



Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Saúde Mental de *Gamers* e *Streamers*: Uma Aplicação Gamificada

Autor: Eileen Lima Ribeiro, Guilherme Mendes Pereira e
Samuel Barros Borges.

Orientador: Prof. Dr. Wander Cleber M. Pereira da Silva

Brasília, DF

2022



Eileen Lima Ribeiro, Guilherme Mendes Pereira e Samuel Barros Borges.

Saúde Mental de *Gamers* e *Streamers*: Uma Aplicação Gamificada

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Wander Cleber M. Pereira da Silva

Brasília, DF

2022

Eileen Lima Ribeiro, Guilherme Mendes Pereira e Samuel Barros Borges.
Saúde Mental de *Gamers* e *Streamers*: Uma Aplicação Gamificada/ Eileen
Lima Ribeiro, Guilherme Mendes Pereira e Samuel Barros Borges.. – Brasília, DF,
2022-

48 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Wander Cleber M. Pereira da Silva

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2022.

1. saúde mental. 2. gamers e streamers. I. Prof. Dr. Wander Cleber M. Pereira
da Silva. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Saúde
Mental de *Gamers* e *Streamers*: Uma Aplicação Gamificada

CDU 02:141:005.6

Eileen Lima Ribeiro, Guilherme Mendes Pereira e Samuel Barros Borges.

Saúde Mental de *Gamers* e *Streamers*: Uma Aplicação Gamificada

Monografia submetida ao curso de graduação em (Engenharia de Software) da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em (Engenharia de Software).

Trabalho aprovado. Brasília, DF, 07 de Outubro de 2022:

**Prof. Dr. Wander Cleber M. Pereira
da Silva**
Orientador

**Prof. M.Sc. Ricardo Ajax Dias
Kosloski**
Convidado 1

**Prof. M.Sc. Bernardo Tadeu Machado
Verano**
Convidado 2

Brasília, DF
2022

*Este trabalho é dedicado à todas as pessoas que
um dia sonharam em trabalhar com Jogos Digitais.*

Agradecimentos

Gostaríamos de agradecer às nossas famílias por nos proporcionar um ambiente saudável para o nosso desenvolvimento pessoal e educacional, além de todo apoio no decorrer do curso; a todos os Nossos colegas e amigos da Universidade de Brasília, por nos ajudarem a trabalhar na plataforma e proporcionarem um local de estudo saudável e produtivo; e a Deus por nos dar forças para perseverar nesta jornada até o fim.

*“Videogames são feitos para ser uma coisa
apenas: Diversão! Diversão para todos!”*

Satoru Iwata

Resumo

O hábito de consumo de jogos digitais é fenômeno recente, tendo seu início em fins do século passado; com a popularização da internet e do acesso aos jogos digitais, o mercado de consumo de conteúdo produzido em tempo real (*streaming*) de pessoas jogando é uma marca de nosso século. É comum que o jogador entretenha-se por várias horas quando propõe-se a jogar - esse comportamento agrava-se quando focamos nos jogadores *streamers*, pois combina-se a esse hábito o medo de perder audiência. Assim, é de suma importância que esses indivíduos consigam desenvolver o controle do hábito das atividades que exercem e utilizem ferramentas apropriadas, para que o exercício de sua atividade de jogador *streaming* não se torne prejudicial a sua saúde física e mental após longas jornadas. A aplicação *gamificada* voltada para a Saúde Mental de *Gamers* e *Streamers* visa a conscientização sobre a importância de se manter o bem-estar e os bons hábitos diários durante as práticas, muitas vezes excessivas, de jogos eletrônicos e criações de conteúdo. A fim de fomentar e contribuir para uma melhor qualidade de vida desse público-alvo, o software desenvolvido neste trabalho de conclusão de curso utilizou-se de técnicas de *gamificação* criadas por Yu-kai Chou (*Octalysis*), determinando um processo que aplica elementos de jogos em ambientes em que se buscam um maior engajamento e motivação no uso da aplicação.

A aplicação prevista neste trabalho foi desenvolvida em um período de quatro *sprints* e foi entregue como um MVP (Minimal Value Product) devido ao tempo curto de desenvolvimento, que causou a necessidade de uma redução no escopo da aplicação. Apesar disso, obteve-se êxito na execução do processo planejado, abrindo portas para trabalhos futuros que visem à expansão de suas funcionalidades.

Palavras-chaves: gamificação. *gamers* e *streamers*. saúde mental. jogos digitais. desenvolvimento de software.

Abstract

The habit of playing digital games is a recent one, its inception being traced back to the end of the last century. With the advent of the internet and access to digital gaming, the real-time content producing market of people playing video games (*streaming*) is a hallmark of the our century. It's become common for players to play for hours on end - this behaviour is aggravated when it's done by a *streamer*, as not playing more is closely related to the fear of losing viewers. As such, it is of utmost importance that these individuals take control of their habit and use appropriate tools for doing it, so that their job of playing games (*streaming*) is not a burden to their mental and physical health after prolonged hours. The use of a *gamified* approach for Mental Health of *Gamers* and *Streamers* strives to bring awareness to the importance of keeping their well-being and good daily habits during those practises, at times excessively, of playing video games and content creation. To better foment and contribute to the well-being of this target audience, this software developed for our thesis was developed using the **gamification** techniques of author Yu-kai Chou (*Octalysis*), providing a process which applies game elements to environments in which we need great commitment and motivation in its use.

The application made for this final paper was developed in the period of four *sprints* and was delivered as an MVP (Minimal Value Product) as time for development was short, which caused the necessity of reducing the scope of the software. Irregardless, it was an overall success for our proposed plan, opening door for new works that may expand upon these functionalities.

Key-words: welfare. gamification. *gamers*. *streamers*. mental health.

Lista de ilustrações

Figura 1 – Framework Octalysis Proposto por Yu Kai-Chou.	23
Figura 2 – Primeira Pergunta do Questionário.	26
Figura 3 – Segunda Pergunta do Questionário.	27
Figura 4 – Terceira Pergunta do Questionário.	27
Figura 5 – Quarta Pergunta do Questionário.	27
Figura 6 – Quinta Pergunta do Questionário.	28
Figura 7 – Quinta Pergunta do Questionário.	28
Figura 8 – Sexta Pergunta do Questionário.	28
Figura 9 – Sétima Pergunta do Questionário.	29
Figura 10 – Oitava Pergunta do Questionário.	29
Figura 11 – Nona Pergunta do Questionário.	29
Figura 12 – Décima Pergunta do Questionário.	30
Figura 13 – Comunicação entre processos entre cliente e servidor.	32
Figura 14 – Diagrama de Arquitetura.	32
Figura 15 – Modelo de Banco de Dados	34
Figura 16 – Diagrama de Pacotes - Flask.	35
Figura 17 – Diagrama de Pacotes - Flutter.	35
Figura 18 – Diagrama de Construção e Entrega.	37
Figura 19 – Telas de Início e Login.	41
Figura 20 – Telas Principal e Timer.	42
Figura 21 – Tela Fim da Stream e Perfil.	42
Figura 22 – Code Climate do Back-end.	43

Lista de tabelas

Tabela 1 – Requisitos Funcionais	36
Tabela 2 – Cronograma	39

Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
CD	<i>Continuous Delivery</i>
CI	<i>Continuous Integration</i>
FTP	<i>File Transfer Protocol</i>
IPTV	<i>Internet Protocol Television</i>
JSON	<i>JavaScript Object Notation</i>
JWT	<i>JSON Web Token</i>
HTTP	<i>Hypertext Transfer Protocol</i>
MVP	<i>Minimal Value Product</i>
OAuth	<i>Open Authorization</i>
OMS	Organização Mundial da Saúde
pip	<i>Package Installer for Python</i>
PMI	<i>Project Management Institute</i>
PMBOK	<i>Project Management Body of Knowledge</i>
REST	<i>Representational State Transfer</i>
SDM	<i>System Development Methodology</i>
SGBD	Sistema Gerenciador de Banco de Dados
SMTP	<i>Simple Mail Transfer Protocol</i>
SQL	<i>Structured Query Language</i>
WSGI	<i>Web Server Gateway Interface</i>

Sumário

1	INTRODUÇÃO	14
1.1	Contexto	14
1.2	Justificativa	14
1.3	Problema de pesquisa	15
1.4	Objetivos	15
2	REFERENCIAL TEÓRICO	17
2.1	Revisão Bibliográfica	17
2.2	Saúde Mental	17
2.3	Jogos Digitais	18
2.4	Relação entre Jogos Digitais e Saúde Mental	19
2.5	O que é um <i>Streamer</i> ?	19
2.6	Desafios dessa nova profissão.	20
2.7	Gamificação	21
2.7.1	<i>Framework</i> Octalysis	22
3	METODOLOGIA	24
3.1	Metodologias de Desenvolvimento	24
3.1.1	Metodologia Ágil, Scrum e Kanban	24
3.2	Definição de Requisitos	24
3.3	Gerenciamento de Riscos	25
3.4	Coleta de Dados	25
4	PLANEJAMENTO DO SISTEMA	31
4.1	Arquitetura da Solução	31
4.1.1	Visão Geral	31
4.1.2	Modelo Cliente-Servidor	31
4.1.3	Tecnologias	32
4.1.3.1	Front-End	32
4.1.3.2	Back-End	33
4.1.3.3	Banco de Dados	33
4.1.3.4	Modelo de Dados	34
4.1.3.5	Diagrama de Pacotes	34
4.2	Requisitos	36
4.3	Configuração de ambiente	36
4.3.1	Poetry	37

4.3.2	Docker	37
4.4	Construção e Entrega	37
4.4.1	Controle de Versão	37
4.4.2	Integração Contínua	38
4.4.3	Análise Estática	38
4.4.4	Entrega	38
4.5	Escopo Planejado	38
4.6	Cronograma do Desenvolvimento	38
5	RESULTADOS DO DESENVOLVIMENTO	40
5.1	Visão Geral	40
5.1.1	MVP	40
5.1.2	A Escala de Bem-Estar	40
5.1.3	Back-end	41
5.1.4	Front-End	41
5.2	Testes	43
5.2.1	Cobertura de Testes e Análise Estática	43
5.2.2	Estratégia de testes	43
5.3	Privacidade e Segurança	44
6	CONCLUSÕES	45
6.1	Considerações finais	45
6.2	Trabalhos futuros	46
	REFERÊNCIAS	47

1 Introdução

A última década apresentou uma disseminada normalização de novos hábitos de consumo perante os jogos digitais: o que até então era considerada uma atividade direcionada a um nicho específico de pessoas, gradualmente foi sendo adotada por outras camadas e nichos. Essa maior disseminação no consumo de jogos digitais resultou, hoje, no surgimento de uma nova classe de profissionais, chamados *streamers*, que ganham a vida transmitindo para o resto do mundo, por meio da internet, suas experiências imersivas com jogos digitais ao mesmo tempo em que interagem ao vivo com suas comunidades de espectadores.

