



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ
BACHARELADO EM ENGENHARIA DE SOFTWARE

FRANCISCO RONIELSON OLIVEIRA DO NASCIMENTO

**ELABORAÇÃO DE UM JOGO SÉRIO SOBRE BEHAVIOR DRIVEN
DEVELOPMENT PARA ALUNOS DE TI**

QUIXADÁ
2022

FRANCISCO RONIELSON OLIVEIRA DO NASCIMENTO

ELABORAÇÃO DE UM JOGO SÉRIO
SOBRE BEHAVIOR DRIVEN DEVELOPMENT PARA ALUNOS DE TI

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Orientador: prof.^a. Dra. Paulyne Matthews Jucá

QUIXADÁ

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

N195e Nascimento, Francisco Ronielson Oliveira do.

Elaboração de um jogo sério sobre Behavior Driven Development para alunos de TI /
Francisco Ronielson Oliveira do Nascimento. – 2022.

74 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus
de Quixadá, Curso de Engenharia de Software, Quixadá, 2022.

Orientação: Profa. Dra. Paulyne Matthews Jucá.

1. Jogos educativos. 2. Teste de Software. 3. Desenvolvimento guiado por
comportamento. 4. Alunos. 5. Educação. I. Título.

CDD 005.1

FRANCISCO RONIELSON OLIVEIRA DO NASCIMENTO

ELABORAÇÃO DE UM JOGO SÉRIO
SOBRE BEHAVIOR DRIVEN DEVELOPMENT PARA ALUNOS DE TI

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software do Campus de Quixadá da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia de Software.

Aprovada em: __/__/_____.

BANCA EXAMINADORA

Prof^a. Dra. Paulyne Matthews Jucá (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Me. Jefeson Kenedy Morais Vieira
Universidade Federal do Ceará (UFC)

Prof. Me. Virgínia Farias de Sousa
Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

A minha família pelo apoio recebido, ensinamentos, dedicação, proteção e amor.

A prof.^a Dra. Paulyne Matthews Jucá, pela excelente orientação.

Aos colegas da turma de graduação, pelas reflexões, críticas e sugestões recebidas.

“Inteligência é a capacidade de se adaptar
à mudança.”

(Stephen Hawking)

RESUMO

O ensino prático de assuntos sobre testes não é realizado de maneira ideal, os principais motivos são a falta de tempo para abordar sobre o assunto da melhor maneira possível e a diversificação de disciplinas oferecidas durante a graduação. Fazendo com que algumas matérias não tenham o mesmo nível de atenção durante a graduação e a área de testes se inclui nesse aspecto. Esse trabalho mostra a produção de um jogo sério analógico sobre um dos temas do campo de testes, o processo de BDD (Desenvolvimento Orientado ao Comportamento), então o objetivo deste trabalho foi a criação de um jogo sério analógico onde de maneira lúdica tentará melhorar o nível de conhecimento sobre o processo de BDD. Foram analisados aspectos que envolvem esse processo, o trabalho se propôs em adaptá-lo para um contexto de jogo.

Foram feitos questionários e avaliações para captar a percepção dos participantes em relação ao jogo. Os resultados se mostraram promissores e positivos, já no que se diz respeito à abordagem do tema no contexto de um jogo sério, podemos concluir que esse objetivo foi alcançado.

Palavras-chave: jogos sérios; teste de software; BDD; jogos analógicos; educação.

ABSTRACT

Practical teaching of test subjects is not done optimally, the main reasons are the lack of time to approach the subject in the best possible way and the diversification of disciplines offered during graduation. As a result, some subjects do not have the same level of attention during graduation.

This work shows the production of an analogue serious game about one of the themes of the test field, the process of BDD (Behavior Oriented Development), so the objective of this work was the creation of a serious analog game where in a playful way it will try to improve the level of knowledge about the BDD process. Aspects involving this process were analyzed, the work proposed to adapt it to a game context.

Questionnaires and assessments were made to capture the participants' perception of the game. The results were promising and positive, as far as approaching the theme in the context of a serious game is concerned, we can conclude that this objective was achieved.

Keywords: serious games; software testing; BDD; analogue games; education.

LISTA DE FIGURAS

Figura 1 – Ciclo do TDD por Kent Beck	22
Figura 2 – Esquema do ciclo de testes de Sommerville	24
Figura 3 – Ciclo BDD.....	26
Figura 4 – Marcador de risco.....	31
Figura 5 – Primeira casa do tabuleiro.....	33
Figura 6 – Carta de User Story.....	33
Figura 7 – Segunda casa do tabuleiro.....	34
Figura 8 – Cartas de stateholders	35
Figura 9 – Cartas de stakeholders	35
Figura 10 – Terceira casa do tabuleiro.....	36
Figura 11 – Cartas que compõem os critérios de aceitação	36
Figura 12 – Quarta casa do tabuleiro	37
Figura 13 – Carta de testes	38
Figura 14 – Quinta casa do tabuleiro	38
Figura 15 – Carta de código	39
Figura 16 – Sexta casa do tabuleiro	39
Figura 17 – Sétima casa do tabuleiro.....	40
Figura 18 – Carta de refatoração	41
Figura 19 – Oitava casa do tabuleiro.....	41
Figura 20 – Nona casa do tabuleiro	42
Figura 21 – Decima casa do tabuleiro.....	43
Figura 22 – Avaliação da versão final do jogo.....	49
Figura 23 – Avaliação da versão final do jogo.....	50

LISTA DE TABELAS

Tabela 1 – Análise comparativa entre trabalhos relacionados e este trabalho 21

SUMÁRIO

1	INTRODUÇÃO	12
2	OBJETIVOS	15
3	TRABALHOS RELACIONADOS	17
3.1	<i>gTest Learning: Um Jogo para Ensino Básico de Teste de Software</i>	17
3.2	<i>JitterTed's TDD</i>	19
3.3	ITEST LEARNING: UM JOGO PARA O APOIO AO ENSINO DE TESTES DE SOFTWARE	20
4	FUNDAMENTAÇÃO TEÓRICA	23
5	METODOLOGIA	31
6	O JOGO	34
7	AVALIAÇÃO DO JOGO	47
8	RESULTADOS.....	54
9	CONCLUSÃO	57
	REFERÊNCIAS.....	58
	APÊNDICE A – DESCRIÇÃO DO JOGO	60
	APÊNDICE B – QUESTIONÁRIO SOBRE A AVALIAÇÃO	64
	APÊNDICE C – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO	68
	APÊNDICE D – RESPOSTAS DO QUESTIONÁRIO	69

1 INTRODUÇÃO

O desenvolvimento de um software é uma tarefa complexa e a complexidade é proporcional ao tamanho e as características do software, logo o desenvolvimento está sujeito a problemas que podem terminar em um resultado não esperado. Para tentar minimizar esses riscos existem a atividade de testes. Testes são destinados a mostrar se o que o programa faz é de acordo com o que é proposto a fazer. Para então descobrir defeitos do programa antes do uso (SOMMERVILLE 2011).

Muitos elementos podem ser causadores de tais problemas, mas geralmente o erro humano é a maior causa mesmo com a utilização de métodos e ferramentas que estão disponíveis atualmente (DELAMARO; MALDONADO; JINO, 2013).

Delamaro, Maldonado, Jino (2013) afirmam que “Para que tais erros não perdurem, ou seja, para serem descobertos antes do software ser liberado para a utilização, existe uma série de atividades, coletivamente chamadas de Validação, Verificação e Teste, ou VV&T”.

Em qualidade de software, foram classificados diferentes tipos de testes que no final possuem a mesma finalidade de avaliar o grau de qualidade do software a ser testado, no entanto existem diferenças entre eles, pois o software possui diferentes camadas (níveis) que possuem comportamentos diferentes logo existem diversas formas de testar os diferentes tipos de teste existentes. Logo é muito importante ter o conhecimento sobre essa diversidade para que possamos mitigar cada vez mais os possíveis erros, e isso não é tarefa exclusiva do testador (citação).

Segundo (CLEGG; ROJAS; FRASER, 2017), no âmbito acadêmico o grande foco são os aspectos do projeto e codificação do software. A área de TI é extensa com diversos assuntos a serem tratados durante o período da graduação, logo é muito difícil que todas as áreas recebam a mesma atenção, pois o tempo é limitado.

Portanto algumas coisas serão priorizadas e outras serão deixadas em segundo plano, mas não são descartadas totalmente, o que é julgado com uma boa base para o conhecimento teórico dessas áreas deixadas em segundo plano é ensinado na maioria dos cursos (CLEGG; ROJAS; FRASER, 2017).

No entanto, os conteúdos deixados de lado podem ser um problema para o futuro, pois a falta de prática, motivação e o desenvolvimento inicial das habilidades

dessas disciplinas podem afetar a futura carreira profissional do aluno. E essa falta de conhecimento prático pode levar a um pequeno número de profissionais capacitados e disponíveis na área de Testes de Software (NASCIMENTO 2019). Em engenharia de software, na área de testes, existem diversos conceitos que para muitos alunos são vistos de maneira superficial, pois se trata de uma área que não recebe tanta ênfase e isso também é presente em equipes e projetos reais (NASCIMENTO 2019).

Os cenários reais que os profissionais da área de testes vivenciam não são fáceis de simular em um ambiente de ensino, e mesmo simulando os estudantes não conseguem distinguir como e quando devem aplicá-los na prática. A falta de conhecimento prático mesmo no final da formação acadêmica, pode ser considerada insuficiente para os empregadores. Uma possível causa é a diversidade de áreas existentes dentro desenvolvimento na prática aliada ao período do curso.

Logo, não é possível que todas as áreas recebam a mesma atenção, e a área de testes se inclui nessa categoria. Logo, é importante que os futuros profissionais tenham noção da importância dos testes durante o processo de desenvolvimento do software como um todo. Ter uma mínima base de conhecimento aliado a atividades práticas que buscam fixar o conhecimento sobre a importância da qualidade do software é primordial.

Dentro da área de testes existe o processo de BDD (sigla que traduzida para o português Desenvolvimento baseado em comportamentos), esse processo busca aumentar o entendimento geral da equipe a respeito do que o cliente espera para o sistema que o projeto está desenvolvendo.

Uma alternativa plausível para que os conceitos dos testes de software sejam mais disseminados é a criação de um jogo sério onde ele unirá a característica de atrair a atenção do jogador com a prática de um ambiente simulado de uma equipe de desenvolvimento que utiliza as técnicas de testes para avaliar o projeto em um cenário hipotético.

Atualmente existem dois tipos de jogos sérios que são os digitais e os analógicos, eles. Jogos sérios possuem o objetivo de transmitir um conteúdo educativo de maneira lúdica (YAÑEZ-GOMEZ; CASCADO-CABALLERO; SEVILLANNO, 2016) mas se diferem em relação à disponibilidade. Os digitais são disponibilizados por meio das plataformas digitais, já os analógicos estão disponíveis de maneira física por meio principalmente de tabuleiros e cartas

O jogo feito neste trabalho será no formato analógico para garantir as mesmas condições aos jogadores, uma interação presencial a não necessidade de manutenção no futuro e também para ter uma maior disponibilidade pois ele não será dependente das plataformas digitais como acontece com jogos digitais, um outro fator importante na escolha é a facilidade da aplicação deles em salas de aula e em grupos.

Diante do exposto acima o objetivo deste trabalho é a criação de um jogo sério analógico onde de maneira lúdica tentará melhorar o nível de conhecimento sobre o método BDD com foco no ensino deste processo.

O jogo será de tabuleiro com cartas, com turnos onde o jogador tentará passar por todas as casas para então concluir a execução do processo, para isso o jogador utilizará as cartas quando necessário.

A lógica principal do jogo é fazer com que o jogador escolha as cartas mais adequadas para passar para a casa seguinte da melhor forma, fazendo então com que de maneira prática o jogador aprenda a prática do BDD e entenda a importância de executar o processo seguindo todos os passos necessários.

2 OBJETIVOS

O objetivo deste trabalho é a criação de um jogo sério analógico que de maneira lúdica tentará melhorar o nível de conhecimento sobre o processo de BDD para alunos dos cursos de tecnologia da informação.

2.1 Objetivos específicos

- A) Propor o jogo sobre BDD, que envolve definir os elementos, regras e público-alvo.
- B) Desenvolver o jogo que auxiliará o público-alvo no entendimento sobre BDD.
- C) Melhorar o nível de conhecimento sobre BDD.
- D) Avaliar o jogo com o público-alvo.

3 TRABALHOS RELACIONADOS

Nesta seção, alguns trabalhos relacionados com o projeto proposto.

3.1 gTest Learning: Um Jogo para Ensino Básico de Teste de Software

Em (SILVA et al., 2016), foi desenvolvido um jogo de plataforma 2D *single-player* (jogo para somente um jogador) baseado no clássico Mario Word disponível para acesso através de *browser*, cuja finalidade é a execução de pequenos testes de fixação para o jogador. Estes desafios têm como objetivo forçar a busca por artefatos que venham a ajudar na resolução dos problemas, focando em testes de software com base na certificação *Foundation* do ISTQB (*International Software Testing Qualifications Board*).

A qualificação de nível *Foundation* é destinada a indivíduos envolvidos em testes de softwares como: analistas, consultores, engenheiros, consultores e ambos da área de teste de software. As perguntas e diálogos são feitas com interface de *RPG*, no final do turno se a quantidade de acertos for menor que a de erros o jogador perde o jogo, caso contrário, ele continua somando pontos até que chegue no final do estágio. Todas as questões foram obtidas de simulados do ISTQB, no final o jogador perdendo ou finalizando o estágio é exibido um ranking.

O trabalho cita alguns pré-requisitos para que o jogador se sinta confortável dentro do jogo, é necessário ter conhecimentos nos mínimos básicos nos seguintes tópicos:

1. Engenharia de Software: é preciso conhecer as fases do desenvolvimento do software e a importância de teste de software em cada fase na concepção de um software.

2. Teste de Software, ter uma base de conhecimento relativa aos conceitos de software. O conteúdo de questões está baseado na estrutura de capítulos do programa de certificação Syllabus na versão de 2011 [ISTQB. 2011].

