

Associação Propagadora Esdeva
Centro Universitário Academia – UniAcademia
Curso de Sistemas de Informação
Trabalho de Conclusão de Curso

Desenvolvimento de aplicação No-code (AppSheet) e React Native: Um estudo de caso

Linha de Pesquisa: Engenharia de Software

Caio Dutra Aperibense Ottoni¹
Centro Universitário Academia, Juiz de Fora, MG

Tassio Ferenzini Martins Sirqueira²
Centro Universitário Academia, Juiz de Fora, MG

RESUMO

[Contexto] Este artigo analisa a possibilidade do desenvolvimento no-code no contexto da criação de uma aplicação para controle de estoque de um museu, destacando suas características e contribuição para a transformação digital. A crescente adoção dessas plataformas tem possibilitado que projetos sejam desenvolvidos sem a necessidade de conhecimento técnico avançado, permitindo a criação de soluções personalizadas, acelerando o processo de digitalização e reduzindo a dependência de equipes especializadas. **[Objetivo]** O objetivo deste estudo é realizar um estudo de caso entre duas abordagens de desenvolvimento: a tecnologia tradicional React Native e a plataforma no-code AppSheet. A partir desse estudo, buscou-se avaliar suas características, aplicabilidade e limitações no contexto do desenvolvimento da aplicação No-code em relação ao React Native. **[Metodologia]** A metodologia empregada incluiu o desenvolvimento de duas soluções de controle de estoque, uma utilizando o AppSheet e outra com React Native. Em seguida, foi realizada uma análise entre as duas plataformas, levando em consideração aspectos como a facilidade de uso, a flexibilidade, os custos envolvidos e o tempo de desenvolvimento. **[Resultados]** O AppSheet foi uma ferramenta que possibilitou um desenvolvimento mais rápido que o React Native e revelou-se uma possibilidade para o desenvolvimento rápido de protótipos e aplicativos simples. **[Considerações]** As plataformas no-code, como o AppSheet, demonstraram ser alternativas ao método tradicional de desenvolvimento em projetos simples. Além disso, podem atuar como ferramentas complementares na prototipação de aplicações, que poderão ser posteriormente desenvolvidas com outras tecnologias.

Palavras-chave: Aplicação Móvel. No-code. React Native.

¹Discente do Curso de Sistemas de Informação do Centro Universitário Academia – UniAcademia.

² Docente do Curso de Engenharia de Software do Centro Universitário Academia. Orientador.

ABSTRACT

[Context] This article analyzes the feasibility of no-code development in the context of creating an inventory management application for a museum, highlighting its features and contribution to digital transformation. The growing adoption of these platforms has enabled projects to be developed without the need for advanced technical knowledge, allowing the creation of customized solutions, accelerating the digitization process, and reducing dependence on specialized teams. **[Objective]** The objective of this study is to conduct a case study comparing two development approaches: traditional React Native technology and the no-code platform AppSheet. From this study, the aim was to evaluate their characteristics, applicability, and limitations in the context of developing a no-code application compared to React Native. **[Methodology]** The methodology employed included the development of two inventory management solutions, one using AppSheet and the other with React Native. Subsequently, an analysis of the two platforms was carried out, considering aspects such as ease of use, flexibility, costs involved, and development time. **[Results]** AppSheet proved to be a tool that enabled faster development than React Native and emerged as a viable option for the rapid development of prototypes and simple applications. **[Considerations]** No-code platforms, such as AppSheet, have demonstrated to be alternatives to the traditional development method for simple projects. Furthermore, they can act as complementary tools in prototyping applications that can later be developed using other technologies.

Keywords: Mobile Application. No-code. React Native.

1 INTRODUÇÃO

Estamos em um mundo cada vez mais digitalizado, onde a tecnologia evolui diariamente, então a busca pela inovação e pelo empreendedorismo focado em tecnologia tem crescido gradativamente. Somado a isso temos ainda a escassez de programadores e profissionais qualificados para a construção de aplicações nos dias atuais. Isso vem ocasionando o aumento na busca por plataformas onde qualquer pessoa possa desenvolver suas ideias e colocá-las em prática, seja para o início da prototipação de um produto comercial, ou simplesmente para a solução de problemas diários (Santos, 2022).

A gestão de estoque é o processo responsável por toda movimentação física de todo o tipo de estoque. Além disso, é de ciência de todas as pessoas que o estoque é um “produto” que tem o seu valor, e devido a isso a gestão de

estoque tem a responsabilidade financeira no sentido de ter o conhecimento do valor do estoque da empresa (Pani; Filho, 2023).

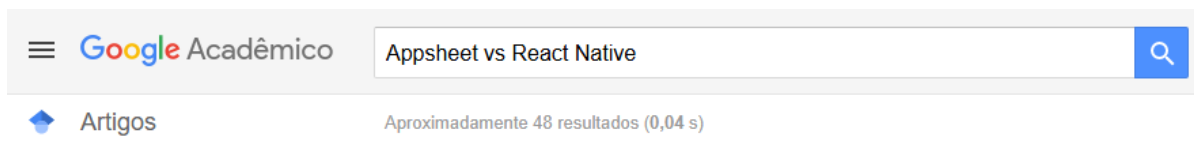
Este trabalho tem como objetivo fazer um estudo entre os métodos de desenvolvimento tradicional e o no-code no contexto do gerenciamento de estoque, utilizando como estudo de caso uma aplicação destinada a um pequeno museu. A pesquisa visa explorar as características de cada abordagem, contribuindo para uma melhor compreensão de suas aplicabilidades.

2 REFERENCIAL TEÓRICO

O Appsheet oferece soluções rápidas e econômicas em comparação ao React Native. Contudo, a versatilidade dos serviços disponíveis no React Native ainda o torna a escolha preferida na indústria. O setor de desenvolvimento de software está mais focado em métodos tradicionais de codificação (React Native, Android, iOS, etc.). Fornecedores de serviços de pagamento e bancos geralmente fornecem documentação para frameworks como React Native, mas não para plataformas como Appsheet. O mesmo ocorre na indústria de jogos, onde fornecedores líderes, como Evolution, apenas oferecem suporte para frameworks tradicionais (Malik, 2021).

Dentro da abordagem de Desenvolvimento Rápido de Aplicações (RAD, do inglês Rapid Application Development), o AppSheet se mostra uma excelente opção. A plataforma minimiza a necessidade de codificação, tornando-o acessível para pessoas com diferentes níveis de habilidade em programação, o que melhora a colaboração dentro das equipes de RAD. Além disso, ele permite uma integração de dados flexível, conectando-se facilmente a diversas fontes de dados, uma característica crucial para projetos RAD com requisitos em constante mudança. Sua interface visual simplifica o desenvolvimento iterativo com base no feedback dos usuários, capacitando os desenvolvedores a criarem interfaces amigáveis e adaptadas às necessidades dos usuários finais (Petrovic, et al., 2024).

