

# Desenvolvimento de um Sistema Web para acompanhamento de pacientes em tratamento de incontinência urinária

Victor Santos Marques, Alessandro André Mainardi de Oliveira

Curso de Ciência da Computação

Universidade Franciscana (UFN)

Santa Maria - RS

victor.marques@ufn.edu.br, alessandroandre@ufn.edu.br

**Resumo**—Este artigo descreve o desenvolvimento de um sistema de telessaúde para o acompanhamento de pacientes em tratamento de incontinência urinária. O sistema permite que os pacientes registrem informações sobre a frequência urinária e a ingestão de líquidos, que são enviadas ao fisioterapeuta para acompanhamento. O profissional pode analisar esses dados e fornecer as orientações necessárias. A aplicação foi desenvolvida para a plataforma web, utilizando a linguagem de programação TypeScript, o framework Next.js do React e a plataforma Node.js. Para a persistência dos dados, foi utilizado o Sistema Gerenciador de Banco de Dados PostgreSQL e o ORM Prisma. Durante o desenvolvimento, foram adotadas boas práticas da metodologia ágil FDD.

**Palavras-chave:** Telessaúde, Fisioterapia, TypeScript, Next.js, Node.js

## I. INTRODUÇÃO

A incontinência urinária (IU) é uma disfunção do assoalho pélvico (DAP) presente em 27% da população mundial, sendo duas vezes mais frequente em mulheres. Na pós-menopausa a situação se agrava, e a probabilidade de se verificar tal distúrbio pode chegar a 70%, variando de acordo com alguns fatores, como a gestação, envelhecimento e condição socioeconômica [1].

O uso de telessaúde no Brasil iniciou-se no ano de 2007 com o programa Telessaúde Redes com três subdivisões: teleconsultoria, telediagnóstico e teleducação. O intuito era melhorar os atendimentos da rede primária no sistema único de saúde (SUS), a ideia inicial melhorar os atendimentos da atenção básica, redução de gastos e tempo sobre deslocamentos, porém não foi tão eficaz devido a dificuldade de acesso, deficit de aprendizado sobre o uso e baixa estrutura [2].

Segundo Stanhope e Weinstein [3], a ciência e a prática da fisioterapia com suporte digital cresceram rapidamente nos últimos anos e nunca houve melhor momento para fornecer uma solução rápida, porque tecnologia adequada, acessível e amigável ao consumidor nunca esteve tão disponível quanto no momento atual. Isto fornece aos fisioterapeutas novas maneiras de aplicar tratamentos com eficácia conhecida e para fornecer estratégias de tratamento inovadoras sustentadas por tecnologias modernas que proporcionarão continuidade do atendimento para pacientes e algum grau de constância de negócios para profissionais no período iminente [4].

Nesse contexto, o propósito deste trabalho é desenvolver uma aplicação web para acompanhamento de pacientes em

tratamento de incontinência urinária, permitindo ao fisioterapeuta acompanhar diariamente o progresso do paciente e possibilitando o envio de orientações para auxiliar no tratamento deste problema.

Olivetto, Lima e Alencar [5] corroboram com esta ideia ao mencionarem que o papel do fisioterapeuta no tratamento da IU é fundamental, pois atuam com mecanismos menos invasivos nas pacientes, sendo o exercício mais eficaz a contração controlada e sistemática dos músculos do assoalho pélvico, que ajudam a aumentar a capacidade dos grupos musculares de se contraírem automaticamente, fortalecendo-os.

## A. Justificativa

A ideia para desenvolvimento deste projeto surgiu do curso de Fisioterapia da Universidade Franciscana (UFN), onde após a realização dos atendimentos de saúde da mulher no Laboratório de Ensinos Práticos da Universidade, foi relatada uma grande demanda de pacientes na área, e que não se conseguiria dar conta de tudo presencialmente. A partir disso projetou-se a ideia de uma aplicação de telessaúde que auxiliasse no tratamento desses pacientes.

Na perspectiva de Pereira [6], a incontinência urinária é um problema que acomete um número muito maior de mulheres de todas as faixas etárias e, por conta disto, suas saúdes uroginecológicas acabam sendo influenciadas, afetando diretamente a qualidade de vida destas.

A utilização da Telessaúde e formas remotas de comunicação teve um grande avanço com o início da pandemia da COVID-19, sendo possível observar a existência de várias formas de comunicação à distância durante este período [7].

O desenvolvimento de um sistema web para acompanhamento de pacientes em incontinência urinária seria extremamente útil, pois ofereceria uma série de benefícios tanto para as pacientes quanto para os profissionais envolvidos no tratamento dessa condição.

A proposta desse sistema busca proporcionar monitoramento contínuo, permitindo acompanhar de perto o progresso das pacientes ao longo do tempo. Além disso, o sistema possibilitaria o registro de dados detalhados, fornecendo um histórico preciso das ocorrências relacionadas à incontinência urinária. Essas informações seriam valiosas, pois ao analisar os registros miccionais, o profissional de saúde pode identificar padrões, desencadeadores e fatores de risco relacionados à

incontinência urinária. Com base nessas informações, um plano de tratamento individualizado pode ser desenvolvido.

### B. Objetivo Geral

Este trabalho teve como objetivo o planejamento e desenvolvimento de um sistema web para o acompanhamento de pacientes em tratamento de incontinência urinária, permitindo que os pacientes registrem informações sobre seu padrão miccional e ingestão de bebidas, enquanto o fisioterapeuta poderá analisar esses dados, fornecendo observações para melhorar o tratamento.

### C. Objetivos Específicos

- Implementar a aplicação web, utilizando a linguagem de programação TypeScript e o *framework* Next.js da biblioteca React juntamente com a plataforma Node.js;
- Fazer uso do SGBD PostgreSQL, visando sua fácil manutenção e quantidade de documentação;
- Utilizar as boas práticas metodologia FDD (Feature Driven Development) para realização da análise e projeto da aplicação;

## II. REFERENCIAL TEÓRICO

Nesta seção serão abordados os conceitos que embasaram o presente trabalho, relacionados à Incontinência Urinária e a Fisioterapia, além das tecnologias utilizadas no desenvolvimento do projeto.

### A. A incontinência urinária e a fisioterapia

A incontinência urinária (IU) é a perda involuntária de urina, podendo estar relacionada a diversos fatores, como a exemplo, a idade, trauma do assoalho pélvico, fatores hereditários, raça, menopausa, obesidade, doenças crônicas, e exercícios intensos na região abdominal. A IU é um problema de saúde pública, pois além de ser multifatorial é capaz de desencadear efeitos negativos para a paciente [8].

A fisioterapia é um dos tratamentos mais eficaz para tratamento de incontinência urinária, podendo auxiliar na percepção corporal, devolvendo e melhorando a função de Assoalho Pélvico (AP), melhora da tonicidade muscular promove assim uma reabilitação em aspectos de continência urinária, e assim devolvendo uma melhora na qualidade de vida da paciente. O tratamento de reabilitação da musculatura do AP consegue abranger mulheres de diferentes idades, e capazes de tratar cada uma com individualidade, pautada em uma avaliação específica para cada caso [6].

