



PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

CÉSAR PEREIRA DOS SANTOS FILHOS
GABRIEL ALEXANDRE MEDRADO ALVES
LEONARDO SILVA JAQUES
TÁSSIA DE LIMA TEIXEIRA
THIAGO ALMEIDA AMORIM

PROJETO: DESENVOLVIMENTO DE APLICAÇÃO WEB FRONT-END

Web site para a organização não governamental O bem nunca para

Belo Horizonte

Abril, 2022

CÉSAR PEREIRA DOS SANTOS FILHOS
GABRIEL ALEXANDRE MEDRADO ALVES
LEONARDO SILVA JAQUES
TÁSSIA DE LIMA TEIXEIRA
THIAGO ALMEIDA AMORIM

PROJETO: DESENVOLVIMENTO DE APLICAÇÃO WEB FRONT-END

Web site para a organização não governamental O bem nunca para

Primeira etapa do projeto apresentado
no curso de Análise e
Desenvolvimento de Sistemas da
Pontifícia Universidade Católica de
Minas Gerais.

Orientadora: Daniella Carvalho
Monteiro Ferreira

Belo Horizonte

Abril, 2022

SUMÁRIO

1. INTRODUÇÃO	4
Problema	4
Objetivos	4
Justificativa	5
Público-alvo	5
Personas	6
Histórias de usuário	7
Requisitos do projeto	8
Requisitos Funcionais	8
Requisitos Não Funcionais	8
Restrições	8
3. METODOLOGIA	9
Relação de ambiente de trabalho	9
Gestão de código fonte	9
Gerenciamento do Projeto	9
4. PROJETO DE INTERFACE	11
Fluxo de Usuário	11
Wireframes	12
5. ARQUITETURA DA SOLUÇÃO	15
Diagrama de Componentes	15
Hospedagem	15
6. TEMPLATE DO SITE	16
7. FUNCIONALIDADES DO SISTEMA	20
Homepage (RF-01)	20
Contato (RF-02)	21
8. PLANO DE TESTES DE SOFTWARE	27
Caso de Teste	27
CT-01 – Funcionalidade dos botões do menu	27
Caso de Teste	27
CT-02 – Funcionamento do formulário de contato	27
9. REGISTRO DE TESTES DE SOFTWARE	28
Registros de testes para CT-01	28
Registros de testes para CT-02	29

1. INTRODUÇÃO

Uma organização não governamental (ONG) é uma instituição que não possui vínculo com nenhum governo, portanto é privada. Elas atuam no âmbito social, não visando obter lucro com seu trabalho. Todas as entidades que não objetivam lucro são consideradas do terceiro setor da economia.

A ONG o Bem nunca para esta situada na cidade de Belo Horizonte (MG) e desenvolve trabalho social na região onde está inserida. Oferece consultas com diversas especialidades, como psicólogos, fonoaudiólogos e nutricionistas. Disponibiliza serviço jurídico, creche e distribui cestas básicas para a comunidade. Seu trabalho é amplo e com impacto positivo na sociedade local.

Com a tecnologia cada vez mais presente na sociedade, modificando a forma como as pessoas vivem e se relacionam, se faz necessário que os mais diversos locais passem a utilizar as redes como forma de apresentar seu trabalho, inclusive aqueles do terceiro setor. Cada vez mais as pessoas buscam na internet serviços e também oferecer apoio as ONGs.

Diante desse quadro, a ONG o Bem nunca para, sendo um local que oferece serviços, necessita de um canal eficiente de agendamentos. A instituição também conta com a colaboração de voluntários para realizarem atividades, além disso, recebe donativos, sendo imprescindível um local para oferecer informações seguras sobre seu trabalho e suas demandas.

Contudo, esse projeto visa atuar para beneficiar a ONG O Bem nunca para através do desenvolvimento de uma aplicação web front-end, com a finalidade de apresentar informações da ONG e tornar possível o agendamento de consultas para a unidade localizada no bairro Lajedo.

Problema

Como podemos dinamizar o agendamento de atendimentos da ONG O bem nunca para?

Objetivos

O objetivo geral desse projeto é tornar eficiente o processo de marcação de atendimentos da ONG O bem nunca para.

Destacamos os seguintes objetivos específicos:

- Identificar melhorias no atual sistema de agendamento da ONG;
- Desenvolver um web site que apresente informações sobre a instituição, fornecidas pelos seus responsáveis;
- Criar um formulário de contato com a instituição;

Justificativa

A ONG O bem nunca para realiza um amplo trabalho social com impacto positivo na comunidade em que está localizada. Apesar disso possui redes sociais com defasagem de conteúdo, apresentando uma parte pequena do trabalho que é realizado no local. É necessário o compartilhamento de maiores informações confiáveis da instituição, possibilitando que possíveis voluntários e pessoas que desejam realizar a ONG a conheçam.

Atualmente seus agendamentos são realizados por WhatsApp, ligações ou pessoalmente. Automatizar esse processo, por meio de um web site, seria de grande valia para a instituição, por possibilitar que os voluntários responsáveis pelo agendamento tenham maior disponibilidade de tempo para realizar outras tarefas.

Desenvolvemos esse projeto visando beneficiar um local específico, por acreditar que em breve ele possa ser utilizado pela instituição em que focamos e seus usuários.


Público-alvo

O público-alvo atendido pela ONG possui ampla faixa etária, indo de crianças até idosos e possuem as mais diversas ocupações. Embora a ONG seja voltada para a comunidade local, pessoas de vários bairros se deslocam até ela para utilizar seus serviços.

No entanto, considerando que as crianças possuem responsáveis legais e os idosos, normalmente, contam com apoio de adultos. Definimos como público-alvo desse projeto homens, mas principalmente mulheres, dentre 30 e 40 anos de idade que não possuem plano de saúde e dispõe de pouca renda para tratamentos.

