

## Programação integrada aplicada à automação de projetos: Uma aplicação em exaustores para aviários

### 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.

**Alexssander Ariel Kuhn Liesenfeld**

[alex.a7.93@hotmail.com](mailto:alex.a7.93@hotmail.com)

Universidade Tecnológica Federal do Paraná (UTFPR), Medianeira, Paraná, Brasil.

**Paulo César Tonin**

[pctonin@utfpr.edu.br](mailto:pctonin@utfpr.edu.br)

Universidade Tecnológica Federal do Paraná (UTFPR), Medianeira, Paraná, Brasil.

## INTRODUÇÃO

O uso da tecnologia computacional vem aumentando substancialmente e estes avanços também ocorrem 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 semelhantes, 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, cria-se um programa que realiza a automação deste procedimento de criação do projeto, de modo que o usuário tenha a possibilidade de definir as dimensões que deseja para o equipamento, e o software, utilizando destes dados, gere todo o projeto final.

O principal objetivo deste trabalho, portanto, foi desenvolver um programa que realize automaticamente 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.

Com o SolidWorks® é possível realizar tarefas repetitivas automaticamente através da gravação de macros. Uma gravação de macro, quando ativada, salva todos os procedimentos que o usuário realiza, permitindo que esta macro seja reproduzida posteriormente e repita os mesmos passos que foram gravados (CHOI; MUN; HAN, 2009). Sendo assim, o usuário pode gravar a modelagem da peça que deseja e, depois, refazer esta mesma peça sempre que quiser utilizando a macro que foi gerada.

Spens (2015) salienta que quando uma macro é gravada, automaticamente um código de programação é gerado, o qual pode ser utilizado para realizar muitas tarefas e alterações, sendo que a maior parte das ações realizadas pelo usuário durante a gravação da macro são gravadas neste código de programação. O SolidWorks® também dá a possibilidade de se trabalhar utilizando sua API (Application Programming Interface), a qual Titus e Bin (2013) definem como sendo uma interface que permite que algum software interaja com outro utilizando uma linguagem de programação. Assim, modelos criados no SolidWorks® podem ser manipulados diretamente através de outros programas, como é o caso do Visual Basic for Applications®, que é efetivamente a plataforma utilizada para editar e escrever novas macros. Um fato interessante a ser observado é que o Visual Basic for Applications® possui ferramentas de criação de interfaces gráficas. Estas interfaces podem ser ajustadas da maneira que o programador pensar ser conveniente, adicionando botões, caixas de mensagem, caixas de seleção entre outros recursos e, com isso, desenvolver uma janela na qual seja simples a interação do usuário com o programa, facilitando que o usuário não tenha necessidade de conhecer a fundo as funcionalidades do *software*, necessitando apenas que ele informe os dados referentes às medidas geométricas que deseja que o projeto final possua.

Yong et al. (2009) salientam que o SolidWorks® permite que o usuário utilize diversas linguagens de programação para trabalhar com sua API como, por exemplo, a linguagem C++, C# e Visual Basic 6. O Visual Basic for Applications® dá suporte OLE (Object Linking and Embedding), o qual pode ser definido como a possibilidade de criar documentos compostos que contenham informações de fontes diferentes e possam ser processados em diferentes aplicativos (MICROSOFT, 2015). No presente trabalho a linguagem utilizada foi a Visual Basic 6 através do Visual Basic for Applications® e, então, esta programação foi integrada com o SolidWorks®, o qual foi responsável pela criação de todo o projeto, utilizando das medidas fornecidas pelo usuário.

## 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 Iguazu - Paraná. Com o modelo já definido e as medidas e imagens dos exaustores já adquiridas é possível realizar as modelagens dos componentes. Para isto, foi utilizado o software *SolidWorks®*, sendo modeladas primeiramente todas as peças que irão compor o exaustor por completo, para então ser feita a montagem de todas as peças formando o equipamento. O último passo da elaboração do projeto é, então, fazer o detalhamento das peças individuais e das montagens. Estes procedimentos são muito relevantes por darem uma real perspectiva de como ficará o equipamento depois de fabricado e montado. Em particular, são de grande importância para a indústria utilizar no momento da manufatura das peças, uma vez que ela mostra todas as medidas, agilizando a fabricação e evitando erros de execução.

