



**Centro Universitário UniRuy Wyden**  
**Bacharelado em Ciência da Computação**

**Antônio Carlos Sena da Conceição Junior - 202102120748**

**Luan Masao Hirahata - 202308337696**

**Bruno Gabriel Figueiredo da Paz Assis - 202308426036**

**ShotStats:**

Software de gerenciamento e ranqueamento de jogos

**SALVADOR - BA**

**2024**

**Antônio Carlos Sena da Conceição Junior - 202102120748**

**Luan Masao Hirahata - 202308337696**

**Bruno Gabriel Figueiredo da Paz Assis - 202308426036**

**ShotStats:**

Software de gerenciamento e ranqueamento de jogos

Trabalho apresentado à disciplina de  
Engenharia de Software do Curso de  
Sistemas de Informação da UNIRUY.

Orientador(a): Heleno Cardoso

**SALVADOR - BA**

**2024**

## Sumário

<b>1 INTRODUÇÃO</b>	4
<b>2 OBJETIVO DA APLICAÇÃO</b>	5
<b>3 FUNCIONALIDADE</b>	6
<b>3.1 Análise de requisitos</b>	6
3.1.1 Requisitos Funcionais	6
3.1.2 Requisitos Não Funcionais	6
<b>3.2 Levantamento de requisitos</b>	7
3.2.1 Entrevista/Questionário/Layout da tela	7
3.2.2 Diagrama de Casos de Uso - UML	9
<b>3.3 Especificação de programas</b>	10
3.3.1 Layout da tela	10
3.3.2 Regras de negócios	10
3.3.3 Entidades envolvidas	11
<b>4 DIAGRAMA DE CONTEXTO</b>	14
<b>5 DFD Nível zero</b>	14
<b>6 DFD POR EVENTO</b>	15
<b>7 DER/ DIAGRAMA DE CLASSE</b>	17
7.1 Artefato gráfico	16
7.2 Dicionário de dados (DDL)	16
7.3 Modelo comportamental	19
<b>8 POLÍTICAS DE TESTE</b>	20
<b>9 HOMOLOGAÇÃO</b>	20
<b>10 APLICAÇÃO PROTÓTIPO</b>	21
10.1 telas funcionais	20
<b>11 CONCLUSÃO</b>	25

## 1 INTRODUÇÃO

O seguinte projeto discorre sobre os conteúdos apresentados durante a disciplina de Engenharia de Software e baseia-se em apresentar uma aplicação protótipo e na documentação técnica de sistemas. Primeiramente, a implementação da aplicação protótipo foi feita somente de forma visual, sem a necessidade de utilizar uma linguagem de programação. A documentação consiste em elementos escritos e visuais, diagramas e tabelas que se tornaram boas práticas na indústria. Tais práticas não somente auxiliam no entendimento da aplicação pelo cliente, como também ajudam os responsáveis pela manutenção do código e base de dados da aplicação.

O domínio escolhido para a aplicação foi o de vídeo games, especificamente envolvendo jogadores que administram comunidades e seus servidores. Apesar de ser uma prática nos primórdios dos jogos online, em que era comum a presença de servidores da comunidade, ainda existem games com foco neste tipo de hosting e que possuem uma alta popularidade. Jogos como Counter Strike, Minecraft, Garry's Mod e DayZ, são alguns exemplos. Exemplificando melhor esta demanda, DayZ se tornou um jogo independente e com financiamento próprio após obter sucesso como uma modificação do jogo ArmA 2, em que foi necessário que as comunidades praticassem o host de seus servidores. Não somente ele, como PUBG também utilizou a série ArmA como um protótipo, lançando a tendência dos jogos Battle Royale.

## **2 OBJETIVO DA APLICAÇÃO**

O cliente requisitou a criação de um banco de dados com informações dos jogadores, a fim de integrá-las em uma aplicação que irá exibir estatísticas e progressão do jogador. Sendo assim, oferecendo um serviço exclusivo para aquele jogo e possibilitando a expansão de funcionalidades da aplicação com os dados coletados. Como o cliente é o responsável por hostear servidores customizados e independentes dos desenvolvedores, os dados serão coletados em seu próprio servidor. Para isso, será necessário utilizar uma API open source para integrar o game com o banco de dados, assim como a API Steam para gerenciar o login dos jogadores na aplicação que será criada. Com os dados coletados, nosso cliente procura facilitar os ajustes que afetam diretamente a jogabilidade do jogo, integrar melhor a comunidade em seu servidor e facilitar a identificação de hackers.

## 3 FUNCIONALIDADE

### 3.1 ANÁLISE DE REQUISITOS

#### Canvas de Mapeamento de Processos

##### Entrevista



1. Quais são os principais objetivos que você deseja alcançar com esta aplicação?
2. Como você imagina a interface do usuário? Você pode descrever ou tem algum exemplo em mente?
3. Quais tipos de informações do jogo você gostaria que fossem coletadas e exibidas?
4. Você precisa de funcionalidades de interação com o jogo, como atualizações em tempo real ou notificações?
5. Quais são os usuários finais do aplicativo e quais são suas principais características?
6. Quais são suas expectativas em relação à segurança dos dados coletados?
7. Como você visualiza o suporte e a manutenção do aplicativo após o lançamento?

##### Funções



- Login de jogador
- Login de administrador
- Exibir informações e estatísticas coletadas do jogo
- Conectar o jogo com o banco de dados
- Conectar a aplicação com autenticação Steam

##### Requisitos Funcionais



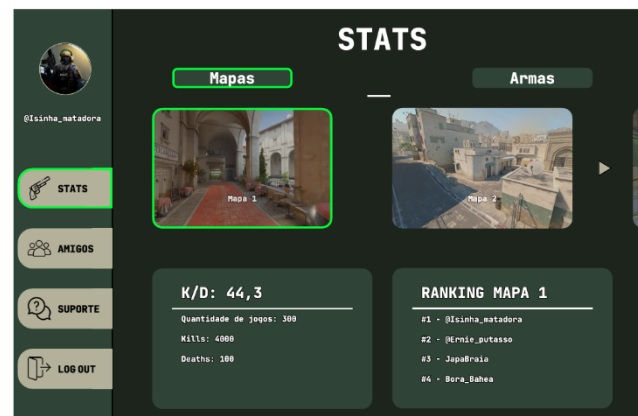
1. Exibir informações do usuário
2. Cadastro automático do jogador ao entrar no banco de dados
3. Integração da aplicação com o jogo

##### Requisitos Não Funcionais



1. Manutenibilidade do site
2. Usabilidade da plataforma
3. Confiabilidade

##### Layout Projeto



## 3.1.1 Requisitos Funcionais

<b>Código</b>	<b>Nome</b>	<b>Detalhe</b>
<b>RF01</b>	Exibir informações do usuário	É necessário exibir as informações enviadas ao banco de dados na aplicação.
<b>RF02</b>	Cadastro automático do jogador no banco de dados ao entrar no servidor	A steamid do jogador será utilizada para cadastrá-lo no banco de dados, permitindo que ele utilize sua conta para login e o progresso salvo pelo jogo seja enviado ao banco de dados. Como é necessário o login na Steam para abrir o jogo, o processo de autenticação é automático.
<b>RF03</b>	Integração da aplicação com o jogo	Enviar as informações coletadas pelo servidor do jogo para o banco de dados.

## 3.1.2 Requisitos Não Funcionais

<b>Código</b>	<b>Nome</b>	<b>Detalhe</b>
RNF01	Rodar a aplicação no navegador	O sistema deve ser acessível por meio de um navegador web, proporcionando uma interface amigável e responsiva aos usuários.
RNF02	Rodar banco de dados no mysql	O sistema deve utilizar um banco de dados MySQL para armazenar todas as informações persistentes, garantindo integridade, segurança e desempenho.

RNF03	Implementar o banco de dados em um servidor separado do jogo	O banco de dados deve ser hospedado em um servidor dedicado, separado do servidor da aplicação principal (jogo), para garantir melhor desempenho, segurança e escalabilidade.
-------	--------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## 3.2 LEVANTAMENTO DE REQUISITOS

Feito através de entrevistas, questionários e prototipação, visando entender as necessidades dos jogadores e administradores para melhorar a experiência de uso.

### 3.2.1 ENTREVISTA/QUESTIONÁRIO/ LAYOUT DA TELA

**Cliente:** Olá, obrigado por concordar em participar desta entrevista. Como representante dos jogadores e administradores de servidores, sua opinião é fundamental para o desenvolvimento do nosso projeto, o ShotStats. Vamos começar?

**Dev:** Claro, estou à disposição para ajudar.

**Cliente:** Ótimo. Primeiramente, poderia me contar um pouco sobre suas experiências com jogos online e servidores de comunidade?