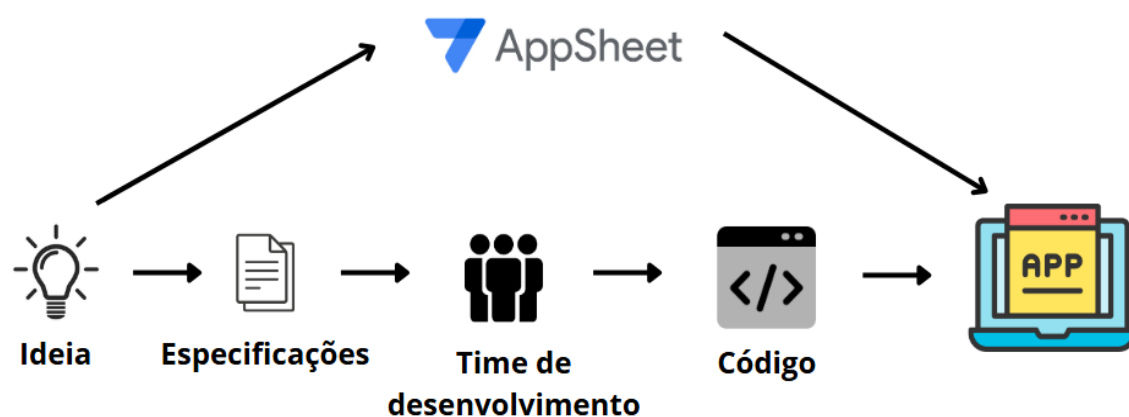
Para a realização da pesquisa, foi feita a busca a seguir apresentada:

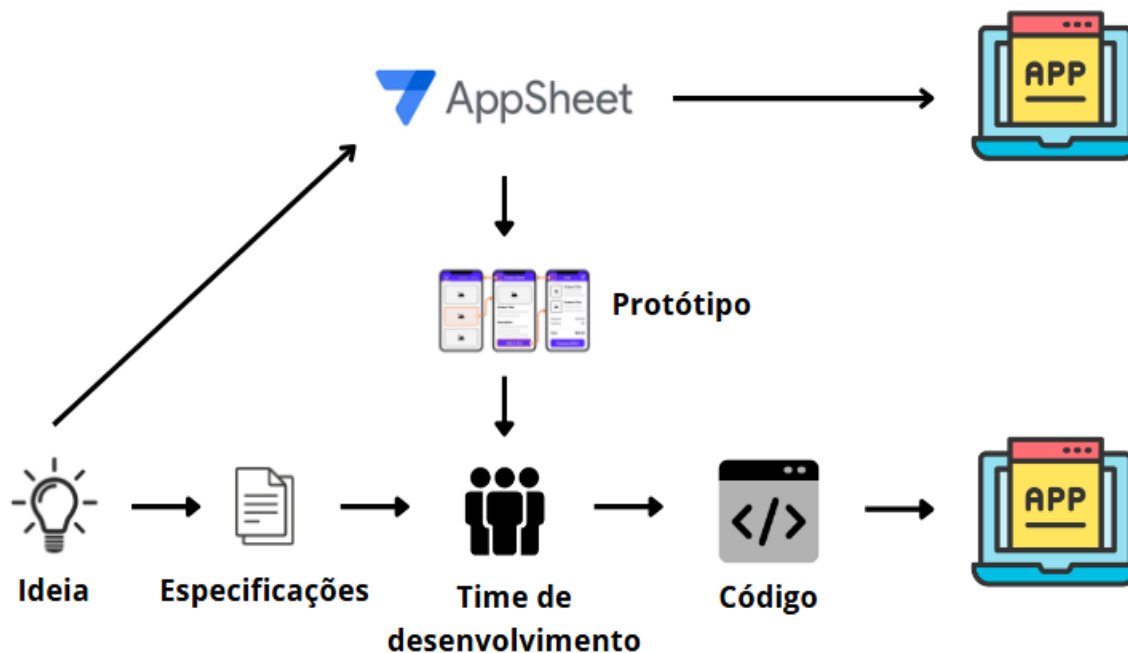


A pesquisa foi realizada utilizando o Google Scholar <<https://scholar.google.com/>> como base de dados devido ao seu acesso gratuito.

A pesquisa inicial resultou em um conjunto de 48 artigos. Dentre esses, apenas três demonstraram pertinência direta ao objetivo do estudo. No presente trabalho, destacou-se como diferencial, em relação aos artigos selecionados, o uso da ferramenta AppSheet em ambiente acadêmico, auxiliando os discentes a desenvolver uma aplicação na plataforma no-code, substituindo a necessidade de desenvolvimento de uma aplicação pelo método tradicional, e evidenciando que a ferramenta também pode ser utilizada no desenvolvimento de protótipos.

Figura 1. Diagramas





Fonte: Elaborada pelo autor.

As ferramentas utilizadas nesse trabalho, plataforma AppSheet, adquirida pela Google em 2020³, destaca-se como uma solução no-code que permite a criação de aplicativos personalizados sem a necessidade de codificação. A plataforma integra-se com diversos serviços populares, como Google Sheets, Excel, Salesforce, entre outros serviços do Google Workspace.

Já o React Native é uma estrutura open-source para desenvolvimento de aplicativos móveis, criada pelo Facebook em 2015⁴. Sendo uma biblioteca para o desenvolvimento móvel cross-platform, é possível aproveitar a mesma base de código para fazer a distribuição da aplicação tanto para sistemas iOS quanto Android, além de outras plataformas, como: Android e Apple TV, Linux, macOS e Windows. No entanto, uma das características que diferencia o React Native de outras ferramentas cross-platform, como o Ionic, é ter a capacidade de gerar aplicações finais com componentes das plataformas nativas, não necessitando

3 Disponível em: <<https://about.appsheet.com/home/>>. Acesso em: 01/12/2024

4 Disponível em: <<https://reactnative.dev/>>. Acesso em 22/10/2024

de uma WebView para poder exibir os elementos de interface do aplicativo (Falcão, 2022).

3 METODOLOGIA

A metodologia empregada neste estudo consiste em uma demonstração prática dos métodos de desenvolvimento de aplicações utilizando as ferramentas AppSheet e React Native, seguida por uma análise comparativa desses tipos de desenvolvimento. Começando pela preparação do ambiente de desenvolvimento para React Native, detalhando todos os passos necessários para começar a criar uma aplicação.

A preparação do ambiente de desenvolvimento para React Native é uma etapa crucial que garante que o desenvolvedor tenha todas as ferramentas e dependências necessárias configuradas corretamente para iniciar o trabalho. O primeiro passo envolve a instalação do Node.js, uma plataforma que permite executar código JavaScript fora de um navegador. O Node.js é essencial para o desenvolvimento com React Native, pois ele gerencia os pacotes necessários e fornece o ambiente de execução para JavaScript, garantindo que todas as dependências do projeto sejam instaladas e atualizadas corretamente. A instalação do Node.js pode ser feita através do site oficial, onde estão disponíveis versões para diferentes sistemas operacionais, como Windows, macOS e Linux⁵.

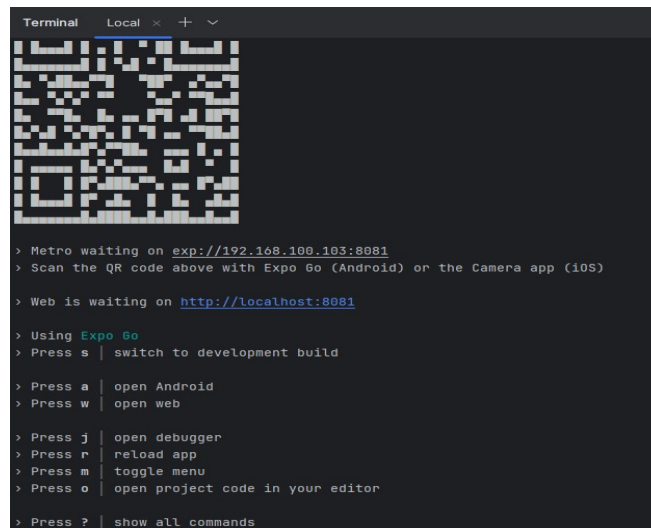
Vale destacar o uso do Expo que traz a possibilidade de utilizar o Snack Expo, uma ferramenta online que permite o desenvolvimento e teste de aplicativos React Native diretamente no navegador. O Snack Expo é ideal para iniciantes ou para quem deseja realizar demonstrações rápidas sem a necessidade de configurar um ambiente local. Ao acessar o site do Snack Expo, o desenvolvedor pode criar um exemplo básico de aplicação, editar o código em tempo real e ver as mudanças refletidas instantaneamente no simulador integrado ⁶. Essa funcionalidade não apenas agiliza o processo de

⁵ Disponível em: <https://nodejs.org/en/about>. Acesso em: 2 nov. 2024.

