

APLICATIVO WEB MOBILE PARA CONTROLE DA VACINAÇÃO A FIM DE FACILITAR OS CUIDADOS COM A SAÚDE DE CRIANÇAS.

Luiz Carlos Ellwanger Parcianello, Hugo André Klauck

luiz.parcianello@gmail.com, hugo.klauck@farroupilha.ifrs.edu.br

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande Do Sul (Ifrs)
Campus Farroupilha

Farroupilha - RS

Resumo: O presente Trabalho de Conclusão de Curso propõe o desenvolvimento de um aplicativo chamado Dia de Vacina Brasil que tem como objetivo auxiliar pais e responsáveis no controle da vacinação infantil. O aplicativo será desenvolvido utilizando o framework Ionic, utilizando tecnologias como Angular, Firebase, HyperText Markup Language (HTML), Cascading Style Sheets (CSS), JavaScript e TypeScript. Sua arquitetura robusta e coesa permitirá autenticação segura de usuário, a gestão eficiente de dados e o envio eficaz de notificações por e-mail e push. O sistema possibilitará o cadastro de usuários, inserção de dados de crianças, inclusão do cronograma de vacinas, envio de alertas, se possível, notificação sonora. A vacinação só poderá ser considerada concluída após confirmação pelo responsável, incluindo data da aplicação e lote do imunizante. A solução proposta visa facilitar o acompanhamento da imunização e incentivar a adesão ao calendário nacional de vacinação. Contribuindo com a saúde pública.

Abstract: This Final Paper proposes the development of a mobile application called Dia de Vacina Brasil, which aims to assist parents and guardians in managing childhood vaccination. The application will be developed using the Ionic framework and various technologies, including Angular, Firebase, HTML, CSS, JavaScript, and TypeScript. Its robust and cohesive architecture will enable secure user authentication, efficient data management, and the effective delivery of notifications via email and push. The system will allow user registration, input of child data, creation of vaccination schedules, and sending of alerts, if possible, sound notification. A vaccination will only be considered completed after confirmation by the guardian, including the application date and the batch number of the vaccine. The proposed solution aims (seeks) to facilitate immunization tracking and (encourage) promote adherence to the national vaccination calendar, (thus) thereby contributing to public health.

1 INTRODUÇÃO

A vacinação é uma das formas mais eficazes de prevenção de doenças, especialmente na infância. No entanto, muitos pais ou responsáveis em todo o mundo enfrentam dificuldades para acompanhar o calendário vacinal, que é essencial para atingir os Índices de Cobertura Vacinal (ICV), colocando em risco a saúde de seus filhos (Domingues *et al.*, 2020). Em tempos de avanço tecnológico e alta penetração de smartphones, o desenvolvimento de soluções digitais para facilitar esse acompanhamento torna-se não apenas viável, mas necessário.

O Estatuto da Criança e do Adolescente (ECA), conforme Brasil (1990), reforça a responsabilidade da sociedade e do poder público na promoção dos direitos das crianças, incluindo o direito à saúde. No entanto, segundo Reis (2022), o período pós-COVID-19 trouxe preocupações com a queda nas taxas de vacinação infantil, destacando a necessidade de soluções inovadoras.

Este projeto tem como proposta a criação de um aplicativo chamado Dia de Vacina Brasil, cujo objetivo é auxiliar os responsáveis no controle e acompanhamento das vacinas infantis. O aplicativo permitirá o cadastro de múltiplas crianças, a geração e visualização de cronogramas vacinais, o envio de lembretes por e-mail e, quando possível, alertas sonoros no dispositivo móvel do usuário, preenchendo a lacuna de soluções automatizadas apontada.

Este artigo apresentará, além dos aspectos técnicos, tecnológicos e metodológicos fundamentais para a compreensão do funcionamento do aplicativo, a visão, os recursos e a possibilidade de benefícios da solução proposta. O objetivo é destacar sua possível contribuição para melhorar a adesão à imunização infantil e, assim, promover a saúde e o bem-estar das futuras gerações. O restante deste artigo apresenta os objetivos e justificativa do projeto, seguidos das seguintes seções: Análise de soluções existentes e proposta de valor, Análise e modelagem, descrevendo as principais ferramentas e tecnologias utilizadas, bem como o desenvolvimento do aplicativo, descrevendo sua arquitetura, componentes e módulos, além de melhorias futuras.

2 OBJETIVO GERAL

O objetivo deste artigo é desenvolver e apresentar um aplicativo multiplataforma web, com o *framework Ionic* e o *ecossistema Firebase*, que permita aos responsáveis um controle eficiente das vacinas dos dependentes conforme o Programa Nacional de Imunização (PNI) (Brasil, 2023).

A proposta inclui funcionalidades como notificações por *e-mail* e alertas via *push*, oferecendo uma solução prática e inovadora para facilitar o acompanhamento das datas de vacinação estipuladas pelo PNI ou adicionadas manualmente pelos pais, fornecendo um meio eficaz de alertas e um local para o armazenamento de informações da caderneta de vacinação, garantindo um registro histórico detalhado e acessível que será um ativo fundamental para a saúde do indivíduo em todas as fases da vida.

Além disso, o artigo busca destacar como o projeto pode contribuir para a melhoria da adesão à imunização infantil no Brasil, promovendo a saúde e o bem-estar das crianças e por fim demonstrar parte das habilidades desenvolvidas no curso superior de Análise e Desenvolvimento de Sistemas (ADS).

2.1 Objetivos Específicos:

- Projetar e implementar uma arquitetura de software robusta e segura, utilizando o conhecimento adquirido no curso de ADS, para o desenvolvimento do aplicativo web mobile Dia de Vacina Brasil.
- Desenvolver funcionalidades de cadastro e gerenciamento personalizado de perfis de usuários e crianças, incluindo a criação e visualização de cronogramas vacinais conforme as diretrizes do PNI.
- Implementar um sistema automatizado de alertas e notificações, assegurando a entrega eficaz de lembretes sobre vacinações futuras e atrasadas.
- Estabelecer um mecanismo de registro detalhado de vacinas aplicadas incluindo data e lote e sirva como um histórico vacinal digital confiável.
- Implementar a integração robusta e segura com um sistema de Inteligência Artificial (IA) que aprimore a interação do usuário, atuando como um guia inteligente e fornecendo informações proativas no uso do aplicativo.

3 JUSTIFICATIVA

A criação do aplicativo Dia de Vacina Brasil surge como uma resposta relevante aos desafios enfrentados na saúde infantil brasileira, especialmente no que se refere à adesão ao Programa Nacional de Imunizações (PNI). Apesar do histórico positivo do Brasil na promoção da vacinação como a campanha liderada por Oswaldo Cruz em 1904, que marcou o início da saúde pública no país (Brasil, 2003) ainda persistem dificuldades no controle e no acompanhamento das vacinas pelas famílias. Entre os principais obstáculos estão a falta de tempo, a sobrecarga de tarefas e a dificuldade de acesso e gestão eficaz das informações (Silva; Teixeira, 2024).

Nesse contexto, o aplicativo se apresenta como uma ferramenta tecnológica com potencial para fortalecer a saúde coletiva e garantir o direito à vida e à saúde das crianças. Essa perspectiva é corroborada Taschner e Almeida (2023), ao afirmarem que “a vacina infantil é um direito da criança, por outro, também é uma iniciativa de saúde coletiva que beneficia toda a sociedade”. Embora o calendário vacinal esteja disponível nos sites oficiais, como o do Ministério da Saúde e da Sociedade Brasileira de Imunizações (La Porta; Lima, 2022), a simples divulgação digital não assegura o acompanhamento ativo por parte dos responsáveis, o que reforça a necessidade de soluções mais acessíveis, automatizadas e personalizadas. Para contextualizar a complexidade e a abrangência do desafio, a Tabela 1 apresenta o calendário básico de vacinação do PNI para crianças.

Tabela 1: Calendário Básico de Vacinação da Criança no PNI.