**Dev:** Claro. Sou um entusiasta de jogos online há muitos anos e também administro um servidor de comunidade para um jogo bastante popular. Tenho experiência tanto como jogador quanto como administrador.

**Cliente:** Excelente. Como jogador e administrador, quais são as principais necessidades ou desafios que você enfrenta em relação ao gerenciamento de jogos e servidores?

**Dev:** Uma das principais necessidades é ter acesso fácil a estatísticas e informações dos jogadores, como tempo de jogo, estatísticas de desempenho e histórico de partidas. Além disso, como administrador, é crucial ter ferramentas eficazes para gerenciar e moderar o servidor, como banimento de jogadores mal-intencionados e controle de acesso.

**Cliente:** Entendi. Essas informações são muito úteis. Em relação à nossa aplicação, o ShotStats, como você acha que ela poderia ajudar a atender a essas necessidades?

**Dev:** Acredito que uma aplicação como o ShotStats seria extremamente útil. Se ela pudesse fornecer estatísticas detalhadas dos jogadores, facilitando o acompanhamento do desempenho e a identificação de possíveis problemas de



comportamento. Além disso, se a aplicação oferecesse ferramentas de moderação eficientes, como relatórios de incidentes e histórico de ações administrativas, seria um grande diferencial.

**Cliente:** Muito interessante. E quanto à integração com o jogo e o banco de dados, você acha que isso seria importante para melhorar a experiência dos jogadores e administradores?

**Dev:** Com certeza. Uma integração sólida com o jogo e o banco de dados permitiria uma troca de informações mais fluida e em tempo real. Isso facilitaria o acompanhamento do progresso dos jogadores, a identificação de tendências e até mesmo o desenvolvimento de novas funcionalidades com base nos dados coletados.

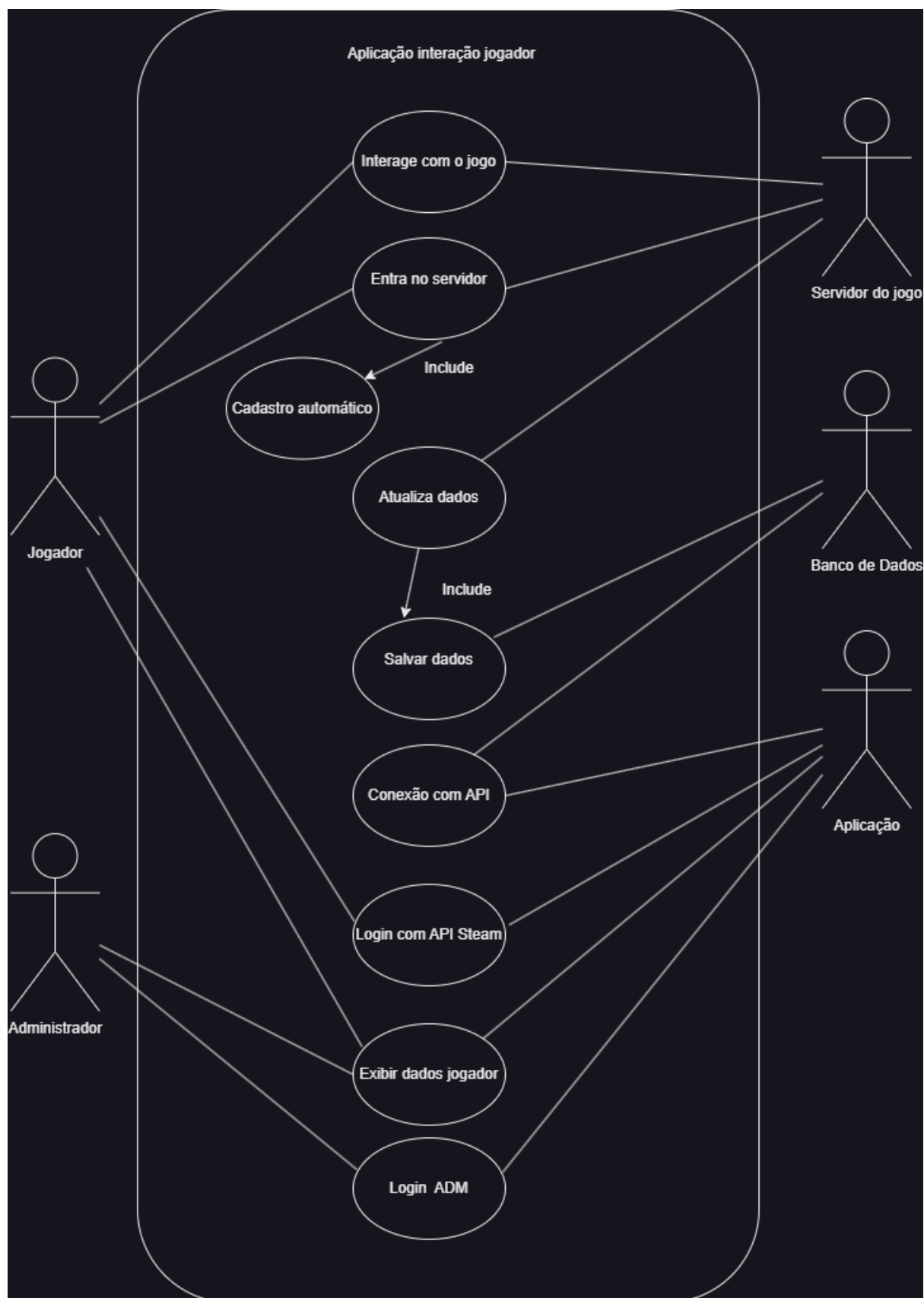
**Cliente:** Compreendo. Suas sugestões são muito valiosas. Mais alguma coisa que você gostaria de acrescentar ou sugerir para o desenvolvimento do ShotStats?

**Dev:** Acho que abordamos os pontos principais. Apenas gostaria de ressaltar a importância de uma interface intuitiva e fácil de usar, tanto para os jogadores quanto para os administradores. Quanto mais acessível e amigável for a aplicação, melhor será a sua aceitação pela comunidade.

**Cliente:** Entendido. Muito obrigado por compartilhar suas ideias conosco. Suas contribuições serão levadas em consideração no desenvolvimento do ShotStats.

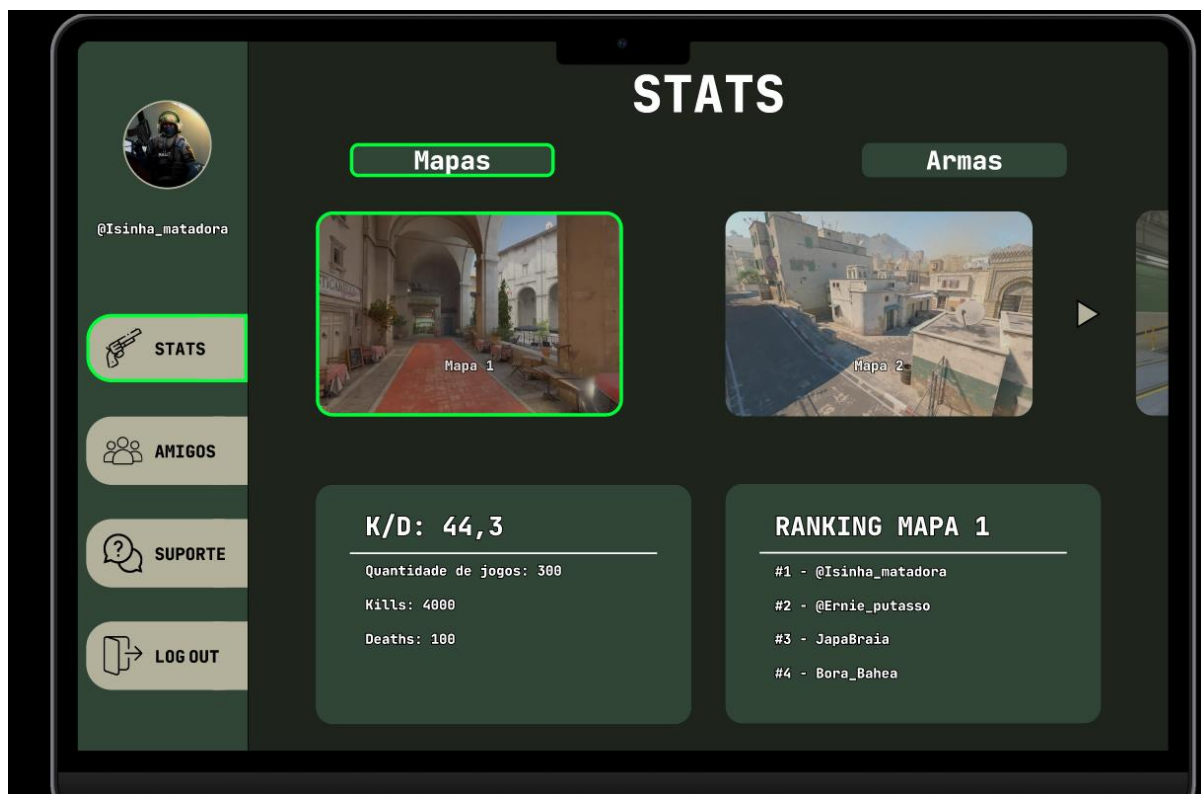
**Dev:** Eu agradeço pela oportunidade de participar. Estou ansioso para ver o resultado final do projeto.