3.2 JitterTed's TDD

Em (Young, 2021) foi desenvolvido o jogo JitterTeds TDD (sigla que traduzida para o português significa Desenvolvimento guiado por testes) foi um jogo analógico criado por Ted M Young, o jogo de tabuleiro foca no ciclo *Red-Green-Refactor* do TDD onde o jogador deve esperar a falha (*Red*), fazer o código para passar no teste (*Green*) e depois refatorar o código do teste. E esse é o ciclo principal do jogo, ele possui casas no formato de hexágono nas quais para sair delas o jogador necessita de cartas, estas cartas podem ser coletadas em diferentes *decks*, que são *deck* de ações que representam as decisões do jogador no processo de TDD. O jogo oferece ao jogador a opção de pular as casas, mas o jogador deve assumir o aumento do risco, esse aumento pode prejudicar ou não o jogador na hora de validar o resultado.

A figura acima mostra o jogo e como deve posicionar os seus elementos, as casas são agrupadas de forma decrescente de cima para baixo formando o tabuleiro. Do lado direito ficam os *decks* organizados em *deck* de testes e *deck* de ações (ações do jogador de acordo com o processo de TDD).

Do lado inferior esquerdo mostra dois elementos de marcação, o elemento de *Commits* representa a pontuação do jogador, toda vez que ele chega na casa final (*Commit code*) e passa após jogar o dado o jogador pontua (marca um *commit*).

O outro elemento do lado esquerdo inferior serve para marcar o risco, ele aumenta cada vez que o jogador pula alguma casa, o *Code Risk* é incrementado, quando o jogador reprova a última casa e não consegue realizar o *commit*.

O jogo proposto no trabalho também aborda o TDD mas apenas como um elemento do todo, diferentemente do jogo de Young, que aborda apenas o TDD. Esse jogo serviu como principal inspiração para este trabalho, o que ajudou na concepção da mecânica e a criação dos elementos que o compõem.

3.3 ITEST LEARNING: UM JOGO PARA O APOIO AO ENSINO DE TESTES DE SOFTWARE

Em (SOUSA, 2012), foi desenvolvido o *iTest Learning*, um jogo *web* que visa apoiar o ensino de Testes de Software. No jogo, o aluno segue um tutorial online que lhe ajuda a entender os conceitos do planejamento de testes de software, a partir da prática com simulação de problemas para possíveis sistemas. O objetivo do jogo é fazer com que o jogador realize um planejamento de teste de software a partir da especificação de projeto.

Ao final do jogo, uma tela com um *feedback* contendo todo o planejamento feito pelo jogador é exibida. Há também um planejamento que seria recomendado para o projeto escolhido. O *feedback* contém uma breve descrição do porquê de cada escolha, a forma de fazer o jogador refletir e também discutir as respostas com outros jogadores. Assim, o jogador poderá fazer uma comparação do seu planejamento com o planejamento recomendado.

A avaliação do jogo foi realizada com 7 alunos da turma de Verificação e Validação do quinto semestre do curso de Engenharia de Software da Universidade Federal do Ceará, Campus Quixadá. Com base nesses resultados, é possível verificar que o *iTest Learning* agradou aos participantes e sua metodologia de ensino é adequada. Ele ainda provê um aprendizado em relação ao conteúdo proposto, faltando apenas direcioná-lo um pouco mais de maneira a estar mais focado no planejamento de testes de software.

Tabela 1 – Análise comparativa entre trabalhos relacionados e este trabalho

Trabalho	Plataforma	Gênero do Jogo	Tipo de interação com o Jogador
(Silva et al., 2016)	Multiplataforma	Plataforma 2D	Intuitiva
(Young, 2021)	Analógica	Tabuleiro	Por turnos
(Sousa, 2012)	Web	Estratégia	Perguntas
Este trabalho	Analógica	<i>Card Game</i> /Tabuleiro	Por turnos

Fonte: elaborado pelo autor

4 FUNDAMENTAÇÃO TEÓRICA

Nesta seção, serão apresentados alguns dos conceitos principais necessários para o entendimento e desenvolvimento do projeto proposto neste trabalho.

4.1 Jogos Sérios

Jogos sérios têm como principal objetivo fornecer uma experiência diferenciada na educação, são jogos cujo propósito não se limita a prover entretenimento, mas também ensino e treinamento (OLIVEIRA et al., 2015), sendo assim uma alternativa à forma de ensino tradicional.

Os jogos sérios possuem elementos dos jogos comuns então eles possuem fases com objetivos e os resultados das tentativas do jogador, mas devem sempre oferecer algum tipo de aprendizagem ao jogador. São voltados à educação, pois ajudam na melhora da comunicação e assimilação de conceitos antes não tão claros ao jogador. Como consequência disso, os jogos sérios podem trazer melhoras ao jogador em futuras tomadas de decisões. Eles para a prática dos conhecimentos técnicos e não-técnicos importantes ao desenvolvimento profissional. Porém, incluem elementos lúdicos e de entretenimento para aumentar a atenção e incentivar o envolvimento do utilizador na aprendizagem. Podendo ser utilizada nas mais diversas áreas de estudo (OLIVEIRA et al., 2015).

Eles diferem de outros meios educativos, como filmes ou livros porque oferecem uma experiência em que o jogador está ativamente envolvido em fazer decisões. O jogador está imerso na experiência do jogo e recebe a resposta sobre as suas decisões.

4.1.1 Jogos sérios analógicos.

Os jogos sérios analógicos se diferem dos digitais em relação ao acesso. Existem diferenças entre eles em relação a disponibilidade e manutenção. Enquanto os jogos digitais são exclusivos das mídias digitais, os analógicos não são dependentes dessas mídias (Medeiros, 2019). A manutenção é algo importante também, jogos analógicos não necessitam de manutenção uma vez que a sua concepção e produção for concluída, logo não é necessário gastar recursos com futuras manutenções.

Os jogos sérios analógicos são mais característicos como tabuleiros e cartas, a forma de como os elementos deste ambiente se relacionam é de acordo com contexto que se é planejado para este tipo de jogo, mas, em geral, envolve turnos com pontuações durante ou no final desses turnos.

Por fim os jogos analógicos oferecem uma proximidade com relação a interação entre os jogadores, por ser uma experiência presencial ele permite que os jogadores troquem experiências de uma forma mais natural e eficaz (Medeiros, 2019).

4.2 Testes de Software

Os testes de software representam uma importante etapa no processo de implementação do software, é a maneira de validar se a aplicação está de acordo com os requisitos funcionais e não funcionais propostos (MALDONADO et al., 2016). Existem várias técnicas que podem ser usadas em diferentes momentos dentro do período de desenvolvimento do software. Os testes necessitam de um processo bem definido que inclui modelagem e caso de testes e também o gerenciamento deles.

O processo de testes tem dois objetivos. O primeiro é garantir e mostrar ao cliente que o sistema atende aos requisitos propostos. Para softwares mais específicos testar cada requisito que foi documentado para o projeto. E para softwares menos específicos testar todas as características do sistema e as suas respectivas combinações que serão feitas na *release* do produto (SOMMERVILLE 2011). O

segundo é descobrir cenários nos quais o software não se comporta da maneira esperada. O objetivo dos testes é eliminar comportamentos indesejáveis do sistema (SOMMERVILLE 2011).

Dentre os tipos de testes existentes temos os testes unitários que validam pequenas partes do software com base em suas entradas possíveis e saídas esperadas, onde normalmente funções do código que recebem argumentos e retornam um determinado valor ou efetuam alguma ação onde resultado pode ser analisado.

Existem testes que visam analisar seu comportamento externo do sistema que são os testes funcionais, eles consistem em uma série de técnicas, com o objetivo de verificar se a aplicação é capaz de desempenhar as funções que se propõe a fazer (MALDONADO et al., 2016). As técnicas mais comuns de testes funcionais são denominadas caixa-branca e caixa-preta. O teste de caixa-branca tem como objetivo a análise do comportamento interno do software, ou seja, o seu código-fonte. Já o teste de caixa-preta é feito em cima das funções que devem ser desempenhadas pelo programa. Normalmente, esse tipo de teste é baseado nos requisitos do sistema e realizam ações de uso cotidiano no sistema, entrando com dados e avaliando sua resposta.

Também existem os testes de integração que tem de analisar o comportamento do software quando interage com outros processos ou aplicações. A ideia do teste é compreender se o processo de interação em questão gera alguma falha ou instabilidade. Eles são realizados após outras etapas de teste que garantem o funcionamento individual das partes.

Os testes de aceitação formam uma das últimas etapas do desenvolvimento do sistema, pois validam se o software está pronto para ir para produção. Os testes de performance, também chamados testes de carga, têm por objetivo submeter o software a situações intensas de uso e avaliar seu comportamento.

A partir dos diagnósticos, a equipe é capaz de compreender os limites do programa sob diversas condições. Para constatar a qualidade da aplicação, ela é submetida a avaliações que simulam eventos e situações previsíveis de acordo com a rotina do cliente, ou seja, testes de carga, estresse e estabilidade. Quaisquer falhas detectadas durante o teste devem ser corrigidas pela equipe. Os testes de segurança visam garantir a correta aplicação das premissas de segurança definidas para o *software*, alcançando assim um ambiente operacional mais seguro. Dessa maneira, o

produto final só é entregue ao cliente quando os requisitos de segurança são devidamente atendidos.

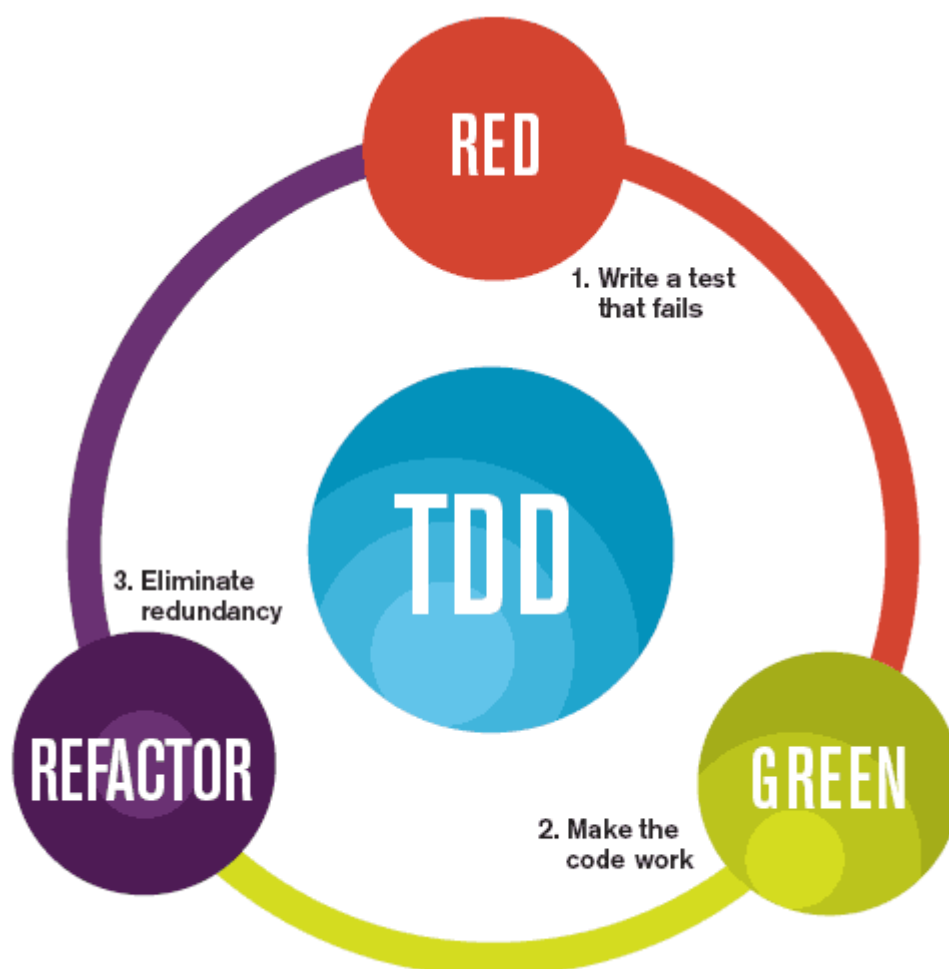
Para o trabalho foram abordados os seguintes temas sobre testes

- TDD

O TDD (Desenvolvimento orientado a testes) é um processo no qual os testes são escritos antes do desenvolvimento do código (Feitosa, 2007). O processo se baseia em criar o teste, ele falha pois o código não foi implementado, então o que se deve fazer a seguir é implementar o código para que o teste enfim passe, logo após a criação do código e refatorado para melhor a sua qualidade (FRASER, S et al. 2003).

O TDD foi criado por Kent Beck, sua premissa para esse processo é de que o código deve ser testado e refatorado de forma contínua em um processo chamado Red-Green-Refactor.

Figura 1 – Ciclo do TDD por Kent Beck



The mantra of Test-Driven Development (TDD) is "red, green, refactor."

Fonte: <https://www.digite.com/pt-br/agile/desenvolvimento-orientado-a-testes-td/>

O ciclo do TDD consiste em três passos:

1. A fase vermelha, onde é escrito o teste que falha.
2. A fase verde, onde o código necessário para que o teste passe é implementado.
3. E por fim a fase de refatoração, onde são eliminadas todas as redundâncias presentes no código.

- Testes Unitários

Testes unitários estão mais próximos ao código fonte, logo é recomendado que o responsável pelo teste seja alguém com conhecimento sobre o código, um bom entendimento sobre ele é importantíssimo para a acurácia dos resultados dos testes.

Para Sommerville(2011) o processo de testes unitários focam nos métodos e classes do sistema, as funções individuais e métodos do objeto são os tipos de componentes mais simples.

