

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
DEPARTAMENTO ACADÊMICO DE COMPUTAÇÃO
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

GUILHERME CASTRO DINIZ

**USO DE *GAMIFICATION* PARA MOTIVAR
ALUNOS DE GRADUAÇÃO A CONTRIBUIR COM
PROJETOS DE SOFTWARE LIVRE**

MONOGRAFIA

CAMPO MOURÃO

2016

GUILHERME CASTRO DINIZ

**USO DE *GAMIFICATION* PARA MOTIVAR
ALUNOS DE GRADUAÇÃO A CONTRIBUIR COM
PROJETOS DE SOFTWARE LIVRE**

Trabalho de Conclusão de Curso de graduação apresentado à disciplina de Trabalho de Conclusão de Curso 2, do Curso de Bacharelado em Ciência da Computação do Departamento Acadêmico de Computação da Universidade Tecnológica Federal do Paraná, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Igor Fábio Steinmacher

Coorientador: Prof. Dr. Marco Aurélio Graciotto Silva

CAMPO MOURÃO

2016



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Campus Campo Mourão
Departamento Acadêmico de Computação
Coordenação do Curso de Bacharelado em Ciência da Computação



ATA DA DEFESA DO TRABALHO DE CONCLUSÃO DE CURSO

Às vinte horas do dia vinte e dois de novembro de dois mil e dezesseis foi realizada no Miniauditório de EAD a sessão pública da defesa do Trabalho de Conclusão do Curso de Bacharelado em Ciência da Computação do acadêmico **Guilherme Castro Diniz** com o título **Uso de gamification para motivar alunos de graduação a contribuir com projetos de software livre**. Estavam presentes, além do acadêmico, os membros da banca examinadora composta pelo professor Dr. Igor Fábio Steinmacher (Orientador-Presidente), pelo professor Me. Filipe Roseiro Côgo, pelo professor Dr. Ivanilton Polato e pelo professor Dr. Marco Aurélio Graciotto Silva. Inicialmente, o aluno fez a apresentação do seu trabalho, sendo, em seguida, arguido pela banca examinadora. Após as arguições, sem a presença do acadêmico, a banca examinadora o considerou **aprovado** na disciplina de Trabalho de Conclusão de Curso e atribuiu, em consenso, a nota 8,0 (oito). Este resultado foi comunicado ao acadêmico e aos presentes na sessão pública. A banca examinadora também comunicou ao acadêmico que este resultado fica condicionado à entrega da versão final dentro dos padrões e da documentação exigida pela UTFPR ao professor Responsável do TCC no prazo de onze dias. Em seguida foi encerrada a sessão e, para constar, foi lavrada a presente Ata que segue assinada pelos membros da banca examinadora, após lida e considerada conforme.

Observações:

Campo Mourão, 22 de novembro de 2016

Prof. Me. Filipe
Roseiro Côgo
Membro

Prof. Dr. Ivanilton Polato
Membro

Prof. Dr. Marco Aurélio
Graciotto Silva
Membro

Prof. Dr. Igor Fábio
Steinmacher
Orientador

A ata de defesa assinada encontra-se na coordenação do curso.

Agradecimentos

É difícil agradecer todas as pessoas que de algum modo, nos momentos serenos e ou apreensivos, fizeram ou fazem parte da minha vida, por isso primeiramente agradeço à todos de coração.

Dedico este trabalho a Deus, por ser essencial em minha vida, autor do meu destino e meu guia. Aos meus pais, Lucas e Eloiza, aos meus irmãos, Gustavo e Gláucio (“*in memoriam*”) e a toda minha família que, com muito carinho e apoio, não mediram esforços para que eu chegasse até esta etapa de minha vida.

Agradeço a meu orientador, Igor Fábio Steinmacher, pela paciência na orientação e incentivo que tornaram possível a conclusão desta monografia. Obrigado por todos os churrascos, brincadeiras e ensinamentos que o senhor me proporcionou nesses quatro anos. Continue sendo um professor amigo e que gosta de ver seus alunos bem.

Agradeço a meu coorientador, Marco Aurélio Graciotto Silva, que contribuiu com ricas discussões sobre o tema desta monografia. Com uma “paciência de Jó” me apresentou e ensinou diversas coisas. Por todas oportunidades oferecidas pelo senhor, realizando projetos e ganhando conhecimento. E também pelo senhor ser um excelente professor e profissional, o qual me espelho.

Agradeço a todos os professores do curso, que foram tão importantes na minha vida acadêmica e no meu desenvolvimento pessoal. De vocês levarei a ética no qual aprendi ao longo desses anos.

Agradeço aos amigos que fiz na faculdade, principalmente a Mariane, Douglas e Willian, amigos que me acompanharam e me ajudaram do início ao fim da graduação. Sem vocês eu não teria me formado, obrigado por tudo amigos.

Um agradecimento também aos amigos e colegas da *HackerSpace*, aos companheiros de república, aos colegas de ambiental, civil e eletrônica.

Resumo

DINIZ, G. C.. Uso de *gamification* para motivar alunos de graduação a contribuir com projetos de software livre. 2016. 68. f. Monografia (Curso de Bacharelado em Ciência da Computação), Universidade Tecnológica Federal do Paraná. Campo Mourão, 2016.

Contexto: O modelo de desenvolvimento de software livre alterou a forma na qual software é construído, estendendo o processo de desenvolvimento para um grande número de voluntários. Sabe-se que a entrada de novatos é custosa, pois para iniciar a contribuição em um projeto é necessário superar várias barreiras. Atualmente, alunos de graduação da área de computação demonstram interesse em (ou são motivados a) contribuir, porém não contribuem devido à desmotivação frente às barreiras encontradas. Por consequência, várias pesquisas foram conduzidas com o objetivo de engajar, reconhecer e motivar desenvolvedores na contribuição de software livre.

Objetivo: Tendo em vista o potencial existente nos alunos de graduação em contribuir com projetos de software livre, o objetivo deste trabalho é verificar se o uso de elementos de *gamification* motiva alunos de graduação do curso de Ciência da Computação a contribuírem com projetos de Software Livre.

Método: Foram selecionados os elementos de *design* de jogos – *Quest*, *Rank*, Ponto e Nível – que serão utilizados para motivar os jogadores. Na sequência foram definidas regras para utilização dos elementos de design de jogos. Tais regras foram implementadas no ambiente GitLab. Para avaliar os elementos, foi conduzido um estudo em sala de aula, com 17 estudantes, utilizando um projeto de software livre real (JabRef). Ao final do estudo, os estudantes avaliaram sua experiência por meio de um questionário estruturado, contendo questões objetivas.

Resultados: As respostas do questionário mostraram que o elemento *Quest* teve um desempenho superior aos outros elementos, obtendo resultados satisfatórios na maioria das avaliações. O elemento Ponto também mostrou bons resultados, as melhores avaliações foram no fato do elemento ajudar a dar *feedback* conforme as ações realizadas pelos usuários. Os demais elementos (nível e ranking) não tiveram resultados tão conclusivos quanto os elementos *Quest* e Ponto. Em resumo o ambiente *gamified* ajudou os estudantes na tentativa de fazer uma contribuição.

Conclusões: Elementos de *gamification* podem motivar e apoiar novatos entrando em

projetos de software livre. Em especial, destacam-se os elementos *quest* e Ponto, que mostraram-se adequados para auxiliar os novatos a superarem barreiras de falta de orientação e *feedback*, além de ser motivador. Assim, indica-se o uso dos elementos *Quest* e Ponto para comunidades de software livre a fim de apoiar os primeiros passos de potenciais novos membros.

Palavras-chaves: gamification, software livre, novatos, barreiras, motivação

Abstract

DINIZ, G. C.. Use of gamification to motivate graduate students to contribute to open source projects. 2016. 68. f. Monograph (Undergraduate Program in Computer Science), Federal University of Technology – Paraná. Campo Mourão, PR, Brazil, 2016.

Context: *The Open Source Software development model has changed the way that software is built, extending the development process for a large number of volunteers. It is known that the onboarding of newcomers is difficult, because to contribute to a project it is necessary to overcome a number of barriers. Currently, undergraduate students in the area of computing show interest (and sometimes are motivated) to contribute to open source. However, the students do not do it due to lack of motivation to overcome the barriers encountered. As a result, several studies have been conducted with the objective of engaging, attracting and motivating developers to open source projects.*

Objective: *Considering the potential of undergraduate students in contributing to open source projects, the objective of this work is to verify if the use gamification elements motivate undergraduate students of Computer Science to contribute to open source software projects.*

Method: *After identifying the elements of game design used to motivate the students, rules were defined for each element. Such rules have been implemented in the environment GitLab. To assess the elements, a study was conducted into the classroom, with 17 students, using a real open source software project (JabRef). At the end of the study, the students evaluated their experience through a structured questionnaire.*

Results: *The questionnaire answers showed that the element Quest had a good performance. The Point element also showed good results, the best ratings were in fact of the element help give feedback as the actions performed by users. The other elements (ranking and levels) had not presented good results. In summary, the gamified environment helped the students in an attempt to make a contribution.*

Conclusions: *Gamification can motivate and support newcomers onboarding to open source projects. In particular, elements Quest and Point, which proved to be adequate to assist newcomers to overcome barriers of lack of guidance and feedback, besides being motivator. Thus, we propose that communities take advantage of using Quests and points to support the first steps of potential new members.*

Keywords: *gamification, open source software, newcomers, barriers, motivation*

Lista de figuras

2.1	Grupos de perfis de jogadores	17
2.2	Aplicação dos elementos de <i>gamification</i>	21
3.1	Visão geral do projeto	23
3.2	Diagrama de como os elementos irão agir no sistema	27
4.1	Modelo de Entidade Relacionamento das tabelas que foram criadas e/ou sofreram modificação	32
4.2	Interface principal do ambiente GitLab <i>gamified</i>	34
4.3	Interface referente ao histórico de tarefas realizadas pelo usuário	35
4.4	Interface do <i>rank</i> , com a opção de exibição por pontos.	35
4.5	Interface do <i>rank</i> , com a opção de exibição por nível.	36
4.6	Interface do elemento de <i>design</i> de jogos <i>quest</i>	37
4.7	Interface do conjunto de tarefas que juntas formam uma <i>quest</i>	38
4.8	Ambiente onde é solicitado a realização de uma atividade	39
5.1	Avaliação do elemento de <i>design</i> de jogos <i>quest</i>	43
5.2	Avaliação do elemento de <i>design</i> de jogos ponto	45
5.3	Avaliação do elemento de <i>design</i> de jogos nível	46
5.4	Avaliação do elemento de <i>design</i> de jogos <i>rank</i>	47
5.5	Conjunto de boxplot referente a avaliação dos elementos x barreiras - Barreira Falta de Orientação	49
5.6	Conjunto de boxplot referente a avaliação dos elementos x barreiras - Barreira Falta de <i>Feedback</i>	50
5.7	Conjunto de boxplot referente a avaliação dos elementos x barreiras - Barreira Falta de conhecimento prévio dos novatos	51
5.8	Conjunto de gráficos de avaliação do Ambiente <i>gamified</i>	54

Lista de tabelas

5.1	Perguntas relacionadas à contribuição em projetos de software livre	42
5.2	Avaliação dos elementos de <i>design</i> de jogos - questão 1	52
5.3	Avaliação dos elementos de <i>design</i> de jogos - questão 2	52
5.4	Avaliação dos elementos de <i>design</i> de jogos - questão 3	53

Sumário

1	Introdução	10
2	Revisão Bibliográfica	13
2.1	Software Livre	13
2.2	Motivação	15
2.3	<i>Gamification</i>	16
2.3.1	Tipos de Jogadores	17
2.3.2	Mecânica de Jogo	19
2.3.3	Elementos de <i>design</i> de jogos	20
2.3.4	Regras	21
2.4	Considerações Finais	22
3	Método de Pesquisa	23
3.1	Definição dos Perfis de Jogadores	24
3.2	Definição dos Elementos de <i>Design</i> de Jogos	25
3.3	Definição das Regras	26
3.4	Elementos de <i>design</i> de jogos x barreiras	29
4	Implementação	31
4.1	GitLab	31
4.2	Modelo de Dados	32
4.3	Descrição das Interfaces	33
4.3.1	Interface Principal	33
4.3.2	Interface de Histórico de Tarefas Realizadas	34
4.3.3	Interface do <i>rank</i>	35
4.3.4	Interface das <i>quests</i>	36
4.3.5	Interface das <i>Tarefas</i>	37
4.3.6	Implementação das Regras	38
5	Estudo Preliminar e Resultados	40
5.1	Questões de Pesquisa	40
5.2	Estudo Preliminar	40

5.2.1	Coleta de dados	41
5.3	Resultados	41
5.3.1	Contribuição com Projetos de Software Livre	42
5.3.2	Avaliação Unitária dos Elementos de <i>Design</i> de Jogos	42
5.3.3	Avaliação dos Elementos de <i>Design</i> de Jogos x Barreiras	48
5.3.4	Avaliação Geral dos Elementos de <i>Design</i> de Jogos	51
5.3.5	Avaliação do Ambiente <i>Gamified</i>	53
6	Conclusão	56
6.0.1	Limitações	57
	Referências	58
A	Questionário utilizado no estudo	62

Introdução

O modelo de desenvolvimento de software livre alterou a forma na qual software é construído, estendendo o processo de desenvolvimento para um grande número de voluntários, muitas vezes sem *design* ou planos explícitos (MOCKUS et al., 2000). Sabe-se que a entrada de novatos é algo difícil, pois para contribuir é necessário superar várias barreiras.

Steinmacher et al. (2015) apresenta diversas barreiras que novatos encontram quando começam a contribuir com projetos de software livre. Dentre as barreiras encontradas, ressaltam-se o desconhecimento do processo de contribuição por parte dos novatos e a má recepção da comunidade (contribuidores mais experientes algumas vezes são rudes) (STEINMACHER et al., 2015). Por exemplo, quando novos membros juntam-se a um projeto é difícil para eles entenderem o raciocínio por trás da estrutura e organização de um projeto. Isso age como uma barreira para a contribuição em projetos de software livre, desmotivando desenvolvedores novatos a fazerem contribuições.

Existe hoje alunos de graduação em Computação de vários níveis de conhecimento, que possuem competência técnica e interesse em contribuir, porém não contribuem devido às barreiras encontradas. Para que esses alunos participem em projetos como contribuintes de código é necessário que as vantagens percebidas por estes sejam maiores que as barreiras encontradas para entrar no projeto. As vantagens não têm um âmbito financeiro (pelo menos não na maioria dos projetos), o que atrai o usuário à participação são os interesses que este possui pelo projeto em si (principalmente se ele for usuário do software), interesse pela tecnologia envolvida, aumento de *status* dentro da comunidade e a possibilidade de ter seu código examinado por especialistas da área (CHRISTOPH, 2004).

Devido às barreiras encontradas por novatos, várias pesquisas foram conduzidas com o objetivo de engajar, reconhecer e motivar desenvolvedores na contribuição de software livre (RYAN; S., 2014; PRAUSE; VINKOVITS, 2012; VASILESCU, 2014; C. REINERS R, 2010; BERGQUIST; LJUNGBERG, 2001; STEINMACHER et al., 2015). Nos últimos anos, o

termo *gamification* tem chamado atenção para ser aplicado além do ambiente dos jogos e realizar a abordagem de tarefas de colaboração, como desenvolvimento de software. Uma das possíveis maneiras de promover engajamento e motivação é por meio do uso de *gamification*, que consiste na utilização de elementos de *design* de jogos em contextos de “não-jogos” (DETERDING et al., 2011).

Adiciona-se à motivação o fato da importância de motivar alunos de ciência da computação a contribuírem com projetos de software livre. Tal abordagem traz diversos benefícios para os alunos: além da questão motivacional, os alunos seriam fortemente estimulados a desenvolver habilidades de compreensão e análise de programas. Os alunos podem praticar suas habilidades de programação em um projeto real e colaborativo, tornando o aprendizado mais interessante e motivador. Para contribuir no desenvolvimento de um software livre, os alunos teriam que previamente analisar e entender o código do software livre em questão. Procurando entender o código alheio, os alunos também entenderiam, na prática, a importância de uma boa documentação, estilo e modularização do código. Além disso, eles estariam desde cedo se habituando a contribuir em projetos de software livre. Isto poderia gerar um aumento significativo no número de colaboradores e, conseqüentemente, fornecer um maior incentivo ao desenvolvimento de projetos desse tipo.

Os mecanismos de jogos podem ser usados para conduzir quase qualquer comportamento do usuário dentro de um ambiente *gamified*. A *gamification* é utilizada em diferentes contextos, por exemplo, Halan et al. (2010) utiliza *gamification* como estratégia de motivação para persuadir os estudantes de medicina à modelarem a conversação de seres humanos virtuais. Já Li et al. (2012) apresenta o GamiCAD, um sistema tutorial interativo *gamified* para novos usuários do AutoCAD. Mais voltado para as características deste trabalho, há também o uso de *gamification* na educação, onde o engajamento foi identificado como um valioso indicador de realizações acadêmicas dos estudantes (MCMAHON; PORTELLI, 2004). Os estudantes engajados são atraídos para o seu trabalho, persistem em suas atividades acadêmicas apesar dos desafios e obstáculos e visivelmente sentem prazer em realizá-las (SCHLECHTY, 1994).

O principal objetivo deste trabalho é utilizar a técnica *gamification* para motivar e apoiar alunos de graduação do curso de Ciência da Computação a contribuírem com projetos de software livre. Para realizar essa tarefa, foram selecionados elementos de *design* de jogos, como: *ranks*, pontos, níveis e *quests* – esses elementos serão utilizados para conduzir os usuários a fazerem contribuições com codificação. Para cada um dos elementos de *design* de jogos foram elaboradas regras, essas regras definem o sistema de recompensas, descreve o funcionamento do ambiente *gamified*. O público escolhido foram alunos de computação, pois há a oportunidade de incentivar esses alunos a contribuir com projetos de software livre como parte da formação acadêmica.

Pretende-se utilizar os elementos de *design* de jogos como uma forma de transpor

as barreiras encontradas por novatos ao tentarem contribuir com projetos de software livre. As barreiras a qual foram trabalhadas no estudo são “*Falta de conhecimento prévio dos novatos*”, “*Falta de feedback*” e “*Falta de orientação*”.

O método aplicado para condução deste trabalho inicia-se com a escolha dos elementos de *design* de jogos e regras relacionadas, seguido da implementação de tais elementos e regras no ambiente *gamified*. Essas alterações foram feitas no GitLab, que é um gerenciador de repositório de software. Em seguida, foi conduzido um estudo preliminar, no qual um grupo de estudantes foi submetido a um estudo utilizando o ambiente com os elementos de jogos. Em seguida foram aplicados questionários aos estudantes a fim de avaliar os elementos utilizados.

O restante desta monografia está organizado da seguinte forma. O Capítulo 2 apresenta uma revisão bibliográfica de trabalhos que utilizaram técnicas de *gamification* em diversos contextos. Também são apresentadas definições formais de software livre, motivação, *gamification* e características importantes que existem na técnica *gamification* como: mecânica de jogos, elementos de *design* de jogos e as regras. No Capítulo 3 é explanado o método proposto para o desenvolvimento deste trabalho, apresentando o que será desenvolvido durante todo o estudo. Na Capítulo 4 são apresentadas as implementações realizadas durante o estudo. No Capítulo 5 são apresentados detalhes do estudo preliminar e os resultados. O último Capítulo 6 é referente às conclusões deste trabalho.

Revisão Bibliográfica

Neste Capítulo foram definidos conceitos relacionados com software livre e *gamification*. Na Seção 2.1 são apresentados o conceito de software livre, a importância da entrada de novos contribuidores e os ambientes sociais. Na Seção 2.2 são descritos alguns estudos de motivação e engajamento no contexto de contribuição com software livre. A Seção 2.3 traz uma definição de *gamification*, conceitos básicos, tipos de jogadores, mecânica de jogo, elementos de *design* de jogos e regras. A Seção 2.4 é uma análise dos conteúdos tratados e as limitações da proposta.

2.1. Software Livre

Software Livre é o software que respeita a liberdade e senso de comunidade dos usuários. Os usuários possuem a liberdade de executar, copiar, distribuir, estudar, mudar e melhorar o software (FSF, 2015). Comunidades de Software Livre são organizações virtuais formadas por indivíduos que trabalham juntos no desenvolvimento de um software livre específico (REIS et al., 2003). O sucesso de um projeto em termos de produção do software está relacionado com o crescimento do tamanho da comunidade de desenvolvedores e pessoas que contribuem para o bem público de um software livre, escrevendo código para o projeto (MOODY, 2001; REYMOND, 1999; SAWHNEY, 2000; WAYNE, 2000).

