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

**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 foi relacionado por RIBEIRO (2016) que propôs um modelo chamado de r4c. Este projeto todavia este modelo proposto não possui uma ferramenta para realizar todas ações realizadas, sendo este o propósito deste trabalho, tendo como base o processo r4c, desenvolver uma ferramenta para o processo de crowdsourcing, assim auxiliando no processo de levantamento de requisitos.

**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 proposal approach concepts and explain a global view of Digital Marketing and it’ evolution in technology context. Furthermore, is approached the role of social media in presente and how they can help in the process of capture of new clientes. The growth of social medias makes the appearance of digital networks to accord a market share of enterprises and organizations that seek new approaches of interatiions. However, like many others sectors it’s necessary a adequated management of requirements and of process, something that proposes seeks to point: a solution through of a tool of process management oriented to Digital Marketing.

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

**LISTA DE IMAGENS**

**LISTA DE SIGLAS**

|  |  |
| --- | --- |
| API | Aplication Program Interface |
| JPA | Java Persistence Application |
| UML | Unified Modeling Language |
| SRS | Software Requirements Specifications |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**Sumário**

[**1.1.** **INTRODUÇÃO** 8](#_Toc483294509)

[**1.2.** **PROBLEMA** 9](#_Toc483294510)

[**1.3.** **JUSTIFICATIVA** 9](#_Toc483294511)

[**1.4.** **OBJETIVOS** 10](#_Toc483294512)

[**1.4.1.** **OBJETIVO GERAL** 10](#_Toc483294513)

[**1.5.2.** **OBJETIVOS ESPECÍFICOS** 10](#_Toc483294514)

[**1.5.3.** **ESTRUTURA DO TRABALHO** 10](#_Toc483294515)

[**2.** **FUNDAMENTAÇÃO TEÓRICA** 11](#_Toc483294516)

[**2.1.** **CROWDSOURCING** 11](#_Toc483294517)

[**2.2.** **ENGENHARIA DE REQUISITOS** 13](#_Toc483294518)

[**2.3.** **REQUISITOS EM CROWDSOURCING** 14](#_Toc483294519)

[**2.4.** **TRABALHOS RELACIONADOS** 15](#_Toc483294520)

[**3.** **MÉTODOS E PROCEDIMENTOS** 18](#_Toc483294521)

[**3.1.** **SCRUM SOLO** 18](#_Toc483294522)

[**3.2.** **TESTE EXPERIMENTAL** 19](#_Toc483294523)

[**3.3.** **ARQUITETURA CANDIDATA** 19](#_Toc483294524)

[**3.3.1.** **MODEL VIEW CONTROLLER** 19](#_Toc483294528)

[**3.3.2.** **REST API** 20](#_Toc483294529)

[**4.** **DESENVOLVIMENTO DO APP** 21](#_Toc483294531)

[**5.** **TESTE EXPERIMENTAL** 24](#_Toc483294532)

## **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;
4. Observar as interações com o Scrum; e
5. A adoção de um processo ágil (Scrum Solo) para o desenvolvimento desta proposta.

## **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*.

A 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 a 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 verifica-se 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**

## **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**

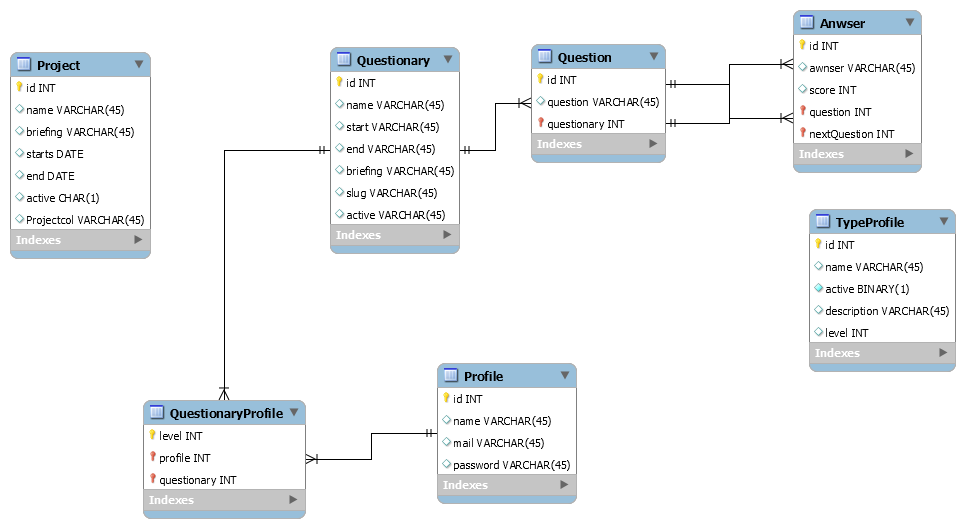
O estilo arquitetural REST API é uma arquitetura baseada em serviços, onde aplicações diferentes podem se comunicar a partir de um protocolo em comum, como o JSON onde

