**FACULDADE METODISTA GRANBERY – FMG**

**CURSO DE SISTEMAS DE INFORMAÇÃO**

Pablo Figueiredo Alberto

Ramon Phillipe Giovanelli da Silva

**TQUEST – UM JOGO SÉRIO PARA O AUXÍLIO DO ENSINO DE MÉTRICAS DE COMPLEXIDADE DE CLASSES EM TESTE DE SOFTWARE**

JUIZ DE FORA

2016

Pablo Figueiredo Alberto

Ramon Phillipe Giovanelli da Silva

**TQUEST – UM JOGO SÉRIO PARA O AUXÍLIO DO ENSINO DE MÉTRICAS DE COMPLEXIDADE DE CLASSES EM TESTE DE SOFTWARE**

Trabalho de conclusão de curso apresentado como requisito para obtenção do título de Bacharel Em Sistemas De Informação

ORIENTADOR: Prof. Marco Antônio Pereira Araújo

JUIZ DE FORA

2016

Pablo Figueiredo Alberto

Ramon Phillipe Giovanelli da Silva



**FOLHA DE APROVAÇÃO**

**TQUEST – UM JOGO SÉRIO PARA O AUXÍLIO DO ENSINO DE MÉTRICAS DE COMPLEXIDADE DE CLASSES EM TESTE DE SOFTWARE**

Após a exposição do(s) discente(s) Pablo Figueiredo Alberto, matrícula 202656 e Ramon P. Giovanelli da Silva, 202574, a Banca Examinadora, composta pelos professores abaixo identificados, reuniu-se e aprovou o presente artigo que, por atender aos requisitos estabelecidos, pelo curso de Bacharelado em Sistemas de Informação da Faculdade Metodista Granbery, para obtenção do título de Bacharel em Sistemas de Informação, recebeu a nota \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_, como sendo a média da Banca.

Juiz de Fora, 29 de Novembro de 2016.

Professor Dr. Marco Antônio Pereira Araújo – Orientador

Faculdade Metodista Granbery

Professora Débora Alvernaz Correa – Examinador

Faculdade Metodista Granbery

Professor Ronney Moreira de Castro – Examinador

Faculdade Metodista Granbery

JUIZ DE FORA

2016

**TQUEST – UM JOGO SÉRIO PARA O AUXÍLIO DO ENSINO DE MÉTRICAS DE COMPLEXIDADE DE CLASSES EM TESTE DE SOFTWARE**

Pablo Figueiredo Alberto

Ramon Phillipe Giovanelli da Silva

**RESUMO**

Com o passar dos anos as instituições educacionais têm buscado novos métodos de ensino para melhorar a compreensão e fixação dos conteúdos abordados em sala de aula. Um mecanismo que vem sendo utilizado para o auxílio do ensino é a informática para o apoio e aperfeiçoamento dos métodos tradicionais de aprendizado. Este trabalho tem como foco fazer uso da gamificação, uma técnica moderna que tem sua base de funcionamento no *design* de *games* de forma que, através dela, foi construído um jogo sério para o apoio ao ensino de métricas de Lorenz & Kidd, Chidamber & Kemerer aplicadas em classes no teste de software.

**Palavras-chave:** Gamificação, Jogo educativo, Jogo educacional, Teste de software**.**

***ABSTRACT***

*Over the years as educational institutions has sought new teaching methods to improve the understanding and fixation of content addressed in the room. One mechanism that is used for the teaching of teaching is informatics for the support and improvement of traditional methods of learning. This article focuses on making use of gamering, a modern technique that has its working base no games design, through it makes a serious game, which was developed to support the teaching of metrics by Lorenz & Kidd, Chidamber & Kemerer Applied In software testing.*

***Keywords:*** *Social responsibility. FaMIdade Project. University social responsibility.*

\*Bacharelado em Sistema de Informação – Faculdade Metodista Granbery.

**1 INTRODUÇÃO**

A tarefa de Teste de Software tem por finalidade a obtenção da qualidade de um produto de software. Esse ramo da Engenharia de Software garante que o produto final que está sendo desenvolvido irá comportar-se da mesma maneira que foi planejado em seus requisitos (Silva, 2010). Porém, esse tema tão primordial não está tão presente nos cursos de computação da maneira que deveria. Segundo Valle *et al.* (2015) a indústria de TI (Tecnologia da Informação) tem se deparado com uma carência de profissionais especializados na área de Teste de Software. A causa para esse problema pode estar na forma que os profissionais receberam seu treinamento nas instituições de ensino. Ribeiro (2014) menciona que a falta de prática e compreensão adequada do tema causa problemas ao mercado de trabalho, impossibilitando que as empresas contratem profissionais recém-formados que ainda não possuem conhecimento e experiência necessária para ingressar no time de Teste de Software dessas empresas. Para reforçar o problema, Silva *et al.* (2011) relatam que alunos que se formam em cursos de computação aprendem mais sobre o tema após terem saído da instituição onde se formaram, demonstrando que há um problema na forma tradicional que a disciplina está sendo abordada nas instituições de ensino, não sendo eficientes ao ponto para que os alunos que têm interesse em trabalhar na área de Teste de Software ingressarem ao mercado de trabalho assim que se formam.

Este trabalho tem como objetivo apresentar os benefícios do uso do *design* de *games*, pilar da Gamificação, para a criação de um jogo sério tendo como foco apoio ao ensino-aprendizagem de métricas de Teste de Software criadas por Lorenz & Kidd e Chidamber & Kemerer. Tendo esses benefícios de aprendizado em vista, Michael *et al.* (2006, apud Silva *et al.*, 2011) afirmam que os jogos sérios proporcionam um modelo inovador de ensino e treinamento, mesclando jogos eletrônicos ao ensino. Podendo, desta forma, ir muito além das limitações das vídeo-aulas e livros, proporcionando não apenas aprendizado, mas também uma forma para que os jogadores possam colocar em prática os conceitos aprendidos.

