Uma imagem contendo comida, quarto

O conteúdo gerado por IA pode estar incorreto.

UNIVERSIDADE FEDERAL DA PARAIBA

CENTRO DE INFORMÁTICA

FELIPE GONTIJO SIQUEIRA

DOCUMENTO DE INTERFACES DE USUÁRIO

João Pessoa

2025

Sumário

[**1 - Introdução** 2](#_Toc196942015)

[**2 - Glossário de Termos** 4](#_Toc196942016)

[**3 - Fluxo de Interface do Usuário** 5](#_Toc196942017)

[**4 - Componentes de Interface** 5](#_Toc196942018)

[**4.3 - Página de Registro** 11](#_Toc196942019)

[**4.3.1 - Step 1 do Registro** 12](#_Toc196942020)

[**4.3.2 - Step 2 do Registro** 15](#_Toc196942021)

[**4.3.3 - Step 3 do Registro** 19](#_Toc196942022)

[**4.4 - Home Page** 23](#_Toc196942023)

[**4.4.1 - LEFTSIDEBAR** 24](#_Toc196942024)

[**4.4.2 - FEED** 27](#_Toc196942025)

[**4.4.3 - Right Sidebar** 30](#_Toc196942026)

[**4.4.4 - Header** 34](#_Toc196942027)

[**4.5 - Profile Page** 39](#_Toc196942028)

[**4.5.1 - Perfil** 42](#_Toc196942029)

[**4.5.2 - SIDEBAR (PÁGINA DE PERFIL)** 46](#_Toc196942030)

[**4.6 - Modais** 50](#_Toc196942031)

[**4.6.1 - Modal de Eventos do Calendário** 50](#_Toc196942032)

[**4.6.2 - Modal de Contratações** 52](#_Toc196942033)

[**4.6.3 - New Post Modal** 54](#_Toc196942034)

[**4.6.4 - Chatmodal** 56](#_Toc196942035)

[**4.6.5 – PostsSalvosModal** 59](#_Toc196942036)

[**4.6.6 - SeguidoresModal** 62](#_Toc196942037)

[**Padrões de Design** 65](#_Toc196942038)

[**MOTIVAÇÃO DO DESIGN** 66](#_Toc196942039)

[**Referências Técnicas** 67](#_Toc196942040)

# **1 - Introdução**

Este documento detalha a interface do usuário da aplicação SoundBuddy, uma plataforma focada em música e conexões sociais. A documentação segue o fluxo de execução da aplicação, começando pela Landing Page e progredindo através das telas de Login ou Registro, Home, Profile. Este documento serve como referência abrangente para desenvolvedores, designers e stakeholders da plataforma.

A organização do código da aplicação segue uma estrutura modular e bem definida para facilitar o desenvolvimento, manutenção e escalabilidade. Dentro da pasta **app**, a estrutura se divide em subpastas específicas para cada funcionalidade ou seção da aplicação. Por exemplo, na raiz da pasta **app** encontra-se o arquivo **page.tsx**, que implementa a Landing Page, além do arquivo **global.css**, que inclui a implementação personalizada da barra de rolagem.

Dentro da subpasta **components** há componentes compartilhados, como o **ChatModal**, que podem ser utilizados em diferentes partes da aplicação. A pasta **Home** contém sua própria subpasta **components**, que abriga os componentes relacionados à página inicial:

* **Feed**: Com arquivo *Index.tsx* para renderizar o feed principal e o *NewPostModal* para criação de posts;
* **Header**: Com *index.tsx* para a barra de busca e navegação superior;
* **LeftSidebar**: Inclui arquivos como *ContrataçõesModal.tsx*, *EventosCalendario.tsx* e *index.tsx* para gerenciar a navegação e exibição de modais relacionados a eventos e contratações;
* **RightSidebar**: Possui *index.tsx* para exibir as conversas e interações laterais;
* Além disso, a própria pasta **Home** possui um *page.tsx* que reúne e estrutura esses componentes para formar a Home Page.

Da mesma forma, a pasta **Login** contém um *page.tsx* dedicado à tela de autenticação, enquanto a pasta **Profile** possui uma subpasta **components** com os arquivos *perfil.tsx* e *sidebar.tsx* para gerenciar, respectivamente, a visualização do perfil principal e a barra lateral com informações rápidas. O arquivo *page.tsx* dentro da pasta **Profile** integra esses componentes para compor a página de perfil do usuário.

Por fim, a pasta **Register** está organizada com uma subpasta **components** que contém os arquivos *Step1.tsx*, *Step2.tsx* e *Step3.tsx*, representando as diferentes etapas do processo de registro, e um *page.tsx* na raiz da pasta que orquestra o fluxo completo de cadastro.

# **2 - Glossário de Termos**

|  |  |
| --- | --- |
| **Termo** | Explicação |
| **Autenticação** | Validação das credenciais (email/senha) para acesso à plataforma. Implementada via handleSubmit no Login. |
| **Botão de Ação** | Elemento interativo (ex: "Login") com estilização responsiva (Tailwind) e efeitos de hover (hover:scale-105). |
| **Client-side** | Componentes renderizados no navegador via JavaScript (ex: Login, Home, Feed). |
| **Componente Funcional** | Bloco reutilizável de UI em React (ex: LeftSidebar), usando hooks como useState. |
| **Design System** | Padrões visuais da aplicação: cores (preto/branco), tipografia responsiva, grids flexíveis. |
| **Feed** | Área central da Home que exibe posts, mídia e permite interações (salvar, ampliar, chat). |
| **Fluxo de Interface** | Sequência: Landing Page → Login/Registro → Home → Perfil. |
| **Hook** | Função do React para estados/efeitos. Ex: useState (dados locais), useEffect (side effects). |
| **Landing Page** | Página inicial para usuários não autenticados, com botões de acesso e slogan. |
| **Layout Responsivo** | Adaptação dinâmica a telas (ex: sidebars recolhem em <1200px, grids flex-col → flex-row). |
| **Modal** | Janela sobreposta para ações secundárias (ex: recuperar senha, adicionar evento). |
| **Next.js** | Framework React para roteamento (ex: useRouter) e estrutura de pastas (page.tsx). |
| **Padrão Mobile-First** | Desenvolvimento priorizando mobile, com ajustes progressivos para desktop (ex: w-full → md:w-auto). |
| **Router** | Gerenciador de navegação do Next.js. Ex: router.push('/login') para redirecionar. |
| **Responsividade** | Adaptação a dispositivos via Tailwind (ex: md:text-6xl) e estados (ex: isCollapsed). |
| **State (Estado)** | Dados dinâmicos que alteram a UI. Ex: showForgotPassword (controla modais). |
| **Tailwind CSS** | Framework CSS para estilização rápida. Classes como rounded-full, bg-gray-900. |
| **UI Components** | Elementos reutilizáveis: Header (busca), Card de Post (exibição de conteúdo), Botão Flutuante. |
| **User Flow** | Caminho do usuário. Ex: Novo usuário → Registro (3 etapas) → Home. |
| **Validação de Formulário** | Verificação de campos obrigatórios (HTML required) e lógica customizada (ex: seleção de tags). |

# **3 - Fluxo de Interface do Usuário**

O fluxo principal da aplicação SoundBuddy segue esta sequência:

1. **Landing Page**: Página inicial com opções para Login ou Registro
2. **Login**: Autenticação de usuários existentes
3. **Register**: Processo de criação de conta em múltiplas etapas
4. **Home**: Feed principal com conteúdo personalizado
5. **Profile**: Visualização e edição de perfil do usuário

# **4 - Componentes de Interface**

**4.1 - Landing**

A página Landing é a primeira tela que os usuários visualizam ao acessar a aplicação SoundBuddy. Apresenta opções para fazer login ou criar uma nova conta, direcionando os usuários para o fluxo apropriado.

**Componente**: Landing  
**Arquivo**: app/page.tsx

**Funcionalidades:**

1. **Navegação para Login**
   * Função: handleLogin()
   * Redireciona o usuário para a página de login (/Login)
2. **Navegação para Registro**
   * Função: handleRegister()
   * Redireciona o usuário para a página de registro (/Register)

**Elementos de UI:**

1. **Container Principal**
   * Classe: min-h-screen bg-conic-180 from-zinc-900 via-gray-700 to-zinc-900 flex items-center justify-center p-4
   * Ocupa toda a altura da tela
   * Fundo gradiente cônico escuro (preto-cinza-preto)
   * Conteúdo centralizado vertical e horizontalmente
   * Preenchimento interno (padding) em todas as direções
2. **Efeito de Fundo**
   * Classe: absolute inset-0 bg-zinc-900/10 backdrop-blur-2xl
   * Preenche toda a área da tela
   * Adiciona uma camada semi-transparente sobre o fundo
   * Efeito de desfoque intenso para melhorar a aparência visual
3. **Container do Conteúdo**
   * Classe: text-center space-y-6 md:space-y-8 max-w-4xl w-full z-10
   * Alinhamento de texto centralizado
   * Espaçamento vertical responsivo entre elementos filhos
   * Largura máxima limitada (max-w-4xl) para melhor legibilidade
   * z-index elevado para garantir visibilidade sobre o fundo
   * Contém:
     + **Seção de Título e Tagline**
       - Classe: space-y-3 md:space-y-4
       - Espaçamento vertical responsivo entre elementos
       - Contém:
         * **Título Principal**: "Soundbuddy" (h1)

Classe: text-4xl sm:text-5xl md:text-6xl font-bold text-slate-200

Tamanho de texto adaptável para diferentes dispositivos

Cor cinza claro (slate-200)

Peso da fonte em negrito

* + - * + **Slogan da Aplicação** (p)

Classe: text-gray-400 text-base md:text-xl px-4

Texto: "Encontre seu som"

Cor cinza médio

Tamanho de texto responsivo

Preenchimento horizontal para melhor espaçamento

* + - **Seção de Botões de Ação**
      * Classe: flex flex-col md:flex-row gap-3 sm:gap-4 justify-center items-center w-full px-4
      * Layout em coluna em dispositivos móveis, linha em telas médias
      * Espaçamento adaptável entre botões
      * Centralização de conteúdo
      * Largura total com preenchimento horizontal
      * Contém:
        + **Botão de Login** (Principal)

Classe: bg-zinc-700/50 border-2 border-white/50 text-white w-full md:w-auto px-6 py-3 md:px-8 md:py-4 rounded-full hover:bg-gray-800 transition-all duration-300 transform hover:scale-105 shadow-xl font-semibold text-base sm:text-lg

Fundo cinza escuro semi-transparente

Borda branca semi-transparente

Texto branco com fonte semi-negrito

Largura total em dispositivos móveis, automática em telas médias

Preenchimento responsivo

Bordas totalmente arredondadas (formato de pílula)

Efeito hover: escurecimento do fundo e leve aumento de escala

Sombra pronunciada para efeito de elevação

* + - * + **Botão de Registro** (Secundário)

Classe: bg-zinc-700/50 border-2 border-white/50 text-white w-full md:w-auto px-6 py-3 md:px-8 md:py-4 rounded-full hover:bg-gray-700 transition-all duration-300 transform hover:scale-105 shadow-xl font-semibold text-base sm:text-lg

Estilo visual similar ao botão de login

Efeito hover levemente diferente (cinza mais claro que o botão principal)

**Notas Adicionais:**

1. **Acessibilidade**
   * Botões incluem atributos aria-label para descrever suas funções
   * Estrutura semântica com uso apropriado de tags h1 e p
2. **Responsividade**
   * Layout adapta-se entre dispositivos móveis e telas maiores
   * Botões empilhados em dispositivos móveis, lado a lado em telas médias
   * Tamanhos de fonte e espaçamentos responsivos

**4.2 - Login**

A página de Login permite que usuários existentes se autentiquem na aplicação, fornecendo seu email e senha. Também inclui funcionalidade para recuperação de senha.