Faixa Etária	Vacina	Descrição
Ao nascimento	BCG	Protege contra formas graves de tuberculose (meningite tuberculosa e tuberculose miliar).
Ao nascimento	Hepatite B (1ª dose)	Protege contra o vírus da hepatite B.
2 meses	Penta (DTP + Hib + Hepatite B)	Combinada, protege contra Difteria, Tétano, Coqueluche, Hib (meningite e pneumonia) e Hepatite B.
2 meses	VIP (Poliomielite)	Protege contra a paralisia infantil (poliomielite).
2 meses	Pneumocócica 10 valente	Protege contra infecções graves causadas pela bactéria <i>Streptococcus pneumoniae</i> .
2 meses	Meningocócica C (1ª dose)	Protege contra infecções por <i>Neisseria meningitidis</i> tipo C (meningite e septicemia).
2 meses	Rotavírus (1ª dose)	Protege contra o rotavírus, principal causador de diarreias graves em crianças pequenas.
4 meses	Penta (DTP + Hib + Hepatite B)	2ª dose, reforço da vacina combinada.
4 meses	VIP (Poliomielite)	2ª dose, reforço da vacina contra poliomielite.
4 meses	Rotavírus (2ª dose)	2ª dose da vacina contra o rotavírus.

4 meses	Hepatite B (2ª dose)	Reforço da vacina contra hepatite B.
6 meses	Hepatite B (3ª dose)	Completa o esquema de vacinação contra hepatite B.
6 meses	Pneumocócica 10 valente (2ª dose)	Reforço da vacina pneumocócica.
6 meses	Meningocócica C (2ª dose)	Reforço da vacina contra meningite tipo C.
12 meses	Tríplice viral (SCR)	Protege contra Sarampo, Caxumba e Rubéola.
12 meses	Hib (3ª dose)	Reforço da vacina contra infecções causadas por Haemophilus influenzae tipo b.
12 meses	Pneumocócica 10 valente (3ª dose)	Reforço da vacina pneumocócica contra infecções respiratórias.
12 meses	Meningocócica C (Reforço)	Reforço da vacina contra meningite tipo C.
15 meses	DTP (Reforço)	Reforço contra Difteria, Tétano e Coqueluche.
15 meses	Poliomielite (Reforço)	Reforço da vacina contra poliomielite.
15 meses	Hepatite A (1ª dose)	Protege contra a Hepatite A.
15 meses	Febre amarela (1ª dose)	Protege contra a febre amarela.
15 meses	Varicela (1ª dose)	Protege contra a catapora.
4 anos	DTP (Reforço)	Reforço contra Difteria, Tétano e Coqueluche.
4 anos	Poliomielite (Reforço)	Reforço da vacina contra poliomielite.
4 anos	Tríplice viral (Reforço)	Reforço contra Sarampo, Caxumba e Rubéola.
9 anos	HPV (1ª dose)	Vacina contra o Papilomavírus Humano (HPV), para meninos e meninas.
9 anos	Meningocócica ACWY	Protege contra a Meningite causada por Neisseria meningitidis (tipos A, C, W e Y).
10 a 11 anos	HPV (2ª e 3ª doses)	Reforço da vacina contra o Papilomavírus Humano (HPV).
10 a 11 anos	DTP (Reforço)	Reforço contra Difteria, Tétano e Coqueluche.
10 a 11 anos	Meningocócica B	Vacina contra a Meningite causada por Neisseria

Fonte: Adaptado de BRASIL (2025).

A queda nas taxas de vacinação infantil no período pós-COVID-19, apontada pela UNICEF (2023) como o maior retrocesso global dos últimos 30 anos, evidencia a urgência de estratégias inovadoras. O Dia de Vacina Brasil ocupar esse espaço ao oferecer funcionalidades como lembretes por *e-mail* e notificações *push*, armazenamento do histórico vacinal e possibilidade de adicionar vacinas conforme o PNI ou extras manualmente. A proposta é empoderar pais e cuidadores, melhorar o controle vacinal e ampliar a cobertura imunológica de forma prática e acessível.

Além de contribuir com a saúde pública, o desenvolvimento do aplicativo representa uma oportunidade de aplicar conhecimentos adquiridos ao longo do curso de Análise e Desenvolvimento de Sistemas, como gerenciamento de projetos,

engenharia de *software*, programação, banco de dados, usabilidade e integração de tecnologias.

O projeto ao colaborar com a efetivação dos direitos à saúde, à vida e à convivência familiar, também reforça os princípios do Estatuto da Criança e do Adolescente (Artigo 4º):

Art. 4º É dever da família, da comunidade, da sociedade em geral e do poder público assegurar, com absoluta prioridade, a efetivação dos direitos referentes à vida, à saúde, à alimentação, à educação, ao esporte, ao lazer, à profissionalização, à cultura, à dignidade, ao respeito, à liberdade e à convivência familiar e comunitária. (BRASIL, 1990).

Diante do contexto apresentado, este artigo propõe-se a descrever o aplicativo, seus recursos e a sua relevância para a saúde pública, apoiando-se no legado histórico da imunização no Brasil (BRASIL, 2003), na urgência expressa por Reis (2022) e nos benefícios esperados para pais, cuidadores e toda a sociedade, com especial destaque para o valor inestimável de um histórico vacinal digital completo e acessível, que acompanhará o indivíduo desde a infância até a vida adulta.

4 ANÁLISE DE SOLUÇÕES EXISTENTES E PROPOSTA DE VALOR

Esta seção dedica-se a analisar as principais ferramentas e plataformas digitais existentes no cenário brasileiro que visam, de alguma forma, auxiliar no controle e acompanhamento da vacinação. A identificação de suas funcionalidades, pontos fortes e limitações é crucial para justificar a proposta de valor do aplicativo Dia de Vacina Brasil, evidenciando as lacunas que a presente solução busca preencher e seus diferenciais inovadores.

4.1 Ferramentas Governamentais: Meu SUS Digital (Antigo Conecte SUS Cidadão)

Parte do ecossistema Estratégia de Saúde Digital para o SUS (e-SUS) o Meu SUS Digital (antigo Conecte SUS Cidadão), aplicativo oficial do Ministério da Saúde, se destaca como a principal plataforma do governo federal para que o cidadão acesse seu histórico de saúde, incluindo dados de vacinação registrados na rede pública. Ele permite a visualização da caderneta de vacinação digital e a emissão de certificados,

como o de vacinação contra a COVID-19. Embora seja uma ferramenta oficial e de abrangência nacional, sua natureza como repositório de dados apresenta algumas limitações que o Dia de Vacina Brasil se propõe a mitigar:

- Proatividade das notificações: O foco principal do Meu SUS Digital é o histórico de registros. Não há um sistema robusto e proativo de alertas e lembretes por e-mail e notificações *push* para as próximas doses do calendário do PNI, ou para datas de vacinação adicionadas manualmente, exigindo que o usuário consulte ativamente o aplicativo.
- Flexibilidade de inserção de dados: A plataforma depende exclusivamente da inserção de dados pelas unidades de saúde no sistema público. Não permite que pais ou responsáveis adicionem manualmente vacinas da rede privada, informações adicionais ou corrijam inconsistências de forma direta, o que pode resultar em um histórico incompleto sob a perspectiva do usuário final.
- Usabilidade focada: Por ser um aplicativo de saúde mais amplo, sua interface pode ser menos intuitiva para o dia a dia do gerenciamento exclusivo das vacinas infantis de múltiplos dependentes.

4.2 Aplicativos de Clínicas/Laboratórios Privados e Outras Iniciativas

Alguns laboratórios, clínicas de vacinação ou hospitais privados oferecem seus próprios aplicativos para acompanhamento das vacinas administradas em suas instalações. Paralelamente, existem aplicativos genéricos de saúde como "Minhas Vacinas" da Sociedade Brasileira de Imunização (SBIIm) ou módulos em *apps* de acompanhamento de bebês, que incorporam funcionalidades de controle vacinal. Essas soluções, embora úteis em seus respectivos nichos, frequentemente apresentam desafios como:

- Fragmentação da informação: O uso de múltiplos aplicativos de diferentes instituições privadas fragmenta o histórico vacinal da criança, exigindo que o responsável consulte diversas fontes para ter uma visão completa.
- Escopo limitado: Aplicativos de clínicas focam apenas nas vacinas administradas por elas, não consolidando o calendário completo do PNI ou vacinas tomadas em outras redes (pública ou particular). Aplicativos genéricos podem ter funcionalidades de vacinação mais superficiais, sem a profundidade de alertas e personalização necessária.

- Personalização restrita: A capacidade de adicionar vacinas manualmente, com detalhes como lote e data de aplicação, ou de personalizar alertas para vacinas fora do calendário padrão do PNI, é limitada ou inexistente.