1.1 Contexto

Essa nova profissão apresenta diversas vantagens em relação a outras profissões de cunho mais “tradicional”, tais como a autonomia e flexibilidade da jornada de trabalho. Ao mesmo tempo, porém, a profissão de *streamer* apresenta desafios específicos – como por exemplo a dificuldade de separar a vida pessoal da profissional (GJESTANG, 2020), e a prevalência *burnouts* por causa de falta de balanceamento das horas *online* das horas *offline* (YU; ALVAREZ; CHEN, 2021) – causados pela sua forma de remuneração que, segundo Yu, Alvarez e Chen (2021), fazem os *streamers* pensarem que eles devem estar constantemente transmitindo conteúdo. Porém, por conta de sua natureza recente, ainda não há em gama suficiente registros, pesquisas e bibliografias que tratem dos impactos na saúde mental causados por eles.

1.2 Justificativa

Segundo Gjestang (2020), muitos *streamers* são pressionados a trabalhar horas muito longas para manter o engajamento dos fãs, com vários deles transmitindo de 40 a 50 horas por semana. Como expõe um dos *streamers* entrevistados pelo autor:

Eu trabalho 8 horas por dia, de segunda a sexta, 40 horas no total. Sem pausa para o almoço. Não posso fazer uma pausa de 30 minutos porque se eu saio pra comer a audiência terá ido embora. Minhas horas *online* na semana podem chegar a 40, enquanto as *offline* podem ser de 10 a 20. (GJESTANG, 2020)

Nesse contexto de insalubridade, é de se esperar que ocorram eventuais fatalidades neste meio, como a ocorrida no ano de 2019, quando a comunidade de *streaming* foi

surpreendida com a notícia do trágico suicídio do *YouTuber* e *streamer* Desmond Amofah, mais conhecido como seu nome de usuário “Etika”. Segundo a notícia escrita por [Peldon \(2022\)](#), Amofah era filho do político ganense Owuraku Amofah, e vivia no bairro *Brooklyn* em Nova Iorque. Os seus episódios de instabilidade psicológica são bem documentados:

Notavelmente, eles começaram em Outubro de 2018, quando Amofah demonstrou comportamento estranho e fez upload de conteúdo gráfico sexual no seu canal *EWNetwork* violando as diretrizes de comunidade do youtube, causando a suspensão do canal. ([PELDON, 2022](#)).

Em 2019, as coisas começaram a escalar, quando Etika começou a postar uma série de perturbadoras mensagens suicidas nas redes sociais. Um de seus *tweets* em janeiro revelou sua dificuldade em lidar com seus problemas psicológicos. ([PELDON, 2022](#)).

No dia 24 de Junho de 2019, depois de estar desaparecido por cinco dias, o corpo de Amofah foi encontrado no rio *East River*, e a causa da sua morte foi determinada como “Suicídio por afogamento” ([ORJOUX, 2019](#)). Apesar de Amofah já apresentar um histórico de problemas psicológicos anterior a se tornar *streamer*, fica o questionamento se a sua escolha de profissão influenciou no seu trágico destino.

1.3 Problema de pesquisa

Uma das discussões que desponta em relevância ao se discutir os rumos dessa profissão que lida tão intimamente com a tecnologia digital seria a de como tais profissionais poderiam cuidar melhor de sua saúde mental. Portanto, o problema alvo deste trabalho é decorrido da seguinte questão: Como desenvolver uma aplicação móvel gamificada que auxilie no acompanhamento da saúde mental de *gamers* e *streamers*? Com essa questão o trabalho pretende investigar quais as angústias mais comuns entre *streamers* e, com base nos dados levantados, projetar um processo gamificado que acompanhe a evolução da saúde mental dos seus usuários.

1.4 Objetivos

Dado esse breve contexto, o objetivo principal deste trabalho é lançar bases para a proposta de desenvolvimento de um aplicativo digital que auxilie na promoção e adoção de hábitos saudáveis – visando particular à saúde mental – desses profissionais que lidam intimamente com as tecnologias digitais, utilizando como ferramenta base para esse acompanhamento interativo a própria tecnologia digital familiar a esse nicho. Assim, como objetivos específicos podemos elencar:

-
- Identificar quais os desafios psicológicos específicos das pessoas que se identificam como streamers de games;
 - Sistematizar por meio da gamificação maneiras eficientes e humanas de lidar com tais desafios;
 - Desenvolver uma aplicação móvel que, por meio de processos gamificados, promova a adoção de hábitos saudáveis com relação à saúde mental e que, também, quando necessário, aconselhe a busca de ajuda profissional.
 - Possível integração desta aplicação com um bot na plataforma Discord, a qual é extensamente utilizada pelo público-alvo em questão.

2 Referencial Teórico

2.1 Revisão Bibliográfica

Para o levantamento das informações usadas no referencial teórico foi realizada uma revisão bibliográfica para conferir o estado da arte; ou seja, foram levantadas diversas publicações científicas feitas previamente sobre o tema proposto (saúde mental de *streamers*). Foram buscados artigos científicos que continham os termos chave “saúde mental”, “gamers” e “streamers”; foram levantados 15 artigos recentes sobre o assunto, dos quais, após uma leitura mais aprofundada, sete foram descartados e os oito restantes serviram como a base teórica do trabalho. As informações mais relevantes destas publicações foram, então, compiladas e citadas. Isso contribuiu para a definição das perguntas que foram feitas na pesquisa, o levantamento dos requisitos, e outras decisões importantes do projeto.

Os artigos foram aproveitados ou rejeitados de acordo com a relevância das informações apresentadas para a utilização neste trabalho. Os artigos rejeitados, apesar de terem sido obtidos com as palavras-chave mencionadas anteriormente, apenas tangenciavam o tema e de nada contribuiriam para o desenvolvimento do trabalho.

2.2 Saúde Mental

Segundo a [OMS \(2018\)](#), a saúde mental não é apenas a ausência de doenças mentais, mas sim que a saúde mental é uma parte intrínseca da saúde como um todo. Não havendo, assim, “saúde” sem “saúde mental”.

A saúde é um estado de completo bem-estar físico, mental e social, e não consiste apenas na ausência de doença ou de enfermidade. ([OMS, 2006](#)).

Sendo assim, a saúde mental pode ser definida como “um estado de bem-estar no qual um indivíduo percebe que com suas habilidades consegue: lidar com os estresses normais do cotidiano; trabalhar de forma produtiva; e fazer contribuições para a sua comunidade.” ([OMS, 2018](#)). Apesar disso, alguns autores discordam desta perspectiva de saúde mental, que segundo [Galderisi et al. \(2015\)](#), são influenciadas por tradições hedônicas e eudemônicas que advogam por emoções positivas e excelência funcional. Isso, segundo [Galderisi et al. \(2015\)](#), arrisca excluir as pessoas introvertidas, pessoas que lutam contra o que percebem como injustiça e outras minorias que experienciam rejeição e

discriminação. Tendo isso em mente, a definição de “saúde mental” que utilizamos para o projeto e acreditamos que mais se aproxima da realidade seria:

Um estado dinâmico de equilíbrio interno que permite que os indivíduos usem suas habilidades em harmonia com os valores universais da sociedade. Habilidades cognitivas e sociais básicas; a habilidade de reconhecer, expressar e modular suas próprias emoções além de ter empatia com outros indivíduos; flexibilidade e habilidade em lidar com eventos adversos da vida enquanto funciona em papéis sociais; e uma relação harmoniosa entre corpo e mente representam componentes importantes da saúde mental que contribuem, de maneiras variadas, para o estado de equilíbrio interno (GALDERISI et al., 2015).

2.3 Jogos Digitais

Segundo Tavinor (2008) “Video Game” é um conceito inerentemente difícil de se definir. Isso acontece porque apesar de existir diversos termos que buscam estreitar o seu significado, como “jogos de computador” ou “jogos eletrônicos”, estes termos nem sempre podem ser utilizados intercambiavelmente. Com “jogos de computador” normalmente sendo utilizado para se referir a jogos que são jogados em um computador pessoal e “Jogos eletrônicos” sendo utilizado também para descrever brinquedos físicos que se utilizam de dispositivos eletrônicos.

Existe também a problemática da classificação de “Video Games”; afinal falar que alguém está jogando um jogo é tão útil quanto falar que alguém está assistindo um filme, ainda mais no contexto de transmissões *on-line*, onde é de critério do usuário pesquisar jogos cujo gênero tem-se interesse de assistir. De acordo com Arsenault (2009), classificá-los pode se tornar impreciso ou até mesmo desafiador diante da vasta quantidade de gêneros e sub-gêneros que tem sobreposição um no outro, além da subjetividade do que é ou não um gênero em específico. Uma inovação recente que pode ser citada aqui é na plataforma de Jogos Digitais *Steam*, que em 2019 introduziu um sistema de *tags* definidas por usuários dinamicamente, deixando a sua classificação mais como um consenso público sobre cada jogo.

Tendo isso em mente, neste projeto utilizaremos o termo propositalmente abrangente “Jogos Digitais” para traduzir o termo inglês “Video Game” que é definido como: “Um jogo em que o jogador controla imagens que se movimentam em uma tela se utilizando de botões ou pequenas alavancas.” (CAMBRIDGE, 2022).

Para o contexto de *streaming*, existem duas categorias principais a serem exploradas: jogos *on-line*, muitas vezes ligados ao cenário de *eSports* e jogos *single-player*, mais

ligados a *Reviews*, *Let's Plays* e *Speedruns*.

2.4 Relação entre Jogos Digitais e Saúde Mental

Desde a introdução do hábito de consumo de jogos digitais na vida humana, o senso comum nos diz que há uma correlação direta entre o número bruto de horas jogadas por dia e a deterioração da saúde mental do jogador, porém pesquisas recentes contrariam essa visão. Segundo [Sauter, Braun e Mack \(2020\)](#) o número bruto de horas jogadas mesmo servindo como um preditor para a satisfação com a vida, sua relevância prática é insignificante. Por exemplo, estes autores perceberam que um aumento de 10 horas bruto apenas aumentava o escore de ansiedade generalizada em um terço de ponto em uma escala que vai de 0 a 21. O que realmente impacta na saúde mental dos *gamers* aparenta ser o motivo pelo qual eles jogam:

Comparado com o motivo mais prevalente entre *gamers* ('se divertir'), ter um impulso avassalador por conquistas aparenta ser prejudicial à satisfação com a vida e ansiedade. A relevância da motivação aparenta ser especialmente proeminente em pessoas que jogam para escapar das dificuldades da vida cotidiana ('distração') ([SAUTER; BRAUN; MACK, 2020](#)).

De acordo com [Cabeza-Ramírez, Fuentes-García e Muñoz-Fernandez \(2021\)](#), Jogos Digitais têm se tornado mais do que um simples hobby ligado a crianças e adolescentes; agora se tornando uma atividade de lazer associado também a adultos e uma forma de conexão social, transformando não apenas os seus jogadores mas também os jogos em si, que agora são criados com o aspecto de *streaming* em mente durante seu desenvolvimento.

