



Colégio Técnico de Campinas
Universidade Estadual de Campinas
Departamento de Processamento de Dados

Relatório Científico

Ajuda UBS

Software de denúncias e informações de Centros de Saúde

Autor: Fabrício Onofre Rezende de Camargo - 20130

Autora: Ligia Keiko Carvalho - 20143

Orientadora: Simone Pierini Facini Rocha

Coorientador: Andréia Cristina de Souza

Campinas

2022

Fabício Onofre Rezende de Camargo

Ligia Keiko Carvalho

Ajuda UBS

Software de denúncias e informações de Centros de Saúde

Orientadora: Simone Pierini Facini Rocha

Coorientador: Andréia Cristina de Souza

Colégio Técnico de Campinas - Unicamp

R. Culto à Ciência, 177 - Botafogo,

Campinas - SP, 13087-261

Campinas - SP

2022

1. Introdução

As ouvidorias públicas são consideradas instrumentos da democracia, na medida em que possibilitam à sociedade o exercício do direito à participação e ao controle da administração pública, bem como o auxílio na aplicação da norma que exige que todos os atos públicos tenham transparência em todas as fases de sua execução, constituindo-se em um canal de interlocução entre o cidadão e o poder público, segundo a Cartilha de Ouvidoria Municipal (2022). Elas estão presentes no Brasil desde o período colonial. Entretanto, é somente com a redemocratização e o fim da ditadura militar que encontram espaço para crescimento (Moraes et al., 2010; Ouvidoria Geral da União, 2012). A discussão em torno da criação de um Estado Democrático de Direito verdadeiramente popular resultou na Constituição Federal Brasileira de 1988, nossa Carta Magna, na qual expressa em seu artigo 37, a livre manifestação do cidadão em questionar as condutas da administração pública, e em seu parágrafo 1º a exigibilidade de transparência de todos os atos públicos.

Atualmente, existem cerca de 2.602 ouvidorias para o exercício da cidadania, em vários segmentos da administração pública, distribuídas nas esferas Federal, Estadual e Municipal (CGU, 2022). Além, de que mais de 80% da população brasileira vive em áreas urbanas, o que equivale aos níveis de urbanização dos países desenvolvidos. Com isso, a maioria das cidades brasileiras estão se urbanizando, tornando necessário planejamento da gestão pública, para que assim, a qualidade de vida e o saneamento básico do local se mantenham a altura da oferecida por Países já urbanizados, uma vez que a cidade se encontra em processo de urbanização, surgem diversos problemas, desde áreas que envolvem a cultura até o meio-ambiente, abrangendo diferentes questões econômicas e de infraestrutura, as quais a gestão da cidade deve enfrentar com prioridade máxima, com o intuito de obter um ambiente social desenvolvido para os seus habitantes (CERQUEIRA, 2019).

O segmento da Esfera Municipal chama-se Ouvidoria Municipal, com o intuito de garantir a democracia participativa, fazendo com que as denúncias, críticas ou sugestões dos cidadãos sejam acolhidas pela administração municipal e colocadas em prática o mais rapidamente possível. Porém, as cidades enfrentam dificuldades em resolver os problemas recorrentes dos cidadãos, não sendo necessariamente a falta de vontade política ou de recursos públicos o maior problema, mas sim a falta de projetos de médio e longo prazos aliados a identificação das pendências da população. Portanto, é necessário se ter uma boa gestão e planejamento dos dados adquiridos pela Ouvidoria Municipal, permitindo mapear a qualidade e satisfação do serviço prestado na perspectiva da comunidade municipal (LYRA, 2004).

Avaliando todas as áreas de atuação em que o Estado atua para atender a população em geral, a área da saúde é na nossa avaliação pessoal, a mais importante, sendo a que mais necessita de projetos e investimentos por parte do Estado. Além disso, uma pesquisa realizada pelo instituto Datafolha a pedido do

Conselho Federal de Medicina (CFM) aponta que 93% dos eleitores brasileiros avaliam os serviços público e privado de saúde como péssimos, ruins ou regulares.

Desse modo, um canal de denúncias no setor da saúde municipal é primordial para o fortalecimento de uma cultura ética e transparente na gestão da saúde, trazendo mais segurança aos pacientes e qualidade aos serviços prestados à população em geral. Sabendo disso, este projeto visa desenvolver uma aplicação mobile que atenda aos cidadãos e a prefeitura, de forma que esse sistema permitirá o controle das manifestações públicas (Dúvidas, Sugestões, Elogios e Denúncias) enviadas pelos pacientes dos centros de saúde, a fim de ampliar sua eficiência e utilidade quanto a análise dos dados fornecidos, promovendo uma melhora no atendimento aos usuários do sistema de saúde municipal.

2. Problema

Como desenvolver uma aplicação capaz de aumentar a eficiência das Ouvidorias Municipais, em relação ao atendimento dos pacientes das Unidades Básicas de Saúde (UBS)?

3. Objetivos

3.1. Objetivo geral

O presente projeto tem como objetivo desenvolver um sistema que consiga facilitar para os órgãos responsáveis pelas UBSs a visualização e tratamento das manifestações públicas (reclamações, dúvidas, sugestões, elogios e denúncias) feitas pela população em relação ao uso dos serviços da saúde municipal.

