



UFOP

Universidade Federal
de Ouro Preto

**Universidade Federal de Ouro Preto
Instituto de Ciências Exatas e Aplicadas
Departamento de Computação e Sistemas**

Plataforma Web Colaborativa para Organização Doméstica e Gestão Familiar

Matheus Oliveira Vilela

TRABALHO DE CONCLUSÃO DE CURSO

**ORIENTAÇÃO:
Samuel Souza Brito**

**Agosto, 2025
João Monlevade—MG**

Matheus Oliveira Vilela

Plataforma Web Colaborativa para Organização Doméstica e Gestão Familiar

Orientador: Samuel Souza Brito

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

Universidade Federal de Ouro Preto

João Monlevade

Agosto de 2025

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

V699p Vilela, Matheus Oliveira.
Plataforma web colaborativa para organização doméstica e gestão familiar. [manuscrito] / Matheus Oliveira Vilela. - 2025.
47 f.: il.: color., tab..

Orientador: Prof. Dr. Samuel Souza Brito.
Monografia (Bacharelado). Universidade Federal de Ouro Preto.
Instituto de Ciências Exatas e Aplicadas. Graduação em Sistemas de Informação .

1. Aplicações Web. 2. Casa - Divisão do trabalho. 3. Economia doméstica. 4. Orçamento familiar. 5. Software - Desenvolvimento. 6. Software de aplicação - Desenvolvimento. I. Brito, Samuel Souza. II. Universidade Federal de Ouro Preto. III. Título.

CDU 64:004.774

Bibliotecário(a) Responsável: Flavia Reis - CRB6/2431



FOLHA DE APROVAÇÃO

Matheus Oliveira Vilela

Plataforma Web Colaborativa para Organização Doméstica e Gestão Familiar

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação

Aprovada em 04 de setembro de 2025

Membros da banca

Dr. Samuel Souza Brito - Orientador - Universidade Federal de Ouro Preto
Dr. Fernando Bernardes de Oliveira - Universidade Federal de Ouro Preto
Dr. Racyus Delano Garcia Pacifico - Universidade Federal de Ouro Preto

Samuel Souza Brito, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 06/03/2026



Documento assinado eletronicamente por **Samuel Souza Brito, PROFESSOR DE MAGISTERIO SUPERIOR**, em 06/03/2026, às 15:54, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1070759** e o código CRC **0DE6B823**.

Dedico este trabalho, antes de tudo, a Deus, pela saúde, força e perseverança concedidas em cada etapa desta jornada. À minha esposa, que me incentivou a iniciar este curso e, com isso, mudou para sempre o rumo da minha vida. Sua presença diária, seu amor e sua confiança deram novo sentido à minha caminhada e me inspiraram a buscar sempre o melhor de mim. Aos meus pais, que, com amor incondicional, nunca mediram esforços para me oferecer as melhores oportunidades. Sua luta diária e os sacrifícios feitos ao longo da vida foram fundamentais para que eu pudesse chegar até aqui. Cada conquista minha é, também, reflexo da dedicação, do trabalho árduo e dos valores que me transmitiram.

Agradecimentos

Agradeço, em primeiro lugar, a Deus, pela vida e pela oportunidade de trilhar este caminho com saúde e determinação.

À minha esposa, pela paciência, incentivo e apoio constante, fundamentais para que eu me mantivesse firme durante esta jornada acadêmica.

Aos meus pais, pelo amor, pela dedicação incansável e pelos valores transmitidos, que foram a base de cada passo dado até aqui.

Aos meus colegas de curso, que compartilharam desafios, conquistas e aprendizados ao longo desses anos, tornando a caminhada mais leve e enriquecedora.

Aos professores e orientadores, pela dedicação, paciência e pelo conhecimento transmitido, que não apenas enriqueceram este trabalho, mas também contribuíram de maneira decisiva para meu crescimento pessoal e profissional.

“Se eu vi mais longe, foi por estar de pé sobre ombros de gigantes.”

— Isaac Newton

Resumo

O presente trabalho apresenta o desenvolvimento do *Convivu*, uma plataforma web voltada à organização da vida doméstica de casais e pequenos grupos familiares. A aplicação integra funcionalidades de planejamento de tarefas, gestão financeira, controle de listas de compras, organização de refeições, registro de anotações e calendário compartilhado, visando centralizar e melhorar a gestão da rotina. O projeto foi concebido com foco na usabilidade, acessibilidade e colaboração entre os usuários, buscando promover maior eficiência na comunicação e no gerenciamento das responsabilidades do lar. Foram aplicados princípios de design centrado no usuário. Como resultado, obteve-se uma solução funcional e escalável, capaz de atender às demandas de organização doméstica de maneira prática e colaborativa. O trabalho contribui para o campo das soluções digitais voltadas à gestão familiar, sugerindo como o uso de tecnologias modernas pode transformar a maneira como famílias e casais administram suas rotinas.

Palavras-chave: Organização doméstica. Aplicações web. Gestão colaborativa.

Abstract

This paper presents the development of *Convivu*, a web platform designed to organize the domestic life of couples and small family groups. The application integrates features such as task planning, financial management, shopping list control, meal organization, note-taking, and a shared calendar, aiming to centralize and optimize routine management. The project was developed with a focus on usability, accessibility, and collaboration among users, seeking to promote greater efficiency in communication and the management of household responsibilities. User-centered design principles were applied throughout the process. The result is a functional and scalable solution capable of meeting the demands of home organization in a practical and collaborative manner. This work contributes to the field of digital solutions for family management, demonstrating how modern technologies can transform the way families and couples manage their routines.

Keywords: Domestic organization. Web applications. Collaborative management.

Lista de ilustrações

Figura 1 – <i>Business Model Canvas</i> do Convivu	24
Figura 2 – Diagrama Entidades do Convivu	30
Figura 3 – Estruturação de pastas do <i>frontend</i>	33
Figura 4 – Tela Inicial do Convivu	35
Figura 5 – Gestão Financeira	36
Figura 6 – Metas	37
Figura 7 – Desejos	37
Figura 8 – Planejamento de Refeições	38
Figura 9 – Adicionar refeição	38
Figura 10 – Refeições Salvas	39
Figura 11 – Lista de Compras	39
Figura 12 – Configurações do Grupo	40
Figura 13 – Tela de Login e cadastro	40
Figura 14 – Grupo Familiar	41
Figura 15 – Cabeçalho da Aplicação	42
Figura 16 – Responsividade para Dispositivos Móveis	42

Lista de tabelas

Tabela 1 – Comparativo de funcionalidades entre aplicativos relacionados	22
Tabela 2 – Requisitos funcionais do Convivu	27
Tabela 3 – Requisitos não-funcionais do Convivu	27

Lista de abreviaturas e siglas

API	Application Programming Interface
BaaS	Backend as a Service
CDN	Content Delivery Network
COSI	Colegiado do Curso de Sistemas de Informação
CSS	Cascading Style Sheets
DECSI	Departamento de Computação e Sistemas
DOM	Document Object Model
HMR	Hot Module Replacement
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
I/O	Input/Output
ICEA	Instituto de Ciências Exatas e Aplicadas
JM	João Monlevade
JWT	JSON Web Token
RLS	Row Level Security
SI	Sistemas de Informação
SPA	Single Page Application
SQL	Structured Query Language
SSL	Secure Sockets Layer
UI	User Interface
UFOP	Universidade Federal de Ouro Preto
UUID	Universally Unique Identifier
UX	User Experience

Sumário

1	INTRODUÇÃO	14
1.1	Objetivos	15
1.2	Organização do Trabalho	15
2	REVISÃO BIBLIOGRÁFICA	17
2.1	Referencial Teórico	17
2.1.1	Organização Doméstica Digital	17
2.1.2	Design Centrado no Usuário	17
2.1.3	Personas	18
2.1.4	Prototipação	18
2.2	Tecnologias	18
2.2.1	<i>Supabase</i>	18
2.2.2	<i>TypeScript</i>	19
2.2.3	<i>React</i>	20
2.2.4	<i>Figma</i>	21
2.3	Trabalhos relacionados	21
3	METODOLOGIA	23
3.1	Modelagem do Negócio	23
3.2	Personas	24
4	DESENVOLVIMENTO	26
4.1	Levantamento de Requisitos	26
4.1.1	Requisitos Funcionais	26
4.1.2	Requisitos Não-Funcionais	26
4.2	Camada de Dados e Serviços (<i>Supabase</i>)	28
4.2.1	Esquema de Dados	28
4.2.1.1	Políticas de Segurança (RLS)	29
4.3	<i>Frontend</i>	31
4.3.1	Ambiente de Desenvolvimento	31
4.3.2	Bibliotecas	31
4.3.3	Estrutura de Diretórios	32
4.3.4	Principais Funcionalidades do <i>Frontend</i>	32
4.4	<i>Deploy</i> e Hospedagem	33
4.4.1	<i>Build</i> e Variáveis de Ambiente	33
4.4.2	Hospedagem do <i>Frontend</i>	34

4.4.3	Infraestrutura de Dados	34
5	RESULTADOS	35
6	CONCLUSÃO	43
6.1	Trabalhos futuros	43
	REFERÊNCIAS	45

1 Introdução

A organização da rotina doméstica envolve a coordenação de tarefas, responsabilidades, compromissos e recursos financeiros compartilhados entre os membros de um lar. Quando estruturadas de maneira adequada, essas atividades contribuem para reduzir conflitos, melhorar o aproveitamento do tempo e promover maior equilíbrio na divisão de responsabilidades. Estudos indicam que falhas de planejamento e comunicação estão entre os principais fatores associados à sobrecarga e ao estresse em contextos familiares (RODRIGUES, 2020).

No cotidiano, tarefas como compras, controle de despesas, organização de refeições e compromissos costumam ser administradas por meio de diferentes aplicativos ou registros manuais. Essa fragmentação dificulta a visualização integrada da rotina e aumenta a carga cognitiva dos usuários, que precisam alternar entre múltiplas ferramentas (SILVA; OLIVEIRA; SANTOS, 2021).

Diante desse cenário, este trabalho apresenta o *Convivu*, uma plataforma web voltada à gestão colaborativa da vida doméstica. A proposta consiste em reunir, em um único ambiente digital, funcionalidades de planejamento financeiro, listas de compras, organização de refeições, notas compartilhadas e calendário de compromissos. A centralização dessas informações busca melhorar a comunicação entre os membros do grupo e favorecer a corresponsabilidade nas decisões cotidianas.

