

DESENVOLVIMENTO DE UMA FERRAMENTA DE MODELAGEM PARAMÉTRICA PARA PROJETOS EM SISTEMA CONSTRUTIVO ESTRUTURADO EM PERFIS METÁLICOS

Flávia de Souza Bastos^a, Michèle Cristina Resende Farage^a, José Humberto Góes Alves^b,
Rafaela Juliany Mendes^c and Natalia da Silva Rossi de Resende^c

^a*Departamento de Mecânica Aplicada e Computacional, Universidade Federal de Juiz de Fora, Minas Gerais, Brasil, flavia.bastos@ufjf.edu.br, michele.farage@ufjf.edu.br <http://www.ufjf.br/mac>*

^b*Programa de Pós-Graduação em Modelagem Computacional, Universidade Federal de Juiz de Fora, Minas Gerais, Brasil, josegoes@outlook.com, <http://www.ufjf.br/ppmc>*

^c*Programa de Educação Tutorial - Engenharia Civil, Universidade Federal de Juiz de Fora, Minas Gerais, Brasil, rafaela.juliany@engenharia.ufjf.br, natalia.resende@engenharia.ufjf.br <http://www.ufjf.br/petcivil>*

Keywords: Modelagem Paramétrica, BIM, Light Steel Framing

Abstract. O Sistema Construtivo Estruturado em Perfis Metálicos, ou Light Steel Framing (LSF), também designado como sistema autoportante de construção a seco em aço, é composto por um esqueleto estrutural de perfis formados a frio, em aço galvanizado, ligados entre si. Os vários elementos ou subsistemas leves que compõem a estrutura trabalham em conjunto, resistindo aos carregamentos solicitantes existentes na edificação. Por se tratar de um sistema que permite uma construção rápida, precisa, realizada a seco e sem desperdícios, o LSF pode ser uma opção viável para a construção de habitações populares frente ao enorme déficit habitacional brasileiro, estimado em 5,4 milhões em 2011, de acordo com o estudo do Instituto de Pesquisa Econômica Aplicada - IPEA. O LSF apresenta uma coordenação modular que permite o desenvolvimento de recursos tecnológicos baseados em Modelagem da Informação da Construção, ou *Building Information Modeling* (BIM). BIM não é um *software*, mas uma tecnologia que permite planejar, projetar, construir, usar e manter uma construção. Para isso, não é necessário que exista um aplicativo que englobe todas essas tarefas. Por utilizar informação parametrizada, aplicativos distintos podem compartilhar uma mesma informação e gerar seus resultados, processo esse denominado de interoperabilidade. Um sistema BIM fornece classes de objetos que pré-definem, por exemplo, o que é uma parede, uma laje, um telhado, etc, em termos de como eles interagem com outros objetos. Há uma enorme demanda para o desenvolvimento de novos objetos paramétricos e extensões de bibliotecas existentes, criando recursos personalizados para os usuários que desejam estabelecer suas próprias práticas, interagindo com análises, estimativas de custos e outras aplicações. A ideia de uma plataforma comum que permita gerir e transferir os dados entre diferentes aplicações de forma unificada afigura-se como adequada, mas ainda está longe de ser concretizada. Propõe-se neste trabalho o desenvolvimento de uma ferramenta BIM para a criação de pré-projetos em LSF, associada à análise estrutural. Prevê-se a utilização de programas de Modelagem Geométrica 3D, que possuam Interface de Programação de Aplicativos.

1 INTRODUÇÃO

O desenvolvimento de sistemas computacionais que ajudem os engenheiros a aumentarem a eficiência do seu trabalho, melhorando o nível de automação de tarefas repetitivas e não criativas, é área de interesse da Engenharia do Conhecimento (Rocca, 2012), ou *Knowledge Based Engineering* (KBE), em inglês. Esses sistemas visam capturar e sistematicamente reutilizar produtos e conhecimentos de engenharia de processos, com o objetivo final de reduzir o tempo e custos de desenvolvimento de produtos.

A indústria da Construção Civil, quando comparada às indústrias automotiva, aeroespacial e aos demais setores industriais que alcançaram aumento de produtividade e globalização, tem sido resistente a mudanças na concepção e no gerenciamento dos projetos. Segundo Santiago and Araújo (2008), predominam, no cenário brasileiro da construção civil, processos essencialmente artesanais em que são marcantes a baixa produtividade e o enorme desperdício.

De forma a reduzir tal atraso tecnológico (Schwegler et al., 2001), tem se promovido a implantação da tecnologia de Modelagem da Informação da Construção, em inglês *Building Information Modeling* (BIM), alinhada à Engenharia do Conhecimento. A tecnologia BIM pretende modificar substancialmente a forma de se elaborar um projeto de engenharia, bem como as relações e interações entre os diversos participantes do processo construtivo (Mattei, 2008).

Embora o conceito esteja em desenvolvimento desde os anos 70, até hoje não se pode dizer que essa tecnologia esteja consolidada no mercado. Isso se deve a diversos fatores intrínsecos do setor, como, por exemplo, a grande quantidade de profissionais envolvidos. Deve-se ressaltar que os *softwares* empregados diferem-se dos seus precursores, os programas CAD (*Computer Aided Design*). Durante a adaptação ao novo processo, as empresas necessitam investir economicamente na capacitação dos profissionais e na aquisição dos programas e de computadores com alta capacidade de processamento. Além dessa, a aquisição e codificação do conhecimento das empresas ou das necessidades do cliente, através do uso de uma linguagem de programação orientada a objetos, é outra barreira no desenvolvimento de sistemas (Rocca, 2012).