3.2. Objetivos específicos

Os objetivos específicos desta pesquisa são:

- Estudar, analisar e escolher algoritmos de análise de dados;
- Implementar um algoritmo capaz de analisar as informações acerca das UBS de forma eficiente;
- Utilizar recursos, como *APIs* e *Frameworks* que mostre os dados analisados forma eficiente;
- Desenvolver um aplicativo mobile destinado aos cidadãos, para o cadastro e acompanhamento das manifestações;
- Desenvolver sistema de tratamento e mostra de dados.
- Testar e avaliar o uso do sistema em casos reais de uso.

4. Justificativa

A Constituição Federativa do Brasil de 1988, assegurou que “Saúde é um direito de todos e dever do Estado”. Desta maneira, foi criado o Sistema Único de Saúde (SUS), um grande sistema de saúde público que abrange atendimentos de pequena, média e grande complexidades, garantindo o acesso integral, universal e gratuito à saúde pública para toda a população brasileira. Nesse complexo sistema, a Unidade Básica de Saúde (UBS) é o primeiro contato do cidadão com o SUS, e tem como função primordial promover a atenção integral à saúde básica dos indivíduos, estando presente em locais de fácil acesso, o que a deixa próxima do cotidiano da população, desempenhando papel fundamental na garantia de acesso à saúde gratuita e de qualidade.

Assim, como qualquer parte do setor público, as UBS enfrentam grandes desafios, sendo eles, administrativos, financeiros, de gestão e relacionados a qualificação e o bom atendimento na Atenção Básica da Saúde.

Para dificultar mais ainda o atendimento ao público, os gestores municipais possuem canais de denúncias e comunicação com o cidadão descentralizados, dificultando a solução dessas denúncias e consequentemente prejudicando a qualidade do atendimento aos pacientes. Os principais desafios são relacionados ao financiamento insuficiente da Atenção Básica de Saúde, infraestrutura das UBS inadequadas, baixo uso das informações disponíveis para a tomada de decisões na gestão, necessidade de ampliar o acesso, reduzindo o tempo de espera, para assim melhorar a qualidade dos serviços, incluindo o acolhimento, resolubilidade e longitudinalidade dos cuidados (SCIELOSP, 2015).

Atualmente, as tecnologias de informação (TI) tem cada vez mais se integrado com as diversas áreas do setor público, devido a agilidade, transparência e qualidade dos serviços e informações que são gerados a partir da Tecnologia da Informação, de modo que essas informações devem estar alinhadas com os negócios do governo e a comunidade, sendo focadas na gestão de planejamentos e ações de melhorias constantes. As Ouvidorias Municipais têm sido cada vez mais modernizadas, a fim de promover uma melhora crescente nos atendimentos a população, porém o recebimento de informações através das ouvidorias municipais continua sendo muito burocrático, uma vez que o encerramento do protocolo aberto pelo cidadão em um desses órgãos, dura em média 58 dias, podendo chegar a até 150 dias para seu encerramento, sem que isso signifique que a demanda será resolvida (MANUAL, 2019).

Segundo os dados do site (RESOLVEU?, 2022) da Controladoria-Geral da União, cerca de 44% da população que se manifestou em uma Ouvidoria, diz que sua demanda não foi resolvida. Assim sendo, fica nítido o descontentamento e a insatisfação dos cidadãos com o serviço público, visto que cerca de 45% dos atendidos em uma Ouvidoria dizem estar insatisfeitos ou muito insatisfeitos com o atendimento prestado.

A organização e o planejamento público devem ser constante avaliados e modernizados, dado a velocidade de inovação tecnológica e de informações que a

globalização proporciona, desse modo, em torno de 60% dos prefeitos de cidades de pequeno e médio porte no Brasil concordam que a desburocratização e a flexibilização dos processos administrativos são os pilares basilares da inovação na gestão pública municipal, diz a pesquisa Desafios para Inovação na Gestão Municipal do Instituto Arapyaú & CDE (2018).

Portanto, para um bom desempenho dos serviços prestados pelas UBS à comunidade, é necessário que as informações recebidas diariamente pelas Ouvidorias Municipais sejam gerenciadas adequadamente, permitindo agilidade, segurança e integridade dos dados coletados, e isso somente é possível com o uso de um sistema que utilize métodos que visam a otimização e dinamização dos processos, zelando pela segurança, guarda e qualidade dos dados produzidos e recebidos pelas ouvidorias das prefeituras, dando novas funcionalidades ao sistema, como a análise gráfica dos dados e serviços oferecidos pelas UBS.

5. Referencial Teórico

Visando utilizar as limitações das UBS no Brasil como inspiração para o desenvolvimento do projeto, a aplicação a ser desenvolvida visa utilizar a união de ferramentas inovadoras como frameworks e APIs auxiliadas a algoritmos de tratamento de dados para possibilitar uma melhor administração e tratamento de denúncias.

