

UNIVERSIDADE FEDERAL DO PAMPA

Gustavo Carvalho da Costa

Um Framework Conceitual de Apoio à
Gestão de Controle de Qualidade em Jogos
Digitais

Alegrete
2023

Gustavo Carvalho da Costa

Um Framework Conceitual de Apoio à Gestão de Controle de Qualidade em Jogos Digitais

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Software da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Engenharia de Software.

Orientador: Prof. Dr. João Pablo Silva da Silva

Alegrete
2023

Gustavo Carvalho Da Costa

Um Framework Conceitual de Apoio à Gestão de Controle de Qualidade em Jogos Digitais

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia de Software da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho de Conclusão de Curso defendido e aprovado em: 06 de julho de 2023.

Banca examinadora:

Prof. Dr. João Pablo Silva da Silva

Orientador

UNIPAMPA

Prof. Me. Jean Felipe Patikowski Cheiran

UNIPAMPA

Prof. Me. Miguel da Silva Ecar

Pub-SE Consultoria



Assinado eletronicamente por **JOAO PABLO SILVA DA SILVA, PROFESSOR DO MAGISTERIO SUPERIOR**, em 06/07/2023, às 22:17, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **Miguel da Silva Ecar, Usuário Externo**, em 07/07/2023, às 10:03, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **JEAN FELIPE PATIKOWSKI CHEIRAN, PROFESSOR DO MAGISTERIO SUPERIOR**, em 10/07/2023, às 15:08, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.

Este trabalho é dedicado à minha família,
em especial minha mãe Cláudia, meu pai Donel
e meu irmão Eduardo.

AGRADECIMENTOS

Primeiramente, queria agradecer a minha família, em especial a minha mãe Cláudia, meu pai Donel, meu irmão Eduardo e por que não, a minha querida cunhada Tamires, pois são pessoas que sempre estiveram ao meu lado, fornecendo apoio moral e suporte em momentos difíceis da minha vida pessoal e da vida acadêmica, no qual eu agradeço do fundo do meu coração.

Em segundo lugar, gostaria de agradecer a amigos pessoais que cultivei amizades durante a vida e também a amigos que felizmente encontrei na faculdade, pois sem eles a vida não teria graça nenhuma, e momentos de lazer e diversão são fundamentais durante a nossa caminhada, e eu tive o privilégio de andar ao lado de pessoas extremamente legais e de confiança.

Em terceiro lugar, agradecer o apoio e os puxões de orelha de professores que mais tive contato durante meu período na faculdade, em especial a Amanda Melo e João Pablo, já que são eles um dos pilares principais para a pesquisa e estudo na nossa universidade.

“Em um mundo repleto de miséria e incertezas,
é um grande conforto em saber que, no fim,
há luz na escuridão.”
(Joshua Graham)

RESUMO

O seguinte Trabalho de Conclusão de Curso trata sobre a área de controle de qualidade em jogos digitais, tendo como maior foco as etapas de testes e suas técnicas aplicáveis, qualidade do produto e feedback de usuários. O problema a ser explorado se baseia na necessidade das equipes de desenvolvimento de jogos em explorar, priorizar e implementar técnicas de teste que busquem uma qualidade do produto final de excelente nível, tendo em vista os requisitos inicialmente levantados. Neste trabalho foi proposto um software de recomendação que serve de apoio à gestão da etapa de controle de qualidade nas empresas desenvolvedoras de jogos, com o diferencial de prover um relatório adaptado ao contexto interno. De forma ad hoc, buscou-se estudos na literatura para compreender e mapear quais técnicas de teste estão sendo aplicadas na indústria e suas contribuições efetivas em projetos. Inicialmente, observou-se que existem muitas práticas e técnicas de teste sendo utilizadas na indústria. Porém, os estúdios de jogos não as usufruem de maneira otimizada por uma série de fatores, como gerenciamento de custos e cronograma. Para a validação do estudo, foi proposta uma pesquisa com profissionais da área de teste de software e/ou controle de qualidade focada em jogos digitais, onde houve a avaliação de 10 cenários com características em relação a etapa de testes. Com isso, foi desenvolvido uma aplicação preliminar que oferece suporte aos desenvolvedores fornecendo a descrição das melhores técnicas de teste à serem utilizadas. Por fim, percebe-se a relevância de um software de recomendação que dará suporte às equipes de desenvolvimento de jogos digitais, uma vez que não existe uma padronização de testes e ferramentas definidas ao contexto de cada jogo, além do fato de haver uma grande diferença de opiniões de profissionais da área sobre quais técnicas são melhores dependendo de um contexto específico.

Palavras-chave: Controle de Qualidade. Jogos digitais. Técnicas de teste. Qualidade de produto. Software de recomendação.

ABSTRACT

The following Graduation Conclusion Thesis addresses about the quality control field in digital games, focusing on testing phase and its techniques, product quality and user feedback. The problem to be explored is based on the game development team's necessity to explore, prioritize and implement testing techniques that will achieve an excellent final product quality, considering the requirements previously gathered. On this paper, it was proposed a recommendation software which will serve as a support tool for quality control phase management on gaming development companies, having the differential aspect to elaborate an adapted report considering the game's context. Using an ad hoc approach, studies were searched on the literature to comprehend and map what testing techniques and its effective contributions were being applied in game projects. Initially, it was observed that there are many testing practices and techniques being utilized on the gaming industry. However, game studios does not seem to take advantage of such techniques for a couple reasons, mainly because of schedule and cost management. For the validation of the study, a survey was proposed with professionals in the area of software testing and/or quality control focused on digital games, where there was an evaluation of 10 scenarios with characteristics in relation to the testing stage. With this, a preliminary application was developed that supports developers by providing a description of the best test techniques to be used. Finally, it is noted the relevance of a recommendation software which will give support to digital game development teams, since there is no testing and tools use standardization defined to the context of each game, in addition to the fact that there is a great difference regarding the opinions of professionals in the field about which techniques are better depending on a specific context.

Key-words: Quality Control. Digital games. Testing techniques. Product quality. Recommendation software.

LISTA DE FIGURAS

Figura 1 – Características da Qualidade em Uso exploradas na literatura	24
Figura 2 – Características da Qualidade do Produto exploradas na literatura	25
Figura 3 – Visão da metodologia aplicada	28
Figura 4 – Estrutura padrão de uma árvore de decisão (adaptado)	44
Figura 5 – Estrutura adaptada da árvore gerada pelo dataset da Iris	46
Figura 6 – Processo de seleção dos estudos	55
Figura 7 – Arquitetura geral da solução	61
Figura 8 – Estrutura parcial da árvore criada pelo algoritmo	67
Figura 9 – Exemplo de galho gerado pelo algortimo	68
Figura 10 – Exemplo de relatório gerado pelo sistema	69
Figura 11 – Exemplo de cenário proposto ao participante	73
Figura 12 – Exemplo de justificativa de cenário apresentado	73
Figura 13 – Cenário 1 do formulário	74
Figura 14 – Cenário 2 do formulário	75
Figura 15 – Cenário 3 do formulário	75
Figura 16 – Cenário 4 do formulário	76
Figura 17 – Cenário 5 do formulário	77
Figura 18 – Cenário 6 do formulário	78
Figura 19 – Cenário 7 do formulário	79
Figura 20 – Cenário 8 do formulário	79
Figura 21 – Cenário 9 do formulário	80
Figura 22 – Cenário 10 do formulário	81
Figura 23 – Avaliação de todos os cenários pelos participantes	82

LISTA DE TABELAS

Tabela 1 – Dados de treinamento (adaptado)	44
Tabela 2 – Termos de busca	50
Tabela 3 – Informações de busca	51
Tabela 4 – Critérios de Inclusão	51
Tabela 5 – Critérios de Exclusão	51
Tabela 6 – Critérios de Qualidade	52
Tabela 7 – Métricas dos estudos analisados	52
Tabela 8 – Dados para extração de dados	53
Tabela 9 – Informações dos estudos selecionados.	53
Tabela 10 – Resumo das respostas de cada participante	83

LISTA DE SIGLAS

CQ Controle de Qualidade

DLC Downloadable Content

FPS First-Person Shooter

GDD Game Design Document

MMO Massively Multiplayer Online Game

MOBA Multiplayer Online Battle Arena

QoL Quality of Life

RPG Role-Playing Games

UX User Experience

XP Experience Points

SUMÁRIO

1	INTRODUÇÃO	23
1.1	Motivação	25
1.2	Objetivo	26
1.3	Metodologia da pesquisa	26
1.4	Organização e Capítulos da Monografia	27
2	FUNDAMENTAÇÃO TEÓRICA E TECNOLÓGICA	29
2.1	Visão Geral do Desenvolvimento de Jogos Digitais	29
2.1.1	Papéis no Desenvolvimento de Jogos Digitais	31
2.1.2	Gêneros de Jogos Digitais	32
2.2	Controle de Qualidade	34
2.2.1	Práticas Aplicáveis de Controle de Qualidade em Jogos Digitais	36
2.3	Percepções e Feedback de Usuários de Jogos Digitais	40
2.4	Aspectos de Usabilidade em Jogos Digitais	41
2.5	Árvores de Decisão	43
2.5.1	Algoritmo de Árvores de Decisão	45
2.6	Lições do Capítulo	47
3	REVISÃO DA LITERATURA	49
3.1	Revisões Relacionadas	49
3.2	Protocolo de Revisão	49
3.2.1	Questões de Pesquisa	49
3.2.2	Processo de Busca	50
3.2.2.1	String de Busca	50
3.2.2.2	Informações de Busca	50
3.2.3	Processo de Seleção	51
3.2.3.1	Métricas dos Estudos Analisados	52
3.3	Estratégia de Análise e Extração de Dados	52
3.4	Discussão das Questões de Pesquisa	52
3.4.1	Resposta da Questão 1	55
3.4.2	Resposta da Questão 2	59
3.5	Ameaças à Validade	59
3.6	Lições do Capítulo	60
4	DESENVOLVIMENTO DA SOLUÇÃO	61
4.1	Arquitetura Geral do Protótipo de Software	61
4.2	Entradas da Solução	62
4.3	Cenários da Solução	63

4.4	Protótipo Preliminar de Apoio à Gestão de Controle de Qualidade	65
4.5	Lições do Capítulo	69
5	AVALIAÇÃO DA SOLUÇÃO	71
5.1	Planejamento da Avaliação	71
5.2	Execução da Avaliação	71
5.3	Análise dos Resultados	72
5.4	Lições do Capítulo	85
6	CONSIDERAÇÕES FINAIS	87
	REFERÊNCIAS	89

1 INTRODUÇÃO

Na indústria de jogos (especialmente em empresas *Triple A* (AAA) que possuem maior relevância no mercado), a qualidade final de seus produtos é um fator importantíssimo para seus jogadores (MCV, 2019). Em alguns casos recentes, jogos digitais que eram aguardados há meses apresentaram um resultado e *feedback* não muito desejados pela crítica (Exemplos: Tassi (2018) menciona o estado abaixo do padrão no lançamento de *Fallout 76*, MacDonald (2020) descreve o estado irregular do jogo *Cyberpunk 2077*, e Woog (2019) descreve como o jogo *No Man's Sky* sofreu inúmeras mudanças ao longo do tempo) em vários aspectos, como *bugs* que atrapalhavam a *gameplay*, detalhes de mecânica (*engine*) interna, *features* inovadoras e vendidas como carro chefe do produto, não atendendo a satisfação mínima, além outros fatores vitais. Neste contexto, essas empresas tiveram que realizar trabalho extra e lidar com custos fora do orçamento, cronograma extrapolado, pós-produção para atender a demanda e necessidade de seus jogadores, insatisfação de seus funcionários, fazendo com que houvesse uma sensação geral de “*whiplash*” (efeito adverso às expectativas) nas suas respectivas comunidades e internamente no ambiente de trabalho.

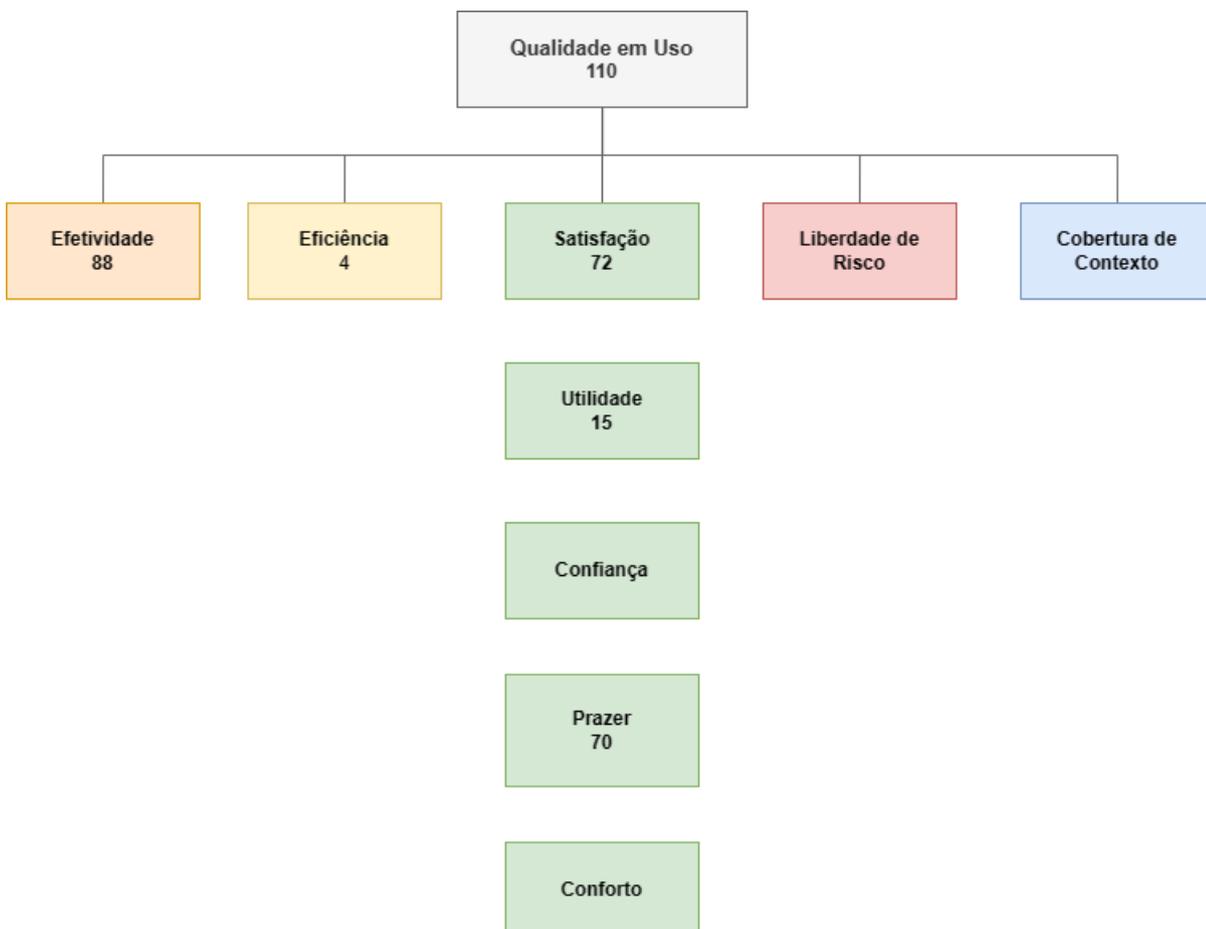
O processo de Controle de Qualidade (CQ) auxilia no momento de identificação desses problemas, em especial a etapa de verificação, que se baseia na ideia de “estou construindo o produto corretamente?” (BOEHM, 1984). Tarefas como teste de usabilidade, testes de aceitação pelos *stakeholders*, testes de integração, entre vários outros, conseguem mapear de forma ampla em quais aspectos um produto demonstra falhas e inconsistências comparadas aos requisitos levantados no início do desenvolvimento. Apesar disso, no mercado de jogos digitais tem se tornado uma prática “comum” lançar um produto em um estado de “baixa” qualidade e realizar *hotfixes* ao longo do tempo (FAHEY, 2022). Isso se dá ao fato de que primeiramente, desenvolver um jogo digital é muito complexo. A quantidade de recursos humanos para testar diversos aspectos de *gameplay*, documentar e consertar *bugs* se torna inviável, considerando as horas necessárias para realizar todas essas tarefas, tendo em vista o *deadline* para a entrega da versão final de um jogo, conciliando os interesses dos *stakeholders* (ALEMM; CAPRETZ; AHMED, 2018). Logo, torna-se muito mais prático lançar *patches* gratuitos para todos os jogadores através da internet, contendo diversas mudanças e correções, com a garantia de que uma vez que o jogo já foi lançado, não há nenhuma quebra ou infração de contrato com empresas de *marketing* e distribuição, caso houvessem atrasos no lançamento oficial, por exemplo (FAHEY, 2022).

A indústria de jogos digitais há muito tempo tem se definido como uma tentativa constante de criar estratégias e respostas para uma conjunção altamente instável, fluída e dirigida a crises, na qual gerenciar mercados, trabalhadores, consumidores e *commodities* progridem em uma improvisação incessante, na qual a solução de hoje se torna, da noite para o dia, o problema de amanhã (KLINE et al., 2003). Entender a estrutura

básica da indústria de jogos (comunidade de jogadores, mercado interno, negócios entre desenvolvedores e distribuidoras de jogos e ciclo de vida de desenvolvimento) torna-se um conhecimento valioso, especialmente no aspecto comercial (FULLERTON, 2014).

Desde o ano de 2007, o número de pesquisas acerca de qualidade em jogos digitais têm aumentado consideravelmente (VARGAS et al., 2014). Os resultados mostram que o modelo de qualidade mais investigado na literatura foi o modelo de Qualidade em Uso, enquanto que outros estudos focavam em características ou sub-características particulares do modelo de qualidade do produto. A Figura 1 apresenta as características da Qualidade em Uso exploradas na literatura, sendo os fatores de Efetividade, Satisfação e Prazer as mais citadas. Já a Figura 2 explora as características de Qualidade do Produto mais investigadas na literatura, sendo as mais citadas Usabilidade, Operabilidade e Aprendibilidade (VARGAS et al., 2014).

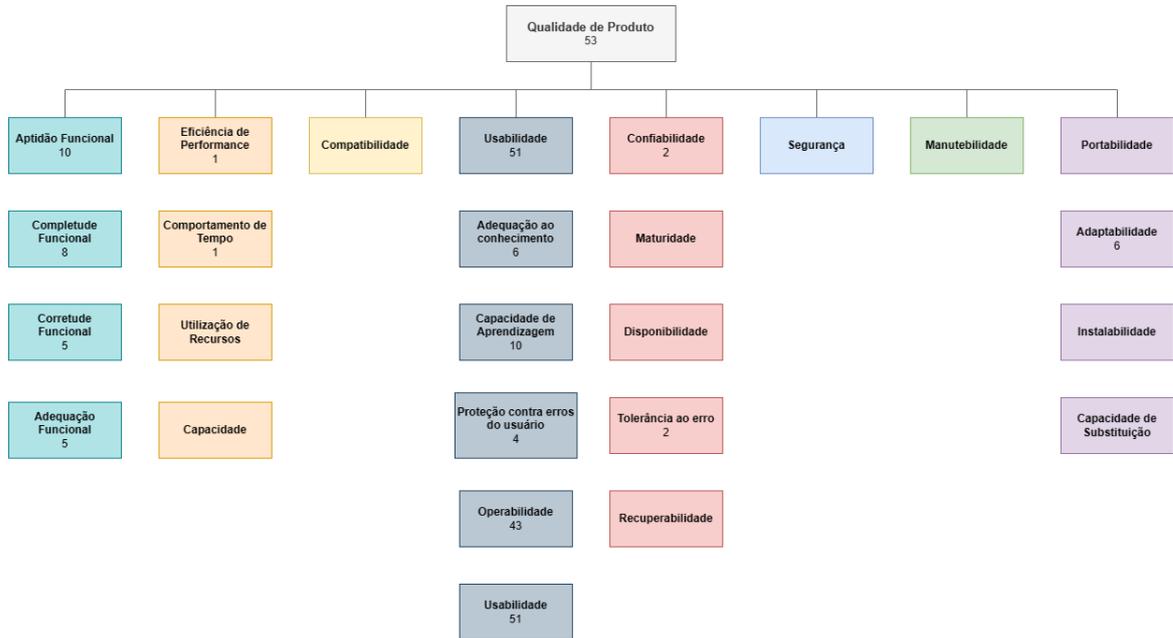
Figura 1 – Características da Qualidade em Uso exploradas na literatura



Apesar dos resultados serem otimistas, Vargas et al. (2014) afirmam ainda que os pesquisadores estão mais preocupados em demonstrar e confirmar a efetividade de jogos digitais em relação à sua capacidade de prover entretenimento do que o motivo pelo qual as características de qualidade fazem um jogo digital ser mais efetivo para seus jogadores.

Considerando o contexto atual do desenvolvimento de jogos digitais, torna-se re-

Figura 2 – Características da Qualidade do Produto exploradas na literatura



levante a compreensão e entendimento da aplicação de práticas robustas de CQ para que empresas minimizem o esforço necessário para testar e entregar produtos de altíssima excelência para seus clientes. Tendo isso em vista, este Trabalho de Conclusão de Curso se baseia em torno do entendimento e análise do processo de CQ em um contexto de jogos digitais, mapeando as técnicas de CQ sendo aplicadas na indústria e sua real efetividade dentro de projetos reais, aplicando esse conhecimento na elaboração de um software de recomendação para apoio à gestão ao controle de qualidade adaptável ao contexto, gerando como produto final um relatório que serve de suporte para a equipe de desenvolvimento através da utilização do *software* preliminar.

1.1 Motivação

Atualmente as empresas de desenvolvimento de jogos têm mantido uma prática de obter uma qualidade mínima e aceitável do produto final ao desenvolver apenas as funcionalidades que sejam realmente fundamentais (KASURINEN; SMOLANDER, 2014). Somado a isso, leva-se em conta o tempo dedicado para testar as funcionalidades do sistema e a prioridade de testes em geral. Uma pesquisa realizada por Neto et al. (2019) perguntando a relação média de tempo gasto com testes, mostrou que cerca de 50% dos entrevistados usa entre 1% e 25% do tempo com tarefas de teste, o que é um fator preocupante, tendo em vista a demanda de qualidade final do produto e seus resultados.

Observando a quantidade de falhas encontradas em diversos jogos digitais após o lançamento oficial, fica claro que em alguns projetos há falta e necessidade de uma etapa de controle de qualidade que forneça uma melhor e maior eficácia para que erros graves não estejam presentes na versão final do jogo. Logo, busca-se uma maior clareza em

torno desse processo tão fundamental, uma vez que diversos títulos lançados ultimamente apresentaram um resultado indesejável. Como se trata de uma etapa importante do desenvolvimento e é dada sequência na pós-produção (SYED, 2019), deve-se levar em conta a sua necessidade dentro do projeto e seu impacto na qualidade do jogo, observando as técnicas de teste que melhor se adaptam ao contexto do produto, as limitações de recursos, a satisfação de usuários, etc.

Tendo isso em vista, aspectos como: quais técnicas de teste estão sendo aplicadas em jogos digitais e suas reais contribuições para o valor agregado do sistema, serão investigadas neste trabalho, bem como métricas e aspectos de qualidade do produto e qualidade em uso.

1.2 Objetivo

O objetivo geral deste Trabalho de Conclusão de Curso é a elaboração de um software de recomendação flexível na gestão do controle de qualidade, para que seja aplicável e adaptável em qualquer projeto que esteja desenvolvendo jogos digitais (abrangendo jogos educacionais, jogos *indies*, etc.), oferecendo suporte mesmo na fase de pós-produção. O software tem como fundamento possibilitar a identificação e medição da qualidade do produto, disponibilizando técnicas de teste adaptadas ao contexto da aplicação e gerando um relatório avaliando aspectos internos.

Como Objetivos Específicos, atrelados ao Objetivo Geral apresentado, serão abordados os seguintes itens:

- promover a identificação na literatura sobre o estado da arte do controle de qualidade em jogos digitais;
- identificar práticas aplicáveis de controle de qualidade aplicadas à jogos e suas contribuições efetivas;
- mapear os aspectos de qualidade de produto presentes na ISO 25001 para aplicação real em jogos digitais;
- criação de um protótipo de software preliminar de apoio à gestão de controle de qualidade em jogos digitais adaptado ao contexto.

1.3 Metodologia da pesquisa

Este capítulo visa apresentar a metodologia utilizada para a realização deste trabalho. A Figura 3 apresenta as etapas para a escrita e elaboração deste Trabalho de Conclusão de Curso, buscando mapear o processo por completo, desde a concepção da ideia a ser solucionada, até a realização da técnica de *snowballing* e a etapa de avaliação dos trabalhos relacionados para responder as questões de pesquisa.