2. ESPECIFICAÇÃO DO PROJETO

Personas

		Rodrigo Santos Coelho Filho	
Idade: 38 anos Ocupação: Motorista de aplicativo		Aplicativos: Instagram Facebook.	
Motivações: Sua família: filha e esposa.	Frustrações: Tempo escasso e falta de dinheiro.	Hobbies: Gosta de jogar futebol com os amigos. História: Nasceu em uma periferia de Minas Gerais, chamada Lajedo, um local onde tem carência de estabelecimentos, segurança e serviços. Rodrigo teve sua filha cedo, aos 19 anos, então desde sempre teve que se adaptar com grandes responsabilidades. Concluiu o Ensino Médio. Aos 19 anos, junto de sua mulher, com seu salário de 1100 reais, sustentavam a casa e ajudavam os pais com contas. Por esse motivo, acabou adquirindo dívidas, então ele foi em busca de um novo trabalho, com um salário melhor. Conseguiu ocupar uma vaga de motorista, onde atua até atualmente. Para complementar a renda Rodrigo sacrifica seus fins de semana e seu tempo livre para ser motorista de aplicativo. A mulher de Rodrigo se chama Clara Luise trabalha com marketing/comunicação para empresas focadas em venda de cursos e serviços.	

		Fernanda Lima de Carvalho	
		Idade: 39 anos Ocupação: Vendedora (loja de roupas)	Aplicativos: Instagram Facebook.
Motivações: Seus dois filhos.	Frustrações: Falta de dinheiro e baixa escolaridade.	Hobbies: Ama plantas e gosta de sair com os filhos.	

		<p>História:</p> <p>Nasceu em Pequi, uma pequena cidade de Minas Gerais. Aos 21 anos casou-se e mudou-se para o bairro Lajedo, na cidade de Belo Horizonte, local com uma deficiência na infraestrutura e suporte à população. Fernanda teve seu primeiro filho aos 25 anos e o segundo aos 27 anos. Se separou, aos 30 anos, e obteve a guarda total deles. Pela situação financeira e por sempre se dedicar ao casamento e aos filhos, ela lamenta não ter seguido com os estudos, tendo concluído somente o Ensino Médio. Desde sua mudança para Belo Horizonte, Fernanda trabalha como vendedora em lojas de departamentos e de vestuário.</p> <p>Aos 39 anos, Fernanda recebe um salário fixo de 1.230 reais e uma renda variável de comissão pelas suas vendas, que gira em torno de 600 reais. Vive com seus filhos em uma casa alugada e recebe 650 reais de pensão alimentícia para as crianças. Mesmo com uma renda restrita, ela se dedica a fornecer suporte para que seus filhos complementem os estudos com cursos e atividades extra curriculares.</p>
--	--	--

Histórias de usuário

Eu como ... [PERSONA]	... quero/desejo ... [O QUE]	... para ... [POR QUE]
Fernanda Lima de Carvalho	Desejo realizar marcações com mais facilidade	existem poucos voluntários na parte de atendimento focado em marcação e, por consequência, demora muito para o retorno
Fernanda Lima de Carvalho	Desejo ter acesso aos profissionais	para que eu possa escolher de acordo com a formação/experiência do profissional.
Rodrigo Santos Coelho Filho	Desejo poder visualizar os horários livres	para poder me programar para marcar um atendimento.
Rodrigo Santos Coelho Filho	Desejo poder ver todos os serviços oferecidos	para que eu veja o que será prioridade para mim no momento

Requisitos do projeto

Requisitos Funcionais

O sistema deve, principalmente, criar formulário para marcação de consultas e por consequência facilitar tanto para o usuário, quanto para o voluntário.

Interface

O sistema será formado por abas, sendo elas: homepage; quem somos; nossa história; contato; localização; como doar e; parceiros.

Requisitos funcionais da interface

ID	Descrição	Prioridade
RF-01	Menu fixo	Média
RF-02	Formulário: Entre em Contato	Alta
RF-03	Botão de redirecionamento	Média

Requisitos Não Funcionais

A velocidade do sistema dependerá do hardware, e da velocidade de navegação do usuário;
A facilidade de uso pode ser medida pelo número de janelas ou o tempo de treino do usuário.

Restrições

Descrição
Servidor para o site

3. METODOLOGIA

Relação de ambiente de trabalho

Ambiente	Plataforma	Link de Acesso
Repositório de Código fonte	GitHub	Clique aqui
Documentos do Projeto	Word	Clique aqui
Projeto de Interface e Wireframes	Figma	Clique aqui
Gerenciamento do projeto	GitHub	Clique aqui

Gestão de código fonte

Para gestão do código fonte do software desenvolvido pela equipe, o grupo utiliza um processo baseado no Git Flow abordado por Vietro (2015). Desta forma, todas as manutenções no código são realizadas em branches separados.

Gerenciamento do Projeto

A equipe utiliza metodologias ágeis, tendo escolhido o Scrum como base para definição do processo de desenvolvimento.

A equipe está organizada da seguinte maneira:

- Scrum Master: Thiago Almeida
- Product Owner: Tássia Teixeira
- Equipe de Desenvolvimento: César Pereira; Gabriel Alexandre; Leonardo Jaques; Tássia Teixeira e Thiago Almeida.

Para organização e distribuição das tarefas do projeto, a equipe está utilizando o GitHub estruturado com as seguintes listas:

- **Recursos:** esta lista mantém template de tarefas recorrentes com as configurações padronizadas que todos devem seguir. O objetivo é permitir a cópia destes templates para agilizar a criação de novos cartões.
- **Backlog:** recebe as tarefas a serem trabalhadas e representa o Product Backlog. Todas as atividades identificadas no decorrer do projeto também devem ser incorporadas a esta lista.
- **To Do:** Esta lista representa o Sprint Backlog. Este é o Sprint atual que estamos trabalhando.
- **Doing:** Quando uma tarefa tiver sido iniciada, ela é movida para cá.

- **Test:** Checagem de Qualidade. Quando as tarefas são concluídas, eles são movidas para o “CQ”. No final da semana, eu revejo essa lista para garantir que tudo saiu perfeito.
- **Done:** nesta lista são colocadas as tarefas que passaram pelos testes e controle de qualidade e estão prontos para ser entregues ao usuário. Não há mais edições ou revisões necessárias, ele está agendado e pronto para a ação.
- **Locked:** Quando alguma coisa impede a conclusão da tarefa, ela é movida para esta lista juntamente com um comentário sobre o que está travando a tarefa.

