



**UNIVERSIDADE FEDERAL DA PARAÍBA**  
**CENTRO DE INFORMÁTICA**  
**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

ENRIQUE PEDROSA SOUSA  
VICTOR GABRIEL BATISTA REINALDO  
WESLEY PAULO DA SILVA  
JOSÉ ALISSON DA SILVA

**PROJETO TRACKAUDIO PROJECT**  
Documento de Requisitos

João Pessoa  
2024

## Histórico de Revisões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor(es)</b>
<08/08/2024>	<0.01>	<Adicionamos ao documento a descrição do uso de 2 técnicas de elicitação e resultados a partir delas>	<Enrique Pedrosa Sousa, Wesley Paulo da Silva, José Alisson da Silva e Victor Gabriel Batista Reinaldo>
<29/08/2024>	<0.02>	<Adicionamos ao documento o diagrama de classe de análise>	<Enrique Pedrosa Sousa, Wesley Paulo da Silva, José Alisson da Silva e Victor Gabriel Batista Reinaldo>
<02/09/2024>	<0.03>	<Adicionamos ao documento a seção de histórico de revisões>	<Enrique Pedrosa Sousa, Wesley Paulo da Silva, José Alisson da Silva e Victor Gabriel Batista Reinaldo>

## SUMÁRIO

<b>1. Introdução.....</b>	<b>5</b>
1.1 Propósito do Documento.....	5
1.2 Escopo do Documento.....	5
1.3 Acrônimos e Abreviações.....	5
1.4 Visão Geral do Documento.....	5
<b>2. Descrição Geral.....</b>	<b>6</b>
2.1 Perspectiva do Produto.....	6
2.2 Usuários do Software.....	6
2.2.1 Administradores.....	6
2.2.2 Clientes.....	6
2.3 Suposições e Restrições Gerais.....	6
<b>3. Requisitos.....</b>	<b>7</b>
3.1 Requisitos Funcionais.....	7
3.1.1 [RF001] Sistema de Criação de Contas Novas.....	7
3.1.2 [RF002] Sistema de login.....	7
3.1.3 [RF003] Registro de Scores.....	7
3.1.4 [RF004] Sistema de Dificuldade.....	8
3.1.5 [RF005] Filtro de Músicas.....	9
3.1.7 [RF006] Sistema de Vidas.....	9
3.1.6 [RF007] Sistema de Dicas.....	9
3.1.8 [RF008] Seleção de Sugestões.....	10
3.1.9 [RF009] Sistema Auto Sustentável.....	10
3.1.10 [RF010] Personalização de Jogo.....	10
3.1.11 [RF011] Escolha de Recomendação.....	10
3.2 Requisitos Não-Funcionais.....	10
3.2.1 [RNF001] Desempenho do Sistema.....	10
3.2.2 [RNF002] Repertório Musical do Jogo.....	11
3.2.3 [RNF003] Funcionamento do Jogo.....	11
3.2.4 [RNF004] Responsividade do Sistema.....	11

3.2.5 [RNF005] Acesso à API do YouTube.....	11
3.2.6 [RNF006] BackEnd do Sistema.....	11
3.2.7 [RNF007] FrontEnd do Sistema.....	11
3.2.8 [RNF008] Banco de Dados do Sistema.....	12
<b>4. Diagrama de Caso de Uso.....</b>	<b>12</b>
<b>5. Descrição dos Casos de Uso.....</b>	<b>13</b>
5.1 O Usuário Joga o Jogo.....	13
5.2 O Usuário Cria uma Conta.....	13
5.3 O Usuário Faz Login.....	13
5.4 O Usuário Edita Seu Login.....	14
5.5 O Usuário Vê os Dados da Conta.....	14
<b>6. Descrição da interface com o usuário.....</b>	<b>15</b>
6.1 Telas e descrições.....	15
6.1.1 Tela inicial do jogo.....	15
6.1.2 Tela principal do jogo.....	16
6.1.3 Tela de login.....	17
6.1.4 Tela de cadastro.....	18
6.1.5 Tela de recuperação de senha.....	19
6.1.6 Tela do administrador.....	20
<b>7. Rastreabilidade:.....</b>	<b>21</b>
7.1 Requisitos Funcionais/Não funcionais e Requisitos Funcionais/Não funcionais:.....	21
7.2 Casos de uso e Requisitos:.....	23
<b>8. Diagrama de Contexto:.....</b>	<b>24</b>
<b>9. Diagrama de Fluxo de Dados - Nível 1:.....</b>	<b>25</b>
<b>10. Diagrama de Classe de Análise:.....</b>	<b>26</b>
<b>11. Elicitação de Requisitos.....</b>	<b>27</b>
11.1. Brainstorm.....	27
11.2. Workshop de Requisitos.....	27
11.3. Considerações Finais.....	27

# 1. Introdução

## 1.1 Propósito do Documento

Este documento possui como objetivo descrever todos os requisitos do projeto TrackAudio Project que o jogo deve atender de maneira detalhada.

## 1.2 Escopo do Documento

Este documento serve para as partes interessadas no projeto, bem como o professor Raoni e os envolvidos no desenvolvimento dele. Serve para entender de maneira clara e coesa o que deve ser desenvolvido no jogo, visando obter ótimos resultados. Os clientes também podem fazer uso do documento até para verificarem se o jogo atendeu às expectativas e objetivos esperados.

