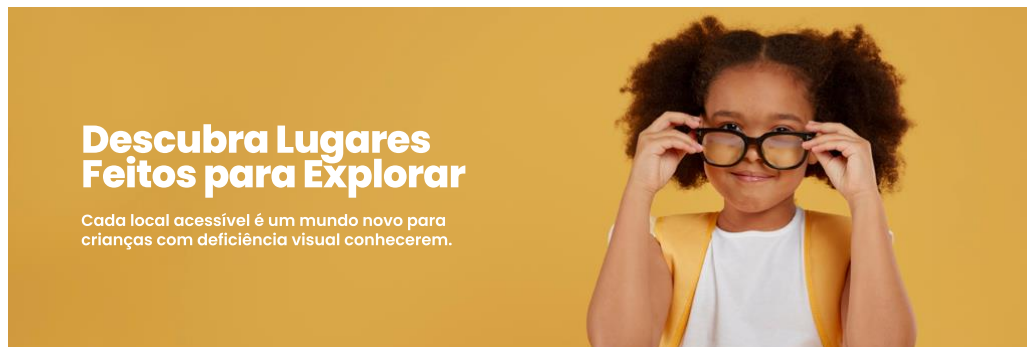


Manual de usuário:

Com a implementação deste sistema, elaboramos nosso manual de usuário de forma simplificada para facilitar a compreensão e o uso.

1ª tela – Landing Page:



Nossos compromissos

Nosso compromisso é tornar cada detalhe acessível!



Acessibilidade Universal

Compromisso em tornar os recursos do site acessíveis para pessoas com deficiência visual, digitalmente e experiências nos locais destacados.



Empatia e Sensibilidade

Enfatizar a importância de entender as necessidades específicas das crianças com deficiência visual e suas famílias na sua rotina.



Compromisso com a Inclusão

Demonstração do comprometimento em destacar espaços inclusivos e acessíveis para crianças com deficiência visual.



Inclusão

Um site que lista lugares com acessibilidade visual para crianças é essencial para garantir que todas possam explorar o mundo ao seu redor de forma segura e igualitária.

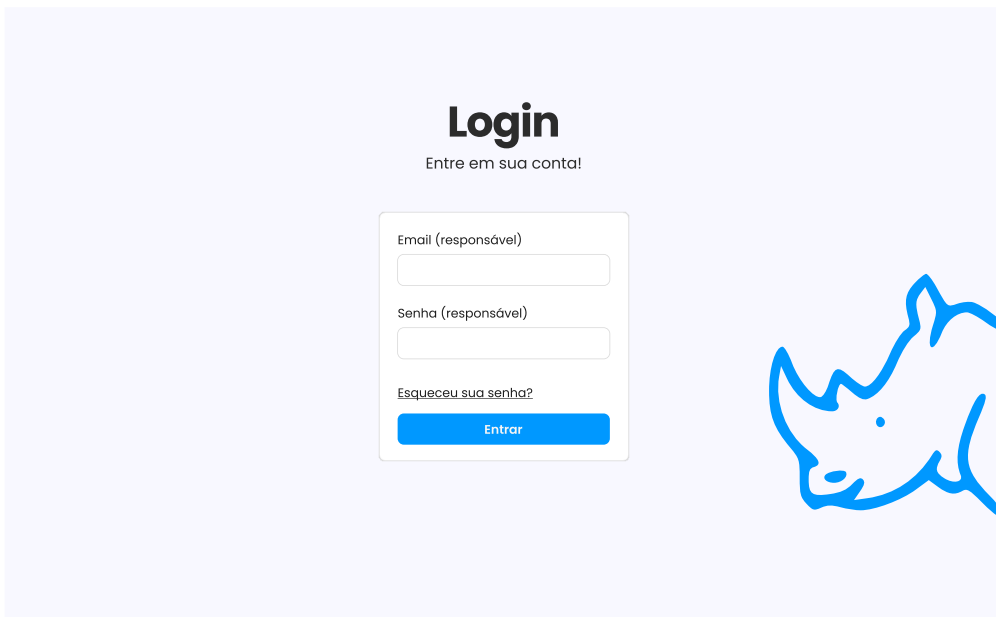


Facilidade para Famílias

Ter acesso a informações sobre locais adaptados ajuda famílias a planejar atividades, proporcionando experiências seguras e enriquecedoras para crianças com deficiência visual.