A divisão deste documento foi feita da seguinte maneira: na seção 2 é apresentado o referencial teórico que foi a base para a pesquisa e criação deste artigo; na seção 3, será abordada a proposta e motivação para a criação do jogo, qual ferramenta foi utilizada para o desenvolvimento e contextualização e tarefas propostas no jogo; na seção 4 são descritos os resultados coletados com a aplicação do jogo; e por fim, a seção 5 apresenta as considerações finais do trabalho.

**2 REFERENCIAL TEÓRICO**

Nos últimos anos, tem se observado um grande aumento do uso da tecnologia na educação, a informática está se tornando parte essencial do aprendizado, alcançando resultados bem satisfatórios nas pesquisas realizadas na área do ensino. De acordo com Borges *et al.* (2015) inúmeras pesquisas envolvendo a aplicação de tecnologias computacionais estão sendo usadas em ambientes educacionais para se aperfeiçoar o processo de ensino-aprendizagem. Com o passar do tempo novas práticas de ensino foram surgindo, pareadas com o avanço tecnológico, sendo assim permitiu-se a introdução dos jogos educacionais ou jogos sérios nas instituições de ensino, cujo o objetivo é apoiar os professores e estudantes no processo de ensino-aprendizagem (Oliveira, 2013).

A expressão “jogos sérios” une a seriedade do pensamento e da resolução de problemas com o experimental e a liberdade emocional do jogo, sendo assim, os jogos sérios combinam a concentração analítica e questionadora do ponto de vista científico com a liberdade intuitiva e recompensa construtiva dos atos artísticos Abt (1987, apud Silva *et al*., 2011, p. 5).

Para comprovar a eficiência da utilização de jogos sérios foi feita uma revisão sistemática da literatura (Kitchenham, 2004), utilizando o mecanismo de busca do Google Acadêmico através da *string* de busca:

(“*gamification*” <*OR*> “gamificaçao” <*OR*> “*educational game*” <*OR*> “jogo educativo” <*OR*> "jogo educacional") <*AND*> (“*software test*” <*OR*> “teste de software”)

Através da busca realizada no Google Acadêmico foram encontrados 126 artigos relacionados que possuem algum contexto similar ao tema deste trabalho, porém após ser aplicado os critérios de inclusão e exclusão, foram selecionados 12 artigos de maior relevância, dentre eles foram encontrados alguns dos jogos sérios como TestEG, iLearnTest, MaintES, iTest Learning, JETS, U-TEST, SimulES, Jogo das 7 Falhas, esses são alguns exemplos de jogos que já estão sendo usados para o auxílio no ensino-aprendizagem da área de Engenharia de Software, quase em sua maioria foram testados em sala de aula por seus respectivos criadores, onde foi constatado nos resultados das avaliações que todos os alunos que utilizaram os jogos acima descritos obtiveram um aproveitamento superior de aprendizado em comparação de quando somente utilizavam a forma convencional de ensino.

**2.1 TRABALHOS RELACIONADOS**

O jogo TestEG (Oliveira, 2013) se passa em um cenário empresarial, onde o jogador controla um Gerente de Teste e recebe um orçamento inicial para contratar sua equipe que é constituída por três funcionários. Ao decorrer do jogo o gerente deve solucionar as dúvidas de sua equipe dando as informações necessárias para o término do trabalho. O jogador pode ministrar treinamentos para os funcionários contratados, acompanhar o desempenho da equipe e consultar um material sobre teste de software. Quando um treinamento acontece é feito um desconto no orçamento proporcional ao tipo do mesmo. Esse desconto ocorre a cada dois minutos de jogo, tempo equivalente a 1 mês de trabalho. O jogo possui um *ranking* em que os jogadores podem acompanhar o desempenho dos outros jogadores. No decorrer do jogo devem ser respondidas dentro do tempo de 10 minutos, sem que ocorra o término do orçamento.

O jogo iLearnTest (Ribeiro, 2014) foi desenvolvido com a ideia de auxiliar no ensino de Teste de Software, porém seu objetivo principal é apoiar os usuários que estão interessados em realizar o exame de obtenção do nível base de certificação do ISTQB (*International Software Testing Qualification Board*) primeiro nível do programa de certificação internacional em testes de software. O sistema oferece desafios em cada matéria para atrair cada vez mais a atenção dos estudantes durante a prática dos conhecimentos e técnicas de teste de software, ele também possibilita que os usuários estudam sozinhos e aprendam em seu próprio ritmo, utilizando o tempo que acharem necessário para a aprendizagem, possui também um sistema de pontuação a cada conclusão, incentivando o aluno a se obter pontuação cada vez maior.

No jogo MaintES (Carvalho, 2014) o usuário assume um personagem de um estudante que busca vencer torneios de perguntas. O *game* possui três níveis de dificuldade e suas perguntas estão todas relacionadas com a temática de manutenção de software. O usuário inicia o jogo na dificuldade iniciante, sem pontos e pouca quantidade de dinheiro. Na posição inicial em que se encontra, o jogador poderá ir respondendo a perguntas referentes aos temas do seu nível atual e, dessa forma, pode ir aumentando sua pontuação e ganhar dinheiro. Após certa quantia de pontos e dinheiro for alcançado, o jogador poderá pagar sua inscrição para participar de um torneio. Com o passar do tempo, o dinheiro acumulado pelo jogador poderá ser utilizado na contratação de funcionários para a sua equipe. Possuindo uma equipe o estudante pode eliminar alternativas falsas nas perguntas feitas nos torneios, tornado sua jornada mais fácil. A evolução do nível só acontece quando o jogador vence o torneio de seu nível atual e possui pontuação adequada para a próximo nível. Para se alcançar o nível máster no jogo, o usuário deverá vencer três torneios.

