



FLUTTER PARA CIDADES INTELIGENTES: DESENVOLVIMENTO DE APLICAÇÕES ESCALÁVEIS NO CONTEXTO DA SAÚDE

FLUTTER FOR SMART CITIES: DEVELOPMENT OF SCALABLE APPLICATIONS IN THE HEALTH CONTEXT

FLUTTER PARA CIUDADES INTELIGENTES: DESARROLLO DE APLICACIONES ESCALABLES EN EL CONTEXTO DE LA SALUD

Andrey Alencar Quadros¹
Charleston Ribeiro dos Passos²

DOI: 10.54751/revistafoco.v19n6-079

Received: May 22nd, 2026

Accepted: Jun 15th, 2026



RESUMO

Este artigo apresenta uma proposta de arquitetura e desenvolvimento de aplicativo móvel para o contexto de cidades inteligentes, com foco em saúde pública municipal, utilizando Flutter e princípios SOLID. O problema investigado está relacionado à fragmentação de fluxos de agendamento, baixa rastreabilidade de demandas assistenciais e dificuldade de escalabilidade de soluções digitais quando os municípios ampliam a cobertura de serviços. A pesquisa foi conduzida por meio de abordagem aplicada, com revisão narrativa da literatura, análise documental e estudo de caso técnico vinculado ao projeto Cidades Inteligentes em Ariquemes (RO). Como resultado, propõe-se um modelo de implementação orientado a camadas, contratos e tratamento centralizado de exceções, capaz de reduzir acoplamento, facilitar manutenção evolutiva e acelerar entregas incrementais. Também são discutidos requisitos de interoperabilidade, segurança da informação e governança de dados em conformidade com a Lei Geral de Proteção de Dados. Conclui-se que o Flutter, quando combinado com decisões arquiteturais adequadas e práticas de engenharia de software, constitui alternativa viável para soluções móveis de saúde escaláveis, sustentáveis e alinhadas à transformação digital do setor público.

Palavras-chave: Flutter; cidades inteligentes; saúde digital; escalabilidade; princípios SOLID.

¹ Mestre em Propriedade Intelectual e Transferência de Tecnologia em Inovação. Universidade Federal de Mato Grosso (UFMT). Instituto Federal de Rondônia (IFRO). Rodovia RO-257, Km 13, sentido Machadinho do Oeste, Zona Rural de Ariquemes, Rondônia. E-mail: andrey.quadros@ifro.edu.br

² Especialização em Arquitetura de Software. Instituto Federal de Rondônia (IFRO), Facuminas. Rodovia RO-257, Km 13, sentido Machadinho do Oeste, Zona Rural de Ariquemes, Rondônia. E-mail: charleston.passos@gmail.com

ABSTRACT

This article presents an architectural and development proposal for a mobile health application in smart city environments using Flutter and SOLID principles. The research problem addresses fragmented scheduling flows, poor traceability of healthcare demand, and limited scalability of municipal digital services. The study follows an applied approach combining narrative literature review, document analysis, and a technical case study linked to the Smart Cities initiative in Ariquemes, Rondônia, Brazil. The main contribution is a layered implementation model based on contracts and centralized exception handling, designed to decrease coupling, improve maintainability, and support incremental releases. The paper also discusses interoperability, data governance, and information security requirements according to Brazil's General Data Protection Law. Findings indicate that Flutter, combined with consistent software engineering decisions, is a feasible strategy for scalable, sustainable, and citizen-centered mobile health services in public-sector digital transformation programs.

Keywords: Flutter; smart cities; digital health; scalability; SOLID principles.

RESUMEN

Este artículo presenta una propuesta de arquitectura y desarrollo de una aplicación móvil para el contexto de ciudades inteligentes, con énfasis en la salud pública municipal, utilizando Flutter y principios SOLID. El problema investigado se relaciona con la fragmentación de los flujos de programación de consultas, la baja trazabilidad de las demandas asistenciales y la dificultad de escalabilidad de las soluciones digitales cuando los municipios amplían la cobertura de servicios. La investigación se desarrolló mediante un enfoque aplicado, con revisión narrativa de la literatura, análisis documental y estudio de caso técnico vinculado al proyecto Ciudades Inteligentes en Ariquemes (RO). Como resultado, se propone un modelo de implementación orientado a capas, contratos y tratamiento centralizado de excepciones, capaz de reducir el acoplamiento, facilitar el mantenimiento evolutivo y acelerar entregas incrementales. También se discuten requisitos de interoperabilidad, seguridad de la información y gobernanza de datos conforme a la Ley General de Protección de Datos de Brasil. Se concluye que Flutter, cuando se combina con decisiones arquitectónicas adecuadas y prácticas de ingeniería de software, constituye una alternativa viable para soluciones móviles de salud escalables, sostenibles y alineadas con la transformación digital del sector público.

Palabras clave: Flutter; ciudades inteligentes; salud digital; escalabilidad; principios SOLID.

1. Introdução

A digitalização dos serviços públicos de saúde vem se consolidando como agenda estratégica em municípios que buscam maior eficiência administrativa, ampliação de acesso e qualificação da experiência do cidadão. No cenário de cidades inteligentes, essa agenda é potencializada pela integração entre infraestrutura digital, dados em tempo quase real e serviços orientados por

evidências. Ainda assim, muitas iniciativas municipais permanecem restritas a sistemas isolados, com baixa interoperabilidade e pouco reaproveitamento tecnológico entre secretarias, unidades de saúde e canais de atendimento à população.