Com o modelo definido e realizada a modelagem do exaustor padrão, foram desenvolvidos os códigos responsáveis pela automação do projeto. Para tal, além dos recursos de macro já citados, foi utilizada a ferramenta de ajuda da API

(Application Programming Interface) do SolidWorks, de modo a encontrar os objetos e os seus respectivos métodos necessários por realizar cada uma das ações pretendidas dentro da modelagem.

Com o recurso de gravação e edição de macros foi 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. Isto faz com que seja muito importante a edição destas macros. Como vários recursos iguais são utilizados em diferentes partes do projeto, foi possível reutilizar trechos do código, fazendo com que fique mais compreensível para o programador que for ler o código, além de economizar um tempo que seria gasto gravando e editando macros que podem ser escritas manualmente.

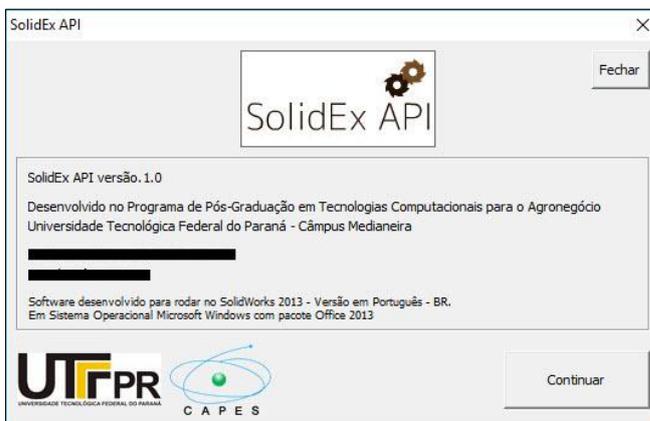
Cada peça, montagem e detalhamento do SolidWorks® foi tratado no código como um módulo, que é um espaço que armazena rotinas gerais e reutilizáveis dentro do programa (PERRY, 1999). Deste modo, foi possível escrever e testar cada um dos procedimentos de criação de peças, para então passar para as sub-montagens 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. A 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 tem atribuída a ela o 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. Para isso, utilizaram-se variáveis auxiliares a fim de que todas as medidas que sofressem alguma interferência das que foram informadas pelo usuário fossem modificadas e se adequassem ao projeto.

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. Cada campo de texto foi vinculado à sua respectiva variável dentro do código, a qual é alterada conforme o valor inserido pelo usuário. O usuário deve informar

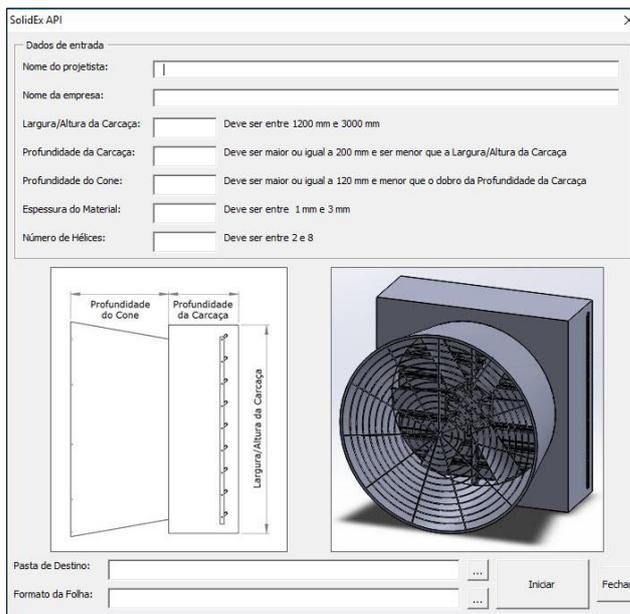
ao software a largura/ altura da carcaça, profundidade da carcaça, profundidade do cone, espessura do material e número de hélices, respeitando os limites estipulados pelo programa. Foram incluídos, também, campos específicos para determinar nome de projetista e empresa, que são informações opcionais, além de botões com função de abrir o explorador de arquivos que solicita ao usuário determinar um local específico para salvamento dos arquivos e outro para que determine um arquivo de formato de folha do SolidWorks, sendo este último também opcional.

Figura 1 - Interface de apresentação do software.



Fonte: Autoria Própria, (2010).

Figura 2 - Interface para inserção de dados.



