

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

SAMUEL LEIVANS DOS SANTOS BOEIRA

**IMPLEMENTAÇÃO E ATUALIZAÇÃO DE FRONTEND PARA O SISTEMA
SOCIAL CARE**

GUARAPUAVA

2023

SAMUEL LEIVANS DOS SANTOS BOEIRA

**IMPLEMENTAÇÃO E ATUALIZAÇÃO DE FRONTEND PARA O SISTEMA
SOCIAL CARE**

Frontend implementation and update for the Social Care system

Proposta de Trabalho de Conclusão de Curso de Graduação apresentado à disciplina de Trabalho de Conclusão de Curso 1, do Curso Superior de Tecnologia em Sistemas para Internet da Universidade Tecnológica Federal do Paraná, Campus Guarapuava, como requisito parcial para obtenção do título de Tecnólogo em Tecnologia em Sistemas para Internet.

Orientador: Prof. Dr. Andres Jessé Porfirio

Coorientador: Prof. Dr. Emerson Fedechen

GUARAPUAVA

2023



[4.0 Internacional](https://creativecommons.org/licenses/by/4.0/)

Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

LISTA DE ABREVIATURAS E SIGLAS

Siglas

API	<i>Application Programming Interface</i>
PPLs	Pessoas Privadas de Liberdade
PR	<i>Pull Request</i>
SSR	<i>Server-side Rendering</i>
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1	INTRODUÇÃO	3
1.1	Considerações iniciais	3
1.2	Objetivos	4
1.2.1	Objetivo geral	4
1.2.2	Objetivos específicos	4
1.3	Justificativa	4
2	PROTÓTIPO SOCIAL CARE	6
2.0.1	Funcionamento	6
2.0.2	Limitações e construção da <i>Application Programming Interface</i> (API)	7
2.0.3	Limitações técnicas da interface	8
3	MATERIAIS E MÉTODOS	9
3.1	Materiais	9
3.1.1	GitHub	9
3.1.2	NodeJS e ReactJS	9
3.1.3	NextJS e Mantine	9
3.1.4	Playwright e testes end-to-end	10
3.2	Métodos	10
3.2.1	Prototipagem e Design Thinking	10
3.2.2	Processo de desenvolvimento	10
4	CONCLUSÃO	12
	REFERÊNCIAS	13

1 INTRODUÇÃO

Esta seção inicial apresenta brevemente o domínio da aplicação.

1.1 Considerações iniciais

A assistência social é uma política pública que tem como objetivo fornecer orientação, recursos e apoio para aqueles que precisam de algum amparo, facilitando o acesso a programas e serviços sociais. Além disso, atua como defensor dos direitos humanos, trabalhando para garantir que as pessoas sejam tratadas com dignidade (GOV.BR, 2023). Neste contexto, os conselhos das comunidades, são um dos órgãos da Execução Penal, regulado pela Lei 7.210, de 11/07/1984, Lei de Execução Penal (TJDFT, 2016), e representa a real possibilidade de intervir nas relações sociais dentro e fora da prisão, mas além de apoio às varas de execução penal, os conselhos prestam também assistência a comunidade carcerária, seus familiares e comunidade em geral que se refere ao serviço, no intuito de reduzir os impactos dos danos causados pelo cárcere (TALON, 2017).

O Serviço Social no Conselho da Comunidade tem como objetivo promover o acesso aos recursos sociais das Pessoas Privadas de Liberdade (PPLs) e seus familiares, principalmente no que se refere à Previdência, à Lei de Execução Penal e à Assistência Social. Para tanto, o profissional de Serviço Social deve conhecer as expressões da questão social (dificuldade de acesso à saúde, descriminalização de raça e gênero, relacionamento familiar, casos de violência, dentre outros) pertinentes à população assistida, propondo programas, projetos e serviços que façam frente às demandas identificadas nos atendimentos, garantindo assim que todo e qualquer tipo de assistência seja garantida às PPLs (PRZYTOCKI, 2023).

No conselho da comunidade da comarca de Guarapuava, é atendida uma grande demanda de PPLs que necessitam deste tipo de apoio. No entanto, a instituição enfrenta alguns desafios para gerenciar os atendimentos de forma satisfatória, pois estes atendimentos são realizados utilizando recursos como, arquivos de texto, imagens e outros tipos de arquivos como PDF, armazenados localmente em um computador ou enviados para um serviço de hospedagem em nuvem, o que dificulta o controle, a organização, e a atualização destas informações.

