

## **DESENVOLVIMENTO DE APLICAÇÕES MULTIPLATAFORMAS PARA A ÁREA DE SAÚDE**

*Franciane Aprigio de Oliveira Castro<sup>1</sup>*

*Centro Universitário Academia, Juiz de Fora, MG*

*Gustavo Andrade Medeiros<sup>2</sup>*

*Centro Universitário Academia, Juiz de Fora, MG*

*Laura de Faria e Castro<sup>3</sup>*

*Centro Universitário Academia, Juiz de Fora, MG*

*Tassio Ferenzini Martins Sirqueira<sup>4</sup>*

*Centro Universitário Academia, Juiz de Fora, MG*

### **RESUMO**

O parto é um marco na vida de uma mulher e sua família, portanto contatos de qualidade entre mulheres e profissionais de saúde durante a gravidez são fundamentais para prevenir riscos, reduzir complicações e enfrentar desigualdades de saúde. Nesse contexto, o aplicativo "Parto Seguro" foi desenvolvido. O aplicativo visa proporcionar a médicos e gestantes acesso fácil às informações e orientações necessárias durante esse processo. As informações são centralizadas e organizadas por perfil de interesse, facilitando a compreensão e busca dos dados. Inicialmente foi criado um prototipo de tela no Figma (ferramenta de prototipação de telas), e para o desenvolvimento foi utilizado o React Native por ser um framework contemporâneo, que permite a criação de interfaces de usuário visualmente atraentes e intuitivas. Além disso também possui uma ampla comunidade de desenvolvedores e bibliotecas disponíveis e permite o desenvolvimento simultâneo para plataformas iOS e Android, otimizando a criação da aplicação. O aplicativo desenvolvido permite acessar informações de forma rápida e prática, proporciona pesquisas dentre os tópicos disponibilizados e ainda oferece a facilita o acesso de informações externas, com uma navegação web embutida. A aplicação então se mostra intuitiva, com informações para

---

<sup>1</sup> Discente do Curso de Engenharia de Software do Centro Universitário Academia – UniAcademia. E-mail: francianefad@outlook.com.

<sup>2</sup> Discente do Curso de Engenharia de Software do Centro Universitário Academia – UniAcademia. E-mail: gustavomedeiros15@outlook.com.br

<sup>3</sup> Discente do Curso de Engenharia de Software do Centro Universitário Academia – UniAcademia. E-mail: laurfariacastro@gmail.com

<sup>4</sup> Docente do Curso de Engenharia de Software do Centro Universitário Academia – UniAcademia. E-mail: tassiosirqueira@uniacademia.edu.br

gestantes e médicos facilmente acessáveis, viabilizando consultas e processos de parto mais seguros e confiáveis.

**Palavras-chave: Gestão da gravidez, Saúde materna, Aplicações móveis, React Native.**

### ***ABSTRACT***

Childbirth is a milestone in the life of a woman and her family, therefore quality contacts between women and health professionals during pregnancy are essential to prevent risks, reduce complications and address health inequalities. In this context, the "Parto Seguro" application was developed. The application aims to provide doctors and pregnant women with easy access to the information and guidance they need during this process. Information is centralized and organized by profile of interest, making it easier to understand and search for data. Initially, a screen prototype was created in Figma (screen prototyping tool), and React Native was used for development as it is a contemporary framework, which allows the creation of visually attractive and intuitive user interfaces. In addition, it also has a wide community of developers and libraries available and allows simultaneous development for iOS and Android platforms, optimizing application creation. The application developed allows you to access information quickly and conveniently, provides searches among the topics available and also offers easy access to external information, with built-in web navigation. The application then appears intuitive, with information for pregnant women and doctors easily accessible, enabling safer and more reliable consultations and birth processes.

## **1 INTRODUÇÃO**

O parto é um evento significativo na vida de uma mulher e de sua família, com implicações profundas para a saúde e bem-estar tanto da mãe quanto do bebê. A Organização Mundial da Saúde (OMS) enfatiza que cada gravidez e nascimento é único e que o melhor cuidado intraparto para cada mulher e seu bebê é individualizado e baseado em evidências [1].

Mais e melhores contatos de qualidade entre todas as mulheres e seus profissionais de saúde durante a gravidez facilitam a adoção de medidas preventivas, a detecção oportuna de riscos, a redução de complicações e o enfrentamento das desigualdades de saúde. Portanto é crucial que as gestantes e os médicos tenham orientações facilmente acessíveis durante esse processo, para que as gestantes recebam cuidados de qualidade durante a gravidez, o parto e o período pós-natal, garantindo que elas e seus bebês alcancem seu pleno potencial para a saúde e o bem-estar [2].

Os aplicativos móveis desempenham um papel fundamental na era digital atual, proporcionando acesso rápido e conveniente a informações e dados. Eles permitem que os usuários obtenham informações instantâneas, realizem tarefas diárias com eficiência e se conectem com o mundo ao seu redor. Além disso, os aplicativos móveis adicionam conforto e conveniência, sendo facilmente baixados e usados para diversos fins. A maioria das pessoas ao redor do mundo os utiliza para negócios e atividades do dia a dia, sendo ferramenta um grande facilitador no acesso de informações [3].

Neste contexto, foi desenvolvido o aplicativo “Parto Seguro”, que visa proporcionar a médicos e gestantes acesso as informações e orientações necessárias a este processo. O aplicativo foi feito buscando centralizar as informações, organizando-as por perfil de interesse (médicos e gestantes), com diversas seções, para facilitar a compreensão e busca dos dados.

