**MusicConnect: Sistema de conexão entre Músicos e Contratantes**

**João Víctor Dias Souza, Lucas Coutinho, Lucas Morais, Victor Vidal**

1. **Introdução**

A contratação de músicos para eventos ao vivo tem se tornado cada vez mais comum, facilitando o início de carreiras para músicos iniciantes. Contudo, a busca por músicos e a visibilidade de vagas ainda apresentam desafios, como sendo necessário conhecer o músico ou por meio de indicações de lugares onde já possuem esse tipo de serviço, gerando uma limitação e dificuldades para ambos os lados. A MusiConnect, é uma plataforma desenvolvido para conectar e facilitar este processo de contratação e visibilidade de vagas disponíveis, ele concentra os músicos e estabelecimentos tudo em um único ambiente, ou como dito no sistema os contratantes, pois qualquer pessoa pode se cadastrar e contratar um músico para o seu evento. Está conexão é feito a partir de anúncios dos contratantes ou contratação diretamente pelo perfil do músico desejado, neste perfil se têm o próprio portfólio do músico contendo suas habilidades, descrições, contratos realizados, feedbacks e muito mais. Os músicos podem acessar anúncios disponíveis e se candidatar as vagas oferecidas pelos contratantes, a partir disso o usuário é selecionado e eles fecham um contrato.

1. **Validação da Ideia**

Para confirmar os problemas e necessidades citadas, foi realizado entrevistas com músicos conhecidos e estabelecimentos que costumam apresentar músicas ao vivo. Nas entrevistas para músicos foi perguntado as seguintes questões: Como atualmente você procura freelances? Como você apresenta seu portfólio ao interessado? Se tivesse uma plataforma que facilitasse este processo, você usaria? Nas entrevistas para contratantes foi perguntado as seguintes questões : Como você atualmente encontra músicos para contratar para eventos? Quais métodos você utiliza para avaliar e selecionar músicos? Quais desafios você enfrenta ao buscar músicos para eventos? Você usaria uma plataforma que centralizasse a busca e a contratação de músicos? As respostas de ambos indicaram uma alta probabilidade de adoção da plataforma, especialmente os músicos que estão começando suas carreiras.

1. **Regras de Negócio**

As regras de negócio, ditam o que o sistema deve possuir como regras, ou seja, algo que será sempre levado em consideração para que ele funcione como foi pensado, evitando possíveis erros. Segue as regras de Negócio adotadas para o sistema MusiConnect :

* Um músico poderá se candidatar a um determinado anúncio apenas uma vez.
* Somente o contratante poderá gerar um contrato.
* Um contratante poderá contratar apenas um músico por anúncio.
* Um anúncio tem prazo de validade até 30 dias, sendo necessário renova-lo para visibilidade novamente.
* É necessário o preenchimento do cadastro avançado, para que o contratante gere um contrato e para que o músico aceite o mesmo.
* Na atual situação do projeto, não é possível que um usuário seja músico e contratante.

1. **Requisitos Funcionais e Não Funcionais**

Requisitos funcionais são especificações que descrevem o que o sistema deve fazer, como funcionalidades, processos, ou comportamentos que o sistema deve suportar. Já Requisitos não funcionais são critérios que definem como o sistema deve operar, incluindo desempenho, segurança, usabilidade, e confiabilidade.

**4.1 Requisitos Funcionais**

* **Cadastro de Usuários:** O sistema deve permitir que músicos e contratantes se registrem com um perfil único.
* **Busca de Anúncios:** O sistema deve permitir que músicos busquem anúncios com base em gênero musical, localização, tipo de evento, etc.
* **Busca de Músicos:** O sistema deve permitir que contratantes busquem perfis de músicos para contrata-los.
* **Sistema de Avaliações:** O sistema deve permitir que contratantes avaliem músicos após a conclusão de um contrato.
* **Upload de Mídia:** O sistema deve permitir que músicos façam upload de vídeos e fotos em seus perfis.

**4.2 Requisitos Não Funcionais**

* **Segurança:** O sistema deve garantir a proteção dos dados pessoais dos usuários.
* **Usabilidade:** O sistema deve ter uma interface intuitiva, permitindo que usuários naveguem facilmente e completem tarefas sem dificuldades.

1. **Ferramentas Utilizadas**

**5.1 Figma**

Figma é uma ferramenta de design colaborativa usada para criar interfaces de usuário e protótipos interativos. Ele permite que equipes de design trabalhem juntas em tempo real, facilitando a criação e o ajuste de layouts, wireframes e protótipos de alta fidelidade. No MusiConnect, o Figma foi utilizado para o design das interfaces, auxiliando na criação de um layout visualmente atraente e na prototipagem das telas antes do desenvolvimento final.

**5.2 HTML**

HTML é a linguagem base para a criação de páginas web. Ele define a estrutura do conteúdo, organizando elementos como textos, imagens, links e tabelas, permitindo que sejam exibidos de maneira correta nos navegadores. No MusiConnect, o HTML é utilizado para estruturar as interfaces, organizando a apresentação das informações para os usuários.

**5.3 CSS**

CSS é uma linguagem de estilo usada para definir a aparência visual de uma página web. Através do CSS, é possível controlar o layout, cores, fontes e outros aspectos visuais de um site. No MusiConnect, o CSS é utilizado para garantir um design consistente e agradável, melhorando a experiência do usuário e a usabilidade do site.

**5.4 JavaScript**

JavaScript é uma linguagem de programação que permite adicionar interatividade às páginas web. Com ela, é possível manipular elementos do HTML, responder a eventos de usuário e criar funcionalidades dinâmicas. No MusiConnect, o JavaScript é empregado para criar interações responsivas e dinâmicas, como validações de formulários e etc.

**5.5 PHP**

PHP é uma linguagem de programação voltada para o desenvolvimento de aplicações web no lado do servidor. Ela permite a criação de páginas dinâmicas, conectando a interface do usuário ao banco de dados e processando dados no servidor. No MusiConnect, o PHP é utilizado para gerenciar o backend do site, incluindo a manipulação de dados de usuários, contratos e anúncios.

**5.6 GitHub e Git**

Git é um sistema de controle de versão utilizado para rastrear mudanças no código durante o desenvolvimento, permitindo colaboração eficiente entre desenvolvedores. GitHub é uma plataforma baseada na nuvem que hospeda repositórios Git, facilitando o gerenciamento de projetos, revisão de código e colaboração em equipe. No MusiConnect, Git e GitHub foram utilizados para versionar o código, coordenar o trabalho da equipe e garantir um fluxo de desenvolvimento organizado e seguro.