**Componente**: Login  
**Arquivo**: app/Login/page.tsx

**Estados:**

1. **email** - Armazena o valor do campo de email
2. **password** - Armazena o valor do campo de senha
3. **showForgotPassword** - Controla a exibição do modal de recuperação de senha
4. **recoveryEmail** - Armazena o email para recuperação de senha
5. **isRecoverySubmitted** - Controla o estado de envio do email de recuperação
6. **authError** - Armazena mensagens de erro de autenticação
7. **touched** - Controla quais campos foram tocados para validação

**Elementos de UI:**

1. **Container Principal**
   * Classe: min-h-screen flex flex-col items-center justify-center bg-gradient-to-br from-zinc-900 via-gray-700 to-zinc-800 p-4
   * Ocupa toda a altura da tela
   * Fundo gradiente de preto a cinza escuro
   * Conteúdo centralizado vertical e horizontalmente
   * Layout em coluna
2. **Efeito de Fundo**
   * Classe: absolute inset-0 bg-zinc-900/10 backdrop-blur-3xl
   * Efeito de desfoque para melhorar a aparência do gradiente
3. **Título Principal**
   * Classe: text-3xl md:text-5xl font-bold text-white md:text-zinc-100 mb-6 md:mb-10 z-10
   * Texto "Soundbuddy" em tamanho responsivo
   * Cor branca com variações sutis entre dispositivos móveis e telas médias
   * Margem inferior responsiva
   * z-index para garantir visibilidade sobre o fundo
4. **Container do Formulário de Login**
   * Classe: w-full md:w-100 p-6 bg-black/30 shadow-2xl shadow-white/20 rounded-xl md:rounded-3xl z-10
   * Largura total em dispositivos móveis, fixa em telas médias
   * Fundo preto semitransparente com sombra branca sutil
   * Bordas arredondadas, mais pronunciadas em telas médias
   * Contém:
     + **Título do Formulário**: "Login" (h2)
       - Classe: text-xl md:text-2xl text-white font-bold mb-4 text-center
       - Texto centralizado, tamanho responsivo
     + **Mensagem de Erro** (condicional)
       - Classe: mb-4 p-3 border border-red-500 bg-red-500/10 text-red-500 rounded-lg text-sm
       - Exibida quando há erro de autenticação
     + **Campo de Email**
       - Label e input com placeholder "Digite seu email"
       - Classe do input: input\_register w-full md:w-auto
       - Validação visual com borda vermelha quando inválido
       - Mensagem de erro condicional
     + **Campo de Senha**
       - Label e input tipo password com placeholder "Digite sua senha"
       - Classe do input: input\_register w-full md:w-auto
       - Validação visual com borda vermelha quando inválido
       - Mensagem de erro condicional
     + **Link de Recuperação de Senha**
       - Botão tipo text: "Esqueci minha senha"
       - Classe: text-sm text-gray-400 hover:text-white transition-colors
     + **Botão de Envio**
       - Classe: w-full md:w-1/2 border-2 border-white text-white p-2 rounded-full hover:bg-white hover:text-black transition-colors duration-300 font-semibold text-sm md:text-base
       - Bordas brancas com texto branco
       - Em hover, fundo branco com texto preto
       - Largura total em dispositivos móveis, metade em telas médias
5. **Texto Rodapé**
   * Classe: mt-6 md:mt-8 text-lg md:text-xl text-white md:text-white font-medium z-10
   * Texto: "IT'S ALL ABOUT MUSIC"
   * Cor branca consistente em todos os dispositivos
   * Tamanho responsivo
   * z-index para garantir visibilidade
6. **Modal de Recuperação de Senha**
   * Exibido condicionalmente baseado no estado showForgotPassword
   * Classe do overlay: fixed inset-0 bg-black/50 backdrop-blur flex items-center justify-center z-50 p-4
   * Classe do container: bg-black p-6 rounded-xl w-full max-w-md
   * Contém:
     + **Título do Modal**
       - Texto dinâmico baseado no estado isRecoverySubmitted
     + **Formulário de Recuperação** (exibido quando isRecoverySubmitted é falso)
       - Texto explicativo sobre o processo
       - Campo de email para recuperação
       - Botões para cancelar ou enviar solicitação
     + **Mensagem de Confirmação** (exibida quando isRecoverySubmitted é verdadeiro)
       - Texto de confirmação em verde
       - Botão para fechar o modal

**Funcionalidades:**

1. **Validação de Campos**
   * Função: handleChange(field, value)
     + Atualiza o valor do campo correspondente
     + Marca o campo como tocado
     + Limpa mensagens de erro existentes
   * Função: handleBlur(field)
     + Marca o campo como tocado para validação
2. **Autenticação de Usuário**
   * Função: handleSubmit(e)
   * Previne o comportamento padrão do formulário
   * Marca todos os campos como tocados para validação
   * Simulação de erro para email específico (test@example.com)
   * Implementação atual navega para a Home como simulação
3. **Recuperação de Senha**
   * Função: handlePasswordRecovery(e)
   * Previne o comportamento padrão do formulário
   * Atualiza o estado isRecoverySubmitted
   * Simula o fechamento do modal após 3 segundos

## **4.3 - Página de Registro**

A página de Registro é uma interface multi-etapas que permite aos novos usuários criar uma conta na plataforma SoundBuddy. A interface implementa um fluxo de cadastro em três etapas, guiando o usuário desde o fornecimento de informações básicas até a finalização do perfil.

**Componente**: Register  
**Arquivo**: app/components/Register/page.tsx

**Estados:**

1. **currentStep** - Controla qual etapa do formulário está sendo exibida (1, 2 ou 3)
2. **formData** - Armazena todos os dados do formulário em todas as etapas, incluindo:
   * username
   * email
   * password
   * confirmPassword
   * userType
   * about
   * tags
   * profileImage

**Elementos de UI:**

1. **Container Principal**
   * Classe: min-h-screen
   * Garante que o conteúdo ocupe pelo menos a altura total da tela
2. **Renderização Condicional de Etapas**
   * Cada etapa é exibida com base no valor do estado currentStep
   * Apenas uma etapa é visível por vez
   * Componentes de etapa recebem props consistentes:
     + **formData** - Estado atual do formulário
     + **setFormData** - Função para atualizar o estado
     + **handleNextStep/handlePreviousStep** - Funções para navegação entre etapas
     + **handleSubmit** - Função para finalizar o cadastro (apenas na última etapa)

**Funcionalidades:**

1. **Navegação entre Etapas**
   * Função: handleNextStep() - Incrementa o estado currentStep se menor que 3
   * Função: handlePreviousStep() - Decrementa o estado currentStep se maior que 1
2. **Gestão de Estado Centralizada**
   * Mantém todos os dados do formulário em um único objeto de estado
   * Compartilha o mesmo objeto de estado entre todas as etapas
3. **Finalização do Cadastro**
   * Função: handleSubmit() - Processa o envio do formulário completo
   * Redireciona o usuário para a Home após o cadastro bem-sucedido
   * Note: Contém TODOs para implementação da lógica real de persistência de dados

### **4.3.1 - Step 1 do Registro**

A primeira etapa do formulário de cadastro multi-etapas que permite ao usuário inserir informações básicas (username, email, senha) e escolher o tipo de conta desejado (Artista, Cliente ou ambos).

**Componente**: Step1  
**Arquivo**: Register/components/Step1.tsx

**Props:**

1. **formData** - Estado atual do formulário
2. **setFormData** - Função para atualizar o estado do formulário
3. **handleNextStep** - Função para avançar para o próximo passo

**Elementos de UI:**

1. **Container Principal**
   * Classe: h-screen flex overflow-hidden
   * Layout flexível que ocupa toda a altura da tela
   * Oculta overflow para evitar rolagem não intencional
2. **Painel Esquerdo (Formulário)**
   * Classe: w-full md:w-1/2 bg-black p-12 flex flex-col justify-between items-center
   * Ocupa toda a largura em dispositivos móveis e metade em telas médias ou maiores
   * Fundo preto com padding interno de 3rem (p-12)
   * Layout flexível em coluna com espaçamento entre elementos e alinhamento central
   * Contém:

a. **Título do Formulário**

* + - Classe: text-3xl font-bold text-white text-center mb-8
    - Texto grande e centralizado com espaçamento inferior

b. **Campos de Entrada de Dados**

* + - Grupo com classe: space-y-4 (espaçamento vertical entre campos)
    - Quatro campos de entrada:
      * Username (input text)
      * Email (input email)
      * Senha (input password)
      * Confirmar Senha (input password)
    - Cada campo utiliza a classe personalizada input\_register
    - Eventos onChange atualizam o estado formData correspondente

c. **Seleção de Tipo de Usuário**

* + - Título da seção: text-lg font-semibold text-center text-white mb-4
    - Grid com 3 colunas: grid grid-cols-3 gap-2
    - Opções de tipo de usuário apresentadas como radio buttons customizados:
      * Container de cada radio: flex flex-col items-center cursor-pointer
      * Indicador visual circular: w-6 h-6 rounded-full border-2
      * Estilização condicional baseada no estado selecionado
      * Rótulo textual abaixo do indicador: text-white text-sm font-medium

d. **Botão de Avançar**

* + - Classe: mt-6 w-full border-2 border-white rounded-full py-3 text-white hover:bg-white hover:text-black transition-colors flex items-center justify-center gap-2 text-sm
    - Ocupa toda a largura do container
    - Estilo de botão outline com hover para background branco
    - Inclui ícone de seta para direita (ArrowRightIcon)

1. **Painel Direito (Informações)**
   * Classe: hidden md:flex w-1/2 bg-white p-12 flex-col justify-center items-center
   * Visível apenas em telas médias ou maiores (md:flex)
   * Fundo branco com padding interno de 3rem
   * Layout flexível em coluna com alinhamento central
   * Contém:

a. **Título da Seção**

* + - Classe: text-2xl font-bold text-black mb-6

b. **Descrições dos Tipos de Usuário**

* + - Cada tipo apresentado em um bloco: space-y-1 mb-6
    - Título de cada tipo: text-lg font-semibold text-gray-900
    - Descrição detalhada: text-gray-600 text-sm leading-relaxed

c. **Nota Informativa**

* + - Separada por uma borda superior: pt-6 border-t border-gray-200
    - Texto pequeno e discreto: text-xs text-gray-500

**Constantes e Dados:**

1. **userTypes** - Array de objetos definindo os três tipos de usuário disponíveis:
   * Artista - Para músicos, produtores ou técnicos que oferecem serviços
   * Both - Para usuários que desejam contratar e oferecer serviços simultaneamente
   * Cliente - Para indivíduos ou organizações que buscam contratar profissionais da música

**Funcionalidades:**

1. **Captura de Dados de Entrada**
   * Cada campo de formulário atualiza o estado central via setFormData
   * Utiliza spread operator para preservar outros campos do estado
2. **Seleção de Tipo de Conta**
   * Sistema de radio buttons personalizados para seleção visual
   * Feedback visual imediato ao selecionar uma opção
3. **Responsividade**
   * Layout adaptativo com diferentes comportamentos em dispositivos móveis e desktop
   * Painel informativo oculto em telas pequenas para foco no formulário

### **4.3.2 - Step 2 do Registro**

A segunda etapa do formulário de cadastro multi-etapas que permite ao usuário fornecer informações sobre si mesmo (biografia) e selecionar tags relacionadas aos seus interesses musicais.

