PROGRAMA LIVRE PARA ANÁLISE DE GRELHAS DE CONCRETO

Andrew Cass Junior*
Roberto Chust Carvalho**

RESUMO

Apresenta-se neste artigo uma ferramenta com pré processador gráfico que permite a análise de grelhas por barras, visando seu emprego no cálculo de estruturas de pavimentos de concreto. Embora programas desse tipo existam desde a década de setenta, não existia – até o momento – um com as características de "*Software* Livre" e feito especificamente para as estruturas de concreto armado. O programa foi desenvolvido usando um sistema "RAD" (*Rapid Application Development*), neste caso, o "Lazarus *Free* Pascal" que é um *software* livre e usando as técnicas de programação orientada a objetos (OOP). Técnicas, que associadas ao caráter livre do programa, permitirão que diferentes usuários programadores possam contribuir para a implementação de melhorias no mesmo. O pré-processador gráfico, totalmente desenvolvido para esse fim, permite a entrada e saída de dados do programa de forma gráfica, evitando-se assim, o uso de interfaces comerciais como os CAD's atuais. Descreve-se sucintamente tanto o módulo de cálculo como o módulo gráfico.

Palavras-chaves: Programa livre, grelha, concreto, análise

1 INTRODUÇÃO

A determinação dos esforços solicitantes e os deslocamentos em diversos pontos de uma estrutura de concreto armado ou protendido foi até pouco tempo a parte mais trabalhosa do projeto. Hoje, na prática, tem-se usado programas computacionais bastante sofisticados com entradas e saídas de dados gráficas que permitem rapidamente vencer estas etapas. Na verdade os programas além da análise fazem a determinação das armaduras e as detalham nos diversos elementos estruturais. No Brasil, especificamente, tem-se usado o modelo de grelha equivalente para a análise dos pavimentos e conectando-o a um pórtico tridimensional para análise de ações laterais (de vento em geral).

* Mestrando do Programa de Pós-Graduação em Engenharia Civil da Universidade Federal de São Carlos
** Professor Doutor do Programa de Pós-Graduação em Engenharia Civil da Universidade Federal de São Carlos
Deciv, UFSCar, Campus São Carlos Rodovia Washington Luís, km 235 - SP-310

O ensino de engenharia ainda não chegou neste ponto. Ainda faz-se a análise de estruturas sem o auxílio de programas individualizando-a em seus elementos, como sendo únicos. Algumas escolas introduziram o ensino de "Análise Matricial", mas na maioria delas a disciplina acaba sendo oferecida como optativa.

Os autores Gere e Weaver Jr citam que havia um curso de análise matricial de estruturas reticulados para alunos de graduação e pós-graduação em 1965 nos Estados Unidos. Ao se pesquisar na internet encontram-se somente programas gratuitos de pórticos planos.

Desta forma, o grupo de pesquisa GESC do qual os autores fazem parte, iniciou há algum tempo, a confecção de programas que além de serem gratuitos, são livres. A característica do programa livre é permitir acesso a listagem e desta forma criar modificações e acrescentar novas bibliotecas ao mesmo. O conjunto de programas usados para tal fim foi batizado de CALCO (cálculo de concreto armado). Para maiores detalhes acessar:

(http://www.deciv.ufscar.br/calco/CALCO1f.htm).

O sistema básico do calco é dividido em entrada de dados, cálculo das ações e finalmente, saída de dados. Os módulos de entrada e saída de dados fazem parte de um único procedimento gráfico que usa a técnica de programação orientada aos objetos. Neste trabalho apresenta-se o desenvolvimento, principalmente da parte gráfica de um programa de grelha, mostrando apenas de forma resumida a parte de análise matricial.

2 METODOLOGIAS

2.1 Módulos de Análise Matricial

A parte de análise matricial, ou seja, a resolução do pavimento como uma grelha equivalente foi feita ainda de maneira clássica com programação estruturada e seguindo os conceitos do método dos deslocamentos. Usou-se também, a sistemática apresentada por Gere e Weaver (1981) por ser clássica e a mais conhecida no Brasil.

Para a determinação dos esforços solicitantes de uma estrutura formada por barras o esquema seguido foi basicamente o indicado no esquema da Figura 1. A versão inicial foi feita por COTTA (2003. O programa sofreu uma otimização em 2011 com Anjoletto Filho (2011), a técnica empregada ainda foi a da programação estruturada baseada nos fluxogramas de Gere e Weaver (1985). Foram implementados outros tipos de carregamentos, a colocação de rótulas, variação de temperatura etc. Além destes componentes básicos, foram ainda desenvolvidos outros como programa de pórtico tridimensional e cálculo de armadura.