⁶ Disponível em: <https://docs.expo.dev/more/expo-cli/>. Acesso em: 2 nov. 2024.

desenvolvimento, mas também reduz barreiras técnicas para aqueles que estão começando com React Native.

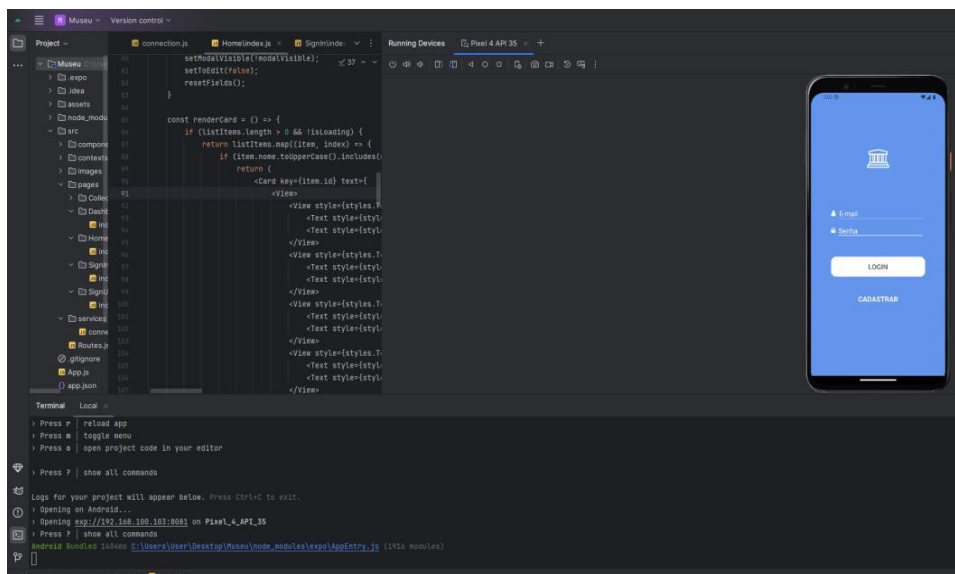
Figura 2. Servidor de desenvolvimento iniciado.



Fonte: Elaborada pelo autor.

A Figura 2 exibe a saída do terminal ao iniciar um projeto Expo. O terminal aguarda conexão com um dispositivo via endereço de rede fornecido, permitindo acesso pelo app Expo Go (escaneando o QR code) ou pelo link <http://localhost:8081> no navegador. Instruções para abrir no Android (a), navegador (w) ou depurar (j) tornam o desenvolvimento mais ágil. Neste trabalho, utilizou-se o Android Studio para visualização em dispositivo Android.

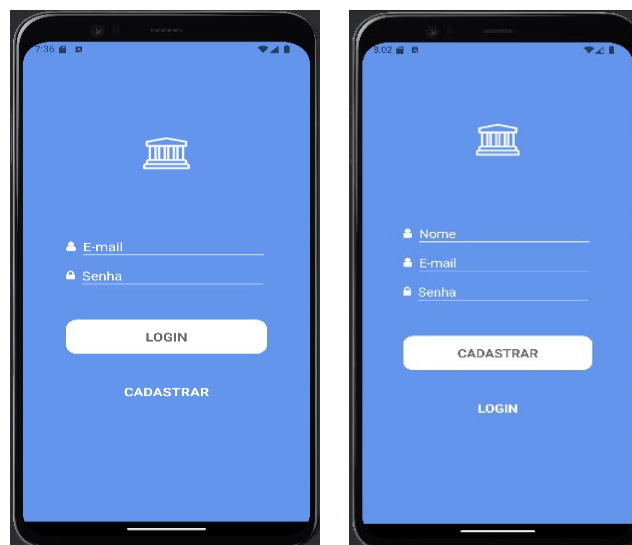
Figura 3. Aplicação rodando no Android Studio



Fonte: Elaborada pelo autor.

A Figura 3 mostra a visualização da aplicação rodando no ambiente Android, com o device Pixel 4 como modelo.

Figura 4. Telas de autenticação



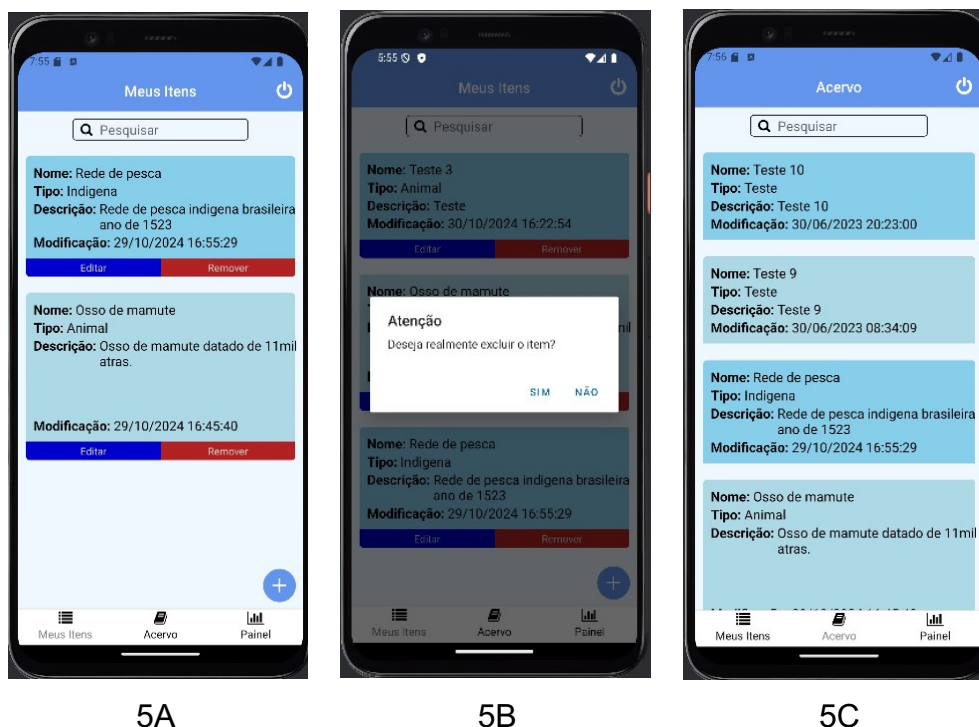
4A

4B

Fonte: Elaborada pelo autor.

Na Figura 4 estão presentes as telas referentes a autenticação. A Figura 4A consiste na tela de validação, onde é necessário informar e-mail, senha e em seguida clicar no botão Login, desse modo, o sistema irá verificar a autenticidade dos dados informados, liberando o acesso ao aplicativo. A Figura 4B apresenta o cadastro de usuário, sendo necessária a inserção das informações solicitadas, após, é preciso clicar no botão Cadastrar e um novo cadastro será realizado. As informações obtidas são armazenadas no banco de dados *FireBase*.