## 1.3 Acrônimos e Abreviações

- **TAP** - TrackAudio Project
- **RF** - Requisito Funcional
- **RNF** - Requisito Não-Funcional
- **CU** - Caso de Uso

## 1.4 Visão Geral do Documento

O documento abaixo de requisitos vai seguir a estrutura abaixo:

- **Introdução:** Tal tópico passa uma visão geral sobre o documento de requisitos. Aqui demonstramos o escopo, propósito e afins.
- **Descrição Geral:** Uma explicação sobre o jogo em si a ser produzido.
- **Requisitos do Projeto:** Requisitos funcionais e não-funcionais que devem estar presentes no software.
- **Diagrama Geral dos Casos de Uso:** Fornece um diagrama que demonstra os casos de uso do aplicativo.
- **Descrição Geral dos Casos de Uso:** Fornece a descrição de casos de uso no diagrama de casos de uso.
- **Descrição da Interface com o Usuário:** Contém imagens do protótipo do sistema.
- **Rastreabilidade:** Contém uma tabela com a relação entre os requisitos e os casos de uso.
- **Diagrama de Contexto:** Contém o diagrama de fluxo de dados de nível 0.
- **Diagrama de Fluxo de Dados:** Contém o diagrama de fluxo de dados expandido (nível 1).
- **Plano de Elicitação de Requisitos:** Documento desenvolvido na parte inicial do projeto que descreve a abordagem geral e os métodos

específicos que serão utilizados para coletar requisitos do sistema ou projeto.

## 2. Descrição Geral

### 2.1 Perspectiva do Produto

Nossa equipe enxerga esse produto como uma oportunidade de mercado. Sabendo que o mercado de jogos eletrônicos é o maior mercado de entretenimento do mundo e que cresce ano após ano é interessante para qualquer pessoa tentar entrar nesse mercado. Nossa equipe, inspirada em um jogo existente de sucesso, resolveu criar um jogo único baseado numa insatisfação nossa e de alguns amigos em comum que buscávamos um jogo que tocasse músicas de jogos aleatórios e que o jogador tivesse o dever de adivinhar ou o jogo ou o nome da música pelo toque. Essas músicas, dependendo do nível de dificuldade, poderiam estar lentas, invertidas, rápidas ou serem de jogos menos conhecidos.

### 2.2 Usuários do Software

#### 2.2.1 Administradores

Estes, serão aqueles que irão trabalhar no jogo para adicionar ou remover músicas do banco de dados.

#### 2.2.2 Clientes

Estes serão os jogadores que irão jogar o jogo. Em geral, o nosso público-alvo será pessoas interessadas em jogos e que já jogaram diversos jogos eletrônicos diferentes, e assim foram expostas a diversas trilhas sonoras diferentes.

### 2.3 Suposições e Restrições Gerais

Muitas trilhas sonoras de jogos não são disponibilizadas para uso livre, possivelmente impedindo sua inclusão no nosso projeto. Assumimos que existam músicas relevantes de uso livre suficientes para que possamos criar um repertório de ao menos duzentas músicas.

## 3. Requisitos

Os requisitos terão classificações descritas abaixo:

- **Imprescindível:** Requisito que é parte essencial do funcionamento do jogo. Se esse requisito não for cumprido, o jogo não funcionará.
- **Importante:** O jogo pode funcionar sem esses requisitos, porém não funcionará de uma forma agradável.

- **Seria bom ter:** O jogo pode funcionar de maneira agradável sem estes requisitos. Estes são apenas para tornar a experiência do usuário cada vez melhor.

### 3.1 Requisitos Funcionais

#### 3.1.1 [RF001] Sistema de Criação de Contas Novas

Descrição: O jogador deve ter a opção de criar uma conta com e-mail, apelido e senha para poder registrar o histórico de pontuações (scores) que o jogador obteve ao longo das partidas jogadas. Após cadastrar os dados, um e-mail de confirmação é enviado ao e-mail cadastrado pelo usuário para validar a conta.

Prioridade: Imprescindível

#### 3.1.2 [RF002] Sistema de login

Descrição: O jogador que tem uma conta cadastrada deve poder entrar em sua conta através de uma tela de login. Para o login, será utilizado email e senha.

Prioridade: Imprescindível

#### 3.1.3 [RF003] Registro de Scores

Descrição: O jogo deve ter um sistema de gravação de scores(pontuações) por conta. Esses scores são obtidos após uma partida jogada pelo usuário em que ele perdeu todas as vidas ou quis parar seu streak (sequência de acertos). Cada vez que o usuário acerta uma música, ele ganha uma pontuação, que depende da quantidade de corações que ele gastou antes de acertar a música e do nível de dificuldade que ele estiver jogando. A quantidade de pontos ganha após um acerto é definida pela seguinte fórmula:

$$1000 * \left( 1 + \frac{11 - \text{vidas\_config}}{10} * 9 + \frac{\text{vidas\_restante}}{\text{vidas\_config}} \right) * \text{multiplicador\_de\_dificuldade}$$

Dado que:

- `vidas_config` é quantas vidas o usuário escolheu ter na configuração. A quantidade máxima é definida no RF006.
- `vidas_restante` é quantas vidas o usuário ainda tinha na hora que acertou
- `multiplicador_de_dificuldade` é um multiplicador extra definido pela dificuldade escolhida pelo usuário. Seus valores são definidos no RF004.

Caso o usuário selecione vidas infinitas, a seguinte fórmula será utilizada:

$$1000 * \text{multiplicador\_de\_dificuldade}$$

Prioridade: Importante

### 3.1.4 [RF004] Sistema de Dificuldade

Descrição: O jogo deve ter sistema de dificuldades (variando entre fácil, médio, difícil e extremo), onde as músicas podem ser alteradas ao ponto de virem invertidas, lentas ou muito rápidas.

No nível fácil, o jogo tocará músicas de jogos mais conhecidos e músicas mais icônicas, aquelas que normalmente até quem não joga muito videogame provavelmente já ouviu, como músicas dos jogos de Mario clássicos. Nesse nível, o usuário só precisará informar o nome do jogo da música que está sendo tocada.