No caso específico do agendamento de consultas, a fragmentação de fluxos gera efeitos diretos sobre o desempenho do sistema local de saúde: filas invisíveis, duplicidade de demandas, absenteísmo sem monitoramento e dificuldade de priorização clínica. Esses problemas não se resolvem apenas com informatização parcial. Eles exigem desenho de produto digital, arquitetura de software, governança de dados e rotinas operacionais coerentes entre si. Portanto, discutir o desenvolvimento de aplicativos móveis nesse contexto implica compreender tanto os aspectos técnicos da implementação quanto os condicionantes institucionais da gestão pública.

O Flutter surge como alternativa relevante para esse tipo de desafio por combinar desenvolvimento multiplataforma, alto desempenho de interface e ecossistema maduro de bibliotecas. Ao permitir uma base única de código para Android e iOS, o framework tende a reduzir custo de manutenção e acelerar ciclos de entrega, aspectos fundamentais em projetos públicos sujeitos a restrição orçamentária e necessidade de continuidade administrativa. Contudo, adotar Flutter sem disciplina arquitetural pode apenas deslocar os problemas de legados tradicionais para um novo ambiente tecnológico.

Diante desse cenário, este trabalho parte da premissa de que a escalabilidade de soluções móveis em saúde não depende exclusivamente da ferramenta de desenvolvimento, mas da combinação entre framework adequado, princípios de projeto e práticas consistentes de engenharia. Nesse sentido, os princípios SOLID são analisados como mecanismos de governança interna do código, com impacto direto na capacidade de evolução do sistema ao longo do tempo. A discussão envolve tratamento de erros, abstração de rotas, desacoplamento de dependências e definição de contratos entre camadas.

O estudo está vinculado ao projeto Cidades Inteligentes em Ariquemes, no estado de Rondônia, iniciativa que busca estruturar soluções tecnológicas para atendimento de demandas urbanas prioritárias, incluindo saúde. A proposta

de aplicativo de agendamento, tomada como estudo de caso, foi utilizada para examinar decisões arquiteturais, desafios de implementação e resultados técnicos preliminares. Embora o foco empírico seja local, as lições extraídas apresentam potencial de replicação para outros contextos municipais com desafios semelhantes de transformação digital.

A questão norteadora do artigo pode ser formulada da seguinte forma: como estruturar, com Flutter, uma aplicação móvel de saúde capaz de escalar funcionalmente e manter qualidade de manutenção em cenário de gestão pública municipal? A partir dessa questão, o objetivo geral consiste em propor e analisar um modelo de desenvolvimento escalável para aplicativo de agendamento no contexto de cidades inteligentes. Como objetivos específicos, busca-se: a) sistematizar fundamentos teóricos sobre saúde digital e cidades inteligentes; b) discutir o papel de Flutter e dos princípios SOLID na engenharia da solução; c) apresentar a arquitetura adotada no estudo de caso; d) analisar ganhos e limitações observados durante a implementação.

Do ponto de vista científico e aplicado, o trabalho contribui em três frentes. Primeiro, aproxima literatura de smart cities e práticas concretas de desenvolvimento móvel, tema ainda tratado de forma fragmentada. Segundo, oferece uma organização metodológica de implementação que pode ser reutilizada por equipes técnicas municipais e acadêmicas. Terceiro, explicita requisitos de conformidade com normas e regulações brasileiras, especialmente a Lei Geral de Proteção de Dados, frequentemente subestimada na fase inicial de projetos digitais em saúde.

2. Referencial Teórico

2.1 Cidades Inteligentes e Saúde Digital

A literatura sobre cidades inteligentes indica que o conceito ultrapassa a simples adoção de sensores ou plataformas digitais. Trata-se de modelo de governança urbana no qual infraestrutura tecnológica, planejamento institucional e participação social convergem para melhorar qualidade de vida. Em saúde

pública, essa convergência se traduz em capacidade de monitorar indicadores territoriais, antecipar demandas, qualificar decisões clínicas e reduzir assimetrias de acesso entre populações com diferentes condições socioeconômicas (Rocha *et al.*, 2019).

No campo internacional, estudos apontam que cidades inteligentes podem elevar padrões de prevenção, resposta a emergências e coordenação entre níveis assistenciais, especialmente quando integradas a políticas de saúde digital e segurança da informação (Ahmed *et al.*, 2017). A contribuição central dessas iniciativas está na articulação entre dados e serviços: sistemas deixam de operar apenas como repositórios administrativos e passam a apoiar decisões clínicas e gerenciais em ciclos contínuos de aprendizagem.

No contexto brasileiro, a transformação digital em saúde ganhou maior institucionalidade com estratégias nacionais que reconhecem interoperabilidade, governança e cidadania digital como pilares da modernização do SUS. A Estratégia de Saúde Digital para o Brasil 2020-2028 reforça a necessidade de soluções centradas no usuário, capazes de integrar informações entre pontos de cuidado e de reduzir redundâncias de coleta de dados. Esse direcionamento é particularmente relevante para municípios de médio porte, onde equipes enxutas precisam equilibrar inovação e sustentabilidade operacional.

Entretanto, a implementação local enfrenta obstáculos recorrentes: legado de sistemas heterogêneos, dependência de fornecedores com baixo nível de documentação técnica, ausência de padronização de APIs e insuficiência de mecanismos de observabilidade. Esses fatores comprometem escalabilidade e elevam o custo de manutenção após a fase inicial do projeto. Nesse cenário, aplicativos móveis podem atuar como camada de experiência do cidadão, desde que acoplados a serviços de dados robustos e a contratos de integração estáveis.

