

Reserve+ App: Aplicativo para Reservas de Ambientes

Daniel Santos Prado¹, Luis Paulo da Silva Carvalho¹

¹Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA)
campus de Vitória da Conquista
Av. Sérgio Vieira de Melo, 3150 - Zabelê
Vitória da Conquista, Bahia, Brasil

danprado17.dp@gmail.com, luiscarvalho@ifba.edu.br

Abstract. Context: *The Reserve+ App emerged from the need to adapt the original web version to mobile devices. With the growing use of smartphones, it became evident that responsive and accessible reservation systems were in demand, capable of optimizing the management of spaces such as rooms and laboratories. This project aims to offer an integrated, efficient, and modern solution for reservation management, enhancing accessibility and usability. Goal:* *The primary goal of this project was to develop a mobile version of Reserve+ that maintained integration with the existing web system while meeting the specificities of mobile devices. Method:* *We adopted a evolutionary/iterative development model. This approach allowed for continuous refinement of functionalities and design through constant feedback cycles. Development was carried out using Flutter for the front-end, Strapi for the back-end, and PostgreSQL as the database, integrating modern and robust technologies. Results:* *The Reserve+ App was successfully implemented, resulting in: A cross-platform mobile application with responsive design; Enhanced functionalities, such as status filters, title search, and detailed pop-ups in the calendar; Efficient integration with the existing web system, ensuring real-time synchronization; Positive user feedback regarding the app's usability and design. Conclusions:* *The adaptation to mobile devices not only resolved the limitations of the web version but also expanded the system's reach and usability. Technological choices, such as using Flutter, played a key role in ensuring performance, consistency, and ease of development.*

Resumo. Contexto: *O Reserve+ App surge da necessidade de adaptar a versão web original, desenvolvida previamente, para dispositivos móveis. Com o crescente uso de smartphones, tornou-se evidente a demanda por sistemas de reservas responsivos e acessíveis, capazes de otimizar a gestão de espaços como salas e laboratórios. Este trabalho visa oferecer uma solução integrada, eficiente e moderna para a visualização de reservas, promovendo maior acessibilidade e usabilidade. Objetivo:* *O objetivo principal deste projeto foi desenvolver uma versão móvel do Reserve+ que mantivesse a integração com o sistema web existente, enquanto atendia às especificidades dos dispositivos móveis. Método:* *O método utilizado foi o modelo de desenvolvimento evolutivo/iterativo. Esta abordagem permitiu o refinamento contínuo das funcionalidades e do design, com ciclos de feedback constantes. O desenvolvimento foi realizado utilizando*

*Flutter para o front-end, Strapi para o back-end e PostgreSQL como banco de dados, integrando tecnologias modernas e robustas. **Resultados:** O Reserve+ App foi implementado com sucesso, resultando em: Um aplicativo móvel multi-plataforma com design responsivo; Funcionalidades aprimoradas, como filtros de status, busca por título e pop-ups detalhados no calendário; Integração eficiente com o sistema web existente, garantindo sincronização em tempo real; Feedback positivo do usuário quanto à usabilidade e ao design do aplicativo. **Conclusões:** A adaptação para dispositivos móveis não apenas solucionou limitações da versão web, mas também ampliou o alcance e a praticidade do sistema. As escolhas tecnológicas, como o uso de Flutter, desempenharam papel fundamental ao garantir desempenho, uniformidade e facilidade de desenvolvimento.*

1. Introdução

A gestão de reservas é crucial para otimizar o uso de recursos em instituições. Segundo [Konia 2024], “Sistemas de reservas de espaços de trabalho surgem como soluções cruciais, permitindo que as empresas otimizem a utilização dos recursos físicos de maneira estratégica. A adoção dessas tecnologias não somente facilita a gestão do espaço físico, mas também incentiva uma cultura de trabalho mais flexível e adaptativa”. Um artigo sobre a gestão de áreas públicas ociosas em São Paulo enfatiza que “a gestão eficiente dos espaços públicos é fundamental para evitar desperdícios e promover o uso adequado dos recursos disponíveis” [Silber 2020]. Este ponto reforça a necessidade de um sistema eficaz para visualização de reservas e utilização de espaços em instituições.

Neste contexto, o Reserve+ App tem o intuito de ser um aplicativo móvel de acompanhamento de reservas de ambientes, o aplicativo irá oferecer uma solução completa para garantir maior flexibilidade, agilidade e eficiência no processo de reserva de salas e outros espaços.

O Reserve+ App se baseia no trabalho desenvolvido por [Cotrim 2023]. [Cotrim 2023] desenvolveu a versão web do Reserve+. Neste novo escopo de trabalho, uma versão voltada ao uso por dispositivos móveis foi criada. Isto tem sentido, pois as tendências tecnológicas atuais, como o aumento do uso de dispositivos móveis, impactam diretamente o desenvolvimento de sistemas [SALES et al. 2023].

Conforme [Zendesk 2024], “Acompanhar as tendências da tecnologia da informação nas empresas, mais do que uma forma de ficar por dentro das novidades, é uma estratégia vital para quem quer manter seu negócio atualizado com o que há de mais moderno no mercado”. Isto se encontra refletido nos motivos que levaram ao desenvolvimento deste aplicativo. O primeiro motivo está no fato que as pessoas atualmente estão utilizando mais dispositivos móveis para navegarem e acessarem serviços e softwares do que computadores desktop, o que fica evidenciado no gráfico da Figura 1.