5.1. Unidades Básicas de Saúde (UBS)

As unidades básicas de saúde são locais que oferecem serviços de saúde gratuitos e básicos relacionados à Pediatria, Ginecologia, Clínica Geral, Enfermagem e Odontologia. As Unidades Básicas de Saúde (UBS) fazem parte da Política Nacional de Urgência e Emergência, feita pelo Ministério da Saúde em 2003, estruturando e organizando a rede de urgência e emergência no país, para integrar a atenção às urgências, além de ser a porta de entrada para o atendimento do paciente no SUS (Sistema Único de Saúde) (Sociedade Paulista Para o Desenvolvimento da Medicina, 2022).

Cada UBS é responsável pela assistência à saúde de uma população definida, assim, cada cidadão tem agora uma UBS de referência a partir do seu endereço de moradia. Cada Unidade Básica tem equipes de saúde da família (ESF) que são compostas por médico, enfermeiro, técnico de enfermagem e agentes comunitários de saúde.

5.2. Ouvidorias Municipais

As ouvidorias municipais tem como objetivo atender os manifestos feitos pela população, a fim de melhorar a qualidade dos serviços públicos. Algumas das funções desempenhadas pelas ouvidorias, segundo a Cartilha de Ouvidoria Municipal (2022) são:

- Reconhecer os cidadãos como sujeitos de direito, sem qualquer distinção;
- Ouvir e compreender as diferentes formas de manifestações dos cidadãos;
- Dar tratamento adequado às demandas apresentadas pelos cidadãos, usando linguagem clara para explicar seus direitos e as formas de obtê-los.
- Caracterizar corretamente as situações e seus contextos, explicitando as consequências sobre cada caso concreto de sua demanda;

Demonstrar os resultados produzidos em razão da participação dos cidadãos. Ou seja, deve utilizar o conteúdo das solicitações para sugerir mudanças nos processos na administração pública, contribuindo para que os agentes públicos providenciem medidas corretivas.

5.3. Modelo MVC (Model, View, Controller)

Para facilitar o desenvolvimento de uma aplicação, são utilizados alguns padrões de arquitetura de desenvolvimento. O Modelo MVC é o acrônimo de Model-View-Controller (Modelo-Visão-Controle) é um padrão de projeto de software, foi inicialmente visualizado por Trygve Reenskaug nos anos 70 no Xerox Parc. Segundo ele, “o propósito essencial do MVC é fazer a ponte entre o modelo mental do usuário humano e o modelo digital que existe no computador” (POP; ALTAR, 2014).

Sua dinâmica é simples, todas as requisições da aplicação são direcionadas para a camada Controller, que acessa a camada Model para processar a tal requisição, e por fim exibe o resultado da camada View. Esse modelo é focado no reuso de código e a separação de conceitos em três camadas interconectadas, onde a apresentação dos dados e interação dos usuários (front-end) são separados dos métodos que interagem com o banco de dados (back-end).

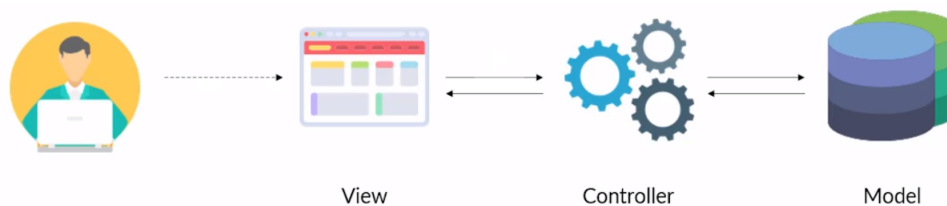


Figura 1. Esquema representando o modelo MVC. Fonte: DevMedia.

5.4. Computação em Nuvem

A Computação em Nuvem ou Cloud Computing é uma tecnologia que usa a conexão da internet para hospedar diferentes tipos de recursos, programas e informações na nuvem. O termo “nuvem” é utilizado para descrever uma rede global de servidores, cada um com uma única função. A nuvem não é uma entidade física, mas uma vasta rede de servidores remotos que são conectados e operam como um único ecossistema, no qual é responsável por armazenar, executar ou gerenciar os arquivos ali armazenados. Em vez de acessar arquivos e dados do local ou de um

PC, é possível acessá-los online, de qualquer dispositivo com acesso à Internet, como computador, tablet ou telefone celular.

Essa tecnologia se tornou uma das principais escolhas de empresas por oferecer rapidez e flexibilidade ao adquirir recursos, custo benefício e segurança (IBM, 2022). Dessa forma, traz diversos benefícios tanto para os desenvolvedores quanto para os usuários de sistemas que usam a Cloud Computing, dentre as vantagens dessa tecnologia, o custo é um ótimo indicador, uma vez que são economizados gastos com compras e manutenções de softwares e hardwares para o armazenamento de dados. Além disso, o alto desempenho, a confiabilidade e a velocidade são outros fatores que indicam os benefícios de utilizar essa tecnologia de ponta (MAGALHÃES, 2018).

5.5. Application Programming Interface (API)

Uma API (Application Programming Interface) é o conjunto de padrões de programação para acesso a um aplicativo de software ou plataforma baseado na WEB. Através das API's, os aplicativos conseguem interagir um com o outro sem intervenção do usuário. De modo geral, ela é composta de uma série de funções acessíveis somente por meio de programação (CANALTECH, 2018). O seu funcionamento se baseia no cliente realizando uma requisição em uma rota (endereço), para acessar alguma informação qualquer, podendo ser passados parâmetros diversos dependendo da rota. Ao ser acionada, a API consulta o servidor e manda a informação que foi obtida de volta para o cliente.

