

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Colegiado de Sistemas de Informação**



UFOP
Universidade Federal
de Ouro Preto

**BOPE Game: Um sistema para
gamificação de processos de *software***

**GEOVANA DA CONCEIÇÃO
GONÇALVES**

**João Monlevade/MG
2017**

GEOVANA DA CONCEIÇÃO GONÇALVES

BOPE Game: Um sistema para gamificação de processos de *software*

Orientador: Igor Muzetti Pereira

Coorientador: Vicente José Peixoto de Amorim

Monografia apresentada ao curso de Sistemas de Informação do Departamento de Computação e Sistemas da Universidade Federal de Ouro Preto como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação

Universidade Federal de Ouro Preto

João Monlevade

2017

G635b Gonçalves, Geovana da Conceição.
Bope game [manuscrito]: um sistema para gamificação de processos de software / Geovana da Conceição Gonçalves. - 2017.

75f.: il.: color; tabs.

Orientador: Prof. Me. Igor Muzetti Pereira.

Monografia (Graduação). Universidade Federal de Ouro Preto. Instituto de Ciências Exatas e Aplicadas. Departamento de Computação e Sistemas de Informação.

1. Gamificação. 2. Motivação (Psicologia). 3. Android - programa de computador. 4. Software de aplicação. 5. Jogos. I. Pereira, Igor Muzetti. II. Universidade Federal de Ouro Preto. III. Título.

CDU: 004.9

Catálogo: ficha@sisbin.ufop.br



UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
COLEGIADO DO CURSO DE SISTEMAS DE INFORMAÇÃO

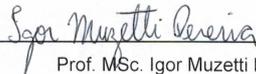
ATA DE DEFESA

Aos 4 dias do mês de Abril de 2017, às 15 horas, no Laboratório de Engenharia de Software, do Instituto de Ciências Exatas e Aplicadas, foi realizada a defesa de Monografia pela aluna **Geovana da Conceição Gonçalves**, sendo a Comissão Examinadora constituída pelos professores: Prof. MSc. Igor Muzetti Pereira, Prof. MSc. Vicente José Peixoto de Amorim, Prof. Dr. Rafael Frederico Alexandre e Prof. Dr. Fernando Bernardes de Oliveira.

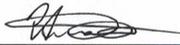
O candidato apresentou a monografia intitulada: "*BOPE Game: Um sistema para gamificação de processo de software*". A comissão examinadora deliberou, por unanimidade, pela aprovação do candidato, com nota 9,5 (nove e meio), concedendo-lhe o prazo de 15 dias para incorporação das alterações sugeridas ao texto final.

Na forma regulamentar, foi lavrada a presente ata que é assinada pelos membros da Comissão Examinadora e pelo graduando.

João Monlevade, 04 de abril de 2017



Prof. MSc. Igor Muzetti Pereira
Professor Orientador/Presidente



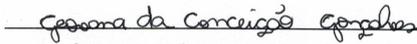
Prof. MSc. Vicente José Peixoto de Amorim
Professor Coorientador



Prof. Dr. Rafael Frederico Alexandre
Professor Convidado



Prof. Dr. Fernando Bernardes de Oliveira
Professor Convidado



Geovana da Conceição Gonçalves
Graduando



UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
COLEGIADO DO CURSO DE SISTEMAS DE INFORMAÇÃO

Folha de Aprovação
Curso de Sistemas de Informação

FOLHA DE APROVAÇÃO DA BANCA EXAMINADORA

BOPE Game: Um sistema para gamificação de processo de software

Geovana da Conceição Goncalves

Monografia apresentada ao Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto como requisito parcial da disciplina CSI499 – Trabalho de Conclusão de Curso II do curso de Bacharelado em Sistemas de Informação e aprovada pela Banca Examinadora abaixo assinada:

Prof. MSc. Igor Muzetti Pereira

DECSI - UFOP

Prof. MSc. Vicente José Peixoto de Amorim

DECSI - UFOP

Prof. Dr. Rafael Frederico Alexandre

DECSI - UFOP

Prof. Dr. Fernando Bernardes de Oliveira

DECSI - UFOP

João Monlevade, 04 de abril de 2017



UFOP
Universidade Federal
de Ouro Preto

UNIVERSIDADE FEDERAL DE OURO PRETO
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS
COLEGIADO DO CURSO DE SISTEMAS DE INFORMAÇÃO

Termo de Responsabilidade

TERMO DE RESPONSABILIDADE

Eu, Geovana da Conceição Gonçalves, declaro que o texto do trabalho de conclusão de curso intitulado "BOPE Game: Um sistema para gamificação de processos de software" é de minha inteira responsabilidade e que não há utilização de texto, material fotográfico, código fonte de programa ou qualquer outro material pertencente a terceiros sem as devidas referências ou consentimento dos respectivos autores.

João Monlevade, 04 de Abril de 2017

Geovana da conceição gonçalves
Assinatura do aluno

*Aos meus pais, irmãos, namorado e amigas que,
com carinho e paciência contribuíram para que eu pudesse alcançar esta etapa tão sonhada
e almejada*

Agradecimentos

Agradeço primeiramente a Deus por toda benção, força e sabedoria durante a minha graduação.

Aos meus familiares, em especial aos meus pais, Antônio e Ana, que sempre acreditaram em mim, apoiaram e não mediram esforços para que meu sonho fosse possível. Aos meus irmãos João Paulo e Felipe pelo companheirismo e pela torcida.

Ao meu namorado Flávio por todos esses anos juntos em que sempre acreditou em mim, pelo incentivo, companheirismo, conselhos principalmente nos momentos mais difíceis.

Aos amigos e amigas da faculdade que foram grandes companheiros durante toda a graduação, pelos momentos difíceis superados sempre com muita paciência.

As amigas de infância por todo incentivo e carinho desde sempre.

Ao meu professor e orientador Igor Muzetti pela oportunidade, por toda orientação, confiança e compartilhamento de conhecimentos.

A toda equipe do laboratório iMobilis pela convivência e amizade.

Obrigada a todos que contribuíram de alguma maneira para que a conclusão desta etapa tão importante em minha vida fosse possível.

“Se não puder se destacar pelo talento, vença pelo esforço.”
(Dave Weinbaum)

Resumo

Este trabalho é a documentação do resultado da implementação de um sistema *web* responsivo para a gamificação do processo de *software* BOPE no laboratório de computação Móvel e Sistemas Embarcados, o iMobilis-JM. Foi proposto o desenvolvimento de um *software* com o objetivo de motivar os estudantes universitários que fazem parte do laboratório a participarem das atividades, cumprirem com seus cronogramas e estarem em conformidade com as atividades do BOPE. Para alcançar esse objetivo, foi adaptado a gamificação no processo de *software* utilizado. Esta adaptação utilizou elementos de jogos em um contexto que não é um jogo. Os elementos da gamificação foram identificados de acordo com as atividades que são desenvolvidas no laboratório, considerando o perfil dos colaboradores e o ambiente de trabalho. Este trabalho relata todas as etapas utilizadas na construção do *software*, assim como as práticas de engenharia de *software* que foram aplicadas. O principal resultado deste trabalho foi a implantação do *software* BOPE Game que culminou na manutenção da produtividade e comprometimento dos colaboradores e na criação de um registro organizado dos dados da gamificação.

Palavras-chaves: Gamificação; Motivação; Comprometimento; Gestão de Projetos Ágil; Processo de *Software*; *Ruby on Rails*.

Abstract

This paper is the implementation results documentation of a responsive web system inside BOPE software process in the Mobile and Embedded Systems computing laboratory, iMobilis-JM. It was proposed a software development with the objective of motivating the university students who are part of the laboratory to participate in the activities, to comply with their schedules and the activities of BOPE. In order to achieve this goal, it was adapted to gamification in the software process used. This adaptation used games elements in a context that is not a game. The gamification elements were identified according to the activities that are developed in the laboratory, considering the profile of the collaborators and the work environment. This paper reports all the stages used in the software construction, as well as the software engineering practices that were applied. The main result of this work was the implementation of the BOPE Game software that culminated in the maintenance of the productivity and commitment of the collaborators and in the creation of an organized registry of the gamification data.

Key-words: Gamification; Motivation; Commitment; Agile Project Management; Software Process; Ruby on Rails.

Lista de ilustrações

Figura 1 – Protótipo de papel da tela Geração de Relatórios	32
Figura 2 – Estrutura dos arquivos e pastas gerados por padrão no <i>Rails</i>	34
Figura 3 – Configuração do Banco de Dados <i>PostgreeSQL</i>	36
Figura 4 – Exemplo de um arquivo de migração (<i>migrate</i>)	38
Figura 5 – Autenticação dos usuários	39
Figura 6 – Autorização dos usuários	40
Figura 7 – Planilha de inserção de pontuação individual da gamificação no período letivo 16/01	43
Figura 8 – Planilha de inserção de pontuação por equipe da gamificação no período letivo 16/01	44
Figura 9 – Tela de <i>login</i>	46
Figura 10 – Tela Principal	46
Figura 11 – Página de cadastro dos usuários do sistema	47
Figura 12 – Atividades cadastradas no sistema	47
Figura 13 – Tela lançamentos de pontos	48
Figura 14 – Tela criação de <i>sprint</i>	48
Figura 15 – Tela criação da semana	49
Figura 16 – Não conformidades observadas por atividade com a gamificação no período letivo 16/01	50
Figura 17 – Não conformidades observadas por atividade com a gamificação no período letivo 16/02	50
Figura 18 – História de Usuários – Serviço de Autenticação	62
Figura 19 – História de Usuários – Serviço de Gerenciamento de Usuários	62
Figura 20 – História de Usuários – Serviço de Gerenciamento de Usuários	63
Figura 21 – História de Usuários – Serviço de Atividades	63
Figura 22 – História de Usuários – Serviço de Visualização de Informações	64
Figura 23 – História de Usuários – Serviço de Visualização de Informações	64
Figura 24 – História de Usuários – Serviço de Gerenciamento de Sprin	65
Figura 25 – História de Usuários – Serviço de Não-conformidades	65
Figura 26 – Cenários de Teste – Serviço de Autenticação	66
Figura 27 – Cenários de Teste – Serviço de Visualização de Informações	67
Figura 28 – Cenários de Teste – Serviço de Visualização de Informações	67
Figura 29 – Cenários de Teste – Serviço de Gerenciamento de <i>sprint</i>	68
Figura 30 – Cenários de Teste – Serviço de Gerenciamento de Usuários	68
Figura 31 – Cenários de Teste – Serviço de Gerenciamento de Usuários	68
Figura 32 – Cenários de Teste – Serviço de Gerenciamento de Usuários	68

Figura 33 – Cenários de Teste – Serviço de Gerenciamento de Usuários	69
Figura 34 – Cenários de Teste – Serviço de Gerenciamento de Usuários	69
Figura 35 – Cenários de Teste – Serviço de Gerenciamento de Usuários	69
Figura 36 – Cenários de Teste – Serviço de Atividades	70
Figura 37 – Cenários de Teste – Serviço de Atividades	70
Figura 38 – Cenários de Teste – Serviço de Atividades	70
Figura 39 – Cenários de Teste – Serviço de Atividades	70
Figura 40 – Cenários de Teste – Serviço de Atividades	71
Figura 41 – Cenários de Teste – Serviço de Atividades	71
Figura 42 – Cenários de Teste – Serviço de Não conformidades	71
Figura 43 – Modelo de Entidade-Relacionamento	73
Figura 44 – Instalando <i>Ruby</i>	75
Figura 45 – Instalando <i>Rails e PostgreSQL</i>	76

Lista de tabelas

Tabela 1 – <i>Product Backlog</i>	28
Tabela 2 – Atividade e critérios de pontuação	44
Tabela 3 – Quantidade de semanas por <i>sprint</i>	49
Tabela 4 – Resultado Final Individual por <i>Sprint</i> no período letivo 16/01	52
Tabela 5 – Resultado Final Individual por <i>Sprint</i> no período letivo 16/02	53
Tabela 6 – Resultado Final Equipes por <i>sprint</i> no período letivo 16/01	54
Tabela 7 – Resultado Final Equipes por <i>sprint</i> no período letivo 16/02	55

Lista de abreviaturas e siglas

ICEA	Instituto de Ciências Exatas e Aplicadas
UFOP	Universidade Federal de Ouro Preto
FBM	<i>Fogg Behavior Model</i>
RoR	<i>Ruby on Rails</i>
MVC	Modelo-Visão-Controle
SGBD	Sistema Gerenciador de Banco de Dados
ORDBMS	<i>Object Relational Database Management System</i>
RUP	<i>Rational Unified Process</i>
PMBOK	<i>Project Management Body of Knowledge</i>

Sumário

1	INTRODUÇÃO	17
1.1	Problema	17
1.2	Objetivos	18
1.2.1	Objetivo geral	18
1.2.2	Objetivos específicos	18
1.3	Justificativa	18
1.4	Estrutura do trabalho	19
I	REVISÃO BIBLIOGRÁFICA	20
2	REVISÃO BIBLIOGRÁFICA	21
2.1	Conceitos básicos	21
2.1.1	Gamificação	21
2.1.2	A linguagem <i>Ruby</i>	22
2.1.3	<i>O framework Ruby on Rails</i>	23
2.1.4	<i>RailsAdmin</i>	23
2.1.5	Utilização do Gerenciador de Banco de Dados <i>PostgreSQL</i>	24
2.2	Trabalhos Correlatos	24
II	METODOLOGIA	26
3	METODOLOGIA	27
3.1	Concepção e Elaboração	27
3.1.1	<i>Product Backlog</i>	27
3.1.2	Histórias de Usuários	29
3.1.3	Cenários de Teste	30
3.1.4	Protótipos de Tela	32
3.1.5	Projeto de Banco de Dados	33
3.2	Implementação	33
3.2.1	Criação do Projeto	33
3.2.2	Autenticação e Autorização no sistema	38
3.2.3	Associação de tabelas no sistema	40
3.2.4	Geração de PDF no sistema	41
3.2.5	Implantação do sistema no laboratório iMobilis	41

III	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	42
4	APRESENTAÇÃO E ANÁLISE DOS RESULTADOS	43
4.1	Métodos de coleta de dados da gamificação	43
4.2	Monitoramento da gamificação utilizando o sistema BOPE Game	45
4.2.1	Acesso ao sistema	45
4.2.2	Cadastros dos Usuários e das Atividades	46
4.2.3	Lançamentos da Pontuação no Sistema	47
4.3	Resultados do Monitoramento da Gamificação	49
4.3.1	Identificação das Não Conformidades	49
4.3.2	Resultados Individuais nos períodos letivos 16/01 e 16/02	51
4.3.3	Resultados por Equipes	53
IV	CONCLUSÃO	56
5	CONCLUSÃO	57
6	LIMITAÇÕES E TRABALHOS FUTUROS	58
7	REFERÊNCIAS BIBLIOGRÁFICAS	59
	ANEXOS	61
	ANEXO A – HISTÓRIAS DE USUÁRIOS	62
	ANEXO B – CENÁRIOS DE TESTE	66
	ANEXO C – PROJETO DE BANCO DE DADOS	72
	ANEXO D – TUTORIAL DE IMPLANTAÇÃO DO SISTEMA NO LABORATÓRIO IMOBILIS	74

1 Introdução

Nos últimos anos vem crescendo o uso de ideias e mecanismos de jogos para incentivar as pessoas e mantê-las focadas em suas atividades organizacionais.

O termo criado para descrever esta estratégia é a gamificação (do original em inglês *gamification*). De acordo com Vianna *et al.* (2013), gamificação é o uso de mecanismos e dinâmica de jogos para engajar pessoas, resolver problemas e melhorar o aprendizado, motivando ações e comportamentos em ambientes fora do contexto de jogos.

Os desafios e recompensas destinados aos colaboradores que realizam as atividades preestabelecidas podem ser alcançados por meio de elementos de jogos e estratégias de gamificação que possuem a finalidade de motivar os colaboradores como também ajudar na solução de problemas.

Os elementos da gamificação utilizados no sistema envolvem pontuação, *ranking* e medalhas. O sistema de pontuação permite visualizar os pontos obtidos por cada colaborador em cada semana e *sprint*. Assim, é possível realizar o acompanhamento do placar que indica a posição relativa dos colaboradores, permitindo um *feedback* imediato e seguimento do processo de *software*.

A classificação dos colaboradores no sistema gamificado é denominada *rankings* de líderes e permite visualizar o desenvolvimento dos outros colaboradores e, assim, criar um senso de competição motivacional dentro do sistema. As medalhas simbolizam conquistas através de elementos gráficos que os colaboradores e equipes recebem por alcançarem o primeiro lugar no *sprint*.