Sabe-se que, para uma comunidade ser mantida, a entrada de novos membros é essencial (ALENCAR et al., 2009). Porém existem diversas barreiras encontradas por novatos quando estes tentam contribuir com codificação (STEINMACHER et al., 2015). Fichman e Kemerer (1997) verificaram que, em desenvolvimento de sistemas de software comerciais, tecnologias complexas podem erguer barreiras significativas ao entendimento e contribuição, para ambos os usuários e desenvolvedores de software, e a integração dos recém-chegados pode ser árdua.

Complementarmente, além disso há um custo elevado para novatos e veteranos tornarem-se desenvolvedores do núcleo de um projeto, principalmente em projetos de software livre consolidados. É preciso uma quantidade significativa de tempo para aprender os aspectos técnicos e sociais dos projetos e um montante adicional de tempo para ganhar confiança, demonstrar habilidades e acumular reputação (KROGH et al., 2003).

Segundo Steinmacher et al. (2015) as principais barreiras encontradas por novatos são:

- **Problemas de recepção** - isso ocorre muito em projetos grandes e nas comunidades mais antigas onde geralmente programadores experientes são rudes, não respondem às perguntas dos recém-chegados ou demoram a responder e respondem de uma forma complexa. Programadores mais experientes acreditam que os novatos deveriam saber mais sobre o projeto antes de entrar em contato.
- **Comportamento dos novatos e conhecimento prévio dos novatos** - os aprendizes não são proativos, não têm compromisso e nem paciência. Geralmente eles não possuem conhecimentos técnicos necessários e subestimam os desafios.
- **Falta de orientação** - geralmente o processo de contribuição não possui um mapa sobre como o novato deve começar e uma descrição dos primeiros passos para começar a contribuir. Ter um membro mais experiente para auxiliar o novato no início é de grande ajuda, pois é mais fácil selecionar um *bug* que esteja nivelado aos seus conhecimentos, bem como instalar o ambiente desenvolvedor, reproduzir o *bug* entre outras tarefas.
- **Documentação com problemas** - a qualidade da documentação em projetos de software livre como um todo ainda deixa muito a desejar. Muitas vezes a documentação é considerada uma atividade não tão “nobre” quanto à codificação, o que acarreta que estas sejam deixadas em segundo plano, resultando em uma documentação incompleta, desatualizada ou insuficiente.
- **Diferenças culturais** - alguns recém-chegados precisam entrar em contato com uma pessoa real. Acontece também de suas mensagens serem interpretadas de forma errada.
- **Características do código ou problemas de entendimento** - muitas vezes, quando novos membros juntam-se a um projeto, é difícil para eles entenderem o raciocínio por trás da estrutura e organização de um projeto. Esta característica está muito ligada à falta ou desatualização da documentação.

Essas barreiras desmotivam fortemente novatos a entrarem para uma comunidade e contribuírem. Dadas estas dificuldades iniciais, existem pesquisas que visam diminuir essas barreiras e engajar recém-chegados, facilitando a entrada de novos membros para que contribuam frequentemente (HARS; OU, 2002; BERGQUIST; LJUNGBERG, 2001; WU et al., 2007).

Neste estudo, tem-se algumas dessas características, isto é, estudantes que estão envolvidos com a computação, que possuem conhecimentos técnicos e também estudantes

com pouco conhecimento e competência técnica, que por estas barreiras ou mesmo por não conhecerem o mundo do software livre, não contribuem ou não são motivados a fazerem contribuições. Nosso objetivo é motivar esses alunos, fazendo com que eles superem essas barreiras.

2.2. Motivação

Antes de entrar no contexto de *gamification*, uma distinção clara entre motivação intrínseca e extrínseca deve ser feita.

Motivação extrínseca: tem origem em fatores externos ao indivíduo, como qualquer recompensa monetária. O indivíduo faz a tarefa para ser recompensado. A recompensa é o “combustível” que faz mobilizar o sujeito. Quando retirado, o sujeito vai deixar de se mobilizar, de estar motivado, visto que não tem nada a ganhar nem a perder se não executar a tarefa. Estudos indicam que este tipo de motivação, é muito inconstante, visto que depende de fatores externos (LOMBRISER, 2014). O indivíduo não gosta da tarefa em si, mas gosta da recompensa que a tarefa ao ser executada lhe pode trazer, o que implica necessariamente pouca satisfação e prazer na execução da tarefa.

Motivação intrínseca: tem origem em fatores internos ao indivíduo, esta relaciona-se com a sua forma de ser, os seus interesses, os seus gostos. Coon (2008) descreve motivação intrínseca quando as pessoas simplesmente desfrutam de uma atividade ou quando a veem como uma oportunidade para explorar, aprender e atualizar suas potencialidades sem qualquer recompensa. Este tipo de motivação é constante, visto que depende unicamente do sujeito e não de fatores externos (LOMBRISER, 2014).

Muitas pesquisas já foram desenvolvidas para analisar a motivação de desenvolvedores que contribuem para projetos software livres. Por exemplo, Hars e Ou (2002) sugerem que os desenvolvedores intrinsecamente motivados gastam mais tempo e esforço em projetos de software livre do que os desenvolvedores extrinsecamente motivados, mas não examinou isto empiricamente. Outros estudos empíricos se concentraram em motivação intrínseca ao invés de motivação extrínseca. Por exemplo, Lakhani e Hippel (2000) vinculam sentimentos de competência e diversão à vontade de ajudar outros desenvolvedores. Outros autores também têm visto motivação em relação de reciprocidade, por exemplo, dar *patches* de software como “presente” para a Comunidade (BERGQUIST; LJUNGBERG, 2001; WU et al., 2007).

Para ambas as motivações, a técnica de *gamification* pode ser aplicada. É importante determinar qual motivação (extrínseca ou intrínseca) irá guiar o estudo. A motivação define os elementos de *design* de jogos. Cada um dos elementos motiva o jogador de certa forma. Assim, para este trabalho o foco será utilizar a motivação extrínseca.

2.3. *Gamification*

gamification é definido por Deterding et al. (2011) como a utilização de elementos de *design* de jogos em contextos de “não-jogos”. O conceito de *gamification* é diferente de um jogo educativo ou um jogo comum. Jogos educativos ou jogos comuns descrevem o *design* de jogos completos para fins de entretenimento, enquanto aplicações “*gamified*” apenas empregam elementos de jogos.

Nos últimos anos, *gamification* teve uma rápida aprovação em negócios, marketing e gestão empresarial. Isto é impulsionado pelo seu potencial para moldar o comportamento dos usuários em uma direção desejável. Segundo Zichermann e Cunningham (2011), *gamification* é 75% de psicologia e 25% de tecnologia relacionada, e tem um forte incentivo para um maior engajamento. Programas como *Foursquare* e *Nike+* são dados frequentemente como exemplos de produtos de sucesso que utilizam um ambiente *gamified* (DICHEVA et al., 2015). Sites como o *eBay* e *Fitocracy* utilizam elementos de *design* de jogo para manter as pessoas envolvidas e para incentivar competição amigável entre os usuários.

Uma das áreas em que a *gamification* é aplicada é na Engenharia de Software. Um mapeamento sistemático realizado por Pedreira et al. (2015) sobre *gamification* em engenharia de software mostra em quais as áreas da engenharia de software a *gamification* foi mais aplicada: Desenvolvimento de Software, Colaboração e Projetos de avaliação e controle. No mapeamento foram considerados 29 artigos, sendo estes estudos primários. Na área de colaboração, que é a que segue a linha de pesquisa deste trabalho, os artigos utilizam *gamification* para motivar colaboradores com a documentação e melhores práticas de desenvolvimento dos softwares livres (RYAN; S., 2014; PRAUSE; VINKOVITS, 2012; C. REINERS R, 2010).

Nos estudos que abordam a aplicação de *gamification*, não existem artigos cujo objetivo principal é engajar alunos de graduação na contribuição com software livre. É verdade, no entanto, que foram encontrados trabalhos que seguem uma linha de pesquisa próxima ao objetivo proposto neste trabalho, porém aplicadas em contextos diferentes. Nesta linha, Vasilescu (2014) busca compreender como os aspectos humanos, *gamification* e mídias sociais (por exemplo, participação em ambientes sociais, como *Stack Overflow* ou *GitHub*) impactam na colaboração distribuída em softwares de código aberto. Sheth et al. (2012) mostram que *gamification* pode ser usado para aumentar o envolvimento do aluno e o aprendizado em teste de software básico, propondo o envolvimento do aluno no desenvolvimento, documentação, relatórios de *bugs*, e cobertura dos testes. Melo et al. (2014) propõem um *ranking* via *feed* de notícias que extrai as informações de repositórios utilizando métricas de engenharia de software, a fim de construir um sistema de classificação, que destaca e premia desenvolvedores mais ativos. O objetivo é determinar se a utilização de *gamification* encoraja a colaboração e o empenho de todos os envolvidos em projetos de

desenvolvimento de software.

2.3.1. Tipos de Jogadores

Pesquisas têm sido feitas para classificar os jogadores em o que é conhecido como “Tipos de Jogadores” para sintetizar vários comportamentos exibidos durante um jogo. Psicólogos, por exemplo, identificaram que os tipos de jogadores possuem relações com os pré-existentes tipos de personalidade e que tipos de jogadores são essencialmente outra sintetização da personalidade (MCMAHON et al., 2012; LORENA, 2010). Essa relação indica que os tipos de jogadores e tipos de personalidades são essencialmente as mesmas sintetizações, apenas em contextos diferentes. Independente de tais contextos, nossa personalidade se reflete em como interagir com mundos virtuais como fazemos na realidade (FERRO et al., 2013).

A diversidade de jogos existentes reflete a pluralidade de perfis reconhecíveis entre os próprios jogadores (VIANNA et al., 2013). Bartle (1996) passou a considerar que a vasta gama de perfis poderia ser resumida em quatro grupos abrangentes de jogadores, de acordo com a Figura 2.1. Esses grupos são detalhados a seguir:



Figura 2.1. Grupos de perfis de jogadores
Fonte: (BARTLE, 1996)

- **Predadores (*Killers*):** jogadores com esse perfil entram na competição apenas motivados por derrotar os adversários. Não importa o que está em disputa, apenas ser o melhor. Ao longo do jogo, adotam comportamento agressivo, sendo suas intervenções notadas como mais incisivas e focadas em assegurar a condição de liderança (VIANNA et al., 2013). Eles costumam provocar os demais jogadores com a exaltação de seus triunfos, ou mesmo prejudicá-los diretamente, caso eles atrapalhem seus objetivos dentro do jogo (BARTLE, 2003, 1996).

- **Conquistadores (*Achievers*):** esses jogadores apreciam a sensação constante de vitória, ainda que o objetivo a ser alcançado não seja tão significativo. Sua motivação central consiste na realização de todas as atividades que o jogo apresenta, a partir de uma dedicada imersão em seu universo (BARTLE, 2003, 1996). Não zelam pelas relações sociais, mas as estabelecem de maneira cordialmente competitiva, mesmo que não estejam à frente no placar. Os realizadores são mais identificados com o fato de se destacarem dos oponentes de maneira leal, por meio de conquistas próprias (VIANNA et al., 2013).
- **Exploradores (*Explorers*):** o terceiro grupo é composto por interessados em desvendar todas as possibilidades e porquês do jogo. Por serem curiosos, podem chegar a dedicar-se a estudos ou ao desenvolvimento de habilidades que os ajudem a solucionar desafios específicos. Em sua visão, outros jogadores também adicionam positivamente à experiência, mas não são considerados essenciais. Como fator de diferenciação, valorizam o reconhecimento da comunidade quanto ao grau de conhecimento demonstrado, apresentando duas motivações paralelas: a fuga da realidade e, principalmente, o aprendizado decorrente da atividade. Para esse perfil, o mais importante é a trajetória e não a conquista (BARTLE, 2003, 1996).
- **Socializadores (*Socializers*):** O quarto e último tipo dentre os mais significativos perfis de jogadores são os socializadores. Como a própria denominação sugere, tratam-se daqueles que vislumbram, por intermédio dos jogos, uma oportunidade de interação social. Mais importante do que atingir os objetivos propostos ou concluir as tarefas designadas, é a ocasião do jogo em si e seu potencial de estimular vínculos sociais que os interessam. Os socializadores costumam preferir jogos cooperativos, que demandam trabalho conjunto e evidenciam personalidades colaborativas (BARTLE, 2003, 1996).

Fullerton (2008) descreve os diferentes tipos de jogadores com base nas agendas e a necessidade ao entrar em um espaço de jogo, bem como abordar os prazeres do jogo do ponto de vista do jogador. Ele classifica os tipos de jogadores mais detalhado, como: Competidores (*Competitor*), Coletador (*Collectors*), Brincalhão (*Joker*), Artista (*Artist*) e vários outros tipos.

Com base nos estudos em Ferro et al. (2013), Lorena (2010), Staewen et al. (2014), Lombriser (2014), Bartle (2003, 1996), Fullerton (2008) foi feita uma classificação dos perfis de jogadores. O objetivo foi identificar as características, recompensa, motivações e os tipos de elementos de *design* de jogos que é mais adequado para cada tipo de jogador.

Após análise, foi necessário entender quais eram os perfis dos potenciais participantes para este trabalho, tendo em vista o universo de estudantes de computação, que é o foco deste trabalho. Assim, foi discutido as características dos alunos que poderiam ser levadas em consideração no contexto do jogo. Definiu-se que classificar os alunos com relação ao conhecimento técnico seria interessante. Decidiu-se estudar neste trabalho os alunos que

possuem conhecimento intermediário sobre desenvolvimento de software, alunos dos períodos finais do curso de Ciência da Computação. É importante ressaltar que os perfis de jogadores são transversais aos tipos de jogadores, sendo que podem ser encontrados jogadores com baixo e alto conhecimento técnico que se enquadrem nos quatro tipos de jogadores definidos por Bartle (2003).

2.3.2. Mecânica de Jogo

Transformar uma atividade cotidiana em uma experiência de jogo, tornando-a desafiadora e divertida ao mesmo tempo, requer um esforço muito grande. Há várias abordagens que descrevem mecanismos dos jogos. De acordo com Cook (2006), mecanismos do jogo são baseados em um sistema de regras e simulações, que facilitam e incentivam o usuário a explorar e aprender em muitas possibilidades através da utilização de mecanismos de *feedback*. Snow (2010) escreve que mecânica do jogo pode ser usado para conduzir quase qualquer comportamento do usuário. Ela tem o potencial para tocar em toda a gama de emoções humanas e motivar uma vasta gama de comportamentos.

Mecânica de jogo muitas vezes motivam as pessoas, fornecendo *feedback* positivo, tais como o acúmulo de pontos, obtendo medalhas, maior visibilidade do estado, reconhecimento do progresso, personalização, surpresas agradáveis etc (DORLING; MCCAFFERY, 2012). Na teoria, *feedback* negativo também pode ser útil, mas é menos eficaz na prática. Mecanismos de *feedback* negativo podem levar ao abandono completo da atividade *gamified*, a não ser que os usuários estejam extremamente motivados, ou usado em um contexto social (FADEL et al., 2014). As mecânicas, ainda que sem representação visual direta, fazem a transição entre a experiência do usuário e o detalhamento da interação homem-computador. Elas são menos abstratas e servem como guias para os designers de jogos incluírem esses elementos nos sistemas de informação. As principais mecânicas são as seguintes (WERBACH, 2012):

- **Desafios:** objetivos definidos para o jogador alcançar.
- **Sorte:** nem tudo é definido pela habilidade. Um resultado aleatório pode estar associado a uma ação do jogador e criar uma sensação de surpresa e incerteza.
- **Cooperação e competição:** ambas tratam da sensação de vencer e perder. Os jogadores podem atuar em conjunto ou uns contra os outros.
- **Retorno (*feedback*):** permite que os jogadores verifiquem como estão avançando no jogo. Apoia a sensação de progressão e diz se o jogador está fazendo as coisas certas para vencer no jogo.
- **Aquisição de recursos:** o jogador pode coletar itens para auxiliá-lo a vencer o jogo. Algumas vezes os itens podem ser coletados em uma ordem específica.
- **Recompensas:** é um benefício oferecido ao jogador por atingir um marco.

- **Transações:** indica aquisição ou venda de algo. Podem ser realizadas entre jogadores ou diretamente com o jogo.
- **Turnos:** cada jogador tem o momento e a oportunidade corretos para realizar suas ações.
- **Estado de vitória:** condições que, uma vez preenchidas, indicam que houve um vencedor no jogo.

2.3.3. Elementos de *design* de jogos

Elementos de *design* de jogos são elementos específicos que podem ser vistos e utilizados na interface do jogo. É o objeto mais concreto presente em um jogo e normalmente é o que vem primeiro à mente. Os componentes são elementos claros da interação homem-computador e se tratam comumente de peças visuais na elaboração de uma interface (GALEGALE, 2014).

Existem diversos elementos de *design* de jogos que são utilizados em *gamification*. Pedreira et al. (2015) fez um mapeamento sistemático analisando estudos primários, abordando os principais elementos utilizados, sendo eles: **Sistema de Pontuação, Medalhas, Rank, Barras de progresso, Votação, Metáfora Visual e Quests**. Além do mapeamento, outros estudos foram utilizados para entender como são aplicados os elementos *design* de jogos. (DORLING; MCCAFFERY, 2012; SINGER; SCHNEIDER, 2012; VASILESCU, 2014; SNIPES et al., 2014; SHETH et al., 2012; DUBOIS; TAMBURRELLI, 2013; DICHEVA et al., 2015; WERBACH, 2012). A seguir é apresentado alguns elementos de *design* de jogos (DICHEVA et al., 2015; WERBACH, 2012):

- **Pontos:** significa que as ações realizadas possuem um significado desejado e estão alinhadas ao que se espera que o jogador faça.
- **Medalhas :** são dadas para realizações especiais, ou seja, uma representação visual de um atingimento dentro do jogo.
- **Rank:** listam os jogadores e seus pontos de forma ordenada, elas refletem seu desempenho em comparação com outros usuários.
- **Níveis:** representação da capacidade do jogador dentro do jogo. Aumenta com o decorrer do jogo e com a evolução do engajamento do jogador.
- **Barra de progresso:** oferece um percentual de representação gráfica do progresso dos jogadores. A barra de progresso exhibe, por exemplo, quantas atividades o usuário já realizou e o quantas ainda faltam para ele aumentar seu nível.
- **Presentear:** possibilidade de troca gratuita entre os jogadores dentro do sistema.
- **Quests:** uma tarefa com uma descrição e um objetivo bem definido que o jogador deve realizar.
- **Votação:** o jogador pode avaliar uma tarefa feita por outro jogador.

Na Figura 2.2 é apresentada a interface do Stack Overflow que utiliza alguns dos elementos citados acima, pontos, *feedback*, medalhas (*badges*) e outros.

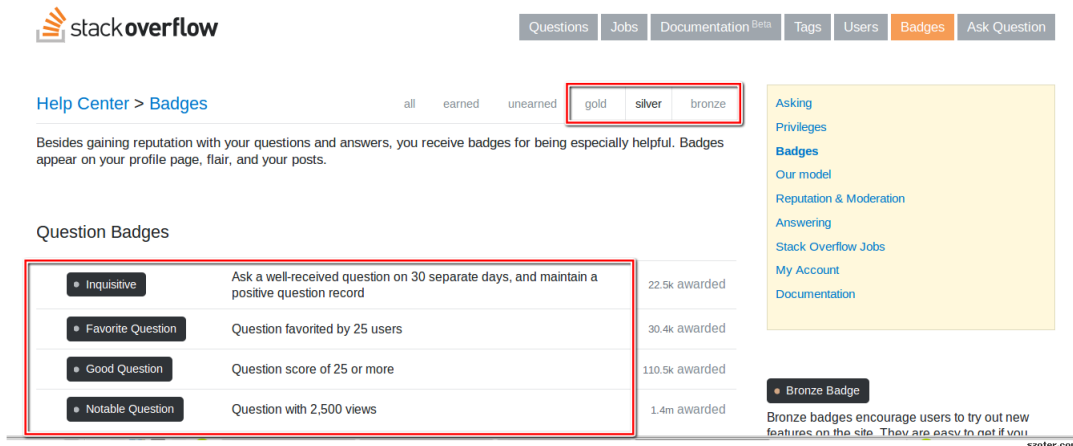


Figura 2.2. Aplicação dos elementos de *gamification*

Fonte: autoria própria

Os elementos de *design* de jogos utilizados pelo Stack Overflow foram destacados com um retângulo vermelho. O primeiro destaque é o retângulo menor, onde são apresentados os níveis das perguntas e resposta, os níveis foram rotulados de **gold**, **silver** e **bronze**. No retângulo maior é apresentado o mecanismo de como você consegue as medalhas (*badges*), por exemplo: para conseguir a medalha **Good Question**, do tipo **silver**, a pergunta do usuário terá que ter uma pontuação de 25 ou mais; um outro exemplo, para conseguir a medalha **Notable Question** do nível **silver**, a pergunta do usuário terá que ter mais de 2500 visualizações.

