

Desenvolvimento de um Protótipo de Aplicação para Problemas de Infraestrutura Urbana em Santa Maria-RS

Pedro Muller Poll, Ana Paula Canal
Curso de Sistemas de Informação
UFN - Universidade Franciscana
Santa Maria - RS
pedro.mpoll@ufn.edu.br, apc@ufn.edu.br

Resumo—A infraestrutura urbana é um assunto importante e faz toda a diferença no cotidiano das pessoas, que por não terem ao seu dispor ferramentas atrativas acabam por não participar na solução de problemas relacionados à infraestrutura em suas cidades. Com foco na cidade de Santa Maria, o trabalho tem como objetivo desenvolver uma aplicação móvel com Flutter e DART aliado a um sistema web com Laravel e PHP para permitir ao cidadão reportar problemas de infraestrutura de maneira facilitada, bem como se informar sobre problemas reportados por outros usuários, os reportes então serão encaminhados à prefeitura para facilitar a solução do problema, criando um ambiente urbano mais agradável de se interagir.

Palavras-chave : Cidades Inteligentes; Aplicação Móvel; Sistema Web;

I. INTRODUÇÃO

Quando se fala de problemas com a infraestrutura urbana de uma cidade, percebe-se que não se tem muitas opções além de ligar para a prefeitura e reportar o problema, mas com o avanço da tecnologia muitos não percebem as ferramentas online ao dispor de todos. Por ser algo que não está no controle da população, no cotidiano acaba-se não dando a devida atenção para estes problemas de infraestrutura que são observados, como buracos em calçadas, fios soltos, bueiros entupidos, falhas de iluminação pública e outros pontos críticos comuns no ambiente urbano. Em parte, isso ocorre pela falta de uma maneira facilitada de se reportá-los. Todo cidadão tem acesso às ferramentas da prefeitura que permitem reportar problemas de iluminação e solicitar coleta de resíduos, mas, se aliadas a uma solução interativa, a eficiência destas ferramentas poderia ser aprimorada.

Uma solução interativa ligada às ferramentas já existentes seria algo mais atraente para um usuário, criando incentivo para que mais funcionalidades do site sejam descobertas e utilizadas com mais frequência.

A partir de ferramentas como Flutter, Dart, Laravel e PHP, busca-se criar uma plataforma interativa que conecte os cidadãos ao processo de identificação e solução dessas questões para incentivar a participação e aumentar a eficiência da resolução de problemas de infraestrutura.

A. Justificativa

A mobilidade urbana tem grande impacto na qualidade de vida do cidadão, e quando a infraestrutura urbana apresenta problemas, esta mobilidade pode ser comprometida [1]. Com o aumento do tamanho das cidades, surgem diversos problemas no meio urbano, especialmente, os de infraestrutura e que afetam diretamente a vida das pessoas. Por exemplo, problemas de iluminação pública, saneamento básico, coleta de lixo, pavimentação das ruas, condições das calçadas, entre outros [2].

Em Santa Maria, por conta de um crescimento urbano desordenado, muitas localidades não têm a infraestrutura necessária para resistir a grandes crises climáticas, por isso as chuvas que assolaram o Rio Grande do Sul foram extremamente devastadoras para a cidade, abalando sua infraestrutura de maneira radical [3]. esta pauta nunca foi tão importante e se vê cada vez mais necessário um foco maior na infraestrutura local e em formas de se facilitar a solução de problemas relacionados a infraestrutura na cidade.

O aplicativo utilizará o GPS (*Global Positioning System*) para alertar o usuário sobre registros próximos feitos por outros cidadãos. Além disso, cada ocorrência informada será visualizada pelos demais usuários do aplicativo, de forma colaborativa. Essa abordagem poderá auxiliar na segurança, mantendo os usuários informados e ajudando a prefeitura a coletar relatos sobre a infraestrutura da cidade.

B. Objetivos

Para possibilitar ao usuário, por meio de um mapa interativo, reportar, visualizar e ser notificado sobre problemas de infraestrutura como buracos em calçadas e problemas de iluminação pública na cidade de Santa Maria. o objetivo geral do trabalho é:

- Desenvolver um protótipo de aplicação para reportar problemas de infraestrutura urbana na cidade de Santa Maria - RS.
- Elaborar um referencial teórico sobre problemas de infraestrutura urbana na cidade de Santa Maria-RS.
- Desenvolver um protótipo de aplicativo Android para registro e visualização de problemas urbanos, usando o framework Flutter com Dart.

- Desenvolver protótipo de sistema web para integrar os registros recebidos, priorizá-los e possibilitar a visualização e encaminhamento do registro para sua solução, em PHP e o framework Laravel.

II. REFERENCIAL TEÓRICO

Nesta seção serão abordados os temas relevantes a proposta do trabalho e uma visão mais detalhada das tecnologias e a metodologia que poderão ser empregadas no futuro desenvolvimento do aplicativo.

A. Cidades Inteligentes

Uma cidade inteligente é um sistema urbano que utiliza tecnologia de informação e comunicação para trazer mais interatividade tanto aos aspectos de infraestrutura como aos serviços públicos em geral. Termos semelhantes são comumente empregados: Cidades Virtuais, Cidades do Conhecimento, Cidades Digitais, ou ainda Cidades de Informação [4].

A infraestrutura pode ser definida como um conjunto de estruturas de engenharias e instalações que são a base necessária para o desenvolvimento de atividades produtivas de prestação de serviços, política, social e pessoal nas cidades [5].