Inicialmente, foi proposta a elaboração inicial do Plano de Escrita para identificar a base e fundamentos para que houvesse a identificação de um problema para ser solucionado durante a realização do trabalho. Após isso, planejou-se o protocolo para mapeamento sistemático na literatura, no qual houve a definição das questões de pesquisa e *strings* de busca. Com isso, foram realizadas pesquisas nas bases de dados do Google Scholar e SCOPUS para identificação de artigos acadêmicos referentes a área de estudo, analisando estudos que atendiam os requisitos os critérios de inclusão, removendo artigos que estavam de acordo com critérios de exclusão e definindo a qualidade do estudo através dos critérios de qualidade. Uma vez que os artigos considerados sementes foram encontrados, ou seja, os principais trabalhos que servirão como base ao estudo, iniciou-se o processo da aplicação da técnica de *snowballing*.

A técnica de *snowballing* se baseia na ideia de analisar as referências dos estudos previamente coletados de diferentes bases de dados, com o intuito de identificar novos estudos que possam colaborar para a pesquisa, além dos identificados através do uso de **strings** de busca (JALALI; WOHLIN, 2012). Neste estudo, foram utilizadas duas etapas do *snowballing*: o *backward*, que analisa as referências dos estudos já coletados, e o *forward*, que analisa estudos que mencionaram os trabalhos já coletados. Com isso, rodadas de análise são executadas com ambas as técnicas para identificação de novos estudos que possam agregar ao estudo, até não haver mais trabalhos novos identificados.

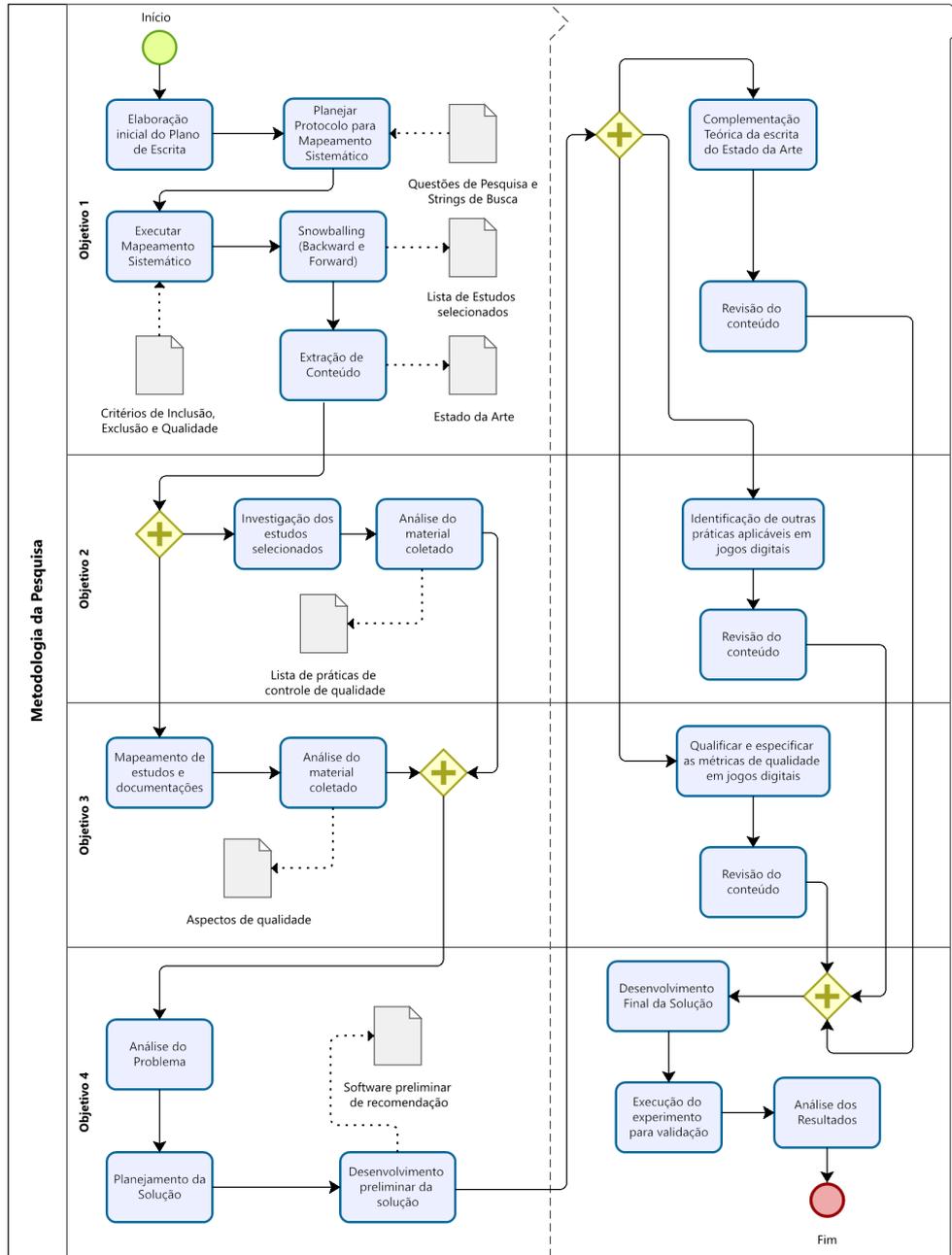
Após a seleção de uma lista de estudos utilizando a técnica de *snowballing*, analisaram-se os conteúdos de cada artigo e estudo identificado, afim de compreender e gerar o estado da arte do trabalho. Com isso, investigou-se o material coletado com o objetivo de gerar uma lista com as práticas de controle de qualidade, além dos aspectos de qualidade. Após isso, o problema de estudo foi analisado e planejado, no qual se buscou desenvolver um protótipo de software para apoio à gestão de controle de qualidade em jogos digitais.

Após as devidas correções do trabalho, buscou-se a complementação teórica da escrita do trabalho, além da identificação de novas práticas de teste aplicáveis em jogos digitais. Além disso, foi realizada uma avaliação com profissionais da área de teste de software focada em jogos para que houvesse a validação dos dados contidos na solução preliminar, que consiste em uma aplicação baseada em árvores de decisão. Por fim, os resultados da avaliação foram analisados e conclusões foram discutidas a cerca de todo o trabalho realizado até então.

1.4 Organização e Capítulos da Monografia

O trabalho está organizado da seguinte forma: No Capítulo 2, é abordado a fundamentação teórica para entendimento da solução, buscando adaptar o leitor ao tema estudado. No Capítulo 3, é apresentado de que maneira esta pesquisa foi realizada, incluindo os protocolos e processo de busca, além dos critérios para seleção dos materiais utilizados. No Capítulo 4, é descrita a solução proposta deste Trabalho de Conclusão

Figura 3 – Visão da metodologia aplicada



de Curso em profundidade, tendo em vista a motivação que levou ao resultado final e demais resultados. No Capítulo 5, é explicado como a pesquisa foi planejada, executada e seus resultados, além da apresentação do software desenvolvido. Por fim, no Capítulo 6, é apresentado as considerações finais, além de discussões a respeito dos resultados alcançados até o momento da aplicação do formulário com profissionais da área de testes e o desenvolvimento do *software* de apoio à gestão de controle de qualidade em jogos digitais.

2 FUNDAMENTAÇÃO TEÓRICA E TECNOLÓGICA

O presente Capítulo possui como propósito realizar e prover ao leitor a fundamentação teórica sobre o tema de pesquisa em questão, abordando assuntos de extrema importância para o entendimento do estado da arte do Controle de Qualidade em Jogos Digitais, servindo como base para compreender a visão geral da solução proposta no capítulo 4. O capítulo está organizado da seguinte forma: a Seção 2.1 apresenta uma visão geral do desenvolvimento de jogos digitais, tendo como base o estado atual do desenvolvimento e como está sendo realizado na indústria. A Seção 2.2 apresenta o referencial teórico e o estado da arte da área de controle de qualidade. A Seção 2.3 aborda as principais percepções de jogadores a respeito do uso de jogos digitais. A Seção 2.4 discute os aspectos de usabilidade utilizados para a avaliação deste trabalho e, por fim a Seção 2.5 que apresenta os conceitos sobre o algoritmo de árvores de decisão.

2.1 Visão Geral do Desenvolvimento de Jogos Digitais

O desenvolvimento de jogos digitais provou ser uma tarefa que vem aumentando sua complexidade nos últimos anos e, ainda, uma das áreas que mais crescem na economia mundial (AMPATZOGLOU; STAMELOS, 2010). Isso ocorre pois os consumidores finais do produto, ou seja, os jogadores, buscam uma experiência totalmente nova e repleta de expectativas comparada aos jogos anteriormente lançados na indústria. Uma vez que os desenvolvedores notaram esta demanda, foi necessária uma adaptação em relação ao aumento de funcionalidades de maior complexidade, melhoria da qualidade da escrita e conexão do roteiro da história, *assets* de alto nível (referente aos gráficos e animações comparados à vida real), entre outros fatores (LAHTI et al., 2015).

O ciclo de vida do desenvolvimento de jogos digitais pode variar de acordo com o projeto e a sua duração, porém, é seguido um processo definido e que se tornou padrão na indústria. Schultz e Bryant (2016) e Aleem, Capretz e Ahmed (2016b) definem as etapas e seus detalhamentos, tendo em vista o que está sendo aplicado na vida real: **Planejamento de Conceito, Fase de Pré-Produção, Fase de Produção e Fase de Pós-Produção.**

A fase de Planejamento de Conceito trata-se da parte genérica a respeito da concepção de um jogo digital. Nessa etapa, o time que está planejando os detalhes é relativamente pequeno, abrangendo um designer, um artista de conceito, um produtor e um líder de tecnologia. O foco principal é decidir sobre o que se trata o jogo em questão para que haja compreensão e entendimento fundamental de todos os envolvidos. Aspectos iniciais como elementos da *gameplay*, artes conceituais (como o jogo irá aparecer na tela) e pontos da história, são tratados nessa fase.

Uma vez que a fundamentação do jogo esteja plena e validada, os desenvolvedores iniciam a caracterização dos aspectos já vistos de uma forma mais detalhada e voltada ao produto final. Descrições de gênero, *gameplay*, funcionalidades, cenário, história, au-

diência alvo, lista de plataformas para lançamento, cronograma e estimativas de custos, análise competitiva, análise de riscos, recursos humanos, entre vários outros, são considerados para prover a contextualização completa e identificar pontos fundamentais antes da fase de Pré-Produção (SCHULTZ; BRYANT, 2016).

A fase de Pré-Produção possui como fundamento completar o conceito do jogo e o planejamento do projeto (NETO et al., 2019), elaborando o artefato chamado Documento de Projeto do Jogo (em inglês, *Game Design Document (GDD)*), que provê a descrição dos componentes básicos e suas respectivas inter-relações, especificações de requisitos funcionais e não-funcionais, contendo ainda direções para alcançar um desenvolvimento eficiente (ALEEM; CAPRETZ; AHMED, 2016b). Deve-se levar em consideração que, uma vez estabelecida a importância do GDD, a qualidade do produto final está fortemente ligada à qualidade e especificação do GDD, afetando negativamente em aspectos como retrabalho e perdas financeiras (SALAZAR et al., 2012). Outros artefatos relevantes elaborados na Pré-Produção são o Plano de Produção de Arte e o Documento de Projeto Técnico.

A Fase de Produção possui como foco a implementação de código-fonte e elaboração do conteúdo pelo time de desenvolvimento, atuando em melhorias incrementais e eficientes, tendo em vista o que foi estabelecido nas etapas anteriores (ALEEM; CAPRETZ; AHMED, 2016b) (LAHTI et al., 2015). Mudanças na mecânica central do jogo são evitadas, uma vez que *assets* como mapas e animações são feitas a partir das informações coletadas anteriormente.

Após o desenvolvimento do jogo digital estar fundamentalmente concluído, inicia-se a etapa de testes para um grupo limitado de pessoas, que pode ser dividida em duas etapas: fases *Alpha* e *Beta* de testes. A fase *Alpha* pode ser definida como o ponto de partida onde seja possível jogar do início ao fim do jogo, contendo a estrutura geral de interface do usuário e mecânicas internas, mesmo que alguns *assets* estejam faltando (SCHULTZ; BRYANT, 2016). Ambos testadores e jogadores selecionados observam o comportamento do jogo em diversos aspectos, procurando melhorias e defeitos durante a *gameplay* realizada. Nesta fase, o time de controle de qualidade deve manter um banco de dados de todos os defeitos e *bugs* encontrados para serem corrigidos posteriormente, possibilitando o progresso para a fase *Beta* de testes. Novak (2011) garante que alguns requisitos devem ser considerados para que a fase *Alpha* seja concluída com sucesso, como a possibilidade de jogar do início ao fim, harmonia com a documentação preliminar, compatível com as plataformas e configurações de hardware e software, além de arte e áudio estarem funcionando de acordo com o estabelecido. Caso o jogo atenda a essa checklist, inicia-se a etapa *Beta* de testes, onde todos os recursos e *assets* são integrados, o feedback dos usuários e requisitos de novas funcionalidades são analisados e defeitos identificados são corrigidos (SCHULTZ; BRYANT, 2016).

Por fim, quando o jogo é lançado para o público geral, inicia-se a fase de Pós-

Produção (ou fase *Gold*). Atividades de suporte como atualizações, melhorias de qualidade de vida (no inglês, *Quality of Life (QoL)*), conteúdos extras (no inglês, *Downloadable Content (DLC)*), gestão da comunidade de jogadores e marketing são fornecidas aos usuários (MANNINEN et al., 2006). Essas atividades podem ser lançadas e oferecidas de forma conjunta ou isolada, sendo gradualmente adicionadas e refinadas como conteúdos extras (GODOY; BARBOSA, 2010). Esta prática recente de publicar um jogo digital na data de lançamento prevista (mas com uma baixa qualidade final) e ao longo do tempo ir publicando *patches* (que significam atualizações) gera um desconforto aos jogadores, onde dependerão desses conteúdos extras, correções de *bugs* e melhoria de funcionalidades para aproveitar de forma completa o jogo mesmo após algumas semanas da compra ter sido realizada (FAHEY, 2022).

2.1.1 Papéis no Desenvolvimento de Jogos Digitais

As empresas possuem times de diferentes tamanhos e organização interna, contendo pessoas com habilidades e conhecimentos técnicos diversos. Esse conjunto de capacidades deve ser trabalhado em conjunto para completar as atividades necessárias na implementação de um jogo digital (SCHULTZ; BRYANT, 2016), garantindo que o produto final seja de qualidade e que atenda aos requisitos. Novak (2011) e Manninen et al. (2006) caracterizam os principais cargos e atribuições no desenvolvimento de jogos digitais, o que incluem: **Produtor, Time de Gestão, Diretor do Projeto, Time de Projeto, Diretor Técnico, Time de Programação, Diretor de Controle de Qualidade, Time de Teste, Diretor de Arte, Time de Arte, Diretor de Áudio e Time de Áudio.**

O Produtor de um jogo digital provê a liderança para o restante dos times de desenvolvimento. Geralmente, não possui envolvimento na criação de conteúdo para o jogo (SYED, 2019), porém, suas principais atribuições giram em torno de apresentar o progresso do jogo, resolver problemas que surgem e garantir que o produto seja entregue no tempo estipulado, dentro do orçamento e o mais perto possível da visão apresentada no Documento de Projeto do Jogo (CHANDLER, 2009). Em relação ao Time de Gestão, o mesmo orienta e oferece suporte à várias áreas do desenvolvimento, que incluem produção de arte, *level design* (em português, projeto de níveis, é o processo de criação de mapas, missões e estágios ou fases internas do jogo) (DEMIRCIN, 2022), interface de usuário, áudio, comunicações entre licenciadores e empresas de jogos, etc. (NOVAK, 2011).

O Diretor de Projeto é responsável pelo design geral do jogo, sendo o principal contribuidor do Documento de Projeto e possui a visão criativa completa, determinando se o produto final é agradável de se jogar ou não (SCHULTZ; BRYANT, 2016). O Time de Projeto possui a função de definir e elaborar a parte criativa por trás do jogo, onde consiste de diferentes cargos como designer de mapas e missões, *designer* de interface, escritores, pesquisadores e *designer* de *scripts* (NOVAK, 2011).

O Diretor Técnico possui como principal responsabilidade a arquitetura do jogo, possuindo também como atribuições a criação do Documento de Projeto Técnico, gestão de software, construção do jogo e deve entregar um produto de qualidade ao final do desenvolvimento (LAHTI et al., 2015). O Time de Programação é constituído por diferentes papéis internamente, e entender cada aspecto do jogo em que eles interagem é de suma importância (SYED, 2019). Suas respectivas habilidades em programação são o diferencial para construir o jogo a partir do Documento de Projeto, trabalhando em equipe para realizar as tarefas atribuídas dentro do cronograma e com o menor número de defeitos (SCHULTZ; BRYANT, 2016).

O Diretor de Controle de Qualidade (ou similar) é responsável por criar o plano de testes e gerenciar os testadores durante o projeto (NOVAK, 2011), geralmente quando se estão trabalhando com contratos baseados em marcos e conquistas (LAHTI et al., 2015). Outras funções serão discutidas em detalhes na Seção 2.2 que trata sobre controle de qualidade. A equipe de testes é uma parte crucial no time de desenvolvimento. O time irá testar o conteúdo criado para garantir a sua qualidade de acordo com os requisitos. Aspectos como interface do usuário, física do jogo, colisões, efeitos sonoros, progressão do jogo, jogabilidade, entrada de controles, compatibilidade de plataforma, entre vários outros, são testados por eles (SYED, 2019).

O Diretor de Arte é o principal autor responsável pelo guia de estilo de arte e aspectos visuais do jogo, promovendo gestão ao time de arte (NOVAK, 2011). Além disso, possui o trabalho de gerenciar a qualidade dos *assets* de arte, garantindo que a visão artística seja mantida durante a realização do projeto (CHANDLER, 2009). O Time de Arte possui como responsabilidades a criação de texturas, modelagem, animação, artes de interfaces do usuário, iluminação de ambientes, caracterização de personagens, vídeos cinemáticos, etc. (NOVAK, 2011).

O Diretor de Áudio é responsável por todos os efeitos sonoros, diálogos, música ambiente e instrumental. Geralmente, as empresas de desenvolvimento utilizam ferramentas terceirizadas para a gestão e base de todos os sons e músicas disponíveis (SCHULTZ; BRYANT, 2016). O time de áudio trabalha diversas particularidades para garantir que a imersão do jogador seja a melhor possível. Quando uma ação é realizada (por exemplo, abrir o menu principal, andar sobre uma superfície de um material específico, quebrar uma janela, etc.) um som diferente será reproduzido para que haja uma harmonia entre os itens que são interativos com o usuário (SYED, 2019).

2.1.2 Gêneros de Jogos Digitais

Jogos digitais são comumente classificados em gêneros, no qual auxiliam a definir o jogo em termos de possuir um estilo ou um conjunto de características em comum, classificadas em termos de perspectiva, *gameplay*, interatividade, objetivos, etc (NOVAK, 2011). Os principais gêneros de jogos digitais atualmente são: **Ação**, **Aventura**, **First-**

Person Shooter, Plataforma, Puzzle, Role-Playing Game, Estratégia, Terror e Sobrevivência.

O gênero de Ação é definido por uma jogabilidade dinâmica, geralmente por conter aspectos como movimentação do personagem, coordenação dos olhos e tempo de reação, testando as habilidades do jogador (NOVAK, 2011). Jogos como as franquias *DOOM*, *God of War* e *League of Legends* são exemplos de jogos de ação.

O gênero de Aventura é definido por jogos que possuem uma jornada, experiência ou evento inesperado e extraordinário no qual o jogador participa. Geralmente esse gênero foca na resolução de quebra-cabeças e mistérios, ao mesmo tempo que caminha com a narrativa do jogo (NOVAK, 2011). A franquia *The Legend of Zelda*, *Death Stranding* e *It Takes Two* são exemplos de jogos de aventura.

O gênero de *First-Person Shooter (FPS)*, em português, jogo de tiro em primeira pessoa, trata-se de colocar o jogador em uma perspectiva da visão humana, aos olhos de seu personagem, onde o foco primário é o combate contra inimigos ou outros jogadores, utilizando vários tipos de armamento (NOVAK, 2011). A franquia *Halo* e *Counter-Strike* são grandes exemplos de jogos *FPS*.

O gênero de Plataforma cobre uma grande variedade de jogos que ainda mostra suas raízes aos clássicos jogos 2D, como as franquias *Super Mario* e *Sonic*. Empresas *indies* preferem esse tipo de gênero por causa da sua simplicidade de controlar o personagem e geralmente possuem um estilo de arte simples e agradável (NOVAK, 2011).

O gênero de *Puzzle*, em português, quebra-cabeça, instiga o jogador a resolver problemas para continuar o progresso da história e narrativa, além de ter a possibilidade de desbloquear áreas exploráveis e recompensas escondidas (NOVAK, 2011). Jogos como *Portal 2* e a franquia *Tomb Raider* são exemplos de jogos de *Puzzle*.

O gênero de *Role-Playing Games (RPG)*, em português (jogo de interpretação de papéis), possui como princípio fundamental controlar um personagem que possui atributos e perícias, onde é possível ganhar pontos de experiência (do inglês, *Experience Points (XP)*) ao realizar missões, tarefas, etc. Como esse gênero possui um leque enorme de subcategorias, classificá-los de forma precisa é um trabalho complexo (NOVAK, 2011). Exemplos de *RPGs* famosos na indústria são: franquia *Fallout*, *The Elder Scrolls*, *World of Warcraft*, *Bioshock*, entre inúmeros outros jogos.

O gênero de Estratégia coloca o jogador em várias situações onde suas decisões possuem um impacto significativo em todos os aspectos. Possui como fundamentos um planejamento ao longo prazo, análise de acontecimentos e pensamento criativo para alcançar a vitória (NOVAK, 2011). A franquia *Civilization*, *Warhammer 40,000* e *Age of Empires* são exemplos de jogos de estratégia.

O gênero de Terror abrange jogos que focam no aspecto psicológico, tensão na imersão da jogatina que o jogador experiencia, além de utilizar da mecânica de *jump scares* (sustos inesperados que podem acontecer a qualquer momento, sem aviso prévio)

(NOVAK, 2011). *Alien: Isolation*, *Resident Evil* e *Outlast* são exemplos de jogos de terror.

O gênero de Sobrevivência possui como lógica a gestão de recursos, muitas vezes incorporando um sistema de criação de itens que o jogador pode utilizar para manter seu personagem vivo (NOVAK, 2011). Exemplos de jogos notáveis de sobrevivência são *Don't Starve Together*, *Green Hell* e *The Long Dark*.

Outros gêneros, sub-gêneros e junção de gêneros existem na indústria de jogos, como *tower defense* (em português, defesa de torres), simuladores, esportes, luta e combate, corrida, musicais, educacionais, exercício físico, furtivo, *Massively Multiplayer Online Game (MMO)*, em português, jogo online multijogador massivo, *Soulslike* (termo em inglês e recente que faz referência aos jogos que são semelhantes ao *Demon's Soul* em sua jogabilidade única), *Multiplayer Online Battle Arena (MOBA)*, em português, arena de batalha online multijogador, entre tantos outros (NOVAK, 2011).

2.2 Controle de Qualidade

O Controle de Qualidade (CQ) se baseia no conceito de minimizar o número e impacto de *bugs*, falhas e outros comportamentos não desejáveis antes que o produto seja totalmente finalizado e disponibilizado aos usuários finais. CQ garante que cada produto de trabalho atenda aos requisitos impostos por meio de revisões, inspeções e especialmente, testes através do processo de desenvolvimento (PRESSMAN, 2005). Qualquer *software* ou aplicação desenvolvida com o intuito de ser utilizada por usuários precisa ser validada, e esse processo continua mesmo na fase de pós-produção (SYED, 2019).

Tendo em vista a descrição de CQ, procedimentos, práticas e recursos podem ser estipulados para definir a sua utilização de forma mais ampla, o que incluem (BURNSTEIN, 2003):

- políticas e padrões;
- programas de treinamento;
- programas de medição;
- processo de planejamento de testes;
- uso efetivo de técnicas e ferramentas de teste;
- monitoramento do processo e controle de sistema;
- sistema de gestão de configuração;
- prevenção de erros;
- detecção de erros.

A equipe de Controle de Qualidade em geral pode ser definida em três papéis: Gerente, Testador e Clientes. Os Gerentes devem garantir que atividades, políticas e plano de teste incluam problemas relacionados à CQ, alocando recursos e treinamento para a equipe (BURNSTEIN, 2003), determinando ainda o tamanho do time de acordo com o escopo do projeto e criação do ambiente de testes (SYED, 2019).

O time de testadores continua o seu papel como avaliadores de qualidade, e junto com os Gerentes devem garantir que os objetivos de qualidade estejam claramente expressos e alinhados com os documentos de requisitos (BURNSTEIN, 2003). Testadores devem receber treinamentos e especificações para realizarem as seguintes atividades: desenvolver planos de teste (dependendo da técnica utilizada), projetar e executar testes para medir vários critérios de qualidade como usabilidade, disponibilidade, confiança, monitorar, analisar e reportar resultados de testes, entre outras funções (BURNSTEIN, 2003). Usuários possuem a responsabilidade de estabelecer e identificar requisitos de qualidade nos projetos que estão envolvidos. Além disso, usuários devem auxiliar em cenários de uso comuns, comunicando suas opiniões a respeito das vantagens e fraquezas do *software* sendo produzido (BURNSTEIN, 2003).