4.3 Diferenciais e Proposta de Valor do Dia de Vacina Brasil

Diante das análises das ferramentas existentes, o aplicativo Dia de Vacina Brasil se posiciona como uma solução complementar e inovadora, focada em otimizar o controle e a adesão à vacinação infantil. Seus principais diferenciais e proposta de valor residem em:

- Proatividade aprimorada: O sistema de notificações por e-mail e alertas via *push* (com notificação sonora, se possível) representa um diferencial crucial. Diferentemente de soluções que exigem consulta ativa, o Dia de Vacina Brasil age como um lembrete inteligente, garantindo que pais e responsáveis não percam datas importantes de vacinação, minimizando atrasos na imunização.
- Flexibilidade e controle parental: A possibilidade de adicionar vacinas manualmente (seja conforme o PNI ou outras vacinas adicionais), registrar detalhes como data da aplicação e lote do imunizante, e ter um local centralizado para o armazenamento das informações da caderneta de vacinação, oferece aos pais um controle completo e um registro digital unificado, independente da fonte da vacina.
- Foco específico e usabilidade: Ao concentrar-se exclusivamente na gestão de vacinas infantis (até 12 anos), o aplicativo oferece uma interface mais limpa, intuitiva e otimizada para essa tarefa específica, tornando a experiência do usuário mais eficiente e agradável do que em plataformas multifuncionais.
- Contribuição para a saúde pública: Ao facilitar a adesão ao calendário vacinal e empoderar os responsáveis com informações e lembretes proativos, o projeto tem a intenção de colaborar diretamente com o aumento das coberturas vacinais no Brasil, promovendo a saúde e o bem-estar das crianças em conformidade com o PNI e os princípios do ECA.

5 ANÁLISE E MODELAGEM

Esta seção detalha o processo de análise dos requisitos do projeto Dia de Vacina Brasil e a subsequente modelagem do sistema, delineando sua arquitetura, funcionalidades e interações. Aqui, são apresentadas as ferramentas e metodologias utilizadas para transformar as necessidades levantadas em uma estrutura de software compreensível e implementável, culminando na definição das tecnologias que sustentam o desenvolvimento do aplicativo.

5.1 Modelagem

A modelagem desempenha um papel crucial na engenharia de software, servindo como um guia estruturado e visual para a criação de sistemas eficazes e funcionais (Sommerville, 2018). Neste tópico, apresentaremos os artefatos de modelagem propostos para a construção do software Dia de Vacina Brasil. Esses artefatos desempenham um papel fundamental na definição, planejamento e implementação do aplicativo, permitindo uma compreensão clara dos componentes e processos envolvidos. A Tabela 2 detalha a classificação e a descrição de cada um desses requisitos, servindo como base para as próximas etapas de design e implementação.

Tabela 2 – Requisitos funcionais, não funcionais, de sistema e regras de negócio.

ID	Tipo	Descrição
RF 01	Requisito Funcional	O aplicativo deve permitir o cadastro de usuários com informações como nome, senha e <i>email</i> .
RF 02	Requisito Funcional	O sistema deve possibilitar o registro de vacinas tomadas, incluindo data, tipo de vacina, lote e unidade de saúde.
RF 03	Requisito Funcional	Deve haver um calendário de vacinação para cada dependente do usuário, com alertas para vacinas futuras e atrasadas.
RF 04	Requisito Funcional	O aplicativo deve permitir a criação de perfis para múltiplos dependentes, com controle separado das vacinas de cada um.
RF 05	Requisito Funcional	O usuário deve poder acessar um histórico completo das vacinas tomadas e agendadas.
RNF 01	Requisito Funcional Não	O aplicativo deve ser compatível com dispositivos Android e iOS.
RNF 02	Requisito Funcional Não	A interface deve ser intuitiva e fácil de usar para pessoas de todas as idades.
RNF 03	Requisito Funcional Não	Os dados do usuário devem ser armazenados com segurança, utilizando criptografia adequada.
RNF 04	Requisito Funcional Não	O sistema deve ser capaz de enviar notificações <i>push</i> , (<i>Firebase Cloud Messaging</i> (FCM) para Android e iOS) para os usuários sobre vacinas próximas ou atrasadas.
RNF 05	Requisito Funcional Não	O aplicativo deve ter um desempenho eficiente, carregando páginas em menos de 2 segundos e funcionando de forma responsiva mesmo com uma grande quantidade de dados.

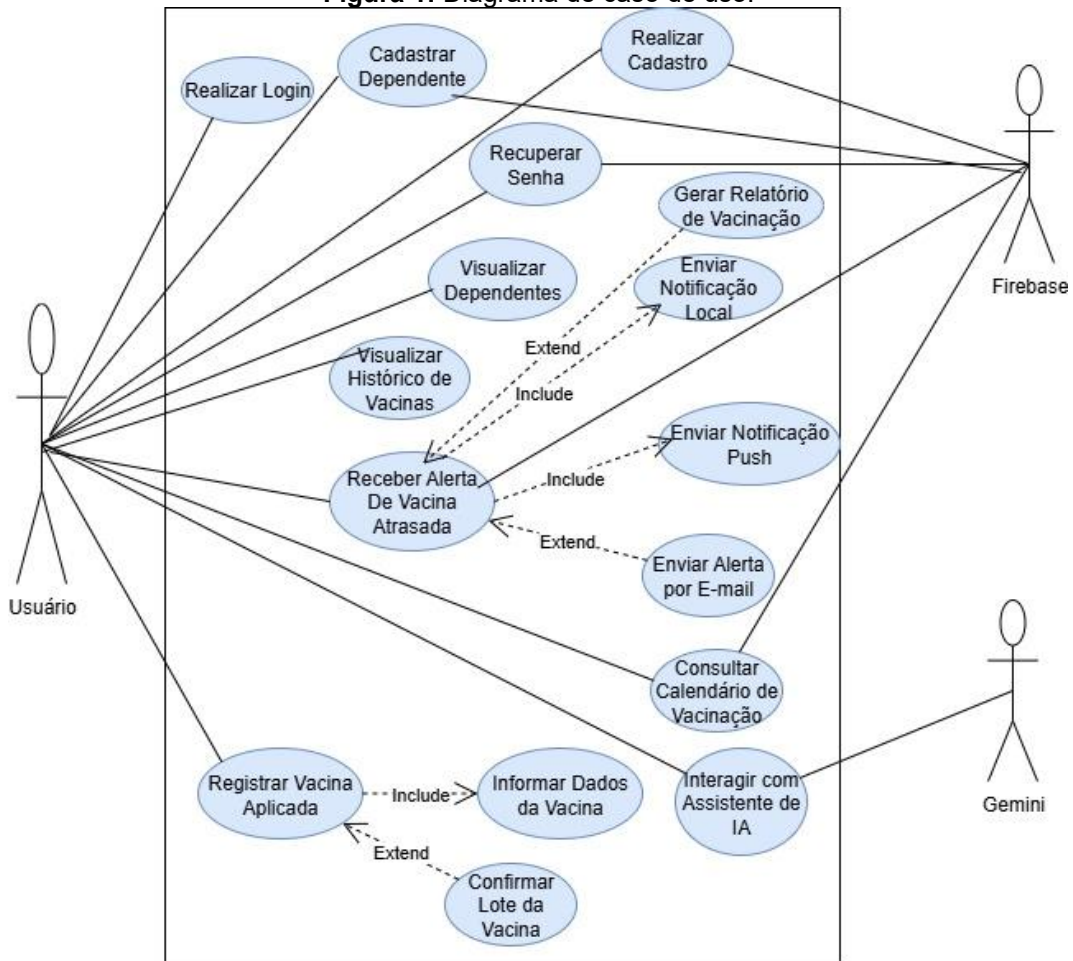
RS 01	Requisitos Sistema	de	Possibilitar a criação de login e senha.
RS 02	Requisitos Sistema	de	Possibilitar a inserção de dependente.
RS 03	Requisitos Sistema	de	O sistema deve gravar e persistir todas as informações inseridas pelos usuários e geradas internamente.
RS 04	Requisitos sistema	de	Coletar r organizar dados de ações e histórico do aplicativo para futuras análises.
RS 05	Requisitos Sistema	de	Sistema emite alerta externo via e-mail. Nos emails cadastrados.
RS 06	Requisitos Sistema	de	O sistema deve ser capaz de integrar e interagir com uma inteligência artificial (Google Gemini API) para fornecer respostas e informações sobre o próprio aplicativo, vacinação e saúde.
RN 01	Regra de negócio		Se o aviso não for validado com o lote da vacina, os disparos de aviso serão refeitos até a validação.
RN 02	Regra de negócio		Após o 3° disparo de aviso via <i>push</i> não atendido para a mesma dose, será executado um aviso por e-mail.
RN 03	Regra de negócio		O sistema não deve permitir o cadastro de dois usuários com o mesmo e-mail, devendo apresentar mensagem de erro caso a tentativa ocorra.