Apesar das contrariedades, é incontestável a importância que a tecnologia BIM assume em termos de desenvolvimento colaborativo dos projetos e de parametrização dos modelos. De acordo com Tse et al. (2005) sua função excede a de visualização do espaço projetado. O modelo digital criado é composto por um banco de dados que permite agregar informações para diversas finalidades. Conforme conclui Mattei (2008), o BIM consiste em um sistema inteligente, onde cada aspecto do modelo está ligado a todos os outros aspectos num só objetivo: refletir a realidade. Uma mudança que seja, em qualquer característica do projeto, seja ela de caráter gráfico ou textual, é imediatamente refletida nas outras características. Além disso, trabalhe-se com o conceito de integridade paramétrica: a geometria pode ser alterada modificando-se dimensões ou outros parâmetros.

A promoção de mudanças na indústria da construção civil exige, em concomitância à aplicação de alternativas tecnológicas, uma modificação no âmbito executivo, buscando-se incentivar métodos construtivos mais eficazes e sustentáveis. Um sistema que atende a essas demandas é o Sistema Construtivo Estruturado em Perfis Metálicos, ou *Light Steel Frame* (LSF), em que as edificações são estruturadas por perfis leves de aço galvanizado. Algumas características próprias desse sistema, tais como modularização e uso de elementos pré-fabricados, fazem com que ele seja compatível com o planejamento em plataformas BIM.

O estado da arte em plataformas BIM especificamente para uso em projetos de LSF inclui o FrameCad System® , SteelSmart System® e Metal Wood Framer®, sendo que o último é, na verdade, uma ferramenta desenvolvida para uso integrado no Revit® . É também possível

projetar LSF no Revit® através do uso de *templates*, que são conjuntos de dados previamente gravados e que servem de base à inserção de outros dados.

Os programas supracitados são todos de natureza comercial, o que torna difícil aprimorar os conhecimentos em desenvolvimento de novos objetos paramétricos e extensões de bibliotecas existentes. Já o SketchUp®, um *software* de modelagem tridimensional bastante intuitivo, através da sua versão gratuita, *Make*, apresenta-se como uma alternativa, graças à sua Interface de Programação de Aplicativos (ou API) em *Ruby*. O *Ruby* é uma linguagem de programação flexível e totalmente livre de custos tanto para uso, quanto cópia, modificação e distribuição.

De acordo com Melo (2012), a construção de um *plug-in* para o SketchUp® é uma solução que concilia desempenho técnico e baixo custo, já que acrescenta as funcionalidades necessárias, preservando as ferramentas pré-existentes.

O presente trabalho busca apresentar uma ferramenta computacional em desenvolvimento, cujo propósito consiste em atender à necessidade de automatizar o processo de elaboração de projetos e, dessa forma, contribuir para a expansão do emprego do sistema LSF. Espera-se que o recurso desenvolvido possa futuramente integrar plataformas BIM, reforçando o objetivo de promover a acessibilidade do sistema em questão. Nesta etapa, em que o aplicativo ainda encontra-se em construção, utilizaram-se como testes plantas-baixas de habitações unifamiliares, mais propícias ao estudo por apresentarem constituição estrutural simplificada, além de representarem um setor estratégico para implantação do LSF no Brasil.

2 SISTEMA CONSTRUTIVO ESTRUTURADO EM PERFIS METÁLICOS

O *Light Steel Framing* (LSF), também conhecido como Estrutura de Aço Leve, é um sistema construtivo estruturado a partir de perfis de aço galvanizado formados a frio. Devido a características específicas do sistema, também recebe o nome de Sistema Autoportante de Construção a Seco. O sistema completo é constituído por subsistemas interdependentes, tais como o estrutural, os de fundação, de isolamento termo-acústico, de fechamento interno e externo, vertical e horizontal, e de instalações elétricas e hidráulicas (Santiago and Araújo, 2008).

2.1 O cenário no Brasil

Sistemas construtivos pré-fabricados, como o LSF, representam uma importante alternativa aos sistemas tradicionais, uma vez que o detalhamento adequado dos projetos contribui para a redução de manifestações patológicas dos produtos e combate o desperdício de material e a lentidão no processo de produção (Vivan, 2011).

Apesar do Brasil ser um dos maiores produtores mundiais de aço (Tavares, 2011), o emprego de estruturas metálicas em edificações tem sido pouco expressivo no mercado da construção civil (Crasto, 2005). No ano de 2013, foram produzidos 34,2 milhões de toneladas de aço bruto e 33,0 milhões de toneladas de produtos siderúrgicos, segundo dados do Instituto Aço Brasil (IABr). Enquanto que nos países desenvolvidos o percentual da produção destinada ao setor da construção civil costuma ser acima de 50%, no Brasil é em torno de apenas 35,4%, incluindo-se nessas estatísticas o aço empregado para produção do concreto armado.