Em um contexto histórico nacional, entre as décadas de 1950 e 1970, o governo que geralmente planejava a execução de projetos envolvendo a infraestrutura, apenas em 1990 as privatizações começaram a ter participação mais ativa neste tema, colocando o setor privado como contribuintes maiores [1].

A Confederação Nacional da Indústria (CNI) mostrou que investimentos em infraestrutura podem aumentar o Produto Interno Bruto (PIB) de uma cidade em até 2,9% [6]. O investimento em infraestrutura está intrinsecamente ligado ao aumento do PIB, a diminuição da pobreza e a melhoria da qualidade de vida das pessoas.

B. Serviços de Infraestrutura Urbana disponíveis (TICs) na cidade de Santa Maria

A prefeitura de Santa Maria conta com alguns serviços dedicados a infraestrutura urbana, pelo site [7]¹ o cidadão que estiver logado na sua conta do site da prefeitura pode usufruir dos seguintes serviços, cujos formulários vão servir de base para o sistema de reporte do atual trabalho:

1) *Solicitação de resíduos*: Por meio de um formulário o cidadão pode requisitar a coleta de lixo e descartáveis no geral, assim como fazer uma reclamação ou sugestão sobre focos de resíduos².

2) *Solicitação de Reparos de Iluminação Pública*: Por meio de um formulário o cidadão pode reportar a necessidade de reparos em postes de luz, este formulário requisita ao usuário suas informações pessoais, as informações do local da ocorrência e também aceita fotos como referência³.

C. Trabalhos relacionados

Para os trabalhos correlatos foram escolhidos projetos com foco em arranjar soluções para problemas de infraestrutura por meio da tecnologia.

1) *Urban Voice – Uma abordagem baseada em perfil e recomendação para relatar problemas de infraestrutura urbana*: O Urban Voice é um projeto de um aplicativo mobile que permite o usuário reportar problemas de infraestrutura, interagir com usuário que tiverem problemas parecidos e acompanhar o andamento da resolução do problema reportado. O projeto conta com um algoritmo de recomendação e nota dos reportes e usuários [2].

2) *O uso de tecnologia mobile como ferramenta auxiliar em ações de acessibilidade espacial em ambiente de ensino universitário*: Este projeto visou usar tecnologias como QR Code e Geolocalização para auxiliar na acessibilidade espacial com foco no campus da Universidade Federal de Juiz de Fora (UFJF). Um protótipo foi pensado para usar as tecnologias citadas para melhorar a locomoção e orientação dos usuários no ambiente do campus, com foco nos usuários que sofrem com mobilidade reduzida [8].

3) *SMALL: soluções computacionais em mobilidade urbana para auxiliar na consolidação de cidades inteligentes*: No projeto de Alef Menezes, é abordada a mobilidade urbana em cidades inteligentes, onde ele estuda soluções computacionais existentes em prol de facilitar o uso de soluções inteligentes para problemas comuns de mobilidade urbana, como congestionamento e planejamento de rotas, usando um guia como uma referência prática para a tomada de decisões [9].

Nota-se que entre os trabalhos explorados o uso da tecnologia é foco central para a melhora da qualidade de vida e a acessibilidade, no projeto SMALL a ideia geral do atual trabalho sobre problemas de infraestrutura se mostra presente, já em 'O uso de tecnologia mobile como ferramenta auxiliar em ações de acessibilidade espacial em ambiente de ensino' cria uma visão futura do projeto onde opções de acessibilidade podem ser uma opção e o Urban Voice foi uma inspiração para o sistema de notas.

D. Metodologia do Processo de Desenvolvimento

O FDD (*Feature Driven Development*) foi a metodologia escolhida para a proposta do aplicativo, ele é uma metodologia ágil que busca trazer um desenvolvimento focado em funcionalidades pré-definidas.

Por ser uma metodologia ágil, o FDD conta com a colaboração da equipe para lidar com a complexidade do

¹<https://www.santamaria.rs.gov.br/>

²<https://www.santamaria.rs.gov.br/usuario/servicos/863>

³<https://www.santamaria.rs.gov.br/usuario/servicos/888>

projeto, decompondo o projeto em funcionalidades, isso é feito por meio de gráficos e comunicação verbal, que, no desenvolvimento, resulta em incrementos de software.

Neste contexto, uma funcionalidade pode ser vista como uma função valorizada pelo cliente passível de ser implementada em duas semanas ou menos [10].

A etapa de Construir uma Lista de Funcionalidades é onde são definidas as funcionalidades e suas especificações, planejar por funcionalidades é onde é feito um diagrama de classe para se ter uma ideia melhor de como o sistema irá funcionar, projetar por funcionalidade e desenvolver por funcionalidade.

E. Tecnologias para o desenvolvimento

Foi escolhida a linguagem PHP com o *Framework* Laravel para proposta de desenvolvimento do sistema web e a linguagem Dart aliada ao *framework* Flutter para a aplicação móvel, o armazenamento de informações será feito com uma base de dados no *Firebase*.

1) *PHP*: Lançado em 1995 por Rasmus Lerdorf, o PHP é uma Linguagem de script open-source popularmente usada para o desenvolvimento de aplicações web devido a sua versatilidade e facilidade de uso [11], além disso a linguagem é flexível e tem boa compatibilidade com HTML, CSS e Javascript.

2) *Laravel*: Laravel é um *framework* PHP para o desenvolvimento de aplicações web, ela é considerada simples de usar pois tem grande nível de abstração e sua robustez faz ele ser bom para projetos de todos os tamanhos, por isso atualmente o Laravel é uma das formas mais populares de se desenvolver aplicações web *full-stack* com PHP. para melhorar a integração entre o backend e o frontend, o Laravel conta com a biblioteca *Livewire*, que permite a criação de páginas mais dinâmicas por meio de código Javascript⁴ [12].

