

UNIVERSIDADE CRUZEIRO DO SUL
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS
ANÁLISE E PROJETO DE SISTEMAS 2

PROTÓTIPO FUNCIONAL DE APLICATIVO PARA:
VIU: Gestor de Mídia Digital

LUCAS FERNANDES RIBEIRO DE OLIVEIRA
RAFAEL DE OLIVEIRA BATISTA
RENAN SANTOS DO NASCIMENTO
VINICIUS COSTA BRAGA

São Paulo
2025

LUCAS FERNANDES RIBEIRO DE OLIVEIRA
RAFAEL DE OLIVEIRA BATISTA
RENAN SANTOS DO NASCIMENTO
VINICIUS COSTA BRAGA

PROTÓTIPO FUNCIONAL DE APLICATIVO PARA: VIU: Gestor de Mídia Digital

Projeto apresentado como requisito parcial da disciplina Análise e Projeto de Sistemas II, do Curso de Análise e Desenvolvimento de Sistemas da Universidade Cruzeiro do Sul, sob orientação dos Professores Luis Pires.

São Paulo
2025

RESUMO

A VIU é uma plataforma inovadora de exibição e gerenciamento de mídias em TVs comerciais, desenvolvida para modernizar a comunicação visual em ambientes físicos como restaurantes, academias, escolas e comércios locais.

Com o VIU, os usuários podem atualizar conteúdos em tempo real de forma prática e centralizada, substituindo materiais impressos por uma solução digital, sustentável e atrativa.

O sistema opera diretamente no navegador da TV, dispensando instalações complexas e garantindo compatibilidade com diferentes dispositivos.

A interface é intuitiva e oferece experiência otimizada para telas grandes, além de métricas no dashboard que permitem monitorar desempenho e engajamento.

O projeto representa uma união entre tecnologia, praticidade e inovação, transformando a maneira como empresas se comunicam visualmente com seus clientes.

Palavras-chave: tecnologia, comunicação digital, inovação, sustentabilidade, experiência visual, mídia indoor, interface intuitiva, modernização.

ABSTRACT

VIU is an innovative platform for displaying and managing media on commercial TVs, designed to modernize visual communication in physical spaces such as restaurants, gyms, schools, and retail stores.

With VIU, users can update content in real time, replacing printed materials with a digital, sustainable, and engaging solution.

The system runs directly through the TV browser, requiring no complex installation and ensuring compatibility across devices.

Its intuitive interface provides an optimized large-screen experience and includes dashboard metrics to track performance and engagement.

The project combines technology, practicality, and innovation, redefining how businesses visually connect with their audiences.

Keywords: technology, digital communication, innovation, sustainability, visual experience, indoor media, intuitive interface, modernization.

LISTA DE FIGURAS

Figura 01 - SiteMap.....	21
Figura 02 - VIU Acesso.....	22
Figura 03 - SiteMap Agenda.....	23
Figura 04 - Protótipo Tela Inicial.....	51
Figura 05 - Protótipo Tela de Cadastro.....	52
Figura 06 - Protótipo Tela de Login.....	53
Figura 07 - Protótipo Tela de Mídias.....	54
Figura 08 - Protótipo Detalhes do Grupo.....	55
Figura 09 - Protótipo Tela de Analytics.....	56
Figura 10 - Protótipo Tela de Perfil.....	57
Figura 11 - Protótipo Tela /tv.....	58
Figura 12 - Diagrama de Classes.....	59
Figura 13 - Diagrama de Entidade e Relacionamento.....	60
Figura 14 - Fluxograma.....	61
Figura 15 - Fluxograma Acesso e Cadastro/Login.....	62
Figura 16 - Fluxograma Acesso ao dashboard, Mídias, Perfil.....	63
Figura 17 - Fluxograma Exibição de Mídias na TV.....	64
Figura 18 - Diagrama de Fluxo de Dados.....	65
Figura 19 - Diagrama de Caso de Uso.....	66

LISTA DE TABELAS

Tabela 01 - Requisitos Funcionais.....	25
Tabela 02 - Requisitos Não Funcionais.....	26
Tabela 03 - Requisito Associado 01.....	28
Tabela 04 - Requisito Associado 02.....	28
Tabela 05 - Requisito Associado 03.....	29
Tabela 06 - Requisito Associado 04.....	30
Tabela 07 - Requisito Associado 05.....	30
Tabela 08 - Requisito Associado 06.....	31
Tabela 09 - Requisito Associado 07.....	31
Tabela 10 - Requisito Associado 08.....	32
Tabela 11 - Requisito Associado 09.....	33
Tabela 12 - Requisito Associado 10.....	33
Tabela 13 - Requisito Associado 11.....	34
Tabela 14 - Requisito Associado 12.....	34
Tabela 15 - Requisito Associado 13.....	35
Tabela 16 - Requisito Associado 14.....	36
Tabela 17 - Requisito Associado 15.....	36
Tabela 18 - Requisito Associado 16.....	37
Tabela 19 - Requisito Associado 17.....	37
Tabela 20 - Requisito Associado 18.....	38
Tabela 21 - Requisito Associado 19.....	39
Tabela 22 - Requisito Associado 20.....	39
Tabela 23 - Requisito Associado 21.....	40
Tabela 24 - Requisito Associado 22.....	41
Tabela 25 - Requisito Associado 23.....	42
Tabela 26 - Requisito Associado 24.....	42
Tabela 27 - Requisito Associado 25.....	43
Tabela 28 - Requisito Associado 26.....	44
Tabela 29 - Requisito Associado 27.....	45
Tabela 30 - Requisito Associado 28.....	46
Tabela 31 - Requisito Associado 29.....	46
Tabela 32 - Requisito Associado 30.....	47
Tabela 33 - User Story 1 Cadastro de Mídia.....	48
Tabela 34 - User Story 2 Criação de Campanha.....	48
Tabela 35 - User Story 3 Visualização de Dashboard.....	48
Tabela 36 - User Story 4 Remoção de Mídia.....	49
Tabela 37 - User Story 5 Pesquisa de Campanhas.....	49
Tabela 38 - User Story 6 Upload de Arquivo em Massa.....	49
Tabela 39 - User Story 7 Relatórios de Exibição.....	49

SUMÁRIO

RESUMO.....	3
ABSTRACT.....	4
LISTA DE FIGURAS.....	5
LISTA DE TABELAS.....	6
1. INTRODUÇÃO.....	12
2. MISSÃO, VISÃO E VALORES.....	13
2.1. Missão.....	13
2.2. Visão.....	13
2.3. Valores.....	13
3. 5W2H.....	14
3.1. What? (O que é?).....	14
3.2. Why? (Por Que?).....	14
3.3. Where? (Onde?).....	14
3.4. Who? (Quem?).....	14
3.5. When? (Quando?).....	15
3.6. How? (Como?).....	15
3.7. How Much? (Quanto Custa?).....	15
4. ARQUITETURA, FERRAMENTAS E TECNOLOGIAS.....	16
4.1. Ferramentas e Ambiente de desenvolvimento.....	16
4.2. Tecnologias.....	16
4.2.1. Next.Js.....	16
4.2.2. Node.Js.....	17
4.2.3. SupaBase.....	17
4.3. Padrão de Arquitetura (MVC).....	18
4.3.1. Model (Modelo).....	18
4.3.2. View (Visualização).....	18
4.3.3. Controller (Controlador).....	19
4.4. Deploy.....	20
4.4.1. Vercel.....	20
4.4.2. Render.....	20
5. DESIGN E ESTRUTURA DO PRODUTO.....	21
5.1. SiteMap.....	21
5.2. Requisitos Funcionais.....	25
5.3. Requisitos Não Funcionais.....	26
5.4. Requisitos Associados.....	28
5.5. User Story.....	48
5.6. Protótipo.....	50

6. DIAGRAMAS	59
6.1. Diagrama de Classe.....	59
6.2. Diagrama de Entidade e Relacionamento (DER).....	60
6.3. Fluxograma.....	61
6.4. Diagrama de Fluxo de Dados.....	65
6.5. Diagrama de Caso de Uso.....	66
CONCLUSÃO	69
REFERÊNCIAS	70

1. INTRODUÇÃO

O VIU é um sistema inovador de Digital Signage desenvolvido para modernizar e simplificar a forma como empresas se comunicam com seu público. A proposta do projeto é oferecer uma plataforma acessível e intuitiva que permita exibir conteúdos digitais em TVs de forma automatizada, substituindo cartazes, banners e cardápios impressos por uma solução interativa, sustentável e dinâmica.

Por meio do sistema, o usuário pode criar uma conta, montar grupos de mídias e gerar códigos únicos que conectam suas produções diretamente à TV, bastando apenas acessar uma URL no navegador do televisor. Dessa forma, o VIU elimina processos manuais e oferece uma experiência fluida tanto para quem administra os conteúdos quanto para quem os consome.

Com um design limpo e responsivo, o sistema foi pensado para garantir compatibilidade com diferentes modelos de TVs, inclusive aquelas que se tornam “smart” por meio de dispositivos externos como o Fire Stick. Além disso, a arquitetura do projeto segue o padrão MVC (Model-View-Controller), o que assegura organização, escalabilidade e facilidade de manutenção.

O VIU também conta com um dashboard completo, que exibe métricas relevantes sobre o desempenho das mídias, engajamento e tempo de exibição, ajudando o usuário a entender melhor o impacto de suas comunicações. Tudo isso reforça o compromisso da plataforma com a eficiência, a inovação e a melhoria da experiência do cliente.

Mais do que um produto tecnológico, o VIU representa uma ferramenta de transformação digital para pequenos e médios empreendedores, oferecendo uma maneira prática de atrair a atenção dos consumidores, fortalecer a identidade visual do negócio e reduzir custos operacionais, ao mesmo tempo em que contribui para um ambiente mais sustentável.

2. MISSÃO, VISÃO E VALORES

2.1. Missão

Disponibilizar e facilitar a veiculação de propagandas, oferecendo ferramentas simples, acessíveis e eficazes que aumentam a visibilidade das marcas e geram valor para clientes, parceiros e sociedade.

2.2. Visão

Ser referência nacional em soluções acessíveis e criativas de publicidade, transformando a forma como empresas e pessoas se conectam por meio de anúncios.

2.3. Valores

- Inovação: buscamos constantemente novas formas de comunicar e conectar.
- Simplicidade: tornar o processo de anunciar fácil e descomplicado.
- Transparência: atuar com clareza e confiança em todas as relações.
- Acessibilidade: garantir que negócios de qualquer porte possam anunciar.
- Impacto positivo: gerar valor real para clientes, parceiros e a comunidade.

3. 5W2H

3.1. What? (O que é?)

Desenvolver um sistema de gestão de conteúdo digital para substituir cartazes e placas de promoção em lojas. O sistema deve permitir que o lojista crie, agende e exiba promoções em telas digitais (TVs ou monitores), eliminando a necessidade de material impresso.

3.2. Why? (Por Que?)

- Reduzir custos e tempo com impressão e produção de cartazes.
- Aumentar a flexibilidade e agilidade das campanhas de promoção, permitindo que o lojista altere preços e ofertas em tempo real.
- Modernizar a comunicação visual do ponto de venda, tornando-a mais atrativa para os clientes.

3.3. Where? (Onde?)

- Plataforma de hospedagem: O sistema será uma plataforma web hospedada na nuvem (AWS, Google Cloud ou Azure) para garantir acessibilidade e escalabilidade.
- Mercado-alvo: Inicialmente, o foco será em lojas de varejo nas grandes cidades, com potencial de expansão para outras regiões e segmentos.

3.4. Who? (Quem?)

- Time de Desenvolvimento: Desenvolvedores front-end, back-end e UI/UX designers para construir a plataforma.
- Time de Vendas e Marketing: Responsáveis por vender o sistema e criar estratégias para alcançar lojistas de diversos segmentos (varejo, farmácias, restaurantes, etc.).
- Público-alvo: Proprietários de lojas de pequeno e médio porte que desejam modernizar suas operações.

3.5. When? (Quando?)

- Fase 1 (MVP - Produto Mínimo Viável): Lançamento de uma versão básica do sistema em 4 semanas. Esta versão deve incluir as funcionalidades essenciais de criação, upload e agendamento de conteúdo.
- Fase 2 (Melhorias e Novas Funcionalidades): Implementar novas funcionalidades, como relatórios de desempenho e templates mais elaborados, nos 1 meses seguintes ao lançamento do MVP.
- Fase 3 (Escalabilidade): O sistema deve estar pronto para atender um grande volume de clientes em 1 ano.

3.6. How? (Como?)

- Desenvolvimento Ágil: Usar a metodologia Scrum para dividir o projeto em ciclos curtos (sprints), garantindo entregas contínuas e a capacidade de adaptação às necessidades do mercado.
- **Tecnologias:**
 - **Frontend:** Next.js para criar uma interface de usuário intuitiva.
 - **Backend:** Node.js, para a lógica do sistema e a gestão de dados.
 - **Banco de Dados:** PostgreSQL do Supabase para armazenar informações de usuários e conteúdo.
- **Modelo de Negócio:** Venda de licenças ou assinatura mensal (SaaS - Software as a Service).

3.7. How Much? (Quanto Custa?)

- **Estimativa de Custo Inicial:** O orçamento inicial para desenvolvimento e lançamento do MVP é de aproximadamente R\$3.000,00, cobrindo despesas com desenvolvedores, licenças de software e infraestrutura.
- **Custo por Cliente:** O custo para servir cada cliente individual será baixo, visto que o sistema é totalmente digital e escalável. O foco é garantir um modelo de assinatura acessível para o pequeno varejista.

4. ARQUITETURA, FERRAMENTAS E TECNOLOGIAS

4.1. Ferramentas e Ambiente de desenvolvimento

Para garantir um fluxo de trabalho eficiente, seguro e moderno, foram utilizadas as seguintes ferramentas de desenvolvimento e ambiente:

- **Editor de Código: Visual Studio Code (VS Code)** foi o editor principal, escolhido por seu ecossistema robusto com terminal integrado.
- **Controle de Versão: Git** foi utilizado para o versionamento local. O **GitHub** foi usado como repositório centralizado na nuvem, facilitando a integração contínua e o deploy automatizado.
- **Gerenciador de Pacotes: npm** (Node Package Manager) foi usado para gerenciar todas as dependências de bibliotecas, tanto do backend quanto do frontend.
- **Testes de API: Postman** foi uma ferramenta crucial durante o desenvolvimento do backend para testar e validar cada endpoint da API (CRUD, Auth, Analytics) de forma isolada, antes da integração com o frontend.
- **Ambiente de Runtime: O Node.js** serviu como o ambiente de execução local para ambos os servidores de desenvolvimento (Next.js e Node.js/Express).

4.2. Tecnologias

A arquitetura da aplicação é baseada em um stack desacoplado, separando o frontend, e backend (API) e os serviços de dados (BaaS).

4.2.1. Next.Js

O frontend (a aplicação do lado do cliente) foi construído com **Next.js 14**, um framework React de nível de produção. Ele foi escolhido pelas seguintes razões:

- **App Router:** Utilizamos o moderno App Router para criar rotas baseadas em sistema de arquivos, incluindo layouts aninhados (para o dashboard) e rotas de grupo ((app)).
- **Renderização no Cliente (CSR):** Quase toda a aplicação (Login, Dashboard, TV, Analytics) é renderizada no lado do cliente ('use client'), pois depende de estado (Hooks) e interações do usuário.

