

Sistema Web para Geração e Análise de Gráficos de Crescimento Infantil

Andriel Maicom Friedrich
Curso de Sistemas de Informação
Universidade Franciscana
Santa Maria, Brasil
andriel.friedrich@ufn.edu.br

Alessandro Andre Mainardi de Oliveira
Curso de Sistemas de Informação
Universidade Franciscana
Santa Maria, Brasil
alessandroandre@ufn.edu.br

Resumo—Este trabalho apresenta o desenvolvimento de um software web que auxilie na análise de crescimento infantil e possível detecção de doenças com base em informações fornecidas pela Organização Mundial da Saúde e da Sociedade Brasileira de Pediatria. Sua codificação será realizada por meio das tecnologias .NET e React, Banco de dados SQL Server para armazenamento dos dados e como metodologia de desenvolvimento são utilizadas boas práticas do framework ágil FDD.

Index Terms—Desenvolvimento Web, Crescimento Infantil, React, C Sharp, ASP.NET

I. INTRODUÇÃO

A monitorização do crescimento infantil é uma prática importante para a detecção precoce de problemas de saúde e nutrição em crianças. Para facilitar essa monitorização, muitas instituições utilizam gráficos de crescimento infantil, que permitem acompanhar o desenvolvimento físico de uma criança ao longo do tempo e identificar possíveis desvios em relação às curvas de crescimento esperadas.

No entanto, o uso de gráficos de crescimento infantil ainda é pouco difundido em algumas regiões do mundo, seja por falta de acesso a informações atualizadas ou por dificuldades em interpretar os dados. Nesse contexto, a utilização de tecnologias digitais pode ser uma alternativa para facilitar o acesso e a compreensão das informações sobre o crescimento infantil [1].

Nesse sentido, este trabalho realiza o desenvolvimento de um Sistema Web para Geração e Análise de Gráficos de Crescimento Infantil. Foi desenvolvido uma plataforma que permita a geração de gráficos de crescimento infantil a partir de dados fornecidos pela Sociedade Brasileira de Pediatria e da Organização Mundial da Saúde e que possibilite a análise

desses dados de forma intuitiva e acessível para profissionais de saúde e pais ou responsáveis pelas crianças [1].

Para o desenvolvimento do sistema, foram utilizadas tecnologias, como o SQL Server para o gerenciamento de banco de dados, o ASP.Net Core no backend e o React no front-end. Foi desenvolvido uma plataforma robusta e escalável, que pode ser facilmente integrada aos sistemas já existentes nas unidades de saúde.

Possibilitando a monitorização do crescimento de crianças de forma mais eficiente e acessível, contribuindo para a detecção precoce de possíveis problemas de saúde e nutrição e, conseqüentemente, para a redução da mortalidade infantil e para a melhoria da qualidade de vida das crianças.

A. Justificativa

O desenvolvimento do presente trabalho é de significativa importância pois irá ajudar a contribuir no trabalho do profissional quanto ao gerenciamento de todo o ciclo de controle do crescimento infantil.

O controle do crescimento em crianças e adolescentes é um indicador valioso da saúde geral e do desenvolvimento físico. Identificar desvios no padrão de crescimento pode ser o primeiro sinal de diversas condições médicas, incluindo distúrbios endócrinos, deficiências nutricionais, ou mesmo condições genéticas. Dessa forma, o acompanhamento contínuo se torna uma ferramenta indispensável para a detecção precoce e a intervenção adequada, contribuindo para a promoção do bem-estar infantil. [2].

Esse acompanhamento é necessário para que o pediatra veja se o paciente está com o crescimento normal. Caso contrário, o médico deve investigar para descobrir problema.

Portanto, o desenvolvimento deste trabalho não apenas alinha-se com a busca contínua por avanços na área da saúde infantil, mas também representa um passo significativo em direção à otimização dos recursos disponíveis, contribuindo para a qualidade e eficácia do atendimento pediátrico.

B. *Objetivo Geral*

Desenvolver um sistema web para geração de gráficos de crescimento infantil, que permita aos usuários inserir dados de crescimento, gerar gráficos personalizados e acompanhar o desenvolvimento da criança ao longo do tempo.

C. *Objetivos Específicos*

- Realizar um levantamento bibliográfico para identificar os principais conceitos relacionados ao monitoramento do crescimento infantil e às tecnologias utilizadas no desenvolvimento de sistemas web;
- Definir as especificações técnicas do sistema, incluindo as funcionalidades, tecnologias e requisitos de segurança;
- Definir as especificações técnicas do sistema, incluindo as funcionalidades, tecnologias e requisitos de segurança;
- Desenvolver o backend do sistema, utilizando o framework ASP.NET Core e o banco de dados SQL Server;
- Desenvolver o frontend do sistema, utilizando a biblioteca React e as melhores práticas de design de interface do usuário;

II. REFERENCIAL TEÓRICO

O monitoramento do crescimento infantil é um aspecto importante da saúde infantil e pode ajudar a identificar problemas de saúde precocemente. Um sistema web para geração de gráficos de crescimento infantil pode ser uma ferramenta útil para os pais e profissionais de saúde, permitindo que eles monitorem e analisem o crescimento da criança ao longo do tempo. Neste referencial teórico, serão abordados conceitos importantes relacionados ao crescimento infantil, sistemas de informação e gráficos, bem como o uso dessas ferramentas para o monitoramento do crescimento infantil.

A. *Crescimento Infantil*

O crescimento normal da criança é o maior sinal de que está tudo bem com sua saúde. Desde a vida intrauterina, a criança deve ter seu crescimento monitorado. Nessa fase, isso é feito pela ultrassonografia obstétrica. Após o nascimento, altura e peso devem ser aferidos até os 18 anos, colocando os resultados nas curvas de crescimento para avaliar se o

padrão de crescimento da criança está compatível com outras de mesmo sexo e idade e verificar se o crescimento está de acordo com o padrão de estatura da família [2].

O crescimento infantil é um processo complexo e multifatorial que envolve uma série de fatores, incluindo genética, ambiente, nutrição e saúde. O crescimento pode ser medido por meio de indicadores como peso, altura, perímetro cefálico e índice de massa corporal (IMC). O monitoramento do crescimento infantil é importante porque permite a identificação precoce de problemas de saúde e a avaliação do desenvolvimento infantil [1].