Outro ponto crítico é a proteção de dados pessoais sensíveis. Informações de saúde exigem tratamento diferenciado em todas as etapas do ciclo de vida dos dados: coleta, armazenamento, compartilhamento, retenção e descarte. A LGPD estabelece bases legais, princípios e direitos dos titulares que precisam ser incorporados ao desenho da solução desde o início, sob lógica de

privacy by design. Assim, decisões de arquitetura móvel não podem ignorar criptografia, autenticação forte, trilhas de auditoria e segregação de responsabilidades técnicas.

Do ponto de vista de valor público, a digitalização do agendamento não deve ser interpretada apenas como automação de fila. Ela representa oportunidade de reorganizar o acesso à rede assistencial com base em transparência, previsibilidade e rastreabilidade. Quando acompanhada de indicadores operacionais, a solução permite identificar gargalos por especialidade, unidade e território, apoiando planejamento de oferta. Portanto, discutir aplicativo móvel em saúde implica discutir também gestão e inteligência do cuidado.

2.2 Flutter e Desenvolvimento Multiplataforma

Flutter é um framework open source mantido pela Google, baseado na linguagem Dart, orientado à construção de interfaces multiplataforma com alto nível de consistência visual e desempenho (Flutter, 2026). Diferentemente de abordagens híbridas centradas em webview, o Flutter renderiza componentes por meio de engine própria, o que tende a reduzir discrepâncias de comportamento entre sistemas operacionais e ampliar controle sobre experiência de uso. Para serviços públicos, essa previsibilidade representa ganho importante em acessibilidade e suporte.

A estratégia de código único para Android e iOS oferece vantagem econômica em cenários de orçamento limitado: menos duplicação de esforço, testes mais padronizados e curva de manutenção mais estável ao longo do ciclo de vida da solução. Em equipes pequenas, essa característica facilita planejamento de backlog e libera capacidade para tarefas de maior valor, como observabilidade, segurança e integração com serviços legados. Entretanto, para que os ganhos se concretizem, a base de código precisa ser organizada de forma modular e orientada a responsabilidades claras.

No domínio de saúde digital, o Flutter também se destaca pela disponibilidade de bibliotecas para autenticação, criptografia, armazenamento

local seguro, notificações e integração HTTP. Esses recursos aceleram prototipação e experimentação, porém não substituem decisões arquiteturais. O uso indiscriminado de pacotes, sem critérios de versionamento e sem encapsulamento, tende a introduzir dependências frágeis. Por isso, práticas de engenharia recomendam que bibliotecas externas sejam acessadas por camadas de abstração, minimizando impacto de futuras trocas.

Em aplicações voltadas a agendamento, a experiência de usuário exige respostas rápidas, comunicação clara de estado e navegação intuitiva. Flutter permite construir fluxos com feedback imediato, componentes reutilizáveis e personalização de comportamento conforme perfil do usuário. Contudo, a qualidade percebida não depende apenas de interface; depende também de consistência de regras de negócio, manejo de erros e disponibilidade de dados. Assim, a arquitetura precisa equilibrar desempenho de front-end e robustez de integração com backend.

Outro aspecto relevante é a evolução recente do ecossistema, que ampliou suporte para web e desktop. Embora o escopo deste artigo esteja concentrado em dispositivos móveis, a possibilidade de reaproveitamento de componentes e camadas de domínio em múltiplos canais representa vantagem estratégica para projetos municipais, que frequentemente precisam ofertar serviços também em totens, portais e painéis administrativos. Nessa perspectiva, Flutter pode funcionar como núcleo de uma estratégia omnicanal de serviços digitais em saúde.

Por fim, a adoção institucional de Flutter exige governança técnica: padronização de arquitetura, definição de convenções de código, gestão de dependências, esteira de integração contínua e critérios de qualidade. Sem esses elementos, o framework pode entregar produtividade inicial, mas perder sustentabilidade em médio prazo. Portanto, a discussão sobre Flutter em cidades inteligentes deve considerar não apenas sua capacidade tecnológica, mas também sua aderência a práticas organizacionais de engenharia de software.

2.3 Princípios Solid e Escalabilidade

Os princípios SOLID foram formulados para aumentar coesão, reduzir acoplamento e ampliar adaptabilidade de sistemas orientados a objetos. Em ambientes de saúde digital, onde regras de negócio mudam com frequência e integrações são contínuas, esses princípios oferecem base prática para evoluir software sem comprometer estabilidade. Mais do que um conjunto de recomendações abstratas, eles funcionam como critérios de decisão arquitetural para separar responsabilidades e reduzir efeitos colaterais de alterações incrementais (Martin, 2019; Martin, 2023).

No princípio da responsabilidade única, cada módulo deve possuir motivo claro para mudança. Em um aplicativo de agendamento, isso significa separar lógica de interface, validação de regras clínicas, persistência local e comunicação com APIs. Quando essas funções se misturam em um mesmo componente, qualquer ajuste pontual pode gerar regressões amplas e elevar custo de testes. A separação explícita por responsabilidade facilita manutenção, acelera onboarding de novos desenvolvedores e melhora observabilidade de falhas.

O princípio aberto/fechado é particularmente útil para acomodar novas políticas públicas sem reescrever estruturas estáveis. Por exemplo, se o município precisar introduzir regra de prioridade para determinados grupos, a camada de aplicação pode estender comportamento por novos casos de uso, preservando contratos já consolidados. Essa abordagem reduz risco operacional durante mudanças normativas, algo comum em serviços públicos sujeitos a atualização regulatória e reorganização de fluxos assistenciais.