Segundo Patricio [9], uma das ferramentas utilizadas para monitorar a IU pelos pacientes é o diário miccional, conforme mostra a Figura 1, sendo este um recurso simples e com baixo custo benefício. Nele é preciso registrar a frequência em que ocorrem os episódios de incontinência, os horários se é mais diurno ou noturno, ajudando na avaliação para averiguar qual tipo de incontinência o paciente se encaixa.

Com base nas informações registradas no diário miccional, o fisioterapeuta especializado em incontinência urinária pode fazer uma análise detalhada do padrão de micção do paciente.

HORA	BEBIDA		XIXI NORMAL			PERDA DE XIXI			COMO FOI QUE PERDEU XIXI		QUANTIDADE DE ABSORVENTES OU FRALDAS
	TIPO	QUANTIDADE	QUANTAS VEZES	POUCO	MÉDIO	MUITO	POUCO	MÉDIO	MUITO	TOSSE, FORÇA, MUDAR DE POSIÇÃO	QUANTIDADE
00:00 - 01:00											
01:00 - 02:00											
02:00 - 03:00											
03:00 - 04:00											
04:00 - 05:00											
05:00 - 06:00											
06:00 - 07:00											
07:00 - 08:00											
08:00 - 09:00											
09:00 - 10:00											
10:00 - 11:00											
11:00 - 12:00											
12:00 - 13:00											
13:00 - 14:00											
14:00 - 15:00											
15:00 - 16:00											
16:00 - 17:00											
17:00 - 18:00											
18:00 - 19:00											
19:00 - 20:00											
20:00 - 21:00											
21:00 - 22:00											
22:00 - 23:00											
23:00 - 00:00											

Figura 1. Diário Miccional [9]

Isso permite identificar gatilhos ou fatores desencadeantes da incontinência urinária, como determinadas atividades, ingestão de líquidos, horários específicos, entre outros.

### B. Tecnologias

1) TypeScript: TypeScript é uma linguagem de programação fortemente tipada que se baseia em JavaScript, oferecendo melhores ferramentas em qualquer escala [10]. Além da tipagem estática, o TypeScript também oferece suporte a recursos como interfaces, classes, herança, polimorfismo e módulos, que são comuns em linguagens de programação orientadas a objetos. Esses recursos permitem uma organização mais estruturada do código e a reutilização de componentes.

2) React e Next.js: O React é uma biblioteca JavaScript para criação de interfaces para usuários, desenvolvido por engenheiros de software do Facebook para resolver desafios ligados a interfaces complexas com dados que mudam com o passar do tempo [11].

O Next.js é um *framework* de desenvolvimento web para criação de aplicações React. Ele foi projetado para facilitar a construção de aplicativos web escaláveis, oferecendo recursos avançados como renderização do lado do servidor (SSR), renderização estática e roteamento [12]. Uma das principais vantagens do Next.js é a sua facilidade de uso e configuração mínima. Ele fornece um conjunto de funcionalidades padrão que aceleram o processo de desenvolvimento.

3) Chakra UI: O Chakra UI é uma biblioteca de componentes React que facilita a criação de interfaces de usuário acessíveis, reutilizáveis e compostas [13]. O principal objetivo do Chakra UI é simplificar o processo de criação de interfaces de usuário, evitando a criação repetitiva de estilos e comportamentos comuns, economizando tempo e esforço.

4) Node.js: Como um ambiente de execução JavaScript assíncrono orientado a eventos, o Node.js é projetado para desenvolvimento de aplicações escaláveis de rede. Os usuários do Node.js não precisam se preocupar com *deadlock* de processos, pois não existem *locks*. Quase nenhuma função no Node.js realiza diretamente operações de E/S, por essa

razão o processo nunca bloqueia. Por não existirem operações bloqueantes, sistemas escaláveis são razoavelmente fáceis de serem desenvolvidos em Node.js [14].

Atualmente, o principal bloco de construção de aplicações web com JavaScript e Node.js é o Express.js. Segundo sua documentação oficial [15], Express é um *framework* para aplicativo da web do Node.js mínimo e flexível que fornece um conjunto robusto de recursos para aplicativos web e móvel.

5) PostgreSQL: O PostgreSQL é um sistema gerenciador de banco de dados objeto-relacional, o qual, como o nome sugere, usa e estende a linguagem SQL, combinado com funcionalidades que guardam e escalam de forma segura dados complexos. O SGBD vem com muitos recursos destinados a ajudar os desenvolvedores a criar aplicativos, os administradores a proteger a integridade dos dados e a criar ambientes tolerantes a falhas, além de ajudá-lo a gerenciar seus dados, independentemente do tamanho do conjunto de dados. Além de ser gratuito e de código aberto, o PostgreSQL é altamente extensível. [16]. Sobre a sua arquitetura de servidor, o PostgreSQL possui uma estrutura simples e consiste de uma Memória Compartilhada, Processos de Background e uma estrutura de Diretório de Dados.

6) Prisma: O Prisma é uma ferramenta de desenvolvimento de banco de dados moderna e de código aberto. Ele oferece uma interface de programação (API) para consultar e gerenciar bancos de dados relacionais, como o PostgreSQL, de forma mais eficiente e produtiva [17]. O Prisma atua como um *Object-Relational Mapping (ORM)*, fornecendo uma camada de abstração entre a aplicação e o banco de dados. Isso permite que os desenvolvedores escrevam consultas de banco de dados usando uma linguagem de programação em vez de SQL puro, simplificando o processo de interação com o banco de dados e reduzindo a quantidade de código necessário.

### III. TRABALHOS CORRELATOS

Nesta seção serão apresentados alguns trabalhos com características semelhantes ao sistema proposto, com o intuito de contribuir como base de conhecimento para a elaboração e organização do presente projeto.

#### A. Uma aplicação para o auxílio no tratamento de pacientes com incontinência urinária de esforço

O trabalho realizado por Rodrigues [18] teve como objetivo desenvolver um aplicativo para dispositivos móveis, a fim de auxiliar mulheres sob tratamento da Incontinência Urinária de Esforço (IUE), através de exercícios que possam ser praticados em qualquer local. A aplicação permite que o usuário defina o nível de dificuldade dos exercícios e a partir disso recomenda o número de repetições e a velocidade que o mesmo deve ser executado. Além disso, também é disponibilizado um calendário onde o usuário pode informar diariamente se houve perda de urina.

No trabalho o autor [18] ainda destacou que seria relevante para atender aos anseios do fisioterapeuta um outro aplicativo, onde o mesmo pudesse visualizar em tempo real o progresso dos seus pacientes e interagir com eles de forma remota. Assim

o profissional possivelmente estenderia suas recomendações para momentos pós-sessão.

