INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

COMCLAVE: AS MÚSICAS DE TODOS

CAIO PINHEIRO GUEDES

Cajazeiras
2025

CAIO PINHEIRO GUEDES

COMCLAVE: AS MÚSICAS DE TODOS

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Francisco Paulo de Freitas Neto.

Cajazeiras
2025

IFPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

G924c Guedes, Caio Pinheiro.

Comclave : as músicas de todos / Caio Pinheiro Guedes. – Cajazeiras, 2025.
54f. : il.

Trabalho de Conclusão de Curso (Tecnólogo em Análise de Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2025.

Orientador: Prof. Me. Francisco Paulo de Freitas Neto.

1. Arquitetura de redes. 2. Buscador de internet. 3. Arte. 4. Música. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.

IFPB/CZ CDU: 004.738.52:78(043.2)

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

CAIO PINHEIRO GUEDES

COMCLAVE: AS MÚSICAS DE TODOS

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Me. Francisco Paulo de Freitas Neto

Aprovada em: **10 de setembro de 2025.**

Prof. Me. Francisco Paulo de Freitas Neto - Orientador

Prof. Me. Fábio Abrantes Diniz - Avaliador

IFPB - Campus Cajazeiras

Prof. Me. Janderson Ferreira Dutra - Avaliador

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Francisco Paulo de Freitas Neto**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 10/09/2025 16:41:16.
- **Fabio Abrantes Diniz**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 10/09/2025 16:44:26.
- **Janderson Ferreira Dutra**, **PROFESSOR ENS BASICO TECN TECNOLOGICO**, em 10/09/2025 18:27:19.

Este documento foi emitido pelo SUAP em 10/09/2025. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/autenticar-documento/ e forneça os dados abaixo:

Código 764420
Verificador: 19a18a3e86
Código de Autenticação:

RESUMO

Atualmente, a arte é uma das maneiras mais presentes e abrangentes de propagação da cultura, com grande destaque para a música devido a sua intercomunicabilidade com as demais manifestações existentes, fácil consumo e ritmo acelerado de promoção em diversas plataformas. Todavia, frente ao cenário de homogeneização da arte, as obras produzidas vêm trazendo consigo cada vez mais limitações aos conteúdos culturais abordados e, consequentemente, uma representatividade social de menor significância. Com isso em mente, este projeto visa mitigar o problema a partir da instauração de uma aplicação *web* em formato de site, definido em prol da maior facilidade de acesso e utilização, que abrange a visualização e compreensão de músicas em suas características mais básicas, com o objetivo de agregar um melhor conhecimento sobre a sua estrutura, e habilita o processo criativo conjunto entre diversos usuários para a composição de novas obras de arte que englobam uma multiplicidade de perspectivas sociais e individualidades culturais.

Palavras-chave: site; composição; arte; cultura; música.

RESUMEN

Actualmente, el arte es una de las maneras más presentes e integrales de propagación de la cultura, con gran énfasis en la música debido a su intercomunicabilidad con las otras demostraciones existentes, fácil consumo y ritmo acelerado de promoción en varias plataformas. Entretanto, frente a la escena de homogeneización del arte, las obras producidas traen con ellas cada vez más restricciones a los contenidos culturales abordados y, consecuentemente, una representatividad social con menor significancia. Con esa escena en mente, este proyecto busca mitigar el problema por medio del establecimiento de una aplicación web en el formato de sitio web, elegido a favor de la mayor facilidad de acceso y usabilidad, que cubre la visualización y comprensión de canciones en sus características más simples, con el objetivo de agregar el mejor conocimiento acerca de su estructura, y permite el proceso creativo conjunto por diversos usuarios para la composición de nuevas obras de arte que abarcan una multiplicidad de perspectivas sociales e individualidades culturales.

Palabras clave: sitio web; composición; arte; cultura; música.

ABSTRACT

Currently, art is one of the most present and comprehensive manners of culture propagation, with great emphasis on music due to its intercommunicability with the other existing manifestations, easy consumption and rapid promotion rhythm on various platforms. Nonetheless, in the scenario of homogenization of art, the produced works bring with them progressively more restrictions to the cultural contents addressed within and, consequently, a less significant social representativity. With that in mind, this project aims to mitigate the problem through the establishment of a web application in the format of a website, defined in favor of its easier access and usage, that embraces the viewing and comprehension of songs in their most basic characteristics, with the objective of assembling a better understanding of their structure, and enables the joint creative process amongst many users for the composing of new pieces of art that encompass multiple social perspectives and cultural individualities.

Keywords: website; composition; art; culture; music.

LISTA DE FIGURAS

Figura 1 — Comunicação em uma API REST 17
Figura 2 — Diagrama de busca e atualização de banco em cache 22
Figura 3 — Protótipo da página de cadastro 26
Figura 4 — Protótipo da página de perfil do usuário 27
Figura 5 — Arquitetura do projeto 28
Figura 6 — Modelo do banco principal 30
Figura 7 — Página inicial 33
Figura 8 — Página de artista (visualização) 34
Figura 9 — Página de configurações 35
Figura 10 — Página de acesso 39
Figura 11 — Página de cadastro 40
Figura 12 — Página de busca 41
Figura 13 — Página de perfil do usuário (visualização) 42
Figura 14 — Página de perfil do usuário (edição) 43
Figura 15 — Página de artista (edição) 44
Figura 16 — Página de composição/música [letras] (visualização) 45
Figura 17 — Página de composição/música [partituras] (visualização) 46
Figura 18 — Página de composição/música [letras] (edição) 47
Figura 19 — Página de composição/música [letras] (edição) 48
Figura 20 — Página de álbum (visualização) 49
Figura 21 — Página de álbum (edição) 50
Figura 22 — Página de *playlist* (visualização) 51
Figura 23 — Página de *playlist* (edição) 52
Figura 24 — Página de notificações 53
Figura 25 — Página não encontrada 54

LISTA DE QUADROS

Quadro 1 — Requisitos funcionais do sistema..... 23

LISTA DE ALGORITMOS

Algoritmo 1 — Documento estruturado como JSON..... 20

LISTA DE ABREVIATURAS E SIGLAS

ACID	*Atomicity, Consistency, Isolation, Durability*
API	*Application Programming Interface*
BASE	*Basic Availability, Soft state, Eventually Consistent*
CAP	*Consistency, Availability, Partition tolerance*
CSS	*Cascading Style Sheets*
HTML	*HyperText Markup Language*
HTTP	*HyperText Transfer Protocol*
JS	*JavaScript*
JSON	*JavaScript Object Notation*
NoSQL	Não apenas SQL
REST	*Representational State Transfer*
SQL	*Structured Query Language*

SUMÁRIO

1 INTRODUÇÃO.....13
1.1 PROBLEMÁTICA.....13
1.2 DESCRIÇÃO DA SOLUÇÃO.....13
1.3 OBJETIVOS.....14
1.3.1 Objetivo geral.....14
1.3.2 Objetivos específicos.....14
1.4 ATIVIDADES.....14
1.5 ORGANIZAÇÃO DO DOCUMENTO.....15
2 FUNDAMENTAÇÃO TEÓRICA.....16
2.1 API REST.....16
2.2 BANCO DE DADOS NÃO-RELACIONAL.....17
2.2.1 Orientado a documentos.....19
2.2.2 Modelo chave-valor.....21
3 PROJETO.....23
3.1 REQUISITOS FUNCIONAIS.....23
3.2 PROTÓTIPO DA INTERFACE.....25
3.3 ARQUITETURA.....27
3.3.1 Modelo de dados do banco principal.....29
3.4 TECNOLOGIAS UTILIZADAS.....30
3.4.1 React.....30
3.4.2 Spring Boot.....31
3.4.3 MongoDB.....31
3.4.4 Redis.....31
3.4.5 Firebase Authentication e Firestore.....32
3.5 TELAS DA APLICAÇÃO.....32
4 CONSIDERAÇÕES FINAIS.....36
REFERÊNCIAS.....36

APÊNDICE A – PÁGINAS DA APLICAÇÃO..... 39

1 INTRODUÇÃO

“Arte é um tipo de manifestação humana universal que se comunica de forma simbólica e criativa com a sociedade” (AIDAR, 2024). Cultura é tudo que inclui capacidades ou hábitos adquiridos enquanto membro de uma sociedade (TYLOR, 1871), “é parte da condição humana, do ser, estar e se relacionar com o mundo” (ROSA, 2020).

A música, por sua vez, é uma manifestação artística composta por meio da combinação de ritmo, harmonia e melodia, que refletem a cultura de um povo em uma determinada época ou região (SIGNIFICADOS, 2022). Logo, promovê-la significa ampliar o acesso à arte, difundir culturas e deixar um marco histórico-social.

1.1 PROBLEMÁTICA

Embora atualmente a música seja propagada dinamicamente por meios virtuais variados, seu conteúdo está limitado às percepções de poucos, o que torna algumas mensagens enviesadas e restringe seu impacto cultural. De acordo com dados de 2020, as 3 maiores empresas do ramo juntas influenciaram diretamente cerca de 70% do mercado musical global (SOUNDPLATE, 2021), o que implica na maior parte dos lançamentos estarem atrelados a visões e limitações similares.

Ademais, o acesso a conhecimentos e ferramentas para a composição de músicas não é igualitário a todos, bem como aspectos individuais como afinidade maior a alguma etapa do processo em detrimento de outra, resultando em pessoas com ideias artística e culturalmente significantes, porém, sem meios para transformá-las em uma obra completa.

1.2 DESCRIÇÃO DA SOLUÇÃO

Este documento apresenta uma aplicação desenvolvida para o ambiente *web* que busca expandir o alcance da composição musical e reduzir as limitações existentes para seu processo de criação. Assim, seu alicerce está em habilitar um

ambiente propício à disponibilização de músicas com seus elementos básicos, visando atuar como meio de aprendizado e inspiração, além de proporcionar o trabalho colaborativo entre usuários para a elaboração e apresentação de suas próprias composições. Consequentemente, ao se tornar uma grande coleção de músicas, também atuará como amplificador do repertório musical de seus usuários, incentivando a descoberta de novos projetos, artistas e talentos.

Com isso, é esperado que obras mais miscigenadas sejam criadas, com colaboração direta de várias pessoas que possuem perspectivas únicas e vivências distintas, atuante para a perpetuação e evolução das culturas envolvidas.

1.3 OBJETIVOS

Neste tópico serão apresentados os objetivos geral e específicos do projeto ao qual este documento se refere.

1.3.1 Objetivo geral

Instaurar um ambiente de fácil acesso e utilização que centraliza músicas e seus elementos estruturais, com a finalidade de expandir a propagação da arte a partir do aprendizado pelo contato direto com a informação e prática pela execução dos processos criativo e construtivo em conjunto com os demais integrantes da comunidade.

1.3.2 Objetivos específicos

- Possibilitar a colaboração entre usuários para a composição de músicas originais;
- Habilitar a visualização de músicas de maneira modular, com isolamento por instrumento e letra;
- Incentivar a expansão da biblioteca e repertório musicais.

1.4 ATIVIDADES

Abaixo estão descritas as atividades que foram realizadas para o andamento do projeto, enumeradas com base na ordem cronológica em que foram praticadas.

- **A1 – definição da ideia base**: escolha do domínio da aplicação, qual seria seu diferencial e a maneira como seria abordada;
- **A2 – estabelecimento de requisitos**: delimitação e direcionamento do escopo da aplicação a partir da descrição objetiva do que seria incluso, restrições e comunicação entre elementos do sistema;
- **A3 – prototipação**: formação da identidade visual e desenho da interface do cliente, com posterior validação e melhorias a partir de exemplos;
- **A4 – fundamentação teórica**: busca, estudos e escrita sobre temas correlacionados ao projeto;
- **A5 – decisões técnicas**: escolha da arquitetura e tecnologias para o desenvolvimento da aplicação;
- **A6 – desenvolvimento da aplicação**: codificação do projeto proposto, de acordo com os requisitos e protótipo previamente elaborados;
- **A7 – evolução da implementação**: melhoria das funcionalidades com base em novas análises sobre a ideia inicial e a então atual implementação;
- **A8 – atualização da descrição**: reescrita deste documento para adequar o discorrido ao produto final.

1.5 ORGANIZAÇÃO DO DOCUMENTO

Em seguida, serão apresentados conceitos pertinentes ao desenvolvimento, englobando *Application Programming Interface* (API) *Representational State Transfer* (REST) e bancos de dados não-relacionais. Logo depois, o projeto será descrito e detalhado, desde seu processo inicial de ideação, com o levantamento de requisitos e elaboração do protótipo de interface, até o modelo para a sua implementação, com a arquitetura escolhida, tecnologias utilizadas e visualização de telas. Finalmente, serão discorridos os resultados esperados a partir da aplicação do projeto e a visão de futuro para evolução da solução.

2 FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste documento abordará conceitos importantes para a compreensão completa do projeto que será descrito posteriormente, o que envolve conhecer o que é uma API REST e quais são as principais características de um banco de dados não-relacional, especializando-os em bancos orientados a documentos e bancos que operam no modelo chave-valor.

2.1 API REST

Dizer que um *software* é uma *Application Programming Interface* (API) significa dizer que possui comportamento ou semântica com finalidade de modelar efetivamente a troca de informações, representando uma abstração da implementação e sendo representada por uma especificação que introduz tipos (GOUGH; BRYANT; AUBURN, 2022). Ou seja, quando algum programa se comunica com uma API, ele entende apenas o que precisa ser enviado e o que será devolvido, sem se preocupar com como a informação será tratada.