A substituição de Liskov e a segregação de interfaces complementam esse desenho ao garantir que implementações concretas possam ser trocadas sem quebrar expectativas de consumo. Em termos práticos, um repositório remoto pode ser substituído por mock de teste ou por fonte local temporária em cenários offline, desde que mantenha contrato comportamental. Já a segregação de interfaces evita dependências excessivas: cada módulo acessa apenas as

operações de que realmente necessita, simplificando testes e reduzindo superfície de erro.

A inversão de dependência fecha o ciclo ao estabelecer que módulos de alto nível dependam de abstrações, não de implementações concretas. Em Flutter, isso permite encapsular bibliotecas de navegação, cliente HTTP, armazenamento e analytics por adaptadores internos. Com essa estratégia, a equipe preserva autonomia sobre o código da aplicação e reduz risco de lock-in tecnológico. Em projetos de longa duração, essa independência é decisiva para manter sustentabilidade técnica e financeira.

"Os módulos de software devem ser abertos para extensão, mas fechados para modificação, de forma que novas capacidades possam ser adicionadas sem comprometer comportamentos já estáveis" (Martin, 2023, p. 122).

No conjunto, SOLID contribui para transformar produtividade pontual em capacidade estrutural de evolução. A adoção disciplinada desses princípios não elimina complexidade, mas torna essa complexidade explícita, governável e testável. Para cidades inteligentes, onde soluções tendem a crescer em escopo e criticidade, esse atributo é central: escalar não é apenas atender mais usuários, mas preservar confiabilidade, segurança e previsibilidade durante o crescimento.

2.4 Interoperabilidade, Segurança e Governança de Dados

A escalabilidade de aplicativos de saúde depende fortemente da capacidade de interoperar com sistemas clínicos e administrativos já existentes. Sem integração, o aplicativo se torna ilha funcional, duplicando cadastros e ampliando inconsistências. Por isso, a arquitetura proposta considera integração por APIs padronizadas, versionamento explícito de contratos e registro de eventos críticos para auditoria. A interoperabilidade deve ser tratada como requisito não funcional prioritário, e não como etapa posterior de projeto.

No plano da segurança, os controles mínimos incluem autenticação robusta, gestão de sessão, criptografia em trânsito, criptografia de dados sensíveis em repouso e monitoramento de tentativas de acesso indevido. Em

ambiente móvel, também é recomendável minimizar persistência local de dados clínicos, adotar tokens com expiração adequada e implementar mecanismos de revogação. Essas medidas reduzem impacto de perda de dispositivo e de comprometimento de credenciais.

A LGPD estabelece princípios como finalidade, adequação, necessidade, transparência e segurança que precisam orientar o desenho da aplicação desde a fase de requisitos (Brasil, 2018). No caso do agendamento, dados coletados devem ser estritamente necessários para o serviço, com informações claras sobre tratamento e retenção. A governança deve prever papéis institucionais, procedimentos de resposta a incidentes e canais de atendimento ao titular de dados, em alinhamento com a administração pública.

Além de proteção legal, a governança de dados tem função estratégica: transformar registros operacionais em inteligência para gestão da rede. Indicadores como tempo médio até consulta, taxa de comparecimento e distribuição territorial da demanda podem orientar alocação de equipes e horários. Contudo, a geração de valor analítico requer qualidade de dados, padronização semântica e processos de validação contínua. Sem isso, dashboards sofisticados apenas reproduzem ruído informacional.

Portanto, arquitetura móvel, segurança e governança de dados devem ser tratadas de modo integrado. A solução técnica não pode dissociar experiência do cidadão de responsabilidades institucionais sobre o dado em saúde. Esse entendimento reforça o caráter socio-técnico da transformação digital em cidades inteligentes: tecnologia viabiliza, mas política pública e gestão sustentam a efetividade do serviço.

3. Metodologia

A pesquisa adota abordagem aplicada, de natureza qualitativa, com finalidade descritivo-analítica. A estratégia metodológica combinou revisão narrativa da literatura, análise documental de normas e diretrizes e estudo de caso técnico no contexto do projeto Cidades Inteligentes em Ariquemes (RO). A escolha por revisão narrativa foi adequada ao objetivo de integrar conceitos de

idades inteligentes, saúde digital e engenharia de software sem restringir a investigação a um único protocolo de busca bibliográfica.

Na etapa de revisão teórica, foram considerados artigos, documentos institucionais e referências clássicas de arquitetura de software relevantes para o tema. O recorte priorizou publicações com contribuição direta para: a) uso de tecnologias digitais na saúde pública; b) fundamentos de smart cities aplicados a serviços urbanos; c) boas práticas de desenvolvimento de software escalável; d) requisitos de proteção de dados no contexto brasileiro. As fontes foram examinadas para construir base conceitual e critérios de desenho arquitetural.

A análise documental concentrou-se em normas e diretrizes que impactam o desenvolvimento da solução, incluindo a Lei Geral de Proteção de Dados e orientações de saúde digital em âmbito nacional e internacional. Esses documentos foram utilizados para derivar requisitos não funcionais de segurança, governança e transparência. O procedimento permitiu alinhar decisões técnicas a exigências regulatórias, aspecto central para projetos públicos que operam com dados pessoais sensíveis.