## **DESENVOLVIMENTO DO APP**

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

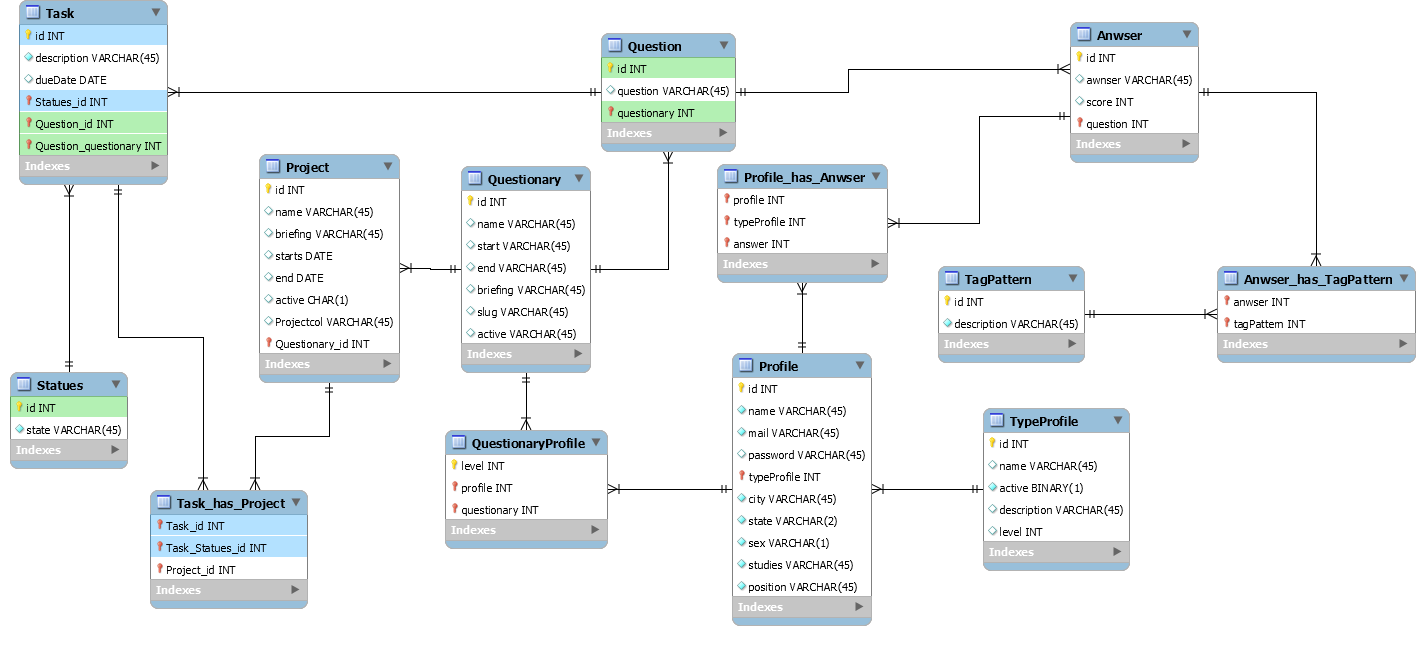
A aplicação central, onde existe a versão web e seus webserivces faz uso da tecnologia e linguagem Java, tecnologia Rest API, JPA; para controle transacional e de mapeamento de dados da aplicação e o banco de dados Postgres.

O desenvolvimento se deu inicialmente 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, 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.