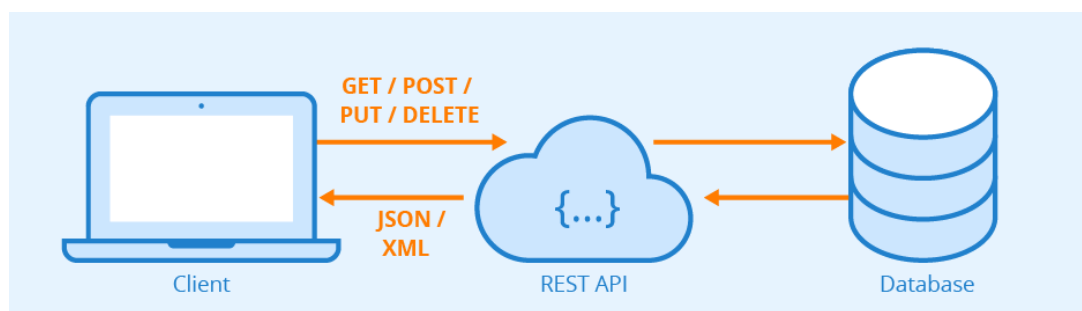


Figura 3. Esquema representando uma API. Fonte: Astera.

5.6. Flutter, um framework híbrido

Antes da criação de frameworks híbridos, como o React Native e o Flutter, as grandes empresas que desenvolviam aplicativos multiplataforma, tinham que desenvolver códigos não-relacionados, com o intuito de alcançar o maior número de usuários possíveis. Dessa forma, era necessário desenvolver um código para cada plataforma mobile que se desejasse atender, como IOS, Android ou Windows Phone.

O Flutter é um framework que utiliza a linguagem Dart da Google para criação de aplicativos móveis híbridos, dos quais podem ser executados em

múltiplas plataformas a partir de uma única base de código. O objetivo desse framework é possibilitar que desenvolvedores possam criar aplicações de alta performance, com uma experiência nativa em diferentes plataformas.

5.7. Testes automatizados

A partir da utilização em massa de softwares e tecnologias provenientes da computação, tornou-se necessário averiguar a qualidade destes produtos (ZHU, HALL, 1997). Para fazer a integração do código de forma segura e eficiente, é necessário criar softwares que testem os componentes especificados, testando a entrada e saída de valores, *pipeline* e comportamento do software. Ao contrário dos testes manuais, os testes automáticos capturam de forma mais eficiente as falhas e futuras exceções.

O desenvolvimento dos testes automatizados deve seguir etapas de processos para ser modelado. A sequência de etapa mais comumente seguidas são:

- I. Seleção da ferramenta de teste;
- II. Definição do escopo de automação;
- III. Desenvolvimento;
- IV. Execução do teste;
- V. Manutenção.

5.7.1. Testes unitários

Dentre os testes automáticos, existem diversos tipos que podem ser utilizados para as mais diversas finalidades. Dentre elas podemos citar os testes unitários, os quais visam analisar a corretude do código nas suas menores partes. Em suma, os testes unitários são testes de tempo de processamento menor e que atuam em partes mais isoladas do software (DIGITÉ, 2022). Podemos notar que em muitos casos os testes unitários são aplicados em funções, analisando o fluxo entre as variáveis e outros elementos presentes na função.

Por serem testes de maior agilidade, os dados de entrada nunca são provenientes de banco de dados ou pelo usuário em tempo de execução. Para testar o código, é necessário criar valores fictícios que estejam declarados dentro do escopo do teste unitário em questão.

5.8. Behavior-Driven Development

Esta técnica de desenvolvimento ágil consiste em integrar as funcionalidades previstas no escopo do projeto focando no comportamento do software. O foco desta técnica está presente na linguagem e as interações de seus componentes no processo de desenvolvimento do software.

Durante o desenvolvimento dos testes baseados em BDD e TDD, são colocados cenários baseados nas funcionalidades previstas no escopo do projeto.

Inicialmente, os testes BDD seguiam histórias baseadas no levantamento de requisitos pelo cliente. Entretanto, nota-se que atualmente diversos frameworks, como por exemplo o framework Chai e Mocha que foram usados neste projeto, foram construídos visando melhorar a interação e fluxo do software a partir da análise do comportamento do produto.

Os cenários inseridos nos testes se baseiam em três blocos principais:

- *Given* - contexto especificado;
- *When* - quando acontecer um evento;
- *Then* - evento esperado a partir de uma interação ou ação no software.

O padrão de escrita destes testes foi introduzido por Dan North em 2003. A padronização da escrita destes tipos de teste facilita não somente a compreensão entre os desenvolvedores mas também entre o cliente (Figura x).

```
Título (uma linha descrevendo a história)

Narrativa:
Como [o papel]
Eu quero [recurso]
Assim que [benefício]

Critérios de Aceitação: (apresentado como Cenários)

Cenário 1: Título
Dado contexto []
  E [um pouco mais de contexto] ...
Quando [eventos]
Então [resultado]
  E [outro resultado ...]

Cenário 2: ...
```

Figura x. Padrão de escrita dos testes. Fonte: DevMedia.

