

UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS
DEPARTAMENTO DE ENGENHARIA DA COMPUTAÇÃO
CURSO DE ENGENHARIA DE COMPUTAÇÃO

VICTOR EMANUEL SANTOS MOURA

Marandu Hub: Desenvolvimento de um Sistema Low-Code para Gerenciamento de Espaços
de Coworking com FlutterFlow

SÃO LUÍS

2025

VICTOR EMANUEL SANTOS MOURA

Marandu Hub: Desenvolvimento de um Sistema Low-Code para Gerenciamento de Espaços de Coworking com FlutterFlow

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia da Computação da Universidade Estadual do Maranhão - UEMA como requisito para obtenção do grau de Bacharel em Engenharia da Computação, sob orientação do Prof. Dr. Antônio Fernando Lavareda Jacob Junior.

SÃO LUÍS

2025

Moura, Victor Emanuel Santos.

Marandu Hub: desenvolvimento de um sistema low-code para gerenciamento de espaços de coworking com flutterflow / Victor Emanuel Santos Moura. - São Luís - MA, 2025.

70 f.

Monografia (Graduação em Engenharia de Computação) - Universidade Estadual do Maranhão, São Luís, 2025.

Orientador: Prof. Dr. Antônio Fernando Lavareda Jacob Junior.

1. sistema de reservas. 2. coworking. 3. FlutterFlow. 4. Firebase. 5. low-code. I. Título.

CDU: 004.4'2

Elaborado por Cassia Valéria da Silva Diniz - CRB 13/910

VICTOR EMANUEL SANTOS MOURA

Marandu Hub: Desenvolvimento de um Sistema Low-Code para Gerenciamento de Espaços de Coworking com FlutterFlow

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia da Computação da Universidade Estadual do Maranhão - UEMA como requisito para obtenção do grau de Bacharel em Engenharia da Computação.

Aprovado em: 24 / 07 / 2025

BANCA EXAMINADORA



Documento assinado digitalmente
ANTONIO FERNANDO LAVAREDA JACOB JUNIOR
Data: 11/08/2025 14:44:49-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Antônio Fernando Lavareda Jacob Junior (Orientador)

Doutor em Engenharia Elétrica (UFMA)



Documento assinado digitalmente
ANTONIO ROBERTO COELHO SERRA
Data: 07/08/2025 20:03:57-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Antônio Roberto Coelho Serra

Doutor em Administração (FGV EBAPE)



Documento assinado digitalmente
ARIANA BARROS PIMENTEL
Data: 11/08/2025 14:37:03-0300
Verifique em <https://validar.iti.gov.br>

Profa. Ma. Ariana Barros Pimentel

Mestra em Contabilidade e Administração (FUCAPE Business School)



Documento assinado digitalmente
WESLEY BATISTA DOMINICES DE ARAUJO
Data: 07/08/2025 16:37:32-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Wesley Batista Dominices de Araujo

Doutor em Engenharia Elétrica (UFMA)

Dedico este trabalho aos meus pais, Célia e Cristiano. Sob o calor do sol, seu amor foi o abrigo e sombra que me permitiu crescer e chegar até aqui.

AGRADECIMENTOS

Chegar até aqui foi um caminho repleto de desafios, aprendizados e conquistas. Este trabalho é muito mais do que um requisito acadêmico — ele representa a soma de esforços, incentivos e amor de pessoas que caminharam ao meu lado durante toda essa jornada.

Agradeço, primeiramente, à minha família, meu maior alicerce. Aos meus pais, Célia e Cristiano, por me ensinarem, com amor e exemplo, o valor da persistência, do esforço e da honestidade. A vocês, devo tudo. Aos meus irmãos, Valdivino Neto e José Vinícius, pelo companheirismo, pelas conversas, pelas risadas e por estarem sempre presentes, mesmo nos silêncios.

À minha namorada, Anna Izabel, minha parceira de vida, obrigado por todo apoio, paciência, carinho e incentivo. Sua presença tornou os momentos difíceis mais leves e os momentos bons ainda mais especiais.

Quero também deixar um agradecimento especial aos meus avós. Ao meu avô Valdivino, que hoje brilha no céu — ele certamente estaria radiante ao me ver me formar, como o primeiro da família a conquistar um diploma de ensino superior. Essa vitória também é sua, vô Valdivino. Onde quer que esteja, sei que está orgulhoso e em festa, assim como adorava estar. Às minhas queridas avós, Vilma e Mariquinha, que sempre me apoiaram e estiveram ao meu lado em cada passo dessa jornada, deixo meu carinho e gratidão. Aos meus tios e primos, que sempre torceram por mim e celebraram cada conquista, meu eterno agradecimento.

Agradeço imensamente ao professor Antônio Jacob, meu orientador, por sua calma, paciência e pelas cobranças na medida certa, que foram fundamentais para meu crescimento acadêmico e pessoal. Sua orientação foi essencial para que este projeto se concretizasse.

Também agradeço à Universidade Estadual do Maranhão (UEMA), por ter sido o espaço onde pude me desenvolver, aprender e crescer como estudante e como pessoa.

Aos meus amigos, desde os de infância até os que a vida universitária me presenteou: obrigado pelas conversas, pelos conselhos, pelas piadas, por cada aventura e por me lembrarem que nenhuma jornada precisa ser solitária.

Também deixo aqui meu sincero agradecimento a todos os professores que cruzaram meu caminho ao longo da minha vida. Cada um, à sua maneira, contribuiu para a formação do profissional e da pessoa que me tornei.

Por fim, agradeço a todos que, direta ou indiretamente, fizeram parte dessa caminhada. Esse trabalho é nosso.

“A tecnologia pode e deve ser usada para melhorar o mundo, ajudando as pessoas a realizar mais e fazer mais”.

Sheryl Sandberg

RESUMO

Este estudo apresenta o desenvolvimento de um sistema de gerenciamento de reservas para a Marandu Hub, espaço de *coworking* mantido pela Marandu – Agência UEMA de Inovação e Empreendedorismo. Com a expansão dos ambientes colaborativos e a necessidade de soluções mais práticas para organizar o uso desses espaços, principalmente em contextos acadêmicos, surgiu a proposta de automatizar o processo de agendamento de salas. A plataforma foi construída com o FlutterFlow, uma ferramenta *low-code* que permite criar aplicativos de forma visual, rápida e com boa escalabilidade. O sistema substitui os antigos métodos manuais de reserva, facilitando o uso das salas e evitando conflitos de horário. Entre as funções implementadas, destacam-se o login seguro, os painéis de controle para administradores, o envio automático de notificações por e-mail e a geração de relatórios dinâmicos. A pesquisa teve caráter aplicado, com abordagem descritiva e qualitativa, buscando soluções que atendessem efetivamente às necessidades da instituição. Foram utilizadas tecnologias como o Firebase (para login e banco de dados), o Node.js (na criação de uma *API* para relatórios em Excel) e o EmailJS (responsável pelo envio dos e-mails automáticos). O sistema foi pensado para atender perfis distintos de usuários e recebeu ajustes para funcionar bem tanto em celulares quanto em computadores. Para avaliar a experiência dos usuários, foi utilizada a Escala SUS (*System Usability Scale*), que indicou uma média de 94,5 pontos, demonstrando que a aplicação foi bem recebida. O sistema já está em uso e tem contribuído para uma gestão mais eficiente, servindo de exemplo sobre como ferramentas *low-code* podem ser úteis na modernização de processos institucionais com poucos recursos.

Palavras-chave: sistema de reservas; *coworking*; FlutterFlow; Firebase; *low-code*.

ABSTRACT

This study presents the development of a reservation management system for Marandu Hub, a coworking space maintained by Marandu – UEMA Agency for Innovation and Entrepreneurship. With the expansion of collaborative environments and the need for more practical solutions to organize the use of these spaces, particularly in academic contexts, the proposal to automate the room scheduling process emerged. The platform was built with FlutterFlow, a low-code tool that allows for the visual, rapid, and scalable creation of applications. The system replaces outdated manual reservation methods, facilitating room usage and preventing scheduling conflicts. Implemented features include secure login, administrator control panels, automatic email notifications, and dynamic report generation. The research adopted an applied approach with descriptive and qualitative methods, seeking solutions that effectively met the institution's needs. Key technologies used were Firebase (for user authentication and database), Node.js (to create an API for Excel reports), and EmailJS (for sending automated emails). The system was designed to serve distinct user profiles and was adjusted to function well on both mobile phones and computers. To evaluate the user experience, the System Usability Scale (SUS) was applied, resulting in an average score of 94.5 points, which demonstrates that the application was well-received. The system is currently in use and has contributed to more efficient management, serving as an example of how low-code tools can be valuable for modernizing institutional processes with limited resources.

Keywords: reservation system; coworking; FlutterFlow; Firebase; low-code.

LISTA DE ILUSTRAÇÕES

Figura 1 - Projeção de Crescimento do Mercado de Low-Code.....	21
Figura 2 - Arquitetura em Camadas do Sistema Marandu Hub.....	36
Figura 3 - Tela inicial do sistema (Home).....	37
Figura 4 - Outro trecho da página Home.....	38
Figura 5 - Tentativa de reserva com alerta de reserva indisponível.....	39
Figura 6 - Tela de reservas com foco nas solicitadas.....	40
Figura 7 - E-mail para reserva solicitadas.....	41
Figura 8 - E-mail para reserva confirmadas.....	41
Figura 9 - E-mail para reserva canceladas.....	41
Figura 10 - Tela de login.....	42
Figura 11 - Tela de cadastro com e-mail.....	43
Figura 12 - Tela de finalização de conta (dados institucionais).....	43
Figura 13 - Mensagem indicando CPF vinculado a uma empresa incubada.....	44
Figura 14 - Listagem com as reservas registradas.....	49
Figura 15 - Filtros para relatórios.....	50
Figura 16 - Dashboard de reservas por sala.....	51
Figura 17 - Dashboard por status.....	51
Figura 18 - Dashboard de reservas por empresa.....	51
Figura 19 - Dashboard total de horas utilizadas por empresa.....	51
Figura 20 – Dashboard de evolução temporal.....	52
Figura 21 - Custom Widget criado.....	54
Gráfico 1 - Média das Respostas por Item da Escala SUS.....	57

LISTA DE TABELAS

Tabela 1 - Análise Comparativa das Principais Plataformas Low-Code.....	25
Tabela 2 - Estrutura de Dados da Coleção users.....	45
Tabela 3 - Estrutura de Dados da Coleção rooms.....	45
Tabela 4 - Estrutura de Dados da Coleção reservations.....	46
Tabela 5 - Estrutura de Dados da Coleção incubatedCompanies.....	46
Tabela 6 - Níveis de acesso.....	47

SUMÁRIO

1 INTRODUÇÃO.....	12
1.1 Objetivos.....	13
1.1.1 Objetivo Geral.....	13
1.1.2 Objetivos Específicos.....	14
1.2 Justificativa.....	14
1.3 Estrutura do trabalho.....	15
2 FUNDAMENTAÇÃO TEÓRICA.....	16
2.1 Coworking.....	16
2.1.1 O que é coworking?.....	16
2.1.2 Origem e evolução do coworking.....	16
2.1.3 Coworking no Brasil.....	17
2.1.4 Benefícios do Coworking.....	18
2.2 Plataformas de Desenvolvimento Low-Code.....	19
2.2.1 Definição, Características e Classificações.....	20
2.2.2 Vantagens do Desenvolvimento Low-Code.....	21
2.2.3 Limitações e Desafios do Low-Code.....	22
2.2.4 Principais Plataformas de Low-Code.....	24
2.3 FlutterFlow no Contexto de Desenvolvimento Ágil com Low-Code.....	25
2.3.1 Visão Geral e Recursos Chave.....	25
2.3.2 Benefícios Observados.....	27
2.3.3 Contextos de Uso e Popularidade do FlutterFlow.....	27
3 METODOLOGIA.....	29
3.1 Tipo de Pesquisa.....	29
3.2 Procedimentos Metodológicos.....	29
3.3 Ferramentas e Tecnologias Utilizadas.....	31
3.4 Etapas do Projeto.....	31
3.4.1 Levantamento de Requisitos.....	31
3.4.2 Prototipação Visual no FlutterFlow.....	32

3.4.3	Integração com Firebase.....	32
3.4.4	Criação de Lógica de Negócio.....	32
3.4.5	Painel Administrativo e Relatórios.....	33
3.5	Validação e Avaliação do Sistema.....	33
4	DESENVOLVIMENTO.....	34
4.1	Aplicação no Projeto Marandu Hub.....	34
4.2	Arquitetura Geral da Aplicação.....	35
4.2.1	Camada de Apresentação (Frontend - FlutterFlow).....	37
4.2.2	Camada de Lógica e Dados (FlutterFlow + Firebase + Dart).....	38
4.2.3	Camada de Integração Externa (Node.js + API em Vercel).....	39
4.2.4	Camada de Comunicação (EmailJS).....	39
4.2.5	Camada de Autenticação e Segurança (Firebase Auth + Firestore Rules).....	42
4.3	Estrutura de Dados (Coleções Firebase).....	44
4.3.1	Coleção users: Cadastro e Perfil de Usuários.....	44
4.3.2	Coleção rooms: Cadastro de Salas.....	45
4.3.3	Coleção reservations: Controle de Reservas.....	46
4.3.4	Coleção incubatedCompanies: Empresas Incubadas.....	46
4.4	Controle de Acessos e Perfis de Usuário.....	47
4.4.1	Usuários Vinculados a Empresas Incubadas.....	47
4.4.2	Usuários Não Vinculados a Empresas Incubadas.....	48
4.5	Módulo de Relatórios e Visualizações.....	49
4.5.1	Visualização em Tabela.....	49
4.5.2	Exportação em Excel.....	49
4.5.3	Dashboards Visuais.....	50
4.6	Funcionalidades Personalizadas e Regras de Negócio.....	52
4.7	Hospedagem do Sistema.....	54
5	RESULTADOS E DISCUSSÃO.....	55
5.1	Avaliação da Usabilidade com a Escala SUS.....	55
5.2	Estrutura da Escala e Cálculo.....	55

5.3 Análise dos Resultados.....	56
5.4 Análise Crítica e Trabalhos Futuros.....	57
6 CONSIDERAÇÕES FINAIS.....	61
REFERÊNCIAS.....	63
APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE).....	66
APÊNDICE B – QUESTIONÁRIO DE USABILIDADE (SUS).....	68

1 INTRODUÇÃO

A pandemia da COVID-19 atuou como um **catalisador decisivo** para a transformação tecnológica no ambiente de trabalho, impulsionando a adoção do trabalho remoto em larga escala e consolidando novas formas de colaboração e modelos de trabalho não presenciais (LOPES, 2023). Esse cenário evidenciou a **imperatividade** das organizações em se adaptarem rapidamente a um contexto de instabilidade, tornando o **trabalho dinâmico e a modernização dos processos de trabalho** um valor fundamental para a continuidade dos negócios.

Como resultado dessa **revolução digital e da globalização dos mercados de trabalho**, as estruturas tradicionais de emprego foram alteradas. Houve um aumento na informalização do trabalho e o surgimento de novos modelos de organização, como terceirização, subcontratação e o próprio trabalho remoto. Embora essa adaptação tenha permitido maior independência em relação ao tempo e ao espaço do indivíduo, ela também exigiu, em muitos casos, uma reorganização do trabalho e a criação de novos espaços para os profissionais (Diehl, 2024). No entanto, em paralelo, surgiram novos paradigmas para o trabalho que visam não apenas flexibilidade de horário e ambiente, mas também a colaboração e a inovação no âmbito da produtividade.

Nesse cenário, os *coworkings* surgiram como uma solução estratégica para empresas e profissionais, oferecendo espaços compartilhados que não só reduzem custos operacionais, mas também incentivam ativamente a colaboração e a inovação (Gandini, 2015). Uma alternativa eficiente ao *home office*, tendência radical de muitas corporações modernas, os *coworkings* oferecem muito mais do que a infraestrutura básica para operações profissionais: cada um desses espaços é uma oportunidade pela troca de ideias, inovações e *networking* entre pessoas com diferentes formações e experiências. Além disso, os *coworkings* são vistos como um excelente veículo de diminuição de custos por empresas e trabalhadores independentes, já que permitem o uso compartilhado de reuniões, espaço de internet com boa velocidade e, por vezes, até mesmo de administração de recursos (Diehl, 2024; Gandini, 2015). Tendo o índice de adesão aumentado, principalmente em *startups*, por *freelancers* e por pequenas corporações, esses estabelecimentos têm adotado um modelo de colaboração e um modelo mais adaptado à evolução geral em muitas questões do mercado de trabalho.

