

Planejamento de Necessidade de Materiais (MRP): Desenvolvimento de software como apoio para cursos tecnológicos

MARCOS VINICIUS SABINO SILVA (Faculdade de Tecnologia Guarulhos)

marcos.silva312@fatec.sp.gov.br

RAFAEL OLIVEIRA RUFINI (Faculdade de Tecnologia Guarulhos)

rafael.rufini@fatec.sp.gov.br

JANE MARIA DOS SANTOS EBERSON (Faculdade de Tecnologia de Guarulhos)

jane.santos@fatec.sp.gov.br

RESUMO

Este trabalho tem como objetivo demonstrar o desenvolvimento de um software de Planejamento de Necessidade de Materiais (MRP), sendo esse uma ferramenta de ensino da teoria do sistema MRP no ensino superior de tecnologia. Para os requisitos abordados no desenvolvimento desse software foram considerados uma empresa fictícia de produção de computadores. Foi utilizado como suporte uma pesquisa bibliográfica, onde foi descrito três aspectos fundamentais para a realização desta pesquisa: Planejamento de Necessidade de Materiais (MRP), Engenharia de Software e a Gestão de Projetos. Este projeto de pesquisa utiliza também de uma metodologia experimental acerca do desenvolvimento prático de um MRP, abordando uma linguagem de programação, a modelagem de um banco de dados e os fundamentos da Engenharia de Software.

PALAVRAS-CHAVE: Planejamento de Recursos de Materiais. Engenharia de Software. Gestão de Projetos.

ABSTRACT

This article aims to demonstrate how to work a Material Resource Planning (MRP) software as a MRP higher theory teaching project. The requirements for the development of this software in view of a fictitious computer production company. A bibliographic research was used as a support, where three aspects were described: Resource Planning of Research Materials (MRP), Software Engineering for Materials Management. This research project uses an experimental methodology about the practical development of an MRP, approaching a programming language as well, a modeling of a database and the fundamentals of Software Engineering.

Keywords: *Material Resource Planning. Software engineering. Project Management*

1. INTRODUÇÃO

O tema abordado nesse projeto de pesquisa é o desenvolvimento de um software de MRP, em como a Gestão de Projetos e a Engenharia de Software auxiliam no desenvolvimento do mesmo, identificando quais são as dificuldades para o desenvolvimento de um software para um segmento de curso superior.

A produção de qualquer que seja o produto depende de um controle dos materiais, componentes e subcomponentes necessários para produzir um produto acabado, além de garantir um bom controle no estoque de materiais, a fim de que não ocorra falhas no fluxo dos dados sobre os materiais utilizados na produção, falhas, como: materiais podem não estar em quantidade ou especificação suficiente; o fornecedor não cumprir os prazos de entrega ou quantidades solicitadas; não conseguir rastrear um produto depois que ele entrou na produção e; não conseguir controlar o estoque após o fim da produção, é necessário ter uma boa gestão para entender os requisitos de negócio, garantir a produção e entrega dos produtos no prazo, e buscar sempre o melhor preço ofertado pelos fornecedores.

Considerando a criticidade de um sistema MRP, um software que o demonstre é de suma importância para o ensino da teoria sobre o assunto, pois ajuda a entender os conceitos básicos das regras de negócios pré-estabelecidas por uma organização, nesse caso baseadas nos requisitos de uma empresa fictícia de montagem de computadores.

As dificuldades do desenvolvimento do software MRP é de realizar a modelagem de software e criar a estrutura de dados do mesmo. Outras dificuldades de se desenvolver esse software estão relacionados aos problemas de requisitos de negócio e Engenharia de Software, não conhecer bem os requisitos funcionais e não-funcionais geram um problema de erro de escopo de projeto, e isso acaba gerando problemas de desenvolvimento e manutenção do software.

Dessa forma, o objetivo geral desta pesquisa é demonstrar o desenvolvimento de um software MRP com o intuito de auxiliar o ensino da teoria desse sistema em uma instituição de curso superior, baseado em uma empresa fictícia de computadores, propondo por fim melhorias no desenvolvimento de um software. A pesquisa também aponta como a Gestão de Projetos e a Engenharia de software podem ajudar a minimizar os impactos na dificuldade de desenvolvimento deste produto.

2. EMBASAMENTO TEÓRICO

2.1 Planejamento de Necessidades de Materiais (MRP)

Até a década de 60, as empresas realizavam os cálculos de planejamento de recursos manualmente, a fim de garantir que sempre teriam disponíveis os materiais corretos no tempo certo. Porém, com o avanço da tecnologia e o uso de computadores nas empresas, o trabalho de executar esses cálculos detalhados e demorados, tornou-se possível de virar um trabalho automatizado através do desenvolvimento de um software que realize esses cálculos. (COLADAWEB, 2022, s.p.)