## **2 REFERENCIAL TEÓRICO**

A prototipação é importante antes do desenvolvimento de software, pois permite visualizar como o software funcionará, facilitando a identificação e correção de problemas potenciais no início do processo. Além disso, a prototipação ajuda a garantir que o produto atenda às expectativas do usuário e aos requisitos do projeto [4]. Por isso foi criado um protótipo das telas do aplicativo na ferramenta Figma, framework de prototipação, escolhido por sua eficiência, capacidade de reutilização de código, desempenho e velocidade, permitindo um fluxo de trabalho suave e eficiente [5].

Para desenvolvimento do projeto em questão foi realizado utilizando o framework React Native em conjunto com o Expo. A escolha por essas tecnologias foi motivada por uma série de fatores. O React Native é conhecido por sua eficiência e pela capacidade de reutilização de código, o que permite um desenvolvimento mais rápido e eficaz. Além disso, ele oferece um desempenho robusto para aplicativos móveis e conta com uma grande comunidade de suporte, o que facilita a resolução de problemas e o compartilhamento de soluções [6].

O Expo, por sua vez, é uma plataforma de código aberto que simplifica ainda mais o desenvolvimento com React Native. Ele fornece um conjunto de ferramentas que permitem criar aplicativos de maneira mais rápida e sem a necessidade de configurações complexas. Isso contribui para a agilidade do processo de desenvolvimento e permite que os desenvolvedores se concentrem mais na lógica do aplicativo e menos na configuração do ambiente de desenvolvimento [7].

Além do Expo, o projeto também utilizou o Styled Components para a estilização dos componentes do aplicativo. O Styled Components é uma biblioteca que permite escrever CSS real em JavaScript, o que traz inúmeros benefícios, como a capacidade de aproveitar toda a flexibilidade do CSS e a potência do JavaScript [8].

Para a navegação entre as telas do aplicativo, foi utilizado o React Navigation. Esta é uma solução popular e eficaz para a navegação em aplicativos construídos com React Native [9]. Por fim, para a inclusão de ícones no aplicativo, foram utilizados os ícones disponibilizados pelo próprio React Native [10].

### **3 METODOLOGIA**

O projeto seguiu uma metodologia de desenvolvimento incremental, começando com a criação de um protótipo do aplicativo no Figma, com as principais telas, o layout desejado e o design dos componentes necessários ao sistema, que são reutilizados em toda a aplicação, mantendo a padronização e coesão do aplicativo.

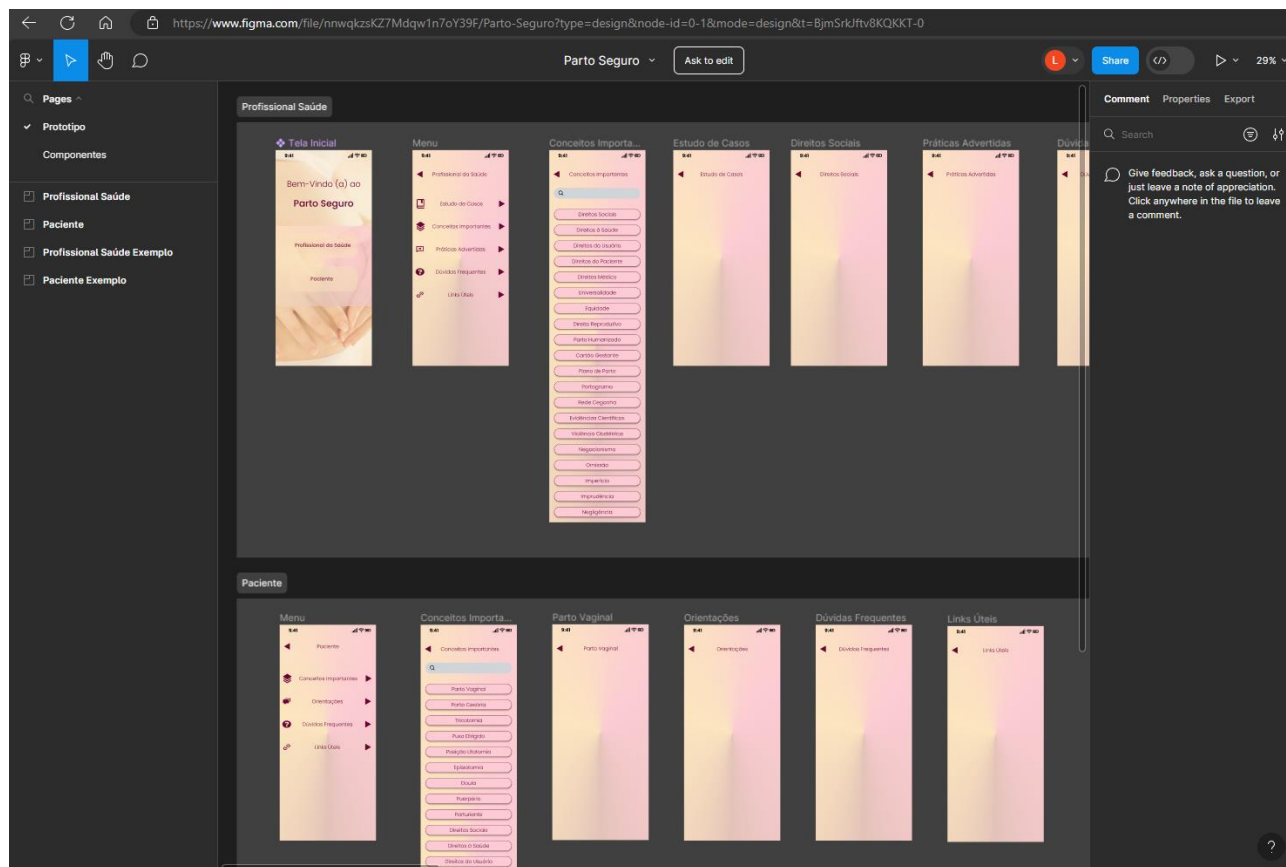


Figura 1: Protótipo do aplicativo criado no Figma

O código foi desenvolvido progressivamente no GitHub [11], plataforma de desenvolvimento de software online que facilitou o compartilhamento de arquivos de código e a colaboração com entre os desenvolvedores do projeto. Na plataforma, foi criada uma branch específica para manter a versão mais estável e configurável do aplicativo, e para implementar novas alterações era necessário criar *pull requests* que eram revisados por outros membros da equipe (que não tivessem participado no desenvolvimento do novo código). Isso permitiu refinar o aplicativo ao longo do tempo para garantir eficiência e qualidade.