## 3.2.2 Diagrama de Casos de Uso - UML



### 3.3 ESPECIFICAÇÃO DE PROGRAMAS

#### 3.3.1 Layout da tela



#### 3.3.2 Regras de negócio

##### **Regra de Negócio 1:** Bloqueio de Acesso por Denúncias de Hacking

**Regra:** Se um usuário receber denúncias comprovadas por hacking, seu acesso à plataforma será automaticamente bloqueado.

**Critério de Avaliação:** As denúncias precisam ser comprovadas após análise detalhada.

**Ação:** Bloqueio imediato do acesso à conta do usuário.

**Notificação:** O usuário será notificado por e-mail sobre o bloqueio, com detalhes sobre as denúncias e instruções para apelar da decisão, se aplicável.

##### **Regra de Negócio 2:** Utilização de Dados Originais para Cadastro

**Regra:** Durante o processo de cadastramento, os usuários devem utilizar os dados originais da plataforma onde estão registrados.

Dados Necessários: Nome completo, e-mail, data de nascimento, e qualquer outro dado que a plataforma original tenha solicitado.

Verificação: O sistema verificará automaticamente a autenticidade dos dados com a base de dados da plataforma original.

Rejeição de Cadastro: Caso os dados não correspondam, o cadastro será rejeitado até que o usuário forneça informações corretas.

### **Regra de Negócio 3: Canal de Suporte ao Usuário**

Regra: Um canal de suporte dedicado deve estar disponível para auxiliar os usuários em todas as etapas de interação com a plataforma.

Disponibilidade: O suporte deve estar disponível 24/7 através de chat ao vivo, e-mail e telefone.

Escalonamento de Problemas: Problemas que não puderem ser resolvidos imediatamente pelo suporte de nível 1 serão escalados para níveis superiores.

Feedback: Os usuários poderão fornecer feedback sobre a qualidade do suporte recebido, que será utilizado para melhorias contínuas.

### **3.3.3 Entidades envolvidas**

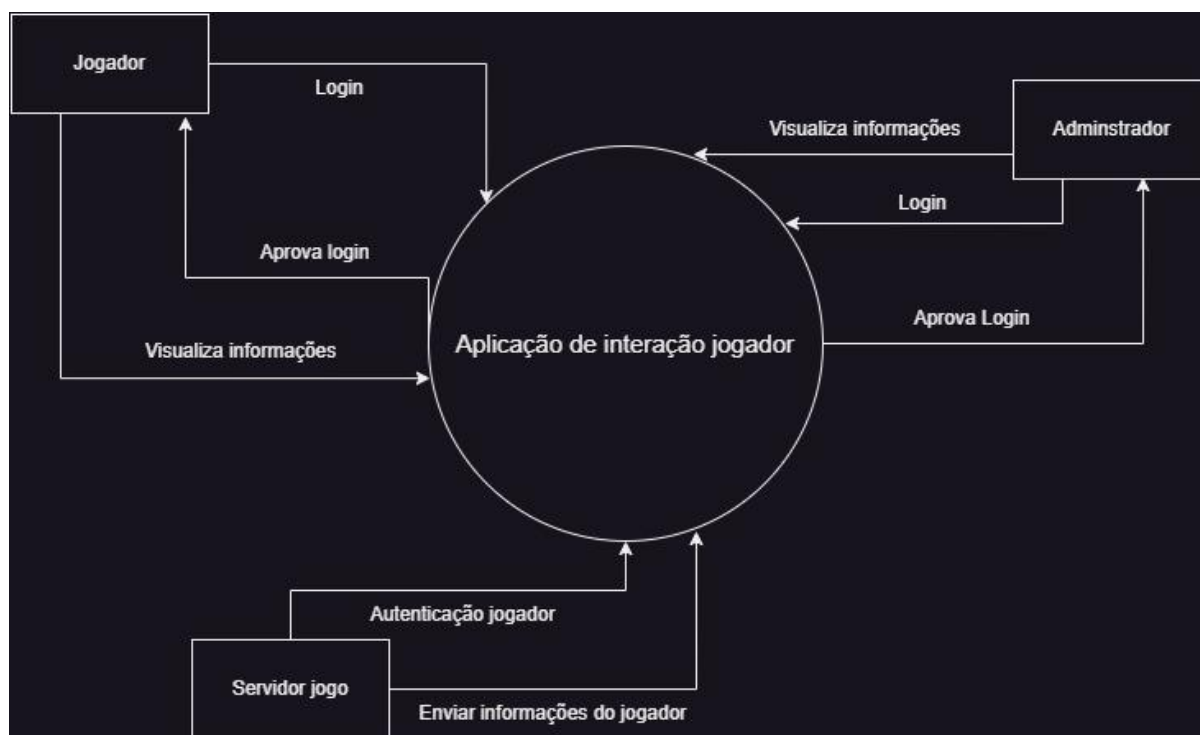
As seguintes entidades estão envolvidas no projeto: jogador, moderador, grupo, denuncia, eliminacao, arma, morte e partida.

#### **Entidades:**

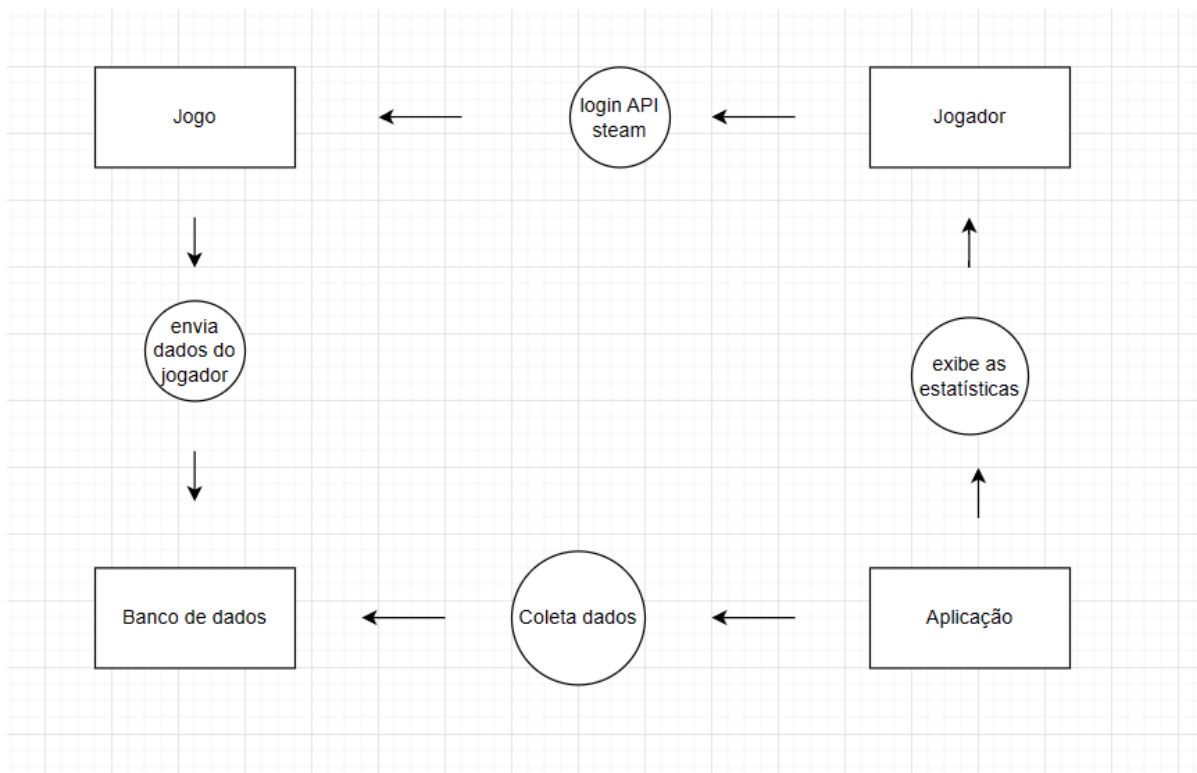
- Grupo: Representa um grupo(clã), com atributos como idgrupo, nomegrupo e datacriacao.
- Moderador: Esta é uma entidade que representa um moderador, com atributos como idmoderador, nomemoderador, senha e emailmoderador.