Assim, os *coworkings* não apenas conseguiram atender à demanda crescente por flexibilidade e colaboração como também se beneficiaram de uma oportunidade de mercado promovida por empresas com interesse em diminuir os custos de operação sem abrir mão do

conforto e da comodidade de um ambiente de trabalho preparado. A necessidade das organizações de se adaptarem de maneira mais célere em um cenário de instabilidade fez com que o modelo de *coworking* se tornasse real e viável para muitos profissionais e empreendedores – sobretudo aqueles que não tinham interesse em um contrato duradouro ou que desejassem resultados escalonáveis. (Boaventura, 2022).

A popularização dos espaços de *coworking*, contudo, trouxe consigo desafios. Embora esses ambientes sejam amplamente reconhecidos por suas vantagens — como a flexibilidade, o custo reduzido e a possibilidade de *networking* — ainda se observa, em muitos casos, a ausência de sistemas de gestão adequados, principalmente quando se trata de espaços voltados para iniciativas públicas ou de baixo orçamento. Essa lacuna pode comprometer a organização, gerar conflitos de uso e até subutilização dos recursos disponíveis.

No contexto descrito, a Marandu, Agência UEMA de Inovação e Empreendedorismo, desenvolveu seu próprio espaço de *coworking*: a Marandu Hub. Esse espaço visa proporcionar um ambiente de trabalho dinâmico, moderno e colaborativo, atendendo à crescente demanda por lugares que integrem inovação e redes de contato. Ao promover o uso compartilhado de infraestrutura e recursos, a Marandu Hub se apresenta como uma alternativa acessível para profissionais e empreendedores, oferecendo uma plataforma que favorece a troca de experiências e a construção de soluções inovadoras. Além disso, a Marandu Hub tem como objetivo criar um ecossistema de colaboração que favoreça o desenvolvimento de novos projetos e a consolidação de novas ideias.

Com o objetivo de contribuir para a eficiência na administração do espaço, este trabalho propõe o desenvolvimento de um sistema de reservas para a Marandu Hub, desenvolvido com a ferramenta FlutterFlow, uma plataforma *low-code* que proporciona rapidez e baixo custo na concepção de soluções tecnológicas. A partir desse sistema, será possível otimizar o processo de reserva de salas e outros recursos, por meio de uma interface que fornece facilidade e interatividade para os usuários e controle para os administradores.

1.1 Objetivos

1.1.1 Objetivo Geral

Desenvolver um sistema de gerenciamento de *coworking* (Marandu Hub) utilizando uma ferramenta *low-code*, demonstrando a viabilidade e os benefícios dessa abordagem no desenvolvimento ágil de soluções tecnológicas.

1.1.2 Objetivos Específicos

- Investigar os principais desafios enfrentados na gestão de espaços de *coworking*, com ênfase no agendamento e uso eficiente de salas.
- Avaliar as potencialidades e limitações das plataformas *low-code*, com destaque para o FlutterFlow;
- Prototipar e implementar um sistema funcional de gerenciamento de salas para a Marandu Hub utilizando FlutterFlow;
- Validar a solução desenvolvida por meio de testes com usuários e coleta de *feedback* sobre usabilidade, eficiência e desempenho.

1.2 Justificativa

Administrar espaços de *coworking* envolve obstáculos consideráveis, como a tarefa de coordenar agendamentos de salas, aperfeiçoar o aproveitamento de recursos e proporcionar uma experiência de excelência para os membros. Tais questões podem ser particularmente intrínsecas para *coworkings* em expansão ou para aqueles que ainda operam com métodos de gerenciamento manuais (DIEHL, 2024). A relevância do *coworking* no cenário atual é evidenciada pelo seu crescimento e pelo seu papel na transformação dos modelos de trabalho (UNIOFFICE, 2023).

Em contrapartida, o emprego de instrumentos *low-code* tem ganhado relevância como uma opção dinâmica e eficaz para a criação de soluções tecnológicas. O mercado de *low-code* tem crescido significativamente, transformando os processos de desenvolvimento de *software* (GRAND VIEW RESEARCH, 2024). Estes instrumentos viabilizam a elaboração de plataformas funcionais com agilidade e economia, mesmo para equipes com poucos recursos técnicos, popularizando o acesso à tecnologia e incentivando a inovação (UPSITES, 2023).

Diante dos desafios de gestão de *coworking*, torna-se relevante investigar o potencial das tecnologias *low-code* na elaboração de plataformas de gestão, visto que elas se mostram capazes de simplificar procedimentos complexos, como o agendamento de salas, a organização de recursos e a produção de análises para aprimoramento constante (UPSITES, 2023). Por essa razão, o FlutterFlow, uma ferramenta *low-code* baseada no *framework* do Google, que permite a criação de aplicativos nativos com componentes visuais e customização, será empregado na construção de uma solução prática, demonstrando essa viabilidade (FLUTTERFLOW, 2025).

Ao implantar a plataforma, almeja-se não apenas solucionar problemas específicos enfrentados por *coworkings*, otimizando o desempenho e a rentabilidade desses espaços (DIEHL, 2024), mas também destacar as vantagens do *low-code* como uma estratégia atual e eficiente no desenvolvimento de *software* (UPSITES, 2023). Esta solução poderá servir como um caso concreto de como tecnologias inovadoras podem ser aplicadas para aprimorar a gestão de espaços compartilhados, fomentando a eficiência e a inovação no setor. Com o desenvolvimento desse sistema, o objetivo não é unicamente melhorar a experiência dos usuários — tornando o processo de agendamento mais simples e eficiente — como também otimizar o uso dos recursos disponíveis, evitar conflitos de agendamento, maximizar a utilização do espaço e tornar eficiente todo o gerenciamento da Marandu Hub.

Por fim, este projeto posiciona as ferramentas *low-code* como alternativas estratégicas e acessíveis para a criação de soluções tecnológicas, especialmente em contextos que demandam velocidade e adaptabilidade.

1.3 Estrutura do trabalho

Este trabalho está organizado da seguinte maneira:

- **Capítulo 1** – Apresenta a introdução ao tema, seus objetivos, justificativas e a contextualização do problema;
- **Capítulo 2** – Fundamentação teórica, abordando os conceitos centrais de *coworking*, plataformas *low-code* e a ferramenta FlutterFlow;
- **Capítulo 3** – Metodologia utilizada no desenvolvimento do sistema e na condução da pesquisa;
- **Capítulo 4** – Descreve detalhadamente o processo de desenvolvimento da aplicação, estrutura de dados, lógica de negócio e funcionalidades implementadas;
- **Capítulo 5** – Apresenta e discute os resultados obtidos, com base nos testes e validação prática, incluindo a análise crítica e sugestões para trabalhos futuros;
- **Capítulo 6** – Traz as considerações finais e a conclusão do estudo.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Coworking

2.1.1 O que é coworking?

O *coworking* representa um formato de espaço laboral colaborativo, acolhendo diversos profissionais, desde autônomos e donos de *startups* até equipes remotas de grandes empresas. Tais locais proporcionam uma infraestrutura versátil e abrangente – incluindo mesas de trabalho, salas para reuniões, atendimento, internet, apoio técnico e áreas de relaxamento –, pensada para otimizar a produtividade, o bem-estar e as chances de interação entre os membros.

Segundo Boaventura (2022), os *coworkings* são áreas projetadas para estimular a interação, o compartilhamento de saberes e a criação de contatos profissionais. Além de dividirem recursos materiais, os coworkers prezam pelo clima de grupo, pela independência no trabalho e pelo *networking*. A ideia vai além de "alugar um local" e se aproxima de um sistema que combina união, criatividade e adaptabilidade.

Após a pandemia, os *coworkings* firmaram-se como uma opção interessante à inflexibilidade dos escritórios comuns. Seu aumento na procura reflete o desejo por mais harmonia entre vida pessoal e profissional, diminuição dos gastos com estrutura e independência de localização – pontos-chave no âmbito do trabalho adaptável e à distância.

2.1.2 Origem e evolução do coworking

Brad Neuberg criou, em 2005, uma alternativa que visava unir a autonomia do *freelancer* e a estrutura de um escritório tradicional: o *coworking*. Ele buscava corresponder ao desejo crescente de profissionais que queriam horários flexíveis, mas sentiam falta de um convívio social e de um espaço que impulsiona o trabalho em equipe.

O primeiro espaço de *coworking*, denominado ***Coworking Group***, foi instalado por Neuberg em um centro comunitário chamado ***Spiral Muse***. O local funcionava apenas três dias por semana, aproveitando períodos ociosos do espaço. Ainda que modesta, essa iniciativa representou um marco inicial para o desenvolvimento de uma nova forma de organização do trabalho.

Pouco tempo depois, Neuberg, em parceria com outros profissionais, fundou o *Hat Factory*, um dos primeiros ambientes dedicados exclusivamente ao *coworking*. Contudo, apenas com a criação do *Citizen Space*, em colaboração com *Tara Hunt* e *Chris Messina*, que o conceito ganhou maior visibilidade e iniciou sua expansão global (Medina; Krawulski, 2016).

A proposta inicial de Neuberg era criar uma alternativa aos escritórios corporativos tradicionais, como os da Google, promovendo ambientes mais acessíveis, colaborativos e humanizados. Esses espaços logo passaram a atrair *freelancers*, *startups* e pequenos empreendimentos, interessados em diminuir seus custos operacionais e ampliar suas possibilidades de *networking* e inovação (Gandini, 2015).

Logo, o modelo de *coworking* evoluiu de maneira significativa, impulsionado por transformações tecnológicas, mudanças nas dinâmicas de trabalho e, mais recentemente, pelos efeitos da pandemia da COVID-19. Para a Nex Coworking (2023), os *coworkings* representam uma revolução na concepção dos espaços de trabalho, com foco na criação de comunidades profissionais autônomas, criativas e interconectadas. Atualmente, o *coworking* é reconhecido como uma das principais expressões do trabalho adaptável e colaborativo, combinando economia de recursos, bem-estar profissional e estímulo à inovação (Diehl, 2024).

2.1.3 Coworking no Brasil

No Brasil, o conceito de *coworking* começou a ganhar destaque por volta de 2007, marcando o início de um movimento que, desde então, não para de crescer. Essa modalidade de trabalho rapidamente se consolidou como uma alternativa atraente para profissionais e pequenas empresas que buscam flexibilidade, infraestrutura de qualidade com custos otimizados, e a valiosa oportunidade de *networking*. É uma solução ideal para empreendedores, *freelancers*, profissionais autônomos e equipes em regime de trabalho remoto, oferecendo um ambiente dinâmico que estimula a colaboração (Medina; Krawulski, 2016).

A robustez desse crescimento é evidenciada pelos números: o Censo Coworking 2024, realizado pela Woba, aponta um salto impressionante de 238 espaços em 2015 para mais de 2.986 em 2024. Esse dado demonstra não apenas a consolidação do modelo, mas também sua capilaridade, espalhando-se por diversas regiões do país. A popularidade do *coworking* no Brasil está intrinsecamente ligada ao aumento do trabalho informal, à busca por economia de custos operacionais e à crescente valorização da flexibilidade no mercado de trabalho. Prova

disso é a posição de destaque do Brasil no cenário global de *coworking*, especialmente impulsionada pelas transformações no mercado e pelo sucesso do trabalho híbrido e à distância, conforme destaca uma matéria da Carta Capital (2023). A CNN Brasil (2022) corrobora essa tendência de forte adesão nacional: no primeiro trimestre de 2022, a procura por espaços de *coworking* no país subiu 97% em relação ao mesmo período do ano anterior, e a busca por salas de reunião cresceu 216%, demonstrando a rapidez do Brasil em adotar e consolidar esse modelo inovador.

A expansão do *coworking* no Brasil não se restringe apenas aos grandes centros do Sudeste. Regiões como o Nordeste têm mostrado um dinamismo notável, com um aumento significativo no número de espaços e na demanda por ambientes de trabalho flexíveis. Um estudo que analisou 55 *coworkings* em nove estados nordestinos confirmou a presença da modalidade, destacando que a ideia funcionou fora das capitais do Sudeste (Miki et al., 2021). Um exemplo notável é Salvador, na Bahia, que, como aponta Boaventura (2022), já contava com pelo menos 27 espaços de *coworking*.

A pesquisa de Miki et al. (2021) também aponta que o **Maranhão** se destaca como um dos poucos estados nordestinos na amostra com espaços de *coworking* não apenas na capital, São Luís, mas também em sua segunda cidade mais populosa e desenvolvida, Imperatriz. Isso sublinha a **interiorização do modelo** e a busca por soluções de trabalho flexível para além dos grandes centros urbanos do país. Em São Luís, essa tendência se manifesta de forma consistente. A presença e o crescimento de espaços de *coworking* na capital maranhense, como a **Locomotiva Hub – uma iniciativa pioneira do Governo do Estado do Maranhão que integra o Parque Tecnológico e promove o desenvolvimento de negócios de base científica** – e a **Marandu Hub**, são um indicativo de que a cidade está alinhada a essa transformação nacional e regional. A Marandu Hub se insere, portanto, nesse contexto de **popularização e expansão do *coworking***, contribuindo para o ecossistema de inovação e empreendedorismo local ao oferecer infraestrutura adequada e fomentar a comunidade profissional na região.

2.1.4 Benefícios do Coworking

O *coworking* tornou-se uma alternativa estratégica para quem busca soluções maleáveis, acessíveis e de equipe. Seu crescimento se deve aos inúmeros benefícios que oferece, alinhando-se às necessidades atuais do mercado. Os principais pontos positivos são:

- **Redução de gastos fixos:** Usar espaços divididos evita alugar prédios inteiros, pagar contas e cuidar da estrutura. É mais barato para quem trabalha sozinho, para *startups* e para empresas pequenas.
- **Adaptação para o trabalho híbrido:** Possuem espaços que são usados quando necessário, ideal para quem divide o trabalho entre casa e escritório, de forma prática e eficiente.
- **Ambiente colaborativo:** Convivência com profissionais de diferentes áreas favorece o *networking*, assim, promovendo um ecossistema propício à inovação e ao desenvolvimento profissional.
- **Estrutura e serviços compartilhados:** Esses espaços, geralmente, oferecem recepção, salas de reunião, escritórios privativos, áreas de convivência e descanso, além de serviços como internet de boa velocidade, limpeza, suporte técnico e impressão. Isso permite que os usuários foquem exclusivamente em suas atividades.
- **Contrato flexível:** O modelo de locação é adaptável às necessidades do usuário, podendo ser contratado por hora, dia, semana ou mês. Essa flexibilidade permite escalar ou reduzir operações conforme a demanda, sem os encargos e rigidez dos contratos tradicionais.
- **Serviços adicionais:** Muitos oferecem endereço fiscal, gestão de correspondência, o que é muito vantajoso para micros e pequenas empresas.

Além de ser prático, o coworking ajuda a criar uma cultura nova nas empresas, focada em equipe, propósito e inovação. A Unioffice (2023) diz que esses lugares mudam o jeito tradicional de trabalhar, ajudando a construir estruturas mais humanas e que funcionam melhor.

Segundo o Club Coworking (2022), a combinação de economia, infraestrutura adequada e um ambiente estimulante impulsiona o rápido crescimento dessa modalidade no Brasil. Depois da pandemia e com o trabalho remoto ficando comum, empresas e profissionais querem lugares mais dinâmicos para trabalhar, escolhendo cada vez mais os espaços compartilhados.

2.2 Plataformas de Desenvolvimento Low-Code

A programação em low-code busca dar mais velocidade à produção de ferramentas digitais. Ela usa áreas de trabalho intuitivas, partes prontas e dispensa a necessidade de

escrever muito código. É um jeito novo de fazer softwares, diferente do método tradicional, possibilitando a criação rápida de apps, sites e processos automáticos, diminuindo prazos e gastos.

2.2.1 Definição, Características e Classificações

O desenvolvimento low-code representa uma abordagem inovadora na criação de ferramentas digitais, que visa acelerar a produção de software por meio de ambientes de trabalho intuitivos, componentes pré-construídos e uma redução significativa na necessidade de escrita manual de código. Diferente do método tradicional, essa abordagem possibilita a criação rápida de aplicativos, sites e processos automatizados, impactando diretamente na diminuição de prazos e custos (UPSITES, 2023).

Essas plataformas se destacam por oferecerem Ambientes de Desenvolvimento Integrados (IDEs) visuais, onde os usuários podem montar aplicações utilizando elementos gráficos, conectores de dados e lógicas de negócio predefinidas. Mesmo sendo necessário certo conhecimento técnico, essas ferramentas minimizam a necessidade de um profundo domínio de linguagens de programação, tornando o desenvolvimento acessível a profissionais com variados níveis de experiência (MORAES, 2022).

