Open Coding Tool: Uma Ferramenta de Codificação Colaborativa para Análise de Dados Qualitativos

Maurício dos Santos Escobar, Alex Severo Chervenski e Andréa Sabedra Bordin

[1]Curso de Engenharia de Software, Universidade Federal do Pampa - UNIPAMPA
Av. Tiaraju, 810 - Ibirapuitã, Alegrete - RS, 97546-550

mauricioescobar.aluno@unipampa.edu.br, alex.chervenski@gmail.com

andreabordin@unipampa.edu.br

Abstract. *In software engineering research problems have been studied from a qualitative perspective. Several studies adopt qualitative text-based methods, which have a procedure called coding, through which data are broken down to produce new knowledge. As it is an analytical task, it requires people and execution time. There are some software to perform this procedure. However, most are not free and do not allow coding to be performed by a group of people. This article presents a collaborative encoding tool for textual data. In a case study, the use of the tool significantly reduced the coding time, without prejudice to the quality of the generated codes.*

Resumo. *Na engenharia de software problemas de pesquisa vêm sendo estudados sob a ótica qualitativa. Vários estudos que adotam métodos qualitativos baseados em texto, possuem um procedimento denominado codificação, através do qual os dados são desmembrados para a produção de um novo conhecimento. Por ser uma tarefa analítica, a codificação demanda pessoas e tempo de execução. Existem alguns softwares para realização desse procedimento, no entanto, a maioria não é gratuita e não permite que a codificação seja executada por um conjunto de pessoas. Este artigo apresenta uma ferramenta de codificação colaborativa de dados textuais. Em um estudo de caso, o uso da ferramenta diminuiu significativamente o tempo de codificação, sem prejuízo para a qualidade dos códigos gerados.*

1. Introdução

Métodos de pesquisa qualitativa foram desenvolvidos nas Ciências Sociais para habilitar pesquisadores a entender pessoas e os contextos culturais e sociais no qual eles vivem [Denzin and Lincoln 2005]. De acordo com [Gibbs 2009], uma característica essencial dos dados qualitativos é que eles se originam de praticamente qualquer forma de comunicação humana – escrita, auditiva, ou visual. No entanto, o tipo mais comum de dado qualitativo usado em análises é o texto, que pode ser transcrições de entrevistas, notas de campo de trabalho etnográfico ou outros tipos de documentos.

Na Engenharia de Software questões complexas podem, quando estudadas sob a ótica qualitativa, gerar hipóteses bem fundamentadas e resultados que incorporam a complexidade do fenômeno estudado. Além disso, estudos qualitativos oferecem explicações mais ricas e novas oportunidades para estudos futuros. Eles são apropriados quando as variáveis não estão definidas ou quantificadas e existe pouco estudo

teórico ou empírico [Dybå et al. 2011]. Algumas pesquisas de Engenharia de Software que empregaram métodos qualitativos são as de [Salinger et al. 2013], [Hoda et al. 2012] e [McLeod et al. 2011].

Alguns métodos de análise de dados qualitativos baseados em texto como a Análise de Conteúdo, a Análise Temática e a Teoria Fundamentada em Dados, utilizam um procedimento denominado de codificação. A codificação de dados qualitativos, segundo [Gibbs 2009], é a forma como o pesquisador define de que tratam os dados em análise, através da aplicação de nomes a passagens de texto, de sua categorização, de forma a estabelecer uma estrutura temática, que possibilite possíveis interpretações do seu conteúdo. De acordo com [Elliott 2018], a codificação é quase um processo universal em pesquisa qualitativa, sendo um aspecto fundamental do processo analítico e a maneira pela qual pesquisadores desmembram seus dados para produzir algo novo.

A codificação é uma tarefa analítica e, por isso, muito dependente da intervenção humana. Assim, demanda tempo para ser realizada. É comum que esse procedimento seja realizado através de Softwares para Análise de Dados Qualitativos (SADQs). No entanto, poucos softwares permitem a colaboração de vários pesquisadores em um mesmo projeto e, mais importante, essa forma de colaboração pode não ser tão eficiente, uma vez que permite que todos os pesquisadores tenham acesso ao mesmo conjunto de dados, sem atribuição de responsabilidade de codificação.