**Componente**: Step2  
**Arquivo**: Register/components/Step2.tsx

**Props:**

1. **formData** - Estado atual do formulário
2. **setFormData** - Função para atualizar o estado do formulário
3. **handleNextStep** - Função para avançar para o próximo passo
4. **handlePreviousStep** - Função para retornar ao passo anterior

**Estados e Constantes:**

1. **tags** - Array de strings contendo os gêneros musicais disponíveis para seleção
   * Lista estática incluindo: 'Jazz', 'Rock', 'Pop', 'Eletrônica', 'Clássica', 'MPB'
   * Note: Contém TODO para implementação de carregamento dinâmico via API

**Elementos de UI:**

1. **Container Principal**
   * Classe: min-h-screen flex flex-col md:flex-row overflow-auto
   * Altura mínima de tela completa
   * Layout flexível que muda de coluna (mobile) para linha (desktop)
   * Permite rolagem quando o conteúdo excede o tamanho da tela
2. **Seção "Sobre Você" (Lado Esquerdo)**
   * Classe: md:w-1/2 w-full bg-white p-6 md:p-12 flex flex-col
   * Ocupa toda a largura em dispositivos móveis e metade em telas médias ou maiores
   * Fundo branco com padding adaptativo
   * Layout flexível em coluna
   * Contém:

a. **Título da Seção**

* + - Classe: text-2xl md:text-3xl font-bold text-black mb-4 md:mb-6
    - Tamanho de texto responsivo

b. **Texto Descritivo**

* + - Classe: text-gray-400 text-sm md:text-base mb-6 md:mb-8
    - Texto explicativo sobre a finalidade do campo de biografia

c. **Campo de Texto (Textarea)**

* + - Classe: w-full h-60 md:h-96 p-4 md:p-6 bg-gray-100 rounded-xl text-black placeholder-gray-500 focus:outline-none resize-none
    - Campo de texto grande com altura responsiva (mais alto em desktop)
    - Cantos arredondados (rounded-xl)
    - Não permite redimensionamento manual (resize-none)
    - Conectado ao estado formData.about

1. **Seção "Tags" (Lado Direito)**
   * Classe: md:w-1/2 w-full bg-black p-6 md:p-12 flex flex-col justify-between
   * Ocupa toda a largura em dispositivos móveis e metade em telas médias ou maiores
   * Fundo preto com padding adaptativo
   * Layout flexível em coluna com espaço entre elementos
   * Contém:

a. **Cabeçalho da Seção**

* + - Título: text-2xl md:text-3xl font-bold text-white mb-4 md:mb-6
    - Texto descritivo: text-gray-400 text-sm md:text-base mb-6 md:mb-8
    - Explicação sobre a finalidade e uso das tags

b. **Grid de Tags**

* + - Classe: grid grid-cols-1 sm:grid-cols-2 gap-3 md:gap-4
    - Uma coluna em dispositivos pequenos, duas colunas em telas maiores
    - Cada tag renderizada como um botão:
      * Classe base: p-2 md:p-3 rounded-full border-2 transition-all text-sm md:text-base
      * Estilização condicional baseada na seleção:
        + Selecionada: bg-white text-black border-white
        + Não selecionada: border-gray-500 text-white hover:border-white
      * Prefixo "#" antes do nome da tag
      * Atributo aria-pressed para acessibilidade

c. **Barra de Navegação**

* + - Classe: flex flex-row md:flex-row justify-between items-center md:mt-0
    - Posicionada na parte inferior da seção
    - Contém dois botões de navegação:
      * **Botão Voltar**:
        + Classe: md:w-auto bg-black text-white rounded-full p-3 md:p-3 hover:bg-gray-800 transition-colors mt-4 md:mt-0
        + Ícone de seta para esquerda (ArrowLeftIcon)
        + Aciona handlePreviousStep quando clicado
      * **Botão Avançar**:
        + Classe: md:w-auto bg-white text-black rounded-full p-3 md:p-3 hover:bg-gray-300 transition-colors mt-4 md:mt-0
        + Ícone de seta para direita (ArrowRightIcon)
        + Aciona handleNextStep quando clicado

**Funcionalidades:**

1. **Gestão de Biografia**
   * Campo de texto livre para o usuário descrever sua jornada artística
   * Atualização do estado central via evento onChange
2. **Seleção de Tags**
   * Função: toggleTag(tag) - Alterna a seleção de uma tag específica
   * Lógica de alternância:
     + Se a tag já está selecionada, a remove do array
     + Se a tag não está selecionada, a adiciona ao array
   * Feedback visual imediato sobre a seleção
3. **Navegação entre Etapas**
   * Botões para avançar para a próxima etapa ou retornar à anterior
   * Posicionamento consistente na interface
4. **Responsividade**
   * Layout adaptativo com diferentes comportamentos em dispositivos móveis e desktop
   * Mudança de layout flexível de coluna para linha
   * Ajustes de tamanho, espaçamento e padding baseados no tamanho da tela

### **4.3.3 - Step 3 do Registro**

A terceira e última etapa do formulário de cadastro multi-etapas que permite ao usuário fazer upload de uma foto de perfil e finalizar o processo de registro.

**Componente**: Step3  
**Arquivo**: Register/components/Step3.tsx

**Props:**

1. **formData** - Estado atual do formulário
2. **setFormData** - Função para atualizar o estado do formulário
3. **handleSubmit** - Função para enviar o formulário completo
4. **handlePreviousStep** - Função para retornar ao passo anterior

**Elementos de UI:**

1. **Container Principal**
   * Classe: h-screen flex flex-col md:flex-row overflow-y-auto md:overflow-hidden relative
   * Altura fixa de tela completa
   * Layout flexível que muda de coluna (mobile) para linha (desktop)
   * Permite rolagem vertical em dispositivos móveis, oculta em desktop
2. **Seção "Foto de Perfil" (Lado Esquerdo)**
   * Classe: w-full md:w-1/2 bg-black p-12 flex flex-col justify-between min-h-screen md:min-h-0
   * Ocupa toda a largura em dispositivos móveis e metade em telas médias ou maiores
   * Fundo preto com padding interno de 3rem
   * Altura mínima de tela completa em mobile, sem mínimo em desktop
   * Contém:

a. **Título da Seção**

* + - Classe: text-3xl font-bold text-white mb-5 mt-10 text-center
    - Texto grande centralizado com espaçamento

b. **Área Central (Container Flexível)**

* + - Classe: flex-1 flex flex-col justify-center items-center
    - Ocupa o espaço disponível com alinhamento central
    - Contém o container para a imagem e botões:
      * Classe: flex flex-col items-center gap-6
      * Layout flexível em coluna com espaçamento entre elementos
      * **Área da Foto de Perfil**:
        + Classe: relative w-40 h-40 md:w-64 md:h-64 rounded-full bg-[#D9D9D9] border-2 border-gray-400
        + Formato circular com tamanho responsivo
        + Fundo cinza claro com borda
        + Exibe preview da imagem quando selecionada:

Classe: w-full h-full object-cover rounded-full

Imagem redimensionada para cobrir o círculo mantendo proporções

* + - * **Botão de Upload**:
        + Classe: px-6 py-2 bg-gray-800 text-white rounded-full hover:bg-gray-700 transition-colors text-sm
        + Aciona o input file oculto quando clicado
      * **Input File Oculto**:
        + Atributo hidden para ocultar visualmente
        + Aceita apenas formatos de imagem (image/jpeg, image/png)
        + Conectado à função handleFileUpload
      * **Texto Informativo**:
        + Classe: text-gray-400 text-center text-sm
        + Indica os formatos suportados e limitações

1. **Seção "Confirmação" (Lado Direito)**
   * Classe: w-full md:w-1/2 bg-white p-12 flex flex-col justify-between relative min-h-screen md:min-h-0
   * Fundo branco com mesmas proporções do lado esquerdo
   * Contém:

a. **Faixa "SOUNDBUDDY" (Versão Mobile)**

* + - Classe: overflow-x-hidden block md:hidden absolute top-0 left-0 w-full bg-black py-2 text-white text-center text-lg font-bold whitespace-nowrap
    - Faixa horizontal no topo para dispositivos móveis
    - Texto repetido como efeito decorativo

b. **Faixa "SOUNDBUDDY" (Versão Desktop)**

* + - Classe: hidden md:block absolute top-20 -right-40 w-[500px] bg-black py-1 text-white text-xl font-bold rotate-45 overflow-hidden
    - Faixa diagonal posicionada no canto superior direito
    - Texto repetido como efeito decorativo, rotacionado 45 graus

c. **Mensagem de Confirmação**

* + - Classe: flex-1 flex flex-col items-center justify-center text-center
    - Layout flexível centralizado
    - Título: text-3xl md:text-4xl font-bold text-black mb-4
    - Subtítulo: text-lg md:text-xl text-gray-600

d. **Barra de Navegação**

* + - Classe: flex flex-row md:flex-row justify-between items-center md:mt-0
    - Posicionada na parte inferior da seção
    - Contém dois botões:
      * **Botão Voltar**:
        + Classe: md:w-auto text-black rounded-full p-3 md:p-3 hover:bg-gray-800 transition-colors mt-4 md:mt-0
        + Ícone de seta para esquerda (ArrowLeftIcon)
        + Aciona handlePreviousStep quando clicado
      * **Botão Finalizar**:
        + Classe: md:w-auto bg-black text-white rounded-full p-3 md:p-3 hover:bg-gray-300 transition-colors mt-4 md:mt-0
        + Ícone de seta para direita (ArrowRightIcon)
        + Aciona handleSubmit quando clicado

**Funcionalidades:**

1. **Upload de Imagem de Perfil**
   * Função: handleFileUpload(e) - Processa o arquivo de imagem selecionado
   * Captura o arquivo do evento e atualiza o estado formData.profileImage
   * Exibe preview da imagem quando selecionada
2. **Navegação entre Etapas**
   * Botão para retornar à etapa anterior (handlePreviousStep)
   * Botão para finalizar e enviar o formulário completo (handleSubmit)
3. **Feedback Visual**
   * Mensagem de confirmação indicando conclusão do cadastro
   * Preview em tempo real da imagem selecionada
4. **Responsividade**
   * Layouts diferentes para dispositivos móveis e desktop
   * Duas versões da faixa decorativa "SOUNDBUDDY" (horizontal para mobile, diagonal para desktop)
   * Ajustes de tamanho, espaçamento e comportamento de scroll

## **4.4 - Home Page**

A Home Page é a página principal após a autenticação do usuário, servindo como hub central para todas as funcionalidades da aplicação. Esta página apresenta um layout dividido em quatro seções principais: uma barra lateral esquerda (LeftSidebar), um feed central de conteúdo, um header e uma barra lateral direita (RightSidebar).

**Componente**: Home  
**Arquivo**: app/Home/page.tsx

**Elementos de UI:**

1. **Container Principal**
   * Classe: 'flex flex-col h-screen bg-gradient-to-br from-zinc-800 via-gray-700 to-zinc-900'
   * Layout de coluna flexível que ocupa toda a altura da tela
   * Organiza os componentes filhos verticalmente
2. **Header**
   * Posicionado no topo da interface
   * Componente importado separadamente (./components/Header)
3. **Área de Conteúdo Principal**
   * Classe: flex gap-4
   * Layout flexível com espaçamento entre os componentes (gap-4)
   * Contém:
     + **LeftSidebar**
       - Wrapper com classe: bg-white
       - Componente importado de ./components/LeftSidebar
     + **Conteúdo Principal (Feed)**
       - Classe: flex-1 overflow-y-auto px-4 py-6
       - Cresce para ocupar o espaço disponível (flex-1)
       - Rolagem vertical quando o conteúdo excede o espaço (overflow-y-auto)
       - Padding horizontal de 1rem (px-4) e vertical de 1.5rem (py-6)
     + **RightSidebar**
       - Wrapper com classe: bg-white
       - Componente importado de ./components/RightSidebar

### **4.4.1 - LEFTSIDEBAR**

O LeftSidebar é um componente de navegação lateral responsivo que exibe o perfil do usuário, tags de gêneros musicais e funcionalidades de navegação específicas para artistas/músicos, incluindo acesso a eventos, contratações e posts salvos.

**Componente:** LeftSidebar  
**Arquivo:** Home/components/LeftSidebar/index.jsx

**Estados:**

1. **router** - Hook de roteamento do Next.js para navegação entre páginas
2. **isOpen** - Controla a visibilidade da sidebar (expandida ou recolhida)
3. **isCollapsed** - Rastreia se a sidebar deve estar no modo colapsado (em telas menores)
4. **modalAtivo** - Estado unificado para controlar todos os modais
   * **tipo** - Indica qual modal está ativo ('eventos', 'contratacoes', 'seguidores', 'salvos')
   * **subtipo** - Usado para diferenciar subtipos de modais ('seguidores', 'seguindo')