Fonte: Autoria própria, 2025.

5.1.2 Casos de Uso

Para uma melhor visualização das funcionalidades básicas será elaborado um Diagrama de Casos de Uso utilizando o *software Enterprise Architect*, conforme descrito em (SOMMERVILLE, 2019) e apresentado na Figura 1.

Figura 1: Diagrama de caso de uso.



Fonte: Autoria Própria, 2025.

5.1.3 Entidade e Relacionamento

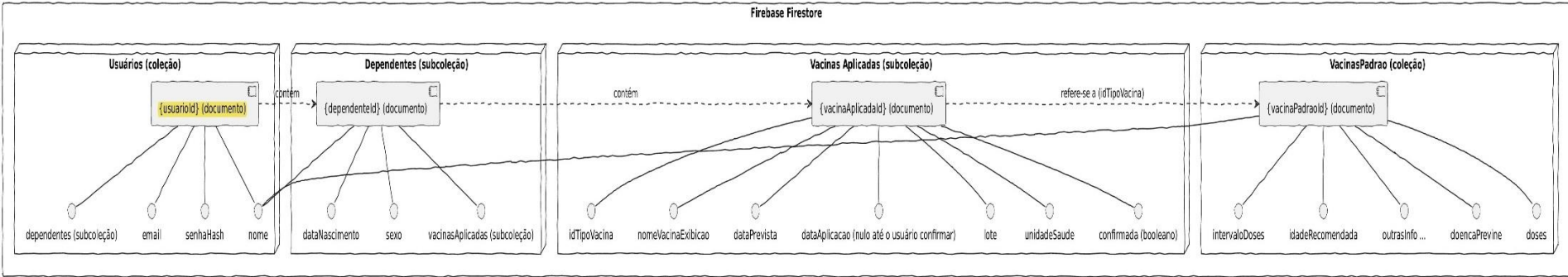
Nesta seção, é apresentada a organização dos dados que compõem a base do aplicativo Dia de Vacina Brasil, agora estruturada conforme os princípios de um banco de dados não relacional, utilizando o *Firebase*. Ao contrário dos modelos relacionais tradicionais, que se baseiam em tabelas e chaves estrangeiras, o *Firestore* trabalha com coleções e documentos aninhados, o que proporciona maior flexibilidade e escalabilidade, especialmente em aplicações móveis.

A Figura 2 ilustra essa estrutura de forma hierárquica. Nela, o usuário autenticado possui seus dados armazenados em um documento dentro da coleção Usuários. Associada a esse documento, há uma subcoleção dependentes, representando as crianças sob responsabilidade do usuário. Cada dependente possui

Figura 2: Diagrama de relacionamento NoSQL.

```

Usuários (coleção)
├── L- {usuarioId} (documento)
│   ├── nome
│   ├── email
│   ├── senhaHash
│   └── dependentes (subcoleção)
│       └── L- {dependenteId} (documento)
│           ├── nome
│           ├── dataNascimento
│           ├── sexo
│           └── vacinasAplicadas (subcoleção)
│               └── L- {vacinaAplicadaId} (documento)
│                   ├── idTipoVacina <-- OPCIONAL (se for vacina da VacinasPadrao)
│                   ├── nomeVacina <-- OBRIGATÓRIO (se não tiver idTipoVacina)
│                   ├── doencaPrevine <-- OBRIGATÓRIO (se não tiver idTipoVacina)
│                   ├── doses <-- OPCIONAL
│                   ├── intervaloDoses <-- OPCIONAL
│                   ├── idadeRecomendada <-- OPCIONAL
│                   ├── dataPrevista
│                   ├── dataAplicacao (nulo até o usuário confirmar)
│                   ├── lote
│                   ├── unidadeSaude
│                   └── confirmada (booleano)
└── VacinasPadrao (coleção) -- Esta coleção continua existindo e é a fonte para vacinas "oficiais"
    └── L- {vacinaPadraoId} (documento)
        ├── nome
        ├── doencaPrevine
        ├── doses
        ├── intervaloDoses
        ├── idadeRecomendada
        └── outrasInfo ...
    
```



Fonte: Autoria Própria, 2025

sua própria subcoleção Vacinas, onde são registradas informações como o nome da vacina, a data prevista para aplicação, a data efetiva (quando confirmada), o lote do imunizante, a unidade de saúde e o status de confirmação.

Essa modelagem facilita a geração automatizada de alertas baseados na data de nascimento do dependente e na data prevista para aplicação da vacina. O campo de confirmação com data e lote é indispensável para encerrar o ciclo de notificações, garantindo que o processo de vacinação esteja concluído de forma segura e registrada.

5.1.4 Protótipo

Uma das interfaces mais importantes do projeto abordado foi desenvolvida durante o curso em uma fase preliminar do projeto, e está sendo exposta na figura 3. Visa trazer uma ilustração onde pode-se perceber total compatibilidade, tendo representados principais itens de persistência no banco de dados.

Figura 3: Protótipo da tela de confirmação de vacina e atalhos permanentes.

O protótipo da tela de confirmação de vacina apresenta o seguinte layout:

- Barra superior: Nome do usuário "Junior" com ícones de menu e notificação.
- Formulário de entrada:
 - Nome da vacina: campo de texto.
 - Lote: campo de texto.
 - Dose: menu suspenso com "2ª dose" selecionado.
 - Data: campo de data com "11/11/2011" e ícone de calendário.
 - Responsável: campo de texto com placeholder "nome / crm / de quem fez a vacina / info".
- Status da Vacina: seção com botões "Realizados", "Agendadas" e "Agendar".
- Barra de atalhos: ícones para notificação, calendário, documento, link e informações.

Fonte: Autoria própria, 2025

5.2 Tecnologias

Das tecnologias utilizadas neste projeto, é interessante salientar que foram amplamente estudadas ao longo do curso de Análise e Desenvolvimento de Sistemas, tornando sua aplicação especialmente relevante para a consolidação do conhecimento adquirido. Este projeto representa não apenas a aplicação prática desses conhecimentos, mas também o desenvolvimento de um modelo funcional inicial que poderá ser aprimorado futuramente. A seguir, são descritas as ferramentas e tecnologias utilizadas:

Enterprise Architect: é um ambiente de modelagem completo para grupos de trabalho, analistas e desenvolvedores. Inclui suporte para projetos multiusuário e engenharia de código (importação / exportação / sincronização), utilizado para a criação de diagramas de casos de uso e outros artefatos de modelagem. (SPARXSYSTEMS, 2019).

Balsamiq: é uma ferramenta rápida de prototipagem da interface do usuário de baixa fidelidade foi usada na figura 3, ela reproduz a experiência de desenhar em um bloco de notas ou quadro branco, mas usando um computador, utilizada para criar protótipos de telas como a de confirmação de vacina. (BALSAMIQ, 2019).

HTML (HyperText Markup Language): Linguagem padrão para criação de páginas web, utilizada na estruturação do conteúdo do aplicativo. (W3SCHOOLS, 2019).

CSS (Cascading Style Sheets): é responsável por definir o estilo visual dos elementos criados com HTML. Com ele, é possível configurar cores, fontes, posicionamentos, tamanhos e tornar o design responsivo, para que se adapte bem em diferentes dispositivos, como celulares e tablets. (W3SCHOOLS, 2019).

(SCSS) Sassy Cascading Style Sheets: é um pré-processador da linguagem *Cascading Style Sheets* (CSS) que amplia suas funcionalidades ao permitir o uso de variáveis, aninhamento de seletores, *mixin* (mecanismo de reutilização de código que permite que classes compartilhem funcionalidades sem a necessidade de herança), funções e estruturas de controle. Diferentemente do CSS tradicional, o código escrito em SCSS não é interpretado diretamente pelos navegadores, ele precisa ser compilado e sofrer conversão de código para CSS puro ("*transpilado*"), que então é aplicado ao documento HTML. Essa abordagem oferece maior organização, reutilização e manutenção dos estilos, sendo especialmente vantajosa

no desenvolvimento de interfaces complexas e escaláveis, como em aplicações desenvolvidas com *frameworks* modernos, como o *Angular* e o *Ionic*. (SASS, 2025).

