

Framework computacional aplicado na modelagem e programação da produção

Computational framework applied to modeling and production scheduling

André Felipe Henriques LIBRANTZ [1](#); Cleber Gustavo DIAS [2](#); Valdecir de Oliveira PEREIRA [3](#)

Recibido:31/08/16 • Aprobado: 28/09/2016

Conteúdo

- [1. Introdução](#)
- [2. Fundamentação teórica](#)
- [3. Metodologia de pesquisa](#)
- [4 Exemplos de aplicação do framework proposto](#)
- [5. Resultados obtidos após a implantação do software protótipo](#)
- [6. Conclusão](#)

[Referências Bibliográficas](#)

RESUMO:

Este trabalho apresenta um framework com aplicação na Programação avançada da produção. Tal *framework* pode fornecer uma infraestrutura básica para o desenvolvimento de software, a fim de atender as demandas dos sistemas de produção atuais. Um software protótipo foi implementado utilizando o framework proposto que permitiu o uso das técnicas de programação CSP (*Constraint Satisfaction Problem*), e algoritmos genéticos e regras de liberação, como coordenadores da programação. O sistema foi implantado em uma indústria de moldes e os resultados indicam que ele poderia ser uma boa alternativa, combinando facilidade de modelagem e bons resultados na programação da produção.

Palavras-chave: Programação da Produção, Planejamento Avançado da produção; Regras de Despacho; Algoritmos Genéticos.

ABSTRACT:

This paper presents a framework with application to advanced production scheduling. The Framework can provide a basic infrastructure for the development of software to meet the demands of today's production systems can be extended to meet other requirements. A prototype software was modeled and implemented using the proposed framework, which allowed the use of CSP programming techniques (*Constraint Satisfaction Problem*), genetic algorithms and dispatching rules and programming engineers. The system was deployed in a mold industry and the results pointed that it could be a good alternative, combining ease of modeling and good results in the production scheduling.

Keywords: Framework, Advanced Production scheduling; Order Rules; Genetic Algorithms.

1. Introdução

O ambiente competitivo atual leva as empresas a produzir melhor, mais rápido e com menor

custo. Para se manter à frente de seus concorrentes, é necessário o uso de ferramentas que podem ajudar na tomada de decisão. Neste contexto, um software de planejamento de produção, pode ser uma ferramenta importante de melhoria de desempenho. Neste contexto, a programação da produção despertou o interesse de inúmeros pesquisadores nos últimos anos, especialmente nas últimas décadas, no sentido de alcançar soluções ótimas para a redução de tempo e custo de *setup*, *makespan*, tempo de fluxo, atraso das tarefas e também no atendimento das ordens de produção dentro de prazos de entrega e com o uso eficiente dos recursos de fábrica (Morais, 2009; Tubino, 2009). Deve-se ressaltar o grande desafio para realizar uma programação das ordens de produção, considerando todas as características de m processo de fabricação, ao mesmo tempo em que se atende às suas restrições técnicas e tecnológicas (Duin, Sluis, 2006; Hvolby, 2010; Xing, 2009).

Adicionalmente, cabe salientar que, a deficiência na geração de uma programação detalhada ocorre em função de diversas características como: *lead times* superestimados, programação infinita, mesmo com a adoção do módulo CRP (*Capacity Requirements Planning*) que faz uma verificação aproximada da capacidade e não detalhada como nas soluções de programação da produção (Girotti, 2011).

Em um sentido mais amplo, e tomando por base uma revisão da literatura, vale destacar que muitos autores dedicaram-se ao desenvolvimento de estudos sobre programação da produção. Reis (1996) fez um resumo das ferramentas de sequenciamento de produção desenvolvidas até 1996 e as tecnologias empregadas em cada solução com os benefícios e problemas de cada ferramenta. Brailsford, Potts e Smith (1999) fizeram uma revisão das principais técnicas utilizadas para resolução de restrições como arco consistência e propagação de restrições, concluindo que não é o método mais eficiente em grandes espaços de busca em comparação com métodos como Algoritmos Genéticos ou *Simulated Annealing*, mas que pode ser bastante eficiente principalmente em problemas de busca local.

Pacheco e Santoro (2001) realizaram um estudo de caso em seis empresas brasileiras consideradas líderes em seus setores, identificando a falta de conhecimento sobre a tecnologia APS (*Advanced Planning and Scheduling*) empregada, bem como definição clara dos objetivos e critérios formais de escolha. Brucker (2002) demonstrou uma generalização do problema RCPS (Resource-Constrained project scheduling problem), com o uso de propagação de restrições. Ichihara (2002) tratou o problema da alocação de recursos em projetos, com objetivo de minimizar o tempo de execução do projeto bem como o custo e valor presente do projeto, utilizando de métodos heurísticos.

Novas e Henning (2010) apresentaram um *framework* com técnicas para realização da realocação das operações, identificando somente as alterações necessárias, diminuindo o custo computacional ante o sequenciamento de todas as operações, reagindo instantaneamente ou não com informações em tempo real da fábrica, alterando somente as operações afetadas por mudanças e as que sofrerão impacto destas mudanças. Diversas estratégias foram abordadas e analisadas. A proposta de um *framework* para corrigir a programação utilizou orientação a objetos e CP (*Constraint Programming*). Framinan e Ruiz (2010) propuseram uma arquitetura básica para o desenvolvimento de soluções APS (*Advanced Planning and Scheduling*). A arquitetura por eles proposta apresenta os principais módulos e as diversas técnicas que podem ser empregadas para solução do sequenciamento de operações.

Os autores ressaltam a grande quantidade de artigos que não consideram as características reais da fábrica. Zeballos (2010) apresentou uma metodologia que utiliza conceitos de programação de restrição aplicada em sistemas de manufatura flexível. Para a resolução das restrições o autor utilizou o solver comercial da IBM, ILOG e conseguiu bons resultados em menos de 2000 segundos de processamento.

Hvolby e Steger-Jensen (2010) fizeram um levantamento dos principais trabalhos sobre sequenciamento de produção baseado em restrições, demonstrando através do uso da solução Oracle APS os benefícios em relação aos algoritmos MRP. Mainieri e Ronconi (2010) propuseram uma nova regra de despacho denominada PRTTa para o problema flowshop flexível. Os autores

encontraram resultados satisfatórios com a redução de *makespan*.

Weglaz *et al.* (2010) fizeram uma vasta pesquisa bibliográfica sobre as mais importantes publicações dos últimos anos, sobre sequenciamento de atividades de projetos, analisando as diversas técnicas empregadas até hoje. Os autores analisaram diferentes pontos de vista sobre sequenciamento com as diferentes estratégias.

Xu, Xu e Xie (2011) fizeram um levantamento das últimas pesquisas relacionadas ao problema de dupla restrição entre máquinas e operadores, conhecido na literatura como DRC (*Dual Resource Constrained*). Os autores fazem uma crítica à adoção de regras de liberação estáticas e sugerem uso de outras técnicas como Rede Neural Artificial e Algoritmos Genéticos defendendo o uso de regras dinâmicas de programação.

Silva, Morabito e Yanasse (2011) propuseram uma solução utilizando programação inteira mista, para indústria aeronáutica com uso de gabaritos de montagem que sofrem restrição adicional com a limitação de uso de recursos adjacentes.

Esta revisão da literatura destaca diversos esforços da comunidade científica em buscar melhores soluções para programação de produção. No entanto não foi encontrada durante as buscas uma proposta de framework que pudesse ser usada como base para o projeto de um software de programação avançada de produção.

Frente ao acima exposto, o presente trabalho apresenta um *framework* computacional, empregando um diagrama de classes em UML (*Unified Modeling Language*), bem como a aplicação do método de busca CSP (*Constraint Satisfaction Problem*), como método de busca local, associado ainda a regras de liberação e Algoritmos Genéticos, finalizando com o desenvolvimento de software protótipo. O restante deste artigo está organizado como segue: na seção dois são apresentados os principais fundamentos teóricos. A metodologia é apresentada na seção 3. Na quinta seção são apresentados os resultados da implementação do sistema proposto e na última seção são apresentadas as conclusões bem como sugestões de trabalhos futuros.

2. Fundamentação teórica

2.1. Framework conceitual

Como descrito por Pereira e Librantz (2015), o *framework* conceitual é o resultado do processo qualitativo de teorização e pode ser compreendido como uma rede de conceitos que juntos oferecem a compreensão abrangente de um fenômeno. O termo *framework* está bastante associado os princípios da programação orientada a objetos. O trabalho de Hautamäki (2005) sugeriu como definição para framework um esqueleto de implementação de uma aplicação ou de um subsistema em um domínio de problema em particular.

Assim, tal estrutura é formada por classes concretas, ou abstratas, e define um modelo de interação ou colaboração entre as instâncias de classes definidas pelo framework.

