

TECNOLOGIA E INCLUSÃO: DESENVOLVIMENTO DE PROTÓTIPO DE APLICATIVO DE CROWDSOURCING PARA IDENTIFICAÇÃO DE LOCAIS SEGUROS DA COMUNIDADE LGBTQIA+

Breno Cardoso Bezerra de Menezes¹, Ana Cláudia Garcia Barbosa²

Resumo: A comunidade LGBTQIA+ enfrenta desafios relacionados à violência, discriminação e insegurança em espaços urbanos, o que torna relevante o desenvolvimento de soluções digitais voltadas à identificação de ambientes acolhedores e seguros. Embora existam iniciativas de mapeamento colaborativo e registro de experiências em determinados locais, ainda há espaço para aplicações que associem *crowdsourcing*, geolocalização e práticas de engenharia de software voltadas à confiabilidade, privacidade e segurança dos usuários. Este trabalho tem como objetivo desenvolver uma aplicação colaborativa para identificação, avaliação e compartilhamento de informações sobre locais seguros destinados à comunidade LGBTQIA+. O protótipo foi desenvolvido com Laravel no *backend*, React.js no *frontend* e Inertia.js como camada de integração, utilizando a Google Maps API para visualização espacial dos locais cadastrados. A arquitetura foi organizada com foco na separação de responsabilidades entre *Actions*, *Services*, *Requests* e *Policies*. A validação técnica foi realizada por meio de testes automatizados com Pest e análise estática com PHPStan. Constatou-se que o protótipo oferece uma base funcional e tecnicamente consistente para apoiar o mapeamento colaborativo de espaços seguros.

Palavras-chave: crowdsourcing; espaços seguros; LGBTQIA+; Laravel; geolocalização.

ABSTRACT: The LGBTQIA+ community faces challenges related to violence, discrimination, and lack of safety in urban spaces, which makes the development of digital solutions aimed at identifying welcoming and safe environments relevant. Although there are initiatives for collaborative mapping and recording experiences in specific places, there is still room for applications that combine *crowdsourcing*, geolocation, and software engineering

¹Acadêmico, Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), brenocbmnz@gmail.com

²Orientadora, Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), agb@unesc.net

practices focused on reliability, privacy, and user safety. This work aims to develop a collaborative application for identifying, evaluating, and sharing information about safe spaces intended for the LGBTQIA+ community. The prototype was developed with Laravel for the *backend*, React.js for the *frontend*, and Inertia.js as the integration layer, using the Google Maps API for the spatial visualization of registered spaces. The architecture was organized with a focus on separation of responsibilities between *Actions*, *Services*, *Requests*, and *Policies*. The technical validation was carried out through automated tests with Pest and static analysis with PHPStan. It was found that the prototype provides a functional and technically consistent foundation to support the collaborative mapping of safe spaces.

Keywords: crowdsourcing; safe spaces; LGBTQIA+; Laravel; geolocation.

1 INTRODUÇÃO

O avanço das tecnologias consolidou as aplicações como ferramentas centrais de interação social, comunicação e acesso a informações. Aplicativos podem oferecer praticidade e acessibilidade, permitindo que indivíduos tenham acesso a soluções que impactam diretamente o seu cotidiano urbano (Henry; Abou-Zahra; Brewer, 2014).

Tais tecnologias podem ser pensadas em diferentes maneiras, sendo uma delas a utilização de modelos colaborativos, como, por exemplo, o *crowdsourcing*, que se destaca como uma estratégia de inovação social, uma vez que possibilita a participação dos próprios usuários na coleta, avaliação e compartilhamento de informações (Howe, 2006).

O *crowdsourcing* pode ser compreendido como um modelo de produção colaborativa no qual tarefas, informações ou soluções são obtidas a partir da contribuição de um grupo amplo de participantes, geralmente mediado por plataformas digitais.

Diferentemente de modelos centralizados, nos quais a coleta e a validação das informações dependem exclusivamente de uma instituição ou equipe técnica, o *crowdsourcing* distribui parte desse processo entre os próprios usuários, permitindo que experiências individuais sejam convertidas em dados coletivos, o que torna o modelo especialmente relevante em contextos nos quais a informação depende da vivência cotidiana dos sujeitos, como ocorre no mapeamento de problemas urbanos, serviços públicos, acessibilidade, segurança e percepção social dos espaços.

De acordo com Certomà, Corriní e Rizzi (2020), iniciativas de

crowdsourcing financiadas pela Comissão Europeia têm se mostrado relevantes na formulação de políticas públicas, reforçando a importância da participação digital em processos democráticos. Salim e Haque (2015) complementam essa perspectiva ao afirmar que tais práticas podem ter um papel importante na obtenção de dados em ambientes urbanos, tornando o *crowdsourcing* uma ferramenta estratégica para enfrentar problemas sociais complexos.

No contexto da comunidade LGBTQIA+, a utilização de tecnologias colaborativas se mostra ainda mais relevante, tendo em vista que essa população (podendo também ser conhecida como *queer*), historicamente marginalizada, enfrenta situações de violência, discriminação e exclusão social (Silva, 2020; Cerqueira; Bueno, 2025).

Com isso em mente, a criação de soluções que auxiliem essa população, como a identificação de espaços seguros, surge como uma estratégia de resistência, acolhimento e afirmação de direitos humanos, possibilitando que pessoas LGBTQIA+ vivenciem ambientes mais respeitosos e protegidos (Lamont, 2020).

Locais seguros podem ser entendidos como espaços nos quais indivíduos se sentem protegidos, respeitados e livres para expressar sua identidade sem medo de violência, discriminação ou constrangimento (Kirby et al., 2021). Entretanto, a definição de um espaço como seguro não é neutra nem universal, pois depende das experiências concretas dos sujeitos que o frequentam.

Para apoiar a compreensão das necessidades do público-alvo, foi elaborado um mapa de empatia, apresentado na Figura 1.

Esse recurso sintetiza percepções, dores e expectativas de uma pessoa LGBTQIA+ ao buscar espaços seguros, evidenciando preocupações relacionadas à insegurança em locais desconhecidos, à dificuldade de encontrar informações confiáveis e ao desejo de frequentar ambientes onde sua identidade seja respeitada, contribuindo para aproximar justificativa social da aplicação proposta.

Por esse motivo, o mapeamento colaborativo desses locais não deve ser compreendido apenas como uma representação técnica do território, mas como uma prática social e política de visibilidade (LaRochelle, 2020).

Figura 1 - Mapa de empatia relacionado à busca por espaços seguros

Espaços Seguros



Fonte: Elaborado pelo autor (2026).