4. PROJETO DE INTERFACE

Dentre as preocupações para a montagem da interface do sistema, estamos estabelecendo foco em questões como agilidade, acessibilidade e usabilidade. Desta forma, o projeto tem uma identidade visual padronizada em todas as telas que são projetadas para funcionamento em desktops e dispositivos móveis.

Fluxo de Usuário

O diagrama apresentado na figura 1 mostra o fluxo de interação do usuário pelas telas do sistema. Cada uma das telas deste fluxo é detalhada na seção de Wireframes que se segue.



22 April 2022

Figura 1: Fluxo de interação do usuário

Wireframes

Conforme fluxo de telas do projeto, apresentado no item anterior, as telas do sistema são apresentadas em detalhes nos itens que se seguem. Nesta estrutura, existem 2 blocos:

- Cabeçalho: local onde são dispostos elementos fixos de identidade (logo) e navegação principal do site (menu da aplicação);
- Conteúdo: apresenta o conteúdo da tela em questão.

Tela – Tela Inicial

Apresenta opções para o usuário navegar pelo site, escolhendo entre as abas disponíveis. Possui um carrossel de imagens que contem informações gerais da ONG.



Figura 2: Tela inicial

Tela – Quem somos?

Apresenta um breve relato sobre o que é a ONG e lista os serviços prestados por ela.



Figura 3: Quem somos?

Tela – Nossa História

Mostra a história da instituição detalhadamente e quem são seus colaboradores.



Figura 4: Nossa História

Tela - Formulário de contato

Ao preencher o formulário, o usuário entra em contato com a ONG pelo WhatsApp já enviando as principais informações ao atendente.



Figura 5: Formulário de contato

Tela - Localização

Mostrar a localização da ONG através do Google Maps.

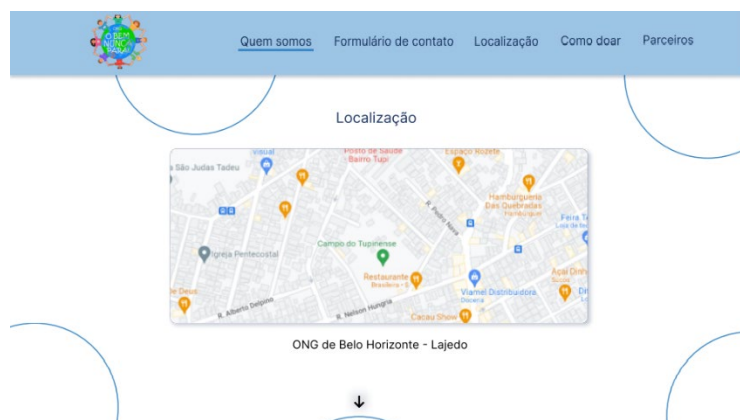


Figura 6: Localização

Tela – Informações de Doação

Informações sobre como ajudar a instituição, quais são os produtos que ela recebe e como podemos enviá-los.



Figura 7: Informações de doação

Tela - Parceiro

Apresenta ao público quais são as instituições que apoiam a ONB O bem nunca para.

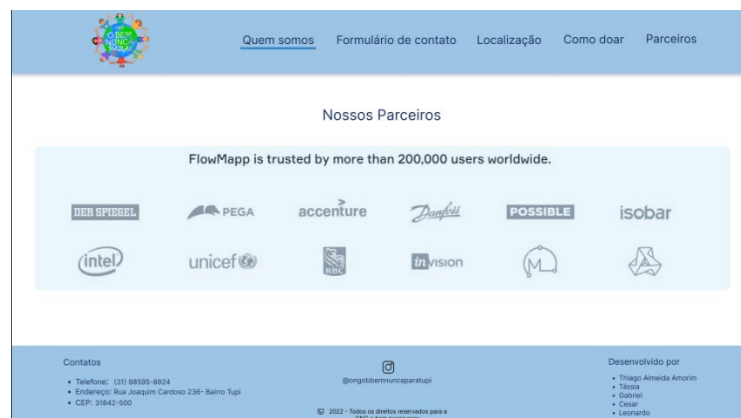


Figura 8: Parceiros

5. ARQUITETURA DA SOLUÇÃO

Nesta seção são apresentados os detalhes técnicos da solução criada pela equipe, tratando dos componentes que fazem parte da solução e do ambiente de hospedagem da solução.

Diagrama de Componentes

Os componentes que fazem parte da solução são apresentados na Figura que se segue.

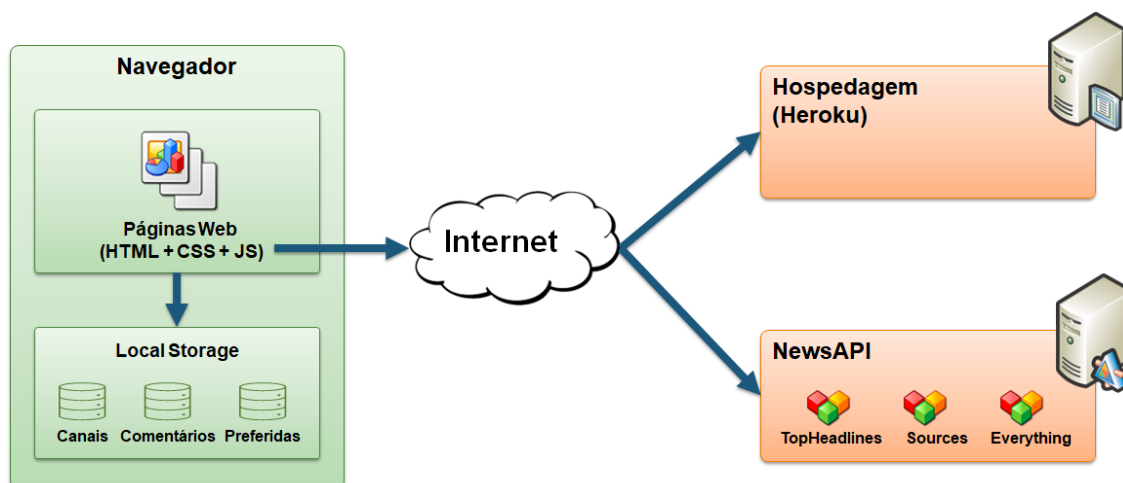


Figura 9: Arquitetura da Solução

A solução implementada conta com os seguintes módulos:

- **Navegador** - Interface básica do sistema
 - **Páginas Web** - Conjunto de arquivos HTML, CSS, JavaScript e imagens que implementam as funcionalidades do sistema.
- **Hospedagem** - local na Internet onde as páginas são mantidas e acessadas pelo navegador.

