



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de São José do Rio Preto

Lucca Augusto Torres Tambor

**INTEGRAÇÃO DAS GESTÕES DE ESCOPO E DE PESSOAS PARA
COMBATER O *CRUNCH* NA INDÚSTRIA DE JOGOS DIGITAIS**

São José do Rio Preto
2022

Lucca Augusto Torres Tambor

**INTEGRAÇÃO DAS GESTÕES DE ESCOPO E DE PESSOAS PARA
COMBATER O *CRUNCH* NA INDÚSTRIA DE JOGOS DIGITAIS**

Trabalho de Conclusão de Curso (TCC) apresentado como parte dos requisitos para obtenção do título Bacharel em Ciência da Computação, junto ao Conselho de Curso de Bacharelado em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Orientadora: Prof^a. Dr^a. Rogéria Cristiane Gratão de Souza

São José do Rio Preto
2023

T155i

Tambor, Lucca Augusto Torres.

Integração das Gestões de Escopo e Pessoas para combater o crunch na Indústria de Jogos Digitais / Lucca Augusto Torres Tambor. -- São José do Rio Preto, 2022

53 f. : tabs.

Trabalho de conclusão de curso (Bacharelado - Ciência da Computação) -

Universidade Estadual Paulista (Unesp), Instituto de Biociências Letras e

Ciências Exatas, São José do Rio Preto

Orientadora: Rogéria Cristiane Gratão de Souza

1. Engenharia de Software. 2. Indústria de Jogos Digitais
3. Gestão de Projetos. I. Título.

Lucca Augusto Torres Tambor

Trabalho de Conclusão de Curso (TCC) apresentado como parte dos requisitos para obtenção do título Bacharel em Ciência da Computação, junto ao Conselho de Curso de Bacharelado em Ciência da Computação, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Câmpus de São José do Rio Preto.

Comissão Examinadora

Prof^a. Dr^a. Rogéria Cristiane Gratão de Souza
UNESP – Câmpus de São José do Rio Preto
Orientadora

Prof. Dr. Leandro Alves Neves
UNESP – Câmpus de São José do Rio Preto

Prof^a. Dr^a. Adriana Barbosa Santos
UNESP – Câmpus de São José do Rio Preto

São José do Rio Preto
15 de dezembro de 2022

AGRADECIMENTOS

À minha família, principalmente aos meus pais Cintia e Marcelo, que me auxiliaram durante toda minha vida, e principalmente nos últimos anos durante a árdua jornada da graduação.

A todos os meus amigos, que durante os últimos anos, juntos encaramos as duras realidades da graduação e do mundo.

Aos meus professores que auxiliaram minha formação, principalmente a minha orientadora professora Rogéria, que mesmo com todos os contratemplos, me auxiliou com toda a execução deste trabalho.

“Quando acha que as pessoas morrem? Quando são atingidas no coração com uma bala? ... Não. Quando têm uma doença incurável? Não! Pessoas morrem quando são esquecidas!”

Eiichiro Oda (One Piece, 2000)

RESUMO

Este trabalho teve como base uma análise da literatura referente ao desenvolvimento de jogos, mais especificamente sobre como as gestões de escopo e de recursos impactam negativamente um projeto quando feitas de forma precária e como isso leva ao *crunch* - o trabalho excessivo dos desenvolvedores. Logo, o objetivo deste trabalho foi encontrar as principais causas que levam ao *crunch*, principalmente em meio a gestão do projeto, e então, foi proposto maneiras de evitá-lo. Neste contexto, foi feito um levantamento em inúmeros *postmortens* de projetos de jogos já concluídos que sofreram com a ocorrência de *crunch*, junto também de uma análise sobre as suas consequências e os seus impactos, não apenas nas pessoas envolvidas, mas também na qualidade do projeto em si. Como resultado, um software foi proposto com o objetivo de melhorar a comunicação entre desenvolvedores e gerentes de projeto. Tal software foi então avaliado por um pequeno grupo de desenvolvedores que atuam na indústria de jogos digitais e a maioria dos participantes concorda que um melhor acompanhamento de possíveis problemas que começam a surgir no projeto e uma comunicação efetiva são a melhor forma de combater a ocorrência de novos problemas, incluindo o *crunch*.

Palavras-chave: Desenvolvimento de Jogos. Gestão de Escopo. Gestão de Recursos. *Crunch*.

ABSTRACT

This paper was based on an analysis of the literature about game development, more specifically, the scope and resources managements, its effects on the project when done poorly, and how these lead to the *crunch* - the overwork of the developers. The goal of this paper was to find the causes that generate *crunch*, mainly the ones in project management, and ways to avoid the need for overwork were proposed. In this context, a survey was done on many *postmortems* of finished game projects that suffered from the *crunch*, together with an analysis of the consequences of the *crunch*, not only in the people who do it but also in the quality of the project. As a result, software that aims to improve the communication between developers and project managers was developed. Such software was evaluated by a small group of developers that have experience in the game industry, and the majority of them agreed that better monitoring of possible problems that start to pop up on the project and communication about these are the best ways to fight the occurrence of other problems, including *crunch*.

Keywords: Game Development. Scope Management. Resources Management.
Crunch.

LISTA DE ILUSTRAÇÕES

Figura 1 - Modelo de Dados	26
Figura 2 – Ilustração da Tela de Login	27
Figura 3 – Ilustração da Tela Inicial Desenvolvedor	28
Figura 4 – Ilustração da Tela de Projetos do Desenvolvedor	28
Figura 5 – Ilustração da Tela de Detalhes do Projeto do Desenvolvedor	29
Figura 6 – Ilustração da Tela de Time do Desenvolvedor	30
Figura 7 – Ilustração da Janela de Criação de Ocorrência	30
Figura 8 – Ilustração da Janela de Confirmação de Logout	31
Figura 9 – Ilustração da Tela Inicial do Gerente	31
Figura 10 – Ilustração da Tela de Desenvolvedores	32
Figura 11 – Ilustração da Janela de Criação de Novo Desenvolvedor	32
Figura 12 – Ilustração da Tela de Projetos do Gerente	33
Figura 13 – Ilustração da Janela de Criação de Novo Projeto	33
Figura 14 – Ilustração da Tela de Detalhes do Projeto do Gerente	34
Figura 15 – Ilustração da Janela de Criação de Novo Time	34
Figura 16 – Ilustração da Tela de Time do Gerente (Membros do Time)	35
Figura 17 – Ilustração da Janela para Adicionar Desenvolvedor ao Time	35
Figura 18 – Ilustração da Tela de Tarefas atribuídas ao Time	36
Figura 19 – Ilustração da Janela de Criação de Nova Tarefa	36
Figura 20 – Ilustração da Tela de Estatísticas (Gráficos)	37

Figura 21 – Ilustração da Tela de Estatísticas (Ocorrências por Time)	37
Figura 22 – Idades dos Participantes da Pesquisa	38
Figura 23 – Área de Atuação dos Participantes no Desenvolvimento de Jogos	39

LISTA DE TABELAS

Tabela 1 – Áreas do Conhecimento da Gerência de Projetos	14
Tabela 2 – Processos do Gerenciamento de Escopo.	15
Tabela 3 – Problemas Encontrados nos Projetos de Desenvolvimento de Jogos.	21

LISTA DE ABREVIATURAS E SIGLAS

CRT *Cathode Ray Tube*

API *Application Programming Interface*

SUMÁRIO

CAPÍTULO 1. INTRODUÇÃO	8
1.1. Contexto	8
1.2. Justificativa e Motivação	9
1.3. Objetivo	11
1.4. Organização da Monografia	11
CAPÍTULO 2. ANÁLISE DAS GESTÕES DE ESCOPO E RECURSOS	13
2.1. Gestão de Projetos	13
2.2. Gestão de Escopo	13
2.3. Gestão de Recursos	16
2.4. Impacto das Gestões de Escopo e Recursos na Indústria de Jogos Digitais	17
2.5. Crunch como Consequência da Má Gestão	20
2.6. Trabalhos Relacionados	23
CAPÍTULO 3. CONSTRUÇÃO DO SOFTWARE - GAMEJECT	25
3.1. Conceito do Software	25
3.2. Modelo de Dados	25
3.3. Tecnologias Utilizadas	26
3.4. Funcionalidade do Software	27
3.4.1. Uso do Software pelo Desenvolvedor	28
3.4.2. Uso do Software pelo Gerente	31
CAPÍTULO 4. AVALIAÇÃO	38

4.1. Público Alvo da Pesquisa	38
4.2. Resultados da Pesquisa	38
CAPÍTULO 5. CONCLUSÃO	41
5.1. Considerações Finais	41
5.2. Contribuições	41
5.3. Trabalhos Futuros	42
REFERÊNCIAS	43
APÊNDICE A - Questionário da Pesquisa	45

CAPÍTULO 1 - INTRODUÇÃO

1.1. Contexto

A indústria de jogos digitais pode ser descrita como uma área híbrida entre tecnologia, entretenimento e produção artística, que surgiu nos anos 60 como um passatempo de programadores de universidades criando simulações em monitores CRT – *Cathode Ray Tube* de osciloscópios (WOLF, 2007). Desde então, tal indústria vem evoluindo junto com a computação em termos de hardware e software, mais ainda nos anos durante a pandemia do COVID-19 (STATISTA, 2021), chegando no patamar das grandes indústrias do entretenimento.

Apesar de ser naturalmente uma extensão da computação gráfica, a produção de jogos digitais se diferencia substancialmente da produção de softwares tradicionais. Devido a sua natureza artística, a produção e desenvolvimento de um jogo não se limita apenas a programadores, engenheiros e outros cargos comuns na produção de softwares tradicionais, mas também há desenhistas, músicos, *game designers* e outros cargos voltados para a parte criativa (SCHREIER, 2017). Isso torna a produção de jogos, muitas vezes, uma tarefa muito subjetiva, afinal não é possível determinar de uma forma direta o que é bom ou não de forma artística. A mesma coisa pode ser dita sobre a diversão que um jogo digital proporciona, não há métricas que funcionem para todas as situações, pois cada caso é único (POLITOWSKI et al., 2021).

Outro fator que torna o desenvolvimento de jogos turbulento é o aumento massivo do escopo dos projetos de jogos nos últimos anos, onde um projeto deve ser sempre maior que o anterior para suprir as demandas do mercado, o que torna um projeto desafiador aos desenvolvedores e empresas, tanto para as que querem entrar na indústria, como para as que já estão consolidadas (ALEMM et al., 2016).

