

TÉCNICA DE DESENVOLVIMENTO ÁGIL PARA PEQUENAS APLICAÇÕES E OS IMPACTOS POSITIVOS DA IMPLANTAÇÃO EM MICROS EMPRESAS

Bruno Gleidson dos Santos

Mestrando em Ciência da Computação pela Universidade Federal de Pernambuco - UFPE.

bgs@cin.ufpe.br

Paulo Romero Martins Maciel

Pós-Doutorado em Ciências Exatas e da Terra pela Duke University - DUKE

prmm@cin.ufpe.br

RESUMO

O uso de sistema para controle de empresas de pequeno porte tem se mostrado interessante e oferece diversas vantagens em relação ao sistema manual comumente utilizado (anotações em papel). O desenvolvimento da aplicação ocorreu em 2011 e sua implantação aconteceu no mesmo ano, houve aumento significativo nas vendas da empresa, melhorando também sua organização em termos de produtos prontos para serem vendidos como também os pedidos de novas mercadorias e trazendo agilidade no atendimento. A construção do software foi realizada no estilo Agile como será apresentado neste trabalho. Os resultados apontam que na empresa houve aumento significativo na lucratividade em relação ao antigo sistema (blocos de papel). Concluindo-se assim, que o uso do sistema automatizado tem se mostrado eficiente e promissor para o fim a que se destina. O objetivo principal é mostrar um processo ágil para pequenas aplicações.

Palavras-Chave: Extreme Programming XP, Eclipse IDE, UML.

ABSTRACT

The use of small business control system has shown to be interesting and offers several advantages over the commonly used manual system (paper notes). The development of the application occurred in 2011 and its implementation occurred in the same year, resulting in a significant increase in the company's sales, also improving its organization with products ready to be sold as well as the requests for new goods, bringing agility to the service. The building of the software was performed in Agile style, as it will be presented in this work. The results indicate that the company had a significant increase in profitability in relation to the old system (paper blocks). It was possible to conclude that the use of automated system has proved to be efficient and promising for its intended purpose. The main objective is to show an agile process for small applications.

Keywords: Extreme Programming XP, Eclipse IDE, UML.

1 INTRODUÇÃO

Para Stair (1998) e Laudon e Laudon (2004), um sistema de informação pode ser definido como um conjunto de componentes inter-relacionados trabalhando juntos para coletar, recuperar, processar, armazenar e distribuir informações com a finalidade de facilitar o planejamento, o controle, a coordenação, a análise e o processo decisório em organizações. Um sistema de informação é formado por um conjunto de partes assim como uma empresa porém, uma das principais funções do software é dar suporte de forma automatizada para os processos organizacionais em especial às micro e pequenas empresas que são fontes de renda e que geram uma alta gama de empregos.

As Micro e Pequenas Empresas se constituem em uma importante solução de geração de trabalho e renda no Brasil e no mundo. De acordo com SEBRAE-SP (2007), o Brasil conta com 5,1 milhões de empresas. Dessas, 98% são micros e pequenas organizações e, correspondem mais de dois terços (67%) das ocupações dos locais de trabalho do setor privado, e por cerca de 20% do PIB. Diante dessa potencial alternativo de informatização de micro e pequenas empresas para auxiliar e alinhar meta a produção fazendo com sejam cada vez mais competitivas e possam sobreviver de forma organizada e cada vez mais descartando o uso de sistemas manuais.

Trabalhos que antes eram normalmente executados de forma manual estão sendo hoje processados por softwares específicos. Softwares podem ser aplicados a qualquer situação para especificar procedimentos (PRESSMAN, 1995). Conforme Sommerville (2003), “software são programas de computador e a documentação associada. Produtos de software podem ser desenvolvidos para um cliente específico ou para o mercado”.