Essa tela tem a funcionalidade de informar os usuários sobre o nosso projeto, nela temos informações sobre o projeto, para que nosso usuário entenda do que se trata e consiga nos conhecer. Além de também encaminhar o usuário para a página de login ou para a página de cadastro, dando acesso a tela inicial do site.

2ª tela – Login:



The login screen features a light purple background. On the left, there is a white login form with the title 'Login' and the subtitle 'Entre em sua conta!'. The form contains two input fields for 'Email (responsável)' and 'Senha (responsável)', a link for 'Esqueceu sua senha?', and a blue 'Entrar' button. On the right, there is a blue line-art illustration of a rhinoceros's head.

Login

Entre em sua conta!

Email (responsável)

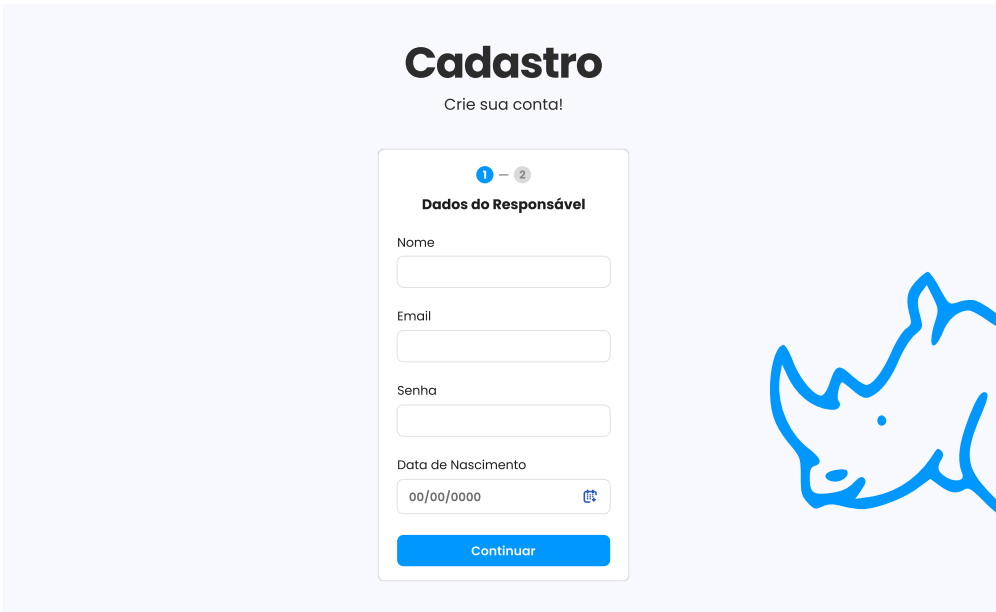
Senha (responsável)

[Esqueceu sua senha?](#)

Entrar

Na tela de login, o responsável deve inserir seu e-mail e senha para acessar o site. Caso tenha esquecido a senha, há a opção de redefini-la.

3ª tela – Cadastro do responsável:



The registration screen features a light purple background. On the left, there is a white registration form with the title 'Cadastro' and the subtitle 'Crie sua conta!'. The form has a progress indicator showing '1' of '2' steps, followed by the title 'Dados do Responsável'. It includes input fields for 'Nome', 'Email', and 'Senha', a date picker for 'Data de Nascimento', and a blue 'Continuar' button. On the right, there is a blue line-art illustration of a rhinoceros's head.

Cadastro

Crie sua conta!

