



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

DANILO CÉSAR RIBEIRO GARCIA DE MEDEIROS

**APPCER: APLICAÇÃO WEB PARA DAR SUPORTE AOS
PROFISSIONAIS DO CER CAMPINA GRANDE**

CAMPINA GRANDE - PB

2022

DANILO CÉSAR RIBEIRO GARCIA DE MEDEIROS

**APPCER: APLICAÇÃO WEB PARA DAR SUPORTE AOS
PROFISSIONAIS DO CER CAMPINA GRANDE**

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

Orientador : Wilkerson de Lucena Andrade

CAMPINA GRANDE - PB

2022

DANILO CÉSAR RIBEIRO GARCIA DE MEDEIROS

**APPCER: APLICAÇÃO WEB PARA DAR SUPORTE AOS
PROFISSIONAIS DO CER CAMPINA GRANDE**

Trabalho de Conclusão Curso apresentado ao Curso Bacharelado em Ciência da Computação do Centro de Engenharia Elétrica e Informática da Universidade Federal de Campina Grande, como requisito parcial para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA:

**Wilkerson de Lucena Andrade
Orientador – UASC/CEEI/UFCG**

**Dalton Dario Serey Guerrero
Examinador – UASC/CEEI/UFCG**

**Francisco Vilar Brasileiro
Professor da Disciplina TCC – UASC/CEEI/UFCG**

Trabalho aprovado em: 02 de Setembro de 2022.

CAMPINA GRANDE - PB

RESUMO

A cidade de Campina Grande abriga uma unidade do CER-IV, um centro especializado em tratamentos de reabilitação, é responsável por atender diversos municípios, incluindo Campina Grande. Isto acarreta em uma enorme quantidade de pacientes visitando o centro por ano, gerando uma enorme carga de trabalho para todos os profissionais alocados no centro, como fisioterapeutas, enfermeiras, fonoaudiólogos, entre outros. Após algumas visitas e entrevistas realizadas com profissionais do centro, foi proposta a criação de uma plataforma web, que funcionaria como um apoio para os profissionais cadastrarem diversas informações relacionadas ao tratamento dos pacientes, além de prover formas de enviar dúvidas ao centro por meio da plataforma. Entre as informações, algumas podem ser citadas como: descrição de atividades que são realizadas com os pacientes e podem ser feitas em casa; utensílios que auxiliam a realizar o tratamento como brinquedos e móveis adaptados, por exemplo. A escolha de implementar a plataforma para web é pelo fácil acesso atualmente que as pessoas têm aos navegadores comuns em seus celulares, facilitando o acesso de todos à informações para o tratamento adequado dos pacientes. Ao final, o aplicativo foi validado por meio de uma entrevista a três fisioterapeutas do CER para que fosse inicializada a utilização do aplicativo no centro de Campina Grande.

AppCER: Aplicação web para dar suporte aos profissionais do CER Campina Grande

Danilo César Ribeiro Garcia de Medeiros

danilo.medeiros@ccc.ufcg.edu.br

Universidade Federal de Campina Grande Campina Grande, Paraíba, Brasil

Wilkerson de Lucena Andrade

wilkerson@computacao.ufcg.edu.br

Universidade Federal de Campina Grande Campina Grande, Paraíba, Brasil

RESUMO

A cidade de Campina Grande abriga uma unidade do CER-IV, um centro especializado em tratamentos de reabilitação, é responsável por atender diversos municípios, incluindo Campina Grande. Isto acarreta em uma enorme quantidade de pacientes visitando o centro por ano, gerando uma enorme carga de trabalho para todos os profissionais alocados no centro, como fisioterapeutas, enfermeiras, fonoaudiólogos, entre outros. Após algumas visitas e entrevistas realizadas com profissionais do centro, foi proposta a criação de uma plataforma web, que funcionaria como um apoio para os profissionais cadastrarem diversas informações relacionadas ao tratamento dos pacientes, além de prover formas de enviar dúvidas ao centro por meio da plataforma. Entre as informações, algumas podem ser citadas como: descrição de atividades que são realizadas com os pacientes e podem ser feitas em casa; utensílios que auxiliam a realizar o tratamento como brinquedos e móveis adaptados, por exemplo. A escolha de implementar a plataforma para web é pelo fácil acesso atualmente que as pessoas têm aos navegadores comuns em seus celulares, facilitando o acesso de todos à informações para o tratamento adequado dos pacientes. Ao final, o aplicativo foi validado por meio de uma entrevista a três fisioterapeutas do CER para que fosse inicializada a utilização do aplicativo no centro de Campina Grande.

Palavras-chaves: CER; atividades; reabilitação; saúde; web; profissional; cuidadores; familiares.