A etapa de Controle de Qualidade em jogos digitais é diferente para cada projeto e empresa, onde o processo em si geralmente está relacionado com as necessidades do usuário (RUUSKA, 2015). Além de detectar e prevenir erros, os times de CQ proveem feedback inicial sobre o projeto, arte, interface do usuário, mecânicas internas, necessitando de profissionais capacitados e altamente técnicos (MCV, 2019). Muitos dos testes realizados em jogos digitais nesta etapa demonstram similaridades com as técnicas já praticadas no desenvolvimento de *software* em geral (CHO, 2022). Técnicas como verificação da *build* interna (versão do sistema), testes de integração, regressão, *smoke*, *stress*, unitários apresentam aspectos adaptáveis ao contexto de jogos digitais.

Apesar disso, as práticas em geral utilizadas em engenharia de *software* não são condizentes no desenvolvimento de jogos, onde o objetivo de se criar um jogo digital é o lazer e entretenimento, no qual é altamente subjetivo e difícil de gerenciar os requisitos envolvidos (MURPHY-HILL; ZIMMERMANN; NAGAPPAN, 2014). Além disso, entre desenvolvedores de jogos digitais, percebe-se uma baixa utilização de aspectos como reusabilidade de código, ferramentas entre projetos, artefatos de automatização (devido ao alto custo de manutenção) e também notou-se que eles possuem fragilidade às mudanças frequentes, (MURPHY-HILL; ZIMMERMANN; NAGAPPAN, 2014). Politowski, Petrillo e Guéhéneuc (2021), apresentam uma revisão da literatura em estudos acadêmicos e na literatura cinza (sites na internet, blogs, etc.), e identificaram que os testes realizados em jogos digitais focam na experiência do usuário, em vez da precisão funcional dos requisitos. Percebe-se também que há um foco insuficiente na realização de testes em geral, onde a configuração de ferramentas de teste e sua utilização ainda não está difundida entre as equipes e gestores de projeto. Além disso, outro fator relevante

que impacta negativamente a etapa de controle de qualidade é a alocação de recursos pelas empresas (RACHEVA, 2019). Ao dividi-las baseado no número de funcionários e no orçamento do projeto, ambas empresas independentes (do inglês, *indies*) e *Triple A* (AAA) possuem abordagens diferentes em relação à alocação desses recursos na etapa de testes (RACHEVA, 2019).

Empresas *indies* se caracterizam por serem tanto independentes, no sentido de não pertencerem a outra empresa, quanto na natureza de serem relativamente pequenas e autofinanciadas (KARTHIKEYAN, 2022). Na indústria de jogos, as equipes são geralmente formadas por menos de dez pessoas e na maioria dos casos, dependem de doações ou projetos em sites de *crowdfunding*, como o *Kickstarter*, onde as pessoas interessadas no projeto o apoiam com valores em dinheiro diversos para tanto ajudar o projeto a bater a meta especificada quanto ganhar recompensas exclusivas, para que obtenham um nível financeiro estável. Exemplos de jogos *indies*: *Minecraft*, *Dragon Fable* e *Papers Please*. Já as empresas AAA desenvolvem jogos *blockbuster*, ou seja, jogos com grande impacto na comunidade de jogadores, possuindo um alto sucesso comercial e financeiro (KARTHIKEYAN, 2022). Essas empresas possuem grande alocação de recursos econômicos e humanos, geralmente trabalhando com mais de um escritório ou distribuidores/estúdios. Exemplos de jogos AAA: *Grand Theft Auto*, *DOOM* e *The Last of Us*.

Tendo isso em vista, é importante a noção e entendimento da área de CQ para que se torne parte fundamental do ciclo de desenvolvimento de jogos digitais.

2.2.1 Práticas Aplicáveis de Controle de Qualidade em Jogos Digitais

De acordo com a literatura, muitas práticas de CQ estão sendo utilizadas pelas empresas para testar seus jogos. Entretanto, ainda não há uma noção plena da real vantagem de investir nessa metodologia, onde uma vasta maioria de testadores utiliza testes ad hoc para verificar *features*, jogabilidade, entre outros aspectos. Isso ocorre por alguns motivos: falta de recursos, ferramentas apropriadas e automação, onde empresas independentes são as mais afetadas.

As técnicas de CQ variam de acordo com o sistema que está sendo testado e seu contexto. Técnicas bem conhecidas como teste de verificação, integração, regressão, *smoke*, *stress*, unidade e experiência do usuário *User Experience (UX)* tendem a ser as mais utilizadas na indústria (BOURQUE; FAIRLEY, 2014), porém, não são as únicas e não devem ser consideradas como “bala de prata”, uma vez que uma técnica específica pode não solucionar o problema encontrado durante o processo de CQ.

As principais técnicas de CQ encontradas na literatura são as seguintes: **Teste de Funcionalidade**, **Teste Unitário**, **Teste de Integração**, **Teste de Progressão**, **Teste *Smoke***, **Teste Exploratório**, **Teste *Cleanroom***, **Teste A/B**, **Teste com Grupos Focais**, ***Playtesting***, **Teste de Performance**, **Teste de Aceitação**, **Testes de Regressão** e **Testes Automatizados**.

Testes de Funcionalidades (ou Testes Funcionais), no nível de sistema, são usados para garantir que o comportamento do sistema é compatível com a especificação de requisitos (BURNSTEIN, 2003). Em jogos digitais, testes funcionais servem para que o testador identifique qualquer problema com a jogabilidade que possa afetar a experiência do jogador, o que inclui performance, estabilidade, *features*, etc. (SYED, 2019)

O foco principal dessa técnica são as entradas e saídas apropriadas para cada função. Entradas impróprias e ilegais devem ser tratadas pelo sistema, onde caso esse comportamento ocorra, o resultado deve ser observado e documentado para ser corrigido pelo time de desenvolvimento. Ao documentar o defeito encontrado, os testadores devem ser aptos a prover as ferramentas e passo a passo para reprodução e correção dos problemas. Um bom testador de funcionalidade (TF) deve possuir atenção aos detalhes, vontade de “quebrar” o jogo e pensar a frente dos desenvolvedores (RUUSKA, 2015). Os melhores TF são jogadores competentes com anos de experiência na área de jogos digitais, além de possuírem excelente *soft skills*, como ótima comunicação, trabalho em equipe e habilidade de trabalhar de forma autônoma.

A vantagem mais clara de se fazer testes funcionais gira em torno do fato de essa verificação ser a mais próxima da experiência final que o usuário terá ao jogar, uma vez que os testadores mapeiam boa parte dos possíveis eventos e cenários que possam gerar *bugs*, semelhantes aos que o jogador tem contato, promovendo assim, um produto com um número mínimo de falhas no mercado.

Testes Unitários (ou Testes de Unidade) são feitos pelos desenvolvedores para garantir que o comportamento desejado de uma seção específica da aplicação seja atendido (NEGRÃO, 2020). Boas práticas de teste remetem à testes unitários que são planejados, incluindo testes para revelar defeitos como descrição funcional, erros no código, de dados, lógica de controle e de sequência (BURNSTEIN, 2003). A unidade a ser testada em questão deve ser validada por um testador independente, ou seja, alguém que não teve contato direto com o código durante a sua implementação.

Essa técnica possui ótimo desempenho para identificar problemas o mais cedo possível no ciclo de vida de desenvolvimento (NEGRÃO, 2020). Caso o teste seja elaborado de forma útil e robusta, no momento que houver mudanças no código será mais prático de detectar erros introduzidos, melhorando o aspecto de manutenção geral e tempo necessário para sua execução.

No momento em que vários testes unitários são realizados e aprovados, eles são combinados para serem testados em grupos (JIMENEZ et al., 2018), e então começa a etapa de Testes de Integração. O objetivo do teste de integração é expôr os defeitos das interações entre as unidades integradas realizadas previamente nos testes unitários (NEGRÃO, 2020). Cada módulo testado anteriormente é adicionado um de cada vez, até o momento em que todos estão integrados. Isso evita que grandes falhas ocorram quando várias unidades são testadas ao mesmo tempo, em conjunto, e auxilia o testador a manter

um ambiente de teste relativamente pequeno o suficiente para obter um foco necessário para análise (BURNSTEIN, 2003).

Testes de Progressão baseiam-se no contexto de testar determinadas pequenas funcionalidades ou aspectos do jogo do início ao fim (missões, mapas, *puzzles*, etc.) (SYED, 2019). O objetivo é completar cenários de jogo em que haja qualquer tipo progresso, evitando encontrar erros que atrapalhem o mesmo, impossibilitando a continuidade da progressão do jogo (ou o termo utilizado em inglês, *dead-end*) ou que façam o jogo parar de funcionar repentinamente (RAMADAN; HENDRADJAYA, 2014). O jogador deve sentir a sensação de constante avanço, onde a progressão deve se manter em um nível bom e consistente, evitando ficar parado em algum mapa, objetivo, etc. (LAHTI et al., 2015).

Teste *Smoke*, (em português, teste de fumaça), tem o objetivo de testar e revelar erros em um sistema já finalizado, verificando as funcionalidades mais básicas (MOZGOVOY; PYSHKIN, 2018). Dependendo do jogo, algumas rotinas podem ser automatizadas para testar funcionalidades básicas, mas que sejam executadas muitas vezes, como criar uma nova conta e completar o tutorial, trocar de vestimentas, realizar *upgrade* em armas, etc. (SYED, 2019). Assim, permitindo que esses cenários de teste fossem excluídos dos testes manuais, possibilitando foco e maior tempo disponível para outras tarefas.

Testes Exploratórios possuem o foco de permitir aos testadores que utilizem a aplicação que está sendo testada da forma que quiserem, explorando o jogo sem nenhuma restrição (LAHTI et al., 2015). Entretanto, casos de teste, documentação de teste e resultados devem ser incluídos. Whittaker (2009) colabora afirmando que ferramentas de captura de tela devem ser usadas para gravar os resultados deste tipo de teste. É vantajoso usar esta técnica quando o projeto não possui ou não especifica casos de teste ou passos e instruções a serem seguidos previamente.

Teste *Cleanroom*, (em português, sala limpa), possui como fundamento produzir testes no qual irão emular o comportamento típico de como os usuários jogam (SCHULTZ; BRYANT, 2016). O time de desenvolvimento investiga os diferentes tipos e modos de jogabilidade que seus jogadores possuem e capturam essas informações para encontrar erros onde uma abordagem típica de testes de funcionalidades não seria capaz de verificar (LAHTI et al., 2015).

Testes A/B possuem a abordagem que gira em torno da divisão de usuários de um *software* em dois grupos com versões diferentes do sistema (LAHTI et al., 2015). Os usuários não podem testar ambas versões e ao final da realização dos testes, os resultados são comparados e avaliados entre as versões, de acordo com o feedback fornecido. Testes A/B geralmente são realizados para maximizar comportamentos específicos de jogadores (hábitos de gastos, jogabilidade, retenção), testar funcionalidades novas e já existentes para otimizar a performance e taxas de aderência dos usuários, além de melhorar o fluxo dos usuários (progressão de níveis, ritmo de recompensas oferecidas, etc) (UNITY, 2022)

Teste com Grupos Focais se baseiam no recrutamento de potenciais usuários para

testar o jogo em uma sessão organizada pela equipe de desenvolvimento ou pelo time de CQ (LAHTI et al., 2015). As informações provenientes das sessões podem ser coletadas simplesmente ao tomar notas dos fatos ocorridos, analisando respostas de questionários ou gravando as sessões. Como vantagem, participantes podem mencionar e notar imperfeições que são esquecidas pelo time de desenvolvimento (LAHTI et al., 2015). As sessões ainda fornecem a ideia do estado atual da jogabilidade e quão desafiador estão os diferentes *assets* do jogo.

Playtesting, considerada a técnica mais importante que um testador irá utilizar, é a atividade que a equipe de testadores e de projeto irão praticar durante toda a elaboração de sistema com o objetivo de ganhar discernimento em quais aspectos o jogo está ou não alcançando os objetivos de experiência do usuário (FULLERTON, 2014). Métricas como eficiência, performance, dificuldade, níveis de aborrecimento e empolgação são analisadas pelos testadores em um documento detalhado, promovendo as informações mais importantes para que hajam melhorias no futuro, geralmente aplicando ferramentas de análise de dados para identificar o comportamento dos jogadores (NEGRÃO, 2020; NETO et al., 2019).

Em relação às diferentes abordagens de *playtesting*, Fullerton (2014) enuncia algumas práticas que funcionam de maneira efetiva, especialmente em dinâmicas de grupo, gerando ideias para o sistema e informações da jogabilidade, mas é relativamente ruim para medir as mesmas. A abordagem irá depender do contexto em que se há internamente da empresa e do jogo em si. Abordagens como testes um-para-um (o responsável pelos testes observa o testador, anotando informações úteis), testes em grupos, formulários de feedback, entrevistas, discussões abertas, gancho de dados (observar onde os usuários ficaram mais tempo presos ou com problemas), etc.

Ao se analisar Testes de Performance, é importante levar em consideração dois grandes aspectos em relação aos requisitos: requisitos funcionais, que descrevem quais funções o *software* deve realizar e qual o nível de conformidade desses requisitos com os testes de sistema; e requisitos de qualidade, que em tese são caracterizados como não-funcionais, mas descrevem níveis de qualidade esperados do *software*, onde usuários podem possuir expectativas a respeito do uso de memória, tempo de resposta, atrasos e taxas de transferências (NETO et al., 2019).

O objetivo dos testes de performance é verificar se o *software* atende aos requisitos de performance. A equipe de testadores pode “sintonizar” o produto, ou seja, otimizar a alocação de recursos do sistema (NETO et al., 2019). Os resultados dos testes devem ser quantificados claramente, especificando a medida de performance utilizada (número de ciclos da CPU utilizada, tempo de resposta em segundos, minutos), número de transações processadas por determinado período de tempo, etc.

Em jogos digitais, parâmetros importantes devem ser cuidadosamente levados em consideração, como tempo de resposta entre servidores e jogadores, tempo de transa-

ções, cobertura de rede, escalabilidade do sistema, restrições de memória e processadores, consumo de bateria e performance gráfica (HAMILTON, 2022).

A realização do Teste de Aceitação é um marco importante para os desenvolvedores, pois é nessa etapa que os jogadores irão determinar se o produto implementado está de acordo com seus requisitos (BURNSTEIN, 2003). Obrigações contratuais podem ser assumidas caso o cliente e/ou empresa publicadora de jogos esteja satisfeita com o *software* final.

Alguns pontos devem ser atendidos previamente antes da execução dos testes de aceitação. Aspectos como requisitos de negócio do produto devem ter sido atendidos, código-fonte finalizado, atividades de CQ completadas, *bugs* e defeitos solucionados e ambiente de aceitação pelo usuário deve ser criado (ALTEXSOFT, 2021).

Testes de Regressão são realizados para verificar se a qualidade do jogo ainda permanece em um bom nível mesmo após várias mudanças terem sido feitas, adicionando funcionalidades novas ou alterações da estrutura interna (ALEEM; CAPRETZ; AHMED, 2016a). A necessidade deste tipo de teste vem do fato que antigos defeitos e erros podem ser reintroduzidos no código e gerar novos erros (NOVAK, 2011). Testes de regressão eficientes irão minimizar o número de testes realizados, mas ainda serão feitos para novos defeitos introduzidos ou que ainda restaram (SCHULTZ; BRYANT, 2016). Geralmente, os testes são refeitos para analisar as diferentes *builds* (em português, versões do jogo) que apresentaram erros e que falharam nos critérios de avaliação (LAHTI et al., 2015). Para haver conformidade, deve-se rodar uma bateria de testes de forma cíclica para garantir que as mudanças continuam funcionando como o esperado (SCHULTZ; BRYANT, 2016).

Devido a complexidade de interações de usuários em jogos digitais ultimamente, testes em geral estão altamente dependentes de testadores humanos (ZHENG et al., 2019). Os fatores de qualidade e experiência do usuário chamam a atenção de jogadores, criando comunidades imensas e altamente interessadas por produtos de software de excelência técnica (ALBAGHAJATI; AHMED, 2020). Tendo isso em vista, com uma demanda maior por jogos mais sofisticados, testes automatizados se tornaram altamente adotados pelas empresas para reduzir os custos de produção e melhorar a qualidade interna (BERTOLINO, 2007).

Como a automatização de testes pode ser considerada uma grande área de teste, englobando e automatizando outras técnicas já discutidas nesta seção, deve-se considerar aspectos como cronograma de desenvolvimento, criação de testes, tempo de execução para a escolha do melhor método de teste automatizado (NILSSON; NILSSON, 2021).

2.3 Percepções e Feedback de Usuários de Jogos Digitais

Lidar com as expectativas de jogadores em relação a determinado jogo sendo implementado, deve ser levado em consideração pelo time de desenvolvimento. Jogos que conseguiram entregar por completo as funcionalidades e interações prometidas, se prova-

ram grandes exemplos do sucesso de um produto (BETHKE, 2003). Garantir a satisfação do consumidor (neste caso, o jogador), é um aspecto crucial do desenvolvimento (FABRICATORE; NUSSBAUM; ROSAS, 2002). Para obter discernimento a respeito das preferências dos jogos que o usuário deseja jogar, é preciso ter noção dos desejos de jogabilidade e decisões de compra. O ponto chave para tal compreensão se inicia no entendimento do público alvo ao qual o jogo será destinado (FULLERTON, 2014), traçando um perfil de características do usuário e da comunidade. Este conhecimento permite aos profissionais do time de projeto e de desenvolvimento uma busca de qualidade aperfeiçoada em seus produtos, tendo em vista a relação entre pessoas e sistemas computacionais, para que haja a possibilidade de construí-las, melhorá-las e inseri-las no meio em que os usuários convivem, buscando a melhor experiência possível (SOUSA et al., 2021).

Fullerton (2014) afirma que cinco fatores de qualidade devem ser atendidos pelos jogos digitais:

- **Funcional:** indicativo que as funcionalidades do jogo estão aptas para serem utilizadas e operadas, tendo sua garantia verificada na etapa de testes;
- **Internamente completo:** fator que indica se todas as regras, condições e seções estão funcionando internamente, governada pela completude do jogo;
- **Balanceado:** indica a dificuldade do jogo em si (se está ajustada ou se é muito difícil ou fácil);
- **Divertido:** indica se o jogo é desafiador, cativante, divertido, etc. Medido por questionários ou feedback direto de usuários;
- **Acessível:** significa que o jogo é fácil de entender, aprender e se é intuitivo o suficiente ao jogador. É medido através da capacidade de se orientar e obter o controle do jogo e da curva de aprendizado da interface de usuário.

Os fatores de qualidade que são mais considerados pelos jogadores ao passarem pela experiência de um jogo digital são os fatores de usabilidade e de satisfação (ALEEM; CAPRETZ; AHMED, 2016b). O fator de usabilidade é relacionado a experiência do usuário, contendo elementos que podem afetar a mesma, como intuitividade, consistência e compreensibilidade. O fator de satisfação gira em volta do engajamento do jogador durante sua jogatina, e apesar de ser um item subjetivo, é preciso ser medido através de um questionário de satisfação (RAMADAN; HENDRADJAYA, 2014).

2.4 Aspectos de Usabilidade em Jogos Digitais

A usabilidade é uma métrica utilizada para medir o quão bem um usuário específico, dentro de um determinado contexto, pode utilizar um produto para alcançar um objetivo pré-determinado de forma efetiva, eficiente e satisfatória (SINGH, 2022). Em

jogos digitais, a usabilidade faz referência ao nível que o jogador pode aprender, controlar e entender um jogo. Considerando que um dos principais objetivos de um jogo digital é o entretenimento e engajamento de usuários, fica claro que os desenvolvedores devem prestar atenção em aspectos como aspectos de design, história a ser narrada, nível de dificuldade e mecânicas internas (PINELLE; WONG; STACH, 2008).

Os desenvolvedores de jogos digitais planejam e executam testes para entender o quão fácil é o *design* do jogo para um grupo de usuários, geralmente completando tarefas de vários tipos, realizando-as de forma repetitiva até que haja a compreensão e análise do produto (SINGH, 2022). Os problemas identificados são encontrados por meio de testes realizados por meio de várias técnicas, como *playtesting* ou grupos focais, onde existe similaridades de usabilidade dependendo do gênero do jogo (PINELLE; WONG; STACH, 2008). Um possível exemplo de similaridade se encontra em jogos do gênero de estratégia, no qual possuem esquemas de navegação e mapeamento de entradas, ou seja, teclas de atalho similares, componentes de *interface* idênticos e mapeamento de teclas do *mouse* equivalentes. Uma vez que há a definição de um gênero e já que existem similaridades entre jogos que compartilham deste gênero, abordagens adaptadas podem ser utilizadas para testar os jogos.

Durante a aplicação da pesquisa com profissionais da área de teste de *software* e/ou controle de qualidade neste estudo, foram utilizados cinco aspectos de usabilidade que podem ser aplicados em jogos digitais, tendo como base a ISO 25001, que são: Conformidade, Operabilidade, Aprendibilidade, Eficiência e Flexibilidade.

A Conformidade gira em torno da capacidade do produto de *software* em aderir padrões, convenções, guias de estilo ou regulações referentes a usabilidade (COMMISSION et al., 2001).

A Operabilidade remete a capacidade do produto de *software* que permite ao usuário operá-lo e controlá-lo da maneira que for desejável (COMMISSION et al., 2001). Bevan et al. (2016) expõe algumas medidas presentes no quesito operabilidade, que envolvem desde consistência operacional, clareza de mensagens, customizabilidade funcional e de interface, capacidade de monitoramento, categorização compreensível de informação, aparência consistente e suporte para dispositivo de entrada.

A Aprendibilidade possui como característica a capacidade do produto de *software* em permitir que o usuário aprenda e entenda a utilização da aplicação em questão (COMMISSION et al., 2001). Como é um atributo de usabilidade fundamental, uma vez que um sistema deve possuir fácil entendimento, o usuário deve ser apto a utilizá-lo de maneira mais rápida possível, tendo em vista o propósito da aplicação e o público alvo (SIPPOLA, 2017).

A Eficiência aborda sobre a capacidade do produto de *software* em prover uma performance apropriada ao contexto, relativa a quantidade de recursos utilizados, dentro de certas condições (COMMISSION et al., 2001). Além disso, diz respeito ao quão rápido

um usuário se adapta naquele sistema para realizar as tarefas, gerando uma boa adaptação. Uma forma de aumentar a eficiência é adicionar atalhos para funções que sejam realizadas frequentemente (SIPPOLA, 2017).

A Flexibilidade é a capacidade do produto de *software* em fornecer um conjunto de funções para tarefas específicas e objetivos de usuários que possam acontecer em momentos diversos (COMMISSION et al., 2001). É um atributo que permite produtos a levarem em consideração circunstâncias, oportunidades e preferências individuais que não tenham sido previstos com antecedência, bem como a capacidade do produto de ser modificado e prover adaptação de suporte à novos tipos de usuários, tarefas ou ambiente de sistema (BEVAN et al., 2016). Estas circunstâncias podem ser medidas pela análise das características do produto e o contexto de uso no qual pode ser usado por diferentes usuários.

A principal vantagem de se basear neste modelo, tendo em vista o suporte às medidas apropriadas, é a clareza na definição de usabilidade e a proposta de medidas para prover evidência sobre um objetivo que se queira alcançar. Apesar disso, o modelo da ISO 25001 apresenta algumas fraquezas, como curva de aprendizado complexa, uso intensivo de recursos para coletar dados necessários e falta de familiaridade, uma vez que, por exemplo, a ISO 9126 é mais amplamente usada ou conhecida (ABRAN et al., 2003).

2.5 Árvores de Decisão

Árvores de Decisão são modelos sequenciais no qual combinam uma sequência de testes simples: cada teste compara um atributo numérico em relação a um limite (do termo em inglês, *threshold*), ou um atributo nominal em relação a um conjunto de possíveis valores (KOTSIANTIS, 2013). A estrutura visual de uma árvore de decisão remete a um fluxograma, no qual cada nó interno representa um teste em uma *feature*, ou seja, uma característica ou aspecto daquele contexto; cada nó considerado uma folha representa uma classe, ou seja, uma decisão tomada após computar todas as *features*; e os galhos representam conjunções das *features* que levaram ao resultado obtido de uma classe (YADAV, 2018). A Figura 4 apresenta um exemplo adaptado de Kotsiantis (2013) onde há a caracterização de uma árvore de decisão simples, no qual os quadrados azuis representam as *features*, os círculos amarelos (as folhas da árvore em questão) remetem as classes e a informação transmitida entre nós denota a uma instância da *feature* selecionada para testes.

