

Gestão de *Layout* com *Visual Basic for Applications* na Interface do *Excel*

Rafael Nagel (IFSul) rafael.gustavo.nagel@gmail.com
Abel Hames (IFSul) abelhammes9604@hotmail.com
Fabio Lorenzi da Silva (IFSul) fabiolorenzi@ifsul.edu.br
Elpidio Oscar Benitez Nara (Unisc) Elpidio@Unisc.br
Richard Silva Martins (IFSul/Unisc) richardmartins@ifsul.edu.br

Resumo:

No cenário de mercado atual, as organizações manufatureiras necessitam, cada vez mais, de tecnologias e inovações para continuarem competindo fortemente com os concorrentes. Nesse contexto, este estudo tem como objetivo a implementação de um algoritmo que visa a otimização de arranjos físicos celulares em um cenário industrial manufaturado. Para isso, foi utilizado a programação *Visual Basic* através da interface do *software Excel* do pacote *Office*.

Palavras Chave: *Layout, Visual Basic, Excel.*

Layout management with *Visual Basic for Applications* in *Excel* Interface

Abstract

In the current market scenario, manufacturing organizations need, increasingly, technology and innovation to continue competing strongly with competitors. In this context, this study aims to implement an algorithm that aims at optimizing cell physical arrangements in a manufactured industrial scenario. For this, the *Visual Basic* programming was used by the *Office Excel* software package interface.

Key-words: *Layout, Visual Basic, Excel.*

1. Introdução

Segundo a Confederação Nacional da Indústria (CNI), o setor industrial exerce considerável influência sobre a produtividade e a inovação dos demais setores econômicos, sendo fundamental para elevar o nível de consumo, emprego e exportações do Brasil. Uma indústria competitiva e diversificada é o caminho para o crescimento sustentado.

Dentro de um sistema produtivo, alguns fatores impactam diretamente nos resultados da manufatura. Dentre os quais pode-se destacar o fator humano, o nível de automação, o grau de organização e a forma como os recursos transformados fluem pelo sistema de fabricação. A maneira como fluem os produtos em transformação durante seu processamento está intimamente ligada à disposição dos recursos dentro do sistema produtivo. Essa disposição é identificada como arranjo físico ou *Layout*.

Objetivando garantir maior eficiência de produção, as organizações têm empregado atenção especial na busca por métodos para reduzir fluxos cruzados, custos logísticos e má utilização dos recursos produtivos (BATAGLIN et al, 2013). A eficiência do arranjo físico é um dos mais

importantes aspectos dos sistemas de manufatura contemporâneos (RAWABDEH E TAHBOUB, 2006).

Este artigo tem como objetivo apresentar a implementação do algoritmo *Close Neighbour* (CNA) no *Visual Basic for Applications* (VBA), linguagem de programação aplicada ao pacote *Office* da *Microsoft*, visto que o pacote *Office* é corriqueiro dentro da indústria e que a interface do *Excel* é difundida entre os profissionais que trabalham com gestão industrial.

A estrutura dessa pesquisa segue da seguinte forma: a seção 2 apresenta a Revisão da Literatura, com objetivo de subsidiar a parte prática do trabalho. Os temas apresentados nessa seção são *layout*, gestão de *layout* e programação VBA; a seção 3 exibe os Materiais e Métodos, nela apresenta-se a metodologia aplicada nesse trabalho, trazendo o passo a passo de como foi efetuado o estudo; a seção 4 traz a implementação, que mostra os resultados alcançados em uma aplicação do algoritmo; na seção 5 são exibidas as Conclusões, observadas de maneira breve e objetiva.

2. Revisão Bibliográfica

2.1 Layout

Em uma definição simplista arranjo físico de uma instalação industrial objetiva definir qual a posição dos recursos de produção dentro dos limites físicos do sistema produtivo. Recursos de produção contemplam máquinas, equipamentos e instalações de processo. Segundo Dalmas (2004), *layout* consiste em um arranjo físico onde ocorrem processos, contendo máquinas e equipamentos dispostos, que visa equilíbrio entre movimentação e produção. Portanto, o arranjo físico, ou *layout*, é responsável pela forma como os materiais, informações e pessoas se movimentam dentro do sistema de manufatura. Para Slack *et al* (2009), “o *layout* envolve o posicionamento relativo dos recursos transformadores dentro dos processos e a alocação de tarefas aos recursos, que juntos ditam o fluxo de recursos transformados ao longo do processo”.