Backend: <https://github.com/danilocsm/t.c.c-backend>

Frontend: <https://github.com/danilocsm/t.c.c-frontend>

Deploy: <https://app-cer.vercel.app>

1. INTRODUÇÃO

O CER-IV (Centro Especializado em Reabilitação) da cidade de Campina Grande - PB foi fundado no ano de 2016 para atender as necessidades de pessoas com deficiência. Ele é responsável por atender 183 municípios, incluindo a cidade de Campina Grande, com mais de 50.000 atendimentos por ano [1]. Desde sua fundação, o centro vem fornecendo auxílio a diversas mães e cuidadores, especialmente de baixa renda, fornecendo tratamentos semanais e os equipamentos necessários para os cuidados dos pacientes. Muitas famílias não possuem os recursos necessários para realizarem o tratamento correto e o CER exerce um papel fundamental assistindo essas pessoas. Entretanto, muitos pais e cuidadores não têm acesso a informação de qualidade sobre tais

cuidados, criando muitas expectativas, na maioria das vezes, inalcançáveis, quando visitam o centro. Isto acaba criando uma sobrecarga nos profissionais de saúde, tendo de lidar tanto com o paciente, como auxiliando os pais a entenderem melhor a situação da pessoa portadora de deficiência. Após visitas ao centro e conversas com os profissionais que estão diariamente vivendo a rotina do centro, foi proposta a criação de uma plataforma de acesso à informação web. A plataforma é pensada como um repositório para acesso e troca de informações entre cuidadores e profissionais. Os pais podem conseguir diretivas sobre os cuidados com os pacientes de maneira mais prática, sendo capazes de realizar depoimentos e tirar dúvidas na própria plataforma. Os profissionais podem cadastrar diversos materiais e referências relacionadas aos cuidados, assim como responder às dúvidas cadastradas pelos cuidadores. A solução envolve a criação de um aplicativo web, com um servidor para armazenar os dados e perguntas cadastrados pelos cuidadores e profissionais, e uma interface para interação dos usuários com as funcionalidades propostas. Para a validação do trabalho, foi realizada uma entrevista com fisioterapeutas do CER, em que foi apresentada a ferramenta e seu funcionamento, além de uma etapa de testes internos utilizando ferramentas para testes automáticos.. Este trabalho permitiu aprimorar a comunicação e cooperação entre cuidador e profissional de saúde, diminuindo a sobrecarga sobre o pessoal capacitado, e aprimorando a capacidade dos familiares e cuidadores de auxiliar os pacientes fora da clínica.

2. TRABALHOS RELACIONADOS

Anteriormente à proposta deste trabalho, foi desenvolvida uma aplicação para auxiliar o tratamento de crianças com microcefalia de maneira remota chamada de Cuidar. O trabalho foi desenvolvido por uma aluna graduanda do curso de Ciência da Computação em conjunto com alunos do mestrado do curso de Design da Universidade Federal de Campina Grande. Entre as funcionalidades principais do sistema Cuidar, a aplicação funcionava como uma forma de continuar o tratamento das crianças remotamente pela o advento da pandemia de Covid-19 em 2020. Os profissionais do CER seriam responsáveis por cadastrar as atividades na plataforma e os pais deveriam acessá-la para consumir essas informações e continuar o tratamento de seus filhos.

Entretanto, após entrevistas com fisioterapeutas do CER de Campina Grande, foi constatado que a ferramenta não chegou a ser utilizada de forma prática com os pais. Muitos relataram que a aplicação era complexa e difícil de ser utilizada, pois dependia de bastante interação dos usuários. Além disso, por exigir constante

interação dos pais com a plataforma para registrar a realização das atividades, muitos deles deixavam de realizar esta etapa por não conseguirem ou terem tempo para tal.

Neste contexto, foi proposta a implementação de uma aplicação web simples, em que ela agiria como um apoio aos profissionais do CER para auxiliar os pais e familiares dos pacientes, em que seu foco seria ajudar todos os tipos de tratamentos de reabilitação que o CER realiza. O AppCER consiste em uma aplicação web em que os pais e familiares têm acesso público às informações cadastradas pelos profissionais do CER, sem necessidade de autenticação, além de poderem enviar dúvidas para o centro. Os profissionais do centros seriam responsáveis por realizar o gerenciamento dos dados que são cadastrados no sistema, além de responder às dúvidas enviadas.

3. SOLUÇÃO

O AppCER é uma aplicação web, de fácil acesso e com interface amigável, em que profissionais do CER, familiares e pacientes possam ter acesso. Ela tem como objetivo principal permitir aos profissionais de saúde cadastrarem informações relacionadas a atividades que podem ser realizadas de maneira remota, utensílios que auxiliam no tratamento, alimentação adequada, roupas e brinquedos. Além disso, a plataforma possui uma seção em que os familiares podem enviar dúvidas para os profissionais e prestar depoimentos sobre suas experiências com o tratamento de pacientes em reabilitação. O profissional do CER será responsável por responder às dúvidas enviadas por meio de uma página privada, além de cadastrar os dados que serão disponibilizados. O acesso às informações é gratuito, disponível para todos que acessarem a aplicação.

