

Freelagile - Uma Ferramenta Para o Gerenciamento Ágil de Projetos de Software Para Desenvolvedores Freelancer

Saulo Mário de Moura
Instituto Federal de Educação, Ciência e
Tecnologia da Bahia
Rua Emídio dos Santos, s/n – Barbalho
saalomario@ifba.edu.br

Simone da Silva Amorim
Instituto Federal de Educação, Ciência e
Tecnologia da Bahia
Rua Emídio dos Santos, s/n – Barbalho
simone.amorim@ifba.edu.br

RESUMO

Devido à necessidade do corte de gastos frente a um mercado cada vez mais competitivo, pequenas e médias empresas buscam cada vez mais suprir as suas carências de mão de obra a partir da contratação de serviços *freelancer*. Por outro lado, com a crescente demanda por este tipo de serviço em conjunto com a necessidade da complementação de renda, o número de profissionais que praticam este tipo de trabalho é cada vez maior. Neste contexto, uma das maiores dificuldades enfrentadas na execução destes serviços fica por parte do controle e acompanhamento das atividades realizadas por estes profissionais, já que uma grande parte não trabalha alocado no cliente. Este trabalho apresenta o *Freelagile*, uma ferramenta concebida a partir do levantamento das maiores necessidades dos contratantes e dos fornecedores deste tipo de serviço, que serve de suporte tanto para o cliente quanto para o desenvolvedor *freelancer*, visando a melhoria em todo o processo de desenvolvimento, desde as definições dos requisitos até a entrega do produto.

Palavras-chave

Freelance; Agile; Gerenciamento Ágil de Projetos, Ferramenta CASE

1. INTRODUÇÃO

O *Freelancer* “é um profissional que executa a sua atividade de forma independente, prestando serviços a vários empregadores com os quais não tem vínculo de trabalho permanente”^[2]. Essa ausência de vínculo trabalhista torna este tipo de profissional bastante versátil, podendo prestar serviços para diversos clientes ao mesmo tempo.

O trabalho informal realizado por *freelancers* vem ganhando a cada ano mais espaço no mercado mundial. Diversas pessoas já vem trabalhando na criação de empresas para divulgação e contratação deste tipo de serviço. As plataformas online Workana¹, Freelancer² e Upwork³ são as principais empresas no mercado a realizar este tipo de intermediação *freelancer* x cliente.

O objetivo deste trabalho é apresentar o *Freelagile*, uma ferramenta para gerenciamento de projetos de software desenvolvidos por profissionais *freelancer*. Esta ferramenta foi desenvolvida com base em um levantamento prévio sobre

quais são as maiores dificuldades dos contratantes e dos profissionais que realizam este tipo de trabalho.

A função principal do *Freelagile* é aprimorar a organização e o acompanhamento das tarefas executadas nos projetos realizados por desenvolvedores *freelancer*. A ferramenta visa facilitar o acompanhamento das demandas por parte dos contratantes e o controle das atividades realizadas pelos desenvolvedores, auxiliando também em problemas comuns na engenharia de software, como o levantamento de prazo e custo de um projeto.

Além da introdução, este documento possui a seção 2 que apresenta qual a motivação e a justificativa para o desenvolvimento deste trabalho. A seção 3, que apresenta os trabalhos relacionados com o objetivo deste trabalho. A seção 4, que aborda funcionalidades de ferramentas relacionadas com o objeto deste artigo. A seção 5, que apresenta os principais conceitos sobre os assuntos relacionados ao *Freelagile*, fazendo uma conexão entre estes assuntos e de que forma eles são abordados no desenvolvimento deste trabalho. A seção 6, que apresenta informações técnicas sobre a ferramenta, de que forma foram levantados os requisitos para a implementação e as tecnologias utilizadas para o seu desenvolvimento. A seção 7 que mostra como foi feita a validação da solução, desde a preparação até os resultados obtidos. A seção 8 que apresenta a conclusão deste trabalho e a seção 9 que apresenta algumas funcionalidades que podem ser implementadas para deixar a solução ainda mais completa.

2. MOTIVAÇÃO E JUSTIFICATIVA

Segundo o relatório do trabalho e empreendedorismo no Brasil [1] publicado em 2017 pela Workana, plataforma voltada para profissionais *freelancers* com atuação em toda a América Latina, onde foram ouvidas 5.151 pessoas que tiveram algum tipo de trabalho remunerado em 2016, o número de profissionais *freelancer* mais que dobrou nos últimos 4 anos. Este número atingiu um crescimento de 181% no ano de 2016 e a perspectiva é continuar crescendo. Ainda de acordo com o relatório, mais de 40% dos profissionais no mercado são *freelancers* em tempo integral ou *part-time*⁴ e a área mais popular é a de Tecnologia da Informação (TI), atingindo 29,4% dessa força de trabalho.

Este relatório ainda destaca que 78,5% destes profissionais não trabalham junto ao cliente, assim o contratante não consegue realizar um acompanhamento de perto do de-

¹<https://www.workana.com>

²<https://www.freelancer.com>

³<https://www.upwork.com>

⁴De acordo com o artigo 58-A da Consolidação das Leis do Trabalho (CLT), considera-se trabalho em regime de tempo parcial aquele cuja duração não exceda a vinte e cinco horas semanais.

envolvimento dessas atividades, acabando assim por se perder entre as diversas atividades a serem feitas em diferentes projetos. Diante deste cenário onde, por um lado cresce o número de empresas que visam o corte de gastos através da contratação deste tipo de serviço e, por outro, cresce o número de profissionais que optam por realizar trabalhos de forma autônoma, se torna cada vez mais complicado gerenciar estas demandas.

Embora sejam serviços bastante procurados, ainda existe um grande problema na execução destes projetos. Assim como as grandes empresas de desenvolvimento de software, os desenvolvedores *freelancer* devem realizar um gerenciamento dos seus projetos. Definir custo, prazo, planejar entregas, organizar e dividir atividades, analisar demandas e riscos e programar são algumas das atividades que devem ser realizadas por esses profissionais. Em grandes ambientes corporativos, cada uma dessas tarefas podem ser alocadas para diferentes pessoas dos mais variados cargos e conhecimentos técnicos, mas na realidade do *freelancer* todas essas atividades competem a apenas uma pessoa e quanto maior for o auxílio computacional para a realização dessas tarefas, maior pode ser a qualidade do produto entregue. No levantamento de dados sobre o perfil do *freelancer* no Brasil realizado em 2017 pelo Aparelho Elétrico, blog que visa debater e divulgar boas práticas relacionadas a trabalho *freelancer* e empreendedorismo criativo [4], uma das maiores dificuldades, existente em 22.3% dos entrevistados, é ter disciplina, ou seja, manter a ordem e a organização ao gerenciar as suas demandas. Em O Incrível Manual do Freelancer Moderno [25], *eBook*⁵ publicado neste mesmo blog, Henrique Pochmann destaca o uso de softwares de gestão de projetos como uma das ferramentas indispensáveis para um *freelancer*, afim de manter todas as informações bem organizadas e aumentar a produtividade do profissional.

3. TRABALHOS RELACIONADOS

Com o crescente número de trabalhadores autônomos e da contratação destes serviços nos últimos anos, os olhos da academia vem se voltando cada vez mais para estes profissionais. Trabalhos que incluem aspectos comportamentais dos profissionais *freelancer* e dos seus contratantes, a influência das tecnologias no dia-a-dia destes profissionais e abordagens de metodologias e modelos de estimativas para melhorar o gerenciamento dos projetos são alguns dos temas abordados nestes artigos.

Nesta seção são apresentados 3 artigos que possuem como tema central ou secundário o trabalho *freelancer*, levantando seus principais pontos e de que forma contribuem para este trabalho.

3.1 Balancing Work, Life and Other Concerns: A Study of Mobile Technology Use by Australian Freelancers

O estudo realizado por Sadler et al.[28], apresenta resultados iniciais do uso de tecnologias móveis por *freelancers* do domínio do cinema e televisão. Os resultados deste estudo demonstram que os telefones celulares foram utilizados principalmente para gerenciar atividades não relacionadas ao trabalho destes profissionais. O artigo revela, em particular, as maneiras pelas quais os *freelancers* gerenciam a confusão de contextos facilitados pelos aparelhos móveis.

⁵Livro Digital

O artigo aborda mais os aspectos de IHC (Interação Humano-Computador)⁶ discutindo formas pelas quais os *freelancers* gerenciam a confusão de contextos que são facilitados por dispositivos móveis e interação remota, revelando que os telefones celulares permitiram que as pessoas interagissem com outras pessoas para coordenar, buscar informações e promover a conscientização, independentemente do contexto da atividade realizada.

Essa facilidade na forma de interagir com outras pessoas a partir de tecnologias móveis e a abordagem da IHC inserida no contexto do trabalho *freelancer*, contribuíram substancialmente para o desenvolvimento deste trabalho, explanando a necessidade da interação destes profissionais com seus clientes a partir do uso de tecnologias móveis, verificando a utilidade da criação de uma ferramenta web, acessível a partir de qualquer dispositivo com acesso à internet.

3.2 Hiring Behavior Models for Online Labor Markets

O estudo realizado por Kokkodis et al. [19] modela o comportamento de contratação dos empregadores, abordando quais as características observadas e os critérios estabelecidos por estes empregadores para a tomada de decisão sobre a contratação de *freelancers* no mercado de trabalho *on-line*. Ele explicita também o desafio da identificação e contratação do melhor candidato possível, devido a heterogeneidade que aparece nas características observadas.

O objeto do estudo foram os empregadores de serviços no mercado *on-line* e, a partir da premissa de que esses empregadores tomam decisões racionais e contratam o melhor candidato possível, foram propostos uma série de modelos probabilísticos que estimam a probabilidade de contratação de cada candidato. Estes modelos foram aplicados em mais de 600.000 aplicativos de trabalho obtidos pela plataforma oDesk⁷. Nesta análise foi observado que os atributos que estão fortemente correlacionados com a probabilidade de contratação são se o *freelancer* e o empregador já trabalharam juntos, as informações disponíveis sobre o perfil do *freelancer*, o país em que ambos residem e as habilidades do *freelancer*.

Kokkodis explica que, para avaliar um candidato, o empregador deve avaliar duas características:

- As características observadas, que geralmente incluem uma lista de habilidades, o histórico educacional e de trabalho e as certificações que o candidato possui;
- As características latentes, que incluem a qualidade do trabalho do *freelancer*, medida a partir de fatores como a média dos *feedbacks* apresentados, o total de horas trabalhadas na plataforma oDesk e o número de trabalhos que o *freelancer* completou, além das verdadeiras habilidades existentes dentre as suas qualificações informadas.

⁶Interação Humano-Computador é a área que se dedica a estudar os fenômenos de comunicação entre pessoas e sistemas computacionais que está na interseção das ciências da computação e informação e ciências sociais e comportamentais e envolve todos os aspectos relacionados com a interação entre usuários e sistemas[5].

⁷Atualmente *Upwork*, a *Elance-oDesk* mudou de nome em 2015, após apresentar uma nova plataforma voltada para *freelancers*, buscando empurrar a receita anual que antes era de \$1 bilhão para \$10 bilhões em 6 anos.[3]

Estas características latentes e a grande diversidade encontrada nas características observadas tornam a prospecção destes profissionais uma tarefa bastante difícil, visto que as decisões são baseadas em expectativas e essas expectativas vem com muitas incertezas, já que, tanto para o *freelancer* quanto para o empregador, é praticamente impossível conhecer antecipadamente o resultado da sua colaboração.

Com base neste estudo, verificando a dificuldade encontrada na seleção destes profissionais e as características observadas pelos empregadores, a ferramenta *Freelagile* propõe a implementação de um perfil público dos usuários, constando os dados sobre características observadas e latentes, informando habilidades, formação, certificações e trabalhos já realizados através da ferramenta, com o nível de qualidade reportado pelo empregador (contratante).