2.3.4. Regras

O próximo elemento-chave é a correta definição de um sistema de recompensas e penalidades, ou seja, definição das regras em geral. As recompensas representam a principal razão pela qual os jogadores se motivam a persistir em um jogo até sua conclusão, ou mesmo permanecerem jogando, nos casos em que a proposta não contempla o conceito de finitude (VIANNA et al., 2013). Para que esse objetivo seja atingido, ao criar-se um jogo, é preciso considerar a necessidade de desenvolver modelos de *feedback* que sejam frequentes, porque, na prática, traduzem-se como um dos mais eficientes caminhos para assegurar a manutenção do engajamento ao curso do desempenho da atividade (WERBACH, 2012).

As regras são uma das qualidades essenciais de jogos: cada jogo tem um conjunto de regras. Por outro lado, cada conjunto de regras define um jogo. As regras são a estrutura formal de um jogo, o conjunto fixo de diretrizes abstratas, descrevendo como é o seu funcionamento (ZIMMERMAN, 2004). Segundo Prensky (2001) as regras são o que diferenciam os jogos um do outro. Provavelmente a definição mais básica de um jogo é que

ele é um jogo organizado, isto é, baseado em regras. As regras impõem limites, elas nos forçam a tomar caminhos específicos para atingir as metas e garantir que todos os jogadores assumam os mesmos caminhos. Elas nos colocam dentro do mundo do jogo, deixando-nos saber o que está dentro e fora de limites.

Na definição das regras, é importante evitar dois extremos. Se as regras do jogo forem pouco restritivas, o jogo irá ser monótono e ganhar vai ser muito fácil. Por outro lado, as regras podem ser muito restritivas, de forma que o jogo se torna muito difícil ou impossível (SUITS, 1978). Nas definições das regras é importante manter o equilíbrio entre o difícil e o fácil, de forma a guiar os jogadores, fazendo-os explorarem possibilidades, antes desconhecidas, para atingir o objetivo final. Assim, liberando a criatividade, estimulando o pensamento estratégico e os mantendo engajado.

2.4. Considerações Finais

O objetivo geral desse trabalho é motivar alunos do curso de computação a contribuírem com código em projetos de software livre. Para tal tarefa será utilizado a técnica de *gamificação* para ajudar esses alunos a superarem as barreiras encontradas e mantê-los engajados. Nesse sentido, foi realizada uma revisão a fim de entender o que é *gamification*, quais as principais áreas de pesquisas ela está sendo aplicada. Também realizamos uma revisão sobre as principais barreiras encontradas por novatos quando estes tentam contribuir com projetos de software livre.

A *gamification* tem muitas características positivas: o fator desafio, aumento no engajamento dos usuários, competição entre os jogadores etc. Apesar de ela motivar vários usuários, muitos tipos de jogadores não são alcançados devido não gostar de nenhum tipo de elementos de *design* de jogos. Por isso é importante elaborar regras bem flexíveis, de forma a não forçar usuários a utilizar ou fazer alguma atividade contra sua vontade. Isso é um desafio a ser superado, motivar o público alvo determinado nessa pesquisa e não desmotivar o público que não for alcançado com a técnica de *gamification*.

Método de Pesquisa

Neste capítulo é apresentada a proposta do método que está sendo empregado no desenvolvimento deste trabalho. A Figura 3.1 representa a abordagem proposta juntamente com procedimentos a serem realizados.

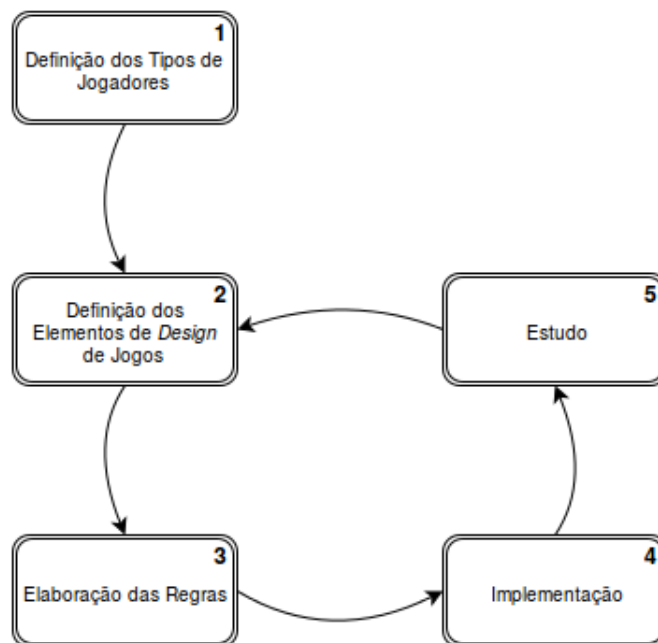


Figura 3.1. Visão geral do projeto

As etapas estão enumeradas de 1 a 5. Na etapa 1 “Definição dos Perfis de Jogadores” é apresentado o perfil de usuários com o qual será trabalhado neste trabalho. Na etapa 2 “Definição dos Elementos de *Design* de Jogos” são apresentados os elementos *design* de jogos que são utilizados neste estudo. Na etapa 3 “Elaboração das Regras” são apresentadas as regras que guiarão o ambiente *gamified*, por exemplo, as regras da pontuação, nível e etc. Na etapa 4 “Implementação” são apresentadas as interfaces com a *gamification*, modelo entidade

relacionamento entre outras atividades que foram implementadas ao longo deste estudo. Na etapa 5 “Estudo” é apresentado os resultados de um estudo preliminar realizado após o uso do ambiente *gamification* implementado na etapa 4. Na seta da etapa 5 para a etapa 2 significa que, como esta pesquisa conta com um método iterativo, nas próximas iterações do método pode-se descartar, aprimorar e experimentar outros elementos.

As etapas 1, 2 e 3 são apresentadas neste Capítulo. As etapas de 4 e 5 são apresentados mais adiante, cada uma em um Capítulo diferente pois são etapas mais longas e mais detalhadas.

3.1. Definição dos Perfis de Jogadores

Conforme mostrado na Figura 3.1, o primeiro passo realizado no trabalho foi definir os perfis de jogadores. Como existem muitos perfis diferentes, é essencial definir o perfil com o qual trabalhar, já que não é viável tratar todos os perfis possíveis em um trabalho inicial. Assim é possível prosseguir com um escopo mais bem definido.

Os perfis de jogadores podem variar de acordo com as características do público e também em qual contexto será aplicado a *gamification*. Podemos analisar diversas características dos potenciais jogadores. No contexto desta pesquisa, que são alunos, é possível observar, por exemplo, a idade, o gênero, a experiência em desenvolvimento de software e outras características.

Na revisão bibliográfica foram encontrados estudos que mostram diferentes perfis de jogadores. Esses perfis são variados e podem ser classificados por diversas características. O que ocorre é que cada público (ou no caso do estudo o grupo de alunos) tem diversos perfis e, para fazer a seleção dos perfis ideais seria necessário realizar um estudo mais aprofundado e direcionado somente na escolha desses perfis. Devido ao prazo não foi possível realizar o estudo de seleção de perfis. Assim, a seleção do perfis foi em relação as características baseadas nas habilidades técnicas dos alunos.

Sendo assim os perfis selecionados foram alunos que tinham conhecimento sobre desenvolvimento de software. Os Alunos que possuem conhecimento em desenvolvimento de software, são os alunos que tinham passado por um conjunto de disciplinas que possibilitem que participem de projeto de software. Sendo assim, focou-se em alunos que estivessem pelo menos no 6º período do curso de Bacharelado em Ciência da Computação.

Esse perfil abrange a maior parte do nosso público (senão todo), ou seja, é o perfil mais recorrente em nosso contexto. É importante selecionar perfis que cubram uma alta gama de jogadores, pois é com base nos perfis que os elementos de *design* de jogos são selecionados e cada um desses elementos age de diferentes maneiras, podendo motivar ou desmotivar os jogadores.

3.2. Definição dos Elementos de *Design* de Jogos

Identificados os tipos de jogadores e selecionados os perfis com o qual trabalhar, é necessário identificar os elementos de *design* de jogos no qual será utilizado para motivar esses jogadores. Em um ambiente pode-se introduzir diversos elementos de jogos para motivar e engajar contribuidores. Contudo, existem elementos que não surtem efeitos em muitos casos, ou até mesmo surtem efeitos contrários ao objetivo do estudo, intimidando e desmotivando novos contribuidores.

Um dos principais problemas é quando elementos não são adaptados aos tipos de jogadores. Ou seja, existem alunos que são mais exploradores, gostam de descobrir o que existe no ambiente. Existem aqueles que são mais sociais, que preferem interagir e fazer amigos. E há aqueles que são mais competitivos, que se importam muito quanto ao seu *status* frente aos colegas. Para cada um desses perfis, são necessárias diferentes estratégias de personalização da *gamification*, diferentes elementos de *design* de jogos.

Baseado nos perfis selecionados e nos mapeamentos sistemáticos de aplicação de *gamification* na engenharia de software e na educação de Dorling e McCaffery (2012), Pedreira et al. (2015), nós escolhemos os elementos que mais se adequam ao contexto e que são mais utilizados nesses estudos.

Os elementos de *design* de jogos selecionados foram:

- **rank:** permitem que os usuários acompanhem o seu desempenho em relação aos outros participantes, eles geram concorrência entre os jogadores e pode ser dividida em várias subcategorias, tais como: global, amigos, regional, ou, em nosso caso, por projetos e etc. O *rank* é uma ordenação do maior para o menor, de acordo com uma métrica estabelecida (quantidade total de pontos, *quests* resolvidas e etc). Segundo (DETERDING et al., 2011), os *ranks* representam uma das técnicas mais comuns usadas para criar *gamification*. O *rank* em ambiente de contribuição para SL, servirá para exibir os usuários que possuem mais pontos (como os pontos são ganhos de com as atividades realizadas, o *rank* mostrará as pessoas que estão mais ativas no projeto, que mais contribuíram). Ter o seu usuário em destaque é vantajoso, pois serve como uma forma de divulgar o seu trabalho, visto que empresas analisam o histórico de usuários antes de contratá-los, ou mesmo ganhar *status* dentro da comunidade.
- **ponto:** é o elemento mais encontrado nos estudos, ele é responsável por fornecer *feedback* e mostrar o progresso do jogador (LOMBRISER, 2014). Significa que as ações realizadas possuem um significado desejado e estão alinhadas ao que se espera que o jogador faça. Os pontos podem ser adquiridos através de diferentes ações do usuário. Por exemplo, pensando em uma contribuição de SL, um usuário pode ganhar pontos através de um *pull-request* aceito, ao entrar em contato com a comunidade fazendo comentários construtivos, ou resolver uma *quest* e etc.

- **nível:** agrupamento de diferentes tipos de desafios, normalmente de maior dificuldade conforme a progressão. Ajuda a tornar as tarefas mais desafiadoras, o usuário ganha novas competências e experiências (LOMBRISER, 2014). Csikszentmihaly (2000) descreve a teoria do “*flow*” de forma a manter o nível de desafio em alinhamento com as habilidades do usuário. Se o nível é muito difícil, o jogador vai se sentir ansioso e perder o interesse no jogo. Por outro lado, se o jogo é muito fácil, o jogador vai ficar entediado e parar de jogar. Portanto, o equilíbrio dessas duas dimensões é importante para manter os jogadores envolvidos. Em uma contribuição com SL, o nível servirá, por exemplo, na escolha do *bug* a ser resolvido. O usuário poderá comparar o nível que se encontra e poderá escolher um *bug* que esteja associado ao seu nível. Lembrando que os *bugs* também possuem níveis.
- **quest:** é uma atribuição ou uma tarefa com um objetivo bem claro que deve ser feito pelo usuário. As recompensas podem existir sob a forma de itens, reconhecimento, pontuação extra e são apresentados ao leitor quando uma determinada tarefa é realizada (LOMBRISER, 2014). O elemento nível é bem atrelado às *quests*, a cada nível alcançado as tarefas são mais desafiadoras, o usuário ganha novas competências e experiências. Em uma contribuição com SL, as *quests* servem para guiar o usuário a fazer a contribuição. Ela servirá como um passo a passo, onde o usuário seguirá uma sequência de atividade até conseguir completar um objetivo (por exemplo, fazer uma contribuição).

Uma limitação em relação a nossa pesquisa é o fato de não poder trabalhar com muitos elementos de *design* de jogos, devido ao tempo destinado para realizar o estudo. A pesquisa será feita em cima de alguns elementos de *design* de jogos, porém pode haver outros elementos importantes, ou que podem ser mais eficazes no contexto e objetivo do nosso estudo. Porém, após finalizar a pesquisa, analisar e validar os dados, é possível selecionar diferentes tipos de elementos e então refazer os experimentos para obter novos resultados.

Na Seção 3.3 poderá ser visto como cada elemento se comportará, qual a relação entre eles e como eles serão utilizados para ajudar os jogadores a superarem as barreiras e mantê-los engajados.

3.3. Definição das Regras

As regras, quando reunidas, compõem um conjunto de disposições que condicionam a realização do jogo, a fim de promover equilíbrio entre um desafio passível de ser concluído sem, no entanto, ser fácil a ponto de desestimular sua resolução (VIANNA et al., 2013). As regras ajustam o nível de complexidade do jogador frente à atividade a ser desenvolvida, liberando a criatividade, estimulando o pensamento estratégico e mantendo os jogadores engajados. A definição das regras estão relacionadas ao conteúdo desenvolvido na fase de definições de tipos de jogadores e elementos de *design* de jogos. A Figura 3.2 apresenta como

os elementos estão relacionados entre si e como irão agir no sistema.

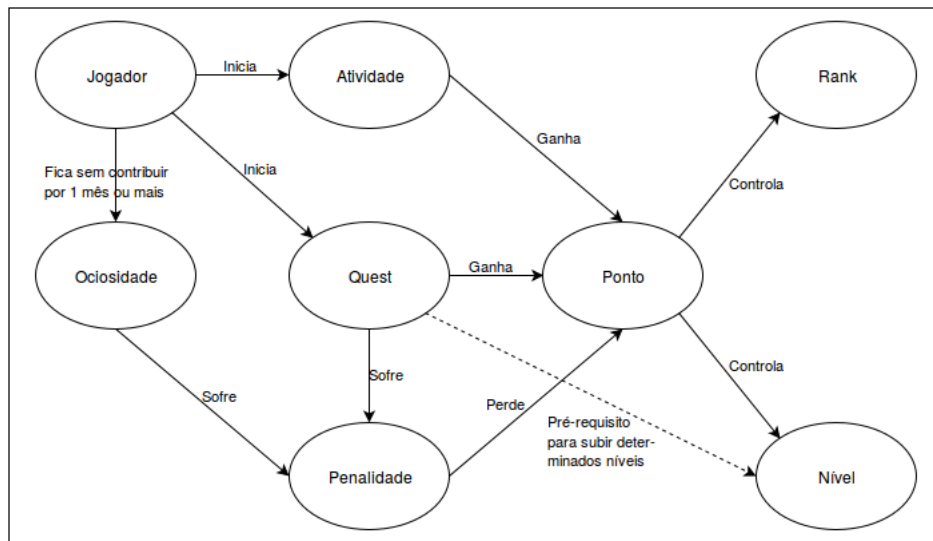


Figura 3.2. Diagrama de como os elementos irão agir no sistema

O jogador inicialmente poderá realizar três ações:

- A primeira ação é fazer uma atividade qualquer: cada uma dessas atividades irá valer uma quantidade de pontos. Para evitar que o jogador fique repetindo a tarefa várias vezes e acumule pontos fazendo a mesma tarefa e não progredindo, as pontuações dessas atividades terão um valor menor de pontos a cada vez que ela for resolvida.
- Outra ação que o jogador poderá realizar é selecionar uma *quest* para resolver. A cada nível será liberado um conjunto de *quests* para se resolver. Essas *quests* terão um objetivo claro e bem definido. As *quests* terão um valor maior de pontos do que quando é realizada uma atividade sem selecionar uma *quest*, incentivando o jogador a resolver as *quests* já que elas irão valer mais pontos. Dependendo da *quest*, elas terão penalidades, por exemplo: quando o jogador entrar em contato com a comunidade, o seu comentário poderá ser avaliado, caso tenha dez avaliações negativas, ele será penalizado. É muito importante que o jogador realize as *quests*, pois elas estão estruturadas de forma a ajudar os usuários, elas irão funcionar como um mapa para os novatos, guiando-os para que eles possam fazer contribuições.
- Pode ocorrer também do jogador ficar muito tempo sem fazer contribuições, ou seja, ficar ocioso. Quando isso ocorrer, será enviado um e-mail notificando o jogador. Se o jogador ficar sem contribuir por um período de 35 dias, serão descontados pontos do mesmo. Se o jogador não tiver pontos, será enviado um e-mail, notificando que ele está há muito tempo sem contribuir.

Os níveis terão rótulos e cada nível liberará um conjunto de *quests*, por exemplo:

- **Silver 1** - é o nível inicial, para passar de nível o jogador terá que acumular 100 pontos. O conjunto de *quests* liberado nesse nível é:

- Entrar em contato com a comunidade. Valor 50 pontos;
 - Obter ajuda para configurar o ambiente. Valor 20 pontos;
 - Configurar ambiente. Valor 50 pontos;
 - Obter ajuda para encontrar um bug. Valor 20 pontos.
- **Silver 2** - é o nível 2, para passar de nível o jogador terá que acumular 250 pontos. O conjunto de *quests* liberado nesse nível é:
 - Reproduzir o *bug*. Valor 75 pontos;
 - Validar um contato com uma pessoa experiente no projeto, para verificar o seu código. Valor 80 pontos;
 - *Pull-request* (mesmo que não seja aceito). Valor 60 pontos. Se for aceito terá outra pontuação.

O responsável por controlar o nível é o elemento pontos. Para subir de nível é necessário obter uma quantidade de pontos determinada naquele nível. Outro fator importante que pode ser observado na Figura 3.2 é a linha pontilhada que sai do elemento *quest* e vai para o elemento nível. Como nosso objetivo é motivar os alunos a contribuírem com código, é essencial que os jogadores tenham feito determinadas *quests*. Entretanto, não queria-se tornar todas as *quests* obrigatórias, pois alguns jogadores não gostam de *quests*. Tornar todas as *quests* compulsórias pode desmotivá-los. Para resolver esse problema, decidiu-se deixar algumas *quests* definidas como pré-requisitos para a troca dos níveis principais (por exemplo: mudança do nível de “Silver” para “Gold”).

Os pontos também são responsáveis por controlar o elemento *rank*. O *rank* poderá ser exibido em duas formas: por quantidade de pontos ou por *nível*. O *rank* será a representação gráfica ordenada do maior para o menor, de acordo com a métrica estabelecida no caso é a quantidade total de pontos e por nível. Os cinco primeiros nomes que constarem no *rank* estarão escritos em maiúsculo, como forma de destaque.

Para os *bugs*, as regras preliminares são: todos os *bugs* relatados nos projetos terão um nível que vai variar entre: Fácil, Médio, Difícil e Muito Difícil. Cada um desses *bugs* terá uma quantidade de pontos que irá variar de acordo com o nível atrelado ao mesmo. Exemplo: *bug* de nível fácil terá um valor inicial de 100 pontos, um *bug* de nível médio terá um valor inicial de 200 pontos, *bug* de nível difícil terá um valor inicial de 400 pontos e um *bug* de nível muito difícil terá um valor inicial de 700 pontos.

Vale ressaltar que os valores foram escolhidos apenas para explorar os elementos. É importante que sejam conduzidos estudos preliminares específicos para determinar os valores mais adequados de pontuação. Entretanto, tal análise não faz parte do escopo do presente trabalho.

Como o objetivo é contribuição com codificação, é necessária uma atenção maior às regras de *pull-requests*, pois a contribuição com código ocorre quando tem-se *pull-requests* aceitos.