3.1 FUNCIONALIDADES

O levantamento das principais funcionalidades que deveriam ser implementadas no AppCER foi feito em conversas com fisioterapeutas do CER de Campina Grande. Além disso, para desenvolvimento da UI (User Interface) da aplicação, foram realizadas reuniões como uma aluna do curso de Design da Universidade Federal de Campina Grande, responsável pela prototipação das telas apresentadas neste trabalho. Desta forma, foi inicializada a implementação das funcionalidades citadas nesta seção.

3.1.1 Cadastro de atividades

A partir da AppCER, os agentes de saúde do CER serão capazes de cadastrar diversas atividades relacionadas ao tratamento de reabilitação dos pacientes, sendo estas públicas para serem acessadas. A Figura 01 mostra o formulário para o cadastro de uma nova atividade no banco de dados da aplicação. O cadastro deve conter dados como o nome, descrição, doenças relacionadas, nível de dificuldade, imagem, utensílios que podem auxiliar e observações.

The image shows a web interface for a health agent. At the top, there's a navigation bar with 'AppCER' and menu items: 'Utensílios', 'Atividades', 'Empenimento', 'Dúvidas', and 'Agentes de Saúde'. Below this is the 'PÁGINA DO AGENTE DE SAÚDE' header with a 'SAIR' button. A user profile section shows a silhouette and the text 'Nome do usuário' and 'email_do_usuario@email.com'. Below the profile are four buttons: 'DÚVIDAS', 'NOVA ATIVIDADE', 'NOVO UTENSÍLIO', and 'LISTA DE ATIVIDADES'. Below these are two more buttons: 'LISTA DE UTENSÍLIOS' and 'EDITAR DADOS DO AGENTE'. The main section is titled 'CADASTRO DE UMA ATIVIDADE' and contains the following form fields: 'NOME DA ATIVIDADE:' (text input), 'SÍNDROMES E DOENÇAS EM FOCO:' (text input), 'DESCRIÇÃO:' (text input), 'SELECIONE A DIFICULDADE:' (radio buttons for 'FÁCIL', 'MÉDIO', 'DIFÍCIL', 'MUITO DIFÍCIL'), 'INSERIR IMAGENS PRSSO-A-PRSSO:' (image upload area with a camera icon and a note: 'Para upload, arraste aqui ou clique no ícone para abrir o arquivo'), 'OBSERVAÇÕES:' (text input), and 'INSERIR UTENSÍLIOS:' (a list of items with '+' and '-' buttons, showing 'utensílio teste' and 'Engrossador de colher de baíno custo'). A 'Cadastrar' button is at the bottom of the form.

Figura 01. Formulário para cadastro de uma atividade.

3.1.2 Cadastro de utensílios

Também será permitido ao profissional de saúde cadastrar os mais diversos itens que possam auxiliar no tratamento dos pacientes. A Figura 03 mostra este formulário, contendo os seguintes campos: nome, preço médio, link para suporte, tipo do utensílio, descrição de como o utensílio deve ser utilizado e uma imagem do utensílio. Além disso, é possível selecionar as atividades disponíveis no sistema para adicionar a relação entre Atividade e Utensílio.

Figura 03. Formulário de cadastro de um novo utensílio.

Figura 04. Editar um utensílio.

3.1.3 Responder a dúvidas

O profissional do CER será responsável por responder às dúvidas enviadas por meio de um dashboard acessado apenas por eles. A Figura 02 mostra como foi implementado esta funcionalidade. Após preencher o campo de resposta, basta clicar no botão de enviar que o servidor será encarregado de encaminhar a resposta para o e-mail presente na dúvida.

Figura 02. Dashboard contendo as dúvidas enviadas.

3.1.4 Editar Atividades e Utensílios

Será responsabilidade do agente de saúde atualizar e editar as atividades e utensílios cadastrados na plataforma, podendo fazer isto por formulários presentes na página do agente de saúde. O profissional de saúde terá acesso a todas as atividades e utensílios cadastrados no sistema, sendo livre para atualizá-los ou excluí-los. As Figuras 04 e 05 mostram, respectivamente, os formulários para editar/excluir uma atividade ou um utensílio.

Figura 05. Editar uma atividade.

3.1.5 Painel do agente de saúde

Todas estas funcionalidades estão agrupadas em um painel com acesso exclusivo dos profissionais do CER, realizado através de

um login com usuário e senha. Para realizar qualquer ação dentro deste painel, o usuário deve possuir um login válido e realizar sua autenticação com o servidor para poder realizar as ações dentro do painel.

Figura 06. Login para profissionais do CER.

A Figura 06 apresenta o formulário de login a ser realizado pelos funcionários do CER. Após autenticação, o usuário será redirecionado para a página do agente de saúde, onde ele terá acesso a todas as funcionalidades citadas acima. A Figura 07 mostra o painel.

Figura 07. Painel do agente de saúde

3.1.6 Enviar dúvidas

Para enviar uma dúvida ao CER, não é necessário cadastro ou login do usuário, sendo esta seção aberta ao público. A Figura 08 demonstra como acessar o formulário para enviar perguntas aos profissionais do CER.