3.3 Predicting budget for Crowdsourced and Freelance Software development Projects

O artigo desenvolvido por Kumar Abhinav e Alpana Dubey [9], aborda o problema de prever efetivamente o orçamento para projetos de desenvolvimento de software *freelancer* e *crowdsourcing*⁸. O trabalho coloca esta estimativa de orçamento como um grande desafio que os contratantes desta modalidade de serviço enfrentam, sendo que esta estimativa é uma parte importante para qualquer plano de projeto, independente de serem realizados por trabalhadores internos ou *freelancers*.

Embora seja uma etapa bastante importante, nenhuma das plataformas levantadas pelo estudo, nem as estudadas para o desenvolvimento deste trabalho, possuem mecanismo para sugerir o orçamento certo para um trabalho, fazendo com que os clientes estimem o orçamento usando regras com base na percepção, experiência ou intuição subjetiva, resultando em preços baixos ou excessivos [27] [14].

Os autores utilizaram como base para o estudo uma análise empírica a partir de 34 mil chamadas de projetos de desenvolvimento de software na *Upwork* e como resultado obtiveram os seguintes achados:

1. O orçamento não afeta o número de candidatos, a classificação média dos candidatos e também não tem influência no resultado do trabalho.
2. O orçamento difere significativamente para diferentes categorias de trabalho.
3. Os modelos preditivos⁹ podem prever o orçamento relativamente bem em comparação com a previsão de linha de base¹⁰.
4. Fatores como habilidades de trabalho, taxa de contratação, contratos passados, horas totais, tipo de engajamento e tamanho do requisito influenciam diretamente o orçamento do trabalho .

⁸*Crowdsourcing* é o ato de fazer um trabalho tradicionalmente executado por um agente designado (geralmente um empregado) e terceirizá-lo para um grupo indefinido, em geral grande, sob a forma de uma chamada aberta.[18]

⁹Funções matemáticas capazes de aprender o mapeamento entre um conjunto de variáveis de entrada de dados, geralmente agrupadas em um registro, e uma variável de resposta ou de destino.[10]

¹⁰Versão aprovada de um produto de trabalho que só pode ser alterada através de procedimentos formais de controle de mudança e é usada como uma base de comparação.[8]

Como frisa o trabalho, é reconhecidamente essencial a definição de um orçamento por parte dos contratantes de serviços *freelancer*, assim como de qualquer outro tipo serviço. Porém, o foco deste trabalho é nos profissionais que ofertam estes serviços. Se há problemas na definição do orçamento por partes de empresas, pior acontece com profissionais autônomos que não tem nenhum tipo de suporte para a definição desses valores. A Workana desenvolveu um aplicativo online¹¹ onde ajuda os profissionais *freelancer* a calcularem o valor do seu trabalho. Segundo a plataforma “É comum que os *freelancers* cobrem menos do que deveriam pelo seu trabalho. Isso ocorre porque não há muitos dados que sejam públicos sobre o quanto eles devem cobrar. Além do mais, eles também têm menos proteção que os trabalhadores de tempo integral.”[32]. O *Freelagile*, produto deste trabalho, busca ajudar esses desenvolvedores na definição do preço final dos trabalhos realizados, a partir da definição do valor da sua hora de trabalho e das tarefas a serem executadas no projeto.

4. FERRAMENTAS RELACIONADAS

Uma das bases para o levantamento da funcionalidade do produto deste trabalho foi a análise de ferramentas CASE utilizadas para o gerenciamento de projetos. Para avaliação neste trabalho foram analisadas as ferramentas Agilefant¹², Phabricator¹³, Taiga¹⁴ e Trello¹⁵. Estas ferramentas foram escolhidas devido a proximidade de características com o *Freelagile* e o fato de já estarem estabelecidas no mercado de gerenciamento de software. As ferramentas Jira e Redmine também estariam neste comparativo, porém foram descartadas por serem reconhecidamente ferramentas utilizadas por grandes times, o que foge do escopo deste trabalho.

4.1 Agilefant

A Agilefant é uma ferramenta de gerenciamento ágil de projetos baseado no *framework* Scrum. Ela permite a criação de projetos *scrum-like* com a configuração de *sprints* com *product backlog*, *sprint-backlog*, *user stories* com atribuição de *story points* e gerenciamento de papéis. Interessante para empresas que trabalham com times Scrum, mas pouco efetivo para profissionais autônomos. Sua usabilidade pouco intuitiva não permite uma boa experiência de uso, principalmente para usuários que não são acostumados a usar este tipo de ferramenta, caso dos contratantes de serviços de profissionais *freelancer*. A ferramenta não dá suporte a integração com sistemas de controle de versão, uma das funcionalidades tidas como essenciais para os profissionais, de acordo com o questionário aplicado na execução deste trabalho para o levantamento das necessidades de profissionais *freelancer* e contratantes destes serviços, além de também não fazer gerenciamento de custos e de prazo.

4.2 Phabricator

O Phabricator é mais que uma simples ferramenta de gerenciamento de projetos, é uma suite colaborativa de aplicações web utilizadas para, além do gerenciamento de projetos, gerenciar bugs, revisar código, navegar por repositórios,

¹¹<https://www.calculadorafreela.com/pt/>

¹²<https://www.agilefant.com>

¹³<https://www.phacility.com>

¹⁴<https://taiga.io>

¹⁵<https://trello.com>

monitorar alterações, além de conter uma ferramenta wiki integrada. Essa grande reunião de funcionalidades faz com que o Phabricator seja uma das ferramentas mais utilizadas por grandes projetos de software livre para realizar o gerenciamento de tarefas colaborativas. A necessidade de uma usabilidade simples e intuitiva, que possa facilmente ser usada por usuários pertencentes ou não à área de Tecnologia da Informação e a não especificidade, faz com que essa super ferramenta se torne ineficiente para a utilização em projetos *freelancer*. Além disso, a ferramenta não trabalha com gerenciamento de custos e é bem voltada para grandes equipes de desenvolvimento.

4.3 Taiga

Outra ferramenta analisada foi a Taiga. Com funções parecidas, está um passo a frente da Agilefant devido a sua usabilidade. Assim como a concorrente, Taiga é uma plataforma de gerenciamento ágil de projetos para desenvolvedores, projetistas e gerentes de software. Permite a criação de projetos *scrum-like*, assim como o Agilefant e também permite a criação de projetos apenas com *Kanban Board*. Apesar de melhor que o Agilefant, a ferramenta Taiga também não se torna eficiente no escopo abordado neste trabalho, tendo como principais pontos negativos a não integração com sistemas de controle de versão e a falta de suporte a gerenciamento de custos do projeto.

4.4 Trello

A última ferramenta analisada foi o Trello, ferramenta básica para gerenciar tarefas cotidianas. Apesar de não ser uma ferramenta com uma finalidade tão complexa quanto as outras estudadas, se torna importante abordá-la devido ao grande número de usuários. Sendo a ferramenta mais utilizada dentre os entrevistados no questionário aplicado, o Trello é o melhor no que ele se propõe a fazer. Sua interface simples e de fácil manuseio torna capaz o uso por qualquer pessoa, com maior ou menor conhecimento no uso de sistemas de informação. Seu grande número de usuários reflete a necessidade, explicitada no questionário, do uso do *kanban board* pelos profissionais. Apesar de ser um grande apoio no gerenciamento das tarefas, o Trello aborda apenas esse ponto, fazendo com que seus usuários tenha que realizar suas outras atividades (gerência de custo, controle de versão, etc.) em outras ferramentas, descentralizando o gerenciamento do seu projeto.

A tabela 1 mostra as ferramentas analisadas, identificando as funcionalidades já existentes e tidas como essenciais pelos profissionais, junto com essas ferramentas é colocada a solução desenvolvida neste trabalho, apresentando quais dessas características foram implementadas.

5. FUNDAMENTAÇÃO TEÓRICA

5.1 Metodologias de Desenvolvimento de Software

Na Engenharia de Software, uma metodologia ou processo de desenvolvimento consiste em um conjunto de ações e atividades inter-relacionadas realizadas para criar um produto, serviço ou resultado pré-especificado [8]. Para se obter um software é necessário um conjunto mínimo de atividades [26]. Embora existam diversos processos de software, Sommerville define atividades fundamentais comuns entre estes processos [29]:

	Kanban Board	Controle de Versão	Gerencia Prazo	Gerencia Custo
Agilefant	X		X	
Phabricator	X	X		
Taiga	X	X		
Trello	X			
Freelagile	X	X	X	X

Tabela 1: Funcionalidades tidas como necessárias para *freelancers* em cada ferramenta.

Especificação Analisa e define os requisitos do sistema. Verifica quais serão as funcionalidades e as restrições na operação do software a ser desenvolvido.

Projeto e Implementação Define o modo como o software deve ser desenvolvido de modo a atender as suas especificações.

Validação Nesta etapa são realizados os teste para a detecção de erros ou falhas provenientes da implementação. É preciso garantir que o software faz o que o cliente deseja.

Evolução A vontade do cliente muda e o software deve evoluir de forma a garantir os desejos e necessidades do cliente.

TAMANHO	MÉTODO	BEM SUCEDIDO	DIFICULTOSO	FRACASSADO
Projetos de todos os tamanhos	Ágil	39%	52%	9%
	Cascata	11%	60%	29%
Projetos grandes	Ágil	18%	59%	23%
	Cascata	3%	55%	42%
Projetos médios	Ágil	27%	62%	11%
	Cascata	7%	68%	25%
Projetos pequenos	Ágil	58%	38%	4%
	Cascata	44%	45%	11%

Figura 1: Standish Group Chaos Report 2015 [6]

Apesar da existência de diversas metodologias de desenvolvimento, duas em particular merecem uma atenção especial por serem empregadas em boa parte das empresas na atualidade, o modelo clássico ou cascata e as metodologias ágeis. Embora o Gartner Group acredite que 75% das empresas devam atuar de forma bimodal [16], gerenciando dois estilos diferentes em um mesmo ambiente, sendo o modo 1 com um enfoque mais previsível (clássico) e o modo 2 mais exploratório (ágil) [7], afirmar que uma metodologia é melhor que a outra é algo subjetivo. É necessário analisar uma série de fatores que vão desde questões particulares do projeto como complexidade, prazo e tecnologias

Indivíduos e interações mais que processos e ferramentas
Software em funcionamento mais que documentação abrangente
Colaboração com o cliente mais que negociação de contratos
Responder a mudanças mais que seguir um plano

Tabela 2: Valores do Manifesto Ágil

utilizadas até questões particulares da empresa, como ambiente, cultura e porte. Na figura 1 podemos ver os casos de sucesso e de falhas em projetos de acordo com o método de desenvolvimento aplicado e o tamanho da empresa, nos dados apresentados podemos perceber que, apesar das organizações que adotam a metodologia ágil terem obtido mais sucesso e menos falhas nos projetos, esse número varia de acordo com uma única variável aplicada (tamanho da empresa), podendo oscilar muito mais de acordo com o número de variáveis aplicadas.

5.2 Metodologias Ágeis

Em fevereiro de 2001, 17 profissionais entusiastas de um método alternativo aos processos de desenvolvimento clássicos, pesados e documentados, se reuniram para encontrar um pensamento comum para a definição de uma forma mais leve de desenvolver software. Esse encontro deu origem a Aliança Ágil e ao Manifesto Ágil Para Desenvolvimento de Software [11]. O manifesto diz:

“Estamos descobrindo maneiras melhores de desenvolver software, fazendo-o nós mesmos e ajudando outros a fazerem o mesmo”.

