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 do Projeto

João Pessoa

2024

Histórico de Revisões

Data	Versão	Descrição	Autor(es)
<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>
<16/09/2024>	<0.04>	<Correções e revisões pontuais do documento inteiro>	<Enrique Pedrosa Sousa, Wesley Paulo da Silva, José Alisson da Silva e Victor Gabriel Batista Reinaldo>
<24/09/2024>	<0.05>	<Adição de Score ao glossário. Separação do RF002, gerando RNF013. Renomeação do RNF009. Adição do RF010. Adição e correção de níveis de prioridades de requisitos. Remoção do CU005. Atualização do backend do sistema>	<Enrique Pedrosa Sousa e Victor Gabriel Batista Reinaldo>
<27/09/2024>	<0.06>	<Adição de Usuário ao glossário. Remoção do RF006, Personalização de Vidas. Os RFs que estavam à frente desceram um número. Adição dos RF010 (novo), RF011 e RF012. Adição dos CU005, CU006 e CU007. Edição dos CU001, CU002, CU003 e CU004. Atualização do diagrama de casos de uso>	<Enrique Pedrosa Sousa, José Alisson da Silva e Victor Gabriel Batista Reinaldo>
<27/10/2024>	<0.07>	<Edição do Diagrama de Classe de Análise para conter as novas classes adicionadas pelo projeto. Criação do tópico 8.1 Padrões de Projetos Utilizados que descreve quais padrões foram utilizados, qual a razão e utilidades deles e a lista de classes que utilizam os padrões.>	<Enrique Pedrosa Sousa e Victor Gabriel Reinaldo, José Alisson da Silva e Wesley Paulo>

SUMÁRIO

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 Motivação

A motivação para o desenvolvimento do TrackAudio Project nasceu quando a equipe identificou uma oportunidade ao perceber que não existiam jogos que atendiam suas expectativas de integrar música e adivinhação de uma forma cativante. O TrackAudio Project é inspirado por jogos já existentes, porém traz uma proposta única: desafiar os jogadores a identificarem músicas de diferentes jogos com variações como velocidades alteradas ou inversões, promovendo uma experiência mais dinâmica e personalizada. O jogo visa preencher essa lacuna no mercado, proporcionando aos jogadores a chance de testar seus conhecimentos musicais de jogos de forma divertida e desafiadora.

2.2 Problemas Identificados

Os principais problemas identificados que motivaram a criação do TrackAudio Project incluem:

- A falta de jogos que permitam aos jogadores adivinhar músicas de jogos em uma variedade de contextos, como músicas lentas, rápidas ou invertidas.
- A ausência de filtros adequados para selecionar o tipo de música que o jogador quer adivinhar, seja por gênero de jogo, ano de lançamento ou nível de dificuldade.

2.3 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.

2.4 Usuários do Software

2.4.1 Administradores

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

2.4.2 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.

2.5 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. Glossário

- **Cliente**
    - Jogador que participa do jogo, interagindo com a plataforma para adivinhar músicas de videogames.
    - Atributos:
        - E-mail: String, até 100 caracteres. Utilizado para login e comunicação com o sistema.
        - Senha: String, até 100 caracteres. Utilizada para autenticação.
        - Apelido: String, até 100 caracteres. Nome público do jogador.
        - Descrição: String, de até 300 caracteres. Atributo opcional. Feito para descrever brevemente o usuário ou o que ele quiser escrever nessa seção.

    - Quando um cliente for criado, ele não terá o atributo “Identificador de Administrador”.

- **Administradores (Admins)**
    - Responsável por gerenciar o conteúdo do jogo, incluindo a adição e remoção de músicas do banco de dados.
    - Atributos:
        - E-mail: String, até 100 caracteres. Utilizado para login administrativo.
        - Senha: String, até 100 caracteres. Autenticação para acesso administrativo.
        - Nome: String, até 100 caracteres. Nome do administrador para identificação.
        - Identificador de Adminstrador: Valor Booleano registrado como “True” para identificar que a conta é de um administrador.

- **Usuário**
    - Categoria mais genérica para pessoas que entram no sistema, jogam o jogo e afins são classificados como usuários. Essa categoria inclui clientes e administradores do jogo.
    - Atributos:
        - E-mail: String, até 100 caracteres. Utilizado para login e comunicação com o sistema.
        - Senha: String, até 100 caracteres. Utilizada para autenticação.

        - Apelido: String, até 100 caracteres. Nome público do jogador.
        - Identificador de Administrador: Valor Booleano registrado como “True” ou “False” para identificar se a conta é ou não é de um administrador.
        - Descrição: String, de até 300 caracteres. Atributo opcional. Feito para descrever brevemente o usuário ou o que ele quiser escrever nessa seção.

