



CONSTRUÇÃO DE APLICATIVO PARA AUXÍLIO NO ENFRENTAMENTO À PANDEMIA DA COVID-19

André Luis Pereira da Luz
Heitor Rosseto Zaremba
Vitor Hugo Cavalcanti
Nikolas Augusto de Almeida
Renato Sellaro Dorighello

RESUMO

O presente artigo tem como objetivo apresentar o aplicativo mobile “XôCovid”, desenvolvido por quatro alunos estudantes das Faculdades da Indústria Unidade Senai CIC. O respectivo aplicativo busca auxiliar a população pela tal realidade vivenciada diante da pandemia da COVID-19, causada pelo vírus SARS-CoV-2. Nesse aspecto, se diferencia de outros aplicativos vigentes por apresentar históricos dos sintomas relatados pelo usuário, realizando um acompanhamento virtual.

Palavras-chave: aplicativo; pandemia; SARS-CoV-2.

MOBILE APPLICATION DEVELOPMENT IN ORDER TO FACE THE COVID-19 PANDEMIC

ABSTRACT

The present article has the objective of presenting the mobile application “XôCovid”, developed by four students of Faculdades da Indústria. The respective application aims to help the population in the reality experienced in face of the COVID-19 pandemic, caused by the SARS-CoV-2 virus. In this aspect, it differentiates itself from other current applications by presenting histories of the symptoms reported by the user, thus accomplishing a virtual follow-up.

Key words: application; pandemic; SARS-CoV-2.

INTRODUÇÃO

A pandemia provocada pelo novo coronavírus SARS-CoV-2, detectado pela primeira vez em 2019, na cidade chinesa de Wuhan, e a crise sanitária associada, tornaram-se rapidamente as maiores preocupações globais para o início da década. Caracterizada por ser uma doença extremamente infecciosa, característica que se agravou ainda mais após o surgimento de novas mutações e variantes, a doença já tomou mais de 4,55 milhões de vidas em todo o globo. Mais de meio milhão destas mortes vieram do Brasil (OUR WORLD IN DATA, 2021), país para o qual o aplicativo foi idealizado.

Muito embora os índices de vacinação estejam apontando para o crescimento no número de vacinados nos últimos meses e para o possível fim, no próximo ano, da crise sanitária no país, o número de casos e de mortes registrados ilustra que os cuidados e todo o empenho no combate à pandemia continua sendo vital para que possamos superar esta fase de



nossa história. Neste sentido, faz-se fundamental a cooperação entre o poder público e a sociedade civil, trazendo novas soluções de tecnologia para o auxílio da vigília e do enfrentamento à doença, no que pese também a importância das contribuições acadêmicas para este embate.

Soluções como a criação de aplicativos para telefones celulares já foram exploradas no Brasil como potenciais aplicações da tecnologia para a disseminação de informação, bem como para facilitar o acesso da população à Saúde. Podemos citar, como exemplo, o aplicativo SaúdeJá Curitiba, já existente desde 2017, que foi utilizado pela prefeitura da cidade de Curitiba para informar à população datas de agendamento para a aplicação das vacinas para cada um dos habitantes cadastrados, tanto para a primeira e para a segunda dose, também incluindo carteira de vacinação com informações sobre todas as outras vacinas já tomadas ou pendentes (ICI Curitiba, 2021). Tal abordagem se provou um sucesso justamente pelo grande número de pessoas que utilizam em seu dia-a-dia um smartphone conectado à internet. De acordo com a Pesquisa Nacional por Amostra de Domicílios (PNAD) de 2019, pesquisa realizada pelo IBGE, 78,3% dos habitantes do país faziam a utilização da internet em qualquer lugar. 94% dos domicílios do país possuíam um telefone móvel celular, e 98,6% das pessoas que acessaram a internet utilizam seus celulares para o fazer (IBGE, 2019). A tendência geral é que este número apenas suba, conforme o país se digitalize, em adição ao leilão para a venda de faixas de frequência 5G, a ser realizado pela Anatel em Novembro de 2021 (G1, 2021).

Observa-se, no entanto, que mesmo com estes esforços, ainda há muito em que a aplicação de tecnologias similares poderia contribuir no combate à crise sanitária, em especial quando tratamos sobre o espalhamento de informações. Justamente porque o acesso à informação se tornou tão facilitado, em virtude do grande uso de aparelhos com conexão à rede, também se facilitou a disseminação de informações enganosas e incorretas (chamadas atualmente por fake news). Pesquisas indicam que cerca de 110 milhões de brasileiros são afetados por e acreditam em fake news sobre o novo coronavírus (GALHARDI, 2020). Também se observa que um grande número destas se refere a supostas curas caseiras (20%), métodos caseiros de prevenção (65%), e ainda algumas se referiam a golpes (10,7%) (GALHARDI, 2020). Não se observam ainda resultados concretos da aplicação de tecnologias semelhantes ao aplicativo para combater essa problemática, apesar do sucesso percebido em outros aspectos.



Propomos, portanto, a criação de um aplicativo para smartphone, intitulado “XôCovid”, feito com o propósito de disponibilizar informações corretas sobre o vírus e a doença para os seus usuários, bem como recomendações e orientações de ações a serem tomadas no caso de suspeita de infecção, através de coleta de dados realizada por meio de formulários contendo possíveis sintomas da doença. O aplicativo se diferencia das atuais propostas existentes ao permitir que o usuário mais facilmente acompanhe a evolução dos sintomas no caso da suspeita de infecção, ao salvar um histórico com todos os sintomas relatados em formulário.