Neste sentido, e conforme apontado ainda por Pereira e Librantz (2015), o *framework* oferece infraestrutura de projeto permitindo ao desenvolvedor reduzir o esforço. As interconexões estabelecidas definem a arquitetura da aplicação. O desenvolvedor faz uso do framework estendendo e moldando as suas necessidades específicas (Dias, 2004). Nas próximas seções serão apresentadas todas as entidades do *framework* proposto, métodos de escolha, restrições, métodos de sequenciamento e relacionamento entre entidades.

2.2. Entidades

Weglaz *et al.* (2010) revelou que a capacidade produtiva da empresa pode ser representada pela disponibilidade de recursos produtivos (máquinas, operadores, ferramentas, materiais, energia, etc.), os quais no *framework* serão denominados de recursos, que podem ser divididos

em recursos discretos e recursos contínuos, sendo que cada um representa melhor algum tipo de recurso. O chamado recurso discreto (RRD) normalmente está relacionado aos postos de trabalho e máquinas, além do fato de ser entendido que sua disponibilidade é a variação da eficiência em função do tempo, ou seja, conforme muda o período, poderá ou não estar disponível com eficiência diferente.

Nos chamados recursos contínuos, que podem representar volume, como consumo de energia, utilização de funcionários, entre outros, a disponibilidade varia em função dos limites inferior e superior, em função do tempo (Pereira e Librantz, 2015). Quando uma operação ou um recurso, adicionar (adiciona), por exemplo, dois funcionários, caso uma segunda operação necessite também de dois funcionários deverá esperar o término da operação e a respectiva devolução (reduzir) dos funcionários. O Quadro 1 mostra, como exemplo, o modo como são utilizados alguns recursos contínuos.

Quadro 1 – Formas de consumo dos recursos contínuos.

Tipo	Descrição
Adicionar ao início setup	Incrementa o valor imediatamente ao início do setup
Reduzir ao início de setup	Decrementa o valor imediatamente ao início do setup
Adicionar ao início da operação	Incrementa o valor imediatamente ao início da operação e fim do setup
Reduzir ao início da operação	Decrementa o valor imediatamente ao início da operação e fim do setup
Adicionar ao término da operação	Incrementa o valor ao término da operação
Reduzir ao término da operação	Decrementa o valor ao término da operação

Fonte: Adaptado de Weglaz et al (2010).

A fim de facilitar o entendimento, deve-se mencionar que, as operações serão nomeadas a partir deste ponto simplesmente como OP e as ordens de produção que são representadas no *framework* como um conjunto de operações, que passarão a ser nomeadas como ORD.

2.3. Métodos de Escolha

Pereira e Librantz (2015) revelaram que as restrições que utilizam recursos RRD e RRC, que serão detalhadas adiante, possuem um método de escolha para quando possuírem mais de um recurso capaz de executar a mesma operação, a escolha de qual será o recurso eleito depende do método escolhido, os quais seguem:

- Disponível primeiro: Dentre todos os recursos definidos na restrição o que estiver disponível para executar a operação o mais breve possível, será o recurso eleito.
- Terminar primeiro: Dentre todos os recursos definidos na restrição o que terminar a operação o mais breve possível, será o recurso eleito.
- Menor Custo: Dentre todos os recursos definidos na restrição, independente do início ou término da operação, o que consumir menos recursos financeiros será o recurso eleito.

d) Menor Tempo de Processamento: Dentre todos os recursos definidos na restrição, o recurso em que a duração total da operação for menor, será o recurso eleito.

2.4 Restrições

As restrições são consideradas como entidades que muitas vezes limitam, condicionam ou impedem a execução de uma dada operação. Deste modo, deve-se avaliar todas as restrições com a finalidade de melhor avaliar aquelas que representam as características relevantes do processo. Por outro lado, vale frisar que, as restrições somente podem ser associadas a operações e recursos. A relação de restrições utilizadas neste trabalho com as suas respectivas características é apresentada a seguir (Pereira e Librantz, 2015):

a) RRD (Recurso Discreto): Esta restrição valida um conjunto de recursos discretos utilizando um método de escolha conforme citado anteriormente, retornando apenas um recurso válido para a restrição. Caso sejam necessários dois recursos simultâneos para a mesma operação, deve-se adicionar outra restrição deste tipo.

b) RRC (Recurso Contínuo): Na restrição de recurso contínuo, devemos além do método de escolha do recurso, que é o mesmo utilizado em recursos discretos, informar quais recursos irão participar do processo de escolha, pois somente um recurso válido contínuo será definido por restrição. Caso deseje mais de um recurso válido contínuo definido, deverá adicionar mais uma restrição com os respectivos recursos contínuos. Para adicionar um recurso contínuo será necessário informar uma quantidade e o método de uso conforme apresentado no Quadro 1.

c) RDP (Dependência entre operações): Para que a operação com esta restrição seja alocada a operação antecessora deverá estar devidamente alocada.

d) ROA (Operação Alocada): Esta restrição garante que apenas será alocada a operação com esta restrição se não existir nenhuma operação alocada neste recurso e adjacentes no mesmo período.

2.5. Método de Programação/Sequenciamento

Como descrito por Pereira e Librantz (2015), as regras de sequenciamento, também conhecidas como regras de despacho, determinam qual o critério de escolha das ordens ou operações que vão disputar os recursos da fábrica. Neste trabalho foram utilizadas as regras de despacho SPT (*shortest process-time*), cujo objetivo principal é programar primeiro as operações com o menor tempo de processamento por meio dos métodos de programação "para frente" de cada operação individualmente mantendo programada a operação que tiver menor duração. Como segundo método de programação foram utilizados os Algoritmos genéticos (AG), no qual os cromossomos que representam operações são sequenciados um após outro utilizando o método de programação para frente.

2.6 Algoritmos de busca e otimização

2.6.1. CSP (Constraint Satisfaction Problem)

Pereira (2011) e Pereira e Librantz (2015) revelaram que o CSP (*Constraint Satisfaction Problem*) pode ser considerado como o estudo de sistemas computacionais baseados em restrições, tendo por fundamento resolver problemas apenas especificando os requisitos (restrições) sobre a área problema ou, em outras palavras, sobre os valores possíveis (domínio) das variáveis. Neste caso, a solução para o problema passa por um conjunto de valores que atendam todos os requisitos.

Pode-se afirmar ainda que, o CSP (*Constraint Satisfaction Problem*) também pode ser modelado por meio de grafo de restrições, formando uma rede onde as variáveis de decisão são representadas pelos vértices e as restrições pelas arestas, que permite a utilização das

ferramentas para manipulação de grafos (Pereira e Librantz, 2015). Assim, as principais formas de busca em restrições modeladas através de grafos podem envolver: consistência de nó, consistência de arco e consistência de caminho, todos com o mesmo objetivo de minimizar o retrocesso do algoritmo por não ter encontrado a solução pelo caminho selecionado inicialmente (Brucker, 2002).

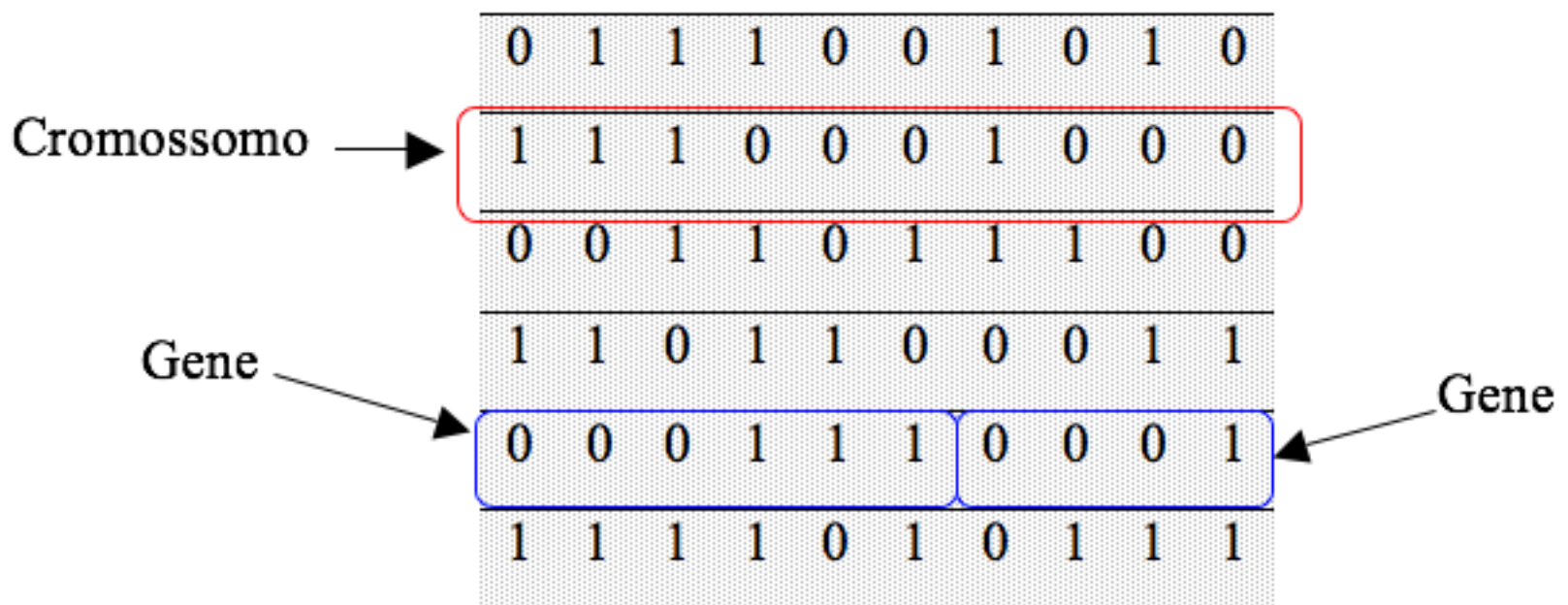
Adicionalmente, cabe destacar a facilidade em modelar o problema, uma vez que se podem adicionar restrições independentemente de como será a solução do problema.