O principal resultado deste trabalho foi a implantação do BOPE Game no iMobilis. Isto auxiliou os responsáveis no acompanhamento da gamificação, manutenção dos dados e identificação de potenciais “gatilhos” para melhoria da motivação, comprometimento, colaboração e habilidades de todos os envolvidos. O trabalho também sugere possíveis adições de funcionalidades ao sistema como trabalhos futuros.

1.1 Problema

O laboratório iMobilis-JM do campus ICEA/UFOP é um laboratório de pesquisa em computação, que desenvolve sistemas para dispositivos móveis, embarcados e *web* utilizando as boas práticas da engenharia de *software*. Segundo Cota (2016), o processo de *software* denominado BOPE, adota mecanismos baseados nos processos de *software* como o Scrum, *Extreme Programming*, RUP e as orientações do guia de melhores práticas de gerenciamento de projetos denominado PMBOK. Todos estes processos foram criados para

atender principalmente empresas profissionais. Por isso, o BOPE adaptou essas práticas para a realidade do iMobilis.

Para guiar essa adaptação, a gamificação foi implantada no processo de *software* em um primeiro momento. A coleta de dados era feita de maneira manual através de planilhas. Isso era bastante custoso e mais propenso a erros. Para resolver esta questão, a gamificação foi mantida e um sistema *web* construído para organizar este acompanhamento e manter um registro de informações ao longo do tempo.

1.2 Objetivos

Esta seção descreve o objetivo geral e os objetivos específicos deste trabalho.

1.2.1 Objetivo geral

Este trabalho teve como principal objetivo definir, construir e implantar um sistema *web* para o projeto de inserção da gamificação no processo de *software* do laboratório de computação móvel e sistemas embarcados, adequado a boas práticas de engenharia de *software*.

Para que o sistema alcançasse seus objetivos, foi necessário que ele fosse desenvolvido de acordo com os elementos da gamificação que facilitam o monitoramento das atividades desenvolvidas pelos colaboradores do laboratório iMobilis durante a execução dos seus projetos.

1.2.2 Objetivos específicos

Os objetivos específicos do trabalho foram:

- I. Manter e melhorar a implantação da gamificação do BOPE no iMobilis.
- II. Aplicar técnicas de engenharia de *software* no desenvolvimento do sistema.
- III. Implantar o sistema BOPE Game.

1.3 Justificativa

Para garantir o seguimento do processo de *software* BOPE, é necessário o engajamento, colaboração e motivação dos colaboradores, portanto, são justificativas plausíveis a manutenção da gamificação e a construção de um sistema *web* para gerenciamento da gamificação.

1.4 Estrutura do trabalho

Este trabalho está estruturado em mais cinco seções: revisão bibliográfica, metodologia, apresentação e análises dos resultados, conclusão e limitações e trabalhos futuros.

A Seção 2, revisão bibliográfica, apresenta duas subseções. A primeira aborda os conceitos básicos para o entendimento deste trabalho, como definição de gamificação, definições das tecnologias utilizadas no desenvolvimento do *software*. A segunda apresenta os trabalhos correlatos a este estudo.

A Seção 3, metodologia, apresenta as etapas realizadas na execução do trabalho. Os principais procedimentos metodológicos utilizados para implementação do *software* também são relatados.

A Seção 4, uma apresentação e análise dos resultados é descrita. São realizadas verificações dos resultados obtidos com a implantação do sistema no laboratório.

A Seção 5, conclusão, apresenta as considerações finais acerca do trabalho realizado.

Por fim, a seção 6, limitações e trabalhos futuros, apresenta as limitações encontradas durante o desenvolvimento do trabalho e as recomendações para trabalhos futuros.

Parte I

Revisão Bibliográfica

2 Revisão Bibliográfica

Este capítulo aborda inicialmente todos os conceitos básicos para o entendimento do trabalho que foi desenvolvido. A seguir, são relatados os trabalhos correlatos a esta pesquisa.

2.1 Conceitos básicos

É apresentada nesta seção um levantamento dos conceitos de gamificação e seus elementos bem como as tecnologias que foram utilizadas no desenvolvimento deste trabalho.

2.1.1 Gamificação

Gamificação é a tradução literal do termo em inglês *Gamification*. Para Kumar e Herger (2013), a gamificação corresponde ao uso de mecanismos de jogos orientados ao objetivo de resolver problemas práticos ou de despertar engajamento entre um público específico. A gamificação é a utilização de conceitos de *game design* para motivar indivíduos a atingirem um objetivo. *Game design* é o processo de planejar o conteúdo e as regras de um jogo. [WERBACH, 2015].

No contexto da gamificação encontraremos as mecânicas de jogos que ajudarão a manter a motivação do colaborador através de estímulos. Vianna *et al.* (2013) evidenciam que para se alcançar esse objetivo na construção de qualquer artefato é preciso apropriar-se dos elementos mais eficientes de um jogo – Mecânicas, Dinâmicas e Estética – para a criação e adaptação das experiências do indivíduo. Segundo Zichermann e Cunningham (2011), a mecânica de um sistema de jogo é composta por várias ferramentas que tem a capacidade de produzir respostas estéticas significativas aos jogadores.

De modo geral, podemos construir sistemas gamificados baseados em elementos de jogos como pontos, medalhas, tabelas de líderes e recompensas. Outro elemento encontrado é o *feedback*, que apresenta dados e resultados aos participantes. De acordo com Werbach (2015), estes elementos exercem grande influência sobre os jogadores e são as formas mais comuns de se recompensarem os jogadores em qualquer jogo ou sistema gamificado. Quanto à escolha dos elementos, e de como aplicá-los em um contexto específico, dependerá da finalidade do projeto.

Para produzir produtos de *software* com qualidade e que atendem as necessidades e aos prazos estabelecidos pelos clientes, é imprescindível ter qualidade no processo que conduz sua construção.

Segundo Sommerville (2011), a qualidade do *software* pode ser classificada em qualidade de produto e qualidade de processo. Para garantir a qualidade de um determinado produto este deve ser construído guiado por bom processo.

Segundo Pressman (2016), processos de *software* englobam metodologias, ações e tarefas necessárias para o desenvolvimento e manutenção de um *software* de alta qualidade. A engenharia de *software* é realizada por pessoas criativas e que possuem um bom conhecimento para compreenderem tecnologias e ferramentas automatizadas que constituem o processo.

Para definir e obter melhorias nos processos de *software* surgiu a necessidade de impulsionar o comportamento humano, buscar por aperfeiçoamento no envolvimento, na motivação e no desempenho do indivíduo durante a realização de tarefas do processo de gerenciamento ágil. Em ambientes de desenvolvimento de *software*, pode se aplicar o modelo de Fogg (2009) que aponta as mudanças de comportamento humano, é o *Fogg Behavior Model* (FBM). O FBM apresenta um conjunto de três fatores que influenciam o comportamento humano: motivação, habilidade e incentivadores. Para estimular o comportamento humano, é necessário que a motivação seja suficiente. Garantir a motivação e o engajamento dos colaboradores através da gamificação colabora com ações de melhoria de processos de *software*. [CAVALCANTE et al. 2015].

Segundo Marins (2013), a gamificação tem se destacado por suas vantagens, tais como aprimoramento do aprendizado, colaboração e engajamento entre membros de equipes e mudanças de comportamento individuais.

2.1.2 A linguagem *Ruby*

*Ruby*¹ é uma linguagem de programação, originalmente planejada e desenvolvida no Japão no ano de 1993, pelo seu criador Yukihiro Matsumoto, conhecido como Matz. *Ruby* foi apresentada ao público pela primeira vez em 1995. É uma linguagem de programação interpretada multiparadigma, ou seja, suporta vários paradigmas de programação, tais como: programação funcional, orientada a objetos, imperativa e reflexiva.

É uma linguagem de *script*, de código aberto (*open source*) com prioridade na produtividade e simplicidade, com tipagem forte e dinâmica, ou seja, não é necessário declarar os tipos dos objetos e variáveis. Foi desenvolvida com o objetivo de apresentar uma linguagem legível, auto documentada com intuito de facilitar o desenvolvimento e manutenção de sistemas, melhorando a produtividade, pois se escreve menos documentação separada do projeto. Por ser uma linguagem interpretada, a *Ruby* necessita da instalação de um interpretador em sua máquina antes de executar algum programa.

A linguagem *Ruby* foi escolhida para o desenvolvimento do sistema por ser uma

¹ <<https://www.ruby-lang.org/pt/>>

das linguagens utilizadas para desenvolver rapidamente aplicações *web*, por apresentar um código simples, flexível e por mostrar alta relevância e demanda no mercado. Em *Ruby* simplificam-se muitas estruturas que em outras linguagens são muito complexas, uma característica destacável da linguagem é o fácil aprendizado. A linguagem oferece uma maneira simples de gerar um esqueleto do projeto, permite a utilização de *plug-ins* com reaproveitamento de código de qualidade.

2.1.3 O framework *Ruby on Rails*

*Ruby on Rails*² (ou também conhecido como apenas “Rails” ou “RoR”) é um *framework open source* para desenvolvimento de aplicações *web* escrito na linguagem de programação *Ruby*. Foi criado em 2003 por David Heinemeier Hansson (ou “DHH”) e extraído de um produto de sua empresa, o BasecampR.

A primeira versão foi apresentada ao público pela primeira vez em 2005. Desde então, ficou muito conhecido, levando também a linguagem *Ruby* ao mundo todo, anteriormente apenas conhecido no Japão e em alguns lugares dos Estados Unidos.

Uma das suas características destacáveis é a facilidade de se desenvolver códigos complexos usando as *Gems* que são bibliotecas ou conjunto de arquivos *Ruby* reutilizáveis. *Rails* possui *frameworks* de testes e seu desenvolvimento tem sido focado em testes.

O *rails* mantém dois princípios de filosofia de *software*: o primeiro é *Don't Repeat Yourself* (DRY - Não se repita). Estimula o princípio de não repetição do código. E o segundo é *Convention Over Configuration* (Coc - Convenção sobre Configuração) *Rails* tem uma boa maneira de padronizar em uma aplicação *web*, garantindo uma fácil configuração do ambiente de trabalho.

As aplicações criadas utilizando o *framework Rails* são desenvolvidas com base no padrão de arquitetura MVC (*Model-View-Controller*). O *Model-View-Controller* – Modelo Visão Controlador, é um padrão de arquitetura de *software*, responsável por fazer a separação da apresentação e a interação dos dados em um sistema.

O *framework Ruby on Rails* foi escolhido para o desenvolvimento do sistema por apresentar uma comunidade ativa que permite manutenção e inovação do *framework*. *Ruby on Rails* segue os conceitos do desenvolvimento ágil que permite um desenvolvimento rápido do *software* e com qualidade.

2.1.4 *RailsAdmin*

O *RailsAdmin*³ é um mecanismo *Rails* que fornece uma *interface* fácil de gerenciamento dos dados. O *RailsAdmin* foi escolhido por ser um recurso comum adicionado aos

² <<https://www.ruby-lang.org/pt/>>

³ <https://github.com/sferik/rails_admin>

aplicativos *Rails*, apresenta um painel de controle de administração de *backend*. A *gem Rails Admin* possui um impressionante conjunto de recursos e, para aplicativos básicos, fornece uma interface onde os usuários podem:

- I. Consultar facilmente o banco de dados.
- II. Exibir tabelas de banco de dados.
- III. Criar novos dados.
- IV. Atualizar facilmente os dados.
- V. Excluir dados com segurança.
- VI. Validação automática de formulários.
- VII. Pesquisa e filtragem.

2.1.5 Utilização do Gerenciador de Banco de Dados *PostgreSQL*

O *PostgreSQL*⁴ é um sistema gerenciador de banco de dados (SGBD) que tem a função de criar e manter um banco de dados. O *PostgreSQL* está entre os mais utilizados e populares sistemas de banco de dados com código fonte aberto.

Segundo Worsley *et al.* (2003), *PostgreSQL* é um *Object-Relational Database Management System* (ORDBMS), ou seja, um Sistema Gerenciador de Base de Dados Objeto Relacional. Possui suporte aos seguintes recursos: consultas complexas, chaves estrangeiras, integridade transacional, controle de concorrência, suporte ao modelo objeto relacional e *views*.

O *postgresql* foi escolhido por ter a capacidade de lidar com aplicações que gerenciam uma quantidade maior de dados e por fornecer a estrutura necessária para alta complexidade com um número cada vez maior de informações.

2.2 Trabalhos Correlatos

Estudos recentes evidenciam que aplicação da gamificação no campo de desenvolvimento de *software* vem sendo abordado cada vez mais.

Cavalcante *et al.* (2014) com o objetivo de melhorar o processo, o desempenho, a capacitação técnica dos colaboradores e a qualidade do produto aplicaram a gamificação em uma empresa de soluções em tecnologia. O estudo definiu uma estratégia de competição incluindo todos os projetos da organização e critérios de avaliação. A correlação com este trabalho é que o estudo também utilizou de um indicador de porcentagem (%) de não conformidades (NC) e porcentagem (%) de colaboradores treinados. Os dados foram

⁴ <<https://www.postgresql.org/>>

coletados antes e depois da aplicação, o resultado obtido foi um aumento da aderência dos projetos aos processos da organização e a campanha motivacional. A capacitação dos colaboradores foi satisfatória, ressaltando que as técnicas da gamificação foram eficientes e aumentaram o comprometimento dos colaboradores.

Ng (2014) constatou que equipes de desenvolvimento de *software* possuem o conhecimento necessário para identificar problemas, definir soluções e possuem habilidades em tecnologia de *software*, porém não apresentam o treinamento adequado para melhorar seus próprios processos. Para alcançar uma melhoria no processo de *software*, é necessária a participação das equipes e para incentivá-las foi introduzido o uso de dinâmicas de jogos, através do *framework* Essence. O *framework* ajuda na construção de processos de desenvolvimento de *software* na empresa de dispositivos. O trabalho correlato apresentou o uso de Alphas que são objetos que dão suporte na identificação de requisitos, na construção da equipe, na definição do modo de trabalho e nas etapas do desenvolvimento de *software*. O *workshop* realizado com equipes estabeleceu elementos da gamificação para definir tarefas a alcançar os objetivos e adotou como métodos a realização de reuniões diárias, o desenvolvimento iterativo baseados nas metodologias ágeis.

Cota (2016) realizou um estudo para implantação da gamificação em um laboratório de pesquisa. Foi analisado o processo de *software* utilizado no laboratório para auxiliar na identificação dos elementos da gamificação que poderiam ser aplicados. Dados foram coletados antes e depois da implantação da gamificação através de questionários onde foi detectada uma insatisfação por partes dos colaboradores que não se sentiam recompensados diante do esforço empreendido nas realizações de suas tarefas. Métricas foram definidas para avaliar o progresso da implantação da gamificação e um sistema de pontuação que possuía as atividades a serem avaliadas com seu respectivo ponto. Os resultados alcançados mostraram que a produtividade do laboratório aumentou, assim como o comprometimento dos colaboradores no seguimento do processo.

Pedreira *et al.* (2014) realizaram um estudo baseado em trabalhos que utilizaram dos elementos da gamificação em áreas específicas da engenharia de *software*, como gerenciamento ágil de projetos e gerenciamento de riscos. A análise realizada sobre os trabalhos, concluiu que os estudos encontram-se em fase introdutória e não realizam uma avaliação antes da implantação da gamificação para basear os resultados coletados após a implantação. Os trabalhos são desprovidos de etapas de identificação dos objetivos e dos perfis dos usuários que ajudam a estabelecer os possíveis elementos da gamificação.

Parte II

Metodologia

3 Metodologia

Neste capítulo, é descrita a metodologia utilizada no desenvolvimento do sistema *web* BOPE Game.

O trabalho descreve o planejamento do sistema em todas suas fases, como treinamento na linguagem escolhida, identificação dos requisitos do sistema, criação das histórias de usuário e dos cenários de teste, prototipação, modelagem do sistema, implementação, testes e implantação.

3.1 Concepção e Elaboração

A concepção do trabalho se iniciou pelo treinamento na linguagem *Ruby* que proporcionou conhecimentos necessários básicos para desenvolvimento do sistema e o entendimento da política da gamificação utilizada no BOPE. Posteriormente, na elaboração, foi realizada a especificação do projeto definido o escopo, os objetivos principais do sistema e a especificação dos requisitos.