Essa perspectiva de locais seguros tem sobreposição com as ideias de cartografia crítica, ao reconhecer que mapas não apenas descrevem espaços, mas também expressam relações de poder, disputas de narrativa e formas de pertencimento (Crampton, 2010).

Sob a ótica da cartografia crítica, mapas colaborativos que apontam espaços seguros tornam-se instrumentos de inclusão e empoderamento, uma vez que permitem que sujeitos historicamente silenciados reivindiquem seus próprios espaços e narrativas (Harley, 2011; Crampton, 2010; Girardi, 2014), evidenciando que o desenvolvimento de uma aplicação colaborativa baseada em *crowdsourcing* representa uma ferramenta tecnológica de relevância social, pois possibilita identificar, avaliar e compartilhar informações sobre locais considerados seguros ou acolhedores para a comunidade LGBTQIA+.

Apesar do potencial dessas soluções, sistemas que prezam pela segurança dos usuários exigem cuidados que vão além da disponibilização de funcionalidades para identificar locais seguros de forma colaborativa (Stach et al., 2022).

Informações que podem envolver dados sensíveis precisam ser tratadas com segurança para serem garantidas a privacidade e a confiança dos usuários, requerendo que o desenvolvimento tenha estrutura técnica que contribua para proteger os dados, organizar os fluxos de uso e reduzir riscos associados ao funcionamento do sistema, como vazamento de dados e exposição de informações sensíveis (Chaves; Benitti, 2025).

Com base nesse contexto, a prototipação da aplicação terá foco em práticas de desenvolvimento que prezam também pela segurança do usuário digital, utilizando Laravel no *backend* (Laravel, 2026c), React na construção do *frontend* (React, 2026) e Inertia.js como camada de integração entre o servidor e a interface (Inertia.js, 2026).

A escolha destas tecnologias para o desenvolvimento justifica-se por proporcionar uma estrutura organizada, baseada no padrão *Model-View-Controller* (MVC), que favorece a separação de responsabilidades entre a manipulação dos dados, o controle das requisições e a apresentação das informações ao usuário (Bucanek, 2009; Tu, 2023; Ceri et al., 2003).

Essa organização contribui para a manutenção, a testabilidade e a evolução da aplicação, especialmente em sistemas que lidam com autenticação, persistência de dados e validação de entradas. (Abduramani et al., 2025).

Tal estrutura também foi associada a práticas de maior segregação arquitetural, com a organização de responsabilidades em camadas, ações e serviços específicos (*Actions* e *Services* evitando a concentração excessiva de lógica nos controladores e tornar os fluxos críticos mais previsíveis e auditáveis (Aniche et al., 2018; Hayashi; Minami; Saeki, 2017).

Além disso, o banco de dados será estruturado por meio de *migrations* e o Eloquent (responsável por relacionar as entidades da aplicação às tabelas do banco de dados) (Laravel, 2026b), recursos do Laravel que permitem versionar e compartilhar a definição do esquema da aplicação, facilitando a criação e a alteração de tabelas, colunas e relacionamentos durante o desenvolvimento (Laravel, 2026a).

Para a integração destas tecnologias com a funcionalidade de geolocalização e mapas interativos, será utilizado o Google Maps API (interface de programação de aplicações que permite a comunicação com serviços externos), possibilitando a visualização espacial e interação com mapas para a identificação de locais seguros (Google Maps Platform, 2026a; Google Maps Platform, 2026b).

A validação técnica do protótipo foi realizada por meio de testes

automatizados com o Pest e análise estática de código com PHPStan, voltados para testagem do sistema (testes unitários, de funcionalidade e arquiteturais) e análise estática do código, respectivamente. (PHPStan, 2026).

A utilização de Laravel (além do React, ligado ao backend pelo Inertia), compõe uma decisão metodológica voltada à construção de uma aplicação mais segura, manutenível e adequada ao tratamento de informações sensíveis (Bombarda; Bonfanti; Gargantini, 2025; Abduramani et al., 2025).

O objetivo geral do trabalho é auxiliar pessoas a identificar informações sobre locais seguros destinados à comunidade LGBTQIA+ por meio de uma aplicação de *crowdsourcing*.

Para alcançar esse objetivo, foram definidos objetivos específicos, que são: Modelar as interfaces e os fluxos de navegação da aplicação por meio da prototipação das telas, considerando clareza na navegação e coerência funcional; Definir a arquitetura de software da aplicação, contemplando autenticação de usuários, persistência de dados, integração com serviços de geolocalização e organização das camadas do sistema; Desenvolver as funcionalidades colaborativas da aplicação, incluindo consulta, avaliação, denúncia e compartilhamento de informações sobre locais seguros; Integrar mapas interativos à aplicação por meio de APIs de geolocalização, permitindo a visualização e localização dos espaços cadastrados; Aplicar práticas de validação, testes automatizados e mitigação de vulnerabilidades para verificar a qualidade, a confiabilidade e a segurança do sistema.

2 TRABALHOS CORRELATOS

Diversos estudos têm abordado a crescente preocupação com a segurança em ambientes urbanos, destacando o impacto da violência na vida das pessoas. Pesquisas como as de Ricardo, Siqueira e Marques (2013) discutem como o cenário de insegurança pode criar barreiras que limitam a liberdade e afetam, de forma mais intensa, grupos historicamente marginalizados. Além das discussões teóricas, é relevante destacar iniciativas práticas e tecnológicas já existentes, como, por exemplo, o Refuge Restroom, um aplicativo colaborativo focado no mapeamento de banheiros públicos seguros para pessoas trans e não binárias, sendo de extrema relevância para a vivência LGBTQIA+ (LaRochelle, 2020).

Ainda no âmbito das aplicações colaborativas, encontram-se algumas outras aplicações utilizando *crowdsourcing*. Um exemplo emble-

mático é a plataforma Wikimapps, concebida para a criação, manutenção e hospedagem de aplicações colaborativas baseadas em mapas. Trata-se de uma generalização do projeto Wikicrimes, originalmente voltado à coleta participativa de informações sobre criminalidade. O Wikimapps configura-se como uma solução integrada ("tudo-em-um"), pois incorpora seus próprios mecanismos de exibição, coleta e processamento de dados (Furtado et al., 2010).

Outro caso notável é o do Google Forms, que exemplifica uma aplicação genérica voltada à criação de questionários. Sua versatilidade permitiu sua adoção em múltiplos contextos, como na educação colaborativa (Goodchild, 2007), concentrando-se na coleta explícita de informações.