3) *Dart*: Dart é uma linguagem multiplataforma normalmente usada para aplicações web, móveis e *desktop*, possui ferramentas para formatação, análise e teste de código e também é a base do *framework* Flutter⁵ [13].

4) *Flutter*: Flutter é um *framework* em Dart de código aberto usado para criar aplicações web, desktop e mobile⁶ [14].

5) *Firebase*: *Firebase* é uma plataforma da Google usada para o host de aplicações na nuvem, ela também conta com banco de dados em tempo real e um sistema de autenticação de usuário [15].

O *Firebase*, assim como a plataforma Google Maps, cobra pelo uso, se ele não exceder uma certa quantia da pra se utilizar de ambos os serviços de graça, que é como foi planejado para o projeto.

⁴<https://laravel.com/docs/11.x>

⁵<https://dart.dev/overview>

⁶<https://flutter.dev/>

III. METODOLOGIA

Primeiramente, foi realizada a revisão bibliográfica, em busca da temática Cidades Inteligentes, Infraestrutura Urbana, Mobilidade e os trabalhos correlatos. Também, foi realizada a busca dos serviços disponíveis pela Prefeitura de Santa Maria, no seu site.

Para elaboração da proposta, a metodologia FDD é usada, focada na entrega de funcionalidades, será usada no futuro para facilitar o desenvolvimento da aplicação. nas seções seguintes, descreve-se o processo de desenvolvimento com a metodologia FDD.

No Diagrama de Domínio na Figura 1 o endereço será associado a ocorrência, nela terá a informação do tipo do reporte, descrição e foto.

A conta normal de usuário será usada para logar no sistema de administração, em uma versão final o usuário administrador teria uma forma especial de logar no sistema web, para este protótipo as contas de usuário e de administrador serão as mesmas.

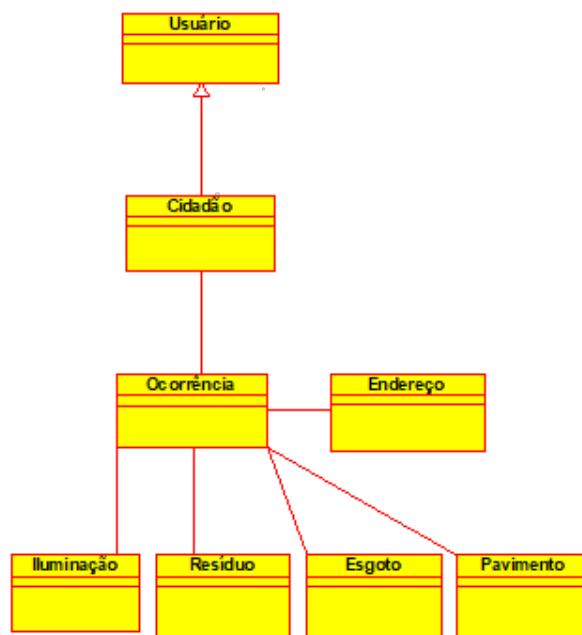


Figura 1. Diagrama de Domínio

A. Construir uma Lista de Funcionalidades

Para mostrar com mais detalhes o funcionamento do sistema e como o usuário interage com ele, foi criado um Diagrama de Caso de Uso (Figura 2) e um diagrama geral (Figura 3). O usuário deverá primeiro criar uma conta e fazer o login, permitindo acesso a tela principal onde ele poderá usar o mapa interativo e realizar novos reportes, além de poder editar ou apagar os mesmos. Ele também poderá receber notificações de reportes feitos por ele que foram finalizados pelos administradores logados no sistema web.

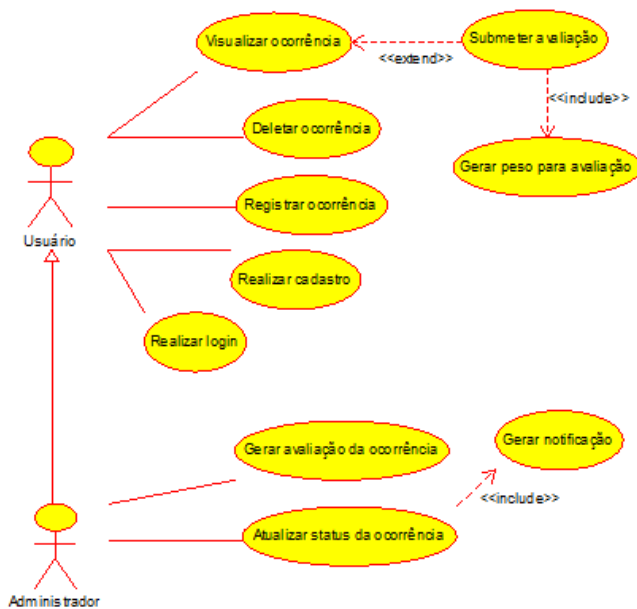


Figura 2. Diagrama de Caso de Uso

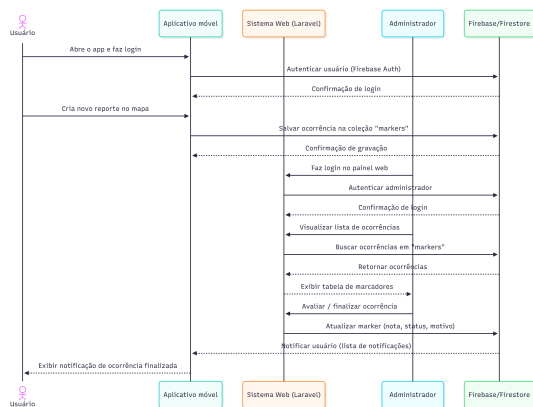


Figura 3. Diagrama Geral

A lista de funcionalidades (requisitos funcionais) da aplicação é:

1) *Realizar cadastro*: Para interagir com o sistema o usuário deverá realizar um cadastro consistindo em um formulário onde o seu nome completo, telefone, informações de endereço e email deverão ser informados, bem como um nome de usuário e senha, sendo redirecionado a tela principal com o mapa.