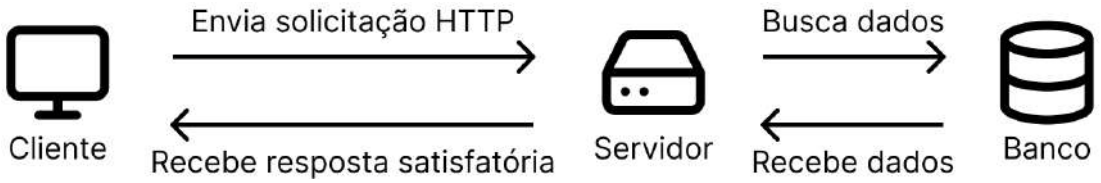
Já *Representational State Transfer* (REST) é um estilo arquitetônico para sistemas de hipermídia distribuídos, derivado de diversos outros estilos baseados em rede e com adições de restrições desejáveis. A partir dessas derivações e restrições, foram definidos os pilares para essa arquitetura, que são cliente-servidor, sem estado, *cache*, interface uniforme, sistema em camadas e, opcionalmente, código por demanda. O intuito do REST é viabilizar uma maior generalização de interfaces e assegurar independência e escalabilidade de componentes, prezando pela redução da latência, garantia da segurança e intercomunicação com sistemas legados (FIELDING, 2000). Atualmente, grande parte da estrutura *web* é baseada nessa arquitetura, que é primordial para o desenvolvimento dos sistemas contemporâneos.

Logo, uma API REST é um tipo de servidor *web* que habilita um programa, chamado de cliente, a acessar recursos que moldam dados e funções de um

sistema (MASSÉ, 2012).

A figura 1 ilustra a comunicação realizada entre cliente, servidor e banco de dados.

Figura 1 – Comunicação em uma API REST



Fonte: Elaborado pelo autor (2025)

2.2 BANCO DE DADOS NÃO-RELACIONAL

Dizer que um banco de dados é NoSQL ou não-relacional refere-se à contraposição existente entre seu modelo e o dos bancos amplamente difundidos antes do seu surgimento. Nesse sentido, ele se consolidou inicialmente como uma alternativa que não se restringia apenas ao uso por meio da *Structured Query Language* (SQL), possibilitava o armazenamento de dados não estruturados e, consequentemente, não dependia de relacionamentos ou lógicas complexas para alcançar o resultado desejado. Contudo, pela forma que surgiu e ganhou espaço por meio das necessidades crescentes da época, as tecnologias NoSQL foram desenvolvidas com inúmeras implementações distintas, com focos específicos para cada caso de uso, o que ampliou o seu escopo e findou dificultando uma definição teórica que as englobasse por completo.

Com o intuito de sumarizar as características principais sem tornar-se excludente à vasta gama de variações dessa tecnologia, Adam Fowler (2015) descreveu bancos de dados NoSQL como *softwares* que não requerem um esquema rigoroso, não utilizam teoria matemática relacional aplicada e são distribuíveis em equipamentos comodato. Dentro desse conceito, o autor destrincha os pontos descritos nos seguintes recursos essenciais:

- Agnóstico a esquemas: é possível simplesmente armazenar diferentes tipos e estruturas de dados dentro do banco, sem a necessidade de especificá-las previamente, o que facilita o trabalho inicial e possíveis alterações;

- Não-relacional: não há presença do conceito de relacionamentos, logo os dados são armazenados de maneira desnormalizada, com agregações para que as informações sejam diretamente acessíveis onde há interesse. Em outras palavras, dados são repetidos em diversos registros, garantindo uma maior eficiência na busca devido à menor complexidade em troca de um maior espaço em disco;

- Altamente distribuível e escalonável com equipamentos comodato: o banco pode estar contido de forma distribuída entre um aglomerado de máquinas com especificações técnicas menos rigorosas, resultando em um crescimento horizontal e com potencial para para uma alta disponibilidade.

Além de possuir uma definição base para o que é um banco NoSQL, também é importante entender os modelos que circundam e direcionam o espaço de banco de dados como um todo. No escopo de bancos relacionais, as transações normalmente seguem as propriedades popularmente conhecidas como ACID: atomicidade, onde uma operação não pode ser completada parcialmente, ou tem êxito em todas as suas instruções ou é um fracasso completo; consistência, que garante que não há nenhuma violação de restrições na transação ou ela será descartada; isolamento, no qual a execução de uma operação não pode interferir em outra e apenas dados finais são utilizados; e durabilidade, que estabelece que uma vez que a transação obteve sucesso, seus resultados serão persistidos no banco de maneira definitiva (REUTER; HAERDER, 1983). Esses conceitos visam determinar o que é necessário para que um banco de dados tenha integridade e confiabilidade.

No escopo de bancos não-relacionais, devido a sua natureza, intuitos de uso e aos compromissos técnicos necessários para seguir os direcionamentos do ACID, é comum que empreguem outro modelo para sua implementação. Como precursor à definição desse modelo, houve a criação do teorema CAP, que representa:

consistência, como todas as cópias de um determinado dado precisam estar iguais a todo momento; disponibilidade (do inglês, *availability*), que garante que toda requisição receberá uma resposta satisfatória, independentemente de quem respondeu ou qual cópia do dado foi enviada; e tolerância a partições (do inglês, *partition tolerance*), onde o sistema permanece operante mesmo que haja alguma falha de comunicação entre suas partições, que devem ter seus dados atualizados quando o problema for resolvido. Esse teorema diz que só é viável a implementação de 2 dentre esses 3 atributos em um banco de dados distribuído, havendo o sacrifício de 1 deles em detrimento do benefício dos demais (BREWER, 2012).

Com ACID e CAP em mente, um novo modelo foi criado, conhecido como BASE: basicamente disponível (do inglês, *basic availability*), que preza pela maior parte dos dados estarem disponíveis na maior parte do tempo; estado flexível (do inglês, *soft state*), que aceita perdas ocasionais de atualizações sobre dados; e eventualmente consistente, que preconiza que todas as cópias dos dados estarão eventualmente na versão mais recente. Uma vez que bancos de dados NoSQL buscam ter alta performance, alta escalabilidade e buscas menos complexas, esse modelo foi adotado por grande parte dos *softwares* não-relacionais (HILLS, 2016).

Por fim, outra característica que se torna atrativa nos bancos de dados não-relacionais é a maneira como os dados são organizados. Para os NoSQL, eles podem ser apresentados em diversos formatos diferentes, o que contribui para a flexibilidade e versatilidade dessas tecnologias. Abaixo serão abordadas as organizações utilizadas para o desenvolvimento deste projeto, que são por meio de documentos e chave-valor.

2.2.1 Orientado a documentos

Documento é a denominação dada a estruturas hierárquicas de dados que podem conter subestruturas, consistem de dados binários ou texto plano e são organizados de maneira não, semi ou bem-estruturada (FOWLER, A., 2015). O modelo do documento em si é um superconjunto de outros modelos, que armazena dados em pares chave-valor e permite que seus valores sejam uma variedade de tipos e estruturas (MONGODB, 2025). Além disso, vale salientar que por se tratar de

um modelo para banco de dados NoSQL, todos os atributos discorridos anteriormente também são válidos.

O algoritmo 1 é um exemplo de documento no formato *JavaScript Object Notation* (JSON) que possui valores variados.

Algoritmo 1 – Documento estruturado como JSON

```
{
    “_id”: “c763685a”,
    “nome”: “Pessoa Pessoa”,
    “dataNascimento”: 1755000000000,
    “contatos”: [“pessoa@email.com”, “(99) 9999-9999”],
    “endereco”: {
            “rua”: “Brigadeiro”,
            “numero”: 29
    }
}
```

Fonte: Elaborado pelo autor (2025)

Embora a descrição de o que é um documento seja muito vasta, o intuito de bancos de dados orientados a documentos, ou seja, que armazenam essas estruturas como seu modelo primário de dados, possuem exatamente esta missão: ser uma solução de cunho mais genérico, flexível e altamente adaptável a inúmeros casos de uso. Em consonância com esse modelo, os bancos também disponibilizam uma série de recursos para interagir com dados de forma otimizada e que reduz a complexidade lógica da aplicação, como funções para manipulação dos documentos e inclusão de índices para agilizar buscas.

Para este projeto, a escolha de um banco NoSQL orientado a documentos ocorre a partir da análise dos dados a serem armazenados: coleções estruturadas com base em valores de tipos variados, sendo simples, múltiplos e até subestruturas. Ao imaginar sua aplicação em um meio diferente, principalmente contrapondo-a a um banco de dados relacional, seria necessário descumprir um dos conceitos que orientam sua devida utilização e realizar a desnormalização dos dados; ou, para seguir consoante ao preconizado, acrescentar lógicas de

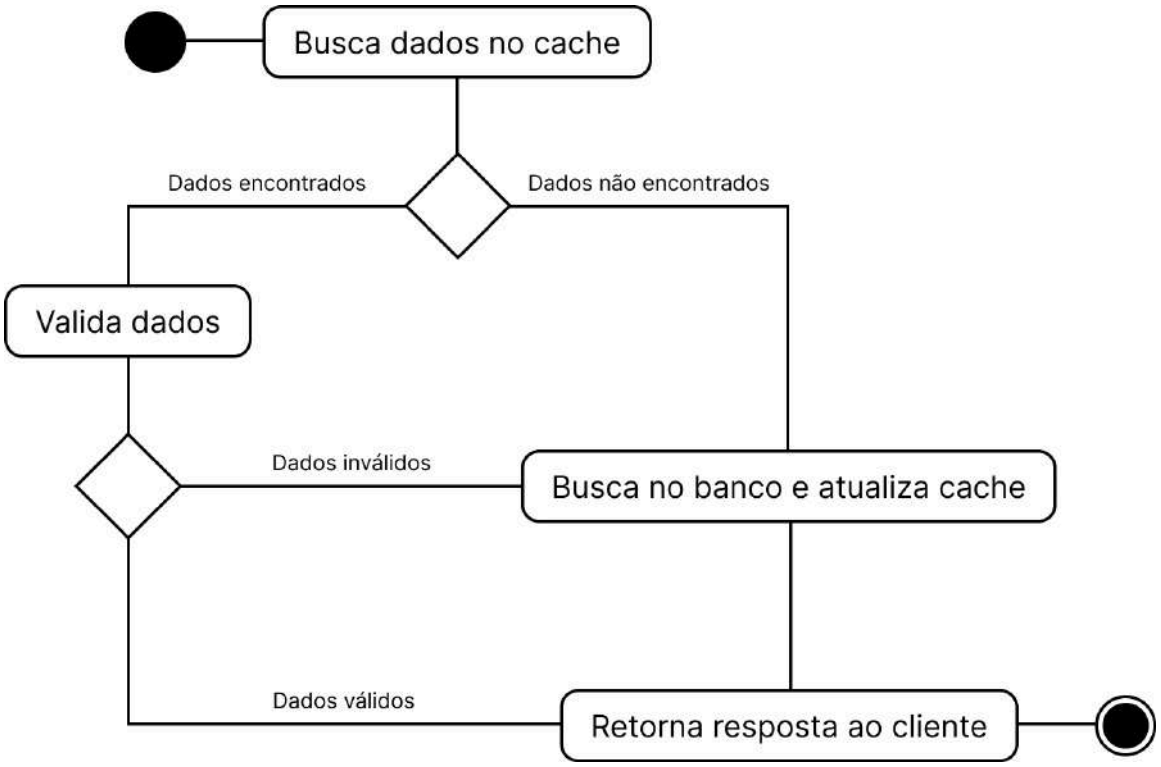
relacionamento a grande parte das entidades, o que agrega maior complexidade para orquestrar as informações de interesse e, consequentemente, resulta em operações mais demandantes e uma experiência de usuário influenciada negativamente pelos tempos de resposta prolongados.

2.2.2 Modelo chave-valor

O modelo chave-valor é extremamente simples quando comparado aos demais modelos de organização de dados. Em suma, cada chave representa um valor que, por sua vez, não é interpretado pelo banco de dados, apenas armazenado. Ao utilizar essa abordagem, o banco consegue indexar performaticamente seus atributos para alcançar uma maior eficiência em operações de gravação, busca e leitura de dados (HILLS, 2016). Em troca do alto desempenho sobre essas funcionalidades, a compreensão e manipulação dos dados é delegada majoritariamente à aplicação, adicionando certo grau de complexidade ao código.

A figura 2 mostra, em notação de diagrama, um exemplo de aplicação que se comunica com um banco qualquer e um banco em cache que utiliza o modelo chave-valor. Note como há necessidade de implementação de uma lógica simples para o bom funcionamento da tecnologia, bem como as três operações mencionadas anteriormente são invocadas.

Figura 2 – Diagrama de busca e atualização de banco em cache



Fonte: Elaborado pelo autor (2025)

Para o caso deste projeto, a manipulação do cache escolhida aplica a estratégia de carregamento preguiçoso, resumida por Martin Fowler (2015) como “um objeto que não possui todos os dados necessários, mas sabe como encontrá-los”. Isso significa que estritamente os dados de interesse são buscados e, quando retornados, ficam disponíveis para utilização por um determinado período de tempo, o que evita a redundância de operações e o uso desnecessário de recursos.