Uma possível solução, é a criação de um sistema *web* que forneça as funcionalidades que substituam a forma atual que os assistentes sociais gerenciam os atendimentos, e que ofereça recursos adicionais como a geração de relatórios, sistema de busca, etc. No primeiro semestre de 2023, o protótipo do “Social Care” foi desenvolvido na disciplina de Sistemas Distribuídos da Universidade Tecnológica Federal do Paraná (UTFPR), com as funcionalidades principais de gerenciar o cadastro de PPLs e atendimentos. Contudo, por se tratar de um protótipo, o sistema não oferece estes recursos, diante disso, optou-se por desenvolver um sistema a partir do protótipo anterior.

Este trabalho propõe o desenvolvimento do *frontend* deste novo sistema, tendo em vista que a interface seja fácil de usar e forneça uma experiência de usuário produtiva, por este motivo, a aplicação será projetada levando em consideração as necessidades dos assistentes sociais. Além disso, também é importante ressaltar que a interface deve ser responsiva e possuir características de *Design System* como ser consistente e seguir um padrão de semântica. E ainda, a interface deve ser intuitiva para permitir ver as informações de maneira eficiente e garantir a fluidez do trabalho diário.

1.2 Objetivos

Esta seção descreve os objetivos para o desenvolvimento do projeto separados entre 1.2.1 Objetivo geral e 1.2.2 Objetivos específicos.

1.2.1 Objetivo geral

Implementar o *frontend* do sistema Social Care, atualizando a interação e usabilidade testadas no protótipo disponível.

1.2.2 Objetivos específicos

- Atualização do *stack* padrão de criação de projetos e roteamento de páginas do *React* para o uso do *framework NextJS*;
- Adaptação do *frontend* para se conectar a uma API de persistência de dados em substituição ao *Firebase*;
- *Design* e implementação de novas páginas;
- Implementação de testes *end-to-end* para os principais fluxos da interface.

1.3 Justificativa

O avanço da *internet* atualmente permite a criação de sistemas que tragam facilidades, automações, entre outros benefícios. E o oposto acontece com a falta de uma aplicação *web*, como dificultar o trabalho e aumentar as chances de erro, principalmente em um ambiente que dependa de trabalhar com uma grande quantidade de dados. É o que acontece no conselho da comunidade, essa falta dificulta o controle de casos, e a consulta por informações de atendimentos que já ocorreram.

Embasado neste problema, o sistema Social Care foi prototipado e será desenvolvido com uma nova estrutura e melhorias com base na coleta de informações de usabilidade desta

versão inicial. Esta atualização permitirá também que os usuários adaptem os formulários sempre que necessário, permitindo que os assistentes sociais gerenciem perguntas, ou seja, terão controle sobre quais perguntas irão compor o formulário, tornando-o mais flexível e coerente com as atividades dos utilizadores. Essa mudança no formato das perguntas facilitará a análise e a tomada de decisão dos assistentes sociais e irá colaborar para a melhoria da qualidade dos serviços, além de oferecer uma melhor portabilidade da aplicação para outros fins, contribuindo também para a comunidade de desenvolvimento de software.

2 PROTÓTIPO SOCIAL CARE

Este capítulo irá descrever o funcionamento do protótipo do sistema e apresentará suas limitações.

2.0.1 Funcionamento

Uma das tarefas que são realizadas no conselho da comunidade é cadastrar os dados pessoais das PPLs como nome, data de nascimento, documentos de identificação, contato, parentesco e dentre outras como, endereço e registrar se a pessoa possui alguma dependência química. Além dos dados pessoais, os assistentes sociais também registram as informações sobre o atendimento como qual a demanda que as PPLs estão buscando. Por exemplo, no caso da pessoa precisar de um atendimento médico, é papel dos assistentes sociais fazer esta intermediação entre as PPLs e o encaminhamento médico, como pode ser observado na Figura 1.

Figura 1 – Processo de atendimentos no Conselho da Comunidade.



Fonte: Adaptado de Diagrama BR (2023).

No protótipo estes cadastros se dão por meio de dois formulários, um para cadastrar as PPLs e outro para cadastrar os atendimentos que contém as demandas. Pois além da demanda citada, as demandas podem ser várias outras como o desejo de obter contato com a família. Neste caso, o protótipo também possui uma página para configurar as demandas e permite que os assistentes sociais cadastrem e controlem as demandas disponíveis no formulário do

atendimento, podendo ser escolhida uma ou mais para um mesmo atendimento, como pode ser observado na Figura 2.

