UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

CAMPUS CORNÉLIO PROCÓPIO

DIRETORIA DE GRADUAÇÃO E EDUCAÇÃO PROFISSIONAL

DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO

CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

RICARDO CARREIRO ALVIM

**DESENVOLVIMENTO DE UMA FERRAMENTA DE LEVANTAMENTO DE REQUISITOS E CROWDSOURCING COM BASE NO MODELO R4C**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO

2017

RICARDO CARREIRO ALVIM

**DESENVOLVIMENTO DE UMA FERRAMENTA DE LEVANTAMENTO DE REQUISITOS COM BASE NO MODELO R4C**

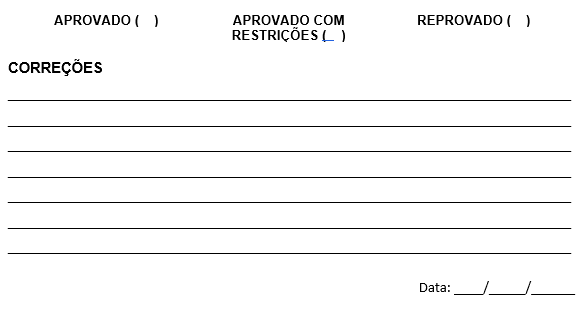
Trabalho de Conclusão de Curso de graduação, apresentado à disciplina Trabalho de Conclusão de Curso, do curso de Análise e Desenvolvimento de Sistemas da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para a obtenção do título de Tecnólogo.

Orientador: Prof. Dr. Alexandre L’Erário

CORNÉLIO PROCÓPIO

2017





**AGRADECIMENTOS**

Gostaria de agradecer primeiramente aqueles que participaram do processo de crowdsourcing, com suas respostas e tempo despendido para colaborarem com este trabalho.

Também agradeço imensamente a meus pais, o Carlos Roberto Pereira Alvim que me ensinou boa parte da bagagem técnica sobre tecnologia, eletrônica, mecânica e automação, bem como me recolheu de volta quando retornei para Assis em 2014, a minha mãe Maria Eloisa Carreiro que colaborou com o processo e indicou para que diversas pessoas para que colaborassem com a pesquisa.

Agradeço também ao Professor Doutor Alexandre L’Erário que me apoiou nas dificuldades do projeto, me orientou para que eu desenvolvesse esse projeto e me direcionou em todas as etapas do projeto. Também agradeço a Fundação Educacional do Município de Assis, a FEMA, que embora não tenha envolvimento com o projeto, foi onde passei as noites de escrita e desenvolvimento do projeto e ponto de reunião com o L’Erário.

Agradeço também meus amigos mais próximos que colaboraram para que mais pessoas colaborassem para o processo de crowd como o Ademir Ricieri, a Aime Reis, o Giliard Almeida de Godoi, o Athos Castro Moreno, o João Roberto Campana, o Rafael Pradella, a Priscilla Tiezzi Fernandes e muitos outros que acreditaram na ideia e colaboraram diretamente e indiretamente.

Também deixo aqui meus agradecimentos para meus companheiros de trabalho da FIT Soluções, que me receberam de braços abertos para colaborar com os mesmos durante o desenvolvimento do projeto, assim ampliando minha visão de mercado e tecnologia.

Obrigado todos vocês que colaboraram para que mais essa etapa seja concluída.

**RESUMO**

ALVIM, Ricardo Carreiro. **Desenvolvimento de uma ferramenta de levantamento de requisitos com base no modelo r4c**. 2017. 29 f. Trabalho de Conclusão de Curso (Graduação) – Tecnologia em Análise e Desenvolvimento de Sistemas. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2017.

Este projeto trata da relação entre o crowdsourcing e suas interações em uma grande rede e a engenharia de requisitos, onde os processos de elicitações podem ser facilitados quando explorados em uma grande rede, contanto que as adaptações e ajustes sejam realizados. Estas ações entre o levantamento de requisitos, crowdsourcing e o framework Scrum foram relacionados por RIBEIRO (2016) que propôs um modelo chamado de r4c. Porém este projeto não possui uma ferramenta dedidcada para realizar as ações realizadas, sendo este o proposito deste trabalho, tendo como base o modelo r4c, o desenvolvimento de uma ferramenta para auxiliar no processo de crowdsourcing

**Palavras-chave:** Crowdsourcing, Engenharia de Requisitos, Elicitação, Engenharia de Software, r4c, Crowdsourcing, Scrum Solo.

**ABSTRACT**

ALVIM, RICARDO. **DEVELOPMENT A CROWDSOURCING TOOL TO REQUIREMENTS SURVEY BASED ON A R4C MODEL**. 2017. 29 f. Trabalho de Conclusão de Curso (Graduação) – Tecnologia em Análise e Desenvolvimento de Sistemas. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2017

This Project is about relationship with crowdsourcing and it iterations at a big network and requirement enginner, where the elicitation process can be simplyfied hen explored in a big network, on condition that adaptations and adjustments are made. These actions between the requirements survey, crowdsourcing and Scrum were related by RIBEIRO (2016) who proposed a model called r4c. However, this project does not have a dedicated tool to carry out the actions carried out, this being the purpose of this work, based on the r4c model, the development of a tool to assist in the crowdsourcing process

**Keywords**: Crowdsourcing, Requirements Engineering, Elicitation, Software Engineering, r4c, Crowdsourcing, Scrum Solo.

**LISTA DE IMAGENS**

Figura 1. Primeiro release do banco de dados 23

Figura 2. Segundo release do banco de dados. 24

Figura 3. Tela do processo de crowdsourcing 25

Figura 4. Tela inicial do administrador 26

Figura 5. Padrões de Resposta 29

Figura 6. Visão Geral das Tags 30

Figura 7. Gráfico de visualização 30

**LISTA DE SIGLAS**

|  |  |
| --- | --- |
| API | Aplication Program Interface |
| JPA | Java Persistence Application |
| UML | Unified Modeling Language |
| SRS | Software Requirements Specifications |
| REST | REprestational State Transferer |
| MER | Modelo Entidade Relacionamento |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Sumário**

[1.1. INTRODUÇÃO 10](#_Toc483782034)

[1.2. PROBLEMA 11](#_Toc483782035)

[1.3. JUSTIFICATIVA 11](#_Toc483782036)

[1.4. OBJETIVOS 12](#_Toc483782037)

[1.4.1. OBJETIVO GERAL 12](#_Toc483782038)

[1.5.2. OBJETIVOS ESPECÍFICOS 12](#_Toc483782039)

[1.5.3. ESTRUTURA DO TRABALHO 12](#_Toc483782040)

[2. FUNDAMENTAÇÃO TEÓRICA 13](#_Toc483782041)

[2.1. CROWDSOURCING 13](#_Toc483782042)

[2.2. ENGENHARIA DE REQUISITOS 15](#_Toc483782043)

[2.3. REQUISITOS EM CROWDSOURCING 16](#_Toc483782044)

[2.4. TRABALHOS RELACIONADOS 17](#_Toc483782045)

[3. MÉTODOS E PROCEDIMENTOS 20](#_Toc483782046)

[3.1. SCRUM SOLO 20](#_Toc483782047)

[3.2. TESTE EXPERIMENTAL 21](#_Toc483782048)

[3.3. ARQUITETURA CANDIDATA 22](#_Toc483782049)

[3.3.1. MODEL VIEW CONTROLLER 22](#_Toc483782050)

[3.3.2. REST API 22](#_Toc483782051)

[4. DESENVOLVIMENTO DO APP 24](#_Toc483782052)

[5. TESTE EXPERIMENTAL 29](#_Toc483782053)

[6. CONCLUSÃO 35](#_Toc483782054)

[6.1. TRABALHOS FUTUROS 35](#_Toc483782055)

## **INTRODUÇÃO**

A engenharia de requisitos é um tópico de grande relevância para a Engenharia de Software, sendo que um dos problemas frequentes no desenvolvimento do software é a elicitação de requisitos (PRESSMAN, s/a).

A elicitação é uma etapa de desenvolvimento para levantar as informações necessárias para o desenvolvimento de um software. Ao longo dos anos a Engenharia de Software evoluiu em formas de processo de software, passando de modelos cascata, evoluindo para modelos incrementais e atualmente, modelos agilistas. Todos eles desenvolvidos com o propósito de melhorar o processamento de desenvolvimento de software e também os problemas relacionados a volatilidade de requisitos, como isso impacta em custos de produção, organização da equipe e o racionamento de recursos.

De início acreditou-se que as mesmas premissas dos modelos tradicionais de engenharia civil onde o projeto era desenvolvido do início ao fim sendo totalmente planejado antes da construção do mesmo. Com a evolução dos processos de software, este tipo de solução se mostrou insuficiente e necessário ser divididas em etapas, com mecanismos de validação, dando assim origem em processos interativos e incrementais que juntamente com experiências de mercado se transformaram em processos agilistas como o Scrum, onde a gestão do projeto é priorizada.

Contudo, paralelamente a esta evolução, ocorreram também mudanças significativas para a indústria de software como a criação do sistema operacional Linux, do movimento de *software* livre e a mídia de web em que o usuário e entusiasta de um software, produto ou serviço deixa de ser um mero protagonista no processo de evolução ou construção de um produto, para passar a ser um protagonista do mesmo, onde ele pode colaborar diretamente ou indiretamente com evoluções no mesmo, seja por *feedbacks* ou por colaborar com alterações de códigos fontes existentes, melhorando-os e os submetendo em um processo de aprovação em comunidades onde estas alterações e sugestões são julgadas e aplicadas as devidas soluções. Este ato de colaboração em uma grande rede, onde as informações podem vir de diversos tipos de usuário e pode ser utilizada para a melhora de produtos e serviços é chamada de crowdsourcing.