Os *layouts* podem ser classificados em três tipos elementares: *Job Shop*, *Flow Shop* e Celular (FOGLIATO e NARA, 2013; NEUMANN, 2013). Davis *et al* (2001) e Slack *et al* (2009) relatam sobre os três tipos de arranjos físicos:

No *Layout* tipo *Job Shop* máquinas e equipamentos com funções similares são agrupados, criando-se seções de máquinas, na qual todas as furadeiras estão em uma área e todas as prensas estão em outra. A peça em trabalho passa de seção em seção, obedecendo ao fluxo especificado de operações de produção. Este tipo de arranjo físico é comumente encontrado em plantas de manufatura com baixo volume de produção. Seu fluxo de trabalho entre os recursos transformadores é complexo, pois ocorre ocupando as máquinas que estão disponíveis no momento.

O *Layout Flow shop* baseia-se no produto, ou seja, as instalações são organizadas observando as sequências de operações do produto. A linha é planejada levando-se em conta a sequência de processos necessários à fabricação de um produto, onde ocorre um alinhamento do processo para que a linha tenha um fluxo linear. É um arranjo com utilização em processos com grande volume e baixo *mix* de produto. Também, emprega automação pesada para tornar e manter o regime de produção constante.

A configuração Celular consiste em agrupar em centros de trabalho, máquinas distintas para produzir trabalhos em peças com características similares. Configura-se por ser similar ao *Job Shop* no sentido que as células são formadas para desempenhar um conjunto específico de processos, e é similar ao *Flow Shop* no sentido de que as células são dedicadas a uma gama limitada de produtos. Esse tipo de arranjo físico é fundamentalmente projetado para ser flexível do ponto de vista operacional bem como do fluxo de materiais. Em sua aplicação, os

trabalhadores envolvidos são multifuncionais dentro das células e emprega-se tecnologia de grupo (TG) para formação das células.

2.2. Gestão De *Layout*

Gestão de *Layout* tem a função de coordenar e determinar a alocação das máquinas, pessoas e processos de maneira a facilitar o fluxo de pessoas, aumentar a eficácia da mão de obra e equipamentos, melhorar os acessos, agilizar as etapas dos processos e melhorar a eficiência tanto de qualidade quanto de tempo dos processos e máquinas envolvidos no serviço (SLACK *et al*, 2009; DAVIS *et al*, 2001).

Um bom planejamento de arranjo físico ou uma gestão de *layout* eficaz requer muito trabalho e estudo sobre os espaços, máquinas, processos e etapas de determinada empresa, uma vez que é necessário realizar uma análise detalhada e complexa, com o intuito de organizar e facilitar a alocação das máquinas e processos. Ter um planejamento eficiente do arranjo físico resulta em máquinas e processos distribuídos adequadamente em determinados espaços, que consequentemente facilita a troca de materiais, assim como agiliza os processos e a troca de informações.

Visto tais ponderações, conclui-se que, inicialmente, a realização de um bom planejamento de arranjo físico gere custos e investimento de tempo, é de fundamental importância, pois tal planejamento é a garantia de eficiência, agilidade, segurança e qualidade na empresa.

2.3. Programação no *Excel* - *Vba*

A linguagem de programação *Visual Basic for Applications* (VBA) é uma linguagem orientada a objetos rica em funcionalidades e flexibilidade, que permite acrescentar capacidades adicionais e aplicações informáticas, concretamente as pertencentes ao *Microsoft Office*, entre as quais estão o *Excel* e o *Word*. Permite ainda automatizar a realização de muitas tarefas rotineiras nessas aplicações (SILVA, 2009). Com o *Excel* e o VBA, existe a possibilidade de criar macros para aperfeiçoar tarefas que são realizadas com frequência, fazendo assim com que várias ações sejam executadas em uma determinada ordem através de um único atalho do teclado (MICROSOFT, 2007).

Para acessar o editor do VBA, é necessário usar o atalho “Alt + F11”, abrindo, assim, o ambiente de programação no qual são desenvolvidos os códigos e a programação do algoritmo. Também, é nessa interface que o computador interpreta todos os comandos do programador, como por exemplo: “Dim” e “As”, que são utilizados na declaração das variáveis.

