**Introdução**

Serão testados os seguintes itens:

* Postagem de publicação
* Conexão com o Backend
* Usabilidade
* Envio e armazenamento de mídias
* Envio do Formulário
* Armazenamento dos dados no banco de dados

O objetivo dos testes será focado primeiramente em testes de caixa branca, onde poderemos verificar o código da aplicação.

Serão feitos os testes em cima da integração com o banco de dados, onde o armazenamento de dados será o foco, em cima das funcionalidades desejadas e em cima da interação do usuário.

**Itens a serem testados**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Atividade** | **Descrição da atividade** | **Teste a ser feito** | **Concluído** |
| **01** | Postagem de Publicação | Verificar a funcionalidade de  postar uma Publicação no modo Mobile | Testar postar a Publicação sem o preenchimento dos campos obrigatórios | Não |
| **02** | Integração do backend | A aplicação deverá registrar no banco de dados os dados preenchidos no formulário e essa postagem deverá constar na página do site | Inserir os dados necessários para preencher o formulário e verificar a comunicação com o backend | Não |
| **03** | Usabilidade | A aplicação deverá funcionar corretamente dentre todos os itens que podem ser acessados. | Verificar item a item se ele está cumprindo a funcionalidade desejada. | Não |
| **04** | Envio de mídias | A aplicação deverá aceitar o envio de fotos, vídeos e áudios. | Verificar se a aplicação aceita e conclui o upload das mídias anteriormente descritas. | Não |
| **05** | Agendamento | A aplicação deverá registrar os dados preenchidos no formulário. | Verificar se a aplicação registra os dados preenchidos no formulário de inscrição do passeio em seu banco de dados. | Não |

**Especificação dos testes**

|  |  |  |
| --- | --- | --- |
| Postagem da publicação | | |
| TESTE | ENTRADA | RESULTADO ESPERADO |
| Postagem com os parâmetros necessários | Nome, telefone, resposta das perguntas e mensagem | Publicação enviada com sucesso |
| Postagem sem os parâmetros necessários | Ausência de algum campo obrigatório | Aviso indicando que os campos obrigatórios devem ser preenchidos |

ID 01 – Postagem da Publicação

ID 02 – Integração do Backend (Teste de API) - POSTMAN

|  |  |  |
| --- | --- | --- |
| Integração | | |
| TESTE | ENTRADA | RESULTADO ESPERADO |
| GET - /agenda/ (Privado) | N/A | Retorna todas as datas mesmo aquelas que foram marcadas por alguém, retorna para o admin |
| GET - /agenda/id (Público) | O id referente a uma data específica | Retorna uma data especifica para o admin |
| GET - /agenda/status (Público) | Uma RAW\_JSON com os parâmetros necessários | Retorna todas as datas disponíveis para o cliente |
| POST - /agenda (Privado) | Uma RAW\_JSON com os parâmetros necessários | Insere uma nova data disponível, feito pelo admin |
| DEL - /agenda/{id} (Privado) | O id referente a uma data específica | Manda na url o id da data que quer excluir retorna 200 se deu senão 404 |
| GET - /agendamento/ (Privado) | N/A | Retorna todos os agendamentos feitos pelas pessoas. |
| GET - /agendamento/{id} (Privado) | O id referente a uma data específica | Retorna um agendamento especifico. |
| POST - /agendamento/ (Público) | Um FormData com os parâmetros necessários | Pessoa fazendo agendamento. |
| DEL - /agendamento/{id} (Privado) | O id referente a uma data específica | Remove uma data marcada |
| POST - /fale-conosco/ (Público) | JSON com os parâmetros necessários | Envio do formulário |
| POST - /post/ (Público) | Um FormData com os parâmetros necessários | Insere um novo post |
| GET - /post/ (Público) | N/A | Retorna todos os posts publicos. |
| GET - /files/get/{id} (Público) | O id referente a um arquivo | Retorna um arquivo. |
| POST - /api/user/save (Privado) | N/A | Acesso do administrador para editar os posts. |

ID 03 – Usabilidade

|  |  |  |
| --- | --- | --- |
| Usabilidade | | |
| TESTE | INTERAÇÃO | RESULTADO ESPERADO |
| Botão Início | Clique | Abrir a página inicial |
| Botão Trajeto | Clique | Abrir a página do Trajeto |
| Botão Agendar | Clique | Abrir a página de Agendamento |
| Botão Dúvidas | Clique | Abrir a página de duvidas |
| Botão Blog | Clique | Abrir a página de blog |
| Botão Contato | Clique | Abrir a página de contato |
| Botão Enviar – Página de contato | Clique | Enviar o formulário |