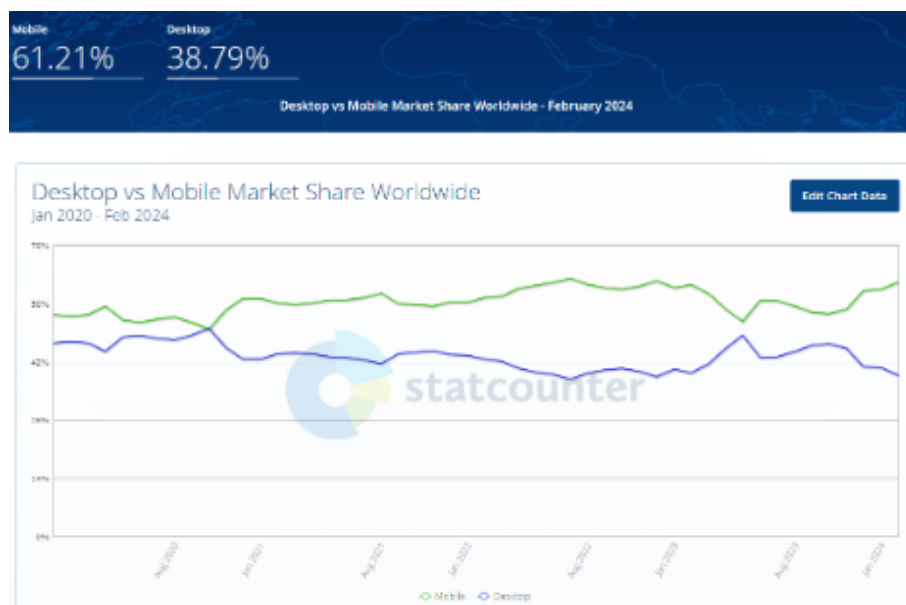


Figura 1. Preferência por navegação através de dispositivos móveis (em verde)¹

O segundo motivo diz respeito a algumas funcionalidades da versão web que não funcionam adequadamente em telas de dispositivos móveis. Exemplo: o calendário de reserva (exibido na Figura 2) que não foi desenvolvido para se comportar de forma responsiva em telas menores, tais como, por exemplo, em dispositivos móveis (celulares e tablets). Embora a versão web do calendário seja responsiva, o sua responsividade não é suficiente para alguns tamanhos de telas de dispositivos móveis.

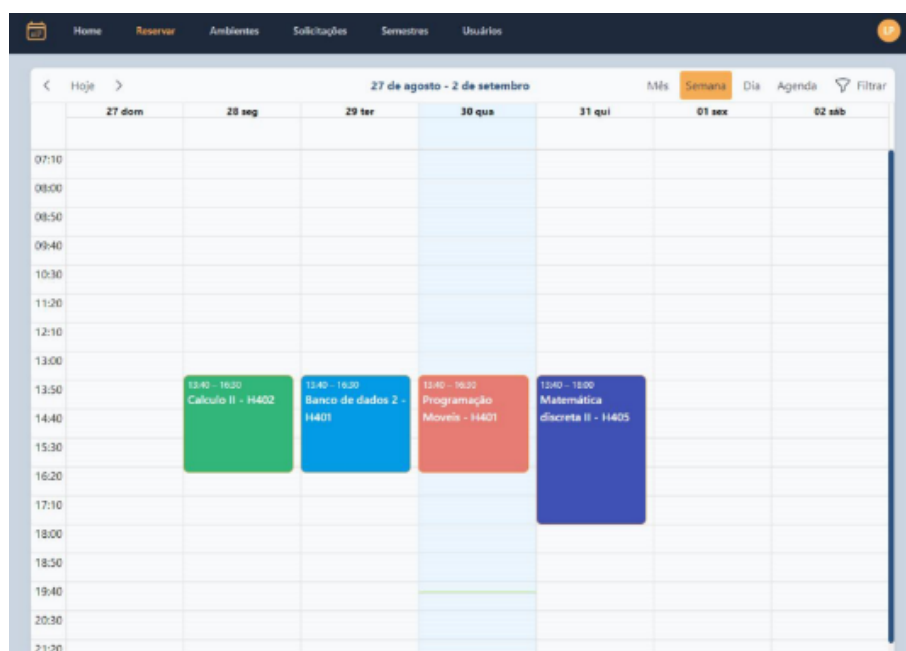


Figura 2. Calendário de Reserva com responsividade limitada [Cotrim 2023]

¹<https://gs.statcounter.com/platform-market-share/desktop-mobile/worldwide/#monthly-202001-202402>

Resumidamente, deseja-se com o Reserve+ App resolver alguns problemas identificados na versão web da solução: (i) incrementar a arquitetura original do Reserve+ para permitir a criação de sistemas de reservas de espaços (salas e laboratórios) acessíveis por dispositivos móveis; (ii) disponibilizar uma versão móvel operacional, com código-fonte aberto e disponível, de um aplicativo android que possa ajudar a visualizar as reservas de forma integrada com o back-end da versão web original.

2. Metodologia

O método utilizado neste trabalho foi o modelo evolutivo/iterativo, que consiste em uma abordagem mais flexível e interativa do desenvolvimento de software, permitindo que as etapas sejam executadas em ciclos, com feedback contínuo. As etapas incluíram: concepção, elaboração e construção de um modelo, na forma de um protótipo de telas, que foi utilizado para capturar e discutir requisitos com um usuário em potencial que já lida constantemente com ações relacionadas à reserva de ambientes no âmbito do IFBA, campus Vitória da Conquista [Appolinário 2006, Cervo 1972]. Para o desenvolvimento do sistema, foram utilizadas as seguintes tecnologias:

- Banco de dados: PostgreSQL;
- Frameworks: Flutter, NextJS e NodeJS;
- Front-end: Dart (linguagem base do Flutter);
- Back-end: Javascript e Typescript.

Para o desenvolvimento do Reserve+ App, a infraestrutura foi configurada em um ambiente local, com a utilização de contêineres Docker para gerenciar os serviços essenciais. O back-end, implementado com Strapi, foi executado em um contêiner Docker, enquanto o PostgreSQL, utilizado como banco de dados, também foi disponibilizado por meio de um contêiner dedicado. Com essa infraestrutura, foi possível garantir segurança no desenvolvimento e um ambiente de trabalho controlado, contribuindo para a agilidade e eficiência das iterações durante a construção do aplicativo.