Podendo assim detectar problemas como a identificação de desnutrição em estágios iniciais permite intervenções adequadas para evitar complicações no futuro. Também a questão do acompanhamento do peso e do IMC auxilia na identificação precoce de obesidade, que pode estar ligado a problemas de saúde, como diabetes tipo 2 e doenças cardiovasculares [1].

Além disso, o monitoramento da criança pode revelar atrasos no desenvolvimento físico e motor, como problemas endócrinos, ósseos ou musculares. Outro aspecto importante é a detecção de deficiências de vitaminas, permitindo a intervenção adequada, seja por meio de suplementação nutricional ou alterações na dieta [1].

B. *Gráficos de Crescimento Infantil*

Eles são ferramentas úteis para a avaliação do crescimento infantil ao longo do tempo. Esses gráficos podem ser usados para comparar o crescimento da criança com os padrões de crescimento esperados para a idade e o sexo da criança. Os gráficos também podem ser usados para identificar tendências de crescimento, como aumento ou diminuição no crescimento em relação ao padrão esperado.

Os gráficos de crescimento infantil podem ser gerados usando diferentes métodos, como o método de percentil e o método de desvio padrão. O método de percentil é baseado em um ranking do crescimento da criança em relação a outras crianças da mesma idade e sexo, a fórmula geral para calcular o percentil em crianças é a seguinte: Percentual de Indivíduos Abaixo da Medida =

$$\left(\frac{\text{Número de Indivíduos Abaixo da Medida}}{\text{Total de Indivíduos na Amostra}} \right) \times 100 \quad (1)$$

, onde o "Número de indivíduos abaixo da medida" é a quantidade de crianças na amostra de referência que possuem uma medida menor do que a medida da criança em questão e o

”Total de indivíduos na amostra” é o número total de crianças na amostra de referência [1].

O método de desvio padrão, por outro lado, é baseado na comparação do crescimento da criança com um padrão de crescimento médio, com base em uma amostra de crianças da mesma idade e sexo. Para calcular o desvio padrão em relação a uma distribuição de referência, como as curvas de crescimento da Organização Mundial da Saúde (OMS), é necessário seguir os passos [1]:

1. Coletar um conjunto de dados de referência que represente a população de interesse (por exemplo, uma amostra de crianças saudáveis de determinada faixa etária).
2. Calcular a média e o desvio padrão dos dados de referência para a medida específica (por exemplo, peso para idade).
3. Calcular a diferença entre a medida observada e a média de referência.
4. Dividir a diferença pelo desvio padrão para obter o desvio padrão em unidades de desvio padrão.

C. ASP.NET Core

ASP.NET Core é um framework open source desenvolvido pela Microsoft para a criação de aplicativos web modernos, escaláveis e de alto desempenho. Ele é uma evolução do ASP.NET, mas com uma arquitetura mais modular e flexível, que permite a criação de aplicativos mais leves e rápidos [3].

O ASP.NET Core pode ser executado em diferentes plataformas, como Windows, Linux e macOS, o que o torna uma ótima opção para desenvolvedores que desejam criar aplicativos para várias plataformas. Ele também é compatível com diferentes tecnologias e linguagens, como Csharp, Visual Basic, Fsharp e outras linguagens baseadas em .NET.

O ASP.NET Core também possui um modelo de desenvolvimento baseado em middleware, que permite a criação de aplicativos web mais flexíveis e personalizáveis. Isso significa que os desenvolvedores podem criar aplicativos com recursos específicos, como autenticação, cache e log, sem precisar incluir todo o código relacionado a esses recursos no aplicativo [3].

D. Banco de dados SQL Server

O SQL Server é um sistema de gerenciamento de banco de dados relacional (RDBMS) desenvolvido pela Microsoft. Ele é um dos sistemas de gerenciamento de banco de dados mais populares do mercado e é amplamente utilizado em empresas de diversos setores [8].

Também conhecido por sua confiabilidade, segurança e escalabilidade. Ele é capaz de lidar com grandes volumes de dados e oferece recursos avançados de segurança, como criptografia de dados, autenticação integrada do Windows e auditoria de segurança [9].

E. React.js

React é uma biblioteca JavaScript de código aberto para a criação de interfaces de usuário (UI) em aplicativos web. Desenvolvida pelo Facebook, o React se tornou uma das bibliotecas mais populares para criação de interfaces de usuário, especialmente em aplicações single-page (SPA) [4].

O React utiliza uma abordagem baseada em componentes, que permite aos desenvolvedores criar interfaces de usuário complexas e interativas a partir de pequenos componentes reutilizáveis. Isso significa que os desenvolvedores podem criar componentes independentes que possam ser reutilizados em diferentes partes do aplicativo, o que reduz o tempo e o esforço de desenvolvimento.

Outra vantagem do React é o fato de ser uma biblioteca ”declarativa”. Isso significa que os desenvolvedores podem descrever como uma interface de usuário deve ser exibida em diferentes estados e deixar o React cuidar da atualização da interface do usuário. Isso permite que os desenvolvedores se concentrem mais na lógica do aplicativo e menos na manipulação do DOM que é uma representação estruturada de um documento HTML (ou XML) que o navegador utiliza para organizar e manipular os elementos desse documento. [4].

React faz com que a criação de UIs interativas seja uma tarefa fácil. Crie views simples para cada estado na sua aplicação, e o React irá atualizar e renderizar de forma eficiente apenas os componentes necessários na medida em que os dados mudam [4].

F. Typescript

TypeScript adiciona tipagem ao JavaScript e usa inferência de tipo para fornecer ótimas ferramentas sem código adicional para suportar uma melhor integração com o editor. O código TypeScript é convertido em JavaScript, que é executado em qualquer lugar em que o JavaScript é executado [5].

G. shadcn/ui

Componentes lindamente projetados para poder copiar e colar em seus aplicativos. Acessível. Customizável. Código aberto [6].