Figura 08. Formulário para envio de perguntas

Basta que o usuário clique na seção *Dúvidas* na barra de navegação que ele será direcionado ao formulário

3.1.7 Prestar depoimento

Esta seção também é pública e permite que os usuários prestem depoimentos que serão acessíveis a todos que visitarem a aplicação. As Figuras 09 e 10 demonstram como funciona a seção de depoimentos. Basta o usuário selecionar *Depoimentos* na barra de navegação e clicar no botão para enviar o seu depoimento.

Figura 09. Página de depoimentos

Figura 10. Formulário para enviar um depoimento

3.1.8 Navegar pelas atividades

As atividades cadastradas no banco de dados terão acesso livre para todos que visitarem a aplicação. Ao selecionar a seção *atividades* na barra de navegação, o usuário será direcionado para a página de atividades. O usuário pode pesquisar por atividades específicas utilizando a barra de busca ou clicar em uma das atividades disponíveis na página. A Figura 11 mostra como é organizada a seção de atividades. Ao selecionar uma atividade específica, será redirecionado para a página contendo os detalhes da atividade, como imagem explicativa, descrição, doenças em foco, utensílios relacionados e observações. Estes detalhes podem ser conferidos na Figura 12.

Figura 11. Página para navegar pelas atividades do sistema

Figura 12. Página contendo os detalhes de uma atividade.

3.1.9 Navegar pelos utensílios

Os utensílios também possuem uma página própria, em que os usuários podem navegar por diversos itens e comidas que podem auxiliar no tratamento dos pacientes. Estes itens devem ser previamente cadastrados pelos profissionais de saúde do CER. Cada sub-seção de utensílios pode ser selecionada clicando em uma opção como mostrado na Figura 13. O usuário pode ainda clicar em cada item individualmente para ser direcionado para uma página mais detalhada do item, em que ele terá acesso a uma descrição mais elaborada sobre como utilizar o item, além de links úteis e um painel com atividades que o item possa ter alguma relação. Todos esses detalhes são apresentados na Figura 14.

Figura 13. Página para navegar pelos utensílios



Figura 14. Página de detalhes do utensílio.

4. ARQUITETURA

A arquitetura principal utilizada é dividida em duas sub-partes: Frontend (Cliente) e Backend (Servidor + Banco de Dados). A Figura 15 ilustra como estão separadas as duas partes principais da aplicação e como elas devem se comunicar. Este tipo de arquitetura é comumente conhecida como Cliente-Servidor [9], em que existem duas entidades principais, Cliente e Servidor, a primeira está em busca de algum tipo de informação e a segunda é responsável por fornecer estas informações.

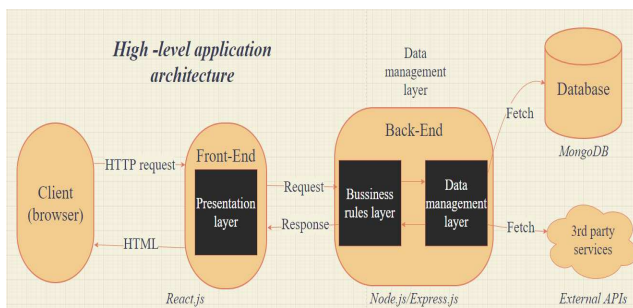


Figura 15: Diagrama da arquitetura da aplicação

Sempre que o Cliente necessitar uma informação do Servidor, será estabelecida uma comunicação entre os dois utilizando o protocolo HTTP [11]. Desta forma, o Cliente enviará requisições ao Servidor por meio de uma linguagem que seja compreendida por ambas as partes, e o Servidor será responsável por retornar as respostas apropriadas.

O Cliente será acessado por meio dos navegadores web modernos como Chrome e Edge, em que ele poderá enviar as requisições necessárias para obter as informações que precisa. As duas entidades irão se comunicar trocando dados no formato JSON (JavaScript-Object-Notation) [6], por ser largamente utilizado em sistemas modernos e por sua fácil legibilidade.

3.1 Front-end

O front-end deste trabalho foi implementado seguindo a arquitetura SPA (Single-page application) [10], largamente utilizada atualmente, que consiste em carregar um único

documento web para o usuário e atualizá-lo sempre que alguma ação for realizada, como busca de algum dado. Para isto, foi utilizada a linguagem Typescript[2] em conjunto com a biblioteca React [15], utilizada para criação de interfaces de usuário em aplicações web, amplamente utilizada por empresas e com uma grande comunidade. Além dessas tecnologias, foi utilizado um framework de CSS para a estilização dos componentes, TailwindCSS [18], pois ele ajuda bastante na produtividade ao facilitar alterações rápidas nos componentes. A Figura 16 apresenta a estrutura básica que foi definida para implementação do front-end, dividida em 4 sub-diretórios: *components*, *hooks*, *lib* e *pages*. Demais arquivos auxiliares foram omitidos. O diretório *lib* contém diversos métodos para tratar algumas das requisições feitas pelo Cliente ao servidor.