Um sistema de controle de estoque foi desenvolvido com a capacidade de controlar e gerenciar pequenas empresas do varejo. A metodologia utilizada para realização do software básico foi feita através de pesquisa bibliográfica por meio de livros e artigos científicos usando também a Programação Extrema XP em conjunto com as ferramentas Eclipse IDE, linguagem Java J2SE, PostgreSQL, iReport, DBDesigner, UML com a ferramenta Astah Community. O presente trabalho tem como objetivo principal apresentar uma técnica de desenvolvimento ágil para sistema de pequeno porte e relatar os pontos positivos da implantação de modo geral. Por sigilo a empresa onde houve a implantação não terá seu nome divulgado, será chamada de empresa X.

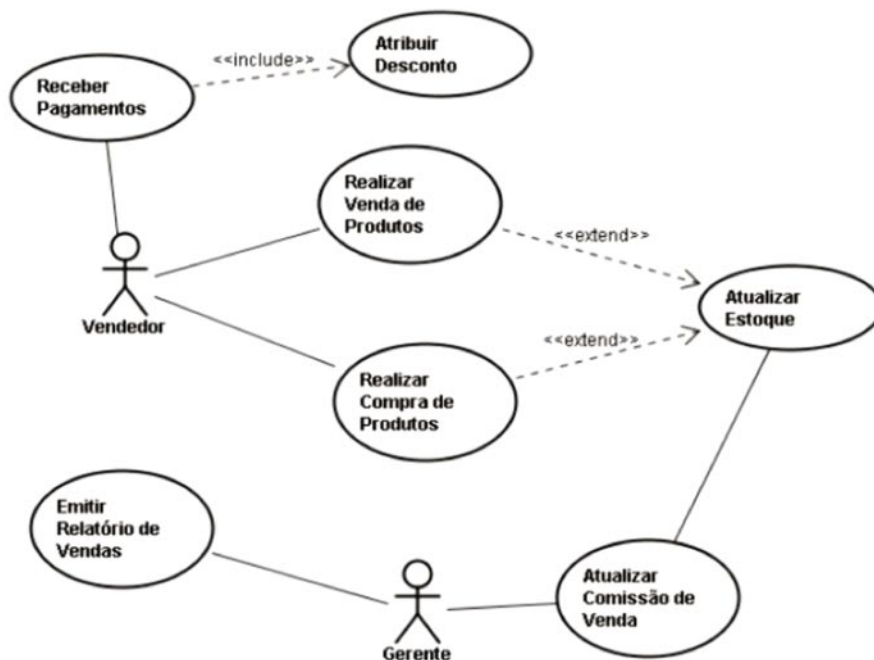
2 UML (Unified Modeling Language)

A *UML – Unified Modeling Language* ou Linguagem de Modelagem unificada é uma linguagem de modelagem de propósito geral que pode ser aplicada a todos os domínios de aplicação. Essa linguagem tornou-se, nos últimos anos, a linguagem padrão de modelagem adorada internacionalmente pela indústria de engenharia de *software*. Deve ficar bem claro, porém, que a *UML* não é uma linguagem de programação, e sim uma linguagem de modelagem, uma notação, cujo objetivo é auxiliar os engenheiros de *software* a definirem as características do sistema, tais como seus requisitos, seu comportamento, sua estrutura lógica e a dinâmica de seus processos (GUEDES, 2009).

Diagrama de Caso de Uso

O diagrama de caso de uso é o diagrama mais geral e informal da *UML*, utilizado normalmente nas fases de levantamento e análise de requisitos do sistema, embora venha a ser consultado durante todo o processo de modelagem e possa servir de base para outros diagramas (GUEDES, 2009). Inicialmente a modelagem do sistema se dá por meio do diagrama de caso de uso o qual dá uma visão de como será a interação do homem e aplicação, a figura 01 mostra parte da modelagem.

Figura 01: Diagrama de Caso de Uso.



Fonte: <https://armandohage.wordpress.com/category/ufpa-espdesenvinternet-analise-e-projeto-de-sistemas/>

Segundo Furlan (1998) a interação em caso de uso é quando o ator comunica-se com o sistema através do envio e recebimento de mensagens, sendo que um caso de uso é sempre iniciado a partir do momento que um ator envia sua mensagem (estimulo).