Hospedagem

O site utiliza a plataforma do Heroku como ambiente de hospedagem do site do projeto. O site é mantido no ambiente da URL:

https://link_exemplo.herokuapp.com

A publicação do site no Heroku é feita por meio de uma submissão do projeto (push) via git para o repositório remoto que se encontra no endereço:

6. TEMPLATE DO SITE

De acordo com o projeto de Interface elaborado anteriormente, o site seguirá o seguinte padrão:

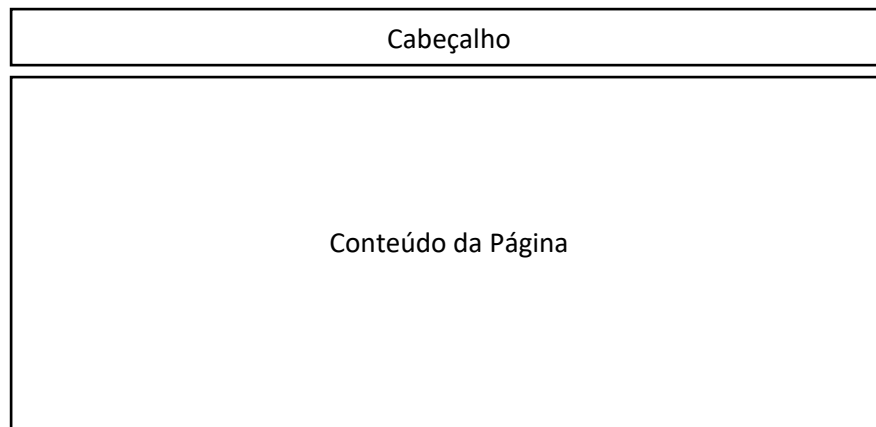


Figura 10: Arquitetura da Solução

O template criado está disponível no site https://github.com/ICEI-PUC-Minas-PMV-ADS/pmv-ads-2022-1-e1-proj-web-t10-ong_bnp.git e é composto pelos seguintes layouts:

- Homepage
- Quem somos
- Nossa História
- Contato
- Localização
- Como doar
- Parceiros

Tela – Homepage

A homepage apresenta um cabeçalho fixo com a logo da instituição O Bem Nunca Para e botões com opções de navegação pelo site. Além disso mostra um carrossel composto por três imagens com informações sobre a instituição.



Figura 11: Homepage

Tela - Quem somos

O layout “Quem somos” apresenta brevemente o que é a instituição e lista os principais serviços oferecidos por ela. Tem uma imagem ilustrativa com a fachada da unidade de Lajedo.

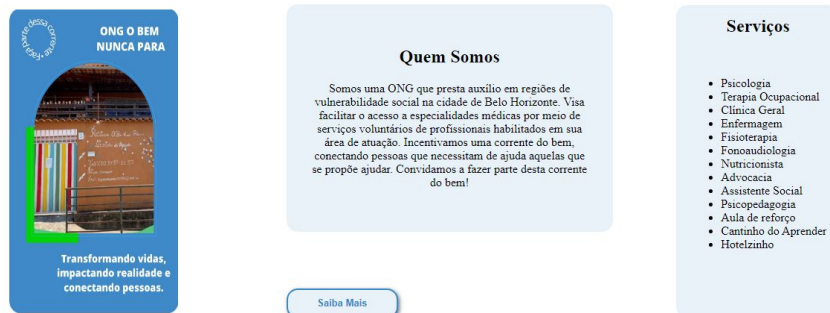


Figura 12: Quem Somos

Tela - Nossa História

A página nossa história é um espaço para que a instituição compartilhe detalhadamente sobre como iniciou o projeto e como desenvolveu até atualmente. Apresenta também seus principais colaboradores.

Tela – Contato

Em contato apresentamos um layout com funcionalidade, trata-se de um formulário onde o usuário preenche com suas informações e escolhe qual serviço da instituição gostaria de saber mais. Ao enviar ele é redirecionado para o WhatsApp de atendimento com uma mensagem inicial baseada nas informações enviadas.



Figura 14: Contato

Tela - Localização

Tela que apresenta ao usuário mapa da localização da ONG através do html do Google Maps.

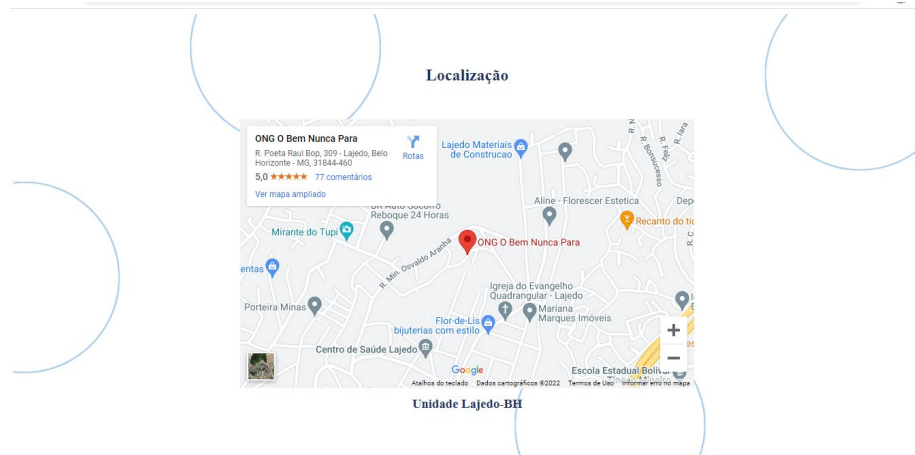


Figura 15: Localização

Tela com Orientações de Como Doar

Tela que orienta o usuário de como realizar doações caso tenha interesse em colaborar com a instituição.