Os 4 pilares expostos na tabela 2 compõem a base do manifesto ágil e das metodologias que o seguem. Nota-se a partir destes valores que os processos, documentações, contratos e um plano rígido a ser seguido, bastante presentes nos modelos clássicos, não são esquecidos, porém são postos em segundo plano e a ênfase maior é dada nos indivíduos, na entrega de software funcional, na colaboração com o cliente e nas mudanças existentes durante o ciclo de vida do projeto. Explicando o primeiro dos 4 valores citados, Cockburn afirma em [13] que a qualidade da interação entre os membros do time é importante para a solução de problemas no desenvolvimento do projeto e Larman [20] diz que o uso da comunicação e do feedback é uma diretiva essencial para a prática ágil. Larman ainda diz que as ferramentas utilizadas para auxiliar o desenvolvimento de software devem ser as mais simples possíveis. Baseado nessa maior interação com o cliente vislumbrada por Cockburn e na simplicidade nas ferramentas abordada por Larman a ferramenta apresentada neste trabalho foi concebida.

Além dos 4 valores citados na tabela 2, os membros da Aliança Ágil definiram 12 princípios, com o intuito de ajudar as pessoas a entenderem melhor o desenvolvimento ágil de software e que são seguidos e aplicados por toda metodo-

logia ágil. A tabela 3 mostra os 12 princípios definidos pela aliança ágil.

De acordo com o último *Annual State of Agile Report* realizado pela VersionOne [31], o Scrum é a metodologia ágil mais comum entre as organizações, sendo usada por 68% das empresas em todo o mundo. Deste número, 58% utilizam a forma pura da metodologia e 10% utilizam uma metodologia híbrida com Scrum e XP. No Brasil não é muito diferente, no levantamento realizado por Claudia de O. Melo et al. [15], 73,7% das organizações utilizam o Scrum como principal metodologia, sendo 51,2% usando de forma pura e 22,5% junto com o XP.

5.3 Metodologias de Desenvolvimento para Freelancer

No âmbito dos projetos *freelance*, onde toda a carga de trabalho é centralizada em um único membro, fica difícil definir o uso de uma das metodologias já existentes, visto que o que existe é um universo onde as metodologias são extremamente baseadas em equipes de desenvolvimento, com cada membro desempenhando um papel específico.

Diante do cenário atual, onde o mercado é dominado quase que totalmente pelos métodos ágeis e pelo modelo cascata, O *Standish Group*, em seu já conhecido relatório sobre o estado da indústria de desenvolvimento de software, o *Chaos Report* [30] apresenta dados que podem fazer com que o desenvolvedor *freelancer* siga uma metodologia baseada no modelo ágil de desenvolvimento. Na figura 1, apresentada na seção 5.1, podemos ver os casos de sucesso e de falhas em projetos de acordo com o tamanho e o método de desenvolvimento aplicado. Podemos aí observar que no universo das pequenas e médias empresas, onde os trabalhos desenvolvidos por *freelancers* está inserido, a média percentual de projetos desenvolvidos de forma ágil fracassados é bastante baixa, e a média de projetos bem sucedidos é bem maior quando comparados aos projetos que utilizam o método cascata.

Metodologias que sejam especialmente aplicadas em projetos *freelancer* ainda não são muito difundidas no mercado. Rodrigo Bernabé et al. em seu artigo *Faat (Freelance as a Team)* [12], apresenta uma metodologia projetada para estes profissionais, integrando práticas já existentes às necessidades e possibilidades de um desenvolvedor *freelance*. O processo descrito pelos autores apresenta um conjunto de práticas, retiradas das principais metodologias ágeis e que são rentáveis e fáceis de serem aplicadas por desenvolvedores únicos.

5.3.1 FaaT - Freelance as a Team

O terceiro pilar para a concepção do produto deste trabalho foi a análise de uma metodologia de desenvolvimento baseada nos métodos ágeis e adaptada para o contexto de profissionais *freelance*. O FaaT - *Freelance as a Team*, é uma metodologia apresentada por Rodrigo Borrego Bernabé, Iván Álvarez Navia e Francisco José García-Peñalvo, pesquisadores da Universidade de Salamanca, na 3ª Conferência Internacional sobre Ecosistemas Tecnológicos para Melhorar a Multiculturalidade, ocorrida na cidade de Porto, em Portugal. Os autores abordam o cenário das metodologias ágeis, que foram concebidas para resolver os principais problemas da abordagem tradicional do desenvolvimento de software, mas que são centradas em um time e na comunicação interna para a concepção de um produto com entregas sucessivas. Porém essa abordagem não pode ser aplicada de forma inte-

Número	Princípio
1	Nossa maior prioridade é satisfazer o cliente, através da entrega adiantada e contínua de software de valor.
2	Aceitar mudanças de requisitos, mesmo no fim do desenvolvimento. Processos ágeis se adequam a mudanças, para que o cliente possa tirar vantagens competitivas.
3	Entregar software funcionando com frequência, na escala de semanas até meses, com preferência aos períodos mais curtos.
4	Pessoas relacionadas à negócios e desenvolvedores devem trabalhar em conjunto e diariamente, durante todo o curso do projeto.
5	Construir projetos ao redor de indivíduos motivados. Dando a eles o ambiente e suporte necessário, e confiar que farão seu trabalho.
6	O Método mais eficiente e eficaz de transmitir informações para, e por dentro de um time de desenvolvimento, é através de uma conversa cara a cara.
7	Software funcional é a medida primária de progresso.
8	Processos ágeis promovem um ambiente sustentável. Os patrocinadores, desenvolvedores e usuários, devem ser capazes de manter indefinidamente, passos constantes.
9	Contínua atenção à excelência técnica e bom design, aumenta a agilidade.
10	Simplicidade: a arte de maximizar a quantidade de trabalho que não precisou ser feito.
11	As melhores arquiteturas, requisitos e designs emergem de times auto-organizáveis.
12	Em intervalos regulares, o time reflete em como ficar mais efetivo, então, se ajustam e otimizam seu comportamento de acordo.

Tabela 3: Princípios por trás do Manifesto Ágil

gral em um cenário onde o desenvolvimento é realizado por apenas um desenvolvedor, trabalhando sozinho em pequenos projetos e sem a proximidade do cliente.

Os autores apresentam um conjunto de práticas divididas em três grupos: práticas estratégicas, práticas de fluxo de trabalho e práticas auxiliares.

Práticas estratégicas definem um guia para as melhores tomadas de decisões em cada ponto do desenvolvimento de software. Os autores abordam basicamente três práticas estratégicas: (a) Simplicidade - conduzindo o desenvolvedor a não perder tempo na criação de funcionalidades que não serão utilizadas pelo contratante. Os autores citam aqui do conceito de MVP (*Minimum Viable Product*) que se referem ao conjunto mínimo de recursos que são capazes de fornecer valor por si mesmos; (b) Abraça as mudanças - que fala sobre a aplicação de uma arquitetura tolerante a mudanças, colocando como requisito obrigatório o uso de sistemas de controle de versão; (c) Tomar decisões - que aplica a ideia de que não se deve sofrer antecipadamente com problemas que não se tem certeza que acontecerão. Os autores ainda falam sobre fazer escolhas no momento certo, quando se tem posse das informações, não antes.

O segundo grupo abordado pelos autores é o do fluxo de trabalho. Já apresentado na seção 5.3, esta prática descreve os processos operacionais a serem realizados durante o desenvolvimento do produto e os elementos envolvidos nele. A figura 2 apresenta o fluxo de trabalho apresentado neste grupo.

O terceiro e último grupo descreve práticas que cada desenvolvedor deve aplicar de acordo com suas capacidades, experiência e julgamento pessoal. Essas práticas são: (i) Refatoração - os autores defendem a refatoração como um ótimo mecanismo para garantia de um melhor produto, mas enfatiza que só se deve refatorar quando o tempo despendido permita um desenvolvimento mais rápido no futuro. (ii) Documentação Limitada - fala sobre criar uma documentação básica do projeto desenvolvido, como um diagrama

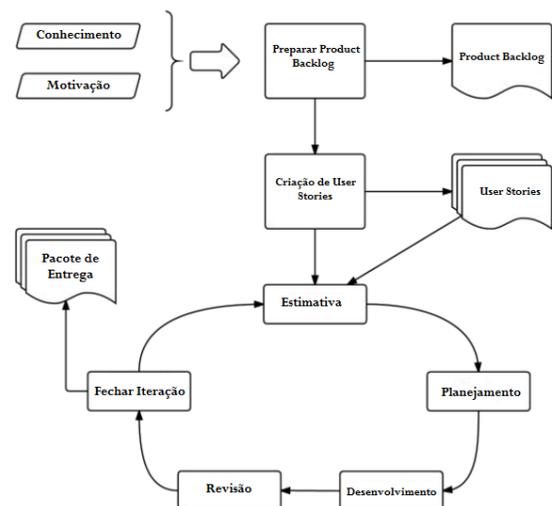


Figura 2: Fluxo de Trabalho FaaT [12].

entidade-relacionamento ou representações equivalentes do modelo de dados. É defendido aqui que a documentação fique o mais próximo possível do código como no cabeçalho de classes e métodos ou num subdiretório da raiz do código fonte do projeto. (iii) Protótipos parciais - defende a criação de protótipos para o esclarecimento de problemas técnicos ou de design. Deve ser desenvolvida com a ideia de posterior eliminação e devem ser incluídas no *backlog* do produto. (iv) Automação - automatizar tudo o que puder o mais rápido possível, dessa forma melhora o reuso em outras tarefas ou em outros projetos, podendo ser útil mesmo após o fim do projeto pelo qual foram criados.

Abaixo são enumerados os passos do fluxo de trabalho proposto por essa metodologia:

1. Preparar *Product Backlog*

Nesta fase é montada a base do projeto, o *product backlog*, que é conceituada por Mike Cohn [21] como uma lista priorizada, contendo descrições curtas de todas as funcionalidades desejadas do produto. A saída esperada é o *product backlog* feito.

2. Criação de *User Stories*

User stories ou histórias de usuário, são simples descrições de uma funcionalidade, contada a partir da perspectiva de uma pessoa que a deseja, geralmente um usuário do sistema [22]. Essas histórias, após serem discutidas e refinadas com o cliente, passam a fazer parte do *product backlog*.

3. Estimativa

Dada pelos autores como um dos desafios do desenvolvimento por um único indivíduo, seja pela falta de proximidade com o cliente ou até mesmo deficiência técnica, a estimativa é necessária para avaliar quanto tempo levará para completar uma tarefa. A possibilidade apresentada é medir o esforço real e comparar ao término de cada iteração para tirar conclusões para futuras estimativas.

4. Planejamento

Para o planejamento os autores determinam 4 pontos principais:

- Agrupar as *user stories* em temas e priorizar as histórias em casa tema seguindo as especificações do cliente.
- Criar um roteiro de entregas baseado na priorização e nas estimativas, com uma carga de trabalho aproximada e orientado a entregas estáveis.
- Trabalhar no marco com o *deadline* mais próximo, revisando o tempo ao final de cada iteração para a medição da velocidade do desenvolvimento
- Adicionar valor ao produto a cada iteração, removendo assim o risco de não entregar nada e corrigir os desvios à medida que ocorrem.

5. Desenvolvimento

Esta fase define o desenvolvimento em si, visa completar as tarefas definidas na iteração na ordem em que foram priorizadas. O artigo indica fortemente o uso de um sistema de controle de versão e a criação de testes que o código deve passar para ser considerado completo.

6. Revisão

Uma vez desenvolvidas as funcionalidades da iteração chega a hora de revisar o que foi feito. Verificar histórias incompletas e comparar as horas estimadas com o tempo real gasto afim de refinar estimativas de outras histórias são os pontos principais nessa fase.

7. Fechamento da Iteração

O fechamento da iteração engloba principalmente as ações de marcação no sistema de controle de versão, entrega e implantação.

A partir da análise da metodologia apresentada, foi possível verificar a necessidade da utilização de um sistema de controle de versão, colocado pelos autores como de uso obrigatório para uma prática ágil.