**5.7 SQL Server**

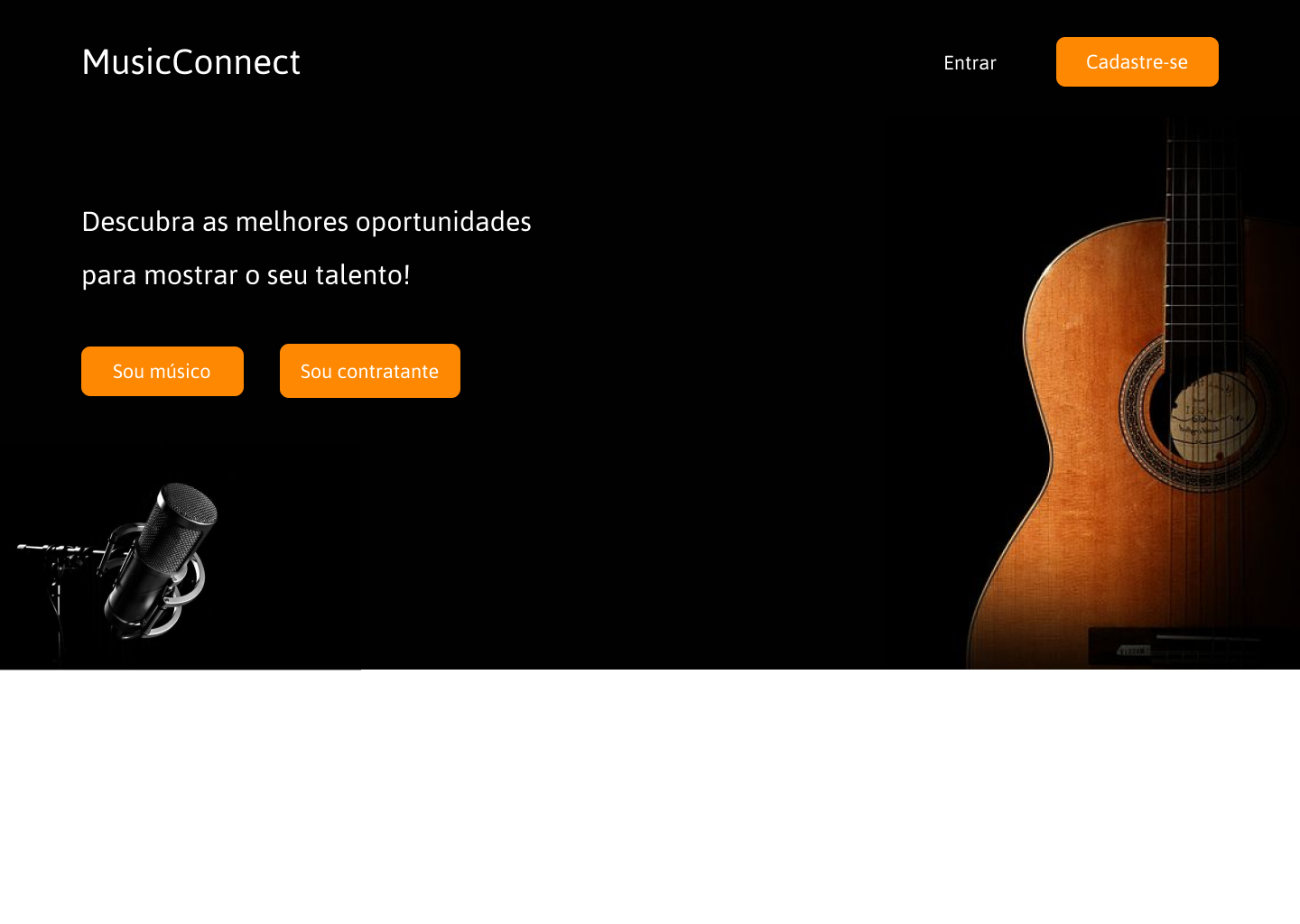
SQL Server é um sistema de gerenciamento de banco de dados relacional (SGBD) desenvolvido pela Microsoft. Ele é responsável por armazenar, consultar e gerenciar grandes volumes de dados de maneira segura e eficiente. No MusiConnect, o SQL Server é usado para organizar e manter o banco de dados, garantindo que as informações dos músicos, contratantes e contratos sejam armazenadas e recuperadas de forma eficiente.

1. **Prototipagem**

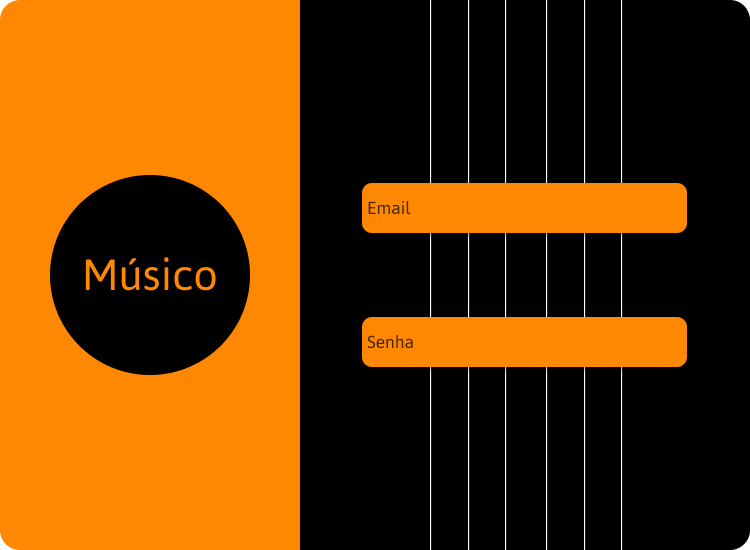
A prototipagem é um processo no qual se cria uma versão inicial da ideia pensada, para reduzir retrabalhos, custos e aumentar a assertividade do projeto, pois a prototipagem se baseia em um protótipo de fácil desenvolvimento e que demonstre a ideia de forma simples e objetiva. Para o sistema MusiConnect, foi desenvolvido alguns protótipos de alto nível através da plataforma Figma, para representar de forma fácil e clara as páginas do sistema.

Os primeiros protótipos foram feitos para as páginas de login, eu sou e home:

