

# PROJETO DE SALAS-COFRE: DIMENSIONAMENTO E CÁLCULO DE COMPONENTES

Ricardo H. Medrano  
Universidade Presbiteriana Mackenzie (Brasil)  
Faculdade de Arquitetura e Urbanismo  
Endereço postal  
hmedrano@gmail.com

## Abstract

### *Strong-rooms project: sizing and calculation of components*

*This paper deals with a project that has been done to Aceco TI Ltda's company. It is about a software development, in AutoLisp programming language (under AutoCAD's platform), which helps to the project and calculation of strong-rooms components. Those rooms are completely sealed, making use of high technologies materials as sealing, allowing computational equipments and enclosed information to be protected. The developed programme contains routines that allow the precise dimension of structural component, as floors, walls, doors and ceilings. Therefore, it is possible to fabricate special pieces with accuracy. Moreover, the programme calculates a series of other components as friezes, drips, luminaries, sealing tapes and silicone conduits. Then, yonder saving time on project, this system allows, on the one hand, each piece to be produced in factory with exactness accuracy, making assembly easier. On the other hand, allows assembly to be started with all necessary elements, reducing time and costs.*

## 1. Introdução

Este projeto trata da realização de um programa em linguagem de AutoLISP, baseada no AutoCAD, visando o projeto de Salas-Cofre, e foi desenvolvido para a empresa ACECO TI Ltda, sediada em São Paulo e com filiais e representações pelo Brasil e América Latina.

Uma Sala-cofre é um produto para proteger equipamentos e informação, como vemos na Figura 01.

O objetivo seguido no desenvolvimento deste programa é facilitar e dar mais segurança ao projeto, bem como obter uma listagem com todos os componentes necessários, visando a montagem. É importante ressaltar que as Salas-Cofre são um produto de alta tecnologia e que envolvem custos consideráveis, o que demanda um controle estrito para que não haja problemas de



figura 1.

montagem, o que acarretaria um custo elevado.

Portanto, a ênfase aqui é dada ao projeto de Salas-Cofre, e não à linguagem de programação. Isto condiz também com nossa formação, que é de arquiteto, sendo que podemos incluir o projeto na área do Desenho Industrial.

É importante ressaltar que nas aulas das disciplinas de informática na Universidade Mackenzie procuramos evidenciar a utilidade deste tipo de desenvolvimento. Mostramos que existem basicamente três tipos de maneiras de trabalhar com programas Cad: uma é utilizar programas que permitam manualmente realizar qualquer (ou quase qualquer) forma, utilizando por exemplo o próprio AutoCAD, o 3D Studio ou o Vector Works, entre muitos outros. Este processo permite uma grande liberdade de criação, mas é quase que totalmente manual. Outra opção são os programas já preparados para a arquitetura, como por exemplo o Architectural Desktop, o Revit, o Vector Works, etc. Programas deste tipo agilizam significativamente o projeto e o desenho técnico, já que uma série de elementos já estão prontos e cortes e elevações são gerados automaticamente. O inconveniente deste tipo de programa é que há limitações quanto aos elementos que podem ser utilizados, que são os existentes no próprio programa. Nos

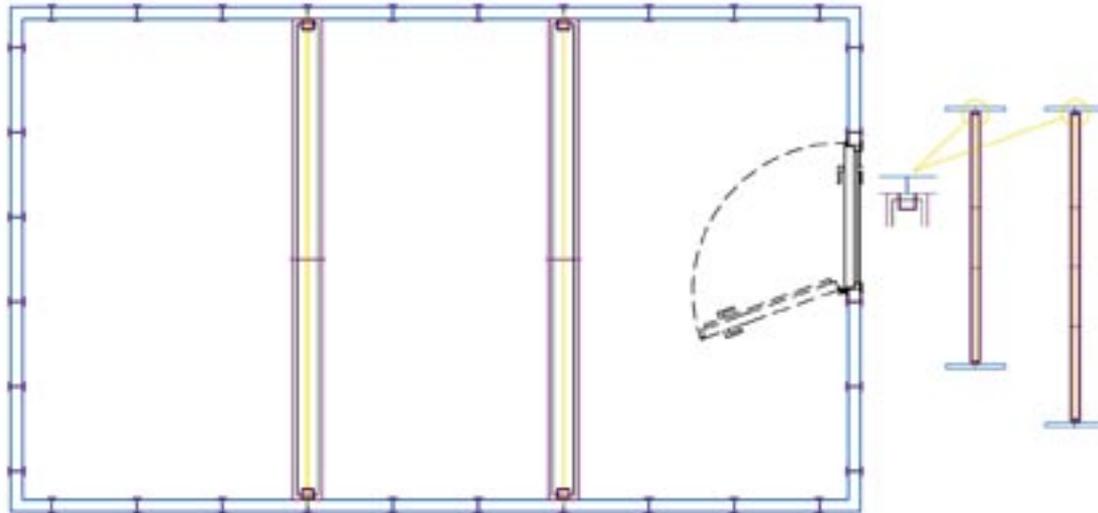


Figura 2.