6. PROJETO

Essa seção tem como finalidade apresentar o sistema *Freelagile*, desenvolvido com o propósito de diminuir os gargalos encontrados no gerenciamento de projetos desenvolvidos por *freelancers*.

6.1 Metodologia

Para chegar ao resultado desejado, foi realizado um estudo sobre os impactos da utilização de metodologias ágeis e tradicionais em pequenas e médias empresas, afim de verificar qual das metodologias se ajustam melhor a esses ambientes e quais os aspectos destas metodologias que conduzem aos melhores resultados.

A partir de uma revisão bibliográfica foram aprofundados os conhecimentos no processo selecionado, sendo realizado um levantamento sobre as suas características, papéis e técnicas existentes e quais as ferramentas mais utilizadas na sua aplicação. A partir daí foi feita uma análise dessas ferramentas com o intuito de extrair as funcionalidades mais comuns entre elas e quais as mais utilizadas.

Também foi aplicado um questionário tendo como público alvo desenvolvedores *freelancer* e contratantes deste tipo de serviço. O intuito deste questionário foi obter como resultado quais as maiores dificuldades enfrentadas por estas pessoas que as técnicas e ferramentas existentes não conseguem resolver.

As características da *Freelagile* foram definidas baseadas no levantamento das necessidades dos profissionais *freelance* e dos contratantes deste tipo de serviço a partir dos pontos citados acima e do artigo *Freelance as a Team* [12]. A análise do artigo foi feita para uma adaptação da abordagem de gerenciamento trazida pelos autores aplicada ao contexto da ferramenta implementada neste trabalho.

6.1.1 Questionário

O questionário aplicado foi disponibilizado para programadores e contratantes de serviços *freelance* de todo o Brasil durante o mês de abril do ano de 2017. Obteve um total de 20 respostas e as perguntas foram separadas de acordo com o perfil do entrevistado, profissional *freelance* ou contratante dessa modalidade de serviço. As perguntas foram elaboradas com o intuito de resgatar informações sobre a atual forma de gerenciamento das demandas de trabalho destes profissionais e dos clientes deles, buscando levantar as dificuldades e as maiores necessidades dos entrevistados no que diz respeito ao gerenciamento, técnicas e decisões tomadas na execução dos projetos. As perguntas do questionário se encontram nas figuras A.12 e A.13 no apêndice deste documento.

Das respostas colhidas do questionário, foram levantadas dificuldades enfrentadas pelos profissionais *freelancers* e contratantes deste tipo de serviço, as técnicas utilizadas por esses profissionais no dia-a-dia de trabalho em projetos realizados de maneira autônoma bem como ferramentas, e as funcionalidades que eles julgam necessário para o bom gerenciamento dos seus projetos. A partir deste levantamento foi possível observar algumas funcionalidades que passaram a fazer parte do escopo da ferramenta *Freelagile*, produto deste trabalho.

Das funcionalidades levantadas a partir da análise das respostas do questionário se destacam a dificuldade na definição de preço e prazo de entrega dos projetos e o cadastro e gerenciamento de tarefas. Esta última é tida como uma das principais funcionalidades em uma ferramenta de apoio ao gerenciamento desses projetos. Isto pode ser visto a partir dos gráficos nas figuras 3 e 4.



Figura 3: Maiores dificuldades no desenvolvimento dos projetos

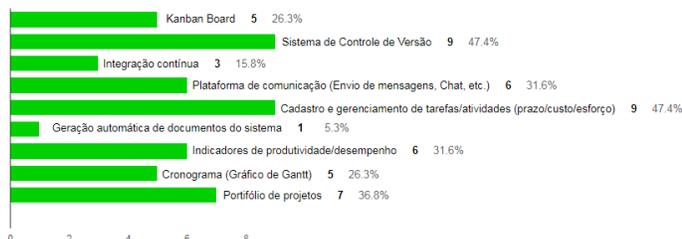


Figura 4: Funcionalidades necessárias em uma ferramenta para gerenciamento de projetos freelance

Outra funcionalidade levantada, também a partir dos dados coletados pelo questionário, foi a integração do sistema com uma ferramenta de controle de versão que, como pode ser visto a partir do gráfico da figura 5, o controle de versão é a ferramenta mais utilizada pelos profissionais *freelance* na gerência dos seus projetos e como pode ser visto na figura 4, é uma das ferramentas consideradas mais usuais em sistemas de gerenciamentos já existentes no mercado.

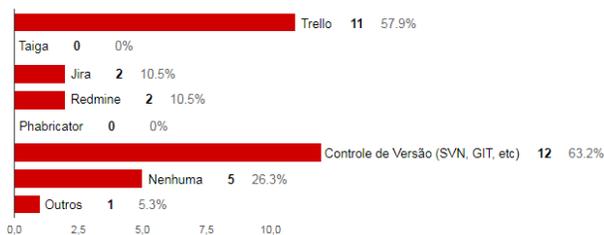


Figura 5: Ferramentas utilizadas para gerenciar o desenvolvimento de projetos freelance

A partir das figuras 4 e 5 também é possível perceber a necessidade de uso de um *Kanban Board*, já que o Trello¹⁶ é uma das ferramentas mais utilizadas, junto com o já citado controle de versão, no gerenciamento do projeto destes profissionais.

¹⁶<https://trello.com>

6.1.2 Ferramentas

O uso de ferramentas CASE para o gerenciamento de projetos realizados por *freelancer* ainda é um problema, devido a falta de possibilidades voltadas para este público. Por outro lado, a cada dia surgem novas e mais robustas implementações para o gerenciamento de projetos com grandes equipes de desenvolvimento.

Para a realização deste trabalho foram avaliadas algumas das ferramentas de gerenciamento de projetos voltadas para grandes times com destaque no mercado. Dentre a grande quantidade de ferramentas existentes foram observadas as que mais se aproximavam do objetivo da *Freelagile*. A partir desta seleção foram observadas as funcionalidades comuns entre elas que poderiam ser úteis para projetos *freelancer* e as funcionalidades que fazem com que estas ferramentas se tornem inviáveis para gerenciar estes projetos, de acordo com as opiniões expostas por profissionais e contratantes deste tipo de serviço a partir do questionário aplicado e pela experiência do autor.

A tabela 4 mostra uma comparação das ferramentas cujas funcionalidades foram analisadas, abordando os seus pontos fortes e fracos, diante do cenário de gerenciamento de projetos *freelancer*.

6.2 A Ferramenta

O *Freelagile* é uma ferramenta web, responsiva, simples e intuitiva, que serve de apoio para o gerenciamento de software desenvolvidos por profissionais *freelance* e para o acompanhamento destes projetos pelos contratantes. Por se tratar de uma aplicação *web* não necessita de instalação, pode ser acessada usando qualquer *web browser* a partir do endereço <http://www.freelagile.com>.

As seções 6.2.1 e 6.2.2 farão uma apresentação técnica com informações sobre as tecnologias utilizadas no desenvolvimento da ferramenta e quais funcionalidades foram implementadas neste projeto. Por fim a seção 6.2.3 apresentará quais os benefícios que o uso da ferramenta traz para os usuários e quais os diferenciais quando comparada às ferramentas CASE de gerenciamento de projetos já existentes no mercado.

6.2.1 Tecnologia

A *Freelagile* foi desenvolvida utilizando duas tecnologias base: No *back-end* foi utilizado o framework PHP Laravel¹⁷ na sua versão 5.4 e no *front-end* foi utilizado o *framework* JavaScript AngularJS na sua versão 1.5.

6.2.1.1 Laravel.

O Laravel é um *framework* PHP para aplicações web que utiliza arquitetura MVC (*Model, View, Controller*). Este *framework* foi criado com o intuito de desenvolver aplicações de forma rápida, segura e com código limpo, para isso ele foi desenvolvido com o padrão PSR-2¹⁸.

Devido a sua curva de aprendizado curta, o Laravel vem se destacando como um dos melhores *frameworks* PHP da atualidade. Além do fácil aprendizado destacam-se também o seu ORM (*Object Relational Mapping*) nativo - o Eloquent, a sua CLI (*Command-Line Interface*) - o Artisan, e o seu

¹⁷<https://laravel.com>

¹⁸ *PHP Standard Recommendation* (PSR) é um conjunto de padrões e boas práticas desenvolvidas pelo PHP-FIG (Grupo com objetivo de falar sobre pontos comuns entre projetos PHP e encontrar maneiras de trabalhar melhor juntos.)

Ferramenta	Funcionalidades inexistentes	Funcionalidades desnecessárias
Taiga	Integração com sistemas de controle de versão, suporte a gerenciamento de custo do projeto, suporte a gerenciamento de tempo do projeto, portfólio de projetos, suporte a <i>feedback</i> .	Atribuição de <i>story points</i> , função “Assigned To” nas tarefas. Ferramenta baseada em equipes de desenvolvimento.
Easy Backlog	Não gerencia projetos, trabalha apenas com <i>backlog</i> e <i>sprints</i> .	Define tempo e custo através da atribuição de pontos de cada tarefa.
Agile Fant	Suporte a <i>kanban board</i> , integração com sistemas de controle de versão, suporte a gerenciamento de custo do projeto, portfólio de projetos, suporte a <i>feedback</i> .	Baseada em equipes de desenvolvimento. Plataforma online com usabilidade ruim.
Phabricator	Suporte a gerenciamento de custo do projeto, suporte a gerenciamento de tempo do projeto, portfólio de projetos, configuração de níveis de acesso, suporte a <i>feedback</i> .	Wiki, Chat. Ferramenta baseada em equipes de desenvolvimento.
Trello	Gerenciamento de custo do projeto, integração com sistemas de controle de versão.	Criação de times, configuração de visibilidade de quadros.

Tabela 4: Quadro comparativo de funcionalidades das ferramentas analisadas

criador nativo de interface gráfica - o *Blade*, que apesar de poderoso, não foi utilizado neste trabalho. Para substituí-lo foi o usado o *framework* JavaScript AngularJS.

6.2.1.2 AngularJS.

Apesar do *framework* Laravel contar com um poderoso criador de interface gráfica, o *Blade*, para este trabalho foi usado para desenvolvimento das *views* o AngularJS.

O AngularJS é um *framework* criado e mantido pela Google e distribuído sob a licença MIT. Seu *bind* bidirecional, denominado *Two-way data binding*, sincroniza os dados de forma que todas as alterações realizadas no *model* são imediatamente refletidas na *view*, e todas as alterações na *view* são igualmente refletidas no *model*.

Além do *bind* bidirecional, o AngularJS provê uma série de diretivas que, a partir de tags HTML¹⁹ personalizadas, facilitam o desenvolvimento, aplicando comportamentos únicos a determinados elementos. O *framework* conta também com um conjunto de componentes UI que implementam o *Google Material Design*, o que torna a aplicação adaptável a diferentes dispositivos e tamanhos de tela.

6.2.1.3 Arquitetura do Sistema.

Para o desenvolvimento do Freelagile foram unidas duas grandes tecnologias já apresentadas acima, o AngularJS, trabalhando do lado cliente e o Laravel pela parte do servidor. Essas tecnologias foram unidas para trabalhar cada um com sua responsabilidade, sendo integradas a partir de uma API RESTful fornecida pelo Laravel e invocada pelos *services* contidos no AngularJS.

O AngularJS adota uma arquitetura denominada por seus desenvolvedores de MVW (*Model-View-Whatever*). A nomenclatura foi dada após várias discussões de qual era o padrão arquitetural adotado pelo *framework* que durante alguns anos trabalhou com um padrão mais perto do MVC (*Model-View-Controller*) e em outro momento algo mais parecido com o MVVM (*Model-View-ViewModel*). Já o Laravel é um *framework* que segue o padrão MVC, com o pequeno aditivo de um arquivo de rotas que mapeia a URL designando ações dos *controllers* para cada uma delas, com

o AngularJS trabalhando como a sua camada de visualização.

A figura 6 mostra em alto nível como funciona o fluxo dessa integração e o modelo de dados da aplicação se encontra no apêndice na figura A.3.