O crescimento do uso de dispositivos móveis reforça a relevância de soluções digitais voltadas à organização pessoal e familiar. Segundo dados do relatório Global Digital Overview 2023, o smartphone é o principal dispositivo de acesso à internet no Brasil, sendo amplamente utilizado para organização de tarefas e comunicação. Nesse contexto, observa-se uma demanda crescente por aplicações colaborativas que integrem diferentes dimensões da vida doméstica.

O desenvolvimento da plataforma considerou requisitos de usabilidade, segurança e possibilidade de evolução futura. Para isso, foram adotadas tecnologias como React e TypeScript na camada de interface e o Supabase como infraestrutura de dados e autenticação. A escolha dessas ferramentas está relacionada à necessidade de manter um sistema modular, seguro e capaz de atender a múltiplos usuários simultaneamente.

Neste trabalho, serão aplicadas técnicas e ferramentas de desenvolvimento de software, aliadas a princípios de design centrado no usuário, com o intuito de criar uma aplicação eficiente, intuitiva e adaptada às demandas reais de famílias e casais. Espera-se que o *Convivu* contribua não apenas para a melhoria da organização doméstica e da gestão financeira, mas também para o fortalecimento da convivência familiar, tornando as

interações diárias mais fluidas, equilibradas e colaborativas.

1.1 Objetivos

O presente projeto tem como objetivo principal desenvolver um aplicativo que integre funcionalidades voltadas à gestão financeira e organizacional de casais ou pequenos grupos familiares que residem na mesma unidade habitacional. O foco é oferecer uma plataforma que:

- Proporcione o planejamento e acompanhamento de metas financeiras conjuntas;
- Disponibilize ferramentas para o registro e categorização de despesas;
- Possibilite a criação de listas de compras integradas ao controle de ingredientes disponíveis;
- Facilite o planejamento de refeições como estratégia de redução de desperdícios e gastos;
- Centralize notas e anotações de interesse comum, promovendo comunicação clara e acessível.

O Convivu pretende não apenas funcionar como uma ferramenta de registro e controle, mas também como um facilitador da comunicação e da cooperação entre os membros do lar, contribuindo para uma gestão mais eficiente e para o fortalecimento da convivência harmoniosa.

1.2 Organização do Trabalho

Este trabalho está estruturado em seis capítulos, além dos elementos pré-textuais e pós-textuais, organizados da seguinte forma:

- **Capítulo 1 – Introdução:** apresenta o contexto da pesquisa, a justificativa para o desenvolvimento do Convivu, os objetivos do projeto e a relevância da proposta no cenário atual.
- **Capítulo 2 – Revisão Bibliográfica:** reúne conceitos teóricos e trabalhos correlatos que fundamentam a solução proposta, abordando temas como organização doméstica digital, design centrado no usuário, criação de personas, prototipação e as principais tecnologias utilizadas.

- **Capítulo 3 – Metodologia:** descreve a abordagem metodológica adotada, incluindo a modelagem de negócio com o Business Model Canvas, a definição de personas e o processo de prototipação, além das estratégias de desenvolvimento utilizadas.
- **Capítulo 4 – Desenvolvimento:** detalha a implementação da plataforma Convivu, abrangendo o levantamento de requisitos, a camada de dados e serviços baseada no Supabase, o frontend desenvolvido em React e TypeScript, e o processo de deploy e hospedagem.
- **Capítulo 5 – Resultados:** apresenta os módulos e funcionalidades implementados, ilustrados por capturas de tela e descrições das principais interfaces, demonstrando a viabilidade técnica e a aplicabilidade prática da solução.
- **Capítulo 6 – Conclusão:** sintetiza as contribuições do trabalho, destacando os objetivos alcançados, as lições aprendidas e as perspectivas para evolução do Convivu, com propostas de trabalhos futuros.

2 Revisão Bibliográfica

A fim de contextualizar o leitor acerca do desenvolvimento deste projeto, este capítulo apresenta conceitos, metodologias, tecnologias e trabalhos correlatos que fundamentam a concepção e implementação do Convivu. A abordagem aqui descrita visa oferecer suporte teórico e prático para compreender as decisões técnicas, estratégicas e funcionais tomadas durante o desenvolvimento da plataforma.

2.1 Referencial Teórico

No referencial teórico deste trabalho são explorados conceitos essenciais para o entendimento do projeto. A *organização doméstica digital*, que compreende o uso de tecnologias para gerenciar tarefas, finanças, compras e compromissos compartilhados, será discutida como elemento central para compreender o contexto e a necessidade da solução. Além disso, o *Design Centrado no Usuário* (DCU) será aplicado como abordagem metodológica para garantir que a experiência seja adaptada às demandas reais dos usuários. A criação de personas, representações fictícias de perfis de uso, será empregada para orientar decisões de design e funcionalidades. Por fim, a prototipação será utilizada como prática para validar e iterar soluções antes de sua implementação definitiva, garantindo eficiência e alinhamento com os objetivos propostos.

2.1.1 Organização Doméstica Digital

Segundo [Silva, Oliveira e Santos \(2021\)](#), a organização doméstica digital refere-se à aplicação de sistemas e ferramentas tecnológicas para coordenar atividades, gerir recursos e facilitar a comunicação entre os membros de um lar. Essa prática tem crescido com a popularização de smartphones e aplicativos colaborativos, possibilitando que famílias e casais centralizem informações e responsabilidades em um único ambiente virtual.

Os benefícios dessa abordagem incluem a redução de desperdícios, o aumento da previsibilidade de tarefas e a promoção da corresponsabilidade. Aplicativos que oferecem listas de compras inteligentes, calendários compartilhados e controle financeiro integrado têm mostrado impacto positivo na produtividade e no relacionamento interpessoal dos usuários ([Rodrigues \(2020\)](#)).

2.1.2 Design Centrado no Usuário

O *Design Centrado no Usuário* (DCU) é uma abordagem de desenvolvimento que coloca as necessidades e expectativas do usuário no centro do processo ([Norman \(2018\)](#)).

Essa metodologia orienta as etapas de pesquisa, concepção, prototipação e avaliação, garantindo que o produto final seja funcional, acessível e satisfatório.

No Convivu, o DCU é aplicado por meio de testes de usabilidade e coleta de *feedback* contínuo, permitindo ajustes rápidos nas funcionalidades e na interface. Essa prática contribui para maior taxa de adoção e para a retenção de usuários, visto que as soluções são construídas de forma alinhada às suas rotinas reais.

2.1.3 Personas

De acordo com Cooper et al. (2014), personas são representações fictícias de perfis de usuários reais, construídas com base em dados demográficos, comportamentais e motivacionais. Elas funcionam como uma ferramenta de design para guiar decisões, garantindo que as funcionalidades atendam a diferentes segmentos do público-alvo.

No contexto do Convivu, as personas ajudam a definir como usuários com diferentes níveis de familiaridade tecnológica e distintos arranjos familiares interagem com o aplicativo. Isso possibilita criar fluxos personalizados e priorizar funcionalidades que tragam maior valor para cada perfil identificado.

2.1.4 Prototipação

A prototipação é uma técnica essencial no processo de design e desenvolvimento, permitindo criar representações tangíveis das ideias para validação prévia com os usuários (Rogers, Sharp e Preece (2019)). Pode assumir a forma de protótipos de baixa fidelidade, úteis para explorar conceitos iniciais, ou de alta fidelidade, quando se deseja avaliar aspectos visuais e funcionais próximos ao produto final.

No desenvolvimento do Convivu, a prototipação iterativa é utilizada para validar a disposição dos elementos na tela, testar a clareza das interações e medir a satisfação do usuário antes da implementação definitiva, reduzindo riscos e custos de retrabalho.

2.2 Tecnologias

Nesta seção, são descritas as principais tecnologias empregadas no desenvolvimento do Convivu.

2.2.1 Supabase

O *Supabase*¹ é uma plataforma de código aberto para o desenvolvimento de aplicativos, baseada em tecnologias padrão de mercado como *PostgreSQL* (Supabase (2023)). Ela

¹ <<https://supabase.com/>>

oferece recursos como autenticação de usuários, banco de dados relacional, armazenamento de arquivos, funções *serverless*, modelo de execução em que a lógica da aplicação é processada sob demanda, sem que o desenvolvedor precise gerenciar diretamente a infraestrutura de servidores e APIs geradas automaticamente a partir do esquema do banco de dados.

Uma das principais vantagens do Supabase é sua arquitetura aberta e compatível com padrões SQL, o que facilita a migração e a integração com outros serviços. Além disso, por ser construído sobre o PostgreSQL, oferece suporte a consultas complexas, gatilhos, funções armazenadas e extensões avançadas, proporcionando maior flexibilidade no desenvolvimento.

No contexto do Convivu, o Supabase é utilizado para gerenciar dados de usuários, armazenar informações relacionadas a tarefas, finanças e listas de compras, além de garantir sincronização em tempo real entre dispositivos. Sua infraestrutura escalável e a possibilidade de hospedagem própria também oferecem maior controle sobre segurança e privacidade dos dados.

2.2.2 *TypeScript*

O *TypeScript*² é um superset do *JavaScript* que adiciona tipagem estática opcional (Microsoft (2022)). Diferentemente do *JavaScript*, que é fracamente tipado e interpretado dinamicamente em tempo de execução, o *TypeScript* permite a definição explícita de tipos de variáveis, funções, objetos e interfaces já em tempo de desenvolvimento. Essa característica possibilita a detecção precoce de erros, reduzindo falhas em produção e aumentando a confiabilidade do código.

Outra vantagem relevante é que o *TypeScript* mantém total compatibilidade com o ecossistema do *JavaScript*, já que seu código é transpilado para *JavaScript* padrão, podendo ser executado em qualquer navegador ou ambiente que suporte a linguagem. Além disso, oferece suporte a recursos modernos da linguagem, mesmo em ambientes que ainda não possuem suporte nativo, garantindo maior portabilidade.

Entre seus principais recursos destacam-se:

- **Tipagem Estática:** definição de tipos em tempo de desenvolvimento, auxiliando na prevenção de erros comuns;
- **Interfaces e Classes:** suporte a princípios de orientação a objetos, promovendo maior organização e clareza na estrutura do código;
- **Ferramentas de Autocompletar e Refatoração:** integração com editores de código como o Visual Studio Code, proporcionando ganho de produtividade;

² <<https://www.typescriptlang.org/>>

- **Compatibilidade Progressiva:** possibilidade de adoção gradual em projetos já existentes em *JavaScript*.

A tipagem explícita favorece a comunicação entre as equipes de desenvolvimento, reduz a ocorrência de inconsistências nos dados manipulados e facilita a manutenção de longo prazo, contribuindo para a escalabilidade e a evolução contínua do projeto.