#### B. *Continance App: aplicativo educativo para prevenção da incontinência urinária em mulheres após o parto*

Buscando uma maneira de capacitar os indivíduos a adotarem comportamentos positivos em relação a sua saúde, o trabalho realizado por Saboia [19] se propôs a construir e validar um aplicativo educativo como tecnologia para prevenção da incontinência urinária em mulheres após o parto. A aplicação foi implementada para as plataformas Android e iOS, disponibilizando cartões informativos sobre a patologia e formas de prevenção, além de um programa de exercícios para fortalecimento do assoalho pélvico. Além disso, contou com a avaliação de especialistas das áreas da saúde e da tecnologia para validar a aplicação quanto ao conteúdo e à aparência.

Segundo Saboia [19], o aplicativo nomeado de *Continance App*® foi desenvolvido com rigor metodológico e pautado em um referencial teórico seguro e mostrou-se como material válido, apresentando avaliações positivas, em sua maioria, na avaliação dos especialistas e do público-alvo.

#### C. *Minha Bexiga*

O Minha Bexiga é um aplicativo destinado ao registro do hábito urinário, que é uma informação importante para diagnóstico e tratamento das disfunções miccionais, na forma de um diário miccional [20]. O paciente pode registrar no aplicativo sua frequência urinária, a quantidade de líquidos ingerida, a urgência em ir ao banheiro, a perda urinária, a micção noturna, o uso de fraldas ou absorventes, o humor e o sono.

Os registros podem ser contabilizados durante três ou sete dias, antes ou depois de um tratamento, de acordo com a orientação médica. Além disso, cada paciente recebe um código, que deve ser informado ao médico para acesso à essas informações. O aplicativo é projetado para sistemas iOS e ainda conta com uma sistema Web, onde os profissionais da saúde podem se cadastrar e acessar os dados do paciente por meio do código recebido.

#### D. *Considerações sobre os trabalhos correlatos*

Os trabalhos citados tiveram grande relevância para o andamento deste trabalho, tendo em comum o objetivo de auxiliar pacientes com incontinência urinária, cada um abordando diferentes aspectos e oferecendo soluções específicas para esse problema.

O trabalho de Rodrigues [18] destaca a importância de exercícios práticos e flexíveis que possam ser realizados em qualquer lugar. Essa abordagem reconhece a necessidade de oferecer aos pacientes opções acessíveis e adaptáveis às suas rotinas diárias. No entanto, um ponto que o trabalho de Rodrigues não aborda é a avaliação em tempo real do progresso dos pacientes pelos profissionais de saúde.

Já o trabalho de Saboia [19] enfatiza a prevenção da incontinência urinária, oferecendo um aplicativo educativo

com informações relevantes sobre a patologia e um programa de exercícios específicos. Essa abordagem complementa o trabalho de Rodrigues, pois destaca a importância da conscientização e do fortalecimento do assoalho pélvico para evitar a ocorrência ou agravamento da incontinência urinária. Além disso, a validação do aplicativo por especialistas da área e as avaliações positivas demonstram a confiabilidade e a qualidade do conteúdo disponibilizado.

O aplicativo Minha Bexiga [20], por sua vez, fornece um diário miccional abrangente, permitindo aos pacientes registrar informações importantes para o diagnóstico e tratamento das disfunções miccionais. Essa funcionalidade complementa os outros dois trabalhos, pois reconhece a importância de monitorar e acompanhar o hábito urinário ao longo do tempo. Além disso, a disponibilização de um sistema web para que os profissionais de saúde possam acessar os dados dos pacientes facilita a comunicação e o acompanhamento remoto.

A partir dessa análise, torna-se evidente a relevância da criação de um sistema para o acompanhamento de pacientes com incontinência urinária, considerando as contribuições dos três trabalhos correlatos mencionados. Um sistema que ofereça informações educativas sobre a patologia e prevenção da incontinência urinária, permitindo o registro e o monitoramento do hábito urinário dos pacientes e facilitando a interação entre pacientes e profissionais de saúde. Essa abordagem seria interessante, pois abordaria diferentes aspectos da condição, oferecendo suporte contínuo aos pacientes e facilitando o acompanhamento pelos profissionais de saúde.

#### IV. METODOLOGIA

As metodologias ágeis visam minimizar os riscos relacionados às incertezas dos projetos e com os imprevistos de maneira mais natural e rápida, permitindo uma abordagem flexível e adaptativa no desenvolvimento de software [21]. Essa abordagem possibilita uma maior capacidade de resposta a mudanças, uma vez que as incertezas e os imprevistos são esperados e tratados como parte do processo. Dessa forma, as metodologias ágeis promovem uma maior eficiência no desenvolvimento, proporcionando resultados mais satisfatórios e alinhados com as necessidades do projeto. 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 contém características de processos ágeis, como foco na programação, interação constante com o cliente e entrega frequente de versão do produto. Enquanto a metodologia se concentra principalmente no desenvolvimento de software em si, é importante destacar que outras considerações, como a escolha de tecnologias e ferramentas, a definição de procedimentos de aquisição, entre outros fatores, devem ser tratadas separadamente para garantir um ambiente propício ao sucesso do projeto. [22].

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 [22].

##### A. Desenvolver um modelo abrangente

O processo de Desenvolver um Modelo Abrangente é responsável pelo estudo detalhado sobre o domínio do negócio e pela definição do escopo do projeto [22]. Isso ajuda a estabelecer uma base sólida para o desenvolvimento e evita a inclusão de elementos desnecessários.

Uma ferramenta valiosa utilizada no processo de desenvolvimento do modelo abrangente é o Diagrama de Domínio. A Figura 2 apresenta esse diagrama, representando as entidades conceituais do domínio, bem como os relacionamentos entre elas, fornecendo uma visão clara e concisa dos componentes do sistema e como eles se conectam.

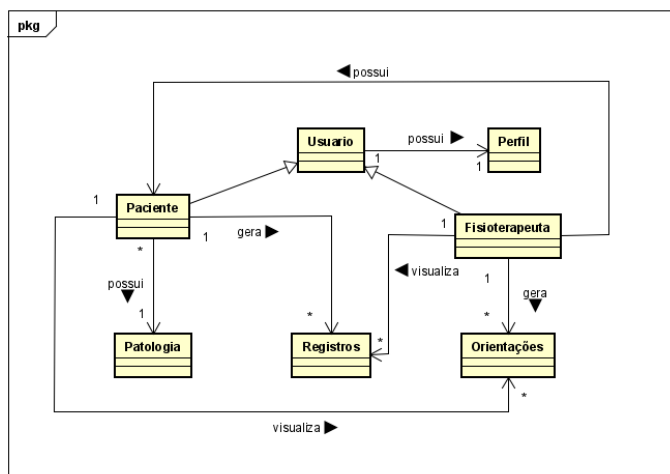


Figura 2. Diagrama de Domínio

Seguindo a metodologia FDD, após o desenvolvimento do diagrama de domínio, o próximo passo é a elaboração de uma lista de funcionalidades. Essa lista descreve as principais características e funcionalidades que o sistema deve ter para atender aos requisitos do projeto.