- **CSS Modules:** Após desafios de configuração com outras ferramentas, optamos pela solução nativa de estilo do Next.js, o CSS Modules, garantindo estilos encapsulados por componente e um build estável.
- **TypeScript:** O projeto foi escrito inteiramente em TypeScript, garantindo segurança de tipos e melhor manutenibilidade do código.

4.2.2. Node.Js

O backend da aplicação é uma **API RESTful** customizada, construída sobre o runtime **Node.js**, utilizando o framework **Express.js**.

Esta API serve como o "meio-campo" seguro e inteligente entre o cliente e o banco de dados. Ela é responsável por toda a lógica de negócios que não deve ser exposta no navegador, como:

- Gerar códigos de compartilhamento numéricos (nanoid).
- Validar a propriedade de um grupo antes de uma exclusão.
- Processar uploads de mídia (multer) antes de enviá-los ao storage.
- Coletar, agregar e fornecer os dados de analytics.

O backend também foi escrito em **TypeScript** para manter a consistência com o frontend.

4.2.3. SupaBase

O Supabase foi escolhido como a plataforma de **Backend-as-a-Service (BaaS)**, fornecendo três serviços cruciais a partir de uma única interface:

- **Database (Postgres):** Um banco de dados relacional Postgres completo, usado para armazenar todos os metadados da aplicação (tabelas: profiles, groups, media, analytics_sessions, analytics_events). Implementamos um "Database Trigger" para sincronizar automaticamente novos usuários da autenticação com a tabela de perfis.
- **Auth:** O Supabase Auth gerenciou todo o fluxo de autenticação, incluindo login por E-mail/Senha e login social (Google OAuth). Ele é responsável por emitir e validar os Tokens JWT (JSON Web Tokens) que protegem nossa API.
- **Storage:** O Supabase Storage foi usado para hospedar fisicamente os arquivos de mídia (imagens e vídeos) enviados pelos usuários, no bucket media_files.

4.3. Padrão de Arquitetura (MVC)

Para o desenvolvimento do backend (API Node.js/Express), foi adotado o padrão de arquitetura **Model-View-Controller (MVC)**. Este padrão é fundamental para organizar o código em aplicações complexas, pois promove uma clara "Separação de Responsabilidades" (Separation of Concerns).

Em vez de ter um único arquivo gigante lidando com rotas, lógica de negócios e acesso ao banco de dados, o MVC divide a aplicação em três componentes principais e independentes. No contexto de uma API RESTful como a deste projeto, o padrão é adaptado da seguinte forma:

- **Model:** Gerencia os dados e a lógica de banco de dados.
- **View:** É a representação dos dados (JSON) enviada de volta ao cliente.
- **Controller:** Recebe a requisição do cliente, processa a lógica de negócios e coordena o Model e a View.

4.3.1. Model (Modelo)

A camada **Model** representa a estrutura de dados da aplicação e é a **única** camada com permissão para "falar" com o banco de dados (neste caso, o Supabase). Ela é responsável por toda a lógica de persistência de dados.

Os arquivos dentro da pasta `/src/models` (como `groupModel.ts`, `userModel.ts`, `analyticsModel.ts`) atuam como nossos Modelos.

- **Exemplo de Função:** A função `createGroup()` (dentro de `groupModel.ts`) contém a consulta SQL (via Supabase) para INSERT um novo grupo na tabela `groups`. Ela também contém a lógica de negócios para gerar um `share_code` único usando `nanoid`. O Controlador não sabe como o código é gerado ou salvo; ele apenas pede ao Modelo para "criar um grupo".

4.3.2. View (Visualização)

Em uma aplicação web tradicional, a View seria um arquivo HTML renderizado. Em uma API RESTful, a View é a representação dos dados que o servidor envia de volta ao cliente.

A View neste projeto é o **JSON** que a API retorna. O frontend (Next.js) é o responsável por "renderizar" esses dados JSON em uma interface gráfica.

- **Exemplo de Função:** Quando um Controlador responde com `res.status(200).json(grupos)`, o array `grupos` (formatado como JSON) é a View que o cliente recebe.
-

4.3.3. Controller (Controlador)

O **Controller** (Controlador) atua como o intermediário central. Ele é o "cérebro" da aplicação que recebe as requisições HTTP, coordena as ações necessárias e envia a resposta.

Os arquivos dentro da pasta `/src/controllers` (como `groupController.ts`) são os nossos Controladores.

- **Exemplo de Fluxo:**
 1. Uma requisição `POST /api/groups` chega do frontend.
 2. O arquivo de Rotas (`groupRoutes.ts`) direciona a requisição para a função `createGroupController`.
 3. O **Controlador** valida os dados da requisição (ex: verifica se `req.body.name` existe).
 4. Ele chama o **Modelo** (ex: `groupModel.createGroup(name, userId)`).
 5. O Modelo salva no banco de dados e retorna o novo grupo.
 6. O Controlador recebe o novo grupo do Modelo e o formata na **View** (JSON), enviando-o de volta com `res.status(201).json(novoGrupo)`.

4.4. Deploy

A aplicação utiliza uma arquitetura de deploy desacoplada, usando as plataformas ideais para cada parte do stack.

4.4.1. Vercel

O frontend (Next.js) foi implantado na **Vercel**, a plataforma otimizada para Next.js.

- Ela foi conectada diretamente ao repositório GitHub (galeria-midia-frontend).
- Cada git push no branch main aciona um deploy automático.
- As variáveis de ambiente públicas (como NEXT_PUBLIC_SUPABASE_ANON_KEY e NEXT_PUBLIC_API_URL) foram configuradas no painel do Vercel para serem injetadas durante o build.

4.4.2. Render

O backend (API Node.js/Express) foi implantado no **Render**.

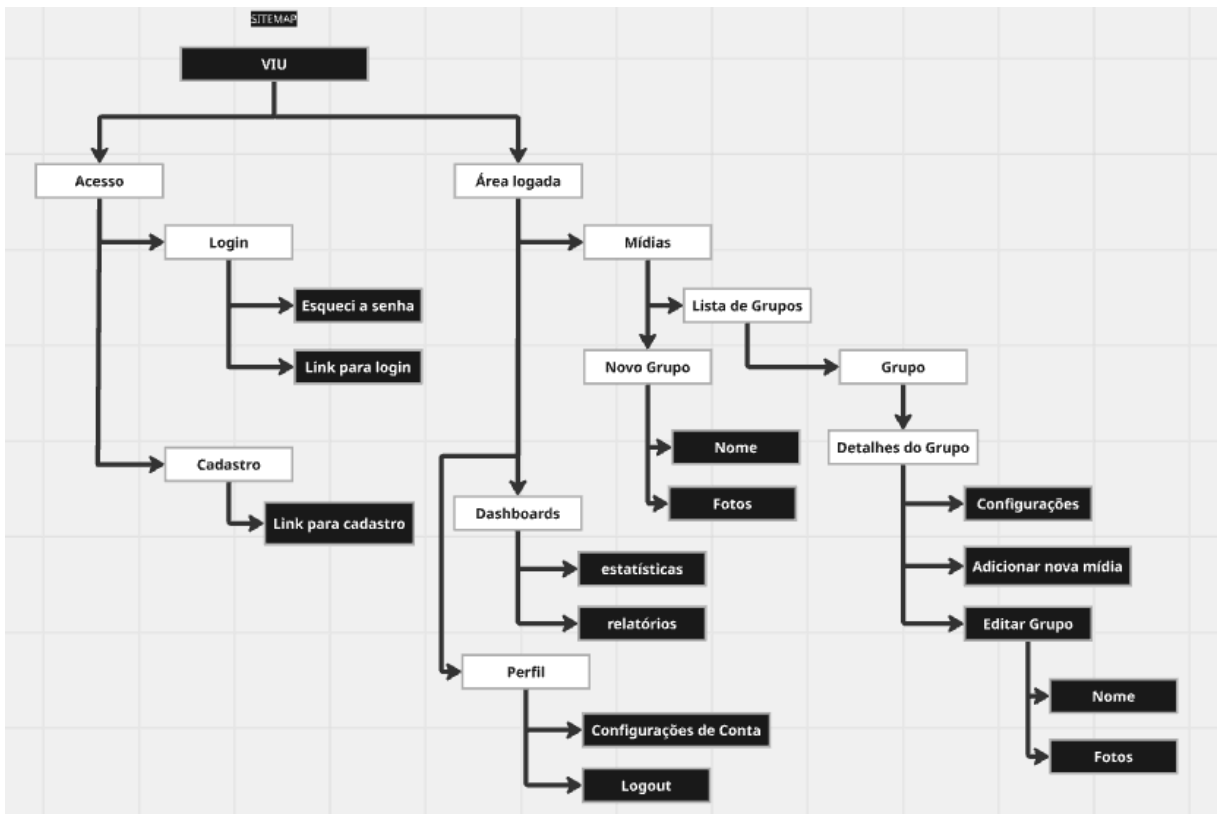
- Ele foi conectado ao repositório GitHub (galeria-midia-backend).
- As variáveis de ambiente secretas (como SUPABASE_SERVICE_ROLE_KEY) e de configuração (como a FRONTEND_URL para o CORS) foram armazenadas de forma segura no painel do Render.

5. DESIGN E ESTRUTURA DO PRODUTO

5.1. SiteMap

O sitemap é uma representação visual da estrutura do sistema, organizada de forma hierárquica. Ele lista todas as páginas e serve como um guia para entender como os conteúdos e funcionalidades estão distribuídos.

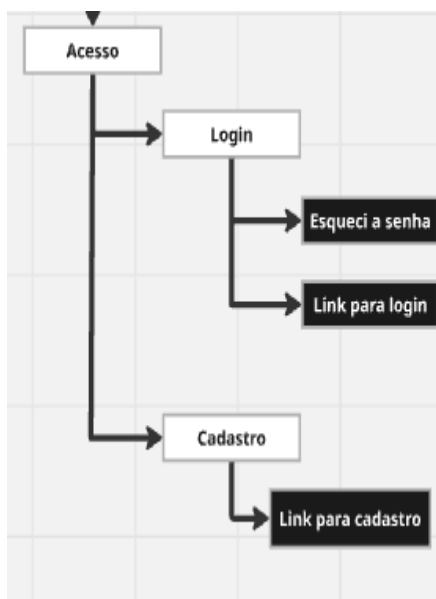
Figura 01 - SiteMap



Fonte: Própria (2025)

Acesso: A tela **Acesso** representa a entrada principal do sistema VIU. A partir dela, o usuário pode realizar o **login** ou efetuar o **cadastro** na plataforma. O objetivo dessa tela é oferecer uma navegação simples e direcionada, facilitando o primeiro contato com o sistema.

Figura 02 - VIU Acesso



Fonte: Própria (2025)

Login: Permite o acesso ao sistema por usuários já cadastrados.

- **Esqueci a senha:** Opção para recuperação de credenciais de acesso.
- **Link para login:** Redireciona o usuário de outras áreas para a tela de login.

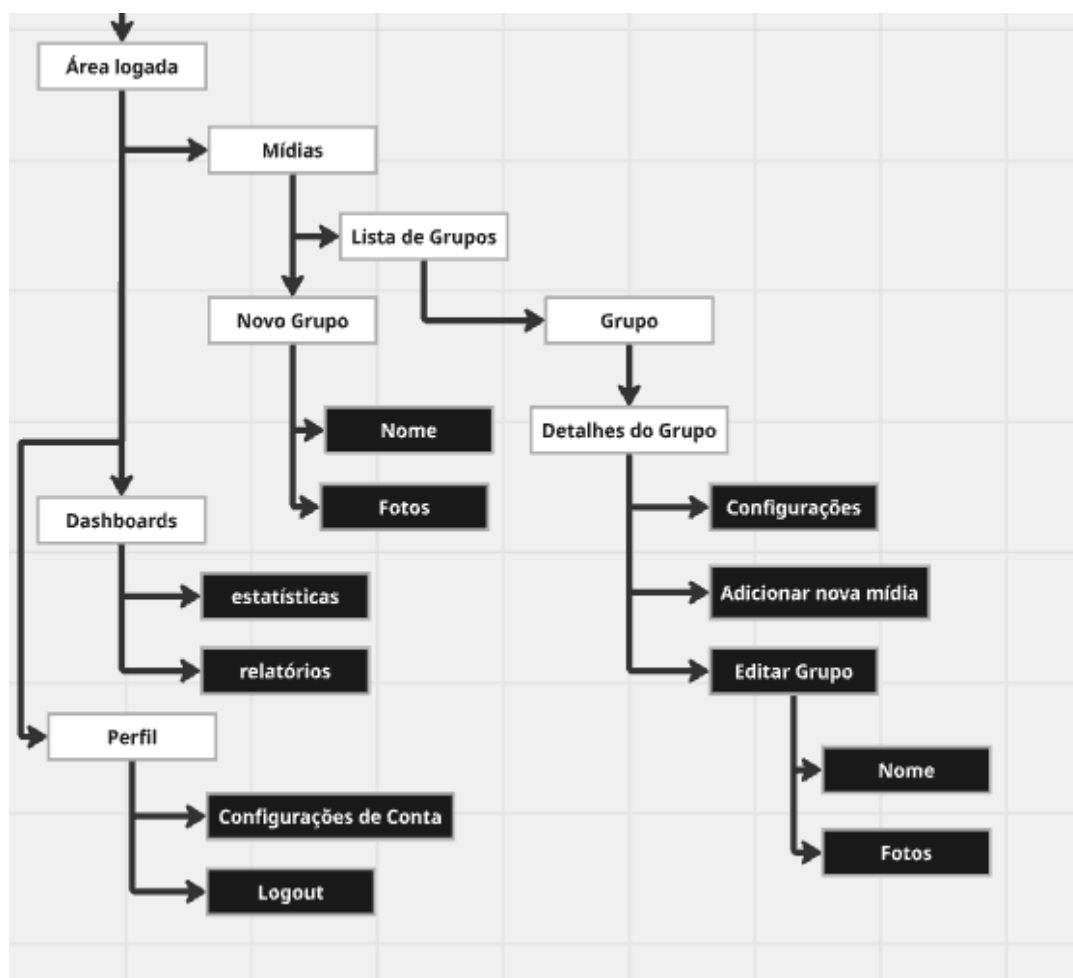
Cadastro: Área destinada ao registro de novos usuários.

- **Link para cadastro:** Atalho para o formulário de criação de conta.

Área Logada:

A **Área Logada** representa o ambiente interno do sistema, acessível apenas a usuários autenticados. Nela, estão reunidas as principais funcionalidades voltadas ao gerenciamento de informações, grupos e mídias. O objetivo dessa seção é oferecer uma navegação prática e organizada, permitindo ao usuário visualizar dados, criar grupos, adicionar mídias e ajustar suas configurações pessoais.

Figura 03 - SiteMap Agenda



Fonte: Própria (2025)

- **Dashboards:**

Apresenta uma visão geral das informações do sistema por meio de **estatísticas** e **relatórios**, auxiliando o usuário na análise e acompanhamento de dados relevantes.

Perfil:

Exibe as informações do usuário logado, permitindo a edição de dados pessoais e acesso às configurações da conta.