Uma das particularidades do mercado brasileiro que impede a expansão do uso de sistemas construtivos em aço é a ausência de mão de obra especializada. Outro fator que contribui para a baixa utilização do sistema é o fato de que as estruturas produzidas em LSF exigem projetos detalhados e especificações de materiais bastante rígidas, por isso, são incompatíveis com improvisações no canteiro de obra. Na racionalização construtiva, em que se visa minimizar as perdas e os prazos na construção, o dimensionamento, a otimização do uso dos materiais e a

compatibilização entre projetos e subsistemas passam a ser essenciais (Craсто, 2005).

2.2 Características técnicas do painel estrutural

As paredes do LSF são compostas por estruturas denominadas painéis, que, por sua vez, são formados por diversos elementos individuais (perfis), os quais interligam-se de forma a constituir o esqueleto estrutural.

A seções transversais mais comuns dos perfis são as do tipo “U” enrijecido (Ue) para montantes e vigas, do tipo “U”, utilizado como guia na base e no topo dos painéis, do tipo “Cartola” (Cr), empregado em ripas, e as cantoneiras “L”.

Segundo Rego (2002), os dois elementos principais que constituem as paredes são os montantes (normalmente perfis “Ue” dispostos verticalmente entre as guias inferior e superior) e as guias (normalmente perfis “U” que unem os montantes nos seus extremos). Quando ocorrem aberturas para portas e janelas em um painel portante, há necessidade de acréscimo de elementos estruturais como vergas, a fim de redistribuir o carregamento dos montantes interrompidos aos montantes que delimitam lateralmente o vão, denominados de ombreiras. Esses elementos estruturais estão representados na Fig. 1.

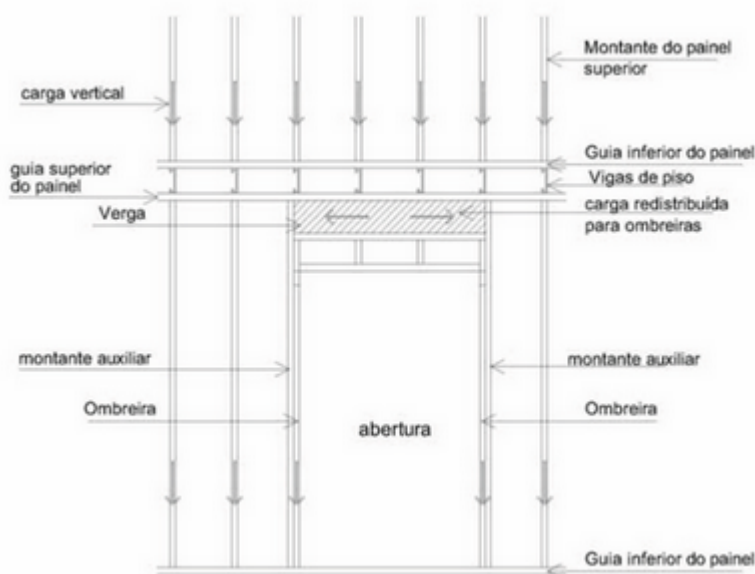


Figure 1: Representação dos elementos constituintes de um painel com abertura e distribuição dos esforços (Freitas and Craсто, 2006).

A distância entre os montantes geralmente varia entre 400 a 600mm, de acordo com a solicitação a que cada perfil será submetido. Há casos em que essa modulação pode chegar a 200mm, como por exemplo painéis que suportam carga da caixa d’água.

A verga pode ter diversas combinações, mas basicamente é composta de dois perfis Ue conectados por meio de uma peça, que seria um pedaço de perfil U ou uma chapa de conexão aparafusada em cada extremidade. Essa peça deve ter uma altura igual à altura da verga (ou seja, a alma dos perfis Ue) menos o tamanho da aba da guia superior do painel. Abaixo da verga existe uma peça chamada guia de verga, que consiste em um perfil U fixado às mesas inferiores dos dois perfis Ue e também conectada à ombreira, permitindo a fixação dos montantes de composição (Fig. 2).

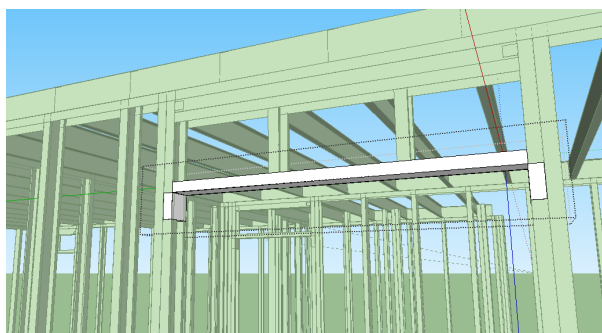


Figure 2: Combinação de perfis para montagem de verga.

As ombreiras que apoiam a verga vão desde a guia inferior do painel até a guia da verga. A quantidade de ombreiras necessárias para o apoio é definida pelo cálculo estrutural. Em uma aproximação, pode se estabelecer que o número de ombreiras a cada lado da abertura será igual ao número de montantes interrompidos pela verga dividido por 2, quando o resultado for um número ímpar deverá somar-se 1. O acabamento superior ou inferior de uma abertura é feito por um perfil U cortado no comprimento de 20 cm maior que o vão. É dado um corte na mesa de 10 cm de cada extremidade. Esse segmento é dobrado em 90° para servir de conexão com a ombreira. Essa peça é chamada de guia de abertura.