Diagrama de Classes

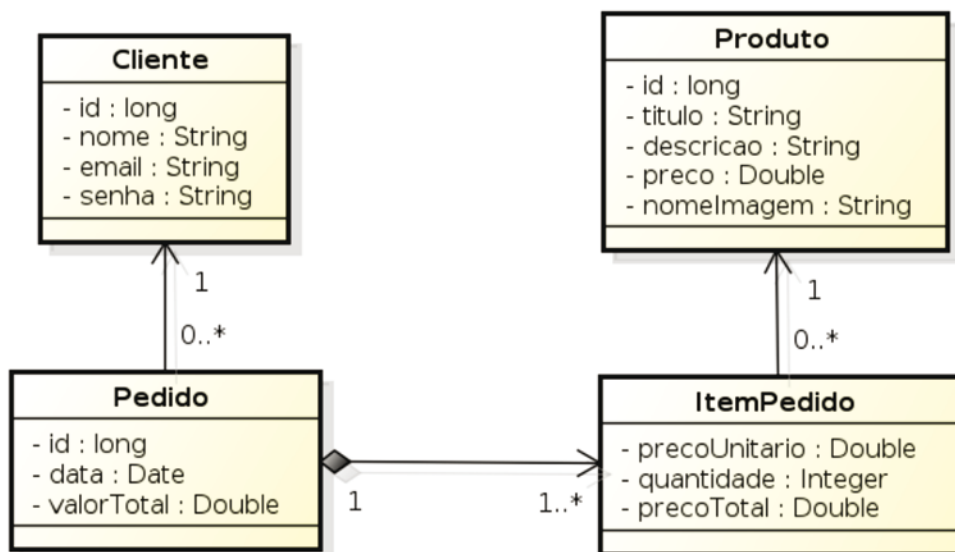
O diagrama de classes é provavelmente o mais utilizado e é um dos mais importantes da *UML*. Serve de apoio para a maioria dos demais diagramas, funciona como uma planta arquitetônica do sistema (GUEDES, 2009).

Trata-se de uma estrutura lógica estática em uma superfície de duas dimensões mostrando uma coleção de elementos declarativos de modelos, como classes, tipos e seus respectivos conteúdos e relações (FURLAN, 1998).

O diagrama de classe é essencial na construção de um projeto de software é o diagrama visto como principal para o desenvolvimento de sistemas.

A Figura 02 ilustra as classes com seus respectivos atributos os quais compõem o sistema. A classe cliente é composta por quatro atributos, para exemplificar apenas atributos estão expostos no diagrama a baixo e cada classe possui suas respectivas ligações entre cada parte que compõe o sistema estando em falta apenas suas operações (métodos).

Figura 02: Diagrama de Classe.



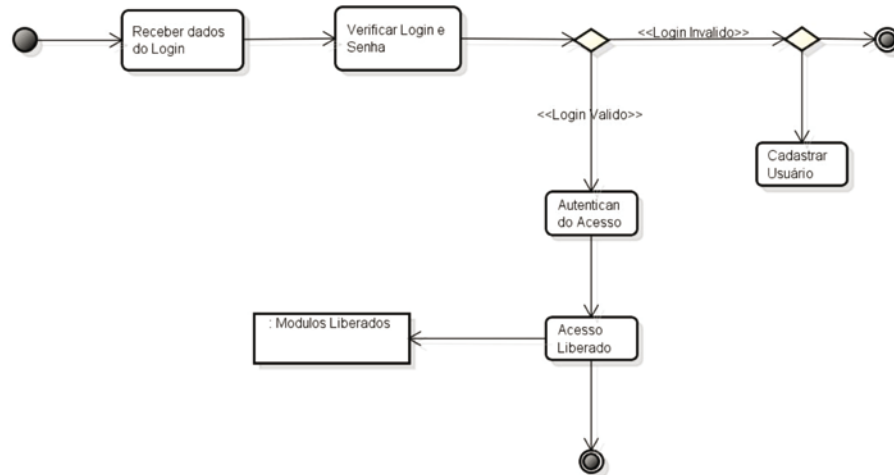
Fonte: Autor (2017).