##### B. Planejar através de funcionalidades

Neste processo é essencial identificar e priorizar as funcionalidades que serão implementadas no sistema. Essa etapa envolve uma análise cuidadosa dos requisitos funcionais, que descrevem as ações e comportamentos específicos que o sistema deve ser capaz de realizar, e também dos requisitos não funcionais, que abrangem aspectos como desempenho, segurança, usabilidade e outros critérios técnicos relevantes.

Para auxiliar nesse processo, a Tabela 1 apresenta os Requisitos Funcionais (RF), na qual cada funcionalidade é listada juntamente com uma estimativa de tempo necessária para o seu desenvolvimento. Essa tabela desempenha um papel fundamental no planejamento e na alocação adequada de recursos para cada funcionalidade.

**Tabela 1. Requisitos Funcionais e Estimativa de Tempo**

<b>RF01</b> - Permitir o cadastro de usuários. <b>Tempo estimado:</b> 10 horas
<b>RF02</b> - Permitir o login de usuários. <b>Tempo estimado:</b> 10 horas
<b>RF03</b> - Permitir ao paciente registrar a quantidade de urina. <b>RF03.1</b> - Informar o horário. <b>RF03.2</b> - Informar a quantidade em ml. <b>RF03.3</b> - Informar se houve urgência de urinar. <b>RF03.4</b> - Informar se houve perda de urina. <b>Tempo estimado:</b> 30 horas
<b>RF04</b> - Permitir ao paciente registrar a ingestão de bebida. <b>RF04.1</b> - Selecionar tipo de bebida. <b>RF04.2</b> - Informar a quantidade em ml. <b>RF04.3</b> - Informar o horário. <b>Tempo estimado:</b> 30 horas
<b>RF05</b> - Permitir ao paciente visualizar dicas relacionadas a incontinência urinária. <b>Tempo estimado:</b> 10 horas
<b>RF06</b> - Permitir ao paciente visualizar registros enviados. <b>Tempo estimado:</b> 10 horas
<b>RF07</b> - Permitir ao paciente editar registros. <b>Tempo estimado:</b> 10 horas
<b>RF08</b> - Permitir ao paciente visualizar orientações do fisioterapeuta. <b>Tempo estimado:</b> 15 horas
<b>RF09</b> - Permitir ao paciente visualizar gráficos de seus registros. <b>Tempo estimado:</b> 30 horas
<b>RF10</b> - Permitir ao fisioterapeuta buscar pacientes no sistema. <b>RF10.1</b> - Permitir ao fisioterapeuta ativar/desativar pacientes. <b>Tempo estimado:</b> 30 horas
<b>RF11</b> - Permitir ao fisioterapeuta visualizar registros de pacientes. <b>Tempo estimado:</b> 20 horas
<b>RF12</b> - Permitir ao fisioterapeuta gerenciar orientações <b>Tempo estimado:</b> 30 horas
<b>RF13</b> - Permitir ao fisioterapeuta visualizar análise de registros. <b>Tempo estimado:</b> 30 horas
<b>RF14</b> - Permitir ao usuário realizar alterações no perfil. <b>Tempo estimado:</b> 10 horas

Já a Tabela 2 descreve os requisitos não funcionais (RNF), que englobam aspectos técnicos e de qualidade do sistema, como desempenho, confiabilidade, compatibilidade, entre outros. Esses requisitos são fundamentais para garantir a eficiência e a qualidade do sistema como um todo.

**Tabela 2. Requisitos Não Funcionais**

<b>RNF01</b> - O sistema será implementado na plataforma Web.
<b>RNF02</b> - O sistema será implementado com o uso da linguagem TypeScript.
<b>RNF03</b> - O sistema será implementado com o uso das tecnologias Next.js, Chakra UI e Node.js.
<b>RNF04</b> - O sistema fará uso do ORM Prisma e do SGBD PostgreSQL.

Ao considerar tanto os requisitos funcionais quanto os não funcionais, é possível construir o sistema alinhado às necessidades dos usuários e aos objetivos do projeto. O diagrama de casos de uso apresentado na Figura 3 complementa essa visão, fornecendo uma representação visual das interações entre os atores e as funcionalidades do sistema, auxiliando no entendimento e na validação das especificações. Após a conclusão da lista de funcionalidades e do planejamento, o próximo passo é projetar através de funcionalidades.

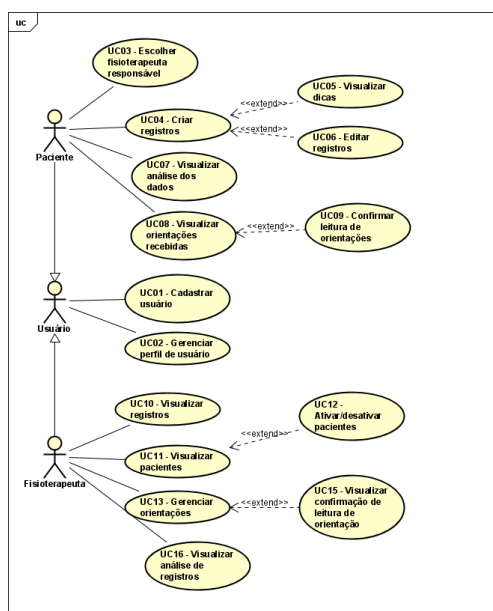


Figura 3. Diagrama de Casos de Uso

### C. Projetar através de 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 das funcionalidades do sistema [22]. Além disso, essa fase também busca identificar possíveis desafios ou obstáculos técnicos que podem surgir durante a implementação das funcionalidades. Essa abordagem proativa permite antecipar problemas e tomar as medidas adequadas para resolvê-los, evitando atrasos e garantindo a qualidade do sistema.

Sendo assim, a Figura 4 apresenta um Diagrama de Classes, que oferece uma visão geral da estrutura do sistema e das relações entre as classes da aplicação. Esse diagrama auxilia no entendimento da arquitetura do sistema, mostrando as entidades principais, seus atributos e métodos, bem como as relações entre elas.

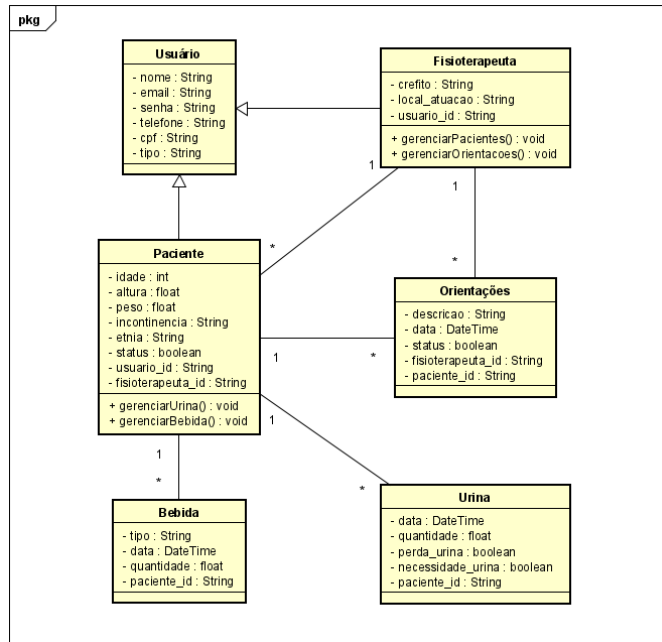


Figura 4. Diagrama de Classes

Além disso, a Figura 5 apresenta um Diagrama de Atividade para os casos de uso do paciente, que descreve o fluxo de atividades sequenciais desde o login até o gerenciamento de seus registros e perfil, incluindo a confirmação de leitura das orientações enviadas pelo fisioterapeuta responsável.