2.6.2. Algoritmos Genéticos

Essencialmente, e conforme relevado por Araújo et al. (2007), os Algoritmos Genéticos (AG) consistem em um método de busca e otimização baseado no processo de seleção natural que simula a evolução das espécies.

O princípio de operação dos AGs leva em conta a geração de indivíduos, ou população, muitas vezes codificados em cadeias de bits (Figura 1), também chamados de cromossomos, que evoluem para encontrar a solução de um dado problema.

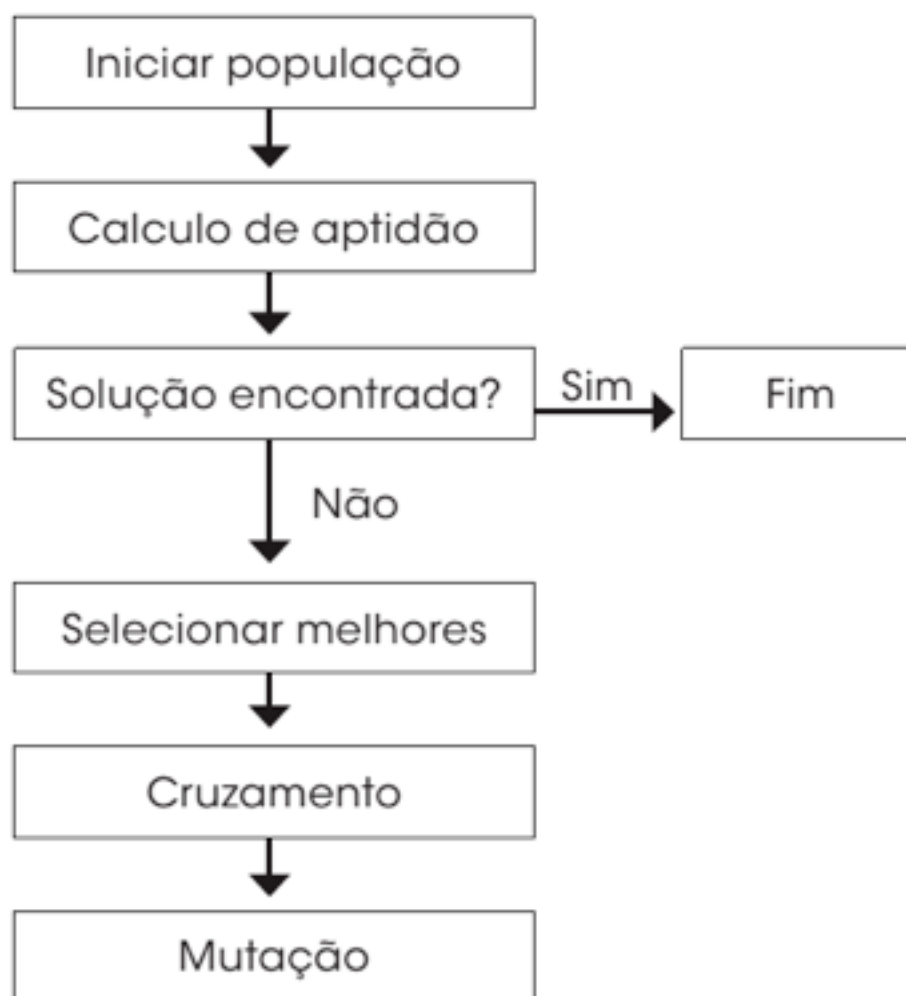
Figura 1 – Exemplo de uma população do AG



Fonte: Autor

Cada indivíduo, ou cromossomo, poderia ser considerado como dois, ou mais genes, por exemplo, codificados em partes distintas, como ilustrado ainda na Figura 1. A Figura 2 apresenta as principais etapas da técnica dos algoritmos genéticos.

Figura 2 . Esquema de funcionamento do Algoritmo Genético.



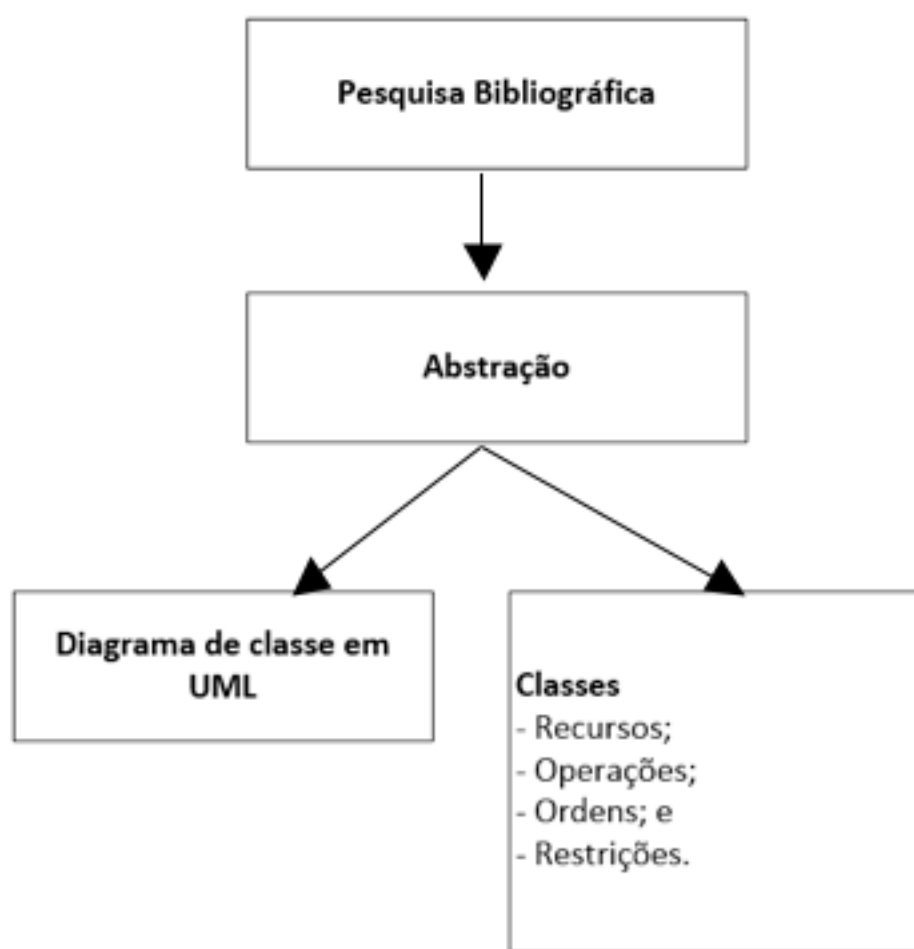
Fonte: Araújo et al. (2007).

Na primeira etapa a população inicial é gerada, na maioria das vezes, de forma aleatória. Na sequência, os indivíduos são avaliados por uma função chamada *fitness* na qual o indivíduo mais apto tem os melhores valores (função objetivo), seja de maximização ou minimização. Sobre a população são aplicados dois operadores, conhecidos como cruzamento (*Crossover*) e mutação (*Mutation*). Após a aplicação dos operadores é realizada novamente a função *fitness*, onde será selecionada apenas uma parte de toda a população restante para a nova geração. Desta forma a cada geração deve haver uma população (soluções) mais próxima do objetivo ou das melhores soluções. O algoritmo pode parar quando atingir um número de gerações, duração de tempo em processamento ou ter atingido um determinado objetivo (Araújo et al., 2007). Para este trabalho foram utilizadas como parâmetros para o algoritmo genético uma população inicial de 20 indivíduos, 30 gerações, taxa de mutação 4%. Em todos os casos o método de cruzamento foi o PMX (*Partially Matched Crossover*) e o método de mutação RSM (*Reverse Sequence Mutation*) (Abdoun, Abouchabacka, 2011).