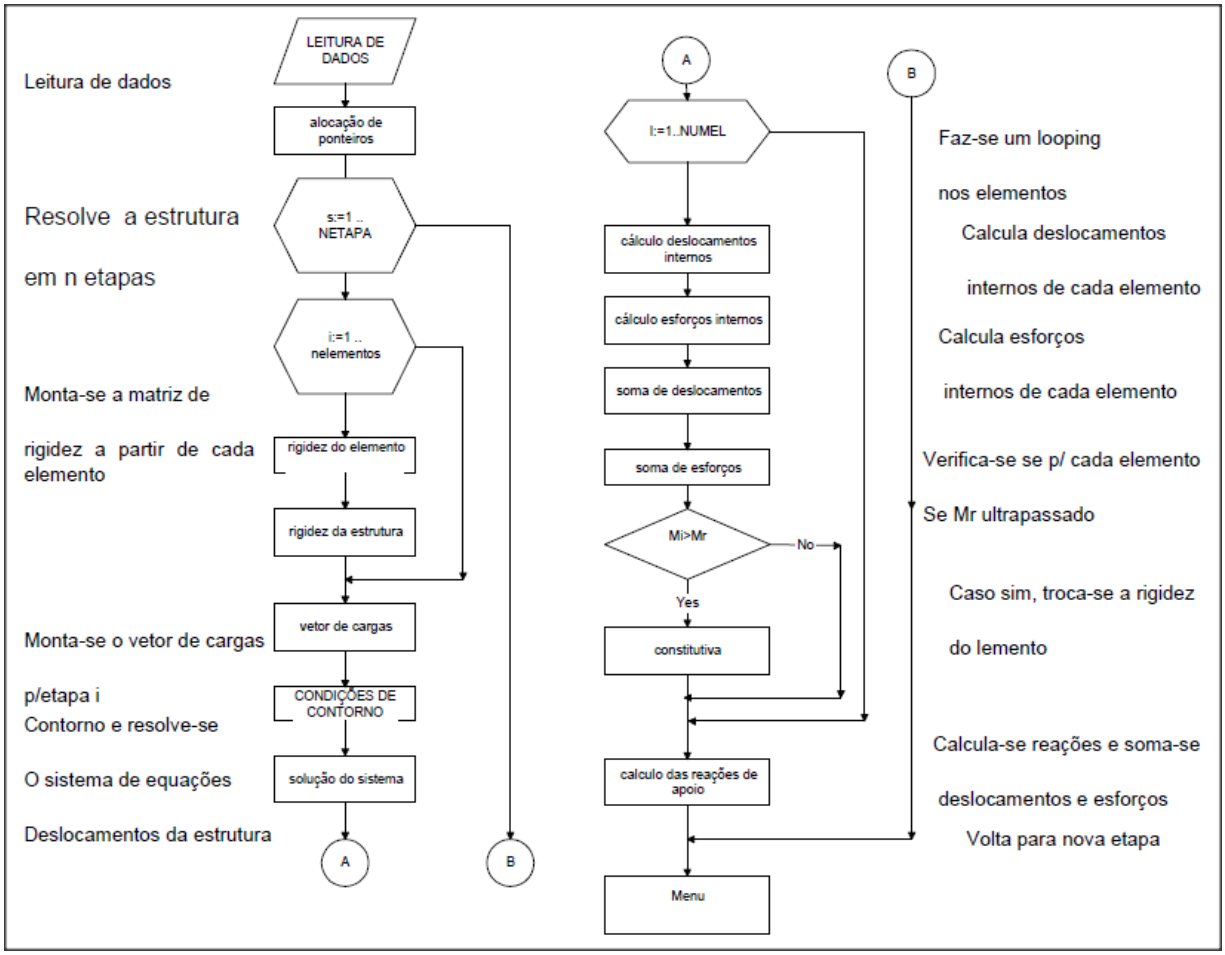

Figura 1: Fluxograma básico do programa de grelhas

2.1 Ambientes Gráficos

O ambiente gráfico foi desenhado tomando-se como base os programas que, até o momento, eram desenvolvidos em Lisp para AutoCad pelo grupo de trabalho orientado pelo prof. CHUST, R. UFSCar e que serviam como pré-processador gráfico para o programa de grelhas. As vantagens e desvantagens do uso de ambientes comerciais como os da Autodesk e similares para a programação pode certamente ser assunto de outro artigo, devido à vasta dicotomia envolvendo esses sistemas.

A escolha da linguagem Lazarus *Free* Pascal como plataforma de desenvolvimento deve-se ao fato dela ser muito comum nos meios acadêmicos e de ter uma curva de aprendizado bastante acentuada. Esta linguagem vem se beneficiando de atualizações constantes, mantidas por uma grande comunidade acadêmica internacional, que a transformou, juntamente com o C, numa das linguagens mais populares de desenvolvimento. Hoje ela possui várias qualidades, como por exemplo, rapidez de processamento, compilação multiplataforma (Windows, Linux, Apple, Windows CE, Android), programação estruturada, programação orientada a objetos e eventos, além do benefício de ser livre e gratuita. O Free-Pascal incorporado a IDE do Lazarus

permite ainda que sejam desenvolvidos componentes reutilizáveis nos moldes dos paradigmas modernos de desenvolvimento de software, que uma vez instalados no sistema podem ser facilmente acessados e reutilizados para o desenvolvimento de outros sistemas.

Como veremos mais adiante, a POO (Programação Orientada a Objeto) é fundamental para este projeto, pois permite que sejam criadas classes de objetos que seguem os princípios de Herança e Polimorfismo. Essas propriedades nos permitirão trabalhar de forma mais estruturada os objetos visuais, carregando-os com métodos e propriedades adaptados para cada tipo de primitiva gráfica e estrutural.

3 DEFINIÇÕES DE POO

Um Objeto em programação é uma entidade independente que tem um propósito bem definido, a capacidade de receber e enviar informações. O objeto é estruturado por métodos e propriedades. As propriedades são os valores das variáveis internas ao objeto (internas ao escopo do mesmo), enquanto que os métodos são os procedimentos e funções internas a esse objeto.

O programa cliente invoca uma função ou procedimento do objeto servidor e este devolve ao cliente uma resposta. O objeto tem encapsulado em seu núcleo todos os métodos e propriedades necessários para a realização de uma dada tarefa. O sistema cliente nunca terá que acessar diretamente o conteúdo deste objeto, somente os comandos e propriedades que fizerem parte do seu escopo público (entende-se pelo termo público os procedimentos e propriedades que podem ser acessados pelo sistema cliente ou periférico ou interface).

Além disso, temos o conceito de herança que permite a especialização da informação ao longo de uma linhagem hierárquica de classes, onde aquelas situadas em níveis mais baixos herdam características (propriedades e métodos) daquelas situadas acima. Assim é possível estender, ou reutilizar classes em várias aplicações, bastando programar as variáveis e métodos necessários para descrever a nova especialização desejada.

Este paradigma é de fundamental importância quando se pretende desenvolver sistemas de forma colaborativa.

No meio acadêmico podemos encontrar exemplos de adaptação bem sucedida da POO na programação de sistemas de cálculo estrutural por elementos finitos (DEVLOO, 1991; DRUMOND,2010).

4 DESENVOLVIMENTO DO PRÉ-PROCESSADOR GRÁFICO

No caso do programa de grelhas optamos por uma estrutura complexa, onde a CAD API (*Computer Aided Design Application Program Interface*) incorporou vários objetos diferentes, assumindo uma amplitude com o seguinte conjunto de propriedades ou "*Framework*":

1- **Capacidade de edição** e/ou criação de um projeto gráfico, permitindo a geração de novas primitivas gráficas como linhas, círculos, retângulos, elipses, etc.
2- **Capacidade de personalizar** as entidades gráficas com informações específicas;
3- **Capacidade de organizar** estas primitivas em camadas (*Layers*) e Blocos.
4- **Capacidade de edição dessas primitivas** como copiar, mover, apagar e etc.
5- **Capacidade de visualização** interativa na tela com funções de *zoom*, pan e etc.
6- **Capacidade de incorporar** ferramentas de auxílio à precisão ao desenho, como o "*snapping*", gerenciamento e configuração de grades e modo ortométrico.
7- **Capacidade de gravação** e leitura de arquivos num formato nativo ao sistema principal que preserve todas as propriedades das entidades parametrizadas no projeto.
8- **Capacidade de exportação** do projeto em formatos Inter-Cad-Applications, como formato DXF (*Drawing Exchange Format*) ou SVG (*Scalable Vectorial Graphics*).

Mas para que esta ferramenta tivesse uma utilidade além do simples desenho vetorial, foi necessário expandir a capacidade das primitivas, ou objetos gráficos, de forma que pudessem interpretar dados externos à própria API gráfica. Explicando melhor, reservou-se espaço de memória em cada objeto para que fossem conectados a ele propriedades ou métodos definidos pelo programador do sistema cliente. De fato, podemos oferecer um exemplo claro, um programa de cálculo estrutural qualquer deve permitir que as entidades gráficas como as barras e nós carreguem dados extras, como por exemplo: Propriedades geométricas da seção, propriedades do material, coordenadas globais e locais do sistema e implementações de cálculo diversas.

Para tanto, criamos um modelo como mostrado no diagrama abaixo na Figura 2.