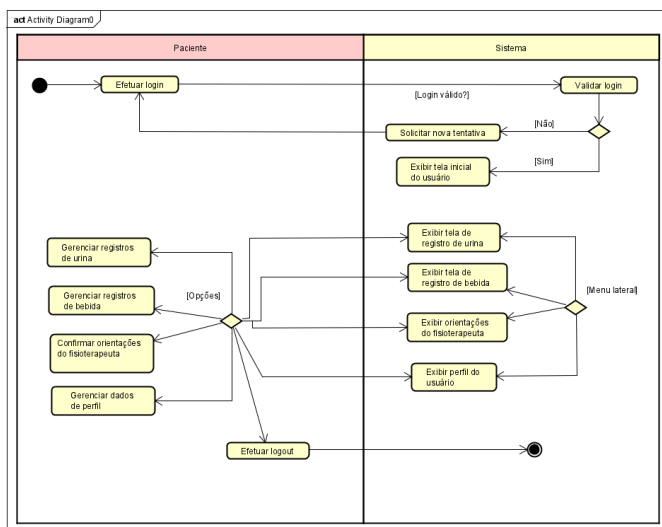


Figura 5. Diagrama de Atividade do Paciente

Seguindo a metodologia FDD, a próxima fase será a de Construir por funcionalidades, que tem como objetivo principal desenvolver o software de forma iterativa e incremental, buscando atender aos requisitos estabelecidos.

#### D. Construir através de funcionalidades

Nessa fase, as funcionalidades são construídas uma a uma, de acordo com as necessidades e prioridades estabelecidas. Ao construir essas funcionalidades, é possível obter um software mais completo e funcional ao longo do tempo. Além disso, essa abordagem facilita a identificação de possíveis ajustes e melhorias durante o processo de desenvolvimento, garantindo que o software atenda efetivamente às expectativas dos usuários.

A Figura 6 ilustra a estrutura de componentes do projeto, proporcionando uma visão abrangente de todas as camadas que compõem o sistema. Por meio dessa representação visual, é possível compreender de forma mais clara a organização e interação entre as diferentes partes do projeto.

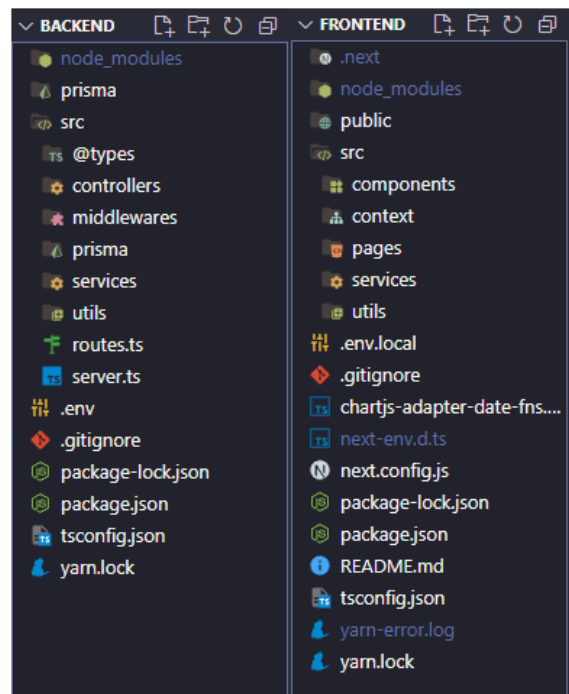


Figura 6. Estrutura do projeto

No *Backend*, a camada de *service* é responsável por conter a lógica de negócio da aplicação, abrangendo as regras e operações relacionadas ao domínio do sistema. A camada de *controller* é responsável por tratar as requisições recebidas pelo servidor e coordenar as ações a serem tomadas, atuando como uma ponte entre as requisições HTTP e a lógica de negócio dos serviços. O *prisma* é responsável pela criação das tabelas no banco de dados, mapeando as entidades e seus relacionamentos, como ilustrado na Figura 7.

```

model Usuario {
  id String @id @default(uuid())
  nome String
  email String
  senha String
  telefone String @default(dbgenerated())
  cpf String
  tipo String
  created_at DateTime? @default(now())
  updated_at DateTime? @updatedAt

  @@map("usuarios")
  fisioterapeutas Fisioterapeuta[]
  pacientes Paciente[]
}

model Fisioterapeuta {
  id String @id @default(uuid())
  crefito String
  atuacao String

  created_at DateTime? @default(now())
  updated_at DateTime? @updatedAt

  usuario Usuario @relation(fields: [usuario_id], references: [id])

  @@map("fisioterapeutas")
  usuario_id String
  pacientes Paciente[]
  orientacoes Orientacao[]
}

```

Figura 7. Estrutura do banco de dados com Prisma

A pasta *utils* é utilizada para organizar e centralizar funções que não estão diretamente relacionadas à lógica de negócios principal do sistema, mas que são úteis em várias partes do código. A camada de *middlewares* serve para intermediar as requisições feitas ao servidor, sendo responsável por processar e manipular os dados antes que eles cheguem às rotas e controladores específicos do aplicativo. A Figura 8 demonstra as funções responsáveis pela gestão de permissões dos usuários dentro da aplicação, que são de grande importância para garantir a segurança e o controle adequado do fluxo de informações.

A função `authUsuario` é um *middleware* de autenticação que verifica se o usuário possui um token JWT (JSON Web Token) de autenticação válido. O *token* é recebido no cabeçalho da requisição e, em seguida, é decodificado usando uma chave secreta definida em uma variável ambiente na pasta *env*. Caso o *token* seja válido, as informações relevantes, como o *sub* (identificador do usuário) e o tipo de usuário, são extraídas e adicionadas ao objeto *req* (requisição). Caso o *token* seja inválido ou inexistente, o usuário não será autorizado a acessar a página.