6. Metodologia

6.1. Etapas

Durante o levantamento de requisitos para definir o escopo do projeto, foram utilizados as palavras chave “SUS”, “UBS”, “Ministério da Saúde”, “Problemas SUS”, “Dificuldades UBS”, entre outras palavras chave que retornasse a literatura para compreender o atual cenário na saúde pública brasileira, e como melhorá-la a partir da criação de novas tecnologias.

Após o estudo da literatura, iniciamos a definição das funcionalidades da aplicação pensando também em quais tecnologias que estão presente em nosso repertório utilizar na construção dos sistemas. Para então iniciar o design das telas da aplicação, e após sua definição, iniciar o desenvolvimento da API e interface.

6.2. Arquitetura da aplicação

Na Figura x está exemplificado o esquema da arquitetura da aplicação, onde primeiramente o usuário insere os dados, sejam eles cadastrais, agendamento de

consultas, manifestações e pesquisa de UBS. Posteriormente, a aplicação trata os dados mandando requisições para a API em NodeJS e posteriormente para o banco de dados, ambos hospedados na plataforma Heroku. Após as requisições, os dados são retornados novamente para o usuário, seguindo o caminho contrário.

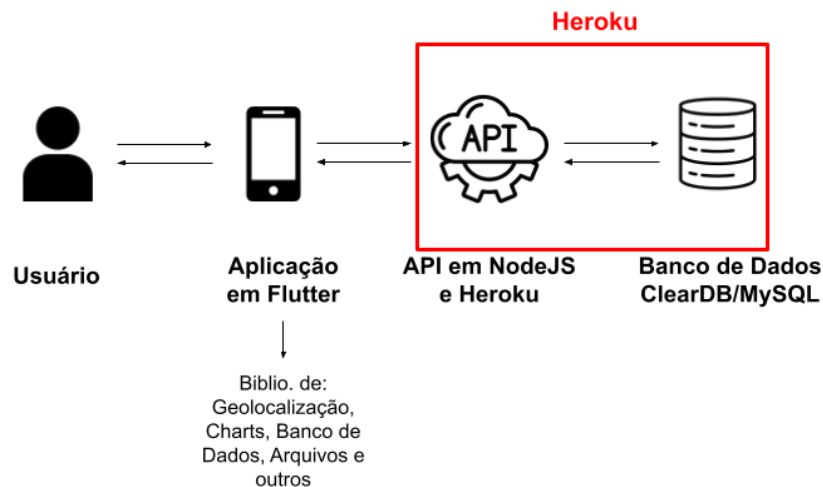


Figura x. Esquema de arquitetura da aplicação. Fonte: Autoria própria.

6.3. Critérios do produto

- Paciente poderá fazer o cadastro e acompanhamento de suas manifestações para a sua respectiva Ouvidoria Municipal ;
- Paciente terá a opção de ligar para a Ouvidoria Municipal;
- Paciente poderá pesquisar medicamentos, pesquisa será baseada na distância do paciente até a UBS mais próxima, e no estoque disponível ao medicamento;
- Paciente conseguirá ver diversos dados sobre a UBS, como a equipe administrativa, equipe médica, horário de funcionamento e contato do local;
- Paciente poderá avaliar UBS de acordo com sua satisfação, essa avaliação será usada para a criação de um ranking das melhores UBS da cidade;
- Gestores da UBS poderão alterar os dados referentes à UBS, como a equipe administrativa, equipe médica e estoque de remédios.
- Algoritmo de análise de dados a fim de exibir as estáticas de cada UBS;

6.4. Materiais e softwares utilizados

Para construir a API e a interface da aplicação, foram utilizadas bibliotecas, frameworks e softwares para conseguir processar e visualizar os dados de entrada na API e retorná-los corretamente na interface, de modo que o usuário consiga realizar as operações da aplicação sem empecilhos.

Abaixo, estão listadas as tecnologias utilizadas:

- Software de edição de código (IDE);
- Framework Flutter;
- Node.JS;

- Bibliotecas para realização de testes automatizados;
- Biblioteca para gerenciamento de arquivos;
- Biblioteca e API para visualização de mapa;
- Sistema gerenciador de banco de dados;
- Banco de dados;
- Database-as-a-service;
- *Hoster* de aplicação online;
- Software para criação e edição de protótipos de tela.

6.5. Desenvolvimento da API

Após a definição do escopo do projeto, criamos o MER do banco de dados para planejar como a aplicação e suas *features* iriam relacionar entre si. Após a finalização do MER, começamos a criar o banco de dados de forma local e realizar algumas mudanças na estrutura do banco de dados para adequá-las às funcionalidades da aplicação e evitar conflitos.

Finalizando a adequação das tabelas, primeiramente foi construído o banco de dados utilizando AWS e MySQL. Entretanto, não obteve-se sucesso, dado que as configurações para acessar o banco de dados remotamente exigia muitas permissões que o computador utilizado para o desenvolvimento não aceitava, atrasando o desenvolvimento da aplicação. Portanto, decidiu-se utilizar a plataforma Heroku, ClearDB e MySQL. Assim, foi possível construir com mais facilidade o banco de dados e integrá-lo a aplicação e realizar a inserção de alguns dados no banco de dados.