De acordo com Furlan (1998), o diagrama de classe é a essência da UML resultado de uma combinação de diagramas propostos pela OMT (*Object-Modeling Technique*) Técnica de Modelagem de Objetos, Booch e vários outros métodos.

Diagrama de Atividade

O diagrama de estado foi introduzido por Rumbaugh, utilizado por Booch e adotado posteriormente na UML. Uma característica particularmente valiosa do enfoque é sua habilidade em generalizar estados que permitem fatorar transições comuns (FURLAN, 1998). A figura 03 mostra as atividades as quais se deve percorrer para que o sistema seja acessado com sucesso caso contrário um erro de login ou senha inválida será exibida para o usuário do sistema, caso o usuário faça parte equipe e não possui conta o administrador será informado para cadastrar o novo usuário no banco de dados da aplicação.

Figura 03: Diagrama de Atividade.



Fonte: Autor (2015).

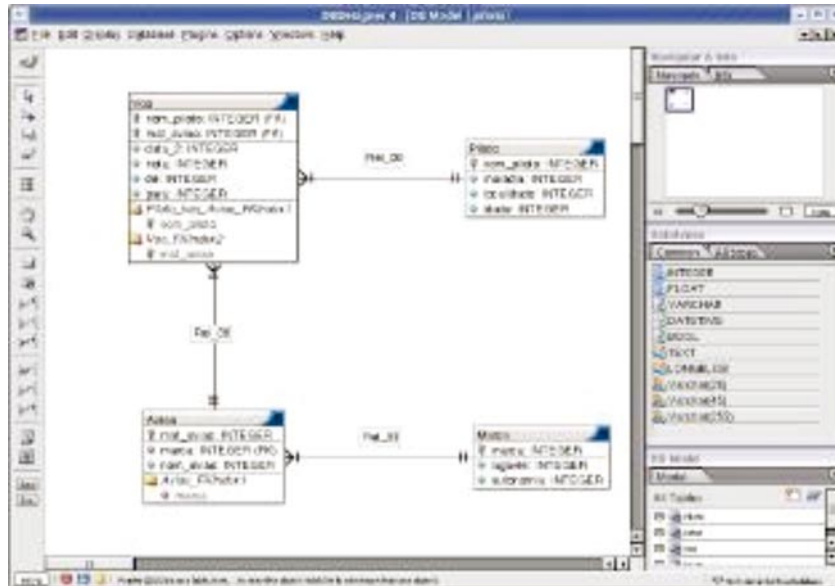
Como mencionado acima para exemplificar a Figura 03 mostra as atividades em sequência na validação do novo membro através das informações fornecidas pelo usuário para obter acesso aos módulos da aplicação.

2.2 Modelagem do banco de dados

“O modelo entidade-relacionamento (E-R) tem por base a percepção de que o mundo real é formado por um conjunto de objetos chamados entidades e pelo conjunto dos relacionamentos

entre esses objetos.” (SILBERSCHATZ; KORTH; SUDARSHAN, 1999, p. 21). Para modelar as tabelas do banco de dados foi utilizada a ferramenta DB Design como mostra a figura 04.

Figura 04: DB Design.



Fonte: <http://www.ebah.com.br/content/ABAAABBJoAD/5457-banco-dados>

A modelagem funciona como um projeto que será executado no banco de dados para que suas tabelas sejam criadas a partir da linguagem SQL, neste caso o banco de dados utilizado foi o PostgreSQL o qual suporta a linguagem supracitada.

Segundo Date (2004), “Um banco de dados é uma coleção de dados persistentes, usada pelos sistemas de aplicação de uma determinada empresa”. Em outras palavras, um banco de dados é um local onde são armazenados dados necessários à manutenção das atividades de determinada organização, sendo este repositório a fonte de dados para as aplicações atuais e as que vierem a existir.