- **Configurações de Conta:** área voltada à personalização e atualização dos dados do usuário.
- **Logout:** opção para encerrar a sessão e retornar à tela de acesso.

- **Mídias:**

Seção dedicada ao gerenciamento de grupos e conteúdo multimídia.

- **Lista de Grupos:** exibe todos os grupos cadastrados, permitindo acesso ou criação de novos.
- **Novo Grupo:** permite criar um novo grupo, definindo o **nome** e adicionando **fotos**.

- **Grupo:** exibe os grupos existentes e seus respectivos detalhes.

- **Detalhes do Grupo:** mostra as informações específicas de cada grupo.

- **Configurações:** possibilita ajustes e personalizações do grupo.
- **Adicionar nova mídia:** permite o upload de novos arquivos e imagens.
- **Editar Grupo:** possibilita alterações no nome e nas fotos do grupo.

5.2. Requisitos Funcionais

Tabela 01 - Requisitos Funcionais

REQUISITOS FUNCIONAIS		
RF	Requisito	Descrição
RF1	Cadastro de Usuário	O sistema deve permitir que novos usuários se cadastrem.
RF2	Login de Usuário	O sistema deve permitir que usuários façam login com credenciais.
RF3	Gerenciamento de Perfil	O usuário deve poder editar suas informações de perfil.
RF4	Upload de Mídias	O sistema deve permitir upload de imagens e vídeos.
RF5	Criação de Grupos de Mídias	O usuário deve poder organizar mídias em grupos.
RF6	Geração de Código de Acesso	O sistema deve gerar um código de 5 dígitos para cada grupo.
RF7	Exibição de Mídias na TV	O sistema deve exibir mídias ao inserir o código em uma TV.
RF8	Dashboard	O sistema deve fornecer uma aba de dashboard com informações gerais.
RF9	Assinatura Mensal	O sistema deve controlar o acesso via plano de assinatura.
RF10	Gerenciamento de Sessão	O sistema deve encerrar a sessão após logout ou inatividade.
RF11	Validação de Arquivos	O sistema deve validar formatos e tamanho dos arquivos de mídia.
RF12	Pré-visualização de Mídias	O usuário deve poder visualizar a mídia antes de salvar.
RF13	Ordenação de Mídias	O sistema deve permitir que o usuário defina a ordem de exibição das mídias no grupo.
RF14	Edição de Grupos de Mídias	O usuário pode editar ou excluir grupos criados.
RF15	Histórico de Exibições	O sistema deve registrar o histórico de mídias exibidas em cada TV.
RF16	Gerenciamento de TVs Conectadas	O usuário deve poder visualizar em quais TVs seus grupos estão sendo exibidos.

RF17	Controle de Tempo de Exibição	O usuário deve poder definir o tempo de exibição de cada mídia.
RF18	Pausa e Retomada de Exibição	O sistema deve permitir pausar e retomar a exibição de mídias.
RF19	Notificações	O sistema deve notificar o usuário sobre eventos importantes (assinatura, falha, expiração).
RF20	Filtros de Mídias	O usuário deve poder filtrar mídias por tipo, data ou nome.
RF21	Busca de Mídias	O sistema deve permitir busca de mídias por palavra-chave.
RF22	Troca de Senha	O usuário deve poder alterar sua senha de acesso.
RF23	Recuperação de Senha	O sistema deve permitir recuperação de senha via e-mail.
RF24	Plano de Assinatura	O sistema deve oferecer diferentes planos de assinatura (ex: básico, premium).
RF25	Cancelamento de Assinatura	O usuário deve poder cancelar sua assinatura pelo sistema.

Fonte: Própria (2025)

5.3. Requisitos Não Funcionais

Tabela 02 - Requisitos Não Funcionais

REQUISITOS NÃO FUNCIONAIS		
RNF	Requisito	Descrição
RNF1	Desempenho	O sistema deve responder às requisições em até 2 segundos.
RNF2	Escalabilidade	O sistema deve suportar aumento de usuários sem perda de desempenho.
RNF3	Segurança	Todas as senhas devem ser armazenadas criptografadas.
RNF4	Autenticação Segura	O login deve usar o protocolo HTTPS.
RNF5	Disponibilidade	O sistema deve estar disponível 99,5% do tempo.

RNF6	Backup	Deve ser realizado backup diário dos dados.
RNF7	Recuperação de Desastres	O sistema deve restaurar dados em até 2h após falhas críticas.
RNF8	Usabilidade	A interface deve ser intuitiva e fácil de usar.
RNF9	Documentação	O sistema deve ter documentação técnica e de uso.
RNF10	Compatibilidade	O sistema deve rodar nos principais navegadores modernos.
RNF11	Portabilidade	Deve funcionar em diferentes sistemas operacionais.
RNF12	Responsividade	O layout deve se adaptar a dispositivos móveis e desktops.
RNF13	Internacionalização	Deve permitir suporte a múltiplos idiomas.
RNF14	Padrões de Código	O desenvolvimento deve seguir boas práticas de clean code.
RNF15	Testabilidade	O sistema deve permitir automação de testes unitários e integrados.
RNF16	Logs de Erro	Todas as falhas devem ser registradas em log.
RNF17	Monitoramento	Deve existir monitoramento ativo do servidor.
RNF18	Privacidade	Dados pessoais devem estar em conformidade com a LGPD/GDPR.
RNF19	Escopo de Sessão	Sessões inativas devem expirar após 30 minutos.
RNF20	Interoperabilidade	Deve ser compatível com APIs externas.
RNF21	Acessibilidade	O sistema deve seguir padrões WCAG para acessibilidade.
RNF22	Atualizações	O sistema deve permitir atualização sem indisponibilidade.
RNF23	Confiabilidade	O sistema deve garantir integridade dos dados.
RNF24	Suporte Técnico	Deve haver canal de suporte em horário comercial.
RNF25	Manutenibilidade	O código deve ser modular para facilitar futuras alterações.

Fonte: Própria (2025)

5.4. Requisitos Associados

Tabela 03 - Requisito Associado 01

Requisitos de Sistemas - Associados		
RF01	Cadastro de Cliente	
Descrição	Permitir o cadastro de um novo cliente com nome, telefone e email (opcional).	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF08	Responsividade	A interface deve ser responsiva e funcionar bem em dispositivos móveis e desktops.
RNF06	Interface Intuitiva	A interface deve ser fácil de usar, com navegação clara e instruções para o usuário quando necessário.
RNF08	Responsividade	A interface deve ser responsiva e funcionar bem em dispositivos móveis e desktops.
RNF13	Conformidade com LGPD	com a Lei Geral de Proteção de Dados, garantindo privacidade.
RNF24	Criptografia em Trânsito	Todas as comunicações entre cliente e servidor devem usar HTTPS para garantir segurança.

Fonte: Própria (2025)

Tabela 04 - Requisito Associado 02

Requisitos de Sistemas - Associados		
RF02	Listagem de Clientes (RF02)	
Descrição	Exibir uma lista de todos os clientes cadastrados, mostrando nome e telefone.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF04	Segurança de Dados	Todos os dados sensíveis (como telefones e emails) devem ser criptografados no banco de

		dados.
RNF10	Backup de Dados	O sistema deve realizar backups automáticos diários do banco de dados.
RNF27	Suporte a Grandes Volumes de Dados	O sistema deve suportar até 10.000 registros em cada tabela sem perda de desempenho.

Fonte: Própria (2025)

Tabela 05 - Requisito Associado 03

Requisitos de Sistemas - Associados		
RF03	Visualização de Detalhes do Cliente	
Descrição	Permitir a visualização dos detalhes completos de um cliente específico.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF04	Segurança de Dados	Todos os dados sensíveis (como telefones e emails) devem ser criptografados no banco de dados.
RNF05	Autenticação e Autorização	O sistema deve usar autenticação baseada em tokens (ex.: JWT) para proteger as rotas da API.
RNF10	Backup de Dados	O sistema deve realizar backups automáticos diários do banco de dados.
RNF27	Suporte a Grandes Volumes de Dados	O sistema deve suportar até 10.000 registros em cada tabela sem perda de desempenho.

Fonte: Própria (2025)

Tabela 06 - Requisito Associado 04

Requisitos de Sistemas - Associados		
RF04	Edição de Cliente	
Descrição	Possibilitar a edição dos dados de um cliente existente.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF04	Segurança de Dados	Todos os dados sensíveis (como telefones e emails) devem ser criptografados no banco de dados.
RNF06	Interface Intuitiva	A interface deve ser fácil de usar, com navegação clara e instruções para o usuário quando necessário.
RNF20	Consistência de Dados	O banco de dados deve garantir a integridade referencial e consistência das transações.
RNF25	Auditoria de Ações	O sistema deve registrar todas as ações de criação, edição e exclusão para auditoria.

Fonte: Própria (2025)

Tabela 07 - Requisito Associado 05

Requisitos de Sistemas - Associados		
RF05	Exclusão de Cliente	
Descrição	Permitir a exclusão de um cliente, desde que não tenha animais ou agendamentos associados.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF06	Interface Intuitiva	A interface deve ser fácil de usar, com navegação clara e instruções para o usuário quando necessário.
RNF20	Consistência de Dados	O banco de dados deve garantir a integridade referencial e consistência das transações.
RNF25	Auditoria de Ações	O sistema deve registrar todas as ações de criação, edição e exclusão para auditoria.

Fonte: Própria (2025)

Tabela 08 - Requisito Associado 06

Requisitos de Sistemas - Associados		
RF06	Cadastro de Animal	
Descrição	Permitir o cadastro de um novo animal associado a um cliente, com nome, tipo, raça, idade e observações.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF01	Tempo de Resposta da API	A API deve responder às requisições em menos de 2 segundos para operações de leitura e escrita.
RNF02	Disponibilidade do Sistema	O sistema deve estar disponível 99,9% do tempo, com downtime máximo de 8 horas por ano.
RNF03	Escalabilidade	O sistema deve suportar até 1.000 clientes e 5.000 agendamentos sem degradação de desempenho.
RNF04	Segurança de Dados	Todos os dados sensíveis (como telefones e emails) devem ser criptografados no banco de dados.

Fonte: Própria (2025)

Tabela 09 - Requisito Associado 07

Requisitos de Sistemas - Associados		
RF07	Listagem de Animais	
Descrição	Exibir uma lista de todos os animais cadastrados, mostrando nome, tipo e cliente associado.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF02	Disponibilidade do Sistema	O sistema deve estar disponível 99,9% do tempo, com downtime máximo de 8 horas por ano.
RNF19	Desempenho em Pico	O sistema deve manter o desempenho durante picos de uso (ex.: 100 requisições simultâneas).

RNF22	Notificações em Tempo Real	O sistema deve enviar notificações de confirmação via WhatsApp em menos de 5 segundos após o agendamento.
RNF27	Suporte a Grandes Volumes de Dados	O sistema deve suportar até 10.000 registros em cada tabela sem perda de desempenho.

Fonte: Própria (2025)

Tabela 10 - Requisito Associado 08

Requisitos de Sistemas - Associados		
RF08	Visualização de Detalhes do Animal	
Descrição	Permitir a visualização dos detalhes completos de um animal específico.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF01	Tempo de Resposta da API	A API deve responder às requisições em menos de 2 segundos para operações de leitura e escrita.
RNF06	Interface Intuitiva	A interface deve ser fácil de usar, com navegação clara e instruções para o usuário quando necessário.
RNF08	Responsividade	A interface deve ser responsiva e funcionar bem em dispositivos móveis e desktops.
RNF04	Segurança de Dados	Todos os dados sensíveis (como telefones e emails) devem ser criptografados no banco de dados.

Fonte: Própria (2025)

Tabela 11 - Requisito Associado 09

Requisitos de Sistemas - Associados		
RF09	Edição de Animal	
Descrição	Possibilitar a edição dos dados de um animal existente.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF02	Disponibilidade do Sistema	O sistema deve estar disponível 99,9% do tempo, com downtime máximo de 8 horas por ano.
RNF10	Backup de Dados	O sistema deve realizar backups automáticos diários do banco de dados.
RNF19	Desempenho em Pico	O sistema deve manter o desempenho durante picos de uso (ex.: 100 requisições simultâneas).
RNF30	Suporte a Atualizações Sem Downtime	O sistema deve permitir atualizações de software sem interromper o serviço para os usuários.

Fonte: Própria (2025)

Tabela 12 - Requisito Associado 10

Requisitos de Sistemas - Associados		
RF10	Exclusão de Animal	
Descrição	Permitir a exclusão de um animal, desde que não tenha agendamentos associados.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF25	Desempenho em Pico	O sistema deve manter o desempenho durante picos de uso (ex.: 100 requisições simultâneas).
RNF20	Consistência de Dados	O banco de dados deve garantir a integridade referencial e consistência das transações.

RNF24	Criptografia em Trânsito	Todas as comunicações entre cliente e servidor devem usar HTTPS para garantir segurança.
RNF25	Auditoria de Ações	O sistema deve registrar todas as ações de criação, edição e exclusão para auditoria.

Fonte: Própria (2025)

Tabela 13 - Requisito Associado 11

Requisitos de Sistemas - Associados		
RF11	Listagem de Serviços	
Descrição	Exibir a lista de serviços disponíveis (banho, tosa, veterinário).	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF02	Disponibilidade do Sistema	O sistema deve estar disponível 99,9% do tempo, com downtime máximo de 8 horas por ano.
RNF03	Escalabilidade	O sistema deve suportar até 1.000 clientes e 5.000 agendamentos sem degradação de desempenho.
RNF25	Auditoria de Ações	O sistema deve registrar todas as ações de criação, edição e exclusão para auditoria.

Fonte: Própria (2025)

Tabela 14 - Requisito Associado 12

Requisitos de Sistemas - Associados		
RF12	Cadastro de Agendamento	
Descrição	Permitir o cadastro de um novo agendamento, selecionando cliente, animal, serviço, data e hora.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF02	Disponibilidade do Sistema	O sistema deve estar disponível 99,9% do tempo, com downtime máximo de 8 horas por ano.
RNF03	Escalabilidade	O sistema deve suportar até 1.000 clientes e 5.000

		agendamentos sem degradação de desempenho.
RNF20	Consistência de Dados	O banco de dados deve garantir a integridade referencial e consistência das transações.
RNF21	Suporte a Fusos Horários	O sistema deve lidar corretamente com diferentes fusos horários para agendamentos.
RNF25	Auditoria de Ações	O sistema deve registrar todas as ações de criação, edição e exclusão para auditoria.

Fonte: Própria (2025)

Tabela 15 - Requisito Associado 13

Requisitos de Sistemas - Associados		
RF13	Listagem de Agendamentos	
Descrição	Exibir uma lista de todos os agendamentos, mostrando data, serviço, animal e status.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF10	Backup de Dados	O sistema deve realizar backups automáticos diários do banco de dados.
RNF21	Suporte a Fusos Horários	O sistema deve lidar corretamente com diferentes fusos horários para agendamentos.
RNF25	Auditoria de Ações	O sistema deve registrar todas as ações de criação, edição e exclusão para auditoria.
RNF27	Suporte a Grandes Volumes de Dados	O sistema deve suportar até 10.000 registros em cada tabela sem perda de desempenho.