JavaScript: é uma linguagem de programação interpretada estruturada, de script em alto nível com tipagem dinamicamente fraca e multi-paradigma. Acompanhado de HTML e CSS, o *JavaScript* é uma das três principais tecnologias da *World Wide Web*. É uma linguagem de programação que permite implementar funcionalidades mais complexas em páginas web. A cada momento uma página web faz mais do que apenas mostrar informações estáticas para você, se ela mostra em tempo real conteúdos atualizados, mapas interativos, animações gráficas em 2D/3D, vídeos etc. o *JavaScript* provavelmente está envolvido. (MDN WEB DOCS, 2025)

TypeScript: é um superconjunto do *JavaScript* que adiciona tipagem estática e outros recursos que tornam o código mais seguro, limpo e fácil de manter. Com *TypeScript*, é possível identificar erros ainda durante o desenvolvimento, o que ajuda a evitar problemas futuros. É muito usado com *frameworks* modernos como o *Ionic* e *Angular*. (TYPESCRIPTLANG, 2024).

Angular: é um *framework* de desenvolvimento de aplicações *frontend* de código aberto, mantido pelo Google. Ele permite construir aplicações web complexas e de página única (*Single Page Applications - SPAs*) de forma estruturada e eficiente, utilizando *TypeScript*. No contexto do *Ionic*, o *Angular* é a base para a criação dos componentes e da lógica da interface do usuário. (ANGULAR, 2025).

Ionic Framework: é o principal *framework* utilizado para o desenvolvimento do aplicativo. Ele permite criar aplicativos híbridos, ou seja, que funcionam tanto em Android quanto em iOS a partir de um único código-fonte. Utiliza tecnologias como HTML, CSS, *JavaScript* e *TypeScript*, oferecendo uma série de componentes prontos e bem adaptados às diretrizes visuais dos sistemas operacionais móveis. (IONIC, 2025).

Capacitor: é um ambiente de execução nativo de código aberto para a criação de aplicativos nativos da Web. Cria aplicativos da Web multiplataforma para iOS, Android e progressivos com *JavaScript*, HTML e CSS. Ele permite que o aplicativo web *Ionic* seja executado como um aplicativo nativo em iOS e Android. Ele atua como uma ponte, encapsulando o aplicativo web em um *WebView* nativo. (CAPACITOR, 2025).

- Capacitor - Local Notifications (@capacitor/local-notifications): Plugin específico do Capacitor utilizado para agendar e exibir notificações

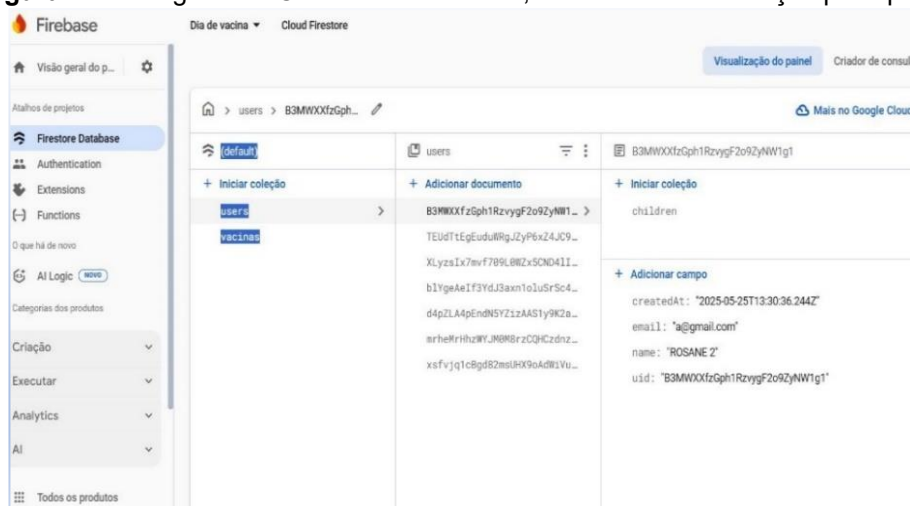
diretamente no dispositivo do usuário, ideal para lembretes de vacinação personalizados sem a necessidade de um servidor. (CAPACITOR, 2025).

- *Capacitor - Push Notifications (@capacitor/push-notifications)*: Plugin específico do *Capacitor* para receber e manipular notificações enviadas por um servidor externo (como o *Firebase Cloud Messaging*), permitindo que o aplicativo reaja a mensagens mesmo quando em segundo plano ou fechado. (CAPACITOR, 2025).

Google Gemini API: Serviço de Inteligência Artificial do Google que fornece acesso a modelos de linguagem avançados. No aplicativo, é utilizada para alimentar o assistente virtual, permitindo que ele compreenda as perguntas dos usuários e gere respostas relevantes e direcionadas sobre saúde infantil, vacinação e uso do aplicativo. (GOOGLE, 2025).

Firebase: é a plataforma usada como *backend* principal do sistema. Ele oferece armazenamento de dados em tempo real (*Cloud Firestore*), autenticação de usuários, notificações *push* (FCM), *analytics* e outras ferramentas. No projeto, será utilizado para armazenar os dados das crianças e vacinas, além de realizar login e cadastro de usuários. (FIREBASE, 2024). Para contextualizar a escolha do *Firebase*, a Figura 4 apresenta uma visão geral do console da plataforma, destacando as principais ferramentas disponíveis para o desenvolvimento e gerenciamento do projeto.

Figura 4: Visão geral do Console do Firebase, destacando os serviços principais.



Fonte: Autoria Própria (captura de tela do console Firebase), 2025.

A escolha pelo *Firebase* se justifica pela sua robustez e pela integração de diversos serviços essenciais para a aplicação:

- *Firebase Cloud Firestore*: Banco de dados NoSQL flexível e escalável,

desenvolvimento. Para o envio de mensagens automáticas, será usado o *Firebase Cloud Messaging*, que permite notificações *push* mesmo com o app em segundo plano. O envio automático de *e-mails* será implementado com as *Trigger Email Extensions* e *Cloud Functions*, que disparam mensagens conforme eventos no banco de dados e regras do sistema.

6 DESENVOLVIMENTO

Esta seção detalha o processo de desenvolvimento do aplicativo móvel dia de vacina Brasil, abordando a arquitetura da solução, os principais componentes e módulos, e a implementação das funcionalidades chave, com foco especial no sistema de notificações, à integração de um assistente virtual e armazenamento de dados.

Embora o desenvolvimento do projeto tenha sido realizado individualmente, adotou-se uma abordagem iterativa e incremental, inspirada nos princípios das metodologias ágeis (SOMMERVILLE, 2018). Essa estratégia permitiu o aprimoramento contínuo do aplicativo por meio de ciclos curtos de implementação, teste e ajuste.

Na prática, essa escolha possibilitou o desenvolvimento e a validação progressiva das funcionalidades, com a identificação e correção rápida de falhas. Um exemplo foi a implementação da verificação de e-mail único no cadastro de usuários: inicialmente, o sistema aceitava e-mails duplicados, o que comprometia a integridade dos dados. Após testes, o código foi ajustado para consultar previamente o banco de dados e bloquear registros duplicados, exibindo uma mensagem informativa ao usuário.

Essa metodologia favoreceu entregas parciais e o alinhamento constante entre o projeto e os requisitos funcionais e não funcionais definidos no escopo.

6.1 Arquitetura da Solução

A arquitetura do aplicativo segue um modelo cliente serviço, com foco em simplicidade, escalabilidade e facilidade de manutenção. A estrutura foi dividida em três camadas principais, uma das formas mais robustas de desenvolvimento, como exposto no texto de IBM (2025). Essa abordagem permite separar claramente a interface do usuário, a lógica de negócios e o armazenamento de dados, promovendo modularidade e evolução independente entre os componentes.

6.1.1 Camada Cliente/Apresentação (*Frontend*)

Esta camada é responsável pela interface gráfica e pela experiência do usuário. Utiliza o *Ionic Framework* com *Angular*, permitindo a construção de um aplicativo multiplataforma (Progressive Web App, *iOS* e *Android*) com uma única base de código. Além da exibição de dados e interação com o usuário, o *frontend* também realiza chamadas aos serviços de *backend* por meio de APIs, mantendo a lógica do cliente separada da lógica do servidor.

6.1.2 Camada de Lógica de Negócios e Serviços (*Backend*)

Ela é API do Sistema nesta camada, a solução adota foi um modelo *Backend-as-a-Service (BaaS)*, tendo o *Firebase* como pilar central. As funcionalidades aqui são expostas como uma API (*Application Programming Interface*), que permite a comunicação e a execução de lógica de negócios de forma segura e escalável, sem a necessidade de gerenciar infraestrutura de servidores.