2.3 Ambiente de Desenvolvimento

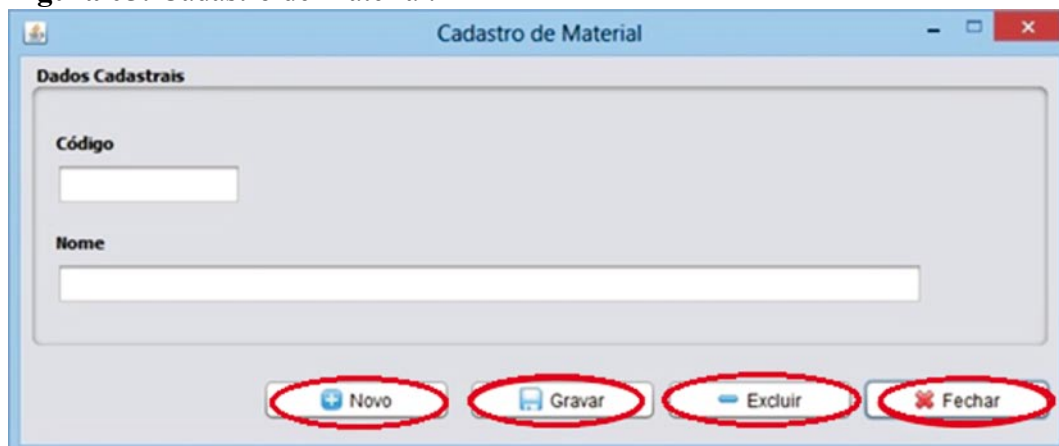
O Eclipse pode ser visto como uma caixa de ferramentas, análoga à caixa de ferramentas de um carpinteiro, que contém vários tipos de ferramentas para fazer seu tipo de trabalho (HEMRAJANI, 2007). É um ambiente de desenvolvimento, que permite o desenvolvimento de aplicações e sistemas corporativos, com codificação em várias linguagens como citado anteriormente. É uma ferramenta de desenvolvimento de aplicação rápida quando aplicado a métodos ágeis.

O Eclipse é uma IDE bastante robusta existem várias funcionalidades que podem ser acoplados a ele como, por exemplo, os plug-ins e frameworks tornando a ferramenta de desenvolvimento muito poderosa. Como o projeto foi codificado em Java e o Eclipse é muito utilizado pela sua robustez e bem aceito pela comunidade esta IDE foi adotada para a implementação dessa aplicação.

2.4 Prototipação

A interface gráfica de um software é algo que deve ser levado em conta por programadores e desenvolvedores na fase de desenvolvimento. É através delas que usuários interagem com as funcionalidades do sistema. As telas devem ser claras e objetivas (SOMMERVILLE, 2003). A prototipação descreve como funciona cada tela do sistema dando melhor visibilidade para desenvolvimento das mesmas com o auxílio dos usuários. A figura 04 apresenta uma das telas desenvolvida utilizando a técnica de prototipação.

Figura 05: Cadastro de Material.



A imagem mostra uma janela de software intitulada "Cadastro de Material". Dentro da janela, há uma seção "Dados Cadastrais" com dois campos de entrada: "Código" e "Nome". Abaixo dos campos, há quatro botões: "Novo" (com um ícone de plus), "Gravar" (com um ícone de disco), "Excluir" (com um ícone de menos) e "Fechar" (com um ícone de X). Os botões "Novo", "Gravar" e "Excluir" estão circunscritos por um círculo vermelho.

Fonte: Autor (2015).

Cada campo é preenchido de acordo com o dado equivalente, o campo código gera código automático do produto ao clicar no botão Novo, para registrar os dados no banco de dados basta pressionar o botão Gravar, para cancelamento basta fechar a tela para excluir registros o botão excluir exerce essa função.