Uma proposta de solução para esse problema partiria do uso do conceito de *crowdsourcing*, onde membros de uma comunidade podem realizar de forma colaborativa uma determinada tarefa, neste caso, o procedimento de codificação. Em um processo de codificação colaborativa, gasta-se menos tempo, pois várias pessoas (codificadores) podem codificar dados textuais ao mesmo tempo. Além disso, se implementar mecanismos de verificação dos códigos pelos pesquisadores, o resultado pode ser mais confiável.

O objetivo deste artigo é apresentar uma proposta de ferramenta para criação colaborativa de códigos a partir de dados textuais. A ferramenta permite criar grupos de dados textuais que podem ser atribuídos a diferentes grupos de usuários. Dessa forma, um mesmo conjunto de dados pode ser codificado por várias pessoas. Além disso, a ferramenta permite que especialistas verifiquem a qualidade da codificação. A avaliação da ferramenta foi realizada através de um estudo de caso real, onde a codificação colaborativa de dados textuais foi executada com celeridade e qualidade.

O artigo está organizado da seguinte forma: na Seção 2 é apresentada a fundamentação teórica com a descrição de alguns métodos qualitativos e funcionalidades encontradas em softwares de análise de dados qualitativos; na Seção 3 as relacionadas a esta proposta; na Seção 4 é apresentada a ferramenta; na Seção 5 o estudo de caso envolvendo o uso da ferramenta e, por fim, as considerações finais.

2. Fundamentação Teórica

2.1. Métodos de Pesquisa Qualitativa

Dentre os métodos qualitativos, destacam-se alguns onde o procedimento de codificação é mais utilizado, são eles: Análise de Conteúdo, Análise Temática e Teoria Fundamentada em Dados (*Grounded Theory*).

A Análise de Conteúdo consiste em uma abordagem sistemática de codificação e categorização utilizada para explorar grandes quantidades de informações textuais, para determinar as tendências e padrões de palavras utilizadas, a sua frequência, as suas relações e as estruturas e discursos de comunicação [Mayring 2004]. Aceita-se que o seu foco seja qualificar as vivências do sujeito, bem como suas percepções sobre determinado objeto e seus fenômenos [Bardin 1977].

A Análise Temática (AT) é um método para identificar, analisar e relatar padrões (temas) dentro dos dados. Ela organiza e descreve o conjunto de dados de forma detalhada [Boyatzis 1998]. Segundo [Braun and Clarke 2006], a AT é muito semelhante à Análise de Conteúdo. No entanto, difere no fato de que os temas geralmente não são quantificados. Além disso, a AT também se concentra em encontrar temas em vez de criar categorias em que a unidade de dados é mais do que uma palavra ou frase, como na Análise de Conteúdo.

Na Teoria Fundamentada em Dados os dados são revisados usando uma análise comparativa constante para marcar códigos e agrupar em conceitos para determinar temas. A análise comparativa constante (categorizar, comparar e conceituar) reduz os dados para identificar conceitos e desenvolver uma teoria fundamentada [Strauss and Corbin 2008]. Esta metodologia serve para que o pesquisador consiga gerar explicações (e.g teorias) de um processo, ação ou uma interação moldada pelas visões de um grande número de participantes [Creswell 2014].

2.2. Software de Análise de Dados Qualitativos

Softwares de Análise de Dados Qualitativos (SADQs) possuem diferentes recursos, cuja aplicabilidade pode variar de acordo com os objetivos e a natureza da pesquisa. Assim, cabe ao pesquisador avaliar a utilidade desse tipo de software em sua pesquisa e quais ferramentas utilizar. As principais funcionalidades encontradas em um SADQ são:

- Pesquisa de conteúdo: permite coletar dados qualitativos, extraindo conteúdo de arquivos de vídeo, áudio, documentos de texto, gráficos e outros.
- Visualização e relatórios de dados: permite visualizar todas as formas de dados eletrônicos, incluindo entrevistas, pesquisas, vídeos com imagens e dados bibliográficos;
- Armazenamento e codificação: permite aos analistas de dados e outros usuários de software executar diferentes formas de codificação, como codificação de palavras-chave e texto. Permite codificar sistematicamente dados em diferentes formatos e categorias;
- Ligação de dados: permite que os usuários formem clusters, redes ou categorias de dados;

De acordo com [Moreira 2007], não existe consenso entre os pesquisadores quanto às vantagens do uso desses softwares, sobretudo na pesquisa qualitativa. Assim, o autor aponta uma série de argumentos contra e a favor ao uso deles. Grande parte dos argumentos a favor dizem respeito à facilidade e rapidez proporcionada pelo uso do computador e seus recursos. Já os argumentos contra estão relacionados aos riscos da mecanização de um processo interpretativo.

Os SADQ fazem, a partir de um computador, o que os pesquisadores vêm fazendo manualmente há décadas: a armazenagem, o gerenciamento e a recuperação de dados. Como fio condutor dessas funções, está o processo de codificação que consiste na

designação de códigos para pequenos trechos do texto. Esses códigos, por sua vez, podem ser sobrepostos, permitindo que vários trechos de texto sejam recuperados a partir de um mesmo código.

Contudo, essa codificação não é executada de forma autônoma pelo software, mas depende da indicação do pesquisador. Assim, embora o processo de análise seja mecanicamente facilitado e acelerado pelo software, a codificação é resultado do raciocínio e da versatilidade do pesquisador. E, considerando que durante o processo de análise o pesquisador passa a ter uma visão mais geral sobre os dados, esses softwares permitem a revisão dos códigos, combinando-os ou dividindo-os [Moreira 2007].

3. Trabalhos Relacionados

Segundo [Gibbs 2009], três SADQs parecem ser os mais utilizados pelos pesquisadores, são eles: Atlas.ti, MAXqda e NVivo. Contudo, essas ferramentas não são gratuitas e não permitem que a codificação seja realizada de forma colaborativa. Nesta pesquisa, entende-se o requisito de colaboração de uma forma mais específica, como a capacidade de permitir que o conteúdo textual seja dividido em vários conjuntos de dados, que sejam criados vários grupos de codificadores e que cada grupo possa acessar um ou mais conjuntos de dados.

Uma pesquisa por SADQs gratuitos, identificou a existência de 14 softwares para uso em diversas metodologias de pesquisa qualitativa. Desses a maioria são aplicações desktop, que não permitem qualquer tipo de atividade colaborativa. Na Tabela 1, são exibidos os únicos SADQs gratuitos e disponíveis para uso na plataforma web, sendo este último critério um indicador da possibilidade de atividades colaborativas.

Dos três softwares encontrados, QCAmap e Computer Assisted Text Markup and Analysis (CATMA) não permitem qualquer colaboração. Já o software Coding Analysis Toolkit (CAT) permite criar subcontas e, a partir de um titular de conta principal, é possível convidar colaboradores para o processo de codificação. São 2 tipos de conta: o usuário com tipo de conta especializada têm permissão para acessar, fazer upload e bloquear conjuntos de dados; o usuário com tipo de conta regular só pode acessar conjuntos de dados quando recebe permissão de alguém com conta especializada, sendo essa conta a indicada para codificadores.

No teste de uso do CAT percebeu-se que as importações de trechos textuais não são fáceis de serem realizadas, mesmo seguindo-se o passo a passo do tutorial. Além disso, foi percebível um visual antiquado, pouco amigável, algo que pode desestimular os usuários. Por fim, recentemente foi relevado aos usuários a descontinuidade deste software.

A ferramenta proposta neste trabalho, soluciona as limitações destacadas no CAT no que tange, por exemplo, à facilidade de importação de trechos textuais e tem potencial para se tornar uma alternativa viável de uso em pesquisas que demandem um processo de codificação colaborativa.