Fonte: Própria (2025)

Tabela 16 - Requisito Associado 14

Requisitos de Sistemas - Associados		
RF14	Visualização de Detalhes do Agendamento	
Descrição	Permitir a visualização dos detalhes completos de um agendamento específico.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF01	Disponibilidade do Sistema	O sistema deve estar disponível 99,9% do tempo, com downtime máximo de 8 horas por ano.
RNF20	Consistência de Dados	O banco de dados deve garantir a integridade referencial e consistência das transações.
RNF25	Auditoria de Ações	O sistema deve registrar todas as ações de criação, edição e exclusão para auditoria.

Fonte: Própria (2025)

Tabela 17 - Requisito Associado 15

Requisitos de Sistemas - Associados		
RF15	Edição de Agendamento	
Descrição	Possibilitar a edição de um agendamento existente (data, hora, serviço, etc.).	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF01	Disponibilidade do Sistema	O sistema deve estar disponível 99,9% do tempo, com downtime máximo de 8 horas por ano.
RNF20	Consistência de Dados	O banco de dados deve garantir a integridade referencial e consistência das transações.
RNF21	Suporte a Fusos Horários	O sistema deve lidar corretamente com diferentes fusos horários para agendamentos.
RNF25	Auditoria de Ações	O sistema deve registrar todas as ações de criação, edição e exclusão para auditoria.

Fonte: Própria (2025)

Tabela 18 - Requisito Associado 16

Requisitos de Sistemas - Associados		
RF16	Cancelamento de Agendamento	
Descrição	Permitir o cancelamento de um agendamento, atualizando seu status.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF16	Tolerância a Falhas	O sistema deve continuar funcionando mesmo em caso de falha parcial (ex.: indisponibilidade de um serviço).
RNF20	Consistência de Dados	O banco de dados deve garantir a integridade referencial e consistência das transações.
RNF21	Suporte a Fusos Horários	O sistema deve lidar corretamente com diferentes fusos horários para agendamentos.
RNF25	Auditoria de Ações	O sistema deve registrar todas as ações de criação, edição e exclusão para auditoria.

Fonte: Própria (2025)

Tabela 19 - Requisito Associado 17

Requisitos de Sistemas - Associados		
RF17	Envio de Confirmação via WhatsApp	
Descrição	Enviar automaticamente uma mensagem de confirmação ao cliente via WhatsApp após o agendamento.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF04	Segurança de Dados	Todos os dados sensíveis (como telefones e emails) devem ser criptografados no banco de dados.
RNF13	Conformidade com LGPD	O sistema deve estar em conformidade com a Lei Geral de Proteção de Dados, garantindo privacidade.

RNF22	Notificações em Tempo Real	O sistema deve enviar notificações de confirmação via WhatsApp em menos de 5 segundos após o agendamento.
RNF25	Auditoria de Ações	O sistema deve registrar todas as ações de criação, edição e exclusão para auditoria.

Fonte: Própria (2025)

Tabela 20 - Requisito Associado 18

Requisitos de Sistemas - Associados		
RF18	Validação de Dados do Cliente	
Descrição	Validar os dados inseridos no cadastro de cliente (ex.: telefone válido).	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF04	Segurança de Dados	Todos os dados sensíveis (como telefones e emails) devem ser criptografados no banco de dados.
RNF05	Autenticação e Autorização	O sistema deve usar autenticação baseada em tokens (ex.: JWT) para proteger as rotas da API.
RNF10	Backup de Dados	O sistema deve realizar backups automáticos diários do banco de dados.
RNF23	Limite de Tentativas de Login	Após 5 tentativas falhas de login, a conta deve ser bloqueada temporariamente por 15 minutos.
RNF24	Criptografia em Trânsito	Todas as comunicações entre cliente e servidor devem usar HTTPS para garantir segurança.

Fonte: Própria (2025)

Tabela 21 - Requisito Associado 19

Requisitos de Sistemas - Associados		
RF19	Validação de Dados do Animal	
Descrição	Validar os dados inseridos no cadastro de animal (ex.: tipo e raça).	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF04	Segurança de Dados	Todos os dados sensíveis (como telefones e emails) devem ser criptografados no banco de dados.
RNF10	Backup de Dados	O sistema deve realizar backups automáticos diários do banco de dados.
RNF05	Autenticação e Autorização	O sistema deve usar autenticação baseada em tokens (ex.: JWT) para proteger as rotas da API.
RNF24	Criptografia em Trânsito	Todas as comunicações entre cliente e servidor devem usar HTTPS para garantir segurança.

Fonte: Própria (2025)

Tabela 22 - Requisito Associado 20

Requisitos de Sistemas - Associados		
RF20	Validação de Agendamento	
Descrição	Garantir que a data e hora do agendamento sejam futuras e não conflitem com outros agendamentos.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF04	Segurança de Dados	Todos os dados sensíveis (como telefones e emails) devem ser criptografados no banco de dados.
RNF05	Autenticação e Autorização	O sistema deve usar autenticação baseada em tokens (ex.: JWT) para proteger as rotas da API.
RNF24	Criptografia em Trânsito	Todas as comunicações entre cliente e servidor devem usar HTTPS para garantir segurança.

RNF22	Notificações em Tempo Real	O sistema deve enviar notificações de confirmação via WhatsApp em menos de 5 segundos após o agendamento.
RNF27	Suporte a Grandes Volumes de Dados	O sistema deve suportar até 10.000 registros em cada tabela sem perda de desempenho.

Fonte: Própria (2025)

Tabela 23 - Requisito Associado 21

Requisitos de Sistemas - Associados		
RF21	Associação de Animal a Cliente	
Descrição	Garantir que um animal seja associado a um cliente existente ao cadastrar.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF04	Segurança de Dados	Todos os dados sensíveis (como telefones e emails) devem ser criptografados no banco de dados.
RNF12	Logging e Monitoramento	O sistema deve registrar logs de todas as operações críticas e monitorar o uso de recursos.
RNF20	Consistência de Dados	O banco de dados deve garantir a integridade referencial e consistência das transações.
RNF25	Auditoria de Ações	O sistema deve registrar todas as ações de criação, edição e exclusão para auditoria.

Fonte: Própria (2025)

Tabela 24 - Requisito Associado 22

Requisitos de Sistemas - Associados		
RF22	Filtro de Agendamentos por Data	
Descrição	Permitir filtrar a lista de agendamentos por data específica ou intervalo.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF01	Tempo de Resposta da API	A API deve responder às requisições em menos de 2 segundos para operações de leitura e escrita.
RNF06	Interface Intuitiva	A interface deve ser fácil de usar, com navegação clara e instruções para o usuário quando necessário.
RNF08	Responsividade	A interface deve ser responsiva e funcionar bem em dispositivos móveis e desktops.
RNF09	Tempo de Carregamento da Página	As páginas devem carregar em menos de 3 segundos em conexões de internet padrão.
RNF15	Acessibilidade	A interface deve seguir as diretrizes de acessibilidade (WCAG 2.1) para usuários com deficiências.

Fonte: Própria (2025)

Tabela 25 - Requisito Associado 23

Requisitos de Sistemas - Associados		
RF23	Ordenação de Listas	
Descrição	Permitir ordenar as listas de clientes, animais e agendamentos por diferentes critérios (ex.: nome, data).	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF01	Tempo de Resposta da API	A API deve responder às requisições em menos de 2 segundos para operações de leitura e escrita.
RNF06	Interface Intuitiva	A interface deve ser fácil de usar, com navegação clara e instruções para o usuário quando necessário.
RNF08	Responsividade	A interface deve ser responsiva e funcionar bem em dispositivos móveis e desktops.
RNF13	Conformidade com LGPD	com a Lei Geral de Proteção de Dados, garantindo privacidade.
RNF27	Suporte a Grandes Volumes de Dados	O sistema deve suportar até 10.000 registros em cada tabela sem perda de desempenho.

Fonte: Própria (2025)

Tabela 26 - Requisito Associado 24

Requisitos de Sistemas - Associados		
RF24	Busca de Clientes	
Descrição	Implementar uma funcionalidade de busca para encontrar clientes por nome ou telefone.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF01	Tempo de Resposta da API	A API deve responder às requisições em menos de 2 segundos para operações de leitura e escrita.

RNF08	Responsividade	A interface deve ser responsiva e funcionar bem em dispositivos móveis e desktops.
RNF13	Conformidade com LGPD	com a Lei Geral de Proteção de Dados, garantindo privacidade.
RNF20	Consistência de Dados	O banco de dados deve garantir a integridade referencial e consistência das transações.
RNF27	Suporte a Grandes Volumes de Dados	O sistema deve suportar até 10.000 registros em cada tabela sem perda de desempenho.

Fonte: Própria (2025)

Tabela 27 - Requisito Associado 25

Requisitos de Sistemas - Associados		
RF25	Busca de Animais	
Descrição	Implementar uma funcionalidade de busca para encontrar animais por nome ou cliente.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF01	Tempo de Resposta da API	A API deve responder às requisições em menos de 2 segundos para operações de leitura e escrita.
RNF08	Responsividade	A interface deve ser responsiva e funcionar bem em dispositivos móveis e desktops.
RNF13	Conformidade com LGPD	com a Lei Geral de Proteção de Dados, garantindo privacidade.
RNF20	Consistência de Dados	O banco de dados deve garantir a integridade referencial e consistência das transações.
RNF27	Suporte a Grandes Volumes de Dados	O sistema deve suportar até 10.000 registros em cada tabela sem perda de desempenho.

Fonte: Própria (2025)

Tabela 28 - Requisito Associado 26

Requisitos de Sistemas - Associados		
RF26	Busca de Agendamentos	
Descrição	Implementar uma funcionalidade de busca para encontrar agendamentos por data, serviço ou animal.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF01	Tempo de Resposta da API	A API deve responder às requisições em menos de 2 segundos para operações de leitura e escrita.
RNF06	Interface Intuitiva	A interface deve ser fácil de usar, com navegação clara e instruções para o usuário quando necessário.
RNF08	Responsividade	A interface deve ser responsiva e funcionar bem em dispositivos móveis e desktops.
RNF13	Conformidade com LGPD	com a Lei Geral de Proteção de Dados, garantindo privacidade.
RNF20	Consistência de Dados	O banco de dados deve garantir a integridade referencial e consistência das transações.
RNF27	Suporte a Grandes Volumes de Dados	O sistema deve suportar até 10.000 registros em cada tabela sem perda de desempenho.

Fonte: Própria (2025)

Tabela 29 - Requisito Associado 27

Requisitos de Sistemas - Associados		
RF27	Dashboard com Estatísticas	
Descrição	Exibir um dashboard com estatísticas básicas (ex.: número de agendamentos do dia, total de clientes).	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF01	Tempo de Resposta da API	A API deve responder às requisições em menos de 2 segundos para operações de leitura e escrita.
RNF06	Interface Intuitiva	A interface deve ser fácil de usar, com navegação clara e instruções para o usuário quando necessário.
RNF08	Responsividade	A interface deve ser responsiva e funcionar bem em dispositivos móveis e desktops.
RNF13	Conformidade com LGPD	com a Lei Geral de Proteção de Dados, garantindo privacidade.
RNF27	Suporte a Grandes Volumes de Dados	O sistema deve suportar até 10.000 registros em cada tabela sem perda de desempenho.
RNF28	Documentação do Sistema	Deve haver documentação completa para desenvolvedores e usuários finais.

Fonte: Própria (2025)

Tabela 30 - Requisito Associado 28

Requisitos de Sistemas - Associados		
RF28	Geração de Relatório de Agendamentos	
Descrição	Permitir a geração de um relatório em PDF ou CSV com os agendamentos de um período.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF12	Logging e Monitoramento	O sistema deve registrar logs de todas as operações críticas e monitorar o uso de recursos.
RNF08	Responsividade	A interface deve ser responsiva e funcionar bem em dispositivos móveis e desktops.
RNF25	Auditoria de Ações	O sistema deve registrar todas as ações de criação, edição e exclusão para auditoria.
RNF27	Suporte a Grandes Volumes de Dados	O sistema deve suportar até 10.000 registros em cada tabela sem perda de desempenho.
RNF28	Documentação do Sistema	Deve haver documentação completa para desenvolvedores e usuários finais.

Fonte: Própria (2025)

Tabela 31 - Requisito Associado 29

Requisitos de Sistemas - Associados		
RF29	Notificação de Agendamento Próximo	
Descrição	Enviar uma notificação ao cliente via WhatsApp um dia antes do agendamento.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF13	Conformidade com LGPD	O sistema deve estar em conformidade com a Lei Geral de Proteção de Dados, garantindo privacidade.

RNF21	Suporte a Fusos Horários	O sistema deve lidar corretamente com diferentes fusos horários para agendamentos.
RNF22	Notificações em Tempo Real	O sistema deve enviar notificações de confirmação via WhatsApp em menos de 5 segundos após o agendamento.
RNF28	Documentação do Sistema	Deve haver documentação completa para desenvolvedores e usuários finais.

Fonte: Própria (2025)

Tabela 32 - Requisito Associado 30

Requisitos de Sistemas - Associados		
RF30	Autenticação de Usuário	
Descrição	Implementar login para funcionários do petshop, com controle de acesso às funcionalidades.	
Associados (RNF)		
RNF	Requisitos	Descrição
RNF04	Segurança de Dados	Todos os dados sensíveis (como telefones e emails) devem ser criptografados no banco de dados.
RNF05	Autenticação e Autorização	O sistema deve usar autenticação baseada em tokens (ex.: JWT) para proteger as rotas da API.
RNF13	Conformidade com LGPD	com a Lei Geral de Proteção de Dados, garantindo privacidade.
RNF24	Criptografia em Trânsito	Todas as comunicações entre cliente e servidor devem usar HTTPS para garantir segurança.

Fonte: Própria (2025)

5.5. User Story

Tabela 33 - User Story 1 Cadastro de Mídia

Mapa User Story 1	Título: Cadastro de Mídia
História do Usuário:	
Como um gestor de campanhas da VIU,	
Eu quero cadastrar uma nova mídia no sistema,	
Para que eu possa organizá-la e vinculá-la a uma campanha de exibição.	
CrITÉrios de Aceitação:	
Dado que eu esteja na aba de cadastro de mídias,	
Quando eu preencher os campos obrigatórios (nome da mídia, tipo e arquivo de upload) e clicar em "Cadastrar",	
Então a mídia deve ser salva no sistema e uma mensagem de confirmação deve ser exibida.	

Fonte: Própria (2025)

Tabela 34 - User Story 2 Criação de Campanha

Mapa User Story 2	Título: Criação de Campanha
História do Usuário:	
Como um gestor de campanhas da VIU,	
Eu quero criar uma nova campanha,	
Para que eu possa organizar mídias e definir períodos de exibição.	
CrITÉrios de Aceitação:	
Dado que eu esteja na aba de campanhas,	
Quando eu preencher os campos obrigatórios (nome da campanha, datas de início e fim) e clicar em "Criar",	
Então a campanha deve ser salva e exibida na lista de campanhas.	

Fonte: Própria (2025)

Tabela 35 - User Story 3 Visualização de Dashboard

Mapa User Story 3	Título: Visualização de Dashboard
História do Usuário:	
Como um administrador da VIU,	
Eu quero visualizar um dashboard com métricas das campanhas,	
Para que eu possa acompanhar desempenho e engajamento.	
CrITÉrios de Aceitação:	
Dado que eu esteja na página do dashboard,	
Quando eu acessar a aba de métricas,	
Então devo visualizar gráficos e indicadores atualizados das campanhas ativas.	