H. Next.js

Usado por algumas das maiores empresas do mundo, Next.js permite que você crie aplicativos Web full-stack, estendendo os recursos mais recentes do React e integrando poderosas ferramentas JavaScript baseadas em Rust para compilações mais rápidas [7].

III. TRABALHOS RELACIONADOS

A. Análise e manutenção do Sistema Gerador de Gráficos de Crescimento Infantil

O acompanhamento do crescimento de uma criança é uma das maneiras mais eficientes para se poder formar adultos saudáveis. Foi desenvolvido um sistema capaz de gerar automaticamente este gráfico de Crescimento para auxiliar nos diagnósticos de pediatras, contribuindo para que possam detectar precocemente futuros problemas que uma criança possa apresentar e tratá-los adequadamente [10].

Foi desenvolvido um Sistema Gerador de Gráficos de Crescimento, onde é possível gerir pacientes, médicos, clínicas e consultas podendo assim analisar gráficos de crescimento levando em conta idade e sexo do paciente, usando as tecnologias Java Server Pages, Mysql, MVC e Struts, Hibernate, XDOCLET.

B. Sistema Gerador de Gráficos de Crescimento

Um dos acompanhamentos que o pediatra faz em crianças e adolescentes é com relação ao crescimento do paciente. Esse acompanhamento é necessário para que o pediatra veja se o paciente está com o crescimento normal. Caso contrário, o médico deve investigar e descobrir o problema. Existe a necessidade do médico pediatra ter acesso as informações fora da clínica. Se as fichas forem apenas no formato físico, acaba dificultando o trabalho do pediatra na obtenção das informações. As tecnologias utilizadas no trabalho foram Java Server Pages, Java 2D, JAVAKRTA STRUTS, Modelo MVC e Struts, Hibernate, XDOCLET e o banco de dados MYSQL [11].

C. Software-protótipo para o acompanhamento do crescimento e desenvolvimento infantil a partir da classificação internacional das práticas de enfermagem (CIPE®)

Este estudo teve como objetivo desenvolver um software (protótipo) para consulta de enfermagem aplicada ao acompanhamento do crescimento e desenvolvimento infantil na

Atenção Primária à Saúde. Essa pesquisa emergiu das experiências profissionais na docência e na APS da doutoranda deste trabalho, em virtude das inquietudes frente à falta de instrumentos padronizados para prestar o cuidado à criança/família, dos longos registros manuscritos das consultas realizadas [12].

D. Conclusão sobre Trabalhos Relacionados

Esses trabalhos apresentam abordagens semelhantes ao desenvolvimento de sistemas web para monitoramento do crescimento infantil, incluindo a utilização de diferentes frameworks e bancos de dados. A leitura desses trabalhos forneceu insights valiosos como orientação da pesquisa, evitar duplicação de esforços, identificar lacunas e oportunidades como a identificação de metodologias e técnicas relevantes, sobre a abordagem de desenvolvimento, escolha de tecnologias, design de interface e análise de dados de crescimento infantil, aprender com experiências anteriores e posicionar o trabalho em relação aos demais.

IV. PROPOSTA

A proposta deste trabalho é desenvolver um sistema para monitorar o crescimento da criança, para isso, propõe-se um sistema web para que o profissional da saúde possa realizar a análise do crescimento infantil e possível prevenção de doenças. A aplicação tem como objetivo proporcionar aos usuários um melhor acompanhamento de todas as etapas do crescimento. Sendo assim a Figura 1 apresenta o direcionamento das telas pensando de forma prática e simples para a facilidade do uso aos pacientes.

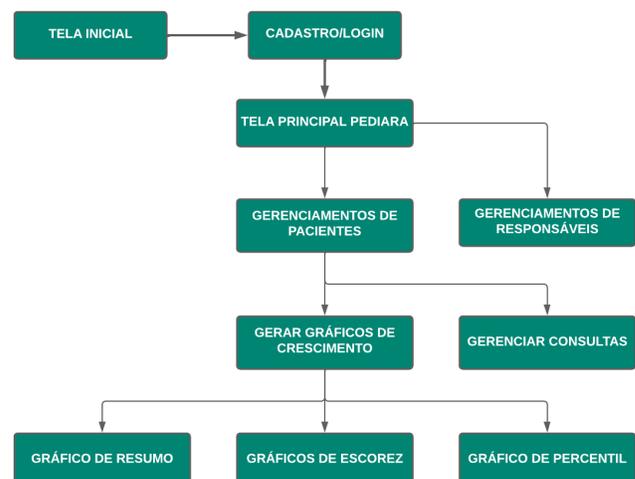


Figura 1. Direcionamento das telas para o usuário.

Por meio desse fluxograma é possível observar as opções que o profissional da saúde terá a sua disposição no sistema. Após o cadastro o usuário poderá visualizar uma breve introdução sobre o crescimento infantil se o mesmo optar. A tela principal é um menu com um dashboard com um resumo das informações mais importantes, podendo acessar a tela Carteira de Acompanhamento de Crescimento onde poderá informar perímetro cefálico, vacinações, medidas de altura, peso. Podendo também visualizar os dados de forma detalhada pelo menu de gráficos de desenvolvimento e análise de prevenção.

O usuário do sistema poderá realizar a busca de seus pacientes cadastros na aplicação e visualizar seus gráficos de desenvolvimento assim como seus registros, podendo assim fazer um acompanhamento periódico e disponibilizar orientações: Alimentação saudável, aleitamento materno caso a criança for um bebê até os 6 meses, vacinação, sono adequado, atividade física, higiene e cuidados gerais, desenvolvimento socio emocional, monitoramento e acompanhamento regular, essas são as principais orientações que um pediatra poderia fornecer, porém as orientações devem se adaptar de acordo com as necessidades individuais de cada criança [2].

V. METODOLOGIA

As Metodologias Ágeis visam minimizar os riscos relacionados as incertezas dos projetos e com os imprevistos de maneira mais natural e rápida [13]. Para o desenvolvimento deste trabalho foi escolhida a metodologia ágil FDD.