Softwares gratuitos na web	Colaborativa	Site
QCAmap	Não	www.qcamap.org
CATMA	Não	catma.de
CAT	Sim	cat.texifter.com

Tabela 1. Ferramentas Gratuitas de Análise Qualitativa de Dados

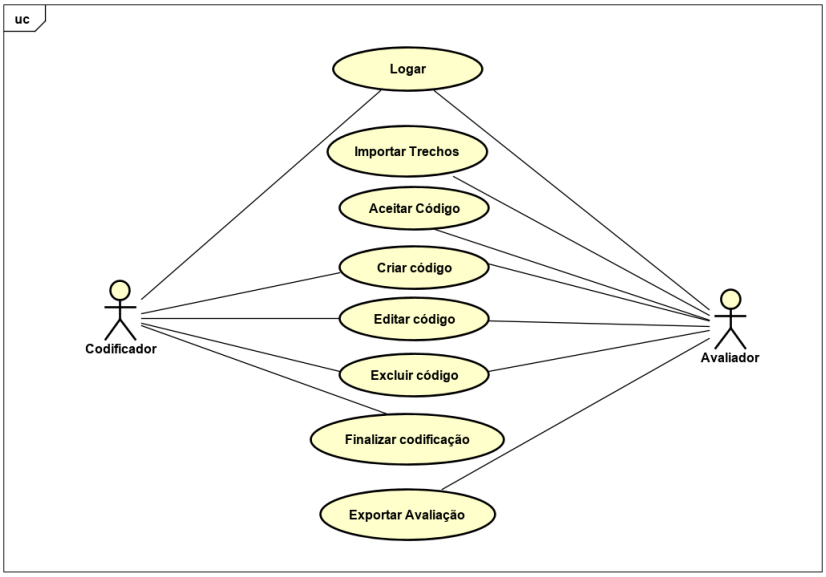
4. Metodologia de Desenvolvimento da Ferramenta

O objetivo da ferramenta ***Open Coding Tool*** é apoiar o processo de codificação de dados textuais de forma colaborativa, permitindo que códigos sejam criados por uma comunidade de codificadores e posteriormente sejam avaliados e chancelados por pesquisadores, de forma online e gratuita.

Para o desenvolvimento da ferramenta, optou-se por um processo ágil de software, entregando em intervalos de tempo curto as funcionalidades solicitadas. A coleta de requisitos foi realizada com uma pesquisadora da área de Engenharia de Software que utiliza métodos qualitativos e seus orientandos; os requisitos foram analisados, modelos foram criados e um documento de especificação de requisitos foi desenvolvido.

A Figura 1 exibe o diagrama de caso de uso com as principais funcionalidades da ferramenta. Existem dois tipos de usuários: o codificador e o avaliador/pesquisador. O codificador tem permissão para analisar os trechos textuais, gerenciar a criação de códigos e finalizar o processo. O avaliador pode avaliar os códigos criados pelos codificadores, podendo aceitar, modificar ou rejeitar os códigos criados anteriormente, além de criar novos códigos, caso os disponíveis não sejam adequados.

Figura 1. Diagrama de Caso de Uso



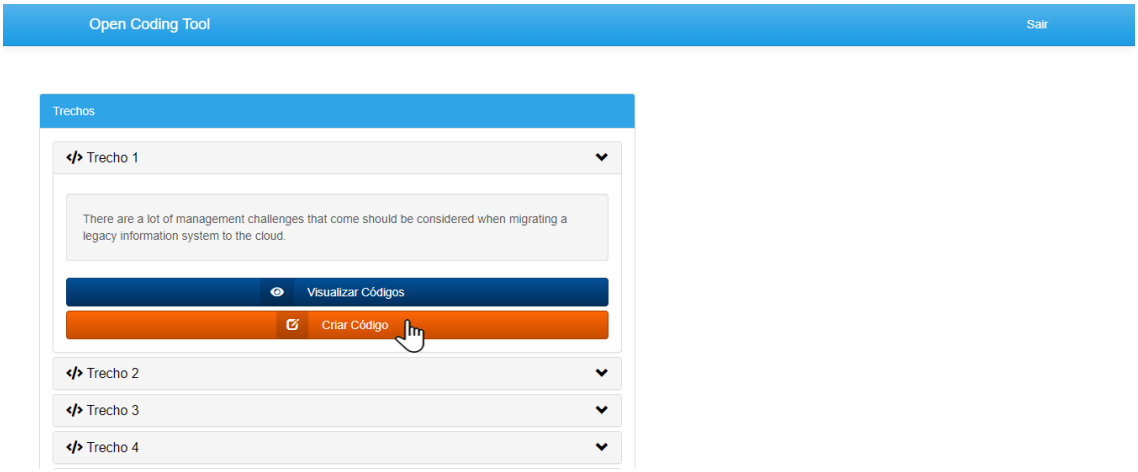
A implementação do software foi feita com a linguagem de programação open source PHP com apoio do framework ZendFramework 1.12.3 para criar o backend. Para o frontend foram utilizados HTML5, CSS3, Javascript, com o apoio do framework Bootstrap3. O gerenciador de banco de dados foi o MySQL.