Como pode ser observado na Figura 3, o desenvolvimento do sistema começou com a adaptação do back-end do Reserve+ web para que fosse utilizado também pela versão móvel, o que corresponde à atividade 1 da figura. Em seguida, no passo 2, foi feita a captura dos requisitos do sistema, que envolveu a identificação e adaptação das funcionalidades do Reserve+ Web em relação a visualização de reservas de ambientes. No passo 3, foi definida a arquitetura do sistema garantindo assim a definição das tecnologias utilizadas e organização do código em módulos funcionais. No passo 4, foi definido o layout da aplicação mobile (esquema de cores, fontes, fluxos de tela). Durante o passo 5, foi criado um repositório no GitHub para gerenciar o código-fonte do sistema. No passo 6, foi desenvolvido o sistema. No passo 7, para garantir que o sistema atendeu às necessidades do usuário, ele foi validado a partir de exibições e entrevistas junto a um usuário potencial. O trabalho foi concluído com a documentação de todo o processo e dos resultados obtidos na forma deste artigo no passo 8.

3. Fundamentação Teórica e Prática

Nesta seção são elencados e explicados os principais conceitos teóricos e elementos práticos que foram usados na criação do Reserve+ App.

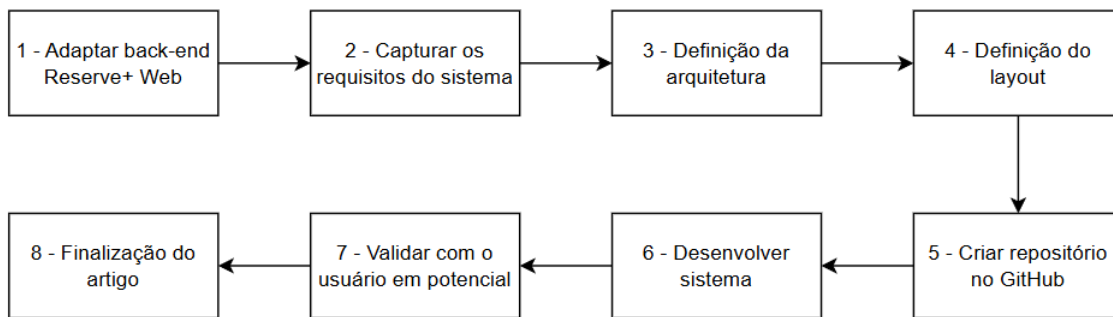


Figura 3. Fluxograma do projeto

3.1. Aplicações Móveis com Flutter

Atualmente, o Flutter é amplamente reconhecido como o *framework* de desenvolvimento de aplicativos móveis multiplataforma popular. Em 2022, uma pesquisa revelou que 46% dos desenvolvedores de *software* preferem o Flutter, tornando-o o *framework* mais utilizado nessa categoria [Dziurzyński 2023]. Esse aumento significativo na adoção reflete o crescimento contínuo e a aceitação do Flutter na comunidade de desenvolvimento. De acordo com os pesquisadores, “a participação de mercado do Flutter teve uma tendência de alta notável, refletida no número de aplicativos desenvolvidos usando esta estrutura. Em meados de 2023, mais de 1 milhão de aplicativos na Play Store foram desenvolvidos com Flutter. . . Em 2022 46% dos desenvolvedores optaram pelo Flutter”.

Além disso, conforme pode ser visto na Figura 4, o número de desenvolvedores que utilizam o Flutter ultrapassou a marca de 2 milhões em 2022, um aumento substancial em relação aos anos anteriores: “em 2021, o Flutter ultrapassou o React Native, tornando-se o framework multiplataforma mais usado. Essa mudança resalta uma preferência notável na comunidade de desenvolvedores, ilustrando a crescente aceitação e confiabilidade do Flutter para o desenvolvimento de aplicativos multiplataforma” [Program-Ace 2023].

Esses dados destacam a importância do Flutter no desenvolvimento de aplicativos móveis, principalmente por sua capacidade de permitir a criação de aplicativos com uma única base de código para múltiplas plataformas, reduzindo o tempo e os custos de desenvolvimento [Sattar et al. 2023]. Além disso, o Flutter possui outras características importantes que o destacam como um dos frameworks mais populares para desenvolvimento de aplicativos móveis multiplataforma. Aqui estão algumas das principais características [Lopes 2023, Sacramento 2022, Sattar et al. 2023]:

- **Desempenho Próximo ao Nativo:** o Flutter oferece desempenho elevado, similar ao de aplicativos nativos, devido à sua capacidade de compilar diretamente para código nativo usando a linguagem Dart e compilação *Ahead-of-Time* (AOT), o que resulta em tempos de resposta mais rápidos e uma experiência de usuário fluida;
- **Recarga Quente (*Hot Reload*):** uma das funcionalidades mais apreciadas pelos desenvolvedores é o *Hot Reload*, que permite refletir as mudanças feitas no código instantaneamente, sem a necessidade de reiniciar o aplicativo;
- **Consistência de Interface de Usuário:** Flutter possui seu próprio mecanismo de renderização, o que garante que a interface do usuário seja consistente em diferentes plataformas (Android, iOS, *Web* e *Desktop*);