Em 1970, surgiram os primeiros sistemas de Planejamento de Necessidades Materiais ou MRP (*Material Requirement Planning*). Esses sistemas permitiram realizar os mesmos cálculos de planejamento de recursos, agora feitos de maneira automatizada. Esses sistemas gerenciavam as listas de materiais e em conjunto com os dados de estoque do setor fabril.

Segundo Davis (2001), indústrias com mais de 10 milhões em vendas anuais, em sua maioria, necessitam de um sistema computadorizado para o controle da produção por causa da

velocidade que as empresas devem ter para reagir a constantes mudanças no sistema e do grande volume de materiais, suprimentos e componentes, que são partes de linhas sempre em expansão.

O MRP tem como finalidade auxiliar as empresas, em sua maioria empresas manufatureiras a planejar e controlar suas necessidades de recursos com o apoio de sistemas de informação computadorizados (DAVIS, 2001), dessa forma traz diversos benefícios para a empresa que decide implementá-lo, benefícios tais como: o processamento de muitos dados em pouco tempo devido a capacidade de calcular as necessidades de materiais com estruturas complexas com vários níveis de forma bem mais rápida do que executada em uma planilha; a redução de falhas humanas, pois o MRP baseia-se em dados cadastrados sobre o *lead time*, saldos de estoque e listas de materiais, devido a isso, as falhas humanas são reduzidas a quase zero, a menos que os usuários cadastrem dados errados (DEVECCHI, 2018).







2.2 Engenharia de Software

A Engenharia de Software reúne um conjunto de práticas e ferramentas que auxiliam o desenvolvimento de um software com qualidade. Com a evolução e incorporação dos softwares ao nosso cotidiano, a engenharia de software ganha cada vez mais importância, pois capacita os profissionais a criarem sistemas complexos com alta qualidade e dentro do prazo (PRESSMAN, 2011), ratificado por:

Efetivamente, não é algo simples conceituar e praticar a Engenharia de Software. Mas é necessário. Primeiramente, deve-se ter em mente que os processos de Engenharia de Software são diferentes, dependendo do tipo de software que se vai desenvolver. [...] dependendo do nível de conhecimento ou estabilidade dos requisitos, deve-se optar por um ou outro ciclo de vida, o qual será mais adequado ao tipo de desenvolvimento que se vai encontrar (WAZLAWICK 2013, p. 4).

Apesar de haver variações de nomenclatura ou classificação dependendo de cada autor, Sommerville (2011), define seis etapas fundamentais de desenvolvimento de sistemas, representadas na figura 1.

Figura 1 - As seis atividades fundamentais do desenvolvimento de sistemas

-  1. Desenvolvimento de requisitos.
-  2. Projeto de sistema.
-  3. Engenharia de subsistema.
-  4. Integração de sistema.
-  5. Teste de sistema.
-  6. Implantação de sistema.

Fonte: Adaptado de Sommerville, 2011

A importância da engenharia de requisitos é notória, uma vez que a detecção precoce de falhas minimiza a necessidade de retrabalho. Seguindo essa perspectiva:

A Engenharia de Requisitos define, sem dúvida, um dos mais importantes conjuntos de atividades a serem realizadas em projetos de desenvolvimento de software. Embora não garanta a qualidade dos produtos gerados, é um pré-requisito básico para que se

obtenha sucesso no desenvolvimento do sistema de informação. (COSTA, 2018, p. 212)

Dentro da engenharia de requisitos, é também necessário manter atenção nas regras de negócio, como definido por Dallavalle e Cazarini (2000), as regras de negócio são regras que permitem aproximar os procedimentos e dinâmicas da empresa com o software, e por meio da adaptação do software ao ambiente organizacional, trazer uma vantagem competitiva.

O Projeto de sistema está alinhado com a engenharia de requisitos, pois é a fase que os requisitos são analisados, a etapa de requisitos também trata de planejar a arquitetura do sistema, além de, compreender os componentes do sistema, e seus relacionamentos entre si (SOMMERVILLE, 2011).

A engenharia de subsistema consiste na etapa na qual será desenvolvido os componentes do sistema, configurando as ferramentas necessárias (tanto software, quanto hardware) e onde será realizado a definição das metodologias operacionais de desenvolvimento.