Fonte: Própria (2025)

Tabela 36 - User Story 4 Remoção de Mídia

Mapa User Story 4	Título: Remoção de Mídia
História do Usuário:	
Como um gestor de mídia,	
Eu quero remover uma mídia do sistema,	
Para manter o banco de mídias organizado.	
Crítérios de Aceitação:	
Dado que eu esteja na lista de mídias,	
Quando eu clicar em "Excluir" em uma mídia,	
Então ela deve ser removida e uma confirmação exibida.	

Fonte: Própria (2025)

Tabela 37 - User Story 5 Pesquisa de Campanhas

Mapa User Story 5	Título: Pesquisa de Campanhas
História do Usuário:	
Como um gestor de campanhas,	
Eu quero pesquisar campanhas por nome ou status,	
Para localizar rapidamente a campanha desejada.	
Crítérios de Aceitação:	
Dado que eu esteja na aba de campanhas,	
Quando eu digitar um termo de pesquisa e clicar em "Buscar",	
Então o sistema deve filtrar as campanhas correspondentes ao termo.	

Fonte: Própria (2025)

Tabela 38 - User Story 6 Upload de Arquivo em Massa

Mapa User Story 6	Título: Upload de Arquivo em Massa
História do Usuário:	
Como um gestor de mídia,	
Eu quero importar várias mídias de uma vez,	
Para agilizar o cadastro de conteúdos.	
Crítérios de Aceitação:	
Dado que eu esteja na aba de mídias,	
Quando eu selecionar múltiplos arquivos e clicar em "Importar",	
Então todos os arquivos válidos devem ser cadastrados e exibidos na lista.	

Fonte: Própria (2025)

Tabela 39 - User Story 7 Relatórios de Exibição

Mapa User Story 7	Título: Relatórios de Exibição
História do Usuário:	
Como um administrador da VIU,	
Eu quero gerar relatórios de exibição de campanhas,	
Para analisar quais mídias tiveram melhor desempenho.	
Crítérios de Aceitação:	
Dado que eu esteja na aba de relatórios,	
Quando eu selecionar período, campanha e clicar em "Gerar Relatório",	
Então o sistema deve exibir os dados de exibição em formato de tabela e gráficos.	

Fonte: Própria (2025)

5.6. Protótipo

O protótipo é uma representação visual mais próxima do produto final. Ele serve para simular a interface real do sistema, com maior nível de detalhamento visual, como cores, tipografia, ícones, espaçamento e elementos interativos.

O protótipo do **VIU: Gestor de Mídia Digital** foi desenvolvido na plataforma **Figma**, uma ferramenta líder de mercado que possibilita a criação de layouts interativos e colaborativos. Com o uso do Figma, foi possível desenhar e validar os fluxos de usuário para as duas principais personas da aplicação: o **Administrador de Conteúdo** (no dashboard de gerenciamento) e o **Visualizador Público** (na tela /tv).

A prototipação permitiu simular a interatividade do slideshow da tela /tv, os modais de upload de mídia e os gráficos do painel de analytics. Isso garantiu uma interface intuitiva, moderna e funcional, focada em um tema escuro (dark mode) coeso — refletindo o compromisso do projeto com a qualidade da experiência do usuário.

Tela Inicial (Home Page):

Figura 04 - Protótipo Tela Inicial



Fonte: Própria (2025)

Cabeçalho (Header): Contém o logo "VIU" (com link para a Home Page) e os botões de navegação "Login" e "Cadastrar".

Seção Hero: A principal chamada visual da página, com uma imagem de fundo (com película escura), o título principal ("Faça seus clientes VEREM suas publicidades") e uma descrição do serviço.

Botão de Ação (CTA): Um botão principal de "Acessar na TV" que direciona o usuário para a tela pública /tv.

Seção "Como Funciona": Uma seção secundária que explica o processo em três etapas (1. Faça Upload, 2. Gere o Código, 3. Exiba na TV), cada uma com seu respectivo ícone e descrição.

Tela de Cadastro:

Figura 05 - Protótipo Tela de Cadastro

cadastro

VIU

Cadastro

Nome:

Email:

Senha:

Cadastrar-se

Ja tem uma contra? Entrar

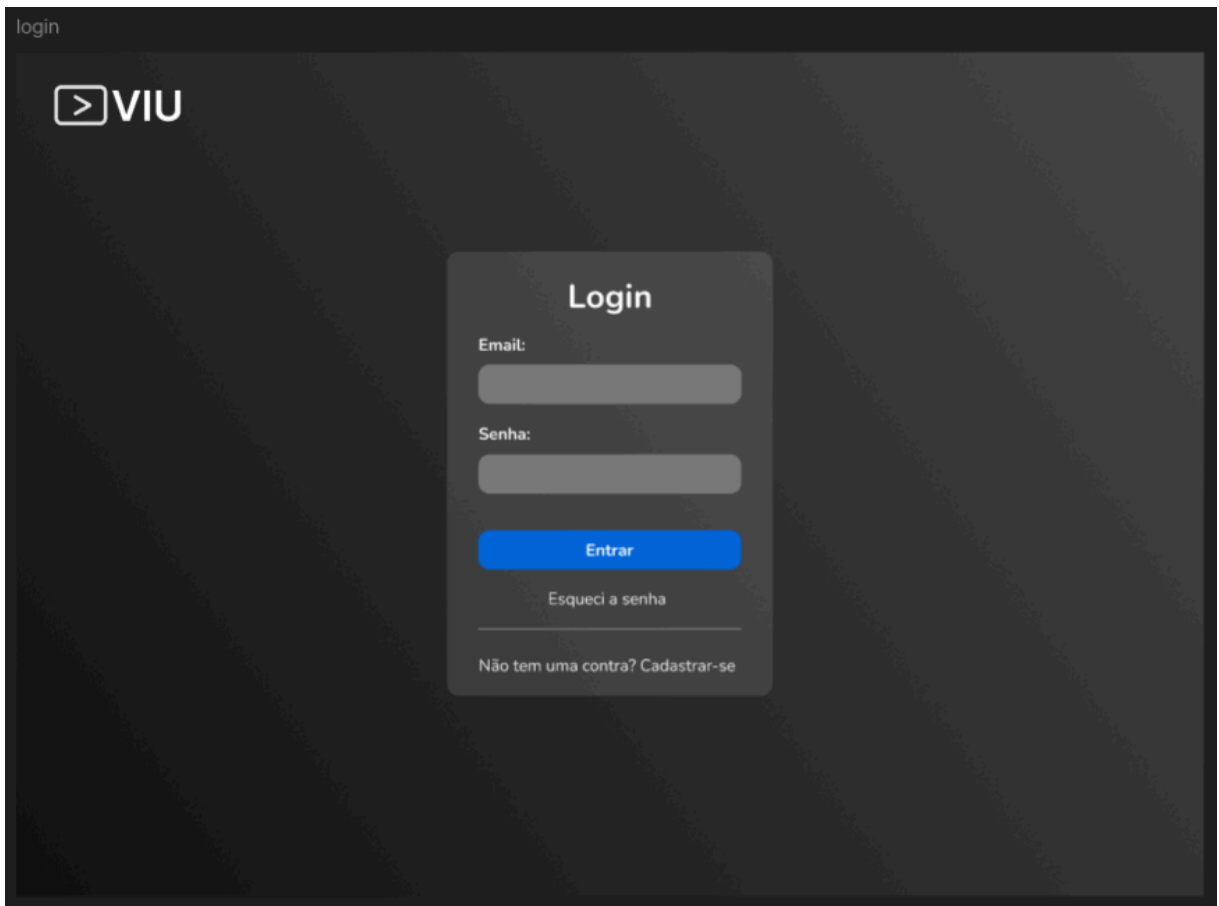
Fonte: Própria (2025)

- Logo "VIU" no cabeçalho, permitindo ao usuário retornar à Tela Inicial.
- Um formulário centralizado para a criação de uma nova conta de usuário, seguindo o tema escuro da aplicação.
- Campos para inserir dados de cadastro: "Nome", "Email" e "Senha".
- Botão de ação principal "Cadastrar-se" para submeter o formulário e criar a conta.

- Link de alternância "Ja tem uma contra? Entrar" para direcionar usuários existentes à tela de Login.

Tela de Login:

Figura 06 - Protótipo Tela de Login

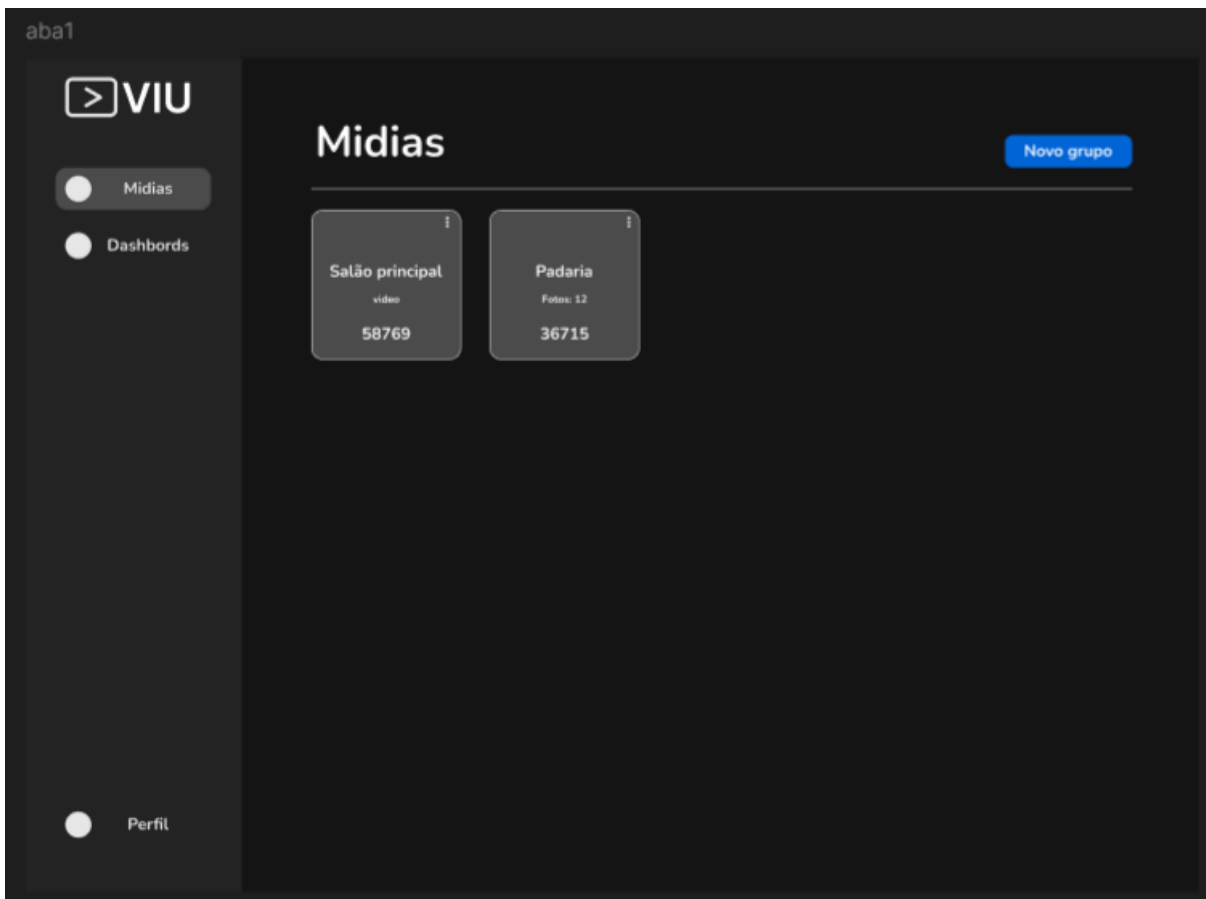


Fonte: Própria (2025)

- Logo "VIU" no cabeçalho para permitir o retorno à Tela Inicial.
- Formulário centralizado para autenticação do usuário.
- Campos para inserir dados de login: "Email" e "Senha".
- Botão de ação principal "Entrar" para concluir o login e acessar o *dashboard*.
- Links secundários para recuperação de senha ("Esqueci a senha") e para alternar para a tela de Cadastro ("Não tem uma contra? Cadastrar-se").

Tela de Mídias (Dashboard Principal):

Figura 07 - Protótipo Tela de Mídias

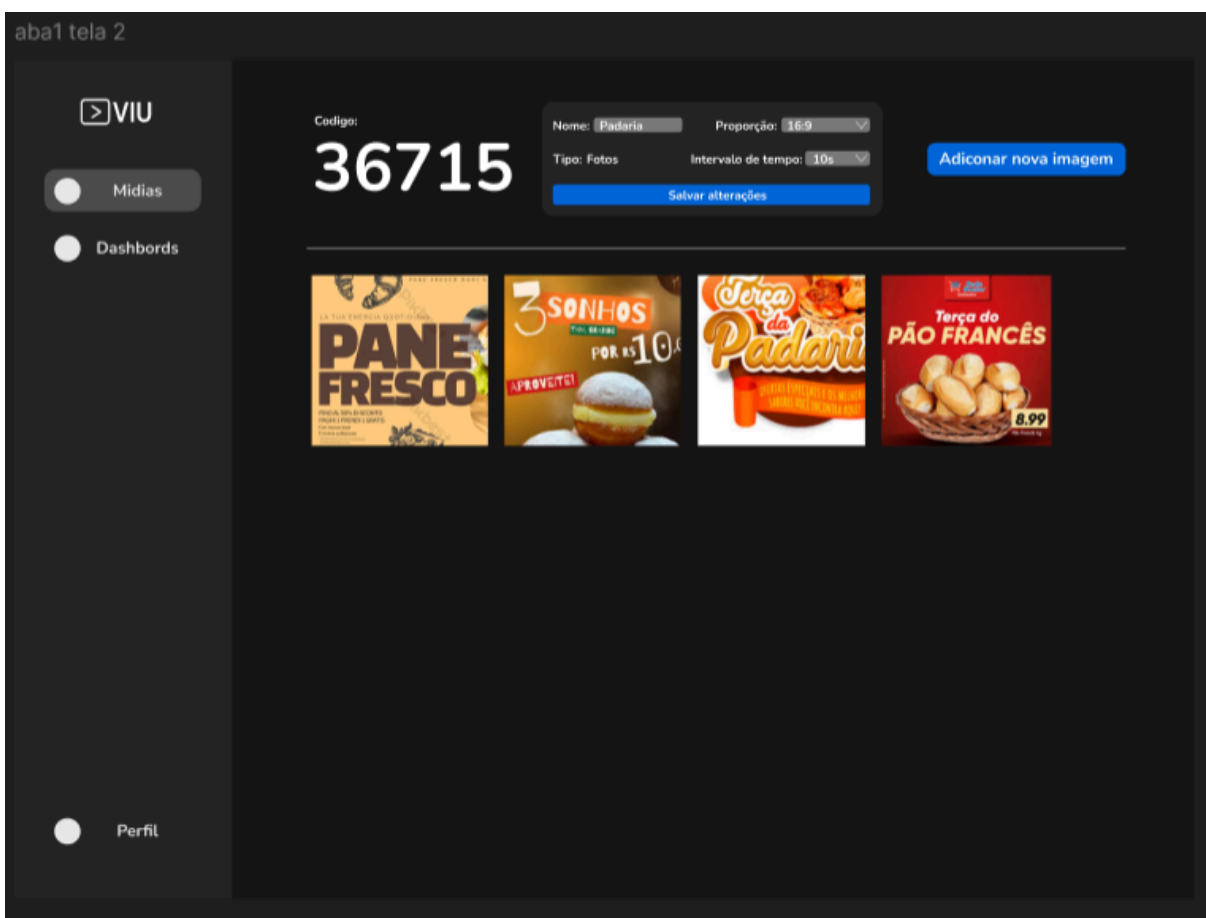


Fonte: Própria (2025)

- Menu de navegação lateral (Sidebar) com o logo "VIU" e os links "Mídias" (ativo), "Dashboards" (Analytics) e "Perfil" (Configuração).
- Cabeçalho da área principal com o título "Mídias" e o botão de ação "Novo grupo".
- Área de conteúdo exibindo os grupos de mídia criados pelo usuário em formato de *cards*.
- Cada *card* (ex: "Salão principal", "Padaria") exibe o nome do grupo, o tipo de conteúdo (vídeo, fotos) e o código de compartilhamento numérico de 5 dígitos.