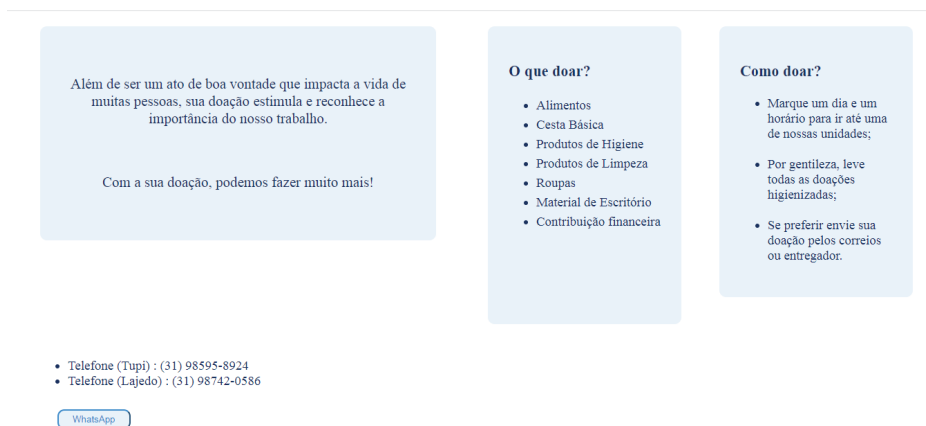


Figura 16: Como doar

Tela – Parceiros

Espaço em que a instituição compartilha quais são seus parceiros que apoiam e viabilizam a realização dos atendimentos ao público e demais serviços comunitários.

NOSSO PARCEIROS



Completar com
informações da ONG

CONTATOS:

Telefone: (11) 88395-8924
Endereço: Rua Joaquim Cardoso 236 -
Bairro Tupy
CEP: 13142-500

@ongobemmanuscritap

DESENVOLVIDO POR:

Thiago Almeida
Jéssica Teixeira
Cristina Pereira
Gabriel Alexandre
Leonardo Riquelme

© 2022 - Todos os direitos reservados para a ONG o bem manuscrito para

Figura 17: Parceiros

7. FUNCIONALIDADES DO SISTEMA

Nesta seção detalharemos as telas Homepage e Entre em contato que apresentam as funcionalidades do sistema.

Homepage (RF-01)

A homepage apresenta um menu fixo com a logo da instituição O Bem Nunca Para e botões com opções de navegação pelo site. Além disso mostra um carrossel composto por três imagens com informações sobre a instituição.



Figura 18: Homepage

Requisitos atendidos

- RF-01

Artefatos da funcionalidade

- index.html
- carrossel.js
- homepage.css
- logo_logo.png

Estrutura do Código JavaScript

```
let totalSlides = document.querySelectorAll('.slider--item').length;
let currentSlide = 0;

document.querySelector('.slider--width').style.width =
  `calc(100vw * ${totalSlides})`;
document.querySelector('.slider--controls').style.height =
  `${document.querySelector('.slider').clientHeight}px`;

window.onload = setInterval(() => {
  goNext();
}, 5000);

function goPrev(){
  currentSlide--;
  if(currentSlide < 0) {
    currentSlide = totalSlides - 1;
  }
  updateMargin();
}

function goNext() {
  currentSlide++;
```

```

    if(currentSlide > (totalSlides-1)) {
        currentSlide = 0
    }
    updateMargin();
}

function updateMargin() {
    let newMargin = (currentSlide * document.body.clientWidth);
    document.querySelector('.slider--width').style.marginLeft =
    `${newMargin}px`;
}

```

Instruções de Acesso

1. Abra o navegador de internet e informe a seguinte URL: <https://exemplo.com.br>
2. A homepage é o primeiro layout exibido no site a tela do menu lhe acompanha.

Contato (RF-02)

A página com título “Entre em contato” trata-se de um formulário onde o usuário preenche com suas informações e escolhe qual serviço da instituição gostaria de saber mais. Ao enviar ele é redirecionado para o WhatsApp de atendimento com uma mensagem inicial baseada nas informações enviadas.

Figura 19: Contato

Requisitos atendidos

- RF-02

Artefatos da funcionalidade

- index.html
- script_form.js
- form.css

Estrutura do Código JavaScript

```

//Pega cada input pelo seu ID:
const ValueInputs = {

```

```

    DataName: document.querySelector("#Name"),
    DataEmail: document.querySelector("#Email"),
    DataTel: document.querySelector("#Tel"),
    DataServicos: document.querySelector("#Servicos"),
};

//Pega cada label pelo seu ID:
const ValueLabels = {
    DataLabelName: document.querySelector("#LabelName"),
    DataLabelEmail: document.querySelector("#LabelEmail"),
    DataLabelTel: document.querySelector("#LabelTel"),
    DataLabelServicos: document.querySelector("#LabelServicos"),
};

//Número do WhatsApp da ONG (país+ddd+número):
const Number = '5531983083705';

//Lista de serviços oferecidos pela ONG:
const ServicosOfertado = [
    'Psicologia',
    'Terapia_Ocupacional',
    'Clínica_Geral',
    'Enfermagem',
    'Fisioterapia',
    'Fonoaudiologia',
    'Nutricionista',
    'Advocacia',
    'Assistente_Social',
    'Psicopedagogia',
    'Reforço_Escolar',
    'Cantinho_Aprender',
    'Hotelzinho',
];

//Recebe como parâmetro o valor (string) que é digitado no input de pesquisa e retorna os valores filtrados:
const AutoComplete = (ServicoSolicitado) => {

    const ServicoFormatUnderlin = ServicoSolicitado.replace(' ', '_');

    const ValueFilter = ServicosOfertado.filter((valor) => {

        const valorMinusculo = valor.toLowerCase();
        const ServicoSolicitadoMinusculo = ServicoFormatUnderlin.toLowerCase();

        return valorMinusculo.includes(ServicoSolicitadoMinusculo)
    });

    return ValueFilter.length === 0 ? ['error'] : ValueFilter
};

//Retorna o valor filtrado na tela do usuário:
ValueInputs.DataServicos.addEventListener('input', ({ target }) => {

    //Valor que está sendo digitado no input
    const DadosDoCampo = target.value;

    //Retorno da função que filtra a lista de serviços disponíveis
    const AutoCompleteValores = AutoComplete(DadosDoCampo);

    //Mapeia os valores filtrados e retorna uma lista deles na tela:
    document.getElementById('List').innerHTML = `
        ${AutoCompleteValores.map((value) => {

            //Css customizado: Quando nenhum valor for digitado no input não faz nada, se não colocar uma cor de fundo
            document.getElementById('CampoList').style.backgroundColor = value === 'error' || target.value === "" ? " : rgba(114,
114, 114, 0.802)";

            return (
                value === 'error' || target.value === "" ? " : `
                <ul class="ContainerSelect" >
                    <option value=${value} id="Opt" >${value.replace('_', ' ')}</option>
                </ul>
            `
            )
        })}
    `
});