No nível médio, terão músicas, ainda bem conhecidas dentro da comunidade gamer, mas nem tanto num contexto de público-geral. Nesse nível, deve jogar quem é de fato um jogador e membro da comunidade gamer.

No nível difícil, terão músicas bem menos conhecidas e o jogador passará a ser forçado a dizer o nome da música e não mais o jogo.

No nível extremo, o jogador contará com qualquer música do banco de dados, até as mais desconhecidas. Além disso, ele terá de lidar com músicas modificadas que podem estar invertidas, lentas ou ultra rápidas de maneira aleatória. Ele ainda precisará dizer o nome da música.

Observação: A quantidade de vidas a escolher será independente do nível de dificuldade que o usuário escolher (ele pode jogar com dificuldade máxima e ter apenas uma vida extra do mesmo jeito que ele pode ter vidas infinitas e ainda jogar na dificuldade máxima).

Observação 2: Cada nível tem seu próprio multiplicador de dificuldade, que é um valor usado para calcular a pontuação (quanto maior o multiplicador, maior será a pontuação). Os valores são:

- Fácil: 1
- Médio: 1.5
- Difícil: 3
- Extremo: 5

Prioridade: Importante

### 3.1.5 [RF005] Filtro de Músicas

Descrição: O jogo deve ter filtragem de músicas. Por exemplo, se o usuário não quiser adivinhar músicas de RPG, ele pode filtrar e cortar consequentemente. Ao usuário entrar no jogo e fazer login (ou não), ele terá uma tela onde ele pode escolher dificuldade, enumerar quantidade de vidas e filtrar os tipos de músicas que ele quer. Esses filtros podem ser por ano (pode filtrar de anos 90 para frente ou 80 para trás e afins) e por gênero de jogo (RPG, esportes, FPS e etc).

Prioridade: Importante



### 3.1.7 [RF006] Sistema de Vidas

Descrição: O jogo deve ter um sistema de vidas para que o usuário possa ter mais chances após erros durante uma tentativa de acerto de música (um chute). Essa quantidade de vidas (corações) independe do nível de dificuldade que o usuário for jogar. A quantidade de vidas personalizadas varia entre nenhuma vida extra (errou, perdeu) até 10 vidas extras.

Quando o usuário estiver jogando, ele contará com a quantidade de vidas extras que ele escolheu dentre as disponíveis. Durante o jogo, quando ele tenta adivinhar uma música (um chute), ele gasta uma vida extra a cada erro. Se ele perder todas as vidas, é game over (fim de jogo) e o score final é registrado. Se ele acertar a música antes de perder todas as vidas, todas as vidas que ele gastou se renovam para a próxima música que ele for adivinhar.

Prioridade: Importante

### 3.1.6 [RF007] Sistema de Dicas

Descrição: O jogo deve ter sistemas de dica única que renovam a cada 20 acertos depois de gasto em algum momento da partida. Por exemplo, durante sua jogatina, o usuário de começo de jogo terá suas vidas (os corações) e uma dica que funciona como uma vida extra especial. Essa dica serve como um coringa, um trunfo, algo que salvará o usuário de um game over.

Ao ser usada, você recebe uma pequena descrição sobre o jogo em que toca essa música junto com o ano de lançamento do jogo, plataformas lançadas (pc, xbox, nintendo e afins) uma imagem referente ao jogo para ajudar. Ao utilizar a dica, você precisará acertar mais 20 músicas para poder ter uma dica de volta. A ideia dela é ser especial, não algo comum que possa ser usado sempre.

Prioridade: Importante

### 3.1.8 [RF008] Seleção de Sugestões

Descrição: O jogo deve ter uma caixa de seleção de sugestões baseada no que o usuário está começando a escrever para tentar adivinhar a música, seja nos níveis onde o usuário deve adivinhar apenas o nome do jogo ou no nível onde ele deve adivinhar o nome da música. Bem semelhante ao sistema de sugestão de palavras do celular.

Caso o jogo ou música que o usuário estiver começando a escrever não corresponder com nenhuma música ou jogo do banco de dados, o sistema deve indicar na caixa de seleção vazio, mostrar nenhuma opção de resposta disponível.

Prioridade: Imprescindível

### 3.1.9 [RF009] Sistema Auto Sustentável

Descrição: A página do jogo deverá conter anúncios para o lucro e manutenção do site. Serão exibidos anúncios nas bordas da tela do site durante a gameplay e nas telas intermediárias (menu, login e afins) no cantos para não atrapalhar o conteúdo das telas de fato.

Prioridade: Imprescindível

### 3.1.10 [RF010] Personalização de Jogo

Descrição: O usuário deve primeiramente escolher a quantidade de vidas, dificuldade do jogo e se deseja filtrar alguma música ou não antes de começar a jogar.

Prioridade: Imprescindível

### 3.1.11 [RF011] Escolha de Recomendação

Descrição: O usuário deve, obrigatoriamente, escolher qual música/jogo é clicando nas recomendações da caixa de seleção de sugestões baseada no que o usuário estiver escrevendo naquele momento. E só vai sugerir algo, se o que o usuário estiver escrevendo bater com alguma música presente no banco de dados. Além disso, essa caixa deve alterar para ou seleção de jogo ou seleção do nome da música baseado no que o usuário escolheu previamente.

Prioridade: Imprescindível

## 3.2 Requisitos Não-Funcionais

### 3.2.1 [RNF001] Desempenho do Sistema

Descrição: O jogo deverá ter um desempenho elevado, carregando músicas e elementos visuais em até 2 segundos(desconsiderando a velocidade de conexão com o usuário).

Prioridade: Importante

### 3.2.2 [RNF002] Repertório Musical do Jogo

Descrição: O jogo deverá dispor de no mínimo 200 músicas em seu repertório. Variando em músicas de diversos gêneros, anos, grau de popularidade e nicho.