- **Jogo**
    - Jogo(Track Audio Project) que desafia os usuários a adivinhar músicas de videogames, com diferentes níveis de dificuldade e opções de personalização.
    - Atributos:
        - Níveis de Dificuldade: Fácil, Médio, Difícil, Extremo. Define a complexidade das músicas e as modificações aplicadas (Por exemplo: música lenta, invertida).
        - Multiplicador de Dificuldade: Fator que afeta a pontuação com base no nível escolhido:
            - Fácil: 1
            - Médio: 1.5
            - Difícil: 3
            - Extremo: 5

- **Música**
    - Trilha sonora de jogos que os jogadores devem adivinhar. As músicas podem ser modificadas para aumentar a dificuldade.
    - Atributos:
        - Nome da Música: String, até 200 caracteres.
        - Nome do Jogo: String, até 200 caracteres. O jogo ao qual a música pertence.
        - Ano de Lançamento: Inteiro, 4 dígitos. Ano de lançamento do jogo.
        - Gênero: String, até 50 caracteres. Gênero do jogo.

- **Efeitos**
    - Modificações que afetam a música para aumentar a dificuldade do jogo.
    - Atributos:
        - Tipo de efeito: Lenta, Invertida ou Rápida.

- **Filtros**
    - Opções de personalização que permitem ao jogador escolher os tipos de música que deseja adivinhar.
    - Atributos:
        - Ano: Inteiro, 4 dígitos. Filtro por ano de lançamento do jogo.
        - Gênero do Jogo: String, até 50 caracteres.
        - Dificuldade: Fácil, Médio, Difícil, Extremo. Permite selecionar o nível de dificuldade do jogo(Baseado em sua popularidade).

- **Pontuação (Score)**
    - É o registro que o cliente obtém após encerrar uma partida no TAP. Basicamente um placar final como em qualquer jogo que registra pontuação.
    - Atributos:
        - Valor: Número inteiro de 64 bits.
        - Apelido: Nome público do jogador.

4. Elicitação de Requisitos

4.1. Brainstorm

Não tivemos apenas um dia de brainstorm como se imagina propriamente dito. Essa ideia veio de muito tempo atrás, mais especificamente desde o começo do ano de 2024. Todos os membros da equipe foram conversando seja no horário do almoço, ou em rodas de estudos, ou em chamadas no Discord sobre essa ideia que foi amadurecendo dia após dia. Mas, após nossa decisão de usar a ideia para o projeto, fizemos uma reunião que seria o brainstorm “final” para elicitar os requisitos principais e firmar o cerne do projeto. Essa data foi:

- 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.

4.2. 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.

4.3. Análise de Concorrentes

Foi feita uma análise dos principais concorrentes jogos de adivinhação, reunindo as principais características de cada jogo e comparando seus diferentes elementos com o TrackAudio Project para abordar features que não estão presentes em todos os jogos comparados.

- Squirdle: Um jogo de adivinhação com foco nos fãs de pokémon. O objetivo é acertar o Pokémon do dia em até sete palpites. Para cada tentativa, o jogo compara a geração, tipo, altura e peso do Pokémon com o palpite do jogador.
- Nerdle: Voltado para quem prefere números e operações matemáticas, desafia os jogadores a adivinhar uma equação correta. Ele utiliza números e operadores matemáticos básicos (soma, subtração, multiplicação e divisão).
- Musicle: Concorrente musical onde os jogadores escolhem um gênero musical e tentam adivinhar a capa do álbum com base em uma música tocada.
- Gamedle: Um jogo de adivinhação focado em videogames, onde os jogadores têm que acertar o jogo do dia com base em uma imagem de capa pixelada.

Tabela comparativa de features dos principais concorrentes com o Track Audio Project:

Feature	TrackAudio	Squirdle	Nerdle	Musicle	Gamedle
Conteúdo base	Músicas de jogos	Pokémon	Matemática	Músicas	Jogos
Palpites máximos por jogo	Sem limite fixo	7	5	3	6
Dicas visuais	Nome do jogo	Setas e cores	Cores	Capas de álbuns	Pixelização
Modos de jogo	N/A	N/A	N/A	Vários estilos musicais	4 modos diários
Níveis de dificuldade	Fácil/Médio/ Difícil/ Extremo	N/A	N/A	N/A	Enigma e fácil
Filtros de conteúdo	Nome e ano dos jogos	N/A	N/A	Gêneros musicais	N/A
Personalização de jogo	Vidas, modificação da música, filtros	N/A	N/A	Escolha de gênero	N/A
Score	Sim	N/A	N/A	N/A	Sim
Experiência ilimitada	Sim	Não	Não	Não	Sim (para alguns modos)

4.4. Considerações Finais

Utilizamos várias técnicas ao longo do processo de elicitação de requisitos do TrackAudio Project, entre elas foi feito um brainstorm nos permitiu levantar os primeiros requisitos funcionais e não funcionais com base nas expectativas pessoais de cada membro da equipe. Durante essa etapa identificamos requisitos fundamentais, como o sistema de criação de contas, o login, o registro de scores e a filtragem de músicas, que formaram a base do projeto.