1. DESENVOLVIMENTO

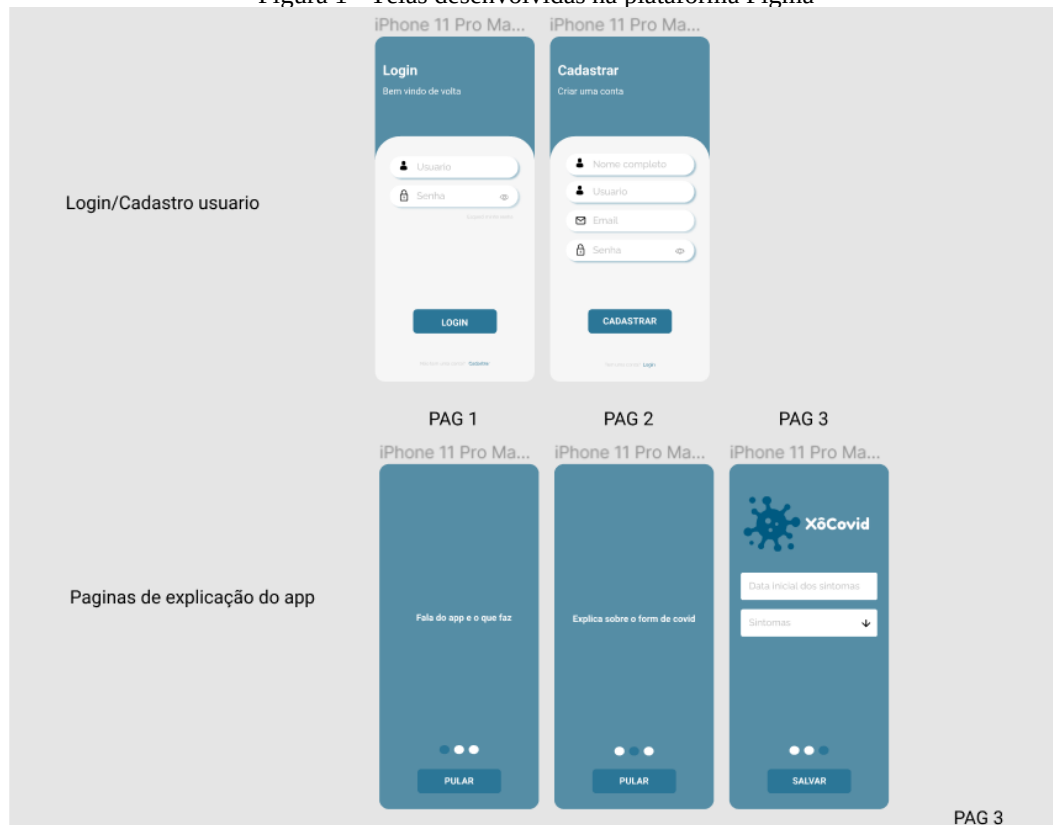
1.1 MÉTODOS

O método seguido para a construção do aplicativo proposto passou primeiramente por uma etapa de elaboração das telas da interface em conjunção à elaboração dos requisitos. Ambas as tarefas foram feitas de forma conjunta por todos os membros da equipe, após pesquisa por outras aplicações que já realizavam funções relacionadas ao combate à pandemia. A opção por unir os dois processos em simultâneo se deu pela maior facilidade de visualizar as necessidades para as funções e interações do programa com um auxílio visual, e muitos dos requisitos, que de outra forma surgiriam em etapas posteriores, já foram constatados logo no levantamento inicial, possibilitando um melhor planejamento para a estrutura e construção do código e interação entre os diferentes elementos envolvidos.

Optou-se pelo uso da plataforma Figma para a elaboração da interface. Trata-se de uma plataforma para desenvolvimento de design de interface (UI), frequentemente usada para a criação de interfaces em programas mobile e web, e que oferece a possibilidade de múltiplos indivíduos em uma mesma equipe alterarem o design e compartilharem suas ideias. Primeiro, criou-se rascunhos para os layouts de páginas mais importantes e comuns à maioria dos aplicativos mobile, como tela de início, cadastro, login, etc. A partir de cada tela criada, então, foram elaborados requisitos de informações que a interface deveria apresentar ao usuário naquela tela, ações possíveis de se realizar a partir das mesmas, e outras telas que o usuário deveria ser capaz de acessar a partir desta. Na Figura 1 observa-se a visualização das telas na plataforma.



Figura 1 - Telas desenvolvidas na plataforma Figma



Fonte: Autores (2021)

Com essa etapa finalizada, iniciou-se a pesquisa sobre os passos envolvidos na construção de um *software* para uma aplicação mobile, elencando os elementos necessários. Constatou-se que, além das interfaces, cujo rascunho e fluxo geral já estava pronto, seria necessário um banco de dados capaz de manter informações importantes como registro de usuário, incluindo nome, login e senha, os sintomas relatados pelo usuário para a composição de um histórico, e também um classificador capaz de receber os dados desses sintomas e enviar para o usuário os textos adequados. Todos esses processos podem ser gerenciados desenvolvendo Interfaces de Programação de Aplicações (API) próprias, que permitem ainda a integração com a interface.

1.1.1 Lista de Requisitos Funcionais

O Gerenciamento de Requisitos é a etapa inicial no processo de desenvolvimento de software, nela são determinadas as funcionalidades e os perfis desejados por clientes e usuários do sistema, de modo a prevenir erros de projeto (KERR, 2016).