3 PROJETO

Nas seções seguintes, será abordado o processo de determinação e concepção do ComClave, o que inclui a definição dos requisitos funcionais desejados, a prototipação da interface *web*, a arquitetura adotada e as tecnologias utilizadas para desenvolvimento do código fonte.

3.1 REQUISITOS FUNCIONAIS

Para iniciar este projeto, foram levantados requisitos que definem as funcionalidades principais a serem implementadas, com o intuito de delimitar o escopo e direcionar o desenvolvimento do *software*. A definição foi pautada em dois pilares principais: análise das características essenciais para que os objetivos propostos fossem viabilizados e compreensão de aplicações alternativas disponíveis para aproximar a usabilidade ao que os usuários já possuem familiaridade.

A fim de simplificar o entendimento, tais requisitos foram compostos por um nome para identificar a funcionalidade a qual se referem, um sinalizador de quão relevante ele é para a completude da experiência do usuário e seu detalhamento. Esses requisitos podem ser observados no quadro 1, exposto abaixo, que contempla todas as ideias previstas para incorporação nesta versão do sistema.

Quadro 1 – Requisitos funcionais do sistema

Nome: RF-01: Distinção de níveis de acesso	**Relevância:** Alta
A plataforma poderá ser utilizada com diferentes permissões, que são: 1. parcial: usuários não cadastrados poderão apenas visualizar os conteúdos presentes na plataforma; 2. completa: usuários cadastrados poderão criar, visualizar, editar e remover conteúdos.	
Nome: RF-02: Autenticação primária do usuário	**Relevância:** Alta
Criar uma conta e utilizá-la para acessar a plataforma.	
Nome: RF-03: Detalhamento do perfil do usuário	**Relevância:** Alta

Usuários cadastrados possuem uma página de perfil contendo suas informações básicas, construções e contatos sociais.	
Nome: RF-04: Inclusão de *playlists*	**Relevância:** Alta
Criação, visualização, edição e remoção de coleções contendo músicas e composições presentes na plataforma.	
Nome: RF-05: Inclusão de artistas	**Relevância:** Alta
Criação, visualização e edição de perfis para artistas, contendo suas informações básicas, lançamentos e contatos sociais.	
Nome: RF-06: Inclusão de álbuns	**Relevância:** Alta
Criação, visualização e edição de coleções contendo músicas lançadas por um artista ou grupo de artistas participantes específicos, bem como informações básicas sobre a coleção.	
Nome: RF-07: Inclusão de músicas	**Relevância:** Alta
Criação, visualização e edição de músicas lançadas por artistas, contendo informações básicas, letras para diferentes idiomas, partituras para diferentes instrumentos, vídeos para cada idioma e partitura, álbuns pertencentes e créditos.	
Nome: RF-08: Inclusão de composições	**Relevância:** Alta
Criação, visualização e edição de músicas compostas pelos usuários da plataforma, contendo informações básicas, letras para diferentes idiomas, partituras para diferentes instrumentos, vídeos para cada idioma e partitura e créditos; a edição será restrita a usuários creditados na composição.	
Nome: RF-09: Créditos referentes às obras	**Relevância:** Alta
Creditar artistas e usuários em diferentes categorias de contribuição com base no seu envolvimento para a criação da música, composição ou álbum.	
Nome: RF-10: Notificações	**Relevância:** Alta
Visualizar sugestões feitas por usuários às composições, com a possibilidade de aprovar e incluir ou reprovar as alterações.	
Nome: RF-11: Pesquisa por conteúdos incluídos	**Relevância:** Alta

Busca textual por: 1. artistas e usuários a partir de seus nomes; 2. músicas, composições, álbuns e *playlists* a partir de seus títulos e criadores.	
Nome: RF-12: Ordenação de músicas/composições por critério	**Relevância:** Alta
Músicas e composições são ordenadas decrescentemente com base em critérios previamente definidos.	
Nome: RF-13: Personalização da aparência do cliente	**Relevância:** Média
É possível personalizar o tema e a tipografia no cliente utilizado.	
Nome: RF-14: Definição de temas	**Relevância:** Média
Criação, visualização e remoção de coleções de cores, seguindo o padrão da plataforma.	
Nome: RF-15: Seleção do idioma da plataforma	**Relevância:** Média
O idioma da plataforma pode ser definido dentre as opções português, espanhol e inglês.	
Nome: RF-16: Sugestão aleatória de conteúdo	**Relevância:** Baixa
Seleção aleatória de músicas, composições, álbuns e/ou *playlists*.	

Fonte: Elaborado pelo autor (2025)

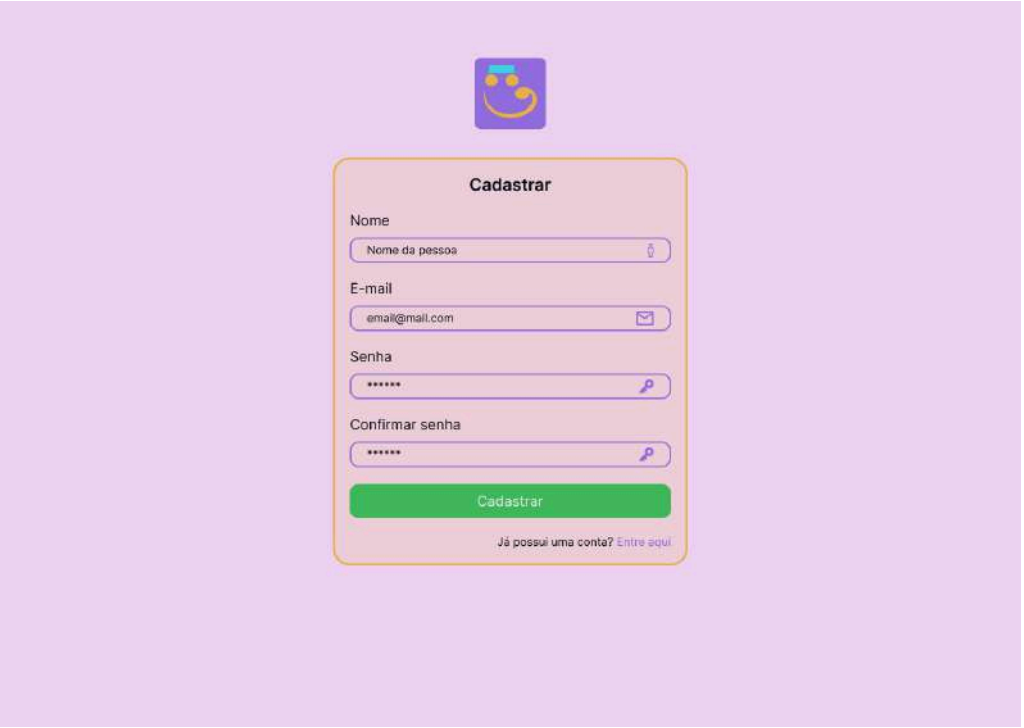
3.2 PROTÓTIPO DA INTERFACE

O intuito de um protótipo é testar estilos de interface, validar os requisitos já levantados, identificar novos problemas e suas respectivas soluções e melhor compreender a usabilidade do produto, uma vez que promove o relacionamento direto entre o sistema e as necessidades do usuário, o que possibilita um melhor planejamento e a antecipação de mudanças de maneira mais fácil e rápida (SOMMERVILLE, 2011). Logo, o protótipo da interface é a tradução dos requisitos funcionais do sistema para um modelo visual pelo qual o usuário terá acesso ao design da plataforma, que visa assegurar um entendimento fiel de como será o *software*. Vale ressaltar que ele não possui nenhuma funcionalidade implementada e

está passível a sofrer alterações a qualquer momento, pois antecede a etapa de codificação para melhor direcioná-la. A seguir, serão apresentadas duas telas do protótipo.

A figura 3 ilustra o visual da página de cadastro, com a funcionalidade de criar uma conta na plataforma utilizando credenciais fornecidas e o redirecionamento para o acesso à plataforma.

Figura 3 – Protótipo da página de cadastro



Fonte: Elaborado pelo autor (2025)

A figura 4 ilustra a página de perfil de usuário, com a visualização de suas informações básicas, redes sociais e construções (composições e *playlists*) dentro do espaço virtual, bem como a opção de edição caso o perfil em questão esteja vinculado à conta autenticada.

Figura 4 – Protótipo da página de perfil do usuário

Fonte: Elaborado pelo autor (2025)

Para interagir com o protótipo completo e compreender a disposição e interação com os requisitos elaborados na etapa anterior, basta acessar o *link*[1] disponibilizado neste documento.

3.3 ARQUITETURA

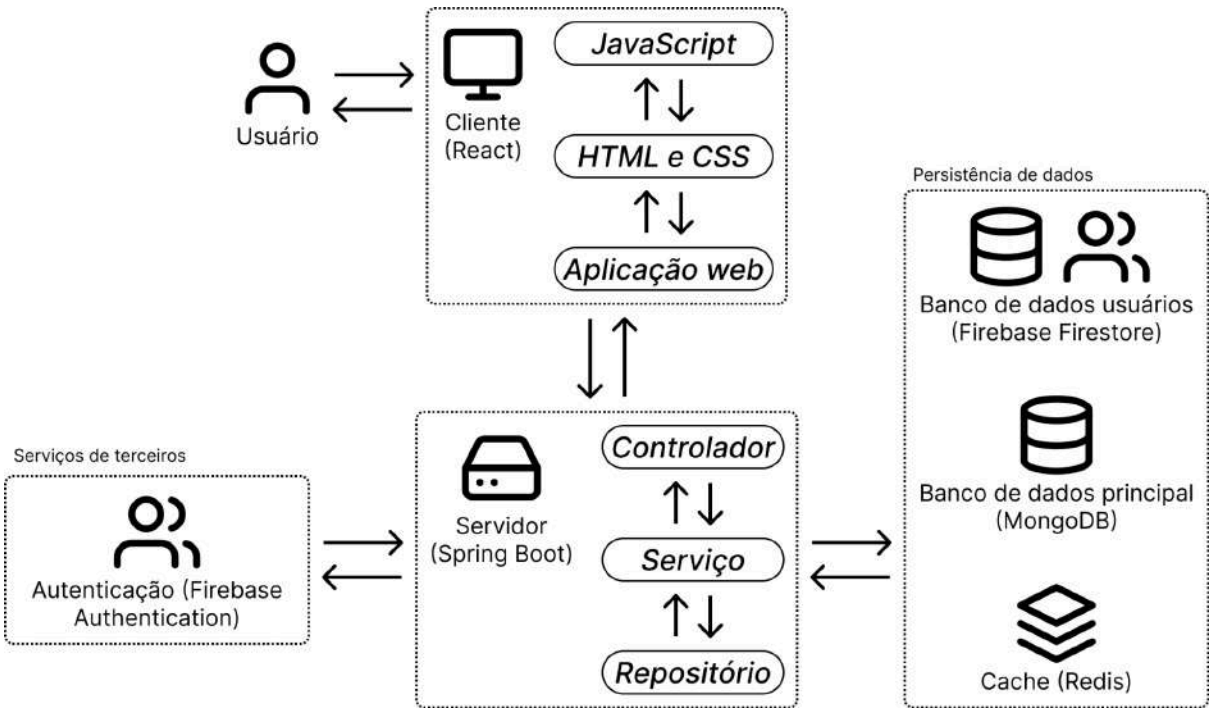
O design da arquitetura de um sistema tem como intuito definir a divisão de camadas, contemplando seus componentes e tecnologias de implementação, a fim de melhor especializar suas funções e explicitar as possíveis interações entre elas.

1 https://www.figma.com/proto/6wqCbyOCnDeAL1SUw5wzcR/TCC?node-id=2-2&t=kUsg9T89FdccQ0KD-1

Assim, temos que o usuário interage única e diretamente com a camada de interação, composta pelo cliente; esta, por sua vez, se relaciona com a camada de API, que contempla o servidor; e, por fim, a API é responsável por se comunicar com a camada de dados, que possui estruturas para armazenamento em banco de dados e em *cache*, e serviços de terceiros, especificamente para autenticação de usuários.

A figura 5 expõe esse processo, com a limitação de camadas representada pelas linhas pontilhadas e, interno a elas, seus componentes e respectivas tecnologias.

Figura 5 – Arquitetura do projeto



Fonte: Elaborado pelo autor (2025)

- **Cliente**: representa o sistema de maneira visual e interativa para o usuário, com a finalidade de ser a entrada para as requisições de recursos; envia estas requisições ao servidor e apresenta o conteúdo recebido de forma intuitiva. Neste projeto, será utilizado o *React* para o desenvolvimento do cliente.

- **Servidor**: recebe a requisição enviada pelo cliente, realiza seu devido processamento e retorna dados satisfatórios ao recurso solicitado; realiza a comunicação com todas as bases de dados utilizadas e manipula seus conteúdos de maneira coerente. Neste projeto, será utilizado o *Spring Boot* para o desenvolvimento do servidor.

- **Banco de dados usuários**: armazena os dados referentes às informações não sensíveis dos usuários cadastrados e disponibiliza-os ao servidor. Neste projeto, será utilizado o *Firebase Firestore*.

- **Banco de dados principal**: armazena os principais dados relacionados à aplicação e disponibiliza-os ao servidor. Neste projeto, será utilizado o *MongoDB*.

- ***Cache***: armazena determinados dados temporariamente e disponibiliza-os ao servidor. Neste projeto, será utilizado o *Redis*.