Para ele quando para testar as classes de objeto se deve projetar os testes para fornecer uma cobertura de todas as características do objeto. Ele cita três passos necessários para isso:

- Testar todas as operações associadas ao objeto.
- Definir e verificar o valor de todos os atributos associados ao objeto
- Colocar o objeto em todos os estados possíveis, o que significa que tem que simular todos os eventos que causam mudanças de estado.

- Teste de aceitação

Testes de aceitação são executados no estágio final do processo de testes, eles buscam representar cenários que o cliente passará durante o uso do sistema. É importante usar dados reais durante esses testes e não dados simulados (Sommerville 2011). Ele pode revelar falhas na hora da elaboração dos requisitos do sistema, pois os dados usados durante a execução não são simulados e são dados

reais fazendo com que o sistema execute de uma maneira diferente do que era executado com os dados de testes.

Os testes de aceitação também podem revelar problemas em que os recursos do sistema não atendam a demanda do cliente e que o desempenho do sistema não seja aceitável (Sommerville 2011).

Sommerville(2011) definiu seis estágios no processo de testes de aceitação foram descritos da seguinte maneira.

- Definir critérios de aceitação.

É ideal que ocorra no início do processo, os critérios de aceitação devem fazer parte do contrato do sistema e serem acordados entre o desenvolvedor e o cliente. Na prática pode ser difícil definir critérios no início do processo, por falta de detalhamento e pode haver mudanças nos requisitos durante o processo de desenvolvimento.

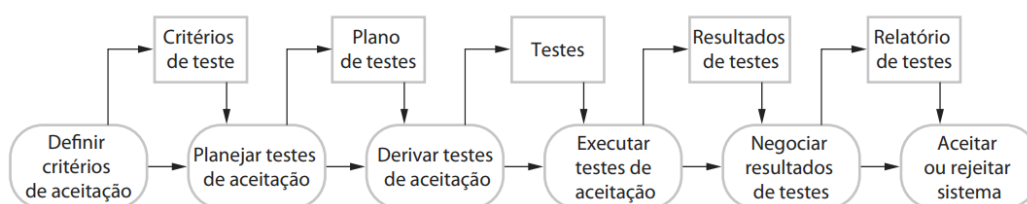
- Planejar testes de aceitação

Trata-se de decidir os recursos, tempo e orçamento necessários para os testes de aceitação e estabelecer um cronograma de testes. O plano de testes deve discutir a cobertura dos requisitos e a ordem em que as características do sistema são testadas. Deve definir os riscos para o processo de testes, como interrupção do sistema e desempenho inadequado e discutir como os riscos podem ser mitigados.

- Derivar testes de aceitação.

Uma vez que os testes de aceitação tenham sido estabelecidos, eles precisam ser projetados para verificar se existe ou não um sistema aceitável. Testes de aceitação devem ter como objetivo testar tanto as características funcionais quanto as não funcionais do sistema. Eles devem, idealmente, fornecer cobertura completa dos requisitos de sistema. Na prática é difícil estabelecer critérios completamente objetivos de aceitação. Muitas vezes fica margem para discussão sobre se o teste mostra ou não se um critério foi definitivamente cumprido.

Figura 2 – Esquema do ciclo de testes de Sommerville



Fonte: adaptado de SOMMERVILLE p.160

- Executar testes de aceitação.

Os testes de aceitação acordados são executados no sistema. Idealmente, isso deve ocorrer no ambiente real em que o sistema será usado, mas isso pode ser interrompido e impraticável. Portanto, um ambiente real de testes de usuário pode ter de ser configurado para executar esses testes. É difícil automatizar esse processo, pois parte dos testes de aceitação pode envolver testes de interações entre usuários finais e o sistema. Pode ser necessário algum treinamento para os usuários finais.

- Negociar testes de aceitação.

É muito improvável que todos os testes de aceitação definidos passem e que não ocorra nenhum problema com o sistema. Se esse for o caso, então o teste de aceitação está completo, e o sistema pode ser entregue. O mais comum, contudo, é que alguns problemas sejam descobertos. Nesses casos, o desenvolvedor precisa negociar para decidir se o sistema é bom o suficiente para ser colocado em uso. Eles também devem concordar sobre a resposta do desenvolvedor para os problemas identificados.

- Rejeitar/aceitar sistema.

Este estágio envolve uma reunião entre os desenvolvedores e o cliente para decidir se o sistema deve ser aceito. Se o sistema não for bom suficiente para o uso, então será necessário um maior desenvolvimento para corrigir os problemas identificados. Depois de concluída, a fase de testes de aceitação é repetida.

4.3 BDD

Antes de falar sobre BDD (Desenvolvimento Orientado ao Comportamento) é necessário o seu contexto histórico, tradicionalmente no modelo de cascata clássico os casos de testes (testes unitários) eram feitos após a fase de desenvolvimento. Somente após a conclusão da implementação do software para verificar a validade do código (Soeken, Wille, Drechsler 2012). Com o surgimento do movimento ágil, esse cenário mudou, os testes passaram a ser incorporados desde o início do processo de desenvolvimento. Nesse período foi criado o TDD *Test Driven Development* (traduzindo para o português, desenvolvimento guiado por testes). O

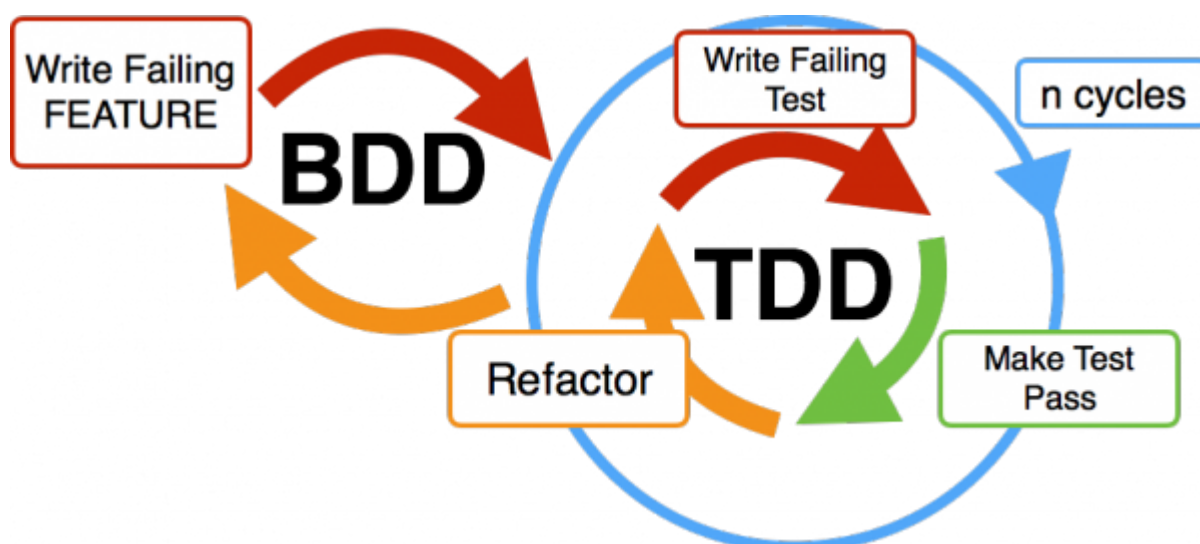
TDD usa como ponto de partida os testes de aceitação para dar início ao processo de desenvolvimento (Soeken, Wille, Drechsler 2012).

Como falado na seção 4.2, testes de aceitação representam aquilo que o cliente espera para o sistema. O software proposto é considerado aceitável no momento que todos os testes de aceitação são validados, logo ele é classificado como completo. No entanto, teste unitário e teste de aceitação se diferem, teste unitário verificar a corretude de uma única unidade (parte) de código enquanto o teste de aceitação verifica um cenário para o sistema sem considerar o código a ser implementado futuramente (Soeken, Wille, Drechsler 2012).

Geralmente os testes unitários são feitos pelos desenvolvedores. Já os testes de aceitação são elaborados pelos *stakeholders* e alinhados com os desenvolvedores permitindo um entendimento geral sobre os objetivos que o software deve atingir.

O BDD foi proposto para tentar resolver os problemas que surgiram com o TDD, onde o maior desafio encontrado pelos desenvolvedores era definir um bom ponto para começar o desenvolvimento e sobre o que se deve testar ou não. O BDD usa linguagem natural para assegurar o entendimento geral entre os membros da equipe para garantir que seja entregue aquilo que o cliente espera.

Figura 3 – Ciclo BDD



Fonte: <https://saucelabs.com/blog/a-two-minute-bdd-overview>

Atualmente o ciclo do BDD é complementado pelo TDD, onde o teste de funcionalidade é escrito, o teste falha pois não foi implementado, o que se espera dessa funcionalidade é mostrado através da estrutura (Dado, Quando e Então) onde

de maneira mais formal busca mostrar o que se espera dela. **Dado** que o sistema tenha determinada funcionalidade ou recurso, **quando** o usuário faz alguma ação com essa funcionalidade ou recurso, **então** o sistema deve fornecer alguma resposta a essa ação, o código é desenvolvido com a ajuda do TDD onde o é baseado no que se espera da funcionalidade que foi definido no BDD.

O próximo capítulo mostra a metodologia seguida para a concepção do jogo utilizando os conceitos mostrados neste capítulo.

5 METODOLOGIA

Neste capítulo serão apresentadas as etapas para a execução deste trabalho.

O primeiro passo é definir o escopo do jogo, a segunda etapa consiste em selecionar todo o conteúdo sobre o tema que será abordado no jogo. A produção do jogo será realizada após a seleção de conteúdo. E na última etapa foi realizada a experimentação do jogo com o público-alvo onde será coletado dados que serão retirados do questionário e entrevista para a análise dos resultados.

5.1 Definir escopo do jogo

Nesta fase foram definidos os aspectos que se encaixam no escopo do jogo. O escopo foi definido para atender o contexto do jogo buscando assim adaptar ao processo de BDD.

Gênero do Jogo - A característica geral do jogo, relacionado ao seu tipo, forma de jogar, regras e mudanças de estado. O jogo proposto neste trabalho segue uma mistura dos gêneros tabuleiro e cartas.

Público-alvo – O público destinado ao jogo proposto. O público-alvo definido foi o grupo de alunos dos cursos de tecnologia da informação.

Plataforma – A forma pelo qual o jogador tem acesso ao jogo. A plataforma definida para o jogo é analógica.

Mecânica do jogo – Essa característica define as maneiras de mudança de estado do jogador. A mecânica do jogo foi a mecânica de escolhas onde o jogo mostra as consequências para as escolhas do jogador.

Elementos – Tudo o que compõe o jogo em si. Os elementos definidos para o jogo foram, cartas, casas que compõem o tabuleiro, dado e peões que representam os jogadores no tabuleiro.

Ações – Basicamente são todas as interações que o jogador pode fazer durante a jogatina. Para o jogo foram definidas as ações de ir de uma casa para a outra, pular casa, ou seja, quebrar a sequência de casas, pegar cartas do *deck*, usar as cartas no contexto da necessidade que uma casa impôs para o jogador usar e jogar o dado.

5.2 Definir conteúdos sobre o tema a ser abordado no jogo

Nesta etapa, foi feita a seleção dos conteúdos teóricos sobre o tema, foram escolhidos aqueles que se adequem melhor ao contexto do assunto buscado. Os conceitos definidos para o jogo deste trabalho estão destacados em negrito.

Para o jogo foi definido o escopo de conteúdos que envolvem o ciclo do **BDD** durante o desenvolvimento do projeto que são os seguintes.

User Story que nada mais é a **História do Usuário** onde um recurso de software é explicado de maneira informal para um maior entendimento geral.

Critérios de aceitação para o **User Story**, eles definem aquilo que o cliente espera para que o sistema seja aceito. Para ajudar na montagem dos **critérios de aceitação** o jogo aborda a associação dos

Stakeholders com o sucesso de aceitação do cliente, quanto mais tipos diferentes de **stakeholders** presentes nessa fase ajuda a diminuir o risco de fracasso na hora da validação do cliente, logo em uma equipe com mais **stakeholders** interessados no sucesso do projeto na fase ele elaborar esses critérios logo a comunicação e a troca de informações entre membros de diferentes níveis de conhecimento ajuda em um molde mais próximo do que o cliente espera pois essas diferentes formas de interpretar e pensar aliadas às diferentes níveis de experiência concentradas em um esforço conjunto chegam em um consenso mais próximo do objetivo que o cliente espera.

Testes de aceitação que como já foi falado na seção 4.2, verificam se o sistema está de acordo com o que o cliente espera.

TDD que é executado em conjunto com o **BDD**, o jogo busca associar o uso do **TDD** na fase de codificação dentro desta fase também são abordados os

Testes unitários que são executados dentro do ciclo do **TDD**.

Refatoração é uma atividade de revisão de código, nela ocorre a eliminação de renúncias e possíveis erros visando melhor a qualidade do código.

5.3 Produzir o Jogo

Com a definição do escopo e a seleção de conteúdo finalizada, o próximo passo foi a produção do jogo. O objetivo desta etapa é o desenvolvimento da solução proposta seguindo o escopo do projeto. O jogo vai ser apresentado com mais detalhes no capítulo 6.

5.4 Avaliar o jogo com o público-alvo.

Nesta etapa, o jogo foi testado pelo público-alvo, o teste foi feito com os alunos dos cursos de tecnologia da informação, foram selecionados alunos com conhecimento prévio sobre o tema (BDD), tanto de maneira teórica como prática. Após a experimentação um questionário foi aplicado. O método de questionário foi MEGGA+ pelo método qualitativo.

O principal objetivo dessa etapa é coletar o *feedback* dos usuários, apresentar o resultado da análise dos *feedbacks* e fazer conclusões em cima dos dados e analisá-los na próxima etapa.

Essa fase consistiu em realizar teste no protótipo, logo após foi realizado o teste piloto, após o teste foram feitos alguns ajustes no jogo. Então o teste final com o público-alvo foi feito.

Mais detalhes dos testes foram descritos no capítulo 7.

Análise do questionário.