2.5 O que é um *Streamer*?

Segundo [Gjestang \(2020\)](#), um *streamer* é um trabalhador independente que produz conteúdos em vídeo ao vivo em plataformas diversas na internet que qualquer um pode assistir. Ainda de acordo com o autor citado, é entendido que o conteúdo que eles produzem é feito por vontade própria, está intimamente ligado com seus valores e interesses, e em sua grande maioria consiste de atividades de lazer como jogos digitais e música.

De acordo com [Edge \(2013\)](#), o Protocolo de Televisão na Internet (IPTV), que possibilita a transmissão de transmissões vindas da internet para TV, foi um grande marco para o crescimento do ambiente de *streaming*, sendo o *Twitch.TV* a plataforma mais popular para a modalidade de jogos, com 34 milhões de espectadores por mês únicos em 2013. Nos últimos anos, outras plataformas vêm surgindo, sejam elas competidoras diretas com a *Twitch.TV*, como o *Facebook Gaming* e *Youtube Gaming* principalmente, ou alter-

nativas voltadas a um nicho mais específico, como o *Picarto.TV*, voltada a transmissões de desenho ou animação.

Vendo esse massivo potencial em popularidade, vários setores da indústria de jogos hoje em dia já levam em conta a publicidade que transmissões de seus jogos terão nos seus respectivos ciclos de vida de *Esports*, por exemplo, se tornaram bastante populares nestas plataformas devido a *streamers* jogadores profissionais destes jogos que cultivam da base de jogadores que têm interesse por jogos competitivos. Por outro lado, jogos *indie*, que não costumam ter muito orçamento para o *marketing*, podem se tornar populares de um dia para um outro dada a descoberta daquele jogo por um *streamer* popular. (JOHNSON; WOODCOCK, 2019)

2.6 Desafios dessa nova profissão.

O fato da profissão de *streamer* ser de um caráter extremamente atípico, faz com que ela possua tanto benefícios quanto desafios sem precedentes. É evidente que na profissão de *streamer* é difícil de se desconectar do trabalho, tanto porque o trabalho do *streamer* é muito próximo da sua personalidade e identidade, quanto porque o contexto da profissão fomenta uma necessidade de comunicação e interação digital constante com suas respectivas comunidades (GJESTANG, 2020).

Apesar de serem considerados *influencers*, os *streamers* enfrentam problemas específicos do nicho nos quais se encontram (JOHNSON; WOODCOCK, 2019). Os *streamers* têm dificuldade em vender seu próprio valor às empresas patrocinadoras, pois o nicho em que suas audiências costumam se incluir tende a receber com receio formatos óbvios de propaganda e marketing. Isso agrava o *stress* já sentido normalmente porque, como expõe Yu, Alvarez e Chen (2021), os *streamers* dependem de um audiência que influencia diretamente nos seus ganhos monetários, o que cria uma pressão externa que os obriga a transmitir conteúdo regularmente.

O costume de transmitir conteúdo por durações longas (acima de oito horas seguidas) tem levantado preocupações sobre como a privação de sono e exaustão podem afetar os *streamers* (YU; ALVAREZ; CHEN, 2021 apud BONNAR et al., 2019).

Além da grande quantidade de horas que se espera que sejam transmitidas frequentemente, outra desvantagem de depender de uma audiência inconstante é, segundo os *streamers* entrevistados por Gjestang (2020), A falta de liberdade de ser autêntico. Segundo estes mesmos *streamers* uma das coisas mais desgastantes na profissão é ter que ajustar seus posicionamentos de acordo com acontecimentos políticos cotidianos. O impacto disso pode ser bastante significativo na saúde psicológica dos *streamers*, Como

expôs Yu, Alvarez e Chen (2021 apud JOHNSON; WOODCOCK, 2019): “Estar online por longos períodos de tempo enquanto se interpreta um personagem pode causar impactos emocionais negativos”.

2.7 Gamificação

A gamificação é uma tática com grande potencial para influenciar o comportamento das pessoas, sendo uma técnica aplicável em diversos contextos e que também tem sido explorada na engenharia de software (DUBOIS; TAMBURRELLI, 2013). O Oxford English Dictionary define a gamificação da seguinte forma:

A ação ou processo de transformar algo em ou como um jogo; tendo a aplicação de elementos de jogos (como pontuação, competitividade, etc) a outras áreas de atividade, normalmente para incentivar o envolvimento com um produto ou serviço. (OXFORD, 2022)

Isso sugere que, ao gamificar um processo, o engajamento dos usuários de uma aplicação tendem a aumentar, motivando-os a atingir determinados objetivos de uma maneira mais prazerosa (CHOU, 2016), incentivando e popularizando atividades específicas.

O termo gamificação foi dito pela primeira vez em 2002 por Nick Pelling, um desenvolvedor de jogos (MARCZEWSKI, 2013). E os estudos de gamificação na literatura de engenharia de software tendem a se concentrar em técnicas baseadas em produtos, processos ou no ensino de engenharia de software. Em todas as três vertentes de pesquisa, o foco pende a ser o aumento do engajamento. Para softwares gamificados, isso significa maior envolvimento dos usuários finais e, como sugerido acima, para a gamificação de processos de desenvolvimento, isso significa maior envolvimento dos participantes inseridos no contexto.

A gamificação está sendo cada vez mais reconhecida (ROUSSOU, 2004) como o processo/técnica de extração de elementos motivadores e envolventes encontrados em jogos e aplicá-los a atividades produtivas ou educacionais do mundo real. Chou (CHOU, 2016), reconhece este processo como “Design Centrado no Ser Humano” ou então “Design Centrado no Usuário (DCU)”, que valoriza os motivos, estados cognitivos e emocionais do usuário e, portanto, otimiza para seus sentimentos, motivações e engajamento, diferentemente do “Design Focado na Função”, que é projetado para pura eficiência. O termo gamificação foi adotado porque foi a indústria de jogos que foi a primeira a dominar o DCU.

2.7.1 Framework Octalysis

O *Framework Octalysis* é uma estrutura que organiza sistematicamente uma lista de elementos gamificados ou impulsos cognitivos que podem ser utilizados no “Design Centrado no Usuário” para tornar uma aplicação atraente e motivadora (CHOU, 2016).

A estrutura sugere que quase todos os jogos são “divertidos” pois apelam a alguns impulsos centrais do cérebro humano que os influenciam a praticarem certas atividades. O Octalysis organiza esses fatores motivacionais de acordo com a Fig. 1 nos seguintes 8 Core Drives principais (CHOU, 2016), que são baseados em uma forma octogonal, o que explica seu nome.

- *Significado épico e Chamado* é a necessidade de participar de algo maior apenas do que você mesmo;
- *Desenvolvimento e Realização* tem a ver com motivar as pessoas pois elas estão sentindo que estão melhorando, estão subindo de nível e alcançando a maestria;
- *O Empoderamento da criatividade e do Feedback* é o principal motor que motiva as pessoas a incorporar sua criatividade, experimentar diferentes combinações e estratégias, buscar uma resposta e se adequar;
- *Propriedade e Posse* é o principal impulso que motiva as pessoas a acumular posses, melhorá-las, protegê-las e obter mais;
- *Influência social e Relacionamento* refere-se às atividades motivadas pela influência de outras pessoas;
- *Escassez e Impaciência* é o que motiva as pessoas a quererem algo que não podem ter de forma imediata e fácil;
- *Imprevisibilidade e Curiosidade* é a vontade de descobrir o resultado desconhecido e envolver o acaso;
- *Perda e evitação* referem-se aos fatores motivadores que ajudam as pessoas a evitar situações que não querem que aconteçam.

Faz-se importante ressaltar que os Core Drives podem ser aplicados de maneira isolada, porém são capazes de conversar entre si, existindo técnicas de design de jogos que estimulam um ou mais core drives, não sendo estes elementos tratados de forma isolada.



Figura 1 – Framework Octalysis Proposto por Yu Kai-Chou.

3 Metodologia

3.1 Metodologias de Desenvolvimento

Segundo [Chan \(2008\)](#), uma Metodologia de Desenvolvimento de Sistemas (SDM) é definido como uma coleção documentada de políticas, processos e procedimentos comumente utilizado por equipes de desenvolvimento de software para melhorar para melhorar seus processos de desenvolvimento. Sendo assim, temos em nossa disposição diversas alternativas que podem otimizar o processo de desenvolvimento deste projeto.

3.1.1 Metodologia Ágil, Scrum e Kanban

Projetos de software podem se tornar difíceis quando se leva em conta o fato das tarefas nem sempre estarem claramente estabelecidas, ou até mesmo se elas serão viáveis para se desenvolver devido ao seu grau de dificuldade de implementação não ser fixo. Cada problema e sua respectiva solução precisa ser muito bem pensado dada a especificidade de seu contexto, o que pode tornar até mesmo o cronograma da tarefa por vezes imprevisível devido a lógicos e ocasionais *bugs*. Por isso, metodologias de projeto iterativas e princípios Ágeis, como o *Scrum*, *Extreme Programming*, *Kanban* entre outros servem como guia para diversos projetos hoje em dia. ([WEFLEN CAMERON A. MACKENZIE, 2022](#))

3.2 Definição de Requisitos

Na etapa de elicitação de requisitos, serão utilizadas as técnicas de questionário, *brainstorming* e prototipação. Num primeiro momento enviaremos um questionário para alguns *streamers* cujo contato a equipe de desenvolvimento deste projeto possui; após o período de aplicação do questionário, faremos um *brainstorming* entre a equipe de desenvolvedores visando a levantar o máximo de funcionalidades relevantes possíveis a partir das respostas obtidas. Com a elicitação dos requisitos identificados na etapa anterior, faremos uma rápida prototipação, a fim de validar os requisitos levantados; uma vez validados, serão incorporados em um *Product Backlog* ([SCHWABER; SUTHERLAND, 2020](#)), o qual, por sua vez, passará por um processo de priorização por *MoSCoW* ([WIEGERS; BEATTY, 2016](#)).

