

# EXPLORANDO SOFTWARES COM CÓDIGO ABERTO PARA VISUALIZAÇÃO E GERAÇÃO DE QUANTITATIVOS A PARTIR DE MODELOS IFC

**Fabiano Rogerio Corrêa; Sérgio Leal Ferreira;  
Daniel dos Reis Neto; Eduardo Toledo Santos**

Departamento de Engenharia de Construção Civil, Escola Politécnica da Universidade de São Paulo  
e-mail: [fabiano.corrêa@poli.usp.br](mailto:fabiano.corrêa@poli.usp.br)

## **Resumo**

*O objetivo deste trabalho é demonstrar que, a partir da utilização de bibliotecas com código aberto, é possível criar softwares para explorar as informações presentes em modelos IFC. Para tanto, desenvolveu-se um programa para visualização da geometria tridimensional e a extração de quantitativos destes modelos. A biblioteca mais importante para o desenvolvimento realizado neste trabalho é a IfcOpenShell, que processa um arquivo contendo um modelo IFC, carregando em memória todas as informações nele contidas, por meio de uma estrutura de dados que reflete a própria hierarquia do modelo. Além disso, foi utilizada uma das funcionalidades desta biblioteca, que extrai a geometria implícita do modelo e a transforma em geometria explícita que pode ser visualizada no computador. Para a geração dos quantitativos, os valores bases como altura, comprimento e largura de cada objeto foram calculados diretamente da geometria obtida e as demais quantidades foram calculadas a partir destes valores. Os quantitativos escolhidos e a metodologia dos cálculos seguiram o padrão utilizado pela ferramenta Revit da Autodesk, como um estudo de caso. As informações extraídas do modelo IFC foram posteriormente armazenadas num banco de dados orientado a objetos, que é uma maneira mais natural de lidar com o tipo de informação de um modelo BIM.*

**Palavras-chave:** Modelo 3D, IFC, BIM, Modelagem da Informação da Construção.

## **Abstract**

*The objective of this research is to demonstrate that starting from open-source libraries it is possible to develop software to explore information inside IFC models. Thus, we implemented tridimensional geometric visualization and quantity take-off from those models. The most important library in this development is IfcOpenShell, which processes a file containing an IFC model, loading in memory all the information in it, through a data structure that resembles the model hierarchy. Besides that, there is a functionality that extracts the implicit geometry present in the model, transforming it in explicit geometry that could be visualized in the computer. The base values of each object, as width, length and height, were extracted directly from geometry, and other quantities were calculated using these base values. As a case study, the chosen quantity types and the calculation method followed Revit standards. The information extracted from IFC models were then stored in an object database, which is a natural structure to interact with information from a BIM model.*

**Keywords:** 3D model, Industry Foundation Classes, BIM, Building Information Modeling.

## **1. INTRODUÇÃO**

A utilização do BIM como modelo de informações da construção pressupõe a possibilidade de gerenciamento, compartilhamento e troca de informações de forma colaborativa entre as diferentes disciplinas envolvidas no ciclo de vida da edificação. Desta forma surge a demanda pela troca de informações entre distintos programas de computador que suportam BIM e

programas de apoio às atividades de projeto, gerenciamento e orçamento. Dentro deste cenário, a interoperabilidade é essencial para a otimização do projeto e a viabilização da cooperação entre os diversos agentes envolvidos.

Visando à interoperabilidade, foi desenvolvido o esquema IFC (*Industry Foundation Classes*) que define um conjunto de classes para o domínio da AEC (Arquitetura, Engenharia Civil e Construção) que permite representar um edifício por meio de um conjunto de objetos interrelacionados. Sendo o pilar da iniciativa OpenBIM (OPEN BIM, 2012) da *buildingSmart* (antiga *International Alliance for Interoperability*), o IFC permite que aplicativos BIM importem ou exportem o modelo da edificação num formato neutro, permitindo a troca de informações entre softwares de diferentes fabricantes.