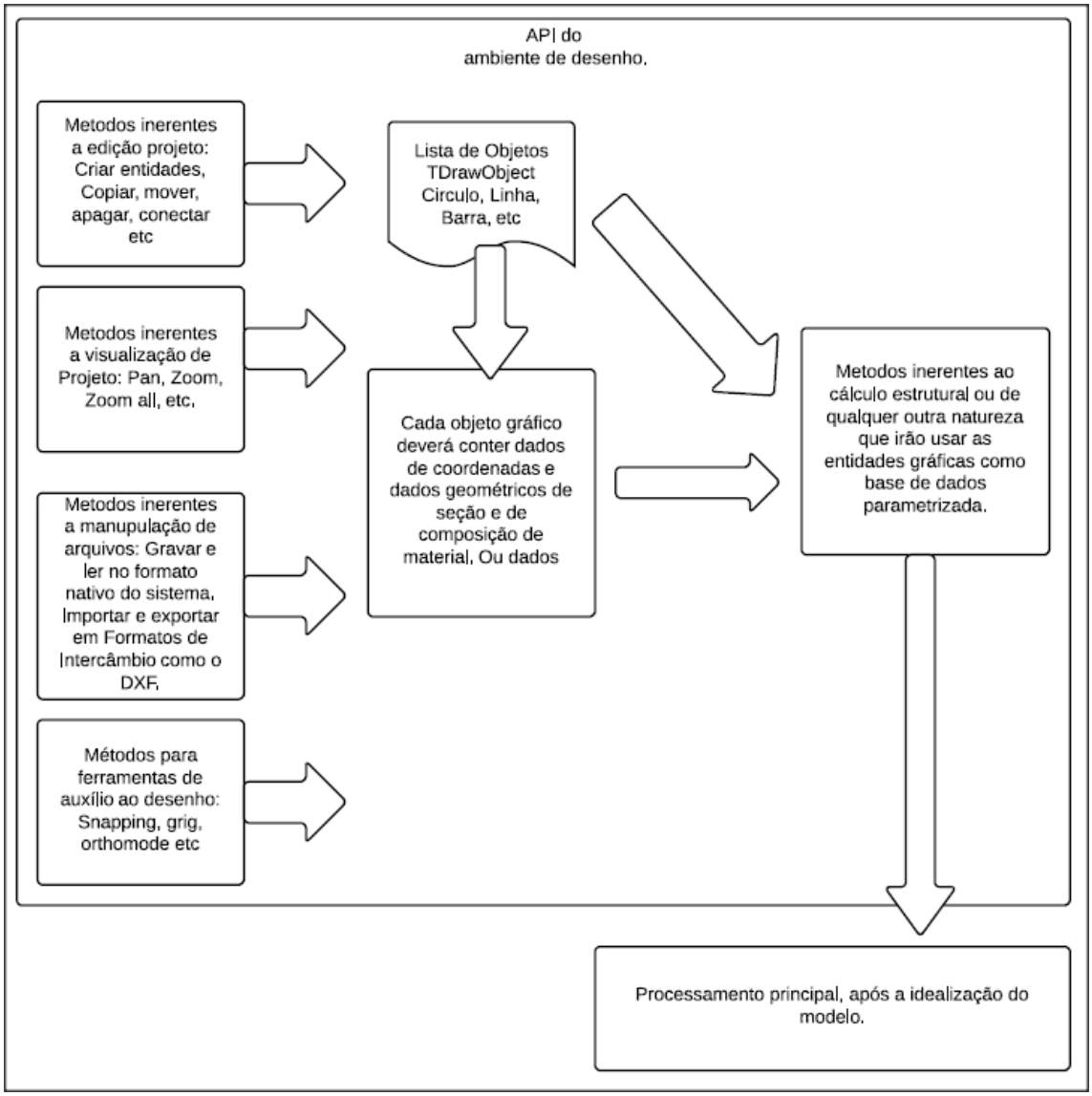

Figura 2: Modelo do sistema adotado pelo programa Grelhas.

Como sugerido por esse modelo (que foi criado antes do desenvolvimento da interface gráfica atual), criamos primeiramente um ambiente gráfico operacional com objetos de edição de entidades gráficas, visualização de tela, leitura e gravação de dados e auxílio à precisão do desenho.

Por comodidade derivamos o ambiente gráfico de uma Classe pré-existente tanto no Lazarus como no Delphi cujo o nome é TPaintBox, que é um componente visual em ambas as IDE's (*Integrated Development Environment*) e que encapsula uma grande quantidade de métodos e propriedades que foram muito úteis, como por exemplo, a classe TCanvas utilizada pelo sistema na composição de imagens na tela.

A classe descendente foi batizada de TCADBox, que além de herdar todas as propriedades e métodos da classe anterior (TPaintBox) recebeu uma ampla implementação de novos métodos e propriedades, alguns, já enumerados anteriormente.

5 PRÉ-CONDIÇÕES GEOMÉTRICAS

Obviamente, o simples desenho de entidade gráficas na tela não é suficiente para implantação de um sistema vetorial. O ambiente deve preservar as coordenadas originais de cada entidade, ou seja, mesmo quando há mudança de posicionamento (o que acontece quando se dá um zoom ou pan de imagem) os objetos gráficos devem manter suas coordenadas iniciais de projeto registradas. O que muda é a "*ViewPort*", ou seja as coordenadas na tela do computador.

Quando é executada a função "Pan" ocorre uma soma vetorial de dois pontos. Todo o desenho é transladado de (x1,y1) para o ponto(x2,y2) = (x1,y1) + (dx,dy). Em termos de coordenada de tela, todo o desenho sofreu uma mudança de base. Para que isso não afete os dados originais dos objetos de desenho a origem do sistema de coordenadas também deve ser transladada na mesma intensidade e direção. Dessa forma, as coordenadas dos objetos relativas aos eixos de origem permanecerão as mesmas. De forma análoga a função zoom que é o produto escalar entre as coordenadas e um escalar n, dilata as coordenadas num fator de n, ficando a coordenada original (x1,y1) transferida para (x2,y2)=(x1*n,y1*n).

A preservação das coordenadas originais foi uma condição imprescindível na parametrização dos elementos estruturais, já que delas dependeram a geração das coordenadas nodais da estrutura e as malhas resultantes.

A geração da imagem da estrutura na tela teve de ser tratada de forma virtual, ié, todos os elementos estruturais sempre preservam suas coordenadas originais, somente são geradas coordenadas virtuais para a geração da imagem na tela que é função do sistema de coordenadas, da base e da escala adotada.

A manipulação da tela implica em multiplicações e divisões sucessivas o que pode provocar um erro de ponto flutuante em cada conversão de coordenadas, principalmente quando se trabalha com escalas muito grandes ou com produtos que estão muito próximos de zero.

Os erros mais comuns são os de:

1. Proliferação: operações geométricas com números de K-Dígitos de precisão que podem gerar resultados que necessitem de precisão ainda maior;
2. Irracionalidade: algumas operações resultam em números sem precisão finita, como por exemplo, sqrt(3), PI/3 e etc.

Segundo Hoffman (2001), os *softwares* de cálculo geométrico, ocasionalmente, tendem a ser frágeis e falhos. Este problema de robustez se deve essencialmente a dificuldade em se tomar decisões inequívocas sobre a incidência e não incidência, fundamentalmente prejudicando a confiabilidade do sistema geométrico, o que é antagônico ao próprio princípio do sistema que gera um grande número de situações especiais e concretas.

No intuito de inibir esse tipo de erro no sistema adotamos escalas básicas condizentes com as necessidades do cálculo de estruturas de concreto, i.e., um pixel para cada milímetro (1 pixel = 1mm). Assim trabalhamos com objetos que na maior parte do tempo têm medidas inteiras. Por exemplo, 1.55m terão 1550 pixels que permanece um valor inteiro. Portanto a medida mínima inteira fica em 1 mm.

As coordenadas convertidas são mantidas em registradores separados (somente de leitura) a fim de evitar a propagação de imprecisões numéricas. As coordenadas originais são reservadas em registradores específicos e nunca são alterados pelas funções de gerenciamento de tela. Para diminuir os riscos de imprecisões pudemos ainda aplicar métodos de eliminação de incertezas (SAADE, J.J., 2000) e assim determinar quando há pequeno deslocamento de coordenadas ou quando estamos frente a uma imprecisão de ponto flutuante.

6 EXEMPLOS E RESULTADOS

Ao iniciar o programa Grelha é iniciado o processador gráfico como mostrado na Figura 3 acima, as primeiras variáveis do sistema devem ser introduzidas nesse momento define-se o fck do concreto, pesos específicos e módulos de elasticidade. Estes dados são então incorporados no arquivo principal de projeto.

