# EXTRAINDO CONHECIMENTO A PARTIR DE BASE DE GERENCIAMENTO DE FÁBRICA DE SOFTWARE

Elias Delgobo Junior (Acadêmico UFPR) E-mail: eliasdelgobojr@gmail.com
Deborah Ribeiro Carvalho (Professora PUC-PR) ribeiro.carvalho@pucpr.br

**Resumo**:Com o aumento de produção de sistemas de informação a demanda por qualidade nos produtos de *software* obriga cada vez mais a adoção de equipes de testes para verificar a conformidade dos sistemas desenvolvidos. A partir da identificação de não conformidades encontradas em testes de software, no período de 2011 e 2012, totalizando 1386 e 2427 ocorrências, respectivamente, verificou-se que o tipo "não conformidade simples" ocorre em 50% dos casos em 2011 e 60% em 2012, fato este que motiva buscar um melhor entendimento sobre os dados. Desta forma foram cruzados os padrões de classificação adotados pela empresa com os resultados de mineração de texto obtidos a partir da ferramenta Sobek. O resultado deste cruzamento evidencia divergências o que sinaliza que as não conformidades são classificadas de forma equivocada pelos testadores, interferindo assim no acompanhamento e controle das não conformidades. Uma das medidas alternativas para minimizar esta situação problema de classificação é rever os critérios adotados para a classificação, evitando assim dubiedade por parte do testador na designação da não conformidade encontrada.

**Palavras-chave**: Testes de *software*, Mineração de texto, Ferramenta de mineração.

# EXTRACTING KNOWLEDGE FROM BASE MANAGEMENT SOFTWARE FACTORY

**Abstract**: With the increased production of information systems, the demand for quality in software products makes it increasingly necessary to adopt test teams to check compliance of the developed systems. From the identification of nonconformities found in software tests in 2011 and 2012, totaling 1,387 and 2,427 occurrences, respectively, it was found that the type "simple nonconformity" occurred in 50% of the cases, in 2011, and 60%, in 2012, which motivates the search for a better understanding of the data. Thus, the classification standards adopted by the company and the text mining results obtained with the Sobek tool were cross-checked. The result of this cross-checking shows divergences, indicating that nonconformities are classified wrongly by testers, thus interfering with their monitoring and control. One alternative to minimize this problem of classification is to revise the criteria of classification to avoid duplicity on the part of the tester in the designation of the nonconformity found.

**Keywords**: Software testing, Text Mining, Mining Tool.

## 1. INTRODUÇÃO

Com o aumento de produção de sistemas de informação a demanda por qualidade nos produtos de *software* obriga cada vez mais a adoção de equipes de testes para verificar a conformidade dos sistemas desenvolvidos. Essas equipes têm por obrigação garantir que o produto entregue ao cliente esteja coerente com os requisitos do projeto e que atenda às expectativas dos usuários.

A partir da análise dos dados sobre os tipos de erros encontrados em teste de *software*, no período de 2011 e 2012, totalizando 1386 e 2427 ocorrências respectivamente, verificou-se que o tipo "não conformidade simples" ocorre em 50% dos casos em 2011 e 60% em 2012. As ocorrências do tipo de não conformidade médio aparecem com 33% do total em 2011 e 14% em 2012 (Gráficos 1 e 2). Chegando a duas vezes a quantidade das outras seis não conformidades somadas em 2011, no caso do tipo simples chega a 3,12 vezes no mesmo ano. Esses percentuais motivam a busca por melhor entendimento dos dados.

O registro das não conformidades encontradas em teste é realizado em dois níveis, primeiro o testador classifica a não conformidade de acordo com uma relação previamente

elaborada, posteriormente o erro é descrito textualmente, bem como indicadas as ações tomadas.