- *Firebase Authentication*: Gerencia o registro, login e estado de autenticação dos usuários, atuando como o serviço de identidade da API.
- *Cloud Firestore*: Banco de dados NoSQL baseado em documentos, utilizado para armazenar os dados dos usuários, seus dependentes e o histórico de vacinas. A interação direta do *frontend* com o *Firestore* é mediada por regras de segurança que atuam como parte da API de dados.
- *Firebase Cloud Functions e Extensions* (para e-mails): Este é o cerne da API personalizada do sistema. As *Cloud Functions* são responsáveis por orquestrar a lógica de negócio complexa e sensível, incluindo:

Trigger do Firestore: Funções acionadas automaticamente, como a *checkVaccinesForChildren*, que gerencia o ciclo de vida das vacinas, populando automaticamente a lista de vacinas de uma criança ao seu cadastro, com base em sua idade.

Funções Agendadas: Como a *sendVacinaNotifications*, que executa rotinas periódicas para enviar notificações push lembrando sobre vacinas pendentes.

Funções Chamáveis (*Callable Functions*): função de e-mail, como a `sendEmailNotification` função acionada diretamente pelo frontend via requisições seguras, utilizadas para operações que requerem lógica de servidor e proteção de credenciais, como o envio de e-mails de notificação ou confirmação.

- Serviço Externo de Inteligência Artificial: O sistema se comunica com um serviço externo de inteligência artificial, por meio da LLM API (como a Gemini API), que atua como o núcleo do assistente virtual. As requisições a esse serviço são intermediadas por uma *Cloud Function*, garantindo segurança e encapsulamento da chave da API.

6.2 Componentes e Módulos Principais

O aplicativo é modularizado em serviços Angular, ele adota a separação de preocupações, com serviços dedicados à lógica de negócios, manipulação de dados e integração com APIs. Os componentes focam na interface e interação com o usuário, consumindo esses serviços. Essa abordagem garante modularidade, permitindo que os serviços e componentes sejam organizados e carregados sob demanda, e promove a reutilização de código. Além disso, a clara distinção de responsabilidades otimiza a testabilidade da aplicação, tornando-a mais fácil de manter e expandir. A seguir o exemplo prático cada um responsável por uma parte específica da lógica de negócios:

AuthService: Gerencia a autenticação de usuários com o *Firebase Auth*, incluindo registo, login e logout.

ChildrenService: Lida com as operações CRUD (*Create, Read, Update, and Delete*, ou Criar, Ler, Atualizar e Deletar) para os dados dos dependentes no *Cloud Firestore*, organizando-os sob o perfil de cada usuário.

VacinaService: Gerencia os dados das vacinas e a lógica de alertas. Interage com o *Firestore* para guardar o histórico de vacinas e com o *EmailService* para enviar notificações.

LLMAssistantService: Responsável por enviar as perguntas dos usuários para a API do LLM e processar as respostas.

Componentes de Páginas: (*LoginPage, HomePage, ProfilePage, etc.*) que interagem com os serviços para exibir a interface e capturar as interações do usuário.

6.3 Implementação das Principais Funcionalidades

Este tópico apresenta, de forma prática e descritiva, as principais funcionalidades implementadas no aplicativo “Dia de Vacina Brasil.” O objetivo é demonstrar como cada recurso foi desenvolvido para atender às necessidades dos usuários, especialmente no que se refere ao controle do cronograma vacinal de crianças.

6.3.1 Cadastro e Login de Usuários

- Permite o acesso seguro e individualizado dos pais/responsáveis.
- Suporta login por e-mail e senha, com persistência de sessão, como mostra figura 6.

Figura 6: Tela de cadastro.



Fonte: Autoria própria, 2025.

6.3.2 Cadastro de Dependentes Via Tela de Perfil

- Cada usuário pode cadastrar um ou mais dependentes.
- Armazena dados como nome, CPF, data de nascimento, gênero e outros dados relevantes a criança, conforme figura 7.

Figura 7: Tela de cadastro de dependente.

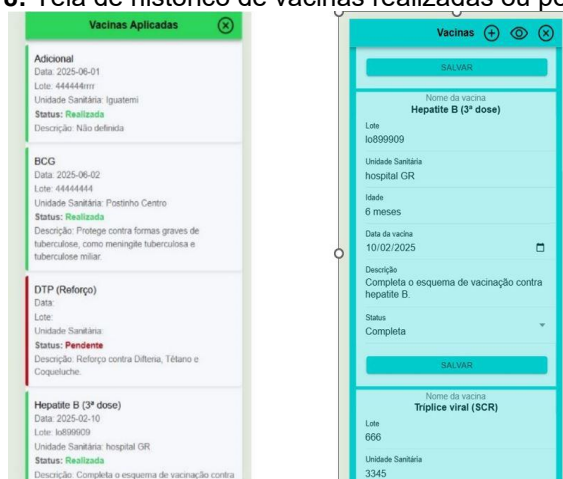


Fonte: Autoria própria, 2025.

6.3.3 Gerenciamento de Vacinas e Histórico Vacinal

- Permite registrar vacinas aplicadas, datas, lote, local.
- Organiza o cronograma vacinal de acordo com a idade de dependente ou nome de vacina, mostra figura 8.

Figura 8: Tela de histórico de vacinas realizadas ou pendentes.



Fonte: Autoria própria, 2025.

6.3.4 Alertas e Notificações de Vacinação (via e-mail e/ou push)

- Envia lembretes automáticos sobre vacinas próximas.
- Utiliza *Firestore Cloud Messaging* (FCM) para notificações *push* e *Firestore Cloud Functions* com *Trigger Email Extension* para envio automatizado de e-mails, visto na figura 9.

Figura 9: Tela Controle.



Fonte: Autoria própria, 2025.

Esses alertas e notificações tem a função de lembrar o usuário do aplicativo a fazer as vacinas do calendário em seus dependentes, o que pode contribuir para a melhoria nas taxas de adesão a imunização infantil no Brasil.

6.3.5 Confirmação de Vacina Aplicada

- O usuário pode confirmar que a vacina foi aplicada.
- Após a confirmação, o sistema salva no banco de dados e suspende os alertas daquela dose, como demonstra a figura 10.

Figura 10: Tela onde é feita a edição das vacinas.

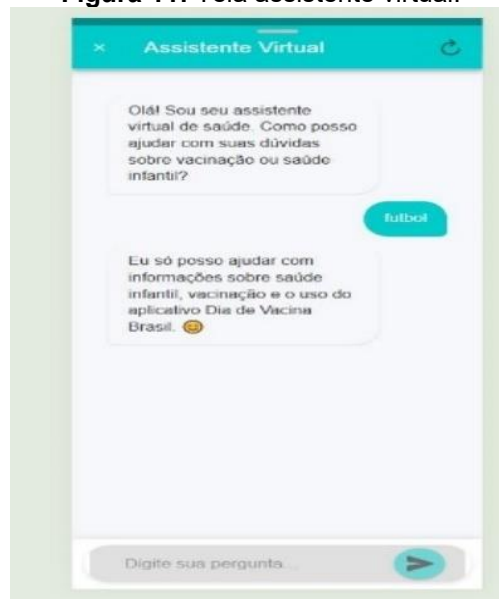


Fonte: Autoria própria, 2025.

6.3.6 Assistente Virtual Inteligente

- Responde perguntas sobre vacinas, sintomas e cronogramas.
- Utiliza linguagem natural para facilitar a interação dos pais. Como mostra a figura 11.

Figura 11: Tela assistente virtual.



Fonte: Autoria própria, 2025.

6.4 Gestão de Dados de Dependentes

O “*ChildrenService*” é responsável por interagir com o *Cloud Firestore* para criar, ler, editar, apagar e recuperar os dados dos dependentes. A estrutura de dados utiliza subcoleções para associar os “dependentes” a “usuários” específicos (`users/{userId}/children/{childId}`), a figura 12 demonstra passos e métodos do CRUD e reflexo nas subcoleções.

Figura 12: Trecho do código adiciona dependente.

```
export class ChildrenService {  
  constructor(  
    private firestore: Firestore  
  ){  
  }  
  
  addChild(child: Child) {  
    const childrenRef = collection(this.firestore, 'users', child.parentId, 'children');  
    return addDoc(childrenRef, child);  
  }  
  
  getChildren(userId: string): Observable<Child[]> {  
    const childrenRef = collection(this.firestore, 'users', userId, 'children');  
    return collectionData(childrenRef, { idField: 'id' }) as Observable<Child[]>;  
  }  
  
  updateChild(userId: string, childId: string, data: Partial<Child>) {  
    const childRef = doc(this.firestore, 'users', userId, 'children', childId);  
    return updateDoc(childRef, data);  
  }  
  
  async deleteChild(userId: string, childId: string) {  
    const vacinasRef = collection(this.firestore, 'users', userId, 'children', childId);  
    const vacinasSnapshot = await getDocs(vacinasRef);  
  }  
}
```

Fonte: Autoria própria, 2025.