Posteriormente, adotamos um workshop de requisitos que nos ajudou a alinhar as expectativas da equipe e a priorizar os requisitos de forma mais estruturada, abordando aspectos como desempenho e responsividade do sistema.

A análise de concorrentes foi outra técnica essencial, que nos permitiu entender as forças e fraquezas de jogos similares já existentes no mercado, como Squirdle, Nerdle, Musicle e GAMEDLE. Essa análise comparativa nos forneceu insights valiosos sobre funcionalidades que poderiam ser incorporadas ao TrackAudio Project.

5. Análise dos 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.

5.1 Requisitos Funcionais

5.1.1 [RF001] Criação de Contas Novas

Descrição: O jogador deve ter a opção de criar uma conta que guardará os dados do *cliente* 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: Importante

5.1.2 [RF002] Login de Contas

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

Prioridade: Importante

5.1.3 [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).

Prioridade: Importante

5.1.4 [RF004] Escolha de Dificuldade

Descrição: O usuário terá a opção no jogo de escolher a dificuldade dentro sistema de dificuldades, que varia entre fácil, médio, difícil e extremo. A dificuldade afeta o tipo de música e a forma como ela será alterada.

Prioridade: Importante

5.1.5 [RF005] Filtro de Músicas

Descrição: O sistema deve permitir que o jogador filtre as músicas com base no gênero do jogo e ano de lançamento.

Prioridade: Importante

5.1.6 [RF006] Uso de Dicas

Descrição: O jogo deve ter sistemas de dica única. A dica pode ser usada a qualquer momento e fornece informações extras acerca da música alvo para adivinhação.

Prioridade: Importante

5.1.7 [RF007] 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. As sugestões devem corresponder às músicas cadastradas no jogo.

Prioridade: Imprescindível

5.1.8 [RF008] 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: Importante

5.1.9 [RF009] 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.

Prioridade: Imprescindível

5.1.10 [RF010] Edição dos Dados da Conta

Descrição: O usuário pode alterar os atributos do *Cliente*.

Prioridade: Importante.

5.1.11 [RF011] Visualização do Scores

Descrição: O usuário pode visualizar seu histórico de scores para todas as partidas jogadas ordenadas do maior para o menor.

Prioridade: Seria bom ter

5.1.12 [RF012] Adição de Músicas

Descrição: Os administradores do sistema terão a permissão de adicionar ou remover músicas dentro do banco de dados do jogo.

Prioridade: Imprescindível

5.2 Requisitos Não-Funcionais

5.2.1 [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

5.2.2 [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

5.2.3 [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

5.2.4 [RNF004] Responsividade do Sistema

Conteúdo: O conteúdo visual do jogo (a página) deve ser acessível e funcional em dispositivos desktop, tablets e smartphones, adaptando, reorganizando menus e botões para garantir uma navegação fluida em telas maiores que ou iguais à 1280x720, e que tenham proporção 16:9, 18:9, 19:9, 19.5:9, 20:9, 21:9 ou 22:9.

Prioridade: Importante

5.2.5 [RNF005] Acesso à API do YouTube

Descrição: O sistema deve usar a API chamada YouTube Data API (versão 3 ou superior contanto que a versão funcione de maneira compatível no sistema) para o acesso das músicas que serão utilizadas no jogo.

Prioridade: Imprescindível

5.2.6 [RNF006] BackEnd do Sistema

Descrição: O sistema deverá contar com o BackEnd desenvolvido em Spring (versão 3.4.4 ou superior contanto que a versão funcione de maneira compatível no sistema).

Prioridade: Imprescindível

5.2.7 [RNF007] FrontEnd do Sistema

Descrição: O sistema deverá contar com o FrontEnd em HTML/CSS/JavaScript usando o framework Angular (versão 18 ou superior contanto que a versão funcione de maneira compatível no sistema).

Prioridade: Imprescindível

5.2.8 [RNF008] Banco de Dados do Sistema

Descrição: O jogo deverá conter um banco de dados PostgreSQL (versão 16.4 ou superior contanto que a versão funcione de maneira compatível no sistema) onde estará localizado o link de cada música separadamente e os dados dos usuários.