Tela de Detalhes do Grupo (Gerenciamento de Mídia):

Figura 08 - Protótipo Detalhes do Grupo

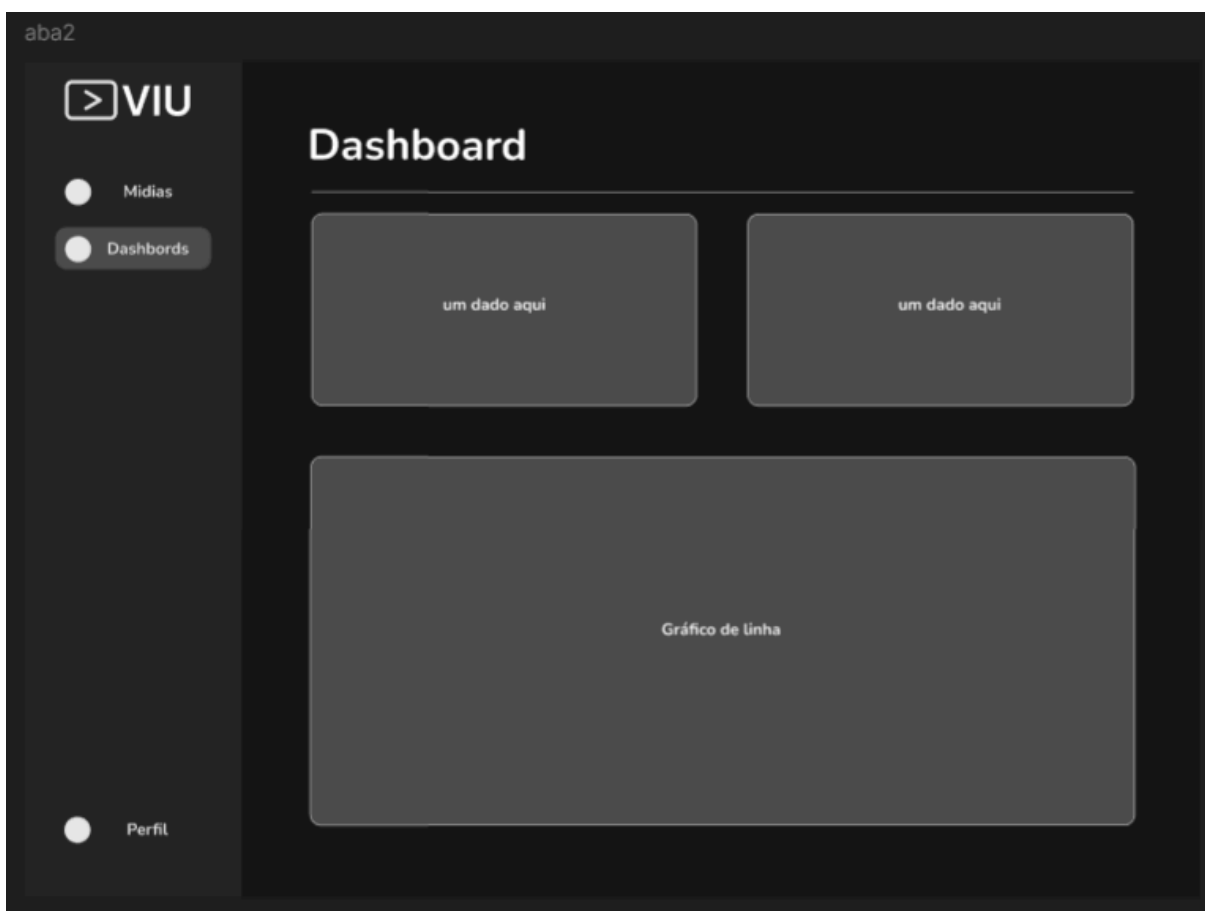


Fonte: Própria (2025)

- A navegação lateral (Sidebar) permanece visível à esquerda.
- O código de compartilhamento de 5 dígitos ("36715") do grupo é exibido com destaque.
- Campos de configuração permitem ao usuário editar o "Nome" do grupo, a "Proporção" da tela (ex: 16:9) e o "Intervalo de tempo" (ex: 10s) para o *slideshow*.
- Um botão "Salvar alterações" para persistir as configurações do grupo.
- Um botão de ação "Adicionar nova imagem" que aciona o *upload* de novas mídias.
- Uma galeria em grade exibe as miniaturas de todas as mídias (imagens ou vídeos) que já foram carregadas neste grupo.

Tela de Dashboard (Analytics):

Figura 09 - Protótipo Tela de Analytics

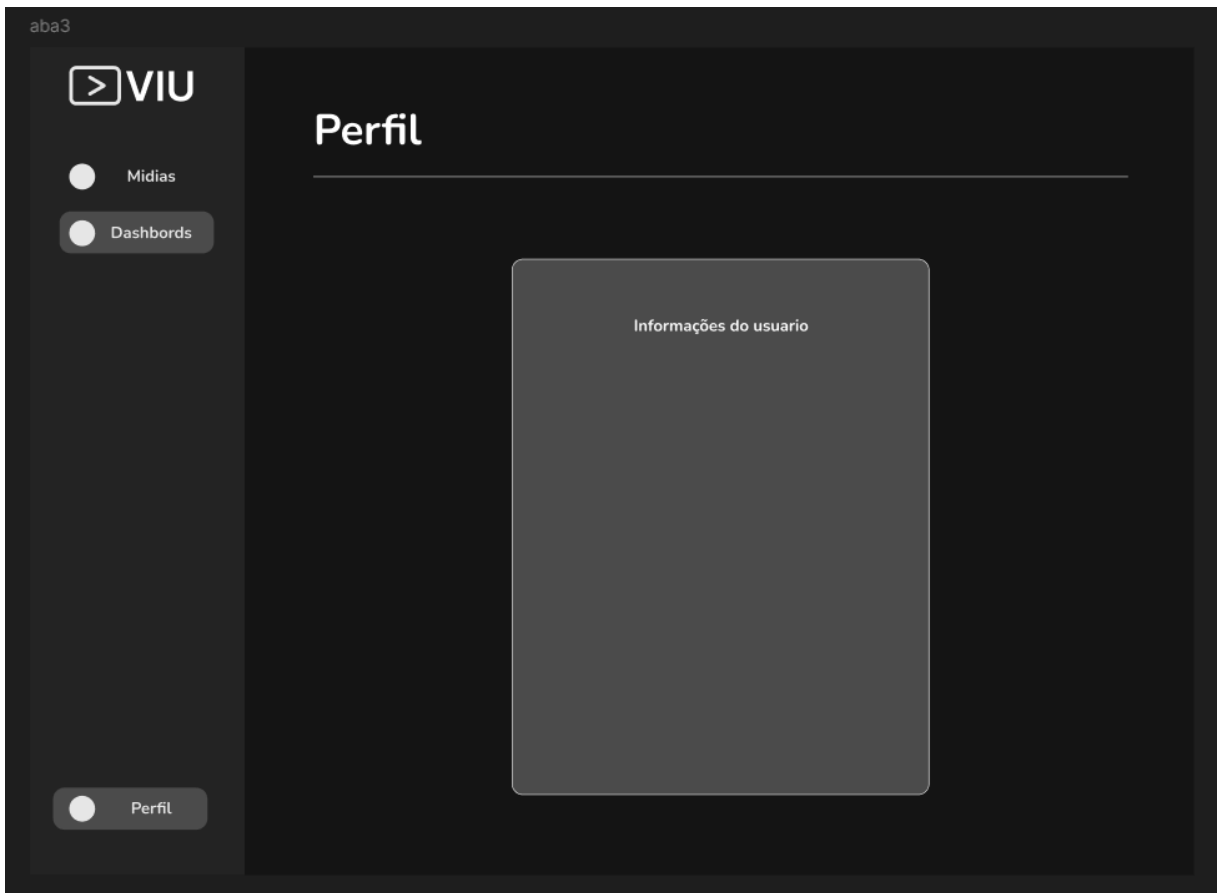


Fonte: Própria (2025)

- Menu de navegação lateral (Sidebar) com o link "Dashboards" (Analytics) ativo.
- A área principal exibe o título "Dashboard" e os dados de *analytics* coletados.
- Dois *cards* superiores exibem indicadores-chave de desempenho (KPIs), como "Sessões Ativas" ou "Tempo Total de Exibição".
- Um *card* principal (maior) na parte inferior é destinado à exibição de um "Gráfico de linha", permitindo a visualização de tendências de engajamento ao longo do tempo.

Tela de Perfil (Configuração):

Figura 10 - Protótipo Tela de Perfil

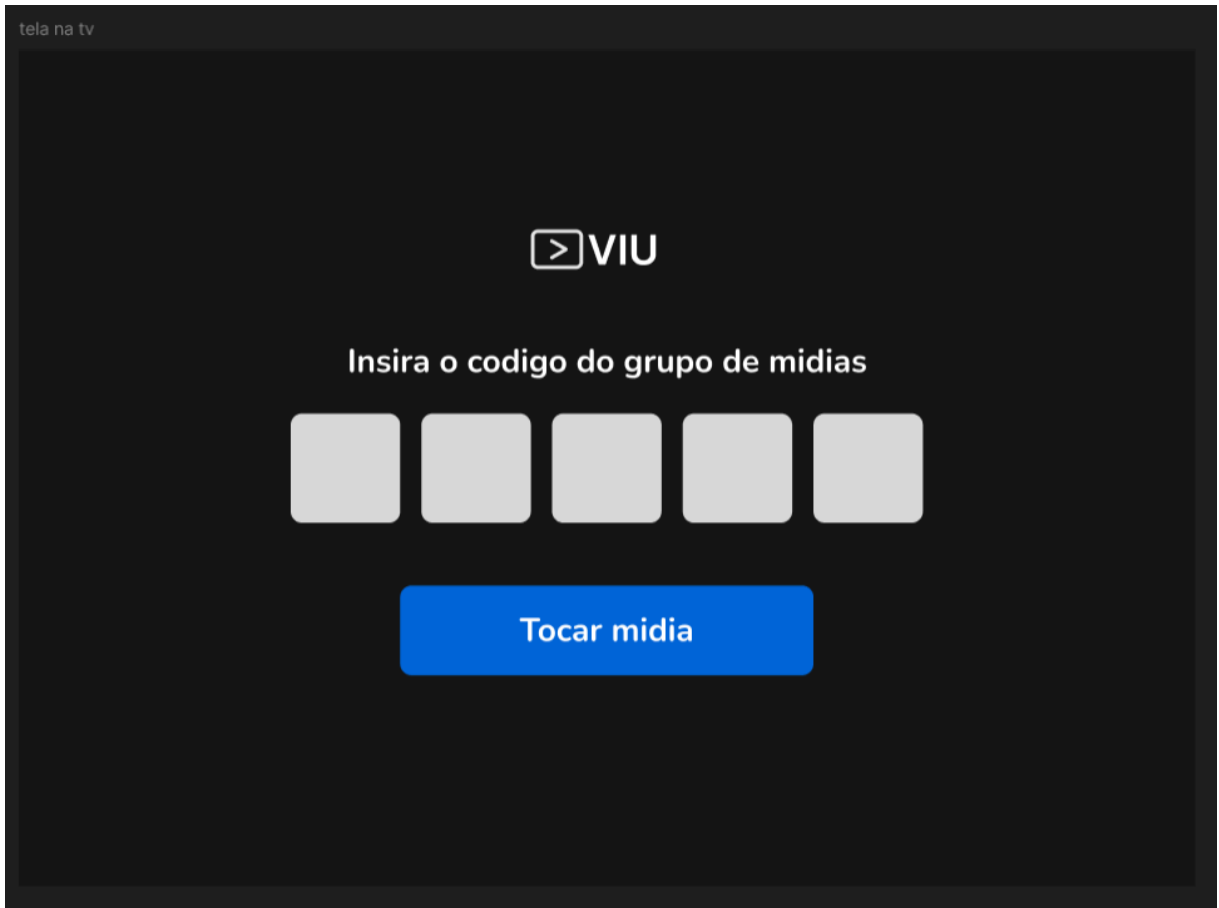


Fonte: Própria (2025)

- Menu de navegação lateral (Sidebar) com o link "Perfil" ativo.
- Área principal com o título "Perfil", indicando a seção de gerenciamento da conta.
- Um *card* central destinado a conter as "Informações do usuário".
- Esta tela é onde o usuário pode editar seus dados cadastrais, como o "Nome Completo" (que chama a API do backend) e "Alterar Senha" (que interage com o Supabase Auth).