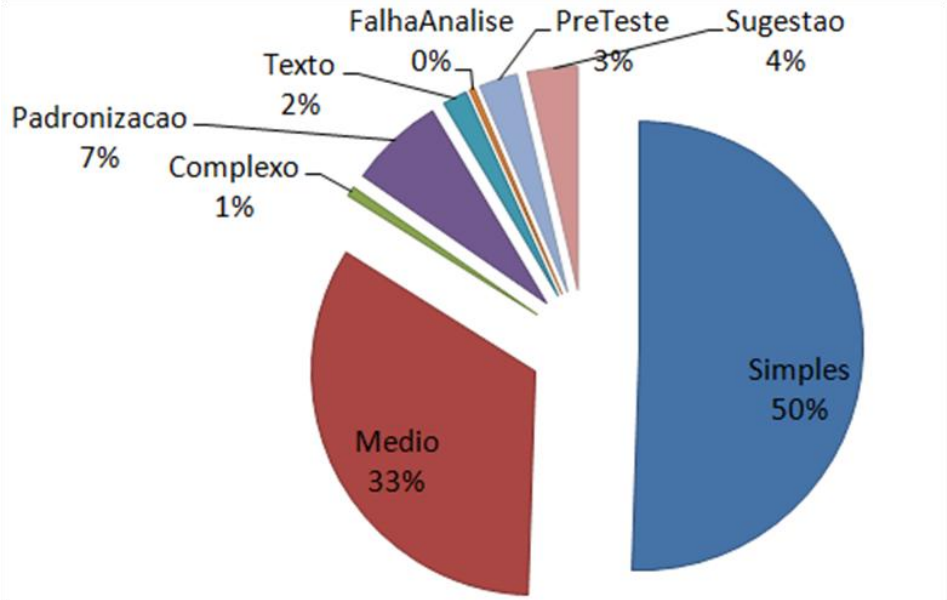


Gráfico 1 – Frenquência relativa de erros encontrados em 2011

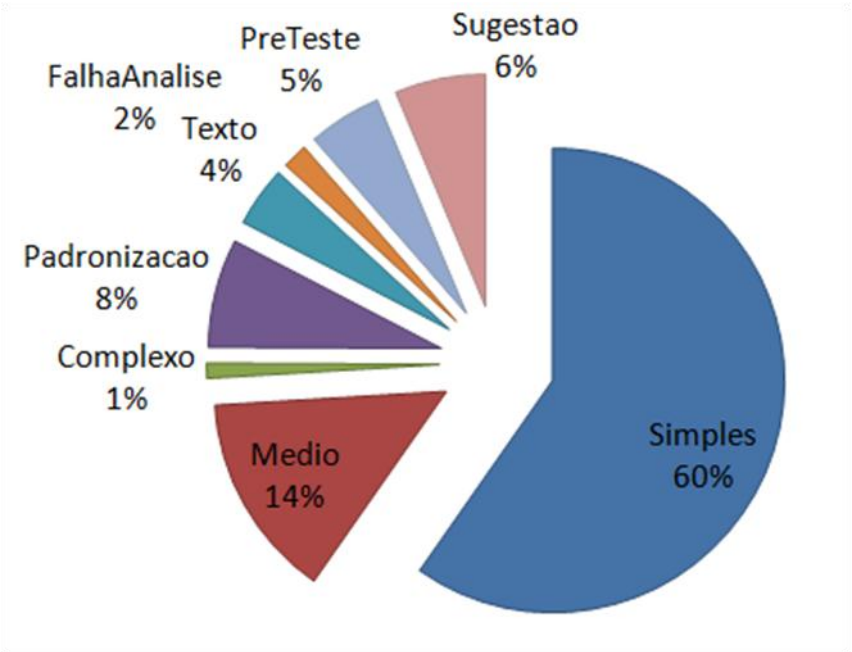


Gráfico 2 - Frenquência relativa de erros encontrados em 2012

Para melhor entender esta relação entre os registros de não conformidade classificação e respectiva descrição surge a pergunta: a mineração de textos pode comprovar a coerência entre estes dois registros?

Este artigo contribui socialmente para que as empresas desenvolvedoras possam adotar mecanismos mais automáticos no acompanhamento do desenvolvimento de software com qualidade, permitindo que em casos de inconsistência, ações tomadas visando melhor entender e prevenir novas situações, inclusive reduzindo custos despendidos de manutenção.

A abordagem recuperação e extração de informação é frequentemente usada quando se dispõe de grande quantidade de texto. Wu He (2012) utiliza extração, agrupamento e indexação em mineração de textos e de dados em um ambiente educacional para revelar padrões de comportamento e aprendizado de alunos, Fouzi Harrag (2013) utiliza a extração entidade nomeada em escritas religiosas para detectar e extrair passagens ou sequências de palavras que contêm informações relevantes a partir dos textos narrações proféticas, Poelmans et al.(2011) utilizam o método de seleção conhecido como mínimo de redundância, máximo de relevância em textos de ocorrências policiais para automatizar a detecção de violência doméstica a partir de textos não estruturados. Yang e Lee(2004) utilizam mineração de texto para descobrir relação entre conteúdos de páginas web e redes neurais para construir um mapa auto-organizado e agrupar as páginas web de acordo com as relações descobertas e criar automaticamente diretórios em web *sites.* Fuller, Biros e Delen(2011) baseados em ADKINS e al. (2004) usam mineração de texto e de dados combinados com técnicas de aprendizado de máquinas para detecção de fraudes em mensagens de depoimentos de envolvidos em crimes.