Além disso, durante a colaboração entre os diferentes profissionais que manipulam o modelo BIM, cada qual tem um interesse particular em partes do modelo (ZHANG e ISSA, 2011). É necessário então prover meios para extrair e processar as informações do modelo que lhe são relevantes. Muitas aplicações podem utilizar-se dos dados presentes no modelo IFC.

Portanto decidiu-se desenvolver um software, utilizando bibliotecas de código aberto, capaz de ler um modelo IFC, exibir a sua geometria tridimensional e disponibilizar os quantitativos de alguns de seus elementos, para demonstrar que é possível explorar o modelo IFC, atendendo a demanda de funcionalidades pelos profissionais da AEC.

## 2. O ARQUIVO .IFC

Na Tabela 1 está representada parte da estrutura de um arquivo .ifc. O arquivo é composto por linhas numeradas (ex:#893), contendo classes específicas (ex: IFCWALLSTANDARDCASE) seguidas de alguns parâmetros (ex: '117653'). As informações pertinentes a um determinado objeto do edifício estão espalhadas em diversas linhas do arquivo, ligadas entre si por referências. Objetiva-se processar essas informações para extrair a geometria e calcular quantitativos. Maiores detalhes sobre o esquema IFC estão disponíveis em EASTMAN *et al.*, (2008) e BUILDINGSMART (2012).

Tabela 1 – Parte do conteúdo de um arquivo IFC

```
...
#893=IFCWALLSTANDARDCASE('37ijgimhnBwQPpGBsMWWJi',#34,'Basic
Wall:Foundation - 305 Concrete:117653',$,'Basic Wall:Foundation - 305
Concrete:29243',#880,#892,'117653');
#894=IFCQUANTITYLENGTH('Height',$,$,6727.778863182019);
...
```

### 2.1. Geometria

A geometria do edifício encontra-se implicitamente definida nos modelos IFC. Tome-se como exemplo uma parede: um meio de defini-la é representar seu perfil (que pode ser um retângulo), e indicar sua construção como sólido por meio da extrusão deste perfil até uma determinada altura (seguindo uma direção normalmente vertical). Além disso, é necessária a definição de sistemas de coordenadas locais para o posicionamento do perfil e, consequentemente, da parede no edifício.

Assim, a geometria de uma parede necessita de várias classes diferentes para sua representação em IFC: *IfcShapeRepresentation*, *IfcExtrudedAreaSolid*, *IfcRectangleProfileDef*, *IfcAxis2Placement2D*, *IfcCartesianPoint*, *IfcLocalPlacement*, *IfcAxis2Placement3D*, *IfcRelContainedInSpatialStructure*, *IfcDirection*,

*IfcWallStandardCase*.

## 2.2. Quantitativos

As quantidades exportadas em IFC encontram-se principalmente nas classes *IfcQuantityArea*, *IfcQuantityCount*, *IfcQuantityLength* e *IfcQuantityVolume*. Além disso, no caso da exportação do modelo IFC pelo aplicativo *Autodesk Revit*, é associado aos objetos um conjunto de propriedades (*Pset\_Revit\_XXX*) do qual é possível obter suas dimensões principais.

Para demonstrar a manipulação de dados do modelo IFC, decidiu-se calcular as mesmas quantidades que o programa *Autodesk Revit* avalia quando opta-se pela exportação de quantidades no modelo em IFC. Isto implica na necessidade de se calcular as três dimensões principais de cada objeto (altura, largura e comprimento de uma parede, por exemplo) a partir da geometria encontrada para, destes valores, determinar-se grandezas como área lateral e volume. Os valores contidos nos conjuntos de propriedades do *Revit* não foram utilizados para os cálculos, mas apenas para validação dos resultados.

## 3. BIBLIOTECAS DE MANIPULAÇÃO

A biblioteca essencial para o funcionamento do programa proposto é a **IfcOpenShell** (2012), que permite o tratamento das informações de um modelo IFC. No entanto, para obter outras funcionalidades, como visualização e armazenamento de dados, outros programas/bibliotecas foram utilizados.