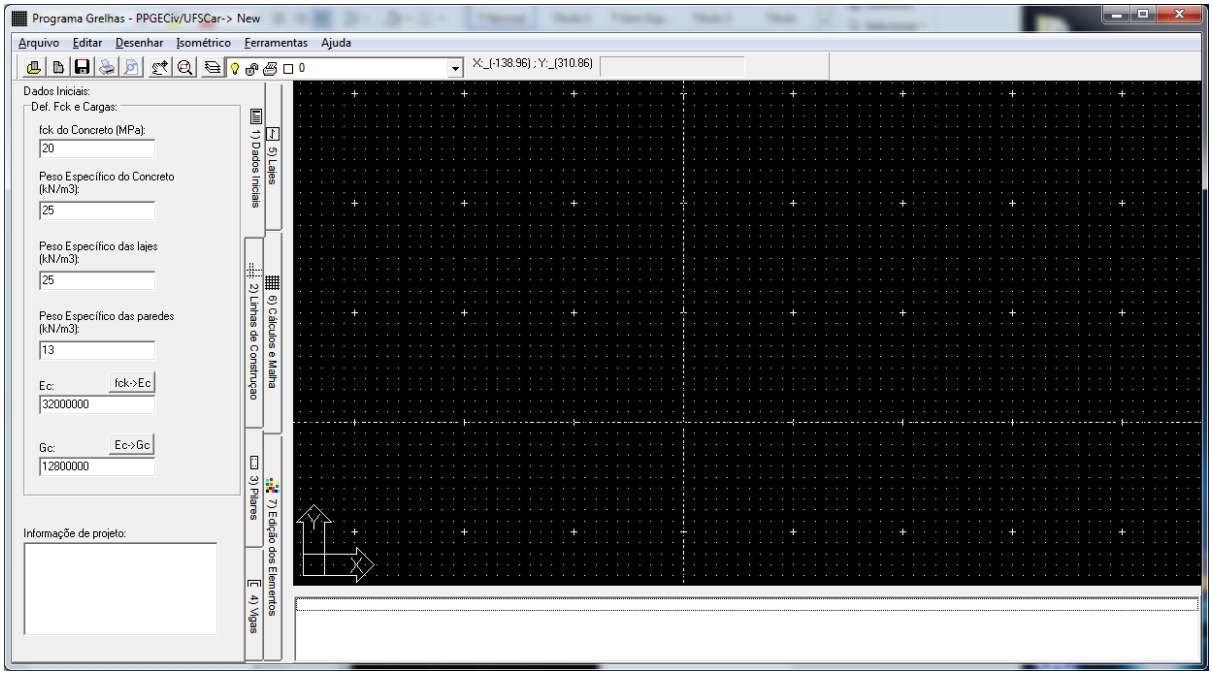

Figura 3: Tela inicial

Na segunda etapa, como mostrado na Figura 5, são introduzidos os eixos auxiliares (linhas de construção) que são utilizados no auxílio a montagem da estrutura a ser calculada.

Nas etapas seguintes são inseridos os pilares da estrutura baseando-se nas linhas de construção até que todos os pilares sejam inseridos. Os pilares podem ser inseridos em outro momento sem afetar o sistema de processamento. No momento da inserção dos pilares pode-se optar por um nome específico e pelas restrições de apoio nos pilares.