Requisitos em um software se dividem entre requisitos funcionais e não funcionais. Requisitos funcionais se referem a todas as necessidades que o software deve atender, ou seja, trata de suas funções e ações que podem ser tomadas. A listagem de requisitos funcionais feita e seguida para a conclusão do projeto segue abaixo, separada em tópicos.

1.1.1.1 Cadastro de Usuário

O aplicativo deverá ter um cadastro de usuário, com preenchimento de:

- Email;
- Nome;
- Nome de usuário;
- Senha;
- Capacidade de colocar uma foto.

1.1.1.2 Login de Usuário

O aplicativo deverá ter a opção de fazer login caso já tenha uma conta cadastrada no *app*, com preenchimento de:

- Nome de usuário;
- Senha;
- Recuperação de senha.

1.1.1.3 Formulário

O aplicativo deverá ter uma tela de formulário, que será mostrada após o cadastro, com alguns campos onde o usuário poderá preencher com informações de acordo com seus sintomas, cujos campos são:

- Lista de sintomas;
- Data inicial dos sintomas.

1.1.1.4 Tela Inicial



O aplicativo deverá ter uma tela de início, exibida após o login. Nesta tela, serão apresentadas:

- Recomendações filtradas com os sintomas relatados;
- Ícone de acesso ao histórico com os diferentes formulários;
- Ícone de acesso à tela de preenchimento de usuário;
- *Drawer* expandindo menu lateral com abas e navegação para Tela Inicial, Perfil, Configurações e Dúvidas.

1.1.1.5 Tela de Perfil

O *app* deverá ter uma tela de perfil onde o usuário poderá ver suas informações de conta, dentre elas:

- Nessa tela será apresentado a foto do usuário e nome de usuário;
- Possuirá um botão de alterar perfil, para que ele possa trocar sua foto ou nome de usuário;
- Mostrará as informações preenchidas no formulário de COVID-19.

1.1.1.6 Tela de Configurações

O aplicativo deverá ter opções de configuração, sendo elas:

- Configurar conta, podendo alterar senha ou nome de usuário;
- Configuração de notificações;
- Termos de privacidade e segurança.

1.1.1.7 Tela de Dúvidas

O aplicativo deverá ter uma área destinada à resposta de perguntas que os usuários possam ter durante seu uso, contendo as perguntas:

- Como faço para alterar meus dados de perfil?
- Como faço para alterar a senha da minha conta?



- Como faço para atualizar meu formulário de covid-19?
- Para que serve o formulário de covid-19?
- O que seria a tela de ajuda?

1.1.1 Lista de Requisitos Não Funcionais

Os requisitos não funcionais, por sua vez, descrevem o modo como a equipe de desenvolvedores pretende tornar o software factível, desde comportamento até fatores externos ao funcionamento, como a política de proteção de dados. A listagem de requisitos não funcionais feita segue abaixo, em tópicos.

- Usabilidade: O sistema terá uma forma de apresentação de dados e informações de fácil entendimento, ou seja, que permita ao usuário a utilização intuitiva desde o primeiro acesso;
- Eficiência: O sistema terá o processamento de dados com bastante eficiência, para que os usuários possam acessar as informações com agilidade;
- Manutenção: Será um sistema modularizado, ou seja, cada item terá sua funcionalidade separada, favorecendo a manutenção no sistema;
- Segurança: O sistema terá um procedimento de cadastro caso o usuário ainda não tenha uma conta, e aqueles que já possuem conta, devem utilizá-la para acessar o aplicativo. Ou seja, usuários que não estejam cadastrados no aplicativo não poderão acessá-lo.

1.1.2 Plataforma e linguagem

A escolha da plataforma passou por uma análise baseada em alguns critérios pré-definidos pela equipe, entre eles: a facilidade de uso, rápido aprendizado, disponibilidade de bibliotecas e soluções para suporte do desenvolvimento, e ampla documentação. Tais necessidades deram-se, primeiramente, por conta da relativa inexperiência da equipe em desenvolvimento mobile e, portanto, muito tempo de desenvolvimento poderia ser desperdiçado buscando solucionar problemas provocados por um mau entendimento das plataformas e linguagens usadas. Perda essa que é significativamente mitigada quando utiliza-se um software bem documentado e com amplo suporte disponível, pelos próprios criadores ou



pela comunidade de programadores que o utilizam. A IDE (Integrated Development Environment) de escolha foi o Android Studio, usando o kit de desenvolvimento Flutter.

O Android Studio é um Ambiente de Desenvolvimento Integrado (*Integrated Development Environment*, ou IDE) voltado ao desenvolvimento para aparelhos de sistema operacional Android, disponibilizado de forma gratuita sob a licença Apache 2.0. Tornou-se a principal IDE de desenvolvimento nativo para Android da Google, substituindo o Eclipse Android Development Tools (ADT) (EXAME, 2014). A preferência particular pelo Android Studio se dá pela fácil disponibilidade de plugins para desenvolvimento em Flutter, o *framework* escolhido para a programação das interfaces do aplicativo.

O Flutter, por sua vez, é um kit de desenvolvimento UI criado pela Google, frequentemente utilizado para criar aplicações para Android, iOS, Linux, Mac, Windows e Google Fuchsia (ARS TECHNICA, 2018). Utiliza primariamente a linguagem Dart, que é orientada a objetos e com sintaxe similar ao C, e se distingue por transformar todo seu código-fonte em código nativo diretamente. Sendo assim, programas desenvolvidos com auxílio do Flutter possuem acesso direto a recursos nativos do sistema, proporcionando uma maior performance, e a possibilidade de compilar nativamente para mais de uma plataforma. A arquitetura de aplicativos programados em Flutter é estruturada primariamente em *widgets* que, combinados, compõem a interface (FLUTTER, 2017). Pode-se fazer uma analogia com peças de montar, sendo que os *widgets* são as peças, e o aplicativo, o brinquedo que é montado com elas quando todas estão encaixadas.