ID 04 – Envio de mídia

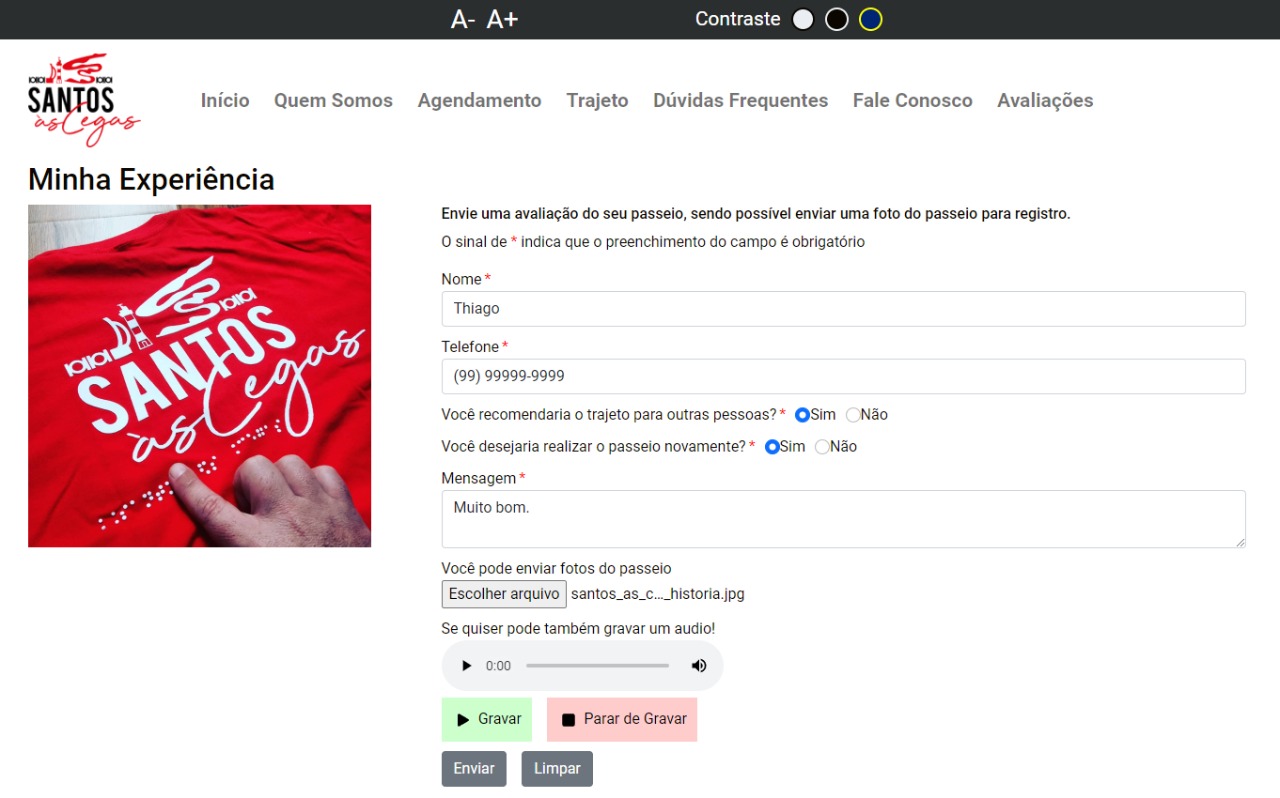
|  |  |  |
| --- | --- | --- |
| Envio de mídia | | |
| TESTE | ENTRADA | RESULTADO ESPERADO |
| Áudio | Envio de um áudio | Ser armazenado e mostrado no blog |
| Imagem | Envio de uma imagem | Ser armazenado e mostrado no blog |

ID 05 – Agendamento

|  |  |  |
| --- | --- | --- |
| Agendamento | | |
| TESTE | ENTRADA | RESULTADO ESPERADO |
| Testar se consegue agendar mesmo com respostas negativas | Envio dos parâmetros necessários | É esperado que não seja possível concluir o agendamento caso o usuário responda negativamente as questões |
| Testar se o agendamento consta no banco de dados | N/A | JSON e constando no banco de dados |
| Testar se com os dados errados ainda ocorre o agendamento |  |  |

**Casos de teste – Postagem da publicação**

(A publicação é feita apenas através do site, e no aplicativo apenas é possível visualizar)



Teste com os parâmetros obrigatórios preenchidos:



Teste sem os campos obrigatórios:



Nesse caso, qualquer campo obrigatório que não for preenchido resultará na mesma mensagem.