Prioridade: Imprescindível

5.2.9 [RNF009] Sistema de Anúncios

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: Importante

5.2.10 [RNF010] Sistema 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 d a 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:

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

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:

$$1000 * multiplicador\_de\_dificuldade$$

Prioridade: Importante

5.2.11 [RNF011] Sistema de Dificuldade

Descrição: O jogo deve ter sistema de dificuldade (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

5.2.11 [RNF011] Sistema de Filtragem 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

5.2.12 [RNF012] 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

5.2.13 [RNF013] Verificação de Login

O sistema deverá verificar se as credenciais de uma tentativa de login correspondem a uma conta válida registrada no banco de dados.

Prioridade: Imprescindível

6. Descrição dos Casos de Uso

6.1 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 e Administradores

Pré condições: Não possui

Fluxo Principal:

1. O usuário abre o jogo no Web Site
2. {Tela de login}
3. {Nova partida}
4. O usuário entra na tela de personalização do jogo. Onde ele escolhe as *Opções de Customização*
5. O usuário inicia a partida do jogo
6. {Nova música}
7. Uma nova música é selecionada
8. {Ouvir a música}
9. Caso o usuário queira uma dica, ir para o {Fluxo Alternativo C}
10. O usuário digita um nome de música. Conforme ele faz isso, o sistema mostra uma lista de opção das músicas que se encaixam no que ele está digitando.
11. O usuário seleciona uma das opções da lista e confirma o palpite. Se ele errar, ir para o {Fluxo Alternativo A}.
12. A pontuação da música acertada é adicionada ao score.

13. Ir para {Nova música}

Fluxo Alternativo A:

1. Uma nova parte da música é revelada.
2. Uma vida é subtraída do total de vidas.
3. Se o usuário tiver zero vidas, ir para o {Fluxo Alternativo B}
4. Ir para {Ouvir a música}

Fluxo Alternativo B:

1. Guardar o score da partida, associando-o à conta do usuário.
2. Caso o usuário queira jogar novamente, ir para {Nova partida}

Fluxo Alternativo C:

1. Uma nova parte da música, ou uma informação sobre ela é revelada.
2. O contador de dicas disponíveis é decrementado.
3. Ir para {Ouvir a música}

6.2 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 guardar os *dados do perfil do usuário*.

Atores: Cliente e Administrador

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.

Extends:

- <<extends>> CU003 em {Tela de Login}, se o usuário não tiver conta.

6.3 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 e Administrador

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. {Tela de Login}
4. Cliente preenche apelido e senha
5. 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

Extends:

<<extends>> CU001 em {Tela de Login}, se o usuário quiser fazer login

6.4 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 e Administrador

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

Extends:

- <<extends>> CU003 em {Tela de Login} se o usuário não lembrar do seu nome de usuário ou senha

6.5 O Administrador Adiciona Músicas ao Banco de Dados

Nome: O Administrador Adiciona Músicas ao Banco de Dados
Identificador: CU005

Importância: Imprescindível

Sumário: Semelhante ao CU003, porém agora quem acessou sua conta foi um administrador.

Atores: Administradores

Pré condições: O usuário deve ter uma conta cadastrada no sistema, e, obviamente, sua conta deve ter poderes administrativos.

Fluxo Principal:

1. <<includes>> CU003
2. O administrador acessa a tela “Tela do Administrador”
3. O administrador adiciona uma música ao Banco de Dados

6.6 O Administrador Remove Músicas do Banco de Dados

Nome: O Administrador Remove Músicas do Banco de Dados
Identificador: CU006

Importância: Imprescindível

Sumário: Semelhante ao CU003, porém agora quem acessou sua conta foi um administrador.

Atores: Administradores

Pré condições: O usuário deve ter uma conta cadastrada no sistema, e, obviamente, sua conta deve ter poderes administrativos.

Fluxo Principal:

1. <<includes>> CU003
2. O administrador acessa a tela “Tela do Administrador”
3. O administrador remove uma música do Banco de Dados

6.7 O Usuário Visualiza os Scores

Nome: O Usuário Visualiza os Scores

Identificador: CU007

Importância: Imprescindível

Sumário: O usuário acessa a sua lista de scores.