Antes de ser colocada em produção a ferramenta foi testada junto a grupo de codificadores voluntários. As percepções foram coletadas e serviram para o seu aprimoramento. O deployment da versão final foi feito em um servidor institucional [1].

4.1. Funcionalidades da Ferramenta

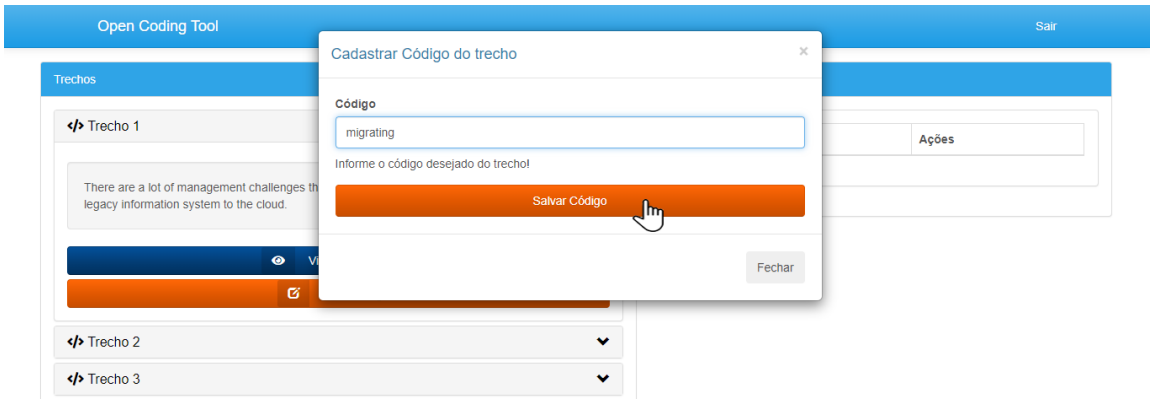
Cada codificador deve possuir um *login* e senha para acessar a ferramenta e visualizar os trechos textuais alocados para a sua análise, como exposto na Figura 2. Após análise do trecho, o codificador pode criar um ou mais códigos referentes à sua visão analítica, de forma individual.

Figura 2. Perfil de Codificador: Análise dos trechos textuais



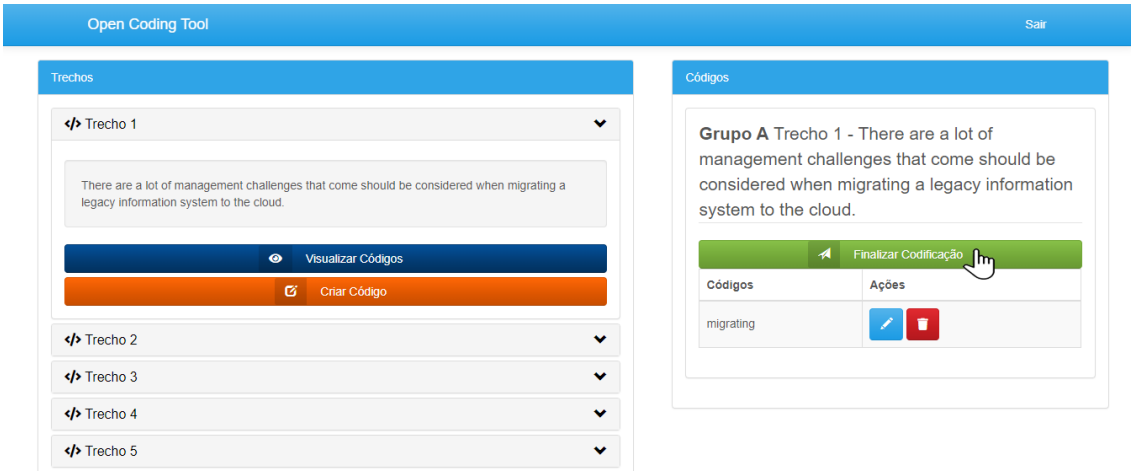
Como exibido na Figura 3 podem ser criados vários códigos relacionados ao trecho textual. Cada código criado poderá ser editado e até mesmo excluído antes de ser salvo conforme explicitado na Figura 4.