Com o crescente aumento de dados e necessidade de compreender a massa de informações geradas diariamente, ferramentas que facilitem a visualização das informações extraídas são de grande ajuda. A ferramenta Sobek (LORENZATTI, 2007) é de fácil entendimento e possui boa visualização de resultados da mineração. Em Barbosa, Severo e Reategui (2009), Klemann, Reategui e Rapkiewicz(2011) a ferramenta é comparada a outras ferramentas de mineração de texto, sob os critérios de facilidade de operação, visualização de termos relevantes e disponibilidade na web destacando-se com maior capacidade de extração de informações relevantes . Klemann, Reategui e Lorenzatti (2009) utilizam a ferramenta como apoio a produção textual, Damasceno, Ribeiro e Reategui (2011) utilizam o Sobek em um ambiente educacional de Ciências da Saúde para investigada a possibilidade de associar uma ontologia de domínio na área da saúde à sua Classificação Internacional de Doenças (CID) e Macedo, e Reategui, Lorenzatti e Behar (2009) para apoiar a avaliação de textos produzidos em uma ferramenta de escrita colaborativa.

A ferramenta Sobek (LORENZATTI, 2007) foi desenvolvida na linguagem Java, com possibilidade de execução em qualquer sistema operacional com instalação da máquina virtual Java, segundo Macedo(2009), o programa foi desenvolvido por uma necessidade de professores universitários que trabalham com educação necessitando ler muitos textos dos alunos. À visualização do Sobek (LORENZATTI, 2007) pretende fornecer pistas sobre problemas e qualidade de um texto. O texto pode ser copiado e colado na ferramenta ou convertido em texto a partir de formatos PDF ou DOC.

O processo de mineração de texto nesta ferramenta é baseado na estatística a partir do modelo *n-simpledistance* (Schenker, 2003), no qual inicialmente "expressões são extraídas, procurando por caracteres alfabéticos, separados por espaços ou outros tipos de sinais de pontuação mais comuns". Estes termos são extraídos são pré-processados para a eliminação das *stopwords* e também a identificação das palavras equivalentes retirando seus sufixos (por exemplo: teste, testes = test).

No Sobek (LORENZATTI, 2007) processo de mineração de texto passa pelos estágios: preparação (seleção dos dados), Indexação e Normalização (identificação dos termos, remoção de *stopwords* e eliminação de variáveis morfológicas), cálculo de relevância (frequência absoluta) e análise dos resultados (Morais e Ambrósio, 2007).

Uma vez realizadas as etapas de pré-processamento o método *n-simpledistance*, busca ligar um termo aos próximos *n* termos, e assim sucessivamente considerando uma "borda" parametrizadas que estabelece a distância entre os termos. Dado que as arestas de ligação não

são rotuladas, só se tem a informação de que a distância entre dois termos ligados atende ao *n* parametrizado.

Em outras palavras, tendo nessa ordem "palavra1, palavra2, palavra3 e palavra 4" e *n*=2, para a "palavra1" o algoritmo encontrará ligação com "palavra2(*n*=1) e palavra3 (*n*=2)" mas não encontrará vínculo com a "palavra4 (n=3)" pois a distância é superior a *n*.

Este artigo descreve a adoção da mineração de textos para verificar se a classificação dos erros encontrados nos testes realizados e documentados pelos testadores, está de acordo com a classificação empregada como padrão pela respectiva empresa. Essa classificação organiza as não conformidades em oito categorias (Simples; Médio; Complexo; Padronização; Texto; Falha Analise; Pré-teste; Sugestão).

A oportunidade de avaliar e melhorar a classificação de não conformidades empregadas em desenvolvimento contribui para o aperfeiçoamento do controle, bem como com a redução dos erros encontrados, o que implica diretamente na otimização do processo de desenvolvimento e da qualidade do produto entregue ao cliente.

## 2. MATERIAIS E MÉTODOS

Os dados utilizados neste trabalho foram obtidos a partir de banco de dados de uma empresa de desenvolvimento de software. Essa empresa possui uma equipe especializada em testes funcionais que documento o processo alimentando um banco de dados com os registros de não conformidades encontradas. Os dados apresentam os seguintes atributos: o código da tarefa testada, na qual se descobriu a(s) não conformidade(s); a classificação da não conformidade, a empresa adota oito tipos de não conformidade: Simples, Médio, Complexo, Texto, Padronização, Falha Analise, Pré-teste, Sugestão; a data de início dos testes; e a descrição da não conformidade (Quadro 1).

Quadro 1 - Identificação, tipos base de medição e descrição das variáveis adotadas

| **Variável** | **Tipo de variável** | **Medição da variável** | **Descrição** |
|---|---|---|---|
| Identificação da tarefa | Continuo | 1 → … | Identificação individual da tarefa realizada |
| Identificação do desenvolvedor | Continuo | 1 → … | Identificação individual do desenvolvedor responsável pela tarefa. |
| Tipo de não conformidade | Nominal | Simples<br>Médio<br>Complexo<br>Texto<br>Padronização<br>Falha Analise<br>Pré-teste<br>Sugestão | Classificação de erro encontrado nos testes (com base nos dados da empresa). **Simples**: fácil resolução e/ou demandará menos de uma hora de desenvolvimento; **Médio**: difícil resolução e/ou demandará mais de uma hora de desenvolvimento; **Complexo**: param, de alguma forma, a ação do usuário no sistema; **Padronização**: se refere à padronização visual imposto pela empresa; **Texto**: se refere à gramática, concordância ou sintaxe; **Falha Analise**: se refere à falhas de análise do da tarefa; **Pré-teste**: se refere à falta de arquivos, scripts ou dados para efetuar os testes; **Sugestão**: sugestões de melhorias no sistema. |
| Quantidade de erros | Discreto | 1 → 100 | Quantidade de erros separados por tarefas e por tipo de erro. |
| Data de Início da tarefa | Discreto | 01/01/2011 → 31/12/2012 | Data de início da tarefa. |
| Descrição de não conformidades | Texto | --- | Descrição das não conformidades encontradas nos testes funcionais de sistemas de informação. |