Tela Pública /tv (Inserção de Código):

Figura 11 - Protótipo Tela /tv



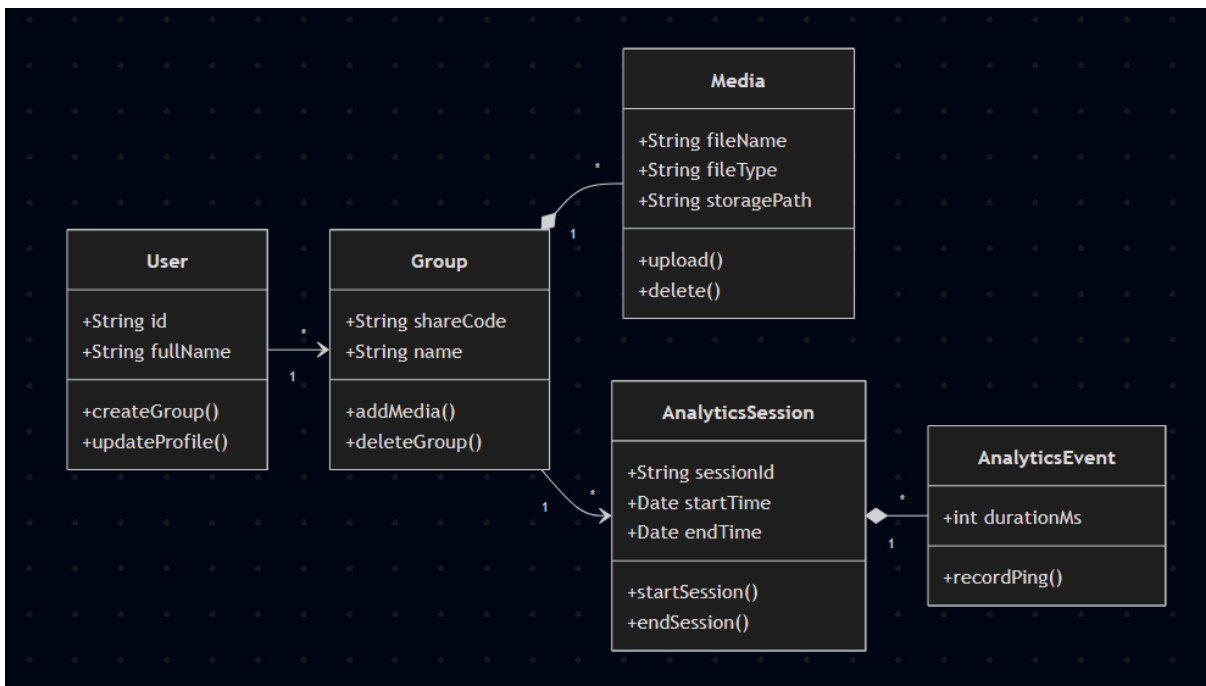
Fonte: Própria (2025)

- Interface minimalista em tela cheia, com fundo escuro, otimizada para visualização em Smart TVs.
- Exibição do logo "VIU" centralizado no topo.
- Um texto de instrução claro: "Insira o código do grupo de mídias".
- Seis caixas de *input* separadas, indicando que o usuário deve inserir o código numérico de 6 dígitos.
- Um botão de ação principal "Tocar mídia" para submeter o código e iniciar o *slideshow*.

6. DIAGRAMAS

6.1. Diagrama de Classe

Figura 12 - Diagrama de Classes



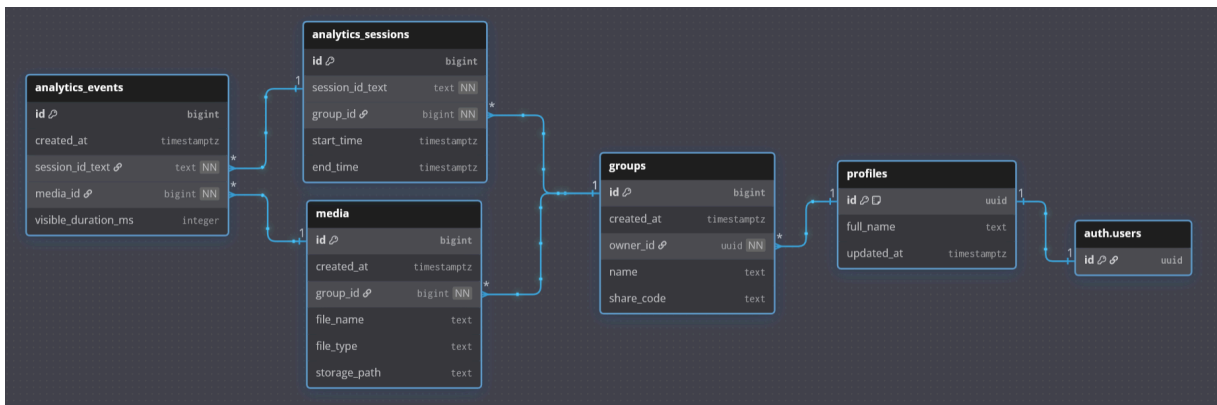
Fonte: Própria (2025)

- O diagrama ilustra a estrutura estática e as principais entidades de dados do sistema "VIU: Gestor de Mídia Digital".
- **Classe User:** Representa o usuário administrador. Possui atributos como `id` e `fullName` (nome completo) e métodos para `createGroup()` (criar grupo) e `updateProfile()` (atualizar perfil).
- **Classe Group:** Representa um grupo de mídias. Contém os atributos `shareCode` (código de compartilhamento) e `name` (nome), além de métodos para gerenciar mídias (`addMedia()`, `deleteGroup()`).
- **Classe Media:** Representa um arquivo de mídia individual (imagem ou vídeo). Possui atributos como `fileName`, `fileType` e `storagePath` (caminho no Supabase Storage) e métodos para `upload()` e `delete()`.

- **Classe AnalyticsSession:** Representa uma sessão de visualização na tela /tv. Possui atributos como `sessionId`, `startTime` e `endTime`, e métodos para `startSession()` e `endSession()`.
- **Classe AnalyticsEvent:** Representa um "ping" de engajamento, registrando o tempo que uma mídia ficou visível (`durationMs`).
- **Relacionamentos:** O diagrama define as seguintes cardinalidades:
 - Um User (usuário) pode possuir um ou mais (*) Group (grupos).
 - Um Group (grupo) pode conter uma ou mais (*) Media (mídias).
 - Um Group (grupo) pode estar associado a uma ou mais (*) AnalyticsSession (sessões).
 - Uma AnalyticsSession (sessão) pode registrar um ou mais (*) AnalyticsEvent (eventos).

6.2. Diagrama de Entidade e Relacionamento (DER)

Figura 13 - Diagrama de Entidade e Relacionamento



Fonte: Própria (2025)

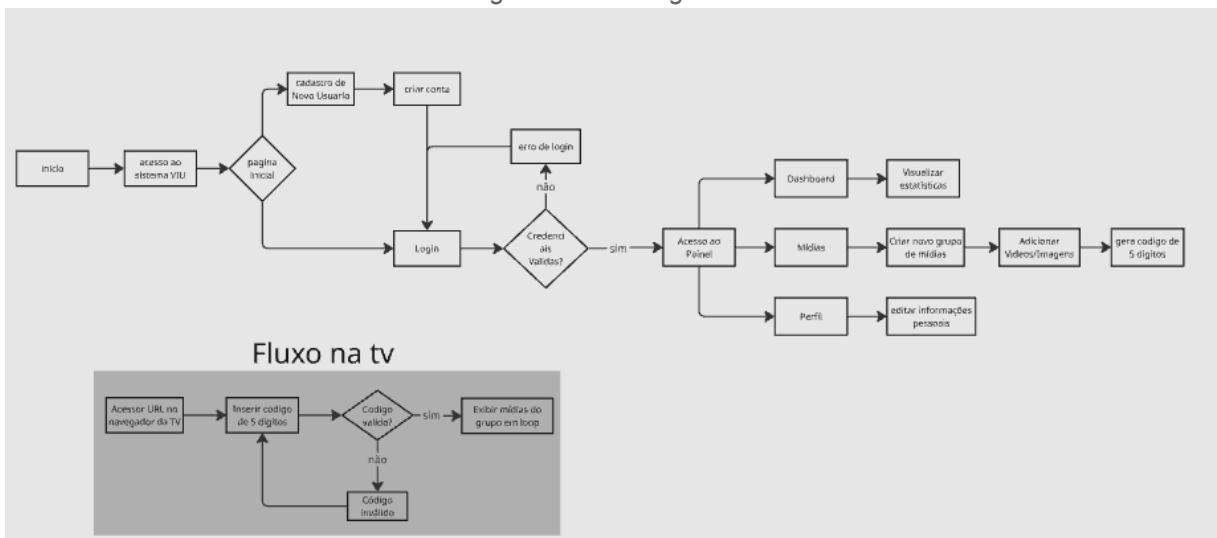
- O diagrama detalha o *schema* do banco de dados relacional (Postgres) hospedado no Supabase, mostrando as tabelas, seus atributos (colunas) e as chaves estrangeiras que definem os relacionamentos.
- **Tabela auth.users:** Tabela interna do Supabase Auth que armazena os dados de autenticação (email, senha criptografada).
- **Tabela profiles:** Armazena os dados públicos do usuário (`full_name`).

Possui um relacionamento de **um-para-um (1:1)** com a `auth.users` (via `id` como chave primária e estrangeira).

- **Tabela `groups`:** Armazena os grupos de mídia. Possui um relacionamento de **um-para-muitos (1:*)** com `profiles` (um perfil pode ter muitos grupos, via `owner_id`).
- **Tabela `media`:** Armazena os metadados dos arquivos. Possui um relacionamento de **um-para-muitos (1:*)** com `groups` (um grupo pode ter muitas mídias, via `group_id`).
- **Tabela `analytics_sessions`:** Armazena as sessões de visualização. Possui um relacionamento de **um-para-muitos (1:*)** com `groups` (um grupo pode ter muitas sessões, via `group_id`).
- **Tabela `analytics_events`:** Armazena os "pings" de visibilidade. Possui dois relacionamentos de **um-para-muitos (1:*)**:
 - Um com `analytics_sessions` (uma sessão pode ter muitos eventos, via `session_id_text`).
 - Um com `media` (uma mídia pode aparecer em muitos eventos, via `media_id`).

6.3. Fluxograma

Figura 14 - Fluxograma

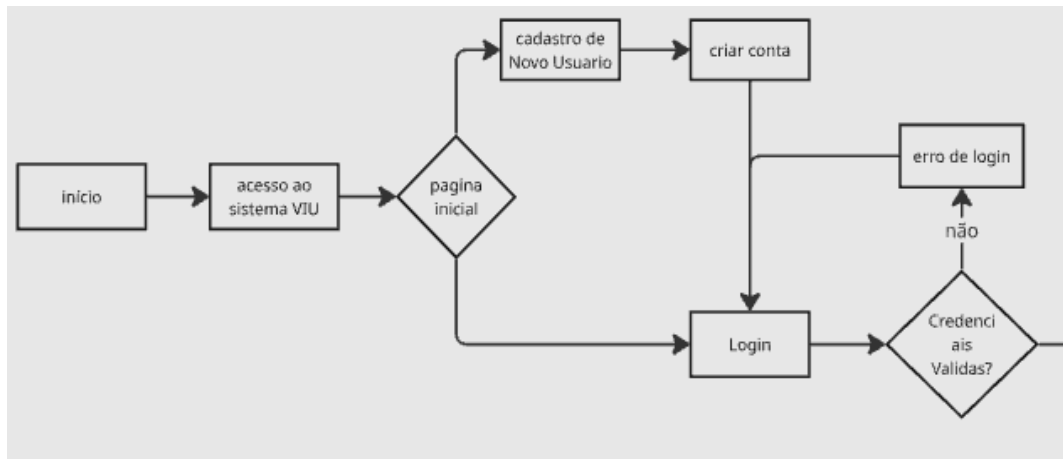


Fonte: Própria (2025)

Explicação do Fluxograma:

Acesso ao Sistema VIU

Figura 15 - Fluxograma Acesso e Cadastro/Login



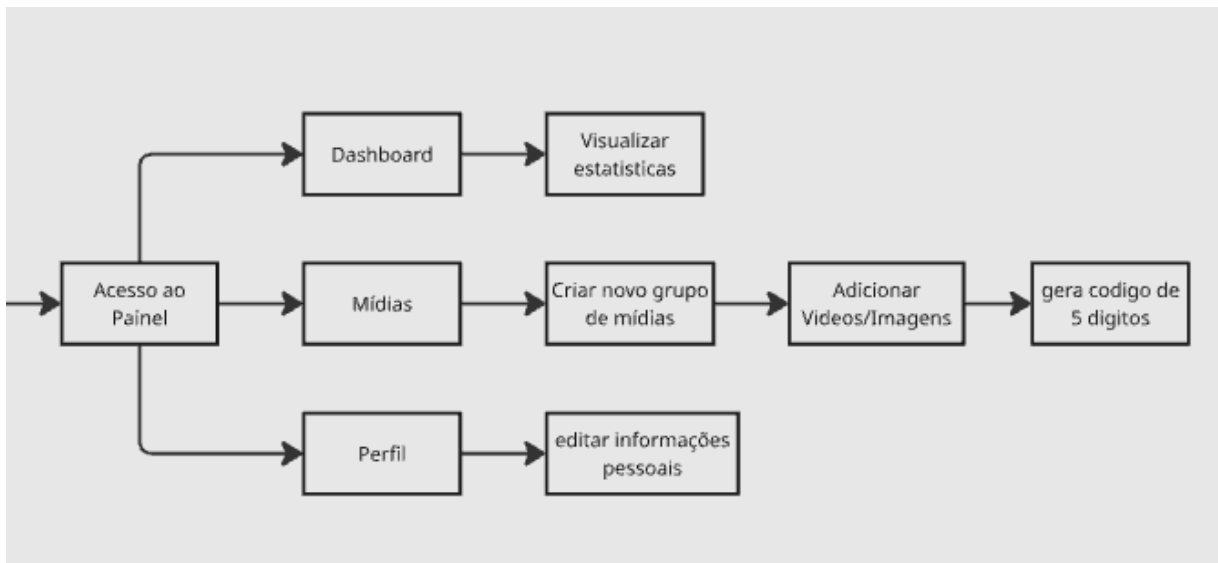
Fonte: Própria (2025)

1. Acesso ao sistema: O usuário inicia o processo acessando o sistema VIU.
2. Página inicial: O usuário pode optar por realizar login ou efetuar o cadastro de um novo usuário
3. Cadastro de novo usuário: Caso o usuário não possua conta, é direcionado ao formulário de criação de conta. Após o cadastro, retorna à tela de login.
4. Login:
 - O usuário insere suas credenciais.
 - Se as credenciais não forem válidas, o sistema exibe uma mensagem de erro e retorna à tela de login.
 - Se forem válidas, o usuário é autenticado e direcionado ao painel principal.

Objetivo: Facilitar o acesso dos usuários ao sistema VIU, permitindo gerenciar mídias, visualizar estatísticas e editar informações de perfil de forma simples e integrada.

Acesso ao painel

Figura 16 - Fluxograma Acesso ao dashboard, Mídias, Perfil



Fonte: Própria (2025)

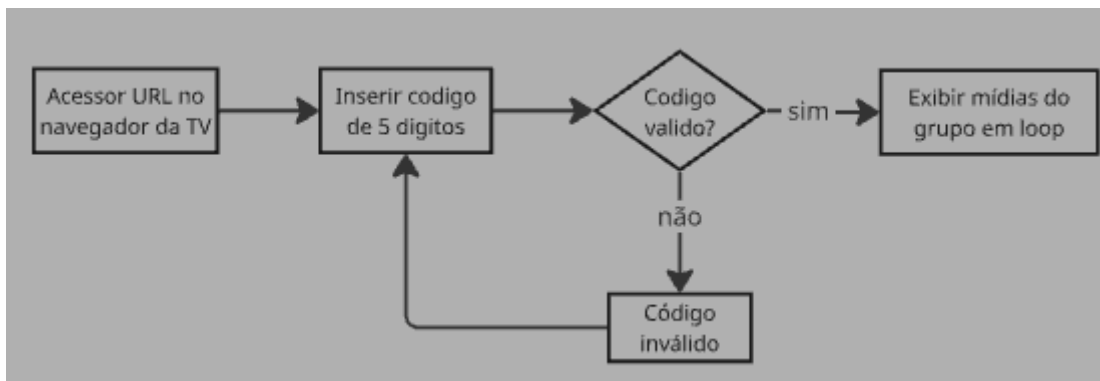
1. Após o login bem-sucedido, o usuário tem acesso às funcionalidades principais do sistema, divididas em três áreas:
 - Dashboard: Exibe estatísticas e informações gerais do uso.
 - Mídias: Permite criar novos grupos de mídias, adicionar vídeos e imagens, além de gerar o código de 5 dígitos para exibição na TV.
 - Perfil: Possibilita a edição de informações pessoais do usuário.