Mesmo *softwares* de código fechado estão mudando suas concepções para terem mecanismos de *feedback* e sugestões de mudanças dos usuários, onde as mesmas são ponderadas e podem serem acrescentadas, melhoradas ou resolvidas. Um exemplo atual do uso de *crowdsourcing* é o Windows Insider Preview, que é uma versão *rolling release* do sistema operacional Windows onde existe um mecanismo de obtenção de *feedback* de usuários sobre as alterações do sistema, reportagem de problemas e sugestão de novas funcionalidades. Estas alterações são analisadas por uma equipe da Microsoft que pode ou não adicionar uma sugestão, correção ou alteração enviada por um usuário Windows nas próximas builds ou de testes do sistema operacional.

Por tanto, o *crowdsourcing* já é uma realidade associada a engenharia de requisitos, e como poderá ser visto no capítulo dedicado ao tema de *crowdsourcing*, existem diversas soluções de mercado ou da academia para o gerenciamento de requisitos, sendo uma delas o modelo r4c que será analisado adiante.

## **PROBLEMA**

O problema apontado por esta proposta é o acompanhamento de requisitos de software por meio de modelo de *crowdsourcing* por meio do desenvolvimento de uma ferramenta.

## **JUSTIFICATIVA**

O trabalho justifica-se pelo número reduzido de propostas de ferramentas de *crowdsourcing* conforme será descrito a seguir em uma pesquisa na IEEE Explore. A falta de uma ferramenta de apoio também se faz necessário conforme relatou RIBEIRO (2016) em seu teste experimental de *crowdsourcing* em que desenvolveu o modelo r4c, que servirá de modelo para o desenvolvimento de uma ferramenta de *crowdsourcing* e requisitos.

Isso aliado aos potenciais da chamada grande rede e o uso do conceito do crowdsourcing em desenvolvimento de software livre e em software proprietário comercial, como é o caso do Windows justificam e evidenciam a necessidade de uma ferramenta nos moldes.

## **OBJETIVOS**

## **OBJETIVO GERAL**

O objetivo geral deste projeto é desenvolver uma ferramenta para requisitos em *crowdsoucing* baseado no modelo r4c desenvolvido por RIBEIRO (2016).

## **OBJETIVOS ESPECÍFICOS**

Esta proposta tem como objetivos específicos do desenvolvimento da ferramenta de gestão de processos:

1. Compreender o processo de *crowdsoucing*;
2. Compreender o modelo r4c;
3. Desenvolver uma ferramenta com base no r4c; e
4. Observar as interações com o Scrum;

## **ESTRUTURA DO TRABALHO**

Este trabalho está estruturado em capítulos onde o capítulo 2 descreve a fundamentação teórica sobre *crowdsourcing*, engenharia de requisitos e requisitos em *crowdsourcing*, onde são apresentados os conceitos elementares sobre os mesmos e descrevendo como os mesmos se associam.

O capítulo 3 descreve os métodos e procedimentos utilizados pelo projeto como o Scrum Solo e o Teste Experimental. O capítulo 4 descreve o desenvolvimento da ferramenta, seus processos e tecnologias utilizadas. Já o capítulo 5 descreve o teste experimental realizado e no capítulo 6, os resultados do teste são apresentados. No capítulo 7, é concluída as informações do projeto e apresentando os resultados obtidos.

## **FUNDAMENTAÇÃO TEÓRICA**

Este capítulo apresenta uma visão geral dos temas de *crowdsourcing*, engenharia de requisitos e requisitos, demonstrando a associação entre os temas e a relevância dos mesmos, apresentando conceitos acadêmicos e soluções de mercado associada aos mesmos.

## **CROWDSOURCING**

Crowdsourcing é um termo cunhado por Jeff Howe, em 2006, para representar o processo de obtenção de ideias, serviços e conteúdo em um canal de grande público tendo em mente a internet onde estas características e ações se desenvolviam rapidamente graças as novas possibilidades na internet.

O termo foi criado em um contexto de que serviços da web como a Wikipédia, as distribuições Linux e demais projetos *open source*. Além disso, na época, também estava em expansão serviços na web como as redes sociais, devido ao advento da Web 2.0, que segundo (O’RELLY, 2005) permitiu a evolução para uma web de conteúdos dinâmicos e alta interatividade. Junto com elas, segundo (Leite et al., 2013) temos o surgimento de blogs, plataformas de compartilhamento de vídeos, *wikis* e outras aplicações voltadas a interatividade com usuários. Esta interatividade descrita por (Leite et al., 2013) caracterizam alguns pontos dos pilares do *crowdsourcing* que veremos adiante.

Segundo (BRABHAM, DAREN, 2008) o *crowdsourcing* é um ato de uma empresa ou organização transferir as funções realizadas por uma rede indefinida, tornando assim este trabalho de forma colaborativa e distribuída.

Estas pessoas da chamada rede indefinida, segundo (POENG, BABAR. 2014) se tornaram prestadoras de tarefas para estas organizações, por isto o processo de *crowdsourcing* também é visto como um processo de terceirização. No IEE Explorer da IEEE, uma das mais relevantes bases sobre o escopo deste projeto, que é a engenharia de software, possui mais de 900 artigos relacionados ao tema *crowdsourcing* sendo que estes números crescem com o passar dos anos, comprovando a importância do tema.

Além da redução de custos, o *crowdsourcing* é um processo de obtenção de *feedback* de usuários, contudo, estes usuários, segundo HOSSAIN (2012) devem estar cientes de tudo que ocorre no projeto e na equipe, sendo chamado este processo de Consciência em Grupo.

O *crowdsourcing*, contudo, pode ser classificado por suas atividades e seus pilares. Segundo (HOSSEIN et al., 2005) podem ser divididas em quatro pilares, que serão detalhadas a seguir.

O primeiro pilar é relacionado as pessoas envolvidas, que constituem as atividades do processo (HOSSEIN et al. 2014). O segundo pilar é a tarefa propriamente dita que irá atrair o público que irá participar da atividade de *crowdsourcing*. O quarto pilar é o sistema, meio ou canal onde ocorrerá os processos de obtenção de informações que podem ou não ser um meio digital (HOSSEIN et al, 2014)

Além dos pilares, também há 5 características principais como a diversidade, a *know-less, largeness* e a *suitability*. Diversidade refere-se a distinção entre as pessoas que participam do processo, classificados em gênero, localidade e faixa etária.

A segunda característica chamada de *unknow-less* que trata do anonimato, contudo, este pode ser subdividido em dois tipos, quando o *crowd* não sabe quem está realizando o processo ou quando um dado participante não conhece os demais.

Já a característica *largeness* refere-se ao tamanho da abrangência de colaboração que em contextos muito abrangentes não se deve fazer com que o *crowd* assuma muitas responsabilidades para evitar confusão na multidão. A quarta característica é a *suitability* que se refere aos ajustes necessários quanto as habilidades dos usuários para realizar uma tarefa (HOSSEIN et al, 2014).

Das características mais detalhadas e importantes para o escopo deste projeto se trata das características do segundo pilar do *crowdsourcer*. Esta seria relacionada aos incentivos que seria o ânimo para desempenhar o processo onde o *crowdsourcer* pode fornecer incentivos como financeiro, social ou de entretenimento. Outra característica do segundo pilar refere-se a audiência aberta ao público em geral ou ao menos que tenham relevância ao escopo do projeto. (HOSSEIN et al., 2015).

A partir de estudos associando as áreas da engenharia de software e o *crowdsourcing*, analisou-se que, o uso dos recursos de obtenção de feedback do *crowdsourcing* pode facilitar o processo de levantamento de requisitos de modo a atender as necessidades dos usuários projetando um software sob medida. Estes estudos estão relacionados com *empirical validation*, *requirements identification*, *stakeholders discovery*, *requirements engineering*, *requirement-driven social adaptation* e *feedback-based requirement* (HOSSEIN et al., 2015)

## **ENGENHARIA DE REQUISITOS**

Segundo (PRESSMAN, s/a) a Engenharia de Requisitos pode ser descrita como uma ação associada a Engenharia de Software que é iniciada na atividade de comunicação e continua na modelagem.

Em outras palavras, a Engenharia de Requisitos permite entender as necessidades do cliente não apenas sob aspecto funcional, isto é, das funcionalidades de um software, mas também as consequências de determinadas escolhas no projeto de software.

Segundo (CHRISTELE, KONG, s/a), já no levantamento de requisitos os problemas ocorrem, por problemas no escopo, tendo a má definição dos limites do projeto ou então com detalhes desnecessários ou imprecisos, gerando confusão e problemas de entendimento. Também ocorrem problemas quando os clientes não possuem o total domínio dos limites e capacidades de seus sistemas computacionais. Outro problema é quando o cliente ou usuário omite certos requisitos por considerarem certas informações obvias demais, acreditando que o analista já tenha todo o conhecimento do domínio e regras de negócios internalizadas pela empresa ou organização.

Esta questão do domínio é um problema, pois se por um lado um analista com conhecimentos do domínio do projeto ajuda a melhorar na precisão dos requisitos, sabendo assim ponderar e adequar todas as perguntas ao escopo do projeto, isso torna os custos de projeto mais altos devido a uma mão de obra mais qualificada.

Um outro problema da engenharia de requisitos é a volatilidade dos requisitos, que podem alterar a qualquer momento no projeto. Sendo justamente por isto, metodologias de processo de desenvolvimento de software e frameworks de processos com tendência agilista tentam criar mecanismos onde a alteração de requisitos seja menos custosa, previsível e em certos casos, como no *framework* Scrum, desejável.

No entanto, todo processos de desenvolvimento de software passa por uma etapa onde são especificados os requisitos. Segundo (PRESSMAN, s/a) esta especificação pode ser um documento textual, um conjunto gráfico, cenários de uso, protótipos, modelagem matemática ou combinação destes.

Um dos modelos citados por PRESSMAN é o SRS, *Software Requirements Specifications*, que em tradução livre seria Especificação de Requisitos de Software que é um modelo de documento escrito detalhando todos os aspectos do software desenvolvido por Karl Wiegers da Process Impact Inc.