Feature-Driven Development (FDD) é uma metodologia de desenvolvimento de software que inclui alguns benefícios de processos rigorosos, como modelagem, planejamento prévio e controle do projeto, assim como contem características de processos ágeis, como foco na programação, interação constante com o cliente e entrega frequente de versão do produto. Prevê praticas apenas para o desenvolvimento de software em si não se preocupando com outros fatores como a escolha de tecnologias e ferramentas, a definição de procedimentos de aquisição, dentre outros [14].

São cinco os processos da metodologia ágil FDD: Desenvolver um Modelo Abrangente, Construir uma Lista de Funcionalidades, Planejar Através de Funcionalidades, Projetar Através de Funcionalidades e Construir Através de Funcionalidades [14].

A. Desenvolver um modelo abrangente

O processo Desenvolver um Modelo Abrangente é o responsável pelo estudo detalhado sobre o domínio do negócio e pela definição do escopo do projeto [14]. A Figura 2 representa o Diagrama de Domínio inicial do projeto, onde é possível ver as entidades conceituais do domínio e seus relacionamentos, concedendo uma visão dos elementos do sistema.

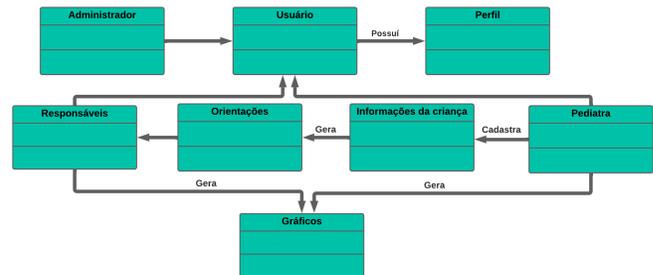


Figura 2. Diagrama de Domínio.

Seguindo a metodologia FDD, após desenvolver o diagrama de domínio, devemos realizar a elaboração de uma lista de funcionalidades.

B. Construir e planejar por funcionalidades

Neste processo da metodologia, com base no domínio da aplicação são mencionadas as funcionalidades que atendem os requisitos do sistema, classificados com funcionais, aqueles que descrevem o que o sistema fará, ou não funcionais, que retratam como o sistema fará. [14]

C. Projetar por funcionalidades

Neste processo, para cada funcionalidade da lista é definida uma atividade a ser realizada com o objetivo de produzir um pacote de arquitetura por funcionalidade, além de demonstrar o mapeamento das classes e métodos que serão empregados no desenvolvimento do sistema [14]. Na figura 3 é apresentado o Diagrama de Classes, que apresenta a visão da estrutura e das relação entre as classe dessa aplicação.

D. Construir atreves de funcionalidades

Nessa fase, as funcionalidades são construídas uma a uma de acordo com as necessidades e prioridades. Após a construção dessas funcionalidades, é possível obter um software mais completo e funcional ao longo do tempo. Além disso facilita a identificação de possíveis ajustes e melhorias durante o processo de desenvolvimento, garantindo assim que o software atenda as necessidades do usuário.

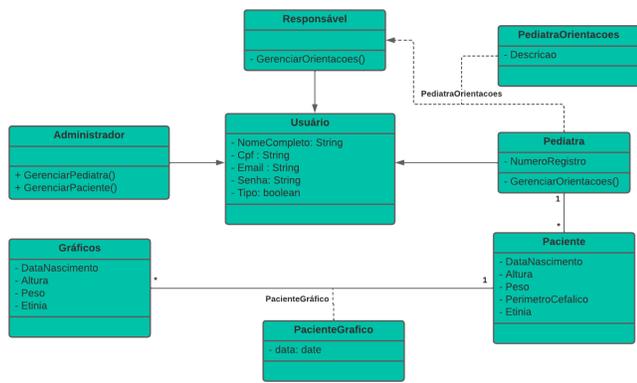


Figura 3. Diagrama de Classes

A Figura 4 ilustra a estrutura de componentes do projeto, proporcionando uma visão abrangente de todas as camadas do sistema. Assim sendo possível visualizar de forma mais clara a organização e iteração do projeto como um todo.

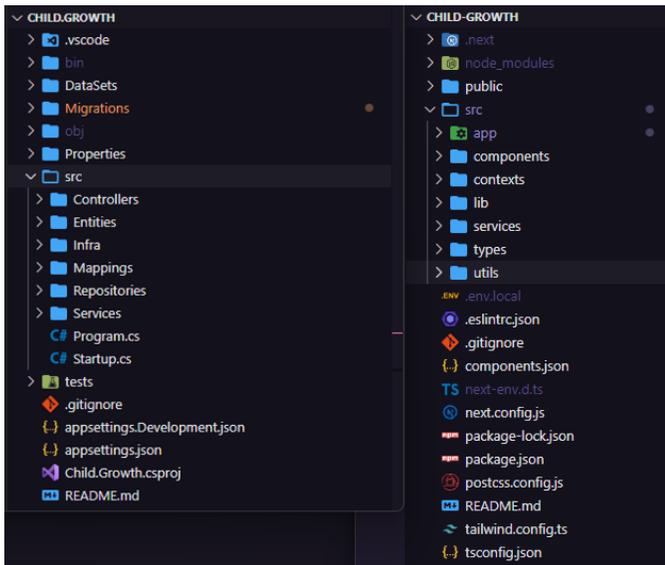


Figura 4. Estrutura do Projeto

No Backend com sua estrutura representada na esquerda da figura 4, a autenticação ao sistema foi criando usando JWT (JSON Web Token) onde foi criado um serviço que realiza a geração do Token, foi utilizado uma chave secreta, adicionando algumas configurações no Token para verificar as permissões do usuário logado, após gerado, quando o usuário é autenticado na aplicação o Token é enviado no cabeçalho da requisição e decodificando a chave secreta mencionada anteriormente.