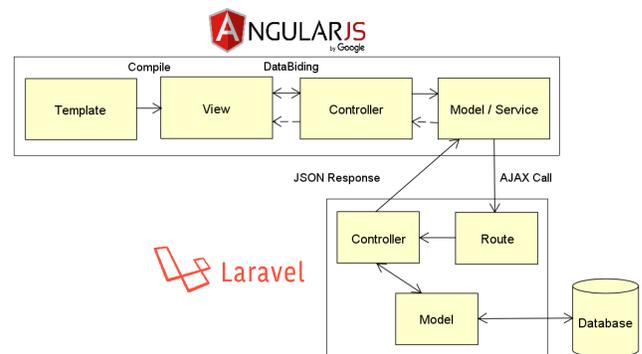


Figura 6: Arquitetura da ferramenta Freelagile com a integração AngularJS x Laravel

6.2.2 Funcionalidades

Uma vez mapeadas as dificuldades dos profissionais *freelancer* e dos contratantes destes serviços baseado nos métodos aplicados, foram levantadas as funcionalidades necessárias, a partir de um levantamento de requisitos, para que a ferramenta Freelagile atendesse a essas necessidades.

De acordo com o *IEEE standard glossary of software engineering terminology* [17], requisitos são definidos da seguinte forma:

1. Uma condição ou capacidade necessária para um usuário para resolver um problema ou alcançar um objetivo.
2. Uma condição ou capacidade que deve ser cumprida ou possuída por um sistema ou componente do sistema para satisfazer um contrato, padrão, especificação ou outro documento formalmente imposto.
3. Uma representação documentada de uma condição ou capacidade como em 1 ou 2.

¹⁹Linguagem de marcação padrão para criação de páginas na Web

Diante disso, foram especificados os requisitos funcionais, que definem recursos específicos que devem ser fornecidos pelo sistema, como definido por Sommerville [29]. A tabela 5 apresenta estes requisitos. Os diagramas de caso de uso de usuário desenvolvedor e usuário contratante são apresentados nas figuras A.1 e A.2 no apêndice.

Identificador	Nome
RF01	O sistema deve permitir gerenciar usuários
RF02	O usuário pode criar projetos
RF03	O usuário criador deve ser definido como desenvolvedor do projeto
RF04	O usuário desenvolvedor pode adicionar informações de valor por hora do projeto
RF05	O usuário desenvolvedor pode definir o usuário contratante do projeto
RF06	O usuário contratante poderá propor o valor da hora que será cobrado no projeto
RF07	O sistema deve apresentar o valor total do projeto com base no valor da hora definido, multiplicado pelo somatório do tempo estimado de todas as tarefas do projeto
RF08	O sistema deve listar o histórico das atividades do projeto
RF09	O usuário desenvolvedor pode adicionar tarefas ao projeto
RF10	O sistema deve permitir a adição de comentários às tarefas
RF11	O usuário desenvolvedor pode adicionar <i>sprints</i> ao projeto
RF12	O usuário desenvolvedor pode adicionar tarefas à <i>sprint</i>
RF13	O sistema deve apresentar o tempo estimado da <i>sprint</i> com base no somatório das tarefas adicionadas
RF14	O sistema deve apresentar o valor estimado da <i>sprint</i> com base no valor da hora definido, multiplicado pelo somatório do tempo estimado das tarefas daquela <i>sprint</i>
RF15	O usuário desenvolvedor pode finalizar <i>sprints</i>
RF16	O usuário desenvolvedor pode adicionar <i>releases</i> aos projetos
RF17	O usuário desenvolvedor pode adicionar <i>sprints</i> à <i>release</i>
RF18	O usuário desenvolvedor pode finalizar <i>releases</i>
RF19	O sistema deve permitir gerenciar visualmente o projeto a partir de um <i>Kanban Board</i>
RF20	O sistema deve permitir gerenciar entregas a partir de um sistema de controle de versão

Tabela 5: Requisitos Funcionais do Freelagile

6.2.3 Benefícios

Diante das poucas alternativas de ferramentas para gerenciamento de ciclo de vida de software voltadas para o público *freelancer* existentes no cenário atual, se tornam claros alguns benefícios que a ferramenta Freelagile pode trazer de

imediatos a esses profissionais e seus contratantes. Abaixo são listados os principais vantagens que podem ser observadas para os usuários que venham a usar a ferramenta para o gerenciamento de seus projetos:

1. Ambiente web com layout simples e responsivo, podendo ser acessado de qualquer lugar em qualquer dispositivo com internet.
2. Sistema de fácil navegação, podendo ser facilmente utilizado por usuários pouco familiarizados com sistemas de gerenciamento de projetos.
3. Funcionalidades mais utilizadas por profissionais *freelancer* centralizadas em uma única ferramenta.
4. Estimativa de custo com base no tempo estimado das tarefas servindo como suporte aos envolvidos na tomada de decisão sobre os custos do projeto.
5. Facilitação na entrega do projeto a partir da configuração de um sistema de controle de versão com possibilidade de download dos arquivos.
6. Informações sobre prazos e valores das atividades e do projeto acessíveis a todos os membros do projeto.
7. Portfólio com informações podendo ser visualizadas mesmo após a finalização do projeto.

7. VALIDAÇÃO

Essa seção descreve o método utilizado para realizar os testes e validações da ferramenta Freelagile. O objetivo desta validação foi verificar a diminuição dos problemas apresentados no gerenciamento de projetos *freelancer* a partir da utilização da ferramenta por usuários assumindo diferentes papéis nos projetos criados.

7.1 Prática

A realização desta atividade foi feita no período entre 12 e 18 de março de 2018. Foram selecionados dois usuários para participar da validação, um contratante e um desenvolvedor e a atividade foi realizada em três etapas: (i) aplicação de questionário para caracterização do usuário, (ii) disponibilização da ferramenta para validação, (iii) aplicação de questionário de *feedback* para colher opiniões sobre a usabilidade e as funcionalidades da ferramenta.

No primeiro momento foi aplicado o questionário para traçar o perfil dos usuários que participaram da validação do sistema e que pode ser visualizado na figura A.10. Da aplicação deste questionário podemos observar que ambos usuários já possuem experiência com TI e, apesar deles já conhecerem algum tipo de metodologia ágil de desenvolvimento de software, apenas o desenvolvedor já trabalhou com projetos com essa metodologia. O usuário que já realizou trabalho como *freelancer* não usou nenhum tipo de metodologia para gerenciar esse desenvolvimento, mas o contratante já utilizou o desenvolvimento baseado no método cascata. Todos os usuários que participaram desta atividade tiveram problemas de entregas no projeto: O contratante enfrentou o não cumprimento de prazos e o não entendimento dos requisitos, culminando na insatisfação com os resultados dos projetos realizados; O desenvolvedor enfrentou dificuldades com a interação direta com o cliente, gerando também o não entendimento integral dos requisitos.

No segundo momento foi realizado o uso da ferramenta pelos usuários. A ferramenta foi disponibilizada a partir do endereço web <http://www.freelagile.com>, podendo ser acessada de qualquer dispositivo com acesso a internet. A figura 7 apresenta a tela principal de um projeto na ferramenta Freelagile. Junto com a ferramenta foi disponibilizado também um vídeo tutorial demonstrando como utilizar as principais funcionalidades do Freelagile, esse vídeo pode ser acessado a partir do endereço <https://youtu.be/CKbcVnUMfCE>.

Por fim, foi disponibilizado um questionário para o recebimento de um *feedback* sobre o uso da ferramenta, que pode ser visualizado na figura A.11. A partir desse questionário foi possível avaliar a opinião dos usuários, verificando a eficácia do Freelagile no que diz respeito às necessidades levantadas por desenvolvedores e contratantes de serviços *freelancer* durante o desenvolvimento deste trabalho. Outras telas do sistema podem ser vistas no apêndice nas figuras A.4, A.5, A.6, A.7, A.8, A.9.

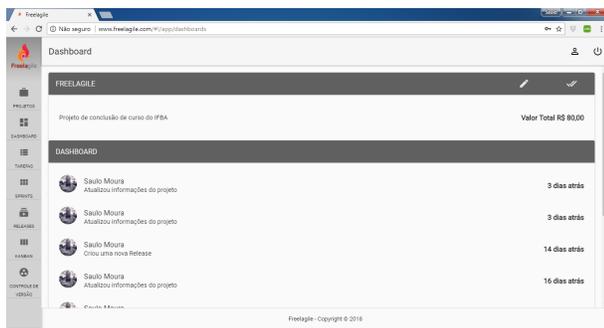


Figura 7: *Dashboard* da ferramenta Freelagile

7.2 Resultados

Aqui serão apresentados os resultados obtidos a partir das respostas do questionário de *feedback* disponibilizado para todos os usuários participantes da validação da ferramenta. Vamos discorrer sobre os principais pontos abordados nesse questionário apresentando os resultados esperados e obtidos a partir das respostas dos usuários.

7.2.1 Facilidade de uso

Um dos principais pontos abordados no desenvolvimento da ferramenta Freelagile foi a facilidade no seu uso, buscando ser um sistema que fosse tranquilamente utilizado por qualquer usuário, com ou sem experiência na área de tecnologia. A usabilidade foi um dos requisitos buscados a todo o tempo durante o desenvolvimento. Através da análise das respostas obtidas, verificamos que a facilidade de uso da ferramenta foi destacada por todos os usuários participantes da atividade.

7.2.2 Emprego da metodologia ágil

Outro ponto de destaque e um dos principais pontos de foco deste trabalho é o emprego da metodologia ágil no gerenciamento dos projetos *freelance*. A ferramenta foi desenvolvida com base na metodologia *Scrum*. As respostas dadas por todos os usuários na validação da ferramenta mostram a boa utilização desta metodologia. Segundo eles, a metodologia ágil foi bem empregada na ferramenta Freelagile.

7.2.3 Facilidades para o contratante

A ferramenta foi desenvolvida com foco em facilitar as duas partes envolvidas no desenvolvimento de projetos *freelancer*, o contratante e o desenvolvedor. Dos contratantes podemos notar que os principais pontos relacionados ao acompanhamento de atividades do desenvolvedor foram atingidas, ficando como ponto positivo a usabilidade do sistema e como ponto negativo a não segmentação do valor por tipo de atividade.

7.2.4 Facilidades para o desenvolvedor

Em relação aos profissionais que desenvolvem projetos *freelancer*, o principal ponto foi atingido, que foi a definição de uma metodologia bem definida no fluxo de desenvolvimento do projeto. Este foi colocado como destaque positivo da solução implementada para este trabalho. Como ponto negativo ficou a impossibilidade de integração com repositórios de controle de versão privados.

8. CONCLUSÃO

Este trabalho apresentou o Freelagile, uma ferramenta *web* que busca a diminuição dos problemas enfrentados por profissionais *freelancer* em desenvolvimento e dos contratantes destes profissionais, a partir de uma ferramenta simples e intuitiva que integra, em uma única plataforma, funcionalidades colocadas como essenciais para o acompanhamento dos projetos desta natureza.

Baseado em três pilares (i) a metodologia *FaaT*[12] (ii) a análise de ferramentas para gerenciamentos de projetos voltadas para equipes de desenvolvimento e (iii) aplicação de um questionário para *freelancers* e contratantes para o levantamento das maiores dificuldades existentes no gerenciamento destes projetos. Este trabalho buscou reunir as maiores dificuldades enfrentadas por estes profissionais junto com as principais ferramentas utilizadas por eles para tentar diminuir estas dificuldades. Nós buscando implementar uma ferramenta que fosse de fácil navegação e, ao mesmo tempo, eficaz na resolução dos problemas enfrentados pelos envolvidos.