2.2.3 *React*

O *React*³ é uma biblioteca de código aberto para construção de interfaces de usuário, criada e mantida pelo Facebook (atualmente Meta) e por uma ampla comunidade de desenvolvedores. Seu principal objetivo é facilitar o desenvolvimento de aplicações web dinâmicas e escaláveis, com foco na criação de interfaces reativas e de fácil manutenção (Meta (2023)).

A abordagem do *React* baseia-se no uso de *componentes*, que são blocos reutilizáveis e independentes responsáveis por representar partes da interface. Essa arquitetura modular promove maior organização do código, reutilização de elementos e separação clara de responsabilidades. Outro recurso central é o *Virtual DOM*, uma representação virtual da árvore de elementos que otimiza o processo de atualização da interface, tornando as mudanças na página mais rápidas e eficientes.

Entre suas principais características, destacam-se:

- **Componentização:** desenvolvimento de interfaces por meio de componentes reutilizáveis e independentes;
- **Virtual DOM:** mecanismo que reduz o custo de renderização e melhora a performance das aplicações;
- **Unidirecionalidade de Dados:** modelo de fluxo de dados previsível e de fácil rastreamento, aumentando a confiabilidade do sistema;
- **Ecossistema Amplo:** vasta quantidade de bibliotecas, ferramentas e suporte da comunidade, que expandem suas funcionalidades.

No contexto do desenvolvimento do *Convivu*, o *React* foi escolhido por sua capacidade de lidar com interfaces dinâmicas e de alto desempenho, essenciais para funcionalidades como atualização em tempo real. A componentização facilita a manutenção e evolução do sistema, permitindo que diferentes módulos sejam desenvolvidos de forma independente e posteriormente integrados em uma experiência unificada e consistente para o usuário.

³ <<https://react.dev/>>

2.2.4 Figma

O *Figma*⁴ é uma ferramenta de design de interfaces e prototipagem baseada em nuvem, amplamente utilizada para o desenvolvimento colaborativo de projetos digitais. Diferentemente de softwares tradicionais instalados localmente, o *Figma* funciona diretamente no navegador ou em aplicativos desktop e móveis, permitindo que equipes trabalhem de forma simultânea em um mesmo arquivo, em tempo real (Figma (2022)).

Entre seus principais recursos destacam-se:

- **Colaboração em Tempo Real:** múltiplos usuários podem editar, comentar e visualizar protótipos de forma simultânea;
- **Protótipos Interativos:** criação de fluxos navegáveis que simulam a experiência final do usuário sem a necessidade de código;
- **Bibliotecas de Componentes:** possibilidade de criar sistemas de design reutilizáveis, promovendo consistência visual em todo o projeto;
- **Acessibilidade:** por ser uma ferramenta em nuvem, dispensa instalações complexas e é compatível com diferentes sistemas operacionais.

No contexto do desenvolvimento do *Convivu*, o *Figma* foi utilizado para a criação de protótipos de baixa e alta fidelidade, permitindo a validação antecipada das interfaces com base nos princípios de design centrado no usuário.

2.3 Trabalhos relacionados

Para enriquecer o presente estudo, foram analisados aplicativos e plataformas que oferecem recursos voltados à organização da rotina doméstica, à gestão financeira compartilhada e à colaboração entre membros de uma família ou casal. As principais funcionalidades identificadas nessas soluções foram catalogadas e comparadas, conforme apresentado na Tabela 1.

Entre os principais exemplos estão o **Cozi**, desenvolvido pela Cozi Inc., que disponibiliza calendários compartilhados, listas de tarefas e listas de compras; o **OurGroceries**, mantido pela HeadCode, que tem como foco principal o gerenciamento de listas de compras; o **Splitwise**, desenvolvido pela Splitwise Inc., que se destaca pela divisão de despesas entre grupos; e o **Google Keep**, pertencente ao Google LLC, que oferece notas rápidas e listas colaborativas.

Outras soluções relevantes incluem o **Honeydue** (Honeydue Inc.), voltado para casais que desejam compartilhar e acompanhar suas finanças; o **Between** (VCNC Inc.), que

⁴ <<https://www.figma.com/>>

oferece um espaço privado para casais trocarem mensagens, fotos, calendários e memórias; o **Merge** (Merge Technologies), aplicativo que combina lista de tarefas e comunicação; o **Raft** (What If Holdings), que organiza eventos sociais e calendários compartilhados; o **Love Nudge** (Moody Publishers), baseado nos conceitos dos "5 linguagens do amor", ajudando casais a fortalecer a comunicação; o **Tandem** (Tandem App), voltado à colaboração e comunicação de casais; e o **YNAB - You Need A Budget** (YNAB LLC), uma das ferramentas mais populares de controle de finanças pessoais, utilizada também em contextos familiares.

A análise comparativa apresentada na Tabela 1 evidencia que, embora existam diversas soluções voltadas a aspectos específicos da rotina doméstica, a maioria delas concentra-se em apenas um ou dois tipos de funcionalidade, como gestão financeira, organização de listas ou compartilhamento de calendários.

Ainda que ofereçam funcionalidades específicas, tais soluções tendem a ser fragmentadas, o que obriga os usuários a alternar entre diferentes aplicativos para organizar finanças, tarefas e interações pessoais. Segundo [Ferreira \(2021\)](#), essa fragmentação compromete a eficiência na gestão da rotina doméstica. Nesse cenário, o *Convivu* propõe uma abordagem integrada, centralizando funcionalidades em um único ambiente, aumentando a praticidade, reduzindo a dispersão de informações e promovendo maior colaboração entre os usuários.

Tabela 1 – Comparativo de funcionalidades entre aplicativos relacionados

Aplicativo	Financeiro	Calendário	Listas	Refeições	Notas
Cozi (Cozi, 2023)	Não	Sim	Sim	Não	Sim
OurGroceries (OurGroceries, 2023)	Não	Não	Sim	Não	Não
Splitwise (Splitwise, 2023)	Sim	Não	Não	Não	Não
Google Keep (Google, 2023)	Não	Não	Sim	Não	Sim
Trello (Atlassian, 2023)	Não	Parcial	Parcial	Não	Sim
Notion (Notion, 2023)	Parcial	Parcial	Parcial	Não	Sim
Honeydue (Honeydue, 2023)	Sim	Não	Não	Não	Sim
Between (Between, 2023)	Não	Sim	Não	Não	Sim
Merge (Merge, 2023)	Não	Sim	Sim	Não	Sim
Raft (Raft, 2023)	Não	Sim	Não	Não	Sim
Love Nudge (Love Nudge, 2023)	Não	Sim	Não	Não	Não
Tandem (Tandem, 2023)	Não	Sim	Não	Não	Sim
YNAB (YNAB, 2023)	Sim	Não	Não	Não	Não
Convivu (Este trabalho)	Sim	Sim	Sim	Sim	Sim

3 Metodologia

Esta seção descreve a abordagem estratégica e prática adotada para o desenvolvimento da plataforma web Convivu. A metodologia proposta busca alinhar princípios de design centrado no usuário, modelagem de negócios, pesquisa de público-alvo e prototipagem iterativa, garantindo que o produto final atenda às necessidades reais dos usuários e seja tecnicamente viável.

O processo metodológico iniciou-se com a utilização da ferramenta *Business Model Canvas* (BMC) para estruturar a proposta de valor e entender de forma holística os elementos-chave do empreendimento. Em seguida, foi realizada a construção de *personas*, que representam perfis típicos dos potenciais usuários, permitindo compreender comportamentos, necessidades e desafios do público-alvo. Posteriormente, a prototipagem no *Figma* possibilitou a validação prévia da interface e da experiência de uso, permitindo ajustes antes da implementação definitiva. A etapa final contemplou a implementação utilizando tecnologias modernas para o desenvolvimento de sistemas web, garantindo desempenho, segurança e escalabilidade.

3.1 Modelagem do Negócio

A modelagem do negócio foi realizada com base no *Business Model Canvas* (Figura 1), ferramenta que permite visualizar e organizar de maneira sistêmica os elementos fundamentais do projeto. No caso do Convivu, o BMC foi estruturado considerando:

- **Proposta de valor:** centralizar a gestão da vida doméstica em uma única plataforma web, oferecendo recursos de planejamento de tarefas, finanças, lista de compras, organização de refeições e registro de notas.
- **Segmentos de clientes:** casais, famílias pequenas, estudantes que dividem moradia e grupos que compartilham despesas e tarefas.
- **Canais:** plataforma web responsiva acessível via navegadores, comunicação por notificações, redes sociais e campanhas de marketing digital.
- **Relacionamento com clientes:** suporte integrado na plataforma, notificações inteligentes e personalizadas, e feedback contínuo por meio de formulários e enquetes.
- **Fontes de receita:** plano gratuito com funcionalidades básicas e plano premium com recursos avançados, como histórico detalhado e exportação de relatórios.

- **Recursos-chave:** equipe de desenvolvimento, infraestrutura de nuvem no *Supabase*, ferramentas de design (*Figma*) e bibliotecas de interface.
- **Atividades-chave:** desenvolvimento de software, manutenção e suporte técnico, pesquisa com usuários e aprimoramento contínuo das funcionalidades.
- **Parcerias-chave:** provedores de autenticação, gateways de pagamento e plataformas de análise de dados.
- **Estrutura de custos:** infraestrutura em nuvem, manutenção e atualizações, marketing e suporte ao cliente.

Essa modelagem permitiu identificar claramente os objetivos estratégicos, alinhar a proposta de valor ao perfil dos usuários e antecipar desafios na implementação.