**Elementos de UI:**

1. **Barra Lateral Principal**
   * Classe: bg-black fixed border-2 border-gray-700 shadow-sm shadow-blue-950 left-0 top-[60px] bottom-[15px] w-64 rounded-r-xl transition-transform duration-300 z-50 flex flex-col
   * Container com posicionamento fixo, cor de fundo preta e bordas arredondadas à direita
   * Transição suave para animação de abertura/fechamento
   * Estado visual controlado por: ${isOpen ? 'translate-x-0' : '-translate-x-full'}
2. **Área de Perfil do Usuário**
   * Classe: text-center mb-6
   * Contém:
     + Foto de perfil:
       - Classe: w-20 h-20 rounded-full mx-auto mb-1 object-cover
       - Imagem circular centralizada com cover para preenchimento
     + Nome de usuário:
       - Classe: text-lg font-medium mb-4 truncate text-white cursor-pointer
       - Texto com prefixo "@" e cursor pointer para navegação ao perfil
     + Avaliação/Rating:
       - Classe: border-2 border-white rounded-full px-3 flex items-center text-white
       - Exibe número de avaliação junto com ícone de estrela (StarIcon)
     + Contadores de seguidores:
       - Classe: flex justify-center gap-8 text-sm
       - Exibe quantidade de seguidores e seguindo com interatividade para abrir modais
3. **Tags/Gêneros Musicais**
   * Classe: grid grid-cols-3 gap-2 text-white
   * Grid de 3 colunas com tags de gêneros musicais
   * Cada tag:
     + Classe: px-3 py-1 bg-gray-800 rounded-full text-xs text-center truncate
     + Formato hashtag (#) com fundo escuro e cantos arredondados
4. **Menu de Navegação**
   * Classe: flex-1 align-center border-t border-gray-700
   * Lista de opções (ul) com itens clicáveis:
     + Classe item: p-3 rounded-lg hover:bg-gray-800 cursor-pointer transition-colors
     + Efeito hover que escurece o fundo
     + Opções disponíveis: Eventos, Contratações, Salvos
5. **Área de Rodapé**
   * Classe: border-t text-white border-gray-700 pt-4 space-y-2 p-5
   * Contém botão de sair:
     + Classe: w-full rounded-full bg-red-600 hover:bg-red-400 p-1 text-sm transition-colors
     + Botão vermelho de largura total com hover em tom mais claro
6. **Controles Responsivos**
   * Botão para fechar sidebar (visível quando isCollapsed && isOpen):
     + Classe: absolute right-4 top-4 bg-black text-white rounded-full w-8 h-8 flex items-center justify-center hover:bg-gray-800 shadow-lg border border-gray-600
     + Botão "×" posicionado no canto superior direito da sidebar
   * Botão para abrir sidebar (visível quando isCollapsed):
     + Classe: fixed z-50 bg-black text-white rounded-full cursor-pointer text-2xl w-10 h-10 flex items-center justify-center transition-all duration-300 bottom-5 hover:bg-gray-800
     + Botão "☰" fixo no canto inferior esquerdo da tela
     + Posição controlada por: ${isOpen ? 'left-64 opacity-0' : 'left-3 opacity-100'}
7. **Modais**
   * **Modal de Eventos:**
     + Componente importado (EventosCalendarioModal)
     + Visibilidade controlada por modalAtivo.tipo === 'eventos'
   * **Modal de Contratações:**
     + Componente importado (ContratacoesModal)
     + Visibilidade controlada por modalAtivo.tipo === 'contratacoes'
   * **Modal de Seguidores:**
     + Componente importado (SeguidoresModal)
     + Visibilidade controlada por modalAtivo.tipo === 'seguidores'
     + Utiliza modalAtivo.subtipo para diferenciar entre 'seguidores' e 'seguindo'
   * **Modal de Posts Salvos:**
     + Componente importado (PostsSalvosModal)
     + Visibilidade controlada por modalAtivo.tipo === 'salvos'

**Funcionalidades:**

1. **Responsividade Automática**
   * Função: Efeito useEffect para controlar o comportamento responsivo
   * Detecta automaticamente o tamanho da tela e ajusta a visibilidade da sidebar
   * Em telas menores que 1200px, colapsa automaticamente
   * Em telas maiores, permanece sempre visível
   * Listener de redimensionamento para ajuste dinâmico
2. **Sistema de Autenticação**
   * Função: handleLogout() - Realiza o logout do usuário
   * Navega de volta para a página inicial ('/')
   * Preparado para integração futura com API de autenticação
3. **Navegação para Perfil**
   * Função: handleProfileClick() - Navega para a página de perfil do usuário
   * Utiliza o hook useRouter do Next.js para redirecionamento
4. **Sistema Unificado de Modais**
   * Função: abrirModal(tipo, subtipo = null) - Abre o modal especificado
   * Função: fecharModal() - Fecha qualquer modal ativo
   * Gerenciamento centralizado de todos os modais através do estado modalAtivo
   * Suporte para subtipos de modais (usado para seguidores/seguindo)
5. **Exibição de Informações do Perfil**
   * Exibe dados do perfil do usuário: imagem, username, avaliação (rating)
   * Mostra contadores de seguidores e seguindo com interatividade
   * Preparado para integração com API de perfil do usuário
6. **Gestão de Tags Musicais**
   * Exibe grid de tags/gêneros musicais do artista
   * Layout responsivo com truncamento para tags longas
   * Preparado para integração com API de perfil para carregar tags específicas do usuário

### **4.4.2 - FEED**

O Feed é o componente central da Home Page, exibindo um feed de posts de vários usuários em formato de timeline e permitindo interações como salvar posts, iniciar conversas e visualizar diferentes tipos de mídia (imagens, vídeos e áudio).

**Componente:** Feed  
**Arquivo:** Home/components/Feed/index.jsx

**Estados:**

1. **savedPosts** - Set que rastreia quais posts foram salvos pelo usuário
2. **showModal** - Controla a visibilidade do modal de novo post
3. **selectedUser** - Armazena o usuário selecionado para chat
4. **enlargedImage** - Controla a visualização ampliada de imagens
5. **enlargedVideo** - Controla a visualização ampliada de vídeos
6. **posts** - Array que armazena os dados dos posts exibidos no feed
7. **router** - Hook de roteamento do Next.js para navegação entre páginas

**Elementos de UI:**

1. **Container Principal**
   * Classe: top-[50px] flex-1 overflow-hidden
   * Container para todo o conteúdo do feed, posicionado abaixo do Header
2. **Botão de Novo Post**
   * Classe: fixed z-40 bottom-5 left-1/2 -translate-x-1/2 bg-black shadow-lg text-white rounded-full w-14 h-14 flex items-center justify-center text-3xl hover:bg-gray-800 shadow-xl transition-all
   * Botão flutuante fixo na parte inferior central da tela
   * Ícone "+" em branco sobre fundo preto
   * Efeito hover que escurece levemente o botão
   * Aciona o modal de criação de post quando clicado
3. **Container do Feed**
   * Classe: fixed top-[60px] bottom-0 md:bottom-[15px] left-1/2 -translate-x-1/2 w-full max-w-4xl
   * Posicionamento fixo com distância específica do topo e fundo
   * Centralizado horizontalmente na tela
   * Largura máxima definida
   * Responsivo para diferentes tamanhos de tela (notação md:)
4. **Área de Posts**
   * Classe: h-full bg-gradient-to-t from-white/10 to-transparent rounded-xl mx-auto overflow-hidden scrollbar-right shadow-2xl relative
   * Container com gradiente de transparência com cantos arredondados
   * Sombra elevada para destaque visual
   * Contém:
     + Área rolável com os posts:
       - Classe: h-full overflow-y-auto px-4 py-6 pb-25 space-y-6
       - Rolagem vertical quando o conteúdo excede o espaço
       - Espaçamento uniforme entre os posts (space-y-6)
       - Padding interno nas quatro direções
5. **Card de Post**
   * Classe: relative bg-white shadow-lg rounded-2xl p-4
   * Fundo branco com sombra suave e cantos arredondados
   * Posição relativa para posicionamento absoluto dos elementos filhos
   * Contém:
     + Cabeçalho do post:
       - Classe: flex items-center mb-4
       - Avatar do usuário (imagem circular)
       - Nome de usuário com prefixo "@" e cursor pointer para navegação ao perfil
     + Botão de salvar:
       - Classe: absolute top-2 right-2 p-2 hover:bg-gray-200 rounded-full transition-colors
       - Ícone BookmarkIcon que muda de cor quando o post é salvo
     + Conteúdo do post:
       - Classe: text-gray-800 pr-8
       - Texto principal do post
     + Mídia (condicional):
       - Renderização específica para cada tipo de mídia (imagem, vídeo ou áudio)
     + Botão de chat:
       - Classe: absolute bottom-2 right-2 p-2 hover:bg-gray-200 rounded-full transition-colors
       - Ícone PaperAirplaneIcon rotacionado em -45 graus
6. **Visualização de Mídia**
   * **Imagem:**
     + Classe do container: mt-3 mb-10 flex justify-center
     + Container da imagem: relative rounded-lg overflow-hidden cursor-pointer max-w-md
     + Classe da imagem: w-full h-auto rounded-lg object-cover
     + Imagem clicável que abre visualização ampliada
   * **Vídeo:**
     + Classe do container: mt-3 mb-10 flex justify-center
     + Container do vídeo: relative rounded-lg overflow-hidden max-w-md
     + Player de vídeo com controles nativos
     + Vídeo clicável que abre visualização ampliada
   * **Áudio:**
     + Classe do container: mt-3 mb-10 bg-gray-100 rounded-lg p-3
     + Cabeçalho com ícone MusicalNoteIcon e nome do arquivo
     + Player de áudio com controles nativos
     + Exibe o nome do arquivo de áudio com ícone musical
7. **Modais**
   * **Modal de Novo Post:**
     + Componente importado (NewPostModal)
     + Visibilidade controlada por showModal
   * **Modal de Chat:**
     + Componente importado (ChatModal)
     + Exibido quando selectedUser não é null
   * **Modal de Imagem Ampliada:**
     + Classe: fixed inset-0 z-50 bg-black/80 backdrop-blur-sm flex items-center justify-center p-4
     + Fundo preto semi-transparente com desfoque
     + Exibe a imagem em tamanho maior com limites máximos
     + Botão de fechar no canto superior direito
   * **Modal de Vídeo Ampliado:**
     + Classe: fixed inset-0 z-50 bg-black/80 flex items-center justify-center p-4
     + Estrutura similar ao modal de imagem, mas para vídeos
     + Reprodução automática do vídeo (autoPlay)
     + Botão de fechar no canto superior direito

**Funcionalidades:**

1. **Gerenciamento de Posts Salvos**
   * Função: toggleSave(postId) - Alterna o salvamento de um post
   * Interface visual que reflete o estado de salvamento via cor do ícone de bookmark
   * Utiliza um Set para armazenar e verificar eficientemente os IDs dos posts salvos
2. **Criação de Novos Posts**
   * Função: handleNewPost(postData) - Adiciona um novo post ao feed
   * Suporta posts de texto simples (string) ou com mídia (objeto)
   * Gera um ID único baseado no timestamp (Date.now())
   * Posiciona novos posts no topo do feed
3. **Interação com Usuários**
   * Função: handleOpenChat(user) - Abre o modal de chat com um usuário específico
   * Permite iniciar conversas a partir de qualquer post
   * Captura username e profileImg do usuário selecionado
4. **Navegação para Perfil**
   * Função: handleProfileClick() - Navega para a página de perfil
   * Utiliza o hook useRouter do Next.js para redirecionamento
5. **Visualização de Mídia**
   * Função: renderMedia(post) - Renderiza o componente de mídia apropriado
   * Suporte para três tipos de mídia: imagem, vídeo e áudio
   * Renderização condicional baseada no tipo de mídia
   * Modais de visualização ampliada para imagens e vídeos
6. **Responsividade**
   * Estrutura responsiva que se adapta a diferentes tamanhos de tela
   * Uso de classes flex, max-width e media queries (md:) para garantir compatibilidade
   * Containers com largura máxima e centralização horizontal

### **4.4.3 - Right Sidebar**

A barra lateral direita exibe conversas recentes do usuário e permite iniciar novas conversas. Possui comportamento responsivo, colapsando automaticamente em telas menores e exibindo um botão flutuante para reabertura.