- Denúncia: Esta é uma entidade que representa uma denúncia, com atributos como iddenuncia, moderador\_idmoderador, textodenuncia, jogador\_steamid e partida\_idpartida.
- Jogador: Esta é uma entidade que representa um jogador, com atributos como steamid, grupo\_idgrupo, email, nickname e nivel.
- Arma: Esta é uma entidade que representa uma arma, com atributos como iditens, nomearma e danocausado.
- Morte: Esta é uma entidade que representa uma morte em um jogo, com atributos como idmorte, eliminacao\_ideliminacao, grupo\_idgrupo e jogador\_steamid.
- Partida: Esta é uma entidade que representa uma partida de jogo, com atributos como idpartida, mapa\_idmapa, data e horacomeco.
- Eliminação: Esta é uma entidade que representa uma eliminação em um jogo, com atributos como ideliminacao, hs, danototal, jogador\_steamid e grupo\_idgrupo.

## 4 DIAGRAMA DE CONTEXTO

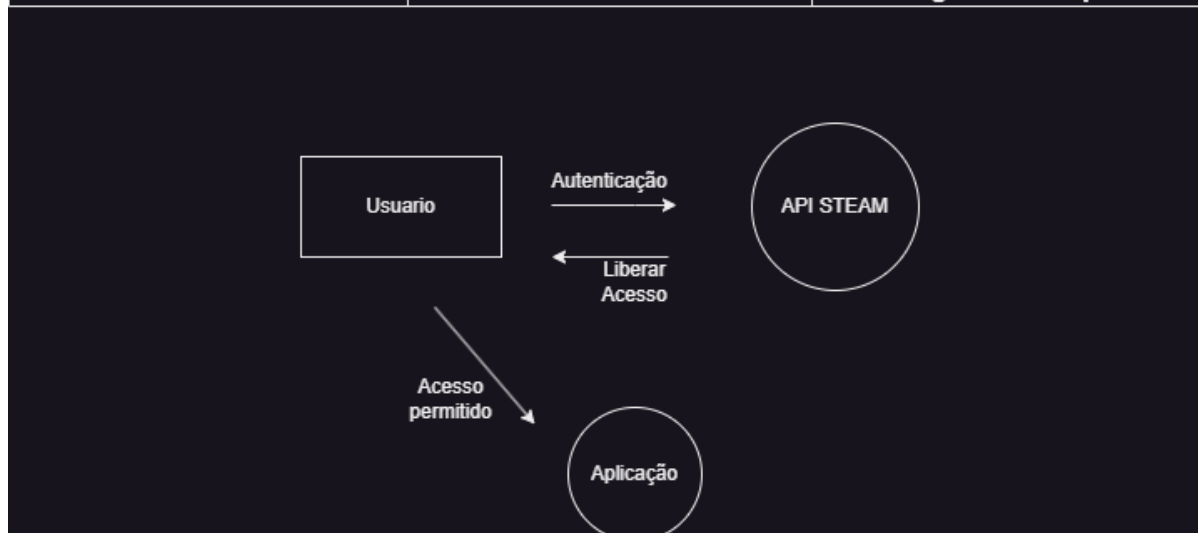


## 5 DFD Nível zero

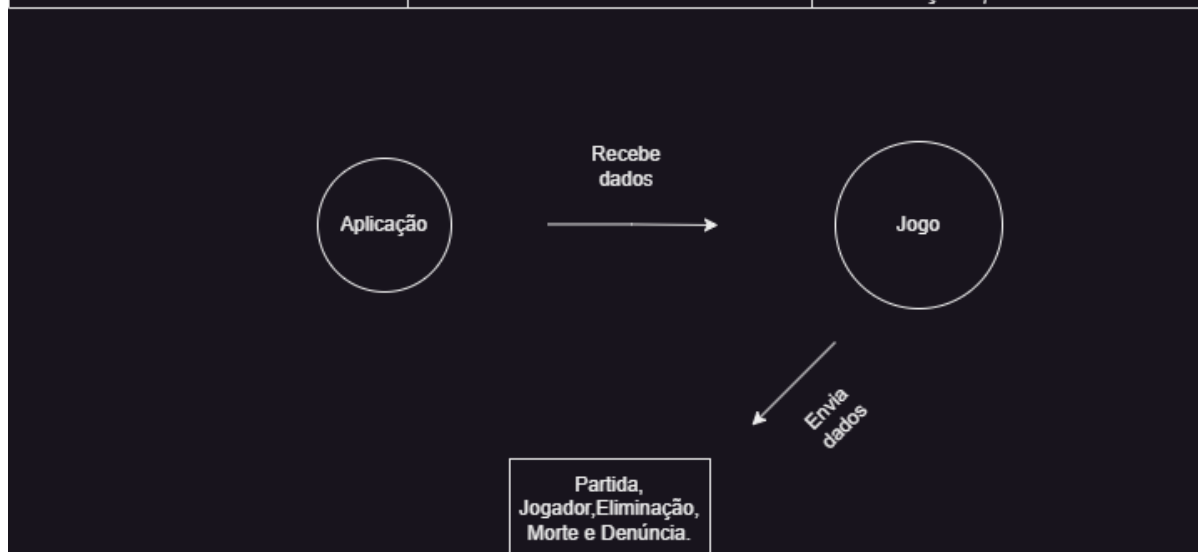


## 6 DFD POR EVENTO

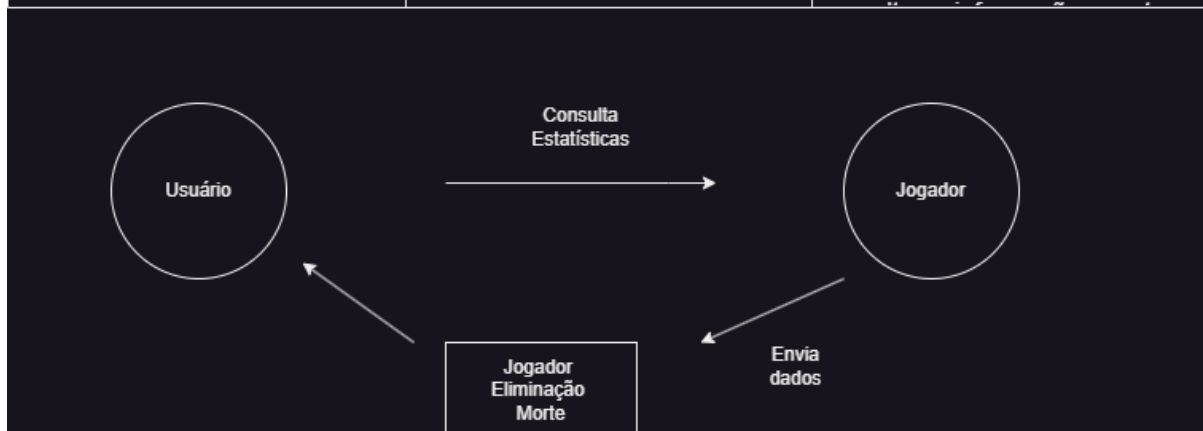
Nº	Evento	Descrição do evento
01	Cadastro do jogador	<b>Usuário</b> fornece dados ao processo <b>Cadastrar Jogador</b> , que armazena os dados em <b>Jogador e Grupo</b> .



Nº	Evento	Descrição do evento
02	Receber dados do jogo	o sistema de jogo envia informações sobre uma nova partida, jogadores envolvidos, eliminações, mortes e outras



Nº	Evento	Descrição do evento
03	Consultar Estatísticas do Jogador	Este evento é acionado quando um usuário solicita estatísticas específicas de um jogador, como número de eliminações, mortes, níveis, etc. O banco de dados