O jogo iTest Learning (Faria *et al.*, 2012) foi desenvolvido com o objetivo de auxiliar no ensino de teste de software na fase de planejamento, possui 3 dificuldades (fácil, médio e difícil), após a escolha da dificuldade o estudante deverá selecionar um projeto referente ao nível de dificuldade. Suas atividades giram em torno de uma breve descrição de um projeto hipotético em um ambiente simulado para o planejamento de teste, o jogo também disponibiliza ao usuário tirar dúvidas em relação aos conceitos que são abordados em cada fase e visualizar a descrição do projeto em que está desenvolvendo o plano de teste sem a necessidade de sair da fase. O principal público alvo são alunos de graduação de cursos da área de computação.

JETS (Silva *et al.*, 2011) é um jogo *multiplayer*, porém pode-se jogar *single player* e foi desenvolvido sobre uma plataforma 3D que simula o setor de uma empresa fictícia de desenvolvimento de software. Constituído de quatro fases, sendo que em cada fase o estudante assumirá um cargo diferente dentro de uma equipe de teste. Na primeira fase, o estudante ingressa no jogo com o cargo de testador de software; na segunda fase, assume o cargo de analista de teste e, em seguida, é promovido ao cargo de arquiteto de teste (terceira fase); na quarta e última fase do jogo, o estudante assume o cargo de líder (gerente) da equipe de teste. Seu objetivo é aperfeiçoar o conhecimento de estratégias de Teste de Software, fornecendo ao jogador uma visão mais ampla de como o teste de software pode ser implementado em uma instituição, auxiliando os estudantes com as temáticas de teste que são abordadas nas grades dos cursos de computação.

U-Test (Silva, 2010) é um jogo de simulação cujo objetivo é auxiliar no ensino de teste de software, com foco voltado para teste de unidade, o *game* aborda situações práticas e teóricas. No jogo o usuário assume o personagem de um testador *freelancer* ou empregado cujas funções são elaborar testes de unidade para um sistema hipotético. O principal objetivo U-Test é a seleção de dados de entrada para o teste de unidade. Ao decorrer do jogo, o estudante deverá colocar em prática seus conhecimentos em teste de software, como particionamento em classes de equivalência e análise de valor limite. Ao final do jogo o usuário recebe informações de seu desempenho ao longo do jogo que poderão ser críticas ou elogios.

O SimulES (Figueiredo *et al.*, 2006) é baseado em um jogo educacional desenvolvido pelo departamento de informática da Universidade da Califórnia chamado PnP (*Problems and Programmers*), um jogo de cartas voltado para a área de Engenharia de Software. Sua abordagem demonstra aos estudantes o processo de desenvolvimento de sistemas desde a concepção até a fase de entrega do software. PnP é um jogo direcionado para iniciantes, porém ao decorrer do jogo são apresentados conceitos avançados. Pode ser jogado por 4 a 8 jogadores, o objetivo do jogo é que os usuários disputam para concluir um projeto de software, o vencedor será quem implementar o projeto primeiro, através do jogo os estudantes aprendem importantes conceitos sobre Engenharia de Software. O jogo utiliza alguns recursos como cartões de projeto, um tabuleiro, cartas e um dado. Os cartões de projeto possuem descrição textual do mesmo, a forma de composição de módulos e referências para bibliografia relacionada, o tabuleiro é a área que cada jogador coloca seus engenheiros de software em colunas e os artefatos em linhas e as cartas são os principais recursos do jogo, elas possuem quatro tipos: problemas, conceitos, engenheiros de software e artefatos.

O objetivo do jogo das 7 falhas (Diniz *et al.*, 2011) é descobrir as falhas existentes nas funcionalidades de um sistema a ser testado em menos tempo possível. O jogo possui dois níveis de complexidade, baixa e média, a cada nível existem sete falhas a serem descobertas. O jogador só passará para o próximo nível quando descobrir as 7 falhas dentro de um tempo limite de 25 minutos para o nível 1 e 40 minutos para o nível 2. Se o jogador não conseguir identificar as falhas dentro do tempo estimado é eliminado e o jogo é encerrado. Caso consiga descobrir as 7 falhas nos dois níveis no tempo estimado, vence o jogo e alcança o objetivo proposto pelo mesmo.

**3 PROPOSTA DO TQUEST**

Apesar dos jogos que foram descritos no referencial teórico, não foi encontrado na literatura nenhum jogo que aborde as métricas criadas por Lorenz & Kidd (Lorenz et al, 1994), Chidamber & Kemerer (Chidamber, 1994). Sendo assim, foi decidida pela criação de um jogo sério que tenha como foco o auxílio ao ensino dessas métricas.

A motivação para criar um jogo sério para o auxílio do ensino pode ser constatada por uma citação de Ribeiro (2014) que diz que os jogos fazem parte da vida dos humanos desde os tempos mais remotos, estando presentes não só na infância, como na idade adulta. Ao longo dos últimos anos, estudos realizados comprovam que o uso de jogos como um complemento à tradicional aprendizagem é muito mais eficiente do que apenas a utilização do tradicional método de ensino. Os jogos podem ser ferramentas educacionais eficientes, uma vez que, enquanto divertem e motivam o utilizador, facilitam a aprendizagem e aumentam a capacidade de retenção do que foi ensinado, exercitando a mente e os pensamentos intelectuais de seus jogadores.

O jogo criado para o apoio a este trabalho se chama TQuest. Baseia-se no início de carreira de um jovem chamado "Bit", que está finalizando sua graduação em Sistemas de Informação. Com sua dedicação e esforço, consegue uma vaga de estágio em uma fábrica de software de grande porte, chamada *Recycle Factory*, referência na área. O setor no qual o protagonista ingressará é a de Teste de Software, na qual terá a oportunidade de aplicar seu conhecimento nas métricas de Teste de software criadas por Lorenz & Kidd e Chidamber & Kemerer. O objetivo de Bit é ser contratado. Porém, para que isso se concretize, terá que mostrar suas habilidades e conhecimentos em vários desafios que lhe serão propostos durante o jogo, aplicando corretamente cada técnica em seu momento apropriado.