```

```

    }).join(")}
    `;

//Seleciona todos as opções do autocomplete:
const ListOption = document.querySelectorAll('#Opt');

//Lista cada uma das opções, adiciona um evento de click em cada uma delas
for (let Option of ListOption) {
    Option.addEventListener('click', (event) => {

        //Tira a lista de auto complete da tela:
        document.getElementById('CampoList').style.backgroundColor = "transparent";
        document.getElementById('List').innerHTML = "";

        //O input de serviços passa a ter o valor de uma das opções do autocomplete:
        ValueInputs.DataServicos.value = event.target.value.replace('_', ' ');
    });
};

});

//Regex - Não permite caracteres especiais:
const RegexName = /^[w\^\.,,][^0-9_!;?+?¿\+=@#%$^&*(){}|~<>:;[\]]{2,}$;/;
const RegexEmail = /^[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]\.[a-zA-Z0-9_-]+\.$/;
const RegexTel = /^(?:\d{3}|\d{4})?(55)?\d{2}?(?:\d{1-9}[0-9])?\d{2}?(?:\d{1-9}[0-9])?\d{3})\d{3})\d{3})$/;

//Testando o nome do usuário:
const TestInputName = (Data) => {

    //Verificando se o campo não está vazio:
    if (Data.value === "") {
        return {
            error: true,
            message: 'Campo obrigatório'
        };
    };

    //Convertendo a primeira letra do nome e sobre nome maiúscula:
    const StringCapitalize = Data.value.toLowerCase().split(' ').map((TextPart) => {
        return TextPart[0].toUpperCase() + TextPart.slice(1);
    }).join(' ');

    //Contando a quantidade de palavras do nome
    const TwoString = Data.value.split(' ').length;

    //Verificando se não existe caracteres especiais no nome ou números:
    if (!RegexName.test(StringCapitalize)) {
        return {
            error: true,
            message: 'Não deve conter caracteres especiais ou números'
        };
    };

    //Verificando se foi inserido nome e sobre nome:
    if (TwoString < 2) {
        return {
            error: true,
            message: 'É necessário informar seu nome completo'
        };
    };

    return {
        error: false,
        message: ""
    };
};

//Testando o email do usuário:
const TestInputEmail = (Data) => {

    //Verificando se o campo não está vazio:
    if (Data.value === "") {
        return {

```

```

        error: true,
        message: 'Campo obrigatório'
    };
};

//Verificando se o e-mail tem um formato válido:
if (!RegexEmail.test(Data.value)) {
    return {
        error: true,
        message: 'Formato do e-mail inválido'
    };
}

return {
    error: false,
    message: ""
};
};

//Testando o telefone do usuário:
const TestInputTel = (Data) => {

    //Verificando se o campo não está vazio:
    if (Data.value === "") {
        return {
            error: true,
            message: 'Campo obrigatório'
        };
    };

    //Verificando se o telefone tem um formato válido:
    if (!RegexTel.test(Data.value)) {
        return {
            error: true,
            message: 'Número de telefone inválido'
        };
    }

    return {
        error: false,
        message: ""
    };
};

//Testando o telefone do usuário:
const TestInputServicos = (Data) => {

    const ServicoValue = Data.value;

    //Verificando se o campo não está vazio:
    if (Data.value === "") {
        return {
            error: true,
            message: 'Campo obrigatório'
        };
    };

    //Verificando se o serviço inserido existe:
    const ServicoFilter = ServicosOfertado.filter((Servico) => Servico === ServicoValue.replace(' ', '_')? Servico : "");

    if (ServicoFilter[0]) {
        return {
            error: false,
            message: ""
        };
    } else {
        return {
            error: true,
            message: 'Esse serviço não existe! Selecione ou insira um serviço válido'
        };
    };
};

//Css para erros:

```



```

const StyledError = (Type) => {
  document.getElementById(`${Type}`).style.borderColor = 'red'
  document.getElementById(`Label${Type}`).style.color = 'red'
  document.getElementById(`Label${Type}`).style.marginBottom = '15px'
};

const StyledTrue = (Type) => {
  document.getElementById(`${Type}`).style.borderColor = '#3A89C9'
  document.getElementById(`Label${Type}`).style.color = '#3A89C9'
  document.getElementById(`Label${Type}`).style.marginBottom = '0px'
};

//Formata o número de telefone que será criado
let NumberFormat = "";

ValueInputs.DataTel.addEventListener('input', ({ target }) => {
  const NumberTel = target.value;

  const DDD = NumberTel.slice(0,2);
  const PrimaryNumber = NumberTel.slice(2,7);
  const SecondaryNumber = NumberTel.slice(7,11);

  NumberFormat = `+55 (${DDD}) ${PrimaryNumber}-${SecondaryNumber}`;
});

//Redireciona o usuário para o WhatsApp da ONG e envia a mensagem customizada automaticamente:
document.getElementById('form').addEventListener('submit', (e) => {
  e.preventDefault();

  //Mensagem Customizada:
  let Msg = "";

  Msg = `
    Olá, me chamo ${ValueInputs.DataName.value} e gostaria de saber mais sobre o serviço de
    ${ValueInputs.DataServicos.value}.
    %0A
    %0A
    Meus dados para contato são:
    %0A
    %0A
    Número: ${NumberFormat}
    %0A
    E-mail: ${ValueInputs.DataEmail.value}
  `;

  //Pegando o valor dos inputs:
  const IsErrorName = TestInputName(ValueInputs.DataName);
  const IsErrorEmail = TestInputEmail(ValueInputs.DataEmail);
  const IsErrorTel = TestInputTel(ValueInputs.DataTel);
  const IsErrorServicos = TestInputServicos(ValueInputs.DataServicos);

  //Verificando o nome do usuário
  if (IsErrorName.error) {
    ValueLabels.DataLabelName.innerHTML = `<div>${IsErrorName.message}</div>`
    StyledError('Name');
    return undefined
  } else {
    ValueLabels.DataLabelName.innerHTML = ""
    StyledTrue('Name');
  };

  //Verificando o email do usuário
  if (IsErrorEmail.error) {
    ValueLabels.DataLabelEmail.innerHTML = `<div>${IsErrorEmail.message}</div>`
    StyledError('Email');
    return undefined
  } else {
    ValueLabels.DataLabelEmail.innerHTML = ""
    StyledTrue('Email');
  };

  //Verificando o telefone do usuário
  if (IsErrorTel.error) {
    ValueLabels.DataLabelTel.innerHTML = `<div>${IsErrorTel.message}</div>`
    StyledError('Tel');
  };