Os atributos que permitiriam identificar a empresa, clientes e funcionários foram omitidos dos experimentos, garantindo assim o sigilo e integridade da empresa em questão e de pessoas envolvidas. Foram selecionados dados para o período compreendido entre 2011 e 2012, totalizando 3813 instâncias, sendo1386 em 2011 e 2427 em 2012. A opção por esse período se deu porque essa classificação foi adotada a partir de meados em 2010, estando em 2011 contando com registros mais estáveis. Vale destacar que classificação segue um padrão empírico baseado na experiência dos testadores durante os testes de *software*.

Os dados da variável "Descrição de não conformidades" foram agrupados em 8 blocos de acordo com a variável "tipo de não conformidade". Cada bloco possui apenas não conformidades do mesmo tipo, por exemplo: no bloco de descrição de não conformidades do tipo simples existem apenas extrações que indicavam não conformidades do tipo simples.

Como mencionado anteriormente uma solução para validação dos dados é o cruzamento dos dados estruturados (classificação utilizada pela empresa) e não estruturados (descrição dos erros) encontrados na base, para isso, deve-se gerar palavras e expressões chave a partir dos dados não estruturados e comparar com os dados estruturados extraídos da base de dados. Neste caso as expressões serão geradas com abordagem de recuperação e extração de informação em base de dados não estruturados.

A partir dos dados dos Gráficos 1 e 2 optou-se por efetuar a validação dos tipos simples, médio e padronização que possuem as três maiores ocorrências entre as não conformidades. A opção pela mineração se deu em função das relações entre as palavras, pois uma simples análise de frequência não seria suficiente para atestar resultados expressivos, a frequência e quantidade das palavras haveria como comparar com a classificação adotada pela empresa por isso optou-se por uma metodologia mais complexa que relacionasse palavras em orações, assim seria possível comparar com a classificação inicial.

Os dados do campo descrição foram separados conforme a classificação utilizando comandos em SQL para extraí-los do banco. Depois de extraídos foi utilizado técnicas de mineração de texto empregadas na ferramenta Sobek. O resultado indicado pela ferramenta foi comparado com a classificação inicial nos resultados e discussões.

Dos textos selecionados foram retirados códigos html e optou-se por não retirar os caracteres especiais (!,@,#,$,%,¨,&,",*,_-,+,=,<,>,:,;,/,?},],[, etc.), numéricos e identificadores de tabelas e células de banco de dados (Ex.: cliente_codigo, cliente_nome, cliente_data_nascimento) pois algumas não conformidades estão associadas à esses caracteres, já que a equipe de testes verifica dados de compra e venda(-, +, >, <, =, *), valores monetários($), cadastros de clientes, funcionários com e-mail (@, :), mensagens padronizadas no sistema(!,?), erros de codificação (#, %, >, <, =, ;, /, &), banco de dados [registros de dados, tipo e tamanho de dados (cliente_codigo, cliente_nome, cliente_data_nascimento)].

Utilizou-se o conjunto padrão de *stopwords* oferecidos pela ferramenta e depois se analisou os resultados com base na estrutura de erros listada, pois, incluir todas as exceções na ferramenta (caracteres especiais, codificação e banco de dados), se tornaria inviável, a frequência mínima para os tipos de não conformidades simples e médio foi de 100 repetições, enquanto que para padronização foi reduzido para 20 repetições, pois acima desse valor a ferramenta não encontra relações entre as palavras, e impossibilita a análise das relações. Como resultado, o Sobek (LORENZATTI, 2007) traz grafos que mostram a relação de proximidade entre os conceitos mais encontrados no texto através das ligações entre eles. O grafo mostra a partir de uma escala de fonte (tamanho da letra e espessura do negrito) a quantidade proporcional da ocorrência do conceito no texto. A ferramenta também mostra os trechos onde ocorreram os conceitos mais encontrados, facilitando a discussão dos resultados. Os grafos e os resultados serão comentados no próximo tópico.

Foram realizados três experimentos, uma para cada tipo de não conformidade (simples, médio e padronização), pois o objetivo da mineração é gerar palavras chaves de cada uma das não conformidades e compará-las a classificação empregada. Para cada mineração repetiu-se o processo de preparação da base e a utilização da ferramenta e a análise do pós-processamento.