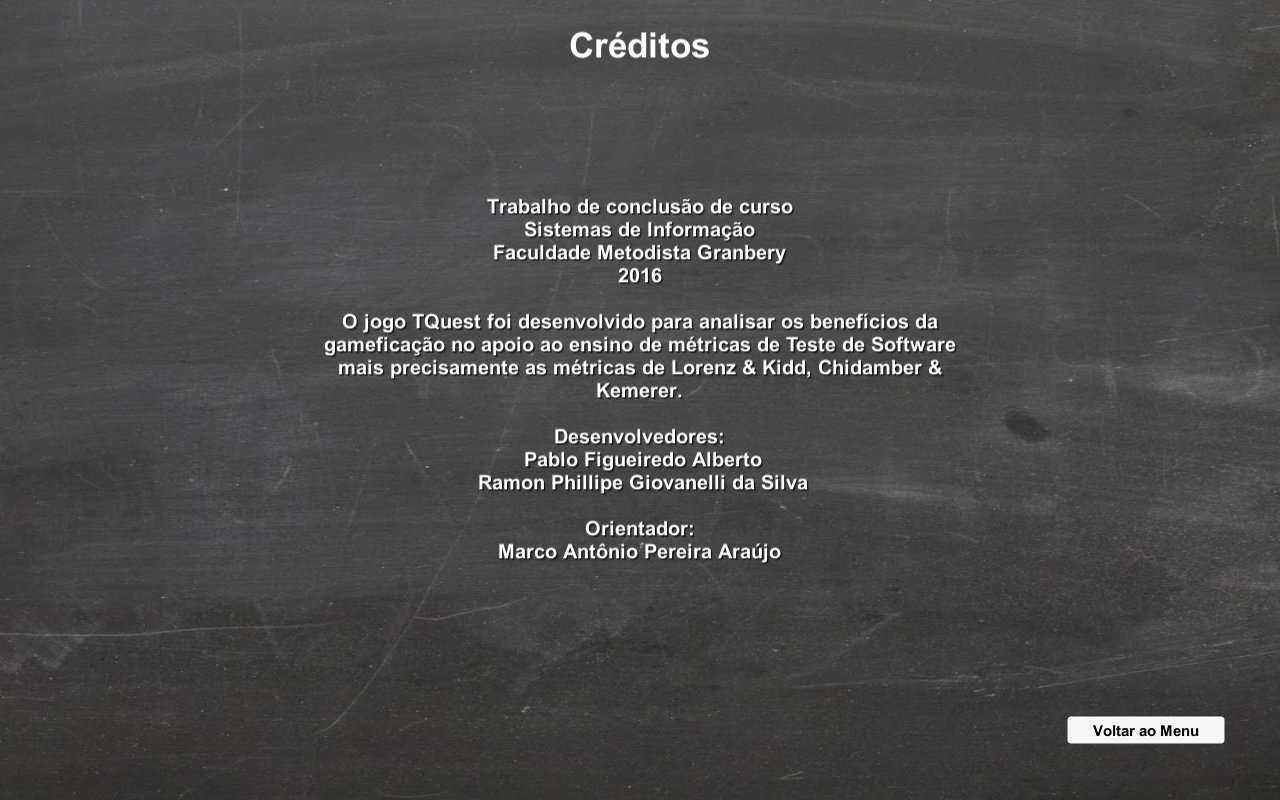
Como demonstrado na Figura 1, o jogo inicia apresentando uma tela onde o jogador poderá escolher iniciar o jogo, visitar a tela de Créditos ou Sair do jogo.

Figura 1. Tela inicial do jogo TQuest.



Caso queira visitar a tela de Créditos lhe será apresentado uma tela contendo as informações dos desenvolvedores do projeto e algumas informações complementares (Figura 2).

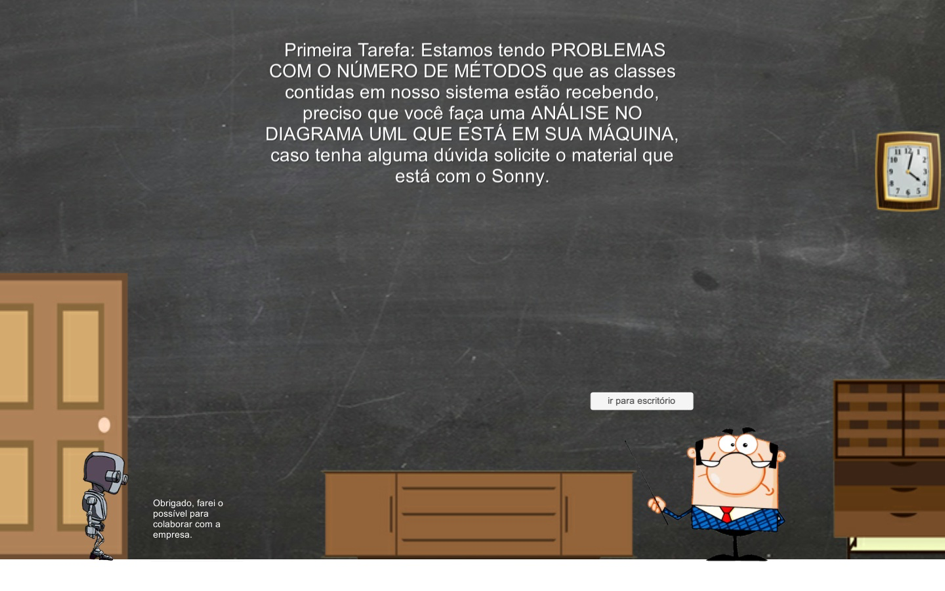
Figura 2. Tela de Créditos.