casos não previstos é necessário trabalhar de forma complementar com formas primitivas, extrusões ou as muitas possibilidades existentes para as superfícies. Por último, mostramos alguns exemplos de rotinas em LISP, todas elas extremamente simples e curtas, mas que permitem uma enorme economia de tempo, pois realizam automaticamente tarefas que manualmente seriam exaustivas. Estas rotinas funcionam como aplicativos dentro do CAD, fazendo o mesmo papel que os programas já prontos mas adaptados à própria realidade específica. Quando não existe um programa no mercado pronto para determinada aplicação, a solução mais econômica e eficiente é o desenvolvimento de um sistema próprio.

Na seqüência descrevemos partes do programa (por limitações de espaço), e são explicadas com maiores detalhes aquelas operações que exigiram soluções específicas.

## 2. O programa

### 2.1. Construção das paredes

Na primeira parte do menu temos as opções para construir as paredes da Sala-Cofre, ou seja, os elementos laterais e cantos.

É importante ressaltar que o programa permite que estes elementos possam ser tanto padrão como especiais, na forma e nas dimensões. Toda vez que temos um elemento com medidas padrão usamos um bloco com atributos. Caso seja um elemento especial utilizamos polylines com dados estendidos (xdata). Em tempo: utilizamos neste trabalho como exemplo uma sala retangular para melhor compreensão, mas a mesma pode ter qualquer forma e dimensão. Para facilitar ao máximo o processo de projeto, incluímos uma série de opções que fazem as modificações necessárias automaticamente.

### 2.2. Portas, pilares, vigas e frisos

Na seqüência temos um outro menu. As opções são:

- Portas direita e esquerda.
- Pilares.
- Vigas.
- Perfis dos elementos laterais.
- Perfis das vigas. Selecionando todas as vigas, o programa as detecta automaticamente e coloca os perfis na medida certa, além das emendas que nesses perfis têm função estética (Figura 02). Neste caso, dependendo do comprimento da viga, o programa coloca as peças de forma simétrica, ou seja, os perfis da ponta, que

possuem um recorte por causa do pilar, são inteiros (a menos que o comprimento total seja menor que a soma de 2 perfis) enquanto o(s) central(is) são iguais (Figura 02).

### 2.3. Alturas

A próxima parte do menu serve para atribuir valores às alturas dos elementos laterais, frisos e portas e soleiras. Neste caso, como são blocos com atributos, utilizamos a seguinte rotina para selecionar os blocos:

```
(setq conj (ssget "X" '((-4. "<OR")
.
(-4. "OR>"))
)
)
```

Na seqüência utilizamos a rotina abaixo para substituir os valores dos atributos, dentro de um looping:

```
(setq c (entget (entnext (entnext a))))
(setq compat (assoc '1 c))
(setq compnov (cons 1 comp))
(setq c (subst compnov compat c))
(entmod c)
(setq n (+ n 1))
(setq a (ssname conj n))
```

### 2.4. Elementos de fundo e teto

A próxima etapa de projeto é construir os elementos de fundo e teto:

O elemento de fundo é realizado a partir de uma hachura já com as dimensões corretas (opção “Offset e Hachura”). Depois utilizamos a opção “Polylines na Malha”, que é uma rotina que permite fazer uma polyline (e apenas uma) dentro de cada área da hachura, como uma espécie de “bpoly” automático.

Para evitar que existam duas ou mais polylines iguais, uma subrotina faz uma verificação comparando o ponto médio das polylines que vão sendo construídas com o ponto médio das existentes. Se já existir uma polyline igual, a nova será apagada. Senão, acrescentada a um conjunto de dados.

Continuando, ao usarmos a opção “Elementos de

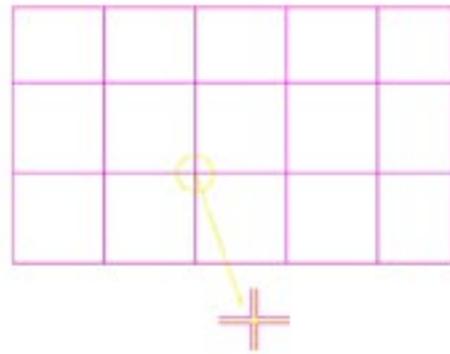


Figura 3.