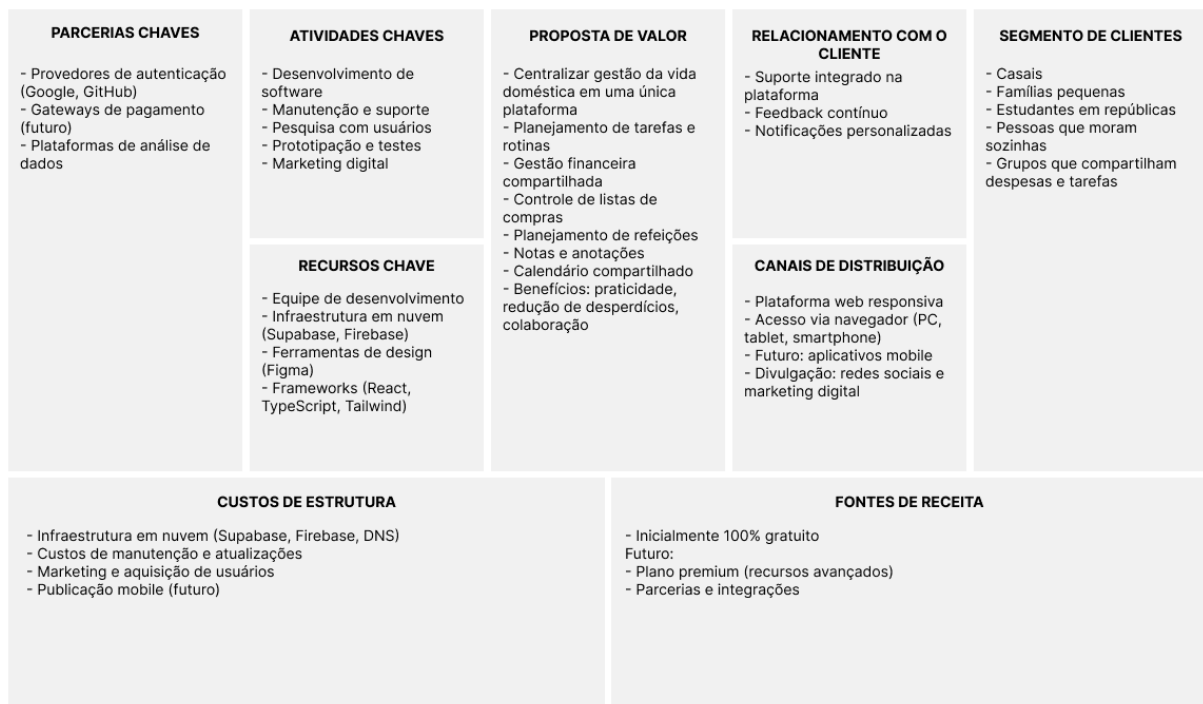


Figura 1 – *Business Model Canvas* do Convivu

Fonte: Elaborado pelo autor

3.2 Personas

A construção de personas foi fundamental para compreender as necessidades e expectativas dos usuários da plataforma Convivu. Essas representações fictícias, baseadas em entrevistas, observações e estudos de mercado, permitiram direcionar decisões de design e desenvolvimento.

- **Persona 1 - Marina e João**

Casal de 30 e 32 anos, moradores de São Paulo. Trabalham em tempo integral e têm pouco tempo para organizar a rotina doméstica. Buscam uma plataforma que permita dividir tarefas, controlar despesas mensais e planejar refeições para reduzir gastos.

- **Persona 2 - Camila**

Estudante universitária de 24 anos que divide apartamento com duas colegas. Utiliza o notebook e o celular para se organizar e precisa de um sistema prático para gerenciar compras coletivas e dividir contas de forma justa.

- **Persona 3 - Lucas**

Profissional de 27 anos que mora sozinho, mas costuma receber amigos e familiares. Utiliza a plataforma para manter a lista de compras atualizada e organizar eventos e encontros em casa.

- **Persona 4 - Rafael e Ana**

Casal jovem, 25 e 26 anos, recém-casados e em início de carreira. Procuram um sistema que ajude a organizar as finanças do lar, criar metas de economia e acompanhar compromissos do casal.

- **Persona 5 - Juliana**

Mãe solo de 35 anos, com uma filha pequena. Busca uma solução que centralize lembretes de compromissos escolares, listas de compras e organização de refeições, otimizando seu tempo limitado.

Essas personas foram utilizadas como referência em todas as etapas do desenvolvimento, desde a definição das funcionalidades até as decisões de design de interface e priorização de requisitos.

4 Desenvolvimento

De acordo com [Sommerville \(2015\)](#), o processo de *software* é crucial para o desenvolvimento de sistemas. No caso do Convivu, uma plataforma *web* para organização doméstica colaborativa, essa afirmação se traduz em um conjunto de decisões arquiteturais e tecnológicas que impactam diretamente a experiência do usuário, a segurança dos dados e a eficiência operacional. Este capítulo apresenta os elementos centrais do progresso do projeto, com foco no levantamento de requisitos, na camada de dados e serviços (baseada no *Supabase*), no *frontend* (*Vite*, *TypeScript* e *React*) e no processo de *deploy*.

O objetivo é oferecer uma visão clara das escolhas realizadas, justificando-as segundo critérios de eficácia, escalabilidade, segurança, manutenibilidade e alinhamento com os objetivos do Convivu.

4.1 Levantamento de Requisitos

Conforme [Wiegers e Beatty \(2013\)](#), a visão geral do sistema deve contemplar elementos essenciais obtidos por meio de entrevistas e observações dos principais interessados. No Convivu, foram conduzidas sessões exploratórias com potenciais usuários (casais, famílias pequenas e pessoas que compartilham moradia) para mapear tarefas recorrentes, pontos de dor e expectativas em relação a uma solução digital centralizada.

A análise de requisitos define o escopo, objetivos e funcionalidades necessárias, minimizando retrabalho e assegurando que o sistema atenda às necessidades reais dos usuários. Em linha com [Kotonya e Sommerville \(2017\)](#), a identificação e a especificação dos requisitos, mesmo que parciais, já permitem antecipar equívocos e orientar decisões técnicas subsequentes.

4.1.1 Requisitos Funcionais

Os requisitos funcionais (Tabela 2) descrevem as funções que o Convivu deve prover para sustentar a gestão doméstica colaborativa.

4.1.2 Requisitos Não-Funcionais

Os requisitos não-funcionais (Tabela 3) tratam de qualidades como desempenho, segurança e usabilidade.

Tabela 2 – Requisitos funcionais do Convivu

Fonte: Elaborado pelo autor

Requisito	Descrição
Autenticação e Perfis	Usuários devem poder criar conta, autenticar-se e gerenciar perfil. Suporte a redefinição de senha.
Lares (<i>households</i>) e Convites	Criação/entrada em um lar, convites para outros membros e controle de permissões (admin/membro).
Notas Colaborativas	Registro de notas e links de interesse comum com histórico de edição.
Finanças Domésticas	Registro e categorização de despesas/receitas, metas e visualização de saldo por período.
Listas de Compras	Criar listas, marcar itens concluídos, sugerir itens com base no histórico e <i>pantry</i> .
Planejamento de Refeições	Planejar cardápios, relacionar ingredientes e cruzar com a despensa/listas.
Calendário Compartilhado	Eventos do lar com lembretes e visualização por semana/mês.

Tabela 3 – Requisitos não-funcionais do Convivu

Fonte: Elaborado pelo autor

Requisito	Descrição
Desempenho	Páginas devem responder em até 2s sob carga típica; buscas e filtragens em até 500ms.
Segurança	Autenticação segura, políticas de acesso por linha (RLS) no banco
Usabilidade	Interface responsiva com navegação consistente.
Compatibilidade	Suporte a navegadores modernos e <i>devices</i> móveis (layout responsivo).
Manutenibilidade	Código modular tipado (TypeScript), <i>lint</i> , formatação automática e testes.
Escalabilidade	Banco relacional escalável, cache no cliente e estratégias de paginação.
Disponibilidade	Serviços núcleo com alta disponibilidade; recuperação de falhas documentada.

4.2 Camada de Dados e Serviços (*Supabase*)

A escolha do Supabase como backend da aplicação está relacionada à sua arquitetura baseada em serviços gerenciados sobre PostgreSQL. Diferentemente de um backend tradicional hospedado em servidor próprio, o Supabase opera em infraestrutura de nuvem que permite provisionamento automático de recursos conforme a demanda.

A escalabilidade do Supabase ocorre por meio de três mecanismos principais:

- **Infraestrutura gerenciada em nuvem:** os serviços são executados em ambientes containerizados, com capacidade de alocação dinâmica de CPU e memória. À medida que o número de requisições aumenta, a plataforma ajusta recursos automaticamente, reduzindo risco de indisponibilidade.
- **Banco PostgreSQL otimizado:** o PostgreSQL suporta índices, particionamento de tabelas, replicação e consultas eficientes. Em cenários de crescimento, é possível utilizar réplicas de leitura, balanceamento de carga e ajustes de configuração para suportar maior volume de dados e usuários simultâneos.
- **APIs geradas automaticamente:** o Supabase expõe endpoints REST e funcionalidades em tempo real baseadas em WebSockets, reduzindo a necessidade de desenvolvimento de camadas intermediárias. Isso diminui complexidade e melhora desempenho sob carga.

Além disso, o modelo *serverless* adotado permite que a aplicação escale horizontalmente, atendendo múltiplas conexões simultâneas sem que seja necessário gerenciar manualmente servidores. Esse fator é especialmente relevante para aplicações colaborativas, como o Convivu, nas quais múltiplos usuários podem acessar e modificar dados em tempo real.

Portanto, a escalabilidade do Supabase não se limita à capacidade de armazenar dados, mas envolve elasticidade de infraestrutura, eficiência do banco relacional e suporte a operações concorrentes de forma segura.

4.2.1 Esquema de Dados

O esquema relacional foi modelado para refletir entidades centrais do domínio:

- **events:** armazena eventos e compromissos do calendário compartilhado entre os membros do grupo familiar.
- **expenses:** registra despesas do grupo, categorizadas por tipo e associadas aos membros responsáveis.

- **goals**: define e acompanha metas financeiras ou pessoais do grupo, como poupança ou objetivos coletivos.
- **group_invites**: gerencia convites enviados para novos usuários participarem de um grupo familiar.
- **group_members**: mantém a relação entre perfis e grupos, incluindo papéis como membro ou administrador.
- **groups**: representa cada grupo familiar ou doméstico, centralizando a gestão dos seus integrantes.
- **incomes**: armazena informações sobre receitas ou entradas financeiras registradas pelos membros.
- **investments**: controla registros de aplicações financeiras do grupo, como fundos ou economias específicas.
- **meals**: organiza refeições planejadas, incluindo cardápios e preparações agendadas.
- **notes**: permite o registro de anotações colaborativas acessíveis aos membros do grupo.
- **profiles**: guarda dados de cada usuário, como informações pessoais, preferências e credenciais de acesso.
- **saved_meals**: armazena refeições já cadastradas para reutilização futura em planejamentos semanais.
- **shopping_items**: contém itens de compras adicionados às listas compartilhadas do grupo.
- **tasks**: registra tarefas domésticas a serem distribuídas entre os membros do grupo.
- **wishlist_items**: reúne desejos e objetivos de consumo, como produtos ou experiências que o grupo pretende adquirir.

A Figura 2 informa os diagramas de entidades.