3.1.1 *Product Backlog*

O *product backlog* descreve em alto nível o escopo do *software*. Pode-se notar na Tabela 1 a identificação dos serviços que o sistema oferece. O *software* foi decomposto em oito componentes, nos quais cada um exerce um tipo de serviço, apresentando as funcionalidades identificadas. Uma funcionalidade corresponde a um item do *product backlog*.

Tabela 1 – *Product Backlog*

SERVIÇOS	REQUISITOS ESPECÍFICOS
Serviço de Autenticação	Autenticar admin Autenticar usuário
Serviço de Visualização de Informações	Pontuação dos usuários de cada semana Pontuação final dos usuários de cada <i>sprint</i> Pontuação das equipes de cada semana Pontuação final das equipes de cada <i>sprint</i> Visualizar tela de medalhas Visualizar usuários Visualizar equipes Visualizar integrantes da equipe Visualizar apostas do <i>sprint</i> Associar ponto de usuário à equipe
Serviço de Geração de Relatórios	Gerar o <i>ranking</i> da semana individual Gerar o <i>ranking</i> da <i>sprint</i> individual Gerar o <i>ranking</i> da semana da equipe Gerar o <i>ranking</i> da <i>sprint</i> da equipe Gerar um gráfico com não conformidades
Serviço de Gerenciamento de <i>Sprint</i>	Adicionar <i>sprint</i> Excluir <i>sprint</i> Adicionar semana Excluir semana
Serviço de Gerenciamento de Usuário	Cadastrar novo usuário Editar perfil de usuário Excluir usuário Pesquisar usuário Criar equipe Excluir equipe Adicionar usuário a equipe Excluir usuário da equipe Admin trocar senha de usuário
Serviço de Atividades	Cadastrar as atividades que serão avaliadas Editar as atividades que serão avaliadas Excluir as atividades que serão avaliadas Inserir pontuação nas atividades Editar as pontuações nas atividades
Serviço de Não Conformidades	Verificar as atividades que não tiveram pontos atribuídos
Serviço de Gerenciamento de Perfil	Colocar <i>email</i> e foto de cada usuário Trocar senha

3.1.2 Histórias de Usuários

Foram criadas histórias de usuários para cada item do *backlog* que auxiliaram no desenvolvimento e indicaram como o sistema deve ser utilizado. A estrutura utilizada para descrevê-las é encontrada em metodologias ágeis que usam de linguagem natural e palavras-chave para um melhor direcionamento do objetivo. Foram especificadas de acordo com o princípio SMART:

- S (*Specific*) cada história deve ser independente e referir-se somente a uma funcionalidade do sistema.
- M (*Measurable*) cada história deve ser medida.
- A (*Achievable*) cada história deve ser implementada em uma iteração.
- R (*Relevant*) cada história deve ser realmente uma funcionalidade importante para o sistema.
- T (*Timeboxed*) cada história não deve se exceder em uma iteração e se necessário ser dividida em mais de uma história.

A estrutura utilizada na criação das histórias para definir e organizar os requisitos do sistema deve especificar o papel, a funcionalidade e o objetivo a ser alcançado, como o exemplo abaixo:

- COMO UM (papel)
- EU GOSTARIA (funcionalidade)
- PARA CONSEGUIR (objetivo)

Abaixo são apresentadas as histórias de usuários para as funcionalidades do serviço de “Geração de Relatórios”. As histórias de todos os serviços foram escritas e documentadas em uma ferramenta utilizada pelo laboratório iMobilis, denominada *Dokuwiki* e estão no Anexo A.

Serviço de Geração de Relatórios

- **Gerar o *ranking* da semana individual**

COMO UM usuário administrador

EU GOSTARIA DE obter o *ranking* semanal individual

PARA CONSEGUIR saber qual colaborador atingiu maior pontuação da semana no sistema

- **Gerar o *ranking* da *sprint* individual**

COMO um usuário administrador

EU GOSTARIA DE obter o *ranking* da *sprint* individual

PARA CONSEGUIR saber qual colaborador atingiu maior pontuação da *sprint* no sistema

- **Gerar o *ranking* da semana da equipe**

COMO um usuário administrador

EU GOSTARIA DE obter o *ranking* semanal da equipe

PARA CONSEGUIR saber qual equipe atingiu maior pontuação da semana no sistema

- **Gerar o *ranking* da *sprint* da equipe**

COMO um usuário administrador

EU GOSTARIA DE obter o *ranking* da *sprint* da equipe

PARA CONSEGUIR saber qual equipe atingiu maior pontuação da *sprint* no sistema

- **Gerar um gráfico que apresenta as não conformidades individuais e de equipe em cada semana**

COMO um usuário administrador

EU GOSTARIA DE gerar um gráfico que apresenta as não conformidades individuais e de equipe em cada semana

PARA CONSEGUIR saber qual colaborador e qual equipe apresentaram mais não conformidades no sistema

3.1.3 Cenários de Teste

Para cada item do *backlog* foram criados os critérios de aceitação descritos em cenários de teste. Os critérios de aceitação foram definidos para que cada história corresponda às funcionalidades necessárias do sistema e o mesmo diz respeito se o *software* comporta adequadamente.

Os critérios de aceitação foram descritos em termos de cenários utilizando um padrão de definição, estabelecendo seus cenários de testes principais e alternativos.

A estrutura utilizada na criação dos cenários de teste deve especificar a pré-condição, o evento e a pós-condição, como o exemplo abaixo:

- DADO (pré-condição)
- QUANDO (evento)
- ENTÃO (pós-condição)

Abaixo é apresentado os cenários de teste para cada item do serviço de “Geração de

Relatórios”. Os cenários de teste foram escritos e documentados na *Dokuwiki* do iMobilis e estão no Anexo B.

Serviço de Geração de Relatórios

- **História de usuário: Gerar o *ranking* da semana individual**

Cenário: Principal – Usuário acessa página de *ranking*

DADO que um usuário acesse a página de *ranking*

QUANDO o botão “Gerar *ranking* da semana individual” for pressionado

ENTÃO o sistema deve exibir na tela os dados sobre o *ranking*

MAS os dados podem não estar disponíveis e uma mensagem “Dados não disponíveis” deve ser exibida

- **História de usuário: Gerar o *ranking* da *sprint* individual**

Cenário: Principal – Usuário acessa página de *ranking*

DADO que um usuário acesse a página de *ranking*

QUANDO o botão “Gerar *ranking* do *sprint* individual” for pressionado

ENTÃO o sistema deve exibir na tela os dados sobre o *ranking*

MAS os dados podem não estar disponíveis e uma mensagem “Dados não disponíveis” deve ser exibida

- **História de usuário: Gerar o *ranking* da semana da equipe**

Cenário: Principal – Usuário acessa página de *ranking*

DADO que um usuário acesse a página de *ranking*

QUANDO o botão “Gerar *ranking* da semana da equipe” for pressionado

ENTÃO o sistema deve exibir na tela os dados sobre o *ranking*

MAS os dados podem não estar disponíveis e uma mensagem “Dados não disponíveis” deve ser exibida

- **História de usuário: Gerar o *ranking* da *sprint* da equipe**

Cenário: Principal – Usuário acessa página de *ranking*

DADO que um usuário acesse a página de *ranking*

QUANDO o botão “Gerar *ranking* do *sprint* da equipe” for pressionado

ENTÃO o sistema deve exibir na tela os dados sobre o *ranking*

MAS os dados podem não estar disponíveis e uma mensagem “Dados não disponíveis” deve ser exibida

- **História de usuário: Gerar um gráfico que apresenta as não conformidades individuais e de equipe em cada semana**

Cenário: Principal – Usuário acessa página de relatório

DADO que um usuário acesse a página de relatório

QUANDO o botão “Gerar gráfico das não conformidades” for pressionado

ENTÃO o sistema deve exibir na tela os dados sobre as não conformidades em cada semana

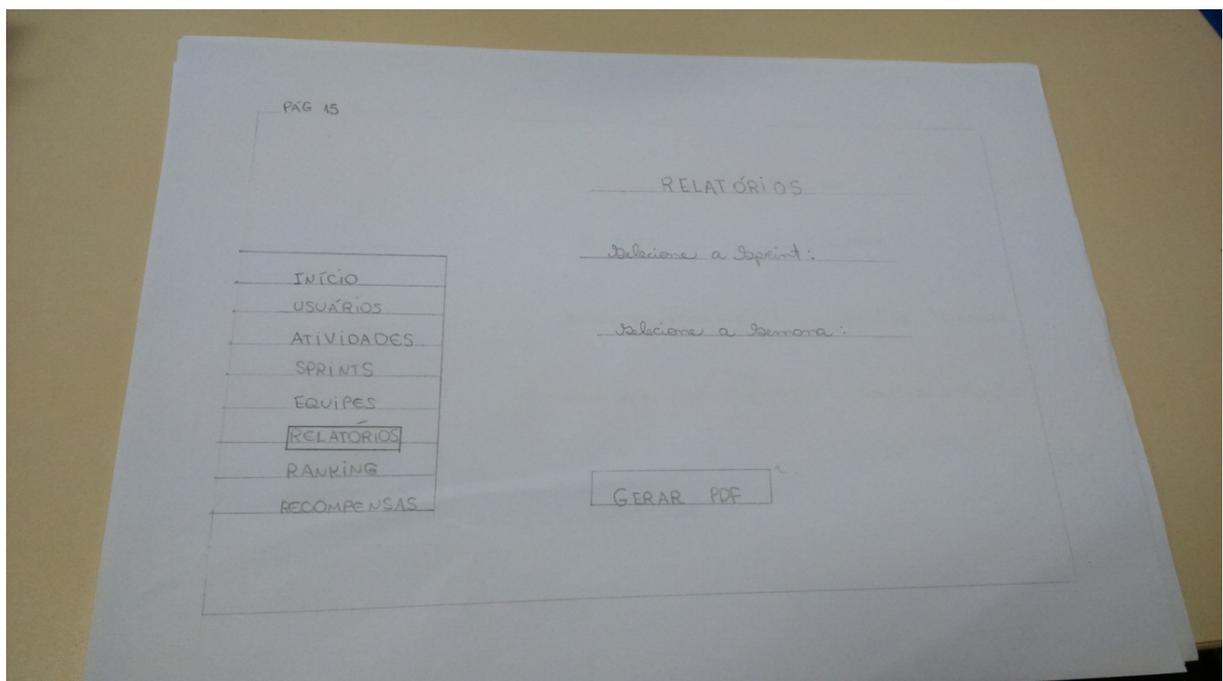
MAS os dados podem não estar disponíveis e uma mensagem “Dados não disponíveis” deve ser exibida

3.1.4 Protótipos de Tela

Em conjunto com a definição dos serviços, histórias de usuários e cenários de teste, foram criados protótipos de papel das telas do sistema, com objetivo de definir e experimentar a *interface*. Um exemplo deles pode ser observado na Figura 3. Estes protótipos foram criados em paralelo com as atividades anteriores, pois ajudaram na identificação de funcionalidades. Todos os protótipos de papel estão documentados na *Dokuwiki* do iMobilis.

A Figura 1 apresenta a tela de “Geração de Relatórios” onde o usuário administrador pode selecionar a semana ou *sprint* que deseja visualizar o PDF gerado com o *ranking* individual e por equipe.

Figura 1 – Protótipo de papel da tela Geração de Relatórios



3.1.5 Projeto de Banco de Dados

O Modelo Entidade-Relacionamento (ER) é uma representação gráfica do banco de dados, que apresenta as entidades do sistema, os atributos e os relacionamentos entre eles.

A geração do modelo entidade-relacionamento em aplicações *Rails* é realizada pela *gem rails-erd*. Essa *gem* permite gerar facilmente um modelo baseado nas tabelas do sistema, que fornece uma visão geral de como as tabelas estão relacionadas. O objetivo da *gem rails-erd* é fornecer uma ferramenta para inspecionar como as tabelas estão associadas no sistema. O modelo entidade-relacionamento do BOPE Game pode ser visualizado no Anexo C.

3.2 Implementação

Para o desenvolvimento do sistema *web*, foram adotadas as versões 2.3 do *Ruby* e 5.0 do *Rails* e a utilização do banco de dados *PostgreSQL* na versão 9.4. O sistema foi desenvolvido a partir de uma distribuição *Linux*, o *Ubuntu* 14.04 e para desenvolvimento do código foi utilizado o editor de texto *ATOM* ¹.

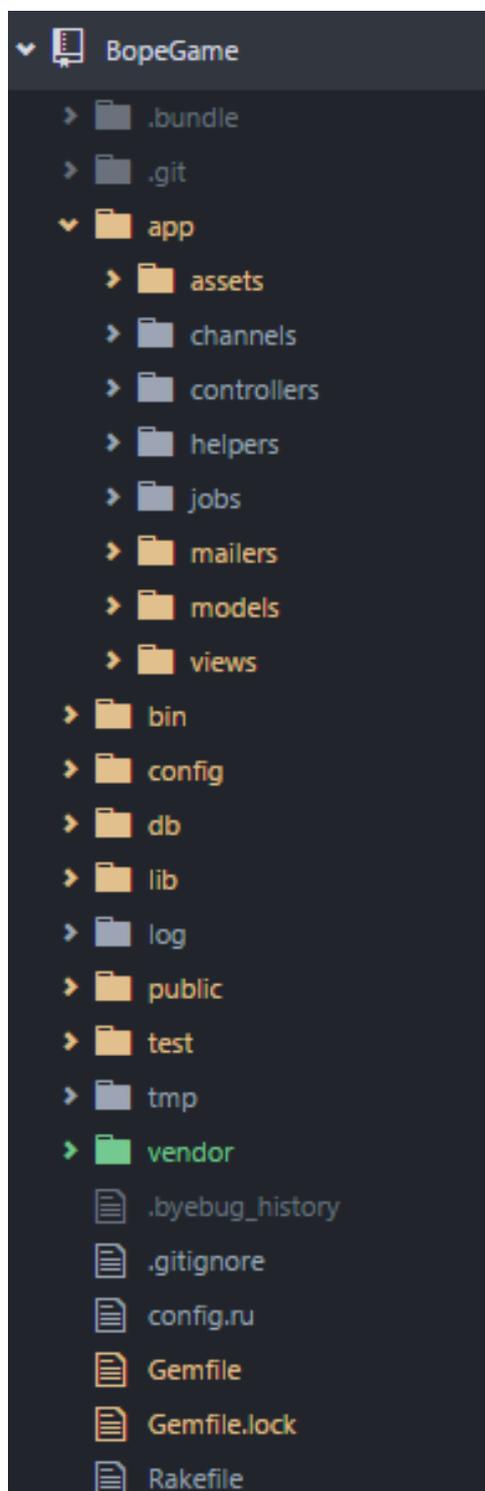
3.2.1 Criação do Projeto

Após toda instalação necessária para desenvolver um sistema em *Rails*, foi necessário digitar o comando para criação do projeto BOPE Game.

```
$ rails new bopegame
```

Depois de executado o comando mencionado acima, toda a estrutura padrão da aplicação *Ruby on Rails* foi criada. No diretório do projeto existem vários arquivos e pastas geradas automaticamente, conforme Figura 2.

¹ <<https://atom.io/>>

Figura 2 – Estrutura dos arquivos e pastas gerados por padrão no *Rails*

A seguir segue um resumo da estrutura de pastas:

- *app/*: Contém os *controllers*, *models*, *views*, *helpers*, *mailers* e *assets* (onde ficam armazenados os arquivos CSS e *JavaScript*) da aplicação.
- *bin/*: Contém os *scripts* que inicializam a aplicação.
- *config/*: Contém os arquivos raízes de configuração do projeto como:
application (configurações gerais da aplicação);
routes(rotas da aplicação);
database(configuração do banco de dados);
environments (ambiente: produção, desenvolvimento e teste);
initializers (arquivos de inicialização);
locales (arquivos de internacionalização da língua Português).
- *db/* - Contém o esquema (*schema*) e as migrações (*migrations*) do banco de dados do sistema.
- *Gemfile*: Neste arquivo foi especificado as bibliotecas (*gem*) que foram necessárias para o desenvolvimento do sistema.
- *lib/* - Arquivos que foram estendidos para a geração de PDF devido a utilização do *RailsAdmin*.
- *log/* - Arquivos de log.
- *public/* - Contém arquivos para armazenar todas as imagens do projeto, como logo do iMobilis e logo BOPE Game.
- *test/* - A pasta padrão para os testes unitários do *Rails*.
- *vendor/assets/* - A pasta onde fica toda a pré-compilação dos *assets* do *Rails*.