- Pontuação do *Pull-request*
 - A pontuação do *pull-request* será de acordo com o nível do *bug* resolvido. Se resolvido de primeira, então valerá o valor total de pontos que valia o *bug*.
 - * Exemplo: o jogador faz um *pull-request* de um *bug* de nível fácil, que vale um total de 100 pontos e ele é aceito, então o jogador receberá os 100 pontos do *bug* resolvido.
 - A pontuação do *pull-request* pode ser alterada de acordo com a quantidade de *review* necessária, ou seja, de acordo com a quantidade de vezes que é enviado o *pull-request* para análise. A cada uma *review* será descontado 10% do valor total do *pull-request*. O *pull-request* terá um valor mínimo de 50% do seu valor inicial. Então se conclui que depois da quinta *review*, não serão mais abatidos os 10% e ele valerá só metade do valor inicial.
 - * Exemplo: o jogador faz um *pull-request* de um *bug* de nível fácil que vale um total de 100 pontos e ele não é aceito, então será abatido 10% do valor do *bug* ($100 - 10\% = 90$ pontos). Depois o jogador envia um novo *pull-request* do mesmo *bug* e este é aceito, então ele receberá 90 pontos, que é o valor inicial do *bug* menos 10% abatido na primeira *review* que não foi aceita. Isso é uma solução para o jogador não ficar fazendo vários *pull-requests* por tentativa e erro, e sim revisar o código e tentar resolver de fato o problema.
- Primeiro *pull-request*.
 - Será pontuado mesmo que não seja aceito, porém irá valer menos pontos. Se for aceito, irá receber a pontuação total do *bug* corrigido.

Algumas das regras foram explicadas no início do estudo prático, mais de forma geral os jogadores exploraram e descobriram como funcionam as regras conforme progressão no ambiente *gamified*.

3.4. Elementos de *design* de jogos x barreiras

Um dos objetivos deste trabalho é usar os elementos de *design* de jogos como forma de reduzir algumas das barreiras encontradas por novatos. Como apresentado no estudo de Steinmacher et al. (2015), existem diversas barreiras encontradas por novatos. Neste trabalho foi utilizado os elementos de *design* de jogos – *quest*, ponto e nível para tentar diminuir essas barreiras.

O elemento *quest* pode ser claramente utilizado como forma a reduzir a categoria de barreiras citada. As *quests* de uma forma geral foram construídas para orientar o usuário a fazer um série de tarefas para chegar ao seu objetivo final. Por exemplo, no presente trabalho, o objetivo é fazer uma contribuição com codificação. Dado esse objetivo o usuário terá que fazer um série de tarefas, como:

1. entrar em contato com a comunidade
2. criar um *fork* do projeto
3. configurar o software na sua máquina pessoal
4. escolher um *bug* que esteja associado ao seu nível
5. reproduzir o *bug* selecionado
6. fazer o *pull request* quando solucionado o *bug* selecionado

Com as tarefas acima, um novato saberá por onde começar, ou seja, ele terá um caminho (um passo a passo) para iniciar sua contribuição.

O elemento ponto oferece indicações que também visam reduzir a “Falta de orientação”, sendo uma ferramenta de *feedback* constante ao novato. Por exemplo, quando o novato executa uma tarefa, ele pode saber se esta foi feita de forma correta, então ele será bonificado com pontos, isso mostra para o usuário que ele de um passo adiante, que progrediu com relação ao seu objetivo.

O elemento nível também facilita o fato do usuário inicial se sentir perdido no início do projeto. O usuário inicia no nível 1, esse nível está ligado com as *quests* e nesse nível é liberado somente algumas tarefas para o usuário realizar, isso limita o espaço de busca do usuário guiando ele a começar de forma correta, de um nível iniciante. Os *bugs* também possuem níveis o que facilita na escolha do mesmo quando usuários forem solucioná-los.

É possível ainda que os elementos ponto e *ranks* amenizem ainda barreiras da categoria “Comportamento do novato”. Tal categoria que contempla, por exemplo, falta de proatividade e paciência do novato. Os pontos funcionam como um elemento motivador, uma vez que indicam o andamento do novato a cada tarefa e comparam-o com seus pares. Para novatos com perfil de competição, tais elementos podem despertar algumas características que o tornem mais proativo e mais comprometidos em completar as tarefas propostas.

Implementação

A fim de validar a proposta foi implementado em um software livre (página *web*) a técnica de *gamification*. Este capítulo apresenta as tecnologias utilizadas nessa implementação, os requisitos necessários e detalhes da implementação do sistema, modelo entidade relacionamento (ER), descrição das interfaces e a implementação dos elementos de *design* de jogos.

Todas as implementações e modificações realizadas no GitLab estão hospedadas em um repositório do GitHub ¹. O software é livre, sendo que qualquer pessoa pode copiar, utilizar, modificar e distribuir.

4.1. GitLab

O GitLab é um software no qual será desenvolvido e inserido os elementos de *design* de jogos, ou seja, é a página no qual serão feitas as implementações.

O GitLab é um gerenciador de repositório de software. Ele é similar ao GitHub ², porém ele permite que os desenvolvedores armazenem o código em seus próprios servidores, ao invés de armazenar em servidores operados pelo GitHub. Ele é distribuído como software livre, sob licença MIT ³ e é escrito em Ruby. O código do GitLab é bem organizado e a documentação é atualizada.

O GitLab utiliza o *framework* Ruby On Rails⁴. Ruby on Rails é um projeto de código aberto, escrito na linguagem de programação Ruby. As aplicações criadas utilizando o *framework* Rails são desenvolvidas com base no padrão de arquitetura MVC (*Model-View-Controller*).

¹ github.com/guilhermecd/gitlab-gamification

² github.com

³ opensource.org/licenses/mit-license

⁴ rubyonrails.org

Foram implementados os elementos de *design* de jogos definidos na Seção 3.2 e as Regras definidas na Seção 3.3.

4.2. Modelo de Dados

O modelo de dados contém o diagrama entidade relacionamento (ER) e a descrição das tabelas que foram adicionadas e modificadas conforme necessidade. O ER completo do GitLab encontra-se no github junto ao código fonte. Na Figura 4.1 é apresentado o ER das tabelas que sofreram modificações e/ou foram criadas.

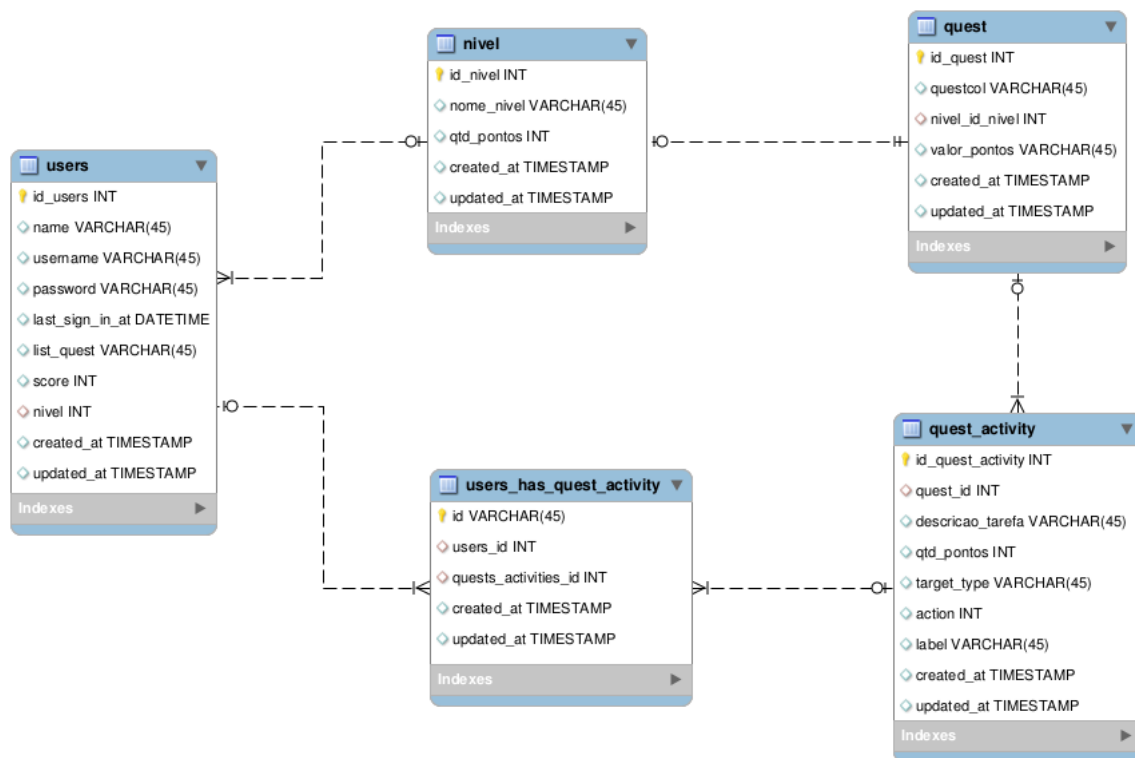


Figura 4.1. Modelo de Entidade Relacionamento das tabelas que foram criadas e/ou sofreram modificação

Fonte: autoria própria

A tabela “users” existe no modelo original do GitLab. Nela foram adicionados os atributos *score* e *nivel*, onde *score* representa a quantidade total de pontos do usuário e tem como valor *default* 0. O atributo *nivel* representa o nível em que o jogador se encontra. Por *default* todos usuários começam no nível 1.

A tabela “nível” não existia no modelo original e foi criada para suportar os elementos de jogos. Ao criar as tabelas utilizando o *framework* Ruby on Rails, três atributos são criados automaticamente, são eles: o *id_nivel*; *created_at*; e *updated_at* que representam, respectivamente, a chave primária da tabela, a data que o registro foi criado e a data da última atualização no registro. O atributo *nome_nivel* representa o rótulo que o nível recebe,

exemplo: silver1, silver2, gold e etc. O atributo *qtd_pontos* refere-se à quantidade de pontos que é necessária para atingir determinado nível.

A tabela “quest” também não existia no modelo original do GitLab e foi criada para suportar os elementos propostos. A função desta tabela é representar as *quests* que existem no sistema. Além dos atributos gerados automaticamente pelo *framework*, ela também possui o atributo *nome_quest*, que define qual o nome da *quest*; o atributo *niveis_id*, que representa a que nível aquela *quest* está vinculada e o atributo *valor_pontos*, que define a quantidade total de pontos que é necessária para se completar determinada *quest*. Cada *quest* libera uma série de atividades, quando uma atividade é resolvida o usuário recebe os pontos referentes aquela atividade. Para completar uma *quest* o usuário deverá conseguir a quantidade de pontos desejado na *quest*, assim, não é preciso resolver todas as atividades liberadas desde que o usuário consiga a quantidade de pontos necessário.

A tabela “quest_activity” também foi criada para apoiar os elementos de *gamification*. Essa tabela também possui os atributos gerados pelo *framework*, tendo como chave primária *id_quest_activity*. Essa tabela representa as atividades de cada *quest*. Uma *quest* possui uma ou mais atividades. O atributo *quests_id* representa a qual *quest* determinada atividade pertence. O atributo *descricao_tarefa* tem uma descrição detalhada de uma tarefa a se fazer, o atributo *qtd_pontos* define o valor em pontos de uma determinada atividade. Os atributos *target_type*, *action*, *label* são utilizados para identificar que tipo de atividade é, por exemplo: uma das atividades é “Abra uma *issue* e entre em contato com a comunidade. Quando for criar a *issue* utilize a label Quest1_1.”, assim os atributos *target_type*, *action* e *label* seriam respectivamente “Issue”, “1”, “Quest1_1”.

Por fim, a tabela “users_has_quest_activity” foi criada. Essa tabela tem a função de armazenar quais atividades um determinado usuário realizou. Além dos atributos gerados automaticamente pelo *framework*, ela possui dois atributos – *users_id*, *quests_activities_id* – esses atributos significam respectivamente o *id* do usuário e o *id* da atividade.

4.3. Descrição das Interfaces

As interfaces do sistema foram desenvolvidas utilizando-se Ruby on Rails, HAML, bootstrap e CSS . Para o estudo conduzido, ela foi instanciada em um servidor local da Universidade Tecnológica Federal do Paraná - Campus Campo Mourão. As Subseções a seguir descrevem cada uma das interfaces criadas.

4.3.1. Interface Principal

A primeira modificação, que está destacada com o número 1, é a pontuação e o nível em que o usuário se encontra. O *dashboard* é a parte da interface que fica visível na maior parte da aplicação e é de extrema importância que o usuário acompanhe seu desempenho a todo

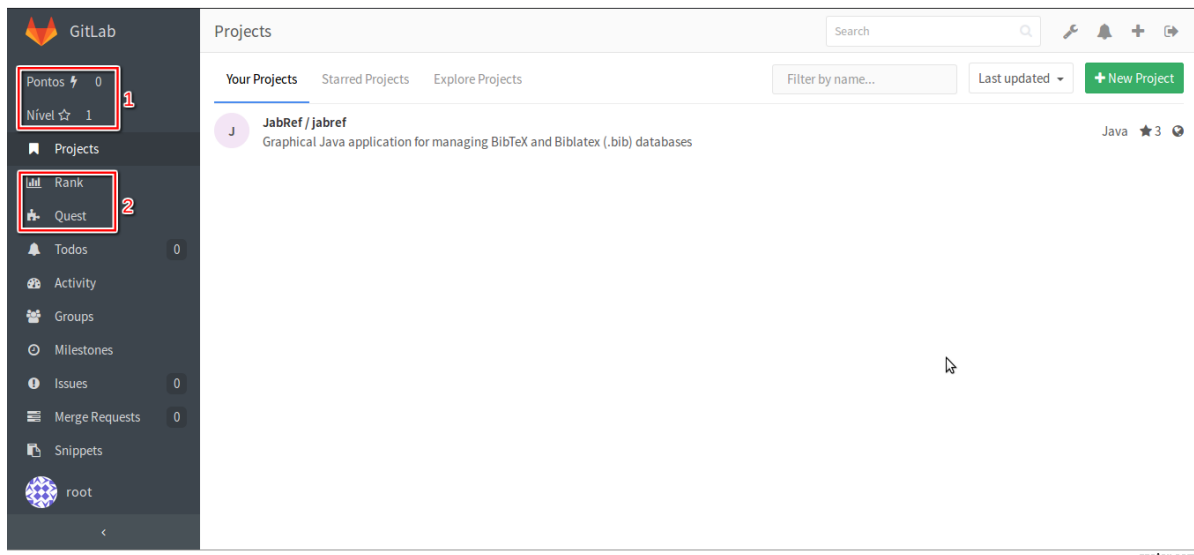


Figura 4.2. Interface principal do ambiente GitLab *gamified*
 Fonte: autoria própria

momento. Ao clicar no campo pontos, o usuário é redirecionado para a página de histórico de tarefas realizadas, e ao clicar no campo nível, o usuário é redirecionado para a página de exibição do *rank*.

O destaque com número 2 representa o *rank* e as *quests*. Ao clicar na opção *rank*, o usuário é redirecionado para uma página onde têm-se duas opção de exibição do *rank*, por ponto e nível. Ao clicar na opção de *quests*, o usuário é redirecionado para uma página onde são exibidas as *quests* disponíveis para o usuário.

4.3.2. Interface de Histórico de Tarefas Realizadas

A interface de histórico, apresentada na Figura 4.3, exibe quais atividades foram realizadas pelo usuário. Ela também pode ser acessada por outros usuários, de forma a exibir quais tarefas o outro competidor realizou.

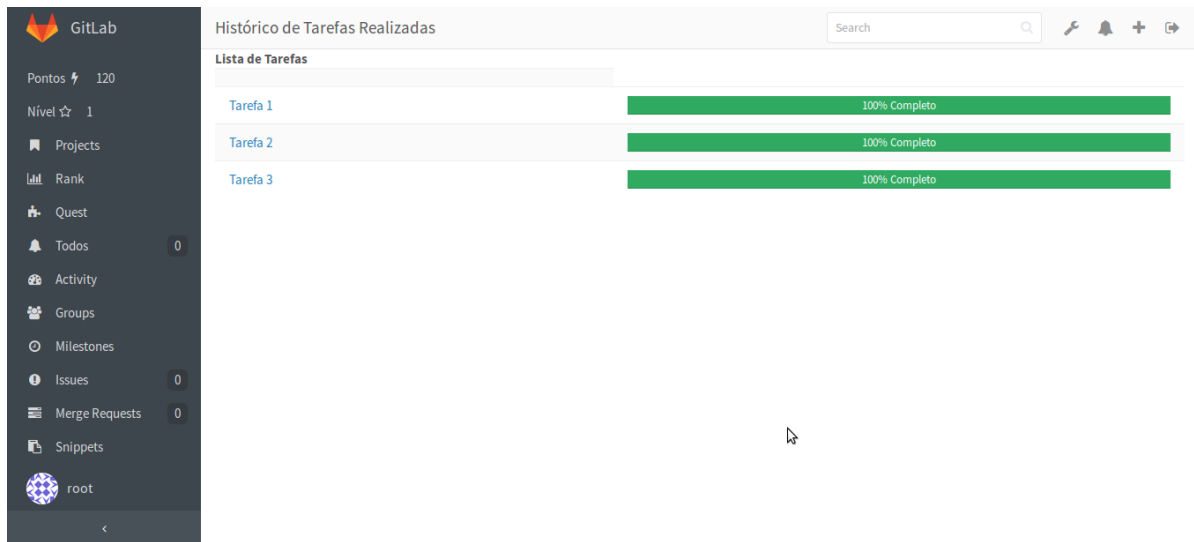


Figura 4.3. Interface referente ao histórico de tarefas realizadas pelo usuário
Fonte: autoria própria

Ao clicar nas tarefas, o usuário é redirecionado para a página de descrição da tarefa selecionada.

4.3.3. Interface do *rank*

A interface do *rank* pode ser exibida de duas formas: a primeira é por ponto (*score*) e a segunda é por nível. A forma de *exibição dos rankings pode variar, podendo ser apresentada por projetos, quantidade de quests realizadas e até mesmo temporal*. Para o escopo deste trabalho, por ter apenas um projeto para contribuição, limitou-se a duas formas de exibição (pontos e nível), apresentadas nas Figuras 4.4 e 4.5.

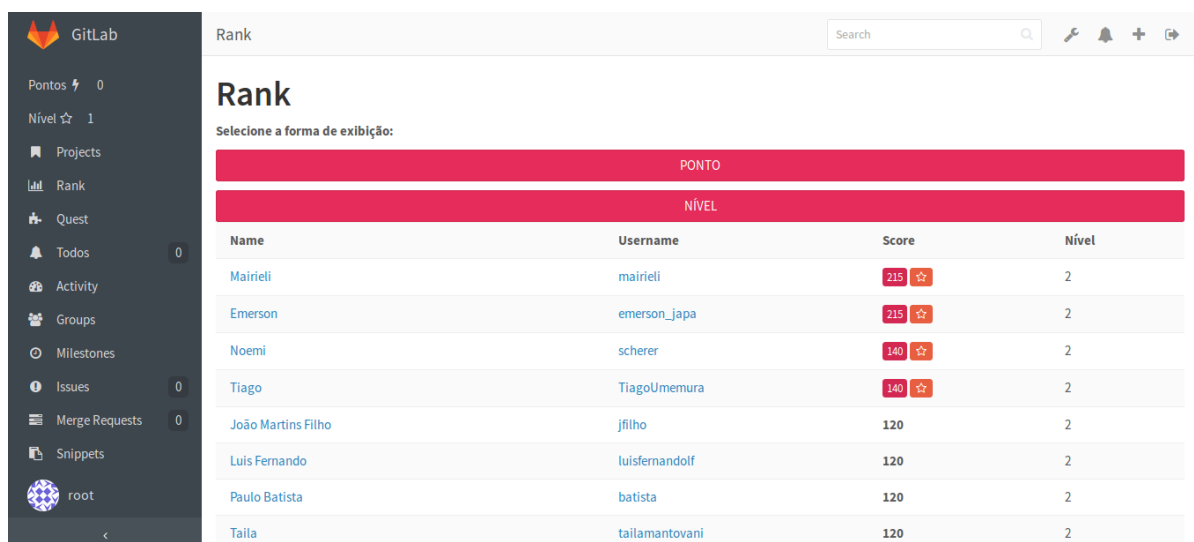
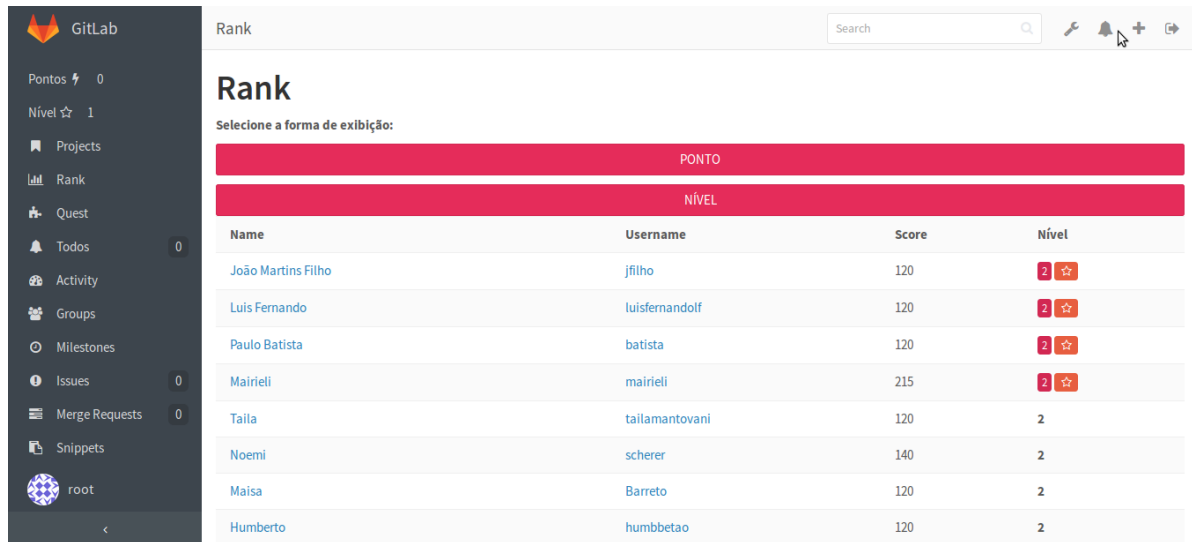


Figura 4.4. Interface do *rank*, com a opção de exibição por pontos.
Fonte: autoria própria

Na Figura 4.4 é apresentado o *rank* com a opção pontos selecionada. Na página é apresentado o *rank* com os campos *name* (nome de cada usuário), o *username*, *score* (pontuação de cada usuário) e o nível (nível de cada usuário). O campo *score* é apresentado em forma decrescente, ou seja, do maior para o menor e utiliza-se a métrica de quantidade total de pontos na ordenação. Também são exibidos os quatro primeiros usuários em destaque com uma estrela na cor vermelha, indicando que os mesmos são os quatro mais bem colocados no *rank*.