## 3. RESULTADOS E DISCUSSÃO

As três minerações efetuadas pela ferramenta mostraram vínculos diferentes entre expressões encontradas nos blocos de texto. As ligações visualizadas na ferramenta tem base na proximidade das palavras, algumas relações não serão aprofundadas por não representarem valor informacional, ou seja, em alguns casos a relação de proximidade entre palavras não expressa uma relação com significado, nesses casos as expressões geradas foram descartados.

Presume-se que a explicação do erro esteja de acordo com a classificação empregada, porém, o resultado das análises pode ser interpretado como um erro de classificação, onde os testadores classificariam erros de tipos diferentes como erro "simples" ou "médio", para verificar se a classificação é empregada corretamente, é necessário comparar os registros dos dois níveis citados.

A Figura 1 mostra o grafo gerado com base no bloco de não conformidades de padronização. A imagem mostra que a maioria das palavras está conectada à "PDV (ponto de venda)" que ocorreu 67 vezes durante todo o texto (a maior ocorrência no texto). Podem-se retirar expressões como "Corrigir + alinhamento", "Corrigir + mensagem", "Padronizar + mensagem", "alterar + mensagem", "Corrigir + coluna", "campo + nome", "campo + Código" essas relações são palavras-chave coerentes com a classificação de não conformidades do tipo padronização. A empresa busca desenvolver programas padronizados, com mensagens, tabelas, campos, formatos, estilos e leiaute iguais em todas as telas, as palavras listadas indicam essa preocupação com padrões, sendo assim, a classificação da equipe de testes está coerente para esse tipo de não conformidade.

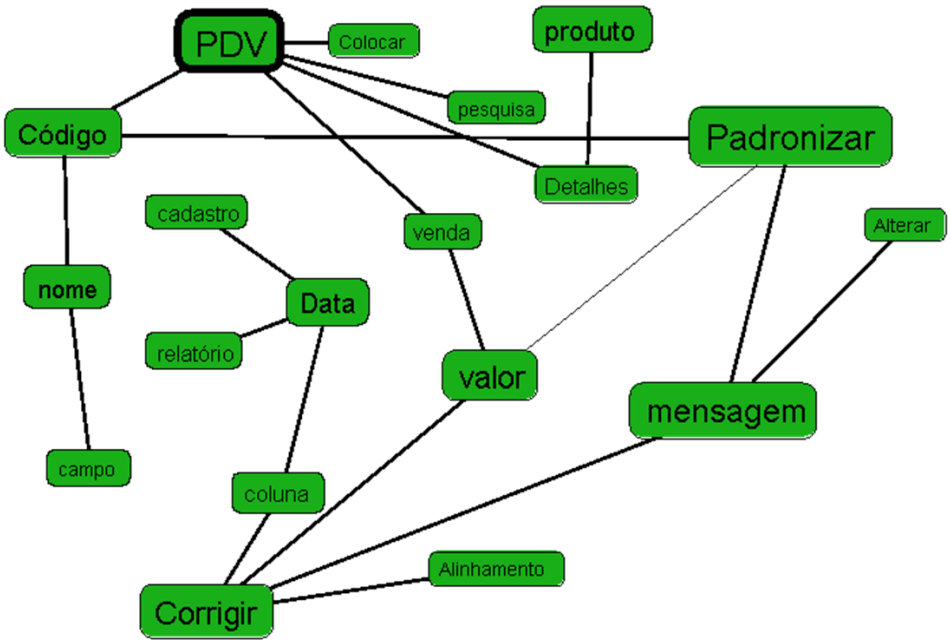


Figura 1 - Grafo tipo de não conformidade padronização

O campo "Colocar" está associado a PDV, porém, verificando o texto através do Sobek e comparando com o original verificou-se que "colocar" estava associado a caracteres especiais e numéricos (dois pontos, ponto final, formado de data[XX/XX/XXXX], etc.) que mesmo mantidos no texto foram ignorados pela ferramenta pois esta não está preparada para estes caracteres, o que causou uma associação sem sentido, mesmo assim as relações são baseadas em problemas com leiaute o que mantém a assertividade da equipe de testes.

A Figura 2 mostra o grafo da mineração de texto do bloco de não conformidades do tipo médio. Neste caso verificam-se relações como: "error + SQL", "SQL + código", "SQL + codigo", "tentar + Gravar + aparece a mensagem", "itens + inserir + produto", "warning + function", "pesquisa + aparece + mensagem + retorno", "pesquisa + aparece + mensagem + sistema", "tentar + Gravar +aparece + mensagem", "Ocorreu um erro + ajax". As ligações mostram que existe uma tendência a indicação de não conformidades de SQL (Structured Query Language), que seriam do tipo pré-teste, como erros do tipo médio, essa diferença de classificação deturpa os indicadores de qualidade e pode gerar decisões erradas para controle e redução de problemas no desenvolvimento de sistemas. Uma solução seria revisar os indicadores de não conformidades e reduzir o escopo desse indicador pra evitar que novos registros sejam efetuados de forma equivocada.