Devido à crescente demanda por modernização digital e à necessidade de agilizar a criação de softwares, as plataformas de baixo código têm ganhado espaço em organizações de todos os portes. Elas viabilizam customizações e simplificam conexões com APIs e bases de dados externas, além de contribuírem para a diminuição da sobrecarga das equipes de TI e promover maior envolvimento de outras áreas de negócio no ciclo de desenvolvimento (UPSITES, 2023).

De acordo com a Grand View Research (2024), o mercado global de plataformas de desenvolvimento *low-code* foi estimado em 24,8 bilhões de dólares em 2023 e deve alcançar aproximadamente 101,6 bilhões de dólares até 2030, com uma taxa de crescimento anual composta (CAGR) de 22,5% entre 2024 e 2030. Esse crescimento reflete a ampla adoção dessas plataformas como alternativa às abordagens tradicionais de desenvolvimento, permitindo maior agilidade, redução de custos e acessibilidade para profissionais com pouca ou nenhuma experiência em programação (**Figura 1**).

Figura 1 - Projeção de Crescimento do Mercado de Low-Code



Fonte: Grand View Research (2024)

As plataformas *low-code* apresentam características comuns que as tornam eficazes para desenvolvimento ágil:

- Ambiente visual baseado em componentes pré-configurados;
- Criação e integração de banco de dados;
- Automação de lógica de negócios com *builders* visuais;
- Capacidade de personalização por meio de *scripts* ou código;
- Integração nativa com autenticação, serviços em nuvem e *APIs* externas;
- Deploy automatizado em *web*, *mobile* e *progressive web apps (PWA)*.

Essas plataformas variam conforme o grau de complexidade e escalabilidade que oferecem. Algumas são ideais para prototipação rápida e validação de ideias, enquanto outras suportam sistemas empresariais complexos, com gestão de usuários, *workflows* e segurança robusta.

2.2.2 Vantagens do Desenvolvimento Low-Code

A adoção de plataformas *low-code* oferece uma série de benefícios estratégicos para o desenvolvimento de *software*, impactando positivamente a velocidade, o custo e a acessibilidade tecnológica. Entre as principais vantagens, destacam-se:

- **Agilidade e Velocidade de Entrega:** A utilização de componentes visuais e pré-construídos acelera significativamente o ciclo de desenvolvimento, permitindo que aplicações sejam prototipadas e lançadas em tempo recorde, o que é crucial em mercados dinâmicos (MORAES, 2022; UPSITES, 2023).
- **Redução de Custos:** Ao diminuir a necessidade de codificação manual e o tempo de desenvolvimento, o *low-code* contribui para a otimização de recursos e a redução de despesas com equipes de desenvolvimento extensas ou projetos de longa duração (MORAES, 2022; UPSITES, 2023).
- **Democratização do Desenvolvimento:** Ferramentas *low-code* tornam a criação de aplicações acessível a um público mais amplo, incluindo *citizen developers*¹, o que acelera a inovação e o atendimento a demandas internas sem sobrecarregar as equipes de TI (UPSITES, 2023).
- **Flexibilidade e Adaptabilidade:** A natureza modular e visual dessas plataformas facilita a modificação e a adaptação de aplicações às mudanças nos requisitos de negócio, promovendo um ciclo contínuo de melhoria e evolução (UPSITES, 2023).
- **Integração Simplificada:** Muitas plataformas *low-code* oferecem conectores e *APIs* pré-construídas, simplificando a integração com sistemas legados, bancos de dados e serviços em nuvem, o que é um desafio comum no desenvolvimento tradicional (MORAES, 2022).

Essas vantagens posicionam o *low-code* como uma alternativa promissora para organizações que buscam inovar e digitalizar seus processos de forma eficiente e escalável.

2.2.3 Limitações e Desafios do Low-Code

Apesar das inúmeras vantagens, o desenvolvimento *low-code* não está isento de desafios e limitações que devem ser cuidadosamente considerados, especialmente em projetos de engenharia de *software* que exigem controle e personalização rigorosos. Compreender esses pontos é fundamental para uma escolha tecnológica consciente e para a mitigação de riscos.

- **Vendor Lock-in (Dependência do Fornecedor):** Uma das principais preocupações é a dependência da plataforma escolhida. Ao construir uma aplicação em um ecossistema *low-code*, a migração para outra plataforma ou para um desenvolvimento *full-code*

¹ *Citizen developers* são usuários de negócio sem profundo conhecimento técnico

pode ser complexa e dispendiosa, pois o código gerado pode ser proprietário ou não facilmente portátil (REFINE DEV, 2024). Essa dependência decorre da natureza proprietária de muitas dessas plataformas, que geram código específico e usam abstrações que dificultam a exportação limpa e a reutilização em outros ambientes. Mitigar esse risco exige uma avaliação rigorosa do suporte à exportação de código e da abertura da plataforma desde a fase de seleção.

- **Personalização e Flexibilidade Limitadas:** Embora o *low-code* ofereça grande agilidade para casos de uso comuns, ele pode apresentar restrições para requisitos muito específicos, complexos ou altamente personalizados. A agilidade do *low-code* é alcançada por meio de componentes padronizados; contudo, quando requisitos de negócio são altamente específicos ou a interface de usuário demanda um design inovador e fora do padrão, as ferramentas visuais podem se tornar um gargalo. Nesses cenários, a necessidade de recorrer a código manual (*escape hatches* ou extensões) pode anular parte dos ganhos de produtividade e complexificar a manutenção (UPSITES, 2023).
- **Escalabilidade e Performance em Cenários Complexos:** Para aplicações de grande escala, com alta demanda de usuários ou processamento intensivo, pode haver preocupações quanto à performance e à otimização. Embora muitas plataformas sejam escaláveis, o controle granular sobre o código-fonte e a infraestrutura subjacente é reduzido. Isso pode impactar aplicações com volume de dados massivo ou lógicas de processamento muito intensivas, onde o controle granular é crucial para ajustes finos (MORAES, 2022).
- **Segurança e Governança:** Estes representam desafios críticos, especialmente em projetos de maior escala ou com dados sensíveis. A abstração do código e das camadas de infraestrutura, embora facilite o desenvolvimento, pode reduzir a visibilidade sobre vulnerabilidades potenciais e dificultar auditorias de segurança detalhadas. A dependência dos recursos de segurança nativos da plataforma exige uma confiança robusta no fornecedor. Além disso, a possibilidade de *citizen developers* criarem aplicações sem supervisão adequada pode gerar riscos de *shadow IT*², desvio de dados ou falha em aderir a políticas de segurança e conformidade internas. Para mitigar esses riscos, é fundamental que as organizações implementem políticas rigorosas de governança, com revisões de segurança contínuas, controle de acesso baseado em

² *Shadow IT* é o desenvolvimento de sistemas sem controle da TI central

funções e capacitação adequada dos desenvolvedores sobre as melhores práticas de segurança na plataforma (MORAES, 2022).

- **Debugging e Resolução de Problemas:** A depuração de erros em ambientes *low-code* pode ser mais desafiadora, pois o desenvolvedor tem menos visibilidade sobre o código gerado. Isso exige uma boa compreensão da lógica da plataforma e de suas ferramentas de diagnóstico.

Apesar dessas limitações, o *low-code* continua sendo uma ferramenta poderosa quando aplicado em cenários adequados, onde a agilidade e o custo-benefício superam a necessidade de controle granular total. A escolha da plataforma deve ser alinhada com os requisitos e o contexto do projeto.

2.2.4 Principais Plataformas de Low-Code

A seguir, destacam-se algumas das plataformas *low-code* mais relevantes atualmente:

- **OutSystems:** Voltada para empresas de grande porte, com foco em aplicações corporativas robustas, integração com sistemas legados e governança de TI.
- **Mendix (Siemens):** Permite o desenvolvimento de aplicações *web* e *mobile* com foco em arquitetura empresarial, forte integração com *DevOps* e automação de processos.
- **Microsoft Power Apps:** Integrada ao ecossistema Microsoft, permite o desenvolvimento de aplicativos com conexões diretas ao SharePoint, Azure, Excel, Dynamics 365, entre outros.
- **FlutterFlow:** Plataforma baseada no *Flutter (framework* do Google), destinada à criação de aplicativos nativos Android, iOS e *Web*. Permite o desenvolvimento por meio de componentes visuais, criação de banco de dados com Firestore, e customização com *Custom Functions* e *Custom Actions*. É ideal para *startups*, projetos acadêmicos e iniciativas que exigem uma interface moderna e entregas rápidas com escalabilidade futura.

Uma análise comparativa detalhada dessas plataformas pode ser encontrada na **Tabela**

1.

Tabela 1 - Análise Comparativa das Principais Plataformas Low-Code

Característica / Plataforma	OutSystems	Mendix	Microsoft Power Apps	FlutterFlow
Foco Principal	Grandes empresas, apps corporativos, integração com legados (OutSystems, 2025)	<i>Apps web / mobile</i> , arquitetura empresarial, <i>DevOps</i> (Mendix, 2025)	Ecosistema Microsoft, SharePoint, Azure, Excel (Microsoft, 2025)	<i>Apps</i> nativos (Android / iOS / <i>Web</i>), componentes visuais, integrações Firebase (FLUTTERFLOW, 2025)
Público-alvo / Cenário Ideal	Empresas de grande porte com governança de TI	Desenvolvimento empresarial, automação	Usuários Microsoft, automação de fluxo de trabalho	<i>Startups</i> , projetos acadêmicos, interfaces modernas, entregas rápidas
Ecosistema / Base	Proprietário	Proprietário	Microsoft	<i>Flutter</i> (Google)
Nível de Customização / Flexibilidade	Alto, mas pode exigir expertise na plataforma	Alto para <i>low-code</i>	Médio, integrado ao ecossistema MS	Alto para <i>low-code</i> (<i>Custom Actions / Functions</i>)
Modelo de Licenciamento	Baseado em usuários aplicativos, geralmente alto custo	Baseado em usuários capacidade, alto custo	Assinaturas por usuário / app, variadas	Gratuito (limitado), planos pagos (Pro, Team)
Curva de Aprendizado	Média/Alta	Média	Baixa para usuários MS Office	Média (com familiaridade <i>Flutter / Firebase</i>)

Fonte: Autor (2025)

2.3 FlutterFlow no Contexto de Desenvolvimento Ágil com Low-Code

2.3.1 Visão Geral e Recursos Chave

O FlutterFlow é uma plataforma de desenvolvimento visual (*low-code*) que permite criar aplicações *mobile*, *web* e *Progressive Web Apps (PWA)* de forma eficiente e acessível. Desenvolvida com base no *framework Flutter* — mantido pelo Google — a ferramenta visa democratizar o processo de criação de sistemas, combinando a praticidade de um editor visual com a robustez da programação tradicional (FLUTTERFLOW, 2025).

Sua proposta é atender desde iniciantes, que não dominam linguagens de programação, até desenvolvedores experientes que desejam acelerar a prototipação e entrega de produtos digitais. O FlutterFlow oferece integração com banco de dados como o Firestore e mecanismos para lógica personalizada.

A plataforma apresenta um conjunto de recursos que se adequa tanto ao desenvolvimento inicial de projetos quanto à construção de sistemas corporativos. Entre os principais destaques, incluem-se:

- **Editor visual de interfaces:** As telas são criadas por meio de componentes arrastáveis com personalização de estilos, responsividade e interações. O *layout* adaptativo permite que o mesmo projeto seja acessado com fluidez em *desktop* e *mobile* (FLUTTERFLOW, 2025).
- **Integração com Firebase:** O FlutterFlow se conecta ao Firebase para utilizar autenticação (por email ou redes sociais), banco de dados NoSQL (Cloud Firestore), armazenamento de arquivos, funções em nuvem e hospedagem (FIREBASE, 2025). No projeto, essa estrutura serve como base para persistência de dados, como usuários, salas, reservas e empresas incubadas.
- **Lógica com *Actions*, *Workflows* e *Custom Functions*:** É possível criar fluxos de navegação, interações condicionais, filtros avançados, cálculos dinâmicos e validações, com ou sem código. Na Marandu Hub, por exemplo, funções personalizadas em Dart, linguagem de programação utilizada no FlutterFlow para desenvolver lógicas de negócio e funcionalidades diversas, são utilizadas para validar horários de reserva, aplicar regras administrativas e estruturar relatórios.
- **Integração com APIs externas:** A ferramenta permite chamadas HTTP REST com métodos POST, GET, entre outros. O sistema da Marandu Hub realiza duas integrações principais:
 - Requisição ao *endpoint* do Firebase para redefinir senhas: [https://identitytoolkit.googleapis.com/v1/accounts:resetPassword?key=\[API_KEY\]](https://identitytoolkit.googleapis.com/v1/accounts:resetPassword?key=[API_KEY]).
 - Integração com uma *API* customizada hospedada na **Vercel**, uma plataforma de nuvem de alto desempenho que permite a implantação e escalabilidade de aplicações web. Essa integração gera planilhas de empresas incubadas através do *endpoint*: <https://marandu-hub-relatorioxlsx.vercel.app/api/generate-xlsx>.

- **Exportação e manutenção:** Os projetos podem ser exportados como código *Flutter* completo, facilitando a manutenção futura ou transição para desenvolvimento *full-code* (White Label Fox, 2024).

2.3.2 Benefícios Observados

Durante o desenvolvimento e uso do sistema, destacaram-se os seguintes benefícios:

- **Agilidade de prototipação e ajustes visuais rápidos**, favorecendo entregas contínuas;
- **Baixo custo de infraestrutura**, com uso exclusivo do Firebase;
- **Alta personalização** com funções personalizadas em Dart (validações, conversão de datas, tratamento de dados, entre outras);
- **Possibilidade de escalar ou migrar** para *Flutter* nativo a partir do código exportável;
- **Facilidade de manutenção**, ideal para uma equipe técnica reduzida.
- **Curva de aprendizado curta**, que possibilitou o desenvolvimento autônomo e completo da solução por um único desenvolvedor.

Essas vantagens reforçam o FlutterFlow como uma escolha estratégica para projetos acadêmicos e institucionais que exigem funcionalidade robusta, baixo custo e facilidade de uso, como é o caso da Marandu Hub.

2.3.3 Contextos de Uso e Popularidade do FlutterFlow

O FlutterFlow tem se destacado bastante nos meios tecnológico e de ensino, sendo visto como uma das plataformas de *low-code* mais fáceis de usar e adaptáveis para criar aplicativos para a internet e celulares. Seu principal diferencial está na interface visual intuitiva, que facilita a construção de telas e fluxos de navegação utilizando a estrutura do Flutter. Isso permite a criação de interfaces responsivas, adaptadas a diferentes dispositivos, com a lógica de funcionamento integrada e a possibilidade de exportação do código em Dart para personalizações futuras. Essa abordagem, aliada à integração simplificada com o Firebase, torna o FlutterFlow uma ferramenta poderosa tanto para prototipagens rápidas quanto para o desenvolvimento de sistemas completos.

Em muitos setores, a plataforma tem sido escolhida por empresas novas, escolas e organizações sem fins lucrativos, que querem desenvolver seus projetos de forma ágil, mas sem perder a chance de personalizar. Usar o FlutterFlow tira boa parte da dificuldade que

existe em projetos tradicionais, permitindo que desenvolvedores sozinhos e times pequenos entreguem produtos que funcionam bem, de forma rápida e com boa qualidade.

No mundo acadêmico, é notável o aumento do uso do FlutterFlow em trabalhos de conclusão de curso (TCC), aplicativos para redes sociais e sistemas para ajudar na administração. Um exemplo disso é o projeto **Açaitour**, feito por estudantes do Instituto Federal do Maranhão (IFMA), que criou um aplicativo para divulgar lugares de diversão e turismo na cidade de Açailândia-MA. O FlutterFlow foi muito importante para construir o aplicativo, por ser fácil de usar e conseguir criar aplicativos para várias plataformas de forma rápida (RIBEIRO et al., 2023).

Outro caso de destaque é o sistema **Adopet**, do Instituto Federal Catarinense (IFC), que tem como objetivo otimizar o processo de adoção de animais abandonados. A equipe optou por utilizar o FlutterFlow devido à sua curva de aprendizado reduzida, interface visual intuitiva e fácil integração com recursos como formulários e banco de dados. O sistema visa facilitar a conexão entre ONGs e possíveis adotantes, destacando o papel social que o FlutterFlow também pode assumir (SILVA; FERREIRA, 2023).

Além desses casos, a comunidade ativa ao redor da plataforma contribui significativamente para seu crescimento. Repositórios abertos com *templates*, tutoriais detalhados e integração com plataformas como GitHub e Vercel ampliam as possibilidades de aprendizado e aplicação prática. Isso tem atraído não apenas estudantes e iniciantes, mas também desenvolvedores seniores interessados em acelerar entregas e reduzir o tempo de desenvolvimento.