O estudo de caso técnico foi conduzido sobre um protótipo funcional de aplicativo de agendamento de consultas, estruturado em Flutter. A unidade de análise foi a arquitetura da aplicação e seus mecanismos de escalabilidade técnica, não o impacto clínico direto sobre desfechos em saúde. Foram observados componentes de navegação, tratamento de erros, organização de camadas, contratos de integração e padrões de reutilização de código. Essa delimitação permitiu foco na contribuição de engenharia proposta pelo artigo.

Para avaliação qualitativa da implementação, adotaram-se quatro critérios analíticos: manutenibilidade, extensibilidade, consistência de experiência do usuário e governança de dados. A manutenibilidade foi examinada pela clareza de responsabilidades e facilidade de localizar pontos de mudança. A extensibilidade foi avaliada pela capacidade de incluir novos fluxos sem refatorações disruptivas. A consistência de experiência considerou previsibilidade de navegação e mensagens de erro. A governança observou aderência a controles de proteção e rastreabilidade de operações.

A coleta de evidências técnicas incluiu registros de implementação, inspeção de código, análise de estrutura de pastas, revisão de contratos entre camadas e observação de cenários de exceção simulados. Embora não se trate de experimento controlado, o método permitiu triangulação entre teoria, normativa e prática de desenvolvimento, fortalecendo a validade analítica do estudo. Limitações do procedimento são reconhecidas na seção de discussão, especialmente quanto à ausência de avaliação longitudinal em produção.

Do ponto de vista ético, o estudo não utilizou dados clínicos identificáveis de pacientes. O foco permaneceu em artefatos de software e fluxos técnicos, com descrição de boas práticas de anonimização e minimização de dados para cenários reais de implantação. Essa decisão metodológica preserva conformidade com requisitos legais e evita extrapolações indevidas sobre eficácia assistencial, tema que demandaria desenho de pesquisa específico em saúde coletiva.

Como complemento metodológico, foi elaborado um protocolo de rastreabilidade entre requisito, decisão arquitetural e evidência observada durante a implementação. Esse protocolo permitiu documentar, para cada componente relevante, a justificativa técnica de sua adoção, os riscos associados e os resultados percebidos em testes de fluxo. A rastreabilidade fortalece transparência analítica do estudo, facilita revisão por pares e cria base para replicação em outros cenários municipais. Além disso, reduz dependência de memória tácita da equipe, um ponto sensível em iniciativas públicas com alta rotatividade de profissionais.

Quadro 1 - Camadas da arquitetura proposta para o aplicativo de agendamento

Camada	Responsabilidade	Principais artefatos
Apresentação	Interface com o usuário, fluxos de navegação e acessibilidade	Páginas Flutter, widgets reutilizáveis, temas e rotas
Aplicação	Orquestração de casos de uso e validação de regras	UseCases, controladores, contratos de serviço
Domínio	Entidades centrais e regras de negócio independentes de framework	Entidades, objetos de valor, interfaces
Dados	Integração com APIs, cache local e tratamento de falhas	Repositórios, DTOs, mapeadores e cliente HTTP

Fonte: Elaborado para este estudo, com base em Martin (2019) e na documentação Flutter (2026).

4. Estudo de Caso: Aplicativo de Agendamento de Consultas

O estudo de caso parte de uma necessidade operacional recorrente em redes municipais: reduzir fricção no processo de marcação de consultas e aumentar previsibilidade do atendimento para usuários e equipes de saúde. No cenário observado, parte dos agendamentos ocorria de forma presencial ou por canais não integrados, dificultando controle de oferta, remarcações e confirmação de presença. O aplicativo foi concebido como camada de acesso cidadão para unificar essas interações em fluxo digital rastreável.

A solução foi estruturada em quatro camadas: apresentação, aplicação, domínio e dados. Na camada de apresentação, componentes Flutter foram organizados em widgets reutilizáveis e páginas orientadas a estados de interface. A camada de aplicação concentrou casos de uso, validações de fluxo e orquestração de eventos. A camada de domínio reuniu entidades e contratos independentes de framework. Por fim, a camada de dados encapsulou integração com API, serialização e persistência local de informações não sensíveis.

No fluxo principal, o usuário autentica-se, seleciona especialidade, consulta agenda disponível, confirma horário e recebe retorno com protocolo digital. O desenho do fluxo priorizou clareza cognitiva e redução de passos, com mensagens explícitas de sucesso, espera e erro. A navegação foi encapsulada por abstração de rotas para evitar dispersão de chamadas diretas ao mecanismo nativo. Esse encapsulamento permitiu centralizar regras de transição e manter previsibilidade da jornada em diferentes cenários de uso.

O tratamento de erros seguiu padrão orientado a tipos de exceção conhecidos, com interface comum para comunicação de falhas à camada de apresentação. Essa estratégia evitou mensagens técnicas ao usuário final e facilitou roteamento de erros para observabilidade. Exceções transitórias de rede, por exemplo, acionam orientações de tentativa posterior; inconsistências de entrada acionam mensagens de validação imediata; falhas não mapeadas são registradas para análise de desenvolvimento. Dessa forma, o sistema diferencia responsabilidade de experiência e responsabilidade de diagnóstico.

A aplicação dos princípios SOLID mostrou-se particularmente útil em três pontos. Primeiro, no desacoplamento entre widgets e regras de negócio, reduzindo riscos de regressão visual durante mudanças de lógica. Segundo, na troca de mecanismos de navegação sem impacto sobre casos de uso. Terceiro, na substituição de repositórios por implementações de teste, agilizando validação de cenários sem dependência de backend instável. Esses ganhos reforçam a hipótese de que arquitetura orientada a abstrações é fator crítico de escalabilidade.