Em resumo, foram empregadas as seguintes bibliotecas abertas:

- **Open CASCADE**: é uma plataforma de desenvolvimento de software, disponibilizada gratuitamente por meio de seu código-fonte. Inclui componentes para modelagem 3D de superfícies e sólidos, visualização e troca de dados, especificados na linguagem C++ (OPEN CASCADE, 2012);
- **IfcOpenShell**: é uma biblioteca de código aberto, também em C++, que auxilia usuários e desenvolvedores a trabalhar com os modelos IFC. Ela usa internamente o Open CASCADE para converter a geometria implícita no modelo IFC para uma geometria explícita que qualquer software CAD ou pacote de modelagem possa interpretar. Duas classes destacam-se em importância: *IfcParse* e *IfcObj*;
- **OpenGL**: é uma biblioteca gráfica aberta, disponível para uso em várias linguagens de programação, que facilita o desenvolvimento de aplicativos para visualização computacional (OPENGL, 2012);
- **Db4o**: é um banco de dados orientado a objetos, que possui duas versões, uma delas gratuita. Basta adicionar as bibliotecas ao seu projeto de desenvolvimento para adquirir as funcionalidades de armazenamento e manipulação de objetos. Possui uma versão desenvolvida para .NET e outra em Java.

A seguir, são apresentados detalhes sobre a utilização de cada uma delas neste trabalho.

### 3.1. Manipulação das informações do modelo IFC

No site da Open BIM Network (2012), estão disponíveis programas com código aberto para auxiliar na modelagem e visualização das informações da construção. O **IfcOpenShell** é um destes programas. Foi criado para interpretar arquivos IFC e permitir o rápido

desenvolvimento de aplicativos que fazem uso das informações contidas nestes arquivos, como, por exemplo, a extração da geometria implícita (no formato *Wave OBJ*).

A interpretação do arquivo IFC é feito por objetos da classe *IfcParse*, da biblioteca **IfcOpenShell**, que procura pelo caractere # indicando o início de uma linha no arquivo. Para cada classe encontrada, é montada uma lista incluindo o ID (#ID) e o local na memória referente ao arquivo, onde esta linha se encontra.

Depois de processado todo o arquivo, é possível consultar os objetos por meio de suas classes. Existem três maneiras de acessar os objetos vindos do arquivo IFC:

- *EntitiesByType*: permite o acesso às entidades do modelo a partir de um tipo específico de objeto.

Por exemplo, *Ifc::EntitiesByType<Ifc2x3::IfcWallStandardCase>()* permite o acesso a uma lista das entidades *IfcWallStandardCase* presentes no modelo IFC;

- *EntitiesByReference*: a partir da identificação de um objeto específico, permite o acesso a uma lista de todas as entidades que fazem referência à identificação utilizada.

Por exemplo, a partir de uma parede é possível recuperar todas as suas propriedades, espalhadas em diversas classes.

- *EntitiesById*: retorna o objeto que possui determinada identificação;

As entidades obtidas (*IfcUtil::IfcBaseClass*) desta maneira precisam ser convertidas para classes específicas (*Ifc2x3::IfcWallStandardCase*) para uma manipulação adequada e para a utilização de todas as funcionalidades do objeto. Isto pode ser feito através de uma conversão de tipos:

```
reinterpret_pointer_cast<IfcUtil::IfcBaseClass,Ifc2x3::IfcWallStandardCase>
```

Objetos da classe *IfcObj* desta biblioteca são um exemplo do uso do *IfcParse* (interpretador) para obter a geometria. A extração da geometria utiliza o **Open CASCADE** que contém inúmeras operações de construção de sólidos e superfícies, para recriar qualquer geometria no formato *Wave OBJ*.