Os procedimentos do VBA são códigos reservados entre a palavra “Sub”, que define o início do código, e a palavra “End Sub”, que define o fim do código. É dentro desse trecho localizado entre as palavras “Sub” e “End Sub” que é escrito todo o código do algoritmo. Ainda existem duas coisas básicas e essenciais para iniciar a programação no VBA: os comandos de entrada e saída de dados. Enquanto para a entrada de dados é utilizado o comando “InputBox”, para a saída de dados é utilizada uma caixa de mensagem, chamada de “MsgBox”, onde é escrito entre aspas o texto que o usuário visualizará (SILVA, 2009).

Tal conjunto de comandos de programação serve para expandir os horizontes da planilha do *Excel* e desenvolver novas funcionalidades, além de poder ser um grande aliado na programação de sistemas direcionados ao planejamento de arranjos físicos com melhor eficiência, visto que o *Excel* possibilita a interação entre o ambiente de programação do VBA com as planilhas que representam de forma clara e organizada as máquinas e processos de uma empresa.

3. Materiais e Métodos

O algoritmo *Close Neighbour* é desenvolvido por meio de uma matriz de incidência, trabalhando com o rearranjo das linhas e colunas da matriz, objetivando concentrar as incidências ao longo da diagonal principal e, dessa forma, proporcionar a visualização de agrupamento de máquinas. Apresentado por Boe e Cheng em 1991, torna-se uma ferramenta eficiente nos trabalhos de estudo de arranjos físicos.

O desenvolvimento do algoritmo acontece a partir de uma matriz inicial cujo tamanho será de acordo com o número de máquinas e processos envolvidos no espaço físico. As máquinas serão transformadas nas linhas e os processos são as colunas dessa matriz.

Sabe-se que cada processo dessa matriz de incidência tem um roteiro de produção, constituído por uma ou mais máquinas. O preenchimento da matriz, dessa forma, acontece orientado através do roteiro de produção e observando o seguinte critério: toda vez que um processo utilizar determinada máquina, assinala-se '1' na célula (posição da matriz) correspondente ao cruzamento do processo com a máquina. A célula continua vazia caso contrário.

Com essa etapa de preenchimento concluída, cria-se uma segunda matriz cujo tamanho será o número de máquinas para as linhas e colunas, proporcionando cruzamentos de máquinas com máquinas. As células dessa matriz serão preenchidas da seguinte forma: soma-se, na matriz inicial, o número total de processos que passaram por cada par de máquina. Isso deve ser feito com todos os pares de máquinas possíveis. Essa soma preencherá as células correspondentes, na matriz 2, desses pares de máquinas. Formando, assim, uma matriz simétrica e de diagonal principal vazia. Soma-se, na matriz 2, os valores inseridos nas células de cada linha. Cada linha, assim, terá uma soma total.

Com a matriz 2 finalizada, a próxima etapa consiste em construir uma matriz 3 cujo conteúdo é a matriz inicial preenchida ('1's ou vazio) com uma nova ordenação dos índices das linhas.

O primeiro índice da nova ordenação de linhas corresponderá à linha que possuir a maior soma na matriz 2. Para completar os índices restantes da nova ordenação, deve-se seguir esta diretriz: analisar a linha do índice que foi escolhido anteriormente, sempre na matriz 2. Nessa linha, cada posição cruza com uma coluna (máquina). O número índice de cada coluna significa a nova linha escolhida. Para escolher a posição correta: opta-se pela qual possuir o maior número em seu conteúdo e que ainda não foi escolhida para a ordenação. Em caso de empate: dar preferência à posição de menor índice. Quando não existirem mais posições disponíveis para serem escolhidas: preencher com os índices remanescentes.

Após a ordenação de linhas concluída, tem-se uma matriz 3. Nela falta reordenar os índices das colunas para formar a matriz final.

Para estabelecer a ordem das colunas, primeiramente, será identificado, em cada coluna, os agrupamentos de '1's existentes. Um agrupamento significa uma sequência existente quando o conteúdo de uma posição for igual à '1' e a(as) posição(es) seguinte(s) também possuir(em) '1'. Ressaltando que um agrupamento pode ser considerado tal com apenas um '1'. Cada agrupamento, logicamente, deve começar em uma posição da matriz que tenha '1' como conteúdo e terminar quando alcançar uma posição da matriz sem '1' ou chegar ao limite da mesma. É possível que as colunas tenham vários grupos ou nenhum.