Após a coleta das respostas do questionário, uma análise das respostas foi realizada para avaliar se o jogo atendeu a proposta do trabalho e se ele representa o tema como planejado. Também foi analisado o *feedback* dos usuários sobre os pontos positivos e negativos para futuras melhorias.

6 O JOGO

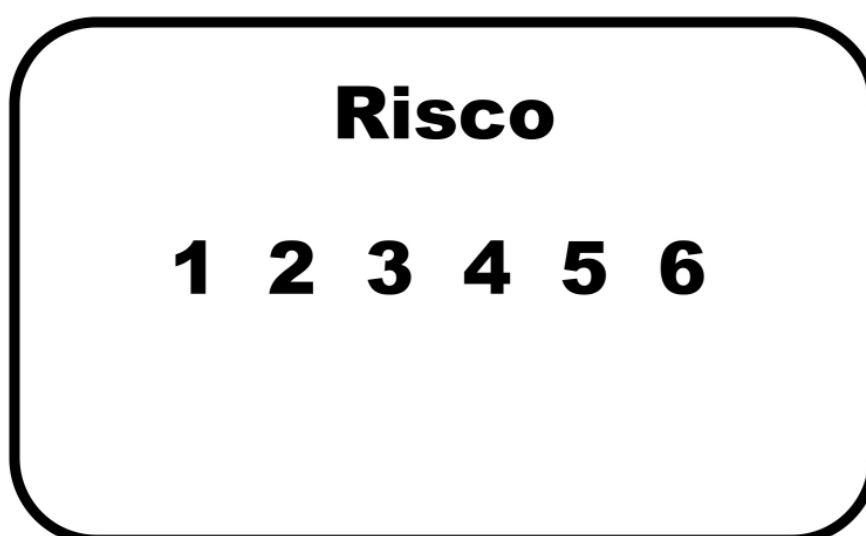
Neste capítulo serão apresentados os elementos presentes no jogo.

O fluxo principal do jogo é passar por todo o processo do BDD, para ter sucesso na execução do processo, o jogador precisa passar por cada casa até chegar na parte final do jogo para então validar os resultados com o PO (*Product Owner*) e o Cliente. Para chegar nas fases finais com boas chances de aprovação, o jogador deve seguir à risca o que se pede em cada casa, caso contrário se o jogador decide pular fases logo processos necessários para a aprovação, o seu risco de reprovação aumenta.

Para auxiliar o jogador na progressão, o jogo oferece cartas (recursos) que o jogador pode usar caso seja necessário e permitido pelo jogo. Essas cartas representam elementos reais de um projeto real e que foram abstraídos para serem usados de exemplos para o ensino.

O jogo pode ser jogado com uma a quatro pessoas, ganha quem faz mais pontos dentro do tempo limite estipulado, recomenda-se que os tempos sejam de 15, 30 e 45 minutos, e ou quem marca a quantidade de pontos estipulada de 5, 8 e 10 pontos. O jogo começa com cada jogador pegando 4 cartas do *deck* de recursos e cada jogador recebe um marcador de risco (Figura 10).

Figura 4 – Marcador de risco



Fonte: elaborado pelo autor

O jogador usara essas cartas quando for necessário, caso ele queira pegar mais cartas ele deve descartar uma carta da mão e pegar outra do *deck*, fazendo

assim uma troca de recursos. Logo o jogador pode ficar com no máximo 4 cartas do deck de recursos na mão. O primeiro jogador joga o seu turno passando de uma casa para outra usando os recursos que ele tem ou pulando para outra casa (aumentando o risco), ele só pode avançar uma casa por turno.

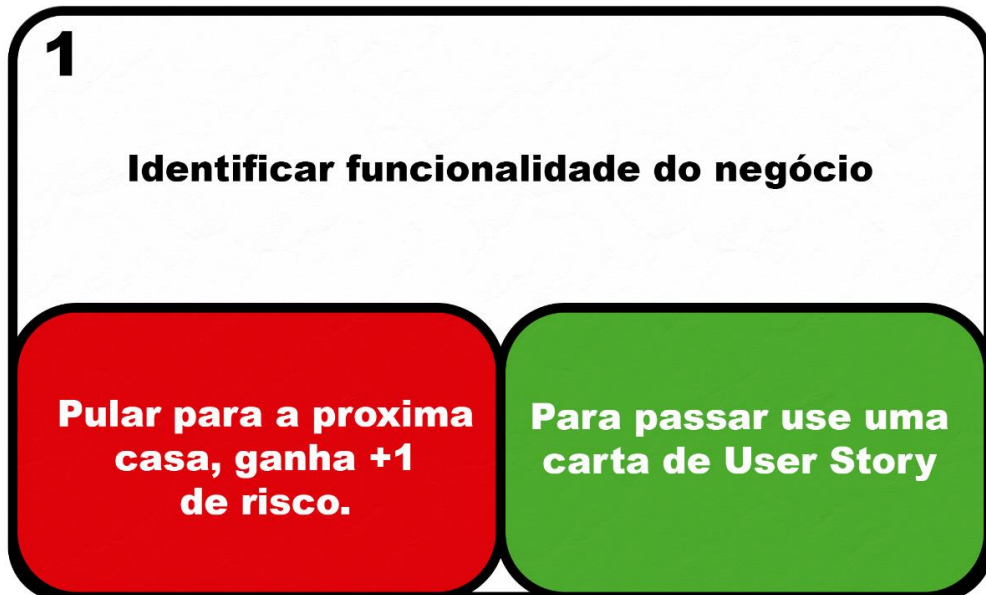
Ao fim do turno do primeiro jogador, é a vez do segundo, ele realiza o mesmo processo que o primeiro jogador fez, o mesmo acontece com os próximos jogadores. Para pontuar o jogador precisa passar por todas as casas do tabuleiro. Assim executado o processo do jogo.

O *deck* de recursos agrupa as cartas de *user story*, código escrito, teste e refatoração. Jogadores iniciantes no assunto podem jogar o jogo, mas a experiência adquirida anteriormente pode ajudar no entendimento das regras e mecânicas do jogo logo de início. Não é um jogo exclusivo para pessoas que já tem conhecimento sobre BDD e TDD.

A seguir serão descritos os elementos presentes ao mesmo tempo que será mostrado o passo a passo do fluxo principal do jogo.

6.1 Elementos e ciclo do jogo

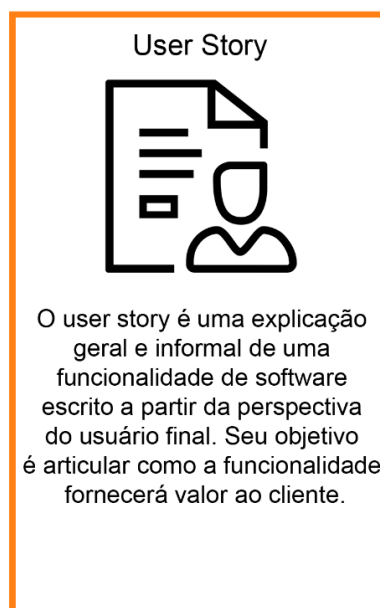
Figura 5 – Primeira casa do tabuleiro



Fonte: elaborado pelo autor

Nesta fase o jogador deve usar a carta de *User story* para avançar, a carta ajuda a identificar as funcionalidades que são esperadas pelo cliente para o sistema.

Figura 6 – Carta de User Story



Fonte: elaborado pelo autor

Figura 7 – Segunda casa do tabuleiro

2 **Definir critérios de aceitação do User Story**

**Agora é hora de montar sua equipe!!!
Você vai elencar os interessados no sucesso do projeto**

Para passar você necessita usar no mínimo uma carta de stakeholder para montar a equipe.

Fonte: elaborado pelo autor

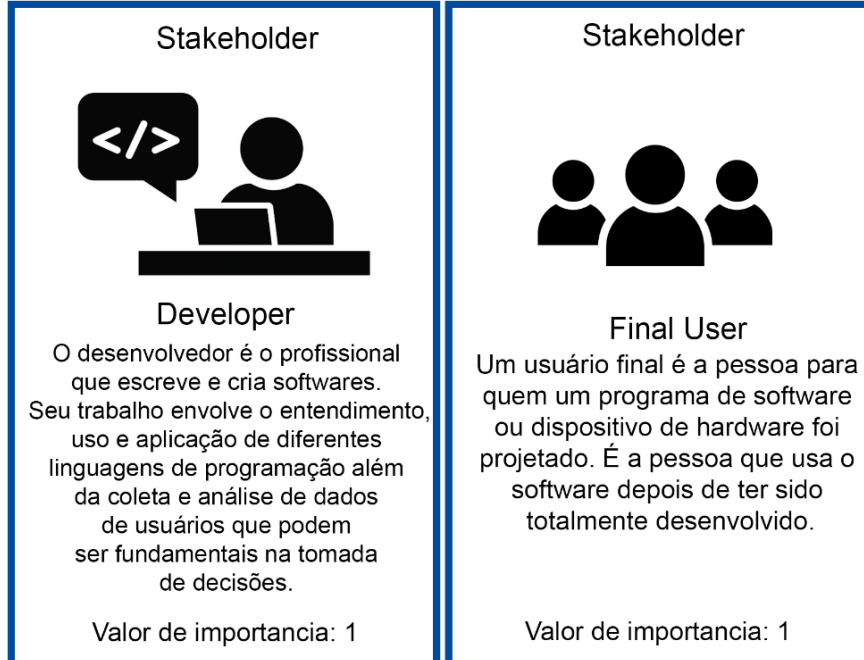
Nesta fase o jogador precisa montar a equipe, ele puxa 3 cartas do *deck* de *stakeholders*, o jogador deve montar a equipe utilizando as cartas, ele pode jogar no máximo 4 cartas na mesa sendo as 4 de diferentes tipos de *stakeholders*, ele tem o limite de 3 cartas de *stakeholders* na mão e 4 na mesa, essas cartas possuem um valor de importância, a soma desses valores é usada para validar o resultado, o jogo busca de maneira geral passar pagar o jogador a mensagem de que quanto mais interessados(*stakeholders*) no projeto maiores são as chances de aprovação do cliente no final do ciclo para montar a equipe, de maneira abstraída essa equipe vai ajudar na definição dos critérios de aceitação dos *user story*, é recomendado que o jogador elenque a maior quantidade de interessados possíveis, o jogo permite que o jogador use apenas uma carta de cada tipo de *stakeholder* na equipe, o jogador deve usar no mínimo uma carta de *stakeholder* podendo ir até 4 cartas.

Figura 8 – Cartas de stakeholders



Fonte: elaborado pelo autor

Figura 9 – Cartas de stakeholders



Fonte: elaborado pelo autor

Figura 10 – Terceira casa do tabuleiro

3

Elaborar teste de aceitação

Aqui você vai descrever o cenário do teste utilizando as cartas de teste de aceitação

Se não possuir as 3 cartas é a vez do proximo jogador

Para passar você necessita das cartas: Dado, Quando e Então

Fonte: elaborado pelo autor

Nesta fase o jogador deve elaborar o teste de aceitação, para passar ele tem que montar a sequência correta das cartas de teste de aceitação. Ele deve pegar 3 cartas do *deck* de cartas dos testes de aceitação, caso ele não consiga montar a sequência pegando cartas repetidas ele deve esperar a sua próxima vez para então pegar mais 3 cartas e pegar as que faltam, antes de pegar mais 3 cartas o jogador deve descartar as cartas repetidas de volta ao final do *deck*.

Figura 11 – Cartas que compõem os critérios de aceitação

Teste de aceitação	Teste de aceitação	Teste de aceitação
Dado	Quando	Então
Dica: Consiga as 3 cartas deste tipo na sequencia correta para formar o teste de aceitação.	Dica: Consiga as 3 cartas deste tipo na sequencia correta para formar o teste de aceitação.	Dica: Consiga as 3 cartas deste tipo na sequencia correta para formar o teste de aceitação.
1	2	3

Fonte: elaborado pelo autor

Após o jogador identificar as funcionalidades do negócio, definir os critérios com a equipe e montar a estrutura do teste de aceitação da *feature*, nesta casa ele deve testar o cenário pela primeira vez. Como o jogador ainda não pegou a carta de código escrito logo o teste tem que falhar, neste momento o jogo pede para que o jogador pegue uma carta de testes

Figura 12 – Quarta casa do tabuleiro

4 **Testar a feature de cenário**

Agora você vai testar o cenário que foi elaborado na etapa anterior.

Pegue uma carta de testes!!!

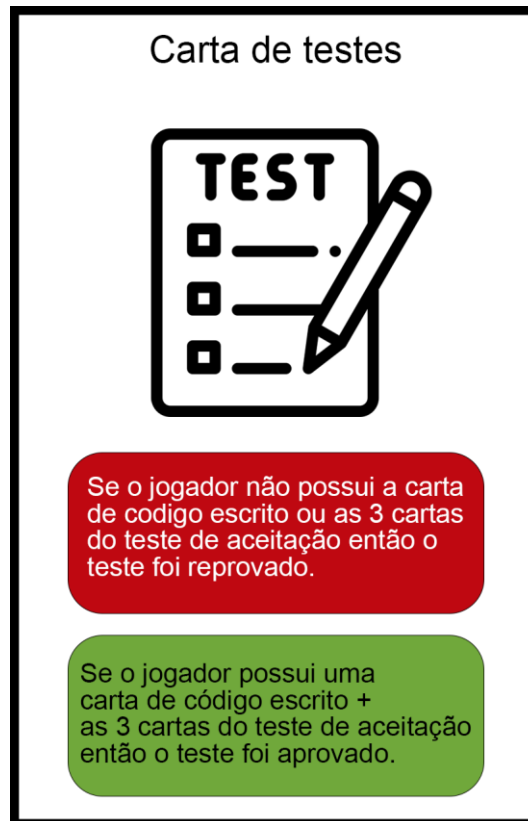
Você não pode usar uma carta de código (TDD) neste momento!!! Pule e ganhe +1 de risco

Para passar o teste tem que falhar.

Fonte: elaborado pelo autor

Esta carta permite que o jogador teste os elementos, neste primeiro momento o teste será reprovado pois o jogador não está com a carta de código na mesa.

