

Universidade do Minho
Escola de Engenharia

Mestrado em Engenharia Informática



Sistemas Interativos Confiáveis Arquitetura de Aplicações

Grupo 7

Ano Letivo de 2021/2022

PG47164, Eduardo Coelho
PG47578, Pedro Veloso
PG47089, Carlos Preto
PG47398, Leonardo Marreiros
PG47559, Pedro Fernandes

Gualtar, Junho de 2022

Conteúdo

1	Introdução	1
1.1	Propósito do sistema	1
1.1.1	Motivação e Contextualização	1
1.1.2	Objetivos do projeto	2
1.2	Cliente, <i>customer</i> e outros <i>Stakeholders</i>	2
1.2.1	Cliente	2
1.2.2	<i>Customer</i>	2
1.2.3	<i>Stakeholders</i>	2
2	Utilizadores do sistema	4
2.1	<i>Gamer</i>	4
2.2	Administrador	4
3	Restrições do projeto	5
3.1	Restrições da solução	5
3.2	Ambiente de Implementação do Sistema Atual	5
3.3	Aplicações parceiras	5
3.4	Ambiente da utilização do sistema	6
3.5	Restrições de calendário	6
3.6	Restrições de orçamento	6
4	Taxonomia e definições	7
5	Factos e assunções	9
5.1	Decisões Legais e Políticas	9
5.2	Ambiente de Desenvolvimento	9
5.3	Dependências Externas	9
5.4	Componentes Disponíveis	9
6	Âmbito do projeto	10
6.1	Situação atual	10
6.2	Contexto do trabalho	10
7	Âmbito do sistema	11
7.1	Registrar	12
7.2	<i>Login/Logout</i>	13
7.3	Aceder a <i>feed</i>	14
7.4	Juntar a grupo	15
7.5	Criar <i>post</i>	16
7.6	Seguir jogo	19
7.7	Enviar mensagem	20
8	Requisitos funcionais	22
9	Requisitos não funcionais	28
9.1	Requisitos de Aparência	28
9.2	Requisitos de Usabilidade	29
9.3	Requisitos de Performance	30
9.4	Requisitos Operacionais	30
9.5	Requisitos de Manutenção e Suporte	31

9.6 Requisitos de Segurança	32
9.7 Requisitos Legais	32
10 Alcance do sistema e contexto	33
10.1 Contexto de negócio	33
10.1.1 APIs externas	33
10.1.2 Base de dados	34
10.1.3 Utilizador	34
10.2 Contexto técnico	34
11 Vista de blocos de construção	36
11.1 Nível 1	36
11.1.1 Componente - Frontend	36
11.1.2 Componente - Backend	36
11.1.3 Componente - AtribuiRecompensas	37
11.1.4 Componente - RealizaPostsNoticias	37
11.2 Nível 2 - Backend	37
11.3 Nível 3 - Backend	39
12 Vista de execução	41
12.1 Eliminar Post	41
12.2 Banir Jogador	41
12.3 Registar	41
12.4 Login	41
12.5 Enviar mensagem	41
12.6 Aceder feed	42
12.7 Adicionar amigo	43
12.8 Avaliar jogador	43
12.9 Juntar a grupo	43
12.10 Seguir jogo	43
12.11 Editar perfil	43
12.12 Criar post grupo	44
12.13 Criar post multimédia	44
12.14 Avaliar post	44
13 Vista dos dados	45
13.1 Dicionário de dados	45
13.2 Estimativa do espaço em disco da base de dados e taxa de crescimento anual	47
14 Vista de deployment	48
14.1 Infraestrutura	48
14.1.1 Servidor da WebApp	49
14.1.2 Servidor do Load Balancer	49
14.1.3 Servidor do GameHub	50
14.1.4 Servidor da base de dados	50
14.2 Adicionar servidores do GameHub em runtime	50
15 Análise de escalabilidade	51
15.1 JMeter	51
15.1.1 100 Clientes	52
15.1.2 200 Clientes	53
15.1.3 500 Clientes	54
15.1.4 1000 Clientes	55

15.1.5 1500 Clientes	56
16 Manual de utilização	58
16.1 <i>Gamer</i>	58
16.1.1 Registrar	58
16.1.2 Login	59
16.1.3 Editar perfil	61
16.1.4 Seguir novo jogo	65
16.1.5 Deixar de seguir jogo	66
16.1.6 Aceder <i>feed</i> de um jogo	68
16.1.7 Criar publicação grupo	69
16.1.8 Juntar a grupo	71
16.1.9 Sair de um grupo	74
16.1.10 Enviar mensagem grupo	75
16.1.11 Adicionar amigo	77
16.1.12 Remover amigo	81
16.1.13 Avaliar jogador	83
16.1.14 Receber notificação de mensagem	85
16.1.15 Enviar mensagem privada	85
16.1.16 Criar publicação multimédia	87
16.1.17 Eliminar publicação multimédia	90
16.1.18 Avaliar publicação	93
16.1.19 Logout	95
16.2 Administrador	97
16.2.1 Eliminar publicação multimédia	97
16.2.2 Eliminar publicação grupo	99
16.2.3 Banir utilizador	100
17 Análise de usabilidade	102
17.1 <i>Cognitive Walkthrough</i>	102
17.1.1 Perito 1	102
17.1.2 Perito 2	105
17.1.3 Perito 3	108
17.1.4 Conclusões do <i>Cognitive Walkthrough</i>	111
17.2 SUS - <i>System Usability Scale</i>	112
17.2.1 Participantes:	112
17.2.2 Resultados	112
18 Think aloud	114
19 Conclusão	115
20 Anexos	117
20.1 <i>Cognitive Walkthroughs</i> - Formulários	117
20.1.1 Registrar	117
20.1.2 Criar <i>post</i> de grupo	119
20.1.3 Criar <i>post</i> multimédia	120
20.1.4 Seguir novo jogo	120
20.1.5 Juntar a equipa	121

Índice de figuras

1.1	.	.	.	1
4.1	Modelo de domínio	.	.	7
7.1	Modelo de Use Cases	.	.	11
7.2	Registrar	.	.	12
7.3	<i>Login/Logout</i>	.	.	13
7.4	Aceder ao <i>feed</i>	.	.	14
7.5	Juntar a grupo - Modelo de tarefas	.	.	15
7.6	Juntar a grupo	.	.	16
7.7	Criar <i>post</i> - Diagrama de tarefas	.	.	17
7.8	Criar <i>post</i> para a publicação de multimédia	.	.	17
7.9	Criar <i>post</i> para a criação de um grupo	.	.	18
7.10	Seguir jogo - Diagrama de tarefas	.	.	19
7.11	Seguir jogo	.	.	19
7.12	Enviar mensagem	.	.	20
10.1	Diagrama do contexto de negócio	.	.	33
10.2	Diagrama de contexto técnico	.	.	35
11.1	Diagrama de componentes (nível 1)	.	.	36
11.2	Diagrama de classe - PIM	.	.	37
11.3	Diagrama de classe - PSM	.	.	39
12.1	Diagrama de Sequência - Enviar mensagem amigo	.	.	41
12.2	Diagrama de Sequência - Enviar mensagem num grupo	.	.	42
12.3	Diagrama de Sequência - Aceder ao <i>feed</i>	.	.	42
12.4	Diagrama de Sequência - Juntar a grupo	.	.	43
12.5	Diagrama de Sequência - Criar <i>post</i> grupo	.	.	44
12.6	Diagrama de Sequência - Criar <i>post</i> multimédia	.	.	44
13.1	Diagrama lógico	.	.	45
13.2	Dicionário de dados	.	.	46
14.1	Diagrama de <i>deployment</i>	.	.	48
14.2	<i>Dockerfile</i> criado para a implementação da componente <i>frontend</i>	.	.	49
14.3	<i>Dockerfile</i> criado para a implementação da componente <i>backend</i>	.	.	50
15.1	Workload no Apache JMeter	.	.	52
15.2	Percentagem de erro dos pedidos com 100 Clientes	.	.	52
15.3	Tempo de resposta médio com 100 Clientes	.	.	53
15.4	Throughput com 100 Clientes	.	.	53
15.5	Percentagem de erro dos pedidos com 200 Clientes	.	.	54
15.6	Tempo de resposta médio com 200 Clientes	.	.	54
15.7	Throughput com 200 Clientes	.	.	54
15.8	Percentagem de erro dos pedidos com 500 Clientes	.	.	55
15.9	Tempo de resposta médio com 500 Clientes	.	.	55
15.10	Throughput com 500 Clientes	.	.	55
15.11	Percentagem de erro dos pedidos com 1000 Clientes	.	.	56
15.12	Tempo de resposta médio com 1000 Clientes	.	.	56
15.13	Throughput com 1000 Clientes	.	.	56
15.14	Percentagem de erro dos pedidos com 1500 Clientes	.	.	57
15.15	Tempo de resposta médio com 1500 Clientes	.	.	57
15.16	Throughput com 1500 Clientes	.	.	57
16.1	Página de registo	.	.	58
16.2	Página de registo - preenchida	.	.	59
16.3	Página de registo - preenchida erradamente	.	.	59

16.4 Página de <i>login</i>	60
16.5 Página de <i>login</i> - preenchida	60
16.6 Página de <i>login</i> - preenchida incorretamente	61
16.7 Página <i>home</i>	61
16.8 Página <i>home</i> - carregar em ver perfil	62
16.9 Página do perfil	62
16.10 Página do perfil - adicionar imagem de perfil	63
16.11 Página do perfil - dar reset à imagem nova	63
16.12 Página do perfil - alterar nome	64
16.13 Página do perfil - alterar nome com sucesso	64
16.14 Página do perfil - alterar nome sem sucesso	65
16.15 Página do adicionar jogos	65
16.16 Página do adicionar jogos - jogos selecionados	66
16.17 Página inicial - com jogos adicionados	66
16.18 Página inicial - deixar de seguir jogo	67
16.19 Página inicial - deixar de seguir jogo	67
16.20 <i>Feed</i> do jogo - deixar de seguir jogo 2	68
16.21 Página inicial - deixar de seguir jogo 2	68
16.22 Página inicial - aceder ao <i>feed</i> de um jogo	69
16.23 Página <i>feed</i> do Among Us	69
16.24 Página inicial - criar publicação de grupo	70
16.25 Página criar publicações - criar grupo	70
16.26 Página criar publicações - introdução dos dados incorretamente	71
16.27 Página criar publicações - introdução dos dados de forma correta	71
16.28 Página inicial - procurar grupo	72
16.29 Página procura de grupo	72
16.30 Página procura de grupo - com filtros	73
16.31 Página procura de grupo - confirmação	73
16.32 Página procura de grupo - juntou-se a novo grupo	74
16.33 Sair de um grupo - confirmação	74
16.34 Sair de um grupo - conclusão	75
16.35 Pagina inicial - conversa de grupo	75
16.36 Pagina inicial - conversa de grupo aberta	76
16.37 Pagina inicial - escrever mensagem para o grupo	76
16.38 Pagina inicial - mensagem para o grupo enviada	77
16.39 Pagina inicial - adicionar amigo	77
16.40 Menu adição de um amigo	78
16.41 Menu adição de um amigo - Amigo introduzido	78
16.42 Menu adição de um amigo - Amigo introduzido	79
16.43 Menu adição de um amigo - Erro ao introduzir	79
16.44 Menu inicial - Adicionar amigo, forma 2	80
16.45 Página perfil - Adicionar amigo	80
16.46 Página perfil - Amigo adicionado	81
16.47 Página inicial - Abrir conversa com amigo	81
16.48 Página inicial - Ver perfil a partir do <i>chat</i>	82
16.49 Página perfil - Remover amigo	82
16.50 Página perfil - Remover amigo	83
16.51 Página perfil - Classificar utilizador	83
16.52 Página perfil - Escolher classificação	84
16.53 Página perfil - Classificação escolhida	84
16.54 Página perfil - Classificação realizada	85

16.55	Página Inicial - receber notificação	85
16.56	Página perfil - Enviar mensagem privada	86
16.57	Página perfil - Mensagem privada escrita	86
16.58	Página perfil - Mensagem privada enviada	87
16.59	Página perfil - Criar publicação multimédia	87
16.60	Página criar publicação - Imagem/vídeo	88
16.61	Página criar publicação - Selecionar media	88
16.62	Página criar publicação - Campos preenchidos	89
16.63	Página <i>feed</i> jogo - publicação realizada	89
16.64	Página criar publicação - Campos incorretos	90
16.65	Página inicial - Ver próprio perfil	90
16.66	Página inicial - Ver próprio perfil	91
16.67	Página do perfil	91
16.68	Página do perfil - secção meus <i>posts</i>	92
16.69	Página do perfil - botão do caixote selecionado	92
16.70	Página do perfil - confirmar eliminação	93
16.71	Página do perfil - publicação eliminada	93
16.72	Página inicial - publicação sem avaliação	94
16.73	Página inicial - publicação com <i>like</i>	94
16.74	Página inicial - publicação com <i>dislike</i>	95
16.75	Página inicial - imagem da publicação	95
16.76	Página inicial - carregar no avatar para terminar sessão	96
16.77	Página inicial - carregar em <i>logout</i>	96
16.78	Página <i>Login</i> - sessão terminada	97
16.79	Página admin publicações multimédia	97
16.80	Página admin publicações multimédia - filtros	98
16.81	Página admin publicações multimédia - multimédia	98
16.82	Página admin publicações multimédia - confirmação	98
16.83	Página admin publicações multimédia - publicação removida	99
16.84	Página admin publicações grupo	99
16.85	Página admin publicações grupo - filtrada	99
16.86	Página admin publicações grupo - confirmação	100
16.87	Página admin publicações grupo - publicação eliminada	100
16.88	Página admin banir utilizador	100
16.89	Página admin banir utilizador - filtro	101
16.90	Página admin banir utilizador - confirmação	101
16.91	Página admin banir utilizador - remoção	101
17.1	Formulário Perito 1 - Registar	103
17.2	Formulário Perito 1 - Criar <i>post</i> de grupo	103
17.3	Formulário Perito 1 - Criar <i>post</i> /multimédia	104
17.4	Formulário Perito 1 - Seguir novo jogo	104
17.5	Formulário Perito 1 - Juntar a Equipa	105
17.6	Formulário Perito 2 - Criar <i>post</i> de grupo	106
17.7	Formulário Perito 2 - Criar <i>post</i> multimédia	107
17.8	Formulário Perito 2 - Seguir novo jogo	107
17.9	Formulário Perito 2 - Juntar a Equipa	108
17.10	Formulário Perito 3 - Criar <i>post</i> de grupo	109
17.11	Formulário Perito 3 - Criar <i>post</i> /multimédia	110
17.12	Formulário Perito 3 - Seguir novo jogo	110
17.13	Formulário Perito 3 - Juntar a Equipa	111
17.14	Valores obtidos nos passos intermédios	113

20.1	Formulário - Registar	117
20.2	Formulário Utilizador 2 - Registar	117
20.3	Formulário Utilizador 3 - Registar	118
20.4	Formulário - Criar <i>post</i> de grupo	119
20.5	Formulário - Criar <i>post</i> multimédia	120
20.6	Formulário - Seguir novo jogo	120
20.7	Formulário - Juntar a equipa	121

Índice de tabelas

3.1	Aplicações Parceiras	6
3.2	Tabela salarial dos trabalhadores envolvidos	6
7.1	Especificação do <i>use case</i> : Registar	13
7.2	Especificação do <i>use case</i> : <i>Login</i>	14
7.3	Especificação do <i>use case</i> : Aceder a <i>feed</i>	15
7.4	Especificação do <i>use case</i> : Juntar a grupo	16
7.5	Especificação do <i>use case</i> : Criar <i>post</i>	18
7.6	Especificação do <i>use case</i> : Seguir jogo	20
7.7	Especificação do <i>use case</i> : Enviar mensagem	21
17.1	Valores médios obtidos, por questão, no questionário SUS	113

1 Introdução

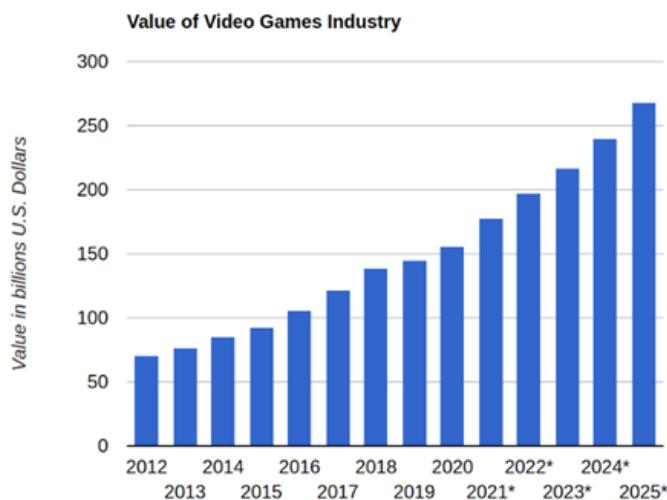
1.1 Propósito do sistema

1.1.1 Motivação e Contextualização

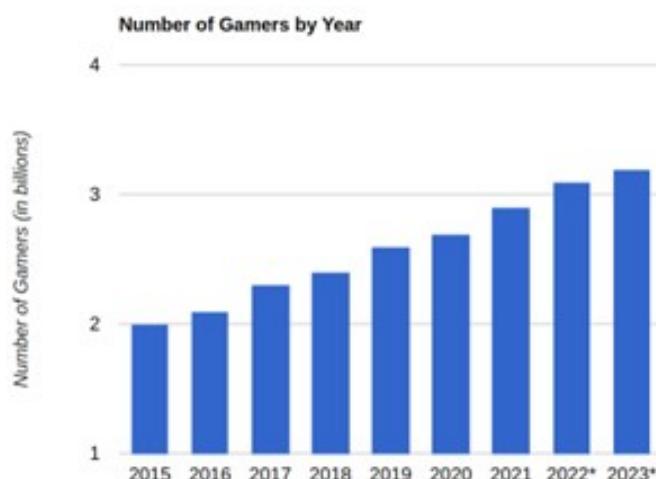
Em 2021, o mercado de gaming foi avaliado em 198.40 mil milhões de dólares, e espera-se que, por 2027, atinja o valor de 339.95 biliões de dólares, registando assim uma taxa de crescimento anual composta de 8.94% (2022-2027) (figura 1.1a).

Devido às restrições de confinamento implementadas internacionalmente devido à pandemia do COVID-19, um grande número de pessoas investiu em plataformas *gaming* para entretenimento, o que gerou um enorme fluxo de utilizadores neste mercado (figura 1.1b).

Para além de entretenimento, os videojogos são, muitas vezes, uma forma de socialização uma vez que cada vez mais a indústria *gaming* oferece experiências online interativas com vários jogadores. Assim, a plataforma GameHub procura fornecer a todos os jogadores um local virtual onde podem encontrar e socializar com outros jogadores, de modo a formarem grupos e participarem em atividades em conjunto.



(a) Valor da indústria de videojogos



(b) Número de *gamers* por ano

Figura 1.1

1.1.2 Objetivos do projeto

O projeto GameHub consiste na criação de uma rede social focada nos jogadores de videojogos. A criação desta rede social apresenta como objetivo permitir aos vários jogadores solos conseguirem criar equipas com outros jogadores, comunicando entre si. Para além disto, também permitirá que os jogadores consigam aprofundar a sua paixão pelos jogos, partilhando as suas experiências e visualizando as de outros, bem como se informar das últimas notícias referentes aos jogos.

Assim, com este projeto pretende-se dar uma resposta imediata ao grande crescimento deste mercado e do número de jogadores, sendo uma rede social focada na área da indústria *gaming*.

Ao longo do relatório serão identificadas as funcionalidades e requisitos do sistema, bem como os seus *stakeholders*, que permitirão perceber melhor o objetivo deste projeto.

1.2 Cliente, *customer* e outros *Stakeholders*

1.2.1 Cliente

O cliente do produto é a companhia norte-americana de jogos eletrónicos e softwares, Epic Games. Atualmente esta empresa conta com a presença de dezenas de jogos incluindo alguns dos jogos *multiplayer* mais populares como League of Legends, Rocket League e Fortnite. Além disso, a empresa desenvolveu um motor de jogo, Unreal Engine, que é utilizado para criar videojogos. Desempenha um papel importante na vida da maioria dos jogadores disponibilizando uma oferta diversa de jogos e incluindo uma rotação semanal de jogos grátis.

1.2.2 Customer

O produto será indicado para qualquer pessoa com interesse em jogos *multiplayer*.

1.2.3 Stakeholders

Jogadores

Constituem a entidade principal do projeto. Os principais interessados na concretização da aplicação são aqueles com interesse e/ou experiência em jogos *multiplayer*. Os jogadores encontram no produto um sistema confiável onde podem encontrar companhia para jogar casualmente ou encontrar jogadores do mesmo nível para sessões competitivas. Por sua vez, constituem também a fonte principal da *feedback* para a aplicação.

Empresas de comunicação

Empresas como o Discord ou o TeamSpeak3 poderão beneficiar da concretização do produto na medida em que inevitavelmente, após utilizar a aplicação para encontrar jogadores, os envolvidos sentiriam a necessidade de comunicar verbalmente durante o jogo. Assim sendo, uma integração destes serviços de comunicação no produto iria resultar num aumento do seu número de utilizadores. No entanto estas empresas podem rapidamente passar a *stakeholders* negativos, caso a plataforma comece a desenvolver funcionalidades que já são oferecidas por estas aplicações.

Empresas de jogos *multiplayer*

Desenvolvedores de jogos *multiplayer*, independentemente da plataforma (consola, computador, etc.), terão interesse em colocar o seu jogo na aplicação. Como consequência, poderá dar maior reconhecimento ao seu jogo.

Sponsors

Entidades como eventos, influências, clubes de *eSports*, etc. podem dar maior visibilidade ao sistema em troco de contratos monetários. Da mesma forma, clubes de *eSports* ou influências menos conhecidas podem ganhar visibilidade ao fazer parte do sistema.

Developers e Engenheiros de Software

É esperado que a equipa encarregue do desenvolvimento do produto possua as competências para tal, tais como saber programar orientado a objetos, gerir um projeto e testar o software. O desenvolvimento do produto irá gerar oportunidades de emprego na área de desenvolvimento de software.

2 Utilizadores do sistema

Ao analisar o contexto do projeto e também todos os clientes, *customers* e *stakeholders* foi possível identificar a existência de dois utilizadores do sistema - *gamer* e administrador.

2.1 *Gamer*

O utilizador *gamer*, representa os utilizadores gerais do sistema, que deverão realizar publicações e participar em equipas criadas. Assim, este utilizador é identificado pelas seguintes características:

Nome: *Gamer*

Papel: Criar publicações, grupos, participar em grupos e socializar com os seus amigos inscritos no sistema.

Experiência na área: Este tipo de utilizador pode ser inexperiente ou mestre na área (a experiência não tem influência).

Experiência tecnológica: O utilizador deve ter experiência tecnológica suficiente para ser capaz de utilizar plataformas de videojogos.

2.2 Administrador

Para além do utilizador anterior, identifica-se o utilizador: administrador. Este utilizador tem como papel garantir que os vários utilizadores respeitam as regras impostas. Assim, este utilizador é caracterizado por:

Nome: Administrador.

Papel: Gerir o conteúdo presente no sistema. Este utilizador é responsável por banir *gamer* irresponsáveis ou eliminar publicações que infrinjam as regras.

Experiência na área: Este tipo de utilizador pode ser inexperiente ou mestre na área (a experiência não tem influência).

Experiência tecnológica: O utilizador deve ter experiência tecnológica suficiente para ser capaz de utilizar a plataforma de gestão.

3 Restrições do projeto

A seguinte secção descreve as restrições obrigatórias levantadas aquando da fase inicial de planeamento do projeto a ser realizado.

3.1 Restrições da solução

Requisito: R01

Descrição: A linguagem de programação a ser utilizada deve ser Java.

Fundamentação: Java é uma linguagem orientada a objetos com a qual os desenvolvedores do projeto já se encontram muito familiarizados, de modo a permitir diminuir o tempo de desenvolvimento, acelerando assim a entrega do produto.

Origem: Reunião com a empresa responsável pela criação do produto.

Critério de ajuste: Java contém inúmeras bibliotecas que irão facilitar a conexão de todo o projeto de forma eficaz.

Prioridade: *Should.*

Requisito: R02

Descrição: Utilização de uma Base de Dados MySQL.

Fundamentação: A equipa de desenvolvimento já trabalha com este tipo de base de dados e seria de interesse que o GameHub também o fosse para facilitar a manutenção da mesma sem ter de recorrer a terceiros.

Origem: Reunião com a empresa responsável pela criação do produto.

Critério de ajuste: A empresa já se encontra habituada com a mesma para além de ser simples de usar com Java.

Prioridade: *Should.*

3.2 Ambiente de Implementação do Sistema Atual

O sistema a ser desenvolvido deve ser feito tendo a consideração que apenas as infraestruturas da Epic Games pré-existentes podem ser utilizadas, uma vez que a empresa não pretende despender mais dinheiro nessa área. Desta forma a equipa responsável por este documento tem a função de instalar a aplicação nos servidores da Epic Games.

3.3 Aplicações parceiras

De modo a organizar melhor o projeto entre toda a equipa, foram tomadas algumas decisões quanto às aplicações de desenvolvimento desejadas de modo a poderem-se evitar problemas de compatibilidade entre diferentes ferramentas. Deste modo foram escolhidas as aplicações presentes na tabela em seguida:

Aplicação	Objetivo
	Para apoiar a elaboração do projeto, irá ser utilizada a ferramenta IntelliJ como ambiente de desenvolvimento do mesmo.
	Toda a base de dados irá ser criada e mantida utilizando MySQL.

Tabela 3.1: Aplicações Parceiras

3.4 Ambiente da utilização do sistema

O sistema irá apresentar dois tipos de utilizadores, os jogadores e os administradores. Desta forma, em seguida apresentam-se os modos de utilização ligados a cada tipo de utilizador.

- **Gestão (administrador)** - durante a gestão, um administrador pode tomar decisões sobre o comportamento dos *gamers*. Assim, este pode banir *gamers* e apagar posts realizados.
- **Procura de grupo (gamer)** - um jogador pode criar uma publicação para procurar novos companheiros de equipa. Para além disto, um gamer pode pesquisar por grupos já existentes.
- **Criar publicação** - Um *gamer* pode criar uma publicação multimédia, texto ou de criação de grupo.
- **Browsing (gamer)** - durante o *browsing*, um jogador pode ver posts de outros jogadores assim como avaliá-los.

3.5 Restrições de calendário

É do interesse da Epic Games que a aplicação esteja pelo menos operacional com as suas funcionalidades principais até dia 17 de junho de 2022. No entanto só no fim de julho do mesmo ano é que se espera que todas as funcionalidades inicialmente previstas estejam totalmente refinadas.

3.6 Restrições de orçamento

Durante todo o processo de desenvolvimento do sistema, estima-se que os custos associados à equipa de criação do mesmo serão os seguintes:

Tarefa	Salário Mensal/Pessoa	Nº Pessoas	Nº Meses	Total
Engenheiro de Requisitos	2000€	2	2	8000 €
Arquiteto de Software	2500€	3	4	30000€
Testador	2000€	2	3	12000€
Developer	2500€	5	4	50000€
Designer	1500€	2	3	9000€
Total				109000€

Tabela 3.2: Tabela salarial dos trabalhadores envolvidos

4 Taxonomia e definições

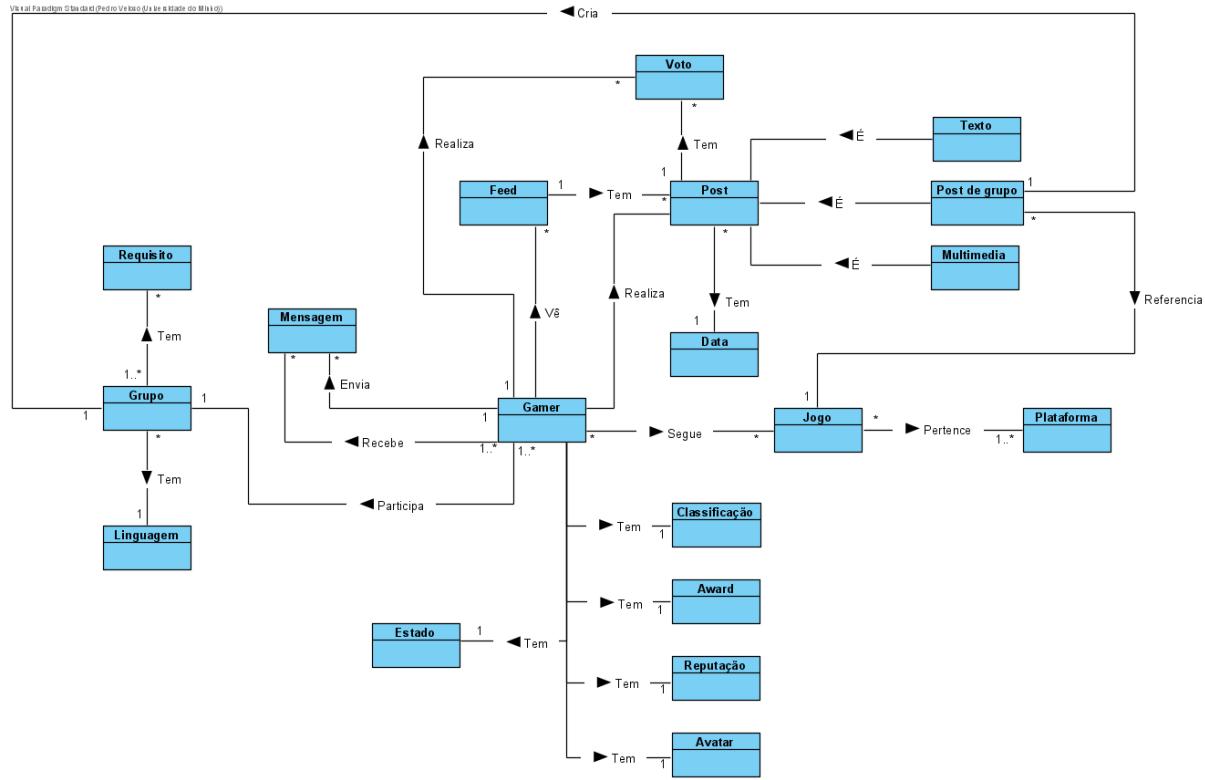


Figura 4.1: Modelo de domínio

Os conceitos utilizados no modelo de domínio acima representado estão corretamente descritos no seguinte glossário:

- **Gamer:** Entidade responsável pela utilização do sistema. Por jogador referencia-se todos os indivíduos que manifestam interesse pela área do projeto.
- **Classificação:** Valor atribuído aos jogadores de modo a classificar a sua habilidade.
- **Award:** Prémio virtual oferecido aos jogadores por determinados feitos.
- **Reputação:** Representa a atitude e comportamento do jogador. Este valor é modificado de acordo com as avaliações de outros jogadores que interagiram com o primeiro.
- **Avatar:** Imagem representativa de cada jogador. Esta imagem pode ser personalizada por cada jogador.
- **Jogo:** Representa um videojogo presente no sistema. Pode ter várias plataformas.
- **Plataforma:** Hardware de videojogos em que um jogo existe.
- **Data:** Data em que uma publicação foi postada por um dado utilizador.
- **Post:** Publicação postada por um utilizador que pode conter vários tipos de informação e intenções.
- **Voto:** Classificação que um utilizador dá à publicação em questão (*like* ou *dislike*).
- **Texto:** Tipo de publicação que contém unicamente caracteres textuais.

- **Post de grupo:** Tipo de publicação postada por um jogador que manifesta a intenção de procurar por outros jogadores.
- **Multimédia:** Tipo de publicação que contém vídeo ou imagens.
- **Mensagem:** Representa troca de informação entre dois ou mais usuários (no caso de um *chat* de grupo).
- **Grupo:** Equipa formada por um conjunto de jogadores.
- **Requisito:** Entidade que engloba restrições do jogador para a entrada de elementos num grupo.
- **Linguagem:** Esta entidade representa a linguagem de preferência do jogador que realizou a criação de um grupo.
- **Feed:** Conjunto de posts realizados para um determinado jogo.

5 Factos e assunções

Numa fase inicial do desenvolvimento do projeto, para evitar futuros conflitos de ideias, é importante determinar o conjunto de factos que se consideram pré-estabelecidos.

5.1 Decisões Legais e Políticas

Uma vez que se trata de uma plataforma de jogos, é importante delimitar uma série de informações legais e políticas, de forma a evitar futuros problemas jurídicos.

1. Os utilizadores apenas poderão aceder a jogos cujo ESRB (Entertainment Software Risk Board – sistema de classificação que define a idade mínima associada a um determinado jogo) seja igual ou inferior à sua idade.

5.2 Ambiente de Desenvolvimento

Um dos aspetos mais importantes no desenvolvimento do sistema será definir a maneira e as condições nas quais este vai ser desenvolvido, para que assim não haja incoerências durante o desenvolvimento do projeto.

1. Todos os elementos do grupo terão de trabalhar na versão mais recente do projeto, para evitar perdas de informação.
2. Será necessário dividir tarefas entre cada elemento, de modo que a carga de trabalho seja uniformemente distribuída.
3. Todos os elementos do grupo terão de estar de acordo relativamente às linguagens de programação escolhidas para o desenvolvimento do sistema.

5.3 Dependências Externas

A plataforma apresentará um conjunto de jogos que poderão ser adicionados pelo utilizador. Assim, é importante definir as dependências externas do sistema, isto é, quais os fatores que terão de estar presentes de maneira a ser possível aceder a esses mesmos jogos.

1. Todos os jogos disponíveis para adição terão de existir, isto é, não poderá haver jogos no sistema que ainda não tenham sido disponibilizados ao público
2. A lista de jogos disponíveis deverá ser regularmente atualizada, para que o utilizador consiga ter acesso aos jogos mais recentes.

5.4 Componentes Disponíveis

Apesar de haver bastantes funcionalidades que se pretendem implementar, é necessário definir quais destas serão fundamentais para garantir o funcionamento correto do sistema.

- Seguir Jogos
- Formar Grupos
- Visualizar Publicações
- Avaliar Outros Jogadores
- Criar publicações

6 Âmbito do projeto

6.1 Situação atual

Embora já existam alguns serviços semelhantes ao proposto, o número de jogadores solo que se associam de forma aleatória ainda é bastante representativo. Assim, com o desenvolvimento deste novo serviço, vai ser possível não só abranger um público maior, como também tornar a experiência mais fácil para os jogadores atuais.

6.2 Contexto do trabalho

Como foi referido acima, o GameHub pretende ser uma rede social focada no mercado *gaming*. Ora, para este produto conseguir atingir este objetivo, é necessário compreender e conhecer perfeitamente o funcionamento e contexto deste mercado/indústria.

- **Estudo de jogos *multiplayer*:** De forma a compreender de forma completa a área dos jogos *multiplayer* é necessário entender que conceitos envolvem, nomeadamente número de elementos de cada grupo, plataformas, *ranks* e modos de jogo. Para atingir este objetivo é importante interagir com alguns jogadores versados para entender como procedem para criar um grupo num determinado grupo e questionando como gostariam que o GameHub funcionasse.
- **Estudo de plataformas semelhantes:** Plataformas como o *Reddit* onde existem comunidades (entre outras) de vários jogos serão estudadas para perceber o método de criação de *posts* em estruturas já existentes e populares.
- **Estudo de aplicações conversacionais:** Todo o tipo de plataformas que ofereçam interação entre diferentes utilizadores.

7 Âmbito do sistema

Um use case é uma lista de ações ou passos cujo objetivo é definir as interações entre um ator e um sistema de modo a atingir um determinado objetivo. De seguida, apresentam-se todas as interações entre o sistema e os atores existentes.

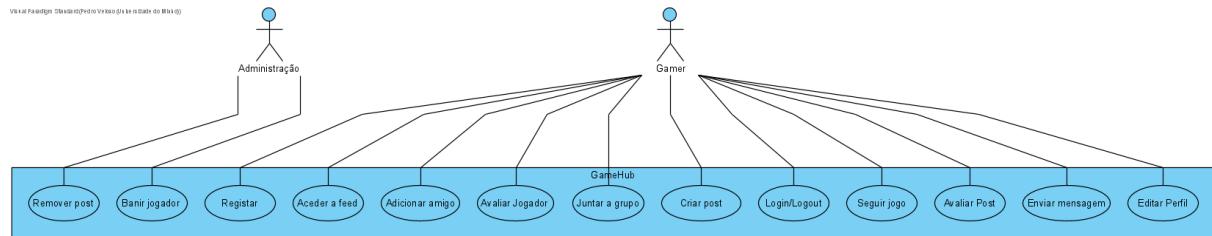


Figura 7.1: Modelo de Use Cases

Ao observar o diagrama de use case nota-se a existência de dois intervenientes no sistema, identificados como atores – *Gamer* e Administração. Um *Gamer* pode:

- **Realizar o seu registo no sistema.** Desta iteração resulta a criação de uma nova conta associada a um novo *gamer*;
- **Realizar o login no sistema.** Para realizar operações no sistema, um *gamer* deverá ter a sessão iniciada;
- **Realizar o logout do sistema.** Após realizar todas as operações, um *gamer* deverá realizar o logout, desvinculando assim a plataforma da sua conta;
- **Aceder a um feed.** Durante a utilização do sistema deverá ser possível, aos *gamers*, visualizar os posts realizados, que se encontram associados a um determinado jogo.
- **Adicionar amigos.** Ao utilizar o sistema os *gamers* deverão ser capazes de adicionar outros *gamers* à sua lista de amigos.
- **Avaliar jogadores.** Ao dissolver uma equipa deverá ser possível a cada participante classificar os restantes, consoante as habilidades demonstradas no jogo.
- **Juntar-se a grupos.** A principal característica do sistema é a capacidade de cada *gamer* formar uma equipa para um determinado jogo.
- **Criar posts.** Deverá ser possível aos *gamers* partilhar as suas experiências – através de vídeo, imagens ou texto – com outros jogadores. Para além disto, também deverá ser possível partilhar com outros *gamers* a intenção de formar uma equipa, para um determinado jogo.
- **Seguir jogos.** Cada *gamer* deverá seguir os jogos sobre os quais pretende visualizar as experiências de outros jogadores.
- **Avaliar publicações.** Para cada publicações realizada deverá ser possível a cada *gamer* expressar a sua opinião através de um sistema de gostos ou desgostos.
- **Enviar mensagens.** De modo a comunicar com a sua equipa ou amigos presentes no sistema cada *gamer* será capaz de lhes enviar mensagens.
- **Editar perfil.** Cada *gamer* terá associado um avatar e um *username*, que poderá alterar durante a sua participação no sistema.

Após realizar o registo e o respetivo *login* no sistema, o *gamer* pode aceder às suas funcionalidades descritas acima através de um menu presente na interface. Ao contrário do *gamer*, a administração tem o papel de gerir o sistema, podendo assim realizar as seguintes operações:

- **Remover publicações.** Sempre que uma publicações seja considerada inadequada deverá ser possível a sua remoção.
- **Banir jogadores.** Quando um *gamer* demonstrar um comportamento inadequado poderá ter a sua conta eliminada do sistema.

Por fim, descreve-se nas secções seguintes alguns dos use cases apresentados acima com um maior detalhe. Para além disto realizou-se um protótipo da interface, de modo a perceber melhor o fluxo do sistema. Este protótipo encontra-se disponível no seguinte link: https://drive.google.com/file/d/1247EXQj_JwUHDKrYVCh3Yp75_Eqspk14/view?usp=sharing

7.1 Registar

Uma das funcionalidades do sistema disponíveis aos *gamers* é a criação de uma conta pessoal, que lhes permite realizar as suas apostas. Este processo é expresso na seguinte *mockup*:

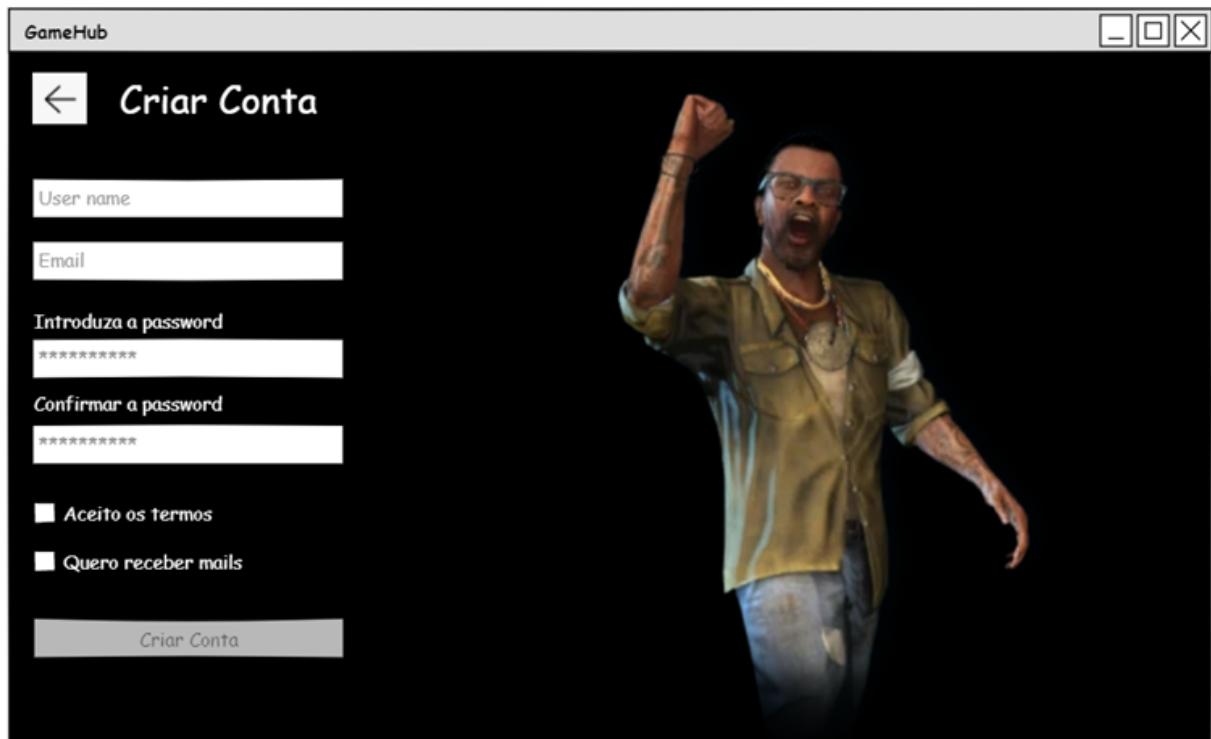


Figura 7.2: Registar

Ao observar a mockup, é possível perceber que um *gamer* deve introduzir os seus dados pessoais – *username*, password e um email – de modo a criar uma nova conta no sistema. Após a introdução dos dados, o sistema deve registá-los automaticamente de forma persistente, permitindo, posteriormente, ao *gamer* realizar o *login*. Contudo, antes de registrar os dados introduzidos, o sistema deve verificar a sua validade, de modo a garantir que o *gamer*:

- Não possui nenhuma conta no sistema, i. e., não existir nenhuma conta com os mesmos dados;
- Introduziu um email válido.

No caso de alguma das validações falhe, a criação da conta também deve falhar, sendo possível corrigir os campos inválidos. Este fluxo pode ser evidenciado pela seguinte especificação do *use case* relativo a esta funcionalidade:

Atores envolvidos - Utilizador
Pré-condição: True
Pós-condição: A conta é registada no sistema
Descrição: Utilizador regista uma nova conta no sistema
Fluxo normal:
1 – Utilizador acede à página de registo 2 – Utilizador insere os seus dados 3 – Utilizador clica em criar conta 4 – Sistema valida os dados inseridos 5 – A conta é criada com sucesso
Fluxo alternativo 1 [Os dados inseridos são inválidos] (passo 4) 4.1 - Sistema indica que dados são inválidos e retorna para o passo 2

Tabela 7.1: Especificação do *use case*: Registar

7.2 Login/Logout

Um *gamer* registado na aplicação GameHub, deve realizar o *login* no sistema sempre que pretender aceder ao mesmo. Desta forma qualquer procedimento realizado posteriormente é devidamente identificado pela conta do *gamer*. Assim, para realizar o *login*, um *gamer* deve introduzir o seu email/username e a password definidas durante o período de registo no sistema, tal como se observa na seguinte *mockup*.

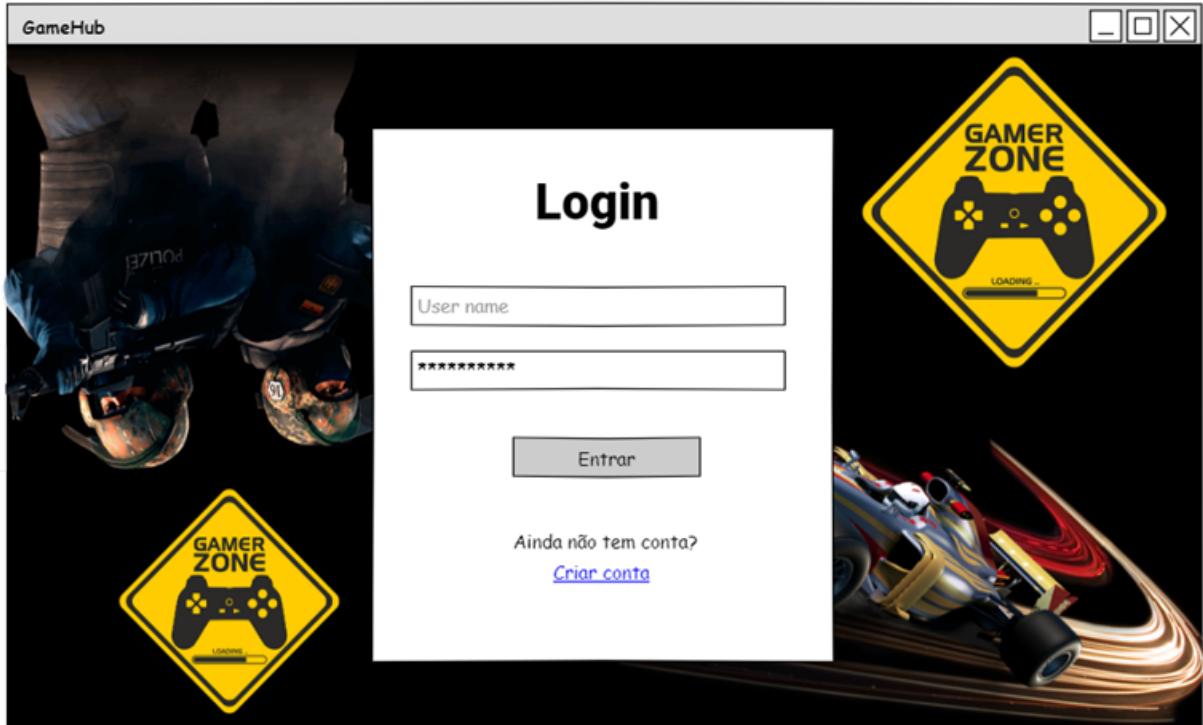


Figura 7.3: Login/Logout

Durante a realização desta funcionalidade, um *gamer* poderá exercer perante o sistema o comportamento especificado de seguida:

Atores envolvidos - Utilizador
Pré-condição: True
Pós-condição: O utilizador fica com a sessão iniciada
Descrição: Utilizador autentica-se no sistema
Fluxo normal:
1 – Utilizador acede à página de <i>login</i>
2 – Utilizador insere os seus dados
3 – Utilizador clica em entrar
4 – Sistema valida os dados inseridos
5 – O utilizador é autenticado e acede ao sistema
Fluxo alternativo 1 [Os dados inseridos são inválidos] (passo 4)
4.1 - Sistema indica que os dados são inválidos e retorna para o passo 2

Tabela 7.2: Especificação do *use case*: *Login*

Após ter realizado o *login*, um apostador pode sair da aplicação quando pretender, realizando para esse efeito o devido *logout*.

7.3 Aceder a *feed*

Ao utilizar o sistema, cada *gamer* poderá visualizar o *feed* de um jogo. Nesta área será possível visualizar *posts* realizados por outros *gamers*. Assim, ao utilizar esta área cada *gamer* encontrará informações relacionadas ao jogo escolhido.

Uma vez que cada *gamer* poderá estar interessado em mais do que um jogo existe uma área na interface onde estão identificados todos os *feeds* a que ele tem acesso. Desta forma, será possível trocar rapidamente e de forma intuitiva entre *feeds*.

A *mockup* associada a esta funcionalidade é apresentada de seguida:

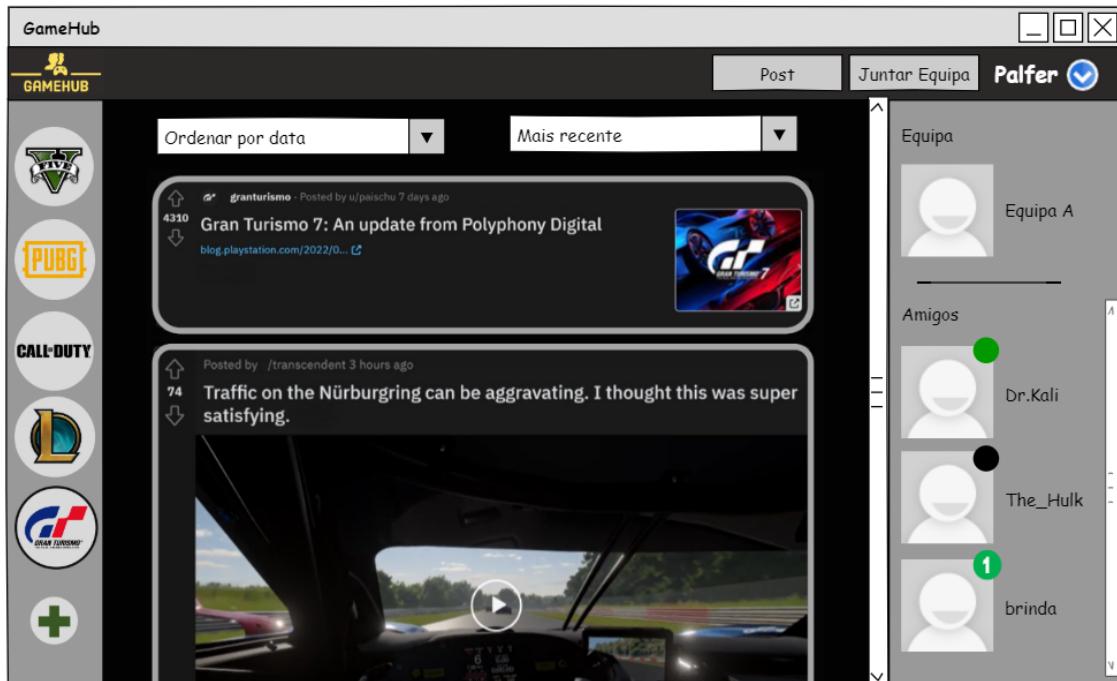


Figura 7.4: Aceder ao *feed*

Ao observar a *mockup*, encontra-se duas áreas em destaque. Na área mais à esquerda está presente a lista com os *feeds* disponíveis, bem como a opção para adicionar mais *feeds* à lista. Na área central

é possível encontrar a secção do *feed*. Nesta área central deverá ser possível encontrar todos os *posts* realizados em relação ao jogo escolhido. De modo a tornar a navegação nesta área mais intuitiva para o gamer, serão apresentados alguns filtros capazes de definir a ordem de apresentação dos *posts*, sendo introduzidos por ordem cronológica descendente como *default*.

O *use case* referente a aceder a um determinado *feed* é especificado na seguinte tabela:

Atores envolvidos - Utilizador
Pré-condição: Utilizador está autenticado no sistema
Pós-condição: O utilizador tem acesso ao <i>feed</i> do jogo selecionado
Descrição: Utilizador acede ao <i>feed</i> de um determinado jogo
Fluxo normal:
1 – Utilizador clica no botão do jogo pretendido
2 – O utilizador modifica a ordem que aparecem os <i>posts</i> do <i>feed</i>
3 – Sistema retorna as publicações referentes a esse jogo, pela ordem especificada
4 - O <i>feed</i> do jogo é apresentado ao utilizador pela ordem definida
Fluxo alternativo 1 [O utilizador não altera a ordem dos <i>posts</i>] (passo 2)
2.1 - Sistema retorna as publicações referentes a esse jogo
2.2 – O <i>feed</i> do jogo é apresentado ao utilizador por ordem cronológica

Tabela 7.3: Especificação do *use case*: Aceder a *feed*

7.4 Juntar a grupo

Uma das funcionalidades apresentadas a um *gamer* é a capacidade de formar um grupo com outros *gamers*. Assim, é possível a estes criar *posts* manifestando essa intenção, tal como referido posteriormente neste relatório. Por outro lado, será disponibilizado aos *gamers* a opção de se juntar a um grupo. Para manifestar esta intenção, um *gamer* poderá utilizar a interface disponibilizada para exercer esta funcionalidade.

Assim, um *gamer* deverá selecionar o botão específico, presente na interface geral, de modo a ser redirecionado para a interface referida. Nesta interface cada *gamer* terá acesso à lista de pedidos de formação de equipas (cronologicamente ordenada). Para além disto, será possível aplicar um conjunto de filtros que permitam obter uma lista com grupos do seu interesse.