Problemas técnicos como *bugs*, projeto não podendo ser traduzido para o desenvolvimento, mudança na tecnologia de desenvolvimento durante o projeto, prototipação e documentação feitas de forma precária ou não sendo feitas, problemas no uso de ferramentas, como *engines* gráficas e gerenciadores de bancos de dados, e problemas com as unidades de controle de qualidade, sempre existiram (POLITOWSKI et al., 2021) e eram constantes. Junto ao crescimento da indústria de jogos e aperfeiçoamento das técnicas e ferramentas de

desenvolvimento, problemas de gerenciamento de projeto, *stakeholders*, negócios e *marketing* foram os que mais aumentaram em quantidade nos últimos anos (POLITOWSKI et al., 2021). Entre esses problemas, é possível destacar as dificuldades na realização das gestões de escopo e de pessoas nos projetos de jogos digitais como os principais sintomas de um projeto com problemas. Uma gestão precária realizada sem consciência sobre a capacidade da empresa, pode levar a custos extras, mudanças não planejadas no produto, corte de recursos em algumas áreas e falta de tempo para a entrega (AL-RUBAIEI et al., 2018).

Um dos principais problemas na gestão de pessoas, causado pela turbulência do desenvolvimento de jogos, é o período de horas extras de trabalho, muitas vezes não remuneradas, que os membros de uma equipe de desenvolvimento de jogos ficam sujeitos, o que é conhecido como *crunch*. Esta é uma prática que se tornou comum na indústria dos jogos, mesmo com consequências negativas aos funcionários, e continua sendo utilizada por empresas (COTE; HARRIS, 2020). Diferentes ocorrências podem acarretar a prática do *crunch* que, infelizmente, algumas vezes já é intrínseca à cultura de trabalho da empresa: um *bug* descoberto perto do prazo de entrega, um recurso que não funciona como esperado, *playtesters* relatando problemas resultantes de projeto inadequado ou uma total mudança de escopo de projeto, ou seja, qualquer problema ou conflito durante a produção do jogo.

1.2. Justificativa e Motivação

A prática de *crunch* ocorre em todas as indústrias de tecnologia e de desenvolvimento de software, mas há um destaque maior na literatura em relação ao *crunch* na indústria de jogos digitais devido a inúmeros escândalos envolvendo grandes estúdios na última década (SCHREIER, 2017). Uma particularidade da indústria de jogos digitais que auxiliou na descoberta dos casos de *crunch* é a prática comum entre desenvolvedores de jogos de compartilhar informações sobre como se desenrolou um certo projeto que participaram, o que é chamado de *postmortems*: relatos de desenvolvedores e líderes de projeto após o término de um jogo, onde contam os aspectos positivos e negativos da jornada de desenvolvimento (POLITOWSKI et al., 2021).

No trabalho de Politowski et al. (2021) foi criada uma base de dados com 200 *postmortems* de projetos de desenvolvimento de jogos realizados desde o ano de 1997 até 2019, contemplando projetos de escopo e resultados diferentes. Dessa base foram identificados os problemas que mais afligiam estes projetos, podendo se observar que quando um problema ocorre, ele acaba por gerar inúmeros outros que começam a surgir em consequência do primeiro.

Segundo Schreier (2017), muitas empresas, ao enfrentarem um ou mais problemas de desenvolvimento, optam por fazer seus desenvolvedores trabalharem mais para compensar os imprevistos. Já Edholm et al. (2017) destacam que, anos atrás, ainda se discutia na literatura se essa prática era algo inevitável, de forma que sem ela projetos teriam que ser alterados, atrasados ou até mesmo cancelados, e que sua existência possuía mais benefícios que desvantagens. Porém, os autores ainda observam que, com o passar do tempo, mais estudos foram realizados sobre o assunto, mostrando os resultados negativos causados pelo *crunch*, não só na integridade e saúde dos desenvolvedores, como também na qualidade do projeto em si e até mesmo na própria empresa. É possível observar casos de estúdios que passaram por um período de *crunch* em um projeto e, logo após a entrega e finalização do projeto em questão, muitos desenvolvedores acabaram saindo do estúdio devido às dificuldades que passaram no período em que realizaram *crunch*. Isso faz com que o estúdio tenha que contratar novos profissionais juniores, o que fará com que os próximos projetos acabem passando por problemas adicionais devido a falta de experiência da equipe (SCHREIER, 2017).

Observa-se que as maiores consequências vindas de períodos de *crunch* prolongados estão na saúde e integridade dos desenvolvedores. Na literatura é comum encontrar relatos de pessoas sobre o *crunch* e ambientes abusivos na indústria de jogos digitais, onde a saúde física e/ou mental dos desenvolvedores é sacrificada em decorrência de longos períodos de trabalho extra em um projeto que está passando por problemas. “Finais de semana se tornaram para nós uma coisa que apenas outras pessoas usufruíam” é um desses relatos descritos em Cote; Harris (2020).

Lin; Cheng; Jiang (2020) relatam sobre a preocupação do governo de Taiwan após um aumento preocupante no número de mortes relacionadas ao trabalho excessivo no país. O estudo mostra como não apenas o trabalho em si, mas os

ambientes tóxicos gerados por este, tem impactos negativos sobre os trabalhadores e, conseqüentemente, sobre a produção.

Diante do exposto, tem-se como principal justificativa do presente trabalho a grande quantidade de casos de *crunch* que cada vez mais são expostos na indústria de jogos digitais, e como esses casos afetam os projetos de jogos, os desenvolvedores desses e a indústria como um todo. Logo, tem-se como motivação contribuir para o combate ao *crunch*.

1.3. Objetivo

O objetivo geral deste trabalho é contribuir para uma gestão efetiva do escopo dos projetos de jogos digitais e das pessoas envolvidas em seu desenvolvimento com o intuito de permitir a identificação precoce de problemas que levariam à prática do *crunch* e, conseqüentemente, contribuir para que tal prática seja evitada.

Para tanto, como objetivos específicos tem-se:

- Identificação dos principais problemas que afetam os projetos de jogos digitais: isso possibilita analisar suas causas e conseqüências que levam ao *crunch*;
- Desenvolvimento de um *software* capaz de permitir o acompanhamento efetivo do escopo do projeto de jogos digitais e das pessoas envolvidas: tal *software* prevê a coleta de dados de funcionários sobre a realização de um projeto, tendo como base as atividades que exercem junto ao projeto, e a geração de informações que possibilitem a tomada de decisões que evitem a ocorrência do *crunch*, tanto para o projeto analisado quanto para outros futuros.

1.4. Organização da Monografia

Além do presente capítulo introdutório, tem-se:

- Capítulo 2: é descrita a importância da gestão de projetos em um âmbito geral para, posteriormente, apresentar uma análise mais específica voltada para gestão de escopo e gestão de recursos, com foco em pessoas. Para tanto, esses dois conceitos são abordados separadamente, enfatizando como cada um pode afetar um projeto de *software*. Na sequência, os dois tipos de gestão são revistos sobre o contexto da indústria de jogos digitais, evidenciando como a má realização destes prejudica os projetos na indústria;

- Capítulo 3: é feita uma descrição detalhada sobre o software desenvolvido, por meio de seus modelos e telas, evidenciando os diferentes perfis de usuários que o utilizarão, e quais funcionalidades cada um desses perfis têm acesso;

- Capítulo 4: é descrita a pesquisa avaliativa realizada junto ao software, qual foi o público-alvo da pesquisa, como ela foi divulgada, além dos resultados obtidos;

- Capítulo 5: é a conclusão do trabalho, destacando os principais pontos e resultados levantados, além de uma seção sobre trabalhos futuros.

CAPÍTULO 2 - ANÁLISE DAS GESTÕES DE ESCOPO E RECURSOS

2.1. Gestão de Projetos

Problemas durante o andamento de um projeto sempre ocorreram e sempre ocorrerão, pois não é possível eliminar completamente a chance de sua ocorrência na duração de um projeto, em razão das especificidades existentes nos projetos de software em geral que acarretam lidar com situações novas rotineiramente. Mas é possível tornar a estrutura de seu projeto e das partes envolvidas nele mais consistentes, de forma a minimizar a ocorrência desses problemas e, mesmo com o acontecimento de um problema, saber como lidar com este sem prejuízos impactantes ao projeto. Esse é um dos principais objetivos da área de gestão de projetos.

Neste trabalho, os conceitos adotados sobre gestão de projetos têm como base o *Guia do Conhecimento em Gerenciamento de Projetos* (PMI, 2017), um manual de gerenciamento de projetos e seus aspectos envolvidos criado pelo *Project Management Institute* - PMI, uma instituição internacional sem fins lucrativos composta por profissionais da área de gestão de projetos de todo o mundo.

O PMI (2017) define as metodologias básicas necessárias para a construção de um plano de gerenciamento de projeto, além das dez áreas de conhecimento, resumidas na Tabela 1, que se deve focar.

Segundo Politowski et al. (2021) é possível observar que a maioria das causas de problemas em projetos de jogos digitais envolvem o gerenciamento de Escopo e de Pessoas.

2.2. Gestão de Escopo

Segundo o PMI (2017), a área de gerenciamento de escopo contempla os processos necessários para garantir que um projeto inclua todo o trabalho necessário e apenas o trabalho necessário, para ser completado com sucesso. O gerenciamento de escopo é também a definição do que será e o que não será incluído no projeto.

Escopo pode se referir tanto ao escopo do produto, que compreende as características e funções que compõem o produto final, além de escopo do projeto,

que seria o trabalho realizado para entregar o produto com as características e funções especificadas (PMI, 2017).

Tabela 1 – Áreas do Conhecimento da Gerência de Projetos.

Área do Conhecimento	Descrição
Integração	Processo de união de todas as partes de um projeto de forma que estas não conflitem entre si
Escopo	Processo utilizado para garantir que apenas o trabalho necessário seja realizado no projeto
Cronograma	Processo que estipula os prazos do projeto
Custo	Processo para estipulação de orçamentos e controle de recursos financeiros do projeto
Qualidade	Processo para controle da qualidade do produto a ser entregue, estipulando métricas e testes para a garantia disso
Recursos	Processo para gerenciamento dos recursos físicos e humanos do projeto
Comunicação	Processo para assegurar que as informações do projeto serão geradas, documentadas e compartilhadas entre as partes do projeto
Riscos	Processo que visa a redução do impacto de imprevisto no projeto
Aquisições	Processo para controle da compra de materiais, serviços e produtos necessários para a execução do projeto
Partes Interessadas	Processo para identificação, planejamento e engajamento dos interessados no projeto

Fonte: Adaptado de PMI(2017).