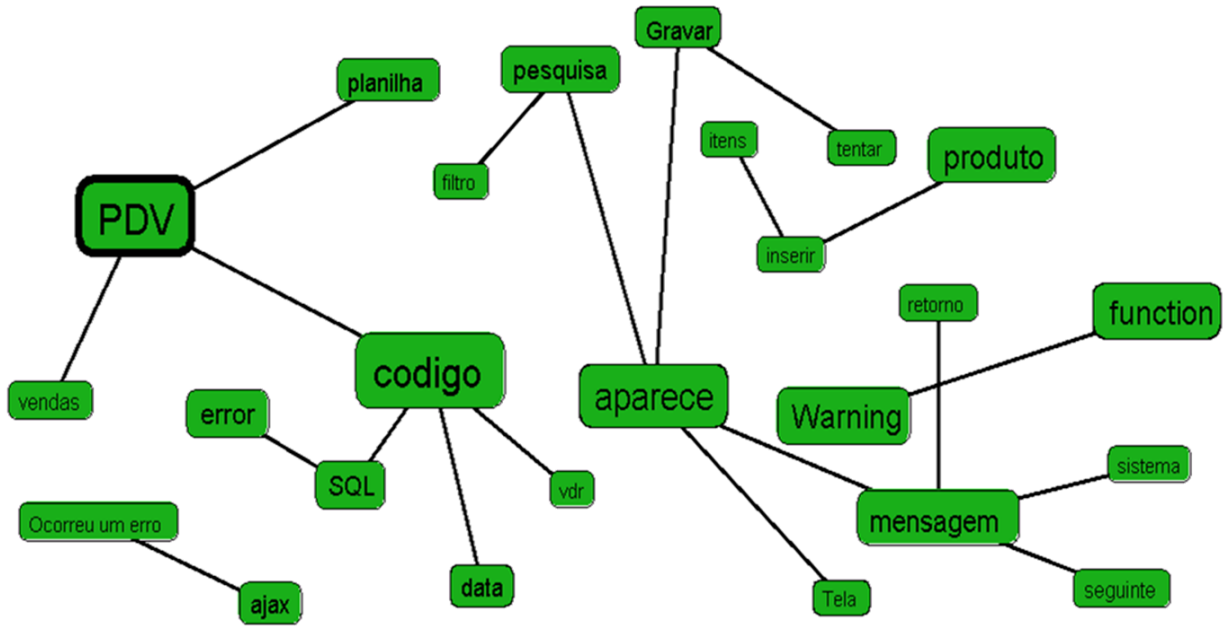


Figura 2 - Grafo tipo de não conformidade médio

A Figura 3 mostra o grafo gerado a partir dos textos classificados como não conformidade simples. Assim como nos outros grafos, a palavra PDV ocorre em maior número e tem vínculo com outras palavras (pesquisa, Sistema, cadastro, vendas, etc.) isso ocorre porque a maioria dos testes estão relacionados a funções ligadas a PDVs.

O Sobek não encontrou ligação para algumas palavras (total, clicar, filtro), porém encontrou vínculo entre "Warning + Error" e "query + function" que indicam erros de funções de armazenamento, edição e consulta de banco de dados, esse erro de classificação e semelhante ao encontrado nos erros do tipo médio da figura 2.

A ligação entre "mensagem" e "aparece" não ressalta significados pela Figura 3, neste caso decidiu-se verificar no texto em quais momentos as duas palavras apareciam. A palavra mensagem ocorre em frases como "Padronização de mensagem", que mostra outra classificação equivocada por parte da equipe de testes. Foram encontrados ocorrências dessa

ligação com requisições de banco de dados ou seja erros de SQL, que são classificadas como pré-teste.