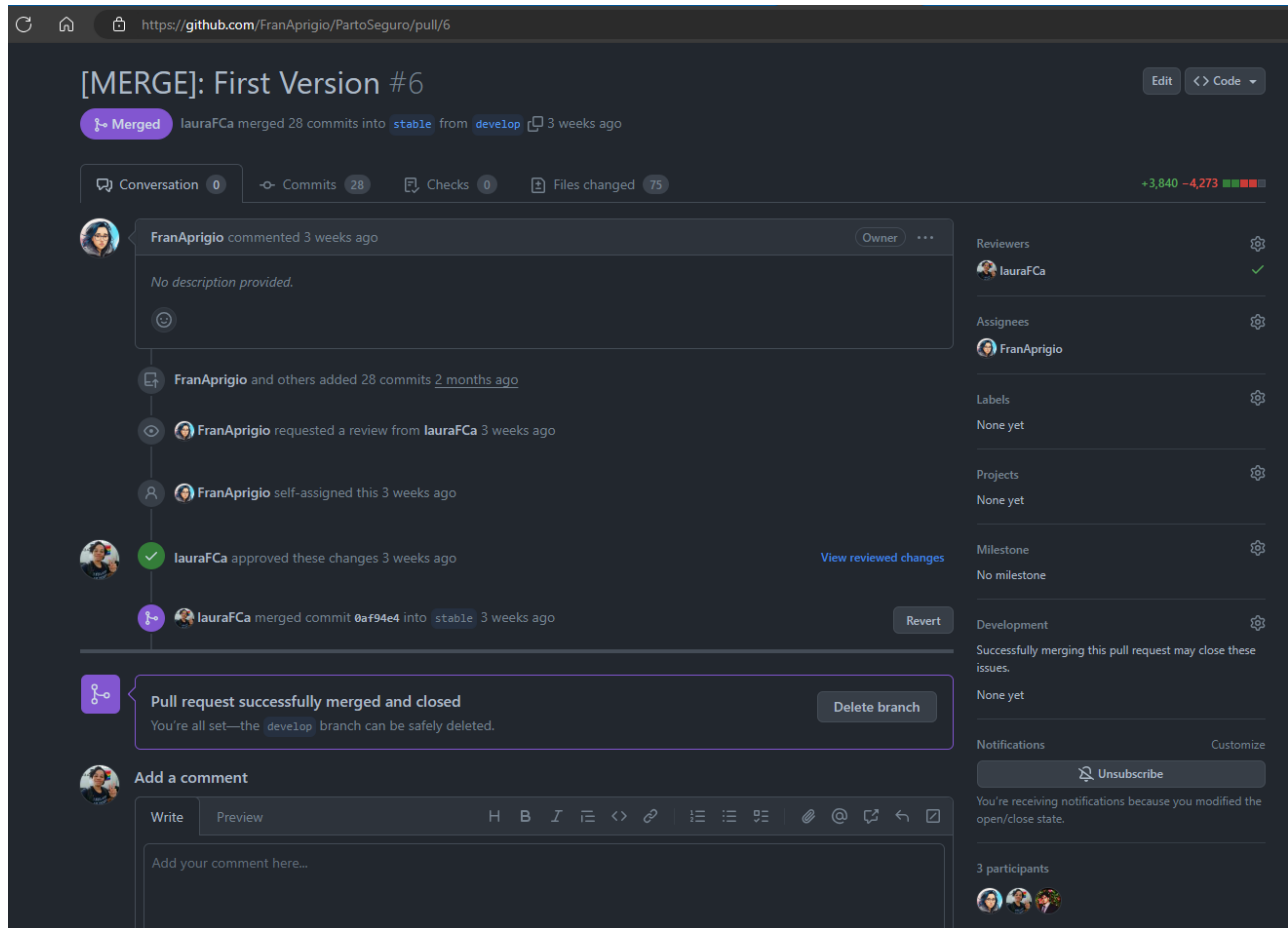


Figura 2: Exemplo de pull request criado no GitHub

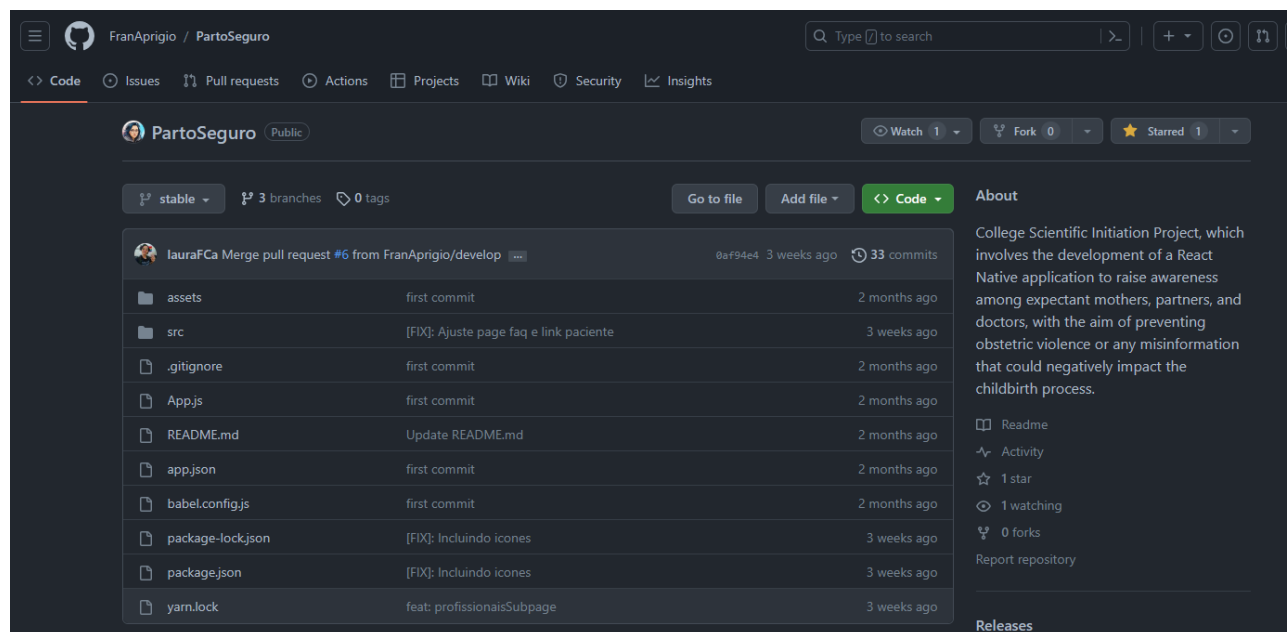


Figura 3: GitHub do projeto

As atividades foram distribuídas entre os membros do grupo, considerando pequenas tarefas que aos poucos foram levando a fomentação