2) *Realizar login*: O usuário deverá informar seu nome de usuário e senha, sendo então redirecionado para a tela principal.

3) *Registrar ocorrência*: Nesta tela, o usuário irá incluir informações sobre o reporte, como a sua natureza, descrição, ponto de referência e fotos do local e submeter as informações de algum problema.

4) *Visualizar ocorrência*: Terá a função de visualizar as informações de qualquer ocorrência que o usuário interagir no mapa.

5) *Submeter avaliação*: Dentro da parte de visualização, o usuário poderá dar uma nota de 1 a 5 estrelas para ocorrências feitas por outros usuários clicando em um reporte de outra pessoa.

6) *Deletar ocorrência*: Ocorrências criadas pelo usuário podem ser deletadas por ele a qualquer momento por meio de um botão ao selecionar um reporte feito por ele na tela principal.

7) *Gerar avaliação da ocorrência*: O administrador responsável irá validar a ocorrência e ver se ela é relevante, se sim ela será encaminhada para o agente responsável na prefeitura e o usuário será notificado.

8) *Gerar peso para avaliação*: Quando o usuário submete a avaliação em algum reporte de outra pessoa, o sistema calcula por meio de pesos a chance da ocorrência ser legítima.

9) *Atualizar status da ocorrência*: Um usuário apontado como administrador terá a função de atualizar o usuário do status da ocorrência, se o problema está sendo tratado e eventuais informações adicionais que venham a ser necessárias para o andamento da solução, cada atualização gera uma notificação ao usuário alvo.

10) *Gerar notificação no aplicativo*: O sistema gera notificações quando o administrador finaliza algum reporte criado por ele, as notificações podem ser acessadas por meio de um botão onde será listado todos os reportes fechados e o motivo pelo seu fechamento.

Os requisitos não funcionais da aplicação são:

- A aplicação mobile será desenvolvida em Flutter.
- O sistema web será desenvolvido em PHP com Laravel.
- O *Firebase* será responsável pelo banco de dados em tempo real e a autenticação de usuário.
- *frontend* e *backend* do sistema web serão feitos com Laravel e PHP.

B. Planejar por Funcionalidade

Nesta etapa será estimado o tempo necessário para o desenvolvimento de funções específicas, ilustrado na tabela I.

C. Projetar e Construir por Funcionalidade

No projeto, foi elaborada uma estrutura de classes que pode ser vista por meio de um diagrama de classes na Figura 4, o diagrama segue como o banco de dados do Firestore foi estruturado para o sistema. Nesta sessão também serão apresentadas algumas partes do código explicando como certas funcionalidades foram implementadas.

No FDD, cada funcionalidade passa por um ciclo de *design* e desenvolvimento antes de ser entregue. Essa abordagem permite uma melhor adaptação ao longo do tempo, as iterações ocorrem dentro do desenvolvimento de cada

Tabela I
ESTIMATIVA DE TEMPO POR FUNCIONALIDADE

Funcionalidade	Horas Estimadas
Realizar cadastro	30 h
Realizar login	30 h
Registrar ocorrência	20 h
Visualizar ocorrência	20 h
Deletar ocorrência	20 h
Submeter avaliação	20 h
Gerar peso para avaliação	40 h
Gerar avaliação da ocorrência	20 h
Atualizar status da ocorrência	20 h
Gerar notificação	30 h

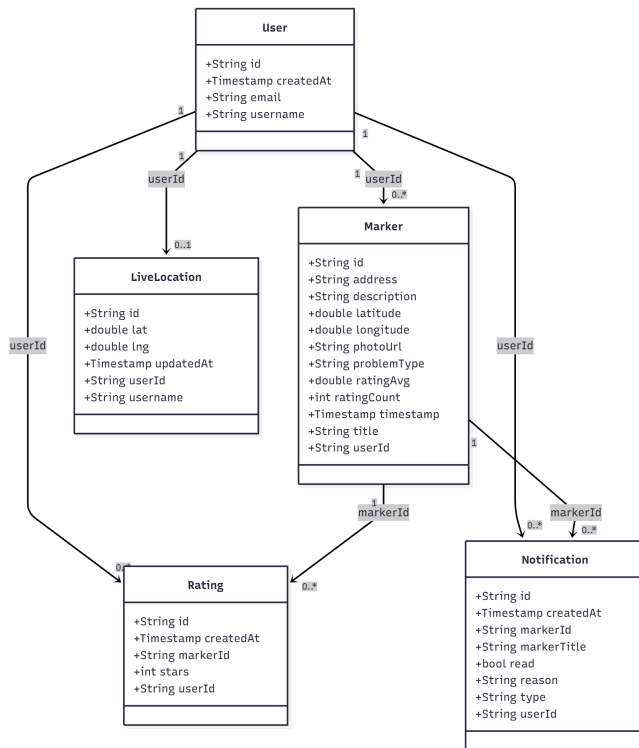


Figura 4. Diagrama de classes

funcionalidade, estas iterações irão ocorrer quando o desenvolvimento da aplicação proposta começar a ser desenvolvida no trabalho futuro, com base nos diagramas e listas de funcionalidades apresentados.