Estes passos podem ser observados no diagrama de tarefas seguinte:

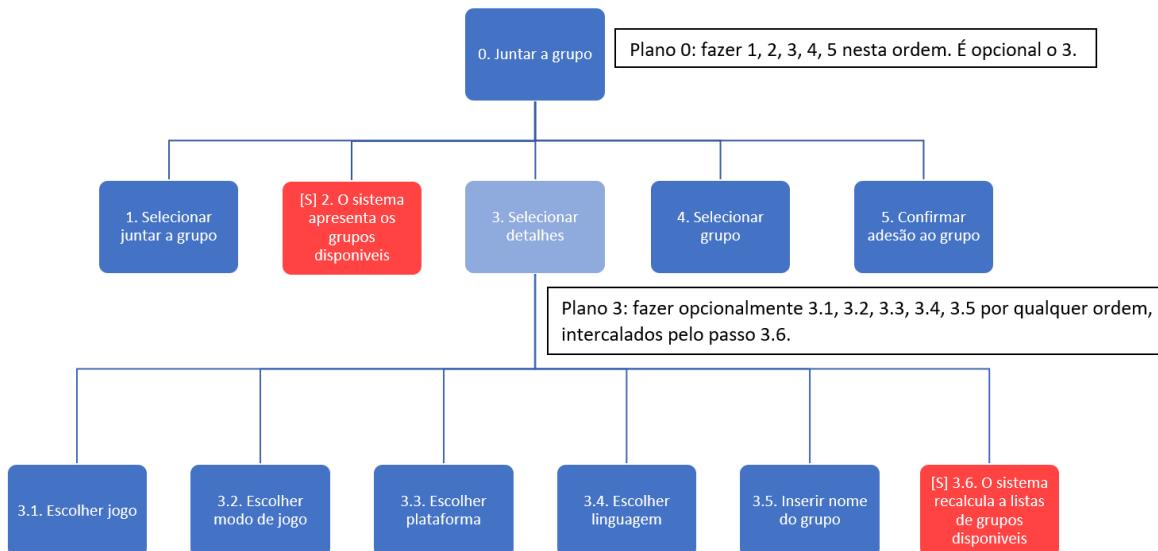


Figura 7.5: Juntar a grupo - Modelo de tarefas

O conjunto de passos referidos no diagrama podem ser representados na seguinte *mockup*:

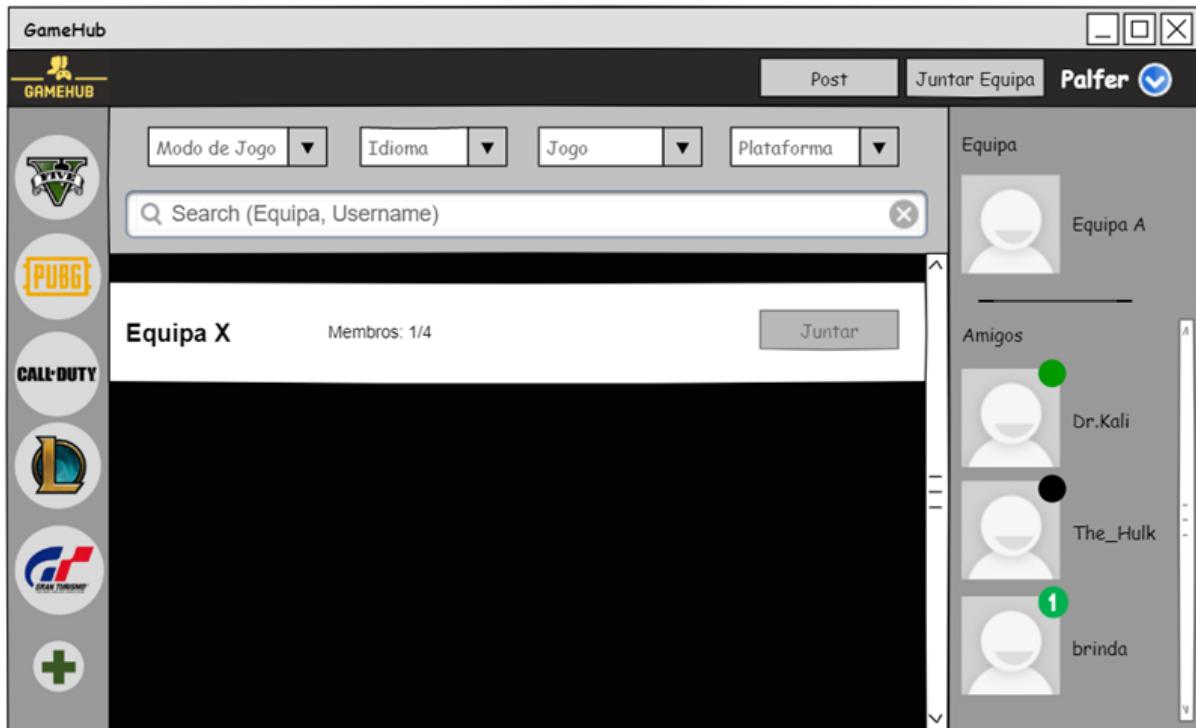


Figura 7.6: Juntar a grupo

Por fim, é possível verificar a especificação do *use case* que originou esta funcionalidade na seguinte tabela:

Atores envolvidos - Utilizador
Pré-condição: O utilizador está autenticado no sistema
Pós-condição: O utilizador faz parte do grupo selecionado
Descrição: Utilizador junta-se a um grupo
Fluxo normal:
1 – Utilizador clica em “Juntar a Grupo” 2 – Utilizador insere os seus parâmetros de pesquisa 3 – Sistema apresenta os grupos que correspondem aos parâmetros definidos 4 – Utilizador clica para se juntar a um dos grupos 5 – Utilizador confirma a adesão 6 - O sistema regista o utilizador como membro do grupo
Fluxo alternativo 1 [O utilizador não recorre aos filtros] (passo 2)
2.1 – O sistema apresenta os todos os grupos por ordem cronológica e retorna para o passo 4
Fluxo de exceção 1 [Não existem grupos para os parâmetros definidos] (passo 3)
3.1 - Sistema indica que não existem grupos que respeitam os parâmetros de pesquisa definidos

Tabela 7.4: Especificação do *use case*: Juntar a grupo

7.5 Criar post

Durante a utilização do sistema cada gamer poderá realizar *posts* em determinados *feeds*. Assim, cada *post* poderá representar uma experiência que pretenda partilhar, tanto por texto, como por multimédia – vídeo ou imagem – como também um pedido para formar um grupo.

Assim, durante a realização de um *post* cada gamer deverá introduzir um conjunto de informações que varia consoante o tipo de *post*. Desta forma, um dos principais passos realizados durante uma publicação será a escolha do tipo de *post*.

Os passos necessários para realizar esta funcionalidade podem ser evidenciados na seguinte diagrama de tarefas:

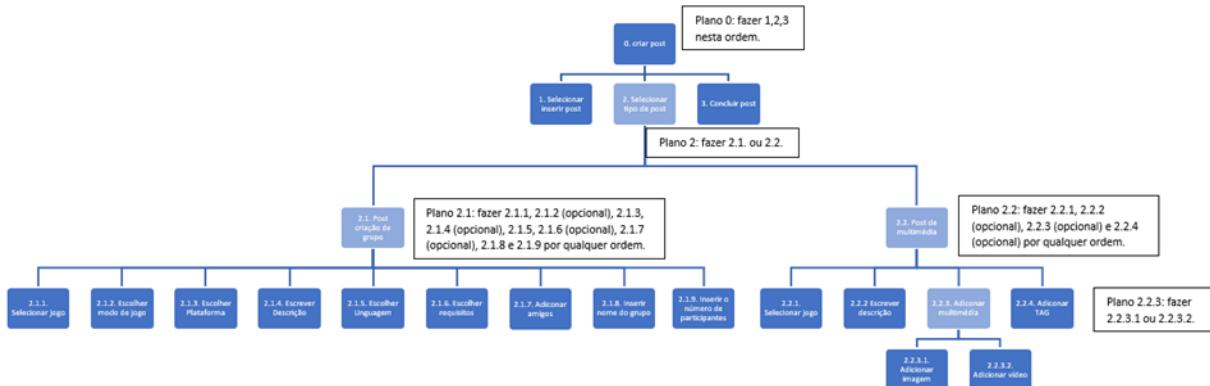


Figura 7.7: Criar *post* - Diagrama de tarefas

Ao observar o esquema acima percebe-se a necessidade de introduzir um conjunto de dados para a realização de um *post* de multimédia – imagem ou vídeo. Porém, o formulário geral pode ser representado na seguinte *mockup*:

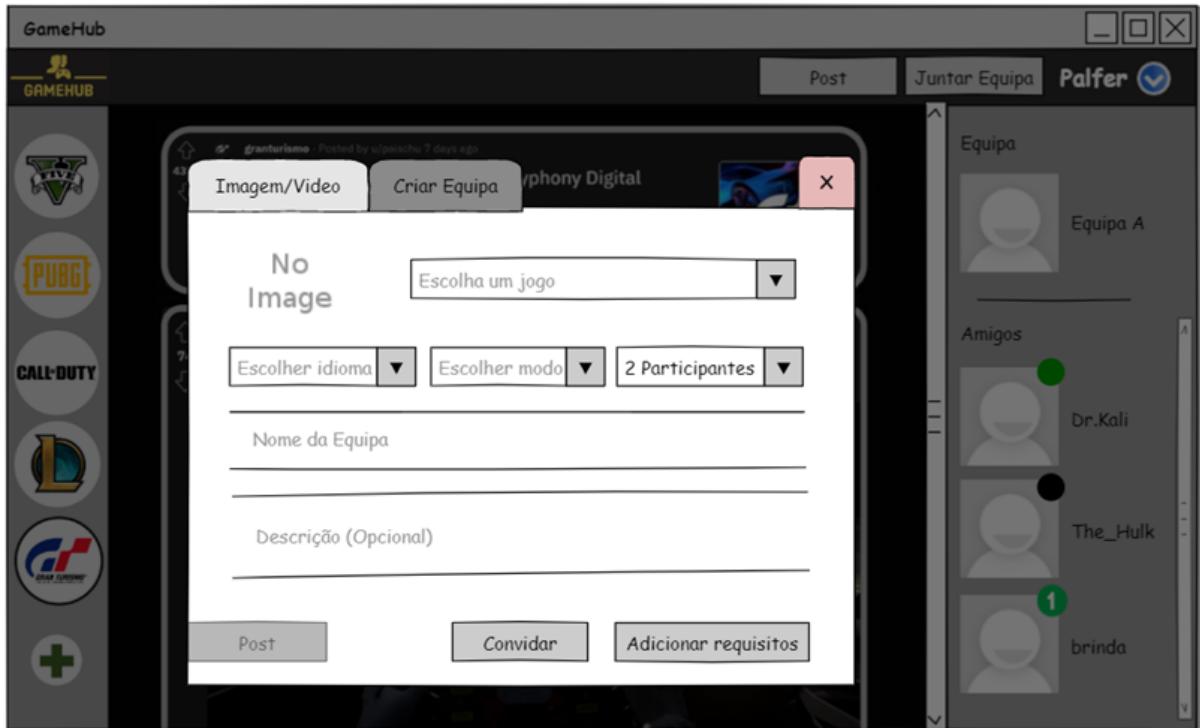


Figura 7.8: Criar *post* para a publicação de multimédia

Por fim, o formulário para a criação de um *post* de criação de equipa é representado na *mockup* presente na figura 7.9.

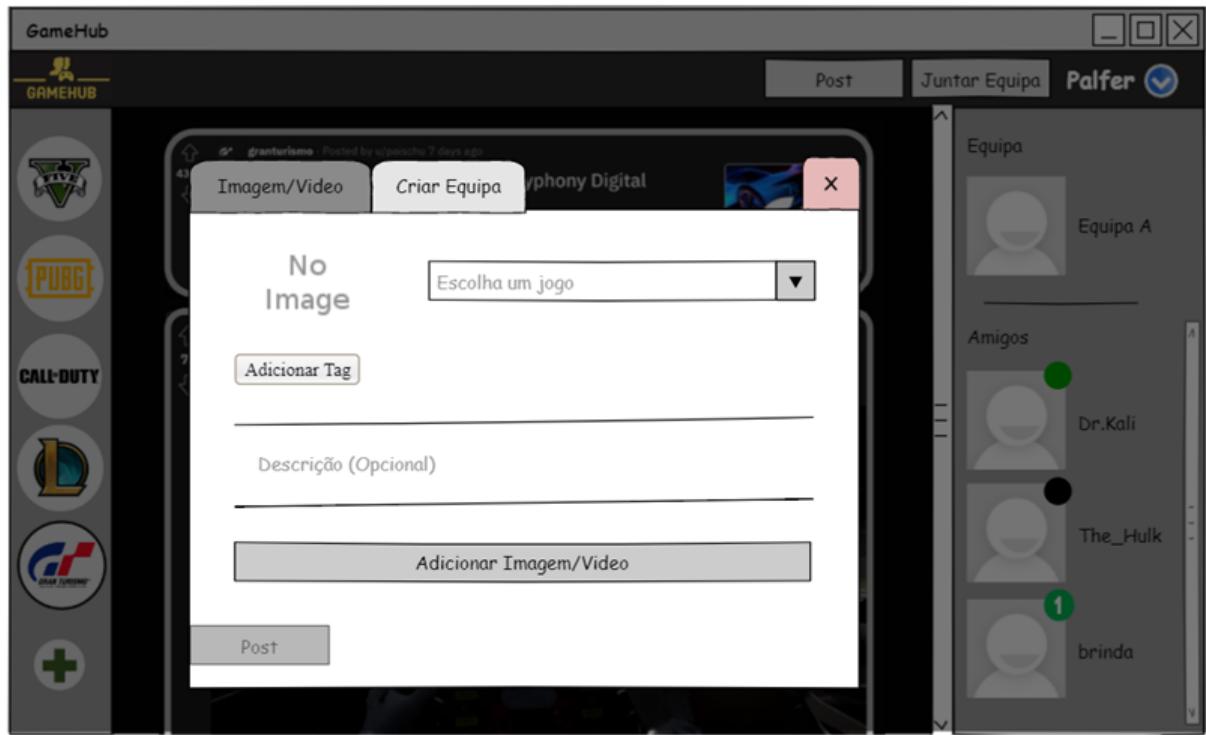


Figura 7.9: Criar *post* para a criação de um grupo

A especificação do *use case* referente à criação de um *post*, e explicado anteriormente, pode ser especificado através da seguinte tabela:

Atores envolvidos - Utilizador
Pré-condição: O utilizador está autenticado no sistema
Pós-condição: A publicação é armazenada no sistema
Descrição: Utilizador cria uma nova publicação
Fluxo normal: 1 – Utilizador clica em “inserir post” 2 – Utilizador escolhe criar um post de multimédia 3 – Utilizador seleciona o jogo 4 – Utilizador escreve a descrição do post 5 – Utilizador adiciona multimédia 6 – Utilizador adiciona TAG 7 – Utilizador conclui a criação do post
Fluxo alternativo 1 [Utilizador não insere multimédia] (passo 5) 5.1 - Continua para o passo 6
Fluxo alternativo 2 [Utilizador não seleciona TAGs] (passo 6) 6.1 - Continua para o passo 7
Fluxo alternativo 3 [Utilizador escolhe criar um post de criação de grupo] (passo 2) 2.1 - Utilizador seleciona jogo 2.2 - Utilizador define os requisitos para esse grupo 2.3 - Retorna ao passo 7

Tabela 7.5: Especificação do *use case*: Criar *post*

7.6 Seguir jogo

Para visualizar os *posts* presentes num *feed* de um determinado jogo, um *gamer* deverá ter a capacidade de adicionar ou remover *feeds* de jogos da sua lista de opções.

Assim, um *gamer* é deverá realizar os passos presentes no diagrama de tarefas 7.10, para aceder à interface que disponibiliza a funcionalidade referida.

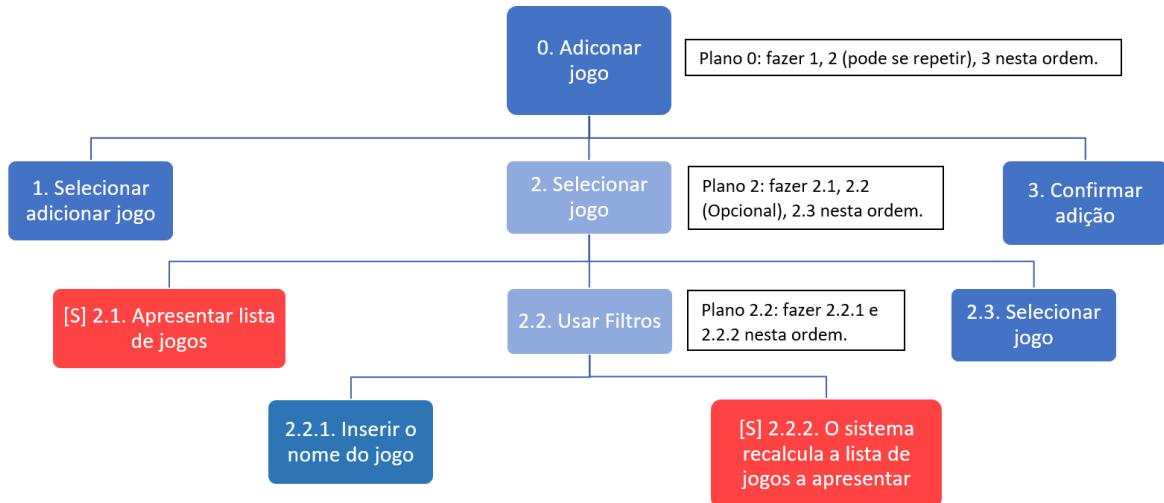


Figura 7.10: Seguir jogo - Diagrama de tarefas

Estes passos podem estar expressos nas *mockups* realizadas. Assim, na figura 11 é possível observar uma *mockup* que exprime um dos conjuntos de passos disponíveis.

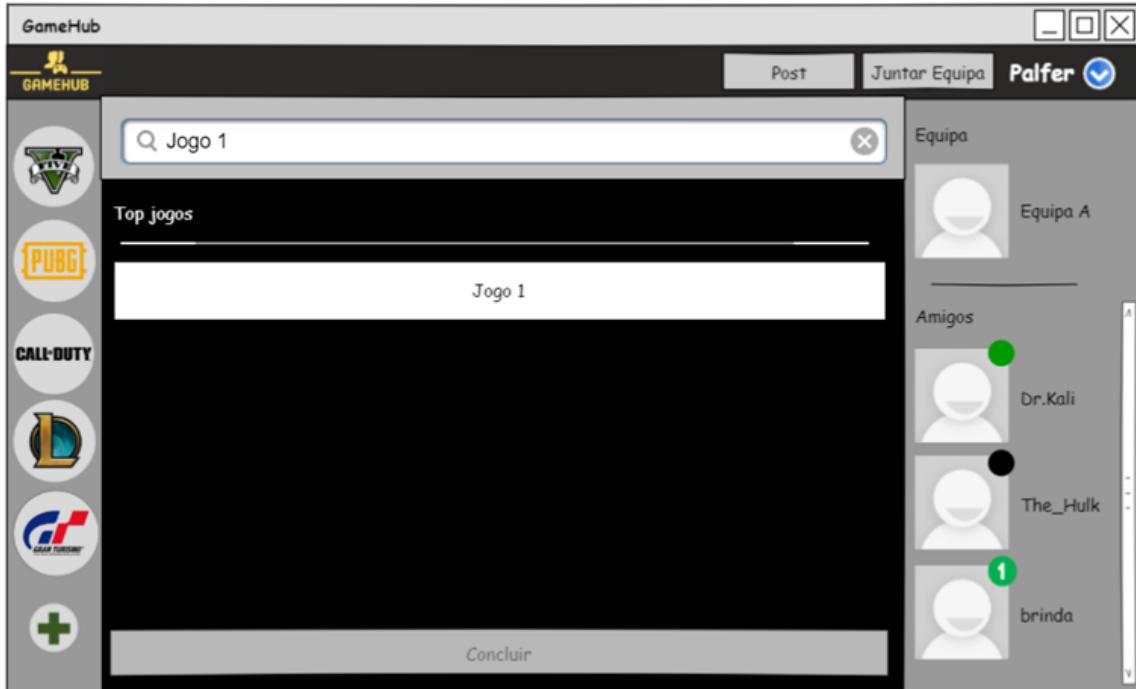


Figura 7.11: Seguir jogo

De modo a abrir esta interface será disponibilizado, na parte final da área dos *feeds*, um botão, devidamente identificado e diferenciado dos restantes, capaz de realizar essa funcionalidade – tal como se observa na *mockup* acima representada.

Por fim, a realização desta funcionalidade implica a realização dos passos presentes na seguinte especificação do *use case* respetivo:

Atores envolvidos - Utilizador
Pré-condição: O utilizador está autenticado no sistema
Pós-condição: O jogo em questão faz parte dos jogos seguidos pelo utilizador
Descrição: Utilizador segue um jogo
Fluxo normal:
1 – Utilizador clica em "adicionar jogo" 2 – Utilizador insere o nome do jogo que quer seguir 3 – Sistema apresenta lista de jogos com esse nome 4 – Utilizador seleciona o jogo 5 – Utilizador segue o jogo
Fluxo de exceção 1 [Não existem jogos com esse nome] (passo 3) 3.1 - Sistema indica que não existem jogos com esse nome
Fluxo alternativo 1 [Os jogos apresentados não correspondem ao jogo pretendido] (passo 3) 3.1 - O utilizador cancela a operação
Fluxo alternativo 2 [O utilizador não pesquisa por um jogo] (passo 2) 2.1 - O sistema apresenta todos os jogos disponíveis e retorna ao passo 4

Tabela 7.6: Especificação do *use case*: Seguir jogo

7.7 Enviar mensagem

Cada *gamer* deverá ser capaz de enviar mensagens de texto para os seus amigos ou para a equipa em que pertence. Assim, tal como se pode visualizar na *mockup* apresentada de seguida, será apresentado o chat com as mensagens sobreposto à interface atual do *gamer*.

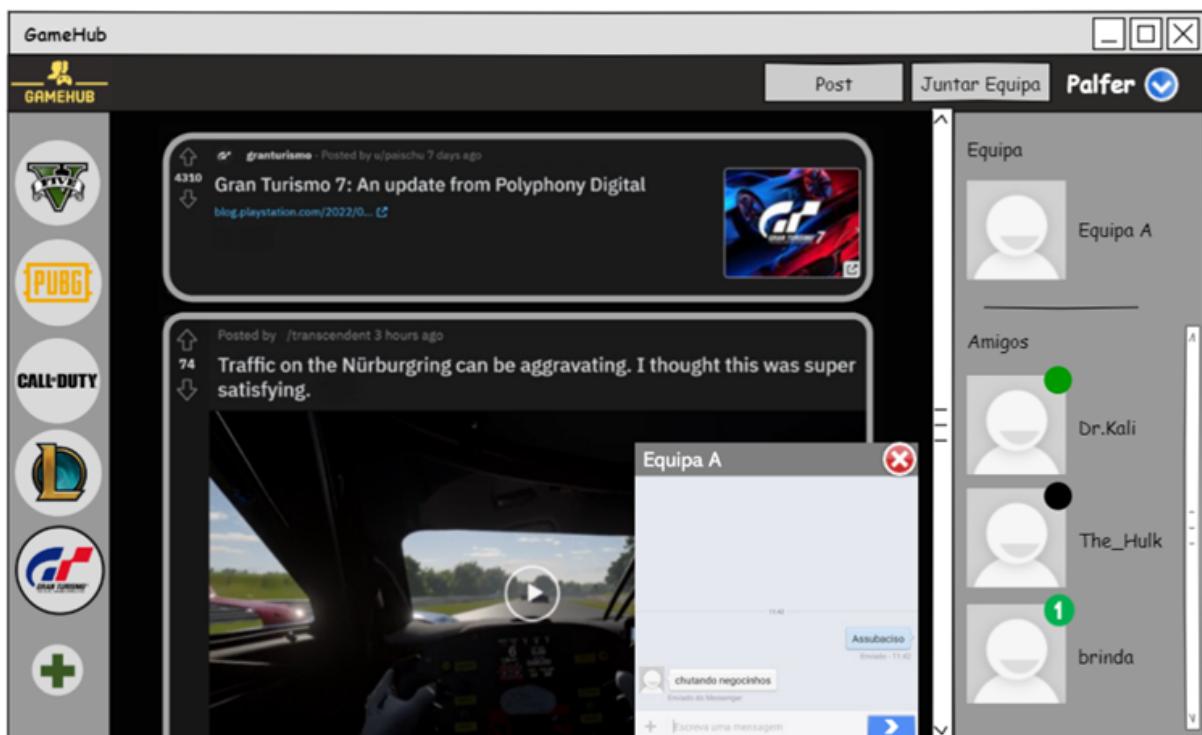


Figura 7.12: Enviar mensagem

Para além disto, serão apresentadas informações sobre estados dos amigos, bem como a quantidade

de mensagens não visualizadas por amigo. Desta forma, sempre que um amigo se encontre online aparecerá um ícone com a cor verde junto ao seu perfil, por outro lado, caso o amigo se encontre offline o ícone terá a cor preta.

Relativamente à quantidade de mensagens por ler, o ícone apresentado deverá seguir o mesmo conjunto de cores às quais se sobrepõe o número. Estas características estão evidenciadas na figura 7.12.

Por fim, a abertura de um determinado chat será dada pelo clique sobre o perfil disponível na área de amigos (área lateral direita). A realização desta funcionalidade é expressa na seguinte especificação:

Atores envolvidos - Utilizador
Pré-condição: O utilizador está autenticado no sistema
Pós-condição: A mensagem enviada é visível pelos destinatários
Descrição: Utilizador envia mensagem
Fluxo normal: 1 – Utilizador clica no destinatário pretendido para a mensagem (grupo ou utilizador) 2 – Utilizador escreve o conteúdo da mensagem 3 – Utilizador clica em enviar mensagem 4 – Sistema regista a mensagem e notifica os destinatários

Tabela 7.7: Especificação do *use case*: Enviar mensagem

8 Requisitos funcionais

Com o intuito de perceber melhor o contexto de aplicação do produto, foi necessário aplicar diferentes métodos de aquisição de requisitos. Deste modo, realizou-se:

- Um estudo do domínio em análise, através do estudo de documentação já existente;
- Um processo de observação de vários jogadores a utilizar produtos semelhantes, de modo a perceber quais seriam as metodologias de utilização mais atípicas;
- Um conjunto de entrevistas com os *stakeholders* da aplicação;
- Uma sessão de *brainstorming*.

Como resultado destas técnicas, foi possível escrever um conjunto de requisitos funcionais, para os quais se identificou os seguintes parâmetros:

- **Identificador único:** identificador que refere a forma inequívoca cada requisito;
- **Tipo de requisito:** identificador numérico referente à secção no documento onde se inclui o requisito: 9 (Funcional) ou 10–18 (Não-Funcional).
- **Use cases:** lista de use cases que dependem do requisito;
- **Descrição:** descrição da funcionalidade oferecida pelo requisito;
- **Prioridade:** nível de prioridade do requisito;
- **Fundamentação:** motivo pelo qual o requisito deve ser implementado no sistema;
- **Critério de ajuste:** critério que permite determinar o momento em que o requisito se encontra implementado no produto;
- **Origem:** tipo de *stakeholder* que deu origem ao requisito.

Assim, foram registados os seguintes requisitos funcionais:

Requisito: RF01	Tipo: 9	UC: Registar
Descrição: Um utilizador regista-se no sistema, com e-mail, username e uma palavra-passe.		
Fundamentação: O sistema requer a criação de uma conta para os utilizadores novos terem acesso às suas funcionalidades.		
Origem: Introspeção.		
Critério de ajuste: Um utilizador novo deve conseguir registar-se com os seus dados pessoais. Não é possível criar uma conta com um e-mail já existente.		
Prioridade: Must.		

Requisito: RF02

Tipo: 9

UC: Login

Descrição: O utilizador autentica-se com o email e password.

Fundamentação: De modo a ter acesso às funcionalidades do sistema, o utilizador deve-se autenticar, previamente, utilizando o email e password.

Origem: Introspecção.

Critério de ajuste: O username e email devem ser único ao utilizador. O utilizador só pode autenticar-se com uma combinação de email e password correta.

Prioridade: *Should.*

Requisito: RF03

Tipo: 9

UC: Editar perfil

Descrição: O utilizador registado edita o seu perfil.

Fundamentação: O sistema permite editar as informações da conta do utilizador.

Origem: Introspecção.

Critério de ajuste: Um utilizador já registado deve conseguir aceder ao seu perfil e editar o seu nome de utilizador, avatar e eventualmente outras informações.

Prioridade: *Could.*

Requisito: RF04

Tipo: 9

UC: Seguir jogo

Descrição: Um utilizador acede à lista de jogos.

Fundamentação: Para ser possível o utilizador ter acesso a um feed, pode consultar a lista de jogos existente na plataforma.

Origem: Observação.

Critério de ajuste: Um utilizador novo deve conseguir ver os jogos existentes na plataforma.

Prioridade: *Must.*

Requisito: RF05

Tipo: 9

UC: Aceder a feed

Descrição: Um utilizador acede ao feed de um determinado jogo.

Fundamentação: Cada jogo tem um feed personalizado que contém posts relativos a si.

Origem: Observação.

Critério de ajuste: Em cada feed de um jogo, só devem aparecer posts relativos a esse jogo.

Prioridade: *Must.*

Requisito: RF06

Tipo: 9

UC: Aceder a feed

Descrição: Um utilizador filtra o *feed* de um determinado jogo.

Fundamentação: Cada utilizador pode ver os *feeds* dos jogos que segue ordenados por diferentes formas.

Origem: Entrevista.

Critério de ajuste: Um utilizador que siga pelo menos um jogo pode escolher se pretende ver os inúmeros *posts* organizados por datas ou *likes*, de forma ascendente ou descendente.

Prioridade: *Should.*

Requisito: RF07

Tipo: 9

UC: Seguir Jogo

Descrição: Um utilizador adiciona um jogo à sua lista de jogos a seguir.

Fundamentação: O utilizador seleciona o(s) jogo(s) que pretende seguir.

Origem: Entrevista.

Critério de ajuste: Depois do jogo ser adicionado, este deve aparecer na lista de jogos a ser seguidos pelo utilizador.

Prioridade: *Must.*

Requisito: RF08

Tipo: 9

UC: Adicionar Amigo

Descrição: Um utilizador adiciona outro utilizador à sua lista de amigos.

Fundamentação: Como o objetivo do sistema é a socialização dos jogadores, ter uma lista de amigos na qual os utilizadores podem revisitar outros utilizadores é bastante relevante.

Origem: Observação

Critério de ajuste: Depois de adicionado como amigo, esse utilizador deve aparecer na lista de amigos.

Prioridade: *Should.*

Requisito: RF09

Tipo: 9

UC: Criar Post

Descrição: Um utilizador acede ao feed de um jogo e cria uma publicação multimédia.

Fundamentação: De modo a conseguir partilhar informação ao resto da comunidade de um dado jogo, um utilizador poderá partilhar uma publicação.

Origem: Observação.

Critério de ajuste: Depois de criada e submetida a publicação, esta deverá aparecer no feed do jogo.

Prioridade: *Must.*

Requisito: RF10 **Tipo:** 9 **UC:** Avaliar Post

Descrição: Um utilizador avalia um *post* de outro utilizador num determinado *feed* (*like* ou *dislike*).

Fundamentação: Esta avaliação permite aos utilizadores verificarem a qualidade dos posts e darem a sua opinião sobre os mesmos.

Origem: Observação.

Critério de ajuste: O número de *likes/dislikes* é alterado após cada avaliação do jogador.

Prioridade: *Must*.

Requisito: RF11 **Tipo:** 9 **UC:** Avaliar jogador

Descrição: Um utilizador avalia outro utilizador.

Fundamentação: Este sistema de avaliação permite não só o sistema saber que jogadores têm mau comportamento, mas permite também aos utilizadores verificarem a reputação dos jogadores com os quais vão formar grupo.

Origem: Entrevista.

Critério de ajuste: Após a avaliação, a reputação de um jogador é alterada.

Prioridade: *Should*.

Requisito: RF12 **Tipo:** 9 **UC:** Banir jogador

Descrição: Um administrador bane um jogador.

Fundamentação: Utilizadores com comportamentos abusivos poderão ser banidos por um administrador.

Origem: Entrevista.

Critério de ajuste: Um utilizador que seja banido não poderá voltar a aceder à sua conta.

Prioridade: *Should*.

Requisito: RF13 **Tipo:** 9 **UC:** Remover Post

Descrição: Um administrador poderá remover um post patilhado por um utilizador.

Fundamentação: Caso um utilizador use vocabulário ofensivo num post, o administrador poderá remover esse mesmo post.

Origem: Entrevista.

Critério de ajuste: Depois de removido, o post não deverá aparecer no feed do jogo.

Prioridade: *Should*.

Requisito: RF14 **Tipo:** 9 **UC:** Enviar Mensagem

Descrição: O utilizador envia mensagem a outro utilizador.

Fundamentação: Dois utilizadores poderão trocar mensagens diretas entre si.

Origem: Observação.

Critério de ajuste: A mensagem enviada deve aparecer nas mensagens do utilizador destinatário.

Prioridade: *Should.*

Requisito: RF15 **Tipo:** 9 **UC:** Enviar Mensagem

Descrição: O utilizador envia mensagens para o grupo.

Fundamentação: Uma vez formados os grupos, os utilizadores nesse mesmo grupo poderão trocar mensagens entre si.

Origem: Entrevista.

Critério de ajuste: As mensagens enviadas deverão aparecer no feed do grupo.

Prioridade: *Should.*

Requisito: RF16 **Tipo:** 9 **UC:** Criar *post*

Descrição: Um utilizador cria um grupo.

Fundamentação: De modo a formar um grupo, um utilizador precisa de primeiramente o criar.

Origem: Introspeção.

Critério de ajuste: O grupo após ser criado tem que aparecer a todos os utilizadores que seguem o jogo.

Prioridade: *Must.*

Requisito: RF17 **Tipo:** 9 **UC:** Juntar a grupo

Descrição: O utilizador deve ter a possibilidade de juntar-se a um grupo de pessoas para jogarem em conjunto.

Fundamentação: Com a existência de publicações para pedidos de criação de grupos, deve ser possível aos utilizadores endereçarem nesses grupos, de modo a jogarem em conjunto.

Origem: Introspeção.

Critério de ajuste: O utilizador deve ficar registado no grupo em que entrou.

Prioridade: *Must.*

Requisito: RF18 **Tipo:** 9 **UC:** Adicionar amigo

Descrição: Depois de um amigo ser adicionado, um dos utilizadores pode poder terminar esta amizade.

Fundamentação: Caso um utilizador deixe de querer interagir com outro, deve poder remover este da sua lista de amigos.

Origem: Entrevista.

Critério de ajuste: Após a remoção da amizade, o amigo removido deve desaparecer da lista de amigo do utilizador.

Prioridade: *Should.*

Requisito: RF19 **Tipo:** 9 **UC:** Avaliar jogador

Descrição: Após a adição de um jogador na lista de amigos, o utilizador poderá visualizar o seu perfil.

Fundamentação: Um utilizador poderá visualizar o conjunto de informações referentes a ao perfil - recompensas, classificação, etc. - de um dos seus amigos.

Origem: Entrevista.

Critério de ajuste: O utilizador deverá ficar ciente das características do perfil de outro jogador (amigo).

Prioridade: *Should.*

Requisito: RF20 **Tipo:** 9 **UC:** Logout

Descrição: Um utilizador pode terminar sessão.

Fundamentação: Após a utilização do sistema o utilizador pode se quiser sair da sua conta.

Origem: Observação.

Critério de ajuste: O utilizador deverá desvincular a sua conta e aparecer *offline* para os seus amigos.

Prioridade: *Must.*

Requisito: RF21 **Tipo:** 9 **UC:** Enviar Mensagem

Descrição: Um utilizador deverá ser capaz de perceber a receção de mensagens.

Fundamentação: Para determinar a necessidade de visualizar as mensagens não lidas, o utilizador deverá de ser capaz de determinar a sua receção e identificar o número de mensagens não lidas.

Origem: Observação.

Critério de ajuste: Após a receção de mensagens o utilizador deverá ser notificado da sua receção e ser capaz de identificar a quantidade de mensagens não lidas por cada jogador.

Prioridade: *Could.*

9 Requisitos não funcionais

9.1 Requisitos de Aparência

Requisito: RNF01

Tipo: 10

Descrição: O produto deve seguir a paleta de cores da marca GameHub.

Fundamentação: Desta forma permite a todos os utilizadores uma melhor associação entre o produto e a marca GameHub.

Origem: Introspeção.

Critério de ajuste: É preciso que se verifique uma predominância de 70% das cores da marca nos elementos do produto.

Prioridade: *Could.*

Requisito: RNF02

Tipo: 10

Descrição: O produto deve parecer fidedigno.

Fundamentação: Assim iremos transmitir confiança aos utilizadores que utilizem o nosso produto, favorecendo a expansão da marca.

Origem: Inquérito.

Critério de ajuste: É necessário fazer referência à ligação com a Epic Games de modo a transferir a confiança que as pessoas têm nesta organização para a GameHub.

Prioridade: *Must.*

x

Requisito: RNF03

Tipo: 10

Descrição: A interface principal do produto deve distinguir o *feed*, os jogos e os amigos.

Fundamentação: Para uma melhor interação com os utilizadores.

Origem: Introspeção.

Critério de ajuste: É necessário existir uma divisão entre os componentes referidos, não misturando tudo na mesma secção.

Prioridade: *Must.*

9.2 Requisitos de Usabilidade

Requisito: RNF04

Tipo: 11

Descrição: O produto deve ser fácil de usar para utilizadores sem experiência de utilização da aplicação.

Fundamentação: Para garantir uma boa experiência ao utilizador.

Origem: Introspeção.

Critério de ajuste: 90% de utilizadores sem experiência devem ser capazes de aceder a qualquer funcionalidade do sistema em menos de 1 minuto.

Prioridade: *Should.*

Requisito: RNF05

Tipo: 11

Descrição: O produto deve ser considerado fácil de utilizar.

Fundamentação: Garantir satisfação dos utilizadores.

Origem: Introspeção.

Critério de ajuste: O produto deve ser considerado de fácil utilização, por pelo menos 90% dos utilizadores.

Prioridade: *Should.*

Requisito: RNF06

Tipo: 11

Descrição: O produto deve ter as funcionalidades acessíveis com poucos cliques.

Fundamentação: Garantir um rápido acesso às funcionalidades.

Origem: Inquérito.

Critério de ajuste: O utilizador deve conseguir aceder às diversas funcionalidades do sistema em menos de 6 cliques.

Prioridade: *Could.*

Requisito: RNF07

Tipo: 12

Descrição: O produto deve prevenir que os utilizadores introduzam dados inválidos nos campos de preenchimento;

Fundamentação: Garantir que os utilizador introduzam dados inválidos.

Origem: Introspeção.

Critério de ajuste: Nenhum dado errado poderá ser submetido pelo utilizador.

Prioridade: *Should.*

9.3 Requisitos de Performance

Requisito: RNF08 **Tipo:** 12

Descrição: O produto deve ser responsivo.

Fundamentação: Garantir uma boa interação com o utilizador.

Origem: Introspeção.

Critério de ajuste: Em 90% dos casos, qualquer ação no sistema não pode demorar mais de 3 segundos a processar.

Prioridade: *Should.*

Requisito: RNF9 **Tipo:** 12

Descrição: O produto deve estar altamente disponível.

Fundamentação: Minimizar os tempos em que o produto não se encontra disponível para realização de manutenção e adição de novas funcionalidades.

Origem: Instrospeção.

Critério de ajuste: A plataforma deve estar funcional 97% do tempo do ano (10 dias para manutenção).

Prioridade: *Should.*

9.4 Requisitos Operacionais

Requisito: RNF10 **Tipo:** 13

Descrição: O produto deve ser funcional em diferentes *browsers*.

Fundamentação: Permitir aos utilizadores a escolha de forma livre do seu *browser* de eleição.

Origem: Inquérito.

Critério de ajuste: A aplicação deve funcionar pelo menos nos 3 browsers mais utilizados (Chrome, Safari e Firefox).

Prioridade: *Must.*

Requisito: RNF11

Tipo: 13

Descrição: O produto deve estar disponível num vasto leque de dispositivos.

Fundamentação: Permitir ao utilizador utilizar o produto no seu dispositivo favorito.

Origem: Inquérito.

Critério de ajuste: A aplicação deve ser compatível com pelo menos 90% dos computadores e dispositivos móveis.

Prioridade: *Should.*

9.5 Requisitos de Manutenção e Suporte

Requisito: RNF12

Tipo: 14

Descrição: A aplicação deve permitir a adição de novos jogos no sistema.

Fundamentação: Com o passar do tempo novos jogos são lançados e portanto será necessário atualizar o produto para poder incorporar estes.

Origem: Introspeção.

Critério de ajuste: Quando um jogo *multiplayer* se tornar popular deve ser adicionado num prazo inferior a 3 dias.

Prioridade: *Must.*

Requisito: RNF13

Tipo: 14

Descrição: A equipa de desenvolvimento deve corrigir problemas técnicos ao longo do ciclo de vida da plataforma.

Fundamentação: De modo a não prejudicar a experiência de utilização dos utilizadores finais.

Origem: Introspeção.

Critério de ajuste: Após um *bug* ser reportado, este deve ser resolvido no prazo de 1 dia a 2 semanas consoante a sua complexidade e relevância.

Prioridade: *Should.*

9.6 Requisitos de Segurança

Requisito: RNF14

Tipo: 15

Descrição: Apenas o administrador pode banir utilizadores e remover *posts* de um utilizador arbitrário.

Fundamentação: De modo a manter um ambiente positivo de convívio no produto.

Origem: Inquérito.

Critério de ajuste: Nenhum utilizador poderá banir um outro utilizador ou remover *posts* que não lhe pertençam.

Prioridade: *Must.*

Requisito: RNF15

Tipo: 15

Descrição: Os dados dos utilizadores devem ser protegidos.

Fundamentação: De modo a manter a privacidade dos dados dos utilizadores.

Origem: Introspeção.

Critério de ajuste: O produto deve proteger a informação dos utilizadores de acordo com a política de proteção de dados.

Prioridade: *Could.*

Requisito: RNF16

Tipo: 15

Descrição: O produto deve manter os dados dos utilizadores incompreensíveis para o exterior.

Fundamentação: De modo a manter a privacidade dos dados dos utilizadores.

Origem: Introspeção.

Critério de ajuste: O produto deve encriptar os dados persistentes.

Prioridade: *Could.*

9.7 Requisitos Legais

Requisito: RNF17

Tipo: 17

Descrição: Todo o processamento de dados pessoais deve estar de acordo com o GDPR e com as leis nacionais relevantes à privacidade e proteção de dados.

Fundamentação: Manter a legalidade do produto.

Origem: Reunião.

Critério de ajuste: As regras definidas não podem ser violadas.

Prioridade: *Should.*

10 Alcance do sistema e contexto

10.1 Contexto de negócio

A figura seguinte representa os principais acessos realizados por utilizadores humanos ou por servidores automatizados às componentes do sistema.

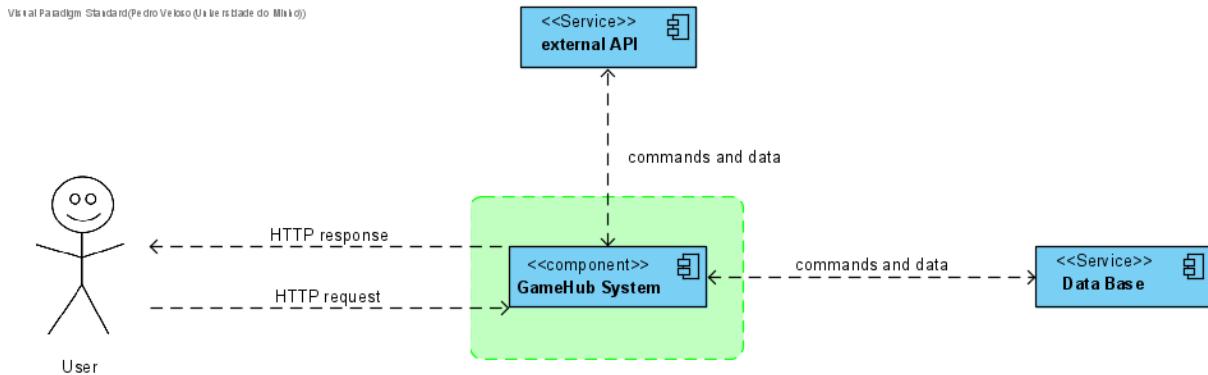


Figura 10.1: Diagrama do contexto de negócio

Ao observar a figura é de notar a existência de dois serviços externos que fornecem funcionalidades ao sistema a desenvolver - base de dados e uma API externa. Para além destes serviços também é possível identificar a existência de um utilizador humano responsável por interagir com o sistema.

10.1.1 APIs externas

Ao analisar os requisitos anteriormente referidos foi identificada a necessidade de registar permanentemente ficheiros multimédia - vídeos ou imagens. Sendo a operação de transferência de ficheiros entre servidores uma transação computacionalmente exigente, optou-se por recorrer a um serviço externo para a sua realização.

Assim, elegeu-se o *Cloudinary* como responsável por armazenar todos os ficheiros submetidos pelo utilizador e posteriormente disponibilizá-los através de um determinado *link*. Este prestador de serviço está especializado na criação de uma Media API capaz de lidar com vários pedidos em simultâneos. Para além disto, desenvolveu um conjunto de servidores que permitem aos seus usuários obter taxas de transferência mais altas.

A utilização deste serviço torna-se uma mais valia para o desenvolvimento do sistema, na medida em que permite lidar com um conjunto de pedidos simultâneos muito superiores, aumentando assim a sua performance. Porém, a utilização desta API apresenta os seguintes riscos identificados:

- Uma vez que as API's são externas ao sistema a desenvolver, a comunicação com as mesmas poderá sofrer atrasos no tempo de resposta, ou até falha, sendo desta forma necessário implementar opções para mitigar este problema. Em caso de falha deste serviço será impossível apresentar aos utilizadores o conteúdo dos *posts* realizados ou realizar novos *posts*;
- Em caso de falha na comunicação entre os utilizadores e os servidores - API ou servidor aplicacional a desenvolver - poderá ser originada uma incoerência entre os dados presentes no servidor aplicacional e o conteúdo presente nos servidores da API.

Para além dos serviços da *Cloudinary* decidiu-se utilizar também os da *newsapi*. Esta nova API, fornece notícias relativas aos jogos disponibilizados no sistema. Assim, a utilização deste serviço permite obter dados, que possibilitam a realização automática de publicações nos *feeds* de cada jogo, com informações pertinentes aos utilizadores.

A conexão do sistema a desenvolver com a API da *newsapi*, permite acrescentar qualidade ao sistema, porém implica assumir que a incapacidade de comunicação com os serviços externos invalidam a criação das publicações automáticas.

10.1.2 Base de dados

O serviço relativo à *Database* representa o servidor de base de dados, onde os dados serão registados, à qual o componente principal do GameHub acede.

Este serviço torna-se indispensável ao sistema a desenvolver, uma vez que permite registar de forma permanente as suas informações pertinentes. Porém, é necessário considerar os seguintes riscos a ele associado:

- Uma vez que este servidor se pode encontrar num ambiente diferente do *localhost*, pode existir riscos associados com a comunicação. Nesse caso, poderá ocorrer um aumento do tempo de espera para obter os dados desejados devido à latência imposta pela comunicação via internet;
- Os dados transportados entre o servidor do GameHub e o servidor da base de dados contém informação sensível que deve ser protegida de possíveis ataques.

10.1.3 Utilizador

Por fim, identificou-se a existência de um interveniente humano. Este interveniente representa a entidade que visualiza a interface gráfica da aplicação.

Assim, de modo a disponibilizar-lhe a interface deve ser utilizado o protocolo HTTP. Este protocolo apesar de ser utilizado nos serviços WEB apresenta algumas vulnerabilidades que deverão ser consideradas. Para o sistema a desenvolver deve-se ter em atenção aos seguintes riscos:

- Ao utilizar o protocolo HTTP para obter a interface, é necessário ter em atenção que poderá existir uma disruptão do tempo de resposta (ou falha do serviço) infligida pelas características dos serviços web;
- A utilização do protocolo HTTP não apresenta nenhum tipo de segurança. Neste tipo de protocolo a informação entre o cliente e o servidor viaja na internet sem qualquer codificação, podendo partilhar dados críticos com pessoas não autorizadas.

10.2 Contexto técnico

O diagrama de contexto técnico seguinte, representa as ligações entre os vários nodos existentes no contexto do sistema, juntamente com as ligações técnicas entre eles.

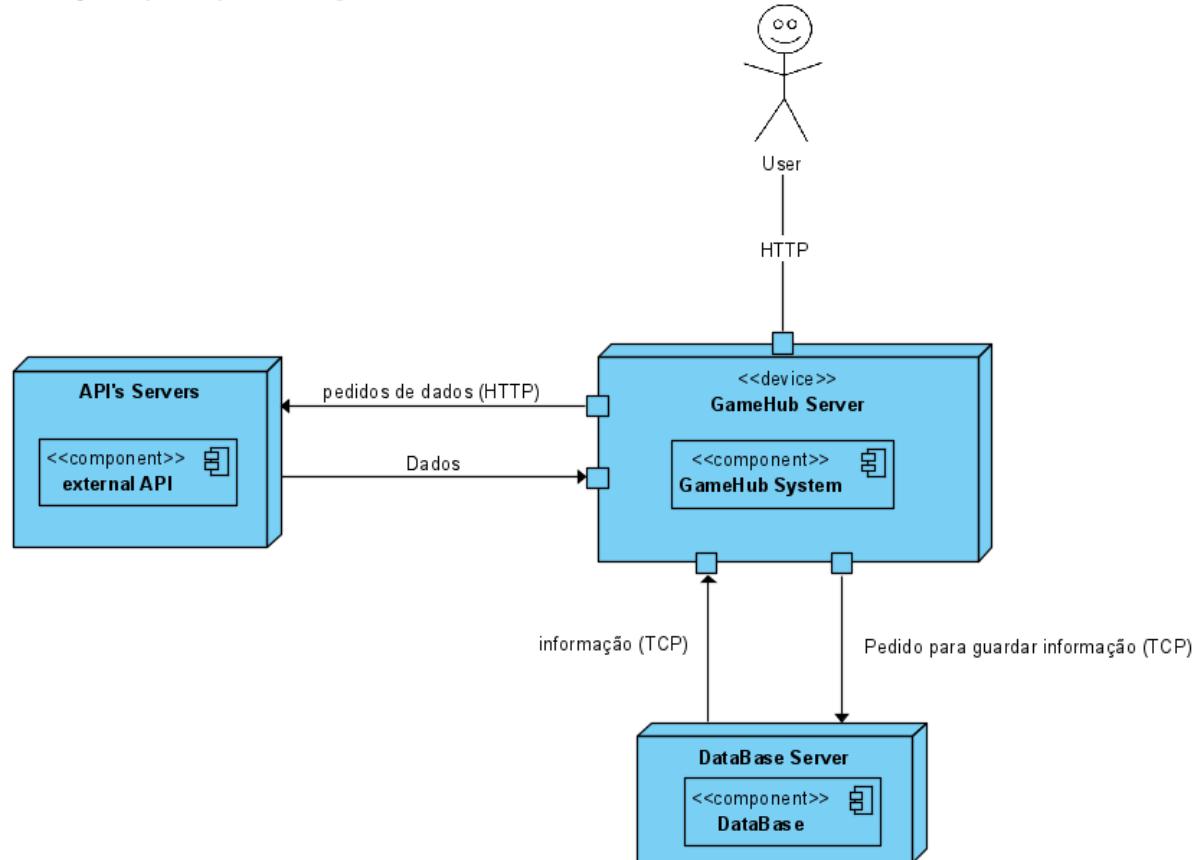


Figura 10.2: Diagrama de contexto técnico

Ao observar a figura é possível identificar a necessidade de comunicação entre o sistema e mais dois componentes externos - *API server* e *Database server*.

A comunicação entre o sistema e estes dois componentes externos é realizada via internet, contudo serão utilizados protocolos distintos para cada uma das comunicações.

Para a comunicação entre o sistema e os servidores da API deverá ser utilizado o protocolo HTTP. Estes servidores ao receber os pedidos HTTP deverão registar os ficheiros recebidos e responder de volta, com o resultado da operação através de um outro ficheiro com formato JSON. Neste ficheiro deverá ser disponibilizado o resultado da operação - sucesso ou insucesso - para além do *link* que permite aceder aos dados registados.

Por fim, os servidores da base de dados deverão comunicar com o sistema através de conexões TCP. Estes servidores representam o conjunto dos servidores onde estará hospedada a base de dados utilizada pelo sistema. A comunicação com estes servidores deverá resultar no conjunto de dados que o sistema pretenda obter.

11 Vista de blocos de construção

11.1 Nível 1

O sistema a desenvolver é constituído por quatro componentes (figura 11.1) - componente da Frontend, componente da Backend, componente para atribuir recompensas e a componente para criar publicações com notícias.