2.5 Planejamento

O planejamento ajuda as corporações a não perderem o controle do cumprimento das suas metas e objetivos traçados, como exemplo, o controle do tempo e a elaboração de estratégias. Pois,

esse planejamento deve ser repassado a todos os membros envolvidos no trabalho, para que entenda toda sua importância e, assim, possam trabalhar na mesma sintonia, com mais qualidade e eficiência para alcançar o seu objetivo.

A tabela 01 apresenta como foi elaborado o planejamento de iterações e testes, as iterações ajudam aos usuários e desenvolvedores a corrigirem eventuais problemas e colaboram também com modificações. O teste realizado foi de aceitação junto com as iterações, os testes foram executados em uma sequência de passos fornecidos pelos programadores para que o usuário as executasse para verificar possíveis erros tanto de usabilidade quanto de codificação. O teste de aceitação é feito antes da implantação do software, o cliente é quem executa este tipo de teste no ambiente de homologação, tendo como objetivo verificar se o software está pronto para ser utilizado pelos usuários finais.

Tabela 01: Planejamento de Iterações.

Versão	Semana	Iteração	Ficha ID	Ficha Título	Lançamento	Tipo de Teste	Defeito Encontrado	Situação
1.0	1	0		Planejamento	03/02/14 à 07/02/14			
1.0	2	0		Planejamento	10/02/14 à 14/02/14			
1.0	3	1	07	Produto	17/02/14 à 28/02/14	Teste de Aceitação	100%	Passou
1.0	4	2	01	Material	03/03/14 à 07/03/14	Teste de Aceitação	100%	Passou
1.0	5	3	02	Fornecedor	10/03/14 à 14/03/14	Teste de Aceitação	100%	Passou
1.0	6	4	05	Estoque	17/03/14 à 21/03/14	Teste de Aceitação	40%	Não Passou
1.0	7	5	05	Estoque	24/03/14 à 28/03/14	Teste de Aceitação	100%	Passou
1.0	8	6	04	Caixa	31/03/14 à 04/04/14	Teste de Aceitação	40%	Não Passou
1.0	9	7	04	Caixa	07/04/14 à 11/04/14	Teste de Aceitação	100%	Passou
1.0	10	8	06	Fechamento Caixa	14/04/14 à 18/04/14	Teste de Aceitação	40%	Não Passou
1.0	11	9	06	Fechamento Caixa	21/04/14 à 02/05/14	Teste de Aceitação	100%	Passou
1.0	12	10	03	Usuário	05/05/14 à 09/05/14	Teste de Aceitação	100%	Passou
1.0	13	11		Implantação	12/05/14 à 16/05/14			

Fonte: Autor (2015).

2.6 Propriedade de Código Coletivo

Em um projeto XP, qualquer membro da equipe tem permissão para modificar qualquer parte do projeto, a qualquer momento. Ou seja, a propriedade do código é coletiva e todos são igualmente responsáveis por todas as partes. Com isso, os desenvolvedores ganham tempo, pois não precisam esperar a autorização de um colega para editar uma área do código e há maior disseminação de conhecimento.

Para que um projeto seja bem-sucedido, é necessário que o ativo envolvimento do interessado, a simplicidade (design, ferramentas, documentação), senso comum e um processo básico/mínimo de desenvolvimento de software é a solução (HEMRAJANI, 2007).

2.7 Técnica de Teste

Segundo Hemrajani (2007), os testes de aceitação podem ser usados como nossos requisitos detalhados da mesma maneira como são utilizados em muitos projetos no estilo Agile. Um exemplo é uma lista de operações válidas que o usuário pode realizar em uma determinada tela.

O teste de aceitação é feito antes da implantação do software, o cliente é quem executa este tipo de teste no ambiente de homologação, tem como objetivo verificar se o software está pronto para ser utilizado pelos usuários finais.