O fato da linguagem primária Dart ser orientada a objetos, foi aproveitado pela equipe para aplicar conceitos como a abstração do código, que economiza tempo e linhas de programação. Assim, um dado elemento comum a várias telas só precisa ser definido uma única vez, e então chamado dentro de cada tela com o contexto apropriado, trazendo consigo todos os métodos a si associados que não precisam ser mais repetidos toda vez que se cria uma nova tela. Isso também torna o código mais modular, e mais simples de se modificar.

1.1.3 Banco de Dados

A criação de um banco de dados, bem como as regras que o sistema seguiria para interpretação dos mesmos, seria a parte mais importante para a implementação do aplicativo. Inicialmente, considerou-se duas plataformas para a hospedagem do banco de dados e criação



das API's necessárias: Hasura e Supabase. Ambos funcionam como bancos de dados relacionais, oferecendo acesso à informação de forma consistente, no formato de tabelas que podem também ser relacionadas entre si.

Após alguns testes de utilização, decidiu-se pelo uso do Supabase, ilustrado na Figura 2, que possui sua própria biblioteca de funções em Flutter, intitulada “supabase”. A integração com o aplicativo é realizada a partir de uma URL (Uniform Resource Locator) que acessa os dados e realiza pedidos de acesso e de registro ao banco de dados, e uma chave de autenticação, que é verificada pelo mesmo, garantindo que somente dados pertinentes sejam acessados. Uma entrada que não passe pela autenticação não será permitida.

Figura 2 - Integração com Supabase

```
import 'package:flutter/material.dart';
import 'package:xo_covid/app/app_colors.dart';
import 'package:xo_covid/login/login_page.dart';

import 'package:supabase/supabase.dart';

const supabaseUrl = 'https://wrklrnkppwhwteqpgqpf.supabase.co';
const supabaseKey = 'eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJyY2x1IjoiaW50MTM2MCwiZm90TUxNzg1MzYwYmF0LmEgEzEzZca9b0';
final supabaseClient = SupabaseClient(supabaseUrl, supabaseKey);

class AppWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'XoCovid',
      theme: ThemeData(primaryColor: AppColors.backgroundSecondary),
      home: LoginPage(),
    ); // MaterialApp
  }
}
```

Fonte: Autores.

O uso do Supabase se provou simples, e passou-se a utilizar a plataforma para registro de dados do usuário em cadastro, autenticação do usuário durante login, registro de sintomas e histórico de sintomas no banco de dados, e obtenção das recomendações, como é possível observar na Figura 3.

Figura 3 - Obtenção da lista de recomendações

```
Future<List<RecomendacoesModel>> getRecomendacoes() async {
  final response = await supabaseClient.from('recomendacoes').select().order('nome', ascending: true).execute();

  final dataList = response.data as List;
  return dataList.map((map) => RecomendacoesModel.fromJson(map)).toList();
}
```

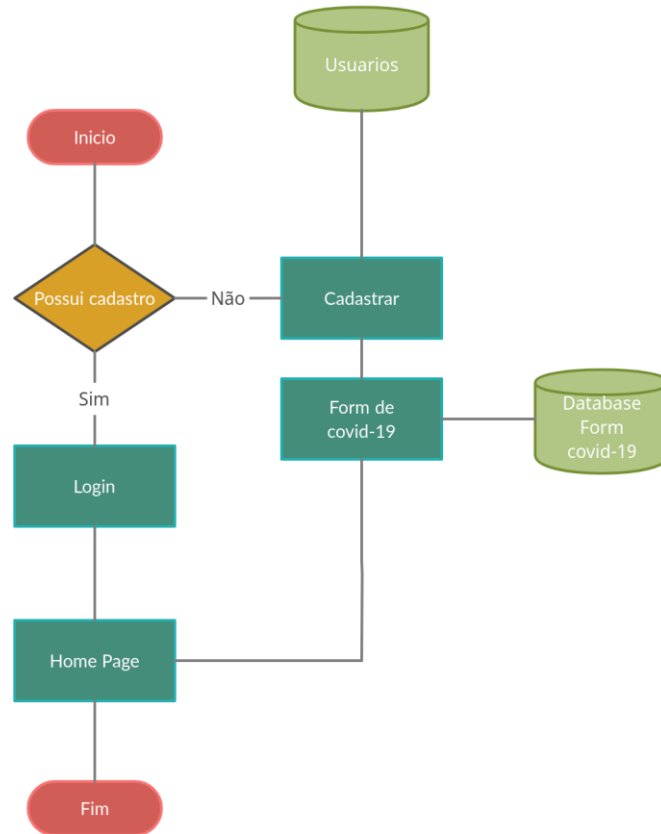
Fonte: Autores.

1.1.4 Fluxograma

Na Figura 4 observa-se o fluxograma que ilustra a usabilidade do aplicativo.



Figura 4 - Fluxograma do aplicativo



Fonte: Autores (2021)

2. RESULTADOS

Ao fim do desenvolvimento, a ideia foi implementada com sucesso, resultando em um aplicativo com diversas telas e de fácil interação. As principais telas são apresentadas nas Figuras 5 a 11.



Figura 5 - (a) Página de login e (b) cadastro.