Figura 2 – Cadastro de atendimentos no protótipo do sistema.

Cadastrar atendimento

✓ Fulano Silva admin@example.com

Campos com asteriscos * são obrigatórios

Data e hora *

12/10/2023 20:53

Demandas *

Odontologia

Contato e Atendimento Familiar

Outra

Encaminhamento

Fonte: Projeto MVP Social Care (2023).

2.0.2 Limitações e construção da API

Tanto o formulário de cadastro de PPLs quanto o formulário de atendimentos do protótipo possuem questões específicas para o fluxo de trabalho no conselho da comunidade, o que limita o sistema para uso apenas em instituições com um padrão de atendimento igual ou muito semelhante.

Além desta limitação, para o uso do *Firebase* é necessário configurar o sistema vinculando a uma conta *Google*, o que atribuirá responsabilidade sobre o indivíduo dono desta conta para manter a conta e os dados em segurança. E como o protótipo se prende ao *Firebase*, não é possível hospeda-lo em um servidor em que as instituições tenham disponibilidade, pois por se tratar de serviços públicos, as instituições podem ter disponível um servidor fornecido pelo governo para hospedagem do sistema e armazenamento destes dados, tornando a estrutura de

uma API melhor de ser implementada nestas instituições e em diferentes servidores. Mais detalhes da API podem ser encontrados no projeto que está sendo desenvolvido separadamente intitulado Atualização do *Backend* do Sistema Open Social Care: Migrando da arquitetura *serverless* para uma API em *Laravel* (SOUZA, 2023).

2.0.3 Limitações técnicas da interface

Com o desenvolvimento da API, o *frontend* terá que ser reformulado e reconstruído para funcionar nesta nova estrutura, mas algumas melhorias técnicas também podem ser realizadas nesse processo, como substituir o uso da ferramenta *create-react-app* que já é considerada defasada pela comunidade de desenvolvimento e na própria documentação do *React* é mencionado como alternativa fazer o uso de um *framework* como *NextJS* para construir a aplicação. Além disso, *frameworks* trazem algumas implementações que não existem no *React* puro, como é o caso do roteamento de páginas baseado em arquivo que o *NextJS* traz nativamente, possibilitando que as rotas sejam definidas baseadas nos arquivos que são criados na pasta destinada para as páginas.

O *NextJS* também implementa *Server-side Rendering* (SSR), o que permite renderizar os componentes compatíveis no lado do servidor e com isso conseguir uma velocidade maior de renderização, mas o *Mantine*, na versão que é utilizada no protótipo não é compatível com este recurso. Entre tanto, recentemente foi publicada a versão 7 deste *framework* que suporta SSR e funciona com o *NextJS*. Ao utilizar a versão mais atualizada da ferramenta, esta migração de versão não precisará ser realizada no futuro e a aplicação poderá contar com este ganho de desempenho de SSR já na fase de implementação.

3 MATERIAIS E MÉTODOS

A finalidade deste capítulo é descrever as principais ferramentas e metodologias utilizadas para alcançar o objetivo do trabalho. Este capítulo está subdividido em duas seções, 3.1 Materiais e 3.2 Métodos.

3.1 Materiais

Essa seção apresenta as tecnologias e ferramentas que serão utilizadas para o desenvolvimento da aplicação frontend.

3.1.1 GitHub

O *GitHub* é uma plataforma de hospedagem que permite aos desenvolvedores armazenar repositórios *Git* na nuvem, trabalhar em equipe e colaborar com outros projetos. Além disso, também oferece recursos como controle de acesso, solicitações de *pull*, e revisão de código (ARAUJO, 2023). Também é possível gerenciar atividades com o *GitHub Projects* (GITHUB, 2023) e automatizar tarefas como testes e *deploys* com *GitHub Actions* (MATOS, 2020).

3.1.2 NodeJS e ReactJS

Node é uma plataforma de código aberto que permite a execução de código *JavaScript* fora de um navegador que pode ser executada em diferentes sistemas operacionais. E *React* é uma biblioteca *frontend JavaScript* de código aberto com foco em criar interfaces de usuário em páginas *web* estáticas. A combinação de *Node* e *React* permite que os desenvolvedores criem sistemas dinâmicos e que executem em vários dispositivos. Também permite escalabilidade ao trabalhar em grandes projetos e manter o desempenho do site (MISHRA, 2023).