4.2.1.1 Políticas de Segurança (RLS)

Em todas as tabelas do Convivu são aplicadas políticas de *Row Level Security* (RLS), mecanismo que restringe o acesso a registros específicos de acordo com o usuário logado. Assim, cada integrante acessa apenas dados pertencentes aos grupos familiares (*groups*) dos quais participa. Por exemplo, operações de **SELECT** na tabela **tasks** só são permitidas quando o usuário está vinculado ao respectivo **group_id** em **group_members**. Já

expenses	goals	group_members	meals	incomes
id	id	id	id	id
group_id	group_id	group_id	group_id	group_id
description	title	user_id	name	description
amount	description	role	date	amount
category	target_amount	joined_at	type	category
date	current_amount	events	ingredients	date
paid_by	target_date	id	notes	received_by
shared_with	priority	group_id	created_by	type
type	completed	title	created_at	created_by
created_by	created_by	description	updated_at	created_at
created_at	created_at	date	groups	updated_at
updated_at	updated_at	time	id	is_recurring
is_recurring	investments	type	name	recurring_type
recurring_type	id	priority	invite_code	installments
installments	group_id	created_by	created_by	installment_number
installment_number	description	created_at	created_at	original_income_id
original_expense_id	amount	updated_at	updated_at	profiles
notes	category	shopping_items	is_active	id
id	date	id	tasks	email
group_id	invested_by	group_id	id	name
title	type	name	group_id	avatar_url
content	goal_id	quantity	title	created_at
tags	is_recurring	category	description	updated_at
created_by	recurring_type	purchased	assigned_to	wishlist_items
created_at	installments	added_by	due_date	id
updated_at	installment_number	created_by	completed	group_id
saved_meals	original_investment_id	created_at	priority	title
id	created_by	updated_at	category	description
group_id	created_at	from_meal	created_by	estimated_price
name	updated_at	meal_date	created_at	priority
type		group_invites	updated_at	category
ingredients		id		added_by
notes		group_id		created_by
created_by		invited_user_id		created_at
created_at		invited_by_user_id		updated_at
updated_at		status		
		created_at		
		updated_at		

Figura 2 – Diagrama Entidades do Convivu

Fonte: Elaborado pelo autor

operações de INSERT, UPDATE ou DELETE em `group_members` dependem do papel do usuário: membros podem gerenciar apenas suas próprias informações, enquanto administradores podem adicionar ou remover integrantes e excluir o grupo.

4.3 Frontend

O *frontend* do Convivu é responsável por toda a camada de interação com o usuário, oferecendo uma interface intuitiva, responsiva e acessível. Essa camada concentra os elementos visuais e a lógica de apresentação, permitindo que os membros de um grupo familiar realizem de forma simples atividades como gerenciar tarefas, acompanhar finanças, organizar listas e visualizar eventos no calendário compartilhado.

4.3.1 Ambiente de Desenvolvimento

O *frontend* foi implementado com **Vite** (inicialização rápida e suporte a *Hot Module Replacement* – *HMR*, mecanismo que atualiza módulos modificados em tempo de execução sem a necessidade de recarregar toda a aplicação), **TypeScript** (tipagem estática), **ESLint** e **Prettier** (padronização e formatação de código fonte), garantindo qualidade e produtividade durante o desenvolvimento.

4.3.2 Bibliotecas

Essas bibliotecas foram escolhidas por sua maturidade, ampla adoção na comunidade e capacidade de acelerar o processo de desenvolvimento sem comprometer a qualidade do código.

- **React**: biblioteca principal para construção de interfaces declarativas e componentizadas.
- **@supabase/supabase-js**: cliente oficial do Supabase, utilizado para autenticação, acesso ao banco de dados e armazenamento de arquivos.
- **React Router DOM**: roteamento da aplicação no modelo SPA (*Single Page Application*).
- **Tailwind CSS e tailwind-merge**: framework utilitário para estilização rápida e consistente, com suporte ao **tailwind-merge** para composição eficiente de classes.
- **Radix UI e class-variance-authority**: bibliotecas que fornecem componentes acessíveis e estilização baseada em variantes, garantindo consistência no design.
- **Lucide React**: conjunto de ícones leves e escaláveis para uso em componentes visuais.
- **React Hot Toast**: gerenciamento de notificações (*toasts*) de forma simples e elegante.
- **Recharts**: criação de gráficos e visualizações de dados interativas.

- **React Day Picker e date-fns:** manipulação de datas, seleção em calendários e formatações amigáveis ao usuário.
- **UUID:** geração de identificadores únicos universais para entidades da aplicação.

4.3.3 Estrutura de Diretórios

A organização da estrutura de diretórios do projeto foi planejada para favorecer a clareza, a escalabilidade e a manutenção do código. A seguir, são descritas as principais pastas e arquivos que compõem o *frontend* do Convivu:

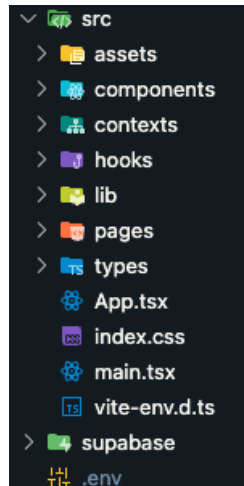
- **assets:** contém imagens, ícones e outros recursos estáticos.
- **components:** reúne componentes reutilizáveis da interface, como botões, modais e listas.
- **contexts:** provê contextos de estado global para autenticação, preferências e demais dados compartilhados.
- **hooks:** armazena *custom hooks* para abstração de lógica de negócio e integração com serviços.
- **lib:** funções auxiliares e integrações externas reutilizáveis em múltiplas partes da aplicação.
- **pages:** define as páginas e rotas principais da aplicação.
- **types:** declarações de tipos TypeScript compartilhadas.
- **supabase:** configuração do cliente Supabase e utilitários de integração com o backend.
- **App.tsx, main.tsx, index.css, vite-env.d.ts:** arquivos centrais de inicialização do projeto.
- **.env:** arquivo de variáveis de ambiente (não versionado) para configuração segura de chaves e credenciais.

A Figura 3 apresenta a organização atualmente adotada no *frontend* do Convivu.

4.3.4 Principais Funcionalidades do *Frontend*

Tarefas compartilhadas: lista com filtros por responsável, prioridade e status; arrasto para reordenar; conclusão com histórico;

Finanças domésticas: cadastro de despesas/receitas, categorias, metas mensais e gráficos de acompanhamento;

Figura 3 – Estruturação de pastas do *frontend*

Fonte: Elaborado pelo autor

Listas de compras e despensa: criação/edição rápida de itens; marcação de concluído; sincronização com a despensa e sugestões;

Planejamento de refeições: grade semanal, vínculo com ingredientes e checagem automática da disponibilidade;

Notas e calendário: notas colaborativas com histórico básico e calendário com lembretes (notificações do navegador).

4.4 *Deploy* e Hospedagem

Esta seção descreve o processo de preparação e publicação da aplicação Convivu em ambientes de produção, detalhando desde a compilação do código até a configuração da infraestrutura de dados.

4.4.1 *Build* e Variáveis de Ambiente

A aplicação, desenvolvida como uma *Single Page Application* (SPA) (aplicação de página única que carrega um único documento HTML e atualiza dinamicamente seu conteúdo via JavaScript), é compilada através do comando `yarn build`. O Vite, ferramenta moderna de *build* (processo de transformação do código-fonte em arquivos otimizados para produção) e *bundling* (empacotamento e agrupamento de módulos e dependências em arquivos finais), gera a pasta `dist` contendo os artefatos prontos para publicação.

Variáveis sensíveis, como `VITE_SUPABASE_URL` (endereço do backend do Supabase) e `VITE_SUPABASE_ANON_KEY` (chave de acesso anônima para autenticação), são injetadas a partir de arquivos de configuração por ambiente (`.env`), garantindo que credenciais e endpoints não fiquem expostos diretamente no código-fonte.

4.4.2 Hospedagem do *Frontend*

A versão estática gerada é publicada no *Firebase Hosting*, plataforma de hospedagem de aplicações web fornecida pela Google. Esse ambiente possibilita *deploy* (processo de publicação da aplicação em ambiente de produção) rápido, integração contínua com repositórios Git e suporte nativo a *redirects* (redirecionamentos automáticos de rota) para compatibilidade com o *client-side routing* (roteamento realizado no navegador, sem recarregar a página inteira). Além disso, o *cache* (armazenamento temporário de arquivos para agilizar carregamento) e o *hashing* (inclusão de identificadores únicos nos nomes de arquivos, evitando conflitos de versão) dos ativos estáticos asseguram melhor desempenho no carregamento da aplicação.

Entre suas principais vantagens estão a distribuição via *Content Delivery Network* (CDN, rede de servidores distribuídos geograficamente para entregar conteúdo com baixa latência), que garante baixa latência para usuários em diferentes regiões, a emissão automática de certificados *HTTPS* (protocolo seguro de comunicação web com criptografia) para segurança de comunicação e a escalabilidade transparente, permitindo que a aplicação suporte picos de acesso sem necessidade de configuração manual.

4.4.3 Infraestrutura de Dados

A camada de dados é provisionada pelo Supabase, uma plataforma de *Backend as a Service* (BaaS) que oferece banco relacional PostgreSQL, *storage* (armazenamento de arquivos) e autenticação de usuários. Migrações de esquema são versionadas por meio de scripts SQL, enquanto as políticas de segurança baseadas em *Row Level Security* (RLS), mecanismo que restringe acesso a registros específicos de acordo com o usuário, são tratadas como parte do código de infraestrutura, garantindo consistência entre ambientes de desenvolvimento e produção.

5 Resultados

A fase de implementação do Convivu resultou em uma plataforma web funcional e responsiva, atendendo aos requisitos propostos e oferecendo aos usuários um conjunto integrado de ferramentas para organização doméstica. A seguir, são apresentados os principais módulos e telas desenvolvidas.

- **Página Inicial** (Figura 4): funciona como o ponto de entrada para os usuários, apresentando uma visão geral das funcionalidades do Convivu, atalhos para módulos principais e informações resumidas sobre tarefas, eventos e finanças do lar. O layout prioriza clareza, responsividade e fácil navegação.
- **Calendário Colaborativo**: módulo que reúne compromissos, eventos e lembretes relevantes para o grupo familiar. Permite que os integrantes adicionem e visualizem datas importantes, como aniversários, consultas médicas, reuniões escolares e eventos sociais, garantindo que todos estejam informados.
- **Notas**: espaço destinado ao registro de informações relevantes e anotações rápidas que podem ser acessadas por todos os membros do grupo. As notas funcionam como lembretes colaborativos, podendo incluir listas de afazeres, ideias, registros de decisões ou informações importantes do cotidiano.