O frontend foi desenvolvido para ser acessado tanto pelos profissionais do CER quanto pelos familiares dos pacientes, utilizando uma interface *web* bem amigável e simples. O deploy do frontend foi feita utilizando o Vercel [19]

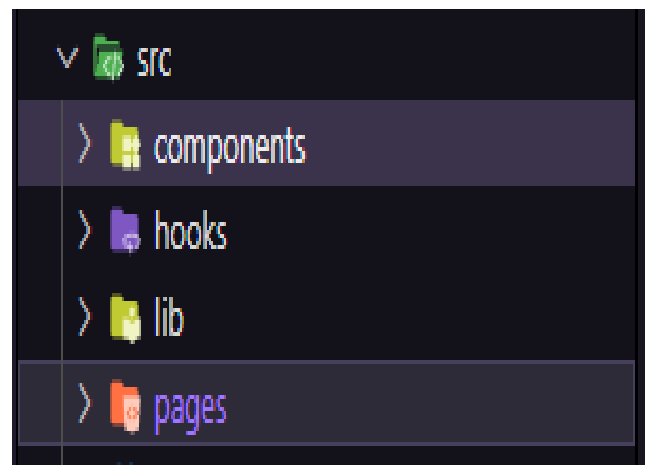


Figura 16. Estrutura dos diretórios front-end

3.2 Back-end

O back-end da aplicação é uma das partes englobadas pelo servidor. Ele funciona como um intermediário entre o cliente e o banco de dados. Assim, ele é responsável por realizar todas as operações padrões de acesso, recuperação, atualização, remoção e adição de dados ao banco, como também selecionar como e quais dessas informações serão repassadas ao cliente. Para a construção do back-end, foi utilizado o padrão REST API [16], que possui um design simples, moderno e fácil de ser implementado.

As tecnologias selecionadas para a implementação foram: Node.js [12] e Express.js [2], comumente utilizadas para se desenvolver REST API's atualmente. Além disso, foi utilizado o ORM Prisma [13] para comunicação com o banco de dados MongoDB. Para armazenar imagens, foi utilizada a técnica de conversão de imagens para Base64 [8], facilitando seu armazenamento no banco de dados em formato de texto. Para autenticação dos usuários, foi utilizado JWT (JSON Web Token) [5], que consiste em gerar um token especial para o usuário que realizar login, em que este token é enviado junto as requisições HTTP para que o back-end possa autenticar os pedidos e permitir ou não que alguma ação seja realizada, como cadastrar um novo

documento no banco. O deploy do backend foi feito com o Railway App [14].

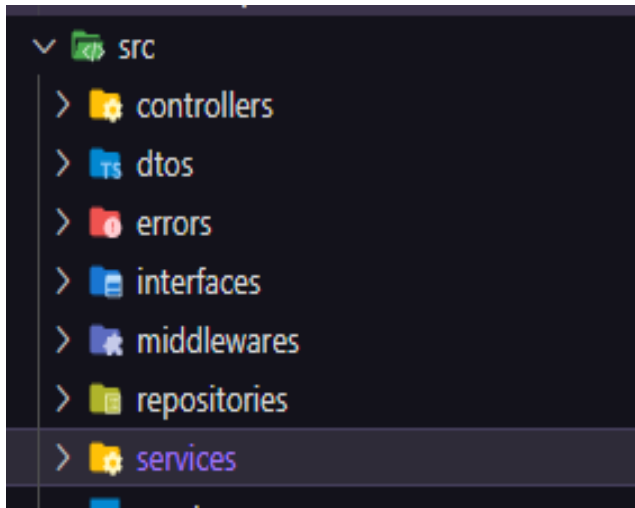


Figura 17. Estrutura dos diretórios back-end

A Figura 17 apresenta uma estrutura de diretórios bem comum para a implementação de REST API atualmente, contendo sete diretórios principais: *controllers*, *dtos*, *errors*, *interfaces*, *middlewares*, *repositories* e *services*.

Os *services* são responsáveis por realizar processamentos específicos de informações relativas a entidades presentes no banco de dados.

O *repositories* contém interfaces a serem implementadas pelos *services*. Esta solução foi aplicada pois permite que outros tipos de ORMs sejam utilizados, desde que obedçam às regras estabelecidas pelas interfaces.

O *middlewares* contém métodos que são utilizados para interceptar requisições e adicionar algum tipo de processamento extra como: validação de dados enviados, autenticação das requisições ou retorno de erros padronizados.

Já o diretório *interfaces* contém um conjunto de interfaces utilizadas por todo o projeto, como interface para criação de novos erros ou controladores.

O diretório *errors* possui novos tipos de erros para situações específicas que possam acontecer ao se enviar requisições ao servidor. Todo novo erro adicionado deve ser um padrão estabelecido por uma interface.