2.8 Implantação

Campos Filho (1994) propõem Sistemas de Informações com conceitos formais tendo-se em vista um cliente que recebe um produto. Essa visão de marketing é o meio de se alcançar os objetivos principais das organizações (KOTLER, 1994). Drucker (1995), também compartilha dessa visão, mostrando que se deve partir das necessidades dos clientes e confirma que os especialistas em informação são os criadores das ferramentas, sendo o gerenciamento da utilização dessas ferramentas os fatores de sucesso. Nota-se, portanto, dois processos distintos: a criação da ferramenta pelos especialistas da área de informática e o gerenciamento da utilização da ferramenta pelos profissionais (clientes) das áreas-fim da empresa. Tanto a criação como a implantação deve facilitar e promover a utilização de todos os recursos do SI.

2.9 Metodologia da Implantação de Albertin

Tendo-se em vista o período pré-implantação, Albertin (1996), apresenta uma metodologia de análise do ambiente e do processo de implantação de SI, propondo três grupos de variáveis conforme a seguir:

Atores:

- a) Capacitação: avaliação da capacitação dos responsáveis pelo desenvolvimento e implantação;
- b) Equipes: avaliação da união entre os envolvidos;
- c) Usuários: como são e qual a experiência com TI;
- d) Apoio da alta gerência: sem o seu apoio não é recomendada a implantação de qualquer projeto;
- e) Patrocinador: executivo que apoie a implantação e garanta a continuidade do projeto.

Treinamento dos Usuários

O treinamento foi dividido em duas etapas, visão geral e específica na qual o sistema foi apresentado para os usuários do aplicativo abrangendo todas as partes envolvidas no software, já na visão específica cada um acompanhou o processo de forma separada e mais intensa. Também foi realizado no treinamento a regra de troca de função entre os usuários assim todos participaram do processo de maneira global no uso do sistema. O nível dos atores envolvidos é básico mais suficiente para manipular o software.

3 RESULTADO

Para desenvolver software é necessário identificar e levantar as necessidades, conhecer o problema e as funcionalidades que o sistema deve ter. Para que se fizesse possível à análise e a especificação de um sistema para gerenciamento e controle de estoque, reuniões com os funcionários foram realizadas para entender o funcionamento, e através disso, levantar todas as funcionalidades que são importantes para a construção do sistema. O objetivo final de um projeto de software é que o sistema construído realize seu papel. Na grande parte dos projetos, todas as pessoas que fazem parte do processo (gerentes, desenvolvedores, clientes e usuários) desejam o total sucesso do projeto.

Se referindo ao final de um sistema de informação é que os usuários tenham melhores condições de trabalho. O software deve permitir que o seu usuário produzisse mais e/ou melhor, para que haja equilíbrio do custo do sistema. Isto significa não somente que o software deve atender às suas especificações, mas também que ele deve ser devidamente estudado minuciosamente para que os requisitos atendidos realmente sejam prudentes com as necessidades dos usuários.

O sistema atende aos requisitos segundo as análises feitas até o momento em que este trabalho foi escrito. Se falando em resolução de problemas, não era possível aos funcionários da X efetuar vendas com rapidez no sistema antigo (manual). No sistema atual, todas as vendas são efetuadas de forma rápida houve melhorias na localização de peças na prateleira. As operações feitas para efetuar devolução ou troca eram bastante demoradas acarretando em erros e divergências, pois as anotações eram feitas em blocos de papel a localização era quase sempre ineficiente. Com a implantação do sistema, a lucratividade da X aumentou de forma considerável. Foi possível alinhar meta a produção obtendo impacto positivo no que diz respeito ao lucro da empresa.

O sistema facilitou o controle do estoque de forma geral, otimizou consulta a notas fiscais de saída, melhorou a contabilização de valores associados a produtos vendidos. Pesquisas tais como: para saber a disponibilidade de uma determinada peça, dentre outras, passaram a ser realizadas de forma segura e eficiente. Existem hoje, no mercado vários pacotes de software prontos para controle de estoque, mas a necessidade de se construir um sistema de informação específico para uma determinada empresa o fato é que, os softwares prontos podem não atender as reais necessidades do cliente ou atender parcialmente. O resultado deste trabalho mostra que um sistema de informação é necessário para o desenvolvimento da empresa, onde o crescimento é notório. O sistema vem sendo acompanhando desde de 2011 e os resultados relatados neste trabalho.