3.1.3 NextJS e Mantine

NextJS e *Mantine* são *frameworks* que visam otimizar o desenvolvimento de aplicações *React*. O *NextJS* fornece uma melhor experiência de desenvolvimento para a construção da aplicação com compilações mais rápidas e melhora alguns aspectos do projeto como roteamento de páginas baseado em sistema de arquivo, otimização de imagens, e capacidade de renderização híbrida tanto no lado do cliente quanto do servidor (RAVOOF, 2023). Já o *Mantine* é uma biblioteca de componentes *React* responsivos pré-fabricados, projetados e construídos pelos mantenedores e pela comunidade *Mantine* (MANTINEDEV, 2023).

3.1.4 Playwright e testes end-to-end

A playwright é uma biblioteca que permite a criação de testes *end-to-end* (MICROSOFT, 2023). Os testes *end-to-end* são utilizados para testar se o fluxo de um aplicativo está sendo executado conforme o esperado. O objetivo da realização de testes *end-to-end* é identificar dependências do sistema e garantir que a informação certa seja passada entre vários componentes do sistema simulando o ambiente real e as interações entre os usuários (MARQUES, 2023).

3.2 Métodos

Nesta seção serão apresentados os métodos que contemplam o desenvolvimento deste projeto.

3.2.1 Prototipagem e Design Thinking

A prototipagem é uma estratégia eficaz para validar uma ideia antes de sua implementação. Como o protótipo do projeto já foi desenvolvido, o próximo passo é coletar *feedback* dos usuários. Neste caso, as informações serão obtidas diretamente com os assistentes sociais por meio de reuniões, seguindo alguns elementos da metodologia *Design Thinking*.

O *Design Thinking* é uma metodologia que orienta o desenvolvimento de produtos e serviços. Ela envolve várias etapas, como compreender as necessidades dos usuários, gerar e organizar ideias, além de desenvolver e testar soluções. Esta metodologia se baseia em alguns princípios fundamentais. Um destes princípios é o princípio da empatia, que será cultivada durante estas reuniões.

Este elemento do *Design Thinking* nos permite entender melhor as necessidades e experiências dos usuários, garantindo que as soluções desenvolvidas sejam verdadeiramente eficazes e significativas para quem irá utilizar a aplicação. Uma vez entendido algo como necessidade dos usuários, essa necessidade será convertida em um ou mais requisitos que o sistema terá que implementar.

3.2.2 Processo de desenvolvimento

Para este projeto, as atividades identificadas serão adicionadas à *cards* no projeto *Kanban* do repositório como *“issues”*. Estes *cards* são representativos visuais que conterão uma descrição das tarefas a serem elaboradas.

Quando uma atividade for planejada, com base nessa *issue* será iniciado o desenvolvimento das tarefas com a criação de uma nova *branch* a partir da *branch main* do *GitHub*. Nessa

nova *branch* é onde todo o código para essa tarefa específica será escrito como, novas páginas, novos componentes, arquivos e modificações. Essas mudanças também serão atualizadas no planejamento para relacionar o que foi feito.

Uma vez que o código é escrito, além de testar manualmente, testes automatizados do fluxo de interface referente a atividade também serão escritos. Em seguida, estes testes serão executados em ambiente de desenvolvimento para garantir que tudo está funcionando como esperado. Finalizando a criação dos testes necessários, o código é considerado pronto para revisão e uma *Pull Request* (PR) desta *branch* é iniciada.

A revisão do código é um passo crucial para garantir o funcionamento e a qualidade. Se a revisão for bem-sucedida, o código é mesclado na *branch main* e na *branch staging*, uma atualização na *branch staging* irá disparar um *deploy* em um servidor configurado, o que permitirá testar o código em um ambiente que é o mais próximo possível do ambiente de produção.

Finalmente, se nenhum problema for encontrado no ambiente de *staging*, o código é mesclado na *branch production*, com isso, uma *action* será disparada que irá executar novamente os testes da interface para garantir o funcionamento, em seguida o código será implantado no ambiente de produção. E assim, o ciclo de desenvolvimento será concluído.

Ainda se algum problema for identificado em qualquer momento e que não esteja relacionada com uma atividade em andamento, será criado uma nova *issue* identificada como “*Fix*” para resolver os problemas encontrados.

4 CONCLUSÃO

O trabalho apresentado foca no desenvolvimento de uma interface de usuário para complementar a infraestrutura de *backend* da atualização do sistema Social Care, que irá adicionar novos recursos como buscas e relatórios.