As funções `protegeRotaFisioterapeuta` e `protegeRotaPaciente` são *middlewares* de autorização que verificam se o tipo de usuário corresponde aos perfis esperados ("Fisioterapeuta" e "Paciente", respectivamente). Caso o tipo de usuário não seja o esperado na rota, o usuário é proibido de acessar a página, recebendo uma mensagem de erro.

```

export function authUsuario(req: Request, res: Response, next: NextFunction) {
  const authToken = req.headers.authorization;

  if (!authToken) {
    return res.status(401).end();
  }

  const [, token] = authToken.split(" ");

  try {
    const { sub, tipo_usuario } = verify(
      token,
      process.env.JWT_USUARIO
    ) as Payload;

    req.usuario_id = sub;
    req.tipo_usuario = tipo_usuario;

    return next();
  } catch (err) {
    return res.status(401).end();
  }
}

export function protegeRotaFisioterapeuta(
  req: Request,
  res: Response,
  next: NextFunction
) {
  if (req.tipo_usuario !== "Fisioterapeuta") {
    return res.status(403).json({ mensagem: "Usuário não autorizado" });
  }
  next();
}

export function protegeRotaPaciente(
  req: Request,
  res: Response,
  next: NextFunction
) {
  if (req.tipo_usuario !== "Paciente") {
    return res.status(403).json({ mensagem: "Usuário não autorizado" });
  }
  next();
}

```

Figura 8. *Middleware* para a gestão de permissões do usuário

Partindo para a parte do *Frontend*, com sua estrutura representada no lado direito da Figura 6, a pasta *pages* é uma das mais importantes no Next.js. Ela contém os arquivos responsáveis por definir as páginas da aplicação, portanto, cada arquivo colocado nessa pasta é automaticamente mapeado para uma rota acessível na aplicação, o que facilita a navegação entre as diferentes telas do sistema.

Já a pasta *public* é usada para armazenar arquivos estáticos que serão acessíveis publicamente, como as imagens utilizadas no sistema. Em *components* são armazenados os componentes reutilizáveis da aplicação. Esses componentes podem ser implementados uma vez e utilizados em várias partes do sistema, evitando a duplicação de código e tornando o desenvolvimento mais eficiente.

A pasta *contexts* é utilizada para armazenar os contextos da aplicação, que podem ser utilizados pelos componentes para acessar e atualizar o estado globalmente, independentemente da hierarquia dos componentes. Eles são particularmente úteis quando há a necessidade de compartilhar informações, como o estado de autenticação do usuário, por exemplo.

Ao organizar o código-fonte do *frontend* dessa maneira, é possível manter uma estrutura clara e modular, o que facilita a manutenção e a reutilização de código, economizando tempo e esforço no desenvolvimento e manutenção do sistema.

## V. RESULTADOS

Como resultado deste presente trabalho, tem-se a construção do sistema web para acompanhamento de pacientes em tratamento de incontinência urinária.

Durante o desenvolvimento do projeto, optou-se pela utilização da linguagem TypeScript devido à sua capacidade de adicionar tipagem estática ao JavaScript. Com o TypeScript, é possível definir tipos para variáveis, parâmetros de função, retorno de função e outros elementos do código. Essa tipagem estática ajuda a detectar erros de tipo durante o desenvolvimento, tornando o código mais confiável e reduzindo a ocorrência de erros comuns. A escolha de utilizar o Node.js com o *framework* Express para o desenvolvimento do *backend* deve-se às suas vantagens e adequação ao projeto.

Na camada de *frontend* optou-se pela utilização do *framework* Next.js e da biblioteca de componentes Chakra UI, devido às suas vantagens e benefícios. O Next.js oferece renderização híbrida, facilitando a entrega rápida do conteúdo inicial e melhorando o desempenho da aplicação. Já o Chakra UI fornece uma biblioteca de componentes pré-estilizados e personalizáveis, agilizando o desenvolvimento e facilitando a criação de interfaces modernas e responsivas.

A Figura 9 ilustra a página inicial do paciente após efetuar o *login*, onde são apresentados todos os seus registros, além de oferecer opções de filtragem com base no tipo de registro selecionado. A barra lateral exibe as páginas disponíveis para o paciente dentro da aplicação, possibilitando cadastrar um registro de urina, cadastrar um registro de ingestão de bebidas, acessar as orientações recebidas pelo fisioterapeuta, visualizar análises dos registros por meio de gráficos e editar os dados do perfil.

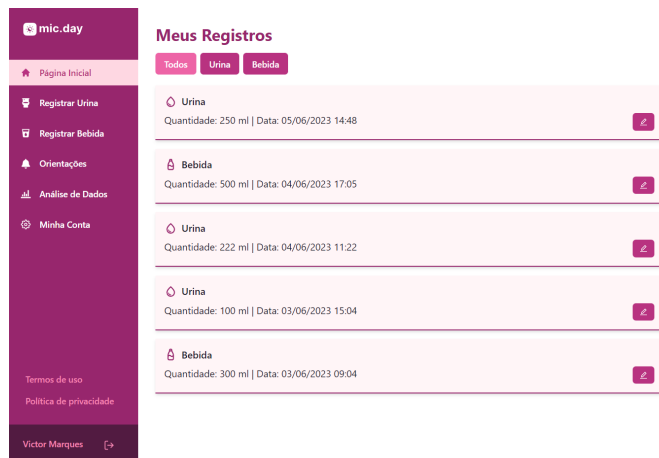


Figura 9. Tela inicial do paciente

A Figura 10 representa a página de registro de urina do paciente, na qual são coletadas informações relevantes sobre a micção. Nessa página, o paciente é solicitado a inserir a data e a hora do registro, indicar a quantidade de urina eliminada e informar se houve episódios de urgência ou perda de urina. Esses dados são fundamentais para avaliar o funcionamento da bexiga e identificar possíveis padrões ou

problemas relacionados à incontinência urinária. Além disso, a página oferece dicas educativas ao paciente, fornecendo orientações sobre como prevenir a incontinência urinária e melhorar a saúde do trato urinário.

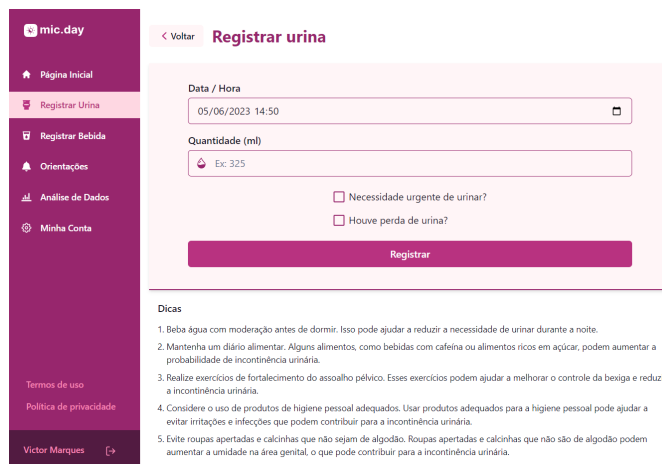


Figura 10. Tela de registro de urina do paciente

A Figura 11 exibe a página de registro de bebida do paciente, onde é necessário fornecer a data e hora do registro, selecionar o tipo de bebida consumida e indicar a quantidade ingerida. Esses detalhes são essenciais para o acompanhamento do paciente, pois permitem que o fisioterapeuta avalie o impacto da ingestão de líquidos na incontinência urinária e faça ajustes necessários no tratamento.