D. Aplicativo Mobile

Nesta sessão serão explicados alguns trechos de código do aplicativo *mobile* e *web*, tentando seguir o fluxo natural de execução das aplicações.

1) *Tela de Registro*: No trecho 5 a parte de validação do registro pode ser vista com `FirebaseService.register()`, além da mensagem de sucesso ou erro. O fluxo natural, depois de uma conta criada com sucesso, é o redirecionamento para a tela de login.

```

36 ~ Future<void> _doRegister() async {
37   if (!formKey.currentState.validate()) return;
38   setState(() => _loading = true);
39   try {
40     await FirebaseService.register(
41       _usernameCtrl.text.trim(),
42       _emailCtrl.text.trim(),
43       _passwordCtrl.text.trim(),
44     );
45     if (!mounted) return;
46     ScaffoldMessenger.of(context).showSnackBar(const SnackBar(content: Text('Conta criada! Faça login.')));
47     Navigator.of(context).pop();
48   } catch (e) {
49     if (mounted) {
50       ScaffoldMessenger.of(context).showSnackBar(SnackBar(content: Text('Erro ao registrar: $e')));
51     }
52   } finally {
53     if (mounted) setState(() => _loading = false);
54   }
55 }
  
```

Figura 5. Função de registro

2) *Tela de login*: Numa tentativa de login o aplicativo conecta ao *Firebase* para autenticar as credenciais inseridas, na imagem 6 o `FirebaseService.login()` valida o email e a senha apenas se ambos os campos estiverem preenchidos. Na sequência, se as credenciais estiverem corretas, o sistema redireciona o usuário para a tela principal do sistema.

```

110 Future<void> _login() async {
111   final email = emailController.text.trim();
112   final password = passwordController.text.trim();
113
114   if (email.isEmpty || password.isEmpty) {
115     ScaffoldMessenger.of(context).showSnackBar(
116       const SnackBar(content: Text("Preencha todos os campos")),
117     );
118     return;
119   }
120
121   setState(() => loading = true);
122
123   try {
124     final userData = await FirebaseService.login(email, password);
125
126     if (userData == null) {
127       throw Exception("Usuário ou senha incorretos");
128     }
  
```

Figura 6. Autenticação do *Firebase*

3) *Tela principal*: A parte de funções relevantes de posição do usuário se encontram em um único arquivo chamado `location_page`, nele tem algumas partes para destacar:

- A atualização do marcador de usuário no mapa por meio de `updateMyMarker`, e o envio da posição em tempo real para o backend do *Firebase* com `updateLiveLocation` como visto na figura 7.
- Na figura 8 o sistema permite o usuário dar uma nota para um reporte, ao selecionar uma nota, chama `rateMarker` no *Firebase* para atualizar o valor, depois, o `SnackBar` mostra na tela uma confirmação de sucesso.
- O código ilustrado da imagem 9 mostra a parte responsável pela atualização do mapa em tempo real a medida que reportes são criados, editados ou deletados no *Firebase*, com cada atualização, os documentos recebidos da nuvem serão transformados no tipo marcadores (`Marker`) do Google Maps.

4) *Aplicativo web*: Para ajudar a ilustrar a parte dos pesos de avaliação, a figura 9 mostra como o cálculo no sistema *web* é feito. Esse trecho mostra o algoritmo de atualização da média de avaliações de um marcador. O sistema não

```

188     _locSub = Geolocator.getPositionStream(
189         locationSettings: const LocationSettings(
190             accuracy: LocationAccuracy.best,
191             distanceFilter: 5,
192         ),
193     ).listen((pos) {
194         _updateMyMarker(pos);
195         FirebaseService.updateLiveLocation(
196             userId: widget.userId,
197             username: widget.username,
198             latitude: pos.latitude,
199             longitude: pos.longitude,
200         );
201     });

```

Figura 7. Stream de localização

```

InteractiveStars({
  initial: selected,
  onSelect: (v) async {
    await FirebaseService.rateMarker(
      markerId: markerId,
      stars: v,
    );
    setState(() => selected = v);
    if (mounted) {
      ScaffoldMessenger.of(context).showSnackBar(
        const SnackBar(
          content: Text('Avaliação registrada!'),
        ), // SnackBar
      );
    }
  },
},

```

Figura 8. Sequência de avaliação de reporte

```

void _listenMarkers() {
  _markersSub = FirebaseService.markersStream().listen((list) {
    setState(() {
      // preserva apenas o marcador "me"
      final me = _markers.where((mk) => mk.markerId == _meMarkerId).toList();
      _markers.clear();
      if (me.isNotEmpty) _markers.add(me.first);

      for (final m in list) {
        final lat = (m['latitude'] as num).toDouble();
        final lng = (m['longitude'] as num).toDouble();
        final id = (m['id'] ?? '').toString();
        final title = (m['title'] ?? '').toString();
        final desc = (m['description'] ?? '').toString();

        _markers.add(
          Marker(
            markerId: MarkerId('m_$id'),
            position: LatLng(lat, lng),
            infoWindow: InfoWindow(
              title: title,
              snippet: desc,
              onTap: () => _showMarkerDetailsDialog(m),
            ), // InfoWindow
            onTap: () => _showMarkerDetailsDialog(m),
          ), // Marker
        );
      }
    });
  });
}