A camada Controller é responsável por tratar as requisições HTTP recebidas, coordenando assim as ações a serem toma-

das, se comunicando diretamente com a camada Services. Para a proteção das rotas após o usuário autenticado foi feito o uso de Roles (papéis) junto com o Atributo Authorize do AspNetCore onde temos dois tipos de usuário no sistema atualmente com papel Doctor se tratando do pediatra que irá gerenciar responsáveis e pacientes, também podendo realizar análise das consultas realizadas e o papel Responsible que se trata do responsável pela criança, conforme exemplo na figura 5.

```

1 [HttpGet("GetByFilters")]
2 [Authorize(Roles = "Doctor")]
3 public List<PatientConsultation> GetByFilters(string filters)
4 {
5     var PatientConsultationByFilter = _patientConsultationService.GetByFilters(filters);
6     return PatientConsultationByFilter;
7 }
8
9
10 [HttpGet("GetConsultsByUserId")]
11 [Authorize(Roles = "Doctor,Responsible")]
12 public ResponseBody GetConsultsByUserId(Long userId)
13 {
14     return _patientConsultationService.GetConsultsByUserId(userId);
15 }
16
17 [HttpGet("GetGraphComparativeData")]
18 [Authorize(Roles = "Doctor")]
19 public List<ComparativeData> GetGraphComparativeData(Long childrenId)
20 {
21     return _patientConsultationService.GetComparativeData(childrenId);
22 }

```

Figura 5. Segurança das rotas

A camada Services é responsável por lidar com a lógica de negócios da aplicação, tendo regras e operações relacionadas ao domínio do sistema, assim utilizando o princípio de separação de responsabilidades. Onde a lógica de negócio é mantida de forma independente da interface do usuário e da camada de acesso a dados, tendo benefícios como a reutilização de código, testes, manutenção e evolução do projeto, também facilitando o gerenciamento de transações.

A pasta DataSets é utilizada para armazenar as planilhas retidas da OMS para realizar as análises de crescimento, na pasta Mappings foi isolado a injeção de dependência dos mapeamentos das entidades do banco de dados para adicionar referências entre entidades e configurações específicas.

O Entity Framework Core é responsável pela criação da estrutura do banco de dados, mapeando as entidades e seus devidos relacionamentos, tendo na pasta Migrations todas as alterações desde o início do projeto, conforme figura 6 e 7.

A camada Entities é utilizada pra representar os modelos completos das informações de cada entidade representada no projeto. A camada de Infra é utilizada para as funcionalidades que fornecem suporte e abstrações para as camadas superiores da aplicação. Desempenhando um papel fundamental na

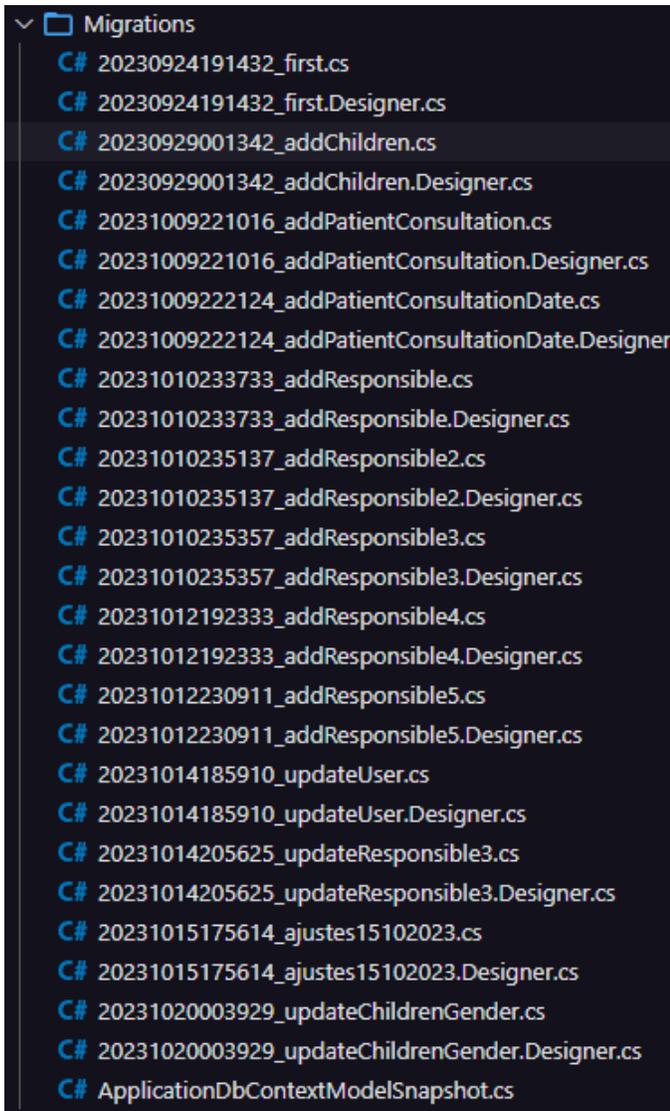


Figura 6. Migrations



Figura 7. Exemplo de Migration com Entity Framework Core

facilitação da comunicação entre a aplicação e os recursos externos, como bancos de dados, serviços web, sistemas de arquivos, entre outros.

A camada Repository (pasta Repositories) é utilizada para separar da camada de regra de negócio da aplicação, dessa forma toda a lógica relacionada com acesso a dados é encapsulada sendo assim as entidades não são afetada, tendo assim alguns benefícios na utilização do padrão Repository como evitar duplicação de código, impulsiona o uso de injeção de dependência e facilmente alterar a forma como armazenamos os dados sem afetar o projeto como um todo.

Partindo para a parte do Frontend, com sua estrutura representada no lado direito da figura 4, a pasta app é uma das mais importantes do Next.js. Contendo os arquivos responsáveis por definir as páginas da aplicação, portando cada pasta com um arquivo 'page.tsx' é automaticamente criado uma página para uma rota acessível na aplicação, facilitando assim a navegação dentro do sistema.

Já a pasta public é usada para armazenar arquivos estáticos que serão acessíveis de forma pública, como as imagens e gifs utilizados no sistema. A pasta contexts é utilizada para armazenar os contextos da aplicação usando a context api do React como o contexto de autenticação armazenando dados do usuário, Token e outros, sendo assim é possível acessar e atualizar o estado global independente da hierarquia dos componentes da aplicação.