3.3 Gerenciamento de Riscos

O gerenciamento de riscos do projeto será realizado de acordo com o que foi proposto pelo PMBOK em sua 6ª edição (PMI, 2017). O sistema proposto inclui os seguintes passos: condução do planejamento; identificação dos riscos; análise qualitativa e quantitativa dos riscos; planejamento das respostas aos riscos; implementação das respostas e monitoramento dos riscos em um projeto. Estes passos são explicados em mais detalhes a seguir:

1. *Planejar o Gerenciamento dos Riscos*: O processo de definição de como conduzir as atividades de gerenciamento dos riscos de um projeto.
2. *Identificar os Riscos*: É o processo de identificação dos riscos individuais do projeto, bem como fontes de risco geral do projeto, e de documentar suas características.
3. *Realizar a Análise Qualitativa dos Riscos*: O processo de priorização de riscos individuais do projeto para análise ou ação posterior, através da avaliação de sua probabilidade de ocorrência e impacto, assim como outras características.
4. *Realizar a análise quantitativa dos riscos*: O processo de analisar numericamente o efeito combinado dos riscos individuais identificados no projeto e outras fontes de incerteza nos objetivos gerais do projeto.
5. *Planejar as Respostas aos Riscos*: O processo de desenvolver alternativas, selecionar estratégias e acordar ações para lidar com a exposição geral de riscos, e também tratar os riscos individuais do projeto.
6. *Implementar Respostas a Riscos*: O processo de implementar planos acordados de resposta aos riscos.
7. *Monitorar os Riscos*: O processo de monitorar a implementação de planos acordados de resposta aos riscos, acompanhar riscos identificados, identificar e analisar novos riscos, e avaliar a eficácia do processo de risco ao longo do projeto.

Este processo é importante para o projeto porque, segundo o PMBOK (PMI, 2017), estes passos vão trabalhar para aumentar a probabilidade de ocorrência e/ou impacto dos riscos positivos e diminuir a probabilidade de ocorrência e/ou impacto dos riscos negativos. Isto teria como resultado final a otimização das chances de sucesso do projeto.

3.4 Coleta de Dados

Na etapa de coleta de dados, foi feita uma pesquisa por meio do Google Forms para *streamers* com quem obtivemos contato. O questionário continha o total de 10 perguntas

abrangendo de questões sobre a profissão de *streamer*, os hábitos de transmissão e contatos com espectadores e outros *streamers*, sendo essas perguntas:

- Você se considera um streamer profissional?
- Transmissões são uma fonte de renda para você?
- Quando transmite, qual a duração da stream média?
- Você já fez alguma transmissão que já durou 10 ou mais horas? Se sim, como foi?
- Você pausa suas streams com que frequência?
- Que tipo de contato você tem com seus espectadores?
- Geralmente, você acha que o número de viewers influencia na sua motivação de continuar transmitindo?
- Você normalmente transmite o que você quer ou o que você acha que os viewers vão gostar?
- Você transmite com que frequência?
- Você acompanha o conteúdo de outros streamers?

No total, foram 7 respondentes da pesquisa, cujas respostas estão anexas abaixo.

Você se considera um streamer profissional?

7 responses

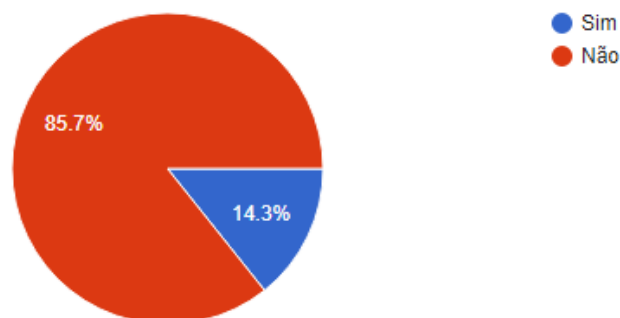


Figura 2 – Primeira Pergunta do Questionário.

Transmissões são uma fonte de renda para você?

7 responses

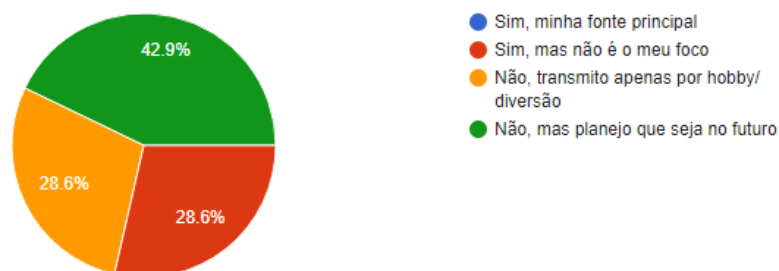


Figura 3 – Segunda Pergunta do Questionário.

Quando transmite, qual a duração da stream média?

7 responses

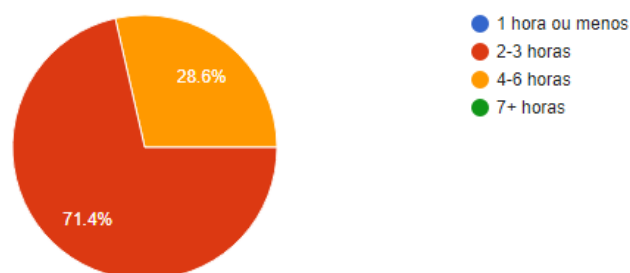


Figura 4 – Terceira Pergunta do Questionário.

Você já fez alguma transmissão que já durou 10 ou mais horas? Se sim, como foi?

7 responses

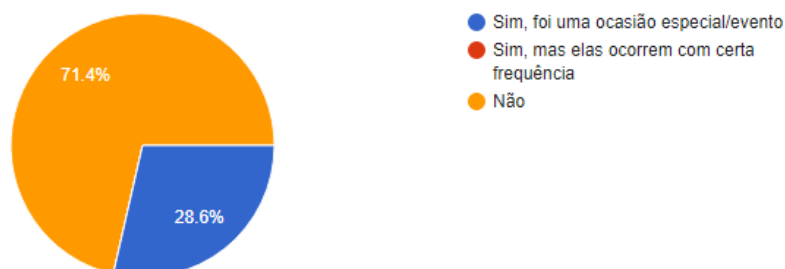


Figura 5 – Quarta Pergunta do Questionário.

Você pausa suas streams com que frequência?

7 responses

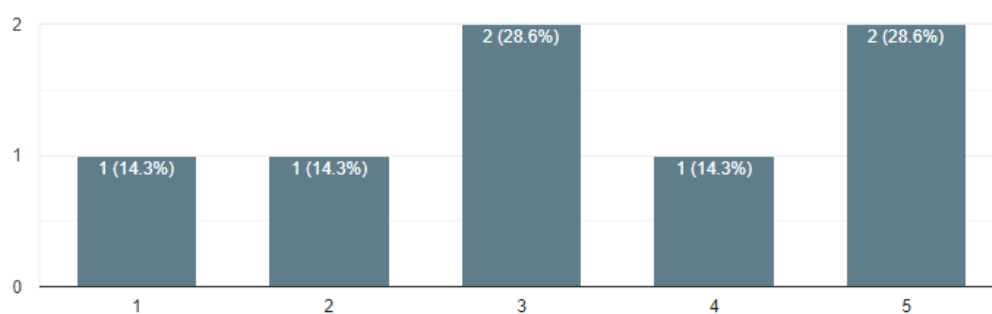


Figura 6 – Quinta Pergunta do Questionário.

Você pausa suas streams com que frequência?

7 responses

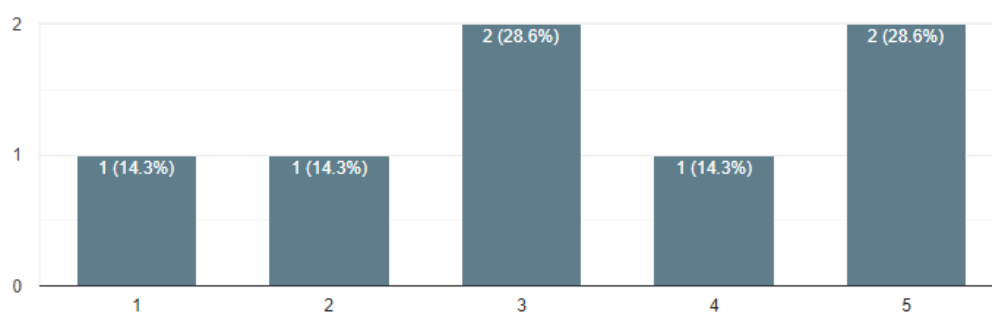


Figura 7 – Quinta Pergunta do Questionário.

Que tipo de contato você tem com seus espectadores?

7 responses

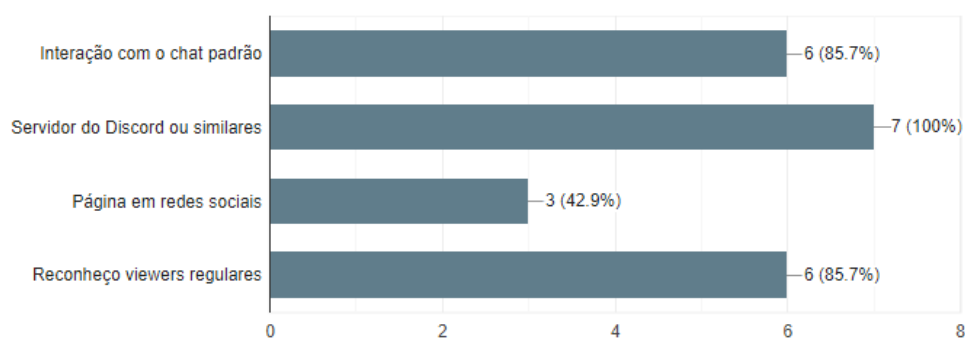


Figura 8 – Sexta Pergunta do Questionário.

Geralmente, você acha que o número de viewers influencia na sua motivação de continuar transmitindo?

7 responses

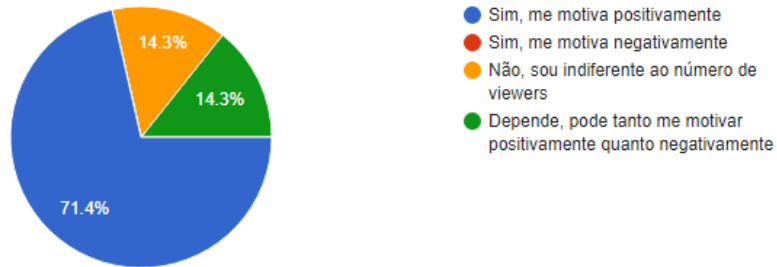


Figura 9 – Sétima Pergunta do Questionário.

Você normalmente transmite o que você quer ou o que você acha que os viewers vão gostar?

7 responses

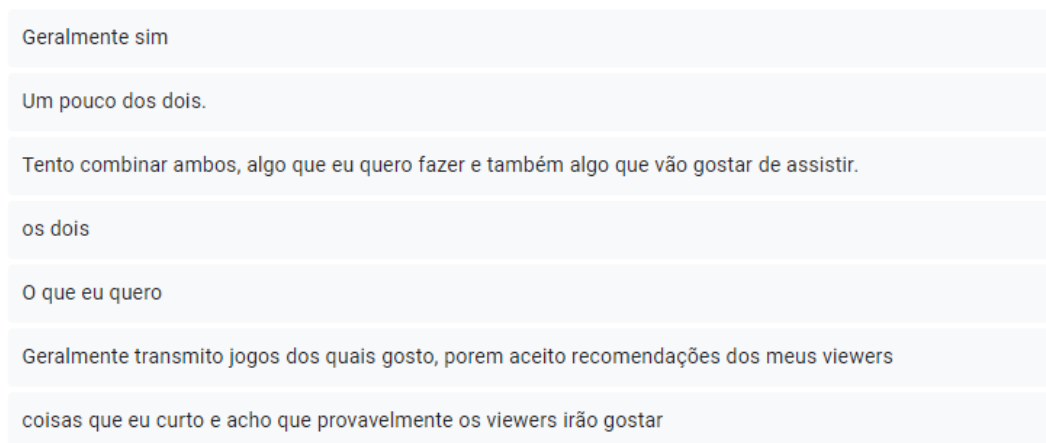


Figura 10 – Oitava Pergunta do Questionário.