```

Figura 9. Listener de marcadores do Firestore

guarda os votos individualmente, ele sempre recalcula a média usando a média anterior, o número de votos no total (oldCount), e a nota do painel de administração (adminRating) que foi recém atribuída.

```

36     $oldAvg = (float) ($marker['ratingAvg'] ?? 0);
37     $oldCount = (int) ($marker['ratingCount'] ?? 0);
38     $oldAdminScore = $marker['adminRating'] ?? null;
39
40     if ($oldAdminScore == null) {
41         $newCount = $oldCount + 1;
42         $newAvg = $oldCount > 0
43             ? (($oldAvg * $oldCount) + $rating) / $newCount
44             : $rating;
45     } else {
46         $newCount = $oldCount;
47         if ($newCount > 0) {
48             $newAvg = (($oldAvg * $oldCount) - $oldAdminScore + $rating) / $newCount;
49         } else {
50             $newCount = 1;
51             $newAvg = $rating;
52         }
53     }
54
55     $newAvg = round($newAvg, 2);

```

Figura 10. Cálculo do peso de avaliação no sistema web

E. Sistema de peso das avaliações

Para calcular os pesos, são considerados dois tipos de avaliação: de usuários usando o *app* de celular e de administradores. As notas dos usuários do aplicativo e as do administrador terão basicamente o mesmo peso, tudo contando para a média final da nota. Pensando como se fosse uma formula, dá para dizer que o cálculo seria a soma das notas dos usuários mais a nota do avaliador, em seguida divide o total pela quantidade de votos normais, mais os votos do avaliador.

IV. RESULTADOS

Inicialmente, serão ilustradas as interfaces finais de ambas aplicações móvel e *web*, com uma explicação simples e a menção de alguns pontos ainda não comentados, e ao fim serão pensadas visões para um possível futuro do projeto e o que ele pode alcançar.

A. Aplicativo móvel

É a parte principal do projeto, o fluxo de interação começa na tela de login da figura 11, onde o usuário pode navegar para a tela de registro de nova conta ou colocar suas credenciais e efetuar o *login*.

Na tela de registro vista na figura 12 temos os campos de nome de usuário, email, e campos para a senha.

Em um registro bem sucedido, o usuário é redirecionado para a tela de *login* novamente.

Chegando na tela principal da figura 13 nós temos as funcionalidades mais importantes do projeto, o marcador azul mostra a posição do usuário, os pontos vermelhos são marcadores de reportes.

no canto superior direito da figura 13 existem botões para ver notificações, atualizar posição (usado se o GPS estava inicialmente desligado para recuperar a posição em tempo real) e *logout* (que retorna para a tela de *login*).

Ao tocar em algum ponto do mapa que não seja a posição atual ou outro reporte, uma modal da figura 14 para criar um novo marcador será aberta, nela o usuário poderá informar a natureza do problema e tirar uma foto para ilustrá-lo, as informações de endereço são automaticamente coletadas pelo sistema.

Quando o usuário toca em um marcador no mapa, uma modal de visualização vista na imagem 15 é apresentada, mostrando todas as informações daquele reporte, se o marcador tiver sido criado por ele, também terá a opção de editá-lo como visto na figura 14 por meio de uma modal de edição, caso não for dele, o usuário poderá dar uma nota ao marcador.

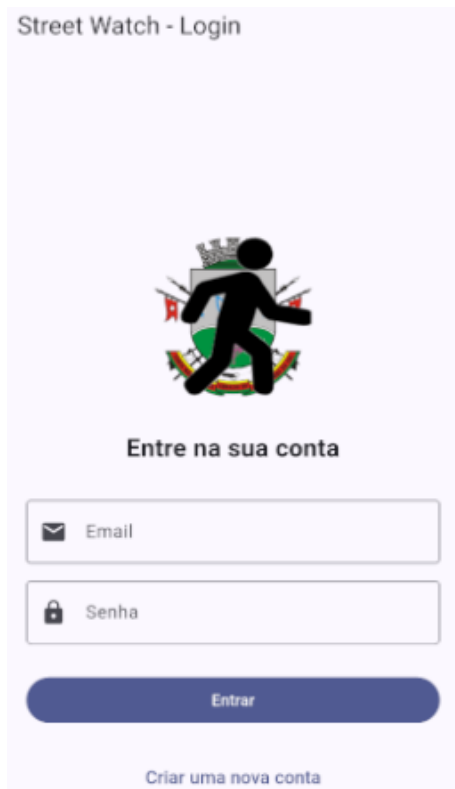


Figura 11. Tela de login do aplicativo móvel

B. Sistema Web do administrador

O sistema *web* desenvolvido é menos complexo, consistindo apenas de uma tela de login vista na figura 16 que usa as mesmas credenciais e autenticações do *Firebase* que o *app mobile* usa.

O administrador logado na tela principal então terá a capacidade de ver todos os marcadores existentes no sistema por meio de uma tabela ilustrada na figura 17, nela é possível dar uma nota para os reportes e, por meio de um botão que abre uma modal na figura 18, excluir um marcador e escrever a razão em uma caixa de texto, essa razão é depois