1 — 2

Dados do Responsável

Nome

Email

Senha

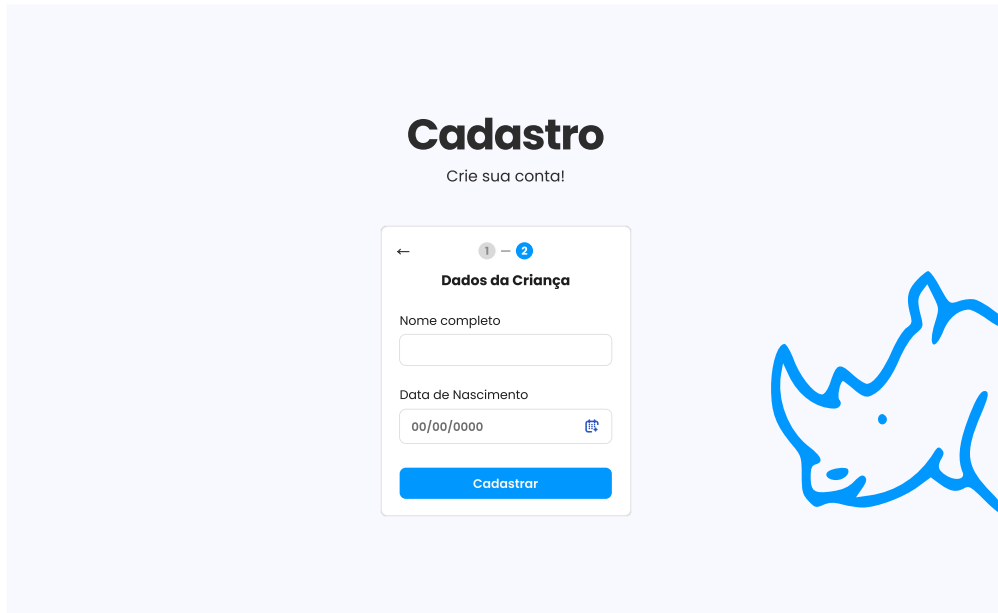
Data de Nascimento

00/00/0000

Continuar

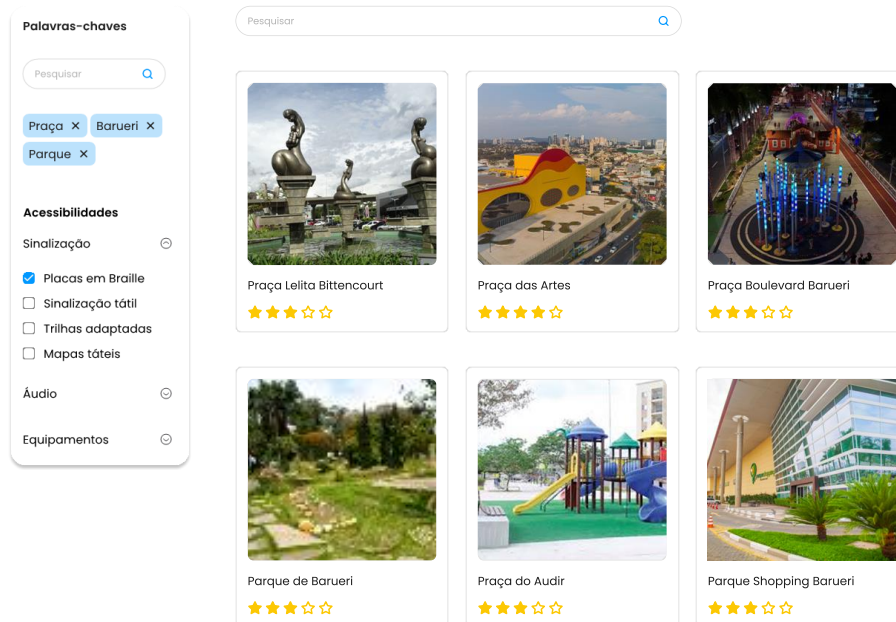
Na tela de cadastro, o responsável legal pela criança deve inserir seu nome, e-mail, senha e data de nascimento. Após preencher essas informações e clicar em "Continuar", o usuário será redirecionado para a página de cadastro da criança.

3ª tela – Cadastro da criança:



Nessa tela, será possível fazer o cadastro de seu dependente, adicionando as seguintes informações: nome completo e a data de nascimento. Após isso o usuário será direcionado para página inicial, para começar sua jornada utilizando o Rhinny.