A publicação está sendo publicada da forma correta:



**Casos de teste – Integração do Backend**

Os testes foram feitos utilizando-se do Heroku como hospedeiro do backend e com o PostMan, pois com ele é possível automatizar esse processo.  
  
Em todos os casos sem exceção, o envio de qualquer parâmetro errado via API retorna um status HTTP 400 (Bad Request).  
  
Em casos de sucesso, onde a API retorna os dados requisitados, é retornado o status HTTP 200 (OK).

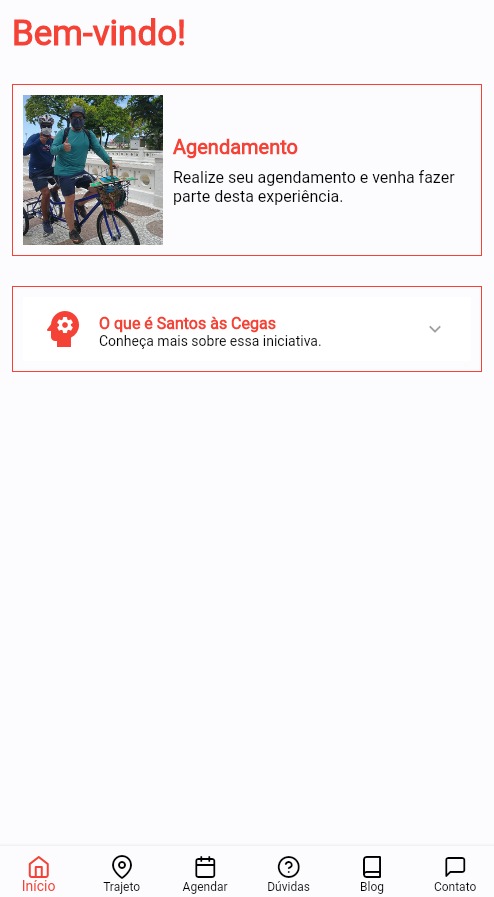
Em todos os casos, onde essa API é de uso privado, ou seja, é obrigatório ter um token de autenticação, o envio sem o token retorna o status HTTP 403 (Forbidden), ou “Não autorizado”.

|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de teste** | **Entrada** | **Retorno 1** | **Retorno 2** |
| GET - /agenda/ (Privado) | N/A | [{"id":1,"date":"2022-12-23T09:00:00.000+00:00","isAvailable":true}] | Caso não haja nenhuma data, é retornado apenas "[]" |
| GET - /agenda/id (Público) | id = 1 | [{"id":1,"date":"2022-12-23T09:00:00.000+00:00","isAvailable":true}] | Caso seja passado um id inexistente, é retornado apenas "[]" |
| GET - /agenda/status (Público) | N/A | [{"id":1,"date":"2022-12-23T09:00:00.000+00:00","isAvailable":true}] | Caso não haja nenhuma data, é retornado apenas "[]" |
| POST - /agenda (Privado) | {"date": "2022-12-23T09:00:00.000+00:00", "isAvailable": true} | {"id": 2,  "date": "2022-12-23T09:00:00.000+00:00",  "isAvailable": true} | N/A |
| DEL - /agenda/{id} (Privado) | id = 1 | "true" (Retorna um HTTP 200) | Caso não haja um id relacionado, é retornando um status HTTP 400 |
| GET - /agendamento/ (Privado) | N/A | {"id": 1,  "name": "1241",  "email": "a@a.com",  "doctorsNote": "1",  "phone": "(11) 11111-1111",  "idDocument": "11111111111112451",  "agenda": {  "id": 1,  "date": "2022-12-23T09:00:00.000+00:00",  "isAvailable": false }}] | N/A |
|  |  |  |  |
| GET - /agendamento/{id} (Privado) | id = 1 | {"id": 1,  "name": "1241",  "email": "a@a.com",  "doctorsNote": "1",  "phone": "(11) 11111-1111",  "idDocument": "11111111111112451",  "agenda": {  "id": 1,  "date": "2022-12-23T09:00:00.000+00:00",  "isAvailable": false}} | Caso não haja nenhuma data, é retornado apenas "[]" |
| POST - /agendamento/ (Público) | "name": "Eduardo",  "email": "Eduardo@gmail.com",  "doctorsNote": "1",  "phone": "(11) 11111-1111",  "idDocument": "675403428" | Status HTTP 200 | Caso haja um parâmetro errado, é retornando um status HTTP 400 |
| DEL - /agendamento/{id} (Privado) | id = 1 | "true" (Retorna um HTTP 200) | Caso não haja um id relacionado, é retornando um status HTTP 400 |
| POST - /fale-conosco/ (Público) | {"name": "joao",  "email": "a@a.com",  "phone": "(11) 99999-9999",  "subject": "TESTE",  "message": "a"} | Status HTTP 200 |  |
| POST - /post/ (Público) | {"name": "joao",  "phone": "(11) 99999-1111",  "message": "Otimo",  "question\_1": "1",  "question\_2": "1",  "audio": File,  "image": File} | Status HTTP 200 | Caso haja um parâmetro errado, é retornando um status HTTP 400 |
| GET - /post/ (Público) | N/A | Retorna todos os posts do Blog\* |  |
| GET - /files/get/{id} (Público) | id = 1 | Retorna o arquivo referenciado pelo id  (status HTTP 200) | Caso não haja um arquivo referenciado, é retornado um status HTTP 400 |