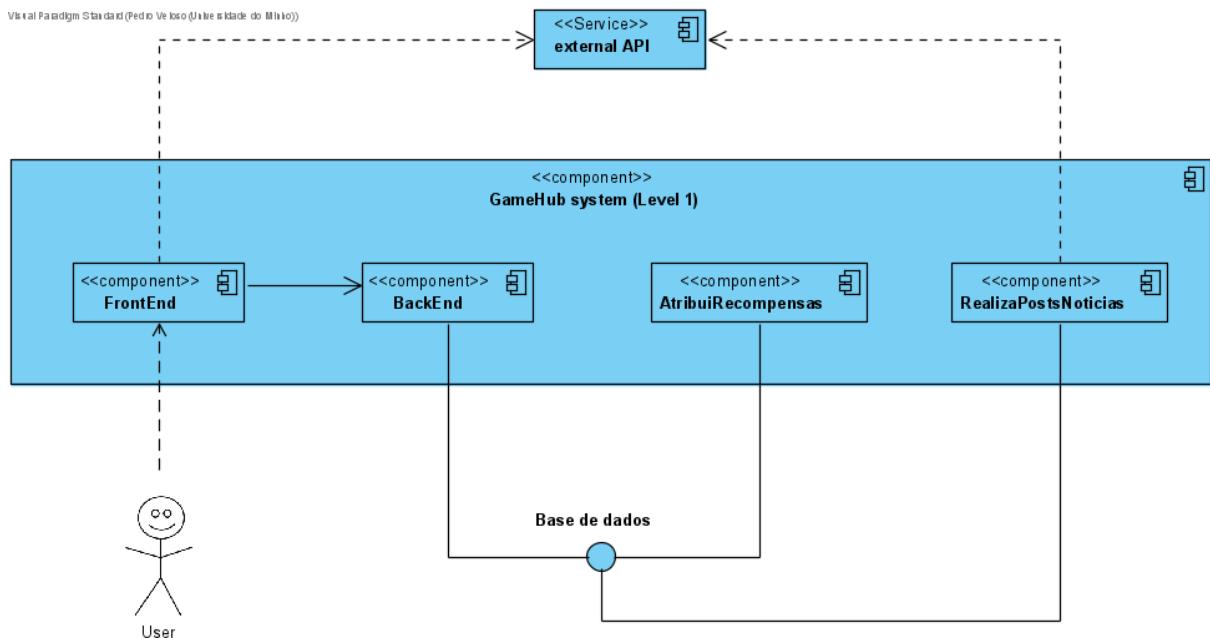


Figura 11.1: Diagrama de componentes (nível 1)

O sistema necessita de ter presente os componentes Frontend e Backend de modo a ilustrar de forma clara o que é executado pelo servidor e o que é executado diretamente no web browser.

11.1.1 Componente - Frontend

Este componente deverá permitir aos utilizadores interagirem com as funcionalidades evidenciadas nos requisitos referidos anteriormente.

Assim, deverá desenvolver-se este componente através de HTML, CSS e Javascript. Contudo, para auxiliar no seu desenvolvimento recorre-se à *framework vue.js*.

Uma vez que esta componente é isolada das restantes componentes, deverá ser estabelecida uma comunicação entre ambas de modo a assegurar as dependências existentes entre elas. Desta forma, deverá ser possível obter os dados necessários para a elaboração da interface, com a realização de pedidos à API externa e à API desenvolvida para suportar as restantes componentes.

11.1.2 Componente - Backend

Este componente incorpora toda a lógica de negócio permitindo desta forma o processamento de eventos e a comunicação com a base de dados.

Assim, desenvolveu-se esta componente recorrendo à linguagem de programação JAVA. Para auxiliar no desenvolvimento deste componente utiliza-se as seguintes *frameworks*: *Spring* e *Hibernate*.

A *framework Spring* permitirá o desenvolvimento de *REST Controllers*. Estes *controllers* possibilitam o mapeamento de pedidos HTTP em métodos JAVA. Desta forma, torna-se mais simples aplicar a lógica de negócio e responder de forma adequada aos pedidos recebidos.

Por fim, uma vez que esta componente necessita de registrar dados de forma permanente através de uma base de dados relacional, optou-se por recorrer a uma *framework ORM - Hibernate*. Esta *framework* é bastante famosa no mercado devido à sua capacidade de fornecer um alto desempenho e confiabilidade.

11.1.3 Componente - AtribuiRecompensas

Dada a necessidade de atribuir recompensas aos utilizadores que venceram determinados torneios, desenvolveu-se este componente. Este componente é responsável por correr num processo separado dos restantes componentes e deverá diariamente verificar o *gamer* que ganhou um torneio terminado.

Assim, desenvolveu-se um *script* em *Python* responsável por verificar os dados presentes na base de dados e atribuir recompensas. Para garantir que os utilizadores ganham as suas recompensas este *script* deverá correr diariamente a uma hora determinada (1 hora da manhã).

11.1.4 Componente - RealizaPostsNoticias

Para permitir uma melhor imersão dos utilizadores no sistema, deve-se garantir que as publicações disponibilizadas aos utilizadores são adicionadas constantemente. Assim, de modo a garantir que os utilizadores do sistema tenham ao seu dispor um conjunto numeroso e pertinente de publicações, decidiu-se inserir diariamente novas publicações com notícias relativas aos jogos existentes.

Desta forma, desenvolveu-se um componente responsável por aceder à API *newsapi* e publicar novos *posts* com os dados recebidos. Esta componente foi desenvolvida em *JAVA*, recorrendo às tarefas *Scheduling* disponibilizadas pela *framework Spring*.

11.2 Nível 2 - Backend

De modo a que esta componente - BackEnd - seja capaz de lidar com as várias funcionalidades esperadas, desenvolveu-se um conjunto de classes para as suportar. Assim, apresenta-se o seguinte diagrama de classes:

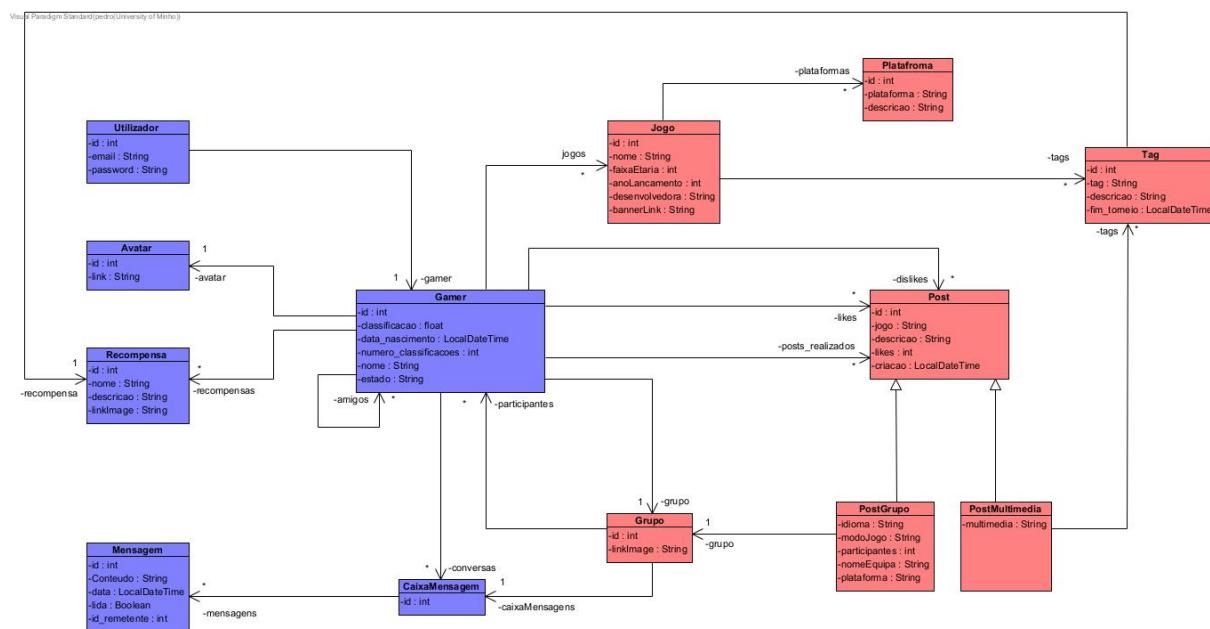


Figura 11.2: Diagrama de classe - PIM

Ao analisar este diagrama é possível perceber a existência das 13 classes seguintes:

- **Utilizador**: Esta classe é responsável por guardar os dados do utilizador registados para posterior *login*. Esta classe regista informação relativa ao email e *password* escolhidos pelo utilizador durante

o seu registo. Para além disto esta classe conhece o *gamer* a que este utilizador corresponde. A separação destes dados da classe *Gamer* permite que se aceda a determinados conjunto de dados sem carregar a informação desnecessária para certas funções.

- **Avatar:** Esta classe representa a imagem de perfil utilizada por um determinado *gamer*. Atualmente esta classe apenas incorpora o link do avatar atual, contudo espera-se que no futuro seja possível ao utilizador personalizar o avatar ao seu gosto. Assim, desenvolveu-se esta classe de modo facilitar a adição de nova informação imposta pela funcionalidade.
- **Recompensa:** Esta classe representa uma recompensa que um *gamer* pode receber ao ganhar torneios. Desta forma é possível identificar o nome da recompensa, a sua descrição (informação relativa à origem da recompensa) e um link para a imagem de do troféu ganho.
- **Mensagem:** Esta classe representa uma mensagem recebida ou enviada por um *gamer*. Assim, é possível identificar informações pertinentes, como por exemplo: conteúdo da mensagem, o identificador único do remetente, a data de envio e um booleano a identificar se essa mensagem já foi lida.
- **CaixaMensagem:** Esta classe permite identificar determinadas conversas e as suas respetivas mensagens. Assim, é possível identificar uma caixa de mensagens por cada *gamer* e os seus respetivos amigos e por cada *gamer* e um determinado grupo.
- **Gamer:** Esta classe regista toda a informação pertinente de um determinado *gamer*. Ao observar os atributos da classe é possível identificar a informação presente no perfil de um *gamer*. Por fim, esta classe apresenta algumas ligações com outras classes, sendo de notar a ligação recursiva de *Gamer* para *Gamer*. Esta ligação representa a lista de amigos que cada *gamer* foi construindo ao longo da utilização do sistema.
- **Grupo:** Esta classe representa o grupo em que um determinado *gamer* está presente. Para cada grupo é possível existir uma imagem que o represente.
- **Jogo:** Esta classe contém toda a informação de um determinado jogo. Assim, regista-se o nome do jogo, ano de lançamento, desenvolvedora, idade mínima para o jogar e um *link* para o *banner* do jogo, criado especificamente para ser apresentado na interface ao utilizador.
- **Post:** Esta classe representa as publicações que os utilizadores irão realizar ao longo da utilização do sistema. Porém, é possível diferenciar a existência de duas publicações diferentes - publicações para criar um grupo ou publicações para partilhar experiências relativas ao jogo.
- **PostGrupo:** Esta classe generaliza a classe *Post* e apresenta as informações específicas para as publicações responsáveis pela criação de um grupo. A criação deste tipo de publicação implica a criação de um grupo, onde o usuário que realizou a publicação irá fazer parte.
- **PostMultimedia:** Esta classe generaliza a classe *Post* e apresenta as informações específicas para as publicações responsáveis por partilhar experiências relativas ao jogo. No caso de existir algum conteúdo de multimédia anexado à publicação deverá ser registado o *link* de acesso ao mesmo.
- **Plataforma:** Esta classe caracteriza as plataformas (exemplo: PC, PS5 ou XBOX X/S) em que o jogo pode estar presente.
- **Tag:** Esta classe foi desenvolvida para dar suporte aos torneios de um determinado jogo. Assim, para participar num determinado torneio, o utilizador deverá utilizar a tag que o identifica. Posteriormente, o utilizador vencedor deverá receber a recompensa associada à tag. Porém, também é possível identificar tags que não representam torneios, contudo, nestes casos a recompensa e a data de término do torneio não apresentam um valor associado.

11.3 Nível 3 - Backend

Ao analisar o diagrama de classes anterior e a tecnologia identificada para o desenvolvimento do sistema, generalizou-se o diagrama já apresentado para o seguinte diagrama de classes:

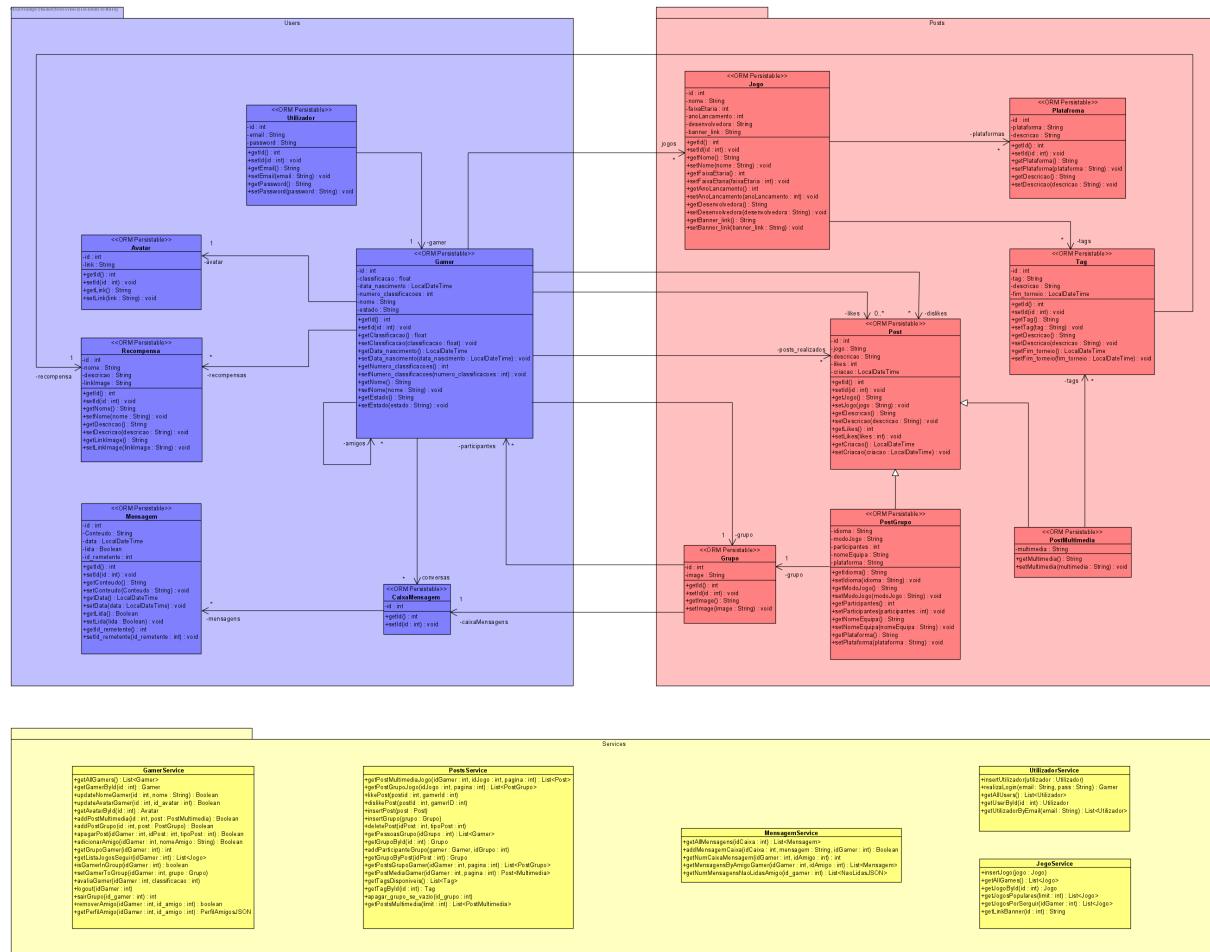


Figura 11.3: Diagrama de classe - PSM

De modo a organizar as várias classes já apresentadas desenvolveu-se a criação de *packages* - Users, Posts e Services. Assim, no *package User* encontra-se as classes relacionadas com os usuários enquanto no *package Posts* encontram-se as classes relativas à realização de publicações e aos jogos existentes. Por fim, no *package Service* encontra-se os *Services* criados para suportar o sistema.

Nesta fase, optou-se também por dividir as operações da lógica de negócios por *Beans*. Os Java *Beans* são componentes de um sistema com a função de executar tarefas para um determinado cliente. Estes componentes são também caracterizados pela sua portabilidade, segurança e rapidez. Desta forma, ao analisar a *framework String* percebeu-se que a criação de *Beans* era dada por serviços.

Assim, dividiu-se a arquitetura do sistema em cinco áreas distintas. Cada uma das áreas deu origem a um dos seguintes serviços:

- **GameService**: Este serviço é responsável por trabalhar com informações relativas aos *gamers*. Desta forma, definiu-se o conjunto de métodos identificados no diagrama 11.3.
 - **PostsService**: Neste serviço definiu-se um conjunto de métodos que envolvem as publicações de forma direta.
 - **MensagemService**: O serviço relativo às mensagens, permite trabalhar diretamente com as caixas de mensagens, sem ser necessário passar por outras classes. Este serviço define métodos que permitem obter as mensagens de determinadas conversas ou enviar novas mensagens.

- **UtilizadorService**: Este serviço permite suportar as funcionalidades de *login* e registo de novos utilizadores do sistema.
- **JogoService**: Este serviço define um conjunto de métodos que permitem aceder de forma direta aos objetos relativos aos jogos suportados no sistema.

Por fim, neste novo diagrama é possível perceber a existência de um estereótipo "ORM Persistence" em determinadas classes. Este estereótipo refere a necessidade de armazenar os dados presentes nessa classe de forma persistente. Tal como referido anteriormente, para persistir os dados recorreu-se à framework *ORM Hibernate*. Esta framework implica a anotação do código gerado e permite através dessas anotações trabalhar de forma transparente com a base de dados.

Assim, os serviços gerados devem conhecer os *CrudRepositorys* de modo a aceder aos dados presentes na base de dados.

12 Vista de execução

A interação do utilizador com a aplicação GameHub é bastante simples. De seguida são apresentados alguns exemplos de diagramas de sequência para as principais funcionalidades do sistema. Para as outras irá ser realizada apenas uma breve explicação.

12.1 Eliminar Post

Na aplicação Gamehub os utilizadores podem remover os *posts* que realizaram. Existem ainda administradores que podem remover qualquer *post* que exista que tenha sido considerado impróprio. O método para realizar tal tarefa é bastante simples, sendo apenas necessário eliminar um *post* dado o seu id, de modo que não será apresentado o seu diagrama de sequência.

12.2 Banir Jogador

Esta ação apenas pode ser realizada por um administrador e como o próprio nome indica, elimina a conta de um jogador. Mais uma vez sendo apenas a remoção dos dados persistidos, não foi realizado nenhum diagrama de sequência.

12.3 Registar

Para um utilizador usufruir das funcionalidades do GameHub precisa de se registar, para tal precisa de fornecer um *email*, *password* e nome de utilizador. Estas informações são usadas para criar um novo objeto utilizador e *gamer*, de modo que a sua visualização num diagrama de sequência não é necessária.

12.4 Login

Para realizar o *login* um utilizador tem que fornecer o seu *email* e *password*, sendo estes então verificados pelo sistema, o utilizador entra na aplicação. Mais uma vez sendo um método trivial, o diagrama de sequência não se encontra representado.

12.5 Enviar mensagem

Durante a utilização da aplicação um utilizador pode enviar mensagens diretamente para um utilizador, ou para um grupo. O método de enviar uma mensagem consiste em colocar uma nova mensagem na caixa de mensagens, quer esta seja pertencente a um grupo, ou a uma conversa entre dois utilizadores. Em seguida apresenta-se o método para enviar uma mensagem diretamente para um amigo:

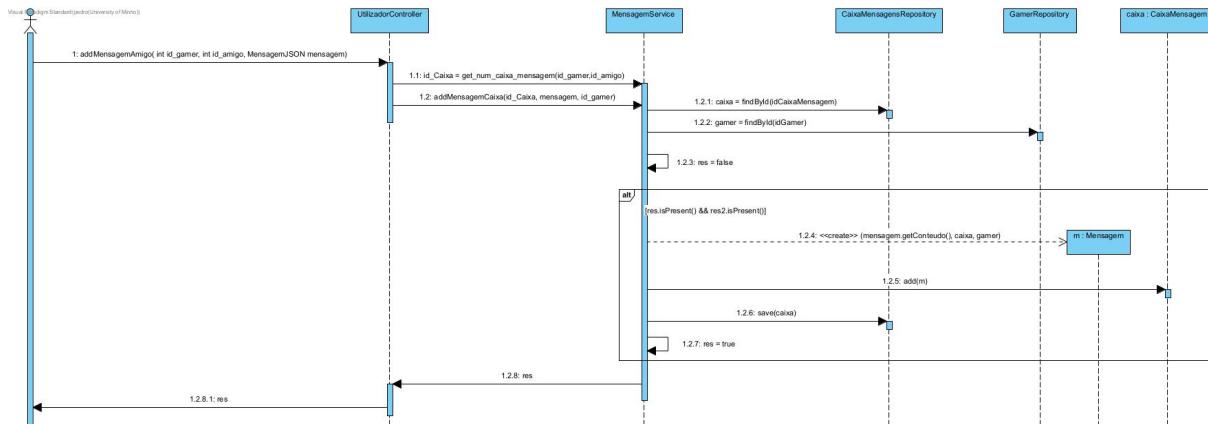


Figura 12.1: Diagrama de Sequência - Enviar mensagem amigo

Para enviar uma mensagem para um grupo, o método acaba por ser ligeiramente diferente encontrando-se em seguida:

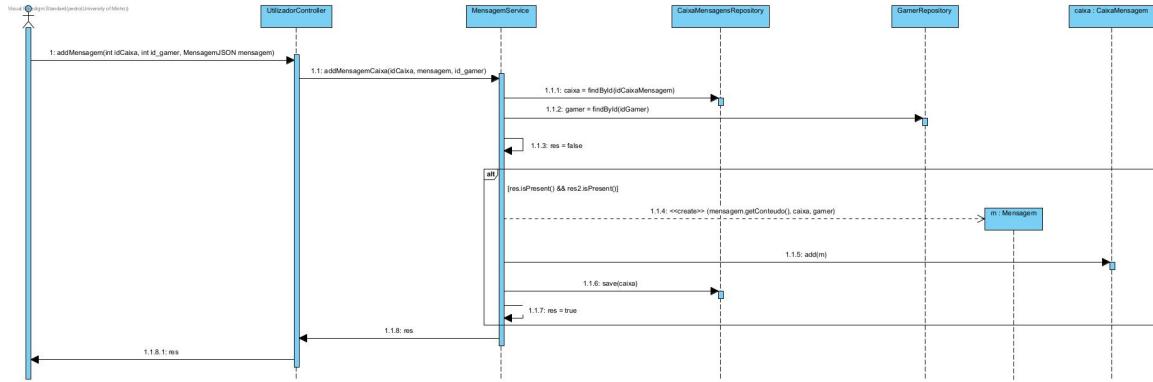


Figura 12.2: Diagrama de Sequência - Enviar mensagem num grupo

12.6 Aceder feed

Para poder aceder ao *feed* de um determinado jogo foi necessário realizar um método que nos retorne todos os *posts* multimédia de um determinado jogo. Em seguida encontra-se a implementação desse mesmo método de forma esquemática:

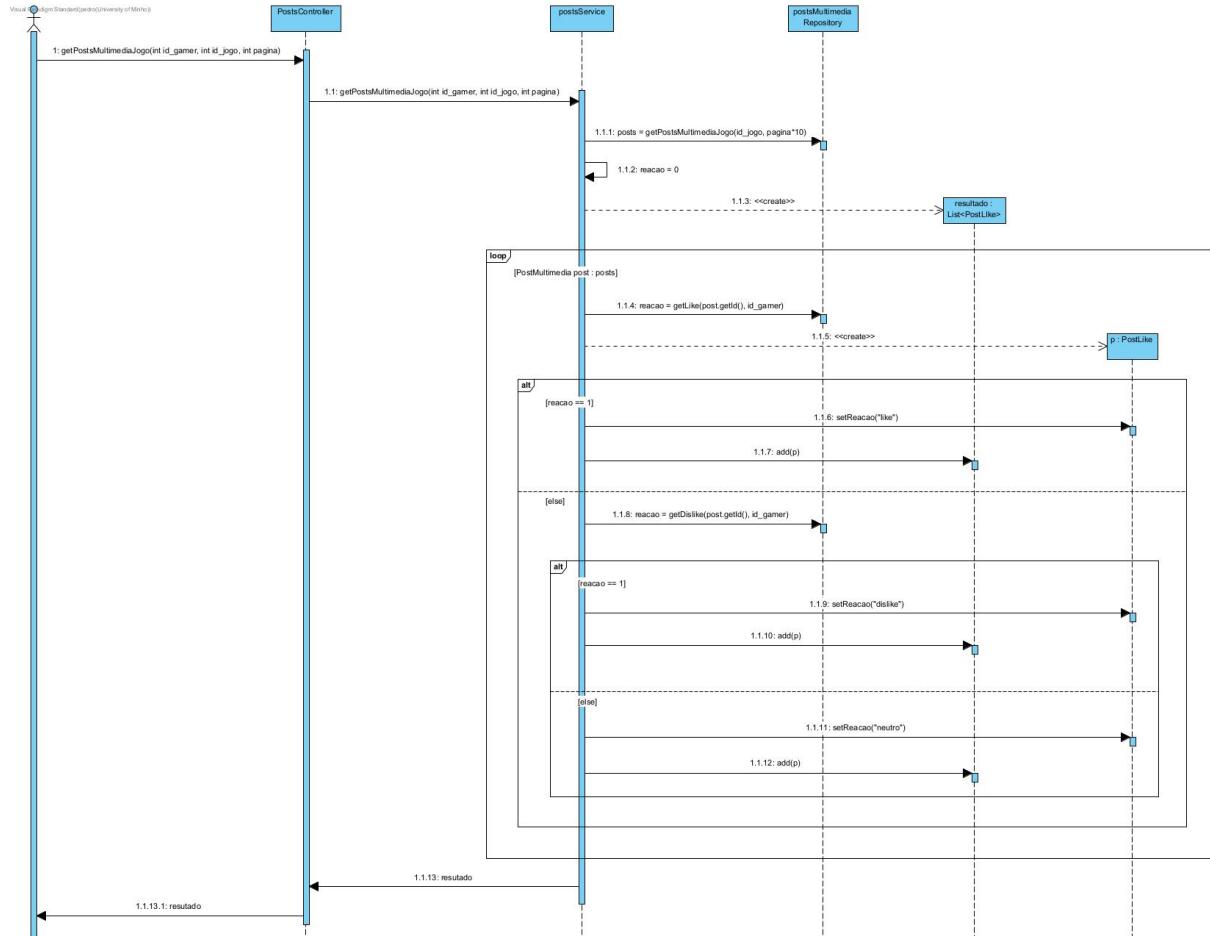


Figura 12.3: Diagrama de Sequência - Aceder ao *feed*

12.7 Adicionar amigo

Um utilizador pode adicionar um outro como amigo. Assim foi preciso criar um método que adiciona uma ligação entre estes dois. Sendo algo bastante lógico, não foi representado num diagrama.

12.8 Avaliar jogador

Todos os utilizadores têm uma reputação no GameHub. Assim um utilizador pode avaliar um outro aumentando no avaliado o número de avaliações e atualizada a sua classificação média. Por ser muito simples não indicaremos o diagrama de sequência.

12.9 Juntar a grupo

Poder juntar a um grupo é fundamental para poder jogar com outras pessoas da plataforma. Assim foi necessário a criação do método **juntar grupo** que se encontra em seguida esquematizado:

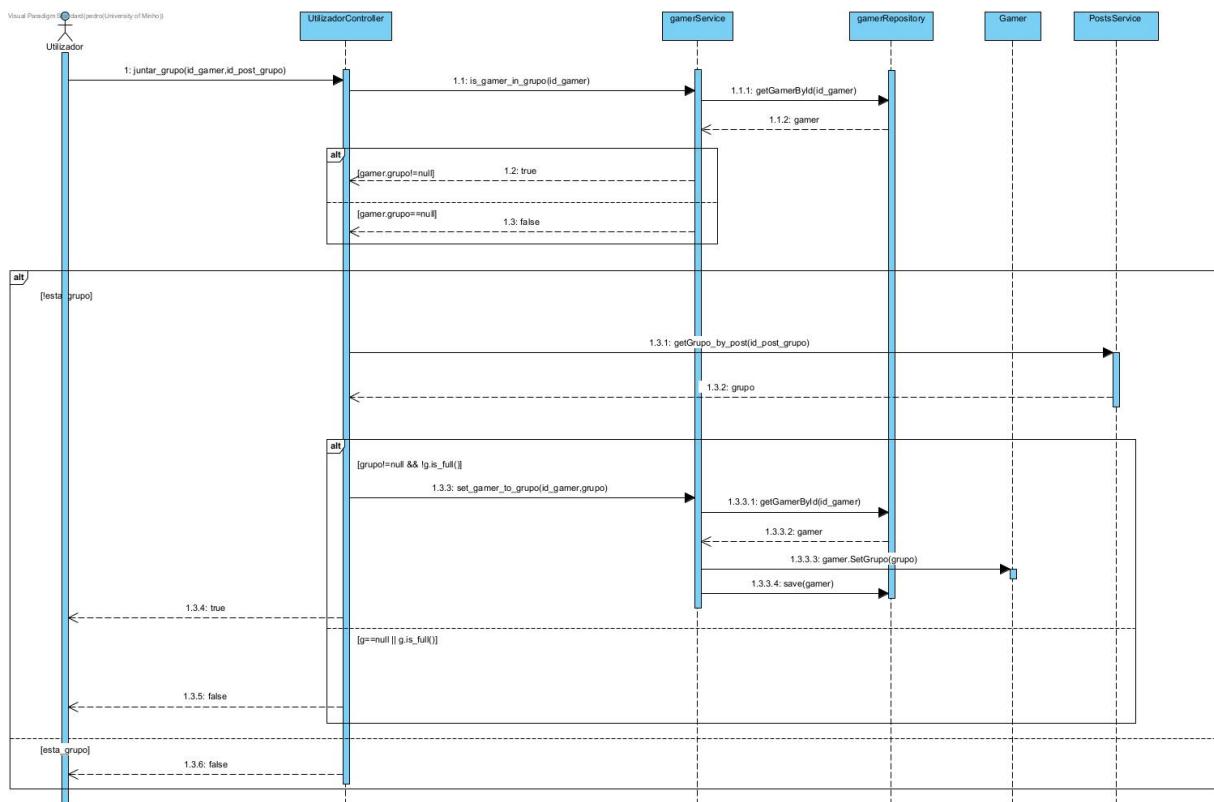


Figura 12.4: Diagrama de Sequência - Juntar a grupo

12.10 Seguir jogo

Para aceder aos diversos *feeds* de jogos existentes, um utilizador tem que seguir jogos. Para tal foi necessário criar um método que adiciona uma relação entre um jogador e um jogo. Pela simplicidade do mesmo, este não foi apresentado num diagrama de sequência.

12.11 Editar perfil

Por algum motivo um utilizador pode querer alterar algum dos campos relativos à sua conta daí a necessidade de métodos para alteração dos dados do perfil. É possível a alteração do nome do perfil, assim como o avatar. Devido a estes métodos serem uma simples alteração de uma variável de um objecto, os diagramas de sequência correspondentes não se justificam.

12.12 Criar post grupo

Na plataforma GameHub existe a criação de *posts* de grupo, estes servem para um jogador poder procurar outros com quem possa jogar. Assim o método **Create_post_grupo** foi criado para tal efeito, da forma que se encontra em seguida:

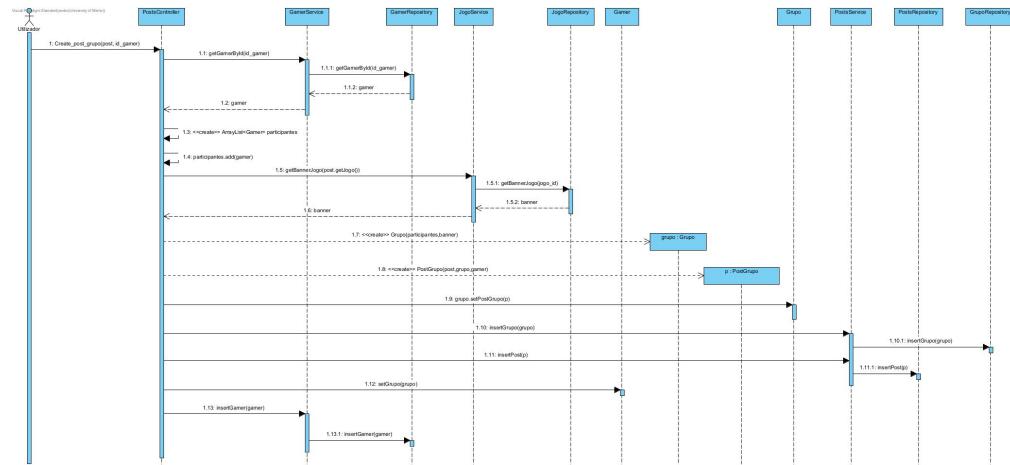


Figura 12.5: Diagrama de Sequência - Criar *post* grupo

12.13 Criar post multimédia

Na plataforma GameHub para além de *posts* de grupo, existem também *posts* multimédia em que os jogadores da plataforma podem partilhar, jogadas, informações, entre outros. Para usufruir desta funcionalidade foi necessário a criação do método seguinte:

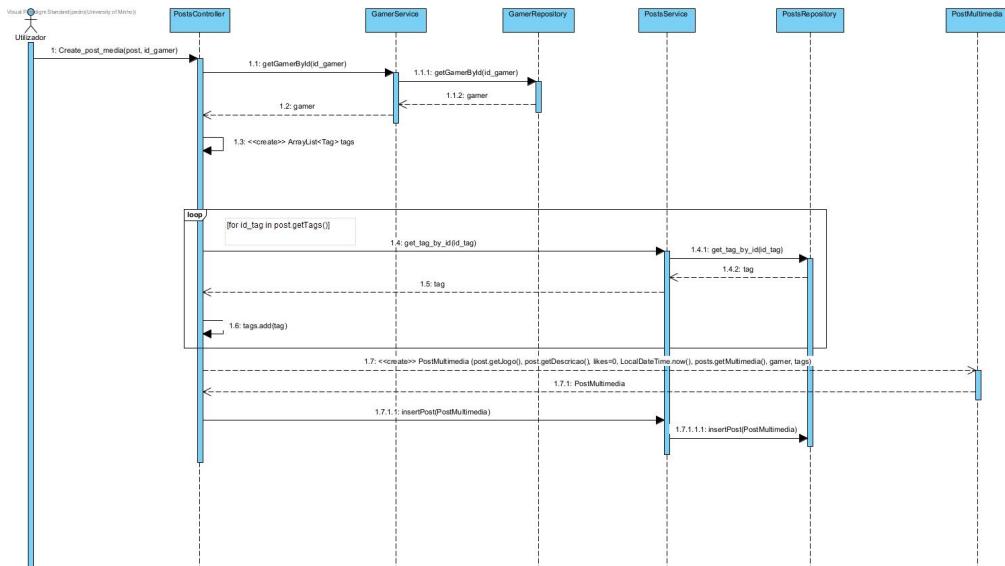


Figura 12.6: Diagrama de Sequência - Criar *post* multimédia

12.14 Avaliar *post*

De modo aos utilizadores poderem interagir entre si, assim como ganhar alguns torneios, foi necessário a adição de avaliar *posts* multimédia. Assim os utilizadores podem dar um *like*, um *dislike*, ou manterem-se neutros perante os *posts* que veem. Este método foi realizado através de verificar se o utilizador que irá avaliar já possui *like/dislike* e proceder à adição/remoção de *likes* no *post*, consoante a escolha do utilizador.

13 Vista dos dados

A aplicação GameHub revolve imenso à volta dos dados que irão ser persistidos. De tal forma surgiu o seguinte modelo lógico que é representativo das tabelas existentes na base de dados assim como as relações que estas estabelecem entre si.

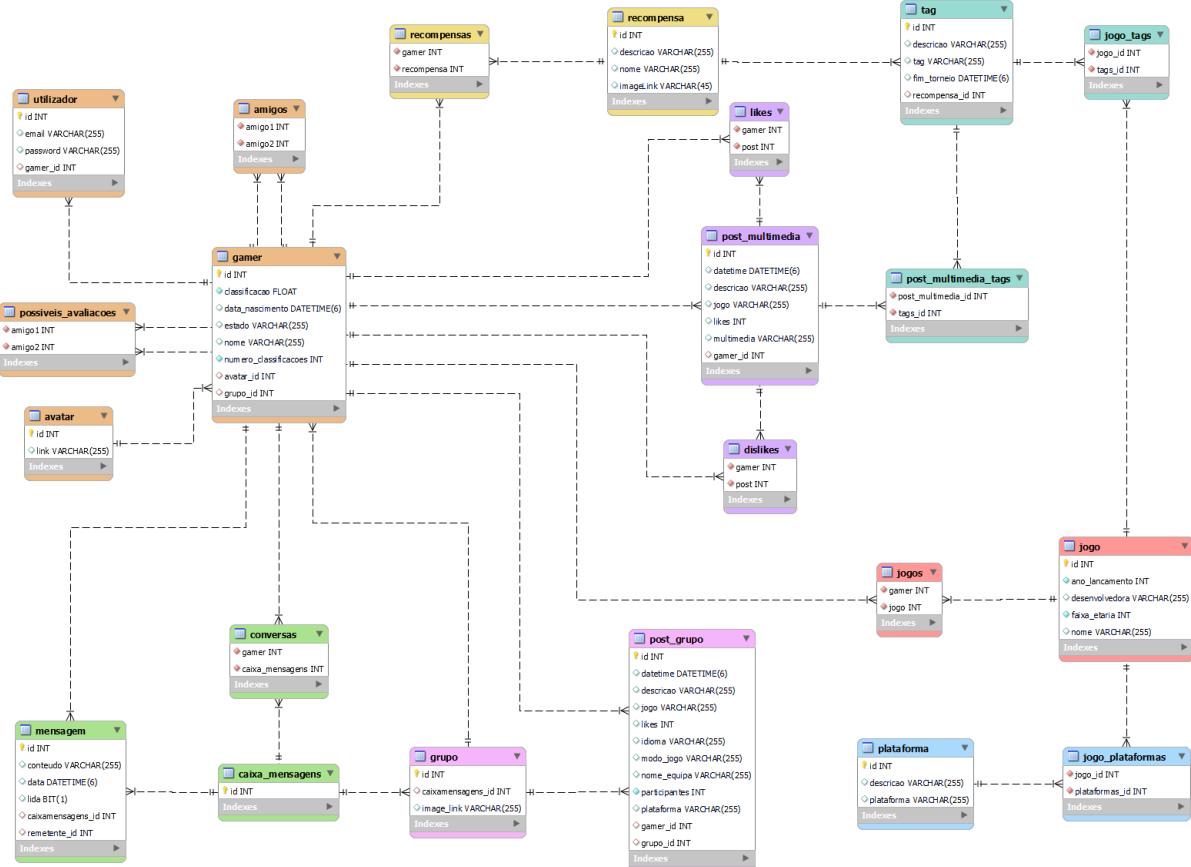


Figura 13.1: Diagrama lógico

13.1 Dicionário de dados

Ao analisar o diagrama lógico da base de dados apresentado anteriormente (figura 13.1) é possível notar a existência de 22 tabelas.

Algumas das tabelas existentes são provenientes diretamente das classes JAVA criadas, apresentando desta forma os atributos e relações necessárias para suportar o diagrama de classes já apresentado. Porém, as nove tabelas existentes a mais relativamente ao número de classes são provenientes da existência de relações de muitos para muitos.

Na tabela seguinte, demonstra-se o dicionário de dados da base de dados criada. Esta tabela apresenta de forma resumida os atributos de cada tabela, demonstrando ainda exemplos de cada atributo, bem como os seus tipos de dados e tamanho ocupado em disco (medido em bytes). Por fim, também se demonstra uma estimativa para o espaço ocupado por cada entrada numa determinada tabela.

Para calcular o tamanho ocupado por atributos do tipo *varchar* recorreu-se a uma estimativa, uma vez que o seu tamanho pode variar consoante o valor registado. Desta forma, analisou-se os valores mais comuns para cada campo e calculou-se a média de espaço ocupado.

Entidade	Atributos	Tipo de Dados	Descrição	Exemplo	Tamanho por Atributo (byte)	Tamanho por Entidade (byte)
amigos	amigo1	int	Identificador do gamer que tem um amigo	1	4	8
	amigo2	int	Identificador do gamer que é amigo de um outro	2	4	
avatar	id	int	Identificador único	1	4	261
	link	varchar(255)	Endereço para a imagem do avatar	https://roboshash.org/pv.png	257 (*)	
caixa_mensagens	id	int	Identificador único	1	4	4
conversas	gamer	int	Identificador único do gamer a que a caixa de mensagens pertence	1	4	8
	caixa_mensagens	int	Identificador único da caixa de mensagens	21	4	
dislikes	gamer	int	Identificador único do gamer que deu o dislike	1	4	8
	post	int	Identificador único da publicação que recebeu o dislike	25	4	
gamer	id	int	Identificador único	1	4	71
	classificacao	float	Valor que identifica a classificação do gamer	4,63	8	
	data_nascimento	datetime	Data de nascimento do gamer	10/09/2002 11:38	8	
	estado	varchar(255)	Estado atual do gamer	ONLINE	9 (*)	
	nome	varchar(255)	Nickname escolhido pelo gamer	brinda	30 (*)	
	numero_classificacoes	int	Número de classificações realizadas no gamer	3	4	
	avatar_id	int	Identificador único do avatar utilizado pelo gamer	1	4	
	grupo_id	int	Identificador único do grupo a que o gamer pertence	21	4	
grupo	id	int	Identificador único	21	4	266
	image_link	varchar(255)	Endereço para a imagem que representa o grupo	https://i.imgur.com/WSoqYkB.png	257 (*)	
	caixamensagens_id	int	Identificador único da caixa de mensagens utilizada pelo grupo	22	4	
jogo	id	int	Identificador único	1	4	331
	ano_lancamento	int	Ano de lançamento do jogo	2018	4	
	banner_link	varchar(255)	Endereço para a imagem do banner do jogo	https://i.imgur.com/4s7AxMo.png	257 (*)	
	desenvolvedora	varchar(255)	Desenvolvedor do jogo	InnerSloth	32 (*)	
	falha_etaria	int	Idade mínima para jogar o jogo	10	4	
jogo_plataformas	nome	varchar(255)	Nome do jogo	Among Us	30 (*)	8
	jogo_id	int	Identificador único do jogo	1	4	
jogo_tags	plataformas_id	int	Identificador único da plataforma	1	4	8
	jogo_id	int	Identificador único do jogo	1	4	
jogos	tags_id	int	Identificador único da tag	1	4	8
	gamer	int	Identificador único do gamer	1	4	
likes	jogo	int	Identificador único do jogo	1	4	8
	post	int	Identificador único da publicação que recebeu o like	8	4	
mensagem	id	int	Identificador único	66	4	282
	conteudo	varchar(255)	Conteúdo da mensagem enviada	"Olá amigo"	257 (*)	
	data	datetime	Data e hora em que a mensagem foi enviada	04/06/2021 06:36	8	
	lida	bit(1)	Boolean com o estado de leitura da mensagem	0	1	
	caixamensagens_id	int	Identificador da caixa de mensagens a que a mensagem pertence	21	4	
	remetente_id	int	Identificador único do gamer que enviou a mensagem	1	4	
	id	int	Identificador único	1	4	
plataforma	descricao	varchar(255)	Breve descrição sobre a plataforma	Microsoft Windows	257 (*)	276
	plataforma	varchar(255)	Nome da plataforma	PC	15 (*)	
post_grupo	id	int	Identificador único	1	4	460
	datetime	datetime	Data e hora de criação da publicação	04/06/2021 06:36	8	
	descricao	varchar(255)	Descrição inserida na publicação	"Quero amigos para jogar"	257 (*)	
	jogo	varchar(255)	Jogo a que a publicação se refere	Among Us	50 (*)	
	likes	int	Número de likes que a publicação recebeu	0	4	
	idioma	varchar(255)	Idioma escolhido pelo criador da publicação, para os diálogos do grupo	PT	50 (*)	
	modo_jogo	varchar(255)	Modo de jogo pretendido pelo gamer que realizou a publicação	PVE	10 (*)	
	nome_equipa	varchar(255)	Nome do grupo criada pela publicação	A Team	50 (*)	
	participantes	int	Número máximo de participantes	4	4	
	plataforma	varchar(255)	Plataforma em que se pretende jogar	1	15 (*)	
post_multimedia	gamer_id	int	Identificador do gamer que realizou a publicação	1	4	584
	grupo_id	int	Identificador do grupo criado	25	4	
	id	int	Identificador único	6	4	
	datetime	datetime	Data e hora de criação da publicação	04/06/2021 06:36	8	
	descricao	varchar(255)	Descrição inserida na publicação	"Que jogada!"	257 (*)	
recompensa	jogo	varchar(255)	Jogo a que a publicação se refere	GTA-V	50 (*)	315
	likes	int	Número de likes que a publicação recebeu	879	4	
	multimedia	varchar(255)	Endereço para a multimedia anexada na publicação	https://i.imgur.com/4s7AxMo.png	257 (*)	
post_multimedia_tags	gamer_id	int	Identificador do gamer que realizou a publicação	1	4	8
	post_multimedia_id	int	Identificador da publicação	1	4	
recompensas	tags_id	int	Identificador da tag usada na publicação	54	4	8
	id	int	Identificador único	5	4	
	descricao	varchar(255)	Descrição sobre a recompensa	Melhor jogada da semana 4	257 (*)	
tag	image_link	varchar(255)	Endereço para a imagem do troféu ganho	https://i.imgur.com/4s7AxMo.png	4	288
	nome	varchar(255)	Nome da recompensa	POW-4	50 (*)	
recompensas	gamer	int	Identificador do gamer que ganhou uma recompensa	1	4	8
	recompensa	int	Identificador da recompensa ganha	2	4	
tag	id	int	Identificador único	3	4	288
	descricao	varchar(255)	Descrição da tag	Multiplayer	257 (*)	
	fin_torneio	datetime	Data para o fim do torneio que a tag se refere	NULL	8	
	tag	varchar(255)	Nome da tag	MP	15 (*)	
utilizador	recompensa_id	int	Identificador da recompensa ganha	NULL	4	138
	id	int	Identificador único	8	4	
	email	varchar(255)	Email do utilizador	mail@mail.pt	100 (*)	
possiveis_avaliacoes	password	varchar(255)	Password utilizada pelo utilizador	AAAAA12345678	30 (*)	8
	gamer_id	int	Identificador dos dados do utilizador	5	4	
	amigo1	int	Identificador do gamer que pode avaliar	1	4	
	amigo2	int	Identificador do gamer que pode ser avaliado	2	4	

* Espaço estimado para o valor comum dos atributos

3347

Figura 13.2: Dicionário de dados

13.2 Estimativa do espaço em disco da base de dados e taxa de crescimento anual

Com base nas estimativas de espaço ocupado apresentadas na tabela 13.2 previu-se o espaço ocupado na Base de Dados após um ano.

Primeiramente, admitiu-se que os jogos suportados pelo sistema não iria sofrer grandes adições. Apesar de todos os anos serem lançados inúmeros jogos, no sistema apenas são adotados os jogos *multiplayer* mais famosos, tendo por isso uma baixa taxa de crescimento na base de dados, podendo-se descartar desta forma a tabela jogo deste estudo. Com isto a relação de jogo com tag também não irá sofrer grandes alterações.

Para além disto, admite-se também que as plataformas existentes não variam e por isso a relação de muitos para muitos entre um jogo e uma plataforma não irá alterar.

Por fim, o número de tag's existentes não deverá sofrer um aumento suficiente para ser considerado. Esta característica dá-se pelo facto de que as tag's são eliminadas sempre que um torneio acaba. Assim, o inicio de novos torneios, que implicaria o aumento do número de tag's, é atenuado pelo termino de torneios a decorrer.

Ao considerar estas entidades como estáticas é necessário estimar a taxa de crescimento das restantes tabelas. Assim, espera-se que:

- No final do primeiro ano sejam registados 50 mil novos utilizadores;
- Cada utilizador realize 2 publicações para a criação de grupo por semana;
- Cada utilizador realize 2 publicações de multimédia por dia;
- Em média cada grupo contém cinco pessoas. Assumindo que todos os *gamer's* estarão incorporados num determinado grupo;
- Cada *gamer* muda de avatar todos os meses;
- Em média cada *gamer* apresenta interesse em 6 jogos;
- Três a cada quatro publicações de multimédia tem associadas uma tag;
- Todas as semanas irá existir um torneio de jogada da semana. Para além disto irá existir um torneio para a jogada do mês.
- Todos os dias espera-se que cada utilizador envie 8 mensagens em média.
- Cada utilizador dá, em média, 10 *like/dislike* nos *posts* multimédia, por dia.
- Cada utilizador vai avaliando os *gamers* com quem contacta ao logo da sua experiência no sistema.

Com base nas estimativas acima apresentadas estima-se que a base de dados, no final do primeiro ano, ocupe cerca de 63.38 Giga Bytes.

Desta forma, torna-se necessário eliminar dados mais antigos que não sejam relevantes para a plataforma. Um exemplo claro, são as mensagens, não sendo o foco da aplicação a comunicação via *chat* a longo prazo, as mensagens antigos podem facilmente ser eliminadas sem qualquer dano para os utilizadores.

14 Vista de deployment

Após o termo da fase de desenvolvimento do sistema foi necessário considerar o *deployment* do mesmo.

Para providenciar um *deployment* simples decidiu-se recorrer à criação de *containers* Docker. A utilização desta plataforma permite facilitar a criação de ambientes isolados, ao recorrer à tecnologia de *containers*. Assim, os *containers* criados podem ser considerados micro-ambientes, que permitem isolar os ambientes criados do sistema operativo em que se encontram.

A utilização de *containers docker* torna-se mais eficiente do que a criação de máquinas virtuais, na medida em que os ambientes criados são considerados *Lightweight*, ou seja não necessitam de simular um sistema operativo distinto, tornando-se assim mais leves. Para além disto, a sua utilização permite obter um maior isolamento (os vários *containers* criados estão isolados entre si e do sistema operativo *host*, em termos de segurança, desempenho e falhas), portabilidade (torna-se simples migrar os *containers* para diferentes ambientes) e um melhor *deployment* (para adicionar novos serviços apenas é necessário criar novos *containers*). Para além disto, a utilização deste tipo de *deployment* permite a instalação dos componentes desenvolvidos em vários sistemas operativos distintos, sem a necessidade de configurar os componentes de forma distinta entre sistemas.

Por fim, um outro aspecto importante a considerar na fase de *deployment* é a sua automatização. Sendo o processo de *deployment*, um processo repetitivo e aborrecido, optou-se por possibilitar a sua automatização, recorrendo a *scripts bash* e à ferramenta *ansible*. Com a utilização destas ferramentas foi possível desenvolver um conjunto de diretrizes que serão executadas de forma automática nas máquinas remotas indicadas.

Assim, para realizar o *deployment* dos vários componentes desenvolvidos deve-se configurar o *script bash* com os endereços das máquinas disponíveis para instalar cada componente, seguido da sua execução.

14.1 Infraestrutura

Considerando os vários componentes desenvolvidos para o sistema e um nível de *performance* adequado, idealizou-se o seguinte diagrama de *deployment*:

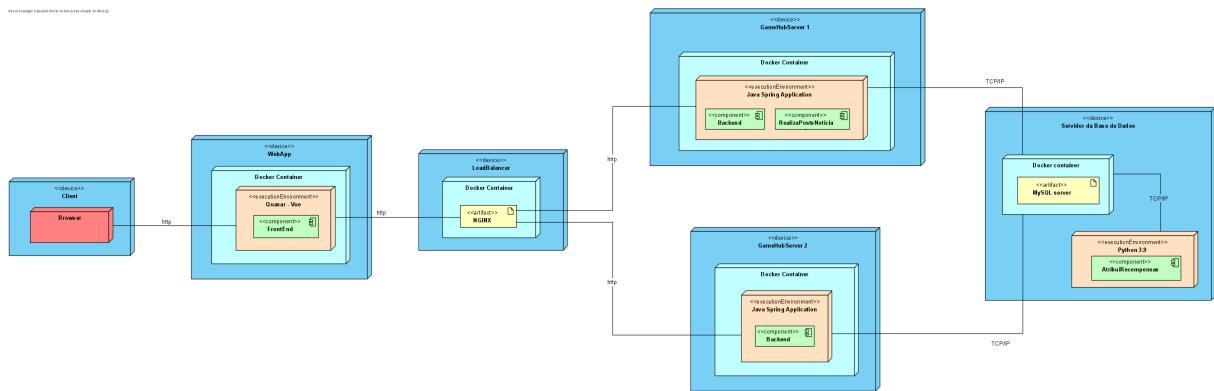


Figura 14.1: Diagrama de *deployment*

Ao analisar o diagrama de *deployment* apresentado é possível identificar a existência de cinco servidores fundamentais - servidor da *WebApp*, servidor do *Load Balancer*, servidores do *GameHub* e o servidor da base de dados. Também é possível verificar a utilização de servidores separados para diferentes componentes, bem como a replicação de componentes em servidores diferentes. Esta separação e introdução de redundância permite tornar o sistema mais robusto, e mais tolerante a falhas, garantindo uma maior disponibilidade do sistema.