No aspecto de segurança, o protótipo foi desenhado para trafegar dados via conexões seguras, com tokens de sessão de curta duração e armazenamento local restrito a informações necessárias para experiência de continuidade. Dados sensíveis de saúde não foram persistidos em texto aberto no dispositivo. Além disso, eventos críticos de autenticação e confirmação de agendamento foram mapeados para trilha de auditoria, com potencial de integração a mecanismos institucionais de monitoramento.

Em relação à interoperabilidade, a integração foi tratada por contratos de API versionados, com mapeadores explícitos entre modelos internos e payloads externos. Esse desenho reduz impacto de mudanças no backend e facilita evolução incremental de endpoints. Também favorece manutenção por equipes distintas, pois limita dependência de conhecimento implícito. No médio prazo, a adoção de padrões interoperáveis mais amplos pode ampliar integração com prontuários eletrônicos e sistemas de regulação, desde que acompanhada de governança semântica dos dados.

A implementação evidenciou desafios típicos de projetos públicos: variação de maturidade digital entre unidades, necessidade de compatibilidade com dispositivos diversos e dependência de disponibilidade de serviços externos. Esses desafios foram mitigados com fallback de interface, tratamento robusto de indisponibilidade e monitoramento de tempo de resposta por fluxo. Ainda assim, a consolidação da solução em produção demandará evolução contínua de infraestrutura e capacitação de equipes gestoras.

Do ponto de vista de governança do produto, a equipe adotou planejamento incremental por ciclos curtos, com priorização de funcionalidades

críticas para acesso inicial. Essa abordagem permitiu reduzir risco de escopo excessivo e incorporar feedback de uso nas iterações seguintes. Em contexto de cidade inteligente, essa lógica incremental é especialmente útil porque articula entregas rápidas com aprendizagem institucional progressiva, evitando projetos monolíticos de difícil sustentação.

Para apoiar a sustentação do produto, foi proposta uma rotina de gestão de configuração com versionamento semântico, documentação de alterações e critérios formais de aceite por funcionalidade. Essa rotina reduz ambiguidade entre áreas técnicas e gestoras, melhora previsibilidade de implantação e cria histórico auditável de decisões. Em projetos públicos, onde mudanças de equipe e transições administrativas são frequentes, esse histórico funciona como mecanismo de continuidade institucional.

A camada de observabilidade também recebeu atenção específica no desenho do caso. Foram definidos eventos essenciais para acompanhamento operacional, como início e término de sessão, tentativas de agendamento, confirmações concluídas e falhas de integração. A coleta desses eventos permite construir painéis de monitoramento de uso e desempenho, além de apoiar investigação de incidentes. Sem observabilidade, a gestão da solução tende a se basear em percepção subjetiva, dificultando priorização de melhorias.

No processo de testes, a estratégia combinou validação de componentes isolados, testes de fluxo completo e inspeção manual de acessibilidade básica. A existência de contratos de interface favoreceu criação de cenários de teste previsíveis e reutilizáveis. Embora a maturidade de automação ainda possa evoluir, a estrutura atual já permite reduzir regressões em entregas frequentes. Para expansão em escala municipal, recomenda-se ampliar cobertura de testes de integração e consolidar execução contínua em pipeline automatizado.

Em termos de capacidade de expansão territorial, o modelo arquitetural adotado permite parametrizar unidades de atendimento, especialidades e regras de agenda sem necessidade de replicar a aplicação para cada contexto local. Essa parametrização aumenta reutilização da base de código e facilita governança centralizada de políticas de acesso. Assim, a solução deixa de ser

protótipo isolado e passa a configurar plataforma potencial para evolução regional, respeitando especificidades operacionais de cada município.

5. Análise e Discussão dos Resultados

Os resultados observados no estudo de caso indicam que o Flutter atende adequadamente aos requisitos de desenvolvimento multiplataforma para serviços móveis de saúde, desde que combinado com disciplina arquitetural. A produtividade inicial da equipe foi favorecida por hot reload, ecossistema de componentes e padrão único de interface. Entretanto, o ganho mais consistente não veio apenas da ferramenta, mas da forma como o código foi organizado por camadas e contratos. Em outras palavras, o framework potencializou resultados porque a arquitetura reduziu fragilidade estrutural.

Quanto à manutenibilidade, a separação de responsabilidades facilitou identificação de pontos de alteração e diminuiu efeitos colaterais em mudanças pontuais. A substituição de implementações concretas por abstrações permitiu ajustes em dependências externas sem reescrita ampla de módulos de alto nível. Esse comportamento é coerente com a literatura de arquitetura limpa, que defende isolamento do domínio como estratégia para preservar estabilidade evolutiva (Martin, 2019).

No critério de extensibilidade, a aplicação mostrou capacidade de incorporar novos fluxos de agendamento com impacto limitado em componentes existentes. A introdução de novas regras de validação foi realizada predominantemente na camada de aplicação, mantendo páginas e serviços desacoplados. Embora esse resultado seja preliminar, ele sugere viabilidade de expansão funcional para cenários como teleatendimento, confirmação automática e integração com campanhas de saúde preventiva.

Em termos de experiência do usuário, o modelo de tratamento de erros contribuiu para comunicação mais clara em situações de falha. A categorização de exceções evitou respostas genéricas e permitiu orientar o usuário sobre próxima ação possível. Esse ponto é relevante em serviços públicos digitais, nos quais mensagens ambíguas tendem a aumentar abandono de fluxo e sobrecarga

de atendimento presencial. A consistência de navegação também foi beneficiada pela abstração de rotas, que reduziu divergências entre telas e comportamentos.