Nos testes efetuados podemos validar a solução recebendo um *feedback* positivo em relação às questões chave buscadas pela ferramenta. Questões como a adoção de uma metodologia ágil de desenvolvimento para o acompanhamento dos projetos e a facilidade de navegação pela ferramenta foram plenamente atendidas. A ferramenta pode ser facilmente usada por contratantes que não são da área de Tecnologia da Informação. Além disso, a partir do retorno destes usuários podemos coletar algumas funcionalidades que poderão ser futuramente implementadas para deixar a solução ainda mais completa.

9. TRABALHOS FUTUROS

Em decorrência do curto espaço de tempo ou de fatores que fugiam ao escopo deste trabalho, algumas funcionalidades deixaram de ser implementadas ou foram posteriormente pensadas ou sugeridas a partir do *feedback* apresentado pelos usuários que realizaram a validação da ferramenta, ficando para um futuro desenvolvimento. São elas:

- **Integração com outros serviços de hospedagem de projetos com controle de versão** - Uma das principais funcionalidades presentes na ferramenta, considerada como uma necessidade importante durante o

levantamento dos requisitos do Freelagile, é a visualização do código-fonte do projeto a partir da integração com um sistema de controle de versão. Para a versão desenvolvida neste trabalho foi realizada a integração com a plataforma GitHub²⁰ que gerencia código-fonte com controle de versão usando GIT.

A ideia da abertura de integração para outras plataformas oferecerá ao usuário a possibilidade de escolher o repositório de sua preferência, incluindo a possibilidade de integração com um repositório privado de forma gratuita.

- **Login com credenciais de redes sociais** - Em um país onde, segundo levantamento realizado pelo Centro de Estudos sobre as Tecnologias da Informação e da Comunicação (CETIC)²¹, 78% dos usuários de internet e 51% das empresas com acesso a internet possuem perfil ou fizeram uso de alguma rede social [24, 23]. É necessário atribuir uma parcela de atenção às facilidades possibilitadas com o uso de APIs públicas disponibilizadas por essas ferramentas.

A intenção é que seja incluída na ferramenta a opção de realização de login a partir das credenciais de redes sociais como Facebook, Instagram e Twitter usando autenticação OAuth²² disponibilizada a partir das APIs destas aplicações.

- **Anexar documentos** - Apesar do Manifesto Ágil defender software em funcionamento mais que documentação abrangente[11], a documentação ainda se faz necessária em momentos específicos do projeto. Esta documentação é importante quando feita pelo contratante para dar suporte ao desenvolvedor na tomada de decisões ou tirar suas dúvidas, ou quando feita pelo desenvolvedor, para documentar informações sobre o projeto desenvolvido, como a arquitetura do sistema ou um manual de uso.

Uma proposta interessante seria mapear momentos onde seriam possíveis adicionar estes anexos e permitindo a visualização e/ou edição tanto para o contratante, como para o desenvolvedor ou ambos, de acordo com o qual papel a funcionalidade possa beneficiar.

- **Notificações por e-mail** - No *dashboard* são mostradas todas as ações realizadas no projeto, como mostrado na figura 7, porém, para ter conhecimento destas ações, o usuário deve realizar o login no sistema e selecionar o projeto desejado. Para se manter informado sobre as ações, o usuário pode desejar ser notificado assim que essa ação acontecer.

Uma das alternativas dessas notificação é por e-mail, podendo ser uma possível melhoria na ferramenta a possibilidade do usuário selecionar, em cada projeto, quais ações ele deseja ser notificado.

²⁰<https://github.com>

²¹Órgão responsável pela produção de indicadores e estatísticas sobre a disponibilidade e uso da Internet no Brasil, divulgando análises e informações periódicas sobre o desenvolvimento da rede no país. <http://www.cetic.br>

²²Um protocolo aberto para permitir a autorização segura em um método simples e padronizado de aplicações web, móveis e *desktop*.

- **Notificações via *push notification*** - Assim como foi dito no item anterior, pode ser necessário para o usuário ser notificado sobre ações realizadas no projeto e, além do e-mail já sugerido na mesma seção, existe a alternativa das notificações serem feitas a partir de *push notifications*.

Um *push service* é um serviço que permite que os servidores de aplicativos de terceiros enviem dados ativamente para seus aplicativos instalados, mesmo quando o aplicativo instalado não está sendo executado[33]. No caso do Freelagile, por ser uma ferramenta web, essa notificação seria por uma *web push notification*, que é a mesma coisa do *push notification* tradicional, porém feito a partir de um *web browser*.

- **Diferentes formas de precificação** - Uma das facilidades oferecidas pela ferramenta Freelagile é a possibilidade de definição de preço a partir do valor de hora definido para o projeto. Porém sabemos que pessoas diferentes podem usar formas diferentes para precificar um projeto, utilizando estimativas já conhecidas ou outras formas particulares de realizarem esta precificação. Assim, uma possível melhoria seria implementar outras possibilidades de estimativas de custo para um projeto. O usuário poderia decidir qual estimativa usar em cada novo projeto criado.

10. REFERÊNCIAS

- [1] 2017: Relatório do trabalho e empreendedorismo no Brasil. <https://www.workana.com/blog/pt/freelanceando-pt/trabalho-e-empreendedorismo-brasil/>. Acessado em: 2017-10-31.
- [2] Dicionário priberam da língua portuguesa. <https://www.priberam.pt/dlpo/freelance>. Acessado em: 2017-09-02.
- [3] Elance-oDesk Becomes 'Upwork' In Push To Build \$10B In Freelancer Revenues. <https://www.forbes.com/sites/elainepofeldt/2015/05/05/elance-odesk-becomes-upwork-today-odesk-brand-gets-phased-out/#51d9a2b151f5>. Acessado em: 2017-12-04.
- [4] Infográfico: dados sobre o perfil do freelancer no Brasil 2017. <http://aparelhoeltrico.com/geral/infografico-pesquisa-sobre-o-perfil-do-freelancer-no-brasil-2017/>. Acessado em: 2017-10-31.
- [5] Interação Humano-Computador. <http://www.sbc.org.br/14-comissoes/390-interacao-humano-computador>. Acessado em: 2017-11-07.
- [6] Standish Group 2015 Chaos Report - Q&A with Jennifer Lynch. <https://www.infoq.com/articles/standish-chaos-2015>. Acessado em: 2017-05-09.
- [7] What is Bimodal IT? <http://www.gartner.com/it-glossary/bimodal/>. Acessado em: 2017-04-27.
- [8] *Um Guia do Conhecimento do Gerenciamento de Projetos (Guia PMBOK) - 5ª Edição*. Project Management Institute, Newtown Square, Pennsylvania, 2013.
- [9] ABHINAV, K., AND DUBEY, A. Predicting budget for crowdsourced and freelance software development

- projects. In *Proceedings of the 10th Innovations in Software Engineering Conference* (New York, NY, USA, 2017), ISEC '17, ACM, pp. 165–171.
- [10] ALEX GUAZZELLI. O Que é a Análise Preditiva? <https://www.ibm.com/developerworks/br/industry/library/ba-predictive-analytics1/index.html>. Acessado em: 2018-04-23.
- [11] BECK, K., BEEDLE, M., VAN BENNEKUM, A., COCKBURN, A., CUNNINGHAM, W., FOWLER, M., GRENNING, J., HIGHSMITH, J., HUNT, A., JEFFRIES, R., KERN, J., MARICK, B., MARTIN, R. C., MELLOR, S., SCHWABER, K., SUTHERLAND, J., AND THOMAS, D. Manifesto for Agile Software Development. <http://www.agilemanifesto.org/>, 2001. Acessado em: 2007-02-05.
- [12] BERNABÉ, R. B., NAVIA, I. A., AND GARCÍA-PEÑALVO, F. J. Faat: Freelance As a Team. 687–694.
- [13] COCKBURN, A. *Agile Software Development: The Cooperative Game (2Nd Edition) (Agile Software Development Series)*. Addison-Wesley Professional, 2007.
- [14] D. R. RABAN, S. R. The effect of source nature and status on the subjective value of information. *Journal of the American Society for Information Science and Technology*, pp. 321–329.
- [15] DE O. MELO, C., SANTOS, V. A., KATAYAMA, E. T., CORBUCCI, H., PRIKLADNICKI, R., GOLDMAN, A., AND KON, F. The evolution of agile software development in brazil - education, research, and the state-of-the-practice. *J. Braz. Comp. Soc.* 19, 4 (2013), 523–552.
- [16] GARTNER GROUP. Gartner Says CIOs Need Bimodal IT to Succeed in Digital Business. Tech. rep., nov 2014.
- [17] IEEE. IEEE standard glossary of software engineering terminology. *IEEE Std 610.12-1990* (Dec. 1990).
- [18] JEFF HOWE. CROWDSOURCING: A DEFINITION. <http://www.crowdsourcing.com/>. Acessado em: 2017-12-06.
- [19] KOKKODIS, M., PAPADIMITRIOU, P., AND IPEIROTIS, P. G. Hiring behavior models for online labor markets. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining* (New York, NY, USA, 2015), WSDM '15, ACM, pp. 223–232.
- [20] LARMAN, C. *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley Professional, 2003.
- [21] MIKE COHN. Scrum product backlog. <https://www.mountangoatsoftware.com/agile/scrum/scrum-tools/product-backlog>. Acessado em: 2017-07-29.
- [22] MOUNTAIN GOAT SOFTWARE. User stories. <https://www.mountangoatsoftware.com/agile/user-stories>. Acessado em: 2017-07-29.
- [23] NIC.BR / CETIC.BR. Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nas empresas brasileiras - TIC Empresas 2015, 2017.
- [24] NIC.BR / CETIC.BR. Pesquisa sobre o uso das Tecnologias de Informação e Comunicação nos domicílios brasileiros - TIC Domicílios 2016, 2017.
- [25] POCHMANN, H. *O Incrível Manual do Freelancer Moderno*. Aparelho Elétrico, 2014.
- [26] PRESSMAN, R. *Engenharia de Software - 6ª Edição*. McGraw Hill, 2006.
- [27] RABAN, D. R. User-centered evaluation of information: A research challenge. *Internet Research*, pp. 306–322.
- [28] SADLER, K., ROBERTSON, T., KAN, M., AND HAGEN, P. Balancing work, life and other concerns: A study of mobile technology use by australian freelancers. In *Proceedings of the 4th Nordic Conference on Human-computer Interaction: Changing Roles* (New York, NY, USA, 2006), NordiCHI '06, ACM, pp. 413–416.
- [29] SOMMERVILLE, I. *Engenharia de Software - 8ª Edição*. Addison-Wesley Brasil, 2007.
- [30] THE STANDISH GROUP INTERNATIONAL. The chaos report, 2015.
- [31] VERSIONONE. 11th annual state of agile report, 2017. Acessado em: 2017-07-25.
- [32] WORKANA. Valor Hora Freelance. <https://www.calculadorafreela.com/pt/valor-hora-freelance>. Acessado em: 2017-12-11.
- [33] XU, Z., AND ZHU, S. Abusing notification services on smartphones for phishing and spamming. In *Proceedings of the 6th USENIX Conference on Offensive Technologies* (Berkeley, CA, USA, 2012), USENIX Association.