Você transmite com que frequência?

7 responses

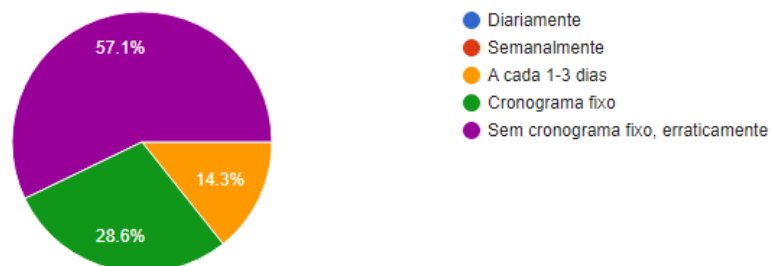


Figura 11 – Nona Pergunta do Questionário.

Você acompanha o conteúdo de outros streamers?

7 responses

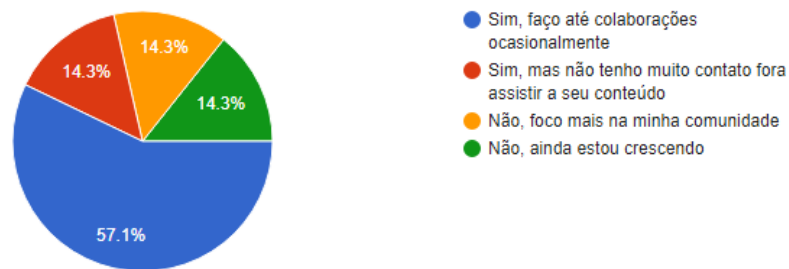


Figura 12 – Décima Pergunta do Questionário.

Devido ao pequeno conjunto de dados de apenas sete respondentes, não é possível tirar grandes conclusões acerca do público *streamer* como um todo. Porém, para nosso recorte de respondentes podem chegar a algumas conclusões: são, na maioria, pessoas que buscam fazer transmissões como forma de lazer e algo que potencialmente pode levar a fazer dinheiro, mas não necessariamente sendo este o foco. O que motiva esse grupo é principalmente a criação de um senso de comunidade que são proporcionados ao ter um canal com espectadores regulares, dentro e fora do ambiente de transmissão.

4 Planejamento do Sistema

4.1 Arquitetura da Solução

4.1.1 Visão Geral

A parte arquitetural tem a função de especificar e documentar decisões relevantes na produção e implementação do projeto, descrevendo os aspectos do sistema de forma clara, estruturada e objetiva. Bem como o modelo de representação dos serviços implementados e as interações estabelecidas entre esses serviços, do mesmo modo que a natureza dessas interações.

O escopo da arquitetura da solução envolve a implementação de uma aplicação *mobile*, utilizando a linguagem de programação *Flutter* como Front-End e microsserviços em Flask, um *micro-framework* em *Python* no Back-End, além do banco de dados relacional PostgreSQL no armazenamento dos dados.

4.1.2 Modelo Cliente-Servidor

O modelo cliente/servidor minimizou cada vez mais o tempo de desenvolvimento de aplicativos, devido a divisão das funções de compartilhamento de informações entre os ambos sistemas. O cliente é o solicitante, enquanto o servidor é o provedor do serviço. Na maioria dos ambientes cliente-servidor, o processamento de dados é feito pelo servidor, e os resultados são devolvidos aos clientes, o que é feito para acelerar a taxa de desempenho (KAMBALYAL, 2010) e escalabilidade. Por exemplo, em uma estação de trabalho, uma impressora pode ser conectada a um computador (representando os clientes), enquanto outros computadores que são compartilhados a partir dele são os servidores.

O sistema cliente-servidor tornou-se tão popular pois está sendo usado praticamente todos os dias para diferentes aplicações. Alguns dos protocolos padronizados que o cliente e os servidores utilizam para se comunicar são: *File Transfer Protocol (FTP)*, *Simple Mail Transfer Protocol (SMTP)* e Protocolo de Transferência de Hipertexto (*HTTP*). Assim, o sistema cliente-servidor pode ser definido como uma arquitetura de software composta tanto pelo cliente quanto pelo servidor, onde os clientes sempre enviam solicitações enquanto o servidor responde às solicitações enviadas (KRATKY; REICHENBERGER, 2013). O cliente-servidor fornece uma comunicação entre processos porque envolve a troca de dados do cliente e do servidor, onde cada um deles desempenha funções diferentes (KRATKY; REICHENBERGER, 2013).

Dentre os benefícios do modelo cliente-servidor estão: divisão do processamento da

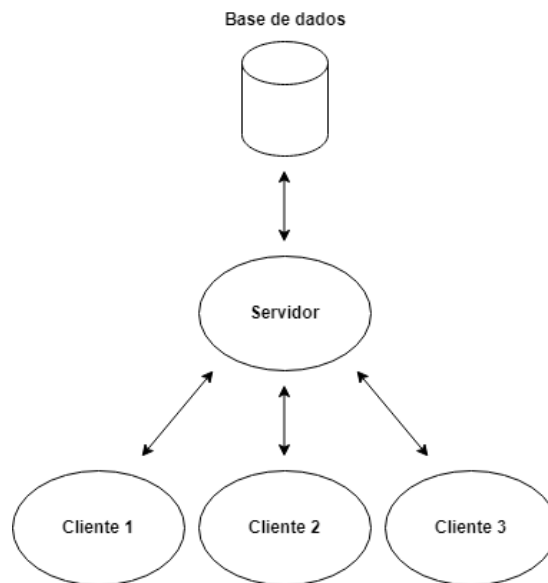


Figura 13 – Comunicação entre processos entre cliente e servidor.

aplicação em várias máquinas, conforme descrito na figura acima (KAMBALYAL, 2010). Além da maior facilidade no compartilhamento de recursos do cliente para os servidores e na redução a da replicação de dados armazenados em cada servidor em vez de no cliente. Dito isso, sendo o modelo mais adequado para o contexto atual da aplicação.

4.1.3 Tecnologias

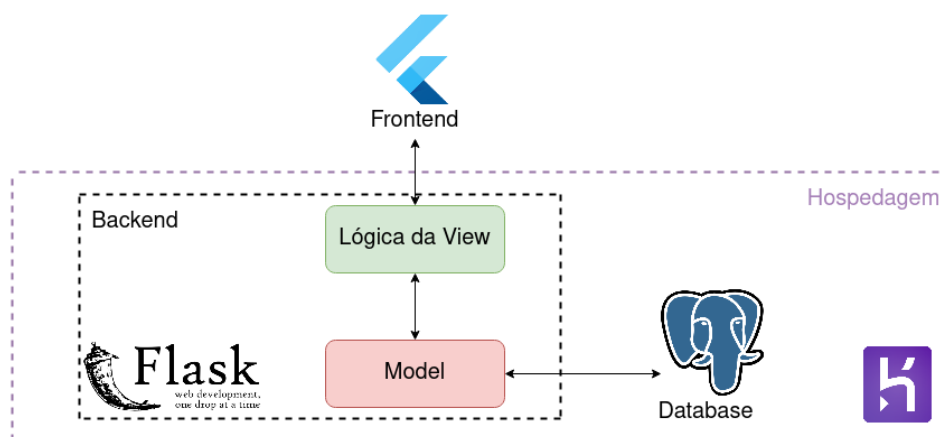


Figura 14 – Diagrama de Arquitetura.

4.1.3.1 Front-End

A aplicação *mobile*, implementada utilizando a framework Flutter, disponibilizará uma interface de comunicação entre o Back-end e o usuário. Dessa forma, será possível gerenciar as atividades que serão realizadas dentro do aplicativo.

- Flutter: Kit de ferramentas de interface de usuário da Google para criar aplicativos belos e compilados nativamente para dispositivos móveis, Web e *Desktop* a partir de uma única base de código. É um framework que possui como linguagem base o *Dart*.

4.1.3.2 Back-End

O Back-End refere-se a qualquer parte de uma aplicação ou programa de software que os usuários não enxerga. Ele contrasta com o Front-End, que se refere à interface de usuário, sendo a camada de acesso a dados, onde ficam as logicas negociais.

- Flask: *Micro-framework* de *Python*, possui toda a flexibilidade da linguagem e provê um modelo simples para desenvolvimento web. É baseado em 3 pilares: *Werkzeug*, uma biblioteca para desenvolvimento de aplicações *WSGI*, *Jinja2*, um *template engine* escrito em *Python* e *good intentions*, que são alta qualidade na legibilidade, liberdade de estruturar a aplicação na maneira que desejar, entre outros aspectos.

4.1.3.3 Banco de Dados

O Banco de Dados se trata da parte de uma aplicação responsável pelo armazenamento e gerência de dados, dicionários e outros objetos relevantes para seus respectivos modelos, geralmente por meio de um Sistema Gerenciador de Banco de Dados (SGBD). Ele nunca é acessado diretamente pelo usuário, sendo suas chamadas feitas por meio do Back-End cuja função é resgatar estes dados armazenados.

- PostgreSQL: Um *SGBD* de código aberto *Objeto-Relacional* que expande a linguagem *SQL* ao mesmo tempo que tenta não contradizer seus princípios básicos.

4.1.3.4 Modelo de Dados

O modelo de dados a ser utilizado para aplicação está anexo na figura abaixo.