\*[{"id":1,"name":"Joao","phone":"(22) 22222-2222","created\_at":"2022-12-05T18:28:04.000+00:00","isPublic":true,"message":"Muito bom.","image":{"id":2,"fileName":"1670264884177-file.jpg","path":"uploads\\1670264884177-file.jpg","description":null,"created\_at":"2022-12-05T18:28:04.000+00:00"},"audio":{"id":1,"fileName":"1670264884192-file.wav","path":"uploads\\1670264884192-file.wav","description":null,"created\_at":"2022-12-05T18:28:04.000+00:00"},"question\_1":true,"question\_2":true}]  
  
\*\*O JSON é muito extenso para a tabela.

**Casos de teste – Usabilidade**

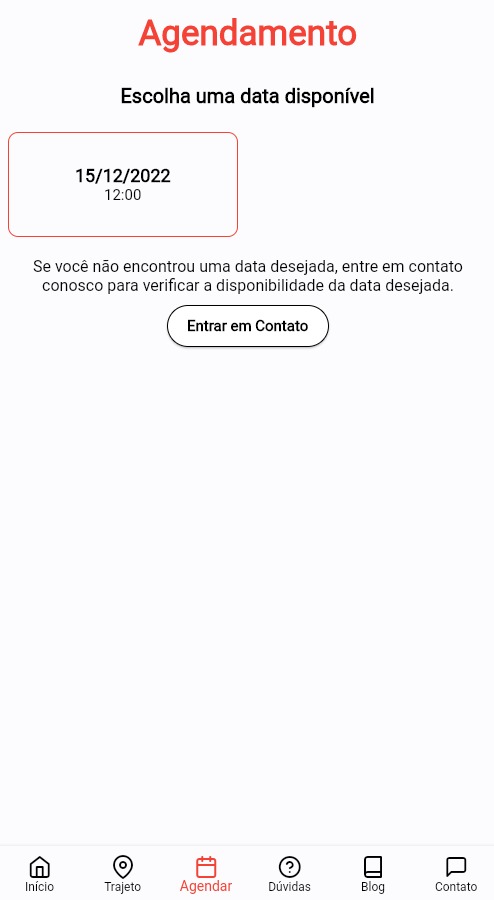
Botão (Início) – Funcionando sem erros, era esperado dele abrir a página inicial do aplicativo:



Botão (Trajeto) – Funcionando sem erros, era esperado dele abrir a página que mostra o percurso no qual é feito o passeio:

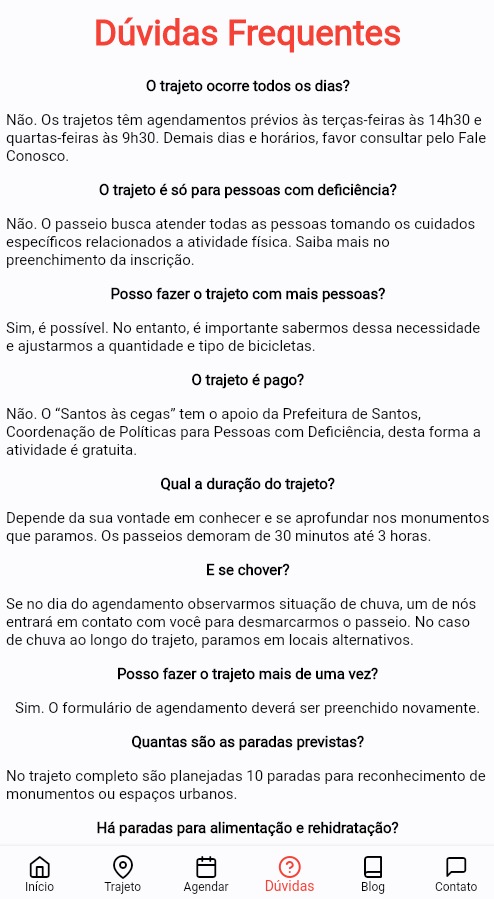


Botão (Agendar) – Funcionando sem erros, era esperado dele abrir a página onde é feito o agendamento:

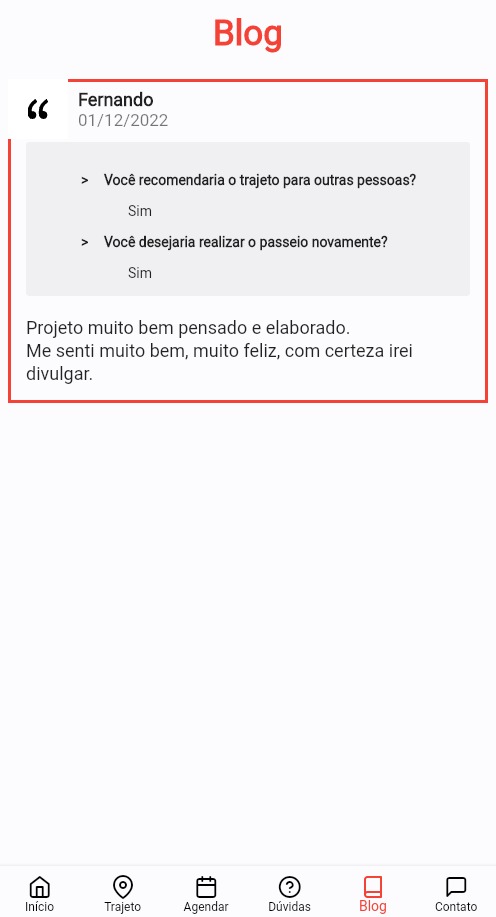


O botão (Entrar em contato) redireciona o usuário a página (Contato), está funcionando de acordo com o esperado.

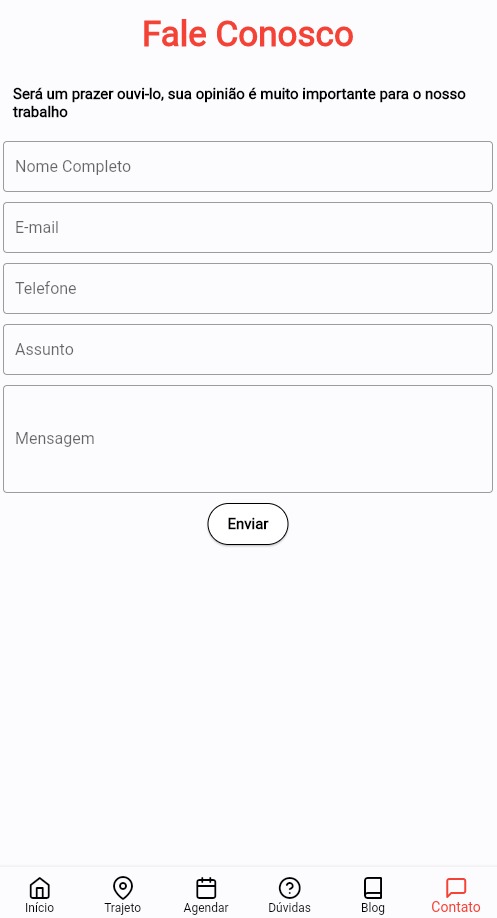
Botão (Dúvidas) – Funcionando sem erros, era esperado dele abrir a página onde são respondidas as dúvidas mais comuns:



Botão (Blog) – Funcionando sem erros, era esperado dele abrir a página onde é possível ver as postagens de outros usuários:



Botão (Blog) – Funcionando sem erros, era esperado dele abrir a página  
onde é possível o usuário entrar em contato:



E após o preenchimento correto dos campos, clicando em enviar, é prevista a seguinte mensagem:



Caso o sistema esteja fora do ar, ou o banco de dados, ou o Backend:



**Casos de teste – Envio de mídias**

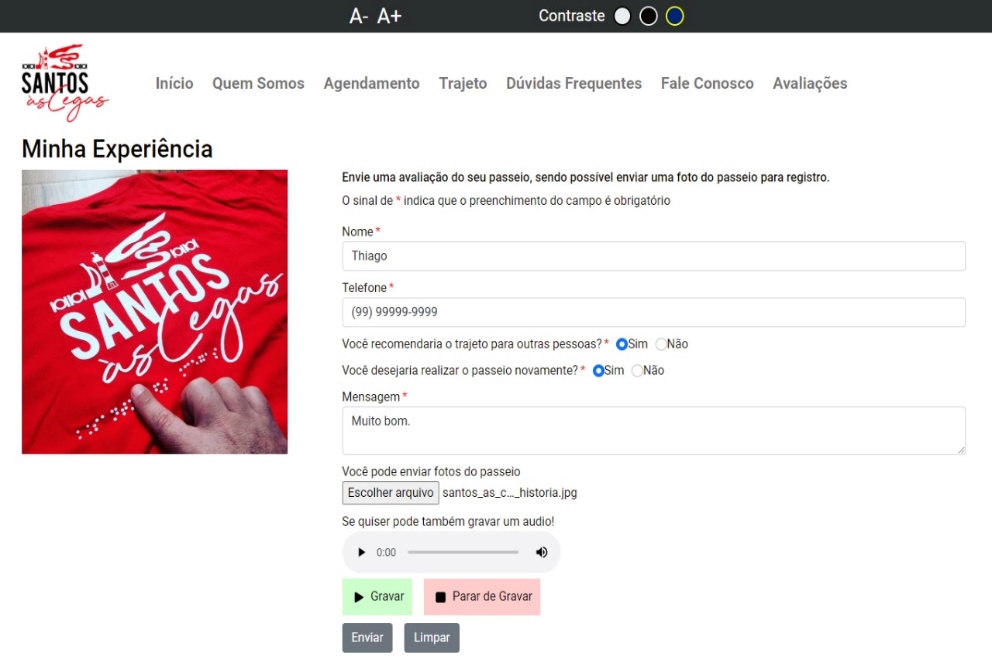
Os testes foram realizados com o envio de imagens e áudios, os quais foram armazenados normalmente no banco de dados e mostrados no blog:  
****

Figura 1 - Envio da postagem

****

Figura 2 - Postagem no blog

Como não são requisitos obrigatórios da postagem, o usuário pode postar a publicação com ou sem as mídias.

**Casos de teste – Agendamento**

Neste caso de teste verificaremos se as funcionalidades do agendamento estão funcionando como deveriam.

|  |  |  |
| --- | --- | --- |
| Agendamento | | |
| TESTE | ENTRADA | RESULTADO ESPERADO |
| Verificação se é possível agendar mesmo com respostas negativas | Envio dos parâmetros necessários | É esperado que não seja possível concluir o agendamento caso o usuário responda negativamente as questões. |
| Verificação se com dados errados ainda ocorre o agendamento | Envio dos parâmetros errados | A aplicação deve verificar e não permitir que o agendamento seja feito. |
| Verificar se o cadastro permite sem os parâmetros necessários | Envio sem os parâmetros obrigatórios | A aplicação deve verificar e não permitir que o agendamento seja feito. |