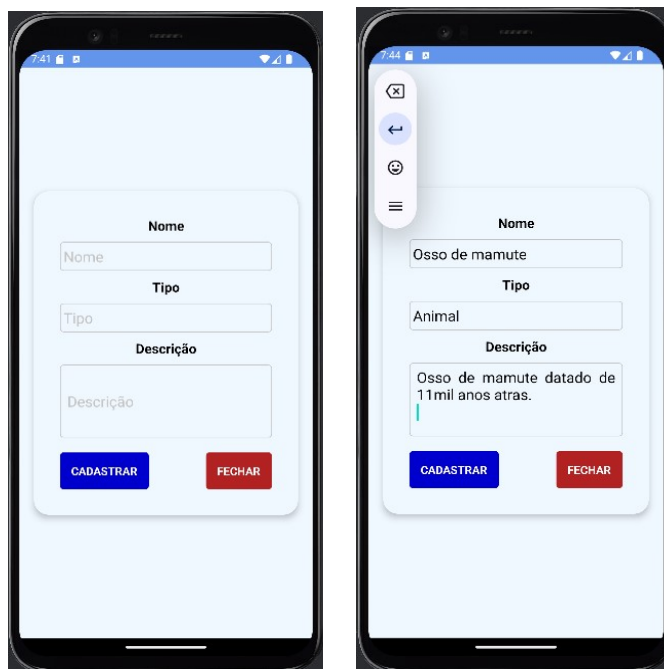
Figura 5. Telas de Itens e objetos adicionados e acervo



Fonte: Elaborada pelo autor.

A Figura 5 apresenta as telas referentes aos itens cadastrados no sistema e aos itens já presentes no acervo. A Figura 5A mostra alguns itens recém-adicionados, a Figura 5B exibe a funcionalidade para apagar ou editar um item adicionado, e a Figura 5C apresenta o histórico do acervo. Na tela de itens podemos notar também que temos um botão azul com sinal de “+” no canto inferior direito para adicionar um novo item com direcionamento para outra tela, e como já estamos logados no sistema, as telas possuem um botão no canto superior direito para fazer o logoff.

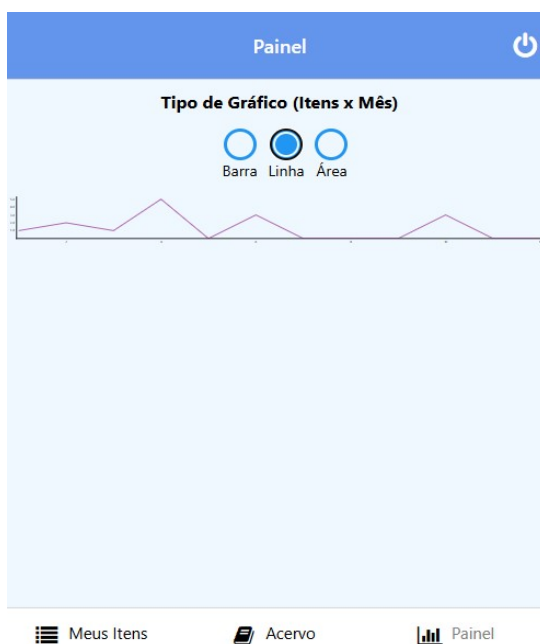
Figura 6. Tela de cadastro



Fonte: Elaborada pelo autor.

A Figura 6 apresenta a tela de cadastro dos itens mencionados na Figura 5. Essa tela é acessada pelo botão de adicionar ou pelo botão de editar, onde são realizadas, respectivamente, as ações de incorporação e edição de itens.

Figura 7. Tela do gráfico



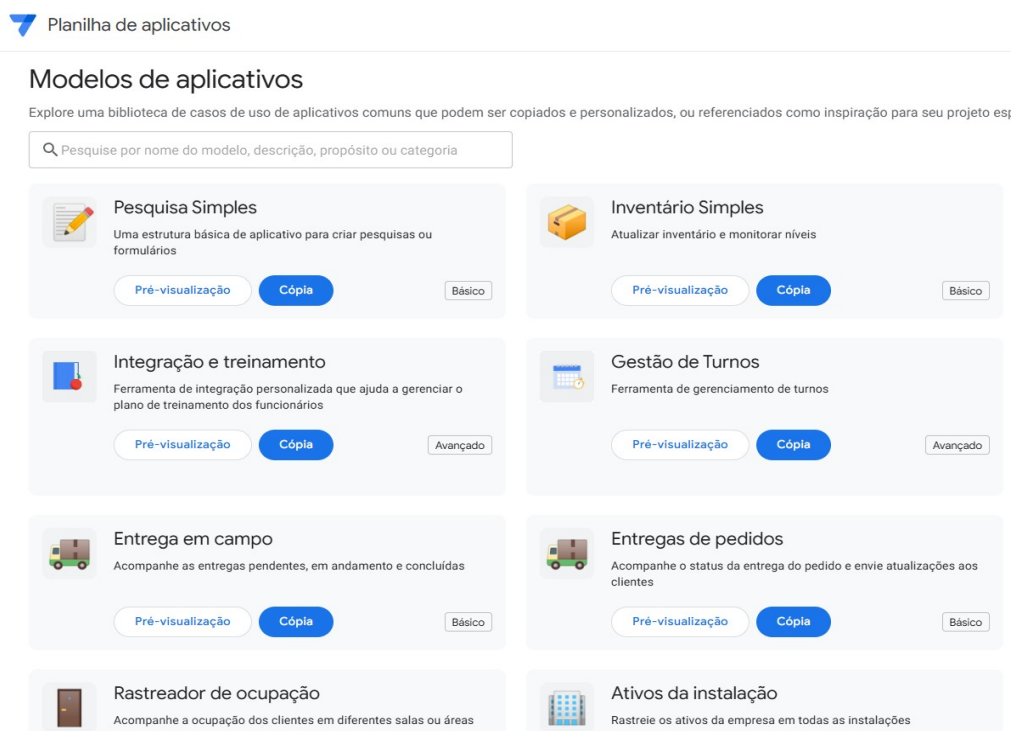
Fonte: Elaborada pelo autor.

A Figura 7 mostra somente a quantidade de itens x mês cadastrados no sistema por meio de três opções de gráficos (barra, linha e área).

Na parte do React Native, visto acima, a preparação de um ambiente para programar em React Native exige a configuração de uma série de ferramentas: instalação de um editor de código, Node.js, o gerenciador de pacotes npm ou yarn, o SDK do Android e/ou iOS, além de um emulador para testes.

Agora veremos o lado do AppSheet, a plataforma facilita o processo ao oferecer templates prontos e a possibilidade de iniciar um sistema diretamente no navegador, sem necessidade de instalação de ferramentas ou configurações de ambiente. O AppSheet também pode ser acessado pelo celular e pode ser baixado pelas lojas oficiais do Android e iOS.

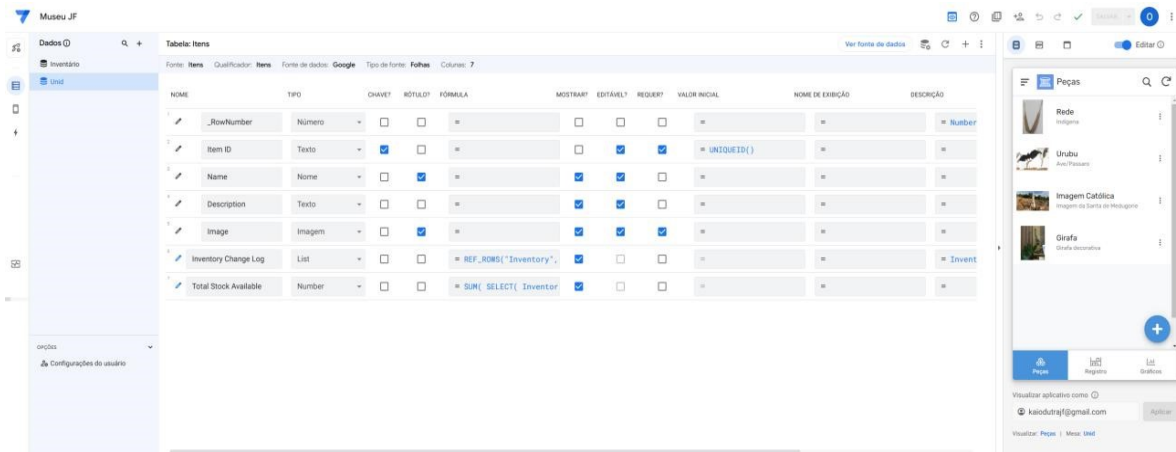
Figura 8. Tela de templates do AppSheet



Fonte: Elaborada pelo autor.