```

```
        return undefined
    } else {
        ValueLabels.DataLabelTel.innerHTML = "
        StyledTrue('Tel');
    };

    //Verificando se algum serviço foi selecionado
    if (IsErrorServicos.error) {
        ValueLabels.DataLabelServicos.innerHTML = `<div>${IsErrorServicos.message}</div>`
        StyledError('Servicos');
        return undefined
    } else {
        ValueLabels.DataLabelServicos.innerHTML = "
        StyledTrue('Servicos');
    };

    window.open(`https://web.whatsapp.com/send?phone=${Number}&text=${Msg}`)
});
```

Instruções de Acesso

1. Abra o navegador de internet e informe a seguinte URL: <https://exemplo.com.br>
2. Navegue até o layout “Entre em Contato” ou escolha a opção “Contato” no menu.
3. Preencha o formulário
4. Clique no botão enviar.

8. PLANO DE TESTES DE SOFTWARE

Os requisitos para realização dos testes de software são:

- Navegador da Internet - Chrome, Opera GX e Microsoft Edge.
- Conectividade de Internet para acesso às plataformas (APIs)

Os testes funcionais a serem realizados no aplicativo são descritos a seguir.

Caso de Teste	CT-01 – Funcionalidade dos botões do menu
Requisitos Associados	RF-01: Menu fixo
Objetivo do Teste	Verificar se o menu está fixo e visível em todo o site.
Passos	1) Acessar o navegador; 2) Informar o endereço do site; 3) Visualizar a página principal e rolar nas demais.
Critérios de Êxito	<ul style="list-style-type: none">• Menu ficar visível em toda a extensão da página;• Quando utilizarmos determinado botão ele deve redirecionar para a parte correta da página.

Caso de Teste	CT-02 – Funcionamento do formulário de contato
Requisitos Associados	RF-02: Formulário: Entre em Contato
Objetivo do Teste	Verificar se o formulário redireciona o usuário para contato direto com os responsáveis da ONG.
Passos	1) Acessar o navegador; 2) Informar o endereço do site; 3) Visualizar a página “Formulário”; 4) Preencher todos os dados requisitados; 5) Clicar em “enviar”.
Critérios de Êxito	<ul style="list-style-type: none">• Usuário é redirecionado para contato via WhatsApp;• Instituição recebe todos os dados informados pelo usuário.

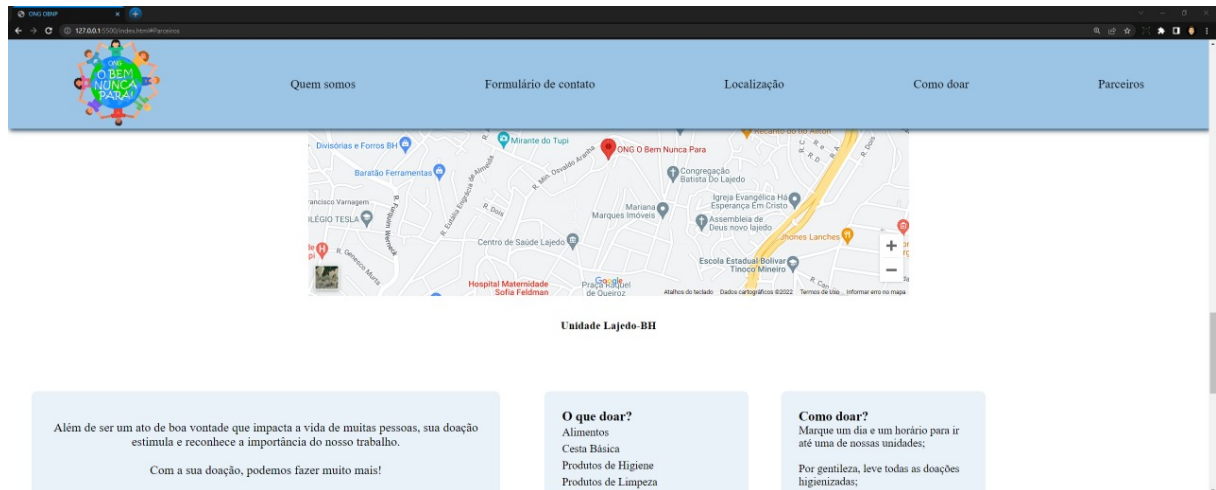
9. REGISTRO DE TESTES DE SOFTWARE

Será descrito a seguir os testes realizados para cada um dos casos de testes. Previstos para as duas funcionalidades contidas no site.

Registros de testes para CT-01

Teste 1: Rolar por toda a extensão da página do site.

Foi verificado que o menu estava visível ao longo de toda a extensão da página.



Teste 2: Apertar no botão “Quem somos”.

Foi verificado que o usuário é redirecionado para o local correspondente.