**Figura 1. Diagrama do release 0.1 do banco de dados**

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



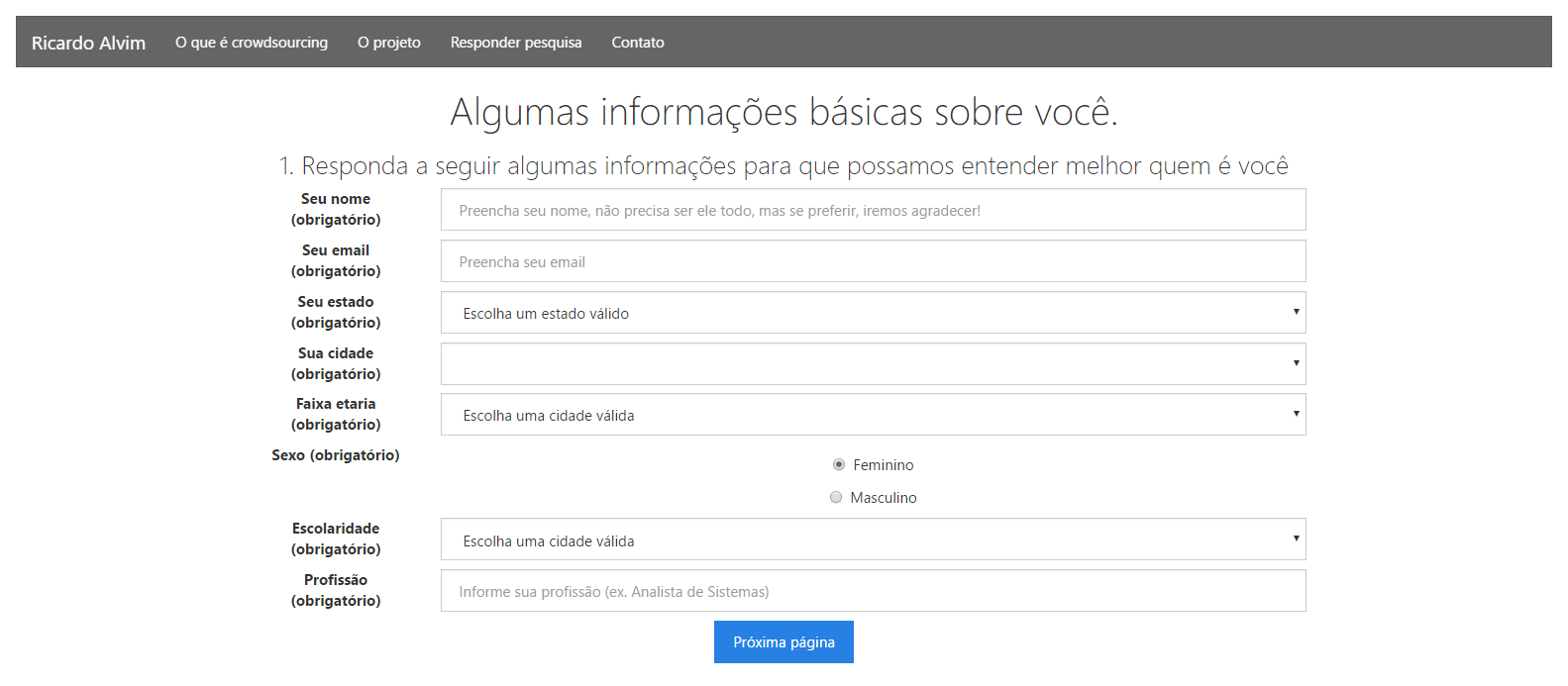
**Figura 2. Segundo release do banco de dados. A estrutura do mesmo ficou melhor definida.**

A estrutura do banco de dados a partir do release 2 ficou melhor definida e seguiu sem maiores alterações. Após esta versão, deu-se o desenvolvimento da aplicação principal, fazendo uso da ferramenta NetBeans 8.2.

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* para compactação dos scripts e assim tornar o carregamento das páginas mais rápido.



**Figura 3. Tela do processo de crowdsourcing do primeiro release/milestone do projeto.**

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* para compactação dos scripts e assim tornar o carregamento das páginas mais rápido.

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 4 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.

## **TESTE EXPERIMENTAL**

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

O teste experimental foi realizado de forma livre a quem tivesse interesse em participar do processo de crowdsourcing, onde foram disponibilizadas a versão web e a versão móvel da aplicação.

O teste consiste em preencher informações simples da pessoa que está participando do *crowd*, como nome, idade, localidade, escolaridade e área de atuação 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, dando assim total liberdade ao usuário em responde-las como preferir. A única limitação no caso era do limite de caracteres.

O link de acesso ao crowdsourcing foi liberado no Facebook para o grupo de alunos da Universidade Tecnológica Federal do Paraná – Campus Cornélio Procópio e também no perfil do autor deste projeto, Ricardo Carreiro Alvim. Onde os usuários iriam responder a mesma pergunta realizada por (RIBEIRO, 2016) que era “Como seria um celular ideal para você?”

Ao todo participaram xxx usuários no processo de *crowd*, yy respostas totalizando uma média de zz respostas por usuários.

Destes usuários, abaixo está a distribuição por faixa etária.

A escolaridade, pode ser vista a seguir

Destes usuários x % responderam ter familiaridade com o tema.

A distribuição das respostas pode ser vista a seguir

**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.*