Árvores de decisão podem ser utilizadas para resolver tanto problemas de classificação quanto de regressão, aplicando uma abordagem *top-down* para o *dataset*, ou seja, a fonte de dados utilizada, durante o treinamento do algoritmo (RAJ, 2021), sendo ainda baseados em aprendizado supervisionado (YADAV, 2018). A Tabela 1 define os dados utilizados para a geração da árvore de decisão apresentada na Figura 4 acima.

Para cada interação, o algoritmo considera a partição do conjunto de dados de treinamento usando a saída de uma função dos atributos de entrada (KOTSIANTIS,

Figura 4 – Estrutura padrão de uma árvore de decisão (adaptado)

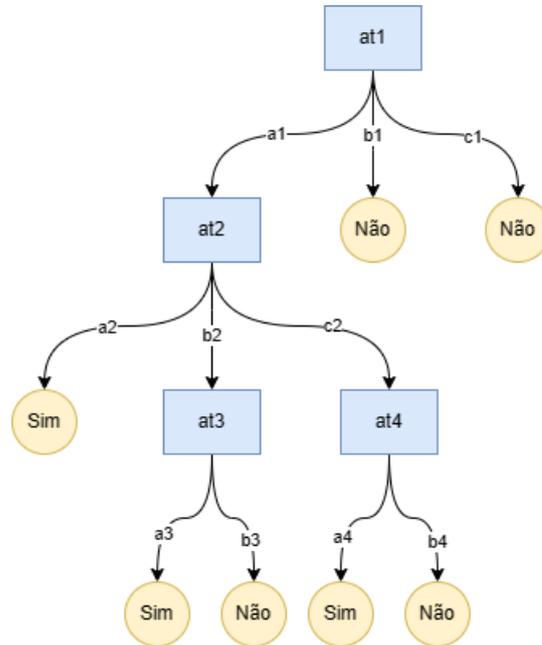


Tabela 1 – Dados de treinamento (adaptado)

at1	at2	at3	at4	Classe
a1	a2	a3	a4	Sim
a1	a2	a3	b4	Sim
a1	b2	a3	b4	Sim
a1	b2	b3	b4	Não
a1	c2	a3	a4	Sim
a1	c2	a3	b4	Não
b1	b2	b3	b4	Não
c1	b2	b3	b4	Não

2013). A seleção da função mais apropriada é feita de acordo com algumas medidas de divisão. Ou seja, o ganho de informação é usado para decidir em qual *feature* haja a divisão em cada passo ao construir a árvore (YADAV, 2018).

Para cada nó da árvore, o valor da informação mede quanta informação uma *feature* fornece sobre uma classe. A divisão com o maior ganho de informação será selecionada como a primeira partição e o processo irá continuar até que todos os nós estejam puros, ou seja, em uma amostragem do *dataset*, toda a informação pertence a mesma classe, ou até que o ganho de informação seja igual a zero (YADAV, 2018). Ao final do treinamento, uma árvore de decisão é gerada e pode ser utilizada para realizar previsões categorizadas otimizadas (RAJ, 2021).

As vantagens de se utilizar árvores de decisão se baseiam em sua eficiência, já que o custo de predição é logarítmico no número de pontos de dados usados para treinar a árvore, desde que os parâmetros estejam otimizados e bem estruturados (PEDREGOSA et al., 2011); lógica de fácil entendimento com baixo custo, uma vez que as árvores po-

dem ser visualizadas; classificação rápida de registros que sejam desconhecidos; além de desconsiderar *features* com pouca ou nenhuma importância para a predição (RAJ, 2021).

Entretanto, como desvantagens, árvores de decisão possuem a tendência de possuírem alguns aspectos não desejáveis, como sobre-ajuste (do inglês, *overfit*), dos dados de treinamento, precisando aplicar mecanismos como a poda das ramificações ou definir um número mínimo de galhos; mudanças desnecessárias do resultado final caso hajam alterações no *dataset*; complexidade de interpretação de árvores muito grandes, onde cada divisão ocorra em *features* que possuam um certo número de níveis ou de características (RAJ, 2021), além de conceitos difíceis de compreender onde o algoritmo de árvore de decisão não os expressam facilmente, como XOR, problemas de multiplexador ou paridade (PEDREGOSA et al., 2011).

2.5.1 Algoritmo de Árvores de Decisão

Como explicado anteriormente, árvores de decisão são métodos supervisionados não-parametrizados usados para problemas de classificação e regressão (PEDREGOSA et al., 2011). O objetivo é criar um modelo que prevê o valor de uma variável definida como classe ao aprender regras simples de decisão inferidas nas *features* do *dataset* elaborado (PEDREGOSA et al., 2011).

Para a elaboração do algoritmo da árvore de decisão, a biblioteca *Scikit Learn* que utiliza a linguagem de programação Python auxilia na criação e entendimento dos conceitos já apresentados. Primeiramente, é preciso realizar a importação das bibliotecas *tree* e *DecisionTreeClassifier* no código, como mostrado no trecho de código abaixo.

```
1 from sklearn import tree
2 from sklearn.tree import DecisionTreeClassifier
```

Após a definição final do *dataset* a ser utilizado, começa a parte de classificação do modelo. O método *DecisionTreeClassifier* recebe dois valores como parâmetros no formato de *array* (em português, uma lista de elementos): primeiramente, um *array* X contendo a quantidade de elementos e as *features* envolvidas, e outro *array* Y contendo as classes para a classificação das *features* de treinamento. Após isso, é criada a instância do modelo que irá suportar a árvore de decisão em si, e o método *fit()* é utilizado, recebendo como parâmetros os valores de X e Y previamente definidos, fazendo assim, a criação da árvore de decisão pelo *dataset* de treinamento. A saída do método é uma estimativa ajustada do modelo (PEDREGOSA et al., 2011). O trecho de código abaixo apresenta como seria a implementação desse conceito, onde o exemplo utiliza a importação de um *dataset* pré-definido pela própria biblioteca chamada *Iris* para gerar o cenário. O conjunto de dados contém 3 classes de 50 instâncias cada, onde cada classe se refere a um tipo de planta íris. Uma classe é linearmente separável das outras 2, no qual os últimos não são linearmente separáveis uns dos outros (FISHER, 1936).

$$Gini = 1 - \sum_{i=1}^j P(i)^2$$

Onde j representa o número de classes na variável alvo (no caso deste estudo, onde será explicado no Capítulo 5, remetem as combinações de técnicas de teste), e $P(i)$ representa a razão de números de combinações de observações do nó. Após o cálculo do índice, o atributo com menor valor no índice *Gini* é escolhido como atributo para divisão para elaboração da árvore de decisão (YADAV, 2018). A divisão acontecerá até que o ganho de informação através do índice *Gini* seja igual a zero, chegando assim, em um nó folha da árvore.

2.6 Lições do Capítulo

Neste Capítulo, buscou-se o entendimento e compreensão das etapas presentes no desenvolvimento de jogos, seus papéis e contribuições. Além disso, identificou-se que a etapa de controle de qualidade possui uma baixa adesão entre as empresas de desenvolvimento, afetando diretamente a qualidade do produto. Práticas de teste podem ser utilizadas dependendo do contexto do jogo, mas devem ser analisadas previamente, uma vez que a prioridade de testes varia de acordo com os recursos disponíveis. Identificou-se também as características de qualidade relevantes aos jogadores após testarem um jogo, que servem de apoio e feedback para as empresas melhorarem e oferecerem funcionalidades de maior excelência. Observou-se os aspectos de usabilidade que são analisados e utilizados em jogos digitais para obter um produto de melhor qualidade e, por fim, buscou-se entender melhor como o algoritmo de árvores de decisão funciona, definindo a sua teoria.

3 REVISÃO DA LITERATURA

Este Capítulo possui como propósito apresentar toda a fundamentação por trás do processo de Revisão da Literatura, abordando em detalhes as técnicas e como os estudos foram selecionados para a escrita deste Trabalho de Conclusão de Curso. A Seção 3.1 aborda resumidamente as Revisões Relacionadas, contendo informações do que está sendo visto na literatura. A Seção 3.2 apresenta o Protocolo de Revisão, expondo critérios e métricas de avaliação. A Seção 3.3 aborda a Estratégia de Análise e Extração de Dados, apresentando quais informações foram utilizadas para responder as Questões de Pesquisa. A Seção 3.4 apresenta o processo e discussão das Questões de Pesquisa.

3.1 Revisões Relacionadas

Dos estudos selecionados, Syed (2019), Lahti et al. (2015), Burnstein (2003) e Schultz e Bryant (2016) tratam de definir o que é a etapa de controle de qualidade e sua relevância ao processo de desenvolvimento de jogos digitais. Vargas et al. (2014) apresenta um mapeamento sistemático sobre qualidade em jogos sérios. Lahti et al. (2015), apresenta e discute em detalhes as técnicas de CQ no desenvolvimento de jogos em jogos da Polônia. Albaghajati e Ahmed (2020), Nilsson e Nilsson (2021), Negrão (2020), Politowski, Guéhéneuc e Petrillo (2022), Schatten et al. (2017) focam na descrição de testes automatizados, utilizando-se em conjunto com outras técnicas. Novak (2011) e Bethke (2003) tratam de abordar o ciclo de vida geral no desenvolvimento de jogos, técnicas de teste na etapa de controle de qualidade, papéis envolvidos no desenvolvimento de jogos digitais e a visão teórica dos possíveis gêneros de jogos.

Com os resultados encontrados, percebe-se a falta de um entendimento e aplicação de técnicas de teste na etapa de controle de qualidade, onde muitas vezes são ignoradas ou utilizadas por muito pouco tempo pelos desenvolvedores durante o processo de desenvolvimento, além da falta de discernimento sobre quais técnicas e processos utilizar.

3.2 Protocolo de Revisão

O presente Protocolo de Revisão tem como objetivo planejar, mapear e responder as Questões de Pesquisa que servirão de base para este Trabalho de Conclusão de Curso. Seguindo o *framework* PICO (*Population-Intervention-Comparison-Outcome*), foram definidas duas Questões de Pesquisa, com seus respectivos itens à serem analisados, afim de conduzir e validar o estudo proposto.

3.2.1 Questões de Pesquisa

- **Q1.** Quais as técnicas do processo de controle de qualidade que estão sendo aplicadas/são aplicáveis em relação ao desenvolvimento de jogos digitais?

- **Population:** qualidade
 - **Intervention:** processo de controle de qualidade
 - **Comparison:** desenvolvimento de jogos digitais
 - **Outcomes:** principais técnicas
- **Q2.** Quais são as contribuições efetivas das técnicas de controle de qualidade para o processo de desenvolvimento de jogos digitais, contribuindo com valor agregado ao produto de software final?
 - **Population:** qualidade
 - **Intervention:** técnicas de controle de qualidade
 - **Comparison:** desenvolvimento de jogos digitais
 - **Outcomes:** contribuições efetivas

3.2.2 Processo de Busca

Inicialmente no processo de busca, elaborou-se uma *string* de busca genérica de acordo com as Questões de Pesquisa a serem respondidas. Após isso, encontrou-se as sementes do trabalho, ou seja, artigos e estudos que serviram como base para a elaboração e entendimento do estado da arte. A partir disso, utilizou-se da técnica de *snowballing* para encontrar outros estudos relevantes a partir das sementes encontradas, aplicando tanto a etapa de *backward* quanto a etapa de *forward* através das referências encontradas nas sementes e nos trabalhos que mencionam as mesmas. A Tabela 2 apresenta os termos e sinônimos utilizados para criação da *string* de busca.

3.2.2.1 String de Busca

Tabela 2 – Termos de busca

Termo	Sinônimo
quality control	quality assurance
games	gaming, gaming industry, digital games

String Genérica: “quality control” OR “quality assurance” AND “games” OR gaming OR “video games” OR “digital games”

3.2.2.2 Informações de Busca

Logo após a criação da *string* de busca, iniciou-se o processo de busca em duas bases de dados: A SCOPUS e Google Scholar. A Tabela 3 apresenta as bases, as suas respectivas *strings*, filtros utilizados e a quantidade de materiais obtidos na busca.

Percebe-se a quantidade exagerada de resultados obtidos através da base Google Scholar. Isso se dá ao fato de que o motor de busca não é muito eficiente comparado ao da SCOPUS, e que as opções avançadas de filtros não oferecem muitas opções.

Tabela 3 – Informações de busca

Base de Dados	String na Database	Filtro	Quantidade
SCOPUS	((“quality control” OR “quality assurance”) AND (games OR gaming OR “gaming industry” OR “digital games”))	TITLE-ABS-KEY	226
Google Scholar	“quality control” OR “quality assurance” AND games OR gaming OR “gaming industry” OR “digital games”	All	> 17.000 (Cerca de 130 estudos foram analisados)

3.2.3 Processo de Seleção

A partir dos resultados obtidos das bases de dados, iniciou-se o Processo de Seleção dos estudos que auxiliaram a escrever este Trabalho de Conclusão de Curso. Foram aplicados em todos os estudos que retornam das bases critérios, tanto de Inclusão (Tabela 4), Exclusão (Tabela 5) e de Qualidade (Tabela 6). Em cada um dos critérios foram definidos números de identificação e descrições. Por fim, após os critérios terem sido aplicados, os estudos que foram aceitos sofreram uma filtragem maior, sendo aplicados às métricas de avaliação que correspondem as Questões de Pesquisa. A Tabela 7 apresenta todas as métricas utilizadas.

Tabela 4 – Critérios de Inclusão

ID	Descrição
IC-001	Estudos primários que apresentem no resumo alguma menção ao controle de qualidade no desenvolvimento de jogos.

Tabela 5 – Critérios de Exclusão

ID	Descrição
EC-001	Estudos que não tratam sobre controle de qualidade
EC-002	Estudos duplicados
EC-003	Estudos com menos de 6 páginas
EC-004	Estudos não escritos em inglês ou português
EC-005	Íntegra do estudo não disponível

Tabela 6 – Critérios de Qualidade

Score	Descrição
1.0	Y - Aceito
0.5	P - Parcialmente
0	N - Não

3.2.3.1 Métricas dos Estudos Analisados

O método e as métricas utilizadas para a seleção dos estudos estão descritas na Tabela 7, identificando-se os detalhes à serem explorados.

Tabela 7 – Métricas dos estudos analisados

ID	Peso	Descrição	Métricas
QC1 - Q1	0.6	O estudo deve apresentar as técnicas aplicáveis em detalhes.	<p>Y: O estudo apresenta e explica em detalhes as técnicas encontradas.</p> <p>P: O estudo apresenta, mas não detalha as técnicas encontradas.</p> <p>N: O estudo não menciona ou detalha quaisquer técnicas.</p>
QC2 - Q2	0.4	O estudo deve apresentar as contribuições efetivas das técnicas de CQ.	<p>Y: O estudo apresenta e explica em detalhes as contribuições.</p> <p>P: O estudo apresenta contribuições, mas que não foram aplicadas na prática.</p> <p>N: O estudo não apresenta ou detalha quaisquer contribuições.</p>

3.3 Estratégia de Análise e Extração de Dados

Ao longo da elaboração desta pesquisa, houve a necessidade da aplicação do processo de Análise de Dados, investigando cada artigo selecionado na etapa de Mapeamento Sistemático e aplicando suas etapas de categorização. Apesar disso, surgiram complicações de cronograma e adaptações ao protocolo, o que levou no ajuste dos tipos de informações e dados a serem coletados para responderem às Questões de Pesquisa. A Tabela 8 apresenta as informações bibliométricas gerais utilizadas.

3.4 Discussão das Questões de Pesquisa

Neste capítulo, serão abordados quais dados são relevantes para responder ambas Questões de Pesquisa, tendo em vista a estratégia de extração de dados dos estudos selecionados e sua análise. A listagem dos estudos e as informações utilizadas estão divididas e descritas na Tabela 9.

Tabela 8 – Dados para extração de dados

Descrição	Tipo	Questão de Pesquisa
Título	Text	-
Autor	Text	-
Ano	Date	-
Resumo	Text	-
Tipo do Documento	Text	-
Sentença que indique técnicas de CQ aplicadas ao desenvolvimento de jogos	Text	Q1
Parágrafo que indique técnicas de CQ aplicadas ao desenvolvimento de jogos	Text	Q1
Sentença que indique contribuições efetivas das técnicas de CQ	Text	Q2
Parágrafo que indique contribuições efetivas das técnicas de CQ	Text	Q2

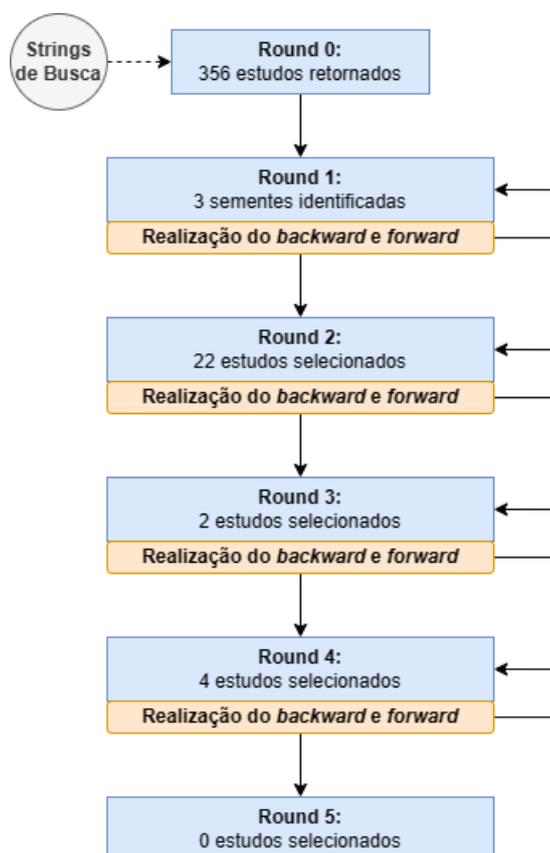
Tabela 9 – Informações dos estudos selecionados.

ID	Nome do estudo	Autor e Ano
01	Game Testing in Finnish Game Companies	(LAHTI et al., 2015)
02	Is there time for Software Testing in the Indie Games Development?	(NETO et al., 2019)
03	Quality Control in Games Development	(SYED, 2019)
04	Game Development and Production	(BETHKE, 2003)
05	Game Development Essentials an Introduction 3rd Edition	(NOVAK, 2011)
06	Game Testing All in One	(SCHULTZ; BRYANT, 2016)
07	Critical success factors to improve the game development process from a developer's perspective	(ALEEM; CAPRETZ; AHMED, 2016a)
08	Game development software engineering process life cycle: a systematic review	(ALEEM; CAPRETZ; AHMED, 2016b)
09	Playtesting with a Purpose	(CHOI et al., 2016)
10	What do game developers test in their products?	(KASURINEN; SMOLANDER, 2014)
11	Comparison of playtesting and expert review methods in mobile game evaluation	(KORHONEN, 2010)
12	Playtesting for indie studios	(MIRZA-BABAEI; MOOSAJEE; DRENIKOW, 2016)

13	Cowboys, Ankle Sprains, and Keepers of Quality: How is Video Game Development Different from Software Development?	(MURPHY-HILL; ZIMMERMANN; NAGAPPAN, 2014)
14	Practical software testing a process-oriented approach	(BURNSTEIN, 2003)
15	GBGallery: A benchmark and framework for game testing	(LI et al., 2022)
17	Development of Game Testing Method for Measuring Game Quality	(RAMADAN; HENDRADJAYA, 2014)
18	Automated testing and validation of computer graphics implementations for cross-plataforms game development	(HOOPER, 2017)
19	Game Development and Testing in Education	(PIETRIKOVÁ; SOBOTA, 2022)
20	A Postmortem on Playtesting: Exploring the Impact of Playtesting on the Critical Reception of Video Games	(MIRZA-BABAEI et al., 2020)
21	A Survey of Game Usability Practices in North American Game Companies	(RAJANEN; TAPANI, 2018)
22	Testing practices in indie game development from a software engineering perspective: an exploratory study	(RACHEVA, 2019)
23	Automated playtesting in videogames	(NEGRÃO, 2020)
24	A Comprehensive Approach to Quality Assurance in a Mobile Game Project	(MOZGOVOY; PYSHKIN, 2018)
25	Bughunting on a Budget Exploring Quality Assurance Practices and Tools for Indie Game Developers	(CHO, 2022)
26	Testing Methods for Mobile Game Development	(ANNANPERÄ et al., 2018)
27	Game Development Life Cycle Guidelines	(RAMADAN; WIDYANI, 2013)
28	Video Game Automated Testing Approaches: An Assessment Framework	(ALBAGHAJATI; AHMED, 2020)
29	A co-evolutionary genetic algorithms approach to detect video game bugs	(ALBAGHAJATI; AHMED, 2022)
30	Comparing Automated Testing Approaches for FPS Games	(NILSSON; NILSSON, 2021)
31	Towards an Agent-Based Automated Testing Environment for Massively Multi-Player Role Playing Games	(SCHATTEN et al., 2017)
32	Towards Automated Video Game Testing: Still a Long Way to Go	(POLITOWSKI; GUÉHÉNEUC; PETRILLO, 2022)

Utilizando um método ad hoc, buscou-se no acervo de publicações de conteúdo científico da Scopus e Google Scholar estudos científicos no intuito de descobrir o estado da arte a respeito do controle de qualidade em jogos digitais e suas técnicas utilizadas na indústria. Cerca de 356 trabalhos e estudos foram analisados, e após a aplicação do processo de *snowballing*, aplicando as etapas de *backward* e *forward* em 5 rodadas para identificar novos trabalhos, foram selecionados 33 estudos que serviram de base para a escrita deste Trabalho de Conclusão de Curso. A Figura 6 apresenta o processo em detalhes.

Figura 6 – Processo de seleção dos estudos



3.4.1 Resposta da Questão 1

Nesta seção, serão discutidas quais foram as informações utilizadas para responder a Questão de Pesquisa 01: **Quais as técnicas do processo de controle de qualidade que estão sendo aplicadas/são aplicáveis em relação ao desenvolvimento de jogos digitais?**. Inicialmente, serão analisadas sentenças e parágrafos que dissertem sobre tais técnicas sendo empregadas na indústria de jogos, explicando resumidamente o objetivo proposto em cada estudo selecionado e suas informações.

Syed (2019) aborda sobre uma visão geral de controle de qualidade em jogos, os papéis envolvidos e as fases no ciclo de vida de desenvolvimento. Além disso, tratou-se de

caracterizar algumas técnicas de teste sendo aplicadas na indústria, tendo em vista os resultados de um *survey* realizado com profissionais, como Testes Unitários, Funcionalidade, Progressão, a fim de apresentar os objetivos das mesmas.

Lahti et al. (2015) executou uma pesquisa em empresas de desenvolvimento de jogos digitais na Finlândia e foram identificadas oito técnicas de teste sendo utilizadas atualmente, como testes ad hoc, exploratórios, automatizados, *cleanroom*, regressão, grupos focais e testes A/B. Além disso, demonstrou como cada uma das empresas selecionadas no experimento utilizam das técnicas de teste (apresentando gráficos individuais contabilizando a menção de cada atividade de teste), sua efetividade na prática, em quais momentos elas são utilizadas no ciclo de vida de desenvolvimento, etc.

Neto et al. (2019) realizaram uma pesquisa em empresas de desenvolvimento *indie* para compreender quais atividades de teste estão sendo utilizadas com mais frequência. Segundo a pesquisa, identificou-se que a abordagem de *playtesting* é a que ocorre com mais frequência (compondo cerca de 48% das respostas), enquanto técnicas como teste de Usabilidade, Exploratório, Desempenho, ad hoc, Árvores de teste, *Cleanroom*, Aceitação, Combinatório, Unidade, entre outros integram o restante dos resultados. Além disso, identificou-se o tempo médio gasto para priorização da implementação de testes na produção de um jogo digital, onde cerca de 50% dos participantes constataram que utilizam por volta de 1%, e 25% do tempo de produção é utilizado para realizar tarefas de teste.

Bethke (2003) apresenta em seu livro a fundamentação teórica acerca do desenvolvimento de jogos em geral, os papéis envolvidos (abordando o time de controle de qualidade), a documentação produzida e outros fatores chave no ciclo de vida de implementação. Em relação as técnicas mencionadas, o autor faz questão de trazer a visão geral das categorias de teste (caixa branca e preta), além de focar a argumentação sobre as fases *Alpha* e *Beta* de testes, tendo como base Grupos Focais sendo a principal técnica definida e utilizada. Outras técnicas são mencionadas (testes de regressão, *ad hoc*, automatizados), porém, não apresentam uma descrição completa.

Novak (2011) apresenta e caracteriza com detalhes, em seu livro, o processo completo de como desenvolver um jogo digital e obter resultados satisfatórios em fatores como enredo, história, personagens, mecânicas de jogo, etc. O foco relacionado entre as técnicas de teste gira em torno das etapas *Alpha* e *Beta* de desenvolvimento, abordando brevemente técnicas como teste de Progressão e *stress*, trazendo a técnica de *playtesting* como a mais efetiva e produtiva na etapa de testes.