**Componente**: RightSidebar  
**Arquivo**: Home/components/RightSidebar/index.tsx

**Estados:**

1. **isOpen** - Controla a visibilidade da sidebar (booleano)
2. **isCollapsed** - Rastreia se a sidebar deve estar no modo colapsado em telas menores (booleano)
3. **selectedChat** - Armazena o chat selecionado atualmente (objeto do tipo ChatType ou null)

**Elementos de UI:**

1. **Container da Sidebar**
   * Classe: `bg-black fixed border-2 border-gray-700 shadow-sm shadow-blue-950 right-0 top-[60px] bottom-[15px] w-64 rounded-l-xl transition-transform duration-300 z-50 flex flex-col`
   * Posição fixa no lado direito da tela
   * Altura do topo (60px) até o fundo (15px) da janela
   * Largura de 64 unidades (16rem)
   * Borda arredondada apenas no lado esquerdo
   * Transição suave para animação de abertura/fechamento
   * Layout em coluna (flex-col)
   * Translação horizontal controlada por estado isOpen
2. **Área de Chats**
   * Classe: flex-1 text-center overflow-y-auto scrollbar-right p-5
   * Cresce para preencher o espaço disponível (flex-1)
   * Conteúdo centralizado horizontalmente
   * Rolagem vertical automática quando necessário
   * Espaçamento interno de 5 unidades
   * Contém:
     + **Título**: "Chats" (h2)
       - Classe: text-xl font-bold text-white pb-4 border-b border-gray-700
       - Texto branco, negrito, tamanho xl
       - Borda inferior cinza escura
3. **Lista de Chats**
   * Classe: space-y-3 mt-4 pr-2
   * Espaçamento vertical de 3 unidades entre os itens
   * Margem superior de 4 unidades
   * Margem direita de 2 unidades
   * Itens de Chat:
     + Classe: p-3 rounded-lg hover:bg-gray-800 cursor-pointer transition-colors relative
     + Preenchimento de 3 unidades
     + Bordas arredondadas
     + Mudança de cor ao passar o mouse (hover)
     + Cursor do tipo ponteiro
     + Layout interno:
       - **Avatar de Contato**
         * Classe: w-10 h-10 rounded-full bg-gray-700 flex items-center justify-center text-white
         * Círculo de 10x10 unidades
         * Exibe a primeira letra do nome do contato
       - **Informações do Chat**
         * **Nome do Contato**

Classe: font-medium text-white truncate

Texto branco, peso médio, truncado se necessário

* + - * + **Horário**

Classe: text-xs text-gray-400

Texto pequeno, cinza claro

* + - * + **Prévia da Mensagem**

Classe: text-sm text-gray-400 truncate

Texto pequeno, cinza claro, truncado se necessário

1. **Botão para Fechar Sidebar**
   * Visível apenas quando isCollapsed e isOpen são verdadeiros
   * Classe: absolute right-4 top-4 bg-black text-white rounded-full w-8 h-8 flex items-center justify-center hover:bg-gray-800 shadow-lg border border-gray-600
   * Botão "X" no canto superior direito
   * Posição absoluta
   * Tamanho 8x8 unidades
   * Borda circular com texto branco
2. **Botão Flutuante para Abrir Sidebar**
   * Visível apenas quando isCollapsed é verdadeiro
   * Classe: fixed z-50 bg-black text-white rounded-full cursor-pointer text-2xl w-10 h-10 flex items-center justify-center transition-all duration-300 bottom-5 hover:bg-gray-800
   * Posição fixa no canto inferior direito
   * Índice z elevado (50)
   * Ícone de mensagem (ChatBubbleOvalLeftIcon)
   * Posição horizontal controlada por estado isOpen
3. **Modal de Chat**
   * Renderizado condicionalmente quando um chat é selecionado (selectedChat não nulo)
   * Componente importado ChatModal com propriedades:
     + user: objeto com username do chat selecionado
     + onClose: função para fechar o modal

**Funcionalidades:**

1. **Controle Responsivo da Sidebar**
   * Função: Efeito useEffect com handleResize
   * Detecta largura da janela e ajusta comportamento:
     + Em telas menores que 1200px:
       - Define isCollapsed como verdadeiro
       - Fecha a sidebar (isOpen falso)
     + Em telas maiores que 1200px:
       - Define isCollapsed como falso
       - Mantém a sidebar aberta (isOpen verdadeiro)
   * Executa na montagem do componente e em cada redimensionamento da janela
2. **Seleção de Chat**
   * Função: onClick nos itens de chat
   * Define o chat clicado como selectedChat
   * Abre o modal de chat correspondente
3. **Abertura/Fechamento da Sidebar**
   * Botão flutuante para abrir: alterna o estado isOpen
   * Botão "X" para fechar: define isOpen como falso

### **4.4.4 - Header**

O componente Header implementa uma barra de busca inteligente com sugestões dinâmicas que são filtradas conforme o usuário digita, incluindo tags populares e perfis de usuários.

**Componente**: Header  
**Arquivo**: Home/components/Header.tsx

**Estados:**

1. **searchQuery** - Armazena o texto digitado na barra de pesquisa (string)
2. **showSuggestions** - Controla a visibilidade do painel de sugestões (booleano)
3. **filteredSuggestions** - Armazena as sugestões filtradas baseadas no input do usuário (array)

**Refs:**

1. **searchRef** - Referência para o elemento da barra de pesquisa (usado para detectar cliques fora)

**Dados estáticos:**

1. **popularTags** - Array de strings contendo tags populares para sugestões (18 tags)
2. **suggestedProfiles** - Array de strings contendo perfis sugeridos (12 perfis)

**Elementos de UI:**

1. **Container Principal**
   * Classe: fixed top-0 w-full z-50 h-16
   * Posição fixa no topo da tela
   * Largura total
   * Índice z elevado (50)
   * Altura de 16 unidades (4rem)
   * Contém:
     + **Container centralizado**
       - Classe: mx-auto h-full px-4
       - Margem horizontal automática
       - Altura total
       - Preenchimento horizontal de 4 unidades
2. **Container de Conteúdo**
   * Classe: relative flex items-center justify-center h-full
   * Referência searchRef para detecção de cliques externos
   * Posição relativa
   * Layout flexível centralizado vertical e horizontalmente
   * Altura total
3. **Container da Barra de Pesquisa**
   * Classe: w-full max-w-md md:max-w-lg lg:max-w-xl transition-all duration-300
   * Largura total com máximos responsivos (md, lg)
   * Transição suave para mudanças de tamanho
   * Contém:
     + **Barra de Pesquisa**
       - Classe: flex items-center bg-gray-100 rounded-full px-4 py-2 transition-colors hover:bg-gray-200
       - Layout flexível, itens centralizados
       - Fundo cinza claro, mais escuro no hover
       - Formato arredondado (pill)
       - Preenchimento interno horizontal (4) e vertical (2)
       - Contém:
         * **Ícone de Lupa**

Classe: w-4 h-4 md:w-5 md:h-5 text-gray-500 mr-2

Tamanho responsivo

Cor cinza

Margem à direita

* + - * + **Campo de Input**

Classe: w-full bg-transparent outline-none text-gray-800 placeholder-gray-500 text-sm md:text-base

Largura total

Fundo transparente

Sem contorno de foco

Texto cinza escuro, placeholder cinza

Tamanho de texto responsivo

1. **Painel de Sugestões**
   * Visível apenas quando showSuggestions é verdadeiro
   * Classe: absolute top-16 left-1/2 transform -translate-x-1/2 w-full max-w-md md:max-w-lg lg:max-w-xl mt-1 bg-white rounded-xl shadow-lg p-3
   * Posição absoluta logo abaixo da barra de pesquisa
   * Centralizado horizontalmente usando transform
   * Largura responsiva correspondente à barra de pesquisa
   * Fundo branco com sombra
   * Bordas arredondadas
   * Contém duas seções:
     + **Seção de Tags Populares**
       - Classe: mb-3
       - Margem inferior de 3 unidades
       - Contém:
         * **Título da Seção**

Classe: text-xs text-gray-500 mb-2

Texto pequeno e cinza

Margem inferior de 2 unidades

* + - * + **Lista de Tags**

Classe: flex space-x-1 overflow-x-auto pb-2

Layout flexível horizontal

Espaçamento horizontal de 1 unidade

Rolagem horizontal quando necessário

Tags Individuais:

Classe: px-3 py-1 bg-gray-100 hover:bg-gray-200 rounded-full text-xs md:text-sm transition-colors whitespace-nowrap flex-shrink-0

Formato pill (arredondado)

Preenchimento horizontal (3) e vertical (1)

Tamanho de texto responsivo

Fundo cinza claro, mais escuro no hover

Impede quebra de linha (nowrap)

Não encolhe (flex-shrink-0)

* + - **Seção de Sugestões de Busca**
      * Visível apenas quando há sugestões filtradas
      * Contém:
        + **Título da Seção**

Classe: text-xs text-gray-500 mb-2

Texto pequeno e cinza

Margem inferior de 2 unidades

* + - * + **Lista de Sugestões**

Classe: flex flex-col space-y-2

Layout flexível vertical

Espaçamento vertical de 2 unidades

Itens de Sugestão:

Classe: flex items-center p-2 hover:bg-gray-100 rounded-lg cursor-pointer

Layout flexível horizontal

Preenchimento de 2 unidades

Bordas levemente arredondadas

Fundo destacado no hover

Contém:

**Ícone de Lupa**

Classe: w-4 h-4 text-gray-400 mr-2

Tamanho 4x4 unidades

Cor cinza clara

Margem direita de 2 unidades

**Texto da Sugestão**

Classe: text-sm

Texto pequeno

Prefixo "#" para tags

**Indicador de Tipo**

Classe: ml-auto text-xs text-gray-400

Alinhado à direita (ml-auto)

Texto muito pequeno e cinza

Exibe "Tag" ou "Profile"

**Funcionalidades:**

1. **Detecção de Cliques Externos**
   * Função: Efeito useEffect com handleClickOutside
   * Detecta cliques fora do componente de pesquisa
   * Fecha o painel de sugestões quando clica-se fora da área de pesquisa
   * Utiliza searchRef para verificar se o clique foi dentro ou fora
2. **Filtragem de Sugestões**
   * Função: Efeito useEffect dependente de searchQuery
   * Acionado quando o termo de busca muda
   * Filtra tags e perfis que correspondem ao termo de busca:
     + Converte a consulta para minúsculas
     + Filtra as tags que incluem o termo (limitado a 5)
     + Filtra os perfis que incluem o termo (limitado a 3)
     + Combina resultados em um array estruturado com tipo diferenciado
     + Define o estado filteredSuggestions
3. **Manipulação do Input**
   * Função: handleInputChange
   * Atualiza o estado searchQuery com o valor do input
   * Exibe o painel de sugestões (showSuggestions verdadeiro)
4. **Seleção de Sugestão**
   * Função: handleSuggestionClick
   * Define o termo de busca como o valor da sugestão selecionada
   * Fecha o painel de sugestões
5. **Seleção de Tag Popular**
   * Função: handleTagClick
   * Define o termo de busca como a tag selecionada
   * Fecha o painel de sugestões

**4.5 - Profile Page**

A página ProfilePage é o componente principal que renderiza o perfil de um usuário, com conteúdo adaptável conforme seja o próprio perfil do usuário logado ou o perfil de outro usuário.

**Componente**: ProfilePage

**Arquivo**: Profile/page.tsx

**Hooks:**

1. **useParams** - Utilizado para obter o parâmetro de username da URL
2. **useState** - Gerencia estados locais do componente
3. **useEffect** - Gerencia efeitos colaterais e ciclo de vida

**Estados:**

1. **windowWidth** - Armazena a largura atual da janela para responsividade
2. **isOwnProfile** - Indica se o perfil sendo visualizado pertence ao usuário logado

**Dados mockados:**

1. **profileData** - Objeto contendo todos os dados do perfil do usuário:
   * username - Nome de usuário
   * image - URL da imagem de perfil
   * bio - Biografia/descrição do usuário
   * followers - Número de seguidores
   * following - Número de usuários que o perfil segue
   * posts - Array de publicações do usuário
   * events - Array de eventos agendados do usuário
   * hires - Array de contratações recentes
   * savedItems - Array de itens salvos pelo usuário

**Elementos de UI:**

1. **Container Principal**
   * Classe: min-h-screen bg-[#424242] md:bg-[linear-gradient(to\_left,\_black\_10%,\_#1a1a1a\_20%,\_#424242\_100%)]
   * Ocupa toda a altura da tela
   * Fundo cinza em dispositivos móveis
   * Gradiente em telas médias e maiores
2. **Botão de Alternância (teste)**
   * Classe: fixed top-4 left-4 z-50 bg-blue-500 hover:bg-blue-600 text-white px-4 py-2 rounded-md shadow-lg transition-colors
   * Posição fixa no canto superior esquerdo
   * Alterna entre visualização de próprio perfil e perfil de outro usuário
   * Texto dinâmico baseado no estado atual
3. **Container de Layout Principal**
   * Classe: relative
   * Posição relativa para controle de layout
   * Contém:
     + **Sidebar** (condicional)
       - Exibida apenas quando isOwnProfile é verdadeiro
       - Recebe props de eventos, contratações e itens salvados
     + **Área de Conteúdo Principal**
       - Classe dinâmica baseada no estado isOwnProfile:
         * Com sidebar: w-full transition-all duration-300 lg:pr-96
         * Sem sidebar: w-full transition-all duration-300
       - Transição suave para ajuste de largura
       - Espaço reservado para sidebar em telas grandes quando o perfil é do próprio usuário
       - Contém:
         * **Container Centralizado**: container mx-auto
         * **Componente ProfileMain**: Recebe userData e isOwnProfile como props

**Funcionalidades:**

1. **Obtenção de Parâmetros da URL**
   * Extrai o username da URL usando useParams
   * Utiliza para personalizar o perfil sendo visualizado
2. **Detecção Responsiva**
   * Função: useEffect com handleResize
   * Monitora e armazena a largura da janela
   * Executado no carregamento inicial e a cada redimensionamento
   * Remove o event listener na desmontagem do componente
3. **Alternância de Modo de Visualização**
   * Botão para alternar entre visualizações (próprio perfil vs. perfil de terceiro)
   * Controla a exibição da sidebar e o ajuste do layout principal
   * Funcionalidade de teste para simular diferentes visualizações
4. **Layout Adaptativo**
   * Ajusta o layout baseado no estado isOwnProfile
   * Reserva espaço para sidebar em telas grandes quando é o próprio perfil
   * Background responsivo que muda em breakpoints diferentes

### **4.5.1 - Perfil**

O componente ProfileMain exibe a página de perfil de um usuário, permitindo visualizar e editar informações pessoais, gerenciar tags de interesse musical, ver seguidores e seguindo, além de exibir e gerenciar publicações.