APÊNDICE

A. FIGURAS

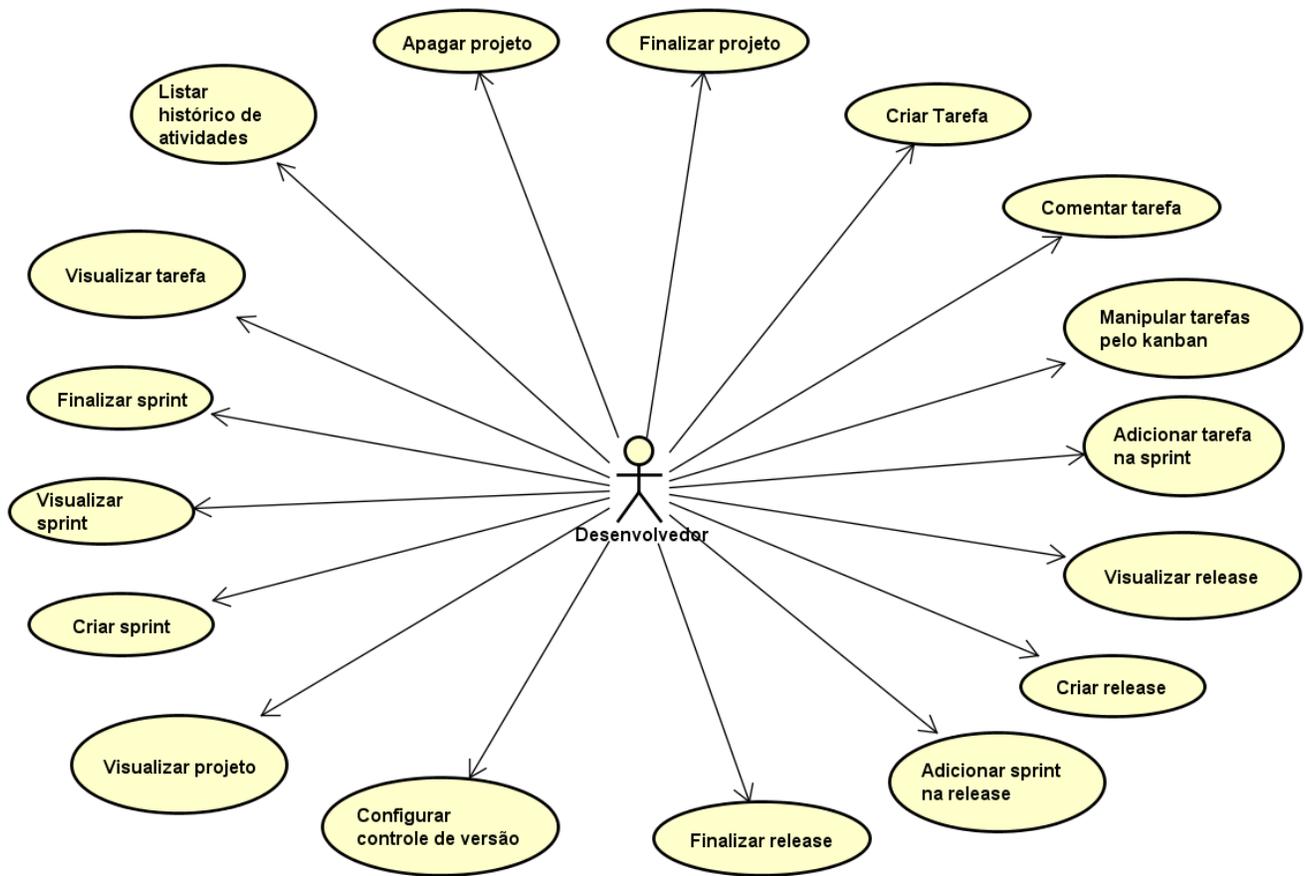


Figura A.1: Caso de uso usuário desenvolvedor

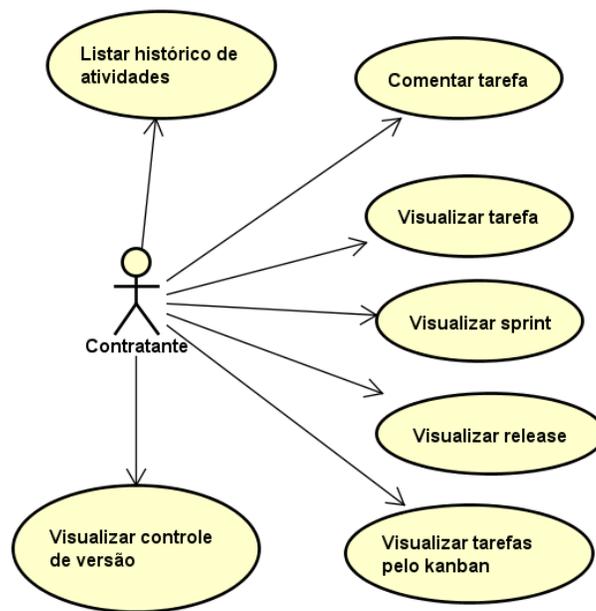


Figura A.2: Caso de uso usuário contratante

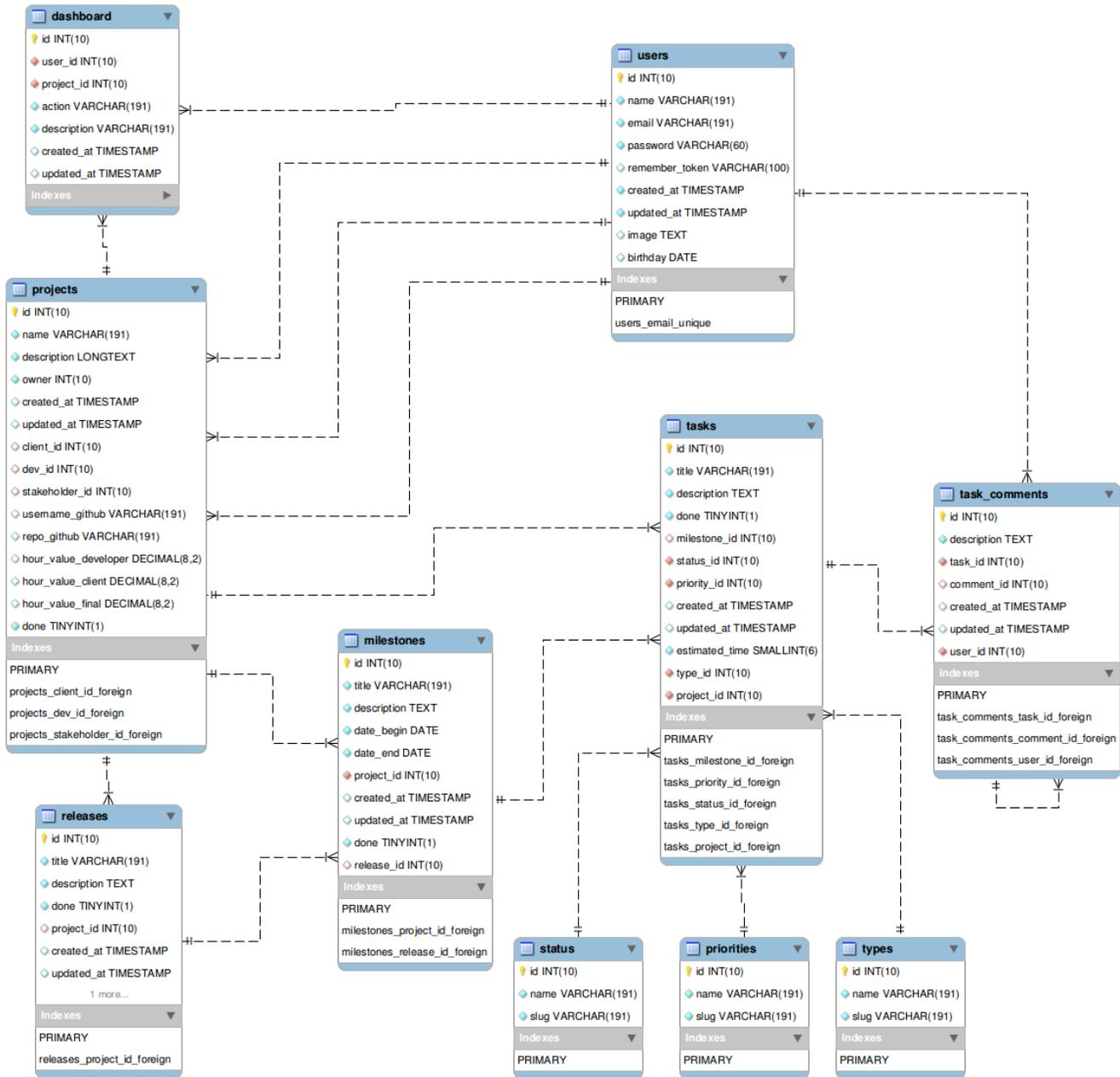


Figura A.3: Modelo de dados do Freelagile

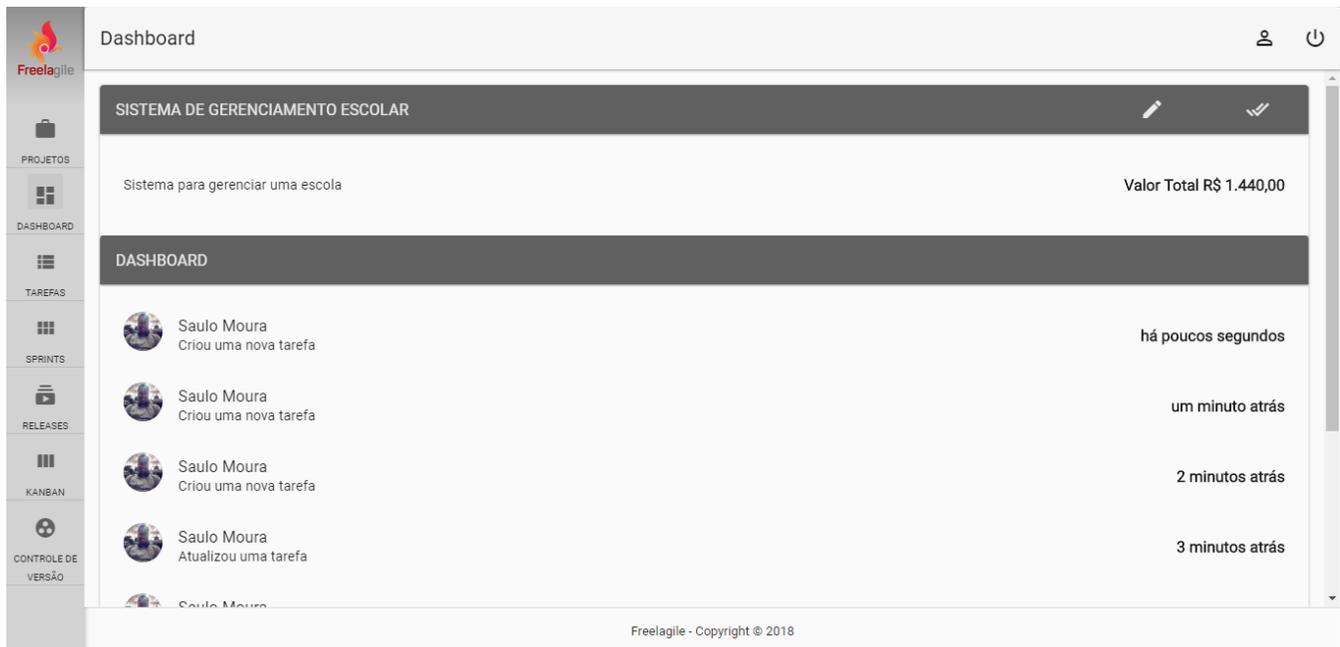


Figura A.4: Tela de dashboard de projeto no Freelagile

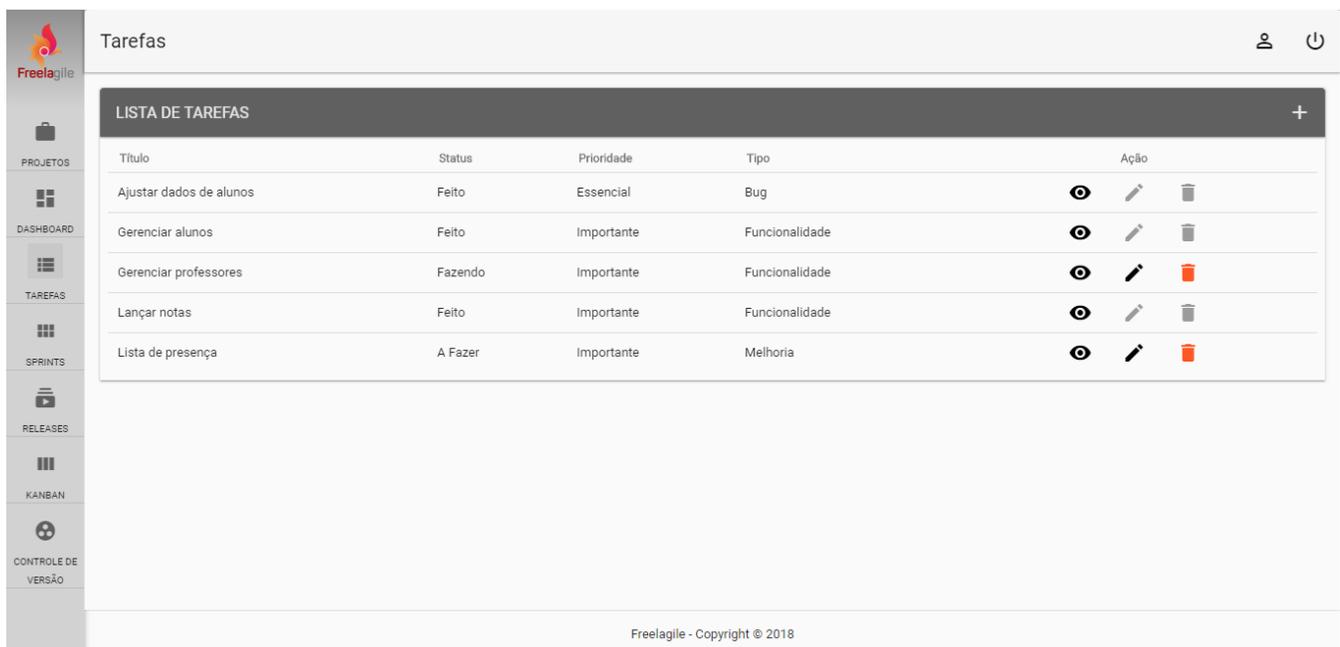


Figura A.5: Tela de tarefas do Freelagile

Sprints

Freelagile

PROJETOS

DASHBOARD

TAREFAS

SPRINTS

RELEASES

KANBAN

CONTROLE DE VERSÃO

LISTA DE SPRINTS

Título	Descrição	Data Estimada para Início	Data Estimada para Fim	Tempo Estimado	Valor Estimado	Ação
Gerenciar Alunos	Funcionalidades de gerencia de alunos	01/03/2018	17/03/2018	4 dias	R\$ 640,00	  
Gerenciar professores	Funcionalidades de gerencia de professores	18/03/2018	31/03/2018	5 dias	R\$ 800,00	  

TOTAL: 2 REGISTRO(S)

Freelagile - Copyright © 2018

Figura A.6: Tela de *sprints* do Freelagile

Releases

Freelagile

PROJETOS

DASHBOARD

TAREFAS

SPRINTS

RELEASES

KANBAN

CONTROLE DE VERSÃO

LISTAR

Título	Descrição	Data de Entrega	Ação
Módulo de gerência de ano letivo	Entrega que contempla o gerenciamento de pessoas e dados do ano letivo	Não Entrega	   

TOTAL: 1 REGISTRO(S)

Freelagile - Copyright © 2018

Figura A.7: Tela de *releases* do Freelagile

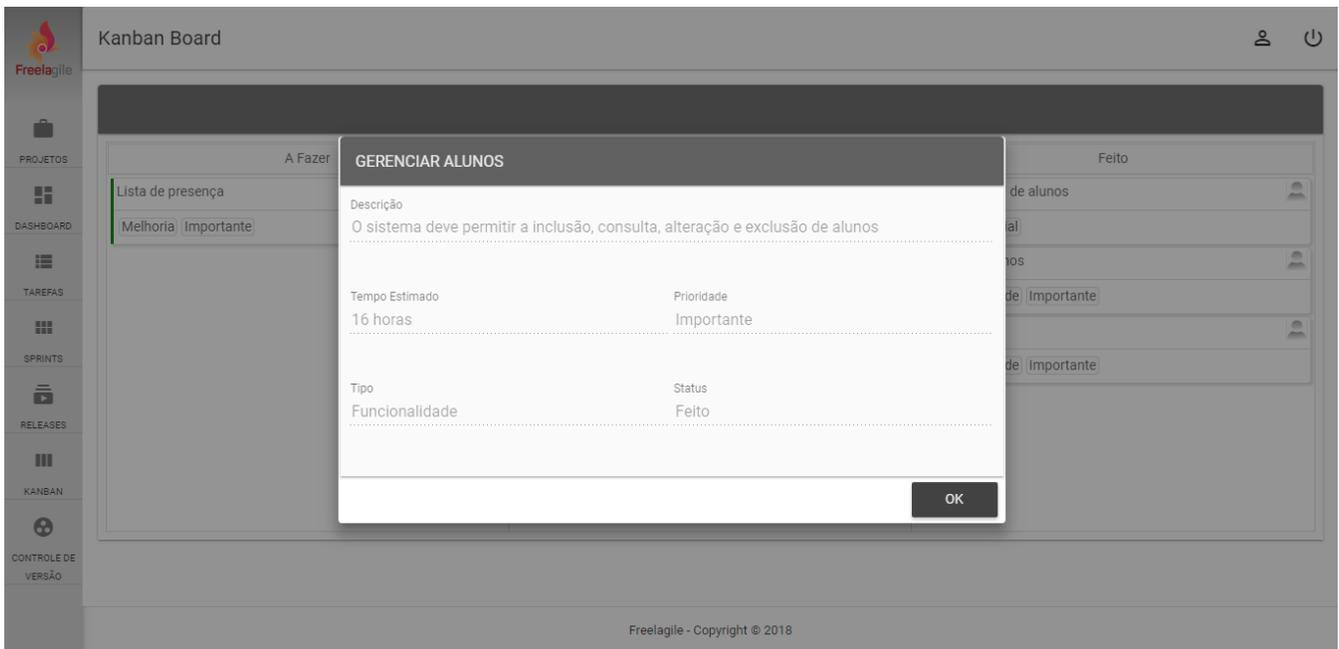


Figura A.8: *Kanban Board* do Freelagile



Figura A.9: Repositório de controle de versão no Freelagile



Data: __/__/____

Gênero: () Masculino () Feminino

1 – Possui ensino superior?

() Sim () Não

2- Caso possua, qual o curso?

3 – Qual o seu conhecimento na área de TI?

() Desenvolvimento

() Análise de Sistemas

() Infraestrutura (Redes de Computadores)

() Não possui conhecimento

4 – Você conhece algum método ágil de desenvolvimento de software?

() Sim () Não

5 – Já participou de algum projeto utilizando métodos ágeis?

() Sim () Não

6 – Já realizou algum trabalho freelancer?

() Sim () Não

7 - Utilizou algum tipo de metodologia para desenvolver o projeto?

() Sim () Não

8 - Descreva brevemente o fluxo de desenvolvimento do projeto.

9 – Já contratou o trabalho de algum desenvolvedor freelancer?

() Sim () Não

10 - Utilizou algum tipo de metodologia para acompanhar o projeto?

() Sim () Não

11- Descreva a forma de acompanhamento do projeto.

12 – Quais problemas foram enfrentados durante o projeto?

13 – Os projetos realizados obtiveram resultados satisfatórios? Se não, por quê?

() Sim () Não

Figura A.10: Formulário de Caracterização



Questionário de Feedback

Nome: _____

Papel no projeto: Desenvolvedor Contratante

1 - O vídeo tutorial apresentado ajudou você a entender o funcionamento da ferramenta?
 Sim
 Não

2 - Todo o processo foi claro e conciso?
 Sim
 Não

3 - Houve alguma dúvida não esclarecida? Se sim descreva a dúvida.
 Sim
 Não

4 - A metodologia ágil é bem empregada na ferramenta?
 Sim
 Não

5 - A ferramenta é intuitiva e de fácil navegação?
 Sim
 Não

6 - Você enfrentou problemas na utilização da ferramenta? Se sim descreva quais.

7 - Quais pontos positivos você pode ressaltar sobre a ferramenta?

8 - Quais pontos negativos você pode ressaltar sobre a ferramenta?

9 - Quais melhorias ou novas funcionalidades você poderia sugerir para a ferramenta?

Figura A.11: Formulário de *Feedback*

QUESTIONÁRIO PARA FREELANCER