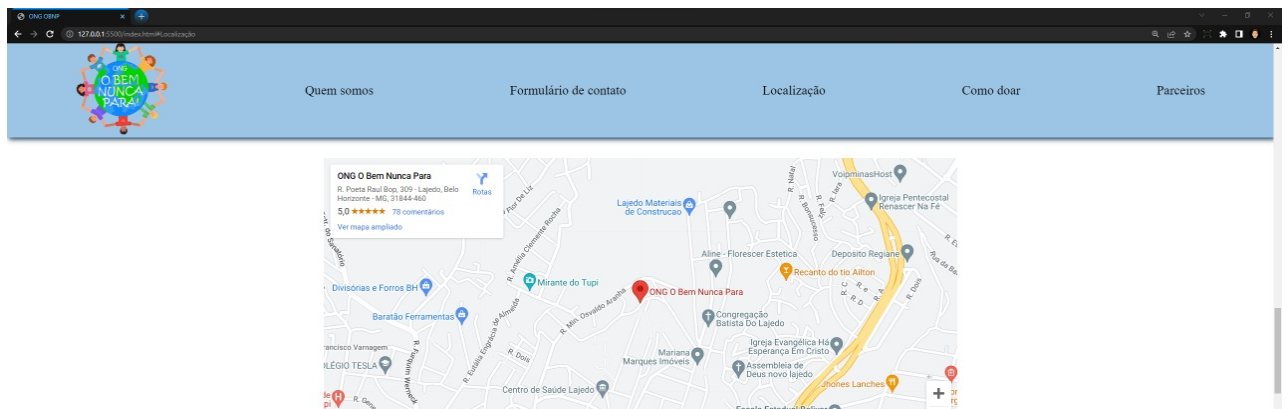
Teste 3: Apertar no botão “Formulário”.

Foi verificado que o usuário é redirecionado para o local correspondente.



Teste 4: Apertar no botão “Localização”.

Foi verificado que o usuário é redirecionado para o local correspondente.



Teste 5: Apertar no botão “Como doar”.

Foi verificado que o usuário é redirecionado para o local correspondente.



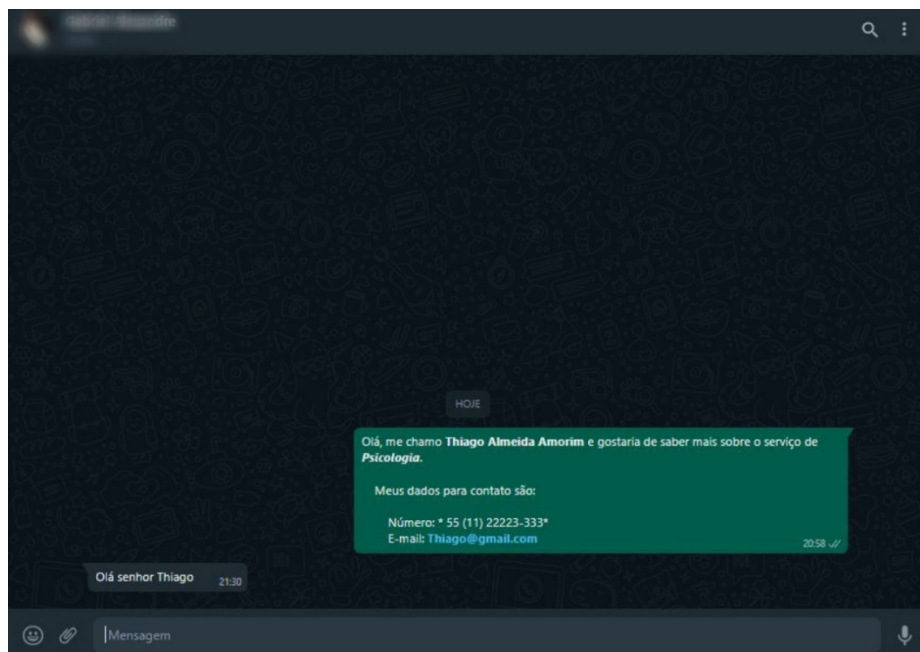
Teste 6: Apertar no botão “Parceiros”.

Foi verificado que o usuário é redirecionado para o local correspondente.

Registros de testes para CT-02

Teste 1: Preencher todas as informações corretamente e clicar em enviar.

Foi verificado que o usuário foi redirecionado para contato via WhatsApp com mensagem contendo todas as informações enviadas.



Teste 2: Tentar enviar formulário sem nome ou com caracteres especiais.
Foi verificado que o formulário não é enviado ou o usuário é informado de que não deve conter números ou caracteres especiais nesse campo.

Entre em Contato

Nome

Campo obrigatório

Email

Telefone

Pesquisar por serviços...

Enviar

Entre em Contato

Não deve conter caracteres especiais ou números

Email

Telefone

Pesquisar por serviços...

Enviar

Teste 3: Tentar encaminhar formulário sem e-mail ou em formato errado.

Foi verificado que é informado ao usuário que esse campo é obrigatório ou que o formato está inválido.

Entre em Contato

The image displays two screenshots of a contact form titled "Entre em Contato". The form contains five input fields: a name field (filled with "Thiago Almeida Amorim"), an email field, a phone field, a search field (placeholder "Pesquisar por serviços..."), and a submit button labeled "Enviar".

In the first screenshot, the email field is empty and highlighted with a red border. Below it, the text "Campo obrigatório" is displayed in red.

In the second screenshot, the email field contains the text "User.com" and is highlighted with a red border. Below it, the text "Formato do e-mail inválido" is displayed in red.

Teste 4: Tentar encaminhar o formulário sem informar telefone ou informando em formato inadequado.

Foi verificado que a mensagem de campo obrigatório apareceu ou mensagem de que o telefone está em formato inválido.

Entre em Contato

The image displays a screenshot of a contact form titled "Entre em Contato". The form contains five input fields: a name field (filled with "Thiago Almeida Amorim"), an email field (filled with "Thiago@gmail.com"), a phone field, a search field (placeholder "Pesquisar por serviços..."), and a submit button labeled "Enviar".

In this screenshot, the phone field is empty and highlighted with a red border. Below it, the text "Campo obrigatório" is displayed in red.

Entre em Contato



Thiago Almeida Amorim

Thiago@gmail.com

111111111111111111111111

Número de telefone inválido

Pesquisar por serviços...

Enviar

Teste 5: Tentar enviar o formulário sem a parte de “Pesquisar por serviços...”
Foi verificado que a mensagem de campo obrigatório aparece para o usuário.

Entre em Contato



Thiago Almeida Amorim

Thiago@gmail.com

1122222222

Pesquisar por serviços...

Campo obrigatório

Enviar