Este modelo tenta mapear os aspectos do software como proposta, escopo, referências, perspectiva, característica, classes de usuário, ambiente operacional, restrições, documentos para usuários, hipóteses, características, interfaces e requisitos não funcionais; assim passando por todos os aspectos possíveis de um software.

Entender como estes requisitos fazem parte do projeto do software é importante para compreender um modelo como o r4c para a obtenção de requisitos por meio de uma grande rede e transformar então os requisitos em informação relevante para o projeto de software.

## **REQUISITOS EM CROWDSOURCING**

Dado o caráter do crowdsourcing de sua ampla disponibilidade em uma grande rede, chamada por Howe de uma maneira de terceirizar a solução de problemas fazendo uso da inteligência coletiva (HOWE, 2006) sendo constituídos pelos pilares do crowdsourcing.

Segundo HOSSEIN et al (2015) o crowdsourcing pode colaborar na etapa de elicitação de requisitos no desenvolvimento de um projeto, em especial em sistemas utilizados por um grande número de usuários que possam colaborar com o processo. A partir dos estudos descritos, pode-se afirmar que o crowdsourcing auxilia na eficiência da elicitação de requisitos, onde a sabedoria coletiva pode auxiliar no processo de elicitação e atender de maneira precisa e concisa as necessidades do cliente do projeto de software.

## **TRABALHOS RELACIONADOS**

Os trabalhos relacionados podem ser categorizados em trabalhos acadêmicos e softwares que fazem uso dos conceitos do crowdsourcing para obtenção de requisitos.

Dos artigos acadêmicos temos o trabalho de RIBEIRO (2016) que específica o modelo r4c para uso de requisitos em crowdsourcing, sendo que este artigo se baseia no de HOSSEIN et al (2015) que apresenta maneiras para o ajuste do crowdsourcing para aperfeiçoar a elicitação de requisitos. Neste trabalho houve uma atividade de levantamento de requisitos por meio da ferramenta Google Forms com a questão “Para você o que um celular deve possuir?” solicitando aos alunos do curso de Engenharia Elétrica gerasse 10 requisitos. Nos resultados foi verificado que, embora o processo r4c possua aderência a proposta de gerenciar requisitos seja verdadeira, o processo manual inviabiliza o procedimento, sendo inclusive um dos trabalhos futuros sugeridos, a criação de uma ferramenta automática ou semi automática para a melhoria do mesmo.

HOSSEIN et al (2015) defende o uso do crowdsourcing na etapa de elicitação apresentando em seu artigo formas de realizar o ajuste para o aprimoramento de requisitos e para isso, realizou um experimento onde envolveu dois grupos de 14 participantes, envolvendo usuário e desenvolvedores. Além disso foram envolvidas outras 34 pessoas da comunidade de engenharia de requisitos. Na execução, os aspectos humanos foram divididos e mapeados de modo a se estabelecer como seria realizado o processo onde os formulários aplicados foram os mesmos, tendo apenas as variações de nível e conhecimento, demonstrando assim a preocupação com os diferentes níveis de conhecimento e visão dos envolvidos, estando assim alinhado o processo de modo a se extrair de forma adequada as informações dos usuários.

Os tempos de execução foram calculados e dois avaliadores avaliaram os resultados, com um terceiro observando para evitar problemas. No final do processo, observou-se a ligação entre crowdsourcing e atributos de qualidade induzido com 34 relações classificadas em 10 categorias, sendo que cada uma delas expressava uma característica do crowdsourcing.

Considerando que não haviam conhecimentos suficientes para aumentar a qualidade dos requisitos induzidos, foi confirmada a relação entre o Crowd e requisitos.

Forno e Zanatta (2016) realizaram uma proposta de processo de testes funcionais onde este processo foi dividido em três etapas, sendo que a primeira contemplou o teste de Somerville (2011), para investigar as tarefas que poderiam ser executadas fazendo uso do crowdsourcing. Em seguida foram aplicadas as correções para permitir o uso do crowdsourcing no processo e por fim, a execução propriamente dita em uma empresa de desenvolvimento de software, tendo como alvo um sistema ERP (Enterprise Resource Planning) onde foram elaborados 50 casos de teste que, na execução do projeto conforme o modelo proposto dos autores e os mesmos seriam executados simultaneamente e na ocasião da publicação, a configuração do ambiente estava em andamento.

Contudo, mesmo, não tendo sido finalizado e com maiores resultados, os autores concluem que a proposta era viável e inovadora por permitir que usuários externos pudessem colaborar com a evolução do produto. Este tipo de abordagem de testes dirigidos por feedback é utilizado por softwares como o Windows em seu programa chamado de Windows Insider Preview. Além do Windows Insider Preview, também há o Fiat Mio, plataforma onde clientes da Fiat poderiam opinar sobre o modelo Mio, que é um *city car* futurista. O carro conceito foi construído sob a ação do crowdsourcing, uma evolução do que já é adotado na indústria automobilística e na produção de novos produtos com as chamadas clínicas.

Já VLAANDERN et al (2010) realizaram uma aplicação real com o propósito de mostrar como os métodos ágeis podem trabalhar em conjunto com a gestão de produto. As experiências relatadas no artigo de VLAANDEER et al em uma empresa fazem o uso do framework Scrum, destacando os ajustes aplicados e como a partir do caso de uso podem ser ajustados, aplicados e mensurados em um cenário real.

A primeira etapa onde se levanta as ideias da empresa ou cliente é chamado de *vision*. Outra etapa é a *theme* onde ocorre a execução formal tornando assim a visão com mais detalhes.

Posteriormente ocorre a definição de requisitos, dividida em três etapas onde a primeira é realizada pela equipe de produtos de software, chamada de SPM, a equipe de SPM traduz os conceitos em listas realizando os ajustes necessários entre os requisitos. Após isso a equipe de desenvolvimento elabora os requisitos, onde podem ser criados documentos para clarificar o processo.

Conclui-se que é necessário, para que a equipe SPM ágil desenvolva com sucesso as tarefas do projeto, determinar o tamanho das tarefas, estruturas de backlog e realizar o refinamento dos requisitos.

Além destes trabalhos, existem outros no IEE Explore, sendo que a grande maioria deles, não existe aplicação prática.

## **MÉTODOS E PROCEDIMENTOS**

Este capítulo será dedicado para apresentação geral do processo de software, tecnologias e arquitetura escolhidas para o desenvolvimento desta proposta de projeto.

## **SCRUM SOLO**

A citação do scrum solo está em pagoto,2016, ([10.1109/CISTI.2016.7521555](https://doi.org/10.1109/CISTI.2016.7521555)).

Scrum é uma metodologia de desenvolvimento ágil, interativo e incremental usado para o processo de desenvolvimento de software. Foi concebido, segundo SILVA (2014) nos anos 1990 por Jeff Sutherland, John Scummiotales e Jeff McKenna quando os modelos de processo tradicionais atingiram seus limites.

O Scrum Solo é uma metodologia baseada no Scrum, com a ênfase em equipes pequenas ou de apenas um desenvolvedor, ou seja, o número de pessoas que irão consumir as informações dos artefatos do processo será baixo (PAGOTO, 2016). Algumas características gerais do Scrum são mantidas como o *Product Backlog*, que é uma avaliação geral do escopo do projeto, e o *Sprint Backlog*, que é uma avaliação geral das atividades de uma determinada Sprint.

As mudanças passam a existir quanto ao número de reuniões, no Scrum Solo possui apenas reuniões semanais, diferentemente das reuniões diárias do Scrum. Porém, no final de cada Sprint, que em média tem uma duração de duas semanas, o desenvolvedor deverá entregar um protótipo funcional dos requisitos coletados pela aquela Sprint. (L’ERÁRIO et al, 2016)

Sendo assim, o Scrum Solo se mostra uma alternativa adequada para um projeto com as características descritas da ferramenta proposta, além das possibilidades de os requisitos serem alterados e a capacidade de gestão do processo citado.

## **TESTE EXPERIMENTAL**

O método de pesquisa deste trabalho onde o processo de *crowdsourcing* é executado na prática é chamado de experimental, que segundo Travassos (2002) é o centro do processo científico, pois é através de experimento que se verificam teorias, explorando fatos e fazendo com que teorias sejam aprimoradas.

O processo consiste em levantar um novo modelo, que pode ou não ser necessariamente baseado em um existente e assim realizar um estudo do efeito ou produto sugerido por este modelo.

O autor afirma que a experimentação oferece um modo sistemático, disciplinado, mensurável e controlado para as atividades humanas e devido a isso, novos métodos, técnicas, linguagens e ferramentas, não deveriam ser apenas sugeridos, publicados ou apresentados sem experimentação, segundo o autor, para que isso seja realizado, é preciso avaliar novas metodologias em comparação com as existentes.

Segundo L’Erario (2009) a ideia central do processo experimental, é o controle; relacionado ao ambiente onde o teste está sendo feito e suas variáveis, a mensuração; quanto a questão quantitativa do projeto e a simulação de um modelo representativo real.

No caso deste projeto, o modelo de crowdsourcing adotado foi validado por um processo experimental, onde Ribeiro (2016) realizou o processo de crowdsourcing com seus participantes e validou o mesmo na obtenção de requisitos de maneira manual em seu ambiente.

## **ARQUITETURA CANDIDATA**

A seguir neste subcapítulo será apresentada as arquiteturas da aplicação como a *Model View Controller* e o REST.

## **MODEL VIEW CONTROLLER**

Model View Controller, MVC, é um padrão de projeto que visa separar os projetos de software em três camadas a Visão, onde os dados são apresentados para o usuário; o Modelo, onde ficarão as classes de modelo de negócios e a modelagem da abstração dos dados e o Controlador, onde é feita a comunicação com o banco de dados entre o Modelo e a Visão.

Este projeto adota o MVC como estilo arquitetural de sua versão web e de seus webservices, sendo que este segue o estilo arquitetural REST API que será descrito a seguir.

## **REST API**

REST é um acrônimo para *REpresentational State Transferer*, que em tradução livre seria Transferência de Representação de Estado, é um padrão arquitetural de comunicação (CAMPOMORI, 2017).