A Figura 8 mostra as opções de templates que a plataforma oferece aos usuários, mas também tem a opção de começar um aplicativo do zero.

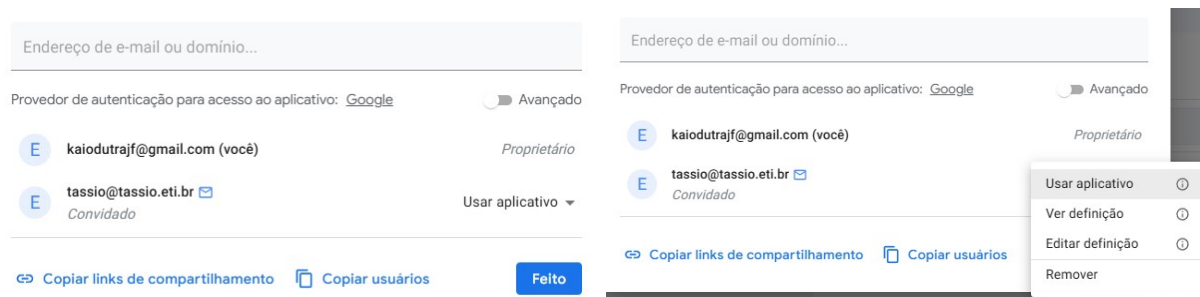
Figura 9. Edição da aplicação



Fonte: Elaborada pelo autor.

A Figura 9 exibe a edição do layout das peças com uma pré-visualização ao lado, permitindo o acompanhamento da aplicação rodando em tempo real.

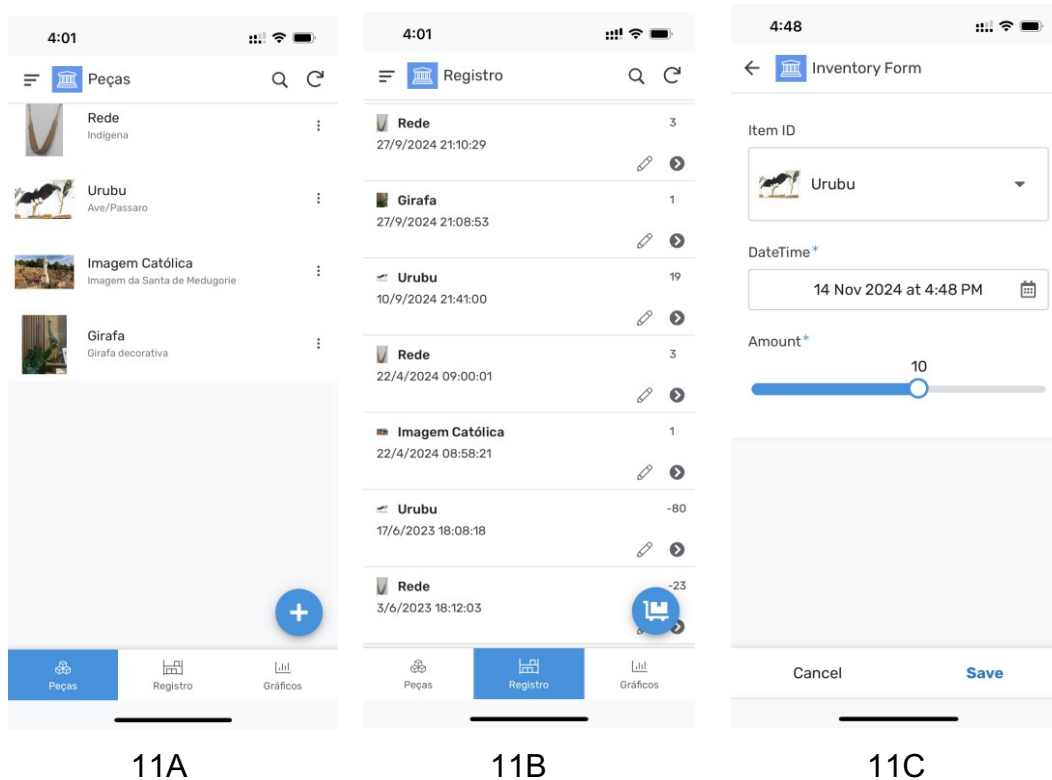
Figura 10. Acesso de colaboradores



Fonte: Elaborada pelo autor.

A Figura 10 exibe a parte de acesso da aplicação para usar a aplicação autenticado pelo Google. Na parte superior, há um campo para inserir novos endereços de e-mail ou domínios de participantes. Como proprietário, você possui controle total sobre o aplicativo e suas configurações de compartilhamento para outros usuários convidados.

Figura 11. Tela de peças

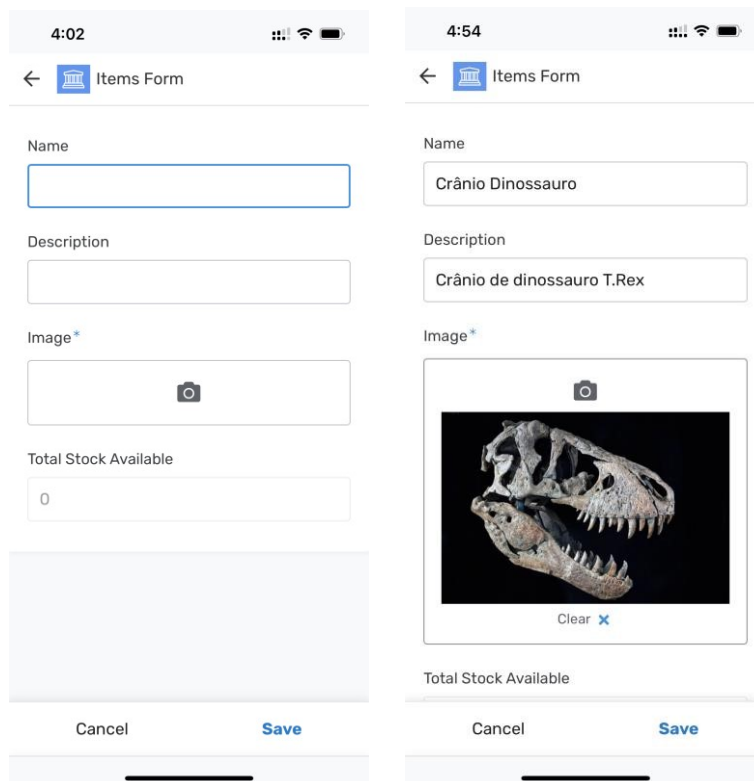


Fonte: Elaborada pelo autor.

A Figura 11 apresenta as telas referentes aos itens cadastrados no sistema e aos itens já presentes no acervo. A Figura 11A mostra alguns itens recém-adicionados, e a Figura 11B apresenta o histórico do acervo. Na tela de Peças podemos notar também que temos um botão azul com sinal de “+” no canto inferior direito para adicionar um novo item.