Figura 11. Tela de registro de bebida do paciente

A Figura 12 ilustra a página de Orientações do paciente, onde é possível visualizar todas as orientações enviadas pelo fisioterapeuta responsável. Nessa página, as orientações são exibidas de forma organizada, apresentando informações como a data de envio e o conteúdo da orientação.



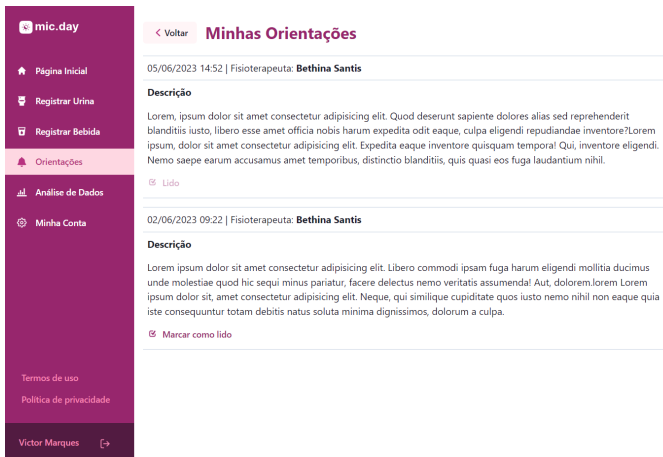


Figura 12. Tela de orientações do paciente

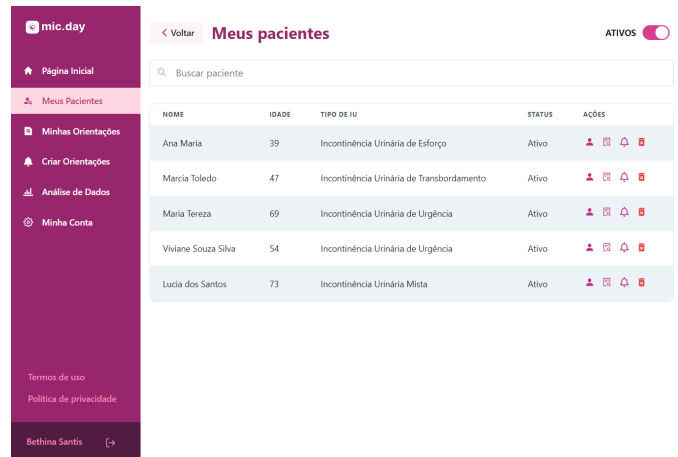


Figura 14. Tela de pacientes do fisioterapeuta

Para o fisioterapeuta, ao realizar o *login*, a Figura 13 representa a Página inicial da aplicação, que exibe os últimos registros dos seus pacientes no dia atual. Essa página oferece ao fisioterapeuta a visão geral dos registros mais recentes, permitindo que ele acompanhe facilmente a atividade dos pacientes. Além disso, a página oferece opções de filtro, permitindo que o fisioterapeuta visualize os registros com base no tipo selecionado, o que facilita a análise e o acompanhamento específico de determinados aspectos da incontinência urinária.

Ao acessar os registros de um paciente específico, o sistema exibirá tabelas separadas para os registros de urina e de bebida, como exemplificado na Figura 15. Na tabela de registros de urina, serão apresentados dados como data e hora, quantidade urinada e indicação de urgência ou perda de urina. Já na tabela de registros de bebida, serão exibidos detalhes como data e hora, tipo de bebida e quantidade ingerida.

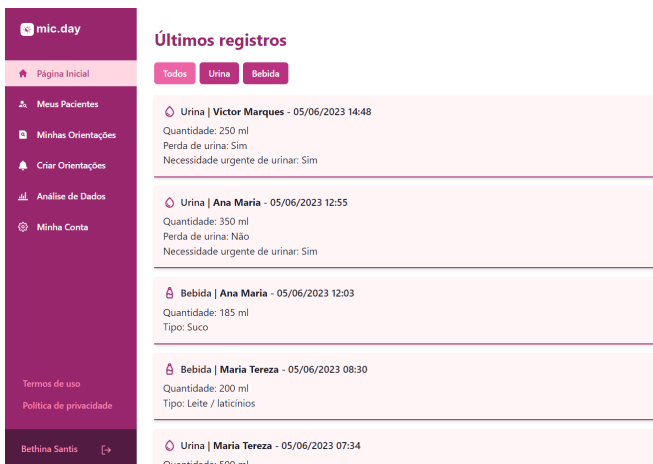


Figura 13. Tela inicial do fisioterapeuta



Figura 15. Tela de pacientes do fisioterapeuta

A página *Meus pacientes* demonstrada na Figura 14, é onde o fisioterapeuta tem acesso a uma tabela onde estão listados todos os seus pacientes. Essa tabela permite que o fisioterapeuta realize diversas ações, como visualizar o perfil do paciente, acessar os registros de urina e bebida, e visualizar as orientações enviadas ao paciente. Além disso, a página oferece a opção de ativar ou desativar um paciente, o que pode ser útil quando o tratamento para a incontinência urinária já foi concluído.

A Figura 16 apresenta a página *Criar orientações*, onde o fisioterapeuta poderá enviar orientações aos pacientes. Nessa página, o profissional tem a opção de selecionar o paciente específico para o qual a orientação será direcionada. Além disso, é possível definir a data e hora desejadas para o envio da orientação. O campo de texto permite ao fisioterapeuta descrever de forma detalhada as instruções ou conselhos que devem ser transmitidos ao paciente.



Figura 16. Tela de criar orientações do fisioterapeuta

Para auxiliar o fisioterapeuta na análise dos dados de registros do paciente de forma mais clara e intuitiva, foram implementados gráficos que apresentam os registros diários de forma visual. A Figura 17 exemplifica a página de análise de dados, onde são exibidos gráficos que permitem a visualização dos registros diários de um determinado paciente. Esses gráficos mostram a soma de todos os registros em um determinado dia, permitindo o filtro desses registros a partir de um intervalo de datas, além do tipo de registro que deseja visualizar.

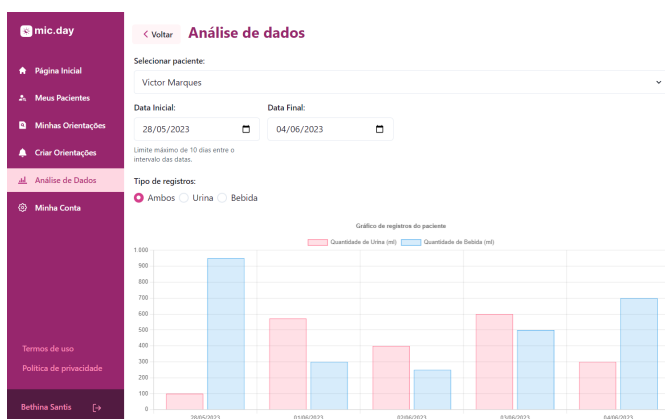


Figura 17. Tela de análise de dados do fisioterapeuta

## VI. CONCLUSÃO

Neste trabalho foi apresentado o desenvolvimento de um sistema web com o objetivo de acompanhar e auxiliar no tratamento de pacientes com incontinência urinária. O sistema tem como propósito facilitar o processo de recuperação dos pacientes, bem como otimizar o acompanhamento realizado pelos profissionais de fisioterapia.