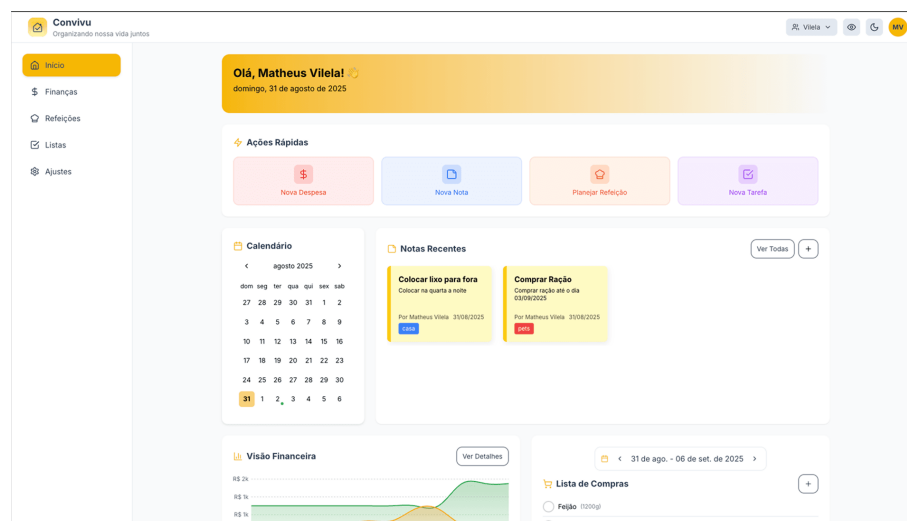


Figura 4 – Tela Inicial do Convivu

Fonte: Elaborado pelo autor

- **Gestão Financeira** (Figura 5): módulo para registro e categorização de receitas e despesas, definição de metas financeiras e acompanhamento do saldo do lar. Os gráficos interativos permitem visualizar tendências e auxiliar na tomada de decisões.

- **Metas** (Figura 6): módulo voltado para o acompanhamento de objetivos financeiros do grupo familiar. Cada meta possui um valor-alvo, data limite de conclusão e indicador visual de progresso. O usuário pode adicionar novos aportes de forma prática através do botão *Investir*, permitindo monitorar a evolução ao longo do tempo. Essa funcionalidade auxilia no planejamento financeiro coletivo, oferecendo maior transparência e organização para atingir objetivos comuns.
- **Desejos** (Figura 7): módulo destinado ao registro de itens ou experiências que o grupo deseja conquistar, como viagens, bens de consumo ou atividades de lazer. Diferente das metas financeiras, que possuem prazos e valores definidos, os desejos funcionam como uma lista de aspirações compartilhadas. Essa funcionalidade possibilita priorizar objetivos futuros e alinhar expectativas entre os membros do grupo familiar.

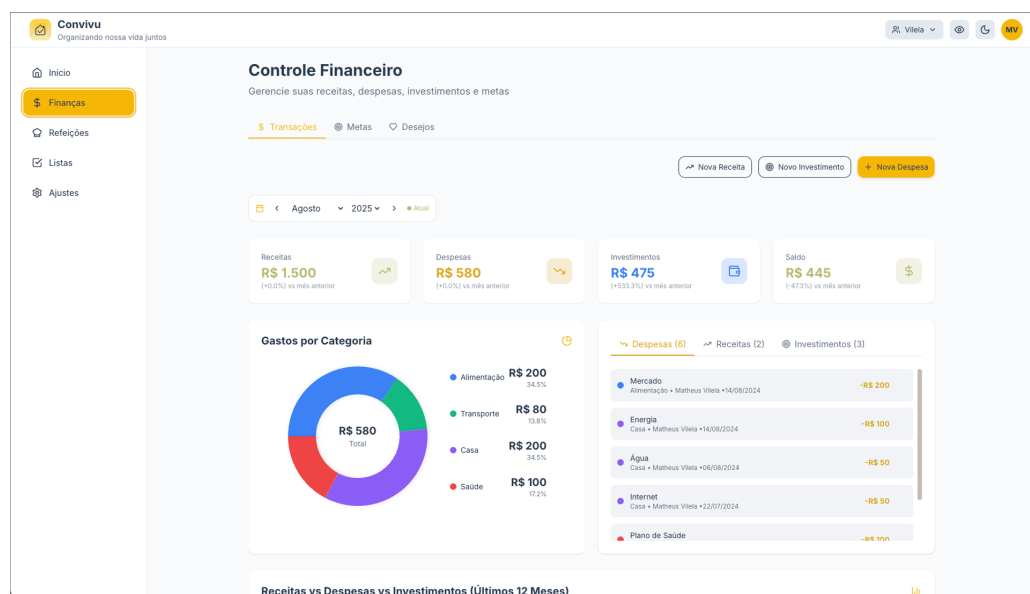


Figura 5 – Gestão Financeira

Fonte: Elaborado pelo autor

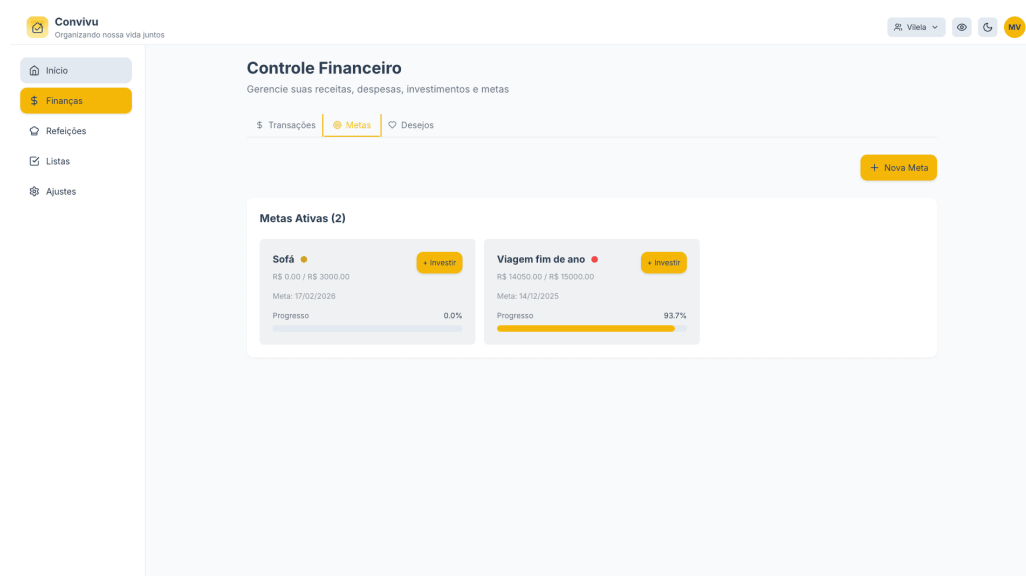


Figura 6 – Metas

Fonte: Elaborado pelo autor

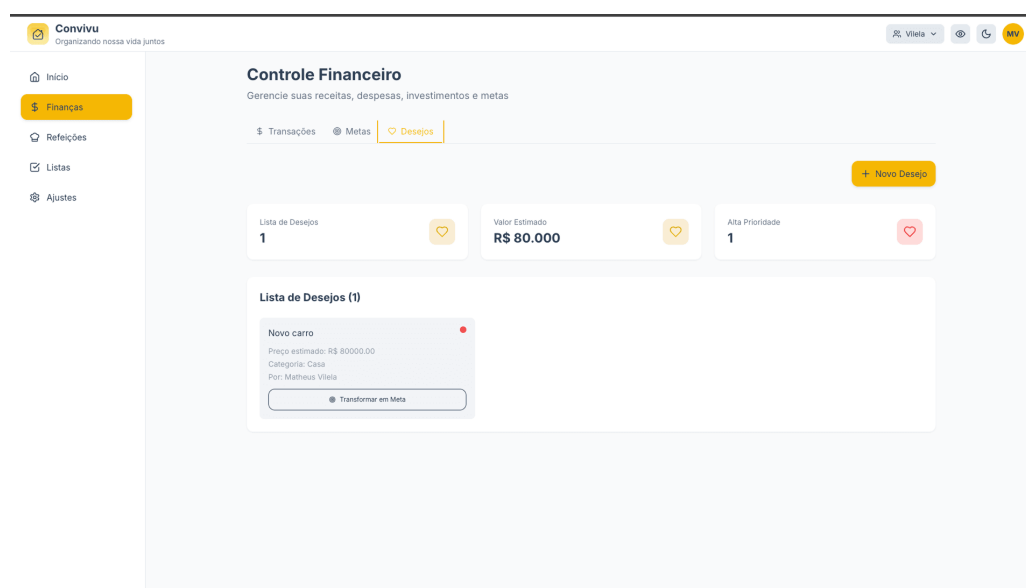


Figura 7 – Desejos

Fonte: Elaborado pelo autor

- **Planejamento de Refeições** (Figura 8): possibilita organizar cardápios semanais e mensais, vinculando cada refeição aos ingredientes necessários. Os itens utilizados são adicionados automaticamente à lista de compras correspondente (semanal ou mensal), o que contribui para otimizar a reposição de alimentos e reduzir desperdícios no lar.
- **Refeições Salvas** (Figura 10): funcionalidade que permite ao usuário cadastrar e armazenar refeições completas para utilização futura. Ao selecionar uma refeição salva, todos os alimentos e suas quantidades são preenchidos automaticamente, agili-

zando o processo de planejamento semanal ou mensal. Esse recurso é especialmente útil para receitas recorrentes no dia a dia da família, evitando retrabalho e garantindo padronização das porções. Além disso, os ingredientes podem ser integrados diretamente à lista de compras, mantendo a coerência entre o cardápio planejado e os itens adquiridos. Dessa forma, o recurso contribui para maior praticidade, redução do tempo gasto na organização e melhor controle dos insumos domésticos.