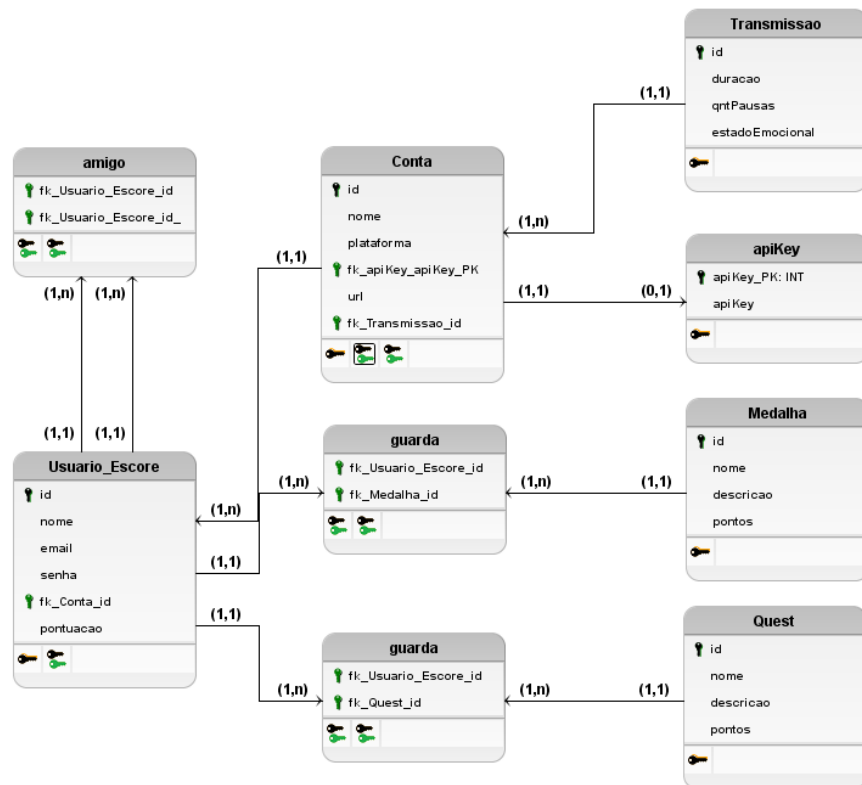


Figura 15 – Modelo de Banco de Dados

4.1.3.5 Diagrama de Pacotes

O Diagrama de Pacotes tem como objetivo estruturar hierarquicamente as diversas pastas que compõem o projeto, representando as dependências lógicas que elas possuem entre si. O diagrama de pacotes do *backend* do projeto está representados na Figura 16, e o diagrama de pacotes do *frontend* do projeto está representado na Figura 17

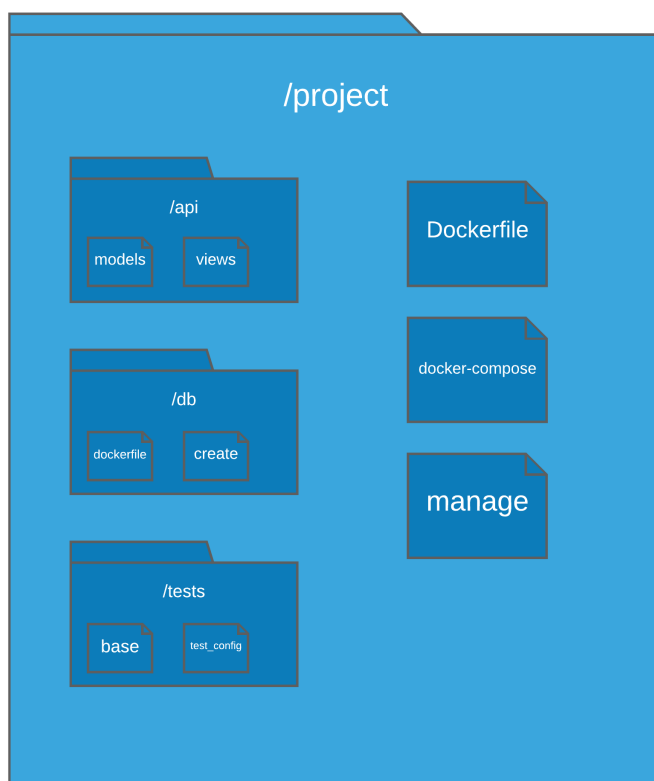


Figura 16 – Diagrama de Pacotes - Flask.

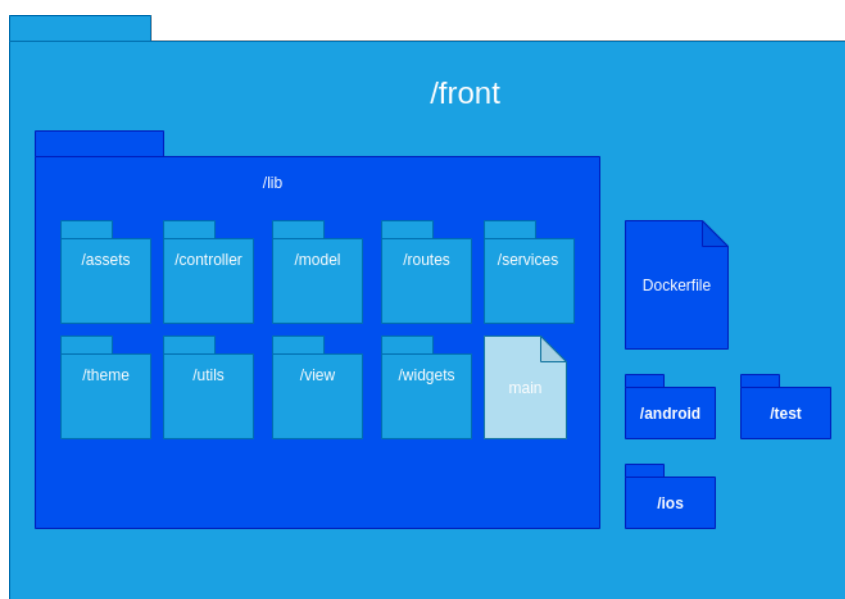


Figura 17 – Diagrama de Pacotes - Flutter.

4.2 Requisitos

Os requisitos funcionais da aplicação foram levantadas através de um processo de *brainstorming* sobre as respostas dos questionários que foram respondidos por *streamers* conhecidos pela equipe de desenvolvimento, priorizados através do processo MoSCoW, e disponibilizados na tabela abaixo.

Tabela 1 – Requisitos Funcionais

ID	Requisito	Prioridade
RG001	Fazer login com email	<i>Must Have</i>
RG003	Timer de duração de <i>stream</i>	<i>Must Have</i>
RG004	Relatório com a duração média semanal/mensal	<i>Must Have</i>
RG005	Alerta se a <i>stream</i> passar de uma duração limite (sugerir pausa)	<i>Must Have</i>
RG006	Pergunta sobre o estado mental do usuário na primeira vez que o app é aberto no dia	<i>Must Have</i>
RG007	Pergunta sobre o estado mental do usuário assim que uma <i>stream</i> termina	<i>Must Have</i>
RG009	Recompensas se o usuário responder as perguntas sobre o estado mental dele com frequência	<i>Must Have</i>
RG010	Mensagem de consolação se a resposta após uma <i>stream</i> indicar piora do estado mental	<i>Must Have</i>
RG017	Indicar ajuda proissional se necessário	<i>Must Have</i>
RG008	Lista de amigos	<i>Should Have</i>
RG011	Sugerir mudar de jogo se o estado mental da pessoa piorar muito	<i>Should Have</i>
RG015	Disponibilizar recompensas para cada resposta	<i>Should Have</i>
RG016	Indicar alongamentos/exercícios/meditação	<i>Should Have</i>
RG002	OAuth com Discord ou Twitch	<i>Could Have</i>
RG012	Mostrar um <i>leaderboard</i> de respostas consecutivas entre amigos	<i>Could Have</i>
RG013	Possibilitar um planejamento de cronograma das <i>streams</i>	<i>Could Have</i>
RG014	Disponibilizar uma sala de convivência dos <i>streamers</i> com a audiência	<i>Could Have</i>

4.3 Configuração de ambiente

Se tratando do *backend*, o ambiente de desenvolvimento será configurado através do gerenciador de pacotes *poetry* e a tecnologia de containerização *Docker*. Já no *frontend*, como foi decidido a utilização do framework *Flutter*, serão utilizadas as soluções nativas do framework.

4.3.1 Poetry

Poetry é um gerenciador de pacotes para o ecossistema *Python* que facilita o manejo de dependências. Diferente da solução normalmente utilizada, o *pip* (Package Installer for Python), o poetry possibilita a identificação de dependências não utilizadas, facilitando assim a remoção destas. Além disso o poetry também separa os contextos das dependências de desenvolvimento e produção, facilitando assim a instalação de ferramentas de análise estática, formataadores automáticos e bibliotecas de teste.

4.3.2 Docker

Docker é uma plataforma aberta de containerização de software. Ele funciona isolando o ambiente de execução do software do resto do Sistema Operacional da máquina em que ele está hospedado possibilitando o acesso a apenas alguns recursos. Isto facilita o isolamento de dependências e possibilita a reprodução do mesmo ambiente em várias máquinas diferentes. Tendo em mente estes pontos, chegamos a conclusão de que o Docker seria uma ótima ferramenta para a containerização tanto para o ambiente de desenvolvimento quanto para o ambiente de produção no processo de entrega.

4.4 Construção e Entrega



Figura 18 – Diagrama de Construção e Entrega.

Para maior agilidade no desenvolvimento da aplicação, o processo de construção e entrega do software será automatizado utilizando diversos serviços disponíveis sem custos para aplicações *open-source*.

4.4.1 Controle de Versão

O controle de versão do software será feito utilizando o *git*, um software que monitora as alterações feitas em uma determinada base de código, mostra as diferenças aplicadas e facilita na unificação das alterações feitas mesmo em processos de trabalho assíncronos. Já a hospedagem do repositório *git* será feito pela plataforma *GitHub*.

4.4.2 Integração Contínua

O processo de integração contínua será feito utilizando o github actions que, após cada atualização ou solicitação de alteração, executa comandos pre-definidos. Estes comandos serão usados para executar um verificador de folha de estilo no código, executar todos os testes automatizados e verificar que nenhum deles esteja falhando, além de coletar a cobertura de testes para verificar se os estes, escritos, precisam ser melhorados.

4.4.3 Análise Estática

O processo de Análise estática do código será feita pela plataforma *Code Climate*, que exibirá os resultados da medição de cobertura de testes feitas no passo de Integração Contínua e, além disso, verificará o código escrito em busca de padrões de implementação que são considerados hostis para a manutenibilidade do código. Após todo esse processo, a plataforma dará para o software uma nota de “A” até “F” dependendo do tempo que se levaria para fazer uma alteração qualquer na base de código.

4.4.4 Entrega

A entrega da aplicação será feita pela plataforma *Heroku*. O processo será realizado de maneira automatizada por um procedimento realizado pelo *GitHub Actions* após a verificação dos testes e da folha de estilo; a aplicação então será hospedada em uma instância de máquina virtual oferecida, sem custo, com um limite de execução de 1000 minutos por mês.

4.5 Escopo Planejado

Devido ao prazo curto para a execução do trabalho, foi decidido executar apenas os requisitos definidos como *Must Have* utilizando a priorização MoSCoW. Estes requisitos foram incluídos no planejamento do cronograma abaixo, com os demais requisitos priorizados como *Should Have* e *Could Have* sendo deixados para projetos futuros.

4.6 Cronograma do Desenvolvimento

Estipulando a duração de cada *sprint* para 2 semanas, são planejadas 5 *sprints* para a entrega do MVP (*Minimum Valuable Product*).