Foi criado por Roy Fielding tendo como base o estilo arquitetural HTTP da web, uma vez que através dos métodos tradicionais do padrão HTTP seria possível a intercomunicação na web. Nesse sentido, o padrão REST é uma implementação deste padrão HTTP que se utiliza de seus métodos para realizar comunicação através de interfaces de comunicação com *web services* (MASSE, 2011).

O padrão HTTP possui algumas características como, ser um protocolo cliente-servidor, onde há uma conexão entre a máquina cliente e a máquina servidora do serviço, recurso ou aplicação. A comunicação entre cliente e servidor ocorre com mensagens HTTP, onde o cliente envia mensagens do tipo *request* e tem como resposta a esta comunicação uma resposta do tipo *response*, sendo que *request* e *response* são interdependentes, um depende necessariamente do outro para ocorrer (CURIOSO, 2017).

Outra característica do padrão HTTP é que a conexão dura até o momento de *request*-*response* é concluído, ou seja, não é do tipo *keep alive*. Caso seja necessário um novo conteúdo ou correção do mesmo, é aberta uma nova conexão, sendo elas totalmente independentes e também, por padrão, o protocolo não armazena informações sobre as requisições, uma característica chamada de *stateless*.

Estas requisições são assíncronas do lado do cliente, logo, duas requisições podem ser executadas ao mesmo tempo, dependendo sempre do lado cliente definir uma ordem de sincronização, caso necessário.

Uma outra característica importante do HTTP é sua semântica para recursos e métodos que um servidor HTTP pode oferecer através *Unique Resource Identifier*, de URI que são convertidas em *Unique Resource Locator* e assim disponibilizar páginas, arquivos e outros recursos.

Geralmente os verbos de acesso a métodos do HTTP são o GET, que recupera um recurso do servidor; o POST, que insere um recurso no servidor; o PUT, que atualiza um recurso no servidor e o DELETE, que remove um recurso no servidor.

Apresentada estas informações sobre o HTTP, é necessário dizer sobre o JSON que é forma de representação de informações para *request* e *response*, utilizadas por este projeto. Também é possível no padrão REST realizar requisições *request* e *response* com o XML, contudo este é considerado mais pesado e mais verboso. Devido a aplicação necessitar respostas rápidas e comunicação simples, preferiu-se a adoção do JSON contra o XML (CAMPOMORI, 2017).

A arquitetura REST neste projeto trabalha em conjunto com a MVC, de modo que a camada de visão da aplicação é o próprio Webservice, que é consumido por uma interface HTML, como será explicado no próximo capítulo.

## **DESENVOLVIMENTO DO APP**

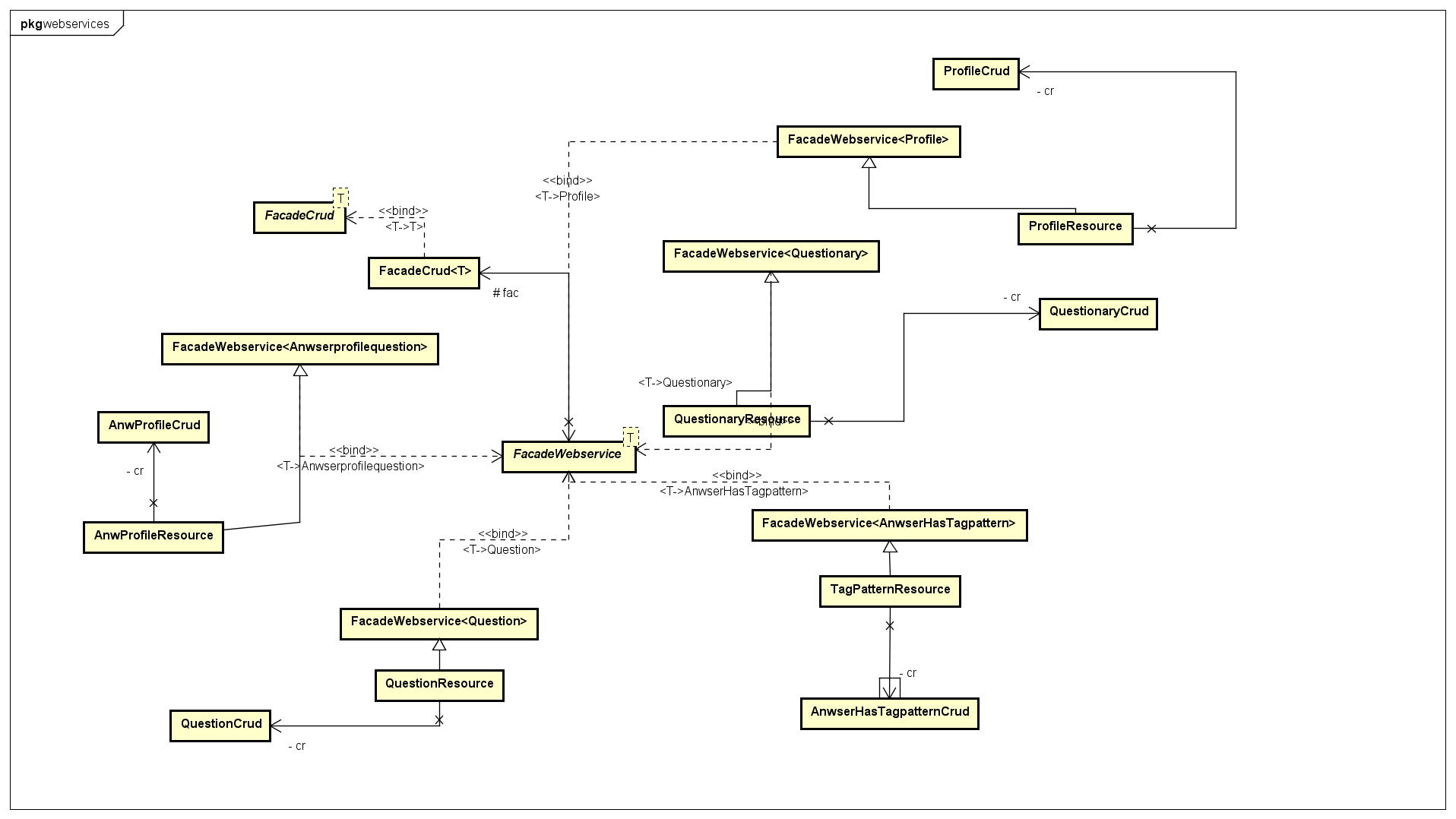
Este capítulo será dedicado para apresentação geral do processo de desenvolvimento do app, quanto a web e seu webservice integrado.

A aplicação central, onde existe a versão web e seus webservices faz uso da tecnologia e linguagem Java e JPA; para controle transacional e de mapeamento de dados da aplicação e o banco de dados Postgres.

As *Entitties*, implementadas com o JPA fazem papel da camada *Model* do padrão MVC, as classes relacionadas ao CRUD, fazem o papel do *Controller* e as classes de Webservices que fazem uso da biblioteca Jersey fazem o papel de *View* do projeto, sendo que elas interagem por meio de REST com as páginas em HTML do projeto.

A seguir uma exemplificação com um Diagrama de Classes com a interação das classes *Enitity* para mapeamento via JPA, as classes do Webservices e as classes de CRUD.

**Figura 1. Webservice e suas interações com as classes de CRUD**



**Fonte: Autoria própria**

As classes de CRUD implementam a classe genérica FacadeCrud<> que disponibiliza de forma genérica as operações básicas como o criar, recuperar, atualizar e deletar, estrutura básica do CRUD para toda a classe que o implemente.

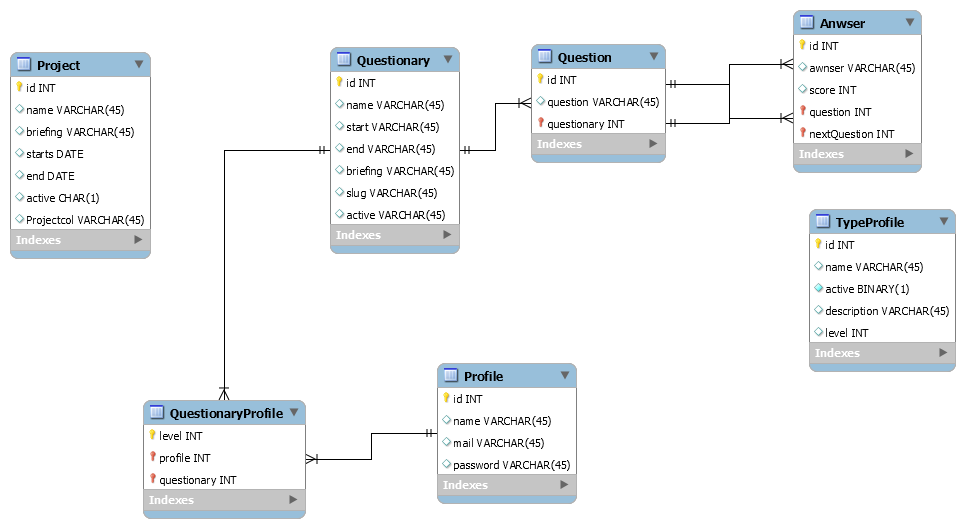
Estas classes concretas também podem ter outros métodos com outras operações de pesquisa, atualização, inserção ou remoção. Para isso, o critério de busca no banco de dados, foi feito uso das chamadas *Named Queries* do JPA onde é possível mapear o objeto do Java com uma consulta relacional SQL.

Classe genérica do *Webservice* também foi criada para realizar as operações de CRUD, sendo ela a FacadeWebservice<T> onde foram criadas as operações genéricas do HTTP como GET, POST, PUT e DELETE, disponibilizando assim estes recursos para todas as *Entities*.