* Home



* Login

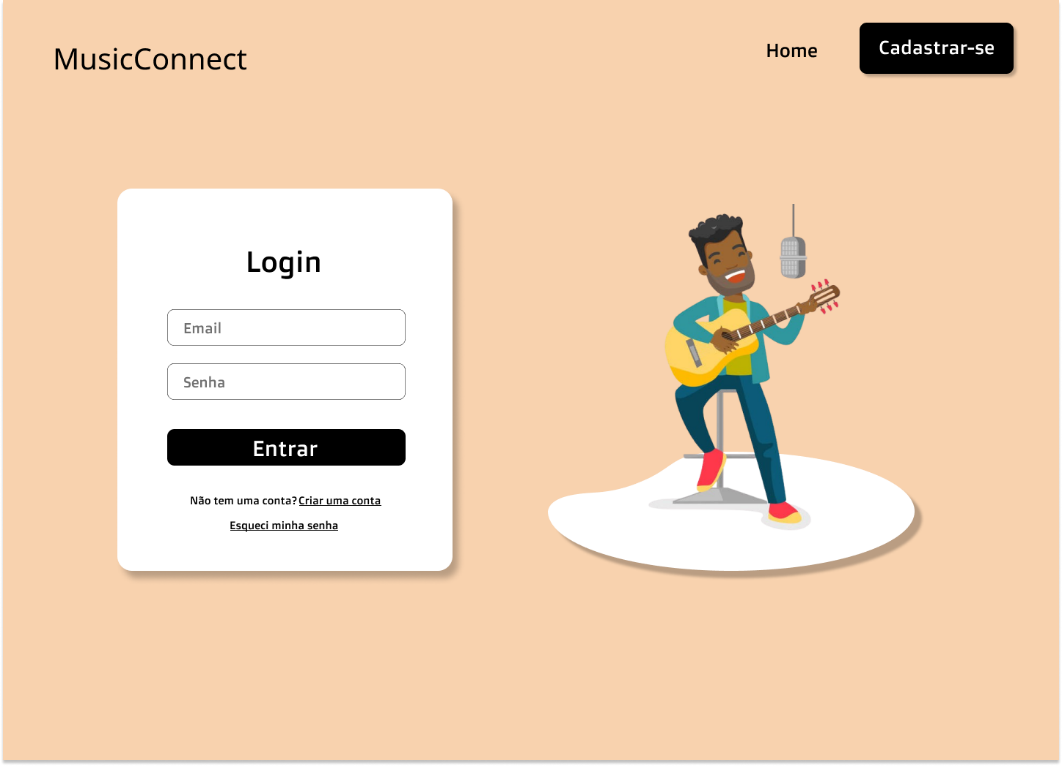


* Eu sou

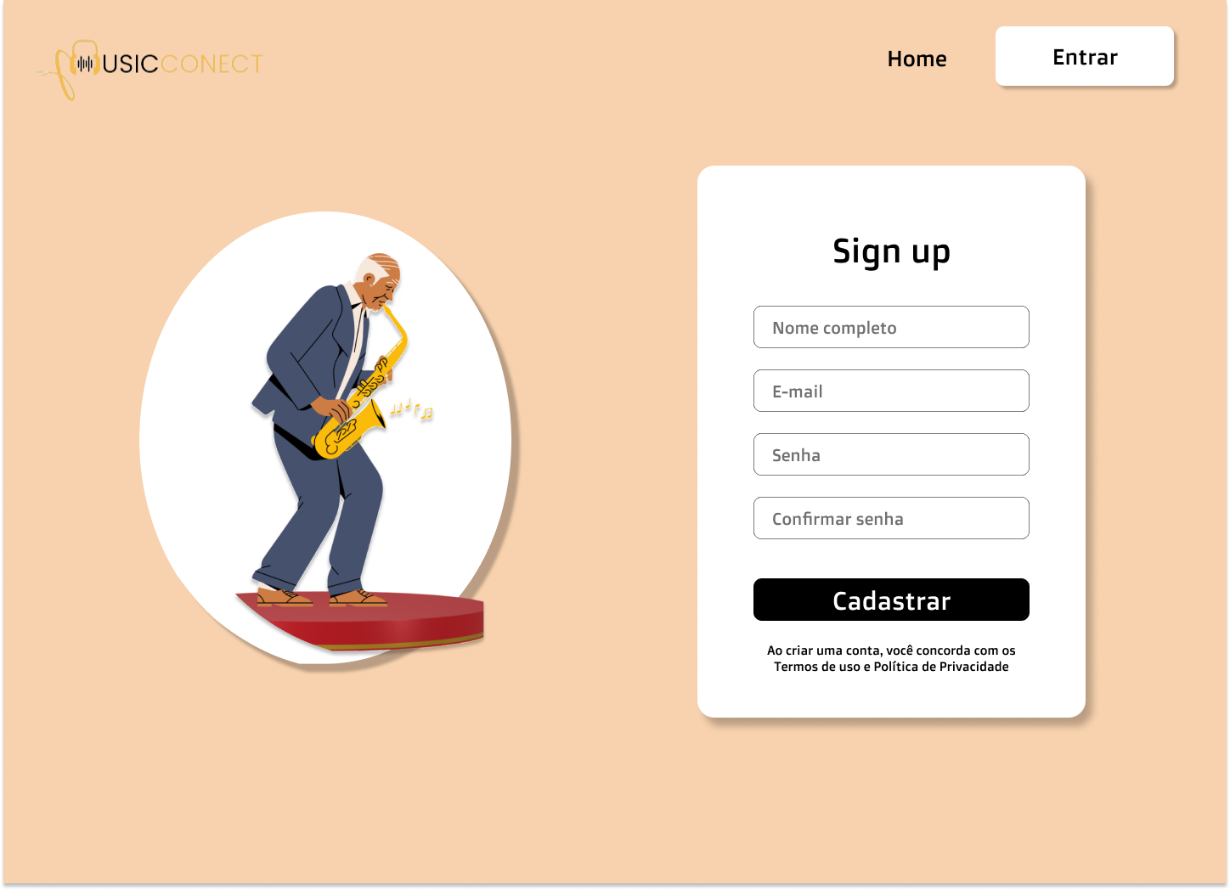


Após algumas observações e melhorias, foi desenvolvido os protótipos que seria utilizado na codificação do sistema, sendo as páginas de login, cadastro, objetivo e buscar anúncios:

* Login



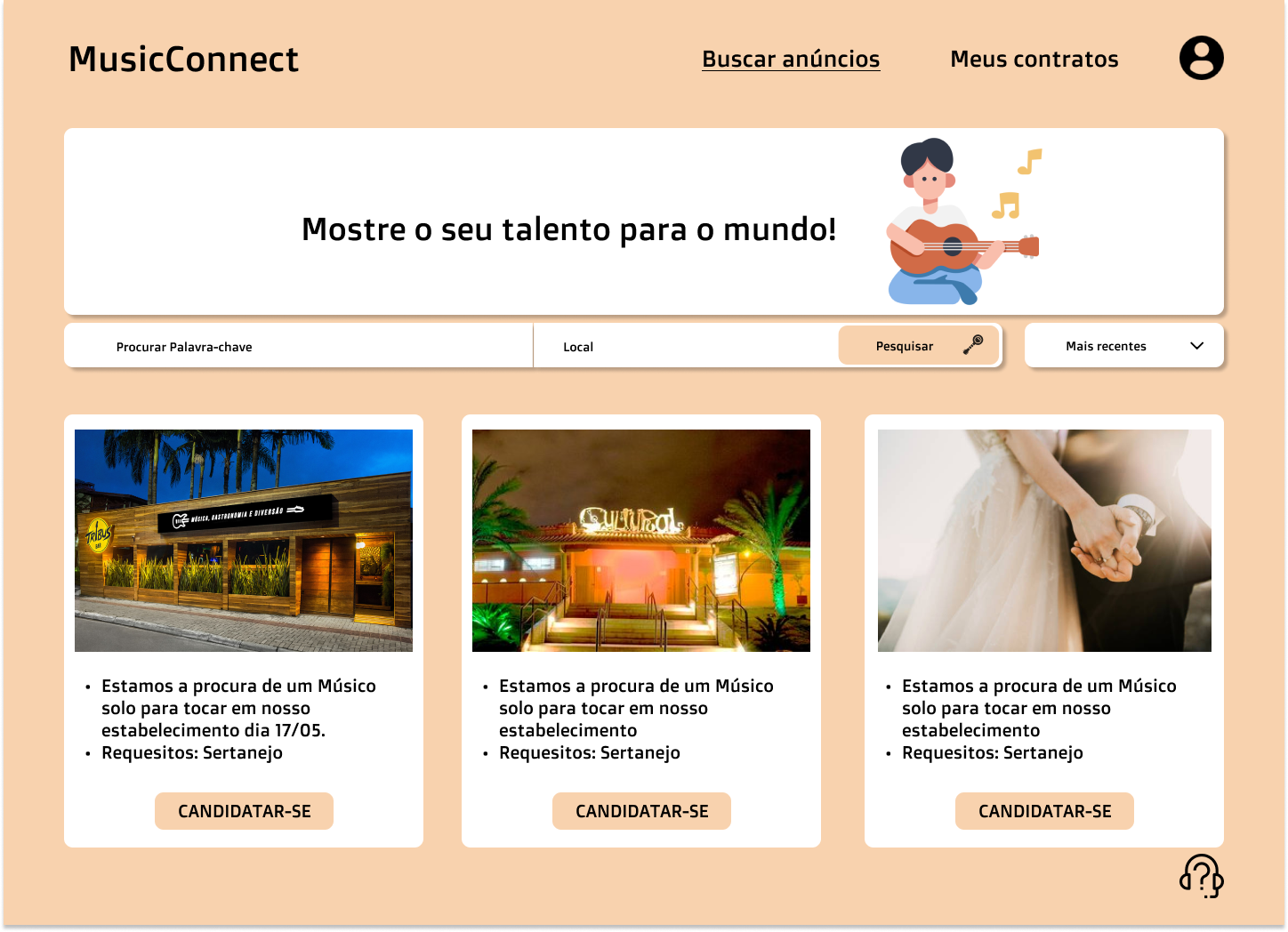
* Cadastro



* Objetivo



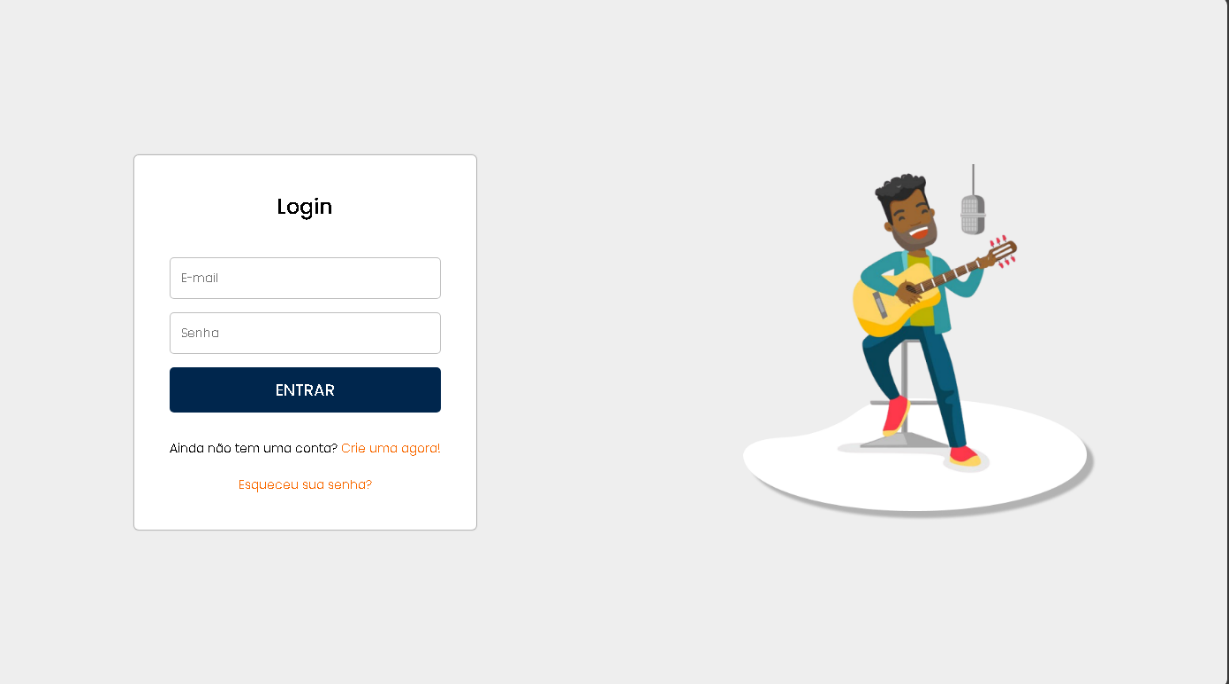
* Buscar Anúncios



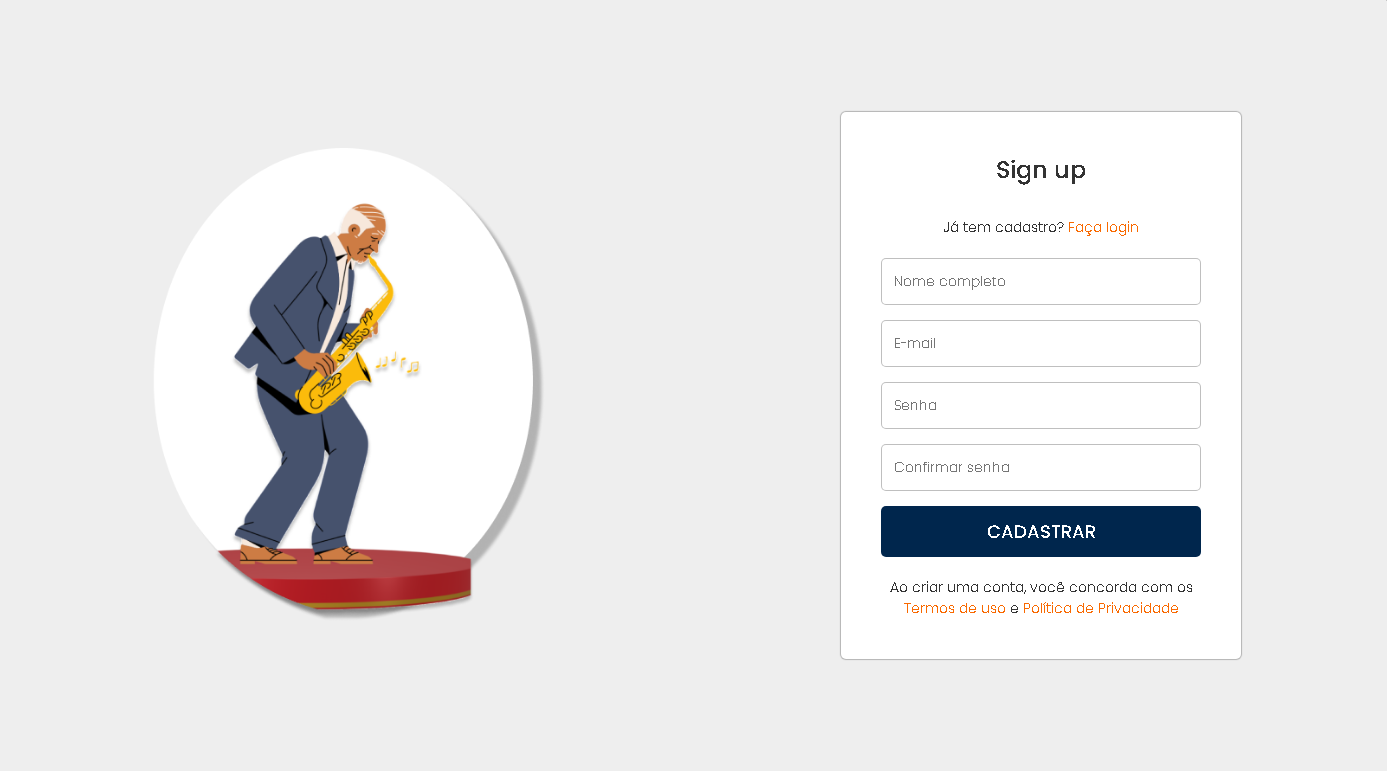
1. **Desenvolvimento Web**

Para o desenvolvimento das páginas web foi utilizado o padrão HTML, CSS e JavaScript. Levando em consideração os protótipos feitos no Figma, foi desenvolvido as primeiras páginas sendo a de login, cadastro, objetivo, home (músico e contratante), buscar anúncios e perfil.