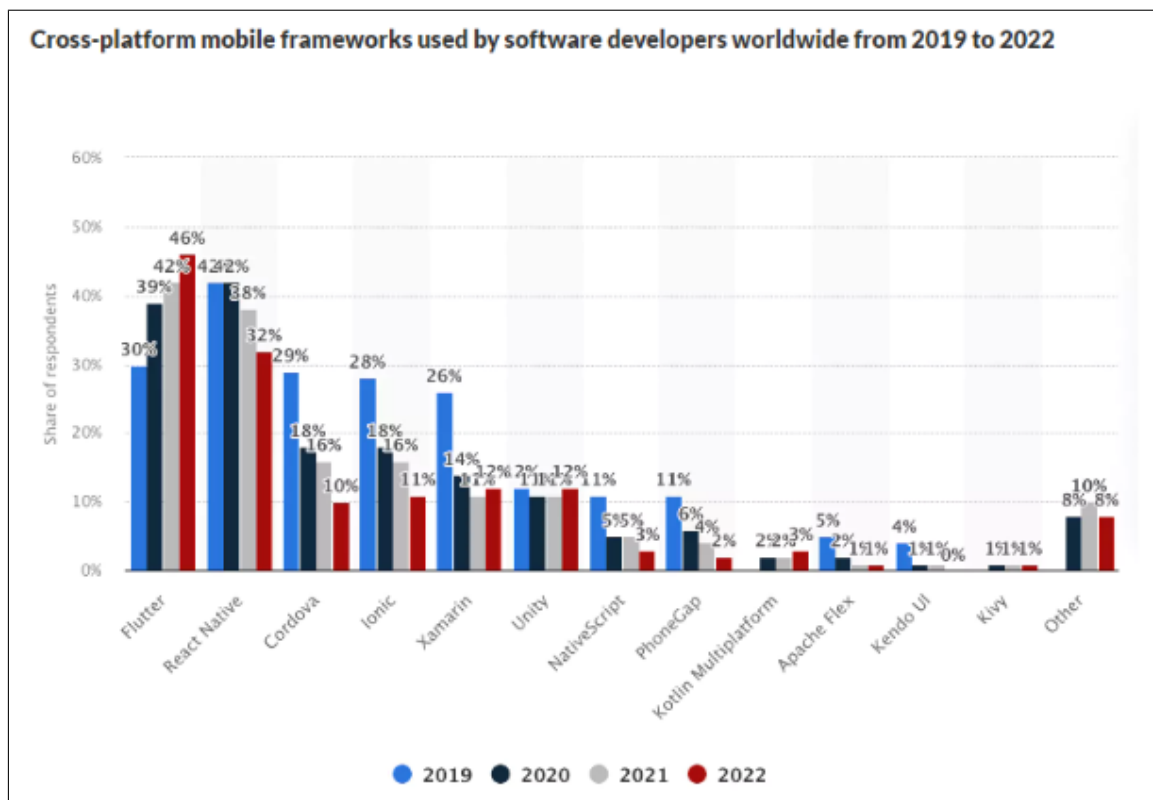


Figura 4. Frameworks mobile usados por desenvolvedores de software em todo o mundo de 2019 a 2022. [Program-Ace 2023]

- Biblioteca de *Widgets* Extensa: Flutter oferece uma vasta coleção de *widgets* personalizáveis que permitem criar interfaces de usuário complexas e atraentes de maneira eficiente. Os *widgets* são a base da construção de interfaces de usuário no Flutter. Eles representam todos os elementos visíveis em um aplicativo, como botões, campos de texto, imagens e *layouts*;
- Comunidade e Suporte Extensivo: O Flutter possui uma vasta quantidade de recursos, tutoriais, plugins e suporte disponíveis online para ajudar os desenvolvedores em todas as etapas do processo de desenvolvimento.

3.2. Reuso da Arquitetura do Reserve+ Web

O Reserve+ App reutilizou o back-end do Reserve+ Web, pois o intuito da versão *mobile* é facilitar o acesso a visualização das reservas agendadas através do Reserve+ Web.

A arquitetura do Reserve+ Web [Cotrim 2023] se encontra ilustrada na Figura 5. Nos parágrafos seguintes a arquitetura se encontra detalhada.

O front-end, que não foi (re)utilizado neste projeto, sendo substituído pela sua versão em Flutter, contém:

- React: Uma biblioteca JavaScript popular para construir interfaces de usuário. React é conhecida por sua eficiência em criar componentes reutilizáveis, facilitando o desenvolvimento de aplicações web complexas;
- Next.js: Um *framework* React que permite a renderização do lado do servidor (SSR) e a geração de sites estáticos (SSG). Next.js é utilizado para melhorar o



Figura 5. Arquitetura do Reserve+ Web [Cotrim 2023].

desempenho e SEO das aplicações React;

- Zustand: Uma pequena biblioteca de gerenciamento de estado para React, oferecendo uma API simples e intuitiva para gerenciar estados globais em aplicações React;
- Typescript: Um superconjunto de JavaScript que adiciona tipos estáticos. TypeScript melhora a robustez do código, ajudando a prevenir erros durante o desenvolvimento.

O back-end, que foi reutilizado pelo novo front-end, é composto de:

- Strapi: Um sistema de gerenciamento de conteúdo (CMS), construído com Node.js. Strapi permite criar e gerenciar APIs rapidamente, fornecendo uma interface amigável para a administração de conteúdo;
- Postgres (PostgreSQL): Um sistema de gerenciamento de banco de dados relacional de código aberto, conhecido por sua robustez e conformidade com os padrões SQL. PostgreSQL é usado para armazenar e gerenciar os dados do aplicativo.

4. Prototipando o Reserve+ App

Antes da realização ou implementação do Reserve+ App, foi realizada uma prototipação das suas telas. O objetivo era permitir a discussão do protótipo com um usuário em potencial do aplicativo, o mesmo que foi envolvido na criação da versão web.

As figuras seguintes ilustram os protótipos das telas que foram apresentadas para um usuário, com bastante experiência com controle de reservas de ambientes do IFBA, campus Vitória da Conquista. A maioria das telas foram aceitas como foram inicialmente prototipadas, mas foram solicitadas algumas mudanças, que foram prontamente aceitas e já se encontram refletidas nas telas do protótipo aqui apresentadas.

A Figura 6 apresenta a tela de acesso ao sistema, onde o usuário deve informar o e-mail e a senha cadastrados. Após a inserção desses dados, o Reserve+ App realizará a validação no back-end, autorizando ou recusando o acesso ao sistema conforme a autenticidade das informações fornecidas.