As informações mais importantes do formato *Wave OBJ* utilizado pela biblioteca são uma lista de vértices (v -2.27006 -19.3556 0), uma lista de normais nos vértices (vn 0 0 -1) e uma lista de faces que referenciam um conjunto de vértices (f -23//-23 -24//-24 -22//-22), em sequência, para cada objeto. O número negativo significa que a numeração dos vértices é contada a partir do último, e as barras separam a informação: vértice, textura, normal. Como o arquivo não contém a textura, nota-se a presença de duas barras uma ao lado da outra.

### 3.2. Visualização Computacional

A grande maioria das aplicações que incluem modelos tridimensionais a serem visualizados num computador empregam a biblioteca **OpenGL**. Para a visualização da geometria do modelo IFC criou-se um programa em C++ que transforma o arquivo OBJ em dados para a biblioteca **OpenGL**. Cada face do objeto é representada por um conjunto de triângulos, definidos por seus vértices.

### 3.3. Banco de dados orientado a objetos

Dentro da mesma iniciativa que produziu a biblioteca *IfcOpenShell*, existe o programa *BIMServer*, para gerenciamento de modelos BIM. Ele implementa um banco de dados relacional, que é o tipo mais largamente utilizado pela maior parte das empresas e aplicações.

Decidiu-se avaliar uma solução alternativa, e testar um banco de dados orientado a objetos, já que a própria estrutura do IFC é de objetos e relações entre eles.

Assim, as informações foram armazenadas num banco de dados orientado a objeto, o **Db4o** (2012), também gratuito. Ele possui um *plug-in* que pode ser instalado no ambiente de desenvolvimento *Visual Studio*, permitindo abrir uma conexão com um banco de dados, visualizar os objetos armazenados e recuperar objetos específicos por meio de buscas.

Existem três mecanismos para se realizar uma busca (*query*) no banco de dados: busca por exemplos (QBE – Query By Example), buscas nativas, e a LINQ. Todas elas usam a API SODA para interagir com o banco de dados.

#### 4. PROGRAMA DESENVOLVIDO E RESULTADOS OBTIDOS

O ambiente de desenvolvimento utilizado para a construção do programa foi o *Visual Studio C++ 2010*, da *Microsoft*. A versão *Express* deste ambiente pode ser obtida gratuitamente no site da empresa. Tendo os códigos fonte de todas as bibliotecas, é necessário compilar para um sistema em particular.

Foram implementadas as seguintes funcionalidades: visualização da geometria e extração de quantitativos.

Para a extração dos quantitativos, é necessário analisar a relação entre diferentes objetos para determinar os valores que são calculados pelo *Revit*.

Por exemplo, considere-se o caso de uma planta simples, com quatro paredes e um piso. Dados os comprimentos de cada uma das paredes, tem-se o seguinte arranjo entre elas:

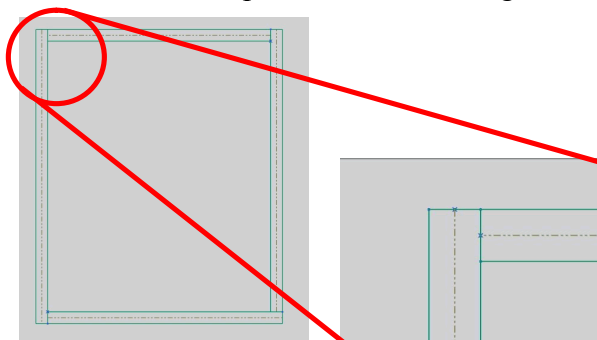


Figura 1-Detalhes do arranjo entre as paredes.

As linhas internas às paredes correspondem ao comprimento das mesmas. Ao juntar uma parede na outra, as paredes interiores tem um comprimento diferente.

Os cálculos são feitos considerando a linha média das paredes. Apesar das conexões entre as paredes serem as mesmas - *IfcRelConnectsPathElements* (ATSTART, ATEND) - a ordem das conexões faz com que a primeira e a última parede acabem recebendo duas conexões iguais, ficando uma com comprimento maior e a outra, menor.