Login
Bem vindo de volta

Usuário

Senha

[Esqueci minha senha](#)

LOGIN

Não tem uma conta? [Cadastrar](#)

Cadastrar
Criar uma conta

Nome completo

Usuário

Email

Senha

CADASTRAR

Tem uma conta? [Login](#)

Fonte: Autores.

A página de login, representada na Figura 5, contém dois campos digitáveis e faz a autenticação para acesso do usuário. A opção “Cadastrar” leva para a tela onde serão requeridos o Nome Completo e o e-mail. Nela, a opção “Login” leva o usuário de volta para a tela de login.



Figura 6 (a) e (b) - Introdução de funcionamento



Fonte: Autores.

Caso o usuário não tenha ainda uma conta, após ele terminar o cadastro na tela de cadastro, aparecerão as telas de introdução ao funcionamento do aplicativo, representadas na Figura 6, que explicam sua função e direcionam o usuário para a tela de preenchimento do formulário.



Figura 7 (a) e (b) - Telas de Formulário

12:33 12:32

Data inicial dos sintomas 07/10/2021

Selecione os sintomas Seleccione os sintomas

SALVAR SALVAR

Sintomas

Febre Tosse Seca Cansaço

Perda de Paladar (Parcial)

Perda de Olfato (Parcial)

Perda de Paladar (Completa)

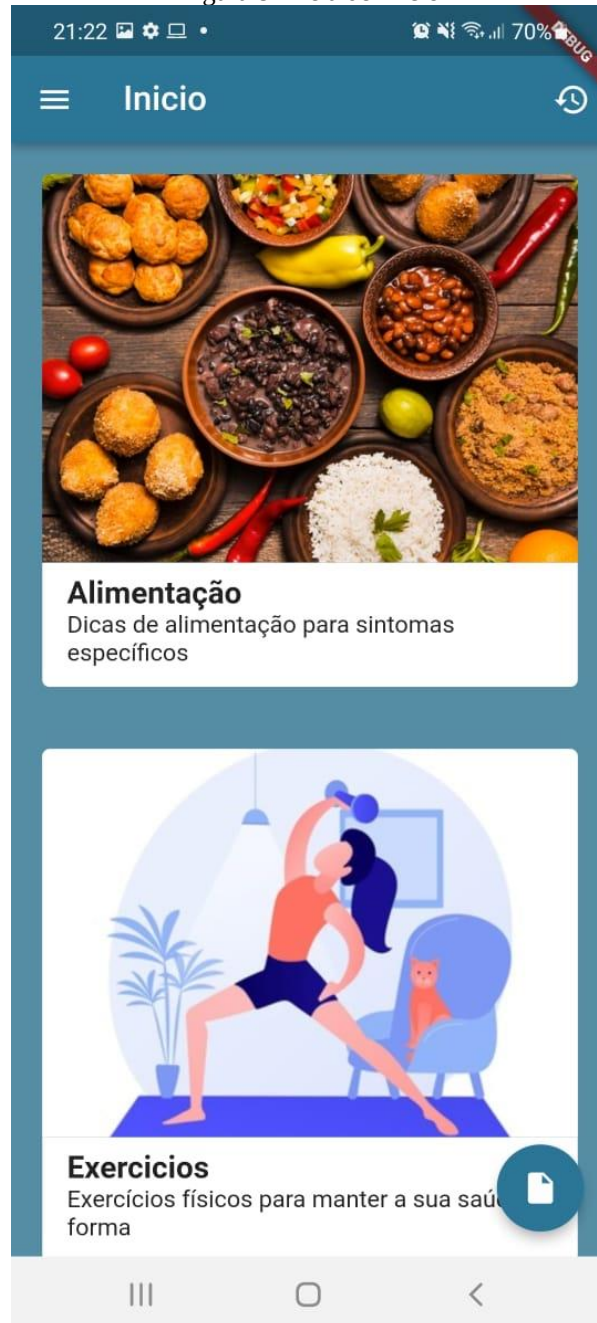
CANCEL OK

Fonte: Autores.

A tela de formulário, apresentada na Figura 7, possui um campo digitável, onde o usuário insere a data em que os sintomas começaram a ser notados. O aplicativo automaticamente formata para DD/MM/AAAA. Logo abaixo, um botão expande para uma lista que contém sintomas, o usuário pode selecioná-los ou então digitar na aba de busca. Tocando no botão “Salvar”, os dados são enviados e o usuário é direcionado para a tela de início.