do aplicativo. Atividades desde a alteração de textos, criação de componentes, a criação de novas páginas e rotas, foram feitas aos poucos para que o projeto pudesse ter uma base sólida e uma organização coesa.

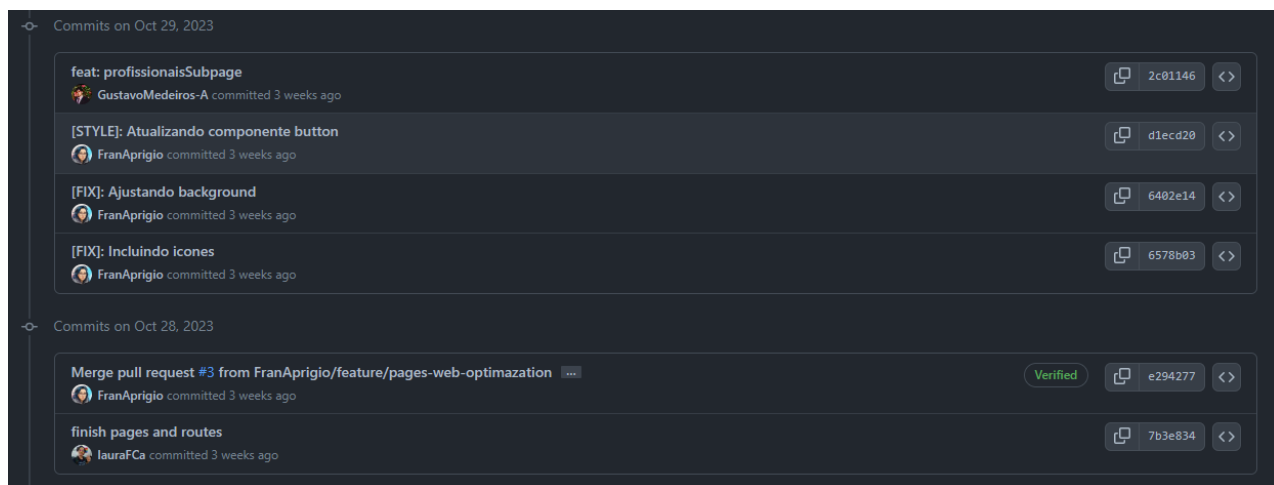
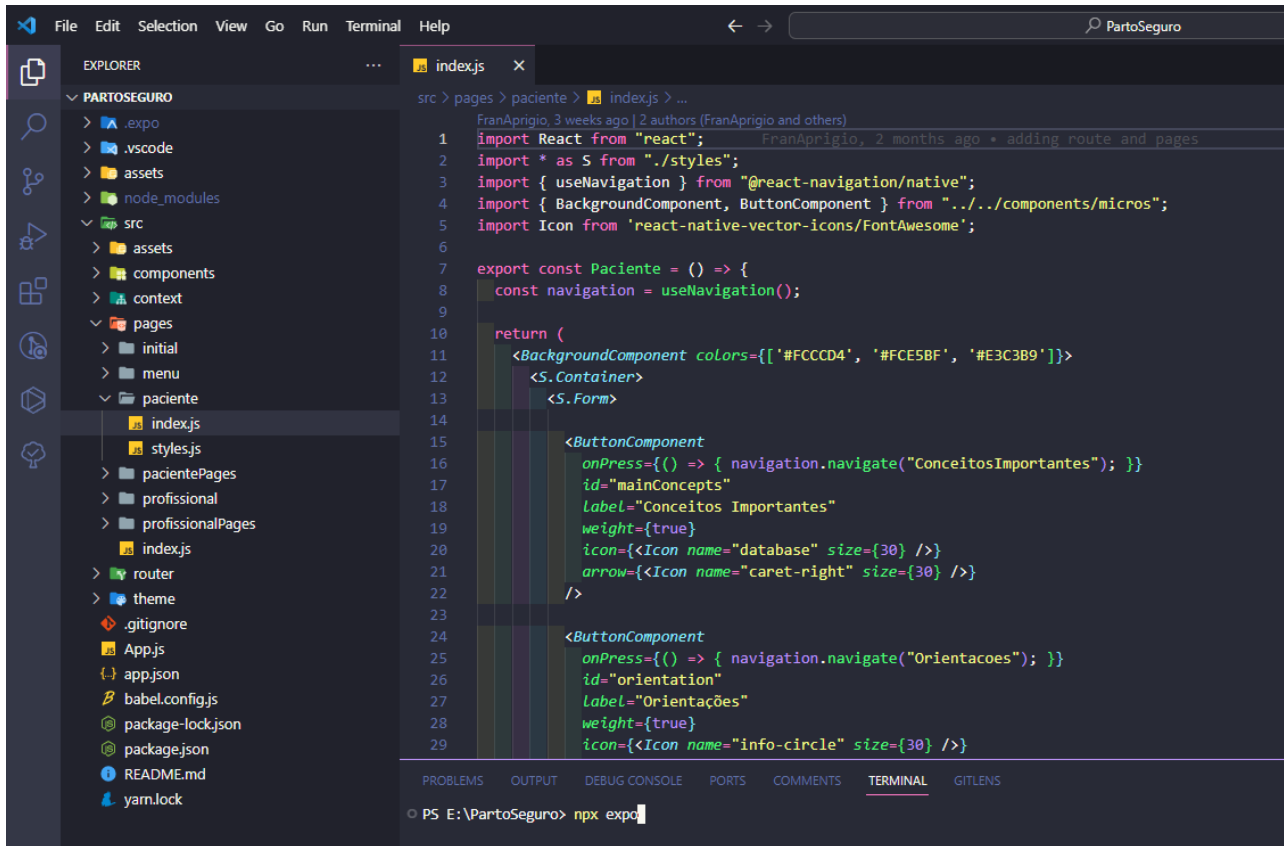