A etapa de Integração de sistema é a parte do processo na qual os componentes serão unidos, e pela primeira vez, é quando o sistema estará com suas propriedades funcionais em funcionamento.

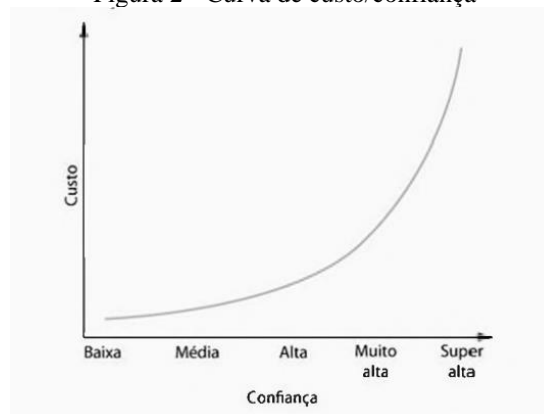
A etapa de testes do sistema:

É uma atividade extensiva, prolongada, em que os problemas são descobertos. As fases de engenharia de subsistema e de integração de sistema são reiniciadas para reparar esses problemas, ajustar o desempenho do sistema e implementar novos requisitos (SOMMERVILLE, 2011, p. 195).

A confiança do sistema, está configurada na etapa de testes e tem um destaque por ser um dos principais fundamentos da engenharia de software, trata-se da confiança depositada pelo usuário, que seu sistema computacional não falhará em situações de uso comum. Sommerville (2011) categoriza as falhas entre: falha de hardware (erros no projeto dos componentes físicos do sistema, defeitos de fabricação ou até mesmo fim da vida útil do componente), falha de software (erros no projeto do software, erros de implementação do software ou até erros de especificação de requisitos) e falhas operacionais (falhas ocasionadas pelo usuário que podem afetar o funcionamento geral do sistema).

Sommerville (2011) ainda evidência, com suporte da Figura 2, que sistemas com confiança muito alta e super alta tendem a escalar de custos (estes sendo sistemas críticos, onde uma falha pode ocasionar danos irreversíveis a empresa ou colocar vidas em perigo).

Figura 2 - Curva de custo/confiança



Fonte: Sommerville, 2011

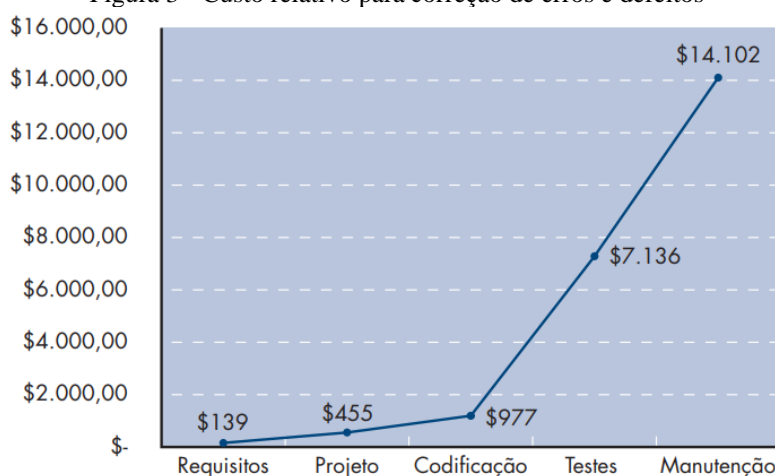
Identificar e tratar erros que possam vir a ocorrer é fundamental, portanto, o software deve possuir sistemas de tolerância a falhas, evitando erros de entradas e tratando o fluxo de dados, a fim de não comprometer a base de dados. “Prevenção e remoção de falhas não são

suficientes quando o sistema exige alta confiabilidade ou alta disponibilidade. Nesses casos o sistema deve ser construído usando técnicas de tolerância a falhas.” (WEBER, 2003, p. 8).

De acordo com Pressman (2011), os custos de falhas em um sistema são os custos evitáveis, que não teriam surgido se o sistema não apresentasse determinados erros, os custos de falhas podem ser divididos entre custos internos (antes da entrega do sistema) e externos (após a entrega do sistema ao cliente), sendo custos internos associados a custos de retrabalho para corrigir erros ou danos colaterais, e custos relacionados a detecção dos erros. Já custos de falhas externos, são associados a custos de suporte, reparos, substituições e devoluções do produto, o autor ainda menciona o custo não quantificável que surge do dano de reputação na entrega de um software de baixa qualidade.