Após identificar esses agrupamentos nas colunas da matriz 3, é preciso destacar o agrupamento de maior tamanho ou o único existente de cada coluna. O tamanho de cada agrupamento é a soma das posições que constituem o mesmo. Se um agrupamento tiver uma sequência de '1's ao longo de 4 células da matriz, seu tamanho é 4.

A próxima etapa consiste na ordenação dos índices das colunas. Para tanto, será feita uma leitura através de cada linha da matriz 3.

Em uma linha, cada posição cruza com uma coluna. O número índice de cada coluna significa a nova coluna escolhida. É preciso identificar as posições da linha que a partir desta formam agrupamentos na coluna correspondente.

Para escolher a posição correta: caso o agrupamento formado na coluna a partir da linha na qual a leitura está sendo feita corresponder ao maior agrupamento desta coluna ou igualar-se com outro do mesmo tamanho, então se seleciona essa posição para a ordenação. Nenhuma coluna que já constitui a nova ordenação de colunas pode ser selecionada novamente.

É possível que muitas posições de determinada linha iniciem os maiores agrupamento de colunas, então essas posições, respectivamente, irão constituir a ordenação de colunas.

Após essa leitura por todas as linhas, se ainda existirem colunas que não foram selecionadas, estas são posicionadas completando a ordenação com os índices remanescentes.

Com a ordenação de colunas estabelecida, é possível criar a matriz final. Essa matriz corresponde à matriz inicial preenchida, com a reordenação de linhas e, na sequência, reordenação de colunas.

No exemplo a seguir, temos uma matriz formada por 8 máquinas e 20 processos. Observa-se que nas posições cujos conteúdos aparece '1', significa que o processo (coluna) está sendo utilizado pela máquina (linha) da posição. O processo 3, por exemplo, percorre as máquinas 1, 2, 4, 7 e 8.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1		1	1					1	1		1		1	1		1	1		1	
2			1	1		1	1							1				1		1
3		1						1	1		1		1	1		1	1		1	
4			1	1		1	1			1								1		1
5	1				1	1				1		1			1		1			
6	1				1				1	1		1			1					1
7			1	1		1	1				1	1						1		1
8			1	1		1	1											1		1

Figura 1 - Matriz 1 (Fase de preenchimento)

Com a matriz 1 completamente preenchida, a próxima etapa consiste na criação de uma segunda matriz. As linhas e colunas serão o número de máquinas. O objetivo é preencher as posições com a soma de processos utilizados a cada par de máquina. É possível observar na matriz 1 as máquinas 7 e 8, como exemplo, tendo os processos 3, 4, 6, 7, 18 e 20 passando por ambas em seus caminhos, somando 6 processos ao todo para o par 7 e 8. Essa soma aparece, na matriz 2, nas posições onde o par se cruza (7x8 e 8x7).

	1	2	3	4	5	6	7	8	
1		2	9	1	1	1	2	1	17
2	2		1	6	1	1	6	6	23
3	9	1		0	1	1	1	0	13
4	1	6	0		2	2	6	6	23
5	1	1	1	2		5	2	1	13
6	1	1	1	2	5		2	1	13
7	2	6	1	6	2	2		6	25
8	1	6	0	6	1	1	6		21

Figura 2 - Matriz 2 com somas das linhas

A diagonal principal da matriz 2 sempre ficará vazia, pois não é possível uma máquina formar um par com ela mesma. Após o cálculo dessa matriz, é necessário somar todas as linhas, como aparece ao lado de cada linha.

Agora a reordenação das linhas começa. Essa reordenação consiste em uma nova sequência de índices para as máquinas da matriz 1. A primeira linha a constituir a primeira posição da nova ordenação é a que possui maior soma na matriz 2. Seguindo o exemplo, é a linha 7. As próximas são selecionadas analisando a linha selecionada anteriormente.

Índice da reordenação	Máquinas pré-selecionadas	Máquina selecionada	Explicação
1	7	7	Linha maior soma.
2	2, 4, 8	2	Menor índice.
3	4, 8	4	Menor índice.
4	8	8	Maior valor.
5	1, 5, 6	1	Menor índice.
6	3	3	Maior valor.
7	5, 6	5	Menor índice.
8	6	6	Linha restante.