3. Metodologia de pesquisa

O presente trabalho caracteriza-se como pesquisa de natureza aplicada, com a finalidade de gerar conhecimento, a partir da aplicação da solução em problemas específicos. Ainda, conforme exposto por Gil (2002), este trabalho pode ser classificado como exploratório-descritivo. O trabalho teve origem em uma pesquisa bibliográfica, realizada principalmente por meio de sistemas de busca no portal de periódicos da CAPES, Science Direct, Scielo, EBSCO host. A figura 3 apresenta as etapas do projeto.

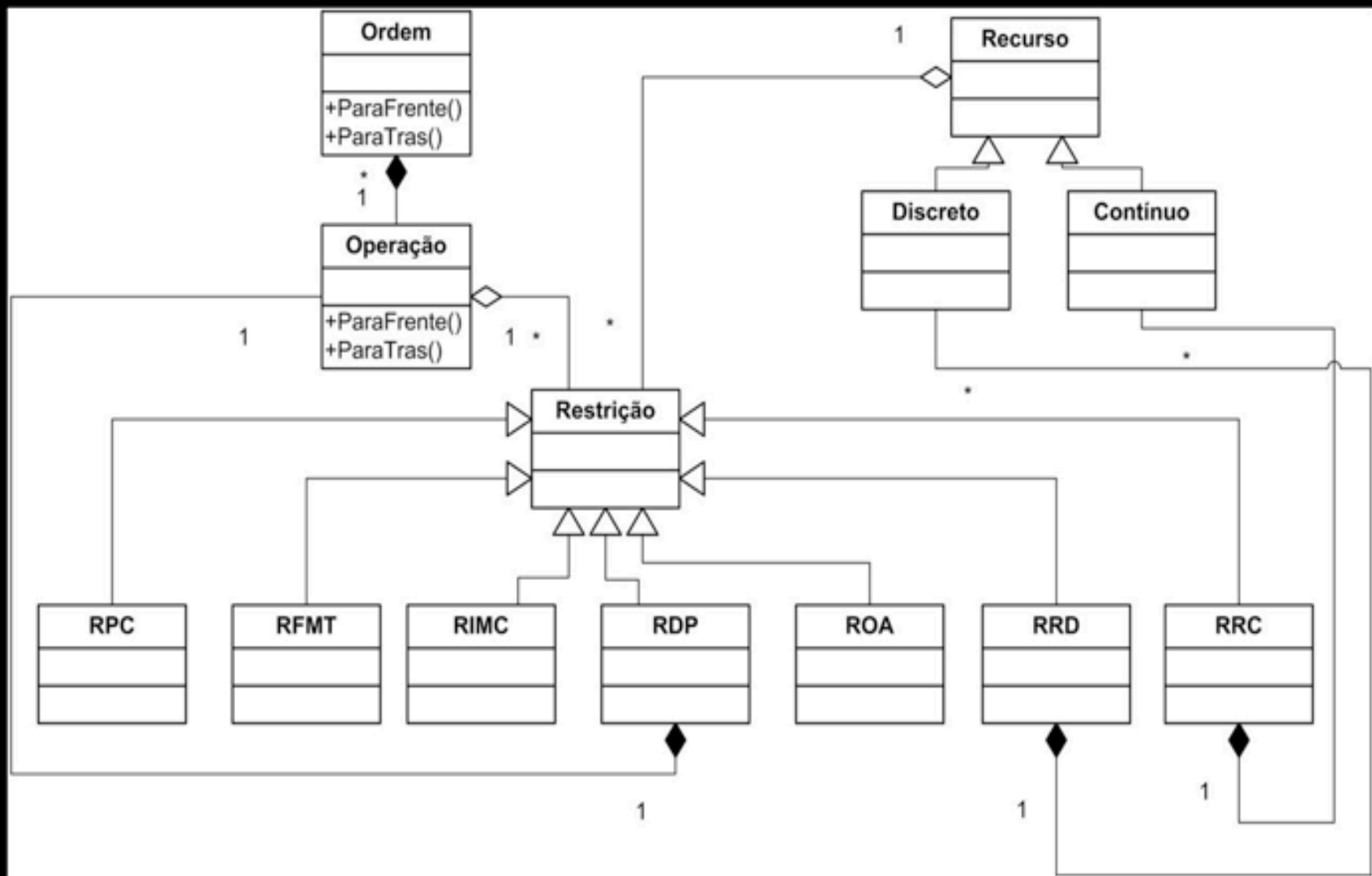
Figura 3. Metodologia de pesquisa



Fonte: Pereira e Librantz (2015)

O diagrama de classes foi desenvolvido utilizando a notação UML (*Unified Modeling Language*) e as premissas de abstração em modelagem de sistemas baseados em orientação a objetos. A figura 4 apresenta o diagrama de classes do framework proposto.

Figura 4 - Diagrama de Classes do *Framework* proposto.



Fonte: Pereira e Librantz (2015)

Como descrito por Pereira e Librantz (2015), o diagrama em UML acima representa a agregação de restrições em recursos ou operações. As restrições são especializadas em diversos tipos, sendo que cada tipo de restrição tem seu próprio objetivo, e as restrições RRC e RRD são compostas por um ou mais recursos dependendo da necessidade. Os recursos são especializados em discretos e contínuos e as ordens compostas por um conjunto de operações.

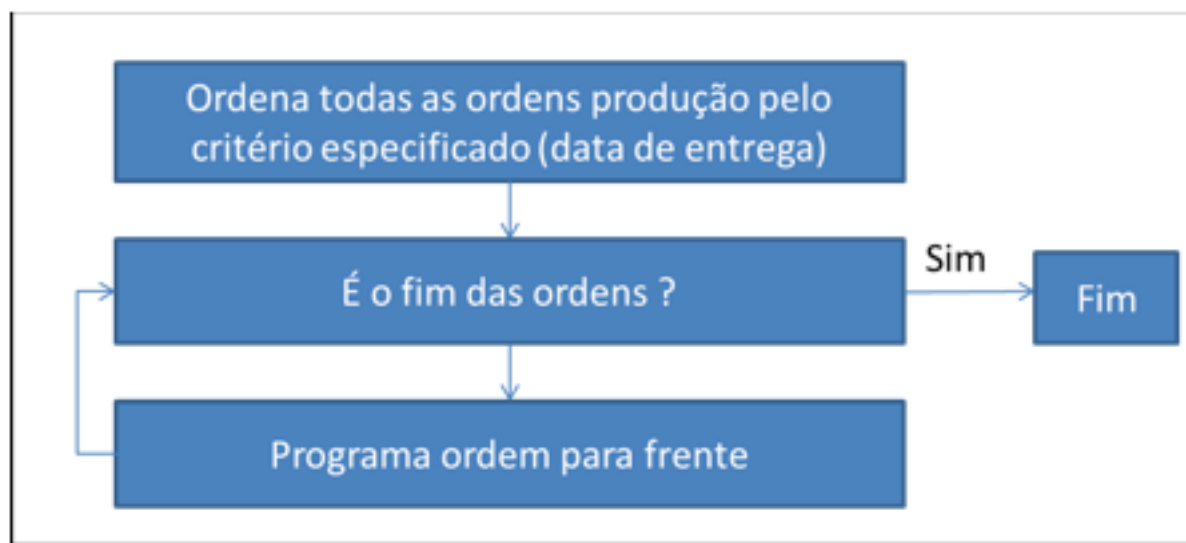
3.1 Sequenciamento das ordens e operações

Para o *framework* foram utilizadas regras de liberação e a técnica dos Algoritmos Genéticos como forma de demonstrar sua aplicação.

3.2. Regras de liberação

As regras de liberação, também conhecidas como regras de despacho, determinam qual o critério de escolha das ordens ou operações vão disputar os recursos da fábrica. As regras de liberação seguem normalmente as políticas de atendimento das ordens definidas pela empresa, ressaltando que podem ser utilizadas diversas regras de sequenciamento sobre um mesmo ambiente, a fim de se obter a melhor programação para a empresa naquele momento. Como citado anteriormente existe na literatura uma grande quantidade de regras de liberação e que podem ser adotadas pelo *framework*. No *framework* pode ser utilizado qualquer um dos tipos de regras de sequenciamento citados, bastando apenas que a regra chame o método da respectiva classe de ordem ou operação, conforme as regras exemplo abaixo. As regras estáticas que utilizam as informações da ordem seguem de forma geral o mesmo mecanismo da regra MDT (Menor data de Entrega) conforme o exemplo da Figura 5.