Prioridade: Seria bom ter

### 3.2.3 [RNF003] Funcionamento do Jogo

Conteúdo: O sistema deverá estar em pleno funcionamento 24/7, ou no mínimo das 8:00 às 00:00.

Prioridade: Importante

#### 3.2.4 [RNF004] Responsividade do Sistema

Conteúdo: O conteúdo visual do jogo (a página) deve se adaptar ao tamanho da tela do dispositivo onde está sendo exibida. Ou seja, independente da resolução do dispositivo, a página deve se adaptar e conseguir exibir todo o conteúdo.

Prioridade: Imprescindível

#### 3.2.5 [RNF005] Acesso à API do YouTube

Descrição: O sistema deve dispor de acesso à API do YouTube para o acesso das músicas que serão utilizadas no jogo.

Prioridade: Imprescindível

#### 3.2.6 [RNF006] BackEnd do Sistema

Descrição: O sistema deverá contar com o BackEnd desenvolvido em Django

Prioridade: Imprescindível

#### 3.2.7 [RNF007] FrontEnd do Sistema

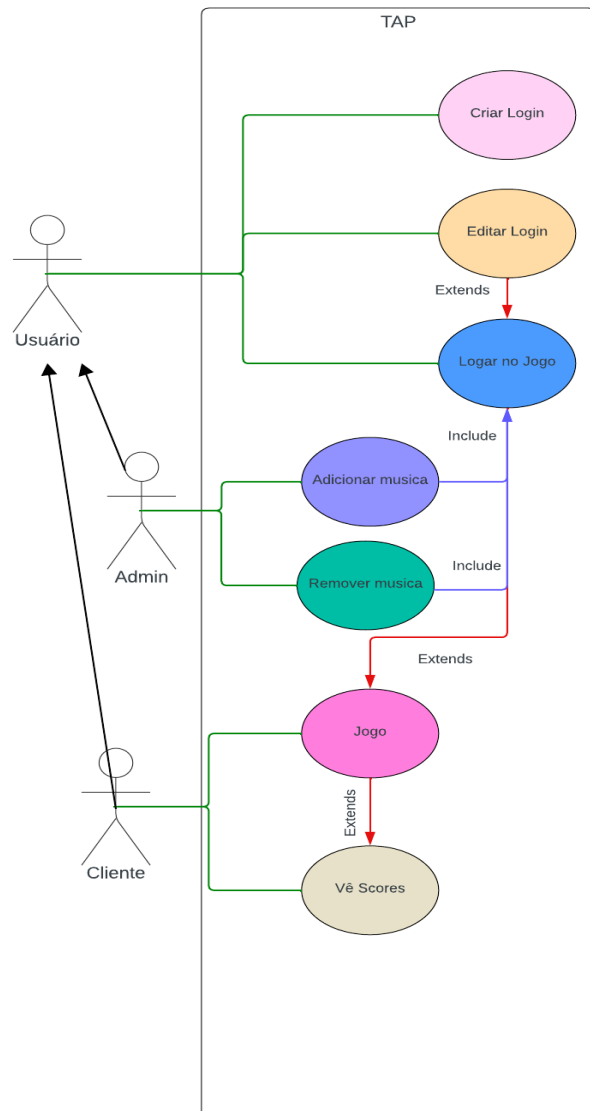
Descrição: O sistema deverá contar com o FrontEnd em HTML/CSS/JavaScript usando o framework Angular.

#### 3.2.8 [RNF008] Banco de Dados do Sistema

Descrição: O jogo deverá conter um banco de dados PostgreSQL onde estará localizado o link de cada música separadamente e os dados dos usuários.

## 4. Diagrama de Caso de Uso

Figura 1



(Fonte: Acervo da equipe)

## 5. Descrição dos Casos de Uso

### 5.1 O Usuário Joga o Jogo

**Nome:** O Usuário Joga o Jogo

**Identificador:** CU001

**Importância:** Imprescindível

**Sumário:** O processo principal do jogo. Inclui o processo de entrar no jogo preparação para o jogo e jogar em si.

**Atores:** Cliente

**Pré condições:** Não possui

**Fluxo Principal:**

1. O cliente abre o jogo no Web Site
2. Se o cliente tiver conta, ele pode fazer login ou entrar sem fazer login
3. Se o cliente não tiver conta, ele pode criar uma conta ou não
4. O cliente entra na tela de personalização do jogo. Onde ele escolhe a dificuldade do jogo, a quantidade de vidas, se deseja filtrar as músicas ou não
5. Por fim, o cliente começa a jogar

## 5.2 O Usuário Cria uma Conta

**Nome:** O Usuário Cria uma Conta

**Identificador:** CU002

**Importância:** Imprescindível

**Sumário:** Sessão onde o usuário pode criar uma conta para salvar seus scores.

**Atores:** Cliente

**Pré condição:** O cliente ter ao menos um e-mail válido

**Fluxo Principal:**

1. O cliente abre o jogo no WebSite
2. Cliente acessa o campo de cadastro de conta
3. Cliente preenche campos de dados pessoais (e-mail, senha e apelido)
4. Cliente cria a conta no jogo

**Fluxo de Exceção:**

[CU002.1] Erro de Cadastro de E-mail

1. O sistema retorna um erro quando o cliente tenta registrar um e-mail inválido.

## 5.3 O Usuário Faz Login

**Nome:** O Usuário Faz Login

**Identificador:** CU003

**Importância:** Imprescindível

**Sumário:** Sessão onde o usuário pode acessar sua conta para poder jogar o jogo com a possibilidade de registrar os scores