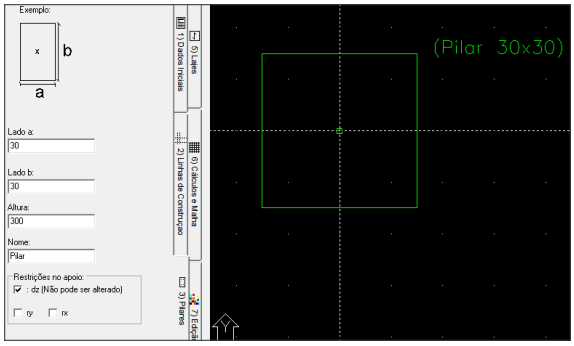

Figura 4: Representação de um pilar

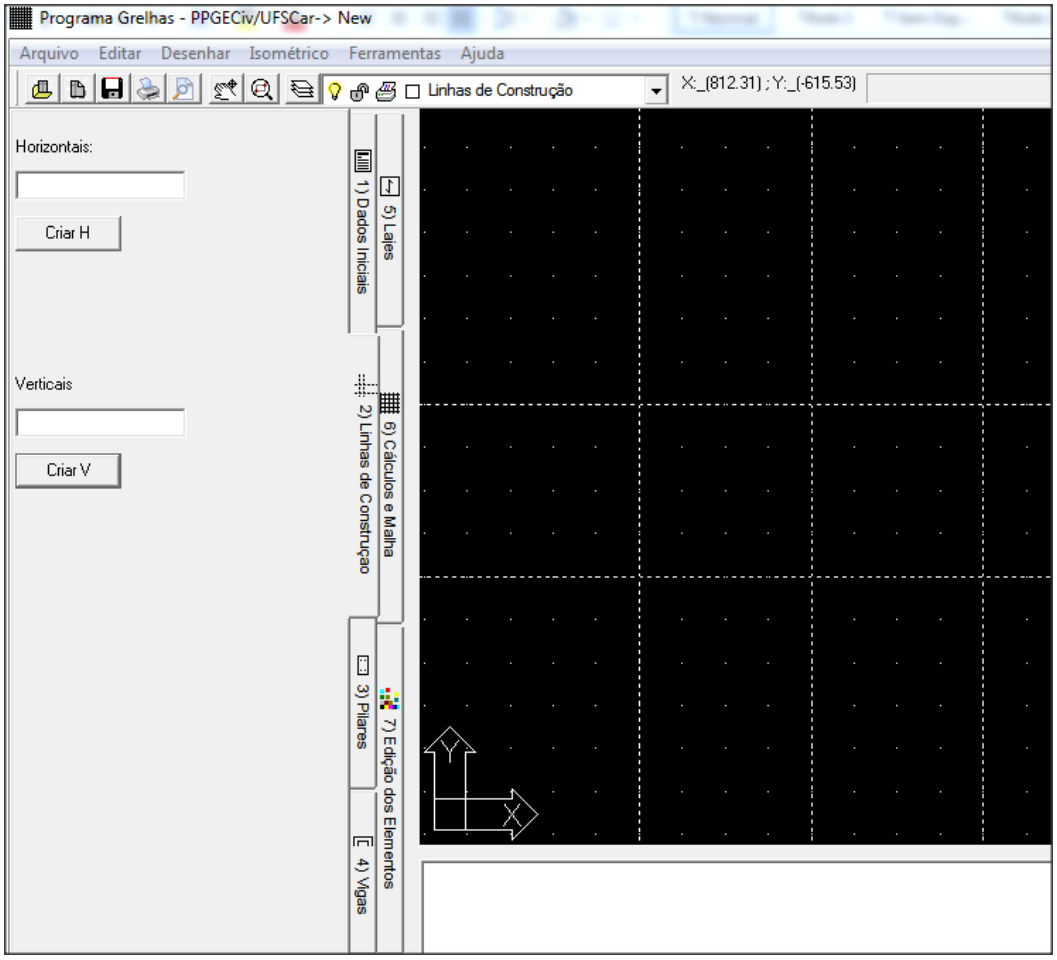

Figura 5: Geração de linhas de construção

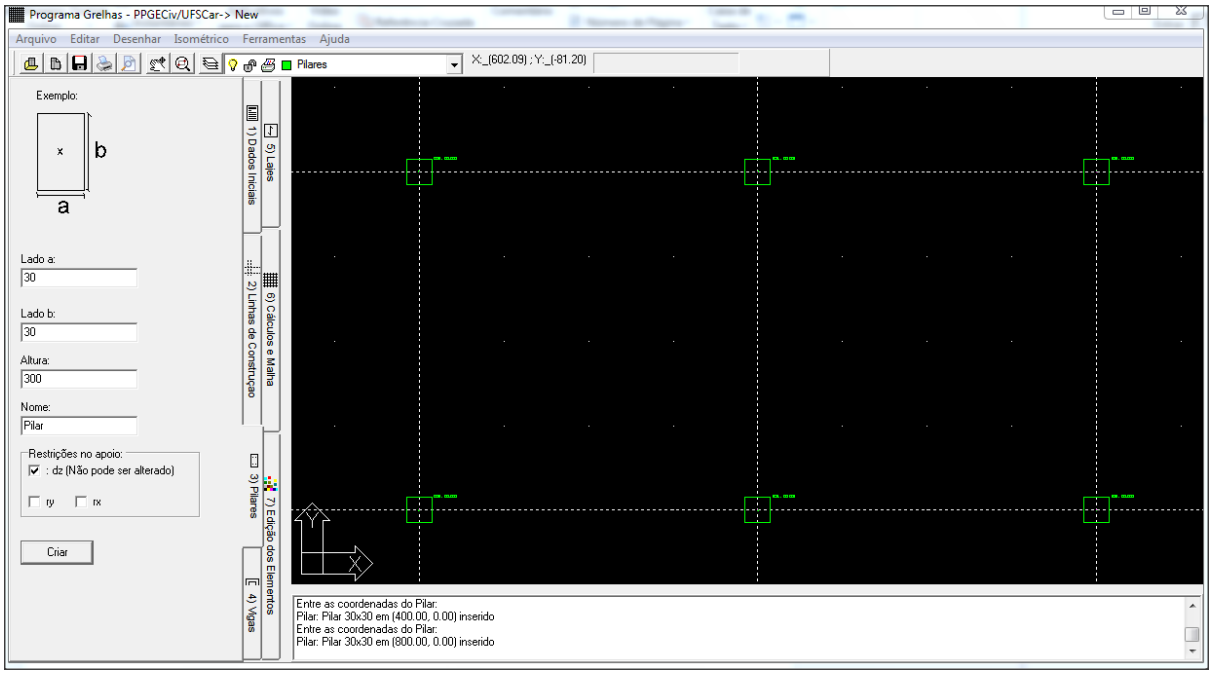

Figura 6: Pilares inseridos sobre as linhas de construção

Na quarta etapa inserimos as vigas e definimos as cargas sobre elas, como mostrado na Figura 7.

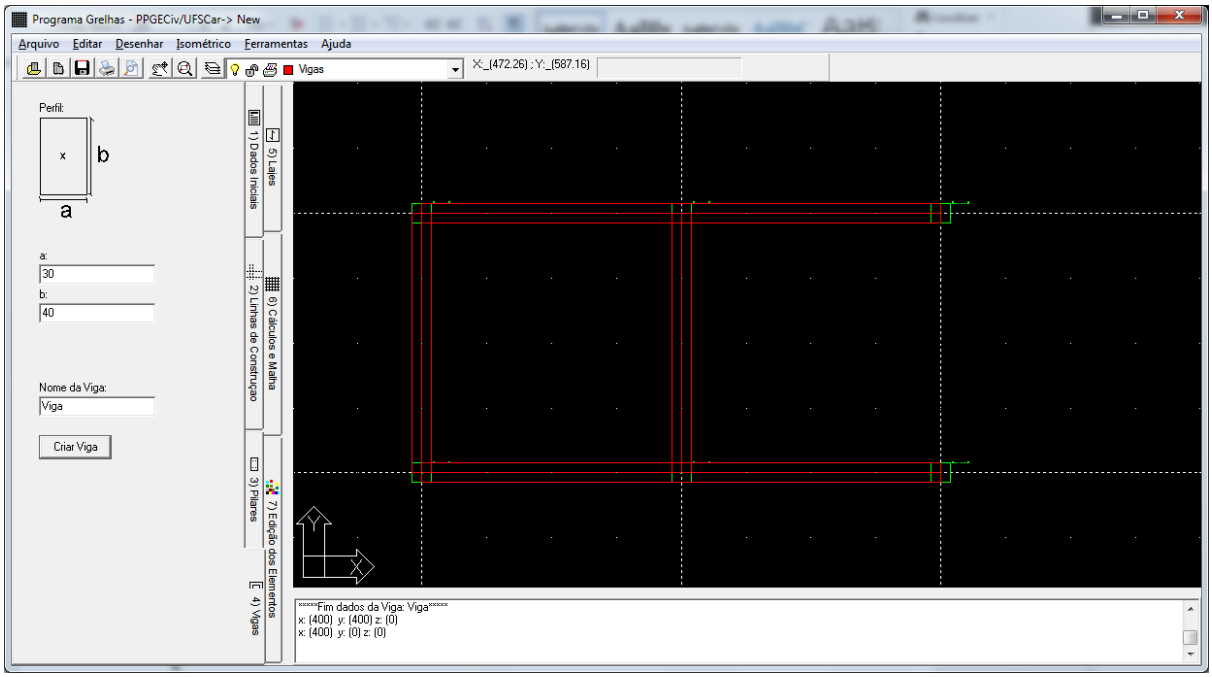

Figura 7: Vigas inseridas

As vigas são elementos não obrigatórios. Após inserir a vigas inserimos as lajes indicando os pontos da diagonal.

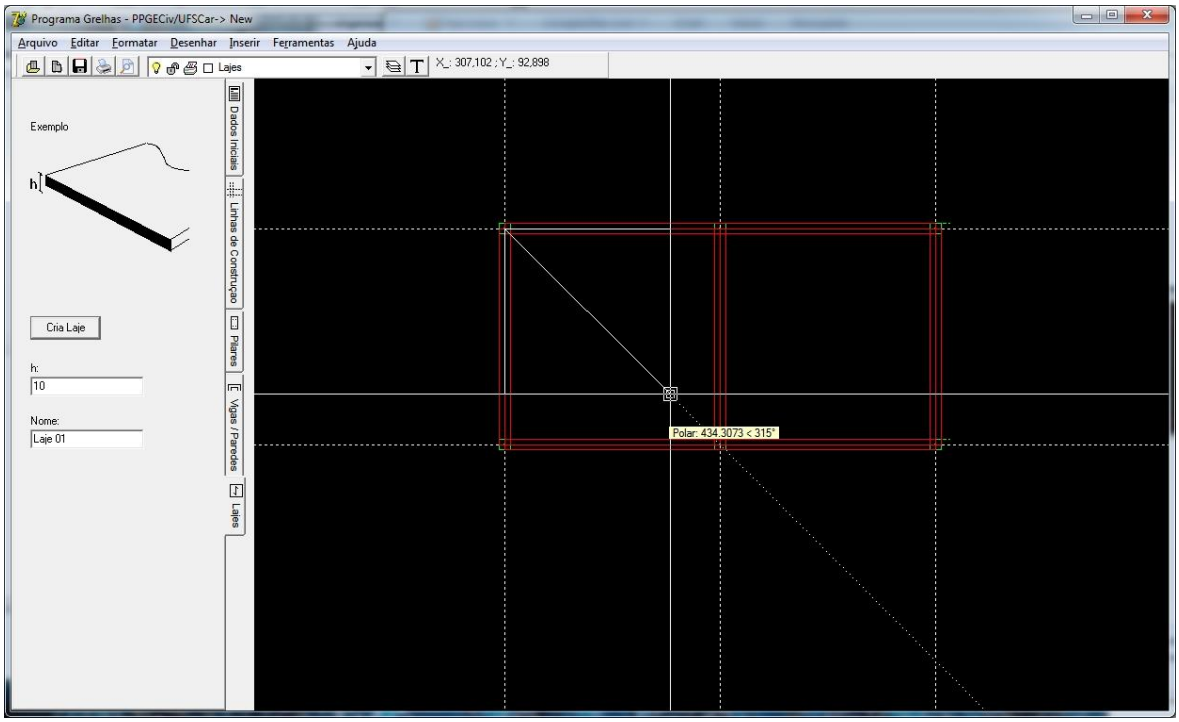

Figura 8: Inserção de lajes

As lajes podem ser inseridas em várias unidades ou uma única. Se as lajes tiverem uma mesma espessura é mais simples inserir uma única laje. Como pode-se ver na Figura 9 foram inseridas 2 lajes, uma de 12 cm e a outra de 20 cm.