Com base nas pesquisas relacionadas, observou-se que a maioria das aplicações disponíveis para o tratamento da incontinência urinária se concentra em exercícios de fortalecimento ou em fornecer informações educativas sobre a doença. No entanto, percebeu-se uma lacuna em relação a aplicações

que permitam um acompanhamento mais próximo por parte dos fisioterapeutas, bem como uma melhoria no processo de tratamento para o usuário.

Com o desenvolvimento deste sistema, busca-se ampliar o acompanhamento dos fisioterapeutas aos pacientes, diminuindo a dependência de consultas presenciais em grande quantidade. Isso permitirá que os profissionais possam priorizar os momentos em que a presença física é necessária, otimizando o tempo de consulta e proporcionando um acompanhamento mais regular e eficaz ao longo do tratamento.

Para o desenvolvimento deste trabalho, escolheu-se a metodologia ágil FDD (Feature-Driven Development), devido à sua capacidade de se adaptar a mudanças em qualquer estágio do projeto. Os principais diagramas dessa metodologia foram utilizados para exemplificar a construção do trabalho, representando os processos, a arquitetura e os requisitos do sistema.

Durante o desenvolvimento do projeto, uma das dificuldades enfrentadas foi a implementação da autenticação do fisioterapeuta no processo de cadastro. Para resolver esse problema, optou-se por exigir que o paciente selecione o fisioterapeuta responsável no momento do cadastro, além de realizar uma segunda confirmação para evitar possíveis erros por parte do usuário. Essa abordagem garante que somente o fisioterapeuta designado tenha acesso aos dados pessoais e registros do paciente, assegurando a privacidade e a segurança das informações.

Como sugestão para trabalhos futuros, seria interessante considerar o desenvolvimento de um aplicativo móvel complementar, permitindo aos pacientes registrar seus dados miccionais e receber orientações de forma mais conveniente. Além disso, o aplicativo poderia oferecer a funcionalidade de acesso *offline* aos registros, garantindo a disponibilidade mesmo em locais com conexão limitada. Outra sugestão seria a criação de um recurso de chat dentro da aplicação, permitindo um contato mais direto e fácil entre o fisioterapeuta e o paciente. Isso facilitaria a comunicação, possibilitando esclarecer dúvidas, fornecer orientações em tempo real e manter um acompanhamento mais próximo ao longo do tratamento.

## REFERÊNCIAS

- [1] E. Baracho, *Fisioterapia Aplicada à Saúde da Mulher*. Guanabara Koogan LTDA, 6 edition, 2018.
- [2] C.A. Schmitz e E.Harzheim, *Oferta e utilização de teleconsultorias para atenção primária à saúde no Programa Telessaúde Brasil Redes*. Revista Brasileira de Medicina de Família e Comunidade, v. 12, n. 39, p. 1-11, 2017.
- [3] J. Stanhope e P. Weinstein, *Learning from COVID-19 to improve access to physiotherapy*. Aust J Prim Health, Estados Unidos, 2020.
- [4] C. Adamse, M.G. Dekker-Van Weering, F.S. van Etten-Jamaludin e M.M. Stuiver, *The effectiveness of exercise-based telemedicine on pain, physical activity and quality of life in the treatment of chronic pain: A systematic review*. J Telemed Telecare, Estados Unidos, 2018.
- [5] M.S. Olivetto, B.E. Lima, I. Alencar, *A intervenção da fisioterapia no tratamento da incontinência urinária de esforço*. Research, Society and Development, v. 10, n. 12, 2021.

- [6] E. Pereira, E.C. Machado, L. Carvalho, L.A. Silveira e G. Ribeiro, *A influência da cinesioterapia no tratamento da incontinência urinária em mulheres: revisão*. Brazilian Journal of Development, v. 7, n. 1, p. 9734-9748, 2021.
- [7] COFFITO nº 424/2013 e Artigo 15, inciso II e Artigo 39 da Resolução COFFITO nº 425/2013 e estabelece outras providências durante o enfrentamento da crise provocada pela Pandemia do COVID-19. Brasília - 327ª Reunião Plenária Ordinária, 2020.
- [8] A. Barbosa, K.L. Feliciano, D. Dei Tos, L. Fabiano, *Incontinência urinária e qualidade de vida em cuidadoras de pacientes da associação norte paranaense de reabilitação (ANPR)*. Arquivos do Mudi, v. 25, n. 3, 2021.
- [9] M. Patricio, *Os efeitos da fisioterapia pélvica sobre a incontinência urinária após prostatectomia crônica*. Revista Fisio e terapia. v.26, n.111, 2022.
- [10] TypeScript. *Introdução ao TypeScript*. TypeScript. Disponível em: <https://www.typescriptlang.org/pt/>. Acesso em: 17 de maio de 2023.
- [11] A. Banks e E. Porcello, *Learning React: Functional Web Development with React and Redux*. O'Reilly Media, Estados Unidos, 2017.
- [12] Next.js. *What is Next.js?*. Disponível em: <https://nextjs.org/docs/getting-started>. Acesso em: 17 de maio de 2023.
- [13] Chakra UI, *Chakra UI: Create accessible React apps with speed*. Disponível em: <https://chakra-ui.com/>. Acesso em 06 de junho de 2023.
- [14] The OpenJS Foundation, *Sobre Node.js*. Disponível em: <https://nodejs.org/pt-br/about/>. Acesso em 06 de setembro de 2022.
- [15] The OpenJS Foundation, *Express: Framework web rápido, flexível e minimalista para Node.js*. Disponível em: <https://expressjs.com/pt-br/>. Acesso em 06 de setembro de 2022.
- [16] PostgreSQL. *What is PostgreSQL?*. Disponível em: <https://www.postgresql.org/about/>. Acesso em 07 de setembro de 2022.
- [17] Prisma. *Prisma: A next-generation ORM for Node.js and TypeScript*. Disponível em: <https://www.prisma.io/>. Acesso em: 23 de maio de 2023.
- [18] A.C. Rodrigues, *Uma Aplicação para o auxílio no tratamento de pacientes com incontinência urinária de esforço*. Centro de Informática UFPE, Recife, 2019.
- [19] D. Saboya, *Construção e Validação de aplicativo educativo para prevenção da incontinência urinária em mulheres após o parto*. Universidade Federal do Ceará, Fortaleza, 2017.
- [20] EMS, *Minha Bexiga*. Disponível em <https://apps.apple.com/br/app/minha-bexiga/id1537991229?l=en>. Acesso em 15 de outubro de 2022.
- [21] 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/>. Acesso em 29 de outubro de 2022.
- [22] F.G. Silva, S.C.P. Hoentsch e L.Silva. *Uma análise das metodologias ágeis fdd e scrum sob a perspectiva do modelo de qualidade*, mps. br. Scientia Plena, 5(12), 2009.