Na tela de Registro temos um botão azul no canto inferior direito para verificar e alterar o estoque de cada peça conforme Figura 11C.

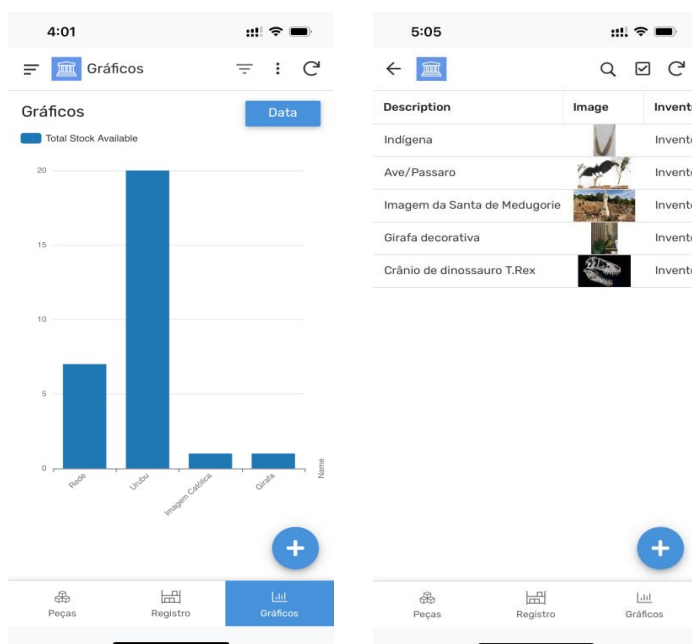
Figura 12. Tela de cadastro



Fonte: Elaborada pelo autor.

A Figura 12 apresenta a tela de cadastro dos itens. Essa tela é acessada pelo botão de adicionar.

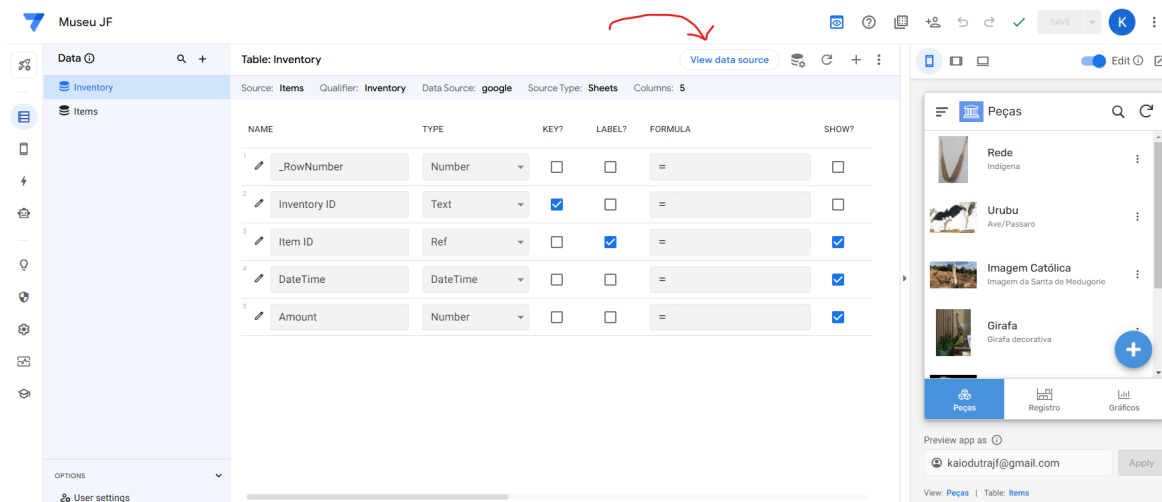
Figura 13. Tela do gráfico



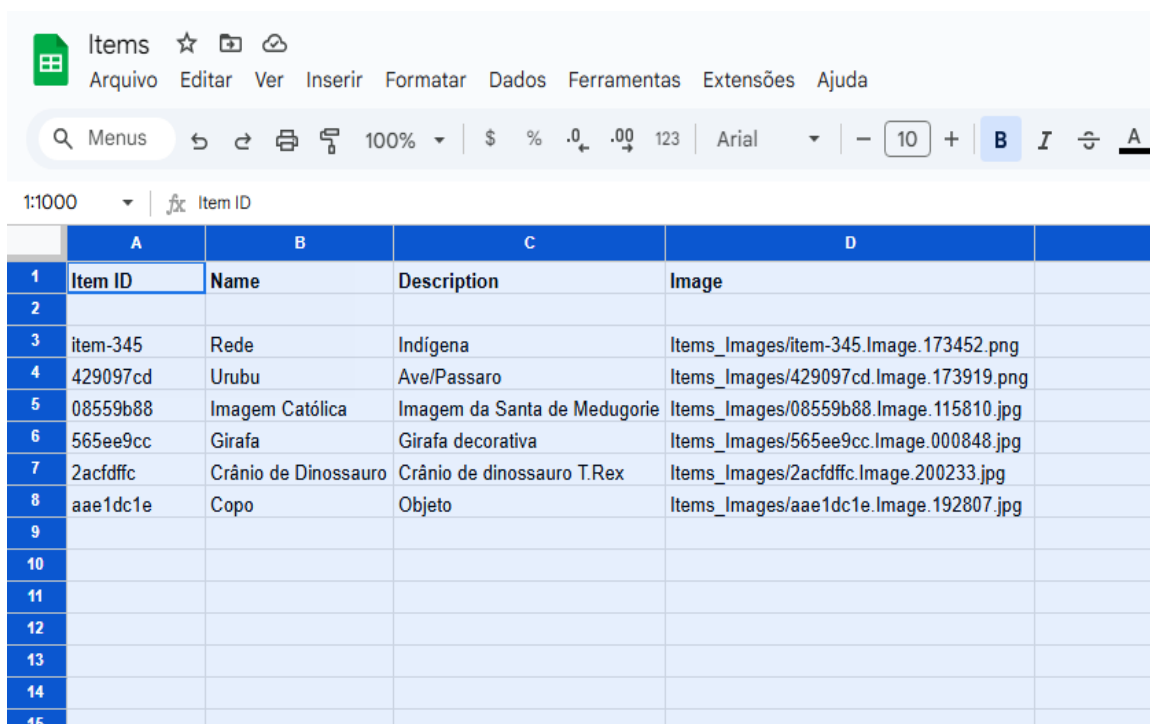
Fonte: Elaborada pelo autor.

A Figura 13 apresenta a quantidade de peças em estoque cadastrados no sistema, o botão azul com sinal de “+” no canto inferior direito para adicionar um novo item e o botão azul no canto superior para mudar a visualização para o modo de lista.