Fundo”, o programa detecta todas essas polylines, realiza um “offset” para dentro (de forma a que exista uma junta entre as peças), e através de dados estendidos (xdata) coloca diversas informações em cada polyline, entre as quais as dimensões (Figura 03). Detalhe: esta rotina utiliza um sistema que garante que o offset será sempre para dentro, o que não aconteceria em certas formas irregulares se usássemos como referência o centro geométrico.

Esta rotina funciona de uma forma bastante simples: primeiro calculamos o número de vértices, como fizemos para o caso anterior, de “Polylines na Malha”. Se a polyline tiver 4 vértices, então o centro geométrico é adequado já que o “offset” será sempre para dentro. Entretanto, se forem mais de 4 vértices, existe a possibilidade de que o centro geométrico seja externo à polyline, neste caso o offset seria realizado para o lado errado. Para evitar este erro primeiro selecionamos os três primeiros pontos da lista de vértices da polyline original (p1, p2 e p3). Depois fazemos uma triangulação

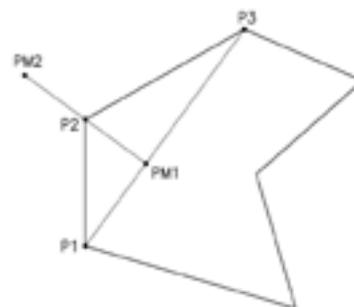


Figura 4.

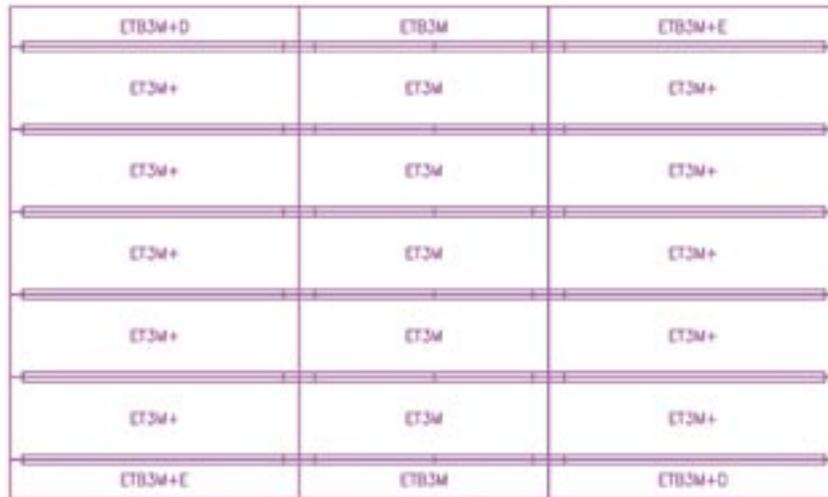


Figura 5.

de tal forma que obtemos um ponto na bissetriz (pm1) e outro a 180° deste (pm2), como vemos na Figura 04. Com isso podemos fazer um “offset” da polyline, usando o ponto pm1. Para saber se o “offset” foi realizado para dentro ou para fora comparamos o perímetro (através da variável “perimeter”) da polyline original com o da feita pelo “offset”. Se o primeiro for maior, então o “offset” foi feito corretamente. Caso contrário, apagamos a última polyline e realizamos um novo “offset” usando o ponto pm2. Esta técnica não é genérica, mas funciona corretamente em nosso caso já que não utilizamos formas muito recortadas.

A forma de projetar os elementos de teto guarda algumas semelhanças com o processo acima descrito. Entretanto, possui também algumas particularidades. A partir dos elementos básicos, utilizamos novamente a opção “Polyline na Malha”, que constrói polylines fechadas dentro da malha. De forma semelhante ao caso anterior, a opção “Elemento de Teto” realiza um “offset” para a inclusão da junta. Mas as semelhanças acabam por aqui: como pode ser observado na Figura 05, cada peça possui uma nomenclatura. Também há uma série de perfis desenhados.

No caso dos perfis, o programa calcula as dimensões de todos eles: os inferiores são colocados entre as vigas, e já é descontada a distância necessária para os perfis das luminárias. No caso dos perfis superiores,

estes são colocados sequencialmente em função do comprimento padrão de um perfil, enquanto o último é cortado. Todos estes perfis são blocos com atributos, que passam a conter as principais dimensões.