Figura 3. Perfil de Codificador: Criação de códigos



[1] Ferramenta Open Coding Tool: `https://opencodingtool.com.br`

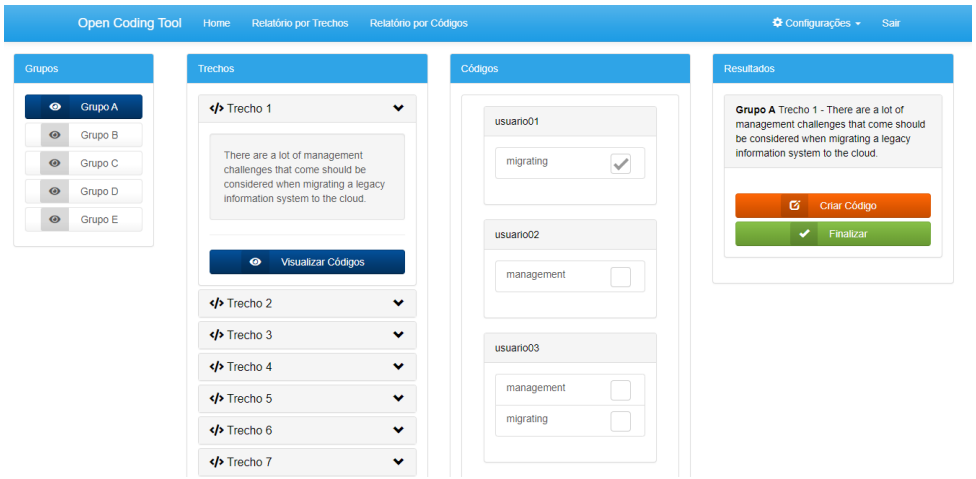
Figura 4. Perfil de Codificador: Salvar código



Após a finalização da codificação, não é mais permitido ao codificador adicionar ou modificar códigos referentes ao trecho textual que estava sendo analisado.

Os avaliadores também devem possuir *login* e senha para acessar a ferramenta. Com esse perfil é possível visualizar todos códigos criados pelos codificadores em todos os trechos, como mostra a Figura 5. O avaliador possui a opção de aceitar (selecionando o *check box*) o(s) código(s) mais apropriados. Também é possível criar códigos de acordo com suas análises, bem como editar códigos já criados pelos codificadores.

Figura 5. Perfil de avaliador: Aceitar ou criar códigos



5. Estudo de caso

A ferramenta *Open Coding Tool* foi desenvolvida para dar apoio ao procedimento de codificação aberta da Teoria Fundamentada em Dados, método utilizado na pesquisa de [Chervenski and Bordin 2020] sobre entendimento de sistemas legados a partir da literatura científica. A referida pesquisa analisou trechos textuais oriundos da literatura com o objetivo que elucidar os elementos que causam ou contribuem para um sistema se tornar legado, as consequências e as possíveis estratégias de evolução de tais sistemas, bem como suas relações, contribuindo para a criação de uma teoria.