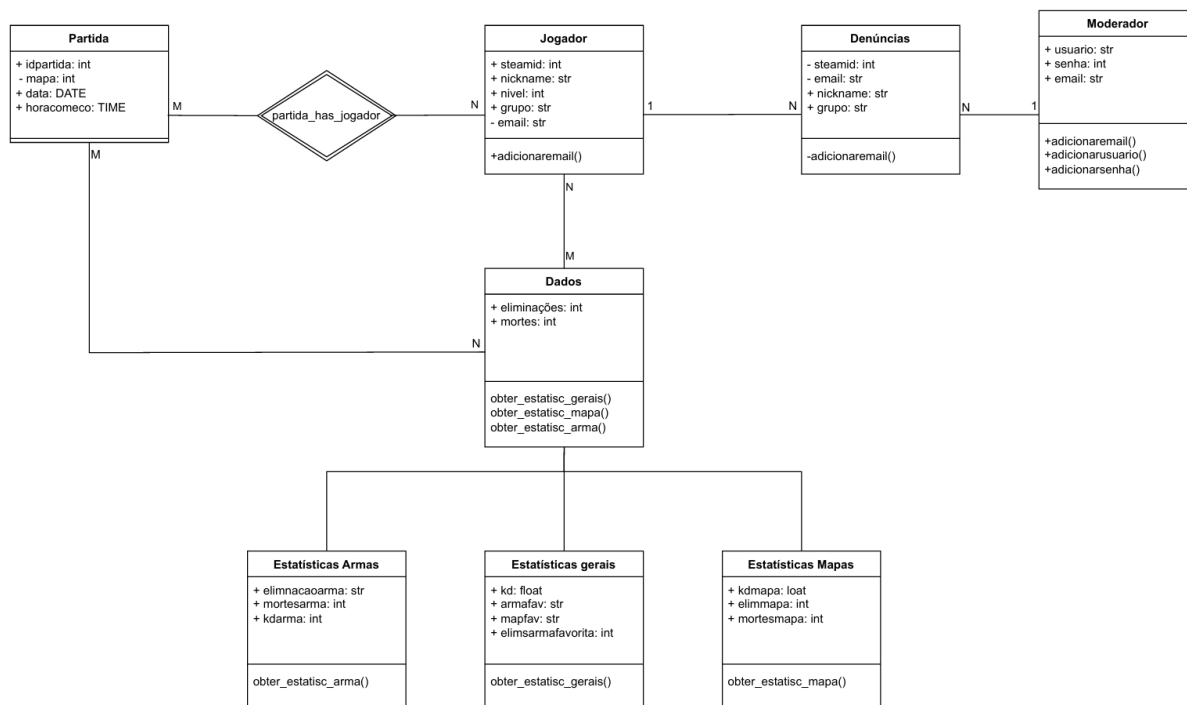
Nº	Evento	Descrição do evento
04	Consultar Estatísticas da Partida	Este evento ocorre quando um usuário solicita estatísticas de uma partida específica. O banco de dados consulta todas as informações relevantes sobre a





## 7 DER/ DIAGRAMA DE CLASSE

### 7.1 Artefato gráfico



### 7.2 Dicionário de dados (DDL)

Nome da tabela: jogador				
Descrição: Armazena informações do jogador				
Coluna	Tipo de dado	Tamanho	Restrições	Descrição
steamid	int		PK/ Not Null	Identifica e registra o jogador utilizando login Steam
grupo_idgrupo	int		FK/Not Null	Foreign Key da tabela grupo
email	varchar	45	Not Null	Após primeiro login na aplicação, registrar email do usuário
nickname	varchar	32	Not Null	Nickname Steam do jogador
nivel	int		Not Null	Nível do jogador no game

Nome da tabela: grupo				
Descrição: Armazena informações do grupo(clã) do jogador				
Coluna	Tipo de dado	Tamanho	Restrições	Descrição
idgrupo	int		PK/ Not Null	Identificador único do grupo
nomegrupo	varchar	20	Not Null	Nome do grupo
datacriacao	date		Not Null	Identifica a data de criação do grupo

Nome da tabela: moderador				
Descrição: Armazena informações do moderador				
Coluna	Tipo de dado	Tamanho	Restrições	Descrição
idmoderador	int		PK/ Not Null	Identificador único do moderador
nomemoderador	varchar	10	Not Null	Nome do moderador
senha	int		Not Null	Senha para acessar o espaço do moderador
emailmoderador	varchar	45	Not Null	E-mail do moderador para contato

Nome da tabela: denuncia				
Descrição: Armazena denúncias feitas pelos jogadores				
Coluna	Tipo de dado	Tamanho	Restrições	Descrição
iddenuncia	int		PK/ Not Null	Identificador único da denúncia
moderador_idmoderador	int		PK/FK/Not Null	Identifica o moderador responsável pela denúncia
textodenuncia	mediumtext		Not Null	Texto da denúncia feita pelo jogador
jogador_steamid	int		FK/Not Null	Identifica o jogador que criou a denúncia
partida_idpartida	int		FK/Not Null	Identifica a partida em que a denúncia ocorreu

Nome da tabela: partida				
Descrição: Armazena as partidas jogadas pelos usuários				
Coluna	Tipo de dado	Tamanho	Restrições	Descrição
idpartida	int		PK/ Not Null	Identificador único da partida
mapa_idmapa	int		PK/FK/Not Null	Identifica o mapa utilizado na partida
data	date		Not Null	Data em que ocorreu a partida
horacomeco	time		Not Null	Horário em que começou a partida

Nome da tabela: mapa				
Descrição: Armazena os mapas que podem ser utilizados em uma partida				
Coluna	Tipo de dado	Tamanho	Restrições	Descrição
idmapa	int		PK/ Not Null	Identificador único do mapa
mapa_idmapa	varchar	30	Not Null	Nome do mapa

Nome da tabela: partida_has_jogador				
Descrição: Tabela responsável por conectar jogadores a partidas				
Coluna	Tipo de dado	Tamanho	Restrições	Descrição
partida_idpartida	int		PK/FK/Not Null	
jogador_steamid	int		PK/FK/Not Null	

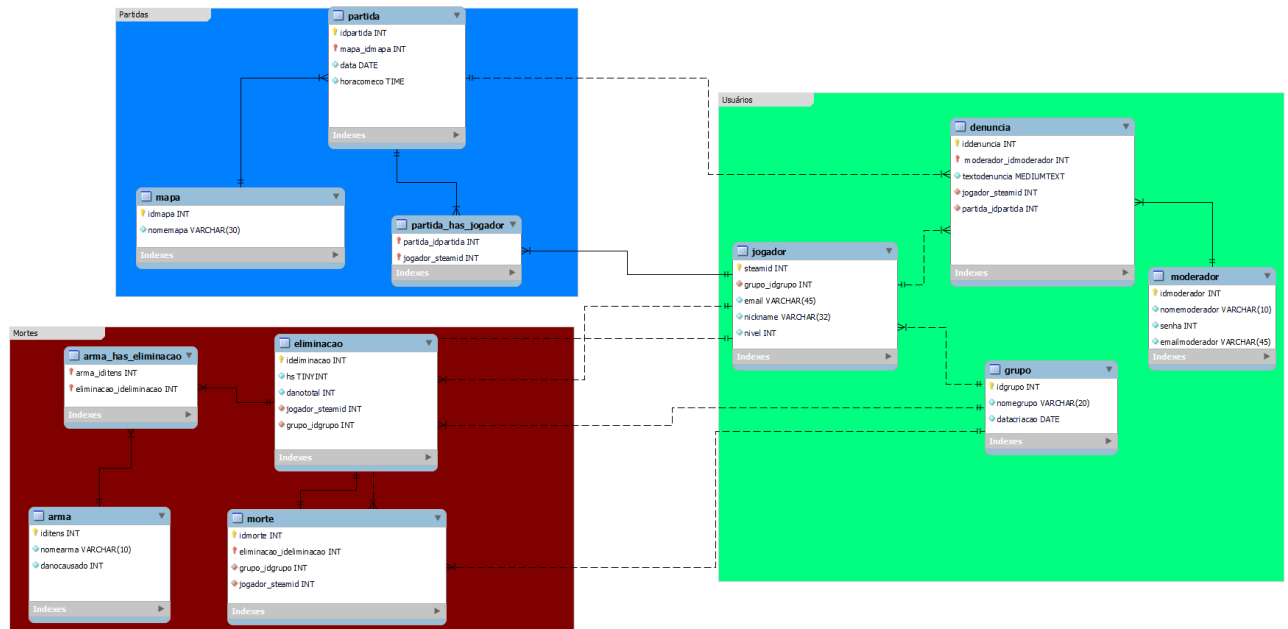
Nome da tabela: eliminacao				
Descrição: Armazena as eliminações feitas por um jogador				
Coluna	Tipo de dado	Tamanho	Restrições	Descrição
ideliminacao	int		PK/ Not Null	Identificador único da eliminação
hs	tinyint		Not Null	Identifica (Verdadeiro ou Falso) se a eliminação foi um headshot
danototal	int		Not Null	Quantidade de dano causado pelo jogador
jogador_steamid	int		FK/Not Null	Foreign Key que identifica o jogador responsável pela eliminação
grupo_idgrupo	int		FK/Not Null	Foreign Key que identifica o grupo do jogador responsável pela eliminação

Nome da tabela: morte				
Descrição: Armazena as mortes sofridas por um jogador				
Coluna	Tipo de dado	Tamanho	Restrições	Descrição
idmorte	int		PK/ Not Null	Identificador único da morte
eliminacao_ideliminacao	int		PK/FK/Not Null	Identifica a eliminação que causou a morte
grupo_idgrupo	int		FK/Not Null	Quantidade de dano causado pelo jogador
jogador_steamid	int		FK/Not Null	Foreign Key que identifica o jogador que morreu