Atores: Cliente e Administrador

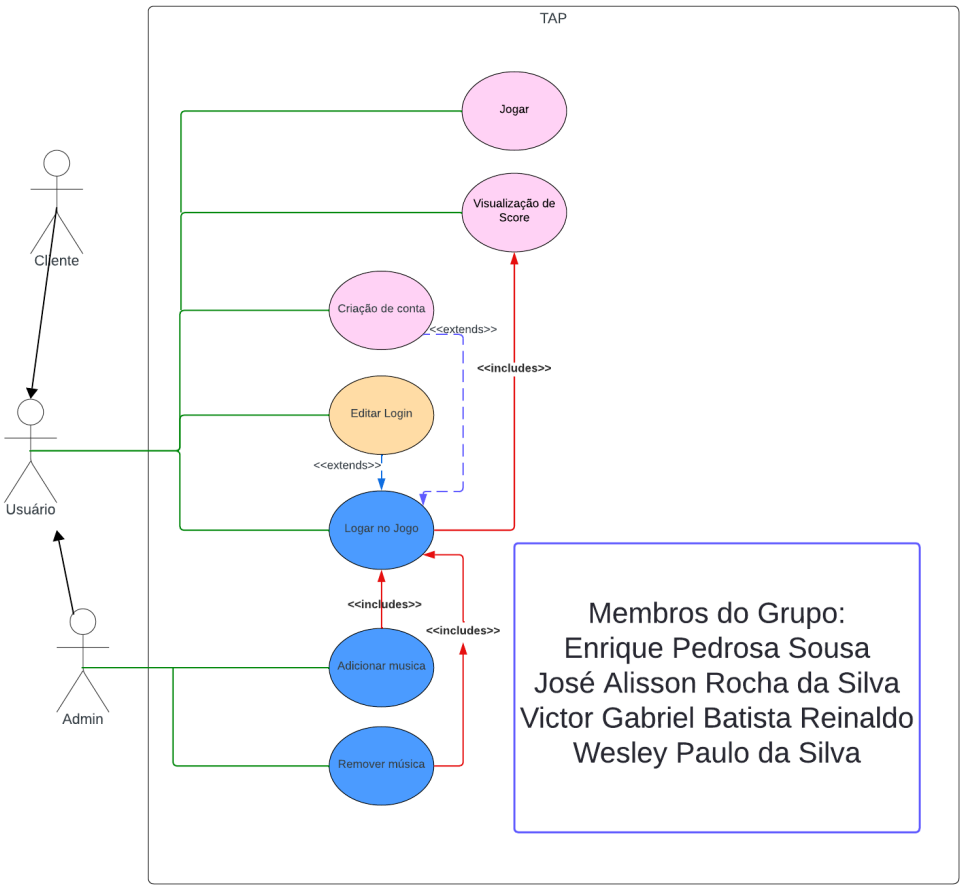
Pré Condição: O usuário possuir uma conta cadastrada no banco de dados do jogo

Fluxo Principal:

1. O usuário abre o jogo no WebSite
2. <<includes>> CU003
3. O usuário acessa a área de scores
4. O sistema exibe uma lista dos scores do usuário, ordenados do maior para o menor