Figura 13 – Carta de testes



Fonte: elaborado pelo autor

Figura 14 – Quinta casa do tabuleiro

5 **Escrever código do teste**

Agora você necessita do código escrito para o teste passar.

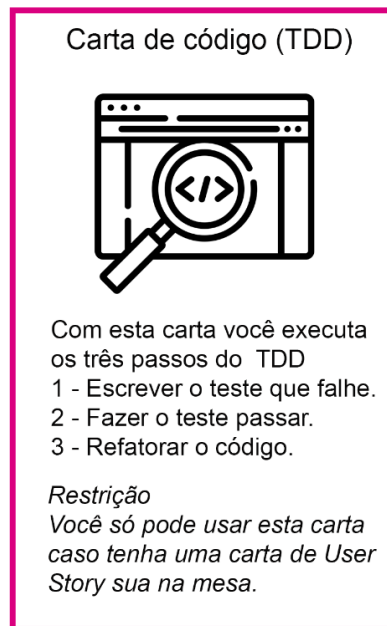
Pule e ganhe +1 de risco

Para passar você necessita da carta de código escrito

Fonte: elaborado pelo autor

Agora o jogador tem que fazer o teste passar, ele deve usar uma carta de código escrito para passar desta casa, de maneira abstraída o jogo usa a carta de código escrito para representar a implementação do teste de cenário que foi montado na casa 3.

Figura 15 – Carta de código



Fonte: elaborado pelo autor

A carta mostra o modelo de TDD usado para implementar o código de teste.

Figura 16 – Sexta casa do tabuleiro

6 Executar teste novamente

Agora você tem o do teste escrito puxe uma carta de testes para validar o código.

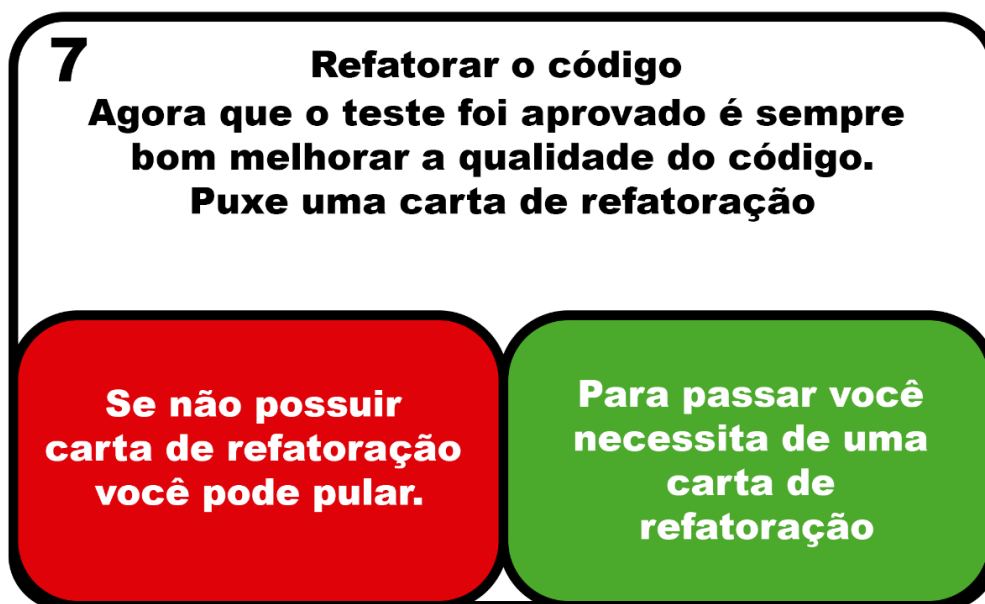
Para passar o teste tem que ser aprovado. Pule e ganhe +1 de risco

Para passar o teste tem que ser aprovado

Fonte: elaborado pelo autor

Nesta casa o jogador deve realizar o teste novamente, agora o jogo espera que o teste passe já que o jogador realizou os passos anteriores. A carta usada é a mesma da figura 13.

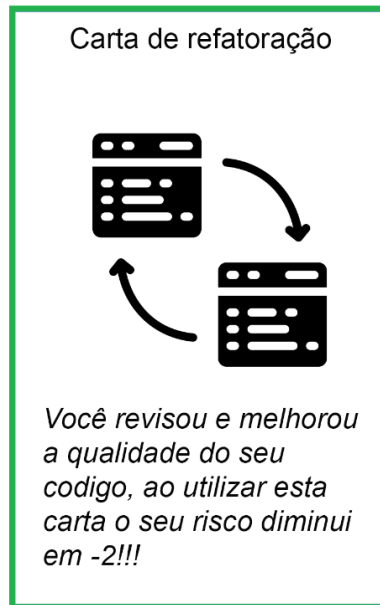
Figura 17 – Sétima casa do tabuleiro



Fonte: elaborado pelo autor

O jogo oferece ao jogador a opção de refatorar o código, caso ele tenha acrescentado o risco o jogador pode usar uma carta de refatoração, a carta de refatoração representa uma revisão a mais no código, durante a refatoração além de melhorar a qualidade do código em termos de legibilidade e redundâncias ele também pode ajudar a encontrar erros que não foram encontrados na fase de implementação do código. Logo diminuindo as chances de reprovação.

Figura 18 – Carta de refatoração



Fonte: elaborado pelo autor

Figura 19 – Oitava casa do tabuleiro

8 Executar teste do código refatorado

Agora você refatorou o código puxe uma carta de testes para validar o código refatorado.

Pule e ganhe +1 de risco

Para passar o teste tem que ser aprovado

The image shows a board game instruction card for the eighth space. It is titled '8 Executar teste do código refatorado' (8 Execute test of refactored code). The main instruction reads: 'Agora você refatorou o código puxe uma carta de testes para validar o código refatorado.' (Now you have refactored the code, draw a test card to validate the refactored code.). Below this, there are two colored boxes: a red one on the left that says 'Pule e ganhe +1 de risco' (Skip and gain +1 risk) and a green one on the right that says 'Para passar o teste tem que ser aprovado' (To pass the test you must be approved).

Fonte: elaborado pelo autor

Após a refatoração o jogador tem que testar o código refatorado para garantir que tudo está ok.

Nas duas fases finais o jogador tem que validar os resultados. Primeiramente ele deve validar com o PO (*Product Owner*). A primeira validação é feita com base no risco do jogador que está marcado na tabela da figura. Neste momento o jogador lança um dado de 6 lados e o resultado é validado de acordo com as condições que estão descritas na casa que está apresentada na figura a seguir.

Figura 20 – Nona casa do tabuleiro

9 Validar resultados com o PO.

Se o valor do risco < valor do dado logo foi validado pelo PO.
Se o valor do risco >= valor do dado logo foi reprovado pelo PO.

Se o resultado foi reprovado volte ao início e devolva as cartas para os respectivos decks.

Para passar o resultado deve ser aprovado pelo PO.

Fonte: elaborado pelo autor

Após a aprovação do PO, o jogador deve validar o resultado com o cliente. Agora ele deve somar o valor da equipe, o jogo tenta mostrar ao jogador que quanto maior o número somado da equipe, ou seja, quanto mais pessoas engajadas para o entendimento geral sobre o que o cliente espera para o projeto maiores são as chances de aprovação. Ele lança o dado novamente e é aprovado de acordo com as condições que estão apresentadas na figura a seguir.

Figura 21 – Decima casa do tabuleiro

10 Validar resultados com o cliente

Se o valor somado dos stakeholders \geq valor do dado logo o resultado foi aprovado
Se o valor somado dos stakeholders $<$ valor do dado logo o resultado foi reprovado.

Se o resultado foi reprovado volte ao inicio e devolva as cartas para os respectivos decks.

Se o resultado for aprovado você pontua!!!

Fonte: elaborado pelo autor

7 AVALIAÇÃO DO JOGO

Nesta seção, serão descritas as etapas de testes que foram realizadas durante a concepção do jogo, desde a fase de prototipação até a fase de finalização do projeto.

7.1 Avaliação do protótipo do jogo.

A primeira avaliação foi realizada com o protótipo inicial do jogo, fiz a produção do protótipo e realizei a simulação de algumas ações que o jogador pode realizar. O objetivo inicial desta avaliação foi buscar possíveis erros de fluxo na medida em que as ações são tomadas.

Também foi importante analisar a capacidade do jogo punir o jogador que tenta não seguir os passos do BDD, logo aquele jogador que pula muitas casas, também foi importante verificar possíveis brechas e entraves em determinados cenários.

Para exemplificar, o teste tentou identificar se a ausência de uma determinada carta impossibilita o seguimento da jogatina. Durante a avaliação, foram identificadas algumas pequenas falhas na carta de código escrito, o jogador conseguia usá-la mesmo sem ter a carta de *User Story* isso implica ter o código sem ter o *User Story*, não tem como escrever um código sem saber a finalidade dele. Outra falha encontrada foi na carta de refatoração, ela poderia ser usada sem ter a carta do código escrito na mesa, visto que em um cenário real para ter refatoração tem que ter um código escrito logo não faz sentido esta carta ser usada sem que o jogador tenha a carta de código escrito. Essas falhas foram encontradas e melhorias foram propostas para a versão mais próxima da final, versão que foi usada no próximo teste (teste piloto).

Melhorias foram feitas nas cartas e nas regras para resolver essas falhas. Foram adicionadas algumas restrições de uso para tentar trazer uma exemplificação de um cenário real como, por exemplo, na carta de TDD (Código Escrito), após o primeiro teste foi decidido que ela só pode ser usada quando a carta de *User Story* estiver na mesa. A carta de refatoração só pode ser usada se o jogador estiver com uma carta de código escrita na mesa. Também foram feitas alterações nas casas para

ter um melhor balanceamento em relação ao pulo das casas (riscos), na casa 4 (Teste de *feature* do cenário) foi adicionado +1 de risco caso o jogador pule, na casa 5 (Escrever código do teste) foi adicionado +1 de risco caso o jogador pule e, por fim, na casa 7 (Refatoração de Código) a condição de pular e ganhar o risco +1 foi substituída pela condição: “se não possuir carta de refatoração o jogador pode pular”, a punição por pular está relacionada a carta de refatoração, ela diminui o risco logo a punição por pular a casa é não ter a oportunidade de diminuir o risco. Essa alteração impactou no nível de punição do jogo ao jogador que pulou muitas casas, com a adição da condição de aumentar os riscos nessas casas o jogo se tornou mais punitivo.

7.2 Avaliação do piloto

A avaliação foi realizada com a última versão do protótipo. O aluno do curso de Ciência da Computação já conhecia BDD. A duração do teste foi de aproximadamente 40 minutos. No primeiro momento foram explicadas as regras do jogo, casa por casa, e a função de cada carta. Após a explicação, o aluno jogou por conta própria sem qualquer auxílio.

No primeiro teste, na primeira casa, ele decidiu pegar a carta de *User Story*. Na segunda casa, ele decidiu montar a equipe de *stakeholders* com todos os tipos de membros possíveis. Por regra o jogo só permite o jogador pegar 3 cartas por vez e não permite pegar o mesmo tipo de *stakeholder* para fazer a equipe, o jogador levou 4 rodadas para pegar o último membro que faltava para completar.

Na próxima casa (Elaborar teste de aceitação), o jogador teve que elencar os critérios de aceitação com as cartas Dado, Quando e Então. Da mesma maneira que ele montou a equipe, o jogo permite que ele pegue 3 cartas por vez e não pode repetir as cartas, ele precisou pegar apenas uma vez logo na primeira tentativa ele pegou as 3 cartas (Dado-Quando-Então). Na casa seguinte (Testar a *feature* de cenário) ele fez o teste, o teste reprovou porque ele não pegou a carta código, na próxima casa ele pegou a carta de código feito, e na casa seguinte ele fez o teste novamente e o teste passou. Na próxima casa, ele refatorou o código pegando a carta de refatoração diminuindo o risco que era de 0 para -2.

Após refatorar o código, ele tem que testar. Logo, ele tem que pegar outra casa de testes, ele pegou a carta e o teste passou, pois ele tinha as cartas necessárias. Na última casa, foi aprovado pelo PO pois o risco estava -2. Então qualquer valor do dado daria maior, pela regra seria reprovado caso o valor do dado fosse maior ou igual que o valor do risco. Também foi aprovado pelo cliente, o valor da equipe era 6 e o dado deu valor 6 logo pela regra da casa seria aprovado caso o número da equipe fosse maior ou igual ao do dado.

No segundo teste, na primeira casa ele pegou a carta de *user story*, na casa de montar a equipe de *stakeholders* ele decidiu pegar apenas duas cartas de *stakeholders* (PO e DEV valor somado = 3). Ele juntou as 3 cartas de critérios de aceitação, ele pulou a casa de teste, risco ficou 1.

Na próxima casa ele pegou a carta de código (TDD), ele decidiu pular a casa de teste o risco foi para +2. Depois ele pulou a casa de refatoração risco foi para +3. Pulou a próxima casa de testar a código refatorado, pois ele não pegou a carta de refatoração. Na última casa, ele foi reprovado pelo PO, pois o valor do dado foi 3 logo igual ao risco. Logo, ele perdeu o jogo, um objetivo que eu quis atingir é o de punir o jogador que tome decisões erradas. No caso do segundo teste o jogador tomou algumas decisões de pular casas, no final foi reprovado e perdeu o jogo. As figuras a seguir mostram os critérios de aprovação.

A seção contém as observações do autor e do usuário sobre o teste piloto que foi realizado.

7.3 Observações do autor

Ele não teve dificuldades em jogar no primeiro momento. Ele percebeu a importância da equipe para a aprovação do cliente. Foi uma experiência fluida e ele não ficou muito tempo preso em uma casa sem entender o que ele deveria fazer. Ele percebeu que cada escolha é importante para a aprovação tanto do PO quanto do Cliente. Ele percebeu que é importante juntar a maior quantidade de *stakeholders* possíveis para melhor entendimento geral, levando ao sucesso do projeto.