Figura 3 - Reordenação das linhas

Nenhuma linha pode ser selecionada para mais de um índice. É por isso que algumas linhas não são pré-selecionadas mesmo se encaixando nos critérios de seleção. Dado momento da reordenação, em eventuais matrizes, não terá mais linhas que entrem nos critérios de seleção. Sendo assim, opta-se pela linha remanescente. Quando há empate entre determinados valores, o menor índice prevalece.

Na coluna de máquinas selecionadas tem escrito a nova sequência de linhas a constituírem a matriz 3.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
7			1	1		1	1				1	1						1		1
2			1	1		1	1							1				1		1
4			1	1		1	1			1								1		1
8			1	1		1	1											1		1
1		1	1					1	1		1		1	1		1	1		1	
3		1						1	1		1		1	1		1	1		1	
5	1				1	1				1		1			1		1			
6	1				1				1	1		1			1					1

Figura 4 - Matriz 3 (Reordenação das linhas da matriz 1)

Com a matriz 3 pronta, agora parte-se para a reordenação das colunas para construir a matriz final. Para isso, analisa-se todas as colunas a partir de cada linha, na matriz 3.

A análise começa no primeiro índice entre as linhas. No exemplo as posições 3, 4, 6, 7, 11, 12, 18 e 20 da linha o conteúdo corresponde ao '1'. Esses valores correspondem ao índice de cada coluna e são os pré-selecionados. Entre essas 8 colunas, a maior sequência de '1's que acontece, em suas respectivas colunas, partindo da linha primeira são as de índice 3, 4, 6, 7, 18 e 20. Elas, respectivamente, constituirão as primeiras posições da nova sequência de colunas.

Observa-se que na coluna 11 existe uma sequência de '1's (a maior da coluna) que parte da linha número 1, índice 5. Na 12, a maior sequência de '1's da coluna começa na linha número 5, índice 7. É por isto que essas colunas foram apenas pré-selecionadas: existe '1' na coluna partindo da primeira linha.

Após a análise de todas as colunas partindo do primeiro índice, faz-se o mesmo a partir da linha 2. Coluna pré-selecionada: 14. As outras colunas que possuem '1' na linha número 2 já foram selecionadas anteriormente, então não entram na pré-seleção. A coluna 14 não constituirá a reordenação de colunas ainda, pois na mesma coluna existe uma sequência de '1's maior, que parte da linha número 1, índice 5.

É possível verificar toda a reordenação de colunas, desse exemplo, observando a tabela seguinte.

Linha analisada	Coluna(s) pré-selecionada(s)	Coluna(s) escolhida(s)	Explicação
7	3, 4, 6, 7, 11, 12, 18, 20	3, 4, 6, 7, 18, 20	Formam maiores sequências de '1'.
2	14	Nenhuma	Menor sequência da coluna.
4	10	Nenhuma	Menor sequência da coluna.
8	Nenhuma	Nenhuma	Nenhum '1' disponível na linha.
1	2, 8, 9, 11, 13, 14, 16, 17, 19	2, 8, 9, 11, 13, 14, 16, 17, 19	Formam maiores sequências de '1'.
3	Nenhuma	Nenhuma	Nenhuma entra nos <u>critérios</u> .
5	1, 5, 10, 12, 15	1, 5, 10, 12, 15	Formam maiores sequências de '1'.
6	Nenhuma	Nenhuma	Sem colunas remanescentes.

Figura 5 - Reordenação das colunas

Com a reordenação das colunas prontas, tem-se a matriz final estabelecida. Ela apresenta a matriz 1 com uma nova ordenação de linhas e de colunas, respectivamente.

	3	4	6	7	18	20	2	8	9	11	13	14	16	17	19	1	5	10	12	15
7	1	1	1	1	1	1				1										
2	1	1	1	1	1	1						1								1
4	1	1	1	1	1	1														
8	1	1	1	1	1	1														1
1	1						1	1	1	1	1	1	1	1	1					
3							1	1	1	1	1	1	1	1	1					
5			1											1		1	1	1	1	1
6						1		1								1	1	1	1	1

Figura 6 - Matriz Final (Reordenação das colunas da matriz 3)

Na matriz final é possível observar os agrupamentos que foram formados ao longo da diagonal principal.