Nome da tabela: arma				
Descrição: Armazena as armas e itens que podem causar uma eliminação no jogo				
Coluna	Tipo de dado	Tamanho	Restrições	Descrição
iditens	int		PK/ Not Null	Identificador único do item/arma
nomearma	int	10	varchar	Nome do item/arma
danocausado	int		Not Null	Dano padrão causado pela arma

Nome da tabela: arma_has_elimizacao				
Descrição: Tabela responsável por conectar armas com eliminações				
Coluna	Tipo de dado	Tamanho	Restrições	Descrição
arma_iditens	int		PK/FK/Not Null	Chave que identifica a
eliminacao_ideliminacao	int		PK/FK/Not Null	

### 7.3 Modelo comportamental (relação de entidade pai x filho)



## 8 POLÍTICAS DE TESTE

Para que a nossa aplicação se mantenha estável e segura durante o seu uso, foram desenvolvidas políticas internas de teste e de qualidade do produto. Uma delas envolve o acompanhamento de atualizações feitas no jogo por seus desenvolvedores, adequando a aplicação e o banco de dados às alterações nos dados fornecidos. Como as mudanças constantes são práticas comuns na indústria de jogos, ter esta política é essencial para qualquer aplicação de jogos que deseja manter-se em funcionamento sem bugs e erros. Sendo assim, é recomendável testar a aplicação junto com o servidor do jogo, criando partidas teste e acompanhando o envio de informações para o banco de dados. Não somente é necessário acompanhar o jogo em si, como quaisquer softwares que podem ser utilizados pelo cliente e que são fornecidos pela empresa do jogo, como a aplicação de servidor dedicado do game. Também é importante acompanhar a infraestrutura utilizada pelo banco de dados, considerando que existe um alto fluxo de dados sendo transmitidos por conta do grande número de partidas. Logo, é necessário acompanhar o armazenamento interno da máquina responsável pelo banco de dados, assim como o uso do sistema.

## 9 HOMOLOGAÇÃO

Primeiramente, a base do projeto necessita que o login do usuário seja validado através da API Steam, então é tratado com prioridade a conexão entre a nossa aplicação e os servidores Valve. A autenticação é feita através do OpenID, um protocolo de autenticação conhecido na indústria e utilizado por diversas empresas, como Amazon, Google, Microsoft e Steam. Além da autenticação, é importante que o servidor jogo esteja conectado com a base de dados através do plugin ou modificação que é fornecido pela comunidade (Open source). Assim como descrito na política de testes, partidas testes serão jogadas a fim de testar essa conexão. Antes disso, o funcionamento do banco de dados e suas respectivas tabelas, colunas e relações, serão testados antes de qualquer conexão com o servidor do jogo ou aplicação. Logo, os comandos SQL serão testados manualmente. Para a instalação e implementação, além da criação de senhas seguras para acessar o banco de dados, o cliente será treinado para criar o login dos administradores da plataforma, que será feito diretamente através do MySQL. É necessário ambientar os administradores em relação aos chamados e login, que possui uma indicação na página inicial para separá-lo da área dos jogadores. Por fim, com o design simples e intuitivo que foi criado, é esperado que os usuários não tenham dificuldades em relação ao acesso de suas informações.

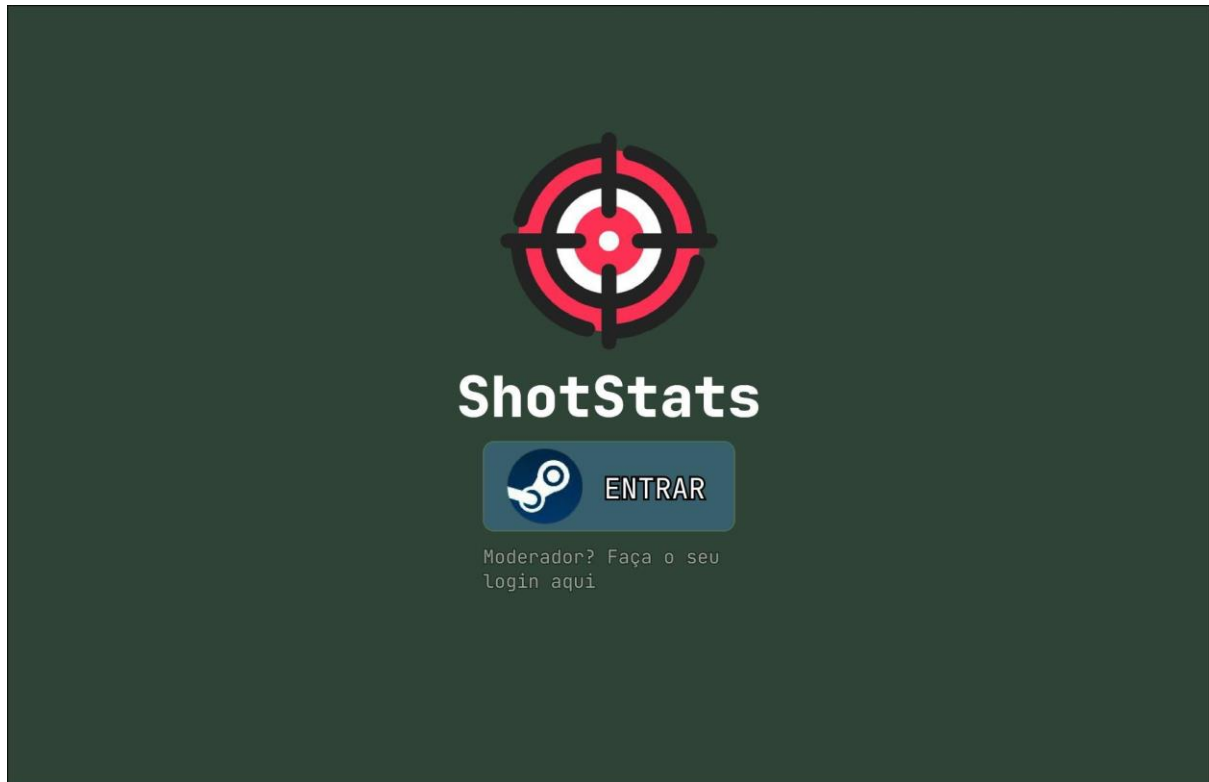
## 10 APLICAÇÃO PROTÓTIPO

Utilizamos o Figma que é uma ferramenta de design colaborativo baseada na web, utilizada principalmente para design de interfaces de usuário (UI), design de experiência do usuário (UX) e prototipagem. Ela permite que designers, desenvolvedores e outras partes interessadas trabalhem juntos em tempo real.

## 10.1 telas funcionais

### [Telas Funcionais](#)

#### Tela 1 - Login Moderador:

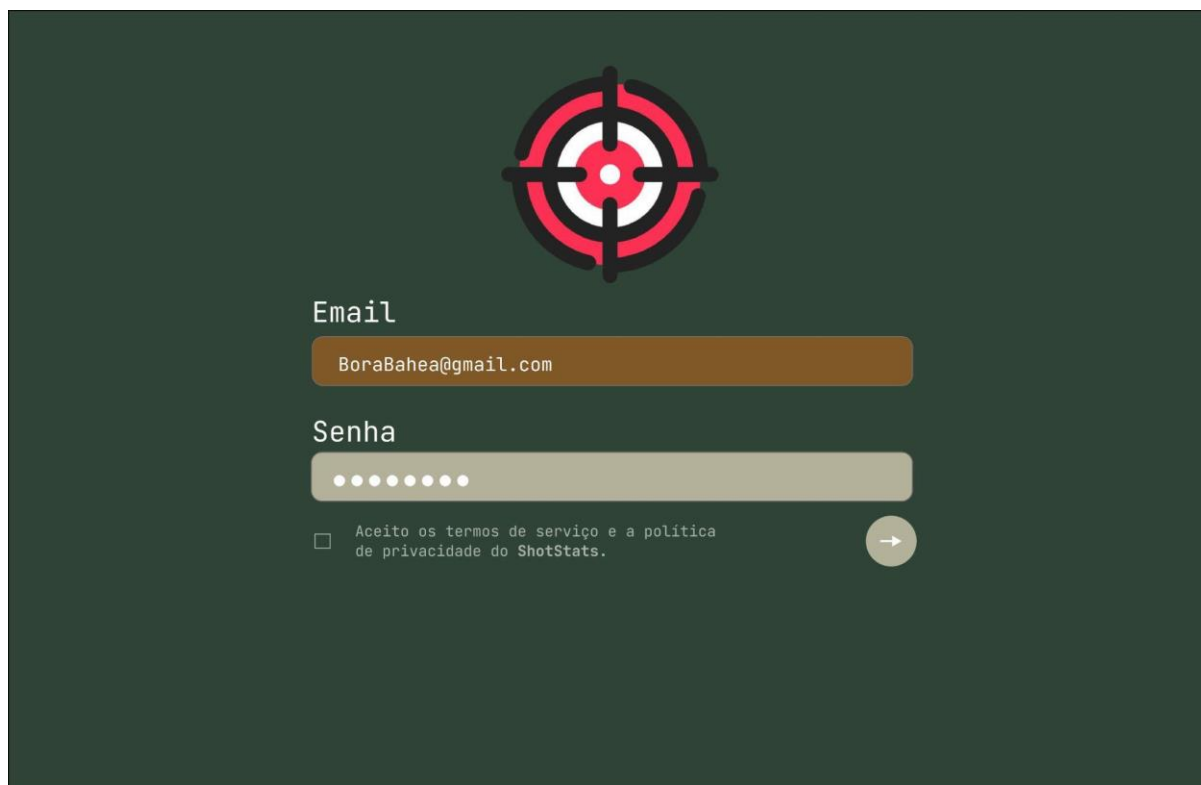


**Descrição Login:** Botão com o logotipo do Steam e a palavra "ENTRAR". Este botão é usado para realizar o login através da plataforma Steam.