Na literatura, pode-se encontrar inúmeras definições sobre como é possível realizar a construção do escopo de um projeto, quais as pessoas que devem estar envolvidas e quais situações adversas que podem ocorrer durante o projeto e que estas pessoas devem levar em conta ao planejar um escopo (KOMAL et al. ,2020).

A definição criada pelo PMI (2017) é feita dividindo o gerenciamento do escopo em seis processos, como mostrado na Tabela 2.

Tabela 2 – Processos do Gerenciamento de Escopo.

Processo	Descrição
Planejar o Gerenciamento do Escopo	Criação da documentação inicial do projeto, onde são descritos os objetivos e detalhes do projeto. Especificação de como o escopo será validado e controlado
Coletar os Requisitos	A lista de requisitos para a realização do projeto é feita de forma que atenda todas as necessidades do projeto e que não haja mudanças inesperadas durante este
Definir o Escopo	Uma descrição detalhada do projeto e suas características chaves que devem ser entregues ao fim do projeto. Tem como objetivo principal delimitar o que é possível e o que não é ao longo da duração do desenvolvimento
Criar a Estrutura Analítica de Projeto	Os componentes do projeto são divididos em elementos essenciais para garantir que todos os itens desejados estão envolvidos. Esses elementos representarão o escopo do trabalho de uma forma hierárquica, onde cada um terá sua estimativa de tempo, custo e recursos necessários para serem concluídos
Validar o Escopo	Os elementos essenciais são quantificados e testados baseados nos requisitos levantados no início do projeto. Esta é também a etapa onde o dono do projeto deve aceitá-lo formalmente
Controlar o Escopo	As mudanças e modificações no escopo do projeto são avaliadas com base no escopo base do projeto. Além disso, todo novo elemento adicionado deve ser avaliado com base no escopo atual do projeto, levando em conta os recursos e tempo restante ao projeto

Fonte: Adaptado de PMI (2017).

Todos estes processos são os passos que a gerência de um projeto de software deve ter em mente para que o desenvolvimento do projeto tenha o menor número de obstáculos possível. Ignorar a análise de escopo de um projeto é como ignorar a própria base deste. A falta de um escopo base, a adição de novas funções ao projeto durante o andamento deste sem uma análise prévia, e até mesmo subestimar os recursos necessários para a realização das funções básicas de um projeto podem ser chamados de *scope creep*.

Segundo Komal et al. (2020), *scope creep* é uma das maiores causas de falhas em projetos de software, levando a atrasos na entrega, requisitos inacabados sendo entregues, problemas críticos na versão final do produto que levam a baixa satisfação dos clientes finais.

É possível dividir os problemas com gestão de escopo de um projeto de software em: problemas que surgem durante o projeto, devido a imprevistos que não conseguem ser resolvidos, o que leva a perda de recursos; e problemas causados pela construção inicial do escopo de um projeto, onde a análise inicial foi precária, não levando em conta algum elemento do projeto ou dos *stakeholders* envolvidos nele (AL-RUBAIEI et al., 2018). Esse último tipo de problema de escopo é mais crítico devido a serem problemas na própria base do projeto. Assim, toda a documentação do projeto, objetivos, divisões de elementos e equipes já terão sido afetadas por esses assim que o projeto teve seu início. A falta de experiência de gestores ou má liderança dos financiadores do projeto, geralmente são os principais causadores destas situações (KOMAL et al., 2020).

2.3. Gestão de Recursos

O recurso humano é um dos mais importantes recursos que deve ser considerado durante o planejamento de um projeto, afinal são pessoas que irão desenvolver, testar e qualificar o produto sendo produzido.

Segundo o PMI (2017), a gestão de pessoas é uma das principais partes da Gestão de Recursos, e tem como etapas o planejamento, criação e gestão das equipes que trabalham em cada etapa do projeto.

No início de um projeto é necessário ser levado em conta a capacidade da equipe em relação ao planejamento do projeto para criação das equipes. Para tanto, deve ser considerada a quantidade de pessoas envolvidas no projeto, em quais tarefas essas podem ser alocadas segundo suas habilidades, a média de tempo em que essa quantidade de pessoas consegue realizar as tarefas necessárias para o projeto (PMI, 2017). Outro fator importante ao planejar as equipes de um projeto é a experiência de cada pessoa envolvida, na maioria das vezes equipes com um maior número de membros mais experientes possuem mais chances de terminar tarefas dentro do prazo ou contornar possíveis erros (ADOLFSEN, 2020). A definição de líderes é um exemplo de prática usada para ajustar a experiência de cada time.

A gestão de pessoas e times é feita durante toda a duração do projeto, visando garantir boas condições de trabalho para as pessoas envolvidas no projeto, para que possam se dedicar ao desenvolvimento dos objetivos estabelecidos. Isso envolve inúmeros fatores, como: ambiente de trabalho saudável, boas relações com as lideranças do projeto, infraestrutura necessária para realização das tarefas e uma gestão que se adapte a possíveis problemas ao longo do projeto para que os envolvidos não acabem por perder o foco nos objetivos necessários para a conclusão do projeto (ADOLFSEN, 2020).

Nos últimos anos, cada vez mais os estilos para gestão de equipes vêm se tornando menos focados em estruturas onde pessoas respondem aos seus superiores e assim por diante, formando uma cadeia de comando. Atualmente, tem-se maior ênfase na colaboração e suporte entre todos os membros de uma equipe, onde decisões não são tomadas por um único líder, mas sim tomadas em conjunto por todos os membros (PMI, 2017). Isso não significa que uma liderança é desnecessária ou obsoleta, pois um líder com uma carga de experiência é capaz de contribuir durante as tomadas de decisões de uma equipe em razão de seu conhecimento adquirido em projetos anteriores, além de guiar os menos experientes durante crises e problemas.

Como dito anteriormente, pessoas são um dos recursos mais importantes de um projeto, pois são elas que realmente realizam o projeto. Logo, uma crise na gestão desse recurso tão importante sempre irá causar graves problemas ao andamento de um projeto. Problemas com a gestão de pessoas muitas vezes surgem por consequência de outras deficiências na gestão do projeto, apesar de também poderem ocorrer isoladamente. Os envolvidos no projeto enxergam nos problemas ocasionados pela má gestão preocupações sobre seus cargos e papéis no projeto, o que pode levar esses à uma queda na produtividade ou até mesmo saída do projeto, o que obviamente afetará ainda mais a integridade do projeto e, dependendo do estado da equipe após o término do projeto em questão, a integridade de futuros projetos.

2.4. Impacto das Gestões de Escopo e Recursos na Indústria de Jogos Digitais

A análise do escopo de um projeto de jogo digital é, assim como em todo outro projeto, essencial, mas o que o difere são os fatores exclusivos desta indústria:

a complexidade do *software* de um jogo, as múltiplas equipes com funções totalmente diferentes (artística, técnica, narrativa) e a maneira como a indústria trata seus produtos, pois cada lançamento de produto deve ser maior e mais ambicioso que o anterior, principalmente em estúdios maiores e com altos investimentos financeiros (SCHREIER, 2017). Isso acaba gerando planejamentos de escopo ambiciosos demais ou irrealistas.

De forma complementar, mesmo quando o escopo não é ambicioso e inflado demais, ele ainda sim pode ser maior do que a capacidade da equipe. Um exemplo, retirado do *postmortem* do jogo *Bioshock* (FINLY, 2008), mostra que o projeto tinha o tempo inicial de dois anos de desenvolvimento, mas acabou levando três anos, 50% a mais do que o esperado. Durante o desenvolvimento, o projeto chegou a ter 93 desenvolvedores internos, 30 desenvolvedores terceirizados e 8 pessoas dedicadas apenas aos testes. Para o jogo foram produzidos 3.775 arquivos, com 758.903 linhas de código em C++ e 187.114 linhas de código *script*. Além disso, as equipes estavam separadas em locais diferentes, pois haviam desenvolvedores na América do Norte, Austrália e China trabalhando simultaneamente no projeto.

Assim como a gestão de escopo, a de pessoas se diferencia principalmente pelos perfis de profissionais específicos existentes em projetos de jogos digitais. De um lado é possível ver uma equipe de tecnologia, que tem seu funcionamento relativamente similar ao de uma equipe de desenvolvimento de qualquer outro projeto de software. Mas em contraponto, tem-se times de desenhistas, músicos, animadores e roteiristas que possuem rotinas de trabalho muitas vezes atípicas em relação às equipes de tecnologia.

Tais diferenças entre equipes podem levar a problemas na gestão, pois apesar do foco e objetivos gerais de todas as equipes serem os mesmos, as pequenas tarefas de cada equipe serão extremamente distintas (SCHREIER, 2017). Essas diferenças muitas vezes são o estopim para problemas comuns em qualquer projeto, mas tais problemas são exacerbados por elas nos projetos de jogos digitais, tais como problemas de comunicação, de objetivos muito distintos ao longo do projeto, falta de coesão entre o que é entregue pelos times e lideranças que possuem expertise em apenas uma das múltiplas áreas de conhecimento que compõem um projeto (POLITOWSKI et al., 2021). Com isso, é possível observar aumentos nestes problemas em projetos com mais equipes, onde cada uma delas necessita ser gerida de maneira diferente, e onde o tamanho destas equipes

também acaba se tornando um fator de complicação. Enquanto em estúdios independentes, os quais possuem recursos humanos limitados, muitas vezes pessoas realizam múltiplas tarefas em múltiplas funções diferentes, em uma estrutura de trabalho não organizada e hierárquica. Isso, apesar de gerar outros problemas, tornam problemas como os descritos nesta seção, causados por uma má gestão de recursos humanos, mais raros.