Figura 12. Tela de registro de usuário

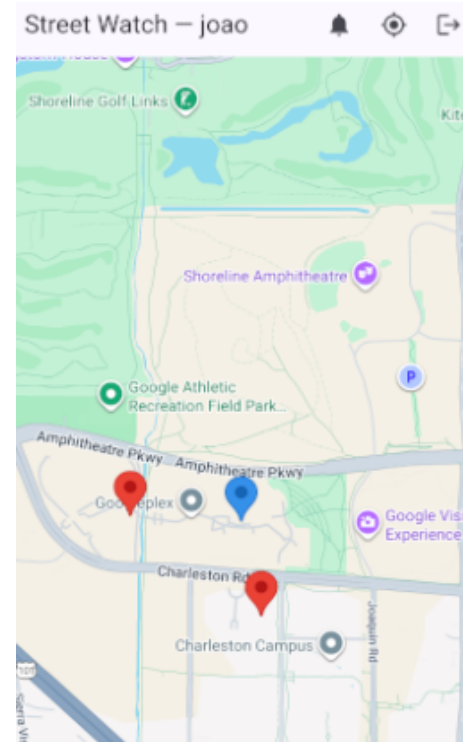


Figura 13. Tela do mapa

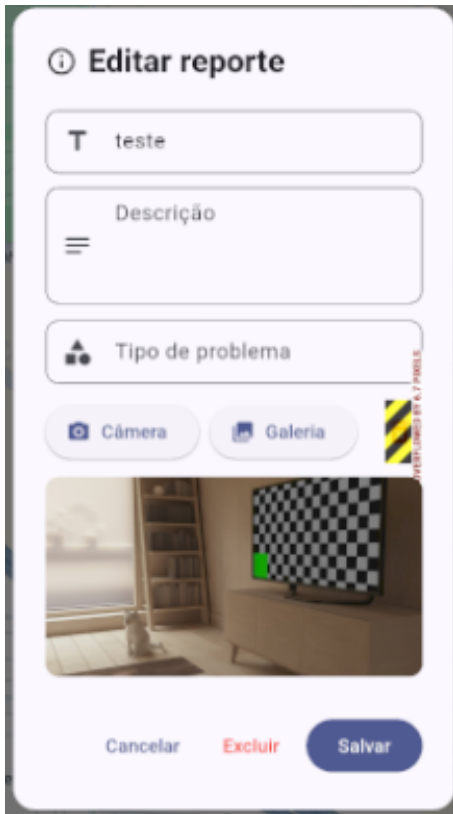


Figura 14. Tela para registro e edição de reporte

mostrada ao usuário na caixa de notificações do aplicativo móvel como visto na figura 19.

V. CONCLUSÕES DO TRABALHO

Conclui-se que, com os recursos necessários, a implementação do protótipo em escala maior seria possível.

A metodologia FDD guiou a definição das funcionalidades e da organização do projeto, possibilitando estruturar o escopo em etapas claras e orientadas a entregas. Em seguida, foram planejados os componentes principais, elaborados diagramas e implementados os protótipos iniciais do aplicativo mobile e do sistema web.

Com base nos resultados obtidos, observou-se que o uso do *Firebase*, aliado ao Flutter no aplicativo e ao Laravel no sistema web, demonstrou potencial para fornecer uma solução integrada, com atualizações em tempo real e comunicação entre usuários e administradores. As telas desenvolvidas, funcionalidades de reporte, visualização, avaliação e notificações ilustram a viabilidade do sistema em criar uma forma interativa de se participar em questões de mobilidade urbana.

Alguns testes foram cogitados e planejados, e a implementação prática poderá ser aplicada em projetos futuro.

Entre as contribuições do trabalho, da para destacar a criação de uma ferramenta que incentiva a participação cidadã,

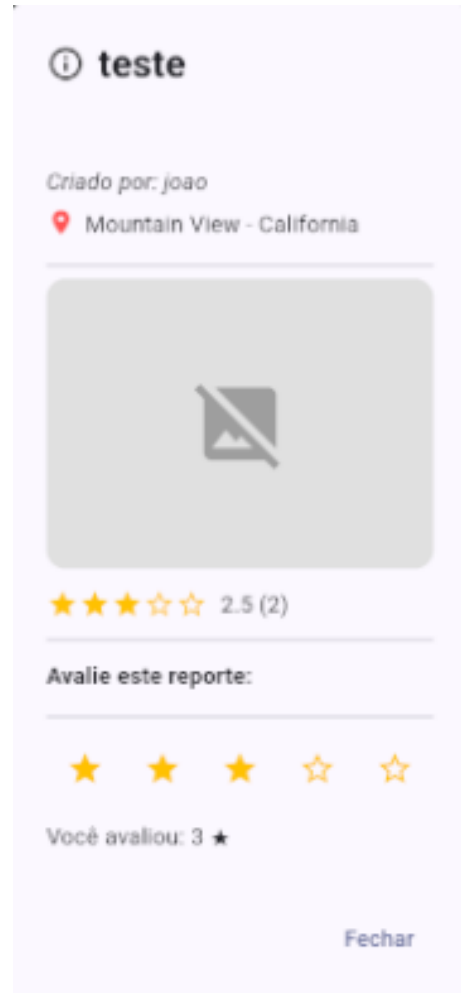


Figura 15. Tela de visualização de reporte

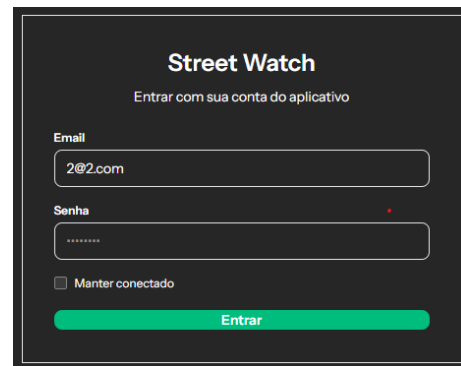


Figura 16. Tela de login do sistema web

Título	Endereço	Tipo	Midia	Votos	Foto	Avaliar
teste	Mountain View - California	-	2★	1	Sem foto	Avaliar
Avaliado pelo painel (2x)						
Mostrando 1 - 1 de 1 reportes						