Justamente pelo banco de dados ser o núcleo central da aplicação, o desenvolvimento começou pelo banco de dados, onde inicialmente previa-se uma arquitetura onde as respostas seriam de múltipla escolha. Porém, com reuniões entre o autor e o orientador, o Drº Alexandre L’Erário, decidiu-se por respostas abertas, o que exigiu alterações na estrutura do projeto. Além disso, o projeto e o tipo de perfil ainda não eram integrados as demais funcionalidades.

A seguir, o Diagrama do Primeiro *Release* do Banco de Dados, apresentado ao Alexandre L’Erário na segunda reunião do projeto, onde se teve o primeiro *release* do banco de dados.

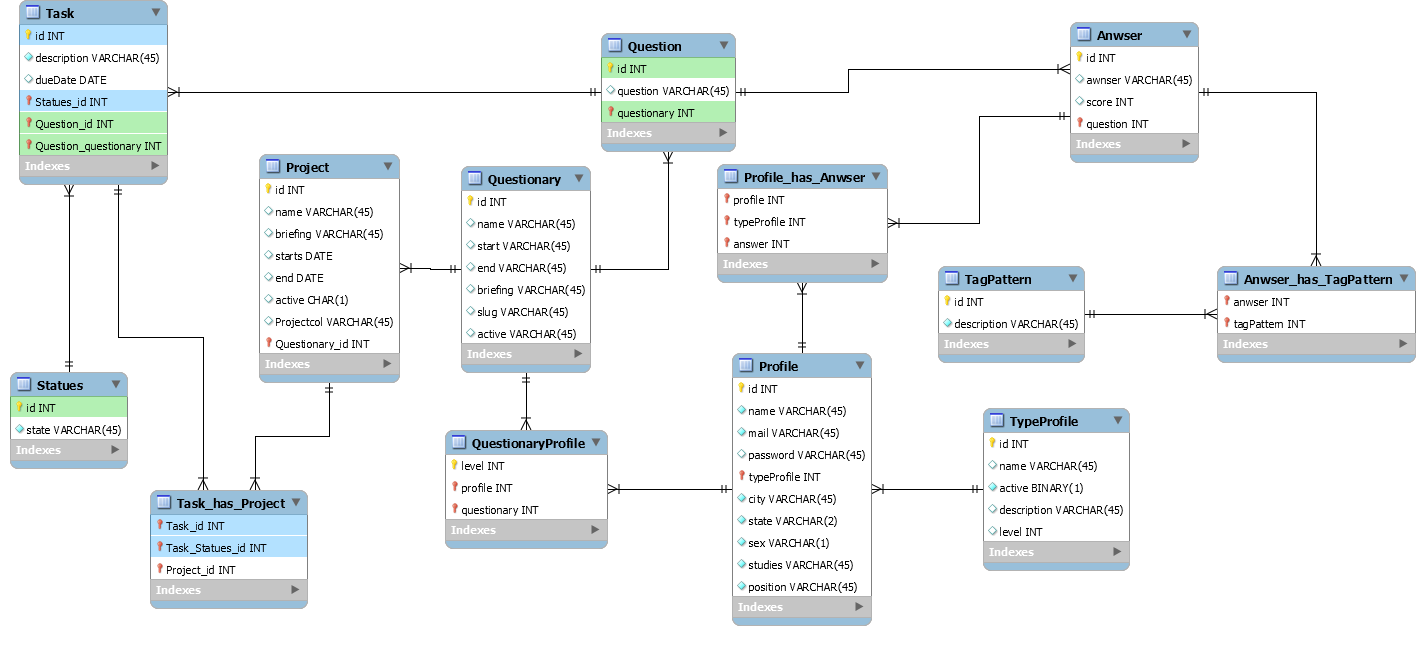
**Figura 2. Diagrama do Primeiro Release**



**Fonte: Autoria própria**

Na arquitetura desenvolvida do banco de dados do release 2 e suas camadas de persistência na aplicação principal, também foi considerada a relação de respostas, projetos e tarefas de projetos, algo que como pode ser visto na imagem a seguir.

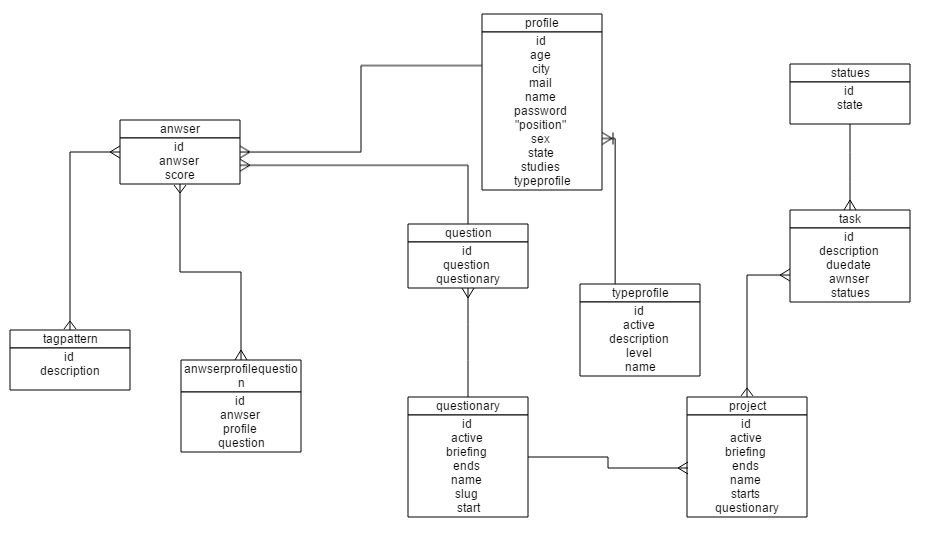
**Figura 3. Segundo Release do Banco de Dados**



**Fonte: Autoria própria**

A estrutura do banco de dados a partir do release 2 ficou melhor definida e com apenas algumas correções como veremos a seguir e a inclusão de *views*. Após esta versão, deu-se o desenvolvimento da aplicação principal, fazendo uso da ferramenta NetBeans 8.2. A seguir o Modelo Entidade Relacionamento, o MER, da terceira fase de implantação, este é o banco de dados final do projeto.

**Figura 4. Modelo Entidade Relacionamento, MER, do projeto final**



**Fonte: Autoria própria**

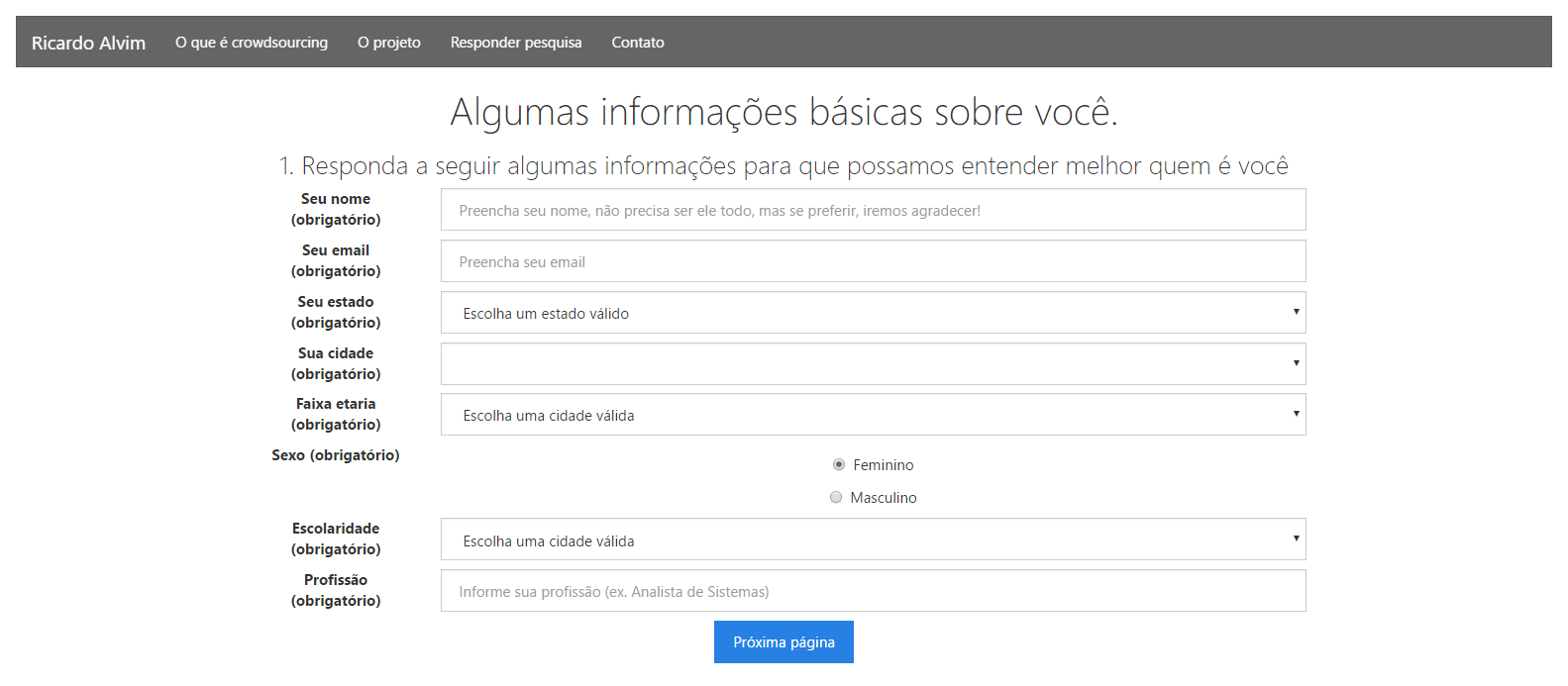
Nesta segunda fase de projeto, previa-se uma versão da aplicação para Android, porém a mesma não foi desenvolvida e foi abortada dos planos de desenvolvimento do projeto.

O mapeamento objeto-relacional se deu através de uma ferramenta wizard do Netbeans fazendo uso da tecnologia EclipseLink, derivado da especificação JPA, na versão 2.5.2. Além disso, todas as dependências do projeto são gerenciadas pelo Maven através de um arquivo XML. A versão do Java Enterprise Edition neste projeto é a 7.