**Descrição Logo:** Logo representando um alvo com um ponto de mira vermelho e branco, seguido do nome do aplicativo "ShotStats".

**Descrição Moderador:** Apenas para moderadores.

## Tela 2 - login Usuario:

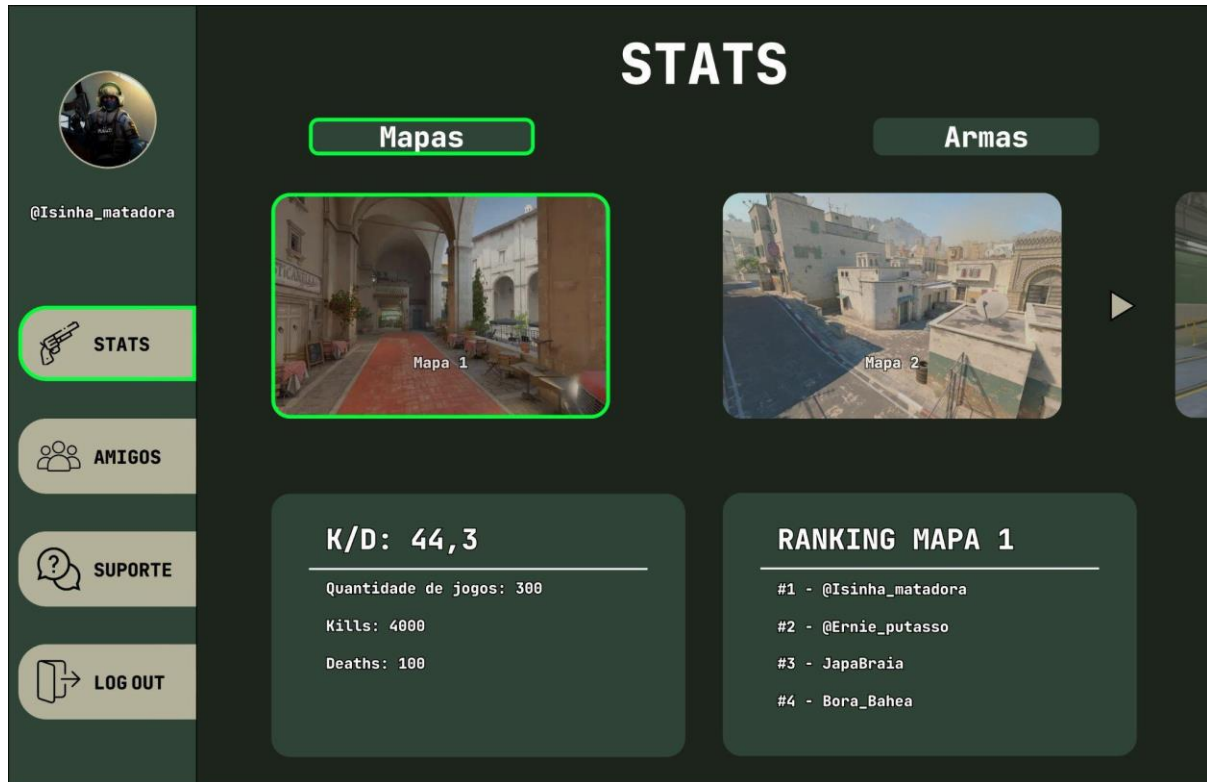
The image shows a login screen for ShotStats. At the top center is a logo consisting of a red bullseye with a black crosshair. Below the logo, the word "Email" is displayed in a light gray font. Underneath is a brown input field containing the text "BoraBahea@gmail.com". Below the email field, the word "Senha" is displayed in a light gray font. Underneath is a light gray input field with eight dots, indicating a password. Below the password field, there is a checkbox followed by the text "Aceito os termos de serviço e a politica de privacidade do ShotStats." To the right of the checkbox is a circular button with a right-pointing arrow.

**Login do cliente:** email de cadastro no ShotStats

**Senha do usuário:** senha do usuário no ShotStats

**Termo de serviço:** Preencha a caixinha para aceitar os termos de serviços, e as políticas de privacidade.

## Tela 3 - Menu:



## Itens:

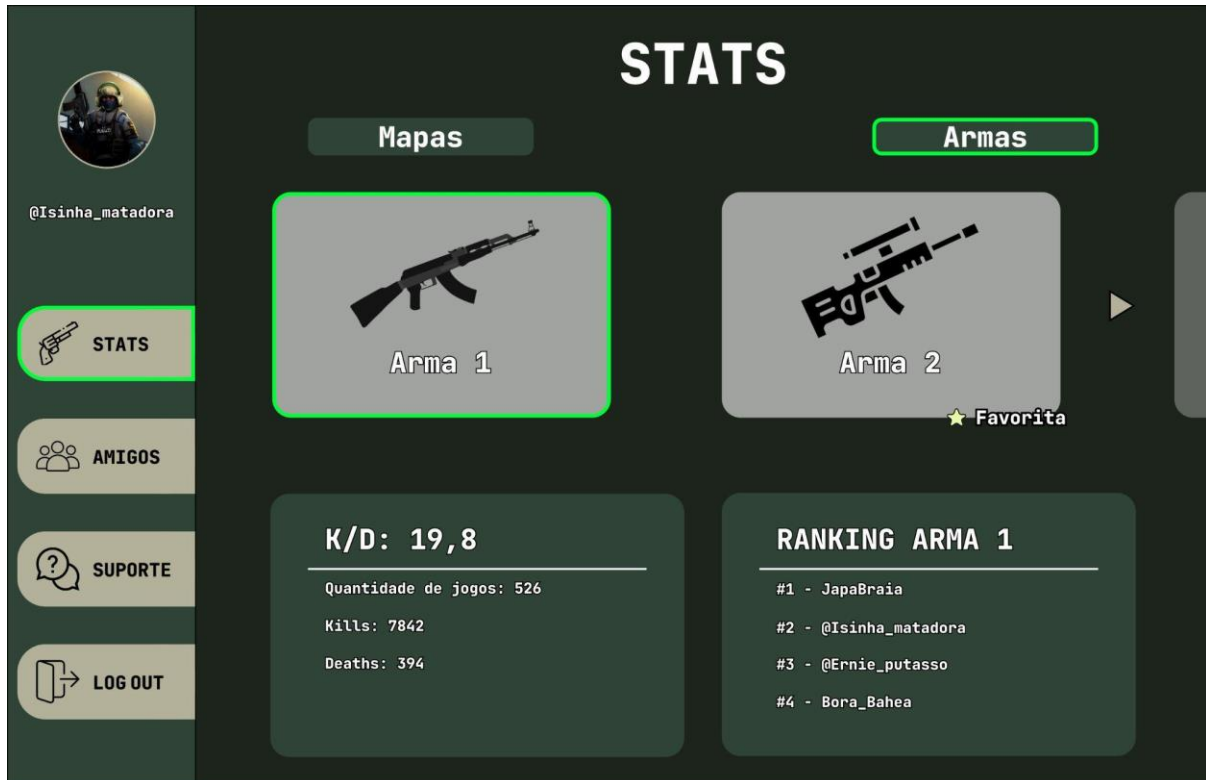
- **Stats:** Ícone de uma arma e texto "STATS" (selecionado).
- **Amigos:** Ícone de dois usuários e texto "AMIGOS".
- **Suporte:** Ícone de um balão de fala e texto "SUPORTE".
- **Log Out:** Ícone de porta e texto "LOG OUT"

## Descrição:

- **Mapas:** mapas do game(Mapa 1 e Mapa 2).
- **Armas:** tabela de armas para ver seu arsenal
- **K/D Ratio:** Taxa de abates/mortes.
- **Quantidade de Jogos:** quantidade de partidas jogadas.
- **Kills:** quantidade de personagens matados
- **Deaths:** quantidades de mortes no jogo
- **RANKING:** ranqueamento do top 4 do mapa 1



## Tela 4 - Armas:



## Perfil do Usuário:

- **Localização:** Parte superior esquerda.
- **Descrição:** Imagem de perfil do usuário com o nome de usuário @Username.