A discussão sobre governança de dados reforça que escalabilidade técnica sem conformidade regulatória representa risco operacional elevado. A incorporação de requisitos da LGPD desde o desenho da solução mostrou-se adequada para reduzir retrabalho e evitar adaptações tardias. Ainda assim, há necessidade de amadurecimento institucional em temas como classificação de dados, gestão de consentimento em casos aplicáveis, políticas de retenção e resposta a incidentes. Esses elementos extrapolam a esfera do desenvolvimento e demandam articulação com gestão pública.

Comparando-se os achados com estudos de smart health, observa-se convergência no entendimento de que valor público emerge da integração entre tecnologia, dados e governança (Solanas *et al.*, 2014; Rocha *et al.*, 2019). Aplicativos móveis são porta de entrada importante, mas precisam estar conectados a arquitetura de informação mais ampla para produzir impacto sistêmico. Sem integração, a solução melhora experiência individual, porém não altera substancialmente a coordenação da rede de atenção.

Também se destacam limitações. A avaliação concentrou-se em protótipo técnico e não mediu desfechos clínicos ou econômicos em ambiente operacional de longa duração. Além disso, variáveis organizacionais, como cultura de uso de dados e continuidade de equipes, podem influenciar fortemente resultados futuros. Assim, os achados devem ser lidos como evidências de viabilidade técnica e não como prova definitiva de efetividade assistencial. Pesquisas futuras podem avançar para estudos longitudinais com indicadores de acesso, absenteísmo e satisfação.

Mesmo com essas limitações, a experiência analisada oferece contribuição prática imediata: demonstra que municípios podem iniciar transformação digital em saúde com soluções móveis escaláveis sem depender de arquiteturas monolíticas. O fator-chave é alinhar framework, princípios de design e governança institucional desde o início. Quando esse alinhamento ocorre, o aplicativo deixa de ser solução isolada e passa a atuar como componente estruturante de ecossistema digital urbano orientado ao cidadão.

5.1 Indicadores Operacionais e Monitoramento Contínuo

A discussão de resultados torna-se mais robusta quando associada a indicadores de desempenho monitorados continuamente. Para o contexto de agendamento móvel em saúde, recomenda-se acompanhar, no mínimo, taxa de conclusão de agendamento, tempo médio de resposta por transação, taxa de erro por categoria de exceção e índice de comparecimento após confirmação digital. Esses indicadores permitem avaliar simultaneamente eficiência tecnológica e efetividade operacional do serviço.

A taxa de erro por categoria, quando combinada com trilhas de observabilidade, apoia decisões de engenharia de forma objetiva. Em vez de priorizar backlog por percepção, a equipe passa a atuar sobre evidências. Erros de validação recorrentes podem demandar simplificação de fluxo; erros de rede podem requerer estratégias de retry e cache; erros inesperados podem indicar lacunas de tratamento em camadas específicas. Esse ciclo fortalece governança técnica e reduz tempo de resolução de incidentes.

5.2 Riscos de Implantação e Estratégias de Mitigação

A transição de protótipo para produção envolve riscos técnicos e organizacionais que precisam ser mapeados antecipadamente. Entre os riscos técnicos, destacam-se indisponibilidade de serviços legados, divergência de versões de API e crescimento não planejado de dependências externas. Entre os riscos organizacionais, sobressaem rotatividade de equipes, baixa adesão de usuários e ausência de processos formais de suporte. Ignorar esses fatores compromete a sustentabilidade da solução independentemente da qualidade do código.

Como estratégia de mitigação, recomenda-se plano de implantação em fases com critérios explícitos de entrada e saída. Na fase inicial, priorizam-se unidades piloto e fluxos críticos, com acompanhamento intensivo de indicadores. Na fase intermediária, amplia-se cobertura de especialidades e integrações, mantendo janelas controladas de atualização. Na fase de consolidação,

institucionalizam-se processos de suporte, treinamento e governança de versão. Esse modelo reduz risco de interrupções abruptas e melhora aprendizado organizacional.

Também é essencial estabelecer política de continuidade de negócio para cenários de indisponibilidade parcial. O aplicativo deve prever modos degradados de operação, comunicação transparente ao usuário e procedimento de contingência para equipes administrativas. Em paralelo, a infraestrutura de backend precisa de monitoramento proativo, alertas automáticos e plano de recuperação documentado. A combinação entre resiliência técnica e prontidão operacional é condição para manter confiança do cidadão no serviço digital.

Do ponto de vista de pessoas e capacitação, a implantação sustentável exige trilha de formação para perfis distintos: equipe técnica, equipe de atendimento e gestão de saúde. Cada grupo interage com o sistema de forma diferente e demanda materiais específicos de treinamento. A institucionalização desse conhecimento reduz dependência de indivíduos-chave e fortalece autonomia local. Em cidades inteligentes, capital humano qualificado é tão decisivo quanto a plataforma tecnológica adotada.

Por fim, a mitigação de riscos deve incluir governança contratual e documental. Requisitos de segurança, níveis de serviço, responsabilidades de suporte e critérios de evolução precisam estar formalizados para evitar conflitos e lacunas de execução. A experiência demonstra que projetos com documentação técnica mínima e acordos operacionais claros respondem melhor a mudanças de contexto. Dessa forma, o potencial de escalabilidade do aplicativo é preservado mesmo diante de variáveis externas.