Objetivo:

Facilitar o acesso dos usuários ao sistema VIU, permitindo gerenciar mídias, visualizar estatísticas e editar informações de perfil de forma simples e integrada.

Fluxo na TV

Figura 17 - Fluxograma Exibição de Mídias na TV



Fonte: Própria (2025)

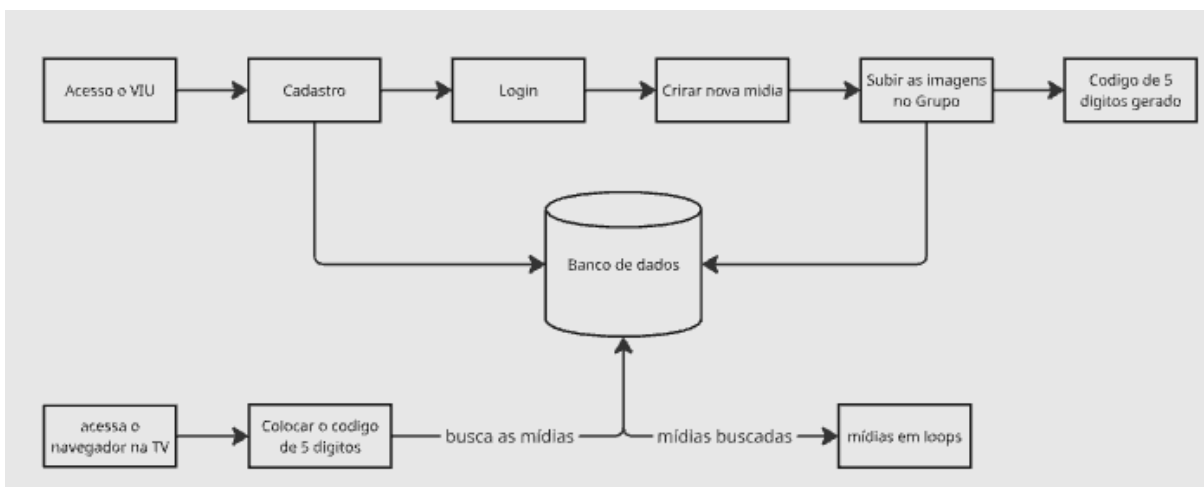
1. Acesso via navegador da TV:
 - O usuário acessa a URL informada no navegador da TV.
2. Inserção de código:
 - É solicitado que o usuário insira o código de 5 dígitos gerado no painel do sistema VIU.
3. Validação do código:
 - Código inválido: O usuário é informado e deve inserir um novo código.
 - Código válido: A exibição das mídias do grupo correspondente é iniciada automaticamente em loop.

Objetivo:

Permitir que os grupos de mídias criados no sistema VIU sejam exibidos de forma contínua na TV, garantindo integração simples e prática entre o painel de controle e o dispositivo de exibição.

6.4. Diagrama de Fluxo de Dados

Figura 18 - Diagrama de Fluxo de Dados



Fonte: Própria (2025)

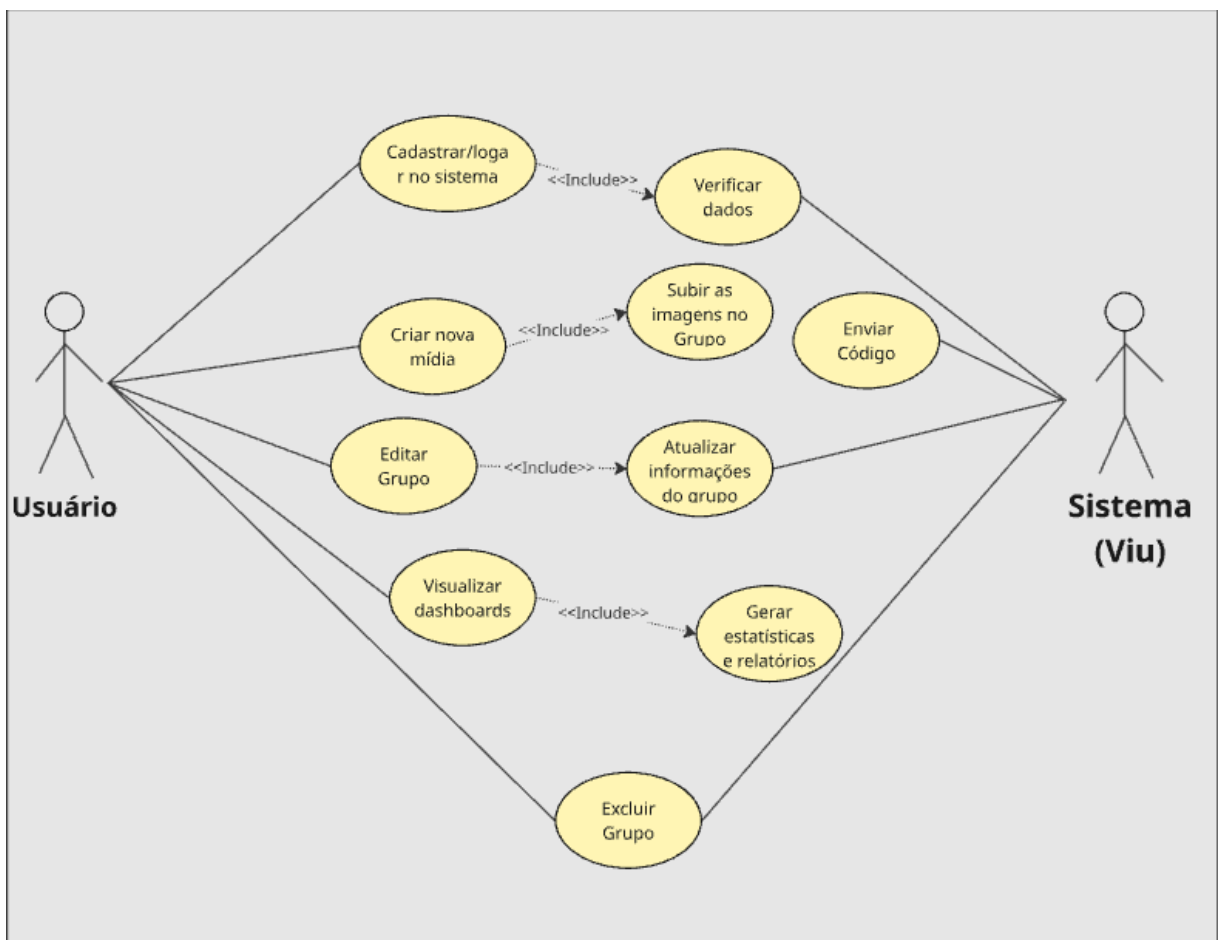
- **Acesso ao VIU**
 - O usuário acessa o sistema.
- **Cadastro**
 - As informações são armazenadas no banco de dados.
- **Login**
 - O usuário entra com suas credenciais para acessar o sistema.
- **Criar Nova Mídia**
 - Após o login, o usuário pode criar um novo grupo de mídias.
- **Subir as imagens no grupo**
 - O usuário faz o upload das imagens ou vídeos no grupo criado.
 - O sistema gera um código de 5 dígitos associado ao grupo.
- **Banco de dados**
 - Armazena informações de usuários, mídias e códigos gerados.
 - Mantém o vínculo entre os grupos de mídia e seus respectivos códigos.
- **Acesso via TV**
 - O usuário acessa o navegador da TV.
 - Insere o código de 5 dígitos fornecido pelo sistema.
- **Busca das Mídias:**

- O sistema realiza a busca das mídias no banco de dados.
- As mídias correspondentes são retornadas e exibidas
- **Exibição**
 - As mídias buscadas são apresentadas em **loop** na tela da TV.
- **FIM**

6.5. Diagrama de Caso de Uso

O **diagrama de casos de uso** apresenta as interações entre o **usuário** e o **sistema VIU**, destacando as principais funcionalidades oferecidas pela aplicação. Esse diagrama tem como objetivo demonstrar as ações que podem ser executadas pelo usuário dentro da plataforma e as respostas automáticas realizadas pelo sistema.

Figura 19 - Diagrama de Caso de Uso



Fonte: Própria (2025)

Atores:

- **Usuário:** responsável por interagir diretamente com o sistema, realizando tarefas como cadastro, login, criação e gerenciamento de mídias e grupos.
- **Sistema (VIU):** executa ações automáticas e complementares, como a verificação de dados, envio de códigos e processamento das informações para geração de relatórios

Principais Casos de Uso:

Autenticação e Acesso:

- **Cadastrar/logar no sistema:** permite que o usuário acesse o sistema ou crie uma nova conta.
 - *Inclui:* **Verificar dados**, responsável por validar as informações inseridas.
 - O **sistema VIU** também realiza o **envio de código**, caso haja necessidade de autenticação adicional.

Gerenciamento de Mídias e Grupos:

- **Criar nova mídia:** possibilita o envio de arquivos e imagens vinculadas a um grupo.
 - *Inclui:* **Subir as imagens no grupo**, processo interno executado pelo sistema.
- **Editar grupo:** permite ao usuário atualizar informações já existentes em um grupo.
 - *Inclui:* **Atualizar informações do grupo**, que garante a consistência dos dados modificados.
- **Excluir grupo:** função destinada à remoção de grupos criados pelo usuário.

Visualização e Relatórios:

- **Visualizar dashboards:** oferece acesso a dados analíticos e painéis informativos.

- *Inclui:* **Gerar estatísticas e relatórios**, ação automatizada do sistema responsável por compilar e apresentar informações de forma visual e organizada.

Relacionamentos:

- Casos de uso como **“Cadastrar/Logar no sistema”** incluem (<<include>>) os subcasos **“Verificar dados”** e **“Enviar código”**.
Da mesma forma, **“Criar nova mídia”** inclui **“Subir as imagens no grupo”**, **“Editar grupo”** inclui **“Atualizar informações do grupo”**, e **“Visualizar dashboards”** inclui **“Gerar estatísticas e relatórios”**.

CONCLUSÃO

O desenvolvimento do projeto "VIU: Gestor de Mídia Digital" atingiu com sucesso seus objetivos principais, culminando na entrega de uma plataforma *full-stack* robusta, escalável e funcional. O objetivo inicial era criar um sistema que permitisse a usuários gerenciar e exibir conteúdo de mídia (imagens e vídeos) em telas remotas, ao mesmo tempo em que coletava dados de engajamento, e isso foi plenamente alcançado.

Utilizando um *stack* tecnológico moderno e desacoplado, o projeto foi arquitetado em três componentes principais: um *frontend* reativo construído com Next.js (utilizando o App Router e CSS Modules), uma API de *backend* segura e modularizada em Node.js (seguindo o padrão MVC), e uma infraestrutura de serviços (BaaS) gerenciada pelo Supabase (PostgreSQL, Auth e Storage).

Durante o desenvolvimento, diversos desafios técnicos de integração foram encontrados e superados. Destacam-se a complexidade da configuração de segurança, como as políticas de CORS (Cross-Origin Resource Sharing) entre o *frontend* (implantado na Vercel) e o *backend* (implantado no Render), e a correta implementação da RLS (Row Level Security) no Supabase. Esta última foi crucial para proteger os dados do usuário, ao mesmo tempo em que permitia o acesso público controlado à tela /tv. A depuração de bugs de *build* no *frontend* também se mostrou um desafio significativo, levando à decisão estratégica de pivotar do Tailwind CSS para CSS Modules, garantindo maior estabilidade e performance.

O resultado final é uma aplicação coesa e funcional. O sistema gerencia com sucesso o cadastro e autenticação de usuários (via E-mail e Google OAuth), a criação de grupos de mídia, o *upload* de arquivos para o *storage*, e a geração de códigos de compartilhamento numéricos de 5 dígitos.

Este projeto não apenas solidifica o conhecimento em arquitetura *full-stack* e na integração de múltiplas plataformas de nuvem, mas também serve como uma fundação sólida para futuras expansões, como a implementação de diferentes níveis de assinatura ou a expansão dos tipos de métricas de *analytics* coletadas.

REFERÊNCIAS

- **AXIOS**. Axios - Promise based HTTP client. Disponível em: <https://axios-http.com/docs/intro>. Acesso em: 01 out. 2025.
- **CSS-MODULES**. CSS Modules. Disponível em: <https://github.com/css-modules/css-modules>. Acesso em: 05 out. 2025.
- **EXPRESS.JS**. Express - Node.js web application framework. Disponível em: <https://expressjs.com/>. Acesso em: 02 set. 2025.
- **FIGMA**. Figma: The collaborative interface design tool. Disponível em: <https://www.figma.com/>. Acesso em: 15 ago. 2025.
- **GITHUB**. GitHub Docs. Disponível em: <https://docs.github.com/>. Acesso em: 16 ago. 2025.
- **LUCIDE**. Lucide - Beautiful & consistent icon toolkit. Disponível em: <https://lucide.dev/>. Acesso em: 12 out. 2025.
- **MDN WEB DOCS**. Cross-Origin Resource Sharing (CORS). Disponível em: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>. Acesso em: 01 nov. 2025.
- **MDN WEB DOCS**. MVC - Model-View-Controller. Disponível em: <https://developer.mozilla.org/en-US/docs/Glossary/MVC>. Acesso em: 03 set. 2025.
- **MERMAID**. Mermaid - Generation of diagrams from text. Disponível em: <https://mermaid.js.org/>. Acesso em: 10 nov. 2025.
- **NANOID**. Nanoid - A tiny, secure, URL-friendly, unique string ID generator. Disponível em: <https://github.com/ai/nanoid>. Acesso em: 15 set. 2025.
- **NEXT.JS**. Next.js by Vercel - Documentation. Disponível em: <https://nextjs.org/docs>. Acesso em: 18 ago. 2025.
- **NODE.JS**. Node.js Documentation. Disponível em: <https://nodejs.org/en/docs>. Acesso em: 18 ago. 2025.
- **POSTGRESQL**. PostgreSQL Documentation. Disponível em: <https://www.postgresql.org/docs/>. Acesso em: 20 ago. 2025.
- **POSTMAN**. Postman API Platform. Disponível em: <https://www.postman.com/docs/>. Acesso em: 05 set. 2025.

- **REACT.** React Documentation. Disponível em: <https://react.dev/learn>. Acesso em: 18 ago. 2025.
- **RECHARTS.** Recharts - A composable charting library built on React components. Disponível em: <https://recharts.org/en-US/>. Acesso em: 18 out. 2025.
- **RENDER.** Render Documentation. Disponível em: <https://render.com/docs>. Acesso em: 21 ago. 2025.
- **SUPABASE.** Supabase Documentation. Disponível em: <https://supabase.com/docs>. Acesso em: 20 ago. 2025.
- **TYPESCRIPT.** TypeScript Documentation. Disponível em: <https://www.typescriptlang.org/docs/>. Acesso em: 18 ago. 2025.
- **VERCEL.** Vercel Documentation. Disponível em: <https://vercel.com/docs>. Acesso em: 21 ago. 2025.