No caso da Marandu Hub, o uso do FlutterFlow foi estratégico. Permitiu que o sistema fosse desenvolvido individualmente, com alto grau de personalização, integração com Firebase e entrega escalável em ambiente real. A autonomia proporcionada pela ferramenta foi essencial para o sucesso do projeto, demonstrando como tecnologias *low-code* podem democratizar o desenvolvimento e reduzir as barreiras técnicas em projetos institucionais e públicos.

Dessa forma, o FlutterFlow se consolida como uma solução robusta e moderna, cada vez mais presente em iniciativas acadêmicas, sociais e empreendedoras. Sua capacidade de unir velocidade, flexibilidade e poder técnico o posiciona como uma das principais escolhas entre as plataformas de desenvolvimento visual da atualidade.

3 METODOLOGIA

3.1 Tipo de Pesquisa

Esta pesquisa se classifica como **aplicada**, empregando uma perspectiva **descritiva e qualitativa**, com o intuito de solucionar um desafio concreto identificado na Marandu Hub: a administração pouco eficaz das reservas de salas. A pesquisa aplicada busca resolver problemas genuínos e, neste estudo, o objetivo foi desenvolver um recurso que auxilie na organização otimizada do uso das salas no ambiente.

A opção pela **abordagem descritiva** se justifica pela necessidade de ilustrar o processo de desenvolvimento da solução e seu potencial impacto na gestão do ambiente. Tal escolha possibilita detalhar minuciosamente as fases do desenvolvimento, como a seleção de tecnologias e as decisões tomadas durante o percurso, sempre visando aprimorar a operacionalidade do espaço da Marandu Hub.

Por fim, a **abordagem qualitativa** é crucial para compreender as demandas e vivências dos frequentadores do espaço. Na Marandu Hub, o uso das salas é diversificado e abrange distintos tipos de usuários, **incluindo profissionais autônomos, empreendedores e equipes de empresas incubadas**, portanto, a pesquisa qualitativa auxiliou na escuta e compreensão de como o sistema pode ser verdadeiramente útil para quem utiliza o espaço. Isso assegura que a solução não apenas resolva uma questão técnica, mas que também seja prática e eficiente para as pessoas cotidianamente.

3.2 Procedimentos Metodológicos

A metodologia adotada foi **flexível e empírica**, alinhada à natureza de um projeto de desenvolvimento individual focado em uma solução prática para um problema real. Em vez de seguir uma metodologia formal como *Scrum*, *Design Thinking* ou cascata, optou-se por um **ciclo de desenvolvimento iterativo e incremental informal**, focado na execução direta da aplicação e em **ajustes contínuos baseados no feedback direto do stakeholder e do uso real do espaço Marandu Hub**. A não adoção de um *framework* formal, como o *scrum*, deveu-se às limitações de alinhamento de cronogramas, considerando a agenda do pesquisador e dos *stakeholders*. No entanto, esse processo foi guiado por entregas de funcionalidades com prazos definidos pelo professor orientador, seguidas de reuniões periódicas com a participação do técnico administrativo para demonstrações e alinhamentos. A comunicação contínua via

e-mail e mensagens também foi fundamental para sanar dúvidas pontuais. Essa flexibilidade se mostrou vantajosa para **garantir agilidade na entrega de funcionalidades e rápida adaptação às demandas do ambiente**, especialmente considerando o desenvolvimento por um único pesquisador. No entanto, é importante reconhecer que essa abordagem, embora eficaz para a situação específica da Marandu Hub, **limita a replicabilidade e a generalização dos resultados** para outros contextos, uma vez que não seguiu padrões formais de documentação e controle de mudanças comumente empregados em metodologias estruturadas.

O projeto foi desenvolvido com o apoio fundamental do orientador acadêmico e de Gabriel Bezerra, técnico administrativo da Marandu Hub, que atuou como *stakeholder*³ diretamente envolvido no processo de agendamento dos espaços. Realizaram-se reuniões ocasionais com esses interlocutores para tirar dúvidas, validar funcionalidades e **compreender melhor as rotinas operacionais e os desafios do agendamentos dos espaços de coworking**. Esse contato direto foi essencial para garantir que as decisões tomadas ao longo do desenvolvimento estivessem alinhadas com as necessidades reais do dia a dia da Marandu Hub.

Vale destacar que, antes de sua plena implantação, o sistema desenvolvido passou por uma **fase de testes internos e validação** com a equipe da Marandu Hub para garantir sua aderência aos processos e necessidades. **Em maio de 2025**, o sistema entrou em **produção real**, substituindo o antigo processo de gerenciamento de reservas que se baseava no recebimento de solicitações por e-mail e posterior registro manual na agenda do e-mail. Essa prática gerava desafios como a lentidão no processamento das solicitações e a dependência exclusiva de uma única pessoa, resultando em eventuais esquecimentos e falhas na organização dos agendamentos. A aplicação foi criada para digitalizar e organizar esse processo, e já está sendo utilizada de forma efetiva por administradores e usuários do espaço, **acumulando 93 usuários ativos e 227 reservas realizadas até o momento da redação deste trabalho**. O caráter empírico do projeto fica evidente na constante interação entre observação, implementação e ajustes, já que cada nova funcionalidade foi pensada com base em demandas práticas, reforçando a proposta de uma solução que realmente resolva o problema identificado.

³ *Stakeholder* é o termo utilizado para designar pessoas ou grupos envolvidos ou afetados por decisões institucionais. No presente contexto, refere-se ao técnico administrativo responsável pelo processo de agendamento dos espaços.

3.3 Ferramentas e Tecnologias Utilizadas

Para o desenvolvimento da aplicação, foi utilizado um conjunto de ferramentas integradas, com foco em acessibilidade, escalabilidade e facilidade de manutenção. A seguir, são descritas as principais tecnologias utilizadas:

- **FlutterFlow:** Plataforma de desenvolvimento *low-code* utilizada para a construção da interface visual (*frontend*), definição das rotas, criação de formulários e aplicação da lógica condicional. Também foi utilizada para desenvolver *Custom Actions*, *Custom Functions* e um *Widget* personalizado.
- **Firebase:** Utilizado como *backend* da aplicação, fornecendo autenticação de usuários, banco de dados NoSQL (Cloud Firestore), armazenamento de arquivos e hospedagem. Toda a persistência de dados — como usuários, reservas, salas e empresas incubadas — está estruturada neste serviço.
- **Node.js + API personalizada:** A *API* foi desenvolvida para gerar relatórios em formato Excel com base nos dados armazenados no Firebase. Ela está hospedada na Vercel e se comunica via chamadas HTTP com o sistema principal.
- **Dart:** Linguagem utilizada no FlutterFlow para a criação de funções personalizadas (*Custom Functions*), ações específicas (*Custom Actions*) e validações de dados.
- **EmailJS:** Serviço externo utilizado para o envio automático de e-mails transacionais (solicitação, confirmação e cancelamento de reservas). Integrado ao FlutterFlow por meio de *Custom Actions* que disparam as mensagens conforme eventos definidos na lógica da aplicação. (EMAILJS, 2025)

3.4 Etapas do Projeto

As etapas de desenvolvimento foram realizadas conforme a lógica prática de prototipação e refinamento iterativo, ainda que não seguissem um modelo metodológico formal. As atividades foram executadas conforme descrito abaixo:

3.4.1 Levantamento de Requisitos

O levantamento de requisitos caracterizou-se como um **estudo exploratório**, com o objetivo de identificar as principais dificuldades e necessidades no processo de agendamento de espaços na Marandu Hub. Isso foi realizado por meio de uma combinação de abordagens:

- **Observações:** Acompanhamento e registro do processo de agendamento existente na Marandu Hub, buscando compreender as ineficiências e os desafios enfrentados.
- **Reuniões com o *stakeholder*:** Diálogos diretos e focados com o técnico administrativo da Marandu Hub (Gabriel Bezerra), que é o responsável direto pelo processo de agendamento. Essas reuniões permitiram compreender as rotinas operacionais, os desafios diários e as expectativas quanto à nova solução.
- **Revisão de processos existentes:** Análise do fluxo de trabalho atual, baseado em e-mails e registros manuais, e também a análise comparativa de sistemas de agendamento em outros *coworkings* locais, como a **Locomotiva Hub**, cujo sistema de reservas é desenvolvido em uma **plataforma No-Code (Glide)**, que serviu como referência para funcionalidades e design de interface, para mapear as funcionalidades essenciais que deveriam ser digitalizadas e otimizadas.

Essa abordagem exploratória e direta garantiu que os requisitos fossem alinhados às necessidades reais e imediatas do ambiente, servindo como base para a prototipação e o desenvolvimento inicial da aplicação.

3.4.2 Prototipação Visual no FlutterFlow

Elaboração das telas e estrutura da aplicação por meio do editor visual da plataforma. Essa etapa envolveu o design das abas de salas, reservas, painel administrativo, relatórios, além da criação e edição do próprio usuário.

3.4.3 Integração com Firebase

Estruturação das coleções de dados, regras de autenticação e configuração de segurança no Firestore. Foi feita a modelagem de coleções como *users*, *rooms*, *reservations* e *incubatedCompanies*.

3.4.4 Criação de Lógica de Negócio

Implementação de ações personalizadas para controle de reservas (como evitar duplicações), fluxo de confirmação/cancelamento, e controle de visualização por tipo de usuário.

3.4.5 Painel Administrativo e Relatórios

Construção das telas de controle administrativo com filtros, dashboards, botões de ação e campos de leitura de informações detalhadas da reserva.

3.5 Validação e Avaliação do Sistema

A validação do sistema foi realizada com os usuários e administradores da Marandu Hub por meio da **Escala SUS (System Usability Scale)**, um questionário padronizado e amplamente utilizado com dez itens para avaliar a usabilidade percebida de um sistema, reconhecida por sua eficácia na mensuração de usabilidade em sistemas interativos. A coleta ocorreu via formulário online, disponibilizado no **Jotform**, uma plataforma de criação e coleta de respostas de formulários digitais, assegurando participação anônima e voluntária. Os detalhes sobre as perguntas e a análise dos resultados estão presentes no Capítulo 5 – Resultados e Discussão.

É importante ressaltar que a avaliação foi conduzida com um grupo de apenas dez usuários. Embora a literatura sobre usabilidade, incluindo o próprio John Brooke (criador da SUS), sugira que amostras entre 8 e 12 participantes já possam fornecer resultados representativos para testes exploratórios e avaliações iniciais de usabilidade, e que 5 a 10 usuários são frequentemente considerados suficientes para identificar a maioria dos problemas de usabilidade, reconhece-se que o tamanho limitado da amostra impacta diretamente a capacidade de generalização dos resultados obtidos. Portanto, apesar do alto índice de usabilidade (que será detalhado no Capítulo 5), a validade externa dessas descobertas é restrita ao contexto da Marandu Hub. Para futuras pesquisas, sugere-se a ampliação do estudo com uma amostra maior e mais diversificada de usuários, o que permitiria validar e aprofundar esses achados em uma população mais ampla e em diferentes contextos de *coworking*.

4 DESENVOLVIMENTO

4.1 Aplicação no Projeto Marandu Hub

A Marandu Hub é um sistema de gerenciamento de reservas e dados institucionais do espaço de *coworking* vinculado à UEMA, voltado para salas de reunião, mesas compartilhadas e mini auditório. Sua implementação com o FlutterFlow permitiu integrar funcionalidades administrativas e operacionais de forma escalável e com boa experiência do usuário.

A estrutura geral do sistema é composta por:

- **Página inicial:** com apresentação institucional do hub;
- **Aba de salas:** listagem categorizada (por exemplo, "Incubadora - Mesa 3") com detalhes como capacidade, tipo, localização e galeria de fotos;
- **Aba de reservas:** dividida em três categorias: solicitadas, confirmadas e canceladas. O sistema impede o envio de novas solicitações para horários que já possuam reservas confirmadas, garantindo consistência e evitando conflitos.
- **Painel administrativo:** os gestores podem confirmar ou cancelar reservas, acessar dados do solicitante (nome, cpf, e-mail, instituição, setor) e visualizar o número de convidados;
- **Aba de relatório:** visualizações gráficas e filtros baseados em período, local ou status da reserva.

O *backend* utiliza Cloud Firestore com coleções organizadas da seguinte forma:

- **users:** dados de cadastro, autenticação e perfil dos usuários;
- **rooms:** salas disponíveis com tipo, descrição, capacidade e fotos;
- **reservations:** registros com data, horário, status, convidados e IDs relacionados a salas e usuários;
- **incubatedCompanies:** empresas incubadas com respectivos representantes e documentos.

A integração entre *frontend* (FlutterFlow) e *backend* (Firebase/API) é intermediada por *Custom Actions* para chamadas HTTP, verificações de conflito e relatórios automatizados, como a geração de planilhas com dados filtrados.

4.2 Arquitetura Geral da Aplicação

A Marandu Hub possui uma estrutura modular, dividida em cinco partes fundamentais, cada qual com tarefas específicas. Essa divisão segue a ideia de separar as funções, o que ajuda a aumentar a capacidade de expansão, a facilidade de manutenção e a organização no desenvolvimento e na correção de erros. Essa forma de organização não só ajuda a entender como o sistema funciona por dentro, mas também permite que ele seja ampliado e adaptado no futuro.

A forma como o sistema foi construído mostra a sua complexidade, sem prejudicar a facilidade de uso, e foi muito importante para garantir que as necessidades da Marandu Hub fossem atendidas de maneira forte, segura e eficaz. Abaixo, cada uma das partes é explicada detalhadamente, como mostrado na **Figura 2**.

Figura 2 - Arquitetura em Camadas do Sistema Marandu Hub



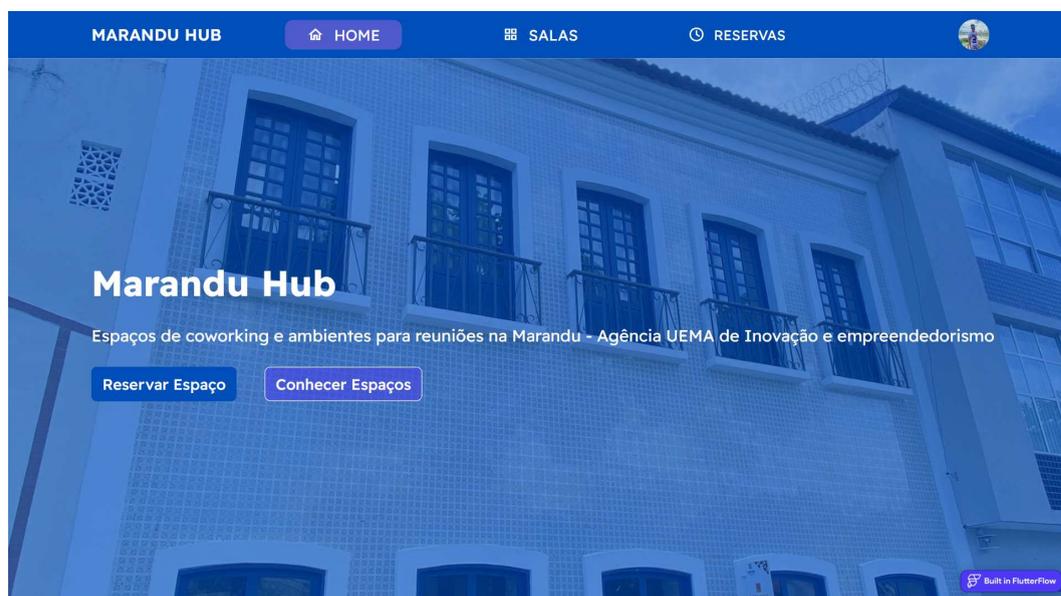
Fonte: Autor (2025)

4.2.1 Camada de Apresentação (Frontend - FlutterFlow)

Essa camada representa a interface gráfica do sistema — o ponto de contato entre o usuário e a aplicação. Foi desenvolvida utilizando a plataforma FlutterFlow, que possibilitou a criação de telas de forma visual, responsiva e com boa usabilidade, mesmo para um único desenvolvedor com recursos limitados.

Como ilustrado na **Figura 3**, a tela inicial do sistema (*Home*) oferece uma visão clara e organizada, facilitando a navegação e interação do usuário.