Na pasta components são armazenados os componentes a serem reutilizados em toda a aplicação. Como componentes de Input, Cabeçalho e entre outros, sendo assim evitando a duplicação de código e agilizando o processo de desenvolvimento, em types são armazenados os tipos compartilhados por toda a aplicação com Typescript, em utils tratam se de utilitários que são utilizados pela aplicação como de formatação de data, valor monetário dentre outros.

Na pasta services se trata dos serviços que se comunicam com o backend via requisição http, criando nela a implementação dos seus métodos put, post, get, delete todos utilizando a biblioteca axios que é uma biblioteca que simplifica a comunicação via protocolo http, criando também interceptors do axios que servem como um middleware para tratar possível problemas ou respostas das requisições.

Organizando assim o código do frontend, é possível criar uma estrutura clara e de fácil manutenção, separando bem cada responsabilidade, facilitando a manutenção e reutilização de código, economizando o tempo e o esforço no desenvolvi-

mento.

VI. RESULTADOS

Neste trabalho, concretizou-se a implementação de um sistema web destinado ao acompanhamento e à geração de gráficos relacionados ao crescimento infantil.

Ao longo do desenvolvimento do projeto, a escolha da linguagem Typescript para o frontend foi deliberada, dada a sua capacidade de introduzir tipagem estática ao Javascript. Essa escolha visa facilitar a manutenção do código e mitigar possíveis erros. A tipagem estática permite a definição precisa das propriedades que um componente irá receber, bem como a especificação da função e do seu retorno. Essa abordagem contribui para a detecção precoce de erros de tipo e referência nula, simplificando correções e fornecendo suporte ao IntelliSense do editor, que automaticamente adiciona documentação ao código.

A implementação adotou o framework Next.js, que oferece uma renderização híbrida, seja no lado cliente ou no lado do servidor. Essa escolha visa agilizar a entrega inicial de conteúdo, aprimorando o desempenho da aplicação. Além disso, a biblioteca shadcn/ui foi utilizada como conjunto de componentes pré-construídos. Seguindo o conceito de copy/paste, essa biblioteca disponibiliza componentes base com código-fonte aberto, permitindo ajustes conforme necessário. A partir desses componentes, foi possível criar adaptações personalizadas, alinhadas ao contexto da aplicação.

A Figura 8 ilustra a página inicial após o login do pediatra, apresentando todos os pacientes e possibilitando a pesquisa por nome. Na barra superior, encontra-se um menu na lateral esquerda, fornecendo opções para o gerenciamento de pacientes e responsáveis. À direita, há o gerenciamento do usuário logado. As ações na tabela incluem editar, excluir, gerenciar e gerar diversos gráficos, como os das últimas consultas, o gráfico de percentil médio P50 e os gráficos Escore-z.

Sobre a tabela da Figura 8, destaca-se a presença de um botão para a adição de novos registros. Já na Figura 9, observa-se a interface para o registro de novos pacientes, demandando a inserção de informações essenciais, tais como nome, data de nascimento, gênero e responsável pela criança.

Na terceira opção destacada na Figura 10, concentra-se a gestão das consultas do paciente, possibilitando a execução de ações como edição, exclusão e inclusão de novas consultas. Para a adição, é imperativo fornecer informações cruciais, como altura, peso, perímetro cefálico, observações e a data

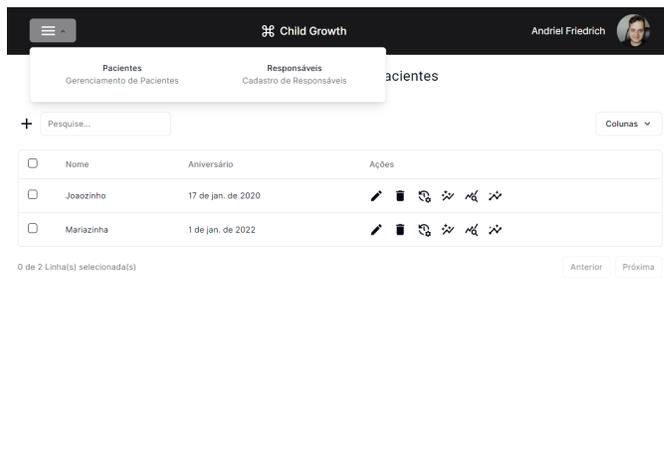


Figura 8. Tela inicial de Gerenciamento de Pacientes

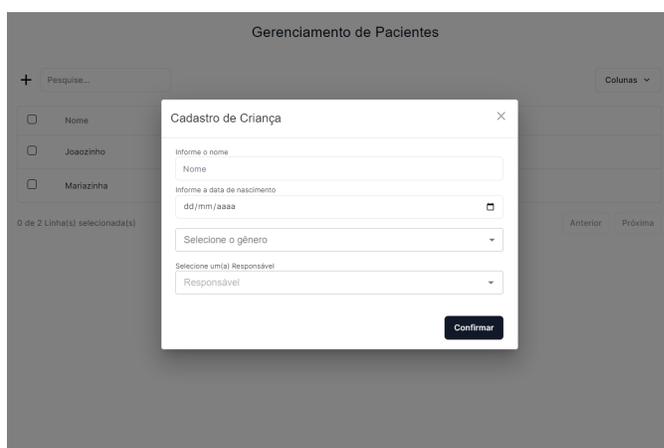


Figura 9. Tela de cadastro de novo paciente

da consulta, conforme ilustrado na Figura 10. Esses detalhes revestem-se de importância crucial no acompanhamento do paciente, viabilizando a análise por parte do pediatra e, adicionalmente, facilitando a geração dos gráficos pertinentes.

Na quarta opção apresentada na Figura 8, são exibidos os gráficos que resumem as últimas 10 consultas realizadas, conforme representado na Figura 11, onde um exemplo é apresentado. Este gráfico proporciona uma visão abrangente dos registros mais recentes, possibilitando um acompanhamento fácil da evolução dos pacientes e identificação de eventuais desvios.

Ao selecionar a quinta opção na Figura 8, acessamos os gráficos de percentil P50. O percentil 50 (P50) representa a mediana, indicando que metade da população está acima e a outra metade está abaixo desse valor. Assim, o P50 figura como uma medida central na distribuição do crescimento infantil.