Os *dtos* são classes que representam as entidades do sistema e facilitam sua manipulação, desde a validação de novos dados recebidos pelo servidor, até os *services* que serão responsáveis por passar esses dados ao banco.

Por fim, os *controllers* são responsáveis por gerenciar as chamadas feitas a API, realizar algum tipo de processamento ou delegar para os *services*, e retornar as respostas adequadas para o cliente. No contexto de cada *controller* que ocorrerá a validação dos dados e a autenticação requisições, assim como o retorno dos erros adequados.

A Figura 18 apresenta, resumidamente, o fluxo que ocorre quando uma nova requisição chega ao servidor. O *endpoint*

representa as rotas disponíveis na API. O *controller* representa a entidade responsável por tratar as requisições enviadas para suas rotas previamente configuradas. Os *middlewares* podem ou não ser chamados, dependendo da requisição que foi realizada, e, ao final, os *services* alocados para cada *controller* irá realizar a comunicação propriamente dita com o banco de dados,

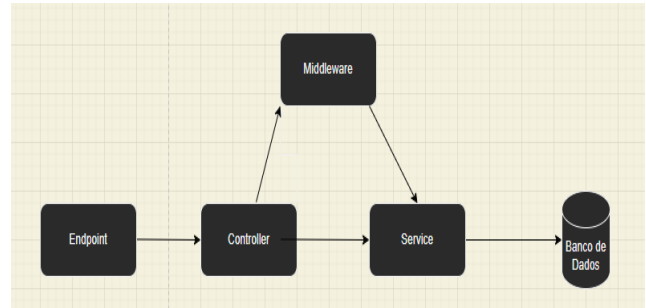


Figura 18. Diagrama ilustrando o fluxo de uma requisição ao servidor.

3.3 Banco de Dados

O banco selecionado para a persistência dos dados foi o MongoDB [7], por prévia experiência com a ferramenta e sua compatibilidade com o ORM Prisma utilizado no backend.

MongoDB é um banco de dados não relacional em que as entidades são separadas em Coleções de Documentos, em que cada entidade apresentada na Figura 19 será mapeada para uma coleção de documentos própria dentro do banco de dados. Entretanto, ainda que o MongoDB não suporte o relacionamento tradicional de tabelas, como em bancos relacionais (i.e PostgreSQL), ele fornece um tipo de relação por referência dos IDs dos documentos. Na Figura 05, as entidades *Items* e *Activities* compartilham uma relação *many-to-many*, em que os itens referenciam as atividades relacionadas por meio de seus ids e vice-versa. Como o MongoDB fornece um gerenciamento automático dos IDs por meio de um tipo especial *ObjectID*, um ID gerado automaticamente em formato textual pelo próprio banco, isso possibilita a criação de relacionamento entre as coleções de uma maneira fácil e rápida.

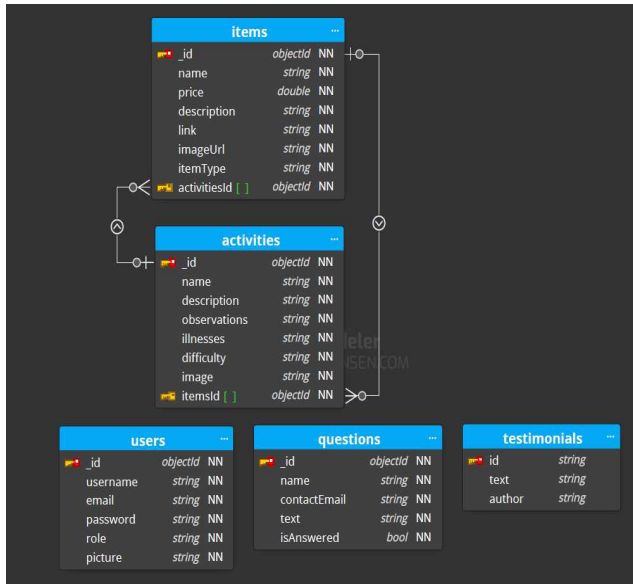


Figura 19. Entidades do banco de dados

5. VALIDAÇÃO

A etapa de validação passou por duas etapas. A primeira foi a realização de testes internos com a aplicação, na qual foram realizados uma bateria de testes automáticos utilizando a biblioteca Jest [4], largamente utilizada para automatização de testes para aplicações construídas com Javascript. Foram realizados testes automáticos no backend para validar os endpoints presentes na API cobrindo os seguintes requisitos:

- Retorno dos dados esperados;
- Retorno dos erros esperados;
- Utilização do método HTTP correto para aquele endpoint;
- Validação dos dados enviados dentro das requisições;
- Autenticação correta dos endpoints protegidos;

Os requisitos foram testados para cobrir todos os endpoints do backend. Para cada requisito, foram realizados 10 casos de testes automatizados, totalizando em 50 casos testes.