A Figura 7 representa a segunda tela que será apresentada ao usuário após a validação de seus dados de acesso. Essa tela possui um calendário que contém os registros de reservas realizados pelos usuários em seus respectivos dias.



Figura 6. Tela de Login.

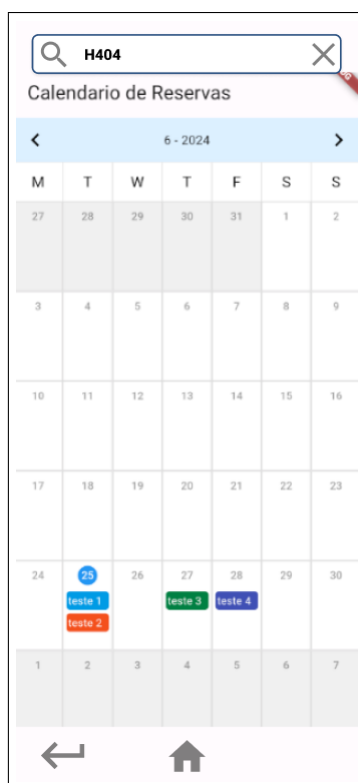


Figura 7. Tela do calendário de reservas.

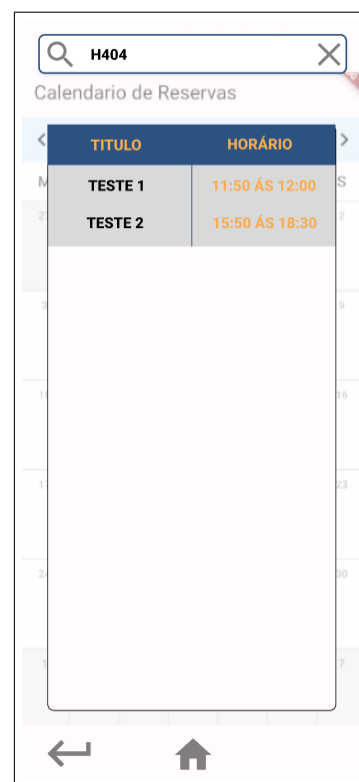


Figura 8. Tela do Pop-up ao clicar em um dia do calendário.

A Figura 8 apresenta um pop-up que será exibido quando o usuário selecionar um dia no calendário mostrado na Figura 7. Esse pop-up exibirá todas as reservas realizadas para o dia selecionado, incluindo os respectivos horários.

A Figura 9 mostra a tela de solicitações. Essa tela exibe todas as solicitações de reservas feitas pelo usuário logado, indicando quais são as solicitações e o respectivo status, que pode ser: Aprovado, Reprovado ou Pendente.

A Figura 10 apresenta a tela de detalhes que será exibida quando o usuário selecionar o ícone de olho em uma solicitação de reserva na tela de solicitações. Essa tela exibirá as principais informações da solicitação de reserva selecionada.

5. Desenvolvendo o Reserve+ App

O desenvolvimento do Reserve+ App visou criar uma solução móvel eficiente e responsiva, baseada na versão web previamente desenvolvida e nos protótipos de tela apresentados na Seção 4. Para atingir esse objetivo, o processo seguiu o modelo de desenvolvimento evolutivo/iterativo, amplamente utilizado em projetos de software que demandam flexibilidade e ciclos curtos de feedback. Esse modelo foi escolhido por sua capacidade de permitir ajustes contínuos e refinamento do produto ao longo das etapas. A seguir, destacam-se as principais fases:

- Adaptação do Back-End: O back-end, desenvolvido com Strapi e PostgreSQL na versão web do Reserve+, foi reutilizado com adaptações para atender às demandas

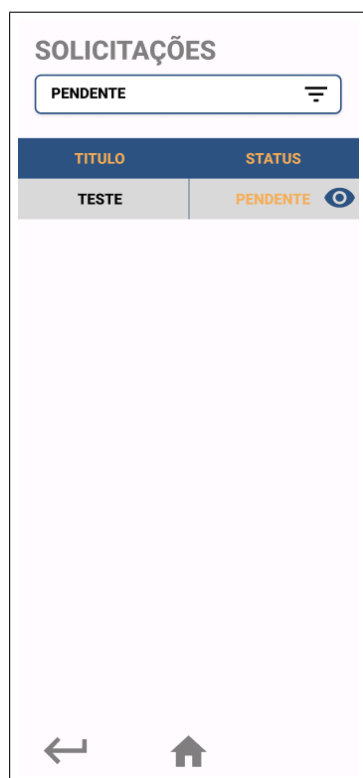


Figura 9. Tela de solicitações.



Figura 10. Tela de detalhes.

específicas da versão móvel. Esse reaproveitamento permitiu uma integração perfeita entre as plataformas, reduzindo custos de desenvolvimento e minimizando riscos de inconsistências nos dados;

- **Levantamento de Requisitos:** Com base na análise de usabilidade da versão web, foram identificados requisitos específicos para a versão móvel. Entre eles, destacaram-se a necessidade de uma interface responsiva, acessível e compatível com dispositivos móveis, além de recursos aprimorados para interação do usuário, como filtros de status de reservas e buscas por título;
- **Definição da Arquitetura e Tecnologias:** A arquitetura foi projetada para ser modular e extensível, utilizando Flutter como framework principal para o front-end. O uso do Flutter permitiu o desenvolvimento multiplataforma, garantindo uma interface uniforme para o Android. Além disso, foram implementadas práticas modernas de organização de código para facilitar a manutenção e futuras expansões do aplicativo;
- **Prototipação e Design:** Prototipações foram desenvolvidas para visualizar e validar o fluxo das telas, incluindo login, calendário, pop-up de eventos, solicitações e detalhes de reservas. Feedbacks de um usuário em potencial foram incorporados para melhorar a experiência do usuário e adequar o design às expectativas;
- **Implementação das Funcionalidades:** O componente `CalendarView`², do Flutter, foi integrado para exibir e gerenciar eventos no formato de calendário. Essa escolha solucionou o problema de falta de responsividade presente na versão web.