Figura 8 - Tela de Início



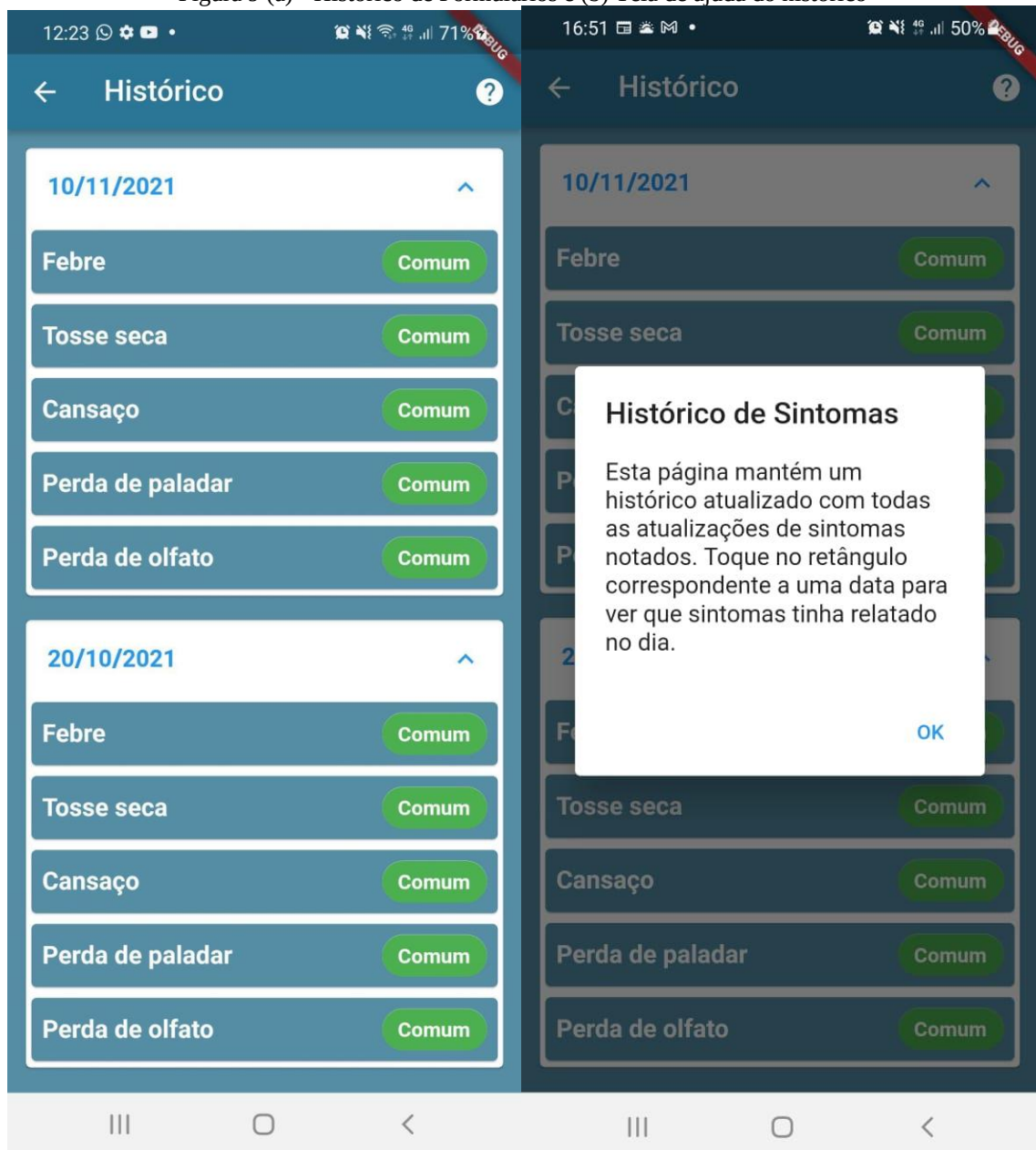
Fonte: Autores.

A tela de início, observada na Figura 8, é o ponto a partir do qual o usuário seleciona e interage com a maior parte das funcionalidades do aplicativo. Os grandes *widgets* no meio da tela, acompanhados de imagens, podem ser acessados para expandir as recomendações que foram filtradas de acordo com os sintomas relatados. O botão no canto inferior direito, com um ícone de folha, direciona o usuário para a tela de formulário novamente, caso o usuário deseje atualizar as informações. O ícone de relógio no canto superior direito leva ao histórico com



todos os formulários preenchidos. Por fim, o ícone no canto superior esquerdo abre o *drawer*, que dá acesso a mais opções.