1. Qual a quantidade anual de projetos que você executa?
2. Qual(is) a(s) sua(s) maior(es) dificuldade(s) no desenvolvimento dos projetos?
3. Como você estipula os prazos para entrega dos projetos?
4. Qual a quantidade média de projetos que você entrega dentro do prazo estipulado?
5. Como você define o valor a ser cobrado pelo desenvolvimento do projeto?
6. Como você define o tamanho do projeto?
7. Qual a técnica de estimativa de tamanho você utiliza?
8. Quais ferramentas de estimativa de tamanho você utiliza?
9. Qual a quantidade média dos projetos que você percebe que acaba por possuir um tamanho diferente do estipulado inicialmente?
10. Como você define a complexidade do projeto?
11. Qual a técnica de estimativa de complexidade que você utiliza?
12. Quais ferramentas para estimativa de complexidade você utiliza?
13. Qual a quantidade média dos projetos que você percebe que acaba por possuir uma complexidade diferente da estipulada inicialmente?
14. Como você define quantas horas serão gastas no projeto?
15. Qual a técnica de estimativa de horas você usa?
16. Quais as ferramentas de estimativa de horas você usa?
17. Qual a quantidade média dos projetos que você percebe que serão gastas uma quantidade de horas diferente da estipulada inicialmente?
18. Quais disciplinas do processo de desenvolvimento, além da codificação, você costuma realizar nos seus projetos?
19. Qual(is) ferramenta(s) você utiliza para gerenciar o desenvolvimento?
20. Como você apresenta os resultados ao contratante?
21. Qual ambiente virtual você utiliza para realizar a entrega do projeto?
22. Qual a importância que você atribui a geração de documentos sobre o projeto desenvolvido?
23. Qual documentação você costuma gerar nos projetos que desenvolve?
24. Quais as funcionalidades existentes em ferramentas para gerenciamento de projetos você considera usual para projetos freelance?
25. Caso exista alguma funcionalidade que você considera importante mas ainda não encontrou em nenhuma ferramenta descreva-a abaixo
26. Qual a quantidade anual de projetos que você contrata usando serviços de freelancers?

Figura A.12: Perguntas do questionário aplicado para *freelancers*

QUESTIONÁRIO PARA CONTRATANTE

1. Você costuma contratar serviços de freelancers para qual(is) das finalidades abaixo?
2. Como você define o prazo de entrega do projeto?
3. Como você define o custo do trabalho?
4. Qual(is) o(s) maior(es) benefício(s) ao contratar este tipo de serviço?
5. Qual(is) a(s) maior(es) dificuldade(s) ao contratar este tipo de serviço?
6. Como você acompanha o desenvolvimento do projeto contratado?
7. Qual(is) ferramenta(s) você usa para acompanhar o desenvolvimento da demanda?
8. Como é feita a comunicação entre você e o profissional contratado para tratar de questões referentes ao projeto?
9. Como é realizada a entrega do projeto?
10. Qual ambiente virtual você utiliza para realizar a entrega do projeto?
11. Qual a importância que você atribui a geração de documentos do projeto desenvolvido?
12. Qual documentação você costuma gerar ou cobrar a geração nos projetos contratados?
13. Quais as funcionalidades existentes em ferramentas para gerenciamento de projetos você considera usual para acompanhar os trabalhos contratados?
14. Caso exista alguma funcionalidade que você considera importante mas ainda não encontrou em nenhuma ferramenta descreva-a abaixo

Figura A.13: Perguntas do questionário aplicado para contratantes