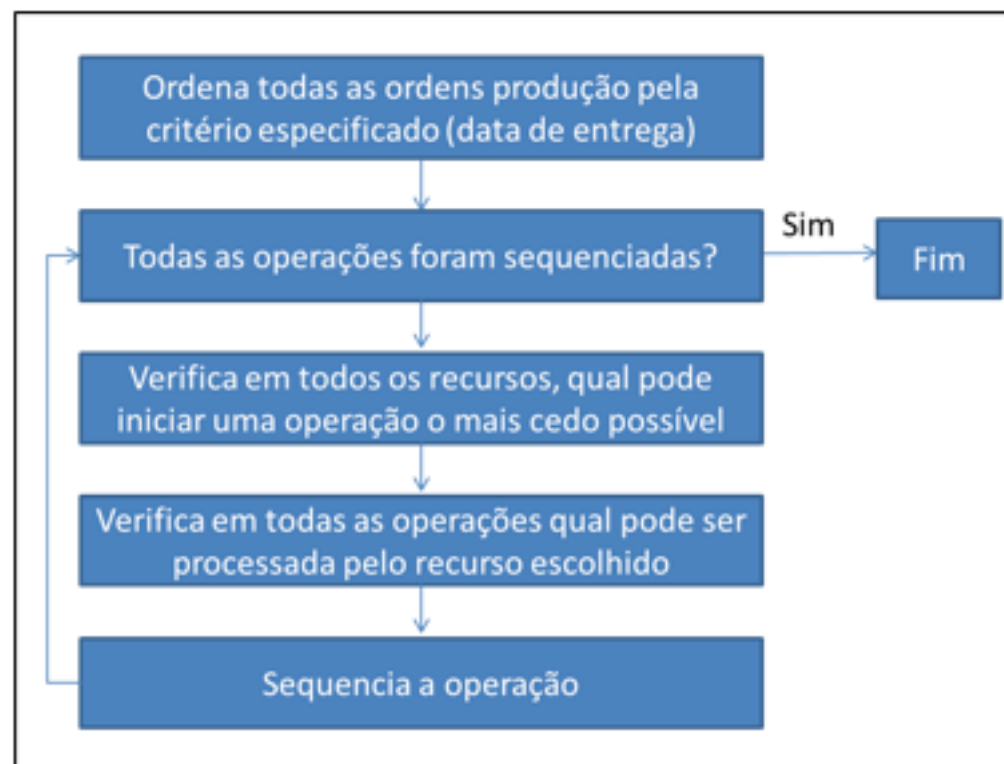
Figura 5. Fluxo da regra de liberação MDT.



Fonte: O autor

Também podem ser desenvolvidas regras dinâmicas como as regras de carregamento paralelo apresentado na Figura 6.

Figura 6 . Fluxo da regra de carregamento paralelo.



Fonte: O autor

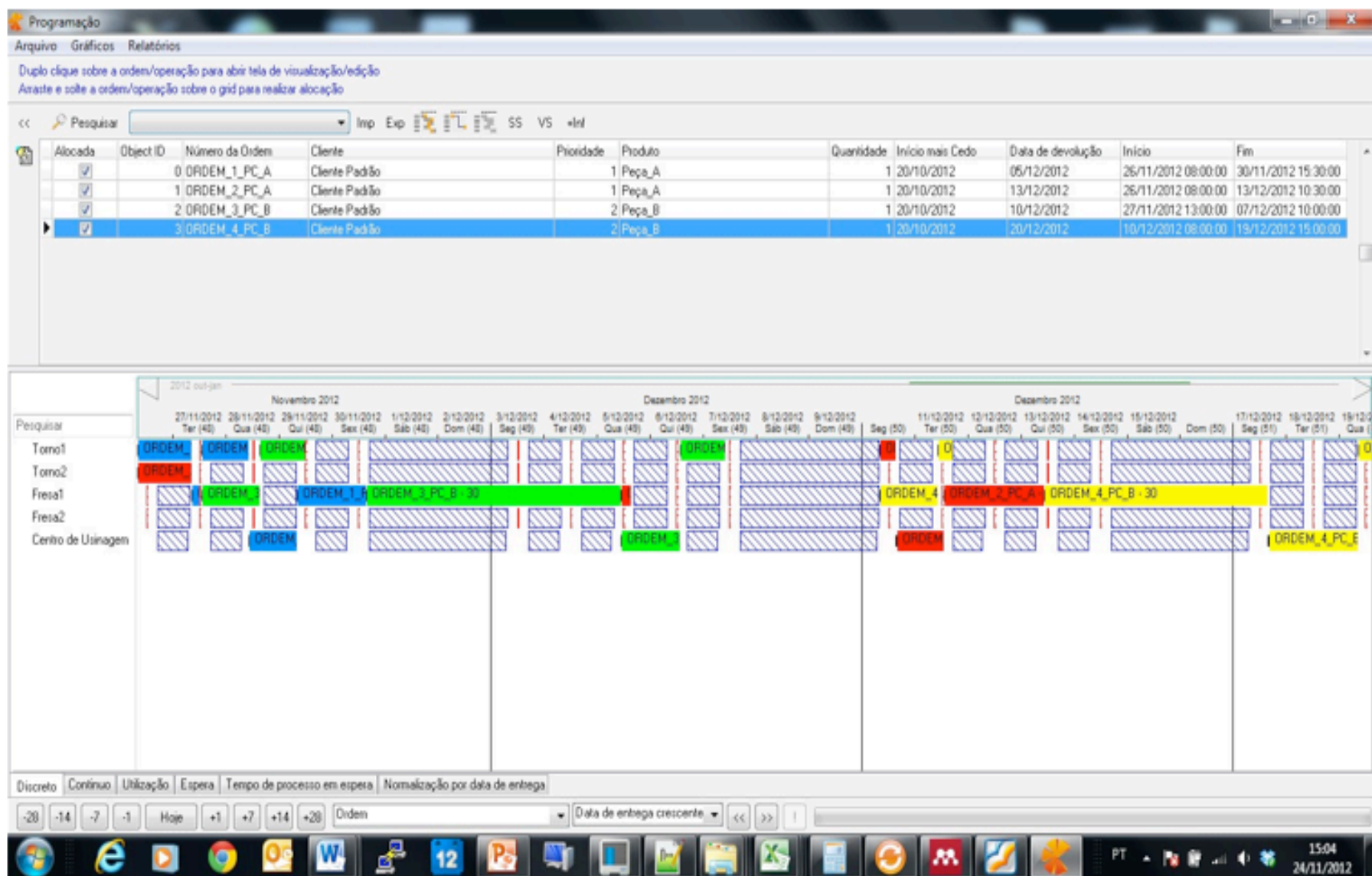
Para o processo de seleção dos indivíduos, foi feito o sequenciamento utilizando o método de sequenciamento para frente de todas as operações do indivíduo. São coletadas informações da programação de cada indivíduo como *makespan*, *leadtime* acumulado, atraso total, tempo de *setup* total, tempo em operação total, tempo em espera total. Estes indicadores têm valores em escalas diferentes e devem ser normalizados. Após a normalização o índice de cada indicador foi multiplicado pelo peso informado pelo usuário.

4. Exemplos de aplicação do framework proposto

O presente *framework* foi empregado no desenvolvimento de um software protótipo em linguagem Delphi e possui os seguintes módulos: **Cadastro de Recursos, Cadastro de Turnos, Cadastro de Produtos/Roteiros, Cadastro de Ordens, Estados de calendário, Tela de Programação (Pereira, Librantz, 2013)**. Na Figura 7 pode-se observar a tela

orientada ao sequenciamento das operações. Como exemplo, já se tem uma ordem com sua respectiva seqüência de operações em função do tempo alocado no gráfico de Gantt.