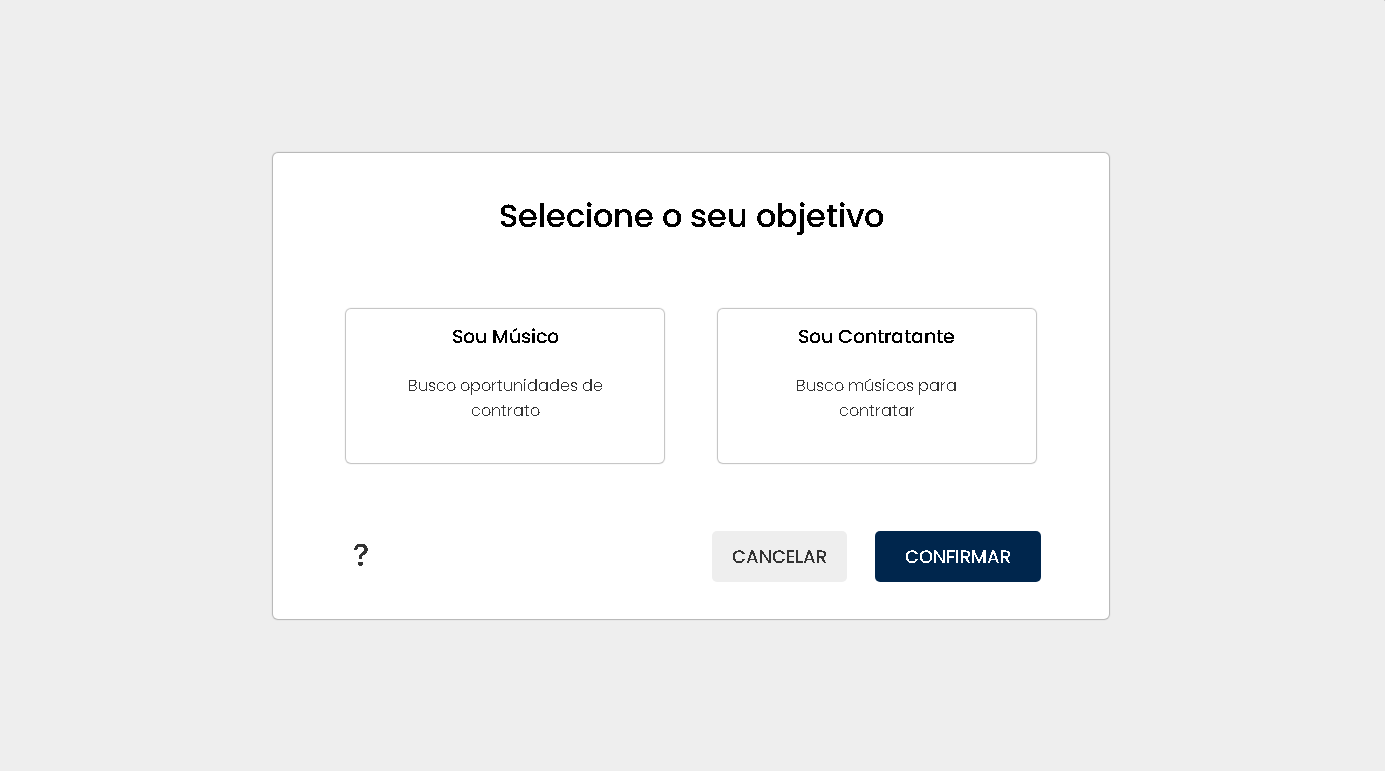
* Login



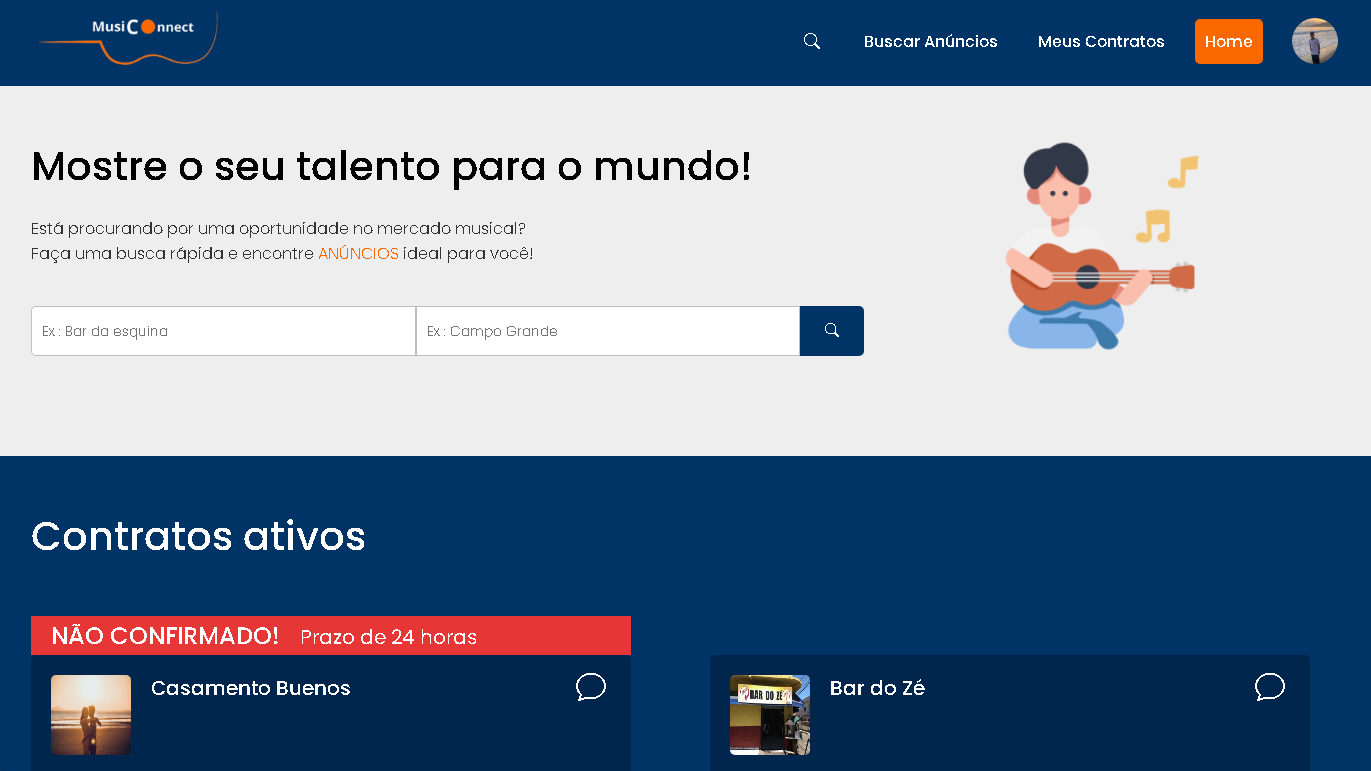
* Cadastro



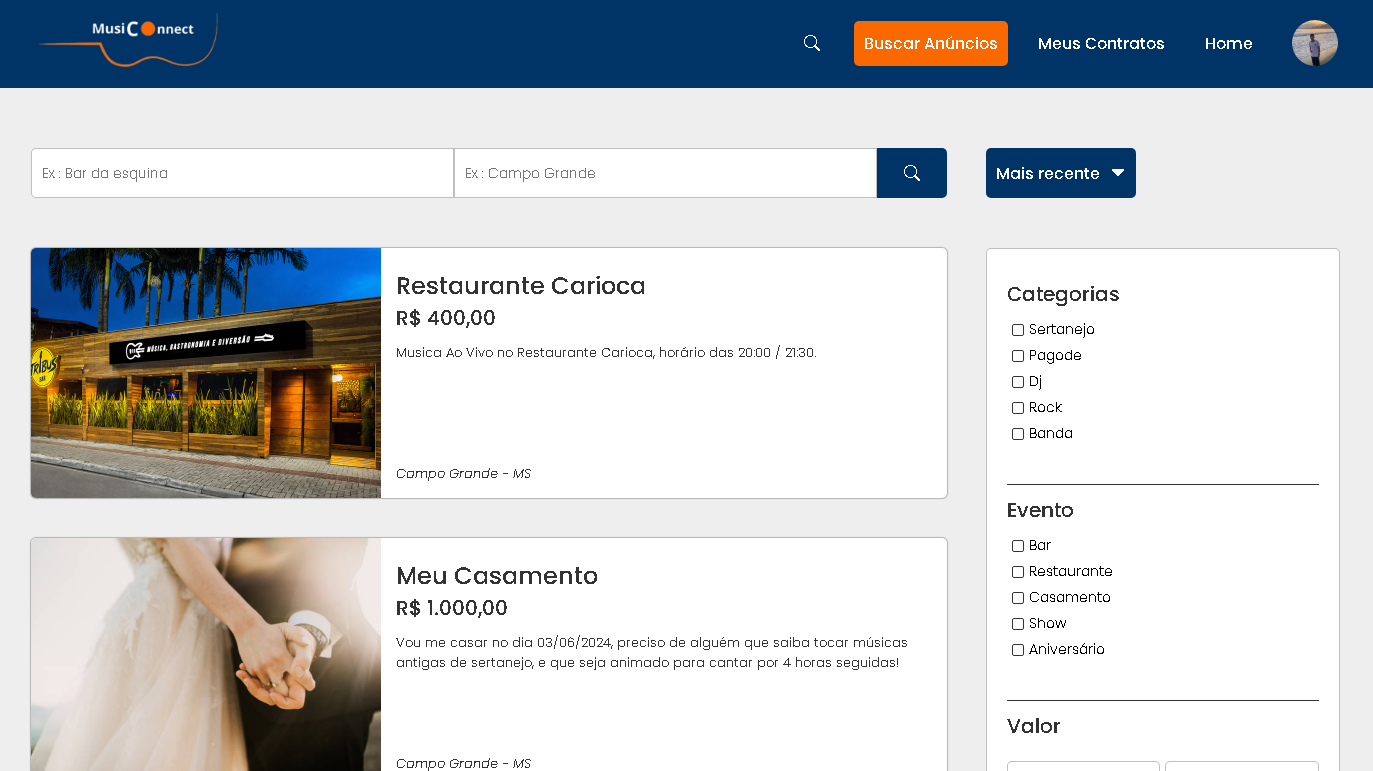
* Objetivo



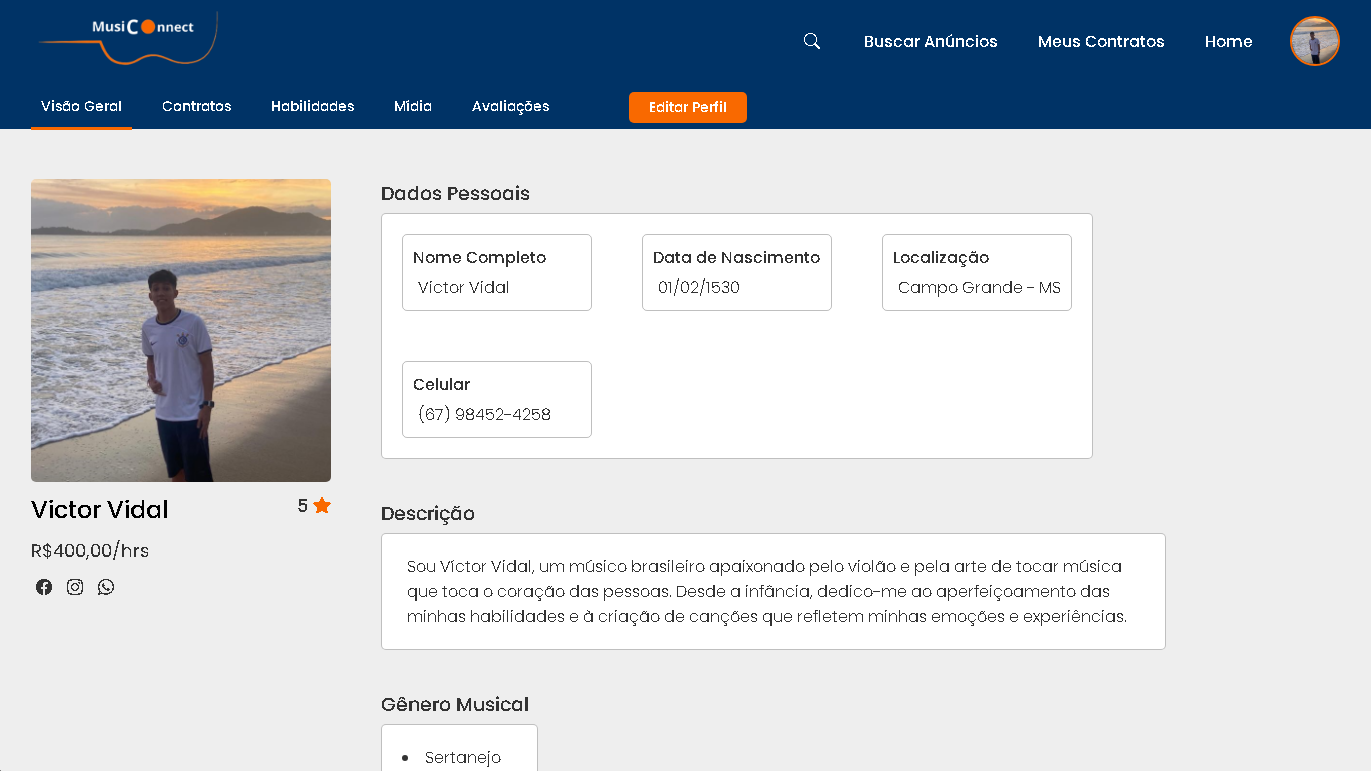
* Home Músico



* Buscar Anúncios



* Perfil Músico



1. **Banco de Dados**

Para armazenar, organizar e gerenciar informações de forma estruturada, permitir fácil acesso, consultas, e manipulação dos dados, optamos por utilizar o SQL Server para o desenvolvimento do nosso sistema devido a várias razões que atendem às necessidades do projeto. O SQL Server é conhecido por sua robustez e escalabilidade, o que é essencial para gerenciar grandes volumes de dados, como informações sobre os músicos, contratantes e contratos, de forma eficiente. Outro fator importante é que um dos integrantes do nosso grupo possui experiência prática com o SQL Server, o que contribui para uma curva de aprendizado mais rápida e a aplicação de melhores práticas no desenvolvimento.

**8.1 Banco de Dados Relacional**

O SQL Server usa um modelo de dados relacional para armazenar e gerenciar informações. Isso significa que ele organiza os dados em tabelas que estão relacionadas entre si por meio de chaves primárias e estrangeiras, permitindo consultas e manipulações complexas dos dados usando a linguagem SQL. SQL Server usa um método de armazenamento de dados que usa tabelas, linhas e colunas para organizar os dados. As colunas contêm os atributos dos dados, e as linhas contêm os valores.

1. **Modelo Conceitual**

O Modelo Conceitual é o modelo de mais alto nível, ou seja, que está mais próximo da realidade dos usuários. O nível conceitual é desenvolvido com alto nível de abstração, a partir dos requisitos do sistema, extraídos na fase de levantamento de requisitos.

**9.1 Modelo Entidade-Relacionamento (MER)**

O Modelo Entidade-Relacionamento (MER) é uma representação conceitual de um sistema de banco de dados. Ele foca na abstração das entidades, atributos e relacionamentos. O MER foi utilizado no projeto com o objetivo de capturar a lógica de negócios e requisitos sem entrar em detalhes técnicos de implementação.

**9.2 Identificando as Entidades**

Uma entidade é um objeto ou conceito do mundo real que possui existência independente e é importante para o sistema, como um cliente, produto ou funcionário.

Entidades:

* Músico
* Contratante
* Anúncio
* Contrato