Figura 4: Exemplo das implementações incrementais

A organização do código seguiu o padrão já criado pelo React Native, tendo as os arquivos referentes ao conteúdo do projeto alocados em uma pasta “src”. Dentro deste diretório há a subdivisão das partes do projeto, e a abordagem adotada foi a organização de telas e componentes em pastas contendo um arquivo “index.js”, onde fica o código referente a tela.



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays the project structure for 'PARTOSEGURO'. The 'src' directory is expanded to show 'pages', which contains a 'paciente' folder. Inside 'paciente', the 'index.js' file is selected. The main editor area shows the code for 'index.js', which imports React, styles, navigation, and icons. It defines a 'Paciente' component that returns a nested structure of components: a BackgroundComponent with colors, a Container, a Form, and two ButtonComponents. The first ButtonComponent is labeled 'Conceitos Importantes' and the second is labeled 'Orientações'. The terminal at the bottom shows the command 'PS E:\PartoSeguro> npx expo'.

```
1 import React from "react";
2 import * as S from "./styles";
3 import { useNavigation } from "@react-navigation/native";
4 import { BackgroundComponent, ButtonComponent } from "../../components/micros";
5 import Icon from 'react-native-vector-icons/FontAwesome';
6
7 export const Paciente = () => {
8   const navigation = useNavigation();
9
10  return (
11    <BackgroundComponent colors={['#FCCCD4', '#FCE5BF', '#E3C3B9']}>
12      <S.Container>
13        <S.Form>
14
15          <ButtonComponent
16            onPress={() => { navigation.navigate("ConceitosImportantes"); }}
17            id="mainConcepts"
18            Label="Conceitos Importantes"
19            weight={true}
20            icon={<Icon name="database" size={30} />}
21            arrow={<Icon name="caret-right" size={30} />}
22          />
23
24          <ButtonComponent
25            onPress={() => { navigation.navigate("Orientacoes"); }}
26            id="orientation"
27            Label="Orientações"
28            weight={true}
29            icon={<Icon name="info-circle" size={30} />}
30          />
31        </S.Form>
32      </S.Container>
33    </BackgroundComponent>
34  );
35}
```

Figura 5: Organização do código do projeto

Para a definição das rotas, devido a quantidade de telas, foi criada uma lógica de geração de rotas, onde, cada nova rota era adicionada apenas a um vetor, com os parametros necessários. A partir desse vetor a nova rota era criada, já com todos os outros parametros padrão necessários.





```
route.js M X
src > router > route.js > Route
You, 6 seconds ago | 4 authors (Gustavo Medeiros and others)
1 > import React from "react"; ...
52
53 const Stack = createNativeStackNavigator();
54
55 const Route = () => {
56 >   const pacientePages = [ ...
82   ];
83
84 >   const conceitosImportantesSubPagePaciente = [ ...
100   ];
101
102   const OrientacoesSubPage = [
103     {
104       page: "Direitos do Acompanhante?",
105       name: "DireitosAcompanhante",
106       componente: DireitosAcompanhante,
107     },
108     {
109       page: "Plano de Partos",
110       name: "PlanoParto",
111       componente: PlanoParto,
112     },
113   ];
114
115 >   const profissionalPages = [ ...
146   ];
147
148 >   const conceitosImportantesSubPage = [ ...
164   ];
```

Figura 6: Exemplo de vetores criados.

```
/* Paginas do paciente */
{pacientePages.map((item, index) => (...))}

{conceitosImportantesSubPagePaciente.map((item, index) => (...))}

{OrientacoesSubPage.map((item, index) => (
  <Stack.Screen
    key={index}
    style={{ flex: 1, justifyContent: "space-between" }}
    name={item.name}
    component={item.componente}
    options={({ navigation }) =>
      getHeaderOptions(item.page, navigation)
    }
  />
)}}

/* Paginas do profissional de saude */
{profissionalPages.map((item, index) => (...))}
```