Schultz e Bryant (2016) apresentam e definem em seu livro a importância da etapa de testes e as etapas do ciclo de desenvolvimento. Técnicas como teste Combinatório, *Cleanroom*, ad hoc, Diagramas de Fluxo, Árvores de teste e *Playtesting* são abordadas em detalhes, além da sua efetividade.

Alem, Capretz e Ahmed (2016a) definem quais etapas são úteis na etapa de

Produção do ciclo de desenvolvimento de jogos digitais. As técnicas de *Smoke*, Regressão, Conectividade, Performance, Aceitação e Testes Funcionais são caracterizadas como as mais efetivas, tendo em vista a gestão do plano de testes elaborado na etapa de Pré-Produção.

Aleem, Capretz e Ahmed (2016b) tratam de apresentar a fase *Beta* de testes como principal fator na etapa de desenvolvimento e sua relevância, além da descrição de técnicas baseadas em Heurísticas, Testes Empíricos e testes Funcionais.

Choi et al. (2016) apresentam e focam na técnica de *playtesting* em seu estudo, apresentando como ela tem sido utilizada na indústria, a sua efetividade, aplicando-a em alguns projetos sendo desenvolvidos, analisando e obtendo resultados dos experimentos.

Kasurinen e Smolander (2014) abordam sobre como as empresas testam seus produtos, seus principais métodos e objetivos a serem alcançados nesta etapa. Uma pesquisa qualitativa com sete empresas foi elaborada e os resultados mostraram que as técnicas de teste mais utilizadas, segundo os desenvolvedores, são testes de Usabilidade, Exploratório, Unidade e de Integração.

Korhonen (2010) define uma comparação entre a técnica de *Playtesting* e métodos de revisões avançadas em jogos digitais, abordando quais procedimentos, etapas e aspectos foram analisados para definir as vantagens e problemas de ambas abordagens.

Mirza-Babaei, Moosajee e Drenikow (2016) apresentam a técnica de *Playtesting* em empresas *indies*, tendo como objetivo observar a efetividade da técnica, aplicando diversas métricas para entender sua praticidade, como questionários, heurísticas, protocolos e entrevistas. Observou-se que ainda é preciso um melhor entendimento sobre *Playtesting* e a agregação de valor para o produto final.

Murphy-Hill, Zimmermann e Nagappan (2014) tratam sobre a diferença entre o desenvolvimento de jogos digitais e softwares comuns. Uma pesquisa foi realizada com desenvolvedores que atuaram em ambos cenários e foi identificado que testes de Unidade e Automatizados são geralmente práticas comuns à serem realizadas.

Burnstein (2003) aborda em seu livro práticas de teste em geral, contextualizando suas respectivas características e objetivos na etapa de testes. Práticas como testes Funcionais, Performance, *Stress*, Configuração, Segurança, Recuperação, Regressão, Aceitação (nas etapas *Alpha* e *Beta* de desenvolvimento), Exploratório, Validação e de Comparação são abordadas em detalhes.

Li et al. (2022) apresentam uma ferramenta que permite testes automatizados em jogos digitais (*GBGallery*), com o objetivo de facilitar a pesquisa e prática de técnicas na etapa de testes. A ferramenta ainda fornece um banco de dados contendo uma listagem de *bugs* conhecidos, além das possibilidades a serem testadas internamente no produto em questão.

Ramadan e Hendradjaya (2014) apresentam um modelo de testes para jogos digitais que possibilita a medição de qualidade, onde a principal técnica foca em testes ad hoc,

além da utilização de revisões e discussões em grupos focais e demonstrações do produto final.

Hooper (2017) foca na automação de testes e validação da parte gráfica de jogos digitais, no qual geralmente apresentam erros constantes que impactam negativamente a jogabilidade. Técnicas como teste *Smoke*, Regressão, *Playtesting*, Grupos Focais, Balanceamento, Compatibilidade, Aceitação, Localização e Usabilidade foram automatizados para a realização dos experimentos e análise dos resultados em diferentes jogos.

Pietriková e Sobota (2022) abordam sobre o desenvolvimento e testes em jogos educacionais, obtendo *feedback* do uso de técnicas como Grupos Focais, *Playtesting* e Usabilidade, analisando o resultado das práticas e sua efetividade.

Mirza-Babaei et al. (2020) analisam o impacto da técnica de *Playtesting* na Pós-Produção de jogos digitais e sua recepção pelos usuários. Um experimento com três jogos foi elaborado e, a partir da análise dos resultados, identificou-se em quais aspectos a técnica de *Playtesting* afetava tanto positivamente quanto negativamente.

Rajanen e Tapani (2018) propõem uma entrevista com empresas de desenvolvimento de jogos dos Estados Unidos para identificar práticas de teste de Usabilidade. Notou-se que as empresas testam a usabilidade de seus jogos a partir de alguns princípios e abordagens, como protótipos de baixa e alta fidelidade, versões *Alpha* e *Beta* de seus jogos, além da comparação com os produtos de empresas competidoras presentes na indústria.

Racheva (2019) exploram as técnicas de teste utilizadas em jogos *indies* de acordo com uma visão de engenharia de *software*. Práticas como *Playtesting*, teste Exploratório, Regressão e Unidade são abordadas.

Negrão (2020) foca na caracterização da técnica de *Playtesting* de forma automatizada, apresentando o contexto de se utilizar uma abordagem de automatização de testes relacionada à técnica principal abordada no estudo. Além disso, o ferramental proposto pelo software *Unity* é apresentado em detalhes, e a realização do experimento sobre como os testes automatizados foram feitos em dois jogos digitais são relatados no estudo.

Mozgovoy e Pyshkin (2018) definem a etapa de garantia de qualidade em um jogo digital para celular. Técnicas como *Playtesting*, Usabilidade, testes manuais e *Smoke* são definidas e apresentadas no estudo.

Cho (2022) trata sobre a descoberta de *bugs* encontrados no desenvolvimento de jogos digitais, tendo em vista o orçamento do projeto, apresentando técnicas como testes de Unidade, Integração, *Smoke*, Regressão, Performance, além da automatização dos mesmos.

Annanperä et al. (2018) apresentam métodos de teste no desenvolvimento de jogos *mobile*. Identificou-se que a técnica de Grupo Focais é a mais importante e que avalia de melhor forma um jogo para celular, presente exclusivamente nas etapas de teste *Alpha* e *Beta* de produção, já que o *feedback* de observações e discussões são analisados de forma

ágil e consistente pelos desenvolvedores.

Ramadan e Widyani (2013) definem *guidelines* no ciclo de vida do desenvolvimento de um jogo digital, o que inclui a fase de testes, onde técnicas de detalhamento formal (para identificar as funcionalidades, dificuldade, etc) e teste de refinamento são propostos, além das fases de desenvolvimento *Alpha* e *Beta* para que sejam testados aspectos funcionais e internos de mecânica do jogo.

3.4.2 Resposta da Questão 2

Nesta seção, serão discutidas quais foram as informações utilizadas para responder a Questão de Pesquisa 02: **Quais são as contribuições efetivas das técnicas de controle de qualidade para o processo de desenvolvimento de jogos digitais, contribuindo com valor agregado ao produto de software final?**. Inicialmente, serão analisadas sentenças e parágrafos que dissertem sobre tais contribuições efetivas que estão sendo utilizadas na indústria de jogos. Dados como experiências, relatos, experimentações em *cases* reais servirão como base na coleta das contribuições.

Percebe-se que a aplicação de diferentes técnicas de teste em jogos digitais assemelham contribuições efetivas entre si e são comumente mencionadas nos estudos selecionados que fazem questão de definir tais contribuições. Questões como identificação de comportamento e modo de jogabilidade de usuários são utilizadas para melhorar a *gameplay* do produto final (WHITTAKER, 2009; UNITY, 2022).

Redução de custos do orçamento e tempo estimado de teste são características positivas em relação ao se utilizar de técnicas de teste específicas em jogos digitais (NOVAK, 2011; BERTOLINO, 2007). Verificação robusta e completude das funcionalidades de acordo com os requisitos previamente especificados são outros fatores positivos que a aplicação das técnicas provêm para o time de desenvolvimento (NEGRÃO, 2020; RUSKA, 2015).

Diversos fatores como o gerenciamento organizado do ambiente de testes (BURNSTEIN, 2003), identificação prévia e assertiva de falhas que interrompam a *gameplay* (RAMADAN; HENDRADJAYA, 2014), cenários de testes mais efetivos (WHITTAKER, 2009), identificação de funcionalidades não planejadas e desejadas pelos usuários (LAHTI et al., 2015), além da obtenção de métricas de performance, níveis de dificuldade e eficiência da aplicação (NETO et al., 2019) são outras contribuições ao se aplicar técnicas de teste diversas na etapa de controle de qualidade em jogos digitais.

3.5 Ameaças à Validade

A ameaça interna está atrelada à análise dos estudos, onde o viés de publicação pode vir a incluir alguns estudos com baixa qualidade para responder às questões de pesquisa, bem como a falta de experiência do pesquisador pode ser uma ameaça grave.

A fim de contornar essa ameaça os estudos passam pela avaliação de qualidade, onde na tabela de controle de qualidade são removidos os artigos com pontuação zero. Ressalta-se que a seleção de poucos estudos pode ser outra ameaça às revisões de literatura.

O aspecto de validação externa se preocupa em até que ponto é possível generalizar os resultados de um estudo. Assim buscou-se estabelecer um processo de busca mais amplo, abrangente e diversificado possível, para isso, após se ter realizado um teste piloto das *strings* de busca, começou-se definitivamente as buscas em duas bases de dados, sendo elas: SCOPUS e Google Scholar. Também ressalta-se que será realizada a síntese de dados de acordo com o método de extração definido, no intuito de recuperar informações relevantes para responder às questões estabelecidas.

A validade de construção trata-se da exclusão de estudos relevantes. No intuito de mitigar tal ameaça, definiu-se a estratégia do processo de seleção, onde os estudos são selecionados conforme os critérios de exclusão e inclusão, bem como avaliação da qualidade para qualificação e classificação dos estudos. No entanto, ressalta-se que não é possível garantir que todos os estudos relevantes sejam incluídos (ALMANASREH; MOLES; CHEN, 2019).

A ameaça à confiabilidade se atenta em até que ponto os dados e a análise dependem de pesquisadores específicos. Assim, com o propósito de amenizar essa ameaça foram definidos e incluídos neste trabalho as diretrizes, processos, métodos, critérios e métricas. Permitindo assim, que terceiros atinjam os mesmos resultados ao realizar a mesma busca.

3.6 Lições do Capítulo

Neste capítulo, buscou-se apresentar toda a estratégia por trás da revisão, busca e análise de conteúdo presentes neste Trabalho de Conclusão de Curso, além de definir as questões de pesquisa e como elas foram respondidas, apresentando o detalhamento de cada estudo identificado e selecionado.

4 DESENVOLVIMENTO DA SOLUÇÃO

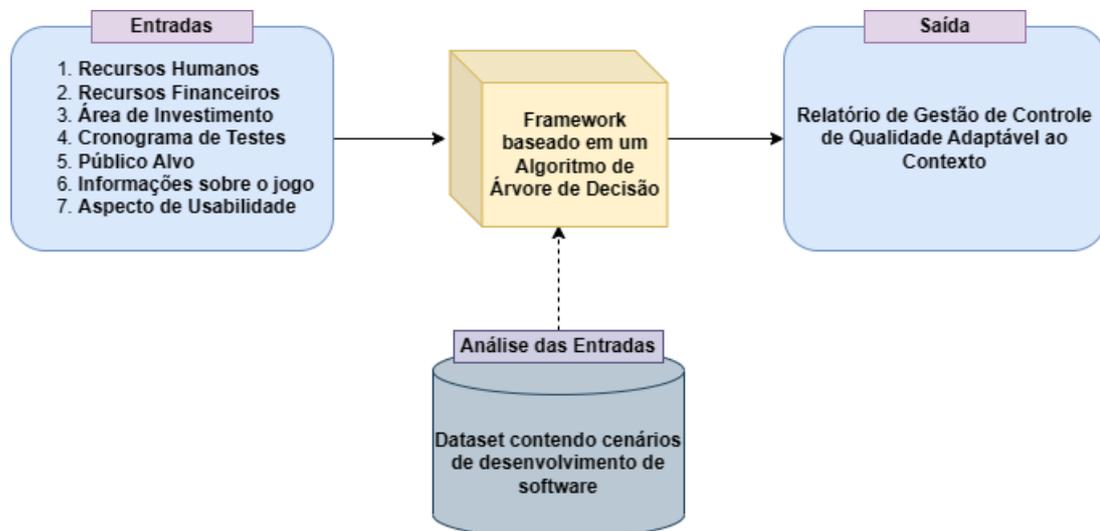
O propósito deste Capítulo se baseia na apresentação da solução proposta neste Trabalho de Conclusão de Curso, tratando-se de um software de recomendação para apoio à gestão do controle da qualidade em jogos digitais implementado utilizando o conceito de árvores de decisão. Este capítulo está organizado da seguinte forma: na Seção 4.1, será apresentada a Arquitetura Geral da solução proposta, levando em consideração suas entradas e saídas, apresentadas de forma detalhada. A Seção 4.2 apresenta todas as entradas do software e suas descrições e, por fim, a Seção 4.4 apresenta o desenvolvimento da aplicação, especificando o seu funcionamento.

4.1 Arquitetura Geral do Protótipo de Software

O intuito desse software de recomendação é realizar especificações e detalhamentos tendo em vista vários pontos-chave acerca da etapa de testes de jogos digitais. Considerando o fato de que a gestão de testes de muitos jogos digitais é realizada de forma ad hoc (KASURINEN; SMOLANDER, 2014), percebe-se que não existe uma padronização e orientação de práticas de testes no desenvolvimento de jogos. Tal padronização pode auxiliar o principal público-alvo da elaboração desta solução (desenvolvedores e testadores de jogos digitais) a obter um produto de *software* de melhor qualidade para seus jogadores, utilizando-se de técnicas adaptadas ao contexto do projeto. Com isso, eles poderão tanto entender melhor sobre o seu próprio contexto de desenvolvimento quanto o entendimento das técnicas propostas.

A Figura 7 apresenta de forma geral e simplificada as entradas do sistema pelo usuário, a maneira que as entradas são analisadas pelo algoritmo e, por fim, a saída da aplicação, contendo o relatório de gestão de controle de qualidade.

Figura 7 – Arquitetura geral da solução



4.2 Entradas da Solução

As entradas da solução se baseiam em características específicas que são analisadas pelo software preliminar e são divididas em sete grupos, que são as seguintes: Recursos Humanos, Recursos Financeiros, Área de Investimento, Cronograma de Testes, Público-Alvo, Informações sobre o jogo e Aspecto de Usabilidade.

A primeira entrada do software é em relação aos Recursos Humanos. São consideradas informações como: a quantidade de desenvolvedores, ou seja, o tamanho da equipe de testes, e a experiência na área de testes predominante dessa equipe.

A segunda entrada do software é em relação aos Recursos Financeiros. É levada em consideração a quantia a ser investida durante a etapa de testes do jogo digital. Uma vez que recursos financeiros são limitados em muitos projetos, há a possibilidade da falta ou não investimento em algumas etapas do desenvolvimento, por exemplo, na priorização de testes e controle de qualidade.

A terceira entrada do software se baseia na Área de Investimento. A partir disso, é possível definir uma análise mais focada e avaliar técnicas de teste mais acuradas ao contexto.

A quarta entrada do software é em relação ao Cronograma de Testes. É considerado o tempo empenhado para a realização da tarefa/fase de testes em questão, seja analisando um aspecto específico do sistema ou como um todo. Da mesma forma como os Recursos Financeiros, o Cronograma de um projeto pode vir a ser uma dificuldade, uma vez que funcionalidades ou outras tarefas podem levar mais tempo para serem concluídas do que o estipulado previamente.

A quinta entrada do software é em relação ao Público-Alvo. São consideradas informações como identificação do tipo de comunidade de jogadores, *fanbases*, nicho de jogadores, etc. Geralmente, diferentes comunidades de jogos possuem uma mentalidade e vida própria, buscando sempre discutir aspectos sobre um jogo ou franquia de jogos, fatos interessantes, dinâmica de mecânicas internas, uma vez que as expectativas por resultados são diferentes entre tais comunidades. Investidores, gerentes, sócios, e outras partes interessadas não são incluídas aqui neste momento, já que o foco é saber o *feedback* dos jogadores.

A sexta categoria do software é em relação a Informações sobre o jogo. Uma vez que “cada jogo é um jogo”, há a necessidade de entender o estilo e a combinação de gêneros e subgêneros, uma vez que diferentes abordagens devem ser utilizadas para testar jogos digitais. Uma análise será feita para determinar o que se está utilizando de testes em jogos de diferentes gêneros. A listagem de gêneros está apresentada na Seção 2. Além dos principais gêneros do jogo, a plataforma na qual o jogo será lançado também é uma entrada do sistema.

A sétima e última entrada do software se baseia no Aspecto de Usabilidade que será testado. Uma vez que diferentes abordagens de teste podem ser aplicadas ao software

dependendo do objetivo a ser alcançado pela equipe de desenvolvimento, é necessário entender a finalidade do teste para recomendar técnicas que atendam o mesmo.

A análise das entradas é realizada pelo algoritmo de decisão, onde ao analisar um *dataset* pré-definido, o sistema irá escolher qual o cenário que mais corresponde as entradas do usuário. Por fim, a saída da solução proposta é a listagem das técnicas de teste mais adaptadas para o contexto fornecido pelo usuário, onde uma descrição de cada técnica é exibida para que haja entendimento da vantagem de sua aplicação em um cenário real.

4.3 Cenários da Solução

Para alimentar o algoritmo de árvore de decisão, foi preciso criar um *dataset* que servirá como base de avaliação para as entradas do usuário. É por meio desse *dataset* que o algoritmo realiza a recomendação das técnicas de teste dependendo do contexto informado. Para fins de pesquisa, apenas dez cenários foram elaborados para serem utilizados como parâmetros de avaliação, uma vez que um número maior de cenários poderia escalar a complexidade da avaliação de forma abrupta.

Os dez cenários apresentados correspondem a características que compõem o desenvolvimento de um jogo digital, onde apesar de não representarem a sua magnitude por completo, expõe fatores essenciais que refletem a realidade dos estúdios de jogos. Como para a maioria das entradas não foi possível encontrar um padrão bem definido na literatura, padrões foram criados para cada aspecto, tendo em vista o seu contexto. A relação de todas as possíveis entradas elaboradas para cada característica utilizada está listada a seguir:

- **Quantidade de Testadores:** Menos que 5; Entre 5 e 15; Entre 15 e 20; Mais de 20.
- **Experiência em Testes:** Predominante: Júnior; Predominante: Pleno; Predominante: Senior e Equilíbrio entre Júnior, Pleno e Senior.
- **Investimento geral em Testes:** Entre R\$ 1.000 e R\$ 5.000; Entre R\$ 6.000 e R\$ 20.000; Entre R\$ 21.000 e R\$ 50.000; Entre R\$ 51.000 e R\$ 100.000 e Acima de R\$ 100.000.
- **Área mais relevante para investimento:** Atualizações e Correções; Certificação em Plataformas; Testes Internos e Testes com Usuários.
- **Cronograma de Testes:** Entre uma semana e duas semanas; Entre três semanas a quatro semanas; Entre um mês a dois meses e Mais que três meses.
- **Principais Gêneros:** Ação e Aventura; RPG e Ação; Estratégia e RPG; Aventura e Puzzle; Corrida e Simulação; Ação e FPS e Terror e Ação.

- **Público Alvo:** Jogadores Casuais; Jogadores *Hardcore* e Jogadores de Jogos Sociais.
- **Plataforma:** *PC*; *XBox*; *Playstation* e *Nintendo Switch*.
- **Aspecto de Usabilidade a ser testado:** Conformidade; Operabilidade; Aprendibilidade; Eficiência e Flexibilidade.

Por meio destas informações, foi possível elaborar combinações para os dez cenários, com base em similaridades e ocorrência de casos reais das informações, baseado em valores aproximados. A definição de cada técnica de teste recomendada para seus respectivos cenários se deu por uma avaliação individual das suas características, afim de compreender, com base em suposições, quais técnicas eram melhor alocadas em cada cenário. Além disso, utilizou-se da aplicação *Chat GPT* como uma ferramenta de trabalho e produtividade para reduzir o espaço de busca, com o objetivo de criação sistemática de cenários, uma vez que as características foram pré-definidas anteriormente. A relação dos cenários propostos é a seguinte, tendo em vista as informações definidas para cada característica, respectivamente:

- **Cenário 1:** Entre 5 e 15; Equilíbrio entre Júnior, Pleno e Senior; Entre R\$ 1.000 e R\$ 5.000; Testes internos; Entre uma semana e duas semanas; Ação e Aventura; Jogadores Casuais; *PC*; Aprendibilidade; **Técnicas Recomendadas:** Teste Unitário, Teste de Regressão e Teste Exploratório.
- **Cenário 2:** Mais de 20; Predominante - Senior; Acima de R\$ 100.000; Certificação em plataformas; Mais que três meses; RPG e Ação;; Jogadores *Hardcore*, *PlayStation*; Eficiência; **Técnicas Recomendadas:** Teste de Regressão, Teste de Aceitação, Teste de Performance e Teste de Funcionalidade.
- **Cenário 3:** Entre 15 e 20; Predominante - Pleno; Entre R\$ 21.000 e R\$ 50.000; Atualizações e correções; Entre um mês e dois meses; Estratégia e RPG; Jogadores de Jogos Sociais; *Nintendo Switch*; Conformidade; **Técnicas Recomendadas:** Teste de Aceitação, Teste Exploratório e Teste Automatizado.
- **Cenário 4:** Menos que 5; Predominante - Júnior; Entre R\$ 6.000 e R\$ 20.000; Testes com usuários; Entre três semanas e quatro semanas; Aventura e Puzzle; Jogadores Casuais; *XBox*; Operabilidade; **Técnicas Recomendadas:** Teste Smoke, Teste Unitário e Teste de Integração.
- **Cenário 5:** Entre 5 e 15; Predominante - Júnior; Entre R\$ 21.000 e R\$ 50.000; Certificação em plataformas; Mais que três meses; Corrida e Simulação; Jogadores Casuais; *PC*; Flexibilidade; **Técnicas Recomendadas:** Teste A/B, Teste de Performance e Teste de Aceitação.

- **Cenário 6:** Mais de 20; Equilíbrio entre Júnior, Pleno e Senior; Entre R\$ 6.000 e R\$ 20.000; Testes internos; Entre uma semana e duas semanas; Ação e FPS; Jogadores *Hardcore*; PlayStation; Aprendibilidade; **Técnicas Recomendadas:** Teste com Grupos Focais, Teste de Regressão e Teste Exploratório.
- **Cenário 7:** Entre 15 e 20; Predominante - Senior; Acima de R\$ 100.000; Atualizações e correções; Entre três semanas e quatro semanas; Terror e Ação; Jogadores *Hardcore*; PC; Eficiência; **Técnicas Recomendadas:** Teste de Regressão, Teste de Performance e Teste de Aceitação.
- **Cenário 8:** Menos que 5; Predominante - Pleno; Entre R\$ 1.000 e R\$ 5.000; Testes com usuários; Entre um mês e dois meses; Ação e Aventura; Jogadores Casuais; Nintendo Switch; Operabilidade; **Técnicas Recomendadas:** Teste de Progressão, Teste de Aceitação, Teste Exploratório e Playtesting.
- **Cenário 9:** Entre 5 e 15; Predominante - Júnior; Entre R\$ 6.000 e R\$ 20.000; Certificação em plataformas; Entre uma semana e duas semanas; RPG e Ação; Jogadores Casuais; Xbox; Flexibilidade; **Técnicas Recomendadas:** Teste Unitário, Teste de Regressão e Teste de Aceitação.
- **Cenário 10:** Mais de 20; Equilíbrio entre Júnior, Pleno e Senior; Entre R\$ 51.000 e R\$ 100.000; Testes internos; Mais que três meses; Aventura e Puzzle; Jogadores de Jogos Sociais; PlayStation; Conformidade; **Técnicas Recomendadas:** Teste de Progressão, Teste de Performance, Teste de Aceitação e Teste Exploratório.