Para iniciar o desenvolvimento da API em NodeJS, criou-se as rotas para login, cadastro e obtenção dos pacientes, UBS, funcionários, prefeituras e outras entidades que compõem a aplicação. Além disso, também criou-se os outros componentes da API para integrá-la com a interface, como os controllers e arquivos para conectar API ao banco e executá-la. Cada *model* presente na API possui um método GET para obter um registro em específico e registros em gerais, além de um método POST para inserir registros.

Dentre as rotas, foram criadas funções para mostrar ao usuário alguns dados da UBS como manifestações, quantidade de remédios e ranqueamento de UBS. Assim, dados referentes às UBS poderiam ser vistos com maior facilidade pelos usuários.

6.6. Desenvolvimento da interface

Primeiramente fizemos o design das telas da aplicação no editor gráfico online Figma ¹ utilizando a premissa de implementar uma interface que fosse de fácil uso e entendimento, para que a aplicação pudesse abranger grande parte do

¹ Figma é um editor gráfico online que permite criar designs diversos
<<https://www.figma.com/>>

público utilitário de UBS. O design das telas também foi feito pensando nas funcionalidades estabelecidas no início da elaboração do projeto. Foram utilizados alguns sites do governo para basear as telas e funcionalidades relacionadas a denúncias. Alguns exemplos utilizados foram: OuvidoraSus - Portal², e-SIC Livre³, Portal Transparência de Campinas das UBS⁴ e Painel Resolveu⁵.

Ao finalizar o design inicial das telas, iniciou-se o desenvolvimento das telas utilizando a linguagem de programação Dart com o framework Flutter. O desenvolvimento das telas foi iniciado com as telas que já haviam sido feitas no Figma. Foram feitas algumas mudanças comparada aos esquemas de tela feitos previamente devido a inserção de funcionalidades na aplicação. Após o término destas telas, foram desenvolvidas novas telas como: pesquisa de medicamento utilizando a API Google Maps para buscar as unidades de saúde mais próximas do usuário em um raio de 2 a 3 KM, perfil do usuário, realização de manifestações, login, agendamento de consultas, gráficos das informações das UBS, consulta das manifestações realizadas e visualização das informações das UBS.

Seguindo as propostas iniciais estabelecidas no escopo do projeto, uma das *features* da aplicação era mostrar gráficos com os dados das UBS, como quantidade de manifestações, quantidade de remédios e ranking das UBS. Para tal, foi utilizada a biblioteca em Flutter *fl_charts*. Já para a realização das outras funcionalidades, buscamos o suporte de outras bibliotecas em Flutter para a visualização de mapas, gerenciamento de arquivos provenientes do usuário e conexão com o banco de dados, por exemplo. As bibliotecas utilizadas neste projeto são provenientes do site Pub.dev⁶.

Alguns desafios encontrados durante o desenvolvimento da interface encontraram-se principalmente na disponibilização de horários livres dos médicos e também no cálculo de divisão dos horários disponíveis para serem mostrados na interface. Para tal, foi utilizada uma matriz com as colunas sendo os horários e as linhas os médicos daquela especialidade, onde cada célula armazena o horário livre.

6.7. Validação técnica

Ao finalizar a API e para implementá-la à aplicação de forma que menos erros fossem retornados, foram desenvolvidos testes automáticos unitários para validá-la. As tecnologias utilizadas para realizar as validações foram as bibliotecas Mocha e Chai, ambas em JavaScript e comumente utilizadas para validar APIs em NodeJS.

A biblioteca Chai é uma biblioteca de testes baseada em Test-Driven Development e Behavior-Driven Development, que são, em suma, técnicas de

² OuvidoriaSus Portal é um site onde o usuário pode submeter suas manifestações em relação ao serviços do SUS <<http://ouvprod02.saude.gov.br/ouvidor/CadastroDemandaPortal.do>>

³ e-Sic <<https://esic.campinas.sp.gov.br/manual/LeiAcessoMundo.php>>

⁴ Portal Transparência Campinas <https://www.campinas.sp.leg.br/portal-transparencia>

⁵ Painel Resolveu <<http://paineis.cgu.gov.br/resolveu/index.htm>>

⁶ <https://pub.dev/> - Site de bibliotecas para Flutter.

desenvolvimento a fim de validar o comportamento do software. Para testar os algoritmos, são enviados JSONs hipotéticos. No caso dos testes implementados, foram enviados JSONs contendo as informações corretas e JSONs contendo informações erradas para capturar possíveis erros e comportamentos inesperados.

Foram desenvolvidos métodos para validar todas as requisições dos *models* de login, manifestação, ranqueamento de UBS, relatório e remédio UBS. Dentro destes métodos foram avaliados os parâmetros que compunham os JSONs retornados. Dentre esses parâmetros, foram avaliados os tipos, tamanho e correspondência com os dados hipotéticos.