Figura 3 - Tela inicial do sistema (Home)



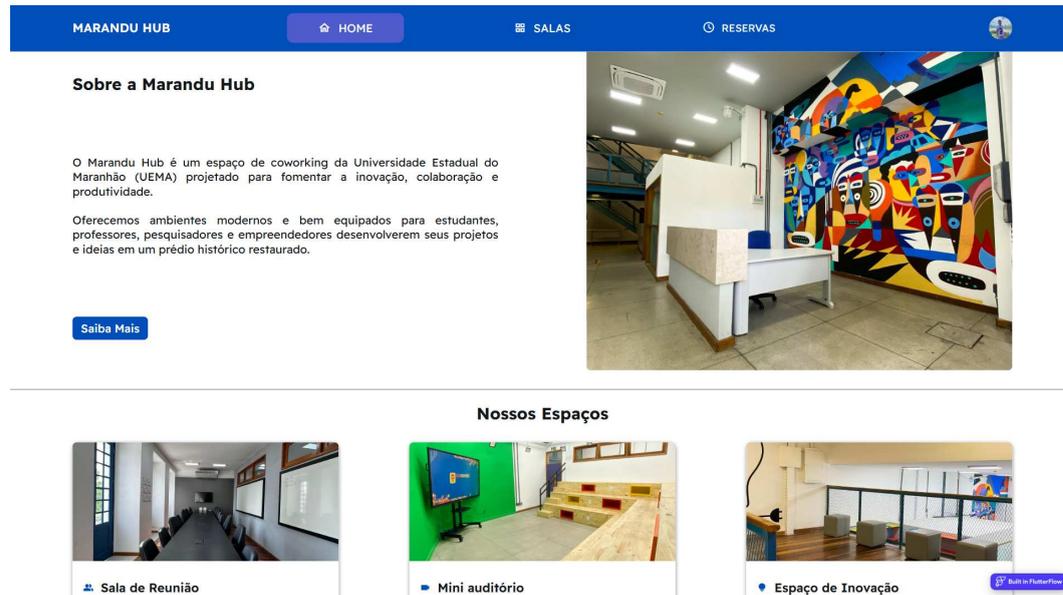
Fonte: Autor (2025)

Nessa camada estão incluídas funcionalidades como:

- Apresentação institucional do hub;
- Navegação entre as diferentes seções do sistema (salas, reservas, relatórios, etc.);
- Formulários com validações visuais (por exemplo: campos obrigatórios, formato de e-mail ou CPF);
- Visibilidade condicional de elementos, como botões e menus, com base no tipo de usuário (“admin” ou “user”).

Essa abordagem garantiu que a experiência do usuário fosse fluida, intuitiva e responsiva, mesmo em dispositivos móveis. Na **Figura 4**, é possível observar outro trecho da página *Home*, destacando mais detalhes da estrutura visual e interação.

Figura 4 - Outro trecho da página Home



Fonte: Autor (2025)

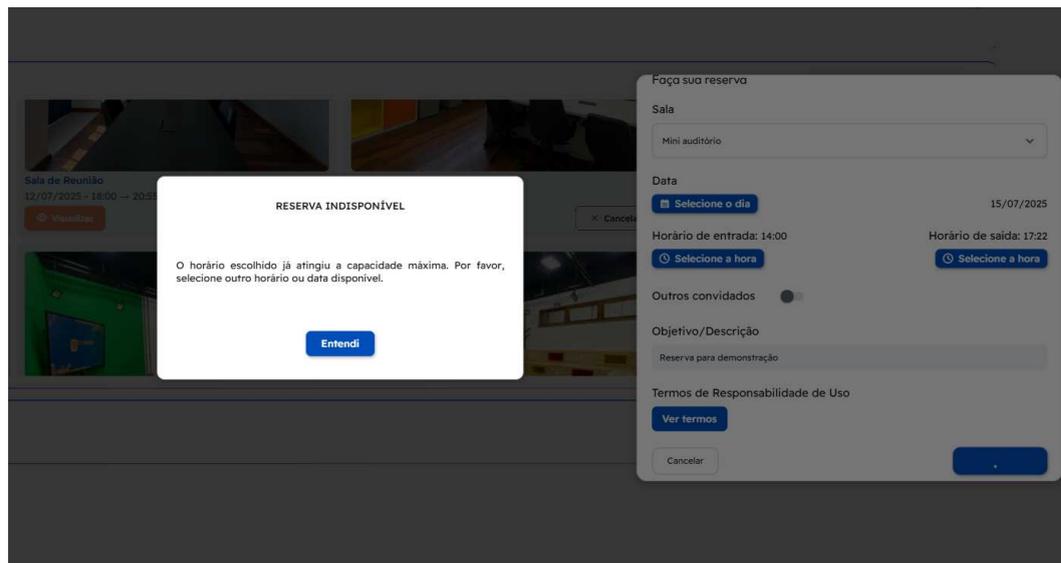
4.2.2 Camada de Lógica e Dados (FlutterFlow + Firebase + Dart)

Essa camada atua como o "cérebro" da aplicação, sendo responsável por interpretar ações do usuário, aplicar regras de negócio e manipular dados. Ela é composta por:

- **Cloud Firestore**, o banco de dados NoSQL que armazena informações estruturadas em coleções como *users*, *rooms*, *reservations* e *incubatedCompanies*;
- **Funções personalizadas em Dart (*Custom Functions*)**, usadas para converter e tratar dados em tempo real, como horários e formatos de datas;
- **Ações personalizadas (*Custom Actions*)**, que executam operações complexas, como validações de conflito de reservas, envio de e-mails e integração com outras camadas.

Por meio dessa estrutura, a lógica do sistema consegue impedir reservas duplicadas, validar horários, filtrar usuários vinculados a empresas incubadas e garantir o correto fluxo entre o que o usuário vê e o que está registrado no banco de dados. Essa validação é ilustrada na **Figura 5**, que mostra o alerta exibido quando há tentativa de agendamento para um horário já ocupado.

Figura 5 - Tentativa de reserva com alerta de reserva indisponível



Fonte: Autor (2025)

4.2.3 Camada de Integração Externa (Node.js + API em Vercel)

Essa camada foi desenvolvida para lidar com tarefas que exigem maior flexibilidade e capacidade de processamento, especialmente aquelas que vão além das funcionalidades nativas do FlutterFlow. O principal recurso dessa camada é uma *API* em Node.js, hospedada na plataforma Vercel, que realiza a geração de planilhas no formato Excel (.xlsx) com base em filtros definidos pelos administradores.

Essa *API* é acionada a partir do FlutterFlow por meio de uma *Custom Action* que envia os dados via requisição HTTP (POST). A *API* processa os dados, monta a planilha com a biblioteca ExcelJS e devolve o arquivo pronto para download.

4.2.4 Camada de Comunicação (EmailJS)

Para garantir a comunicação entre sistema e usuário, foi implementada uma camada específica para envio de mensagens automáticas por e-mail. Utilizando o serviço EmailJS, a aplicação envia notificações nos seguintes cenários:

- Solicitação de reserva (enviado para e-mail administrador da Marandu);
- Confirmação ou cancelamento da reserva por parte do administrador (enviado para o usuário solicitante);

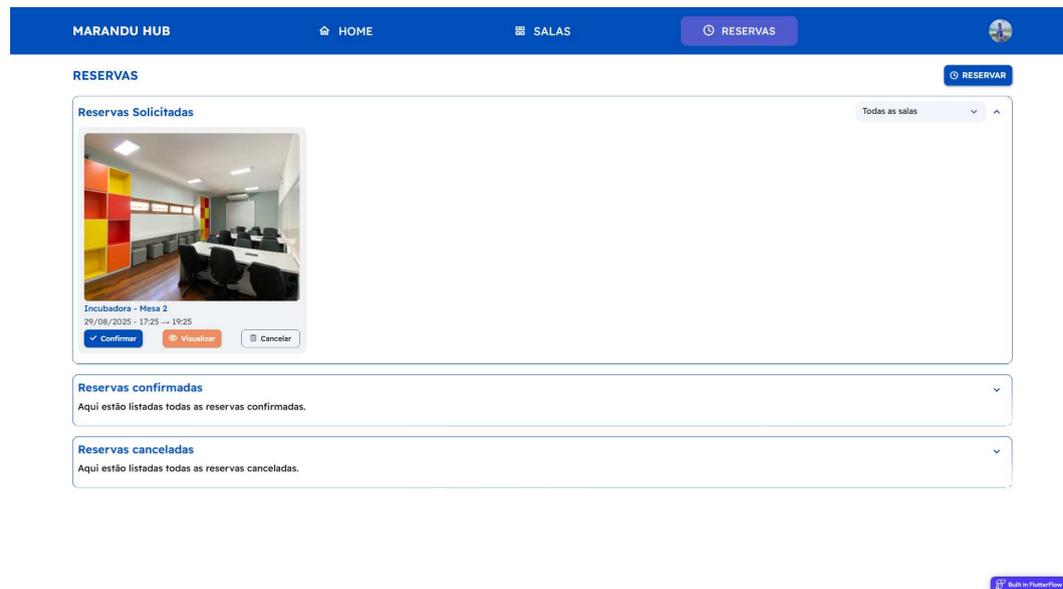
Essa funcionalidade foi integrada ao **FlutterFlow por meio de *Custom Actions***, que são acionadas diretamente em eventos como cliques em botões ou mudanças de status dentro da lógica do sistema. Dessa forma, o envio de e-mails ocorre de maneira automática, proporcionando uma comunicação eficiente sem exigir o uso de servidores próprios para isso.

Considerando a **limitação do plano gratuito do EmailJS**, que permite o **envio de até 200 mensagens por mês**, foi adotada uma solução prática: a **criação de duas contas distintas** no serviço. Uma é responsável pelos e-mails de **confirmação de reserva**, enquanto a outra cuida das mensagens de **solicitação e cancelamento**. Cada conta foi configurada com seus próprios *templates* de e-mail e conectada ao sistema com *Custom Actions* específicas.

Essa separação permite distribuir o volume de envios e **garantir que o sistema continue funcionando sem interrupções**, mesmo com o uso gratuito da ferramenta — assegurando uma experiência estável e transparente para os usuários.

Na **Figura 6**, observa-se a tela de reservas com destaque para aquelas que ainda aguardam confirmação.

Figura 6 - Tela de reservas com foco nas solicitadas



Fonte: Autor (2025)

Os modelos de e-mail gerados automaticamente pelo sistema podem ser vistos nas **Figuras 7 a 9**, que ilustram os cenários de solicitação, confirmação e cancelamento de reservas.

Figura 7 - E-mail para reserva solicitadas



Fonte: Autor (2025)

Figura 8 - E-mail para reserva confirmadas



Fonte: Autor (2025)

Figura 9 - E-mail para reserva canceladas



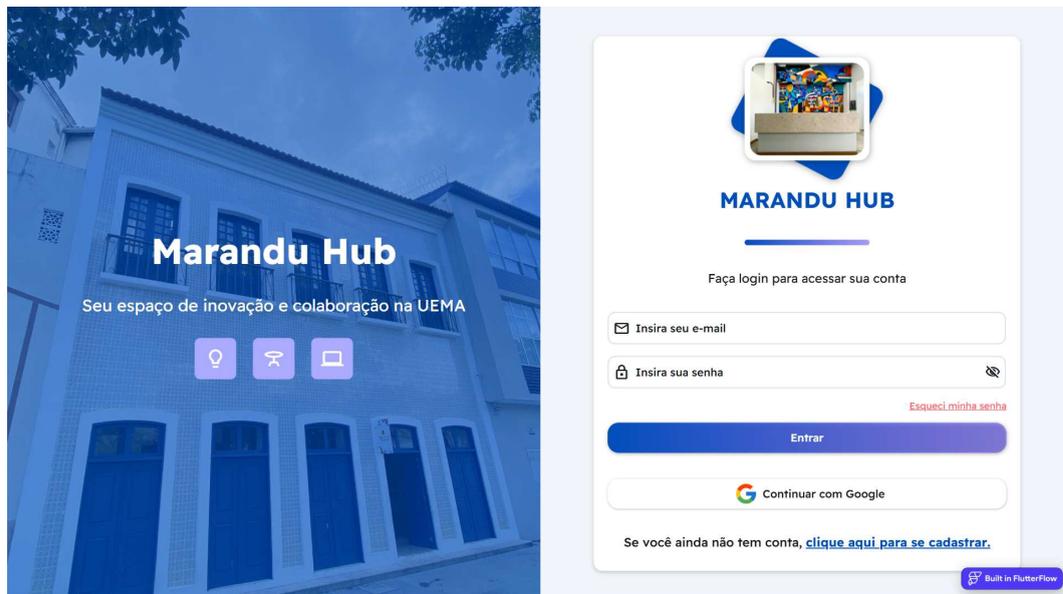
Fonte: Autor (2025)

4.2.5 Camada de Autenticação e Segurança (Firebase Auth + Firestore Rules)

Por fim, a camada responsável por controlar o acesso ao sistema e proteger os dados sensíveis é baseada nos serviços do *Firebase Authentication* e nas regras de segurança do *Firestore*. Cada usuário é autenticado no momento do login, e as permissões são controladas conforme o tipo de perfil (“admin” ou “user”), armazenado na coleção *users*.

Os usuários podem acessar o sistema utilizando autenticação via e-mail (com senha) ou login direto pela conta Google, recurso oferecido pelo *Firebase Authentication*. Isso facilita o acesso tanto para usuários técnicos quanto não técnicos, garantindo flexibilidade e segurança no processo de autenticação. A interface de autenticação é apresentada na **Figura 10**, onde o usuário pode optar entre login com e-mail/senha ou conta Google.

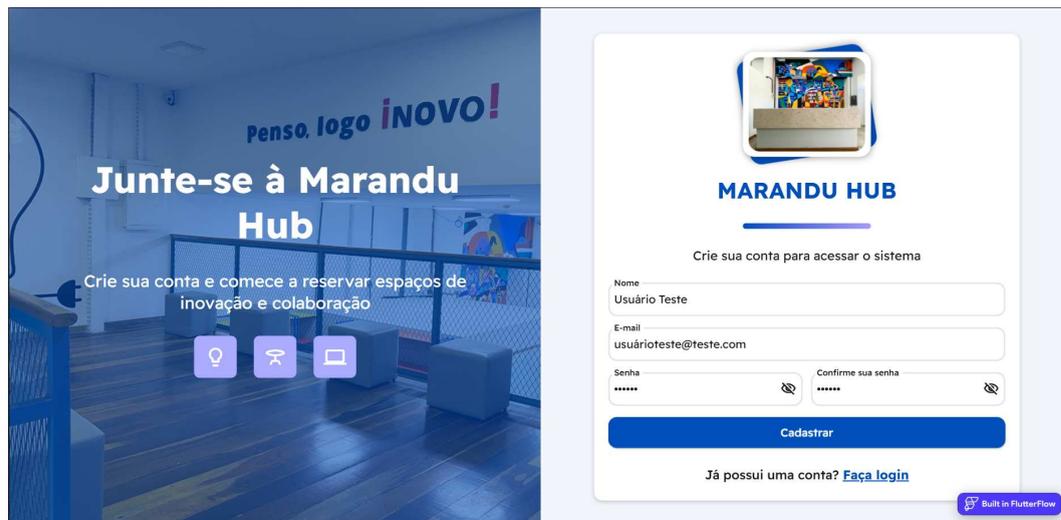
Figura 10 - Tela de login



Fonte: Autor (2025)

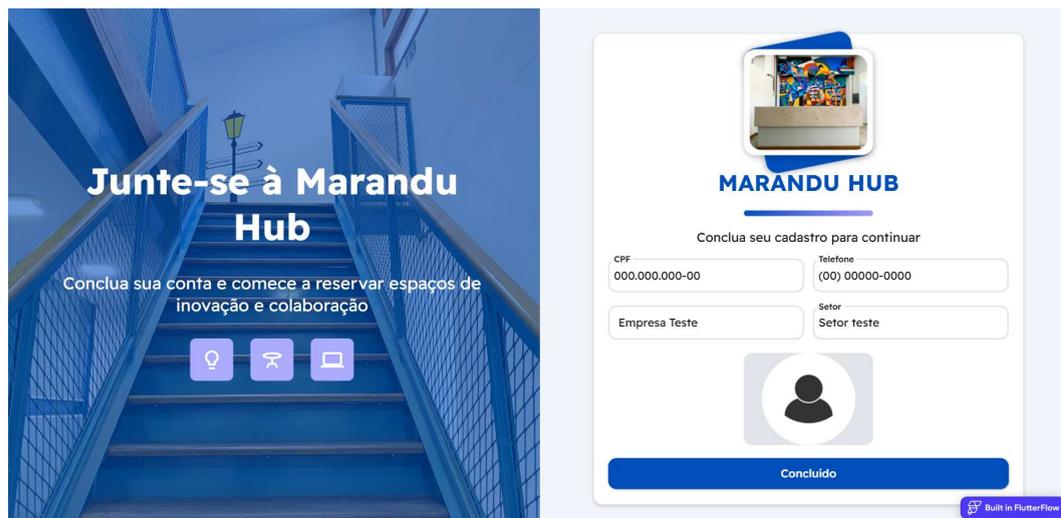
Se for o primeiro acesso, o usuário realiza o cadastro inicial (conforme ilustrado na **Figura 11**) — seja por e-mail ou Google. Em seguida, completa seu perfil com dados institucionais, como mostrado na **Figura 12**. Essa etapa de preenchimento de dados se aplica a ambos os métodos de cadastro, ou seja, tanto para acesso via Google quanto via e-mail.