Para configurar o banco de dados *PostgreSQL*, foi necessário abrir o arquivo *config/database.yml* e escrever o nome de usuário do banco, o nome do banco, a senha do usuário como segue na Figura 3:

Figura 3 – Configuração do Banco de Dados *PostgreSQL*

```
database.yml
1
2  development:
3    adapter: postgresql
4    encoding: unicode
5    database: bopegame_development
6    pool: 5
7    username: geovana
8    password: 123456
9
10
11  test:
12    adapter: postgresql
13    host: localhost
14    encoding: unicode
15    database: bopegame_test
16    pool: 5
17    username: geovana
18    password: 123456
19
20  production:
21    adapter: postgresql
22    host: localhost
23    encoding: unicode
24    database: bopegame_production
25    pool: 5
26    username: geovana
27    password: 123456
28
```

Após a configuração do banco de dados foi necessário os seguintes comandos para apagar todas as entradas existentes, criar, migrar as informações no banco de dados e inicializar com um usuário padrão registrado no arquivo *seed*:

```
$ rake db:drop
$ rake db:create
$ rake db:migration
$ rake db:seed
```

Para criar cada tabela no banco de dados foi necessário um comando com informações de cada tabela. Com esse comando, o *Rails* cria um modelo *user*, com o atributo *name* do tipo *string*. Esse atributo foi automaticamente adicionado à tabela de *user* na base de dados.

```
$ rails g model user name:string
```

O gerador cria automaticamente um novo arquivo de modelo no diretório *app/models*, uma unidade de teste para testar a funcionalidade do seu modelo e uma migração (*migrate*) (Figura 4) para o banco de dados.

Para inserção de novos atributos foi utilizado um comando que insere quantos atributos necessários foram identificados durante o desenvolvimento. Como exemplo, inserção do campo *status* a tabela usuário. Para exclusão de alguma variável apenas trocar a palavra *Add* por *Remove*.

```
$ rails g migration AddStatusToUser status:string
```

Esses comandos foram utilizados para criação de todas as tabelas do sistema que representam os Serviços de Gerenciamento de Usuários, Gerenciamento de Perfil, Serviço de Atividades e o Serviço de Gerenciamento de *sprint*, através da criação do CRUD - *Create*, *Read*, *Update* e *Delete*, termo usado para designar as quatro operações em um banco de dados do sistema. Por utilizar o mecanismo *RailsAdmin* não foi necessária a criação de *controllers* e *views*.

Figura 4 – Exemplo de um arquivo de migração (*migrate*)

```
20170204012451_create_activity.rb
1  class CreateActivity < ActiveRecord::Migration[5.0]
2    def change
3      create_table :activities do |t|
4        t.string :name
5        t.decimal :punctuation
6
7        t.timestamps
8      end
9    end
10  end
11
```

3.2.2 Autenticação e Autorização no sistema

Para realizar a autenticação de usuários no sistema foi utilizado a *gem devise* do *Ruby on Rails* que permite a implementar o *login*, o *logout* e o registro de usuários facilmente. A mesma é composta pelos seguintes módulos:

- *Database Authenticatable* (Banco de dados autenticável): Permite armazenar no banco de dados uma senha de cada usuário para validar a autenticidade ao fazer *login* no sistema.
- *Omniauthable*: Permite realizar autenticação utilizando mecanismos como *Facebook*, *Twitter*, etc.
- *Confirmable* (Confirmável): Permite enviar *e-mails* de confirmação de registro no sistema.
- *Recoverable* (Recuperável): Permite redefinir a senha do usuário.
- *Registerable* (Registável): Permite criar, editar e excluir a conta do usuário através de um registro.
- *Rememberable* (Recordável): Permite salvar senhas de *login*.
- *Timeoutable*: Permite expirar sessões inativas por determinado tempo.
- *Validatable* (Validação): Permite o fornecimento de validações de *e-mail* e senha.
- *Lockable*: Permite o bloqueio de uma conta após tentativas de *login* falhadas.

No sistema BOPE Game, foi ativado o módulo que permite que as senhas dos usuários sejam gravadas no banco de dados e as mesmas possam ser lembradas na realização de *login*. O administrador do sistema é responsável por registrar os usuários, atualizar e realizar a remoção do cadastro se necessário. A Figura 5 representa a tabela de usuário implementada com o *devise*.

Figura 5 – Autenticação dos usuários

```
user.rb
1 class User < ApplicationRecord
2   enum role: [:Acesso_Completo, :Acesso_Restrito]
3   enum status: [:Ativo, :Inativo]
4   # Include default devise modules. Others available are:
5   # :confirmable, :lockable, :timeoutable and :omniauthable
6   devise :database_authenticatable, :registerable,
7         :recoverable, :rememberable, :trackable, :validatable
8
9   has_attached_file :perfil, styles: { thumb: "100x100>" }
10  validates_attachment_content_type :perfil, content_type: /\Aimage\/.*\z/
11
12  has_many :ranking_individuals_launches
13  has_many :ranking_sprint_individual_launches
14  belongs_to :team
15  belongs_to :individual_point_launch
16 end
17
```

A autorização no sistema BOPE Game consiste no processo de gerenciar quais recursos o usuário cadastrado poderá acessar dentro do sistema. Para realização da autorização foi utilizada a *gem Cancancan* que gerencia o acesso aos recursos através de um arquivo que limita o acesso a dados dependendo do tipo de usuário, se trata de um usuário que possui acesso restrito ou completo.

No BOPE Game, os usuários que possuem acesso restrito (*time* de desenvolvimento) podem apenas listar a página que contém os nomes de todos os usuários, as atividades, as *sprints*, as equipes cadastradas e visualizar a página de medalhas no sistema, mas não possuem permissão de inserir ou remover quaisquer dados no sistema podendo apenas atualizar as informações do próprio perfil. Apenas o administrador do sistema possui acesso completo e permissão de gerenciar dados da tabela. Na Figura 6 é representado a classe *ability* que contém as regras de acesso implementadas no sistema.

Figura 6 – Autorização dos usuários

```
1 class Ability
2   include CanCan::Ability
3
4   def initialize(user)
5     p user
6     if user
7       can :access, :rails_admin # grant access to rails_admin
8       can :dashboard # grant access to the dashboard
9       if user.role == "Acesso_Completo"
10        can :manage, :all
11      else
12        can :read, [User]
13        can :update, [User]
14        can :read, [Team]
15        can :read, [Sprint]
16        can :read, [Activity]
17        can :read, [RankingIndividualsLaunch]
18        can :pdf, [RankingIndividualsLaunch]
19        can :read, [RankingTeamLaunch]
20        can :pdf, [RankingTeamLaunch]
21        can :read, [RankingSprintIndividualLaunch]
22        can :pdf, [RankingSprintIndividualLaunch]
23        can :read, [RankingSprintTeamLaunch]
24        can :pdf, [RankingSprintTeamLaunch]
25        can :read, [Medal]
26      end
27    end
28  end
end
```

3.2.3 Associação de tabelas no sistema

Para relacionar diversos modelos (tabelas), foi necessário estabelecer o tipo de relacionamento entre eles. Foram elaborados alguns métodos para manipular os elementos envolvidos nos relacionamentos. Existem vários tipos de relacionamentos que o *Rails* disponibiliza, mas no desenvolvimento foram utilizados apenas os relacionamentos:

- *belongs_to* - usado quando um modelo tem como um de seus atributos o *id* de outro modelo, ou seja, um modelo pertence ao outro.
- *has_many* - O *has_many* é um relacionamento um-para-muitos (*has_many* significa, “tem muitos”).

3.2.4 Geração de PDF no sistema

Para geração de PDF foi necessário a utilização da *gem praw* que é uma biblioteca que oferece suporte ao *Ruby*. E para geração dos gráficos foi utilizado a *gem gruff*.

Os relatórios, arquivos, pdf, possuem o *ranking*, os gráficos com as não conformidades em cada semana e *sprint*.

3.2.5 Implantação do sistema no laboratório iMobilis

Foi desenvolvido um tutorial que contém os passos necessários executados na implantação do BOPE Game no servidor Mariana no laboratório iMobilis. O tutorial de implantação pode ser visualizado no Anexo D.

Parte III

Apresentação e Análise dos Resultados

4 Apresentação e Análise dos Resultados

Nesta seção, serão apresentados os métodos utilizados para coleta de dados antes e depois da implantação do sistema. Serão apresentados os dados coletados do monitoramento da gamificação durante o desenvolvimento do *software* no laboratório nos períodos letivos 16/01 e 16/02.

4.1 Métodos de coleta de dados da gamificação

O monitoramento da gamificação no laboratório durante o período letivo 16/01 foi realizado pela coordenadora do sistema de pontuação juntamente do gerente de projetos que foram responsáveis por monitorar, coletar e reportar os dados coletados das atividades exercidas pelos colaboradores.

No período da implantação da gamificação no processo de *software* BOPE, os dados coletados foram registrados em planilhas do programa *Microsoft Excel*, onde foi inserida a pontuação individual e por equipe obtida pelos colaboradores. Na Figura 7, é apresentado um modelo da planilha que possibilita visualizar o *ranking* individual e na Figura 8 o *ranking* por equipe, gerado a cada semana e *sprint*, além de apresentar quais as atividades foram avaliadas com a pontuação associada a elas.

Figura 7 – Planilha de inserção de pontuação individual da gamificação no período letivo 16/01

INDIVIDUAL	Dedicar no mínimo 100% por iteração	Dedicação igual ou acima de 75 % e abaixo 100%	Comparecer nas reuniões semanais	Comparecer nas reuniões técnicas e de validação	Logar corretamente no NetProject	Construir tutoriais publicáveis de novas tecnologias	Colaborar com outro membro oferecendo assistência	Total	RANKING	
									Posição	Nome
Colaborador 1	x	Não	x	x	x	Não	Não	6		
Colaborador 2	Não	x	x	x	Não	Não	Não	3	1	Colaborador 3
Colaborador 3	x	Não	x	x	x	x	x	10	2	Colaborador 5
Colaborador 4	Não	x	x	x	x	Não	Não	4	3	Colaborador 8
Colaborador 5	x	Não	x	x	x	Não	x	7	4	Colaborador 1
Colaborador 6	x	Não	x	x	Não	Não	Não	5	5	Colaborador 7
Colaborador 7	x	Não	x	x	x	Não	Não	6	6	Colaborador 11
Colaborador 8	Não	x	x	x	x	x	Não	7	7	Colaborador 6
Colaborador 9	Não	x	x	x	x	Não	x	5	8	Colaborador 9
Colaborador 10	Não	x	x	x	x	Não	Não	4	9	Colaborador 4
Colaborador 11	x	Não	x	x	x	Não	Não	6	10	Colaborador 10
PONTUAÇÃO									11	Colaborador 2
Colaborador 1	3	0	1	1	1	0	0			
Colaborador 2	0	1	1	1	0	0	0			
Colaborador 3	3	0	1	1	1	3	1			
Colaborador 4	0	1	1	1	1	0	0			
Colaborador 5	3	0	1	1	1	0	1			
Colaborador 6	3	0	1	1	0	0	0			
Colaborador 7	3	0	1	1	1	0	0			
Colaborador 8	0	1	1	1	1	3	0			
Colaborador 9	0	1	1	1	1	0	1			
Colaborador 10	0	1	1	1	1	0	0			
Colaborador 11	3	0	1	1	1	0	0			

Figura 8 – Planilha de inserção de pontuação por equipe da gamificação no período letivo 16/01

EQUIPES	Dedicar no mínimo 100% por iteração	Dedicação igual ou acima de 75 % e abaixo 100%	Todos os colaboradores participaram de todas as reuniões semanais	Todos os colaboradores participaram de todas as reuniões técnicas e de validação	Todos os colaboradores logaram corretamente no NetProject em todas as iterações	
Equipe 1	Não	x	x	Não	Não	
Equipe 2	Não	Não	Não	x	Não	
Equipe 3	Não	Não	Não	Não	Não	
PONTUAÇÃO						
Equipe 1	0	1	1	0	0	
Equipe 2	0	0	0	1	0	
Equipe 3	0	0	0	0	0	
RANKING		Membros Equipe				
Posição	Nome	Mob Mine	Mobile	Processos	TOTAL	Nome
1	Equipe 1	Colaborador 1	colaborador 2	Colaborador 3	2	Equipe 1
2	Equipe 2	Colaborador 5	colaborador 9	Colaborador 4	1	Equipe 2
3	Equipe 3	Colaborador 6		Colaborador 8	0	Equipe 3
		Colaborador 7				
		Colaborador 11				
		Colaborador 10				

Durante a semana de trabalho, a coordenadora e o gerente de projetos fazem uma supervisão de cada atividade que os colaboradores deveriam ter realizado. Foram definidos em conjunto com todos os envolvidos as atividades e seus critérios de pontuação individual e por equipe. Este conjunto pode ser observado na Tabela 2.

Tabela 2 – Atividade e critérios de pontuação

ATIVIDADES	PONTUAÇÃO
Dedicar no mínimo 100% de horas por semana	3 pontos
Dedicação igual ou acima de 75 % e abaixo 100% de horas	1 ponto
Comparecer nas reuniões semanais	1 ponto
Comparecer nas reuniões técnicas e de validação	1 ponto
Logar corretamente no <i>NetProject</i>	1 ponto
Construir tutoriais publicáveis de novas tecnologias	3 pontos
Colaborar com outro membro oferecendo assistência	1 ponto
Atualizar corretamente a <i>Dokuwiki</i> por <i>sprint</i>	1 ponto
Atualizar corretamente as atividades no <i>GitLAB</i> por <i>sprint</i>	1 ponto
Atualizar a completude do <i>NetProject</i> a cada validação por <i>sprint</i>	1 ponto
Importar o código-fonte do projeto no <i>GitLAB</i> no final do <i>sprint</i>	1 ponto
Realizar todas as atividades do projeto planejadas	3 pontos

O gerente de projeto coleta os dados da ferramenta de gestão de projetos utilizada no laboratório (*NetProject*), posteriormente reporta à coordenadora da gamificação quais os

colaboradores cumpriram com a atividade de dedicação mínima exigida e quais utilizaram a ferramenta de forma adequada registrando suas horas trabalhadas.

A atividade de comparecer a reunião técnica é uma atividade para esclarecer dúvidas a cerca dos projetos, acompanhar o andamento das atividades definidas do *sprint backlog* e entregar algum artefato para validação. A avaliação da participação da reunião técnica pode ser confirmada diante do professor orientador do projeto, se o colaborador cumpriu com essa atividade.

A coordenadora dos pontos visualiza durante a reunião semanal se todos os colaboradores encontram-se presentes. A construção de tutoriais é uma atividade não obrigatória, porém reflete valor aos colaboradores que realizam e esses devem publicar o tutorial construído na ferramenta de suporte *Dokuwiki* para pontuação na gamificação. O colaborador que possui algum conhecimento ou suporte técnico em uma tecnologia utilizada nos desenvolvimentos dos projetos e oferece assistência aos demais colaboradores é gratificado na gamificação, os colaboradores que recebem essa assistência devem reportar à coordenadora.

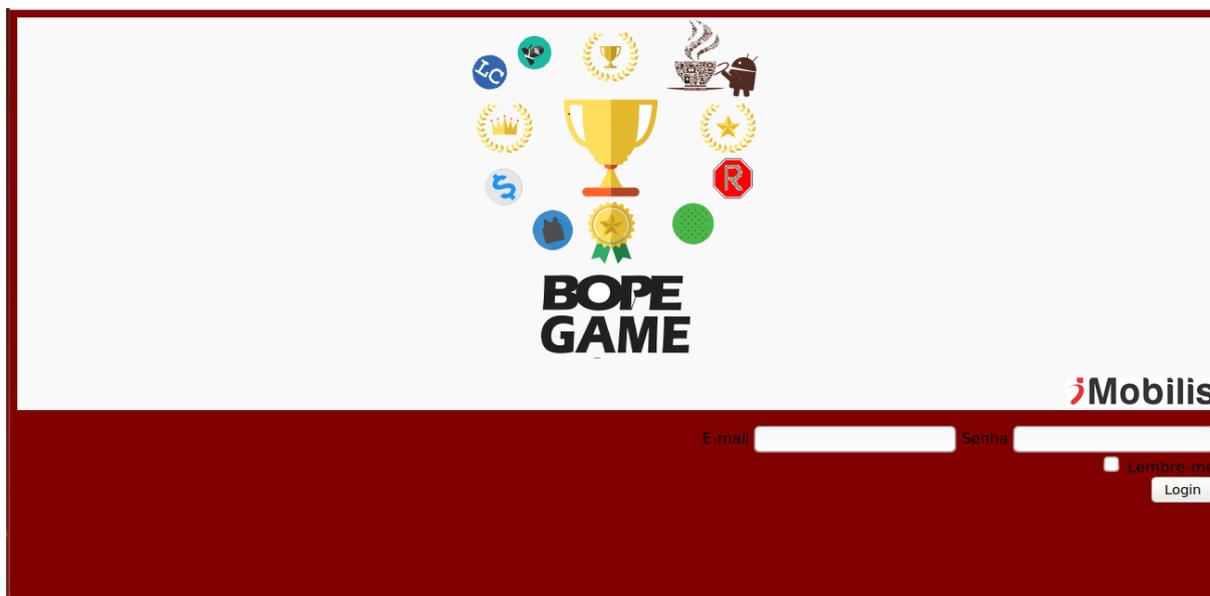
As atividades que foram definidas para serem avaliadas por *sprint*, são analisadas na última semana do *sprint*, onde é verificado se todos os colaboradores atualizaram corretamente seus projetos na *Dokuwiki* e na ferramenta de suporte *GitLab*. A atualização da completude dos entregáveis no *NetProject* a cada validação, deve ser feita com orientação do professor e verificada pelo gerente de projetos que remete a coordenadora da gamificação. Para manter uma versão segura dos códigos fontes dos projetos, é verificado se os mesmos foram importados no *GitLab*. Por fim, a última atividade analisada é a verificação de todas as tarefas estabelecidas no início do *sprint* se foram concluídas, esta avaliação é feita com base nos relatórios realizados pelo gerente de projeto. O resultado do *ranking* dos vencedores da gamificação é enviado por *email* e anexado nas paredes do laboratório.