7. Diagrama de Caso de Uso

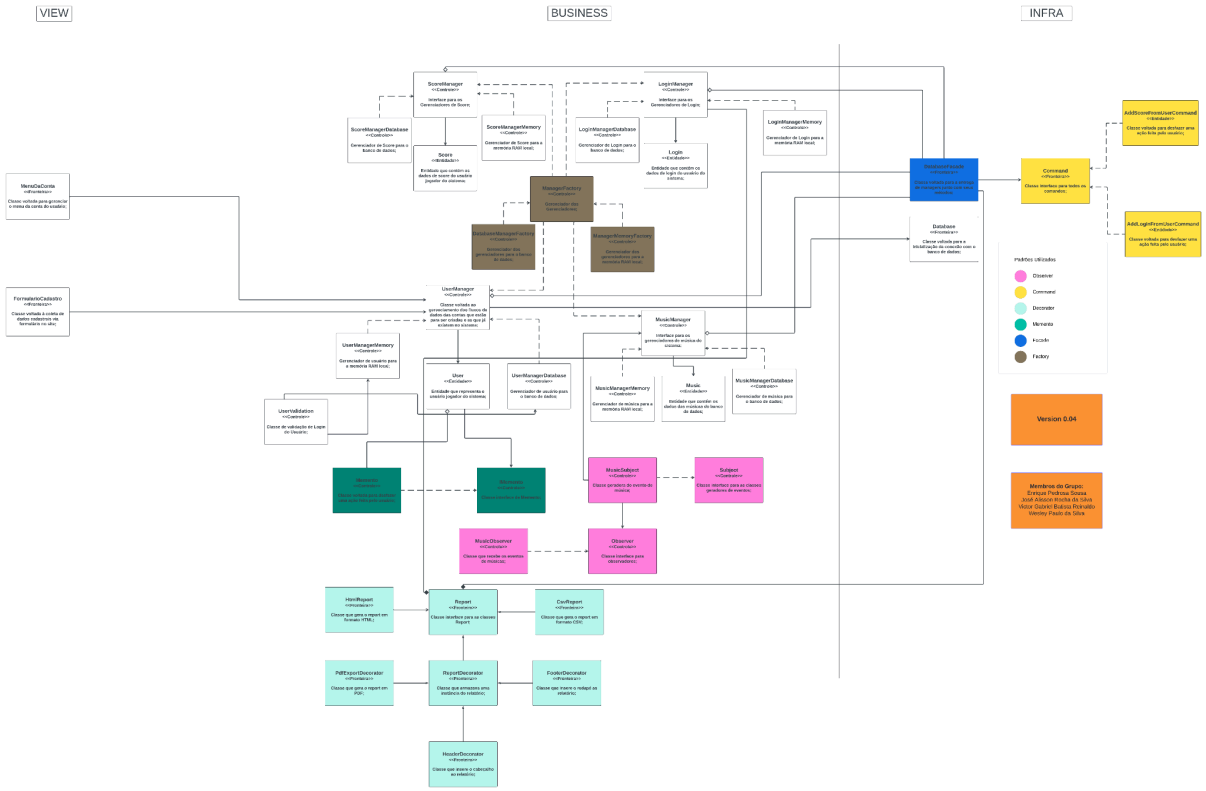
Figura 1



(Fonte: Acervo da equipe)

8. Diagrama de Classe de Análise:

Figura 2



Fonte: Acervo da equipe

Para uma melhor compreensão da imagem, siga o link e veja em melhor resolução: https://lucid.app/lucidchart/0dd23d48-1f98-4a74-be85-3be4f9a609f1/edit?viewport_loc=-5075%2C-3375%2C13363%2C6488%2C0_0&invitationId=inv_23a940de-6e7b-463f-a049-128e414308b3

8.1 Padrões de Projeto Utilizados

- **Facade e Singleton (Azul)**
  Padrão de projeto criado para prover uma interface única para acesso à persistência de dados e mascarar algumas interações complexas através de métodos únicos. Esse padrão é singleton, pois garante que só existirá um único objeto da facade.
    - **Lista De Classes:**
    - DatabaseFacade

- **Command (Amarelo)**
  Padrão de projeto criado com o objetivo de reduzir a quantidade de métodos da classe DatabaseFacade a fim de diminuir a complexidade do código e quebrar a classe DatabaseFacade em mais partes.
    - **Lista De Classes:**
    - Command
    - AddLoginFromUserCommand
    - AddScoreFromUserCommand

- **Decorator (Ciano Claro)**
  Padrão de projeto criado com o objetivo de mexer do cabeçalho ou rodapé de report sem modificar a classe original.
    - **Lista De Classes:**
    - ReportDecorator
    - PdfExportDecorator
    - FooterDecorator
    - HeaderDecorator

- **Template (Verde)**
  Padrão de projeto criado com o objetivo de criar diferentes implementações de um mesmo processo de gerar reports (ou em HTML ou CSV).
    - Report
    - HtmlReport
    - CsvReport

- **Memento (Ciano)**
  Padrão de projeto criado com o objetivo de desfazer uma ação cometida por um usuário na hora de cadastrar ou alterar algum dado da sua conta.
    - **Lista De Classes:**
    - Memento
    - IMemento

- **Observer (Rosa)**
  Padrão de projeto criado com o objetivo de notificar ao sistema quando um administrador adiciona uma música.
    - **Lista de Classes:**
    - Observer
    - MusicObserver

    - Subject
    - MusicSubject

- **Factory (Marrom)**
  Padrão de projeto criado para criar os objetos dos managers para memória RAM ou banco de dados.
    - **Lista de Classes:**
    - Manager
    - MemoryManagerFactory
    - DatabaseManagerFactory