O projeto *Queering the Map* se mostra muito pertinente para este trabalho, já que se trata de uma plataforma digital de mapeamento colaborativo na qual usuários podem marcar lugares no mapa e associá-los a relatos, memórias e experiências *queer* (pessoas LGBTQIA+) vividas nesses espaços (Queering the Map, 2026), mostrando o potencial de aplicações colaborativas de mapeamento para dar visibilidade a experiências LGBTQIA+ relacionadas aos espaços sociais.

A partir desses exemplos, pode-se constatar que aplicações de *crowdsourcing* possuem relevância social e já são amplamente utilizadas em diferentes contextos, porém, essas iniciativas possuem escopos distintos, seja por se restringirem a categorias específicas de locais, seja por não explicitarem cuidados arquiteturais voltados ao desenvolvimento do sistema.

3 MATERIAIS E MÉTODOS

Esta pesquisa caracteriza-se como aplicada, de natureza tecnológica, pois tem como finalidade o desenvolvimento de um protótipo funcional de aplicação colaborativa voltada à identificação, avaliação e compartilhamento de informações sobre espaços seguros para a comunidade LGBTQIA+. A proposta parte de um problema social relacionado à dificuldade de localizar ambientes acolhedores e seguros, traduzindo essa demanda em uma solução computacional baseada em *crowdsourcing*, geolocalização e participação comunitária, abordagem que se observa em sistemas colaborativos (Basak et al., 2020).

O desenvolvimento do protótipo foi orientado por práticas de engenharia de software voltadas à organização, confiabilidade, privacidade e segurança da aplicação. Entre essas práticas, destaca-se a adoção da

arquitetura Model-View-Controller (MVC), a segregação de responsabilidades por meio de camadas, *Actions* e *Services*, a validação de entradas, o controle de acesso, a autenticação de usuários, a modelagem estruturada do banco de dados por meio de *migrations*, a prototipação dos fluxos de interface e a realização de testes automatizados e análise estática de código (Jones, 2009). O banco de dados utilizado no protótipo foi o SQLite, um mecanismo de banco de dados SQL incorporado à aplicação, que dispensa um processo de servidor separado e armazena a base completa em um único arquivo, simplificando a configuração e a reprodução do ambiente de desenvolvimento (SQLite, 2025).

A aplicação foi desenvolvida utilizando Laravel no *backend*, React.js no *frontend* e Inertia.js como camada de integração entre servidor e interface. Também foram incorporados recursos de mapas interativos e geolocalização por meio da Google Maps API, permitindo a representação espacial dos locais cadastrados. O processo metodológico foi organizado em etapas que envolveram a prototipação das interfaces, definição da arquitetura do sistema, implementação das funcionalidades colaborativas, integração com serviços de mapas e validação técnica do protótipo.

3.1 APLICAÇÃO PROPOSTA

O protótipo foi concebido como uma aplicação web com design responsivo, consistindo em um programa colaborativo destinado à identificação, avaliação e compartilhamento de informações sobre locais considerados seguros ou acolhedores para a comunidade LGBTQIA+, permitindo que usuários possam consultar espaços previamente cadastrados, visualizar sua localização em mapa, registrar novos locais, além de avaliar experiências e sinalizar conteúdos inadequados ou potencialmente inseguros.

A estrutura da aplicação foi pensada a partir da relação entre participação comunitária, por meio de *crowdsourcing*, e a proteção informacional dos usuários, compreendida como o conjunto de medidas de engenharia de software para construir um sistema robusto, buscando oferecer um sistema que não apenas tenha como objetivo identificar locais seguros, mas que também seja desenvolvido de forma segura.

De modo geral, o protótipo foi organizado em torno de quatro fluxos principais: cadastro e autenticação de usuários; consulta de locais seguros; registro e avaliação de espaços; e denúncia ou sinalização de informações inadequadas.

Foram pensados quatro menus para a aplicação; o menu Início,

onde é mostrada uma mensagem de boas vindas ao usuário, além de links para outras partes do sistema, o menu Locais onde se pode consultar uma listagem dos locais, com filtragem e aba de favoritos, o menu Mapa, onde é mostrado o mapa colaborativo da aplicação, com opção de registro e consulta espacial dos locais, além de filtragem com visualização no próprio mapa, e o menu Perfil onde o usuário pode trocar suas preferências relacionadas ao próprio perfil, como seu e-mail, pronomes, modo de tela, além de verificar seus locais favoritos, cadastrados e avaliações.

No contexto da aplicação, os locais seguros foram tratados como espaços cadastrados e avaliados a partir da percepção dos usuários. Essa escolha parte do entendimento de que a segurança de um ambiente não pode ser definida apenas por características físicas ou institucionais, mas também pelas experiências concretas de quem frequenta o local. Dessa forma, o protótipo considera que a avaliação comunitária é um elemento central para identificar espaços acolhedores, respeitosos e acessíveis à comunidade LGBTQIA+.

Cada local cadastrado foi planejado para reunir informações básicas, como nome do estabelecimento ou espaço, endereço, localização geográfica, categoria, características e avaliações dos usuários.

O sistema prevê, então, funcionalidades voltadas à sinalização de experiências negativas ou conteúdos inadequados, permitindo que o usuário contribua não apenas para indicar espaços seguros, mas também para qualificar e revisar as informações disponíveis na plataforma. Essas sinalizações, também conhecidas como denúncias, são analisadas pelas contas com poderes de administração no sistema.

A organização dessas informações busca equilibrar colaboração e segurança. Por um lado, o sistema depende da contribuição dos usuários para formar uma base de dados útil e contextualizada. Por outro, essa participação exige critérios mínimos de estruturação, validação e controle, a fim de reduzir inconsistências, evitar uso indevido da aplicação e proteger informações que possam expor usuários ou comunidades vulneráveis.

3.1.1 Mapas Interativos e *Crowdsourcing*

A aplicação utiliza recursos de mapas interativos como elemento central de navegação e consulta dos locais cadastrados. A integração com a Google Maps API foi adotada para permitir a visualização geográfica dos espaços, a marcação de pontos no mapa e a localização dos ambientes registrados pelos usuários, utilizando dados e serviços geográficos dispo-

nibilizados pela plataforma.