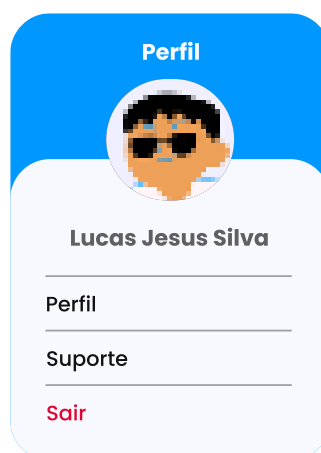
4ª tela – Página inicial:



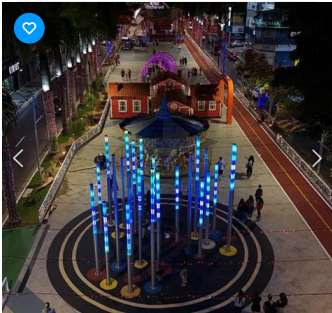
Na página inicial, o usuário pode pesquisar locais específicos na barra de pesquisa, como parques, praças, jardins, shoppings e outras atividades. Ao lado da barra, há um campo com várias palavras-chave relacionadas a locais e tipos de acessibilidade, como sinalizações, áudio e equipamentos acessíveis. Após realizar uma pesquisa, serão exibidos exemplos de locais que o usuário pode querer visitar, ou o local específico buscado.

Na seção do perfil, ao clicar, o usuário é direcionado para as configurações do perfil.

*Modelo de dropdown do perfil de usuário:




5ª tela – Página do lugar:




Praça Boulevard Barueri

[Piso Tátil](#) [Placas de Braille](#) [Mapas Táteis](#)



Adicione as acessibilidades disponíveis no local:

Sinalização	Áudio	Equipamentos
<input type="text" value="Value"/>	<input type="text" value="Value"/>	<input type="text" value="Value"/>

**Lucas Jesus Silva**
15-10-24

Escrever comentário

Answer the frequently asked question in a simple sentence, a longish paragraph, or even in a list.

☆☆☆☆☆

Enviar

Outros comentários

Veja a opinião de outros visitantes

☆☆☆☆☆

Gostei!

As acessibilidades condizem com as do site.

**Ana Paula**
05-06-24

☆☆☆☆☆

Sem título

É um lugar legal

**Bruno**
14-06-24

☆☆☆☆☆

Sem título

Bonito

**Giovana Julia**
12-03-24


Localize

Encontre o local no qual deseja ir



Nesta página, o usuário seleciona o local acessível que deseja visitar. São exibidas informações sobre as acessibilidades do local, além de comentários e avaliações de outras pessoas que o visitaram. Abaixo, o usuário tem a opção de deixar seu próprio comentário e avaliação. Além disso, a página irá incorporar geolocalização para ajudar o usuário a encontrar o melhor caminho, o tempo estimado de chegada e a localização do local.

6ª tela – Perfil:



Lucas Jesus Silva


Responsável

Ana Maria Jesus Silva

Editar perfil


Meus favoritos

Lugares marcados como favorito por você




Parque de Barueri

★★★★☆



Praça Boulevard Barueri

★★★★☆




Praça das Artes

★★★★☆


Meus comentários

Sua opinião por onde você passou


Exemplo 1

 Lucas Jesus Silva
XX-XX-XX


Exemplo 2

 Lucas Jesus Silva
XX-XX-XX


Exemplo 3

 Lucas Jesus Silva
XX-XX-XX


Exemplo 4

 Lucas Jesus Silva
XX-XX-XX

Exemplo 5

 Lucas Jesus Silva
XX-XX-XX


Exemplo 6

 Lucas Jesus Silva
XX-XX-XX

Na página de perfil, o usuário pode visualizar seu perfil, seus comentários, locais visitados e favoritos. Além disso, é possível desfavoritar locais.

A página também conta com uma tela flutuante que permite alterar o nome, e-mail, senha e data de nascimento.