4.2 Monitoramento da gamificação utilizando o sistema BOPE Game

A partir da conclusão do sistema BOPE Game, o monitoramento da gamificação passou a ser realizado no sistema facilitando a inserção da pontuação e a geração de relatórios com os resultados obtidos.

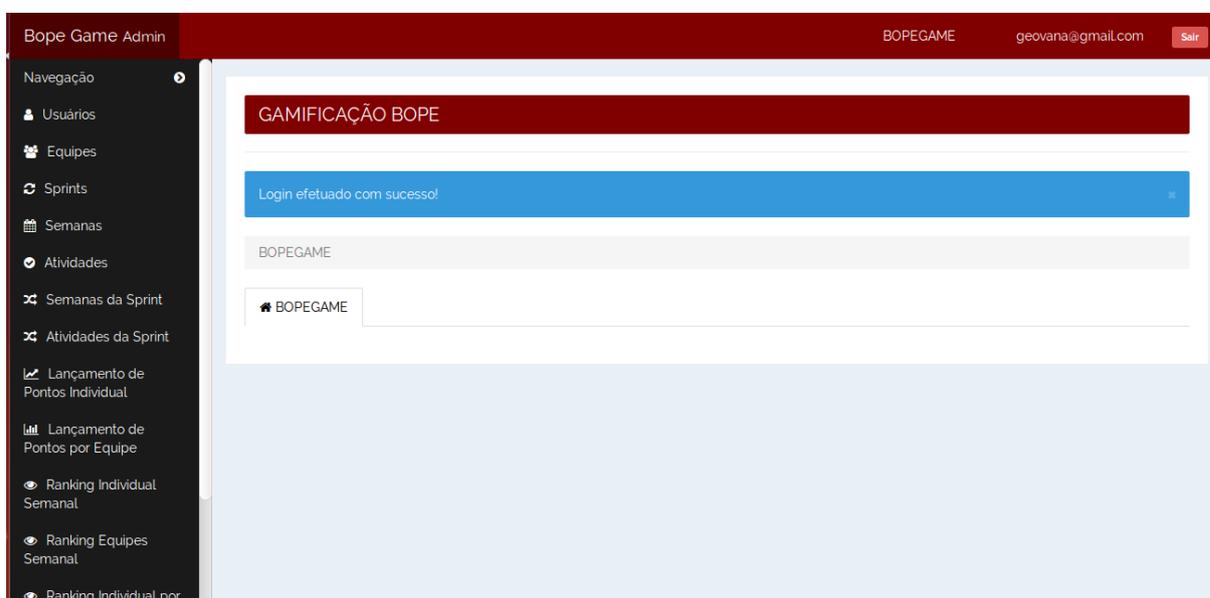
4.2.1 Acesso ao sistema

Para acessar o sistema, o usuário cadastrado deve informar seu *login* e senha conforme Figura 9.

Figura 9 – Tela de *login*

Após realizar a autenticação, o usuário terá acesso a página inicial do sistema, no qual poderá interagir com as funcionalidades implementadas através dos elementos apresentados no menu principal, obtendo acesso aos relatórios, visualização dos perfis de usuários e tela de medalhas conforme Figura 10.

Figura 10 – Tela Principal



4.2.2 Cadastros dos Usuários e das Atividades

Esta fase deu início ao cadastro de informações importantes no sistema. Primeiro foi realizado o cadastro de todos os usuários do sistema que poderão acompanhar o resultado da gamificação pelo sistema. A página de cadastro pode ser visualizada na Figura 11. Para

realização do mesmo é necessário o preenchimento dos dados necessários. E a página de cadastro de atividades pode ser visualizada na Figura 12.

Figura 11 – Página de cadastro dos usuários do sistema

Figura 12 – Atividades cadastradas no sistema

Nome			
<input type="checkbox"/> Nome	-		
<input type="checkbox"/> Produtividade relativa destaque			
<input type="checkbox"/> Realizar todas as atividades planejadas para a Sprint			
<input type="checkbox"/> Importar o código-fonte de seu projeto no GitLAB no final da iteração por sprint			
<input type="checkbox"/> Atualizar a completude dos entregáveis no NetProject a cada validação por sprint			
<input type="checkbox"/> Atualizar corretamente as atividades no GitLAB por sprint			
<input type="checkbox"/> Atualizar corretamente a Dokuwiki por sprint			
<input type="checkbox"/> Pontos Extras Equipe Vencedora do Sprint anterior			
<input type="checkbox"/> Colaborar com outro membro oferecendo assistência			
<input type="checkbox"/> Construir tutoriais publicáveis de novas tecnologias			

4.2.3 Lançamentos da Pontuação no Sistema

Para lançamento dos pontos individuais e por equipe, Figura 13, o usuário administrador deve criar o *sprint* definindo início e fim, Figura 14, criar a semana que irá lançar os pontos conforme Figura 15, selecionar a atividade que foi criada e os usuários que receberam os pontos.

Figura 13 – Tela lançamentos de pontos

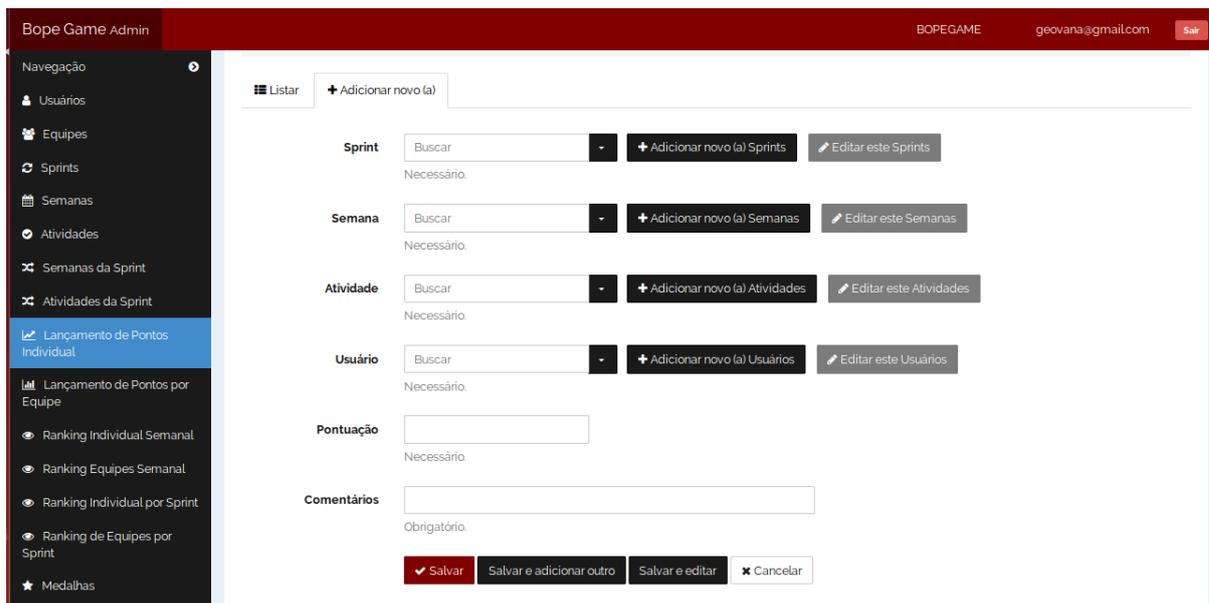


Figura 14 – Tela criação de *sprint*

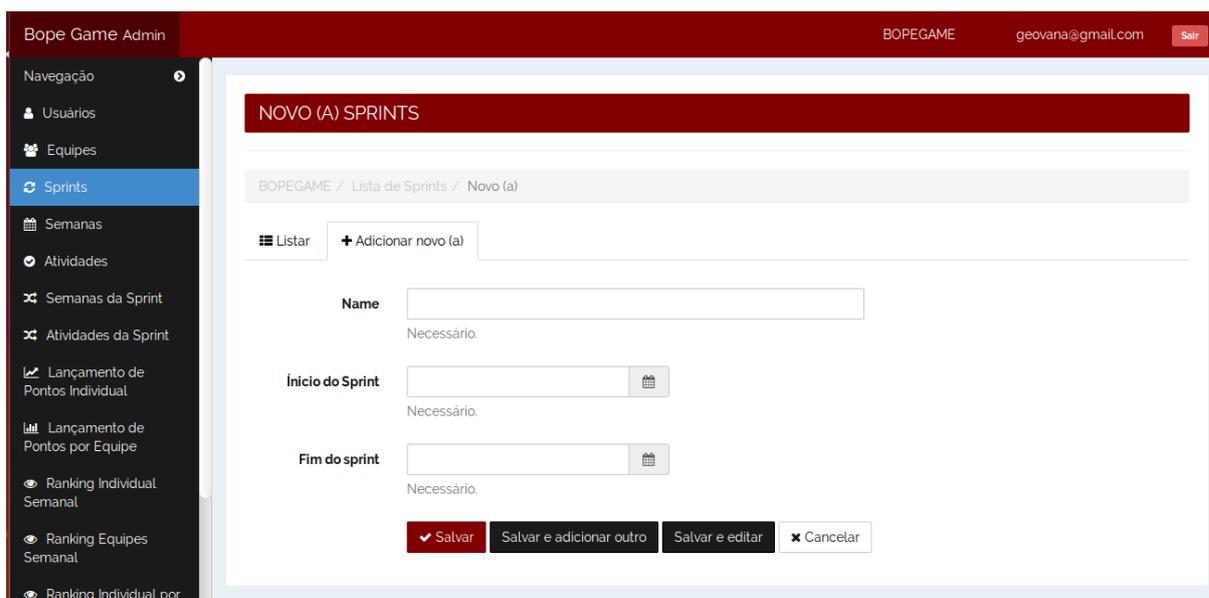
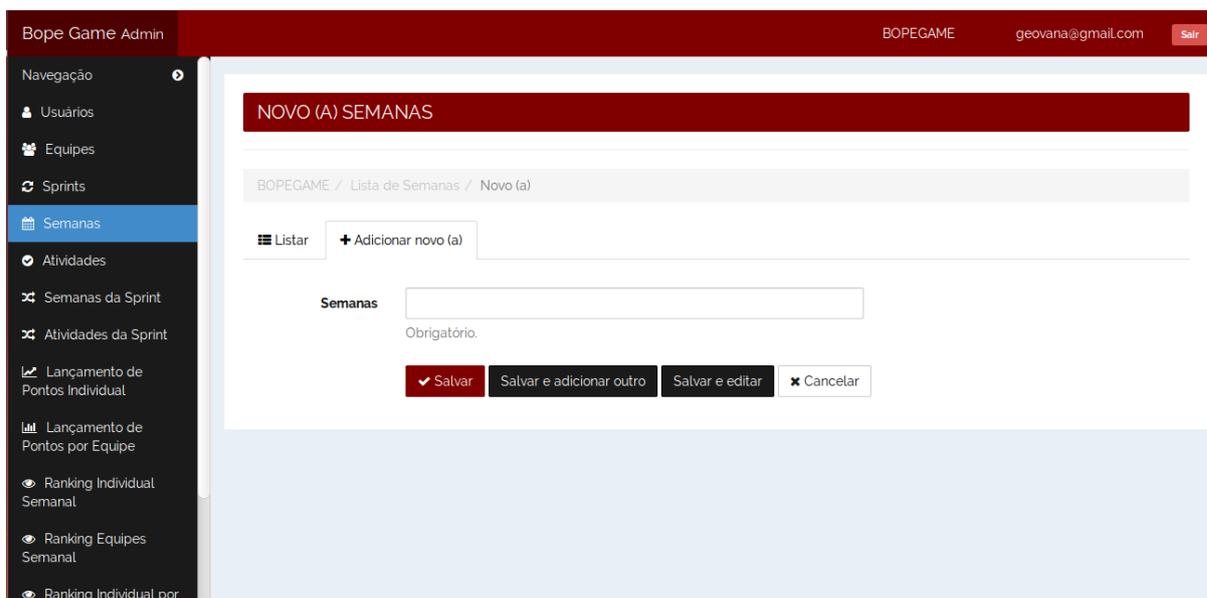


Figura 15 – Tela criação da semana



4.3 Resultados do Monitoramento da Gamificação

A seguir serão apresentados os resultados obtidos do monitoramento da gamificação durante o desenvolvimento do *software* no laboratório nos períodos letivo 16/01 e 16/02. As quantidades de semanas de cada *sprint* são apresentadas na Tabela 3 abaixo:

Tabela 3 – Quantidade de semanas por *sprint*

	Gamificação 16/01	Gamificação 16/02
<i>Sprint 1</i>	5 semanas	5 semanas
<i>Sprint 2</i>	4 semanas	4 semanas
<i>Sprint 3</i>	5 semanas	4 semanas
<i>Sprint 4</i>	4 semanas	5 semanas

4.3.1 Identificação das Não Conformidades

Nas Figuras 16 e 17 abaixo, foram analisadas as atividades semanais que cada colaborador deve realizar e identificadas as não conformidades individuais da gamificação nos períodos letivos 16/01 e 16/02.

Podemos observar que em geral, a taxa de não conformidades se manteve constante em ambos os períodos. A utilização do sistema *web* mantém a motivação dos colaboradores, porém não melhora o desempenho destes, ele apenas organiza melhor o processo de gerenciamento da gamificação e seus resultados.