Figura 7: Uso do vetor na criação de rotas

A mesma ideia de vetores foi aplicada a páginas com subpáginas na aplicação. Isso foi possível graças a reutilização de componentes, garantindo a padronização dos resultados em qualquer página com subpáginas. Um exemplo de componente reutilizado é o “<BackgroundComponent>” que garante que o desing definido no Figma seja aplicado a todas as telas, sendo necessário apenas englobar o código com esse elemento.

```
index.js x
src > pages > pacientePages > conceitosimportantes > index.js > ...
FranAprigio, 3 weeks ago | 2 authors (Laura C and others)
1 import React from "react";
2 import { BackgroundComponent } from "../../components/micros";
3 import { StyleSheet, View } from "react-native";
4 import SearchInput from "../../components/search";
5 Laura C, last month • update pages
6 export const ConceitosImportantes = () => {
7
8   const data = [
9     { pagina: "Diretos Reprodutivo", nav: "DireitoReprodutivo" },
10    { pagina: "Cartão Gestante", nav: "CartaoGestante" },
11    { pagina: "Violência Obstétrica", nav: "ViolenciaObstetrica" },
12  ]
13
14  return (
15    <BackgroundComponent colors={['#FCCCD4', '#FCE5BF', '#E3C3B9']}>
16      <View style={styles.body}>
17        <SearchInput data={data} />
18      </View>
19    </BackgroundComponent>
20  );
21 };
22
23
24 const styles = StyleSheet.create({
25   body: {
26     flex: 1,
27     flexDirection: 'column',
28     padding: 30,
29     marginTop: 70,
```

Figura 8: Uso dos vetores para geração de subpáginas

Outro componente reutilizado é o componente de pesquisa nas telas com subtelas. Esse é o elemento que recebe a lista de subpaginas e renderiza na tela todas as subpaginas

passadas, e ainda o componente de busca, que permite filtrar as opções da tela, por opções com o valor relacionado ao de pesquisa.

```
index.js x
src > components > search > index.js > SearchInput
FranAprigio, 3 weeks ago | 2 authors (FranAprigio and others)
1 import React, { useState } from "react";
2 import { StyleSheet, View, ScrollView, TextInput } from "react-native";
3 import { PinkButtonComponent } from "../micros";
4 import { useNavigation } from "@react-navigation/native";
5 import Icon from 'react-native-vector-icons/FontAwesome';
6
7 export default function SearchInput({ data, isOutLink }) {
8   const navigation = useNavigation();
9   const [searchText, setSearchText] = useState('');
10
11   const filterData = (text) => {
12     const filteredItems = data.filter(item => item.pagina.toLowerCase().includes(text.toLowerCase()));
13     return filteredItems;
14   };
15
16   const filteredData = filterData(searchText);
17
18   return (
19     <View>
20       <TextInput
21         style={styles.search}
22         placeholder="Pesquisar..."
23         onChangeText={text => setSearchText(text)}
24         value={searchText}
25       />
26
27       <Icon style={styles.icon} name="search" size={25} color="#6C0434" />
28       <ScrollView>
29         {filteredData.map((item, index) => (
30           <PinkButtonComponent
31             key={index}
32             btnText={item.pagina}
33             goToPage={() => {
34               if (isOutLink) {
35                 navigation.navigate('WebPage', { webUrl: item.nav });
36               } else {
37                 navigation.navigate(item.nav);
38               }
39             }}
40           />
41         ))}
42       </ScrollView>
43     </View>
44   );
45
46
```

Figura 9: Componente de pesquisa

Por fim, para conseguir atingir o resultado esperado do aplicativo, foi necessário utilizar algumas bibliotecas do react-native, além das já mencionadas, como navigation e styled-components. Essas bibliotecas seguem a tabela [1].

Tabela 1: Bibliotecas adicionais do projeto

<b>Biblioteca</b>	<b>Origem</b>	<b>Uso</b>
expo-linear-gradient	Expo	Criação do fundo com gradiente do aplicativo
react-native-vector-icons	React Native	Biblioteca de ícones do react native
react-native-webview	React Native	Permite abrir páginas (urls) dentro do aplicativo, nativamente

#### **4 RESULTADOS E DISCUSSÃO**

O desenvolvimento de um aplicativo em React Native é rápido e otimizado, possuindo apenas uma curva normal de aprendizado inicial, que pode ser superada em pouco tempo. Isso faz com que, mesmo um desenvolvedor com menor experiência consiga participar ativamente do processo em pouco tempo de rump up (aprendizado).

Com as ferramentas listadas o desenvolvimento foi simplificado, com a criação das páginas e componentes mais rápidas, sem ser necessário configurações ou implementações complexas. Além disso, or ser um framework multi plataforma, o React Native permitiu que o mesmo código desenvolvido gerasse um aplicativo para Android e para iOS, otimizando o processo de desenvolvimento e release.

O aplicativo criado permitiu um acesso facilitado do público alvo as informações desejadas. A organização e separação de páginas, conteúdos e acessos faz com que a aplicação fique mais intuitiva, prática e visualmente agradável. Além de ser consistente em quaisquer plataformas de uso.