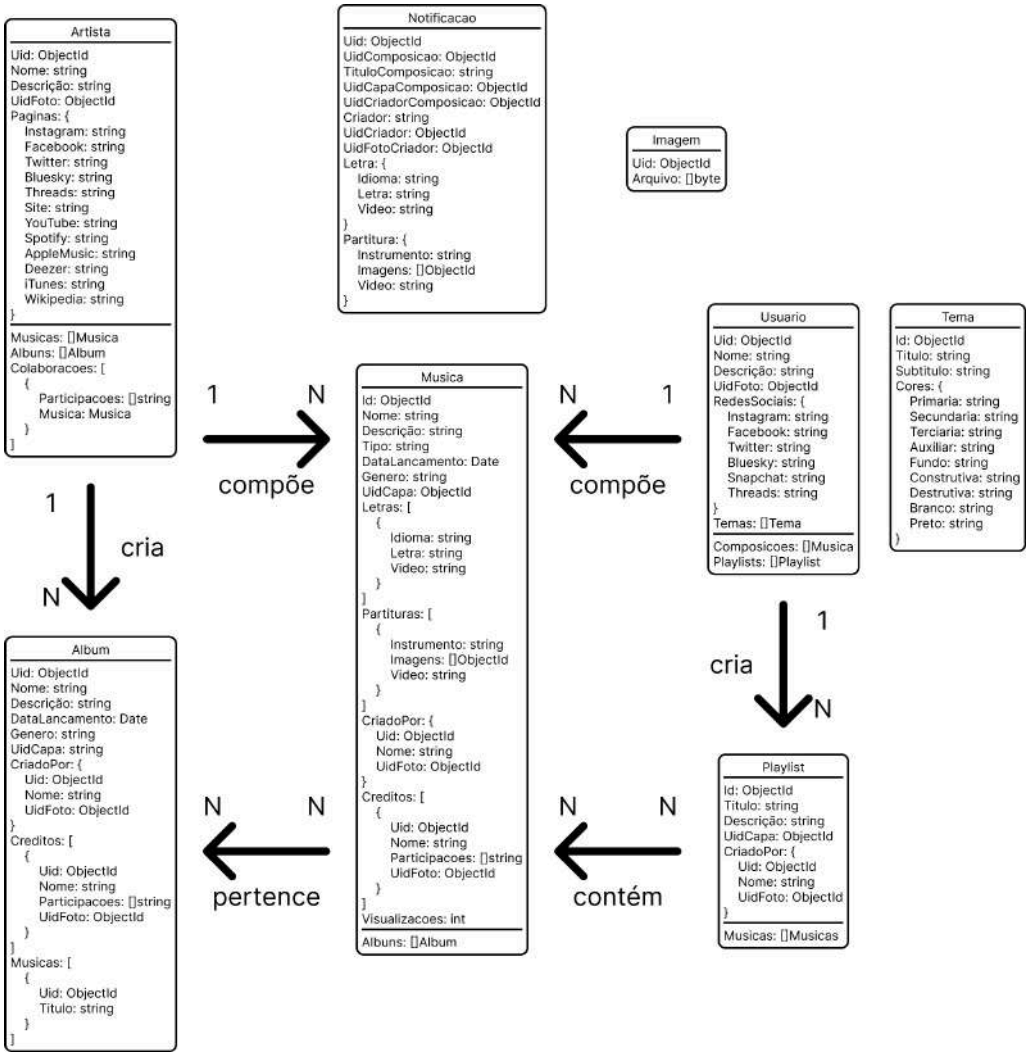
- **Autenticação**: armazena os dados referentes às credenciais utilizadas pelos usuários para acessar a plataforma e disponibiliza-os ao servidor. Neste projeto, será utilizado o *Firebase Authentication*.

3.3.1 Modelo de dados do banco principal

Além de delimitar as camadas da aplicação, também é vital planejar e estruturar o banco de dados, a fim de garantir que seu objetivo seja alcançado de forma adequada e performática. Assim, a modelagem ocorre a partir de características pontuais, como o corpo das coleções e a maneira como elas interagem entre si, com a escolha entre relacionamentos ou dados incorporados.

A figura 6 é o desenho do modelo definido para o banco de dados principal deste projeto. As coleções estão delimitadas pelas linhas contínuas, organizadas no formato nome, dados simples e incorporados e dados provenientes de relacionamentos; estes, por sua vez, estão representados pelas setas, que indicam a direção e a cardinalidade.

Figura 6 – Modelo do banco principal



Fonte: Elaborado pelo autor (2025)

3.4 TECNOLOGIAS UTILIZADAS

Aqui são apresentadas as tecnologias utilizadas para o desenvolvimento do projeto, que foram mencionadas brevemente na seção anterior.

3.4.1 React

React[2] é uma biblioteca criada para auxiliar no desenvolvimento de interfaces

[2] https://react.dev/

para o ambiente *web*, atuante na potencialização da escrita de código na linguagem de *scripts JavaScript* (JS), que, em conjunto com as linguagens de marcação *HyperText Markup Language* (HTML) e *Cascading Style Sheets* (CSS), tornaram-se o principal conjunto de desenvolvimento para o ambiente mencionado. Dentre as vantagens obtidas ao incorporar essa biblioteca às tecnologias do projeto, se destacam o reaproveitamento de código, dinamicidade ao renderizar conteúdos variáveis e praticidade para controle e manipulação de estados da aplicação.

3.4.2 Spring Boot

Spring Boot[3] é uma ferramenta que simplifica os processos de configuração e desenvolvimento de aplicações com o *Spring Framework*[4], um *framework* da linguagem de programação *Java* que, dentre outras funcionalidades, permite a criação de APIs REST, ideal para a codificação de servidores para projetos como este. Com isso, as APIs criadas a partir dessa ferramenta obtêm um ótimo desempenho para comunicação com clientes, bancos de dados e até outras aplicações, além de garantir um rápido processamento de dados internamente.

3.4.3 MongoDB

MongoDB[5] é um banco de dados não-relacional orientado a documentos que armazena as informações de maneira flexível e extensível, além de garantir pesquisas de dados extremamente otimizadas. Quando comparado a bancos relacionais, essa tecnologia traz uma maior maleabilidade para a criação de modelos/esquemas, possibilitando sua construção de forma que evite concatenações com lógicas complexas e incoerências nas estruturas de armazenamento utilizadas.

3.4.4 Redis

Redis[6] é um banco de dados em memória que se destaca pela sua alta velocidade para armazenamento e recuperação de dados. Pela sua natureza, é

[3] https://spring.io/projects/spring-boot
[4] https://spring.io/
[5] https://www.mongodb.com/
[6] https://redis.io/pt/

comum ser implementado em projetos para lidar com informações que não necessitam de atualização em tempo real e são muito requisitadas pelos usuários, garantindo à aplicação um menor tempo de processamento e uma melhor experiência aos requisitantes.

Ademais, o *Redis* também pode atuar como agente de mensagens, ao criar e monitorar uma fila de atividades que serão executadas pelo servidor assim que possível, expandindo o alcance das vantagens supracitadas.

3.4.5 Firebase Authentication e Firestore

Firebase[7] é uma plataforma que oferta diversos serviços de computação em nuvem, no formato de soluções por demanda, com o intuito de auxiliar no desenvolvimento de aplicações, em especial para os ambientes *web* e móvel. Dentre as várias opções presentes, destacam-se *Authentication* e *Firestore*, descritas a seguir.

Firebase Authentication é um serviço que possibilita a fácil manipulação de informações relacionadas ao acesso do usuário à plataforma, gerenciando suas credenciais de forma segura e disponibilizando métodos de autenticação e validação da identidade da conta sem agregar complexidades de implementação ao projeto.

Firebase Firestore, assim como o *MongoDB*, é um banco de dados não-relacional orientado a documentos, com o diferencial de conseguir se comunicar de maneira simplificada com os demais serviços da plataforma. Para este projeto, sua escolha foi feita para o armazenamento das informações não-credenciais dos usuários, visando a centralização de todos os dados das contas criadas em um único ecossistema.

3.5 TELAS DA APLICAÇÃO

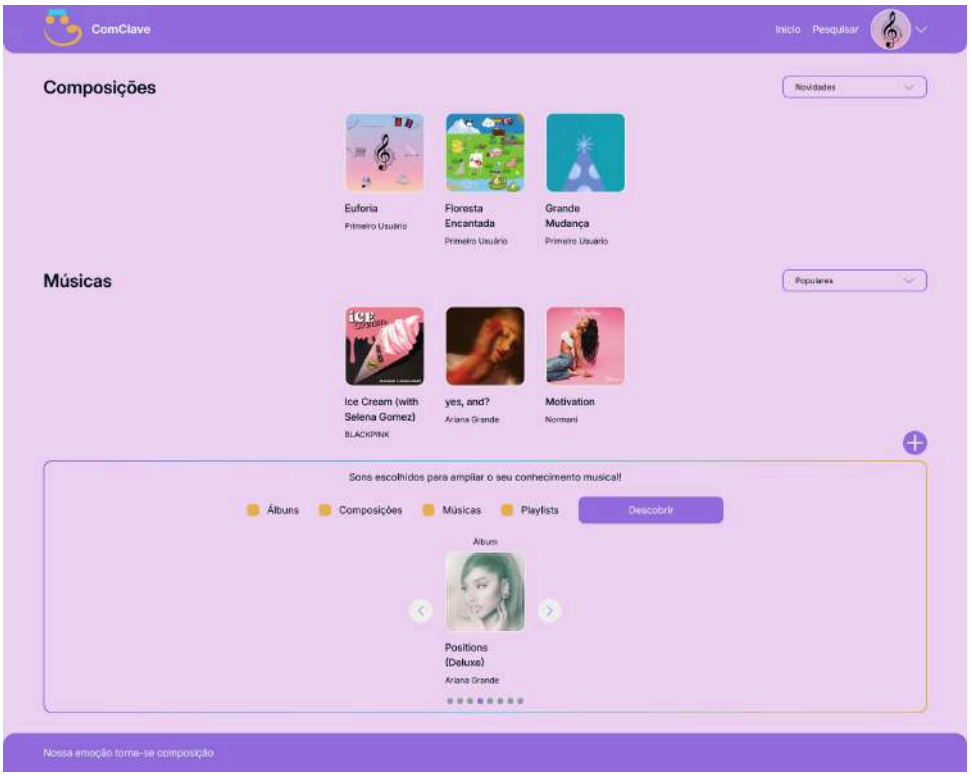
Para concluir a descrição do projeto e demonstrar sua versão final, a seguir serão listadas algumas figuras referentes ao sistema e sua aparência.

A figura 7 apresenta a página inicial do ComClave, com a exposição de

[7] https://firebase.google.com/?hl=pt-br

composições e músicas sortidas por data de lançamento e popularidade, respectivamente, e uma seleção de itens aleatórios a partir do filtro definido pelo usuário. Também é possível ver um ícone de mais que, quando pressionado, tornará visível o submenu para inclusão de novos itens à plataforma.

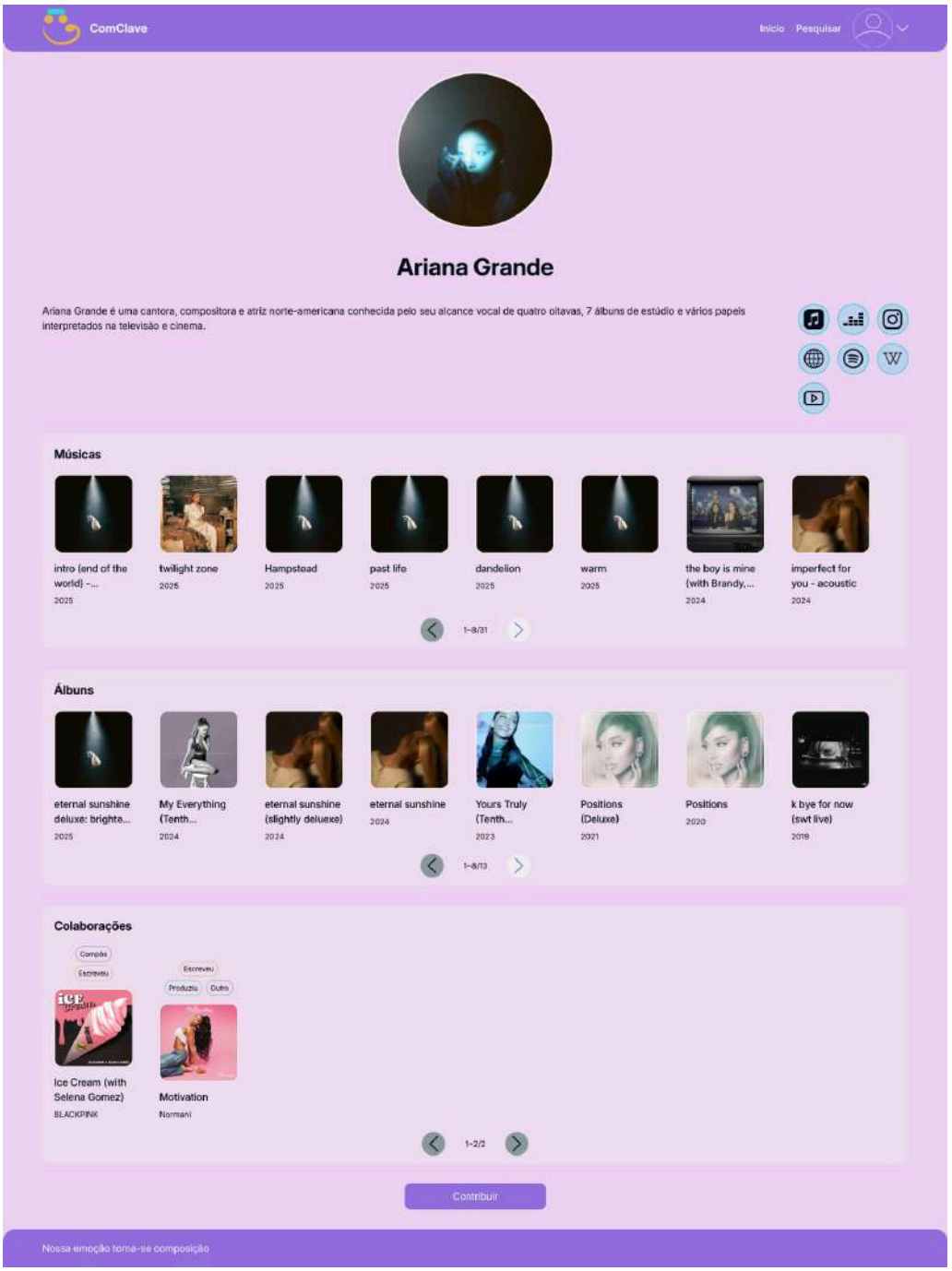
Figura 7 – Página inicial



Fonte: Elaborado pelo autor (2025)

A figura 8 apresenta a página de artista em sua versão de visualização, com a disposição inicial de sua foto, nome, descrição e demais plataformas onde está presente. Em seguida, são elencados, em formato de carrossel, três coleções: a primeira exibe as músicas em que é destaque principal; a segunda exibe os álbuns em que é destaque principal; e a terceira exibe as músicas em que participou da construção, bem como um indicador de qual foi a sua contribuição para a completude da obra.