4.4 Protótipo Preliminar de Apoio à Gestão de Controle de Qualidade

Para o desenvolvimento do *software* preliminar, foi utilizado a linguagem de programação Python, com suporte da biblioteca de Árvores de Decisão do *Scikit Learn*. Como *dataset* escolhido, ou seja, as informações de treinamento, utilizou-se os mesmos cenários aplicados no formulário. Conforme apresenta o trecho de código abaixo, são exibidas apenas três das dez *features* utilizadas, no qual o *dataset* completo está dividido entre as *features* e as classes (folhas da árvore) que serão obtidas.

```
1 def decision_tree():
2
3     #criando o dataset
4     data = pd.DataFrame({
5
6     #Quantidade de Testadores
7     'qtdTestadores':
8     ['Entre 5 e 15', 'Mais de 20', 'Entre 15 e 20', 'Menos que 5', '
Entre 5 e 15', 'Mais de 20', 'Entre 15 e 20', 'Menos que 5', 'Entre 5
e 15', 'Mais de 20'],
```

```

9
10 #Experiencia dos Testadores
11 'experiencia':
12 ['Equilibrio entre Junior, Pleno e Senior', 'Predominante -
Senior', 'Predominante - Pleno', 'Predominante - Junior', '
Predominante - Junior', 'Equilibrio entre Junior, Pleno e Senior', '
Predominante - Senior', 'Predominante - Pleno', 'Predominante -
Junior', 'Equilibrio entre Junior, Pleno e Senior'],
13
14 #Investimento em testes
15 'investimento':
16 ['Entre R$ 1.000 e R$ 5.000', 'Acima de R$ 100.000', 'Entre R$
21.000 e R$ 50.000', 'Entre R$ 6.000 e R$ 20.000', 'Entre R$ 21.000
e R$ 50.000', 'Entre R$ 6.000 e R$ 20.000', 'Acima de R$ 100.000', '
Entre R$ 1.000 e R$ 5.000', 'Entre R$ 6.000 e R$ 20.000', 'Entre R$
51.000 e R$ 100.000'],

```

Após esta etapa, é feita a transformação das informações por meio da biblioteca *OneHotEncoder*, definindo quais são as *features* e as classes nas variáveis *X_encoded* e *y*, respectivamente. Em seguida, é criada a instância do modelo utilizado pela árvore de decisão e, a partir disso todos os métodos referentes a ela podem ser usados. Na sequência, o modelo é treinado por meio do método *fit()* recebendo como parâmetro as duas variáveis *X* e *Y* previamente definidas. Por fim, acontece a exibição da árvore gerada por meio do algoritmo, contendo as informações necessárias para entender como aconteceu cada divisão em todos os nós contendo as *features* estabelecidas. O trecho de código abaixo exemplifica o que foi explicado acima.

```

1 #encoding nas features
2 encoder = OneHotEncoder(sparse=False)
3 X_encoded = encoder.fit_transform(data[['qtdTestadores', '
experiencia', 'investimento', 'areaInvestimento', 'cronograma', '
generos', 'publico', 'plataforma', 'aspectoUsabilidade']])
4
5 #criando a split_list de features
6 feature_names = list(encoder.get_feature_names(['qtdTestadores', '
experiencia', 'investimento', 'areaInvestimento', 'cronograma', '
generos', 'publico', 'plataforma', 'aspectoUsabilidade']))
7
8 #definindo a variavel alvo, ou seja, as saidas possiveis
9 y = data['outcomeTecnicas']
10
11 #criando o modelo de decisao
12 model = DecisionTreeClassifier()
13
14 #treinando o modelo, recebendo como parametro as features e as
saidas
15 model.fit(X_encoded, y)

```

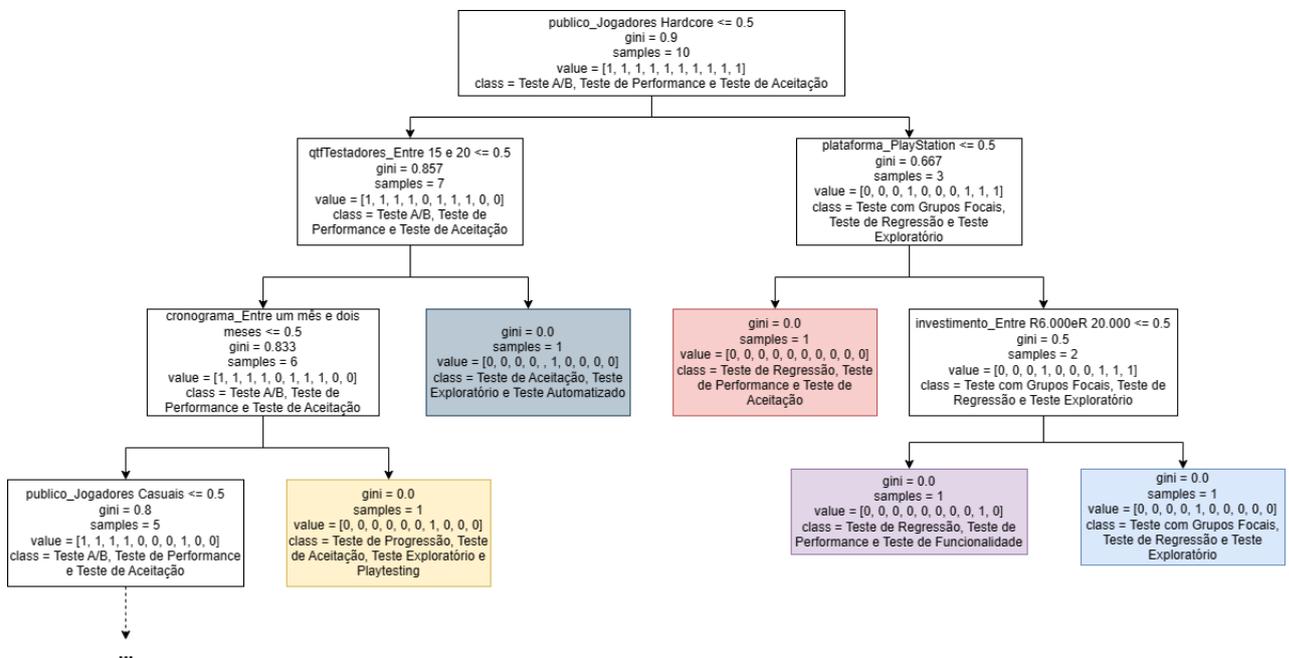
```

16 plt.figure(figsize=(60, 60))
17 tree.plot_tree(model, feature_names=feature_names, class_names=model
18 .classes_, filled=True, fontsize=11)
19 plt.savefig('estrutura_arvore_treino', bbox_inches='tight', dpi=100)

```

No momento em que o usuário utiliza o *software* preliminar, o mesmo é apresentado a uma mensagem inicial do sistema, e a partir disso responderá uma série de perguntas para que haja a melhor classificação das suas respostas de acordo com a estrutura da árvore gerada. Um exemplo de saída da estrutura da árvore de treinamento elaborada pelo algoritmo, é apresentado na Figura 8.

Figura 8 – Estrutura parcial da árvore criada pelo algoritmo

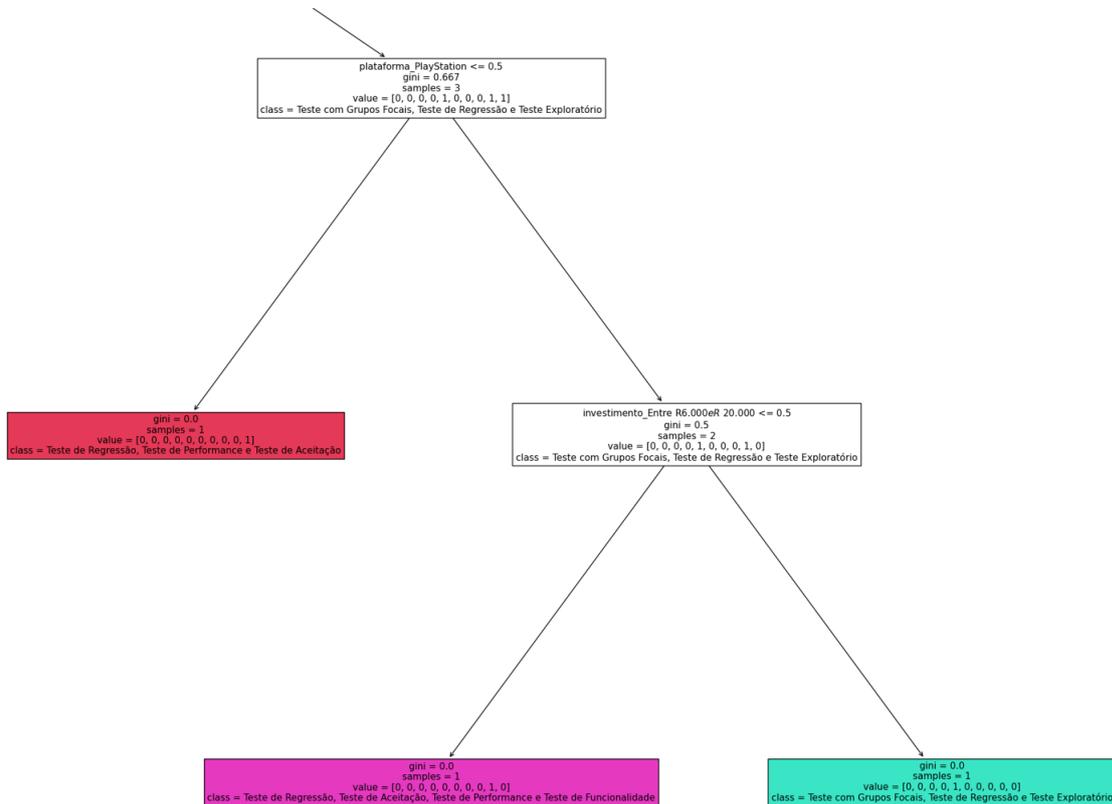


Já na Figura 9, é possível visualizar em maiores detalhes a disposição das informações. O primeiro nó que aparece é proveniente de outra divisão da raiz da árvore e neste cenário restam apenas 3 *samples* dos 10 exemplos de classes deste lado da árvore. Além disso, outras informações são apresentadas, como o valor da impureza dos valores do *dataset* pelo índice *Gini*, quantos valores de classe restam e quais valores de cada *feature* ainda restam, e a classe na qual esse conjunto de valores representa. Ao final da classificação, um nó folha é escolhido de acordo com a divisão, sendo destacados com cores fortes.

A execução do *software* preliminar pelo usuário ocorre de forma simples e objetiva: nove perguntas serão feitas onde o usuário irá respondê-las de acordo com o contexto, tanto escolhendo uma das opções apresentadas quanto digitando informações específicas. As perguntas realizadas durante a execução do sistema são as seguintes:

- Qual a QUANTIDADE de testadores na sua equipe?

Figura 9 – Exemplo de galho gerado pelo algoritmo



- Qual a EXPERIÊNCIA geral da equipe em relação a TESTES DE SOFTWARE?
- Qual o valor, em reais (R\$), a ser INVESTIDO na etapa de testes?
- Qual a ÁREA mais relevante para investimento?
- Qual a estimativa de duração do CRONOGRAMA de testes?
- Quais os principais GÊNEROS que se enquadram no seu jogo?
- Qual o PÚBLICO-ALVO do seu jogo?
- Qual a PLATAFORMA principal que seu jogo será lançado?
- Qual o aspecto de USABILIDADE que será testado?

É importante salientar que, como se trata de um projeto preliminar, é necessária uma validação mais completa e robusta com mais pessoas participando desse processo. As entradas do usuário são rígidas e não permitem uma flexibilidade tão grande. Por exemplo, caso um desenvolvedor esteja utilizando o *software* e caso seu jogo seja lançado em mais de uma plataforma, o relatório não englobaria todo o seu contexto interno e poderia resultar em um falso positivo, já que no momento só é possível escolher uma plataforma onde o

jogo será lançado. Vale também ressaltar que, tanto os valores de investimento, duração de estimativa de testes, quanto as outras informações, foram baseadas em hipóteses no qual ocorrem em cenários reais de desenvolvimento de jogos na indústria atual.

Ao final da execução do sistema, um simples relatório é apresentado ao usuário, apresentando quais foram as entradas até então inseridas, a definição das melhores técnicas de teste para aquele contexto e a explicação de cada uma das técnicas. A Figura 10 apresenta como seria essa visualização.

Figura 10 – Exemplo de relatório gerado pelo sistema

```
===== RELATÓRIO FINAL v1 =====
Tendo em vista o seu cenário:

> Quantidade de Testadores: Entre 5 e 15
> Experiência em testes: Predominante - Júnior
> Investimento em testes: Entre R$ 1.000 e R$ 5.000
> Área mais relevante para testes: Testes internos
> Cronograma para atividades de teste: Entre uma semana e duas semanas
> Principais gêneros do jogo: Estratégia e RPG
> Público alvo: Jogadores Casuais
> Plataforma de lançamento: PC
> Aspecto de Usabilidade a ser testado: Operabilidade

As melhores técnicas de teste para seu jogo são: Teste Unitário, Teste de Regressão e Teste Exploratório

>>> Teste Unitário: Testes Unitários são realizados pelos desenvolvedores para garantir o comportamento desejado de uma seção específica da aplicação. Boas práticas de teste incluem testes unitários para revelar defeitos funcionais, erros de código, dados, controle e sequência. A unidade a ser testada deve ser validada por um testador independente, alguém sem contato direto com o código durante a implementação. Essa técnica é eficiente para identificar problemas precocemente no ciclo de desenvolvimento. Testes bem elaborados facilitam a detecção de erros introduzidos durante mudanças no código, melhorando a manutenção geral e o tempo de execução.

>>> Teste de Regressão: A necessidade de testes de regressão decorre do fato de que, quando os desenvolvedores estão corrigindo bugs, eles podem facilmente reintroduzir defeitos antigos no código ou gerar defeitos completamente novos. O teste de regressão procura essencialmente e pros antigos no código atual. É realizado decidindo quais testes devem ser executados em cada versão do jogo, mas também pode envolver a modificação de testes existentes ou a criação de testes completamente novos. O teste de regressão eficiente minimizará o número de testes executados, mas ainda testará defeitos recém introduzidos e remanescentes.

>>> Teste Exploratório: Os Testes Exploratórios permitem que os testadores utilizem a aplicação sem restrições, explorando o jogo de acordo com sua vontade. No entanto, é importante incluir casos de teste, documentação e resultados. Ferramentas de captura de tela podem ser usadas para registrar os resultados desses testes. Essa técnica é vantajosa quando o projeto não possui casos de teste ou instruções pré-definidas a serem seguidas.
```

4.5 Lições do Capítulo

Tendo em vista o que foi apresentado neste capítulo, percebe-se que a maneira de como um software é testado, especificamente um jogo digital complexo, não é uma tarefa trivial. Inúmeras características devem ser levadas em consideração por toda a equipe de desenvolvimento, a fim de se realizar um bom planejamento de testes e execução de testes, utilizando-se das melhores práticas disponíveis, levando em conta o contexto do projeto em si. Propõe-se que a solução definida neste capítulo tenha um impacto positivo no processo interno das empresas de desenvolvimento de jogos digitais, auxiliando na entrega de um produto de software de melhor qualidade.

5 AVALIAÇÃO DA SOLUÇÃO

O propósito deste Capítulo se baseia na discussão dos resultados da avaliação aplicada e a apresentação da avaliação da solução preliminar proposta neste Trabalho de Conclusão de Curso II, tratando-se de um *software* preliminar que fornece suporte à gestão do controle da qualidade em jogos digitais. Este Capítulo está organizado da seguinte forma: a Seção 5.1 trata sobre mudanças e definição de escopo da pesquisa e como a avaliação foi planejada a fim de obter os resultados. A Seção 5.2 exemplifica a execução da avaliação em si, tratando sobre os resultados obtidos. A Seção 4.4 demonstra o *software* desenvolvido para gestão de controle de qualidade e, por fim, a Seção 5.3 trata da análise final dos resultados levantados.

5.1 Planejamento da Avaliação

Para a elaboração do escopo e planejamento da avaliação proposta, foi realizado um refinamento da proposta inicial do software de recomendação de apoio à gestão de controle de qualidade. Em vez de abranger vários aspectos que caracterizam um jogo digital, foi definido a escolha da característica de Usabilidade como foco central da pesquisa e busca da resolução do problema em questão. Esta mudança trouxe algumas vantagens, como o foco em apenas uma linha de pesquisa, gerando uma análise mais precisa e com resultados simples, além de diminuir a complexidade do escopo, mas realizando uma análise mais precisa do produto de trabalho realizado.

Além disso, foi discutida a ideia de elaborar uma proposta de *software* utilizando do conceito de árvores de decisão, uma vez que o escopo do problema pode ser resolvido com algoritmos de classificação, sendo visualmente fácil de entender a sua estrutura e conceitos fundamentais. O objetivo é auxiliar a equipe de desenvolvimento a compreender quais técnicas de teste são mais adequadas de acordo com o seu contexto, tendo em vista características como cronograma, quantidade de testadores, área de teste mais relevante, etc.

Ao definir os novos objetivos e artefatos deste trabalho, foi iniciada a elaboração de um formulário de pesquisa no site do *Google Forms* para que pessoas com um perfil específico de participante colaborassem no estudo. Diversos contatos e convites foram enviados para empresas brasileiras de desenvolvimento de jogos e para professores de múltiplas instituições, onde, preferencialmente, os participantes de ambos os setores possuíssem experiência na área de teste de *software* e/ou controle de qualidade aplicada a jogos digitais.

5.2 Execução da Avaliação

Após a etapa de elaboração e validação das informações e perguntas do formulário de pesquisa, o mesmo foi distribuído via e-mail para os participantes que mostraram

interesse em participar do estudo. O formulário está dividido nas seguintes seções:

- Texto introdutório e consentimento de participação;
- Coleta de informações pessoais do participante;
- Disponibilização de material de suporte e apresentação dos cenários de desenvolvimento de jogos digitais.

A primeira seção refere-se a introdução do problema da pesquisa, o que será avaliado durante a participação do formulário e o consentimento de participação do entrevistado. A segunda seção possui o intuito de coletar as informações pessoais de cada participante, a fim de entender melhor o seu histórico profissional e fazer a relação com as suas respectivas respostas, como tempo de experiência em testes de software e/ou controle de qualidade em jogos e a empresa ou instituição que se está relacionado. Por fim, a terceira seção apresenta dez cenários elaborados contendo características distintas da etapa de testes e controle de qualidade durante o desenvolvimento de um jogo digital. Além disso, foi disponibilizado um documento contendo dados introdutórios sobre todas as técnicas de teste mapeadas durante este estudo, com o intuito de fornecer um entendimento maior ao participante da pesquisa. O objetivo do participante foi analisar as informações apresentadas e definir por meio de uma escala a sua opinião a respeito da técnica de teste a ser utilizada naquele momento. A escala é conhecida como Escala Likert (LIKERT, 1932) e possui notas de 1 (discordo fortemente) a 5 (concordo fortemente). Além disso, o participante obrigatoriamente deveria expôr uma explicação do porquê da sua escolha. A Figura 11 representa um exemplo de cenário que foi proposto no formulário.

Após realizar a análise do contexto, o participante foi convidado a explicar o porquê da sua escolha, com o objetivo de validar de forma robusta se realmente as técnicas de teste apresentadas são válidas para utilização daquele cenário. A Figura 12 mostra o espaço para elaboração da justificativa de escolha do participante.

5.3 Análise dos Resultados

Como explicado anteriormente, o convite para participação desta pesquisa contendo o formulário do *Google Forms* foi enviado para diversas empresas e profissionais da área de teste de *software* e/ou controle de qualidade. Entretanto, não houve uma forte adesão à participação do estudo e apenas três pessoas colaboraram com o resultado final, avaliando os cenários apresentados e discutindo sobre as suas escolhas. Apesar deste número baixo de participantes, os resultados se mostraram interessantes e de alto valor ao estudo.

Considerando o Cenário 1 apresentado na Figura 13, onde as possíveis técnicas recomendadas foram Teste Unitário, Teste de Regressão e Teste Exploratório, a avaliação

Figura 11 – Exemplo de cenário proposto ao participante

Cenário 1 *

Como você avalia a utilização das **Técnicas de Teste Recomendadas** no contexto do Cenário 1?

Quantidade de Testadores	Entre 5 e 15
Experiência em Testes	Equilíbrio entre Júnior, Pleno e Sênior
Investimento Geral em Testes	Entre R\$ 1.000 e R\$ 5.000
Área mais relevante para investimento	Testes internos
Cronograma de Testes	Entre uma semana e duas semanas
Principais Gêneros	Ação e Aventura
Público Alvo	Jogadores Casuais
Plataforma	PC
Aspecto de Usabilidade a ser testado	Aprendibilidade
Técnicas de Teste Recomendadas	<ol style="list-style-type: none"> 1. Teste Unitário, 2. Teste de Regressão, 3. Teste Exploratório.

1 2 3 4 5

Discordo Fortemente Concordo Fortemente

Figura 12 – Exemplo de justificativa de cenário apresentado

Sobre o **Cenário 1**, argumente sobre o porquê da sua escolha acima. *

Sua resposta

dos entrevistados 1 e 3 mostrou que as técnicas de Teste Unitário e de Regressão são necessárias em qualquer trabalho que envolva desenvolvimento de *software*, uma vez que, segundo à opinião pessoal de cada um, são ideais para verificar a funcionalidade correta de componentes individuais e também para encontrar defeitos e casos de uso não pensados durante o levantamento de requisitos. Apesar disso, o entrevistado 2 afirma que neste caso específico, onde o foco está na aprendizagem, as técnicas seriam genéricas demais, no qual uma avaliação de usabilidade do tipo percurso cognitivo seria mais relevante, envolvendo inspeções com especialistas focadas em aprendizagem com usuários novatos.

A análise no Cenário 1 se mostrou balanceada com as informações apresentadas, onde 33% (1 participante) avaliou com nota 5, 33% (1 participante) com nota 4 e 33% (1 participante) com nota 2 na escala de avaliação, observando-se que apenas o participante 2 sugeriu avaliações de usabilidade como uma nova técnica a ser aplicada.

Considerando o Cenário 2 apresentado na Figura 14, onde as possíveis técnicas recomendadas foram Teste de Regressão, Teste de Aceitação, Teste de Performance e

Figura 13 – Cenário 1 do formulário

Quantidade de Testadores	Entre 5 e 15
Experiência em Testes	Equilíbrio entre Júnior, Pleno e Sênior
Investimento Geral em Testes	Entre R\$ 1.000 e R\$ 5.000
Área mais relevante para investimento	Testes internos
Cronograma de Testes	Entre uma semana e duas semanas
Principais Gêneros	Ação e Aventura
Público Alvo	Jogadores Casuais
Plataforma	PC
Aspecto de Usabilidade a ser testado	Aprendibilidade
Técnicas de Teste Recomendadas	<ol style="list-style-type: none"> 1. Teste Unitário, 2. Teste de Regressão, 3. Teste Exploratório.

Teste de Funcionalidade, as opiniões dos entrevistados divergem em relação às técnicas de teste mais adequadas neste contexto. O entrevistado 1 destaca a importância do teste de performance, especialmente para equipes sênior e um público-alvo *hardcore*, sugerindo uma abordagem mais detalhada. Já o entrevistado 2 enfatiza a relevância dos testes de usabilidade, considerando a eficiência sob a perspectiva da usabilidade e ressaltando a importância dos testes de progressão em jogos de RPG com histórias lineares. No entanto, ele considera o teste de regressão genérico para o contexto em questão. Por sua vez, o entrevistado 3 concorda com as técnicas recomendadas, mas discorda da ordem sugerida, propondo uma sequência de testes diferente, colocando o teste de regressão em primeiro lugar, seguido pela funcionalidade, performance e aceitação. Ele argumenta que, nesse estágio avançado de desenvolvimento, o tempo mais longo necessário para o teste de regressão é justificável. Embora haja discordâncias quanto à ordem dos testes, todas as opiniões reconhecem a importância de técnicas de teste adequadas para garantir a qualidade do jogo durante a fase de testes.

A análise no Cenário 2 se mostrou condizente com as informações apresentadas, mas preferencialmente que haja adaptações na realização dos testes, onde 66% (2 participantes) avaliaram com nota 3 e 33% (1 participante) com nota 4 na escala de avaliação, onde houve concordância sobre as técnicas recomendadas.

Considerando o Cenário 3 apresentado na Figura 15, onde as possíveis técnicas recomendadas foram Teste Unitário, Teste de Regressão e Teste Exploratório, o entrevistado 1 defende a aplicação do teste de aceitação como a única técnica viável no cenário apresentado, destacando que o teste exploratório e o teste automatizado são abordagens mais específicas. Ele sugere o uso do teste *Cleanroom* para emular o comportamento do jogador e testar a conformidade do aspecto de usabilidade. Já o entrevistado 2 propõe a inclusão de técnicas de inspeção de conformidade para detectar e corrigir problemas. Por sua vez, o Entrevistado 3 discorda da ordem sugerida, propondo o teste automatizado em

Figura 14 – Cenário 2 do formulário

Quantidade de Testadores	Mais de 20
Experiência em Testes	Predominante - Sênior
Investimento Geral em Testes	Acima de R\$ 100.000
Área mais relevante para investimento	Certificação em plataformas
Cronograma de Testes	Mais que três meses
Principais Gêneros	RPG e Ação
Público Alvo	Jogadores Hardcore
Plataforma	PlayStation
Aspecto de Usabilidade a ser testado	Eficiência
Técnicas de Teste Recomendadas	<ol style="list-style-type: none"> 1. Teste de Regressão, 2. Teste de Aceitação, 3. Teste de Performance. 4. Teste de Funcionalidade

primeiro lugar, seguido pelo teste exploratório e, por fim, o teste de aceitação. Ele argumenta que essa sequência busca garantir o menor número de *bugs* e proporcionar uma boa experiência ao usuário final. No entanto, todos os entrevistados concordam que o tempo alocado para essa etapa de avaliação é condizente com sua importância. Embora haja divergências em relação às técnicas prioritárias, as opiniões convergem na necessidade de aplicar abordagens adequadas para avaliar a conformidade e a usabilidade do jogo.