Tabela 2 – Cronograma

Nº Sprint	ID Requisito(s)	Data inicial	Data Final
Sprint 1	RG001 & RG003	18/07/2022	31/07/2022
Sprint 2	RG004 & RG005	01/08/2022	14/08/2022
Sprint 3	RG006 & RG007	15/08/2022	28/08/2022
Sprint 4	RG009 & RG0010 & RG017	29/08/2022	11/09/2022

Após a *Sprint* 4, ainda teremos à disposição 1 semana para revisar as dívidas técnicas.

5 Resultados do Desenvolvimento

5.1 Visão Geral

Para o desenvolvimento da aplicação, foi priorizado o fluxo principal de autenticação e registro do usuário, visualização das informações dos usuários, *timer* para contagem do tempo de uma *stream* em andamento e subsequente registro desta no banco de dados, perguntas e respostas sobre o estado psicológico do usuário e, finalmente, contagem a acúmulo de pontos de usuários que respondem às perguntas com frequência.

5.1.1 MVP

Devido ao semestre atípico em que este trabalho foi desenvolvido, o escopo do trabalho foi reduzido para um MVP (*Minimal Value Product*), ou seja, o mínimo de funcionalidades para que seja considerado um produto funcional. Para isto, foram escolhidos somente os requisitos funcionais priorizados como *Must-Have*. O produto entregue ser um MVP implica que, após o período inicial de desenvolvimento, caso se verifique interesse real no produto proposto, haverá melhoras incrementais progressivas até se obter um produto completo.

5.1.2 A Escala de Bem-Estar

No contexto de uma versão mais completa do produto seria necessário fazer uma pesquisa mais a fundo de quais perguntas melhor capturariam o estado mental presente do usuário, atribuir pesos dinâmicos para cada uma das respostas e rotacionar estas perguntas cada vez que fossem exibidas. Porém devido ao contexto de MVP do escopo do produto, foi decidido utilizar um processo mais simplificado para dar prosseguimento ao desenvolvimento da aplicação, deixando este processo ideal para projetos futuros.

Com isso, como *placeholder* para o MVP foi escolhida uma pergunta utilizada no “Teste de Satisfação Geral com a Vida” da Universidade da Pensilvânia, que por sua vez prevê 4 possibilidades de resposta: “Concordo Totalmente”, “Concordo Parcialmente”, “Discordo Parcialmente” e “Discordo Totalmente”. Cada uma destas perguntas podem subtrair ou somar pontos à escala de bem-estar do usuário, que por conveniência foi decidida que seria uma escala simples de 0 até 14 representando respectivamente o pior e o melhor estados mentais possíveis.

A principal funcionalidade da aplicação é a flutuação desta pontuação de bem-estar, com o fato de que se a pontuação chegar a 0, e o usuário ainda assim responder

negativamente à pergunta feita, a aplicação recomendará a procura de ajuda profissional.

5.1.3 Back-end

Seguindo exatamente o que foi proposto pelo planeamento de arquitetura, foi implementada uma *API* baseada em *Flask*, utilizando *Poetry*, *Docker* e *Docker-Compose* para sua *containerização*. Para integração e entrega contínua, foi usado o *CodeClimate* onde são rodados os testes unitários e testes de integração a cada commit. Este ambiente todo está hospedado na nuvem através do *Heroku*, que é acessado posteriormente com suas devidas autenticações pela nossa camada de *Front-end*.

5.1.4 Front-End

Do mesmo modo, foi implementada uma aplicação *Flutter* que se comunica com o ambiente *Back-end* hospedado no *Heroku*. Entre os requisitos elicitados, todos os da categoria *MUST* foram feitos e com suas respectivas telas acessíveis ao usuário. Dentre elas, temos:

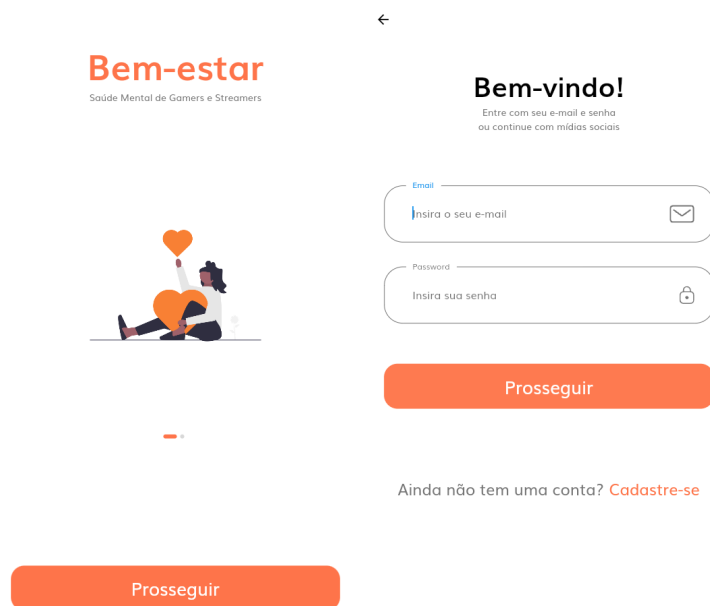


Figura 19 – Telas de Início e Login.

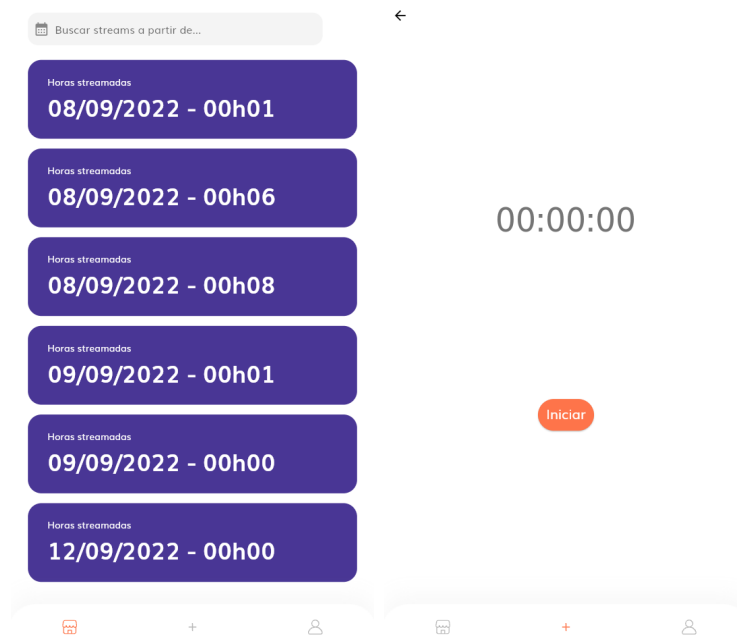


Figura 20 – Telas Principal e Timer.

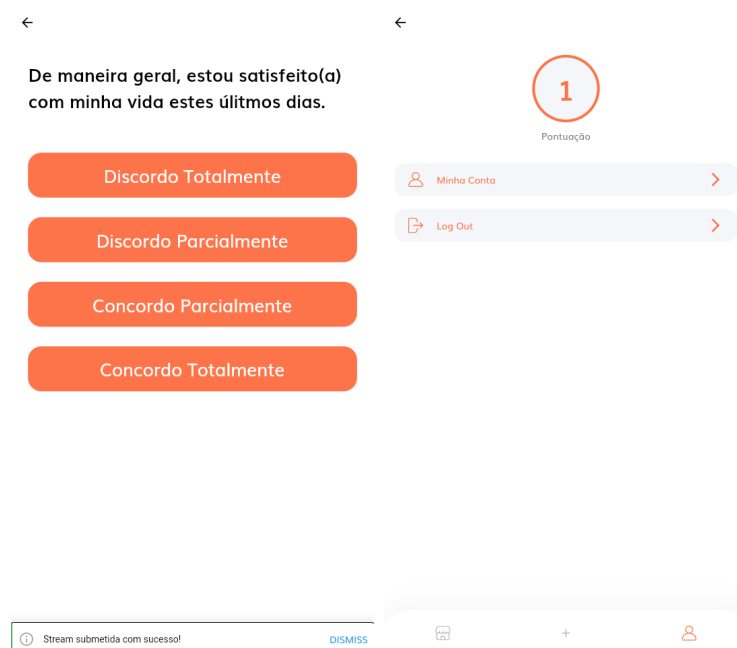


Figura 21 – Tela Fim da Stream e Perfil.

5.2 Testes

5.2.1 Cobertura de Testes e Análise Estática

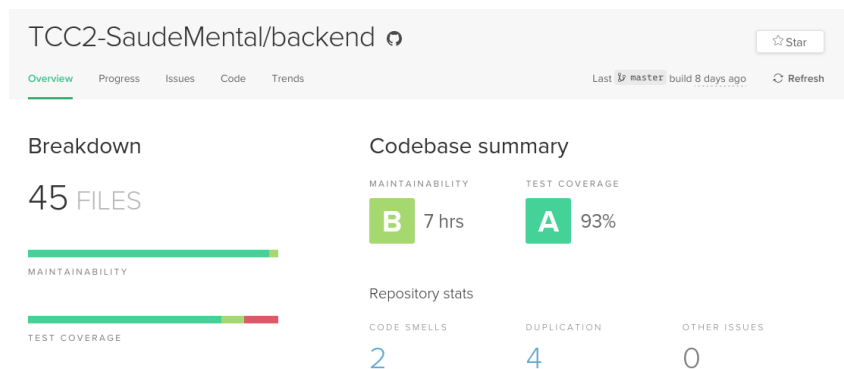


Figura 22 – Code Climate do Back-end.

Como mencionado anteriormente, foi utilizado o Code-Climat como ferramenta auxiliar na integração contínua do projeto, acusando erros quando não passavam e mantendo o controle da manutenibilidade do código e cobertura de testes.

De acordo com a ferramenta, o código implementado tem o ranking B em manutenibilidade devido à algumas repetições de código, que embora não afetem o desempenho do projeto poderiam ser refatorados para melhor aproveitamento. Em quesitos de cobertura de testes, as únicas partes não testadas do código são partes que não são normalmente testadas, como arquivos de inicialização(*init*) vazios, dando um escore de 93 por cento.

5.2.2 Estratégia de testes

Os testes foram feitos utilizando o que Myers (1979) classifica como *black-box testing*, ou seja: os testes foram feitos orientados a *input* e *output*, com o ambiente de teste esperando que cada dado passado como parâmetro para uma determinada funcionalidade gere uma saída que esteja de acordo com as regras de negócio da aplicação; caso não estejam de acordo, o teste falha.

Utilizando essa estratégia geral, foram feitos testes de funcionalidade em todos os *endpoints* da aplicação utilizando um cliente HTTP de teste. Cada *endpoint* foi acessado programaticamente em todos os estados possíveis que a aplicação poderia se encontrar e suas respostas foram analisadas. Após os testes de funcionalidade, foram feitos testes unitários em todas as funções que não interagem diretamente com as funcionalidades dos *endpoints* da aplicação. Estes testes unitários exploram todos os resultados possíveis que estas funções podem retornar. O resultado geral da cobertura de código da aplicação gerada pela biblioteca *coverage.py* foi de 93%.