Figura 9 (a) - Histórico de Formulários e (b) Tela de ajuda do histórico

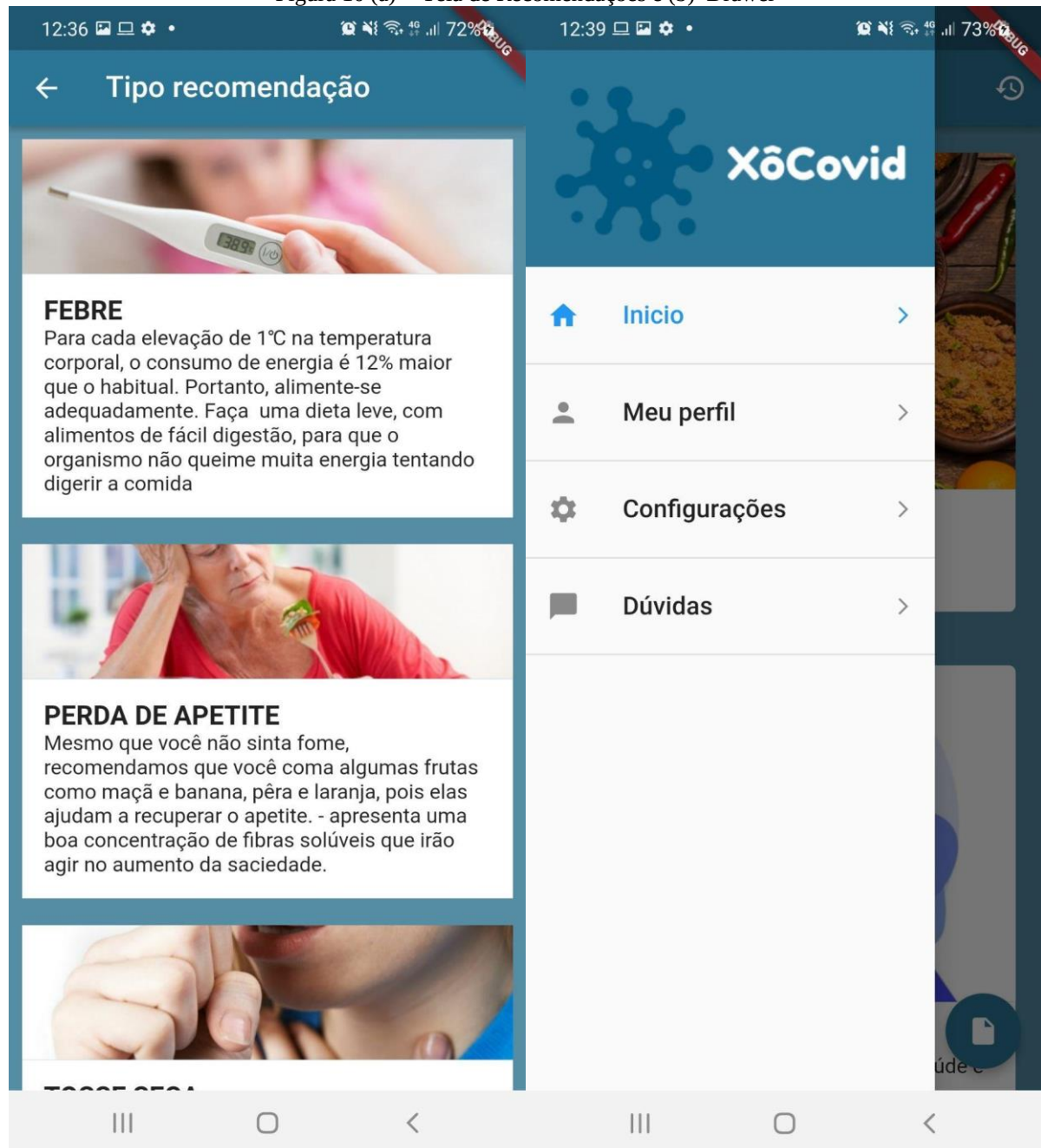


Fonte: Autores.

A tela de histórico de formulários, presente na Figura 9 (a), contém as informações salvas de todos os formulários preenchidos pelo usuário, organizados cronologicamente. Ao tocar no botão, o widget expande e mostra a lista com os sintomas, e também uma classificação do quão comuns eles são. O ícone de interrogação no canto superior direito abre uma tela de ajuda, mostrada na Figura 9 (b), explicando a função da tela.



Figura 10 (a) - Tela de Recomendações e (b) Drawer



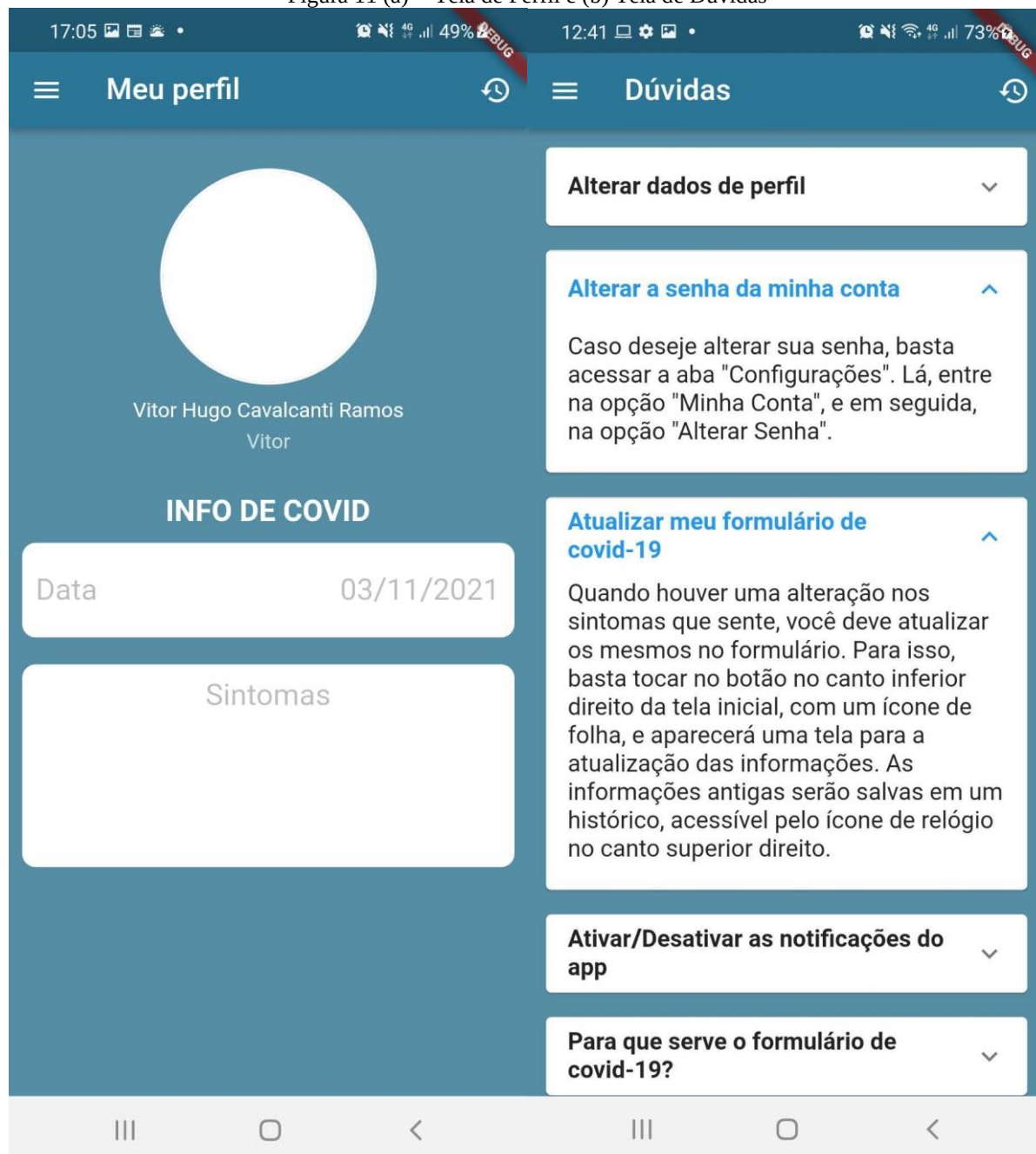
Fonte: Autores.