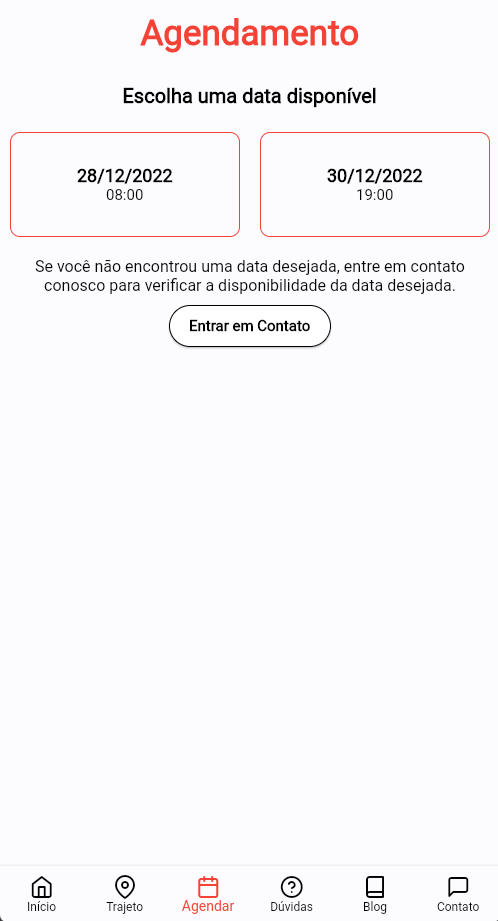
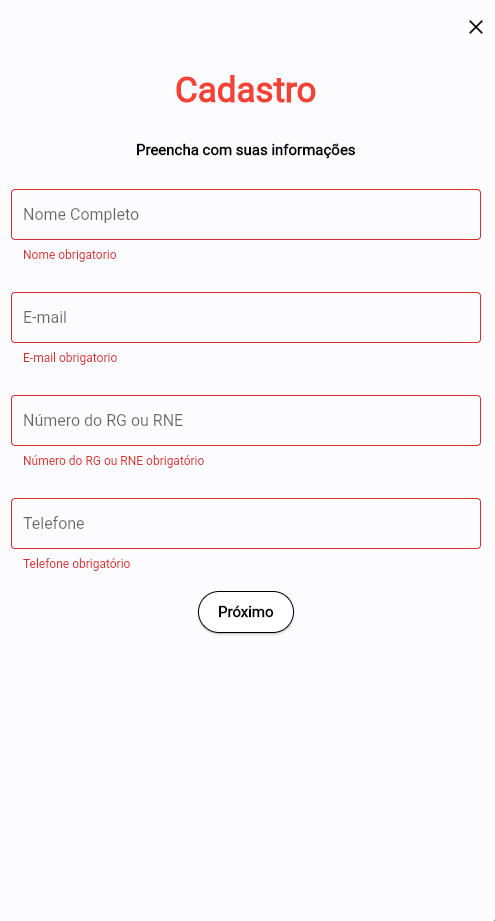
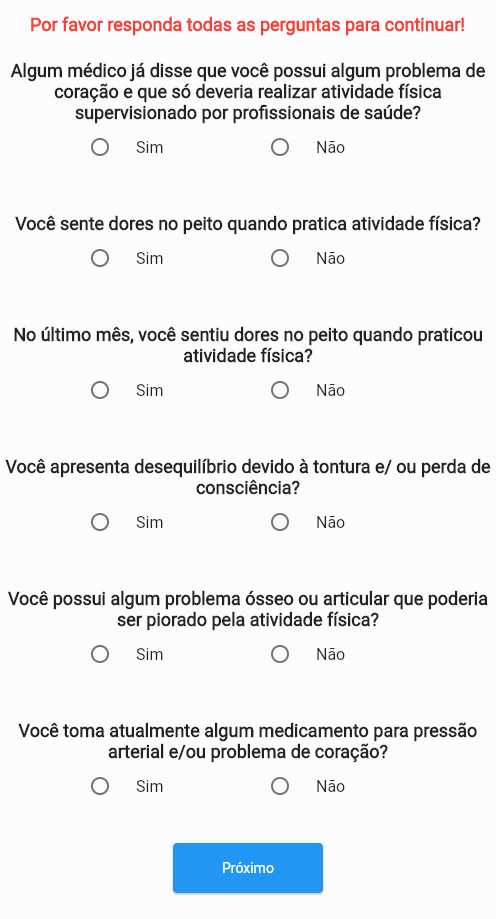
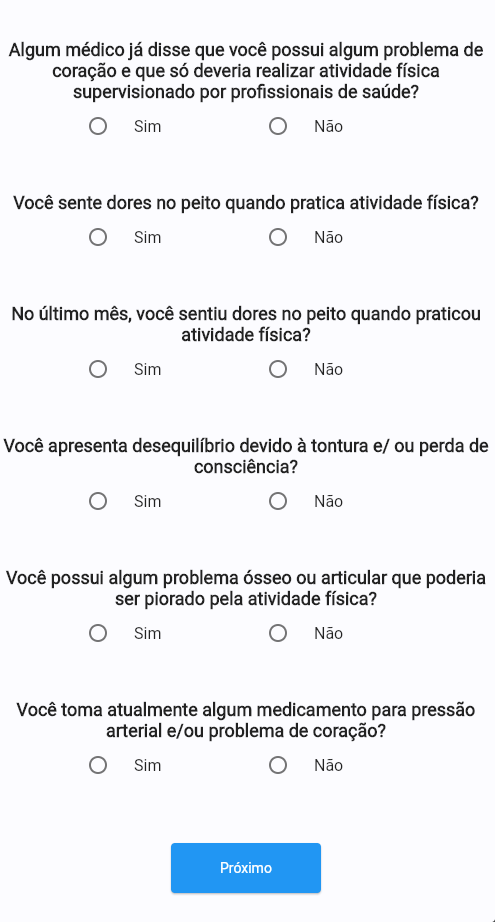