**9.3 Identificando os Atributos**

São as características ou propriedades de uma entidade que fornecem detalhes sobre ela, como o nome de um cliente, o preço de um produto ou a data de nascimento de um funcionário.

* Atributos da Entidade Músico:
  + Nome
  + CPF
  + Data de Nascimento
  + Gênero
  + Endereço
    - Rua
    - Número
    - Bairro
    - CEP
    - Cidade
    - Estado
  + Telefone
  + E-mail
  + Senha
  + Habilidades
  + Gênero Musical
  + Redes Sociais
  + Data Cadastro
* Atributos da Entidade Contratante:
  + Nome
  + CPF
  + Data de Nascimento
  + Gênero
  + Endereço
    - Rua
    - Número
    - Bairro
    - CEP
    - Cidade
    - Estado
  + Telefone
  + E-mail
  + Senha
  + Redes Sociais
  + Data Cadastro
* Atributos da Entidade Anúncio:
  + Código de Anúncio
  + Nome Contratante
  + CPF Contratante
  + Data e hora do Anúncio
  + Data de Vencimento
  + Categoria
  + Tipo de Evento
  + Endereço do Anúncio
    - Rua
    - Número
    - Bairro
    - CEP
    - Cidade
    - Estado
  + Descrição
  + Valores
* Atributos da Entidade Contrato:
  + Código de Contrato
  + Nome Músico
  + CPF Músico
  + Nome Contratante
  + CPF Contratante
  + Data e horado Contrato
  + Data e hora de Atividade
  + Endereço de Atividade
    - Rua
    - Número
    - Bairro
    - CEP
    - Cidade
    - Estado
  + Descrição da Atividade
  + Valores
  + Penalidade
  + Status

**9.4 Identificando os Relacionamentos**

Representam as associações ou interações entre duas ou mais entidades. Por exemplo, um cliente faz um pedido, ou um funcionário trabalha em um departamento.

Relacionamentos:

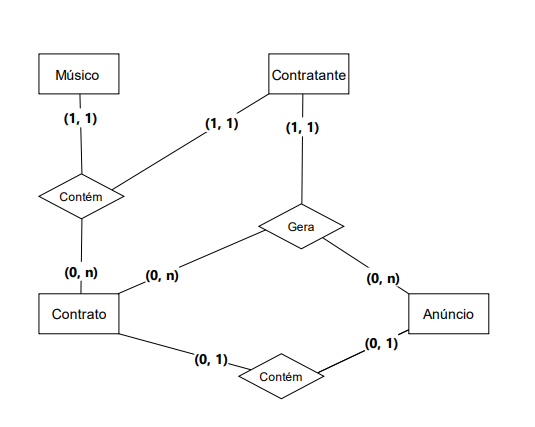
* O Músico pode ou não conter vários Contratos.
* O Contratante pode ou não conter vários Contratos.
* O Contratante pode gerar nenhum ou muitos Contratos.
* O Contratante pode gerar nenhum ou muitos Anúncios.
* Um Anúncio pode ou não conter um Contrato.

**9.5 Diagrama Entidade-Relacionamento (DER)**

O DER é a representação gráfica do MER, para facilitar o entendimento e visualização dos elementos utilizados, sendo eles:

* Retângulos para Entidades.
* Losangos para Relacionamentos.

Segue Diagrama Entidade-Relacionamento:

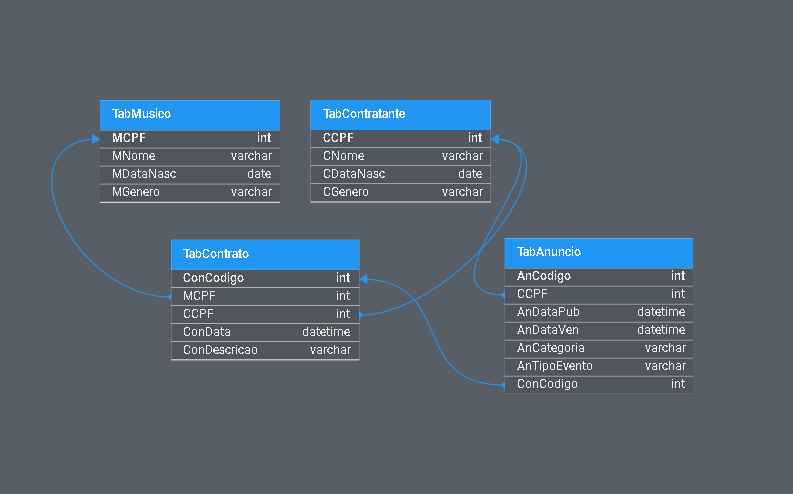


**10. Modelo Lógico**

O Modelo Lógico descreve como os dados serão armazenados no banco e também seus relacionamentos. Esse modelo adota alguma tecnologia, pode ser: relacional, orientado a objetos, orientado a colunas, entre outros.