²https://pub.dev/packages/calendar_view/

Esse componente foi fundamental para a aplicação pois resolveu o problema de responsividade do calendário da aplicação Web (conforme explicado na Seção 1). Além disso, foram adicionadas funcionalidades como filtros de status e busca por título diretamente no calendário, oferecendo maior controle e personalização ao usuário.

Durante o desenvolvimento do aplicativo Reserve+, foi necessário criar um método específico para recuperar as reservas associadas ao usuário autenticado. O método `getReservationsByUser` foi implementado na Figura 11, o serviço responsável pelas requisições da API, permitindo que o aplicativo filtrasse as reservas de acordo com o usuário logado.

```
Future<ReservationResponse> getReservationsByUser(int userId) async {
  final url = Uri.parse(
    "${URL_RESERVATIONS.toString()}?populate=%2A&filters[requester][id][$eq]=$userId&pagination[page]=1&pagination[pageSize]=999",
  );

  final response = await http.get(url);

  if (response.statusCode == 200) {
    final jsonResponse = jsonDecode(response.body);
    return ReservationResponse.fromJson(jsonResponse);
  } else {
    throw Exception('Falha ao carregar reservas do usuário');
  }
}
```

Figura 11. Método `getReservationsByUser`

6. Resultado Final

O Reserve+ App foi finalizado com sucesso, com o seu código-fonte contendo todos os recursos de widgets e gerenciamento do fluxo de telas do Flutter de forma a corresponder ao que havia sido planejada e discutido na fase de prototipação. A seguir, apresentam-se as telas e funcionalidades principais do aplicativo.

A tela de login, na Figura 12, permite que o usuário insira suas credenciais para acessar o sistema. Com um design limpo e intuitivo, essa tela prioriza a experiência do usuário, facilitando a navegação desde o primeiro uso.

O calendário é o centro do aplicativo, exibindo todas as reservas de forma clara e organizada. O usuário pode navegar entre os meses, visualizar os dias com reservas marcadas. O resultado final, com o uso do widget, `CalendarView`, se encontra ilustrado na Figura 13.

Ao selecionar um dia no calendário, o aplicativo exibe um pop-up com os eventos daquele dia (Figura 14), mostrando o título e o horário de cada reserva. Essa funcionalidade organiza a visualização de informações e melhora a experiência do usuário.

Na tela de solicitações, exibida na Figura 15, o usuário pode visualizar todas as reservas feitas por ele, separadas por status. A interface inclui um filtro para selecionar apenas um status por vez, permitindo um controle mais detalhado sobre as solicitações. Cada reserva conta com um ícone de visualização que leva à tela de detalhes.

Na tela de detalhes, ilustrada na Figura 16, o usuário pode visualizar todas as informações relevantes sobre uma reserva específica, incluindo sala, data de início e fim,

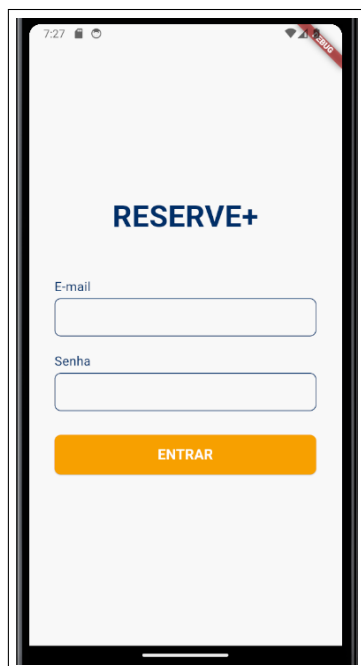


Figura 12. Tela de Login do Reserve+ App.

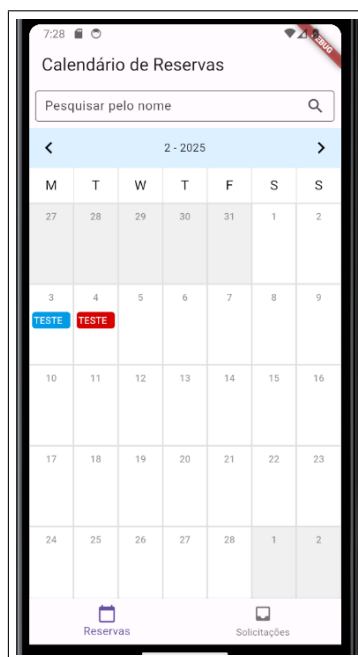


Figura 13. Tela com o calendário de reservas.

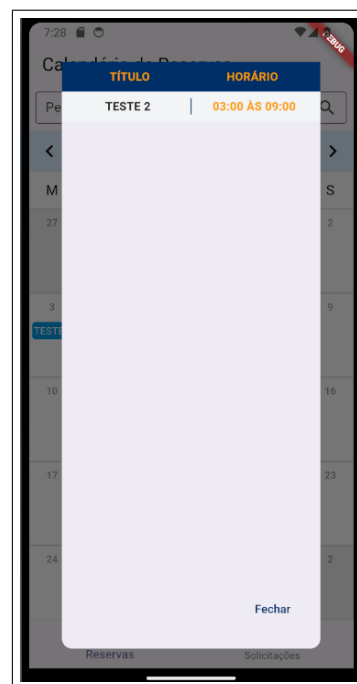


Figura 14. Pop-up com eventos do dia selecionado.

horário, status e o solicitante. Essa tela foi projetada para exibir os dados de forma clara e direta, com destaque para o status, que possui cores específicas para fácil identificação.

O Reserve+ App foi desenvolvido com foco na responsividade, garantindo uma experiência consistente em dispositivos Android. A integração com o back-end, utilizando Strapi e PostgreSQL, foi realizada de forma eficiente, permitindo que os dados fossem sincronizados em tempo real.