5.3 Privacidade e Segurança

No estado atual da aplicação, não é guardado nenhuma informação sobre o usuário além das a seguir: informações necessárias para a autenticação do usuário (nome, email, senha), escore de respostas do usuário que permite a recomendação de procura por ajuda profissional, data da última resposta para calcular a quantidade de pontos por resposta consecutiva o usuário tem.

A senha dos usuários é guardada no banco de dados apenas após de passar por um processo de *hashing* utilizando o algoritmo *bcrypt*, este processo não permite p descobrimento da senha original mesmo se houver vazamento dos dados. A autenticação dos usuários é feita por um token JWT (JSON Web Token) com tempo de expiração, o que garante que se o *token* for vazado, o suposto atacante poderá fazer nada com ele depois de passado seu tempo de expiração.

6 Conclusões

6.1 Considerações finais

Ao realizar-se a etapa inicial de revisão bibliográfica do estado da arte com relação aos estudos sobre a influência de jogos digitais no bem-estar e na saúde psicológica de jogadores, alguns conceitos e nuances sobre o assunto clarificaram-se e ganharam forma - em específico nas relações entre tempo de jogo e bem-estar do jogador -, apontando um norte confiável e melhor delimitado para os passos que se seguiram do presente projeto.

À etapa de planejamento e arquitetura, os conhecimentos que o time possui em Engenharia de Software possibilitaram o levantamento e a priorização de requisitos eficientes, planejamento de arquitetura coerente e detalhada, estratégia de construção e entrega completa, além de cronograma de desenvolvimento razoável. Apesar do escopo de desenvolvimento proposto a priori ter sido reduzido, devido à curta janela de tempo proporcionada pelo semestre, considera-se satisfatório o resultado obtido. A decisão de se utilizar a priorização MoSCoW foi fundamental nessa etapa, pois houve a necessidade de se delimitar quais requisitos seriam imprescindíveis (Must Have) de serem executados para o trabalho e quais poderiam se encaixar em categorias de Should Have e Could Have.

No tocante ao desenvolvimento do produto em si, o *back-end* demonstrou-se ser menos dificultoso que o *front-end*, talvez devido ao fato da equipe possuir maior experiência e familiaridade nas tecnologias de *back-end* escolhidas e utilizadas neste trabalho, como o Python, Flask, e PostgreSQL. O tempo de desenvolvimento do *back-end*, assim, deu-se em curto espaço de tempo, com a maior parte deste sendo utilizado para *Dockerização*, Testes, criação da *pipeline* de CI/CD e entrega via *Heroku*. Quanto ao desenvolvimento do *front-end* o processo transcorreu por caminhos mais tortuosos, devido à pouca experiência prévia da equipe com a tecnologia utilizada. Entretanto, vencido esse entrave inicial, foi possível extrair os proveitos esperados do *Flutter*, os quais foram os motivos de escolha dessa tecnologia: criar as funcionalidades propostas e entregar o MVP num aplicativo veloz e responsivo sem necessitar de programação nativa para o sistema *android*.

Durante o desenvolvimento do trabalho pudemos perceber que os desafios psicológicos que os *streamers* enfrentam não são exclusivos apenas a eles, mas, principalmente neste contexto pós-pandêmico, são desafios comuns às diversas profissões que precisam lidar autonomamente com as suas responsabilidades e horários. Ainda lidaremos com estes problemas por diversos anos e, com sorte, surgirão soluções novas para eles.

Por fim, a experiência de desenvolvimento deste projeto mostrou-se construtiva, pois além de conferir uma gratificante oportunidade para transpor para a prática o que

foi estudado em formato teórico ao longo do curso de Engenharia de Software, também possibilitou o estudo de áreas não correlatas às Engenharias, como é o caso dos tópicos referentes à bem-estar e saúde mental.

6.2 Trabalhos futuros

O principal problema enfrentado pela equipe durante o processo de desenvolvimento do trabalho foi a redução de escopo, advinda da curta janela de tempo proporcionada pelo semestre vigente. Posto isso, o primeiro caminho de evolução para dar continuidade ao trabalho seria prosseguir no desenvolvimento dos requisitos levantados na Tabela 1 da seção 4.2, iniciando pelos classificados como *Should Have* e, em ordem, prosseguindo para os classificados como *Could Have*. Isso proporcionaria mais oportunidades de desenvolver processos gamificados, o que ajudaria a melhorar a assiduidade dos usuários.

Uma interface bem projetada poderia ajudar o usuário a se manter engajado à plataforma, portanto também seria recomendado fazer testes de usabilidade com *streamers* conhecidos pelo time que estiver evoluindo o trabalho. Levantando, assim, novos requisitos funcionais e melhorias na interface de usuário. Esse processo poderia ser repetido ao longo do semestre, podendo também ser combinado à validação dos requisitos levantados anteriormente, proporcionando, assim, uma evolução contínua da plataforma.

Vale também ressaltar que, como o projeto trata de saúde mental de pessoas, seria imprescindível submetê-lo ao comitê de ética da Universidade de Brasília antes de continuar o seu desenvolvimento, principalmente se for necessário o recolhimento de mais informações de caráter pessoal dos usuários.

Referências

- ARSENAULT, D. Video game genre, evolution and innovation. *Eludamos: Journal for Computer Game Culture*, v. 3, n. 2, p. 149–176, Oct. 2009. Disponível em: <<https://septentrio.uit.no/index.php/eludamos/article/view/vol3no2-3>>. Citado na página 18.
- BONNAR, D. et al. Risk factors and sleep intervention considerations in esports: A review and practical guide. *Sleep Medicine Research*, p. 59–66, 2019. Citado na página 20.
- CABEZA-RAMÍREZ, L. J.; FUENTES-GARCÍA, F. J.; MUÑOZ-FERNANDEZ, G. A. Exploring the emerging domain of research on video game live streaming in web of science: State of the art, changes and trends. *International Journal of Environmental Research and Public Health*, v. 18, n. 6, 2021. ISSN 1660-4601. Disponível em: <<https://www.mdpi.com/1660-4601/18/6/2917>>. Citado na página 19.
- CAMBRIDGE. *Cambridge Dictionary*. 2022. Disponível em: <<https://dictionary.cambridge.org/dictionary/english/>>. Citado na página 18.
- CHAN, J. Y. T. F. K. Acceptance of agile methodologies: A critical review and conceptual framework. v. 46, p. 803–814, 2008. Citado na página 24.
- CHOU, Y.-K. Actionable gamification: beyond points ,badges, and leaderboards. Octalysis media, 2016. Citado 2 vezes nas páginas 21 e 22.
- DUBOIS, D. J.; TAMBURRELLI, G. Understanding gamification mechanisms for software development. In 9th Proceedings of the 2013 Joint Meeting on Foundations of Software Engineering (ESEC/FSE), p. 659–662, 2013. Citado na página 21.
- EDGE, N. Evolution of the gaming experience: Live video streaming and the emergence of a new web community. 2013. Disponível em: <<https://eloncdn.blob.core.windows.net/eu3/sites/153/2017/06/03NathanEdgeEJFall13.pdf>>. Citado na página 19.
- GALDERISI, S. et al. Toward a new definition of mental health. *World Psychiatry*, p. 231–233, 2015. Citado 2 vezes nas páginas 17 e 18.
- GJESTANG, R. Working as a live streamer, effect of audience interaction on psychosocial work environment. Norwegian University of Science and Technology, 2020. Citado 3 vezes nas páginas 14, 19 e 20.
- JOHNSON, M. R.; WOODCOCK, J. The impacts of live streaming and twitch.tv on the video game industry. *Media, Culture & Society*, 41(5) pp. 670–688., 2019. Citado 2 vezes nas páginas 20 e 21.
- KAMBALYAL, C. 3-tier architecture. retrieved on, 2. 2010. Citado 2 vezes nas páginas 31 e 32.
- KRATKY, S.; REICHENBERGER, C. Client/server development based on the apple event object model. Atlanta, 2013. Citado na página 31.

- MARCZEWSKI, A. Gamification: A simple introduction and a bit more. p. 24, 2013. Citado na página 21.
- MYERS, G. J. *The art of software testing*. Hoboken, New Jersey: John Wiley & Sons, Inc., 1979. Citado na página 43.
- ORGANIZAÇÃO MUNDIAL DA SAÚDE. *Constitution of the World Health Organization*. 2006. Disponível em: <https://www.who.int/governance/eb/who_constitution_en.pdf>. Citado na página 17.
- ORGANIZAÇÃO MUNDIAL DA SAÚDE. *Mental health: Strengthening our response*. 2018. Disponível em: <<https://www.who.int/news-room/fact-sheets/detail/mental-health-strengthening-our-response>>. Citado na página 17.
- ORJOUX, A. *Body pulled from the East River is that of missing YouTuber Etika, New York police say*. 2019. Disponível em: <<https://edition.cnn.com/2019/06/25/us/new-york-missing-youtuber-found-dead/index.html>>. Citado na página 15.
- OXFORD. *Oxford English Dictionary*. 2022. Disponível em: <<https://www.oed.com>>. Citado na página 21.
- PELDON, S. *Why did Etika kill himself? The YouTuber's mental health issues consumed him till his death*. 2022. Disponível em: <<https://www.voxbliss.com/why-did-etika-kill-himself/>>. Citado na página 15.
- PROJECT MANAGEMENT INSTITUTE. *A Guide to the Project Management Body of Knowledge (PMBOK Guide)*. Newtown Square, PA, 2017. Citado na página 25.
- ROUSSOU, M. Learning by doing and learning through play: An exploration of interactivity in virtual. *ACM Journal of Computers in Entertainment, Educating Children Through Entertainment*, ACM Press, New York, NY, USA, 2004. Citado na página 21.
- SAUTER, M.; BRAUN, T.; MACK, W. Social context and gaming motives predict mental health better than time played: An exploratory regression analysis with over 13,000 video game players. *Institute for Psychology, Bundeswehr University Munich*, 2020. Citado na página 19.
- SCHWABER, K.; SUTHERLAND, J. *The 2020 Scrum Guide*. 2020. Disponível em: <<https://scrumguides.org/scrum-guide.html>>. Citado na página 24.
- TAVINOR, G. Definition of videogames. v. 6, 2008. Citado na página 18.
- WEFLEN CAMERON A. MACKENZIE, I. V. R. E. An influence diagram approach to automating lead time estimation in agile kanban project management. v. 1, p. 11, 2022. Citado na página 24.
- WIEGERS, K.; BEATTY, J. *Moscow or Kano - how do you prioritize?* 2016. Disponível em: <<https://www.hotpmo.com/management-models/moscow-kano-prioritize/>>. Citado na página 24.
- YU, V.; ALVAREZ, K. P. B.; CHEN, V. H. H. Game streamers' practices on twitch and management of well-being. *Nanyang Technological University, Singapore*, 2021. Citado 3 vezes nas páginas 14, 20 e 21.