Como foi relatado nos testes, o primeiro teste que ele fez, demorou 4 rodadas para conseguir montar a equipe completa pela restrição de apenas pegar 1 tipo de *stakeholder* por equipe. Essa falta de sorte na hora de pegar uma carta que necessita poderia ser prejudicial caso o jogador jogasse contra outro, ele ficaria preso por algumas rodadas para montar a equipe completa enquanto o outro jogador vai avançando rumo ao objetivo final.

Logo não coloquei a restrição de prosseguir apenas se o jogador montar a equipe completa, o jogo permite o jogador jogar com no mínimo um *stakeholder* na equipe, fica a critério do jogador colocar mais membros na equipe uma vez que ele fica ciente que quanto mais interessados (mais membros na equipe) as chances de aprovação aumentam.

7.4 Observações do usuário

Após a realização da avaliação do piloto, pedi para que o usuário fizesse alguns comentários a respeito da experiência que ele teve durante a avaliação e sugestões de melhorias para a versão final.

Ele gostou da dinâmica de decisões de escolher as cartas. Ele percebeu que o jogo pune qualquer decisão errada por ter pulado etapas. Ele gostou da mistura de tabuleiro com cartas e ele gostou do protótipo que achou que ele é bem próximo de uma versão final. Ele percebeu bem o processo como todo e que representa o BDD, ele viu familiaridade com processo de *Scrum*, ele achou o jogo bem autoexplicativo.

Gostou das regras de dependências do jogo, como não conseguir usar a carta de código caso ele não pegue uma carta de *User Story*. Ele percebeu que quanto melhor for a montagem da equipe de *stakeholders* maiores são as chances de sucesso. Ele observou que o jogo passou mensagem de que a comunicação e o entendimento do objetivo do cliente entre os membros é importante esse é um dos objetivos que quis passar com o jogo, o BDD preza muito pela boa comunicação entre

todos os interessados no sucesso do projeto, uma vez que o jogo passa essa mensagem ao jogador ele atinge esse objetivo.

Ele finalizou dizendo que a experiência seria ainda melhor caso ele jogasse contra alguém. Ele sugeriu embaralhar o *deck* cada vez que o jogador for pegar cartas para aumentar a aleatoriedade.

Ele citou como pontos negativos do protótipo a falta de ilustrações, as cartas deveriam ter que ter figuras para ilustrar o que cada uma representa, mas ele ponderou, pois se trata de um protótipo. Ele também sugeriu que o jogador conseguisse voltar para casas anteriores no momento que algo desse errado.

É uma sugestão válida, mas tornaria o jogo menos punitivo nas decisões do jogador pois uma vez que o jogador tome uma decisão errada e momentos depois ele decide voltar e tomar outra decisão o jogo não o puniria pela primeira decisão que ele tomou. O planejamento é que o jogo mostre ao jogador um contexto de ação e consequência, logo decisões incorretas no ciclo afetarão o resultado dele.

Logo, o jogador deve repensar melhor as suas decisões no próximo ciclo com a experiência que ele teve no ciclo anterior. Também vale ressaltar que se o jogo permitisse voltar casas o tornaria ainda mais demorado. Ele achou que tinha muitas cartas disponíveis, mas entendeu que não seria problema no caso de jogar com mais pessoas.

7.5 avaliação da versão final com o público-alvo.

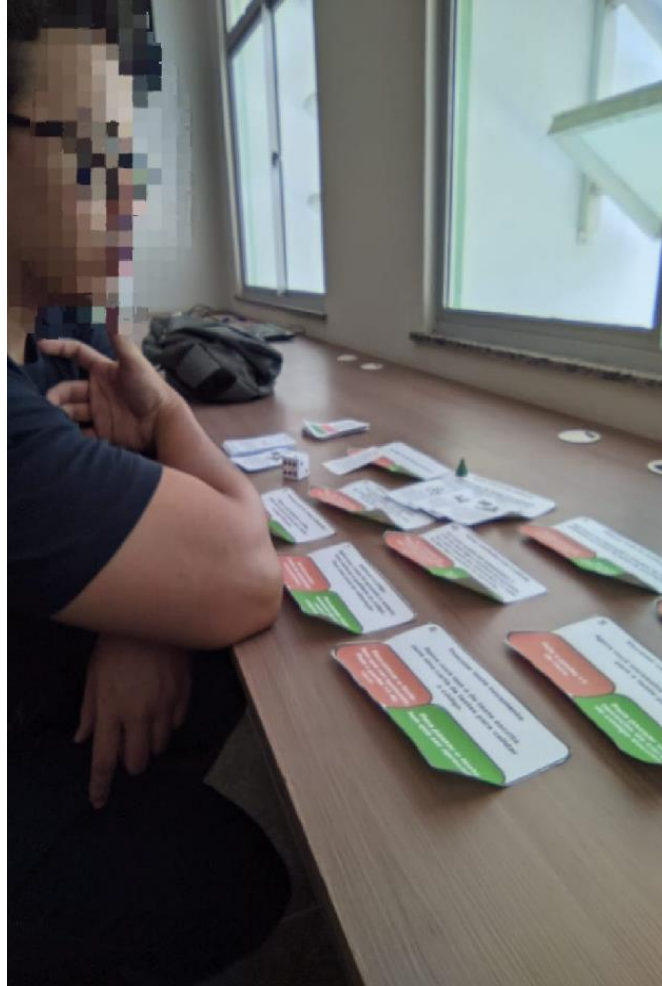
A avaliação versão final foi realizada com os alunos da Universidade Federal do Ceará no campus de Quixadá foi feita a explicação das regras e do funcionamento do jogo e sobre o que cada carta faz. Logo, após eles começaram a jogar e no fim foram feitas algumas perguntas e discussão sobre o que eles acharam do jogo. Depois foi aplicado o questionário para a análise de resultados e a validação para atingir os objetivos deste trabalho, que é fazer um jogo que represente o tema proposto.

Figura 22 – Avaliação da versão final do jogo



Fonte: próprio autor

Figura 23 – Avaliação da versão final do jogo



Fonte: próprio autor

8 RESULTADOS

Os resultados foram coletados após a aplicação do jogo. O jogo foi aplicado com alunos que possuem conhecimento sobre BDD tanto no âmbito acadêmico quanto no profissional. O questionário foi elaborado utilizando o modelo MEGAA+ com perguntas descritivas e objetivas essas seguindo a escala *likert*, onde os resultados são apresentados por 1) Concordo Totalmente, 2) Concordo, 3) Neutro, 4) Discordo e 5) Discordo Totalmente.

O modelo MEGAA+ é um modelo de avaliação de qualidade de jogo educativos usados para o ensino de temas de computação e engenharia de software. Ele avalia a experiência do jogador usabilidade, motivação, experiência do usuário e aprendizagem na visão de quem busca aprender.

Estava presente na aplicação e observei todas as interações feitas durante a aplicação. Para coletar todos os comentários feitos pelos alunos durante a avaliação. A seguir farei alguns comentários sobre os resultados tanto das perguntas subjetivas quanto das objetivas.

A primeira questão fala sobre o usuário achou do design do jogo, as respostas foram positivas, em resumo foram que as cartas e o tabuleiro foram bem elaborados, com informações claras e de fácil entendimento.

A segunda questão fala sobre se a experiência do jogador, ela pergunta se o jogador teve dificuldade para entender as regras, em resumo a resposta é de que as regras são claras e bem descritas e de que o fluxo é simples, mas que de início não ficou claro sobre quando os jogadores podem pegar as cartas dos *decks*.

A questão três fala sobre a dificuldade de entendimento do jogo. A maioria não teve dificuldades, por já conhecerem o BDD, mas um participante sugeriu a presença de alguém experiente para fazer a dinâmica de ensino, outro participante comentou que teve dificuldade para entender o fluxo do jogo pois era incomum para ele.

Na questão 4, onde a maioria avaliou que as experiências ajudaram no entendimento do jogo. O que pode dificultar de início a experiência de alunos e professores inexperientes ao tema, mas nas questões 5 e 15 onde de acordo com as respostas, o jogo é de fácil entendimento e adequado para o ensino que são características que podem ajudar a mitigar o impacto da ausência de conhecimento prévio sobre o tema de início.

Logo recomenda-se que se tenha uma contextualização sobre o tema antes da aplicação do jogo.

Na questão 5, o usuário 4 reportou que a cor vermelha nem sempre significa que o risco aumenta, de fato nem todas as casas aumentam o risco, mas em certos casos, a alternativa vermelha apresenta restrições, o jogo faz esse alerta. Acredito que pela ideia de dualidade de escolhas o jogador sempre espera que se não tomar a decisão boa(verde) então a vermelha vai puni-lo, mas nem sempre isso acontece no jogo. Uma cor que representa atenção resolveria esse problema, por exemplo o amarelo, fazendo uma analogia ao semáforo onde o amarelo significa quem nem pode seguir em frente nem que pode parar, mas sim para que ele tenha atenção ao sinal vermelho.

A questão 6 fala sobre fontes e cores, e questiona se prejudicou a experiência. Em geral não prejudicou, mas alguns usuários fizeram sugestões para a criação de uma versão voltada para acessibilidade utilizando cores que atendam essa necessidade.

A questão 7 pergunta se o jogo pune quando o jogador comete erros. A maioria achou que não, um fator que explica isso é o conhecimento prévio que eles possuem sobre o tema.

A questão 8 fala sobre a primeira impressão que eles tiveram ao ver o jogo. Os jogadores tiveram diferentes reações, um já teve de início que era um tabuleiro tradicional, o segundo achou que tinha muitas cartas disponíveis para os poucos jogadores isso implica uma avaliação futura para calcular a quantidade necessária de cartas que precisam ser produzidas de acordo com a quantidade de jogadores, o terceiro não assimilou que as casas eram o tabuleiro, e o quarto achou que o dado seria usado em todas as rodadas

Na questão 9, a respeito dos desafios oferecidos pelo jogo. Os 2 participantes da avaliação não se sentiram desafiados, logo não tiveram dificuldades em jogar o jogo, pois eles já possuem experiências práticas. Então conseguiram resolver os desafios do jogo sem dificuldades. Não é objetivo do jogo ser muito desafiador pois isso pode prejudicar a experiência de pessoas menos experientes. Uma maneira de deixar o jogo mais desafiador é adicionar o elemento de cenário, onde existiriam diversos cenários hipotéticos que adicionam limitações ao jogador, o que torna o jogo desafiador e mais complexo, que não é uma característica que acho que seria boa

para o jogo. Pois ele deve de maneira objetiva ensinar os passos do BDD e os elementos presentes nele para um primeiro entendimento prático.

A questão 10 é complementar a pergunta 9 onde desafios geram esforço, a pergunta fala sobre como o esforço pessoal o ajudou a passar no jogo. Um participante respondeu que “inventou uma história em cima da mecânica do jogo para dar mais sentido”, como na questão 9 a criação de cenários onde eles impõem ações que os jogadores devem tomar durante o ciclo ajuda a resolver essa deficiência. Isso exige um estudo mais aprofundado sobre como trazer os cenários reais onde os profissionais enfrentam para um jogo sem o tornar muito complexo e enjoativo para as pessoas iniciantes no assunto.

Na questão 11, uma participante respondeu que “O visual do jogo, mas a mecânica desprende a atenção rapidamente”, no teste os *decks* foram separados por tipo, logo ficava fácil do jogador pegar uma carta no momento que ele precisava dela ele somente pegava a carta do *deck* e jogava, para resolver isso as cartas foram agrupadas em diferentes *decks* para gerar mais dificuldade ao jogador de conseguir o recurso necessário fazendo com que ele preste mais atenção a mecânica de jogo.

Na questão 12, foi unânime a relevância do assunto para os participantes. Todos responderam que consideram o tema bastante importante para eles, o que reforça o conhecimento prévio que eles possuem sobre BDD.

A questão 13 fala sobre quais elementos chamaram a atenção dos jogadores. Foram citadas características visuais do jogo, a mecânica de montar equipes com diferentes níveis de importância e o gerenciamento de recursos com cartas repetidas.

Na questão 14 pergunta a importância do tema do jogo para os participantes. Foi unânime o nível de importância para eles, todos citaram que é bastante importante pois é usado no mercado de trabalho e que é importante para a construção do projeto.

A pergunta 15 avalia se o jogo ajudou no entendimento do tema (BDD) e que ele é adequado ao ensino, todos concordaram que sim, o jogo ajudou no entendimento e que ele é adequado ao ensino

Na questão 16 o usuário 1 respondeu que “Tradicional. O jogo explora bem o fluxo de BDD, mas não prende atenção o suficiente nas informações dispostas para fixá-las”, como se trata de um jogador com experiência sobre o assunto logo o jogo não prendeu tanta atenção dele. Para que o jogo melhore esse aspecto ele deve oferecer ao jogador mais desafios no qual o jogador se sinta desafiado e aumente a atenção. Acredito que com a adição de vários cenários fictícios nos quais impõem

restrições sobre os recursos que o jogo oferece através das cartas, o jogador deve ter mais noção sobre a importância deles e passarão a olhar mais para as cartas fazendo com que ele fixe melhor as informações dispostas por ele.

A questão 17 pede para que o participante descreva o processo de BDD.

Todos descreveram o processo de maneiras distintas, mas que são condizentes com o que o processo de BDD representa.

Após essas considerações conclui-se que o jogo possui alguns pontos para melhorias, mas que não afetam o com o que se diz respeito ao objetivo do trabalho. O jogo não impôs muitos desafios aos participantes da amostra da avaliação to mas, ponderasse que eles possuem conhecimento prévio sobre o tema. Para fazer essas melhorias se deve criar um ponto de equilíbrio no que se diz respeito ao público-alvo, no consiste em oferecer um jogo de fácil entendimento para pessoas inexperientes e ao mesmo desafiador para pessoas que já conhecem o tema.

Com os resultados obtidos pode-se concluir que o jogo foi bem avaliado e os seus objetivos foram alcançados. O jogo proposto e desenvolvido neste trabalho foi bem avaliado e de acordo com os dados ele representa o tema do trabalho e é considerado apto para o uso educacional mais detalhes sobre os resultados estão presentes no apêndice D.