4 CONSIDERAÇÕES FINAIS

Diante dos resultados obtidos com a construção do sistema, pode-se comprovar a viabilidade do problema de pesquisa proposto neste trabalho. Observa-se que a implantação do software na empresa X no principal setor da corporação o estoque, permitiu uma melhora considerável em seu clima organizacional. Percebeu-se, também, que a organização alcançou, com o novo modelo de trabalho, resultados superiores ao esperado. Perante este cenário, o sistema como ferramenta de trabalho apareceu para facilitar e auxiliar a organização no cumprimento dos seus objetivos e metas.

Conclui-se ainda que, além da finalidade proposta, que era oferecer agilidade em seus processos empresariais, flexibilidade e qualidade com baixo investimento, proporcionou, ainda, melhores serviços aos seus clientes agregando valores. O trabalho foi realizado em apenas uma empresa e esta pertencente ao ramo varejista com vendas especializadas em auto perfurantes, por essa razão, pelos resultados obtidos, não é possível fazer maiores generalizações de que o sucesso da implantação desse sistema de informação em qualquer outro ramo empresarial varejista seja positivo.

REFERÊNCIAS

- ALBERTIN, A. L. *Aumentando as chances de sucesso no desenvolvimento e implementação de sistemas de informação*. Revista de Administração de Empresas. São Paulo, v. 36, n.3. Jul/Ago/Set 1996.
- CAMPOS Filho, M. P. **Os sistemas de informação e as modernas tendências da tecnologia e dos negócios**. Revista de administração de empresas. São Paulo, v.34, n.6, p. 33-45. Nov/Dez 1994.
- CODD, E. F. (Ted) Codd. **For his fundamental and continuing contributions to the theory and practice of database management systems**. Disponível em: http://amturing.acm.org/award_winners/codd_1000892.cfm. Acesso em 12 Out. 2014.
- FURLAN D. J. **Modelagem de objetos através da uml**. São Paulo, SP: Ed. Makron Books, 1998. 325 p.
- GUEDES T. A. **Uml 2. Uma abordagem prática**. São Paulo, Ed. Novatec Rubens Prates, 2009. 481 p.
- HEMRAJANI, A. **Desenvolvimento Ágil com Spring, Hibernate e Eclipse**. 1º ed. Editora Pearson, 2007.
- KOSCIANSKI, A.; SOARES, M. dos S. **Qualidade de software: aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. 2. ed. São Paulo: Novatec, 2007.
- KOTLER, P. **Administração de Marketing: análise, planejamento, implementação e controle**. 4ª ed. São Paulo: Atlas, 1994.
- LAUDON K. C.; LAUDON J. P. **Sistemas de Informações Gerenciais**. São Paulo: Prentice Hall, 2004.

PRESSMAN, R. S. **Engenharia de software**. Tradução de José Carlos Barbosa dos Santos. São Paulo: Makro Books, 1995. 1056 p.

SEBRAE-SP, **O que é uma Micro Empresa e Empresa de pequeno porte**. Disponível na internet <<http://www.sebraesp.com.br/principal/abrindo%20seu%20neg%C3%B3cio/orienta%C3%A7%C3%B5es/cria%C3%A7%C3%A3o%20de%20empresas/prin03micro.aspx>>. Acesso em 23 jul 2007.

SETZER, V. W.; SILVA, F. S. C. da. **Bancos de dados: aprenda o que são, melhore seu conhecimento, construa os seus**. São Paulo: Edgard Blücher, 2005.

SOMMERVILLE, L. **Engenharia de software**. Tradução de André Maurício de Andrade. São Paulo: Addison Wesley, 2003. 591 p.

STAIR, R. M. **Princípios de Sistemas de Informação**. Rio de Janeiro: LTC, 1998.