O link da Figura 17 contém um vídeo³ que demonstra o uso do Reserve+ App, exibindo dados de reservas adicionados através do Reserve+ Web.

6.1. Trabalhos Relacionados

O trabalho de Igor Cotrim Santos [Cotrim 2023], intitulado “Desenvolvendo uma Aplicação Web para o Gerenciamento de Reservas de Ambientes”, também realizado no Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA), foi a base para o desenvolvimento do Reserve+ App. A proposta de Igor focou na criação de um sistema web para atender às diversas demandas de instituições educacionais, permitindo a reserva eficiente de salas e laboratórios. Porém a sua principal funcionalidade, a visualização das reservas, é realizada através de um calendário que não possui responsividade para dispositivos móveis.

O Room Booking System⁴ da NFS Technology é uma solução para gerenciamento de reservas em ambientes corporativos. Semelhante ao Reserve+, ele oferece uma inter-

³https://www.youtube.com/watch?v=1JdL_BWoDeE

⁴<https://www.roombookingsystem.co.uk/>



Figura 15. Tela de solicitações do usuário.



Figura 16. Tela de detalhes da reserva



Figura 17. Demonstração de uso do Reserve+ App

face web e móvel, permitindo reservas em tempo real e notificações automáticas. Apesar de sua robustez e funcionalidades avançadas, ele pode ser complexo para pequenas empresas, destacando-se pelo custo e pela necessidade de uma infraestrutura maior para sua plena utilização.

O Skedda⁵ é uma plataforma de gestão de espaços que se destaca pela sua interface amigável e facilidade de uso. Com integrações a calendários como Google Calendar e Outlook, Skedda permite uma gestão de reservas simplificada e acessível. No entanto, muitas funcionalidades avançadas são pagas, o que pode limitar sua atratividade para organizações com orçamento restrito, contrastando com a proposta do Reserve+.

Por fim, o Microsoft Bookings⁶ é uma ferramenta integrada ao Microsoft 365,

⁵<https://www.skedda.com/>

⁶<https://www.microsoft.com/pt-br/microsoft-365/business/scheduling-and-booking-app/>

facilitando a reserva de espaços e a gestão de compromissos com uma integração nativa às outras ferramentas da Microsoft. Embora seja uma solução poderosa, pode ser excessiva para usuários que não estão completamente integrados ao ecossistema Microsoft, o que pode representar uma barreira para adoção em certas organizações.

Comparativamente, juntamente com o Reserve+ App (e também o Reserve+ Web) se encontra disponibilizado o código-fonte abertamente para quem desejar adaptar ou modificar o sistema conforme necessidades específicas. Qualquer instituição ou usuário poderá personalizar a aplicação, adicionando funcionalidades ou ajustando o layout de acordo com as demandas de seus ambientes e dispositivos. Essa abertura visa incentivar a colaboração e o aprimoramento contínuo, permitindo que a comunidade possa, eventualmente, contribuir para o desenvolvimento de uma solução cada vez mais robusta.

7. Conclusões

O desenvolvimento do Reserve+ App representou um avanço na adaptação da solução web original para um ambiente móvel, atendendo a necessidades de acessibilidade em dispositivos móveis. Durante o processo, foram enfrentados diversos desafios técnicos que exigiram a aplicação de soluções criativas e tecnológicas específicas.

Um dos desafios mais relevantes foi a adaptação do sistema de calendário para dispositivos móveis. Na versão web, o calendário não era responsivo, o que dificultava sua utilização em telas menores. Para resolver essa limitação, foi necessária a integração do componente `CalendarView`, que permitiu uma visualização eficiente dos eventos, mesmo em dispositivos com diferentes tamanhos de tela.

Outro obstáculo foi a integração do back-end existente, desenvolvido em Strapi e PostgreSQL, que apresentou desafios relacionados à compatibilidade de dados e à sincronização em tempo real. Esses problemas foram superados com adaptações específicas na API e na lógica de negócios, garantindo uma comunicação eficiente entre os sistemas.

A reutilização do back-end baseado em Strapi e PostgreSQL demonstrou a eficácia de uma arquitetura modular e escalável. A abordagem permitiu minimizar o esforço de reimplementação, mantendo a integração entre as versões web e móvel do sistema.

As tecnologias escolhidas desempenharam um papel fundamental na superação dos desafios e no sucesso do projeto. O recurso de Hot Reload, característico do Flutter, foi essencial para acelerar o ciclo de desenvolvimento e testar rapidamente alterações de design e funcionalidade.

Por fim, a aplicação do modelo evolutivo/iterativo no ciclo de desenvolvimento garantiu que o feedback contínuo fosse incorporado ao longo das etapas. Isso resultou em um produto final que atendeu não apenas aos requisitos funcionais, mas também às expectativas dos usuários, proporcionando uma solução prática e moderna para o acompanhamento de reservas.

7.1. Trabalhos Futuros

Embora o Reserve+ App tenha alcançado seus objetivos principais, atendendo às necessidades de acompanhamento de reservas de ambientes de forma eficiente, algumas melho-

rias e expansões podem ser exploradas em trabalhos futuros para enriquecer ainda mais o sistema e ampliar suas funcionalidades.

Uma das funcionalidades que poderiam melhorar a experiência do usuário é a implementação de cadastro de reservas diretamente pelo aplicativo. A versão atual do Reserve+ App, entregue ao final deste trabalho, contempla somente a visualização do status de pedidos de reserva, como pode ser visto na Figura 15. Esta funcionalidade está restrita à aplicação web e caberia então um trabalho futuro para replicar este recurso na versão móvel. Esse recurso permitiria que os usuários tenham maior agilidade no agendamento de suas reservas.

Dentro do escopo principal deste trabalho esteve, principalmente, provar que era possível criar uma versão móvel para o Reserve+ Web, replicando as funcionalidades e reutilizando o back-end. Todavia, não foram realizados testes com usuários finais em cenários reais ou próximos do real. Outro trabalho futuro importante, portanto, é avaliar se o Reserve+, Web e móvel, conseguem, de fato, atender usuários interessados em reservar ambientes.