Em relação aos objetivos propostos no início do trabalho, propor o jogo sobre BDD apesar de não ter sido testado com uma amostra mais significativa os resultados se mostraram satisfatórios e promissores. E para minimizar o impacto de uma amostra com poucos participantes, foram escolhidos participantes que já possuem conhecimento sobre o tema do jogo.

9 CONCLUSÃO

Este trabalho teve como propósito propor um jogo sério analógico que ensine o BDD para os alunos dos cursos de tecnologia da informação, de forma que auxilie o ensino sobre o tema.

Para que o jogo fosse elaborado e que alcançasse os seus objetivos foi necessário definir o escopo dele e definir os temas sobre o assunto a serem abordados no jogo. Foi possível notar um resultado positivo com os participantes da avaliação. Onde avaliaram positivamente o jogo e concluíram que ele é apto para o ensino.

Uma das preocupações em relação ao jogo é se ele abordaria realmente o tema e que fosse válido para o ensino. Para isso o jogo abordou o fluxo geral do BDD com os elementos que o compõem em um ambiente de jogo, mostrando então a importância de seguir os passos para conseguir alcançar a aprovação do cenário.

No geral apesar do jogo apresentar alguns pontos de melhorias, o trabalho se mostrou positivo pois demonstrou a capacidade de abordar sobre BDD em um contexto de jogo, e segundo as respostas se mostrou ser uma alternativa para o ensino sobre o tema para os alunos dos cursos de tecnologia da informação, logo o jogo alcançou os seus objetivos.

Para trabalhos futuros seria interessante se aprofundar ainda mais no processo de BDD para então adicionar cenários com restrições para tentar criar diversas simulações o que ocorrem no âmbito profissional, mantendo um bom balanceamento para os jogadores que possuem ou não experiência no assunto e mantendo ao máximo a atenção deles.

REFERÊNCIAS

- RIBEIRO, Tânia et al. iLearnTest: educational game for learning software testing. In: **10th Iberian Conference on Information Systems and Technologies (CISTI)**, [S.l.], p. 1-6, 2015.
- ARAUJO, Nayara et al. Avaliando a Viabilidade do BlackBox em Sala de Aula: um jogo sério para ensino de teste funcional de software. In: **Brazilian Symposium on Computers in Education** (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 817, out. 2017.
- SILVA, Cleibson et al. gTest Learning: Um Jogo para Ensino Básico de Teste de Software. In: **Congresso Regional sobre Tecnologias na Educação (CTRL+ e)**, [S.l.], p. 450- 458, 2016.
- SOUSA, Virginia Farias de. **iTest Learning** : um jogo para o apoio ao ensino de testes de software. 2012. 65 f. TCC (graduação em Sistemas de Informação) - Universidade Federal do Ceará, Campus Quixadá, Quixadá, 2012.
- DARWESH, Aso. Concepts of serious game in education. **International Journal Of Engineering And Computer Science**. [S.l.], 2016.
- OLIVEIRA, H. J. C. **Uma metodologia participativa para o desenvolvimento de jogos sérios**. [S.l.: s.n.], 2015.
- MEDEIROS, D. P. Jogos analógicos como ferramentas estratégicas para as marcas. **Design e Tecnologia**. [S.l.], v. 9, n. 17, p. 56-63, 10 jan. 2019.
- FEITOSA, Daniela Soares. **Um estudo sobre o impacto do uso de desenvolvimento orientado por testes na melhoria da qualidade de software**. [S.l.: s.n.], 2007.
- FRASER, S. et al. Test Driven Development (TDD). **Extreme Programming and Agile Processes in Software Engineering**, [S.l.], p. 459–462, 2003.
- SOEKEN, M.; WILLE, R.; DRECHSLER, R. Assisted behavior driven development using natural language processing. **Objects, Models, Components, Patterns**. [S.l.], p. 269–287, 2012.
- SOMMERVILLE, I. **Engenharia de software**. [S.l.]: Pearson Prentice Hall, 2011.
- YÁÑEZ-GÓMEZ, R.; CASCADO-CABALLERO, D.; SEVILLANO, J.-L. Academic methods for usability evaluation of serious games: a systematic review. **Multimedia Tools and Applications**. [S.l.], v. 76, n. 4, p. 5755–5784, 2016.

PEREIRA LS, FRAGOSO S. FCECF: um método interativo composto aplicado ao desenvolvimento de jogos analógicos. In: Proceedings of SBGames 2016, **Anais...** São Paulo, 2016.

DELAMARO, MARCIO EDUARDO; MALDONADO, JOSÉ CARLOS; JINO, MÁRIO. **Introdução ao teste de software**. [S.l: s.n.], 2016.

NASCIMENTO, Eduardo Henrique Rocha do. **Aplicando gamificação no ensino de teste de software**. 2019. 142f. Dissertação (Mestrado em Sistemas e Computação) - Centro de Ciências Exatas e da Terra, Universidade Federal do Rio Grande do Norte, Natal, 2019

APÊNDICE A – DESCRIÇÃO DO JOGO

O jogo será de tabuleiro com cartas, tem que ser jogado por uma pessoa ou mais, podendo ser jogado no modo cooperativo ou versus. No modo com mais de um jogador ganha quem conseguir a pontuação definida. O jogo contará com um dado onde em algumas casas ele é jogado para definir alguns resultados. O fluxo do jogo será o seguinte. O jogador deve seguir todas as etapas do BDD, caso o jogador escolher todos os elementos necessários e siga todas as etapas ele tem um bom resultado no final da rodada, caso contrário o jogo tenta punir o jogador que pule etapas e não escolha todos os elementos necessários fazendo com que o teste de aceitação seja recusado.

Regras

Elementos do jogo

- **Cartas**

Cartas de ação.

Elaborar user story

Escrever teste de aceitação(BDD)

Escrever código (TDD).

Cartas de teste.

Executar teste de código(TDD).

Cartas de integrantes da equipe

Carta do Product Owner

Carta do Quality Analyst

Carta do Dev

Carta do Final User

- Dado de 6 lados
- Tabuleiro
- Marcador de pontos

Ciclo do Jogo

Para pontuar o jogador deve passar pelas as casas do tabuleiro e passar pela fase final com o teste de aceitação aprovado, em caso de reprovação o jogador deve voltar para a casa inicial.

Casa 1 – Identificar as funcionalidades do negócio.

Para sair desta fase o jogador deve usar uma carta de user story, o user story ajuda a exemplificar o requisito de funcionalidade que o cliente deseja.

Casa 2 – Definir critérios de aceitação do user story

Para passar de fase o jogador tem que usar no mínimo uma carta de interessados, mas a probabilidade do cliente aceitar na fase de validação aumenta na medida que o jogador use mais cartas de interessados para passar de fase. Product Owner (PO), o Quality Analyst (QA), developer e final user.

Valores das cartas

PO = 2

QA = 2

Dev =1

Final User = 1

O jogador pode usar apenas uma carta de cada tipo. Essas cartas devem ficar na mesa até a fase de validação.

Casa 3 – Escrever teste de aceitação.

Para passar desta fase você necessita de uma carta cenário, ele deve juntar as cartas de exemplo no formato Gherkin (dado-quando-então). O jogador deve pegar 3 cartas do deck de BDD, caso ele não consiga pegar as 3 cartas necessárias ele passa a vez para o próximo jogador, caso ele esteja jogando sozinho ele pode pegar uma carta mas ele deve descartar a mesma quantidade de cartas que ele pega para a sua mão.

Casa 4 – Executar o teste da feature do cenário, o teste falha pois o código não foi implementado.

Para sair dessa casa você necessita pegar uma carta deck de testes, e para passar desta fase o teste tem que falhar. Logo o jogador não pode usar uma carta de código escrito.

Casa 5 – Escrever o código necessário para que o teste passe nesta fase será usado a carta de TDD.

Para sair da casa o jogador necessita de uma “Carta Código de Teste Escrito TDD”, esta carta descreve as 3 etapas do TDD.

Casa 6 – Teste passa

Nesta casa o jogador deve esperar que o teste passe pois ele possui as cartas necessárias (Carta cenário + Código de Teste Escrito), para sair dela o jogador precisa dessas duas cartas. Ao usar as duas cartas o jogador ganha uma carta “Código Aprovado(Green).

Casa 7 – Refatorar o código

O jogador necessita de uma carta de código aprovada na mão.

Casa 8 – Esperar que o teste refatorado passe

Nesta casa o jogador usa a sua carta de teste aprovado para ganhar uma carta de teste refatorado feito. O jogador decide se usa a carta de refatoração ou não. A carta de refatoração faz com que o código seja revisado e melhorado.

Casa 9 – Validação de resultados

Após a aprovação do código refatorado o jogador faz uma avaliação do cenário com o time e com o cliente será feita a pontuação do jogador de acordo com os pontos de riscos.

Esses são os valores das cartas de cenário

Um dado deve ser jogado para validar o cenário

Nesta fase o jogador vai precisar das cartas da equipe novamente

Se o valor da soma das cartas \geq Valor do dado, cenário aprovado, jogador pontua.

Se o valor da soma das cartas $<$ Valor do dado, cenário reprovado.

Cenário que foge muito do que o cliente espera, provavelmente o cliente irá recusar a entrega.

Ao final da rodada o jogador deve devolver as cartas para o final dos respectivos decks.

O critério de rodadas

Serão 3-5 Rodadas

Ganha o jogador que conseguir mais pontos ou conseguir uma quantidade determinada de pontos primeiro.

APÊNDICE B – QUESTIONÁRIO SOBRE A AVALIAÇÃO

Questionário sobre a avaliação

Gostaríamos que você respondesse as perguntas abaixo para ajudar na validação deste jogo. Todos os dados coletados serão somente usados para a pesquisa.

Nome:

Matrícula:

Curso:

O que você achou do design do jogo em geral(tabuleiro, cartas)?

Sobre a experiência do jogo,você teve dificuldades para entender as regras? Se sim descreva as dificuldades que você teve.

Você teve dificuldade em entender o jogo? Se sim quais.

Sobre as suas experiências anteriores, elas te ajudaram no entendimento do jogo?

- Discordo Totalmente
- Discordo
- Neutro
- Concordo
- Concordo Totalmente

O jogo é de fácil entendimento?

- Discordo Totalmente

- Discordo
- Neutro
- Concordo
- Concordo Totalmente

Sobre as fontes e cores, prejudicaram a experiência? Se sim, descreva pontos positivos e negativos.

Na sua opinião o jogo pune quando você comete erros?

- Discordo Totalmente
- Discordo
- Neutro
- Concordo
- Concordo Totalmente

Qual foi a sua primeira impressão ao ver o jogo?

Na sua opinião, quais desafios o jogo te oferece?

Como você acha que o seu esforço pessoal ajudou a avançar no jogo?

Você recomendaria este jogo aos seus colegas?

- Discordo Totalmente

- Discordo
- Neutro
- Concordo
- Concordo Totalmente

Na sua opinião como o jogo promoveu o sentimento de competição?

- Discordo Totalmente
- Discordo
- Neutro
- Concordo
- Concordo Totalmente

Quais elementos te chamaram a atenção durante a jogatina?

O quão relevante é o conteúdo do jogo para você?

O jogo ajuda no entendimento deste tema, você acha ele adequado para o ensino ?

- Discordo Totalmente
- Discordo
- Neutro
- Concordo
- Concordo Totalmente

Você prefere aprender através deste jogo ou através de uma maneira mais tradicional? Por quais motivos.

Descreva com as suas próprias palavras como é o processo de BDD.

APÊNDICE C – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Declaro, por meio deste termo, que concordei em participar da pesquisa referente ao projeto que visa a criação de um jogo sobre BDD(Desenvolvimento Orientado ao Comportamento). O projeto está sendo desenvolvido na Disciplina de Trabalho de Conclusão de Curso 2, na Universidade de Federal do Ceará, campus Quixadá. Afirmando que aceitei participar por minha própria vontade, sem receber qualquer incentivo financeiro ou ter qualquer bônus e com a finalidade exclusiva de colaborar para o sucesso da pesquisa. Fui informado(a) dos objetivos estritamente acadêmicos do estudo e me coloquei à disposição para colaborar. O acesso aos dados se farão apenas pelo avaliador e equipe responsável pelo projeto.

Quixadá, ____ de _____ de ____
Assinatura do(a) participante: _____

APÊNDICE D – RESPOSTAS DO QUESTIONÁRIO

Questão 1 – *O que você achou do design do jogo em geral (tabuleiro, cartas)?*

Usuário 1 – As cartas e o tabuleiro em termos visuais, foram bem elaborados, com informações claras sobre o funcionamento do jogo e estímulo visual o bastante para entender o que estava acontecendo e para que cada coisa servia.

Usuário 2 – Gostei das informações do tabuleiro para a proximidade ao jogador. Achei simples, algo bom pois quanto menos texto melhor para o jogo não ficar cansativo. Sobre as cartas, não vi a necessidade de leitura, a não ser que mudasse algo ou mostrasse exemplos

Usuário 3 – Bonito e funcional, pode ser disponibilizado como em Pdf e impresso em qualquer lugar.

Usuário 4 – Sinto que algumas regras poderiam ser colocadas nas cartas.

Questão 2 – *Sobre a experiência do jogo, você teve dificuldades para entender as regras? Se sim, descreva as dificuldades que você teve.*

Usuário 1 – Não, as regras são claras e bem descritas no tabuleiro.

Usuário 2 – No início eu me encontrei perdida para entender como iniciar o jogo. “Iriamos iniciar com uma carta? Como funciona a punição ao jogador?”Pensei. Mas ao decorrer do jogo, o fluxo ficou fiel ao tabuleiro e foi melhor de entender.

Usuário 3 – Sim o fluxo do jogo é bem simples, mas no começo do jogo me deixou perdido sobre o que fazer.

Usuário 4 – Inicialmente, por não saber quando pegar as cartas.