Figura 17. Tabela do administrador

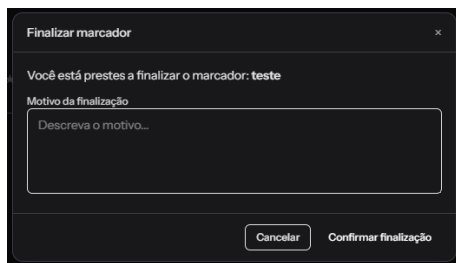


Figura 18. Modal de conclusão de marcador

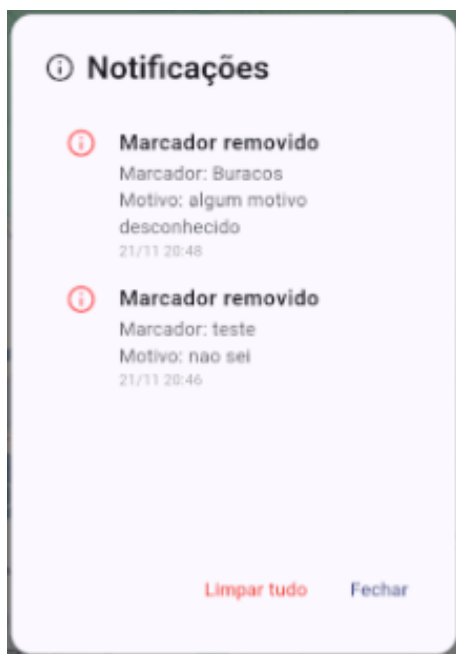


Figura 19. Modal de notificações

ajudando a prefeitura a agilizar soluções de problemas de infraestrutura.

O projeto ainda não foi testado com um número significativo de usuários, não houve integração oficial com sistemas da prefeitura e não foram realizados testes de desempenho ou validação em ambiente de produção.

Para possíveis trabalhos futuros, planeja-se uma implementação mais completa de funcionalidades planejadas e a realização de testes com usuários reais, também seria pertinente explorar outros meios mais baratos para lidar com o fluxo de dados maior que os novos usuários teriam.

Enfim, o protótipo desenvolvido apresentou potencial para ajudar na melhoria da comunicação entre cidadãos e órgãos públicos na tarefa de criar um ambiente urbano mais seguro e agradável de se viver. .

REFERÊNCIAS

[1] Aline Schindler Gomes da Costa, Fausto Ferreira Costa Guimarães e Lúcia Boaventura Pimenta. “Aplicativos de mobilidade e as cidades inteligentes:

Possibilidade de ampliação de acesso e gestão”. Em: *Boletim do Gerenciamento* 42.42 (2024). ISSN: 2595-6531. URL: <https://nppg.org.br/revistas/boletimdogerenciamento/article/view/1167>.

- [2] Gilberto Gampert. “Urban voice: uma abordagem baseada em perfil e recomendação para relatar problemas de infraestrutura urbana”. Dissertação. Programa de Pós-Graduação em Computação Aplicada - Universidade de Passo Fundo, 2016.
- [3] G20. *Santa Maria: a cidade em que mais choveu no mundo sofre com a mudança do clima*. <https://www.g20.org/pt-br/noticias/santa-maria-a-cidade-que-mais-choveu-no-mundo-sofre-com-a-mudanca-do-clima>.
- [4] Andre L. L. Aquino et al. “Cidades Inteligentes, um Novo Paradigma da Sociedade do Conhecimento”. Em: *Blucher Education Proceedings* 1.1 (2015), pp. 165–178.
- [5] Jair Andrade De Araújo, Guaracyane Campelo e Emerson Marinh. *O Impacto Da Infraestrutura Sobre A Pobreza Para O Brasil*. Anais do XLI Encontro Nacional de Economia [Proceedings of the 41st Brazilian Economics Meeting] 226. ANPEC - Associação Nacional dos Centros de Pós-Graduação em Economia [Brazilian Association of Graduate Programs in Economics], 2014. URL: <https://ideas.repec.org/p/anp/en2013/226.html>.
- [6] CNN Brasil. *CNI melhora projeção do PIB*. <https://www.cnnbrasil.com.br/economia/macroeconomia/>.
- [7] Prefeitura Municipal de Santa Maria. *Site oficial da Prefeitura Municipal de Santa Maria*. <https://www.santamaria.rs.gov.br/>. Acesso em: 20 nov. 2024. 2024.
- [8] Miriam Carla do Nascimento Dias. “O uso de tecnologia mobile como ferramenta auxiliar em ações de acessibilidade espacial em ambiente de ensino universitário”. Dissertação. Programa de Pós-Graduação em Ambiente Construído - Universidade Federal de Juiz de Fora, 2015.
- [9] Alef Menezes dos Santos. “SMALL: soluções computacionais em mobilidade urbana para auxiliar na consolidação de cidades inteligentes”. Dissertação. Universidade Federal de Sergipe, São Cristóvão, 2021.
- [10] Roger S. Pressman, Bruce R. Maxim e Reginaldo Arakaki. *Engenharia de Software: Uma Abordagem Profissional*. AMGH, 2010.
- [11] Christian Apiag, Eugene Cadiz e Darllaine Lincopinis. “A Review on PHP Programming Language”. Em: (mai. de 2023).
- [12] Laravel Holdings Inc. *Laravel*. <https://laravel.com/>. Acesso em: 19 nov. 2024. 2024.
- [13] Global Open Source Community Google. *DART*. online. Nov. de 2024. URL: <https://dart.dev/>.
- [14] Global Open Source Community Google. *Flutter*. <https://flutter.dev/>. Acesso em: 20 nov. 2024. 2024.

- [15] Google LLC. *Firebase*. <https://firebase.google.com/?hl=pt>. Acesso em: 13 nov. 2025. 2025.