Para a segunda etapa de validação, foi realizada uma apresentação para três fisioterapeutas do centro de reabilitação, na qual elas realizaram um pequeno teste manual do AppCER e avaliaram os seguintes critérios:

- Controle e liberdade do usuário;
- Consistência e padrões
- Flexibilidade
- Visibilidade
- Facilidade
- Possibilidade de evolução

Todos esses critérios foram avaliados por meio de uma entrevista, feita de maneira remota. Alguns pontos que se destacaram e que mais impressionaram as fisioterapeutas são: o design simples e amigável da aplicação, a facilidade de adicionar e gerenciar os dados presentes na aplicação, como atividades e utensílios, o impacto que a utilização da ferramenta como suporte no acompanhamento dos pacientes pode trazer no longo prazo e a

possibilidade de evolução que o trabalho tem para outros profissionais do CER.

A apresentação para as fisioterapeutas foi essencial para medir o efeito positivo que a aplicação pode ter para facilitar a relação entre os pacientes e os profissionais, que é o objetivo principal deste trabalho. Todas afirmaram que o projeto vai facilitar a divulgação de informações que auxiliem os pais e familiares a fornecerem o melhor tratamento possível aos pacientes.

6. PROCESSO

O processo de desenvolvimento do trabalho se baseou na seguintes etapas:

1. Levantamento das funcionalidades necessárias.
2. Criação de um *backlog*.
3. Atribuição de prioridades para cada funcionalidade especificada.
4. Ordenação da implementação de cada funcionalidade.
5. Criação dos repositórios para *backend* e *frontend*.
6. Especificação das tecnologias a serem utilizadas.
7. Desenvolvimento da aplicação.
8. Testes e Validação da aplicação.
9. Deploy da aplicação.

Após esse processo, as funcionalidades consideradas essenciais para estarem presentes no MVP (Minimum Viable Product) foram as seguintes:

1. Cadastro dos dados: Atividades e Utensílios.
2. Autenticação dos profissionais de saúde
3. Atualização dos dados.
4. Cadastro de perguntas e depoimentos pelos usuários externos (pacientes e familiares).
5. Envio automático de emails de resposta feito pelos profissionais de saúde.
6. Recuperação dos dados para apresentação aos usuários externos.

Com isso, foram criados dois repositórios: um dedicado ao *backend* da aplicação e outro dedicado ao *frontend*. Ambos os repositórios estão hospedados no GitHub [3]. Com os repositórios criados, foi executada a etapa de definição das tecnologias para a implementação, todas citadas em seções anteriores.

Após essas etapas, foi iniciada a implementação da aplicação. Primeiramente, o *backend* foi desenvolvido, junto com a configuração do banco de dados. Com o *backend* pronto, deu-se início a implementação do *frontend*.

Com o desenvolvimento completo, foi iniciada a etapa de testes internos, utilizando bibliotecas para automatização deste processo. Além disso, uma pequena validação com integrantes do CER foi realizada para validar as funcionalidades especificadas anteriormente.

Por fim, foi realizado o deploy do *backend* e do *frontend* utilizando ferramentas distintas, pois ambas apresentaram benefícios específicos para cada parte do sistema. O *backend* foi hospedado utilizando-se Railway e o *frontend* utilizando o Vercel

5.1 Desafios

Durante a realização deste trabalho, dois grandes desafios tiveram que ser superados. Primeiramente, a falta de conhecimento e experiência com desenvolvimento de aplicações web. Foram

necessárias diversas horas para estudar as tecnologias citadas neste trabalho, além de padrões de projeto adequados e arquiteturas. Por causa disso, durante o desenvolvimento era necessário fazer pausas para revisar alguns conceitos e revisar as documentações. Porém, foi uma grande oportunidade para adquirir novos conhecimentos e permitir que ingressasse em uma área totalmente nova. O segundo desafio foi a dificuldade para conseguir preencher o banco de dados com informações reais, dado que essa tarefa depende bastante dos profissionais do CER se disponibilizarem para povoar o banco de dados. Entretanto, o trabalho foi aprovado por um pequeno grupo de fisioterapeutas do CER durante uma pequena apresentação e pode começar a ser utilizado.

que possibilitou que este trabalho fosse realizado. Para finalizar, gostaria de agradecer também a Tarsila, aluna do curso de Design da Universidade Federal de Campina Grande que teve um papel essencial na construção da identidade visual da aplicação e seu design como um todo.

7. CONSIDERAÇÕES FINAIS

O AppCER consegue facilitar a troca de informações entre os profissionais do CER e os pais e familiares dos pacientes. Por ser web, podendo ser acessada por qualquer navegador moderno, os pais podem obter informações de forma prática e rápida, evitando o deslocamento para o CER sempre que surgirem dúvidas, além de ser capaz de se comunicar com o centro enviando suas perguntas pela aplicação. Isso diminui a carga de trabalho dos profissionais do CER, permitindo que eles cadastrem dados confiáveis e úteis na plataforma, sem a necessidade de terem que atender presencialmente todos os pais. Desta forma, o AppCER conseguiu aprimorar essa relação entre pais e profissionais do CER, em que a aplicação é utilizada como um suporte para que os integrantes do CER possam auxiliar os pais dos pacientes fora do centro.