Questão 3 – *Você teve dificuldade em entender o jogo? Se sim quais.*

Usuário 1 – Apenas em relação a como o fluxo do jogo deveria funcionar, pois é bastante incomum

Usuário 2 – Não muito, na verdade . Já tenho conhecimento em BDD, então eu entendi normalmente a finalidade do jogo.

Usuário 3 – Não o jogo como um todo é simples, mas necessita de alguém experiente para fazer a dinâmica de ensino.

Usuário 4 – Não, pois jogo a muito tempo.

Questão 4 – *Sobre as suas experiências anteriores, elas te ajudaram no entendimento do jogo?*

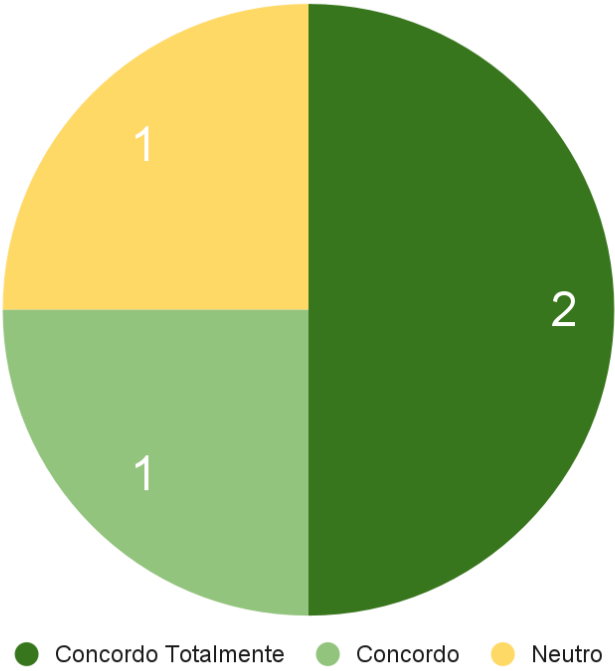


Gráfico 1: Resultado da quarta questão

Questão 5 – O jogo é de fácil entendimento?

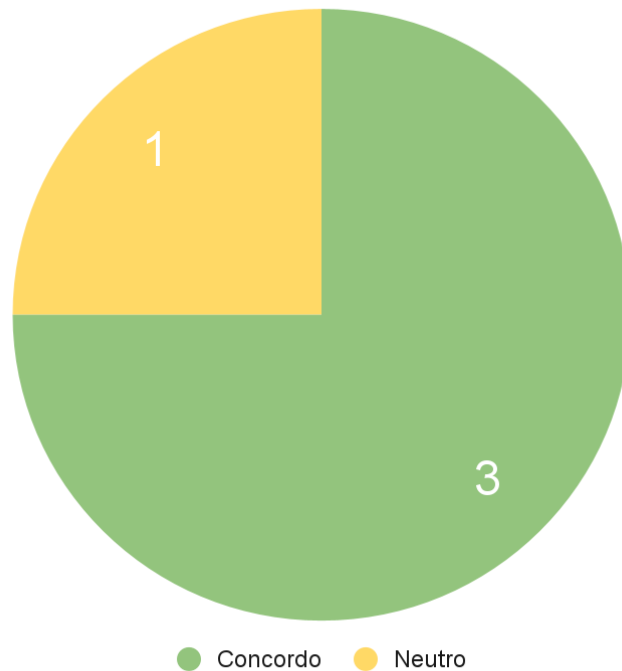


Gráfico 2: Resultado da quinta questão

Questão 6 – Sobre as fontes e cores, prejudicaram a experiência? Se sim, descreva pontos positivos e negativos.

Usuário 1 – Não, tudo estava visualmente compreensível

Usuário 2 – Não pra mim. Ficou bem simples, não incomodou. O que eu acho que daria problema seria a pessoa daltônica jogar o jogo. Identificar o verde e o vermelho seria problemático.

Usuário 3 – Talvez usar o verde e o vermelho faça necessário a criação de uma versão de acessibilidade mas pelos outros casos serviu bem.

Usuário 4 – Deveria ter uma cor para neutro, em algumas casas a decisão pela opção vermelha nem sempre aumentava o risco.

Questão 7 – Na sua opinião o jogo pune quando você comete erros?

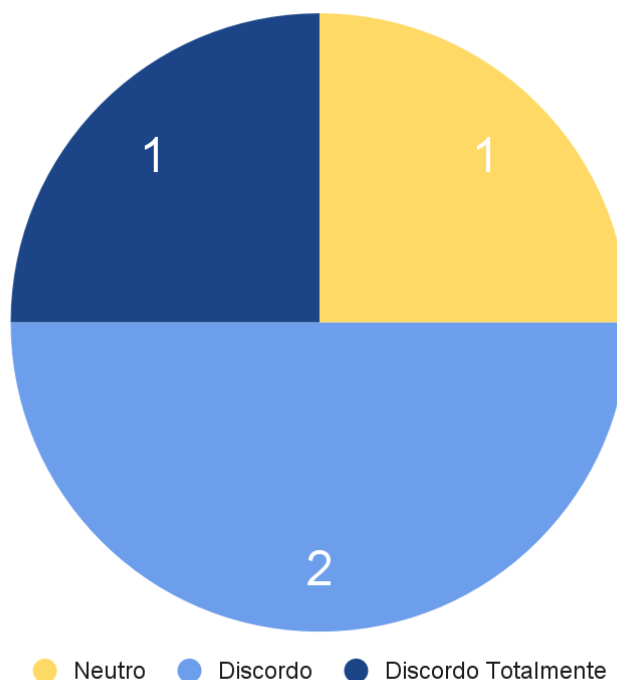


Gráfico 3: Resultado da sétima questão

Questão 8 – Qual foi a sua primeira impressão ao ver o jogo?

Usuário 1 – Que parecia um jogo de tabuleiro tradicional.

Usuário 2 – Muitas cartas para poucos jogadores mas bem completo

Usuário 3 – Não entendi que as cartas maiores eram o tabuleiro.

Usuário 4 – Achei que ia rodar o dado em todas as rodadas.

Questão 9 – Na sua opinião, quais desafios o jogo te oferece?

Usuário 1 – Nenhum. O jogo te faz escolher diretamente entre uma coisa boa e uma ruim sem haver nenhum motivo para escolher a ruim na maior parte das casas, a escolha é muito fácil.

Usuário 2 – Não tive muitos desafios, foi tranquilo jogar

Usuário 3 – O de conseguir pegar as cartas necessárias para passar de fase.

Usuário 4 – Pensar em como aquele cenário seria no mundo real.

Questão 10 – Como você acha que o seu esforço pessoal ajudou a avançar no jogo?

Usuário 1 – Não ajudou. A mecânica do jogo não incentiva esforço ou atenção.

Usuário 2 – O maior esforço que achei foi puxar as cartas, mas se eu tivesse escolhido pular, com certeza mudaria algo.

Usuário 3 – Lendo as cartas da minha rodada enquanto os outros jogavam para agilizar o andamento.

Usuário 4 – Inventei uma história em cima da mecânica do jogo, que ajudou a dar mais sentido.

Questão 11 - *Você recomendaria este jogo aos seus colegas?*

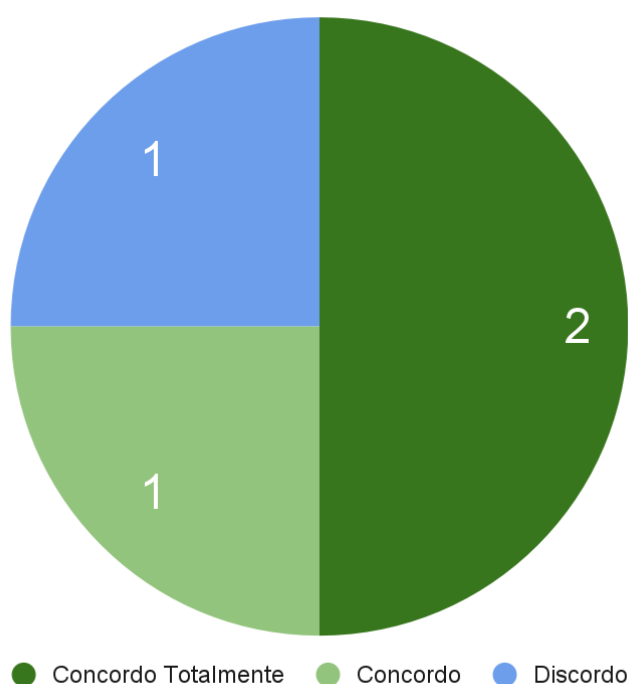


Gráfico 4: Resultado da décima primeira questão

Questão 12 – Na sua opinião como o jogo promoveu o sentimento de competição?

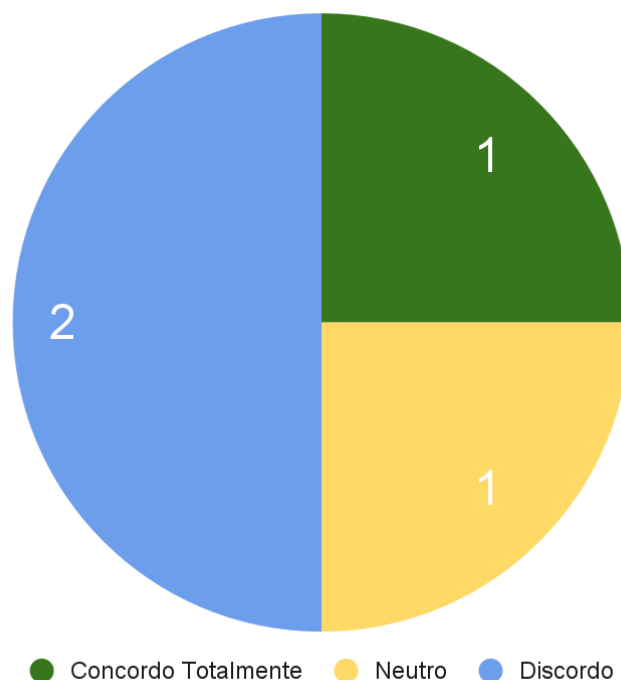


Gráfico 5: Resultado da décima segunda questão

Questão 13 – Quais elementos te chamaram a atenção durante a jogatina?

Usuário 1 – O visual do jogo, mas a mecânica desprende a atenção rapidamente.

Usuário 2 – A parte de montar a equipe pelas cartas puxadas, foi bem criativo e muito interessante. Gostei também do nível de importância contido nessas cartas.

Usuário 3 – Como as cartas repetidas seriam usadas ou se pode ter mais de uma carta implicava em gerenciamento de recursos.

Usuário 4 – As cartas e os dados.

Questão 14 – O quão relevante é o conteúdo do jogo para você?

Usuário 1 – Bastante, considerando que o fluxo de BDD é bastante usado no mercado

Usuário 2 – Bastante relevante, levando em consideração que, para cientistas, devs, engenheiros, etc. Obdd é bastante importante para a construção de um projeto.

Usuário 3 – Bastante, é uma área importante que poderia ser mais facilmente explicada com um jogo.

Usuário 4 – Bastante, trabalho na área e sei a importância.

Questão 15 – O jogo ajuda no entendimento deste tema, você acha ele adequado para o ensino?

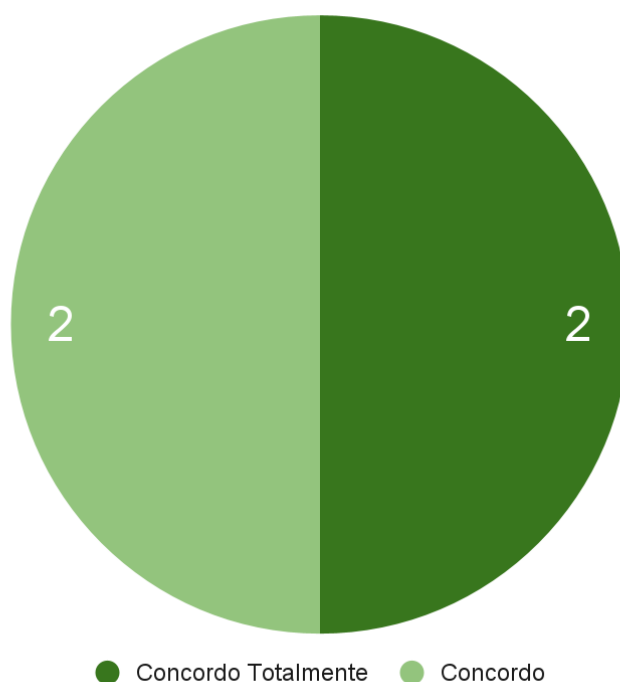


Gráfico 6: Resultado da décima quinta questão

Questão 16 – *Você prefere aprender através deste jogo ou através de uma maneira mais tradicional? Por quais motivos.*

Usuário 1 – Tradicional. O jogo explora bem o fluxo de BDD, mas não prende atenção o suficiente nas informações dispostas para fixá-las

Usuário 2 – Pelo jogo, acaba sendo mais legal. Para mim uma maneira diferente não fica entediante de aprender.

Usuário 3 – Pelo jogo, se for usado em uma aula seria bem interessante.

Usuário 4 – Pelo jogo, por ser mais divertido e intuitivo.

Questão 17 – *Descreva com as suas próprias palavras como é o processo de BDD.*

Usuário 1 – Profissionais de testes escrevem testes em cima do comportamento previsto nos requisitos e nas *user stories*, e desenvolvedores buscam alcançar este comportamento esperado durante o processo de desenvolvimento.

Usuário 2 – No início, devemos entender os requisitos do projeto e escrevê-los no formato “Dado que, Quando eu e então eu”. Isso é o mais básico mas é o que entendo de BDD.

Usuário 3 – Processo onde descrevemos um fluxo de atividades em passos concretos com começo, meio e fim desde a preparação até o resultado esperado. Dado -> Quando -> Então.

Usuário 4 – Escolher uma história para ser desenvolvida, definir o time, escolher cenário, preparar testes, testar, refatorar e validar com *stakeholders* e cliente.