Figura 8 – Página de artista (visualização)

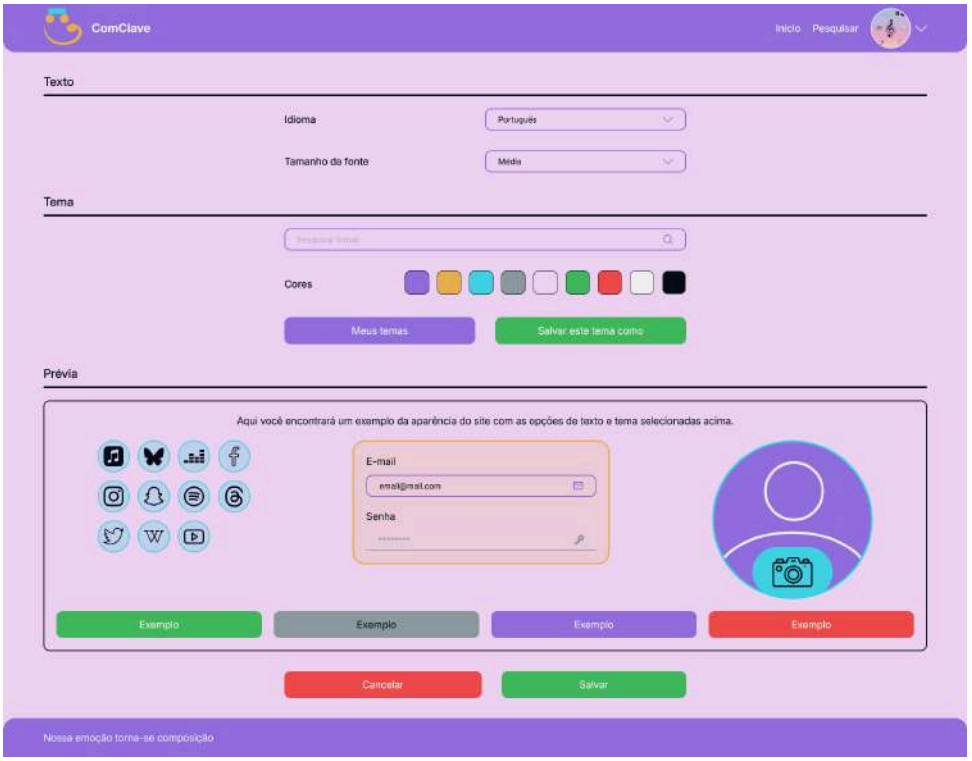


Fonte: Elaborado pelo autor (2025)

A figura 9 apresenta a página de configurações da plataforma, que possibilita sua modificação por meio da alteração do idioma principal, mudança no tamanho

dos textos e personalização das cores. Ao final está disposto uma prévia de como o visual do site será caso as novas opções forem aplicadas.

Figura 9 – Página de configurações



Fonte: Elaborado pelo autor (2025)

Caso queira conferir todas as telas, as demais figuras que apresentam a aplicação estão dispostas no Apêndice A deste documento.

4 CONSIDERAÇÕES FINAIS

Ao objetivar o compartilhamento de músicas em sua estrutura mais básica — enquanto compostas por letras e instrumentais —, viabilizar um ambiente no qual os usuários podem interagir entre si para criar novas obras de maneira colaborativa e, assim, reduzir as barreiras de conhecimentos técnicos, baixa afinidade com determinadas etapas e ausência de meios de execução que poderiam limitar a produção artística de uma pessoa, o projeto obteve êxito na sua proposta inicial e conseguiu estabelecer um espaço para que as inúmeras características sociais e culturais de cada indivíduo possam circular, transparecer e se complementar por meio da música.

Ademais, na medida em que a visão de futuro da plataforma se expande, é esperado que os próximos passos para a sua evolução sejam direcionados a pontos que contribuam para a maior complexidade das criações, por exemplo: uma maneira para que os usuários possam interagir com maior dinamicidade durante os processos construtivos, com a adição de comentários particulares às sugestões ou a implementação de mensagens em grupo para cada composição; complementar a estrutura das músicas com mais aspectos que a contemplam, como a temporização de letras e partituras conforme a sua progressão para habilitar uma visualização mais fluida; e a instauração de uma área destinada a explicar os conceitos e teorias musicais de maneira direta, atuante para facilitar a compreensão da estrutura das obras e auxiliar durante os momentos de construções. Também, para a evolução da experiência do usuário, foram mapeadas a criação de um sistema de recomendações baseado nos gostos e preferências individuais e o estabelecimento de meios que garantem a autenticidade e autoria das composições, como a manutenção do histórico de edições e validações internas prévias às modificações.

REFERÊNCIAS

AIDAR, Laura. **Significado de Arte (O que é, Conceito e Definição)**. 2024. Disponível em: <https://www.significados.com.br/arte/>. Acesso em 04 de mar. de 2025.

BREWER, E. CAP Twelve Years Later: How the “Rules” Have Changed. **Computer**, v. 45, n. 2, p. 23–29, fev. 2012.

FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures**. 2000. 180 p. Dissertação (Doutorado em Filosofia em Ciências da Computação e Informação) - Universidade da Califórnia, [*S. l.*], 2000. Disponível em: <https://ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf>. Acesso em: 26 fev. 2025.

FOWLER, A. **NoSQL for Dummies**. Hoboken, Nj: John Wiley & Sons, Inc, 2015.

FOWLER, M. **Patterns of Enterprise Application Architecture**. Boston, Mass. ; Munich: Addison-Wesley, 2015.

GOUGH, J.; BRYANT, D.; AUBURN, M. **Mastering API Architecture**. [s.l.] “O’Reilly Media, Inc.”, 2022.

HAERDER, T.; REUTER, A. Principles of transaction-oriented database recovery. **ACM Computing Surveys**, v. 15, n. 4, p. 287–317, 2 dez. 1983.

HILLS, T. **NoSQL and SQL Data Modeling**. [s.l.] Technics Publications, 2016.

MASSÉ, M. **REST API design rulebook [designing consistent RESTful web service interfaces]**. [s.l.] Beijing [U.A.] O’reilly, 2012.

MONGODB. **Document Database - NoSQL**. 2025. Disponível em: <https://www.mongodb.com/resources/basics/databases/document-databases>. Acesso em 25 de fev. de 2025.

ROSA, Letícia. **O que é Cultura?**: E outras perguntas. Jundiaí - SP: [*s. n.*], 2020. 21 p. Disponível em: <https://cultura.jundiai.sp.gov.br/wp-content/uploads/2021/03/E-book-O-que-e-Cultura_-1.pdf>. Acesso em: 5 mar. 2025.

SIGNIFICADOS. **O que é Música?**. 2022. Disponível em: <https://www.significados.com.br/musica/>. Acesso em 24 de ago. de 2025.

SOMMERVILLE, I. **Software Engineering**. [s.l.] Pearson Higher Ed, 2011.

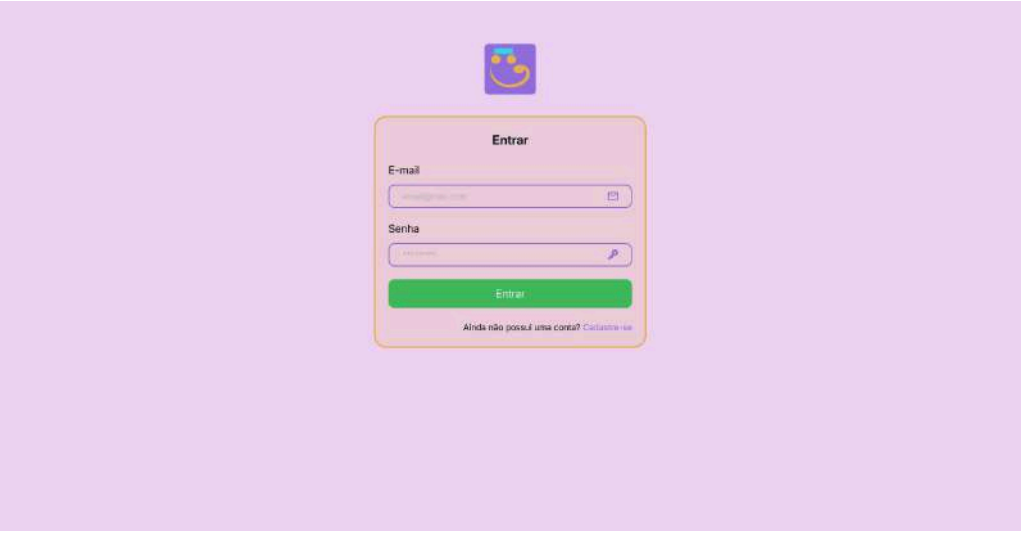
SOUNDPLATE. **The Major Labels – Everything You Need To Know About Major Record Labels**. 2021. Disponível em: <https://soundplate.com/major-record-labels/>. Acesso em 05 de mar. de 2025.

TYLOR, E. B. **Primitive culture : researches into the development of mythology, philosophy, religion, language, art and custom**. Mineola, New York: Dover Publications, 1871. p.1.

APÊNDICE A – PÁGINAS DA APLICAÇÃO

A figura 10 apresenta a página para acesso da conta, com a logo do ComClave na posição superior central, seguida pelo formulário de entrada, para que o usuário informe seu e-mail e senha. Caso não possua uma conta, a opção para criá-la está disposta ao final do corpo.

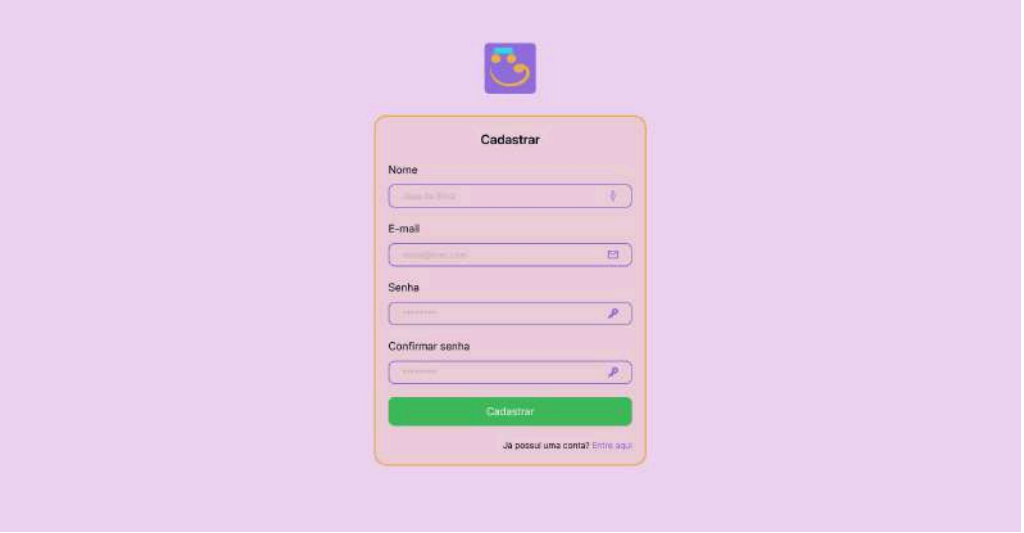
Figura 10 – Página de acesso



Fonte: Elaborado pelo autor (2025)

A figura 11 apresenta a página de cadastro, com a logo do ComClave na posição superior central, seguida pelo formulário de criação de conta, para que o usuário informe seu nome e e-mail e defina uma senha. Caso já possua uma conta, a opção para acessá-la está disposta ao final do corpo.