No trabalho de [Chervenski and Bordin 2020], a coleta de trechos textuais foi realizada através de um mapeamento sistemático na literatura, seguindo os passos bem definidos por [Petersen et al. 2008]. Foram utilizadas 3 bases digitais (ACM Digital library, IEEE Xplore e Science Direct), após a criação e utilização de *strings* de busca foi possível obter um número de 87 estudos. Após a leitura dos estudos, foi possível identificar 111 trechos textuais que foram armazenados em planilhas no *Google Drive* e importadas para a ferramenta *Open Code Tool* através de arquivos CSV.

O processo colaborativo de codificação foi pensado como alternativa, em razão da necessidade de se obter um consenso confiável de grupos de especialistas em um espaço de tempo reduzido, pois o conjunto de dados textuais analisados era volumoso.

A codificação colaborativa na ferramenta *Open Coding Tool* foi realizada por 14 participantes, todos alunos dos cursos de Engenharia de Software e Ciência da Computação, e por 2 avaliadores que possuiam o papel de validar os códigos criados pelos participantes. A escolha dos participantes foi feita levando em consideração a experiência acadêmica de cada aluno. Os alunos do curso de Engenharia de Software já deveriam ter cursado a disciplina de Evolução de Software e os alunos do curso de Ciência da Computação a disciplina de Engenharia de Software 2, onde tópico Sistemas Legados é abordado.

Os participantes foram separados em 5 grupos, sendo esse número definido em função da quantidade e do tamanho dos trechos textuais. Preferencialmente cada grupo deveria ser composto por 3 participantes, de forma que as decisões de codificação não ficassem polarizadas e que cada grupo tivesse condições de codificar entre 25 a 30 trechos. Para cada grupo foi alocado uma quantidade de trechos textuais diferentes, sem sobreposições. Como alguns trechos eram bem maiores, optou-se por alocar uma quantidade menor aos grupos D e E. Alguns exemplos de trechos textuais extraídos da literatura estão destacados na Tabela 2. Na Tabela 3 é possível visualizar os grupos, números de participantes e quantidade de trechos alocados.

Tabela 2. Exemplos de trechos textuais extraídos da literatura

Exemplo de trecho textual
Older legacy software systems are frequently replaced as they become obsolete.
Recent estimates confirm at least 180 billion lines of legacy, smelly code are target of software refactoring.
legacy systems usually lack complete and updated engineering documents.
In a legacy system, component-aging is unavoidable.
legacy systems are operated across decades.

Tabela 3. Relação de grupos para codificação colaborativa.

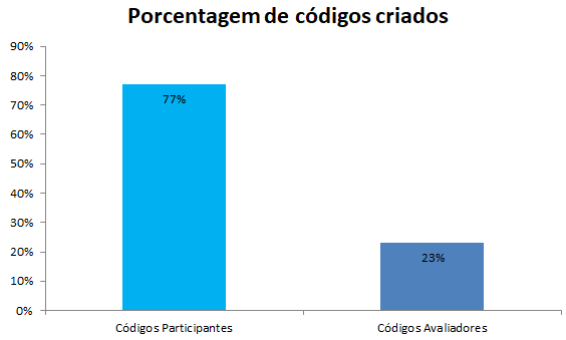
Grupo	Número de participantes	Quantidade de Trechos
A	3	31
B	3	30
C	2	30
D	3	15
E	3	14

Cabe destacar que a construção dessa aplicação *web* possibilitou a todos os participantes a oportunidade de acessarem a ferramenta e realizarem o processo de criação de códigos de qualquer lugar e a qualquer momento que desejassem, permitindo que

o trabalho fosse interrompido e continuado de onde se havia parado. Ao término das análises textuais e criações de códigos, cada participante finalizava o processo. Assim, os códigos criados pelos participantes eram salvos e foram analisados pelos avaliadores/pesquisadores.

Ao final do processo, obteve-se um total de 237 códigos avaliados, desse total cerca de 23% (54 códigos) foram criados pelos avaliadores e 77% (183 códigos) foram criados pelos participantes e aceitos pelos avaliadores, como mostra a Figura 6. Esses números evidenciam a forte contribuição da codificação colaborativa e da ferramenta *Open Coding Tool* para o desenvolvimento desse procedimento de codificação.