Figura 11 - Tela de cadastro com e-mail



Fonte: Autor (2025)

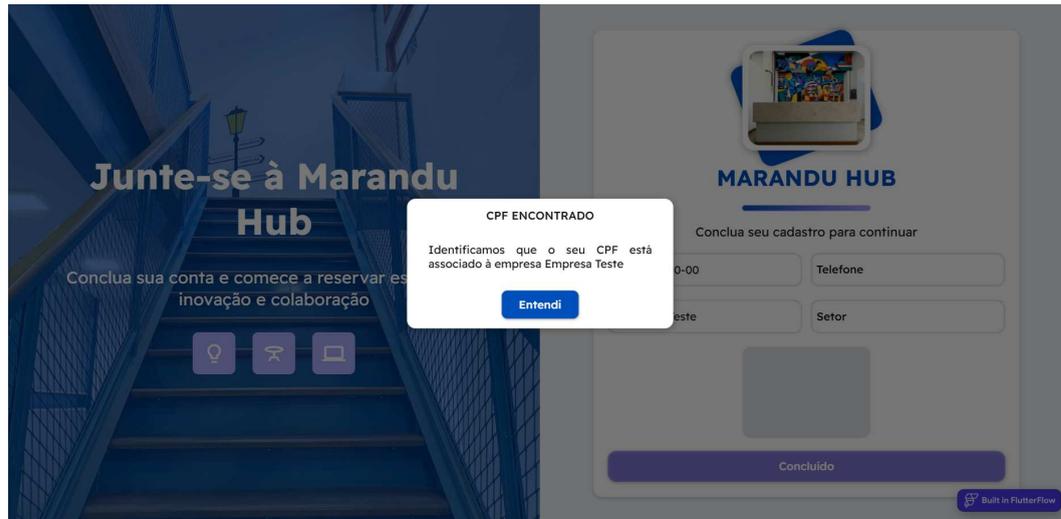
Figura 12 - Tela de finalização de conta (dados institucionais)



Fonte: Autor (2025)

Além disso, foi implementada uma lógica adicional para identificar usuários vinculados a empresas incubadas. Essa verificação é feita com base no **CPF do usuário logado**, que é comparado, em tempo real, com os CPFs cadastrados nas empresas incubadas. Essa funcionalidade foi essencial para permitir regras diferenciadas de agendamento, mantendo, ao mesmo tempo, a simplicidade na estrutura de dados. Como mostrado na **Figura 13**, o sistema exibe uma notificação sempre que o CPF informado corresponde a um colaborador de empresa incubada.

Figura 13 - Mensagem indicando CPF vinculado a uma empresa incubada



Fonte: Autor (2025)

4.3 Estrutura de Dados (Coleções Firebase)

Para garantir o funcionamento adequado do sistema Marandu Hub, a modelagem de dados no **Cloud Firestore** foi cuidadosamente estruturada a partir das necessidades reais do ambiente institucional e das funcionalidades previstas. A escolha por um banco de dados **NoSQL**, como o Firestore, permitiu maior flexibilidade na organização das informações, adaptando-se bem ao caráter dinâmico das reservas e ao relacionamento entre os usuários e os espaços físicos.

A aplicação trabalha com quatro coleções principais: *users*, *rooms*, *reservations* e *incubatedCompanies*. Cada uma delas foi modelada com campos específicos, de forma a manter a coerência dos dados, otimizar consultas e garantir a escalabilidade do sistema em uso real.

4.3.1 Coleção users: Cadastro e Perfil de Usuários

A coleção *users* é responsável por armazenar os dados cadastrais e informações de autenticação dos usuários da plataforma. Cada documento representa um indivíduo autenticado via Firebase Authentication, com campos que vão além dos dados básicos, incluindo elementos importantes para controle de acesso e identificação institucional.

Tabela 2 - Estrutura de Dados da Coleção *users*

Campo	Tipo	Descrição
uid	string	ID do usuário
display_name	string	Nome do usuário
email	string	E-mail
cpf	string	CPF
instituicao	string	Instituição vinculada
photo_url	string	URL da foto (opcional)
phone_number	string	Telefone
created_time	datetime	Data de criação do cadastro
setor	string	Setor interno
typeUser	string	"admin" ou "user"

Fonte: Autor (2025)

4.3.2 Coleção *rooms*: Cadastro de Salas

A coleção *rooms* contém os dados referentes aos espaços físicos disponíveis para reserva na Marandu Hub. Cada sala, mesa compartilhada ou auditório é representado por um documento contendo características que orientam os usuários e fundamentam regras de agendamento.

Tabela 3 - Estrutura de Dados da Coleção *rooms*

Campo	Tipo	Descrição
uid	string	ID da sala
name	string	Nome da sala
type	string	"Incubadora", "Reunião", "Auditório"
description	string	Descrição da sala
capacity	number	Capacidade máxima
location	string	Localização
photos	array<string>	Lista de URLs de imagens

Fonte: Autor (2025)

4.3.3 Coleção reservations: Controle de Reservas

A coleção *reservations* centraliza o processo de agendamento dos espaços, com registros que conectam usuários e salas em horários específicos. Essa modelagem permite aplicar regras de negócio como verificação de conflitos, identificação de usuários incubados e filtragem por status.

Tabela 4 - Estrutura de Dados da Coleção *reservations*

Campo	Tipo	Descrição
uid	string	ID da reserva
idUser	string	ID do usuário
idRoom	string	ID da sala
date	datetime	Data da reserva
inicialHour	datetime	Hora de início
finalHour	datetime	Hora de término
numberGuests	string	Número de convidados
desc	string	Descrição da reserva
status	string	"confirmed", "pending", "canceled"

Fonte: Autor (2025)

4.3.4 Coleção incubatedCompanies: Empresas Incubadas

A coleção *incubatedCompanies* representa as empresas incubadas na Marandu Hub. Essa estrutura permite associar colaboradores às respectivas organizações, utilizando o CPF como chave de verificação no momento da solicitação de reservas. Assim, o sistema pode liberar permissões específicas, como o acesso direto às mesas da incubadora.

Tabela 5 - Estrutura de Dados da Coleção *incubatedCompanies*

Campo	Tipo	Descrição
name	string	Nome da empresa
usersCompany	array<object>	Lista de colaboradores (nome, CPF)

Fonte: Autor (2025)

Essa modelagem de dados buscou encontrar o equilíbrio entre simplicidade e funcionalidade. A utilização de identificadores únicos (uid) e o uso de subestruturas (como *array<object>*) garantem flexibilidade na manipulação dos dados, ao mesmo tempo em que preservam a integridade relacional entre as entidades do sistema.

Além disso, o uso de campos bem definidos permite que filtros, dashboards e relatórios sejam gerados de forma eficiente, atendendo às necessidades administrativas e operacionais do hub. A separação entre usuários comuns e administradores, bem como a diferenciação de permissões por meio da vinculação com empresas incubadas, reforçam o caráter dinâmico e adaptável do sistema.

4.4 Controle de Acessos e Perfis de Usuário

A gestão de acessos no sistema Marandu Hub foi projetada para equilibrar simplicidade, segurança e flexibilidade. Para isso, a modelagem de perfis de usuário foi estruturada de forma enxuta, utilizando o campo *typeUser* na coleção *users* para diferenciar os dois níveis principais de permissão: usuários administradores e usuários comuns.

A seguir, a **Tabela 6** apresenta as permissões atribuídas a cada perfil:

Tabela 6 - Níveis de acesso

Tipo de usuário	<i>typeUser</i>	Permissões Principais
Administrador	“admin”	Acesso total ao sistema: confirmação/cancelamento de reservas, relatórios e gestão de empresas incubadas.
Usuário comum	“users”	Pode visualizar e solicitar reservas, além de editar seus próprios dados cadastrais.

Fonte: Autor (2025)

Esse modelo permite um controle direto e eficiente de permissões, ao mesmo tempo em que mantém a base de dados organizada e de fácil manutenção.

4.4.1 Usuários Vinculados a Empresas Incubadas

Além da distinção por tipo de perfil, o sistema implementa uma verificação dinâmica de vínculo com empresas incubadas, com base no campo cpf. Essa lógica foi criada para

permitir que colaboradores de empresas incubadas na Marandu Hub tenham acesso ampliado aos espaços do *coworking*, de forma automatizada e sem necessidade de validação manual por parte da administração.

- **Funcionamento da Lógica:**

No momento da solicitação de reserva, o sistema executa uma *Custom Action* no FlutterFlow que:

- Consulta a coleção *incubatedCompanies*;
- Compara o CPF do usuário logado com os CPFs cadastrados na lista de colaboradores (*usersCompany*);
- Caso haja correspondência, o sistema reconhece o usuário como vinculado à incubadora.

- **Permissões adicionais:**

- Podem reservar todos os tipos de salas, incluindo mesas da incubadora;
- As reservas são confirmadas automaticamente, desde que não haja conflito de horário com outra já existente;
- Não necessitam de aprovação administrativa para suas reservas.

4.4.2 Usuários Não Vinculados a Empresas Incubadas

Usuários que não possuem vínculo com empresas incubadas têm acesso mais restrito às funcionalidades do sistema, o que reflete a política de uso preferencial do espaço pela comunidade residente.

- **Regras aplicadas:**

- Podem reservar apenas o **mini auditório** e as **salas de reunião**;
- Suas reservas ficam inicialmente com o status "pending" e **devem ser validadas** por um administrador;
- Não têm acesso às mesas da incubadora;
- A lógica de verificação por CPF é executada em tempo real, **garantindo agilidade e segurança no controle de acesso.**

Esse modelo de permissões dinâmicas mantém a estrutura de usuários simples, sem a necessidade de criar múltiplos níveis fixos de acesso, mas ainda assim garante flexibilidade operacional com base no contexto de uso.

4.5 Módulo de Relatórios e Visualizações

A área administrativa do sistema conta com um módulo completo de relatórios, permitindo que os gestores da Marandu Hub acompanhem a ocupação dos espaços, identifiquem padrões de uso e tomem decisões baseadas em dados. O módulo foi dividido em três seções principais:

4.5.1 Visualização em Tabela

Exibe uma listagem detalhada das reservas registradas. Essa listagem apresenta informações cruciais como o **E-mail** do usuário, sua **Instituição** de origem, o **Nome da Sala** reservada, a **Data**, a **Hora Inicial** e a **Hora Final** do agendamento, o número de **Convidados**, o **Status** do agendamento e a **Descrição** fornecida pelo usuário. Na **Figura 14**, é possível visualizar essa listagem detalhada de reservas, que faz parte do módulo de relatórios do sistema.

Figura 14 - Listagem com as reservas registradas

Resultado do Relatório
Selecione os filtros para gerar o relatório de reservas

E-mail	Instituição	Nome da sala	Data	Hora inicial	Hora final	Convidados	Status	Descrição
Medeirosanna2@gmail.com	AGÊNCIA MARANDU	Sala de Reunião	10/07/2025	13:00	17:30	0	Confirmado	Acompanhamento
gabipanqueca@gmail.com	UEMA	Sala de Reunião	10/07/2025	08:00	18:00	0	Cancelada	Palestra
gabipanqueca@gmail.com	UEMA	Sala de Reunião	09/07/2025	18:00	20:55	0	Confirmado	PROFITE
gabipanqueca@gmail.com	UEMA	Sala de Reunião	09/07/2025	18:00	18:00	0	Cancelada	Mentoria quizanal

Linhas por página: 10 41 - 50 de 193

Fonte: Autor (2025)

4.5.2 Exportação em Excel

Para facilitar o compartilhamento e análise dos dados, foi implementada uma funcionalidade de **exportação em formato .xlsx**, utilizando uma **Custom Action** que:

- Prepara os dados com base nos filtros definidos;
- Envia os dados via requisição POST para uma **API em Node.js** hospedada na Vercel;
- A **API** converte os dados com uso do pacote **ExcelJS** e retorna o arquivo gerado para download.

A seleção de filtros pode ser realizada conforme mostrado na **Figura 15**, que exibe os campos disponíveis para a geração de relatórios. Os filtros disponíveis são:

- Por sala específica ou “Todas as salas”;
- Por intervalo de datas ou “Todas as datas”.

Figura 15 - Filtros para relatórios

Filtros
Selecione os filtros para gerar o relatório de reservas

Nome da sala: Datas das reservas Todas as reservas

De 01/07/2025 a 31/07/2025

relatório de reservas

Nome da sala	Data	Hora inicial	Hora final	Convidados	Status	Descrição
Sala de Reunião	10/07/2025	13:00	17:30	0	Confirmado	Acompanhamento
Sala de Reunião	10/07/2025	08:00	18:00	0	Cancelada	Palestra

Fonte: Autor (2025)

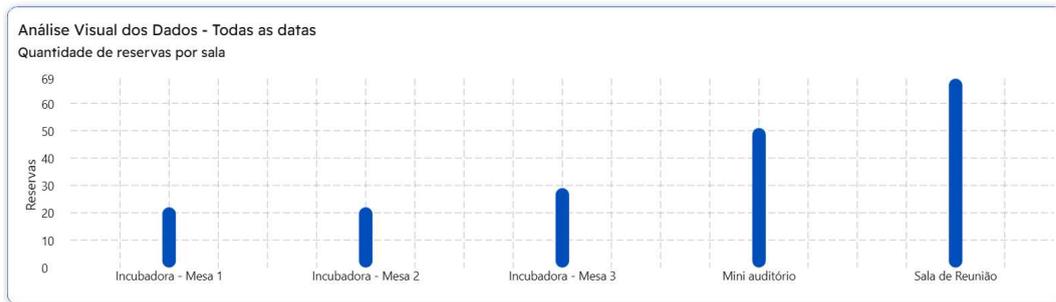
4.5.3 Dashboards Visuais

Além dos dados em planilhas, o sistema fornece **gráficos dinâmicos** com diferentes formas de visualização, facilitando a análise de gerenciamento. As representações disponíveis estão:

- **Reservas por sala**
- **Distribuição por status**
- **Reservas por empresa**
- **Total de horas utilizadas por empresa**
- **Evolução temporal**

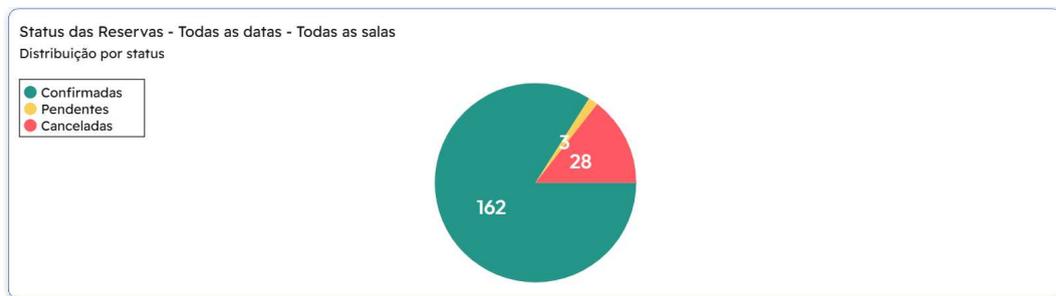
Essas visualizações, presentes nas **figuras 16 a 20**, ajudam os administradores a tomar decisões assertivas com base em dados reais. O sistema contribui para uma gestão mais eficiente dos ambientes da Marandu Hub, otimizando recursos e melhorando a experiência dos usuários.