Com relação à codificação de cada elemento de teto, é uma indicação do tipo de elemento em função das formas de encaixe existentes nas vigas e elementos laterais. Como mostra a Figura 05, algumas peças são únicas, como as dos cantos. Não é possível trocar estas peças de posição, já que não encaixariam, portanto é fundamental diferenciá-las para que não exista a possibilidade de que sejam enviadas para a obra peças erradas.

Esta foi uma das rotinas mais complexas, foi adaptada em parte de um programa destinado à montagem de pisos elevados, e funciona da seguinte forma: a primeira parte consiste em montar uma lista com todos os pontos da malha, resultado do cruzamento das linhas da mesma, utilizando:

```
(setq pinterseção (inters pi1 pfl pi2 pf2))
```

Na seqüência, é montada uma lista com os pontos finais, utilizando:

```
(setq pi1 (cdr (assoc 10 linha))) e  
(setq pfl (cdr (assoc 11 linha)))
```

Por último é montada uma lista com os vértices da polyline de contorno. Estas três listas definem três tipos de pontos: os de interseção, os finais e os de contorno. Assim, é possível atribuir a cada placa um código de quatro letras, o que irá definir a sua codificação, de acordo com o tipo de encaixe que será necessário: na parede, em outra placa ou na viga. Se a placa tiver mais de 4 vértices, será especial e receberá um código próprio. Evidentemente, embora a descrição que fizemos seja sumária, a rotina ocupa muitas páginas, e seria impossível aqui mostrá-la por inteiro.

## 2.5. Extração de lista

A última parte do programa é a extração da lista. São os seguintes os elementos que fazem parte da mesma: Elementos de fundo e teto, inclusive os especiais, e cantos positivos e negativos exclusivamente especiais. Todos são polylines com dados extendidos (xdata), e são selecionados da seguinte forma:

```
(setq sel (ssget "W" p1 p2 '((0 . "LWPOLYLINE") (-3 ("CDG")))))
```

No caso dos elementos de fundo e teto, são separados os que têm 4 vértices, cujas medidas serão extraídas. Os demais terão a descrição "especial". Em função das medidas padrão, são atribuídos os códigos. Estes códigos mais as descrições completas de cada elemento são gravados em um arquivo texto:

```
(setq arquivo (open txt "a")) ;Abro arquivo
(opção adicionar).
```

```
(setq descrição (strcat descrição "\n"))
(princ descrição arquivo) ;Gravo a
linha.
```

```
(close arquivo)
```

Há uma série grande de elementos que são blocos com atributos. Neste caso usamos uma função do tipo:

```
(setq xxxx (cdr (assoc 1 (entget (entnext (ssname conj t))))))
```

Estão incluídos nesta categoria os seguintes elementos, que são gravados no mesmo arquivo texto acima citado, já com codificação e descrição: portas, perfis e soleiras das portas; vigas; pilares; perfis de elementos laterais, perfis inferiores e superiores dos elementos de teto; trilhos, perfis e emendas das luminárias; perfis e emendas externas do teto; perfis e emendas das vigas.

Além disso, em função das áreas de contato entre os diversos elementos, obtidos quando da contagem dos mesmos, é possível calcular todas as fitas de isolantes. São calculados em metros lineares e depois convertidos para quantidade de bobinas por aproximação. Também é obtida a quantidade de tubos de silicone.

Finalmente usamos a função "startapp" para abrir o Microsoft Access, que realiza as operações necessárias e extrai uma listagem completa com todos os produtos, seus códigos, quantidades e descrição (fundamental para o caso das peças especiais).

A utilização desta forma de procedimento de extração de lista, embora pareça pouco sofisticada, é uma herança dos tempos do AutoCAD 14, e como sempre funcionou sem qualquer problema acabou não sendo atualizada até o momento.

## 3. Conclusão

Com este programa procuramos facilitar o processo de projeto, evitando erros e ganhando tempo, obtemos uma listagem de todas as peças com descrições detalhadas, as quais permitem conhecer de antemão todas as peças que serão necessárias bem como suas dimensões.



**Ricardo Hernán Medrano**

*Pesquisador - Universidade Presbiteriana Mackenzie (Brasil) - Faculdade de Arquitetura e Urbanismo*

*Bacharel em Física (Universidade de Campinas, 1986).*

*Arquiteto e Urbanista (Universidade de São Paulo, 1991).*

*Doutor em Estruturas Ambientais Urbanas (Universidade de São Paulo, 2003).*

*Áreas de interesse: História e Projeto de Arquitetura e Urbanismo; Desenho Industrial*