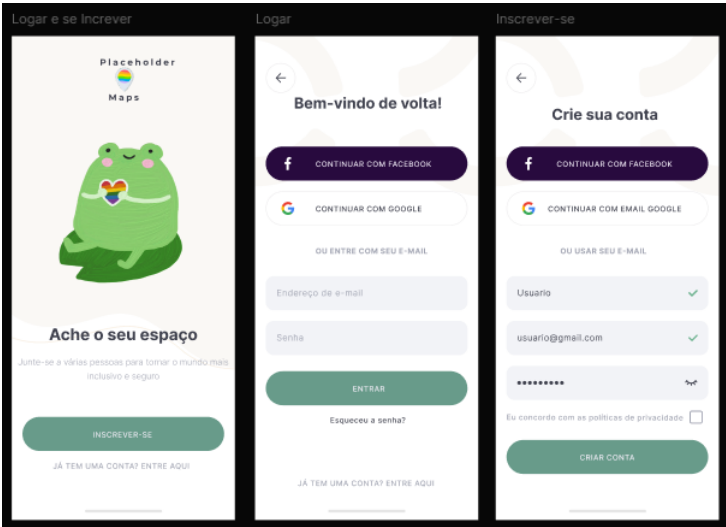
O uso de mapas no protótipo serve como mecanismo de organização espacial das informações, permitindo que os usuários compreendam a distribuição dos locais cadastrados e encontrem espaços próximos ou relevantes para sua realidade, funcionando como interface de acesso ao conhecimento coletivo produzido pela comunidade.

Portanto, o *crowdsourcing* foi incorporado ao sistema por meio da possibilidade de os próprios usuários contribuírem com o cadastro, avaliação e atualização das informações sobre os locais, utilizando recursos de geolocalização com práticas colaborativas de compartilhamento de experiências, que promovem a visibilidade de experiências vividas por pessoas LGBTQIA+ (Kirby et al., 2021), permitindo que experiências individuais sejam convertidas em dados coletivos.

3.2 DESENVOLVIMENTO DO PROTÓTIPO

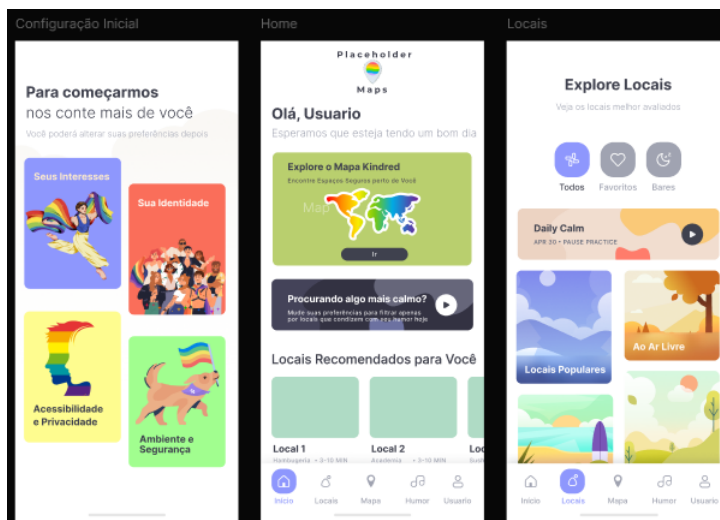
A primeira etapa do desenvolvimento foi a prototipação das telas no Figma, que serviram de base para o eventual desenvolvimento:

Figura 2 - Prototipação Figma 1



Fonte: Elaborado pelo autor (2025).

Figura 3 - Prototipação Figma 2



Fonte: Elaborado pelo autor (2025).

Os protótipos do Figma tiveram papel importante no planejamento visual e funcional da aplicação, pois permitiram antecipar a organização das telas, os fluxos de navegação e a disposição dos elementos de interface antes da implementação do sistema, facilitando a análise da experiência do usuário, a identificação de possíveis inconsistências e a definição de uma identidade visual coerente para o protótipo (Lauff; Kotys-Schwartz; Rentschler, 2018).

Após a prototipação das telas, foi então realizado o desenvolvimento do sistema, que foi organizado a partir de uma arquitetura *full-stack*, utilizando Laravel para construção do *backend*, React.js para a construção do *frontend* e Inertia.js como camada de integração entre as duas partes. Essa combinação permitiu construir uma aplicação web dinâmica sem a necessidade de separar completamente o sistema em uma API independente e uma aplicação cliente isolada. Com isso, as rotas, os controladores, as ações do sistema e os componentes de interface puderam ser desenvolvidos de forma integrada.

3.2.1 Tecnologias utilizadas: Laravel, React E Inertia.js

A escolha das tecnologias foi orientada por critérios de organização, manutenção, produtividade e segurança.

O Laravel foi utilizado como *framework* principal do *backend*, sendo responsável pela organização das rotas, autenticação, validação de requisições, persistência de dados e das ações do sistema, o que auxilia a construção de uma base estruturada, manutenível e adequada ao de-

envolvimento de sistemas que exigem controle de acesso, segurança e consistência nos dados manipulados.

No *frontend*, o React.js foi empregado para a construção das interfaces da aplicação. A utilização de componentes favorece a reutilização de elementos visuais, a padronização das telas e a manutenção do código, aspecto que se prova importante quando estão envolvidos diferentes fluxos de navegação, como login, consulta de locais, visualização no mapa, cadastro de espaços, avaliações e denúncias.

A integração entre Laravel e React.js foi realizada por meio do Inertia.js, ferramenta que permite que o Laravel envie dados diretamente para componentes React, evitando a necessidade de construir uma API completa apenas para comunicação interna entre servidor e interface, mantendo uma estrutura integrada e que ofereça ao usuário uma experiência mais dinâmica e próxima de uma aplicação moderna de página única.

Para a construção do banco de dados foram utilizadas as *migrations* do Laravel para estruturar e versionar o banco de dados da aplicação. Esse recurso permite definir, por meio de arquivos de código, a criação e alteração das tabelas, campos, chaves e relacionamentos necessários ao funcionamento do sistema. Dessa forma, a modelagem do banco deixa de depender de alterações manuais realizadas diretamente no sistema gerenciador de banco de dados, passando a fazer parte do próprio histórico de desenvolvimento da aplicação, contribuindo para organizar entidades como usuários, locais cadastrados, avaliações e denúncias, garantindo maior controle sobre a evolução da estrutura de dados e facilitando a reprodução do ambiente em diferentes etapas do projeto.

A validação técnica do protótipo foi planejada a partir da combinação de testes automatizados e análise estática de código. Para isso, foram utilizados o Pest, voltado à execução de testes unitários, testes de *feature* e testes arquiteturais, e o PHPStan, destinado à análise estática do código PHP.