Assim, com a arquitetura de *deployment* sugerida é possível adicionar novos servidores durante a execução do serviço, aumentando assim a capacidade do sistema para tratar de novos pedidos.

Por fim, realizou-se o *deployment* de cada componente considerando-se para cada servidor a utilização da seguinte arquitetura:

- **CPU:** CPU Intel Skylake;
- **CORES:** 6 vCPU;
- **RAM:** 22 GB;
- **DISCO:** SSD de 500 GB.

14.1.1 Servidor da *WebApp*

Durante o desenvolvimento do sistema, optou-se por desenvolver uma aplicação híbrida. Assim, para disponibilizar a componente da *frontend* aos utilizadores, recorreu-se a um servidor diferente dos restantes, como se pode observar no diagrama de *deployment* presente na figura 14.1.

Ao observar o diagrama, também é possível perceber que esta componente está a ser executada num *container docker*. Contudo, como este componente não tem uma imagem *docker* oficial foi necessário criar o seguinte *dockerfile*:

```
# develop stage
FROM node:latest as develop-stage
WORKDIR /app
COPY package*.json .
RUN yarn global add @quasar/cli
COPY .

# build stage
FROM develop-stage as build-stage
RUN yarn
RUN quasar build

# production stage
FROM nginx:latest as production-stage
COPY --from=build-stage /app/dist/spa /usr/share/nginx/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

Figura 14.2: *Dockerfile* criado para a implementação da componente *frontend*

Neste ficheiro, foi necessário ter em atenção a utilização da *framework vue*. Desta forma, foi necessário instalar todas as dependências do componente dentro do *container* de modo a disponibilizar a interface aos utilizadores.

Na última fase da criação do *container*, recorre-se ao *NGINX* para disponibilizar via HTTP a interface desenvolvida.

14.1.2 Servidor do *Load Balancer*

Este servidor é constituído por um *reverse proxy* capaz de receber pedidos HTTP e redirecioná-los para um conjunto de servidores diferentes. Assim, optou-se por utilizar o *NGINX* como *reverse proxy*.

O *reverse proxy* do *NGINX* disponibiliza uma imagem oficial *docker*. Desta forma, para instalar este componente é simplesmente necessário configurar o ficheiro de configuração do programa e associá-lo ao *docker container* criado. Este ficheiro de configurações é por sua vez gerado a partir de uma *template* pelo *ansible*, e configura no serviço um servidor *http* que redireciona as sessões para outras instâncias do serviço, através de um algoritmo *round robin*, escolhido pelo grupo.

14.1.3 Servidor do *GameHub*

O servidor do *GameHub* é constituído pelo componente da *backend* desenvolvida. Esta componente serve de apoio à interface desenvolvida, proporcionado as funcionalidades do sistema. Para além disto, é possível identificar os casos em que se pretende executar a componente, que cria as publicações de notícias, em conjunto.

De modo a implementar esta componente, recorreu-se a *containers docker*, porém, uma vez que não existe uma imagem oficial foi necessário criá-la através da utilização de um *dockerfile*.