Figura 7. Tela do Gráfico de Gantt com todas as ordens programadas



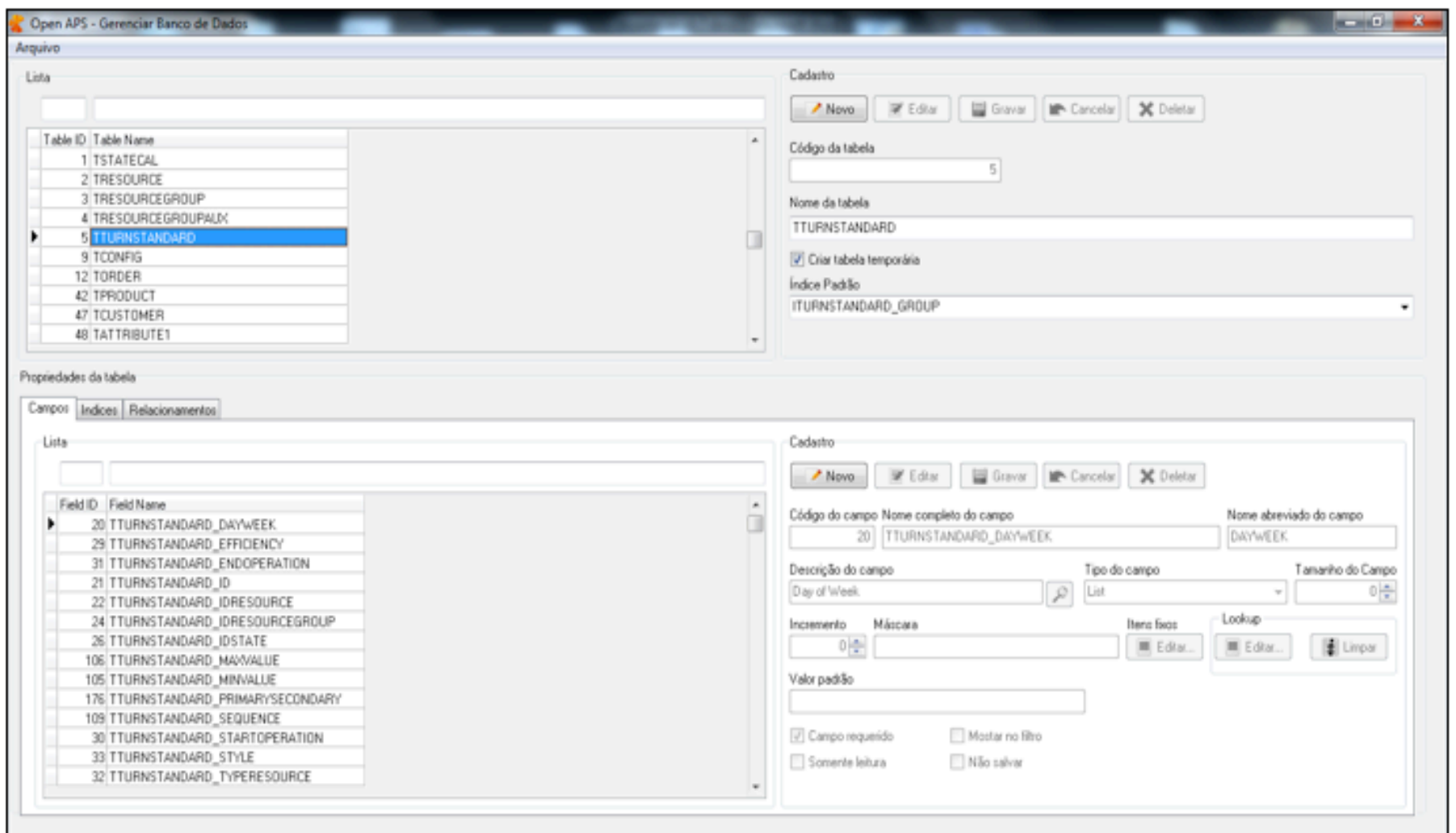
Fonte: Pereira e Librantz (2015)

Deve-se ressaltar que, além do sequenciamento manual, o software desenvolvido possibilita a utilização de duas técnicas distintas para o sequenciamento das ordens de produção, sendo a primeira baseada em regras de liberação e a segunda com a utilização de algoritmos genéticos, bem como a customização.

4.1. Personalização do software protótipo

Para permitir que cada empresa possa personalizar o software em função de suas necessidades, o software protótipo foi desenvolvido a partir de uma arquitetura que permite total personalização, como a criação de campos no banco de dados do software, como pode ser observado na Figura 8.

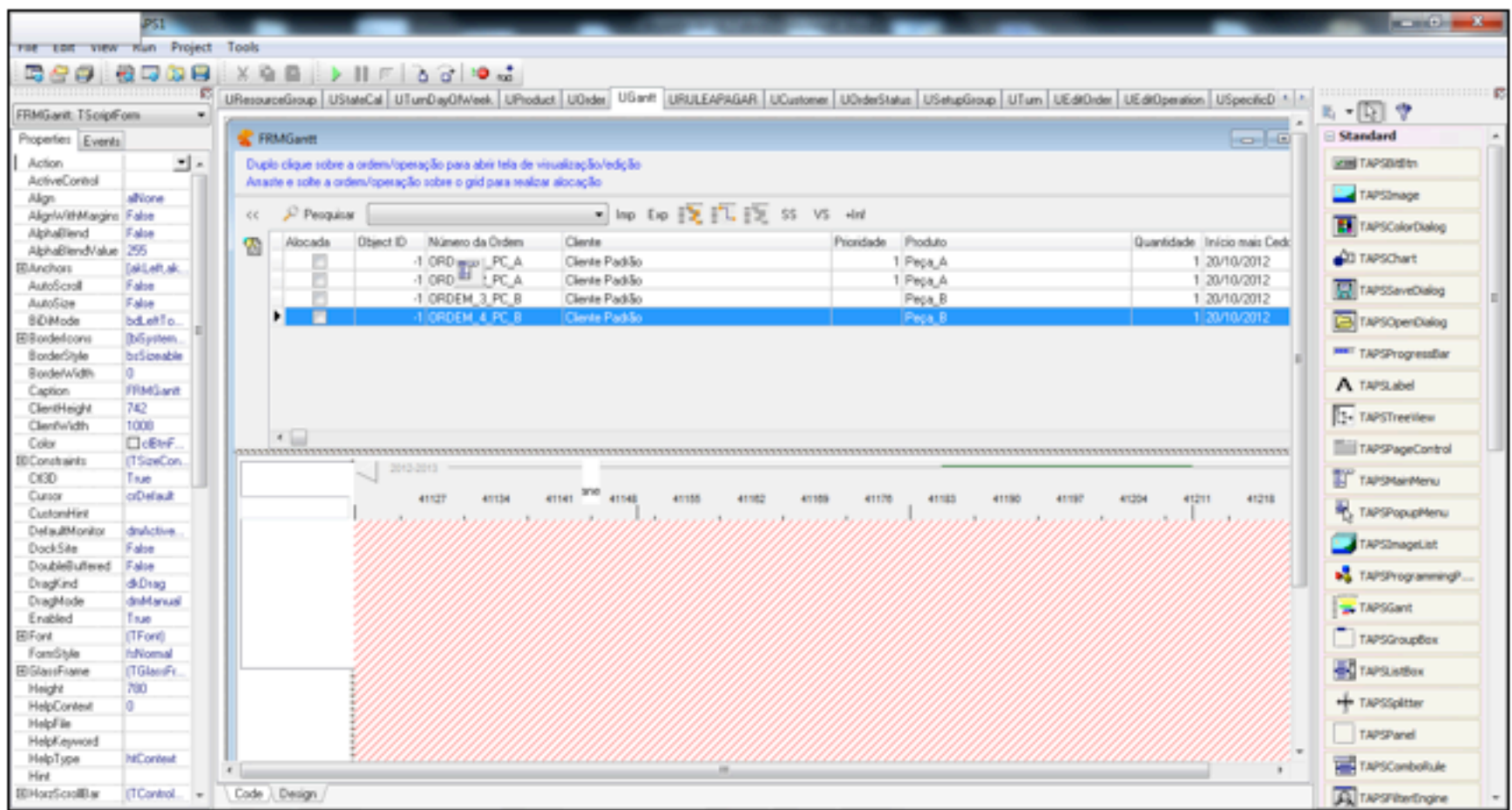
Figura 8. Janela de manutenção no banco de dados do software.



Fonte: o autor

Todas as interfaces com o usuário foram desenvolvidas com o próprio software protótipo o que permite o desenvolvimento de novas janelas, gráficos, etc. Esse ambiente de desenvolvimento possui as principais características da maioria dos principais ambientes de desenvolvimento de softwares comerciais. A Figura 9 apresenta o ambiente de desenvolvimento onde no lado direito está a paleta de componentes, ao centro o desenho da interface e a esquerda o editor de propriedades e eventos.