O servidor de aplicação adotado é o Apache Tomcat 8.0.27.0 que acompanha a versão 8.2 do NetBeans. O serviço de webservices faz uso das bibliotecas Jersey e JAX-RS, sendo que esta última realiza a conversão dos objetos do tipo *Entity* da JPA em saída do tipo JSON, assim completando a disponibilidade de webservices para aplicação.

Na aplicação web utiliza-se de HTML 5; CSS 3; Bootwatch, um derivado do Bootstrap 3.3.7 para estilo CSS em página; Angular.js 1.5 e jQuery 1.10 sendo que estes dois últimos são em versão *minified*, que são versões compactadas dos frameworks, que tornam o carregamento das páginas mais rápido.

**Figura 5. Tela do processo de crowdsourcing do primeiro release do projeto**



**Fonte: Autoria própria**

Na versão web, na tela inicial há uma breve apresentação sobre o conceito de crowdsourcing e um botão para ir para a página O que é Crowdsourcing. A navegação se dá na barra superior onde é possível descobrir mais informações sobre este projeto, responder a pesquisa disponível e entrar em contato com o autor.

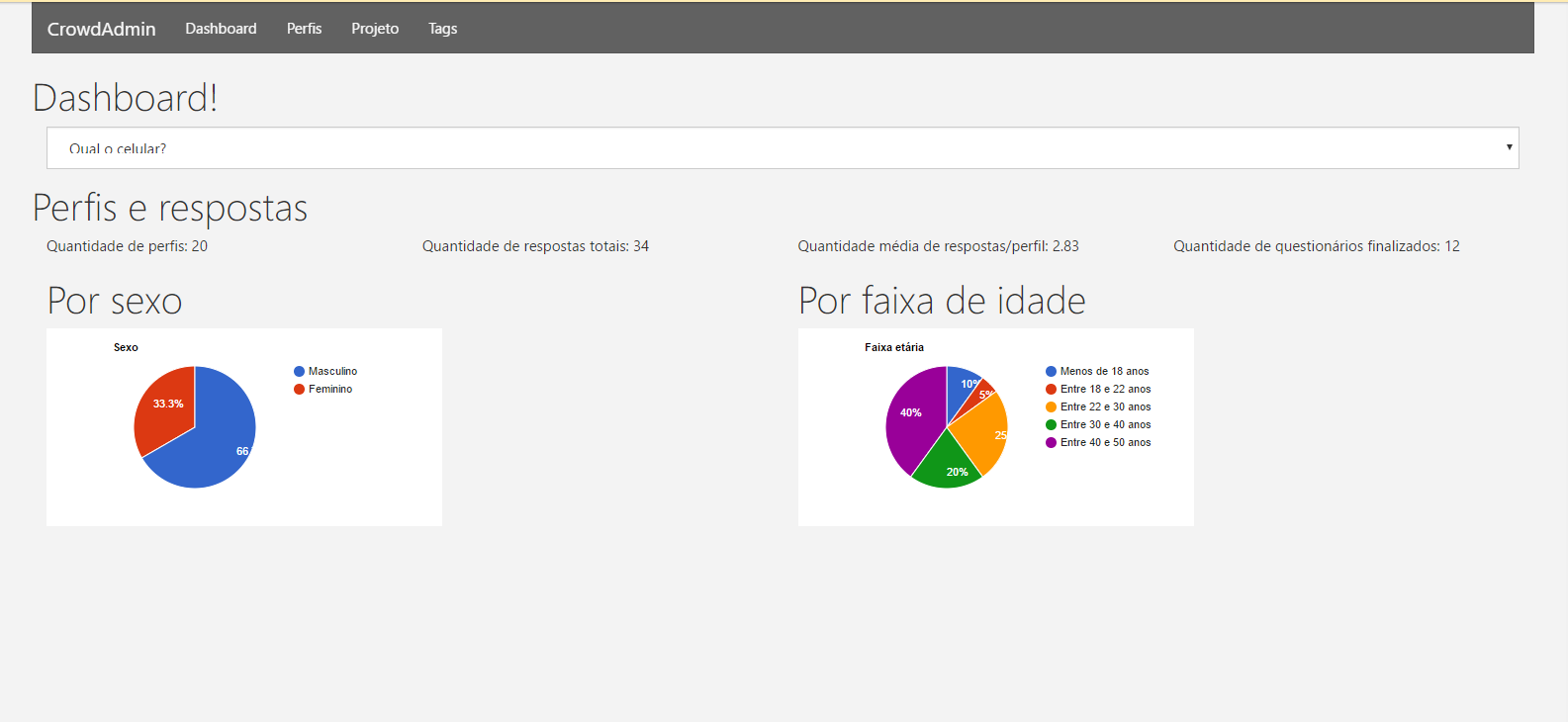
As respostas da pesquisa são temporariamente salvas em memória pelo CrowdController, o controlador do Angular.js e após todas as questões devidamente respondidas, é enviado uma requisição do tipo POST para o WebService da aplicação onde é salva todas as informações associadas ao usuário e assim finalizando o processo de *crowdsourcing*.

Na página de respostas há 3 div onde a navegação das mesmas se dá quando é apertado o botão de Próxima Página, assim reduzindo a poluição visual e deixando cada página de forma contextual com as informações que ali são exigidas.

Na página onde há as múltiplas respostas, um acionamento através de um botão verde para adicionar uma nova resposta é feito via jQuery e a remoção da resposta também é autorizada. Estas respostas são enviadas como um *array* para o *webservice*, que serão armazenadas devidamente na tabela *anwsers* no banco de dados.

Além da página de interação com o usuário, há também um painel gerencial onde é possível acompanhar as atualizações e respostas dadas pelos usuários a partir de tabelas e gráficos. Bem como é possível acompanhar inúmeros parâmetros do projeto como podemos ver a seguir:

**Figura 6. Tela inicial do administrador**



**Fonte: Autoria própria**

O projeto também contempla a visão por perfil do participante do *crowd* e de projetos, para a conversão das chamadas *tags* em tarefas de um dado projeto. O acompanhamento do projeto é feito em uma estrutura chamada Kanban, onde o estado do projeto passa pelas etapas. Para fazer, sendo feito e feito. A ênfase do projeto não se concentra nesta funcionalidade, portanto, apenas as funcionalidades de adicionar novo projeto, vinculado com o questionário; criar tarefa com base em uma *tag*, adicionar nova tarefa e mudar estado da tarefa foram contempladas. A ideia com esta funcionalidade é demonstrar a transformação das *tags* em requisitos para as etapas de projeto e não a gestão dos mesmos.

Já a verificação por perfil é para que seja analisada todas as respostas dos participantes do *crowd* individualmente, caso isso seja do interesse de quem está analisando os dados do crowdsourcing.

Com estas janelas e informações, pode-se ter uma visão ampla do crowdsourcing e seus participantes, auxiliando assim no desenvolvimento de produtos e serviços com requisitos recebidos em uma multidão de pessoas, construindo conhecimento colaborativo para auxiliar na tomada de decisão da empresa ou organização que opta pelo crowdsourcing como método de levantamento de requisitos.

## **TESTE EXPERIMENTAL**

Este capítulo será dedicado para apresentação geral do teste experimental do *app*.

Os telefones celulares tiveram início no Brasil em 1990, quando houve a primeira ligação de um telefone móvel no Brasil. Com a massificação de celulares e sua evolução, os mesmos deixaram de serem simples aparelhos de comunicação (MORIMOTO, 2009 ) e se tornarem aparelhos cada vez mais sofisticados e presentes no cotidiano dos usuários.

A *International Telecomunications Union* estimou em 2014 que 95% das pessoas no mundo terão um dispositivo móvel. E segundo (BRIGATTO, 2017) as vendas de smartphones no Brasil em 2017 irão superar em 3,5% em relação a 2016.

Tendo em vista o mercado de telefonia móvel no Brasil e no mundo, escolheu-se o mesmo tema escolhido por RIBEIRO (2016) para realizar uma pesquisa com usuários de mídias sociais sobre como seria um smartphone ideal para estes usuários.

O teste experimental foi realizado de forma livre a quem tivesse interesse em participar do processo de *crowdsourcing*, onde foi disponibilizado a versão web com otimizações para a web e dispositivos móveis.

O teste consiste em preencher informações simples da pessoa que está participando do *crowd*, como email, idade, localidade, escolaridade e área de atuação, sexo e após isso, é liberado para a primeira e única pergunta.

Esta única pergunta é de resposta aberta e permite mais de uma resposta possível no limite de até 5, contudo, se um mesmo participante quisesse participar do processo novamente utilizando o mesmo email, isso é permitido, dando assim total liberdade ao usuário em responde-las como preferir. A única limitação no caso era do limite de caracteres.

* 1. **PÚBLICO ALVO**

O link de acesso ao crowdsourcing foi liberado no Facebook no perfil do autor e compartilhado entre os participantes do *crowd* para que outras pessoas o respondessem. Onde os usuários iriam responder a mesma pergunta realizada por (RIBEIRO, 2016) que era: Como seria um celular ideal para você?

Também foi solicitado aos participantes compartilhassem a página de pesquisa para que outras pessoas participassem. Do mesmo, o público alvo cerca de 8 usuários eram mulheres e 12 homens, no todo foram 34 respostas, sendo que nem todos os 20 usuários responderam à pergunta inicial, algo que era livre para acontecer, sendo que deste total de 20, apenas 12 responderam a pergunta sobre qual seria o celular ideal.

A distribuição entre escolaridade era 0% por Ensino Médio Incompleto, 5% de Ensino Médio Completo, 50% de Ensino Superior Incompleto, 45% de Ensino Superior Completo.

As profissões mais comuns foram as de desenvolvimento de sistemas, seguido de áreas relacionadas a ensino, publicidade, meio ambiente, instrumentação e comercial, também houveram, declarações como estudante, por exemplo e a maioria, 65% dos participantes declarou ser de Assis, interior de São Paulo; 10% são de São Paulo e as demais, que somam 25% do total, são compostas por Santo Antonio da Platina, Cândido Mota, Dois Vizinhos e Carapicuíba.