O tempo também é essencial quando se trata de custos e correção de falhas, o custo de correção pode ser altamente minimizado se a detecção ocorrer ainda nas primeiras fases do projeto, como é apontado por Pressman (2011) na figura 3.

Figura 3 - Custo relativo para correção de erros e defeitos



Fonte: Adaptado de Boe *apud* Pressman, 2011

A implantação do sistema é a última etapa, onde os testes foram feitos e os defeitos corrigidos, nesta etapa, o sistema está finalizado e já pode ser usado pelo cliente (SOMMERVILLE, 2011).

3.3 Gestão de Projetos de Software

A gestão de projetos é a realizada por meio de experiências, ferramentas, conhecimentos e técnicas aplicadas nas atividades do projeto, com objetivo de alcançar ou exceder o que as partes interessadas (stakeholders) almejam (PMI *apud* BACK, 2002).

Figura 4 - Gestão de projetos



Fonte: VITADIGITAL, [21--]

Os três fundamentos que compõem a base da gestão de projetos são: promover um trabalho em equipe produtivo e colaborativo, ter uma boa administração dos recursos dos projetos e ter o cliente como foco (FILHO, 2009).

Sendo assim, a gestão de um **projeto de software** tem como objetivo organizar o processo de desenvolvimento. Essa atividade de gerenciamento tem relação com o tempo, custo e escopo desse projeto. Assim, através da gestão de **projeto de software** é possível:

- Prever o orçamento e mantê-lo;
- Contratar o número correto de colaboradores;
- Estabelecer prazos e trabalhar para cumpri-los;

Determinar cada etapa do desenvolvimento (MONITORATEC, 2021, s. p.).

Segundo Filho (2009), a gestão de projetos de software, visa garantir que o software seja entregue ao cliente no prazo determinado e que o software atenda as regras de negócio pré-estabelecidas pelo cliente. Além disso, o desenvolvimento do software deve ser orientado pela arquitetura, pois o desenvolvimento deve ser adaptado para as especificações de cada sistema.

É também necessário, um documento de plano de projeto, onde é definido e listado os marcos do projeto, esse registro deve conter também os prazos e as atividades a serem cumpridas. É natural aqueles que desenvolvem sistemas comecem a desenvolver antes de entender por completo os requisitos do cliente, Filho (2009) ainda aponta que, cerca de 40% das falhas em projetos são causadas por erros de planejamento

3. DESENVOLVIMENTO DA TEMÁTICA

A técnica de pesquisa utilizadas neste trabalho foi a pesquisa bibliográfica, pois a pesquisa é baseada em materiais bibliográficos e utiliza as referências como apoio e embasamento do projeto. A pesquisa também utilizou uma observação direta intensiva, com a utilização do recurso de entrevistas com especialistas na área MRP e produção industrial.

Quanto ao objetivo de esclarecer as principais dificuldades de desenvolver um Software MRP, pode ser citado a dificuldade no processo de modelagem de software, o que impacta diretamente no tempo que foi utilizado para que o desenvolvimento deste software foi concluído, pois a modelagem de software é responsável por facilitar o entendimento do sistema, sendo assim desenvolver o software sem uma boa modelagem torna o processo muito difícil.

As ferramentas escolhidas para o desenvolvimento desse software foram o Ambiente de Desenvolvimento Integrado (IDE) Visual Studio devido a familiaridade das partes envolvidas (Stakeholders) e o fornecimento gratuito e otimizado da IDE por parte da instituição de ensino. A linguagem de programação escolhida para o desenvolvimento foi o C# pelos motivos de fácil aprendizado por parte da equipe de desenvolvimento do software e do bom desempenho da linguagem para atender os requisitos funcionais e não-funcionais do cliente. Para a modelagem e desenvolvimento do banco de dados foi escolhido o MySQL pois os recursos deste software satisfazem a armazenagem de dados de forma segura e prática. As ferramentas de versionamento de código e o repositório utilizados para garantir a qualidade, distribuição e uma atualização segura do código foram o Git e o GitHub, levando em consideração o fácil acesso e ao conhecimento dos desenvolvedores nessas ferramentas.

O software de Planejamento de Necessidades de Materiais foi desenvolvido pensando nas principais funcionalidades que um aluno que estuda produção industrial, logística, gestão empresarial necessita aprender para desenvolver habilidades de gerenciamento de estoque, cálculo do plano mestre de produção, montagem da estrutura da árvore de produto, cálculo das necessidades líquidas de componentes, gestão de *Lead Time*, emissão da ordem de compra e

requisição de compra de componentes para os diversos fornecedores de uma indústria que tem como função produzir computadores dos mais diversos modelos.