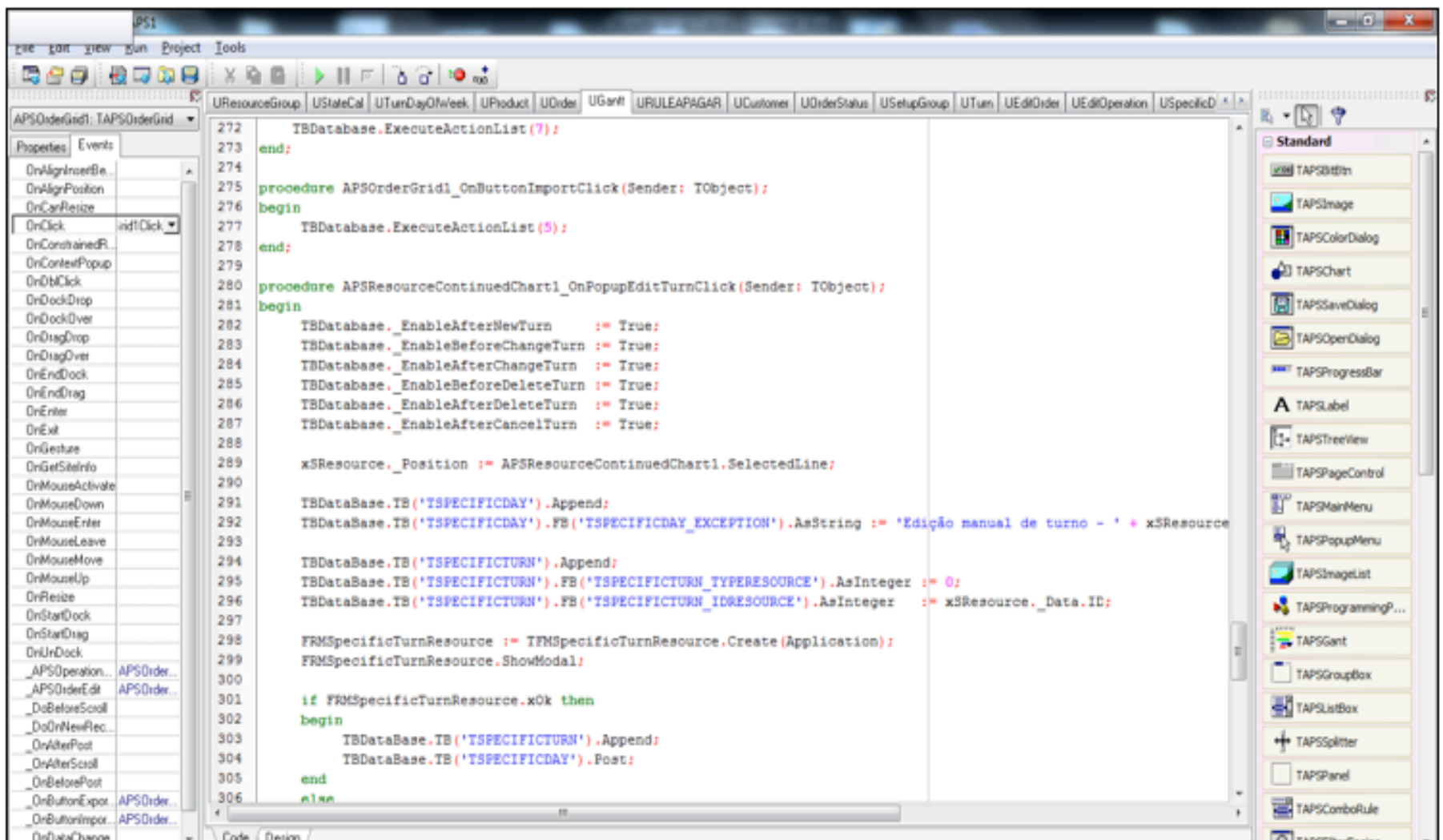
Figura 9. Ambiente de desenvolvimento do software protótipo.



Fonte: o autor

A Figura 10 apresenta um exemplo com o editor de código fonte do software protótipo.

Figura 10. Exemplo de código fonte no software protótipo.



Fonte: o autor

4.2. Estudo de caso

O software desenvolvido foi implantado em uma empresa que foi referenciada neste trabalho apenas como empresa A. Trata-se de uma empresa localizada no interior do estado de São Paulo, atuante no mercado de embalagens industriais e utensílios domésticos desde 1978. Possui hoje aproximadamente 900 funcionários divididos em duas estruturas organizacionais, sendo uma responsável pela injeção plástica e a outra pelo departamento de ferramentaria, que tem como objetivo desenvolver moldes de injeção plástica que serão utilizados para a fabricação de todas as linhas de produtos da empresa. O objeto de estudo foi a área de ferramentaria, uma vez que foi definida como diferencial estratégico da empresa em relação a seus concorrentes.

Por se tratar de uma ferramentaria cativa, ou seja, é uma ferramentaria que apenas fabrica moldes para uso interno, seu posicionamento estratégico está pautado em dois focos, sendo o primeiro para a melhoria contínua da qualidade e o segundo para redução nos prazos de entrega dos moldes. Para atingir estes objetivos estratégicos a gerência da ferramentaria tem direcionado seu foco no projeto do ferramental e no planejamento e controle da fabricação, reduzindo etapas e buscando ganhos de escala como a padronização de componentes e de projeto. A estrutura da ferramentaria apresentada no Quadro 2, conta com 22 funcionários trabalhando dois turnos e produzem aproximadamente 4000 horas mensalmente distribuídas entre a construção e manutenção das ferramentas.

Quadro 2 - Recursos e pessoal ocupado na ferramentaria.

Equipamento/Tipo de operação	Equipamentos	Funcionários
Projeto	3	3
Detalhamento 2D	1	1
Programador CAM	1	1
Centro de Usinagem	5	3
Eletroerosão Penetração	3	3
Torno CNC	2	2
Tridimensional	1	1
Retificas	3	3
Operações de bancada (Ajuste e polimento)	5	5

Fonte: Empresa A

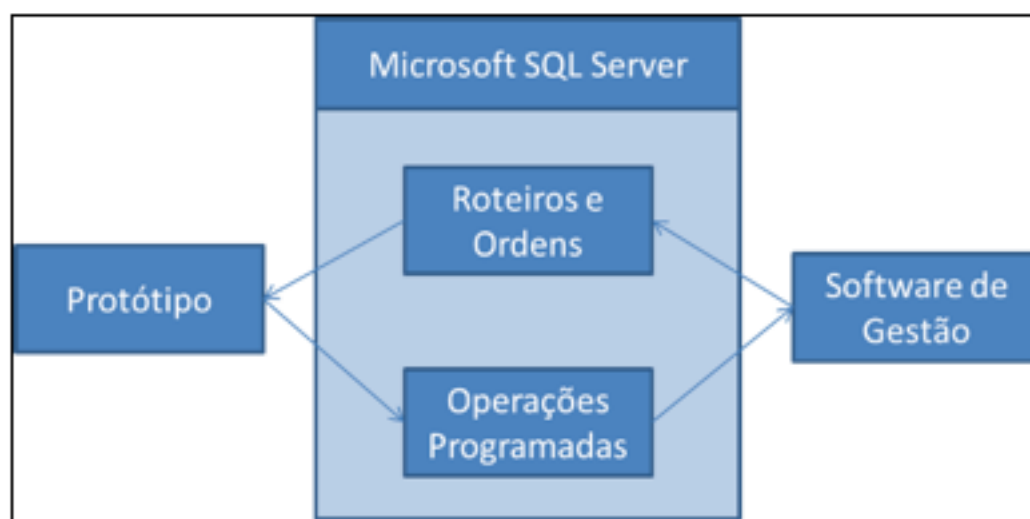
4.3. Ambiente de software e integração com o sistema de gestão

A ferramentaria possui um software de gestão o qual possui um módulo para a criação e manipulação da estrutura dos produtos (molde de injeção) e integração com as outras soluções

da empresa como as ferramentas CAD e CAM. Possui também um módulo de apontamento da produção através de coletores de dados espalhados pela fábrica, que permitem acompanhamento *on-line* de cada etapa de fabricação. Além da estrutura dos produtos, possui também a lista de operações e os recursos que podem executar cada operação com tempos destacados de *setup* e operação. Possibilita também a dependência entre ordens, ou seja, organização de montagens do molde. A integração com o software de gestão foi realizada com o recebimento das ordens e operações liberadas, ficando a responsabilidade de gerar os cenários e definir a programação da fábrica para o software protótipo que posteriormente devolve ao software de gestão as ordens e operações programadas.

A integração foi realizada diretamente através de tabelas compartilhadas no banco de dados Microsoft SQL Server®. Algumas tabelas são disponibilizadas com as informações do software de gestão para o software protótipo, que devolve em outra tabela as operações com informações de início e término além dos recursos que foram selecionados para executar a operação (Figura 11).

Figura 11. Integração do software protótipo ao sistema de gestão.



Fonte: o autor

5. Resultados obtidos após a implantação do software protótipo

Após decorrido dois anos do início dos trabalhos foi realizado um estudo dos benefícios da adoção do software na ferramentaria. Este estudo foi realizado pela gerência da ferramentaria que disponibilizou alguns dados para este trabalho. Seguem abaixo os principais pontos do estudo realizado:

- Redução do tempo gasto com a programação da fábrica, antes realizada com a Microsoft Excel®, além de trabalhoso não tinha informações atualizadas da fábrica ficando constantemente desatualizado.
- Priorização real dos trabalhos, pois cada molde dependendo da complexidade e número de cavidades pode ser composto por centenas de componentes o que dificulta ter uma visão onde cada componente está, o quanto já foi produzido e quanto tempo resta para sua conclusão. Com a adoção do software a mudança de prioridade do projeto é repassada a todos os componentes do molde.
- Previsibilidade de real entrega, pois a diretoria ou os clientes do segmento de embalagens exigiam prazos realistas o que era muito difícil com as ferramentas computacionais utilizadas anteriormente.
- Análise de diferentes cenários antes da efetivação da programação com visão global da fábrica.
- Atendimento da estratégia da empresa, redução do tempo para lançamentos de produtos ou

atendimento dos clientes nos prazos que desejavam. Este benefício foi conseguido pois a partir do momento em que era possível ter uma visão geral da ferramentaria, foi possível iniciar um trabalho de melhoria de processos.