Figura 16 – Não conformidades observadas por atividade com a gamificação no período letivo 16/01

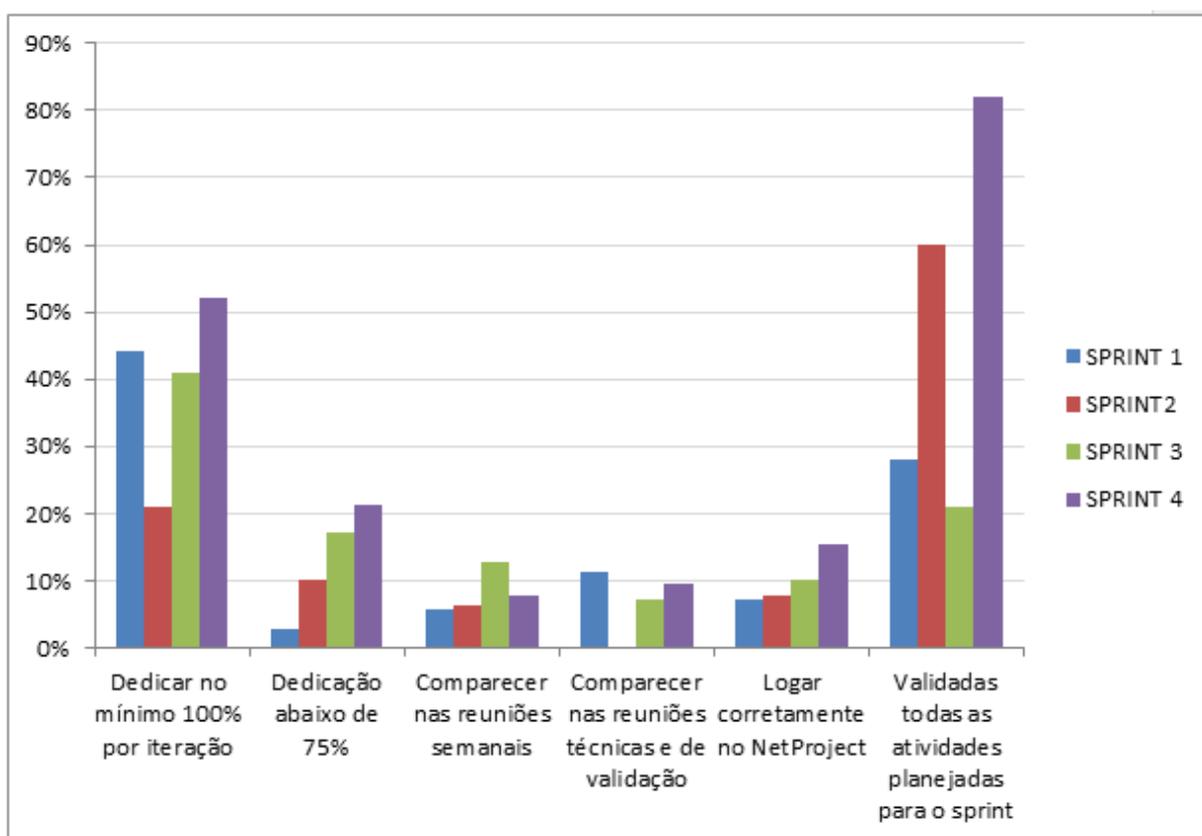
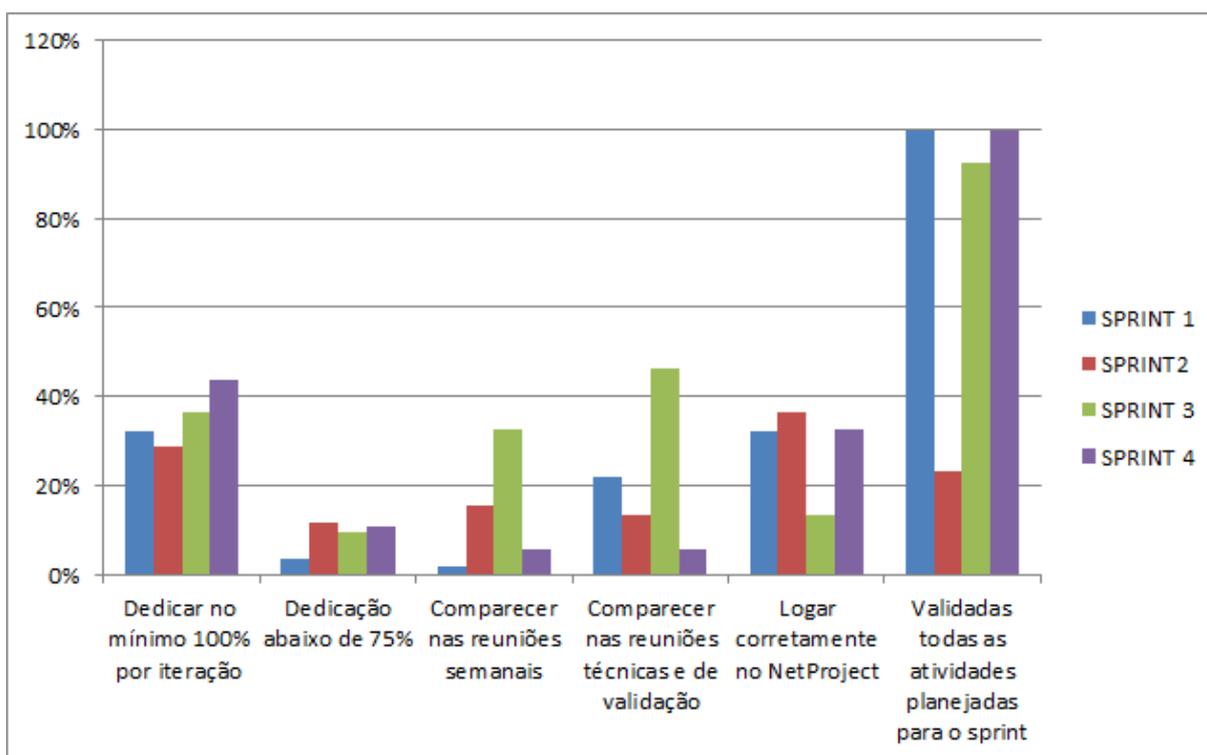


Figura 17 – Não conformidades observadas por atividade com a gamificação no período letivo 16/02



4.3.2 Resultados Individuais nos períodos letivos 16/01 e 16/02

Nas Tabelas 4 e 5 podemos observar o resultado final de cada *sprint* dos períodos letivos 16/01 e 16/02, identificando o vencedor da gamificação e a colocação de todos os colaboradores.

Podemos observar na Tabela 4, que durante o período letivo 16/01 houve um vencedor diferente para cada *sprint*. Em relação a pontuação final foi identificado um equilíbrio entre todos os colaboradores, indicando que a dedicação individual obteve uma melhora durante os *sprints*, ressaltando que apenas no *sprint* 1 - 16/01 a colocação do último colaborador apresentou maior dispersão em relação ao vencedor do *sprint*.

No período letivo 16/02, o resultado de cada *sprint* teve o mesmo vencedor para todas os *sprints*. Porém a pontuação dos demais colaboradores em todas os *sprints* apresentaram pouca diferença, o que podemos concluir que houve uma disputa para conseguir a primeira colocação.

Tabela 4 – Resultado Final Individual por *Sprint* no período letivo 16/01

	<i>Sprint 1</i>		<i>Sprint 2</i>		<i>Sprint 3</i>		<i>Sprint 4</i>	
	Nome	Total	Nome	Total	Nome	Total	Nome	Total
Posição 1	R1	45	L12	37	S15	39	C6	31
Posição 2	B2	39	R1	37	C6	38	M5	30
Posição 3	A3	37	M4	35	M4	37	M11	26
Posição 4	M4	37	B2	34	M5	37	L13	25
Posição 5	M5	36	A3	32	B2	34	M4	24
Posição 6	C6	33	C6	31	A3	33	S15	23
Posição 7	L7	32	M11	31	L13	33	A3	21
Posição 8	A8	31	G9	28	L7	32	M14	21
Posição 9	G9	31	M14	21	L10	32	L7	20
Posição 10	L10	31	L7	27	R1	32	L10	20
Posição 11	M11	29	S15	27	G9	31	L12	19
Posição 12	L12	26	L13	26	L12	31	G9	18
Posição 13	L13	23	L10	26	M11	29	A16	15
Posição 14	M14	19	M5	25	A16	24	-	-
Posição 15	-	-	A16	24	M14	9	-	-

Tabela 5 – Resultado Final Individual por *Sprint* no período letivo 16/02

	<i>Sprint 1</i>		<i>Sprint 2</i>		<i>Sprint 3</i>		<i>Sprint 4</i>	
	Nome	Total	Nome	Total	Nome	Total	Nome	Total
Posição 1	L10	32	L10	31	L10	31	L10	27
Posição 2	A3	31	M4	27	S15	27	W19	27
Posição 3	M4	31	A3	26	L7	26	A3	26
Posição 4	C6	30	S15	26	W19	26	G9	26
Posição 5	A16	29	C6	25	G9	25	L7	26
Posição 6	S15	29	M17	25	A3	24	M4	26
Posição 7	G9	28	A16	24	M4	24	S21	24
Posição 8	M17	28	G9	24	M17	23	S15	24
Posição 9	M11	27	M11	24	M18	22	M18	18
Posição 10	M18	27	W19	23	L20	17	C6	17
Posição 11	L7	25	L20	22	M11	16	L20	13
Posição 12	W19	18	M18	20	A16	6	-	-
Posição 13	L20	17	L7	18	C6	6	-	-

4.3.3 Resultados por Equipes

Ao final de cada *sprint* dos períodos letivos 16/01 e 16/02 alcançamos os seguintes resultados finais por equipes, podemos compara-los nas Tabelas 6 e 7.

Tabela 6 – Resultado Final Equipes por *sprint* no período letivo 16/01

	<i>Sprint 1</i>		<i>Sprint 2</i>		<i>Sprint 3</i>		<i>Sprint 4</i>	
	Nome	Total	Nome	Total	Nome	Total	Nome	Total
Posição 1	MM	28	MM	27	E1	32	E1	25
Posição 2	P2	25	E1	23	RS	27	RS	21
Posição 3	E1	23	BM	22	BM	23	MM	15
Posição 4	M3	14	P2	18	MM	21	BM	13
Posição 5	-	-	M3	15	P2	20	M3	13
Posição 6	-	-	-	-	M3	16	P2	13

Podemos observar que no período letivo 16/01 a equipe vencedora dos *sprints* 1 e 2, passou a assumir a quarta colocação no *sprint* 3 e segunda colocação no *sprint* 4, concluindo que houve um equilíbrio na pontuação entre as equipes. Nos *sprints* 3 e 4 o resultado final foi destinado a mesma equipe que finalizou seu projeto justificando o alto grau de comprometimento e o sucesso na conclusão do projeto. E a equipe da segunda colocação concluiu o projeto no final do *sprint* 4 com apenas quatro pontos de diferença da equipe vencedora.

Nos *sprint* 3 e 4, a equipe vencedora e a equipe que ficou em segunda colocação no *ranking* foram compostas apenas por um membro em cada equipe, e estes tiveram um bom desempenho individual repercutindo diretamente nos resultados de suas equipes. Ressaltando que o desempenho individual é um fator de grande importância que interfere no resultado final de cada equipe, uma vez que para cada equipe ganhar os pontos equivalentes às atividades no sistema gamificado, cada colaborador deve realizar as atividades individualmente.

Tabela 7 – Resultado Final Equipes por *sprint* no período letivo 16/02

	<i>Sprint 1</i>		<i>Sprint 2</i>		<i>Sprint 3</i>		<i>Sprint 4</i>	
	Nome	Total	Nome	Total	Nome	Total	Nome	Total
Posição 1	BM	23	M3	22	M3	16	MM	15
Posição 2	M3	22	BM	19	P2	13	M3	13
Posição 3	P2	19	P2	14	MM	10	P2	12
Posição 4	MM	18	MM	11	-	-	-	-

Ao analisar os resultados do período letivo 16/02, conseguimos observar um equilíbrio entre a pontuação obtida pelas equipes, ressaltando que a equipe em vencedora em cada *sprint* teve uma diferença de no máximo três pontos para segunda colocada.

Para cada equipe conseguir o ponto referente à atividade avaliada, todos os membros da equipe devem ter realizado a atividade individualmente, refletindo no desempenho da equipe a não realização da atividade. Quando todos os colaboradores de uma equipe não realizam todas as suas atividades planejadas para o *sprint*, sua equipe não ganha os pontos nessa atividade.

Parte IV

Conclusão

5 Conclusão

O trabalho realizado surgiu da necessidade de melhorar o gerenciamento da gamificação. Para atender as necessidades dos envolvidos com a gamificação, o desenvolvimento do BOPE Game seguiu várias práticas de engenharia de *software* aprendidas durante a graduação e principalmente no iMobilis. Destaca-se que a gamificação implantada teve como foco o gerenciamento dos projetos e que o BOPE Game pode ser utilizado para auxiliar na gamificação de qualquer processo de *software*.

A utilização de práticas ágeis de engenharia de *software* durante o desenvolvimento do projeto permitiu uma visão ampla de quais etapas devem ser definidas, realizadas e concluídas em conjunto com a implementação, testes e implantação do *software*. Durante o processo de desenvolvimento houveram mudanças na definição dos requisitos buscando por melhorias ou devido a impossibilidades de implementações. As entregas das funcionalidades do *software* foram feitas de forma contínua, algumas dessas apresentando maior complexidade no desenvolvimento.

O *software* foi projetado buscando apresentar uma boa *interface* e interação com os usuários. Foram considerados requisitos não funcionais como: (i) Usabilidade: Foi objetivo deixar o sistema fácil de usar garantindo a satisfação do usuário; (ii) Portabilidade: Um sistema *web* e responsivo foi construído para ser acessado em diferentes plataformas com acesso à internet; e (iii) Manutenibilidade: Foi escolhido um *framework* de desenvolvimento bem aceito no mercado, com uma comunidade ativa e vários artefatos de documentação também foram gerados.

Podemos ressaltar que o aumento da dedicação na realização das tarefas é um fator que vem melhorando e a busca por concluir todas as atividades propostas pode ser alcançada gradativamente. A motivação do colaborador é fundamental para o seguimento no processo de *software* e o reconhecimento pelo esforço individual são fatores importantes para manter o comprometimento e dedicação do colaborador no projeto.

6 Limitações e Trabalhos Futuros

Durante o desenvolvimento do *software* foram encontradas limitações na implementação devido a utilização do mecanismo *RailsAdmin* que dificultou a mudança do *layout* da página. Essa limitação impediu a utilização da *gem ratyrate* que oferece o uso de estrelas de classificação para avaliação que seriam utilizadas nas funcionalidades de apostas e votações.

Para trabalhos futuros, considera-se a implementação de novas melhorias e outras funcionalidades, como implementar novos mecanismos como de apostas e votação que permitirá um nova competição entre os colaboradores.

7 Referências Bibliográficas

CAVALCANTE, Niedja; AMANCIO, Francisco Dione de Sousa; JUCÁ, Emanuelle Nogueira; RODRIGUES, Morgana Viana. Uso de gamificação como auxílio para melhoria de processos: relato de experiência. In: SIMPÓSIO BRASILEIRO DE QUALIDADE DE SOFTWARE, 14., 2015, Manaus. **Anais**. Fortaleza: Universidade de Fortaleza, 2014, p. 268-275.

COTA, M.A. **Gamificando um processo de software em um laboratório de pesquisa e desenvolvimento**. Instituto de Ciências Exatas e Aplicadas (ICEA), Universidade Federal de Ouro Preto João Monlevade, MG – Brasil, 2016.

DUBOIS, Daniel J.; TAMBURRELLI, Giordano. **Understanding Gamification Mechanisms for Software Development**. ESEC/FSE 2013 Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering – New York: ACM, 2013.

FOGG, B. J. **A Behavior Model for Persuasive Design**. 4a International Conference on Persuasive Technology – Claremont, ACM, 2009.

PEREIRA, Igor Muzetti ; Amorim ,Vicente J. P; Cota, Marcos A; Gonçalves Geovana C. **Uso de Gamificação no Gerenciamento Ágil de Projetos: Um relato de Experiência**. WBMA – 7 Workshop Brasileiro de Métodos Ágeis – 2016.

NG, Pan-Wei. **Software Process Improvement and Gaming using Essence: An Industrial Experience**. Journal of Industrial and Intelligent Information Vol. 2, No. 1, Mar. 2014.

PEREIRA, Igor Muzetti. **Desenvolvendo software inovador em universidades públicas: Adaptando processos ágeis para a realidade de laboratórios de pesquisa e desenvolvimento**. 2014. 63 f. Mestrado (Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Ouro Preto, Ouro Preto. 2014.

PEREIRA, I. M. **Domain Driven Design: Trabalhando com histórias de usuário**. Revista Engenharia de Software Magazine. Ed. 72. Kumar, J; Herger, M. Gamification at Work: Designing Engaging Business Software. Nova York: IDF, 2013, 168 p. Werbach, K. **Curso de Gamificação**. Universidade da Pensilvânia, 2015. Disponível em: < <https://www.coursera.org/learn/gamification/>>.

PMI. **Um guia do conhecimento em gerenciamento de projetos**. Guia PMBOK. 5. ed. [S.l.:s.n.], 2013. ISBN 9781628250091. Acesso em: 27 abril 2015. Citado na página 36.

PRESSMAN, Roger S. **Engenharia de Software - Uma Abordagem Profissional**. 18a Ed. Amgh Editora, 2016.

SOMMERVILLE, Ian. **Engenharia de Software**. 9o Edição. PEARSON EDUCATION - BR, 2011.

VIANNA, Ysmar; VIANNA, Mauricio; MEDINA, Bruno; TANAKA, Samara.

Gamification, Inc : como reinventar empresas a partir de jogos. 1. ed. Rio de Janeiro: MJV Press, 2013. 116p.