7.2. Contribuições

O desenvolvimento do Reserve+ App gerou diversas contribuições relevantes que podem ser utilizadas como base para trabalhos futuros ou por desenvolvedores interessados em aprimorar ou expandir o sistema.

Todo o código-fonte do Reserve+ App está disponível em um repositório público no GitHub⁷, com documentação detalhada para facilitar sua utilização. O repositório inclui: código do front-end desenvolvido em Flutter; configurações do back-end em Strapi e o banco de dados PostgreSQL e scripts de configuração para inicialização do ambiente de desenvolvimento. A Figura 18 contém um link para acesso ao repositório.



Figura 18. Repositório de código do Reserve+ App

Referências

Appolinário, F. (2006). *Metodologia da ciência: filosofia e prática da pesquisa*. Thomson.

Cervo, A. L. (1972). *Metodologia científica, para uso dos estudantes universitários*.

⁷<https://github.com/reservemais/reserve-mais-mobile>

- Cotrim, I. (2023). Desenvolvendo uma aplicação web para o gerenciamento de reservas de ambientes. Acessado em: 16 out. 2024.
- Dziurzyński, D. (2023). Flutter in 2024: Maximizing roi in cross-platform app development. <https://applover.com/blog/flutter-in-2024-maximizing-roi-in-cross-platform-app-development>. Acessado em: 9 jun. 2024.
- Konia (2024). Transforme seu ambiente de trabalho com sistemas de reservas inteligentes. <https://konia.com.br/transforme-seu-ambiente-de-trabalho-com-sistemas-de-reservas-inteligentes/>. Acessado em: 9 jun. 2024.
- Lopes, J. (2023). Introdução ao flutter: Um framework para desenvolvimento de aplicativos multiplataforma. <https://www.dio.me/en/articles/introducao-ao-flutter-um-framework-para-desenvolvimento-de-aplicativos-multiplataforma>. Acessado em: 9 jun. 2024.
- Program-Ace (2023). Flutter app development insights. <https://program-ace.com/blog/flutter-app-development-insights/>. Acessado em: 9 jun. 2024.
- Sacramento, J. (2022). Flutter para iniciantes: Tudo o que você precisa saber. <https://blog.cubos.academy/flutter-para-iniciantes-tudo-o-que-voce-precisa-saber/>. Acessado em: 9 jun. 2024.
- SALES, D. F. et al. (2023). Desenvolvimento de aplicativos móveis: o mercado das aplicações nativas e híbridas.
- Sattar, A. M., Soni, P., Ranjan, M. K., Kumar, A., Sahu, C., Saxena, S., and Chaudhari, P. (2023). Accelerating cross-platform development with flutter framework.
- Silber, A. (2020). Espaços públicos ociosos. https://issuu.com/amandasilber/docs/espacos-publicos-ociosos_issu-compressed_1_. Acessado em: 9 jun. 2024.
- Zendesk (2024). Tecnologia da informação nas empresas. <https://www.zendesk.com.br/blog/tecnologia-da-informacao-nas-empresas/>. Acessado em: 9 jun. 2024.

APÊNDICE

Os conceitos adquiridos durante o curso de Bacharelado em Sistemas de Informação (BSI) pelo Instituto Federal de Educação, Ciência e Tecnologia da Bahia contribuíram para a construção do projeto, desde o levantamento teórico até a aplicação prática. A Figura 19 demonstra todas as disciplinas que contemplam a ementa do curso de BSI e que se relacionam de forma direta e indireta.

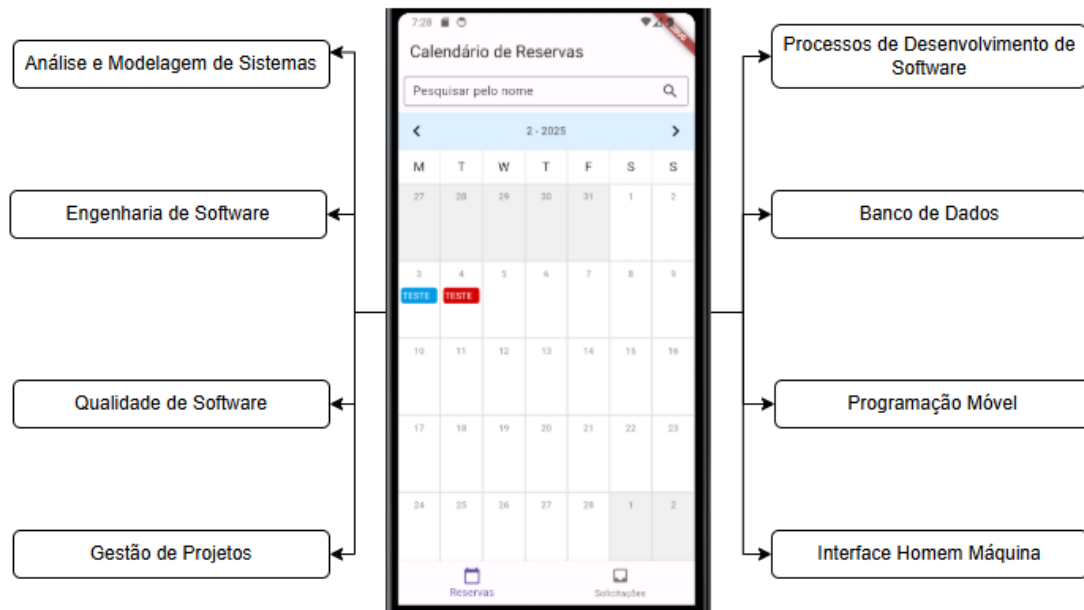


Figura 19. Disciplinas do Curso de BSI relacionadas à Realização do Reserve+ App