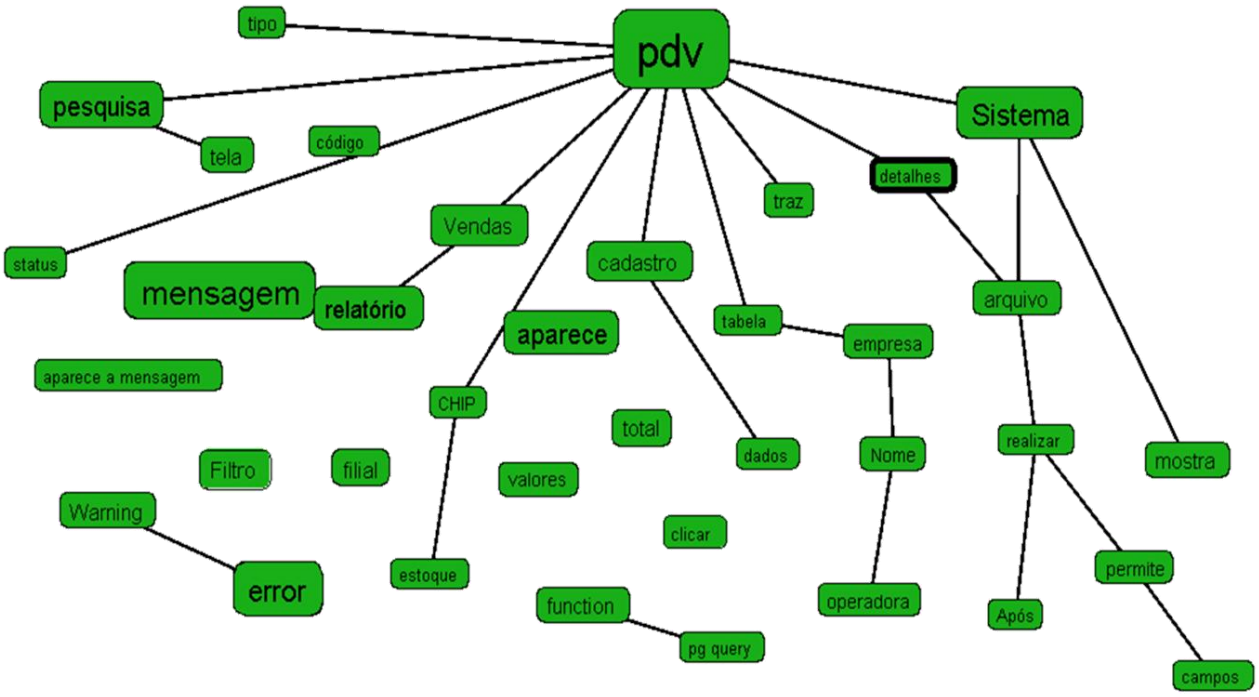


Figura 3 - Grafo tipo de não conformidade simples

O uso da palavra Sistema relacionada ao PDV não se considera válida, pois o uso dessa palavra pelo testador simboliza o sistema a ser testado como um todo. Outros vínculos cabem nesta análise, pois não trazem informação sobre as classificações das não conformidades.

Verificou-se que os tipos simples e médios absorvem tipos de não conformidades diferentes e que deveriam estar alocadas em outra classificação, isso ocorre, pois estes tipos são abrangentes, não tendo foco em situações pontuais como erros de texto (concordância, gramática), padronização (leiaute, formatação, numeração de páginas, etc.), o que provoca escolhas erradas dos testadores. Por esse motivo necessitasse rever os tipos de não conformidades e padronizá-las para que não possam ser classificado em mais de um classificador.

Pode-se optar por alterar as classificações simples e médio para que todos os erros classificados dessa forma passem a ser adicionados em outros tipos, não necessitando desses itens que serão excluídos. Dessa forma simplificaria a classificação e evitaria erros no registro das não conformidades.

A mineração de texto contribuiu para o trabalho facilitando a visualização dos dados e indicando expressões e relações entre palavras que uma simples análise de frequência não relacionaria, permitindo confrontar as informações da mineração com a classificação inicial (quadro1). Dessa forma a mineração auxiliou no trabalho de validação dos dados mostrando que estes não devem ser usados como indicadores de não conformidade.

## 4. CONCLUSÕES

Este trabalho utilizou uma ferramenta de mineração de texto, que gera resultados gráficos, para análise de coerência entre indicadores de não conformidade em sistemas de informação, verificou-se que para não conformidades do tipo médio, simples e padronização,

a quantidade de ocorrências é muito superior aos outros tipos(Complexo, Texto, Falha Análise, Pré-teste, Sugestão). Os dados de descrição das não conformidades foram extraídos e comparados com a descrição original. Verificou-se que os tipos simples e médio absorvem não conformidades de outras classificações, isso acontece porque os tipos simples, médio são genéricos, não tem foco em problemas pontuais como erros de texto (concordância, gramática), padronização (leiaute, formatação, numeração de páginas, etc.). Se observou a classificação das não conformidades e melhor ajuste para manter o escopo delimitado e sem redundância.

Os resultados mostraram que a classificação de não conformidades está equivocada, sendo assim, os métodos estatísticos empregados pela empresa obtém resultados incorretos. Esses resultados, se utilizados para tomada de decisões relacionadas a qualidade de desenvolvimento de sistemas, não terão efeito corretivo para as não conformidades.

O trabalho mostrou que a mineração de texto pode ser usada para descobrir padrões em dados não estruturados em dados de desenvolvimento de software e assim aperfeiçoar os índices de não conformidades no processo de testes. A classificação correta dos índices auxilia no controle dos erros encontrados em testes e facilita a tomada de decisão para reduzir os problemas de não conformidade do *software*.