Anexos

ANEXO A – HISTÓRIAS DE USUÁRIOS

Figura 18 – História de Usuários – Serviço de Autenticação

Histórias de Usuários
Serviço de Autenticação
Autenticar usuário
COMO um usuário cadastrado EU GOSTARIA DE logar no sistema mediante fornecimento do "nome do usuário" e "senha" PARA CONSEGUIR acesso as funcionalidades do sistema
Autenticar usuário administrador
COMO um usuário administrador cadastrado EU GOSTARIA DE logar no sistema mediante fornecimento do "nome do usuário" e "senha" PARA CONSEGUIR acesso as funcionalidades do sistema

Figura 19 – História de Usuários – Serviço de Gerenciamento de Usuários

Serviço de Gerenciamento de Usuários
Cadastrar novo usuário
COMO um usuário administrador EU GOSTARIA DE cadastrar novos usuários mediante fornecimento de "nome", "senha" e "e-mail" PARA CONSEGUIR incrementar a quantidade de usuários que terão acesso as funcionalidades do sistema
Editar perfil de usuário
COMO um usuário administrador EU GOSTARIA DE editar usuários mediante fornecimento de "nome" e "novo nome" PARA CONSEGUIR atualizar o nome do usuário
Excluir usuário
COMO um usuário administrador EU GOSTARIA DE excluir usuários mediante fornecimento de "nome" PARA CONSEGUIR deixar de ter acesso as funcionalidades do sistema
Pesquisar usuário
COMO um usuário administrador EU GOSTARIA DE pesquisar usuários mediante fornecimento de "nome" PARA CONSEGUIR encontrar um usuário específico no sistema

Figura 20 – História de Usuários – Serviço de Gerenciamento de Usuários

Criar equipe COMO um usuário administrador EU GOSTARIA DE criar equipes mediante fornecimento de "nome da equipe" PARA CONSEGUIR criar as equipes dos projetos no sistema
Excluir equipe COMO um usuário administrador EU GOSTARIA DE excluir equipes mediante fornecimento de "nome da equipe" e "id" da equipe PARA CONSEGUIR excluir equipes no sistema
Adicionar usuário a equipe COMO um usuário administrador EU GOSTARIA DE adicionar usuários às equipes mediante fornecimento de "nome do usuário" e "nome da equipe" PARA CONSEGUIR adicionar usuários às equipes dos projetos
Excluir usuário da equipe COMO um usuário administrador EU GOSTARIA DE excluir usuários de equipes mediante fornecimento de "nome do usuário" e "nome da equipe" PARA CONSEGUIR excluir usuários de equipes dos projetos

Figura 21 – História de Usuários – Serviço de Atividades

Serviço de Atividades
Cadastrar as atividades que serão avaliadas COMO um usuário cadastrado administrador EU GOSTARIA DE cadastrar atividades mediante fornecimento de "nome da atividade" PARA CONSEGUIR cadastrar atividades no sistema
Editar as atividades que serão avaliadas COMO um usuário cadastrado administrador EU GOSTARIA DE editar atividades mediante fornecimento de "nome da atividade", "novo nome da atividade" PARA CONSEGUIR editar atividades no sistema
Excluir as atividades que serão avaliadas COMO um usuário cadastrado administrador EU GOSTARIA DE excluir atividades mediante fornecimento de "nome da atividade" PARA CONSEGUIR excluir atividades no sistema
Inserir pontuação nas atividades COMO um usuário cadastrado administrador EU GOSTARIA DE inserir pontuação nas atividades mediante fornecimento de "nome da atividade", "pontuação" PARA CONSEGUIR inserir pontuação para cada atividades no sistema
Editar as pontuações nas atividades COMO um usuário cadastrado administrador EU GOSTARIA DE editar pontuação nas atividades mediante fornecimento de "nome da atividade", "pontuação" PARA CONSEGUIR editar pontuação para cada atividades no sistema
Associar ponto de usuário à equipe COMO um usuário cadastrado administrador EU GOSTARIA DE associar usuário à equipe mediante fornecimento de "nome do usuário", "nome da equipe" PARA CONSEGUIR associar usuário à equipe no sistema

Figura 22 – História de Usuários – Serviço de Visualização de Informações

Serviço de Visualização de Informações
Visualizar pontuação dos usuários de cada semana
COMO um usuário cadastrado EU GOSTARIA DE visualizar a pontuação semanal PARA CONSEGUIR saber qual colaborador obteve a maior pontuação na semana no sistema
Visualizar pontuação final dos usuários de cada sprint
COMO um usuário cadastrado EU GOSTARIA DE visualizar a pontuação final dos usuários de cada sprint PARA CONSEGUIR saber qual colaborador obteve a maior pontuação no sprint no sistema
Visualizar pontuação das equipes de cada semana
COMO um usuário cadastrado EU GOSTARIA DE visualizar a pontuação das equipes de cada semana PARA CONSEGUIR saber qual equipe obteve a maior pontuação na semana no sistema
Visualizar pontuação final das equipes de cada sprint
COMO um usuário cadastrado EU GOSTARIA DE visualizar a pontuação final das equipes de cada sprint PARA CONSEGUIR saber qual equipe obteve a maior pontuação no sprint no sistema

Figura 23 – História de Usuários – Serviço de Visualização de Informações

Visualizar tela de medalhas
COMO um usuário cadastrado EU GOSTARIA DE visualizar a tela de medalhas no sistema PARA CONSEGUIR saber qual colaborador e equipe são vencedores
Visualizar usuários
COMO um usuário administrador EU GOSTARIA DE visualizar os usuários PARA CONSEGUIR saber qual usuário está cadastrado no sistema
Visualizar equipes
COMO um usuário administrador EU GOSTARIA DE visualizar as equipes PARA CONSEGUIR saber quais equipes estão cadastradas no sistema
Visualizar integrantes da equipe
COMO um usuário administrador EU GOSTARIA DE visualizar integrantes de cada equipe PARA CONSEGUIR saber quais integrantes fazem parte de cada equipe no sistema

Figura 24 – História de Usuários – Serviço de Gerenciamento de Sprint

Serviço de Gerenciamento de Sprint
Adicionar Sprint
COMO um usuário administrador EU GOSTARIA DE adicionar Sprint PARA CONSEGUIR controlar o número de Sprints necessárias em um período
Excluir Sprint
COMO um usuário administrador EU GOSTARIA DE excluir Sprint PARA CONSEGUIR controlar o número de Sprint necessárias no período
Adicionar semana
COMO um usuário administrador EU GOSTARIA DE adicionar semana PARA CONSEGUIR controlar o número de semana necessárias no Sprint
Excluir semana
COMO um usuário cadastrado administrador EU GOSTARIA DE excluir semana PARA CONSEGUIR controlar o número de semana necessárias no Sprint

Figura 25 – História de Usuários – Serviço de Não-conformidades

Serviço de Não-conformidades
Verificar as atividades que não tiveram pontos atribuídos
COMO um usuário cadastrado administrador EU GOSTARIA DE verificar as atividades que não tiveram pontos atribuídos PARA CONSEGUIR saber quais atividades não tiveram pontos no sistema

ANEXO B – CENÁRIOS DE TESTE

Figura 26 – Cenários de Teste – Serviço de Autenticação

Serviço de Autenticação
<p>História de usuário: Autenticar Admin</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário admin cadastrado fornece credenciais válidas <p>DADO que um usuário admin cadastrado forneça “senha” válida QUANDO o botão “Entrar” for pressionado ENTÃO a aplicação deve transitar para a tela principal a partir da qual o usuário poderá selecionar</p>
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário admin cadastrado fornece “senha” inválida <p>DADO que um usuário cadastrado forneça “senha” inválida QUANDO o botão “Entrar” for pressionado ENTÃO (a) a mensagem “Senha inválida” deve ser apresentada e (b) a aplicação deve permanecer na tela de login</p>
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário não cadastrado tentar usar a aplicação <p>DADO que um usuário não cadastrado forneça uma “senha” qualquer QUANDO o botão “entrar” for pressionado ENTÃO (a) a mensagem “Usuário desconhecido” deve ser apresentada e (b) a aplicação deve permanecer na tela de login</p>
<p>História de usuário: Autenticar usuário</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário cadastrado fornece credenciais válidas <p>DADO que um usuário cadastrado forneça “senha” válida QUANDO o botão “Entrar” for pressionado ENTÃO a aplicação deve transitar para a tela principal a partir da qual o usuário poderá selecionar</p>
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário cadastrado fornece “senha” inválida <p>DADO que um usuário cadastrado forneça “senha” inválida QUANDO o botão “Entrar” for pressionado ENTÃO (a) a mensagem “Senha inválida” deve ser apresentada e (b) a aplicação deve permanecer na tela de login</p>
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário não cadastrado tentar usar a aplicação <p>DADO que um usuário não cadastrado forneça uma “senha” qualquer QUANDO o botão “entrar” for pressionado ENTÃO (a) a mensagem “Usuário desconhecido” deve ser apresentada e (b) a aplicação deve permanecer na tela de login</p>

Figura 27 – Cenários de Teste – Serviço de Visualização de Informações

Serviço de Visualização de Informações
<p>História de usuário: Visualizar pontuação dos usuários de cada semana</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário acessa página de pontuação <p>DADO que um usuário acesse a página de pontuação QUANDO o botão "Visualizar pontuação semanal" for pressionado ENTÃO o sistema deve exibir na tela os dados sobre a pontuação semanal MAS a pontuação pode não estar disponível e uma mensagem "Pontuação dos usuários não disponível" deve ser exibida</p>
<p>História de usuário: Visualizar pontuação final dos usuários de cada sprint</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário acessa página de pontuação <p>DADO que um usuário acesse a página de pontuação QUANDO o botão "Visualizar pontuação final dos usuários" for pressionado ENTÃO o sistema deve exibir na tela os dados sobre a pontuação final do sprint MAS a pontuação pode não estar disponível e uma mensagem "Pontuação final não disponível" deve ser exibida</p>
<p>História de usuário: Visualizar pontuação das equipes de cada semana</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário acessa página de pontuação <p>DADO que um usuário acesse a página de pontuação QUANDO o botão "Visualizar pontuação das equipes" for pressionado ENTÃO o sistema deve exibir na tela os dados sobre a pontuação das equipes de cada semana MAS a pontuação pode não estar disponível e uma mensagem "Pontuação das equipes não disponível" deve ser exibida</p>
<p>História de usuário: Visualizar pontuação final das equipes de cada sprint</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário acessa página de pontuação <p>DADO que um usuário acesse a página de pontuação QUANDO o botão "Visualizar pontuação das equipes" for pressionado ENTÃO o sistema deve exibir na tela os dados sobre a pontuação das equipes de cada sprint MAS a pontuação pode não estar disponível e uma mensagem "Pontuação das equipes não disponível" deve ser exibida</p>

Figura 28 – Cenários de Teste – Serviço de Visualização de Informações

<p>História de usuário: Visualizar tela de medalhas</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário acessa página de pontuação <p>DADO que um usuário acesse a página de medalhas QUANDO o botão "Visualizar medalhas" for pressionado ENTÃO o sistema deve exibir na tela o vencedor individual e a equipe MAS as medalhas podem não estar disponíveis e uma mensagem "Medalhas não disponíveis" deve ser exibida</p>
<p>História de usuário: Visualizar usuários</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário acessa página de pontuação <p>DADO que um usuário acesse a página de usuários QUANDO o botão "Visualizar usuários" for pressionado ENTÃO o sistema deve exibir na tela os nomes dos usuários MAS os usuários podem não estar disponíveis e uma mensagem "Usuários não disponíveis" deve ser exibida</p>
<p>História de usuário: Visualizar equipes</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário acessa página de pontuação <p>DADO que um usuário acesse a página de equipes QUANDO o botão "Visualizar equipes" for pressionado ENTÃO o sistema deve exibir na tela as equipes cadastradas MAS as equipes podem não estar disponíveis e uma mensagem "Equipes não disponíveis" deve ser exibida</p>
<p>História de usuário: Visualizar integrantes da equipe</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário acessa página de pontuação <p>DADO que um usuário acesse a página de equipes QUANDO o botão "Visualizar integrantes da equipe" for pressionado ENTÃO o sistema deve exibir na tela os integrantes das equipes cadastradas MAS os integrantes da equipe podem não estar disponíveis e uma mensagem "Integrantes não disponíveis" deve ser exibida</p>

Figura 29 – Cenários de Teste – Serviço de Gerenciamento de *sprint*

Serviço de Gerenciamento de Sprint
<p>História de usuário: Adicionar sprint</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário acessa página de sprint <p>DADO que um usuário administrador acesse a página de sprint QUANDO o botão “Adicionar sprint” for pressionado ENTÃO o sistema deve redirecionar para a página de adicionar sprint MAS a página pode não estar disponíveis e uma mensagem “Página não disponível” deve ser exibida</p>
<p>História de usuário: Excluir sprint</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário acessa página de Sprint <p>DADO que um usuário administrador acesse a página de sprint QUANDO o botão “Excluir Sprint” for pressionado ENTÃO (a) a mensagem “Tem certeza que deseja excluir?” deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>
<p>História de usuário: Adicionar semana</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário acessa página de sprint <p>DADO que um usuário administrador acesse a página de sprint QUANDO o botão “Adicionar semana” for pressionado ENTÃO o sistema deve redirecionar para a página de adicionar semana MAS a página pode não estar disponíveis e uma mensagem “Página não disponível” deve ser exibida</p>
<p>História de usuário: Excluir semana</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário acessa página de sprint <p>DADO que um usuário administrador acesse a página de sprint QUANDO o botão “Excluir semana” for pressionado ENTÃO (a) a mensagem “Tem certeza que deseja excluir?” deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>

Figura 30 – Cenários de Teste – Serviço de Gerenciamento de Usuários

Serviço de Gerenciamento de Usuários
<p>História de usuário: Cadastrar novo usuário</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário fornece “nome”, “e-mail”, “senha” válidos <p>DADO que um usuário forneça “nome”, “e-mail”, “senha” corretamente QUANDO o botão “Cadastrar” for pressionado ENTÃO (a) a mensagem “Usuário cadastrado com sucesso” deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p> <ul style="list-style-type: none"> • Cenário: Alternativo – Usuário fornece e-mail inválido <p>DADO que um usuário forneça “e-mail” inválido QUANDO o botão “Cadastrar” for pressionado ENTÃO (a) a mensagem “E-mail inválido, entre com um e-mail no formato válido” deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p> <ul style="list-style-type: none"> • Cenário: Alternativo – Usuário deixa campos em branco <p>DADO que um usuário deixe algum campo em branco QUANDO o botão “Cadastrar” for pressionado ENTÃO (a) a mensagem “Preencha este campo” deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>

Figura 31 – Cenários de Teste – Serviço de Gerenciamento de Usuários

<p>História de usuário: Editar perfil de usuário</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário administrativo fornece “nome”, “e-mail” válidos do usuário comum que deseja editar <p>DADO que um usuário forneça “nome”, “e-mail” corretamente QUANDO o botão “Editar” for pressionado ENTÃO (a) a mensagem “Usuário atualizado com sucesso” deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p> <ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador fornece e-mail inválido <p>DADO que um usuário forneça “e-mail” inválido QUANDO o botão “Editar” for pressionado ENTÃO (a) a mensagem “E-mail inválido, entre com um e-mail no formato válido” deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p> <ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador deixa campos em branco <p>DADO que um usuário deixe algum campo em branco QUANDO o botão “Editar” for pressionado ENTÃO (a) a mensagem “Preencha este campo” deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>

Figura 32 – Cenários de Teste – Serviço de Gerenciamento de Usuários

<p>História de usuário: Excluir usuário</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário administrador busca usuário comum que deseja excluir por “nome ou e-mail” <p>DADO que exista um usuário comum com “nome” e “e-mail” informados QUANDO o botão “Excluir” for pressionado ENTÃO (a) a mensagem “Usuário comum excluído com sucesso” deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p> <ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador busca usuário comum que deseja excluir por “nome ou e-mail” inexistentes <p>DADO que não exista um usuário comum com “nome” e “e-mail” informados QUANDO o botão “Excluir” for pressionado ENTÃO (a) a mensagem “Usuário comum não existente” deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>
--

Figura 33 – Cenários de Teste – Serviço de Gerenciamento de Usuários

História de usuário: Pesquisar usuário
<ul style="list-style-type: none"> • Cenário: Principal – Usuário administrador fornece "nome ou e-mail" para usuário comum válidos
DADO que usuário administrador forneça "nome" e "e-mail" corretamente QUANDO o botão "Pesquisar" for pressionado ENTÃO (a) o sistema deve exibir dados sobre o usuário se existir ou (b) exibir a mensagem "Usuário comum não encontrado"
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador deixa campos em branco
DADO que usuário administrador deixe algum campo em branco QUANDO o botão "Pesquisar" for pressionado ENTÃO (a) a mensagem "Preencha esse campo" deve ser apresentada e (b) a aplicação deve permanecer na tela atual