Configurações de perfil



Nome

Lucas Jesus Silva

Nascimento

00/00/0000

Email

responsavel@gmail.com

Senha

••••••••

Excluir minha conta

Salvar

7ª tela – Fale Conosco:

Fale Conosco

Conte com a ajuda do nosso suporte

Perguntas frequentes

Precisa de uma resposta? Você pode encontrar aqui!

Filtrar por categoria

Geral

Geral

Landing Page

Tela inicial

Perfil

Fale Conosco

Política e Privacidade

Termos de Uso

O que é este site?



Como funciona a busca por locais acessíveis?



Quais tipos de acessibilidade posso encontrar?



Há algum custo para utilizar o site?



Envie seu feedback

Gostou da nossa iniciativa? Envie sugestões, críticas ou elogios para melhorarmos sua experiência conosco!

Email

Mensagem

Enviar

Na tela "Fale Conosco", você encontrará uma seção com perguntas frequentes que abordam dúvidas comuns sobre o uso do site e suas funcionalidades. Além disso, há a opção de enviar seu feedback ou fazer uma pergunta que não esteja listada nas perguntas frequentes.

Funcionalidades presentes em todo o site:

1ª funcionalidade – Navegação (sem login):



Login

Cadastro

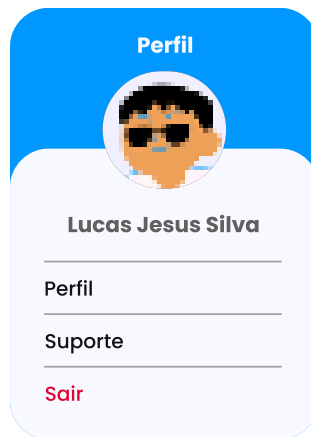
No menu de navegação sem login, o usuário encontrará dois botões: "Login" e "Cadastro". Ao clicar em cada botão, o usuário será redirecionado para a página correspondente.

2ª funcionalidade – Navegação (com login):



No menu de navegação (com login), o usuário terá acesso a um menu dropdown com opções para visualizar o perfil, acessar o suporte e sair da conta.

Dropdown:



3ª funcionalidade – Botão de acessibilidade:



Este será o botão de acessibilidade, que permitirá ao usuário ajustar o tamanho da fonte, alterar o espaçamento entre linhas, trocar para uma fonte serifada, modificar o contraste de cores, ajustar a saturação, adicionar uma máscara de leitura e usar uma lupa de leitura.

4ª funcionalidade – Rodapé (footer):

	Páginas	Redes Sociais	Contato
	Lading page	Instagram	Telefone
	Tela inicial	Linkedin	Email
	Perfil		Endereço
	Fale conosco		
	Política e Privacidade		
	Termos de uso		

Copyright © 2024 Rhinny

No rodapé, o usuário encontrará links para outras páginas, acesso às redes sociais e as informações de contato do suporte.

Manual de Desenvolvedor:

Com a implementação deste sistema, elaboramos nosso manual de desenvolvedor de forma objetiva, focado em fornecer orientações técnicas claras para facilitar o desenvolvimento e a manutenção do código.

```
//iniciando firebase
```

```
const firebaseConfig = {  
  apiKey: "AlzaSyAK1Y4DAftt63mVx3tngghMWXYDfg5zzTo",  
  authDomain: "rhinny-319e6.firebaseio.com",  
  projectId: "rhinny-319e6",  
  storageBucket: "rhinny-319e6.appspot.com",  
  messagingSenderId: "860506842460",  
  appId: "1:860506842460:web:e3176d1e88d16664590d99",  
  measurementId: "G-KKE67JZ0M6",  
};
```

```
//iniciando funções
```

```
const app = firebase.initializeApp(firebaseConfig);
```

```
const auth = firebase.auth();
```

```
const db = firebase.firestore();
```

```
//identifica o usuario a partir do estado de autenticação
```

```
firebase.auth().onAuthStateChanged((user) => {
```

```
  if (!user) {
```

```
    window.location.href = "../pagina-login/login.html";
```

```
  }
```

```
});
```

```
function logout() {
```

```
  firebase
```

```
    .auth()
```

```
    .signOut()
```

```
    .then(() => {
```

```
      window.location.href = "../pagina-login/login.html";
```

```
    })
```

```
    .catch(() => {
```

```
      alert("Erro ao fazer logout");
```

```
    });
```

```
}
```

```
// Este código escuta as mudanças no estado de autenticação do Firebase
```

```
firebase.auth().onAuthStateChanged((user) => {  
  
  // A função é chamada sempre que o estado de autenticação muda  
  
  if (user) {  
  
    // Se um usuário estiver logado, 'user' contém as informações do usuário  
  
    globalUserId = user.uid; // Armazena o UID do usuário globalmente  
  
    console.log("Usuário logado: " + globalUserId); // Exibe o UID do usuário no  
console  
  
  
    // Chama a função para obter os dados do usuário no Firestore  
  
    getUserData(globalUserId);  
  
  
    // Chama a função para mostrar os dados de autenticação na interface  
  
    mostrarDadosAuth(user);  
  
  } else {  
  
    // Se não houver usuário logado, exibe uma mensagem no console  
  
    console.log("Nenhum usuário logado.");  
  
  }  
  
});
```

```
// Função para obter dados do usuário no Firestore usando o UID  
  
function getUserData(globalUserId) {  
  
  const userRef = firebase.firestore().collection("users").doc(globalUserId); //  
Referência ao documento do usuário na coleção 'users'  
  
  
  userRef
```

```

.get() // Obtém os dados do documento

.then((doc) => {

  if (doc.exists) {

    const userData = doc.data(); // Se o documento existir, obtém os dados

    mostrarDadosFirestore(userData); // Chama a função para mostrar os
    dados na interface

  } else {

    console.log(globalUserId); // Exibe o UID se o documento não for
    encontrado

    console.log("Nenhum documento encontrado!"); // Mensagem caso não
    exista o documento

  }

})

.catch((error) => {

  console.error("Erro ao obter documento:", error); // Exibe erro caso ocorra um
  problema ao buscar os dados

});

}

```

// Função para exibir dados do usuário na tela

```

function mostrarDadosFirestore(userData) {

  document.getElementById("nome_usuario").innerHTML =
  userData.nome_adulto; // Mostra o nome do adulto na interface

  document.getElementById("nome_crianca").innerHTML =
  userData.nome_crianca; // Mostra o nome da criança na interface

```

```
document.getElementById("nome_adulto").innerHTML =  
userData.nome_adulto; // Mostra novamente o nome do adulto (opcional)  
  
document.getElementById("nome_trocar").placeholder =  
userData.nome_crianca; // Define o placeholder com o nome da criança  
  
document.getElementById("nome_trocar").value = userData.nome_crianca; //  
Define o valor do campo com o nome da criança  
  
}
```

// Função para mostrar dados de autenticação na tela

```
function mostrarDadosAuth(user) {  
  
    document.getElementById("email_trocar").placeholder = user.email; // Define o  
placeholder com o e-mail do usuário logado  
  
    document.getElementById("email_trocar").value = user.email; // Define o valor  
do campo com o e-mail do usuário logado  
  
}
```