4. Implementação

A implementação do algoritmo *Close Neighbour* aconteceu no ambiente de desenvolvimento do pacote *Office*, exclusivamente do *Excel*. A linguagem de programação usada foi o *Visual Basic for Applications*. Optou-se por essas ferramentas em função do *Excel* fornecer grande acessibilidade para se trabalhar com matrizes. O ambiente de programação que fornece diversos módulos, classes e opções de criação de interfaces gráficas é acessado através da janela Modo Desenvolvedor, figura 7. Através desse ambiente, que não é visível para um usuário convencional, o processo de desenvolvimento do algoritmo aconteceu.

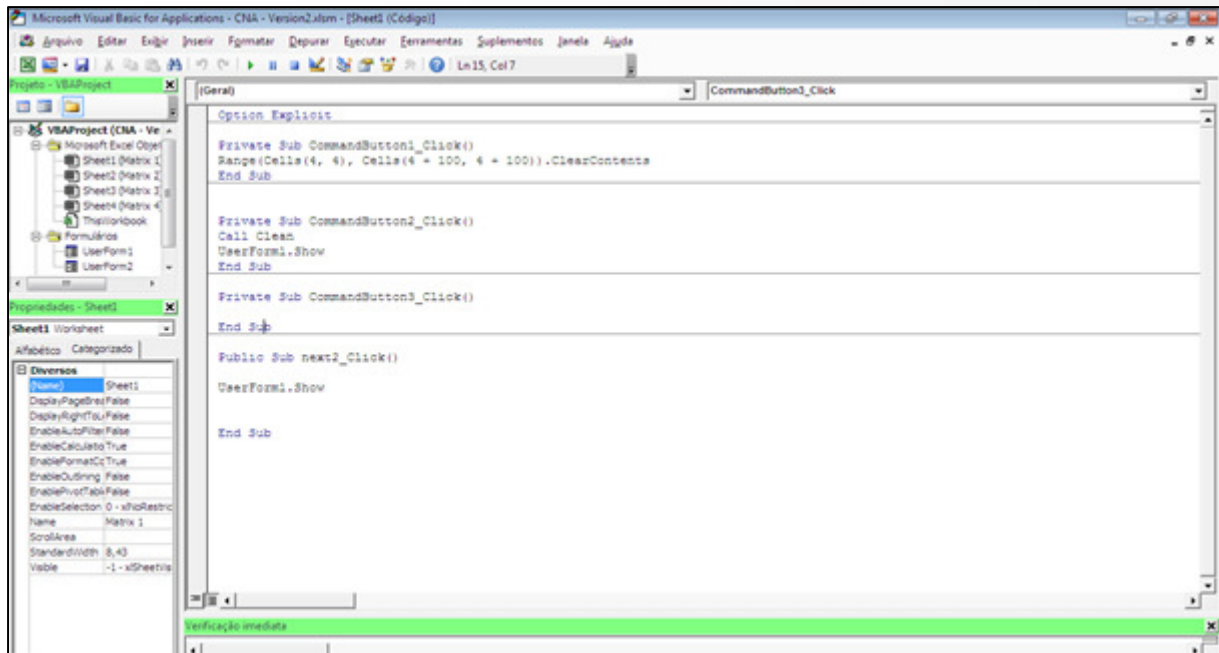


Figura 7 - Modo Desenvolvedor

As primeiras etapas foram, em uma tela inicial, figura 8, construir uma interface gráfica onde o futuro usuário entraria com os dados relacionados à matriz de máquinas e processos. Esses dados iniciais (número de máquinas e o número de processos), eram usados para construir uma matriz de entrada após o botão “Next” ser pressionado. Desde o botão de confirmação até o lugar para informar as quantidades, são ferramentas fornecidas pela interface gráfica do *Visual Basic*.