- Redução de compras antecipadas, pois estão alinhadas à programação da ferramentaria.

A empresa estudada forneceu indicadores antes da implantação e após a implantação do software, permitindo uma visão mais detalhada dos ganhos obtidos que podem ser observados na Tabela 1.

Tabela 1. Indicadores da empresa A antes e após a implantação do software protótipo.

Indicador	Antes	Depois
Percentual dos moldes entregue com atraso	40%	8%
Percentual do custo que ultrapassa o orçamento	40%	-11%
Percentual de horas extras sobre o total de horas normais	10%	3%
Percentual gasto com horas extras sobre o valor da folha de pagamento	10%	6%

Fonte: Empresa A

Analisando os dados da Tabela 1, pode-se observar ganhos significativos na diminuição do atraso na entrega das ferramentas o que além de benefícios financeiros como a redução no pagamento de eventuais multas, permite também um aumento na confiabilidade na empresa em cumprir seus prazos com os clientes e a diretoria. Outro indicador que merece destaque foi a significativa melhora na precisão dos custos dos orçamentos, pois com uso do software de programação de produção associado ao sistema de apontamento permite comparar efetivamente o tempo gasto com o orçado e planejado. Evitando uma subestimação de tempos de fabricação, impactando diretamente em imprecisões de prazo e custo.

Os demais indicadores fornecidos pela gerência da empresa estão relacionados à redução no custo e na utilização de horas extras. O principal ganho foi a mudança na forma como eram utilizadas as horas extras que principalmente eram utilizadas para atender os prazos prometidos e pouco cumpridos. Atualmente são utilizadas de forma planejada como aumento de capacidade temporária em recursos gargalos.

6. Conclusão

Este trabalho apresentou um *framework* computacional para o desenvolvimento de software de programação de produção, que considerou condições reais de fabricação e permitiu o uso de diversas regras de liberação como carregamento paralelo, minimização de WIP e além da utilização de Algoritmos Genéticos. Com base nas especificações do *framework* foi desenvolvido um software protótipo que se mostrou demonstrou ser bastante flexível, pois permitiu que a especificação de recursos e operações sem a preocupação de quais alternativas de programação seriam utilizadas. Esta separação permitiu concentrar-se nas características do processo e restrições operacionais, como turnos, manutenções e quebras para posteriormente poder adotar uma ou outra forma de sequenciamento, podendo o usuário analisar diversos indicadores de cada cenário gerado para posteriormente decidir qual é o mais adequado para a empresa naquele momento.

O software permitiu o desenvolvimento de diversas regras de liberação de forma relativamente

simples, possibilitando também a criação de regras especiais ou personalizadas para atender a características específicas de um sistema produtivo.

O software foi implantado em uma empresa de injeção plástica no setor de fabricação de moldes e após a implantação forneceu resultados que apontaram ganhos significativos na redução de atraso na entrega, na redução de horas extras e na melhor visão dos itens em produção. Também foi possível à empresa desenvolver melhorias em seus processos e concentrar-se no projeto de ferramental, lançando produtos com mais rapidez, podendo dessa maneira focar em seus diferenciais competitivos. Trabalhos futuros incluem a aplicação deste modelo em problemas mais complexos de programação da produção, bem como o uso de outras técnicas metaheurísticas como regras de despacho.

Referências Bibliográficas

- Araújo, S. A.; Librantz, A. F. H.; Alves, W.A.L (2007). Uso de algoritmos genéticos em problemas de roteamento de redes de computadores. *Exacta*, 321-327.
- Brailsford, S. C., Potts, C. N., Smith, B. M (1998). Constraint satisfaction problems: Algorithms and applications. *European Journal of Operational Research*, 119, 557-58.
- Brucker, P. (2002) Scheduling and constraint propagation. *Discrete Applied Mathematics*, Osnabrück, 123(1), 227-256.
- Duin, C. W., Sluis, E. V. (2006), "On the complexity of adjacent resource scheduling", *Journal of Scheduling*, 9 (1), 49-62.
- Fernandes, F. C. F.; Filho, M. G.(2010). *Planejamento e Controle da Produção: dos fundamentos ao Essencial*. São Paulo, Editora Atlas.
- Framinan, J. M.; Ruiz, R. (2010). Architecture of manufacturing scheduling systems: Literature review and an integrated proposal. *European Journal Of Operational Research*, 205, 237-246.
- Hvolby, H.-H.; Steger-jensen, K. (2010). Technical and industrial issues of Advanced Planning and Scheduling (APS) system. *Computers in Industry*, 61, 845-851.
- Ichihara, J. A. (2002). O problema de programação de projetos com restrição de recursos (Resource-Constrained Project Scheduling Problem). In *Anais do XXII Encontro Nacional de Engenharia de Produção*, (pp. 1-8).
- Maineri, G. B.; Ronconi, D. P. (2001). Regras de despacho para a minimização do atraso total no ambiente flowshop flexível. *Gestão e Produção*, São Carlos, 17(4), 683-692.
- Morais, M. F.; Menegarde, J. K.; Cantiere, P. C. (2009). Regras de Prioridade e Critérios de Desempenho Adotados em Problemas de Programação da Produção em Ambientes Flow Shop. In *Anais do IV Encontro de Produção Científica e Tecnologia*, Campos Mourão (pp. 1-9).
- Novas, J. M.; Henning, G. P. (2010). Reactive scheduling framework based on domain knowledge and constraint programming. *Computer and Chemical Engineering*, 34,2129-2148.
- Pacheco, R. F.; Santororo, M. C. (2001) A adoção de modelos de scheduling no Brasil: deficiências do processo de escolha. *Gestão e Produção*, 8(2), 128-138.
- Pereira, V.O.; Librantz, A.F.H. (2015). Framework computacional para a programação da produção de gabaritos de montagem com restrição de adjacência. *Espacios*. Vol. 36 (Nº 23), pp. 21.
- Reis, J. (1996). O Scheduling em Inteligência Artificial : um resumo. Relatório Interno - Departamento de Ciências e Tecnologias da Informação - Instituto Superior de Ciências do Trabalho e da Empresa, Lisboa.
- Silva, B. J. V. D.; Morabito, R.; Yanasse, H. D. (2011). Programação da Produção de gabaritos de montagem com restrições de adjacência na indústria aeronáutica *Produção*, 21(1), 696-707.
- Tubino, D. F. (2009). *Planejamento e Controle da Produção: Teoria e Prática*. São Paulo, Editora

Atlas.

Turatti, R.; Marcantonio, R. (2009). A Importância da Utilização dos Sistemas APS no Alinhamento de Estratégias Operacionais. In Anais do SIMPOI (pp. 1-9).

Pereira, V. O., Librantz, A. F. H. (2013). Framework computacional para programação da produção com capacidade finita. (dissertação de mestrado), Universidade Nove de Julho, São Paulo. (por favor, verificar a citação)

Weglaz, J. ; Józefowska, J; Mika, M.; Waligóra, G. (2011). Project scheduling with finite or infinite number of activity processing modes – A survey. European Journal Of Operational Research, 208(3), 177-205.

Xing, L.-N.; Chen, Y.-W.; Yang, K.-W. (2009). Multi-objective flexible job shop schedule: Design and evaluation by simulation modeling. Applied Soft Computing, 9(1), 362-376.

Xu, J.; Xu, X.; Xie, S. Q (2011). Recent developments in Dual Resource Constrained (DRC) system research. European Journal Of Operational Research, 215(2), 309-318.

Zeballos, L. J. A (2010). Constraint programming approach to tool allocation and production scheduling in flexible manufacturing system. Robotics and Computer-Integrated Manufacturing, 26(6),725-743.

1. Universidade Nove de Julho – UNINOVE, Programa de Pós-Graduação em Informática e Gestão do Conhecimento (PPGI). Email: librantz@uninove.br

2. Universidade Nove de Julho – UNINOVE, Programa de Pós-Graduação em Informática e Gestão do Conhecimento (PPGI). Email: diascrg@uni9.pro.br

3. Universidade Nove de Julho – UNINOVE, Programa de Pós-Graduação em Engenharia de Produção (PPGEP). Email: valdecir@grv.com.br

Revista ESPACIOS. ISSN 0798 1015
Vol. 38 (Nº 08) Año 2017

[Índice]

[En caso de encontrar algún error en este website favor enviar email a webmaster]

©2017. revistaESPACIOS.com • Derechos Reservados