// Função para apagar a conta do usuário autenticado e seus dados no Firestore

```
function apagandoUsuario() {  
  
    const user = firebase.auth().currentUser; // Obtém o usuário atualmente  
autenticado  
  
    user  
  
    .delete() // Tenta deletar a conta do usuário autenticado  
  
    .then(() => {  
  
        alert("Conta deletada"); // Mensagem de sucesso ao deletar a conta  
  
        logout(); // Chama a função de logout após deletar a conta  
  
    })  
  
}
```

```
    })  
  
    .catch((error) => {  
  
        alert("Ocorreu um erro: " + error.code); // Mensagem de erro se falhar ao  
deletar a conta  
  
        // ...  
  
    });
```

```
const userRef = firebase.firestore().collection("users").doc(globalUserId); //  
Referência ao documento do usuário no Firestore
```

```
userRef  
  
    .delete() // Tenta deletar os dados do usuário no Firestore  
  
    .then(() => {})  
  
    .catch((error) => {  
  
        alert("Não apagou do banco"); // Mensagem de erro se falhar ao deletar os  
dados no Firestore  
  
    });  
  
}
```

```
// Função para mudar os dados do usuário e, opcionalmente, redefinir a senha
```

```
function mudarDados() {  
  
    const userRef = firebase.firestore().collection("users").doc(globalUserId); //  
Referência ao documento do usuário no Firestore
```

```
    const nome_mudanca = document.getElementById("nome_trocar").value; //  
Obtém o novo nome da criança a partir da interface
```

```
const senha = document.getElementById("password-trocar").value; // Obtém a nova senha (se fornecida)
```

```
userRef
```

```
.update({  
  nome_crianca: nome_mudanca, // Atualiza o nome da criança no Firestore  
})  
  
.then(() => {  
  alert("Dados do usuário modificados com sucesso"); // Mensagem de sucesso ao atualizar os dados  
});
```

```
if (senha) {
```

```
  var email = document.getElementById("email_trocar").value; // Obtém o e-mail atual da interface
```

```
  firebase
```

```
    .auth()  
  
    .sendPasswordResetEmail(email) // Envia um e-mail para redefinição de senha se uma nova senha for fornecida  
  
    .then(() => {  
      alert("Email para redefinição de senha enviado para " + email); // Mensagem de sucesso ao enviar o e-mail de redefinição de senha  
    })  
  
    .catch((error) => {
```

```
    alert("Erro ao enviar email de redefinição de senha: " + error); // Mensagem  
de erro se falhar ao enviar o e-mail de redefinição de senha
```

```
    });  
  
  }  
  
}
```

```
// Objeto Form que contém métodos para obter valores dos campos do formulário
```

```
const Form = {  
  
  email: () => document.getElementById("email-cadastro").value,  
  
  senha: () => document.getElementById("senha-cadastro").value,  
  
  nome: () => document.getElementById("nome-cadastro").value,  
  
  data: () => document.getElementById("data").value,  
  
  nome_crianca: () => document.getElementById("nome-cadastro-  
crianca").value,  
  
  data_crianca: () => document.getElementById("data-cadastro-crianca").value,  
  
  botao_cadastrar: () => document.getElementById("btn-submit-crianca"),  
  
  botao_continuar: () => document.getElementById("btn-continuar"),  
  
};
```

```
function logout() {  
  
  firebase  
  
    .auth()  
  
    .signOut()  
  
    .then(() => {  
  
      console.log("Usuário desconectado com sucesso.");  
  
    });  
  
}
```



```
    })  
    .catch(() => {  
        alert("Erro ao fazer logout");  
    });  
}
```

```
firebase.auth().onAuthStateChanged((user) => {  
    if (user) {  
        // Redirecionar ou realizar ações com o usuário autenticado  
        console.log("Usuário logado:", user.uid);  
        setTimeout(() => {  
            window.location.href = "../pagina-inicial/pagina-inicial.html";  
        }, 4000);  
    }  
});
```

// Adiciona um ouvinte de evento ao botão "btn-submit-crianca"

```
document  
    .getElementById("btn-submit-crianca")  
    .addEventListener("click", function () {  
        // Obtendo os valores dos campos  
        const email = Form.email();  
        const senha = Form.senha();  
        const nome = Form.nome();
```

```
const data = Form.data();

const nome_crianca = Form.nome_crianca();

const data_crianca = Form.data_crianca();


console.log(nome_crianca);

console.log(data_crianca);


// Criar usuário com email e senha

firebase

.auth()

.createUserWithEmailAndPassword(email, senha)

.then((userCredential) => {

    // Se a criação do usuário for bem-sucedida

    const user = userCredential.user; // Obtém o objeto do usuário

    console.log("Usuário criado com ID:", user.uid);

    enviandoParaBanco(user.uid, nome, data, nome_crianca, data_crianca);

    setTimeout(() => {

        window.location.href = "../pagina-inicial/pagina-inicial.html";

    }, 4000);

})

.catch((error) => {

    // Se ocorrer um erro durante a criação do usuário

    const errorMessage = showError(error);

    alert(errorMessage); // Exibe a mensagem de erro
```

```

    });

});

// Envia os dados para a coleção "users" no Firestore

function enviandoParaBanco(uid, nome, data, nome_crianca, data_crianca) {

    db.collection("users")

        .doc(uid) // Usa o UID do usuário autenticado como documento

        .set({

            nome_adulto: nome,

            data_adulto: data,

            nome_crianca: nome_crianca,

            data_crianca: data_crianca,

        })

        .then(() => {

            console.log("Dados enviados para:", uid);

        })

        .catch((error) => {

            console.error("Dados não enviados:", error);

        });

}