No encontro entre painéis são utilizadas várias soluções construtivas de acordo com o número de painéis que se unem e do ângulo entre eles. Podem ser utilizadas peças pré-montadas para esses encontros, mas a união dos painéis se dá basicamente por montantes conectados entre si por meio de parafusos estruturais, também conhecidos como parafusos sextavados. As principais configurações no encontro dos painéis são:

- a) Ligação de dois painéis de canto, que pode ser visto na Fig. 3.

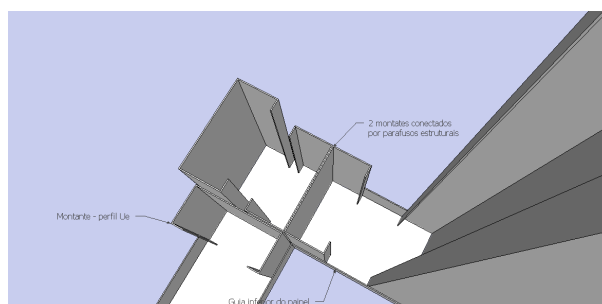


Figure 3: Combinação de perfis para montagem de verga.

- b) Ligação de dois painéis formando um “T”. Quando há ligação de painéis com uma união “T”, o painel 1, que é contínuo e recebe um painel perpendicular, deve ser contínuo sem emendas nas guias superior e inferior no local de união com o painel 2, conforme a Fig. 4.

3 FUNDAMENTOS DE PROGRAMAÇÃO PARA SKETCHUP®

Na programação, uma forma usual de armazenar dados é a criação de variáveis, constantes, seqüências de caracteres e matrizes. No entanto, conforme as aplicações crescem em complexidade, torna-se difícil gerenciar milhares de valores desorganizados. Nesses casos, passa a ser vantajoso agrupar dados e operações em estruturas chamadas de objetos. A natureza dos dados e das operações é determinada pela classe do objeto, e as operações definidas em uma classe

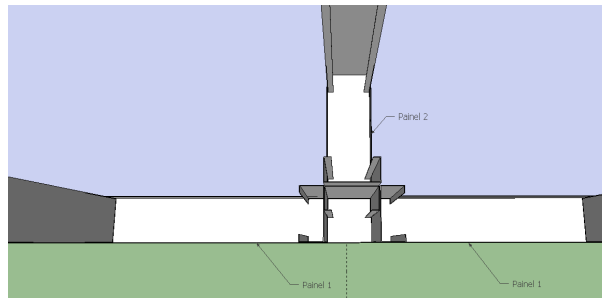


Figure 4: Combinação de perfis para montagem de verga.

são chamadas métodos.

Um objeto pode ser acessado tal qual uma variável, mas ao contrário de uma variável, um objeto contém vários valores relacionados. Por exemplo, num modelo geométrico, enquanto uma variável `altura_porta` identifica a altura de uma porta, um objeto do tipo `Porta` pode conter valores para a altura da porta, a largura, a profundidade, o material e a cor.

Dois objetos do mesmo tipo devem ter as mesmas características, mas não necessariamente os mesmos valores. Se o objeto do tipo `Porta` define a altura e a largura, e objetos `porta1` e `porta2` são ambos do tipo `Porta`, então `porta1` e `porta2` devem armazenar valores em altura e largura, mas não necessariamente os mesmos valores. O termo específico para o tipo de um objeto é Classe.

Um método é um procedimento definido em uma classe que opera em dados do objeto. Simplificando, se um objeto representa um determinado ente, um método proporciona um meio de interação com as características desse ente.

Quase todos os códigos escritos para SketchUp® começam acessando três estruturas básicas: `Sketchup`, `Model` e `Entities`. `Sketchup` é um módulo, que é uma coleção de métodos. Os métodos no módulo `Sketchup` acessam propriedades relacionadas com o programa SketchUp® como um todo.

O método mais importante no módulo `Sketchup` é o `active_model`. Esse método retorna um objeto `Model` correspondente ao projeto aberto no SketchUp® no momento. Assim como o módulo `Sketchup` representa o aplicativo inteiro SketchUp®, o objeto `Model` representa um único arquivo do SketchUp® (*.skp), ou mais precisamente, as informações de projeto contidas no arquivo.

Cada objeto geométrico em um projeto SketchUp® é representado por um objeto `Entity` ou uma subclasse do mesmo, incluindo as linhas, faces, imagens, textos, grupos e componentes. Então, para começar a programar um código para SketchUp®, o primeiro passo é chamar o método `entities` da classe `Model` e acessar o objeto `Entity` do projeto atual, tal qual na linha a seguir:

```
entidades = Sketchup.active_model.entities
```

A partir daí será possível, dentre outras coisas:

1. Adicionar novos objetos `Entity` ao projeto atual do SketchUp®, através dos métodos de adição `add_line`, `add_edges`, `add_circle`, `add_ngon`, `add_face` e `add_text`;
2. Mover, alterar a escala, rotacionar e apagar objetos `Entity` no projeto;
3. Armazenar objetos `Entity` em uma matriz que pode ser acessada pelo índice.

