de projetos. Como já fora citado, este trabalho é uma aplicação, o que denota o fato de ser possível realizar diversos trabalhos pela mesma linha de pesquisa, nas mais diversas áreas do projeto em 3D.

A execução de todo o projeto de um exaustor, que tem uma modelagem que necessita de moderado tempo para conclusão, pode ser realizado em poucos minutos, tornando essa velocidade de execução um grande benefício às empresas (GAO; SANG, 2008). Este tempo de execução pode ter variações dependendo do hardware em que o programa é executado. O objetivo traçado no início do projeto foi plenamente atendido. O software, dadas as limitações de dimensionamento, é capaz de executar todo o projeto de um exaustor, utilizando as medidas fornecidas pelo usuário, entregando a este, como resultado, todos os arquivos necessários para execução do projeto.

REFERÊNCIAS

ALI, Hazrat, KATSUKI, Akio, KUROKAWA, Syuhei, SAJIMA, Takao. Design of mechanical actuator in 3D CAD software. **Procedia Engineering**, v. 64, p. 473–478, 2013. Elsevier B.V. Disponível em: <<http://dx.doi.org/10.1016/j.proeng.2013.09.121>>. Acesso em: 10 set. 2015.

FARHAN, Uday H.; RAD, Majid T. Design of modular fixtures using a 3D-modelling approach. **19th International Congress on Modelling and Simulation**, Perth, Australia, 12-16 Dez. 2011.

FARHAN, Uday H.; O'BRIEN, Simona; RAD, Majid T. SolidWorks Secondary Development with Visual Basic 6 for an Automated Modular Fixture Assembly Approach. **International Journal of Engineering**, v. 6, p. 290–304, 2012.

GAO, Peng; SANG, Zhi-fu. Three Dimensional Parameterized Design of Heat Exchanger Tubesheet Based on SolidWorks. **Computer-Aided Industrial Design and Conceptual Design**. p. 488–492, 22-25 nov. 2008.

OLIVEIRA, Antonio C. de. **Diretrizes de apoio ao esforço de inovação tecnológica no desenvolvimento de produtos em pequenas e médias empresas industriais**. 2007. 218 f. Tese (Doutorado em Engenharia) – Escola Politécnica da Universidade de São Paulo, Universidade de São Paulo, São Paulo, 2008.

PERRY, Greg M. **Aprenda em 21 dias Visual Basic 6**. Gulf Professional Publishing, 1999. 844 p.

SOLIDWORKS 2013. Edição Premium x64, Versão SP0.0. Delaware, Estados Unidos da América: Dassault Systèmes SolidWorks Corporation. 1 DVD-ROM.