Escolhendo a opção "Iniciar" será apresentado ao jogador uma tela contendo as informações iniciais do jogo, como a história do personagem protagonista. Após o jogo ter encerrado de apresentar essas informações, aparecerão para o usuário dois botões, onde poderá voltar ao menu e outro para começar o jogo. A tela pode ser vista na Figura 3.

Figura 3. Tela de apresentação do jogo.

Como mencionado anteriormente, quando a tela de apresentação terminar de expor as informações iniciais serão apresentados dois botões, caso pressione o botão "Voltar ao Menu", será mandado para a tela inicial do jogo, caso escolha o botão "Começar o Jogo", será mandado para a sala do gerente de projetos. É nesta cena que o jogador receberá todas as tarefas do jogo, que serão descritas detalhadamente mais adiante neste trabalho. Quando o gerente de projetos terminar de entregar a tarefa ao jogador, terá que pressionar o botão "Ir para o Escritório" que se encontra próximo ao gerente de projetos. A tela que demonstra a sala do gerente de projetos pode ser vista na Figura 4.

 Figura 4. Escritório do Gerente de Projeto.

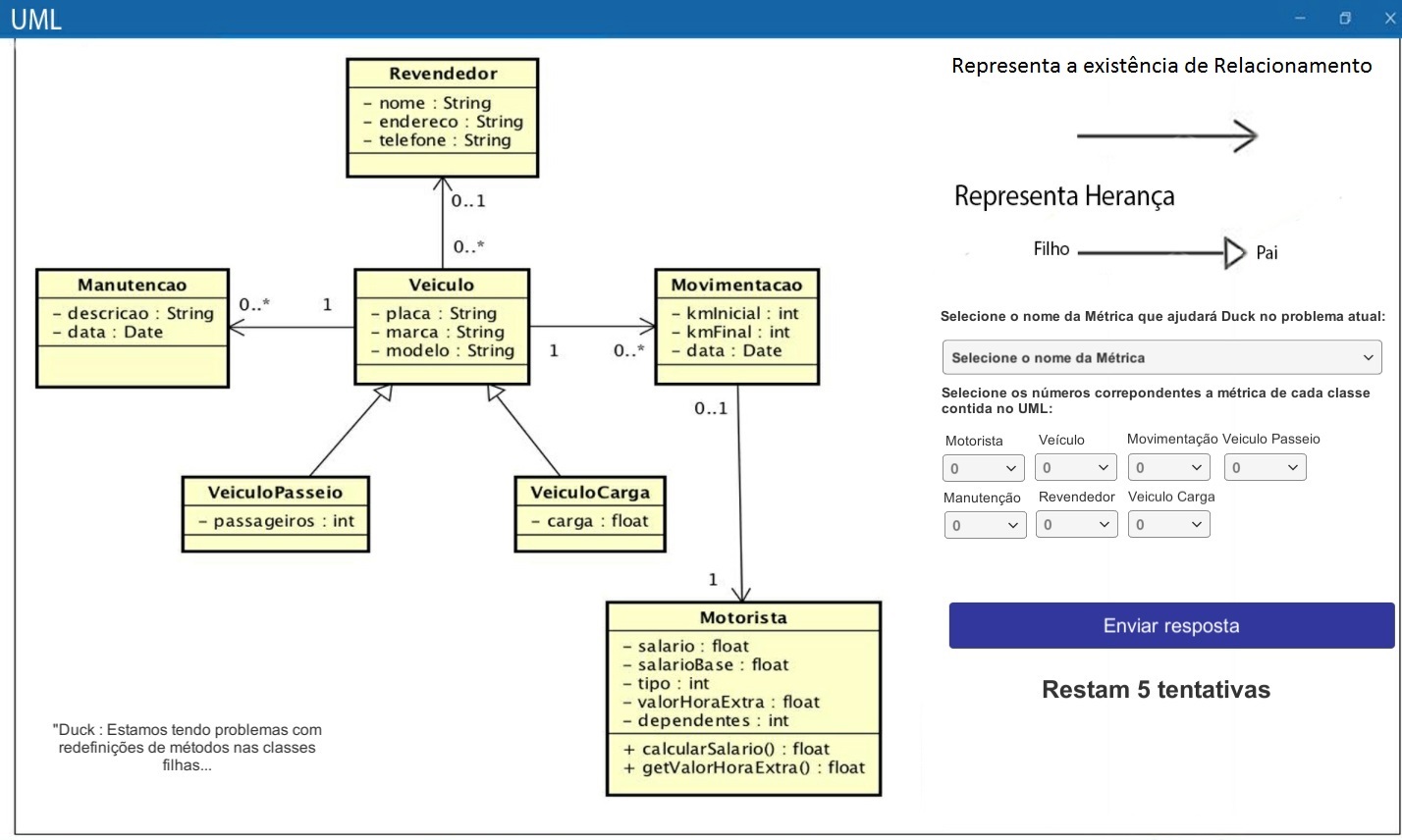
Quando o usuário pressionar o botão "Ir para o Escritório", será mandado para o escritório de Sonny, nesta sala há um computador onde o jogador irá executar as tarefas que foram dadas pelo gerente de projetos. Para acessar o computador basta o usuário se aproximar da máquina que lhe será apresentado uma mensagem para pressionar a tecla “Enter”. O processo para acessar as tarefas no computador pode ser visto na Figura 5.

 Figura 5. Escritório de Sonny.