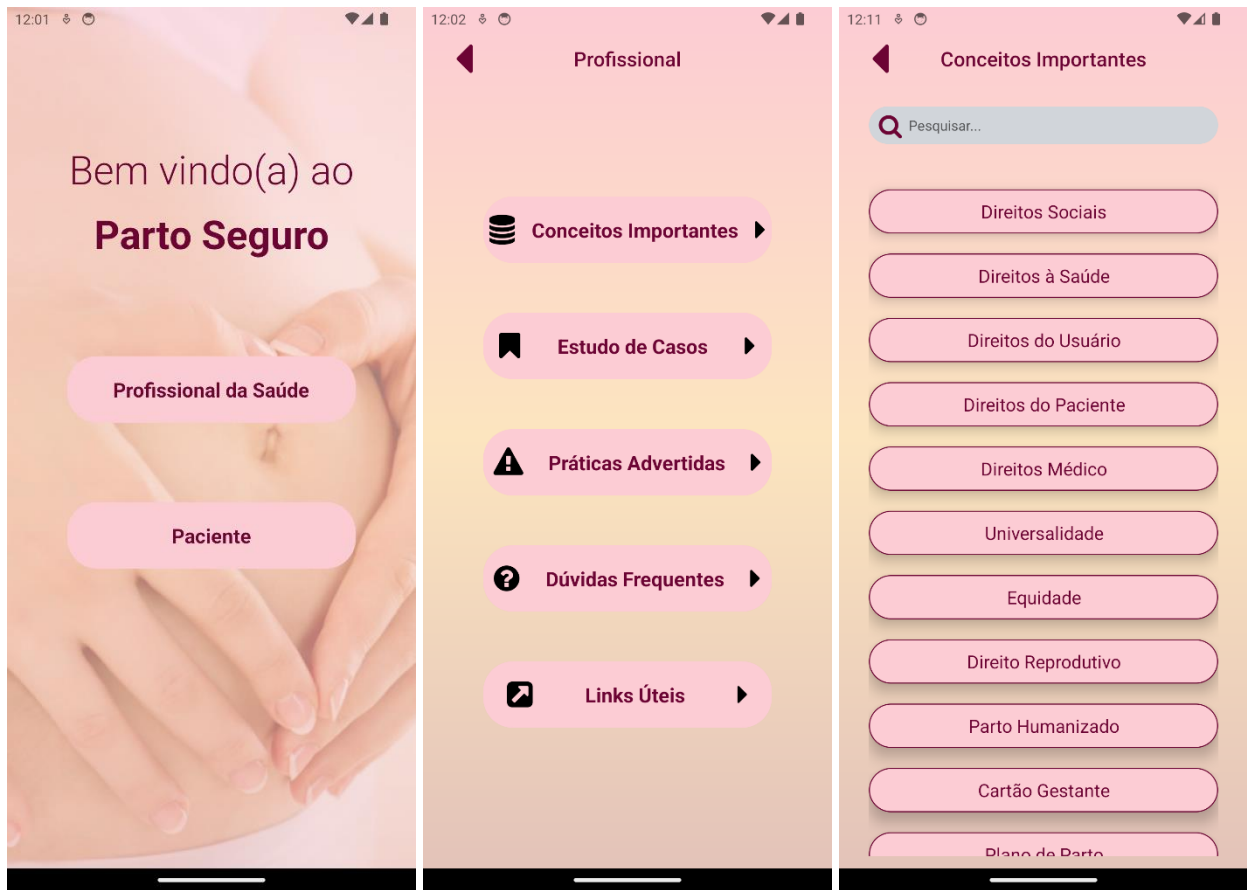


Figura 4: Exemplo das principais telas do aplicativo

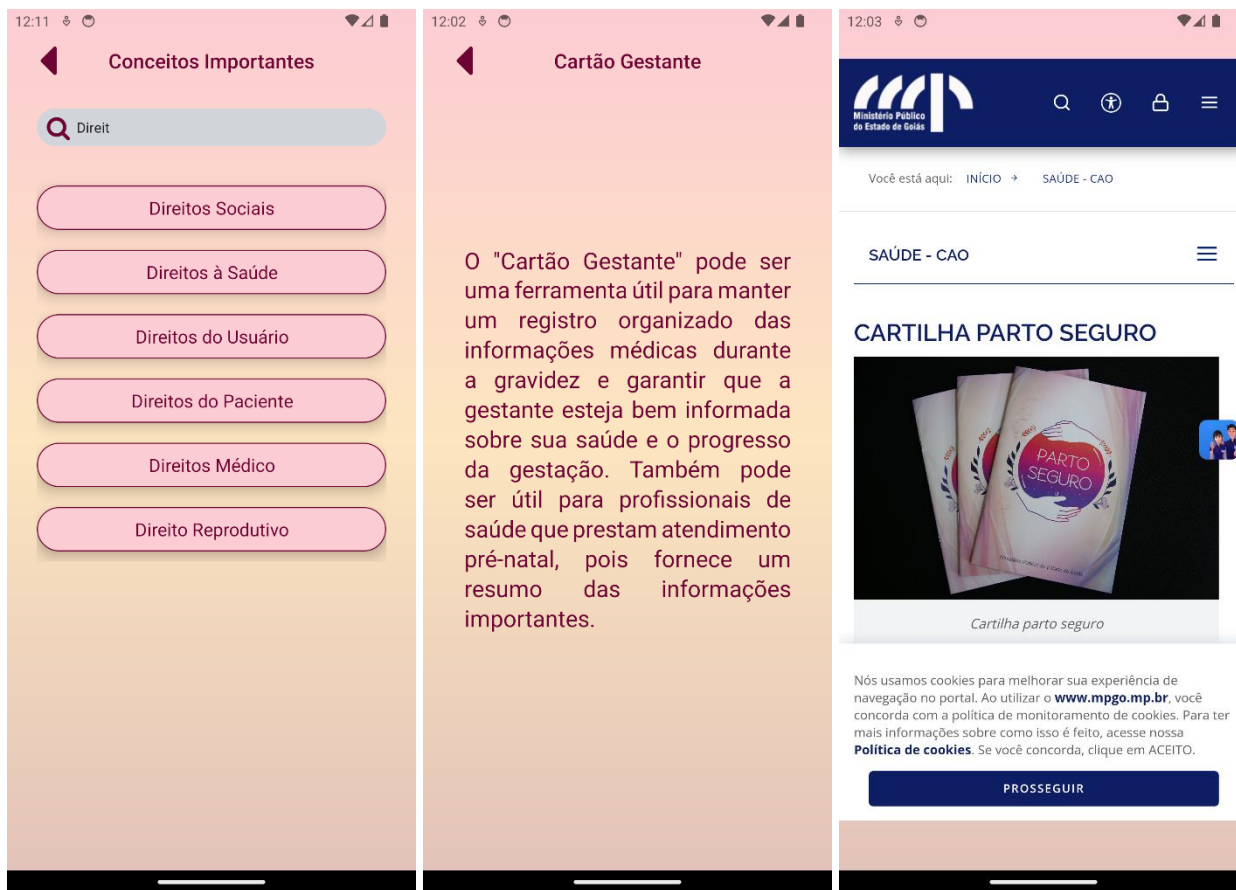


Figura 5: Exemplo das principais telas do aplicativo

Ao longo do desenvolvimento foi possível perceber a importância da colaboração e comunicação durante um processo de criação de software. Isso garante que nenhuma issue ou situação passe despercebida, e mais, garante que cada membro da equipe está trabalhando em partes diferentes, o que evita gasto de trabalho desnecessário, conflitos ao juntar códigos ou ainda, a possível duplicidade de códigos.