O software tem várias tarefas atribuídas, através dele são realizados diversos cálculos que irão gerenciar o estoque a fim de que na hora certa de produção do produto acabado, não haja falta de nenhum componente, para que não ocorra alguma paralisação ou defeito de produção na empresa, o que geraria prejuízos e descontentamento da parte dos clientes.

A modelagem do Software foi construída após os levantamentos dos requisitos necessários para que o usuário final, seja capaz de gerenciar toda a produção de computadores, operando todas as funções do software com baixo tempo de aprendizado e com facilidade nas atividades. No levantamento de requisitos houve diversas dificuldades devido ao fato de falta de conhecimento em saber quais eram as principais atividades que o software deveria executar, não havia uma ideia clara pela parte da equipe de desenvolvedores, quais programas deveriam constituir o software, quais informações sobre produto deveriam ser armazenadas, qual seria a forma correta de realizar o gerenciamento de estoque, como seria desenvolvida a interface para que o usuário cadastrasse a árvore de produto, e como seriam realizados os cálculos que gerariam a ordem de compra ou de produção dos produtos acabados.

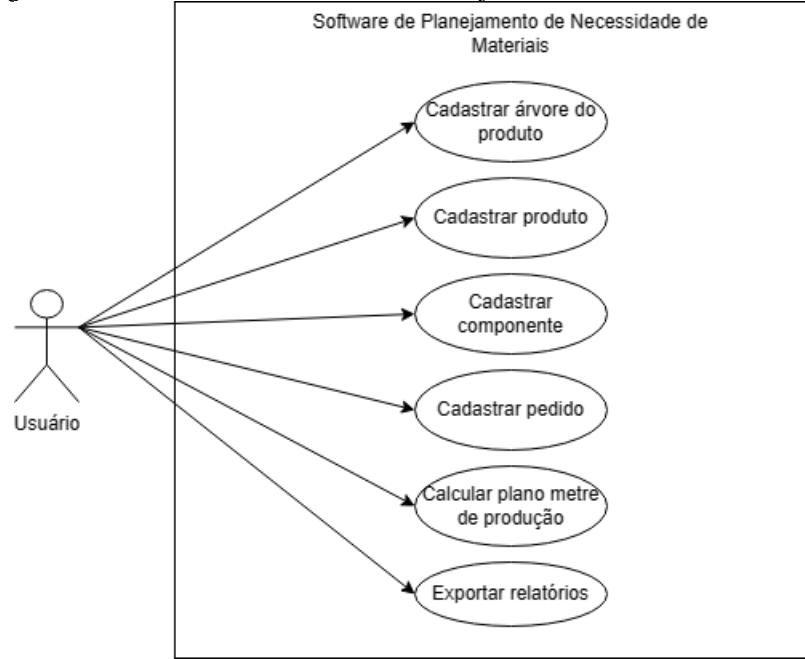
O software de planejamento de necessidade de materiais (MRP), reduz os riscos dos maiores problemas no planejamento da produção: A parada de produção por falta de material, por imprecisão na compra de materiais que formam o produto final, devido a compra no prazo errado, quando se faz a compra de componentes e subcomponentes sem previsão de demanda, e por estar com o saldo no estoque sempre incorreto e desatualizado; O prejuízo da empresa pode ser também por excesso de estoque, quando componentes e subcomponentes são comprados de maneira excessiva, o que resulta em um fluxo de estoque parado, e esse fluxo de estoque parado causa prejuízo no caixa da empresa, outro motivo para o excesso no estoque de matérias-primas é a compra de materiais errados, quando compra excessivamente uma quantidade de material, mas acaba comprando de forma errada, e quando percebe-se que comprou errado, a produção já está parada e causando prejuízos financeiros à empresa.

O desenvolvimento de software MRP, apresentou dificuldades no processo de modelagem do software, compreender quais eram os principais requisitos funcionais e como funciona o processo de negócio do software, levou muito tempo e gerou inúmeras versões com defeitos. O processo foi facilitado quando foi consultado professores que compreendiam a regra de negócio e o funcionamento de software, foram entrevistados professores que entendiam o funcionamento do software e da área de produção industrial, que orientaram como modelar e quais as principais funcionalidades que deveriam ser desenvolvidas.