6.5 Integração com Assistente Virtual (LLM API)

A página *llm-assistent.service* será o ponto de comunicação com a API do LLM. Ele enviará as perguntas do usuário e receberá as respostas, que serão exibidas na interface do chat do assistente. A figura 13 mostra como enviar uma pergunta (*userMessage*) para uma API LLM e receber uma resposta. Essencialmente, ele cria uma requisição HTTP POST para o *endpoint* da API, incluindo a sua pergunta e as configurações desejadas (como o formato da resposta). A API processa a pergunta e retorna uma resposta, que geralmente é um texto gerado pelo modelo de linguagem.

Figura 13: Trecho de *llm-assistent.service* que cria uma requisição HTTP POST para o “*endpoint*” da API (envia a pergunta).

```
// A URL completa já usa a apiKey definida acima
private apiUrl = `https://generativelanguage.googleapis.com/v1beta/models/gemini-2.0-flash:genera

private conversationHistory: { role: string, parts: { text: string }[] }[] = [];

constructor(private http: HttpClient) {
  this.initializeConversationHistory();
}

// Novo método para configurar as instruções iniciais e o conhecimento do app
private initializeConversationHistory() { ...
}

async askQuestion(userMessage: string): Promise<string> {
  if (!this.apiKey) {
    console.warn('API KEY não configurada.');
```

return 'API Key não configurada. Por favor, adicione sua chave para ativar o assistente.';
}

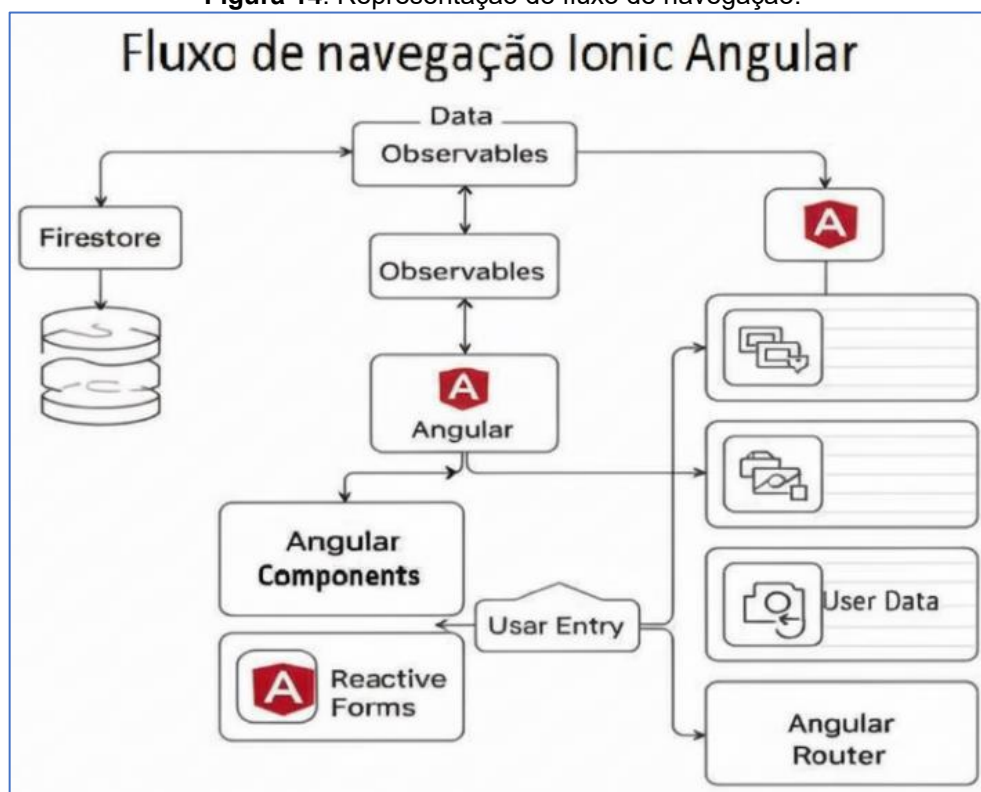
// Adiciona a mensagem do usuário ao histórico da conversa
this.conversationHistory.push({ role: 'user', parts: [{ text: userMessage }] });

const headers = new HttpHeaders({

6.6 Interface do Usuário e Fluxo de Navegação (*Ionic/Angular*).

As páginas do aplicativo são construídas com componentes *Angular*, utilizando formulários reativos na entrada de dados e *Observables* para exibir dados do *Firestore* em tempo real, o roteamento é gerido pelo *Angular Router*. Esse fluxo é representado na figura14:

Figura 14: Representação do fluxo de navegação.



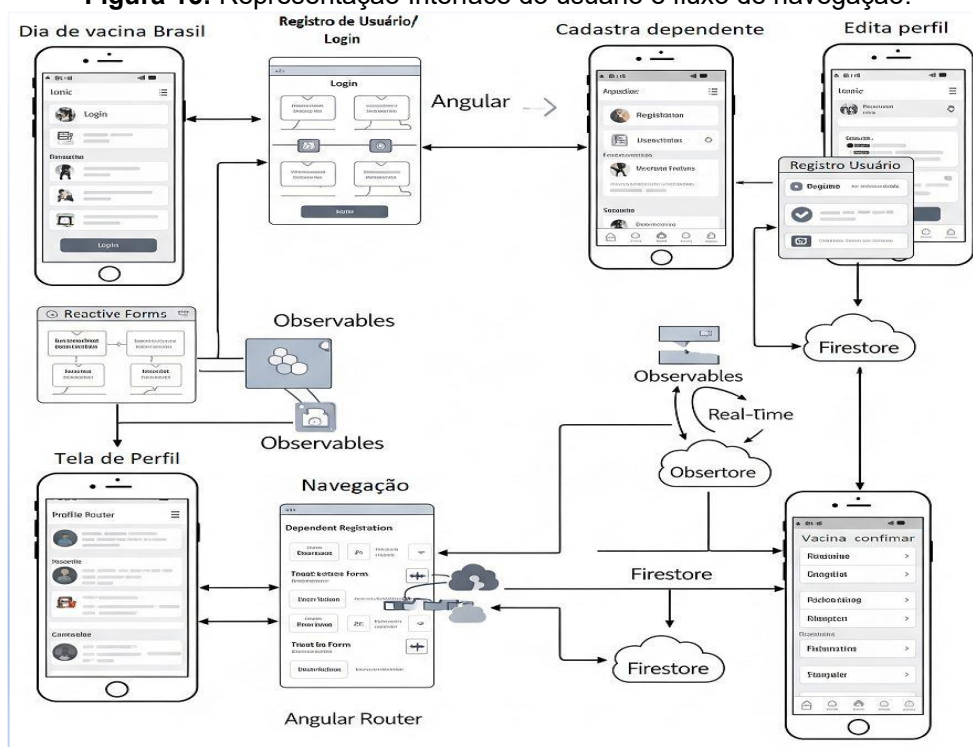
Fonte: Autoria própria, 2025.

Aprofundando o entendimento do fluxo a arquitetura do aplicativo é baseada em componentes reutilizáveis que estruturam tanto a interface quanto a lógica de navegação. Essa abordagem permite maior modularidade e facilita a manutenção do código.

A estrutura dos formulários reativos é aplicada em telas como *Login*, Cadastro de Usuário, Cadastro de Dependente e Edição de Perfil, permitindo capturar os dados de forma dinâmica. Essas informações são armazenadas no *Firestore*, banco de dados em nuvem que opera em tempo real. Com o uso de *Observables*, o aplicativo se mantém sincronizado com o banco de dados, refletindo automaticamente as atualizações.

Para demonstrar o processo de forma prática, foi ilustrada a figura 15 que foca na interface do usuário e a navegação entre as telas. Esta controlada pelo *Angular Router*, o que proporciona uma experiência fluida ao usuário pois atua como um "controlador de tráfego" para as *views* da aplicação, permitindo que o usuário se mova de uma tela para outra de forma ininterrupta e controlada. Em vez de recarregar a página inteira, o *Angular Router* apenas troca os componentes que são visíveis na tela, proporcionando uma experiência de usuário mais rápida e sem interrupções.

Figura 15: Representação Interface do usuário e fluxo de navegação.



Fonte: Autoria própria, 2025.