7. Resultados

Ao finalizar a aplicação, obteve-se uma API capaz de realizar as operações como inserção, atualização e resgate de informações referentes aos pacientes, UBS, funcionários e outras dependências existentes no Sistema Único de Saúde. Grande parte dos *models* seguiram o padrão semelhante de construção, devido às necessidades das funções do frontend. A interface foi desenvolvida seguindo as funcionalidades previstas, como cadastro e login dos pacientes das UBS, consulta de medicamentos por UBS mais próxima, agendamento de consultas médicas e exames, realização de manifestações bem como o acompanhamento destas.

7.1. Resultados da validação técnica

Após realizar a validação, pode-se tirar algumas conclusões acerca dos *models* implementados. Em um cenário geral foi possível desenvolver testes para validar todas as rotas que se comportavam de maneira diferente das requisições básicas que só utilizam um “select”, além de testes. Dentre elas, podemos observar que a metodologia BDD foi eficiente em validar os dados retornados pelas requisições, entretanto, este método pode não validar todos os aspectos do comportamento da requisição, dado que esta metodologia não validou o tempo de processamento e a interação com outros componentes da API. Acreditamos que os testes unitários foram eficientes para validar cada rota da API, mas não sua interação como um todo.

Durante a implementação de testes para a validação do ranqueamento das UBS, observamos que a metodologia BDD não foi a mais adequada pois os testes deixaram alguns erros no ranqueamento passarem despercebidos. Acreditamos que com a implementação de testes de integração para simular a entrada de uma nova UBS no sistema seria possível validar adequadamente como o ranqueamento se comportaria.

7.2. Escassez de dados públicos

De modo a aprimorar a variedade de dados e conhecimento sobre os serviços de saúde públicos, decidiu-se implementar gráficos com dados

provenientes do IBGE e outros órgãos públicos. Ao iniciar a busca pelos dados, surgiram diversos empecilhos, tais como a descentralização das informações, descontinuidade de coletas de dados por parte dos órgãos públicos, ocasionando na inconsistência dos dados. Além de muitos dados relativos às próprias UBS não serem atualizados desde 2013.

Apesar de existirem diversos sites do próprio governo para divulgar orçamentos e dados referentes aos mais variados serviços públicos, nota-se que os dados do ministério da saúde são os menos presentes. O problema agrava-se ainda mais no mapeamento de serviços de saúde em cidades mais afastadas de metrópoles e grandes cidades.

8. Conclusão

Ao fim deste projeto, foi possível desenvolver uma aplicação *mobile* que permite auxiliar os usuários a ministrarem de forma eficiente suas opiniões sobre os serviços utilizados nas UBS. A utilização de tecnologias e ferramentas como Flutter, NodeJS, Heroku, ClearDB e MySQL permitiram a construção de uma aplicação de fácil uso para os cidadãos usuários do Sistema Único de Saúde. Além disso, com o auxílio de bibliotecas para geolocalização também foi possível desenvolver um sistema capaz de auxiliar o paciente a ministrar seus medicamentos e consultas provenientes do SUS, podendo evitar possíveis aglomerações nas UBS.

Durante a realização desta pesquisa na área da saúde, nota-se que a disponibilização e divulgação das informações por parte dos órgãos públicos demonstra-se ineficiente, dado que grande parte dos dados estão todos descentralizados, dificultando a sua busca. Além disso, há também a inconsistência dos dados, pois em muitos casos há a falta de coleta das informações em certos períodos, dificultando a criação de um *dataset* completo.

Foi feita a validação da API após a conclusão das suas rotas principais, onde foram utilizadas as ferramentas Mocha e Chai para realizar sua validação, ambas em JavaScript. A realização dessas validações demonstraram que o sistema é capaz de processar os dados dos usuários e do banco de dados de forma eficiente. Entretanto, ainda é necessário realizar outros tipos de testes automatizados para garantir melhor validação de outros componentes da aplicação, e futuramente implementar melhorias no sistema.

Por fim, espera-se que este projeto seja útil no sentido de auxiliar na construção de melhorias no sistema de saúde público e brasileiro, e também possa servir de alicerce ou contribuição para futuras pesquisas na área de implementação de tecnologias na saúde pública.

9. Referências bibliográficas

SOUZA, Ivan. Entenda o que é Rest API e a importância dele para o site da sua empresa. Disponível em: <<https://rockcontent.com/br/blog/rest-api/>>

PESCARINI, Fábio. Postos de saúde de SP têm falta de remédios e insumos para exames. *Editora: Folha de São Paulo*. Disponível em: <<https://www1.folha.uol.com.br/equilibrioesaude/2021/12/postos-de-saude-de-sp-te-m-falta-de-remedios-e-insumos-para-exames.shtml>>

Introdução ao Flutter: como funciona o framework e sua linguagem Dart. *Digital House*, 2020. Disponível em: <<https://www.digitalhouse.com/br/blog/o-que-e-flutter-e-como-funciona/>>

Falhas na atenção básica de saúde se repetem em várias regiões do país. *G1*, 2016 Disponível em: <<http://glo.bo/29AnmeH>>

DA SILVA, Valquiria Macedo. Principais desafios para organização do trabalho dos enfermeiros na estratégia de saúde da família na cidade de Araçuaí-MG. *Nescon*, 2009. Disponível em: <<https://www.nescon.medicina.ufmg.br/biblioteca/registro/referencia/0000001790>>