A tela de recomendações, mostrada na Figura 10 (a), exibe uma lista de *widgets* exclusivos para leitura. Cada um destes exibe um texto, com uma recomendação em específico, retirada de fontes fidedignas, destinada a algum sintoma relatado pelo usuário.

O *drawer*, mostrado na Figura 10 (b), abre um menu lateral que permite alternar entre quatro telas: a Tela de Início, a Tela de Perfil, a Tela de Configurações, e a Tela de Dúvidas.



Figura 11 (a) - Tela de Perfil e (b) Tela de Dúvidas



Fonte: Autores.

A tela de perfil, mostrada na Figura 11 (a), contém um campo destinado para a exibição de uma imagem que o usuário pode incluir, além de seu nome completo, nome de usuário, data de início dos sintomas, e a lista de sintomas em si.

A tela de dúvidas, presente na Figura 11 (b), é formada por vários widgets que se expandem ao toque. Cada um contém um texto explicando alguma funcionalidade ou dúvida que o usuário possa ter.



CONSIDERAÇÕES FINAIS

Concluimos que os objetivos do projeto foram de fato atingidos de forma satisfatória, segundo todos os requisitos previamente levantados. Os métodos seguidos se mostraram suficientes para a solução da problemática inicialmente proposta. Acredita-se que o potencial demonstrado pela solução que foi implementada é promissor, não somente pela sua capacidade de auxiliar as pessoas em uma situação delicada de crise sanitária, mas também por sua flexibilidade e adaptabilidade. O caráter pessoal da proposta possibilita que ela seja ampliada e se mantenha relevante mesmo após o fim da pandemia da COVID-19 para ser usada no combate a outras doenças, e outras possibilidades como a adoção do aplicativo em grande escala pelo sistema de saúde, com a adição de incrementos futuros.

REFERÊNCIAS

AMADEO, Ron. Google starts a push for cross-platform app development with Flutter SDK. **Ars Technica**, 27 de fev. de 2018. Tecnologia. Disponível em: <<https://arstechnica.com/gadgets/2018/02/google-starts-a-push-for-cross-platform-app-development-with-flutter-sdk/>>. Acesso em: 16, Novembro, 2021.

ANDROID STUDIO: Conheça o Android Studio. Disponível em: <<https://developer.android.com/studio/intro>>. Acesso em: 27, Outubro, 2021.

APLICATIVO realiza mais de 20 mil agendamentos para vacinação em Curitiba. **Instituto das Cidades Inteligentes**, Curitiba, 08, Fevereiro, 2021. Disponível em: <<https://www.ici.curitiba.org.br/noticias/aplicativo-realiza-mais-de-20-mil-agendamentos-para-vacinacao-em-curitiba/22691>>. Acesso em: 27, Outubro, 2021.

FIGMA: the collaborative interface design tool. Disponível em: <<https://www.figma.com/>>. Acesso em: 27, Outubro, 2021.

FLUTTER: Design Beautiful Apps. Disponível em: <<https://flutter.dev>>. Acesso em: 27, Outubro, 2021.

GALHARDI, Cláudia Pereira et al. **Fato ou Fake? Uma análise da desinformação frente à pandemia da Covid-19 no Brasil**. *Ciência & Saúde Coletiva*, v. 25, p. 4201-4210, 2020.

GUSMÃO, Gustavo. Google lança versão 1.0 do IDE de código aberto Android Studio. **Exame**, 09, de dez. de 2014. Tecnologia. Disponível em: <<https://exame.com/tecnologia/google-lanca-versao-1-0-do-ide-de-codigo-aberto-android-studio/>>. Acesso em: 16, Novembro, 2021.



INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Acesso à internet e à televisão e posse de telefone móvel celular para uso pessoal.** 2019 / IBGE, Coordenação de Trabalho e Rendimento. 2019.

INTRODUCTION to widgets. **FLUTTER.**, 2017. Disponível em: <<https://docs.flutter.dev/development/ui/widgets-intro>>. Acesso em: 17 de nov. de 2021.

KERR, Eduardo Santos. **Gerenciamento de Requisitos.** São Paulo: Pearson. 2016. Ebook

OUR WORLD IN DATA. **Coronavirus Pandemic (COVID-19) – the data** [Internet].Disponível em: <<https://ourworldindata.org/coronavirus-data>>. Acesso em: 27, Outubro, 2021.

SANTANA, Jessica. Anatel recebe 15 propostas para o leilão do 5G, marcado para 4 de novembro. **G1**, Brasília, 27, Outubro, 2021. Economia. Disponível em: <<https://g1.globo.com/economia/noticia/2021/10/27/anatel-recebe-15-propostas-para-o-leilao-do-5g-marcado-para-4-de-novembro.ghtml>>. Acesso em: 27, Outubro, 2021.

Recebido em 22/11/2021.
Aprovado em 31/05/2022.