Figura 34 – Cenários de Teste – Serviço de Gerenciamento de Usuários

História de usuário: Criar equipe
<ul style="list-style-type: none"> • Cenário: Principal – Usuário fornece "nome de equipe" válidos
DADO que um usuário forneça "nome de equipe" corretamente QUANDO o botão "Criar" for pressionado ENTÃO (a) a mensagem "Equipe cadastrada com sucesso" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário fornece "nome de equipe" inválido
DADO que um usuário forneça "nome de equipe" inválido QUANDO o botão "Criar" for pressionado ENTÃO (a) a mensagem "Nome de equipe inválido" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário deixa campos em branco
DADO que um usuário deixe algum campo em branco QUANDO o botão "Criar" for pressionado ENTÃO (a) a mensagem "Preencha este campo" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
História de usuário: Excluir equipe
<ul style="list-style-type: none"> • Cenário: Principal – Usuário administrador busca equipe que deseja excluir por "nome de equipe"
DADO que exista uma equipe com "nome" informado QUANDO o botão "Excluir" for pressionado ENTÃO (a) a mensagem "Equipe excluído com sucesso" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador busca equipe que deseja excluir por "nome de equipe" inexistente
DADO que não exista uma equipe com "nome" informado QUANDO o botão "Excluir" for pressionado ENTÃO (a) a mensagem "Equipe não existente" deve ser apresentada e (b) a aplicação deve permanecer na tela atual

Figura 35 – Cenários de Teste – Serviço de Gerenciamento de Usuários

História de usuário: Adicionar usuário a equipe
<ul style="list-style-type: none"> • Cenário: Principal – Usuário administrador fornece "nome de usuário e de equipe" válidos
DADO que um usuário administrador forneça "nome de usuário e equipe" corretamente QUANDO o botão "Adicionar usuário a equipe" for pressionado ENTÃO (a) a mensagem "Usuário foi adicionado a equipe com sucesso" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador fornece "nome de usuário e de equipe" inválido
DADO que um usuário administrador forneça "nome de usuário e equipe" inválido QUANDO o botão "Adicionar usuário a equipe" for pressionado ENTÃO (a) a mensagem "Nome de equipe inválido" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário deixa campos em branco
DADO que um usuário administrador deixe algum campo em branco QUANDO o botão "Adicionar usuário a equipe" for pressionado ENTÃO (a) a mensagem "Preencha este campo" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
História de usuário: Excluir usuário da equipe
<ul style="list-style-type: none"> • Cenário: Principal – Usuário administrador busca usuário da equipe que deseja excluir por "nome de usuário e equipe"
DADO que exista um usuário cadastrado a equipe com "nome" informado QUANDO o botão "Excluir" for pressionado ENTÃO (a) a mensagem "Usuário excluído da equipe com sucesso" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador busca usuário da equipe que deseja excluir por "nome de usuário e equipe" inexistente
DADO que não exista um usuário cadastrado a equipe com "nome" informado QUANDO o botão "Excluir" for pressionado ENTÃO (a) a mensagem "Usuário não existente à equipe" deve ser apresentada e (b) a aplicação deve permanecer na tela atual

Figura 36 – Cenários de Teste – Serviço de Atividades

Serviço de Atividades
<p>História de usuário: Cadastrar as atividades que serão avaliadas</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário fornece "nome da atividade" válida <p>DADO que um usuário forneça "nome da atividade" corretamente QUANDO o botão "Cadastrar" for pressionado ENTÃO (a) a mensagem "Atividade cadastrada com sucesso" deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário fornece "nome da atividade" inválido <p>DADO que um usuário forneça "nome da atividade" inválido QUANDO o botão "Cadastrar" for pressionado ENTÃO (a) a mensagem "Nome da atividade inválido" deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário deixa campos em branco <p>DADO que um usuário deixe algum campo em branco QUANDO o botão "Cadastrar" for pressionado ENTÃO (a) a mensagem "Preencha este campo" deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>

Figura 37 – Cenários de Teste – Serviço de Atividades

<p>História de usuário: Editar as atividades que serão avaliadas</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário administrativo fornece "nome da atividade" válidos do usuário comum que deseja editar <p>DADO que um usuário forneça "nome da atividade" corretamente QUANDO o botão "Editar" for pressionado ENTÃO (a) a mensagem "Atividade atualizada com sucesso" deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador fornece "nome da atividade" inválido <p>DADO que um usuário forneça "nome da atividade" inválido QUANDO o botão "Editar" for pressionado ENTÃO (a) a mensagem "Nome da atividade inválido" deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador deixa campos em branco <p>DADO que um usuário deixe algum campo em branco QUANDO o botão "Editar" for pressionado ENTÃO (a) a mensagem "Preencha este campo" deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>

Figura 38 – Cenários de Teste – Serviço de Atividades

<p>História de usuário: Excluir as atividades que serão avaliadas</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário administrador busca atividades que deseja excluir por "nome da atividade" <p>DADO que exista uma atividade cadastrada informada QUANDO o botão "Excluir" for pressionado ENTÃO (a) a mensagem "Atividade excluída com sucesso" deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador busca atividades que deseja excluir por "nome de usuário e equipe" inexistente <p>DADO que não exista uma atividade cadastrada informada QUANDO o botão "Excluir" for pressionado ENTÃO (a) a mensagem "Atividade não existente" deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>

Figura 39 – Cenários de Teste – Serviço de Atividades

<p>História de usuário: Inserir pontuação nas atividades</p> <ul style="list-style-type: none"> • Cenário: Principal – Usuário administrativo fornece "pontuação da atividade" válida <p>DADO que um usuário administrativo forneça "pontuação da atividade" corretamente QUANDO o botão "Inserir" for pressionado ENTÃO (a) a mensagem "Pontuação inserida com sucesso" deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrativo fornece "pontuação da atividade" inválido <p>DADO que um usuário administrativo forneça "pontuação da atividade" inválida QUANDO o botão "Inserir" for pressionado ENTÃO (a) a mensagem "Pontuação da atividade inválida" deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrativo deixa campos em branco <p>DADO que um usuário administrativo deixe algum campo em branco QUANDO o botão "Inserir" for pressionado ENTÃO (a) a mensagem "Preencha este campo" deve ser apresentada e (b) a aplicação deve permanecer na tela atual</p>

Figura 40 – Cenários de Teste – Serviço de Atividades

História de usuário: Editar as pontuações nas atividades
<ul style="list-style-type: none"> • Cenário: Principal – Usuário administrativo fornece "pontuação da atividade" válida
DADO que um usuário administrativo forneça "pontuação da atividade" corretamente QUANDO o botão "Editar" for pressionado ENTÃO (a) a mensagem "Pontuação atualizada com sucesso" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador fornece "nome da atividade" inválido
DADO que um usuário administrativo forneça "pontuação da atividade" inválido QUANDO o botão "Editar" for pressionado ENTÃO (a) a mensagem "Pontuação da atividade inválida" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador deixa campos em branco
DADO que um usuário administrativo deixe algum campo em branco QUANDO o botão "Editar" for pressionado ENTÃO (a) a mensagem "Preencha este campo" deve ser apresentada e (b) a aplicação deve permanecer na tela atual

Figura 41 – Cenários de Teste – Serviço de Atividades

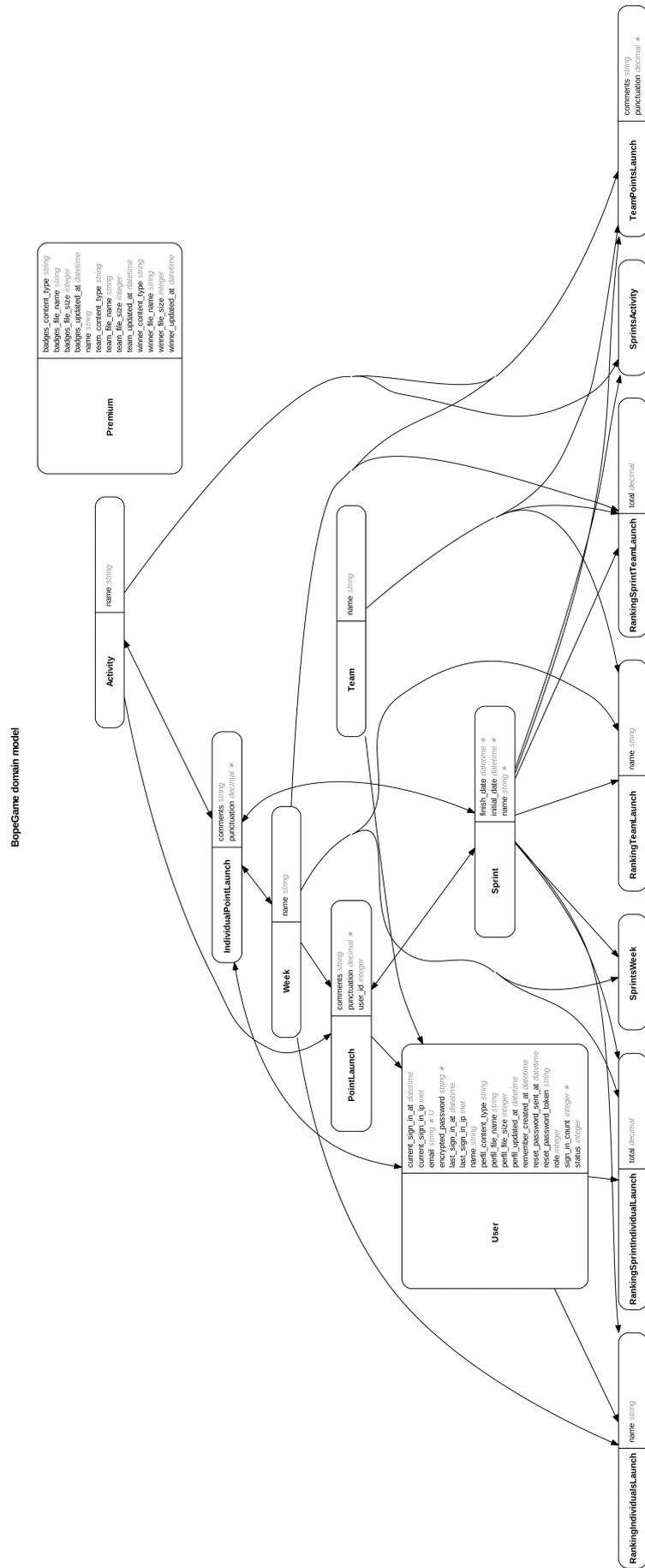
História de usuário: Associar ponto de usuário à equipe
<ul style="list-style-type: none"> • Cenário: Principal – Usuário administrativo fornece "pontuação do usuário", e "nome da equipe" válidos
DADO que um usuário administrativo forneça "pontuação do usuário", e "nome da equipe" corretamente QUANDO o botão "Associar" for pressionado ENTÃO (a) a mensagem "Associação realizada com sucesso" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador fornece "pontuação do usuário", e "nome da equipe" inválidos
DADO que um usuário administrativo forneça "pontuação do usuário", e "nome da equipe" inválidos QUANDO o botão "Associar" for pressionado ENTÃO (a) a mensagem "Associação não realizada" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário administrador deixa campos em branco
DADO que um usuário administrativo deixe algum campo em branco QUANDO o botão "Associar" for pressionado ENTÃO (a) a mensagem "Preencha este campo" deve ser apresentada e (b) a aplicação deve permanecer na tela atual

Figura 42 – Cenários de Teste – Serviço de Não conformidades

Serviço Não-conformidades
História de Usuário: Verificar as atividades que não tiveram pontos atribuídos
<ul style="list-style-type: none"> • Cenário: Principal – Usuário fornece "nome da atividade" e "pontuação da atividade" válidas
DADO que um usuário forneça "nome da atividade" e "pontuação da atividade" corretamente QUANDO o botão "Verificar" for pressionado ENTÃO (a) a mensagem "Verificação realizada com sucesso" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário fornece "nome da atividade" e "pontuação da atividade" inválidas
DADO que um usuário forneça "nome da atividade" e "pontuação da atividade" inválidas QUANDO o botão "Verificar" for pressionado ENTÃO (a) a mensagem "Verificação não realizada" deve ser apresentada e (b) a aplicação deve permanecer na tela atual
<ul style="list-style-type: none"> • Cenário: Alternativo – Usuário deixa campos em branco
DADO que um usuário deixe algum campo em branco QUANDO o botão "Verificar" for pressionado ENTÃO (a) a mensagem "Preencha este campo" deve ser apresentada e (b) a aplicação deve permanecer na tela atual

ANEXO C – PROJETO DE BANCO DE DADOS

Figura 43 – Modelo de Entidade-Relacionamento



ANEXO D – TUTORIAL DE
IMPLANTAÇÃO DO SISTEMA NO
LABORATÓRIO IMOBILIS

Figura 44 – Instalando *Ruby*



Universidade Federal de Ouro Preto - UFOP
Departamento de Computação e Sistemas de Informação - DECSI

LABORATÓRIO
Mobilis
Computação Móvel

Tutorial de Implantação do BOPE Game

Desenvolvido por: Geovana Gonçalves (geovanaqonc@gmail.com)
Data: 13/03/2017
Professor Orientador: Igor Muzetti Pereira (igormuzetti@gmail.com)
Link de acesso Dokuwiki: (<http://200.239.152.228/dokuwiki/doku.php?id=imobilis:projetos:qp:tutoriais>)

Configuração do Ruby on Rails no Ubuntu 14.04

1º Passo: Instalando o Ruby

Instalando a versão do Ruby 2.4.0

Instalando algumas dependências para o Ruby:

```
$ sudo apt-get update  
  
$ sudo apt-get install git-core curl zlib1g-dev build-essential libssl-dev libreadline-dev libyaml-dev  
libsqlite3-dev sqlite3 libxml2-dev libxslt1-dev libcurl4-openssl-dev python-software-properties libffi-dev  
nodejs
```

Em seguida, instalando o Ruby usando o método rbenv.

Para instalar com método rbenv, é necessário duas etapas:

- 1º Instalar rbenv
- 2º Instalar ruby-build

```
cd  
git clone https://github.com/rbenv/rbenv.git ~/.rbenv  
echo 'export PATH="$HOME/.rbenv/bin:$PATH"' >> ~/.bashrc  
echo 'eval "$(rbenv init -)"' >> ~/.bashrc  
exec $SHELL  
  
git clone https://github.com/rbenv/ruby-build.git ~/.rbenv/plugins/ruby-build  
echo 'export PATH="$HOME/.rbenv/plugins/ruby-build/bin:$PATH"' >> ~/.bashrc  
exec $SHELL  
  
rbenv install 2.4.0  
rbenv global 2.4.0  
ruby -v
```

O último passo é instalar o Bundler

```
gem install bundler
```

Figura 45 – Instalando *Rails* e *PostgreSQL*

2º Passo: Instalando Rails
Instalando a versão do Rails 5.0.2
Instalando um runtime JavaScript como NodeJS.
Para instalar o NodeJS:
<pre>curl -sL https://deb.nodesource.com/setup_4.x sudo -E bash - sudo apt-get install -y nodejs</pre>
Instalando o Rails:
<pre>gem install rails -v 5.0.2</pre>
Executar o seguinte comando para tornar os executáveis disponíveis:
<pre>rbenv rehash</pre>
Conferindo a versão instalada:
<pre>rails -v # Rails 5.0.2</pre>
3º Passo: Instalando e Configurando o Banco de Dados PostgreSQL
Para instalar uma versão recente do PostgreSQL, basta adicionar um novo repositório:
<pre>\$ sudo sh -c "echo 'deb http://apt.postgresql.org/pub/repos/apt/ precise-pgdg main' > /etc/apt/sources.list.d/pgdg.list" wget --quiet -O - http://apt.postgresql.org/pub/repos/apt/ACCC4CF8.asc sudo apt-key add - \$ sudo apt-get update \$ sudo apt-get install postgresql-common \$ sudo apt-get install postgresql-9.5 libpq-dev</pre>
Para configurar um usuário no banco de dados:
<pre>\$ sudo -u postgres createuser imobilis -s \$ sudo -u postgres psql postgres=# \password imobilis</pre>
Foi necessário atualizar o arquivo config / database.yml da aplicação BOPE Game para coincidir com o nome de usuário e senha do banco de dados. Para acessar o sistema basta entrar na pasta cd BopeGame, levantar o servidor : Rails server -b 0.0.0.0 -p 8000. Agora você pode visitar http://200.239.154.218:8000 para ver o sistema BOPE Game.