9. Descrição da Interface com o Usuário

9.1 Telas e descrições

9.1.1 Tela inicial do jogo

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

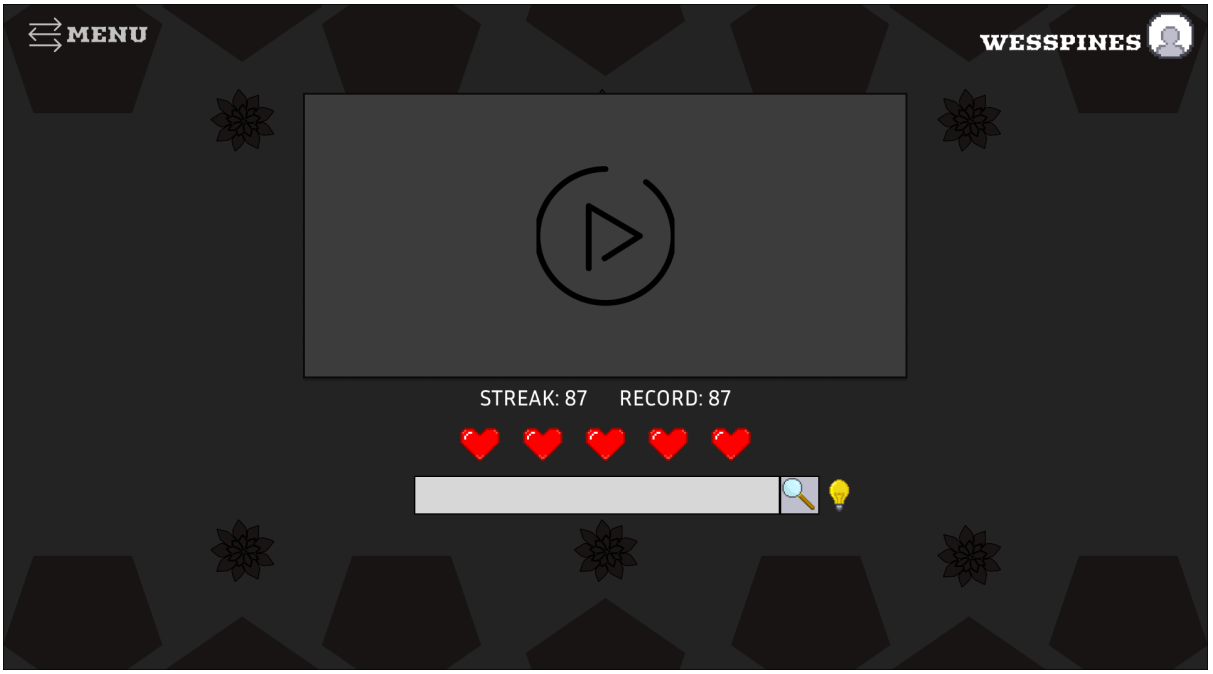
Figura 3

Fonte: Acervo da Equipe

9.1.2 Tela principal do jogo

Descrição: Tela principal do jogo onde o jogador irá jogar

Figura 4



Fonte: Acervo da Equipe

9.1.3 Tela de Login

Descrição: Tela do jogo onde o jogador poderá fazer login

Figura 5

Fonte: Acervo da Equipe

9.1.4 Tela de Cadastro

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

Figura 6

Fonte: Acervo da Equipe

9.1.5 Tela de Recuperação de Senha

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

Figura 7

Fonte: Acervo da Equipe

9.1.6 Tela do Administrador

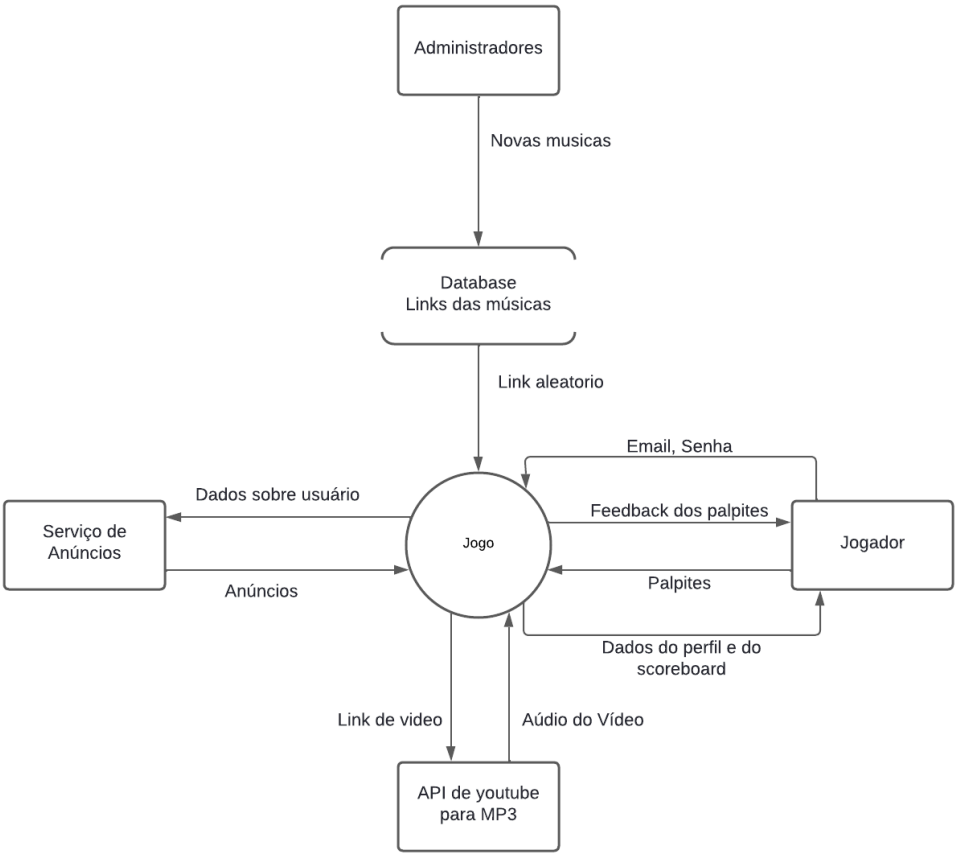
Descrição: Tela onde o administrador pode adicionar e remover músicas.