Não apenas na gestão de equipes, os projetos maiores têm problemas recorrentes. Uma prática comum na indústria atualmente é relatada por Schreier (2017), onde acionistas ou pessoas do alto escalão executivo de empresas colocam em cargos de lideranças, gestores que vieram de outros projetos bem sucedidos, mas de fora da indústria de jogos digitais. Logo, possuem a experiência de gestão, mas não das especificidades do projeto que irá gerir. Isso acaba levando projetos a situações em que problemas começam a ocorrer, turbulências são geradas e sentidas tanto pelas pessoas diretamente envolvidas no projeto, como por aqueles que investiram nele, os quais podem se sentir insatisfeitos com esses problemas e então tomarem decisões drásticas para tentar colocar o projeto de volta aos eixos, o que muitas vezes é ineficaz.

Um exemplo na indústria recente é o caso da Amazon Games relatado pelo portal de notícias Bloomberg (2021). A Amazon ingressou no mercado de jogos digitais na metade da década passada, comprando estúdios já estabelecidos e criando seus próprios, com o objetivo de colocar seu nome também neste mercado. Líderes que já haviam trabalhado em outros projetos internos da empresa foram colocados em posições altas nos novos estúdios. A expectativa era que um desses novos estúdios lançasse um produto que poderia ter alto impacto na indústria. Porém, em 2021 muitos destes estúdios foram fechados, muitas pessoas envolvidas naqueles projetos foram dispensadas e nenhum jogo foi lançado. Segundo relatos de pessoas que trabalharam na Amazon Games, um dos principais problemas desses projetos era que seus escopos e equipes eram enormes, mas poucos recursos foram alocados. Isso indica um menosprezo com as dificuldades de produzir um jogo, resultando no fracasso da referida empresa, uma das gigantes do mercado de tecnologia, na indústria de jogos digitais.

2.5. *Crunch* como Consequência da Má Gestão

As dificuldades acerca da produção de um jogo digital tendem a iniciar um efeito dominó, onde um problema no projeto acaba resultando em muitos outros surgindo em sequência. Politowski et al. (2020) dizem que muitos *postmortens* relatam o “problema inicial” durante o desenvolvimento. Porém, em razão de uma má gestão que dimensiona erroneamente o problema ou os recursos necessários para resolvê-lo, tal problema acaba afetando todas as áreas do projeto, podendo levá-lo a atrasos ou até mesmo mudanças completas de planejamento. Pode também ser observado que problemas inicialmente pequenos, ao serem menosprezados ou tratados de forma errada, podem acabar gerando um efeito cascata ocasionando inúmeros outros problemas.

Com a análise feita destes *postmortens*, é possível a confecção de perfis dos problemas mais comuns nesses projetos, como apresentado na Tabela 3. Com esses perfis, observa-se a relação existente entre os problemas envolvendo a produção de jogos: um problema pode ser o resultado de problemas anteriores que vão se acumulando até que, o que era um problema que envolvia um aspecto do projeto, se torne uma crise geral, podendo afetar a qualidade final do produto.

Um desses problemas que é a consequência de outros problemas anteriormente não resolvidos, é o *crunch*, trabalho além do expediente muitas vezes não remunerado. Ele se inicia com objetivo de resolver os obstáculos do projeto por meio de uma jornada de trabalho maior, onde os desenvolvedores trabalham além de seus horários contratuais, devendo passar mais horas no projeto a fim de consertar o que está quebrado ou corrigir partes que não funcionam mais. Normalmente o líder da equipe informa que o trabalho a mais será apenas durante determinado período de instabilidade do projeto, mas muitas vezes esse período se prolonga até o final do projeto (Cote; Harris, 2020).

Ao se revisar os *postmortens* de projetos que tiveram *crunch*, é possível observar já nos estágios iniciais destes, os sinais que levariam o projeto ao trabalho excessivo.

Escopos muito grandes com objetivos ambiciosos demais se comparados a projetos anteriores, a alocação de poucas pessoas ou poucos recursos a partes cruciais do projeto, com o fim de economizar gastos, e a fixação de uma data de

entrega muito precoce no projeto, são sinais que mostram os problemas que levariam aquele projeto ao *crunch*.

Tabela 3 – Problemas Encontrados nos Projetos de Desenvolvimento de Jogos.

Classificação	Descrição
<i>Bugs</i>	Comprometem o desenvolvimento do jogo ou sua recepção.
Problemas de <i>Game Design</i>	Mecânicas pouco divertidas, dificuldade na tradução para o desenvolvimento, ou balanceamento de mecânicas precário.
Documentação incompleta	Falta de documentação do código, de componentes ou do plano de desenvolvimento.
Prototipação mal feita	Precarização ou falta da fase de prototipação para validação de jogabilidade e recursos.
Problemas técnicos	Código mal escrito, infraestrutura precária, etc.
Testes precários	Falta de fase de controle de qualidade compatível com o produto.
Mal uso das ferramentas	Problemas com as ferramentas de desenvolvimento como <i>engines</i> , bibliotecas ou <i>frameworks</i> .
Comunicação precária	Problemas na comunicação entre os <i>stakeholders</i> , times, investidores, publicadora, etc.
Tempo de <i>crunch</i>	Desenvolvedores trabalhando horas a mais no projeto.
Problemas de equipe	Problemas na formação da equipe, falta de pessoal, ou perda de <i>stakeholders</i> durante o desenvolvimento.
Remoção de recursos	Remoção de recursos planejados devido a outros problemas.
Adição de recursos	Adição de recursos não planejados durante a produção.
Múltiplos projetos simultâneos	Quando há mais de um projeto sendo desenvolvido ao mesmo tempo e falta de investimento ou estrutura para suportar todos.
Orçamento precário	Orçamento limitado ou falta de investimento.
Planejamento	Planejamento ou cronograma feitos precariamente ou inexistentes.
Segurança falha	Recursos ou informações do projeto sendo vazadas.
Gestão de Escopo	Quando o projeto tem recursos demais, ou ambiciosos demais, o que leva a dificuldades de implementação.
Marketing	Problemas envolvendo o marketing e anúncio do jogo, que às vezes podem vender algo que ainda não está pronto.
Monetização	Problemas com o processo de geração de renda com o projeto.

“Eu sei que há estúdios que dizem que não praticam, mas eu acho que provavelmente estão mentindo”, é o que diz um desenvolvedor entrevistado em Edholm et al. (2017). Além deste relato, os autores incluem inúmeros relatos de *crunch* por desenvolvedores e ex-líderes de projetos para formar um perfil dos projetos que passaram por isto. Em seu trabalho é possível perceber que o *crunch* começa aos poucos dentro de um projeto. Muitas vezes, inicia com algumas pessoas tendo que trabalhar mais, o que começa a gerar uma certa pressão nas outras para trabalharem mais também pelo bem do projeto. Como resultado disso, aqueles que não se juntam a esse movimento acabam sendo taxados de culpados por não quererem auxiliar o projeto ao máximo. Isso acaba inserindo todos envolvidos no projeto nesta cultura de trabalho excessivo, que muitas vezes se encerra apenas com o término do desenvolvimento. Essa *cultura do crunch* é alimentada pelos próprios superiores do projeto que muitas vezes prometem bônus aos envolvidos e melhorias na infraestrutura dos estúdios. Isso gera o sentimento de missão a ser cumprida nos desenvolvedores, o que muitas vezes acaba sendo apenas promessas vazias, deixando os membros do projeto afetados física e psicologicamente.

Todo este ambiente de trabalho não é apenas prejudicial aos funcionários envolvidos no *crunch*, mas também ao próprio produto que será resultado deste projeto. Jogos digitais produzidos sobre um ambiente de *crunch*, na maioria dos casos, acabam por terem que ser lançados com alguma de suas funcionalidades não finalizadas devido a não conseguir mais sustentar o ambiente de trabalho. Jogos com *bugs* em grande quantidade, com mecânicas não testadas o suficiente e com recursos que não funcionam completamente, são comuns na indústria atualmente, pois a correlação entre esses problemas e o *crunch* não é feita pelo estúdio.

Aqueles que estavam envolvidos no projeto muitas vezes acabam saindo da empresa em busca de melhores condições. Isso deixa a situação dos projetos mais complicada, pois além de já ter que buscar novas pessoas para o projeto, experiência será algo em falta, o que tornará a gestão de outros projetos ainda mais desafiadora (EDHOLM et al., 2017).

2.6. Trabalhos Relacionados

Os efeitos do trabalho excessivo, não apenas na indústria de jogos digitais, vêm sendo estudados há anos na literatura. Uma pessoa que passa por um período de *crunch* intenso, trabalhando muitas horas a mais além de seu expediente, pode sofrer de síndrome de *burnout* (Síndrome do Esgotamento Profissional), um distúrbio emocional causado por situações de trabalho e esforço extremo (LIN et al., 2020). Pessoas saindo do estúdio direto para o hospital após dias sem dormir direito, noites em claro no trabalho sem retornar ao lar e finais de semanas em que era necessário o trabalho em equipe para terminar algum aspecto do projeto, são relatos comuns de desenvolvedores que trabalham em estúdios de jogos digitais (Cote; Harris, 2020).

A prática do *crunch* em estúdios maiores, onde há cadeias de comando mais claras além de uma maior espera de resultado pelos seus executivos, é diferenciada na literatura da prática adotada em estúdios independentes, muitas vezes compostos de poucas ou até uma única pessoa. Schreier (2017) relata o caso do projeto de *Stardew Valley* que foi realizado inteiramente por uma pessoa. O desenvolvedor descreve como foram os quatro anos do projeto, dado que ele não possuía nenhuma outra fonte de renda pessoal, sendo que apenas a partir da campanha de financiamento público, realizada no segundo ano do projeto, que começou a receber algo. Noites em claro e meses sem folga eram comuns no dia-a-dia do desenvolvimento, mas quando o projeto chegou ao fim, tornou-se um dos jogos independentes mais lucrativos da década. O *crunch* neste caso, mesmo ainda sendo não saudável e podendo ter gerado consequências na vida do desenvolvedor, foi inteiramente de livre escolha dele. O *crunch* que ocorre nas grandes empresas é uma imposição colocada sobre os desenvolvedores, eles, muitas vezes, não têm outra opção, e mesmo que as ordens para aumentar a quantidade de trabalho não sejam oficiais, a empresa gera um ambiente onde aqueles que não contribuem são hostilizados pelos próprios colegas, pois há disparidade de trabalho (Cote; Harris, 2020).