Figura 14. Exportação de Dados

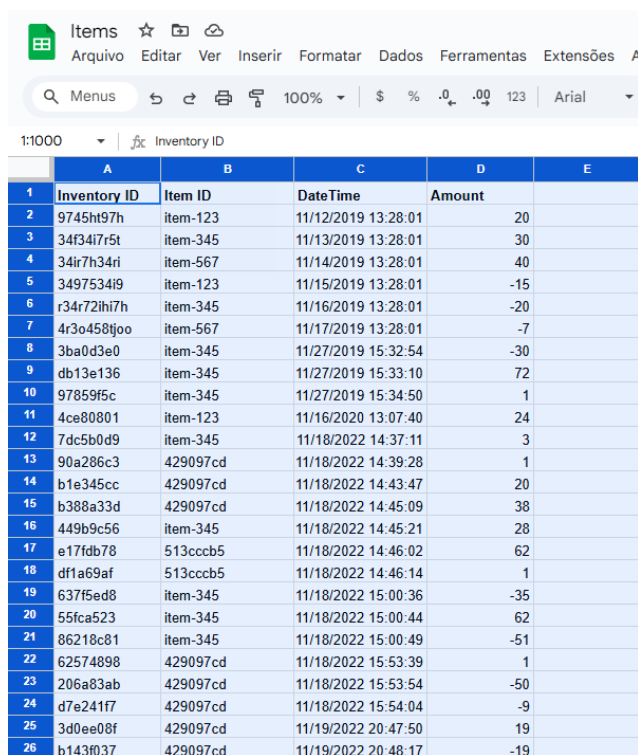


14A



	A	B	C	D
1	Item ID	Name	Description	Image
2				
3	item-345	Rede	Indígena	Items_Images/item-345.Image.173452.png
4	429097cd	Urubu	Ave/Passaro	Items_Images/429097cd.Image.173919.png
5	08559b88	Imagem Católica	Imagem da Santa de Medugorie	Items_Images/08559b88.Image.115810.jpg
6	565ee9cc	Girafa	Girafa decorativa	Items_Images/565ee9cc.Image.000848.jpg
7	2acfdffc	Crânio de Dinossauro	Crânio de dinossauro T.Rex	Items_Images/2acfdffc.Image.200233.jpg
8	aae1dc1e	Copo	Objeto	Items_Images/aae1dc1e.Image.192807.jpg
9				
10				
11				
12				
13				
14				
15				

14B



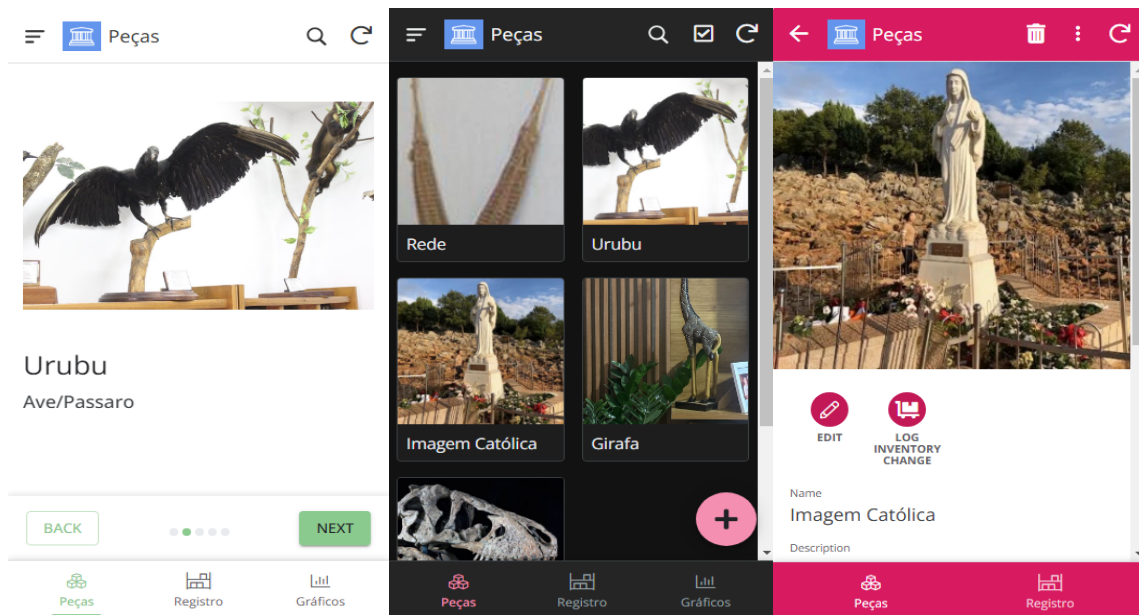
	A	B	C	D	E
1	Inventory ID	Item ID	DateTime	Amount	
2	9745ht97h	item-123	11/12/2019 13:28:01	20	
3	34f34i7r5t	item-345	11/13/2019 13:28:01	30	
4	34ir7h34ri	item-567	11/14/2019 13:28:01	40	
5	3497534i9	item-123	11/15/2019 13:28:01	-15	
6	r34r72ih7h	item-345	11/16/2019 13:28:01	-20	
7	4r3o450tjoo	item-567	11/17/2019 13:28:01	-7	
8	3ba0d3e0	item-345	11/27/2019 15:32:54	-30	
9	db13e136	item-345	11/27/2019 15:33:10	72	
10	97859f5c	item-345	11/27/2019 15:34:50	1	
11	4ce80801	item-123	11/16/2020 13:07:40	24	
12	7dc5b0d9	item-345	11/18/2022 14:37:11	3	
13	90a286c3	429097cd	11/18/2022 14:39:28	1	
14	b1e345cc	429097cd	11/18/2022 14:43:47	20	
15	b388a33d	429097cd	11/18/2022 14:45:09	38	
16	449b9c56	item-345	11/18/2022 14:45:21	28	
17	e17fdb78	513cccb5	11/18/2022 14:46:02	62	
18	df1a69af	513cccb5	11/18/2022 14:46:14	1	
19	637f5ed8	item-345	11/18/2022 15:00:36	-35	
20	55fca523	item-345	11/18/2022 15:00:44	62	
21	86218c81	item-345	11/18/2022 15:00:49	-51	
22	62574898	429097cd	11/18/2022 15:53:39	1	
23	206a83ab	429097cd	11/18/2022 15:53:54	-50	
24	d7e241f7	429097cd	11/18/2022 15:54:04	-9	
25	3d0ee08f	429097cd	11/19/2022 20:47:50	19	
26	b143f037	429097cd	11/19/2022 20:48:17	-19	

14C

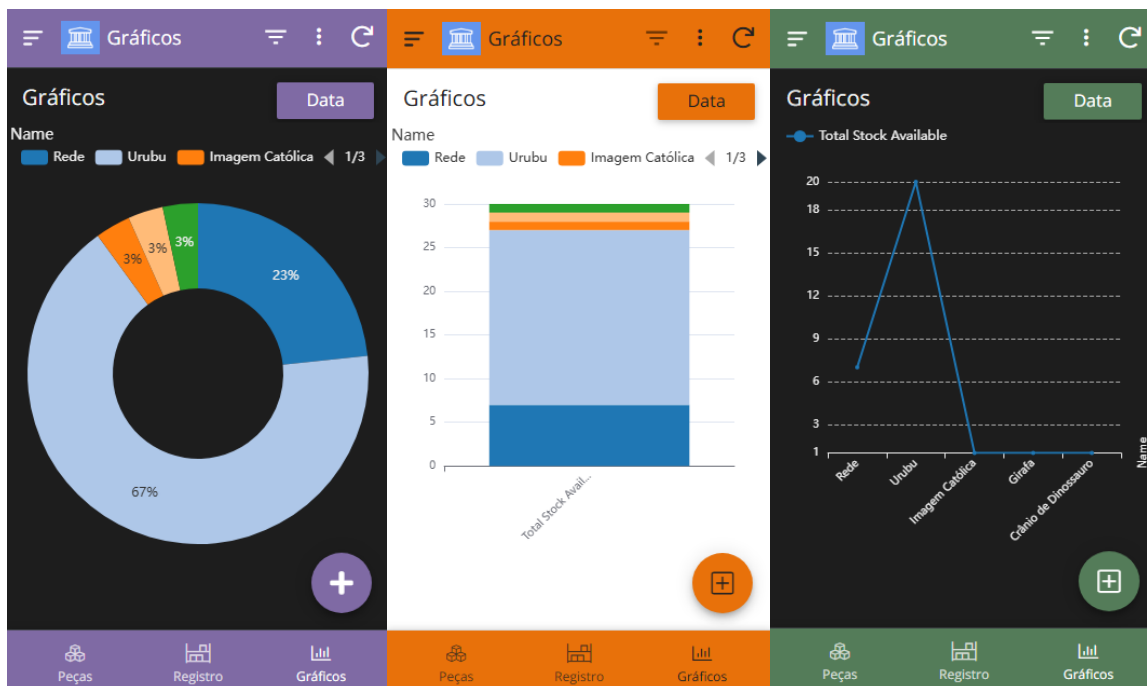
A Figura 14 ilustra o processo de exportação de dados via Google Sheets, utilizando o Google Docs. Na Figura 14A, é destacado o botão da aplicação responsável por gerar a exportação. Já a Figura 14B apresenta as peças disponíveis em estoque, enquanto a Figura 14C exibe a quantidade de entrada e saída de cada tipo de peça. Esses dados exportados podem ser facilmente integrados a outras aplicações, permitindo maior flexibilidade no uso e análise em diferentes plataformas ou sistemas.