Frente a esta proposta, também foi apresentado uma abordagem de perguntas personalizáveis nos formulários dos atendimentos, onde a aplicação permitirá que as instituições moldem a interface do usuário de acordo com suas necessidades específicas. Esta estratégia aumentará a aplicabilidade do sistema para uma gama maior de instituições e casos de uso, consequentemente, ampliando a relevância e o impacto deste trabalho.

Além disso, melhorias técnicas como testes automatizados de interface serão implementadas no *frontend*. A expectativa é que os testes aumentem significativamente a capacidade de identificar falhas na aplicação, resultando em um cenário de manutenção mais efetivo.

REFERÊNCIAS

ARAUJO, E. **A importância do Git e GitHub**. 2023. Disponível em: <https://www.dio.me/articles/a-importancia-do-git-e-github>. Acesso em: 04 out. 2023.

Diagrama BR. **Estruturação de áreas de Informática - Tecnologia da Informação**. 2023. Disponível em: <https://www.diagramabr.com.br/administracao/atuacao/estruturacao-de-areas-de-informatica---tecnologia-da-informacao>. Acesso em: 20 out. 2023.

GITHUB, I. **Sobre Projects**. 2023. Disponível em: <https://docs.github.com/pt/issues/planning-and-tracking-with-projects/learning-about-projects/about-projects>. Acesso em: 05 out. 2023.

GOV.BR. **A Assistência Social**. 2023. Disponível em: <https://www.gov.br/mds/pt-br/acoes-e-programas/assistencia-social>. Acesso em: 08 out. 2023.

MANTINEDEV. **About Mantine**. 2023. Disponível em: <https://mantine.dev/about/>. Acesso em: 23 mar. 2023.

MARQUES, R. **Teste end-to-end**. 2023. Disponível em: <https://www.cedrotech.com/blog/teste-end-to-end>. Acesso em: 04 out. 2023.

MATOS, G. **Automação de Processos com GitHub Actions**. 2020. Disponível em: <https://glerystonmatos.medium.com/automa%C3%A7%C3%A3o-de-processos-com-github-actions-3efdeaf4919c>. Acesso em: 04 out. 2023.

MICROSOFT. **Playwright enables reliable end-to-end testing for modern web apps**. 2023. Disponível em: <https://playwright.dev/>. Acesso em: 01 jun. 2023.

MISHRA, B. **React and NodeJS: A Deadly Combination for Web Application Development**. 2023. Disponível em: <https://www.mindinventory.com/blog/react-with-nodejs-for-full-stack-development/#:~:text=Yes,-,Node.,web%20servers%20for%20React%20apps>. Acesso em: 04 out. 2023.

Projeto MVP Social Care. **Projeto MVP Social Care**. 2023. Disponível em: <https://github.com/andresjesse/sd-mvp-social-care>. Acesso em: 30 out. 2023.

PRZYTOCKI, L. S. **Sistematização da prática profissional do conselho da comunidade da comarca de guarapuava/pr**. Universidade Estadual do Centro Oeste - Unicentro. 2023.

RAVOOF, S. **Next.js vs React? É uma Parceria, Não uma Competição**. 2023. Disponível em: <https://kinsta.com/pt/blog/nextjs-vs-react/#:~:text=Os%20aplicativos%20React%20tradicionalmente%20rodam,js>. Acesso em: 04 out. 2023.

SOUZA, C. E. D. **Atualização do Backend do Sistema Open Social Care: Migrando da arquitetura serverless para uma API em Laravel**. 2023. Disponível em: https://tcc.tsi.pro.br/uploads/academic_activity/pdf/199/GP_COINT_2023_2_CAMILA_EMANUELE_DE_SOUZA_PROPOSTA.pdf. Acesso em: 22 out. 2023.

TALON, E. **O Conselho da Comunidade e a execução penal**. 2017. Disponível em: <https://www.jusbrasil.com.br/artigos/o-conselho-da-comunidade-e-a-execucao-penal/519670378>. Acesso em: 10 out. 2023.

TJDFT, T. de Justiça do Distrito Federal e dos T. **Lei de Execuções Penais**. 2016. Disponível em: <https://www.tjdft.jus.br/institucional/imprensa/campanhas-e-produtos/direito-facil/edicao-semanal/lei-de-execucoes-penais>. Acesso em: 10 out. 2023.