Edholm et al. (2017) apresentam uma análise de inúmeros *postmortens* e entrevistas com ex-desenvolvedores da indústria de jogos digitais, com objetivo de criar um perfil da prática do *crunch*, a fim de descobrir seus efeitos na indústria. Os relatos de trabalho excessivo são divididos em quatro categorias: o *crunch contínuo* que se estende pela maior parte da duração de um projeto; o *crunch final*

representando um período intenso de trabalho excessivo realizado ao final de um projeto para entregá-lo em um prazo; os *mini-crunches* que são pequenos períodos de *crunch* que acontecem em etapas diferentes do projeto; e o *crunch ilusório*, quando o estúdio não acha que está praticando *crunch* promovendo noites em claro no escritório e levando todo esse ambiente como um dinâmica de trabalho em equipe. Os quatro tipos acabam por ter efeitos diferentes no produto final e na equipe, mas uma característica que todos têm em comum é o aumento do nível de estresse de todos os envolvidos diretamente no projeto.

Lin et al. (2020) demonstram as consequências do trabalho excessivo e as doenças e óbitos relacionados a ele em Taiwan de 2015 a 2018. É descrito como o aumento nas leis trabalhistas do país em 2004 ajudou na identificação de casos de trabalho excessivo pelo estado, e como os casos identificados são classificados. É possível perceber que empresas que já possuem processos trabalhistas por trabalho excessivo são mais preocupadas em seguir as normas impostas ao combate dessas práticas, e que a conscientização sobre as doenças e síndromes geradas pelo trabalho em excesso é uma das melhores formas de não apenas evitar que aconteça, mas também de tornar mais fácil a identificação pelas próprias vítimas.

Borg et al. (2020) realizaram uma pesquisa com os participantes da Global Game Jam, um *hackaton* de desenvolvimento de jogos que dura 48 horas. Nesta pesquisa foram levantados os aspectos mais importantes sobre a produção de jogos digitais segundo os participantes, além de quais seus métodos favoritos para a realização dos estágios iniciais de um projeto. Com essas respostas foi possível perceber uma grande quantidade de pessoas que dizem que o trabalho por inúmeras horas é efetivo nesses estágios iniciais, mas que isto não pode ser mantido por um longo período de tempo sem consequências àqueles envolvidos no desenvolvimento. Nesses resultados foi possível traçar os perfis das pessoas da indústria de games e identificar suas opiniões sobre o combate a diferentes práticas abusivas da indústria, incluindo o *crunch*. Além disso, verificou-se que a comunicação e o melhor planejamento são as melhores formas de combater tais práticas.

Pode-se observar que em todos os trabalhos relacionados nesta seção, casos de *crunch* são analisados de forma a entender suas principais causas e consequências. Com isso, o presente trabalho usa como base essas análises para propor mecanismos por meio dos quais o *crunch* possa ser evitado e combatido.

CAPÍTULO 3 - CONSTRUÇÃO DO SOFTWARE - GAMEJECT

3.1. Conceito do Software

O software de apoio criado neste trabalho, o GameJect (nome vem da junção das palavras *game* e *project* que significam jogo e projeto, respectivamente) tem como objetivo permitir o acompanhamento de um projeto de desenvolvimento de jogos digitais para minimizar a ocorrência do *crunch*. Todos os integrantes do projeto devem utilizar o software para registrar o andamento do projeto e as possíveis ocorrências de problemas durante a execução deste, de forma que a comunicação entre as partes seja mais clara.

Os gerentes do projeto podem, além das funcionalidades que um usuário comum tem, visualizar o andamento de todas as equipes do projeto, visualizar todas as ocorrências relatadas pelos usuários e também adicionar tarefas e alterar as prioridades destas. Gráficos e tabelas mostrando o andamento do projeto e o progresso de cada equipe estão disponíveis também ao supervisor. Uma representação visual das ocorrências relatadas e em qual etapa do projeto elas se encontram em maior quantidade pode ser gerada com base nos dados inseridos pelos usuários. Com isso, ao final do projeto, tais dados poderão ser utilizados como documentação e usados como base para que seja possível ter uma visão melhor sobre o projeto, seu andamento e seus integrantes.

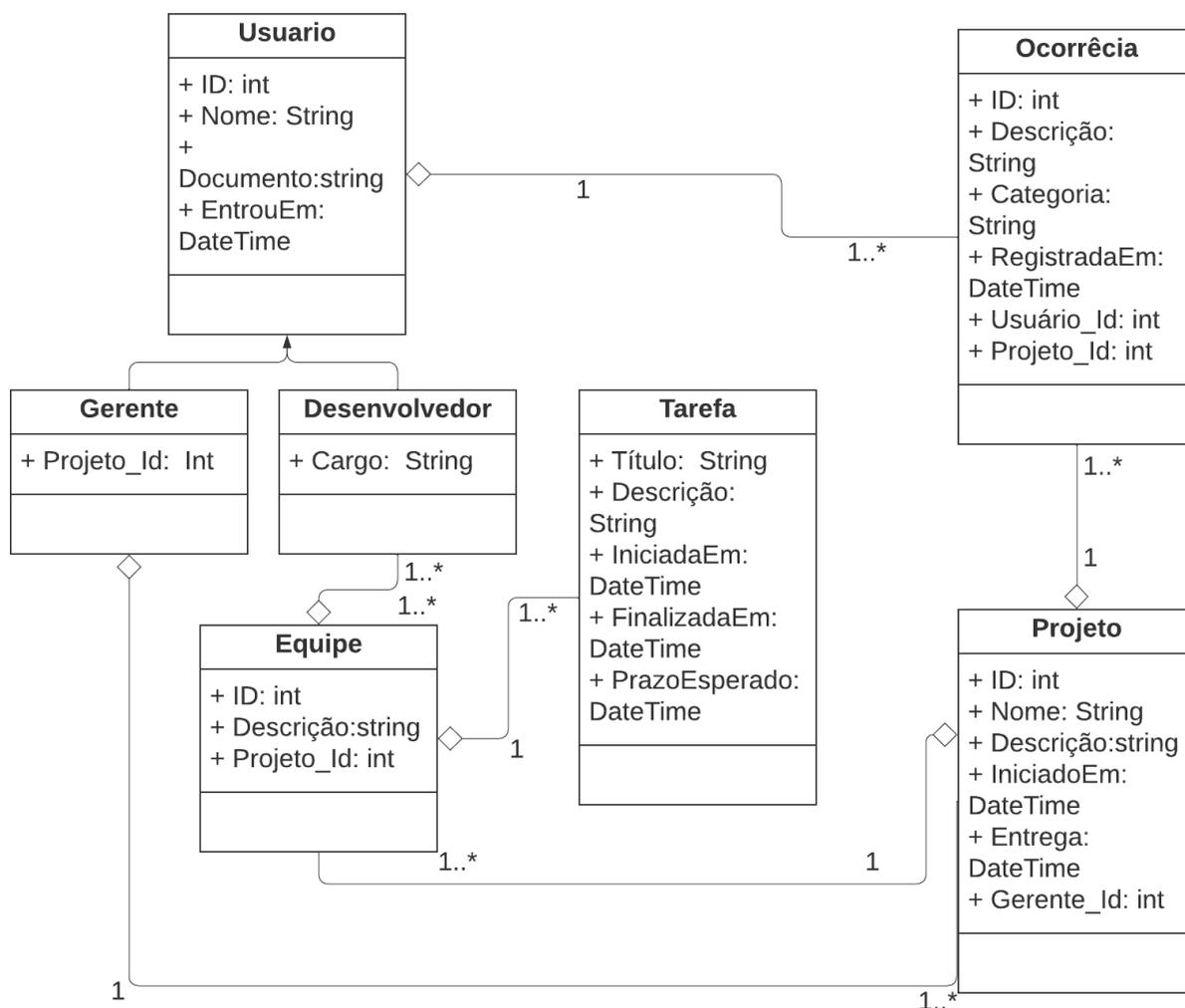
3.2. Modelo de Dados

Na Figura 1 tem-se o modelo de dados utilizado para nortear o desenvolvimento do software.

A partir da Figura 1 é possível observar uma estrutura com usuários que pertencem a equipes, as quais compõem um projeto. Um usuário pode ser um desenvolvedor ou um supervisor.

No caso de um desenvolvedor ele pertence a uma equipe, a qual pertence a um projeto, logo um projeto será composto por uma ou mais equipes que possuem inúmeros desenvolvedores. Estes podem a qualquer momento criar ocorrências, que estão relacionadas tanto à quem a criou quanto ao projeto ao qual este usuário está vinculado.

Figura 1 – Modelo de Dados.



Fonte: Elaborado pelo autor.

No caso dos gerentes, estes são relacionados diretamente a um projeto. Com isso, o gerente terá acesso aos dados de todas as equipes deste projeto e, por consequência, a todos os desenvolvedores alocados a este projeto, suas informações e as ocorrências criadas por estes. As tarefas podem ser criadas e finalizadas apenas pelos gerentes, mas podem ser visualizadas por todos os desenvolvedores membros da equipe a qual aquela tarefa foi atribuída.

3.3. Tecnologias Utilizadas

O software foi desenvolvido para a plataforma *Web*, usando a biblioteca do JavaScript, React.js - uma tecnologia que permite uma construção modularizada de ambientes *web*.

O *backend* é uma API Rest criada no *framework* ASP .NET Core da linguagem C# (Microsoft), uma linguagem que suporta um grande fluxo de dados. Com isso, o *Entity Framework* foi também utilizado com o modelo *CodeFirst*, para construção do código, com todas as classes, propriedades e métodos seguindo o paradigma da orientação a objetos.

O banco de dados e gerenciador de banco de dados utilizados são o SQL Server também da Microsoft, escolhido por sua facilidade de comunicação com a API do *backend*.

3.4. Funcionalidade do Software

Primeiramente, na tela de login mostrada na Figura 2, o usuário insere suas informações, e caso estejam corretas, ele terá acesso às funcionalidades do software de acordo com o perfil do usuário: Desenvolvedor ou Gerente. Nas subseções a seguir, funcionalidades são apresentadas levando em conta cada um desses perfis.

Figura 2 – Ilustração da Tela de Login.