A parede à esquerda perde 200 mm (espessura das paredes), a de baixo e à direita perdem e ganham 100 mm, ficando com o mesmo tamanho, e a parede de cima acaba ganhando 200 mm. São estes os valores obtidos para os cálculos de área lateral bruta, volume bruto, e área bruta da base. A Tabela 2 contém os quantitativos extraídos de cada objeto considerado.

Tabela 2 – Quantitativos extraídos referentes aos objetos testados

Classes	Quantitativos
<i>IfcBeam</i>	<i>GrossVolume, OuterSurfaceArea, CrossSectionArea, Length</i>
<i>IfcColumn</i>	
<i>IfcDoor</i>	<i>Height, Width</i>
<i>IfcWindow</i>	
<i>IfcWallStandardCase</i>	<i>GrossVolume, GrossSideArea, GrossFootprintArea, Width, Length, Height</i>
<i>IfcSlab</i>	<i>CrossArea, GrossVolume, Perimeter, Width</i>

## 5. CONCLUSÕES E TRABALHOS FUTUROS

As bibliotecas que estão disponibilizadas gratuitamente na Internet permitem o desenvolvimento de softwares profissionais. Não é necessário muito esforço para trabalhar adequadamente com as mesmas. Compatibilizar todas elas com um sistema operacional específico, no entanto, pode trazer dificuldades iniciais. Neste trabalho, consegui-se visualizar modelos IFC com centenas de objetos, e extrair de diferentes objetos seus quantitativos.

Atualmente, os trabalhos estão concentrados numa implementação deste programa para dispositivos móveis, que rodam o sistema operacional **Android**, do Google, que também é aberto. Apesar de o desenvolvimento para este sistema ser feito em linguagem Java, é possível combinar a ele código nativo em C/C++. Assim, há uma versão do **Open CASCADE** já pronta para este sistema. O banco de dados **Db4o** possui uma versão para Java, e o **OpenGL** também tem uma versão específica, **OpenGL ES**, para dispositivos móveis. Resta apenas criar uma versão do **IfcOpenShell** para o **Android**.

O direcionamento deste trabalho será a manipulação das informações do modelo IFC para atender a diversas demandas dos usuários de ferramentas BIM. Planos para o futuro incluem tanto um estudo mais aprofundado sobre a interação com um banco de dados orientado a objetos quanto a transformação do modelo IFC numa ontologia (BARBAU *et al.*, 2012; BEETZ *et al.*, 2009), para que seja possível inferir informações por meio de uma linguagem lógica.

## REFERÊNCIAS

- BARBAU, R., et al. OntoSTEP: Enriching product model data using ontologies. In **Computer-Aided Design**, 44, pg. 575-590, 2012.
- BEETZ, J.; LEEUWEN, J. V.; DE VRIES, B. IfcOWL: a case of transforming EXPRESS schemas into ontologies. In: **Artificial Intelligence for Engineering Design, Analysis and Manufacturing**, 23, pg 89-101, 2009.
- BUILDINGSMART. IFC Standard. Disponível em: <http://www.buildingsmart-tech.org/specifications/ifc-overview/ifc-overview-summary>. Acesso em: 01 Jun 2012
- DB4O. Versant. Disponível em: < <http://www.db4o.com> >. Acesso em: 01 Jun 2012
- EASTMAN, C. et al. **BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors**, New Jersey: John Wiley & Sons, Inc., 2008.
- IFCOPENSHELL. Disponível em: < <http://ifcopenshell.org> >. Acesso em: 01 Jun 2012
- OPEN BIM. buildingSmart. Disponível em: < <http://buildingsmart.com/openbim> >. Acesso em: 01 Jun 2012
- OPEN BIM NETWORK. Disponível em: <http://osbim.org>. Acesso em: 01 Jun 2012
- OPEN CASCADE. Disponível em: <http://opencascade.org>. Acesso em: 01 Jun 2012
- OPEN GL. Disponível em: < <http://www.opengl.org/sdk> >. Acesso em: 01 Jun 2012
- ZHANG, L., ISSA, R. R. A. Ontology based partial building information model extraction. In: Proceedings of the CIB W78-W102, 2011.