## Menu de Navegação:

- **Localização:** Barra lateral esquerda.
- **Itens:**
  - **Stats:** Ícone de uma arma e texto "STATS" (selecionado).
  - **Amigos:** Ícone de dois usuários e texto "AMIGOS".
  - **Suporte:** Ícone de um balão de fala e texto "SUPORTE".
  - **Log Out:** Ícone de porta e texto "LOG OUT".

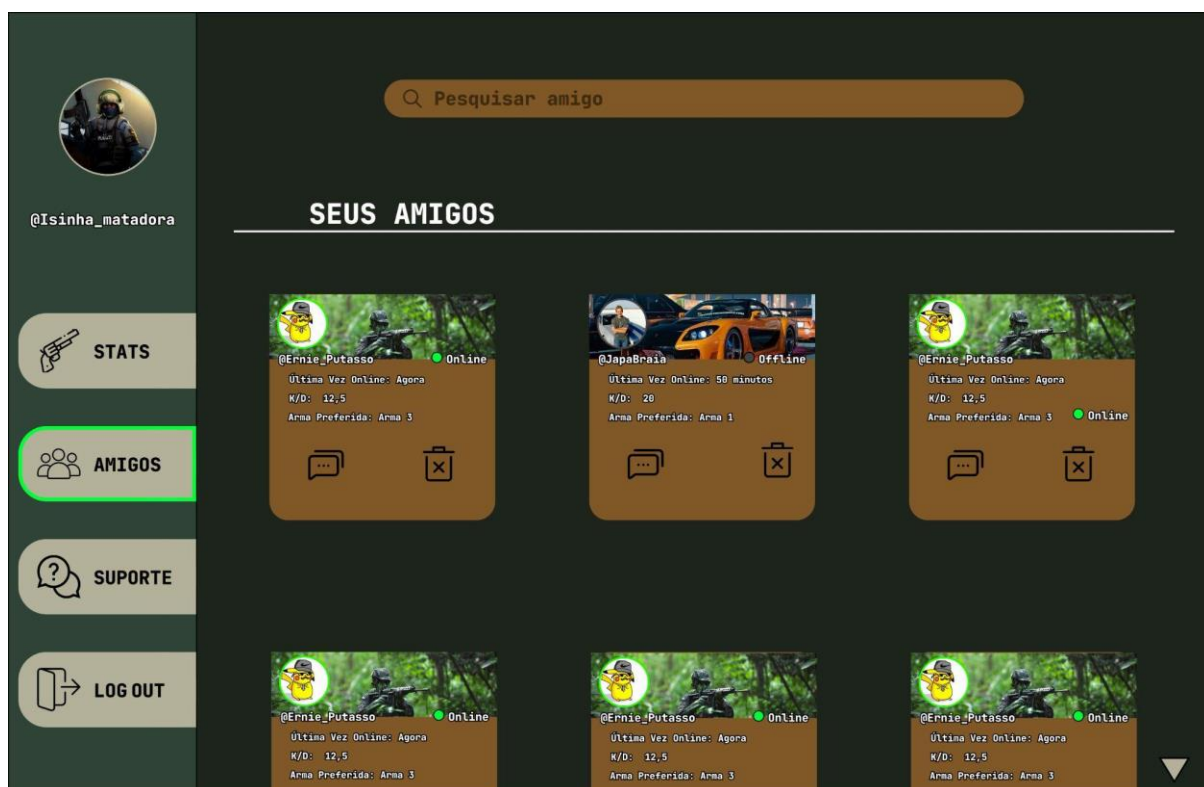
## Seções Principais:

- **Localização:** Parte central e direita da tela.
- **Descrição:**
  - **Mapas:** Contém duas imagens de armas (Arma 1 e Arma 2). A Arma 2 está marcada como "Favorita".
  - **Armas:** Contém duas imagens de armas (Arma 1 e Arma 2).

## Estatísticas do Usuário:

- **Localização:** Parte inferior central.
- **Ranking ARMA 1:** ranqueando das armas de acordo com os usuários
- **Descrição:**
  - **K/D Ratio:** Taxa de kills/deaths, 19,8.
  - **Quantidade de Jogos:** 526.
  - **Kills:** 7842.
  - **Deaths:** 394.

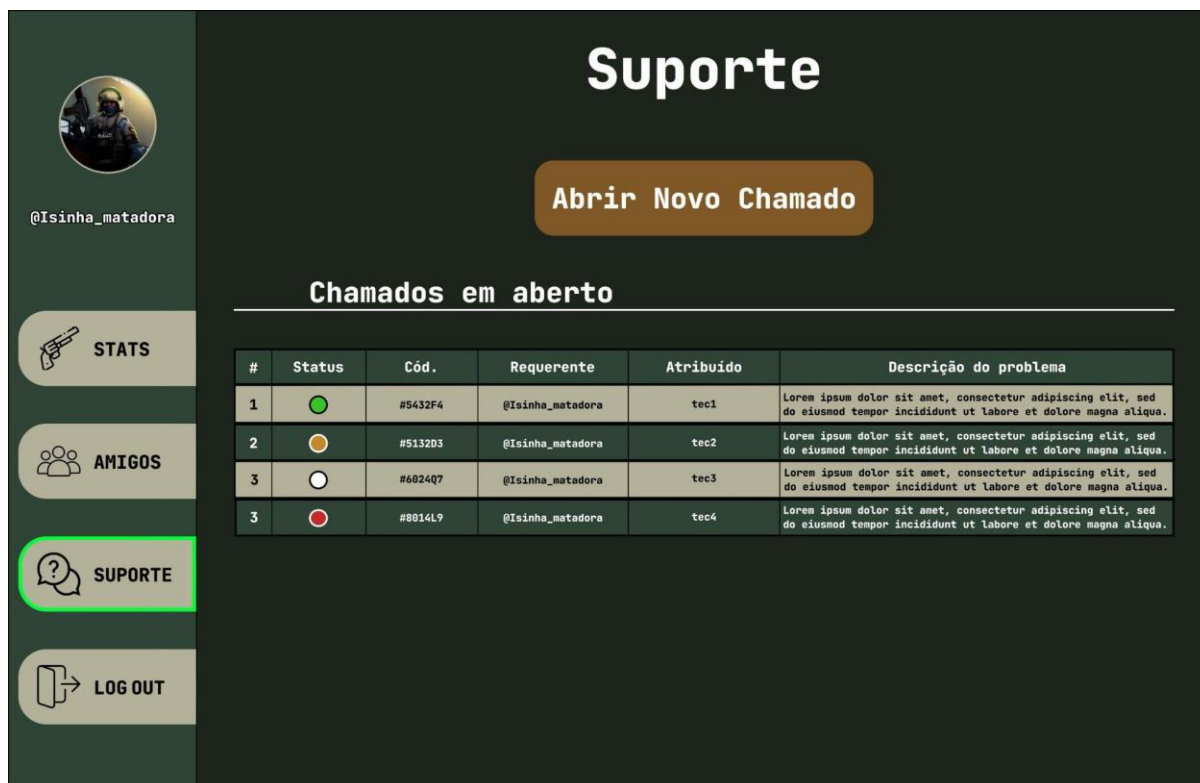
## Tela 5 - Amigos:



## Seções Principais:

- **Localização:** Parte central e direita da tela.
- **Descrição:**
  - **Seus amigos:** Perfil de seus amigos adicionados na lista de amigos.
  - **Barra de pesquisa de amigos:** para filtrar um amigo específico na lista.

## Tela 6 - Suporte:



- **Localização:** Parte central da tela
- **Descrição:**
  - **Botão de abrir novo chamado:** Abre um novo requerimento que vai ser direcionado aos moderadores onde irão tratar a ocorrência.
  - **Chamados em abertos:** Composto por # (ordem de chegada), Status: verde(resolvido), amarelo(em andamento), branco(não lido), vermelho(negado), Cód: código do requerimento, Requerente(Username do usuário), Atribuído (moderador responsável) e Descrição do problema (onde mostra o que precisa ser resolvido)

## 11 CONCLUSÃO

A criação da "ShotStats" exemplifica como uma aplicação bem projetada pode agregar valor significativo aos jogadores de FPS, oferecendo-lhes ferramentas para acompanhar e melhorar seu desempenho. A combinação de uma interface visual intuitiva com funcionalidades robustas não apenas enriquece a experiência do usuário, mas também promove um ambiente competitivo saudável e informativo. Este projeto destaca a importância de uma abordagem centrada no usuário na engenharia de software e demonstra como a tecnologia pode ser utilizada para aprimorar experiências em diversos contextos, incluindo o entretenimento digital.