A análise no Cenário 3 se mostrou não muito otimizada com as informações apresentadas, onde 33% (1 participante) avaliou com nota 4, 33% (1 participante) com nota 3 e 33% (1 participante) com nota 2 na escala de avaliação, tendo em vista as técnicas recomendadas pelos participantes, como Teste *Cleanroom*, técnicas de inspeção e conformidade e alteração da ordem dos testes.

Figura 15 – Cenário 3 do formulário

Quantidade de Testadores	Entre 15 e 20
Experiência em Testes	Predominante - Pleno
Investimento Geral em Testes	Entre R\$ 21.000 e R\$ 50.000
Área mais relevante para investimento	Atualizações e correções
Cronograma de Testes	Entre um mês e dois meses
Principais Gêneros	Estratégia e RPG
Público Alvo	Jogadores de Jogos Sociais
Plataforma	Nintendo Switch
Aspecto de Usabilidade a ser testado	Conformidade
Técnicas de Teste Recomendadas	<ol style="list-style-type: none"> 1. Teste de Aceitação, 2. Teste Exploratório, 3. Teste Automatizado.

Considerando o Cenário 4 apresentado na Figura 16, onde as possíveis técnicas recomendadas foram Teste *Smoke*, Teste Unitário e Teste de Integração, novamente, as

avaliações dos entrevistados apresentam perspectivas diferentes em relação às técnicas de avaliação mais adequadas. O entrevistado 1 sugere o uso de testes unitários, considerando a quantidade limitada de testadores e o fato de serem membros juniores da equipe. Ele destaca que essa abordagem não apenas ajudaria os testadores a aprenderem mais sobre o processo, mas também seria eficiente para identificar problemas precocemente. O entrevistado 2 acrescenta que, além dos testes propostos, o teste de usabilidade com usuários, mesmo que seja realizado com apenas um usuário, é importante nesse contexto. Ele sugere complementar esse teste com testes exploratórios. Já o entrevistado 3 concorda com a ordem proposta, considerando o teste de integração como relevante e potencialmente mais demorado do que os testes exploratórios. Ele ressalta que a experiência dos testadores pode não ser diretamente correlacionada com o tempo necessário para cada tipo de teste, embora haja divergências quanto à relevância de certas técnicas.

A análise no Cenário 4 se mostrou eficaz com as informações apresentadas, onde 66% avaliou com nota 4 e 33% com nota 5 na escala de avaliação, sendo um dos cenários com maior avaliação entre os participantes, apesar da recomendação de algumas técnicas, como *Playtesting* e testes exploratórios.

Figura 16 – Cenário 4 do formulário

Quantidade de Testadores	Menos que 5
Experiência em Testes	Predominante - Júnior
Investimento Geral em Testes	Entre R\$ 6.000 e R\$ 20.000
Área mais relevante para investimento	Testes com usuários
Cronograma de Testes	Entre três semanas e quatro semanas
Principais Gêneros	Aventura e Puzzle
Público Alvo	Jogadores Casuais
Plataforma	XBox
Aspecto de Usabilidade a ser testado	Operabilidade
Técnicas de Teste Recomendadas	<ol style="list-style-type: none"> 1. Teste Smoke, 2. Teste Unitário, 3. Teste de Integração.

Considerando o Cenário 5 apresentado na Figura 17, onde as possíveis técnicas recomendadas foram Teste A/B, Teste de Performance e Teste de Aceitação, o entrevistado 1 destaca a importância do teste de aceitação, considerando que estão lidando com jogadores casuais e desenvolvedores juniores. Ele sugere que quase todos os testes sejam aplicados de alguma forma, com ênfase nesse método. O entrevistado 2 ressalta a importância das inspeções de conformidade e dos testes de sistema, especialmente no contexto de certificação em plataformas. Ele menciona que, devido à pouca experiência dos desenvolvedores juniores, as inspeções de conformidade podem ser guiadas por *checklists*. Por sua vez, o entrevistado 3 comenta que a quantidade de testadores poderia ser maior para garantir uma execução mais precisa do teste A/B, mas considera que, de forma geral, as

abordagens estão corretas. Embora haja divergências em relação a quantidade de testadores e a ênfase nas técnicas de avaliação, todos os entrevistados concordam que a aplicação de testes adequados é essencial para lidar com jogadores casuais, desenvolvedores juniores e assegurar a conformidade e qualidade do jogo.

Figura 17 – Cenário 5 do formulário

Quantidade de Testadores	Entre 5 e 15
Experiência em Testes	Predominante - Júnior
Investimento Geral em Testes	Entre R\$ 21.000 e R\$ 50.000
Área mais relevante para investimento	Certificação em plataformas
Cronograma de Testes	Mais que três meses
Principais Gêneros	Corrida e Simulação
Público Alvo	Jogadores Casuais
Plataforma	PC
Aspecto de Usabilidade a ser testado	Flexibilidade
Técnicas de Teste Recomendadas	<ol style="list-style-type: none"> 1. Teste A/B, 2. Teste de Performance, 3. Teste de Aceitação.

A análise no Cenário 5 mostrou a necessidade de várias refatorações a serem feitas. 33% (1 participante) avaliou com nota 4, 33% (1 participante) com nota 3 e 33% (1 participante) com nota 2 na escala de avaliação.

Considerando o Cenário 6 apresentado na Figura 18, onde as possíveis técnicas recomendadas foram Teste com Grupos Focais, Teste de Regressão e Teste Exploratório, o entrevistado 1 destaca a importância do *playtesting* devido à grande quantidade de jogadores e desenvolvedores envolvidos. O entrevistado 1 enfatiza a necessidade de analisar métricas, como eficiência na coleta de *insights*, e compreender o comportamento de cada jogador em um tipo específico de jogo. O entrevistado 2 menciona que, considerando a restrição de recursos financeiros e de tempo, o percurso cognitivo e significativo. No entanto, o entrevistado 2 sugere substituir os grupos focais por testes de usabilidade com usuários, como o *playtesting* de usabilidade, para aproveitar melhor as percepções e experiências individuais, sem o viés da interação em grupo. O entrevistado ainda argumenta que o foco nesse contexto é na facilidade de aprendizagem e que o teste de regressão parece genérico. Por sua vez, o entrevistado 3 comenta que a bateria de testes parece adequada para essa etapa, mas expressa dúvidas sobre se a quantidade de informações coletadas seria suficiente dentro do tempo determinado (entre uma semana e duas semanas). Embora haja divergências quanto às técnicas prioritárias e às restrições de tempo, todos os entrevistados reconhecem a importância de obter *insights* e entender o comportamento dos jogadores para aprimorar a facilidade de aprendizagem e a qualidade do jogo.

A análise no Cenário 6 mostrou a análise mais contestada pelos participantes, onde 33% (1 participante) avaliou com nota 5, 33% (1 participante) com nota 3 e 33% (1

participante) com nota 1 na escala de avaliação.

Figura 18 – Cenário 6 do formulário

Quantidade de Testadores	Mais de 20
Experiência em Testes	Equilíbrio entre Júnior, Pleno e Sênior
Investimento Geral em Testes	Entre R\$ 6.000 e R\$ 20.000
Área mais relevante para investimento	Testes internos
Cronograma de Testes	Entre uma semana e duas semanas
Principais Gêneros	Ação e FPS
Público Alvo	Jogadores Hardcore
Plataforma	PlayStation
Aspecto de Usabilidade a ser testado	Aprendibilidade
Técnicas de Teste Recomendadas	<ol style="list-style-type: none"> 1. Teste com Grupos Focais, 2. Teste de Regressão, 3. Teste Exploratório.

Considerando o Cenário 7 apresentado na Figura 19, onde as possíveis técnicas recomendadas foram Teste de Regressão, Teste de Performance e Teste de Aceitação, o entrevistado 1 destaca novamente a importância do *playtesting* para medir a eficiência, observar o comportamento dos jogadores e corrigir *bugs*, a fim de evitar feedbacks negativos e garantir mecânicas mais fluidas, especialmente considerando o público-alvo em questão. O entrevistado 2 menciona a relevância dos testes de usabilidade com usuários, mesmo com restrição de tempo, sugerindo a possibilidade de envolver menos usuários nesse contexto. Além disso, o entrevistado 2 destaca a importância de técnicas formais de análise de tempo de reação, como o GOMS (técnica de análise de tarefas que se concentra nos processos cognitivos envolvidos na execução de uma tarefa), para jogos de gênero de terror e ação, onde o tempo de avaliação é limitado e há profissionais experientes envolvidos. Ele considera o teste de regressão genérico para o contexto. Já o Entrevistado 3 comenta que a quantidade de pessoas envolvidas nos testes poderia ser menor ou que a experiência em testes poderia ser mais baixa, mas considera o tempo alocado condizente. Ele ressalta que, dada a natureza do público-alvo *hardcore*, é provável que eles encontrem problemas mais sutis e apresentem reclamações sobre falhas de desempenho. Embora haja divergências quanto ao tamanho da equipe de testes e à experiência necessária, todos os entrevistados concordam que o tempo alocado é apropriado e reconhecem a importância de garantir a eficiência, a jogabilidade fluida e a performance satisfatória para atender às expectativas do público-alvo.

A análise no Cenário 7 mostrou novamente ser motivo de contestação pelos participantes, onde 33% (1 participante) avaliou com nota 5, 33% (1 participante) com nota 3 e 33% (1 participante) com nota 1 na escala de avaliação.

Considerando o Cenário 8 apresentado na Figura 20, onde as possíveis técnicas recomendadas foram Teste de Progressão, Teste de Aceitação, Teste Exploratório e *Play-*

Figura 19 – Cenário 7 do formulário

Quantidade de Testadores	Entre 15 e 20
Experiência em Testes	Predominante - Sênior
Investimento Geral em Testes	Acima de R\$ 100.000
Área mais relevante para investimento	Atualizações e correções
Cronograma de Testes	Entre três semanas e quatro semanas
Principais Gêneros	Terror e Ação
Público Alvo	Jogadores Hardcore
Plataforma	PC
Aspecto de Usabilidade a ser testado	Eficiência
Técnicas de Teste Recomendadas	<ol style="list-style-type: none"> 1. Teste de Regressão, 2. Teste de Performance, 3. Teste de Aceitação.

testing, O entrevistado 1 sugere a aplicação de um teste A/B devido à quantidade de testadores disponíveis e sua experiência plena. Ele considera que essa abordagem seria adequada para o contexto. O entrevistado 2 concorda com as técnicas propostas, indicando que elas são adequadas para o contexto em questão. Por outro lado, o entrevistado 3 expressa conformidade com o tempo alocado, a ordem dos testes e as próprias técnicas. No entanto, ele destaca a preocupação com a quantidade de testadores, sugerindo que poderia ser maior. Embora haja divergência em relação à quantidade de testadores, todos os entrevistados concordam que as técnicas de avaliação propostas são, em geral, apropriadas para o contexto específico, considerando a quantidade de testadores e sua experiência.

A análise no Cenário 8 mostrou ser bastante positiva, onde 33% (1 participante) avaliou com nota 5, 33% (1 participante) com nota 4 e 33% (1 participante) com nota 3 na escala de avaliação.

Figura 20 – Cenário 8 do formulário

Quantidade de Testadores	Menos que 5
Experiência em Testes	Predominante - Pleno
Investimento Geral em Testes	Entre R\$ 1.000 e R\$ 5.000
Área mais relevante para investimento	Testes com usuários
Cronograma de Testes	Entre um mês e dois meses
Principais Gêneros	Ação e Aventura
Público Alvo	Jogadores Casuais
Plataforma	Nintendo Switch
Aspecto de Usabilidade a ser testado	Operabilidade
Técnicas de Teste Recomendadas	<ol style="list-style-type: none"> 1. Teste de Progressão, 2. Teste de Aceitação, 3. Teste Exploratório. 4. Playtesting

Considerando o Cenário 9 apresentado na Figura 21, onde as possíveis técnicas recomendadas foram Teste Unitário, Regressão e Teste de Aceitação, o entrevistado 1 sugere que a técnica de aceitação seja aplicada para garantir que o produto esteja sendo implementado de acordo com os requisitos. Ele considera essa abordagem adequada para o contexto em questão. O entrevistado 2 concorda com a importância das inspeções de conformidade, especialmente guiadas por *checklists* devido à pouca experiência dos desenvolvedores juniores, e destaca a relevância dos testes de sistema, mesmo com o tempo limitado disponível. No entanto, ele considera que os testes unitários e de regressão contribuem pouco nesse contexto específico. Por sua vez, o entrevistado 3 expressa satisfação com as técnicas propostas, com exceção da experiência dos testadores, que ele sugere que poderia ser maior para esses testes. Embora haja divergências em relação a importância dos testes unitários e de regressão e a experiência dos testadores, todos os entrevistados concordam que a técnica de aceitação e as inspeções de conformidade são relevantes para garantir a conformidade com os requisitos e a certificação em plataformas, especialmente considerando a experiência dos desenvolvedores juniores e o tempo disponível.

A análise no Cenário 9 mostrou ser condizente ao contexto para a maioria dos entrevistados, onde 33% (1 participante) avaliou com nota 5,33% (1 participante) com nota 4 e 33% (1 participante) com nota 1 na escala de avaliação.

Figura 21 – Cenário 9 do formulário

Quantidade de Testadores	Entre 5 e 15
Experiência em Testes	Predominante - Júnior
Investimento Geral em Testes	Entre R\$ 6.000 e R\$ 20.000
Área mais relevante para investimento	Certificação em plataformas
Cronograma de Testes	Entre uma semana e duas semanas
Principais Gêneros	RPG e Ação
Público Alvo	Jogadores Casuais
Plataforma	XBox
Aspecto de Usabilidade a ser testado	Flexibilidade
Técnicas de Teste Recomendadas	<ol style="list-style-type: none"> 1. Teste Unitário, 2. Teste de Regressão, 3. Teste de Aceitação.

Considerando o Cenário 10 apresentado na Figura 22, onde as possíveis técnicas recomendadas foram Teste de Progressão, Teste de Performance, Teste de Aceitação e Teste Exploratório, o entrevistado 1 considera todas as técnicas recomendadas como eficazes para esse tipo de situação, indicando uma visão geral positiva. O entrevistado 2 concorda com as técnicas propostas, mas sugere a adição de técnicas de inspeção de conformidade, que seriam adequadas no contexto de investimento em testes internos. Ele considera que os testes de performance contribuem pouco nesse cenário específico. Por sua vez, o entrevistado 3 concorda plenamente com as técnicas propostas, mas sugere uma

alteração na ordem, colocando o teste exploratório antes do teste de progressão, caso a ordem de realização das tarefas seja relevante. Embora haja divergências em relação a adição de técnicas de inspeção de conformidade e a ordem dos testes exploratório e de progressão, todos os entrevistados concordam com a eficácia geral das técnicas de avaliação propostas para o contexto em questão.

A análise no Cenário 10 mostrou ser condizente e com uma das melhores avaliações entre os entrevistados. 33% (1 participante) avaliou com nota 5, 66% (2 participantes) com nota 4 na escala de avaliação.

Figura 22 – Cenário 10 do formulário

Quantidade de Testadores	Mais de 20
Experiência em Testes	Equilíbrio entre Júnior, Pleno e Sênior
Investimento Geral em Testes	Entre R\$ 51.000 e R\$ 100.000
Área mais relevante para investimento	Testes internos
Cronograma de Testes	Mais que três meses
Principais Gêneros	Aventura e Puzzle
Público Alvo	Jogadores de Jogos Sociais
Plataforma	PlayStation
Aspecto de Usabilidade a ser testado	Conformidade
Técnicas de Teste Recomendadas	<ol style="list-style-type: none"> 1. Teste de Progressão, 2. Teste de Performance, 3. Teste de Aceitação, 4. Teste Exploratório.

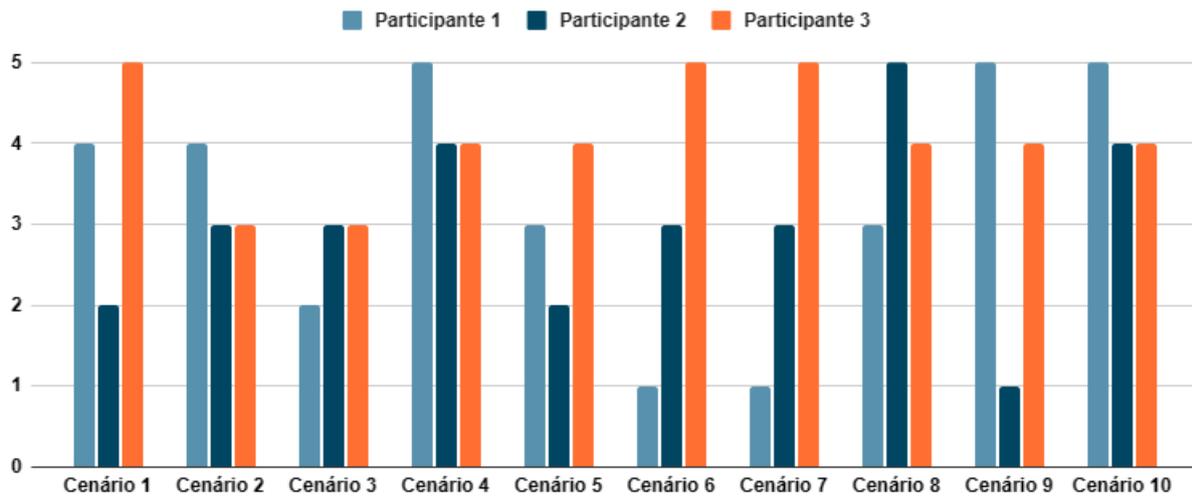
Considerando as análises dos três participantes a respeito dos 10 cenários apresentados, percebe-se que há uma concordância geral de que a escolha e a aplicação de técnicas de avaliação são importantes durante a fase de testes em jogos digitais. As opiniões divergem em relação às técnicas específicas mais adequadas para cada contexto, ordem de aplicação e quantidade de testadores necessários.

No geral, as opiniões convergem para a relevância de técnicas como testes de usabilidade, testes de aceitação, testes exploratórios e inspeções de conformidade. Essas técnicas são consideradas eficazes para garantir a conformidade com requisitos, identificar problemas de usabilidade, coletar *insights* valiosos dos usuários e melhorar a experiência geral do jogo.

No entanto, também existem discordâncias sobre a importância de algumas técnicas, como testes de regressão e testes de performance, que foram considerados menos relevantes em determinados contextos. Além disso, houve sugestões por parte do entrevistado 3 para realização de ajustes na ordem de aplicação das técnicas, considerando critérios como experiência dos testadores e especificidades do jogo. O mesmo participante afirma que a ordem da aplicação é relevante devido ao resultado gerado de uma técnica previamente utilizada durante os testes. Por exemplo, no Cenário 3, o participante sugere a alteração da ordem dos testes. O cenário sugere Teste de Aceitação, Teste Exploratório

e Teste Automatizado, onde o entrevistado sugere a mudança para Teste Automatizado, Exploratório e Aceitação, argumentando que “tentaria garantir o menor número de *bugs* para que o usuário final tenha uma boa experiência e me retorne um bom feedback”. A Figura 23 apresenta o resumo das avaliações dos três participantes em cada cenário, respectivamente.

Figura 23 – Avaliação de todos os cenários pelos participantes



Os cenários 1, 4, 8 e 10 receberam as notas mais altas, indicando que foram considerados pelos entrevistados como os cenários mais adequados em termos de técnicas de avaliação propostas. Esses cenários possuem médias acima de/ou próxima de 4, o que indica uma avaliação positiva, apresentando concordância geral dos três participantes com as técnicas de teste recomendadas. Os cenários 2 e 9 receberam notas intermediárias (por volta de uma média 3,3), indicando uma avaliação neutra, equilibrada e que houve pequenas discordâncias por parte dos entrevistados ou novas sugestões de aplicação de diferentes abordagens de teste ou alterações de ordem da utilização das técnicas. Já os cenários 3, 5, 6 e 7 receberam notas mais baixas, indicando que foram considerados pelos entrevistados como os cenários menos adequados. Esses cenários possuem médias de valor 3, o que indica uma avaliação menos positiva, onde mudanças nos cenários são recomendadas. A Tabela 10 abaixo apresenta o resumo das respostas de cada cenário pelos participantes 1, 2 e 3, respectivamente.

Tabela 10 – Resumo das respostas de cada participante

Cenário	Participante 1	Participante 2	Participante 3
1	Avaliação: 4. Concorda com as técnicas utilizadas, principalmente teste unitário e de regressão	Avaliação: 2. Sugere avaliações de usabilidade. Discorda com o uso das técnicas de teste unitário e de regressão	Avaliação: 5. Concorda com todas as técnicas recomendadas
2	Avaliação: 4. Concorda que teste de performance é o mais aplicável de acordo com o cenário	Avaliação: 3. Concorda com o uso de teste de usabilidade e progressão, analisando outras métricas a serem usadas. Discorda com o uso do teste de regressão.	Avaliação: 3. Concorda com as técnicas recomendadas, mas sugere uma troca da sua ordem.
3	Avaliação: 2. Concorda com o uso de teste de aceitação, mas discorda com o uso dos testes exploratórios e automatizados. Sugere o uso de teste <i>cleanroom</i>	Avaliação: 4. Sugere técnicas de inspeção e conformidade para detectar e corrigir problemas	Avaliação: 3. Sugere alteração de ordem dos testes para garantir menor quantidade de erros
4	Avaliação: 5. Concorda com as técnicas recomendadas, principalmente o teste unitário.	Avaliação: 4. Concorda com as técnicas recomendadas e sugere a aplicação de <i>playtesting</i> e testes exploratórios	Avaliação: 4. Concorda com as técnicas recomendadas, apesar do cronograma não ser condizente com a experiência dos testadores.
5	Avaliação: 3. Concorda que teste de aceitação é a melhor opção, mas possui dúvidas sobre o restante das recomendações	Avaliação: 2. Define que inspeções de conformidade e testes de sistema são essenciais ao contexto	Avaliação: 4. Concorda com as técnicas recomendadas, mas a quantidade de testadores poderia ser maior para o teste A/B ser mais efetivo.

6	<p>Avaliação: 1. Discorda com as recomendações e sugere que o <i>playtesting</i> seja a melhor abordagem, tendo em vista diferentes métricas</p>	<p>Avaliação: 3. Sugere a técnica de percurso cognitivo e a substituição de grupos focais por <i>playtesting</i>. Discorda com o uso de teste de regressão.</p>	<p>Avaliação: 5. Concorda com os testes recomendados, mas aponta que o cronograma não é muito favorável ao contexto.</p>
7	<p>Avaliação: 1. Discorda das técnicas recomendadas e sugere o <i>playtesting</i> como melhor abordagem no cenário.</p>	<p>Avaliação: 3. Sugere as técnicas de <i>playtesting</i> e técnicas formais. Discorda com o uso de teste de regressão.</p>	<p>Avaliação: 5. Concorda com as técnicas recomendadas mas sugere que a quantidade de testadores ou a experiência poderia ser um nível abaixo.</p>
8	<p>Avaliação: 3. Sugere a aplicação da técnica A/B onde os testadores teriam experiência Plena.</p>	<p>Avaliação: 5. Concorda com todas as técnicas recomendadas.</p>	<p>Avaliação: 4. Concorda com todas as técnicas mas sugere que a quantidade de testadores poderia ser maior.</p>
9	<p>Avaliação: 5. Concorda com as recomendações, especialmente com a técnica de teste de aceitação</p>	<p>Avaliação: 1. Afirma que inspeções de conformidade e testes de sistema são essenciais dado o contexto. Discorda no uso de testes unitários e de regressão.</p>	<p>Avaliação: 4. Concorda com as recomendações, mas argumenta que a experiência dos testadores poderia ser melhor.</p>
10	<p>Avaliação: 5. Concorda com todas as técnicas recomendadas.</p>	<p>Avaliação: 4. Sugere a adoção de técnicas de inspeção de conformidade. Discorda com o uso de teste de performance.</p>	<p>Avaliação: 4. Concorda com os testes, mas trocaria a ordem entre teste exploratório e de progressão.</p>

Em relação a aplicação preliminar desenvolvida, tendo em vista o resultado apresentado e a descrição das técnicas de teste definidas ao usuário, nota-se algumas vantagens da utilização do sistema. Ao fornecer um relatório personalizado com técnicas de teste adaptadas ao contexto do sistema e as características fornecidas pelo usuário, o *software* ajuda a equipe de desenvolvimento a tomar decisões mais informadas sobre as melhores práticas de teste a serem aplicadas. Isso economiza tempo e esforço ao evitar a necessidade de pesquisar e avaliar manualmente várias técnicas de teste. Além disso, com o uso adequado de técnicas de teste recomendadas, é mais provável que os problemas de qualidade do jogo sejam identificados e resolvidos de forma mais abrangente. Isso resulta em um produto final de maior qualidade, com menos erros e falhas, proporcionando uma experiência mais satisfatória aos jogadores. Outra vantagem foca na redução de retrabalho pela equipe de desenvolvimento, uma vez que as técnicas recomendadas sejam utilizadas desde o início do processo de desenvolvimento, o *software* ajuda a evitar retrabalhos e revisões significativas em estágios posteriores. Isso economiza tempo, recursos e evita atrasos no lançamento do jogo.