Embora o AppSheet ofereça templates prontos como visto anteriormente, a plataforma também permite a customização de aplicativos, tornando-se uma excelente opção para criar protótipos personalizados. Esses protótipos podem ser desenvolvidos e utilizados diretamente no próprio AppSheet ou passar por uma etapa de avaliação, servindo como base para implementação em outras ferramentas.

Figura 15. Exemplos de outros protótipos personalizados.



15A



15B

Fonte: Elaborada pelo autor.

A Figura 15 apresenta diversos layouts prototipados na ferramenta, destacando a ampla gama de possibilidades de exibição, como troca de ícones, logotipos, fontes, estilos de navegação, tipos de gráficos, listagens de itens entre outros. Essa diversidade evidencia a praticidade na criação de protótipos de telas, demonstrando que o AppSheet não se limita a templates estáticos funcionais, mas permite alterações e personalizações.

4 RESULTADOS E DISCUSSÃO

Os resultados obtidos indicam que o uso de ferramentas no-code podem ser uma alternativa viável para o desenvolvimento de aplicações simples ou de pequeno porte, especialmente em contextos onde rapidez e acessibilidade são fatores críticos. Por exemplo, a aplicação desenvolvida no AppSheet foi concluída em aproximadamente 5 a 6 dias, enquanto o mesmo projeto, utilizando o método tradicional com React Native, levou cerca de 1 mês e meio para ser finalizado, considerando a curva de aprendizado em ambos os cenários.

O primeiro ponto observado é que as aplicações foram desenvolvidas em projetos integradores, ou seja, um ambiente totalmente acadêmico, onde ainda não havia um conhecimento avançado de programação. Nesse contexto, o AppSheet pôde substituir o desenvolvimento tradicional nessa aplicação do trabalho em questão ao oferecer um template pronto para uso.

O segundo ponto destacado é que, mesmo disponibilizando templates prontos, a ferramenta permite a customização desses modelos, tornando-se uma opção eficaz para o desenvolvimento de protótipos. Esses protótipos podem servir como base para implementação no próprio AppSheet ou em outras plataformas.

Ainda assim, é fundamental reconhecer os custos dessas ferramentas. A plataforma no-code apresenta um custo recorrente em seus planos de assinatura, que variam de 5 a 20 dólares por mês por usuário, com os valores aumentando

conforme o número de usuários ou a complexidade das demandas cresce⁷. No entanto, permite até 10 usuários gratuitos para a equipe, o que foi um fator facilitador no contexto acadêmico.

Por outro lado, o desenvolvimento tradicional, embora mais demorado e custoso inicialmente, oferece maior flexibilidade, controle sobre a infraestrutura, integração com bancos de dados externos e custos recorrentes potencialmente menores a longo prazo, especialmente em projetos hospedados em servidores próprios.

Portanto, as ferramentas no-code não substituem completamente as abordagens tradicionais de desenvolvimento em todos os cenários, mas podem funcionar como estratégias complementares.

5 CONSIDERAÇÕES FINAIS

De acordo com o que foi demonstrado neste artigo, foi possível desenvolver uma aplicação funcional de gerenciamento de estoque usando dois tipos distintos de tecnologias, que podem auxiliar muito no ambiente acadêmico.

A experiência de desenvolvimento mostrou que o AppSheet proporcionou agilidade e simplicidade, sendo possível concluir a aplicação em poucos dias, enquanto o React Native, embora mais demorado, garantiu maior flexibilidade e customização, ideal para demandas mais complexas. Essa dualidade reforça a complementaridade entre as abordagens, destacando a importância de escolher a ferramenta com base nas necessidades específicas do projeto.

Para trabalhos futuros, ambos os aplicativos podem receber novas funcionalidades. O código em React Native está disponível no GitHub⁸, e o acesso à aplicação desenvolvida no AppSheet pode ser solicitado através do e-mail do autor. Link do vídeo das aplicações rodando: <https://youtu.be/-WjOtLnzglA>

⁷ Disponível em: < <https://about.appsheets.com/pricing/> > Acesso em: 05/12/2024

⁸ Código-fonte. Disponível em <https://github.com/CaioDutra1412/Museu>. Acessado em 02 de dez. de 24.

REFERENCIAS

APPSHEET. **Pricing**. Disponível em: <https://about.appsheet.com/pricing/>. Acesso em: 6 dez. 2024.

EXPO. **Expo CLI documentation**. Disponível em: <https://docs.expo.dev/more/expo-cli/>. Acesso em: 2 nov. 2024.

FACEBOOK, INC. **React Native documentation. 2024**. Disponível em: <https://reactnative.dev/>. Acesso em: 22 out. 2024.

FALCÃO, Filipe Dourado. **Desenvolvimento do aplicativo Turistando Beberibe utilizando React Native**. 2022. Disponível em: <https://repositorio.ufc.br/handle/riufc/69029>. Acesso em: 5 dez. 2024.

GOOGLE. **Introdução à arquitetura de aplicativos Android**. Disponível em: <https://developer.android.com/topic/architecture/intro?hl=pt-br>. Acesso em: 2 nov. 2024.

NODE.JS. **About Node.js**. Disponível em: <https://nodejs.org/en/about>. Acesso em: 2 nov. 2024.

MALIK, Kashif. **Appsheet vs React Native: Evaluation of performance and development of Android Apps**. 2023. 39 p. Trabalho de conclusão de curso (Bacharelado em Ciência da Computação) - Metropolia University of Applied Sciences, 2023. Disponível em: https://www.theseus.fi/bitstream/handle/10024/505773/Malik_Kashif.pdf?sequence=2. Acesso em: 5 dez. 2024.

PANI, Jeferson Arco de; Filho, Ramilio Ramalho reis. **O impacto da gestão de estoque nas empresas**. 2023. Disponível em: <https://revista.fatectq.edu.br/interfacetecnologica/article/view/1670/929>. Acesso: 30 out. 2024

PETROVIC, Nenad et al. **A RAD-Based Approach to eMonevCLC for Monitoring and Evaluation in Community Learning Center**. Disponível em: https://www.researchgate.net/profile/Nenad-Petrovic/publication/383736279_A_RAD-Based_Approach_to_eMonevCLC_for_Monitoring_and_Evaluation_in_Community_Learning_Center/links/66d82ddf84dd1716c94b2e5/A-RAD-Based-Approach-to-eMonevCLC-for-Monitoring-and-Evaluation-in-Community-Learning-Center.pdf. Acesso em: 5 dez. 2024.

SANTOS, Gabriel Martins dos. **Estudo do uso de plataformas no-code para geração de MVPs**. 2022. Disponível em: <https://lume.ufrgs.br/handle/10183/252466> Acesso em: 5 dez. 2024