CANTALINO, J. L. R. et al. Satisfação dos usuários em relação aos serviços de Atenção Primária à Saúde no Brasil. *Scielo*, 2021. Disponível em: <<https://www.scielo.br/j/rsp/a/K8qkjBH69M6cVQ84kv36fhK/?format=pdf&lang=pt>>

CARTILHA DE OUVIDORIA MUNICIPAL. Ouvidoria Geral do Estado de São de Paulo. 2022. Disponível em: <<http://www.ouvidoriageral.sp.gov.br/PDF/Cartilha%20de%20Ouvidoria%20Municipal.pdf>>

Desafios para a nova gestão - Especialistas de saúde, educação e financiamento público apontam nessa rodada de perguntas quais os principais problemas que os gestores municipais devem encarar na nova legislatura, de 2021 a 2024. *EPSJV*, 2021 Disponível em: <<https://www.epsjv.fiocruz.br/noticias/reportagem/desafios-para-a-nova-gestao>>

GEREMIA, Daniela Savi. Atenção Primária à Saúde em alerta: desafios da continuidade do modelo assistencial. *Physis: Revista de Saúde Coletiva*, v. 30, p. e300100, 2020.

Ministério da Saúde. Desafios da Atenção Primária à Saúde. *Portal da Secretaria de Atenção Primária a Saúde*, 2019. Disponível em: <<https://aps.saude.gov.br/noticia/6555>>

INSTITUTO ARAPYAUÍ; PLANO CDE. Desafios para Inovação na Gestão Municipal. Plano CDE. *Arapyaú*, 2019.

Disponível em:
<https://arapyau.org.br/wp-content/uploads/2019/05/PUBLICA%C3%87%C3%83O_Desafios-para-inova%C3%A7%C3%A3o-na-gest%C3%A3o-municipal-1.pdf>.

CERQUEIRA, Wagner. Problemas sociais nas cidades do Brasil. *Mundo Educação*, 2019.

Disponível em:
<<https://mundoeducacao.uol.com.br/geografia/problemas-sociais-nas-cidades-brasil.htm>>.

RESOLVEU?. Painei “Resolveu?” - Controladoria-geral da União. Disponível em:
<<http://paineis.cgu.gov.br/resolveu/index.htm>>.

Ouvidoria Geral da União (OGU). Orientações para implementação de uma unidade de ouvidoria. 5. ed. Brasília - DF. CGU, 2012. Disponível em:
<<https://www.gov.br/cgu/pt-br/centrais-de-conteudo/publicacoes/ouvidoria/arquivos/ogu-implantacao-unidade-ouvidoria.pdf>>

GONÇALVES, Camila. Ouvidorias Públicas Municipais no Brasil. XIV Congresso Brasileiro de Sociologia. 2009.

Disponível em:<https://www.researchgate.net/publication/220034714_Ouvidorias_Publicas_Municipais_no_Brasil>

GOMES, Alice Vaz. Determinantes da satisfação no atendimento das Unidades Básicas de Saúde (UBS). *Scielo*, 2013.

Disponível em: <<https://www.scielo.br/j/csc/a/8KkBFNhtbDJmPH3Z5GnRndC/#>>

CAVALCANTI, P. C. S. et al. Quais são os desafios para a qualificação da Atenção Básica na visão dos gestores municipais?.

Disponível em: <<https://www.scielo.org/article/sdeb/2015.v39n105/323-336/#>>

LYRA, Rubens. “Ouvidor: o defensor dos direitos na administração pública brasileira”. In: 16/03.Rubens Pinto Lyra (org.) Autônomas x Obedientes. A Ouvidoria Pública em debate, João Pessoa: UFPB, 2004.

MAGALHÃES, Tulio. DESCUBRA FINALMENTE O QUE É O CLOUD COMPUTING E PARA QUE SERVE A COMPUTAÇÃO EM NUVEM. *Rockcontent*, 2018.

Disponível em <<https://rockcontent.com/br/blog/cloud-computing/>>

Unidades Básicas de Saúde (UBS). Secretaria de Saúde do Distrito Federal. 2022.

Disponível em: <<https://www.saude.df.gov.br/unidades-basicas>>.

Unidades Básicas de Saúde (UBS). Sociedade Paulista Para o Desenvolvimento da Medicina. 2022. Disponível em:

<<https://www.spdm.org.br/onde-estamos/outras-unidades/unidade-basica-de-saude-ubs>>

GEORGE, Bobby; WILLIAMS, Laurie. A structured experiment of test-driven development. Information and software Technology, v. 46, n. 5, p. 337-342, 2004. Acesso em: 8 de Novembro de 2022.

MOE, Myint Myint. Comparative Study of Test-Driven Development TDD, Behavior-Driven Development BDD and Acceptance Test-Driven Development ATDD. International Journal of Trend in Scientific Research and Development, v. 3, p. 231-234, 2019.

Acesso em: 8 de Novembro de 2022.

Digité, O que é teste de unidade? 2022. Disponível em:

<<https://www.digite.com/pt-br/agile/testes-unitarios/>>

Acesso em 9 de Novembro de 2022.

ZHU, Hong; HALL, Patrick AV; MAY, John HR. Software unit test coverage and adequacy. Acm computing surveys (csur), v. 29, n. 4, p. 366-427, 1997.