The screenshot shows the GitLab Rank interface. The left sidebar contains navigation options: Pontos (0), Nível (1), Projects, Rank, Quest, Todos (0), Activity, Groups, Milestones, Issues (0), Merge Requests (0), and Snippets. The main content area is titled 'Rank' and includes a search bar and a dropdown menu for 'Selecione a forma de exibição:' with options for 'PONTO' and 'NÍVEL'. The 'NÍVEL' option is selected. Below this is a table with the following data:

Name	Username	Score	Nível
João Martins Filho	jfilho	120	2 ☆
Luis Fernando	luisfernandolf	120	2 ☆
Paulo Batista	batista	120	2 ☆
Mairieli	mairieli	215	2 ☆
Taila	tailamantovani	120	2
Noemi	scherer	140	2
Maisa	Barreto	120	2
Humberto	humbbetao	120	2

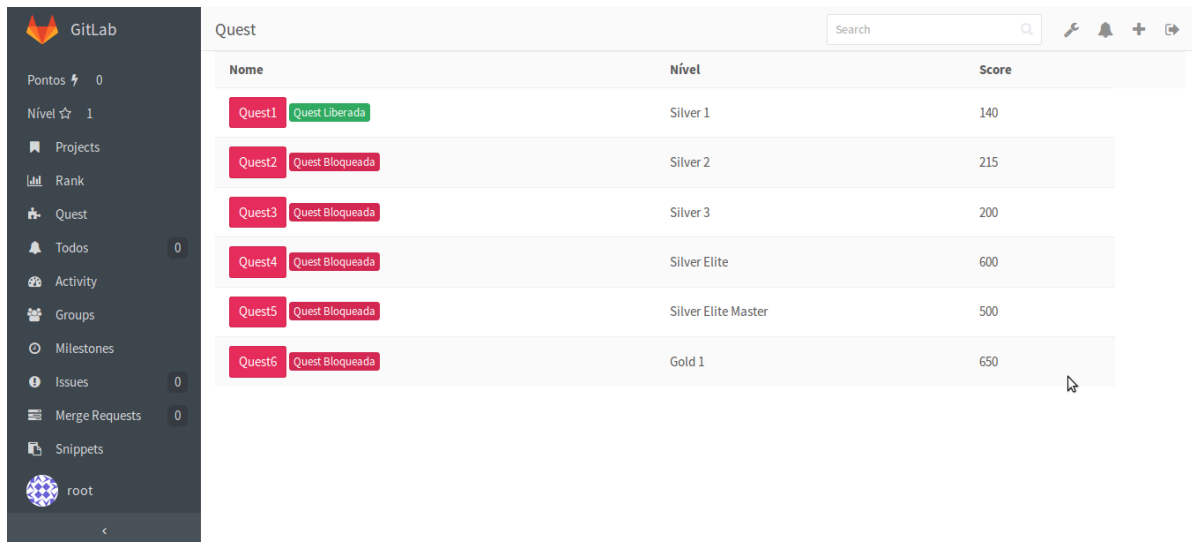
Figura 4.5. Interface do *rank*, com a opção de exibição por nível.

Fonte: autoria própria

Na Figura 4.5 é apresentado o *rank* com a opção nível selecionada. Essa interface apresenta os mesmos campos da interface com pontos, a mudança ocorre apenas no campo em destaque denominado nível. A lógica de exibições e destaques são as mesmas utilizadas na página com a opção pontos selecionada.

4.3.4. Interface das *quests*

A interface referente ao elemento de *design* de jogos *quest* pode ser vista na Figura 4.6. Essa interface exibe todas as *quests* existentes no GitLab.



Nome	Nível	Score
Quest1 Quest Liberada	Silver 1	140
Quest2 Quest Bloqueada	Silver 2	215
Quest3 Quest Bloqueada	Silver 3	200
Quest4 Quest Bloqueada	Silver Elite	600
Quest5 Quest Bloqueada	Silver Elite Master	500
Quest6 Quest Bloqueada	Gold 1	650

Figura 4.6. Interface do elemento de *design* de jogos *quest*

Fonte: autoria própria

O campo Nome é referente ao nome das *quests* e na sequência do nome há uma mensagem que informa o usuário se a *quest* está ou não liberada. Se liberada, ela irá exibir a mensagem na cor verde, escrito “quest Liberada” e se não estiver liberada, a mensagem estará na cor vermelha, escrito “quest Bloqueada”. Ao clicar no campo Nome, o usuário é redirecionado para a página de atividades, onde será apresentado um conjunto de atividades que formam determinada *quest*. O campo nível representa o nível em que o usuário deverá ter para realizar determinada *quest*. E por fim o campo Score, que representa a quantidade total de pontos para se realizar a *quest* por inteiro.

4.3.5. Interface das *Tarefas*

A interface das tarefas exibe um conjunto de atividades que juntas compõem uma *quest*. A Figura 4.7 demonstra como é a página das tarefas.

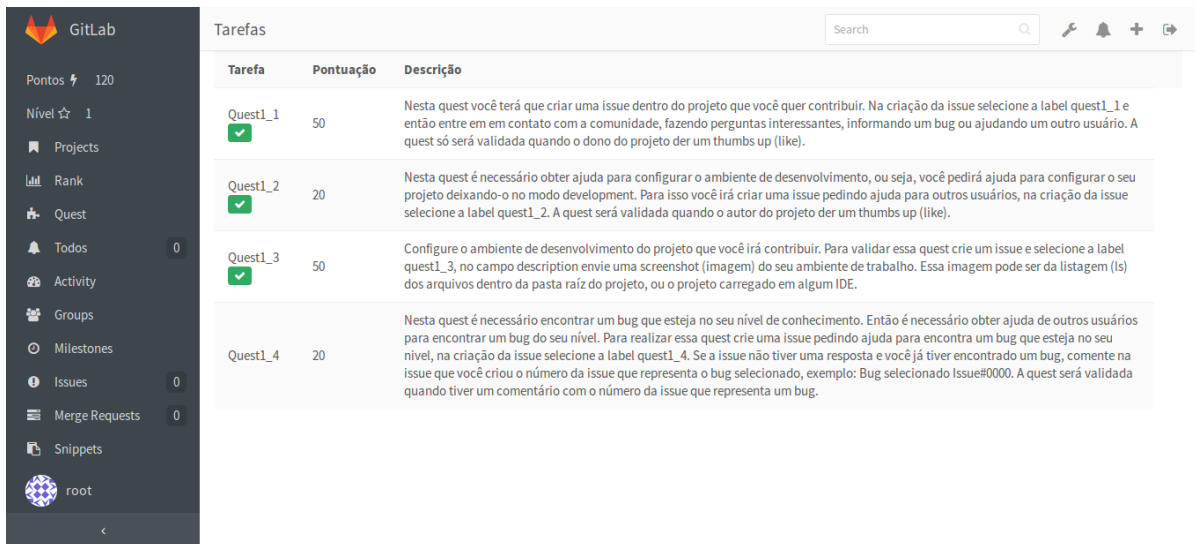


Figura 4.7. Interface do conjunto de tarefas que juntas formam uma *quest*

Fonte: autoria própria

Na interface das tarefas são exibidos os campos Tarefa onde há o nome da tarefa; Pontuação, que é referente à quantidade de pontos que o usuário recebe quando determinada tarefa é realizada e Descrição, que apresenta uma descrição detalhada de como a tarefa pode ser realizada. Quando a atividade é realizada, ela recebe um ícone de *check* na cor verde, que é inserido embaixo do nome da tarefa. Isso ajuda o usuário a não se perder e não correr o risco de realizar uma tarefa mais de uma vez.

4.3.6. Implementação das Regras

Um das etapas da implementação foi a criação das regras. Com as interfaces e o banco de dados funcionando, o último passo foi criar o mecanismo de pontuação do sistema. A implementação das regras não envolve as interfaces diretamente. Todas as ações realizadas nas interfaces são refletidas no banco de dados e vice-versa. Assim, o *script* que implementa as regras trabalha diretamente com banco de dados e os resultados são refletidos nas interfaces, fazendo as atualizações no *rank*, pontuando as atividades realizadas e liberando *quests*.

As regras de uma forma geral cuidam de como os elementos de *design* de jogos atuam no sistema. O que dificulta a implementação da mesma é a identificação e validação das atividades. Assim, para todas as atividades que o usuário pretendia realizar, era necessário antes utilizar uma *label* para identificação da mesma. A Figura 4.8 apresenta a interface onde é feita a solicitação para realizar uma determinada atividade.

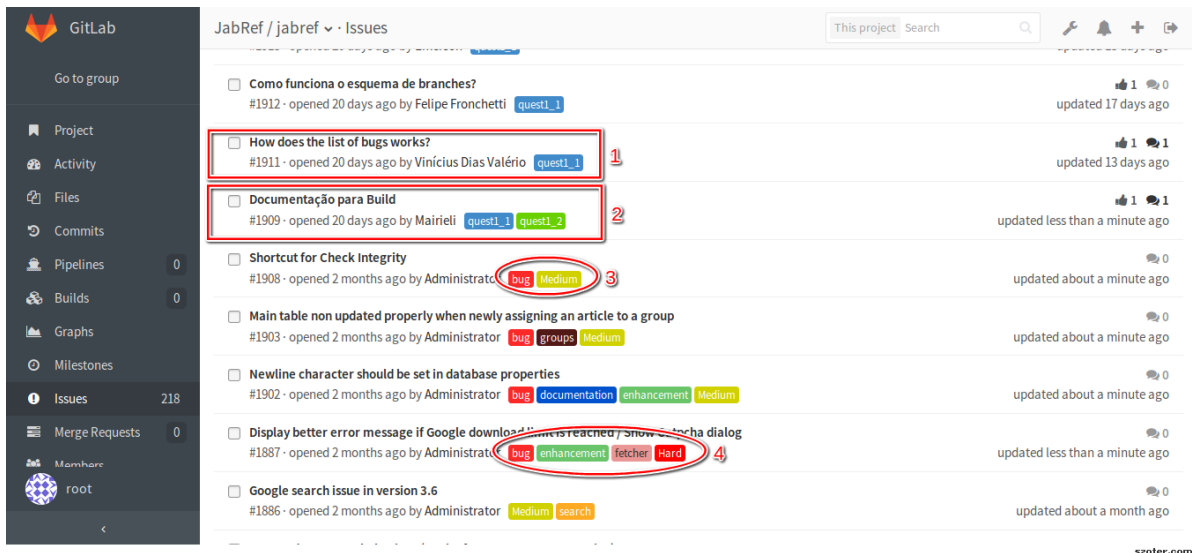


Figura 4.8. Ambiente onde é solicitada a realização de uma atividade

Fonte: autoria própria

A maioria das atividades foram realizadas por meio de *issues*. Por exemplo: na Figura 4.8 o retângulo vermelho de número 1 apresenta a solicitação para a realização da atividade de número 1, na qual o usuário cria uma *issue* e insere como *label* “quest1_1”. O mesmo ocorre para o retângulo de número 2, porém nesse exemplo, o usuário utiliza uma *issue* para realizar duas atividades – “quest1_1 e quest1_2”. As atividades, de uma forma geral, eram validadas após o dono do projeto aprovar a atividade por meio de um *thumbs up* (*Like*) na *issue* de solicitação.

Outra estratégia utilizada foi a rotulagem dos *bugs* com níveis de dificuldade – *Easy*, *Medium* e *Hard*). Na Figura 4.8 os círculos de números 3 e 4 mostram como era realizada essa tarefa. Ao criar uma *issue*, o usuário insere uma *label* de identificação informando do que se trata a *issue*, por exemplo: se é um *bug* ou uma dúvida. Depois o usuário pode inserir também uma *label*, indicando o nível de dificuldade do *bug*. na Figura 4.8, por exemplo, as *issues* indicam que são referentes a um *bug*, onde a de número 3 é um *bug* de nível médio (*Medium*) e a de número 4 é um *bug* de nível difícil (*Hard*).

É importante esclarecer que as *labels* utilizadas para identificação das *quests* e nivelamento dos *bugs* são previamente criadas. Só podem criar *labels* no projeto, o dono ou outros mantenedores, os demais usuários só podem utilizá-las.

Estudo Preliminar e Resultados

Este capítulo descreve a execução do estudo preliminar com estudantes, coleta e análise dos dados e os resultados obtidos da abordagem proposta neste trabalho. Na Seção 5.1 são apresentadas as questões de pesquisa a serem respondidas durante o estudo. A Seção 5.2 é apresentado como foi realizado o estudo, os projetos utilizados e outras partes envolvidas e, por fim, na Seção 5 são explanados os resultados do estudo.

5.1. Questões de Pesquisa

Para guiar a avaliação das estratégias de *gamification*, definimos a seguintes questões de pesquisa a serem respondidas durante o estudo:

1. Os elementos de *design* de jogos - nível, *quest*, *rank* e pontos – motivaram os alunos na tentativa de fazer uma contribuição em projetos de software livre?
2. Os elementos de *design* de jogos – nível, *quest*, *rank* e pontos – ajudaram a reduzir as barreiras encontradas por novatos?
3. A técnica de *gamification* motivou os alunos na tentativa de fazer uma contribuição em projetos de software livre?

5.2. Estudo Preliminar

Nesta etapa foi definido o design do estudo, ou seja, como foi a condução do mesmo. Na pesquisa experimental é essencial controlar as variáveis que possam estar interferindo no estudo, para que o ambiente se torne o mais adequado possível, manipular certas condições e observar os efeitos produzidos (GIL, 1991).

A primeira parte do estudo foi selecionar o projeto a ser utilizado no estudo. O

projeto selecionado foi o JabRef¹, por ser um projeto consolidado, maduro e por ter como um de seus mantenedores o orientador deste trabalho. O JabRef é um gerenciador de referências baseado em bases de dados BibTex. Ele é desenvolvido em Java, é um software livre, que utiliza a licença MIT, está hospedado atualmente no GitHub.

O GitLab possui uma opção de importar projetos. Essa opção funciona como um *fork*, porém ela importa o projeto completo contendo o código, as *issues* com comentários e votação, *branches*, *commits*, etc. Feita essa importação, o estudo foi realizado com *bugs* reais (que ainda não tinham sido resolvidos). É importante ressaltar que a versão do JabRef é a 3.6 e a importação foi realizada em agosto de 2016. O estudo foi realizado na disciplina denominada “Software Livre” do curso de graduação Bacharelado em Ciência da Computação. O estudo prático teve duração de 15 dias, onde os alunos utilizaram o GitLab *gamified* e então tentaram contribuir com codificação para o projeto de software livre JabRef. O estudo prático contou com 17 participantes.

5.2.1. Coleta de dados

A fase de coleta e análise dos dados é de grande importância na elaboração da pesquisa científica. Portanto, é necessário manter alguns cuidados para que se possa garantir a fidedignidade dos resultados.

Ao final do estudo, os participantes foram convidados a responder um questionário. No início do questionário foram explicados aos participantes os objetivos da pesquisa, o procedimento, o anonimato dos dados e sua liberdade de interromper a qualquer momento a realização do questionário se desejasse desistir da sua participação. Era explicado, ainda, que não havia respostas certas ou erradas e que ele deveria apenas responder, de forma sincera, de acordo com a sua opinião.

A coleta de dados contou com um questionário estruturado, contendo questões abertas e questões objetivas, seguindo Escala Likert. O questionário completo encontra-se no apêndice A. A Escala Likert mede atitudes e comportamentos utilizando opções de resposta que variam de um extremo a outro (por exemplo, de nada provável para extremamente provável). Uma Escala Likert permite descobrir níveis de opinião. Todos os 17 participantes responderam ao questionário.

5.3. Resultados

Nesta seção serão explanados os resultados obtidos durante o estudo. Será apresentado as questões do questionário, seguido da apresentação de gráficos e tabelas e exibição dos resultados. Para facilitar o entendimento o presente Seção foi dividida de acordo com o questionário: na Subseção 5.3.1 será apresentado uma o resultados onde o objetivo é saber

¹ www.github.com/JabRef/jabref

se os usuários fizeram contribuição em projetos de software livre. Na Subseção 5.3.2 é apresentado uma avaliação separadamente de cada elemento de *design* de jogos. Na Subseção 5.3.3 são apresentados as avaliações dos elementos em relação as barreiras. A Subseção 5.3.4 apresenta os resultados de uma avaliação geral dos elementos. Por fim a Subseção 5.3.5 uma avaliação do ambiente *gamified*.

5.3.1. Contribuição com Projetos de Software Livre

As primeiras perguntas do questionário foram relacionadas à contribuição para projetos de software livre. Na Tabela 5.1 são apresentadas duas perguntas, em que o objetivo é entender o quanto ou até que ponto os participantes conseguiram chegar ao tentarem contribuir com projetos de software livre.

A primeira pergunta é em relação a uma contribuição de forma geral (com códigos, tradução, relatar um bug, etc). Os participantes, em sua maioria, tentaram contribuir com projetos de software livre (76,5%). A segunda pergunta é em relação a contribuição para projetos de software livre com codificação. A maioria dos participantes não contribuíram com codificação (64,7%).

Tabela 5.1. Perguntas relacionadas à contribuição em projetos de software livre

Perguntas	Sim	Não
Você já tentou contribuir alguma vez com projetos de software livre?	76,5%	23,5%
Você já contribuiu com codificação para algum software livre?	35,3%	64,7%
No estudo em sala de aula você conseguiu fazer uma contribuição com codificação?	11,7%	88,3%

O objetivo do estudo foi de fazer contribuições com codificação para projetos de software livre. Quando analisada a Tabela 5.1, a segunda pergunta apresenta que 35,3% dos estudantes fizeram contribuições com codificação para algum software livre. Por outro lado se analisar a terceira pergunta, relacionada ao presente estudo, apenas 11,7% dos estudantes conseguiram fazer uma contribuição com codificação. Isso pode dar-se ao tempo do utilizado para realizar o estudo que foi de 15 dias.

5.3.2. Avaliação Unitária dos Elementos de *Design* de Jogos

Na sequência, foi realizada uma avaliação de cada um dos elementos de design de jogos. A escala de Likert foi aplicada e o participante marcava a sentença conforme se enquadrava dentro da afirmação. Foi utilizada uma escala Likert de 5 pontos, com as opções Discordo Plenamente (1), Discordo (2), Neutro (3), Concordo (4) e Concordo Plenamente (5).

O número de afirmações utilizadas no questionário varia entre os elementos avaliados, sendo para o elemento *quest*, três afirmações; para o *rank*, duas afirmações; para o elemento ponto, três afirmações e para o elemento nível, três afirmações. Para facilitar a visualização e

entendimento, os gráficos são exibidos com cores e percentual das opções marcadas. As cores predominantemente verde são referentes às opções Concordo e Concordo Plenamente, as cores predominantemente vermelhas são referentes às opções Discordo e Discordo Plenamente e a opção Neutro possui cor cinza.