3.2.2 Arquitetura MVC e Segregação de Responsabilidades

A arquitetura da aplicação foi fundamentada no padrão Model-View-Controller (MVC), utilizado pelo Laravel para separar a manipulação dos dados, o controle das requisições e a apresentação das informações ao usuário. Nesse modelo, os *Models* representam as entidades da aplicação e sua persistência no banco de dados; os *Controllers* coordenam as requisições e o fluxo entre as diferentes camadas; e a *View*, neste projeto,

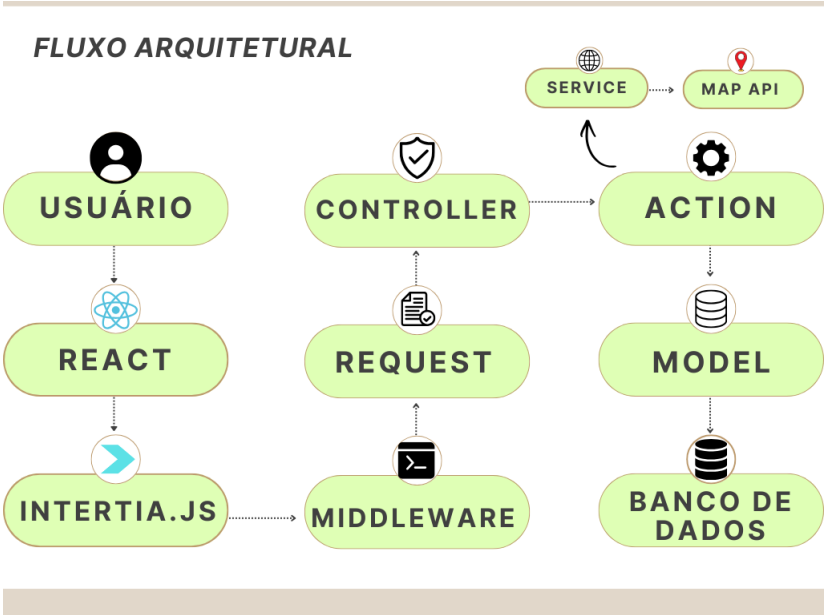
é composta por componentes React apresentados ao usuário e integrados ao servidor por meio do Inertia.js.

Apesar de a estrutura MVC fornecer a organização principal do sistema, o protótipo adotou uma separação complementar de responsabilidades por meio de *Middlewares*, *Requests*, *Actions* e *Services*. Os *Middlewares* atuam antes da execução dos controladores, realizando verificações relacionadas ao acesso e à autenticação. Os *Requests* validam os dados de entrada enviados pela interface antes que sejam processados pelos controladores, reduzindo a possibilidade de informações inválidas cheguem às demais camadas da aplicação.

As *Actions* concentram operações específicas do sistema, como ações iniciadas pelo usuário, enquanto os *Services* são utilizados principalmente para responsabilidades auxiliares e integrações externas, como a comunicação com a Google Maps API.

O fluxo geral parte da interação do usuário com os componentes React. Os dados são encaminhados pelo Inertia.js, passam pelos *Middlewares* e pelos *Requests* de validação e, posteriormente, são direcionados aos *Controllers*. Conforme a operação solicitada, o controlador aciona uma *Action*, que pode interagir com os *Models* responsáveis pelo acesso e pela persistência das informações no banco de dados. A Figura ?? apresenta uma visão sintetizada desse fluxo arquitetural.

Figura 4 - Fluxo arquitetural da aplicação



Fonte: Elaborado pelo autor (2026).

Essa organização favorece a manutenção do código, a testabili-

dade das funcionalidades e a compreensão dos fluxos da aplicação.

4 RESULTADOS E DISCUSSÃO

A análise dos resultados foi organizada com base nos objetivos específicos definidos no trabalho, considerando a implementação das funcionalidades propostas, a estrutura técnica adotada, os testes realizados e a comparação com trabalhos correlatos apresentados anteriormente.

De modo geral, o desenvolvimento do protótipo permitiu transformar a proposta conceitual do trabalho em uma solução computacional funcional, articulando recursos de *crowdsourcing*, geolocalização, mapas interativos, autenticação de usuários, cadastro de locais e avaliações. A adoção de práticas de engenharia de software, como arquitetura MVC, segregação de responsabilidades em *Actions* e *Services*, validação de entradas, testes automatizados e análise estática de código, contribuiu para organizar o sistema de forma mais manutenível, previsível e adequada ao tratamento de informações sensíveis.

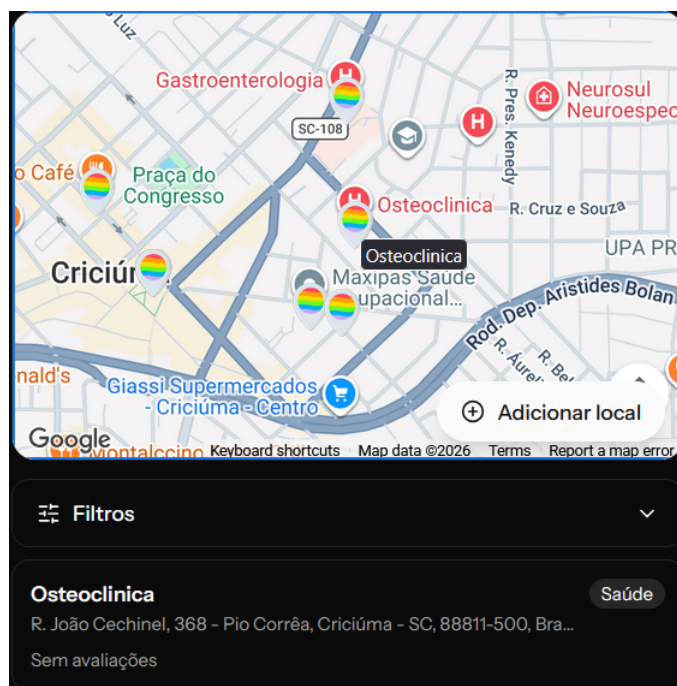
4.1 ANÁLISE DOS OBJETIVOS ESPECÍFICOS

Os objetivos específicos definidos para o trabalho foram atendidos de forma integrada ao longo do desenvolvimento do protótipo. A modelagem das interfaces foi realizada no Figma, permitindo planejar os fluxos de navegação e orientar a construção do *frontend*. Em seguida, a aplicação foi estruturada com Laravel no *backend*, React.js no *frontend* e Inertia.js como camada de integração, adotando o padrão MVC e uma organização complementar baseada em *Actions* e *Services* para reforçar a separação de responsabilidades.

As funcionalidades colaborativas propostas foram materializadas por meio dos fluxos de consulta, cadastro, avaliação e sinalização de locais seguros, permitindo que experiências individuais fossem convertidas em informações compartilhadas e seguras.