Figura 6. Resultados da codificação colaborativa



6. Considerações Finais

Este artigo apresentou uma ferramenta de codificação colaborativa que permite a participação de muitas pessoas (*crowd*) em procedimento de codificação de dados textuais. A ferramenta também garante que os códigos criados sejam validados por um grupo de pesquisadores mais experientes.

A ferramenta traz contribuições para a comunidade acadêmica interessada em pesquisas qualitativas, incluindo a Engenharia de Software, onde estudos dessa natureza estão crescendo, porque é gratuita, pode ser acessada pela web, permite uma configuração que amplia o conceito de colaboração existente em alguns softwares de análise qualitativa e garante que os resultados surjam em período de tempo reduzido com a qualidade necessária.

A ferramenta está operacional, com as funcionalidades já apresentadas, no entanto, testes adicionais, ajustes e novas funcionalidades devem ser realizados, de forma a torná-la disponível para a ampla utilização pela comunidade acadêmica. Como trabalhos futuros de desenvolvimento, elenca-se a disponibilização da ferramenta em inglês, a possibilidade de inserção de vários projetos de análise de dados qualitativos, assim como a criação de agrupamentos de códigos em categorias e o relacionamento entre categorias, permitindo que outros procedimentos presentes em métodos qualitativos, como a categorização no método de Análise de Conteúdo e a codificação axial da Teoria Fundamentada em Dados sejam contemplados. Posteriormente pretende-se avançar com a implementação da coleta e extração semi-automática de trechos de dados textuais, que servirão de insumo aos procedimentos de análise de dados qualitativos.

Referências

Bardin, L. (1977). Análise de conteúdo. *Lisboa: edições*, 70:225.

Boyatzis, R. E. (1998). *Transforming qualitative information: Thematic analysis and code development*. sage.

Braun, V. and Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2):77–101.

Chervenski, A. S. and Bordin, A. S. (2020). Understanding Legacy Systems in the Light of Grounded Theory. *34th Brazilian Symposium on Software Engineering (SBES '20)*.

Creswell, J. W. (2014). *Investigação Qualitativa e Projeto de Pesquisa-: Escolhendo entre Cinco Abordagens*. Penso Editora.

Denzin, N. K. and Lincoln, Y. S. (2005). The discipline and practice of qualitative research introduction. *The landscape of qualitative research*, pages 1–43.

Dybå, T., Prikladnicki, R., Rönkkö, K., Seaman, C., and Sillito, J. (2011). Qualitative research in software engineering. *Empirical Software Engineering*, 16(4):425–429.

Elliott, V. (2018). Thinking about the coding process in qualitative data analysis. *The Qualitative Report*, 23(11):2850–2861.

Gibbs, G. (2009). *Análise de dados qualitativos: coleção pesquisa qualitativa*. Bookman Editora.

Hoda, R., Noble, J., and Marshall, S. (2012). Developing a grounded theory to explain the practices of self-organizing agile teams. *Empirical Software Engineering*, 17(6):609–639.

Mayring, P. (2004). Qualitative content analysis. *A companion to qualitative research*, 1(2004):159–176.

McLeod, L., MacDonell, S. G., and Doolin, B. (2011). Qualitative research on software development: a longitudinal case study methodology. *Empirical software engineering*, 16(4):430–459.

Moreira, D. A. (2007). O uso de programas de computador na análise qualitativa: oportunidades, vantagens e desvantagens. *Revista de Negócios*, 12(2):56–58.

Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic mapping studies in software engineering. *EASE'08 Proceedings of the 12th international conference on Evaluation and Assessment in Software Engineering*, pages 68–77.

Salinger, S., Zieris, F., and Prechelt, L. (2013). Liberating pair programming research from the oppressive driver/observer regime. In *2013 35th International Conference on Software Engineering (ICSE)*, pages 1201–1204. IEEE.

Strauss, A. and Corbin, J. (2008). *Pesquisa qualitativa: técnicas e procedimentos para o desenvolvimento de teoria fundamentada.* 2ª ed. Porto Alegre (RS): Artmed.