```
FROM openjdk:18
ENV TIPOSERVER=0
ADD GameHub-0.0.1-SNAPSHOT.jar GameHub.jar
EXPOSE 8080
ENTRYPOINT [ "sh", "-c", "java $JAVA_OPTS-Djava.security.egd=file:/dev/.urandom -jar /GameHub.jar --deploymentNumber=${TIPOSERVER}" ]
```

Figura 14.3: *Dockerfile* criado para a implementação da componente *backend*

De modo a iniciar a execução do *container* recorre-se ao ficheiro jar da aplicação. Este ficheiro, contém todas as dependências e classes criadas que suportam a execução da aplicação. Contudo, de modo a executar o sistema é necessário identificar o tipo do componente. A identificação do tipo de servidor é realizada através do argumento **–deploymentNumber=X**, onde o X toma o valor de:

- **0:** Se em conjunto com o componente da *backend* deve ser executado também o componente criado para a realização de publicações de notícias;
- **1:** Se o componente *backend* deve ser executado de forma isolada.

Assim, para suportar este argumento no mesmo *dockerfile* recorreu-se às variáveis de ambiente, que permitem definir o seu valor durante a inicialização do *container*.

Desta forma, é possível executar dois servidores do *GameHub* em máquinas distintas, identificando a máquina em que deve ser executada a criação de publicações de notícias, como se observa no diagrama de *deployment* presente na figura 14.1.

14.1.4 Servidor da base de dados

Este servidor apresenta na sua constituição um sistema de gestão de base de dados *MySQL* a executar. Este sistema de gestão de base de dados, está a executar num *container docker*, sendo por isso na sua instalação necessário executar a imagem oficial existente. Para além disto, tira-se partido das variáveis presentes na imagem, para configura-la durante a sua inicialização.

Neste servidor, instalou-se ainda o componente responsável por atribuir recompensas aos utilizadores. Para a sua instalação, recorreu-se simplesmente à execução (em background) do *script python* desenvolvido.

14.2 Adicionar servidores do *GameHub* em *runtime*

Por fim, uma funcionalidade importante é a capacidade de adicionar servidores durante a execução do sistema. A adição de servidores deve incidir-se nos servidores do *GameHub*, uma vez que estes servidores são os que se encontram mais sobrecarregados.

Assim, ao analisar esta necessidade criou-se um *playbook ansible* que permite adicionar novos servidores do *GameHub* durante a execução do sistema. Para além de criar os novos servidores, este *playbook* deve reconfigurar o *reverse proxy*, de modo a permitir que este redistribua a carga por todos os servidores existentes.

Este processo deverá ser uma vez mais executado de forma automática, permitindo que a configuração dos servidores seja executada de forma simples.

15 Análise de escalabilidade

Em Engenharia de Software, tal como noutras áreas, antes de lançar um produto para o mercado, é importante garantir que, o mesmo, se irá comportar como esperado, mesmo com um elevado número de utilizadores concorrentes. É necessário verificar se, com o aumento de carga, a qualidade da experiência de utilizador se mantém elevada. Para tal, é necessário avaliar um conjunto de métricas:

- **Confiança:** O sistema é resistente a falhas durante a execução, ou seja, se o sistema não é interrompido em caso de erro;
- **Funcionalidade:** O sistema possui todas as funcionalidades e tem um comportamento adequado aos requisitos definidos.
- **Performance:** O sistema possui tempos de resposta aceitáveis, tanto em pequena como em grande escala.

15.1 JMeter

Para realizar estes testes de escalabilidade, o grupo decidiu utilizar a ferramenta *Apache JMeter* que permite medir a performance e testar a carga de aplicações web estáticas ou dinâmicas. Pode ser utilizada para simular uma carga pesada num servidor (ou grupo de servidores), rede ou objeto para testar a sua força ou analisar a performance sobre diferentes tipos de carga.

Assim, procurámos realizar testes de escalabilidade que sejam, o máximo possível, representativos da realidade, ou seja, realizar testes que reproduzam o comportamento de utilizadores de redes sociais em simultâneo. Após a observação do comportamento de utilizadores de outras plataformas, decidimos simular as seguintes ações:

1. Aceder à página de login (Pedido HTTP GET)
2. Realizar Login (Pedido HTTP POST)
3. Aceder à página inicial (Conjunto de pedidos HTTP)
 - 3.1. Carregar página inicial (Pedido HTTP GET)
 - 3.2. Obter o perfil de utilizador (Pedido HTTP GET)
 - 3.3. Obter jogos mais populares (Pedido HTTP GET)
 - 3.4. Obter jogos seguidos (Pedido HTTP GET)
 - 3.5. Obter grupo do utilizador (Pedido HTTP GET)
 - 3.6. Obter publicações mais populares (Pedido HTTP GET)
4. Aceder à lista dos jogos (Conjunto de pedidos HTTP)
 - 4.1. Carregar a página de seguir jogos (Pedido HTTP GET)
 - 4.2. Obter jogos por seguir (Pedido HTTP GET)
5. Seguir jogos (Pedido HTTP POST)
6. Aceder à página de criar publicação (Pedido HTTP GET)
7. Criar publicação (Pedido HTTP POST)
8. Aceder a um feed de um jogo (Conjunto de pedidos HTTP)
 - 8.1. Carregar página do feed desse jogo (Pedido HTTP GET)
 - 8.2. Obter publicações desse jogo (Pedido HTTP GET)

9. Dar like numa publicação (Pedido HTTP POST)
10. Dar dislike noutra publicação (Pedido HTTP POST)
11. Realizar logout (Pedido HTTP POST)

Esta carga de trabalho traduz-se, no *Apache JMeter*:

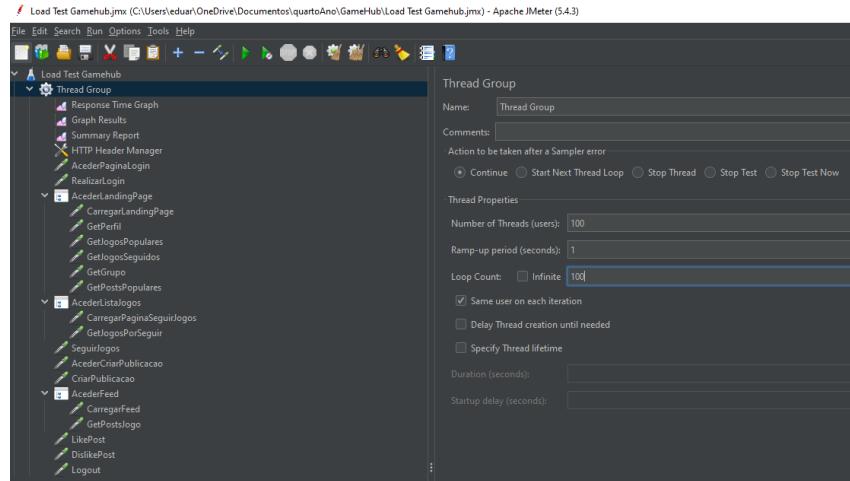


Figura 15.1: Workload no Apache JMeter

Neste exemplo, estão a ser simulados 100 utilizadores com o comportamento especificado acima, com 100 iterações cada um, de modo a estender o tempo do teste de carga. No entanto, serão realizados testes para 100, 200, 500, 1000 e 1500 clientes (Threads).

15.1.1 100 Clientes

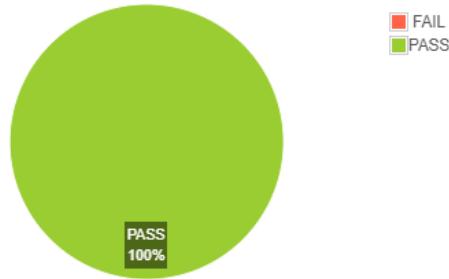


Figura 15.2: Percentagem de erro dos pedidos com 100 Clientes

Neste gráfico circular, podemos verificar a percentagem de pedidos que foram realizados com sucesso e a percentagem de pedidos com erro. Como podemos verificar, para 100 clientes, 100% dos pedidos foram realizados com sucesso.

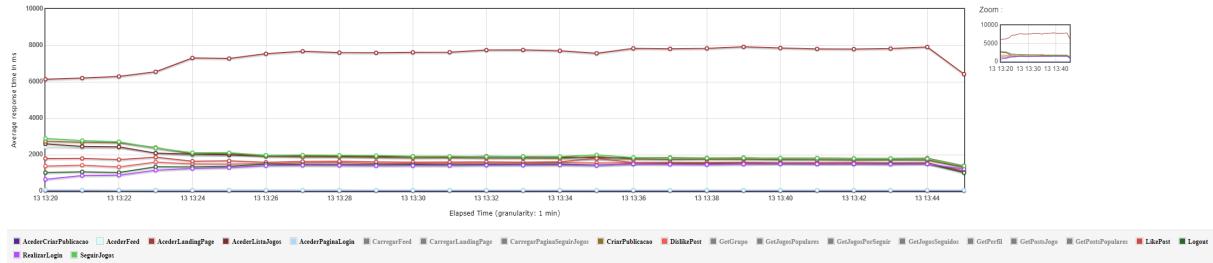


Figura 15.3: Tempo de resposta médio com 100 Clientes

Neste segundo gráfico, podemos analisar o tempo de resposta médio para cada tipo de pedido, ao longo do tempo do teste. Podemos verificar que a ação com pior performance é o acesso à página inicial (landing page). Este resultado é expetável, uma vez que o acesso à página inicial é a primeira ação que cada cliente faz após o *login*, o que implica que sejam feitos vários pedidos adicionais para apresentar toda a informação necessária ao utilizador, tal como os jogos seguidos, amigos, grupo e jogos populares. Uma vez que estes dados ficam armazenados em cache, estes pedidos não são feitos novamente nas restantes ações, pelo que o tempo de resposta é consideravelmente menor.

Outro aspecto importante de realçar é o facto de, em geral, os tempos de resposta serem um pouco altos para o esperado com o *deployment* realizado. Após algum estudo, o grupo chegou à conclusão que estes tempos de resposta devem-se às limitações da plataforma utilizada para o *deployment* (Google Cloud Platform), nomeadamente à necessidade de alojar cada máquina em países distintos devido ao limite de *cores* que se pode alojar na mesma zona. Esta alocação de máquinas em zonas distantes umas das outras causa um *delay* considerável na execução de pedidos devido ao tempo necessário para transportar estes pedidos pela rede entre as várias máquinas. Este *delay*, como iremos observar, é ainda mais visível nos tempos de resposta em testes de larga escala.



Figura 15.4: Throughput com 100 Clientes

Neste último gráfico, podemos observar o *throughput* geral da nossa aplicação, ao longo do tempo de execução do teste realizado. Esta métrica representa a capacidade do nosso sistema de responder a pedidos em função do tempo, neste caso número de *hits* por segundo. Com 100 clientes, o nosso sistema apresentou um *throughput* de, aproximadamente, 85 *hits/sec*.

15.1.2 200 Clientes

Tendo os resultados anteriores como referência base das diferentes métricas, decidimos duplicar o número de utilizadores concorrentes para 200.

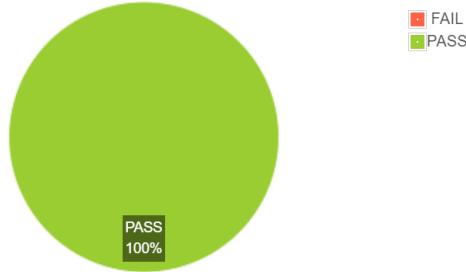


Figura 15.5: Percentagem de erro dos pedidos com 200 Clientes

É visível que, mesmo com 200 clientes, 100% dos pedidos efetuados ao sistema são realizados com sucesso.

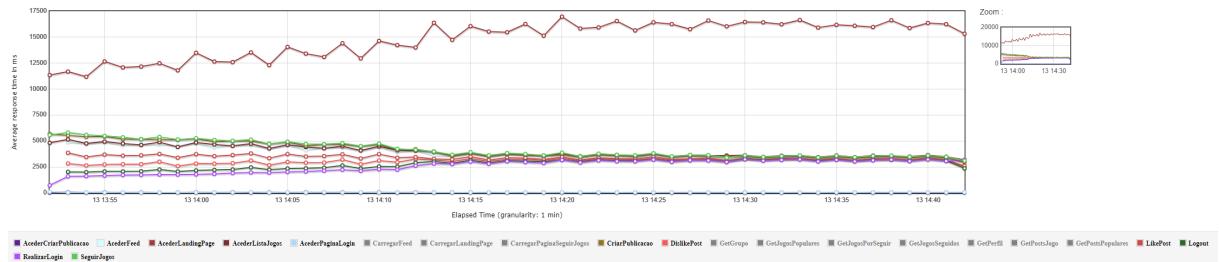


Figura 15.6: Tempo de resposta médio com 200 Clientes

Quanto ao tempo de resposta médio, é possível concluir que, como esperado, houve um maior impacto na ação mais pesada (aceder à página inicial), que sofreu uma diminuição considerável no tempo de resposta. Por outro lado, os restantes pedidos não sofreram uma grande alteração, mantendo-se perto de 2500 milisegundos. Mais uma vez, a ação de aceder à página de login é a ação mais rápida (cerca de 50 milisegundos), o que seria de se esperar, uma vez que não realiza qualquer pedido ao servidor aplicacional. Este valor também demonstra o impacto no tempo de resposta que um pedido efetuado ao servidor aplicacional tem, quando tem de ser transmitido, pela rede, a máquinas muito distantes.

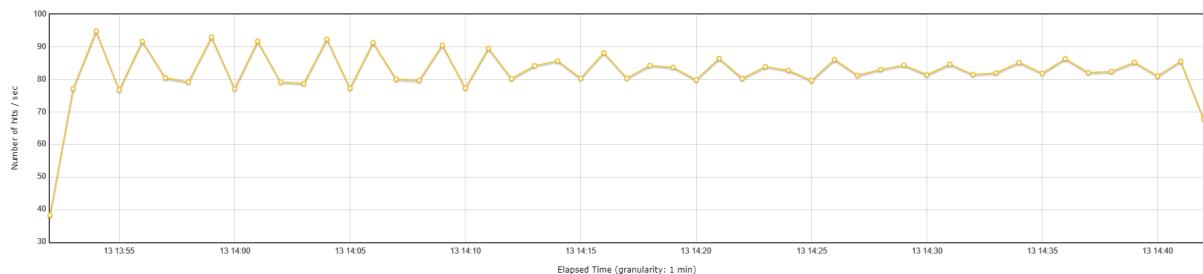


Figura 15.7: Throughput com 200 Clientes

A capacidade de resposta da nossa aplicação teve poucas mudanças, havendo apenas alguns picos de valores maiores que 90 hits/sec, mas mantendo-se perto de 85 hits/sec durante a maior parte do tempo.

15.1.3 500 Clientes

Uma vez que o aumento de 100 para 200 clientes teve um baixo impacto nas métricas calculadas, decidimos aumentar o número de clientes para 500.



Figura 15.8: Percentagem de erro dos pedidos com 500 Clientes

Desta vez ocorreu um único erro que, como estamos a trabalhar em tão grande escala, não é sequer visível na figura nem nas percentagens, pelo que não causou uma redução na taxa de sucesso.

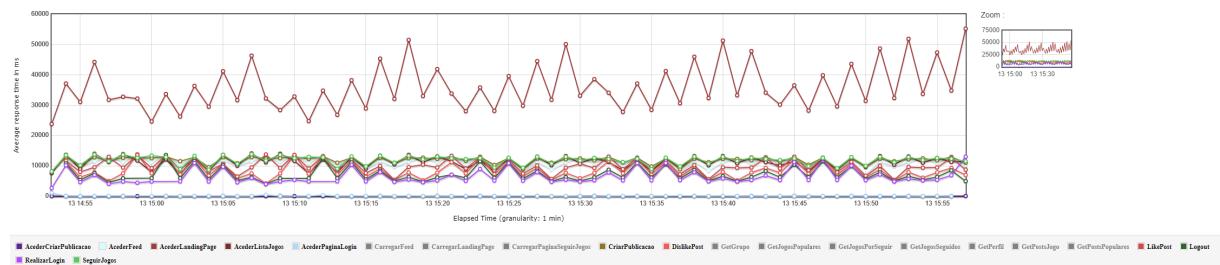


Figura 15.9: Tempo de resposta médio com 500 Clientes

Por outro lado, para as ações que implicam fazer pedidos ao servidor aplicacional, os tempos de resposta sofreram uma grande penalização na sua performance, principalmente no acesso à página inicial que, como referido anteriormente, é a ação que realiza mais pedidos.

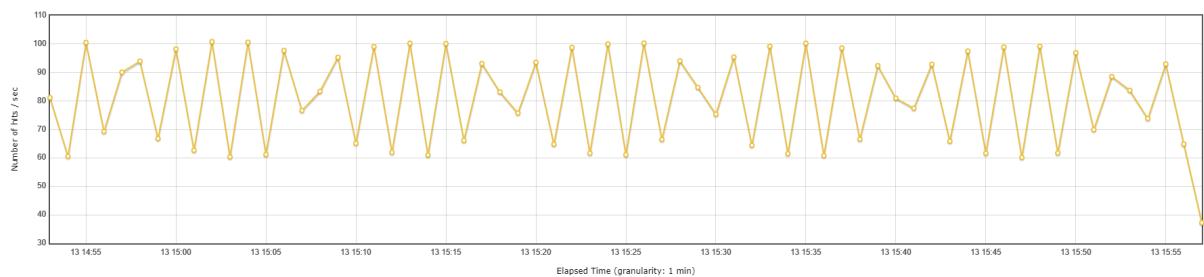


Figura 15.10: Throughput com 500 Clientes

Quanto à capacidade de resposta da aplicação, é interessante notar que este aumento de carga causou instabilidade ao *throughput*, como pode ser verificado pela existência consistente de picos neste gráfico. Existiram momentos em que o *throughput* era tão baixo como 60 *hits/sec*, e outros momentos em que era tão alto como 100 *hits/sec*.

15.1.4 1000 Clientes

Como ainda não obtivemos uma redução da taxa de sucesso dos pedidos realizados à aplicação, realizámos um teste com 1000 clientes.

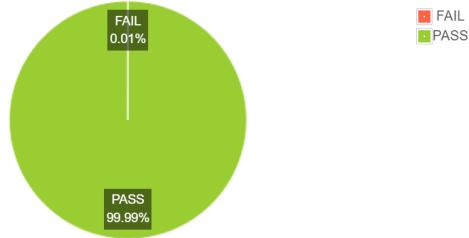


Figura 15.11: Percentagem de erro dos pedidos com 1000 Clientes

Como esperado, ocorreram alguns erros. No entanto, uma vez que essa quantidade de erros traduz-se em 0.01% do número de pedidos total, podemos concluir que a taxa de sucesso ainda é completamente aceitável.

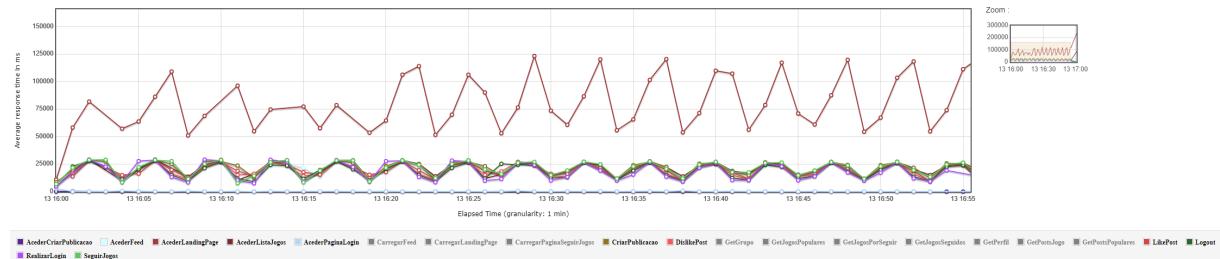


Figura 15.12: Tempo de resposta médio com 1000 Clientes

Quanto ao tempo de resposta médio houve, mais uma vez, uma degradação considerável da performance, com é visível pelos valores observados nesta figura. Começa a ser possível concluir que o tempo de resposta médio tem um crescimento quase linear com o número de clientes.

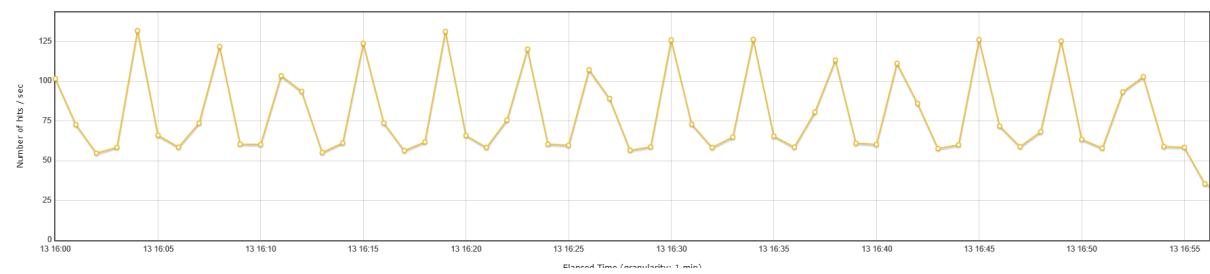


Figura 15.13: Throughput com 1000 Clientes

Tal como no teste de 1000 clientes, houve um aumento da instabilidade da capacidade de resposta da aplicação. Desta vez, houve momentos com *throughput* igual a 125 hits/sec e momentos com o *throughput* igual a 52 hits/sec.

15.1.5 1500 Clientes

De modo a procurar o limite da nossa aplicação, decidimos aumentar ainda mais o número de clientes para 1500.

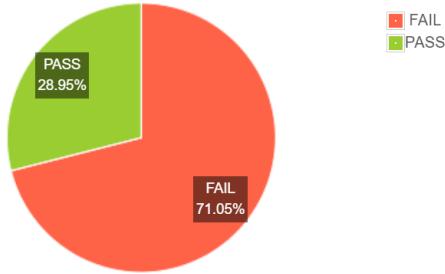


Figura 15.14: Percentagem de erro dos pedidos com 1500 Clientes

Desta vez, fomos capazes de sobrecarregar completamente o sistema, como é visível pela percentagem de erro de 71.05% .

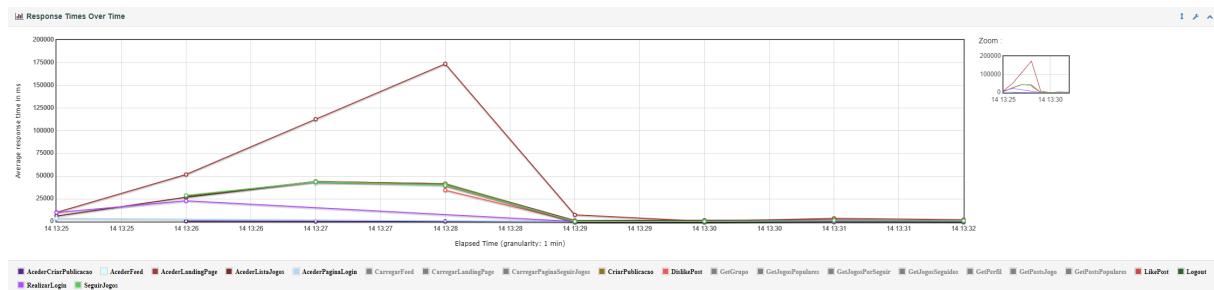


Figura 15.15: Tempo de resposta médio com 1500 Clientes

Desta vez, o gráfico do tempo médio de resposta apresenta um comportamento completamente diferente. Podemos ver que os tempos de resposta médios foram aumentando quase exponencialmente até cerca de 175000 milissegundos (a meio do teste), a partir de onde desceram para valores extremamente baixos. Acreditámos que a maior parte dos clientes (71.05%), que obtiveram erro nos seus pedidos, não continuaram a realizar pedidos ao sistema, pelo que a carga diminuiu extremamente na segunda metade do teste.

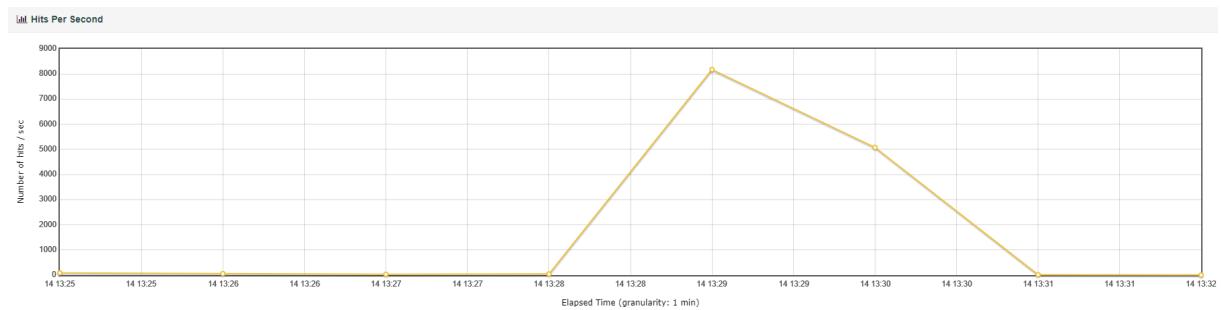


Figura 15.16: Throughput com 1500 Clientes

A curva do *throughput* observada, tal como esperado, teve um comportamento muito semelhante, mas que nos permite observar o sucedido de outra perspetiva. Podemos verificar que a aplicação não respondeu a qualquer pedido até perto do meio do teste, onde devolveu a resposta ao maior número de pedidos possível. Tal como aconteceu com o tempo de resposta, a capacidade de resposta, a partir desse momento, diminuiu a pique, o que também pode ser explicado pelo facto de 71.05% dos clientes não terem obtido resposta, causando uma diminuição enorme na carga do servidor.

16 Manual de utilização

A aplicação GameHub, como já referido anteriormente, possui dois tipos de utilizadores diferentes, os *Gamers* que são os utilizadores normais da plataforma, e os administradores, que servem para manter um bom ambiente no ceio da aplicação.

16.1 *Gamer*

Primeiramente iremos demonstrar e explicar as funcionalidades que um utilizador normal, também conhecido por *Gamer*, pode realizar durante a utilização do GameHub.

16.1.1 Registar

A primeira página que um utilizador encontra ao entrar na plataforma do Gamehub é a página de registo.

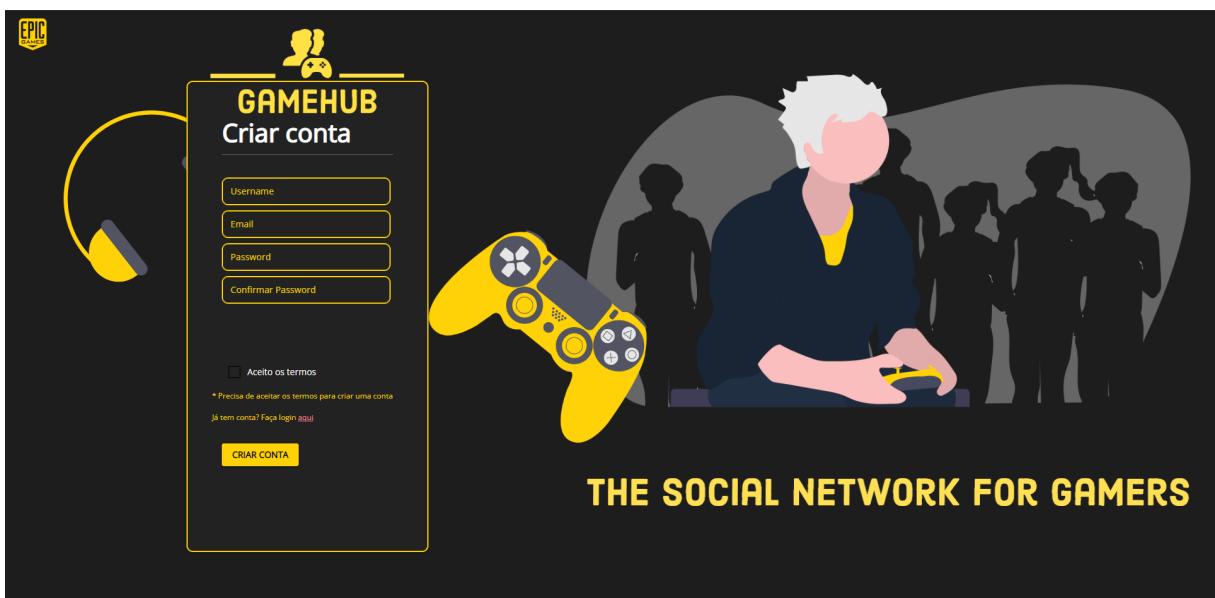


Figura 16.1: Página de registo

Para um utilizador se poder registar este precisa de inserir um *username* que não se encontre já em uso, o seu *email*, uma *password* e em seguida reintroduzi-la para ter a certeza que esta foi escrita corretamente. Em seguida é então necessário aceitar os termos, assinalando o quadrado indicado para o efeito e finalmente criar a conta.



Figura 16.2: Página de registo - preenchida

Caso o utilizador falhe a introdução de algum dos campos anteriormente referidos, irão se deparar com uma mensagem de erro a indicar o problema ocorrido.

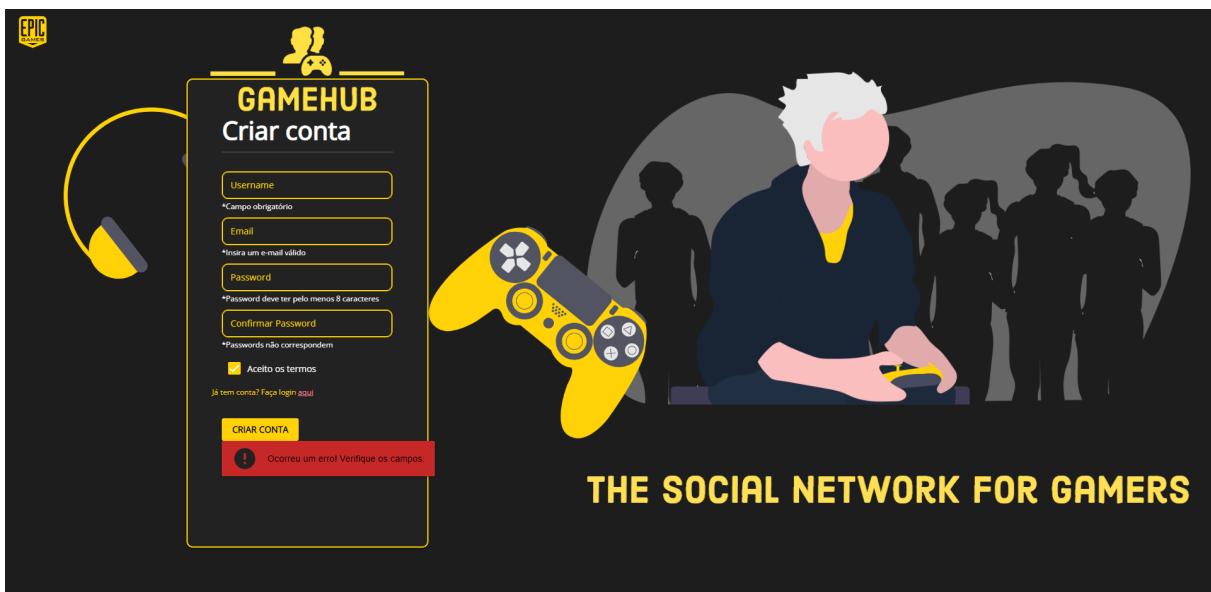


Figura 16.3: Página de registo - preenchida erradamente

16.1.2 Login

Após a realização do registo um utilizador é redirecionado para a página de *login*.



Figura 16.4: Página de *login*

Para poder entrar na aplicação o utilizador terá que introduzir o *email* e a *password* escolhidos aquando do seu registo.

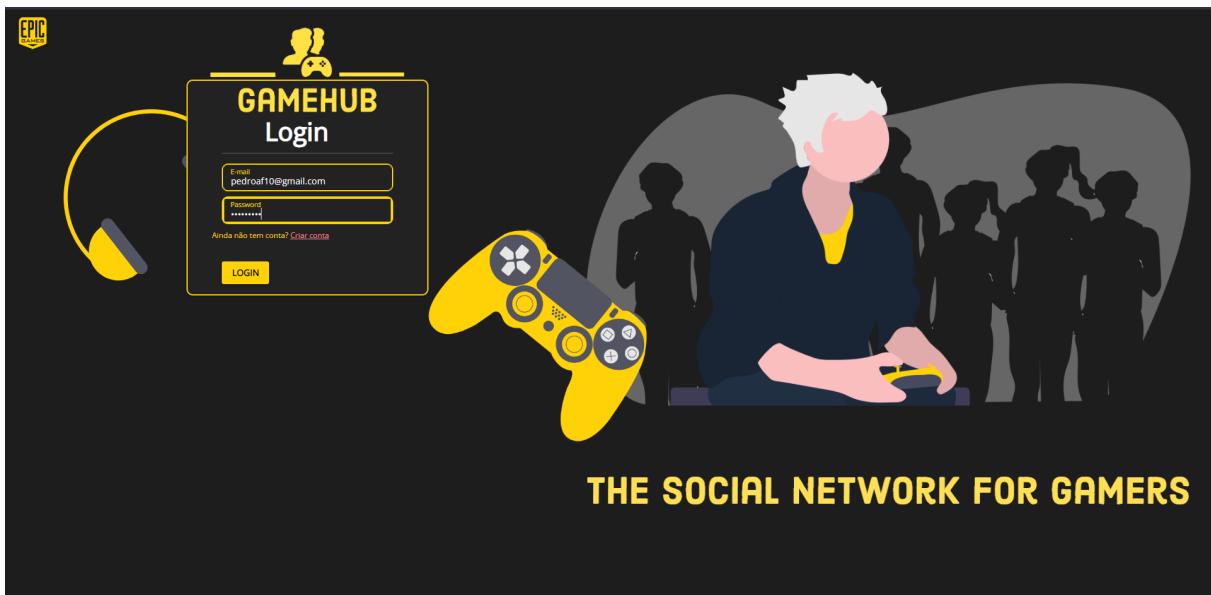


Figura 16.5: Página de *login* - preenchida

Caso o utilizador se engane na introdução dos seus dados a aplicação irá enviar-lhe uma mensagem de erro a explicar o porquê do mesmo.



Figura 16.6: Página de *login* - preenchida incorretamente

Por fim caso os dados sejam introduzidos de forma correta, o utilizador entra na aplicação, sendo encaminhado para a página principal do GameHub, onde aparecem os jogos mais populares, assim como as publicações mais recentes, mais populares.

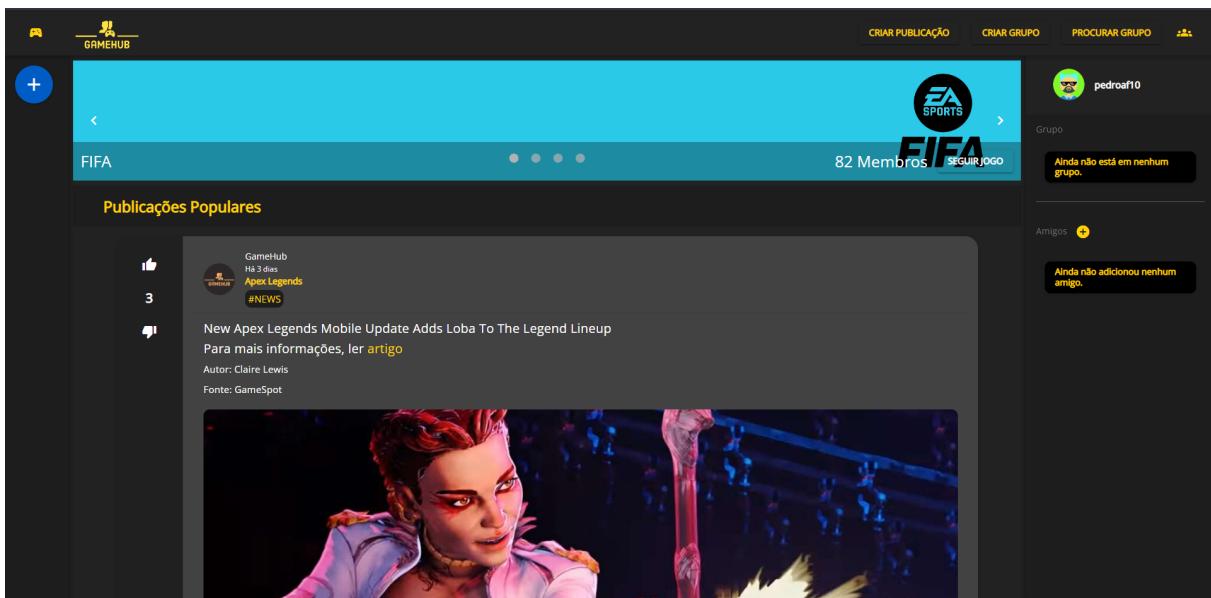


Figura 16.7: Página *home*

16.1.3 Editar perfil

É possível no GameHub, que cada utilizador customize o seu perfil após ao seu gosto, podendo alterar a sua foto de perfil, assim como o seu nome. Para tal precisa primeiramente de carregar na sua imagem de perfil, e em seguida carregar ver perfil.

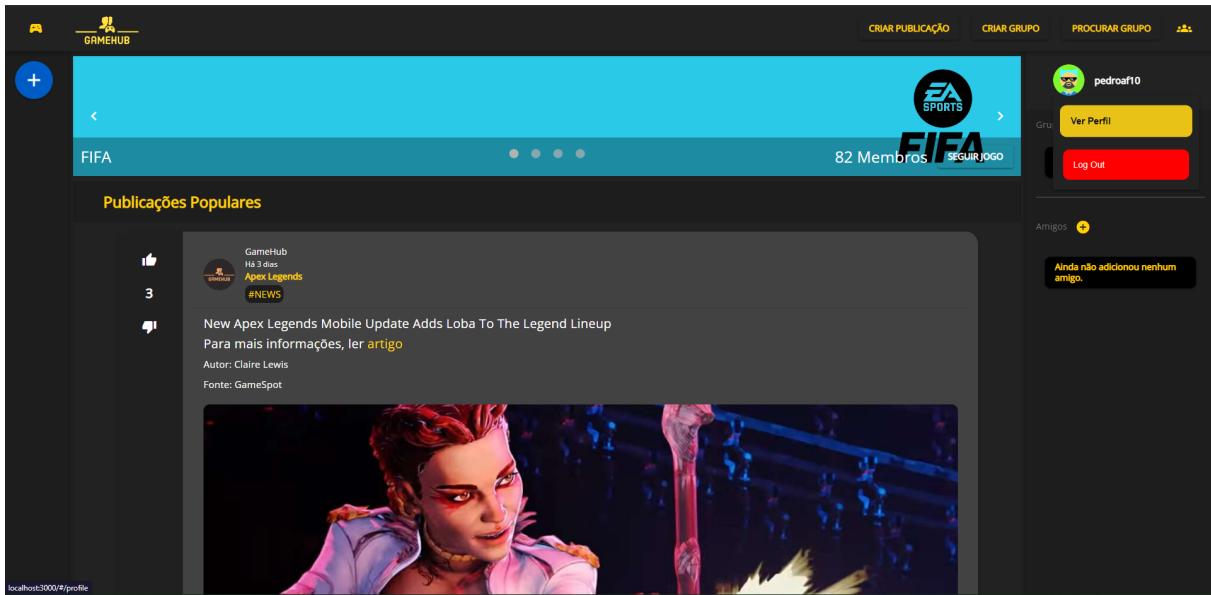


Figura 16.8: Página *home* - carregar em ver perfil

Após realizada esta ação o utilizador é redirecionado para a página do seu perfil. Aqui um utilizador pode alterar o seu nome de perfil, o seu avatar, visualizar as suas recompensas, a sua classificação e ainda visualizar e/ou apagar as publicações multimédia por si realizadas.

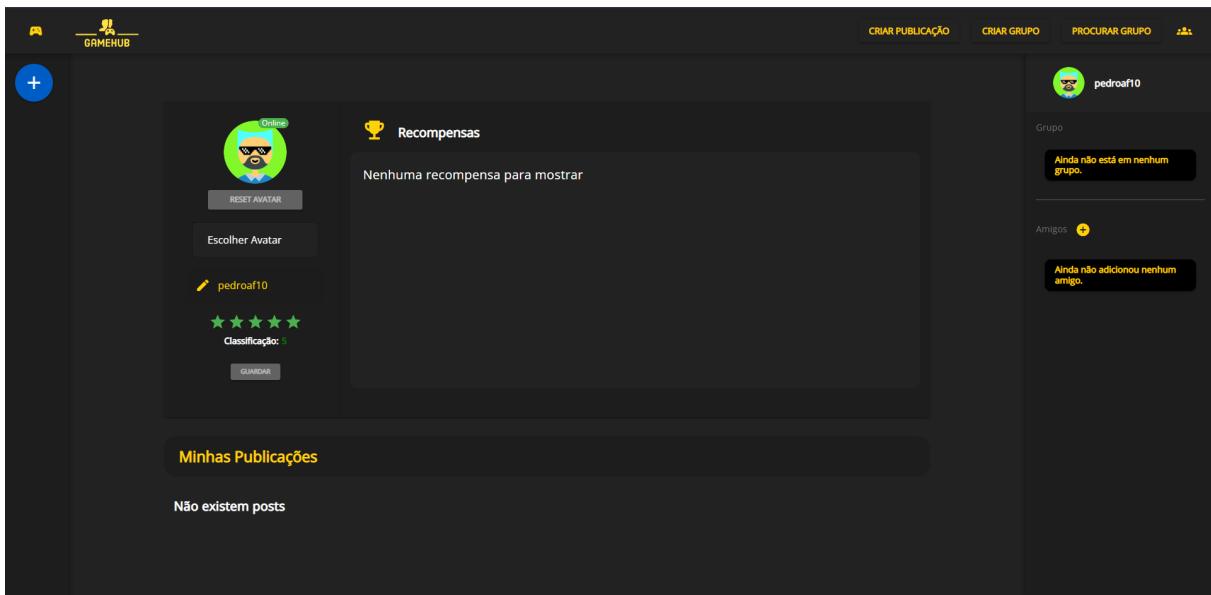


Figura 16.9: Página do perfil

Um utilizador para alterar a sua imagem de perfil apenas precisa de carregar em escolher avatar, selecionar a imagem pretendida a partir do seu dispositivo e em seguida carregar em guardar.

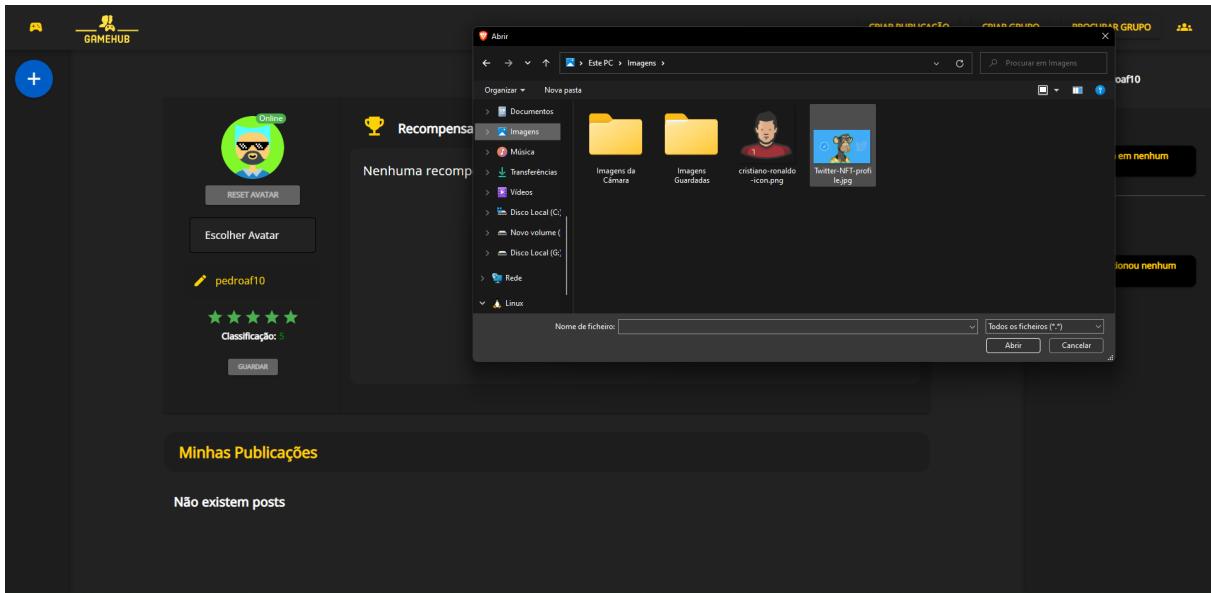


Figura 16.10: Página do perfil - adicionar imagem de perfil

Caso pretenda voltar a utilizar o avatar atribuído pelo GameHub com base no nome do utilizador escolhido, basta carregar em "Reset Avatar".

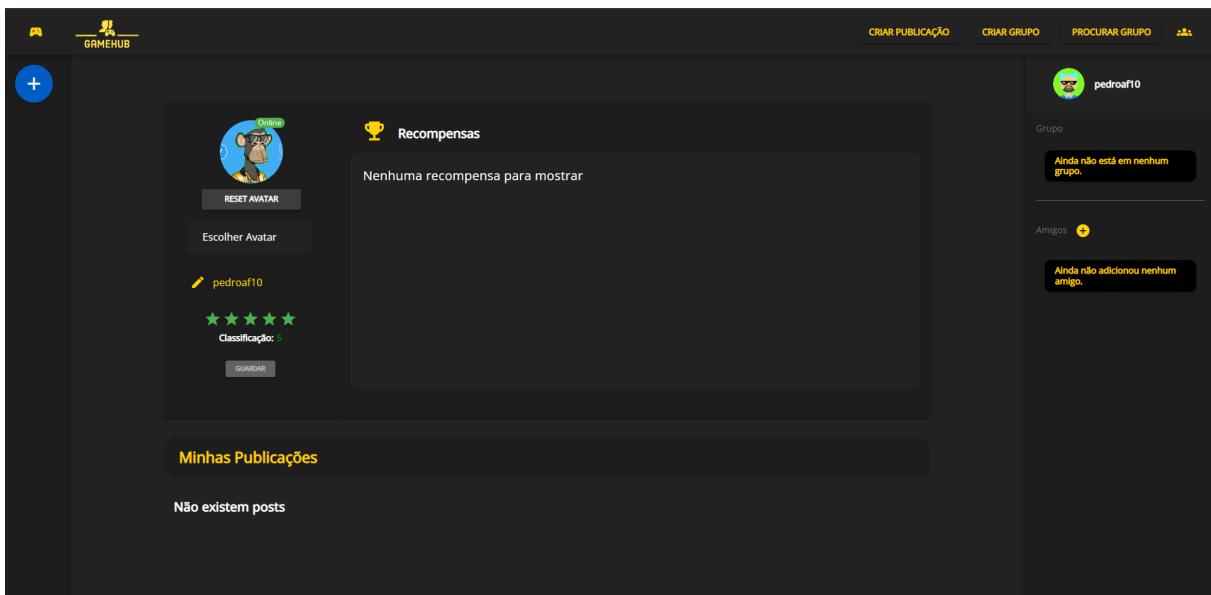


Figura 16.11: Página do perfil - dar reset à imagem nova

Já para poder alterar o seu nome de perfil o utilizador apenas precisa de introduzir o seu novo nome no campo onde aparece o seu nome de utilizador atual e em seguida carregar em guardar.

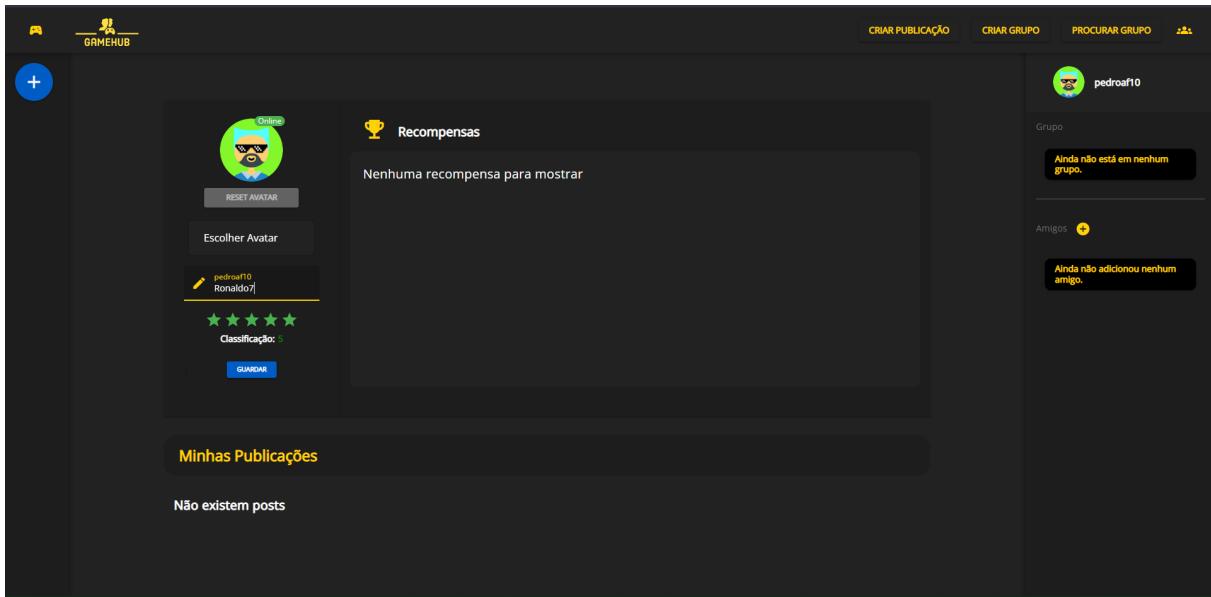


Figura 16.12: Página do perfil - alterar nome

No caso do nome escolhido ser diferente de todos os outros utilizadores da plataforma, este irá ser alterado com sucesso (figura 16.13) caso contrário, uma mensagem de erro irá ser apresentada (figura 16.14).

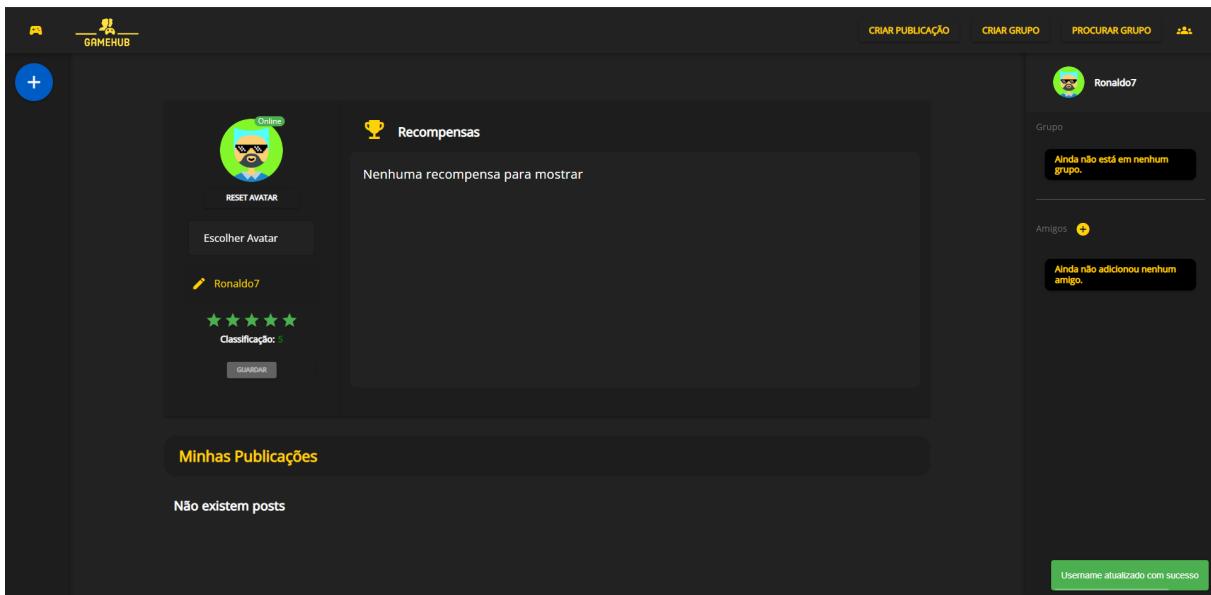


Figura 16.13: Página do perfil - alterar nome com sucesso

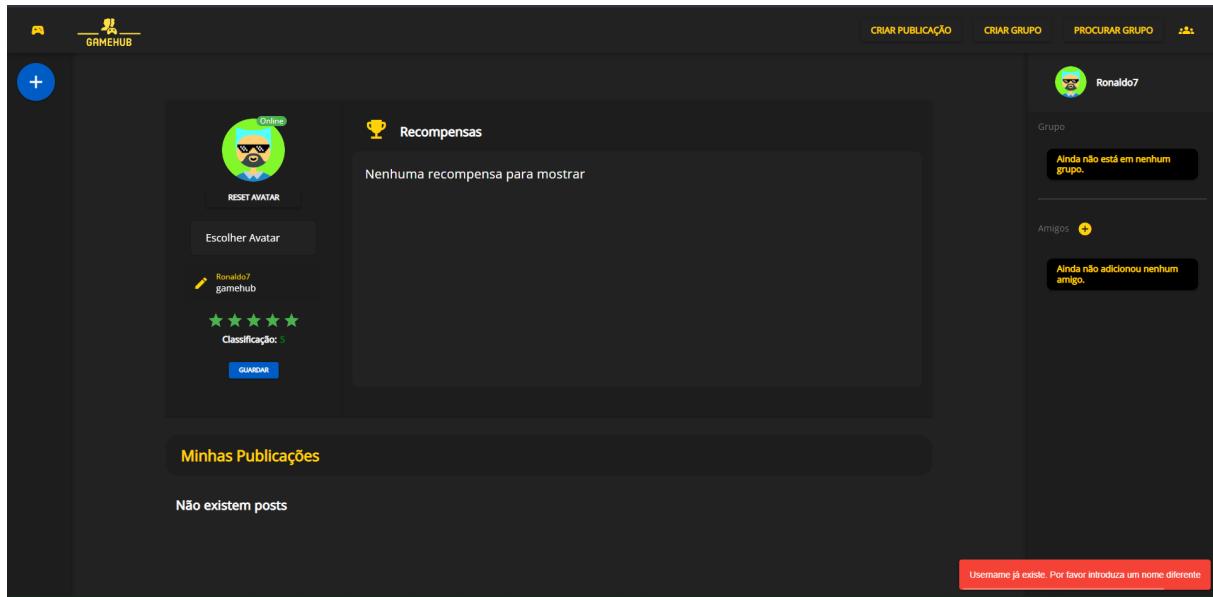


Figura 16.14: Página do perfil - alterar nome sem sucesso

16.1.4 Seguir novo jogo

Para seguir um novo jogo o utilizador primeiro precisa de carregar no botão redondo à esquerda com o ícone '+'.

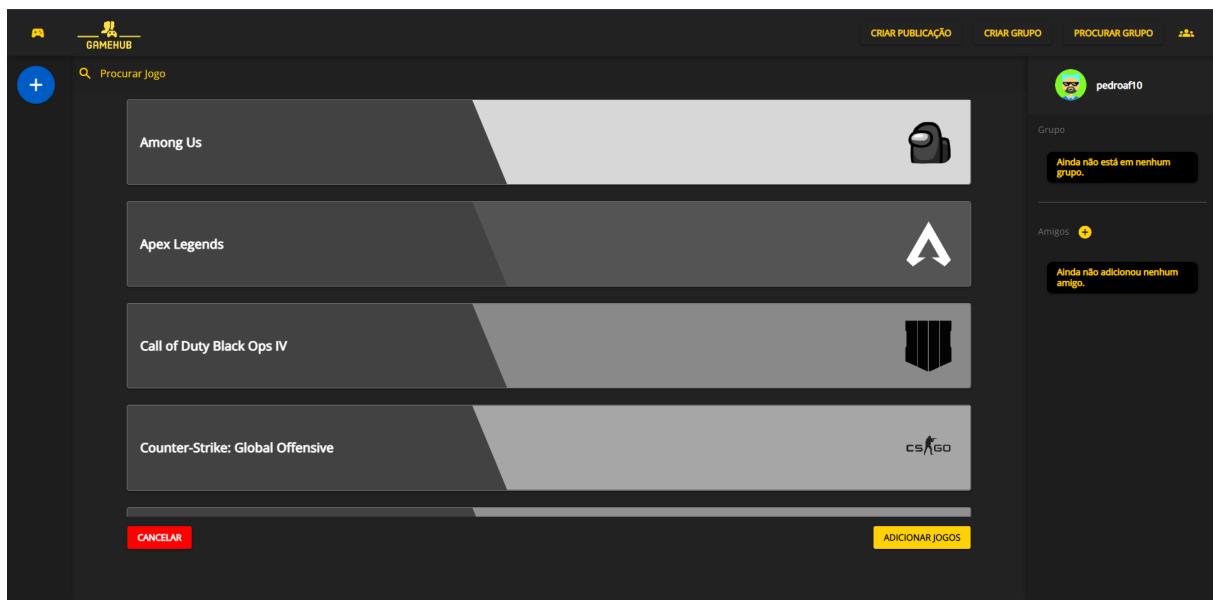


Figura 16.15: Página do adicionar jogos

Agora sendo apresentada uma lista de todos os jogos presentes na plataforma que o utilizador ainda não segue, é possível selecionar um ou mais jogos de uma vez e adicioná-los carregando em seguida em adicionar jogos.

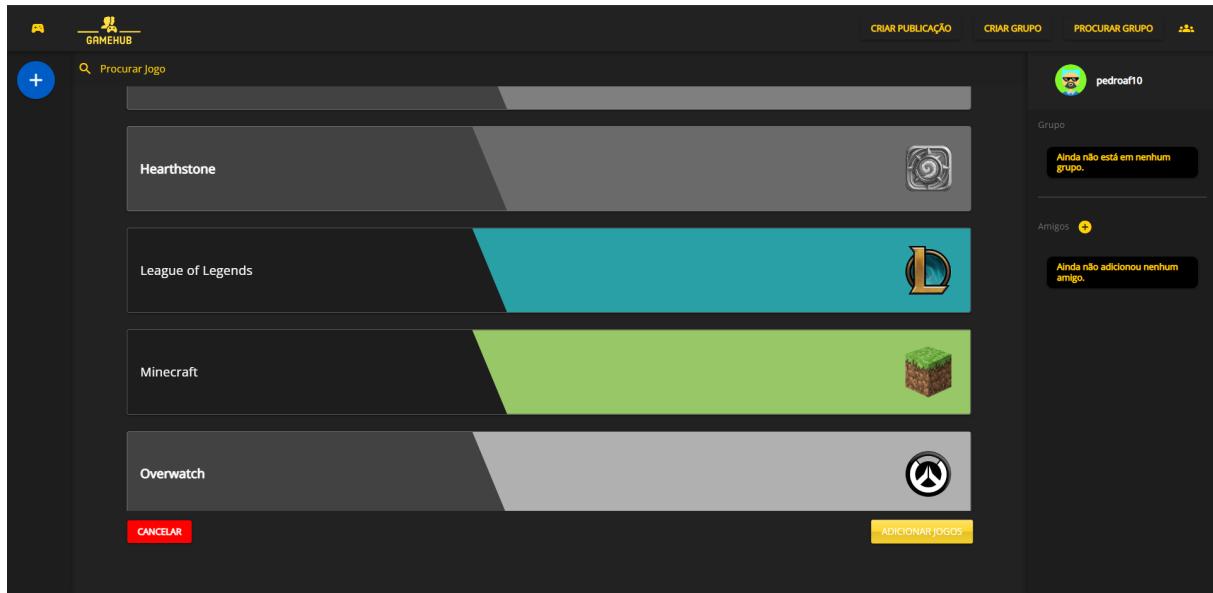


Figura 16.16: Página do adicionar jogos - jogos selecionados

Após a inserção o utilizador é redirecionado para a página principal e os jogos selecionados são apresentados à esquerda sob a forma de botões redondos como se pode ver em seguida.

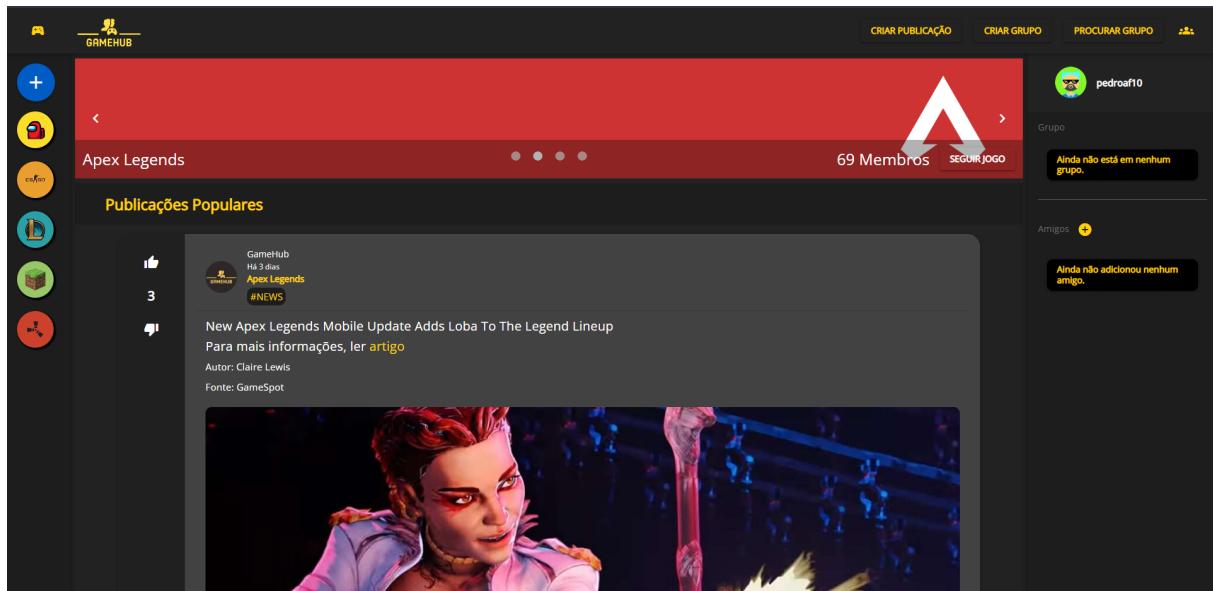