Após a realização da entrevista, a dificuldade que havia sido encontrada em modelar o processo de execução do software foi reduzida, e foi possível concluir o levantamento de requisitos e o processo da estrutura de dados do sistema. O software foi modelado da seguinte maneira: foram construídos módulos de cadastro de produtos e componentes, esses serão inseridos em módulos de estoque; o usuário poderá cadastrar fornecedores e associar os componentes que estes poderão fornecer; o funcionário deve cadastrar uma árvore do produto, e após isso esse produto será cadastrado no MPS (Plano Mestre de Produção) a fim de gerar uma lista que indica o que será produzido; Com o plano mestre de produção é possível calcular a necessidade líquida de componentes que formarão esse produto.

Com base na análise do sistema foram construídos o diagrama de contexto (modelagem do software) conforme figura 5.

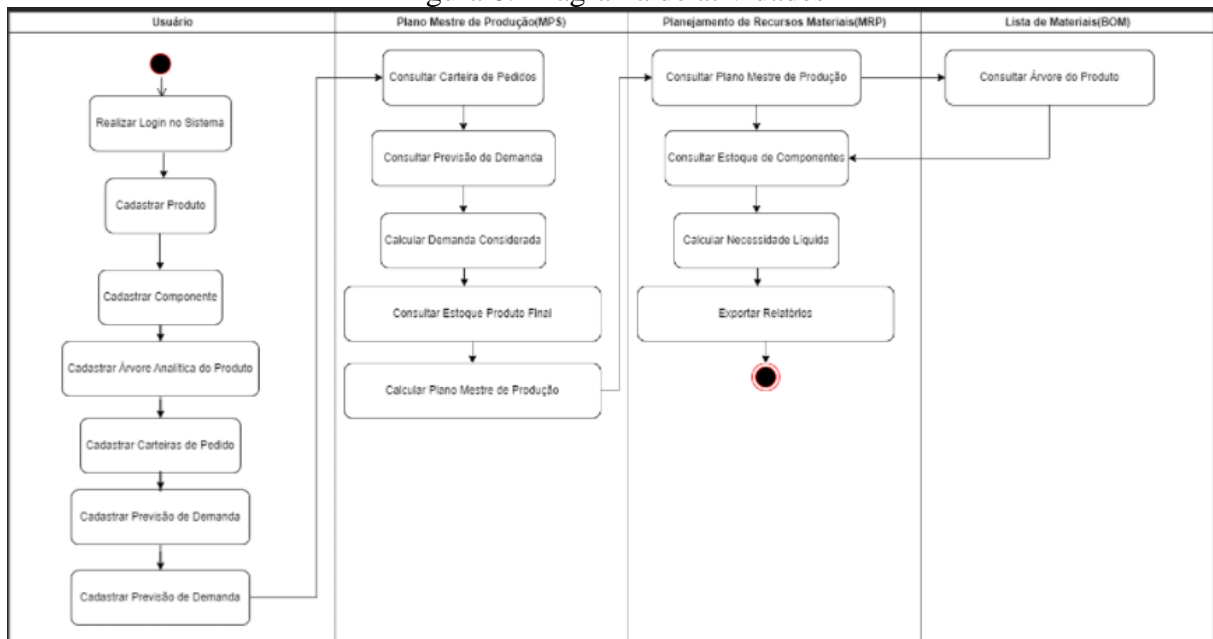
Figura 5 : Diagrama de Casos de Uso de Contexto – Planejamento de Necessidade de Materiais



Fonte: Desenvolvido pelos autores

Podemos observar na Figura 6, o diagrama de atividades, com o objetivo de melhor exemplificar o software, e proporcionando um panorama acerca de seu funcionamento.