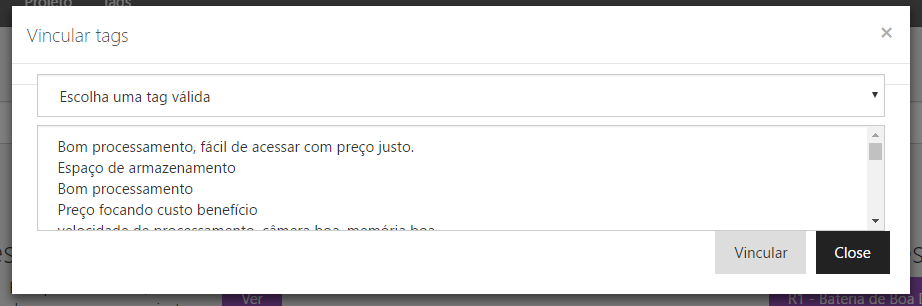
Na análise das respostas de qual seria o telefone ideal, notou-se uma grande disparidade entre o conhecimento dos envolvidos como podemos ver na tabela a seguir:

**Tabela 1 - Distribuição de respostas dos usuários participantes**

|  |
| --- |
| Respostas dos participantes |
| Bateria com alta duração, tela de 4.7, Sistema puro. |
| Espaço de armazenamento |
| Navegação/interface intuitiva |
| Bom processador |
| Boa memória interna |
| Bom processamento, fácil de acessar com preço justo. |
| Velocidade de processamento, câmera boa, memória boa |
| Alta capacidade de armazenamento |
| Sistema Operacional leve e bonito |
| Câmera com boa sensibilidade a luz e qualidade de imagem |
| Bateria de longa duração e recarga rápida |
| Câmera com boa sensibilidade a luz. |
| Bateria de longa duração e recarga rápida. |
| Câmera com boa qualidade de imagem. |
| Bom processamento, boa memória, excelente duração de bateria |
| Dois Chips |
| Câmera Boa |
| Bom processamento |
| Bastante memória |
| Tela grande |
| Impermeável |
| Rápido |
| Internet grátis |
| Resistente para trabalho |
| Muita bateria |
| Bom acabamento |
| Resistente |
| Com uma interface muito personalizável (Ex: Windows Phone) |
| Com processamento muito rápido |
| Com uma câmera de excelente qualidade |
| Com sistema de GPS via satélite (off line como aparelhos de GPS convencionais" |
| Com uma central multimídia intuitiva para música e vídeo |
| Bom processamento |
| Preço focando custo benefício |

Alguns usuários responderam mais de uma resposta em uma linha, o que exigiu uma análise apurada na hora de contabilizar os resultados na ferramenta de auxílio a *crowd*. Abaixo, uma tela de como é feita a vinculação do padrão de resposta, que deve ser criado manualmente, com as respostas dadas pelos usuários.

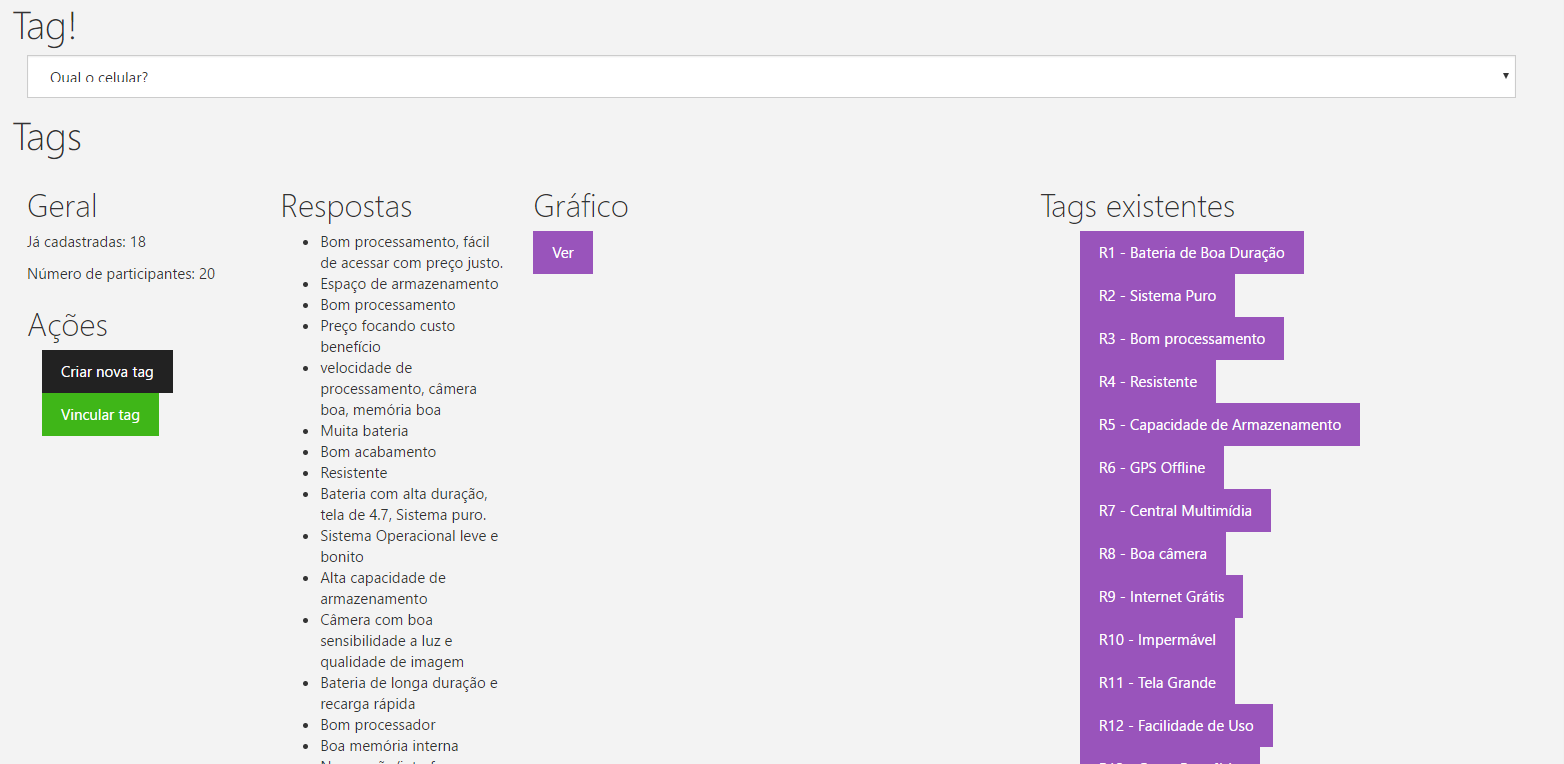
**Figura 7. A vinculação de padrões de resposta**



**Fonte: autoria própria**

Apesar da vinculação manual, a categorização dos padrões da pesquisa realizada demorou menos de 30 minutos para ser totalmente completada. Na tela intitulada *Tags*, a visão das respostas é a seguinte:

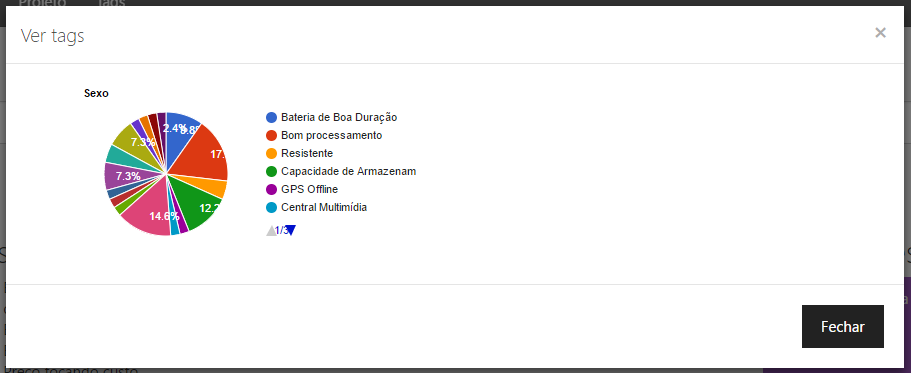
**Figura 8. Interface da tela de tags**



**Fonte: autoria própria**

A análise dos resultados após o processo de aplicação de tags, pode ser melhor analisado por um gráfico em forma de pizza:

**Figura 9 - Tags dispostas em gráficos**



**Fonte: Autoria própria**

Para facilitar a visualização, baixo temos uma tabela com a exibição dos dados do gráfico e com os dados informados pelos usuários já abstraídos e convertidos em *tags* e, portanto, transformados em requisitos.

**Tabela 2. Distribuição das tags com o padrão de resposta dos participantes e o número de ocorrências**

|  |  |  |
| --- | --- | --- |
| Código do Requisito | Requisito | Número de ocorrências |
| R0001 | Bateria de Boa Duração | 4 |
| R0002 | Sistema Puro | 1 |
| R0003 | Bom processamento | 7 |
| R0004 | Resistente | 2 |
| R0005 | Capacidade de Armazenamento | 5 |
| R0006 | GPS Offline | 1 |
| R0007 | Central Multimídia | 1 |
| R0008 | Boa câmera | 6 |
| R0009 | Internet Grátis | 1 |
| R0010 | Impermeável | 1 |
| R0011 | Tela Grande | 1 |
| R0012 | Facilidade de Uso | 3 |
| R0013 | Custo benefício | 2 |
| R0014 | Boa memória RAM | 3 |
| R0015 | Bom acabamento | 1 |
| R0016 | Recarga rápida | 1 |
| R0017 | Interface com boa personalização | 1 |
| R0018 | Dois Chips | 1 |

## **CONCLUSÃO**

No fim deste projeto concluiu-se a importância e o funcionamento do *crowdsourcing* no desenvolvimento de um produto a partir de um grande público, bem como melhor compreensão da linguagem Java.