Figura 16.17: Página inicial - com jogos adicionados

16.1.5 Deixar de seguir jogo

Para deixar de seguir um jogo também é simples, existindo duas maneiras para o fazer. A primeira, carregando primeiro com o botão direito no jogo que deseja deixar. Em seguida carregar em deixar de seguir.

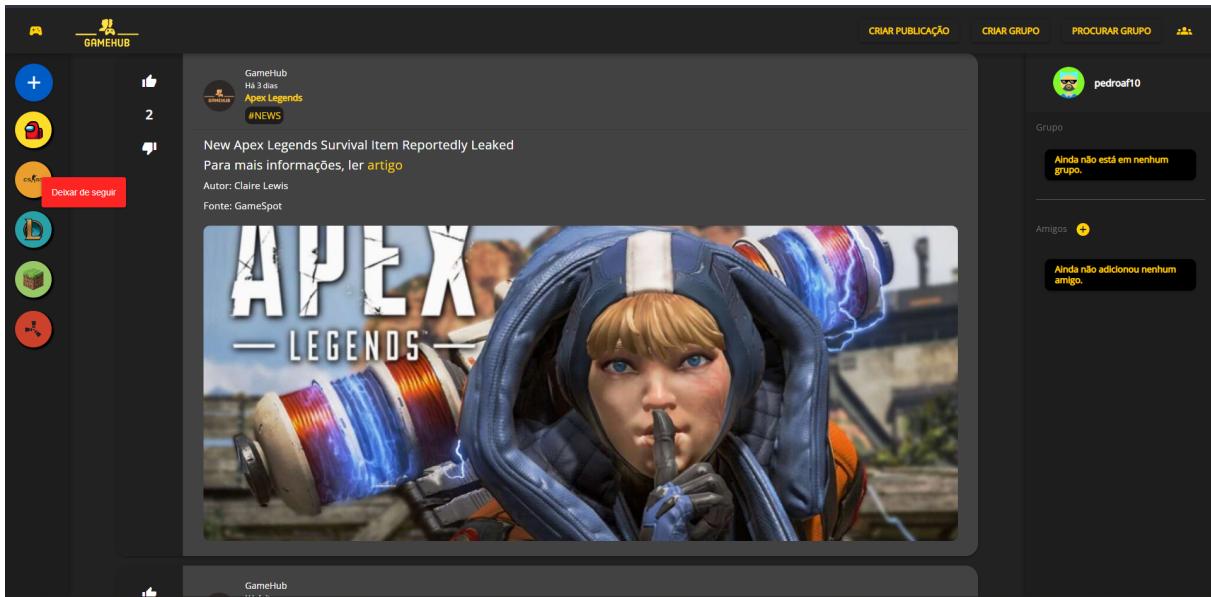


Figura 16.18: Página inicial - deixar de seguir jogo

Após realizada a ação o jogo irá ser removido da coluna lateral, como se pode ver em seguida:

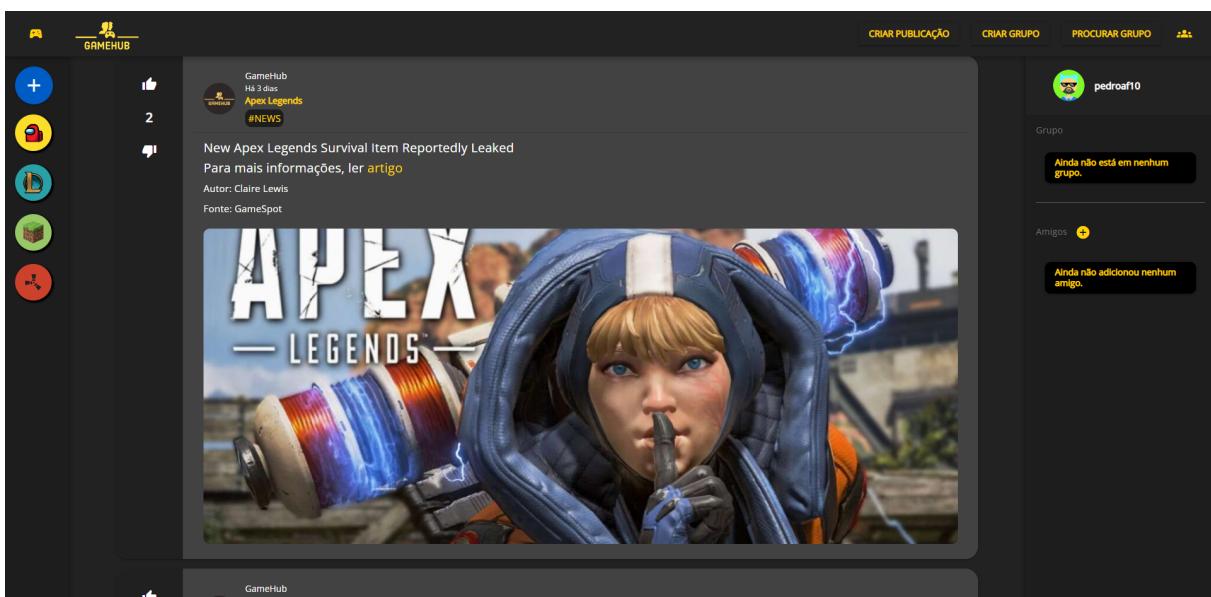


Figura 16.19: Página inicial - deixar de seguir jogo

A segunda basta entrar no *feed* do jogo correspondente e carregar no botão vermelho deixar de seguir jogo.

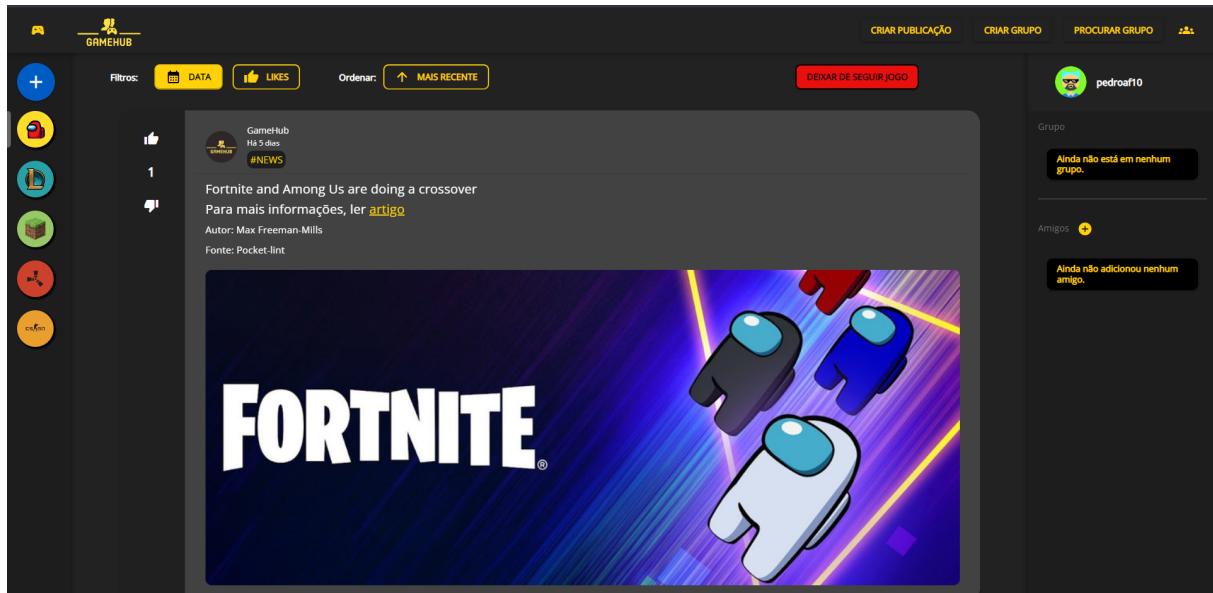


Figura 16.20: Feed do jogo - deixar de seguir jogo 2

Após realizada a ação e a confirmação. o jogo irá ser removido da coluna lateral e redirecionado para a página principal, como se pode ver em seguida:

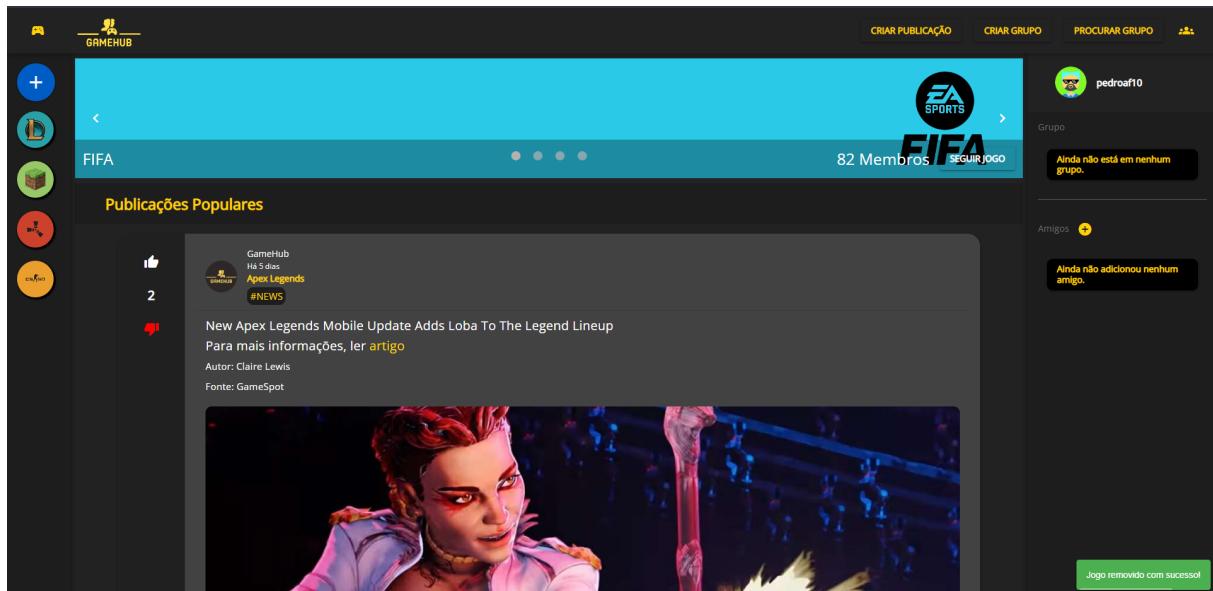


Figura 16.21: Página inicial - deixar de seguir jogo 2

16.1.6 Aceder feed de um jogo

Um utilizador para aceder ao *feed* de um determinado jogo, precisa apenas de carregar à esquerda no botão correspondente ao jogo que pretende visualizar os *posts*. É de salientar que um utilizador apenas pode visualizar o *feed* dos jogos que segue.

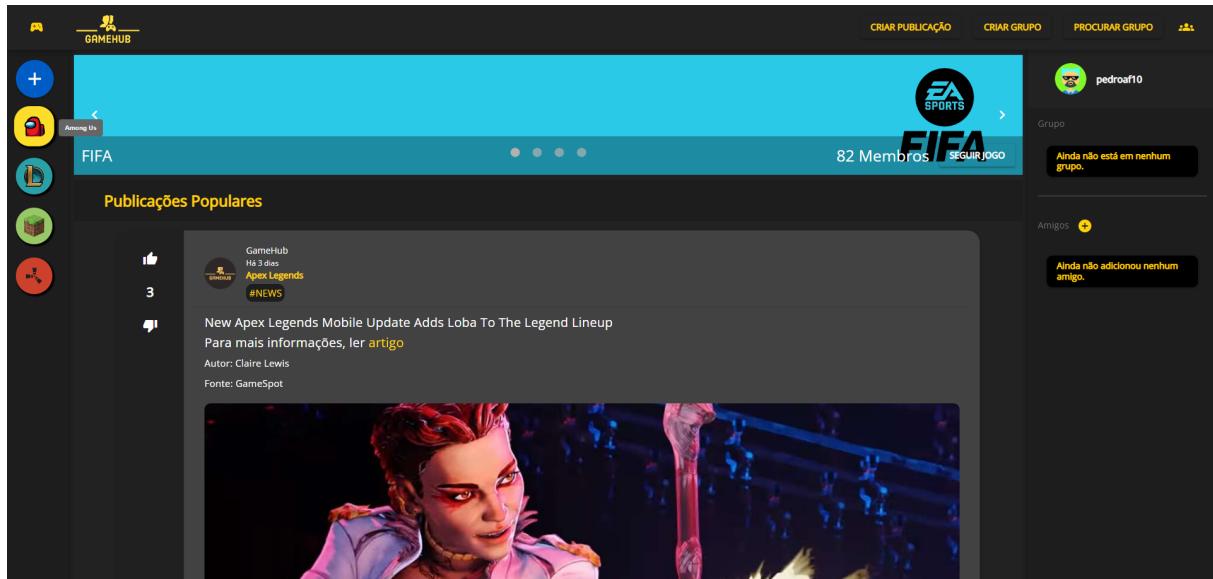


Figura 16.22: Página inicial - aceder ao *feed* de um jogo

Após esta ação ter sido realizada, o utilizador é redirecionado para o *feed* correspondente, podendo agora visualizar todos os *posts* multimédia realizados relativamente a este jogo.

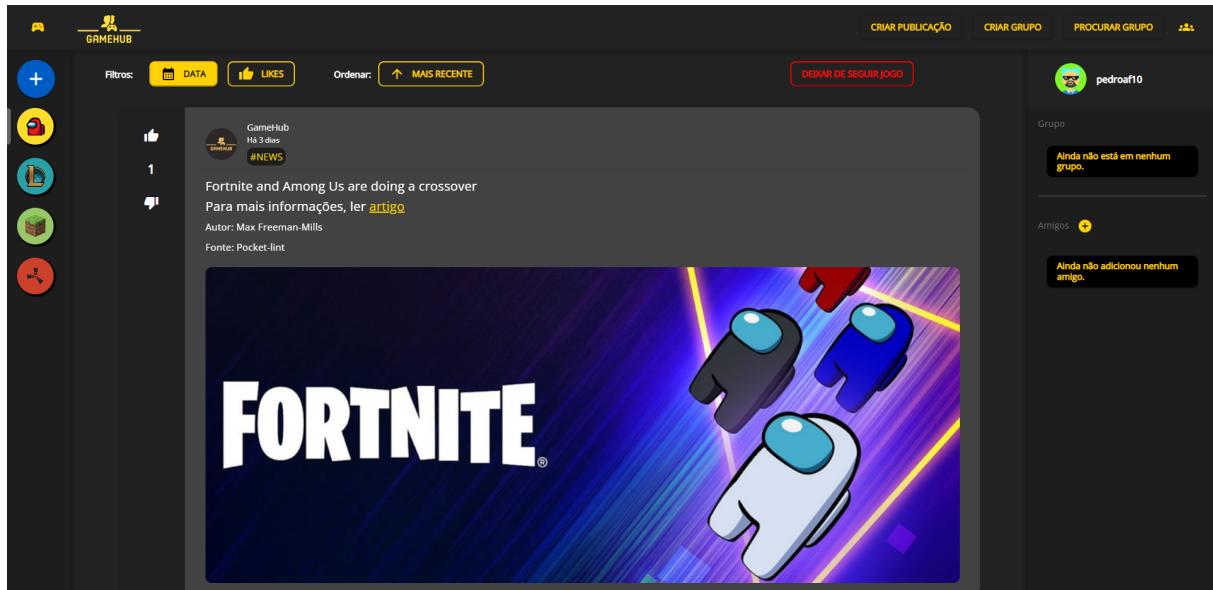


Figura 16.23: Página *feed* do Among Us

16.1.7 Criar publicação grupo

Para um utilizador procurar pessoas com quem jogar, este precisa de criar uma publicação de grupo, ou então juntar-se a um grupo já existente. Primeiramente para criar uma publicação de grupo um utilizador terá que carregar em criar publicação:

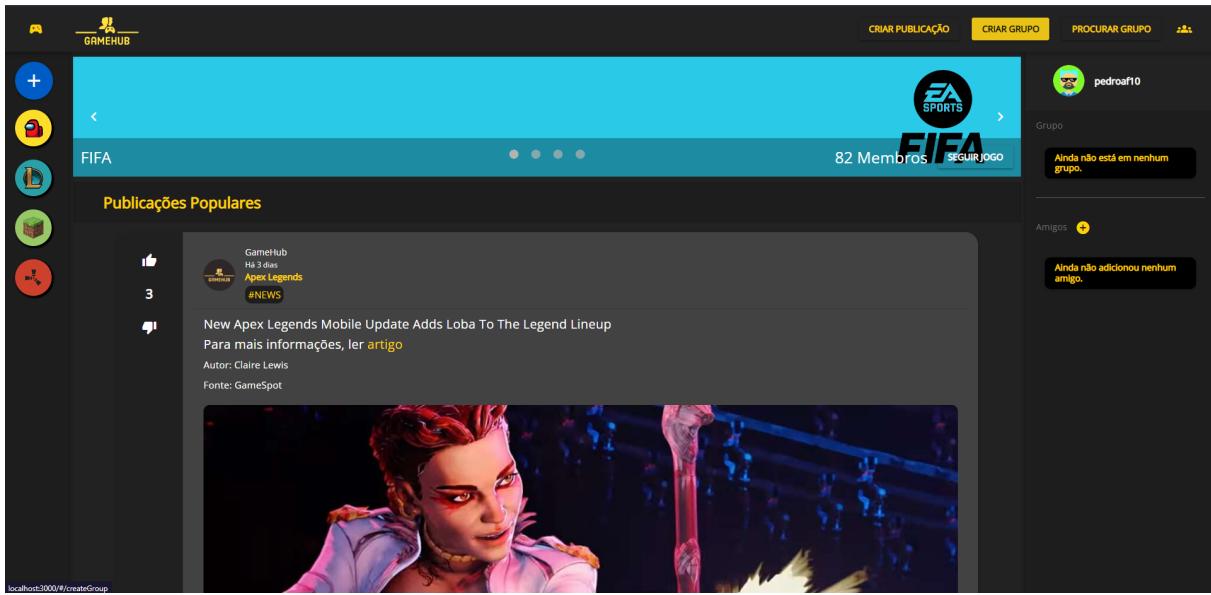


Figura 16.24: Página inicial - criar publicação de grupo

Após realizada esta ação, a página de criação de grupo será apresentada. Aqui será necessário preencher os campos relativos ao grupo que pretende criar.

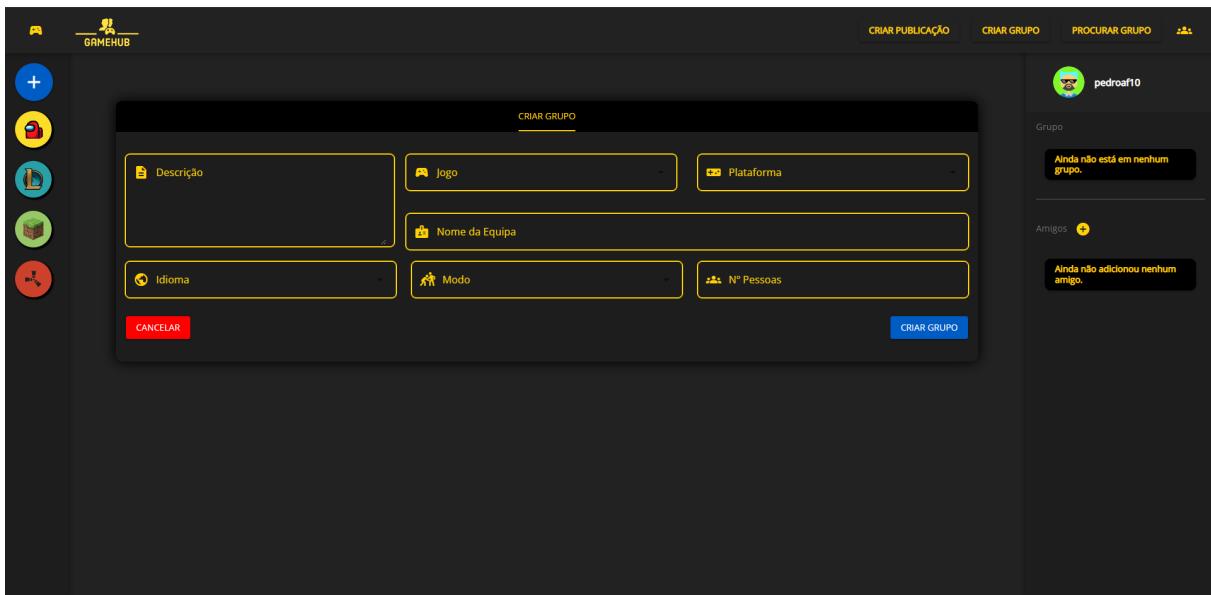


Figura 16.25: Página criar publicações - criar grupo

Em seguida o utilizador será redirecionado para a secção de criar um grupo e terá que introduzir os dados relativos ao grupo que pretende criar. Caso preencha incorretamente os campos necessários irá aparecer uma mensagem de erro, assim como indicar quais os campos obrigatórios para a criação de um grupo (figura 16.26). Caso tudo seja introduzido de forma correta, uma mensagem de sucesso será apresentada e o grupo criado aparecerá na secção de grupo à direita (figura 16.27).

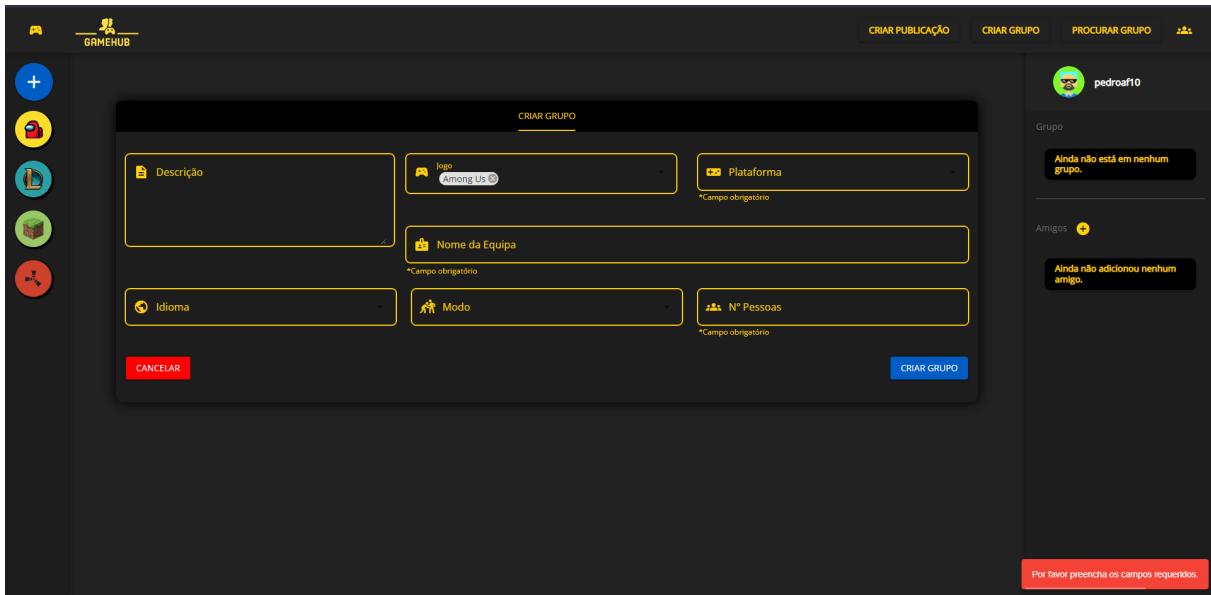


Figura 16.26: Página criar publicações - introdução dos dados incorretamente

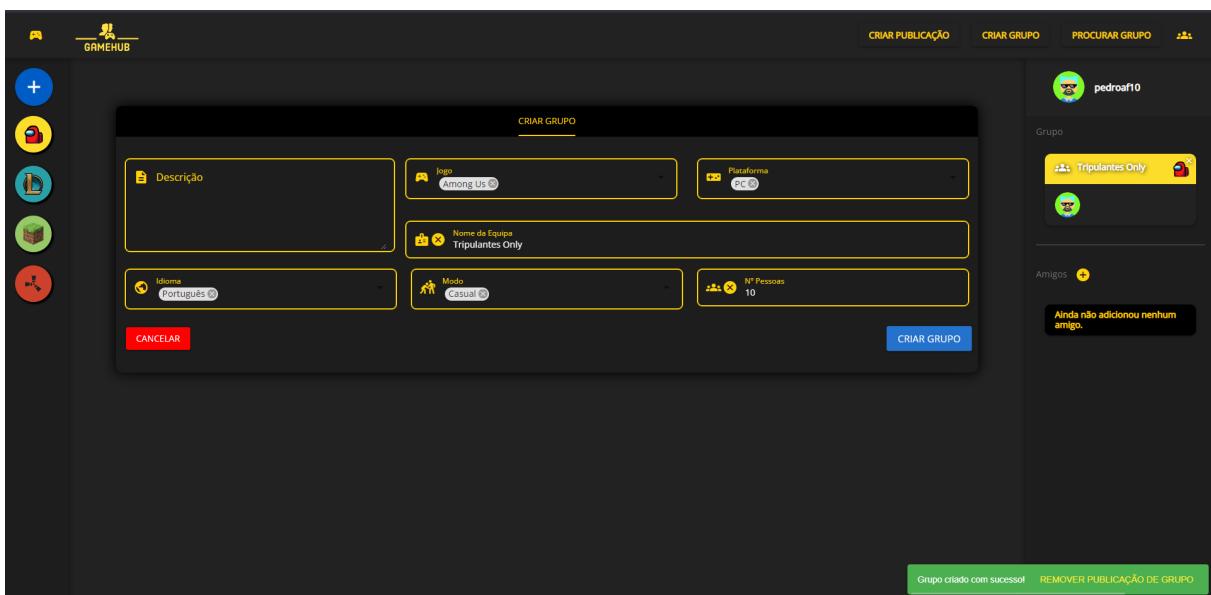


Figura 16.27: Página criar publicações - introdução dos dados de forma correta

16.1.8 Juntar a grupo

A outra forma de encontrar pessoas com quem jogar é procurar grupos criados por outras pessoas. Para isso basta carregar em Procurar Grupo, no canto superior direito.

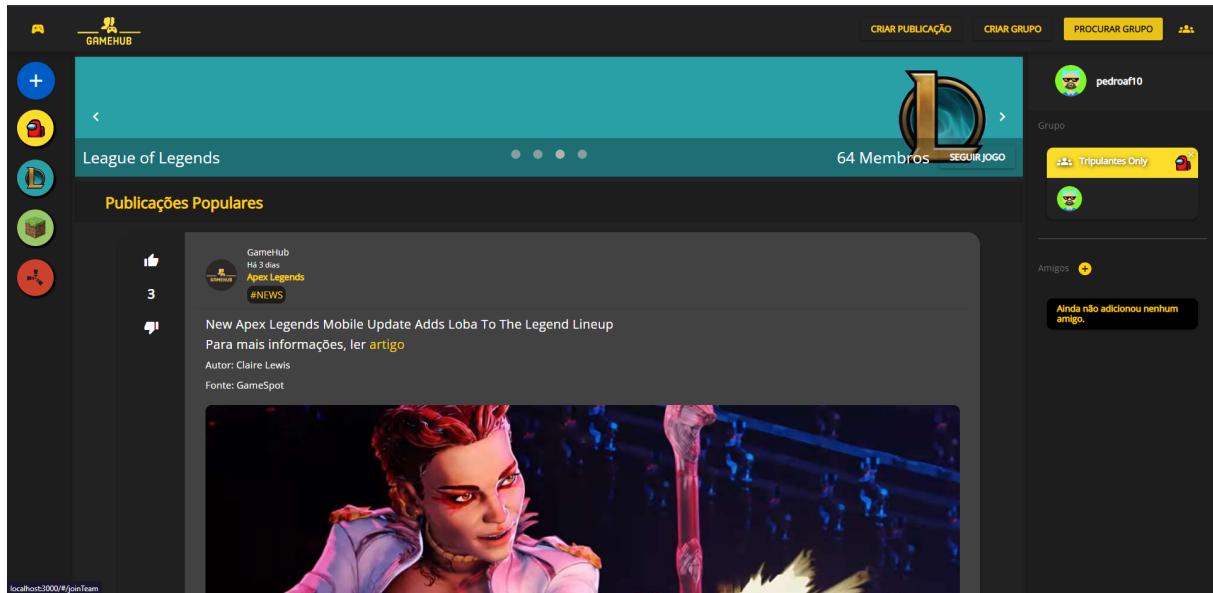


Figura 16.28: Página inicial - procurar grupo

Após realizada a ação anterior, o utilizador depara-se com a página de procura de grupos. É de notar que nesta secção apenas aparecem grupos aos quais é possível juntar, isto é, grupos que já se encontram cheios não irão aparecer.

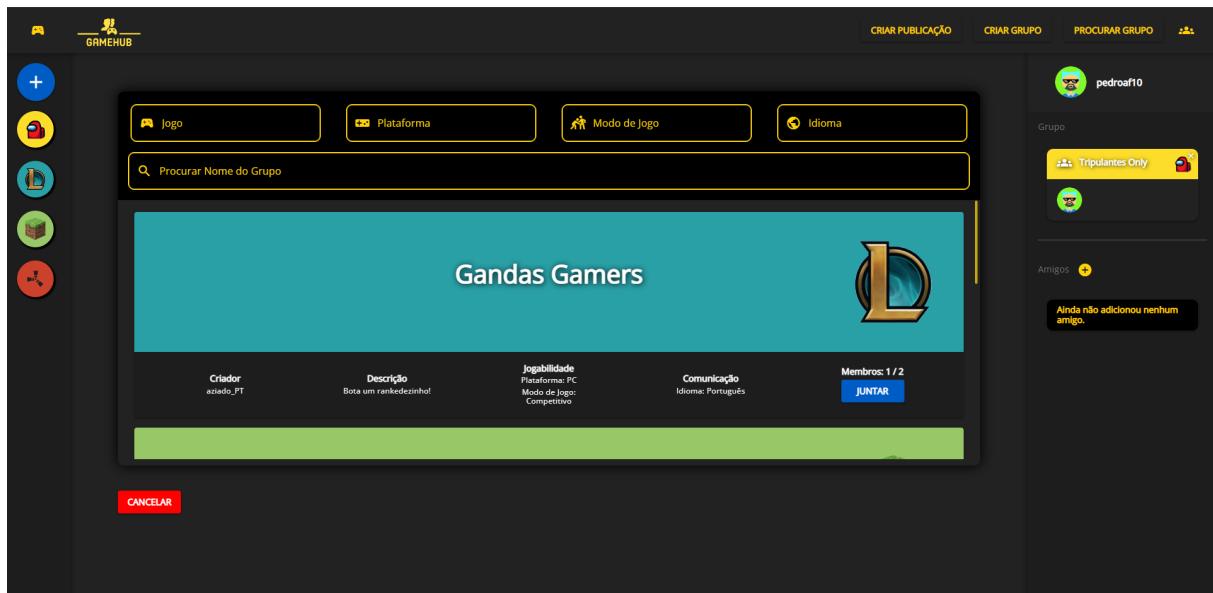


Figura 16.29: Página procura de grupo

Agora um utilizador pode usufruir dos filtros em localizados no topo da página, de modo a encontrar um grupo que melhor o satisfaça. Pode procurar grupos com base no jogo que quer jogar, assim como a sua plataforma, pode indicar a linguagem preferencial das pessoas do grupo, o modo de jogo que pretende jogar e ainda caso já conheça o grupo em que quer entrar, pode pesquisar o nome concreto do mesmo.

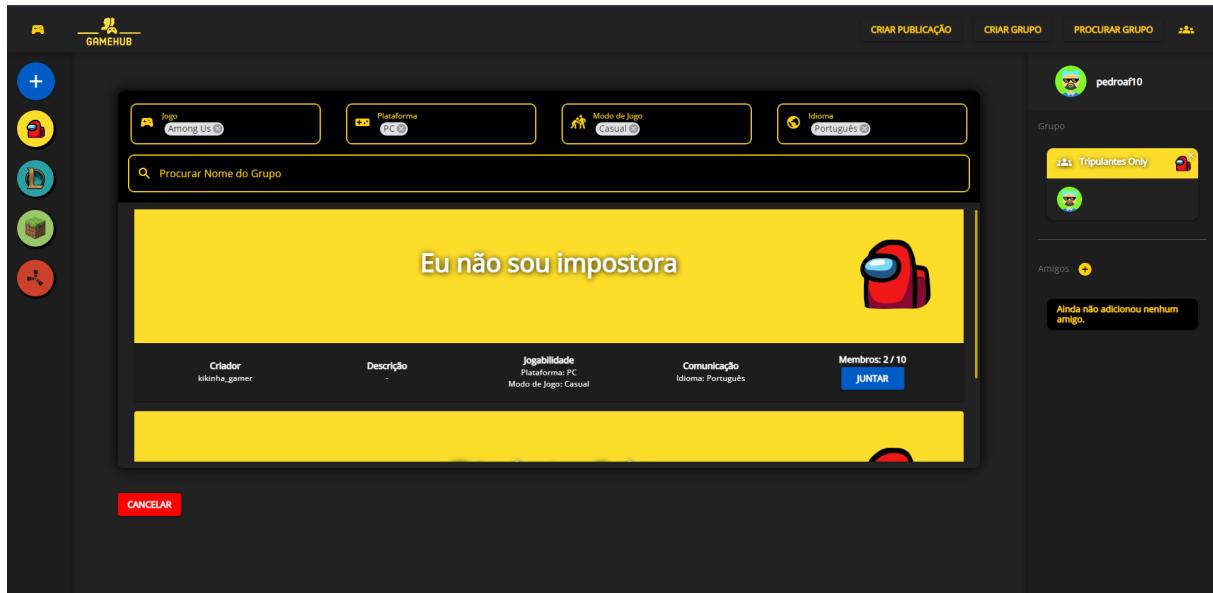


Figura 16.30: Página procura de grupo - com filtros

Após escolhido o grupo, basta carregar em juntar, e em seguida confirmar a sua ação, uma vez que caso já esteja dentro de um grupo, este será substituído.

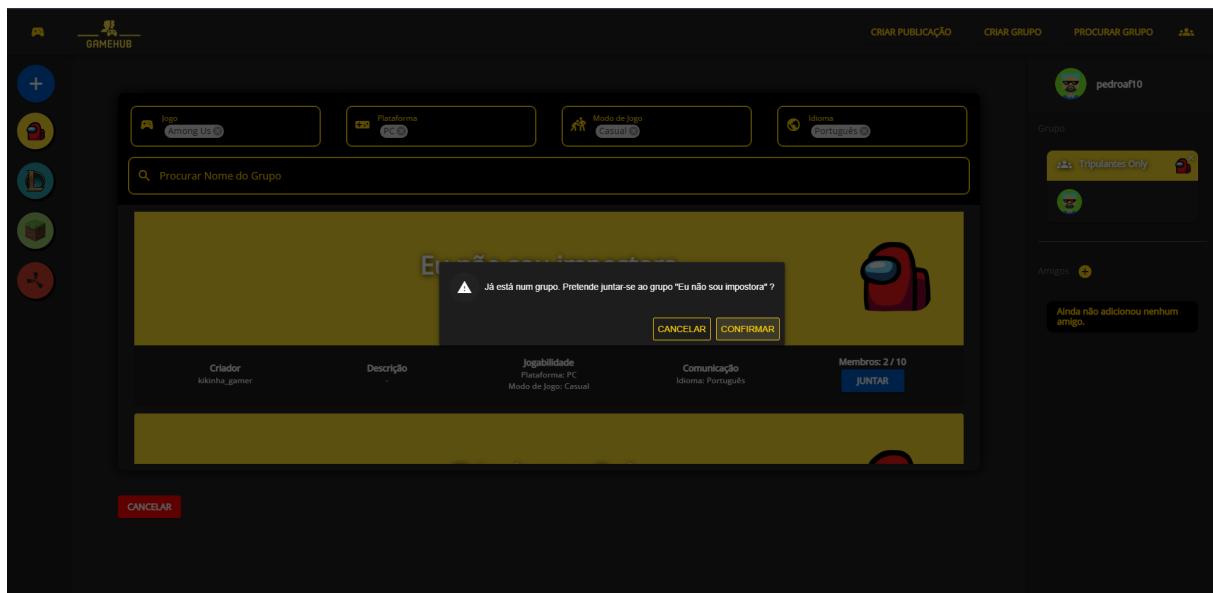


Figura 16.31: Página procura de grupo - confirmação

Finalmente, confirmada a ação anterior, uma mensagem de sucesso é apresentada(onde é possível reverter a ação), assim como o grupo do utilizador à direita é substituído ou introduzido dependendo da pré-existência ou não de um grupo. Para além disso o botão de juntar ao grupo atual deixa de estar disponível para o utilizador.

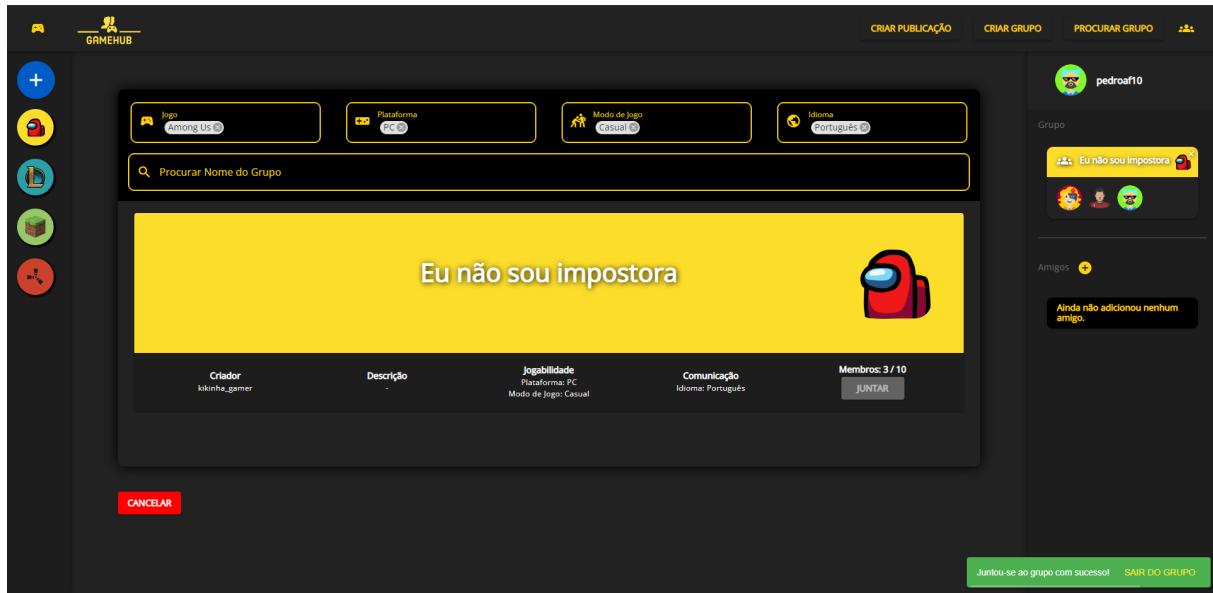


Figura 16.32: Página procura de grupo - juntou-se a novo grupo

16.1.9 Sair de um grupo

Para um utilizador sair de um grupo, pode entrar num novo grupo ou criar uma nova publicação de grupo como demonstrado anteriormente, ou então carregar na cruz do cartão do grupo e em seguida confirmar que pretende sair do grupo.

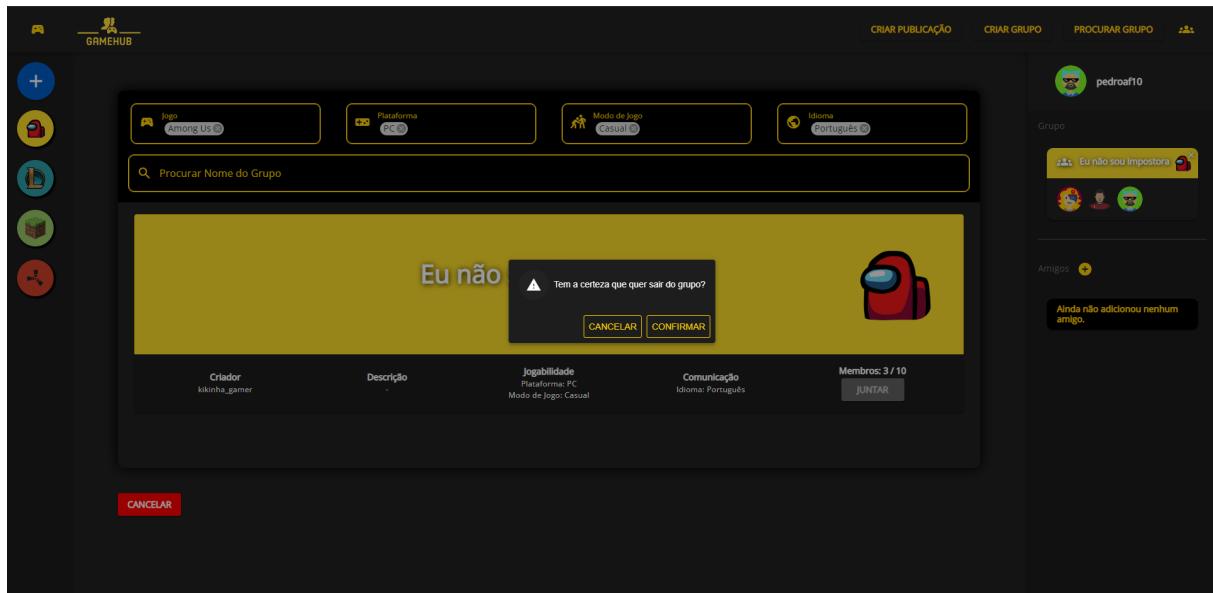


Figura 16.33: Sair de um grupo - confirmação

Após confirmada a ação o utilizador irá encontrar-se novamente sem qualquer grupo.

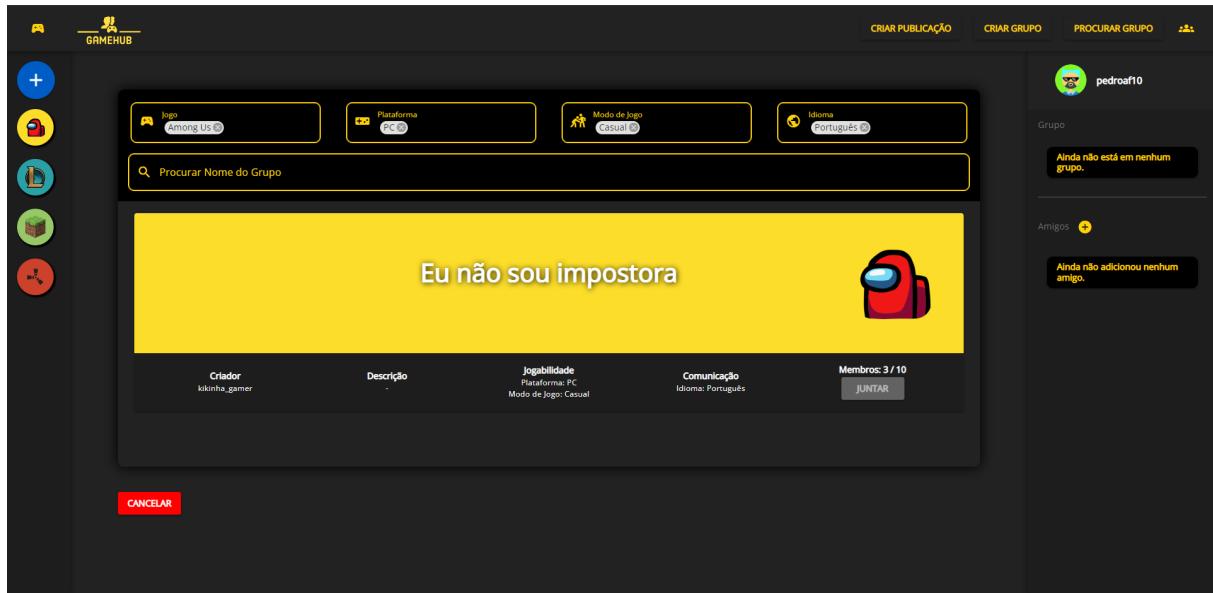


Figura 16.34: Sair de um grupo - conclusão

16.1.10 Enviar mensagem grupo

Quando um utilizador se encontra num grupo, pode comunicar com os seus integrantes. Para tal primeiro precisa de carregar na parte colorida do cartão do grupo.

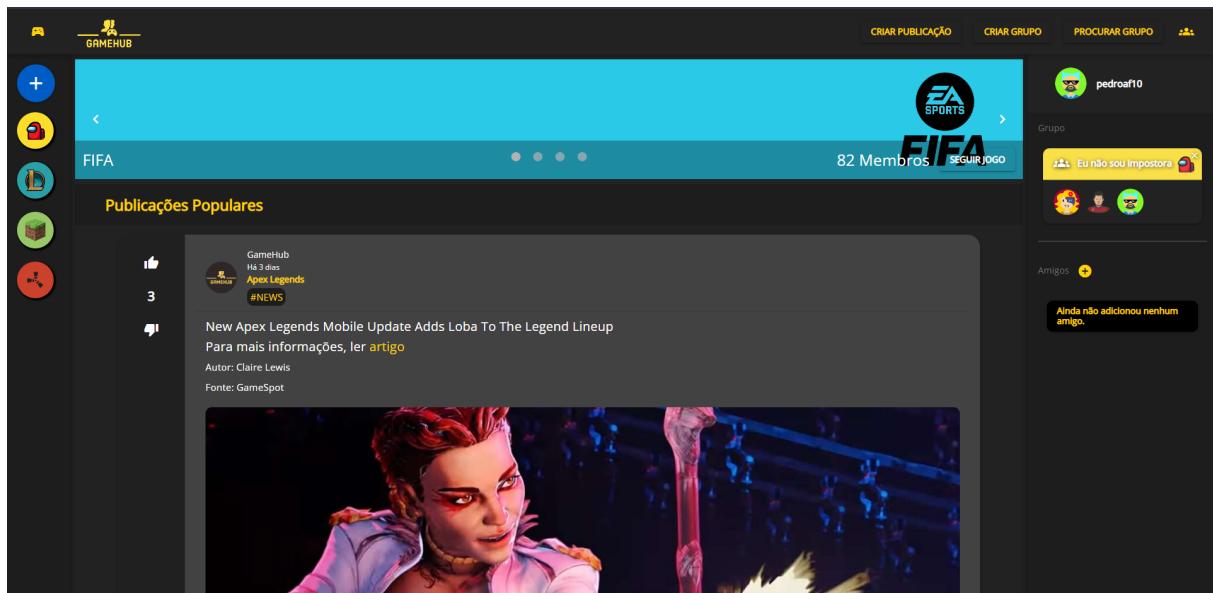


Figura 16.35: Pagina inicial - conversa de grupo

Em seguida a conversa irá abrir e serão apresentadas todas as mensagens já envidas nesse grupo.

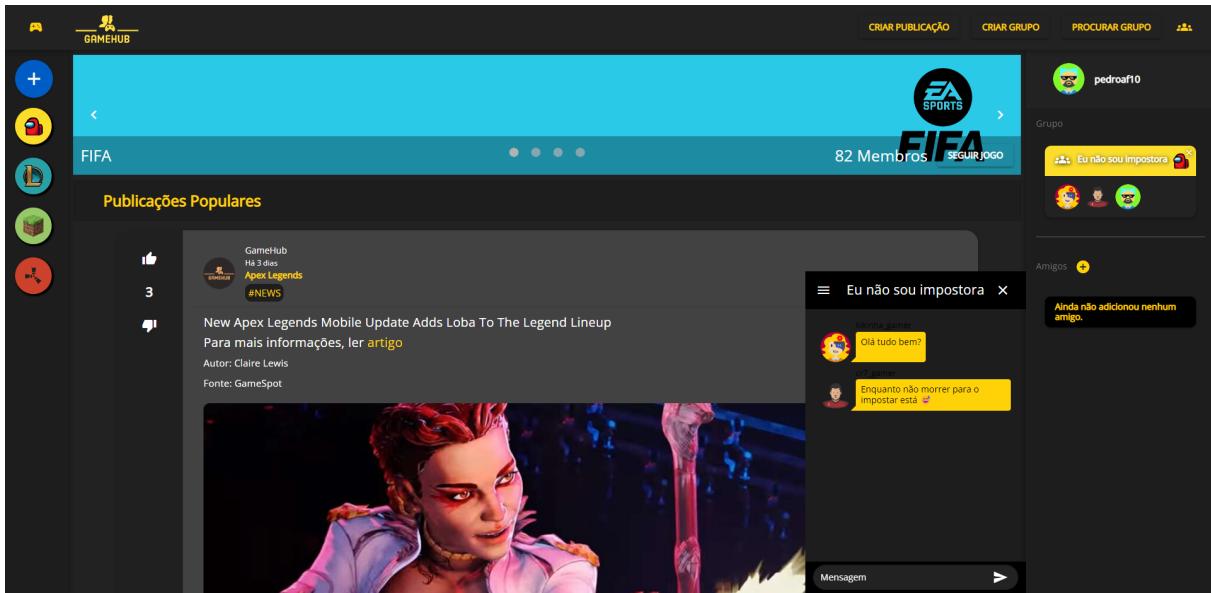


Figura 16.36: Pagina inicial - conversa de grupo aberta

Para enviar uma mensagem um utilizador começa por escrever a mensagem pretendida na secção da mensagem.

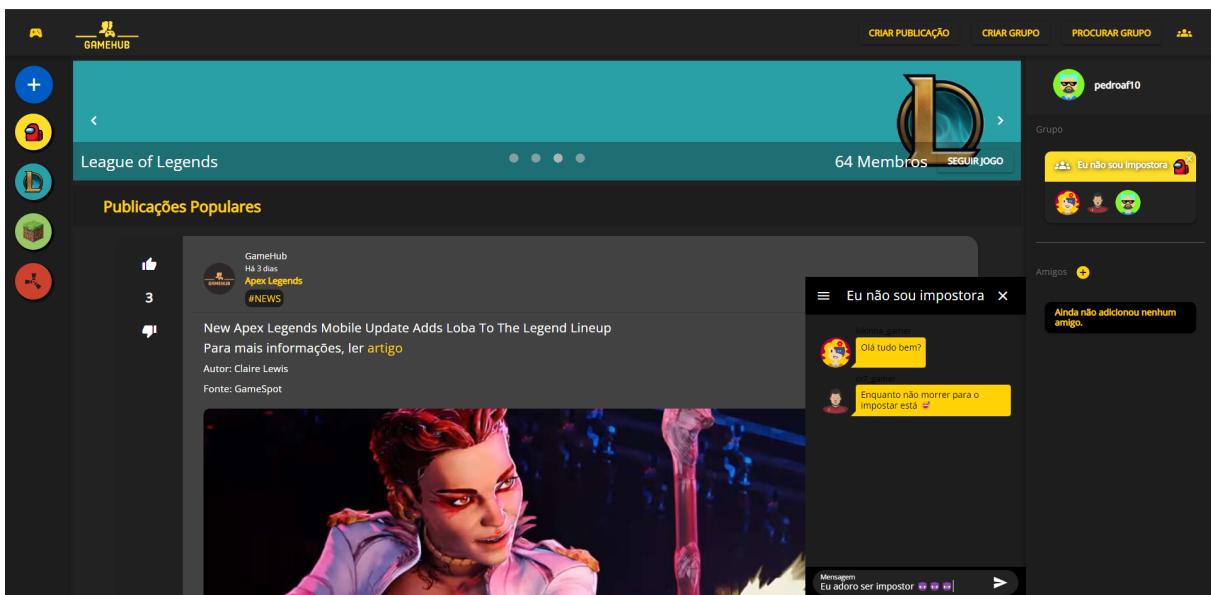


Figura 16.37: Pagina inicial - escrever mensagem para o grupo

Quando estiver contente com a mensagem escrita precisa apenas de carregar no icon de envio, e a sua mensagem será enviada para a conversa do grupo.

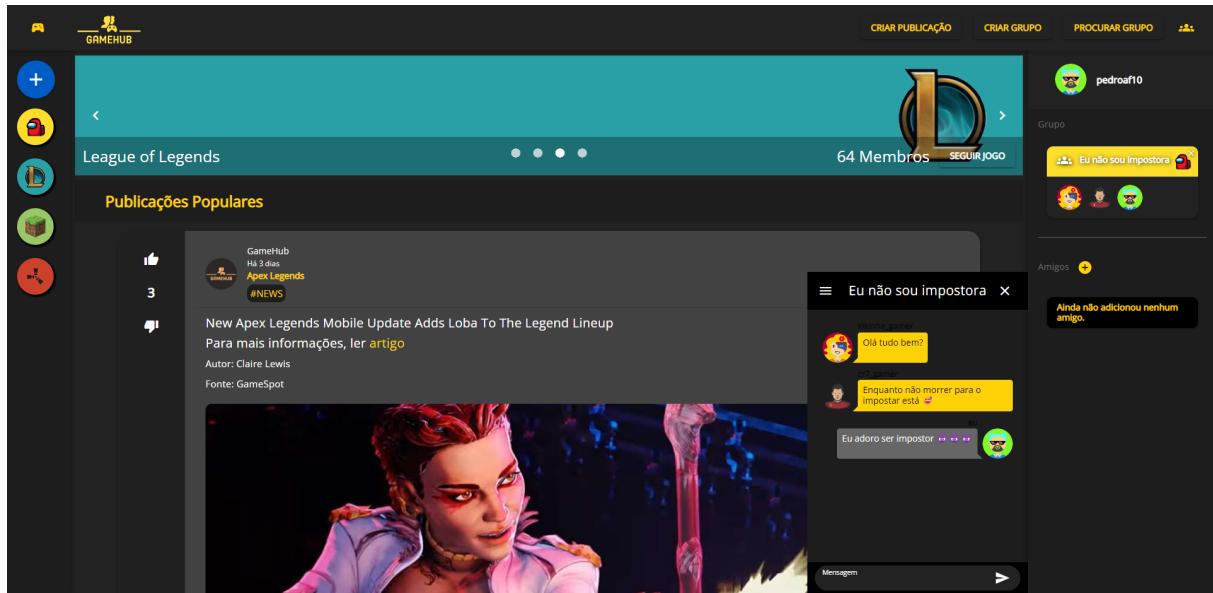


Figura 16.38: Pagina inicial - mensagem para o grupo enviada

16.1.11 Adicionar amigo

Para adicionar um amigo um utilizador pode proceder de duas formas distintas. A primeira é adicionar um amigo carregando primeiro no botão (+) na direita onde diz Amigos:

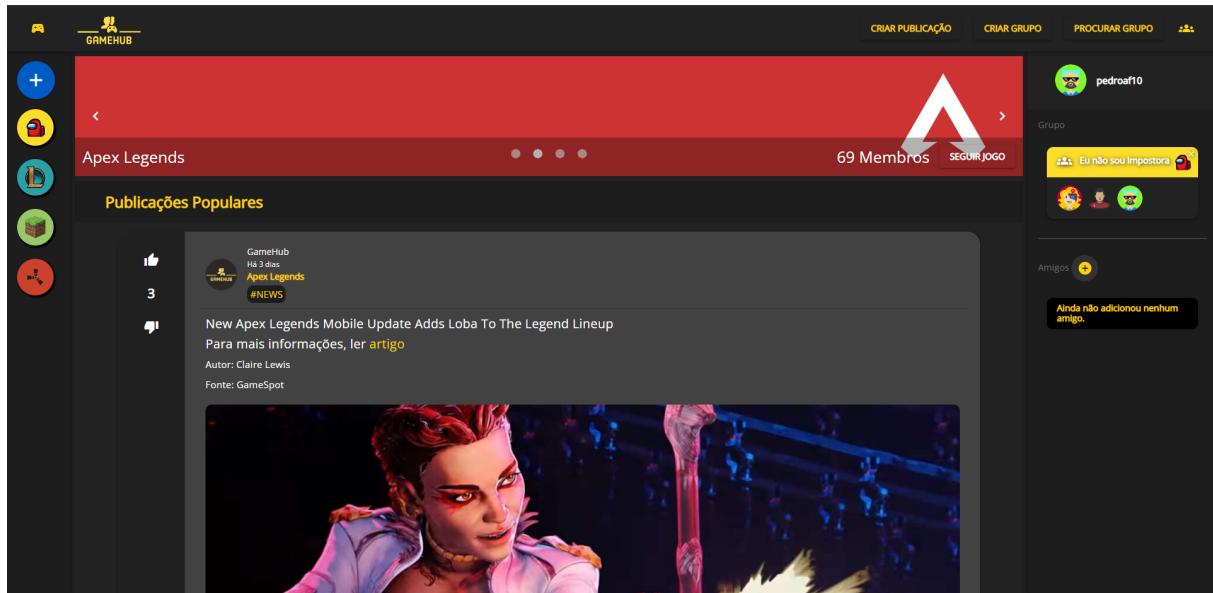


Figura 16.39: Pagina inicial - adicionar amigo

Em seguida é apresentado um sub menu para a adição de amigos. Aqui é apresentada uma secção de amigos sugeridos (utilizadores não amigos dentro do mesmo grupo) e uma secção onde é possível a introdução do nome de um qualquer utilizador do GameHub.

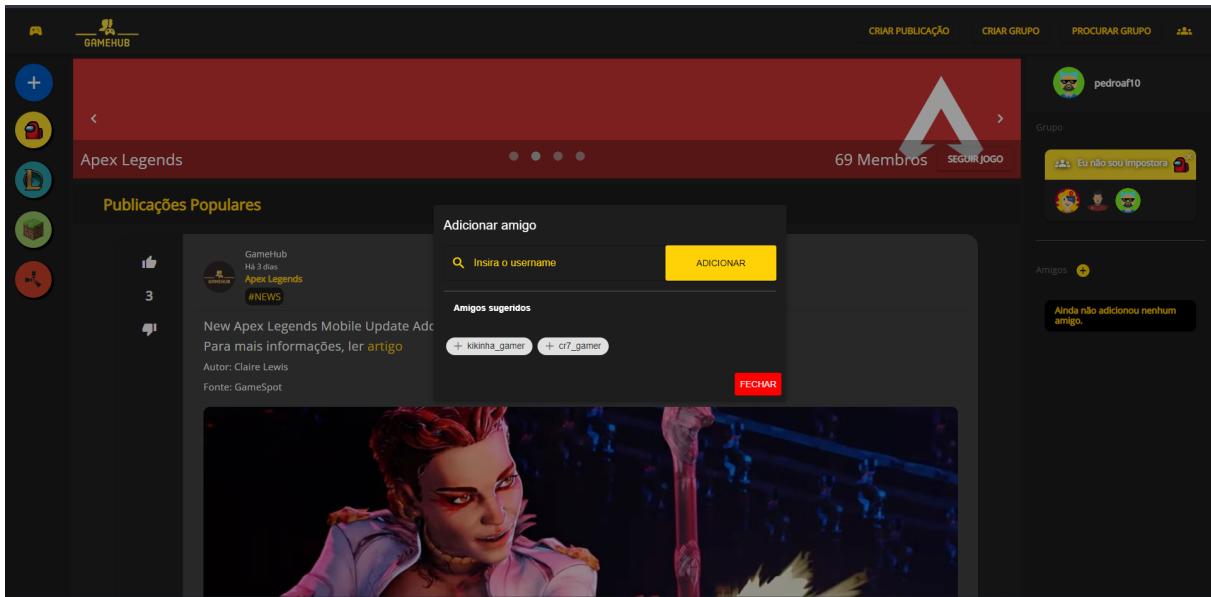


Figura 16.40: Menu adição de um amigo

Carregando numa das sugestões, o campo do utilizador a inserir será preenchido com esse mesmo utilizador:

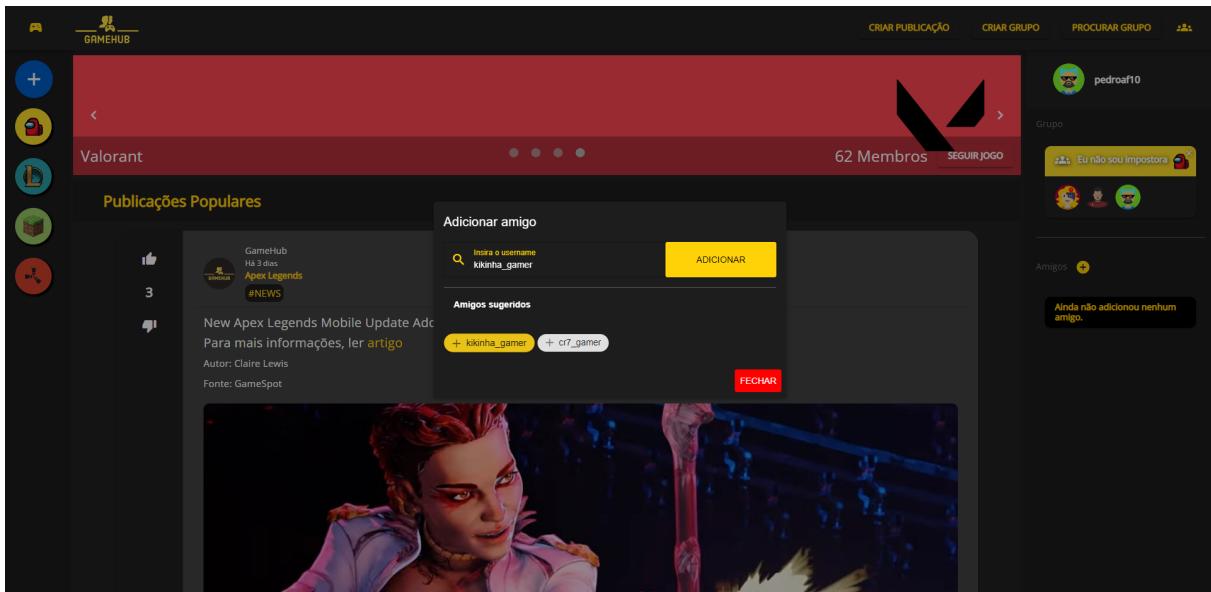


Figura 16.41: Menu adição de um amigo - Amigo introduzido

Por fim basta carregar em adicionar. Uma mensagem de confirmação será apresentada e o amigo será adicionado à lista de amigos.

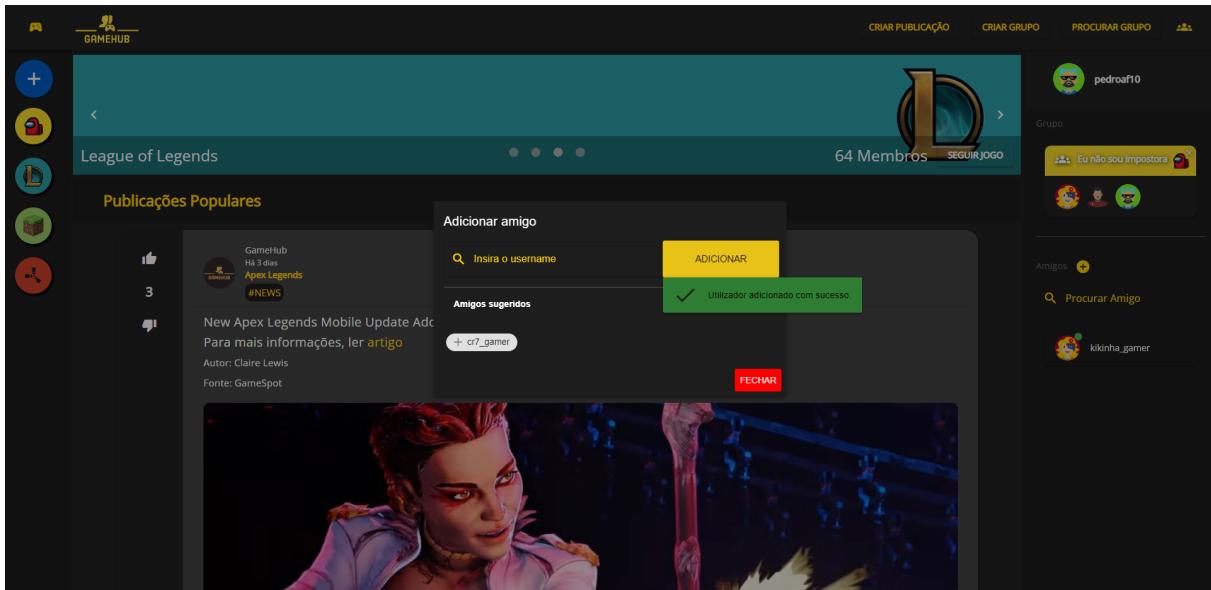


Figura 16.42: Menu adição de um amigo - Amigo introduzido

Caso um utilizador tente adicionar um utilizador que não existe, este obterá uma mensagem de erro afirmando que esse utilizador não se encontra presente na plataforma.

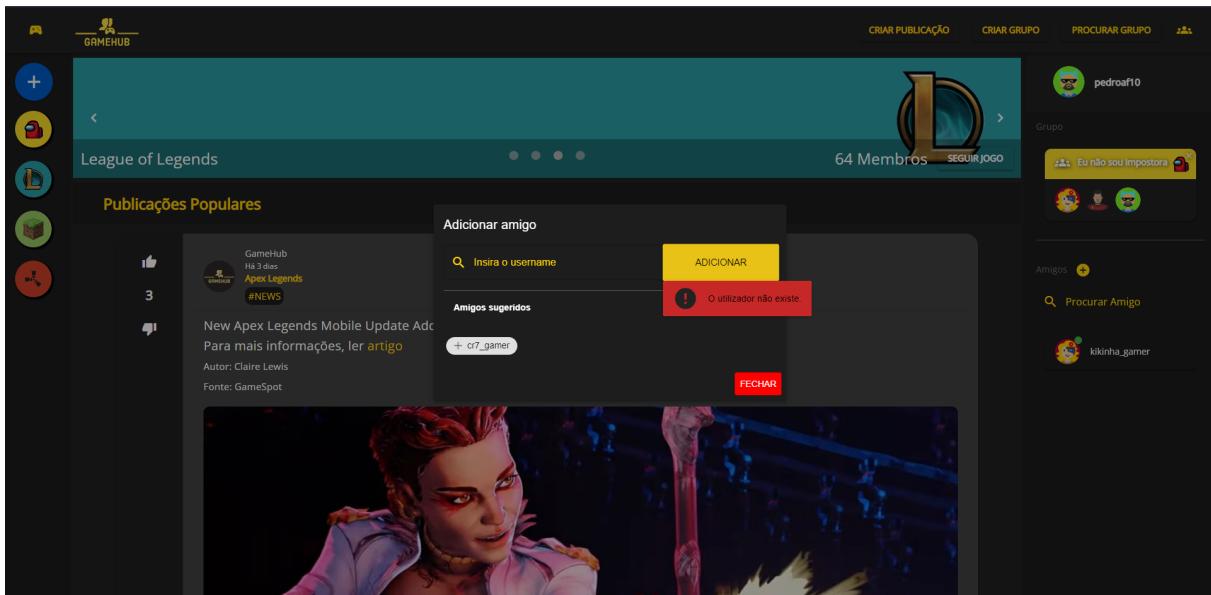


Figura 16.43: Menu adição de um amigo - Erro ao introduzir

Uma outra maneira de adicionar um amigo é a acedendo ao perfil do utilizador a adicionar. Estando o utilizador a adicionar no mesmo grupo, basta carregar no seu avatar para aceder ao seu perfil.

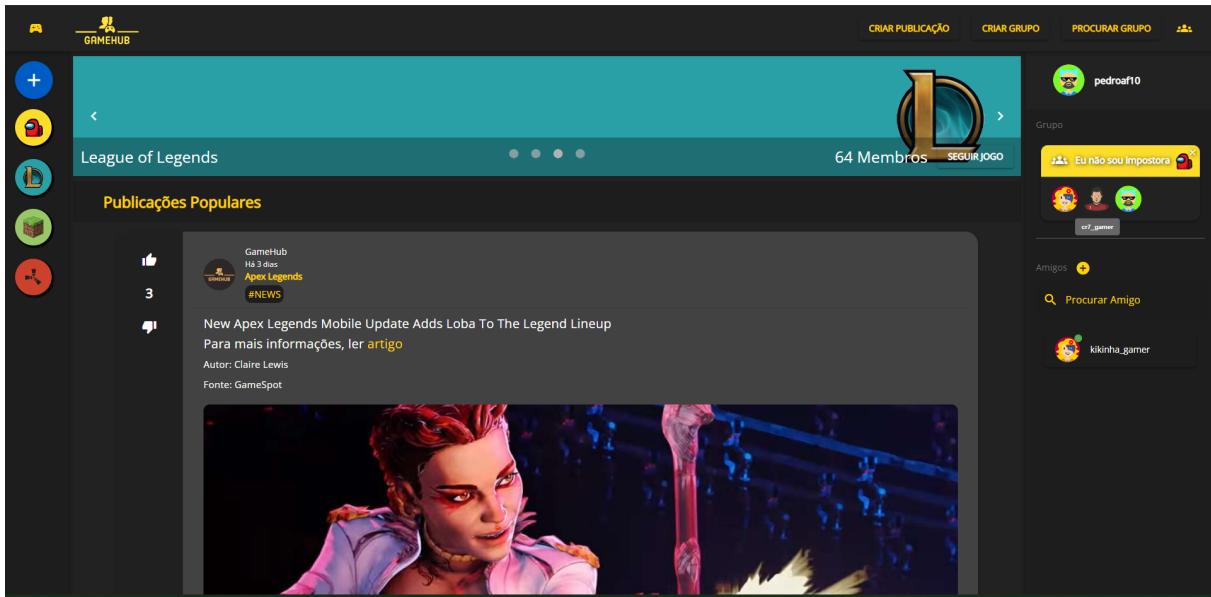


Figura 16.44: Menu inicial - Adicionar amigo, forma 2

Em seguida, já dentro do perfil, basta apenas carregar no botão Adicionar Amigo.

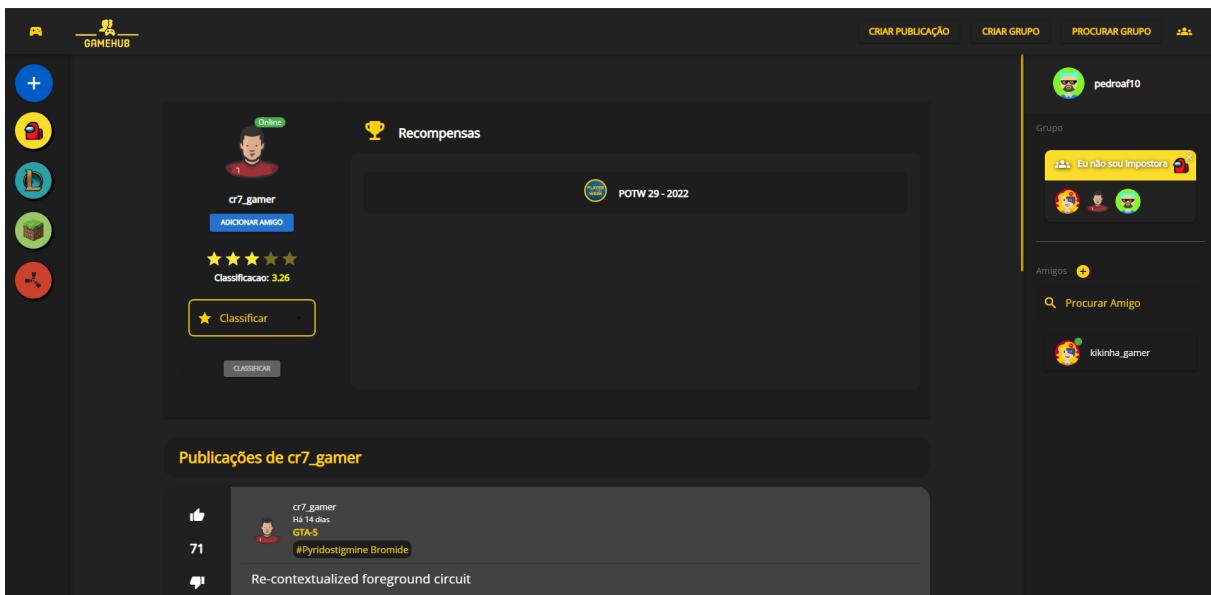


Figura 16.45: Página perfil - Adicionar amigo

Realizada então a adição com sucesso, o utilizador é adicionado à lista de amigos.

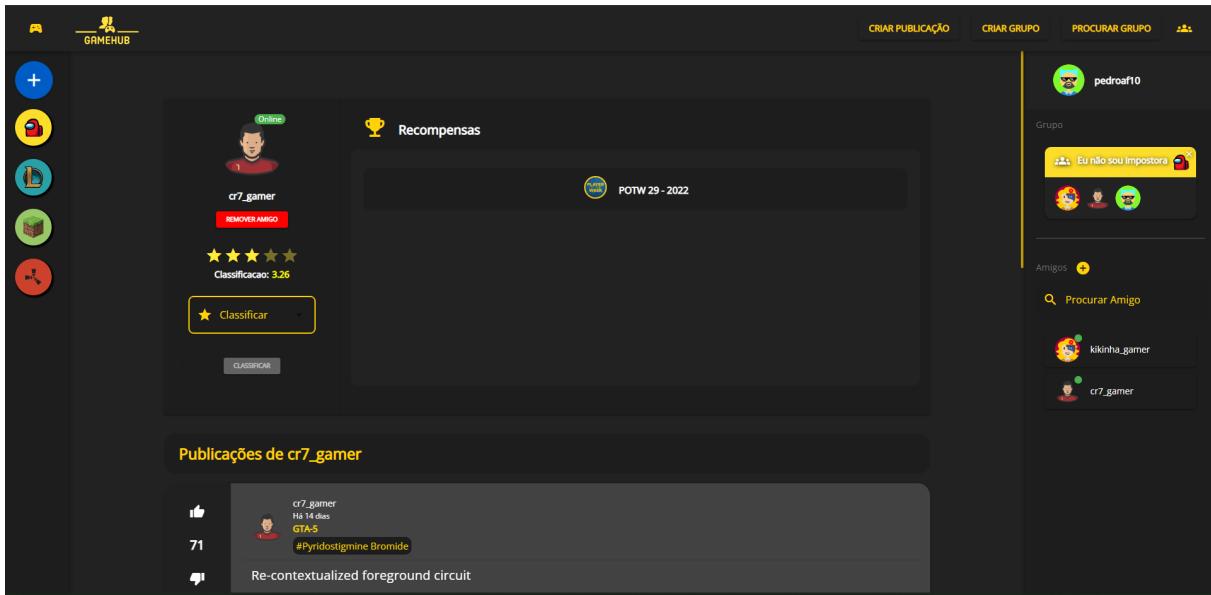


Figura 16.46: Página perfil - Amigo adicionado

16.1.12 Remover amigo

Após adicionar um amigo é possível proceder à sua remoção através do uso do botão direito nele e carregando em remover, ou através do seu perfil. Para tal primeiro é necessário aceder ao perfil do mesmo. Para tal pode-se proceder através do grupo, no caso de estarem no mesmo grupo, ou através da conversa entre os dois. Neste caso irá-se proceder através da conversa, portanto é necessário carregar no amigo na lista de amigos.

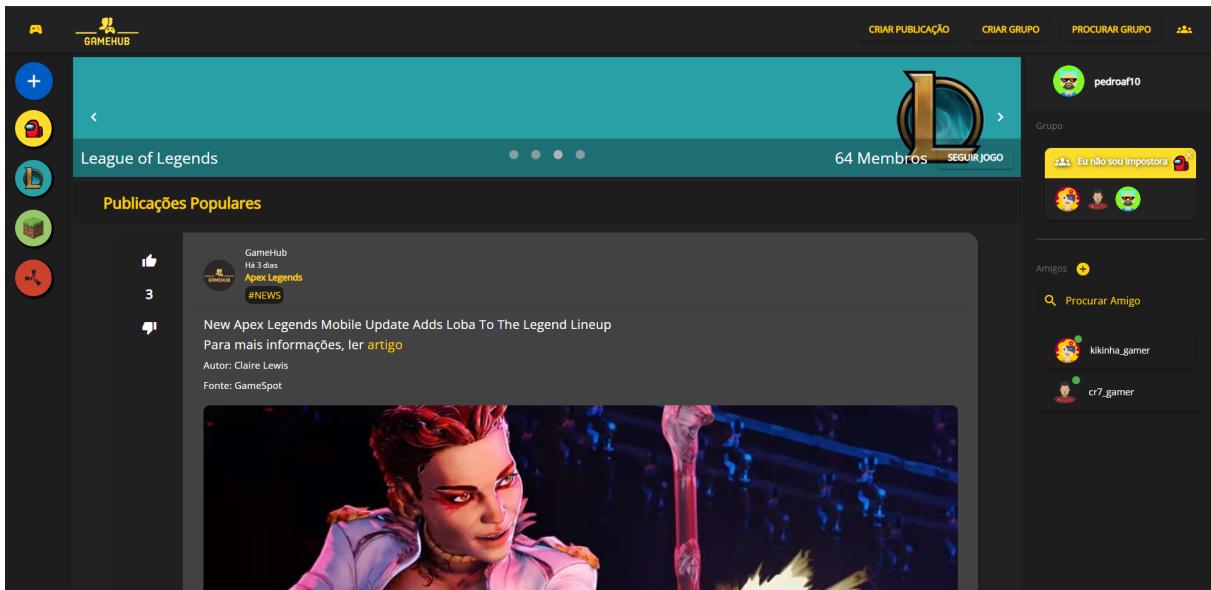


Figura 16.47: Página inicial - Abrir conversa com amigo

Após realizada a ação anterior, a conversa entre os dois irá abrir, e em seguida é necessário carregar no avatar do amigo, de modo a poder ver o perfil do mesmo.

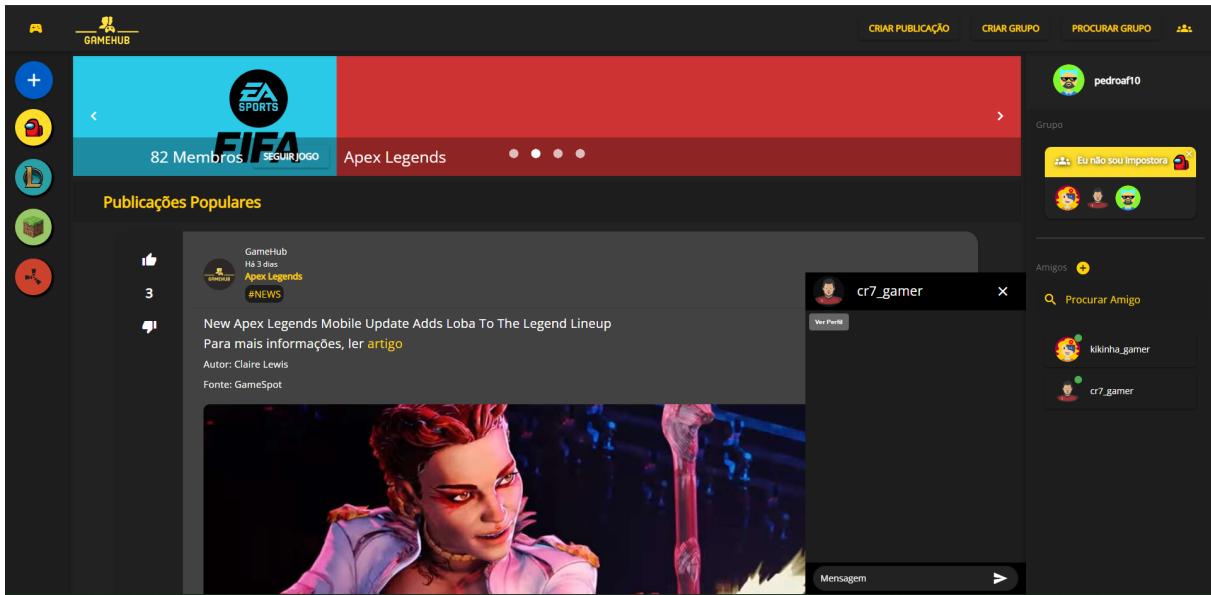


Figura 16.48: Página inicial - Ver perfil a partir do chat

Em seguida o utilizador é reencaminhado para a pagina do perfil do amigo. Nesta página é então possível remover um amigo da sua lista de amigos, carregando simplesmente em remover amigo.

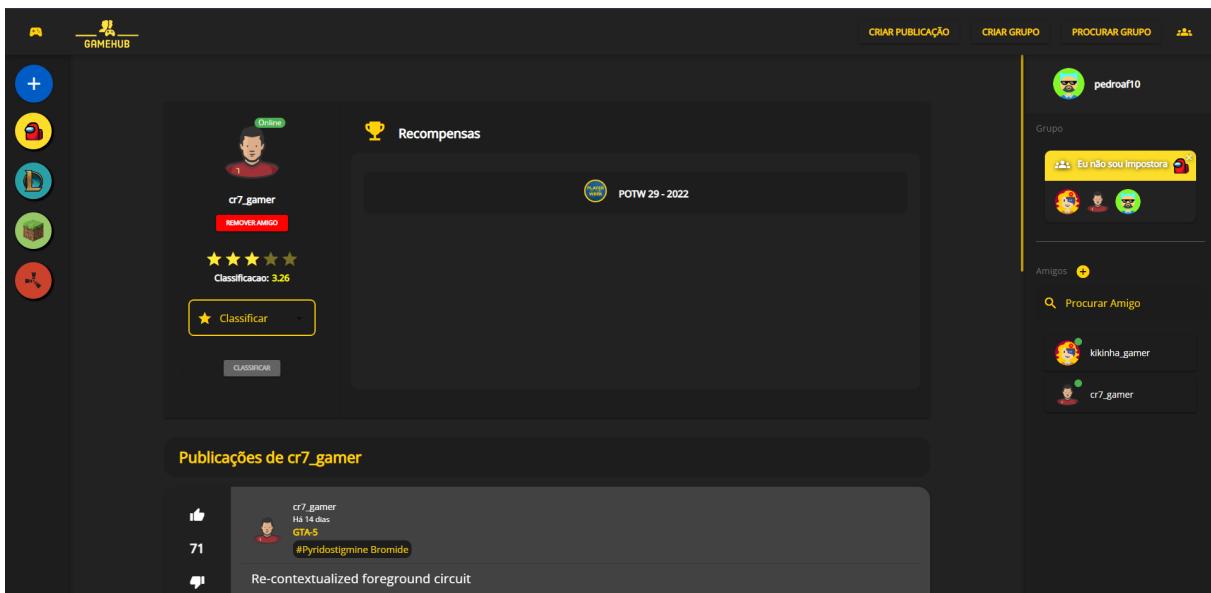


Figura 16.49: Página perfil - Remover amigo

Carregando no botão anterior, o amigo é apagado da lista dos amigos e o botão de remover amigo passa para adicionar amigo.

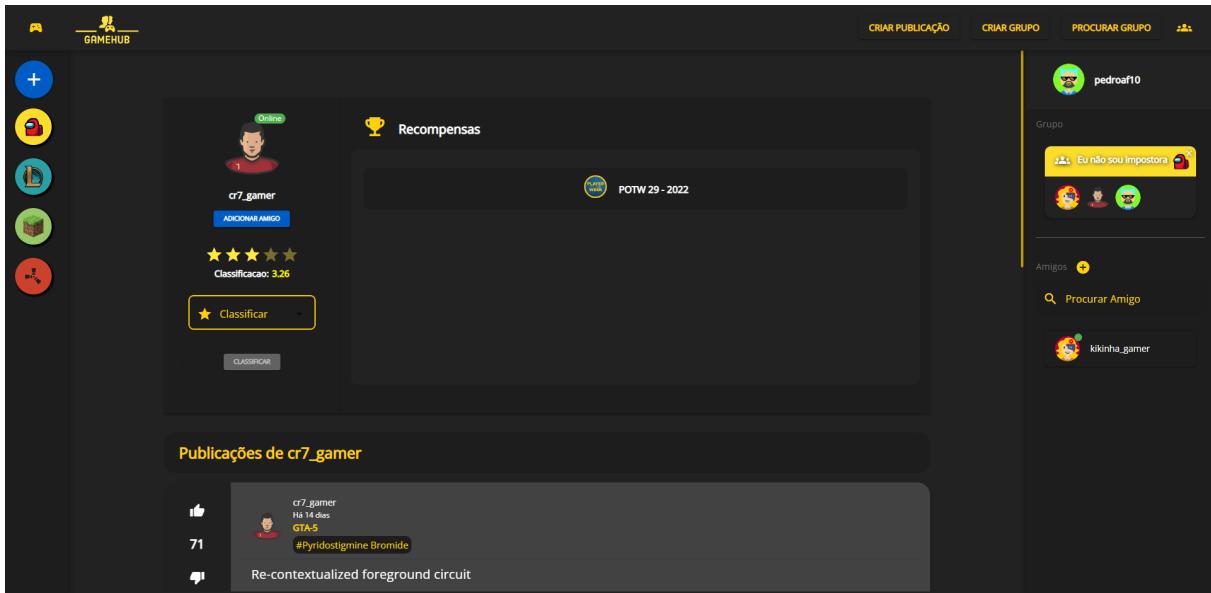


Figura 16.50: Página perfil - Remover amigo

16.1.13 Avaliar jogador

Todos os utilizadores podem avaliar qualquer outro por cada vez que estiveram num grupo juntos, de modo a que cada utilizador vá construindo uma reputação, e assim os outros poderem à partida perceber o tipo de utilizador com o qual vão entrar num grupo. Assim para avaliar um utilizador, é necessário primeiro aceder ao seu perfil, a partir de algum dos métodos já referidos anteriormente.

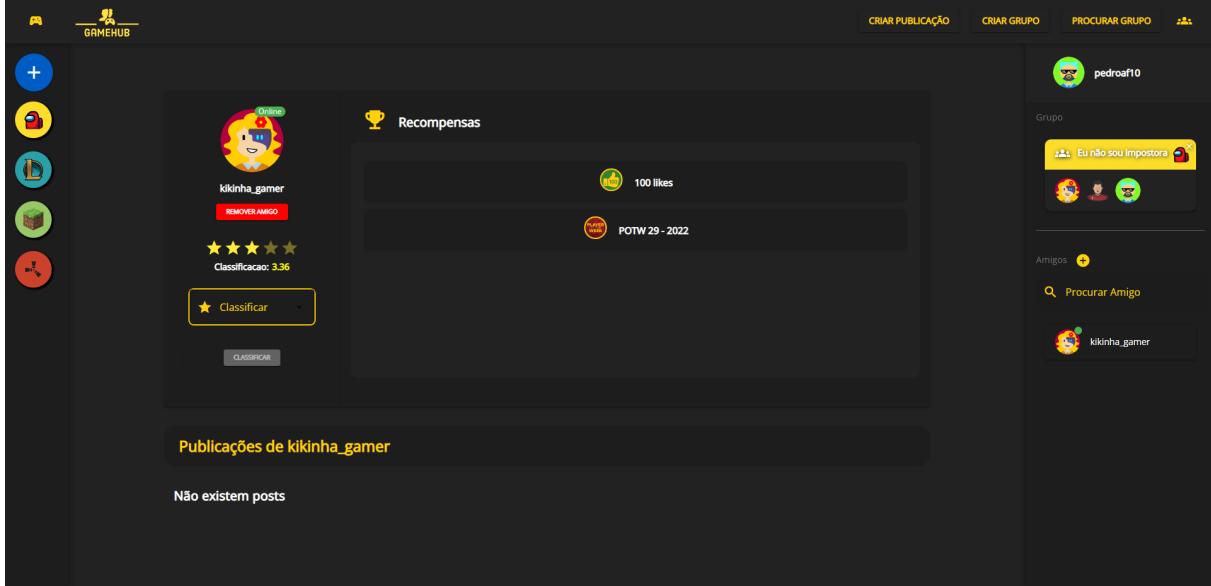


Figura 16.51: Página perfil - Classificar utilizador

Carregando no classificar, irá aparecer uma lista com as notas possíveis de atribuir.

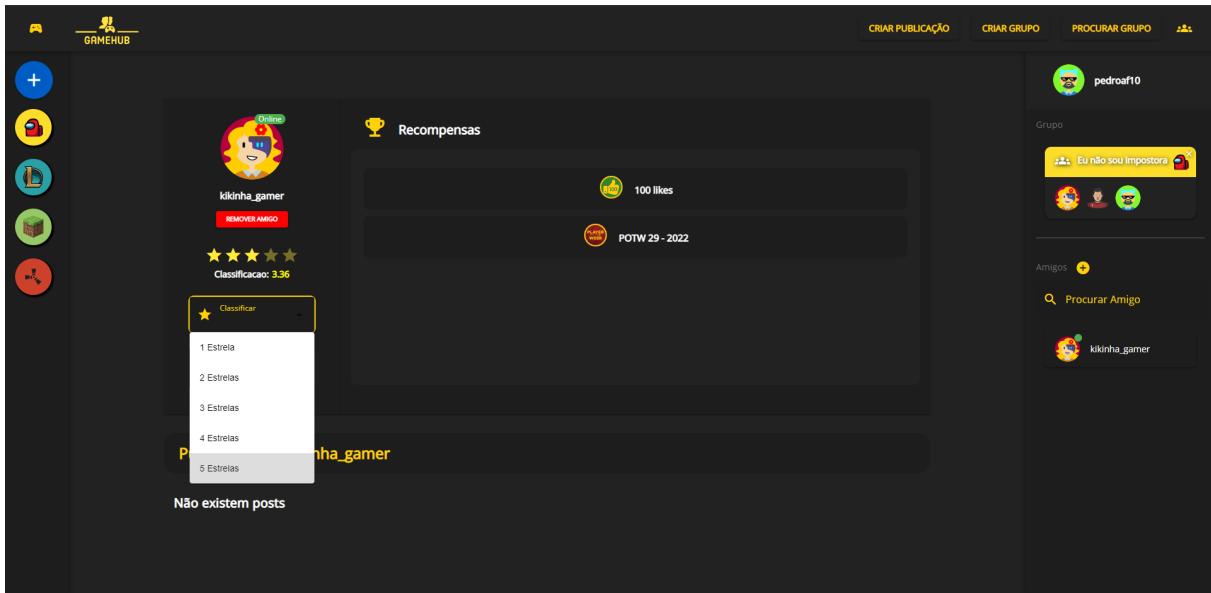


Figura 16.52: Página perfil - Escolher classificação

Após escolhida a classificação resta apenas carregar no botão classificar.

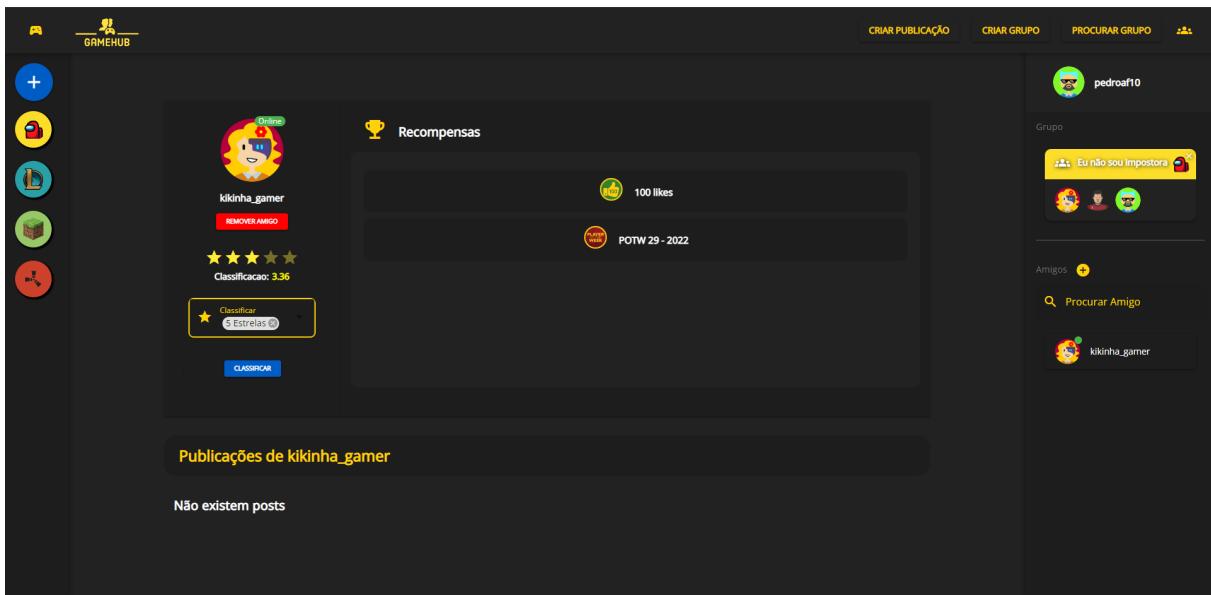


Figura 16.53: Página perfil - Classificação escolhida

Finalmente, a classificação fica bloqueada até voltar a entrar num grupo com esse utilizador, a classificação é alterada e uma mensagem de sucesso é mostrada.

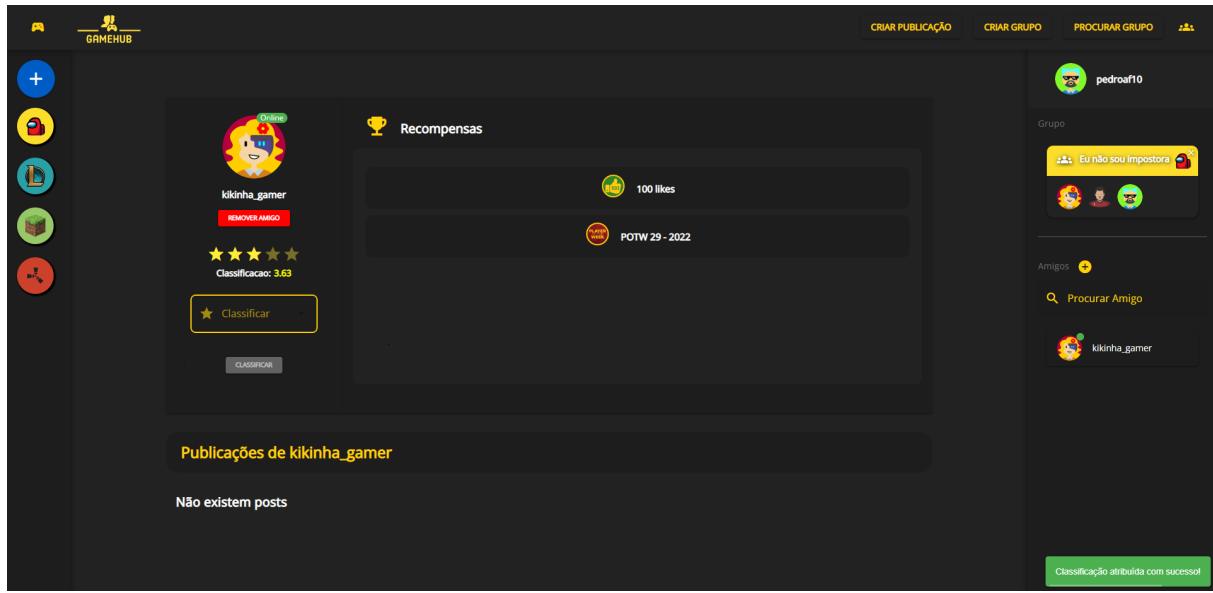


Figura 16.54: Página perfil - Classificação realizada

16.1.14 Receber notificação de mensagem

Quando um utilizador recebe uma mensagem, este recebe uma notificação no canto inferior direito, é reproduzido o som da notificação do GameHub e ainda é adicionado um número de mensagens por ler no amigo que enviou a mensagem.

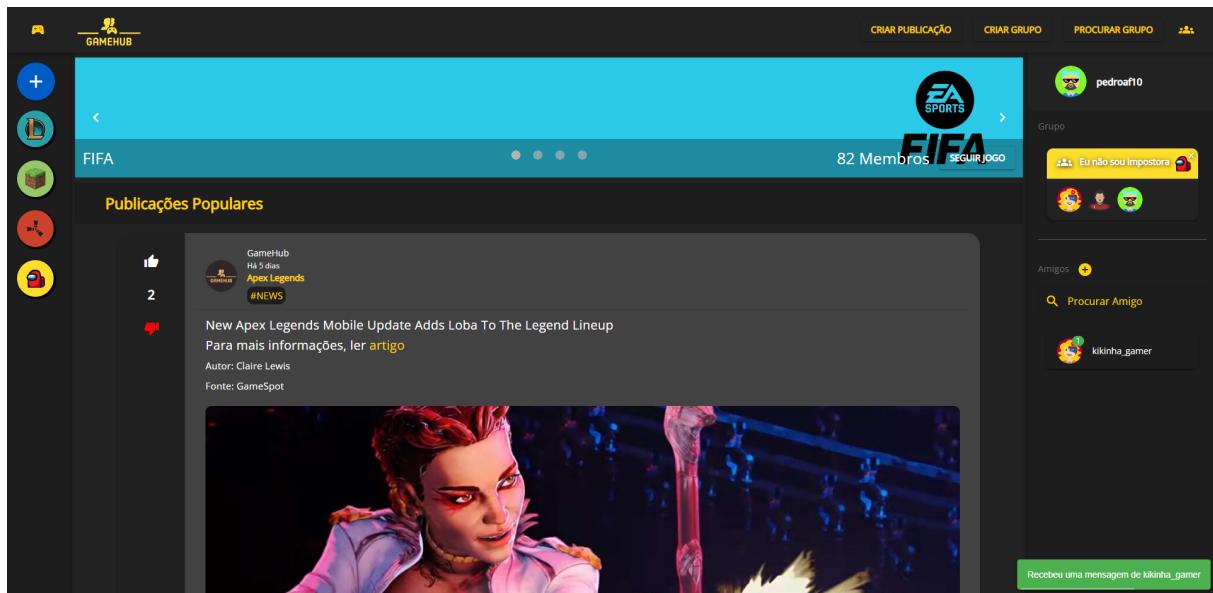


Figura 16.55: Página Inicial - receber notificação

16.1.15 Enviar mensagem privada

Na plataforma é possível não só enviar mensagens para os grupos, como também para amigos de forma privada. Assim primeiro é necessário carregar num amigo da lista de amigos.

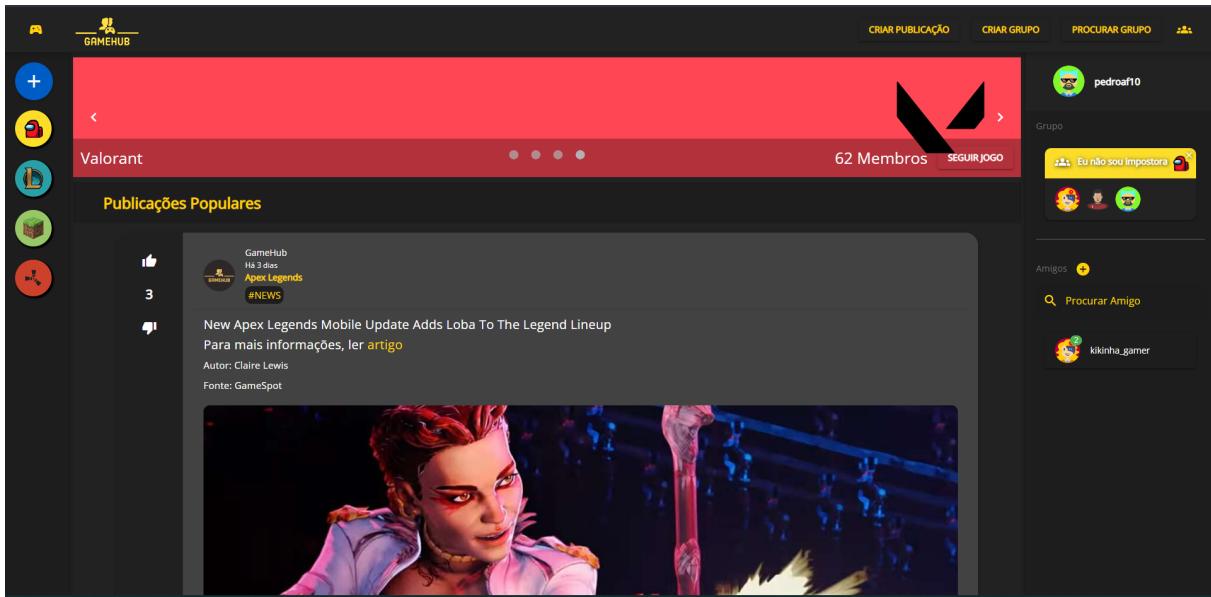


Figura 16.56: Página perfil - Enviar mensagem privada

Após realizada a ação anterior é aberto a conversa entre os dois, podendo então escrever a mensagem desejada.

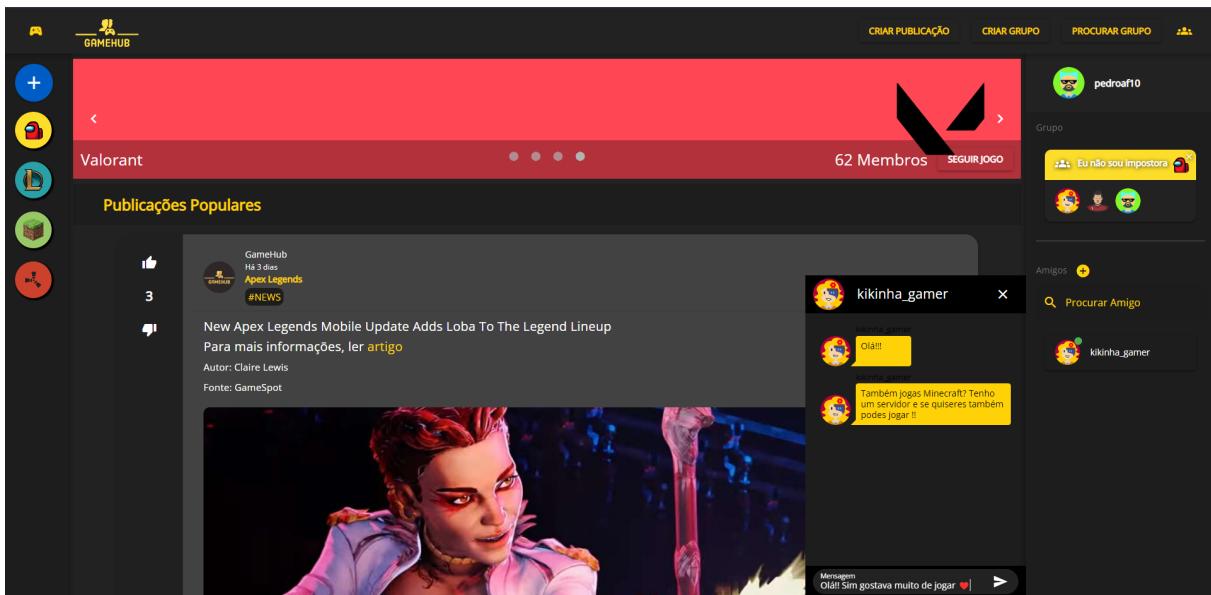


Figura 16.57: Página perfil - Mensagem privada escrita

Escrita agora a mensagem, basta carregar no icon de enviar mensagem, e a mensagem é enviada com sucesso.

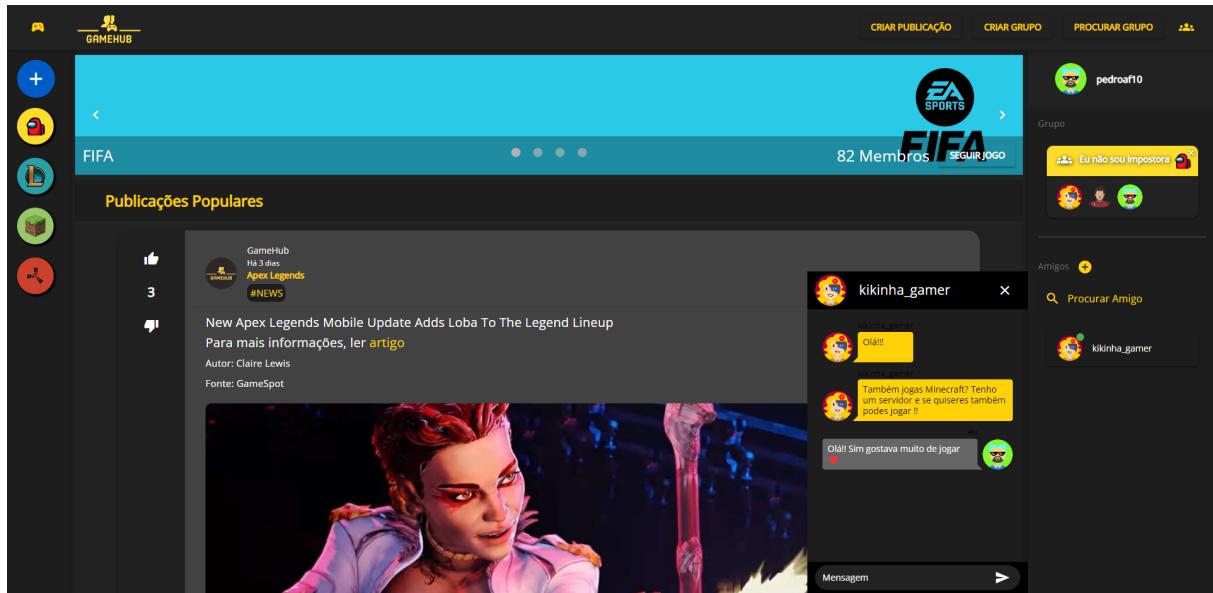


Figura 16.58: Página perfil - Mensagem privada enviada

16.1.16 Criar publicação multimédia

Os utilizadores podem criar publicações nos *feeds* dos jogos que seguem. Estas publicações podem conter apenas texto, ou então contém também imagens, GIFs ou até mesmo vídeos. Para poderem criar, primeiro precisam de carregar em criar publicação.

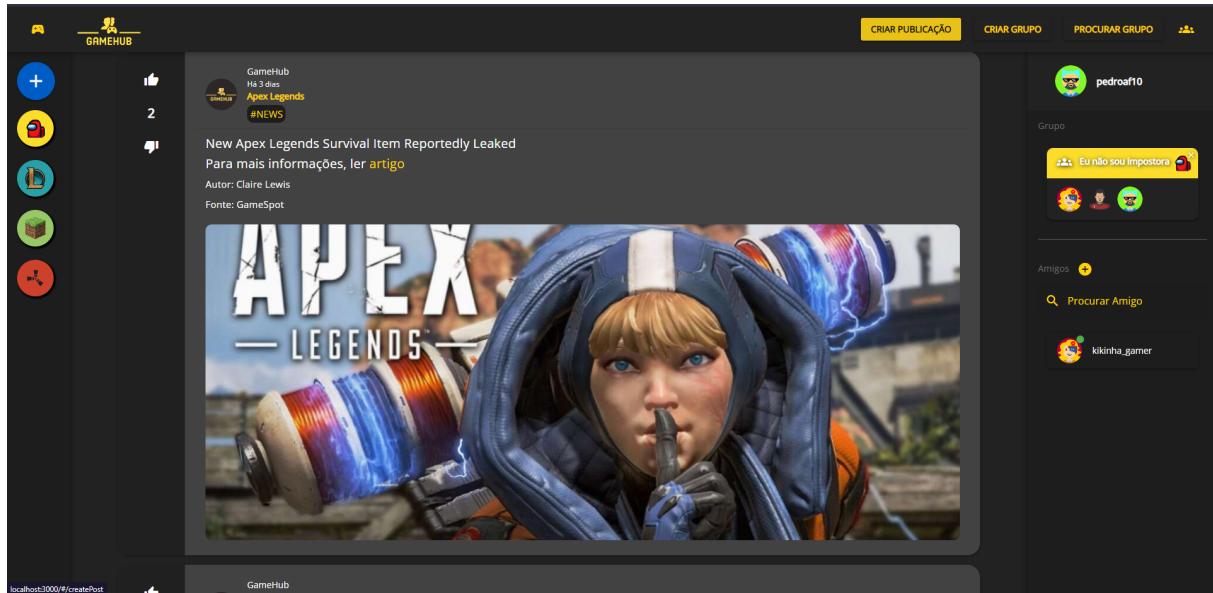


Figura 16.59: Página perfil - Criar publicação multimédia

Carregando no botão anterior, o utilizador é encaminhado para a página de criar publicações, mais concretamente para a secção Imagem/Vídeo. Aqui serão então inseridos os campos para a criação da publicação. Começando pela anexação de um vídeo, é preciso primeiro carregar em adicionar ficheiro.

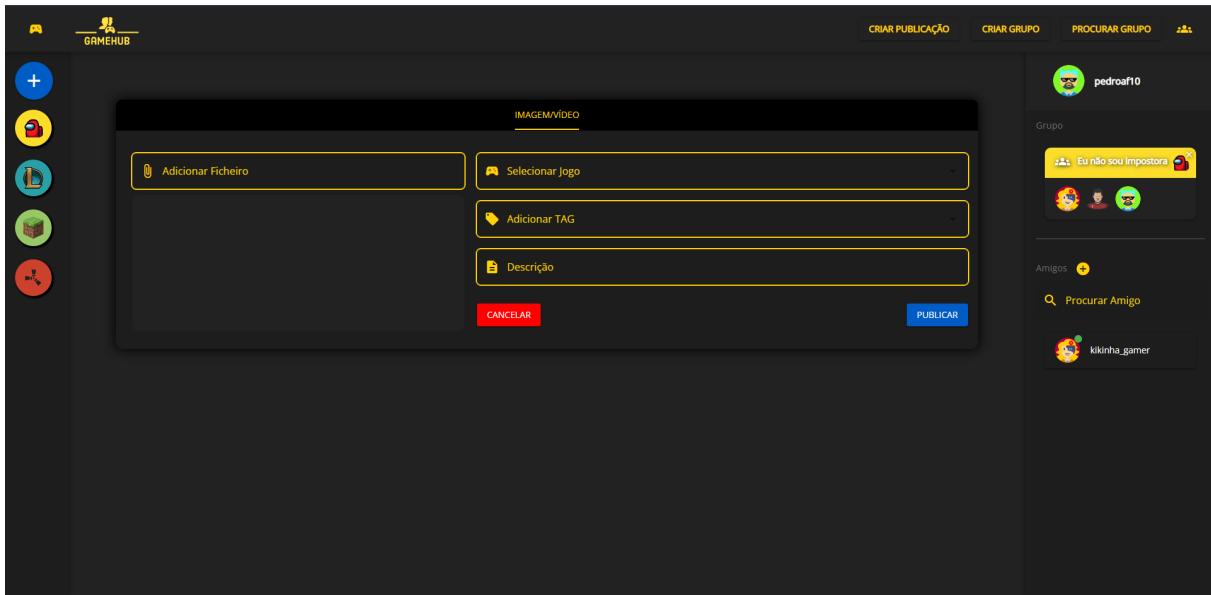


Figura 16.60: Página criar publicação - Imagem/vídeo

Agora é aberto o *explorer* do sistema operativo e pode ser selecionado um ficheiro desde que seja uma imagem, um GIF ou um vídeo.

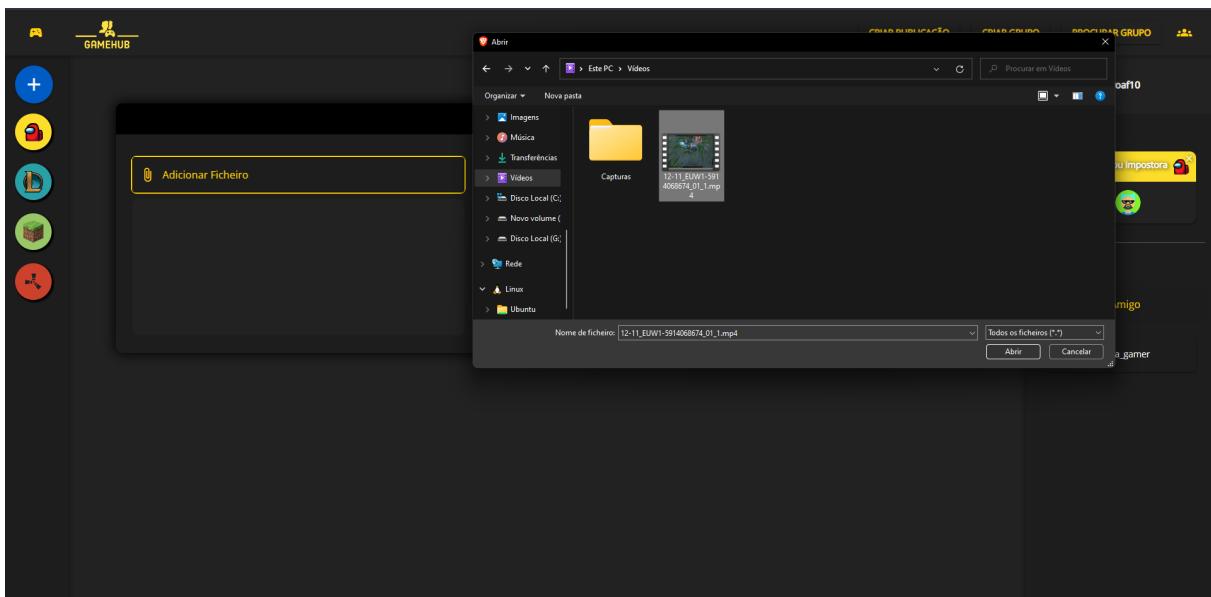


Figura 16.61: Página criar publicação - Selecionar media

Após a adição e preenchimento dos restantes campos, é possível então carregar em publicar, de modo a finalizar a publicação.

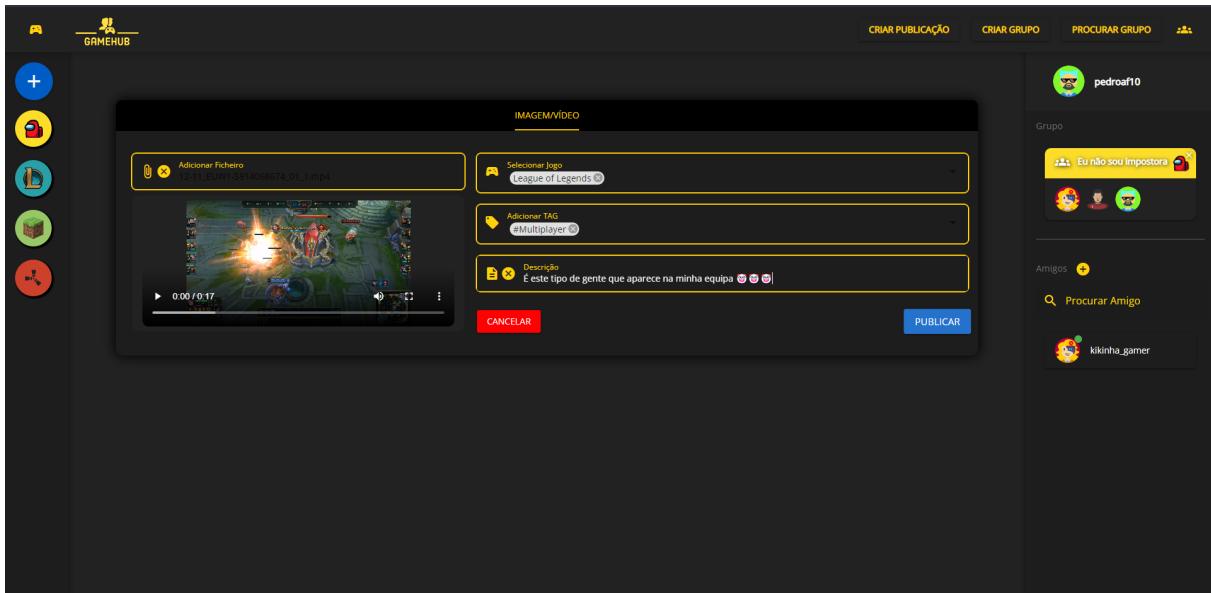


Figura 16.62: Página criar publicação - Campos preenchidos

Após terminada a publicação, esta é colocado no *feed* do jogo correspondente, redireciona o utilizador para o mesmo e ainda fornece ao utilizador uma mensagem de sucesso com a opção de remover imediatamente a publicação.

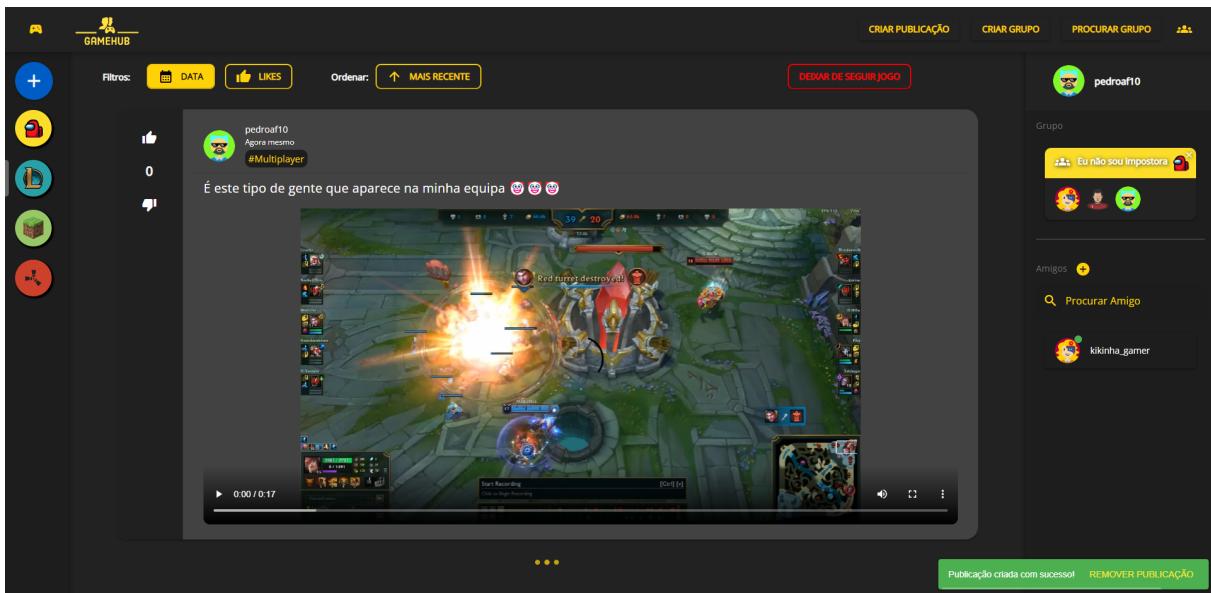


Figura 16.63: Página *feed* jogo - publicação realizada

Em casos em que não são introduzidos todos os campos necessários, o utilizador depara-se com uma mensagem de erro assim como um alerta para os campos obrigatórios para uma publicação.

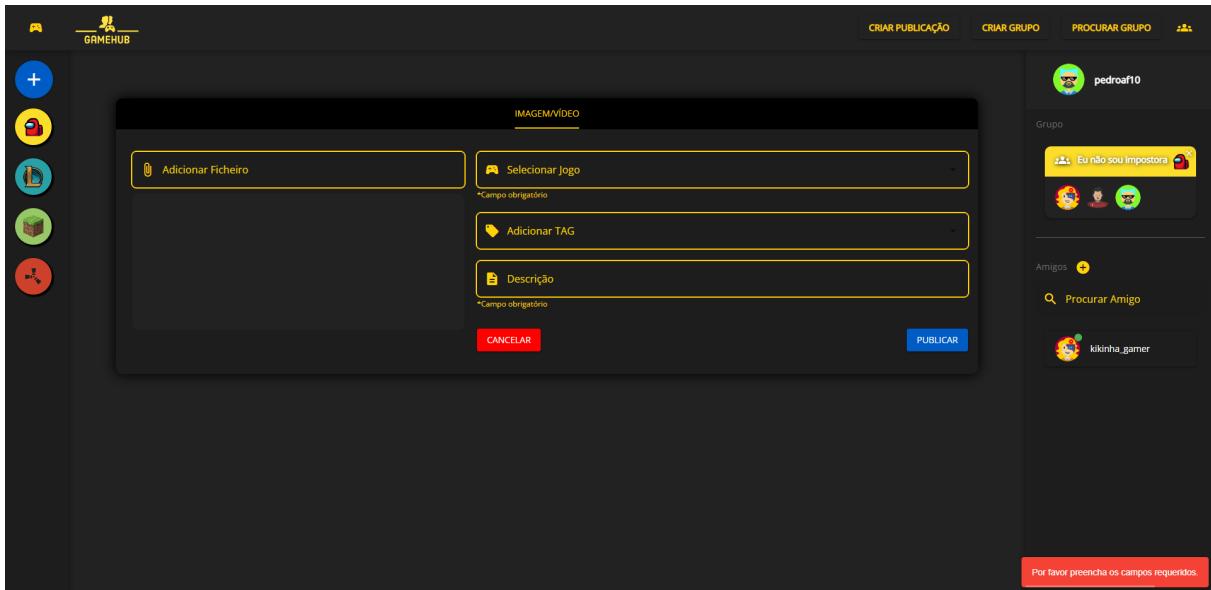


Figura 16.64: Página criar publicação - Campos incorretos

16.1.17 Eliminar publicação multimédia

Um utilizador em qualquer altura pode querer remover uma publicação realizada. Assim primeiro precisa aceder ao seu perfil. Para isso precisa de carregar no seu avatar.

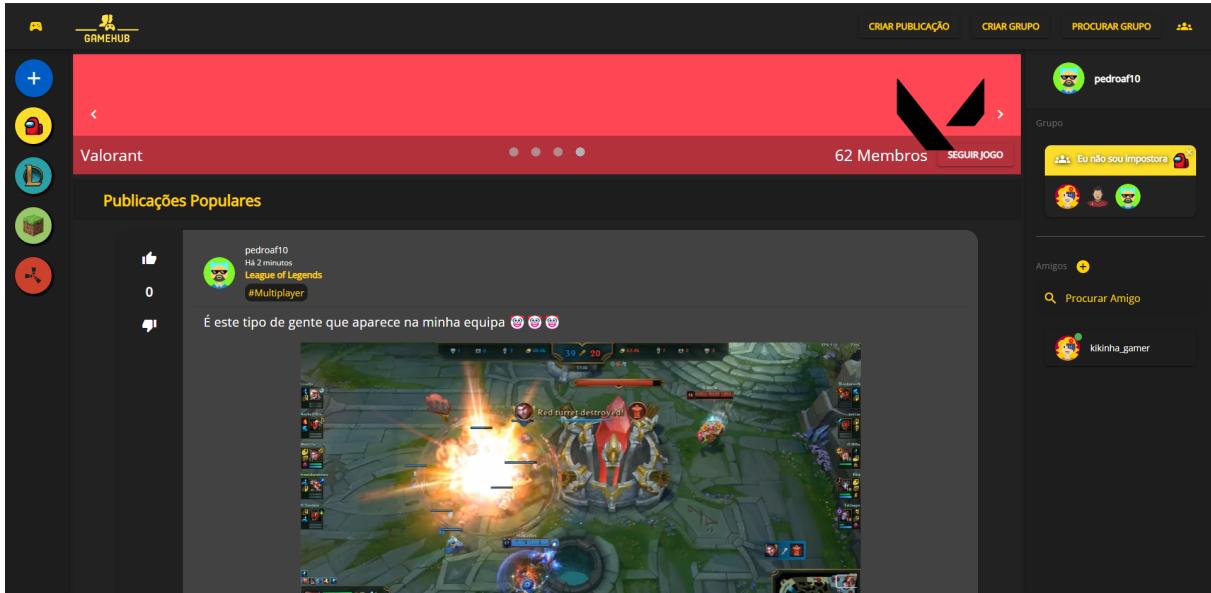


Figura 16.65: Página inicial - Ver próprio perfil

Em seguida precisa de carregar em Ver perfil de modo a completar a sua viagem até ao seu perfil.

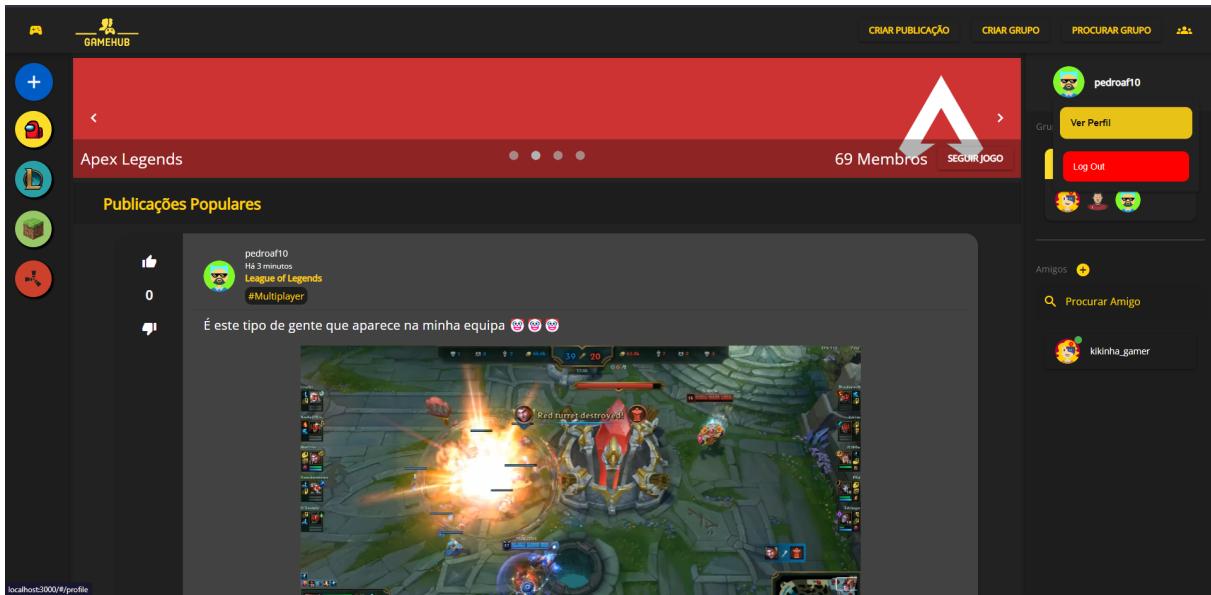


Figura 16.66: Página inicial - Ver próprio perfil

Dentro da página do perfil, existe em baixo uma aba Meus Posts, aí aparecem por ordem cronológica as publicações realizadas pelo utilizador. Realizando um *scroll-down* é possível encontrar a publicação a remover.

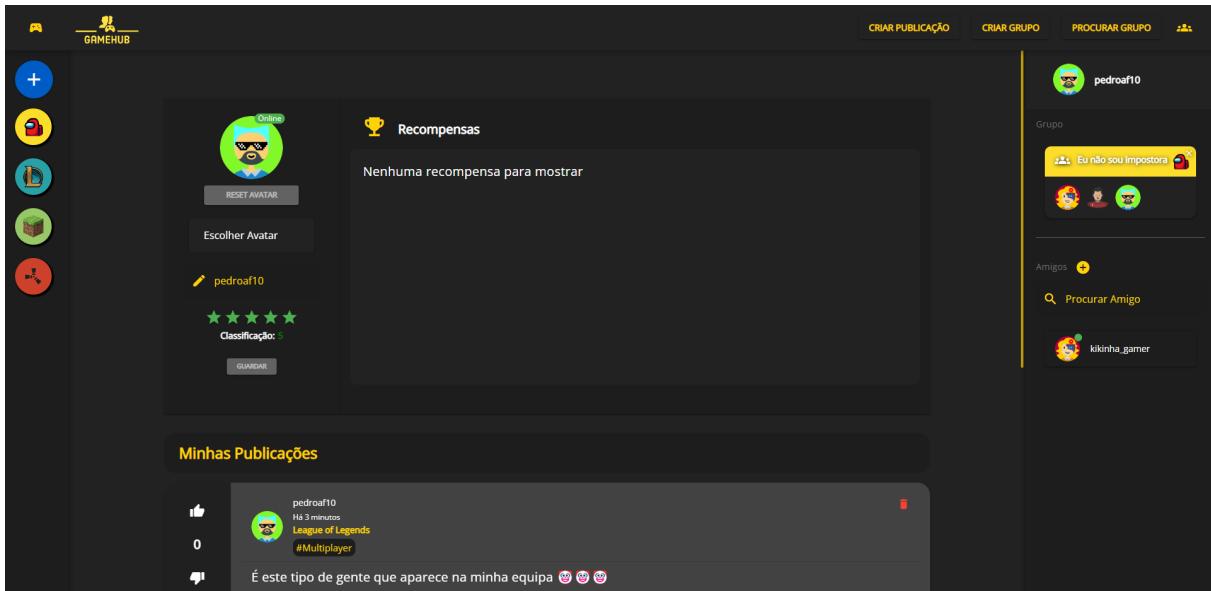


Figura 16.67: Página do perfil

Encontrado então a publicação, é necessário então carregar no caixote do lixo.

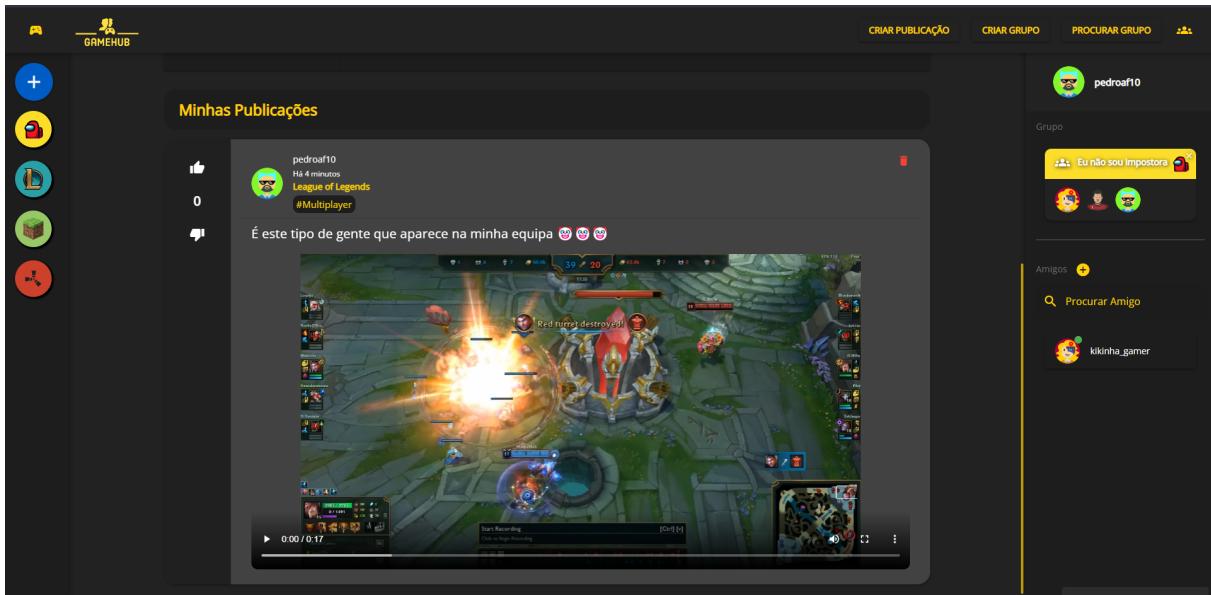


Figura 16.68: Página do perfil - secção meus *posts*

Em seguida é necessário carregar em Remover Post.

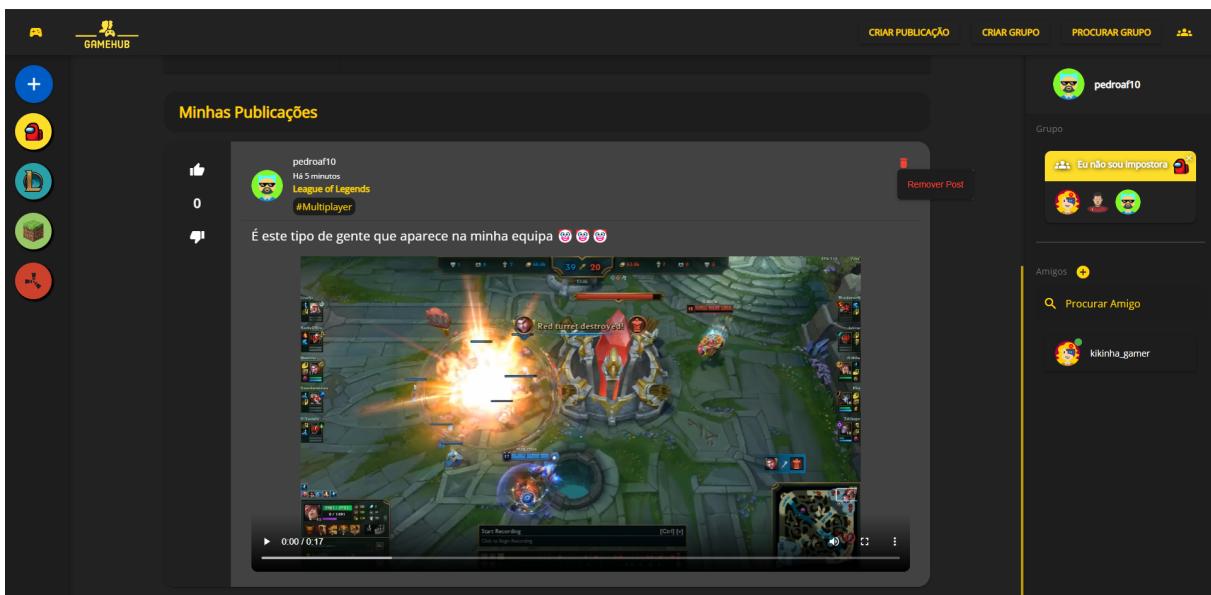


Figura 16.69: Página do perfil - botão do caixote selecionado

Carregado o botão de Remover Post, falta agora confirmar a remoção da publicação.

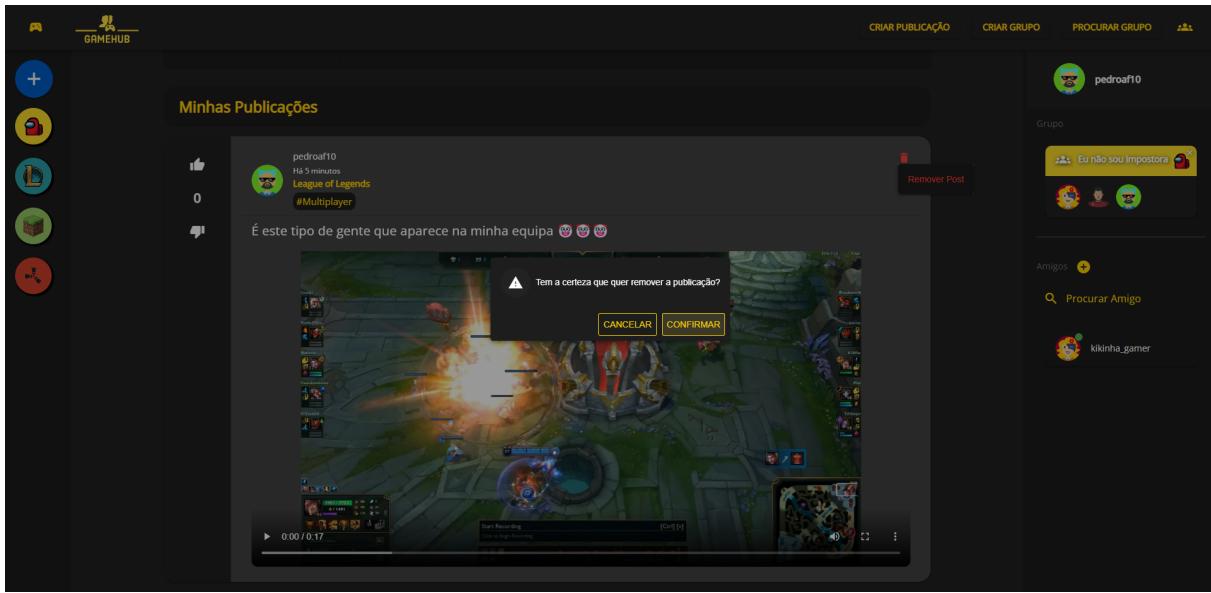


Figura 16.70: Página do perfil - confirmar eliminação

Finalmente a publicação é eliminada, desaparecendo da aba *Meus Posts* e do *feed* em que se encontrava.

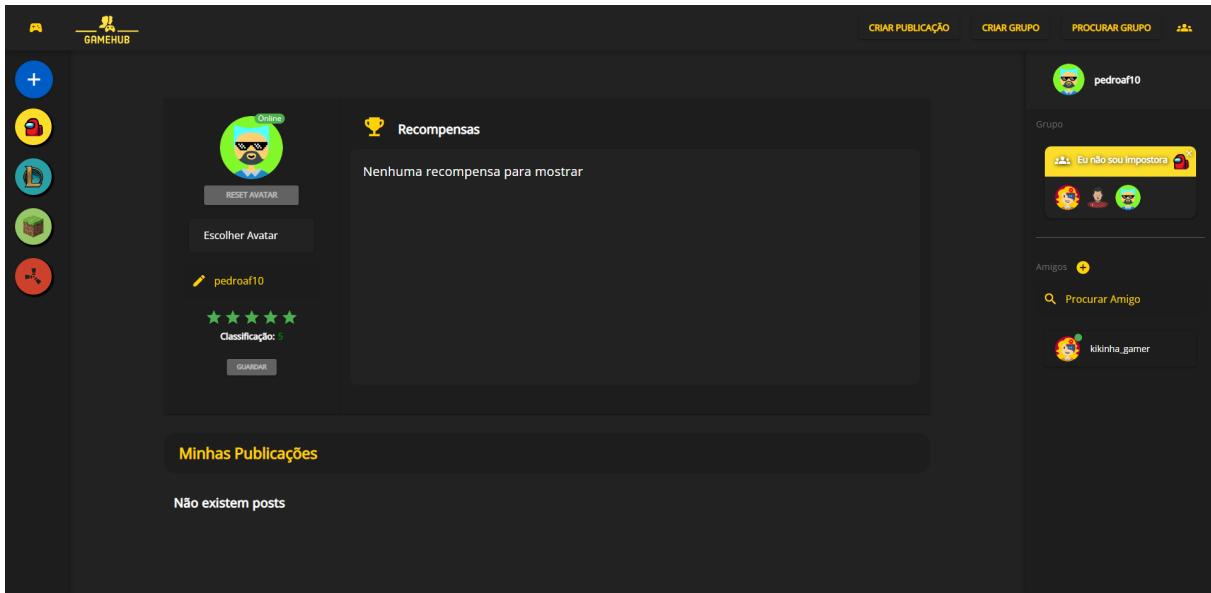


Figura 16.71: Página do perfil - publicação eliminada

16.1.18 Avaliar publicação

Um utilizador pode dar o seu *feedback* relativamente a uma dada publicação. Assim encontrando-se em qualquer página onde existem publicações é possível proceder à sua avaliação.

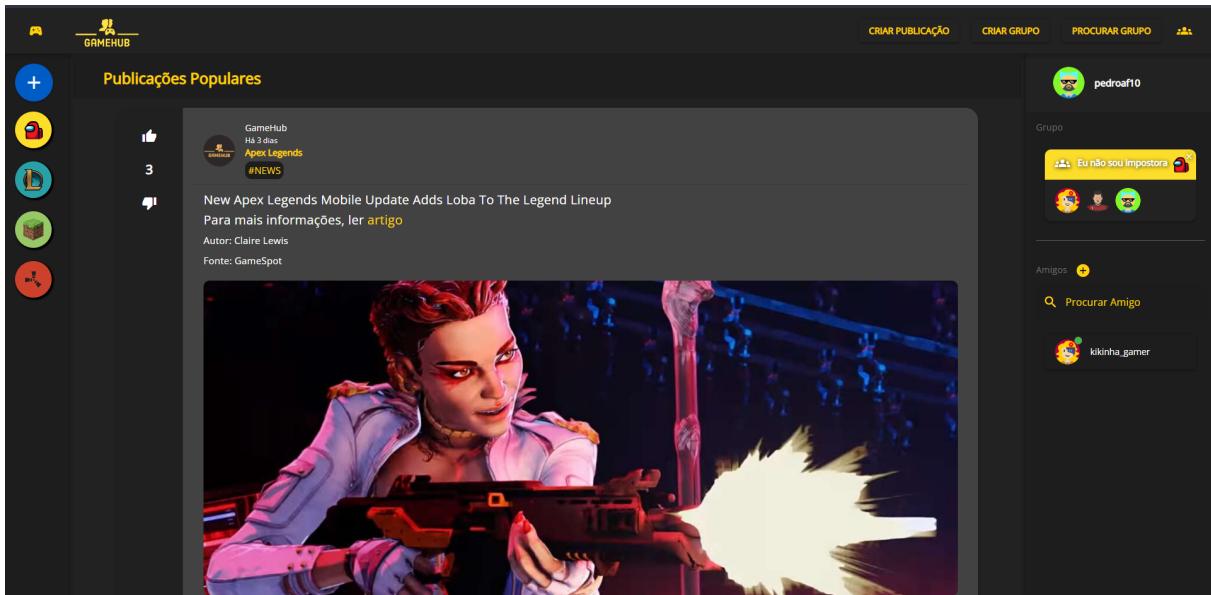


Figura 16.72: Página inicial - publicação sem avaliação

Carregando no botão de *like*, este vai passar a verde, o número de *likes* total da publicação irá aumentar e ainda o sistema vai fornecer uma mensagem de avaliação bem sucedida.

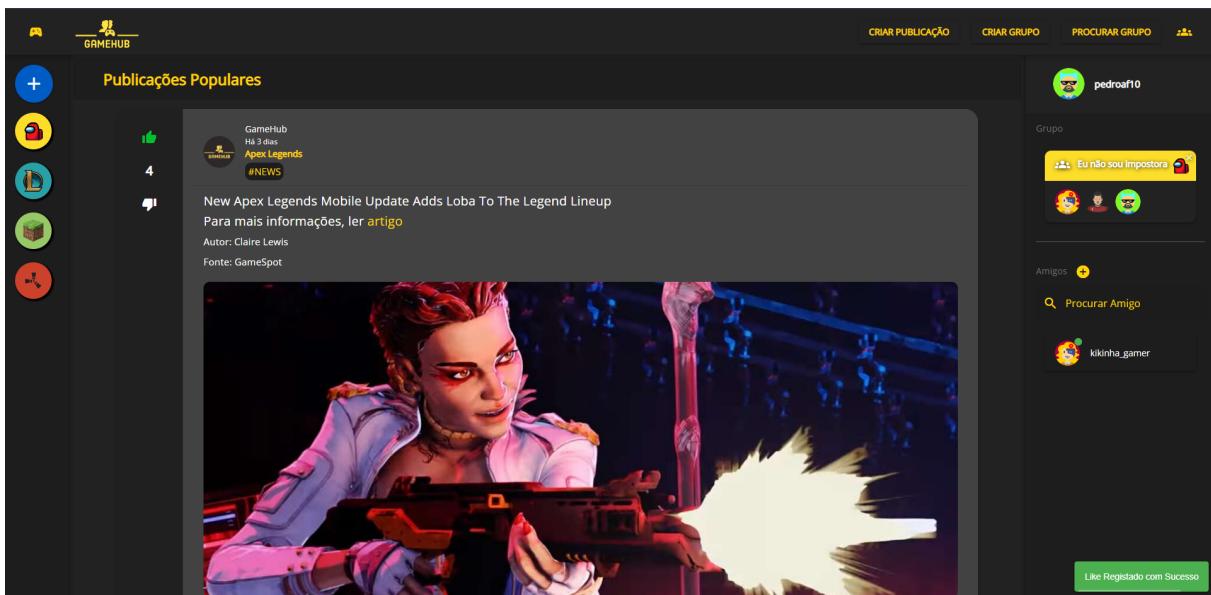


Figura 16.73: Página inicial - publicação com *like*

No caso de carregar no botão de *dislike*, este vai passar a vermelho, o número de *likes* total da publicação irá diminuir e uma mensagem de avaliação bem sucedida irá ser apresentada.

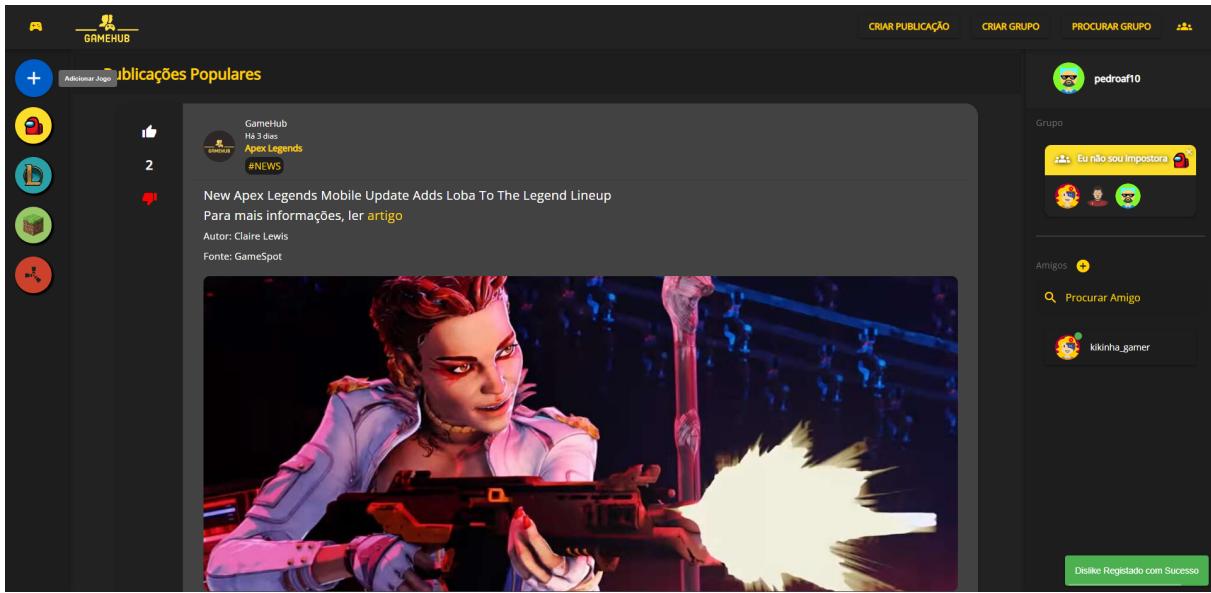


Figura 16.74: Página inicial - publicação com *dislike*

Pode ainda ser visualizada dāa publicação na sua plenitude simplesmente carregando na imagem.

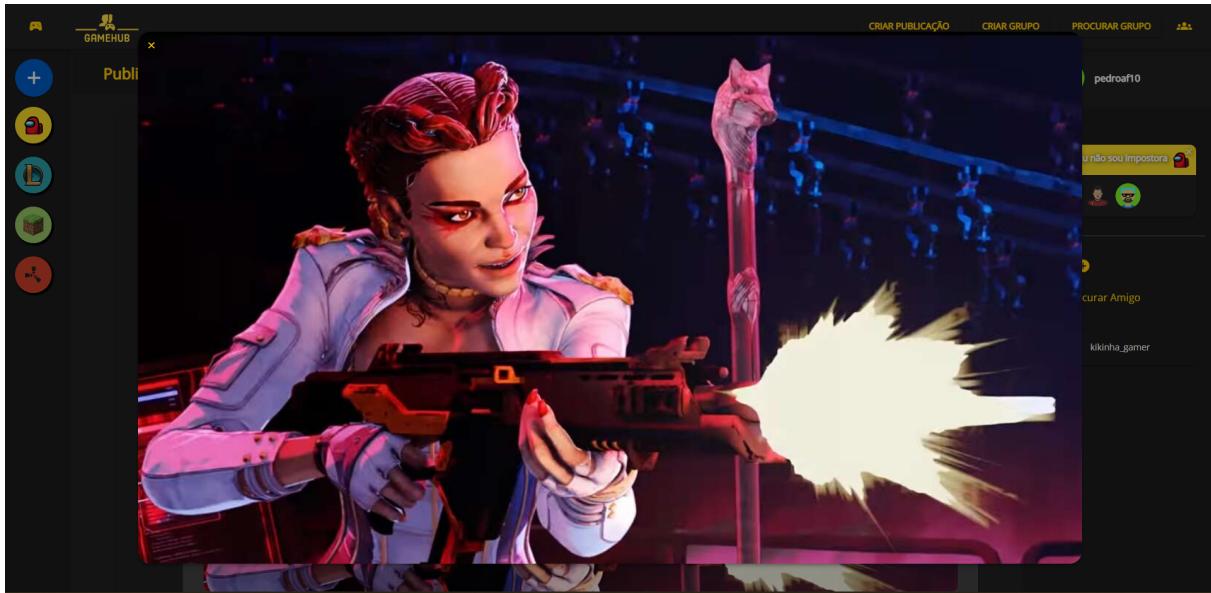


Figura 16.75: Página inicial - imagem da publicação

16.1.19 Logout

A última funcionalidade do GameHub é terminar a sessão do utilizador. Para tal primeiro é necessário carregar no avatar à direita.

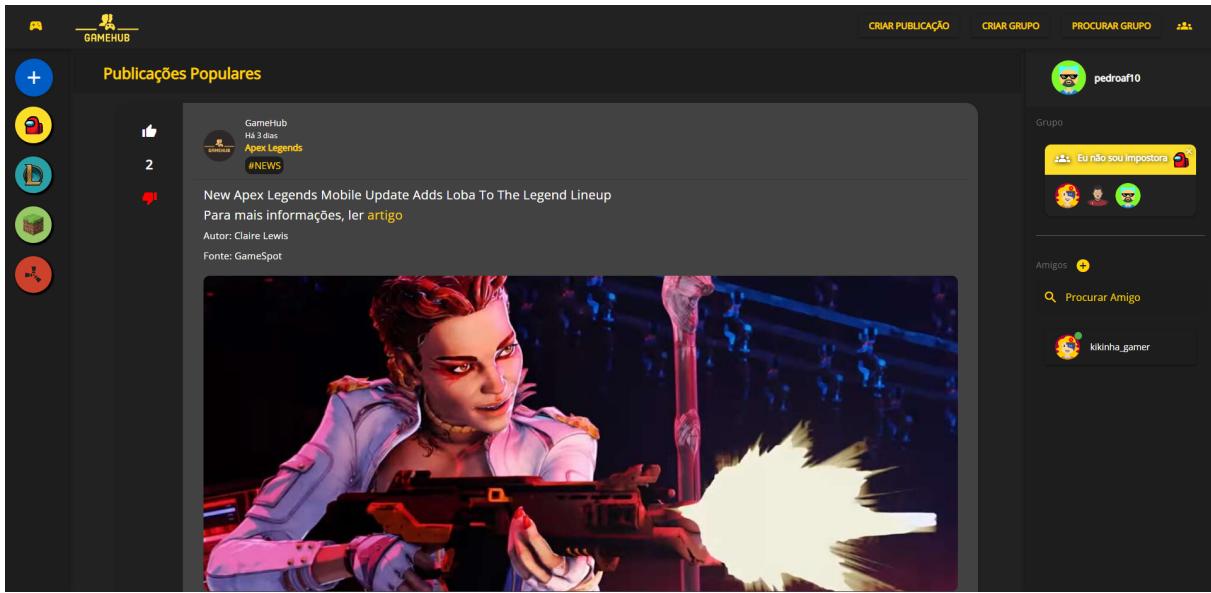


Figura 16.76: Página inicial - carregar no avatar para terminar sessão

Carregar em *Log Out* para terminar a sessão.

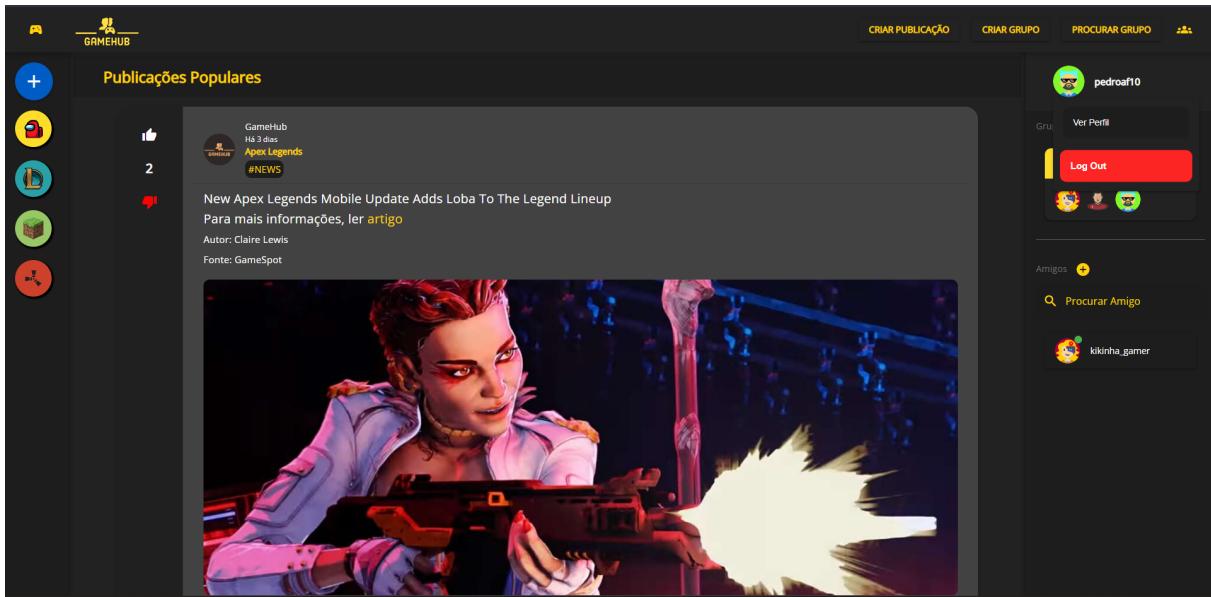


Figura 16.77: Página inicial - carregar em *logout*

Após terminar sessão, o utilizador é encaminhado de novo para a página de *login*.



Figura 16.78: Página *Login* - sessão terminada

16.2 Administrador

Na aplicação GameHub existe um segundo utilizador, o administrador. Este, com o objetivo de manter uma boa atmosfera entre os utilizadores da plataforma, podem eliminar publicações, grupos ou até mesmo banir jogadores, caso as normas estabelecidas sejam violadas.

16.2.1 Eliminar publicação multimédia

Existindo uma publicação multimédia que viole as normas do GameHub, será então necessário proceder à sua remoção. Assim primeiro é necessário selecionar a aba Publicações Multimédia.

Publicações Multimedia							
Identificador	Nome do utilizador	Data de inicio	Data final				
REMOVER	1	GTA-5	Configurable 24 hour open architecture	2022-03-07T02:46:27	Ver	35	bworstally
REMOVER	2	GTA-5	Progressive analyzing application	2022-05-24T10:00:49	Ver	63	skleinpelz1q
REMOVER	3	GTA-5	Stand-alone analyzing forecast	2022-03-14T00:40:32	Ver	13	pdigglesc
REMOVER	4	GTA-5	Ameliorated interactive circuit	2021-10-08T10:16:05	Ver	63	skleinpelz1q
REMOVER	5	GTA-5	Decentralized fault-tolerant frame	2022-03-18T21:54:45	Ver	53	dthinksen1g

Figura 16.79: Página admin publicações multimédia

Em seguida podem ser inseridos múltiplos filtros para ajudar na procura da publicação.

Publicações Multimedia							
Identificador	Nome do utilizador	Data de inicio	Data final				
REMOVER	1	GTA-5	Configurable 24 hour open architecture	2022-03-07T02:46:27	Ver	35	bworstally
REMOVER	11	GTA-5	Future-proofed secondary info-mediaries	2022-05-17T00:11:38	Ver	35	bworstally
Records per page: 5 ▾ 1-2 of 2							

Figura 16.80: Página admin publicações multimédia - filtros