A integração com a Google Maps API possibilitou a visualização espacial dos locais cadastrados, tornando o mapa um elemento central da experiência da aplicação e aproximando o protótipo da proposta de mapeamento colaborativo almejado. Os locais são representados no mapa por meio de pins com as cores do arco-íris, sinalizando que aquele local possui uma avaliação. O mapa também oferece opções de filtragem por categoria e características do espaço.

Figura 5 - Tela de mapa interativo implementada no protótipo com a utilização da Google API



Fonte: Elaborado pelo autor (2026).

A validação técnica do protótipo teve como objetivo verificar se as principais funcionalidades da aplicação apresentavam comportamento compatível com o esperado, além de avaliar a organização interna do código e a aderência à arquitetura proposta. Essa etapa relaciona-se diretamente ao objetivo específico de aplicar práticas de validação, testes automatizados e mitigação de vulnerabilidades para verificar a qualidade, a confiabilidade e a segurança do sistema. Para isso, foram utilizadas ferramentas de análise estática e testes automatizados, contemplando tanto aspectos funcionais quanto arquiteturais da aplicação.

A análise estática foi realizada com o PHPStan, em conjunto com o Larastan, configurado no nível máximo. Essa ferramenta permite verificar o código PHP sem executá-lo, identificando possíveis inconsistências de tipagem, chamadas incorretas de métodos, variáveis indefinidas e usos inadequados de recursos do Laravel. O resultado obtido foi de zero erros encontrados, indicando que o código analisado não apresentou inconsistências detectáveis estaticamente pela ferramenta. Esse resultado contribui para a confiabilidade do protótipo, pois reduz a possibilidade de falhas estruturais relacionadas ao uso incorreto da linguagem ou do *framework*.

Figura 6 - Resultado da análise estática com PHPStan

```
brenocbmnz@B-LAPTOP:~/projects/safer-app$ vendor/bin/phpstan analyse
[OK] No errors
```

Fonte: Elaborado pelo autor (2026).

Também foram executados testes automatizados com o Pest, contemplando verificações de arquitetura, testes unitários e testes de *feature*. De forma geral, esses testes foram utilizados para validar tanto a separação de responsabilidades entre as camadas da aplicação quanto o comportamento de classes isoladas e o funcionamento integrado dos principais fluxos do sistema. A Tabela 1 resume os tipos de teste empregados e seus respectivos objetivos.

Tabela 1 - Tipos de testes utilizados na validação técnica

Tipo de teste	Finalidade e aspectos verificados
Testes de arquitetura	Verificam se a separação de responsabilidades entre camadas está sendo respeitada, observando se as classes <i>Action</i> são finais e imutáveis, além da ausência de dependência entre <i>Actions</i> , <i>Models</i> e <i>Controllers</i> .
Testes unitários	Verificam o comportamento isolado de classes específicas, como modelos Eloquent e <i>Actions</i> (ações) do sistema. Foram avaliados atributos visíveis, conversões de tipos, relacionamentos entre entidades e regras específicas, como a lógica de favoritos, sem dependência direta do banco de dados real.
Testes de <i>feature</i>	Contempla os principais fluxos integrados da aplicação, incluindo autenticação, gerenciamento de locais, favoritos, avaliações e integração com a Google Places API, permitindo verificar o comportamento conjunto dos principais <i>controllers</i> do sistema.

Fonte: Elaborado pelo autor (2026).

Na execução da suíte automatizada com o Pest, foram obtidos 252 testes aprovados, totalizando 722 asserções, sem falhas, como pode ser verificado na Figura 7.

Figura 7 - Resultado geral da execução dos testes automatizados com Pest

```
brenocbmz@B-LAPTOP:~/projects/safer-app$ ./vendor/bin/pest

Tests: 252 passed (722 assertions)
Duration: 13.40s
```

Fonte: Elaborado pelo autor (2026).

De forma geral, os resultados da análise estática e dos testes automatizados indicam que o protótipo apresenta consistência funcional e arquitetural. A ausência de erros no PHPStan e a aprovação integral da suíte de testes com o Pest sugerem que a aplicação foi desenvolvida com atenção à separação de responsabilidades, à correção dos fluxos principais e à redução de falhas estruturais.

A Tabela 2 sintetiza a relação entre os objetivos específicos e os resultados alcançados.

Tabela 2 - Relação entre objetivos específicos e resultados alcançados

Objetivo específico	Resultado alcançado
Modelar as interfaces e os fluxos de navegação da aplicação.	Objetivo atendido por meio da prototipação das telas no Figma, que orientou a organização visual e funcional do protótipo.
Definir a arquitetura de software da aplicação.	Objetivo atendido com a adoção de Laravel, React.js, Inertia.js, MVC, <i>Actions</i> , <i>Services</i> , autenticação e persistência de dados.
Desenvolver funcionalidades colaborativas.	Objetivo atendido com a implementação dos fluxos de cadastro, consulta, avaliação e denúncia de locais seguros.
Integrar mapas interativos à aplicação.	Objetivo atendido com a utilização da Google Maps API para visualização espacial dos locais cadastrados.
Aplicar práticas de validação, testes e mitigação de vulnerabilidades.	Objetivo atendido, considerando a realização de validações técnicas e testes no protótipo, que contribuem para a mitigação de vulnerabilidades e para a qualidade, a confiabilidade e a segurança do sistema.

Fonte: Elaborado pelo autor (2026).

4.2 COMPARAÇÃO COM OS TRABALHOS CORRELATOS

Assim como o Refuge Restroom e o Wikimapps, o protótipo proposto utiliza a contribuição dos usuários para associar informações a locais específicos, transformando experiências individuais em dados coletivos. No entanto, diferencia-se por possuir um recorte voltado à comunidade LGBTQIA+ e por não se restringir a uma única categoria de espaço, como banheiros públicos, nem a uma plataforma genérica de mapeamento.

Em comparação ao Google Forms, o sistema também se diferencia por não se limitar à coleta de respostas, mas por organizar as informações enviadas pelos usuários em uma interface espacial e interativa e vinculada a um mapa.

O trabalho que mais se assemelha ao protótipo, e o que serviu de inspiração para este trabalho, é o *Queering the Map*, por compreender o mapa como espaço de expressão, memória e visibilização de experiências LGBTQIA+. Ambos os projetos reconhecem que lugares não são apenas pontos geográficos, mas espaços carregados de significados sociais, afetivos e políticos para a comunidade (LaRochelle, 2020). Entretanto, enquanto o *Queering the Map* concentra-se principalmente no registro de narrativas queer associadas a diferentes lugares do mundo, o protótipo desenvolvido neste trabalho busca direcionar o mapeamento para uma finalidade prática: identificar, avaliar e compartilhar informações sobre locais considerados seguros ou acolhedores.