A ilustração da tela de login apresenta um ícone no topo centralizado, composto por um documento com uma aba e um controle de videogame dentro dele, ambos em tons de roxo. Abaixo do ícone, há dois campos de entrada de texto empilhados verticalmente. O primeiro campo contém o texto 'insira seu usuário' e o segundo contém 'insira sua senha'. Ambos os campos possuem uma borda arredondada e um fundo branco. Na base da tela, há um botão retangular de cor roxa sólida com o texto 'ENTRAR' em letras maiúsculas brancas no centro.

Fonte: Elaborado pelo autor.

3.4.1. Uso do Software pelo Desenvolvedor

Após realizar o login, o desenvolvedor será apresentado à tela inicial da plataforma com uma mensagem de boas-vindas. O menu lateral é exibido no lado esquerdo da tela durante todo o uso da plataforma, como visto na Figura 3.

Figura 3 – Ilustração da Tela Inicial Desenvolvedor.



Fonte: Elaborado pelo autor.

Ao acessar a opção do menu Projetos, o desenvolvedor pode visualizar todos os projetos aos quais está vinculado, além de informações relacionadas, como visto na Figura 4.

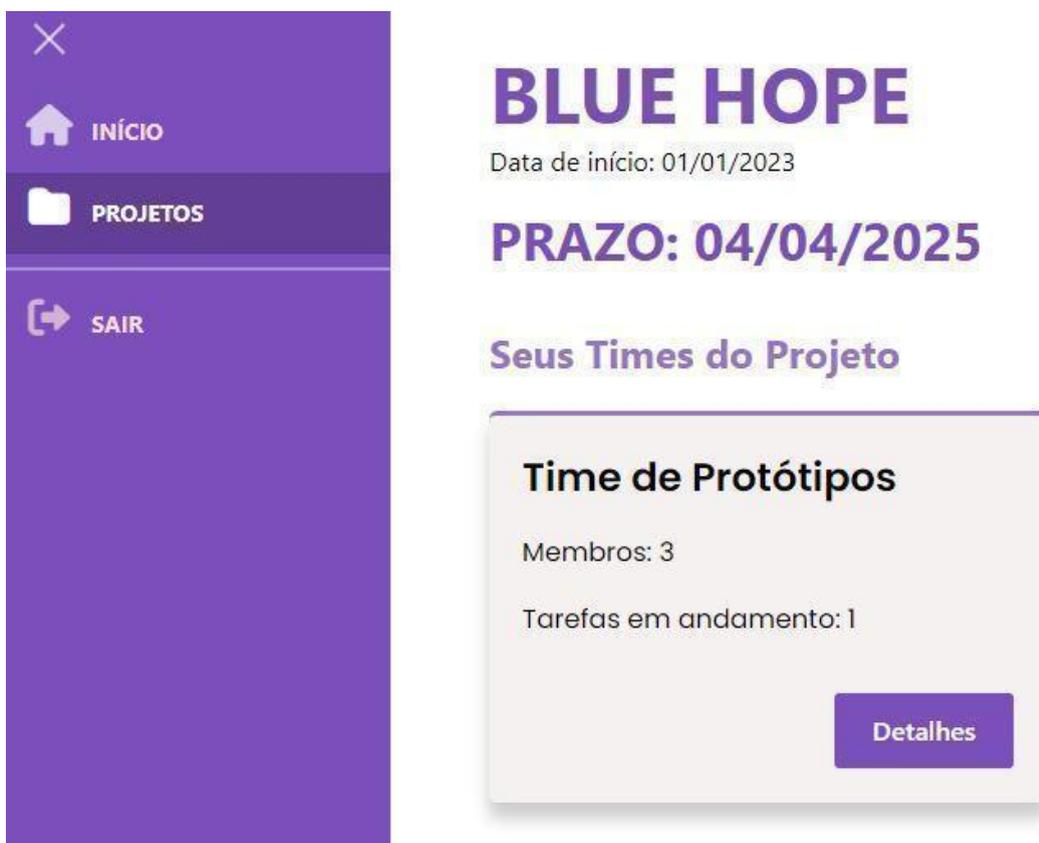
Figura 4 – Ilustração da Tela de Projetos do Desenvolvedor.



Fonte: Elaborado pelo autor.

Cada cartão de projeto dá acesso a uma tela, vista na Figura 5, a qual mostra detalhadamente as informações do projeto selecionado e os times do projeto aos quais o desenvolvedor faz parte.

Figura 5 – Ilustração da Tela de Detalhes do Projeto do Desenvolvedor.



Fonte: Elaborado pelo autor.

Na Figura 6, tem-se a tela de informações de um time que mostra ao desenvolvedor todas as tarefas atribuídas àquele time, tanto as concluídas, em andamento e pendentes.

Ainda na Figura 6, abaixo da seção de tarefas, o desenvolvedor pode ver uma tabela com todas as ocorrências que relatou dentro do time, podendo ver os detalhes de cada ocorrência, sua data de criação e categoria. Além disso, é possível ao desenvolvedor relatar uma ocorrência.

Ao clicar no botão Nova Ocorrência, uma tela de criação de ocorrência é mostrada, vide Figura 7, onde ele deve inserir as informações da ocorrência.

Figura 6 – Ilustração da Tela de Time do Desenvolvedor.

BLUE HOPE / TIME DE PROTÓTIPOS

Tarefas

Pendentes Em andamento Concluídas

12 - Conceitos dos Personagens

Criar os designs base e protótipos de modelos dos personagens

Prazo estimado: 02/02/2023

Suas ocorrências no Time

Data	Categoria	Detalhes
26/11/2022	Problema na Documentação	

Nova Ocorrência

Fonte: Elaborado pelo autor.

Figura 7 – Ilustração da Janela de Criação de Ocorrência.

×

Nova ocorrência

Descreva a ocorrência

Categoria

Bug ▾

Criar

Fonte: Elaborado pelo autor.

A qualquer momento o usuário pode selecionar a opção de sair da plataforma a partir do menu lateral, sendo apresentada uma janela de confirmação sobre se realmente deseja sair da plataforma, como visto na Figura 8. Caso selecione o botão

Não, a tela é apenas fechada e o usuário pode continuar sua navegação, mas caso selecione Sim, o usuário volta à tela de login.

Figura 8 – Ilustração da Janela de Confirmação de Logout.



Fonte: Elaborado pelo autor.

3.4.2. Uso do Software pelo Gerente

No caso do gerente, após o login, este é recebido também pela tela de boas-vindas, como é mostrado na Figura 9.

Figura 9 – Ilustração da Tela Inicial do Gerente.



Fonte: Elaborado pelo autor.

O gerente possui acesso a mais funcionalidades que o desenvolvedor. Um exemplo é a página onde são listados todos os desenvolvedores cadastrados no sistema, como visto na Figura 10, a qual pode ser acessada a partir do botão de Desenvolvedores no menu lateral.

Figura 10 – Ilustração da Tela de Desenvolvedores.



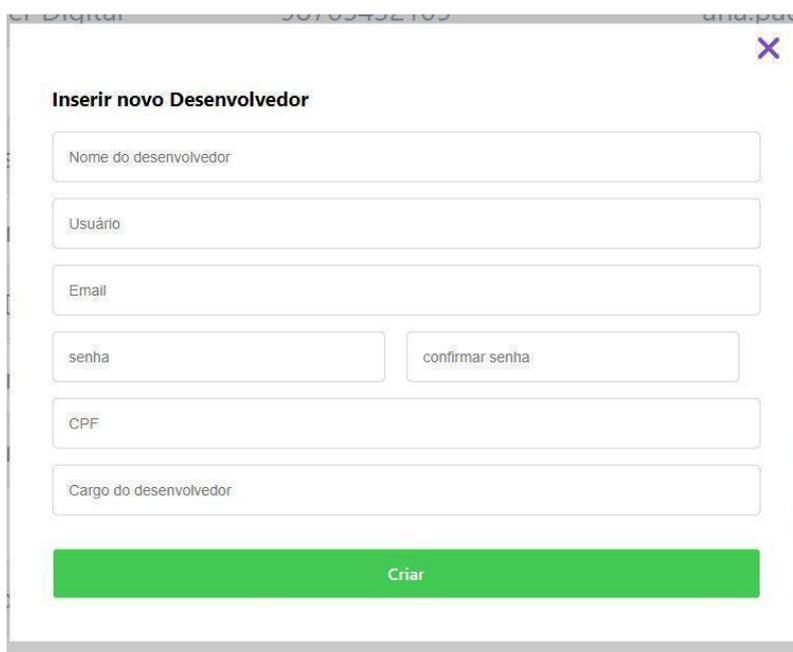
Nome	Cargo	Documento	Email	Ação
Ana Paula	Designer Digital	98765432109	ana.paula@mail.com	
Bruno Vieira	Artista 3D	12345678911	bruno@mail.com	
Janaina Souza	Programadora Unity	12345678911	janaina@mail.com	
Paulo Corrêa	Artista Conceitual	12780955864	paulo@mail.com	

+ Criar Novo

Fonte: Elaborado pelo autor.

Nessa mesma página o gerente tem a opção de criar um novo desenvolvedor no sistema ao clicar no botão de Criar Novo. Ao fazer isso, uma janela com campos de informações sobre o novo desenvolvedor a serem preenchidos é mostrada ao gerente de acordo com a Figura 11.

Figura 11 – Ilustração da Janela de Criação de Novo Desenvolvedor.



Inserir novo Desenvolvedor

Nome do desenvolvedor

Usuário

Email

senha

confirmar senha

CPF

Cargo do desenvolvedor

Criar

Fonte: Elaborado pelo autor.

Ao acessar a página de projetos, por meio do botão de Projetos do menu lateral (Figura 9), o gerente tem, ainda, as opções de deletar projetos e de criar novos projetos, conforme ilustrado na Figura 12.

Figura 12 – Ilustração da Tela de Projetos do Gerente.



Fonte: Elaborado pelo autor.

Ao selecionar o botão Criar Novo Projeto, a janela apresentada na Figura 13 é mostrada ao gerente, com informações sobre o novo projeto a serem preenchidas.

Figura 13 – Ilustração da Janela de Criação de Novo Projeto.

Fonte: Elaborado pelo autor.

Ao entrar na tela de detalhes de um projeto, o gerente pode visualizar todos os times daquele projeto, criar ou deletar times, como visto na Figura 14. Nessa mesma tela, há uma tabela com todas as ocorrências registradas no projeto.

Figura 14 – Ilustração da Tela de Detalhes do Projeto do Gerente.