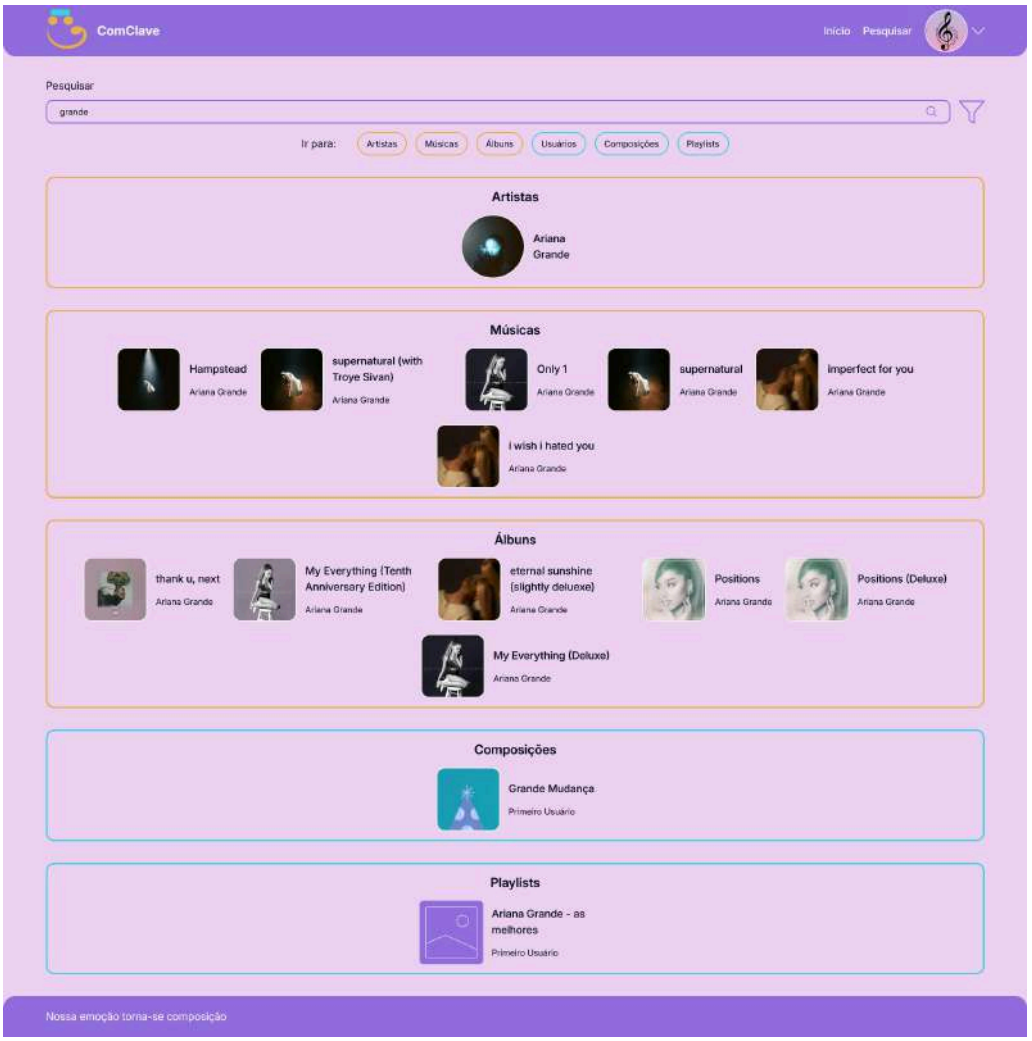
Figura 11 – Página de cadastro



Fonte: Elaborado pelo autor (2025)

A figura 12 apresenta a página de busca, que permite a pesquisa por qualquer item de interesse a partir de um texto simples. Os resultados ficam agrupados por tipo e são elencados em uma lista de coleções disposta logo em seguida.

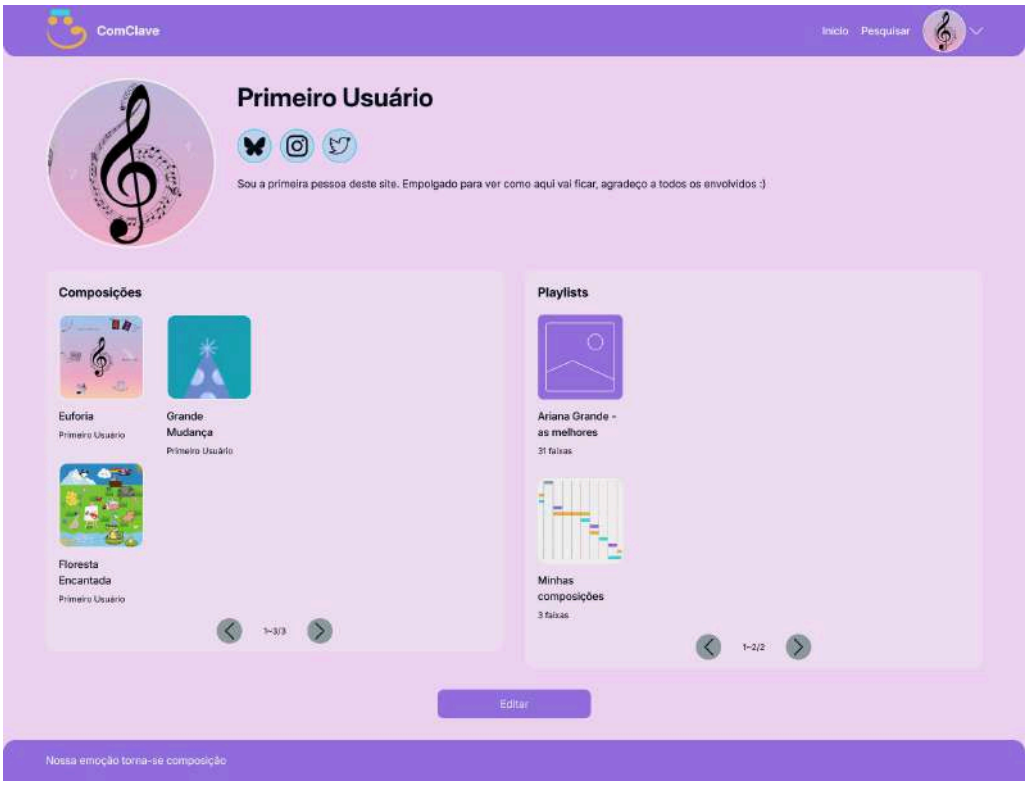
Figura 12 – Página de busca



Fonte: Elaborado pelo autor (2025)

A figura 13 apresenta a página de perfil do usuário na versão de visualização, com a disposição inicial de sua foto, nome, redes sociais e descrição, seguido de suas coleções (composições e *playlists*).

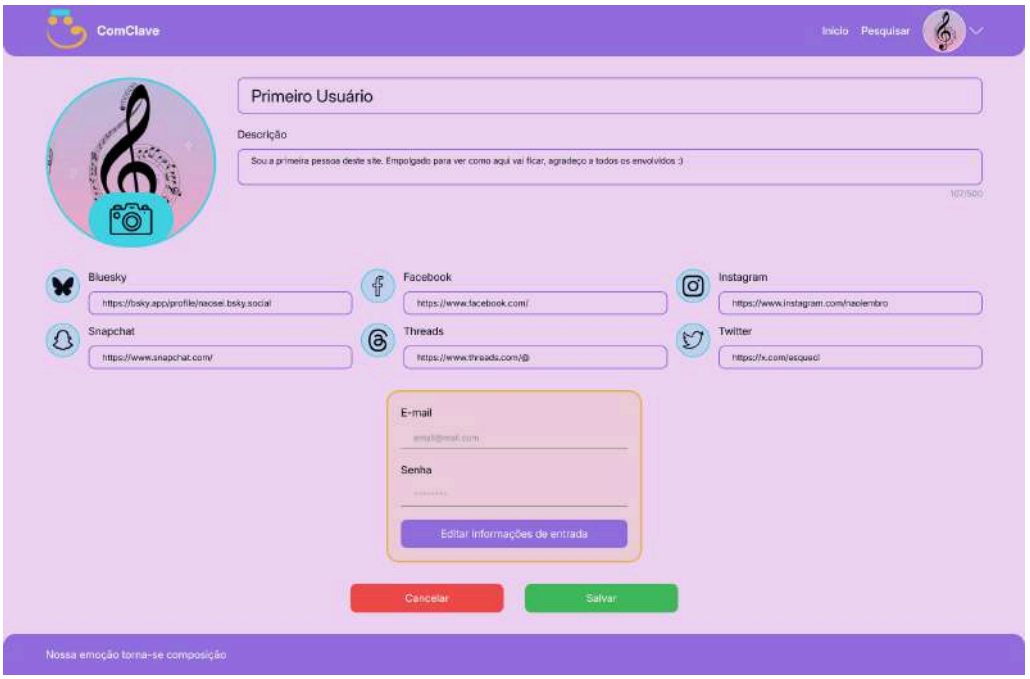
Figura 13 – Página de perfil do usuário (visualização)



Fonte: Elaborado pelo autor (2025)

A figura 14 apresenta a página de perfil do usuário na versão de edição, com a disposição inicial de sua foto, nome e descrição, seguido por suas redes sociais e, finalmente, suas informações de acesso à plataforma.

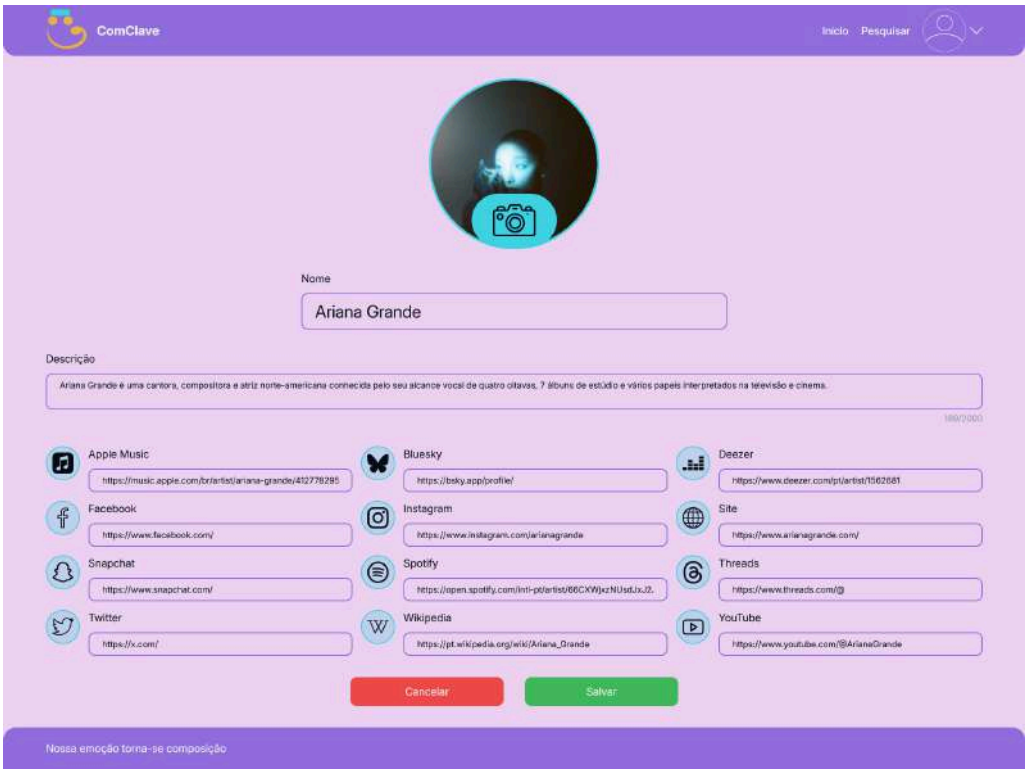
Figura 14 – Página de perfil do usuário (edição)



Fonte: Elaborado pelo autor (2025)

A figura 15 apresenta a página de artista na versão de edição, com a disposição inicial de sua foto, nome e descrição, seguido pelas plataformas nas quais está presente.