Desse modo, a principal contribuição do protótipo está na combinação entre mapeamento colaborativo, foco em locais seguros para a comunidade LGBTQIA+ e preocupação explícita com a estrutura técnica da aplicação. Diferentemente de soluções voltadas apenas à coleta ou à exposição de informações, este trabalho busca associar a proposta social do sistema a práticas de engenharia de software voltadas à organização, confiabilidade, privacidade e proteção informacional dos usuários.

5 CONCLUSÃO

O presente trabalho alcançou seus objetivos com a criação de um protótipo colaborativo para identificação, avaliação e compartilhamento de informações sobre locais seguros destinados à comunidade LGBTQIA+. Com o desenvolvimento do projeto, foi possível integrar conceitos de *crowdsourcing*, mapas interativos e cartografia crítica a uma solução computacional estruturada com Laravel, React.js, Inertia.js e Google Maps API, demonstrando que o mapeamento colaborativo de espaços seguros pode ser

tratado não apenas como uma funcionalidade técnica, mas como uma ferramenta de cunho social, embasada na participação comunitária para a identificação de locais acolhedores à circulação de pessoas LGBTQIA+.

Como limitação, destaca-se que a validação realizada concentrou-se na dimensão técnica do protótipo, sem a participação de usuários reais nesta etapa. Assim, embora os resultados indiquem que a aplicação possui uma base funcional, organizada e tecnicamente validada, ainda não é possível avaliar de forma definitiva sua usabilidade, percepção de segurança e efetividade social junto ao público-alvo.

Como trabalhos futuros, propõe-se realizar testes com usuários LGBTQIA+ e potenciais participantes da aplicação, a fim de analisar a experiência de uso, a clareza dos fluxos e a confiança nas informações apresentadas. Também poderão ser investigadas melhorias nos mecanismos de moderação, privacidade, acessibilidade e expansão da base colaborativa de locais seguros.

DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL

Os autores declaram que ferramentas de Inteligência Artificial generativa foram utilizadas exclusivamente como apoio auxiliar à elaboração deste trabalho, incluindo assistência na revisão linguística, estruturação textual e organização de ideias. Nenhuma ferramenta de IA foi utilizada para substituição da autoria intelectual, análise científica, interpretação dos resultados ou tomada de decisões acadêmicas. A responsabilidade integral pelo conteúdo, originalidade, veracidade das informações e conformidade com os princípios de integridade científica permanece integralmente atribuída aos autores.

REFERÊNCIAS

ABDURAMANI, E. et al. **Testing strategies in mvc-based applications: Model, view, and controller strategies**. JNSM - Journal of Natural Sciences and Mathematics of UT, v. 10, n. 19–20, p. 342–348. 2025. Disponível em: <<https://doi.org/10.62792/ut.jnsn.v10.i19-20.p3197>>. Acesso em: 03 mar. 2026.

ANICHE, M. et al. **Code smells for model-view-controller architectures**. Empirical Software Engineering, v. 23, n. 4, p. 2121–2157. 2018. Disponível em: <<https://doi.org/10.1007/s10664-017-9540-2>>. Acesso em: 04 mai. 2026.

BASAK, J. et al. **A crowdsourcing based information system framework for coordinated disaster management and building community**

resilience. In: **Proceedings of the 21st International Conference on Distributed Computing and Networking.** New York, NY, USA: Association for Computing Machinery, 2020. 2020. (ICDCN '20), p. 1–6. Disponível em: <https://doi.org/10.1145/3369740.3372730>. Acesso em: 24 mar. 2026.

BOMBARDA, A.; BONFANTI, S.; GARGANTINI, A. **Integrating formal specifications in the development and testing of uis by formal model–view–controller pattern.** International Journal on Software Tools for Technology Transfer. 2025. Disponível em: <https://doi.org/10.1007/s10009-025-00812-2>. Acesso em: 20 mai. 2026.

BUCANEK, J. Model-view-controller pattern. In: **Learn Objective-C for Java Developers.** Berkeley, CA: Apress, 2009. p. 353–402. Disponível em: https://doi.org/10.1007/978-1-4302-2370-2_20. Acesso em: 08 abr. 2026.

CERI, S. et al. Advanced hypertext implementation. In: **Designing Data-Intensive Web Applications.** San Francisco: Morgan Kaufmann, 2003, (The Morgan Kaufmann Series in Data Management Systems). p. 457–497.

CERQUEIRA, D.; BUENO, S. **Atlas da Violência 2025.** Brasília: [s.n.], 2025. 174 p. Coordenação de Daniel Cerqueira e Samira Bueno. Disponível em: <https://repositorio.ipea.gov.br/handle/11058/17165>. Acesso em: 28 abr. 2025.

CERTOMÀ, C.; CORRINI, L.; RIZZI, F. Crowdsourcing urban sustainability. In: CERTOMÀ, C.; CORRINI, L.; RIZZI, F. (Ed.). **Data, people and technologies in participatory governance.** London: Routledge, 2020. p. 93–106. Disponível em: <http://doi.org/10.1016/j.futures.2014.11.006>. Acesso em: 02 jun. 2025.

CHAVES, S. A. D.; BENITTI, F. B. V. **User-centred privacy and data protection: An overview of current research trends and challenges for the human–computer interaction field.** ACM Computing Surveys, v. 57, p. 1–36. 2025. Disponível em: <https://doi.org/10.1145/3715903>. Acesso em: 01 jun. 2026.

CRAMPTON, J. W. **Mapping: A Critical Introduction to Cartography and GIS.** Chichester: Wiley-Blackwell. 2010. Disponível em: <https://doi.org/10.1002/9781444317411>. Acesso em: 08 dez. 2025.

FURTADO, V. et al. **Collective intelligence in law enforcement - the wikicrimes system.** Information Sciences, v. 180, n. 1, p. 4–17. 2010, Disponível em: <https://doi.org/10.1016/J.INS.2009.08.004>. Acesso em: 26 mai. 2025.

GIRARDI, E. P. **CARTOGRAFIA GEOGRÁFICA CRÍTICA E O DESENVOLVIMENTO DO 'ATLAS DA QUESTÃO AGRÁRIA BRASILEIRA'.** Revista Do Departamento De Geografia, n. spe, p. 302–331. 2014,

Disponível em: <<https://doi.org/10.11606/rdg.v0i0.540>>. Acesso em: 11 abr. 2025.