Ao acessar o terminal, será apresentado ao usuário uma tela com dois componentes, um botão para sair do terminal e um ícone para acessar o diagrama de classes UML com a tarefa. Caso o usuário clique no botão sair, será enviado de volta para o escritório de Sonny. Caso clique no ícone referente ao diagrama UML (*Unified Modeling Language*), será enviado para a tela onde terá que trabalhar na tarefa que lhe foi confiada. A tela onde o jogador é enviado quando acessa o terminal pode ser vista na Figura 6.

 Figura 6. Tela de seleção do diagrama UML no computador.

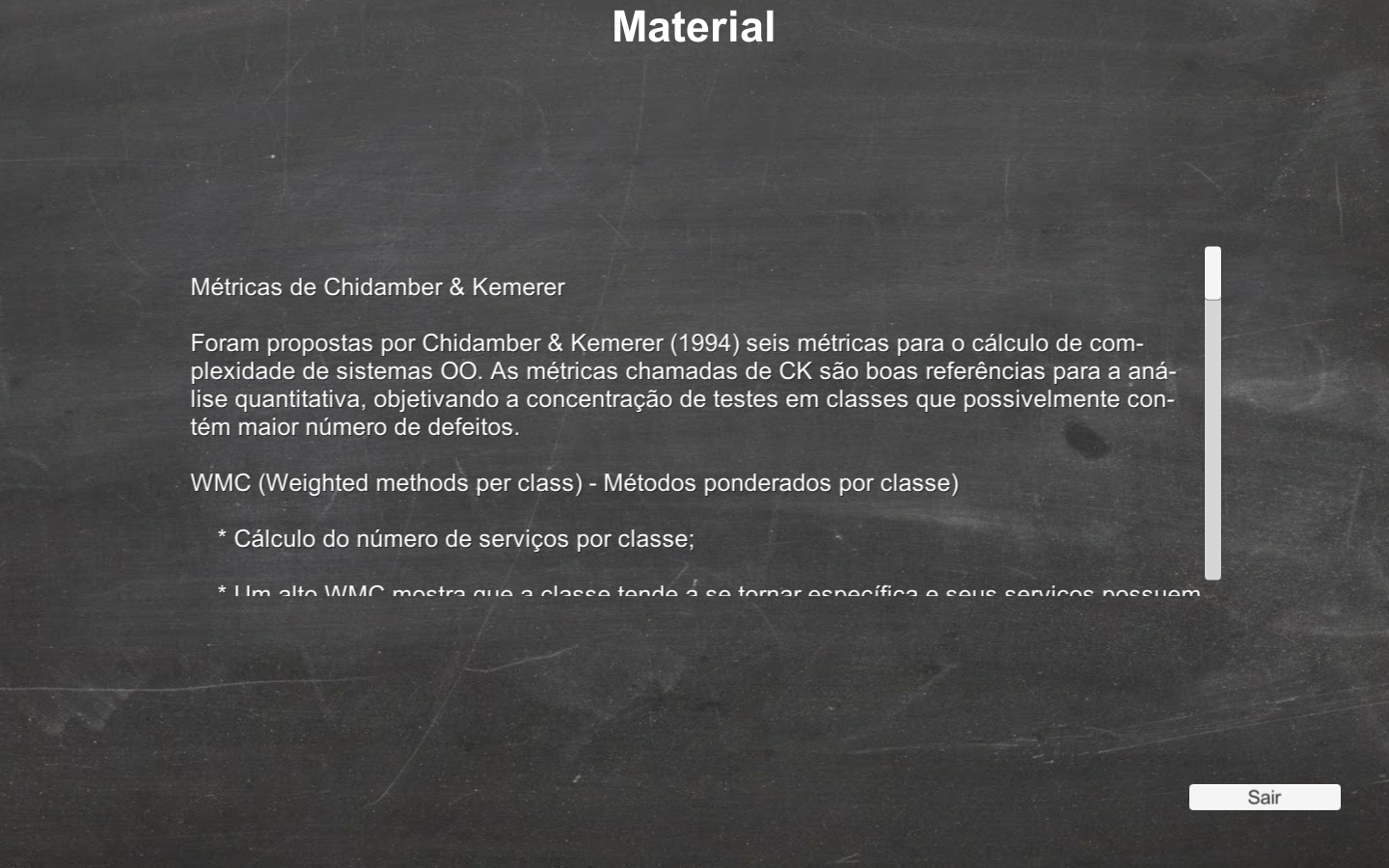
Ao clicar no ícone UML o jogador será enviado para a tela de tarefas, é nesta tela que serão apresentadas todas as sete primeiras tarefas do jogo. A oitava e última tarefas terão uma tela diferenciada, pois ao invés do usuário inserir as métricas terá que selecionar a(s) classe(s) mais complexas do sistema. Nessa tela será apresentado um diagrama UML e uma dica referente à tarefa, o jogador verá também um *ComboBox* onde terá que selecionar o tipo de métrica que irá utilizar, mais cinco *ComboBox* referentes às classes do diagrama UML possibilitando a seleção para cada classe um inteiro da respectiva métrica selecionada e, por fim, um botão "Enviar Resposta". Essa tela é construída dinamicamente a partir do diagrama de classes exportado por ferramentas CASE (*Computer Aided Sotware Engineering*) como a Star UML (Rumbaugh *et al.*, 2004) através da exportação do modelo através de arquivos XML, importados e interpretados pela TQuest. O jogador possui cinco tentativas de acerto, caso ele erre todas as tentativas o botão de "Enviar resposta" será bloqueado. Para que o botão seja desbloqueado aparecerá para o usuário um botão para que vá falar com o Sonny. A tela de tarefas pode ser conferida na Figura 7.

 Figura 7. Tela de tarefas.

Para desbloquear o botão de "Enviar Resposta", na tela de tarefas o usuário terá que se aproximar do personagem Sonny e pressionar a tecla "Enter”, assim aparecerão dois botões, desbloquear PC que possibilita ao jogador voltar a tela de tarefas e continuar tentando resolver a missão que lhe foi proposta. O segundo é um botão que leva o usuário a uma tela onde poderá consultar um material informativo que explica todas as métricas contidas nas tarefas do jogo. O processo de desbloqueio do botão e a tela de material pode ser conferido nas telas das Figuras 8 e 9, respectivamente.