7.1 Trabalhos futuros

Um ponto que foi amplamente comentado durante a etapa de validação com os profissionais do CER foi a possibilidade de evolução do trabalho para uma versão mobile, pois na visão deles, isto facilitaria ainda mais o acesso dos pais e familiares dos pacientes. A versão web teria como foco de usuários os profissionais do CER, responsáveis por gerenciar os dados da aplicação e a versão mobile seria direcionada para o acesso aos dados pelo outro grupo de usuário. Além disso, surgiu a ideia de paulatinamente adicionar mais funcionalidades voltadas para os profissionais do centro, como a adição de um prontuário eletrônico para o CER de Campina Grande. Para finalizar, pode-se construir uma pesquisa que tenha como objetivo avaliar, de uma forma quantitativa e qualitativa, o impacto que a ferramenta trouxe para o dia-a-dia dos profissionais no CER e como a ferramenta os ajuda a melhorar o tratamento com os pacientes e familiares.

8. AGRADECIMENTOS

Primeiramente, gostaria de agradecer à minha família, que proporcionou todos os requisitos necessários para que eu pudesse me dedicar ao meu trabalho de conclusão de curso. À minha namorada, que sempre me apoiou durante todo o período que este trabalho estava sendo realizado. Gostaria também de agradecer ao meu orientador Wilkerson, por todo o acompanhamento e paciência. Gostaria de deixar registrado meu agradecimento à professora Isis, do departamento de Design por ter me auxiliado com as visitas ao CER e me apresentado à fisioterapeuta Rubênia,

9. REFERENCES

- [1] Agenda PB. 2019. CER de Campina Grande amplia de 7 mil para mais de 50 mil atendimentos por ano. Retrieved Agosto 8, 2022 from <https://agendapb.com.br/cer-de-campina-grande-amplia-de-7-mil-para-mais-de-50-mil-atendimentos-por-ano>.
- [2] Express.js [n. d.]. Framework web rápido, flexível e minimalista para Node.js. Retrieved Agosto 8, 2022 from <https://expressjs.com/pt-br/>.
- [3] GitHub [n. d.]. Let's build from here. Retrieved Agosto 8, 2022 from <https://github.com/about>.
- [4] Jest. [n. d.]. Jest é um poderoso Framework de Testes em JavaScript com um foco na simplicidade. Retrieved Agosto 8, 2022 from <https://jestjs.io/pt-BR/>.
- [5] JWT. [n. d.]. Introduction to JSON Web Tokens. Retrieved Agosto 8, 2022 from <https://jwt.io/introduction>.
- [6] JSON. [n. d.]. Introducing JSON. Retrieved Agosto 8, 2022 from <https://www.json.org/json-en.html>.
- [7] MongoDB. [n. d.]. Welcome to the MongoDB Documentation. Retrieved Agosto 8, 2022 from <https://www.mongodb.com/docs/>.
- [8] MDN Web Docs. [n. d.]. Base64. Retrieved Agosto 8, 2022 from <https://developer.mozilla.org/en-US/docs/Glossary/Base64>.
- [9] MDN Web Docs. [n. d.]. Client-Server Overview. Retrieved Agosto 8, 2022 from https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Client-Server_overview.
- [10] MDN Web Docs. [n. d.]. SPA (Single-page application). Retrieved Agosto 8, 2022 from <https://developer.mozilla.org/en-US/docs/Glossary/SPA>.
- [11] MDN Web Docs. [n. d.]. Uma visão geral do HTTP. Retrieved Agosto 8, 2022 from <https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview>.
- [12] Node.js. [n. d.]. About Node.js. Retrieved Agosto 8, 2022 from <https://nodejs.org/en/about/>.
- [13] Prisma [n. d.]. Next-generation Node.js and TypeScript ORM. Retrieved Agosto 8, 2022 from <https://www.prisma.io>.
- [14] Railway App. [n. d.]. Bring your code, we'll handle the rest. Retrieved Agosto 8, 2022 from <https://railway.app>.
- [15] React. [n. d.]. React, Uma biblioteca JavaScript para criar interfaces de usuário. Retrieved Agosto 8, 2022 from <https://pt-br.reactjs.org>.
- [16] Red Hat. 2020. O que é API REST? Retrieved Agosto 8, 2022 from <https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>.
- [17] TypeScript. [n. d.], TypeScript for the New Programmer. Retrieved Agosto 8, 2022 from <https://www.typescriptlang.org/docs/handbook/typescript-from-scratch.html>.
- [18] TailwindCSS [n. d.]. Rapidly build modern websites without ever leaving your HTML. Retrieved Agosto 8, 2022 from <https://tailwindcss.com>.
- [19] Vercel [n. d.]. Introduction to Vercel. Retrieved Agosto 8, 2022 from <https://vercel.com/docs>.