**Atores:** Cliente

**Pré Condição:** O cliente possuir uma conta cadastrada no banco de dados do jogo

**Fluxo Principal:**

1. Cliente abre o jogo no WebSite
2. Cliente acessa o campo de login
3. Cliente preenche apelido e senha

#### 4. Cliente acessa a conta

##### **Fluxo de Exceção:**

[CU003.1] Erro de Login

1. Usuário digita um apelido ou senha errados e o sistema retorna um erro

#### 5.4 O Usuário Edita Seu Login

**Nome:** O usuário edita seu login

**Identificador:** CU004

**Importância:** Imprescindível

**Sumário:** Sessão onde o usuário pode editar seu login e assim fortalece a segurança da sua conta ou recuperar seu login

**Atores:** Cliente

**Pré Condição:** O cliente possuir uma conta cadastrada no banco de dados do jogo

##### **Fluxo Principal:**

1. Cliente abre o jogo no WebSite
2. Cliente esqueceu apelido ou senha
3. Cliente acessa Esqueci a Senha
4. Cliente insere seu Email para envio de alteração de senha
5. Cliente acessa seu email
6. Cliente acessa link que informa nova senha

##### **Fluxo de Exceção:**

1. Erro na digitação de apelido e senha para confirmação de operações

#### 5.5 O Usuário Vê os Dados da Conta

**Nome:** O Usuário Vê os Dados da Conta

**Identificador:** CU005

**Importância:** Imprescindível

**Sumário:** Sessão onde o usuário pode ver seus dados como e-mail, scores

**Atores:** Cliente

**Precondição:** O cliente possuir um conta cadastrada no banco de dados do jogo

##### **Fluxo Principal:**

1. Cliente abre o jogo no WebSite
2. O cliente acessa a conta e seus dados pessoais(apelido, e-mail, scores, etc.)

##### **Fluxo de Exceção:**

[CU005.1] Erro de Login

1. Usuário digita um apelido ou senha errados e o sistema retorna um erro

## 6. Descrição da interface com o usuário

### 6.1 Telas e descrições

#### 6.1.1 Tela inicial do jogo

**Descrição:** Tela inicial onde o usuário define a dificuldade, número de vidas e o filtro de músicas.

*Figura 2*

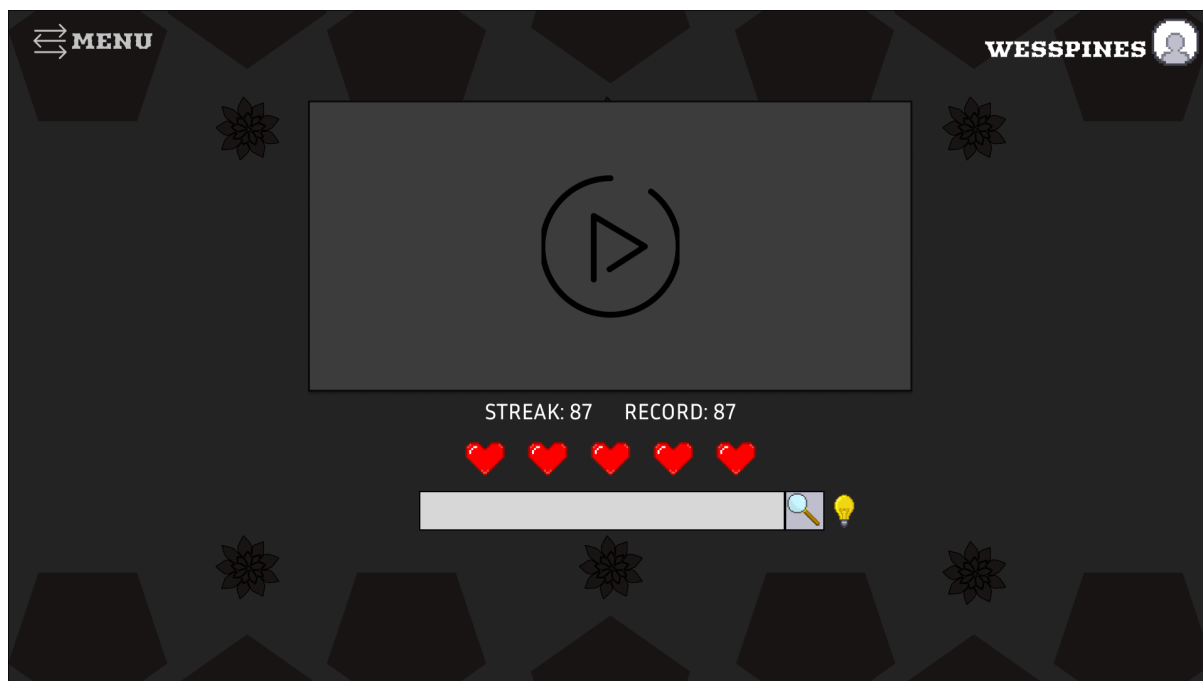


*Fonte: Acervo da Equipe*

### 6.1.2 Tela principal do jogo

**Descrição:** Tela principal do jogo onde o jogador irá jogar

*Figura 3*



*Fonte: Acervo da Equipe*



### 6.1.3 Tela de login

**Descrição:** Tela do jogo onde o jogador poderá fazer login

*Figura 4*



*Fonte: Acervo da Equipe*