O processo de análise de padrões, ainda que seja de maneira manual, cadastrando-os e categorizando os mesmos por padrões de resposta acelerou de forma significativa a apuração de resultados e melhorou a escalabilidade do crowdsourcing. Ainda que seja possível aprimorar mais o projeto na questão de gestão de projetos utilizando-se o *crowdsourcing*, podemos dizer que a resolução do problema de *crowdsourcing* e requisitos foi parcialmente resolvida, uma vez que não se conseguiu aqui analisar os padrões de resposta de forma autônoma e nenhum mecanismo de inteligência artificial foi aplicado.

O projeto então atingiu aquilo que propôs de desenvolver uma ferramenta de *crowdsourcing*, assim automatizando o processo, cumprindo com seu propósito de desenvolvimento de uma ferramenta e apresentando aspectos gerais do *crowdsourcing*.

Além disso, também realizou-se pesquisa com os usuários para tentar simular ao máximo possível as condições do trabalho de RIBEIRO (2016) e seguindo as recomendações do trabalho de Ribeiro sobre uma ferramenta de *crowdsourcing*, atendendo assim a proposta do projeto.

## **TRABALHOS FUTUROS**

Tendo como base o processo de crowdsourcing e sua relação com engenharia de requisitos, um dos trabalhos futuros possíveis é a elaboração de módulos para a ferramenta que possam automatizar a ferramenta no aspecto de análise de padrões de resposta, bem como mecanismos mais precisos de acompanhamento das respostas das pesquisas.

Um aspecto importante seria realizar a parametrização de aspectos da criação do *crowd* como por exemplo: tornar ou não determinados campos obrigatórios, capturar temporização das respostas, análise dos dispositivos acessados, que podem aprimorar na análise de requisitos.

Um ponto importante seria a integração da ferramenta de crowdsourcing com alguma ferramenta de projetos, o que faria, em uma totalidade a integração do processo de Crowdsourcing com o *framework* Scrum, tornando assim as *tags* em atividades de requisitos para o projeto a ser gerenciado, uma das recomendações seria o Scrum.

Uma outra abordagem da ferramenta do *crowdsourcing* com a ferramenta, seria a aplicação do processo de gamificação com o *crowdsourcing* que elevaria o interesse dos usuários participantes do *crowdsourcing*, isso aumentaria o interesse dos usuários em responderem as perguntas pelo processo, conforme sugerido por Hossain (2012).

Outra possibilidade é o uso da ferramenta para o desenvolvimento de um produto fazendo uso do crowdsourcing e a ferramenta para levantar requisitos para este projeto, apenas delimitando o escopo inicial.

**REFERÊNCIAS**

O’REILLY, T. **What Is Web 2.0?**. Set 2005.

STATS, I. W. **INTERNET USAGE STATISTICS:** The Internet Big Picture World Internet Users and 2015 Population Stats. Nov 2015.

LEITE, N.; CAIXINHA, H.; RAMOS, F. **Proposta de uma aplicação web para monitoração do impacto de notícias nas redes sociais Facebook e Twitter**. Revista Comunicando, v. 2, 2013.

HOSSAIN, M. **Users’ Motivation to Participate in Online Crowdsourcing Platforms**. p. 21–22, 2012.

HOSSEINI, M. et al. Configuring crowdsourcing for requirements elicitation.  
Proceedings - International Conference on Research Challenges in InformationScience. v. 2015–June, n. June, p. 133–138, 2015.

HOWE, J. **The Rise of Crowdsourcing.** Wired magazine, n. 14, p. 1–5, 2006

TRAVASSOS, G.; GUROV, D.; AMARAL, E. **Introdução à Engenharia de Software  
Experimental**. Relatório Técnico ES59002Abril Programa de Engenharia de  
Sistemas e Computação. COPPEUFRJ, p. 52, 2002

VLAANDEREN, K. et al. The Agile Requirements Refinery: Applying SCRUMPrinciples to Software Product Management. 2010.

L’ERARIO, 2009. ***M3DS:*** um modelo de dinâmica de desenvolvimento distribuído de software***.*** *São Paulo.*

MASSE, M. **REST API Design Rulebook**: Designing Consistent RESTful Web Service Interfaces. 2011. 5 - 6 p. O'Reilly Media, Inc.

CAMPOMORI, C. **REST não é simplesmente retornar JSON**: *indo além com APIs REST. 2017*. Disponível em: <https://www.treinaweb.com.br/blog/rest-nao-e-simplesmente-retornar-json-indo-alem-com-apis-rest/>

CURIOSO, L. **Entendendo o protocolo HTTP. 2017**. Disponível em: <http://blog.leandrocurioso.com/2017/05/entendendo-o-protocolo-http/>

BRIGATTO, G. **Vendas de smartphones voltam a crescer no Brasil em 2017, estima IDC**. 2017. Valor Econômico. Disponível em: <http://www.valor.com.br/empresas/4848934/vendas-de-smartphones-voltam-crescer-no-brasil-em-2017-estima-idc>

**ANEXO A – Direitos autorais - Lei nº 9.610, de 19 de fevereiro de 1998. Disposições preliminares**

|  |  |
| --- | --- |
| Brastra.gif (4376 bytes) | **Presidência da República Casa Civil Subchefia para Assuntos Jurídicos** |

**LEI Nº 9.610, DE 19 DE FEVEREIRO DE 1998.**

|  |  |
| --- | --- |
| Mensagem de veto | Altera, atualiza e consolida a legislação sobre direitos autorais e dá outras providências. |

**O PRESIDENTE DA REPÚBLICA** Faço saber que o Congresso Nacional decreta e eu sanciono a seguinte Lei:

Título I

Disposições Preliminares

        Art. 1º Esta Lei regula os direitos autorais, entendendo-se sob esta denominação os direitos de autor e os que lhes são conexos.

        Art. 2º Os estrangeiros domiciliados no exterior gozarão da proteção assegurada nos acordos, convenções e tratados em vigor no Brasil.

        Parágrafo único. Aplica-se o disposto nesta Lei aos nacionais ou pessoas domiciliadas em país que assegure aos brasileiros ou pessoas domiciliadas no Brasil a reciprocidade na proteção aos direitos autorais ou equivalentes.

        Art. 3º Os direitos autorais reputam-se, para os efeitos legais, bens móveis.

        Art. 4º Interpretam-se restritivamente os negócios jurídicos sobre os direitos autorais.

        Art. 5º Para os efeitos desta Lei, considera-se:

        I - publicação - o oferecimento de obra literária, artística ou científica ao conhecimento do público, com o consentimento do autor, ou de qualquer outro titular de direito de autor, por qualquer forma ou processo;

        II - transmissão ou emissão - a difusão de sons ou de sons e imagens, por meio de ondas radioelétricas; sinais de satélite; fio, cabo ou outro condutor; meios óticos ou qualquer outro processo eletromagnético;

        III - retransmissão - a emissão simultânea da transmissão de uma empresa por outra;

        IV - distribuição - a colocação à disposição do público do original ou cópia de obras literárias, artísticas ou científicas, interpretações ou execuções fixadas e fonogramas, mediante a venda, locação ou qualquer outra forma de transferência de propriedade ou posse;

        V - comunicação ao público - ato mediante o qual a obra é colocada ao alcance do público, por qualquer meio ou procedimento e que não consista na distribuição de exemplares;

        VI - reprodução - a cópia de um ou vários exemplares de uma obra literária, artística ou científica ou de um fonograma, de qualquer forma tangível, incluindo qualquer armazenamento permanente ou temporário por meios eletrônicos ou qualquer outro meio de fixação que venha a ser desenvolvido;

        VII - contrafação - a reprodução não autorizada;

        VIII - obra:

        a) em co-autoria - quando é criada em comum, por dois ou mais autores;

        b) anônima - quando não se indica o nome do autor, por sua vontade ou por ser desconhecido;

        c) pseudônima - quando o autor se oculta sob nome suposto;

        d) inédita - a que não haja sido objeto de publicação;

        e) póstuma - a que se publique após a morte do autor;

        f) originária - a criação primígena;

        g) derivada - a que, constituindo criação intelectual nova, resulta da transformação de obra originária;

        h) coletiva - a criada por iniciativa, organização e responsabilidade de uma pessoa física ou jurídica, que a publica sob seu nome ou marca e que é constituída pela participação de diferentes autores, cujas contribuições se fundem numa criação autônoma;

        i) audiovisual - a que resulta da fixação de imagens com ou sem som, que tenha a finalidade de criar, por meio de sua reprodução, a impressão de movimento, independentemente dos processos de sua captação, do suporte usado inicial ou posteriormente para fixá-lo, bem como dos meios utilizados para sua veiculação;

        IX - fonograma - toda fixação de sons de uma execução ou interpretação ou de outros sons, ou de uma representação de sons que não seja uma fixação incluída em uma obra audiovisual;

        X - editor - a pessoa física ou jurídica à qual se atribui o direito exclusivo de reprodução da obra e o dever de divulgá-la, nos limites previstos no contrato de edição;

        XI - produtor - a pessoa física ou jurídica que toma a iniciativa e tem a responsabilidade econômica da primeira fixação do fonograma ou da obra audiovisual, qualquer que seja a natureza do suporte utilizado*;*

        XII - radiodifusão - a transmissão sem fio, inclusive por satélites, de sons ou imagens e sons ou das representações desses, para recepção ao público e a transmissão de sinais codificados, quando os meios de decodificação sejam oferecidos ao público pelo organismo de radiodifusão ou com seu consentimento;

        XIII - artistas intérpretes ou executantes - todos os atores, cantores, músicos, bailarinos ou outras pessoas que representem um papel, cantem, recitem, declamem, interpretem ou executem em qualquer forma obras literárias ou artísticas ou expressões do folclore.

        Art. 6º Não serão de domínio da União, dos Estados, do Distrito Federal ou dos Municípios as obras por eles simplesmente subvencionadas.