**Componente**: ProfileMain  
**Arquivo**: app/Profile/components/perfil.tsx

**Props:**

1. **userData** - Objeto contendo os dados do usuário a ser exibido
2. **isOwnProfile** - Booleano que indica se o perfil visualizado pertence ao usuário logado

**Estados:**

1. **showModal** - Controla a exibição do modal de seguidores/seguindo
2. **modalType** - Define o tipo de lista a ser exibida no modal ('seguidores' ou 'seguindo')
3. **isFollowing** - Indica se o usuário logado está seguindo o perfil visualizado
4. **isLoading** - Indica processamento de operações assíncronas
5. **isEditingProfile** - Controla o modo de edição do perfil
6. **bio** - Armazena o texto da biografia do usuário
7. **posts** - Armazena localmente as publicações do usuário
8. **selectedTags** - Armazena as tags selecionadas pelo usuário
9. **tagError** - Armazena mensagens de erro relacionadas às tags
10. **imagePreview** - Armazena a pré-visualização da nova foto de perfil

**Elementos de UI:**

1. **Container Principal**
   * Classe: overflow-y-auto space-y-6 max-w-4xl mx-auto pb-16
   * Permite rolagem vertical
   * Espaçamento vertical entre seções
   * Largura máxima limitada e centralizada
   * Preenchimento inferior para acomodar possíveis elementos fixos
2. **Seção Superior - Perfil**
   * Classe: bg-black text-white rounded-b-xl p-6 shadow-xl
   * Fundo preto com texto branco
   * Cantos inferiores arredondados
   * Sombreamento elevado
   * Contém:
     + **Container Flex**
       - Classe: flex flex-col lg:flex-row items-center gap-6
       - Layout responsivo: colunas em dispositivos móveis, linhas em telas grandes
       - Alinhamento central dos itens
       - Espaçamento entre itens
       - Contém:
         * **Foto de Perfil**

Classe: relative

Imagem circular com borda azul

Em modo de edição, inclui sobreposição para alteração da foto

Input de arquivo oculto para upload da nova imagem

* + - * + **Informações do Usuário**

Classe: text-center lg:text-left flex-1 relative

Alinhamento de texto responsivo

Layout flexível para acomodar nome de usuário e contadores

Contém:

**Nome de Usuário**: @username

**Contadores de Seguidores/Seguindo**

Elementos clicáveis para abrir modais

Exibem números com rótulos descritivos

**Botões de Ação**

"Editar perfil" para perfil próprio

"Seguir"/"Deixar de seguir" para perfis de outros usuários

Em modo de edição, botões "Salvar" e "Cancelar"

1. **Seção de Tags**
   * Classe: flex flex-wrap justify-center gap-2 px-4
   * Layout flexível com quebra de linha
   * Tags centralizadas
   * Espaçamento entre as tags
   * Em modo de visualização:
     + Exibe as tags selecionadas pelo usuário
     + Formato: #TagName em tags com bordas brancas
   * Em modo de edição:
     + Exibe todas as tags disponíveis para seleção
     + Tags selecionadas têm fundo branco com texto preto
     + Tags não selecionadas têm fundo transparente com texto branco
     + Possível mensagem de erro abaixo das tags
2. **Seção Sobre**
   * Classe: bg-white rounded-xl shadow-md p-6
   * Fundo branco com cantos arredondados e sombra
   * Contém:
     + **Título**: "Sobre"
     + Em modo de visualização:
       - Exibe o texto da biografia
       - Mensagem alternativa quando não há biografia
     + Em modo de edição:
       - Textarea para edição da biografia
       - Estilização com foco em acessibilidade (focus rings)
3. **Seção de Posts**
   * Classe: bg-white rounded-xl shadow-md p-6
   * Fundo branco com cantos arredondados e sombra
   * Contém:
     + **Título**: "Publicações"
     + **Lista de Posts**
       - Cada post inclui:
         * Foto do autor
         * Nome do autor e data
         * Conteúdo do post
         * Possível mídia associada
         * Menu de opções (apenas para posts próprios)
     + **Mensagem Alternativa**
       - Exibida quando não há publicações
4. **Modal de Seguidores/Seguindo**
   * Componente: SeguidoresModal
   * Exibido condicionalmente com base no estado showModal
   * Tipo determinado pelo estado modalType
   * Exibe lista de seguidores ou usuários seguidos

**Funcionalidades:**

1. **Gestão de Perfil**
   * Função: handleEditProfile()
     + Ativa o modo de edição
     + Inicializa os estados com valores atuais
   * Função: handleSaveProfile()
     + Valida as tags selecionadas
     + Simula requisição para salvar alterações
     + Atualiza dados localmente
   * Função: handleCancelEdit()
     + Desativa o modo de edição
     + Descarta alterações não salvas
   * Função: handleImageChange(e)
     + Processa o upload de nova foto de perfil
     + Gera preview da imagem antes de salvar
2. **Gestão de Tags**
   * Função: toggleTag(tag)
     + Adiciona ou remove tags da seleção
     + Aplica validações de quantidade mínima (3) e máxima (6)
     + Atualiza mensagens de erro quando necessário
   * Função: validateTags()
     + Verifica se a quantidade de tags está dentro dos limites permitidos
     + Retorna boolean indicando validade
3. **Gestão de Relacionamentos**
   * Função: handleFollowToggle()
     + Alterna entre seguir e deixar de seguir
     + Simula requisição e atualiza estado local
     + Inclui tratamento de erro
   * Função: openModal(type)
     + Abre modal de seguidores ou seguindo
     + Define o tipo de lista a ser exibida
   * Função: closeModal()
     + Fecha o modal
4. **Gestão de Posts**
   * Função: handleDeletePost(postId)
     + Exibe confirmação antes de excluir
     + Simula requisição de exclusão
     + Remove post da lista local

**Validações e Regras:**

1. **Tags**
   * Quantidade mínima: 3
   * Quantidade máxima: 6
   * Feedback visual de erro quando os limites são violados
2. **Edição de Perfil**
   * Restrita ao próprio usuário
   * Confirmação para salvar alterações
   * Possibilidade de cancelar sem salvar alterações
3. **Exclusão de Posts**
   * Restrita ao próprio usuário
   * Confirmação antes da exclusão
   * Atualização instantânea da UI após exclusão

**Notas de Implementação:**

1. **Simulações de API**
   * O código atual simula chamadas de API com timeouts
   * Existem comentários com implementações reais comentadas
   * Estados locais são atualizados para simular respostas do servidor
2. **Tratamento de Erros**
   * Validações client-side antes de enviar requisições
   * Bloco try/catch para lidar com erros de API
   * Estados de loading para feedback ao usuário
3. **Responsividade**
   * Layout adapta-se entre dispositivos móveis e telas grandes
   * Alinhamento de texto e organização de elementos mudam conforme tamanho da tela
   * Foto e informações do usuário empilhadas em mobile, lado a lado em desktop

**4.5.2 - SIDEBAR (PÁGINA DE PERFIL)**

A barra lateral (Sidebar) fornece informações rápidas sobre eventos próximos, contratações recentes e itens salvos. É um componente responsivo que pode ser expandido ou recolhido dependendo do tamanho da tela.

**Componente**: Sidebar  
**Arquivo**: Profile/components/Sidebar.tsx

**Props:**

1. **events** - Array de eventos próximos
2. **hires** - Array de contratações recentes
3. **savedItems** - Array de itens salvos pelo usuário

**Estados:**

1. **isOpen** - Controla se a sidebar está visível ou oculta
2. **isCollapsed** - Determina se a sidebar deve estar no modo recolhido (em telas menores)

**Elementos de UI:**

1. **Container Principal da Sidebar**
   * Classe: bg-black fixed text-white space-y-5 right-0 top-0 bottom-0 w-full md:bg-transparent md:w-96 transition-transform duration-300 z-40 scrollbar-right
   * Posição fixa na tela, ocupando toda a altura
   * Fundo preto em mobile, transparente em telas médias (md)
   * Largura completa em mobile, 24rem em telas médias (md)
   * Transição suave de 300ms para animação de entrada/saída
   * Z-index elevado (z-40) para sobrepor outros elementos
   * Classe dinâmica para controle de visibilidade: translate-x-0 (visível) ou translate-x-full (oculto)
   * Em telas grandes (lg) sempre visível: lg:translate-x-0
2. **Conteúdo da Sidebar**
   * Classe: pb-4 pr-4 pl-4 space-y-5 h-full overflow-y-auto
   * Padding interno (pb-4 pr-4 pl-4)
   * Espaçamento vertical entre elementos (space-y-5)
   * Ocupa toda a altura disponível (h-full)
   * Rolagem vertical quando o conteúdo excede o espaço (overflow-y-auto)
3. **Botão para Fechar Sidebar (Responsivo)**
   * Exibido condicionalmente quando isCollapsed && isOpen
   * Classe: absolute right-4 top-4 bg-black text-white rounded-full w-8 h-8 flex items-center justify-center hover:bg-gray-800 shadow-lg border border-gray-600
   * Posicionado no canto superior direito da sidebar
   * Utiliza ícone XMarkIcon do Heroicons
4. **Seções de Conteúdo**

a. **Próximos Eventos**

* + Cabeçalho com ícone:
    - Flex layout: flex items-center gap-2 mb-4
    - Ícone CalendarIcon do Heroicons
    - Título: text-xl font-semibold
  + Lista de eventos:
    - Container: space-y-3
    - Cada evento:
      * Classe: border-2 border-white rounded-full px-4 py-1 flex justify-between items-center hover:bg-gray-800 transition-colors
      * Borda branca arredondada em formato pill
      * Layout flexível entre nome (truncate) e data (text-sm whitespace-nowrap)
      * Efeito hover que escurece o fundo

b. **Últimas Contratações**

* + Cabeçalho com ícone:
    - Flex layout: flex items-center gap-2 mb-4
    - Ícone UserGroupIcon do Heroicons
    - Título: text-xl font-semibold
  + Grid de avatares:
    - Layout: grid grid-cols-3 gap-3 place-items-center
    - Cada avatar:
      * Classe: w-16 h-16 rounded-full object-cover border-4 border-white hover:border-blue-400 transition-colors
      * Imagem circular fixa com borda branca
      * Efeito hover que muda a cor da borda para azul

c. **Itens Salvos**

* + Cabeçalho com ícone:
    - Flex layout: flex items-center gap-2 mb-4
    - Ícone BookmarkIcon do Heroicons
    - Título: text-xl font-semibold
  + Lista de itens:
    - Container: space-y-3
    - Cada item:
      * Classe: border-2 border-white rounded-lg p-3 hover:bg-gray-800 transition-colors
      * Borda branca arredondada
      * Título: font-medium truncate
      * Descrição: text-gray-400 text-sm truncate
      * Efeito hover que escurece o fundo

1. **Botão para Abrir Sidebar (Responsivo)**
   * Exibido condicionalmente quando isCollapsed
   * Classe: fixed z-50 bg-black text-white rounded-full cursor-pointer w-12 h-12 flex items-center justify-center transition-all duration-300
   * Posição dinâmica baseada no estado da sidebar
   * Z-index muito elevado (z-50)
   * Posicionado no canto superior direito
   * Utiliza ícone Bars3Icon do Heroicons

**Funcionalidades:**

1. **Responsividade Automática**
   * Usa useEffect para detectar o tamanho da tela e ajustar o comportamento
   * Em telas menores que 1024px (lg), colapsa automaticamente
   * Define isOpen baseado no tamanho da tela
2. **Toggle de Visibilidade**
   * Função toggleSidebar() para alternar entre aberto/fechado
   * Botões dedicados para abrir e fechar em modo móvel
3. **Renderização Condicional**
   * Exibe até 3 eventos próximos ou mensagem "Nenhum evento próximo"
   * Exibe até 6 fotos de contratações ou mensagem "Nenhuma contratação recente"
   * Exibe os 3 últimos itens salvos ou mensagem "Nenhum item salvo"
4. **Importações de Dependências**
   * useState e useEffect do React para gerenciamento de estado
   * Ícones do Heroicons para elementos visuais

## **4.6 - Modais**

### **4.6.1 - Modal de Eventos do Calendário**

Modal que permite ao usuário visualizar eventos futuros e adicionar novos eventos ao calendário.