A lista com todas as classes e módulos do SketchUp® e seus métodos encontra-se na página para desenvolvedores em <http://www.sketchup.com/intl/en/developer/>. Já no desenvolvimento de aplicativos específicos torna-se necessária a criação de novas classes. O exemplo a seguir define a classe `Janela` e seus métodos, escritos em linguagem Ruby, denominados `grande` e `pequena`, que são funções compartilhadas. A função `initialize` indica que o objeto deve ser criado já indicando um valor que o identifique.

```
class Janela
  def initialize (id)
    @id = id

  end
  def grande
    largura = 1000.mm
    peitoril = 1100.mm
    puts "A janela "+@id.to_s+" grande e tem altura de
    "+largura.to_s+" e peitoril de "+peitoril.to_s+"."
  end
  def pequena
    largura = 600.mm
    peitoril = 1800.mm
    puts "A janela "+@id.to_s+" pequena e tem altura de
    "+largura.to_s+" e peitoril de "+peitoril.to_s+"."
  end
end
end
%
```

Em seguida, demonstra-se como são criados os objetos, atribuindo-se valores iniciais:

```
janela_grande = Janela::new(1)
janela_pequena = Janela::new(2)
outra_janela_grande = Janela::new(3)
```

E por fim, os objetos requisitam os métodos.

```
janela_grande = Janela::new(1)
janela_pequena = Janela::new(2)
outra_janela_grande = Janela::new(3)
```

4 MÉTODOS

Além de pesquisa bibliográfica para identificar as regras de montagens de painéis e lajes do LSF, foram feitas visitas técnicas a obras com esse sistema construtivo no município de Juiz de Fora, Minas Gerais, Brasil.

Foi realizado simultaneamente o estudo da linguagem de programação *Ruby* para que as classes existentes na biblioteca fornecida para construção de *plug-ins* pudessem ser compreendidas.

Em seguida o conhecimento adquirido foi modelado, explorando o potencial da programação orientada a objetos, através da criação de uma rede coerente de classes. Essas classes constituem o *script*, elaborado para apoiar o projeto em LSF, aliviando o projetista das atividades repetitivas.

A interação usuário-*script* ocorre através da interface do SketchUp®), onde foram criadas ferramentas acessíveis por ícones. O princípio do algoritmo é fundamentado nas estruturas básicas do LSF, a saber: montantes, guias, ombreiras e vergas.

Finalmente foram realizados testes com o *script*, tendo o usuário como o responsável por gerar painéis, janelas e portas sobre uma planta baixa escolhida por ele mesmo, dando origem a uma estrutura baseada em LSF.

4.1 Implementação computacional

O *software* utilizado como base para o projeto é o SketchUp®), o qual possui um ambiente de desenvolvimento integrado, em inglês *Integrated Development Environment* (IDE), que possibilita implementar códigos que gerem resultados no próprio SketchUp®, tornando então a modelagem 3D uma forma de impressão de resultados de algoritmos. A linguagem de programação que permite essa interação é o *Ruby*, que é imperativa, orientada a objetos e de tipagem dinâmica de variáveis. Também possui alto nível de ortogonalidade, facilitando a implementação de algoritmos.

Para Norman (1988, apud ROCHA; BARANAUSKAS, 2003, p. 24), existem quatro princípios para se obter um bom *design* de Interação Humano-Computador (IHC), sendo esses altamente inter-relacionados e difíceis de serem tratados isoladamente. Em poucas palavras, os quatro princípios podem ser definidos como:

- Visibilidade e *affordance*, onde apenas o necessário está visível e as propriedades percebidas determinam como o objeto deve ser usado (por exemplo, botão é para apertar, tesoura para cortar, borracha para apagar). O termo *affordance* pode ser entendido como o quanto a forma de um objeto tem de potencial para que ele seja manipulado da maneira em que foi pensado para funcionar (por exemplo, uma maçaneta redonda de porta convida o usuário a girá-la);
- Modelo conceitual, onde um bom modelo é o que permite prever o efeito das ações;
- Mapeamento, onde um bom mapeamento mostra os resultados gerados a partir da manipulação do usuário. Quanto mais naturais e próximos dos padrões culturais, mais facilmente será aprendido e lembrado (por exemplo, girar o volante de um carro em sentido horário o fará ir para a direita.);
- *Feedback*, que significa mostrar para o usuário o resultado de suas ações.

Procurando respeitar a esses princípios e empregando a Linguagem Unificada de Modelagem, do inglês *Unified Modeling Language* (UML), para demonstrar a composição do *script*, três diagramas foram gerados. Para Guedes (2014), a UML possui diversos diagramas, pois visa fornecer variadas visões sobre o sistema, permitindo encontrar possíveis falhas durante sua fase de desenvolvimento.

O primeiro deles, o diagrama de casos de uso, é o diagrama mais geral e informal da UML, sendo responsável por examinar os requisitos necessários para a compreensão do funcionamento do algoritmo, dando base para os demais diagramas. O diagrama de classes é o diagrama com maior uso e, segundo Guedes (2014), é também o mais importante da UML, uma vez que nele é definida a estrutura das classes empregadas no sistema, o modo como elas interagem entre si e seus métodos e atributos.

A Fig. 5 ilustra o diagrama de casos de uso para o *script* proposto, enquanto que a Fig. 6 representa o diagrama de classes. As classes Tool1, Tool2, Tool3, Tool4, Tool5 e Tool6 estão relacionadas com os ícones exibidos na interface do SketchUp® e são responsáveis pela criação dos painéis, janelas, portas, lajes, pela configuração dos dados e por gerar informação dos dados, respectivamente.