Para a análise de todos os elementos serão consideradas positivas as opções Concordo Plenamente e Concordo; e serão consideradas negativas as opções Discordo Plenamente e Discordo.

Avaliação do Elemento *quest*

A Figura 5.1 mostra as avaliações do elemento *quest*. Ao todo são três afirmações. A primeira afirmação “*O elemento quest me guiou a fazer a contribuição.*” teve como maioria a opção Neutro (46%), em sequência as opções mais marcadas foram Concordo e Concordo Plenamente (ambas com 24%) e como minoria, a opção Discordo (6%).

A segunda afirmação “*O elemento quest serviu como orientação para fazer a contribuição.*” teve como maioria a opção Concordo (35%), seguida da opção Concordo Plenamente (29%), Neutro (24%) e como minoria a opção Discordo (12%).

A terceira e última afirmação relacionada ao elemento *quest* “*O elemento quest me motivou a realizar as tarefas propostas e assim fazer a contribuição.*” obteve Concordo como a opção mais marcada (52%), em segundo a opção de maior percentual foi Concordo Plenamente (18%), depois em ordem decrescente as opções mais marcadas foram Neutro e Discordo Plenamente (12%) e como minoria, a opção Discordo (6%).

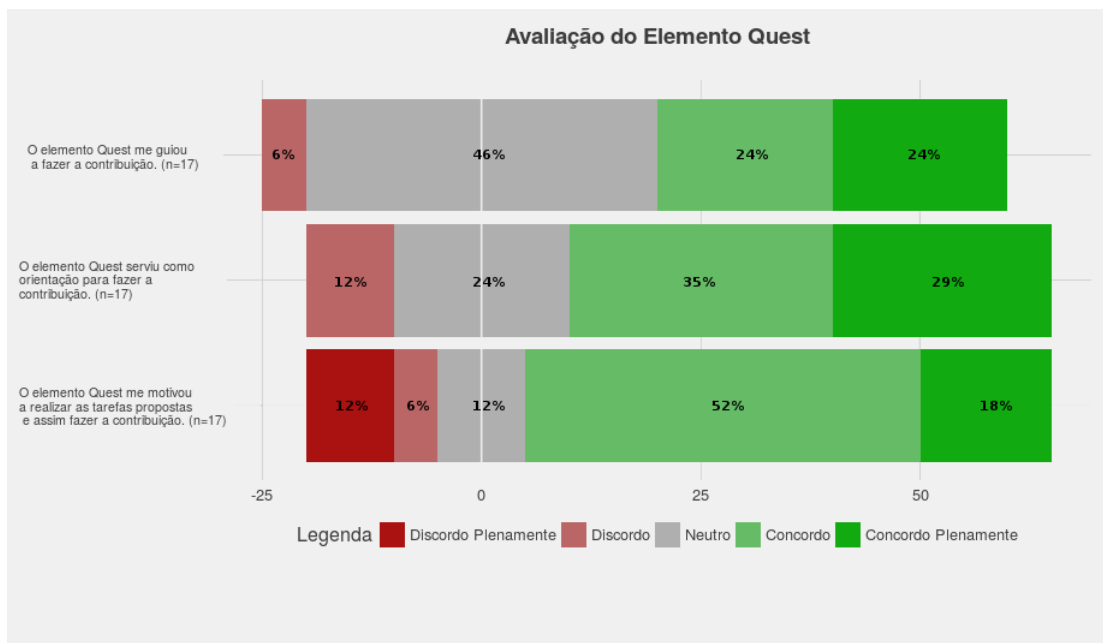


Figura 5.1. Avaliação do elemento de *design* de jogos *quest*

Fonte: autoria própria

Analisando as afirmações e as avaliações dos participantes, nas três afirmações as opções Concordo Plenamente e Concordo foram mais presentes que as avaliações Discordo Plenamente e Discordo. Apenas na primeira afirmação a opção Neutro obteve um percentual maior que Concordo e Concordo Plenamente, porém considerando que as opções Concordo e Concordo Plenamente são avaliações positivas para essa análise, na soma total dos percentuais dos mesmos, obtiveram um total de 48%, superando o percentual da opção Neutro. Nas afirmações em sequência, os resultados mostram que as opções mais marcadas foram Concordo e Concordo Plenamente, ou seja, avaliações positivas. Com isso, teve-se indícios de que o elemento *quest* motivou os alunos na tentativa de fazer contribuição e serviu como guia para os usuários novatos fazerem uma contribuição.

Avaliação do Elemento ponto

A Figura 5.2 apresenta as avaliações do elemento ponto. No total são três afirmações. A primeira afirmação “*Quanto mais pontos eu conseguia mais eu queria ter, a fim de aumentar meu nível e meu rank.*” obteve como maioria a opção Discordo (28%), seguida das opções Concordo Plenamente e Concordo (ambas com 24%), Neutro (18%) e Discordo Plenamente (6%).

A segunda afirmação “*O Elemento ponto me motivou a tentar fazer uma contribuição.*” teve como maioria a opção Concordo (28%) e seguindo em ordem decrescente, aparecem as opções Concordo Plenamente e Discordo (ambas com 24%), Neutro (18%) e Discordo Plenamente (6%).

A terceira e última afirmação “*O elemento ponto me proporcionou feedback das tarefas realizadas.*” não obteve avaliações Discordo e Discordo Plenamente e, obteve como maioria a opção Concordo Plenamente e Concordo (ambas com 35%) e Neutro (30%).

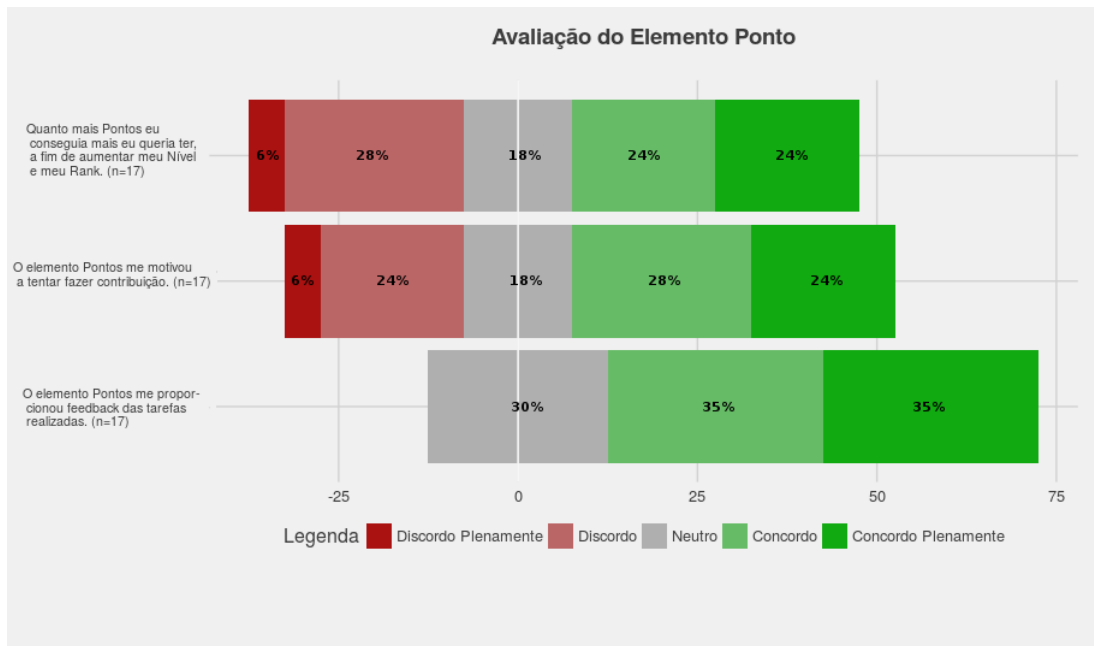


Figura 5.2. Avaliação do elemento de *design* de jogos ponto

Fonte: autoria própria

Analisando os gráficos como um todo, nota-se que o elemento ponto teve mais avaliações positivas. A terceira afirmação não obteve resultados negativos, porém obteve 30% de avaliações Neutras e, 70% das opções selecionadas foram para avaliações positivas. Com um percentual alto de avaliações positivas, há indícios de que o elemento ponto auxiliou na tarefa de fornecer *feedback* das atividades realizadas pelos usuários.

A primeira e a segunda afirmações obtiveram um total percentual maior de avaliações positivas, porém esse total ficou entre 48% e 52% o que não é um percentual alto para concluir que de fato o elemento ponto motivou os alunos na contribuição e gerou concorrência entre os usuários como afirma as sentenças.

Avaliação do Elemento nível

A Figura 5.3 apresenta as avaliações do elemento nível, no total são três afirmações. A primeira “*Eu consultava frequentemente o meu nível para comparar com outros usuários.*” teve como maioria a opção Discordo Plenamente (34%) em sequência a opção Neutro (24%), Concordo Plenamente e Discordo (ambas com 18%) e Concordo (6%).

A segunda afirmação “*Eu utilizei o elemento nível para escolher o bug, de acordo com meu nível eu selecionei um bug de dificuldade compatível.*” obteve como maioria a opção Discordo (28%), seguido da opção Neutro (24%), das opções Concordo Plenamente e Concordo (ambas com 18%) e Discordo Plenamente (12%).

A terceira afirmação “*Eu queria aumentar meu nível para liberar novas quests.*” não obteve avaliações Discordo e Discordo Plenamente e, obteve como maioria a opção Concordo

(53%) seguida de Concordo Plenamente (29%) e Neutro (18%).

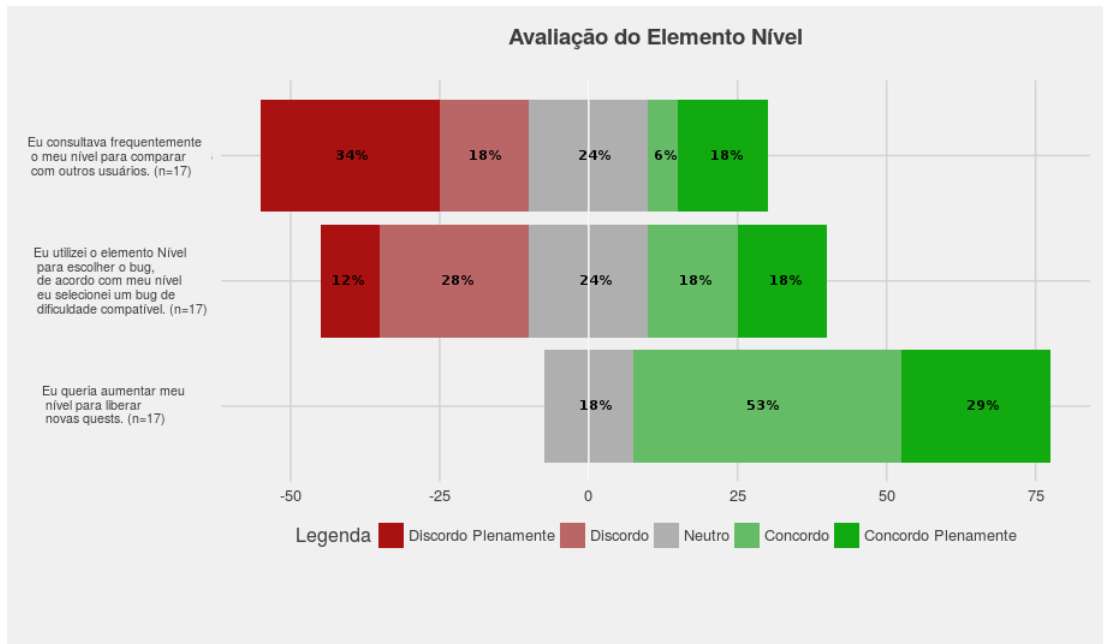


Figura 5.3. Avaliação do elemento de *design* de jogos nível

Fonte: autoria própria

A terceira afirmação não obteve resultados negativos, obteve 18% de avaliações Neutras e, 82% das opções marcadas foram avaliações positivas. Com um percentual alto de avaliações positivas, pode-se afirmar que o elemento ponto instigou a maioria dos usuários ao ponto de motivá-los a aumentar seu nível para liberar mais *quests*.

As demais afirmações obtiveram como maioria avaliações negativas, tanto no percentual individual quanto na soma total dos percentuais. Assim, há indicativos de que os *bugs* eram escolhidos na maioria das vezes sem consultar o nível em que o usuário se encontrava e também foi pouco utilizado como comparação entre os usuários.

Avaliação do Elemento *rank*

A Figura 5.4 apresenta as avaliações do elemento *rank*. No total são duas afirmações. A primeira “*Eu queria fazer mais quests e ganhar mais pontos para melhorar minha posição no rank.*” obteve como maioria a opção Concordo Plenamente (28%), seguida por Concordo e Neutro (ambas com 24%) e Discordo Plenamente e Discordo (ambas com 12%).

A segunda afirmação “*O elemento rank foi motivador, pois por ele eu conseguia ver meu desempenho com a relação a outros usuários.*” obteve como maioria duas opções, Concordo e Neutro (ambas com 29%), seguida da opção Concordo Plenamente (24%), Discordo (12%) e Discordo Plenamente (6%).

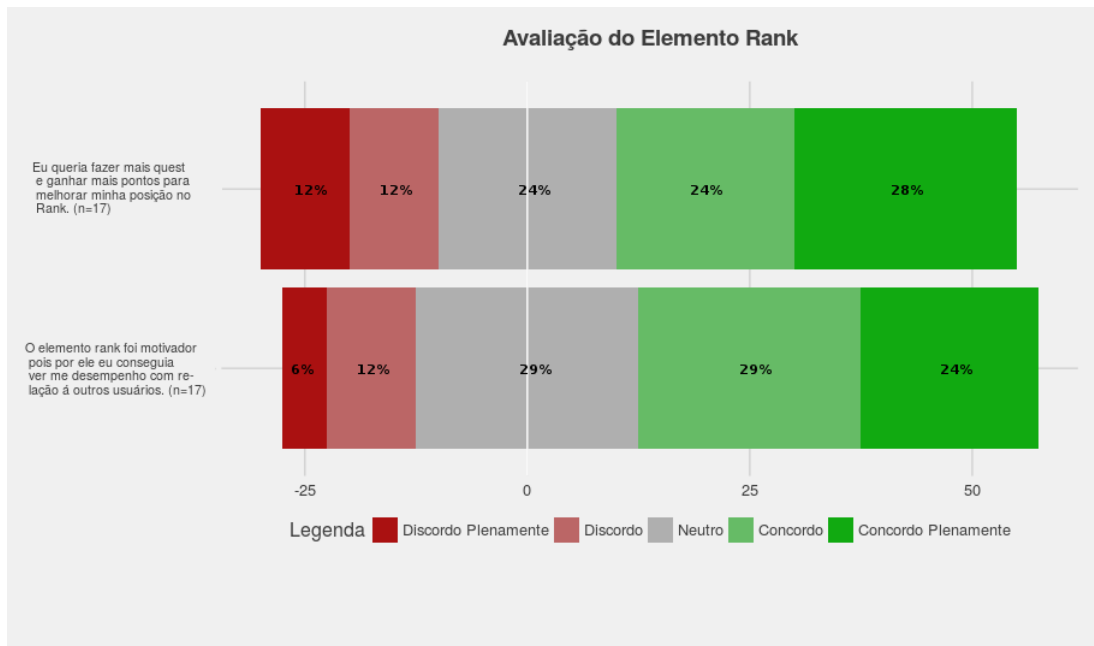


Figura 5.4. Avaliação do elemento de *design* de jogos *rank*

Fonte: autoria própria

Analisando os gráficos de uma forma geral percebe-se que o elemento *rank* obteve mais avaliações positivas. Porém tanto na primeira quanto na segunda afirmação o total percentual de avaliações positivas ficaram respectivamente com 52% e 53%, o que não é um percentual alto para concluir que de fato o elemento *rank* motivou os alunos na contribuição e gerou concorrência entre os usuários como afirma as sentenças.

Respondendo à questão de Pesquisa (1) - Os elementos de *design* de jogos - nível, *quest*, *rank* e pontos – motivaram os alunos na tentativa de fazer uma contribuição em projetos de software livre?

Nenhum dos elementos utilizados obteve resultados negativos em relação ao aspecto motivacional. É verdade, no entanto, que o elemento que se mostrou mais motivador e obteve os melhores resultados foi o elemento *quest*. Esse resultado pode ser atribuído ao fato das atividades das *quests* serem detalhadas e possuírem um objetivo claro, assim, ajudando e motivando os usuários. Na sequência, o elemento ponto também obteve resultados satisfatórios. O elemento ponto funciona como *feedback*, assim, quando o usuário faz um tarefa e ele recebe pontos, significa que o usuário está no caminho correto e isso o motiva a continuar. Os elementos *rank* e nível não tiveram resultados expressivos e não foi possível concluir que foram de fato motivadores, ambos tiveram avaliações medianas.

5.3.3. Avaliação dos Elementos de *Design* de Jogos x Barreiras

Nesta etapa foram feitas perguntas relacionando os elementos de *design* de jogos usados neste estudo com algumas das barreiras apresentadas no estudo de Steinmacher et al. (2015).

No total foram feitas quatro perguntas que relacionam as barreiras com os elementos. A cada pergunta o usuário avaliou cada um dos quatro elementos (*quest*, *rank*, ponto e nível), sendo que essa avaliação possui o intervalo de 1 até 10. Os dados são apresentados utilizando boxplot, onde é possível obter informação de um conjunto de dados, como: localização, dispersão, assimetria, comprimento da cauda e *outliers* (medidas discrepantes).

A Figura 5.5 apresenta o primeiro conjunto de boxplots, referentes à questão “Com relação a barreira Falta de orientação, que é o fato de usuários veteranos não ajudarem e também não ter um passo a passo de como contribuir. Em uma escala de 1 a 10, o quanto cada elemento ajudou a diminuir/quebrar essa barreira?”.

Na Figura 5.5 é possível observar que o boxplot referente ao elemento *quest* obteve um resultado melhor do que os demais elementos. A maioria das notas foram altas, a mediana ficou com 8, o que quer dizer que a metade acima da linha dentro da caixa deram nota 8 ou superior. Além disso, a distribuição mostra que, com exceção de um *outlier*, a nota mais baixa foi um 6.

O boxplot referente ao elemento ponto obteve resultados satisfatórios. Pode-se notar que a mediana ficou um pouco abaixo do que a mediana do elemento *quest*, mesmo assim uma mediana alta com valor 7. A distribuição também variou mais que a do elemento *quest*, tendo notas com valores 2 até 10, porém a maioria das votações ficaram entre 5 e 8, que são notas médias para altas.

Ainda na Figura 5.5, pode-se perceber pelos boxplots que os estudantes relataram que os elementos *rank* e nível não os ajudaram muito a superar a barreira relativa à orientação. As medianas foram respectivamente 4 e 5, o que mostra que metade dos usuários deram notas iguais ou inferiores a essas. A variação do elemento *nível* foi um pouco melhor que a do elemento *rank*, porém nada comparado com a variação do elemento *quest*.

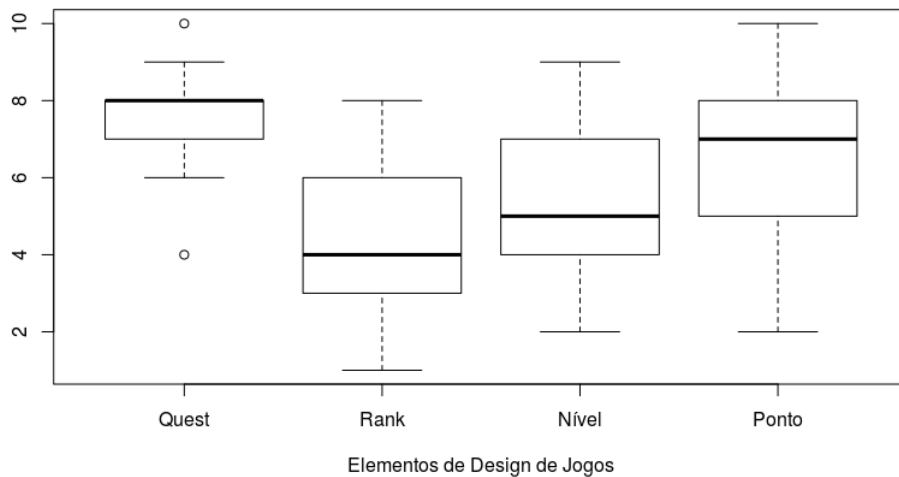


Figura 5.5. Conjunto de boxplot referente a avaliação dos elementos x barreiras - Barreira Falta de Orientação

Fonte: autoria própria

Assim, há indícios de que os elementos *quest* e *ponto* foram os que mais ajudaram os usuários a superar a barreira “Falta de orientação”. Os elementos *rank* e *nível* não tiveram boas avaliações, indicando que, de forma geral, os elementos apoiaram os estudantes de forma moderada.