**10.1 Modelo Relacional**

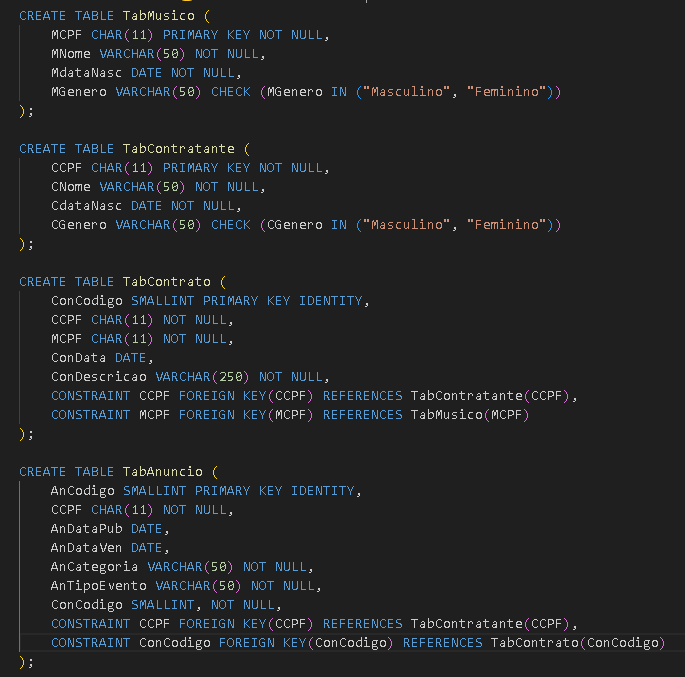
O Modelo Relacional é um modelo utilizado em SGBDS, baseado em tabelas (relações), atributos, chaves primárias, chaves estrangeiras, etc.



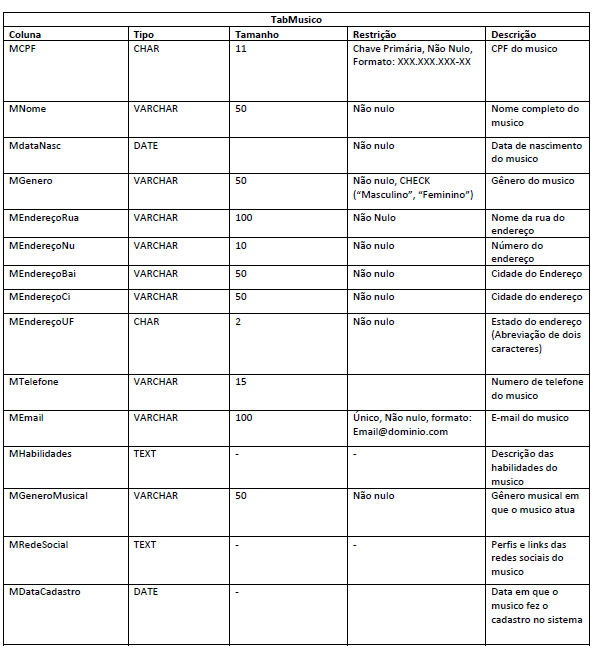
**11. Modelo Físico**

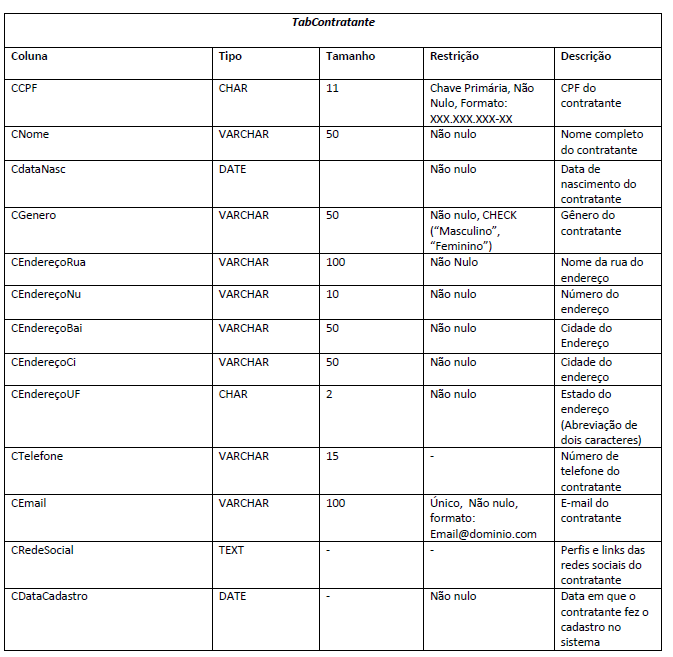
O Modelo Lógico descreve, por meio de alguma linguagem, como será feita a armazenagem no banco. Nesse nível se escolhe qual Sistema gerenciador de Banco de dados (SGBD) será usado, levando em consideração o modelo lógico adotado. Pode ser: PostgreSQL, MySQL, dentre outros.

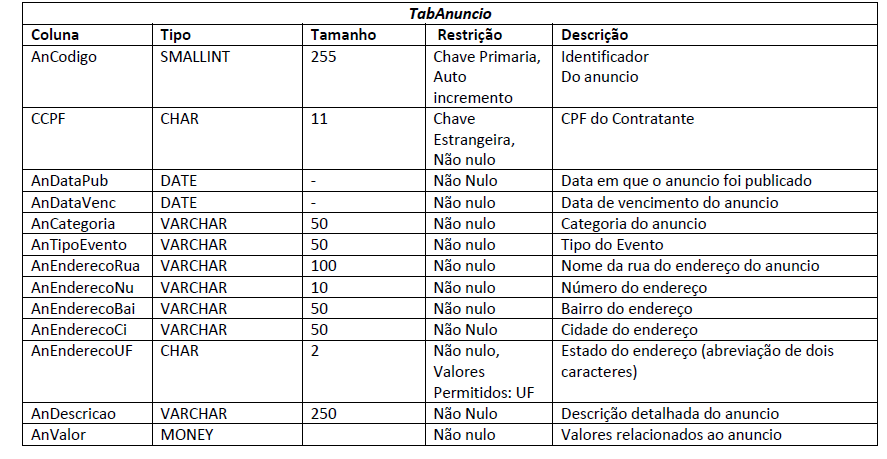
SGBD SQL Server:

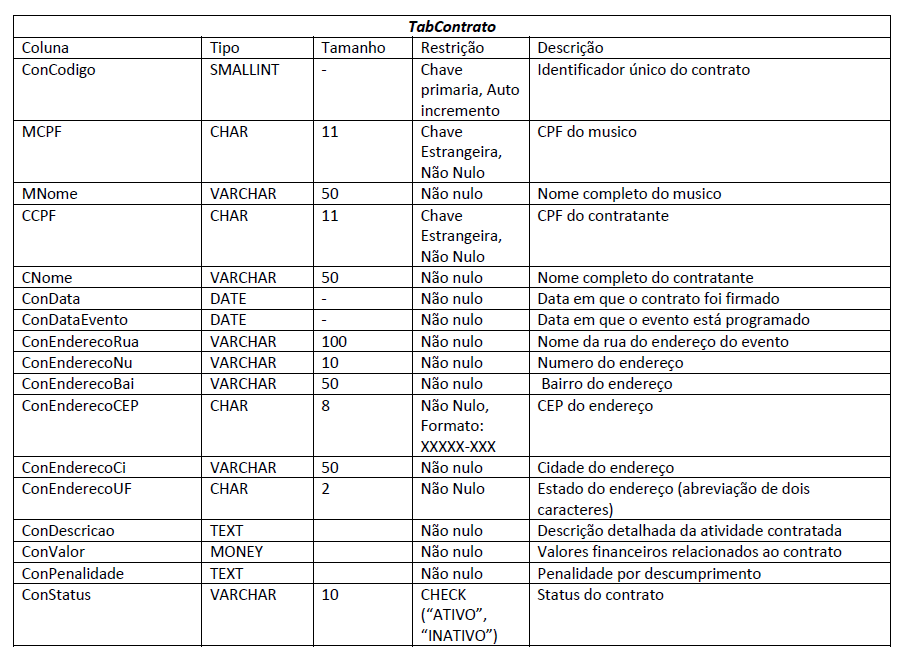


**12. Dicionário de Dados**

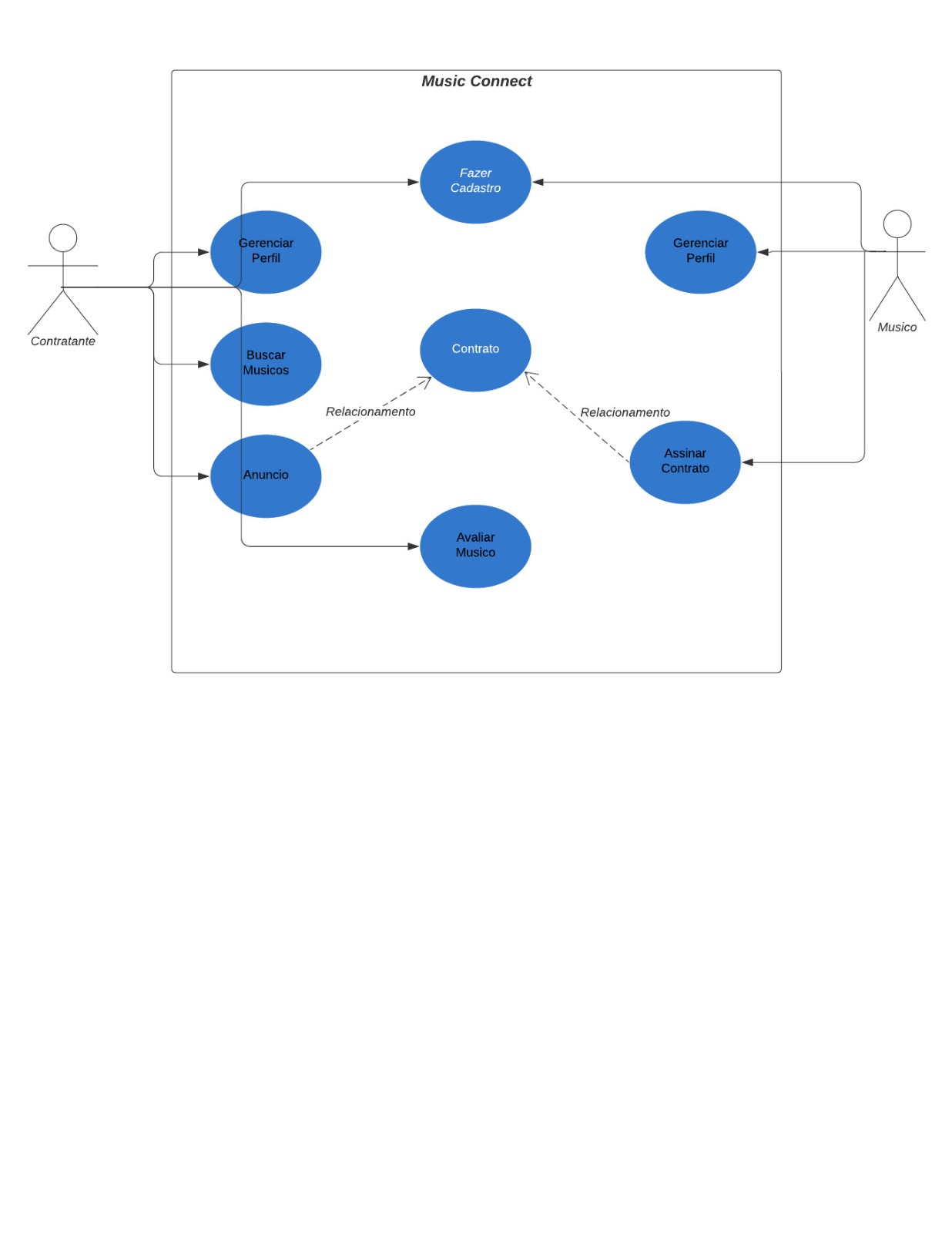








**13. Caso de Uso**



**14. Referências**

 ORACLE. Conceitos Básicos de Banco de Dados. Disponível em: <https://www.oracle.com/database/what-is-database/>. Acesso em: 28 ago. 2024.

 IBM. O que é Banco de Dados Relacional. Disponível em: <https://www.ibm.com/br-pt/topics/what-is-a-relational-database>. Acesso em: 28 ago. 2024.

 MICROSOFT. Modelagem de Dados no SQL Server. Disponível em: <https://docs.microsoft.com/pt-br/sql/ssdt/how-to-design-database-schema?view=sql-server-ver15>. Acesso em: 28 ago. 2024.

 MICROSOFT. Databases (Banco de dados). Disponível em: <https://learn.microsoft.com/pt-br/sql/relational-databases/databases/databases?view=sql-server-ver16>. Acesso em: 28 ago. 2024.

 MICROSOFT. O que é um banco de dados relacional? Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-a-relational-database#:~:text=Exemplos%20de%20banco%20de%20dados,%2C%20MySQL%2C%20PostgreSQL%20e%20MariaDB>. Acesso em: 28 ago. 2024.

 ALURA. SQL e NoSQL: Entenda a diferença entre bancos de dados relacionais e não relacionais. Disponível em: <https://www.alura.com.br/artigos/sql-nosql-bancos-relacionais-nao-relacionais?srsltid=AfmBOooURNbGG_3_vIGd6H2sebdbHhx4Jkim88A0s0j3xS5KjXYpoHr5>. Acesso em: 28 ago. 2024.

 DBA ONBOARDING. Como é a estrutura de um banco de dados SQL Server (bônus de arquitetura de ambiente). Disponível em: <https://www.dbaonboarding.com.br/post/como-%C3%A9-a-estrutura-de-um-banco-de-dados-sql-server-b%C3%B4nus-de-arquitetura-de-ambiente>. Acesso em: 28 ago. 2024.

 BALTA.IO. Começando com SQL Server na prática: estrutura, inserção e consulta. Disponível em: <https://balta.io/blog/comecando-com-sql-server-na-pratica-estrutura-insercao-e-consulta>. Acesso em: 28 ago. 2024.

 IMPACTA. Entenda de uma vez por todas o banco de dados SQL Server. Disponível em: <https://www.impacta.com.br/blog/entenda-de-uma-vez-por-todas-o-banco-de-dados-sql-server/>. Acesso em: 28 ago. 2024.

 MESTRES DA WEB. Requisitos funcionais e não funcionais: o que são? Disponível em: <https://www.mestresdaweb.com.br/tecnologias/requisitos-funcionais-e-nao-funcionais-o-que-sao>. Acesso em: 28 ago. 2024.

 MOZILLA. Guia de HTML. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 28 ago. 2024.

 W3SCHOOLS. Guia de CSS. Disponível em: <https://www.w3schools.com/css/>. Acesso em: 28 ago. 2024.

 JAVASCRIPT.INFO. Guia de JavaScript. Disponível em: <https://javascript.info/>. Acesso em: 28 ago. 2024.

 PHP MANUAL. Documentação Oficial. Disponível em: <https://www.php.net/manual/pt_BR/>. Acesso em: 28 ago. 2024.

 FIGMA. O que é Figma e como usar. Disponível em: <https://www.figma.com/resources/learn-design/>. Acesso em: 28 ago. 2024.

 GITHUB. Introdução ao GitHub. Disponível em: <https://docs.github.com/pt/get-started/quickstart>. Acesso em: 28 ago. 2024.

 GIT. Documentação Oficial. Disponível em: <https://git-scm.com/doc>. Acesso em: 28 ago. 2024.

 MICROSOFT. SQL Server - Documentação Oficial. Disponível em: <https://docs.microsoft.com/pt-br/sql/sql-server/?view=sql-server-ver15>. Acesso em: 28 ago. 2024.

 SMASHING MAGAZINE. Guia de Prototipagem. Disponível em: <https://www.smashingmagazine.com/2017/01/what-prototyping/>. Acesso em: 28 ago. 2024.

 UX DESIGN. O que é Prototipagem? Disponível em: <https://uxdesign.cc/what-is-prototyping-1e9dbb9f4e7e>. Acesso em: 28 ago. 2024.