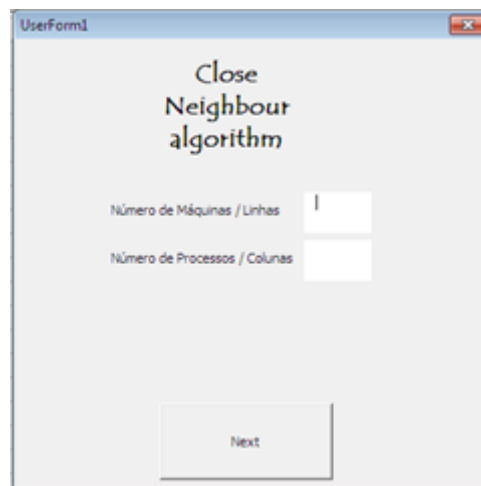


Figura 8 - Tela Inicial

Após a informação referentes à matriz, é criado, numa primeira planilha, os índices e células da matriz, prontas para serem preenchidas pelo usuário com ‘1’s, figura 9. Junto dessa planilha, o *Visual Basic* também fornece o acesso às ferramentas gráficas. O botão “Next”, assim, foi implementado junto à planilha para o usuário informar a conclusão do preenchimento da Matriz Inicial. Com esse preenchimento feito e o botão pressionado, novas abas são automaticamente criadas, cada uma sendo uma planilha, mostrando as matrizes referentes às etapas do algoritmo *Close Neighbour*. A última matriz será a matriz final do algoritmo com linhas e colunas reordenadas. Também, criou-se o botão “clean” para efetuar a limpeza dos dados inseridos e o

botão “reset” com a finalidade de reiniciar os dados referentes a matriz de máquinas x processos.

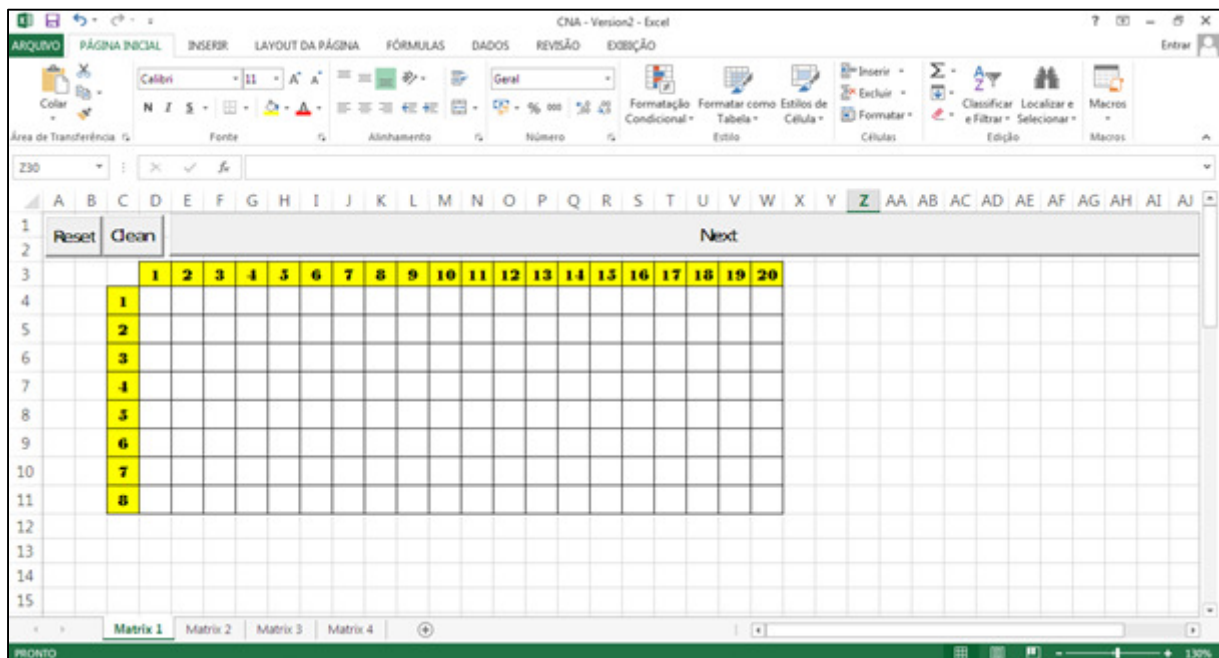


Figura 9 - Matriz Inicial sem preenchimento

De posse da matriz final do CNA, é possível visualizar os possíveis agrupamentos que são indicados para os processos em estudo.

5. Conclusões

A proposta desta pesquisa gira em torno de oferecer aos agentes envolvidos na gestão de *Layout* uma ferramenta capaz de atender as exigências de complexidade que a atividade de planejamento de arranjo físico impõe. Dessa forma, busca-se aumentar a competitividade dos sistemas industriais. A implementação do CNA ocorreu e gerou resultados confiáveis através da validação feita com a matriz de testes apresentada por Boe e Cheng. Futuros trabalhos serão possíveis e necessários no sentido de promover aplicações da ferramenta em estudos de caso.