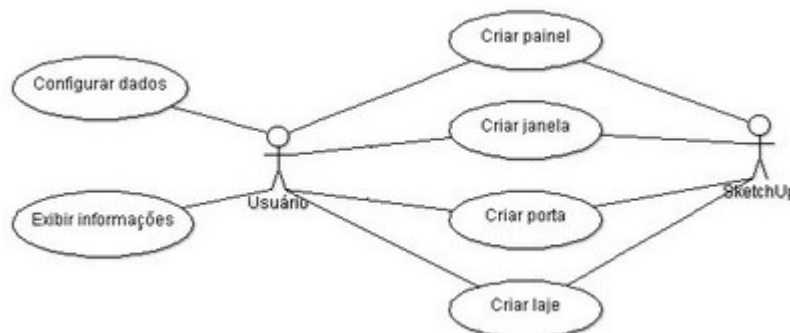


Figure 5: Diagrama de Casos de Uso

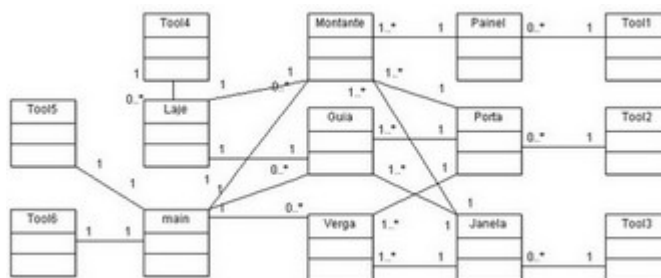


Figure 6: Diagrama de Classes

Já o Diagrama de Atividade busca traçar os caminhos a serem seguidos para efetuar alguma operação. É focado na demonstração do fluxo de controle e no fluxo de objeto de uma atividade. Isso é demonstrado na Fig. 7, apresentando o comportamento dos processos que seguem uma sequência de ações, do início ao fim de uma iteração.

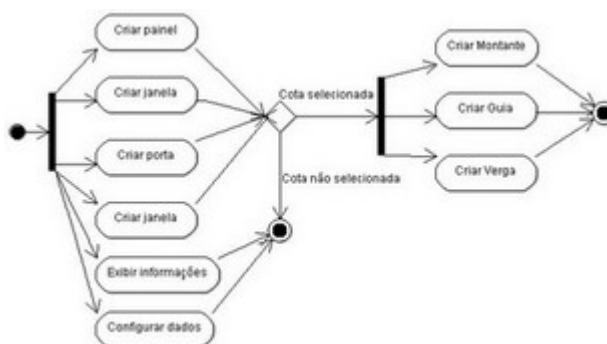


Figure 7: Diagrama de Atividades

5 RESULTADOS E DISCUSSÃO

A primeira fase do projeto consistiu no reconhecimento das dimensões padronizadas dos perfis de séries comerciais, as quais foram consultadas na “ABNT NBR 15253:2005 - Perfis de aço formados a frio, com revestimento metálico, para painéis estruturais reticulados de edificações – Requisitos gerais”. No entanto, para que se cumpra o objetivo de desenvolver uma ferramenta parametrizada, com maior flexibilidade, optou-se por permitir ao usuário determinar livremente as dimensões dos perfis a serem utilizados. A caixa de diálogo com o usuário, que recebe as dimensões, e o perfil gerado podem ser visualizados na Fig. 8

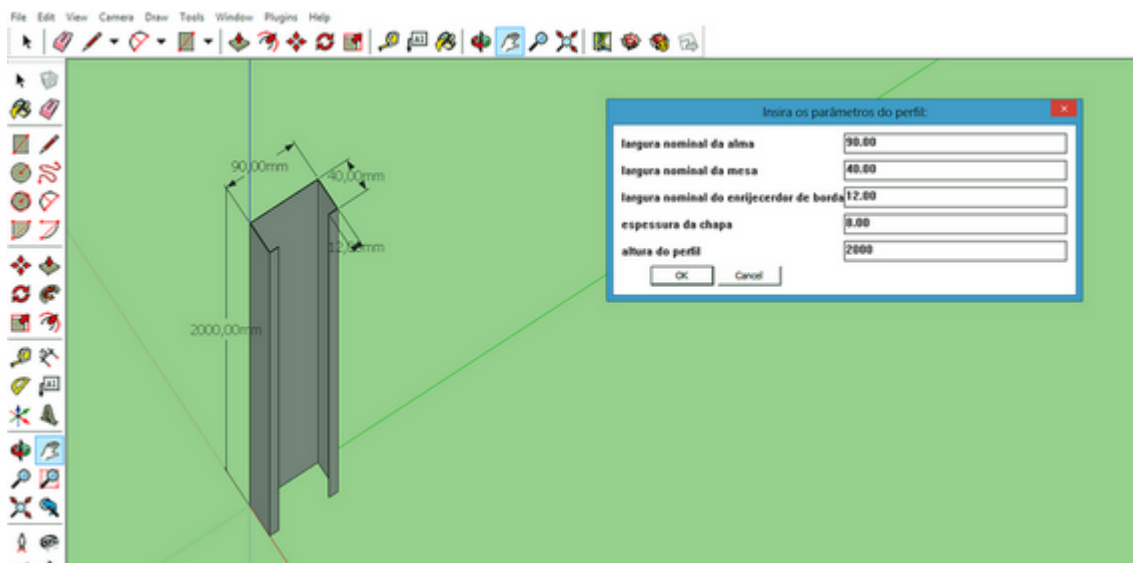


Figure 8: Modelagem de um perfil “U” enrijecido de dimensões 90x40x12x0,80 (mm) e altura 2m.