Figura 6: Diagrama de atividades

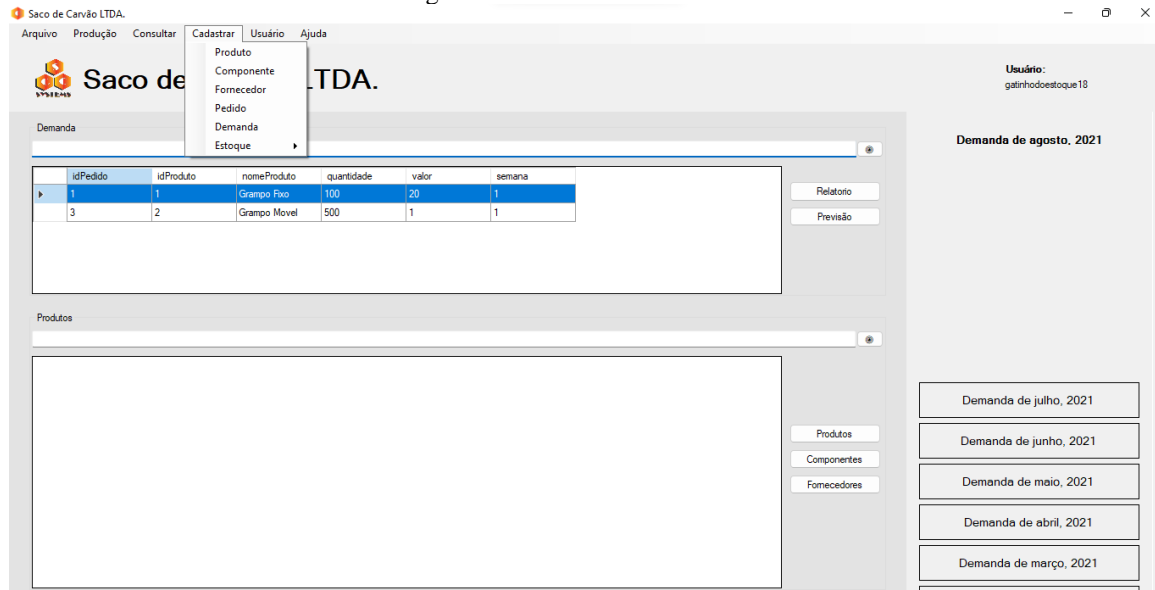


Fonte: Desenvolvido pelos autores

4. RESULTADOS E DISCUSSÃO

O software MRP deste projeto, conforme Figura 7, é capaz de calcular o plano mestre de produção e o planejamento de necessidade de materiais, de uma linha de produção de computadores. Apesar de ter algumas dificuldades para se aprender a utilizar o software, ele consegue realizar todos os processos de cálculo na produção industrial e armazenar os dados que serão utilizados posteriormente para fabricar os produtos.

Figura 7 : Interface atual do MRP



Fonte: Desenvolvido pelos autores.

A figura 8 demonstra como é realizado o cálculo de plano mestre de produção. O usuário deve selecionar o produto que vai ser produzido, e assim o software busca as informações da carteira de pedidos, previsão de demanda e seleciona a maior entre elas. O MRP também busca as informações de estoque atual, e baseado no cálculo de subtração entre o estoque atual e a demanda considerada, o sistema cadastra a quantidade necessária a ser produzida do produto selecionado.

Figura 8 : Calculando o Plano Mestre de Produção

Fonte : Desenvolvido pelos autores.

A figura 9 demonstra a realização dos cálculos de necessidades líquidas de componentes que constituirão a produção de um produto acabado. Nessa tela, o usuário deve selecionar o componente que forma a lista de materiais do produto, e após selecionado o sistema busca informações de estoque e informações de quantidade que devem ser produzidas, e em qual semana devem ser produzidas. O software também recupera informações de quantidade de produtos que foram emitidas as ordens de compra, e realiza a adição no estoque.

Figura 9 : Calculando o Planejamento de Recurso de Materiais

Cadastrar Produção

Nome Componente: Celeron N3060
Semana: 6

Id Árvore do Produto: 1
Necessidade Bruta: 0
Estoque Disponível: 200
Requisição Ordem Planejada: 0
Liberação de Ordens: 400

Estoque Produtos

idEstoqueProduto	idProduto	modeloProduto	qtdeAtualEstoque	estoqueSeguranca	leadTime	lote
1	1	Grampo Fixo	0	0	1	0
2	2	Grampo Moveel	500	100	1	50

Item

idMRP	idComponente	nome	necBruta	estoqueDisp	recOrdensPlan	libDeOrdens	sema
2	1	Celeron N3060	200	0	0	0	1
3	1	Celeron N3060	200	0	0	0	6

Cancelar Cadastrar

Fonte : Desenvolvido pelos autores.

5. CONSIDERAÇÕES FINAIS

Esse trabalho pretendeu demonstrar as dificuldades encontradas no processo de desenvolvimento de software MRP. Após a conclusão do projeto, é relatado que o processo de modelagem de software se mostrou mais difícil do que a parte de codificação do software.

Dentro de um período de 2 anos e meio, foi realizado o desenvolvimento deste software MRP, com base na entrevista e auxílio de professores que compreendem o funcionamento do processo do software MRP e o processo de produção industrial.

Para o desenvolvimento deste MRP foram utilizadas a linguagem de programação C#, o framework .NET, a IDE Visual Studio, o gerenciador de banco de dados MySQL Workbench, a ferramenta de versionamento de código Git e a plataforma de hospedagem de código GitHub.