**Componente**: EventosCalendarioModal  
**Arquivo**: Home/components/LeftSidebar/EventosCalendario.tsx

**Estados:**

1. **activeTab** - Controla qual aba está ativa (eventos ou novo evento)
2. **novoEvento** - Armazena os dados do formulário de novo evento

**Elementos de UI:**

1. **Container Principal do Modal**
   * Classe: fixed inset-0 bg-black/50 backdrop-blur z-50 flex items-center justify-center
   * Cobre toda a tela com fundo semi-transparente e efeito de desfoque
   * Z-index elevado para sobreposição de outros elementos
2. **Box do Modal**
   * Classe: bg-black rounded-xl pt-6 pb-6 pl-6 pr-6 max-w-md w-full mx-4 shadow-2xl
   * Fundo preto com cantos arredondados
   * Padding uniforme de 1.5rem
   * Largura máxima definida com responsividade
3. **Cabeçalho do Modal**
   * Classe: flex justify-between items-center mb-4
   * Título dinâmico baseado na aba ativa
   * Botão de fechar no canto direito
4. **Navegação por Abas**
   * Classe: flex border-b border-gray-700 mb-4
   * Botões de aba com estilização condicional para indicar a aba ativa
5. **Conteúdo da Aba "Eventos"**
   * Lista scrollável: space-y-3 max-h-80 overflow-y-auto pb-5 scrollbar-right
   * Cards de eventos com informações essenciais:
     + Classe de cada card: border-2 border-white rounded-lg p-3 hover:bg-gray-700 cursor-pointer transition-colors
     + Exibe data, local, título e contratante
   * Botão para adicionar novo evento
6. **Conteúdo da Aba "Novo Evento"**
   * Formulário com campos:
     + Data (input type="date")
     + Horário (input type="time")
     + Título (input type="text")
     + Local (input type="text")
     + Contratante (input type="text")
   * Classe de cada campo: w-full bg-gray-800 border border-gray-700 rounded-lg p-2 text-white
   * Botões de ação (Cancelar e Salvar)

**Funcionalidades:**

1. **Navegação entre Abas**
   * Mudança entre visualização de eventos e adição de novos eventos
2. **Gestão de Formulário**
   * Função: handleChange(e) atualiza o estado conforme o usuário digita
   * Todos os campos são obrigatórios (required)
3. **Envio de Formulário**
   * Função: handleSubmit(e) processa os dados enviados
   * Reseta o formulário após o envio bem-sucedido
   * Retorna à aba de listagem de eventos

### **4.6.2 - Modal de Contratações**

Modal que permite ao usuário gerenciar contratações de shows/eventos, visualizando contratações existentes e adicionando novas.

**Componente**: ContratacoesModal  
**Arquivo**: Home/components/LeftSidebar/ContratacoesModal.tsx

**Estados:**

1. **activeTab** - Controla qual aba está ativa (minhas contratações ou nova contratação)
2. **formData** - Armazena os dados do formulário de nova contratação
3. **rating** - Armazena a avaliação (1-5 estrelas)
4. **hoverRating** - Controla o efeito visual ao passar o mouse nas estrelas

**Elementos de UI:**

1. **Container Principal do Modal**
   * Classe: fixed inset-0 bg-black/50 backdrop-blur z-50 flex items-center justify-center
   * Similar ao modal de eventos
2. **Box do Modal**
   * Classe: bg-black rounded-xl pt-6 pb-6 pl-6 pr-6 max-w-md w-full mx-4 shadow-2xl
   * Estilização consistente com o modal de eventos
3. **Cabeçalho do Modal**
   * Classe: flex justify-between items-center mb-4
   * Título "Contratações"
   * Botão de fechar no canto direito
4. **Navegação por Abas**
   * Classe: flex border-b border-gray-700 mb-4
   * Botões para "Minhas Contratações" e "Nova Contratação"
5. **Conteúdo da Aba "Minhas Contratações"**
   * Lista scrollável: space-y-3 max-h-80 overflow-y-auto pb-5 scrollbar-right
   * Cards de contratações com informações detalhadas:
     + Classe de cada card: border-2 border-white rounded-lg p-3 hover:bg-gray-700 cursor-pointer transition-colors
     + Exibe data, valor, nome do evento, cliente (com avatar) e avaliação
   * Botão para adicionar nova contratação
6. **Conteúdo da Aba "Nova Contratação"**
   * Formulário com campos:
     + Nome da Contratação (input type="text")
     + Cliente (input type="text")
     + Data (input type="date")
     + Valor (input type="text")
     + Avaliação (sistema de 5 estrelas interativo)
   * Classe de cada campo: w-full bg-gray-800 border border-gray-700 rounded-lg p-2 text-white
   * Sistema de avaliação por estrelas:
     + Interativo com hover e click
     + Utiliza ícones diferentes para estrelas preenchidas e vazias
   * Botões de ação (Cancelar e Salvar Contratação)

**Funcionalidades:**

1. **Navegação entre Abas**
   * Alternância entre visualização e adição de contratações
2. **Gestão de Formulário**
   * Função: handleInputChange(e) atualiza o estado conforme o usuário digita
   * Todos os campos de texto são obrigatórios (required)
3. **Sistema de Avaliação**
   * Interação para selecionar de 1 a 5 estrelas
   * Feedback visual ao passar o mouse sobre as estrelas
4. **Envio de Formulário**
   * Função: handleSubmit(e) processa os dados enviados
   * Reseta o formulário após o envio bem-sucedido
   * Retorna à aba de listagem de contratações

### **4.6.3 - New Post Modal**

O NewPostModal é um componente que permite aos usuários criar e publicar novos posts, com suporte para texto e diferentes tipos de mídia (imagem, vídeo e áudio).

**Componente**: NewPostModal  
**Arquivo**: Home/components/Feed/NewPostModal.jsx

**Props:**

1. **show** - Boolean que controla a visibilidade do modal
2. **onClose** - Função para fechar o modal
3. **onSubmit** - Função para submeter o novo post
4. **currentUser** - Objeto com dados do usuário atual (avatar e username)

**Estados:**

1. **content** - Texto do post sendo criado
2. **media** - Arquivo de mídia selecionado pelo usuário
3. **mediaPreview** - Preview da mídia selecionada
4. **mediaType** - Tipo da mídia ('image', 'video' ou 'audio')
5. **isLoading** - Indica se o post está sendo enviado

**Refs:**

1. **mediaInputRef** - Referência para o input de arquivo de imagem/vídeo
2. **musicInputRef** - Referência para o input de arquivo de áudio

**Elementos de UI:**

1. **Container Principal do Modal**
   * Classe: fixed inset-0 z-150 bg-black/50 backdrop-blur flex items-center justify-center p-4
   * Cobre toda a tela com fundo semi-transparente e efeito de desfoque
   * Z-index elevado para sobreposição de outros elementos
2. **Box do Modal**
   * Classe: bg-white rounded-2xl w-full max-w-xl shadow-xl relative
   * Fundo branco com cantos arredondados
   * Largura máxima definida com responsividade
   * Botão de fechar (X) no canto superior direito
3. **Área do Conteúdo**
   * Classe: p-6
   * Contém:
     + **Informações do Usuário**:
       - Avatar e nome de usuário
       - Classe: flex items-center space-x-3 mb-4
     + **Textarea**:
       - Classe: w-full min-h-[150px] p-0 border-none outline-none resize-none text-gray-800 placeholder-gray-400
       - Campo para digitação do conteúdo do post
       - Auto-foco quando o modal é aberto
     + **Preview da Mídia** (condicional):
       - Exibição diferente baseada no tipo de mídia
       - Botão para remover a mídia selecionada
4. **Área de Ações**
   * Botão de publicar:
     + Classe: px-8 py-2 bg-black text-white text-sm font-medium rounded-full hover:bg-gray-800 transition-colors disabled:bg-gray-300 disabled:cursor-not-allowed
     + Desabilitado quando não há conteúdo ou durante o envio
     + Texto dinâmico (Publicar/Publicando...)
   * Inputs ocultos para upload de arquivos:
     + Um para imagens/vídeos e outro para áudios
   * Ícones de mídia:
     + Classe do container: flex justify-center p-4 space-x-8 border-t border-gray-200
     + Ícones clicáveis para adicionar música ou imagem/vídeo

**Funcionalidades:**

1. **Envio de Formulário**
   * Função: handleSubmit(e) - Processa os dados do post
   * Suporta envio de texto simples ou com mídia
   * Simulação de upload de mídia (URL.createObjectURL)
2. **Upload de Mídia**
   * Função: handleMediaUpload(e) - Processa uploads de imagem/vídeo
   * Função: handleMusicUpload(e) - Processa uploads de áudio
   * Validações de:
     + Tipo de arquivo (mime type)
     + Tamanho máximo (5MB para imagens, 10MB para áudio, 20MB para vídeos)
   * Criação de previews para feedback visual imediato
3. **Gerenciamento do Modal**
   * Função: handleClose() - Fecha o modal e reseta os estados
   * Reset completo de todos os campos ao fechar o modal
4. **Feedback Visual**
   * Botão de publicar desabilitado quando não há conteúdo
   * Estado de carregamento (isLoading) durante o envio
   * Previews específicos para cada tipo de mídia

### **4.6.4 - Chatmodal**

Modal de chat que permite aos usuários comunicarem-se em tempo real, visualizando histórico de mensagens e enviando novas mensagens com indicadores de status (enviado/lido) e animação de "digitando".

**Componente**: ChatModal  
**Arquivo**: app/componentes/ ChatModal.tsx

**Props**:

1. **user** - Objeto contendo dados do usuário com quem se está conversando
2. **onClose** - Função para fechar o modal

**Estados**

1. **message** - Armazena o texto da mensagem atual sendo digitada
2. **messages** - Armazena todas as mensagens da conversa
3. **isTyping** - Controla a animação de "digitando..."

**Referências (useRef)**

1. **messagesEndRef** - Referência para o elemento do final das mensagens (usado para auto-scroll)
2. **textareaRef** - Referência para o textarea (usado para auto-resize)

**Elementos de UI**

1. **Container Principal do Modal**
   * Classe: fixed inset-0 md:inset-auto md:right-0 md:top-[60px] md:bottom-[15px] md:w-64 md:rounded-l-xl bg-black bg-opacity-95
   * Posicionamento fixo em tela cheia em mobile e lateral direita em desktop
2. **Container do Chat**
   * Classe: bg-gray-900 w-full h-full md:rounded-l-xl flex flex-col shadow-xl border-l border-gray-800
   * Estrutura principal com flex column para organização vertical dos elementos
3. **Cabeçalho do Chat**
   * Classe: bg-black px-4 py-3 flex justify-between items-center md:rounded-tl-xl border-b border-gray-800
   * Contém:
     + Avatar do usuário (imagem ou inicial do nome)
     + Nome de usuário (@username)
     + Botão de fechar (X)
4. **Área de Mensagens**
   * Classe: flex-1 overflow-y-auto p-4 bg-black space-y-4 scrollbar-thin
   * Scrollável com customização da barra de rolagem
   * Exibe indicador de data (Hoje, Ontem, etc.)
   * Contém:
     + Mensagens recebidas (alinhadas à esquerda)
     + Mensagens enviadas (alinhadas à direita)
     + Indicador de "digitando..." (três pontos animados)
     + Elemento invisível para auto-scroll
5. **Balões de Mensagem**
   * Classe para mensagens enviadas: bg-gray-800 text-gray-200 rounded-br-none
   * Classe para mensagens recebidas: bg-gray-700 text-gray-300 rounded-bl-none
   * Exibem:
     + Texto da mensagem
     + Horário (formato HH:MM)
     + Indicadores de status (enviado/lido) para mensagens enviadas
6. **Área de Input**
   * Classe: bg-black px-3 py-3 md:rounded-bl-xl border-t border-gray-800
   * Contém:
     + Botão para adicionar imagens (PhotoIcon)
     + Textarea com auto-resize
     + Botão de enviar mensagem (PaperAirplaneIcon)

**Funcionalidades**

1. **Auto-scroll para Mensagens Novas**
   * Função: scrollToBottom()
   * Rola automaticamente para a mensagem mais recente quando novas mensagens são adicionadas
2. **Auto-resize do Textarea**
   * Ajusta automaticamente a altura do textarea conforme o conteúdo
   * Limita a altura máxima a 100px
3. **Envio de Mensagens**
   * Função: handleSendMessage()
   * Adiciona a mensagem ao estado
   * Limpa o campo de input
   * Simula resposta do destinatário
   * Atualiza o status da mensagem para "lida"
4. **Envio via Teclado**
   * Função: handleKeyPress(e)
   * Permite enviar mensagem ao pressionar Enter (sem Shift)
   * Shift+Enter insere quebra de linha
5. **Formatação de Data**
   * Função: formatDate(date)
   * Formata a data de agrupamento de mensagens (Hoje, Ontem ou data específica)
6. **Indicadores de Status**
   * Exibe check simples para mensagens enviadas
   * Exibe check duplo verde para mensagens lidas
7. **Animação de "Digitando..."**
   * Exibe três pontos com animação de bounce
   * Controlado pelo estado isTyping

### **4.6.5 – PostsSalvosModal**

Modal que exibe posts salvos pelo usuário, permitindo visualizar conteúdo salvo, incluindo diferentes tipos de mídia (imagens, vídeos e áudios), com interface de usuário responsiva e funcionalidade para remover posts da lista de salvos.