Recomenda-se utilizar esse algoritmo juntamente com uma planta baixa para auxiliar a montagem das estruturas. Segue um breve exemplo de funcionamento da ferramenta, apresentada na Fig. 9, que é composta de seis botões, que executam as funções de criar painel, janela, porta, laje, configurar dados e exibir informações sobre as peças utilizadas.



Figure 9: Barra de ferramentas

Na utilização prática dessas funções deve-se inicialmente cotar o local onde a estrutura será montada, seguindo a direção anti-horária. Para a montagem dos painéis, portas e janelas, a linha a ser cotada deve ser a interna do cômodo, enquanto que para a laje deve-se utilizar a linha externa. Com a cota selecionada, basta então clicar no ícone da estrutura desejada. Para obter informações ou alterar valores como pé-direito e espaçamento dos montantes, por exemplo, não há necessidade de selecionar cota. As Figs. 10 a 12 demonstram esses passos.

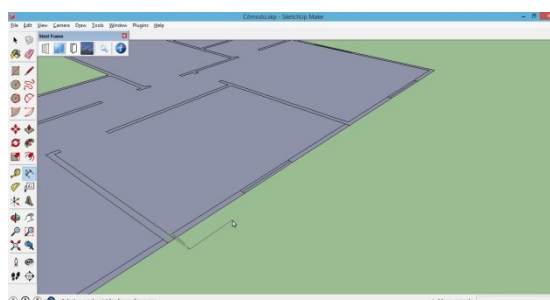


Figure 10: Utilização da Cota e Barra de ferramentas

Depois de cotado, o local estará apto a receber uma das estruturas. No exemplo, foi adicionado um painel sem abertura (Fig. 11) e posteriormente uma janela e outro painel sem abertura (Fig. 12).

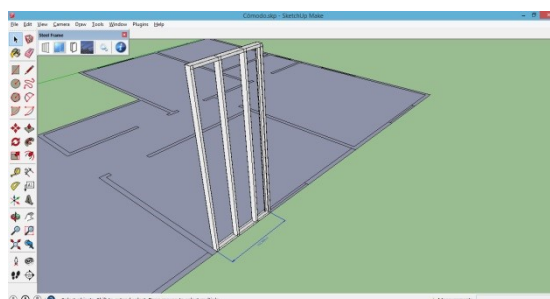


Figure 11: Montagem de painel sem abertura

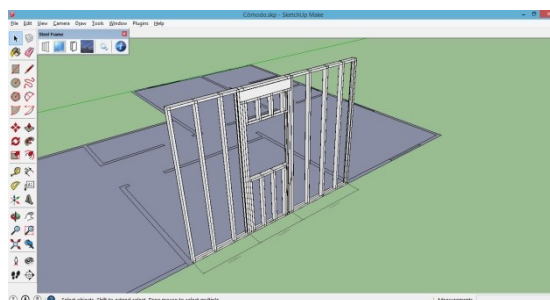


Figure 12: Montagem de painel com abertura

Seguindo o mesmo procedimento, torna-se simples concluir a montagem de um cômodo. Em seguida, para a montagem de uma laje sobre o cômodo, basta cotar toda a área do cômodo, uma cota para a largura e outra para o comprimento. Selecionando as duas cotas simultaneamente e clicando sobre o ícone de montar laje da ferramenta, a laje será criada. Daí em diante os passos se repetem, até a conclusão de toda obra (Fig. 13)

Também é possível configurar valores como pé direito da edificação, espaçamento entre montantes, altura do peitoril, etc. E por fim, é possível obter informações como quantidade de montantes e o comprimento total de guias.

6 CONCLUSÕES

A análise da situação do desenvolvimento do sistema construtivo LSF no Brasil revela a escassez de ferramentas que visem facilitar a concepção de projetos, sendo que a etapa de projeto