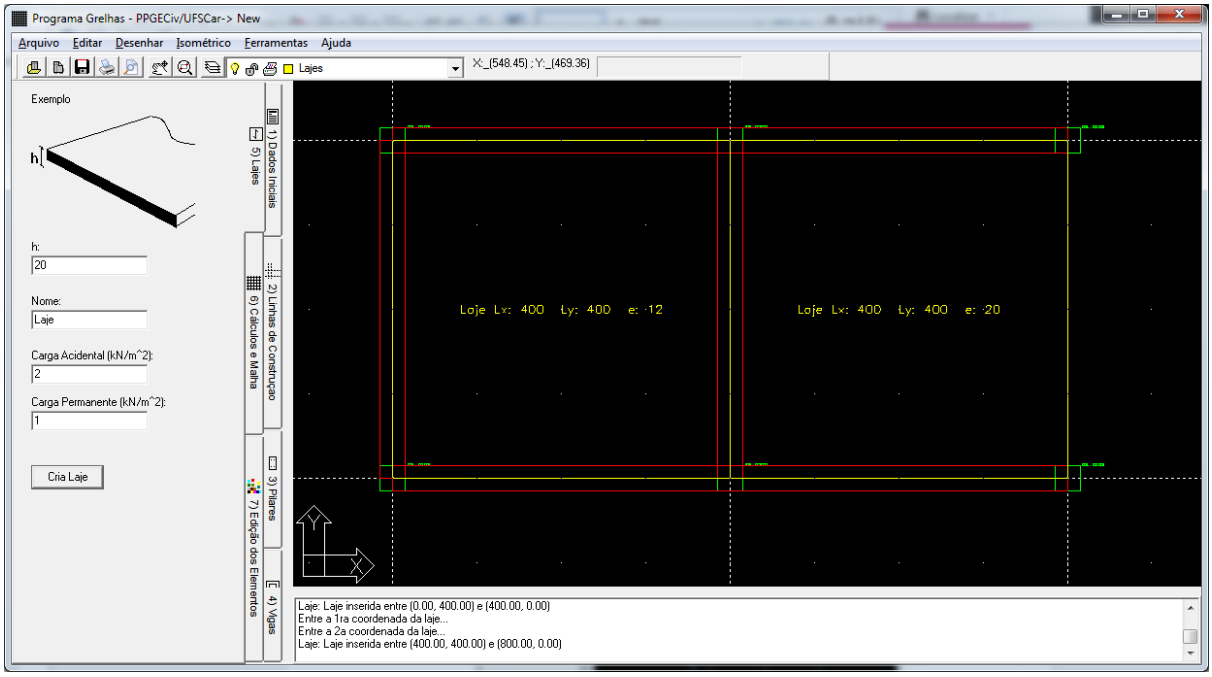

Figura 9: Lajes inseridas

Finalmente quando todos os elementos estruturais foram inseridos, pode-se criar a malha e calcular a grelha equivalente.

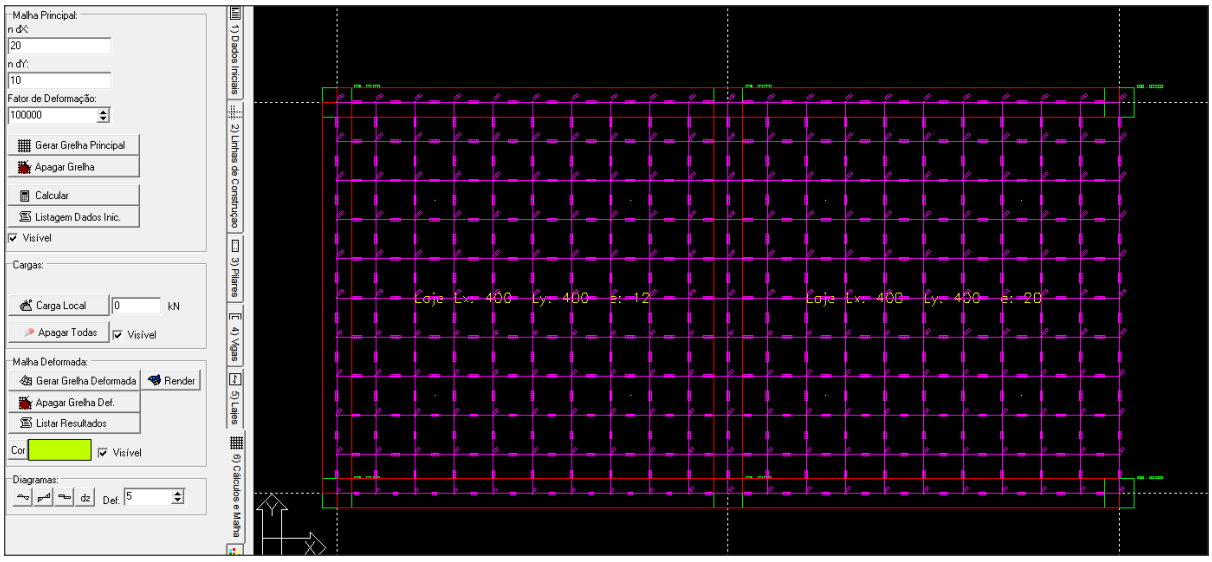

Figura 10: Malha da grelha equivalente gerada.

Na Figura 10 pode-se ver a malha equivalente gerada, no caso mencionado a malha tem 20 linhas verticais por 10 horizontais. Uma vez gerada a malha pode-se criar a malha deformada. A malha deformada é o resultado do processamento da grelha que nos dá informações à respeito dos deslocamentos nodais e as tensões atuantes nas barras. Pode-se ver no exemplo da Figura 11 a malha deformada em perspectiva isométrica.