Com o levantamento bibliográfico não foram localizados documentos cientificos que demonstrem a utilização da ferramenta Sobek e tecnicas de mineração de texto relacionados à levantamentos em testes de software, o que demonstra o ineditismo desse trabalho, por utilizar a aplicação de abordagens de recuperação e extração de informação em ambiente de testes de *softwares e* oportunizar a utilização tanto da ferramenta quanto das técnicas de mineração em desenvolvimento de programas de computador.

Para a empresa, a oportunidade de avaliar e melhorar a classificação de não conformidades empregadas em desenvolvimento melhora o controle de erros encontrados pela equipe de testes. A solução acrescenta um passo na qualidade dos produtos de *software* e melhora o controle do processo de desenvolvimento de sistemas, reduzir a quantidade de erros encontrados em testes, reduzindo o tempo de desenvolvimento e melhorando a qualidade do sistema apresentado ao cliente, pois com dados corretos, as decisões tomadas a partir destes dados poderão ser mais produtivas.

Para empresas do setor de desenvolvimento de sistemas, do ponto de vista de testes de *software*, a solução acrescenta um passo na qualidade dos produtos de *software* e melhora o controle do processo de desenvolvimento de sistemas, pois com dados corretos, as decisões tomadas a partir destes dados poderão ser mais produtivas.

O trabalho contribuiu para reforçar o uso de mineração para encontrar expressões significativas em grande quantidade de texto de forma eficiente e com perdas mínimas de informação. A utilização de uma ferramenta de mineração de texto para validação de índices de classificação de tipos de não conformidades em desenvolvimento de sistemas de informação se mostrou efetiva nas validações e sua forma intuitiva de utilização demandou pouco tempo de aprendizado o que reduziu o tempo de trabalho para validação dos textos extraídos do banco de dados.

## AGRADECIMENTOS

Os autores agradecem a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pela concessão da bolsa de estudo.

## REFERÊNCIAS

**ADKINS, Mark. et al.** *Advances in Automated Deception Detection in Text-Based Computer-Mediated Communication*. Proceedings of the SPIES Defense and Security Symposium, Orlando, Florida, 2004.*

**BARBOSA, Maria Lúcia. SEVERO, Carlos Emilio Padilla. REATEGUI, Eliseo**. *Mineração de padrões no gênero textual blog .CINTED, V. 7 Nº 3, 2009.*

**DAMASCENO, Fábio Rafael. RIBEIRO, Alexandre. REATEGUI, Eliseo**. *Aplicação Educacional de uma Ferramenta de Mineração de Textos Integrada a uma Ontologia de Domínio na Área da Saúde. CINTED, V. 9 Nº 1, 2011.*

**FULLER, Christie M. BIROS, David P. DELEN, Dursun**. *An investigation of data and text mining methods for real world deception detection. Expert Systems with Applications, v 38 p. 8392-8398, 2011.*

**HARRAG, Fouzi**. *Text mining approach for knowledge extraction in Sahîh Al-Bukhari. Computers in Human Behavior, 2013.*

**HE,W**. *Examining students' online interaction in a live video streaming environment using data mining and text mining. Computers in Human Behavior, v. 29, n. 1, p. 90-102, 2013.*

**KLEMANN, Mirian. REATEGUI, Eliseo**. *O Emprego da Ferramenta de Mineração de Textos SOBEK como Apoio à Produção Textual. Anais do XX Simpósio Brasileiro de Informática na Educação, 2009.*

**KLEMANN, Mirian. REATEGUI, Eliseo. RAPKIEWICZ, Clevi**. *Análise de Ferramentas de Mineração de Textos para Apoio à Produção Textual. Anais do XXII SBIE - XVII WIE, 2011.*

**LORENZATTI, A.** *SOBEK: uma Ferramenta de Mineração de Textos*. Caxias do Sul/RS: Universidade de Caxias do Sul, 2007. Trabalho de Conclusão de Curso (Graduação).

**MACEDO, A., REATEGUI, E., LORENZATTI, A., BEHAR, P**. *Using Text-Mining to Support the Evaluation of Texts Produced Collaboratively. Education and Technology for a Better World: Selected papers of the 9th World Conference on Computers in Education, Bento Gonçalves, Brazil. Springer, 2009.*

**MORAIS, E. A. M. AMBRÓSIO, A. P. L**. *Mineração de Texto. Instituto de Informática. Universidade Federal do Goiás. 2007. 30 p.*

**POELMANS, Jonas. et al.** *Text mining with emergent self-organizing maps and multi-dimensional scaling: A comparative study on domestic violence. Applied Soft Computing, v. 11 p. 3870-3876, 2011.*

**SCHENKER, Adam**. *Graph-theoretic techniques for web content mining. 2003. 131 f. Tese (Doutorado em Philosophy – Department of Computer Science and Engineering College of Engineering University of South Florida, Florida 2003.*

**YANG, Hsin-Chang. LEE, Chung-Hong**. *A text mining approach on automatic generation of web directories and hierarchies. Expert Systems with Applications, v.27 p645-663, 2004.*