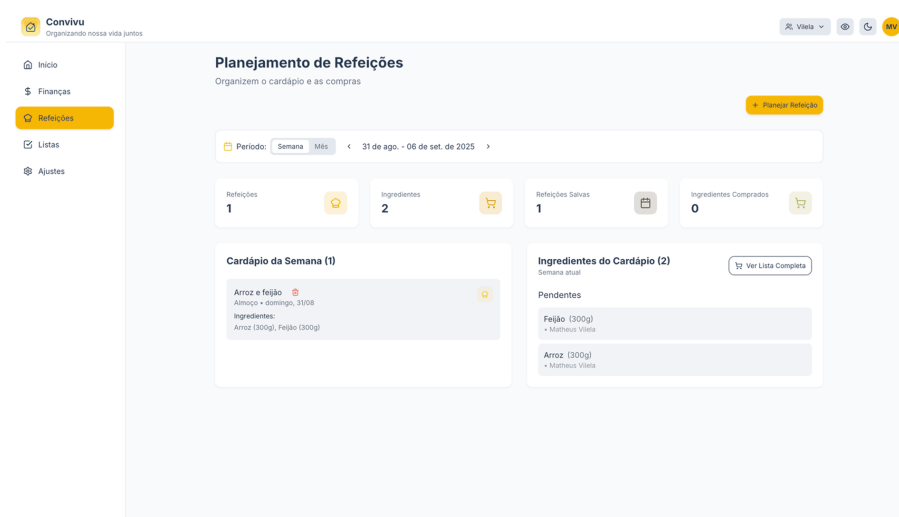


Figura 8 – Planejamento de Refeições

Fonte: Elaborado pelo autor

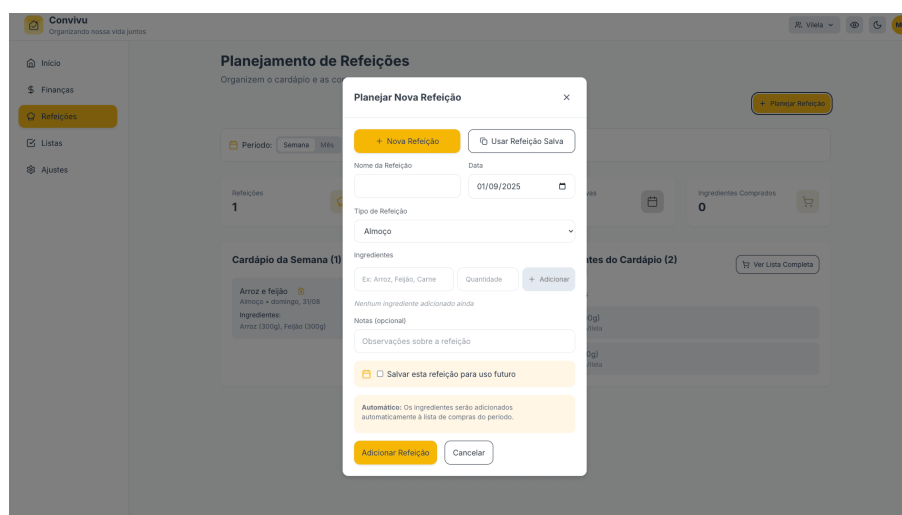


Figura 9 – Adicionar refeição

Fonte: Elaborado pelo autor

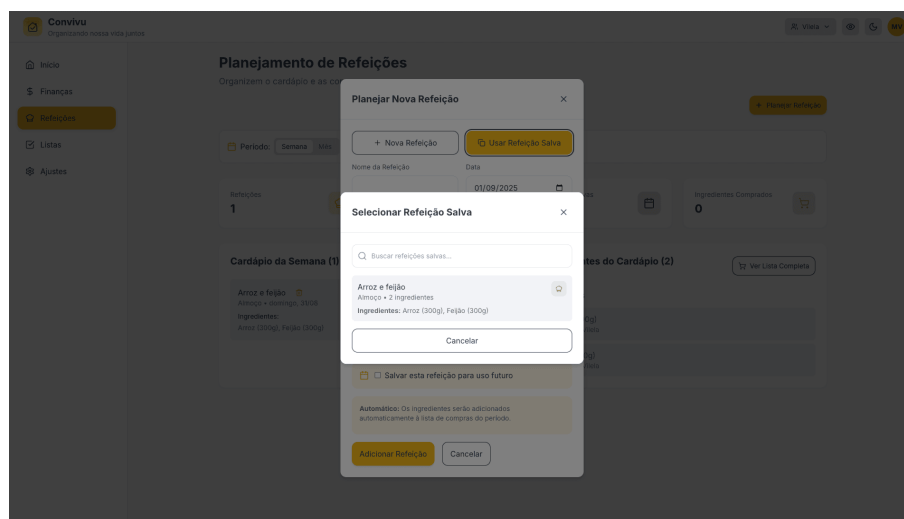


Figura 10 – Refeições Salvas

Fonte: Elaborado pelo autor

- **Lista de Compras** (Figura 11): módulo que centraliza os itens necessários para o lar, permitindo que os membros adicionem, editem e marquem produtos adquiridos em tempo real. A lista principal é criada automaticamente a partir do planejamento de refeições, garantindo alinhamento entre o cardápio definido e a reposição de insumos. Além disso, o sistema possibilita a criação de listas adicionais personalizadas, úteis para situações específicas como viagens, festas ou compras eventuais. Como o recurso é compartilhado entre todos os integrantes do grupo familiar, ele promove colaboração, reduz duplicidades e otimiza o controle do orçamento doméstico.

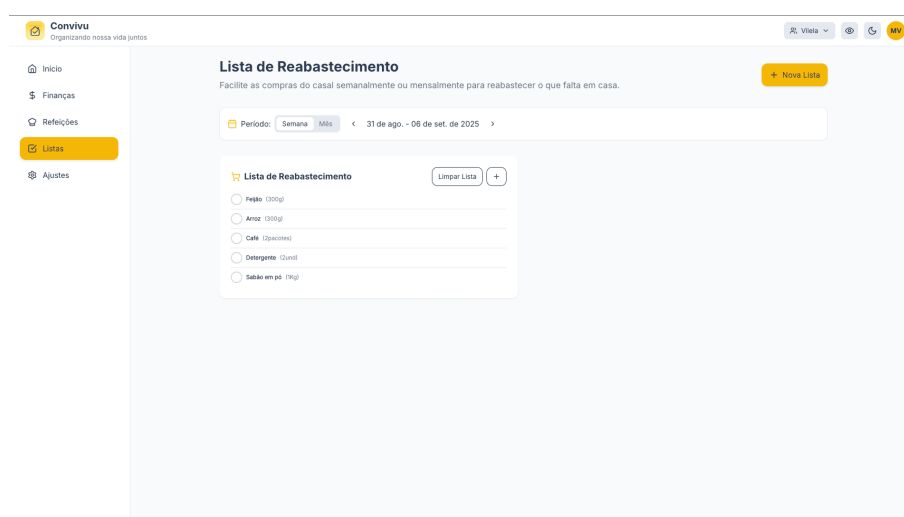


Figura 11 – Lista de Compras

Fonte: Elaborado pelo autor

- **Configurações do Grupo** (Figura 12): módulo responsável pela administração do grupo familiar no Convivu. Nessa tela, os administradores podem editar o

nome do grupo, convidar novos integrantes por meio de convites digitais, remover membros já cadastrados e, caso necessário, excluir o grupo por completo. Essas funcionalidades concentram os controles de gestão em um único espaço, oferecendo praticidade e transparência na administração do lar. Além disso, ao centralizar ações de gerenciamento, o sistema garante que apenas usuários com perfil de administrador tenham acesso a operações críticas, preservando a segurança e a integridade das informações compartilhadas.

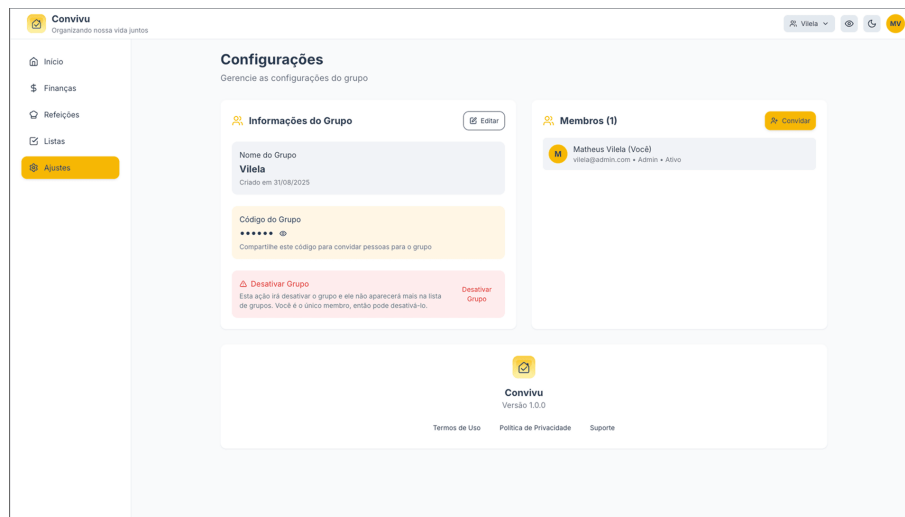


Figura 12 – Configurações do Grupo

Fonte: Elaborado pelo autor

- **Autenticação de Usuários** (Figuras 13): telas de login e cadastro foram projetadas para serem simples e objetivas, solicitando apenas as informações essenciais. A autenticação é gerenciada pelo Supabase, garantindo segurança e escalabilidade.

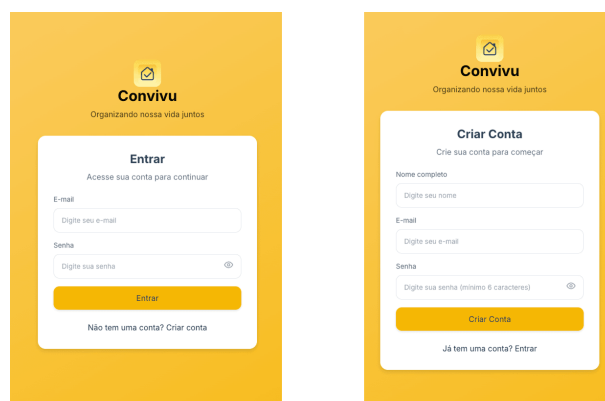


Figura 13 – Tela de Login e cadastro

Fonte: Elaborado pelo autor

- **Grupo Familiar** (Figura 14): tela inicial de gerenciamento de lares dentro do Convivu. Nela, o usuário pode criar um novo grupo familiar, ingressar em um grupo

existente por meio de convite ou selecionar diretamente um grupo ao qual já esteja vinculado. Essa funcionalidade centraliza o acesso e a gestão dos diferentes lares dos quais o usuário participa, permitindo alternar facilmente entre contextos distintos (por exemplo, residência principal, casa de veraneio ou república). Dessa forma, o sistema assegura flexibilidade e praticidade, garantindo que cada ambiente possua seus próprios membros, configurações e informações compartilhadas.



Figura 14 – Grupo Familiar

Fonte: Elaborado pelo autor

- **Cabeçalho da Aplicação** (Figura 15): elemento fixo de navegação presente em toda a interface do Convivu, reunindo funcionalidades que aumentam a usabilidade e a personalização da experiência do usuário. O cabeçalho oferece quatro recursos principais:
 - **Seletor de Grupo Familiar**: permite alternar rapidamente entre diferentes grupos familiares dos quais o usuário participa, facilitando a gestão de múltiplos lares em um único ambiente.
 - **Ocultar Valores Monetários**: recurso de privacidade que possibilita ocultar saldos, despesas e demais valores financeiros exibidos na tela, evitando exposição indevida em situações públicas.
 - **Tema Escuro/Claro**: alternância entre *dark mode* e modo claro, adaptando a interface às preferências do usuário e proporcionando maior conforto visual em diferentes condições de uso.
 - **Configurações de Usuário**: área destinada à edição de dados pessoais, troca de senha e opção de encerrar sessão, centralizando ações de gerenciamento da conta em um único ponto.