## 5 CONSIDERAÇÕES FINAIS

O projeto proporcionou uma grande oportunidade de aprendizado em desenvolvimento colaborativo, permitindo aprimorar habilidades em ferramentas como Expo, Styled Components e React Navigation. A experiência adquirida não apenas aprofundou o conhecimento técnico, mas também reforçou a importância do trabalho em equipe no desenvolvimento de software. Os frameworks selecionados permitiram um desenvolvimento mais ágil e consiso, além de garantir uma interface consisa e visualmente agradável ao usuário.

O uso do Figma no projeto foi uma decisão estratégica que contribuiu significativamente para o seu sucesso pois permitiu à equipe visualizar e testar a experiência do usuário antes mesmo do início do desenvolvimento, economizando tempo e evitando retrabalho.

A abordagem incremental adotada no projeto permitiu uma melhora contínua no aplicativo, garantindo a qualidade e eficiência do resultado final, e a melhor organização das tarefas dentre os membros do projeto. Esta estratégia permitiu a identificação e correção de problemas em estágios iniciais, economizando tempo e recursos, e resultando em um produto mais robusto e confiável.

A colaboração no GitHub desempenhou um papel crucial no fortalecimento da coordenação da equipe. A plataforma facilitou a comunicação, o compartilhamento de código e a resolução de problemas, permitindo que a equipe alcançasse os resultados desejados de maneira eficiente e eficaz. Além de garantir a estabilidade e confiabilidade da versão final do aplicativo, que sempre recebia códigos já revisados, aumentando a qualidade da entrega.

Por fim a comunicação como um todo foi essencial aos membros do projeto para garantir a divisão de tarefas, evitar retrabalhos e conflitos.

## **ABSTRACT**

This work presents a collaborative development project focused on creating an application using technologies such as Expo, Styled Components, and React Navigation. The experience provided technical learning and underscored the importance of teamwork. The strategic use of Figma allowed for early visualization and testing of the user experience, saving time. The incremental approach led to continuous improvements, early issue identification, and a robust end product. Collaboration on GitHub strengthened team coordination, facilitating communication and efficient problem resolution. Emphasizing overall communication was crucial for task delegation and conflict prevention, contributing to the delivery of a stable and reliable application.

## REFERÊNCIAS

- RENAULD MONTEIRO SUÁREZ, Juliana; COSTA PECLAT, Karolina; DORÉA DA SILVA, Marcelle; RODRIGUES DO NASCIMENTO, Mariana; S. B. DE A. DE A. CARVALHO, Yaísa Christina. **A Violência Obstétrica E Sua Repercussão À Vida Da Mulher E Família**. EduCapes, Rio de Janeiro, p. 7-16, 30 ago. 2021. Disponível em: <http://educapes.capes.gov.br/handle/capes/602861>. Acesso em: 26 nov. 2023.
- SAUDI MEDICAL JOURNAL (GENEVA). **Pregnant Women Must Be Able To Access The Right Care At The Right Time, Says Who**. ProQuest, [S. l.], p. Vol. 38, Ed. 1, 109-110, 1 jan. 2017.
- BERTOLI, Michele. **React Design Patterns and Best Practices**. Reino Unido, p. 175-177, O'Reilly Media. 2017.
- BANKS, A.; PORCELLO, E. **Learning React: Functional Web Development With React and Redux**. ISBN 978-1-491-95462-1. O'Reilly Media. 2017.
- EISENMAN, B. **Learning React Native**. ISBN 978-1-491-98914-2. O'Reilly Media. 2018.
- FLANAGAN, D. **JavaScript: The Definitive Guide**. ISBN 978-05-96805-52-4. O'Reilly Media. 2020.
- iOS. **Build Your Apps for iOS 12**. Disponível em <<https://developer.apple.com/ios>>. Acesso em: 20 de Novembro de 2023.
- LECHETA, R. R. **Google Android: Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. ISBN 978-8575224687. Novatec. 2015.
- PONTES, G. **Progressive Web Apps: Construa Aplicações Progressivas com React**. ISBN 978-85-94188-56-4. Casa do Código. 2018.
- RICCIARDI, W.; BARROS, P.P. et all. **How to govern the digital transformation of health services**. European Journal of Public Health, Vol 29. Outubro 2019. <https://doi.org/10.1093/eurpub/ckz165>
- TANENBAUM, A. **Redes de Computadores**. ISBN 978-85-8260-56-08. Bookman, 6ª Edição. 2021.
- WARNER, K. **The economic implications of preventive health care**. Social Science & Medicine. Part C: Medical Economics, Vol 23. Dezembro 1979. Disponível em <<https://sciencedirect.com/science/article/abs/pii/0160799579900042>>. Acesso em: 20 de Novembro de 2023.
- YANG, X. **Health expenditure, human capital, and economic growth: na empirical study of developing countries**. International Journal of Health Economics and Management, Vol 20. Outubro 2019. Disponível em <<https://link.springer.com/article/10.1007/s10754-019-09275-w>>. Acesso em: 20 de Novembro de 2023.