BLUE HOPE
Data de início: 01/01/2023

PRAZO: 04/04/2025

Times do Projeto

Time Azul

Membros: 1

Tarefas em andamento: 1

[Detalhes](#) 🗑️

Time de Protótipos

Membros: 3

Tarefas em andamento: 1

[Detalhes](#) 🗑️

[Criar Time](#)

Ocorrências no Projeto
Ocorrências relacionadas ao longo do projeto: **21**

Data	Categoria	Time	Relator	Detalhes
26/11/2022	Problema na Documentação	Time de Protótipos	Ana Paula	☰

Fonte: Elaborado pelo autor.

Ao selecionar o botão Criar Time, a janela ilustrada na Figura 15 é mostrada ao desenvolvedor, onde ele deve inserir o nome do time a ser criado.

Figura 15 – Ilustração da Janela de Criação de Novo Time.

✕

Novo time do projeto Blue Hope

nome do time

Criar

Fonte: Elaborado pelo autor.

Na tela de um time, ilustrada na Figura 16, o gerente consegue visualizar em uma tabela os desenvolvedores que compõem esse time, além de poder adicionar outros desenvolvedores a ele.

Figura 16 – Ilustração da Tela de Time do Gerente (Membros do Time).

BLUE HOPE / TIME DE PROTÓTIPOS

Membros do time

Nome	Cargo	Documento	Ação
Ana Paula	Designer Digital	98765432109	
Bruno Vieira	Artista 3D	12345678911	
Paulo Corrêa	Artista Conceitual	12780955864	

[Adicionar ao time](#)

Fonte: Elaborado pelo autor.

Ao clicar no botão Adicionar ao Time logo abaixo da tabela, a janela ilustrada na Figura 17 é mostrada ao gerente, onde ele pode escolher desenvolvedores e adicioná-los ao time com a ação de adicionar.

Figura 17 – Ilustração da Janela para Adicionar Desenvolvedor ao Time.

Adicionar desenvolvedor ao time

Nome	Cargo	Ação
Mario Souza	Programador Lua	
Ana Júlia	Engenheira de Software	
Pedro Paulo	Compositor	
David Costa	Roteirista	
Ronaldo da Silva Paz	Engenheiro de Redes	
Roberto de Souza	Designer de Som	
Janaina Souza	Programadora Unity	
João Bosco	Game Designer	
Marina Silva	Programadora Unity	

Fonte: Elaborado pelo autor.

O gerente pode, além de visualizar as tarefas atribuídas ao time, como visto na Figura 18, criar novas tarefas, clicando no botão de Nova Tarefas para que a janela ilustrada na Figura 19 seja aberta.

Figura 18 – Ilustração da Tela de Tarefas atribuídas ao Time.

Tarefas

Pendentes Em andamento Concluídas

12 - Conceitos dos Personagens

Criar os designs base e protótipos de modelos dos personagens

Prazo estimado: 02/02/2023

Finalizar ✓

Criar Tarefa

Fonte: Elaborado pelo autor.

Figura 19 – Ilustração da Janela de Criação de Nova Tarefa.

Criar tarefa no time

Título da Tarefa

Código da Tarefa

Data de conclusão esperada

Descrição

Criar

Fonte: Elaborado pelo autor.

O gerente também possui acesso ao menu de Ocorrências e Dados (Figura 9), onde pode visualizar algumas estatísticas sobre seus projetos. Na tela há uma aba para cada um dos projetos cadastrados, dentro dessas abas há um gráfico mostrando a quantidade de ocorrências em cada mês do projeto e outro que mostra as ocorrências registradas com base em seus tipos definidos, como visto na Figura 20.

Figura 20 – Ilustração da Tela de Estatísticas (Gráficos).

DADOS E ESTATÍSTICAS



Fonte: Elaborado pelo autor.

Abaixo dos gráficos, o gerente pode visualizar as ocorrências separadas por time, onde há uma aba para cada time do projeto selecionado e, dentro dessas abas, uma tabela com as ocorrências de cada time é apresentada, como mostrado na Figura 21.

Figura 21 – Ilustração da Tela de Estatísticas (Ocorrências por Time).

The screenshot displays the 'Ocorrências por time' interface. At the top, there is a tab for 'Time de Protótipos'. Below the tab, it indicates 'O time tem 21 ocorrências'. The main content area is a table with the following data:

Data	Categoria	Relator	Detalhes
26/11/2022	Problema na Documentação	Ana Paula	[Menu Icon]
10/11/2022	Problema com o Marketing	Ana Paula	[Menu Icon]
10/11/2022	Problema com o Marketing	Ana Paula	[Menu Icon]

Fonte: Elaborado pelo autor.

CAPÍTULO 4 - AVALIAÇÃO

4.1. Público-Alvo da Pesquisa

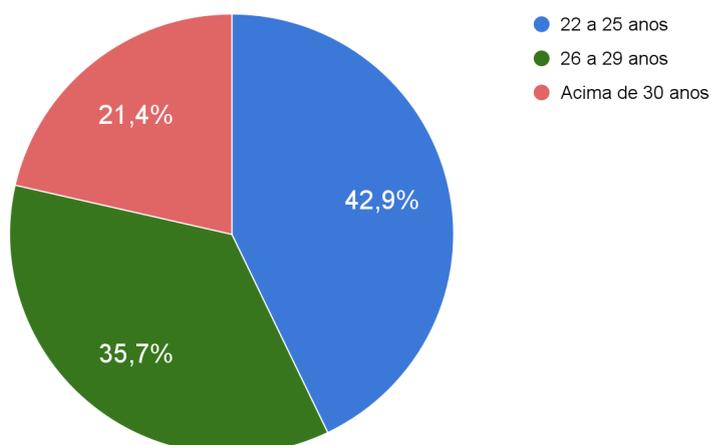
Após o final da implementação e hospedagem da plataforma na internet, foi realizada a sua divulgação juntamente com um formulário de avaliação (ver Apêndice A), criado na ferramenta *Google Forms*, para grupos de desenvolvedores de jogos digitais. O objetivo foi restringir a pesquisa avaliativa do GameJect às pessoas que têm ou já tiveram contato com projetos de desenvolvimento de jogos digitais e, portanto, melhor compreendem os conceitos envolvidos no presente trabalho. Apesar disso, não foi feita nenhuma confirmação sobre as identidades dos participantes e sobre a veracidade de suas informações relacionadas às suas participações na indústria, o que torna essa avaliação apenas uma pré-avaliação do software e deste trabalho.

Observa-se também que o formulário foi composto de questões voltadas para identificação do perfil dos respondentes e de suas experiências profissionais em projetos de jogos digitais para, finalmente, coletar a percepção deles sobre o GameJect e suas possíveis contribuições e limitações.

4.2. Resultado da Pesquisa

A pesquisa obteve um total de 14 respostas. Observou-se uma certa predominância de participantes do gênero masculino na pesquisa, com 11 participantes (78,6%) se declarando como sendo do gênero masculino, enquanto apenas dois (14,3%) do gênero feminino e um participante (7,1%) declarando seu gênero como outro. É possível observar no gráfico da Figura 22 as faixas etárias dos participantes, onde evidencia-se um público bastante jovem.

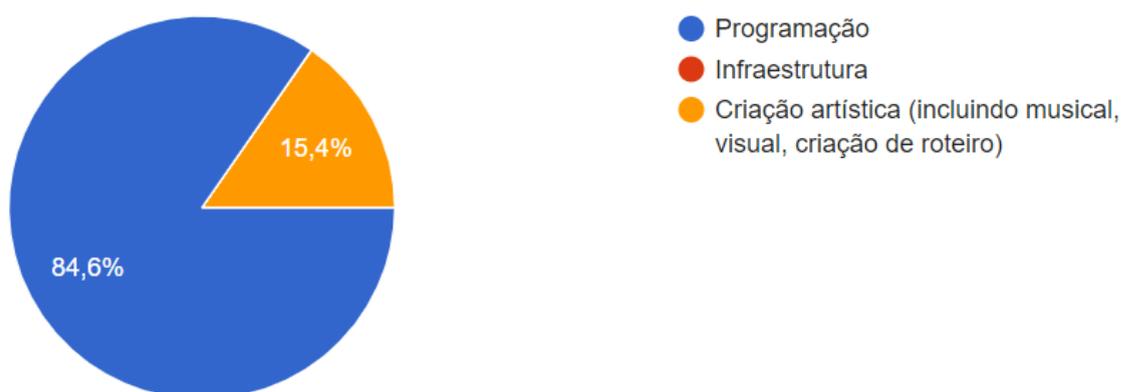
Figura 22 – Idades dos Participantes da Pesquisa.



Fonte: Elaborado pelo autor.

Destaca-se que, dos 14 respondentes, 11 declararam que atuam na indústria de jogos digitais atualmente, e dois declararam que já trabalharam nela no passado. Dentre estes 13 respondentes com experiência na indústria de jogos, a maioria relatou que trabalha ou trabalhava na área de programação dos projetos, como visto no gráfico da Figura 23.

Figura 23 – Área de Atuação dos Participantes no Desenvolvimento de Jogos.



Fonte: Elaborado pelo autor.

De todos esses participantes, 12 deles, correspondente a 92,3%, declararam que já estiveram em projetos que passaram por atrasos no cronograma, e nove (81,8%) também mencionaram que já passaram por *crunch*, ou seja, trabalharam

além de suas horas após problemas no projeto. De maneira complementar, dentre os que já passaram por situações de *crunch*, um total de 8 participantes (88,9%) mencionaram que o *crunch* não foi a solução para os problemas do projeto.

Além disso, entre todos os respondentes da pesquisa com experiência na indústria de jogos, 12 participantes (92,3%) concordam que o *crunch* não é a melhor solução para possíveis problemas em um projeto, pois não é suficiente para resolver os problemas complexos que exigem planejamento e eficácia para serem superados.

A partir desses resultados é possível observar a ocorrência frequente de *crunch* na indústria de jogos digitais, embora os profissionais não concordem que esta seja a melhor solução para os problemas de um projeto. Porém, devido a carências na gestão do projeto é comum que os estúdios promovam o *crunch* para conseguir finalizá-los.