GOODCHILD, M. F. **Citizens as voluntary sensors: spatial data acquisition in the volunteered geographic information era**. International Journal of Geographical Information Science, v. 21, n. 2, p. 211–221. 2007, Disponível em: <<http://doi.org/10.1007/s10708-007-9111-y>>. Acesso em: 19 mai. 2025.

Google Maps Platform. **Maps JavaScript API Overview**. 2026. Disponível em: <<https://developers.google.com/maps/documentation/javascript/overview>>. Acesso em: 06 jun. 2026.

Google Maps Platform. **Places API Overview**. 2026. Disponível em: <<https://developers.google.com/maps/documentation/places/web-service/overview>>. Acesso em: 06 jun. 2026.

HARLEY, J. B. Deconstructing the map. In: **Classics in Cartography**. [S.l.]: John Wiley & Sons, 2011. cap. 16, p. 271–294. Disponível em: <<https://doi.org/10.1002/9780470669488.ch16>>. Acesso em: 11 mar. 2025.

HAYASHI, S.; MINAMI, F.; SAEKI, M. **Inference-based detection of architectural violations in mvc2**. In: **Proceedings of the 12th International Conference on Software Technologies**. Madrid, Spain: SciTePress, 2017. 2017. (ICSOFTE 2017), p. 394–401. Disponível em: <<https://doi.org/10.5220/0006465103940401>>. Acesso em: 24 mai. 2026.

HENRY, S. L.; ABOU-ZAHRA, S.; BREWER, J. **The role of accessibility in a universal web**. In: **Proceedings of the 11th Web for All Conference**. Seoul, Republic of Korea: Association for Computing Machinery, 2014. 2014. (W4A '14), p. 1–4. Disponível em: <<https://doi.org/10.1145/2596695.2596719>>. Acesso em: 25 mai. 2026.

HOWE, J. **The rise of crowdsourcing**. Wired Magazine, v. 14, n. 6, p. 1–5. 2006, Disponível em: <https://www.researchgate.net/publication/200504371_The_Rise_of_Crowdsourcing>. Acesso em: 22 abr. 2025.

Inertia.js. **Introduction**. 2026. Disponível em: <<https://inertiajs.com/docs/v3/getting-started/index>>. Acesso em: 01 jul. 2026.

JONES, C. **Software Engineering Best Practices: Lessons from Successful Projects in the Top Companies**. [S.l.]: McGraw-Hill. 2009.

KIRBY, E. et al. **Queering the map: Stories of love, loss and (be)longing within a digital cartographic archive**. Media, Culture & Society, v. 43, n. 6, p. 1043–1060. 2021. Disponível em: <<https://doi.org/10.1177/0163443720986005>>. Acesso em: 13 mar. 2026.

LAMONT, T. **Safe spaces or brave spaces? re-envisioning practical theology and transformative learning theory.** Religious Education, v. 115, n. 2, p. 171–183. 2020. Disponível em: <<https://doi.org/10.1080/00344087.2019.1682452>>. Acesso em: 30 mai. 2025.

Laravel. **Database: Migrations.** 2026. Disponível em: <<https://laravel.com/docs/13.x/migrations>>. Acesso em: 05 jun. 2026.

Laravel. **Eloquent: Getting Started.** 2026. Disponível em: <<https://laravel.com/docs/13.x/eloquent>>. Acesso em: 05 jun. 2026.

Laravel. **Laravel 13.x Documentation.** 2026. Disponível em: <<https://laravel.com/docs/13.x>>. Acesso em: 01 jul. 2026.

LAROCHELLE, L. Queering the map. In: HOFER, T. (Ed.). **The new politics of identity: queer theory and the online world.** London: Routledge, 2020. Disponível em: <<https://www.taylorfrancis.com/chapters/edit/10.4324/9781003002338-9/queering-map-lucas-larochelle>>. Acesso em: 09 mar. 2025.

LAUFF, C.; KOTYS-SCHWARTZ, D.; RENTSCHLER, M. E. **What is a prototype? what are the roles of prototypes in companies?** Journal of Mechanical Design, v. 140, n. 6, p. 061102. 2018. Disponível em: <<https://doi.org/10.1115/1.4039340>> Acesso em: 03 mar. 2026.

PHPStan. **Getting Started.** 2026. Disponível em: <<https://phpstan.org/user-guide/getting-started>>. Acesso em: 05 jun. 2026.

Queering the Map. **Queering the Map.** 2026. Disponível em: <<https://www.queeringthemap.com/>>. Acesso em: 05 jun. 2025.

React. **Describing the UI.** 2026. Disponível em: <<https://react.dev/learn/describing-the-ui>>. Acesso em: 01 jul. 2026.

RICARDO, C. D. M.; SIQUEIRA, P. Padilha de; MARQUES, C. R. **Estudo conceitual sobre os espaços urbanos seguros.** Revista Brasileira de Segurança Pública, v. 7, n. 1, p. 200–216. 2013. Disponível em: <<https://doi.org/10.31060/rbsp.2013.v7.n1.212>>. Acesso em: 29 mar. 2025.

SALIM, F.; HAQUE, U. **Urban computing in the wild: A survey on large scale participation and citizen engagement with ubiquitous computing, cyber physical systems, and internet of things.** International Journal of Human-Computer Studies, v. 81, p. 31–48. 2015. Disponível em: <<https://doi.org/10.1016/j.ijhcs.2015.03.003>>. Acesso em: 09 jun. 2025.

SILVA, D. C. P. **Materialização discursiva da cis-heteronormatividade em perspectiva escalar: Contribuições para a linguística queer.** Cadernos de Linguagem e Sociedade, v. 21, n. 2, p. 280–306. 2020.

Disponível em: <<https://doi.org/10.26512/les.v21i2.35269>>. Acesso em: 04 mai. 2025.

SQLite. **About SQLite**. 2025. Disponível em: <<https://sqlite.org/about.html>>. Acesso em: 30 jun. 2026.

STACH, C. et al. **Protecting sensitive data in the information age: State of the art and future prospects**. Future Internet, v. 14, n. 11, p. 302. 2022. Disponível em: <<https://doi.org/10.3390/fi14110302>>. Acesso em: 01 jun. 2026.

TU, Z. **Research on the application of layered architecture in computer software development**. Journal of Computing and Electronic Information Management, v. 11, n. 3, p. 34–38. 2023. Disponível em: <<https://doi.org/10.54097/jceim.v11i3.08>>. Acesso em: 01 jun. 2026.