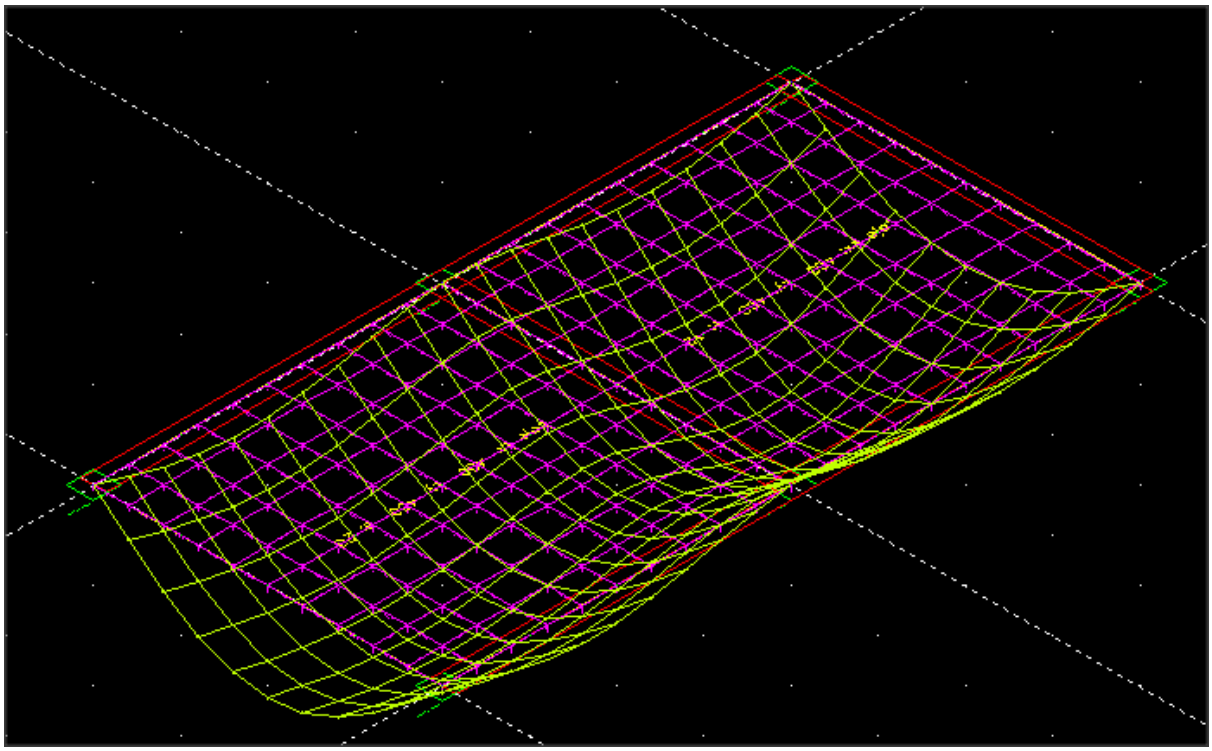

Figura 11: Malha de referência e Malha deformada.

Uma vez calculada a grelha pode-se gerar diagramas das solicitações e deformações de eixos específicos da grelha como pode-se ver na Figura 12.

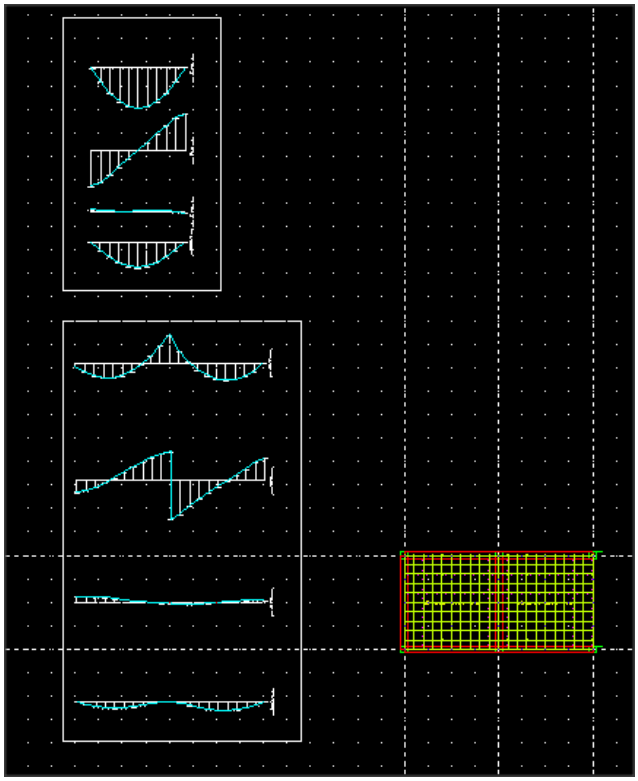

Figura 12: Diagramas de solicitações.

4 VALIDAÇÃO DO SISTEMA

No exemplo apresentado na Figura 13 tem-se uma conformação de pilares 9X9 onde se apoia uma laje de 800cm X 800cm com espessura de 10cm. A mesma grelha foi calculada pelo GPLAN e o sistema aqui apresentado.

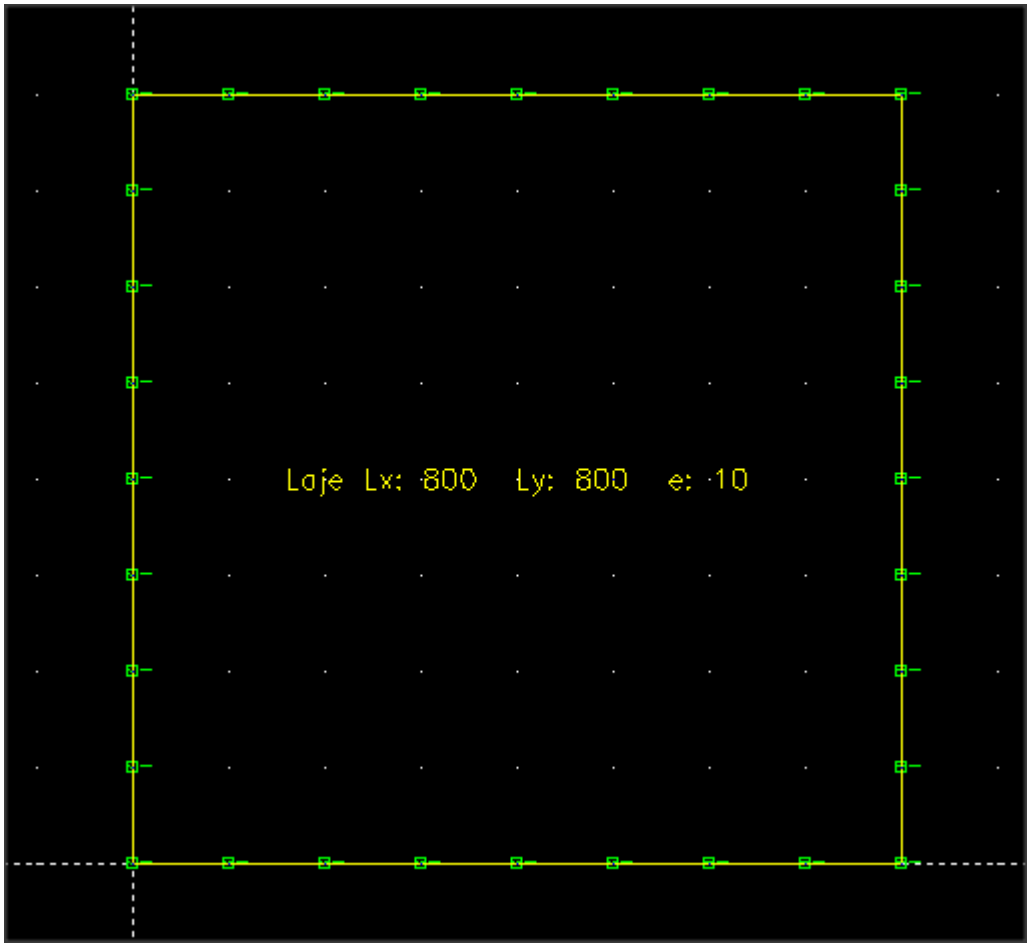

Figura 13: Projeto exemplo

Esta estrutura resultou na grelha deformada mostrada na Figura 14

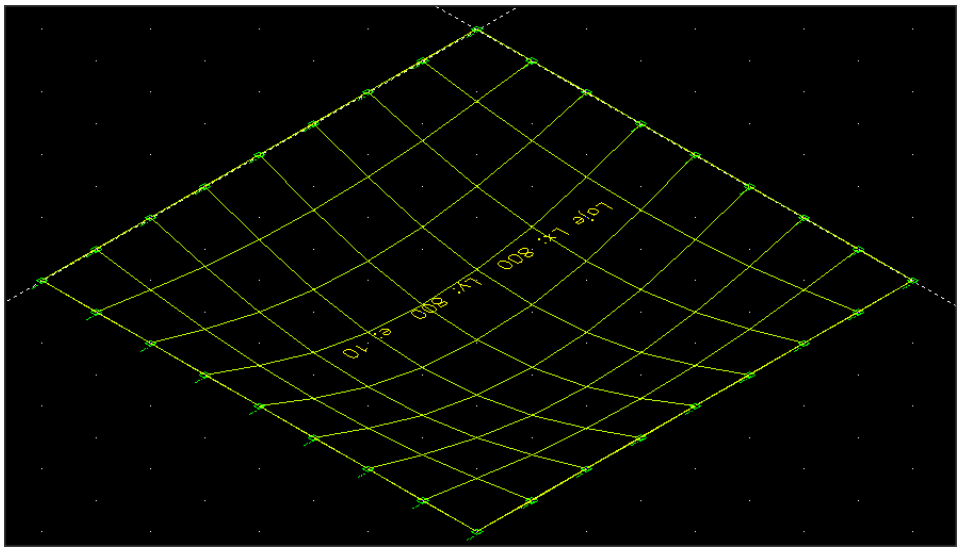

Figura 14: Grelha exemplo calculada.