As maiores dificuldades encontradas para a realização do projeto, foi a modelagem de software, o levantamento de requisitos, e ter uma visão clara do funcionamento do software. Essas dificuldades foram contornadas através de entrevistas e ajuda de professores que compreendiam o funcionamento do processo do software MRP e compreendiam o funcionamento da produção industrial.

Fato importante a se pontuar é, a necessidade de adaptação repentina e inesperada, em decorrência da pandemia do Covid-19, iniciada em março de 2020. Esse acontecimento forçou a uma readaptação no processo de desenvolvimento e comunicação do projeto que necessitou ser via Internet, além das necessidades de equipamento e adaptações.

Concluimos então, que apesar das dificuldades encontradas no desenvolvimento deste software MRP, é possível o desenvolvimento de um software dessa complexidade mesmo com um aprendizado de codificação limitado, basta ter uma clareza na modelagem de software e entender o processo de funcionamento do software desenvolvido.

REFERÊNCIAS

BACK, Reginaldo Salvato. Um método para definição de indicadores de desempenho aplicado à gestão de projetos de sistemas de informação. 2002.

COLADAWEB. **MRP E MRP II – CONCEITOS E APLICAÇÃO**. Disponível em: <https://www.coladaweb.com/administracao/mrp>. Acesso em: 14 dez. 2022

COSTA, E. C. da. A IMPORTÂNCIA DA ENGENHARIA DE REQUISITOS NO PROCESSO DE DESENVOLVIMENTO DE SISTEMAS DE INFORMAÇÃO. Revista Interface Tecnológica, [S. l.], v. 15, n. 1, p. 203-214, 2018. DOI: 10.31510/infa.v15i1.322. Disponível em: <https://revista.fatectq.edu.br/index.php/interfacetecnologica/article/view/322>. Acesso em: 24 nov. 2021.

DALLAVALLE, Silvia Inês; CAZARINI, Edson Walmir. Regras de Negócio, um fator chave de sucesso no processo de desenvolvimento de sistemas de informação. **Anais do XX ENEGEP-Encontro Nacional de Engenharia de Produção. São Paulo, 2000.**

DAVIS, MARK M. Fundamentos da administração da produção, Porto Alegre: Bookman Editora, 2001.

DEVECCHI, Thabata. **SISTEMA MRP: O QUE É, COMO FUNCIONA E QUAIS OS BENEFÍCIOS PARA SUA INDÚSTRIA**. [S. l.], 2018. Disponível em: <https://qserp.com.br/mrp-o-que-e-como-funciona-e-quais-os-beneficios-para-sua-industria/>. Acesso em: 27 nov. 2021.

FILHO, Antonio Mendes da Silva. **Gestão de Projetos de Software**. [S. l.], 2009. Disponível em: <https://www.devmedia.com.br/gestao-de-projetos-de-software/9143>. Acesso em: 27 nov. 2021.

MONITORA. **Afinal, o que é preciso para criar o seu projeto de software?**. [S. l.], 6 abr. 2021. Disponível em: <https://www.monitoratec.com.br/blog/projeto-de-software/>. Acesso em: 27 nov. 2021.

MONITORATEC. **Processo de desenvolvimento de software**. [S. l.], 8 jan. 2021. Disponível em: <https://www.monitoratec.com.br/blog/processo-de-desenvolvimento-de-software/>. Acesso em: 19 set. 2022.

PRESSMAN, Roger S. **Engenharia de software: uma abordagem profissional**. 7. ed. Porto Alegre: AMGH, 2011. E-book.

SLACK, Nigel; BRANDON-JONES Alistair; JOHNSTON Robert. Administração da produção. 8. Ed. São Paulo: Editora Atlas, 1999.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. Ed. São Paulo: Pearson, 2011. E-book.

VITADIGITAL. **Gestão de Projetos**. [S. l.], [21--]. Disponível em: <https://vitadigital.com.br/solucoes/gestao-de-projetos/>. Acesso em: 27 nov. 2021.

WAZLAWICK, Raul Sidnei. Engenharia de software: conceitos e práticas. 1. ed. Rio de Janeiro: Elsevier, 2013.

WEBER, Taisy Silva. Tolerância a falhas: conceitos e exemplos. Apostila do Programa de Pós-Graduação Instituto de Informática UFRGS. Porto Alegre, p. 24, 2003. Disponível em: <http://www.inf.ufrgs.br/~taisy/disciplinas/textos/ConceitosDependabilidade.PDF>. Acesso em 24 nov. 2021.

"O conteúdo expresso no trabalho é de inteira responsabilidade do(s) autor(es)."