Carregando no botão ver, é aberta uma nova janela com a media existente na publicação.



Figura 16.81: Página admin publicações multimédia - multimédia

Selecionando a opção remover é então pedida uma confirmação ao administrador para concluir a tarefa.

Publicações Multimedia							
Identificador	Nome do utilizador	Data de inicio	Data final				
REMOVER	1	GTA-5	Configurable 24 hour open architecture	2022-03-07T02:46:27	Ver	35	bworstally
REMOVER	11	GTA-5	Future-proofed secondary info-mediaries	2022-05-17T00:11:38	Ver	35	bworstally
Records per page: 5 ▾ 1-2 of 2							

Tem certeza que pretende remover?

[NÃO REMOVER](#) [REMOVER](#)

Figura 16.82: Página admin publicações multimédia - confirmação

Aceitando, a publicação multimédia é removida da plataforma.

The screenshot shows the 'Publicações Multimedia' (Multimedia Publications) section of the admin interface. At the top, there are search fields for 'Identificador' (Identifier), 'Nome do utilizador' (User Name), 'Data de inicio' (Start Date), and 'Data final' (End Date). Below the search bar is a table with columns: 'Identificador' (Identifier), 'Nome do Jogo' (Game Name), 'Descrição' (Description), 'Data de publicação' (Publication Date), 'Link da media' (Media Link), 'Identificador do gamer' (Gamer Identifier), and 'Nome do gamer' (Gamer Name). A single row is visible, representing a post by user 'bworstally' with identifier 11, titled 'Future-proofed secondary info-mediaries'. A red 'REMOVER' (Remove) button is present in the first column of the table. At the bottom right, it says 'Records per page: 5 ▾ 1-1 of 1'.

Figura 16.83: Página admin publicações multimédia - publicação removida

16.2.2 Eliminar publicação grupo

Para eliminar uma publicação de grupo é primeiro necessário escolher a aba Publicações de Grupo.

The screenshot shows the 'Publicações Grupo' (Group Publications) section of the admin interface. It features the same search and filter options as the previous screen. The table lists ten posts, each with a red 'REMOVER' button in the first column. The posts are as follows:

Identificador	Nome do Jogo	Descrição	Data de publicação	Identificador do gamer	Nome do gamer
2	GTA-5	Compatible full-range hierarchy	2021-07-28T22:22:44	77	klynthal24
3	GTA-5	Face to face zero defect methodology	2021-07-28T06:21:25	57	bbolare1k
4	GTA-5	Phased 4th generation pricing structure	2022-04-10T22:11:59	12	cfawlob
5	GTA-5	Seamless background framework	2021-05-30T01:50:59	39	jhayball12
6	GTA-5	Managed directional system engine	2021-06-04T21:23:41	99	akolodziej2q
8	GTA-5	Proactive interactive hardware	2022-04-15T14:07:48	78	bkopfer25
11	GTA-5	Fully-configurable heuristic focus group	2021-06-30T18:46:03	10	wsolan9
13	GTA-5	Persevering upward-trending protocol	2021-06-20T15:49:01	57	bbolare1k
14	GTA-5	Decentralized responsive middleware	2021-09-28T22:23:35	24	goglevien
18	GTA-5	Right-sized bandwidth-monitored database	2021-10-11T08:51:53	31	cbernardoniu

At the bottom right, it says 'Records per page: 10 ▾ 1-10 of 11 < >'.

Figura 16.84: Página admin publicações grupo

Podem tal como no caso das publicações multimédia, ser adicionados os filtros indicados em cima para facilitar a procura da publicação de grupo infratora.

This screenshot shows the same 'Publicações Grupo' page but with different filter settings at the top: '31/03/2022' to '12/06/2022'. The table now displays only two posts, both of which have been removed:

Identificador	Nome do Jogo	Descrição	Data de publicação	Identificador do gamer	Nome do gamer
4	GTA-5	Phased 4th generation pricing structure	2022-04-10T22:11:59	12	cfawlob
8	GTA-5	Proactive interactive hardware	2022-04-15T14:07:48	78	bkopfer25

At the bottom right, it says 'Records per page: 10 ▾ 1-2 of 2'.

Figura 16.85: Página admin publicações grupo - filtrada

Em seguida deve ser selecionada a opção remover, na publicação desejada.

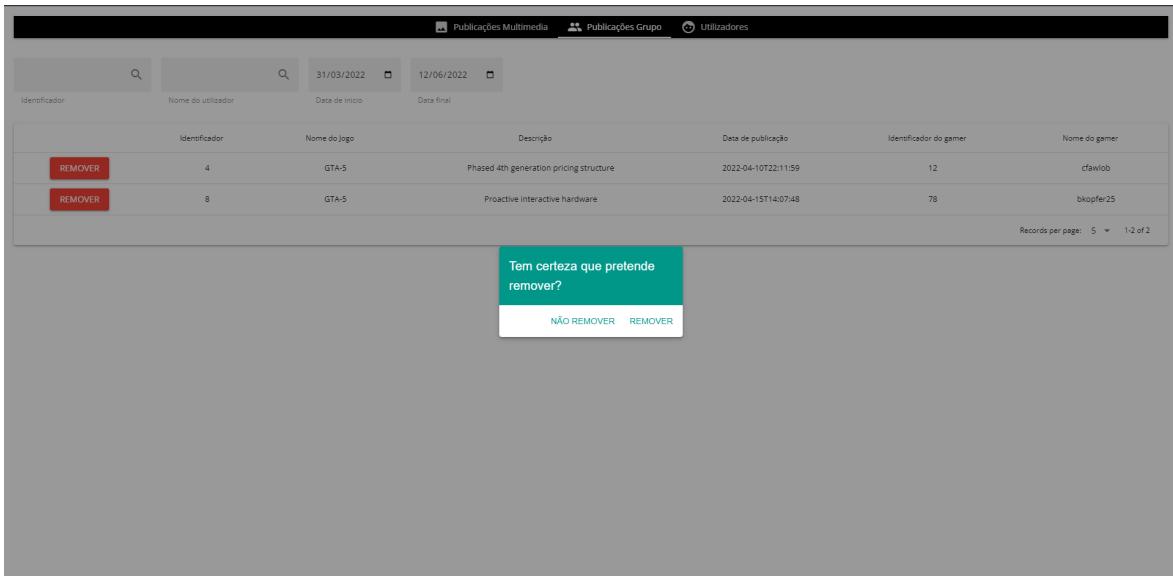


Figura 16.86: Página admin publicações grupo - confirmação

Finalmente basta aceitar a confirmação de remoção, e a publicação é eliminada.



Figura 16.87: Página admin publicações grupo - publicação eliminada

16.2.3 Banir utilizador

A ultima opção do administrador é banir por completo um utilizador, eliminando com ele toda a sua interferência na plataforma. Para tal, primeiro é necessário escolher a aba utilizadores.

Utilizadores				
Identificador	Nome do Utilizador	Nome do Gamer	Classificação	Avatar
BANIR	1	GameHub	3	<button>Ver</button>
BANIR	2	ntrrey1	1.61	<button>Ver</button>
BANIR	3	efrye2	1.16	<button>Ver</button>
BANIR	4	alienney3	1.38	<button>Ver</button>
BANIR	5	pmccafferly4	2.42	<button>Ver</button>
BANIR	6	tconcannon5	4.33	<button>Ver</button>
BANIR	7	mstrugnell6	4.96	<button>Ver</button>
BANIR	8	gfratson7	2.07	<button>Ver</button>
BANIR	9	mcausnett8	1.87	<button>Ver</button>
BANIR	10	wsolan9	4.93	<button>Ver</button>

Figura 16.88: Página admin banir utilizador

Em seguida pode ser usado o filtro para selecionar o utilizador sem haver a necessidade de procurar na lista toda.

The screenshot shows a search interface with two input fields: 'Identificador' and 'Nome do utilizador', both containing 'pedroaf10'. Below the search bar is a table with columns: 'Identificador', 'Nome do Gamer', 'Classificação', and 'Avatar'. A single row is visible, showing '101' in the 'Identificador' column, 'pedroaf10' in the 'Nome do Gamer' column, a green classification icon in the 'Classificação' column, and a small profile picture in the 'Avatar' column. A red 'BANIR' button is located at the top left of the table row. At the bottom right of the table, there are buttons for 'Records per page: 10' and '1-1 of 1'.

Figura 16.89: Página admin banir utilizador - filtro

Após encontrado o utilizador pretendido é agora necessário carregar em banir.

This screenshot is similar to Figure 16.89, showing the user search results table. However, a modal dialog box has appeared in the center of the screen. The dialog box has a teal header with the text 'Tem certeza que pretende remover?'. At the bottom, there are two buttons: 'NÃO REMOVER' and 'REMOVER'. The rest of the page is dimmed to indicate the modal is active.

Figura 16.90: Página admin banir utilizador - confirmação

Aceitada a confirmação para banir o utilizador, este desaparece da plataforma.

The screenshot shows the same search interface and table as before. The table now displays a message 'No data available' instead of the user data. The red 'BANIR' button is still present at the top left of the table area.

Figura 16.91: Página admin banir utilizador - remoção

17 Análise de usabilidade

Estudos provaram que alta usabilidade e uma boa experiência de utilização, pode influenciar os utilizadores a escolherem um produto. Assim, um produto deve ser fácil para um utilizador aprender, usar e executar as suas funções facilmente. Existem inúmeros atributos que podem definir a usabilidade de um produto - *learnability, efficiency, memorability, satisfaction, accessibility*, etc. - contudo, estes podem ser resumidos em: permitir ao utilizador realizar tarefas usando o sistema, com o menor consumo de recursos e com a maior satisfação possível.

Desta forma, os testes de usabilidade são um processo importante durante e após o desenvolvimento do sistema. Os testes durante o desenvolvimento permitem criar protótipos otimizados com previsões de possíveis *feedbacks* de outras pessoas. Enquanto, os testes com o sistema já desenvolvido permitem avaliar as dificuldades atuais dos utilizadores.

Nesta secção serão apresentados os testes de usabilidade realizados no sistema. Para avaliar a usabilidade realizou-se os seguintes testes: *Cognitive Walkthrough*, um questionário SUS - *System Usability Scale* e o método de *think aloud*.

A estratégia de *Cognitive Walkthrough*, sendo um método de inspecção realizou-se após a elaboração das *mockups* apresentadas anteriormente. Este método permitiu detetar os possíveis problemas com que os utilizadores finais poderiam-se deparar.

Por fim, após o desenvolvimento do sistema, recorreu-se à realização de um questionário SUS e ao método de *think aloud*, com alguns dos utilizadores do sistema. De seguida serão apresentados os resultados de cada um dos testes realizados.

17.1 *Cognitive Walkthrough*

De modo a responder à pergunta: "Até que ponto vai este sistema guiar um utilizador não treinado na execução de uma tarefa?", foi decidida a utilização do método *cognitive walkthrough*. Este método permite avaliar a qualidade nosso sistema para utilizadores sem treino próprio.

Assim, para construção da uma melhor interface final, tendo por base as *mockups* realizadas, foi aplicado este método a vários peritos externos à equipa de desenvolvimento.

Os formulários criados pretendem abranger os principais *use cases* da aplicação, sendo que para isso foram criados os 5 formulários apresentados nos anexos na subsecção 20.1.

17.1.1 Perito 1

Em seguida, apresentam-se as respostas realizadas por parte do primeiro perito ao aos formulários criados:

Registrar

Sistema:	GameHub		
Tarefa:	Registrar		
Utiliza.:	Henrique		
Passos	Questões	OK? (S/N)	Risco(1-3)
			Problema/Sugestão
	1	1 S	
Carregar		2 S	
criar		3 S	
conta		4 S	
	2	1 S	
Inserir		2 S	
username		3 S	
		4 S	
	3	1 S	
Inserir		2 S	
email		3 S	
		4 S	
	4	1 S	
Inserir		2 S	
password		3 S	
		4 S	
	5	1 S	
Aceitar		2 S	
termos		3 S	
		4 N	1 Deixar o botão greyed out enquanto não aceitar
	6	1 S	
Criar		2 S	
conta		3 S	
		4 S	

Figura 17.1: Formulário Perito 1 - Registrar

O perito 1, relativamente ao formulário de registo não teve grandes críticas a apontar, demonstrando apenas o seu desagrado no facto da *mockup* realizada não ter o botão de concluir registo a cinzento enquanto que todos os dados não tenham sido totalmente inseridos.

Criar Post de Grupo

Sistema:	GameHub		
Tarefa:	Criar Post Grupo		
Utiliza.:	Henrique		
Passos	Questões	OK? (S/N)	Risco(1-3)
			Problema/Sugestão
	1	1 N	1 Colocar o botão post no home
Carregar		2 N	1 Colocar o botão post no home
criar		3 S	
post		4 S	
	2	1 S	
Carregar		2 S	
criar		3 S	
equipa		4 N	1 Fica greyed out ao selecionar, devia ser o inverso
	3	1 S	
Escolher		2 S	
jogo		3 S	
plataforma		4 S	
	4	1 S	
Escolher		2 S	
Idioma		3 S	
		4 S	
	5	1 S	
Inserir		2 S	
nome		3 S	
equipa		4 S	
	6	1 S	
Inserir		2 S	
número		3 S	
participantes		4 S	
	7	1 S	
Escolher		2 S	
modo		3 S	
jogo		4 S	
	8	1 S	
Convidar		2 S	
amigos		3 S	
		4 N	1 Não diz que o amigo está convidado
	9	1 S	
Adicionar		2 S	
outros		3 S	
requisitos		4 S	
	10	1 S	
Escrever		2 S	
Descrição		3 S	
		4 S	
	11	1 S	
Criar		2 S	
post		3 S	
		4 S	

Figura 17.2: Formulário Perito 1 - Criar post de grupo

No que toca à funcionalidade de criar um *post* de grupo haviam algumas falhas a apontar. A primeira devido ao facto de que na *home page* não existir qualquer botão ou referência relativamente a esta ação. Depois, ao carregar em criar *post*, e pedir para escolher o tipo de *post* que queria realizar, o perito achou que as cores do botão não selecionado e do botão selecionado, deviam ser invertidas. Por fim também afirmou que após inserir um convite de amigo, não tinha como saber se esse amigo teria sido mesmo convidado ou não.

Criar Post Multimédia

Sistema:	GameHub		
Tarefa:	Criar Post Multimedia		
Utiliza.:	Henrique		
Passos	Questões	OK? (S/N)	Risco(1-3) Problema/Sugestão
	1	1 N 2 N 3 S 4 S	1 Colocar o botão post no home
Carregar		1 S	1 Colocar o botão post no home
criar		2 S	
<u>post</u>		3 S	
	multimedia	4 N	1 Fica greyed out ao selecionar, devia ser o inverso
	3	1 S 2 S 3 S 4 S	
Escolher		1 S	
jogo		2 S	
	4	1 S 2 S 3 S 4 S	
Escrever		1 S	
Descrição		2 S	
	5	1 S 2 S 3 S 4 S	
Adicionar		1 S	
ficheiros		2 S	
	multimedia	3 S	
	6	1 S 2 S 3 S 4 S	
Adicionar		1 S	
tags		2 S	
	7	1 S 2 S 3 S 4 S	
Criar		1 S	
<u>post</u>		2 S	
		3 S	
		4 S	

Figura 17.3: Formulário Perito 1 - Criar *post*multimédia

Na criação do *post* multimédia o perito queixou-se exatamente das mesmas falhas relativamente à localização do botão e da inversão de cores nos botões seguintes.

Seguir novo jogo

Sistema:	GameHub		
Tarefa:	Seguir novo jogo		
Utiliza.:	Henrique		
Passos	Questões	OK? (S/N)	Risco(1-3) Problema/Sugestão
	1	1 S 2 S 3 S 4 S	
Carregar		1 S	
Adicionar		2 S	
<u>jogo</u>		3 S	
	2	1 S 2 S 3 S 4 S	
Procurar		1 S	
jogo		2 S	
	3	1 S 2 S 3 S 4 S	
Selecionar		1 S	
jogo		2 S	
	4	1 S 2 S 3 S 4 S	
Concluir		1 S	
<u>adição</u>		2 S	
		3 S	
		4 S	

Figura 17.4: Formulário Perito 1 - Seguir novo jogo

Já a seguir novo jogo, o perito não teve nada a apontar, dizendo que o botão com o símbolo (+), na localização em que se encontrava, tornava-se auto explicativo relativamente à sua ação.

Juntar a Grupo

Sistema:	GameHub		
Tarefa:	Juntar a equipa		
Utiliza:	Henrique		
Passos	Questões OK? (S/N Risco(1-3) Problema/Sugestão		
1	1 N	1	Colocar o botão juntar a equipa no home
Carregar	2 N	1	Colocar o botão juntar a equipa no home
juntar	3 S		
equipa	4 S		
2	1 S		
Escolher	2 S		
jogo	3 S		
plataforma	4 S		
3	1 S		
Escolher	2 S		
modo	3 S		
jogo	4 S		
4	1 S		
Escolher	2 S		
linguagem	3 S		
	4 S		
5	1 S		
Inserir	2 S		
nome	3 S		
equipa	4 S		
6	1 S		
Selecionar	2 S		
equipa	3 S		
	4 S		
7	1 S		
juntar	2 S		
equipa	3 S		
	4 S		

Figura 17.5: Formulário Perito 1 - Juntar a Equipa

Para finalizar, a ação de juntar a grupo sofreu também com a escolha da *home page* selecionada. Esta sendo o perfil do próprio utilizador, não contém o botão de adicionar *post* nem o botão de juntar a equipa. Isto por sua vez causa uma certa confusão ao perito, uma vez que não lhe permite executar as tarefas principais logo a partir do primeiro ecrã.

17.1.2 Perito 2

Para o questionário aplicado ao segundo perito obteve-se as respostas apresentadas nas subsecções seguintes.

Registrar

Relativamente à realização do registo na aplicação, este perito não teve qualquer dificuldade/crítica relativamente à interface apresentada na *mockup*. Assim o seu formulário encontra-se nos anexos na figura 20.2.

Criar Post de Grupo

Sistema:	GameHub	
Tarefa:	Criar Post Grupo	
Utiliza.:	João	
Passos	Questões OK? (S/N) Risco(1-3) Problema/Sugestão	
1		
Carregar	1 N	2 Ter que ir para um feed para poder criar post
criar	2 N	2 Ter que ir para um feed para poder criar post
post	3 S	
	4 S	
2		
Carregar	1 S	
criar	2 S	
equipa	3 S	
	4 S	
3		
Escolher	1 S	
jogo	2 S	
plataforma	3 S	
	4 S	
4		
Escolher	1 S	
Idioma	2 S	
	3 S	
	4 S	
5		
Inserir	1 S	
nome	2 S	
equipa	3 S	
	4 S	
6		
Inserir	1 S	
número	2 S	
participantes	3 S	
	4 S	
7		
Escolher	1 S	
modo	2 S	
jogo	3 S	
	4 S	
8		
Convidar	1 S	
amigos	2 S	
	3 S	
	4 S	
9		
Adicionar	1 S	
outros	2 S	
requisitos	3 S	
	4 N	1 Criar uma zona em que o requisito mínimo for introduzido
10		
Escrever	1 S	
Descrição	2 S	
	3 S	
	4 S	
11		
Criar	1 S	
post	2 S	
	3 S	
	4 S	

Figura 17.6: Formulário Perito 2 - Criar *post* de grupo

O perito 2 na ação de criar um *post de grupo* teve uma certa dificuldade em descobrir como era possível criar um *post* a partir da página inicial, precisando de ajuda para o resolver. Disse que o problema era a necessidade de ir ao *feed* de um jogo para poder realizar esta tarefa. Mais no fim desta ação, queixou-se também que, ao adicionar requisitos adicionais, nada o informava que estes tinham sido adicionados, propondo então criar uma zona em que o registo seja adicionado.

Criar Post Multimédia

Sistema:	GameHub		
Tarefa:	Criar Post Multimedia		
Utiliza.:	João		
Passos	Questões OK? (S/N) Risco(1-3) Problema/Sugestão		
1			
Carregar	1	N	2 Ter que ir para um feed para poder criar post
criar	2	N	2 Ter que ir para um feed para poder criar post
post	3	S	
	4	S	
2			
Carregar	1	S	
post	2	S	
multimedia	3	S	
	4	S	
3			
Escolher	1	S	
jogo	2	S	
	3	S	
	4	S	
4			
Escrever	1	S	
Descrição	2	S	
	3	S	
	4	S	
5			
Adicionar	1	S	
ficheiros	2	S	
multimedia	3	S	
	4	S	
6			
Adicionar	1	S	
tags	2	S	
	3	S	
	4	S	
7			
Criar	1	S	
post	2	S	
	3	S	
	4	S	

Figura 17.7: Formulário Perito 2 - Criar post multimédia

Para a realização de um *post* multimédia, este perito apontou exatamente a mesma falha relativamente ao *post* de grupo, que era a necessidade de ir ao *feed* de um jogo para poder executar a tarefa.

Seguir novo jogo

Sistema:	GameHub		
Tarefa:	Seguir novo jogo		
Utiliza.:	João		
Passos	Questões OK? (S/N) Risco(1-3) Problema/Sugestão		
1			
Carregar	1	S	
Adicionar	2	S	
jogo	3	S	
	4	S	
2			
Procurar	1	S	
jogo	2	S	
	3	S	
	4	S	
3			
Selecionar	1	S	
jogo	2	S	
	3	S	
	4	S	
4			
Concluir	1	S	
adição	2	S	
	3	S	
	4	S	

Figura 17.8: Formulário Perito 2 - Seguir novo jogo

O perito 2 não teve qualquer dificuldade a realizar esta ação.

Juntar a Grupo

Sistema:	GameHub		
Tarefa:	Juntar a equipa		
Utiliza.:	João		
Passos	Questões OK? (S/N)	Risco (1-3)	Problema/Sugestão
1	1 S		
Carregar	2 S		
juntar	3 S		
equipa	4 S		
2	1 N	1	Devia estar o jogo selecionado consoante o feed escolhido
Escolher	2 S		
jogo	3 S		
plataforma	4 S		
3	1 S		
Escolher	2 S		
modo	3 S		
jogo	4 S		
4	1 S		
Escolher	2 S		
linguagem	3 S		
	4 S		
5	1 S		
Inserir	2 S		
nome	3 S		
equipa	4 S		
6	1 S		
Selecionar	2 S		
equipa	3 S		
	4 S		
7	1 S		
juntar	2 S		
equipa	3 S		
	4 S		

Figura 17.9: Formulário Perito 2 - Juntar a Equipa

Por fim no que toca a juntar a grupo, este perito já não apontou a necessidade de ir a um *feed* de um jogo para realizar a ação como sendo uma falha, mas como sendo algo lógico, uma vez que, segundo ele, faz sentido procurar o *feed* do jogo que quer jogar para depois se poder juntar a um grupo do mesmo. No entanto ao realizar o *post* já achou contra intuitivo ter que selecionar o jogo para o qual se quer juntar ao grupo, uma vez que essa estando no *feed* de um determinado jogo, seria esse o jogo para o qual quereria procurar um grupo.

17.1.3 Perito 3

Por fim, realizou-se os testes junto do perito 3. Assim, as respostas dadas por parte do terceiro perito aos formulários realizados, são apresentadas de seguida:

Registrar

Também o perito 3 não teve qualquer dificuldade nem levantou nenhum problema relativamente à realização do seu registo. Assim sendo o seu formulário encontra-se nos anexos, mais concretamente na figura 20.3.

Criar Post de Grupo

Sistema:	GameHub		
Tarefa:	Criar Post Grupo		
Utiliza.:	Eduarda		
Passos	Questões	OK? (S/N)	Risco(1-3) Problema/Sugestão
	1	1 N	2 Não existência do botão no home
Carregar		2 N	2 Não existência do botão no home
criar		3 S	
post		4 S	
	2	1 S	
Carregar		2 S	
criar		3 S	
equipa		4 S	
	3	1 S	