Como trabalhos futuros, planeja-se:

- Melhorar a interface e a experiência do usuário (UX/UI).
- Adicionar novas funcionalidades, como a leitura de *QR Code* para registro de vacinas.
- Realizar a integração com bases de dados oficiais do Ministério da Saúde para calendários vacinais sempre atualizados.
- Explorar o envio de lembretes por outros meios, como *WhatsApp*.
- Implementar funcionalidades para verificar incompatibilidade de doses e prazos, conforme requisitos de sistema.
- Adicionar a possibilidade de armazenar histórico de uso de outros medicamentos.
- Expandir as capacidades do assistente virtual com a *Google Gemini API* para oferecer informações ainda mais detalhadas e personalizadas sobre saúde infantil.
- Exportação de dados para o aplicativo do dependente uma funcionalidade essencial será a possibilidade de transferir o histórico vacinal do dependente para um eventual aplicativo próprio do indivíduo (quando ele atingir a idade apropriada para gerenciar sua própria saúde), garantindo a continuidade do registro e o empoderamento do cidadão com seu histórico de saúde.

7 CONCLUSÃO

O desenvolvimento do aplicativo Dia de Vacina Brasil representou uma experiência prática essencial na aplicação dos conhecimentos adquiridos ao longo do curso de Análise e Desenvolvimento de Sistemas. O projeto teve como foco principal auxiliar pais e tutores no acompanhamento do calendário de vacinação infantil, oferecendo uma solução simples, acessível e funcional para um problema real: as falhas quanto às datas de imunização.

O Dia de vacina Brasil vai de encontro aos princípios do ECA que tem como uma de suas prioridades assegurar o direito a saúde, sendo esse, dever da sociedade em geral. Nesse sentido o aplicativo torna-se uma ferramenta importante capaz de notificar o usuário sobre as vacinas a serem aplicadas conforme o calendário, além de possibilitar ao usuário ter um controle sobre as vacinas, contribuindo para uma possível melhoria na taxa de imunização infantil e prevenção de doenças.

Ao longo do processo, foi possível integrar diversas tecnologias modernas como *Ionic Framework* e *Angular* para a criação da interface multiplataforma. O *Firebase* como banco de dados em nuvem e ferramenta de autenticação, junto de recursos de notificações *push* e envio automático de e-mails garantiu uma comunicação eficaz com os usuários. Essas escolhas permitiram que o sistema fosse construído com rapidez, segurança e escalabilidade, mesmo com recursos limitados.

O projeto também demonstrou a importância da fase de modelagem, prototipação e planejamento com ferramentas como *Balsamiq* e *Enterprise Architect*, que auxiliaram na organização das funcionalidades e na clareza dos objetivos desde o início.

Mesmo sendo uma versão inicial, o sistema já atende ao propósito de oferecer alertas e controle de vacinas de forma personalizada para cada criança cadastrada. Portanto o Dia de Vacina Brasil é um projeto com grande potencial de impacto social e pode evoluir para se tornar uma ferramenta completa no apoio à saúde preventiva infantil. O trabalho reforça o valor do desenvolvimento de soluções tecnológicas focadas em problemas reais e no bem-estar da população.

REFERÊNCIAS

ANGULAR. **Angular**. [S. l.]: Google, 2025. Disponível em: <https://angular.io/>. Acesso em: 25 maio 2025.

BALSAMIQ. **Quick and Easy Wireframing Tool**. [S. l.]: Balsamiq, 2025. Disponível em: <https://balsamiq.com/>. Acesso em: 5 dez. 2019.

BRASIL. **Lei nº 8.069, de 13 de julho de 1990. Dispõe sobre o Estatuto da Criança e do Adolescente e dá outras providências**. Diário Oficial da União, Brasília, DF, 16 jul. 1990.

BRASIL. Ministério da Saúde. **Calendário Nacional de Vacinação**. [S. l.]: Ministério da Saúde, [2025]. Disponível em: <https://www.gov.br/saude/pt-br/vacinacao> Acesso em: 6 jun. 2025.

BRASIL. Ministério da Saúde. **Programa Nacional de Imunizações: 30 anos**. Brasília, DF: Ministério da Saúde, 2003.

CAPACITOR. **The Native Runtime for Modern Web Apps**. Disponível em: <https://capacitorjs.com>. Acesso em: 09 jun. 2025.

DOMINGUES. Carla Magda Allan Santos et al. **46 anos do Programa Nacional de Imunizações: uma história repleta de conquistas e desafios a serem superados**. Cadernos de Saúde Pública, v. 36, n. 2, p. 1–17, 3 abr. 2020. Disponível em: <https://cadernos.ensp.fiocruz.br/ojs/index.php/csp/article/view/7531/16727>. Acesso em: 1 mar. 2025.

FIREBASE. **Firestore**. [S. l.]: Google, 2025. Disponível em: <https://firebase.google.com/>. Acesso em: 29 maio 2024.

GOOGLE. **Google AI: Build with our generative AI models**. [S. l.]: Google, 2025. Disponível em: <https://ai.google.dev/>. Acesso em: 25 maio 2025.

IBM. **O que é arquitetura de três camadas?** [S. l.]: IBM, 2025. Disponível em: <https://www.ibm.com/br-pt/topics/three-tier-architecture>. Acesso em: 23 maio 2025.

IONIC. **Ionic Framework**. [S. l.]: Ionic, [s.d.]. Disponível em: <https://ionicframework.com/>. Acesso em: 25 maio 2025.

LA PORTA, Maria Luiza; LIMA, Everton. **Vacinação infantil sofre queda brusca no Brasil**. Fiocruz, [S. l.], 19 out. 2022 <https://fiocruz.br/noticia/2022/10/vacinacao-infantil-sofre-queda-brusca-no-brasil>. Acesso em: 25 fev. 2025.

MDN WEB DOCS. **JavaScript**. [S. l.]: Mozilla, 2025. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 16 jan. 2025.

REIS, Elisa Meirelles. **3 em cada 10 crianças no Brasil não receberam vacinas que salvam vidas, alerta UNICEF**. [S. l.]: UNICEF, 2022. Disponível em:

<https://www.unicef.org/brazil/comunicados-de-imprensa/3-em-cada-10-criancas-no-brasil-nao-receberam-vacinas-que-salvam-vidas>. Acesso em: 22 set. 2023.

SASS. **Sass: Syntactically Awesome Style Sheets**. [S. l.]: [s.n.], [s.d.]. Disponível em: <https://sass-lang.com/>. Acesso em: 25 maio 2025.

SILVA, Milena Vieira e; TEIXEIRA, Fabiana Cristina. **A parentalidade no desenvolvimento da criança contemporânea**. Revista Ciência Dinâmica, v. 15, p. 1–24, 2024. Disponível em: <https://revista.faculdadedinamica.com.br/index.php/cienciadinamica/article/view/224>. Acesso em: 1 mar. 2025.

SOMMERVILLE, Ian. **Engenharia de software**. 10. ed. São Paulo, SP, Pearson Education Brazil, Inc., 2018. 49-104 p. Disponível em: <<https://archive.org/details/sommerville-engenharia-de-software-10e>>. Acesso em: 03 jan. 2025.

SPARXSYSTEMS. **Enterprise Architect**. [S. l.]: Sparx Systems, 2019. Disponível em: <https://sparxsystems.com/products/ea/>. Acesso em: 29 jun. 2025.

TASCHNER. Natalia Pasternak; ALMEIDA, Paulo Vitor Gomes. **Vacinas como direitos humanos**. Revista da Defensoria Pública da União, v. 19, n. 19, p. 21–34, 23 jun. 2023.

TYPESCRIPTLANG. **TypeScript**. [S. l.]: Microsoft, 2024. Disponível em: <https://www.typescriptlang.org/pt/docs/handbook/typescript-in-5-minutes.html>. Acesso em: 8 abr. 2024.

UNICEF. **1,6 milhão de crianças não receberam nenhuma vacina DTP ao longo de três anos no Brasil, alerta UNICEF**. UNICEF Brasil, 20 abr. 2023. Disponível em: <https://www.unicef.org/brazil/comunicados-de-imprensa/1-virgula-6-milhao-de-criancas-nao-receberam-nenhuma-vacina-dtp-ao-longo-de-tres-anos-no-brasil>. Acesso em: 20 maio 2025.

W3SCHOOLS. **CSS Tutorial**. [S. l.]: W3Schools.com, 2019. Disponível em: <https://www.w3schools.com/css/>. Acesso em: 29 jun. 2025.

W3SCHOOLS. **HTML Tutorial**. [S. l.]: W3Schools.com, 2019. Disponível em: <https://www.w3schools.com/html/>. Acesso em: 29 jun. 2025.