**Componente**:PostsSalvosModal  
**Arquivo**: app/leftsidebar/PostsSalvosModal.tsx

**Props**:

1. **show** - Booleano que controla a visibilidade do modal
2. **onClose** - Função para fechar o modal

**Estados**

1. **postsSalvos** - Armazena a lista de posts salvos pelo usuário
2. **loading** - Controla o estado de carregamento dos dados

**Efeitos (useEffect)**

1. **carregarPostsSalvos** - Efeito para carregar os posts salvos quando o modal é aberto

**Funções**

1. **removerPostSalvo** - Remove um post da lista de salvos
2. **renderMedia** - Renderiza componentes de mídia baseados no tipo (imagem, vídeo ou áudio)

**Elementos de UI**

1. **Container Principal do Modal**
   * Classe: fixed inset-0 bg-black/50 backdrop-blur z-55 flex items-center justify-center
   * Overlay com fundo escuro semitransparente e efeito de blur
2. **Container do Conteúdo**
   * Classe: bg-black rounded-xl w-full max-w-xl h-full md:h-3/4 flex flex-col scrollbar-right
   * Modal responsivo com largura máxima e altura adaptativa
3. **Cabeçalho do Modal**
   * Classe: flex items-center justify-between p-4 border-b
   * Contém:
     + Título "Posts Salvos"
     + Botão de fechar (XMarkIcon)
4. **Área de Conteúdo**
   * Classe: flex-1 overflow-y-auto p-4
   * Scrollável para exibir múltiplos posts
   * Exibe:
     + Indicador de carregamento (spinner animado)
     + Mensagem quando não há posts salvos
     + Lista de posts com espaçamento vertical
5. **Cards de Posts**
   * Classe: bg-white border border-blue-500 rounded-lg p-4
   * Contém:
     + Avatar do usuário
     + Nome de usuário (@username)
     + Conteúdo textual do post
     + Mídia associada (quando existente)
     + Botão para remover dos salvos
6. **Componentes de Mídia**
   * **Imagem**: Container responsivo com imagem em formato de card
   * **Vídeo**: Player de vídeo responsivo com controles nativos
   * **Áudio**: Player de áudio com ícone de nota musical e nome do arquivo
7. **Rodapé do Modal**
   * Classe: border-t p-4
   * Contém botão "Fechar" em largura total

**Funcionalidades**

1. **Carregamento de Posts Salvos**
   * Função: carregarPostsSalvos()
   * Exibe indicador de carregamento durante a busca
   * Simula integração com API (mockado temporariamente)
2. **Remoção de Posts**
   * Função: removerPostSalvo(postId)
   * Remove o post da lista local
   * Prepara integração futura com backend
3. **Renderização Condicional de Mídia**
   * Função: renderMedia(post)
   * Suporta múltiplos tipos de mídia (imagens, vídeos, áudios)
   * Exibe controles apropriados para cada tipo
4. **Feedback Visual**
   * Exibe spinner durante carregamento
   * Mensagem amigável quando não há posts salvos
   * Feedback de interação nos botões (hover states)
5. **Estrutura Responsiva**
   * Layout adaptável para mobile e desktop
   * Altura total em mobile, 3/4 da altura em desktop
   * Largura máxima controlada para melhor experiência em telas grandes

### **4.6.6 - SeguidoresModal**

Modal responsivo para exibição e gerenciamento de seguidores ou usuários seguidos, permitindo visualizar listas de usuários e remover conexões.

**Componente**: SeguidoresModal

**Arquivo**: app/componentes/SeguidoresModal.tsx

**Props**:

1. **show** - Controla a visibilidade do modal (boolean)
2. **onClose** - Função para fechar o modal
3. **type** - Tipo de lista a ser exibida ('seguidores' ou 'seguindo')

**Estados**:

1. **usuarios** - Armazena a lista de usuários (seguidores ou seguindo)
2. **showConfirmModal** - Controla a visibilidade do modal de confirmação
3. **selectedUser** - Armazena o usuário selecionado para remoção

**Efeitos (useEffect)**:

1. **Carregamento de Dados** - Busca os usuários quando o modal é aberto

**Elementos de UI**:

1. **Backdrop**
   * Classe: fixed inset-0 bg-black/50 backdrop-blur z-55
   * Fundo escuro semi-transparente que fecha o modal ao clicar
2. **Container Principal do Modal**
   * Classe: fixed top-1/2 left-1/2 transform -translate-x-1/2 -translate-y-1/2 bg-black text-white border border-gray-700 rounded-xl w-full max-w-md z-55 scrollbar-right
   * Posicionamento centralizado na tela
   * Largura máxima de 400px (max-w-md)
3. **Cabeçalho do Modal**
   * Classe: flex items-center justify-between border-b border-gray-700 px-6 py-4
   * Contém:
     + Título dinâmico baseado no tipo (Seguidores ou Seguindo)
     + Botão de fechar (XMarkIcon)
4. **Lista de Usuários**
   * Classe: max-h-96 overflow-y-auto scrollbar-left py-2
   * Altura máxima com rolagem vertical
   * Para cada usuário exibe:
     + Avatar (imagem de perfil)
     + Nome completo
     + Nome de usuário (@username)
     + Botão para remover conexão
5. **Item da Lista (Usuário)**
   * Classe: px-6 py-3 hover

flex items-center justify-between

* + Efeito hover ao passar o mouse
  + Dividido em:
    - Área clicável para navegar ao perfil do usuário
    - Botão de remover (XMarkIcon)

1. **Modal de Confirmação**
   * Classe: fixed top-1/2 left-1/2 transform -translate-x-1/2 -translate-y-1/2 bg-black text-white border border-gray-700 rounded-xl p-6 z-55 w-80
   * Exibe:
     + Título "Confirmar remoção"
     + Mensagem de confirmação personalizada baseada no tipo
     + Botões "Cancelar" e "Remover"

**Funcionalidades**:

1. **Carregamento de Usuários**
   * Função: fetchUsuarios()
   * Carrega a lista de seguidores ou seguindo (mock implementado, pendente integração com API)
2. **Navegação para Perfil**
   * Função: handleProfileClick(usuario)
   * Navega para o perfil do usuário selecionado (pendente integração com router)
3. **Remoção de Usuário**
   * Função: handleRemoveUsuario(usuario)
   * Inicia o processo de remoção, exibindo o modal de confirmação
4. **Confirmação de Remoção**
   * Função: confirmRemoveUsuario()
   * Remove o usuário da lista
   * Fecha o modal de confirmação
5. **Cancelamento de Remoção**
   * Função: cancelRemoveUsuario()
   * Fecha o modal de confirmação sem remover o usuário
6. **Estado Vazio**
   * Exibe mensagem apropriada quando não há seguidores/seguindo
   * Texto personalizado baseado no tipo

# **Padrões de Design**

A interface do SoundBuddy segue um conjunto consistente de padrões de design:

1. **Sistema de Cores**
   * Cores primárias: Preto (pelo tailwind = black) e Branco (pelo tailwind = white)
   * Cores secundárias: Tons de cinza (do cinza escuro ao cinza claro dentro das especificações do Tailwindcss)
   * Elementos de destaque: Verde e vermelho dentro da padronização do tailwind
2. **Tipografia**
   * Sistema de tamanhos responsivos baseado em classes Tailwind
   * Hierarquia clara: títulos em tamanho grande e negrito, texto regular em tamanhos médios
   * Elementos de interface em tamanho menor
3. **Responsividade**
   * Design mobile-first com breakpoints para telas médias (md) e maiores
   * Adaptação de layout, tamanhos e cores conforme o tamanho da tela
   * Elementos empilhados em dispositivos móveis, alinhados horizontalmente em telas maiores
4. **Interatividade**
   * Transições suaves para interações (hover, clicks)
   * Feedback visual em elementos interativos
   * Modais para ações secundárias ou complexas

# **MOTIVAÇÃO DO DESIGN**

**Foco na Música e Conexão Social:**

A aplicação é definida como uma plataforma de música e conexões sociais. Isso motiva a estrutura geral com um feed central para descoberta de conteúdo, perfis detalhados para artistas e clientes, e funcionalidades de interação direta como chat e contratações.

**Fluxo de Usuário Lógico:**

O design segue uma sequência lógica e comum em aplicações sociais: Landing Page para introdução e acesso, Login/Registro para autenticação, uma Home Page centralizadora e páginas de Perfil detalhadas. O processo de registro em etapas foi projetado para guiar o novo usuário de forma progressiva, coletando informações básicas primeiro e depois detalhes do perfil, tornando o processo menos intimidador.

**Estrutura Modular e Reutilizável**

A organização do código em componentes (como Header, Sidebars, Feed, Modais) e a utilização de um framework como React facilitam a manutenção e escalabilidade. Componentes como ChatModal ou NewPostModal são reutilizados, garantindo consistência visual e funcional.

**Design Responsivo (Mobile-First):**

Há uma preocupação com a adaptabilidade da interface a diferentes tamanhos de tela. Padrões como flex-col mudando para flex-row, sidebars que se recolhem ou se tornam flutuantes em telas menores, e o uso extensivo de classes responsivas do Tailwind CSS indicam uma abordagem que prioriza a experiência móvel, adaptando-se progressivamente para desktops.

**Interface Clara e Consistente:**

* **Sistema de Cores:** A paleta principal baseada em preto, branco e tons de cinza cria um visual moderno e focado no conteúdo, com cores de destaque (verde, vermelho) usadas pontualmente para ações específicas (como botões de confirmação ou logout).
* **Tipografia:** A hierarquia tipográfica definida pelo Tailwind ajuda a organizar a informação visualmente.
* **Layout:** O layout da Home Page (Left Sidebar, Feed, Right Sidebar) é um padrão comum em redes sociais, facilitando a navegação para usuários familiarizados com essas estruturas. A página de perfil também segue uma estrutura clara com seções bem definidas.
* **Interatividade e Feedback Visual:** O design incorpora elementos, transições suaves, indicadores visuais (como o ícone de post salvo ou a seleção de tags) e modais para ações focadas sem perder o contexto da página principal.
* **Foco na Funcionalidade Central:** Elementos como o botão flutuante para criar um novo post ou a barra de busca inteligente com sugestões no Header são projetados para dar acesso rápido às funcionalidades principais da plataforma.

# **Referências Técnicas**

**Tecnologias Utilizadas**

* **Next.js**: Framework React com roteamento embutido
* **React**: Biblioteca para construção de interfaces
* **Tailwind CSS**: Framework CSS para estilização rápida e responsiva

**Componentes React**

* Uso de componentes funcionais com hooks (useState, useRouter)
* Gerenciamento de estado local para controle da interface
* Renderização condicional para elementos dinâmicos