Fonte: Autoria Própria, (2010).

## RESULTADOS E DISCUSSÃO

Ao final do trabalho, todos os códigos que realizam a automação da criação das peças, montagens e detalhamentos estarão dispostos em seus respectivos módulos.

A seguir, na Figura 3, 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. A peça gerada pela execução deste código é exibida na Figura 4.

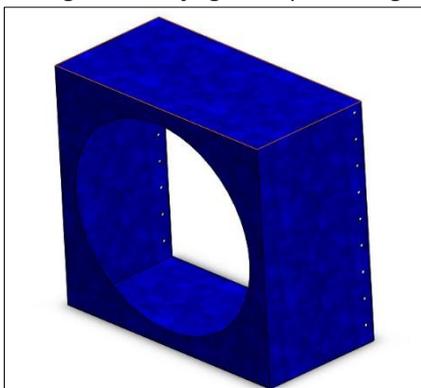
Desta mesma forma, todas as automações das criações das outras peças, montagens e detalhamentos foram codificadas, seguindo cada uma delas suas particularidades de execução, seja na definição dos objetos, classes e métodos a serem utilizados em cada execução, ou das variáveis que devem ser inseridas com intuito de permitir a automação na modificação das medidas pelo usuário.

Figura 3 - Código escrito para execução de uma peça no VBA.

```
Sub Carcaca ()  
  
Set swApp = Application.SldWorks  
swApp.NewPart  
Set swModel = swApp.ActiveDoc  
Set swPart = swModel  
Set swSketchMgr = swModel.SketchManager  
Set swFeatureMgr = swModel.FeatureManager  
  
'Definir os nomes dos planos em String  
Set swFeatPlano = swModel.FirstFeature  
Do While Not swFeatPlano Is Nothing  
    If swFeatPlano.GetTypeName = "RefPlane" Then Exit Do  
    Set swFeatPlano = swFeatPlano.GetNextFeature  
    strPlanoFrontal = swFeatPlano.Name  
Loop  
...
```

Fonte: Autoria própria, (2010).

Figura 4 - Peça gerada pelo código.



Fonte: Autoria própria, (2010).



## 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.

# Integrated programming for automation projects: 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.

---

## 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.

CHOI, Guk-Heon; MUN, Duhwan; HAN, Soonhung. Exchange of CAD part models based on the macro-parametric approach. **International Journal of CAD/CAM**, v. 2, n. 1, p. 13–21, 2009. Disponível em:  
<<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.6566&rep=rep1&type=pdf>\n<http://www.ijcc.org/ojs/index.php/ijcc/article/view/8>>. Acesso em: 09 set. 2015.

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

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.

MICROSOFT. **Visão geral sobre requisitos e conceitos OLE**. Disponível em: <<https://support.microsoft.com/pt-br/kb/86008>>. Acesso em 12 nov. 2015.

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.

SPENS, Mike. **Automating SOLIDWORKS 2015 Using Macros**, SDC Publications, 2015. 384 p.

TITUS, Avitus; BIN, Liu X. Secondary Development of Solid works for Standard Components Based on Database. **International Journal of Science and Research**, v. 2, n. 10, p. 162–164, 2013.

---

YONG, Wu G.; XING, Tang G.; LIANG, Zhang H.; SHAN, Yao. Visual Modeling and Simulation for Laser Action Area in Rapid Prototyping. **2009 WRI World Congress on Computer Science and Information Engineering**, p. 435–439, 2009. Disponível em:  
<<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5170573>>. Acesso em: 12 nov. 2015.

**Recebido:** 04 ago. 2016.

**Aprovado:** 23 nov. 2016.

**DOI:**

**Como citar:** LIESENFELD, A. A. K.; TONIN, P. C. Programação integrada aplicada à automação de projetos: Uma aplicação em exaustores para aviários. R. Eletr. Cient. Inov. Tecnol., Medianeira, v. 2, n. 14, p. 139-149, jul./dez. 2016. Disponível em: <<https://periodicos.utfpr.edu.br/recit>>. Acesso em: XXX.

**Correspondência:**

Alexssander Ariel Kuhn Liesenfeld

Rua Ivo Antônio Coradini, 115, Jardimópolis, CEP 85885-000, Serranópolis do Iguaçu, Paraná, Brasil.

**Direito autoral:** Este artigo está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.