Figura 15 – Página de artista (edição)



Fonte: Elaborado pelo autor (2025)

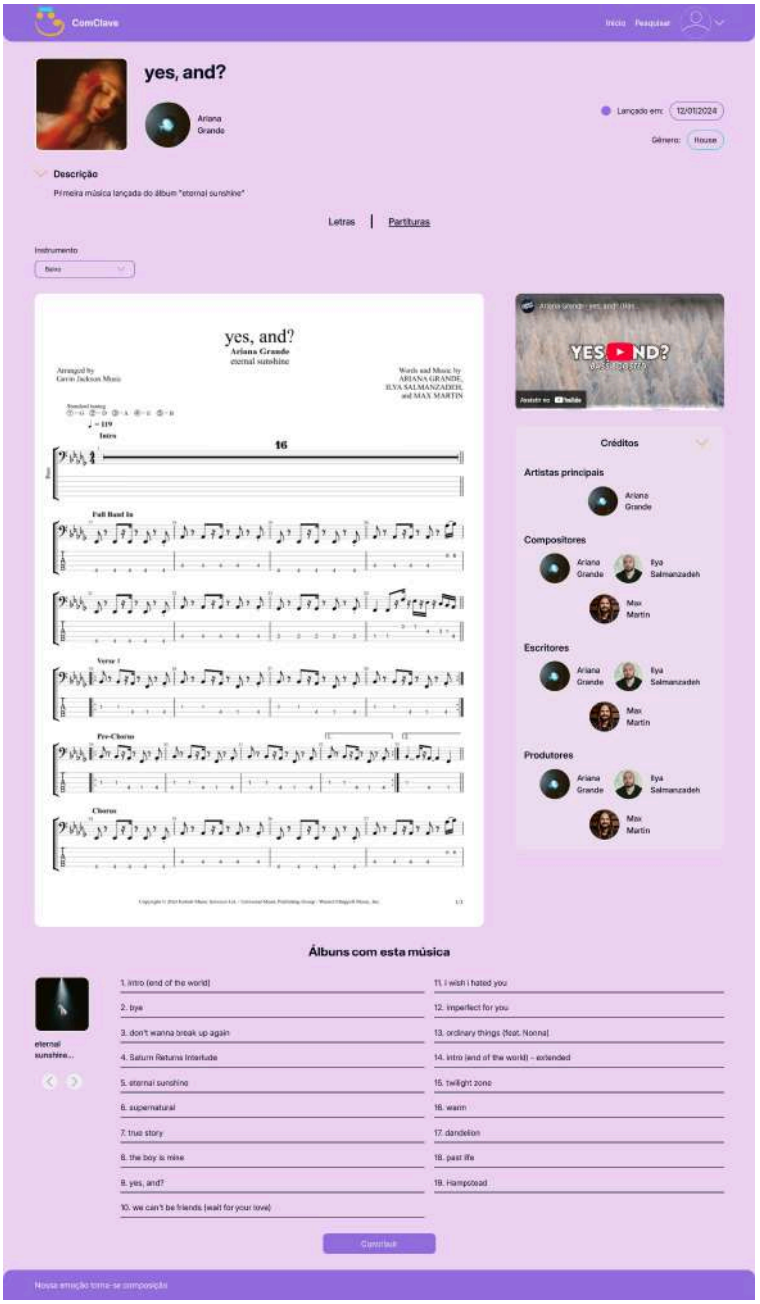
A figura 16 apresenta a página de composição/música na versão de visualização das letras, disposta em três seções: informações básicas, com capa, nome, artista de referência, data de lançamento, gênero e descrição; conteúdo principal, com letra e vídeo referentes ao idioma selecionado e créditos gerais; e conteúdos relacionados, com álbuns ao qual a música pertence (ausente para composições).

Figura 16 – Página de composição/música [letras] (visualização)

Fonte: Elaborado pelo autor (2025)

A figura 17 apresenta a página de composição/música na versão de visualização das partituras, disposta em três seções: informações básicas, com capa, nome, artista de referência, data de lançamento, gênero e descrição; conteúdo principal, com partitura e vídeo referentes ao instrumento selecionado e créditos gerais; e conteúdos relacionados, com álbuns ao qual a música pertence (ausente para composições).

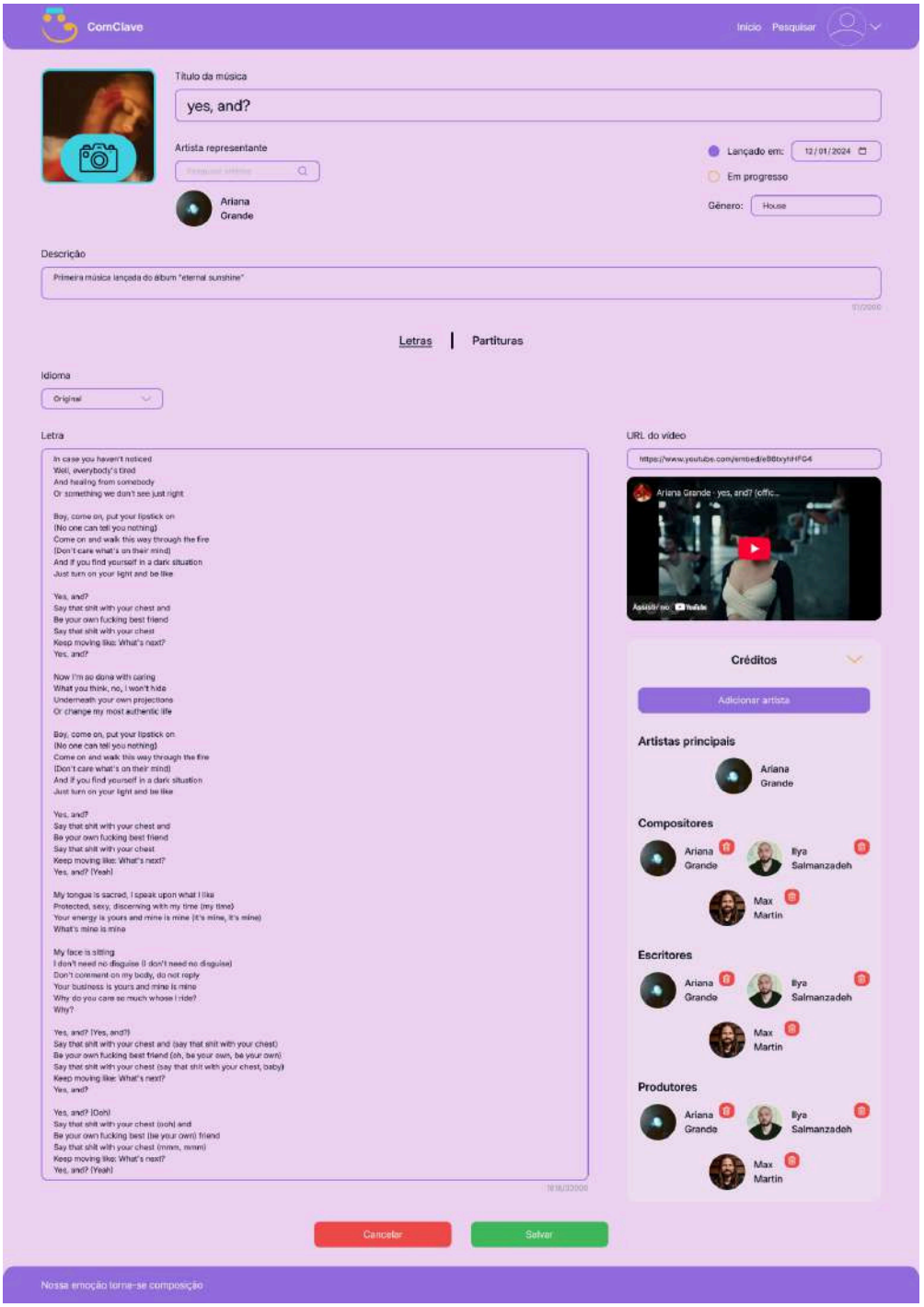
Figura 17 – Página de composição/música [partituras] (visualização)



Fonte: Elaborado pelo autor (2025)

A figura 18 apresenta a página de composição/música na versão de edição das letras, disposta em duas seções: informações básicas, com capa, nome, artista de referência, data de lançamento, gênero e descrição; e conteúdo principal, com letra e vídeo referentes ao idioma selecionado e créditos gerais.

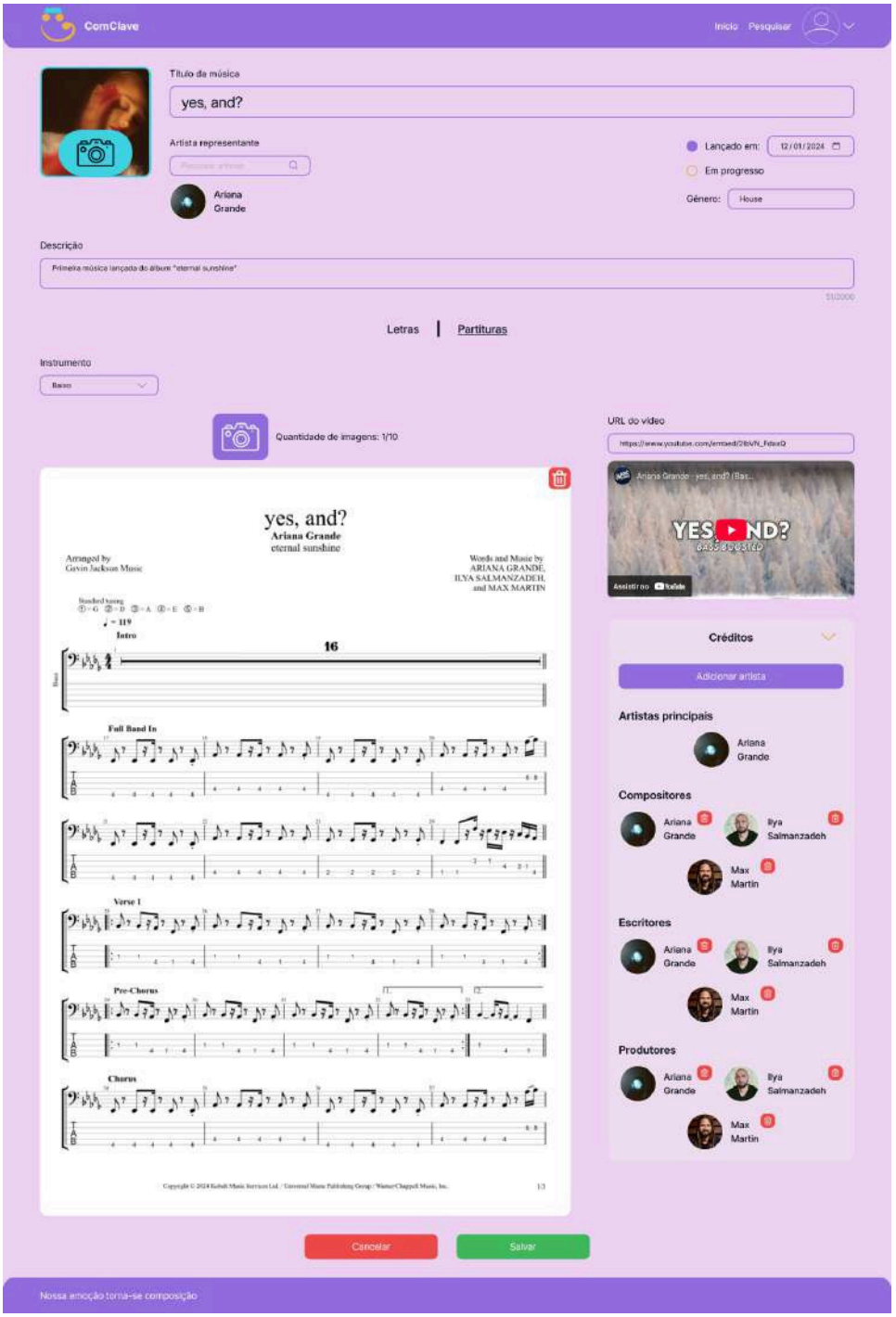
Figura 18 – Página de composição/música [letras] (edição)



Fonte: Elaborado pelo autor (2025)

A figura 19 apresenta a página de composição/música na versão de edição das partituras, disposta em duas seções: informações básicas, com capa, nome, artista de referência, data de lançamento, gênero e descrição; e conteúdo principal, com partitura e vídeo referentes ao instrumento selecionado e créditos gerais.

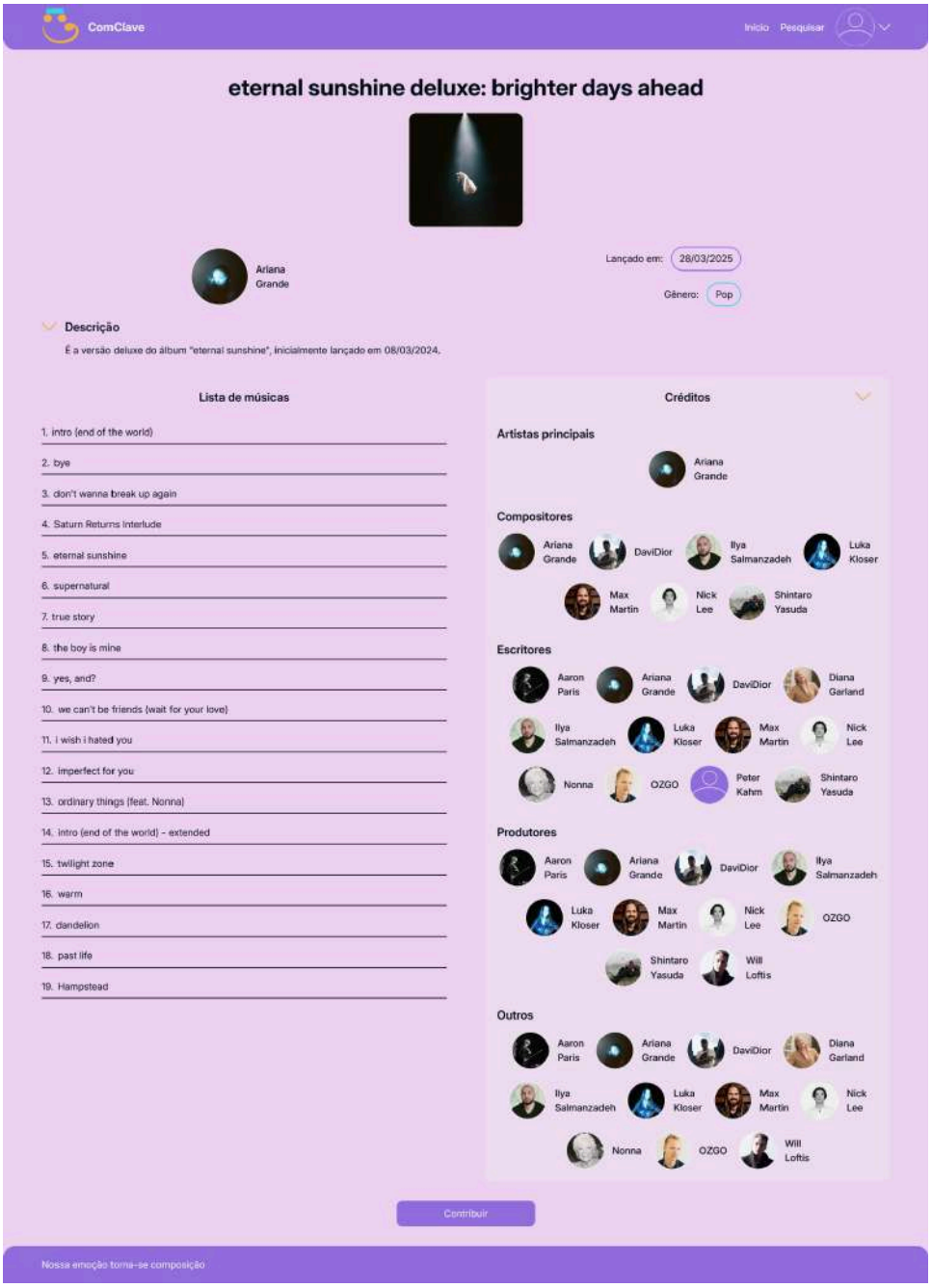
Figura 19 – Página de composição/música [partituras] (edição)



Fonte: Elaborado pelo autor (2025)

A figura 20 apresenta a página de álbum na versão de visualização, que contempla nome, capa, artista de referência, data de lançamento, gênero e descrição, além de uma lista com todas as músicas que o compõem e créditos da construção.