Figura - Página inicial do agendamento

Caso o usuário tente dar continuidade no cadastro sem preencher os parâmetros necessários, a aplicação devidamente não permite que aconteça, assim como o esperado.

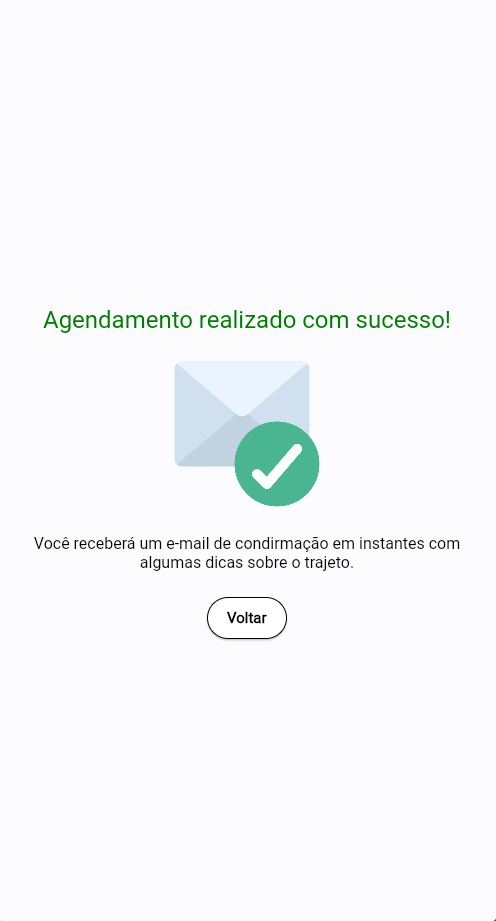
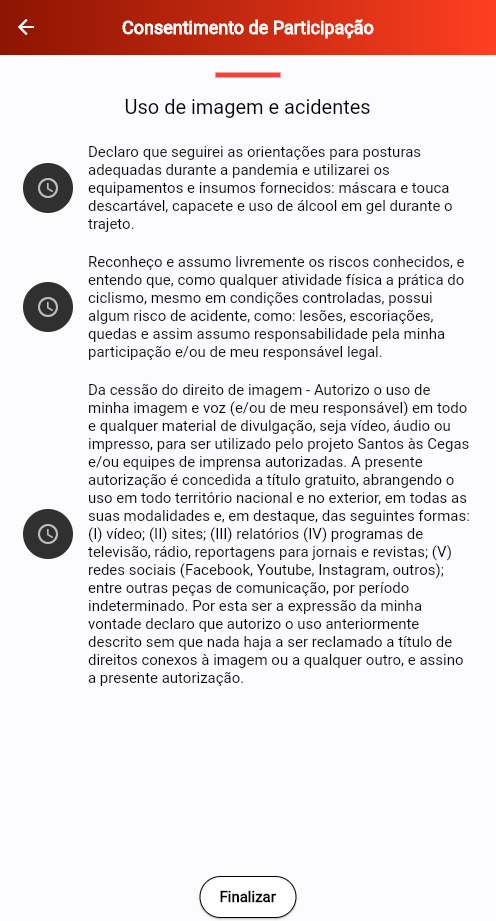


Caso o cadastro seja concluído com sucesso, o usuário será encaminhado as perguntas referentes a saúde, e caso exista alguma resposta negativa, a aplicação não pode dar continuidade ao agendamento, apenas sendo permitida com apresentação de um atestado médico.





Caso todos os casos sejam positivos e o usuário tenha passado por todas as etapas corretamente, será mostrada a imagem do agendamento realizado com sucesso.



Com isso encerramos os testes referentes a aplicação, Backend e banco de dados.