```

```

// Validação do formulário

function isFormContinuarValid() {

    const nome = Form.nome();

```

```
if (!nome) {  
    Form.botao_continuar().disabled = true;  
    return false;  
}
```

```
const email = Form.email();  
if (!email || !validarEmail()) {  
    Form.botao_continuar().disabled = true;  
    return false;  
}
```

```
const senha = Form.senha();  
if (!senha || senha.length <= 5) {  
    Form.botao_continuar().disabled = true;  
    return false;  
}
```

```
const data = Form.data();  
if (!data) {  
    Form.botao_continuar().disabled = true;  
    return false;  
}
```

```
return true;
```

```
}
```

```
// Funções de validação e exibição de erro
```

```
function validarEmail() {
```

```
    return /^[S+@\S+\.\S+/.test(Form.email());
```

```
}
```

```
//dando uma mensagem agradavel para o usuario
```

```
function showError(error) {
```

```
    switch (error.code) {
```

```
        case "auth/invalid-email":
```

```
            return "O endereço de e-mail está mal formatado.";
```

```
        case "auth/user-disabled":
```

```
            return "Esta conta foi desativada.";
```

```
        case "auth/user-not-found":
```

```
            return "Não há registro de usuário correspondente a este e-mail.";
```

```
        case "auth/wrong-password":
```

```
            return "A senha fornecida está incorreta.";
```

```
        case "auth/email-already-in-use":
```

```
            return "Este e-mail já está em uso por outra conta.";
```

```
        case "auth/operation-not-allowed":
```

```
            return "O método de autenticação não está habilitado.";
```

```
        case "auth/too-many-requests":
```

```
            return "Muitas solicitações foram feitas. Tente novamente mais tarde.";
```

```
        case "auth/account-exists-with-different-credential":
```

```
    return "A conta já existe com um provedor diferente. Tente fazer login com esse provedor.";
```

```
    case "auth/invalid-credential":
```

```
        return "As credenciais fornecidas são inválidas.";
```

```
    case "auth/network-request-failed":
```

```
        return "Falha na conexão com a rede. Verifique sua conexão e tente novamente.";
```

```
    default:
```

```
        return "Ocorreu um erro desconhecido. Tente novamente.";
```

```
    }
```

```
}
```

```
//impede do usuario passar valores invalidos no formulario
```

```
function disableContinuar() {
```

```
    Form.botao_continuar().disabled = !isFormContinuarValid();
```

```
    console.log(isFormContinuarValid());
```

```
}
```

```
function onChangeNome() {
```

```
    document.getElementById("erro-nome-obrigatorio").style.display =  
    !Form.nome()
```

```
        ? "block"
```

```
        : "none";
```

```
}
```

```
function onChangeEmail() {
```

```
document.getElementById("erro-email-obrigatorio").style.display =  
    !Form.email() ? "block" : "none";  
document.getElementById("erro-email-invalido").style.display = validarEmail()  
    ? "none"  
    : "block";  
}
```

```
function onChangeSenha() {  
    document.getElementById("erro-senha-obrigatoria").style.display =  
        !Form.senha() ? "block" : "none";  
}
```

```
function onChangeData() {  
    document.getElementById("erro-data-obrigatoria").style.display = !Form.data()  
        ? "block"  
        : "none";  
}
```