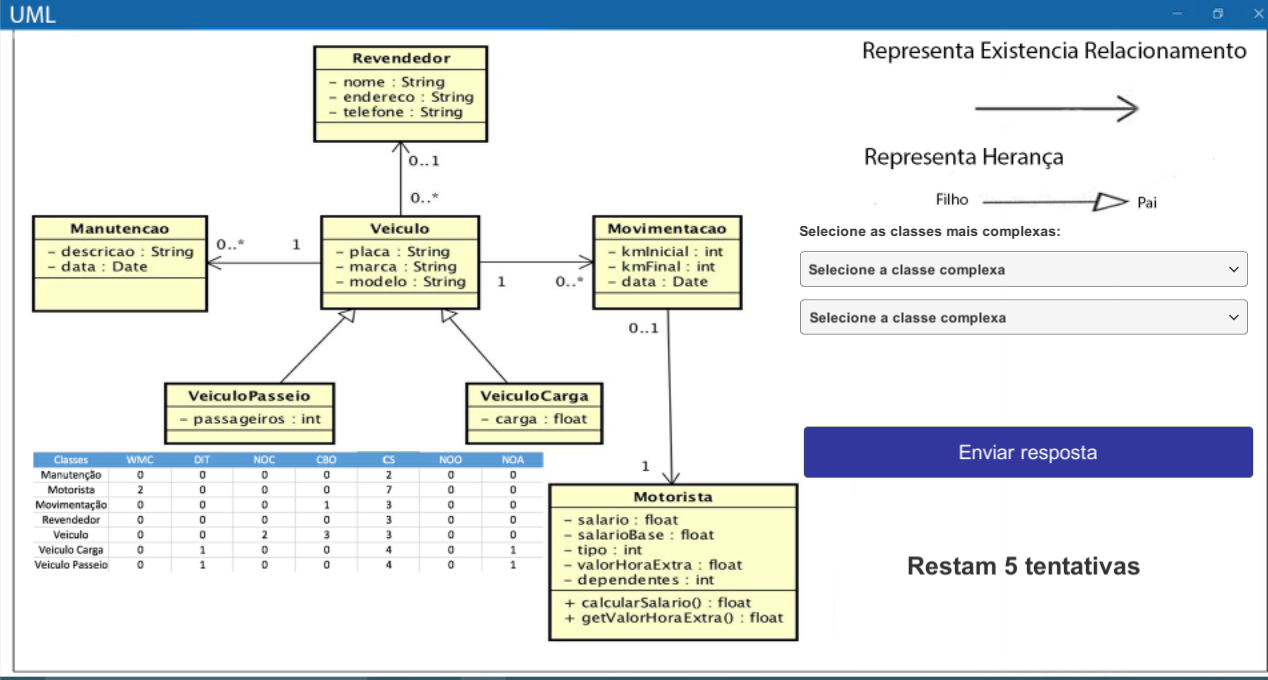
Figura 8. Tela de desbloqueio de máquina.

Figura 9. Tela de material.



O jogador não pode deixar a máquina bloqueia mais de três vezes, caso isso aconteça o jogo se encerra e o jogador é enviado para a tela de *game over*.

Para a última missão, o jogador precisará acessar o computador da mesma maneira, só que desta vez uma nova tela de tarefas lhe é apresentado onde terá que responder quais são as classes mais complexas do diagrama de classes. A tela da oitava e última missão pode ser vista na Figura 10. Caso o jogador acerte essa tarefa é enviado para a tela de congratulação de vencedores do jogo.

 Figura 10. Tela da última tarefa.

O jogo possui 8 tarefas no total, sendo que as quatro primeiras utilizam as métricas criadas por Chidamber & Kemerer e as três posteriores são as métricas criadas por Lorenz & Kidd consecutivamente.

A primeira tarefa é sobre a métrica WMC (*Weighted methods per class* – Métodos ponderados por classe) onde o jogador terá que responder quantos métodos cada classe possui. A segunda tarefa é sobre a métrica DIT (*Depth of the inheritance tree* – Profundidade da árvore de herança) o jogador irá terá que calcular o número máximo de superclasses posicionadas hierarquicamente acima da classe em questão. A terceira tarefa é sobre a métrica NOC (*Number of Children* – Número de filhos) o jogador terá que calcular o número de subclasses posicionadas imediatamente abaixo da superclasse em questão, ou a quantidade de filhos diretos. A quarta tarefa é sobre a métrica CBO (*Coupling between object classes* – Acoplamento entre classes de objetos) será calculado pelo jogador quantas ligações a classe em questão possui. A quinta tarefa é sobre a métrica CS (*Class size* – Tamanho de classe), nessa tarefa o jogador irá calcular o número de métodos e atributos da classe em questão e todos os métodos e atributos por ela herdados. A sexta tarefa é sobre a métrica NOO (*Number of operations overridden by a subclass* – Número de operações redefinidas por uma subclasse) será analisado quantos métodos da subclasse em questão foram redefinidos dos métodos originais da superclasse. A sétima tarefa é sobre a métrica NOA (*Number of operations added by a subclass* – Número de operações adicionadas por subclasses) onde o jogador irá analisar o número de novas operações e atributos definidos na subclasse que não existem na superclasse (Araújo, 2015).

A oitava tarefa o jogador terá que analisar quais as classes que são mais complexas no sistema.

**4 AVALIAÇÃO DO JOGO TQUEST**

O jogo sério TQuest foi aplicado para um grupo de vinte alunos de curso de Bacharelado em Sistemas de Informação, sendo que sete alunos eram do sétimo período e treze alunos do oitavo período. O jogo foi aplicado de forma igualitária para todos os participantes, tendo que jogar e resolver todas as tarefas propostas. Em suma, todos os participantes conseguiram chegar ao final do jogo.