A Figura 5.6 apresenta o segundo conjunto de boxplots, referente à questão “Com relação a barreira Falta de feedback, ou seja, quando o estudante executa uma ação e o mesmo não sabe se está correto ou não, se o caminho escolhido é o correto. Em uma escala de 1 a 10, o quanto cada elemento ajudou a diminuir/quebrar essa barreira?”.

Na Figura 5.6 o boxplot referente ao elemento *quest* também obteve as melhores avaliações. A mediana ficou com 7, o que quer dizer que metade das avaliações são iguais ou superiores a 7. Além disso, é possível notar que as avaliações ficaram entre 2 e 10, com 25% das respostas sendo abaixo ou iguais a 5. Isso mostra que, apesar de um indício de que o elemento se mostrou positivo, alguns estudantes consideraram que o elemento não os ajudou a superar a barreira falta de *feedback*.

Os boxplots referente aos elemento pontos, nível e *rank* não mostraram de grande ajuda para os usuários em relação ao problema de *feedback*. Os boxplots relativos aos elementos *rank* e *nível* são idênticos, com mediana 5 e valores bastante distribuídos. Não muito diferente, pode-se observar a avaliação do elemento *ponto*, que tem valores mais concentrados entre 4 e 5, entretanto com a mesma distribuição de valores por toda a escala de 1 a 10.

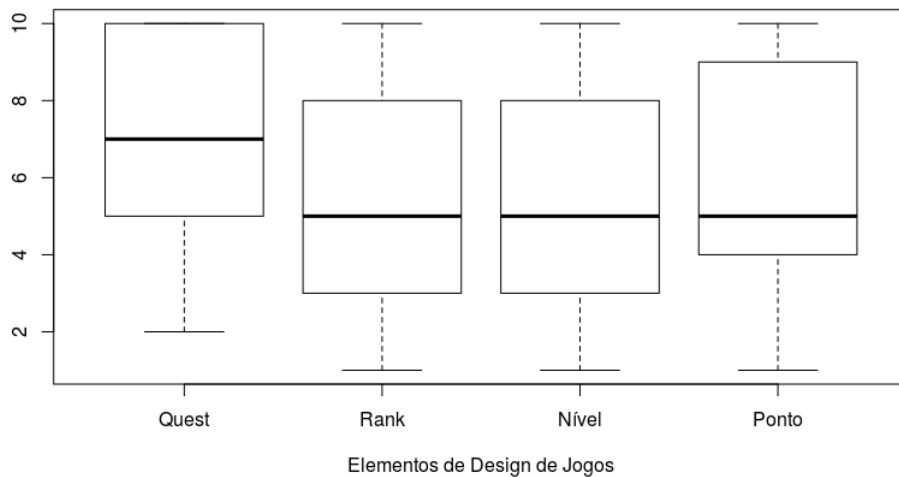


Figura 5.6. Conjunto de boxplot referente a avaliação dos elementos x barreiras - Barreira Falta de *Feedback*

Fonte: autoria própria

Com relação ao apoio para transpor a barreira “Falta de *feedback*” o elemento *quest* obteve resultado satisfatório, com uma mediana de valor 7, entretanto com algumas avaliações negativas, como pode ser observado na distribuição das avaliações no boxplot. Os demais elementos tiveram resultados menos satisfatórios, com pelo menos metade das notas iguais ou abaixo de 5.

A Figura 5.7 apresenta o terceiro conjunto de boxplots, referente à pergunta “*Com relação a barreira “Falta de conhecimento prévio dos novatos”, ou seja, quando o usuário não conhece o repositório utilizado pelo projeto, ou o processo para fazer a contribuição. Em uma escala de 1 a 10, o quanto cada elemento ajudou a diminuir/quebrar essa barreira?*”.

Na Figura 5.7 o boxplot referente ao elemento *quest* obteve as melhores avaliações. A mediana ficou com valor 9, o que quer dizer que metade das pessoas deram nota maior ou igual a 9. Além disso, 75% das notas concentraram-se entre os valores 7 e 10, que são valores altos. Apesar disso, pode-se observar que existem 3 notas iguais ou abaixo de 5.

Por outro lado o boxplot referente ao elemento *rank* não obteve bons resultados. A mediana ficou com valor 3 e maioria das notas estão concentradas entre os valores 1 e 6. Os boxplots referentes aos elementos *nível* e *ponto* obtiveram resultados não satisfatório, ambos com uma mediana 5 e com uma variação alta na distribuição.

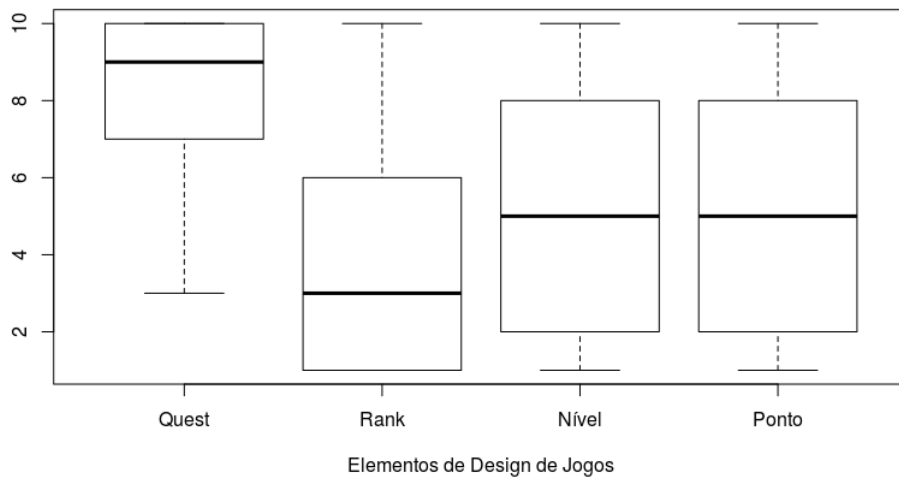


Figura 5.7. Conjunto de boxplot referente a avaliação dos elementos x barreiras - Barreira Falta de conhecimento prévio dos novatos

Fonte: autoria própria

Há indícios de que o elemento *quest* ajudou a maioria dos usuários a superar a barreira “Falta de conhecimento prévio dos novatos” e também foi o elemento que mais ajudou em relação as outras barreiras apresentadas. Por outro lado o elemento *rank* não ajudou os usuários a superar a barreira. Os elementos *nível* e *ponto* não tiveram satisfatório, visto a disparidade dos resultados apresentados no boxplot. Nota-se que os elementos parecem ajudar alguns estudantes e outros não.

Em resposta à questão de Pesquisa (2) - *Os elementos de design de jogos – nível, quest, rank e pontos – ajudaram a reduzir as barreiras encontradas por novatos?*

Em nosso estudo foi possível concluir que o elemento de *quest* mostrou-se o mais útil dos elementos no aspecto de ajudar a reduzir as três barreiras apresentadas. Esse elemento obteve bons resultados para todas as barreiras analisadas (*Falta de Feedback, Falta de Orientação e Falta de Conhecimento Prévio dos Novatos*). Há indícios ainda de que o elemento *ponto* ajudou a reduzir a barreira *Falta de Orientação*, porém para as demais barreiras o mesmo não obteve resultados satisfatórios. Os elementos *rank* e *nível* não mostraram resultados satisfatórios para nenhuma das barreiras apresentadas.

5.3.4. Avaliação Geral dos Elementos de *Design* de Jogos

Foi solicitado aos alunos que avaliassem os elementos de *design* de jogos de uma forma mais geral. O objetivo é saber qual dos elementos foi mais interessante, quais elementos os usuários

não gostaram e quais elementos os usuários gostariam de utilizar em um outro ambiente *gamified*.

A Tabela 5.2 apresenta quais elementos o usuário achou mais interessante como auxílio a contribuição. A Tabela mostra os resultados em ordem decrescente. Os valores inteiros representam quantidade de pessoas que selecionaram aquela opção e os valores entre parênteses representam o percentual dos valores inteiros em relação ao tamanho total da amostra (17). Nessa pergunta os usuários poderiam marcar mais uma ou mais opções.

Tabela 5.2. Avaliação dos elementos de *design* de jogos - questão 1

Pergunta	<i>quest</i>	ponto	nível	<i>rank</i>
Qual (ou quais) elemento de <i>gamification</i> você achou mais interessante como auxílio a contribuição?	10 (58,8%)	9 (53,0%)	4 (23,5%)	3 (17,6%)

A Tabela 5.2 mostra que os elementos mais votados foram os elementos *quest* e ponto. Mais de 50% dos estudantes informaram que os elementos foram interessantes como auxílio à contribuição. Já os elementos nível e *rank* não se apresentaram muito interessantes, sendo votados por apenas 30% dos estudantes.

A Tabela 5.3 mostra as respostas referente aos elementos que os usuários não voltariam a utilizar.

Tabela 5.3. Avaliação dos elementos de *design* de jogos - questão 2

Pergunta	<i>rank</i>	Nenhum	nível	ponto	<i>quest</i>
Qual elemento você não voltaria a utilizar?	7 (42,1%)	5 (29,4%)	5 (29,4%)	3 (17,6%)	1 (5,8%)

A Tabela 5.3 mostra que 42,1% dos usuários não voltariam a utilizar o elemento *rank*, na sequência têm-se a opção Nenhum (29,4%) que são os usuários que gostaram de todos os elementos e voltariam a utilizá-los, seguido do elemento nível (29,4%), ponto (17,6%) e *quest* (5,81%). Nessa Tabela é possível concluir que quase metade dos usuários não voltariam a utilizar o elemento *rank*. Os demais elementos não tiveram uma votação tão representativa quanto o elemento *rank*.

A Tabela 5.4 apresenta os resultados relacionado aos elementos que os usuários gostariam de utilizar em um ambiente *gamified*.

Tabela 5.4. Avaliação dos elementos de *design* de jogos - questão 3

Pergunta	Medalhas	Barra de Progresso	<i>Feed</i> de Notícias	Presentes	<i>Feedback</i>
Qual elemento você gostaria de utilizar em um outro ambiente <i>gamified</i> ?	9 (52,9%)	9 (52,9%)	5 (29,4%)	3 (17,6%)	1 (5,80%)

Os resultados mostram que mais da metade dos usuários gostariam de utilizar os elementos Medalha e Barra de Progresso, seguido do elemento *Feed* de notícia, Presentes e *Feedback*.

5.3.5. Avaliação do Ambiente *Gamified*

Um das etapas do questionário foi a avaliação do ambiente *gamified* essa avaliação conta com quatro afirmações, e utiliza uma escala Likert de 1 a 5 com as mesmas opções das outras avaliações.

A primeira afirmação “O ambiente gamificado não me ajudou em nada ao tentar fazer a contribuição.” obteve como maioria a opção Discordo Plenamente (40%), seguida das opções Discordo e Neutro (ambas com 24%) e como minoria as opções Concordo e Concordo Plenamente (ambas com 6%).

A segunda afirmação “Eu não utilizaria o GitLab gamificado outras vezes.” obteve como maioria a opção Discordo Plenamente (35%), na sequência tem a opção Discordo (29%), Neutro (24%) e como minoria a opção Concordo (12%).

A terceira afirmação “Eu utilizaria o GitLab gamificado outras vezes.” obteve como maioria a opção Concordo Plenamente (34%), seguida das opções Concordo e Neutro (ambas com 24%) e como minoria a opção Discordo (18%).

A quarta e última afirmação “O ambiente gamificado me ajudou ao tentar fazer a contribuição.” obteve como maioria a opção Concordo (47%), seguida da opção Concordo Plenamente (35%), Neutro (12%) e como minoria a opção Discordo (6%).

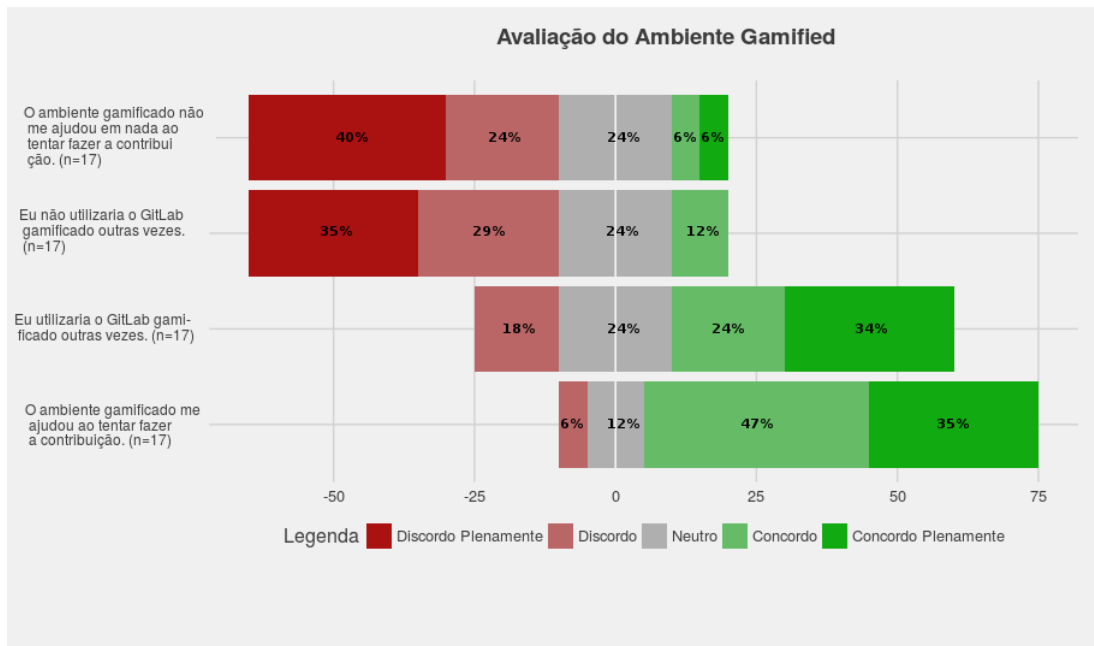


Figura 5.8. Conjunto de gráficos de avaliação do Ambiente *gamified*

Fonte: autoria própria

Em relação a primeira e segunda afirmações, as opções positivas são Discordo Plenamente e Discordo e as negativas são Concordo e Concordo Plenamente. Analisando os resultados referentes a essas afirmações é possível notar que ambos tiveram resultados positivos. A primeira afirmação obteve um total de 64% de valores positivos e um total de 12% de valores negativos. E a segunda afirmação obteve um total de 68% de valores positivos e 12% de valores negativo. Em ambas as perguntas a opção Neutro obteve 24%.

Em relação a terceira e quarta afirmações, as opções negativas são Discordo Plenamente e Discordo e as positivas são Concordo e Concordo Plenamente. Analisando os resultados referentes a essas afirmações, é possível notar que a maioria das avaliações foram positivas, sendo que a terceira possui um total de 58% de avaliações positivas e quarta um total de 82%. Em ambas as afirmações não teve a opção Discordo Plenamente marcada, as avaliações negativas foi da opção Discordo, sendo a terceira com 18% e quarta com 6%.

Analisando o gráfico e comparando-os, a primeira e a quarta afirmação (elas se contrapõe). Assim, pode-se concluir que o ambiente *gamification* ajudou a maioria dos usuários ao tentarem fazer a contribuição, visto que os resultado positivos foram maior que 60% e os negativos menor que 10%. Ao analisar a segunda e a terceira afirmações (elas se contrapõe), pode-se concluir que a maioria dos usuários utilizariam o ambiente *gamification* uma outra vez, visto que as avaliações negativas possuem um total menor que 20% e as positivas um total maior que 60%.Se

Respondendo à questão de Pesquisa (3) - *As técnicas de gamification motivaram os alunos na tentativa de fazer uma contribuição em projetos de software livre?*

Com base nos gráficos e resultados apresentados, é possível concluir que a técnica de *gamification* motiva a grande parte dos alunos na tentativa de fazer uma contribuição em projetos de software livre. É importante antes fazer uma validação dos elementos de *design* de jogos, analisando quais elementos melhor se adaptam ao problema. Em nosso estudo os elementos *quest* e ponto obtiveram os melhores resultados em relação ao aspecto motivacional. Os elementos *rank* e nível tiveram resultados medianos e não significativos no quesito motivacional. Também é apresentado nas avaliações que a maioria dos alunos utilizariam o ambiente *gamification* outras vezes e que o mesmo os ajudou na tentativa de fazer uma contribuição.

Conclusão

Os resultados mostram que as comunidades de software livre podem se beneficiar de elementos de design de jogos para auxiliar novatos. Em especial, destacamos o elemento *quest*, que obteve avaliações ótimas por parte dos estudantes como suporte para superar barreiras de orientação, *feedback* e falta de conhecimento prévio dos novatos. Além disso, o elemento *quest* e pontos mostraram-se elementos que mantêm os novatos motivados durante sua jornada de entrada em um projeto.

Os resultados obtidos das avaliações do elemento ponto também mostraram-se satisfatórios no aspecto motivacional e também como uma ferramenta que pode ser utilizada como forma de fornecer *feedback* para as ações realizadas pelos usuários. Nos dois aspectos as avaliações foram positivas e tiveram um percentual igual ou acima de 70%.

Os resultados relacionados aos elementos nível e *rank* não foram demasiadamente negativos, porém as avaliações positivas ficaram entre 40% e 50%, assim as avaliações que mais destacavam eram as Neutras. Assim foi possível concluir que tais elementos não tiveram impacto motivacional, não ajudaram a superar nenhuma das barreiras e também não despertaram nos usuários as características que ambas tinham potencial de despertar, como: gerar concorrência, ajudar o usuário a escolher um *bug* e assim despertar o interesse dos usuários a produzirem mais.

Os resultados apresentados sobre o ambiente *gamified* foram bem conclusivos e mostraram com clareza que a técnica de *gamification* auxiliou os usuários na tentativa de fazer contribuições em projetos de software live, assim obtendo uma avaliação positiva de mais de 80% dos usuários. Também pode-se concluir que a maioria dos usuários voltariam a utilizar o ambiente *gamified* outras vezes.

Uma conclusão inesperada foi em relação a alguns resultados na avaliações do elementos de *design* de jogos. Os elementos *rank* e nível possuem a característica de promover concorrência e com isso fazer com que os usuários produzam mais, porém isso não ocorreu.

Um dos pontos positivos foi relacionar os elementos com as barreiras. Essa análise trouxe resultados mais claros em relação as outras avaliações. Foi possível validar que alguns dos elementos contribuem bastante com uma ferramenta para quebrar/diminuir determinadas barreiras.

Para um estudo futuro é interessante utilizar os elementos que deram bons resultados neste estudo, como o elemento *quest* e ponto, também adicionar novos elementos. Uma das perguntas no questionário era relacionado a quais elementos os usuários gostariam de utilizar em um outro ambiente *gamified*, sendo os principais elementos: Medalhas, Barra de Progresso e *Feed* de Notícias.

6.0.1. Limitações

Dentro os resultados apresentados, os elementos de *design* de jogos *rank* e nível obtiveram resultados medianos. Esse resultado pode ser atribuído ao fato dos elementos não se encaixarem no contexto de comunidades de software livre. Uma hipótese é que o *rank* ao invés de gerar concorrência e motivar os usuários (que são as características do elemento), ele expunha os usuários de uma forma desagradável e intimidadora, onde o usuário que não tinham pontuação alta sentiam-se diminuídos em relação há outros usuários, o que pode acarretar em desmotivar o usuário. Outro fator foi que o elemento *rank* não ficou visível o tempo todo como os demais elementos, o que levou os usuários a não o utilizar.

O elemento nível tinha um propósito de situar o usuário na comunidade. Por exemplo, se o usuário possui um determinado nível, ele deve resolver *bugs* com um nível similar ao dele. Seria uma forma do usuário encontrar tarefas associadas ao seu nível. Uma das hipótese foi que o nível não variou o seu valor tanto quanto o ponto, ou seja, o ponto possui uma escala mais variável, onde o usuário atingia valores diversos (100, 120, 280, 300, 1000), por outro lado o nível variou pouco, o usuário mais bem colocado atingiu o nível 4. Assim, o usuário não via tanta diferença, por exemplo, na exibição do *rank* por nível quase todos usuários ficaram empatados (entre o nível 2 e 3) e na exibição por pontos o *rank* foi mais distribuído.