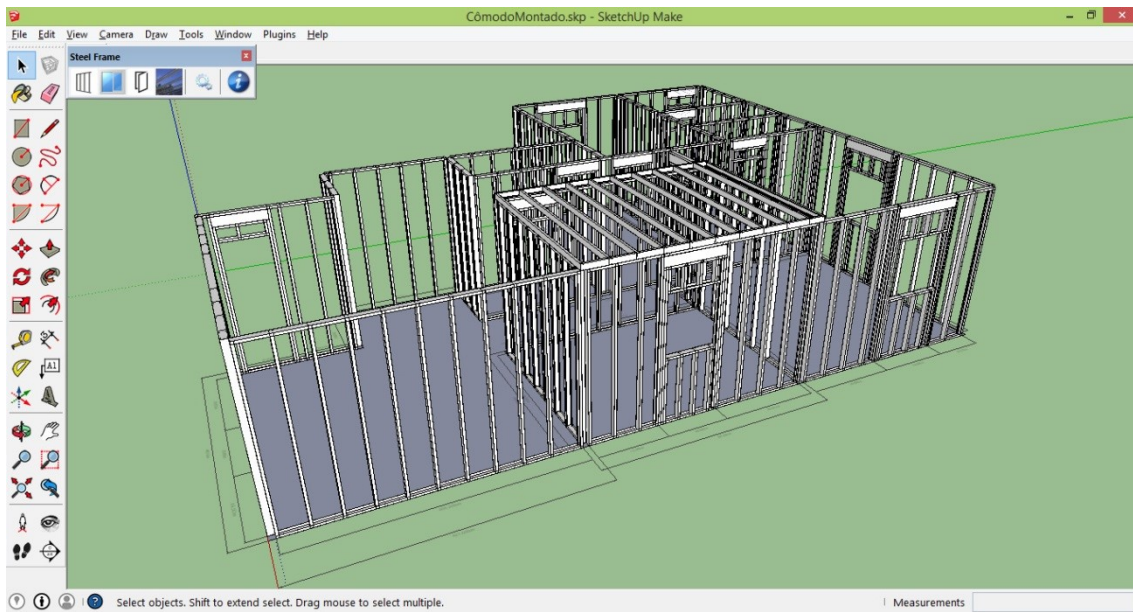


Figure 13: Montagem completa sobre toda a planta

destaca-se por exercer forte impacto em relação à eficiência no processo de produção da obra. Como ressalta Usuda (2003), é nesta etapa que acontecem a concepção e o desenvolvimento do produto que devem ser baseados na identificação das necessidades dos clientes em termos de desempenho, custos e condições de exposição a que será submetida a edificação.

Criar recursos personalizados para que os usuários estabeleçam suas próprias práticas, interagindo com análises, estimativas de custos e outras aplicações é uma proposta que vem ganhando cada vez mais espaço no setor. A ideia de uma plataforma comum que permita gerir e transferir os dados entre diferentes aplicações de forma unificada afigura-se como adequada, mas ainda está longe de ser concretizada. Por isso o campo de pesquisa ainda se apresenta aberto a todos os programas disponíveis no mercado.

À primeira vista, o SketchUp se assemelha a um aplicativo para desenho. Mas, conforme o usuário se torna mais familiarizado com o programa pode apreciar a sua amplitude de funções e capacidades para modelagem profissional. Através da escrita de códigos pode-se tirar o máximo de proveito do programa. Quando executado, um *script* SketchUp pode criar instantaneamente objetos, modelos, materiais e até mesmo animações. Ressalta-se que a API Ruby está completamente incluída na versão gratuita do SketchUp.

As classes do SketchUp e seus métodos são fáceis de entender, mas pode ser difícil de coordená-las para formar um projeto. Por isso esta equipe tem praticado continuamente a fim de se construir um plug-in, que transforme o SketchUp em uma ferramenta gratuita para a criação de pré-projetos em LSF, contribuindo para a difusão do sistema construtivo. Este trabalho apresenta os resultados preliminares da ferramenta em desenvolvimento, sendo que o próximo passo será o convite de um grupo de usuários para testá-lo a fim de que forneçam um retorno sobre o seu funcionamento.

Como metas futuras, deseja-se mesclar as capacidades do aplicativo com sistemas de análise assistida por computador, do inglês *computer aided analysis* (CAA), unindo as competências de raciocínio, captura de conhecimento e capacidade de representação. Dessa forma, além da automação de tarefas repetitivas e não criativas, será dado suporte à otimização do projeto multidisciplinar em todas as fases do processo.

7 AGRADECIMENTOS

Os autores agradecem o apoio concedido pela UFJF, FAPEMIG, CNPq e CAPES para a realização deste trabalho.

REFERENCES

- Crasto R.C.M.d. *Arquitetura e Tecnologia em Sistemas Construtivos Industrializados: LSF*. 2005. Ph.D. thesis, Dissertação (Mestrado em Engenharia Civil) - Escola de Minas, Universidade Federal de Ouro Preto, Ouro Preto, 2005.
- Freitas A.M.S. and Crasto R.C.M.d. *Steel Framing - Arquitetura*, volume 1. Instituto Brasileiro de Siderurgia/Centro Brasileiro da Construção em Aço, 2006.
- Mattei P.L.d.R. Bim e a informação no subsetor de edificações da indústria da construção civil. 2008.
- Melo E.G.d. Geração solar fotovoltaica: estimativa do fator de sombreamento e irradiação em modelos tridimensionais de edificações. 2012.
- Rego D.J.M. *Estruturas de edifícios em Light Steel Framing*. Ph.D. thesis, Dissertação (Mestrado em Engenharia Civil), Universidade Técnica de Lisboa, Lisboa, 2002.
- Rocca G.L. Knowledge based engineering: Between {AI} and cad. review of a language based technology to support engineering design. *Advanced Engineering Informatics*, 26(2):159 – 179, 2012. ISSN 1474-0346. doi:<http://dx.doi.org/10.1016/j.aei.2012.02.002>. Knowledge based engineering to support complex product design.
- Santiago A.K. and Araújo E.C.d. Sistema light steel framing como fechamento externo vertical industrializado. In *Congresso Latino Americano da Construção Metálica*, volume 3. 2008.
- Schwegler B.R., Fischer M.A., O'Connell M.J., Hänninen R., and Laitinen J. Near, medium and long term benefits of information technology in construction. In *Proceedings of the International Conference on Construction*. 2001.
- Tavares E. Os 10 maiores produtores de aço do mundo. *Revista Exame*, 2011.
- Tse T.K., Wong K.A., and Wong K. The utilisation of building information models in nd modelling: a study of data interfacing and adoption barriers. 2005.
- Vivan A.L. Projetos para produção de residências unifamiliares em light steel framing. *São Carlos*, 2011.