O questionário avaliativo também abordou o índice de jogadores que conseguiram chegar até o final do jogo com menos de 5 erros nas questões, porém em suma todos os alunos que foram submetidos ao teste conseguiram chegar ao final sem cometer esse número de erros.

Após a aplicação, os participantes tiveram que avaliar uma série de afirmações em um questionário, dando uma nota que ia de 1 a 5, sendo que 1 para discordo totalmente e 5 para concordo totalmente.

As perguntas do questionário foram:

1-O jogo desperta interesse.

2-O jogo proporciona uma forma diferenciada de aprendizado.

3-Ao jogar me senti motivado a aprender mais sobre métricas de teste de software.

4-O jogo apresenta uma metodologia diferenciada de ensino.

5-Os jogos sérios podem ser grandes aliados ao ensino.

6-Atividades com jogos sérios deveriam estar mais presentes no cotidiano dos alunos.

7-A atividade proporcionou uma maior proximidade do conteúdo abordado com o seu cotidiano.

8-Atividades com jogos sérios são úteis para se fixar o conteúdo das aulas.

9-O contexto das atividades propostas no jogo estão bem constituídas.

10-Ao terminar o jogo consegui adquirir algum conhecimento em métricas de teste de software.

O resultado do teste pode ser avaliado através do gráfico da Figura 11.

Figura 11 – Dados do resultado geral da pesquisa**.**

Com os dados coletados no processo avaliativo pode-se dizer que o jogo desenvolvido obteve resultados positivos no processo de ensino-aprendizagem. Os alunos que o jogaram ficaram motivados a aprender mais sobre as métricas criadas por Chidamber & Kemerer e Lorenz & Kidd, e também relataram que após terem terminado de jogar fixaram o conteúdo abordado com mais facilidade que em aulas tradicionais que lecionam sobre o tema.

Resultados acima demonstram que o jogo desenvolvido cumpriu o seu papel, proporcionando uma forma diferenciada de ensino, abordando o mesmo tema de métricas de Teste de Software, porém de forma diferenciada e dinâmica, atraindo a atenção e o interesse dos alunos que o utilizam.

**5 CONSIDERAÇÕES FINAIS**

Neste trabalho buscou-se desenvolver um jogo sério para se avaliar o quanto este método pode ser benéfico ao auxílio ao ensino de métricas de Teste de Software, proporcionando aos alunos uma forma prática em um ambiente simulado uma maneira de colocar em execução o conhecimento que adquirem durante a aplicação das aulas ministradas tradicionalmente. Para se chegar aos resultados obtidos no desenvolvimento do jogo e também com a escrita deste artigo foi feito uma pesquisa minuciosa nas obras disponíveis na literatura que foram encontradas através do uso de uma *string* de busca, após ter sido realizado o processo de inclusão e exclusão dos artigos encontrados foi feito um referencial teórico.

Com os resultados obtidos após a aplicação do jogo TQuest conclui-se que o jogo alcançou os objetivos esperados por seus desenvolvedores, sendo uma ferramenta eficaz para o auxílio no processo de ensino-aprendizagem das métricas criadas por Chidamber & Kemerer e Lorenz & Kidd.

Para melhor aproveitamento do jogo criado é planejado um aperfeiçoamento do mesmo tirando maior proveito do seu potencial no apoio ao ensino. Como meta de trabalhos futuros pretende-se acrescentar alguns novos recursos ao TQuest, sendo: novos níveis de dificuldade, recursos para pontuação e ranqueamento, novos cenários e métricas para o ensino de Teste de Software e até mesmo outros contextos relacionados à Engenharia de Software.

**REFERÊNCIAS**

ARAÚJO, Marco A. P., 2016. Apostila Teste em Software Orientado a Objetos.

BORGES, Vanessa A.; Nogueira, Bruno M.; Barbosa, Ellen, 2015. Uma análise exploratória de tópicos de pesquisa emergentes em Informática na Educação.

CARVALHO, Moreira C., 2014. MaintES – Um jogo educacional para apoiar o processo de ensino aprendizagem sobre manutenção de software.

DINIZ, Lucio L.; Dazzi, Rudimar, L. S., 2011. Jogo para o Apoio ao ensino do teste de Caixa-Preta.

FARIAS, Virgínia; Moreira, Carla; Coutinho, Emanuel; Santos, Ismayle S., 2012 iTest Learning: Um Jogo para o Ensino do Planejamento de Testes de Software.

FIGUEIREDO, E.; Lobato, C. A.; Dias, K.; Leite, J.; Lucema, C., 2006 – SimulES: Um jogo para o ensino de Engenharia de Software.

OLIVEIRA, Bruno C., 2013. TestEG – Um software educacional para o ensino de teste de software.

RIBEIRO, Tânia P. B., 2012. iLearnTest: Jogo Educativo para Aprendizagem de Testes de Software.

SILVA, Antônio C., 2010. Jogo educacional para apoiar o ensino de técnicas para elaboração de testes de unidade.

SILVA, Tarcila G.; Müller, Felipe M; Bernardi, Giliane, 2011. Panorama do Ensino de Engenharia de Software em Cursos de Graduação Focado em Teste de Software: Uma Proposta de Aprendizagem Baseada em Jogos.

VALLE, Pedro H. D.; Barbosa, Ellen; Maldonado, José C., 2015. Um Mapeamento Sistemático Sobre Ensino de Teste de Software.

UNITY. Disponível em: 21/11/2016, Acesso em: <https://unity3d.com/pt>

RUMBAUGH, J; Jacobson, I; Booch G., 2004. Unified Modeling Language Reference Manual, The (2nd Edition)