É oportuno destacar algumas acessibilidades que o VBA fornece, auxiliando bastante nos momentos de trabalhar com a comunicação entre as tabelas do *Excel* e a programação em si. Tudo que o Pacote *Office* fornece de edição para as células e os textos em geral é possível automatizar através do ambiente de programação. Trabalhar com as células e planilhas do *Excel* através dessa linguagem de programação é bem agradável ao programador, principalmente quando se trata do uso de matrizes e tabelas. Ressalta-se as sintaxes de programação do *Visual Basic*, que são bastante didáticas até mesmo para leigos na área. A maioria delas exige um prévio conhecimento em inglês.

A implementação do algoritmo no *Visual Basic* e *Excel* conta com as facilidades que o *Software* em si fornece, porém toda a metodologia do algoritmo, isto é, transformar a matriz inicial em uma matriz final com os agrupamentos, requer o uso de repetições, condições, testes e sequências, características genéricas das linguagens de programação. Na implementação do algoritmo no *Visual Basic*, foi criada uma classe para construção e cálculo do algoritmo do começo ao fim. Nessa classe têm-se estruturas desde ‘*While*’s, ‘*For*’s, ‘*If*’s, ‘*Then*’s até ‘*Dim*’s. Todas essas palavras fundamentam a linguagem de programação *Visual Basic For Applications*.

Referências

- AIOSA, RODRIGO.** *O que é VBA?* Disponível em <http://www.escolaexcel.com.br/2011/08/o-que-e-vba.html>, acesso em 13 set. 2014.
- CONFEDERAÇÃO NACIONAL DA INDÚSTRIA – CNI.** *Indicador de custos industriais.* Publicação Trimestral. Ano 2, n. 2, abril/junho de 2013. ISSN 2317-7039.
- BATAGLIN, L. M. C.; BASÍLIO, T. L.; VIEIRA, J. G. V.; ROSSETTI, N.** *Avaliação de arranjo físico e custos logísticos na transferência de maquinário em uma unidade fabril.* Revista Eletrônica Produção & Engenharia, v. 3, n. 2, p. 300-308, Jan./Jun. 2013.
- BOE, W.J. CHENG, C. H.** *A close neighbour algorithm for designing cellular manufacturing systems.* International Journal of Production Research, v. 29, n. 10, p. 2097 – 2116, 1991.
- DALMAS, VOLNEI.** *Avaliação de um layout celular implementado: Um estudo de caso em uma indústria de autopeças.* Dissertação (Mestrado em Engenharia) - Universidade Federal do Rio Grande do Sul-UFRGS, Porto Alegre, 2004.
- DAVIS, M.; AQUILANO, N.; CHASE, R.** *Fundamentos da Administração da Produção.* Porto Alegre: Bookman, 2001.
- FOGLIATO, FLÁVIO SANSON; NARA, ELPIDIO OSCAR BENITEZ.** *Processos, Programação, Planejamento e Controle da Produção.* Santa Cruz do Sul: Universidade de Santa Cruz do Sul/ Departamento de Engenharia, 2013.
- MICROSOFT.** *How to automate Microsoft Excel from Visual Basic .NET.* <http://support.microsoft.com/kb/301982/pt-br>, acesso em 20 set. 2014.
- NETO, RUY CARNEIRO GIRALDES.** *Importância do Layout Interno.* Disponível em <http://pt.slideshare.net/ruygiraldes/importncia-do-layout-interno>, acessado dia 14/09/2014.
- NEUMANN, C. S. R.; FOGLIATO, F. S.** *Sistemática para avaliação e melhoria da flexibilidade de layout em ambientes dinâmicos.* Gestão&Produção, São Carlos, v. 20, n. 2, p. 235-254, 2013.
- RAWABDEH, I.; TAHBOUB, K.** *A new heuristic approach for a computer-aided facility layout.* Journal of Manufacturing Technology Management, v. 17, n. 7, p. 962-986, 2006.
- SILVA, ANTÔNIO.** *Programação em VBA: Texto Introdutório*, versão 3. Porto: Instituto Superior de Engenharia do Porto/Departamento de Engenharia Informática, 2009.
- SLACK, N.; CHAMBERS, S.; JOHNSTON, R.** *Administração da produção.* 3ª ed. São Paulo: Atlas, 2009.