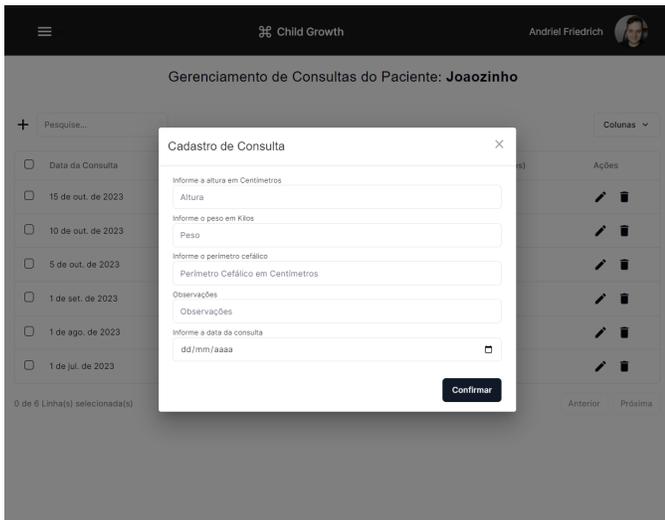


Figura 10. Tela de cadastro de consultas

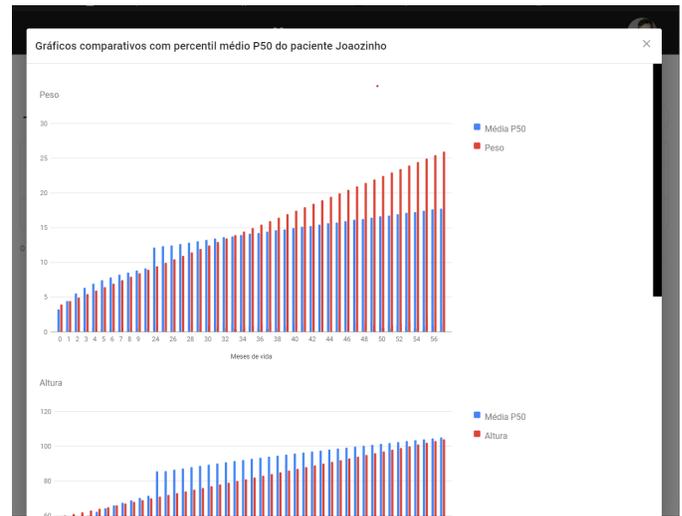


Figura 12. Gráfico comparativo com percentil P50

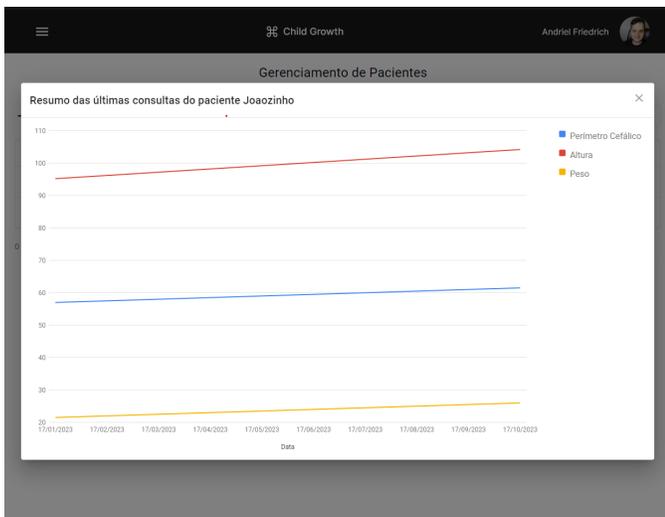


Figura 11. Resumo das últimas 10 consultas

Ao representar graficamente o crescimento infantil com a média P50, estabelecemos uma linha de base para o desenvolvimento típico da população. Em outras palavras, se uma criança está no P50, seu crescimento é considerado normal para a respectiva faixa etária. Essa comparação é evidenciada na Figura 12, que ilustra a análise de peso em relação à média P50.

Ao clicar na sexta opção na Figura 8, somos direcionados aos gráficos de Escore-z, uma medida estatística que indica quantos desvios padrão um valor específico se encontra em relação a uma média de referência.

Um Escore-z de 0 sinaliza que a medida da criança está em conformidade com a média da população de referência. Por exemplo, um Escore-z de +1 indica que a medida da

criança está um desvio padrão acima da média, enquanto um Escore-z de -1 indica que está um desvio padrão abaixo, como exemplificado na Figura 13.

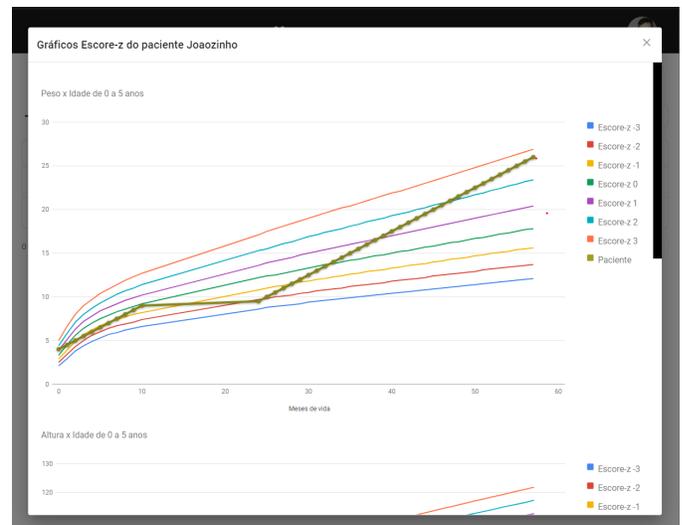


Figura 13. Gráfico Escore-z

A Figura 14 ilustra a página de gestão de responsáveis, na qual é viável realizar todas as operações relacionadas à criação, edição e exclusão de informações acerca do responsável.

Após a liberação de acesso pelo pediatra ao responsável pela criança, este tem a capacidade de visualizar as orientações registradas no sistema ao efetuar o login. Essa prática não apenas aprimora a comunicação, mas também promove a consistência nas informações fornecidas, facilitando o acesso e a compreensão das instruções. Esse processo culmina em cuidados mais eficazes e personalizados para as crianças,



Figura 14. Tela de Gerenciamento de Responsáveis

fomentando seu crescimento e desenvolvimento saudáveis, como ilustrado na Figura 15.