O objetivo do estudo foi de fazer contribuições com codificação para projetos de software livre. Quando analisada a Tabela 5.1, a segunda pergunta apresenta que 35,3% dos estudantes fizeram contribuições com codificação para algum software livre. Por outro lado se analisar a terceira pergunta, relacionada ao presente estudo, apenas 11,7% dos estudante conseguiram fazer uma contribuição com codificação. Assim, pode-se notar que o número de contribuição caiu com uso da *gamification*, porém isso pode dar-se ao tempo utilizado para realizar o estudo que foram apenas de 15 dias.

Referências

- ALENCAR, Anderson; MURILO, Machado; RAFAEL, Evangelista; SÉRGIO, Silveira; VICENTE, Aguiar. *Software livre, cultura hacker e ecossistema da colaboração*. São Paulo, SP, BRA: Momento Editorial, 2009. ISBN 978-85-62080-03-6.
- BARTLE, Richard Allan. Hearts, clubs,diamonds, spades: Players who suit muds. *The Journal of Virtual Environments* 1, 1996. Disponível em: brandeis.edu/pubs/jove/HTML/v1/bartle.html.
- BARTLE, Richard Allan. *Designing Virtual Worlds*. 1. ed. San Francisco, CA, USA: New Riders, 2003. v. 1. (10, v. 1). ISBN 0131018167.
- BERGQUIST, Magnus; LJUNGBERG, Jan. The power of gifts: organizing social relationships in open source communities. *Information Systems Journal*, Blackwell Science Ltd, v. 11, n. 4, p. 305–320, 2001. ISSN 1365-2575.
- C. REINERS R, Dencheva S. Zimmermann A. Prause. Incentives for maintaining high-quality source code. 2010.
- CHRISTOPH, Roberto de Holanda. Engenharia de software para software livre. In: . Rio de Janeiro, RJ, BRA: Pontifícia Universidade Católica do Rio de Janeiro, 2004. p. 118.
- COOK, Daniel. What are game mechanics. lost garden. Disponível em:www.lostgarden.com/. 2006.
- COON, Dennis. *Introduction to Psychology: Gateways to Mind and Behavior*. 13. ed. New York, NY, USA: Wadsworth Publishing, 2008. 768 p.
- CSIKSZENTMIHALY, Mihaly. *Beyond Boredom and Anxiety: Experiencing Flow in Work and Play*. 25. ed. Hoboken, NJ, EUA: Jossey-Bass, 2000. 231 p.
- DETERDING; SEBASTIAN; DIXON; DAN; KHALED; RILLA; NACKE; LENNART. From game design elements to gamefulness: Defining "gamification". In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. New York, NY, USA: ACM, 2011. (MindTrek '11), p. 9–15.
- DICHEVA, Darina; DICHEV, Christo; AGRE, Gennady; ANGELOVA, Galia. Gamification in education: A systematic mapping study. *Educational Technology & Society*, v. 18, n. 3, p. 75–88, 2015.
- DORLING, Alec; MCCAFFERY, Fergal. The gamification of spice. In: *Software Process Improvement and Capability Determination*. Co Louth, IRL: Springer Berlin Heidelberg, 2012. v. 290, p. 295–301.

- DUBOIS, Daniel J.; TAMBURRELLI, Giordano. Understanding gamification mechanisms for software development. In: *Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*. New York, NY, USA: ACM, 2013. (ESEC/FSE 2013), p. 659–662.
- FADEL, Luciane Maria; ULBRICHT, Vania Ribas; BATISTA, Claudia Regina; VANZIN, Tarcísio. *Gamificação na educação*. 2014. ed. São Paulo: Pimenta Cultural, 2014. v. 1. ISBN 978-85-66832-13-6.
- FERRO, Lauren S.; WALZ, Steffen P.; GREUTER, Stefan. Towards personalised, gamified systems: An investigation into game design, personality and player typologies. In: *Proceedings of The 9th Australasian Conference on Interactive Entertainment: Matters of Life and Death*. New York, NY, USA: ACM, 2013. p. 7:1–7:6.
- FICHMAN, Robert G.; KEMERER, Chris F. The assimilation of software process innovations: An organizational learning perspective. *Manage. Sci.*, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 43, n. 10, p. 1345–1363, out. 1997.
- FSF, Free Software Foundation. *What is Free Software?* 5. ed. gnu.org/philosophy/free-sw.html, 2015. Visitado em novembro de 2015.
- FULLERTON, Tracy Jeanne. *Game Design Workshop: A Playcentric Approach to Creating Innovative Games*. 3. ed. Burlington, MA, USA: AK Peters, 2008. ISBN 1482217163.
- GALEGALE, Gustavo Perri. A utilização de gamification em um sistema de informação: Estudo de caso na natura cosméticos s.a. In: . São Paulo,SP, BR: Dissertação (Mestrado) – Universidade de São Paulo, 2014., 2014.
- GIL, Antônio Carlos. *Como elaborar projetos de pesquisa*. 2010. ed. São Paulo: Atlas Editorial, 1991. v. 5. ISBN 9788522458233.
- HALAN, Shivashankar; ROSSEN, Brent; CENDAN, Juan; LOK, Benjamin. High score! - motivation strategies for user participation in virtual human development. In: _____. *Intelligent Virtual Agents: 10th International Conference, IVA 2010, Philadelphia, PA, USA, September 20-22, 2010. Proceedings*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.
- HARS, Alexander; OU, Shaosong. Working for free? motivations for participating in open-source projects. *Int. J. Electron. Commerce*, M. E. Sharpe, Inc., Armonk, NY, USA, v. 6, n. 3, p. 25–39, abr. 2002. ISSN 1086-4415.
- KROGH, Georg von; SPAETH, Sebastian; LAKHANI, Karim R. Community, joining, and specialization in open source software innovation: a case study. *Research Policy*, v. 32, n. 7, p. 1217 – 1241, 2003.
- LAKHANI, Karim; HIPPEL, Eric von. How open source software works: "free" user-to-user assistance. *Research Policy*, v. 32, p. 923–943, 2000.
- LI, Wei; GROSSMAN, Tovi; FITZMAURICE, George. Gamicad: A gamified tutorial system for first time autocad users. In: *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*. New York, NY, USA: ACM, 2012. (UIST '12).

LOMBRISER, Roald van der Valk Philipp. Improving the quality of the software development lifecycle with gamification. p. 22, 2014.

Gamers' Personality and Their Gaming Preferences. Universidad de Belgrano: School of Interactive Arts and Technology, 2010.

MCMAHON, Brenda; PORTELLI, John P. Engagement for what? beyond popular discourses of student engagement. *Leadership and Policy in Schools*, v. 3, n. 1, p. 59–76, 2004.

MCMAHON, Nicole; WYETH, Peta; JOHNSON, Daniel. Personality and player types in fallout new vegas. In: *Proceedings of the 4th International Conference on Fun and Games*. New York, NY, USA: ACM, 2012.

MELO, AlexandreAltair de; HINZ, Mauro; SCHEIBEL, Glaucio; BERKENBROCK, Carla Diacui Medeiros; GASPARINI, Isabela; BALDO, Fabiano. Version control system gamification: A proposal to encourage the engagement of developers to collaborate in software projects. In: *Social Computing and Social Media*. Joinville, SC, Brazil: Springer International Publishing, 2014. p. 550–558.

MOCKUS, Audris; FIELDING, Roy T.; HERBSLEB, James. A case study of open source software development: the apache server. In: *In: Proceedings of the 22nd International Conference on Software Engineering (ICSE 2000)*. Naperville, IL, USA: ACM Press, 2000. p. 263 – 272.

MOODY, G. *Rebel Code: Linux and the Open Source Revolution Penguin*. 1. ed. New York, NY: Basic Books, 2001. 344 p.

PEDREIRA, Oscar; GARCÍA, Félix; BRISABOIA, Nieves; PIATTINI, Mario. Gamification in software engineering – a systematic mapping. *Information and Software Technology*, v. 57, p. 157 – 168, 2015.

PRAUSE, Jan Nonnen Christian R.; VINKOVITS, Mark. A field experiment on gamification of code quality in agile development. 2012.

PRENSKY, Marc. *The Digital Game-Based Learning Revolution*. 0. ed. Palo Alto CA, USA: Paragon House, 2001. ISBN 1557788634.

REIS, C.; CHAIM; FORTES, Marcos L.; MATTOS, Renata P. de; GIMENES; M., Itana. Caracterização de um processo de software para projetos de software livre. In: . São Paulo, SP, BRA: Universidade de São Paulo - USP, 2003.

REYMOND, Eric S. *The Cathedral & the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. 1. ed. Farnham, UK: O'Reilly Media, 1999. 241 p.

RYAN, Sukale; S., Pfaff Mark. Quodocs: Improving developer engagement in software documentation through gamification. In: *CHI '14 Extended Abstracts on Human Factors in Computing Systems*. Indianapolis, IN, USA: Indiana University, 2014.

SAWHNEY, Emanuela Prandelli Mohanbir. Communities of creation: Managing distributed innovation in turbulent markets. *California Management Review*, Vol. 42 No. 4, 2000.

SCHLECHTY, P. Increasing student engagement. *Missouri Leadership Acad*, 1994.

- SHETH, Swapneel Kalpesh; BELL, Jonathan Schaffer; KAISER, Gail E. Increasing student engagement in software engineering with gamification. Columbia University Computer Science Technical Reports, 2012.
- SINGER, L.; SCHNEIDER, K. It was a bit of a race: Gamification of version control. In: *Games and Software Engineering (GAS), 2012 2nd International Workshop on*. Piscataway, NJ, USA: Zurich, 2012. p. 5–8.
- SNIPES, Will; NAIR, Anil R.; MURPHY-HILL, Emerson. Experiences gamifying developer adoption of practices and tools. In: *Companion Proceedings of the 36th International Conference on Software Engineering*. New York, NY, USA: ACM, 2014. (ICSE Companion 2014).
- SNOW, Shane. How to: Use game mechanics to power your business. Disponível: mashable.com/2010/07/13/game-mechanics-business/#bMFSzz2V35qu. 2010.
- STAEWEN, R.; TREVINO, P.; YUN, Chang. Player characteristics and their relationship to goals and rewards in video games. In: *Games Media Entertainment (GEM), 2014 IEEE*. Toronto, ON: IEEE, 2014. p. 1–8.
- STEINMACHER, Igor; CONTE, Tayana; GEROSA, Marco Aurélio; REDMILES, David. Social barriers faced by newcomers placing their first contribution in open source software projects. In: *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work; Social Computing*. New York, NY, USA: ACM, 2015. (CSCW '15), p. 1379–1392.
- SUITS, Bernard Herbert. *The Grasshopper: Games, Life and Utopia*. 0. ed. Tonawanda, NY, USA: Broadview Press, 1978. ISBN 155111772X.
- VASILESCU, Bogdan. Human aspects, gamification, and social media in collaborative software engineering. In: *Companion Proceedings of the 36th International Conference on Software Engineering*. New York, NY, USA: ACM, 2014. (ICSE Companion 2014), p. 646–649.
- VIANNA, Ysmar; VIANNA, Maurício; MEDINA, Bruno; TANAKA, Samara. *Gamification, Inc. - Como Reinventar Empresas A Partir de Jogos*. 0. ed. Janeiro, RJ, Brazil: MJV Press, 2013. ISBN 9788565424080.
- WAYNE, Peter. *Free for All: How LINUX and the Free Software Movement Undercut the High-Tech Titans*. 1. ed. Nova Iorque, Nova Iorque, EUA: HarperBusiness, 2000. 340 p.
- WERBACH, Dan Hunter Kevin. *For the Win: How Game Thinking Can Revolutionize Your Business*. 2012. ed. EUA: Wharton Digital Press, 2012. ISBN 1613630239.
- WU, Chorng-Guang; GERLACH, James H.; YOUNG, Clifford E. An empirical analysis of open source software developers' motivations and continuance intentions. *Information & Management*, v. 44, n. 3, p. 253 – 262, 2007. ISSN 0378-7206.
- ZICHERMANN, G.; CUNNINGHAM, C. *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*. Farnham, UK: O'Reilly Media, 2011. (O'Reilly Series).
- ZIMMERMAN, Katie Salen Tekinbaş Eric. *Rules of Play: Game Design Fundamentals*. 0. ed. Cambridge, Massachusetts, EUA: The MIT Press, 2004. ISBN 0262240459.

Questionário utilizado no estudo

Questionário de Trabalho de Conclusão de Curso - *gamification*

Natureza da Pesquisa

O sra (sr.) está sendo convidada (o) a participar desta pesquisa que tem como finalidade avaliar a página web GitLab que é um gerenciador de repositório de software no qual sofreu alterações em seu código fonte. Essa página foi instanciada em um servidor local (172.16.2.221). Para fins de pesquisa o código do Gitlab foi modificado, foi inserido a técnica de *gamification* que consiste em utilizar elementos de jogos (pontos, *quest*, *Níveis* e etc) a fim de motivar alunos do curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná - Campus Campo Mourão na contribuição de projetos de software livre.

Participantes da Pesquisa

A pesquisa conta com um total de dezoito (17) estudantes.

Envolvimento na Pesquisa

Ao participar deste estudo a sra (sr) permitirá que o (a) pesquisador (a) avalie se a técnica de *gamification* motiva/engaja alunos á contribuição com projetos de software livre. A sra (sr.) tem liberdade de se recusar a participar e ainda se recusar a continuar participando em qualquer fase da pesquisa, sem qualquer prejuízo para a sra (sr.). Sempre que quiser poderá

pedir mais informações sobre a pesquisa através do telefone do (a) pesquisador (a) do projeto e, se necessário através do telefone do Comitê de Ética em Pesquisa.

Riscos e Desconforto

A participação nesta pesquisa não traz complicações legais. Os procedimentos adotados nesta pesquisa obedecem aos Critérios da Ética em Pesquisa com Seres Humanos conforme Resolução no. 196/96 do Conselho Nacional de Saúde. Nenhum dos procedimentos usados oferece riscos à sua dignidade.

Confidencialidade

Todas as informações coletadas neste estudo são estritamente confidenciais. Somente o (a) pesquisador (a) e o (a) orientador (a) terão conhecimento dos dados.

Benefícios

Ao participar desta pesquisa a sra (sr.) não terá nenhum benefício direto. Entretanto, esperamos que este estudo traga informações importantes sobre utilização da técnica de gamification, de forma que o conhecimento que será construído a partir desta pesquisa possa auxiliar pesquisas futuras, onde pesquisador se compromete a divulgar os resultados obtidos, respeitando-se o sigilo das informações coletadas, conforme previsto no item anterior.

Pagamento

A sra (sr.) não terá nenhum tipo de despesa para participar desta pesquisa, bem como nada será pago por sua participação.

Informações Pessoais

Após estes esclarecimentos, solicitamos o seu consentimento de forma livre para participar desta pesquisa. Portanto preencha, por favor, os itens que se seguem.

Os itens que possuem o símbolo * (asterisco) na frente são obrigatórios.

- Nome *

- Registro Acadêmico (RA) *

- Idade *

- Email *
- Que Período está Cursando? *

Conhecimento em Software Livre

Nesta Seção o objetivo é saber o envolvimento do usuário com softwares livres, se fez alguma contribuição em algum projeto e etc.

- Você já tentou contribuir alguma vez com projetos de software livre? *
- Você já contribuiu com codificação para algum software livre? *
- Quanto tempo (meses ou dias) você utilizou para fazer a contribuição? * **IMPOR- TANTE:** Uma contribuição com código só é válida quando o *pull request* é aceito.
- Quais as maiores dificuldade encontradas no processo de fazer a contribuição? *
- No estudo em sala de aula você conseguiu fazer uma contribuição com codificação? *

Avaliação do elemento *quest*

Objetivo do elemento *quest* é servir como guia para o usuário, tendo pequenas tarefas bem definidas a fim de guiá-los a fazerem contribuições com softwares livre. Motivar os usuários provendo desafios que permitem alcançar realizações e quebrar uma das barreira encontradas por novatos que é a falta de orientação.

Em uma escala de 1 a 5, sendo: 1 “Discordo Plenamente” e 5 “Concordo Plenamente”, responda ao questionário a seguir.

- O elemento *quest* me guiou a fazer a contribuição. *
- O elemento *quest* me motivou a realizar as tarefas propostas e assim fazer a contribuição.
*
- O elemento *quest* serviu como orientação para fazer a contribuição. *

Avaliação do elemento ponto

O elemento ponto é responsável por fornecer *feedback* e mostrar o progresso do jogador. pontos são conectados com a necessidade de recompensa e recompensa gera motivação. Ele também está ligado há outros elementos como *rank* e nível pois estes são calculados de acordo com a quantidade de pontos.

- O elemento pontos me proporcionou *feedback* das tarefas realizadas. *
- Quanto mais pontos eu conseguia mais eu queria ter, a fim de aumentar meu nível e meu *rank*. *
- O elemento pontos me motivou a tentar fazer contribuição. *

Avaliação do elemento nível

O elemento nível é a representação da capacidade do jogador dentro do jogo. Aumenta com o decorrer do jogo e com a evolução do engajamento do jogador. Ajuda a tornar as tarefas mais desafiadoras, o usuário ganha novas competências e experiências. Níveis são úteis para demonstrar status e também colocar limites para que o jogador saiba o que no momento ele está apto a realizar. O elemento nível estava conectado com os elementos *quest* e pontos.

- Eu queria aumentar meu nível para liberar novas *quests*. *
- Eu utilizei o elemento nível para escolher o *bug*, de acordo com meu nível eu selecionei um *bug* de dificuldade compatível. *
- Eu consultava frequentemente o meu nível para comparar com outros usuários. *

Avaliação do Elemento *rank*

O elemento *rank* permite que os usuários acompanhem o seu desempenho em relação aos outros participantes, elas geram concorrência entre os jogadores. *Rank* é uma ordenação do maior para o menor de acordo com uma métrica estabelecida (quantidade total de pontos, Níveis e etc).

- Eu queria fazer mais *quest* e ganhar mais pontos para melhorar minha posição no *rank*. *

- O elemento rank foi motivador pois por ele eu conseguia ver meu desempenho em relação a outros usuários. *

Gamification x Barreiras

Na pesquisa sobre quais dificuldades novatos tinham ao tentarem contribuir com projetos de softwares livres, foram encontradas barreiras, como: **Problemas de recepção, Falta de orientação, Documentação com problemas, Características do código ou problemas de entendimento, Falta de *feedback* das tarefas e etc.**

- Com relação a barreira **Falta de orientação** que é o fato de usuários veteranos não ajudarem e também não ter um passo a passo de como contribuir. Em uma escala de 1 a 10, o quanto cada elemento ajudou a diminuir/quebrar essa barreira? *
 - *Quest*
 - *Rank*
 - Nível
 - Ponto
- Com relação a barreira **Falta de *feedback***, ou seja, quando o usuário faz uma ação ele não sabe se está correto ou não, se o caminho escolhido é o correto. Em uma escala de 1 a 10, o quanto cada elemento ajudou a diminuir/quebrar essa barreira? *
 - *Quest*
 - *Rank*
 - Nível
 - Ponto
- Com relação a barreira **Falta de conhecimento prévio de novatos**, ou seja, quando o usuário não conhece o repositório utilizado pelo projeto, ou o processo para fazer a contribuição. Em uma escala de 1 a 10, o quanto cada elemento ajudou a diminuir/quebrar essa barreira? *
 - *Quest*
 - *Rank*
 - Nível
 - Ponto

Perguntas Finais

Essas perguntas são referentes ao ambiente *gamified* (utilizado no experimento) e também servirá como base para pesquisas futuras.

Em uma escala de 1 a 5, sendo: 1 “Discordo Plenamente” e 5 “Concordo Plenamente”, responda ao questionário a seguir.

- Eu utilizaria o GitLab gamificado outras vezes. *
- Eu não utilizaria o GitLab gamificado outras vezes. *
- O ambiente gamificado não me ajudou em nada ao tentar fazer a contribuição. *
- O ambiente gamificado me ajudou ao tentar fazer a contribuição.*
- Qual (ou quais) elemento de gamificação você achou mais interessante como auxílio a contribuição.*
 - *Quest*
 - *Rank*
 - Nível
 - Ponto
 - Nenhuma das opções
- Qual (ou quais) elemento de gamificação você achou mais interessante como auxílio a contribuição.*
 - *Quest*
 - *Rank*
 - Nível
 - Ponto
 - Nenhuma das opções
- Qual elemento você não voltaria a utilizar? *
 - *Quest*
 - *Rank*
 - Nível
 - Ponto
 - Nenhuma das opções
- Qual elemento você gostaria de utilizar em um outro estudo? *
 - Medalhas
 - Barra de Progresso
 - *Feed* de notícias

- Presentes
- Votação
- Outro