#### 6.1.4 Tela de cadastro

**Descrição:** Tela do jogo onde o jogador poderá fazer seu cadastro

*Figura 5*

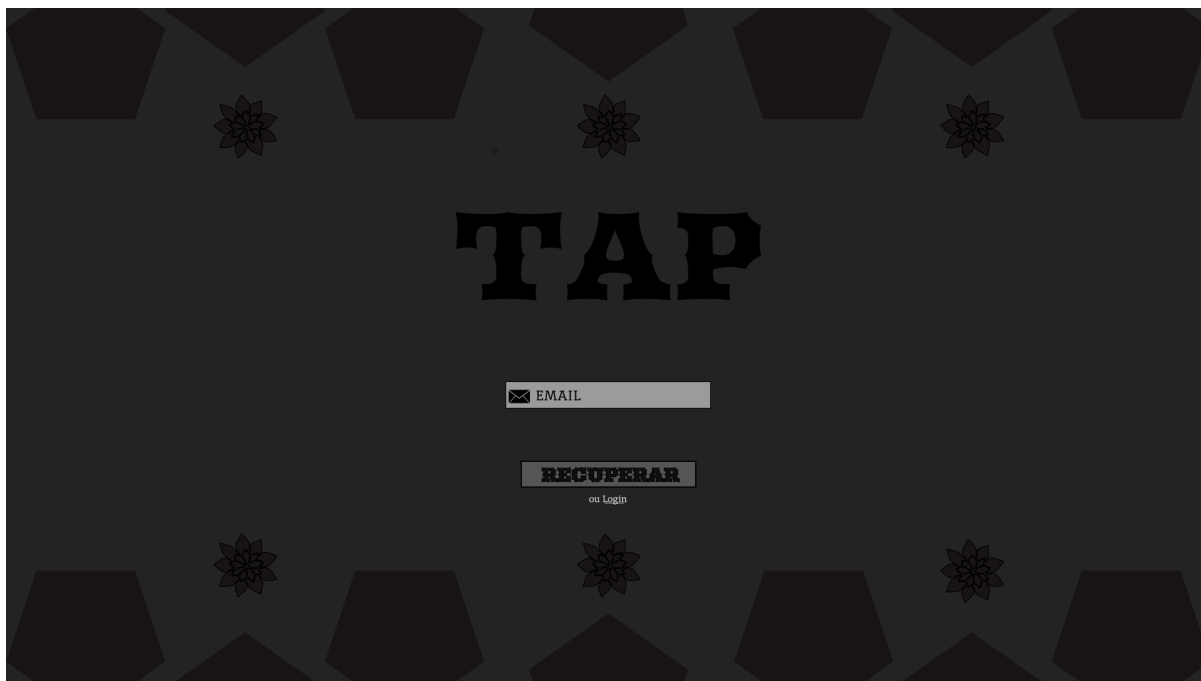


*Fonte: Acervo da Equipe*

### 6.1.5 Tela de recuperação de senha

**Descrição:** Tela do jogo onde o jogador poderá recuperar sua senha

*Figura 6*



*Fonte: Acervo da Equipe*

### 6.1.6 Tela do administrador

**Descrição:** Tela onde o administrador pode adicionar e remover músicas.

*Figura 7*



The image shows a web interface for an application named "TAP". The background is dark grey with a repeating pattern of light grey pentagons and stylized flowers. At the top center, the word "TAP" is written in a large, bold, serif font. Below the title, there are several input fields and buttons. On the left side, there are four stacked input fields labeled "NOME DA MÚSICA", "NOME DO JOGO", "GÊNERO", and "LINK DA MÚSICA". To the right of these, there is a single input field labeled "ID DA MÚSICA". Below the input fields, there are two buttons: "INSERIR" on the left and "REMOVER" on the right. Both buttons are rectangular with a dark background and light text.

*Fonte: Acervo da Equipe*

## 7. Rastreabilidade:

7.1 Requisitos Funcionais/Não funcionais e Requisitos Funcionais/Não funcionais:

	R F 0 0 1	R F 0 0 2	R F 0 0 3	R F 0 0 4	R F 0 0 5	R F 0 0 6	R F 0 0 7	R F 0 0 8	R F 0 0 9	R F 0 0 10	R F 0 0 11	R N F 0 1	R N F 0 2	R N F 0 3	R N F 0 4	R N F 0 5	R N F 0 6	R N F 0 7	R N F 0 8
R F 1		R	R																
R F 2	D		R																
R F 3	D	D		D		D													
R F 4			R																
R F 5													R						
R F 6			R				R												
R F 7						R													
R F 8											R								
R F 9																			
R F 1																			

0																			
R F 1 1								D											
R N F 1																			
R N F 2					R														
R N F 3																			
R N F 4									R										
R N F 5																			
R N F 6																			
R N F 7															R				
R N F 8													R						

## 7.2 Casos de uso e Requisitos:

	R F 0 0 1	R F 0 0 2	R F 0 0 3	R F 0 0 4	R F 0 0 5	R F 0 0 6	R F 0 0 7	R F 0 0 8	R F 0 0 9	R F 0 1 0	R F 0 1 1	R N F 0 1	R N F 0 2	R N F 0 3	R N F 0 4	R N F 0 5	R N F 0 6	R N F 0 7	R N F 0 8
C U 1	R	R								R				R					
C U 2	D																		
C U 3		D																	R
C U 4		D																	R
C U 5			D																R