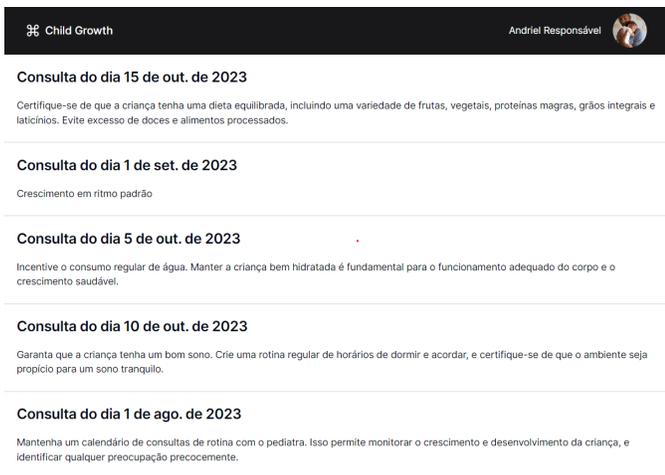


Figura 15. Tela de Gerenciamento de Responsáveis

VII. CONCLUSÃO

Este trabalho delineou o desenvolvimento de um sistema web dedicado ao acompanhamento e suporte ao crescimento infantil, utilizando gráficos e análise de informações como ferramentas principais. O sistema foi concebido com o propósito de simplificar a análise do pediatra, elevando a qualidade das consultas e otimizando o acompanhamento realizado pelos profissionais de pediatria.

A escolha da metodologia ágil FDD (Feature-Driven Development) para o desenvolvimento deste projeto foi motivada por sua adaptabilidade a mudanças em qualquer fase do projeto. Os principais diagramas dessa metodologia foram empregados para exemplificar a construção do sistema, representando os processos, a arquitetura e os requisitos de forma clara.

Como sugestões para trabalhos futuros, seria valioso considerar o desenvolvimento de um aplicativo móvel complementar. Isso permitiria aos pais acompanhar os dados de crescimento de maneira mais conveniente e receber orientações de forma acessível. Além disso, o aplicativo poderia oferecer funcionalidades offline para acesso aos registros, assegurando disponibilidade em locais com conexão limitada. Introduzir um recurso de chat na aplicação também seria benéfico, facilitando a comunicação direta entre o pediatra e os pais. Isso possibilitaria esclarecimento de dúvidas, fornecimento de orientações em tempo real e manutenção de um acompanhamento mais próximo ao longo do processo de crescimento da criança.

REFERÊNCIAS

- [1] WORLD HEALTH ORGANIZATION. WHO Child Growth Standards: Length/height-for-age, Weight-for-age, Weight-for-length, Weight-for-height and Body Mass Index-for-age: Methods and Development. Geneva, Switzerland: World Health Organization, 2006.
- [2] Sociedade Brasileira de Pediatria. "Crescimento". Disponível em: <https://www.sbp.com.br/especiais/pediatria-para-familias/desenvolvimento/crescimento/>. Acesso em: 01 de julho de 2023.
- [3] Microsoft. "Introducing ASP.NET Core". Microsoft Documentation. Último acesso em 30 de maio de 2023. Disponível em: <https://docs.microsoft.com/en-us/aspnet/core/>.
- [4] React. (2021). React - A JavaScript library for building user interfaces. [Online]. Available: <https://reactjs.org/>
- [5] Typescript. (2023). Typescript - TypeScript is a strongly typed programming language that builds on JavaScript, giving you better tooling at any scale. [Online]. Available: <https://www.typescriptlang.org/>
- [6] shadcn/ui. (2023). Beautifully designed components that you can copy and paste into your apps. Accessible. Customizable. Open Source. [Online]. Available: <https://ui.shadcn.com/>
- [7] Next.JS. (2023). The React Framework for the Web. [Online]. Available: <https://nextjs.org/>
- [8] Microsoft Corporation, "SQL Server - Overview." [Online]. Available: <https://docs.microsoft.com/en-us/sql/sql-server/>. Accessed on: May 30, 2023.
- [9] Microsoft Corporation, "SQL Server Overview," Microsoft Docs. [Online]. Disponível: <https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation>. Acessado em: maio 30, 2023.
- [10] N. F. Carlos, S. P. Eduardo, ANÁLISE E MANUTENÇÃO DO SISTEMA GERADOR DE GRÁFICOS DE CRESCIMENTO. Disponível em: <https://repositorio.ufsc.br/handle/123456789/184297>. Acessado em (01/05/2023)
- [11] C. C. Augusto, Sistema Gerador de Gráficos de Crescimento. Disponível em: <https://repositorio.ufsc.br/handle/123456789/183782>. Acessado em (21/05/2023)
- [12] Dantas, N. Cilene, Software-protótipo para o acompanhamento do crescimento e desenvolvimento infantil a partir da classificação internacional das práticas de enfermagem (CIPE®). Disponível em: <https://repositorio.ufrn.br/handle/123456789/22708>. Acessado em (30/05/2023)

- [13] P. Tebaldi. Quais são os benefícios das metodologias ágeis para a gestão de ti?. Disponível em: <https://www.opservices.com.br/beneficios-das-metodologias-ageis/>. (Acessado em 31/05/2023).
- [14] F.G. Silva, S.C.P. Hoentsch e and L.Silva. Uma análise das metodologias ágeis fdd e scrum sob a perspectiva do modelo de qualidade, mps. br. Scientia Plena, 5(12), 2009.
- [15] ALVES, R. S.; OLIVEIRA, J. C. Sistema Web para Geração e Análise de Gráficos de Crescimento Infantil. In: Anais do Congresso Brasileiro de Informática em Saúde, 2021, Florianópolis. Anais do Congresso Brasileiro de Informática em Saúde, 2021. p. 1-5.
- [16] M. Chapple, "SQL Server 2019: A Beginner's Guide."McGraw-Hill Education, 2019.