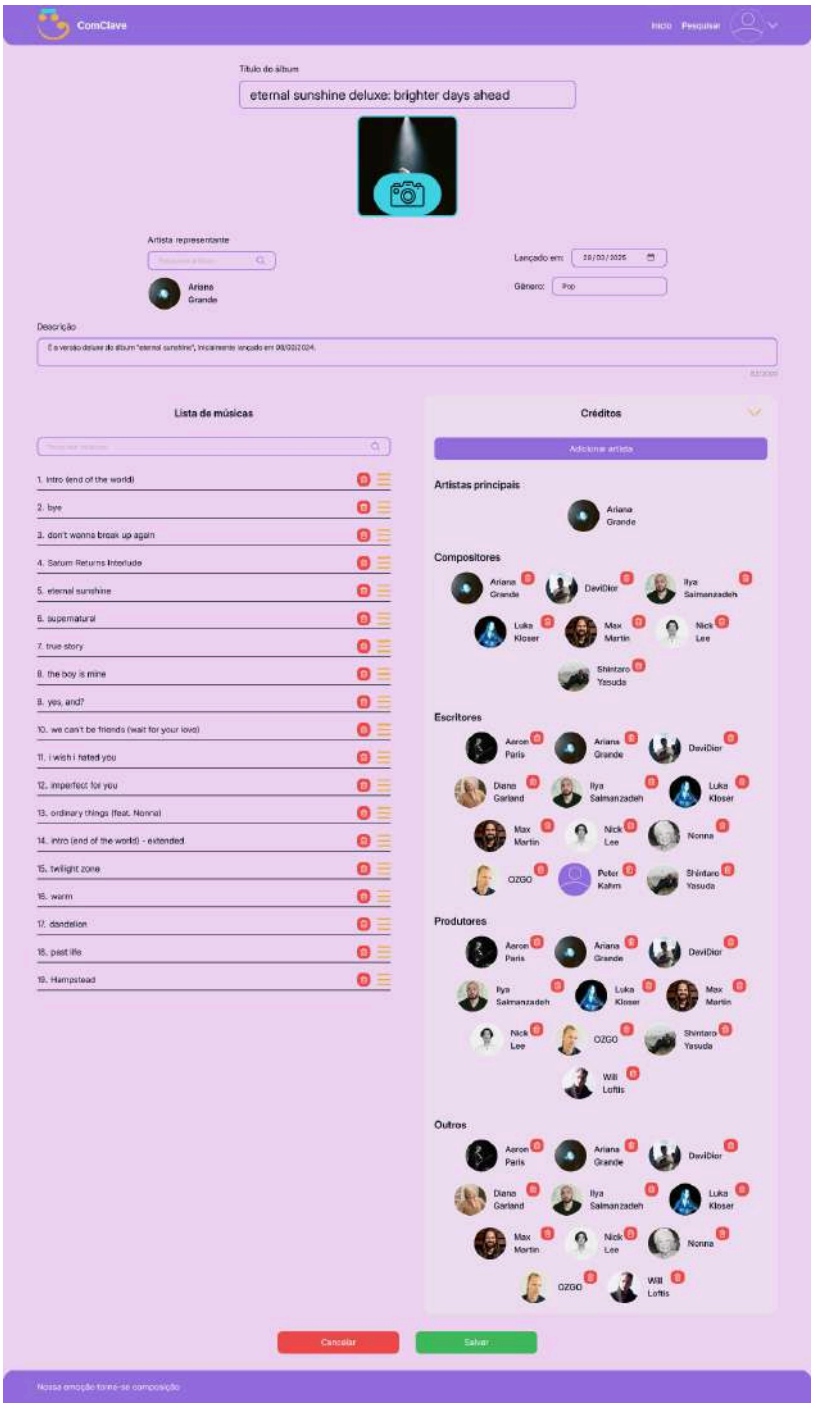
Figura 20 – Página de álbum (visualização)



Fonte: Elaborado pelo autor (2025)

A figura 21 apresenta a página de álbum na versão de edição, que contempla nome, capa, artista de referência, data de lançamento, gênero e descrição, além de uma lista com todas as músicas que o compõem e créditos da construção.

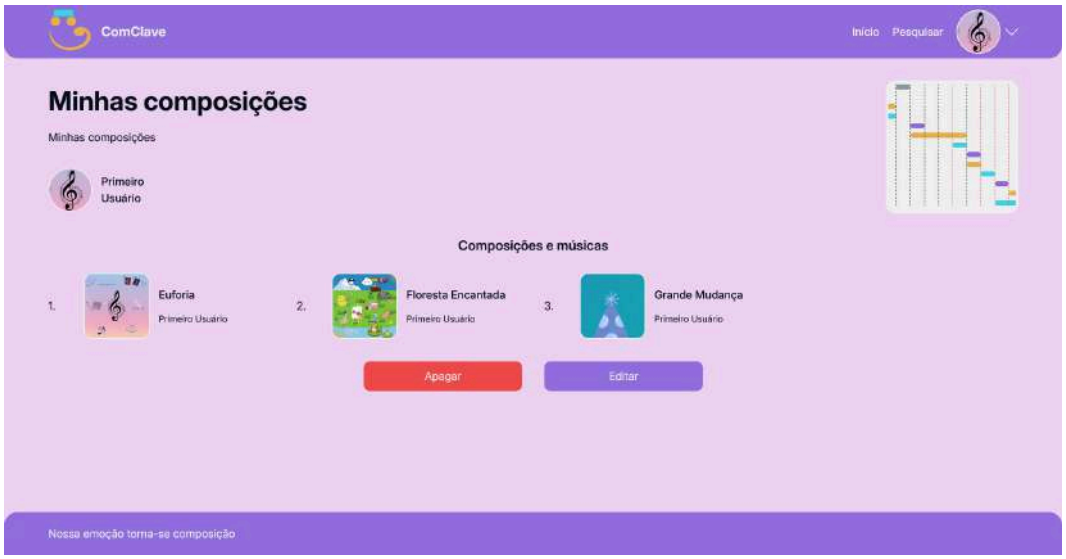
Figura 21 – Página de álbum (edição)



Fonte: Elaborado pelo autor (2025)

A figura 22 apresenta a página de *playlist* na versão de visualização, que contém nome, descrição, usuário criador e capa, seguido de uma grade com todas as composições e músicas que estão englobadas em seu corpo.

Figura 22 – Página de *playlist* (visualização)



Fonte: Elaborado pelo autor (2025)

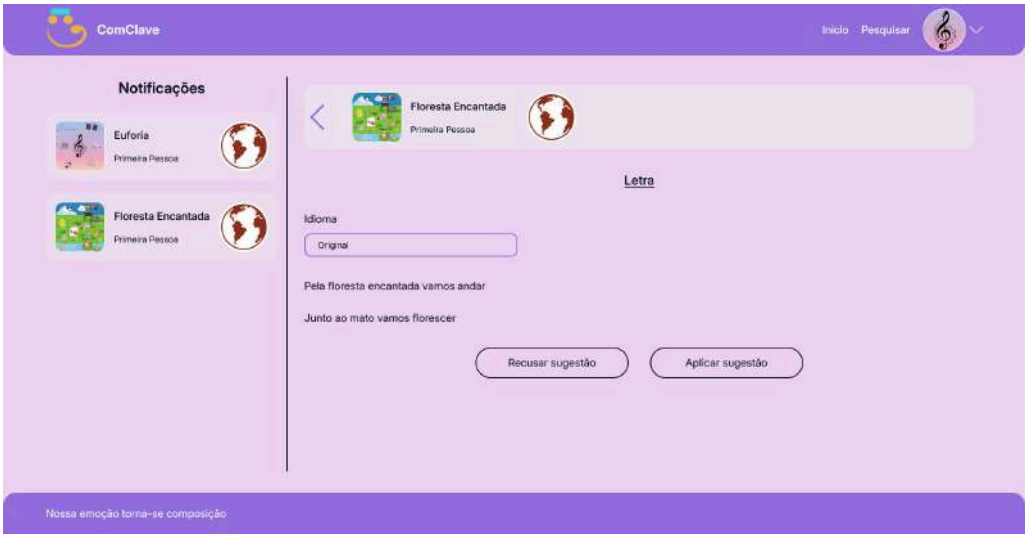
A figura 23 apresenta a página de *playlist* na versão de edição, que contém nome, descrição e capa, seguido de uma grade com todas as composições e músicas que estão englobadas em seu corpo.

Figura 23 – Página de *playlist* (edição)

Fonte: Elaborado pelo autor (2025)

A figura 24 apresenta a página de notificações. À esquerda, está disposta uma lista com todas as notificações que o usuário recebeu referentes às suas composições, organizadas como capa e nome da obra e nome e foto de perfil da pessoa que a enviou; à direita, está disposto o detalhamento da notificação, com a exposição da letra e/ou partitura sugerido(s) para a composição, bem como as opções de recusar ou aplicá-la.

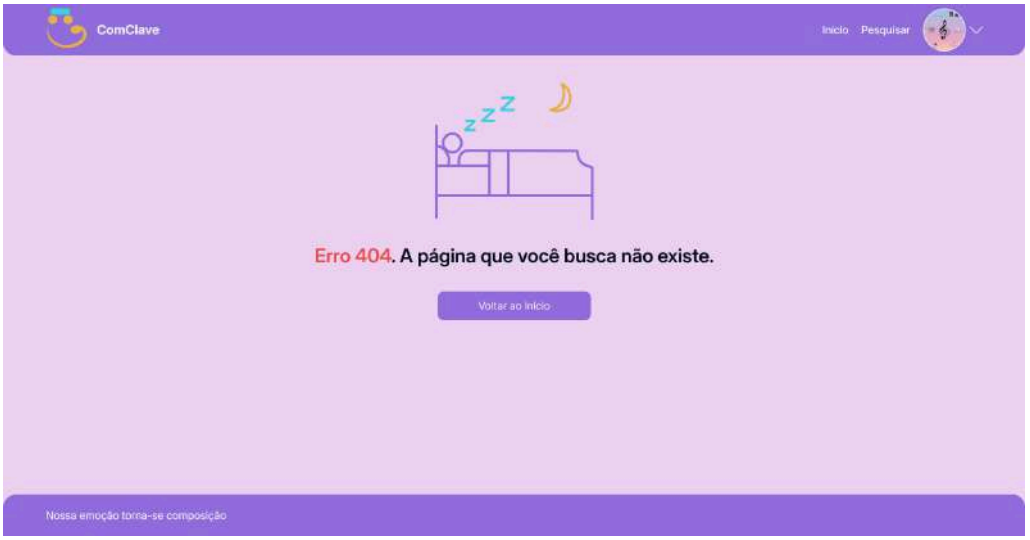
Figura 24 – Página de notificações



Fonte: Elaborado pelo autor (2025)

A figura 25 apresenta a página não encontrada, uma mostra simples de imagem seguida de texto e botão de retorno para indicar que o usuário tentou acessar algo que requer uma permissão mais elevada do que a sua atual.

Figura 25 – Página não encontrada



Fonte: Elaborado pelo autor (2025)

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
Campus Cajazeiras - Código INEP: 25008978
Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

Documento Digitalizado Restrito

Documento do TCC

Assunto:	Documento do TCC
Assinado por:	Caio Pinheiro
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Restrito
Hipótese Legal:	Proteção da Propriedade Intelectual de Software (Art. 2o da Lei no 9.609/1998)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Caio Pinheiro Guedes**, **DISCENTE (202112010018) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 12/09/2025 14:21:55.

Este documento foi armazenado no SUAP em 12/09/2025. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse https://suap.ifpb.edu.br/verificar-documento-externo/ e forneça os dados abaixo:

Código Verificador: 1607299
Código de Autenticação: a6305edc46