Figura 8

Fonte: Acervo da Equipe

10. Diagrama de Contexto:

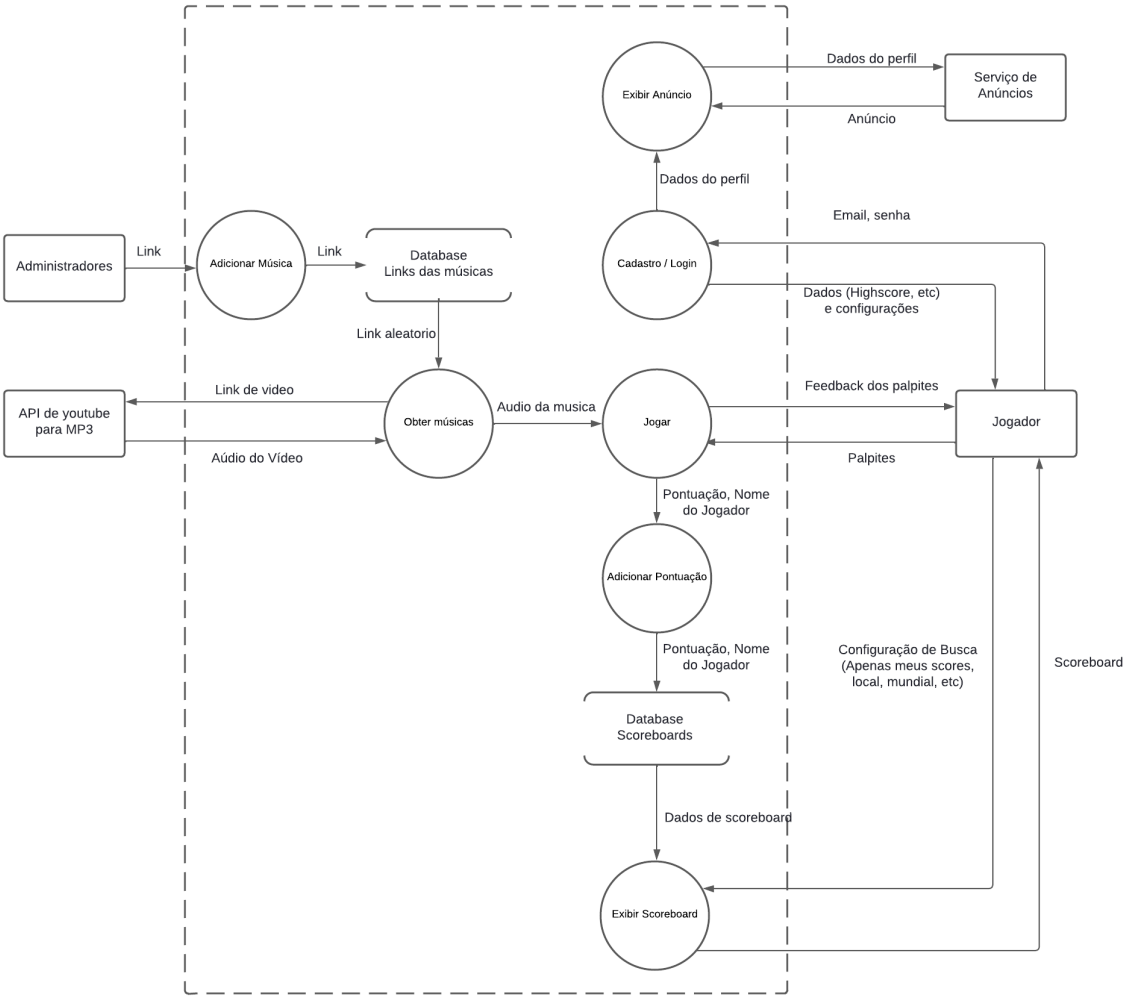
Figura 9



Fonte: Acervo da equipe

11. Diagrama de Fluxo de Dados - Nível 1:

Figura 10



Fonte: Acervo da equipe