6. Considerações Finais

Este artigo analisou o desenvolvimento de aplicação móvel em Flutter para o contexto de cidades inteligentes com foco em saúde pública municipal, defendendo que escalabilidade depende da convergência entre escolha tecnológica, arquitetura de software e governança de dados. A partir de revisão teórica, análise documental e estudo de caso técnico, foi possível demonstrar

que a combinação entre Flutter e princípios SOLID oferece base consistente para construir soluções mais manuteníveis, extensíveis e alinhadas à realidade operacional de governos locais.

Os resultados indicam que decisões arquiteturais orientadas a abstrações, contratos e separação de responsabilidades reduzem acoplamento e ampliam capacidade de evolução do sistema. Também se evidenciou que segurança e conformidade regulatória não podem ser tratadas como camadas adicionais ao fim do projeto; devem ser requisitos estruturantes da solução desde a etapa de desenho. Nesse sentido, a LGPD e as diretrizes de saúde digital atuam como parâmetros técnicos e institucionais para sustentabilidade da transformação digital em saúde.

Como contribuição aplicada, o trabalho apresenta modelo replicável para equipes que precisam construir aplicativos públicos com recursos limitados, mantendo qualidade de engenharia. Como contribuição acadêmica, integra campos frequentemente tratados de forma separada: smart cities, saúde digital e arquitetura de software. A principal limitação está na natureza exploratória da avaliação, ainda sem mensuração longitudinal de impacto assistencial. Recomenda-se, para trabalhos futuros, ampliar validação em ambiente produtivo, com indicadores quantitativos de acesso e eficiência.

Conclui-se, portanto, que o Flutter é alternativa tecnicamente adequada para projetos municipais de saúde digital quando inserido em estratégia arquitetural consistente e em governança pública orientada por dados, segurança e valor ao cidadão. A escalabilidade almejada por iniciativas de cidades inteligentes não é apenas tecnológica; ela é também organizacional, normativa e social. É nessa integração que aplicações móveis podem contribuir de forma efetiva para uma saúde pública mais acessível, transparente e resiliente.

Finalmente, a consolidação de soluções móveis no contexto de cidades inteligentes demanda visão de plataforma: arquitetura preparada para integração, governança de dados orientada a valor público e processos de engenharia capazes de absorver mudanças regulatórias e operacionais. O estudo apresentado reforça que essa visão é viável mesmo em contextos

municipais com recursos limitados, desde que exista coordenação entre decisão técnica e estratégia de gestão. Esse alinhamento amplia a capacidade do poder público de ofertar serviços de saúde mais oportunos, equitativos e confiáveis.

REFERÊNCIAS

AHMED, F. *et al.* Smart cities: health and safety for all. **The Lancet Public Health**, London, v. 2, n. 9, p. e398, 2017. DOI: [https://doi.org/10.1016/S2468-2667\(17\)30140-5](https://doi.org/10.1016/S2468-2667(17)30140-5). Acesso em: 07 abr. 2026.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023: informação e documentação - referências - elaboração**. Rio de Janeiro: ABNT, 2018.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 10520: informação e documentação - citações em documentos - apresentação**. Rio de Janeiro: ABNT, 2023.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14724: informação e documentação - trabalhos acadêmicos - apresentação**. Rio de Janeiro: ABNT, 2011.

BRASIL. **Lei n. 13.709, de 14 de agosto de 2018**. Dispõe sobre a proteção de dados pessoais e altera a Lei n. 12.965, de 23 de abril de 2014. **Diário Oficial da União**: seção 1, Brasília, DF, 15 ago. 2018.

BRASIL. Ministério da Saúde. **Estratégia de Saúde Digital para o Brasil 2020-2028**. Brasília, DF: Ministério da Saúde, 2020. Disponível em: <https://bvsms.saude.gov.br/bvs/publicacoes/estrategia_saude_digital_Brasil.pdf>. Acesso em: 07 abr. 2026.

FLUTTER. **Flutter documentation**. [S. l.]: Google, 2026. Disponível em: <<https://docs.flutter.dev/>>. Acesso em: 07 abr. 2026.

KAMEL BOULOS, M. N.; AL-SHORBAJI, N. M. On the Internet of Things, smart cities and the WHO Healthy Cities. **International Journal of Health Geographics**, London, v. 13, n. 10, p. 1-6, 2014. DOI: <https://doi.org/10.1186/1476-072X-13-10>. Acesso em: 07 abr. 2026.

MARTIN, R. C. **Arquitetura limpa: o guia do artesanato para estrutura e design de software**. Rio de Janeiro: Alta Books, 2019.

MARTIN, R. C. **Functional design: principles, patterns, and practices**. Boston: Addison-Wesley Professional, 2023.

MEYER, B. **Object-oriented software construction**. 2. ed. Upper Saddle River: Prentice Hall, 1997.

ROCHA, N. P. *et al.* Smart cities and public health: a systematic review. **Procedia Computer Science**, Amsterdam, v. 164, p. 516-526, 2019. DOI: <https://doi.org/10.1016/j.procs.2019.12.213>. Acesso em: 07 abr. 2026.

SOLANAS, A. *et al.* Smart health: a context-aware health paradigm within smart cities. **IEEE Communications Magazine**, New York, v. 52, n. 8, p. 74-81, 2014. DOI: <https://doi.org/10.1109/MCOM.2014.6871673>. Acesso em: 07 abr. 2026.

WORLD HEALTH ORGANIZATION. **Global strategy on digital health 2020-2025**. Geneva: World Health Organization, 2021. Disponível em: <<https://www.who.int/publications/i/item/9789240020924>>. Acesso em: 07 abr. 2026.