Figura 16 - Dashboard de reservas por sala



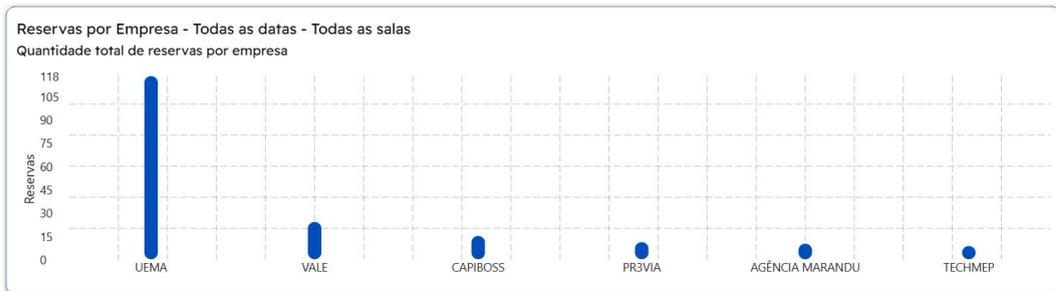
Fonte: Autor (2025)

Figura 17 - Dashboard por status



Fonte: Autor (2025)

Figura 18 - Dashboard de reservas por empresa



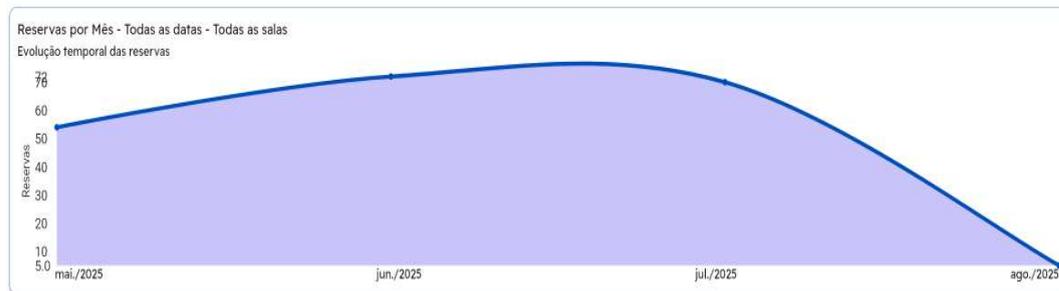
Fonte: Autor (2025)

Figura 19 - Dashboard total de horas utilizadas por empresa



Fonte: Autor (2025)

Figura 20 – Dashboard de evolução temporal



Fonte: Autor (2025)

Essa abordagem visual reforça o papel estratégico do sistema na **gestão eficiente do espaço físico** da Marandu Hub.

4.6 Funcionalidades Personalizadas e Regras de Negócio

Para atender às necessidades específicas da instituição, a lógica de negócio do sistema foi construída com **ações e funções personalizadas em Dart**, explorando ao máximo os recursos do FlutterFlow. A seguir, são apresentadas as principais implementações:

- **Custom Functions (3)**

Funções personalizadas em Dart que podem ser usadas para definir propriedades de *widget* ou ação (FLUTTERFLOW, 2025). Essas funções foram utilizadas para:

- Converter hora após a seleção no *widget* de *timer*;
- Converter minutos após a seleção no mesmo *widget*;
- Formatar datas para exibição no *custom widget* de calendário criado.

- **Custom Actions (20)**

Funções em Dart personalizadas que podem ser acionadas por Action Triggers⁴ ou usadas como nós em um flow de ação (FLUTTERFLOW, 2025). Algumas das principais funcionalidades implementadas incluem:

- Envio de e-mails (via EmailJS):

Mensagens automáticas são enviadas aos usuários nos seguintes casos:

- Solicitação de reserva;

⁴ Action Triggers representam eventos específicos que ocorrem quando o usuário interage com o aplicativo, como tocar em um botão, selecionar uma opção em um menu suspenso ou carregar uma nova página (FLUTTERFLOW, 2025).

- Confirmação da reserva pelo administrador;
- Cancelamento da reserva.

A funcionalidade é implementada com *Custom Actions* no FlutterFlow que utilizam o serviço EmailJS, responsável por processar e entregar os e-mails de forma segura e padronizada. Essa abordagem garante comunicação eficaz sem depender de *backend* próprio para envio de mensagens.

- **Validações:**

- Verificação de conflitos de horário;
- Garantia de que a data e hora estão dentro do horário de funcionamento (07:30 às 21:30);
- Bloqueio de agendamentos aos domingos.

- **Controle de lógica de reserva:**

- Verificação da capacidade máxima da sala
- Identificação do perfil do usuário (incubado ou não)
- Ajuste dinâmico da visibilidade de botões e mensagens, conforme contexto

- **Preparação de dados para relatórios:**

- Prepara dados para visualização dos gráficos dinâmicos
- Conversão de dados para string
- Montagem dos dados formatados para envio à *API* de exportação

- ***Custom widget* (1)**

Widgets personalizados do *Flutter* que também podem importar dependências de pacotes personalizados e ser usados da mesma forma que componentes em todo o projeto (FLUTTERFLOW, 2025). Foi desenvolvido exclusivamente para a seleção de intervalos de datas em calendário, recurso não disponível nativamente no FlutterFlow. Esse *widget* é utilizado nos filtros do painel de relatórios e permite maior precisão e usabilidade na definição dos períodos desejados pelos administradores. A funcionalidade de seleção de intervalo de datas, desenvolvida como *widget* personalizado, é mostrada na **Figura 21**.

Figura 21 - *Custom Widget* criado

Fonte: Autor (2025)

4.7 Hospedagem do Sistema

O sistema Marandu Hub está hospedado diretamente na plataforma FlutterFlow, que oferece serviço integrado de publicação e hospedagem para projetos desenvolvidos em seu ambiente. Dessa forma, a aplicação fica disponível online, acessível para testes e uso real, garantindo facilidade de manutenção e atualizações contínuas. O sistema pode ser acessado no link: marandu-hub.flutterflow.app.

5 RESULTADOS E DISCUSSÃO

5.1 Avaliação da Usabilidade com a Escala SUS

A adoção do sistema de reservas da Marandu Hub trouxe mudanças significativas no processo de agendamento, beneficiando diretamente **administradores e, principalmente, todos os usuários do espaço – desde *freelancers* e empreendedores individuais até equipes de *startups* e empresas incubadas**. Agora, esses usuários não precisam mais enviar e-mails ou mensagens para tentar reservar uma sala. Basta acessar o sistema, escolher a data desejada e selecionar a sala disponível, tornando o processo mais simples, direto e eficiente.

Para avaliar a usabilidade do sistema de reservas da Marandu Hub, foi aplicado um questionário com base na **System Usability Scale (SUS)**, uma metodologia reconhecida internacionalmente para medir a experiência do usuário (Brooke, 1996). Composta por dez declarações simples (listadas na íntegra no Apêndice B), a escala permite avaliar diversos aspectos da usabilidade de um sistema, como a percepção de facilidade de uso, a complexidade aparente, a necessidade de suporte e a confiança na interação. Sua simplicidade e eficácia a tornaram um padrão internacional para a avaliação rápida e quantitativa da usabilidade. A coleta de informações ocorreu através do **Jotform**, plataforma de formulário online, permitindo que um Termo de Consentimento Livre e Esclarecido (TCLE) — disponível no Apêndice A — fosse exibido assim que o formulário era aberto, assegurando que a participação fosse voluntária e com o anonimato garantido.

Participaram da pesquisa **dez usuários do sistema**, entre **administradores e usuários** comuns, que já haviam interagido com a aplicação em uso real. Por se tratar de uma pesquisa anônima, não foi possível obter dados demográficos ou de perfil detalhado dos respondentes. No entanto, todos os participantes relataram familiaridade com as funcionalidades básicas da aplicação, o que valida sua participação como representativa do público-alvo.

5.2 Estrutura da Escala e Cálculo

A Escala SUS é composta por **dez declarações**, apresentadas no Apêndice B, que avaliam diversos aspectos da experiência do usuário — como facilidade de uso, consistência, necessidade de ajuda externa, e confiança na utilização. As respostas são dadas em uma avaliação por escala de 5 pontos, que varia de 1 ("discordo totalmente") a 5 ("concordo totalmente").

A pontuação final de cada participante é calculada por meio da seguinte lógica:

- Para os **itens ímpares**, subtrai-se 1 da nota dada.
- Para os **itens pares**, subtrai-se a nota de 5.
- A soma dos valores ajustados é multiplicada por **2,5**, resultando em uma pontuação total entre 0 e 100.

Segundo Brooke (1996), **pontuações acima de 68 indicam boa usabilidade**, enquanto valores acima de 80 são considerados excelentes.

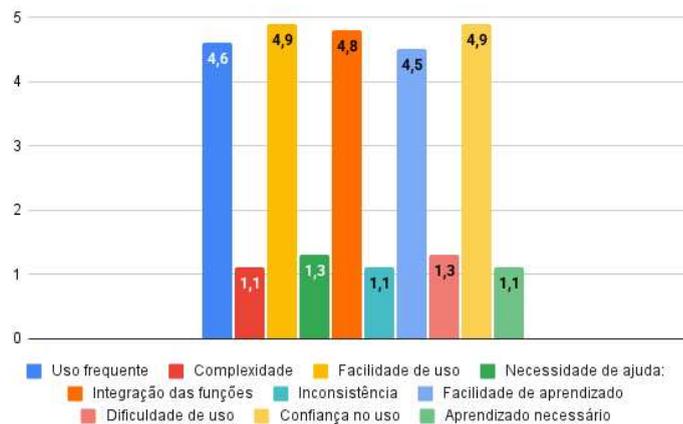
5.3 Análise dos Resultados

Neste estudo, a média geral alcançada foi de **94,5 pontos**. De acordo com a literatura especializada sobre a Escala SUS, uma **pontuação acima de 80,3** é classificada como **excelente**, o que posiciona o sistema da Marandu Hub na **categoria mais alta de usabilidade**. Esse resultado evidencia que o sistema não só atende à sua função técnica, mas também está alinhado com as boas práticas de design centrado no usuário, demonstrando um alto grau de satisfação.

A **excelente avaliação de usabilidade reflete diretamente o impacto positivo das escolhas de design e das tecnologias *low-code* empregadas**. A **interface visual intuitiva do FlutterFlow**, com seus componentes arrastáveis e personalizáveis, foi fundamental para que os usuários percebessem o sistema como fácil de usar e com uma curva de aprendizado baixa. As **integrações com Firebase para autenticação e banco de dados**, aliadas às **Custom Actions e Custom Functions** desenvolvidas em **Dart**, permitiram a implementação de lógicas de negócio complexas, como validações de agendamento e notificações automáticas por EmailJS, de forma transparente e eficiente para o usuário final, resultando em alta confiança na ferramenta.

O gráfico a seguir apresenta a média das respostas obtidas para cada um dos dez itens da escala (Q1 a Q10), possibilitando uma análise detalhada dos pontos fortes e eventuais oportunidades de melhoria da interface.

Gráfico 1 - Média das Respostas por Item da Escala SUS



Fonte: Autor (2025)

Os **maiores índices** de concordância foram registrados nas afirmações relacionadas à **facilidade de uso** e à **confiança na operação do sistema** (itens 1, 3, 5, 7 e 9), evidenciando que a curva de aprendizado é baixa e que os usuários se sentem confortáveis ao interagir com a ferramenta. As **menores médias** — ainda assim, **altamente positivas** — apareceram nos itens negativos (2, 4, 6, 8 e 10), como percepção de complexidade e necessidade de suporte técnico, reforçando a estabilidade e maturidade da solução proposta. O resultado obtido nessas questões negativas, especificamente, sugere que o sistema é percebido como **coerente, integrado** e de **fácil navegação**, necessitando minimamente de apoio externo para sua utilização.

5.4 Análise Crítica e Trabalhos Futuros

Embora a pontuação de **94,5 na Escala SUS** seja excepcional e demonstre a eficácia do sistema em proporcionar uma experiência de usuário altamente satisfatória, é crucial **aprofundar a discussão sobre a percepção dos usuários e as implicações estratégicas para a gestão da Marandu Hub**. A alta usabilidade percebida pelos usuários valida que a transição do processo manual para o digital foi **bem-sucedida**, facilitando o acesso e a realização de reservas. Essa percepção positiva é um indicativo forte de que o sistema está cumprindo seu objetivo principal de otimizar a rotina de agendamento e reduzir a fricção para quem utiliza as salas.

Contudo, a **análise** detalhada das respostas aos itens individuais, apesar das médias positivas, pode fornecer **observações adicionais**. Por exemplo, mesmo que as médias para "complexidade" e "necessidade de suporte" tenham sido baixas, a existência de qualquer

discordância nesses itens sugere **pequenas oportunidades de refinamento na interface** ou em **possíveis guias de uso para cenários específicos**. A percepção do usuário sobre a simplicidade e a autonomia ao usar o sistema diretamente impacta a carga de trabalho da gestão, liberando o técnico administrativo de tarefas repetitivas e permitindo que ele se concentre em atividades mais estratégicas para o *coworking*.

Além dos benefícios operacionais evidentes para a Marandu Hub, que agora conta com uma gestão de reservas digitalizada e menos suscetível a falhas manuais, os resultados demonstram a viabilidade e o potencial do *low-code* como uma abordagem eficiente para o desenvolvimento de soluções institucionais. **Academicamente, este estudo serve como um caso de sucesso que reforça a proposição de que ferramentas *low-code* podem democratizar o desenvolvimento de sistemas e acelerar a inovação em contextos com recursos limitados.** A alta usabilidade alcançada valida o FlutterFlow como uma ferramenta robusta para entregar experiências de usuário de alta qualidade, provando que o desenvolvimento com baixo código não implica em sacrificar a qualidade da interface ou a eficiência da interação. Isso posiciona o projeto como um modelo para futuras implementações em ambientes universitários ou pequenas e médias empresas que buscam a digitalização de processos sem a necessidade de grandes equipes de programação ou orçamentos.

É importante, entretanto, contextualizar esses resultados considerando algumas **limitações da avaliação de usabilidade**. A pesquisa foi realizada com uma **amostra de apenas dez usuários**, o que, embora aceitável para a aplicação da Escala SUS em testes exploratórios, **limita a generalização dos resultados para uma população mais ampla**. Além disso, a natureza das respostas da Escala SUS é **subjetiva** e, como os participantes já eram usuários familiarizados com o sistema, pode haver um potencial **viés de familiaridade**. Adicionalmente, reconhece-se que a Escala SUS, por sua natureza padronizada de questões fechadas, não permite a captura de sugestões abertas ou detalhes específicos sobre melhorias percebidas pelos usuários. Essa limitação é inerente à metodologia escolhida para garantir a mensuração quantitativa e a comparabilidade da usabilidade. Para validar esses resultados em uma escala maior e aprofundar a compreensão da experiência do usuário, estudos futuros com amostras mais robustas e metodologias complementares que incluam questões abertas ou entrevistas qualitativas seriam recomendados.

O sucesso alcançado, evidenciado pela alta pontuação de usabilidade, confirma a viabilidade do *low-code* para soluções como a Marandu Hub. No entanto, é fundamental que a gestão reconheça as implicações estratégicas e os riscos inerentes a essa escolha tecnológica.

A dependência de plataformas *low-code* e de serviços de terceiros (como o EmailJS) implica em considerações sobre a sustentabilidade a longo prazo. Riscos como a dependência do fornecedor, potenciais custos crescentes com o aumento da escala de uso e a flexibilidade limitada para personalizações complexas devem ser monitorados.

Quanto à escalabilidade e segurança, o sistema foi arquitetado sobre o Firebase, uma plataforma que, por natureza, é capaz de lidar com grande volume de usuários e dados, com recursos de segurança avançados que podem ser configurados conforme a demanda. Portanto, a segurança e a escalabilidade são inerentes à escolha tecnológica. Em um cenário hipotético de duplicação da base atual de usuários e reservas, o Firebase estaria bem preparado para suportar o crescimento em termos de recursos computacionais e capacidade de banco de dados, exigindo eventualmente a transição para planos pagos conforme o volume de uso. Entretanto, é reconhecido que serviços como o EmailJS possuem limitações em seus planos gratuitos (200 e-mails por mês), sendo uma consideração importante para um futuro cenário de alto volume de notificações, onde a contratação de planos mais abrangentes seria necessária.

Para garantir a sustentabilidade e o uso adequado do sistema pela equipe da Marandu Hub, será disponibilizado um **manual de gestão de uso**. Este manual está em fase de planejamento com o professor orientador e terá como objetivo documentar as funcionalidades administrativas, procedimentos de manutenção básica e boas práticas para o gerenciamento das reservas, assegurando a transferência de conhecimento e a operação autônoma e eficiente no longo prazo.

A partir desta análise, emergem as seguintes reflexões para a evolução contínua do sistema:

- **Refinamento:** Os pequenos pontos de discordância notados nas questões de "complexidade" e "suporte" sugerem a necessidade de implementar funcionalidades que aprimorem a experiência de novos usuários, como guias interativos ou uma seção de perguntas frequentes (FAQ) na interface, reduzindo a necessidade de suporte direto.
- **Expansão:** Para aprimorar a experiência do usuário e a integração do sistema com o ecossistema de trabalho do hub, é recomendado ampliar as funcionalidades com integrações estratégicas, como a sincronização de reservas com agendas corporativas externas (Google Agenda, Outlook) e o envio de notificações por SMS para lembretes e status de reservas.
- **Análise de Dados:** Desenvolver um dashboard em uma ferramenta de *Business Intelligence (BI)* para ir além dos relatórios básicos, permitindo à gestão uma análise

mais estratégica e preditiva sobre o uso dos espaços, identificando padrões de demanda e otimizando a tomada de decisões.