Após os participantes utilizarem o GameJect, 92,9% declararam a plataforma como intuitiva e simples de se utilizar. Além disso, 100% dos entrevistados declararam que o uso da plataforma ao longo de um projeto de desenvolvimento de jogo digital, tanto pelo gerente de projeto quanto pelos desenvolvedores, é capaz de auxiliar na identificação de problemas no projeto e de ajudar a equipe a ter uma melhor comunicação, contribuindo para evitar que problemas levem ao *crunch*. Alguns participantes chegaram a declarar em suas respostas que a comunicação que a plataforma oferece, principalmente sobre as ocorrências, é uma chave importante de combate a possíveis problemas em projetos, pois possibilita que problemas sejam identificados rapidamente e sejam imediatamente combatidos, de forma a evitar que estes escalem para situações mais críticas.

Ao final da pesquisa, também foi perguntado aos participantes o que gostariam de adicionar na plataforma para torná-la mais eficiente. Com isso, seis participantes mencionaram que a funcionalidade de Tarefas poderia ser melhor trabalhada, de forma que os próprios desenvolvedores pudessem interagir com elas, além da possibilidade de atrelar ocorrências a tarefas para deixar mais claro para a gerência do projeto o contexto no qual a ocorrência foi relatada.

CAPÍTULO 5 - CONCLUSÃO

5.1. Considerações Finais

A pesquisa e análise da literatura apresentadas neste trabalho, mostram como o *crunch* é um problema que aflige a indústria de jogos digitais, e como este é muitas vezes a consequência de uma sequência de problemas causados por más decisões de gerência. Foram também mostrados os efeitos negativos do *crunch* tanto para os profissionais envolvidos quanto para o projeto: os desenvolvedores acabavam por sofrer consequências de saúde pelo *burnout* após longos períodos de trabalho excessivo e isso acabava por refletir negativamente na qualidade dos projetos em que trabalhavam.

Com esses fatos em mente, fica clara a importância deste trabalho em não apenas realizar uma análise sobre a ocorrência atual do *crunch* na indústria de jogos e os sinais e problemas iniciais de um projeto que podem levá-lo ao *crunch*, mas também em apresentar possíveis ações que podem contribuir para evitá-lo, assim como uma plataforma de apoio que auxilie na execução destas ações, denominada GameJect.

5.2. Contribuições

Os resultados obtidos por meio da pesquisa avaliativa realizada mostraram que a plataforma desenvolvida é uma ferramenta viável para auxiliar a gerência de projetos de desenvolvimento de jogos digitais.

Ao auxiliar na comunicação entre desenvolvedores e gerentes de projeto em relação a problemas que estejam afligindo o andamento deste, a plataforma torna possível a identificação e combate a esses problemas de forma rápida, de maneira com que esses não se tornem críticos e acabem por causar outros problemas no projeto, incluindo o *crunch*.

A funcionalidade que torna possível a denúncia de problemas por desenvolvedores por meio do registro de ocorrências durante a execução de projetos, sem dúvida, representou uma importante contribuição ao manter a gerência do projeto bem informada e, portanto, apta a tomar decisões que possam evitar o *crunch*.

5.3. Trabalhos Futuros

Como apontado pelos participantes da pesquisa, a funcionalidade de Tarefas do GameJect pode ser melhorada. Para isso, sugere-se o uso de um método específico para o controle das tarefas de um projeto, como o *kanban* (AHMAD et al., 2018), com o intuito de dar mais apoio aos desenvolvedores sobre suas tarefas. Logo, sua incorporação ao GameJect é algo que provavelmente agregaria mais eficiência nos resultados.

Outra melhoria possível seria aumentar o escopo do projeto além da indústria de jogos digitais, de forma com que abrangesse todos os tipos de desenvolvimento de software, afinal, o trabalho excessivo ocorre não apenas no desenvolvimento de jogos digitais, mas na maioria dos projetos de software.

REFERÊNCIAS

- A. FINLY, Postmortem: 2K Boston/2K Australia's BioShock, 2008. Disponível em: <http://www.gamasutra.com/view/feature/3774/postmortem_2k_boston2k_.php>. Acesso em 12 de junho de 2022.
- A. KHAN, "Project Scope Management," Case Stud. Proj. Program, Organ. Proj. Manag., vol. 48, no. 6, pp. 83–84, 2010.
- ADOLFSEN, Jacob Forsberg. The impact of poor stakeholder management practices A case study of Facebook. 2020. 73 f. Tese (Doutorado) - Curso de Economy, Copenhagen Business School, Copenhagen, 2020.
- AHMAD, M. O. et al. Kanban in software engineering: A systematic mapping study. Journal of Systems and Software, v. 137, p. 96–113, mar. 2018.
- ALEEM, Saiqa; CAPRETZ, Luiz Fernando; AHMED, Faheem. Critical Success Factors to Improve the Game Development Process from a Developer's Perspective. Journal Of Computer Science And Technology, [S.L.], v. 31, n. 5, p. 925-950, set. 2016. Springer Science and Business Media LLC.
- BLOOMBEG, Amazon Can Make Just About Anything—Except a Good Video Game, 2021. Disponível em: <<https://www.bloomberg.com/news/features/2021-01-29/amazon-game-studios-struggles-to-find-a-hit>> Acesso em 21 de junho de 2022.
- BORG, Markus; GAROUSI, Vahid; MAHMOUD, Anas; OLSSON, Thomas; STALBERG, Oskar. Video Game Development in a Rush: a survey of the global game jam participants. Ieee Transactions On Games, [S.L.], v. 12, n. 3, p. 246-259, set. 2020. Institute of Electrical and Electronics Engineers (IEEE).
- COTE, Amanda C; HARRIS, Brandon C. 'Weekends became something other people did': understanding and intervening in the habitus of video game crunch. Convergence: The International Journal of Research into New Media Technologies, [S.L.], v. 27, n. 1, p. 161-176, 26 mar. 2021. SAGE Publications.
- EDHOLM, Henrik; LIDSTROM, Mikaela; STEGHOFER, Jan-Philipp; BURDEN, Hakan. Crunch Time: the reasons and effects of unpaid overtime in the games industry. 2017 Ieee/Acm 39Th International Conference On Software Engineering: Software Engineering in Practice Track (ICSE-SEIP), [S.L.], p. 43-52, maio 2017. IEEE.
- KOMAL, Bakhtawar; JANJUA, Uzair Iqbal; ANWAR, Fozia; MADNI, Tahir Mustafa; CHEEMA, Muhammad Faisal; MALIK, Muhammad Noman; SHAHID, Ahmad Raza. The Impact of Scope Creep on Project Success: an empirical investigation. Ieee Access, [S.L.], v. 8, p. 125755-125775, 2020. Institute of Electrical and Electronics Engineers (IEEE).

PMI. "A Guide to the Project Management Body of Knowledge (PMBOK Guide)". 6^o Edition. Pennsylvania: PMI. 2017.

LIN, Ro-Ting; CHENG, Yawen; JIANG, Yan-Cheng. Exploring Public Awareness of Overwork Prevention With Big Data From Google Trends: retrospective analysis. Journal Of Medical Internet Research, [S.L.], v. 22, n. 6, p. 1-10, 26 jun. 2020. JMIR Publications Inc.

POLITOWSKI, Cristiano; PETRILLO, Fabio; ULLMANN, Gabriel C.; GUÉHÉNEUC, Yann-Gaël. Game industry problems: an extensive analysis of the gray literature. Information And Software Technology, [S.L.], v. 134, p. 106538, jun. 2021. Elsevier BV.

SCHREIER, Jason. Blood, Sweat, and Pixels: The Triumphant, Turbulent Stories Behind How Video Games Are Made. New York: Harper Paperbacks, 2017.

STATISTA. COVID-19 impact on the gaming industry worldwide. Disponível em: <<https://www.statista.com/topics/8016/covid-19-impact-on-the-gaming-industry-worldwide/>>. Acesso em: 20 Jul. 2021.

WOLF, Mark J. P.. The Video Game Explosion: A History from PONG to PlayStation and Beyond. Westport, Connecticut: Greenwood Press, 2007.

AL-RUBAIEI, Qais Hashil Salim; NIFA, Faizatul Akmar Abdul; MUSA, Suria. Project scope management through multiple perspectives: A critical review of concepts. In: Penang, Malaysia: [s.n.], 2018, p. 020025.

APÊNDICE A – Questionário da Pesquisa

1) Seu nome completo

2) Sua idade

3) Seu gênero:

I – Masculino

II - Feminino III – Outro

4) Trabalha atualmente com desenvolvimento de jogos digitais?

I – Sim

II – Não

5) Se não trabalha atualmente na área, já trabalhou no passado com desenvolvimento de jogos?

I – Sim

II – Não

6) Em qual das áreas abaixo você se enquadra em questão do seu cargo no desenvolvimento de jogos digitais?

I - Programação

II – Infraestrutura

III – Outro

7) Nos projetos em que trabalha/trabalhou, já ocorreram atrasos no cronograma?

I – Sim

II – Não

8) Se sim, diria que esses atrasos poderiam ter sido evitados por uma melhor gestão do projeto? Comente sobre suas ideias para solucionar/evitar tais atrasos

9) Você já passou por crunch(trabalho excessivo além das horas de trabalho estabelecidas) em algum desses projetos?

I – Sim

II – Não

10) Se sim, você considera que isso foi causado por uma sequência de problemas no projeto que, para serem resolvidos, demandaram trabalho excessivo?

I – Sim

II – Não

11) Se sim, você considera que os problemas foram solucionados após o período de crunch?

I – Sim

II – Não

12) Você acha que o crunch é a melhor solução para problemas no projeto?

I – Sim

II – Não

13) Justifique sua resposta da pergunta anterior

14) Você teve alguma dificuldade para utilizar a plataforma?

I – Sim

II – Não

15) Se sim, qual?

16) Achou a interface da plataforma simples e intuitiva?

I – Sim

II – Não

17) Se não, diga o que poderia ser melhorado

18) Você considera que o uso da plataforma ao longo de um projeto de desenvolvimento de jogo digital, tanto pelo gerente de projeto quanto pelos desenvolvedores, é capaz de auxiliar na identificação de problemas no projeto e de ajudar a equipe a ter uma melhor comunicação?

I – Sim

II – Não

19) Justifique sua resposta da pergunta anterior

20) Comente sobre o que poderia ser incluído na plataforma para auxiliar na gestão de um projeto de desenvolvimento de jogo digital, visando evitar o crunch