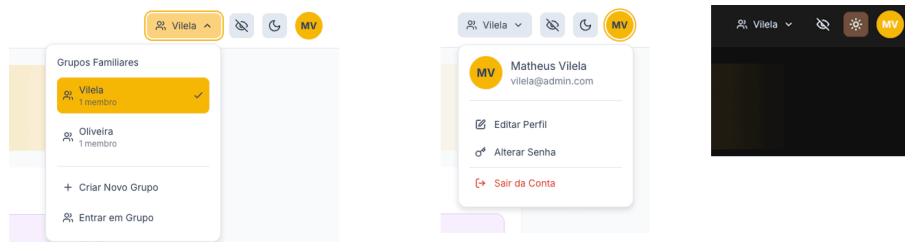


Figura 15 – Cabeçalho da Aplicação

Fonte: Elaborado pelo autor

- **Responsividade para Dispositivos Móveis** (Figura 16): todo o sistema foi desenvolvido com design responsivo, adaptando-se automaticamente a diferentes tamanhos de tela, desde computadores até tablets e celulares. Essa característica garante usabilidade consistente em múltiplos dispositivos, permitindo que os membros do grupo familiar acessem e utilizem as funcionalidades do Convivu em qualquer contexto, seja em casa ou em movimento.

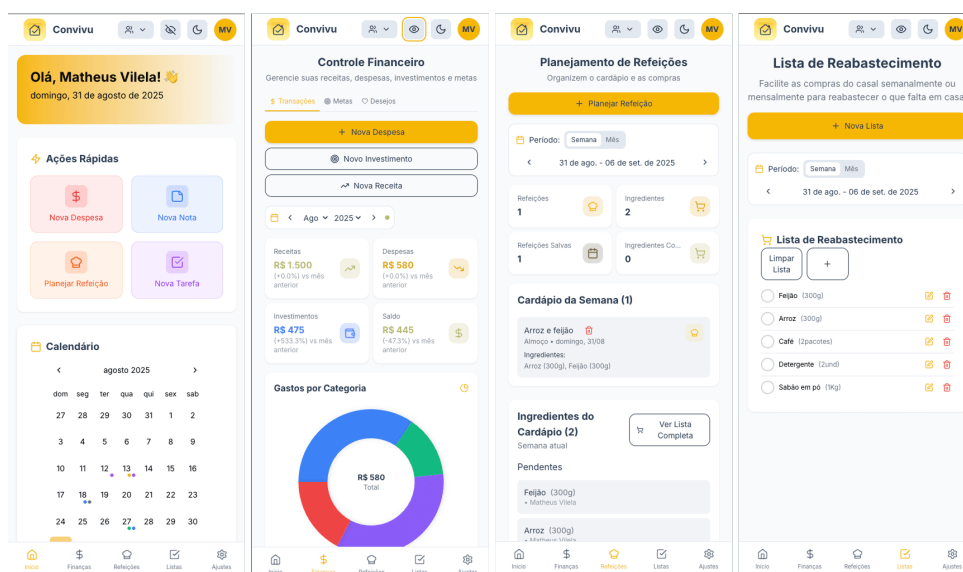


Figura 16 – Responsividade para Dispositivos Móveis

Fonte: Elaborado pelo autor

A implementação das funcionalidades apresentadas confirma a viabilidade técnica e a aplicabilidade prática do Convivu, evidenciando sua capacidade de atender aos requisitos propostos e de proporcionar uma experiência consistente, responsiva e intuitiva aos usuários.

O Convivu encontra-se publicado em ambiente de homologação e pode ser acessado pelo endereço:

[<https://tcc-web-site.web.app/>](https://tcc-web-site.web.app/)

6 Conclusão

Este trabalho apresentou o desenvolvimento do Convivu, uma plataforma web voltada à organização colaborativa da vida doméstica. Ao longo do projeto, foram descritas as etapas de modelagem, definição de requisitos, prototipação e implementação técnica da solução.

A aplicação integra módulos de finanças, tarefas, listas de compras, planejamento de refeições, calendário e notas, permitindo que membros de um mesmo grupo compartilhem informações e acompanhem responsabilidades de maneira estruturada. A adoção de princípios de design centrado no usuário contribuiu para a construção de uma interface simples e consistente.

Do ponto de vista técnico, a utilização de React e TypeScript favoreceu organização e manutenibilidade do código, enquanto o Supabase proporcionou infraestrutura segura e escalável para armazenamento de dados e autenticação de usuários.

Os objetivos estabelecidos foram atingidos, resultando em uma aplicação funcional, responsiva e passível de evolução futura.

6.1 Trabalhos futuros

Mesmo com a entrega completa da versão proposta, há diversas possibilidades de evolução. Entre as melhorias e funcionalidades que poderão ser exploradas em versões futuras, destacam-se:

Integração com assistentes virtuais: implementar integração com assistentes como Google Assistente e Alexa para permitir criação de tarefas, adição à lista de compras e consulta de eventos por comando de voz;

Sistema de notificações inteligentes: adicionar notificações personalizadas, enviadas com base em contexto e comportamento do usuário (por exemplo, lembrar de comprar um item quando o estoque na despensa virtual estiver baixo);

Relatórios financeiros avançados: oferecer relatórios gráficos de despesas e metas, com filtros por tendência e padrões de uso, auxiliando na tomada de decisões financeiras do lar;

Integração com APIs externas: incorporar integração com APIs de supermercados e aplicativos de entrega para facilitar compras a partir da lista de compras do Convivu;

Testes de usabilidade e acessibilidade: ampliar a cobertura de testes com usuários reais, verificando a aderência às diretrizes de acessibilidade para garantir uso inclusivo da plataforma;

Aplicativo Mobile: desenvolver e disponibilizar uma versão nativa do Convivu para dispositivos móveis, publicada nas lojas oficiais (Google Play e App Store). Essa iniciativa ampliará o alcance da plataforma, oferecendo acesso simplificado, notificações nativas e melhor desempenho em smartphones e tablets.

Modo offline: permitir que os usuários utilizem funcionalidades básicas do sistema mesmo sem conexão à internet, sincronizando os dados automaticamente quando o acesso for restabelecido.

Com a continuidade do desenvolvimento e adoção das melhorias propostas, o **Convivu** tem potencial para se consolidar como referência em soluções digitais voltadas à vida doméstica, unindo tecnologia, praticidade e colaboração.

Referências

Atlassian. *Trello*. 2023. Acesso em: 19 ago. 2025. Disponível em: <<https://trello.com>>. Citado na página 22.

Between. *Between App*. 2023. Acesso em: 19 ago. 2025. Disponível em: <<https://between.us>>. Citado na página 22.

COOPER, A. et al. *About Face: The Essentials of Interaction Design*. 4. ed. Indianapolis: Wiley, 2014. Citado na página 18.

Cozi. *Cozi Family Organizer*. 2023. Acesso em: 19 ago. 2025. Disponível em: <<https://www.cozi.com>>. Citado na página 22.

FERREIRA, L. M. Análise comparativa de aplicativos de organização doméstica. *Revista de Sistemas Colaborativos*, v. 5, n. 1, p. 33–48, 2021. Citado na página 22.

FIGMA. *Figma: The Collaborative Interface Design Tool*. 2022. url<https://www.figma.com/>. Acesso em: 19 ago. 2025. Citado na página 21.

Google. *Google Keep*. 2023. Acesso em: 19 ago. 2025. Disponível em: <<https://keep.google.com>>. Citado na página 22.

Honeydue. *Honeydue*. 2023. Acesso em: 19 ago. 2025. Disponível em: <<https://www.honeydue.com>>. Citado na página 22.

KOTONYA, G.; SOMMERVILLE, I. *Requirements Engineering*. Chichester: Wiley, 2017. Citado na página 26.

Love Nudge. *Love Nudge App*. 2023. Citado na página 22.

Merge. *Merge App*. 2023. Aplicativo disponível em lojas móveis. Citado na página 22.

META, I. *React – A JavaScript library for building user interfaces*. [S.l.], 2023. Acesso em: 31 ago. 2025. Disponível em: <<https://react.dev/>>. Citado na página 20.

Microsoft. *TypeScript: JavaScript With Syntax For Types*. 2022. url<https://www.typescriptlang.org/>. Acesso em: 19 ago. 2025. Citado na página 19.

NORMAN, D. A. *The Design of Everyday Things*. Revised and expanded edition. New York: Basic Books, 2018. Citado na página 17.

Notion. *Notion*. 2023. Acesso em: 19 ago. 2025. Disponível em: <<https://www.notion.so>>. Citado na página 22.

OurGroceries. *OurGroceries Shopping List*. 2023. Acesso em: 19 ago. 2025. Disponível em: <<https://www.ourgroceryapp.com>>. Citado na página 22.

Raft. *Raft App*. 2023. Citado na página 22.

RODRIGUES, P. H. Gestão doméstica com tecnologias digitais: impacto em rotinas familiares. *Revista Brasileira de Tecnologias Aplicadas*, v. 9, n. 2, p. 55–70, 2020. Citado na página 17.

- ROGERS, Y.; SHARP, H.; PREECE, J. *Interaction Design: Beyond Human-Computer Interaction*. 5. ed. New York: Wiley, 2019. Citado na página 18.
- SILVA, K. B.; OLIVEIRA, J. S.; SANTOS, E. Organização doméstica digital: ferramentas e práticas de gestão do lar. In: *Anais do Congresso Brasileiro de Informática na Educação*. Brasília: [s.n.], 2021. Citado na página 17.
- SOMMERVILLE, I. *Software Engineering*. 10. ed. Boston: Pearson, 2015. Citado na página 26.
- Splitwise. *Splitwise*. 2023. Acesso em: 19 ago. 2025. Disponível em: <<https://www.splitwise.com>>. Citado na página 22.
- SUPABASE. *Supabase: The Open Source Firebase Alternative*. 2023. url<https://supabase.com/>. Acesso em: 19 ago. 2025. Citado na página 18.
- Tandem. *Tandem App*. 2023. Citado na página 22.
- WIEGERS, K.; BEATTY, J. *Software Requirements*. 3. ed. Redmond: Microsoft Press, 2013. Citado na página 26.
- YNAB. *You Need A Budget*. 2023. Acesso em: 19 ago. 2025. Disponível em: <<https://www.ynab.com>>. Citado na página 22.