- **Sustentabilidade:** A documentação interna do sistema e a capacitação contínua da equipe para gerenciar a plataforma são estratégias importantes para mitigar os riscos mencionados. A realização de análises periódicas de custo-benefício e a exploração de alternativas — como um eventual desenvolvimento *full-code* para maior controle e customização — devem ser consideradas para garantir a evolução da solução a longo prazo, caso o crescimento do sistema ultrapasse as capacidades do FlutterFlow. Essas ampliações, no entanto, podem apresentar desafios relacionados às limitações inerentes às plataformas *low-code*, como a personalização e flexibilidade para integrações muito específicas, ou o potencial de *vendor lock-in*, que poderiam demandar soluções mais complexas ou mesmo a transição para desenvolvimento *full-code* em estágios avançados. Caso a ferramenta necessite de uma expansão significativa que ultrapasse as capacidades do Flutterflow, as opções incluiriam desenvolvimento nativo (para iOS e Android) com linguagens como Swift/Kotlin ou Flutter/React Native para um maior controle sobre o código e o desempenho, ou a criação de um *backend* personalizado em linguagens como Python (com Django/Flask) ou Node.js (com Express) para maior flexibilidade de integração e escalabilidade específica. Também é importante considerar o desenvolvimento de versões móveis nativas para ampliar ainda mais o alcance e a acessibilidade, o que poderia ser uma estratégia para mitigar a dependência do FlutterFlow para certas funcionalidades ou para explorar o desempenho máximo em ambientes nativos.

6 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento e a implementação de um sistema digital de reservas para o espaço de *coworking* Marandu Hub, atendendo a uma demanda real identificada pela administração do local. A opção por uma **solução própria**, desenvolvida com ferramenta *low-code*, justifica-se pela **necessidade de aderência específica aos processos e regras de negócio da Marandu Hub**, que nem sempre são contempladas de forma flexível ou custo-efetiva por soluções de mercado prontas. Adicionalmente, a **abordagem customizada** permitiu uma **otimização de recursos** e um **controle direto** sobre a **evolução do sistema**, conforme as **necessidades locais**. A partir de uma metodologia aplicada, descritiva e qualitativa, foi possível criar uma solução tecnológica que digitaliza, organiza e otimiza o processo anteriormente manual, baseado em planilhas, proporcionando maior eficiência e controle no agendamento das salas.

A arquitetura modular adotada, fundamentada em ferramentas modernas como FlutterFlow, Firebase e Node.js, permitiu o desenvolvimento de um sistema flexível, seguro e escalável, que atende diferentes perfis de usuários, incluindo colaboradores vinculados a empresas incubadas. A integração de serviços externos, como EmailJS para comunicação automatizada, e a implementação de regras dinâmicas de negócio, asseguraram uma experiência de uso intuitiva e funcional.

O sistema foi desenvolvido e hospedado na web utilizando a plataforma FlutterFlow, que oferece suporte para desenvolvimento responsivo. Contudo, para garantir uma experiência consistente e fluida tanto em *desktops* quanto em dispositivos móveis, foram realizados ajustes manuais e correções nos componentes da interface. Essa atenção ao design responsivo permite que os usuários façam reservas de qualquer lugar, utilizando smartphones, tablets ou computadores, com uma usabilidade adequada para cada tipo de dispositivo.

A avaliação da usabilidade por meio da Escala SUS revelou uma excelente aceitação do sistema, com média de 94,5 pontos, indicando que a interface é simples, eficiente e confiável para seus usuários. Este resultado confirma que o projeto não apenas atingiu seu objetivo técnico, mas também foi desenvolvido com foco na experiência e satisfação do usuário final.

Como contribuição, o sistema Marandu Hub digitaliza um processo fundamental para a gestão do espaço, promove economia de tempo, reduz erros operacionais e potencializa o uso adequado das salas, refletindo diretamente na melhoria da produtividade e da organização do ambiente.

A escolha do **FlutterFlow** como principal ferramenta para o desenvolvimento, em detrimento de outras ferramentas *low-code* ou até mesmo *no-code* como o **Glide (utilizado no Locomotiva Hub)**, foi fundamentada principalmente pela **familiaridade prévia do pesquisador com a plataforma e sua maior flexibilidade em comparação com soluções puramente *no-code***, o que proporcionou maior agilidade na prototipagem e implementação das funcionalidades. Adicionalmente, tanto o FlutterFlow quanto o Firebase, utilizado para o *backend*, ofereciam **planos gratuitos robustos** que eram plenamente adequados às necessidades iniciais e ao escopo do projeto, otimizando o uso de recursos.

Por fim, este projeto evidencia a viabilidade e os benefícios do uso de tecnologias *low-code* e serviços em nuvem para a criação de soluções práticas em ambientes institucionais. Embora essas plataformas apresentem limitações — abordadas na **Análise Crítica e Trabalhos Futuros** —, a experiência da Marandu Hub reforça seu potencial estratégico para promover agilidade e eficiência em contextos específicos, servindo como referência para iniciativas semelhantes em espaços colaborativos e *coworkings*.

REFERÊNCIAS

- BOAVENTURA, Megmécia Prazeres. *O trabalho em espaços de coworking: seus aspectos e sentidos para trabalhadores-frequentadores em Salvador-Bahia*. 2022. 72 f. Dissertação (Mestrado em Administração) – Escola de Administração, Universidade Federal da Bahia, Salvador, 2022. Disponível em: <https://repositorio.ufba.br/handle/ri/37077>. Acesso em: 15 maio 2025.
- BROOKE, John. *SUS: a “quick and dirty” usability scale*. In: JORDAN, P. W.; et al. *Usability evaluation in industry*. Londres: Taylor & Francis, 1996. p. 189–194. Disponível em: https://digital.ahrq.gov/sites/default/files/docs/survey/systemusabilityscale%2528sus%2529_c omp%255B1%255D.pdf. Acesso em: 14 jul. 2025.
- CARTA CAPITAL. *Coworking: o que você precisa saber sobre o modelo de trabalho*. 2023. Disponível em: <https://www.cartacapital.com.br/do-micro-ao-macro/coworking-o-que-voce-precisa-saber-sobr e-o-modelo-de-trabalho/>. Acesso em: 22 abr. 2025.
- CENSO COWORKING 2024. Woba Blog. [S. l.]: [s. n.], [s. d.]. Disponível em: <https://blog.woba.com.br/censo-coworking>. Acesso em: 26 jul. 2025.
- CLUB COWORKING. *O panorama do mercado de coworking no Brasil para os próximos anos*. 2022. Disponível em: <https://clubcoworking.com.br/mercado-de-coworking/>. Acesso em: 28 abr. 2025.
- CNN BRASIL. Procura por espaços de coworking sobe mais de 90% em todo o mundo, diz pesquisa: Brasil tem alta de 97% na demanda pelo serviço no primeiro trimestre deste ano, em relação ao mesmo período de 2021. Rio de Janeiro, 2022. Disponível em: <https://www.cnnbrasil.com.br/economia/macroeconomia/procura-por-espacos-de-coworking-s obe-mais-de-90-em-todo-o-mundo-diz-pesquisa/>. Acesso em: 5 maio 2025.
- DIEHL, V. S. Diretrizes para a viabilidade de ambientes de coworking. 2024. 111 f. Dissertação (Mestrado em Gestão e Negócios) – Universidade do Vale do Rio dos Sinos – UNISINOS, Porto Alegre, 2024. Disponível em: <https://www.repositorio.jesuita.org.br/handle/UNISINOS/13471>. Acesso em: 20 jun. 2025.
- EMAILJS. EmailJS documentation. [S.l.]: EmailJS, 2025. Disponível em: <https://www.emailjs.com/docs/>. Acesso em: 3 jul. 2025.
- FIREBASE. Firebase documentation. [S.l.]: Firebase, 2025. Disponível em: <https://firebase.google.com/docs>. Acesso em: 29 jun. 2025.
- FLUTTERFLOW. FlutterFlow documentation. [S.l.]: FLUTTERFLOW, 2025. Disponível em: <https://docs.flutterflow.io/>. Acesso em: 29 jun. 2025.
- GANDINI, Alessandro. *The rise of coworking spaces: a literature review*. *Ephemera*, v. 15, n. 1, p. 193–205, 2015. Disponível em: <https://ephemerajournal.org/sites/default/files/pdfs/contribution/15-1gandini.pdf>. Acesso em: 2 maio 2025.

GRAND VIEW RESEARCH. Low-code application development platform market size, share & trends analysis report. [S.l.]: Grand View Research, 2024. Disponível em: <https://www.grandviewresearch.com/industry-analysis/low-code-application-development-platform-market>. Acesso em: 10 maio 2025.

JOTFORM INC. *Jotform: formulários online gratuitos e personalizados*. [S.l.]: Jotform, 2025. Disponível em: <https://www.jotform.com>. Acesso em: 14 jul. 2025.

MEDINA, P. F.; KRAWULSKI, E. *Coworking como modalidade e espaço de trabalho: uma análise bibliométrica*. Florianópolis: Universidade Federal de Santa Catarina, 2016. Disponível em: https://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1516-37172015000200006. Acesso em: 8 maio 2025.

MENDIX. Mendix Documentation. [S. l.]: Mendix, 2025. Disponível em: <https://docs.mendix.com/>. Acesso em: 18 jul. 2025.

MICROSOFT. Power Apps documentation. [S. l.]: Microsoft, 2025. Disponível em: <https://learn.microsoft.com/en-us/power-apps/>. Acesso em: 18 jul. 2025.

MIKI, A. F. C.; FARIAS, K. R. O.; OLIVEIRA, V. M. *Filosofia dos espaços de coworking no Nordeste brasileiro: análise dos elementos operacionais*. Revista Brasileira de Gestão e Desenvolvimento Regional (G&DR), Taubaté, v. 17, n. 3, p. 286-300, set./dez. 2021. Disponível em: <https://www.rbgdr.net/revista/index.php/rbgdr/article/view/6457/1127>. Acesso em: 26 jul. 2025.

MORAES, P. H. M. M.; PINTO, R. S. *Aplicação de ferramentas low-code para melhoria e automação de processos em uma empresa de contabilidade*. 2022. 90 f. Trabalho de Conclusão de Curso (Graduação em Engenharia de Produção) – Universidade Federal do Ceará, Fortaleza, 2022. Disponível em: https://repositorio.ufc.br/bitstream/riufc/65241/1/2022_tcc_pmoraes.pdf. Acesso em: 27 maio 2025.

LOPES, N. F. *Os avanços tecnológicos e os impactos do trabalho remoto nas relações de trabalho*. 2023. 61 f. Trabalho de Conclusão de Curso (Graduação em Ciências Econômicas) – Faculdade de Ciências Econômicas, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2023. Disponível em: <https://lume.ufrgs.br/bitstream/handle/10183/265822/001185901.pdf>. Acesso em: 26 jun. 2025.

NEX COWORKING. *O futuro do trabalho e o papel do coworking*. [S.l.]: Nex Coworking, 2023. Disponível em: <https://nex.work/o-futuro-do-trabalho-e-o-papel-do-coworking/>. Acesso em: 18 abr. 2025.

OUTSYSTEMS. Documentation. [S. l.]: OutSystems, 2025. Disponível em: <https://success.outsystems.com/documentation/>. Acesso em: 18 jul. 2025.

REFINE DEV. *Low code limitations: exploring the risk of vendor lock-in*. [S.l.]: Refine Dev, 2024. Disponível em: <https://refine.dev/blog/low-code-tools/#introduction>. Acesso em: 12 jun. 2025.

RIBEIRO, Á. S. et al. *Açaitour: guia turístico mobile de empreendimentos de lazer e passeio na cidade de Açailândia*. Revista de Empreendedorismo, Gestão e Pequenas Empresas, v. 12, n. 1, p. 88–105, 2023. Disponível em: <https://www.revistas.editoraenterprising.net/index.php/regmpe/article/view/736/945>. Acesso em: 3 jun. 2025.

SILVA, S.; FERREIRA, W. *Adopet – um sistema de gerenciamento de adoção de animais domésticos*. Feira de Inovação, Ciência e Ensino do Instituto Federal Catarinense – FICE, v. 4, n. 1, 2023. Disponível em: <https://publicacoes.ifc.edu.br/index.php/fice/article/view/3875/3212>. Acesso em: 3 jun. 2025.

UNIOFFICE. *O papel dos espaços de coworking na transformação dos modelos de trabalho*. [S.l.]: Unioffice, 2023. Disponível em: <https://unioffice.com.br/blog/o-papel-dos-espacos-de-coworking-na-transformacao-dos-modelos-de-trabalho/>. Acesso em: 25 abr. 2025.

UPSITES. *Low code: a revolução do desenvolvimento de apps em 2025*. [S.l.]: Upsites, 2023. Disponível em: <https://upsites.digital/desenvolvimento-web/low-code-revolucao-desenvolvimento-apps/>. Acesso em: 16 maio 2025.

VERCEL. *Vercel Documentation*. 2025. Disponível em: <https://vercel.com/docs>. Acesso em: 26 jul. 2025.

WHITE LABEL FOX. *How to create an app with the use of FlutterFlow*. [S.l.]: White Label Fox, 2024. Disponível em: <https://whitelabelfox.com/create-app-with-flutter-flow/>. Acesso em: 3 jun. 2025.

APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

Segue o conteúdo integral do Termo de Consentimento Livre e Esclarecido utilizado na pesquisa:

O(a) Sr.(a) está sendo convidado(a) a participar de uma pesquisa acadêmica. Antes de aceitar, é importante que leia atentamente este Termo de Consentimento Livre e Esclarecido (TCLE). Caso tenha qualquer dúvida, procure o pesquisador responsável para esclarecimentos.

1. Sobre a pesquisa

Esta pesquisa integra um Trabalho de Conclusão de Curso (TCC) do curso de Engenharia de Computação da Universidade Estadual do Maranhão (UEMA), com o objetivo de avaliar a usabilidade do sistema de reservas da Marandu Hub, utilizando como instrumento a System Usability Scale (SUS), uma escala internacionalmente reconhecida para medir a experiência de uso de sistemas interativos.

2. Procedimentos da pesquisa

Caso aceite participar, o(a) Sr.(a) responderá a um questionário com 10 afirmações sobre sua experiência de uso do sistema, marcando o grau de concordância para cada uma (de 1 a 5). A resposta leva em média 3 minutos e não exige conhecimento técnico prévio.

3. Sigilo e proteção de dados

Todas as respostas serão anônimas, e nenhuma informação pessoal será solicitada ou associada ao(a) participante. Os dados coletados serão utilizados exclusivamente para fins acadêmicos e armazenados de forma segura até a conclusão do projeto, sendo então devidamente descartados. A plataforma digital utilizada emprega recursos que protegem a privacidade do(a) respondente, inclusive contra rastreamento por IP, cookies ou identificadores.

4. Riscos e benefícios

Esta pesquisa não envolve riscos físicos, psicológicos ou legais aos participantes. Também não há qualquer compensação financeira ou outro tipo de benefício direto. No entanto, indiretamente, sua contribuição poderá beneficiar o aprimoramento de ferramentas tecnológicas utilizadas na Marandu Hub e fortalecer pesquisas voltadas à melhoria da experiência do usuário em sistemas públicos.

5. Voluntariedade e direito de desistência

A participação é completamente voluntária. O(a) Sr.(a) pode se recusar a responder ou interromper sua participação a qualquer momento, sem a necessidade de justificativa e sem qualquer prejuízo.

6. Divulgação dos resultados

Os resultados poderão ser apresentados em ambiente acadêmico, como parte do TCC ou em eventos científicos, sem qualquer identificação dos participantes, assegurando o sigilo e a confidencialidade das informações.

7. Contato para esclarecimentos

Em caso de dúvidas, o(a) Sr.(a) poderá entrar em contato com os responsáveis pela pesquisa:

Victor Emanuel Santos Moura – discente do curso de Engenharia de Computação
(victorem3181@gmail.com)

Prof. Dr. Antônio Fernando Lavareda Jacob Junior – orientador do TCC
(Antôniojunior@professor.uema.br)

APÊNDICE B – QUESTIONÁRIO DE USABILIDADE (SUS)

A seguir, são apresentadas as 10 afirmações utilizadas no instrumento de avaliação de usabilidade, baseado na **System Usability Scale** (Brooke, 1996):

- 1. Eu acho que gostaria de usar esse sistema frequentemente.**
- 2. Eu achei o sistema desnecessariamente complexo.**
- 3. Eu achei o sistema fácil de usar.**
- 4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.**
- 5. Eu acho que as várias funções do sistema estão muito bem integradas.**
- 6. Eu acho que o sistema apresenta muita inconsistência.**
- 7. Eu imagino que a maioria das pessoas aprenderia a usar esse sistema rapidamente.**
- 8. Eu achei o sistema muito complicado de usar.**
- 9. Eu me senti confiante ao usar o sistema.**
- 10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.**

Escala de resposta para cada item: 1 = Discordo totalmente, 2 = Discordo, 3 = Neutro, 4 = Concordo, 5 = Concordo totalmente.