Para o nó 41, que é o nó central, tem-se os seguintes valores que compara-se com os resultados obtidos (para a mesma estrutura e mesmos nó) pelo GPLAN:

	Programa Grelha	GPLAN
Nó	41	41
Deslocamento	-0.0474100 m	-0,0474 m
Momento Fletor	15.0640 KN.m	15,06 kN.m

Conclui-se portanto que o sistema produz resultados 100% compatíveis com o GPLAN.

FREE PROGRAM FOR CONCRETE GRIDS ANALYSIS

Abstract

This paper presents a software with a graphical pre-processor which allows the calculation of grids by bars for being used on concrete pavements structural analysis. In spite of the existence of such programs since the seventies, the previous ones were not "Free Software" and they were done specifically for reinforced concrete structures. The system was developed by using a "RAD" (Rapid Application Development) tool, in that case, "Lazarus Free Pascal" which is a free software that uses the object oriented programming (OOP). Techniques that adopt open license software will allow different users or programmers to contribute on the improvement of the graphic pre-processor which is fully developed for that purpose by graphically allowing the data input and output avoiding the use of commercial interfaces such as CAD systems. In this paper, calculation and graphic modules were concisely described.

Keywords: Free software, grid, concrete, analysis

REFERÊNCIAS

A. FILHO, J.S.R. DEVLOO, "**Object Oriented Programming in Scientific Computing, The Beginning of a New Era**" - Engineering Computations , 1991.

ALVES, M Z; **Processamento de geometria para geração automática de malhas.** São Paulo: USP, 1995;Simpósio autor sec: Massaroppi Junior, E; (EESC - ESCOLA DE ENGENHARIA DE SAO CARLOS).

ANJOLETTO Filho, M. C. **Otimização de um programa de grelha equivalente do sistema CALCO para resolução de pavimentos de Concreto Armado**. Iniciação Científica FAPESP São Carlos 2011

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS (2003). NBR 6118. **Projeto de Estruturas de Concreto – Procedimento.** Rio de Janeiro, 2003

BEER, F.P.; JOHNSTON,E.R. **Mecânica Vetorial para Engenheiros,** 2da ed.São Paulo: Makron Books, 1988. V.1 e 2.

CARVALHO, ROBERTO CHUST.; **Análise não-linear de pavimentos de edifícios de concreto através da analogia de grelha**; Tese de doutorado; SET-EESC-USP,- São Carlos;1994

CARVALHO, ROBERTO CHUST; FIGUEIREDO FILHO, JASSON ROGRIGUES DE**. Cálculo e detalhamento de estruturas usuais de concreto armado (Segundo a NBR 6118:2003)**; Brasil – São Carlos, SP. 2004. 2ª Edição. EdUFSCar

CARVALHO R. C., COTTA I F. S., RAYMUNDO R. **Sistema Calco**
http://www.deciv.ufscar.br/calco/CALCO1f.htm em 02/05/2014

CARVALHO, R. C. **Análise não-linear de pavimentos de edifícios de concreto através da analogia de grelha**. 1994. Tese de Doutorado. Programa de Pós-Graduação em Engenharia de Estruturas, Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos.

COTTA, I. F. S. **Desenvolvimento de programa livre automático para determinação de esforços solicitantes, deslocamentos e armadura de pavimentos em concreto armado usando a analogia de grelha não linear**. FAPESP 2004.

COTTA I F. S., **Desenvolvimento de programa livre automático para determinação de esforços solicitantes, deslocamentos e armadura de pavimentos em concreto armado usando a analogia de grelha não linear**. Iniciação Científica FAPESP, São Carlos 2003

COTTA I F. S., **Desenvolvimento de programa livre para análise de pórticos tridimensionais considerando-se a não linearidade geométrica, fissuração do concreto e ligações semirrígidas.** Dissertação de Mestrado PPGECIV UFSCar, São Carlos 2007

GERE, J. M.; WEAVER JR., W. **Análise de estruturas reticuladas,** Editora Guanabara. Rio de Janeiro 1981

MARTHA, L.F. **Métodos básicos de Análise de Estruturas.** Rio de Janeiro: Pontifícia Universidade Católica do Rio de Janeiro, 2000.

MOREIRA, D.F. **Análise Matricial das Estruturas.** Rio de Janeiro: EDUSP, 1977.

PARAHYBA, M C**; Utilização do turbo pascal na geração de desenhos de malhas** ,São Carlos : Cetepe, 1991; Congresso de Iniciação Cientifica e Tecnológica em Engenharia, 10 : 1991 : São Carlos; autor sec: Veiga, Jorge Pinheiro da Costa; (EP - ESCOLA POLITECNICA)

PETRY, ROBERTO PADILLA: **Um modelador geométrico uni, bi e tridimensional com aplicação para a geração de malhas de elementos finitos.** / Roberto Padilla Petry; [Orient]

RAYMUNDO H. **Programa para representação da forma de pavimentos de concreto, geração de dados correspondentes para programa de cálculo de estruturas prismáticas e representação da estrutura deformada e esforços solicitantes**. Iniciação Científica Fapesp São Carlos, 2007.

REZENDE, MARCELO NOVAES DE; **Geração automática de malhas de elementos finitos retangulares**; São Carlos : EESC, 1987; Congresso de Iniciação Científica e Tecnológica em Engenharia (6. : 1987 : São Carlos); TRABALHO DE EVENTO-RESUMO. Autor sec.: Paiva, João Batista de, 1952-; Venturini, Wilson Sergio;

REZENDE, SELMA MARIA,CAXAMBU : SBMAC/INPE, 1998: **Sistema gráfico e iterativo de geração de malhas;** autor sec: Teodoro, Alessandra; Fortuna, Armando de Oliveira; (ICMC - INST DE CIENCIAS MATEMATICAS E COMPUTACAO). TRABALHO DE EVENTO-RESUMO

SAADE, J.J., A **Defuzzification Based New Algorithm for the Design of Mamdani-Type Fuzzy Controllers**, Mathware & Soft Computing 7 159-173, 2000.

SILVA FILHO, A.M. **Introdução à Programação Orientada a Objeto com C++.** 1 ed. São Paulo: Campus 2010, 312p.

SORIANO, H.L., LIMA, S.S. **Método de Elementos Finitos em Análise de Estruturas.** UFRJ, 1999.

TEODORO, ALESSANDRA; **Geração automatizada de malhas; São Carlos: ICMC-USP, 1999; autor sec: Fortuna, Armando de Oliveira, orient;** Workshop de Teses e Dissertações Concluídas - ICMC/USP, 4. Anais São Carlos: ICMC-USP, 1999 (ICMC - INST DE CIENCIAS MATEMATICAS E DE COMPUTACAO). TRABALHO DE EVENTO