Devido a complexidade de se capturar as informações específicas fornecidas pelo usuário para gerar um relatório mais amplo e customizável ao contexto de seu jogo, o *software* preliminar apenas abrange a descrição das técnicas de teste e qual seus respectivos objetivos. Novas funcionalidades deverão ser implementadas em futuras atualizações, tendo em vista revisões e validações com outros profissionais da área.

5.4 Lições do Capítulo

Neste capítulo, foi possível entender como foi realizado o planejamento da avaliação proposta para este estudo, demonstrando as etapas definidas, a aplicação na prática, dificuldade de se obter participantes e, por fim, os resultados do formulário juntamente com a análise de cada um dos cenários propostos.

6 CONSIDERAÇÕES FINAIS

Neste Trabalho de Conclusão de Curso, abordou-se a área de controle de qualidade (CQ) em jogos digitais, levando em consideração os problemas críticos que grandes jogos digitais apresentam em suas versões finais. O foco em questão da pesquisa tratou em torno das técnicas de teste na etapa de CQ e sua efetividade. Buscou-se identificar e mapear as práticas de teste de acordo com os objetivos propostos, considerando a metodologia de pesquisa utilizada, tendo como base a atividade de busca nas bases de dados SCOPUS e *Google Scholar*, aplicando a técnica de *snowballing* e realizando a seleção de estudos por meio de critérios de inclusão, exclusão e de qualidade.

Com os resultados obtidos, é possível concluir que as empresas que desenvolvem jogos digitais estão cada vez mais preocupadas com a qualidade final do produto para satisfazer seus usuários da melhor forma possível. Aproveitar-se de um ciclo de desenvolvimento bem definido e com técnicas apropriadas de controle de qualidade auxilia a obter resultados satisfatórios, gerenciando os problemas encontrados e *feedback* recebido. Apesar disso, a etapa de controle de qualidade especificamente focada em jogos digitais não é bem definida na literatura, no sentido de oferecer o *background* necessário para entendimento dessa etapa para os desenvolvedores, que muitas vezes avaliam outros materiais focados apenas no desenvolvimento geral e genérico de *software*.

Existem diversas práticas de testes que estão sendo utilizadas em relação aos jogos digitais, no qual deve-se obter o contexto de qual é a melhor abordagem para o jogo específico que está sendo testado. Tendo isso em vista, a proposta deste Trabalho de Conclusão de Curso foca nesse entendimento e auxílio para equipes de desenvolvimento. Uma vez que haja suporte, considerando os pontos principais do projeto, este software de recomendação oferecerá maior robustez e segurança na implementação de casos de teste, práticas de teste, priorização do que deve ser testado e entendimento do sistema em relação aos recursos disponíveis.

Considerando os resultados da aplicação do formulário com profissionais das áreas de teste de *software* e/ou controle de qualidade em jogos digitais, percebe-se que não há uma concordância geral sobre quais técnicas de teste ou de inspeções devem ser utilizadas dado um cenário específico de desenvolvimento. Isto ocorre por uma série de razões, justamente por causa do cenário atual da indústria de jogos ser amplo, onde cada projeto possui inúmeras características distintas entre si, no qual a realidade interna varia entre diferentes empresas. Além disso, a experiência e conhecimento técnico de cada profissional na área colabora na variação da estimativa sobre o que é melhor para um determinado cenário e o que não é agradável ou possível de se utilizar. Muitas das respostas coletadas se basearam em experiências prévias, sejam aplicadas em casos reais ou meramente por dominarem um conhecimento específico, o que fornece uma noção do que realmente pode ser aplicado em outros cenários que compartilham de determinadas características. Tendo isso em vista, é difícil analisar um cenário de testes em um jogo digital, onde não existe

uma solução definitiva que solucione com altíssima acurácia o problema sobre quais técnicas de teste se devem utilizar, apenas fornecendo uma avaliação que mais se assemelha à realidade do projeto, já que as avaliações técnicas de profissionais da área variam em uma grande escala em vários quesitos. Por exemplo, os cenários 3, 5, 6 e 7, que obtiveram as menores médias atribuídas pelos entrevistados, podem ser avaliadas diferentemente por outro grupo de entrevistados que possuem um histórico e experiências distintas na etapa de testes.

A aplicação desenvolvida durante a escrita deste Trabalho de Conclusão de Curso possui o intuito de auxiliar a equipe de desenvolvimento a entender quais técnicas de teste são mais adequadas ao seu contexto. A validação das informações apresentadas ao usuário no relatório proveniente da execução do *software* foram validadas por meio das respostas coletadas do formulário com profissionais. Com essas informações, o desenvolvedor tem a possibilidade de ver a descrição das técnicas de teste e aplicá-las em seu projeto, fazendo assim que os recursos sejam alocados de melhor forma e o fluxo de teste seja otimizado em um todo.

Apesar disso, o *software* necessita de validação em alguns aspectos. No momento, o relatório final disponibilizado pelo sistema descreve apenas as técnicas de teste adaptadas ao contexto, onde, em um cenário ideal, as informações como análise detalhada de cada aspecto da etapa de testes deveriam ser apresentadas, fazendo assim que a proposta fosse mais efetiva, uma vez que esta análise é difícil de se realizar e demanda um conhecimento maior sobre a área e métodos reais de aplicação e cenários de teste. Outro tipo de informação interessante de se explorar seria a sugestão de adaptação do contexto pela ferramenta, tendo em vista as variáveis informadas pelo usuário. Como trabalho futuro, além desses aspectos já expostos, uma interface de usuário deve ser implementada para melhor usabilidade e entendimento do *software*, podendo ser disponibilizado de melhor forma aos usuários finais. Além disso, um *dataset* maior e com mais casos relevantes para treinamento da árvore de decisão deve ser explorado, uma vez que isso pode abranger mais casos e cenários contendo diferentes características do desenvolvimento de *software*.

Outro ponto a ser explorado no futuro é a questão do *software* poder ajudar a documentar e rastrear o processo de teste, permitindo que a equipe de desenvolvimento acompanhe e avalie o progresso em relação às técnicas de teste recomendadas. Isso oferece uma visão clara do trabalho realizado, facilitando a identificação de áreas que precisam de maior atenção. Para que as informações estejam sempre acuradas e precisas, o *software* pode ser atualizado e aprimorado continuamente com base no *feedback* e nas necessidades da equipe de desenvolvimento. Isso permite uma melhoria contínua dos processos de teste, garantindo que as melhores práticas sejam sempre aplicadas e que o *software* esteja alinhado com as mudanças no contexto do desenvolvimento de jogos.

REFERÊNCIAS

- ABRAN, A. et al. Consolidating the iso usability models. In: **Proceedings of 11th international software quality management conference**. [S.l.: s.n.], 2003. v. 2003, p. 23–25. Citado na página 43.
- ALBAGHAJATI, A.; AHMED, M. A co-evolutionary genetic algorithms approach to detect video game bugs. **Journal of Systems and Software**, Elsevier, v. 188, p. 111261, 2022. Citado na página 54.
- ALBAGHAJATI, A. M.; AHMED, M. A. K. Video game automated testing approaches: An assessment framework. **IEEE Transactions on Games**, IEEE, 2020. Citado 3 vezes nas páginas 40, 49 e 54.
- ALEEM, S.; CAPRETZ, L. F.; AHMED, F. Critical success factors to improve the game development process from a developer's perspective. **Journal of Computer Science and Technology**, Springer, v. 31, n. 5, p. 925–950, 2016. Citado 3 vezes nas páginas 40, 53 e 56.
- ALEEM, S.; CAPRETZ, L. F.; AHMED, F. Game development software engineering process life cycle: a systematic review. **Journal of Software Engineering Research and Development**, SpringerOpen, v. 4, n. 1, p. 1–30, 2016. Citado 5 vezes nas páginas 29, 30, 41, 53 e 57.
- ALEMM, S.; CAPRETZ, L. F.; AHMED, F. A consumer perspective on digital games. **IEEE Consumer Electronics**, v. 7, n. 3, p. 56, 2018. Citado na página 23.
- ALMANASREH, E.; MOLES, R.; CHEN, T. F. Evaluation of methods used for estimating content validity. **Research in social and administrative pharmacy**, Elsevier, v. 15, n. 2, p. 214–221, 2019. Citado na página 60.
- ALTEXSOFT. **How to Conduct User Acceptance Testing: Process Stages, Deliverables, and End-User Testing Place in Quality Assurance**. Altexsoft, 2021. Acesso em: jan. 2023. Disponível em: <<https://www.altexsoft.com/blog/engineering/user-acceptance-testing/>>. Citado na página 40.
- AMPATZOGLOU, A.; STAMELOS, I. Software engineering research for computer games: A systematic review. **Information and Software Technology**, Elsevier, v. 52, n. 9, p. 888–901, 2010. Citado na página 29.
- ANNANPERÄ, E. et al. Testing methods for mobile game development a case study on user feedback in different development phases. In: IEEE. **2018 IEEE 6th International Conference on Serious Games and Applications for Health (SeGAH)**. [S.l.], 2018. p. 1–8. Citado 2 vezes nas páginas 54 e 58.
- BERTOLINO, A. Software testing research: Achievements, challenges, dreams. In: IEEE. **Future of Software Engineering (FOSE'07)**. [S.l.], 2007. p. 85–103. Citado 2 vezes nas páginas 40 e 59.
- BETHKE, E. **Game development and production**. [S.l.]: Wordware Publishing, Inc., 2003. Citado 4 vezes nas páginas 41, 49, 53 e 56.

- BEVAN, N. et al. New iso standards for usability, usability reports and usability measures. In: SPRINGER. **Human-Computer Interaction. Theory, Design, Development and Practice: 18th International Conference, HCI International 2016, Toronto, ON, Canada, July 17-22, 2016. Proceedings, Part I 18.** [S.l.], 2016. p. 268–278. Citado 2 vezes nas páginas 42 e 43.
- BOEHM, B. W. Verifying and validating software requirements and design specifications. **IEEE software**, IEEE Computer Society, v. 1, n. 1, p. 75, 1984. Citado na página 23.
- BOURQUE, P.; FAIRLEY, R. E. Swebok, version 3.0: Guide to the software engineering body of knowledge. **Piscataway (NJ, USA): IEEE**, 2014. Citado na página 36.
- BURNSTEIN, I. Practical software testing: A process-oriented approach. In: **Springer Science & Business Media**. [S.l.: s.n.], 2003. Citado 9 vezes nas páginas 34, 35, 37, 38, 40, 49, 54, 57 e 59.
- CHANDLER, H. M. **The game production handbook**. [S.l.]: Jones & Bartlett Publishers, 2009. Citado 2 vezes nas páginas 31 e 32.
- CHO, J. Bughunting on a budget: Exploring quality assurance practices and tools for indie game developers. 2022. Citado 3 vezes nas páginas 35, 54 e 58.
- CHOI, J. O. et al. Playtesting with a purpose. In: **Proceedings of the 2016 annual symposium on computer-human interaction in play**. [S.l.: s.n.], 2016. p. 254–265. Citado 2 vezes nas páginas 53 e 57.
- COMMISSION, I. O. for S. E. et al. International standard iso/iec 9126-1: Software engineering–product quality–part 1: Quality model. **ISO/IEC, Tech. Rep**, 2001. Citado 2 vezes nas páginas 42 e 43.
- DASH, S. **Decision Trees Explained — Entropy, Information Gain, Gini Index, CCP Pruning**. Towards Data Science, 2022. Acesso em: jun. 2023. Disponível em: <<https://towardsdatascience.com/decision-trees-explained-entropy-information-gain-gini-index-ccp-pruning-4d78070db36c>>. Citado na página 46.
- DEMIRCIN, Y. **What is level design?** Mobidictum, 2022. Acesso em: jan. 2023. Disponível em: <<https://mobidictum.biz/level-design/>>. Citado na página 31.
- FABRICATORE, C.; NUSSBAUM, M.; ROSAS, R. Playability in action videogames: A qualitative design model. **Human-Computer Interaction**, Taylor & Francis, v. 17, n. 4, p. 311–368, 2002. Citado na página 41.
- FAHEY, R. **The costs of buggy game launches are mounting**. GameIndustri.biz, 2022. Acesso em: dez. 2022. Disponível em: <<https://www.gamesindustry.biz/the-costs-of-buggy-game-launches-are-mounting-opinion>>. Citado 2 vezes nas páginas 23 e 31.
- FISHER, R. A. The use of multiple measurements in taxonomic problems. **Annals of eugenics**, Wiley Online Library, v. 7, n. 2, p. 179–188, 1936. Citado na página 45.
- FULLERTON, T. **Game design workshop: a playcentric approach to creating innovative games**. [S.l.]: CRC press, 2014. Citado 3 vezes nas páginas 24, 39 e 41.

- GODOY, A.; BARBOSA, E. F. Game-scrum: An approach to agile game development. **Proceedings of SBGames**, p. 292–295, 2010. Citado na página 31.
- HAMILTON, T. **Game Testing: Types How to Test Mobile/Desktop Apps**. Guru99, 2022. Acesso em: jan. 2023. Disponível em: <<https://www.guru99.com/game-testing-mobile-desktop-apps.html>>. Citado na página 40.
- HOOVER, S. **Automated Testing and Validation of Computer Graphics Implementations for Cross-platform Game Development**. Tese (Doutorado) — Auckland University of Technology, 2017. Citado 2 vezes nas páginas 54 e 58.
- JALALI, S.; WOHLIN, C. Systematic literature studies: database searches vs. backward snowballing. In: **Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement**. [S.l.: s.n.], 2012. p. 29–38. Citado na página 27.
- JIMENEZ, I. et al. Quiho: Automated performance regression testing using inferred resource utilization profiles. In: **Proceedings of the 2018 ACM/SPEC International Conference on Performance Engineering**. [S.l.: s.n.], 2018. p. 273–284. Citado na página 37.
- KARTHIKEYAN, K. **Indie, AAA, and AA Games: A Comparison**. Gameopedia, 2022. Acesso em: jun. 2023. Disponível em: <<https://www.gameopedia.com/indie-aaa-aa-games-comparison/>>. Citado na página 36.
- KASURINEN, J.; SMOLANDER, K. What do game developers test in their products? In: **Proceedings of the 8th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement**. [S.l.: s.n.], 2014. p. 1–10. Citado 4 vezes nas páginas 25, 53, 57 e 61.
- KLINE, S. et al. **Digital play: The interaction of technology, culture, and marketing**. [S.l.]: McGill-Queen’s Press-MQUP, 2003. Citado na página 23.
- KORHONEN, H. Comparison of playtesting and expert review methods in mobile game evaluation. In: **Proceedings of the 3rd International Conference on Fun and Games**. [S.l.: s.n.], 2010. p. 18–27. Citado 2 vezes nas páginas 53 e 57.
- KOTSIANTIS, S. B. Decision trees: a recent overview. **Artificial Intelligence Review**, Springer, v. 39, p. 261–283, 2013. Citado 2 vezes nas páginas 43 e 44.
- LAHTI, M. et al. **Game Testing in Finnish Game Companies**. Dissertação (Mestrado), 2015. Citado 10 vezes nas páginas 29, 30, 32, 38, 39, 40, 49, 53, 56 e 59.
- LI, Z. et al. Gbgallery: A benchmark and framework for game testing. **Empirical Software Engineering**, Springer, v. 27, n. 6, p. 1–27, 2022. Citado 2 vezes nas páginas 54 e 57.
- LIKERT, R. A technique for the measurement of attitudes. **Archives of psychology**, 1932. Citado na página 72.
- MACDONALD, K. **Cyberpunk 2077: how 2020’s biggest video game launch turned into a shambles**. The Guardian, 2020. Acesso em: dez. 2022. Disponível em: <<https://www.theguardian.com/games/2020/dec/18/>>

- cyberpunk-2077-how-2020s-biggest-video-game-launch-turned-into-a-shambles>. Citado na página 23.
- MANNINEN, T. et al. Game production process—a preliminary study. **University of Oulu, Finland**, 2006. Citado na página 31.
- MCV. **Quality Control: Understanding your QA team can only improve your game**. MCV Develop, 2019. Acesso em: dez. 2022. Disponível em: <<https://www.mcvuk.com/development-news/quality-control-understanding-your-qa-team-can-only-improve-your-game/>>. Citado 2 vezes nas páginas 23 e 35.
- MIRZA-BABAEI, P.; MOOSAJEE, N.; DRENIKOW, B. Playtesting for indie studios. In: **Proceedings of the 20th International Academic Mindtrek Conference**. [S.l.: s.n.], 2016. p. 366–374. Citado 2 vezes nas páginas 53 e 57.
- MIRZA-BABAEI, P. et al. A postmortem on playtesting: Exploring the impact of playtesting on the critical reception of video games. In: **Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems**. [S.l.: s.n.], 2020. p. 1–12. Citado 2 vezes nas páginas 54 e 58.
- MOZGOVOY, M.; PYSHKIN, E. A comprehensive approach to quality assurance in a mobile game project. In: **Proceedings of the 14th Central and Eastern European Software Engineering Conference Russia**. [S.l.: s.n.], 2018. p. 1–8. Citado 3 vezes nas páginas 38, 54 e 58.
- MURPHY-HILL, E.; ZIMMERMANN, T.; NAGAPPAN, N. Cowboys, ankle sprains, and keepers of quality: How is video game development different from software development? In: **Proceedings of the 36th International Conference on Software Engineering**. [S.l.: s.n.], 2014. p. 1–11. Citado 3 vezes nas páginas 35, 54 e 57.
- NEGRÃO, P. M. **Automated Playtesting In Videogames**. Tese (Doutorado), 2020. Citado 6 vezes nas páginas 37, 39, 49, 54, 58 e 59.
- NETO, J. N. de O. et al. Is there time for software testing in the indie games development? a survey with practitioners of the game industry. In: **Proceedings of the XXXIII Brazilian Symposium on Software Engineering**. [S.l.: s.n.], 2019. p. 37–46. Citado 6 vezes nas páginas 25, 30, 39, 53, 56 e 59.
- NILSSON, F.; NILSSON, J. **Comparing Automated Testing Approaches for FPS Games**. 2021. Citado 3 vezes nas páginas 40, 49 e 54.
- NOVAK, J. **Game development essentials: an introduction**. [S.l.]: Cengage Learning, 2011. Citado 10 vezes nas páginas 30, 31, 32, 33, 34, 40, 49, 53, 56 e 59.
- PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011. Citado 3 vezes nas páginas 44, 45 e 46.
- PIETRIKOVÁ, E.; SOBOTA, B. Game development and testing in education. IntechOpen, 2022. Citado 2 vezes nas páginas 54 e 58.
- PINELLE, D.; WONG, N.; STACH, T. Using genres to customize usability evaluations of video games. In: **Proceedings of the 2008 conference on future play: Research, play, share**. [S.l.: s.n.], 2008. p. 129–136. Citado na página 42.

- POLITOWSKI, C.; GUÉHÉNEUC, Y.-G.; PETRILLO, F. Towards automated video game testing: Still a long way to go. **arXiv preprint arXiv:2202.12777**, 2022. Citado 2 vezes nas páginas 49 e 54.
- POLITOWSKI, C.; PETRILLO, F.; GUÉHÉNEUC, Y. A survey of video game testing. in 2021 ieee. In: **ACM International Conference on Automation of Software Test (AST)**. [S.l.: s.n.], 2021. p. 90–99. Citado na página 35.
- PRESSMAN, R. S. **Software engineering: a practitioner's approach**. [S.l.]: Palgrave Macmillan, 2005. Citado na página 34.
- RACHEVA, S. Testing practices in indie game development from a software engineering perspective: an exploratory study. 2019. Citado 3 vezes nas páginas 36, 54 e 58.
- RAJ, A. **An Exhaustive Guide to Decision Tree Classification in Python 3.x**. Towards Data Science, 2021. Acesso em: jun. 2023. Disponível em: <<https://towardsdatascience.com/an-exhaustive-guide-to-classification-using-decision-trees-8d472e77223f>>. Citado 3 vezes nas páginas 43, 44 e 45.
- RAJANEN, M.; TAPANI, J. A survey of game usability practices in north american game companies. 2018. Citado 2 vezes nas páginas 54 e 58.
- RAMADAN, R.; HENDRADJAYA, B. Development of game testing method for measuring game quality. In: IEEE. **2014 International Conference on Data and Software Engineering (ICODSE)**. [S.l.], 2014. p. 1–6. Citado 5 vezes nas páginas 38, 41, 54, 57 e 59.
- RAMADAN, R.; WIDYANI, Y. Game development life cycle guidelines. In: IEEE. **2013 International Conference on Advanced Computer Science and Information Systems (ICACISIS)**. [S.l.], 2013. p. 95–100. Citado 2 vezes nas páginas 54 e 59.
- RUUSKA, E. Quality assurance testing on video games: The importance and impact of a misunderstood industry. Tampereen ammattikorkeakoulu, 2015. Citado 3 vezes nas páginas 35, 37 e 59.
- SALAZAR, M. G. et al. Proposal of game design document from software engineering requirements perspective. In: IEEE. **2012 17th International Conference on Computer Games (CGAMES)**. [S.l.], 2012. p. 81–85. Citado na página 30.
- SCHATTEN, M. et al. Towards an agent-based automated testing environment for massively multi-player role playing games. In: IEEE. **2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)**. [S.l.], 2017. p. 1149–1154. Citado 2 vezes nas páginas 49 e 54.
- SCHULTZ, C. P.; BRYANT, R. D. **Game testing: All in one**. [S.l.]: Mercury Learning and Information, 2016. Citado 9 vezes nas páginas 29, 30, 31, 32, 38, 40, 49, 53 e 56.
- SINGH, R. **Why is Usability Crucial for a Good Gaming Experience?** Headspin, 2022. Acesso em: jun. 2023. Disponível em: <<https://www.headspin.io/blog/usability-for-good-gaming-experience#:~:text=Game%20developers%20perform%20usability%20testing,for%20various%20kinds%20of%20designs.>> Citado 2 vezes nas páginas 41 e 42.

- SIPPOLA, T. **Usability is a key element of User Experience**. Landis+Gyr, 2017. Acesso em: jun. 2023. Disponível em: <<https://eu.landisgyr.com/better-tech/usability-is-a-key-element-of-user-experience#:~:text=While%20the%20ISO%20definition%20has,%2C%20memorability%2C%20errors%20and%20satisfaction.>> Citado 2 vezes nas páginas 42 e 43.
- SOUSA, M. A. et al. Aspectos de qualidade em jogos sérios digitais. **Renote**, v. 19, n. 1, p. 207–216, 2021. Citado na página 41.
- SYED, M. Quality control in games development. 2019. Citado 10 vezes nas páginas 26, 31, 32, 34, 35, 37, 38, 49, 53 e 55.
- TASSI, P. **Bethesda’s Silence Of The State Of ‘Fallout 76’ At Launch Is Deafening [Update]**. Forbes, 2018. Acesso em: dez. 2022. Disponível em: <<https://www.forbes.com/sites/insertcoin/2018/11/27/bethedas-silence-of-the-state-of-fallout-76-at-launch-is-deafening/?sh=2e88ad0861e7>>. Citado na página 23.
- UNITY. **Intro to A/B testing**. Unity, 2022. Disponível em: <<https://unity.com/how-to/ab-testing-games>>. Citado 2 vezes nas páginas 38 e 59.
- VARGAS, J. A. et al. A systematic mapping study on serious game quality. In: **Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering**. [S.l.: s.n.], 2014. p. 1–10. Citado 2 vezes nas páginas 24 e 49.
- WHITTAKER, J. A. **Exploratory software testing: tips, tricks, tours, and techniques to guide test design**. [S.l.]: Pearson Education, 2009. Citado 2 vezes nas páginas 38 e 59.
- WOOG, B. **Looking At How Far No Man’s Sky Has Come**. Game Informer, 2019. Acesso em: dez. 2022. Disponível em: <<https://www.gameinformer.com/2019/07/15/looking-at-how-far-no-mans-sky-has-come>>. Citado na página 23.
- YADAV, P. **Decision Tree in Machine Learning**. Towards Data Science, 2018. Acesso em: jun. 2023. Disponível em: <<https://towardsdatascience.com/decision-tree-in-machine-learning-e380942a4c96>>. Citado 3 vezes nas páginas 43, 44 e 47.
- ZHENG, Y. et al. Wuji: Automatic online combat game testing using evolutionary deep reinforcement learning. In: IEEE. **2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)**. [S.l.], 2019. p. 772–784. Citado na página 40.