Escolher		2 S	
jogo		3 S	
plataforma		4 S	
	4	1 S	
Escolher		2 S	
Idioma		3 S	
		4 S	
	5	1 S	
Inserir		2 S	
nome		3 S	
equipa		4 S	
	6	1 S	
Inserir		2 S	
número		3 S	
participantes		4 S	
	7	1 S	
Escolher		2 S	
modo		3 S	
jogo		4 S	
	8	1 S	
Convidar		2 S	
amigos		3 S	
		4 N	1 Pôr notificação a dizer que convidaste tal pessoa
	9	1 S	
Adicionar		2 S	
outros		3 S	
requisitos		4 N	1 Criar campo com o requisito introduzido
	10	1 S	
Escrever		2 S	
Descrição		3 S	
		4 S	
	11	1 S	
Criar		2 S	
post		3 S	
		4 S	

Figura 17.10: Formulário Perito 3 - Criar *post* de grupo

O perito 3 não foi exceção, e também teve problemas para realizar um *post* de grupo, tendo sido necessário pedir ajuda para realizar esta ação. Para além disso criticou tanto o convite de amigos como a adição de requisitos extra, uma vez que tanto num como no outro, o sistema não apresentava informações relativamente à sua adição, deixando o utilizador sem certezas de que realizou a ação correta.

Criar Post Multimédia

Sistema:	GameHub		
Tarefa:	Criar Post Multimédia		
Utiliza.:	Eduarda		
Passos	Questões OK? (S/N) Risco(1-3) Problema/Sugestão		
1			
Carregar	1	N	2 Não existência do botão no home
	2	N	2 Não existência do botão no home
criar	3	S	
post	4	S	
2			
Carregar	1	S	
post	2	S	
multimedia	3	S	
	4	S	
3			
Escolher	1	S	
jogo	2	S	
	3	S	
	4	S	
4			
Escrever	1	S	
Descrição	2	S	
	3	S	
	4	S	
5			
Adicionar	1	S	
ficheiros	2	S	
multimedia	3	S	
	4	S	
6			
Adicionar	1	S	
tags	2	S	
	3	S	
	4	S	
7			
Criar	1	S	
post	2	S	
	3	S	
	4	S	

Figura 17.11: Formulário Perito 3 - Criar postmultimédia

O passo de encontrar o botão para criar *post* de multimédia, é totalmente análoga à tarefa anterior. De resto, o perito não obteve mais num problema ou dificuldade na execução desta tarefa.

Seguir novo jogo

Sistema:	GameHub		
Tarefa:	Seguir novo jogo		
Utiliza.:	Eduarda		
Passos	Questões OK? (S/N) Risco(1-3) Problema/Sugestão		
1			
Carregar	1	S	
Adicionar	2	S	
jogo	3	S	
	4	S	
2			
Procurar	1	S	
jogo	2	S	
	3	S	
	4	S	
3			
Selecionar	1	S	
jogo	2	S	
	3	S	
	4	S	
4			
Concluir	1	S	
adição	2	S	
	3	S	
	4	S	

Figura 17.12: Formulário Perito 3 - Seguir novo jogo

Para seguir um novo jogo, este perito, tal como todos os anteriores, realizou a tarefa sem qualquer dificuldade.

Juntar a Equipa

Sistema:	GameHub		
Tarefa:	Juntar a equipa		
Utiliza.:	Eduarda		
Passos	Questões	OK? (S/N)	Risco(1-3) Problema/Sugestão
	1	1 N	2 Adicionar o botão de adicionar equipa ao home
Carregar		2 N	2 Adicionar o botão de adicionar equipa ao home
juntar		3 S	
equipa		4 S	
	2	1 S	
Escolher		2 S	
jogo		3 S	
plataforma		4 S	
	3	1 S	
Escolher		2 S	
modo		3 S	
jogo		4 S	
	4	1 S	
Escolher		2 S	
linguagem		3 S	
		4 S	
	5	1 S	
Inserir		2 S	
nome		3 S	
equipa		4 S	
	6	1 S	
Selecionar		2 S	
equipa		3 S	
		4 S	
	7	1 S	
juntar		2 S	
equipa		3 S	
		4 S	

Figura 17.13: Formulário Perito 3 - Juntar a Equipa

Para finalizar, o perito 3 obteve exatamente o mesmo problema para executar o passo 1 desta tarefa, como obteve para executar as duas tarefas de criação de *posts*.

17.1.4 Conclusões do *Cognitive Walkthrough*

Tendo então obtido os resultados, segue-se uma pequena análise dos aspetos mencionados pelos vários peritos. Estes aspetos foram tidos em conta para a realização da interface real do projeto, de modo a permitir a todos os utilizadores obterem uma melhor experiência de usabilidade da nossa aplicação.

Registrar

Em relação à tarefa de registrar, verificou-se que nenhum dos peritos teve qualquer dificuldade em realizar esta tarefa, tendo apenas sido mencionado um pormenor por parte do perito 1, dizendo que o botão de criar conta deveria estar cinzento até todos os campos estarem preenchidos, incluindo a seleção da aceitação dos termos.

Criar Post de Grupo

Na tarefa de criação de um *post* de grupo, todos os peritos tiveram dificuldade em iniciar esta ação, uma vez que a página inicial (*home*) da aplicação na *mockup* apresenta-se como sendo a página de perfil. Assim não existia nenhum botão nessa zona para criar *posts* sendo necessário ir a algum *feed* para criação de um *post*. Considerando este problema mais ou menos grave, todos revelaram a necessidade deste botão se encontrar na *home page*. Nesta tarefa ainda houve mais dois passos onde existiram alguns problemas devido ao sistema não apresentar nenhuma informação relativa às ações realizadas (convidar amigos e adicionar outros requisitos). Relativo ao convidar amigos, foi sugerida uma notificação a informar do sucedido e para o caso da adição de outros requisitos, foi mencionado a adição do campo no menu da criação do *post*, com a informação adicionada.

Criar Post Multimédia

Relativamente à tarefa de criar *post* multimédia, sendo necessário carregar no botão adicionar *post* como no caso anterior, sofre do mesmo problema inicial relativamente à localização do botão. No entanto foi aqui também apontado, por parte de um perito que após escolher adicionar *post*, ao escolher o tipo de *post* necessário, as cores do que está selecionado e do que não está não estão de acordo com o resto da interface, devendo ser alteradas para o inverso.

Seguir novo jogo

No que toca à tarefa de seguir novo jogo, nenhum perito teve qualquer dificuldade na sua realização, afirmando que o botão de adição simbolizado por +, na aba dos jogos que o utilizador segue no momento, é de facto auto-explicativo.

Juntar a Equipa

Para finalizar relativamente à tarefa de juntar a uma equipa, houve dois pontos de vista diferentes. O primeiro passa pela necessidade da adição do botão ao *home* tal como o botão de *post*. Um segundo ponto de vista refere que acha totalmente necessário ter a necessidade de ir a um determinado *feed*, para se juntar a uma equipa desse jogo, no entanto acha que por ser necessária esta ação, a seleção do jogo deveria estar já feita, com o jogo do *feed* em que carregou para juntar a uma equipa. Para além disso houve um perito que se queixou da ordem dos filtros de procura da equipa, recomendando a seguinte ordem: plataforma, jogo, modo de jogo, idioma.

17.2 SUS - System Usability Scale

Após o desenvolvimento duma primeira versão do sistema, procedeu-se à sua análise de usabilidade. Para realizar essa análise, utilizaram-se questionários de System Usability Scale (SUS), com os utilizadores que experimentaram a primeira versão desenvolvida.

17.2.1 Participantes:

Assim, o questionário foi realizado com 31 participantes distintos. Nestes participantes incorporou-se um conjunto diversificado de pessoas, que permitiram medir a usabilidade da interface da forma mais realista possível.

O questionário disponível em <https://forms.gle/N77PJLw2tEpHaDu37>, foi preenchido por pessoas do sexo feminino e masculino, com idades分散 pelas entre os 19 e 30 anos. Neste conjunto de participantes, identificaram-se parte deles como *gamers* (jogam frequentemente videojogos) e outros que simplesmente não manifestam interesse nessa atividade.

17.2.2 Resultados

Desta forma, aplicou-se o questionário aos vários utilizadores do sistema. A aplicação do questionário permitiu compilar um conjunto de valores para cada questão. Assim, realizou-se a média de todos os valores obtidos por questão de modo a poder se efetuar o cálculo do *score SUS*. Os valores médios obtidos podem ser analisados na tabela 17.1.

Pergunta	Valor médio obtido
1) Eu acho que gostaria de usar este sistema com frequência.	4.097
2) Eu acho o sistema desnecessariamente complexo.	1.516
3) Eu achei o sistema fácil de usar.	4.71
4) Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.	1.323
5) Eu acho que as várias funcionalidades do sistema estão muito bem integradas.	4.548
6) Eu acho que o sistema apresenta muita inconsistência.	1.323
7) Eu imagino que as pessoas aprenderão como usar este sistema rapidamente.	4.774
8) Eu achei o sistema muito complicado de usar.	1.258
9) Eu senti-me confiante ao usar o sistema.	4.677
10) Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.	1.161

Tabela 17.1: Valores médios obtidos, por questão, no questionário SUS

Após a análise dos valores obtidos foi necessário proceder ao cálculo da usabilidade obtida. Este cálculo seguiu o método desenvolvido pelo John Brooke em 1996. Assim, realizou-se:

- 1. Para perguntas pares:** Subtrai-se ao valor cinco o valor médio obtido. Por exemplo, para a pergunta 2 o valor é calculado através da fórmula $(5 - 1.516)$;
- 2. Para perguntas ímpares:** Subtrai-se um ao valor médio obtido. Por exemplo, para a pergunta 1 o valor é calculado da seguinte forma: $(4.097 - 1)$;
- 3. Após realizar os cálculos referidos para todas as perguntas,** soma-se os valores obtidos e multiplica-se o valor agregado por 2.5;

Ao realizar este processo obteve-se como valor de usabilidade, aproximadamente 90.6. Com a obtenção deste valor pode-se considerar que a interface permite aos utilizadores efetuarem as funcionalidades de forma simples e satisfatória, apresentando um grau de usabilidade elevada. Para analisar este valor assume-se a convenção, de que o normal para estes testes é obter um valor de 68 e que obter um valor acima de 80 determina que a interface apresenta um bom grau de usabilidade.

Perguntas ímpares	17,80645161
Perguntas pares	18,41935484
Soma	36,22580645
Soma multiplicada por 2.5	90,56451613

Figura 17.14: Valores obtidos nos passos intermédios

18 Think aloud

O protocolo de *Think Aloud* consiste em ter os participantes (utilizadores do sistema) a pensar em voz alta enquanto realizam um conjunto de tarefas. Para cada um dos utilizadores, pedíamos que realizasse uma tarefa que corresponde a uma funcionalidade do sistema e pedíamos que dissessem, em voz alta, todo o pensamento durante o processo de realizar essa mesma ação. Ao observar os participantes a realizar este processo, fomos capazes de tomar nota de locais da interface onde os utilizadores demonstravam mais frustração e dificuldade.

A realização destes testes permitiu identificar vários problemas cujas retificações, certamente, contribuíram para o ótimo resultado obtido no questionário *SUS*, tais como:

- Tornar as mensagens de erros durante o registo de uma conta mais visíveis
- Permitir aceder ao perfil de um amigo carregando com o lado direito em cima do seu nome e selecionando "ver perfil".
- Corrigir uma notificação de erro que aparecia mesmo quando o avatar do perfil era atualizado com sucesso.
- Não deixar seguir um jogo nos jogos mais populares (página inicial) quando já se segue esse jogo.
- Adicionar um botão para deixar de seguir um jogo no *feed* de cada jogo.
- Adicionar um som de notificação para tornar a receção de mensagens mais perceptível.
- Não retirar o cursor do local de escrita do chat ao submeter uma nova mensagem.
- Corrigir alinhamento de certos elementos visuais em ecrãs com diferentes resoluções.

19 Conclusão

Após o desenvolvimento da motivação e objetivos deste projeto, o primeiro desafio foi sair da abstração da ideia e concretizar as funcionalidades que os utilizadores gostariam de ter no sistema, e como estas seriam implementadas. Assim, procedeu-se à realização de um diagrama de use cases. De seguida, com a criação de *mockups* e diagramas de tarefas das várias ações, fomos capazes de ter uma base de referência para o desenvolvimento do sistema.

Com estes *mockups*, mesmo antes de implementar o sistema, fomos capazes de realizar uma análise de usabilidade com o teste de *Cognitive Walkthrough*, que nos permitiu levantar vários possíveis problemas a ter em conta durante o desenvolvimento da interface da aplicação.

Numa segunda fase, foi necessário levantar os requisitos funcionais e não funcionais do sistema, tendo em conta todas as funcionalidades abordadas anteriormente, de modo a especificar os objetivos e restrições da implementação do sistema.

Na fase de modelação, procedemos à especificação do sistema, com recurso a múltiplos diagramas, nomeadamente: diagrama de contexto de negócio, diagrama de contexto técnico, diagrama de componentes, diagrama de classes (PIM e PSM) e diagramas de sequência. Com estes diagramas definidos, a fase de implementação tornou-se muito mais simples.

Como referido anteriormente, a fase de implementação foi relativamente simples, uma vez que todo o sistema estava especificado detalhadamente na fase de modelação. Deste modo, com recurso a tecnologias como Vue.js, Spring e Hibernate, fomos capazes de implementar o sistema com grande fidelidade a toda a documentação realizada.

Com o sistema implementado, foi necessário criar uma estratégia de *deployment* da aplicação, para a podermos testar. Assim, após a criação de um diagrama de *deployment*, e com o auxílio de ferramentas como o *Ansible* e *Docker Containers*, alcançámos a automatização da operação de *deployment* da aplicação. Todo este esforço rapidamente se tornou vantajoso, uma vez que minimizou o esforço de fazer *deployment* sempre que tínhamos uma nova versão do sistema pronta a testar.

Com a aplicação funcional e instalada em servidores da *Google Cloud Platform*, foi extremamente importante testar a aplicação. Começando pela análise da escalabilidade, utilizando a ferramenta *Apache JMeter*, fomos capazes de verificar o impacto causado pelo aumento de carga no sistema. Verificamos que a nossa aplicação conseguiu escalar até cerca de 1000 utilizadores com uma percentagem de erro praticamente nula, mas com um tempo de resposta relativamente elevado.

Para além de testar a escalabilidade do sistema, foi muito importante verificar se fomos capazes de desenvolver uma boa interface para a aplicação. Deste modo, utilizando o teste de *Think Aloud* e, posteriormente, o questionário *SUS* para avaliar a usabilidade do sistema, fomos capazes de corrigir todos os possíveis problemas e obtemos um resultado de 90.56, o que demonstrou o sucesso da nossa implementação.

Como trabalho futuro, gostaríamos de adicionar mais funcionalidades ao nosso sistema e acabar de implementar alguns dos requisitos definidos durante o levantamento dos mesmos, tais como: permitir convidar pessoas quando se cria um grupo; permitir ao utilizador estar em vários grupos; acrescentar restrições aos grupos para limitar os utilizadores que se podem juntar; refinar o sistema de reputação para refletir outros atributos do jogador tais como a sua habilidade; adicionar um chat de voz ao sistema; adicionar uma funcionalidade de realizar *livestreams*; verificar a idade dos utilizadores quando seguem jogos, de modo a respeitar o valor de ESRB (Entertainment Software Rating Board); poder recusar pedidos de amizade; aumentar as opções de personalização da interface.

Para além disto, também gostaríamos de lidar com os problemas de escalabilidade encontrados. Para isso, poderíamos começar pela otimização da base de dados com recurso a índices, otimização de *queries*, replicação e diminuir o número de pedidos que o cliente faz ao servidor aplicacional, de modo a diminuir o *overhead* introduzido pelo *delay* da rede.

Referências

- [1] H. Garcia-Molina, J. Ullman and J. Widom. Database Systems: The Complete Book. Prentice-Hall, 2006 (2 nd Edition).
- [2] S. Camille Peres, Tri Pham, Ronald Phillips. VALIDATION OF THE SYSTEM USABILITY SCALE (SUS): SUS IN THE WILD. Prentice-Hall, 2013.

20 Anexos

20.1 Cognitve Walkthroughs - Formulários

20.1.1 Registar

Sistema:	GameHub			
Tarefa:	Registrar			
Utiliza.:				
Passos	Questões	OK? (S/N)	Risco(1-3)	Problema/Sugestão
	1	1		
Criar conta	2			
	3			
	4			
	2	1		
Inserir username	2			
	3			
	4			
	3	1		
Inserir email	2			
	3			
	4			
	4	1		
Inserir password	2			
	3			
	4			
	5	1		
Aceitar termos	2			
	3			
	4			
	6	1		
Criar conta	2			
	3			
	4			

Figura 20.1: Formulário - Registrar

Sistema:	GameHub			
Tarefa:	Registrar			
Utiliza.:	João			
Passos	Questões	OK? (S/N)	Risco(1-3)	Problema/Sugestão
	1	1 S		
Criar conta	2	S		
	3	S		
	4	S		
	2	1 S		
Inserir username	2	S		
	3	S		
	4	S		
	3	1 S		
Inserir email	2	S		
	3	S		
	4	S		
	4	1 S		
Inserir password	2	S		
	3	S		
	4	S		
	5	1 S		
Aceitar termos	2	S		
	3	S		
	4	S		
	6	1 S		
Criar conta	2	S		
	3	S		
	4	S		

Figura 20.2: Formulário Utilizador 2 - Registrar

Sistema:	GameHub		
Tarefa:	Registrar		
Utiliza.:	Eduarda		
Passos	Questões	OK? (S/N)	Risco(1-3)
			Problema/Sugestão
	1	1 S	
Criar		2 S	
conta		3 S	
		4 S	
	2	1 S	
Inserir		2 S	
username		3 S	
		4 S	
	3	1 S	
Inserir		2 S	
email		3 S	
		4 S	
	4	1 S	
Inserir		2 S	
password		3 S	
		4 S	
	5	1 S	
Aceitar		2 S	
termos		3 S	
		4 S	
	6	1 S	
Criar		2 S	
conta		3 S	
		4 S	

Figura 20.3: Formulário Utilizador 3 - Registrar

20.1.2 Criar *post* de grupo

Sistema:	GameHub		
Tarefa:	Criar Post Grupo		
Utiliza.:			
Passos	Questões	OK? (S/N)	Risco(1-3) Problema/Sugestão
	1	1	
Carregar	2		
criar	3		
post	4		
	2	1	
Carregar	2		
criar	3		
equipa	4		
	3	1	
Escolher	2		
jogo	3		
plataforma	4		
	4	1	
Escolher	2		
Idioma	3		
	4		
	5	1	
Inserir	2		
nome	3		
equipa	4		
	6	1	
Inserir	2		
número	3		
participantes	4		
	7	1	
Escolher	2		
modo	3		
jogo	4		
	8	1	
Convidar	2		
amigos	3		
	4		
	9	1	
Adicionar	2		
outros	3		
requisitos	4		
	10	1	
Escrever	2		
Descrição	3		
	4		
	11	1	
Criar	2		
post	3		
	4		

Figura 20.4: Formulário - Criar *post* de grupo

20.1.3 Criar *post* multimédia

Sistema:	GameHub		
Tarefa:	Criar Post Multimedia		
Utiliza.:			
Passos	QuestõesOK? (S/N)	Risco(1-3)	Problema/Sugestão
Carregar	1	2	
criar		3	
post		4	
multimedia	2	1	
Escolher		2	
jogo		3	
		4	
Escrever	3	1	
Descrição		2	
		3	
		4	
Adicionar	4	1	
ficheiros		2	
multimedia		3	
		4	
Adicionar	5	1	
tags		2	
		3	
		4	
Criar	6	1	
post		2	
		3	
		4	

Figura 20.5: Formulário - Criar *post* multimédia

20.1.4 Seguir novo jogo

Sistema:	GameHub		
Tarefa:	Seguir novo jogo		
Utiliza.:			
Passos	QuestõesOK? (S/N)	Risco(1-3)	Problema/Sugestão
Carregar	1	2	
Adicionar		3	
jogo		4	
Procurar	2	1	
jogo		2	
		3	
		4	
Selecionar	3	1	
jogo		2	
		3	
		4	
Concluir	4	1	
adição		2	
		3	
		4	

Figura 20.6: Formulário - Seguir novo jogo

20.1.5 Juntar a equipa

Sistema:	GameHub			
Tarefa:	Juntar a equipa			
Utiliza.:				
Passos	Questões	OK? (S/N)	Risco(1-3)	Problema/Sugestão
	1	1		
Carregar		2		
juntar		3		
equipa		4		
	2	1		
Escolher		2		
jogo		3		
plataforma		4		
	3	1		
Escolher		2		
modo		3		
jogo		4		
	4	1		
Escolher		2		
linguagem		3		
		4		
	5	1		
Inserir		2		
nome		3		
equipa		4		
	6	1		
Selecionar		2		
equipa		3		
		4		
	7	1		
juntar		2		
equipa		3		
		4		

Figura 20.7: Formulário - Juntar a equipa