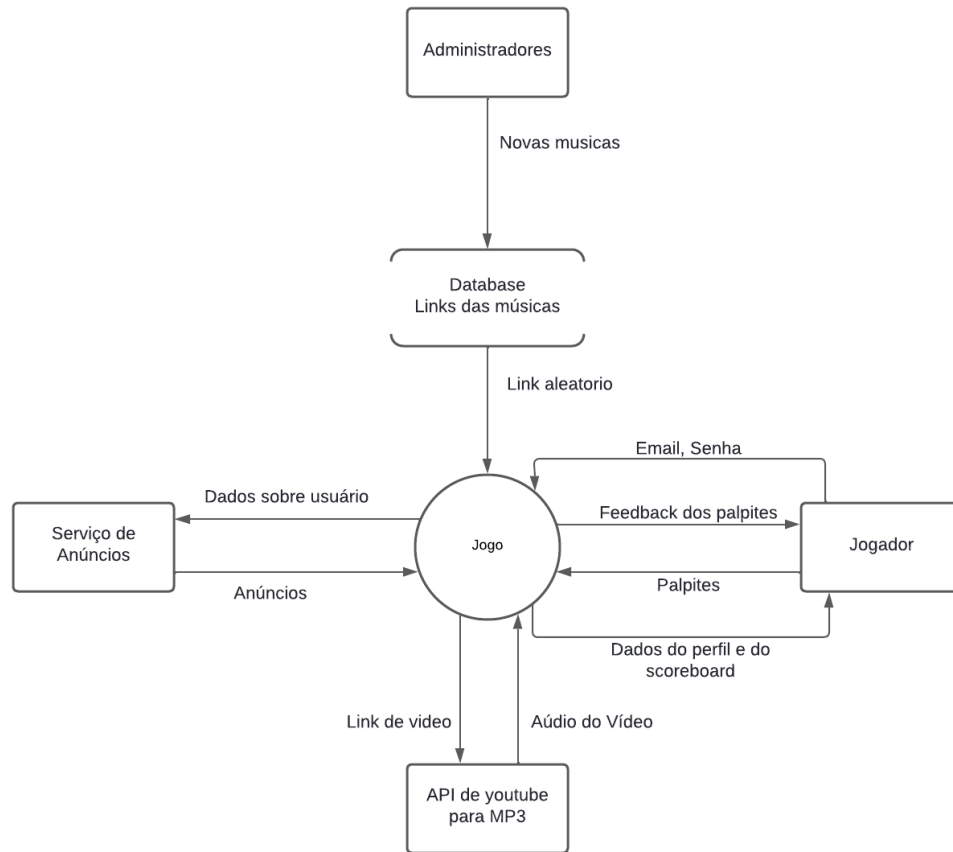
### Legenda:

D - Requisito da linha depende do da coluna

R - Requisito da linha é relacionada ao da coluna

## 8. Diagrama de Contexto:

*Figura 8*

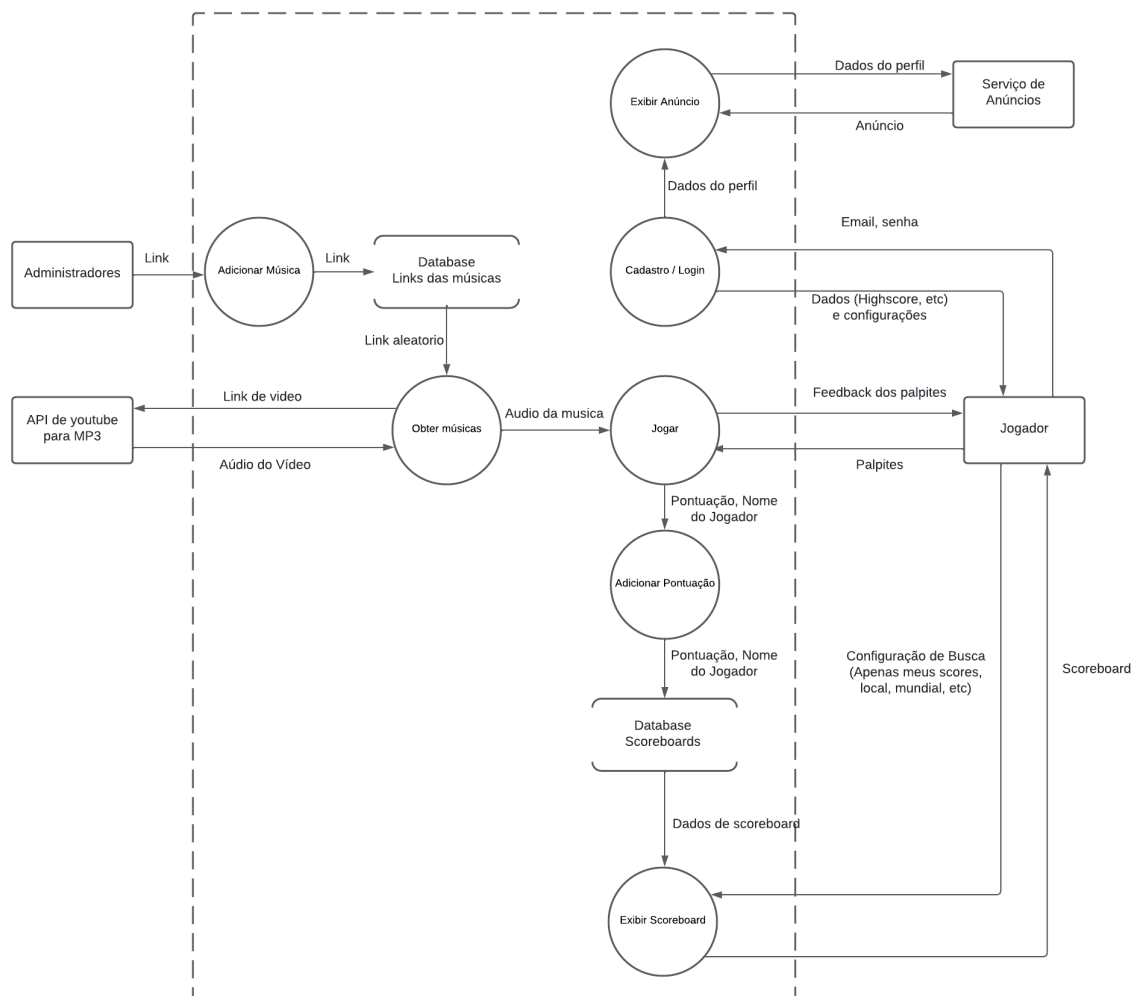


*Fonte: Acervo da equipe*



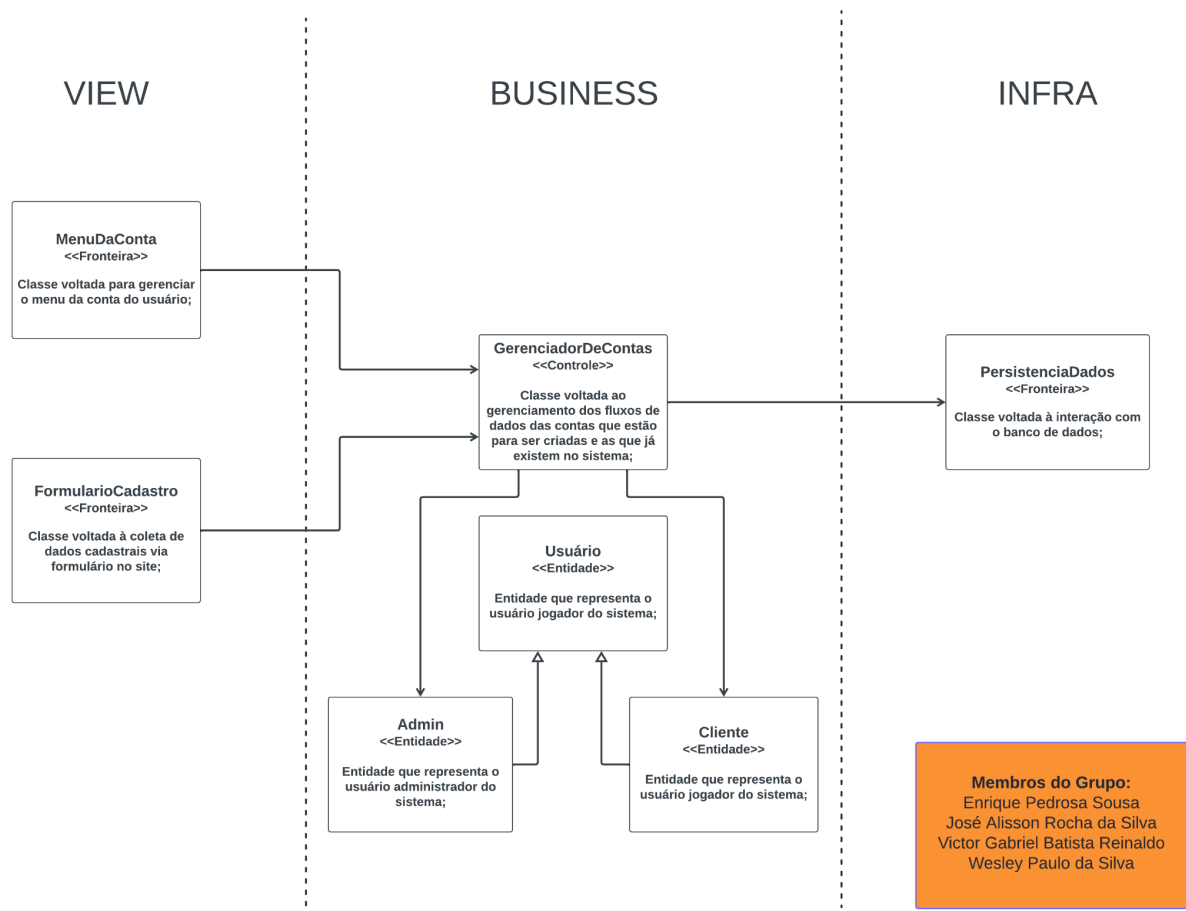
## 9. Diagrama de Fluxo de Dados - Nível 1:

Figura 9



Fonte: Acervo da equipe

## 10. Diagrama de Classe de Análise:



Fonte: Acervo da equipe

## 11. Elicitação de Requisitos

### 11.1. Brainstorm

Técnica escolhida para a elicitação inicial de requisitos, pois ela é eficaz em gerar ideias e levantar discussões rapidamente, especialmente quando todos os membros da equipe podem compartilhar suas visões sobre o que o sistema deve fazer.

- Data e horário: 13 de agosto de 2024, das 14:00 às 16:00.
- Local: Reunião virtual via Discord.
- Participantes: Equipe de desenvolvimento (Enrique Pedrosa, Victor Gabriel, Wesley Paulo, José Alisson).
- Resultados obtidos: Requisitos funcionais [RF001], [RF002], [RF003], [RF005]. Cada membro da equipe trouxe suas sugestões baseadas nas suas expectativas pessoais e experiências anteriores com jogos semelhantes. Foi uma discussão livre e colaborativa.

### 11.2. Workshop de Requisitos

Adotado como técnica de elicitação para refinar e revisar os requisitos identificados no brainstorm e discutir em maior profundidade aspectos mais técnicos do sistema.

- Data e horário: 14 de agosto de 2024, das 22:00 às 00:00.
- Local: Reunião virtual via Discord.
- Participantes: Equipe de desenvolvimento (Enrique Pedrosa, Victor Gabriel, Wesley Paulo, José Alisson).
- Resultados obtidos: Requisitos [RF010], [RNF001], [RNF002], [RNF003], [RNF004], [RNF005], [RNF006], [RNF007], [RNF008]. Durante o workshop, utilizamos ferramentas de modelagem como diagramas e wireframes para revisar as funcionalidades desejadas e alinhar expectativas.

### 11.3. Considerações Finais

As técnicas brainstorm e workshop de requisitos foram complementares e essenciais para garantir uma elicitação eficaz. O brainstorm permitiu levantar um grande número de requisitos em uma fase inicial do projeto, enquanto o workshop de requisitos possibilitou uma discussão mais profunda e focada, resultando na validação e refinamento dos requisitos iniciais.

Exemplo de técnica de análise de concorrentes: Comparando TrackAudio com jogos de adivinhação de populares, como o Gamedle, poderíamos identificar recursos inovadores ou ausentes que poderiam ser incorporados ao nosso projeto.